

```
In [2]: #assignment-03  
#D.prudhvi sai
```

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

```
In [3]: dataset = pd.read_csv('Churn_Modelling1.csv')
```

```
In [4]: type(dataset)
```

```
Out[4]: pandas.core.frame.DataFrame
```

```
In [39]: dataset.isnull().any()  
dataset.isnull().sum()
```

```
Out[39]: RowNumber      0  
CustomerId      0  
Surname      0  
CreditScore      0  
Geography      0  
Gender      0  
Age      0  
Tenure      0  
Balance      0  
NumOfProducts      0  
HasCrCard      0  
IsActiveMember      0  
EstimatedSalary      0  
Exited      0  
dtype: int64
```

```
In [5]: dataset
```

...

```
In [6]: dataset['Age'].fillna(dataset['Age'].mean())
```

```
Out[6]: 0      42.000000  
1      41.000000  
2      42.000000  
3      39.000000  
4      38.918768  
...  
9995    39.000000  
9996    35.000000  
9997    36.000000  
9998    42.000000  
9999    28.000000  
Name: Age, Length: 10000, dtype: float64
```

```
In [7]: dataset.isnull().any()
```

```
Out[7]: RowNumber      False
        CustomerId     False
        Surname         False
        CreditScore     False
        Geography       True
        Gender           True
        Age              True
        Tenure           False
        Balance          True
        NumOfProducts   False
        HasCrCard        False
        IsActiveMember   False
        EstimatedSalary  False
        Exited           False
        dtype: bool
```

```
In [8]: dataset[dataset['Age'].isnull()].index.tolist()
```

```
Out[8]: [4, 28, 43, 59]
```

```
In [9]: dataset['Age'].fillna(dataset['Age'].mean(),inplace=True)
```

```
In [10]: dataset[dataset['Gender'].isnull()].index.tolist()
```

```
Out[10]: [6, 21, 32]
```

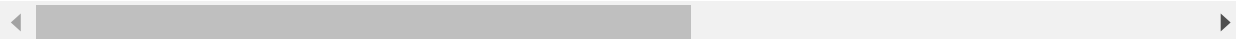
```
In [11]: dataset['Gender'].fillna(dataset['Gender'].mode(),inplace=True)
```

In [12]: dataset

Out[12]:

| | RowNumber | CustomerId | Surname | CreditScore | Geography | Gender | Age | Tenure | |
|------|-----------|------------|-----------|-------------|-----------|--------|-----------|--------|-----|
| 0 | 1 | 15634602 | Hargrave | 619 | France | Female | 42.000000 | 2 | |
| 1 | 2 | 15647311 | Hill | 608 | Spain | Female | 41.000000 | 1 | 8 |
| 2 | 3 | 15619304 | Onio | 502 | France | Female | 42.000000 | 8 | 15 |
| 3 | 4 | 15701354 | Boni | 699 | France | Female | 39.000000 | 1 | |
| 4 | 5 | 15737888 | Mitchell | 850 | Spain | Female | 38.918768 | 2 | 12 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 9995 | 9996 | 15606229 | Obijiaku | 771 | France | Male | 39.000000 | 5 | |
| 9996 | 9997 | 15569892 | Johnstone | 516 | France | Male | 35.000000 | 10 | 5 |
| 9997 | 9998 | 15584532 | Liu | 709 | France | Female | 36.000000 | 7 | |
| 9998 | 9999 | 15682355 | Sabbatini | 772 | Germany | Male | 42.000000 | 3 | 7 |
| 9999 | 10000 | 15628319 | Walker | 792 | France | Female | 28.000000 | 4 | 13 |

10000 rows × 14 columns



In [13]: dataset[dataset['Geography'].isnull()].index.tolist()

Out[13]: [16, 30, 41]

In [15]: dataset['Geography'].fillna(dataset['Geography'].mode(),inplace=True)

In [16]: dataset[dataset['Balance'].isnull()].index.tolist()

Out[16]: [10, 26]

In [18]: dataset['Balance'].fillna(dataset['Balance'].mean(),inplace=True)

```
In [19]: dataset.isnull().any()
```

```
Out[19]: RowNumber      False
CustomerId    False
Surname        False
CreditScore    False
Geography      True
Gender         True
Age            False
Tenure         False
Balance        False
NumOfProducts False
HasCrCard      False
IsActiveMember False
EstimatedSalary False
Exited         False
dtype: bool
```

```
In [20]: dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column                Non-Null Count  Dtype  
---  -
 0   RowNumber              10000 non-null  int64  
 1   CustomerId             10000 non-null  int64  
 2   Surname                 10000 non-null  object  
 3   CreditScore             10000 non-null  int64  
 4   Geography               9997 non-null   object  
 5   Gender                  9997 non-null   object  
 6   Age                    10000 non-null  float64  
 7   Tenure                  10000 non-null  int64  
 8   Balance                 10000 non-null  float64  
 9   NumOfProducts          10000 non-null  int64  
10   HasCrCard               10000 non-null  int64  
11   IsActiveMember          10000 non-null  int64  
12   EstimatedSalary         10000 non-null  float64  
13   Exited                  10000 non-null  int64  
dtypes: float64(3), int64(8), object(3)
memory usage: 1.1+ MB
```

```
In [21]: dataset['Gender'] = dataset['Gender'].fillna(dataset['Gender'].mode()[0])
```

```
In [22]: dataset['Geography'] = dataset['Geography'].fillna(dataset['Geography'].mode()[0])
```

```
In [23]: dataset.isnull().any()
```

```
Out[23]: RowNumber      False
CustomerId      False
Surname         False
CreditScore     False
Geography       False
Gender          False
Age            False
Tenure         False
Balance        False
NumOfProducts  False
HasCrCard       False
IsActiveMember  False
EstimatedSalary False
Exited         False
dtype: bool
```

```
In [31]: from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
In [32]: dataset['Geography']=le.fit_transform(dataset['Geography'])
dataset['Gender']=le.fit_transform(dataset['Gender'])
dataset['Surname']=le.fit_transform(dataset['Surname'])
```

```
In [33]: dataset
```

```
Out[33]:
```

