Data Preprocessing

- Dealing with duplicate values
- Deaking with missing values
- Dealing With categorical values
- Standardization
- Train Test Split

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

df = pd.read_csv('Data.csv')
df
```

```
    Country   Age   Salary Purchased
0    France  44.0  72000.0        No
1     Spain  27.0  48000.0       Yes
2   Germany  30.0  54000.0        No
3     Spain  38.0  61000.0        No
4   Germany  40.0      NaN       Yes
5    France  35.0  58000.0       Yes
6     Spain   NaN  52000.0        No
7    France  48.0  79000.0       Yes
8   Germany  50.0  83000.0        No
9    France  37.0  67000.0       Yes
10   France  37.0  67000.0       Yes
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11 entries, 0 to 10
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Country    11 non-null     object
 1   Age        10 non-null     float64
 2   Salary     10 non-null     float64
 3   Purchased  11 non-null     object
dtypes: float64(2), object(2)
memory usage: 480.0+ bytes
```

```
df.nunique()
```

```
Country      3
Age          9
Salary       9
Purchased    2
dtype: int64
```

```
df['Country'].unique()
```

```
array(['France', 'Spain', 'Germany'], dtype=object)
```

```
df['Purchased'].unique()
```

```
array(['No', 'Yes'], dtype=object)
```

Check if duplicate entries are present

```
df.duplicated()
```

```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10     True
dtype: bool
```

```
df.duplicated().sum()
```

```
1
```

```
# drop duplicate rows
df.drop_duplicates(inplace = True)
df
```

	Country	Age	Salary	Purchased
0	France	44.0	72000.0	No
1	Spain	27.0	48000.0	Yes
2	Germany	30.0	54000.0	No
3	Spain	38.0	61000.0	No
4	Germany	40.0	NaN	Yes
5	France	35.0	58000.0	Yes
6	Spain	NaN	52000.0	No
7	France	48.0	79000.0	Yes
8	Germany	50.0	83000.0	No
9	France	37.0	67000.0	Yes

Check and deal with null values

```
df.isnull().sum()
```

```
Country      0
Age          1
Salary       1
Purchased    0
dtype: int64
```

```
avg_age = df['Age'].mean()
avg_salary = df['Salary'].mean()
print(avg_age)
print(avg_salary)
```

```
38.77777777777778
63777.77777777778
```

```
df['Age'].replace(np.nan, avg_age, inplace = True)
df['Salary'].replace(np.nan, avg_salary, inplace = True)
df
```

```
   Country        Age        Salary Purchased
0   France  44.000000  72000.000000        No
1    Spain  27.000000  48000.000000       Yes
2  Germany  30.000000  54000.000000        No
3    Spain  38.000000  61000.000000        No
4  Germany  40.000000  63777.777778       Yes
5   France  35.000000  58000.000000       Yes
6    Spain  38.777778  52000.000000        No
7   France  48.000000  79000.000000       Yes
8  Germany  50.000000  83000.000000        No
9   France  37.000000  67000.000000       Yes
```

Using Scikit Learn

```
df2 = pd.read_csv('Data.csv')
df2.drop_duplicates(inplace = True)
```

```
X = df2.iloc[:,:-1].values
Y = df2.iloc[:, -1].values
```

```
X
```

```
array([['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]], dtype=object)
```

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

```
array([['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]], dtype=object)
```

Dealing with categorical values

```
dummy1 = pd.get_dummies(df['Country'])
dummy1
```

	France	Germany	Spain
0	1	0	0
1	0	0	1
2	0	1	0
3	0	0	1
4	0	1	0
5	1	0	0
6	0	0	1
7	1	0	0
8	0	1	0
9	1	0	0

```
df = pd.concat([dummy1,df], axis = 1)
df
```

	France	Germany	Spain	Country	Age	Salary	Purchased
0	1	0	0	France	44.000000	72000.000000	No
1	0	0	1	Spain	27.000000	48000.000000	Yes
2	0	1	0	Germany	30.000000	54000.000000	No
3	0	0	1	Spain	38.000000	61000.000000	No
4	0	1	0	Germany	40.000000	63777.777778	Yes
5	1	0	0	France	35.000000	58000.000000	Yes
6	0	0	1	Spain	38.777778	52000.000000	No
7	1	0	0	France	48.000000	79000.000000	Yes
8	0	1	0	Germany	50.000000	83000.000000	No
9	1	0	0	France	37.000000	67000.000000	Yes

Standardization

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X[:, 1:] = sc.fit_transform(X[:, 1:])
X
```

```
array([['France', 0.758874361590019, 0.7494732544921677],
       ['Spain', -1.7115038793306814, -1.4381784072687531],
       ['Germany', -1.2755547779917342, -0.8912654918285229],
       ['Spain', -0.1130238410878753, -0.253200423814921],
       ['Germany', 0.17760889313808945, 6.632191985654332e-16],
```

```
       ['France', -0.5489729424268225, -0.5266568815350361],
       ['Spain', 0.0, -1.0735697969752662],
       ['France', 1.3401398300419485, 1.3875383225057696],
       ['Germany', 1.6307725642679132, 1.7521469327992565],
       ['France', -0.2583402082008577, 0.29371249162530916]],
dtype=object)
```

