DATA SCIENCE

Lab-5

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
DS = pd.read_csv("Titanic_Dataset.csv")
DS.head()
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	Na
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C8
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	Na
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C12
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	Na

In [3]:
```
DS.shape
```

Out[3]:
```
(891, 12)
```

In [4]:

```
DS.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [5]:

```
DS.describe()
```

Out[5]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [29]:

```
# Identification of Categorical Variables
categorical_variables = [feature for feature in DS.columns if DS[feature].dtypes
print('Number of Categorical Variables: ', len(categorical_variables))
DS[categorical_variables].head(2)
```

```
Number of Categorical Variables:  5
```

Out[29]:

	Name	Sex	Ticket	Cabin	Embarked
0	Braund, Mr. Owen Harris	male	A/5 21171	NaN	S
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	PC 17599	C85	C

In [7]:

```
# Drop columns
DS[categorical_variables].drop(['Name','Ticket','Cabin'],axis=1).head()
```

Out[7]:

	Sex	Embarked
0	male	S
1	female	C
2	female	S
3	female	S
4	male	S

In [8]:

```
# Storing a columns of a dataset in new variable for performing Feature Encoding
New_DS=pd.read_csv("Titanic_Dataset.csv",usecols=['Sex','Embarked'])
New_DS.head()
```

Out[8]:

	Sex	Embarked
0	male	S
1	female	C
2	female	S
3	female	S
4	male	S

In [9]:

```
New_DS.shape
```

Out[9]:

```
(891, 2)
```

In [10]:

```
New_DS.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Sex       891 non-null    object
 1   Embarked  889 non-null    object
dtypes: object(2)
memory usage: 14.0+ KB
```

In [11]:

```
New_DS.count()
```

Out[11]:

```
Sex         891
Embarked    889
dtype: int64
```

In [12]:

```
# Finding number of unique values in columns
for i in New_DS.columns:
    print(i, ':', len(New_DS[i].unique()),'labels')
```

```
Sex : 2 labels
Embarked : 4 labels
```

In [13]:

```
# Creating dummie values of dataset
pd.get_dummies(New_DS,drop_first=False).shape
```

Out[13]:

```
(891, 5)
```

In [14]:

```
# Printing dummie values
pd.get_dummies(New_DS,drop_first=False).head(10)
```

Out[14]:

	Sex_female	Sex_male	Embarked_C	Embarked_Q	Embarked_S
0	0	1	0	0	1
1	1	0	1	0	0
2	1	0	0	0	1
3	1	0	0	0	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	0	0	1
7	0	1	0	0	1
8	1	0	0	0	1
9	1	0	1	0	0

Feature Encoding on column Sex

In [15]:

```
# Counting number of values for unique values in Sex columns
New_DS.Sex.value_counts().sort_values(ascending=False).head()
```

Out[15]:

```
male      577
female    314
Name: Sex, dtype: int64
```

In [16]:

```
# Creating dummie value for column Sex
dummies_1=pd.get_dummies(New_DS['Sex'])
dummies_1.head()
```

Out[16]:

	female	male
0	0	1
1	1	0
2	1	0
3	1	0
4	0	1

In [17]:

```
# Selecting top 2 unique values
top_2=[x for x in New_DS.Sex.value_counts().sort_values(ascending=False).head().i
top_2
```

Out[17]:

```
['male', 'female']
```

In [18]:

```
# Coverting values in binary
for label in top_2:
    New_DS[label]=np.where(New_DS['Sex']==label,1,0)
```

In [19]:

```
# Comparing original values with dummie value
New_DS[['Sex']+top_2].head()
```

Out[19]:

	Sex	male	female
0	male	1	0
1	female	0	1
2	female	0	1
3	female	0	1
4	male	1	0

Feature Encoding on column Embarked

In [20]:

```
# Counting number of values for unique values in Embarked columns
New_DS.Embarked.value_counts().sort_values(ascending=False).head()
```

Out[20]:

```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [21]:
```
# Creating dummie value for column Embarked
dummies_2=pd.get_dummies(New_DS['Embarked'])
dummies_2.head()
```

Out[21]:

	C	Q	S
0	0	0	1
1	1	0	0
2	0	0	1
3	0	0	1
4	0	0	1

In [22]:
```
# Selecting top 3 unique values
top_3=[x for x in New_DS.Embarked.value_counts().sort_values(ascending=False).hea
top_3
```

Out[22]:
```
['S', 'C', 'Q']
```

In [23]:
```
# Coverting values in binary
for label in top_3:
    New_DS[label]=np.where(New_DS['Embarked']==label,1,0)
```

In [24]:
```
# Comparing original values with dummie value
New_DS[['Embarked']+top_3].head(6)
```

Out[24]:

	Embarked	S	C	Q
0	S	1	0	0
1	C	0	1	0
2	S	1	0	0
3	S	1	0	0
4	S	1	0	0
5	Q	0	0	1

In this lab, we have performed Feature Encoding on Titanic dataset considering the columns Sex and Embarked. Also, we learned to categories the unique values from a single column present in dataset and find the total count of number of values present in each unique values. We have also created dummies of the data and and compare with the original columns of the dataset and observe and compare the result.

Name: **Rishab Jha**
PRN:**20190802072**