Estudio y análisis de efectividad a través de JIRA

Universidad Internacional de Valencia

Titulación:

Master U. en Big Data y Ciencia de Datos

Curso académico

2021 – 2022

Alumno/a:

Crhistian Orduz Cifuentes
D.N.I: 1072702709

Directora de TFM:

Cristina Caro González

Convocatoria:

Primera

De:
Planeta Formación y Universidades

Índice

Índice de ilustraciones

Índice de tablas

Resumen

En el presente Trabajo de Fin de Master (TFM) se realiza un estudio para evaluar el desempeño laboral de los empleados de una empresa del sector tecnológico, a través del análisis de los datos asociados a la gestión realizada durante el último año (2022) con el fin de valorar objetivamente el desempeño individual de cada empleado y optimizar la toma decisiones de la empresa con respecto al personal para mejorar la productividad y maximizar los resultados esperados.

Por lo cual el análisis fue abordado con un enfoque *data driven*, es decir aplicando juicios y valoraciones basados en datos, en el que a través de la metodología *KDD (Knowledge Discovery in Databases)* se generó un modelo de *machine learning* que categorizo a los empleados según los registros asociados a la gestión individual correspondiente al año 2022. El modelo construido puede ser iterado posteriormente de acuerdo a la dinámica de la operación de la empresa y a los lineamientos establecidos por el área de Talento Humano para el proceso de evaluación de desempeño.

Los datos utilizados en el análisis fueron tomados de *Jira*, la cual es una plataforma en la que los empleados realizan el reporte de sus actividades laborales y que permite visualizar el progreso y el estado actual de los proyectos, al tiempo que proporciona una visión general de las actividades realizadas por los miembros de la empresa. Desde *Jira* fue posible recopilar todos los registros de las actividades realizadas por los empleados de cada una de las sucursales de la empresa y exportarlos en el formato requerido para poder trabajar los datos adecuadamente.

Para llevar a cabo el estudio de desempeño fue necesario implementar principalmente los conocimientos y técnicas vistos en asignaturas del master como Minería de datos, Estadística avanzada, *Machine Learning*, Visualización de datos, Ciencia de datos para la toma de decisiones y Metodologías para la gestión de proyectos de *Big Data*.

El resultado del análisis fue la generación de conocimiento que aporto valor a la empresa al permitir entender cuáles fueron las variables que tuvieron una incidencia directa en el desempeño de los empleados y poder segmentar al personal en diferentes grupos según su rendimiento, lo que facilita que se puedan implementar estrategias y tácticas que fortalezcan las habilidades de los empleados que así lo requieran, así como reconocer el trabajo de empleados excepcionales y se logre reducir la brecha de rendimiento existente entre los empleados para mejorar la productividad de la empresa.

Este proyecto se basa en un trabajo previo del tutor, quien se encargó de otorgar el acceso al repositorio donde se encontraban los datos. Adicionalmente el presente proyecto da cumplimiento al requisito de la asignatura 14MBID como Trabajo de Fin de Master (TFM) del master de Big Data y Ciencia de Datos.

Palabras Clave: Evaluación de desempeño, Selección de variables, Minería de datos, Análisis descriptivo, Modelado de datos, Visualización

Abstract

In this Final Master's Project (TFM) a study is carried out to evaluate the job performance of the employees of a company in the technology sector, through the analysis of the data associated with the management carried out during the last year (2022) with in order to objectively assess the individual performance of each employee and improve the company's decision-making with respect to personnel to optimize productivity and maximize expected results.

Therefore, the analysis was approached with a data-driven approach, that is, applying judgments and assessments based on data, in which through the *KDD (Knowledge Discovery in Databases)* methodology a machine learning model was generated that categorized employees according to the records associated with individual management for the twelve months of 2022. The built model can be iterated later according to the dynamics of the company's operation and the guidelines established by the Human Talent area for the performance evaluation process.

The data used in the analysis were taken from Jira, which is a platform in which employees report their work activities and that allows to visualize the progress and current status of projects, while providing an overview of the activities carried out by the members of the company. From Jira it was possible to collect all the records of the activities carried out by the employees of each of the branches of the company and to export them in the format required to be able to work the data properly.To carry out the performance study it was necessary to implement mainly the knowledge and techniques seen in subjects of the master such as data mining, advanced statistics, machine learning, data visualization, Data Science for Decision Making and Methodologies for Big Data Project Management.

The result of the analysis was the generation of knowledge that brings value to the company by allowing us to understand the variables that had a direct impact on the performance of employees and be able to segment staff in different groups according to their performance, which makes it easier to implement strategies and tactics that strengthen the skills of employees who require it, as well as recognizing the work of exceptional employees and reducing the performance gap between employees to improve the productivity of the company.

This project is based on previous work by the tutor, who was in charge of granting access to the repository where the data was located. Additionally, this project fulfills the requirement of the subject 14MBID as a Final Master's Project (TFM) of the Master of Big Data and Data Science.

Keywords: Performance evaluation, Variable Selection, Data mining, Descriptive analysis, Data Modeling, Visualization.

1. Introducción

En la actualidad es fundamental para las empresas monitorear sus operaciones y llevar registros de sus actividades con el objetivo de conocer el estado de sus procesos e identificar oportunidades de mejora, para la cual deben recolectar un gran volumen de datos. (Vergara, 2019).

Según Granados (2021), gran parte de los datos recolectados son extraídos, transformados y cargados por medio de procesos *ETL* en los diferentes repositorios de las empresas para el desarrollo de herramientas de inteligencia de negocio (*BI*) y herramientas analíticas (*BA*) que les permiten optimizar la toma de decisiones estratégicas y tácticas en los diferentes frentes de la empresa.

Parte de los datos recolectados son los reportes de gestión de los empleados, quienes a través de los resultados producidos afectan directamente el cumplimiento de las metas y objetivos organizacionales. Estos datos son transformados en indicadores de gestión (*KPI´s*) que permiten analizar el comportamiento de un recurso, proceso o proyecto, con el fin de identificar de manera oportuna las desviaciones que se presentan e implementar las acciones correctivas correspondientes (Diez, 2012).

Por lo cual es necesario que se realice en las organizaciones un estudio que permita evaluar periódicamente el desempeño de los empleados de manera objetiva, asegurando que las actividades desarrolladas aporten valor, se enfoquen en las asignaciones establecidas y cumplan con el estándar definido internamente.

Inicialmente para realizar el análisis de desempeño se tomó como insumo principal el conjunto de datos del reporte de gestión de los empleados, el cual fue extraído directamente de la plataforma *Jira*, la cual es una herramienta en línea para la administración de las tareas asignadas a un empleado y desde la cual se registran avances, tiempos de ejecución, estados y demás novedades relacionadas con la gestión.

Adicionalmente fueron utilizados otros conjuntos de datos que complementaron el análisis con datos socio-demográficos del personal y datos de los resultados y valoraciones de los proyectos en los que trabajaron los empleados durante el 2022.

Posteriormente se realizó la revisión completa de las bases de datos, entendiendo su estructura y contenido, se consolidaron todos los registros en una única base de datos. Una vez completa la integración, se procedió a realizar el proceso de selección de datos y minería de datos basándose en la metodología *KDD (Knowledge Discovery in Databases)*, para identificar los atributos críticos requeridos para llevar a cabo el estudio de desempeño.

A continuación, se realizó el análisis descriptivo de los datos (*EDA*) para conocer el comportamiento de los datos y entender de mejor manera los atributos seleccionados para realizar el análisis de desempeño.

Se construyó un modelo de *machine learning* en lenguaje *Python* a partir de las librerías, y módulos vistos en el master con el fin de realizar la clasificación de los empleados según los resultados producidos durante el 2022 de acuerdo al estándar definido por la empresa. El modelo de *machine learning* inicialmente fue trabajado como un modelo de clusterizacion de aprendizaje no supervisado, una vez con el resultado de la agrupación de los empleados se llevó a cabo un modelo de clasificación de aprendizaje supervisado para categorizar a los empleados, ya que esta es una manera útil de mejorar la precisión del algoritmo. por medio de la generación de modelo de machine learning

Los resultados del estudio permitieron conocer como fue el desempeño laboral de cada empleado durante el 2022 y realizar el respectivo análisis en relación a las variables que tuvieron un impacto directo en el rendimiento de estos

Finalmente, para la visualización de los resultados obtenidos del estudio de desempeño fue construido un tablero de control. Adicionalmente se incluyeron algunas recomendaciones e insghts para aportan valor a la empresa.

Para realizar el análisis se han utilizado diferentes técnicas y herramientas estudiadas en el Máster de *Big Data* y *Data Science* de la VIU. Algunas de las herramientas utilizadas fueron: *Google Colab, Excel, Jira* y *Power BI*.

2. Objetivos

El objetivo principal de este proyecto es realizar un estudio sobre el desempeño laboral de los empleados de una empresa del sector tecnológico, a través del análisis de los datos asociados a la gestión realizada durante el año 2022 que permita optimizar la toma de decisiones con respecto al personal de la empresa.

Los objetivos específicos de este Trabajo de Fin de Master (TFM) son:

2.1. Análisis de impacto de factores en el desempeño

Al ser la primera medición de desempeño con un enfoque basado en datos que se realiza en la empresa, para el negocio es fundamental poder identificar el conjunto de atributos que tienen un mayor impacto en la gestión realizada por los empleados, así como entender el nivel de relación entre estos y determinar los problemas que puedan estar limitando el rendimiento de los empleados

Los factores que afectan el desempeño fueron definidos dentro del estudio por medio de técnicas para la selección, preprocesado y análisis exploratorio de los datos.

2.2. Administración adecuada del recurso humano

Por otro lado, el personal es unos de los recursos más importantes con los que cuenta la empresa por lo cual siempre está en busca de gestionarlo de la mejor manera posible, por lo que al realizar el análisis de desempeño se busca generar conocimiento que permita a la empresa gestionar y desarrollar adecuadamente el talento humano para maximizar su potencial y así lograr los objetivos de la organización y mejorar el rendimiento individual y colectivo de los empleados

Adicionalmente por medio de visualizaciones adecuadas, facilitar el entendimiento de los resultados obtenidos por parte de cualquier área de la empresa.

2.3. Optimización de los costos del personal

El sector tecnológico es uno de los sectores con mejores salarios en el mercado, por lo cual también se pretende validar si la relación que existe entre el desempeño y aporte del empleado a la empresa es proporcional al salario que le paga la empresa al empleado.

De forma que se pueda tener conocimiento que sirva para valorar el aporte y conocer si se debe ajustar el salario que se debe pagar a los empleados, con el fin de optimizar los costos de nómina según el rendimiento laboral y mejorar la eficiencia de la empresa.

3. Estado del Arte y Marco teórico

De acuerdo con los objetivos establecidos en esta investigación, es pertinente profundizar el análisis en algunas publicaciones y documentos que dan soporte metodológico a esta investigación y permiten establecer el alcance de los estudios previos realizados sobre el descubrimiento de conocimiento a través de datos para medir el desempeño y analizar el rendimiento laboral en equipos de tecnología.

Sobre los indicadores Salgueiro (2001), afirma que la manera más eficaz de mejorar los resultados globales de la empresa y de los empleados es a través de la medición de indicadores, ya que permiten conocer el estado de los procesos, controlar la evaluación de la empresa y optimizar su desempeño.

Asimismo, los indicadores son la parte más importante de la empresa, por medio de estos se pueden tomar mejores decisiones en los momentos que se pueda requerir, optimizando y garantizando la calidad del servicio que presta la empresa (Perez & Mesanat, 2006).

La metodología seleccionada para este Trabajo de Fin de Master (TFM) es *KDD (Knowledge Discovery in Databases)*, el cual es "un campo de la inteligencia artificial de rápido crecimiento, que combina técnicas del aprendizaje de máquina, reconocimiento de patrones, estadística, bases de datos, y visualización para automáticamente extraer conocimiento, de un nivel bajo de datos" (Fayyad, 1997, como se citó en Nigro. Xodo. Corti. Terren, 2022).

Para esta investigación se busca evaluar objetivamente la gestión realizada por los empleados, con el fin de valorar los aportes generados e identificar brechas y desviaciones existentes en la operación de la empresa, por lo que según lo mencionado por Fayyad (1997), se extrajo conocimiento de las bases de datos de los reportes de gestión del personal y a partir de ahí se construyó un modelo que permite medir el desempeño de los empleados con un enfoque *data driven.*

La evaluación de desempeño "pretende medir el grado con el que un empleado se ajusta y cumple con el perfil deseado. Los resultados de esta medición permiten generar recomendaciones, buscando mejorar el desempeño de las funciones por parte del empleado" (Arbeláez, 2019, p.1).

Oliveira (2016) sostiene que, la productividad en empresas de tecnología se entiende como la efectividad del esfuerzo productivo, es decir, la tasa de producción por unidad de entrada y que la percepción de esta está sujeta a diferentes factores como la entrega de tareas a tiempo, artefactos producidos que no requieran reprocesos, productos que cumplan con las expectativas de los clientes y actitudes de enfoque y proactividad por parte del recurso humano.

3.1. Herramientas utilizadas

Para el desarrollo de este Trabajo de Fin de Master (TFM) fue necesario utilizar algunas herramientas especializadas en el análisis de datos, por lo que en este apartado menciono algunas de ellas.

3.1.1. Google Colab

Google Colab es una plataforma de programación en línea que se basa en el sistema de *Jupyter Notebooks* y utiliza la infraestructura de Google para ofrecer una experiencia de programación colaborativa. Al ser una herramienta en línea los usuarios pueden acceder a sus archivos desde cualquier lugar y dispositivo con conexión a internet.

La plataforma se lanzó en 2017 y se ha convertido en una herramienta popular para la investigación y el aprendizaje automático. *Google Colab* es una herramienta útil para la exploración y el análisis de datos, lo que incluye la realización de procesos de *KDD (Knowledge Discovery in Databases).*

Los cuadernos de *Google Colab* ofrecen una forma interactiva de trabajar con datos, lo que permite a los usuarios importar y manipular conjuntos de datos directamente en el cuaderno. Además, *Google Colab* tiene integraciones con varias bibliotecas populares de ciencia de datos, como *Pandas, NumPy* y *Scikit-Learn*, que pueden ayudar a los usuarios a realizar tareas comunes de *KDD*, como la limpieza y preprocesamiento de datos, la visualización y la modelización de datos.

Ilustración 1. Logo de Google Colab (Google Colab, 2023)

3.1.2. Power BI

Power BI es una herramienta de inteligencia empresarial (*BI)* creada por *Microsoft* que permite a los usuarios conectarse a una variedad de fuentes de datos, transformar y limpiar datos, y crear visualizaciones y paneles interactivos para ayudar a tomar decisiones basadas en datos. La plataforma fue lanzada en 2015 y se ha convertido en una herramienta popular para la visualización y análisis de datos empresariales

.

La funcionalidad de *Power BI* es muy versátil, ya que permite a los usuarios conectarse a una amplia gama de fuentes de datos, incluidas bases de datos, aplicaciones de software y servicios en la nube. *Power BI* también incluye una función de transformación y limpieza de datos, lo que permite preparar y manipular datos para su análisis.

La plataforma de *Power BI* también incluye herramientas avanzadas de análisis y modelado de datos, que permiten a los usuarios crear modelos de datos complejos y realizar análisis estadísticos y de aprendizaje automático. Además, *Power BI* se integra con otras herramientas de *Microsoft*, como *Excel* y *Azure*, lo que permite a los usuarios trabajar con una variedad de herramientas de datos en un solo entorno.

Ilustración 2. Logo de Power BI (Power BI, 2023)

3.1.3. Excel

Excel es un *software* de hojas de cálculo desarrollado por *Microsoft* en 1985 para facilitar la manipulación de datos numéricos y alfanuméricos en una interfaz gráfica. Desde entonces, **Excel** se ha convertido en una herramienta popular y ampliamente utilizada para el análisis de datos, la creación de informes y la gestión de proyectos.

La funcionalidad de *Excel* es muy versátil, ya que permite a los usuarios realizar una amplia gama de tareas, desde cálculos básicos hasta análisis de datos avanzados, como tablas dinámicas, gráficos y modelos de pronóstico. *Excel* también ofrece funciones de colaboración, lo que permite a los usuarios trabajar en el mismo documento de *Excel* en tiempo real y compartir y enviar archivos fácilmente.

Además, *Excel* se integra con otras herramientas de *Microsoft* lo que permite a los usuarios crear informes basados en datos. También ofrece complementos para la integración con herramientas de análisis de datos y visualización.

Ilustración 3. Logo de Excel (Excel, 2023)

3.1.4. Jira

Jira es una herramienta de gestión de proyectos y seguimiento de incidencias desarrollada por Atlassian que se utiliza para ayudar a los equipos a planificar, realizar seguimiento y entregar proyectos de manera más eficiente. Se puede utilizar para una amplia variedad de proyectos, desde el desarrollo de *software* hasta la gestión de proyectos, operaciones, recursos humanos y más.

La herramienta *Jira* se centra en la gestión de incidencias, lo que significa que los usuarios pueden crear, priorizar y asignar incidencias a los miembros del equipo para que las resuelvan. Estas incidencias pueden ser cualquier cosa, desde errores de *software* hasta solicitudes de nuevas funciones y mejoras. Los miembros del equipo pueden ver y actualizar el estado de las incidencias en tiempo real, lo que les permite trabajar de manera más eficiente y colaborar de forma efectiva.

Esta herramienta ayuda a los usuarios a mantenerse actualizados sobre el progreso del proyecto y a tomar decisiones informadas sobre cómo avanzar en su trabajo diario. Adicionalmente *Jira* recopila la información relevante de todas las incidencias dentro de ese período y genera un informe que puede ser exportado a diferentes formatos, como *Excel* o *PDF*

Además de la gestión de incidencias, *Jira* también cuenta con una amplia variedad de características adicionales, como la gestión de proyectos, la creación de informes y la integración con otras herramientas de *software* como *Confluence* y *Bitbucket*. *Jira* es una herramienta muy popular y ampliamente utilizada en la gestión de proyectos y seguimiento de incidencias y es una opción valiosa para cualquier equipo que busque aumentar la eficiencia y la colaboración en su trabajo diario.

Ilustración 4. Ilustración 3. Logo de Jira (Jira, 2023)

.

4. Desarrollo del proyecto y resultados

Una vez definidos los objetivos del Trabajo de Fin de Master (TFM), se realizó la construcción de un roadmap donde se identificaron las actividades requeridas para el cumplimiento de los objetivos establecidos previamente, las cuales se visualiza que están organizadas en orden secuencial para desarrollar el análisis de desempeño.

ID	Actividad	Enero	Febrero	Marzo	Abril
1	Selección de datos				
2	Minería de datos				
3	Análisis exploratorio de datos (EDA)				
4	Modelado de datos				
5	Evaluación de desempeño				
6	Visualización de resultados				
7	Análisis de resultados				

Ilustración 5. Cronograma de las tareas definidas. Elaboración: propia

Adicionalmente se llevó a cabo la revisión de los recursos necesarios para ejecutar adecuadamente el estudio y se realizó la selección de la metodología para la gestión del proyecto, la cual es *KDD (Knowledge Discovery in Databases)* que permite la exploración de los datos para el descubrimiento de patrones y relaciones que son analizados posteriormente para generar conocimiento.

