# Predicción de bajas de empleados en una compañía usando modelos de Machine Learning

Shirley Viviana Jiménez Osorio

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Asesor
Daniela Serna Buitrago
Especialista en analítica y ciencia de datos.

Universidad de Antioquia
Facultad de Ingeniería
Especialización en Analítica y Ciencia de Datos
Medellín, Antioquia, Colombia
2023

| | |
|---|---|
| **Cita** | (Jiménez Osorio, 2023) |
| **Referencia**<br>**Estilo APA 7 (2020)** | Jiménez Osorio, S.V. (2023). *Predicción de bajas de empleados en una compañía usando modelos de Machine Learning* Trabajo de grado especialización]. Universidad de Antioquia, Medellín, Colombia. |



Especialización en Analítica y Ciencia de Datos, Cohorte V.

Centro de Investigación Ambientales y de Ingeniería (CIA).

## Dedicatoria

A mi familia por siempre estar alentando las cosas que emprendo.

## Agradecimientos

Quiero agradecer especialmente a mis profesores por su profesionalismo, paciencia y vocación. A mis compañeros de clase por su alegría, risas y generosidad.

## Tabla de contenido

## Lista de tablas



## Lista de figuras

## Siglas, acrónimos y abreviaturas

| | |
|---|---|
| **GH** | Gestión Humana |
| **ML** | Machine Learning |
| **SVM** | Máquina de vectores de soporte |
| **LR** | Regresión Logística |
| **Treeclass** | Árbol de decisión |
| **CRISP-DM** | Cross Industry Standard Process for Data Mining |
| **UdeA** | Universidad de Antioquia |

## Resumen

La rotación de empleados se ha convertido en un problema para las empresas en la actualidad; una alta rotación implica gasto de mayores recursos en reclutamiento y capacitaciones, para el caso que se está tratando, una empresa de venta de seguros, la contratación de un nuevo vendedor implica un periodo de ajuste en capacidades de ventas, dada por la experiencia, por tal razón la identificación de posibles bajas futuras le permitirá al área de Gestión Humana realizar intervenciones de retención que permitan la permanencia del empleado. En este documento se hará una presentación de diversos experimentos ejecutados para predecir la deserción de un empleado, siendo la variable de salida, 1 deserta, 0 permanece. Se corrieron 3 modelos de Machine Learning, el primero es un árbol de decisión donde se ejecutaron dos iteraciones, probando inicialmente con parámetros elegidos a criterio propio y la segunda usando funciones específicas que determinan los mejores parámetros, de igual manera se corre un modelo de Máquina de vectores de soporte SVM y una regresión logística aplicando las funciones de mejores parámetros. Por último, se aplica una iteración corriendo los 3 modelos bajo validación cruzada. Los resultados indican que el mejor modelo para predecir las bajas de los empleados es el de árbol de decisión usando validación cruzada y funciones de detección de mejores parámetros, esto le permitirá a la compañía identificar a qué empleados deben intervenir urgentemente y comenzar el proceso de retención con los profesionales en Gestión Humana.

https://github.com/shirleyjimenezo/ShirleyMonografiaUdeA/blob/main/Codigo_monografia_final_.ipynb

*Palabras clave*: Rotación, deserción, empleados, Machine Learning, predicción

**Abstract**

Employee attrition has become a problem for companies today; A high turnover implies spending greater resources on recruitment and training, for the case in question, an insurance sales company, the hiring of a new salesperson implies a period of adjustment in sales capabilities, given by experience, by For this reason, the identification of possible future withdrawals will allow the Human Management area to carry out retention interventions that allow the employee to remain. In this document, a presentation will be made of various experiments executed to predict the desertion of an employee, with the output variable being 1 deserts, 0 remains. 3 Machine Learning models were run, the first is a decision tree where two iterations were executed, initially testing with parameters chosen at our own discretion and the second using specific functions that determine the best parameters, in the same way a Machine model is run. of SVM support vectors and a logistic regression applying the best parameter functions. Finally, an iteration is applied running the 3 models under cross validation. The results indicate that the best model to predict employee withdrawals is the decision tree using cross-validation and detection functions of better parameters. This will allow the company to identify which employees should intervene urgently and begin the process of retention with professionals in Human Management.

*Keywords***:** Turnover, attrition, employees, Machine Learning, prediction

# 1. Descripción del problema

La retención y permanencia del talento humano es uno de los principales desafíos que enfrentan las empresas en la actualidad; este se relaciona con aspectos fundamentales para el desarrollo adecuado de sus negocios y también como una arista importante en el ahorro de recursos financieros para la contratación. Es de esperarse que un empleado con alta permanencia entienda y resuelva de forma más eficiente los desafíos y los problemas a los que se vea enfrentado, el conocimiento del negocio le agrega experiencia a la hora de desarrollar su trabajo de forma autónoma, así como en asuntos que requieran liderazgo. Por esta razón se ha vuelto relevante para las empresas determinar y predecir qué factores conllevan a la deserción de sus colaboradores, con el fin de tomar las medidas que permitan ampliar el tiempo de permanencia, esto por su puesto sin dejar de tener en cuenta que contratar un nuevo empleado implica incurrir en gastos de reclutamiento, exámenes médicos, capacitaciones, tiempos de ajustes, etc.

Retener un empleado impacta también directamente sobre la carga que enfrentan áreas como gestión humana o formación. Es por esta razón que se plantea el desarrollo de un modelo predictivo, que permita modelar la deserción de un empleado, teniendo en cuenta diferentes variables de su entorno personal y laboral, esto ayudará que se tomen medidas preventivas que permitan retener al mismo.

## 1.1. Problema de negocio

En una empresa dedicada a la venta de seguros se ha incrementado de manera alarmante la deserción de su talento humano, es por esto que se requiere identificar los empleados que tienen alta probabilidad de abandono y crear medidas anticipadas de retención. Así se espera mitigar el impacto que tiene la rotación del equipo de ventas dentro de la compañía.

## 1.2. Aproximación desde la analítica de datos

Los modelos de Machine Learning que se pretenden desarrollar permitirán identificar, basando en los datos históricos de salidas de la compañía, cuál de los empleados que se encuentran en la actualidad activos, tienen un comportamiento similar y puedan ser futuras bajas.

### 1.3. Origen de los datos

Los datos que se van a usar provienen de una base pública de Kaggle con originalmente 19.104 instancias, esta recoge información mensual por dos años de los datos de los empleados de una compañía dedicada a la venta de seguros. https://www.kaggle.com/datasets/pavan9065/predicting-employee-attrition.

### 1.4. Métricas de desempeño

Las métricas a utilizar para evaluar los modelos será la matriz de Confusión, esta matriz permite observar cómo fueron clasificados los datos del modelo, es decir, cuáles de los que fueron clasificados como baja en realidad sí eran una baja y viceversa. En la diagonal de la matriz de confusión tenemos los éxitos de clasificación.

**Figura 1**

*Matriz de Confusión*

| | | **Predicción** | |
|---|---|---|---|
| | | **Positivos** | **Negativos** |
| **Observación** | **Positivos** | Verdaderos Positivos (VP) | Falsos Negativos (FN) |
| | **Negativos** | Falsos Positivos (FP) | Verdaderos Negativos (VN) |

- **VP** es la cantidad de *positivos* que fueron *clasificados correctamente* como positivos por el modelo.
- **VN** es la cantidad de *negativos* que fueron *clasificados correctamente* como negativos por el modelo.
- **FN** es la cantidad de *positivos* que fueron *clasificados incorrectamente* como negativos.
- **FP** es la cantidad de *negativos* que fueron *clasificados incorrectamente* como positivos.

*Nota.* Fuente Zelada, Carlos (2017). Evaluación de modelos de clasificación. https://rpubs.com/chzelada/275494.

De esta matriz se derivan las siguientes 4 métricas que dan el resultado de éxito del modelo a través de un número, entre más cercano a 1 mayor éxito de clasificación:

**Tabla 1**

*Resumen Métricas*

| Accuracy | Precisión | Recall | F1 Score |
|---|---|---|---|
| Porcentaje total de valores clasificados correctamente, tanto en positivos como en negativos: $\frac{(VP + VN)}{(VP + FN + FP + VN)}$ | Porcentaje de valores que han sido clasificados como positivos son realmente positivos: $\frac{VP}{VP + FP}$ | Ratio de valores positivos, se mide cuantos positivos han sido clasificados correctamente: $\frac{VP}{VP + FN}$ | Esta métrica combina Precisión y Recall: $\frac{Recall * Precisión}{Recall + Precisión} * 2$ |

La otra métrica a utilizar será la curva ROC (característica operativa del receptor), esta curva permite comparar los modelos a utilizar.

