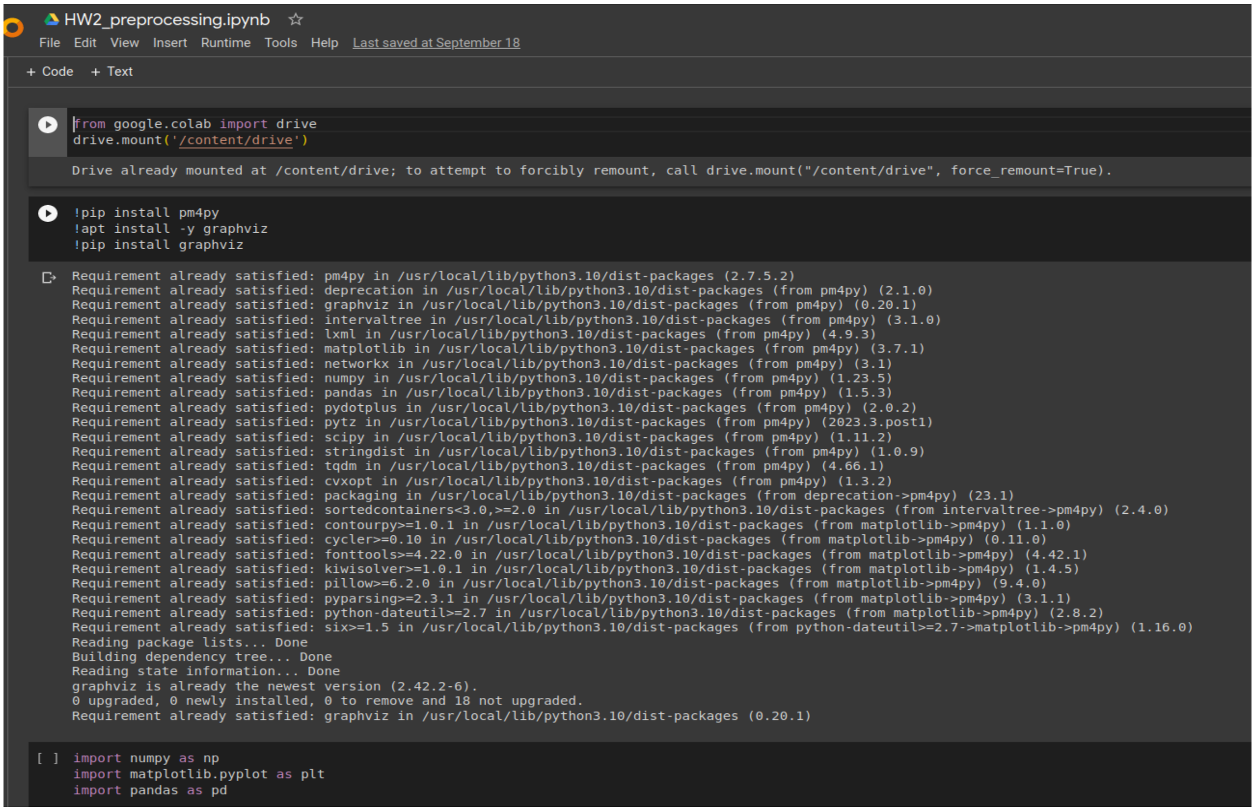
Juan Uriarte
CPSC 483-01 13657

HW2_Preprocessing

Importing the file from the drive
Installing the required tools and libraries to run the program


HW2_preprocessing.ipynb ☆
File Edit View Insert Runtime Tools Help Last saved at September 18
+ Code + Text

```
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).
```