Para la evaluación de desempeño se utilizaron diferentes modelos de *machine learning* de tipo supervisado y no supervisado, con el fin de encontrar el que mejor se adecuara a la necesidad de la empresa y genera los mejores resultados con los datos disponibles.

Dentro del desarrollo del proyecto se priorizó la construcción de artefactos que aporten valor al estudio, mediante la implementación de los temas y conceptos vistos en las asignaturas del master. Por lo cual fueron construidos los siguientes artefactos:

- Base de datos consolidada
- *Notebook* de *Google Colab* con análisis descriptivo (*EDA*)
- Notebook de *Google Colab* con modelo de *machine learning (ML)*
- Tablero de control con visualización de resultados.

4.1. Metodología

4.1.1. KDD (Knowledge Discovery in Databases)

La metodología seleccionada para la ejecución del proyecto es *KDD* o descubrimiento de conocimiento en bases de datos, el cual es un proceso que consiste en identificar patrones en forma de reglas o funciones, a partir de los datos, con el fin de extraer conocimiento de grandes volúmenes de datos para que el usuario los analice posteriormente. (Fayyad et al., 1996).

> La metodología *KDD* está basada en un bien definido proceso de múltiples pasos, para el descubrimiento de conocimiento en grandes colecciones de datos. El proceso *KDD* es iterativo por naturaleza, y depende de la interacción para la toma de decisiones, de manera dinámica". Es decir, es el usuario quien toma las decisiones durante todo el proceso, selecciona las herramientas y técnicas para llevar a cabo el proceso, por lo cual los resultados obtenidos son claramente afectados por este (Gupta. Bhatnagar. Wasan, 1997).

Es importante tener en cuenta que el *KDD* tiene la propiedad de ser altamente interactivo, es decir que es un proceso centrado en el usuario, al ser este quien debe inicialmente identificar la problemática a la que se va a enfrentar, establecer los objetivos que se desean alcanzar, entender el dominio de los datos y analizar el contexto, las propiedades, limitaciones y reglas del escenario en estudio para proponer soluciones viables y factibles (Nigro. Xodo. Corti. Terren, 2022).

El *KDD* es un proceso conformado por un conjunto de 5 etapas que son ejecutadas de manera secuencial, las cuales son:

- Selección.
- Preprocesamiento/ limpieza.
- Transformación.
- Minería de datos.
- Interpretación/ evaluación

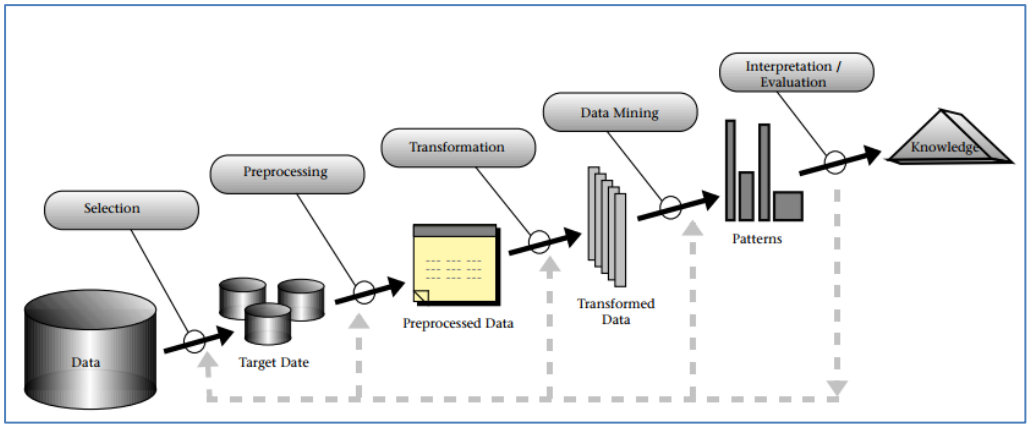


Ilustración 6. Etapas del proceso de minería de datos. Fuente: https://cutt.ly/SH91qit

- **Selección y preparación**

Corresponde a la creación del conjunto de datos objetivo, sobre el cual se realizará el proceso de descubrimiento. La selección de los datos varía de acuerdo con los objetivos establecidos. Una vez preparados los datos, se debe especificar dónde se encuentran los datos de entrada, cuales atributos de los datos de entrada son apropiados para el proyecto, qué atributos se deben utilizar para la función de salida y dónde desea almacenar el modelo final una vez sea construido (Landa, 2018).

- **Preprocesado**

El preprocesamiento de datos es una etapa esencial del proceso de descubrimiento de conocimiento o *KDD (Knowledge Discovery in Databases)* (Han. Kamber. Pei. J. Zaki. Wagner, 2012, 2014). Esta etapa se encarga de la limpieza de datos, su integración y reducción para la siguiente fase de minería de datos (Luengo. Herrera. García, 2014). Se analiza la calidad de los datos seleccionados, por lo cual se realizan operaciones para el manejo de datos faltantes, datos vacíos, datos nulos, datos duplicados y datos ruidosos por medio de técnicas estadísticas estándar.

Los datos vacíos son aquellos a los cuales no les corresponde un valor en el mundo real y los datos faltantes son aquellos que tienen un valor que no fue capturado. Los datos ruidosos son valores que están significativamente fuera del rango de valores esperados; se deben principalmente a errores humanos, a cambios en el sistema, a información no disponible a tiempo y a fuentes.

Los datos nulos son datos desconocidos que son permitidos por los sistemas gestores de bases de datos relacionales. En el proceso de preprocesado estos valores se ignoran, se reemplazan por un valor por omisión, o por el valor más cercano, es decir, se usan métricas de tipo estadístico como media, moda, mínimo y máximo para reemplazarlos.

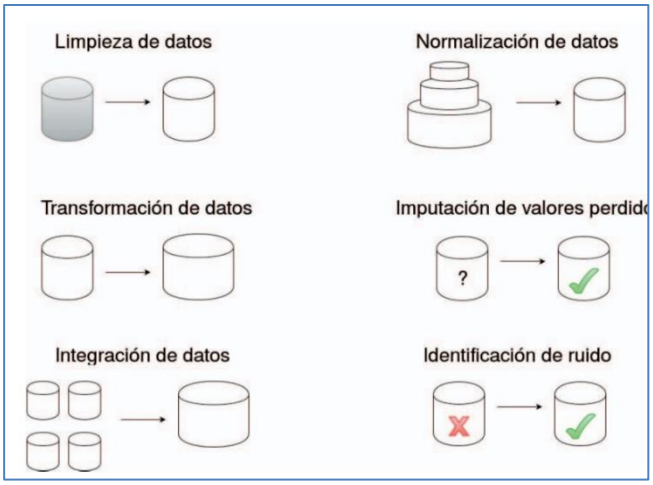


Ilustración 7. Familias de preprocesado de datos. Fuente: https://cutt.ly/NH91dwS

- **Transformación**

Consiste en la búsqueda de características útiles para representar los datos dependiendo de la meta del proceso. Se utilizan métodos de reducción de dimensiones o de transformación para disminuir el número efectivo de variables bajo consideración o para encontrar representaciones invariantes de los datos (Fayyad et al., 1996).

Los métodos de reducción de dimensiones pueden simplificar una tabla de una base de datos horizontal o verticalmente. La reducción horizontal implica la eliminación de tuplas idénticas como producto de la sustitución del valor de un atributo por otro de alto nivel, en una jerarquía definida de valores categóricos o por la discretización de valores continuos. La reducción vertical implica la eliminación de atributos que son insignificantes o redundantes con respecto al problema, como la eliminación de llaves, la eliminación de columnas que dependen funcionalmente.

- **Minería de datos**

El objetivo de esta etapa es la búsqueda y extracción de conocimiento útil, por medio del descubrimiento de patrones insospechados, ocultos, implícitos y de interés, a través de la utilización de modelos de minería de datos que pueden ser predictivos o descriptivos (Moine. Haedo. Gordillo, 2011).

Los modelos predictivos pretenden estimar valores futuros o desconocidos. Entre las tareas predictivas están la clasificación y la regresión. Mientras que los modelos descriptivos identifican patrones que explican o resumen los datos; sirven para explorar las propiedades de los datos examinados.

Entre las tareas descriptivas se encuentran reglas de asociación, patrones secuenciales, clustering y correlaciones. Por lo cual, para la selección de un modelo de minería de datos se debe tener en cuenta la respuesta que se espera generar con los datos, los parámetros que mejor se ajusten al modelo y los tipos de datos a utilizar.

- **Interpretación / Evaluación**

En esta etapa se interpretan los patrones descubiertos y se evalúan los resultados obtenidos. En caso que los resultados no sean satisfactorios, posiblemente se retorna a las anteriores etapas para posteriores iteraciones.

En caso que los resultados sean satisfactorios, se procede a construir la visualización de los resultados, la eliminación de los patrones redundantes o irrelevantes, la traducción de los patrones útiles en términos que sean entendibles y la generación de juicios y valoraciones sobre los patrones resultantes para conocer el rendimiento obtenido y ver si dan cumplimiento a los objetivos establecidos al inicio del proceso. Por último, se aplica el conocimiento encontrado al contexto y se comienza a resolver las problemáticas (Landa, 2018).

4.1.2. Machine Learning

El *machine learning* o aprendizaje automático, es una rama de la inteligencia artificial que permite a las computadoras aprender y mejorar automáticamente a partir de datos y experiencia previa sin ser programadas explícitamente. Tal como lo definen Alpaydin (2010) y Mitchell (1997), el *machine learning* es el "diseño y desarrollo de algoritmos que permiten a las máquinas mejorar automáticamente a través de la experiencia" y la "construcción de sistemas informáticos que mejoran automáticamente con la experiencia".

El *machine learning* se utiliza en una amplia gama de aplicaciones, desde el análisis de datos y la detección de fraude hasta la visión artificial y el procesamiento del lenguaje natural, en el cual se utiliza para la traducción automática y el análisis de sentimientos.

El *machine learning* tiene una historia que se remonta a la década de 1940, cuando los científicos comenzaron a desarrollar algoritmos que permitían a las máquinas aprender de manera autónoma. Uno de los primeros ejemplos de machine learning se produjo en 1952, cuando Arthur Samuel desarrolló un programa que podía aprender a jugar a las damas mejorando con la experiencia. Desde entonces, el *machine learning* ha evolucionado rápidamente, impulsado por avances en la computación y la disponibilidad de grandes cantidades de datos.

Las funcionalidades del *machine learning* son diversas y permiten a los usuarios realizar una amplia gama de tareas, desde el análisis de datos hasta la toma de decisiones automatizada. El *machine learning* se utiliza ampliamente en la industria para el análisis de datos y la predicción de resultados, como la detección de fraude y la optimización de precios. También se utiliza en aplicaciones de visión artificial, como el reconocimiento facial y la clasificación de imágenes.

El *machine learning* también se utiliza en la investigación médica y científica, permitiendo a los científicos analizar grandes cantidades de datos y descubrir patrones que de otra manera podrían pasar desapercibidos. Según Domínguez (2019), "la aplicación del *machine learning* en el campo de la medicina es prometedora y puede ayudar en la prevención, el diagnóstico y el tratamiento de enfermedades".

4.2. Planteamiento del problema

Una empresa del sector tecnológico tiene la necesidad de evaluar el desempeño de los empleados durante el último año (2022).

Actualmente en la empresa no existe un modelo basado en datos que soporte la evaluación de desempeño realizada a los empleados, por lo que dicha actividad suele aplicarse de manera subjetiva de acuerdo a lo que el evaluador considere preponderante y enfocada mayormente en la precepción de la gestión. Es por esta razón que se requiere realizar un análisis que evalué de manera objetiva la gestión desarrollada por los empleados a través de los datos reportados en la plataforma *Jira*, donde se encuentra el registro de las tareas ejecutadas, las cuales se conocen como *"issues"* dentro de las metodologías agiles, estas tareas cuentan con una serie de atributos como el estado, tipo, tiempo planeado, tiempo ejecutado, usuario, año, etc.

Para propósitos de este estudio se debe estandarizar la evaluación de desempeño de los empleados teniendo en cuenta las diferentes variables relacionadas a la gestión como el cumplimiento, calidad, satisfacción, efectividad y trabajo en equipo; Así como atributos descriptivos del personal.

Para llevar a cabo el estudio de desempeño fueron seleccionados aleatoriamente 1843 empleados, los cuales trabajaron en alguna de las sucursales de la empresa duramente el 2022 y contaban con diferentes modalidades de trabajo, salario, edad, cargo, ubicación, etc.

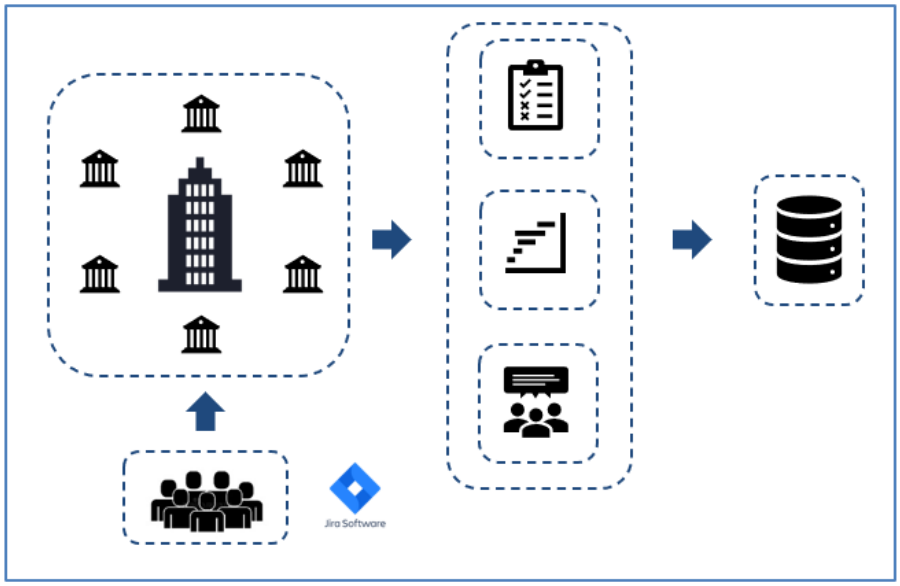


Ilustración 8. Esquema de generación de datos. Fuente; propia

4.3. Desarrollo del proyecto

4.3.1. Selección de los datos

Inicialmente se realizó la revisión completa de todas las bases de datos suministradas, las cuales estaban almacenadas en un repositorio en *GitHub* y se encontraban en idioma alemán, ya que los datos fueron obtenidos en el idioma original de la empresa, por lo que fue necesaria la traducción y entendimiento semántico de los mismos para trabajarlos de forma adecuada. Posteriormente se realizó la validación de los datos con un experto en el dominio para confirmar su consistencia y validez. Las bases de datos suministradas fueron las siguientes:

Reportes de gestión individual: Registros de todas las tareas ejecutadas por los empleados de la empresa durante el año 2022, los cuales se exportaron directamente desde la plataforma *Jira*. Esta base de datos contenía 72 ficheros, debido a que los reportes de gestión individual se encontraban organizados por cada sucursal y llevaba los registros de forma mensual.

Resultados de proyectos: Registro de los resultados obtenidos por cada empleado en el proyecto en el que trabajo durante el año 2022. Está conformado por valoraciones de diferentes atributos en relación a su participación y aporte a los proyectos en general. Estos datos fueron suministrados por el área de Recursos Humanos de la empresa, ya que se encontraban asociados a los perfiles de los empleados en la plataforma de Jira.

Informe del perfil laboral: Registros de los datos sociodemográficos de los empleados de la empresa con contrato vigente durante el año 2022. Estos datos fueron entregados por el área de proyectos (*PMO*) de la empresa.

Una vez completada la revisión de los datos, todos ficheros de reportes de gestión individual fueron integrados en una única base de datos denominada reporte de gestión consolidado, la cual contaba con más de 12.000 registros. La integración de todos los ficheros fue posible gracias a que estos se encontraban en un mismo formato (*csv*) correspondiente a bases de datos de tipo relacional.

Ilustración 9. Distribución de los datos en el repositorio. Elaboración propia

4.3.2. Minería de los datos

Dimensionamiento de datos

Fue necesario integrar el total de las tareas ejecutadas por cada empleado, haciendo la agrupación correspondiente con el objetivo consolidar los registros de cada empleado y generar el resultado individual de la gestión realizada, por lo cual la base de datos quedo con 1843 registros y 8 atributos. (62% cuantitativos y 37% cualitativos)

Carga de datos

Inicialmente fue cargada la base de datos del reporte de gestión consolidado dentro del notebook de *Google Colab* para poder trabajarla como dataframe. También se validó el tipo de atributos disponibles en el dataset.

```
Data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1843 entries, 0 to 1842
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Nombre_Empleado    1843 non-null   object 
 1   Apellido_Empleado  1843 non-null   object 
 2   Nombre_Completo    1843 non-null   object 
 3   Sucursal           1843 non-null   int64  
 4   Cumplimiento       1843 non-null   float64
 5   Efectividad        1843 non-null   float64
 6   Tareas Alto        1843 non-null   float64
 7   Tareas Medio       1843 non-null   float64
 8   Tareas Bajo        1843 non-null   float64
dtypes: float64(5), int64(1), object(3)
memory usage: 129.7+ KB
```

Ilustración 10. Atributos de dataset inicial. Elaboración: propia

Se validó que los datos se encontrarán balanceados, por lo cual se generó un gráfico de barras para ver la distribución de empleados por sucursal.

```
Sucursal
1    312
2    287
3    322
4    310
5    306
6    306
Name: Sucursal, dtype: int64
```

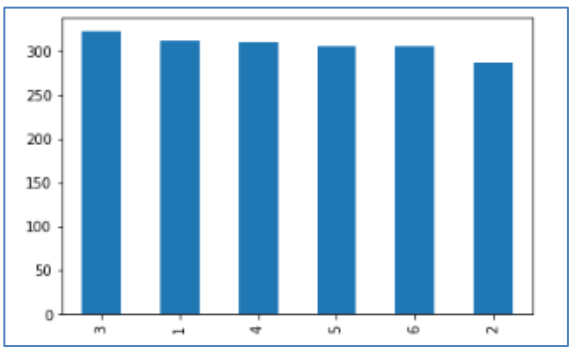


Ilustración 11. Revisión del balanceo de datos. Elaboración: propia

Adicionalmente se concatenaron los atributos Nombre y Apellido del empleado para generar un nuevo atributo denominado "Nombre Completo" con el fin de facilitar el manejo de los datos.

```
Data['Nombre_Empleado'] = Data['Nombre_Empleado'].astype(str)
Data['Apellido_Empleado'] = Data['Apellido_Empleado'].astype(str)
Data['Nombre_Completo'] = Data.apply(lambda row: row['Nombre_Empleado'] + " " + row['Apellido_Empleado'], axis=1)
```

Ilustración 12. Atributo concatenado. Elaboración: propia

Carga de datos adicionales

De igual forma las bases de datos de los resultados de proyectos y del informe del perfil laboral de los empleados fueron cargados en *Google Colab* e integrados al reporte de gestión consolidado con el fin de caracterizar a los empleados y reunir la mayor cantidad de datos posibles para realizar el estudio del desempeño laboral.

```
Empleados = upload_files()
Empleados.head()
```

