**Reto: Deserción de empleados**

**Objetivo:**

A través de este reto vas a lograr:

* **Evaluar** las características de un conjunto de datos con el fin de ver si son suficientes para la actividad en la que serán usados.
* **Crear** nuevas características a partir de las existentes, que logren proporcionar una mayor información para la actividad en la que se usarán.
* **Eliminar** las características que sean irrelevantes para el uso que se le dará al conjunto de datos.

**Introducción:**

Una **empresa de productos médicos** está interesada en **atraer y mantener al mejor talento** porque sabe que es la clave del éxito para cualquier organización. También toma en cuenta que, si un empleado abandona la empresa, se está provocando una pérdida de tiempo y dinero debido, entre otras cosas, a la inversión en capacitación y a la experiencia acumulada del empleado. Desde luego que hay algunas formas de deserción que son inevitables, como cuando un empleado se retira o cambia de ciudad de residencia. Sin embargo, existen algunos factores que se pueden controlar por parte de la empresa con el objetivo de minimizar la deserción al mejorar las condiciones de trabajo. A la empresa le interesa saber **cuáles son los factores que hacen que un empleado siga con ellos** y cuáles son los que se deben cambiar debido a que provocan que los empleados se vayan.

Aunque la empresa sabe que este estudio tiene muchas aristas desea empezar con algo muy simple: determinando si un empleado abandonará la empresa **Attrition** o no en un momento dado. Como se trata de un problema de clasificación binaria, lo más probable es que utilicen un modelo de *machine* *learning*. La empresa ha logrado recolectar 30 datos de 400 de sus empleados, pero no está segura si ese conjunto de datos sean los correctos para lo que pretende hacer, por lo que decidió contratarte como científico de datos para **generar un set de datos adecuado para esta actividad**.

**Instrucciones:**

Después de una revisión completa de los datos con los que se cuentan, has decidido escribir un programa en Pyhton Jupyter Notebook llamado **RetoEmpleados.ipynb** bajo el siguiente plan:

1. **Importa** las librerías requeridas.
2. **Lee** el archivo CSV llamado **empleadosRETO.csv** y coloca los datos en un frame de Pandas llamado **EmpleadosAttrition**.
3. **Elimina** las columnas que, con alta probabilidad (estimada por ti), no tienen relación alguna con la salida. Hay algunas columnas que contienen información que no ayuda a definir el desgaste de un empleado, tal es caso de las siguientes:
   1. **EmployeeCount**: número de empleados, todos tienen un 1
   2. **EmployeeNumber**: ID del empleado, el cual es único para cada empleado
   3. **Over18**: mayores de edad, todos dicen “Y”
   4. **StandardHours**: horas de trabajo, todos tienen “80”
4. Analizando la información proporcionada, detectaste que no se cuenta con los años que el empelado lleva en la compañía y parece ser un buen dato. Dicha cantidad se puede calcular con la fecha de contratación ‘**HiringDate’**:
   1. **Crea** una columna llamada **Year** y obtén el año de contratación del empleado a partir de su fecha ‘**HiringDate’**. No se te olvide que debe ser un entero.
   2. **Crea** una columna llamada **YearsAtCompany** que contenga los años que el empleado lleva en la compañía hasta el año 2018. Para su cálculo, usa la variable **Year** que acabas de crear.
5. La **DistanceFromHome** está dada en kilómetros, pero tiene las letras “km” al final y así no puede ser entera:
   1. **Renombra** la variable **DistanceFromHome** a **DistanceFromHome\_km**.
   2. **Crea** una nueva variable **DistanceFromHome** que sea entera, es decir, solo con números.
6. **Borra** las columnas **Year**, **HiringDate** y **DistanceFromHome\_km** debido a que ya no son útiles.
7. Aprovechando los ajustes que se están haciendo, la empresa desea saber si todos los departamentos tienen un ingreso promedio similar. **Genera** una nuevo frame llamado **SueldoPromedioDepto** que contenga el **MonthlyIncome** promedio por departamento de los empleados y colócalo en una variable llamada **SueldoPromedio**. Esta tabla solo es informativa, no la vas a utilizar en el set de datos que estás construyendo.
8. La variable **MonthlyIncome** tiene un valor numérico muy grande comparada con las otras variables. **Escala** dicha variable para que tenga un valor entre 0 y 1.
9. Todo parece indicar que las variables categóricas que quedan sí son importantes para obtener la variable de salida. **Convierte** todas las variables categóricas que quedan a numéricas:
10. **BusinessTravel**
11. **Department**
12. **EducationField**
13. **Gender**
14. **JobRole**
15. **MaritalStatus**
16. **Attrition**
17. Ahora debes hacer la evaluación de las variables para quedarte con las mejores. **Calcula** la correlación lineal de cada una de las variables con respecto al **Attrition**.
18. **Selecciona** solo aquellas variables que tengan una correlación mayor o igual a 0.1, dejándolas en otro frame llamado **EmpleadosAttritionFinal**. No olvides mantener la variable de salida **Attrition;** esto es equivalente a borrar las que no cumplen con el límite.
19. **Crea** una nueva variable llamada **EmpleadosAttritionPCA** formada por los componentes principales del frame **EmpleadosAttritionFinal**. Recuerda que el resultado del proceso PCA es un numpy array, por lo que, para hacer referencia a una columna, por ejemplo, la 0, puedes usar la instrucción **EmpleadosAttritionPCA[:,0])**.
20. **Agrega** el mínimo número de Componentes Principales en columnas del frame **EmpleadosAttritionPCA** que logren explicar el 80% de la varianza, al frame **EmpleadosAttritionFinal**. Puedes usar la instrucción **assign**, columna por columna, llamando a cada una **C0**, **C1**, etc., hasta las que vayas a agregar.
21. **Guarda** el set de datos que has formado y que tienes en **EmpleadosAttritionFinal** en un archivo CSV llamado **EmpleadosAttritionFinal.csv**. Las últimas columnas que colocaste quedarán después de la variable **Attrition**, lo cual no importa, pero si gustas lo puedes arreglar antes de escribir el archivo.
22. Descarga tu script (archivo con extensión .ipynb) y guárdalo en un archivo que siga la nomenclatura que se te indica en *Formato de entrega de actividad*.
23. Sube el archivo a  [Github](https://github.com/).
24. Descarga el archivo de rúbrica y compara tus resultados con ella.
25. Publica tu archivo presionando el ícono  ubicado en la parte inferior de la pantalla de Plataforma.

**Formato de entrega de actividad:**

Tu trabajo será evaluado con base en el cumplimiento de cada uno de los pasos indicados en las instrucciones, hasta llegar a tu archivo final. Como evidencia deberás entregar:

* Tu programa en Pyhton en un archivo llamado **RetoEmpleados.ipynb** Jupyter notebook con todas las instrucciones necesarias para generar tu archivo final.
* El mismo programa, pero en formato pdf, en un archivo llamado **RetoEmpleados.pdf**.
* El archivo final obtenido (**EmpleadosAttritionFinal.csv**).