## Línea horizontal

Propuesta de modelo predictivo para disminuir la rotación de personal

28.03.2022

**─**

Autor

Luis Fernando Angulo Pérez

Ivón Medrano Mendoza

# 

# 

# 

# 

# Introducción

“Research Lab” (Nombre ficticio) o la **empresa** contratante tiene un problema de rotación de personal. El departamento de Recursos Humanos ha dado a conocer que cerca del 15% de los empleados abandonan la compañía, esta rotación implica anualmente altos costos para la reposición de las plazas de trabajo y a su vez implica que se debe tener un equipo de RR.HH con un número elevado de integrantes para manejar todos los asuntos de los empleados.

Usando tecnicas de analisis de datos la empresa desea tener un modelo predictivo para tratar de predecir qué empleados corren el riesgo de irse de la empresa el siguiente año y hacer los ajustes que sean necesarios con cada empleado para que cada uno se sienta más motivado a continuar su labor dentro de la empresa.

Las técnicas utilizadas para tratar de resolver esta problemática incluyen Análisis Exploratorio de Datos, visualización y ajustes de un modelo de clasificación para predecir qué empleados pueden abandonar la compañía.

La presente propuesta de modelo predictivo fue realizada usando python y librerias que incluyen pero no se limitan a numpy, pandas, matplotlib, seaborn scilearn, sklearn y imblearn

# 

# 

# 

# 

# Datos

Los datos fueron obtenidos de la siguiente liga: [HR Analytics Case Study | Kaggle](https://www.kaggle.com/datasets/vjchoudhary7/hr-analytics-case-study) dado que la empresa es ficticia, con un total de 4410 observaciones.

Los datos proporcionados muestran la información del último año acerca de los empleados donde se proporcionan datos generales del empleado (edad, salario, género, distancia al trabajo…) adicional a esto la empresa ha recolectado una encuesta tanto del empleado sobre su trabajo como del gerente inmediato de cada empleado describiendo su desempeño. También se ha proporcionado información sobre los horarios de entrada y salida de cada empleado durante el periodo de estudio.

Los datos proporcionados en archivos por la empresa son:

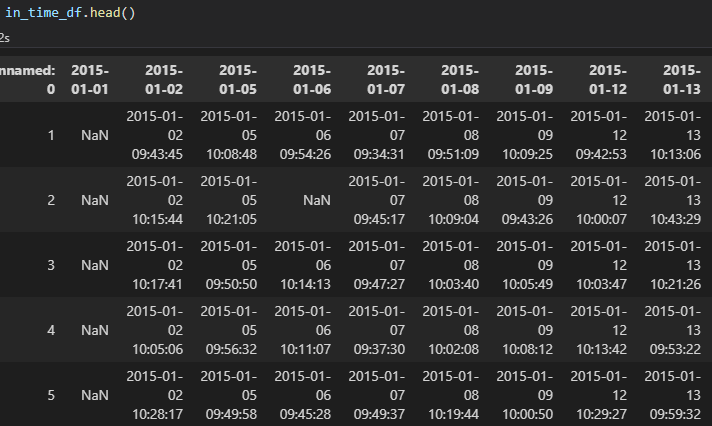
* general\_data.csv: Este archivo contiene información relevante sobre cada empleado
* employee\_survey\_data.csv: Es una encuesta para cada empleado en la que cada uno manifiesta su satisfacción del ambiente laboral, satisfacción del trabajo y balance vida-trabajo
* manager\_survey\_data.csv: Es una encuesta a cada gerente inmediato sobre los empleados a cargo e incluye información como: rendimiento e involucración del empleado en el trabajo
* in\_time: información sobre los horarios de llegada de cada empleado
* out\_time: información sobre los horarios de salida de cada empleado

**Manejando datos de horarios**

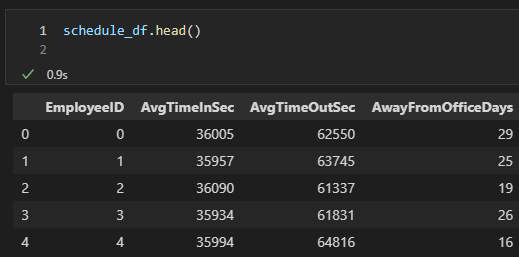
Los datos de entrada y salida nos permiten definir un comportamiento de llegada y salida del empleado como: ¿el empleado llega a tiempo al trabajo o llega tarde?, ¿El empleado cumple su horario laboral o hace horas extras?