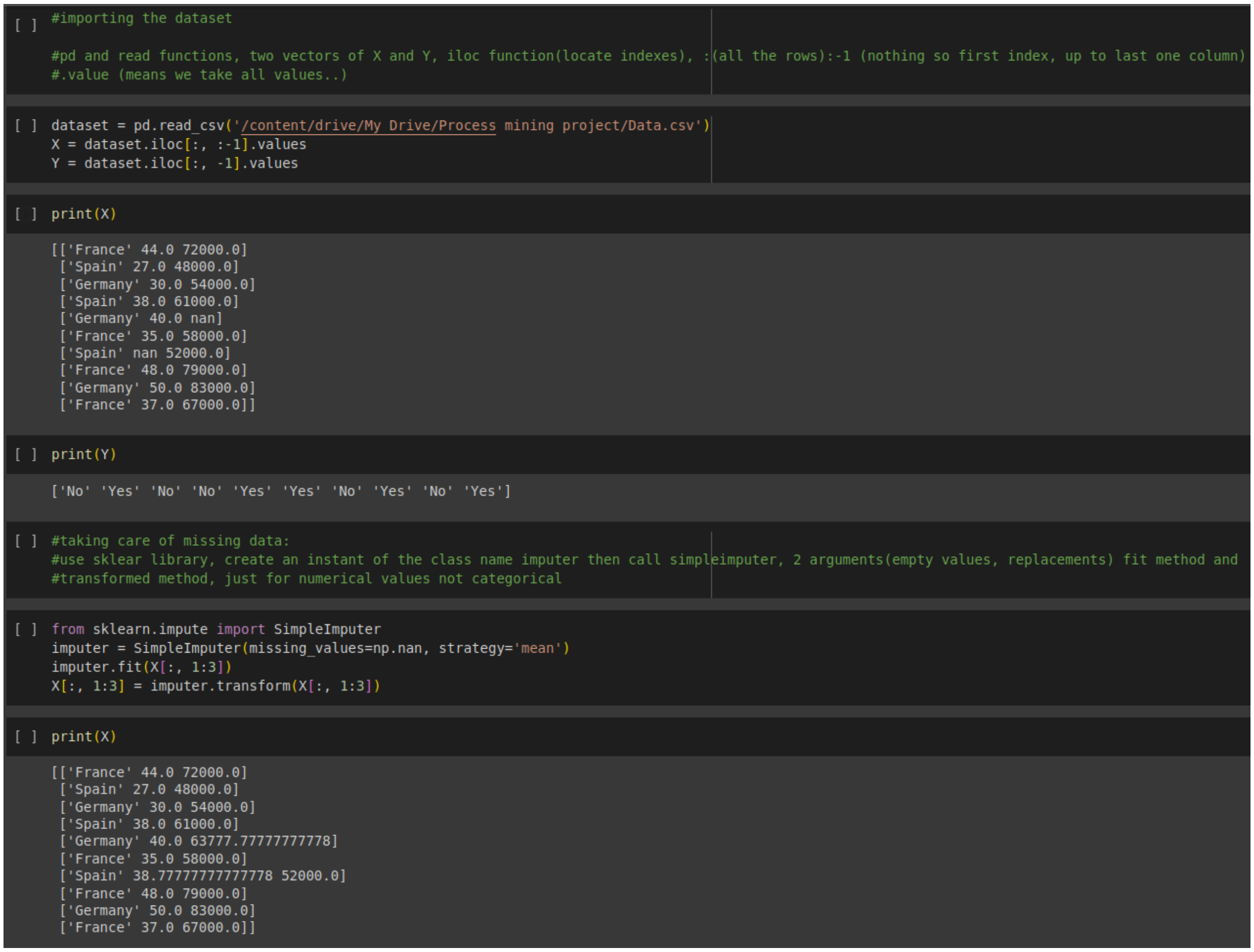
```
!pip install pm4py
!apt install -y graphviz
!pip install graphviz
```

```
Requirement already satisfied: pm4py in /usr/local/lib/python3.10/dist-packages (2.7.5.2)
Requirement already satisfied: deprecation in /usr/local/lib/python3.10/dist-packages (from pm4py) (2.1.0)
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-packages (from pm4py) (0.20.1)
Requirement already satisfied: intervaltree in /usr/local/lib/python3.10/dist-packages (from pm4py) (3.1.0)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages (from pm4py) (4.9.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from pm4py) (3.7.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from pm4py) (3.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from pm4py) (1.23.5)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from pm4py) (1.5.3)
Requirement already satisfied: pydotplus in /usr/local/lib/python3.10/dist-packages (from pm4py) (2.0.2)
Requirement already satisfied: pytz in /usr/local/lib/python3.10/dist-packages (from pm4py) (2023.3.post1)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from pm4py) (1.11.2)
Requirement already satisfied: stringdist in /usr/local/lib/python3.10/dist-packages (from pm4py) (1.0.9)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from pm4py) (4.66.1)
Requirement already satisfied: cvxopt in /usr/local/lib/python3.10/dist-packages (from pm4py) (1.3.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from deprecation->pm4py) (23.1)
Requirement already satisfied: sortedcontainers<3.0,>=2.0 in /usr/local/lib/python3.10/dist-packages (from intervaltree->pm4py) (2.4.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pm4py) (1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pm4py) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pm4py) (4.42.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pm4py) (1.4.5)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pm4py) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pm4py) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pm4py) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib->pm4py) (1.16.0)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
graphviz is already the newest version (2.42.2-6).
0 upgraded, 0 newly installed, 0 to remove and 18 not upgraded.
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-packages (0.20.1)
```

```
[ ] import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd
```