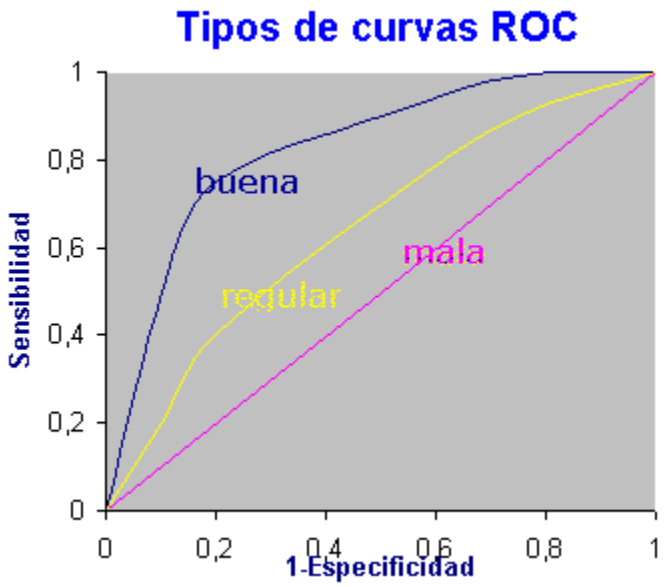
Esta curva permite observar entre varios modelos, cuál es el mejor, entre más cerca está la curva a la parte superior izquierda de la gráfica mejor es el modelo, esto se traduce en que hubo una mejor tasa de verdaderos positivos, y una disminución de falsos positivos.

Torres Luis (s.f.), lo describe en su blog de The Machine Learners como: “Es una gráfica que enfrenta el ratio de falsos positivos (eje x) con el ratio de falsos negativos (eje y). Estos ratios los va obteniendo en función de una serie de umbrales definidos entre 0 y 1”

Así mismo Torres explica los ejes a mayor profundidad señalando que la tasa de falsos positivos es lo que se conoce como tasa de sensibilidad y consiste en dividir el número de positivos verdaderos sobre los falsos negativos más los verdaderos positivos, de igual manera la especificidad vendría siendo la inversa de la tasa de falsos positivos.

**Figura 2**

*Tipos de curvas ROC*




*Nota.* Fuente http://www.hrc.es/bioest/roc_1.html

# 2. Objetivos

## 2.1.Objetivo general

Aplicar modelos de predicción que permitan clasificar a los empleados de la compañía como bajas o no bajas.

## 2.2.Objetivos específicos

- Realizar análisis descriptivo de las variables con las que cuenta el dataset.
- Implementar las transformaciones, limpieza y organización que se requiera sobre los datos.
- Definir 3 modelos de predicción a trabajar en el desarrollo del ejercicio y las modificaciones e iteraciones necesarias para reforzarlos.
- Comparar las métricas de desempeño de los 3 modelos elegidos, y elegir el que mejor funcione.

# 3. Datos

## 3.1. Datos originales

Se tiene un dataset de Kaggle con originalmente 19.104 instancias, corresponde a información mensual del 2016 y 2017 de los empleados de una compañía de seguros.

