



INTERFACES



Revista digital de la Carrera de Ingeniería de Sistemas de la Universidad de Lima

Edición n.º 12 // Enero-diciembre 2019 // ISSN 1993-4912

Objetivo de *Interfases*

Interfases, "Revista Peruana de Ciencias de la Computación y Sistemas de Información", es un foro regional para especialistas involucrados con la gestión de datos, aprendizaje de máquina, formación en ciencias de la computación y sistemas de información, así como sistemas e ingeniería de *software*.

Público objetivo

- Profesionales e investigadores relacionados con la ingeniería de sistemas y afines.
- Estudiantes y docentes del pregrado y posgrado de ingeniería de sistemas.
- Público interesado.

Interfases

ISSN: 1993-4912 (electrónico)

doi: 10.26439/interfases2019.n012

Periodicidad: Anual

Edición: n.º 12

Fecha publicación: Diciembre 2019

Arbitraje editorial: Revisión por pares de doble
ciego

Directorios y catálogos:

CrossRef, Dialnet, Latindex y DOAJ

Las opiniones expresadas en los artículos
firmados son de exclusiva responsabilidad
de los autores. Los contenidos de la revista
Interfases son de acceso abierto y se
encuentran bajo la licencia Creative Commons.

Universidad de Lima

Av. Javier Prado Este 4600, Urbanización
Monterrico Chico, Lima 33, Perú.
Apartado postal 852, Lima 100, Perú.
Teléfono: (511) 437-6767, anexo 30131
fondoeditorial@ulima.edu.pe
<http://www.ulima.edu.pe/>

Edición, diseño y diagramación: Fondo Editorial

Carrera de Ingeniería de Sistemas

<http://www.ulima.edu.pe/pregrado/ingenieria-de-sistemas>

Contacto: cvilchez@ulima.edu.pe

Director

Julio Alejandro Padilla Solís
Facultad de Ingeniería y Arquitectura, Universidad de Lima, Perú

Editor

Carlos Vílchez-Román
Carrera de Ingeniería de Sistemas, Universidad de Lima, Perú

Director

Julio Alejandro Padilla Solís
padilla@ulima.edu.pe

Editor

Carlos Vilchez-Román
cvilchez@ulima.edu.pe

Comité evaluador

Álvaro Talavera-López
ag.talaveral@up.edu.pe
Universidad del Pacífico, Lima, Perú
<https://www.scopus.com/authid/detail.uri?authorId=37075909800>

Carlos Mugruza-Vassallo
cmugruza@yahoo.com
Universidad Nacional Tecnológica de Lima Sur, Lima, Perú
<https://www.scopus.com/authid/detail.uri?authorId=57190843610>

César Beltrán Castañón
cbeltran@pucp.pe
Pontificia Universidad Católica del Perú, Lima, Perú
<https://www.scopus.com/authid/detail.uri?authorId=55602499700>

Hugo Alatrística Salas
h.alatristas@up.edu.pe
Universidad del Pacífico, Lima, Perú
<https://www.scopus.com/authid/detail.uri?authorId=36970459100>

Ian D. Sanders
sandeid@unisa.ac.za
University of South Africa, Pretoria, South Africa
<https://www.scopus.com/authid/detail.uri?authorId=7004408250>

Juan M. Gutiérrez-Cárdenas
jmgutier@ulima.edu.pe
Universidad de Lima, Lima, Perú
<https://www.scopus.com/authid/detail.uri?authorId=55847492600>

Michael Dorin
michael_andrew.dorin@stud-mail.uni-wuerzburg.de
Julius-Maximilians-Universität Würzburg, Würzburg, Germany

Víctor Andrés Ayma Quirita
vaaymaq@ele.puc-rio.br
Pontificia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brazil
<https://www.scopus.com/authid/detail.uri?authorId=56566776600>

POLÍTICA EDITORIAL

Enfoque y alcance

Interfases “Revista Peruana de Ciencia de la Computación y Sistemas de Información” es un foro regional para especialistas involucrados con la gestión de datos, aprendizaje automático, formación en ciencias de la computación y sistemas de información, así como otras áreas de sistemas e ingeniería de *software*, para publicar investigaciones originales e inéditas. Recibimos artículos y avances de investigación, siempre que cumplan el proceso de revisión por pares. La revista *Interfases* está indizada en CrossRef, Dialnet, Latindex y el Directory of Open Access Journals (DOAJ). La revista se publica una vez al año, en diciembre.

Proceso de revisión por pares

Los manuscritos originales enviados a la revista *Interfases* siguen un proceso de evaluación en dos etapas. En la primera, el editor examina el contenido para determinar si el manuscrito está alineado con el alcance y ha seguido las directrices para autores. Si el manuscrito no es aceptado, se devuelve al autor corresponsal, detallando la decisión adoptada. En la segunda etapa, el manuscrito aceptado se envía a un revisor externo, un experto en el tema del artículo. La segunda evaluación corresponde a una revisión por pares doble ciego, donde el revisor y el autor son anónimos.

El revisor evalúa el contenido del manuscrito y, basándose en su experiencia y conocimiento, adopta una de las siguientes recomendaciones:

1. El manuscrito es aceptado sin cambios o con cambios mínimos que no requieren una segunda evaluación.
2. El manuscrito se acepta, a condición de realizar cambios importantes, de acuerdo con las observaciones del revisor. La versión corregida del manuscrito debe ser aprobada en una segunda revisión.
3. El manuscrito no se acepta por las contribuciones limitadas del estudio u otras consideraciones informadas por el revisor.

El editor envía las observaciones del revisor al autor corresponsal, quien tiene hasta 30 días para realizar los cambios (recomendación 1 y 2) o argumentar por qué no se seguirán las observaciones del revisor (solo la recomendación 2).

Una vez que el revisor recibe el manuscrito corregido, tiene hasta 20 días para informar el resultado de la nueva evaluación, haciendo su recomendación final. Una vez que el editor recibe el segundo informe, toma una decisión para publicar el manuscrito, notificándolo al autor corresponsal.

Cualquier objeción del autor corresponsal a las observaciones del revisor será resuelta por el Comité de Redacción, como instancia final.

La revista se adhiere a los criterios establecidos en las “Directrices sobre buenas prácticas de publicación” del Comité de Ética de Publicaciones (COPE), que especifican las sanciones que se aplicarán en los casos de plagio.

DIRECTRICES PARA AUTORES/AS

Envío del manuscrito

Por favor, dé clic en el enlace “Registro” en el menú superior de esta página. Deberá registrarse (o iniciar sesión) para enviar los archivos de cada manuscrito, siguiendo las instrucciones dadas.

Interfases publica tres tipos de artículos: trabajos de investigación (hasta 6000 palabras), avances en investigación (hasta 2800 palabras) y revisiones (hasta 1500 palabras).

Todos los artículos se envían del mismo modo. Una vez que el editor verifique que el contenido del manuscrito pertenece al ámbito de *Interfases*, pedirá a dos árbitros externos que revisen cada envío. Este proceso (compuesto de dos rondas) toma aproximadamente 2-3 meses, pero dependiendo de la complejidad del manuscrito podría extenderse.

Página del título

La página del título debe incluir:

- Un título conciso e informativo (hasta 30 palabras).
- El nombre completo de cada autor, incluyendo la afiliación institucional, la dirección de correo electrónico y el código ORCID.
- Resumen de 200-250 palabras. El resumen debe indicar la naturaleza y contribución del estudio. Evite las abreviaturas no definidas, las ecuaciones matemáticas o las referencias bibliográficas en el texto del resumen.
- Palabras clave (3-5) separadas por comas. Las palabras clave deben tomarse de la taxonomía de la IEEE Computer Society: <https://www.computer.org/digital-library/journals/sc/tsc-taxonomy-list> o de la taxonomía de ACM: https://dl.acm.org/ccs/ccs_flat.cfm

Texto

Los manuscritos enviados deben estar en Word (.doc o .docx), y aquellos aceptados para ser publicados deben usar la plantilla de *Interfases* LaTeX que estará disponible pronto.

Al redactar el manuscrito, usar la opción de numeración automática para enumerar las páginas. Por favor, evite el uso de funciones de campo. Utilice la función de tabla, no una hoja de cálculo pegada, para hacer tablas. Si escribe su manuscrito con Word, use el editor de ecuaciones o MathType para las ecuaciones.

Tablas

Las tablas son el núcleo de los nuevos hallazgos reportados en la corriente principal de la ciencia, por lo tanto, incluya las tablas que considera son estrictamente necesarias. Todas las tablas se enumeran utilizando números arábigos (por ejemplo, tabla 1, tabla 2...) e incluyen un título que detalla la relevancia de los datos presentados.

Las tablas se mencionan en el orden en que aparecen en el manuscrito. Además del número, el título y los datos, las tablas pueden incluir una nota para detallar la fuente de información, así como explicaciones adicionales que no están incluidas en el manuscrito.

Abreviaturas

Use abreviaturas solo si son necesarias para mejorar la legibilidad de su papel. Debe definir cada abreviatura en la primera mención después de usarla de manera consistente.

Conclusiones

Recuerde que las conclusiones no son la versión narrativa y textual de las tablas incluidas en la sección "Resultados". Por el contrario, las conclusiones reseñan y sintetizan los principales argumentos del artículo. Las conclusiones se extraen de los hallazgos y proporcionan una respuesta adecuada a la pregunta de investigación. Además, las conclusiones incluyen las limitaciones del estudio y sugieren nuevas preguntas y aplicaciones para futuros estudios.

Referencias

Citación

Las referencias citadas en el manuscrito se identifican por números en corchetes [], tal como se detalla a continuación:

- "Estudios previos [1], [2] han documentado la efectividad de..."
- "Varios experimentos recientes [3], [5], [6], [8] han establecido que..."
- "Por ejemplo, ver [7], [9]"

Lista de referencias

Las referencias enumeradas al final del manuscrito incluyen artículos citados anteriormente, ya publicados o aceptados para publicación. Evite citar comunicaciones personales y estudios no publicados, ya que no pueden ser corroborados.

Las referencias están numeradas en el orden en que aparecen en el texto. Algunos ejemplos:

Artículo en actas de conferencias

[1] H. Yuan et al., "Sparse Representation Using Contextual Information for Hyperspectral Image Classification," Proc. 2013 IEEE Conf. Cybernetics (CYBCONF 13), 2013, pp. 138-143.

[2] N. Zhong, "Toward Web Intelligence," Advances in Web Intelligence: 1st Int'l Atlantic Web Intelligence Conf. (AWIC 03), LNCS 2663, 2003, pp. 1-14.

Artículo en una revista o magazine

[3] I.E. Sutherland, R.F. Sproull, and R.A. Schumaker, "A Characterization of Ten Hidden-Surface Algorithms," ACM Computing Surveys, vol. 6, n.º 1, 1974, pp. 1-55.

Libro

[4] W.M. Newman and R.F. Sproull, *Principles of Interactive Computer Graphics*, McGraw-Hill, 1979, p. 402.

Libro (series)

[5] Y. Yao et al., "Web Intelligence (WI): Research Challenges and Trends in the New Information Age," *Web Intelligence: Research and Development*, LNAI 2198, N. Zhong et al., eds., Springer, 2001, pp. 1–17.

DOI - Artículo en una revista

[6] D. Kornack and P. Rakic, "Cell Proliferation without Neurogenesis in Adult Primate Neocortex," *Science*; doi:10.1126/science.1065467.

DOI - Artículo en un acta de conferencia

[7] H. Goto, Y. Hasegawa, and M. Tanaka, "Efficient Scheduling Focusing on the Duality of MPL Representation," *Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS 07)*, 2007; doi:10.1109/SCIS.2007.367670.

Publicación solo en línea

[8] F. Kaplan, "From Baghdad to Manila: Another Lousy Analogy for the Occupation of Iraq," *Slate*, 21 Oct. 2003; <http://slate.msn.com/id/2090114>.

Sitio web

[9] R. Bartle, "Early MUD History," Nov. 1990; www.ludd.luth.se/aber/mud-history.html.

Nota: Los ejemplos se tomaron del IEEE Computer Society Style Guide, pp. 32-34 (<https://ieeecs-media.computer.org/assets/pdf/2016CSStyleGuide.pdf>)

Material suplementario electrónico

Los autores pueden incluir archivos de texto (incluyendo tablas y figuras) y hojas de cálculo como material complementario. Sin embargo, para datos de investigación, es recomendable archivarlos en repositorios de datos. Para el código de *software*, los autores pueden usar plataformas como GitHub o similares.

Datos biográficos

Solo para manuscritos aceptados para ser publicados. Proporcione una breve biografía de cada autor, que se incluirá al final del manuscrito.

Declaración de privacidad

Los nombres y las direcciones de correo electrónico introducidos en esta revista se usarán exclusivamente para los fines establecidos en ella y no se proporcionarán a terceros o para su uso con otros fines.

ÍNDICE

Presentación	9
Método de procesamiento de lenguaje natural y técnicas de minería de datos aplicadas a la clasificación de incidentes informáticos <i>Diana Maribel Garcés-Eslava</i>	11
Predicción de deserción laboral utilizando algoritmos genéticos y redes neuronales artificiales <i>Gonzalo Reyes-Huertas</i>	31
Predicción de postulantes que cometerán fraude interno en una compañía con algoritmos de aprendizaje supervisado <i>Sergio Espinoza-Montalvo</i>	49
Sistema de reconocimiento de gestos faciales captados a través de cámaras para analizar el nivel de satisfacción de clientes en restaurantes <i>Edwin Lara-Lévano</i>	61
Programación por objetivos para el dimensionamiento y la asignación de una flota de camiones en una empresa minera <i>Edmundo Quiroz-Silva</i>	87
Perfiles	113

PRESENTACIÓN



Dr. Julio Alejandro Padilla-Solís
Director de la revista *Interfases*

La nueva edición número doce de la revista *Interfases* presenta en esta oportunidad, y como respuesta a las exigencias de la sociedad y de las organizaciones, artículos relacionados con las áreas de ingeniería de *software*, tecnologías de la información, ingeniería de sistemas y ciencias de la computación.

Es importante resaltar que desde que se implementaron las mejoras en los planes de estudio de la Carrera de Ingeniería de Sistemas de la Universidad de Lima con respecto a las asignaturas de Seminario de Investigación, los artículos de este número de *Interfases* son el resultado de las investigaciones de nuestros alumnos; todos los cuales atraviesan por un proceso de revisión pareada constante a lo largo de por lo menos año y medio, en el que está involucrado un asesor metodológico, un asesor técnico y revisiones de varios otros docentes investigadores.

Los artículos presentados están mayormente relacionados al aprendizaje automático (*machine learning*), que es una disciplina que plantea interesantes retos para los profesionales en ciencias de la computación e ingeniería de sistemas, no solo por la mayor capacidad de procesamiento y menor costo de almacenamiento de los dispositivos digitales, sino también por la mayor disponibilidad de datos y algoritmos (cuyo código fuente muchas veces es compartido por los desarrolladores de *software*). El compartir datos y código fuente coincide con la importancia creciente de la ciencia abierta (*open science*), la cual plantea el acceso libre a los resultados de las investigaciones, las bases de datos empleadas (*datasets*), y en el caso del aprendizaje automático los procedimientos y algoritmos utilizados.

En el primer artículo, "Método de procesamiento de lenguaje natural y técnicas de minería de datos aplicadas a la clasificación de incidentes informáticos", de Diana Maribel

Garcés-Eslava, se describe el uso de técnicas de procesamiento de lenguaje natural y minería de datos para clasificar los incidentes informáticos en una institución educativa; en dicho artículo se usaron reglas de asociación como el algoritmo *a priori* así como árboles de decisión del tipo J48 a fin de realizar una clasificación de variables según su nivel de criticidad. En el segundo artículo, "Predicción de deserción laboral utilizando algoritmos genéticos y redes neuronales artificiales", de Gonzalo Reyes-Huertas, se busca predecir la deserción laboral empleando algoritmos genéticos y redes neuronales, para lo cual el autor usa algoritmos genéticos a fin de seleccionar los mejores hiperparámetros que son utilizados por una red neuronal que permite predecir el problema anteriormente mencionado. El tercer artículo, "Predicción de postulantes que cometerán fraude interno en una compañía con algoritmos de aprendizaje supervisado", de Sergio Espinoza-Montalvo, va en dirección similar utilizar un tipo particular de redes neuronales conocido como redes neuronales de tipo bayesianas, realizando una comparativa entre esta técnica conjuntamente con árboles de decisión del tipo C.45 y *random forest*; para predecir el fraude interno de los postulantes a un puesto de trabajo.

En el cuarto artículo, "Sistema de reconocimiento de gestos faciales captados a través de cámaras para analizar el nivel de satisfacción de clientes en restaurantes", de Edwin Lara-Lévano, se muestra la aplicación de algoritmos de máquinas de soporte vectorial junto con la selección de gestos utilizando la técnica de facial *landmarks*, ambas técnicas fueron utilizadas para reconocer los gestos de satisfacción por parte de los clientes de un restaurante. Finalmente, el quinto artículo, "Programación por objetivos para el dimensionamiento y la asignación de una flota de camiones en una empresa minera", de Edmundo Quiroz-Silva, también presenta una propuesta de modelamiento, pero aplicando el enfoque de programación lineal para el dimensionamiento y asignación en una flota de camiones.

Agradecemos a los autores noveles, quienes pusieron todo su esfuerzo en desarrollar sus trabajos aplicados y plasmarlos en artículos científicos. Dichos trabajos se irán consolidando y podrían seguir formando parte de esta revista u otros repositorios de investigación.

Dejamos a disposición de nuestros lectores esta edición que conlleva un trabajo lento pero arduo y diligente de todo nuestro equipo. Nos despedimos hasta nuestra próxima edición, la cual tiene por objetivo continuar mejorando la calidad académica de las publicaciones apuntando a procesos de indización con otras bases de datos.

MÉTODO DE PROCESAMIENTO DE LENGUAJE NATURAL Y TÉCNICAS DE MINERÍA DE DATOS APLICADAS A LA CLASIFICACIÓN DE INCIDENTES INFORMÁTICOS

Diana Maribel Garcés-Eslava

20120535@aloe.ulima.edu.pe

Universidad de Lima. Lima, Perú

Resumen

El presente artículo plantea una metodología en la que se aplica el procesamiento de lenguaje natural y algoritmos de clasificación, haciendo uso de técnicas de minería de datos e incorporando procedimientos de validación y verificación de significancia, de acuerdo al análisis y selección de los datos, así como también de los resultados, con base en estadísticas de calidad de la información, lo que permite garantizar el porcentaje de efectividad en la construcción del conocimiento. Se utiliza como caso de estudio el análisis de incidentes informáticos en una institución educativa y una base de datos estandarizada de incidentes informáticos históricos recopilados de su área de mesa de servicio, la cual vincula con todos los procesos de tecnologías de la información y se centra en las necesidades de soporte para la realización de las actividades de los empleados. Mientras el requerimiento del usuario no sea resuelto en un tiempo adecuado, el impacto del incidente puede traer inconvenientes laborales de niveles variados, de tal manera que se hace difícil planificar o prevenir la resolución de los incidentes debido a la naturaleza imprevista de los mismos.

Palabras clave: análisis de riesgos, incidentes informáticos, mesa de servicio, minería de datos, nivel de urgencia

Abstract

Method of natural language processing and data mining techniques applied to the classification of computer incidents

This article presents a methodology that applies natural language processing and classification algorithms by using data mining techniques, and incorporating procedures for validation and verification of significance. This is conducted according to the analysis and selection of data and results based on quality statistical analysis, which guarantees the effectiveness percentage in knowledge construction. The analysis of computer incidents within an educational institution and a standardized database of historical computer incidents collected by the Service Desk area is used as case study. Such area is linked to all information technology processes and focuses on the support requirements for the performance of employee activities. As long as users' requirements are not fulfilled in a timely manner, the impact of incidents may give rise to work problems at different levels, making it difficult to plan or prevent incidents resolution due to their unforeseen nature.

Keywords: risk analysis, computer incidents, service desk, data mining, emergency level

1. Introducción

El nivel de avance tecnológico y el incremento exponencial en la capacidad de almacenamiento de los dispositivos informáticos, así como el volumen de información que manejan las organizaciones, hace que el proceso de análisis e interpretación de los datos sea complicado.

De los patrones de eventos y a través del uso de técnicas, algoritmos y mecanismos de validación de minería de datos, se puede obtener información sobre cómo surgen los incidentes informáticos y cómo tratarlos. Es importante que, en una organización, los equipos informáticos trabajen el mayor período posible, con el objetivo de garantizar la completa disponibilidad de los mismos y mantener la calidad de los servicios prestados.

La mesa de servicio, en una institución educativa, se encarga de planificar, organizar y suministrar la entrega de diversos servicios de tecnologías de la información, siendo el contacto principal entre el área de gestión de estos servicios y los usuarios. Además, se responsabiliza de atender todas las incidencias y peticiones de servicio, asegurándose de restaurar, lo más pronto posible, el servicio al cliente con bajo impacto, generando informes al respecto, en los cuales se identifica un catálogo de servicios donde se establecen los recursos y procesos que se necesitan para la ejecución de un servicio, permitiendo el ingreso de solicitudes, ya sea por cliente interno, usuario final o miembros del equipo del área de tecnologías de la información.

Actualmente, existen sistemas que se encargan de tratar los incidentes informáticos en la unidad de mesa de servicio, los cuales son herramientas relevantes para la documentación de estos problemas. Estas herramientas consideran ocurrencias desde la perspectiva de la gestión del servicio, teniendo en cuenta la configuración, los costos y el personal. Los fenómenos dinámicos de un entorno de mesa de servicio dan lugar a oportunidades para obtener conocimientos prácticos a partir de la información histórica, a través de la identificación de los posibles orígenes de las eventualidades y mediante la interpretación de los datos relacionados con la gestión de incidentes.

La minería de datos transforma datos en bruto a información útil, a fin de poder analizar los incidentes informáticos y monitorear fallas, haciendo uso de grandes bases de datos con la descripción detallada del problema resuelto; extrayendo así, rápidamente, el contexto del incidente. Para ello será necesario analizar y evaluar los datos históricos y descubrir correlaciones entre las variables que suelen intervenir con mayor frecuencia en la presentación de incidentes informáticos. El análisis y la predicción de la asociación son dos tareas importantes en la minería de datos, y van a representar dos objetivos principales: la exploración de datos para la comprensión y la construcción de modelos para la clasificación.

2. Herramientas para el procesamiento de datos

2.1 Weka e InfoStat

García *et al.* (2013) mencionan herramientas como Weka (*software* libre) e InfoStat (*software* comercial), las cuales fueron empleadas para el proceso de los datos y la construcción del modelo que realizaron. Sostienen que “Weka es utilizada para la transformación de datos, tareas de agrupamiento, regresión, clasificación y asociación e InfoStat se enfoca más al tratamiento de estadísticas, el estudio de datos multivariado y métodos para modelados estadísticos” (p. 108). Según los estudios que realizaron acerca del uso de minería de datos en el análisis de incidentes informáticos, los autores llegaron a las siguientes conclusiones: a) la minería de datos permite encontrar tendencias, patrones y comportamiento en una amplia base de datos que no suelen identificarse a simple vista; b) la técnica de asociación logra identificar asociaciones en un conjunto de datos, proporcionando reglas de asociación que aportan conocimiento para encontrar la relación entre las variables; c) el modelo otorgado por la técnica de minería de datos ayudará a la preparación de un plan de prevención para disminuir los tickets de incidentes informáticos generados constantemente en la organización.

2.2 ITIL (Biblioteca de Infraestructura de Tecnologías de Información)

En los últimos años, como menciona Baca y Vela (2015), se ha implementado un *software* web libre, ayudando en la mejora de la gestión de tecnologías de la información de incidencias, entendiendo la realidad del área y obteniendo información y estadísticas en tiempo real. Se busca, principalmente, utilizar estándares de calidad que permitan implantar un contexto de trabajo que involucre los procesos y servicios tal como propone ITIL. Al aplicar las buenas prácticas de ITIL, mejora el rendimiento en los servicios que ofrece la organización, así como satisfacer las necesidades del usuario y mejorar el rendimiento del personal para poder estandarizar los procesos y enfocarlos a la gestión de servicios de tecnologías de la información proporcionando calidad.

2.3 GLPI (Gestionnaire Libre de Parc Informatique)

Esta herramienta supone un plan de implementación que busca corregir los servicios para el proceso completo de las solicitudes que se presenten: recepción de incidentes, identificación, aplicación, asignación de la persona encargada de la resolución de los casos presentados, documentación y búsqueda de la mejor solución, ayudando así a restablecer el servicio y realizando un análisis exhaustivo, evitando casos o situaciones similares a las ya reportadas. Ballesteros, Hernández y Sánchez (2010) indican que, gracias a la obtención de información existente —como pueden ser datos estadísticos, resolución de incidentes históricos y reportes de auditoría externa e interna—, recopilada

mediante la búsqueda documental, las técnicas de localización de datos y el análisis de documentos, vamos a poder tener el conocimiento que favorezca al entendimiento de la funcionalidad en el proceso del control de incidentes informáticos.

3. Análisis de las técnicas de agrupamiento y asociación

Corso *et al.* (2014) aplicaron las técnicas de agrupamiento y asociación para identificar patrones y características en incidentes informáticos de equipos tecnológicos y realizaron el estudio en el marco de un laboratorio informático. La primera técnica mencionada propone un análisis exhaustivo de la información y los datos con el objetivo de identificar y dar solución a problemas de clasificación. La segunda técnica se usa para indagar e inspeccionar en un gran conjunto de datos, las reglas de asociación que evidencian las tendencias de las relaciones respecto a los datos de las entidades.

Fombona, Rodríguez y Barriada (2012) consideran que, dado los diversos orígenes sociales y culturales de los usuarios que integran los institutos educativos, tiende a haber una variedad de clientes y usos no uniformes en el manejo de los recursos informáticos. En el contexto de un centro educativo, en este caso la Universidad de Oviedo y en una de sus facultades (Formación del Profesorado y Educación), los autores estudiaron los incidentes informáticos frecuentes que recaen directamente en el personal administrativo y docente. Para este proceso eligieron una ficha de comunicación de incidentes con las siguientes variables, que después iban a recuperar en el sistema: fecha de inicio, docente que comunica la anomalía, lugar, descripción, contacto encargado de la incidencia, estado (resuelto, completado, en ejecución), control y seguimiento (pendiente de la siguiente actualización), comentarios u observaciones. Las anomalías que se presentaron fueron atendidas por el personal y alumnos con beca en informática que se hacían cargo de la resolución, en primera instancia, de las incidencias que eran detectadas en los equipos, así como también de la actualización de *software*, realización de inventarios y buen manejo de los recursos.

Después de utilizar este sistema de gestión propuesto, obtuvieron los siguientes resultados: a) el volumen de incidentes generados fue constante, pero hubo un incremento en el número de equipos informáticos; b) los usuarios supieron emplear mejor los dispositivos informáticos; c) la mayoría de incidentes se relacionaban con problemas de *software*, por lo que se propuso programar mantenimiento de los programas a utilizar y actualización de los servicios centrales; d) se incrementó el nivel de conocimiento de los usuarios respecto a la resolución de los problemas de forma automatizada dado que, a la par, aumentaron los recursos y aplicaciones informáticas; e) se implementó un protocolo claro y reducido, a fin de mantener una comunicación fluida con el coordinador y rapidez en la resolución de los incidentes presentados.

4. Comparación de modelos de clasificación

Barreno (2012) muestra un método para comparar el modelo de regresión logística y el de árbol de clasificación —ambos aportan en la clasificación para el estudio de deserción universitaria— y, de esta manera, establecer qué alumno podría ser clasificado como un desertor potencial, identificando previamente las variables más determinantes para el proceso correcto en la selección. Se utilizó el *software* Minitab 16 y Weka 3-7-2 para obtener los dos modelos. La autora consideró “desertor decisivo” al alumno que presente tres períodos consecutivos académicos sin registro de matrícula, no considerando en la muestra a los alumnos expulsados. Las variables explicativas con sus respectivos valores fueron las siguientes: colegio del que procede (1, colegio particular; 0, otro caso); efectividad examen de admisión (0;1); promedio ponderado acumulado (0;20); proporción de créditos aprobados (0;1); alumno entrega acta de compromiso (1, si el alumno presenta acta de compromiso; 0, en caso contrario); solicitud de recategorización (1, si el alumno ha solicitado recategorización; 0, en caso contrario); ingreso familiar (menos de S/ 2000; entre S/ 2000 y S/ 4000; más de S/ 4000). Se realizó el modelo de árbol de decisión, el cual permite el ingreso de variables de entrada tipo numérico y nominal para la categorización respectiva, es decir, acepta atributos cualitativos sin tener que pasar por una normalización o transformación de los datos. Para este modelo, se utilizaron los parámetros por defecto, excepto por la obtención de un árbol sin podar y las instancias mínimas que se aceptaron por hoja fueron 3.

Estas especificaciones fueron dadas para obtener un árbol más grande en el que se observó la discriminación en la mayor cantidad de variables. Este modelo obtuvo un 95,09 % de clasificaciones correctas utilizando gran parte de las variables: SR, PPA, AC, IF, EEA y PCA, pero esto no se recomienda, ya que indica un posible sobreajuste que afectaría al modelo cuando los datos sean distintos a los de entrenamiento, puesto que habría una generalización de los mismos. Por esto, se realizó nuevamente el árbol, pero utilizando solo 4 variables explicativas: SR, PPA, AC e IF, obteniendo un 94,62 % de clasificaciones correctas (Barreno, 2012). La aplicación de estos modelos de clasificación dio resultados parecidos, puesto que, con las dos técnicas, el análisis respecto a la deserción universitaria dio, al menos, un 94 %.