Standardized columns have unit variance

```
X[:,1].var()
```

```
1.0
```

```
X[:,2].var()
```

```
1.0000000000000002
```

```
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))
X
```

```
array([[1.0, 0.0, 0.0, 0.758874361590019, 0.7494732544921677],
       [0.0, 0.0, 1.0, -1.7115038793306814, -1.4381784072687531],
       [0.0, 1.0, 0.0, -1.2755547779917342, -0.8912654918285229],
       [0.0, 0.0, 1.0, -0.1130238410878753, -0.253200423814921],
       [0.0, 1.0, 0.0, 0.17760889313808945, 6.632191985654332e-16],
       [1.0, 0.0, 0.0, -0.5489729424268225, -0.5266568815350361],
       [0.0, 0.0, 1.0, 0.0, -1.0735697969752662],
       [1.0, 0.0, 0.0, 1.3401398300419485, 1.3875383225057696],
       [0.0, 1.0, 0.0, 1.6307725642679132, 1.7521469327992565],
       [1.0, 0.0, 0.0, -0.2583402082008577, 0.29371249162530916]],
      dtype=object)
```

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

```
array([0, 1, 0, 0, 1, 1, 0, 1, 0, 1])
```

```
z = df['Country'].values
z = le.fit_transform(z)
z
```

```
array([0, 2, 1, 2, 1, 0, 2, 0, 1, 0])
```

```
X
```

```
array([[1.0, 0.0, 0.0, 0.758874361590019, 0.7494732544921677],
       [0.0, 0.0, 1.0, -1.7115038793306814, -1.4381784072687531],
       [0.0, 1.0, 0.0, -1.2755547779917342, -0.8912654918285229],
```

```
       [0.0, 0.0, 1.0, -0.1130238410878753, -0.253200423814921],
       [0.0, 1.0, 0.0, 0.17760889313808945, 6.632191985654332e-16],
       [1.0, 0.0, 0.0, -0.5489729424268225, -0.5266568815350361],
       [0.0, 0.0, 1.0, 0.0, -1.0735697969752662],
       [1.0, 0.0, 0.0, 1.3401398300419485, 1.3875383225057696],
       [0.0, 1.0, 0.0, 1.6307725642679132, 1.7521469327992565],
       [1.0, 0.0, 0.0, -0.2583402082008577, 0.29371249162530916]],
      dtype=object)
```

```
Y
```

```
array([0, 1, 0, 0, 1, 1, 0, 1, 0, 1])
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size =
0.3)
```

```
X_train
```

```
array([[0.0, 0.0, 1.0, -1.7115038793306814, -1.4381784072687531],
       [0.0, 1.0, 0.0, -1.2755547779917342, -0.8912654918285229],
       [0.0, 1.0, 0.0, 1.6307725642679132, 1.7521469327992565],
       [0.0, 0.0, 1.0, 0.0, -1.0735697969752662],
       [1.0, 0.0, 0.0, 0.758874361590019, 0.7494732544921677],
       [0.0, 1.0, 0.0, 0.17760889313808945, 6.632191985654332e-16],
       [1.0, 0.0, 0.0, 1.3401398300419485, 1.3875383225057696]],
      dtype=object)
```

```
Y_train
```

```
array([1, 0, 0, 0, 0, 1, 1])
```

```
X_test
```

```
array([[0.0, 0.0, 1.0, -0.1130238410878753, -0.253200423814921],
       [1.0, 0.0, 0.0, -0.2583402082008577, 0.29371249162530916],
       [1.0, 0.0, 0.0, -0.5489729424268225, -0.5266568815350361]],
      dtype=object)
```

```
Y_test
```

```
array([0, 1, 1])
```

Doubts

```
df3 = pd.read_csv('Data.csv')
df3
```

```
    Country   Age   Salary Purchased
0    France  44.0  72000.0        No
1     Spain  27.0  48000.0       Yes
2   Germany  30.0  54000.0        No
```

```
3     Spain  38.0  61000.0        No
4   Germany  40.0      NaN       Yes
5    France  35.0  58000.0       Yes
6     Spain   NaN  52000.0        No
7    France  48.0  79000.0       Yes
8   Germany  50.0  83000.0        No
9    France  37.0  67000.0       Yes
10   France  37.0  67000.0       Yes
```

```
df3[df3['Salary'] > 60000][df3['Country'] == 'France']
```

```
<ipython-input-64-3be6d7d9671d>:1: UserWarning: Boolean Series key
will be reindexed to match DataFrame index.
  df3[df3['Salary'] > 60000][df3['Country'] == 'France']
```

```
   Country   Age   Salary Purchased
0   France  44.0  72000.0        No
7   France  48.0  79000.0       Yes
9   France  37.0  67000.0       Yes
10  France  37.0  67000.0       Yes
```

```
df3.groupby('Country').mean()['Salary']
```

```
Country
France     68600.000000
Germany    68500.000000
Spain      53666.666667
Name: Salary, dtype: float64
```