**Figura 3**

*Resumen de variables*

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19104 entries, 0 to 19103
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   MMM-YY                19104 non-null  object
 1   Emp_ID                19104 non-null  int64
 2   Age                   19104 non-null  int64
 3   Gender                19104 non-null  object
 4   City                  19104 non-null  object
 5   Education_Level       19104 non-null  object
 6   Salary                19104 non-null  int64
 7   Dateofjoining         19104 non-null  object
 8   LastWorkingDate       1616 non-null   object
 9   Joining Designation   19104 non-null  int64
 10  Designation           19104 non-null  int64
 11  Total Business Value  19104 non-null  int64
 12  Quarterly Rating      19104 non-null  int64
dtypes: int64(7), object(6)
memory usage: 1.9+ MB
```

*Nota.* Elaboración propia.

Se dispone inicialmente de 13 de columnas, MMM-YY (Fecha para la cual se captura la información del empleado, está de forma mensual)

- Emp_ID (Borrar)
- Age (Edad del empleado)
- Gender (género, dos categorías, masculino, femenino)
- City (Ciudad, considerar borrarla)
- Education_Level (Nivel de educación, 3 categorías Bachelor, Master, College)

- Salary (Variable numérica en dólares)
- Dateofjoining (fecha de ingreso, con esta y la última fecha laborada se calcula la variable objetivo, renuncia o no), también nos serviría para calcular los años de experiencia en la compañía
- LastWorkingDate (Último día laborado)
- Joining_Designation (De esta variable no tenemos mucha información, pero por lo que se puede encontrar en internet, comprende el nivel del cargo en el cual se unió) Categórica del 1 al 5
- Designation (Bajo especulación, podría ser la nueva designación o nuevo puesto que tiene la persona, si es igual al Joining_Designation es porque no ha cambiado de nivel o cargo) Categórica del 1 al 5
- Total_Business_Value (No se tiene mayor información en Kaggle, pero consultando en internet, esta variable está compuesta por Incremento de Ingresos + Ahorro por Absentismo + Ahorro por Rotación https://faq.sparckco.com/knowledge/understanding-total-business-value)
- Quarterly_Rating (calificación trimestral del empleado) Categórica del 1 al 4

La variable dependiente “Attrition” se obtiene a través de la columna LastWorkingDate, si tiene datos en esta columna se clasifica como 1, baja, de lo contrario 0.
Adicionalmente se crearon dos nuevas variables, retention_days y promoted, la primera se calculó restando la fecha de LastWorkingDate y Dateofjoining, para el caso de personas activas, se restó la fecha donde fue registrada la persona MMM-YY. Promoted se obtuvo de analizar si hubo cambios entre el cargo asignado al inicio Joining_Designation, y el último o cargo en el momento Designation, se otorga un 1 para sí fue promovido 0 si no.

La variable retention_days se decide convertirla a 7 rangos para agrupar los días de permanencia en meses, <1mes, 1-3 meses, 3-6 meses, 6-9 meses, 9-12 meses, 12-36 meses y +36 meses. Por lo que la variable a usar para permanencia será, Retention_Rango.

Dado que el problema planteado en este trabajo corresponde a clasificar el personal activo dentro de la compañía como 1 posible baja 0, permanece en la empresa. Se usará la información más reciente agregada por empleado, por lo que se procede a ordenar la base de datos de registros más recientes a más antiguos, eliminando duplicados y conservando el primer registro encontrado.

Este ejercicio arroja un total 2.381 instancias con 1.616 bajas y un total de 765 empleados activos en la compañía. Será con esta base de datos con la que se procederá a ejecutar todo lo relacionado con análisis de información, datos atípicos, balanceos y todo lo que corresponda al desarrollo del modelo

### 3.2. Datsets

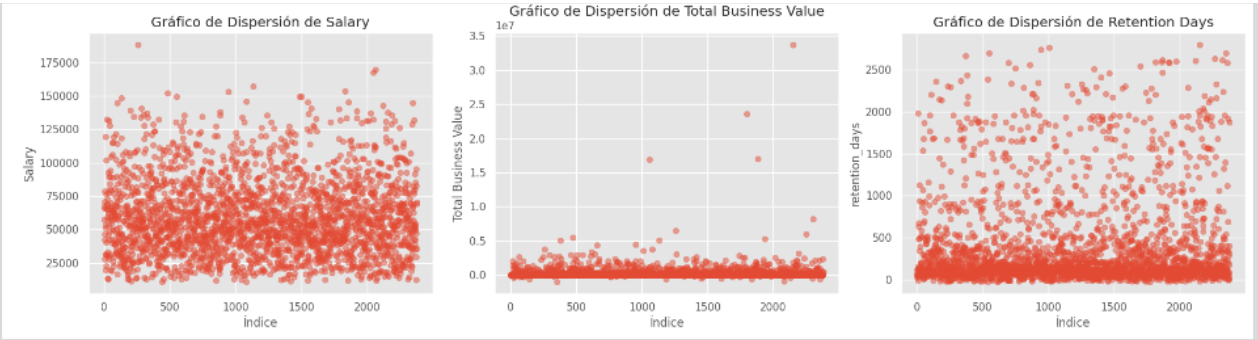
Se establece la variable ***y*** que es igual a Attrition, y será nuestra variable de salida, de igual forma se establece el dataset ***X***, que contiene todas las variables independientes del modelo. Para la primera parte del ejercicio, el dataset se divide en data de entrenamiento, que incluye el 80% de los datos, y en data de prueba, que representa el 20% de los datos. Se generaron los conjuntos de datos tanto para X como y, generando X_train, X_test, y_train, y_test. Utilizando la librería sklearn y su módulo para separar los datos en entrenamiento y prueba train_test_split. Más adelante los modelos se aplicarán con separación a través de validación cruzada que es una forma más robusta de separar los datos.

### 3.3. Analítica descriptiva

Inicialmente se analizaron las variables cuantitativas disponibles en la base de datos, estas no presentaron agrupaciones de datos significativas para ser graficadas con sus frecuencias, es por esto que se utilizaron para estas 3 variables, gráficos de dispersión:

**Figura 4**

*Gráficos de dispersión para variables cuantitativas*



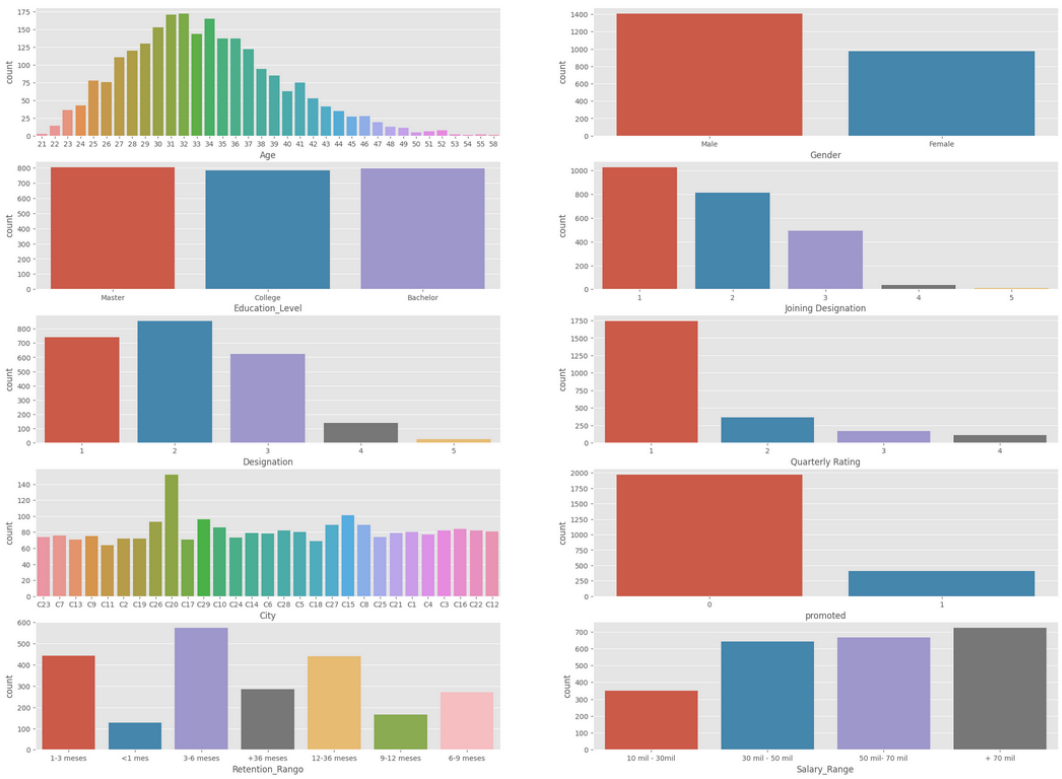
*Nota.* Elaboración propia.

Se puede observar para el caso de la variable Salary una concentración de los datos por debajo de los salarios correspondientes a 75.000 y algunos datos atípicos de salarios. Con la variable Total_business_Value la mayoría de los empleados tiene un cálculo igual a cero (1.621) esto corresponde al 68% de los empleados, por lo que esta variable no aporta información. Retention_ retention_days muestra una concentración de permanencia en 500 días, lo que significa más o menos año y medio dentro de la compañía.