Barreno (2012) sostiene que, a diferencia del modelo de regresión logística, el árbol de decisión no se representa a través de una fórmula matemática, sino mediante reglas de decisión asociadas a las variables elegidas y la evaluación se realiza de acuerdo a la prioridad de cada regla: solo se realizan las evaluaciones que sean necesarias hasta obtener un clasificador para el alumno evaluado. Dicho autor concluyó que la forma de analizar la deserción estudiantil es diversa, no hay una forma para el análisis del mismo porque depende de los objetivos planteados para dicho análisis, teniendo en consideración el alcance y los recursos de tiempo y económicos a utilizar. Indica que la técnica del árbol de decisión permite obtener reglas claras, sin necesitar una transformación previa en los datos y permite realizar una discretización de datos cuantitativos. Asimismo, para la obtención de buenos resultados, se

deberá reformular el modelo realizando diversas pruebas, adecuando los parámetros, a fin de obtener un árbol de decisión que permita descubrir información relevante, evitando el subaprendizaje y sobreajuste de los datos.

5. Metodología

El sistema propuesto implicó el procesamiento de los datos de texto, en el cual un programa de código Python lee los datos de incidentes de un archivo de entrada, línea por línea, y escribe las palabras clave más frecuentes en el archivo de salida. El algoritmo procesa el incidente informático ingresado, tanto el título como la descripción del registro, y asociará las variables clave determinantes para luego encontrar una relación entre ellas y proporcionar una categorización de los incidentes por nivel de urgencia. Este procesamiento reduce el error de entrenamiento durante la clasificación de los datos. Esto, posteriormente, incide en diferentes aspectos como: ayudar en la identificación del posible origen del incidente, elaborar procedimientos y tareas de mantenimiento a realizar periódicamente, disminuir tiempos muertos, y aprovechar y uniformar el trabajo del área para la planificación de actividades. Los árboles de decisión fueron investigados para aprender de la experiencia pasada (fase de entrenamiento) y luego predecir sujetos para nuevas descripciones de incidentes (fase de prueba).

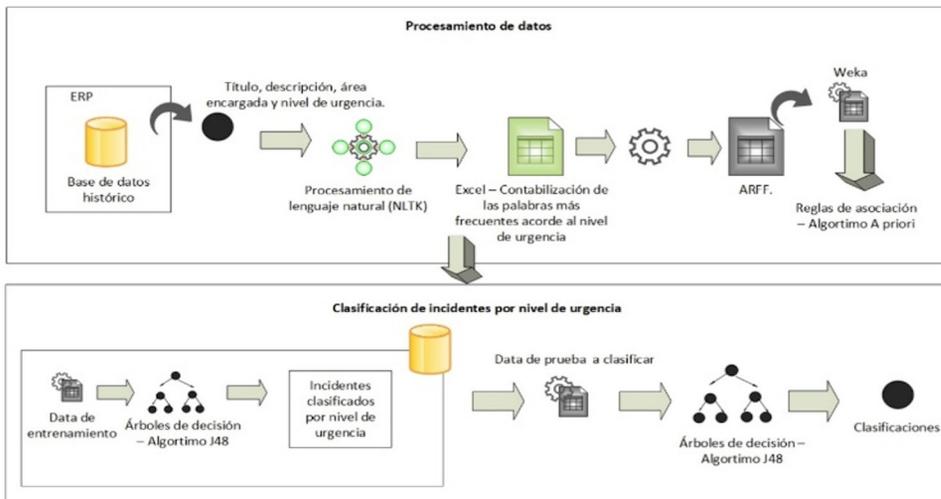


Figura 1. Diagrama de flujo del proceso para la clasificación de incidentes informáticos

Elaboración propia

La aplicación del procesamiento de lenguaje natural y las técnicas de minería de datos a implementar ayudan en la gestión de las bases de datos de incidentes informáticos para recuperar información relevante y, mediante el análisis estadístico y el aprendizaje de los

datos, otorgar una categorización a las incidencias reportadas continuamente, gracias a la identificación de relaciones entre los atributos y las características directamente asociadas al incidente, de tal manera que, al ser estudiadas y analizadas, aporten resultados favorables para el beneficio de la organización.

Se propone la técnica de minería de datos —reglas de asociación— para la búsqueda de relaciones y asociaciones entre los factores más influyentes de los incidentes informáticos. Adicionalmente, se utilizará el algoritmo J48 para la clasificación de los incidentes informáticos utilizando el método de árboles de decisión (ver figura 1). El proceso implica la normalización de los datos de texto, el título y la descripción del incidente, para extraer las palabras clave más recurrentes en una base de datos de incidentes informáticos. Esta normalización reduce el error de entrenamiento durante la clasificación de los datos. Un programa de código Python lee los datos de incidentes del archivo de entrada, línea por línea, y separa los términos clave más representativos de los incidentes informáticos, haciendo una contabilización de palabras clave por título y descripción del incidente.

De acuerdo a los incidentes informáticos reportados en la unidad de mesa de servicio y en base a la propuesta de investigación, se tendrán como objetivo los siguientes puntos: preparar los datos, identificar las variables y búsqueda; identificar asociaciones entre las variables; realizar la clasificación de incidentes informáticos.

Las variables del modelo fueron:

- a) Palabras clave recurrentes: son las palabras clave con mayor recurrencia extraídas del título y descripción de todos los incidentes. i. Detalle pedido. ii. Detalle largo
- b) Área encargada: el tipo de dato es nominal, tiene valores de "Desarrollo", "HelpDesk" y "Producción" de acuerdo al área al que será derivado el incidente informático.
- c) Nivel urgencia: es el atributo más importante pues es la variable base para realizar la clasificación. Es de tipo nominal y tiene 3 valores: 100 (incidente de urgencia alta), 200 (incidente de urgencia media) y 300 (incidente de urgencia baja). Determinar la urgencia de los incidentes de tecnologías de la información sobre la herramienta de gestión es un elemento importante en la atención y gestión de incidentes, puesto que su clasificación permitirá dar relevancia a los incidentes que por su prioridad requieran una solución más pronta.

En base al conjunto de incidentes informáticos pasados, se clasifican los nuevos incidentes y se les asigna un tipo de urgencia de acuerdo a los factores tomados en consideración. Estos niveles de urgencia, en la clasificación de los incidentes informáticos, se toman en cuenta a fin de investigar cuál de ellos tiene un mayor impacto desde la visión de la organización, para que la unidad de mesa de servicio determine el orden deseado para la resolución

de los mismos y se enfoque en lo que es esencial y fundamental para evitar la discontinuidad del negocio.

Para determinar la urgencia, ITIL propone dos conceptos: a) impacto (métrica del efecto que causa en el proceso del negocio); b) criticidad (métrica del tiempo que tiene que pasar para que el impacto en el negocio sea considerado significativo). El nivel de urgencia debe definirse con un valor; también se puede establecer y dar un significado a cada nivel. En una organización los incidentes informáticos pueden ser clasificados en tres niveles: alto, medio y bajo.

6. Procesamiento de lenguaje natural (NLTK)

En primer lugar, se aplicó el método automático NLTK para el procesamiento de datos, a fin de encontrar palabras y expresiones características de un texto, es decir, palabras clave recurrentes en una base de datos de 2000 incidentes informáticos. Mediante la distribución por frecuencias se identificaron automáticamente las palabras más frecuentes de un texto; se utilizó el método "FreqDist" para encontrar las 50 palabras más recurrentes en el título y en la descripción de los incidentes. Se obtuvieron 150 palabras, de las cuales se escogieron las 43 más representativas que ayudaron a leer el título y la descripción del incidente. Dichas palabras clave fueron contabilizadas y clasificadas de acuerdo con el nivel de urgencia correspondiente.

Los campos necesarios para la extracción de datos se obtuvieron utilizando código Python para contar las palabras clave de la información extraída en la base de datos. El programa lee los datos de incidentes en un archivo de entrada en formato .arff, línea por línea, y escribe la clasificación del nivel de urgencia respectivo en el archivo de salida. Para ello, recoge el título y la descripción del incidente de cada línea, pasando cada uno de ellos a una función de conteo, enviando así esa contabilización, la cual devuelve como salida el nivel de urgencia alto, medio o bajo, según corresponda, y, a continuación, poder realizar el entrenamiento.

La función de conteo detecta palabras clave y contabiliza cuántas de cada una ocurren en el título y en la descripción. Estos conteos son enteros pequeños —la mayoría de ellos son cero—, puesto que los atributos tienen una o más palabras clave asociadas.

Una vez normalizados los datos, estos 2000 registros ingresados, incidentes históricos del año 2015 al 2018, fueron almacenados en un archivo con extensión .arff y presentaron los siguientes atributos:

- Palabras recurrentes: 43 palabras clave.
- Área encargada: 553 incidentes se derivan al área de Desarrollo, 1438 al área de HelpDesk y 9 al área de Producción.

- Nivel de urgencia (*class*): 527 incidentes de urgencia alta, 702 de urgencia media y 771 de urgencia baja.

A partir de la selección de cada uno de los atributos, se indica la cantidad de registros que hay en la muestra, de acuerdo a cada valor del atributo. Dado que el conjunto de datos que se está utilizando tiene el mismo conjunto de atributos en cada fila, es adecuado para el formato .arff.

7. Reglas de asociación – Algoritmo *a priori*

La técnica se enfoca en la generación de información en forma de reglas de asociación que tienen lugar comúnmente dado un conjunto transaccional determinado. Se basa en encontrar información útil desconocida en una base de datos y tan pronto como se generen conjuntos frecuentes, permite encontrar reglas de acuerdo a la confianza mínima propuesta por el usuario, dándoles prioridad a las que presentan una confianza mayor o igual a la indicada (Sharma y Bathia, 2016).

La minería de asociación busca relaciones interesantes en un gran conjunto de elementos de datos.

Supongamos que sea $I = \{I_1, I_2, \dots, I_m\}$, un conjunto de m atributos distintos, T la transacción que contiene un grupo de elementos, y D una base de datos con registros de transacciones únicos.

Una regla de asociación es un esquema en forma de $X \Rightarrow Y$, donde X, Y son conjuntos de elementos llamados *itemsets*.

Sean “ x ”, “ y ” dos conjuntos de elementos diferentes en la transacción T , entonces la regla de soporte(s) de asociación se define como la relación de registros que contienen “ x ”, “ y ” con el total de registros. Los dos umbrales se denominan “soporte mínimo” y “confianza mínima”, respectivamente.

Parámetros iniciales: base de datos transaccional con 2000 registros, parámetro de apoyo mínimo con valor 1 y confianza mínima: 90 %

Pasos realizados: a) identificar conjuntos de elementos frecuentes usando el método *a priori*; b) parametrizar los índices de apoyo mínimo y confianza mínima; c) generar las reglas de asociación en base a los límites indicados para cada índice; d) obtener una lista de elementos con 3 variables por cada uno de los registros. Para este caso, se seleccionó el *lift* como el criterio para determinar la validez de las reglas, con un valor mínimo de 1, que se calcula como la confianza de la regla dividida por el soporte del consecuente de la regla. Se encontraron 7 recomendaciones y relaciones dentro de los registros ingresados. El *conf* es el porcentaje de acierto o confianza

que hace relación a las variables elegidas. El algoritmo se inició con el límite superior del soporte, fue disminuyendo en forma incremental y se detuvo al llegar al número especificado de reglas o cuando alcanzó el límite inferior del soporte (ver tabla 1).

Tabla 1
Aplicación de las reglas de decisión

Regla	Descripción	Evaluación
Title=transacción_pendiente 52 ==> category_tree_value=desarrollo 52 conf: (1) Alto (100)	Si el contexto del incidente informático está relacionado a una transacción y, más aún, si se encuentra en estado "pendiente", debe derivarse al área de Desarrollo para que analicen dicha transacción y vean la causa del por qué el pago no se concretó. Es un escenario completo, en el cual el incidente afecta a activos considerados de impacto mayor que influyen directamente a los objetivos de la institución educativa. Estos incidentes deben tener respuesta inmediata.	Coherente: debido a que el valor de la elevación es 1, quiere decir que la ocurrencia no es aleatoria, sino que se debe a una cierta relación entre ellos.
Title=transacción_pendiente 52 ==> category_tree_value=desarrollo 52 conf: (1)	Si el contexto del incidente informático está relacionado a una transacción y, más aún, si se encuentra en estado "pendiente", se determina que es un incidente de urgencia alta. Es necesario tener en cuenta el área a la cual será derivada el incidente para que la regla sea coherente.	Insuficiente: debido a que el valor de la elevación es 0,95, quiere decir que la ocurrencia es aleatoria.
Class=cien 71 ==> category_tree_value=desarrollo 67 conf: (0,94)	No es una regla muy coherente debido a que no indica el título del incidente informático a tratar, por lo que no se puede determinar la categoría y la urgencia del mismo.	No coherente: debido a que el valor de la elevación es 0,94, quiere decir que la ocurrencia es aleatoria.

Elaboración propia

8. Árbol de decisión – Algoritmo J48

El algoritmo para construir el árbol de decisión se logra a partir de los siguientes pasos:

- a) Identificar y comprobar los casos base.
- b) Para cada atributo *a* (palabras clave determinadas previamente), encontrar la ganancia de información normalizada por cada división de *a*.
- c) Dejar que *a_{best}* sea el atributo con la ganancia de información normalizada más alta, en este caso, el atributo clase de acuerdo al nivel de urgencia alto, medio y bajo.

- d) Crear nodos de decisión que dividan a a_best .
- e) Repetir estos nodos en las sublistas obtenidas por la división de a_best y agregar los nodos como hijos de *nodo*.
- f) Entre los atributos explicativos se encuentran las variables cualitativas $\{x_1, x_2 \dots x_n\}$ (palabras clave). Si una nueva instancia de evaluación es tal que su atributo x_2 es menor o igual a c_2 (números binarios de acuerdo a la contabilización de palabras), entonces clasificar la instancia como perteneciente a la categoría A de la clase de interés.

El algoritmo J48 se basa en un método para la ejecución de las funcionalidades propias de la técnica de árboles de decisión, en el cual lee el archivo en formato .arff ingresado con la data total, después procesa los atributos significativos (identificando palabras clave desde el texto del título y de la descripción de los incidentes, realizando la clasificación y contabilización de dichas palabras clave asignándolas a una categoría según su nivel de urgencia), construye el modelo, muestra el árbol generado para clasificación y, por último, indica el porcentaje de instancias correctamente clasificadas e instancias incorrectamente clasificadas. El J48 es supervisado y aprende de cada registro al ir generando el árbol de clasificación.

Se logró obtener una probabilidad de 80,2 % de instancias correctamente clasificadas, lo que indica que el algoritmo de predicción tiene un porcentaje adecuado de confiabilidad que garantiza que la técnica de árboles de decisión evite el sobreajuste y generalice las reglas de decisión identificadas, evitando un aumento de rendimiento, tanto en el conjunto de datos para la prueba, como para los de entrenamiento.

El árbol realiza la clasificación de las instancias de acuerdo a una clase con 43 categorías (palabras clave); los atributos que permiten realizar la clasificación son de tipo cuantitativo y la regla de decisión asociada es una comparación con respecto a un valor constante, ya que se compara el valor del atributo con cada una de las categorías que la conforman. El atributo de clasificación class (nivel de urgencia) es el atributo respuesta. Fueron clasificadas correctamente 80,2 % de las instancias.

A continuación, algunas clasificaciones obtenidas al emplear el árbol de decisión:

- a) Si la palabra clave "matrícula" se encuentra en el título o descripción al menos una vez y la palabra clave "regularización" también figura, el nivel de urgencia va a ser bajo; pero si solo figura "matrícula", va a clasificarse como un incidente con nivel de urgencia medio. Esto quiere decir que la incidencia ingresada para la regularización de una matrícula en el sistema no tiene un impacto mayor en la organización, es decir, no es prioridad.

- b) Si la palabra clave “cobro” figura en el título o descripción al menos una vez y la palabra “adicional” también se presenta, el nivel de urgencia es alto; pero si la palabra “adicional” no figura y la palabra clave “curso” se encuentra al menos una sola vez, entonces estaría clasificado en nivel de urgencia bajo. Esto quiere decir que la incidencia ingresada, debido a que se generó un cobro adicional en la tarjeta del alumno al realizar una matrícula, tiene un impacto mayor, puesto que se tendría que confirmar el doble cobro en la tarjeta del alumno y hacer la devolución, si se requiere.

Observando los resultados que se generaron por la técnica de reglas de asociación, se apunta a que estas puedan, del mismo modo, identificarse en el árbol de decisión, tal como se dio respecto a las reglas coherentes mencionadas anteriormente. Entonces, se puede decir que las reglas fueron importantes y efectivas de acuerdo a las clasificadas por nivel de urgencia por el algoritmo J48. Se obtuvo un porcentaje de 80,2 %, considerado como un porcentaje válido, por que se están generando reglas de ganancia de información y, a veces, el usuario puede utilizar otras palabras para expresar un problema. Para mejorar la técnica sería necesaria una estandarización de los datos, una manera de hacerlos más claros; para ello se debería seguir usando un procesamiento de lenguaje natural que analice y desmenuce las palabras, permitiendo hallar resultados importantes para la clasificación.

9. Validación

Para la validación y verificación de los datos históricos, se aplicarán técnicas de validación usando distintos validadores, con la finalidad de contrastar la precisión de los datos de entrenamiento con la data de prueba. Se pretende determinar si el comportamiento de la clasificación respecto a los incidentes informáticos presenta una alta confiabilidad en los datos propuestos.

Tabla 2

Matriz de confusión del árbol de clasificación

Valor real	Valor predicho		
	Alto	Medio	Bajo
Alto	390	30	107
Medio	45	508	149
Bajo	24	41	706

Elaboración propia

La matriz de confusión, generada a partir de la técnica de validación cruzada, *folds:10*, muestra el número de aciertos representado en los valores de la diagonal; los que están alrededor, serán los errores. Este método ayudará a comparar los resultados y a validar que estos no sean obtenidos al azar, pues los resultados de la clasificación de los datos deberán ser similares a los obtenidos en el entrenamiento con el algoritmo J48. El tomar valores aleatorios y analizar los resultados, permitió posicionarse en un escenario de prueba en el cual se pudieran validar los datos obtenidos por la predicción y clasificación del modelo, lo que facilitó obtener un porcentaje promedio por cada prueba del subconjunto de datos.

En total fueron 10 pruebas que tuvieron una eficacia del 74,6 % como resultado promedio final. Debido a que se tiene gran cantidad de palabras clave y los valores tienden a ser generalizados, el porcentaje de confiabilidad de las instancias correctamente clasificadas disminuyó con las técnicas de validación cruzada en comparación al árbol podado obtenido anteriormente.

Para cada uno de los valores que toma el atributo de clase, en este caso, el nivel de urgencia, el porcentaje de instancias con ese valor son correctamente predichas (TP: verdaderas positivas) y el porcentaje de instancias con diferentes valores son incorrectamente predichas a ese valor, aunque tenían otro (FP: faltas positivas). Las columnas de precisión y recobro son los validadores relacionados con los dos anteriores, que ayudan a probar que los resultados de precisión (fracción de instancias relevantes entre las instancias recuperadas) y recobro (fracción de instancias relevantes que se han recuperado sobre la total cantidad de instancias relevantes) son iguales en el análisis de los datos ya normalizados.

Tabla 3

Resultados de la validación cruzada

Clase	Precisión	Recobro	Medida F
100	0,796	0,653	0,717
200	0,702	0,691	0,738
300	0,694	0,860	0,768
Peso promedio	0,755	0,746	0,744

Elaboración propia

En la matriz de confusión generada a partir de la técnica de validación cruzada, *folds:10*, se obtiene el número de aciertos representado en los valores de la diagonal y el resto vendrían a ser las instancias erróneas. De los 527 incidentes informáticos con urgencia alta, 344 fueron bien clasificados y 183 no. Por otra parte, de los 702 incidentes informáticos con urgencia media, 485 fueron correctamente clasificados y 217 no. Asimismo, de los 771

incidentes informáticos con urgencia baja, 108 no fueron bien clasificados. Podemos decir que respecto a la serie de resultados que nos mostró la técnica de clasificación sobre los datos, contrastándolos con los de la técnica de validación cruzada, ayudan a comparar los resultados y validar que estos no son obtenidos al azar, pues los resultados de los datos son competitivos, uno con otro, teniendo como diferencia solo un 5,6 %.

Para la validación y verificación de los datos, se utilizó un algoritmo realizado en Python en el cual procesa, tanto la técnica de validación cruzada, como las métricas de exactitud y precisión, partiendo los datos en 10 iteraciones para el nivel de urgencia alto (100), medio (200) y bajo (300) por separado, logrando así determinar la confiabilidad en los datos. De igual manera, para los validadores de precisión y exactitud, se aplicaron dos indicadores propios de la técnica de árbol de clasificación: la medición Gini, que determina la probabilidad de una muestra aleatoria que clasifica correctamente si seleccionamos aleatoriamente una etiqueta según la distribución en uno de los nodos (ramas), y la entropía, que calcula la ganancia de información haciendo una división y mide la reducción de la incertidumbre sobre la etiqueta (nodo).

Se entiende que ambas son diferentes técnicas para dividir el nodo en modelos basados en el árbol de decisión. La mayoría de veces el rendimiento de un modelo no tiende a variar mucho si se usa Gini o entropía, pero en términos de computación, se puede decir que la medición de entropía toma más tiempo ya que incluye la función *Log*.

Se dividió aleatoriamente el conjunto disponible de registros en dos partes, uno para entrenar; otro para validar. El modelo de validación se ajusta al conjunto de entrenamiento, dando como resultado un modelo ajustado usado para predecir las respuestas para las observaciones en dicho conjunto.

El conjunto de datos para entrenar se usó a fin de adaptar, tanto el conjunto de validación, como el modelo para estimar el error de la predicción, para la selección correcta del modelo; el conjunto de prueba es usado de igual forma para evaluar el error de generación del modelo final elegido. Entonces, el error del conjunto de pruebas del modelo final subestimaré el verdadero error de la prueba. Se dividió el 80 % de los datos para entrenamiento y el 20 % para la validación y prueba. El error de la prueba se logra estimar de acuerdo al promedio que proporcionan los validadores. Esta sección envuelve una división en un conjunto de observaciones dentro de 10 grupos de igual tamaño, siendo el valor de la media de error cuadrática de 0,3465 MSE; se computan las observaciones en *held-out* partes.

Tabla 4

Resultados por nivel de urgencia

K folds	Nivel de urgencia					
	Alto (100)		Medio (200)		Bajo (300)	
	Gini	Entropía	Gini	Entropía	Gini	Entropía
1	1	0,875	0,975	0,878	1	0,927
2	0,950	0,875	1	0,878	1	0,927
3	0,975	0,875	1	0,878	1	0,950
4	0,900	0,875	0,950	0,900	1	0,950
5	0,975	0,875	0,975	0,900	0,975	0,950
6	0,950	0,875	1	0,900	1	0,950
7	0,975	0,875	0,975	0,900	1	0,950
8	0,100	0,875	0,950	0,878	1	0,950
9	0,950	0,875	1	0,878	1	0,949
10	0,900	0,875	0,975	0,878	1	0,949

Elaboración propia

10. Resultados

Como se observa en los resultados se obtuvo el coeficiente de Gini y de entropía, sumando las diferencias por pares entre todos los valores y normalizando los atributos numéricos, haciendo que el valor mínimo y máximo del atributo sean cero y uno, respectivamente, para una transformación de escala. Se observa que todos los valores son iguales en el conjunto, puesto que la probabilidad está entre 0,9 y 1, lo cual indica que la probabilidad de que existan instancias correctamente clasificadas entre el total de registros es alta para el nivel de urgencia alto, de acuerdo a los incidentes informáticos que se puedan presentar en un futuro. Asimismo, se obtuvieron los valores para el índice de entropía con un promedio de 0,875 mostrando que la mayoría de los datos han sido agrupados por muestras en las clases a las que pertenecen, ya que maximiza la pureza de los grupos, tanto como es posible, cada vez que se crea un nuevo nodo del árbol de clasificación.

Mediante el código Python se obtuvieron los porcentajes de precisión y exactitud para los incidentes clasificados con nivel de urgencia alto, obteniendo un buen desempeño, con un 85 %, en el cual se puede observar que los números están relacionados linealmente; para los incidentes clasificados con nivel de urgencia medio y bajo, se alcanzó un valor de 65 %, aproximadamente. De igual forma, se utilizó la función *accuracy score* y se comparó con el índice de Gini que obtuvo entre 0,975 a 1, determinando la exactitud que se refiere a la proporción de las instancias correctamente clasificadas, para el nivel de urgencia medio y bajo, entre el total de los registros; del mismo modo con el índice de entropía que calculó el aporte de las variables utilizadas dando a conocer que, al no usar gran cantidad de variables (palabras clave), permite obtener un buen desempeño.

Estas dos métricas se utilizaron para el conjunto de prueba y esta función calculó la precisión, analizando la fracción (predeterminada) o el recuento (*normalize=False*) de las predicciones correctas. Dado que *normalize==True* devuelve las muestras clasificadas correctamente (*float*), de lo contrario devuelve el número de muestras incorrectamente clasificadas (*int*).

El mejor rendimiento fue 1 con *normalize==True* y el número de muestras con *normalize==False*. Esto indica que, el conjunto de etiquetas predicho para una muestra, coincide exactamente con el conjunto correspondiente de etiquetas en la matriz de indicador de cada etiqueta/matriz dispersa de datos, y que todo el conjunto de etiquetas pronosticadas para una muestra coincide estrictamente con el conjunto verdadero de etiquetas tal que la precisión es más próxima a 1.0.

Dado que se tuvo como objetivo utilizar un sistema para la correcta clasificación de los incidentes informáticos, de acuerdo al nivel de urgencia alto, medio o bajo, se realizaron pruebas en las cuales los resultados experimentales, al utilizar la técnica de *cross validation* (validación cruzada) con 10 iteraciones y obteniendo los índices de Gini y de entropía, mostraron que todos los árboles de decisión alcanzaron un porcentaje aceptable de exactitud y precisión, teniendo, además, beneficios de velocidad y buen rendimiento.

Uno de los problemas para la minería de datos, en un enfoque basado en palabras clave, es que se contaba con una gran cantidad de atributos que la mayoría de métodos de minería no podían manejar, por lo que se pudo analizar y probar el rendimiento disminuyendo las variables de 50 a 43 atributos, logrando mayor precisión predictiva y menor error cuadrático. Asimismo, los resultados indican que el J48 produce una solución rápida utilizando la técnica de *pruning*, puesto que la construcción del modelo tardó 0,1 segundos, obteniendo un 80,2 % de instancias correctamente clasificadas.

11. Conclusiones

Para efectos del trabajo, se propuso un análisis de los incidentes de tecnologías de la información utilizando datos de una institución educativa. En la actualidad, no se identifica correctamente la prioridad de los incidentes entrantes y se gasta mucho tiempo en resolver el problema y restaurar los servicios, dada la clasificación manual de los mismos.

Dicho esto, se propuso clasificar el título y la descripción de un incidente usando recuentos de palabras clave y algoritmos de clasificación, tales como reglas de asociación y árboles de decisión, con el objetivo de determinar la mejor categorización de los incidentes con un buen porcentaje de confiabilidad, velocidad y precisión. El modelo entrenado analizó el título y la descripción para predecir las posibles variables que podrían estar incluidas en los incidentes. Todo esto implicó la preparación de datos, selección de métodos y exploración de datos.

Los árboles de decisión alcanzaron un 80,2 % de precisión, por lo tanto, es una precisión aceptable, teniendo en cuenta los beneficios de velocidad y mejora en la automatización del proceso. Se mantuvo el enfoque basado en palabras clave para la clasificación del árbol de decisión de los incidentes informáticos de una institución educativa.

Uno de los problemas de este enfoque fue el número elevado de atributos, dado que los métodos de minería no los podían manejar. Sin embargo, se logró analizar y resumir los datos a 43 atributos con más recurrencia en la base de datos. Es así que el modelo propuesto es útil como un sistema de clasificación para predecir el tema de incidentes de tecnologías de la información en aplicaciones del mundo real.

12. Recomendaciones

En el futuro, sería de interés considerar otros algoritmos de aprendizaje automático, tal como la técnica de regresión lineal o análisis de conglomerados, y comparar el resultado en cuanto a precisión y exactitud con la técnica de árboles de decisión. Además, se necesitaría un sistema que incorpore todo el procesamiento, incluido un modelo desarrollado que enlace directamente las incidencias informáticas entrantes, para que una vez que se ingrese el problema, este sea clasificado automáticamente según el nivel de urgencia correspondiente, así pueda ser atendido de acuerdo a la prioridad determinada de manera automática.

El sistema también se puede vincular a una base de datos de conocimiento que ayudaría a resolver incidentes futuros basados en como se resolvieron problemas similares. El modelo de conocimiento generado brinda la clasificación adecuada de los incidentes de acuerdo a su nivel de urgencia alto, medio y bajo, lo que permitiría la elaboración de un plan preventivo que disminuya la generación de solicitudes y el tiempo de resolución del mismo, para actuar eficientemente frente a las nuevas incidencias que se presentan. Asimismo, esto ayudaría a que, posteriormente, se incentiven diferentes aspectos en una mesa de servicio, como la ayuda en la identificación del posible origen del incidente, la elaboración de procedimientos y tareas de mantenimiento a realizar periódicamente, y aprovechar y uniformar el trabajo del área para la planificación de actividades.

