

```
#####  
#----Reto: Deserción de empleados----- #  
# Elaborado por:Claudia Aranda Roque #  
# Fecha: 07-08-2024 #  
#####
```

P R O C E S O:

#1. Librerías Requeridas

```
import pandas as pd  
from datetime import datetime  
from sklearn.preprocessing import OneHotEncoder  
from sklearn.preprocessing import LabelBinarizer  
from sklearn.preprocessing import LabelEncoder  
from sklearn.preprocessing import StandardScaler  
from sklearn.decomposition import PCA
```

#2. Leer archivo CSV

```
EmpleadosAttrition = pd.read_csv('/Users/macbook/Documents/claudia  
personal/learning gate/Ingenieria de conceptos/Final/empleadosRETO.csv')
```

#3. Eliminar columnas

```
EmpleadosAttrition.drop(['EmployeeCount',  
'EmployeeNumber','Over18','StandardHours'], axis=1, inplace=True)
```

#4. Crear columnas

```
EmpleadosAttrition['Year'] = EmpleadosAttrition['HiringDate'].str[-4:].astype(int)  
today = datetime.today()  
EmpleadosAttrition['YearsAtCompany'] = 2018 - EmpleadosAttrition['Year']
```

#8. Renombrar columnas

```
EmpleadosAttrition =  
EmpleadosAttrition.rename(columns={'DistanceFromHome':'DistanceFromHome  
_km'})
```

#9. Crear nueva columna

```
EmpleadosAttrition['DistanceFromHome']  
=EmpleadosAttrition['DistanceFromHome_km'].str[:-3].astype(int)
```

#10. Borrar columnas

```
EmpleadosAttrition.drop(['DistanceFromHome_km', 'HiringDate','Year'], axis=1,  
inplace=True)
```

#11. Nuevo Frame

```
SueldoPromedioDepto =  
EmpleadosAttrition.groupby(['Department'])[['MonthlyIncome']].mean()  
SueldoPromedioDepto['SueldoPromedio']=  
EmpleadosAttrition.groupby(['Department'])[['MonthlyIncome']].mean()  
SueldoPromedioDepto.drop(['MonthlyIncome'], axis=1, inplace=True)  
SueldoPromedio = SueldoPromedioDepto
```

#12. Escalar MonthlyIncome

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

```

EmpleadosAttrition['MonthlyIncome'].min())
#13. Conversión de valores categóricas a numéricas
col_int = ['BusinessTravel', 'Department',
'EducationField', 'Gender', 'JobRole', 'MaritalStatus', 'Attrition', 'OverTime']
label_encoders = {}
for column in col_int:
    le = LabelEncoder()
    EmpleadosAttrition[column] = le.fit_transform(EmpleadosAttrition[column])
    label_encoders[column] = le
#14. Calculo de correlación lineal
correlation_attr= EmpleadosAttrition.corr()['Attrition'].drop('Attrition')
#15. Selección de variables con correlación mayor o igual a 0.1
EmpleadosAttritionFinal =
EmpleadosAttrition.drop(['BusinessTravel', 'Department', 'Education',
'EducationField', 'Gender', 'JobRole', 'NumCompaniesWorked', 'PercentSalaryHike',
'PerformanceRating', 'RelationshipSatisfaction', 'TrainingTimesLastYear', 'WorkLife
Balance', 'YearsSinceLastPromotion', 'DistanceFromHome'], axis=1)
#16. Nueva variable PCA
scaler = StandardScaler()
EmpleadosAttritionScaler = scaler.fit_transform(EmpleadosAttritionFinal)
EmpleadosAttritionPCA = PCA()
EmpleadosAttritionPCA.fit(EmpleadosAttritionScaler)
print('Varianza explicada')
explained_variance_ratio = EmpleadosAttritionPCA.explained_variance_ratio_
print(explained_variance_ratio)
# Inicializa la suma acumulativa y el contador de componentes
cumulative_variance = 0
n_components = 0
# Itera sobre la varianza explicada por cada componente
for i, variance in enumerate(explained_variance_ratio):
    cumulative_variance += variance
    n_components += 1
    if cumulative_variance >= 0.80:
        break
pca = PCA(n_components=n_components)
principal_components = pca.fit_transform(EmpleadosAttritionScaler)
df_pca = pd.DataFrame(data=principal_components, columns=[f'PC{i+1}' for i in
range(n_components)])
EmpleadosAttritionFinal = EmpleadosAttritionFinal.assign(**df_pca)
#18. Guardar en Archivo CSV
EmpleadosAttritionFinal.to_csv('/Users/macbook/Documents/claudia
personal/learning gate/Ingenieria de
conceptos/Final/EmpleadosAttritionFinal.csv', index=False)

```