Para las variables categóricas se obtuvieron los siguientes gráficos de frecuencias, acá se tomó la decisión de convertir la variable retention_days a Retention_Rango, de tal forma que ahora está representada en una variable cualitativa con 7 categorías según los meses de permanencia. De igual forma la variable Salary se agrupa en una nueva variable categórica con 4 rangos: 10 mil-30 mil, 30 mil-50 mil, 50 mil- 70 mil, + 70 mil dólares.

**Figura 5**

*Gráficos de distribución de las variables*




*Nota.* Elaboración propia.

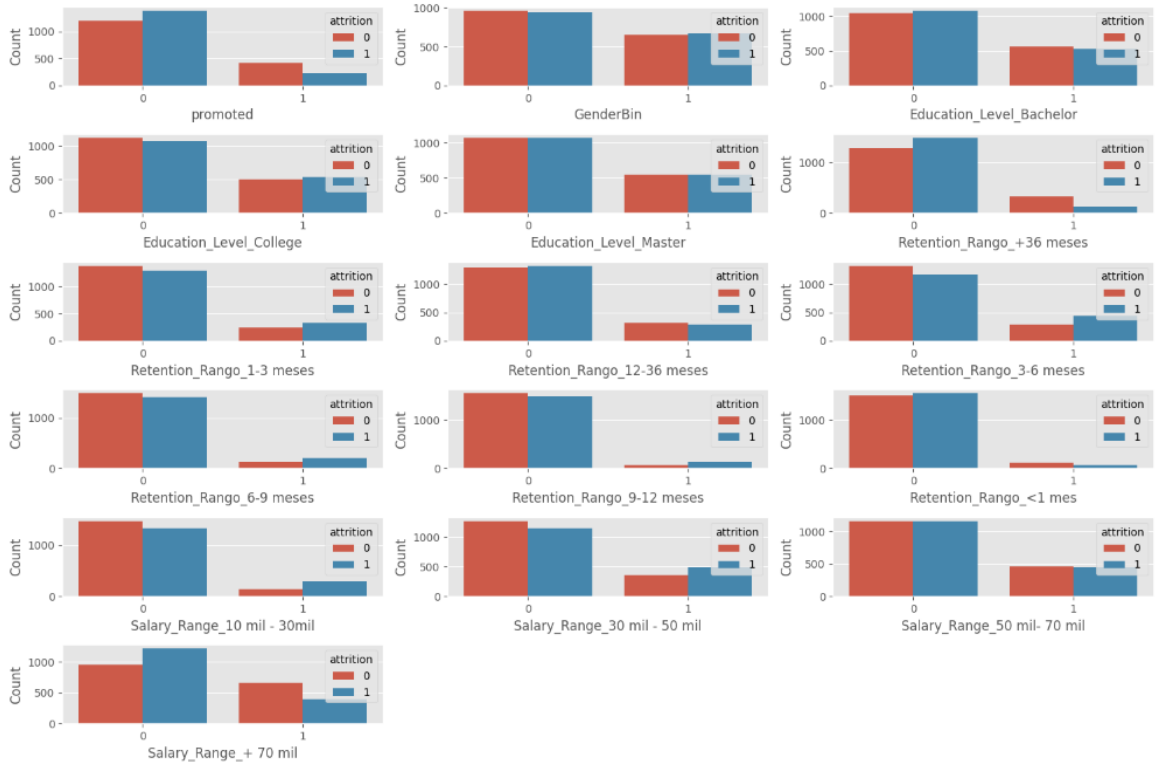
La variable edad muestra una distribución aparentemente normal, edad mínima 21, edad máxima 58, edad promedio de los empleados 33 años. La mayoría de los empleados es de género masculino 59%, mientras que las mujeres representan el 41% de los datos. El nivel educativo se distribuye prácticamente igual entre las 3 categorías, Master College, Bachelor. En cuanto a la variable Joining_Designation los puestos con los que más inician los empleados son el de categoría 1 y 2 que suman el 77% de los datos, por otro lado, la designación final en la que termina la persona está concentrada en el puesto 2, por lo que se podría inferir parcialmente que sí hay promoción en esta compañía, eso se verá más detalladamente cuando se observe la variable promoted. La variable Quarterly_Rating deja ver claramente que la mayoría de los empleados obtuvieron una calificación de 1 en el último trimestre donde fueron calificados. La variable ciudad muestra 29 diferentes ciudades donde laboran los empleados. La variable promoted muestra que aproximadamente el 15% de los empleados logró ascender o crecer dentro de la compañía. En cuanto a la variable Retention_Rango se puede observar una concentración en el rango de 3-6 meses, seguido de 1-3 meses. En cuanto a los rangos salariales, La categoría que más peso tiene es +70 mil con el 33% de los datos, seguida de 50 mil- 70 mil con el 28% y 30 mil- 50 mil con el 26%.

**Análisis bivariado:**

En los siguientes gráficos de la figura 6 se puede observar la distribución de las variables cualitativas independientes vs la variable dependiente Attrition

***Figura 6***

*Distribución de las variables cualitativas independientes vs la variable dependiente Attrition.*



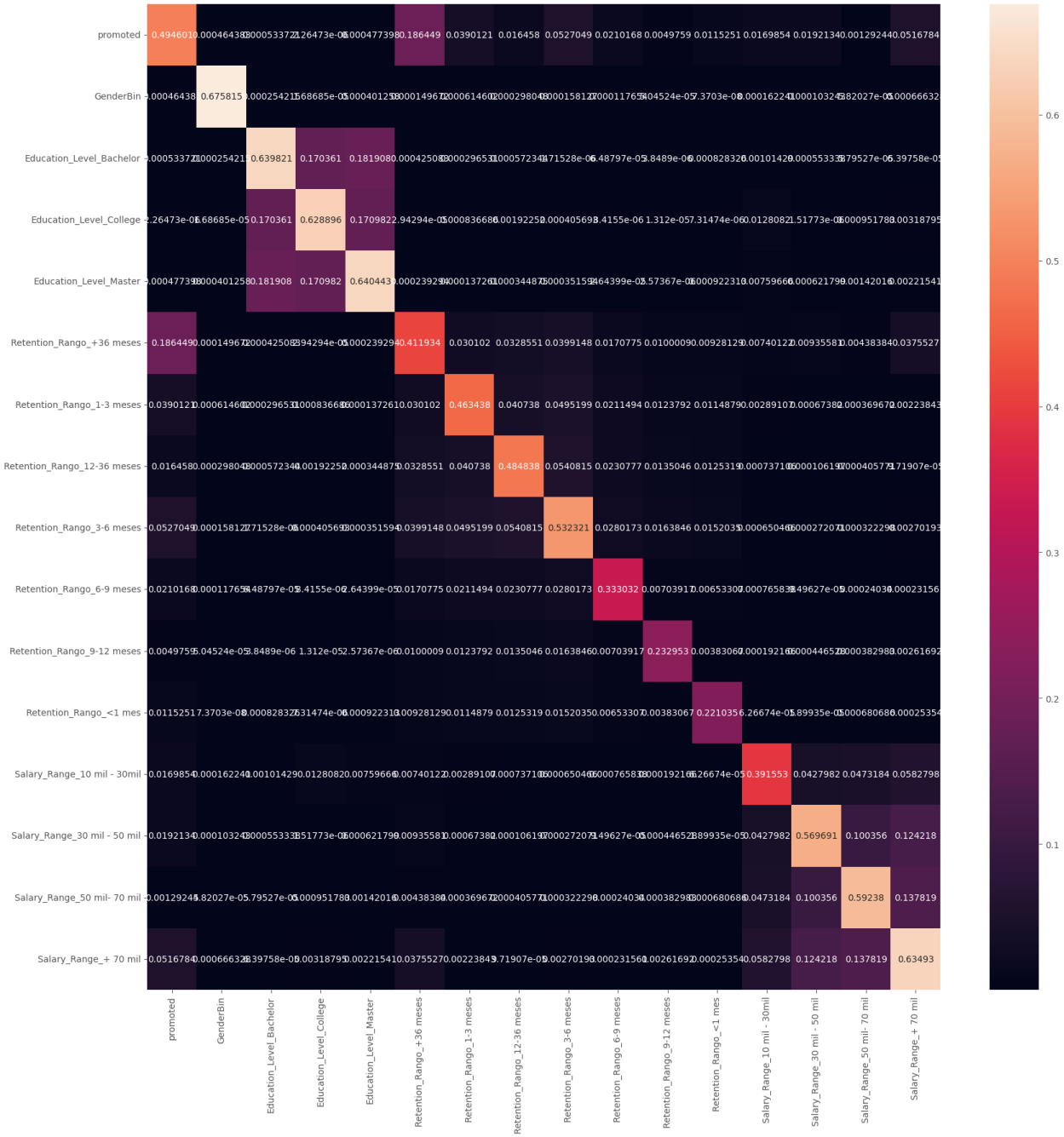

*Nota.* Elaboración propia.

## Matriz de Información Mutua

Esta matriz permite observar la relación entre todas las variables independientes que se tienen, de tal forma que un coeficiente cercano 1 indica correlación entre las mismas, un asunto que se desea evitar para no incurrir en problemas de redundancia y multicolinealidad, esta matriz permite inferir qué variables deben ser descartadas del análisis. La diagonal representa la relación de cada variable consigo misma, por lo que no es tenida en cuenta. Para este caso, no se observan relaciones con coeficientes altos, sólo promoted y el rango de permanencia +36 meses tienen un coeficiente de 0,19 lo que no representa mayor problema, podrían conservarse ambas variables.

**Figura 7**

*Matriz de información mutua*




*Nota.* Elaboración propia.

# 4. Proceso de analítica

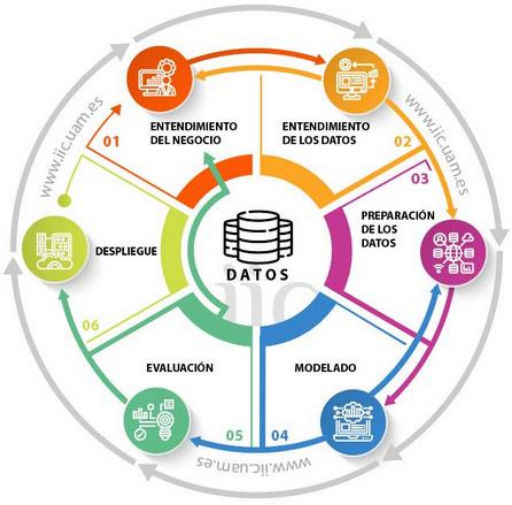
## 4.1. Pipeline principal

La metodología que se va a usar en este trabajo será la de CRISP-DM, en esta metodología se parte del **entendimiento del contexto del negocio**, para este caso la base de datos en uso es la de empleados de activos y retirados de una compañía de seguros, para la que se requiere desarrollar un modelo de clasificación que permita identificar posibles bajas de empleados activos. En este tipo de negocio interesa retener a los empleados, quienes son los agentes de ventas del producto, en este caso vendedores de seguros, la retención de un empleado no sólo implica la continuidad de las ganancias del negocio, sino que además le representa ahorro a la compañía en aspectos tales como reclutamiento de nuevo personal, capacitaciones, periodo de adaptación, entre otros.

De tal manera que identificar qué personas se encuentran en mayor riesgo de convertirse en una baja, le permite al área de recursos humanos, adelantar estrategias de retención, tales como actividades recreativas, salario emocional, beneficios en bonos, reconocimiento de la labor, etc. Este tipo de intervenciones tempranas estimulan la permanencia del empleado en la compañía y de esta manera se compensan los efectos negativos de una baja inesperada.

***Figura 8***

*Esquema del ciclo CRISP-DM estándar*




*Nota.* Fuente: https://www.iic.uam.es/innovacion/metodologia-crisp-dm-ciencia-de-datos/

En la segunda fase de esta metodología se realiza el **entendimiento de los datos** a través de un análisis exploratorio, que ya fue plasmado en el apartado anterior de Analítica descriptiva, la base que se está trabajando está compuesta de 2.381 registros con 1.616 personas que representan los retiros de la compañía y de 765 empleados activos.

Para la tercera fase que trata sobre **la preparación de los datos,** el dataset consta inicialmente de 12 características, de las cuales se procede a borrar las que no aportan información, estas son: MMM-YY, Emp_ID, Dateofjoining, LastWorkingDate, City, Joining Designation, Designation y Total Business Value, esta última se elimina de la base porque más del 60% de los datos están en 0, por lo cual no aportaría más información.