Juan Uriarte
CPSC 483-01 13657

Importing the dataset and reading the two vectors
Take care of missing data



```
[ ] #importing the dataset

    #pd and read functions, two vectors of X and Y, iloc function(locate indexes), :(all the rows):-1 (nothing so first index, up to last one column)
    #.value (means we take all values..)
```

```
[ ] dataset = pd.read_csv('/content/drive/My Drive/Process mining project/Data.csv')
    X = dataset.iloc[:, :-1].values
    Y = dataset.iloc[:, -1].values
```

```
[ ] print(X)

    [['France' 44.0 72000.0]
     ['Spain' 27.0 48000.0]
     ['Germany' 30.0 54000.0]
     ['Spain' 38.0 61000.0]
     ['Germany' 40.0 nan]
     ['France' 35.0 58000.0]
     ['Spain' nan 52000.0]
     ['France' 48.0 79000.0]
     ['Germany' 50.0 83000.0]
     ['France' 37.0 67000.0]]
```

```
[ ] print(Y)

    ['No' 'Yes' 'No' 'No' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes']
```

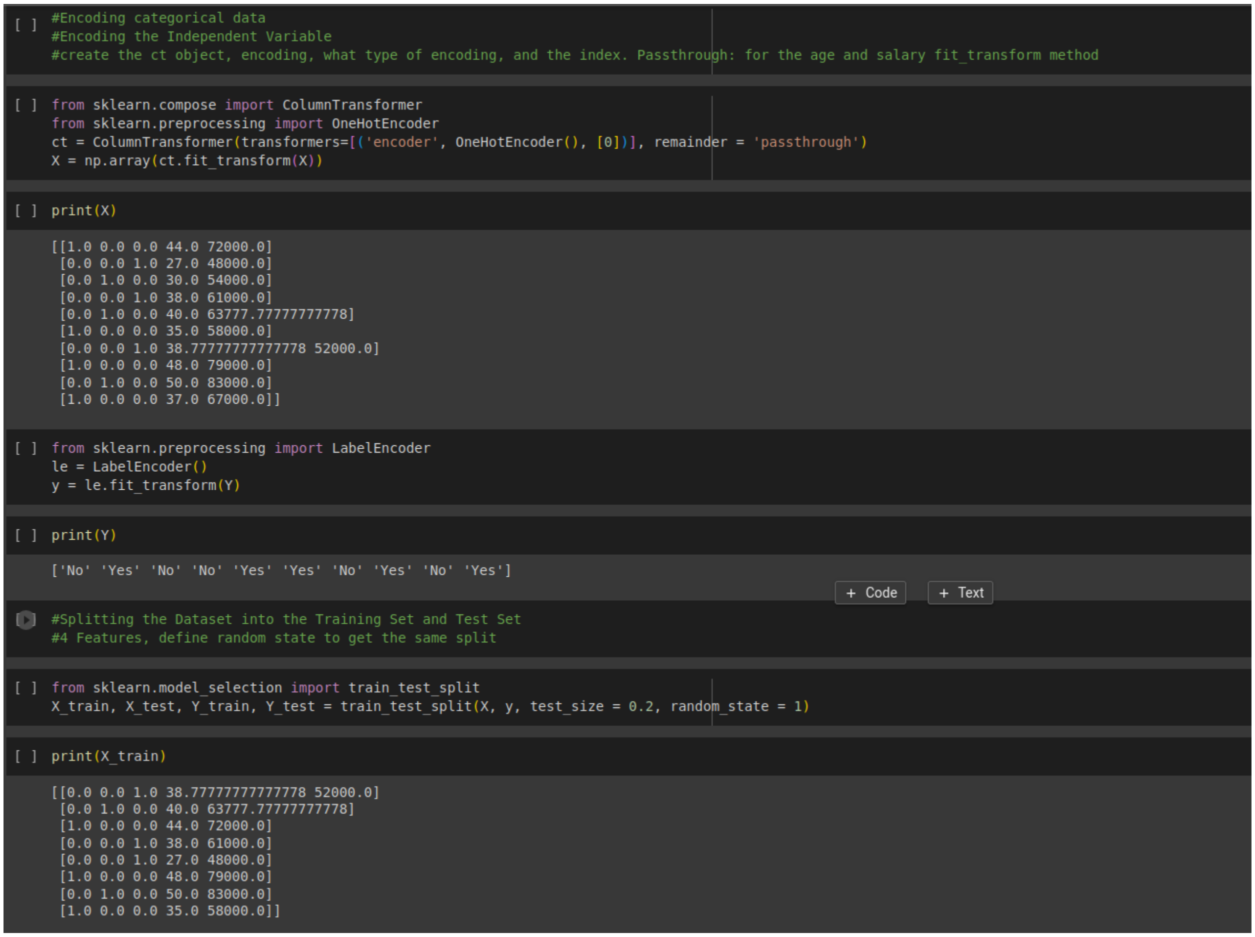
```
[ ] #taking care of missing data:
    #use sklear library, create an instant of the class name imputer then call simpleimputer, 2 arguments(empty values, replacements) fit method and
    #transformed method, just for numerical values not categorical
```

```
[ ] from sklearn.impute import SimpleImputer
    imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
    imputer.fit(X[:, 1:3])
    X[:, 1:3] = imputer.transform(X[:, 1:3])
```

```
[ ] print(X)

    [['France' 44.0 72000.0]
     ['Spain' 27.0 48000.0]
     ['Germany' 30.0 54000.0]
     ['Spain' 38.0 61000.0]
     ['Germany' 40.0 63777.77777777778]
     ['France' 35.0 58000.0]
     ['Spain' 38.77777777777778 52000.0]
     ['France' 48.0 79000.0]
     ['Germany' 50.0 83000.0]
     ['France' 37.0 67000.0]]
```