Elegir archivos Ninguno archivo selec. Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving D2.csv to D2.csv
User uploaded file "D2.csv" with length 214538 bytes
```

	Nombre_Empleado	Apellido_Empleado	Nombre_Completo	Genero	Cargo	Nivel cargo	Edad	Ubicacion	Modalidad	Salario	Estado Civil	Telefono	Contrato Indefinido
0	LIAM	SMITH	LIAM SMITH	M	Arquitecto de Software	Senior	45	Basilea, Suiza	Teletrabajo	$60,000	S	9505405	Si
1	NOAH	JOHNSON	NOAH JOHNSON	M	Ingeniero de software II	Senior	25	Zurich, Suiza	Presencial	$36,000	C	9734761	No
2	OLIVER	WILLIAMS	OLIVER WILLIAMS	M	Arquitecto de Software	Senior	45	Lyon, Francia	Hibrido	$36,000	S	9593611	No
3	ELIJAH	BROWN	ELIJAH BROWN	M	Especialista de Software	Senior	38	Lyon, Francia	Teletrabajo	$36,000	C	9466722	No
4	WILLIAM	JONES	WILLIAM JONES	M	Arquitecto de Software	Senior	46	Touluse, Francia	Teletrabajo	$36,000	C	9621580	Si

Ilustración 14. Dataset de desempeño en proyectos. Elaboración: propia

```
Desempeño = upload_files()
Desempeño.head()
```

Elegir archivos Ninguno archivo selec. Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving D3.csv to D3.csv
User uploaded file "D3.csv" with length 158207 bytes
```

	Nombre_Empleado	Apellido_Empleado	Nombre_Completo	ID_Empleado	Año	Proyecto	Calidad	Satisfacción	Productividad	Trabajo en equipo	Reclamaciones
0	LIAM	SMITH	LIAM SMITH	1001	2022	JustVDB	10.81	0.85	0.81	0.48	99 Reclamaciones
1	NOAH	JOHNSON	NOAH JOHNSON	1002	2022	CloudConnect	25.45	0.74	0.83	0.69	63 Reclamaciones
2	OLIVER	WILLIAMS	OLIVER WILLIAMS	1003	2022	CyberSafe	28.69	0.76	0.74	0.52	66 Reclamaciones
3	ELIJAH	BROWN	ELIJAH BROWN	1004	2022	AIBoost	13.20	0.75	0.82	0.54	19 Reclamaciones
4	WILLIAM	JONES	WILLIAM JONES	1005	2022	AIBoost	9.44	0.89	0.86	0.50	16 Reclamaciones

Ilustración 13. Dataset de información del personal. Elaboración: propia

Integración de datos adicionales

La integración de los datos se realizó por medio de un left join utlizando como llave el atributo generado “Nombre_Completo”.

```
Data=Data.merge(Empleados, on='Nombre_Completo', how='left')
Data=Data.merge(Desempeño, on='Nombre_Completo', how='left')
```

Ilustración 15. Integración de los dataset adicionales. Elaboración: propia

Al validar el dataset definitivo se identificó que contaba con 1843 filas y 25 columnas

	Nombre_Empleado_x	Apellido_Empleado_x	Nombre_Completo	Sucursal	Cumplimiento	Efectividad	Tareas alto	Tareas Medio	Tareas Bajo	Nombre_Empleado_y	Apellido_Empleado_y	Genero	Cargo	Nivel cargo	Edad	Ubicacion	Modalidad	Salario	Estado Civil	Telefono	Contrato Indefinido	Nombre_Empleado	Apellido_Empleado
0	LIAM	SMITH	LIAM SMITH	2	18.20	0.15	0.85	0.06	0.09	LIAM	SMITH	M	Arquitecto de Software	Senior	45.0	Basilea, Suiza	Teletrabajo	$60,000	S	9505405.0	Si	LIAM	SMITH
1	NOAH	JOHNSON	NOAH JOHNSON	5	31.75	0.07	0.57	0.00	0.43	NOAH	JOHNSON	M	Ingeniero de software II	Senior	25.0	Zurich, Suiza	Presencial	$36,000	C	9734761.0	No	NOAH	JOHNSON
2	OLIVER	WILLIAMS	OLIVER WILLIAMS	5	39.71	0.14	0.65	0.11	0.24	OLIVER	WILLIAMS	M	Arquitecto de Software	Senior	45.0	Lyon, Francia	Híbrido	$36,000	S	9593611.0	No	OLIVER	WILLIAMS
3	ELIJAH	BROWN	ELIJAH BROWN	3	23.23	0.06	0.57	0.00	0.43	ELIJAH	BROWN	M	Especialista de Software	Senior	38.0	Lyon, Francia	Teletrabajo	$36,000	C	9466722.0	No	ELIJAH	BROWN
4	WILLIAM	JONES	WILLIAM JONES	4	5.41	0.19	0.55	0.11	0.34	WILLIAM	JONES	M	Arquitecto de Software	Senior	46.0	Toulouse, Francia	Teletrabajo	$36,000	C	9621580.0	Si	WILLIAM	JONES
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1838	MADALYNN	CERNA	MADALYNN CERNA	2	19.07	0.96	0.58	0.17	0.25	MADALYNN	CERNA	F	Especialista de Software	Junior	24.0	Madrid, España	Teletrabajo	$9,000	C	8829509.0	No	MADALYNN	CERNA
1839	MELANI	VIRAMONTES	MELANI VIRAMONTES	3	24.57	0.91	0.58	0.17	0.25	MELANI	VIRAMONTES	F	Ingeniero de software I	Junior	29.0	Zurich, Suiza	Híbrido	$7,000	C	9877030.0	Si	MELANI	VIRAMONTES
1840	LAYLANI	GALDAMEZ	LAYLANI GALDAMEZ	2	4.61	0.95	0.58	0.17	0.25	LAYLANI	GALDAMEZ	F	Especialista de Software	Junior	25.0	Madrid, España	Híbrido	$13,000	O	9237633.0	No	LAYLANI	GALDAMEZ
1841	MAZIKEEN	OLIVER	MAZIKEEN OLIVER	6	25.53	0.99	0.58	0.17	0.25	MAZIKEEN	OLIVER	F	Ingeniero de software I	Junior	21.0	Toulouse, Francia	Híbrido	$13,000	S	9378389.0	No	MAZIKEEN	OLIVER
1842	BELLE	ANDERSON	BELLE ANDERSON	3	22.48	0.98	0.58	0.17	0.25	BELLE	ANDERSON	F	Desarrollador de software II	Junior	22.0	Ginebra, Suiza	Híbrido	$12,000	O	8887108.0	No	BELLE	ANDERSON

1843 rows × 31 columns

Ilustración 16. Dataset consolidado. Elaboración: propia

Consulta tipo de variables

Consulto el nombre de las columnas presentes en el *dataset*, valido la tipología de los atributos (cuantitativos y cualitativos) respectivamente y reviso la existencia de valores duplicados en el dataset.

```
Data.columns
```

```
Index(['Nombre_Completo', 'Sucursal', 'Cumplimiento ', 'Efectividad',
       'Tareas Alto', 'Tareas Medio', 'Tareas Bajo', 'Genero', 'Cargo',
       'Nivel cargo', 'Edad ', 'Modalidad', ' Salario ', ' Estado Civil ',
       ' Telefono ', ' Contrato Indefinido ', 'ID_Empleado', 'Año', 'Proyecto',
       'Calidad', 'Satisfacción', 'Productividad', 'Trabajo en equipo',
       'Reclamaciones', 'Ciudad', 'Pais'],
      dtype='object')
```

```
df_numeric = Data.select_dtypes(include=[np.number])
numeric_cols = df_numeric.columns.values
print(numeric_cols)
```

```
['Sucursal' 'Cumplimiento ' 'Efectividad' 'Tareas Alto' 'Tareas Medio'
 'Tareas Bajo' 'Edad ' ' Salario ' ' Telefono ' 'ID_Empleado' 'Año'
 'Calidad' 'Satisfacción' 'Productividad' 'Trabajo en equipo'
 'Reclamaciones']
```

```
df_non_numeric = Data.select_dtypes(exclude=[np.number])
non_numeric_cols = df_non_numeric.columns.values
print(non_numeric_cols)
```

```
['Nombre_Completo' 'Genero' 'Cargo' 'Nivel cargo' 'Modalidad'
 ' Estado Civil ' ' Contrato Indefinido ' 'Proyecto' 'Ciudad' 'Pais']
```

```
Data.duplicated().value_counts()
```

```
False    1843
dtype: int64
```

Ilustración 17. Consulta de variables del dataset consolidado. Elaboración: propia

Limpieza de datos

Una vez integrados todos los datos en el reporte de gestión consolidado, se realizó la limpieza de los datos con el fin de mejorar la calidad e integridad de los mismos.

Al aplicar este proceso se puede mejorar la precisión de los resultados del análisis a realizar, así como facilitar la toma de decisiones y ahorrar tiempo al eliminar previamente inconsistencias que puedan presentarse en los datos.

Imputación de datos faltantes

Por medio de diferentes métodos fue validada la existencia de datos faltantes en el *dataset*, ya que era necesario asegurar que los atributos a trabajar contarán con la calidad suficiente (es decir valores faltantes menores al 10% de los registros) para aportar valor al análisis, de no ser así el atributo debía ser eliminado.

```
Data.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1843 entries, 0 to 1842
Data columns (total 26 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Nombre_Completo        1843 non-null   object
 1   Sucursal               1843 non-null   int64
 2   Cumplimiento           1843 non-null   float64
 3   Efectividad            1843 non-null   float64
 4   Tareas Alto            1843 non-null   float64
 5   Tareas Medio           1843 non-null   float64
 6   Tareas Bajo            1843 non-null   float64
 7   Genero                 1808 non-null   object
 8   Cargo                  1808 non-null   object
 9   Nivel cargo            1808 non-null   object
 10  Edad                   1808 non-null   float64
 11  Modalidad              1808 non-null   object
 12   Salario               1808 non-null   float64
 13   Estado Civil          1808 non-null   object
 14   Telefono              1808 non-null   float64
 15   Contrato Indefinido   1808 non-null   object
 16  ID_Empleado            1808 non-null   float64
 17  Año                    1808 non-null   float64
 18  Proyecto               1808 non-null   object
 19  Calidad                1808 non-null   float64
 20  Satisfacción           1808 non-null   float64
 21  Productividad          1808 non-null   float64
 22  Trabajo en equipo      1808 non-null   float64
 23  Reclamaciones          1808 non-null   float64
 24  Ciudad                 1808 non-null   object
 25  Pais                   1808 non-null   object
dtypes: float64(15), int64(1), object(10)
memory usage: 388.8+ KB
```

```
Data.isnull().sum()

Nombre_Completo          0
Sucursal                 0
Cumplimiento             0
Efectividad              0
Tareas Alto              0
Tareas Medio             0
Tareas Bajo              0
Genero                  35
Cargo                   35
Nivel cargo             35
Edad                    35
Modalidad               35
 Salario                35
 Estado Civil           35
 Telefono               35
 Contrato Indefinido    35
ID_Empleado             35
Año                     35
Proyecto                35
Calidad                 35
Satisfacción            35
Productividad           35
Trabajo en equipo       35
Reclamaciones           35
Ciudad                  35
Pais                    35
dtype: int64
```

Ilustración 18. Imputación de datos faltantes. Elaboración: propia

Se identificaron valores faltantes por medio de un gráfico de *heatmap* y un gráfico de barras, en los que se observa que los valores faltantes no superaban el 2% de los registros de algunos atributos.

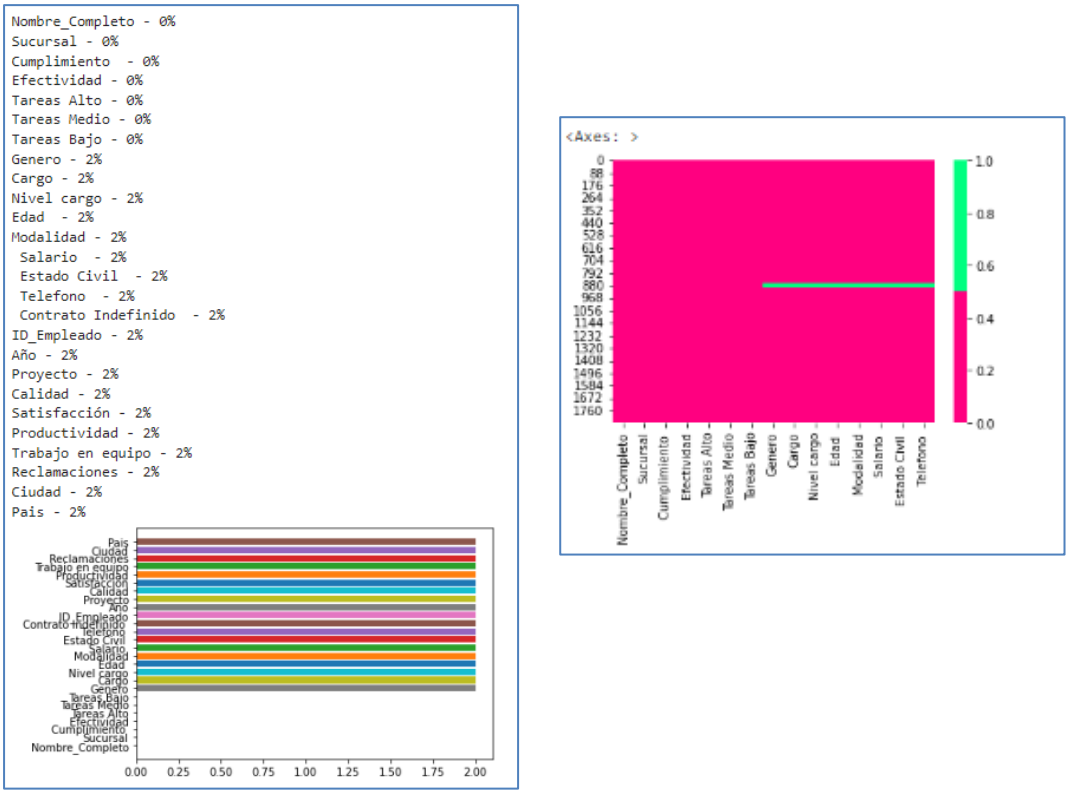


Ilustración 19. Gráficos de valores faltantes. Elaboración: propia

Adicionalmente se hizo uso de un recurso *online* trabajado por medio de una librería externa llamado *Feature Selector* que permite explorar de mejor manera el *dataset*, analizando los valores faltantes, validando la correlación de los atributos y definiendo la importancia de los atributos para el análisis a desarrollar.

Este recurso fue realizado en lenguaje *Python* por Will Koehresen y puede ser consultado en la siguiente página web: https://github.com/WillKoehrsen/feature-selector

Se utilizó feature selector para validar si existían atributos que contaran con más del 80% de valores faltantes, de ser así dichos atributos serian eliminados del dataset. Ninguno de los atributos cumple esta condición.

```
fs = FeatureSelector(data = Data, labels = Data.columns)
fs.identify_missing(missing_threshold=0.8)

0 features with greater than 0.80 missing values.
```

Ilustración 20. Feature selector para validar valores faltantes. Elaboración: propia

Utilizo feature_selector para generar un histograma con la distribución de los valores faltantes. El cual indica que aproximadamente 1805 registros presentan 0 valores faltantes mientras que 19 registros presentan 35 valores faltantes.

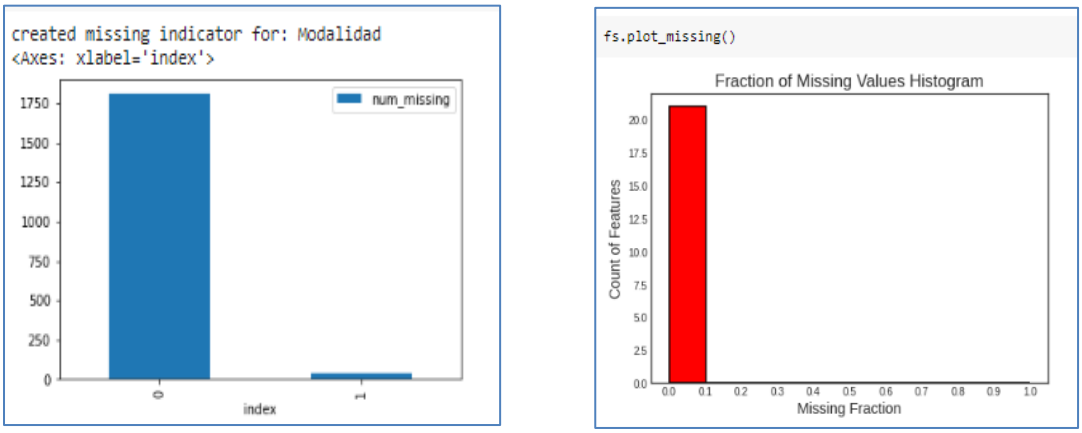


Ilustración 21. Gráficos para validación de valores faltantes. Elaboración: propia

Gestiono los valores faltantes. Podría eliminarlos a través de la función *dropna*, pero estos empleados quedarían por fuera de análisis de desempeño por los cual procedo a imputar los valores nulos con una medida estadística como la moda para cada atributo.

```
Data['Genero'].fillna(Data['Genero'].mode()[0], inplace=True)
Data['Modalidad'].fillna(Data['Modalidad'].mode()[0], inplace=True)
Data['Cargo'].fillna(Data['Cargo'].mode()[0], inplace=True)
Data['Nivel cargo'].fillna(Data['Nivel cargo'].mode()[0], inplace=True)
Data['Edad '].fillna(Data['Edad '].mode()[0], inplace=True)
Data[' Salario '].fillna(Data[' Salario '].mode()[0], inplace=True)
Data['Calidad'].fillna(Data['Calidad'].mode()[0], inplace=True)
Data['Satisfacción'].fillna(Data['Satisfacción'].mode()[0], inplace=True)
Data['Productividad'].fillna(Data['Productividad'].mode()[0], inplace=True)
Data['Trabajo en equipo'].fillna(Data['Trabajo en equipo'].mode()[0], inplace=True)
Data['Reclamaciones'].fillna(Data['Reclamaciones'].mode()[0], inplace=True)
Data['Proyecto'].fillna(Data['Proyecto'].mode()[0], inplace=True)
Data['Ciudad'].fillna(Data['Ciudad'].mode()[0], inplace=True)
Data['Pais'].fillna(Data['Pais'].mode()[0], inplace=True)
Data['ID_Empleado'].fillna(Data['ID_Empleado'].mode()[0], inplace=True)
```

Ilustración 22. Imputación de valores faltantes. Elaboración: propia

Validación de datos inconsistentes

Debido a que los datos fueron extraídos de la plataforma *Jira* y previamente habían sido procesados por la empresa, se evidenció que no existían datos inconsistentes con errores por palabras mal tipeadas, con mayúsculas y minúsculas, espacios, caracteres especiales o inconsistencias similares.

Transformación de variables

Se realizo la transformación de los atributos del dataset para mejorar la interpretación de estos y entender la correlación existente, así como para poder trabajarlos de forma adecuada en el análisis de datos.

Adicionalmente se crearon nuevos atributos que pueden aportar información y tener mayor relevancia dentro del estudio.