Así mismo, se crea la variable retention_days, que calcula el número de días entre la última fecha laboral y el Join Designation para el caso de empleados retirados, y el de la fecha del registro de la información y el Join Designation, para empleados activos. Dado que la variabilidad de días de permanencia es representativa y que, para un mejor análisis del tema de retención, es recomendación de los expertos en el negocio, analizar esta a través de rangos de permanencia; se decide que esta variable se transforme en Retention_Rango con 7 categorías de permanencia en meses: <1 mes, 1-3 meses, 3-6 meses, 6-9 meses, 9-12 meses, 12-36 meses, +36 meses.

Por otra parte, se procede a crear la variable promoted que se obtiene de comparar las columnas Joining Designation y Designation, es decir, cargo que tenía en el momento que entró en la compañía, y cargo que tiene el empleado en la fecha actual. Si la columna Designation es diferente a Joining Designation, se asume que la persona fue promovida de cargo, tomando valores de 1 para este caso y 0 si no fue así.

Para la variable Salary, que según se pudo observar en las gráficas del análisis exploratorio, no tenía una frecuencia alta, es decir, había pocos trabajadores con el mismo salario, situación que se puede explicar dado que este tipo de salario en el sector de ventas de seguros suele ser variable y componerse de un alto factor de comisiones por ventas. Se decidió agrupar esta variable

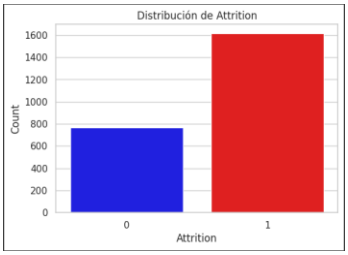
cuantitativa en rangos y así transformarla a una variable cualitativa de 4 rangos: '10 mil - 30mil', '30 mil - 50 mil', '50 mil- 70 mil', '+ 70 mil'.

Después de borrar las variables que no se necesitan en la base de datos, se obtiene un total de 7 variables independientes, 5 categóricas (Gender, Education_Level, promoted, Salary_Range, Retention_Rango) y 2 cuantitativas (age y Quarterly Rating). A través de un proceso para obtener las variables dummies de las variables categóricas se usó la librería Pandas, pd.get_dummies(d, drop_first=0). De esta forma se obtienen 16 variables dummies, 2 cuantitativas y la variable de salida que es attrition, un dataset con 19 columnas y 2.381 filas.

Para finalizar el tratamiento de los datos, se procedió a observar que tan balanceada estaba la variable attrition, que es la variable de salida, como se puede observar en el gráfico hay un desbalance mayoritariamente en las personas retiradas, para que esto no afecte el modelamiento se procede a crear un objeto RandomOverSampler que permite crear copias aleatorias de las muestras de la clase minoritaria, en este caso para empleados activos, e igualar el número de muestras entre las diferentes clases, luego se realiza el sobremuestreo para las variables de entrada y salida y se procede a crear dos bases, una con las variables de entrada y otro con la variable de salida a partir del proceso anterior, para luego concatenarlos horizontalmente.

**Figura 9**

*Distribución de variable de salida, 1 baja, 0 permanece*




*Nota.* Elaboración propia.

De esta forma, se ha dejado lista la base en cuanto a tratamiento de datos, con la que se procederá a realizar el modelamiento.

### 4.2. Preprocesamiento

Como se mencionó en la parte anterior, para balancear las categorías de la variable respuesta se usó RandomOverSampler para crear muestras que permitan igualar las dos categorías, este procedimiento arroja un total de 3.232 filas finales.

### 4.3.Modelos

Se utilizaron tres modelos, un árbol de decisiones, uno de máquina de vectores de soporte (SVC) y una regresión logística.

Los modelos de árbol de decisión, son muy útiles para problemas de clasificación, este utiliza un sistema de reglas para dividir los datos, estas reglas están basadas en otras características, lo más interesante de este tipo de modelos, como lo menciona IBM, es primero que se tienen en cuenta las características que verdaderamente tienen impacto en la predicción de la variable dependiente, y segundo, que al existir reglas se puede entender la lógica de cómo funciona la predicción, evitando así el hecho de que el modelo se genera en una especie de "caja negra" como ocurre con otros modelos de machine learning.

Se utilizó para este modelo la librería de sklearn importando DecisionTreeClassifier. Para este primer modelo se comenzó el ejercicio definiendo a criterio personal, los parámetros a utilizar con modelo tipo 80-20 en entrenamiento y prueba. Como parte del ejercicio se va a correr este mismo modelo, pero usando validación cruzada, para observar los cambios en las métricas. y finalmente se usará la definición de parámetros automáticamente usando best_params_, get_depth que elige la mejor profundidad y get_n_leaves que permite obtener el número de nodos óptimo para el modelo.

En cuanto al modelo de máquina de vectores SVC, este modelo tiene como objetivo seleccionar un hiperplano que permite separar los datos en clases, las líneas de separación se conocen como vectores de soporte, un margen o espacio mayor entre las líneas indica una buena separación. Se usará svm.SVC de sklearn para correr el modelo, se usa parámetro kernel 'linear',

y probabilidad true, para que se calculen las probabilidades, esta librería permite usar los mejores parámetros automáticamente

**Figura 10**

*Modelo Máquina de vectores de soporte*