Para este propósito se creó un algoritmo en el cual obtenemos la media aritmética en segundos de los horarios tanto de entrada como de salida y a su vez se almacenan la cantidad de días en los que no fue a trabajar en el periodo de estudio, los datos registraron el ingreso de cada día salvo fines de semana en los cuales la empresa no labora.

Ejemplo de los datos observados:



Al ejecutar el algoritmo obtenemos la media aritmética en segundos y los días que el empleado faltó a trabajar:



**Datos Generales**

Una vez se tienen los datos de entrada y salida solo basta con unirlos en un solo **dataframe** para facilitar la comprensión:

* Age: edad
* Attrition: Si el empleado ha abandonado la empresa
* BusinessTravel: Frecuencia de viajes del empleado
* Department: En qué departamento trabaja
* DistanceFromHome: Cual es la distancia en KM hacia el trabajo
* Education: Nivel educativo del empleado del uno al cinco
* EducationField: area de especialidad
* EmployeeID: id del empleado
* Gender: género
* JobLevel: nivel del trabajo a realizar
* JobRole: rol dentro de la empresa
* MaritalStatus: estatus marital
* MonthlyIncome: salario mensual
* NumCompaniesWorked: en cuantas compañías ha laborado
* Over18: si es mayor a 18
* PercentSalaryHike: En porcentaje ajuste salarial el último año
* StandardHours: horario de trabajo en horas
* StockOptionLevel: nivel de acciones dentro de la compañia
* TotalWorkingYears: cuántos años ha trabajado
* TrainingTimesLastYear: cuantos entrenamientos tuvo el ultimo periodo
* YearsAtCompany: cuantos años ha trabajado dentro de la empresa
* YearsSinceLastPromotion: hace cuanto fue promovido
* YearsWithCurrManager: cuanto lleva con el gerente actual
* JobInvolvement: cómo se involucra en el trabajo
* PerformanceRating: calificación de desempeño
* EnvironmentSatisfaction: Satisfacción con el ambiente laboral
* JobSatisfaction: Satisfacción con el trabajo
* WorkLifeBalance: Balance trabajo-vida
* AvgTimeInSec: promedio en segundos en llegar al trabajo
* AvgTimeOutSec: promedio en segundos en salir del trabajo
* AwayFromOfficeDays: Cuantos días no asistió a trabajar

Información gráfica en anexos

# 

# 

# 

# 

# 

# 

# Métodos utilizados

**EDA Análisis exploratorio de datos y limpieza de datos**

Para familiarizarse con los datos se hizo un análisis exploratorio de datos para entender la data, se hizo uso de las métricas proporcionadas por **Pandas**, para medir la cantidad de datos nulos, se utilizaron los valores de los cuartiles así como la media aritmética, se usaron técnicas de visualización con **matplotlib y seaborn** para comprender los datos.

**Transformación de variables continuas**

Algunas variables no presentaban evidencias de ser una distribución normal, las mismas presentan asimetrías a la izquierda, por lo que se realizaron transformaciones logarítmicas o exponenciales con el fin de darle simetría al comportamiento de la característica incluyendo transformaciones boxcox.

**Q-Q plots**

Se utilizaron gráficos Q-Q plots para verificar si los puntos se adherían a una distribución normal antes y después de las transformaciones.

**Transformación de variables categóricas nominales**

La data provista incluye muchas características o variables las cuales no pueden transformarse en una secuencia por lo que hubo que aplicar alguna de las siguientes técnicas.

**One-hot encoding**

**“La estrategia que implementa es crear una columna para cada valor distinto que exista en la característica que estamos codificando”[1].** La finalidad de esta técnica consiste en crear tantas columnas como valores únicos existan dejando un 1 en la columna correspondiente y 0 en el resto de las columnas. Un inconveniente de esta técnica es que creará tantas columnas como valores únicos existan.

**Binary encoding**

Otra técnica que nace a partir de One-hot encoding consiste en transformar usando un encoding ordinal las distintas categorías y posteriormente transformar las mismas a un valor binario separando cada posición en una columna distinta.

**One-hot encoding vs Binary encoding**

One-hot encoding construye tantas columnas como categorías existan en la columna original, esto es particularmente problemático ya que puede ocurrir que la cantidad de columnas de nuestro **dataset** aumente considerablemente y así mismo se corre el riesgo de sobreajuste el modelo a construir, por ende la decisión de usar one-hot encoding se limitó a tres categorías por columna donde fuera requerido. Binary encoding permite reducir la cantidad de columnas generadas sin embargo hay pérdida de información así que se usaron cuando las categorías son mayor a 3.