Como trabajo futuro se planea incluir nuevas variables que permitan una clasificación más personalizada de acuerdo al incidente, tomando en cuenta otros criterios, tal como el impacto económico que genera el problema a la organización; el período del año en el que se presenta mayor cantidad de incidentes y en qué situaciones de la institución educativa, por ejemplo, en época de matrícula, en las cuales se deba clasificar con mayor nivel de prioridad determinados incidentes para el buen rendimiento de los servicios; el rol de la persona que ingresa el requerimiento, dado que existe una estructura del personal de la organización que debe ser considerada para fidelizar al usuario brindándole servicios de calidad y priorizando sus solicitudes, en caso sea necesario; etc.

De igual forma, sería conveniente identificar la causa raíz de acuerdo a patrones concurrentes de la clasificación propuesta respecto a las palabras clave obtenidas de las incidencias, disminuyendo los tiempos en la resolución e identificando características relevantes en la presentación de los mismos.

Referencias

- Baca, Y., y Vela, G. (2015). *Diseño e implementación de procesos basados en ITIL v3 para la gestión de servicios de TI del área de Service Desk de la Facultad de Ingeniería y Arquitectura* (tesis de licenciatura). Escuela Profesional de Ingeniería de Computación y Sistemas, Universidad de Lima.
- Ballesteros, J., Hernández C., y Sánchez, S. (2010). *Propuesta para la mejora del proceso de control de incidencias dentro de una mesa de ayuda bajo el ciclo Deming* (tesis de licenciatura). Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas, Instituto Politécnico Nacional.
- Barreno, E. (2012). Análisis comparativo de modelos de clasificación en el estudio de la deserción universitaria. *Interfases*, (5), 45-82. doi:10.26439/interfases2012.n005.149
- Corso, C., García, A., Ciceri, L., y Romero, F. (2014). Minería de datos aplicada a la detección de factores para la prevención de incidentes informáticos. XVI Workshop de Investigadores en Ciencias de la Computación (pp. 168-172). Universidad Tecnológica Nacional. Recuperado de http://sedici.unlp.edu.ar/bitstream/handle/10915/41982/Documento_completo.pdf?sequence=1
- Fombona Cadavieco, J., Rodríguez Pérez, C., y Barriada Fernández, C. (2012). Gestión de incidencias informáticas: el caso de la Universidad de Oviedo y la Facultad de Formación del Profesorado. *Revista de Universidad y Sociedad del Conocimiento*, 9(2), 100-114. Recuperado de <http://www.raco.cat/index.php/RUSC/article/viewFile/284627/372853>
- García, A., Corso, C., Gibellini, F., y Rapallini, M. (2013). Análisis de incidentes informáticos usando modelos de asociación y métodos del análisis de datos multivariante. *XV Workshop de Investigadores en Ciencias de la Computación* (pp.107-111). Universidad Tecnológica Nacional. Recuperado de http://sedici.unlp.edu.ar/bitstream/handle/10915/27107/Documento_completo.pdf?sequence=1
- Gupta, R., Prasad, K. H., y Mohania, M. (2008). Automating ITSM Incident Management Process. *International Conference on Autonomic Computing* (pp. 141-150). Chicago: IEEE. doi:10.1109/icac.2008.22

Han J., y Kamber, M. (2006). *Data Mining: Concepts and Techniques*. (2.ª ed.). San Francisco: Morgan Kaufmann Publishers.

Han, J., Kamber, M., y Pei, J. (2012). *Data Mining. Concepts and Techniques*. (3.ª ed.). Boston: Morgan Kaufmann Publishers/Elsevier.

Sharma, S. & Bhatia, S. (2016). Analysis of association rule in data mining. *ICTS 16 Second International Conference on Information and Communication Technology for Competitive Strategies*, Udaipur, India, March 04-05, 2016. doi: 10.1145/2905055.2905238

PREDICCIÓN DE DESERCIÓN LABORAL UTILIZANDO ALGORITMOS GENÉTICOS Y REDES NEURONALES ARTIFICIALES

Gonzalo Reyes-Huertas

20120535@aloe.ulima.edu.pe

Universidad de Lima. Lima, Perú

Resumen

El objetivo del trabajo de investigación es desarrollar una herramienta que permita identificar la posible deserción de un empleado, entendiendo la deserción laboral como un problema internacional. El método propuesto consiste en un algoritmo genético que identifica las variables relevantes y mejora la arquitectura de una red neuronal artificial como solución. Las variables seleccionadas por la herramienta concordaban con las variables recopiladas de distintos estudios, descubriéndose que no todas eran consideradas en dichos estudios (e.g., distancia del hogar al trabajo y años totales trabajando). Asimismo, las variables y la arquitectura seleccionadas por la herramienta permitieron predecir la deserción laboral hasta un 88,92 % de exactitud.

Palabras clave: red neuronal artificial, algoritmo genético, rotación de personal, arquitectura de redes neuronales

Abstract

Predicting job abandonment through genetic algorithms and artificial neural networks

This research work aims to develop a tool to identify employees who might abandon their position, because job abandonment is considered an international problem. The proposed method consists of a genetic algorithm that allows identifying the significant variables and improving the architecture of an artificial neural network as a solution. The variables selected by the tool were similar to those collected from different studies but not all of them were considered in such studies (e.g., distance between home and workplace, and years of employment). Likewise, the variables and architecture selected by the tool allowed to predict job abandonment up to 88.92 % accuracy rate.

Keywords: artificial neural network, genetic algorithm, employee turnover, neural network architecture

1. Introducción

Se considera la deserción laboral como un problema internacional persistente, incluso a pesar de los avances y la implementación de sistemas de gratificación y retención de empleados. Según Fidalgo y Gouveia (2011), al año 2009, más del 50 % de empleados inmobiliarios en las organizaciones portuguesas se retiró antes de los 4 años de vinculación. En Estados Unidos, en el sector de TI, los empleados tienden a permanecer entre 3 y 5 años en la misma compañía, independientemente de la generación a la que pertenecen (Burnes, 2006). Igualmente, para las empresas de *outsourcing*, en India, la tasa de deserción laboral es alta; especialmente para los *call centers*, alcanzando tasas hasta de 40 % en algunas firmas (Anand, Saravanasudhan y Vijesh, 2012).

En el caso del Perú, es el país con la tercera tasa de rotación de personal más elevada de Latinoamérica. A 2017, el índice de rotación fue del 18 %, mientras que el de otros países oscilaba entre 5-10 % (“Perú es el tercer país con la mayor rotación de personal en Latam”, 2017). Adicionalmente, el promedio de permanencia, a 2014, fue de 3 años; ese mismo año la tasa de rotación en puestos de altos cargos llegó al 25 % (“Rotación laboral de ejecutivos en altos mandos alcanza el 25 %”, 2014). De acuerdo a Jack Welch (empresario estadounidense), citado en Flores (2014), el costo de rotar a un gerente equivale a 5 años de su remuneración. Este tipo de impacto financiero se refleja en la pérdida de 2 millones de dólares por rotación de personal en Arequipa (“Mal clima laboral genera pérdidas millonarias”, 2016). Estos estudios demuestran que en el Perú existe un problema de deserción laboral, el cual dificulta el cumplimiento de los objetivos que se mencionan a continuación.

Para el 2030, el Perú desea “lograr el empleo pleno y productivo y el trabajo decente para todas las mujeres y los hombres, incluidos los jóvenes [...]” (Naciones Unidas/CEPAL, 2016); así como también “reducir considerablemente la proporción de jóvenes que no están empleados [...]” para el 2020 (Naciones Unidas/CEPAL, 2016), objetivos que, de no cumplirse, significarían la existencia de jóvenes con desventajas profesionales y dificultad para estar empleados.

Este problema afecta tanto a los empleados como a las empresas. Las empresas tienen que invertir tiempo y dinero en la capacitación del nuevo empleado; además, sufren retraso en procesos o proyectos, dependiendo de la importancia del puesto; adicionalmente, se pierde el conocimiento tácito de los empleados retirados (Fidalgo y Gouveia, 2011). Desde el punto de vista del empleado, este tiene que adaptarse a la cultura organizacional de la empresa y aprender el conocimiento interno de la misma para llevar a cabo sus tareas.

La principal dificultad, al momento de clasificar a un empleado como “futuro desertor”, es escoger correctamente las variables significativas que afectan la posibilidad de deserción y qué tanto la afectan. Algunos estudios, en el contexto peruano, coinciden en que los empleados se retiran por falta de motivación, mal clima laboral y falta de incentivos. Sin embargo, estos estudios suelen ser empíricos e individuales para cada variable.

En el presente trabajo de investigación se pretende conocer y seleccionar matemáticamente las principales variables que conllevan a los empleados a abandonar su trabajo, con el fin de poder desarrollar un modelo predictivo con arquitectura óptima que permita clasificar a los posibles empleados desertores.

2. Estado del arte

2.1 Identificación de variables

Según Fidalgo y Gouveia (2011), existen dos tipos de variables que contribuyen a la deserción laboral: aquellas que la empresa puede controlar, tales como salario, horario, línea de carrera, etc.; y las que no puede controlar, reubicación, problemas personales, etc. Los estudios revisados tratan sobre el primer tipo.

Es cierto que las razones por las que los empleados se retiran varían de empresa a empresa, y de región en región (Pawar y Chakravarthy, 2014); sin embargo, varios estudios (principalmente empíricos y cualitativos) con frecuencia consideran similares variables (ver tabla 1).

Tabla 1

Relación de variables consideradas en los diferentes estudios

Variables	Anand <i>et al.</i> (2012)	Ashok (2003)	Burnes (2006)	Firth <i>et al.</i> (2004)	Min (2007)	Pawar y Chakravarthy (2014)	Wei y Hei (2007)
Salario	X	X					
Ambiente laboral	X				X	X	
Línea de carrera	X		X			X	
Satisfacción laboral	X			X			X
Beneficios		X					X
Compromiso organizacional			X	X	X		X
Relación con superiores		X	X	X	X	X	
Stress laboral		X				X	
Estado civil						X	
Género						X	

Elaboración propia

Según Firth, Mellor, Moore y Loquet (2004), las responsabilidades de puestos con personal a cargo deben incluir el monitoreo de carga de trabajo, relación con subordinados y jefes (para reducir el estrés) y las fuentes de motivación intrínsecas y extrínsecas de los empleados. Adicionalmente, Ashok (2003) concuerda con que es responsabilidad de la empresa asegurar la satisfacción de sus empleados. También, autores como Min (2007), Pawar y Chakravarthy (2014), Wei y Hai (2007) enfatizan que la principal razón por la que los empleados se van no es el dinero. Por otro lado, las variables encontradas en los estudios del contexto peruano, en general, fueron (Flores, 2014; "Mal clima laboral genera pérdidas millonarias", 2016; "Rotación laboral de ejecutivos en altos mandos alcanza el 25 %", 2014; Sandoval, 2014): (mal) clima laboral, (ausencia de) beneficios laborales, (bajo) salario, búsqueda de mayor competitividad. Estas variables recolectadas de estudios en el Perú tienen coherencia con las variables descritas en la tabla 1.

2.2 Optimización de redes neuronales artificiales con un algoritmo genético

En esta sección se mencionan los estudios revisados para comprender la integración del algoritmo genético (AG) con la red neuronal artificial (RNA); en cada uno se describe su aporte distintivo.

2.2.1 Finding optimal neural network architecture using genetic algorithms

El objetivo de este artículo es encontrar la arquitectura óptima de una RNA para problemas particulares. Los pesos de las RNA fueron ajustados con *back-propagation* y el criterio de evaluación se basa en la exactitud de clasificación de las mismas. Fizelew *et al.* (2007) consideran 20 RNA como un buen número de población sobre la cual aplicar el AG. Para la selección de cromosomas se aplicó el método del torneo. El resultado fue que la RNA generada por el AG tenía mejor porcentaje para la clasificación de elementos previamente no vistos, que las RNA creadas empírica o aleatoriamente. Los autores concluyeron que el AG ayuda a generar la mejor RNA dependiendo del caso dado y el tipo de datos a utilizar.

2.2.2 Neural architectures optimization and genetic algorithms

El artículo propone una formulación matemática para determinar la función de activación de una red neuronal, el número óptimo de capas ocultas y los pesos de sus nodos. El objetivo es resolver el problema de la optimización. La optimización del desempeño se obtiene minimizando el error (distancia) entre la salida y el valor real (Ettaouil y Ghanou, 2009). Para ello se hizo uso del AG. Entonces, se mapean las RNA a dos cromosomas: matriz de pesos y vector (u), que es un arreglo de variables de decisión. La población inicial se generó aleatoriamente; a cada cromosoma se le asignó un *fitness* que corresponde a la *performance* de la RNA correspondiente, y

se calcula con base en el porcentaje de exactitud obtenido. Para la selección de cromosomas se utilizó el método de la ruleta. Además, debido a que se trataba de dos cromosomas, el *crossover* estaba basado en 2 puntos (generalmente es 1). El operador de mutación permitió mantener la diversidad en los cromosomas y no converger en un máximo local.

2.2.3 Genetic algorithm for neural network architecture optimization

El artículo consultado (Idrissi, Ramchoun, Ghanou y Ettaouil, 2016) propone una formulación matemática multiobjetivo, a fin de determinar el número óptimo de capas ocultas, el número de neuronas en cada capa y el mejor valor de los pesos de sus respectivas salidas. Por lo general, la elección de la arquitectura de la RNA es hecha de forma empírica. El modelo propuesto por los autores fue una RNA *feed forward*, es decir, una RNA en la que sus nodos no interactúan con otros nodos de su misma capa, y *back propagation* para el cálculo de la gradiente. La tasa de aprendizaje empleado fue de 0,1. Asimismo, para el AG, se plantearon dos funciones objetivo: menor número posible de capas ocultas y minimizar el MSE de la RNA. Las restricciones que se consideraron fueron: debe existir al menos una capa oculta y si no se usa una capa oculta, no se usan sus nodos. Para poder calcular el *fitness* con base en ambas funciones objetivo, se escaló el valor resultante de cada una y se sumaron; tras lo cual se calcula su inversa.

3. Metodología

3.1 Procesamiento de datos

Primero, se normalizó la data: se filtraron las variables categóricas; de cada variable no categórica, se calculó la media y la desviación estándar. A cada valor se sustrajo la media y se dividió entre la desviación estándar. Con esto, la media de los datos era 0 y su desviación estándar era igual a 1. De esta forma, se redujo la diferencia entre la magnitud de las variables no-categóricas a considerar. En segundo lugar, se eliminaron las columnas/características que no aportaban a la solución. Algunos de los valores eran constantes para todos los registros o representaban lo mismo (ej., *daily rate*, *hourly rate*, *monthly rate* que representaban ingreso económico). Tercero, se aplicaron 2 tipos de codificación a las variables categóricas: para las variables que presentaban 2 categorías (ej., ¿trabaja sobretiempo?: sí o no), se toma una categoría como 0 y la otra como 1 (ej., no = 0, sí = 1). Para el caso de variables con más de 2 categorías (ej., departamento/área), se aplicó el *one hot encoding*, que consiste en crear una columna por cada categoría y, por cada registro (fila), se marca con 1 la categoría a la que pertenece y se pone 0 en las demás.

3.2 Propuesta de solución

Una vez aplicado el procesamiento de datos, estos pasaron por los modelos de aprendizaje elegidos del estado del arte, en este caso, redes neuronales artificiales (RNA) y algoritmo genético (AG). Para tener una mayor visibilidad de la herramienta de solución, en la figura 1 se muestra un gráfico de bloques con los módulos que se trabajaron.

Tomando en cuenta lo antes señalado, el programa se dividió en 3 módulos: el primero (explicado en la figura 1), dedicado a la lectura, normalización, codificación y partición de datos; el segundo, dedicado a entrenar, validar y probar la RNA; y el tercero, dedicado al AG, que invoca al módulo de la RNA cuando se desea convertir el cromosoma a RNA y ejecutar métodos exclusivos de la RNA (entrenamiento, validación, prueba, etc.).

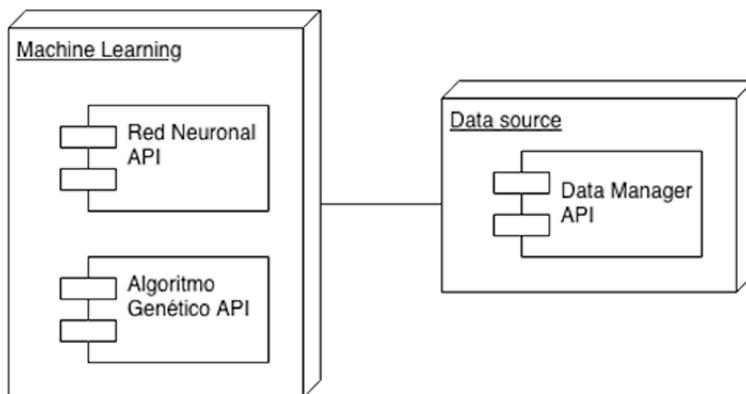


Figura 1. Representación en bloques de la solución

Elaboración propia

En la figura 2 se detalla la integración entre ambos modelos (AG y RNA). Inicialmente se debe tener una población de cromosomas, donde cada uno de ellos representa una RNA. Primero, se transforma cada cromosoma a la RNA que representa (considerando variables de entrada y capas ocultas), para entrenarla y validarla. Una vez entrenada y validada, se procede a hacer la predicción; luego, al ajuste del cromosoma se le asigna el valor de la exactitud resultante de la predicción. Los pasos mencionados se repiten por cada cromosoma en la población. Después de haber calculado el ajuste de todos los cromosomas de la población, se aplican los operadores genéticos escogidos (*crossover* y *mutación*) y se seleccionan los cromosomas de la siguiente generación.

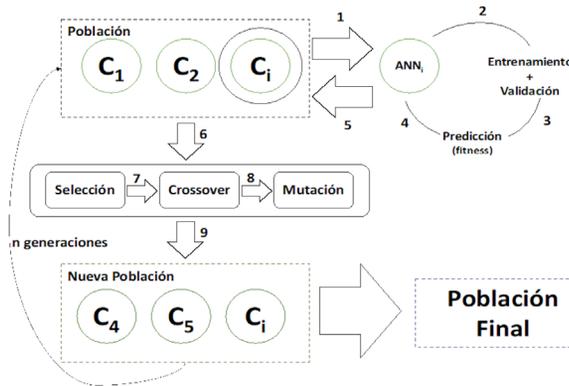


Figura 2. Representación gráfica de la propuesta de solución

Elaboración propia

El conjunto de pasos anteriores se itera n veces, hasta que el modelo ha convergido y se obtiene la población final, dentro de la cual estará el cromosoma que representa a la RNA con mayor exactitud.

Una ejecución significa haber iterado n veces el AG y haber obtenido la población final. Durante el desarrollo de la solución, se ejecutó más de una vez el programa.

3.3 Representación del cromosoma

Cada cromosoma de la población representa una RNA; dicha representación consiste en un vector (a) el cual a su vez estará compuesto por dos vectores, (b) y (c), donde:

$$b = (b_0, b_1, \dots, b_i, \dots, b_n) \quad (1)$$

b_i indica si la variable x_i se considera ($b_i = 1$) o no ($b_i = 0$)

$$c = (c_0, c_1, \dots, c_j, \dots, c_m) \quad (2)$$

donde el tamaño m del vector (c) representa la cantidad de capas ocultas y c_j es la cantidad de nodos en la capa oculta j .

Como se observa, los elementos del cromosoma corresponden a los elementos de la RNA, previamente, seleccionados y clasificados como variables. Debido a que los vectores (b) y (c) representan un cromosoma, ambos están sujetos a cambios de operadores genéticos en cada generación. La función objetivo del cromosoma se calcula con base en la exactitud de la red neuronal. El valor del ajuste varía entre 0 y 1, siendo 1 más favorable.

3.4 Restricciones del modelo

Después de iterar múltiples veces, se optó por mantener las 3 restricciones que se indican a continuación.

- Para las variables de entrada: $10 \leq n \leq 20$
donde n es la cantidad de variables de entradas que tiene la RNA.
- Las restricciones para las capas ocultas son: $1 \leq m \leq 3$
donde m es la cantidad de capas ocultas que tiene la RNA.
- Las restricciones para los nodos ocultos son: $5 \leq k_i \leq 10$
donde k_i es la cantidad de nodos en la i -ésima capa oculta.

3.5 Operadores genéticos utilizados

Como se indicó, los operadores genéticos a utilizar eran *crossover* y mutación. A continuación, se presenta la descripción y el pseudocódigo de sus implementaciones en el módulo del AG.

La función *crossover_single* intercambia los genes de dos cromosomas, con lo que se generan dos cromosomas hijos con genes combinados de los dos cromosomas originales.

Input: l, r : cromosomas padres

Output: cromosomas con genes intercambiados por el operador *crossover*, hijos de l y r

Variables:

$c1, c2$: cromosomas hijos de l y r

i : índice que indica el punto de intercambio de genes de los cromosomas

crossover_single(l, r):

```
i = random(min(l.size, r.size))
c1 = l[0:i] + r[i:r.size]
c2 = r[0:i] + l[i:l.size]
crossover_single = c1, c2
```

La función *mutate_single* altera el valor de algún gen del cromosoma, dependiendo de un porcentaje de probabilidad.

Input: c: cromosoma a mutar

Variables: gene: gen del cromosoma

Constantes:

MIN: valor mínimo posible para el gen del cromosoma

MAX: valor máximo posible para el gen del cromosoma

MUTATION_RATE: probabilidad de que el gen sea mutado

```
mutate_single(c) :
    foreach gene in c do
        if random(0,1) < MUTATION_RATE then
            gene = random(MIN, MAX)
```

Dado que el cromosoma está compuesto por dos vectores: (b) y (c), ambos operadores se aplican a cada vector. Es decir, tanto las variables de entrada como los nodos y capas ocultas están sujetos a posibles modificaciones por el AG en cada generación.

3.6 Método de selección

El método de selección de cromosomas implementado fue el del torneo. Con este método, se seleccionaban los cromosomas que pasarían a la siguiente generación. A continuación, se detalla su uso y pseudocódigo.

La función *select_tournament* selecciona aleatoriamente cromosomas de la población, y devuelve el cromosoma con el ajuste más alto.

Input: p: población de cromosomas

Output: cromosoma con mayor ajuste (precisión)

Variables:

new_p: arreglo de cromosomas seleccionados

i: contador de cromosomas seleccionados

Constantes: TOURNAMENT_SELECTION_SIZE: tamaño máximo de cromosomas seleccionados

```

select_tournament(p):

    new_p = []

    i = 0

    while i != TOURNAMENT_SELECTION_SIZE do

        c = p[random(p.size)]

        new_pop.append(c)

        i = i + 1

    select_tournament = new_p.sorted()[0]

```

3.7 Configuración de la red neuronal

Para las RNA, se utilizó la función sigmoidea como función de activación, empleada en la investigación de Islam, Baharudin, Raza y Nallagownden (2014). De igual manera, el algoritmo de *back-propagation* para el ajuste de pesos, que fue el algoritmo validado en las investigaciones de Ettaouil y Ghanou (2009), Fiszlewe *et al.* (2007), Idrissi *et al.* (2016) e Islam *et al.* (2014).

Según investigaciones revisadas (Fiszlewe *et al.*, 2007; Echegaray y Barrios, 2015), se utilizó el método de validación cruzada para la validación de RNA, por lo que fue el método que se tuvo en cuenta para validar a la RNA durante su entrenamiento, de tal manera que, si estuviera en riesgo de subajuste o sobreajuste, tenga una menor precisión. Esto se debe a que al aplicar *cross-validation* se calcula la precisión promedio de K ejecuciones ($K = 10$ en este caso) con data particionada de diferente manera en cada iteración.

3.8 Configuración del algoritmo genético

Como ya se ha mencionado, los operadores genéticos del GA son *crossover* y mutación. El método de selección, torneo. El tamaño inicial de la población escogido fue de 25, como en el estudio de Echegaray y Barrios (2015).

4. Resultados

Una vez aplicado el procesamiento de datos explicado al inicio de la sección anterior, se obtuvieron las variables de entrada sobre las cuales se llevó a cabo el análisis (ver tabla 2).

Tabla 2
Variables de entrada luego del procesamiento de datos

Id	Variables
X00	Age
X01	DistanceFromHome
X02	Education
X03	EnvironmentSatisfaction
X04	Gender
X05	JobInvolvement
X06	JobLevel
X07	JobSatisfaction
X08	MonthlyIncome
X09	NumCompaniesWorked
X10	OverTime
X11	PercentSalaryHike
X12	PerformanceRating
X13	RelationshipSatisfaction
X14	StockOptionLevel
X15	TotalWorkingYears
X16	TrainingTimesLastYear
X17	WorkLifeBalance
X18	YearsAtCompany
X19	YearsInCurrentRole
X20	YearsSinceLastPromotion
X21	YearsWithCurrManager
X22	Business Travel = Non-Travel
X23	Business Travel = Travel_Frequently
X24	Business Travel = Travel_Rarely
X25	Department = Human Resources
X26	Department = Research & Development
X27	Department = Sales
X28	EducationField = Human Resources
X29	EducationField = Life Sciences
X30	EducationField = Marketing
X31	EducationField= Medical
X32	EducacionField = Other
X33	EducationField = Technical Degree
X34	JobRole = Healthcare Representative

(continúa)

(continuación)

X35	JobRole = Resources
X36	JobRole = Laboratory Technician
X37	JobRole = Manager
X38	JobRole = Manufacturing Director
X39	JobRole = Research Director
X40	JobRole = Research Scientist
X41	JobRole = Sales Executive
X42	JobRole = Sales Representative
X43	MaritalStatus = Divorced
X44	MaritalStatus = Married
X45	MaritalStatus = Single

Elaboración propia. *Dataset* obtenido de <https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset>

4.1 Selección de variables

Para reducir la variabilidad de las capas ocultas de la RNA, se propuso utilizar solo una capa oculta, con cantidad de nodos igual a:

$$\text{Nodos} = 1/2 (1 + \sum_{i=1}^n v_i)$$

donde, v es el vector que representa las variables a utilizar y n , la longitud del mismo vector.

Entonces, la cantidad de nodos en la capa oculta es igual a la mitad de la suma de la cantidad de nodos en la capa de entrada más la cantidad de nodos en la capa de salida (1). Con esto, se logra reducir la influencia de las capas ocultas en el valor de la exactitud.

El programa se ejecutó 6 veces, lográndose una precisión máxima de 87,2 % (ver tabla 3). Se intentó obtener mayor precisión creando las RNA que consideren las variables utilizadas más frecuentemente en las ejecuciones E1 hasta E6, manteniendo los nodos y capas ocultas constantes.

Tabla 3
Precisión obtenida por AG en 6 ejecuciones (E1...E6)

	E1 (Gen 567)	E2 (Gen 407)	E3 (Gen 442)	E4 (Gen 446)	E5 (Gen 342)	E6 (Gen 376)
Precisión	0,866	0,870	0,872	0,865	0,865	0,864

Elaboración propia

Luego se generaron 3 nuevas RNA, la primera consideró las variables que tuvieron una frecuencia (F) mayor o igual a 5; la segunda, mayor o igual a 4; la tercera, mayor o igual a 3 (ver tabla 4).

Tabla 4
Comparación de RNA propuestas en base a la frecuencia de las variables

	E(F>=5)	E(F>=4)	E(F>=3)
Promedio de precisión	0,87458	0,86836	0,868213
Desviación estándar	0,005138	0,00565	0,004236

Elaboración propia

Después de haber corrido 10 veces el módulo de RNA del programa, se obtuvo una nueva relación de variables, luego de comparar las redes neuronales propuestas. Las variables de entrada con una frecuencia de aparición mayor o igual a 5 (F>=5) tuvieron la mayor exactitud (ver tablas 5 y 6).

Tabla 5
Variables que inicialmente aparecieron en 5 o más ejecuciones

Rótulo	Variable
X00	Edad
X01	Distancia (en km) desde la casa al trabajo
X03	Satisfacción con el ambiente laboral
X04	Género
X05	Participación laboral
X07	Satisfacción laboral
X10	Sobretiempo
X18	Cantidad de años en la compañía

(continúa)

(continuación)

X20	Cantidad de años desde último ascenso
X23	Trabajo requiere que viaje frecuentemente
X30	Área de marketing
X40	Rol de investigador

Elaboración propia

Tabla 6
Agrupación de variables recopiladas y seleccionadas

	Variables recopiladas	Variables seleccionadas
V01	Salario	-
V02	Ambiente laboral	X03
V03	(Falta) línea de carrera	X20
V04	Satisfacción laboral	X07
V05	Beneficios	X23
V06	Compromiso organizacional	X05, X18
V07	Relación con superiores y compañeros	X03, X05, X07
V08	Estrés laboral	X10
V09	Estado civil	-
V10	Género	X04

Elaboración propia

Adicionalmente, se obtuvieron variables que fueron descartadas o no se encontraron, explícitamente, en el estado del arte (ej., beneficios no monetarios [X10] de la compañía y frecuencia de viaje por trabajo [X23]), pero que fueron seleccionadas por la herramienta:

X00: edad

X01: distancia (en km) desde la casa al trabajo

X23: trabajo requiere que viaje frecuentemente

X30: área de *marketing*

X40: rol de investigador

Por otro lado, las variables recopiladas que el modelo no seleccionó fueron:

V01: salario

V09: estado civil

En la investigación cualitativa de Burnes (2006) se concluyó que en el sector de las tecnologías de la información no influía tanto la edad, ya que había más similitudes que diferencias entre los empleados de la generación X y de la generación no-X. Sin embargo, los datos utilizados son multiárea; según la herramienta, la edad sí es una variable significativa.

Las variables “área de marketing” (X30) y “rol de investigador” (X40) son variables que tampoco se encontraron en el estado del arte. Parecen ser más específicas al caso dado, por lo que su interpretación sería: la deserción laboral también varía por área y rol del empleado.