```
modelSVC       = svm.SVC(kernel='linear', probability = True).fit(X_train, y_train)
```

*Nota.* Elaboración propia.

Por último, la regresión logística es otro de los modelos predilectos a la hora de correr clasificaciones de datos, haciendo uso de una función logística se hallan las probabilidades de que ocurra el suceso. Nuevamente haciendo uso de los modelos que tiene la librería sklearn que automáticamente elige los mejores parámetros para correr el modelo se entrenará el mismo:

**Figura 11**

*Modelo de Regresión Logística*

```
modelLR        = LogisticRegression(random_state=0).fit(X_train, y_train)
```

*Nota.* Elaboración propia.

Para finalizar, estos tres modelos se correrán haciendo uso de la validación cruzada, con este método, en vez de dividir los datos en entrenamiento y prueba, los modelos se corren con todos los datos, esto ayuda a conservar información valiosa de los datos para aplicar este tipo de validación se hará uso sklearn.model_selection que permite obtener cross_validate y los score.

## 4.4.Métricas

Para obtener las métricas de desempeño de los 3 modelos a usar, se implementará, metrics.accuracy_score este componente de sklearn permite obtener la métrica de accuracy. De igual forma se usará confusion_matrix para obtener la matriz de confusión de cada modelo y metrics.classification_report que arroja las métricas principales de esa matriz, como accuracy, recall y f1-score.

Para calcular la curva ROC, se usará sklearn.metrics importando la función roc_curve y roc_auc_score que permite graficar la curva y hallar la puntuación.

# 5. Metodología

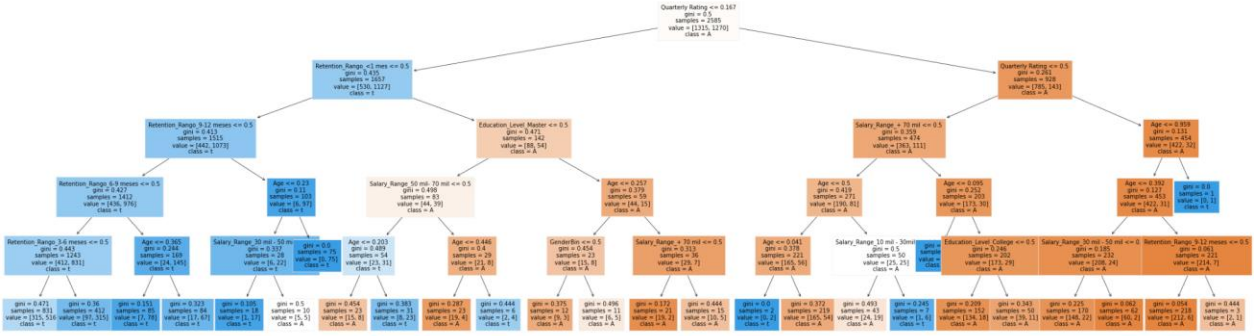
## 5.1. Baseline

El primer ejercicio ejecutado fue un árbol de decisión cuyos parámetros fueron modificados a gusto personal, con el fin de probar los resultados.

**Figura 12**

*Modelo de Árbol de decisión con criterios propios*