Ajuste de formato de los datos

Eliminó el símbolo de dólar del atributo "Salario", adicionalmente reemplazo la coma por punto para que el atributo se entienda como de tipo decimal y se pueda trabajar.

```
Data[' Salario '] = Data[' Salario '].str.replace('$', '').str.replace(',', '.').astype(float)
```

Ilustración 23. Ajuste de formato del atributo salario. Elaboración: propia

Elimino el texto que acompaña a los datos del atributo "Reclamaciones" para dejar únicamente el valor numérico y adicionalmente convierto el atributo en numérico para poder trabajarlo.

```
Data['Reclamaciones'] = Data['Reclamaciones'].str.replace(' Reclamaciones', '')
Data['Reclamaciones']=pd.to_numeric(Data['Reclamaciones'])
```

Ilustración 24. Ajuste de formato del atributo reclamaciones. Elaboración: propia

Separo los datos del atributo "Ubicación", ya que deseo trabajarlos de forma independiente por lo que generó dos nuevos atributos llamados "Ciudad" y "País".

```
Data[['Ciudad', 'Pais']] = Data['Ubicacion'].str.split(',', expand=True)
Data.drop('Ubicacion', axis=1, inplace=True)
print(Data.head())
```

Ilustración 25. Ajuste de formato del atributo ubicación. Elaboración: propia

Análisis de datos fuera de rango

A través de un gráfico de *boxplot* reviso la existencia de *outliers* dentro del *dataset* con el fin de identificar de errores en los datos de entrada y patrones inusuales, así como proveer información valiosa para la toma de decisiones.

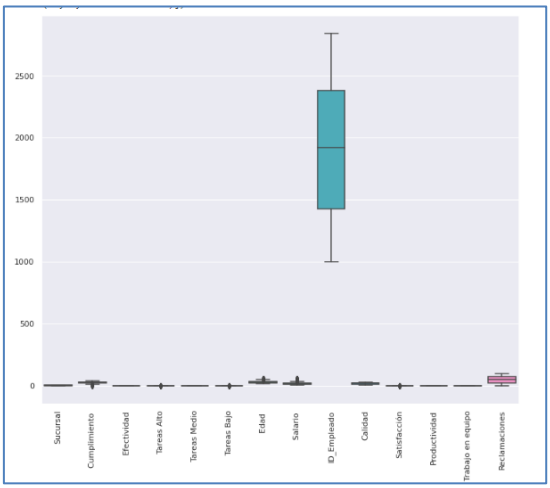


Ilustración 26. Análisis de outliers. Elaboración: propia

Únicamente se identifica una alta variabilidad en los datos del número de teléfono de los empleados, lo cual no es significativo para el análisis, por lo cual este atributo será eliminado posteriormente.

Eliminación datos innecesarios

Se identificaron 5 atributos con información irrelevante que no aportaba valor para el análisis debido al ser de tipo cualitativo, por lo cual fueron eliminados, lo cual no implico una perdida significativa de información.

```
Data=Data.drop(columns=["Nombre_Completo", ' Estado Civil ', " Telefono ", ' Contrato Indefinido ', 'Año'], axis=1)
```

Ilustración 27. Eliminación de atributos. Elaboración: propia

Correlación de las variables

Los atributos que presentan una alta correlación, es decir con un coeficiente de correlación superior al 80% (como se indica en la ilustración) deben ser eliminados ya que sería redundante mantenerlos en el dataset.

```
collinear_features = fs.ops['collinear']
collinear=fs.record_collinear
collinear.sort_values(by='corr_value', ascending=False)
```

	drop_feature	corr_feature	corr_value
1	ID_Empleado	Efectividad	0.953601
0	Tareas Bajo	Tareas Alto	-0.897337

Ilustración 28. Variables correlacionadas 1. Elaboración: propia

Exportación de dataset limpio

Realizo la exportación del *dataset* limpio, para tener una copia de seguridad, en caso de que se necesite añadir información extra o realizar algún proceso adicional al *dataset* en el futuro, ya que esto permite trabajar con la información anterior limpia y depurada.

Selección de atributos para análisis exploratorio

Los atributos considerados para realizar el análisis exploratorio fueron los siguientes:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1843 entries, 0 to 1842
Data columns (total 20 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Sucursal           1843 non-null   int64
 1   Cumplimiento       1843 non-null   float64
 2   Efectividad        1843 non-null   float64
 3   Tareas Alto        1843 non-null   float64
 4   Tareas Medio       1843 non-null   float64
 5   Genero             1843 non-null   object
 6   Cargo              1843 non-null   object
 7   Nivel cargo        1843 non-null   object
 8   Edad               1843 non-null   float64
 9   Modalidad          1843 non-null   object
 10   Salario           1843 non-null   float64
 11  ID_Empleado        1843 non-null   float64
 12  Proyecto           1843 non-null   object
 13  Calidad            1843 non-null   float64
 14  Satisfacción       1843 non-null   float64
 15  Productividad      1843 non-null   float64
 16  Trabajo en equipo  1843 non-null   float64
 17  Reclamaciones      1843 non-null   float64
 18  Ciudad             1843 non-null   object
 19  Pais               1843 non-null   object
dtypes: float64(12), int64(1), object(7)
memory usage: 366.9+ KB
```

Ilustración 29. Atributos seleccionados para EDA. Elaboración: propia

A partir de los atributos seleccionados se puede concluir que se generó una reducción del 24% de los atributos que componían el *dataset* definitivo sin tener una perdida significativa de información. Adicionalmente se evidencia que el 68% de los atributos seleccionados corresponden a atributos de tipo numérico, lo cual se encuentra alineado con el objetivo del proyecto, que es evaluar el desempeño de los empleados.

4.3.3. Análisis exploratorio de datos (EDA)

El análisis descriptivo de los datos (*EDA*) consistió en analizar el comportamiento de los datos del dataset consolidado, explorar la relación entre las diferentes variables, así como obtener las medidas de centralidad y dispersión de los datos, generar la matriz de correlación y los gráficos correspondientes a las principales variables con el fin de identificar patrones o tendencias que ayuden a entender mejor el problema

El análisis fue realizado en un *notebook* de *Google Colab* haciendo la importación y visualización los datos por medio de librerías como *Matplotlib* y *Seaborn*.

Inicialmente se realizó un análisis descriptivo de los atributos que conforman el *dataset* por medio de una función que se generó las principales medidas de centralidad y dispersión para los atributos numéricos; Mientras que para los atributos categóricos generó el listado de los valores presentes en el *dataset* y la cantidad de valores nulos.

```
def descripcionDatosDataset(datos):
    print("Cantidad de filas:", datos.shape[0])
    print("Cantidad de columnas:", datos.shape[1])
    print('-'*100)
    for columna in datos.columns:
        valoresDescripcion = ''
        tipo = ''
        if datos[columna].dtype == 'float64' or datos[columna].dtype == 'int64':
            tipo = 'numérico'
            valoresDescripcion = datos[columna].agg(['min', 'max', 'mean', 'std', 'median'])
        else:
            tipo = 'nominal' #categórico | string | no-numérico
            valoresDescripcion = {'valoresPresentes' : datos[columna].unique(),
                                  'cantidadNulos' : datos[columna].isna().sum()}
                                 #'conteoValores' : pd.value_counts(datos[columna])}
        print('Columna: ' + columna)
        print('Tipo de datos: ' + tipo)
        print('Descripción de valores:')
        if tipo == 'numérico':
            print(valoresDescripcion)
        else:
            print('-- Valores presentes (10 primeros): ' + str(valoresDescripcion['valoresPresentes'][:10]))
            pctNulos = (valoresDescripcion['cantidadNulos'] / datos.shape[0]) * 100
            print('-- Cantidad de nulos: ' + str(valoresDescripcion['cantidadNulos']) + ' = ' + "{0:.2f}".format(pctNulos) + '%')
        print('-'*100)
```

Ilustración 30. Función para generar medidas estadísticas. Elaboración: propia

Medidas de centralidad y dispersión

Se calcularon las medidas de centralidad y dispersión para cada uno de los atributos numéricos del reporte de gestión consolidada:

- ✓ Mínimo
- ✓ Máximo
- ✓ Media
- ✓ Desviación estándar
- ✓ Mediana

```
descripcionDatosDataset(Data)

Cantidad de filas: 1843
Cantidad de columnas: 19
-----------------------------------------------------------------------------------------------------------
Columna: Sucursal
Tipo de datos: numérico
Descripción de valores:
min       1.000000
max       6.000000
mean      3.504069
std       1.704997
median    4.000000
Name: Sucursal, dtype: float64
-----------------------------------------------------------------------------------------------------------
Columna: Cumplimiento
Tipo de datos: numérico
Descripción de valores:
min        0.000000
max       42.080000
mean      25.741085
std        8.455234
median    27.370000
Name: Cumplimiento , dtype: float64
-----------------------------------------------------------------------------------------------------------
Columna: Efectividad
Tipo de datos: numérico
Descripción de valores:
min       0.000000
max       1.000000
mean      0.503863
std       0.266240
median    0.510000
Name: Efectividad, dtype: float64
-----------------------------------------------------------------------------------------------------------
Columna: Tareas Medio
Tipo de datos: numérico
Descripción de valores:
min       0.000000
max       0.230000
mean      0.132100
std       0.057955
median    0.170000
Name: Tareas Medio, dtype: float64
-----------------------------------------------------------------------------------------------------------
Columna: Genero
Tipo de datos: nominal
Descripción de valores:
-- Valores presentes (10 primeros): ['M' 'F']
-- Cantidad de nulos: 0 = 0.00%
-----------------------------------------------------------------------------------------------------------
Columna: Cargo
Tipo de datos: nominal
Descripción de valores:
-- Valores presentes (10 primeros): ['Arquitecto de Software' 'Ingeniero de software II'
 'Especialista de Software' 'Desarrollador de software II'
 'Ingeniero de software I' 'Desarrollador de software I'
 'Analista de Software']
-- Cantidad de nulos: 0 = 0.00%
```

Ilustración 31. Medidas estadísticas. Elaboración: propia

Representación de datos

En este apartado fueron construidos una serie diversa de gráficos en herramientas como *Google Colab y Power BI*, y haciendo uso de librerías y técnicas vistas durante el master.

Los empleados de la empresa fueron categorizados en 3 niveles según el perfil laboral, experiencia y conocimiento técnico. En el gráfico de barras se observa que los empleados de nivel *Semi* y *Junior* son quienes concentran casi el 80% del personal.

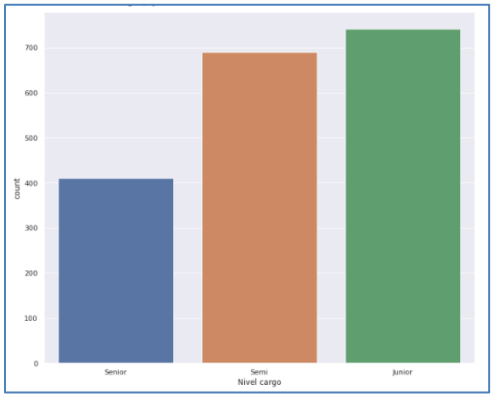


Ilustración 32. Distribución de nivel de cargos. Elaboración: propia

Los gráficos de anillo y barras representan la distribución del personal por modalidad de trabajo donde aproximadamente el 68% de los empleados aun trabaja desde alguna de las sedes de la empresa (ya sea bajo modalidad presencial o modalidad hibrida). Únicamente el 35% del personal cuenta con teletrabajo como modalidad de trabajo.

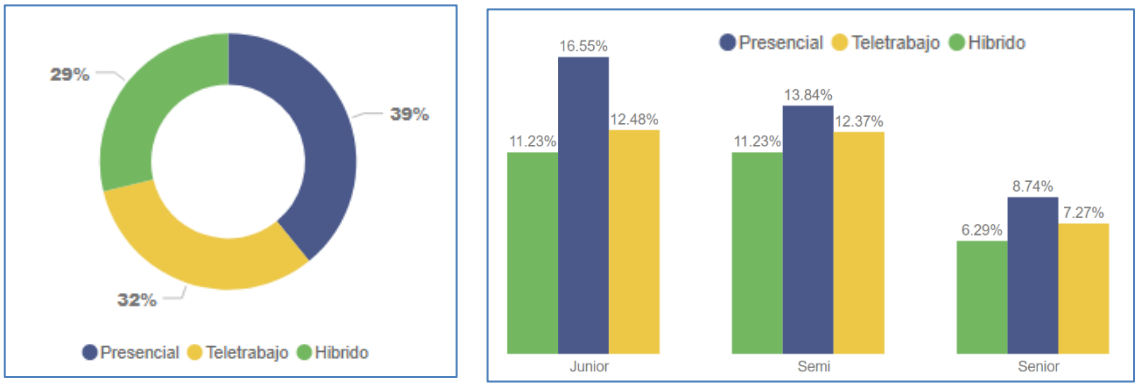


Ilustración 33. Distribución de modalidad de trabajo. Elaboración: propia

En este gráfico de torta se visualiza que el 52% de los colaboradores que fueron seleccionados para el análisis de desempeño son hombres, lo cual se refleja la equitativa participación que tienen las mujeres en esta empresa. En este estudio se evaluará si el género es un factor que determinante en el desempeño laboral.

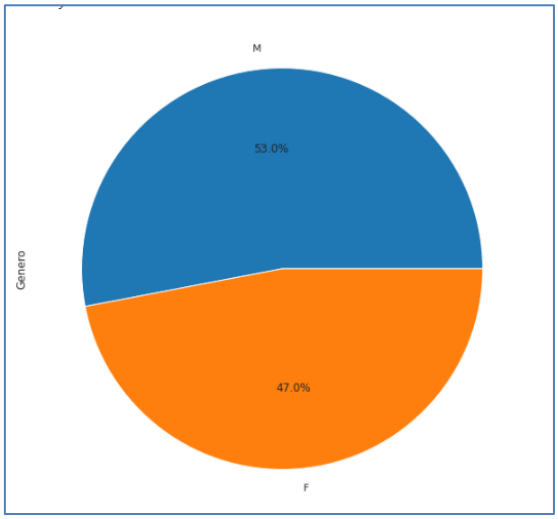


Ilustración 34. Distribución de género. Elaboración: propia

Por medio de un gráfico de georreferenciación ubicaron las ciudades de Europa en las que la empresa tienen sedes. El grafico de árbol representa la distribución del personal en estas ciudades, la cual se encuentra balanceada (aproximadamente 16% por ciudad)

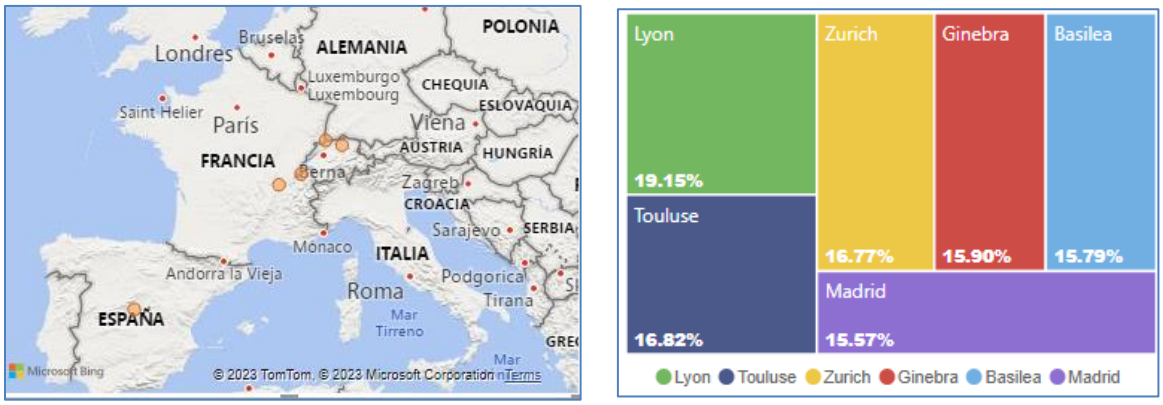


Ilustración 35. Distribución de ubicación por ciudades. Elaboración: propia

Por medio de un histograma se identifica que la mayoría de empleados cuenta con menores salarios mientras que un grupo reducido de empleados acapara los salarios más altos de la empresa, esto probablemente se deba a que son empleados que cuentan un nivel del cargo superior por sus conocimientos técnicos y experiencia.

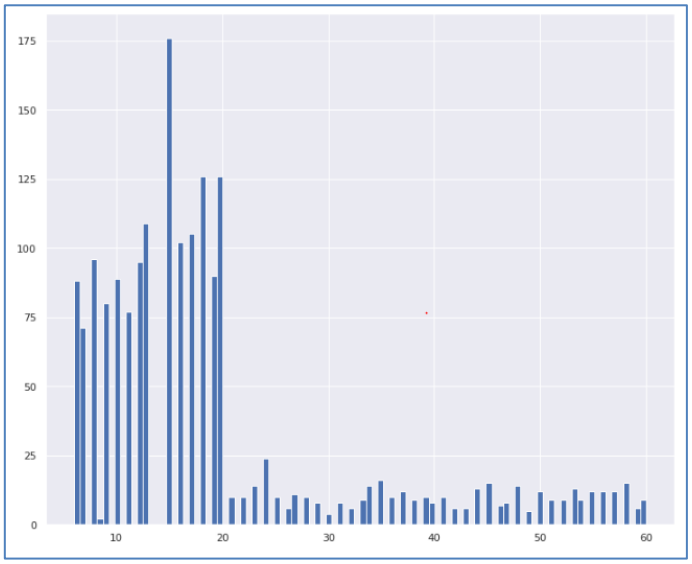


Ilustración 36. Distribución de salario. Elaboración: propia

Este gráfico de barras corresponde al salario promedio devengado por cada nivel de cargo, se observa una brecha considerable entre empleados de nivel *Senior* y *Semi*.

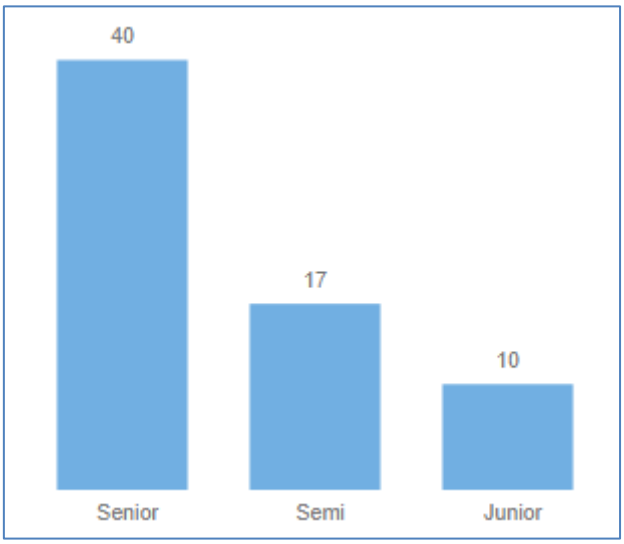


Ilustración 37. Promedio por salario. Elaboración: propia

Este histograma muestra que la mayoría del personal se encuentra entre 18 a 35 años.

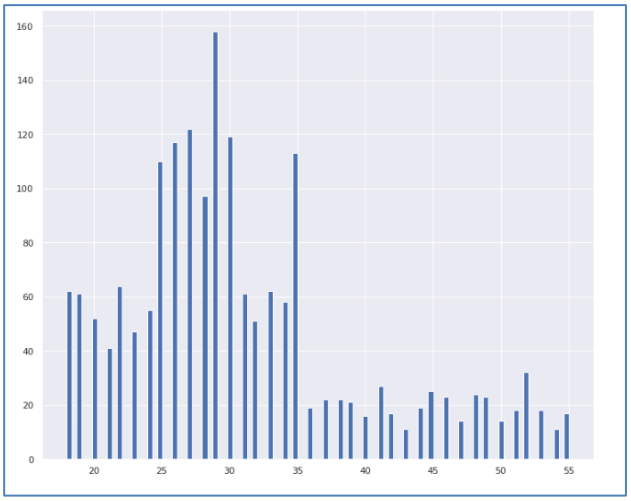


Ilustración 38. Distribución de edad. Elaboración: propia

Este gráfico representa la edad promedio de los empleados por nivel de cargo, lo cual confirma que el personal más joven se desempeña como empleado de nivel *Semi* y *Junior*, mientras que los empleados mayores a 35 años son de nivel *Senior*.

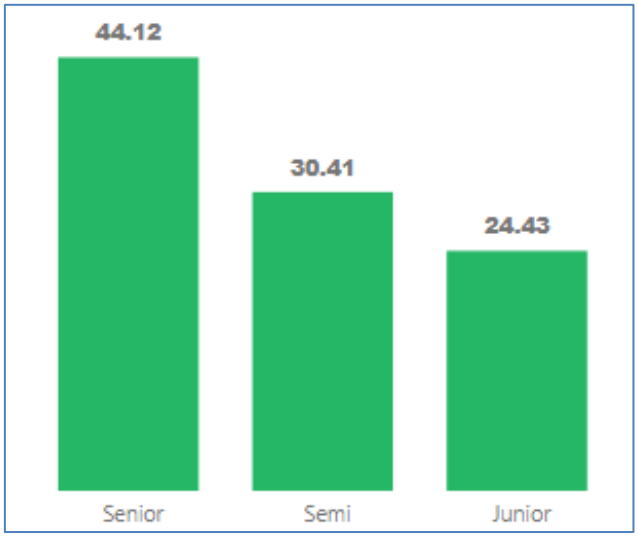


Ilustración 39. Promedio de edad. Elaboración: propia

El cumplimiento de las tareas ejecutadas por parte de los empleados es medido a través de metodología propia de la empresa, la cual va desde 0 a 45 puntos. Se identifica que el resultado se encuentra segmentado en dos grupos, en la que la mayoría de los empleados alcanzaron un resultado entre 15 a 45 puntos.

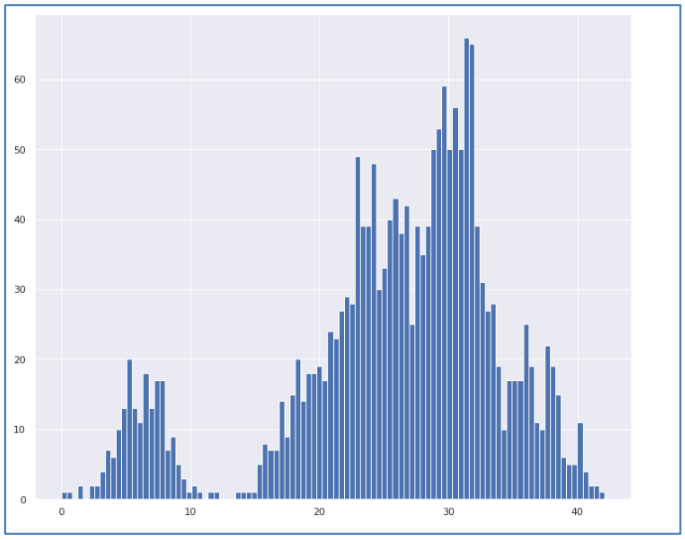


Ilustración 40. Distribución de cumplimiento. Elaboración: propia

La calidad de las tareas ejecutadas por parte de los empleados es medida a través de metodología propia de la empresa, la cual va desde 0 a 30 puntos. Se identifica que el resultado se encuentra segmentado en dos grupos, en la que la mayoría de los empleados alcanzaron un resultado entre 20 a 30 puntos.

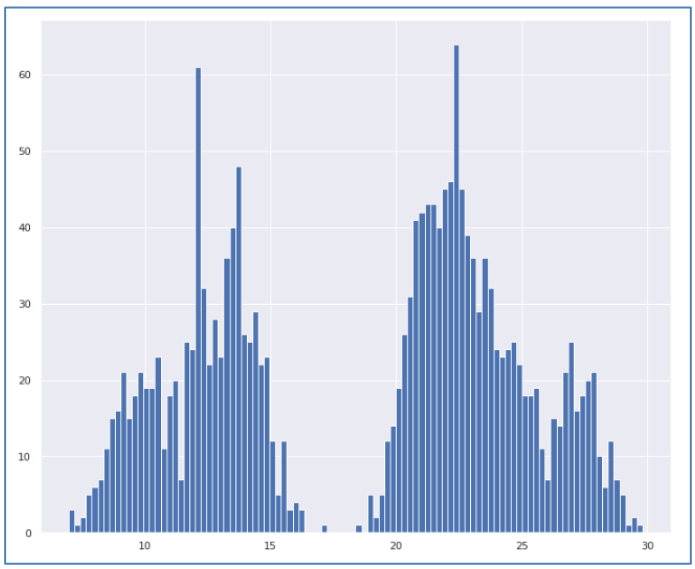


Ilustración 41. Distribución de calidad. Elaboración: propia

A través de gráficos de *boxplot* se validó la variación que existe en el cumplimiento y calidad de las tareas realizadas por los empleados de los niveles de cargo existentes. Se observo que las tareas ejecutadas por los empleados *Senior* presentan menor variación (principalmente en cuanto a al cumplimiento). Los resultados obtenidos para los empleados de nivel *Semi* y *Junior* son bastantes similares.

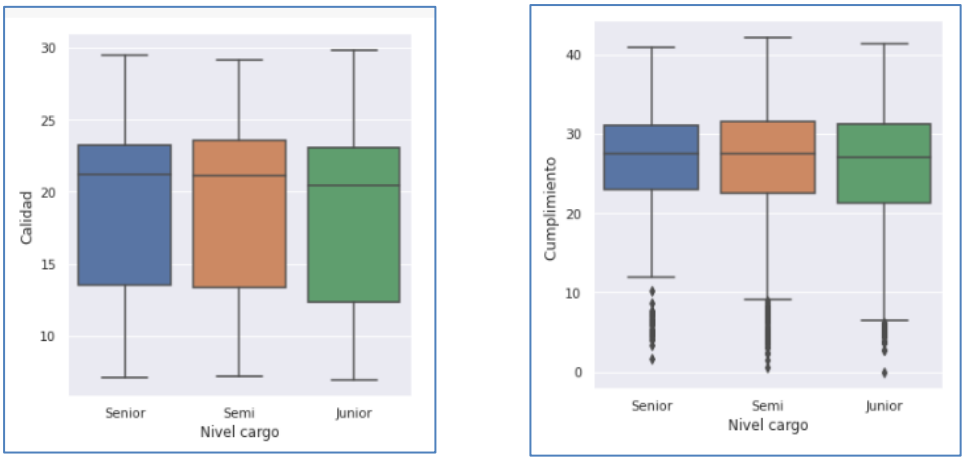


Ilustración 42. Boxplot de cumplimiento y calidad. Elaboración: propia

Este gráfico de dispersión representa la relación entre los atributos de Cumplimiento y Calidad, Se observan 5 segmentos generados en el gráfico, por lo cual se entiende que existe una correlación entre estos atributos.

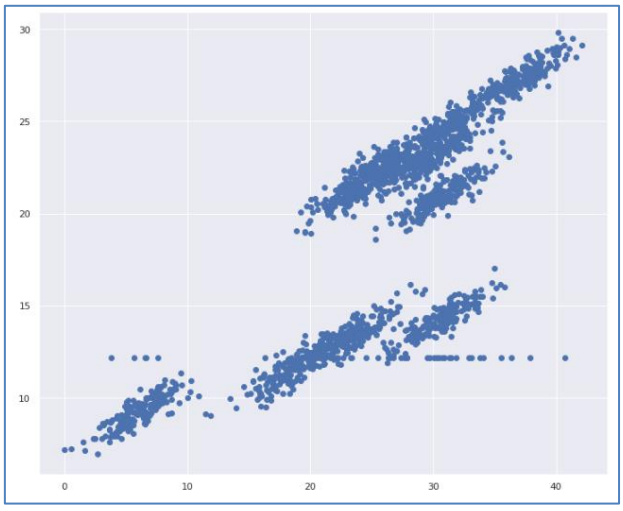


Ilustración 43. Gráfico de dispersión de cumplimiento y calidad. Elaboración: propia

Se genero una matriz de gráficos para conocer el nivel de satisfacción de las tareas que fueron ejecutadas por los empleados de diferentes géneros y niveles de cargo.

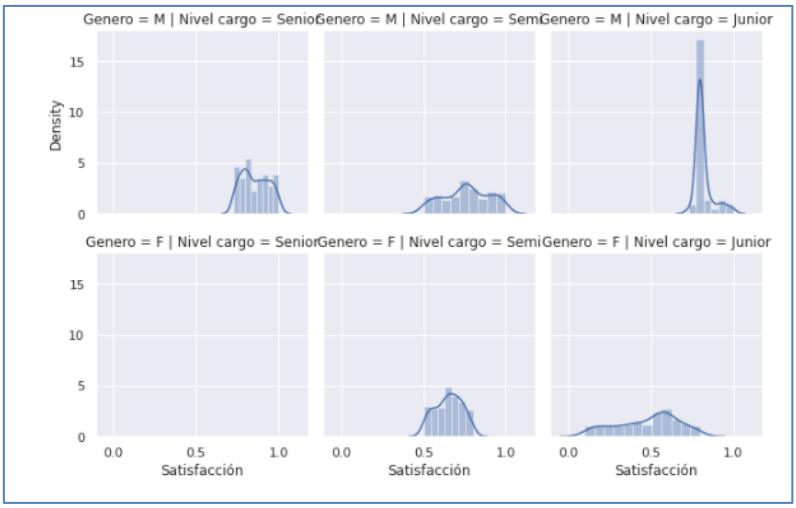


Ilustración 44. Distribución de satisfacción. Elaboración: propia

Se seleccionaron algunos atributos numéricos con los se generó una matriz de gráficos de dispersión para ver el comportamiento de estos.