4.2 Selección de capas y nodos ocultos

Debido a que se encontraron las variables que maximizan la exactitud del modelo, se constató que ellas se mantuvieron constantes (ver tabla 7).

Tabla 7

Precisión lograda por AG en 6 ejecuciones (E1...E6) manteniendo las variables constantes

	E1	E2	E3	E4	E5	E6
Capas y nodos ocultos	(6)	(6, 3)	(12)	(7)	(5, 3, 2)	(5, 3, 1)
Máxima precisión	0,8734	0,8800	0,8746	0,8780	0,8680	0,0865

Elaboración propia

La tabla 7 indica las mejores configuraciones obtenidas por el AG. La interpretación de la nomenclatura de capas y nodos es la misma que se explicó previamente.

Luego se obtuvo la precisión promedio de las RNA, utilizando las configuraciones antes señaladas y ejecutando el módulo de RNA 10 veces (igual al procedimiento empleado previamente).

Tabla 8
Comparación de RNA en base a sus capas y nodos ocultos

Ejecución	C1 (6)	C2 (6, 3)	C3 (12)	C4 (7)	C5 (5, 3, 2)	C6 (5, 3, 1)
1	0,8826	0,8724	0,8742	0,8749	0,8708	0,8461
2	0,8707	0,8699	0,8691	0,8733	0,8571	0,8648
3	0,8724	0,8597	0,8750	0,8775	0,8648	0,8546
4	0,8775	0,8657	0,8681	0,8751	0,8682	0,8486
5	0,8690	0,8734	0,8572	0,8776	0,8657	0,8427
6	0,8758	0,8640	0,8699	0,8750	0,8520	0,8537
7	0,8792	0,8623	0,8759	0,8693	0,8571	0,8622
8	0,8699	0,8716	0,8665	0,8699	0,8682	0,8470
9	0,8844	0,8682	0,8818	0,8892	0,8562	0,8529
10	0,8699	0,8656	0,8674	0,8750	0,8563	0,8529
Promedio de precisión	0,8751	0,8672	0,8705	0,8756	0,8616	0,8525

Elaboración propia

De la tabla 8 se puede ver que la mejor configuración fue la C4, que cuenta con 7 nodos en su única capa oculta, ya que obtuvo la precisión promedio mayor y una precisión máxima de 88,92 %.

Es cierto que no se percibe mucha mejora entre esta configuración (C4) y la hallada anteriormente (C1). Esto se debe a que la propuesta señalada previamente de utilizar una capa oculta y nodos igual a la mitad de variables seleccionadas, coincidió con ser la segunda mejor solución posible. Si se hubiera escogido una configuración de tres capas ocultas para esa sección, se hubiera tenido una precisión de 85 % como en C6. Debido a que la RNA tuvo doce variables de entrada, no fue necesario tener tantas capas ocultas como inicialmente se pensó. Tal vez, si en otro escenario se tuvieran más variables (como las 46 iniciales), sería más factible utilizar configuraciones de tres capas ocultas como C5 y C6.

5. Conclusiones

Se logró identificar y contrastar las variables recopiladas de estudios empíricos pasados con las variables seleccionadas por la herramienta propuesta. La mayoría de variables revisadas también fueron consideradas y son coherentes con las seleccionadas por el modelo, lo cual da más veracidad al mismo. Además, se consideraron nuevas variables que no se habían visto en la literatura revisada (ej., distancia de la casa al trabajo).

Adicionalmente, se logró establecer restricciones razonables para las potenciales arquitecturas de las redes neuronales que mejoraban la precisión (métrica seleccionada). Entonces, se logró crear una herramienta útil y de calidad que permite identificar la deserción de un empleado hasta un 88,92 % de precisión.

Referencias

- Anand, V. V., Saravanasudhan, R., y Vijesh, R. (2012). Employee attrition: A pragmatic study with reference to BPO industry. *Management*, 2(9), pp. 769-775. Recuperado de <https://ieeexplore.ieee.org/document/6215943/>
- Ashok, J. (2003). A study on linkage between organizational practices and employee attrition. *International Journal of Knowledge Management & Practice*, 1(1), pp. 30-34.
- Burnes, P. T. (2006). Voluntary employee turnover: Why IT professionals leave. *IT Professional*, 8(3), pp. 46-48. doi:10.1109/MITP.2006.78
- Echegaray Calderón, O. A., y Barrios Aranibar, D. (2015). Optimal selection of factors using Genetic Algorithms and Neural Networks for the prediction of students' academic performance. *Latin America Congress on Computational Intelligence (LA-CCI)*, pp. 1-6. doi:10.1109/LA-CCI.2015.7435976
- Ettaouil M., y Ghanou, Y. (2009). Neural architectures optimization and genetic algorithms. *WSEAS Transactions on Computers*, 8(3), pp. 526-537. Recuperado de <http://www.wseas.us/e-library/transactions/computers/2009/31-699.pdf>
- Fidalgo F., y Gouveia, L. B. (2012). Turnover impact into organizational knowledge management. The case of employee turnover in Portuguese Real Estate. *6th Iberian Conference on Information Systems and Technologies (CISTI 2011)*, pp. 1-6.
- Firth, L., Mellor, D. J., Moore, K. A., y Loquet, C. (2004). How can managers reduce employee intention to quit? *Journal of Managerial Psychology*, 19(2), pp. 170-187. doi:10.1108/02683940410526127
- Fiszelew, A., Britos, P., Ochoa, A., Merlino, H., Fernandez, E., y García-Martínez, R. (2007) Finding optimal neural network architecture using genetic algorithms. *Advances in Computer Science and Engineering Research in Computer Science*, 27, pp. 15-24. Recuperado de <http://www.iidia.com.ar/rgm/articulos/RCSJ-27-15-24.pdf>
- Flores, C. (22 de octubre de 2014). Perú tiene alta tasa de rotación de personal. *Diario Correo*. Recuperado de <https://diariocorreo.pe/economia/peru-tiene-alta-tasa-de-rotacion-de-personal-9770/>

- Idrissi, M. A. J., Ramchoun, H., Ghanou, Y., y Ettaouil, M. (2016). Genetic algorithm for neural network architecture optimization. *IEEE 3rd International Conference on Logistics Operations Management (GOL)*, pp. 1-4. doi:10.1109/GOL.2016.7731699
- Islam, B., Baharudin, Z., Raza, M. Q., y Nallagownden, P. (2014). Optimization of neural network architecture using genetic algorithm for load forecasting. *IEEE 5th International Conference on Intelligent and Advanced Systems (ICIAS)*, pp. 1-6. doi:10.1109/ICIAS.2014.6869528
- Mal clima laboral genera pérdidas millonarias a las empresas (3 de mayo de 2016). *La República*. Recuperado de <https://larepublica.pe/sociedad/936033-mal-clima-laboral-genera-perdidas-millonarias-a-las-empresas>
- Naciones Unidas/CEPAL (2016). Agenda 2030 y los Objetivos de Desarrollo Sostenible: una oportunidad para América Latina y el Caribe. doi:10.1017/CBO9781107415324.004
- Pawar, A., y Chakravarthy, V. (2014). Factors influencing employee turnover in fusion healthcare organization. *International Journal of Management Research and Reviews*, 4(9), pp- 834-846.
- Perú es el tercer país con la mayor rotación de personal en Latam (24 de octubre de 2017). *El Portal del Capital Humano*. Recuperado de <http://www.infocapitalhumano.pe/recursos-humanos/noticias-y-movidas/peru-es-el-tercer-pais-con-la-mayor-rotacion-de-personal-en-latam/>
- Rotación laboral de ejecutivos en altos mandos alcanza el 25 %. (7 de julio de 2014). *El Comercio*. Recuperado de <https://elcomercio.pe/economia/ejecutivos/rotacion-laboral-ejecutivos-altos-mandos-alcanza-25-346784>
- Sandoval, M. (27 de agosto de 2014). El Perú tiene el más alto índice de rotación laboral en Latinoamérica. *El Portal del Capital Humano*. Recuperado de <http://www.infocapitalhumano.pe/recursos-humanos/noticias-y-movidas/el-peru-tiene-el-mas-alto-indice-de-rotacion-laboral-en-latinoamerica/>
- Wei, X., y Hai, L. (2007). Chinese employee's turnover intentions in relation to organizational identification, work values, job satisfactions in service industry. *IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 4435-4438. doi:10.1109/WICOM.2007.1095

PREDICCIÓN DE POSTULANTES QUE COMETERÁN FRAUDE INTERNO EN UNA COMPAÑÍA CON ALGORITMOS DE APRENDIZAJE SUPERVISADO*

Sergio Espinoza-Montalvo
20142424@aloe.ulima.edu.pe
Universidad de Lima. Lima, Perú

Resumen

El fraude interno es un gran problema para las empresas, ocasionando pérdidas monetarias importantes. Diversas investigaciones han propuesto mejoras al proceso de selección de personal utilizando minería de datos. El presente trabajo propone utilizar la información histórica de postulantes a una empresa para predecir si cometerán fraude durante su estadía. Existen modelos con un nivel de precisión alto, pero que tienen un error de clasificación mayor para encontrar los casos de fraude. Después de diversas experimentaciones, se identifican alrededor de 7 características de este universo que aportan más al modelo. Algunas de estas variables coinciden con variables mencionadas en la literatura encontrada sobre trastornos antisociales. El algoritmo con mejores resultados es una red neuronal convolucional con 80 % de precisión. Se concluye que hay valor en la información de postulantes para determinar si cometerán fraude interno durante su estadía en la empresa.

Palabras clave: aprendizaje supervisado, predicción de fraude, trastorno antisocial, fraude interno

Abstract

Prediction of applicants who will commit internal fraud in a company using supervised learning algorithms

Internal fraud is a big problem for companies since it causes significant monetary losses. Several research studies have proposed to improve the personnel selection process using data mining. The present work suggests to use applicants' historical information in order to predict if they will commit fraud during their working period in a company. There are models with high precision level but with a higher error rate to find fraud. After several experimentations, around seven variables which contribute more to the model were found. Some of these variables match those mentioned in studies about antisocial personality disorder. The algorithm with best results was a convolutional neural network with 80% accuracy rate. It is concluded that applicants' information is important to establish if they will commit internal fraud during their working period in a company.

Keywords: supervised learning, fraud prediction, antisocial personality disorder, internal fraud

* **Agradecimientos.** Se agradece el apoyo de la licenciada en psicología Doris García para analizar las pruebas de Wartegg de los postulantes. De igual manera, la asesoría del profesor José Antonio Taquia en técnicas y herramientas para minería de datos. Además de las recomendaciones de los profesores Juan Gutiérrez, Daniel Cárdenas y Rosario Guzmán para la realización del trabajo.

1. Introducción

El fraude interno es uno de los problemas que está presente en todas las organizaciones y ocasiona pérdidas económicas importantes a las mismas (Kroll, 2012), así como baja reputación, fugas de información, robos de equipos, entre otros (Smith, 2005). En el Perú, aproximadamente, el 80 % de los casos de fraude viene del interior de la compañía y el 22 % de los casos, en 2013, ocasionaron más de 100 mil dólares en pérdidas para las grandes empresas (EY, s. f.).

En vista de las exigentes demandas crecientes de calidad, costo y tiempos en las compañías, la minería de datos es una necesidad para los departamentos de recursos humanos, por el manejo de la información sobre los empleados que se puede explotar, incluyendo los *curriculum vitae*, información de recursos humanos, evaluaciones, entre otros (Horesh, 2016). Reconocer el patrón de comportamiento de los trabajadores en el mercado es uno de los recursos, actualmente, más importantes para las empresas (Rashid, 2016).

En el presente artículo se analizan casos de fraude interno en una empresa de suministros de cómputo en el Perú. Se tuvo acceso a la información histórica de los postulantes para predecir cuales cometerían fraude interno con base en una clasificación con algoritmos de aprendizaje supervisado. El documento consta de las siguientes partes: la sección 2 presentará el contexto relacionado al estado del arte; la 3, la metodología de la investigación; la 4, los resultados obtenidos. Finalmente, las 5 y 6 mostrarán las conclusiones y las recomendaciones para investigaciones futuras.

2. Contexto

Diversas investigaciones han tratado el problema de selección de personal en las empresas utilizando minería de datos. Varshney (2014) utilizó la información de recursos humanos para predecir la aptitud de los postulantes para su cargo, usando SVM (*support vector machine*) y regresión lineal. Una investigación del 2016 utilizó múltiples fuentes de datos sobre los empleados de una organización para poder predecir la aptitud de estos para su empleo en tiempo real; la técnica utilizada fue *ordinal regression clustering*, la cual consiste en aplicar regresión para agrupar datos (Horesh, 2016). En el estudio de Jantan (2011), donde se predijo la aptitud de un empleado para su cargo, el C4.5/J48 demostró ser el más efectivo, con 95,14 % de precisión. En la investigación de Rashid (2016) para predecir el comportamiento de empleados, los árboles de decisión tuvieron un 97,61 % de precisión, mientras que la red neuronal convolucional obtuvo mejores resultados, 98,12 % de precisión.

En investigaciones enfocadas a detectar fraude en transacciones, tales como Bhattacharyya (2011), se tiende a usar el algoritmo *random forest*, donde este mismo tuvo el mejor resultado con 86 % de precisión. Existen diversas técnicas para optimizar los modelos de minería de datos. El experimento de Rashid demostró una mejora en promedio de 1,75 %

de precisión, realizando selección de características con técnicas de *information gain*, *gain ratio*, *oneR* y *fuzzy RST* (Rashid, 2016). El estudio de Horesh (2016) implementó la técnica de terminación de matriz mediante optimización convexa para obtener los datos faltantes (alrededor del 80 %) con buenos resultados. Para optimizar los hiperparámetros, en la investigación de Chang se ha encontrado el uso de métodos Taguchi para predecir el abandono laboral (Chang, 2009).

2.1 Trastornos antisociales

Hay una clara relación entre individuos que cometen actos criminales, tales como fraude interno, e individuos con un trastorno antisocial (Aquino, 2003). De acuerdo al DSM-5 (Diagnostic and Statistical Manual of Mental Disorders, por sus siglas en inglés) (Manual diagnóstico y estadístico de los trastornos mentales), una persona con trastorno de personalidad antisocial (TPA) se caracteriza por incumplir las reglas sociales, mentir seguido, no tener consideración por la seguridad ajena, así como tener una conducta irresponsable y un comportamiento impulsivo (APA, 2013). Algunas prognosis para personas con TPA son suicidio, abuso de sustancias, menor probabilidad de estar casado, violencia, crimen, desempleo y estar sin hogar (Le Corff, 2014; Yavuz, 2016).

2.2 Examen de Wartegg

La prueba de Wartegg (ver figura 1) es un examen proyectivo psicológico que se usa para la selección de personal. Este consiste en ocho campos, donde cada uno tiene un trazo impreso, conocido como estímulo. Los postulantes tienen la libertad de elegir por cual campo empezar y qué ilustrar en cada uno. Hay cuadros en donde se puede escribir el orden en el cual se efectuó la prueba; y, finalmente, se hacen cuatro preguntas sobre los dibujos. Hay un rango de dibujos esperados para cada campo, y basándose en esto, junto con otros factores, se realiza la evaluación.

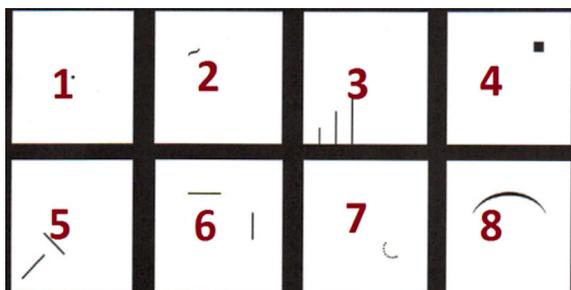


Figura 1. Numeración de los cuadros con los estímulos de la prueba de Wartegg, en ella los postulantes tienen que dibujar en los cuadros de forma libre y responder preguntas.

Elaboración propia

3. Metodología

Para la muestra de la experimentación se utilizaron 63 registros de empleados de una empresa mayorista de suministros de cómputo, los cuales trabajaron entre 2013 y 2016. Dichos registros incluían la hoja de vida del postulante, la hoja de información y pruebas psicológicas realizadas por los mismos. La información fue proporcionada por la empresa de manera física. Adicionalmente, se indicó qué empleados cometieron fraude durante su estadía en la compañía.

Para seleccionar las características de la hoja de vida y de la hoja de información de los postulantes, se usó el conocimiento previo en variables relacionadas con trastornos antisociales. Algunas de estas variables fueron el sexo, los meses promedio entre trabajos, el número de trabajos pasados, el estado civil, entre otras. Para el tratamiento de las pruebas psicológicas, se contó con el apoyo de una psicóloga experta en el examen de Wartegg. La experta dio pautas y entregó material sobre la prueba, el cual se utilizó para crear variables sobre las pruebas psicológicas. Para los 8 campos se colocó una evaluación del 0 al 2, dependiendo de dos criterios de evaluación: uso del estímulo y dibujo del campo. Para los campos 1, 2, 7 y 8 el dibujo debía ser orgánico; para los campos 3, 4, 5 y 6, inorgánico. El uso del estímulo evaluó si el postulante realizaba un dibujo que lo incluyera de forma adecuada. Los criterios para considerar que no se utilizó este de forma adecuada fueron: el dibujo no encapsula o incluye el estímulo y si el dibujo se realiza encima del estímulo. Fallar los criterios mencionados resultaba en una evaluación mejor del campo. Ponderaciones de evaluaciones de los campos se utilizaron para crear variables tales como empatía, objetivos, órdenes, entre otras. Otras variables incorporaban el orden de realización de los campos, complejidad de dibujos y complejidad de descripciones.

Para la experimentación se probó la hipótesis: existen variables relevantes en la información de empleados que permiten predecir fraude interno. Para el procesamiento de la información se utilizaron algoritmos de aprendizaje supervisado encontrados en la literatura, tales como C.45 (Jantan, 2011), redes neuronales (Rashid, 2016) y *random forest* (Bhattacharyya, 2011). Se compararon los resultados obtenidos entre los algoritmos para determinar el más apto para el problema en cuestión. Para optimizar el modelo se emplearon métodos de selección de características (análisis de entropía, análisis de componentes).

4. Resultados

Se realizó el procesamiento de la información con tres algoritmos de aprendizaje supervisado: C.45 (ver figura 2), BRNN (ver tabla 1) y *random forest* (ver tabla 2), los cuales fueron resumidos (ver tabla 3).

En el árbol de decisión se puede apreciar que las dos primeras divisiones se realizaron utilizando las evaluaciones del campo 7 (sensibilidad) y campo 2 (afectividad y emotividad) del examen de Wartegg. De acuerdo a lo indicado por la evaluadora experta, estos campos están relacionados con la empatía. La información del modelo indica que una alta evaluación en estos campos de los postulantes reduce significativamente la probabilidad de que este haya cometido fraude interno. Esto se traduciría a que un postulante con alta empatía tendría poco riesgo de cometer fraude interno, lo que estaría alineado con la información encontrada sobre el TPA.

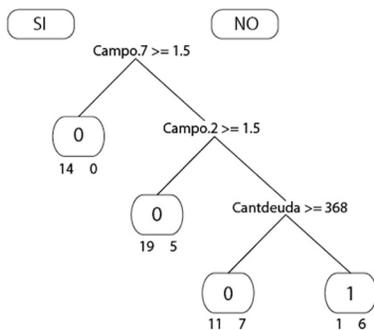


Figura 2. Árbol de decisión obtenido con las características del modelo

Elaboración propia

La red neuronal con regularización bayesiana se realizó con validación cruzada (10 divisiones y 10 repeticiones). El modelo óptimo fue obtenido con 6 neuronas, el cual tuvo un RMSE de 0,4506 y un R cuadrado de 0,1835. El valor de R cuadrado bajo simboliza que el modelo tiene mucha varianza, lo que mostraría que probablemente se necesitan usar características más óptimas (ver tabla 1).

Tabla 1

BRNN obtenida con las características del modelo

Neuronas	RMSE	R2	MAE
1	0,46675	0,1656	0,40528
2	0,45795	0,1822	0,39931
3	0,46050	0,1912	0,40481
4	0,45398	0,1853	0,39732
5	0,45359	0,1860	0,39640

(continúa)

(continuación)

6	0,45061	0,1835	0,39555
7	0,45246	0,1846	0,39647
8	0,45163	0,1856	0,39748
9	0,45213	0,1846	0,39800
10	0,45135	0,1853	0,39702

Elaboración propia

El *random forest* fue generado con 1000 árboles y obtuvo un error OOB (error fuera de caja) de 0,2381, el cual equivale a una precisión de 76,2 %, aproximadamente. Al analizar la matriz de confusión nos podemos percatar que el error de clase, para predecir a las personas que cometieron fraude, es mucho mayor que para predecir a las personas que no cometieron fraude. Esto es preocupante, puesto que simboliza que el modelo está más predispuesto a dar más verdaderos negativos que falsos positivos, lo que no es algo deseado para este tipo de problema.

Tabla 2

Matriz de confusión del *random forest* generado con dos variables

	0	1	Error de clase
0	39	6	13,33 %
1	9	9	50,00 %

Elaboración propia

Comparando los resultados de los tres algoritmos, vemos que, en el contexto del problema, el modelo más efectivo sería el obtenido con *random forest*. Esto se debe a que cuenta con el porcentaje de precisión más alto para encontrar los casos de fraude. El mejor resultado se obtuvo utilizando solo dos variables: trabajos y desempleo (ver tabla 3).

Tabla 3

Resultados del procesamiento de la información con los algoritmos de aprendizaje supervisado

Método	VARIABLES que aportan	Precisión	Precisión (solo para fraudulentos)
RF	Trabajos, desempleo	76,19 %	50,00 %
C.45	Campo.7, Campo.2, Cantdeuda	79,36 %	33,33 %
BRNN	Por determinar	RMSE (0,4647)	Por determinar

Elaboración propia

Para comparar la efectividad de los algoritmos, se realizaron pruebas utilizando diferente número de características (ver tabla 4). Para definir las características a usar, se utilizó el método de análisis de componentes principales (ACP). De este modo, se procesó la información utilizando solo el número indicado de características que aportaban más variación al modelo. La división empleada fue de 52 registros de entrenamiento y 11 de prueba. Se utilizó el mismo valor de semilla para todos los escenarios. Se logró identificar el número de mejores características usadas en la iteración (columna n.º); precisión F o prueba F indica que el porcentaje de precisión es solo para los postulantes que cometieron fraude.

Se puede ver que tanto los algoritmos de C.45 y *random forest* no tuvieron mucha efectividad para poder predecir correctamente la información de prueba, especialmente aquella enfocada en encontrar los casos de fraude. El algoritmo que tuvo los mejores resultados fue la BRNN utilizando 5 registros, donde se obtuvo una precisión de 91 %, aproximadamente, y 50% para los casos de fraude. La selección de características mejoró los resultados hasta cierto punto para cada modelo, donde se puede ver que reduciendo a cierto número de variables los resultados comienzan a empeorar.

Tabla 4
Resultados con diferentes características en los tres algoritmos de aprendizaje supervisado

n.º	C45				Random forest				BRNN			
	Precisión	PrecisiónF	Test	TestF	Precisión	PrecisiónF	Test	TestF	RMSE	Test	TestF	
50	67,50 %	0,00 %	81,82 %	0,00 %	59,62 %	0,00 %	63,64 %	0,00 %	0,5032	72,73 %	50,00 %	
40	66,91 %	0,00 %	81,82 %	0,00 %	59,62 %	0,00 %	63,64 %	0,00 %	0,4882	72,73 %	50,00 %	
30	60,72 %	43,75 %	63,54 %	0,00 %	61,54 %	0,00 %	63,64 %	0,00 %	0,4921	90,91 %	50,00 %	
20	65,03 %	56,25 %	54,55 %	0,00 %	63,46 %	18,75 %	54,55 %	0,00 %	0,4887	90,91 %	50,00 %	
15	62,02 %	50,00 %	45,45 %	0,00 %	69,23 %	18,75 %	54,55 %	0,00 %	0,4919	90,91 %	50,00 %	
10	64,05 %	43,75 %	45,45 %	0,00 %	69,23 %	25,00 %	45,45 %	0,00 %	0,4758	81,82 %	0,00 %	
8	68,25 %	0,00 %	81,82 %	0,00 %	65,38 %	0,00 %	63,64 %	0,00 %	0,4771	81,82 %	0,00 %	
5	66,92 %	0,00 %	81,82 %	0,00 %	51,92 %	0,00 %	63,64 %	0,00 %	0,4689	90,91 %	50,00 %	
3	67,90 %	0,00 %	81,82 %	0,00 %	65,38 %	0,00 %	81,82 %	0,00 %	0,4671	81,82 %	0,00 %	
2	69,63 %	0,00 %	81,82 %	0,00 %	65,38 %	0,00 %	81,82 %	0,00 %	0,4716	81,82 %	0,00 %	
2	64,60 %	0,00 %	81,82 %	0,00 %	59,62 %	0,00 %	63,64 %	0,00 %	0,4636	81,82 %	0,00 %	
2	67,90 %	0,00 %	81,82 %	0,00 %	59,62 %	0,00 %	81,82 %	0,00 %	0,4654	81,82 %	0,00 %	

Elaboración propia

Del análisis (valoración de *random forest*, C.45, PCA y análisis de entropía) se puede determinar que las siguientes fueron las variables más aportantes del modelo.

- Wartegg4 (cuadro elegido en respuesta a la última pregunta del examen de Wartegg, "¿cuál dibujo te pareció más difícil?")
- Desempleo (promedio de tiempo desempleado entre trabajos del postulante. De acuerdo a la teoría de trastornos antisociales [ver sección 2.1], las personas con TPA tienden a estar más tiempo desempleados)
- Trabajos (número de trabajos pasados del postulante. De acuerdo a la teoría de trastornos antisociales [ver sección 2.1], las personas con TPA tienden a rotar más entre trabajos)
- Campo.7 (evaluación del campo 7 del examen de Wartegg utilizando las fórmulas facilitadas por la experta psicóloga, penúltimo campo del examen que está cercanamente relacionado con la empatía)
- Convivientes (número de personas con las que el postulante convive, obtenido de la hoja de información donde se le hacen diversas preguntas al postulante. De acuerdo a la teoría de trastornos antisociales [ver sección 2.1], las personas con TPA tienden más a vivir solos)
- Cantdeuda (cantidad de deuda que tiene el postulante. De acuerdo a la teoría de trastornos antisociales [ver sección 2.1], las personas con TPA tienden a ser más impulsivas y endeudarse más)
- Empatía2 (ponderación de evaluación de los campos de empatía del examen [campos 2 y 8 (unión, armonía y compromiso)])

Todas estas características estarían relacionadas con un trastorno antisocial. Se evaluaron estas variables con un análisis de componentes (ver figura 3). Se puede apreciar que campo.7 y fraude son inversamente proporcionales, lo que concuerda con los modelos hallados y lo esperado; donde a mayor valor de campo.7 (representando una mejor evaluación en el campo para el postulante), menor valor de fraude (equivaliendo fraude = 0 a un postulante que no cometió fraude).

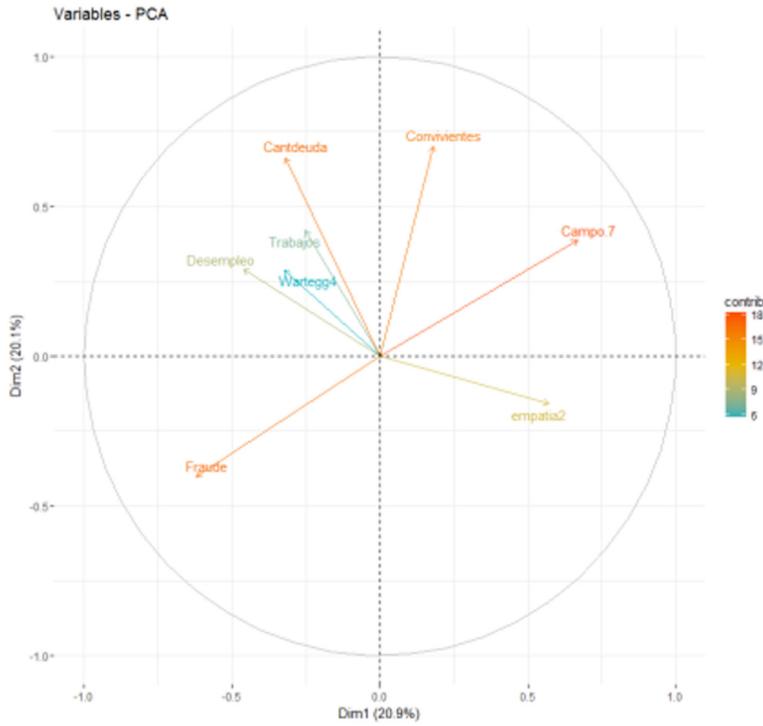


Figura 3. Análisis de componentes principales (ACP) con las variables más relevantes

Elaboración propia

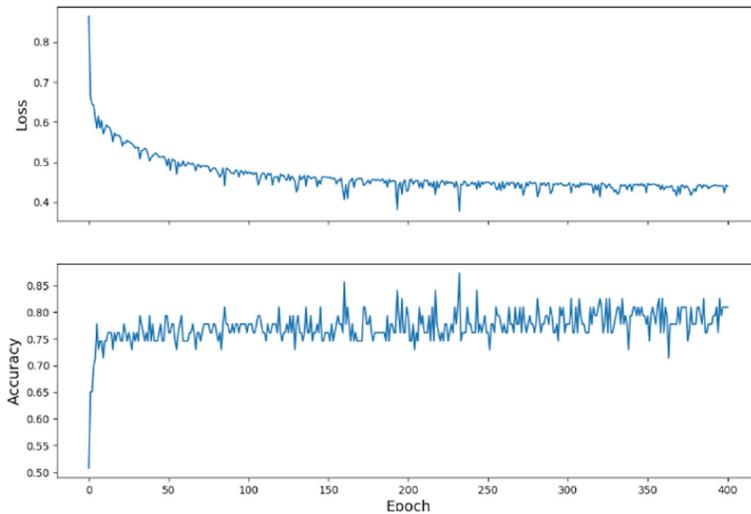


Figura 4. Ratio de aprendizaje de red neuronal entrenada con las mejores características identificadas

Elaboración propia

Finalmente, se entrenó una red neuronal convolucional usando las mejores características identificadas y se obtuvo un porcentaje de precisión de 80 %. El porcentaje de precisión de la red en cada iteración mejoró de forma sistemática (ver figura 4).