```
modelo = DecisionTreeClassifier(
            max_depth         = 5,
            criterion         = 'gini',
            random_state      = 123
           )
```

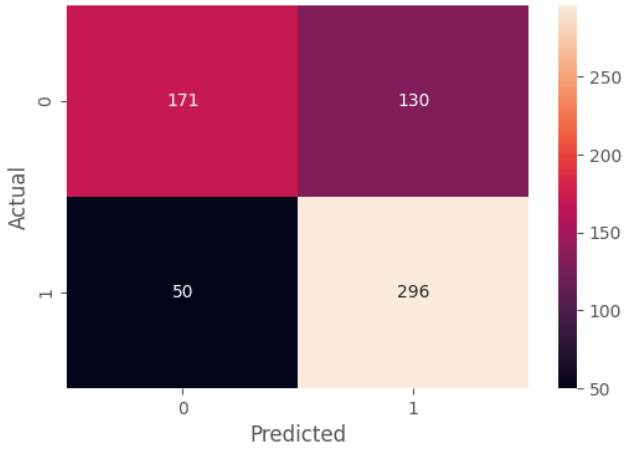


*Nota.* Elaboración propia.

definiendo una profundidad máxima de 5 con criterion ‘gini’ y random state 123, se obtuvieron los siguientes resultados a través de la matriz de confusión:

**Figura 13**

*Matriz confusión primera iteración del modelo de árbol de decisión con parámetros a criterio propio*




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.77 | 0.57 | 0.66 | 301 |
| 1 | 0.69 | 0.86 | 0.77 | 346 |
| accuracy | | | 0.72 | 647 |
| macro avg | 0.73 | 0.71 | 0.71 | 647 |
| weighted avg | 0.73 | 0.72 | 0.71 | 647 |

*Nota.* Elaboración propia.

Los resultados con esta primera iteración arrojan un accuracy, es decir la cantidad de éxitos clasificados por el modelo como Verdaderos positivos, y verdaderos falsos, es del 0.72, una cifra nada mal para haber elegido a criterio propio los parámetros.

En la siguiente iteración se va a usar nuevamente un árbol de decisión, pero esta vez hallando los mejores parámetros en profundidad y los nodos terminales, que permitan obtener el mejor resultado posible. Para esta parte se corrieron los tres modelos juntos de la siguiente manera:

**Figura 14**

*Resumen modelos con elección de mejores parámetros*

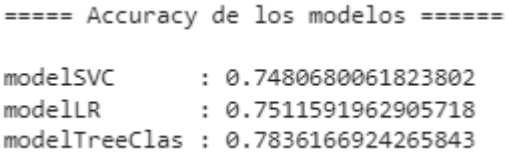
```
modelSVC       = svm.SVC(kernel='linear', probability = True).fit(X_train, y_train)
modelLR        = LogisticRegression(random_state=0).fit(X_train, y_train)
modelTreeClas  = tree.DecisionTreeClassifier(random_state=0).fit(X_train, y_train)
```

*Nota.* Elaboración propia.

Para esta iteración se usaron las librerías de sklearn que automáticamente elige la mejor configuración o combinación de parámetros tanto para máquina de vectores de soporte como para la regresión logística y el modelo de árbol de decisión, los resultados son los siguientes:

**Figura 15**

*Resultado del Accuracy de los modelos con elección de mejores parámetros*



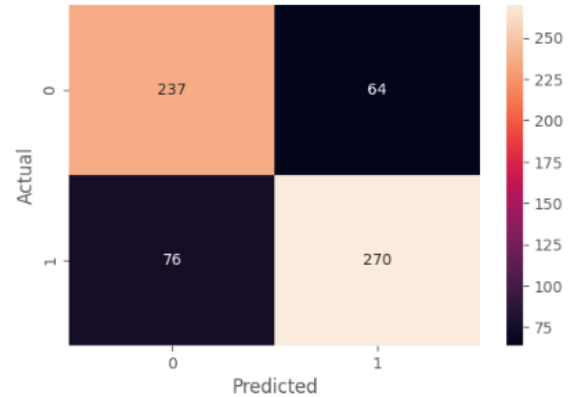

*Nota.* Elaboración propia.

el modelo que precisamente mejores resultados arroja es el de árbol de decisión, acá en esta segunda iteración, se puede observar cómo al elegir los mejores parámetros el modelo da resultados con una mejoría con relación al ejercicio manual que se realizó al inicio, pasando de 0.72 a 0.78 de accurrany. Los mejores parámetros se hallan con best_params_ y este arrojó que usando una profundidad del árbol con 21 y 573 nodos terminales se obtenían los mejores resultados.

las siguientes son las matrices de confusión del ejercicio de correr 3 modelos eligiendo automáticamente los mejores parámetros:

**Figura 16**

*Matriz de confusión de árbol de decisión con elección de mejores parámetros*

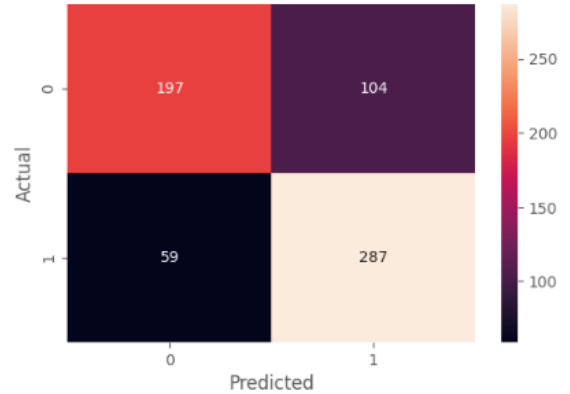



| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.76 | 0.79 | 0.77 | 301 |
| 1 | 0.81 | 0.78 | 0.79 | 346 |
| accuracy | | | 0.78 | 647 |
| macro avg | 0.78 | 0.78 | 0.78 | 647 |
| weighted avg | 0.78 | 0.78 | 0.78 | 647 |

*Nota.* Elaboración propia.

**Figura 17**

*Matriz de confusión de Máquina de Vectores de Soporte SVC con elección de mejores parámetros*

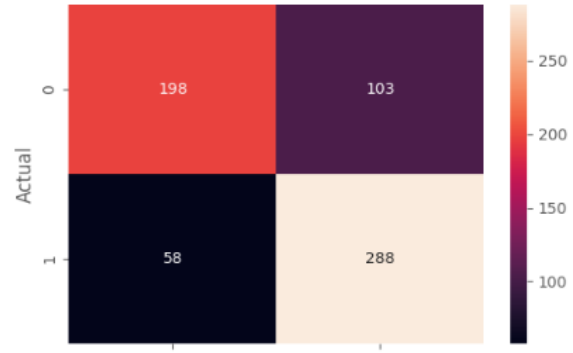



| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.77 | 0.65 | 0.71 | 301 |
| 1 | 0.73 | 0.83 | 0.78 | 346 |
| accuracy | | | 0.75 | 647 |
| macro avg | 0.75 | 0.74 | 0.74 | 647 |
| weighted avg | 0.75 | 0.75 | 0.75 | 647 |

*Nota.* Elaboración propia.

**Figura 18**

*Matriz de confusión de Regresión Logística con elección de mejores parámetros*



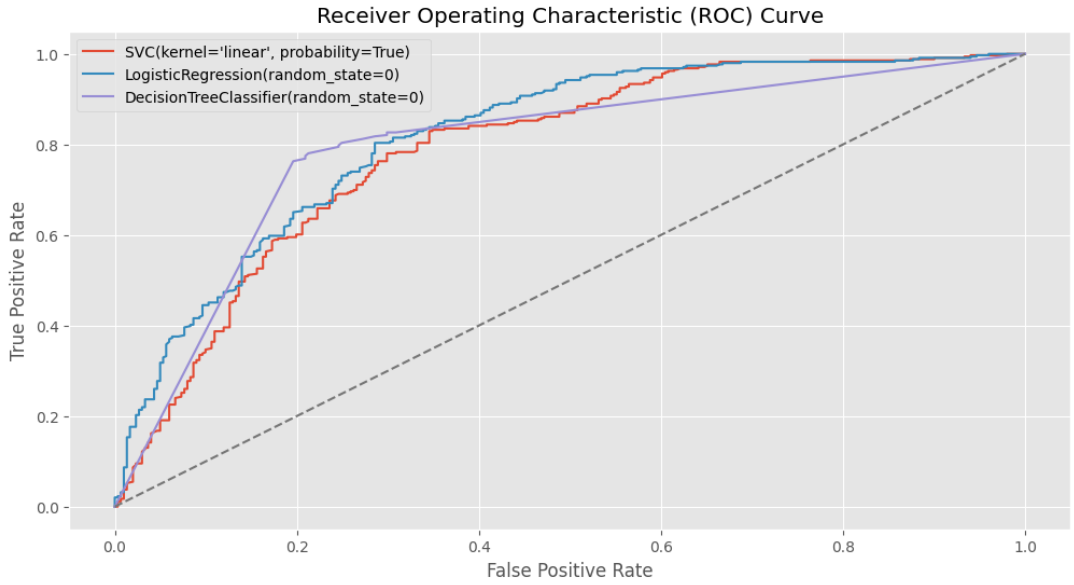

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.77 | 0.66 | 0.71 | 301 |
| 1 | 0.74 | 0.83 | 0.78 | 346 |
| accuracy | | | 0.75 | 647 |
| macro avg | 0.76 | 0.75 | 0.75 | 647 |
| weighted avg | 0.75 | 0.75 | 0.75 | 647 |

*Nota.* Elaboración propia.

En cuanto a la curva ROC, los resultados de este primer ejercicio son:

**Figura 19**

*Curva ROC para modelos corridos con elección de mejores parámetros*




*Nota.* Elaboración propia.

El modelo que presenta mejor sensibilidad tasa de verdaderos y positivos y menor tasa de falsos positivos es el del modelo de árbol de decisión, seguido por SVC y regresión logística.

## 5.2. Validación

la validación de los 3 modelos se realizó en un inicio dividiendo los datos en 80% de entrenamiento y un 20% de prueba, este proceso se realizó con train_test_split, más adelante como parte de los ejercicios de iteraciones que permiten aplicar diversos ejercicios en búsqueda de mejorar los resultados, la validación se realizará usando todos los datos, a través de validación cruzada, de esa forma se evaluará si es mejor usar todos los datos, como se esperaría, o si no hay mayor diferencia al aplicar esta iteración.

## 5.3. Iteraciones y evolución

Para la segunda iteración se corrieron los 3 modelos planteados, pero haciendo uso de validación cruzada, la validación cruzada es un método para entrenar modelos usando el total de

los datos disponibles. Flores, Nelly (2022) explica en su blog sobre validación cruzada que este método permite probar y evaluar los modelos en su rendimiento ya que permite hallar la mejor partición posible con los datos, lo que otorga mejores modelos.

En el blog de Data Science (2023) se plantea que el método más conocido es el de K-folds, este método consiste en darle participación a todos los datos en el ejercicio de prueba y entrenamiento, para ello se separan los datos aleatoriamente en K folds, el parámetro 'K' indica el número de grupos de división, este número de 'K' lo puede elegir de manera óptima funciones de scikit-learn.

Al aplicar este método se disminuye el sesgo de los modelos en sobreajuste, por ejemplo, es decir que cuando se agregan datos nuevos puede predecir eficientemente.

### 5.4 Herramientas

Las herramientas usadas para desarrollar el proyecto fueron google colab, gitHub y microsoft word. Primero se realiza el preprocesamiento y limpieza de los datos, el Análisis Estadístico Exploratorio EDA, se genera una la base de datos lista para el procesamiento de los modelos, llamada: data_final_balanceada en un archivo tipo CSV. Todos estos pasos más la ejecución de los modelos se realiza en el Notebook llamado Codigo_monografia_final, se plantean los pasos correspondientes a la implementación de los modelos, las iteraciones y los resultados finales. Si se desea consultar la base y el notebook, el link a Github se encuentra disponible en la hoja resumen.

## 6. Resultados y discusión

Las primeras iteraciones usando modelos con datos de entrenamiento de 80% y 20% de prueba arrojan buenas métricas de desempeño según las matrices de confusión y la curva ROC, en esta parte se usó elección automática de mejores parámetros para los modelos. El mejor modelo según el accuracy fue el de árbol de decisión con 21 capas y 573 nodos terminales, arrojando un 0.78 de accuracy, lo que, según la métrica de negocio esperada, permitió clasificar al 78% de los de los empleados, en la clase correcta de la base original, esto representaría una identificación de más del 70% de clasificación correcta.

En cuanto a la precisión, que se refiere a cuántos de los empleados que fueron bajas 1, realmente las clasificó como bajas, el resultado para esta métrica en la matriz de confusión fue del 0.80 es decir, está clasificando en un 80% bien las bajas, esta métrica es muy importante, porque para el contexto del negocio es preferible que clasifique como baja a alguien que en realidad no es baja, a que clasifique como activo a alguien que debería ser baja. que para este ejercicio de árbol de decisión representaría una clasificación del 24% de las personas como bajas cuando en realidad eran activos, mientras que clasificó como activo siendo baja en realidad, al 19%.

Estos resultados comparados con el primer ejercicio de elegir arbitrariamente los parámetros del árbol de decisión muestran un gran cambio, pasando de una precisión para clasificar bajas del 0.69 al 0.80.

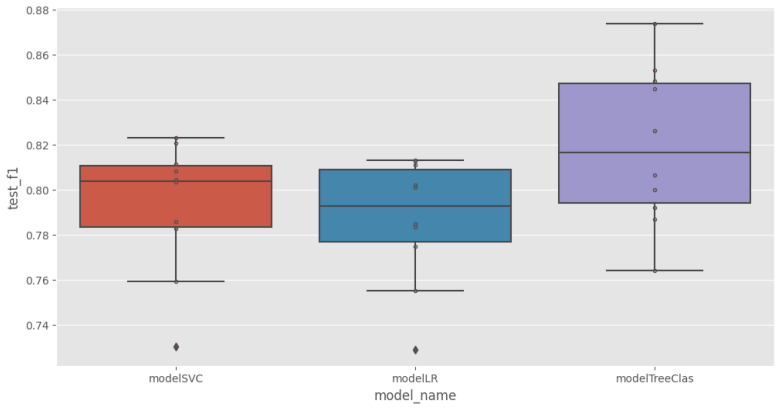
Estos resultados son satisfactorios frente a las métricas de negocio definidas al inicio, sin embargo, se decide realizar el ejercicio con los 3 modelos nuevamente, pero usando validación cruzada, este tipo de validación que no divide los datos en dos subconjuntos de entrenamiento y prueba, si no que confronta todos los datos que se tienen en el dataset, permite tener resultados mucho mejores.

## 6.1. Métricas

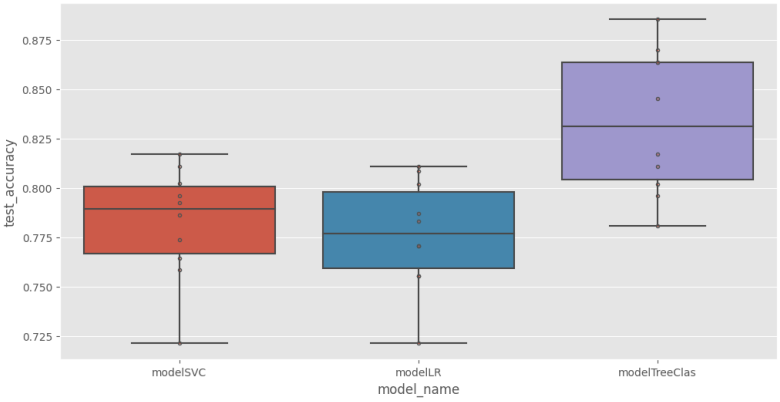
***Figura 20***

*Rendimiento de los 3 modelos según principales métricas con validación cruzada*

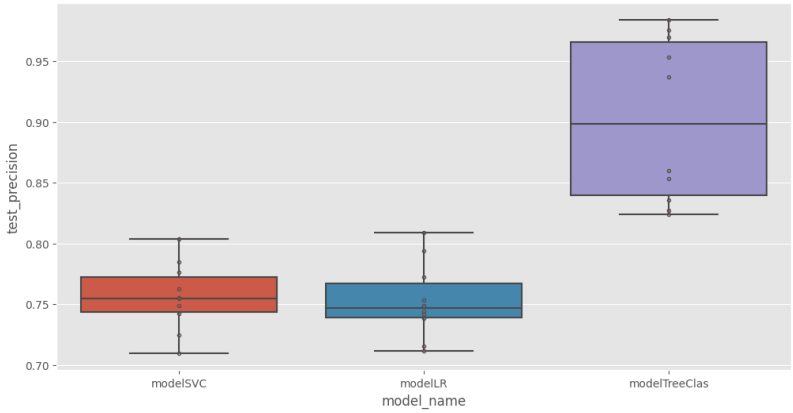
Tes_f1




Test_accuracy




Test_precision




*Nota.* Elaboración propia.

Estas gráficas resumen que el modelo de árbol de decisión posee las mejores métricas sobre los demás, el siguiente es el resumen:

**Figura 21**

*Resumen de los 3 modelos aplicados y sus métricas con validación cruzada*