```
Feat = ['Cumplimiento ', 'Efectividad', 'Calidad',
        'Satisfacción', 'Productividad', 'Trabajo en equipo']
```

Ilustración 45. Matriz de gráficos de dispersión. Elaboración: propia

Matriz de correlación

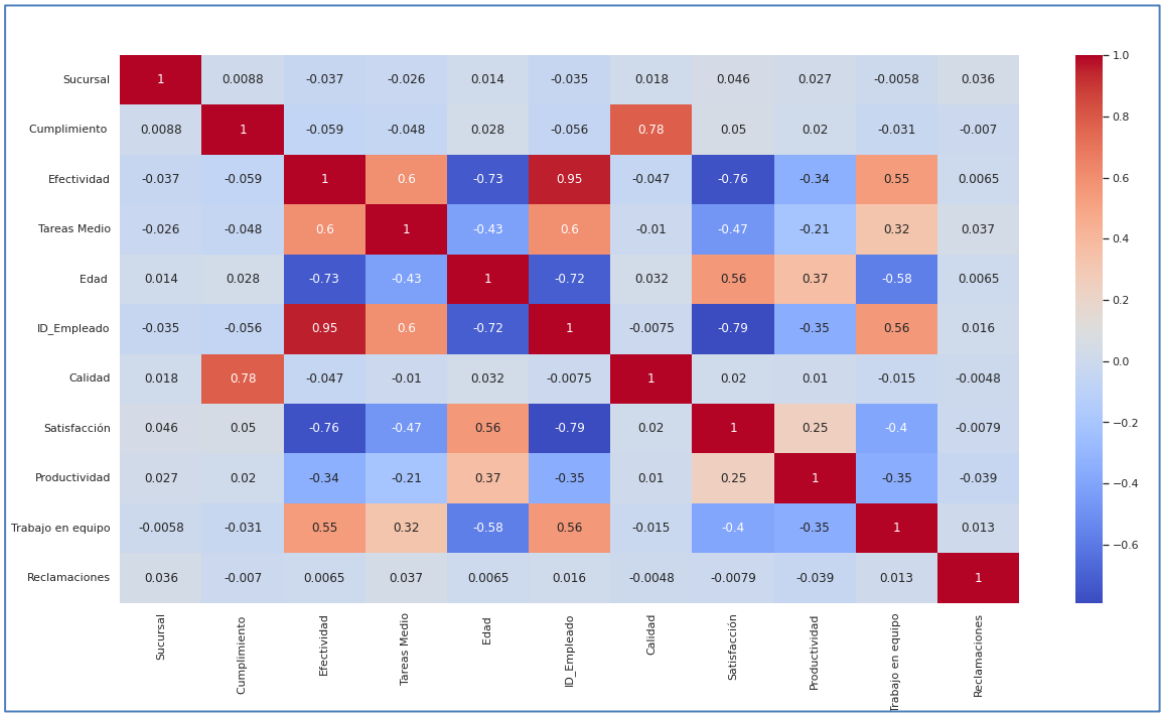


Ilustración 46. Matriz de correlación. Elaboración: propia

En este punto fue generada la matriz de correlación para conocer la relación estadística entre los atributos del dataset, donde se identificó que existe una correlación importante entre algunos atributos, a pesar que en la mayoría de los casos los coeficientes obtenidos no son valores muy altos, es decir que no se aproximaron a (1) o (-1).

Ningún atributo presento un coeficiente de correlación superior al 90% por lo cual se entiende que los datos se comportan de forma independiente y no existe dependencia entre estos.

Limpieza de datos (2)

Eliminación de variables con alta correlación

Según lo visto en el aparatado es necesario eliminar algunos de los atributos debido a que cuentan con una alta correlación

```
Data=Data.drop(['ID_Empleado'], axis=1)
```

Ilustración 47. Atributos correlacionados 2. Elaboración: propia

Eliminación de variables sin importancia

Adicionalmente bajo criterio experto se descartaron algunos atributos que no eran relevantes para el análisis

```
Data=Data.drop(columns=['Cargo', 'Modalidad', 'Proyecto', 'Ciudad', 'Pais'], axis=1)
```

Ilustración 48. Eliminación de variables. Elaboración: propia

Creación de variables dummies

Generó los atributos dummies para los atributos categóricos que aún quedan en el dataset, es decir "Genero" y "Nivel de Cargo".

```
Data = pd.get_dummies(Data)
```

Ilustración 49. Generación de variables dummies. Elaboración: propia

Selección de atributos para modelado

Los atributos seleccionados para construir el modelo de machine learning fueron:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1843 entries, 0 to 1842
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Sucursal            1843 non-null   int64
 1   Cumplimiento        1843 non-null   float64
 2   Efectividad         1843 non-null   float64
 3   Tareas Alto         1843 non-null   float64
 4   Tareas Medio        1843 non-null   float64
 5   Edad                1843 non-null   float64
 6    Salario            1843 non-null   float64
 7   ID_Empleado         1843 non-null   float64
 8   Calidad             1843 non-null   float64
 9   Satisfacción        1843 non-null   float64
 10  Productividad       1843 non-null   float64
 11  Trabajo en equipo   1843 non-null   float64
 12  Reclamaciones       1843 non-null   int64
 13  Genero_F            1843 non-null   uint8
 14  Genero_M            1843 non-null   uint8
 15  Nivel cargo_Junior  1843 non-null   uint8
 16  Nivel cargo_Semi    1843 non-null   uint8
 17  Nivel cargo_Senior  1843 non-null   uint8
dtypes: float64(11), int64(2), uint8(5)
memory usage: 275.1 KB
```

Ilustración 50. Atributos seleccionados para modelado. Elaboración: propia

4.3.4. Modelado de datos

Teniendo en cuenta el objetivo de este Trabajo de Fin de Master (TFM), fue construido un modelo de *machine learning* a partir de los atributos seleccionados en la etapa de minería de datos con el fin de analizar de forma integral la gestión realizada por los empleados durante el año 2022 para aportar valor a la organización, mediante la transformación de datos en conocimiento que permita optimizar la toma de decisiones de la empresa con respecto al personal.

Dado que nuestro conjunto de datos es limitado, hemos optado por categorizar a los empleados en tres niveles de rendimiento: bajo, medio y alto, según las directrices establecidas por la empresa. Si tuviéramos acceso a una cantidad mayor de observaciones y datos históricos, sería más apropiado emplear un algoritmo de regresión para predecir con mayor precisión el rendimiento de los empleados.

Sin embargo, debido a las limitaciones de las variables y la cantidad de datos en nuestro caso, no hemos podido aplicar este enfoque. En su lugar, hemos desarrollado un modelo de clasificación para predecir la categoría de rendimiento de los empleados.

Para lograr la clasificación de los empleados, primero fue necesario generar la agrupación de los mismos utilizando los atributos del dataset, por lo cual se realizó la *clusterización* de los datos con el fin de identificar los grupos o *clusters* a utilizar como categorías de desempeño.

4.3.4.1. Aplicación de modelo de Clusterizacion

Se definió utilizar un algoritmo de aprendizaje no supervisado ya que como se mencionó previamente no se contaba con las etiquetas correspondientes al desempeño de los empleados por lo que a través de un modelo de *clusterizacion* se pretende agrupar a los empleados de acuerdo a los datos correspondientes a la ejecución de tareas, resultados de proyectos trabajados en 2022 y características sociodemográficas del personal.

Se utilizo un modelo de *clusterización* ya que esta técnica de machine learning no supervisado es útil para identificar patrones y estructuras subyacentes en los datos, el cual divide un conjunto de datos en grupos de objetos similares, de manera que los objetos dentro de un *cluster* sean similares entre sí y diferentes de los objetos de otros clusters.

Método del codo

En primer lugar, fue necesario definir el número óptimo de *clústers* a trabajar para generar la agrupación de los datos, por lo cual se utilizó el método del codo por medio de las librerías *skleran y scipy.*

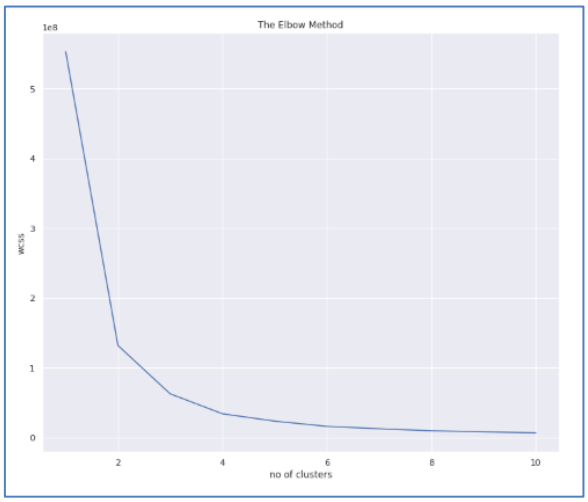


Ilustración 51. Método del codo. Elaboración: propia

Se identifico que se podrían utilizar entre 2 a 4 *clusters* como la mejor opción para realizar el análisis (preferiblemente 3 como lo indica el gráfico). De acuerdo a la definición del negocio se estableció trabajar con 3 *clusters*, es decir 3 categorías de desempeño, que serán alto, medio y bajo.

