Importa las librerías requeridas. Lee el archivo CSV llamado empleadosRETO.csv y coloca los datos en un frame de Pandas llamado EmpleadosAttrition.

```
import pandas as pd
from google.colab import drive

drive.mount('/content/drive')
EmpleadosAttrition = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/empleadosRETO.csv")
```

```
Mounted at /content/drive
```

```
EmpleadosAttrition
```

	Age	BusinessTravel	Department	DistanceFromHome	Education	EducationField
0	50	Travel_Rarely	Research & Development	1 km	2	Medical
1	36	Travel_Rarely	Research & Development	6 km	2	Medical
2	21	Travel_Rarely	Sales	7 km	1	Marketing
3	52	Travel_Rarely	Research & Development	7 km	4	Life Sciences
4	33	Travel_Rarely	Research & Development	15 km	1	Medical
...	...	...	...	...	...	...
395	33	Travel_Rarely	Research & Development	14 km	3	Medical
396	31	Travel_Rarely	Sales	20 km	3	Life Sciences
397	37	Travel_Frequently	Research & Development	11 km	3	Other
398	38	Travel_Rarely	Research & Development	4 km	2	Medical
399	33	Travel_Rarely	Research & Development	14 km	3	Medical

400 rows × 30 columns

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:

- EmployeeCount: número de empleados, todos tienen un 1
- EmployeeNumber: ID del empleado, el cual es único para cada empleado
- Over18: mayores de edad, todos dicen “Y”
- StandardHours: horas de trabajo, todos tienen “80”

```
EmpleadosAttrition.drop(columns=['EmployeeCount','EmployeeNumber','Over18','StandardHours'],inplace=True)
EmpleadosAttrition
```

	Age	BusinessTravel	Department	DistanceFromHome	Education	EducationField
0	50	Travel_Rarely	Research & Development	1 km	2	Medical
1	36	Travel_Rarely	Research & Development	6 km	2	Medical
2	21	Travel_Rarely	Sales	7 km	1	Marketing

Analiza la información proporcionada, si 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'. 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.

```
EmpleadosAttrition['Year']=EmpleadosAttrition['HiringDate'].str[-4:].astype(int)
EmpleadosAttrition['Year']
```

```
0      2013
1      2015
2      2017
3      2010
4      2011
       ...
395    2013
396    2016
397    2008
398    2018
399    2010
Name: Year, Length: 400, dtype: int64
```

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.

```
EmpleadosAttrition['YearsAtCompany']=2018-EmpleadosAttrition['Year']
```

```
EmpleadosAttrition['YearsAtCompany']
```

```
0       5
1       3
2       1
3       8
4       7
       ..
395     5
396     2
397    10
398     0
399     8
Name: YearsAtCompany, Length: 400, dtype: int64
```

La DistanceFromHome está dada en kilómetros, pero tiene las letras "km" al final y así no puede ser entera. Renombra la variable DistanceFromHome a DistanceFromHome_km.

```
EmpleadosAttrition.rename(columns={'DistanceFromHome':'DistanceFromHome_km'},inplace=True)
```

Crea una nueva variable DistanceFromHome que sea entera, es decir, solo con números.

```
EmpleadosAttrition['DistanceFromHome']=EmpleadosAttrition['DistanceFromHome_km'].str[:2].astype(int)
EmpleadosAttrition['DistanceFromHome']
```

Borra las columnas Year, HiringDate y DistanceFromHome_km debido a que ya no son útiles.

```
EmpleadosAttrition.drop(columns=['Year','HiringDate','DistanceFromHome_km'],inplace=True)
```

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.

```
SueldoPromedioDepto = EmpleadosAttrition.groupby(['Department'])['MonthlyIncome'].mean()
SueldoPromedioDepto
```

```
Department
Human Resources           6239.888889
```

```
Research & Development    6804.149813
Sales                     7188.250000
Name: MonthlyIncome, dtype: float64
```

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.

```
EmpleadosAttrition['MonthlyIncome_Norm']=(EmpleadosAttrition['MonthlyIncome']-min(EmpleadosAttrition['MonthlyIncome']))/(max(EmpleadosAt
```

```
EmpleadosAttrition['MonthlyIncome_Norm']
```

```
0      0.864269
1      0.207340
2      0.088062
3      0.497574
4      0.664470
         ...
395    0.075248
396    0.187197
397    0.589327
398    0.121124
399    0.092122
Name: MonthlyIncome_Norm, Length: 400, dtype: float64
```

Convirtiendo a variables Categoricas

Valores para BusinessTravel:

- Travel_Rarely 0
- Non-Travel 1
- Travel_Frequently 2
- (Vacio) 3

```
EmpleadosAttrition['Cat_BusinessTravel']=[0, 0, 0, 0, 0, 0, 0, 1, 0, 2, 0, 2, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2
```

Valores para Department

- Research & Development 0
- Sales 1
- Human Resources 2

```
EmpleadosAttrition['Cat_Department']=[0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

Valores para EducationField

- Medical 0
- Marketing 1
- Life Sciences 2
- Technical Degree 3
- Other 4
- Human Resources 5