6. Conclusiones

Tras el análisis se puede concluir que varias de las características encontradas en los registros de los postulantes están relacionadas con el resultado de que este vaya o no a cometer fraude durante su estadía en la empresa. Sin embargo, la mayoría de modelos tiene más dificultades identificando a los que van a cometer fraude, que a quienes tienen más probabilidad de no hacerlo. Esto explicaría el porqué del error de clase para predecir fraude es tan alto, a pesar de que los valores de precisión global para los modelos estaban entre 70 y 80 %.

Por lo tanto, se concluye que sí se puede predecir si un postulante cometerá fraude interno o no con cierta efectividad. Los resultados parciales muestran que los algoritmos más efectivos para este tipo de modelos serían las redes neuronales. Además, se puede ver que los mejores predictores para este problema están relacionados con trastornos antisociales de acuerdo a lo encontrado en la literatura. Igualmente, con el apoyo de una experta en psicología, se pudieron obtener variables relevantes para el modelo del examen de Wartegg.

7. Trabajos futuros

Se deberían realizar mejoras para reducir el error de clase para detectar fraude, puesto que, como se encontró en la literatura, para este tipo de problemas los falsos positivos son preferibles que los verdaderos negativos.

Uno de los principales problemas es la selección de características. Por un lado, esto se podría solucionar seleccionando, de forma más eficiente, las variables a utilizar en el modelo. Se podrían usar algoritmos genéticos que definan como ajuste del error de clase para predecir el fraude, de modo que se buscaría, específicamente, el mejor modelo que prediga mejor los casos de fraude. Para optimizar los hiperparámetros, se podrían utilizar métodos Taguchi. Por otro lado, no se debería descartar la ayuda de expertos para procesar de manera más efectiva los exámenes psicológicos que se realizan a los postulantes; puesto que se ha evidenciado que de este análisis se pueden obtener características que aportan al modelo. Se recomienda utilizar variables relacionadas a trastornos antisociales según la teoría de TPA, puesto que se puede ver que estas están correlacionadas con la posibilidad de que una persona vaya a cometer fraude o no.

Referencias

- American Psychiatric Association (APA). (2013). *Diagnostic and Statistical Manual of Mental Disorders*. (5th ed.). American Psychiatric Publishing.
- Aquino K., y Douglas, S. (2003). Identity threat and antisocial behavior in organizations: The moderating effects of individual differences, aggressive modeling, and hierarchical status. *Organizational Behavior and Human Decision Processes*, 90(1), pp. 195-208; doi: 10.1016/s0749-5978(02)00517-4
- Bhattacharyya, S., Jha, S., Tharakunnel, K., y Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3), pp. 602-613. doi:10.1016/j.dss.2010.08.008
- Chang, H. Y. (2009). Employee turnover: a novel prediction solution with effective feature selection. *WSEAS International Conference on Computer Engineering and Applications*, 3(6), pp. 417-426.
- EY. (s. f.). Construyendo un ambiente ético. Estudio sobre el riesgo de fraude en el Perú. Recuperado de <http://www.ey.com/pe/es/services/assurance/fraud-investigation---dispute-services/construyendo-un-ambiente-etico-estudio-sobre-el-riesgo-de-fraude-en-el-peru>
- Horesh, R., Varshney, K. R., y Yi, J. (2016). Information retrieval, fusion, completion, and clustering for employee expertise estimation. *IEEE International Conference on Big Data*.
- Jantan, H., Hamdan, A. R., y Othman, A. (2011). Towards applying data mining techniques for talent management. *2009 International Conference on Computer Engineering and Applications, IPCSIT*, 2, pp. 476-481. Singapore: IACSIT.
- Kroll, K. (2012). Keeping the company safe: Preventing and detecting fraud. *Financial Executive*, 28(7), pp. 20-23.
- Le Corff, Y., y Toupin, J. (2014). Overt versus covert conduct disorder symptoms and the prospective prediction of antisocial personality disorders. *Journal of Personality Disorders*, 28(6), pp. 864-872. doi:10.1521/pedi_2012_26_074
- Rashid, T. A., y Asia, L. J. (2016). Improvement on predicting employee behaviour through intelligent techniques. *IET Networks*, 5(5), pp. 136-142. doi:10.1049/iet-net.2015.0106
- Smith, A. D. (2005). *Accountability in EDI systems to prevent employee fraud. Information Systems Management*, 22(2), pp. 30-38.

- Varshney, K. R., Chenthamarakshan, V., Fancher, S. W., Wang, J., Fang, D., y Mojsilović, A. (2014). Predicting employee expertise for talent management in the Enterprise. *KDD 14 Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1729-1738. New York: ACM. doi:10.1145/2623330.2623337
- Yavuz, K., Şahin, O., Ulusoy, S., İpek, O., y Kurt, E. (2016). Experiential avoidance, empathy, and anger-related attitudes in antisocial personality disorder. *Turkish Journal of Medical Sciences*, 46(6), pp. 1792-1800. doi:10.3906/sag-1601-80

SISTEMA DE RECONOCIMIENTO DE GESTOS FACIALES CAPTADOS A TRAVÉS DE CÁMARAS PARA ANALIZAR EL NIVEL DE SATISFACCIÓN DE CLIENTES EN RESTAURANTES

Edwin Lara-Lévano

20142424@aloe.ulima.edu.pe
Universidad de Lima. Lima, Perú

Resumen

La presente investigación tiene como principal objetivo el desarrollo de un sistema que reconozca la satisfacción o insatisfacción de un cliente en un restaurante con base en los gestos que este mismo realiza al momento de recibir el servicio brindado por el establecimiento. La implementación del sistema cuenta con una serie de etapas comunes al desarrollo de un proyecto de computación visual, las cuales inician con un preprocesamiento de los datos a ser empleados para el entrenamiento del clasificador que se utilizará, en este caso, una máquina de soporte vectorial. Dentro de este preprocesamiento se hace uso del histograma de gradientes, orientados a la detección del rostro dentro de la imagen, para que esta sea recortada solo al contorno de la cara. De esta manera, se continúa con la extracción de los hitos faciales de la imagen, luego se determina la probabilidad de como cada uno de los sentimientos básicos se encuentran presentes en la expresión de la persona y, en función de estas probabilidades, se determina si la persona está satisfecha o no. Se descubrió que el sistema detectaba correctamente la mayoría de las imágenes ingresadas para las pruebas, sin embargo, se dieron algunos casos donde, a pesar de que el cliente se mostraba satisfecho, este producía ciertos gestos de insatisfacción ocasionados por factores externos.

Palabras clave: satisfacción del cliente, reconocimiento de expresiones faciales, histograma de gradientes orientados, máquina de soporte vectorial, puntos de referencia faciales

Abstract

Camera-based facial expression recognition system to analyze customer satisfaction level in a restaurant

The main objective of this research is to develop a system that recognizes customers' satisfaction or dissatisfaction in a restaurant based on their facial expressions when receiving the service provided by the establishment. The implementation of the system has a series of stages common to the development of a visual computing project, which will begin with data preprocessing to train the classifier to be used in this case: a support vector machine. This data preprocessing uses the histogram of oriented gradients for detecting a face inside an image, so that only the face outline is cut. In this way, facial landmarks of the image are extracted, the probability that each of the basic feelings appears in the facial expression of people is established, and, based on these probabilities, customers' satisfaction or dissatisfaction is determined. The results show that the system correctly detected most of the images entered for the tests; however, there were some cases in which, despite the fact that customers were satisfied, they showed certain dissatisfaction expressions caused by external factors.

Keywords: customer satisfaction, facial expression recognition, histogram of oriented gradients, support vector machine, facial landmarks

1. Introducción

En los últimos años, el uso de cámaras en los establecimientos comerciales, pequeños locales y empresas, públicas o privadas, ha aumentado considerablemente. Su uso, sin embargo, solo se concentra en brindar un servicio de seguridad. Aunque se conoce, por ejemplo, que los videos grabados por las cámaras captan información importante de la persona, entre otras cosas, expresiones faciales. Es de anotar que la expresión facial es uno de los medios que el ser humano tiene para reflejar emociones y deseos; por ello su análisis es un tema en rápido desarrollo y evoluciona gracias a los avances logrados en áreas como la computación visual, el aprendizaje de máquinas y el aprendizaje profundo.

Este artículo de investigación se enfoca en el análisis de expresiones faciales, tomando como sujeto de pruebas a clientes de un establecimiento de comida, con el objetivo de poder determinar si la persona en cuestión está satisfecha o no con el servicio recibido. Para esto se desarrollará un sistema que, mediante una previa comparación con una línea base de expresiones faciales, pueda determinarse si los gestos realizados por el cliente son de satisfacción o insatisfacción en relación a la calidad del servicio prestado por el establecimiento.

En las siguientes secciones se mostrará el estado del arte, donde se presentarán investigaciones pasadas relacionadas con el tema de esta investigación; luego, los antecedentes, donde se analizarán cada una de las técnicas a ser utilizadas para la implementación del sistema. Posteriormente, se explicará la metodología y, finalmente, se desarrollará una discusión de los resultados obtenidos por las pruebas realizadas. Se culminará con una conclusión general sobre la investigación.

2. Estados del arte

La comunicación no verbal es un área que ha estado fuera del ámbito científico-tecnológico durante mucho tiempo y que, en su momento, solo era reconocida por un grupo de profesionales específicos, por ejemplo, los psicólogos o actores. Sin embargo, a inicios y mediados del siglo xx se empezó a tomar real interés en realizar investigaciones sobre cómo se comunica la gente por medio de las expresiones del rostro (Davis, 1994). Uno de los principales investigadores y pioneros es Paul Ekman, quien en uno de sus libros analizó muchos experimentos realizados sobre el rostro desde mediados del siglo pasado, concluyendo que, al ser realizados en conjunto, las expresiones faciales pueden tomarse como indicadores confiables de algunas de las emociones básicas. En 1978 fue publicado el *Sistema de codificación de acciones faciales (Facial Action Coding System, FACS)* por Ekman y Friesen, el cual fue ideado como un sistema que mide toda conducta facial visible en cualquier contexto, sin limitarse a las acciones relacionadas con la emoción. De esta manera, se definieron siete emociones básicas: alegría, sorpresa, furia, tristeza, desprecio, disgusto y contento, cada una de ellas con un patrón codificado dentro de las FACS (Ekman y Oster, 1979).

El reconocimiento de expresiones faciales es una de las nuevas tecnologías para comprender qué expresión y qué emoción emite o realiza el ser humano al extraer, analizar y clasificar las características faciales. Para este propósito existen principalmente tres etapas para el análisis y reconocimiento de expresiones, las cuales son: la detección de rostros, la extracción de las características faciales y la clasificación de estas características (Gao, Jia y Jiang, 2015). Describir y detectar el tipo de emoción correcto que una persona siente, a través de los cambios en la apariencia de su rostro, es la cuestión clave dentro de lo que es el reconocimiento de expresiones faciales. Para ello existen dos enfoques que permiten la descripción de las imágenes faciales, el método basado en características geométricas y el basado en características de apariencia (Ryu, Rivera, Kim y Chae, 2017).

El primer método se concentra en los ángulos y las áreas formadas por ciertos puntos de referencia colocados en las imágenes que representan la cara (Acevedo, Negri, Buemi, Fernandez, y Mejail, 2017). En sí, se realiza una codificación entre las relaciones de ubicación de las partes principales que componen el rostro representado en la imagen, como lo son los ojos, la nariz o la boca (Ryu *et al.*, 2017). Mientras que el segundo método se enfoca en las características de toda la imagen usando diferentes algoritmos como, por ejemplo, *local binary pattern* (LBP) o *linear discriminat analysis* (LDA).

Dentro de este campo, son muchas las investigaciones que se han realizado sobre el uso de las técnicas de visión computacional para el reconocimiento de objetos, rostros e incluso emociones. Algunas de las aplicaciones expuestas en estudios anteriores son, por ejemplo, el trabajo desarrollado por Lago y Jiménez Guarín (2014), el cual hace referencia al uso de reconocimiento de expresiones, no solo con la intención de identificar las emociones básicas, tristeza, felicidad, ira, sorpresa, disgusto y miedo, sino también tratar de identificar otras emociones no tomadas en cuenta en este tipo de investigaciones, como el aburrimiento, el interés y la confusión. En esta investigación se propuso un modelo de inferencia de las emociones mencionadas. Para lograr la construcción de este modelo, los autores hacen uso del entorno de trabajo LuxandFace SDK que permitió extraer 64 puntos característicos faciales en la cara y a partir de las distancias existentes entre los puntos se pudo obtener una línea de base.

Otro trabajo es el desarrollado por Whitehill, Serpell, Lin, Foster y Movellan en 2014, el cual realiza una grabación de algunos estudiantes interactuando con un *software* de entrenamiento de habilidades cognitivas. El análisis de estos videos se realiza mediante un reconocimiento de cuadro por cuadro en el video en el cual se utiliza una técnica de anotación que permite crear una línea base en cuanto a las imágenes, que servirán de muestra para relacionar el sentimiento de compromiso con el cuadro en el que se encuentra; posteriormente, se pasa a revisar los distintos algoritmos de reconocimiento facial y, finalmente, se hace uso de las técnicas de aprendizaje automático para desarrollar detectores del nivel de compromiso que tiene el alumno.

En cuanto a trabajos realizados con máquinas de soporte vectorial, está el estudio de Xia Li en 2014, en el cual se usa un clasificador para realizar la multclasificación de las seis emociones básicas (véase, felicidad, sorpresa, miedo, enojo, tristeza y disgusto) en imágenes de rostros de personas.

Uno de los métodos a ser considerados como parte del desarrollo de la propuesta de solución, que será planteada y detallada más adelante, para el problema propuesto, es el *Histogram of oriented gradients* (HOG), el cual fue planteado inicialmente en la investigación desarrollada por Dalal y Triggs en 2005 llamada *Histograms of oriented gradients for human detection*. En esta investigación, se detalla paso a paso como es desarrollado este descriptor que básicamente se resume en el uso de la dirección de los gradientes distribuidos en toda la imagen. Dalal y Triggs desarrollaron una primera implementación de este descriptor, logrando diferenciar a las personas que caminaban por las calles del escenario de fondo y de otros objetos que estaban a su alrededor.

Otra investigación utilizada como referencia es la de Slim, Kachouri y Atitallah (2018), llamada *Customer satisfaction measuring based on the most significant facial emotion*, en la cual los investigadores plantean el desarrollo de un sistema que permite medir si un cliente se encuentra satisfecho, basándose únicamente en tres emociones, que ellos consideran significantes para determinar el estado de satisfacción, felicidad, sorpresa y neutral. Hacen uso de los hitos faciales, detallados posteriormente, para poder determinar los puntos principales en el rostro de la persona, y según la variación de las distancias entre la agrupación de ciertos puntos, establecer qué emoción de las mencionadas, la persona está expresando. Cabe resaltar, además, que la clasificación de estos puntos para determinar la emoción representada se realiza mediante una máquina de soporte vectorial (SVM), empleada para desarrollar la solución propuesta en esta investigación.

3. Antecedentes

3.1 Facial landmarks

Uno de los principales enfoques utilizados, tanto para la detección de rostros, como para el reconocimiento de expresiones, es encontrar los *facial landmarks*, los cuales son una serie de puntos en el rostro que toman como referencia ciertas partes de la cara. Las partes de referencia generalmente usadas son los ojos, la boca, la nariz, las cejas, el mentón y los bordes del rostro (Ouanan, Ouanan y Aksasse, 2016).

Principalmente, los puntos de referencia del rostro están organizados en dos tipos de clases: los puntos de referencia principales, aquellos puntos prominentes para la identidad facial (por ejemplo, las esquinas de la boca, los ojos, la punta de la nariz y las cejas); los demás puntos de referencia denominados secundarios (la barbilla, los

contornos de las mejillas, los puntos medios de las cejas y los labios). En los últimos años, han aumentado las investigaciones científicas referentes a la visión artificial con el objetivo de localizar los denominados puntos de referencial faciales usando distintos modelos. Esto se debe a las distintas aplicaciones que realizan la detección de los *facial landmarks*.

Entre las diversas aplicaciones se encuentran la compresión de la expresión, el registro facial, el reconocimiento facial y el seguimiento facial, como también la utilizada para la reconstrucción de modelos de rostros en 3D (Ouanan *et al.*, 2016). Para poder determinar estos puntos característicos del rostro, es necesario el uso de algoritmos que permitan localizarlos. A pesar de las investigaciones realizadas hasta la fecha, la localización de estos puntos es un tema desafiante, considerando entre los principales obstáculos la variabilidad, debido a factores intrínsecos, como por ejemplo la variación de las caras entre los individuos, y a valores extrínsecos, como la oclusión, la iluminación y la resolución de la imagen (Ouanan *et al.*, 2016).

3.2 Support vector machine

Las máquinas de soporte vectorial (SVM, por sus siglas en inglés) son una de las principales herramientas establecidas para el área del aprendizaje automático, sobre todo en lo que respecta al trabajo con imágenes (Xia, 2014). Actualmente son utilizadas en una amplia lista de casos, por ejemplo, en el reconocimiento de dígitos escritos a mano, en la categorización de textos e incluso en la identificación de rostros. A diferencia de paradigmas de aprendizaje, como las redes neuronales, en SVM existe una única solución (Campbell y Ying, 2011).

En cuanto a SVM, la mejor manera de dar su explicación es mediante una clasificación binaria. Cabe resaltar que la clasificación, si bien es la función más aplicada de esta técnica, no es para lo único que es utilizada; puede ser requerida, por ejemplo, para realizar una predicción sobre algún tema en particular mediante un análisis de regresión. La SVM, en general, es una máquina de aprendizaje abstracta que aprenderá de un conjunto de datos de entrenamiento y a partir de estos permitirá generalizar y realizar predicciones correctas sobre un grupo de datos de entrada nuevos (Campbell y Ying, 2011).

Con relación a los datos de entrenamiento, estos básicamente son un conjunto de vectores de entrada, generalmente denotados como X_i , en el cual cada vector contiene varias características. Estos vectores de entrada están emparejados con sus respectivas etiquetas, denotadas como Y_i , y a su vez existen m pares ($i = 1 \dots m$) (Campbell y Ying, 2011). Cuando el problema tiene dos clases de datos bien definidas y separadas, el desarrollo del aprendizaje empieza por encontrar un hiperplano de separación que divide la data clasificada para estas dos clases. El hiperplano de separación es generalmente graficado

en representaciones 2D como una línea media que separa un plano, entonces a los dos lados del hiperplano de separación se encontrarán unos puntos de datos etiquetados como $y_i = +1$ y del otro lado estarán etiquetados como $y_i = -1$. Por su parte, el hiperplano de separación se da como $w \cdot x + b = 0$ (donde \cdot denota el producto interno o escalar), b es el sesgo o desplazamiento del hiperplano desde el origen en el espacio de entrada, x son puntos ubicados dentro del hiperplano y la normal al hiperplano, los pesos w , determinan su orientación (ver figura 1) (Campbell y Ying, 2011).

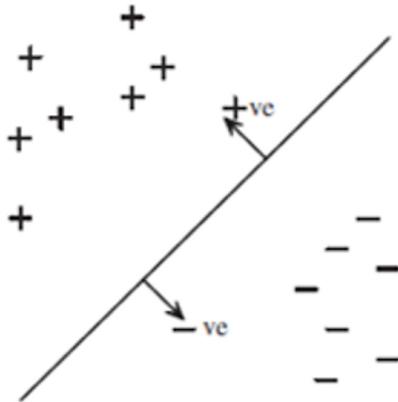


Figura 1. Hiperplano que divide los dos tipos de clases

También está el concepto de hiperplano canónico, formado con los puntos más cercanos de ambas clases (estos puntos son conocidos como vectores de soporte) y la distancia perpendicular entre el hiperplano de separación y un hiperplano canónico es conocido como margen, mientras que la distancia entre los hiperplanos canónicos es denominada como banda de margen (ver figura 4).

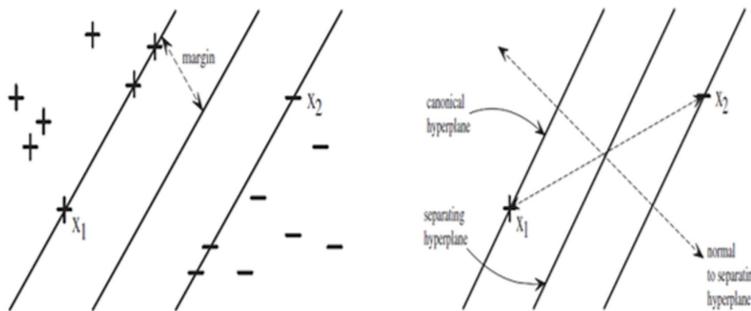


Figura 2. Margen, hiperplano separador e hiperplano canónico

Fuente: Campbell y Ying (2011).

En cuanto a la clasificación binaria mediante SVM se deben considerar dos características importantes:

- a) El límite se minimiza maximizando el margen, γ , es decir, la distancia mínima entre el hiperplano que separa las dos clases y los puntos de datos más cercanos al hiperplano.
- b) El límite no depende de la dimensionalidad del espacio.

Si se considera una clasificación binaria con puntos de datos X_i , con etiquetas correspondientes $Y_i = \pm 1$ y considerando la función de decisión como:

$$f(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) \quad (1)$$

donde \cdot es el producto escalar o interno (entonces $\mathbf{w} \cdot \mathbf{x} \equiv \mathbf{w}^T \mathbf{x}$).

A partir de esta función de decisión, los datos estarán correctamente clasificados si: $y_i (\mathbf{w} \cdot \mathbf{x}_i + b) > 0 \forall i$ desde $(\mathbf{w} \cdot \mathbf{x}_i + b)$ debe ser positivo cuando $y_i = +1$, y debe ser negativo cuando $y_i = -1$.

Para el hiperplano de separación $\mathbf{w} \cdot \mathbf{x} + b = 0$, el vector normal es $\mathbf{w} / \|\mathbf{w}\|_2$ (donde $\|\mathbf{w}\|_2$ es la raíz cuadrada de $\mathbf{w}^T \mathbf{w}$). Por lo tanto, la distancia entre los dos hiperplanos canónicos es igual a la proyección de $\mathbf{x}_1 - \mathbf{x}_2$ sobre el vector normal $\mathbf{w} / \|\mathbf{w}\|_2$, que da $(\mathbf{x}_1 - \mathbf{x}_2) \cdot \mathbf{w} / \|\mathbf{w}\|_2 = 2 / \|\mathbf{w}\|_2$ (ver figura 4, derecha).

Como la mitad de la distancia entre los dos hiperplanos canónicos, el margen es, por lo tanto, $\gamma = 1 / \|\mathbf{w}\|_2$. Maximizar el margen es, por lo tanto, equivalente a minimizar:

$$\frac{1}{2} \|\mathbf{w}\|_2^2 \quad (2)$$

sujeto a las restricciones:

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \forall i \quad (3)$$

Este es un problema de optimización restringido en el que minimizamos una función objetivo (2) sujeta a las restricciones (3). Como un problema de optimización restringido, la formulación anterior puede reducirse a la minimización de la siguiente función de Lagrange, que consiste en la suma de la función objetivo y las m restricciones multiplicadas por sus respectivos multiplicadores de Lagrange. A la siguiente función se le llamará como formulación principal:

$$L(w, b) = \frac{1}{2}(w \cdot w) - \sum_{i=1}^m \alpha_i (y_i (w \cdot x_i + b) - 1) \quad (4)$$

donde α_i son multiplicadores de Lagrange, y por lo tanto $\alpha_i \geq 0$. Como mínimo, podemos tomar los derivados con respecto a "b" y "w" y establecerlos en cero:

$$\frac{\partial L}{\partial b} = - \sum_{i=1}^m \alpha_i y_i = 0 \quad (5)$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^m \alpha_i y_i x_i = 0 \quad (6)$$

Sustituyendo w de (6) en $L(w, b)$, obtenemos la formulación dual, también conocida como Wolfe dual:

$$w(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (7)$$

que debe maximizarse con respecto a α_i sujeto a las restricciones:

$$\alpha_i \geq 0 \quad \sum_{i=1}^m \alpha_i y_i = 0 \quad (8)$$

Un punto importante a tomar en cuenta es que los datos presentados pueden no ser separables linealmente, con lo cual puede que los datos estén superpuestos en el plano y de esta manera no se pueda hallar un hiperplano capaz de separar las dos clases de datos. Para esto se debe determinar una dimensionalidad mayor a la cual se está trabajando, y además se puede hacer uso de un, por ejemplo, *kernel* gaussiano para poder hallar un hiperplano óptimo. Esta introducción de un *kernel* determinado dentro del modelo se conoce como sustitución de *kernel*.

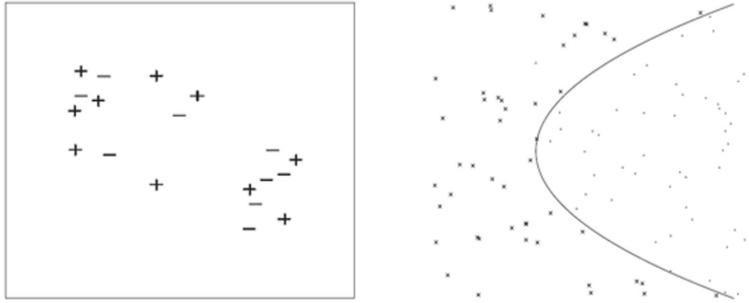


Figura 3. Ejemplo de datos superpuestos en el plano que no pueden ser separados linealmente (izquierda) vs. separación de los datos utilizando una nueva dimensión y un *kernel* gaussiano (derecha)

La idea de usar un *kernel* determinado es, como se mencionó, para casos en los cuales el conjunto de datos no pueda ser separado linealmente, por lo cual se hace uso de este tipo de funciones para poder detectar un hiperplano óptimo sobre este conjunto amplio de características. Por definición, una función *kernel* es una función $K: X \times X \rightarrow \mathbb{R}$ que asigna a cada par de elementos del espacio de entrada, X , un valor real correspondiente al producto escalar de las imágenes de dichos elementos en un nuevo espacio F (Carmona, 2016).

En cuanto a los diferentes tipos de *kernel* que existen, algunos ejemplos son:

- *Kernel* lineal:

$$K(x, x') = \langle x, x' \rangle \quad (9)$$

- *Kernel* polinómico de grado- p :

$$K_p(x, x') = [r \langle x, x' \rangle + \tau]^p \quad (10)$$

- *Kernel* gaussiano:

$$K(x, x') = \exp\left(-r \|x - x'\|^2\right), r > 0 \quad (11)$$

En cuanto al uso de funciones de *kernel* dentro de lo relacionado al reconocimiento de rostros y expresiones, uno de los más utilizados es el *kernel radial basis function* (RBF), el cual es usado por Xia (2014) para el reconocimiento de expresiones faciales. Este *kernel* es, generalmente, representado de la siguiente manera:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad (12)$$

En conclusión, la complejidad del uso de una máquina de soporte vectorial (SVM) dependerá solo de la clasificación de los vectores de soporte dentro del plano y así determinar la dimensionalidad que este debe tener para poder indicar correctamente la ubicación del hiperplano de separación.

3.3 Histograma de gradientes orientados

El histograma de gradientes orientados (HOG) es un descriptor de características utilizado para detectar objetos de imágenes, para lo cual este descriptor hace uso de la distribución de direcciones de los gradientes en distintas porciones de una imagen. Este descriptor fue introducido para el reconocimiento de personas por Dalal y Triggs (2005), quienes además revisaron algunas variantes de descriptores HOG, como por ejemplo R-HOG y C-HOG.

