```
# Conectar con unidad de drive donde esta guardado el csv con los
datos
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
```

```
import pandas as pd

# Leer el archivo con pandas
file_path = '/content/drive/MyDrive/certificado DS/empleadosRETO.csv'
EmpleadosAttrition = pd.read_csv(file_path)

irrelevantColumns = ["EmployeeCount", "EmployeeNumber", "Over18",
"StandardHours"]
FactRel = EmpleadosAttrition.drop(irrelevantColumns, axis=1) #Factores
Relevantes

FactRel["Year"] =
FactRel["HiringDate"].str.split("/").str[2].astype(int)
FactRel["YearsAtCompany "] = 2018 - FactRel["Year"]

FactRel = FactRel.rename(columns={'DistanceFromHome':
'DistanceFromHome_km'})
FactRel["DistanceFromHome"] = FactRel["DistanceFromHome_km"].str[:-
3].astype(int)

irrelevantColumns = ["Year", "HiringDate", "DistanceFromHome_km"]
FactRel = FactRel.drop(irrelevantColumns, axis=1)

SueldoPromedioDepto = FactRel.groupby("Department")
["MonthlyIncome"].mean()
SueldoPromedioDepto.name = "SueldoPromedio"
SueldoPromedioDepto.reset_index()
```


```
{"summary":"{\n  \"name\": \"SueldoPromedioDepto\",\n  \"rows\": 3,\n
\"fields\": [\n    {\n      \"column\": \"Department\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 3,\n        \"samples\": [\n          \"Human
Resources\",\n          \"Research & Development\",\n
\"Sales\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"SueldoPromedio\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 477.0241330555241,\n        \"min\":
6239.888888888889,\n        \"max\": 7188.25,\n
\"num_unique_values\": 3,\n        \"samples\": [\n
6239.888888888889,\n          6804.149812734083,\n          7188.25\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```

```
# Este tipo de escalamiento es el mas adecuado?
FactRel['MonthlyIncome'] = FactRel['MonthlyIncome'] /
FactRel['MonthlyIncome'].max()

# Codificar variables categoricas
'''
a.    BusinessTravel  Ordinal (since it's frequency + HAS NAN)
b.    Department      Nominal
c.    EducationField  Nominal (is this an error?, 'Technical Degree'
appears as a field)
d.    Gender          Nominal
e.    JobRole         Nominal (there are hierarchies, but it ain't
trivial ordering roles)
f.    MaritalStatus   Nominal (no clear order either + HAS NAN)
g.    Attrition       Yes-T, No-F
'''

businessTravelOrder = ['Non-Travel', 'Travel_Rarely',
'Travel_Frequently']
FactRel['BusinessTravel'] = pd.Categorical(FactRel['BusinessTravel'],
categories=businessTravelOrder, ordered=True)
FactRel['BusinessTravelEncoded'] = FactRel['BusinessTravel'].cat.codes

print(FactRel[['BusinessTravelEncoded', 'BusinessTravel']].head(10))
FactRel.drop('BusinessTravel', axis=1, inplace=True)
```

```
   BusinessTravelEncoded     BusinessTravel
0                      1      Travel_Rarely
1                      1      Travel_Rarely
2                      1      Travel_Rarely
3                      1      Travel_Rarely
4                      1      Travel_Rarely
5                      1      Travel_Rarely
6                      1      Travel_Rarely
7                      0         Non-Travel
8                      1      Travel_Rarely
9                      2  Travel_Frequently
```

```
# One-hot encoding con pandas para las nominales
nominalColumns = ['Department', 'EducationField', 'Gender', 'JobRole',
'MaritalStatus']
FactRel = pd.get_dummies(FactRel, columns=nominalColumns)

FactRel['OverTimeEncoded'] = FactRel['OverTime'].replace({'Yes': True,
'No': False})
print(FactRel[['OverTime', 'OverTimeEncoded']])
FactRel.drop('OverTime', axis=1, inplace=True)

FactRel['AttritionEncoded'] = FactRel['Attrition'].replace({'Yes':
True, 'No': False})
```

```
print(FactRel[['Attrition', 'AttritionEncoded']])
FactRel.drop('Attrition', axis=1, inplace=True)
```

```
    OverTime  OverTimeEncoded
0         No            False
1         No            False
2         No            False
3         No            False
4        Yes             True
..       ...              ...
395      Yes             True
396      Yes             True
397      Yes             True
398       No            False
399       No            False

[400 rows x 2 columns]
    Attrition  AttritionEncoded
0          No             False
1          No             False
2         Yes              True
3          No             False
4         Yes              True
..        ...               ...
395       Yes              True
396       Yes              True
397        No             False
398        No             False
399        No             False

[400 rows x 2 columns]
```

```
<ipython-input-102-777d60554080>:1: FutureWarning: Downcasting
behavior in `replace` is deprecated and will be removed in a future
version. To retain the old behavior, explicitly call
`result.infer_objects(copy=False)`. To opt-in to the future behavior,
set `pd.set_option('future.no_silent_downcasting', True)`
  FactRel['OverTimeEncoded'] = FactRel['OverTime'].replace({'Yes':
True, 'No': False})
<ipython-input-102-777d60554080>:5: FutureWarning: Downcasting
behavior in `replace` is deprecated and will be removed in a future
version. To retain the old behavior, explicitly call
`result.infer_objects(copy=False)`. To opt-in to the future behavior,
set `pd.set_option('future.no_silent_downcasting', True)`
  FactRel['AttritionEncoded'] = FactRel['Attrition'].replace({'Yes':
True, 'No': False})
```