Encoding the categorical variable and independent variable
Create ct object



```
#Encoding categorical data
#Encoding the Independent Variable
#create the ct object, encoding, what type of encoding, and the index. Passthrough: for the age and salary fit_transform method
```

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder = 'passthrough')
X = np.array(ct.fit_transform(X))
```

```
print(X)
```

```
[[1.0 0.0 0.0 44.0 72000.0]
 [0.0 0.0 1.0 27.0 48000.0]
 [0.0 1.0 0.0 30.0 54000.0]
 [0.0 0.0 1.0 38.0 61000.0]
 [0.0 1.0 0.0 40.0 63777.77777777778]
 [1.0 0.0 0.0 35.0 58000.0]
 [0.0 0.0 1.0 38.77777777777778 52000.0]
 [1.0 0.0 0.0 48.0 79000.0]
 [0.0 1.0 0.0 50.0 83000.0]
 [1.0 0.0 0.0 37.0 67000.0]]
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(Y)
```

```
print(Y)
```

```
['No' 'Yes' 'No' 'No' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes']
```

```
#Splitting the Dataset into the Training Set and Test Set
#4 Features, define random state to get the same split
```

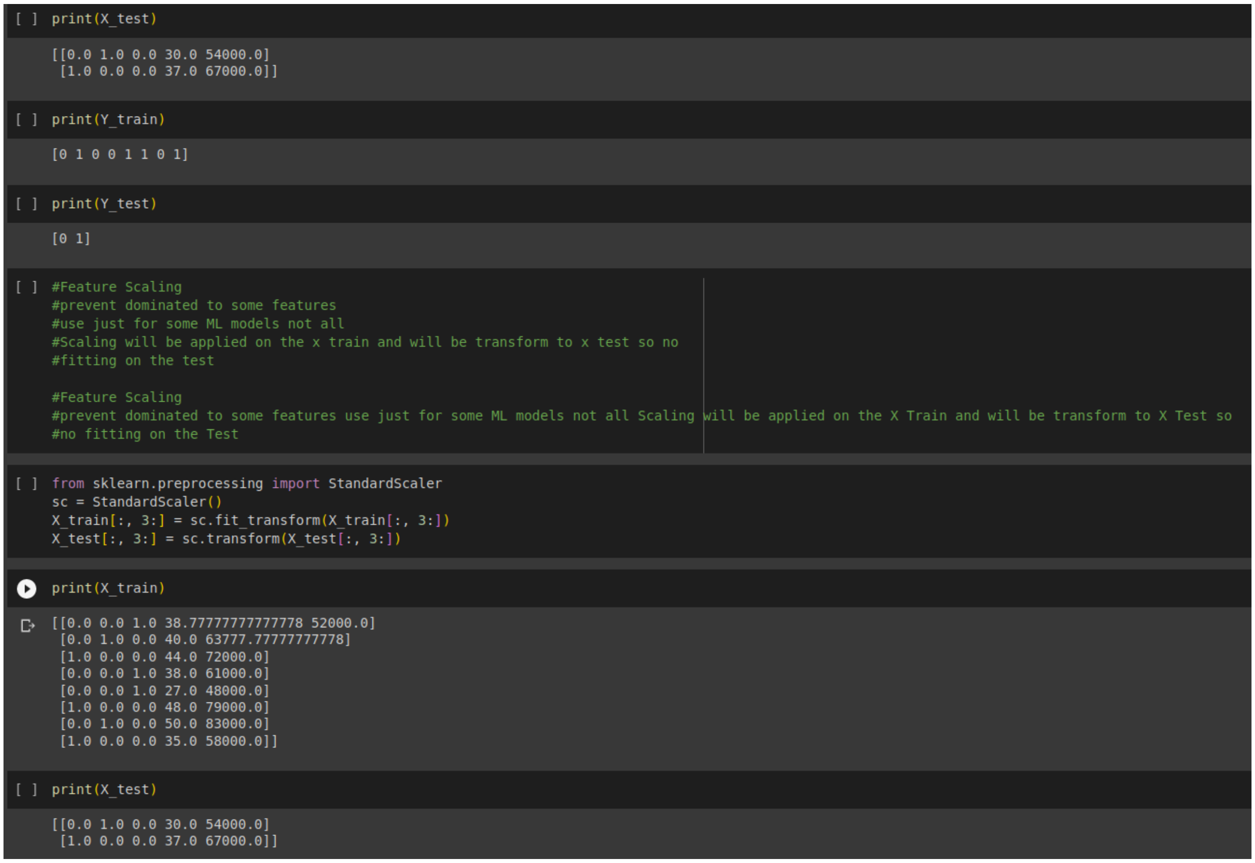
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size = 0.2, random_state = 1)
```

```
print(X_train)
```

```
[[0.0 0.0 1.0 38.77777777777778 52000.0]
 [0.0 1.0 0.0 40.0 63777.77777777778]
 [1.0 0.0 0.0 44.0 72000.0]
 [0.0 0.0 1.0 38.0 61000.0]