En resumen, la idea central de los descriptores HOG es que, tanto la apariencia del objeto como la forma de este, pueden ser descritos mediante el uso de un histograma que represente las direcciones de los bordes de los objetos en la imagen. Para esto, la implementación de este descriptor consta de tres pasos fundamentales (Khan, Rahmani, Ali Shah, y Bennamoun, 2018):

3.3.1 Cálculo del gradiente

El primer paso para el desarrollo del descriptor es el cálculo del gradiente. El gradiente no es más que el cambio direccional en la intensidad de la imagen, la cual es definida por dos valores: la dirección donde se realiza un mayor cambio en cuanto a la intensidad y la magnitud del cambio en esa dirección. En cuanto al cálculo del gradiente, este puede ser calculado de diversas formas, en el caso del descriptor HOG, se realiza con base en la diferencia de intensidad de los píxeles adyacentes en dirección tanto horizontal como vertical. Es así que, si tenemos una imagen I , y se quiere calcular el gradiente en un punto (x, y) , se empieza obteniendo la diferencia en la dirección tanto horizontal como vertical, esto es denotado de la siguiente forma:

$$dx = I(x + 1, y) - I(x - 1, y) \quad (1)$$

$$dy = I(x, y + 1) - I(x, y - 1) \quad (2)$$

Donde dx y dy son la diferencia en la posición horizontal y vertical, respectivamente, de las intensidades de los píxeles. A partir de lo cual se puede calcular la orientación y la magnitud global del gradiente. Para esto se deben tomar en cuenta las diferencias obtenidas anteriormente, y pasarlas a un eje de coordenadas, donde estos dos valores permitirán definir un vector (vector gradiente del píxel), y así calcular la orientación del gradiente en un punto, denotado como $\theta(x, y)$; además, se puede calcular la magnitud en ese mismo punto, denotado como $g(x, y)$. Es así como la orientación se definirá como el ángulo que forma el vector con el eje horizontal, el cual puede ser calculado mediante el arco tangente:

$$\theta(x, y) = \arctan \frac{dy}{dx} \quad (3)$$

Mientras que el cálculo de la magnitud del gradiente se determina por la longitud del vector:

$$g(x, y) = \sqrt{dx^2 + dy^2} \quad (4)$$

Todo lo anterior es a partir del trabajo con imágenes en escala de grises. Si se tomaran en cuenta imágenes a color, el descriptor debe priorizar el color que domina en forma local para cada píxel, con lo cual para cada píxel se debe calcular el gradiente en cada uno de los tres canales de colores existentes (véase RGB, rojo, verde y azul) y luego se toma el gradiente del canal con mayor magnitud (Dalal y Triggs, 2005).

3.3.2 Cálculo del histograma

El segundo paso para el desarrollo del descriptor es el cálculo del histograma de orientaciones para cada celda definida en la imagen. Para lo cual lo primero que debe realizarse es la división de la imagen en un número fijo de celdas de un tamaño determinado. Este tamaño fijo para cada celda, generalmente suele ser entre 6 y 8 píxeles, tanto en alto como en ancho (Khan *et al.*, 2018). Luego, se deberá considerar como se divide el rango de orientaciones en un número de intervalos fijo.

Lo primero a considerar es si la orientación del gradiente se toma con signo o no, para lo cual, si se elige la primera opción, el rango de orientaciones irá desde 0 a 360°; mientras que, si el signo de la orientación no es tomado en cuenta, el rango ira desde 0 a 180°. Con esta última opción, dos gradientes con la misma dirección, pero sentidos inversos se consideran equivalentes y quedan asignados al mismo intervalo. El otro punto a considerar, es en cuántos intervalos se dividirá el rango de orientaciones. Habitualmente, y considerando un rango de 0 a 180°, este rango es

dividido en 9 intervalos, con lo cual cada intervalo agrupa un rango de orientaciones de 20° (Dalal y Triggs, 2005).

3.3.3 Cálculo del descriptor

Uno de los grandes retos que cualquier descriptor de características tiene es el hecho de combatir la invarianza de determinados aspectos en la imagen (véase cambios de escala, iluminación, etc.). En el caso del detector HOG, al ver un cambio de iluminación y contraste en la imagen, la intensidad del gradiente va a cambiar, por lo cual estos cambios de intensidad también se verán reflejados en los histogramas. Para reducir las diferencias en los resultados de estas imágenes, se necesitará normalizar los valores de los histogramas, con el objetivo de que la magnitud global del gradiente sea similar en cada cambio de iluminación (Dalal y Triggs, 2005).

La normalización será realizada en bloques, que no es más que una conjunción de celdas de la imagen. Generalmente el tamaño del bloque es de 2 x 2 o 3 x 3 celdas. Es así que de este bloque se obtendrán los histogramas de cada celda que está contenida dentro de este, los cuales serán agrupados en un vector no normalizado $v = (x_1, \dots, x_n)$. Basados en esto, se procede a la normalización del vector que contiene los histogramas concatenados de un bloque, la cual es desarrollada dividiendo el vector sobre una norma. Entonces, considerando $\|v\|_k$ donde la norma es k para $k = 1, 2$ y ε es una pequeña constante, se pueden determinar los siguientes factores de normalización:

$$L2 - norm: \quad v' = \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon}}, \quad (5)$$

o

$$L1 - norm: \quad v' = \frac{v}{\sqrt{\|v\|_1 + \varepsilon}}, \quad (6)$$

o

$$L1 - sqrt: \quad v' = \sqrt{\frac{v}{\|v\|_1 + \varepsilon}}, \quad (7)$$

La aparición de la constante ε en las fórmulas tiene como objetivo evitar divisiones por cero en los casos donde la intensidad sea constante en todo el bloque, teniendo como resultado que la magnitud total del gradiente sea cero. Se

debe tomar en cuenta que, la norma de un vector es igual a la raíz cuadrada de la suma de todos sus componentes al cuadrado:

$$\|v\| = \sqrt{\sum_{i=1}^n x_i^2} \quad (8)$$

4. Metodología y experimentación

Antes de comenzar con la explicación de las etapas que se siguieron para el desarrollo de la prueba de concepto, se expondrán las herramientas utilizadas para la implementación de esta. En primer lugar, el desarrollo en general se realizó en una *laptop* con las siguientes características:

- Procesador Intel Core i7-6500 de 2.5 GHz con una RAM de 8GB
- Tarjeta gráfica NVIDIA GEFORCE 940 MX
- Sistema Operativo Windows 10

En cuanto al lenguaje de programación utilizado, la implementación se desarrolló en Python 3.6.2, empleando Miniconda 3. Además, se trabajaron principalmente con las siguientes librerías:

- Scikit-learn: es una Librería para la programación en Python, la cual está especializada en el aprendizaje de máquina.
- NumPy: librería de funciones matemáticas de alto nivel que permite una mejor operación con vectores y matrices.
- Scikit-image: se utiliza básicamente para el procesamiento de imágenes.
- Dlib: librería de *software* multiplataforma escrita en C++. Contiene componentes para trabajar con *machine learning* y procesamiento de imágenes.
- OpenCV: es una librería libre utilizada para los temas de inteligencia artificial.

4.1 Obtención de una base de datos con imágenes de emociones

Como se explicó anteriormente, existe una buena cantidad de bases de datos disponibles para la detección de rostros y el reconocimiento de expresiones. Para esto se tomaron en cuenta dos bases de datos para las respectivas pruebas: la Extended Cohn-Kanade

Dataset (CK+) y la base de datos obtenida de la librería Scikit-learn. La primera fue utilizada para la elaboración del sistema final; mientras que la otra, provisionalmente, para la etapa de preprocesamiento de la imagen. Un ejemplo del contenido de este conjunto de imágenes es mostrado a continuación:

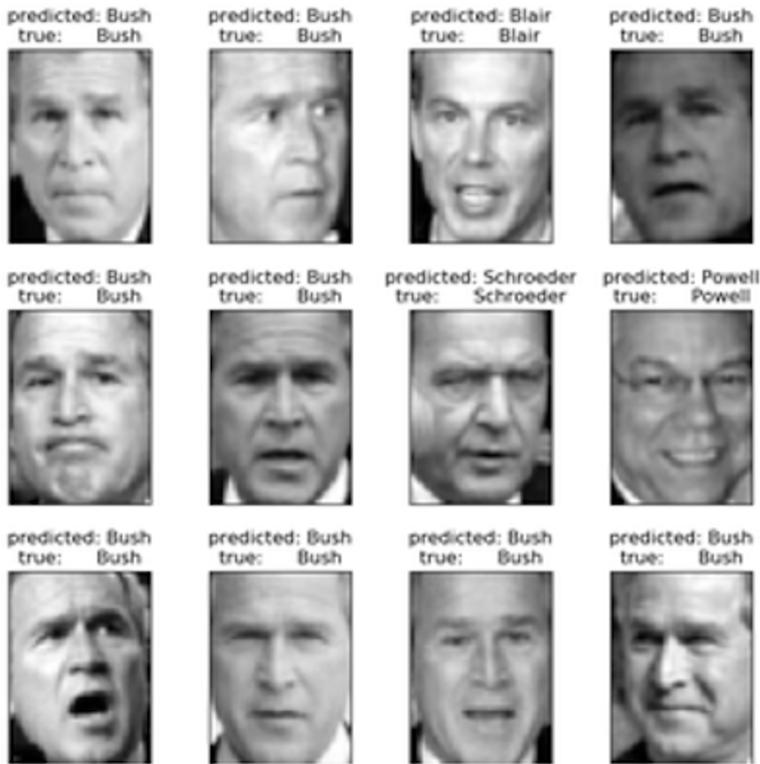


Figura 4. Ejemplo del *dataset* de rostros en Scikit-learn



Figura 5. Ejemplo del *dataset* de rostros en The Extended Cohn-Kanade Dataset (CK+)

Algunos aspectos a tomar en cuenta sobre la base de datos Cohn Kanade es que tiene subdividido cada conjunto de imágenes de emociones según el sujeto de prueba utilizado; además, algunos de los sujetos de prueba no cuentan con una galería de imágenes de todas las emociones que esta base de datos ofrece. Por ejemplo, el sujeto 1 tiene distintas imágenes que muestran las emociones básicas; sin embargo, el sujeto 2 solo tiene imágenes sobre la emoción neutral y la felicidad. Un último punto a tomar en cuenta es que este *dataset* “desarrolla” la creación de la expresión, es decir, cada conjunto de imágenes de emociones representan el paso de un rostro neutral a uno que muestra ciertos gestos que demuestran la emoción. Por lo cual se tuvo que realizar un reordenamiento de las imágenes proporcionadas, separando todas las imágenes en carpetas, cada una según la emoción que se representaba, sin importar el sujeto de prueba que las esté expresando.

Una vez reorganizado el *dataset*, este pasó a un preprocesado para obtener únicamente el área del rostro y desechar el *background* de la imagen. Para ello se usaron los histogramas de gradientes orientados, los cuales recortaron la imagen a solo el área del rostro; además, cada una de estas imágenes fue redimensionada a una escala de 350 x 350 píxeles, esto con el objetivo de estandarizar los tamaños de todas las imágenes obtenidas.

Una vez realizado esto, se pasó a extraer los *facial landmarks* de cada una de las imágenes del *dataset* modificado, los cuales fueron vectorizados para posteriormente servir de entrada para el entrenamiento de la máquina de soporte vectorial usada para la clasificación de las emociones reconocidas en las imágenes. Para ello se empleó 80 % de la data extraída al azar para ser usada como data de entrenamiento, considerando el 20 % restante como data de prueba para el clasificador. Esta máquina contó con un *input* de imágenes que no fueron rostros, con esto se obtuvo un modelo predictor.

4.2 Obtención de videos de personas en un restaurante

Como se mencionó, los videos obtenidos para esta investigación fueron recolectados de la web, siendo debidamente escogidos bajo ciertas características específicas para la finalidad de esta investigación y en favor del desarrollo del sistema. Estas características son, por ejemplo, que los videos deben mostrar la grabación del rostro de la persona a lo largo del servicio recibido en el restaurante; además, para poder validar el resultado del procesamiento, la persona en cuestión deberá dejar su opinión clara y concisa, en algún momento de la grabación, sobre el servicio brindado para así comparar su opinión con los resultados obtenidos. De los seleccionados, se debe tener en cuenta que no todos los minutos del video son necesarios para realizar el análisis propuesto en esta investigación, por lo cual se deberá realizar una extracción de ciertos momentos que puedan ser utilizados por el sistema, es decir, solo fueron necesarios fragmentos en los cuales la persona recibió el servicio y opinó sobre este. Esta extracción de las partes del video la realizó el autor de forma manual.

4.3 Procesamiento de las imágenes captadas del video

Al igual que con las imágenes de las bases de datos obtenidas, en esta etapa se realizó un procesamiento de los *frames* obtenidos del video, el cual comenzó cambiando las imágenes de formato RGB a escala de grises. Una vez que todas estas imágenes fueron unificadas, se usó el histograma de gradientes orientados para realizar el recorte de la imagen solo al área del rostro. Las imágenes fueron pasadas a un tamaño estándar de 350 x 350 píxeles para un uso más rápido del algoritmo, este cambio es de carácter totalmente subjetivo por el autor de este documento y con base en un rango calculado según los trabajos anteriores de otros investigadores.



Figura 6. Ejemplo de características HOG



Figura 7. Resultado de la detección del rostro

La imagen presentada es solo una imagen referencial (escogida por el autor para realizar tan solo una prueba inicial sin contexto) de cómo funciona esta parte del sistema. A continuación, se mostrarán ejemplos de imágenes extraídas a partir de los videos obtenidos (figuras 8 y 9) y como fue el resultado del procesamiento final de estas:



Figura 8. Frame original del video del restaurante



Figura 9. Escalas de grises

4.4 Extracción de las características

Es importante tener en cuenta la cantidad de data que será tomada para realizar un análisis de expresiones. Se conoce que un video es un conjunto de imágenes que pasan por segundos de reproducción. Los videos tiene decenas y hasta centenas de *frames* (imágenes) por segundo, la diferencia entre un *frame* y su consiguiente no tendría casi ninguna diferencia, con lo cual el analizarlos traería consigo un gasto computacional innecesario. Por este motivo se decidió obtener un *frame* cada medio segundo del video, considerando este rango de tiempo lo suficientemente amplio para un mejor análisis y diferencia entre imágenes. Los *frames* seleccionados fueron almacenados en una carpeta aparte, la cual sirvió como un nuevo *dataset*, el cual fue utilizado para la extracción de las características (*facial landmarks*).

Una vez las imágenes estuvieron correctamente procesadas, se realizó la extracción de los *facial landmarks*. Para este proceso, se usó la librería Dlib, la cual contiene un predictor de estos puntos característicos sobre el rostro. Cabe recalcar que el predictor no viene junto con la librería, sino que debe ser descargado previamente para su utilización, el nombre de este es "shape_predictor_68_face_landmarks.dat" (ver figura 10).

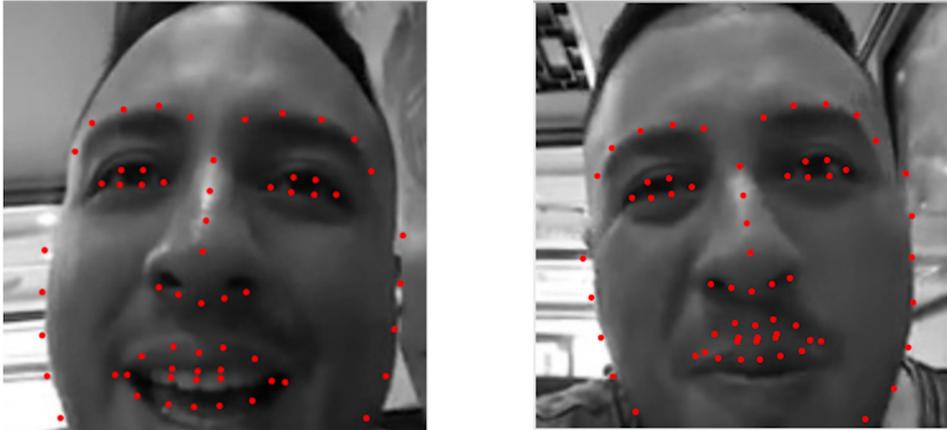


Figura 10. Ejemplo de los facial landmarks

Una vez obtenidos estos facial landmarks, las coordenadas correspondientes a cada uno de estos puntos son almacenados. Después se realizó una normalización de las distancias entre los puntos demarcados respecto del centro del rostro. Estos landmarks "normalizados", posteriormente, fueron almacenados en un vector mediante el uso de la librería Numpy. El proceso anteriormente detallado se utilizó también con las imágenes obtenidas para el entrenamiento de la máquina de vectores.

4.5 Clasificación de las características

Teniendo el conjunto de datos (*landmarks*) normalizados y agrupados en vectores, estos fueron ingresados a la máquina de soporte vectorial para ser clasificados y así determinar la probabilidad de que cada una de las emociones básicas estén presentes en la imagen analizada. Esta clasificación se basa en el modelo previamente creado con el *dataset* descrito en el primer apartado de esta metodología. Por ejemplo, si tomamos como referencia la imagen del sujeto de prueba de las figuras 6 y 7, luego de todo el proceso descrito en los pasos anteriores, nos daría como resultado el siguiente:

- Enojo: 0,25003043
- Desprecio: 0,02009596
- Disgusto: 0,01237244
- Miedo: 0,0028604
- Felicidad: 0,00137583

- Neutral: 0,67577552
- Tristeza: 0,03366757
- Sorpresa: 0,00382185

Con lo cual el sistema determinaría que la emoción que se expresa en la figura es una emoción neutral, detectando también, de una manera menos clara, el sentimiento de enojo. Considerando que en las investigaciones revisadas sobre este tema no se encontró una manera para determinar un patrón o forma de los gestos de la cara que concluyeran si una persona está satisfecha o no, se decidió que, al momento del servicio brindado, las emociones positivas fueran consideradas como un nivel de satisfacción bueno del cliente; al contrario, las emociones negativas, como insatisfacción del cliente. Es así como, según la investigación realizada por Slim, Kachouri y Atitallah, las emociones que están más relacionadas a la satisfacción serán la felicidad y la sorpresa, mientras que las otras emociones serán consideradas como insatisfacción.

Por ello, se tomaron en cuenta solo las dos emociones más resaltantes destacadas por el SVM para realizar la clasificación de la satisfacción y, además, se consideró que estas emociones debían tener una relevancia en la probabilidad mostrada, para este caso se tomó en cuenta la emoción en caso de que su probabilidad fuese superior a 0,2. Por ejemplo, en los resultados mostrados se puede denotar que la emoción que más probabilidad tiene de estar en la imagen es la neutral, sin embargo, la emoción de enojo es la segunda emoción con mayor probabilidad y, además, su probabilidad es mayor a 0,2, por lo cual el sistema arrojaría como resultado que la persona está insatisfecha. En el caso de tener una probabilidad de emoción que sea superior a 0,2, solo se considerará esta, según su carácter positivo o negativo, para determinar si la persona se encuentra satisfecha o no.

4.6 Evaluación del rendimiento y los resultados obtenidos: resultados del sistema con videos de personas en un restaurante

Para la realización de las pruebas del sistema desarrollado, se tuvieron en cuenta videos obtenidos de la plataforma YouTube de personas que “reaccionaban” a la atención y servicio recibido en distintos restaurantes. Para ejemplificar el desarrollo del sistema, se seleccionó uno de los videos al azar, del cual se obtuvieron 1328 imágenes, teniendo en cuenta que el video duraba alrededor de los 11 minutos y que por cada medio segundo se escogía un *frame* de este. Todas estas imágenes fueron pasadas, primero, por el preprocesamiento mencionado, con lo cual las imágenes quedaron en escala de grises y a 350 x 350 pixeles; además, se eliminó el *background* de cada una, realizando el recorte en el contorno del rostro mediante el uso de los histogramas de gradientes orientados,

sin embargo, no todos los *frames* captados contenían el rostro de una persona, por lo cual solo fueron almacenados y procesados aquellos en los cuales el rostro de la persona sí fue detectado.

Con estos pasos, el conjunto de imágenes a ser analizado disminuyó a 562 imágenes, todas ellas contenían la imagen de un rostro que podía ser analizado. Posteriormente, se pasó a realizar el análisis de gestos mediante la obtención de los *facial landmarks* de cada imagen, con lo cual una por una iba ingresando a la máquina de vectores para la clasificación de las emociones (figura 11).

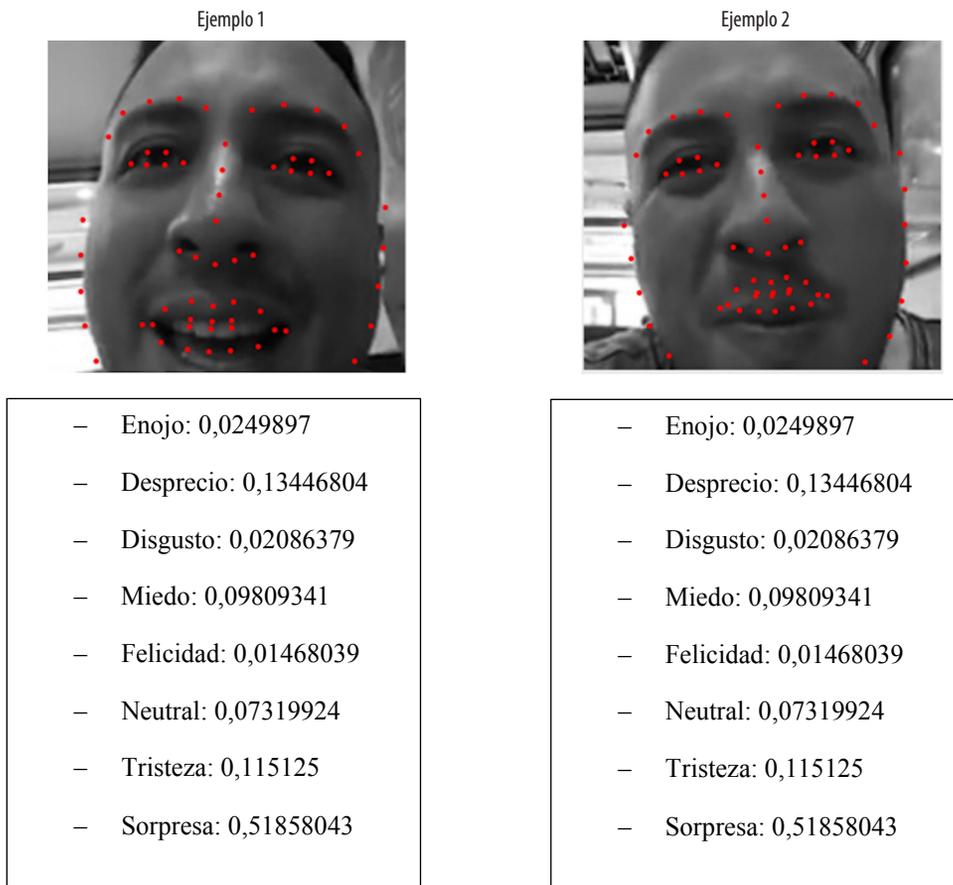


Figura 11. Resultado con los dos rostros mostrados anteriormente

Considerando estos resultados, el sistema muestra que, en el caso del ejemplo 1, la emoción que sobresale es de sorpresa, seguida de la emoción de tristeza, sin embargo, esta última se encuentra por debajo de la probabilidad anteriormente establecida (mayor a 0,2), con lo cual el sistema clasifica que la persona se encuentra satisfecha, puesto que

la única emoción que toma en cuenta para la decisión es la de sorpresa, catalogada como positiva, mientras que la emoción de tristeza es dejada de lado, puesto que no supera el margen establecido para ser considerada como significativa. Por otra parte, en el ejemplo 2, la emoción que más probabilidad tiene es la neutral, sin embargo, la segunda emoción es la tristeza, superando el mínimo establecido para ser considerado por el sistema, con lo cual el sistema clasifica que la persona en esta imagen se encuentra insatisfecha, puesto que la segunda emoción en consideración se encuentra dentro del conjunto de emociones categorizadas como negativas.

Un punto que hasta el momento no se ha abordado en esta investigación es la posibilidad de que dos o más personas aparezcan en la misma escena. Bajo este contexto, el sistema solo tomará en cuenta el rostro de la primera persona que reconozca en la imagen para su análisis. Por ejemplo, se tiene la siguiente imagen, obtenida de un video diferente, en el cual se muestran dos personas en un restaurante. El sistema toma en cuenta solo uno de los rostros y realiza el preprocesamiento anteriormente descrito.



Figura 12. Frame con dos personas en el restaurante

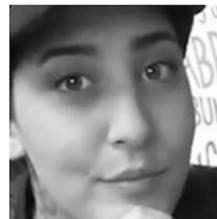


Figura 13. En escalas de grises

Con estos resultados, el sistema podría sufrir cierta variación debido a que puede estar reconociendo a otra persona que no es el objeto del estudio. Es por ello que, para evitar estas situaciones, los videos a ser analizados deben ser previamente recortados a los tramos en los cuales la persona que está recibiendo el servicio aparezca sola. Esto no afecta el objetivo final de la investigación, puesto que la finalidad de esta es realizar el análisis individual de una persona y no de manera grupal. Finalmente, todos los *frames* que han sido procesados por el sistema son evaluados de manera conjunta, es decir, que si en conjunto la mayoría de *frames* han sido etiquetados como una imagen donde la persona muestra satisfacción, se considerará que la persona está satisfecha durante todo el proceso del servicio brindado. Esto se realiza debido a que, durante el procesamiento del video extraído, la persona puede mostrar ciertos gestos que pueden ser catalogados como insatisfacción, los cuales no determinan que la persona este insatisfecha con el servicio.

4.7 Validación de los resultados del sistema

En este apartado se debe tomar en cuenta que la forma ideal de validar estos resultados sería contar con la opinión directa del cliente sobre el servicio brindado en el restaurante, mediante el uso de una encuesta de satisfacción de servicio correctamente enfocada. Sin embargo, los videos analizados fueron obtenidos, en su totalidad, mediante la plataforma de internet YouTube, por lo cual realizar este tipo de validación sería imposible. Es por ello que, con el fin de validar los resultados del análisis, los videos elegidos fueron seleccionados bajo ciertos parámetros que permitieron llevar a cabo la validación de los resultados. Por lo tanto, solo se tomaron en cuenta aquellos videos donde la persona que “reacciona” a la atención recibida, posteriormente da su opinión explícita sobre el tema, con lo cual se tendría la forma de relacionar el sentimiento detectado por el sistema (satisfacción o insatisfacción) con lo que verdaderamente opina el cliente, validando de esta manera los resultados obtenidos.

Para ilustrar lo comentado, se tomará en cuenta el ejemplo 2. Según los resultados arrojados por el sistema, la persona fue catalogada como insatisfecha, siendo las emociones con más alta calificación la neutral y la tristeza. Por otro lado, según el video original del cual se sacaron las imágenes, en ese lapso de tiempo el cliente dice explícitamente lo siguiente (basado en los subtítulos generados por la misma plataforma): “Sinceramente, le voy a poner un 5 sobre 10, porque no es como que ¡huy!, normal, no me sorprendió tanto”, infiriendo de esto que su expectativa era mayor y que, en cierta medida, no se encontraba contento con lo recibido. De esta manera, se puede afirmar que el sistema clasificó correctamente a la persona como insatisfecha.

5. Discusión

Para realizar una correcta evaluación y prueba del sistema, en general, se tomaron en cuenta diversos videos de varios autores, con distintos rasgos faciales para acopiar la mayor cantidad de casos posibles. Bajo este contexto, los resultados que arrojó el sistema no fueron 100 % correctos, algunos de los casos más importantes a tomar en cuenta se describen a continuación. En primer lugar, el sistema desarrollado no es capaz de determinar el “contexto” de la escena, es decir, que, por ejemplo, si la persona que recibe el servicio en el restaurante llega a este con una actitud negativa, por diversos motivos que se desconocen, posiblemente, aun siendo la atención buena, mostrará expresiones faciales negativas, con lo cual el sistema clasificará que el cliente se muestra insatisfecho, sin embargo, bajo este escenario no se puede determinar que esta insatisfacción sea consecuencia del servicio brindado.

Otro caso a tomar en cuenta es cuando los clientes realizan gestos espontáneos que se encuentran fuera de contexto, por ejemplo, el cliente puede estar recibiendo una buena atención y mostrar gestos de satisfacción por esto, sin embargo, en algún momento de

este proceso (que para el sistema es representado como un *frame*/imagen del momento), el cliente puede realizar algún gesto negativo, debido a algo que le dijo la persona que lo acompaña o por algún motivo externo al servicio, en este caso, el sistema clasifica al *frame* analizado como insatisfacción. Sin embargo, el fin de esta investigación es determinar la satisfacción o insatisfacción del cliente durante todo el proceso del servicio brindado, por lo tanto todos los resultados de cada *frame* utilizado fueron analizados en conjunto.

6. Conclusiones

Las empresas y establecimientos están buscando constantemente acaparar la mayor cantidad del mercado de su rubro en el cual se desenvuelven. Es así como uno de los puntos a ser tomados en cuenta, por no decir el más importante, es el hecho de lograr fidelizar a sus clientes, mejorando los aspectos que hacen que estos se sientan satisfechos con lo que reciben y corrigiendo aquellos que los disgustan. La presente investigación tiene como principal objetivo determinar la satisfacción o insatisfacción de un cliente al momento de recibir un servicio brindado por un restaurante, esto con el fin de poder brindar a este tipo de establecimientos, y en general a todos aquellos en los cuales pueda aplicarse el trabajo desarrollado, una nueva herramienta con la cual se pueda añadir un valor agregado en el proceso de atención al cliente.

Esta investigación hace uso de diversas técnicas y algoritmos de visión computacional como los histogramas de gradientes orientados, los cuales fueron utilizados para la detección del rostro de la persona, el uso de los *facial landmarks*, para determinar los puntos característicos del rostro, con lo cual se identificó qué sentimiento expresaba el cliente según el gesto que mostraba. También se hizo uso de una máquina de soportes vectorial para la clasificación de los datos obtenidos.

En un principio, la presente investigación tuvo como objetivo desarrollar las pruebas del sistema haciendo uso de videos recolectados de manera propia en un restaurante, sin embargo, si esto se realizaba, se debía pedir el consentimiento de cada una de las personas que serían grabadas, limitando la obtención de los videos a la buena voluntad de los clientes. Además, una de las principales características planteadas para esta investigación era obtener gestos naturales expresados por la persona, sin embargo, al ser estas informadas de que serían grabadas, la data obtenida podría verse sesgada. Por ello, se determinó que el proceso de las pruebas del sistema desarrollado se realizaría con imágenes extraídas de videos colgados en la web. Los resultados obtenidos mostraron que la clasificación de la satisfacción e insatisfacción de la persona se realizaron correctamente en la mayoría de los casos; sin embargo, existieron casos, por motivos externos al servicio, y que no se pudieron controlar dentro de esta investigación, en donde el sistema clasificó como insatisfacción ciertos *frames* del video en los cuales no hubo responsabilidad por parte del servicio brindado.