| | RowNumber | CustomerId | Surname | CreditScore | Geography | Gender | Age | Tenure | E |
|------|-----------|------------|---------|-------------|-----------|--------|-----------|--------|-----|
| 0 | 1 | 15634602 | 1115 | 619 | 0 | 0 | 42.000000 | 2 | |
| 1 | 2 | 15647311 | 1177 | 608 | 2 | 0 | 41.000000 | 1 | 8: |
| 2 | 3 | 15619304 | 2040 | 502 | 0 | 0 | 42.000000 | 8 | 15: |
| 3 | 4 | 15701354 | 289 | 699 | 0 | 0 | 39.000000 | 1 | |
| 4 | 5 | 15737888 | 1822 | 850 | 2 | 0 | 38.918768 | 2 | 12: |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 9995 | 9996 | 15606229 | 1999 | 771 | 0 | 1 | 39.000000 | 5 | |
| 9996 | 9997 | 15569892 | 1336 | 516 | 0 | 1 | 35.000000 | 10 | 5: |
| 9997 | 9998 | 15584532 | 1570 | 709 | 0 | 0 | 36.000000 | 7 | |
| 9998 | 9999 | 15682355 | 2345 | 772 | 1 | 1 | 42.000000 | 3 | 7: |
| 9999 | 10000 | 15628319 | 2751 | 792 | 0 | 0 | 28.000000 | 4 | 13: |

10000 rows × 14 columns



In [34]: dataset

Out[34]:

| | RowNumber | CustomerId | Surname | CreditScore | Geography | Gender | Age | Tenure |
|------|-----------|------------|---------|-------------|-----------|--------|-----------|--------|
| 0 | 1 | 15634602 | 1115 | 619 | 0 | 0 | 42.000000 | 2 |
| 1 | 2 | 15647311 | 1177 | 608 | 2 | 0 | 41.000000 | 1 |
| 2 | 3 | 15619304 | 2040 | 502 | 0 | 0 | 42.000000 | 8 |
| 3 | 4 | 15701354 | 289 | 699 | 0 | 0 | 39.000000 | 1 |
| 4 | 5 | 15737888 | 1822 | 850 | 2 | 0 | 38.918768 | 2 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 9995 | 9996 | 15606229 | 1999 | 771 | 0 | 1 | 39.000000 | 5 |
| 9996 | 9997 | 15569892 | 1336 | 516 | 0 | 1 | 35.000000 | 10 |
| 9997 | 9998 | 15584532 | 1570 | 709 | 0 | 0 | 36.000000 | 7 |
| 9998 | 9999 | 15682355 | 2345 | 772 | 1 | 1 | 42.000000 | 3 |
| 9999 | 10000 | 15628319 | 2751 | 792 | 0 | 0 | 28.000000 | 4 |

10000 rows × 14 columns

In [35]: x=dataset.iloc[:,0:12].values
x

Out[35]: array([[1.0000000e+00, 1.5634602e+07, 1.1150000e+03, ..., 1.0000000e+00,
1.0000000e+00, 1.0000000e+00],
[2.0000000e+00, 1.5647311e+07, 1.1770000e+03, ..., 1.0000000e+00,
0.0000000e+00, 1.0000000e+00],
[3.0000000e+00, 1.5619304e+07, 2.0400000e+03, ..., 3.0000000e+00,
1.0000000e+00, 0.0000000e+00],
...,
[9.9980000e+03, 1.5584532e+07, 1.5700000e+03, ..., 1.0000000e+00,
0.0000000e+00, 1.0000000e+00],
[9.9990000e+03, 1.5682355e+07, 2.3450000e+03, ..., 2.0000000e+00,
1.0000000e+00, 0.0000000e+00],
[1.0000000e+04, 1.5628319e+07, 2.7510000e+03, ..., 1.0000000e+00,
1.0000000e+00, 0.0000000e+00]])

In [37]: y=dataset.iloc[:,13:].values
y

Out[37]: array([[1],
[0],
[1],
...,
[1],
[1],
[0]], dtype=int64)

```
In [38]: from sklearn.preprocessing import OneHotEncoder  
         oh=OneHotEncoder()
```

```
In [40]: z=oh.fit_transform(x[:,0:1]).toarray()
```

```
In [41]: z
```

```
Out[41]: array([[1., 0., 0., ..., 0., 0., 0.],  
                [0., 1., 0., ..., 0., 0., 0.],  
                [0., 0., 1., ..., 0., 0., 0.],  
                ...,  
                [0., 0., 0., ..., 1., 0., 0.],  
                [0., 0., 0., ..., 0., 1., 0.],  
                [0., 0., 0., ..., 0., 0., 1.]])
```

```
In [43]: x=np.concatenate((x,z),axis=1)  
         y=np.concatenate((y,z),axis=1)
```

```
In [44]: from sklearn.model_selection import train_test_split
```

```
In [45]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

```
In [46]: x_train.shape
```

```
Out[46]: (7000, 20012)
```

```
In [47]: x_test.shape
```

```
Out[47]: (3000, 20012)
```

```
In [48]: y_train.shape
```

```
Out[48]: (7000, 10001)
```

```
In [49]: y_test.shape
```

```
Out[49]: (3000, 10001)
```

```
In [ ]:
```