Análisis de componentes principales

Se realizo el análisis de componentes principales (*PCA*) con el fin de reducir la dimensionalidad del conjunto de datos, entender la estructura de los datos e identificar las variables más importantes para el modelado.

Se genero un gráfico para definir el número de componentes a utilizar. Los componentes principales se ordenan en función de su importancia en la explicación de la varianza de los datos.

El primer componente principal explica la mayor cantidad de varianza, el segundo explica la segunda mayor cantidad, y así sucesivamente.

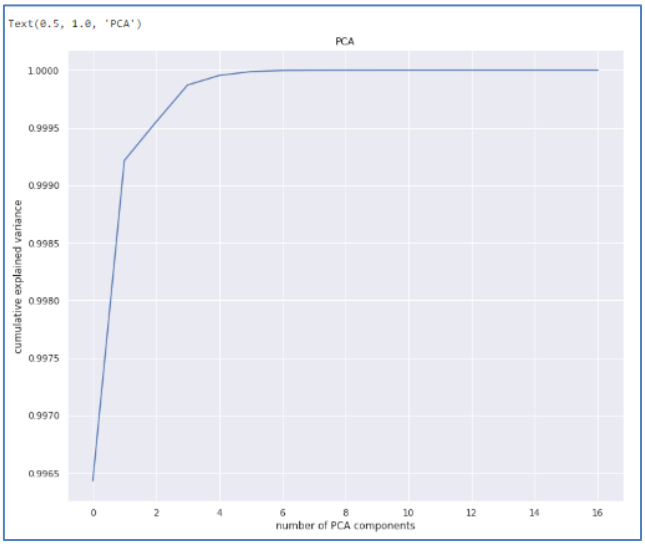


Ilustración 52. Análisis de Componentes Principales (PCA). Elaboración: propia

Como resultado de la prueba, se definió utilizar 3 componentes que representen el *dataset* consolidado. Posteriormente se estandarizaron los datos con la función "StandarScaler" de la biblioteca sklearn.

Se aplicó el Análisis de Componentes Principales (*PCA*) con los 3 componentes a utilizar "PC1", "PC2" y "PC3", por lo que se generó un nuevo *dataframe* denominado "df_pca".

```
pca = PCA(n_components=3)
pca.fit(data_scaled)
data_pca = pca.transform(data_scaled)
df_pca = pd.DataFrame(data=data_pca, columns=['PC1', 'PC2', 'PC3'])
```

df_pca

	PC1	PC2	PC3
0	4.768016	-2.263502	1.486591
1	3.562471	-0.647311	-1.278772
2	3.947472	-1.338151	-2.414168
3	4.173614	-1.446884	0.834894
4	4.047696	-2.092122	2.680291
...	...	...	...
1838	-3.897083	-0.664320	1.244209
1839	-3.365576	-0.809679	-0.485793
1840	-3.388124	-1.128123	2.854320
1841	-3.776245	-1.157594	-0.636037
1842	-3.362662	-1.111715	-0.260003

1843 rows × 3 columns

Ilustración 53. Componentes PCA. Elaboración: propia

Se genero la visualización de los 3 componentes en un gráfico *scatter* 3D.

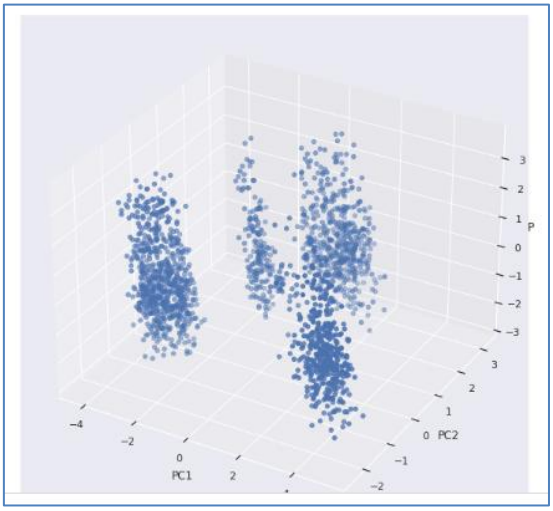


Ilustración 54. Componentes en grafico 3D. Elaboración: propia

Generación de Clusters

Para realizar la *clusterizacion* del dataframe con los 3 componentes definidos en la PCA se utilizaron 3 algoritmos diferentes:

- **Alglomerativo jerárquico:** Es un método que agrupa objetos en un conjunto de *clusters* jerárquicos de forma iterativa y progresiva en función de la distancia entre ellos. No requiere de un número previo de *clusters*, permite la identificación de *clusters* a diferentes niveles de granularidad y es fácil de interpretar

  El resultado del algoritmo es un dendrograma donde los objetos se agrupan en *clusters* grandes y estos se van subdividiendo en *clusters* más pequeños.

- **K-means:** Es un método que agrupa objetos en k *clusters*, donde k es un número predefinido de *clusters*. El objetivo del algoritmo *k-means* es minimizar la suma de las distancias cuadráticas entre cada objeto y su centroide asignado, lo que se conoce como la función de costo del algoritmo. Permite la identificación de outliers y es escalable.

- **DBSCAN:** Es un método basado en la densidad que puede manejar *clusters* de diferentes formas y tamaños, es resistente a los *outliers* y no requiere un número previo de *clusters*. Esto lo hace adecuado para una amplia variedad de aplicaciones de clustering, especialmente en conjuntos de datos donde la densidad varía en diferentes regiones del espacio de características.

Modelo Alglomerativo jerárquico

Se importaron la librerías y módulos requeridos, A continuación, se realizó el *clustering* de los datos utilizando la métrica de distancia “euclidea” con el método *average* que evalúa la disimilitud media.

La función *linkage* utiliza el método y métrica para calcular las distancias entre los clusters y en cada iteración mezclará los dos *clusters* con la distancia más pequeña de acuerdo al método y distancia elegidos. Este es el dendrograma que se generó:

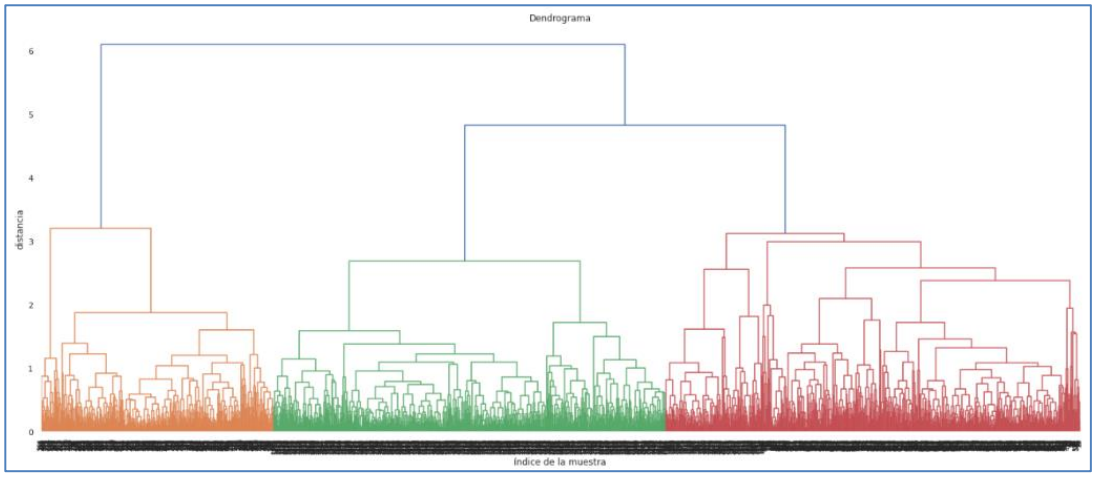


Ilustración 55. Dendrograma. Elaboración: propia

Valido en qué punto la distancia existente entre los 3 *clusters* es menor y finalmente utilizo la métrica de silueta para evaluar la calidad de la agrupación generada.

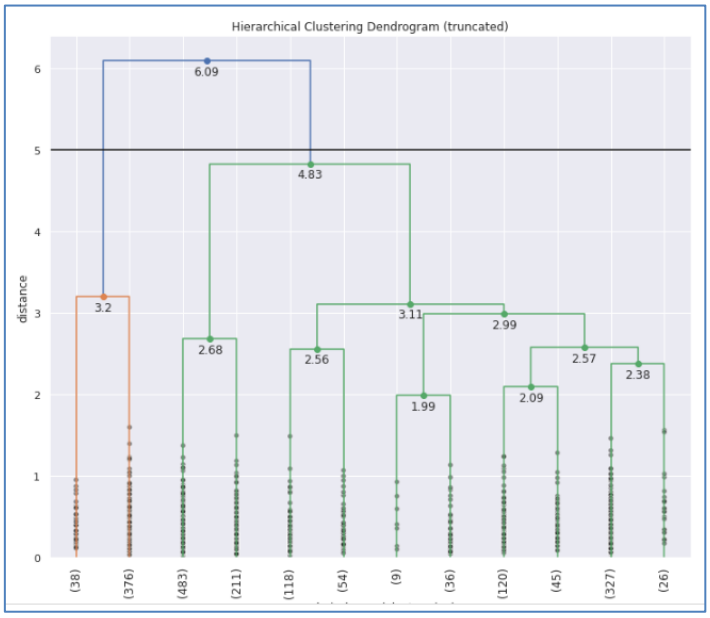


Ilustración 56. Dendrograma truncado. Elaboración: propia

Utilizo la métrica de Silueta para evaluar la calidad de la agrupación de los datos.

```
Z = linkage(df_pca, method='average')
cluster_labels = fcluster(Z, 3, criterion='maxclust')
silhouette_avg = silhouette_score(df_pca, cluster_labels)
print("El coeficiente de silueta es silhouette_avg:", silhouette_avg)

El coeficiente de silueta es silhouette_avg: 0.5725551130440214
```

Ilustración 57. Métrica de Silueta. Elaboración: propia

K- means

Antes de utilizar el algoritmo, es necesario importar las librerías y módulos requeridos para trabajar. Después se instancio el modelo y se clusterizarón los datos usando el algoritmo *K-means*. Visualizo el resultado obtenido con los parámetros asignados.

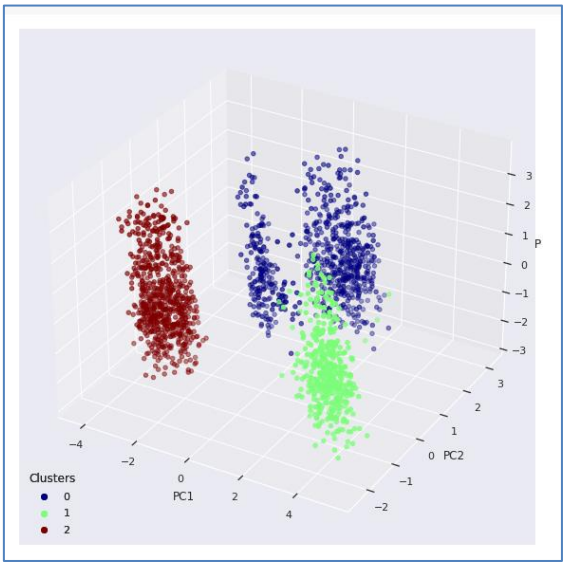


Ilustración 58. Clusterizacion con K-means. Elaboración: propia

El resultado considero que es bastante aceptable teniendo en cuenta que el algoritmo *K-means* está diseñado para trabajar con la distancia Euclidea, no lidia bien con datos con una covarianza alta.

Utilizo la métrica de Silueta para evaluar la calidad de la agrupación de los datos.

```
silhouette_avg = silhouette_score(df_pca, cluster_labels)
print("El coeficiente de silueta promedio es :", silhouette_avg)

El coeficiente de silueta promedio es : 0.5736273566937549
```

Ilustración 59. Métrica de Silueta. Elaboración: propia

DBSCAN

Importo las librerías y módulos requeridos, después se instancio el modelo y se clusterizarón los datos utilizando el algoritmo *DBSCAN* con los siguientes parámetros:

```
dbscan = DBSCAN (eps= 1.02, min_samples= 8)
dbscan.fit(df_pca[['PC1', 'PC2', 'PC3']])
df_pca['Cluster'] = dbscan.labels_
```

Ilustración 60. Parámetros DBSCAN. Elaboración: propia

Visualizo el resultado obtenido de la clusterizacion con los parámetros asignados.

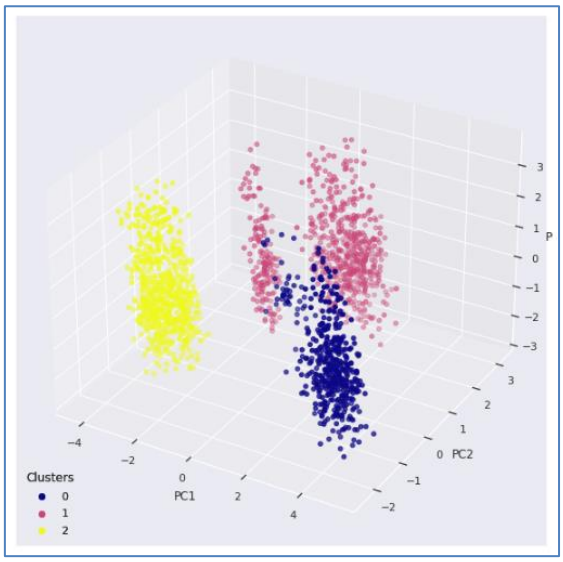


Ilustración 61. Clusterizacion con DBSCAN. Elaboración: propia

Utilizo la métrica de Silueta para evaluar la calidad de la agrupación de los datos.

```
silhouette_avg = silhouette_score(df_pca, cluster_labels)
print("El coeficiente de silueta promedio es :", silhouette_avg)

El coeficiente de silueta promedio es : 0.45599316100171017
```

Ilustración 62. Métrica de Silueta. Elaboración: propia

Evaluación de clusterizacion

Con el fin de identificar el algoritmo que genero la mejor clusterizacion se utilizó la métrica de silueta, la cual es una medida de calidad de *clustering* que evalúa qué tan bien están agrupados los datos. El valor de la métrica de silueta varía entre -1 y 1, donde un valor más cercano a 1 indica que los datos están bien agrupados y un valor cercano a -1 indica que los datos podrían estar mejor asignados a otro grupo. En general, un valor de silueta mayor a 0.5 se considera un buen resultado

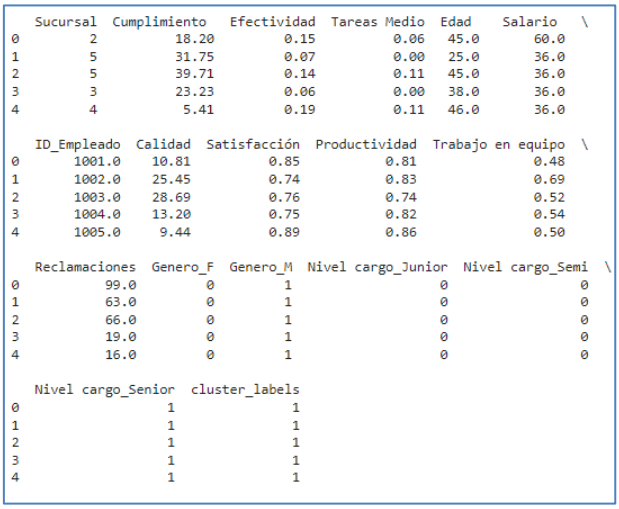
Esta tabla detalla el resultado obtenido al evaluar la métrica de silueta en cada modelo:

Modelo	Resultado de Silueta
Alglomerativo Jerárquico	0,5725
K-means	0,5736
DBSCAN	0,4559

Tabla 1. Comparación métrica de silueta. Elaboración propia

De acuerdo a los resultados obtenidos con el coeficiente de silueta, se definió utilizar la clusterizacion realizada con el método *K-Means*. Por lo cual se procedió a integrar el resultado de la clusterizacion de los datos como un nuevo atributo en el *dataset*.

Este proceso se realiza con la finalidad de continuar con la evaluación de desempeño por medio de un algoritmo de aprendizaje supervisado utilizando un modelo de clasificación en el que los empleados serán categorizados en las 3 etiquetas definidas previamente correspondientes a los 3 niveles de desempeño: Alto, Medio y Bajo



	Sucursal	Cumplimiento	Efectividad	Tareas Medio	Edad	Salario	\
0	2	18.20	0.15	0.06	45.0	60.0	
1	5	31.75	0.07	0.00	25.0	36.0	
2	5	39.71	0.14	0.11	45.0	36.0	
3	3	23.23	0.06	0.00	38.0	36.0	
4	4	5.41	0.19	0.11	46.0	36.0	

	ID_Empleado	Calidad	Satisfacción	Productividad	Trabajo en equipo	\
0	1001.0	10.81	0.85	0.81	0.48	
1	1002.0	25.45	0.74	0.83	0.69	
2	1003.0	28.69	0.76	0.74	0.52	
3	1004.0	13.20	0.75	0.82	0.54	
4	1005.0	9.44	0.89	0.86	0.50	

	Reclamaciones	Genero_F	Genero_M	Nivel cargo_Junior	Nivel cargo_Semi	\
0	99.0	0	1	0	0	
1	63.0	0	1	0	0	
2	66.0	0	1	0	0	
3	19.0	0	1	0	0	
4	16.0	0	1	0	0	

	Nivel cargo_Senior	cluster_labels
0	1	1
1	1	1
2	1	1
3	1	1
4	1	1

Ilustración 63. Dataset clusterizado. Elaboración: propia

Adicionalmente fue necesario asegurar que el atributo 'cluster_labels' se encontrara como un atributo de tipo numérico con el fin de evitar inconvenientes posteriores al momento de la clasificación.

Por medio de este grafico se validó la correlación entre los atributos del *dataset* con el atributo *target*, en este caso con el atributo “Cluster_labels”.