```
EmpleadosAttrition['Cat_EducationField']=[0, 0, 1, 2, 0, 2, 2, 2, 1, 2, 2, 2, 0, 2, 1, 2, 3, 1, 0, 0, 0, 2, 2, 2, 0, 2, 0, 2, 2, 0, 0, 0
```

Valores para Gender

- Male 0
- Female 1

```
EmpleadosAttrition['Cat_Gender']=[0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0,
```

Valores para JobRole

- Research Director 0
- Manufacturing Director 1
- Sales Representative 2
- Healthcare Representative 3
- Manager 4
- Sales Executive 5

- Laboratory Technician 6
- Research Scientist 7
- Human Resources 8

```
EmpleadosAttrition['Cat_JobRole']=[0, 1, 2, 3, 4, 1, 1, 5, 5, 5, 6, 1, 7, 5, 4, 5, 6, 5, 4, 4, 7, 3, 1, 6, 1, 1, 7, 7, 0, 7, 6, 7, 0, 6,
```

Valores para MaritalStatus:

- Divorced 0
- Single 1
- Married 2
- (Vacio) 3

```
EmpleadosAttrition['Cat_MaritalStatus']=[0, 0, 1, 1, 2, 0, 2, 0, 3, 1, 2, 2, 1, 2, 2, 2, 2, 0, 1, 0, 0, 2, 1, 2, 2, 2, 1, 2, 0, 2, 2, 0,
```

Valores para Attrition:

- No 0
- Yes 1

```
EmpleadosAttrition['Cat_Attrition']=[0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0,
```

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.

```
EmpleadosAttrition['Cat_BusinessTravel'].corr(EmpleadosAttrition['Cat_Attrition'])
```

```
-0.016856345954626616
```

```
EmpleadosAttrition['Cat_Department'].corr(EmpleadosAttrition['Cat_Attrition'])
```

```
0.07167282559028201
```

```
EmpleadosAttrition['Cat_EducationField'].corr(EmpleadosAttrition['Cat_Attrition'])
```

```
0.07374800176958936
```

```
EmpleadosAttrition['Cat_Gender'].corr(EmpleadosAttrition['Cat_Attrition'])
```

```
0.02883870932201118
```

```
EmpleadosAttrition['Cat_JobRole'].corr(EmpleadosAttrition['Cat_Attrition'])
```

```
0.09858311178361483
```

```
EmpleadosAttrition['Cat_MaritalStatus'].corr(EmpleadosAttrition['Cat_Attrition'])
```

```
0.0013734698460160315
```

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.

```
EmpleadosAttritionFinal=EmpleadosAttrition[['Cat_Department','Cat_EducationField','Cat_Gender','Cat_JobRole','Cat_Attrition']]
EmpleadosAttritionFinal
```

	Cat_Department	Cat_EducationField	Cat_Gender	Cat_JobRole	Cat_Attrition
0	0	0	0	0	0
1	0	0	0	1	0
2	1	1	0	2	1

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].

```
from sklearn.decomposition import PCA
pca=PCA(5)
#EmpleadosAttritionPCA=EmpleadosAttritionFinal
pca.fit(EmpleadosAttritionFinal)
```

```
PCA(n_components=5)
```

```
print(pca.components_)
```

```
[[-0.03240823 -0.07481542  0.01254557 -0.99645542 -0.01647901]
 [ 0.09411408  0.99209059  0.02113758 -0.07762404  0.02065227]
 [ 0.94872444 -0.099439    0.28878148 -0.02105572  0.07871298]
 [ 0.29199798 -0.00833007 -0.95572221 -0.02042315 -0.02908165]
 [-0.06893013 -0.01419194 -0.05095219 -0.01380761  0.99612271]]
```

```
print(pca.explained_variance_)
```

```
[5.4931882  1.50640411 0.26028451 0.24196303 0.13521529]
```

```
EmpleadosAttritionPCA=pca.transform(EmpleadosAttritionFinal)
```

```
EmpleadosAttritionPCA
```

```
array([[ 4.35713661, -1.15150959, -0.23826493,  0.40389011, -0.03906854],
       [ 3.36068118, -1.22913363, -0.25932065,  0.38346696, -0.05287615],
       [ 2.24052309, -0.19990074,  0.64762204,  0.61763007,  0.84631687],
       ...,
       [ 4.05787491,  2.81685276, -0.63602095,  0.37056982, -0.09583632],
       [-1.60905036, -1.59611624, -0.07581779, -0.674371  , -0.17286637],
       [-2.60550579, -1.67374028, -0.09687351, -0.69479416, -0.18667398]])
```

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.

```
EmpleadosAttritionFinal.insert(1,"C0",EmpleadosAttritionPCA[:,0])
EmpleadosAttritionFinal.insert(3,"C1",EmpleadosAttritionPCA[:,1])
```

Guarda el set de datos que has formado y que tienes en EmpleadosAttritionFinal en un archivo CSV llamado EmpleadosAttritionFinal.csv.

```
EmpleadosAttritionFinal.to_csv("EmpleadosAttritionFinal.csv",index=False)
```

✓ 0 s completado a las 15:47