```
model_name
modelTreeClas   0.8197
modelSVC        0.7930
modelLR         0.7868
Name: test_f1, dtype: float64
```

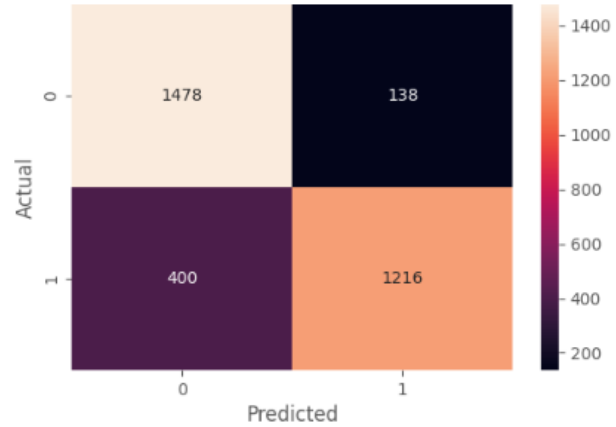
```
model_name
modelTreeClas   0.8336
modelSVC        0.7825
modelLR         0.7766
Name: test_accuracy, dtype: float64
```

```
model_name
modelTreeClas   0.9020
modelSVC        0.7562
modelLR         0.7529
Name: test_precision, dtype: float64
```

*Nota.* Elaboración propia.

**Figura 22**

*Matriz de confusión del modelo con mejor rendimiento con validación cruzada, árbol de decisión*

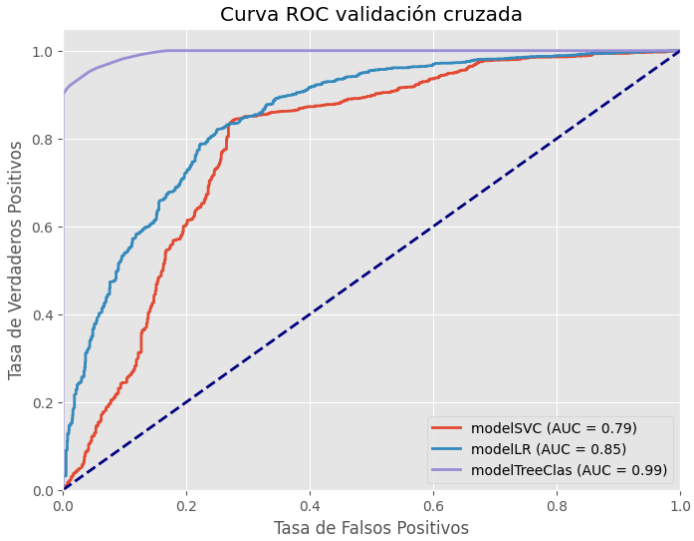



| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.79 | 0.91 | 0.85 | 1616 |
| 1 | 0.90 | 0.75 | 0.82 | 1616 |
| accuracy | | | 0.83 | 3232 |
| macro avg | 0.84 | 0.83 | 0.83 | 3232 |
| weighted avg | 0.84 | 0.83 | 0.83 | 3232 |

*Nota.* Elaboración propia.

**Figura 23**

*Curva ROC de los modelos ejecutados con validación cruzada*




*Nota.* Elaboración propia.

### 6.2. Evaluación cualitativa

Cómo se había mencionado anteriormente, para la compañía es muy importante que la precisión de las bajas, es decir, la clase 1, sea la mejor, porque es menos grave clasificar un activo como baja y realizar la intervención, así esta no sea necesaria por parte de la analista de GH, a que se clasifique una persona que va a ser baja como si fuera a estar activo y se perdería esa intervención por parte de GH. Usando la validación cruzada, es decir, todos los datos para entrenar el modelo, es capaz de predecir el 90% de las bajas. Por lo que este modelo de árbol de decisión contaría con las condiciones en cuanto impacto y mejoras en la producción para ser implementado en operación. En el área de Gestión Humana el objetivo de la identificación de personas que pueden ser posibles bajas es realizar intervenciones previas por parte de los psicólogos, el equipo de analistas de gestión humana está conformado por profesionales en psicología que realizan diversas intervenciones para identificar los aspectos que se puedan mejorar dentro de la compañía para retener el personal, bajo consigna de secreto, son un puente entre el empleado y su jefe, para tramitar conflictos, dudas, quejas etc. Por lo tanto, lo que se requiere de una efectiva identificación de empleados en riesgo de salir, es dirigir las intervenciones a las personas adecuadas, actualmente sin usar los modelos de machine learning, los analistas de gestión humana deben dirigir sus esfuerzos aleatoriamente, o aquellas personas que son referidas por el jefe.

Por lo tanto, la métrica de negocio a evaluar es la capacidad de retención que logre aumentar el equipo de gestión humana en las personas que son intervenidas, lo que se conoce como intervención de éxito, que la persona en riesgo permanezca al menos 3 meses más en la compañía. Esta métrica se mide como el 80% de retención sobre las personas a intervenir.

Una reducción de la rotación en 2% refleja un ahorro para la compañía de 10 mil USD trimestrales, además de los beneficios que no son medibles monetariamente, como trabajo eficiente y autónomo, reducción de tiempos de capacitación, etc.

### 6.3. Consideraciones de producción

El listado actualizado de activos y retirados de la empresa se actualiza diariamente a través de una ETL, teniendo en cuenta que se requieren los listados de posibles bajas, y que las intervenciones que programan las analistas se realizan por mes, este modelo debería contar con nuevos datos que incluyan ingresos, cambio de estado y bajas, cada mes, por lo que se quiere que genere un listado nuevo de personas, que no hayan sido intervenidas.

## 7. Conclusiones

El objetivo de este trabajo es plantear diferentes modelos de machine learning que permitan al área de Gestión Humana, cuáles son los empleados con mayores probabilidades de ser una baja en el futuro, se parte de una base de datos de pública de Kaggle que tiene 2.381 instancias con 1.616 bajas y un total de 765 empleados activos de una compañía que se dedica a la venta de seguros, con variables independientes como Gender, Education_Level, promoted, Salary_Range, Retention_Rango, age y Quarterly Rating. La variable de salida planteada toma valores de 0 si está activo la persona, 1 si es una baja.

Se corrieron 3 modelos de Machine Learning, un árbol de decisión, una máquina de vectores de soporte SVC y una regresión logística. En un primer momento usando criterios manuales se corrió un árbol de decisión que no arrojó la precisión deseada para la clasificación de la clase 1, posteriormente se corrieron los modelos usando funciones de skarlen que permiten definir automáticamente los mejores parámetros, con estos nuevos modelos se hizo un proceso de entrenamiento con el 80% de los datos, arrojando como resultado que el mejor modelo para clasificar los empleados era un árbol de decisión con 21 capas y 573 nodos terminales. obteniendo un accuracy que pasa de 0.72 a 0.78. Esto también se confirma con la curva ROC.

Para terminar de probar iteraciones que permitieran correr modelos con mejores resultados, se hace uso de la validación cruzada, este método ya no entrena partiendo los datos en dos grupos, uno de entrenamiento y otro de prueba, si no que hace uso de todos los datos contenidos en el dataset, validando con todos. Para este ejercicio se obtuvo un mejor rendimiento en todos los modelos, y nuevamente el árbol de decisión muestra las mejores métricas en comparación con los otros modelos, tanto para accuracy, cómo precisión y f1. Pasando de 0.78 a 0.80 de accuracy usando todos los datos para entrenar.

Lo más llamativo de este ejercicio final con validación cruzada, es que la precisión en detectar las bajas que son definidas como clase 1, es del 90% lo que significa que está clasificando las bajas muy bien, y esto es muy importante para el contexto del negocio, ya que es menos grave clasificar un activo como negativo, y realizar una intervención por parte del analista de GH que no

era requerida, a clasificar como activo a una persona que en realidad debía ser baja, ya que se pierde esta intervención.

Por último, al contrastar los resultados de las métricas del modelo de árbol de decisión con las métricas a considerar en el negocio, se llega a la conclusión que sí es un modelo exitoso para Gestión Humana, dado que al identificar las personas con mayores probabilidades de abandonar la compañía, los analistas de gestión humana, quienes son profesionales que se encargan de realizar actividades de intervención aleatorias a empleados, para mejorar permanencia, tendrían una herramienta mucho más exacta para enfocar sus intervenciones en las personas que más urgentemente requieren las mismas, es decir, los que podrían ser posibles bajas.

Cada intervención exitosa que realicen los analistas de retención, que se vea reflejado en una permanencia de 3 meses y una optimización de uso de recursos humanos, le permite a la compañía ahorrar miles de dólares en atracción de nuevo personal para cubrir vacantes por rotación. Según el experto de LLH una empresa líder global en reclutamiento, John Badel: “cuando una persona deja su posición, la empresa incurrirá en un costo estimado en 12 veces el valor del salario de esa persona”. Por lo que este ejercicio en tanto resultado de métricas de modelo como métricas de negocio, resultaría útil de implementar en la compañía.

## 8. Recomendaciones

Los resultados de los modelos obtenidos, Máquina de Vectores de soporte, árbol de decisión y Regresión Logística, especialmente cuando se aplicó validación cruzada fueron satisfactorios en términos de métricas. Resultando como el mejor modelo un árbol de decisión, este tipo de modelos son uno de los que mejores resultados ofrece a la hora de resolver problemas de clasificación, sin embargo, sería interesante la aplicación por ejemplo de modelos con redes neuronales para observar el comportamiento de las métricas, si bien este tipo de modelos presentan un alto grado de costo computacional, en futuros ejercicios sería interesante observar su comportamiento.

## 9. Referencias

Zelada, Carlos (2017). Evaluación de modelos de clasificación. https://rpubs.com/chzelada/275494.

Torres Luis (s.f.). Curva ROC y AUC en Python, https://www.themachinelearners.com/curva-roc-vs-prec-recall/#%C2%BFQue_es_la_curva_ROC.

Haya Pablo (s.f.). La metodología CRISP-DM en ciencia de datos, https://www.iic.uam.es/innovacion/metodologia-crisp-dm-ciencia-de-datos/.

IBM (s.f). Ventajas y desventajas de los árboles de decisión, https://www.ibm.com/es-es/topics/decision-trees.

Flores, Nelly (2022). Cross validation: qué es y su relación con machine Learning, https://blog.maestriasydiplomados.tec.mx/cross-validation-que-es-y-su-relacion-con-machine-learning.

Data Science (2023). Cross-Validation: definición e importancia en Machine Learning, https://blog.maestriasydiplomados.tec.mx/cross-validation-que-es-y-su-relacion-con-machine-learning.