Para posibles trabajos futuros, se deben tomar en cuenta, principalmente, las limitantes que se tuvieron en la realización de esta investigación, como, por ejemplo, el hecho de no poder realizar las pruebas con data propia obtenida de un restaurante. Esta investigación puede ser tomada, además, bajo otro contexto distinto al de clientes en un restaurante, se podría aplicar, por ejemplo, en las ventanillas de un banco o en otro lugar donde el rostro de la persona pueda ser captado directamente. Además, se pueden tomar en cuenta otros tipos de técnicas distintas a las utilizadas en esta investigación para el desarrollo del sistema y, de esa manera, poder tener una comparación en el rendimiento de cada sistema.

Referencias

- Acevedo, D., Negri, P., Buemi, M. E., Fernandez, F. G., y Mejail, M. (2017). A simple geometric-based descriptor for facial expression recognition. *12th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 802-808.
- Campbell, C., y Ying, Y. (2011). Learning with support vector machines. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(1), pp. 1-95.
- Carmona Suárez, E. J. (2014). Tutorial sobre máquinas de vectores soporte (SVM). Recuperado de [http://www.ia.uned.es/~ejcarmona/publicaciones/\[2013-Carmona\]%20SVM.pdf](http://www.ia.uned.es/~ejcarmona/publicaciones/[2013-Carmona]%20SVM.pdf)
- Dalal N., y Triggs, B. (2005). Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1, pp. 886-893. doi:10.1109/CVPR.2005.177
- Davis, F. (1994). *El lenguaje de los gestos*. Buenos Aires: Emecé Editores.
- Ekman P., y Oster, H. (1979). Expresiones faciales de la emoción. *Annual Review of Psychology*, 30, pp. 527-554.
- Gao, G., Jia, K., y Jiang, B. (2015). An automatic geometric features extracting approach for facial expression recognition based on corner detection. *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pp. 302-305. Adelaide, SA, Australia: IEEE. doi:10.1109/IIH-MSP.2015.33
- Khan, S., Rahmani, H., Ali Shah, S. A., y Bennamoun, M. (2018). *A guide to convolutional neural networks for computer vision*. Morgan & Claypool Publishers.
- Lago, P., y Jiménez Guarín, C. (2013). An affective inference model based on facial expression analysis. *IEEE Latin America Transactions*, 12(3), pp. 423-429.

- Ouanan, H., Ouanan, M., y Aksasse, B. (2016). Facial landmark localization: Past, present and future. *4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pp. 487-493.
- Ryu, B., Rivera, A. R., Kim, J., y Chae, O. (2017). Local directional ternary pattern for facial expression recognition. *IEEE Transactions on Image Processing*, 26(12), pp. 6006-6018.
- Slim, M., Kachouri, R., y Atitallah, A. (2018). Customer satisfaction measuring based on the most significant facial emotion. *15th IEEE International Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 502-507.
- Whitehill, J., Serpell, Z., Lin, Y.-C., Foster, A., y Movellan, J. R. (2014). The faces of engagement: Automatic recognition of student engagement from facial expressions. *IEEE Transactions on Affective Computing*, 5(1), pp. 86-98.
- Xia, L. (2014). Facial expression recognition based on SVM. *7th IEEE International Conference on Intelligent Computation Technology and Automation*, pp. 256-259.

PROGRAMACIÓN POR OBJETIVOS PARA EL DIMENSIONAMIENTO Y LA ASIGNACIÓN DE UNA FLOTA DE CAMIONES EN UNA EMPRESA MINERA

Edmundo Quiroz-Silva
edmundoqueiroz@belcorp.biz
Universidad de Lima. Lima, Perú

Resumen

La toma de decisiones en la minería es un proceso complicado. Las decisiones con respecto al dimensionamiento y a la asignación de la flota de camiones a diferentes minas, en consideración con el cumplimiento de la producción y del costo total objetivo, son de suma importancia para las empresas mineras, ya que esto afecta directamente las utilidades que pueden obtener. Este documento presenta un modelo de programación por objetivos (con base en la programación lineal) para optimizar las operaciones mineras de extracción en minas socavón, tomando en consideración tres objetivos: (a) Minimizar las desviaciones de la producción objetivo de diferentes minerales, (b) minimizar el excedente del costo operativo total designado y (c) maximizar el uso de la capacidad máxima de las plantas concentradores. El modelo determina la cantidad de camiones necesarios y proporciona la asignación óptima a cada mina, mientras se abordan los objetivos y las limitaciones de la operación minera; el modelo planteado es luego validado utilizando datos reales de una empresa minera peruana.

Palabras clave: asignación de camiones, minería, programación lineal, programación por objetivos

Abstract

Goal programming for sizing and allocation of a fleet of trucks in a mining company

Decision making in mining is a complicated process. Decisions regarding the sizing and allocation of the fleet of trucks to different mines, considering the compliance with production and the total objective cost, are of paramount importance to mining companies as this directly affects their profits. This document presents a model of goal programming (based on linear programming) to optimize mining operations in underground mining taking into account three goals: (a) To minimize deviations from the target production of different minerals, (b) to minimize the surplus of the total designated operating cost and (c) to maximize the use of concentrator plants' maximum capacity. The model determines the number of trucks needed and provides the optimal allocation to each mine while addressing the goals and limitations of the mining operation. The proposed model was validated using real data from a Peruvian mining company.

Keywords: truck allocation, mining, linear programming, goal programming

1. Introducción

El sector minero es, en principio, un negocio de costos, donde el producto final se controla con los precios que dicta el mercado (basados en la oferta y la demanda mundial), sobre el cual ninguna entidad tiene control directo. Sin embargo, donde las empresas sí tienen control es sobre los costos propios de la operación (perforación, extracción, transporte y tratamiento del mineral), dependiendo de un adecuado control de los mismos será el margen de ganancias que alcancen. Esta es razón suficiente para que las empresas mineras estén atentas a obtener una adecuada optimización de costos, sobre todo en las épocas donde el precio de los metales baja considerablemente. En la industria minera, el dimensionamiento y la asignación de la flota de camiones en las operaciones de extracción y tratamiento de los minerales es un proceso crítico; una correcta asignación de camiones genera ahorros evidentes. Sin embargo, encontrar la solución ante esta clase de problema no es sencillo, pues engloba un gran número de variables. Muchas veces el enfoque de respuesta que se da no es el adecuado, lo cual genera sobrecostos y errores. El uso de modelos matemáticos es necesario para darle frente a este problema de optimización.

Para mitigar los riesgos de no cumplir con la demanda de mineral, debido a incertidumbres de las operaciones, los operadores de minas a menudo asignan más camiones de lo necesario. Este enfoque de aversión al riesgo conduce al uso ineficiente de camiones, lo que resulta en largas colas de camiones en los vertederos, en palas, o ambos. Su asignación, aun cuando no sean necesarios, deja menos camiones para otras tareas, como el transporte de desecho, que a veces es crítico. En respuesta, los operarios mineros suelen resolver la escasez de camiones a corto plazo mediante el alquiler de otros vehículos, pero esta resolución es costosa (Ta, Ingolfsson y Doucette, 2013).

Por lo general, los despachadores de camiones programan la asignación al comienzo en turnos de 12 horas, basados en datos y experiencias del usuario. Dependiendo de qué tanta experiencia haya tenido el encargado de la asignación, se puede esperar una mayor o menor eficiencia. Asegurar que los despachadores tengan las habilidades requeridas para la asignación eficiente de camiones requiere un proceso completo de capacitación, pero esa capacitación no garantiza la coherencia u optimización en la asignación de camiones. En relación a lo anterior, se ha estimado que el costo del transporte en las minas de tajo abierto representa hasta el 60 % de los costos totales de extracción; por otro lado, el 32 % de la energía total utilizada en las minas es consumida por las operaciones de transporte (Zhang y Xia, 2015; Sahoo, Bandyopadhyay y Banerjee, 2014). De manera similar, en minas subterráneas, las operaciones de carga, acarreo y descarga consumen hasta el 26 % de la energía (LCG Energy Management Group, 2009).

Debido a la escala de operaciones en la minería, incluso una pequeña mejora en la eficiencia operativa se traduce en ahorros sustanciales durante las operaciones en cualquier tipo de

mina (Burt *et al.*, 2005). Por esto es importante trabajar profundamente en la planificación de las operaciones por las ventajas que se obtendrán gracias a una buena utilización de los recursos. El resultado de una inadecuada planificación conlleva a tener sobrecostos innecesarios que reducen las ganancias, generan una disminución de la ventaja competitiva y, finalmente, terminan en la pérdida de oportunidades. El modelo de costos de la operación minera es bastante complejo, puesto que agrupa una serie de parámetros, basado en esto, la utilización de un modelo matemático es necesario para calcular el número óptimo de camiones y determinar la mejor asignación.

2. Contexto

Se han utilizado diversos modelos de programación matemática para la optimización de recursos en la industria minera. Esta optimización consiste en la determinación óptima del número de camiones que se deben usar para el transporte del mineral. El desarrollo de una solución pertinente, ante esta clase de problemática, necesita de la identificación y del modelamiento de la agrupación de restricciones y variables que afectan directamente el proceso, esto con el objetivo de que se pueda optimizar la asignación de los recursos (Crovo, Martín y Rojas, 2007).

Según Alvarado (2009), en cualquier situación en donde participen una cantidad determinada de variables no negativas que, además, puedan vincularse entre sí, a través de relaciones de igualdad o de desigualdad, y que evidencien las restricciones del contexto con el objetivo de optimizar una operación, puede ser desarrollado a través de un modelo de programación matemática. Asimismo, si las variables y las restricciones relacionadas al modelo se definen mediante proposiciones lineales, se puede determinar como un modelo de programación lineal.

Según Zhang y Xia (2015), la gestión de la flota de camiones se modela como un problema de programación lineal entera, que tiene como objetivo cumplir la meta de producción con un costo operativo mínimo. Este autor enfoca el problema de tal manera que su modelo esté compuesto por la determinación del número de viajes, desde una zona de extracción a un sitio de descarga, considerando la duración de la jornada de trabajo, de modo que la producción objetivo sea alcanzada y el costo de operación del camión se reduzca al mínimo. Como el costo de operación del camión está directamente relacionado con la cantidad de camiones utilizados, siempre es deseable usar la menor cantidad de camiones para terminar el trabajo de transporte requerido, por tal motivo la investigación también se enfoca en identificar el tamaño óptimo de la flota de camiones.

Otra investigación relacionada es la de Temeng, Francis y Frendewey (1997), quienes adicionalmente a buscar la optimización de la flota de camiones, también se enfocaron en la optimización de las rutas de viaje, pues estas están directamente relacionadas con el tiempo

de viaje desde los puntos de carga, a través de las palas, hasta las zonas de descarga. La maximización de la producción se logra utilizando las rutas de viaje más rápidas en tiempo tanto como sea posible. Los autores encaminan su modelo matemático y su algoritmo de solución para tener respuestas rápidas a cambios inesperados. De acuerdo a lo mencionado, formulan un algoritmo utilizando programación lineal que determina el número óptimo de palas, el número óptimo de camiones, los criterios de asignación de camiones a las palas y un modelo de transporte para la optimización de rutas.

Por otra parte, los autores Ta, Kresta, Forbes y Marquez (2005) desarrollan un modelo de asignación de camiones a través de una programación estocástica, dado que los objetivos de esta solución dependen de variables complejas como la configuración física de la mina, las condiciones de operación, la economía u objetivos de gestión y la cultura de la empresa. Los autores indican que, normalmente, en el desarrollo de los algoritmos matemáticos, utilizando programación lineal, se basan en parámetros con valores promediados, como, por ejemplo, la carga del camión y el tiempo de ciclo del camión, que a su vez están sujetos de otros parámetros con valores promediados como tiempo de carga, tiempo de descarga, tiempo de viaje de ida lleno, tiempo de viaje de regreso vacío, etc. La utilización de una programación estocástica se ve beneficiada siempre y cuando se trabaje con parámetros con valores inciertos y de corte aleatorio.

En la investigación realizada por Upadhyay y Askari-Nasab (2016), en una operación minera de hierro a tajo abierto en el sur de Irán, se presenta otro caso de optimización en la asignación de camiones, pero asociado a las palas. Los autores buscan maximizar la producción, minimizar las desviaciones respecto al tonelaje y a la cantidad de mineral que se suministra a cada planta (con respecto al esperado) y minimizar el costo operativo. Para ello, plantean la utilización de la programación lineal entera mixta, pero haciendo uso de un enfoque basado en la programación por objetivos (dado la cantidad de problemas que buscan abordar a la vez). Las variables que consideran son: las palas, los bancos de extracción, una flota heterogénea de camiones, los tipos de material (mineral y desmonte) y los destinos (plantas concentradoras, *stockpiles* y botaderos). Según explican, el modelo cumplió con los objetivos propuestos, elevando la utilización promedio de plantas, camiones y palas (en 99 %, en 92 % y en 95 %, respectivamente).

Un estudio realizado en la unidad de Kelvin, sobre una operación minera subterránea de cobre, por los investigadores Nehring, Topal y Knights (2010), muestra como, al utilizar la programación lineal entera mixta, se puede minimizar el uso del equipamiento operativo en interior de mina, compuesto por camiones y cargadores frontales de bajo perfil, para darle cumplimiento a la producción objetiva establecida. En este estudio, se revisa la particularidad del acarreo minero en operaciones de socavón, donde se utilizan los camiones para movimientos de mineral por encima de 1 km y los cargadores frontales de bajo perfil para distancias menores. En el desarrollo del modelo matemático utilizan las siguientes variables: una flota de camiones y de cargadores frontales de bajo perfil, la producción objetivo y las distancias de traslado de mineral. Según

los resultados obtenidos se confirmó una evidente optimización en el uso de equipamiento de acarreo sobre las prácticas de asignación manual que se venían realizando.

El uso de la técnica matemática de programación lineal también fue empleado en el trabajo realizado por Bajany, Xia y Zhang (2017) en una operación minera de tajo abierto. En ese trabajo los autores buscaban minimizar el uso de combustible de una flota de camiones heterogénea, cumpliendo con la demanda de los sitios de descarga. Teniendo en cuenta esto, lo que su modelo determina es la cantidad óptima de viajes que cada camión realiza en cada ruta de la mina, teniendo en cuenta variables como capacidad, velocidad cuando están vacíos y velocidad cuando están cargados; la capacidad, consumo de combustible cuando están inactivos y el consumo de combustible cuando están en operación de las palas y la topografía de la mina (distancias entre las palas y los sitios de descarga, demanda por hora de los sitios de descarga). Tal como presentan en su investigación, la solución de esta investigación muestra una notoria disminución en el consumo de combustible de los camiones y de las palas (8,82 % y 4,49 %, respectivamente), logrando así un ahorro de combustible total de 4,64 %.

Según la literatura revisada, y como lo sostiene Elbrond (1979), la solución más adecuada para la optimización de recursos mineros (por ejemplo, la flota de camiones) es la simulación completa de todo el proceso y la utilización de procedimientos de optimización eficientes (por ejemplo, la programación lineal).

3. Metodología

3.1 Programación lineal

Según Stroud (2003), la programación lineal se define como “un método matemático para resolver un problema de optimización en donde la función objetivo es lineal y las restricciones son ecuaciones lineales o desigualdades lineales”. Aunque la programación lineal es aplicable en una amplia variedad de contextos, ha sido desarrollada y aplicada con más frecuencia en las decisiones relacionadas con producción (Salvatore y Brooker, 2015).

Con el fin de aplicar la programación lineal en operaciones específicas, es necesario formular un modelo matemático. El término “formulación” se refiere al proceso de convertir las descripciones verbales y los datos numéricos en expresiones matemáticas, que representan relaciones entre las variables de decisión, la función objetivo y las restricciones (Lawrence, 2017).

Una vez formulado el problema como un programa lineal, se pueden usar herramientas computacionales como STORM, TORA, QSB, LINDO, etc., para solucionar el problema. En este sentido, la resolución del modelo es relativamente sencilla. La parte más difícil de aplicar esta metodología es formular el problema e interpretar la solución (McDill, 1999).

En el campo de optimización, el significado del término “solución” no es necesariamente la respuesta final de un problema. Derigs (2009) explica que, dentro de este ámbito, cualquier especificación de valores para las variables de decisión se denomina una solución, independientemente de si es o no una opción deseable —o incluso permisible. Al resultado que satisface todas las restricciones se le conoce como factible, pero si además cuenta con el valor más favorable en la función objetivo, adquiere el nombre de solución óptima.

Diversos escritos sobre esta materia (Taha, 2004; Matousek y Gärtner, 2007; Bazaraa, Jarvis y Sherali, 2011) coinciden en que un modelo basado en un programa lineal está compuesto por un conjunto de variables de decisión, una función objetivo lineal que indica la contribución de cada variable al resultado deseado y un grupo de restricciones lineales que describen los límites en los valores de las variables.

A continuación se presenta la teoría fundamental de la programación lineal, la cual es universal y fue extraída de los escritos mencionados.

3.1.1 Ecuaciones lineales

Todas las ecuaciones y desigualdades en un programa lineal deben, por definición, ser lineales.

Una función lineal tiene la siguiente forma (1):

$$a_0 + a_1x_1 + a_2x_2 + a_3x_3 + \dots + a_nx_n = 0$$

(1) Forma de una función lineal

En general, las a son los llamados coeficientes de la ecuación; también conocidos como parámetros. Lo importante de los coeficientes es que son valores fijos, basados en la naturaleza subyacente del problema que se está resolviendo. Las x son las variables de la ecuación; se les permite tomar un rango de valores dentro de los límites definidos por las restricciones.

3.1.2 Las variables de decisión

Las variables en la programación lineal son un conjunto de valores que deben determinarse para resolver el problema; es decir, el problema se resuelve cuando se han identificado los mejores totales de las variables. Típicamente, las variables representan la cantidad de un recurso a usar o el nivel de alguna actividad.

3.1.3 La función objetivo

El objetivo de un problema de programación lineal será maximizar o minimizar algún valor numérico. La función objetivo indica cómo cada variable contribuye al valor que se optimizará para resolver el problema.

La función objetivo toma la siguiente forma general (2):

$$\text{maximizar o minimizar } Z = \sum_{i=1}^n c_i X_i$$

(2) Forma de la función objetivo

En donde c_j = coeficiente de la función objetivo correspondiente a la $j^{\text{ésima}}$ variable y X_j = $j^{\text{ésima}}$ variable de decisión.

Los coeficientes de la función objetivo indican la contribución al valor de la función objetivo de una unidad de la variable correspondiente.

3.1.4 Las restricciones

Las restricciones definen los valores posibles que pueden tomar las variables de un problema de programación lineal. Típicamente representan restricciones de recursos, o el nivel mínimo o máximo de alguna actividad o condición.

Toman la siguiente forma general (3):

$$\text{Sujeto a } \sum_{i=1}^n a_{ij} X_i \leq b_j \quad j = 1, 2, \dots, m$$

(3) Forma general de una programación lineal

en donde, X_j = $j^{\text{ésima}}$ variable de decisión

a_{ij} = al coeficiente en en la restricción j y

b_j = el coeficiente del lado derecho en la restricción j

Según Sanders y Reid (2012), para considerar a un programa lineal como tal, este debe contar con las suposiciones implícitas de proporcionalidad, aditividad, divisibilidad y certeza. Estos se explican a continuación:

- Proporcionalidad: la contribución de las variables de decisión y de las restricciones en la función objetivo es proporcional a su valor. Por ejemplo, si se duplica el valor de una variable, también se duplicará la contribución de esa variable a la función objetivo y cada restricción en la cual aparece esa variable.
- Aditividad: el valor total de la función objetivo y cada función de restricción se obtienen sumando los aportes individuales de cada variable.
- Divisibilidad: las variables de decisión pueden tomar cualquier valor numérico real dentro de algún rango especificado por las restricciones. Es decir, las variables no están restringidas a valores enteros.
- Certeza: la solución óptima obtenida para un problema específico siempre será inmejorable. No obstante, si los valores de los parámetros son erróneos, entonces, la solución resultante lo será también.

3.2 Programación por objetivos

Como se ha mostrado, el propósito de los modelos basados en la programación lineal es resolver un único objetivo. Sin embargo, a partir de la práctica industrial y financiera moderna, está comprobado que maximizar las utilidades no necesariamente es el único objetivo que se busca alcanzar.

Charnes y Stedry (1966) no encuentran evidencia de que las empresas, efectivamente, construyan funciones focalizadas en maximizar la utilidad a largo plazo; además, sugieren que “los supuestos beneficios de la maximización de la utilidad no son operativos (incluso) si, lógica o tautológicamente, se puede demostrar que son válidos”.

En las operaciones de planificación a largo plazo es mucho más habitual que los administradores hagan proyecciones que abarquen toda una gama de indicadores e intenten organizar una estrategia coherente para cumplir con estas proyecciones deseadas (Kornbluth, 1973).

Esto sería el caso habitual de la administración de las corporaciones en Estados Unidos, quienes se enfocan en una variedad de objetivos, por ejemplo, mantener ganancias estables, aumentar (o mantener) la cuota de mercado, diversificar productos, mantener precios estables, mejorar la moral del trabajador, mantener el control familiar del negocio y aumentar el prestigio de la compañía (Hillier y Lieberman, 2014). En síntesis, un proceso de toma de decisión no suele acotarse al cumplimiento de un único objetivo, de hecho, las decisiones más importantes suelen surgir en una atmósfera de confrontación en donde diferentes intereses se contraponen para llegar a una solución (Tamiz, Jones y Romero, 1998).

Los métodos de programación de objetivos amplían nuestras capacidades de modelado al ofrecer formas de tratar con más de un objetivo a la vez. La programación por objetivos se puede entender como una extensión o generalización de la programación lineal para manejar múltiples medidas objetivas, normalmente en conflicto. A cada una de estas medidas se le asigna un objetivo o valor objetivo que se debe alcanzar. Las desviaciones no deseadas de este conjunto de valores objetivo se minimizan en una función de logro. Esto puede ser un vector o una suma ponderada, dependiendo de la variante de programación de objetivos utilizada.

4. Caso de investigación

4.1 Datos de entrada

Para este trabajo, se han considerado datos de una empresa minera peruana productora de concentrados de zinc (Zn), plomo (Pb), plata (Ag) y cobre (Cu). Estos minerales los obtiene de seis minas subterráneas: San Cristóbal, Carahuacra, Andaychagua, Ticlio, Animón e Islay. Estos minerales son procesados en cuatro plantas concentradoras: Victoria, Andaychagua, Mahr Tunel y Animón.

Cada una de las minas subterráneas tiene una ley particular sobre los cuatro minerales indicados previamente. La concentración de estos minerales (ley de mineral) por tonelada es cambiante con el tiempo, debido a la geografía de la zona, la ubicación de las vetas y al avance de la operación realizada en cada mina (tabla 1).

Tabla 1
Leyes de los minerales

Minas subterráneas	Porcentaje Zn	Porcentaje Pb	Porcentaje Cu	Ag oz/TM
San Cristóbal	6,39	1,05	0,20	4,28
Carahuacra	6,28	0,50	0,09	2,50
Andaychagua	3,84	0,66	0,10	3,68
Ticlio	6,07	1,63	0,19	1,60
Animón	5,65	1,59	0,15	1,90
Islay	1,89	0,83	0,05	7,15

Elaboración propia

Dada la ubicación de las minas subterráneas y las plantas concentradoras, existe una relación mina-planta. Esto quiere decir que la extracción de minerales de una mina solo se puede tratar en la planta que tiene asignada (figura 1).

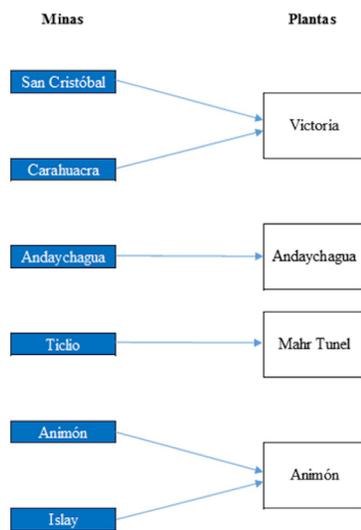


Figura 1. Relación mina-planta

Elaboración propia

Para la extracción del mineral, desde la zona de operación en interior de mina hasta la planta concentradora, se utilizan camiones con capacidad de 20 toneladas. El tipo de camiones es homogéneo en todas las unidades mineras debido a las dimensiones de los túneles de las minas. Sus camiones son de mayor capacidad para que puedan trabajar dadas las características de sus minas subterráneas.

El tiempo de viaje, desde la zona de carguío en interior de mina hasta las plantas concentradoras correspondientes, es variable y depende de la profundidad de la zona de operación subterránea, de las vías de transporte hacia superficie y de las distancias de las bocaminas hacia las plantas concentradoras. Los tiempos promedio, aproximados, de ida y vuelta de un camión desde la mina hasta la planta son los expresados en la tabla 2.

Tabla 2
Tiempos de viaje mina-planta promedio

Minas subterráneas	Planta concentradora	Tiempo viaje promedio (h)
San Cristóbal	Victoria	2,50
Carahuacra		1,50
Andaychagua	Andaychagua	1,50
Ticlio	Mahr Tunel	3,00
Animón	Animón	1,50
Islay		2,50

Elaboración propia

Independientemente del tipo de mina, estas deben cumplir con un tonelaje de extracción mínimo diario, básicamente, para cubrir los costos operativos esenciales de cada mina (bombeo, ventilación y energía eléctrica). Adicionalmente, la empresa debe cumplir con producción objetivo mínima por cada mineral para poder cumplir los acuerdos contractuales pactados con sus clientes.

Cada mina subterránea tiene costos de extracción individuales debido a las características propias de cada una. Entre las variables que afectan estos costos se tienen: la topografía de la mina, el grado de sostenibilidad, el costo de los servicios auxiliares (bombeo, ventilación y energía eléctrica), la perforación, los explosivos, el equipamiento minero y su respectivo mantenimiento, la conservación de vías, el acarreo, etc.

Estos costos unitarios se obtienen sumando todos los costos que se tienen en la extracción del mineral, dividiéndolo por el número de toneladas extraídas en un determinado período de tiempo. En cuanto a las plantas concentradoras, estas son plantas de procesamiento de mineral donde se trata lo extraído y se busca separar el material inerte, elevando la concentración de los minerales a tratar (zinc, plomo, plata y cobre). Debido a la infraestructura de estas plantas, hay una capacidad máxima de tratamiento de mineral por cada una de ellas, la cual se refleja en la tabla 3.

Tabla 3
Capacidad máxima de la planta

Planta concentradora	Capacidad máxima planta (TM)
Victoria	5200
Andaychagua	3450
Mahr Tunel	2750
Animón	5500

Elaboración propia

De manera similar a lo que sucedía con las minas, cada planta concentradora tiene costos de tratamiento individuales debido a las características propias de cada una. Entre las variables que afectan estos costos se tienen: chancado, molienda, reactivos, flotación, filtrado, energía eléctrica, tratamiento de relave, relaveras, etc. Estos costos unitarios se obtienen sumando todos los costos que se tienen para el tratamiento del mineral, dividiéndolo por el número de toneladas tratadas en un determinado período de tiempo. Como se podrá apreciar, es la capacidad de tratamiento de las plantas la que determina los límites de extracción del mineral de cada una de las minas.