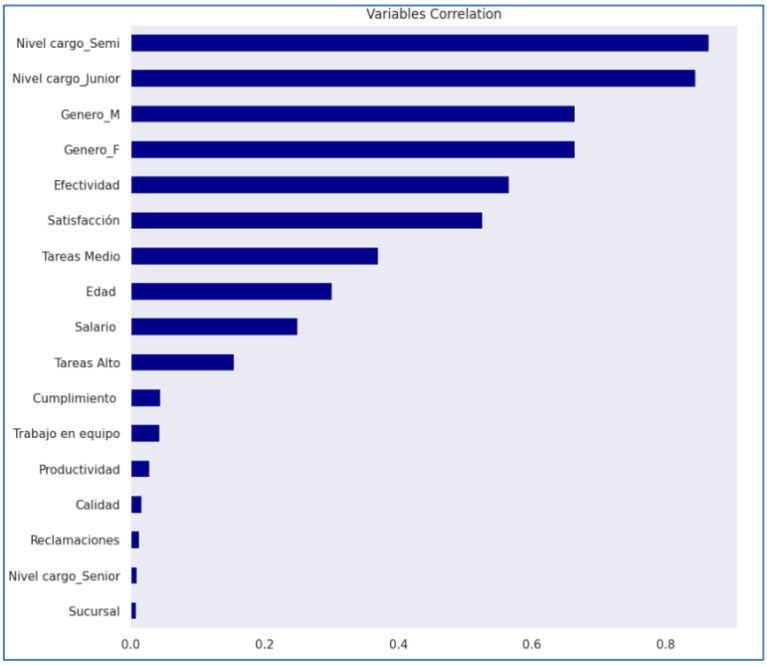


Ilustración 64. Variable que impactan en el desempeño. Elaboración: propia

4.3.4.2. Aplicación de modelo de Clasificación

En este punto se generó la clasificación de los empleados, la cual es una técnica de machine learning supervisado que sirve para predecir la etiqueta o clase de un objeto en función del conjunto de variables predictoras, en este caso de utilizo para predecir la categoría de desempeño correspondiente a los empleados con base en los atributos seleccionados, utilizados como. Esta técnica permite predecir y clasificar datos de manera automática y eficiente.

Para realizar la clasificación de los empleados fueron utilizados diferentes algoritmos, con el fin de encontrar el que generara los mejores resultados con los datos disponibles. A continuación, se describen los algoritmos utilizados:

- **Árbol de Decisión:** Es un modelo de *machine learning* que se utiliza para clasificar datos en diferentes categorías o clases. En el modelo de Árbol de Decisión de clasificación, cada nodo del árbol representa una pregunta que se hace sobre las características de los datos. A medida que se sigue avanzando por el árbol, se van haciendo más preguntas hasta llegar a una hoja del árbol, que representa la clase o categoría a la que pertenece el dato**.**

- **Random Forest:** Es un modelo de *machine learning* que utiliza un conjunto de árboles de decisión para realizar una tarea de clasificación o regresión. En este modelo, cada árbol es construido de manera independiente utilizando un subconjunto aleatorio de las características del conjunto de datos original. El proceso de construcción del modelo *Random Forest* se basa en dos etapas. En la primera etapa, se genera una muestra aleatoria de los datos de entrenamiento y se utiliza para construir un conjunto de árboles de decisión. En la segunda etapa, se utiliza el conjunto de árboles para realizar la tarea de clasificación o regresión.

- **Gradient boosting:** Es un modelo de *machine learning* utilizado en problemas de regresión y clasificación que combina múltiples árboles de decisión para crear un modelo de predicción más preciso y robusto. Se basa en la idea de ajustar iterativamente una serie de modelos débiles (en este caso, árboles de decisión) a los errores residuales del modelo anterior. De esta manera, cada modelo adicional se enfoca en corregir los errores que quedaron sin corregir por el modelo anterior, hasta que se alcanza un nivel de precisión deseado.

- **Regresión Logística:** Es un modelo de *machine learning* utilizado para predecir una variable categórica binaria o múltiple. El modelo utiliza una función logística para transformar la variable de entrada y predecir la probabilidad de que la variable de salida pertenezca a una de las categorías. Este modelo es ajustado a los datos mediante la estimación de los coeficientes de la función logística y se utiliza para predecir la variable de salida para nuevos datos.

- **K-vecinos más próximos (KNN):** Es un modelo de *machine learning* utilizado para tareas de clasificación y regresión. El modelo funciona encontrando los k ejemplos más cercanos en el conjunto de datos de entrenamiento al nuevo ejemplo utilizando una medida de distancia, y determinando la clase o valor numérico basándose en las clases o valores de los vecinos más cercanos. K es un hiperparámetro importante en KNN que determina el número de vecinos más cercanos utilizados para la predicción.

- **Máquinas de vectores de soporte (SVM):** Son un modelo de *machine learning* utilizado para tareas de clasificación y regresión. El modelo funciona encontrando un hiperplano que maximiza la separación entre las clases en el espacio de características. Si los datos no son linealmente separables, se utiliza una técnica de kernel para transformar los datos. SVM es un modelo paramétrico y de margen máximo, y es popular debido a su capacidad para manejar conjuntos de datos complejos y no lineales.

Se utilizaron diversos modelos, comenzando por los más simples progresando hacia modelos más complejos. Esto debido a que, al utilizar un modelo básico inicial, podemos obtener una idea general de cómo deberían ser los resultados y comparar los beneficios y desventajas de cada uno a medida que aumenta la complejidad de los modelos.

Partición de conjunto de training y test

Para entrenar el modelo y realizar la predicción de categoría de desempeño de los empleados, procedemos a separar el conjunto de datos en dos partes, una parte para el entrenamiento de la modelo denominada *training* y otra parte para evaluar la modelo denominada *test*. Para este caso se particiono el conjunto de datos en 75 - 25.

En esta separación de los datos fue necesario eliminar el atributo “cluster_labels” ya que ese es el atributo que se buscaba predecir con el modelo.

```
from sklearn.model_selection import train_test_split
train_X, test_X, train_y, test_y = train_test_split(
                                        Data_clustered.drop(columns = ["cluster_labels"] ),
                                        Data_clustered["cluster_labels"] ,
                                        test_size=0.25,
                                        random_state = 123)
```

Ilustración 65. Partición del conjunto de datos. Elaboración: propia

Adicionalmente fueron reseteados los indicies de los conjuntos de datos correspondientes a *training* y *test* (train_X, test_X, train_y, test_y) antes de iniciar la clasificación.

Modelos a utilizar

Fueron importados e instanciados todos los modelos de clasificación a utilizar. En la grafica se ven algunos de los modelos utilizados.

```
from sklearn.tree import DecisionTreeClassifier   # Árbol de decisión
arbol = DecisionTreeClassifier()


from sklearn.ensemble import RandomForestClassifier   # Random forest
random_forest = RandomForestClassifier()

from sklearn.ensemble import GradientBoostingClassifier  # Gradient Boosting
gradient_boosting = GradientBoostingClassifier()
```

Ilustración 66. Modelos de clasificación instanciados. Elaboración: propia

Parrilla de Hiperparámetros

Para generar los mejores resultados posibles, se construyó una parrilla de hiperparámetros de tal manera que se pudiera iterar de forma automática cada modelo instanciado con dichos parámetros hasta obtener los mejores resultados posibles.

```
grid_arbol = {"max_depth":list(range(1,11))}# Profundidades de 1 a 10

grid_random_forest = {"n_estimators": [150],

                      "max_depth": [3,5,10,15,20],

                      "max_features": ["sqrt", 3, 4]}

grid_gradient_boosting = {"loss": ["deviance"],
                          "learning_rate": [0.05, 0.1, 0.5], # Una learning_rate alta
                                                             # junto con n_estimators alta puede dar sobreajuste.

                          "n_estimators": [20,50,100,200], # En GBT un número
                                                           # elevado de árboles puede darnos sobreajuste.

                          "max_depth": [1,2,3,4,5], # En boosting, los árboles utilizados deben tener poca
                                                    # profundidad ya que van rectificandose poco a poco de forma aditiva.

                          "subsample": [1.0, 0.8, 0.5], # Lo usamos para evitar el sobreentrenamiento

                          "max_features": ["sqrt", 3, 4], }
```

Ilustración 67. Parrilla de hiperparámetros. Elaboración: propia

Gridsearch

Se importo la clase *GridSearchCV* de la biblioteca *sklearn* para realizar la iteración de los modelos de clasificación mediante la búsqueda exhaustiva de valores de hiperparámetros contenidos en la parrilla previamente construida.

```
from sklearn.model_selection import GridSearchCV
```

Ilustración 68. GridSearch. Elaboración: propia

```
gs_arbol = GridSearchCV(arbol,
                        grid_arbol,
                        cv=10,
                        scoring="f1_micro",
                        verbose=1,
                        n_jobs=-1)

gs_random_forest = GridSearchCV(random_forest,
                                grid_random_forest,
                                cv=10,
                                scoring='f1_micro',
                                verbose=1,
                                n_jobs=-1)

gs_gradient_boosting = GridSearchCV(gradient_boosting,
                                    grid_gradient_boosting,
                                    cv=10,
                                    scoring='f1_micro',
                                    verbose=1,
                                    n_jobs=-1)
```

Ilustración 69. GridSearch de modelos de clasificación. Elaboración: propia

Se genero un diccionario con todos los modelos instanciados y sus respectivos *GridSearch* como clave.

```
todos_los_grid_searchs = {"gs_arbol":gs_arbol,
                          "gs_random_forest":gs_random_forest,
                         "gs_svm":gs_svm,
```

Ilustración 70. Diccionario con modelos de clasificación. Elaboración: propia

K- Folds

Para evitar que nuestro modelo se ajuste demasiado a los datos de *training* y pierda la capacidad de generalizar para nuevos datos con características ligeramente diferentes, se utilizó el algoritmo de validación cruzada *k-folds*, que divide el conjunto de datos en k partes y entrena el modelo con una parte mientras lo prueba con las otras k-1 partes.

Por medio de un ciclo *for* se realizó la iteración de los modelos contenidos en el diccionario teniendo en cuenta los hiperparámetros asignados y *k-folds*. Cada modelo se ajusta al conjunto de entrenamiento y devuelve los mejores valores encontrados.

```
for nombre, grid_search in todos_los_grid_searchs.items():
    print("Haciendo Grid Search de %s..." % nombre)
    grid_search.fit(train_X, train_y)
```

Ilustración 71. Iteración de modelos con GridSearch y K-folds. Elaboración: propia

Se genero una tupla con el mejor resultado obtenido por medio del *GridSearch* para cada modelo de clasificación.

```
[('gs_arbol', 0.9956573871337714),
 ('gs_random_forest', 0.9978260869565216),
 ('gs_svm', 0.9956521739130435)]
```

Ilustración 72. Tupla con mejores resultados de GridSearch. Elaboración: propia

Los resultados obtenidos se ordenaron en un *dataframe* nuevo.

	GridSearchCV	Mejor score
1	gs_random_forest	0.997826
0	gs_arbol	0.995657
2	gs_svm	0.995652

Ilustración 73. Dataframe mejores resultados con GridSearch. Elaboración: propia

Se selecciono el objeto del diccionario "todos_los_grid_searchs" correspondiente al algoritmo de *Random Forest* y se le aplica el mejor estimador:

```
mejor_gridsearch_clas = todos_los_grid_searchs["gs_random_forest"]
```

```
mejor_pipeline = mejor_gridsearch_clas.best_estimator_
mejor_pipeline
```

```
RandomForestClassifier
RandomForestClassifier(max_depth=5, max_features=3, n_estimators=150)
```

Ilustración 74. Mejor estimador con Random Forest. Elaboración: propia

Evaluación de clasificación

Se utilizaron las métricas *F1-score* y *Acuraccy* para evaluar el modelo de clasificación. Se opto por estas métricas debido a que el *Acuraccy* mide la proporción de predicciones correctas sobre el total de predicciones, mientras que el *F1-score* combina el *Acuraccy* y el *Recall* que mide la proporción de instancias positivas que fueron identificadas correctamente para para encontrar un equilibrio entre ambos, al identificar los verdaderos positivos como los falsos positivos y falsos negativos.

```
from sklearn.metrics import f1_score
f1_en_test = f1_score(y_true = test_y, y_pred = mejor_pipeline.predict(test_X), average="micro")
print("El modelo tiene un f1 en el conjunto de test de %s" % f1_en_test)
```

```
El modelo tiene un f1 en el conjunto de test de 0.9978308026030369
```

Ilustración 75. Evaluación de F1-Score. Elaboración: propia

```
from sklearn.metrics import f1_score
f1_en_test = f1_score(y_true = test_y, y_pred = mejor_pipeline.predict(test_X), average="micro")
print("El modelo tiene un f1 en el conjunto de test de %s" % f1_en_test)
```

```
El modelo tiene un f1 en el conjunto de test de 0.9978308026030369
```

Ilustración 76. Evaluación de Acuraccy. Elaboración: propia

Esta tabla contiene el resultado de las métricas evaluadas en el modelo:

Modelo Random Forest	Resultado
F1-Score	0,997830
Acuraccy	0.997830

Tabla 2. Resultados de métricas F1 Score y Acuraccy. Elaboración propia

Se genero la matriz de confusión del modelo (Ilustración 77), en la que se pueden observar de forma más clara los resultados, ya que esta muestra cuántas observaciones se clasificaron correctamente y cuántas se clasificaron incorrectamente.

La matriz de confusión se organiza en una tabla que tiene cuatro entradas:

- Verdaderos positivos (TP): registros que se clasificaron correctamente como positivas.
- Falsos positivos (FP): observaciones que se clasificaron incorrectamente como positivas.
- Verdaderos negativos (TN): observaciones que se clasificaron correctamente como negativas.
- Falsos negativos (FN): observaciones que se clasificaron incorrectamente como negativas.

Se observa que solo hay un caso en el que el valor real era 1 y se ha predicho como 2.

En cambio, podemos ver en la diagonal de la matriz de confusión, que en la mayoría de los casos el modelo acierta y predice el valor correcto.

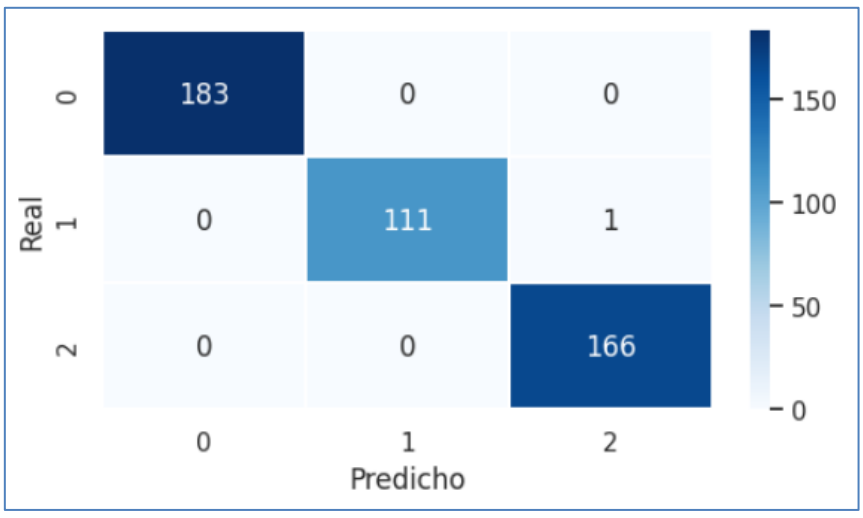


Ilustración 77. Matriz de confusión de Random forest. Elaboración: propia

4.4. Resultados

Una vez realizada la predicción de la categoría de desempeño para cada empleado se procedió a realizar la exploración y análisis de los resultados obtenidos.

Por lo cual inicialmente fue requerido generar un nuevo atributo denominado “desempeño” el cual comparaba el atributo “cluster_labels” con el valor existente (0,1,2). Si el valor coincidía con uno de estos, el valor devuelto era (Bajo, Medio o Alto).

El atributo “desempeño” corresponde a una escala cualitativa tipo semáforo definida por el negocio previamente, la cual es transversal dentro del estudio, es decir aplica para todos los empleados sin importar sucursal, cargo, edad, salario, nivel, etc.

A continuación, se menciona el significado de cada una de las categorías de la escala:

Nivel	Descripción	
Alto	Se refiere a un empleado que supera consistentemente las expectativas de su trabajo y realiza sus tareas de manera eficiente y efectiva. Este nivel está reservado para empleados que sobresalen en su trabajo y que son un valor agregado para la empresa.	
Medio	Se refiere a un empleado que cumple con las expectativas y requisitos de su trabajo, pero no necesariamente sobresale en su desempeño. Este nivel se aplica a empleados que hacen un buen trabajo, pero no destacan en su trabajo.	
Bajo	Se refiere a un empleado que no cumple con las expectativas y requisitos de su trabajo, y cuyo desempeño es insuficiente. Este nivel se aplica a empleados que no realizan sus tareas de manera efectiva y que necesitan mejorar su desempeño para cumplir con los requisitos de su trabajo.	

Tabla 3. Escala cualitativa de desempeño. Elaboración: propia

4.4.1. Visualización de Resultados

A continuación, se presentan los resultados obtenidos por medio de una serie de gráficos de un tablero de control o *dashboard* construido en *Power BI*, con la finalidad de visualizar de manera más clara los datos e identificar patrones, tendencias y oportunidades que permitan tomar decisiones informadas y basadas en datos.

Adicionalmente se pretende identificar cuáles son los atributos que tuvieron un mayor impacto dentro del desempeño de los empleados, establecer comparaciones de rendimiento y determinar su impacto para el estudio.

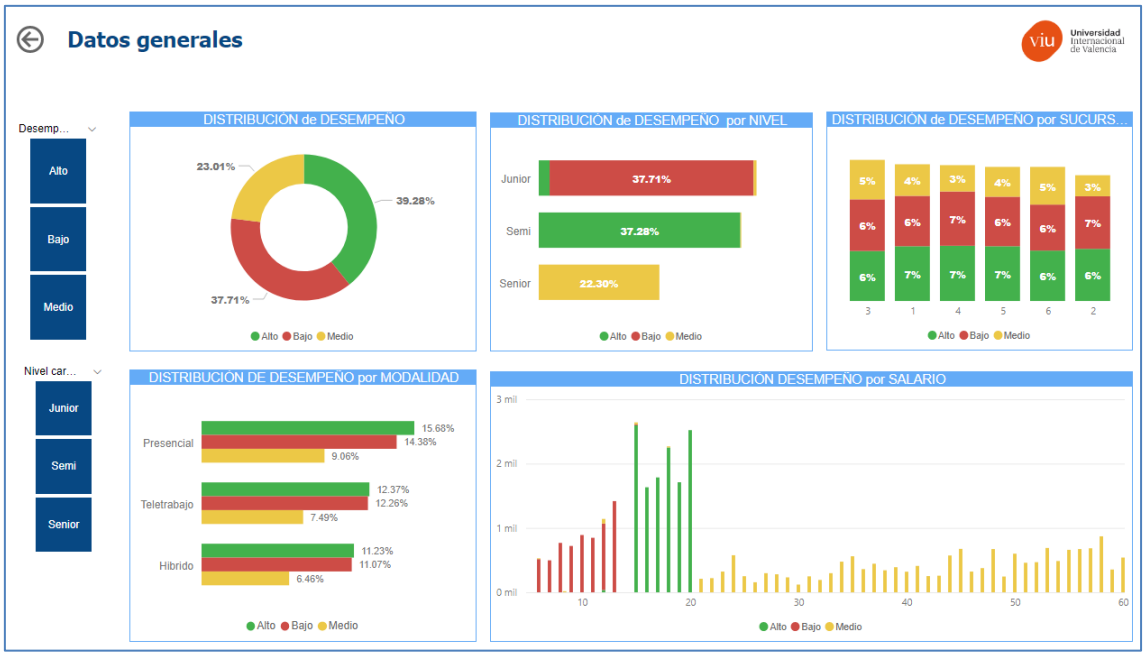


Ilustración 78. Dashboard de resultado en Power BI. Elaboración: propia

A través de los gráficos se puede visualizar la representación de la relación entre el desempeño de los empleados y otras variables como salario, nivel de cargo, modalidad de trabajo, sede, genero, etc. para ello se utilizaron gráficas de barras, anillo, histogramas ya que como lo menciona Casanova (2019) estas gráficas son de fácil comprensión y son adecuadas para la comparación de magnitudes entre varios elementos.