**Normalización de datos**

Una vez transformados los datos se puede observar que en algunas variables los datos suelen estar a escalas muy diferentes lo que puede perjudicar al momento de implementar un algoritmo de aprendizaje por ende fue necesario normalizar los datos usando un algoritmo de escalamiento mínimo-máximo de la librería sklearn:

x = df.values

min\_max\_scaler = preprocessing.MinMaxScaler()

x\_scaled = min\_max\_scaler.fit\_transform(x)

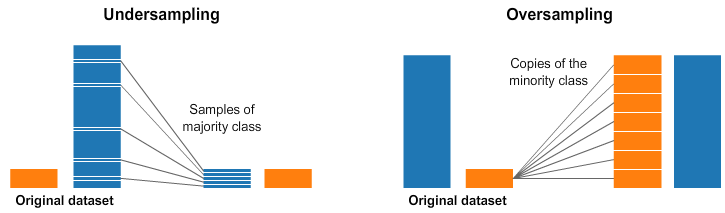
norm\_df = pd.DataFrame(x\_scaled, columns=df.columns)

**Balanceo de datos**

Al hacer el EDA se puede visualizar que la variable **Attrition** la cual identifica qué personal se ha ido de la empresa coincide con el reporte dado por HHRR en el que 15% de los empleados se han ido en el periodo de estudio, esto es particularmente problemático ya que los modelos de predicción binomial funcionan mejor cuando los datos se encuentran en partes iguales o cerca de este valor

**Undersampling Vs Oversampling**

Dos algoritmos populares para resolver una situación en donde la data está desbalanceada son undersampling el cual consiste en eliminar usando alguna estrategia elementos de la clase no minoritaria, esto conlleva a pérdida de información y también tenemos oversampling el cual consiste en de forma aleatoria ingresar registros a la clase minoritaria, esto conlleva a que la data pudiera estar sesgada.



**Modelo de clasificación binaria**

Debido a la naturaleza de los datos los algoritmos seleccionados para definir el modelo el cual mejor se ajuste a las necesidades de la empresa son:

* Regresión logística: “Es el proceso de modelar la probabilidad de obtener una salida discreta dada una entrada” [2]
* SVM (Support Vector Machine) con un kernel lineal: “SVM funciona correlacionando datos a un espacio de características de grandes dimensiones de forma que los puntos de datos se puedan categorizar, incluso si los datos no se puedan separar linealmente de otro modo” [3]

* Random Forests: “Un Random Forest es un conjunto de árboles de decisión combinados con [bagging](https://www.iartificial.net/ensembles-voting-bagging-boosting-stacking/#Bagging). Al usar bagging, lo que en realidad está pasando, es que distintos árboles ven distintas porciones de los datos. Ningún árbol ve todos los datos de entrenamiento. Esto hace que cada árbol se entrene con distintas muestras de datos para un mismo problema. De esta forma, al combinar sus resultados, unos errores se compensan con otros y tenemos una predicción que generaliza mejor.”[4]

# Resultados

**Análisis Exploratorio de datos EDA:**

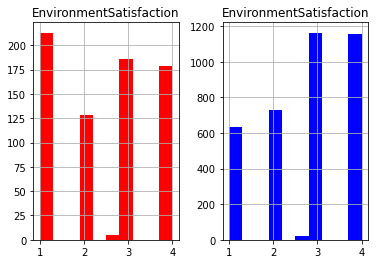
Se formularon las siguientes preguntas y se obtuvieron respuestas a partir de operaciones en el dataframe:

| Observación | Resultado | Visualización |
| --- | --- | --- |
| Edad de los empleados | * La mediana de la edad de los empleados es de 36 años * Rango de edad entre 18 y 60 años |  |
| Distancia al trabajo de los empleados | * La mediana de la distancia al trabajo es de 7Km * Hay personas que viven incluso a 29 Km * Cerca de la mitad de los empleados vive a menos de 7 Km de las oficinas. |  |
| Salario de los empleados | * Los empleados ganan mensualmente un salario que va desde los 10.000 hasta los 199.990. * La mediana salarial es de 49190 unidades monetarias |  |
| Permanencia de los empleados en la compañía | * La mediana indica que los empleados suelen durar 5 años |  |
| ¿Cuál es la distribución de género en la compañía? | * Alrededor del 40% del personal es de género femenino |  |
| ¿Cuántos días suele en promedio tomar los empleados? | * Los empleados suelen tomar en promedio 25 días al año incluyendo festivos pero sin contar fines de semanas. |  |
| ¿Cuál es el aumento porcentual a los empleados anualmente? | * Los empleados reciben en promedio un aumento del 15% de su remuneración |  |
| ¿Tienen los empleados acceso a acciones dentro de la compañía? | * La media indica que la mayoría de los empleados no tienen acciones que superen el nivel 1 |  |
| ¿Es la educación un factor en el salario? | * Estadísticamente el nivel educativo no pareciera aumentar las probabilidades de un mejor salario |  |
| ¿Es el nivel de trabajo un factor en el salario? | * Las personas que tienen un trabajo del nivel 4 suelen ganar mejor que los otros niveles. |  |
| Cual es el porcentaje de rotación en cada departamento | * Cada departamento tiene una rotación de cerca del 15% sin embargo el departamento de RRHH pareciera tener una rotación mucho mayor | Department  Human Resources 30.158730  Research & Development 15.712799  Sales 15.022422 |
| ¿Cuántos empleados no se fueron de la compañía por departamento? | * Después de estudiar el número de empleados en RRHH, se puede decir que debido al tamaño del departamento, el número es susceptible a cambios. | Department  Human Resources 132  Research & Development 2430  Sales 1137 |
| ¿Cuántos empleados se fueron de la compañía por departamento? |  | Department  Human Resources 57  Research & Development 453  Sales 201 |

Una pregunta hecha durante el EDA fue determinar si había una diferencia notable entre el comportamiento de las características de las personas que se fueron y de las que se quedaron. Haciendo uso de diagramas de densidad se vio el comportamiento de varias variables:

| ¿Existe una diferencia de edad entre las personas que se fueron y las que se quedan? | * Podemos inferir que las personas que se van suelen ser más jóvenes |  |
| --- | --- | --- |
| ¿Hay diferencia salarial entre ambos grupos? | * No se puede apreciar diferencia salarial |  |
| ¿Hay alguna diferencia entre la distancia al ir al trabajo? | * No hay diferencias notables en la distancia recorrida por los empleados |  |

Como tal las diferencias entre ambos grupos no fueron muy notables entre las características proporcionadas por la compañía, salvo en la satisfacción con el ambiente laboral, donde se puede apreciar en rojo los empleados que se fueron y en azul los que se quedaron. En la gráfica se puede ver que la satisfacción del ambiente laboral disminuyó para el grupo saliente. Más comparativas en Anexos



**Transformación de variables categóricas y normalización**

Variables Transformadas con la técnica One-hot encoding:

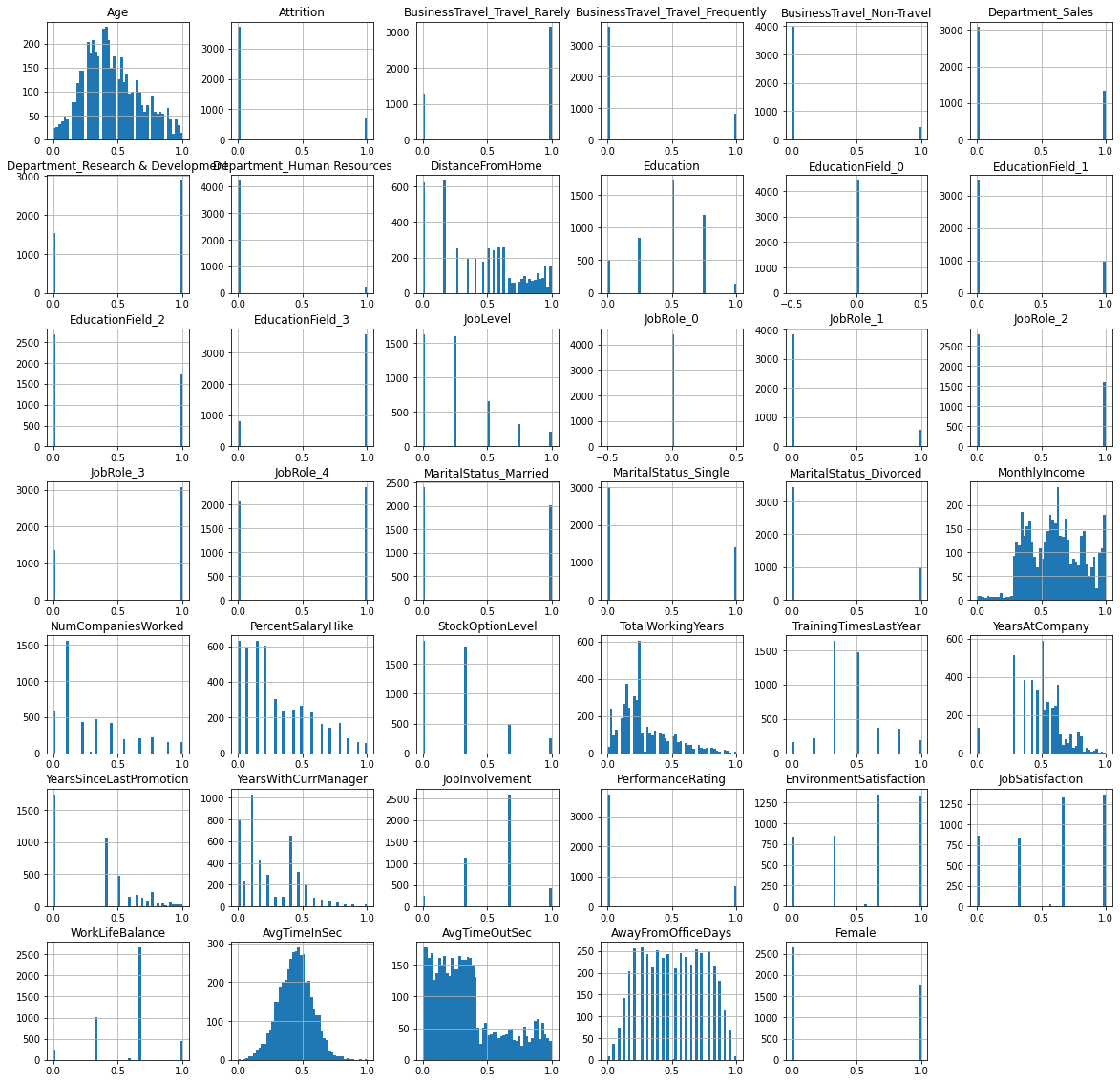
* BusinessTravel
* Department
* MaritalStatus

Variables transformadas usando Binary encoding:

* EducationField
* JobRole

**Datos normalizados**

Después de hacer la limpieza de los datos, eliminación de datos faltantes y transformaciones, tenemos los siguientes histogramas normalizados:

****

**Selección de un modelo de clasificación binaria**

Entre los algoritmos a seleccionar se tienen: **regresión logística**, **SVM** y **Random forests**. Se probaron los algoritmos.

**Undersampling NearMiss vs Oversampling SMOTE**

NearMiss es un algoritmo que mide la distancia entre varios puntos y su forma de actuar es eliminar puntos cercanos y de forma aleatoria con lo que queda la consistencia de los datos lo más similar posible. La técnica de sobremuestreo sintético en minorías (SMOTE) consiste en agregar de forma aleatoria registros para superar el desbalance agregando características “similares”.

**Regresión logística**

Con los datos preparados se probó la regresión logística usando el algoritmo NearMiss para el desbalanceo y se obtuvieron los siguientes datos:

NearMiss:

Precision: 0.765807962529274

precision recall f1-score support

0.0 0.77 0.77 0.77 217

1.0 0.76 0.76 0.76 210

accuracy 0.77 427

macro avg 0.77 0.77 0.77 427

weighted avg 0.77 0.77 0.77 427

Usando SMOTE:

precision recall f1-score support

0.0 0.72 0.69 0.70 1103

1.0 0.70 0.73 0.72 1117

accuracy 0.71 2220

macro avg 0.71 0.71 0.71 2220

weighted avg 0.71 0.71 0.71 2220

**SVM (linear kernel)**

Se utilizó el algoritmo SVM y se obtuvo los siguientes resultados:

precision recall f1-score support

0.0 0.79 0.79 0.79 217

1.0 0.78 0.78 0.78 210

accuracy 0.78 427

macro avg 0.78 0.78 0.78 427

weighted avg 0.78 0.78 0.78 427

Este modelo es ligeramente “mejor”

**Random Forest**

Este algoritmo fue optimizado para usar los el número de estimadores que mejor ajusta el modelo y se obtuvieron los siguientes resultados:

precision recall f1-score support

0.0 0.95 0.98 0.97 217

1.0 0.98 0.95 0.96 210

accuracy 0.96 427

macro avg 0.97 0.96 0.96 427

weighted avg 0.97 0.96 0.96 427

Este modelo ofrece mucha mejor precisión que los anteriores pero hay un riesgo inherente a usar árboles y es que el modelo puede estar sobreentrenado.