4.2 Modelo matemático

El modelo planteado tiene como objetivo minimizar un puntaje basado en las desviaciones de tres objetivos. Los objetivos están relacionados al uso de la capacidad de las plantas, a la producción de los minerales y al costo total de la operación. A cada objetivo se le es asignado un peso (este dependiendo de su importancia relativa). Luego, este valor es multiplicado por la desviación del resultado del objetivo planteado. Mientras menor sea la desviación, menor será el puntaje, lo cual significa una mayor precisión en el cumplimiento de los objetivos.

a) índices

i = mina $i = 1, 2, \dots, x$

j = mineral $j = 1, 2, \dots, y$

k = planta concentradora $k = 1, 2, \dots, z$

en donde:

x = número de minas

y = número de minerales

z = número de plantas concentradoras

b) Variable de decisión

$cant_i$ = cantidad de camiones asignados a la mina i

c) Parámetros

ley_{ij} = ley del mineral j en la mina i

$precio_j$ = precio del mineral j en el mercado internacional

$carg$ = capacidad de transporte de un camión

t_i = tiempo promedio de viaje desde la mina i hasta su respectiva planta

ctr_k = costo de tratamiento (US\$/TM) de la planta k

$cext_i$ = costo de extracción (US\$/TM) de la mina i

$capMAX_k$ = capacidad máxima de tratamiento de la planta k

$costMAX$ = presupuesto límite establecido

$prodMin_j$ = producción mínima mineral j

rln_{ik} = relación de paridad entre la mina i y la planta k

$\begin{cases} 1: \text{en caso el mineral extraído de mina } i \text{ se trabaje en la planta } k \\ 0: \text{caso contrario} \end{cases}$

$Wght_k$ = peso de penalización relacionado a la capacidad de utilización objetivo de la planta k

$Wght_j$ = peso de penalización relacionado a la cantidad de concentrado objetivo del mineral j

$WghtCost$ = peso de penalización relacionado al costo total objetivo

$DesvCap_k$ = desviación de la capacidad de utilización objetivo de la planta k

$DesvMin_j$ = desviación de la cantidad de concentrado objetivo del mineral j

$DesvCost$ = desviación del costo total objetivo

d) Función objetivo

La función objetivo se muestra a continuación:

$$PT = (DesvCap_k \times Wght_k) + (DesvMin_j \times Wght_j) + (DesvCost \times WghtCost)$$

(4) Función objetivo propuesta

e) Restricciones

$$ObjCap_k - \sum_{i=1}^x rlc_{ik} \left(carg \times cant_i \frac{24}{t_i} \right) = DesvCap_k \quad k = 1, 2, \dots, z$$

(5) Restricción 1

Para determinar la desviación de la capacidad de utilización de una planta, primero, se debe precisar cuánto es la cantidad de extracción que será trasladada hacia esta planta. Ese monto se calcula multiplicando la carga de un camión por la cantidad de camiones asignados a las minas que depositan en dicha planta. Hallado el monto, este es restado del objetivo, dando lugar a la desviación.

$$ObjMin_j - \sum_{i=1}^x ley_{ij} \left(carg \times cant_i \frac{24}{t_i} \right) = DesvMin_j \quad j = 1, 2, \dots,$$

(6) Restricción 2

La desviación de la cantidad de concentrado objetivo es hallada restando el monto concentrado total del objetivo. Para calcular el total de concentrado adquirido de un mineral, basta con hacer una sumatoria de la multiplicación de la ley de ese mineral por la cantidad de extracción en cada mina.

$$ObjCost - \sum_{i=1}^x \sum_{j=1}^k rlc_{ik} \left(carg \times cant_i \frac{24}{t_i} \right) (ctr_k + cext_i) = DesvCost$$

(7) Restricción 3

El costo total de la operación se halla sumando los costos de extracción y los de tratamiento. El costo de extracción varía de mina a mina y el costo de tratamiento varía dependiendo de la planta. El costo total es directamente proporcional a la cantidad de extracción de cada mina, mientras mayor sea la extracción mayor será el costo total. De manera similar a las operaciones en la ecuaciones (6) y (7), para hallar la desviación del costo total objetivo, al costo total objetivo se le resta el costo total de la operación.

$$\sum_{i=1}^x rlc_{ik} \left(carg \times cant_i \frac{24}{t_i} \right) \leq capMAX_k \quad k = 1, 2, \dots, z$$

(8) Restricción 4

La cantidad de extracción que se dirige a cada planta debe de ser menor o igual a la capacidad máxima de tratamiento de dicha planta.

$$\sum_{i=1}^x ley_{ij} \left(carg \times cant_i \frac{24}{t_i} \right) \geq prodMin_j \quad j = 1, 2, \dots, y$$

(9) Restricción 5

La cantidad de extracción de cada mineral debe de ser mayor o igual a la producción mínima establecida.

$$\sum_{i=1}^x ley_{ij} \left(carg \times cant_i \frac{24}{t_i} \right) \geq prodMin_j \quad j = 1, 2, \dots, y$$

(10) Restricción 6

$$x_i \geq 0, \quad x_i \in \mathbb{N}$$

(11) Restricción 7

La cantidad de camiones a asignar a cada mina debe ser un número natural mayor o igual a 0.

4.3 Experimentación y hallazgos

Para la muestra de la experimentación, se utilizarán seis casos de prueba que representan diferentes situaciones a las que una empresa minera podría enfrentarse. Cabe mencionar que, para facilitar el análisis y la interpretación de los resultados, se mantendrá una misma distribución de pesos por cada objetivo y en cada una de las ejecuciones. Los casos de prueba en sí están dirigidos a probar escenarios que van más allá de los límites de la operación normal, esto con el fin de medir la solidez del sistema y el manejo de errores.

La construcción de los casos de prueba fue hecha con la ayuda de un grupo de usuarios finales, de la misma minera, que proporcionó los datos de entrada. Tal y como sugiere la literatura, esto es bastante beneficioso, porque puede aprovecharse el dominio y la experiencia de estas personas para establecer escenarios más acordes a la realidad. Los casos de prueba variarán según los valores que se le asigna a cada objetivo, manteniendo los datos paramétricos constantes.

Cabe mencionar que los pesos establecidos para cada objetivo fueron determinados según un análisis exhaustivo hecho por los usuarios finales. El análisis se basó en data histórica de la empresa y en sus futuros planes y convicciones. Considerando lo anterior, el método empleado para la experimentación será una mezcla entre el uso de Test Bases y el de Real-time and Historic Data Sets. Estos métodos conformarán el proceso de validación. Asimismo, para ejecutar el modelo se utilizó un complemento de Microsoft Excel llamado Solver. El motivo fue, básicamente, por la flexibilidad durante la diagramación del modelo en sí que ofrece esta herramienta.

Para el primer caso de prueba, consideraremos una situación en donde la combinación de objetivos son valores pequeños, pero con un presupuesto amplio (tabla 4).

Tabla 4
Objetivos del caso de prueba 1

Variable	Objetivos	Peso por 1 % de desviación	1 % del objetivo
VIC-	5200	3	52
AND-	3450	3	34,5
MAH-	2750	3	27,5
ANI-	5500	3	55
ZN+	500	10	5
PB+	100	10	1
CU+	15	10	0,15
Ag+	10 000	10	100
DOL+	2 000 000	-5	20 000

Elaboración propia

Al ejecutar el sistema obtenemos los resultados dados en la tabla 5.

Tabla 5
Resultados del caso de prueba 1

Variable	Objetivos	Desviación	Puntaje	Minas subterráneas	Mejor arreglo de camiones
VIC-	2304,00	2896,00	167,08	San Cristóbal	12
AND-	2560,00	890,00	77,39	Carahuacra	0
MAH-	2720,00	30,00	3,27	Andaychagua	8
ANI-	2304,00	3196,00	174,33	Ticlio	17
ZN+	502,71	2,71	5,41	Animón	0
PB+	111,42	11,42	114,18	Islay	12
CU+	15,06	0,06	3,95	Total	49
Ag+	10 044,27	44,27	4,43		
DOL+	1 212 480,00	787 520,00	-196,88		
Puntaje			353,15		

Elaboración propia

Como se ve en la tabla 5, el sistema no utiliza las minas Carahuacra y Animón para cumplir con los objetivos planteados. Con esto evidenciamos que el modelo es capaz de determinar cuándo es o no necesario extraer de una mina. Si bien existen programaciones que cumplirían con los objetivos utilizando todas las minas, el porqué el sistema decide no utilizarlo es debido a la menor desviación de los objetivos que tiene utilizando esta programación.

Para el segundo caso de prueba, considerarnos los mismos objetivos que el caso anterior, pero ahora acotaremos el presupuesto que se tiene a \$950 000 (tabla 6).

Tabla 6
Objetivos del caso de prueba 2

Variable	Objetivos	Peso por 1 % de desviación	1 % del objetivo
VIC-	5200	3	52
AND-	3450	3	34,5
MAH-	2750	3	27,5
ANI-	5500	3	55
ZN+	500	10	5

(continúa)

(continuación)

PB+	100	10	1
CU+	15	10	0,15
Ag+	10 000	10	100
DOL+	950 000	-5	9500

Elaboración propia

Al ejecutar el sistema obtenemos los resultados de la tabla 7.

Tabla 7
Resultados del caso de prueba 2

Variable	Objetivos	Desviación	Puntaje		
VIC-	3264,00	1936,00	111,69	Minas subterráneas	Mejor arreglo de camiones
AND-	0,00	3450,00	300,00	San Cristóbal	7
MAH-	2720,00	30,00	3,27	Carahuacra	6
ANI-	2112,00	3388,00	184,80	Andaychagua	0
ZN+	507,80	7,80	15,60	Ticlio	17
PB+	102,80	2,80	27,97	Animón	6
CU+	15,06	0,06	3,95	Islay	1
Ag+	10 219,97	219,97	22,00	Total	37
DOL+	911 360,00	38 640,00	-20,34		
Puntaje			648,94		

Elaboración propia

Nótese que el sistema efectivamente puede abarcar la misma producción objetivo a un menor costo y al mismo tiempo con una significativa menor de cantidad de camiones. Sin embargo, si comparamos el puntaje en ambos casos, 353,15 y 648,94, el puntaje del segundo caso de prueba está cerca a ser el doble que el primero. Para el sistema esto significa que (aunque parezca contradictorio en efectos de lo que verdaderamente le convendría a la empresa), en el primer caso, se están cumpliendo de manera más adecuada los objetivos. Esto es causado porque las desviaciones de todas las variables son mucho más grandes en el segundo caso, puesto que esta es la única forma de cumplir con el presupuesto asignado.

Para el tercer y cuarto caso, se simularán contextos en los cuales, debido a eventos como el alza en el precio de algún mineral, se querrá extraer la mayor cantidad posible. Para esto se

comenzará con una producción objetivo de plata de 22 400 oz que, considerando las leyes y las capacidades productivas impuestas en el modelo, es un valor bastante alto (tabla 8).

Tabla 8
Objetivos del caso de prueba 3

Variable	Objetivos	Peso por 1 % de desviación	1 % del objetivo
VIC-	5200	3	52
AND-	3450	3	34,5
MAH-	2750	3	27,5
ANI-	5500	3	55
ZN+	500	10	5
PB+	100	10	1
CU+	15	10	0,15
Ag+	22 400	10	224
DOL+	2 000 000	-5	20 000

Elaboración propia

Al ejecutar el sistema obtenemos los siguientes resultados de la tabla 9.

Tabla 9
Resultados del caso de prueba 3

Variable	Objetivos	Desviación	Puntaje	Minas subterráneas	Mejor arreglo de camiones
VIC-	5184,00	16,00	0,92	San Cristóbal	27
AND-	2560,00	890,00	77,39	Carahuacra	0
MAH-	0,00	2750,00	300,00	Andaychagua	8
ANI-	5376,00	124,00	6,76	Ticlio	0
ZN+	536,17	36,17	72,35	Animón	0
PB+	123,40	23,40	233,98	Islay	28
CU+	16,15	1,15	76,48	Total	63
Ag+	22 408,37	8,37	0,37		
DOL+	1 585 600,00	414 400,00	-103,60		
Puntaje			664,66		

Elaboración propia

Los resultados están relacionados con los objetivos planteados. Fíjese que para el caso de prueba 3, el sistema solo usa 3 de las 6 minas: San Cristóbal, Andaychagua e Islay. Esto tiene bastante sentido, considerando que estas minas son las que tienen la mayor ley de plata, siendo 4,23, 3,17 y 7,42 oz/TM, respectivamente. Cabe resaltar el desuso de las otras minas. El sistema opta por no usarlas, ya que de lo contrario se podría incurrir en un sobrecumplimiento de las producciones objetivo, tanto de la plata como de los otros minerales, cosa que penaliza el puntaje.

Ahora obtendremos una producción de zinc. Al igual que en el caso anterior, probaremos con una producción objetivo alta, considerando las leyes y las capacidades productivas impuestas en el modelo, esta será de 998 TM (tabla 10).

Tabla 10
Objetivos del caso de prueba 4

Variable	Objetivos	Peso por 1 % de desviación	1 % del objetivo
VIC-	5200	3	52
AND-	3450	3	34,5
MAH-	2750	3	27,5
ANI-	5500	3	55
ZN+	998	10	9,98
PB+	100	10	1
CU+	15	10	0,15
Ag+	10 000	10	100
DOL+	2 000 000	-5	20 000

Elaboración propia

Al ejecutar el sistema obtenemos los resultados de la tabla 11.

Tabla 11
Resultados del caso de prueba 4

Variable	Objetivos	Desviación	Puntaje	Minas subterráneas	Mejor arreglo de camiones
VIC-	5184,00	16,00	0,92	San Cristóbal	27
AND-	3200,00	250,00	21,74	Carahuacra	0
MAH-	2720,00	30,00	3,27	Andaychagua	10
ANI-	5440,00	60,00	3,27	Tidío	17

(continúa)

(continuación)

ZN+	998,24	0,24	0,24	Animón	17
PB+	225,09	125,09	1250,88	Islay	0
CU+	29,62	14,62	974,61	Total	71
Ag+	22 186,98	12 186,98	1218,70		
DOL+	1 938 240,00	61 760,00	-15,44		
Puntaje			3458,20		

Elaboración propia

Como se puede apreciar en la tabla 11, el objetivo es completado satisfactoriamente. Sin embargo, comparando estos resultados con los del caso anterior, vemos que el costo al extraer una gran cantidad de zinc es mucho más alto que al extraer una gran cantidad de plata. Por otro lado, con respecto a la asignación de camiones, las minas en donde se extrae son, además, las más ricas en zinc: San Cristóbal con una ley de 6,45 %, Ticlio con 7,44 %, Animón con 5,86 % y Andaychagua con 4,46 %. A simple vista parecería que hubiera un error, pues se le ha asignado menos camiones a Ticlio que a San Cristóbal, a pesar de tener una ley más grande de zinc. Esto no es un error; es causado por la restricción de la planta concentradora Mahr Tunel que, con una capacidad de tratamiento máximo de 2750 TM diaria, limita qué tanto se puede extraer de la mina Ticlio. La planta concentradora Victoria, con una capacidad de tratamiento máximo de 5200 TM diaria, tiene más potencial para abarcar grandes cantidades de extracción.

Asimismo, cabe señalar otro punto valioso, el porqué de la asignación de 0 camiones a la mina Carahuacra a pesar de tener una ley 5,37 % de zinc, siendo inclusive mayor a la de Andaychagua. Esto se debe a que la extracción de la mina Carahuacra es enviada a la planta Victoria para su tratamiento, igual que la de la mina San Cristóbal. Considerando esto, el sistema indudablemente priorizará ocupar la totalidad de la planta solo con San Cristóbal, pues tiene una ley mayor a la de Carahuacra: 6,45 % y 5,37 %, respectivamente. Para los dos últimos casos, se experimentará una programación ideal, en otras palabras, es donde la producción objetivo busca los niveles más altos de cada uno de los minerales al menor costo posible. Comenzaremos con un caso imposible (tabla 12).

Tabla 12
Objetivos del caso de prueba 5

Variable	Objetivos	Peso por 1 % de desviación	1 % del objetivo
VIC-	5200	3	52
AND-	3450	3	34,5
MAH-	2750	3	27,5
ANI-	5500	3	55
ZN+	895	10	8,95
PB+	195	10	1,95
CU+	24	10	0,2
Ag+	18 600	10	186
DOL+	2 000 000	-5	20 000

Elaboración propia

Al ejecutar el sistema, nos muestra el error de la figura 2.

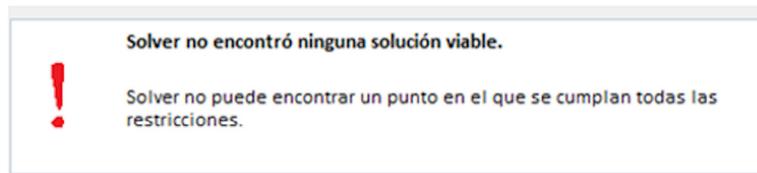


Figura 2. Error en el caso de prueba 5

Esto se origina porque las producciones objetivo que introducimos son excesivamente altas y, considerando las restricciones de la capacidad máxima de las plantas, no existe solución que permita cumplir con tales objetivos.

Ahora se considerarán objetivos posibles con altos objetivos de producción (tabla 13).

Tabla 13
Objetivos del caso de prueba 6

Variable	Objetivos	Peso por 1 % de desviación	1 % del objetivo
VIC-	5200	3	52
AND-	3450	3	34,5
MAH-	2750	3	27,5
ANI-	5500	3	55
ZN+	1000	10	10
PB+	225	10	2,25
CU+	29	10	0,29
Ag+	22 180	10	221,8
DOL+	2 000 000	-5	20 000

Elaboración propia

Al ejecutar el sistema obtenemos los resultados de la tabla 14.

Tabla 14
Objetivos del caso de prueba 6

Variable	Objetivos	Desviación	Puntaje	Minas subterráneas	Mejor arreglo de camiones
VIC-	5184,00	16,00	0,92	San Cristóbal	17
AND-	3200,00	250,00	21,74	Carahuacra	6
MAH-	2720,00	30,00	3,27	Andaychagua	10
ANI-	5440,00	60,00	3,27	Ticlio	17
ZN+	896,29	1,29	1,44	Animón	11
PB+	195,71	0,71	3,65	Islay	10
CU+	24,82	0,82	34,13	Total	71
Ag+	18 603,87	3,87	0,21		
DOL+	1 880 640,00	119 360,00	-29,84		
Puntaje			38,80		

Elaboración propia

Los resultados mostrados en la tabla 14, entonces, representarían el mejor escenario para la empresa.

4.4 Herramienta computacional utilizada

Basado en el modelo matemático propuesto y con los datos proporcionados por la empresa minera, se desarrolló un prototipo de lo que sería la solución final de este trabajo. Para ello, se utilizó el complemento Solver de Microsoft Excel. El motivo por el cual se optó por utilizar esta herramienta sobre otras, como Lingo o Storm, fue básicamente por su flexibilidad durante la diagramación del modelo en sí. Asimismo, teniendo en cuenta que es en realidad el Excel el que invoca a este complemento, el uso de Solver se empodera por las características que trae el Excel como tal, como la capacidad de incluir gráficos o botones programables a través del uso de macros.

A continuación, la definición del modelo matemático propuesto realizado en Solver (figura 3).

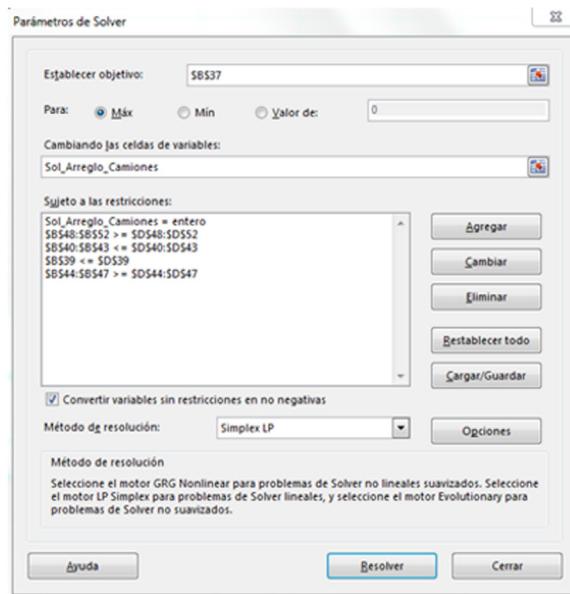


Figura 3. Modelo matemático definido en Solver

Elaboración propia

Como se puede apreciar en la figura 3, tanto las variables de decisión, las restricciones y la función objetivo hacen referencia a celdas de Excel. Esto se debe a que los datos que se ingresaron para tales partes del modelo están definidos en el mismo Excel. Cabe mencionar que si bien en la figura 17 no se encuentran todas las ecuaciones lineales definidas previamente, esto es porque han sido introducidas indirectamente a través de fórmulas entre diferentes celdas de datos.

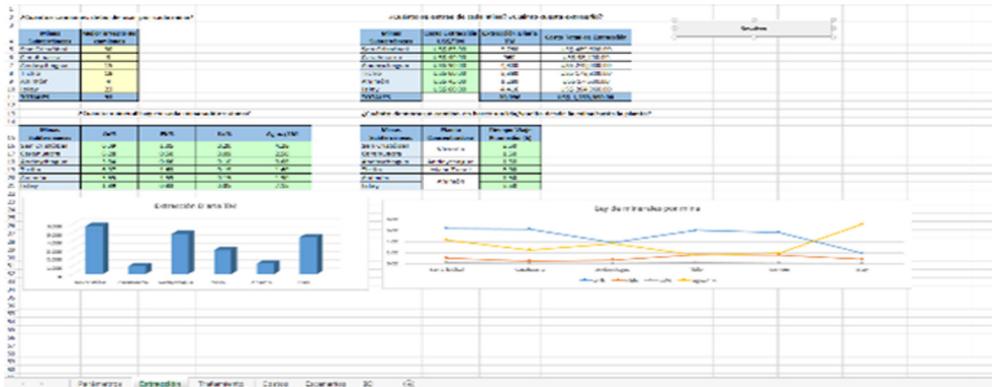


Figura 4. Vista previa del prototipo en Excel

Elaboración propia

El prototipo cuenta con cinco pestañas diferentes: “parámetros”, básicamente donde se introducen todos los valores que funcionan como restricciones para el modelo; “extracción”, pestaña en donde se da inicio a la ejecución del modelo, a través de Solver, y en donde guarda la solución. Además, almacena todo lo referente a la extracción como son las leyes de minerales, los tiempos de viaje promedio y los costos de extracción.

Por otra parte, la pestaña “tratamiento” guarda toda la data relacionada al proceso de tratamiento del mineral como son los costos de tratamiento por planta y la producción de concentrado por mineral de cada una. La de “costos” presenta los valores para hallar la utilidad total: total de ventas, costo de extracción y costo de tratamiento. Por último, las pestañas “escenarios” y “BD” son módulos adicionales que buscan simular las características que ya tendría el SSD, como es la comparación de escenarios y el almacenamiento periódico de las programaciones de camiones en una base de datos.

5. Reflexión

Podemos afirmar que el modelo funciona correctamente. Se logró la construcción de un modelo basado en la programación por objetivos que calcula el tamaño de la flota de camiones óptima y su adecuada asignación a las diferentes minas, con miras a minimizar las desviaciones de la utilización de la capacidad máxima de las plantas concentradoras y la producción de los minerales, al menor costo posible.

El uso de la programación por objetivos y la programación lineal son técnicas de investigación de operaciones bastante útiles y, tal y como se comprobó en otros estudios, particularmente, en cuanto a la asignación de camiones en el sector minero. Considerando la interpretación de los resultados, es indudablemente recomendable que la empresa minera utilice este modelo siempre

que busque establecer objetivos relacionados a la producción de solo minerales, que, como grupo, tienen las leyes más altas. Esto debido a que podrían explotar de mejor manera el uso de sus instalaciones, generar mayores concentrados y, al mismo tiempo, minimizar el costo total.

6. Futura continuación de la investigación

Para darle continuidad a esta investigación, se recomienda investigar la inclusión de otras variables, tales como el uso de combustible, la cantidad de tiempo de transporte y la determinación de rutas entre minas y plantas. Tales factores, si bien no participan en el cumplimiento de los tres objetivos planteados para esta investigación, a través de una correcta medición y control podrían traer una mayor reducción en costos de la que el modelo planteado ofrece. Asimismo, añadir un nuevo conjunto de variables abriría paso a la inclusión de nuevos objetivos, por ejemplo, que la suma total de combustible de todos los camiones a utilizar no supere los X litros.

Referencias

- Alvarado Boirivant, J. (2009). La programación lineal aplicación de las pequeñas y medianas empresas. *Reflexiones*, 88(1), pp. 89-105.
- Bajany, D. M., Xia, X., y Zhang, L. (2017). A MILP Model for Truck-shovel scheduling to minimize fuel consumption. *Energy Procedia*, 105, pp. 2739-2745.
- Bazaraa, M. S., Jarvis, J. J., y Sherali, H. D. (2011). *Linear programming and network flows*. Hoboken, Estados Unidos: John Wiley & Sons.
- Burt, C., Caccetta, L., Hill, S., Welgama, P., Zerger, A., y Argent, R. M. (2005). Models for mining equipment selection. *International Congress Modeling Simulation*. Modelling and Simulation Society of Australia and New Zealand, pp. 1730-1736.
- Charnes, A., y Stredy, A. (1966). *The attainment of organizational goals through the appropriate selection of sub-unit goals*. Operational Research and the Social Sciences. Tavistock Publications.
- Crovo, A. S., Martín, C. O. y Rojas, L. P. (2007). Modelos de programación entera para un problema de programación de horarios para universidades. *Revista Chilena de Ingeniería*.
- Derigs, U. (2009). *Optimization and operations research* (volume I). Oxford, Inglaterra: EOLSS Publications.
- Elbrond, J., y Soumis, F. (1987). Towards integrated production planning and truck dispatching in open pit mines. *International Journal of Surface Mining*, pp. 1-6.
- Hillier, F., y Lieberman, G. (2014). *Introduction to operations research* (10th ed.). New York: McGraw Hill Education.

- Kornbluth, J. S. H. (1973). A survey of goal programming. *Omega*, 1(2), pp. 193-205.
- Lawrence, O. (2017). Linear programming as a tool for water resources management. *International Journal of Constructive Research in Civil Engineering*, 3(4), pp. 30-47.
- LCG Energy Management Group. (2009). *Investigation of current research related to the reduction of energy usage in mines through recycling, reuse, and other means*.
- Matousek, J., y Gärtner, B. (2007). *Understanding and using linear programming*. Berlín: Springer Science & Business Media.
- McDill, M. E. (1999). *Forest Resource Management*. Recuperado de https://www.courses.psu.edu/for/for466w_mem14/PDFs/
- Nehring, M., Topal, E., y Knights, P. (2010). Dynamic short term production scheduling and machine allocation in underground mining using mathematical programming. *Mining Technology*, 119(4), pp. 212-220.
- Sahoo, L. K., Bandyopadhyay, S., y Banerjee, R. (2014). Benchmarking energy consumption for dump trucks in mines. *Applied Energy*, 113, pp. 1382-1396.
- Ta, C. H., Ingolfsson, A., y Doucette, J. (2013). A linear model for surface mining haul truck allocation incorporating shovel idle probabilities. *European Journal of Operational Research*, 231(3), pp. 770-778.
- Ta, C. H., Kresta, J. V., Forbes, J. F., y Marquez, H. J. (2005). A stochastic optimization approach to mine truck allocation. *International Journal of Mining, Reclamation and Environment*, 19(3), pp. 162-175.
- Taha, H. A. (2004). *Investigación de operaciones*. México: Pearson Educación.
- Tamiz, M., Jones, D., y Romero, C. (1998). Goal programming for decision making: An overview of the current state-of-the-art. *European Journal of Operational Research*, 111(3), pp. 569-581.
- Temeng, V. A., Francis, O. O., y Frendewey, Jr., J. O. (1997). Real-time truck dispatching using a transportation algorithm. *International Journal of Surface Mining, Reclamation and Environment*. 11(4), pp. 203-207.
- Upadhyay, S. P., y Askari-Nasab, H. (2016). Truck-shovel allocation optimisation: a goal programming approach. *Mining Technology*, 125(2), pp. 82-92.
- Zhang, L., y Xia, X. (2015). An integer programming approach for truck-shovel dispatching problem in open-pit mines. *Energy Procedia*, 75, pp. 1779-1784.

PERFILES

**Diana
Maribel
Garcés-Eslava**

Correo electrónico: 20120535@aloe.ulima.edu.pe

Egresada de la Carrera de Ingeniería de Sistemas de la Universidad de Lima. Labora en el área de Sistemas-Producción como analista de producción *junior* en la Asociación Cultural Peruano Británica. Sus áreas de interés son técnicas de minería de datos, métodos de procesamiento de lenguaje natural y aprendizaje automático.

Correo electrónico: gonzalorehu.97.31@gmail.com

Graduado por la Carrera de Ingeniería de Sistemas de la Universidad de Lima. Se inició como desarrollador de aplicaciones móviles para dispositivos iOS en el Laboratorio de Aprendizaje en Tecnologías de Información (ITLab), de la misma institución, donde desarrolló parte de la aplicación Ulima App que logró 20 000 usuarios diarios activos. Actualmente es ingeniero iOS en Scotiabank Digital Factory, donde desarrolla la aplicación principal de Scotiabank Perú y frameworks internos. Sus áreas de interés son el desarrollo de aplicaciones iOS, arquitectura de *software* y algoritmos y estructuras de datos.

**Gonzalo
Reyes-Huertas**

Correo electrónico: 20141925@aloe.ulima.edu.pe

**Edwin
Lara-Lévano**

Egresado de la Carrera de Ingeniería de Sistemas de la Universidad de Lima. Ha sido desarrollador de aplicaciones móviles para dispositivos Android en el Laboratorio de Aprendizaje en Tecnologías de Información (ITLab), de la misma institución, donde formó parte del equipo de desarrollo de la aplicación Ulima App y de la aplicación CorrePalabras. Actualmente se desempeña como desarrollador de aplicaciones Android semi *senior* para la agencia digital Perú Apps con el proyecto Caja Arequipa Móvil y forma parte del Laboratorio de Innovación de Caja Arequipa (CajaLab). Sus áreas de interés son el desarrollo de aplicaciones Android, computación visual y análisis de imágenes.

Correo electrónico: 20142424@aloe.ulima.edu.pe

Egresado de la Carrera de Ingeniería de Sistemas de la Universidad de Lima. Ha trabajado en Cementos Pacasmayo en el área de Inteligencia Comercial y actualmente labora en el área Digital de RIMAC Seguros. Sus áreas de interés son la inteligencia artificial y el análisis predictivo.

**Sergio
Espinoza-Montalvo**

Correo electrónico: edmundoqueiroz@belcorp.biz

**Edmundo
Quiroz-Silva**

Egresado de la Carrera de Ingeniería de Sistemas de la Universidad de Lima. Ha laborado en empresas multinacionales como Claro y Volcan. Actualmente trabaja en Belcorp como analista de datos. Sus áreas de interés son *customer analysis* y *social media analytics*.



UNIVERSIDAD
DE LIMA

Método de procesamiento
de lenguaje natural y técnicas
de minería
de datos aplicadas a la
clasificación de incidentes
informáticos

Predicción de deserción laboral
utilizando algoritmos genéticos
y redes neuronales artificiales

Predicción de postulantes
que cometerán fraude interno
en una compañía
con algoritmos de aprendizaje
supervisado

Sistema de reconocimiento de
gestos faciales captados a través
de cámaras para analizar el nivel
de satisfacción de clientes
en restaurantes

Programación por objetivos
para el dimensionamiento
y la asignación
de una flota de camiones
en una empresa minera