Name : Aakriti Singh

PRN : 20190802006

Data Science : **Lab 5**

Date: 12th November, 2021

Aim : Apply Feature encoding techniques to perform Feature Encoding on the nominal and ordinal categorical variables.

In [1]:
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
plt.style.use('ggplot')
from sklearn.preprocessing import LabelEncoder
from sklearn import datasets
```

In [2]:
```
titanic = pd.read_csv('Titanic_Dataset.csv')
titanic.head()
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Filling missing values, applying same operations that were applied in Lab 4

In [3]:
```
titanic.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 [4]:
```
titanic.isnull().sum()[titanic.isnull().sum().apply(lambda x : x>0)]
```

Out[4]:
```
Age         177
Cabin       687
Embarked      2
dtype: int64
```

In [6]:
```
titanic.insert(4, 'Title', titanic.Name.str.extract('([A-Za-z]+)\.').values)
titanic.head()
```

Out[6]:

	PassengerId	Survived	Pclass	Name	Title	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	Mr	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	Mrs	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	Miss	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	Mrs	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	Mr	male	35.0	0	0	373450	8.0500	NaN	S

In [7]:
```
pd.crosstab(titanic['Sex'], titanic['Title']).style.background_gradient(cmap='Dark2')
```

Out[7]:

Title	Capt	Col	Countess	Don	Dr	Jonkheer	Lady	Major	Master	Miss	Mlle	Mme	Mr	Mrs	Ms	Rev	Sir
Sex																	
female	0	0	1	0	1	0	1	0	0	182	2	1	0	125	1	0	0
male	1	2	0	1	6	1	0	2	40	0	0	0	517	0	0	6	1

In [8]:
```
titanic.dropna(inplace=True)
```

In [9]:
```
titanic.groupby('Title').Age.mean()
```

Out[9]:
```
Title
Capt        70.000000
Col         56.000000
Countess    33.000000
Dr          41.666667
Lady        48.000000
Major       48.500000
Master       3.988571
Miss        27.738636
Mlle        24.000000
Mme         24.000000
Mr          40.456790
Mrs         38.236842
Sir         49.000000
Name: Age, dtype: float64
```

In [10]:
```
replace = ['Capt', 'Col', 'Countess', 'Don', 'Jonkheer', 'Lady', 'Major', 'Mlle', 'Mme', 'Ms', 'Sir', 'Rev']
replace_with = ['Mr', 'Other', 'Mrs', 'Mr', 'Other', 'Miss', 'Mr', 'Miss', 'Miss', 'Miss', 'Mr', 'Other']
titanic['Title'].replace(replace, replace_with, inplace=True)
```

In [11]:
```
average_title_age = titanic.groupby('Title')['Age'].mean().astype('int16')
average_title_age
```

Out[11]:
```
Title
Dr        41
Master     3
Miss      27
Mr        41
Mrs       38
Other     56
Name: Age, dtype: int16
```

In [12]:
```
missing_age_passengers = titanic.index[titanic['Age'].isna()]
age_column_index = titanic.columns.get_loc('Age')
initial_column_index = titanic.columns.get_loc('Title')

for passenger in missing_age_passengers:
    mean_age = float(average_title_age[titanic_df.iloc[passenger, initial_column_index]])
    titanic.iloc[passenger, age_column_index] = mean_age
```

In [13]:
```
titanic.Age.isna().sum()
```

Out[13]:
```
0
```

In [14]:
```
titanic.head()
```

Out[14]:

	PassengerId	Survived	Pclass	Name	Title	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	Mrs	female	38.0	1	0	PC 17599	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	Mrs	female	35.0	1	0	113803	53.1000	C123	S
6	7	0	1	McCarthy, Mr. Timothy J	Mr	male	54.0	0	0	17463	51.8625	E46	S
10	11	1	3	Sandstrom, Miss. Marguerite Rut	Miss	female	4.0	1	1	PP 9549	16.7000	G6	S
11	12	1	1	Bonnell, Miss. Elizabeth	Miss	female	58.0	0	0	113783	26.5500	C103	S

In [15]:
```
titanic.isnull().sum()[titanic.isnull().sum().apply(lambda x: x > 0)]
```

Out[15]:
```
Series([], dtype: int64)
```

In [16]:
```
titanic.drop('Cabin', axis=1, inplace=True) # dropping entirety of Cabin feature
```

In [17]:
```
titanic.dropna(inplace=True) # dropping rows with embarked feature value missing
```

In [18]:
```
titanic.isnull().sum()[titanic.isnull().sum().apply(lambda x: x > 0)]
```

Out[18]:
```
Series([], dtype: int64)
```

In [19]:
```
titanic.drop(columns=['Name', 'Ticket'], axis=1, inplace=True)
titanic.head()
```

Out[19]:

	PassengerId	Survived	Pclass	Title	Sex	Age	SibSp	Parch	Fare	Embarked
1	2	1	1	Mrs	female	38.0	1	0	71.2833	C
3	4	1	1	Mrs	female	35.0	1	0	53.1000	S
6	7	0	1	Mr	male	54.0	0	0	51.8625	S
10	11	1	3	Miss	female	4.0	1	1	16.7000	S
11	12	1	1	Miss	female	58.0	0	0	26.5500	S

Now that all the missing values have been swiftly taken care of, using some quicker methods and some methods used from my previous lab reports. We can get started to numerically encode all the features that are of object data type.

In [20]:
```
object_dtype = titanic.select_dtypes(include=['object'])
object_dtype.head()
```

Out[20]:

	Title	Sex	Embarked
1	Mrs	female	C
3	Mrs	female	S
6	Mr	male	S
10	Miss	female	S
11	Miss	female	S

We will numerically encode the 'Title', 'Sex', and 'Embarked' features as these are the features that are of object dtype using LabelEncoder from scikit-learn

In [21]:
```
labelencoder_method = object_dtype.copy()
for column in labelencoder_method.columns:
    labelencoder_method[column] = LabelEncoder().fit_transform(labelencoder_method[column])
labelencoder_method.head()
```

Out[21]:

	Title	Sex	Embarked
1	4	0	0
3	4	0	2
6	3	1	2
10	2	0	2
11	2	0	2

Conclusion: Hence the missing values were handled and categorical variables were converted to numerical values.

In [ ]:
```
```