 [0.0 0.0 1.0 27.0 48000.0]
 [1.0 0.0 0.0 48.0 79000.0]
 [0.0 1.0 0.0 50.0 83000.0]
 [1.0 0.0 0.0 35.0 58000.0]]
```

Juan Uriarte
CPSC 483-01 13657

Implemented Scaling on X train but not all some will be used for X Test
Print X_Train and X_Test



```
print(X_test)

[[0.0 1.0 0.0 30.0 54000.0]
 [1.0 0.0 0.0 37.0 67000.0]]
```

```
print(Y_train)

[0 1 0 0 1 1 0 1]
```

```
print(Y_test)

[0 1]
```

```
#Feature Scaling
#prevent dominated to some features
#use just for some ML models not all
#Scaling will be applied on the x train and will be transform to x test so no
#fitting on the test

#Feature Scaling
#prevent dominated to some features use just for some ML models not all Scaling will be applied on the X Train and will be transform to X Test so
#no fitting on the Test
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train[:, 3:] = sc.fit_transform(X_train[:, 3:])
X_test[:, 3:] = sc.transform(X_test[:, 3:])
```

```
print(X_train)

[[0.0 0.0 1.0 38.77777777777778 52000.0]
 [0.0 1.0 0.0 40.0 63777.77777777778]
 [1.0 0.0 0.0 44.0 72000.0]
 [0.0 0.0 1.0 38.0 61000.0]
 [0.0 0.0 1.0 27.0 48000.0]
 [1.0 0.0 0.0 48.0 79000.0]
 [0.0 1.0 0.0 50.0 83000.0]
 [1.0 0.0 0.0 35.0 58000.0]]
```

```
print(X_test)

[[0.0 1.0 0.0 30.0 54000.0]
 [1.0 0.0 0.0 37.0 67000.0]]
```