```
correlations = FactRel.corr()
attritionCorrelations =
```

```
correlations['AttritionEncoded'].drop('AttritionEncoded')
print(attritionCorrelations)
```

```
Age                                  -0.212121
Education                            -0.055531
EnvironmentSatisfaction              -0.124327
JobInvolvement                       -0.166785
JobLevel                             -0.214266
JobSatisfaction                      -0.164957
MonthlyIncome                        -0.194936
NumCompaniesWorked                   -0.009082
PercentSalaryHike                    -0.060880
PerformanceRating                    -0.006471
RelationshipSatisfaction             -0.030945
TotalWorkingYears                    -0.213329
TrainingTimesLastYear                -0.070884
WorkLifeBalance                      -0.021723
YearsInCurrentRole                   -0.203918
YearsSinceLastPromotion              -0.069000
YearsAtCompany                       -0.176001
DistanceFromHome                      0.052732
BusinessTravelEncoded                 0.091336
Department_Human Resources            0.023389
Department_Research & Development    -0.072269
Department_Sales                      0.066116
EducationField_Human Resources        0.043404
EducationField_Life Sciences         -0.027457
EducationField_Marketing              0.016768
EducationField_Medical               -0.054144
EducationField_Other                 -0.004275
EducationField_Technical Degree       0.129104
Gender_Female                         0.028839
Gender_Male                          -0.028839
JobRole_Healthcare Representative    -0.103274
JobRole_Human Resources               0.032714
JobRole_Laboratory Technician         0.125264
JobRole_Manager                      -0.089885
JobRole_Manufacturing Director       -0.042404
JobRole_Research Director            -0.116263
JobRole_Research Scientist            0.007977
JobRole_Sales Executive              -0.003115
JobRole_Sales Representative          0.191294
MaritalStatus_Divorced               -0.107869
MaritalStatus_Married                -0.094734
MaritalStatus_Single                  0.205849
OverTimeEncoded                       0.324777
Name: AttritionEncoded, dtype: float64
```

```
correlation = FactRel.corr()["AttritionEncoded"]
correlationFiltered = correlation[correlation > 0.1]
```

```
print(correlationFiltered)

EmpleadosAttritionFinal = FactRel[correlationFiltered.index]
```

```
EducationField_Technical Degree    0.129104
JobRole_Laboratory Technician      0.125264
JobRole_Sales Representative       0.191294
MaritalStatus_Single               0.205849
OverTimeEncoded                    0.324777
AttritionEncoded                   1.000000
Name: AttritionEncoded, dtype: float64
```

```
from sklearn.decomposition import PCA

pca = PCA()
EmpleadosAttritionPCA = pca.fit_transform(EmpleadosAttritionFinal)
print(pca.explained_variance_ratio_)
```

```
[0.3123714  0.24768714 0.18718102 0.12680686 0.06711966 0.05883393]
```

```
readableNumberOfRows = 5
print("PCs as numpy array")
print(EmpleadosAttritionPCA[0:readableNumberOfRows])
pcaDf = pd.DataFrame(data=EmpleadosAttritionPCA, columns=[f'PC{i+1}'
for i in range(EmpleadosAttritionPCA.shape[1])])
print("\nPCs as dataframe")
print(pcaDf.head(readableNumberOfRows))
```

```
PCs as numpy array
[[-0.41865799 -0.02533991 -0.13936604  0.08236991 -0.09256093 -
0.00911893]
 [-0.41865799 -0.02533991 -0.13936604  0.08236991 -0.09256093 -
0.00911893]
 [ 0.73238095  0.77027152  0.13209651  0.84681449  0.26097396 -
0.51809579]
 [ 0.12988815  0.75500952 -0.12913339 -0.20245454 -0.0671046
0.0818437 ]
 [ 0.74870823 -0.71618348 -0.0775299   0.41342723 -0.30708013 -
0.03257364]]

PCs as dataframe
        PC1       PC2       PC3       PC4       PC5       PC6
0 -0.418658 -0.025340 -0.139366  0.082370 -0.092561 -0.009119
1 -0.418658 -0.025340 -0.139366  0.082370 -0.092561 -0.009119
2  0.732381  0.770272  0.132097  0.846814  0.260974 -0.518096
3  0.129888  0.755010 -0.129133 -0.202455 -0.067105  0.081844
4  0.748708 -0.716183 -0.077530  0.413427 -0.307080 -0.032574
```

```
acum = 0
for i in range(len(pca.explained_variance_ratio_)):
  acum += pca.explained_variance_ratio_[i]
```

```
  print(f"Variance acummulated until PC{i+1}: {acum}")
  if acum > 0.8:
    break
```

```
Variance acummulated until PC1: 0.3123714003297026
Variance acummulated until PC2: 0.5600585379215175
Variance acummulated until PC3: 0.7472395535216889
Variance acummulated until PC4: 0.8740464130115218
```

```
enoughVariancePCs = pcaDf.iloc[:, :4]
EmpleadosAttritionFinal = pd.concat([EmpleadosAttritionFinal,
enoughVariancePCs], axis=1)
print(EmpleadosAttritionFinal.columns)
```

```
Index(['EducationField_Technical Degree', 'JobRole_Laboratory
Technician',
       'JobRole_Sales Representative', 'MaritalStatus_Single',
       'OverTimeEncoded', 'AttritionEncoded', 'PC1', 'PC2', 'PC3',
'PC4'],
      dtype='object')
```

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

```
from google.colab import files
files.download('EmpleadosAttritionFinal.csv')
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```