Se observa la categorización de los empleados de acuerdo a la escala cualitativa tipo semáforo definida previamente (Tabla 3) en la que se asigna color rojo a los empleados con desempeño bajo, color amarillo a los empleados con desempeño medio y color verde a los empleados que se presentaron desempeño alto de acuerdo con el análisis realizado sobre la gestión desarrollada por el personal en el año 2022.

4.1.2. Análisis de Resultados

Se ha podido concluir que el salario y el nivel de cargo no son variables determinantes que aseguren que los empleados tengan un alto desempeño, ya que a pesar de que los empleados de nivel *Senior* son el gasto más representativo para la empresa en cuanto a nómina no presentan el desempeño esperado, siendo en su gran mayoría categorizados con un desempeño de nivel medio. Al principio del estudio se pensaba que a mayor salario mejor sería el desempeño de los empleados y proporcionalmente mayor sería el aporte realizado a la empresa, lo cual quedo completamente desmentido.

Por otro lado, se identificó que la efectividad en las tareas realizadas y la satisfacción correspondiente a las tareas entregadas en los proyectos fueron variables que si afectaron directamente en el desempeño de los empleados.

Se debe aumentar la participación de las mujeres en cargos *Senior* ya que son ellas quienes aportan en mayor medida a la empresa. Por lo cual se propone implementar una política de equidad que le permitan al personal femenino ganar participación en los cargos de mayor jerarquía y a contar con una mayor visibilidad del trabajo que realizan.

Se debe validar la posibilidad de que la empresa tenga una reestructuración en su modelo organizacional ya que al analizar el rendimiento del personal se identificó:

La mayoría de empleados de nivel *Junior* (90%) no están aportando el valor esperado a la empresa, al ser estos quienes presentan el peor desempeño de todo el personal, por lo cual se propone se revise la continuidad de todos los empleados categorizados con desempeño bajo.

Respecto a los empleados categorizados con desempeño medio (23%) se propone que se intensifique el seguimiento siendo este de forma trimestral y ya no de forma anual para evitar afectaciones mayores a la operación. Para estos empleados deben ejecutarse planes de capacitación y fortalecimiento de capacidades y habilidades que les permitan alcanzar un mejor rendimiento en la siguiente medición de desempeño.

Una estrategia para mejorar el rendimiento laboral del personal es capacitar al personal de nivel *Semi* para que mantenga su aporte de alto valor y con el tiempo completen la curva de aprendizaje que les permita alcanzar los cargos de mayor jerarquía dentro de la empresa como reconocimiento por su excelente gestión.

De igual forma se propone aumentar la contratación de personal de nivel *Semi* y reducir la contratación de empelados de nivel *Junior*, de forma que se redistribuya los porcentajes por nivel de cargo, siendo la mayoría del recurso empelados de nivel *Semi*, ya que estos presentaron el mejor desempeño.

Adicionalmente se propone que se implementen políticas, programas y prácticas efectivas para atraer, retener, motivar y desarrollar el talento humano de la empresa, ya que esto puede conducir a una mejor productividad de la empresa y alineamiento de la gestión para el cumplimiento de los objetivos organizacionales.

5. Conclusión y trabajos futuros

El estudio de desempeño evaluó el rendimiento laboral de los empleados de acuerdo a las tareas reportadas por el personal en la plataforma *Jira* en el año 2022. Para complementar el *dataset* y profundizar el análisis fue necesario aumentar la cantidad de datos disponibles, por lo cual se incluyeron ficheros adicionales compartidos por otras áreas de la empresa, los cuales también fueron extraídos de *Jira*.

Por medio por medio la metodología *KDD (Knowledge Discovery in Databases)* se logró identificar los atributos principales que representaban al conjunto de datos para construir un modelo de machine learning que clasificara a los empleados según el desempeño realizado. Inicialmente este Trabajo de Fin de Master (TFM) se abordó como un problema de regresión, pero debido a los resultados insatisfactorios obtenidos se optó por trabajarlo como un problema de clasificación.

Para complementar la categorización de los empleados, es necesario implementar un sistema de alertas que permita identificar brechas y desviaciones existentes que se presenten en la gestión desarrollada por los empleados para que puedan ser corregidas de forma oportuna y adecuada, para que se genere el menor impacto posible en la operación y en el cumplimiento de los objetivos. De igual forma para que se realice el seguimiento correspondiente con el fin de identificar acciones que permitan la mejora.

Sería interesante contar con resultados de evaluaciones de desempeño previas realizadas a los empleados, es decir el resultado histórico de rendimiento, con el fin de analizar el comportamiento previo y valorar la evolución de la gestión de los empleados en un contexto más amplio para utilizar modelos de series temporales.

En un ámbito aplicado, el modelo de *machine learning* generado podría ser implementado en empresas que cuenten con registros de la ejecución de tareas operativas de sus empleados, ya que el análisis realizado constituye una fuente de valor para ser consultada, aplicada y adaptada a diferentes contextos. Además, puede servir como referente para otros estudiantes que estén interesados en profundizar en estrategias de medición, evaluación de desempeño y visualización de resultados.

De cara a próximos trabajos, utilizaría una mayor cantidad de registros y atributos extraídos de otras fuentes de información diferentes a *Jira* para tener un marco de referencia mayor que permita generar nuevo conocimiento e integre nuevos tipos de datos al modelo construido con el fin de enriquecerlo y que este mejore su precisión.

También se podrían profundizar en la creación de un cuadro de mando para visualizar múltiples métricas en un solo lugar. Puede contener varios gráficos y tablas que muestran diferentes métricas y tendencias.

Adicionalmente si la empresa utiliza *Jira*, se podría considerar la visualización de los datos en tiempo real. Esto podría ser particularmente útil para los gerentes que necesitan tomar decisiones rápidas en función de los datos.

6. Referencias

D. Calvo. “Aprendizaje Supervisado”, Marzo 2019. [En línea]. Recuperado el 24 de noviembre de 2021, a partir de: https://www.diegocalvo.es/aprendizaje-supervisado

F. Sciarrone, "Machine Learning and Learning Analytics: Integrating Data with Learning," [En línea]. Recuperado el 25 de noviembre de 2021, a partir de : 2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET), Olhao, 2018, pp. 1-5, DOI: 10.1109/ITHET.2018.8424780

Scikit, «API Reference - scikit-learn 0.24.2 documentation» [En línea]. Recuperado el 27 de diciembre de 2021, a partir de: https://scikit-learn.org/stable/modules/classes.html

Medium, «Understanding AUC-ROC Curve | by Sarang Narkhede | Towards Data Science» [En línea]. Recuperado el 10 de diciembre de 2021, a partir de: https://towardsdatascience.com/understanding-auc-roccurve-68b2303cc9c5.
Máster Oficial en Big Data y Data Science | Edición Abril 2021 66

Pandas, «API Reference - pandas 1.2.4 documentation» [En línea]. Recuperado el 12 de diciembre de 2021, a partir de: https://pandas.pydata.org/docs/reference/index.html

PowerData, «El valor de la gestión de los datos» [En línea]. Recuperado el 20 de diciembre de 2021, a partir de: https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/312597/la-limpieza-de-datos-la-etapa-previa-a-los-procesos-etl#:~:text=Importancia%20de%20la%20etapa%20de,de%20datos%20repetidos%20o%20inservibles

Actionsdata, «La importancia de la limpieza de datos para tu negocio». [En línea]. Recuperado el 10 de enero de 2022, a partir de: https://www.actionsdata.com/blog/la-importancia-de-la-limpieza-de-datos-para-tu-negocio

Prezi, «Cual es la importancia de la correlación en la investigación» [En línea]. Recuperado el 10 de enero de 2022, a partir de: https://prezi.com/2pehfgzktayn/cual-es-la-importancia-de-la-correlacion-en-la-investigacio/#:~:text=El%20an%C3%A1lisis%20de%20correlaci%C3%B3n%20produce,el%20grado%20de%20la%20relaci%C3%B3n.

Wikipedia, «Expresión regular». [En línea]. Recuperado el 10 de enero de 2022, a partir de: https://es.wikipedia.org/wiki/Expresi%C3%B3n_regular

Management Solutions, «Machine Leaning». [En línea]. Recuperado el 11 de noviembre de 2022, a partir de: https://www.managementsolutions.com/sites/default/files/publicaciones/esp/machine-learning.pdf

Paradigma Digital, «Conceptos de Machine Learning». [En línea]. Recuperado el 12 de enero de 2022, a partir de: https://f.hubspotusercontent00.net/hubfs/2189055/Ebooks/ebook-50-conceptos-machine-learning-Paradigma-Digital.pdf

Britos. Martinez, P. R. (2008). *Procesos De Explotacion De Informacion Basados En Sistemas Inteligentes*. https://core.ac.uk/download/pdf/301025592.pdf

Camacho, M., (2022). Factorialblog Indicadores de RRHH: *Equivalente a Tiempo Completo (FTE).* Tomado de https://factorialhr.es/blog/fte-equivalente-tiempo-completo/

Casanova, H. (2017). Graficación Estadística y Visualización de Datos. Escuela Venezolana de Planificación. Tomado de https://www.redalyc.org/journal/467/46754522005/

Cubillos, M y Núñez, S. (2012). "Guía para la construcción de indicadores de Gestión" Departamento administrativo de la Función Pública (DAFP), Bogotá. Tomado de https://www.funcionpublica.gov.co/documents/418537/506911/1595.pdf/6c897f03-9b26-4e10-85a7-789c9e54f5a3

Cuenca, A. (2010). Sistema automatizado para el control de los indicadores de gestión de un Cuadro de Mando Integral. Empresa EMCOMED. Tomado de http://ninive.ismm.edu.cu/handle/123456789/3187

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, *39*(11), 27–34. https://doi.org/10.1145/240455.240464

Fayyad, U. (1996). "The KDD process for extracting useful knowledge from volumes of data". ACM vol. 39 (11). Tomado de https://dl.acm.org/doi/10.1145/240455.240464

Gwanhoo, L. Weidong, X. (2010). Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility. Management Information Systems Research Center, University of Minnesota. Vol. 34, No. 1. Tomado de https://www.jstor.org/stable/20721416?seq=1

Gupta. Bhatnagar. Wasan, S. K. V. S. K. (1997). *A Proposal for Data Mining Management System*. Researchgate.net. Recuperado mayo de 2022, de https://www.researchgate.net/profile/Vasudha-

Bhatnagar/publication/2403357_A_Proposal_for_Data_Mining_Management_System/links/570a826f08aea660813722e3/A-Proposal-for-Data-Mining-Management-System.pdf

Han. Kamber. Pei, J. M. J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers is an imprint of Elsevier. http://myweb.sabanci-univ.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf

Hernández, G., Martínez, L., Jiménez, R., & Jiménez, F. (2019). Métricas de productividad para equipo de trabajo de desarrollo ágil de software: una revisión sistemática. *TecnoLógicas*, *22*, 63–81. https://doi.org/10.22430/22565337.1510

Hurtado. Zuñiga. Durazno, G. M. S. (2020). Implementación de indicadores de gestión por procesos para empresas de desarrollo de software. *Revista Publicando*. https://revistapublicando.org/revista/index.php/crv/article/view/2101/2122

J. Zaki. Wagner, M. M. (2014). *Data Mining And Analysis*. https://doc.lagout.org/Others/Data%20Mining/Data%20Mining%20and%20Analysis_%20Fundamental%20Concepts%20and%20Algorithms%20%5BZaki%20%26%20Meira%202014-05-12%5D.pdf

Lönnqvist Y Pirttimäki, T. (2006). *Medición de Business Intelligence*. ResearchGate. Recuperado mayo de 2022, de https://www.researchgate.net/figure/Characterising-the-measurement-of-BI-Loennqvist-and-Pirttimaeki-2006_tbl1_220826011

Luengo. Herrera. Garcia, S. J. F. (2014). *Data Preprocessing in Data Mining* (Vol. 72). Springer. http://pzs.dstu.dp.ua/DataMining/preprocessing/bibl/Data%20Preprocessing%20in%20Data%20Mining.pdf

Nigro. Xodo. Corti. Terren, H. O. D. G. D. (2022). *KDD (Knowledge Discovery in Databases): Un proceso centrado en el usuario*. INCA/INTIA - Departamento de Computación y Sistemas. Facultad de Ciencias Exactas - UNICEN – Tandil. http://sedici.unlp.edu.ar/bitstream/handle/10915/21220/Documento_completo.pdf?sequence=1

Oliveira. Conte. Cristo. Mendes, E. T. M. E. (2016). Percepciones de los gerentes de proyectos de software sobre los factores de productividad: resultados de un estudio cualitativo. *ACM Digital Library*. Recuperado mayo de 2022, de https://dl.acm.org/doi/10.1145/2961111.2962626

Rocha Granados, S. C. (2021, mayo). Mejoramiento De Procesos Analiticos Teniendo Como Principal Activo La Informacion Utilizando Tecnicas De Carga, Extraccion Y Transformacion De Los Datos Para Entidades Financieras. *Universidad Católica*. Recuperado mayo de 2022, de https://repository.ucatolica.edu.co/bitstream/10983/27091/1/Documento_Base%20Trabajo%20de%20Grado.pdf

Rojas Caro, J., & Matallana Quiroga, L. (2016). Los indicadores de gestión como herramienta de competitividad empresarial; Universidad de la Salle, Bogotá. Tomado https://ciencia.lasalle.edu.co/cgi/viewcontent.cgi?article=2350&context=administracion_de_empresas

Zapata, Arbeláez, J. J., Gasca-Hurtado, G. P., Manrique-Losada, B., & Machuca-Villegas, L. (2021). Caracterización de métodos de evaluación de desempeño para equipos de desarrollo de software. *Ingeniare. Revista chilena de ingeniería*, *29*(1), 129–140. https://doi.org/10.4067/s0718-33052021000100129

Sarkar, D., Bali, R., & Sharma, T. (2018). Practical Machine Learning With Python. Bangalore: Springer Science.

Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques. Greece: University of Peloponnese.

Will Koehrsen, 2019. Github. Recuperado el 10 de noviembre de 2021, https://github.com/WillKoehrsen/feature-selector

Alpaydin, Ethem. 2016. Machine Learning. The MIT Press Essential Knowledge Series. London, England: MIT Press.

Hothorn, T. (2018). CRAN Task View: Machine Learning & Statistical Learning. Recuperado el 12 de noviembre de 2021, a partir de https://CRAN.R-project.org/view=MachineLearning

Raona. (2017). Machine Learning Whitepaper. Technology. Recuperado el 17 de noviembre de 2021, a partir de https://www.slideshare.net/raona/machine-learning-whitepaper

J. Orellana, “Arboles de decisión - Parte I”, Arboles de decision y Random Forest, Noviembre 2018, [En línea]. Recuperado el 17 de noviembre de 2021, a partir de: https://bookdown.org/content/2031/arboles-dedecision-parte-i.html

7. Anexos

7.1. Repositorio con los ficheros utilizados

En el siguiente repositorio de *Github* se encuentran cargados los diferentes ficheros utilizados para llevar a cabo el presente TFM, incluido el código Python (en notebook), las diferentes fuentes externas y el *dashboard* elaborado en *Power BI* (.pbix).

https://github.com/crisRkr/TFM

7.2. Glosario de términos

A continuación, se ha elaborado el siguiente glosario de términos utilizando como fuente los manuales de las asignaturas del Máster de Big Data y Data Science de la VIU (curso 2021/22) y las sesiones grabadas correspondientes.

Algoritmo: una secuencia lógica de instrucciones que describen detalladamente cómo resolver un problema paso a paso.

Machine learning (Machine Learning): una rama de la Inteligencia Artificial que se enfoca en el diseño de mecanismos para que los sistemas informáticos puedan aprender por sí mismos. Esto implica la capacidad de descubrir patrones y regularidades en datos o situaciones previas y aplicarlos a nuevas situaciones o problemas similares.

Correlación: una medida numérica que evalúa la relación entre dos o más variables.

Coeficiente de correlación de Pearson: una medida que indica el grado de dependencia lineal que existe entre dos variables aleatorias cuantitativas.

Curva ROC: una representación gráfica de la sensibilidad en relación con la especificidad de un sistema clasificador binario, según se varía el umbral de discriminación. El análisis de la curva ROC proporciona herramientas para seleccionar modelos óptimos y descartar modelos subóptimos independientemente del costo de la distribución de las dos clases objetivo.

Accuracy: se define como la proporción de predicciones correctas en relación con el número total de observaciones. Esta medida indica cuán precisa es un modelo de Machine Learning en cuanto a sus predicciones.

Función diferenciable: una función matemática que se puede derivar en cualquier dirección y puede aproximarse, al menos, hasta el primer orden por una aplicación afín.

Grid Search: una búsqueda exhaustiva sobre valores específicos de parámetros para un estimador.

Inteligencia de Negocio (Business Intelligence): la capacidad de transformar datos en información que ayuda a gestionar una empresa. Esto involucra procesos, aplicaciones y prácticas que respaldan la toma de decisiones ejecutivas.

Iterativo: un proceso que se repite muchas veces.

KDD: Knowledge Discovery in Databases o en su traducción es el descubrimiento de conocimiento en bases de datos.

K-folds: Es una técnica de cross validation utilizada para evaluar los resultados de un análisis estadístico y garantizar que sean independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética de las medidas de evaluación sobre diferentes particiones. Se utiliza en entornos donde el objetivo principal es la predicción y se desea estimar la precisión de un modelo que se llevará a la práctica.

Matriz de confusión: una tabla que se utiliza para describir el rendimiento de un modelo de clasificación en un conjunto de datos de prueba para los que se conocen los valores verdaderos.

Media estadística: una medida de tendencia central que representa el valor promedio de un conjunto de datos numéricos.

Mediana: La mediana es un número que divide un conjunto de datos en dos partes iguales, separando la mitad superior de la inferior.

Método: Un método es una forma ordenada y sistemática de abordar una tarea o un problema, con el objetivo de lograr un resultado o un fin determinado.

Modelo matemático: Un modelo matemático es una descripción de un fenómeno o hecho en términos matemáticos, que permite analizarlo y comprenderlo de manera más precisa y rigurosa.

Outlier: Es un punto en un conjunto de datos que difiere significativamente del resto, y que puede haber sido generado por mecanismos distintos a los que generaron los demás datos. La detección de outliers implica analizar los valores de todas las tuplas y columnas de una base de datos relacional, o de todos los documentos de una base de datos NoSQL.

Overfitting: El sobreajuste es una situación en la que un modelo de machine learning se ajusta demasiado bien a los datos de entrenamiento, y por lo tanto, es incapaz de generalizar y producir resultados precisos en nuevos datos. Esto sucede cuando el modelo aprende patrones que son específicos del conjunto de entrenamiento, pero que no se aplican al conjunto de datos completo.

Rango: El rango de un conjunto de datos es el intervalo entre el valor máximo y el valor mínimo de los datos.

Regresión: La regresión es un proceso estadístico que se utiliza para estimar las relaciones entre diferentes variables, y para predecir los valores de una variable en función de los valores de otras variables.