**Importancia de características**

Este modelo nos da las 10 más importantes características en el siguiente orden descendente:

| Predictor | Importancia |
| --- | --- |
| Age | 0.069722 |
| MonthlyIncome | 0.064440 |
| TotalWorkingYears | 0.060138 |
| AvgTimeOutSec | 0.056896 |
| YearsAtCompany | 0.048912 |
| DistanceFromHome | 0.048739 |
| PercentSalaryHike | 0.045514 |
| NumCompaniesWorked | 0.042165 |
| YearsWithCurrManager | 0.041379 |
| AvgTimeInSec | 0.033404 |

# Empleados con posibilidad de irse

Se ha generado una lista de todos los posibles empleados a retirarse de la empresa con una probabilidad mayor al 60%, esta lista ha sido almacenada en el repositorio

Una muestra de los primeros 10 empleados a los cuales hay que darle prioridad serían:

| **employeeId** | **probLeaving** |
| --- | --- |
| 434 | 0.933333 |
| 3374 | 0.933333 |
| 3744 | 0.926667 |
| 2274 | 0.920000 |
| 715 | 0.906667 |
| 804 | 0.906667 |
| 2185 | 0.906667 |
| 1904 | 0.906667 |
| 3655 | 0.893333 |
| 1230 | 0.893333 |

# 

# Conclusiones Los datasets con problemas de balanceo generan un gran problema al momento de elegir un modelo de clasificación binario. En la presente propuesta se hizo un proceso de análisis exploratorio de datos para tratar de conocer los datos, sin embargo algo que causa ruido es la falta de variación entre las características del personal que se queda y el personal que se retira de la empresa. La sugerencia del equipo creador de esta propuesta es de aumentar la información recolectada en la empresa que permita medir con mejor exactitud el bienestar percibido por cada empleado ya que en los modelos trabajados no se obtuvieron características que fueran claves al momento de realizar el modelo.

Algunas sugerencias es que en el momento de realizar el análisis no hubo tratamiento de **outliers** y la cantidad de características ascendía a más de 40 por lo que se sugiere repetir el estudio con análisis de componentes principales para reducir el número de columnas a trabajar, esto pudiera mejorar el modelo y evitar sobre ajustes. El modelo por otro lado no tuvo una verificación de sobreajuste el cual sería beneficioso realizar.

# 

# 

# 

# Referencias y repositorio principal

[1] *One-Hot encoding* https://interactivechaos.com/es/manual/tutorial-de-machine-learning/one-hot-encoding

[2] Thomas W. Edgar, David O. Manz (2017) *Logistic Regression*, https://www.sciencedirect.com/topics/computer-science/logistic-regression#:~:text=Logistic%20regression%20is%20a%20process,%2Fno%2C%20and%20so%20on.

[3] (2017), *Understanding Support Vector Machine(SVM) algorithm from examples (along with code)* , https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[4] Jose Martinez Heras, (2020), *Random Forest (Bosque Aleatorio)*: combinando árboles, <https://www.iartificial.net/random-forest-bosque-aleatorio/>

[5] (2021), *SMOTE,* [*https://docs.microsoft.com/en-us/azure/machine-learning/component-reference/smote*](https://docs.microsoft.com/en-us/azure/machine-learning/component-reference/smote)

[6]RAFAEL ALENCAR, (2018), *Resampling strategies for imbalanced datasets,* <https://www.kaggle.com/code/rafjaa/resampling-strategies-for-imbalanced-datasets/notebook#t4>

[7] Joaquín Amat Rodrigo, ( Octubre ,2020) *Random Forest con Python, https://www.cienciadedatos.net/documentos/py08\_random\_forest\_python.html*

**Repositorio principal**

En el repositorio se pueden encontrar los archivos de jupyter notebook con que se hicieron los cálculos además de un glosario con la explicación de cada variable proporcionada inicialmente

**https://github.com/luisfangulop/hr-attrition**

**Fuente de datos**

https://www.kaggle.com/datasets/vjchoudhary7/hr-analytics-case-study

# Anexos

En el presente se comparte las diferencias gráficas de las variables más relevantes, en rojo se grafican los empleados que se retiraron al finalizar el periodo de estudio y en azul los que se quedaron:

