Experiment:-6

Objective:- Data Preprocessing on Titanic Data

```
import pandas as pd
```

```
data = pd.read_csv('titanic-data.csv')
```

```
data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence	female	38.0	1	0	PC 17599	71.2833

```
data.dtypes
```

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

```
data.columns
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

Explore the columns

Unsupported Cell Type. Double-Click to inspect/edit the content.

Passangers who survived vs not survived

```
data['Survived'].value_counts()
```

```
0    549
1    342
Name: Survived, dtype: int64
```

```
data['Survived']==0
```

```
0       True
1      False
2      False
3      False
4       True
       ...
886     True
887    False
888     True
889    False
890     True
Name: Survived, Length: 891, dtype: bool
```

```
print('Total number of passangers in the training data...', len(data))
print('Number of passangers  who survived...', len(data[data['Survived'] == 1]))
print("Number of passangers  who didn't survived...", len(data[data['Survived'] == 0]))
```

```
Total number of passangers in the training data... 891
Number of passangers  who survived... 342
Number of passangers  who didn't survived... 549
```

```
data['Sex'].value_counts()
```

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

What is the % of men and women who survived?

```
print('% of male who survived', 100*np.mean(data['Survived'][data['Sex']=='male']))
print('% of female who survived', 100*np.mean(data['Survived'][data['Sex']=='female']))
```

```
% of male who survived 18.890814558058924
% of female who survived 74.20382165605095
```

```
np.mean(data['Survived'][data['Sex']=='male'])
```

```
0.18890814558058924
```

what is the % of men and women who survived, and then by the same token with class and age?

```
data['Pclass'].value_counts()
```

```
3    491
1    216
2    184
Name: Pclass, dtype: int64
```

```
print('% of passengers who survived in first class', 100*np.mean(data['Survived'][data['Pclass'] == 1]))
print('% of passengers who survived in second class', 100*np.mean(data['Survived'][data['Pclass'] == 2]))
print('% of passengers who survived in third class', 100*np.mean(data['Survived'][data['Pclass'] == 3]))
```

```
% of passengers who survived in first class 62.96296296296296
% of passengers who survived in second class 47.28260869565217
% of passengers who survived in third class 24.236252545824847
```

```
#data[["Pclass","Survived"]].groupby(["Pclass"], as_index = False).mean()
data[["Pclass", "Survived"]].groupby(['Pclass']).mean()
```

Pclass	Survived
1	0.629630
2	0.472826
3	0.242363

Summary

```
data.shape
```

```
(891, 12)
```

```
data.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
```

```
data['Age'].value_counts()
```

```
24.00    30
22.00    27
18.00    26
19.00    25
30.00    25
         ..
55.50     1
70.50     1
66.00     1
23.50     1
0.42      1
Name: Age, Length: 88, dtype: int64
```

```
data['Cabin']
# You can see NA values here. We have to deal with them before trainig our model
```

```
0       NaN
1       C85
2       NaN
3      C123
4       NaN
       ...
886     NaN
887     B42
888     NaN
889    C148
890     NaN
Name: Cabin, Length: 891, dtype: object
```

Unsupported Cell Type. Double-Click to inspect/edit the content.

```
data['Sex']
```

```
0        male
1      female
2      female
3      female
4        male
        ...
886      male
887    female
888    female
```

```
889      male
890      male
Name: Sex, Length: 891, dtype: object
```

```
df2 = data.copy()
df2['Sex'] = data['Sex'].apply(lambda x: 1 if x == 'male' else 0)
df2['Sex']
```

```
0      1
1      0
2      0
3      0
4      1
      ..
886    1
887    0
888    0
889    1
890    1
Name: Sex, Length: 891, dtype: int64
```

def fun(x):

if x == 'male': return 1

else: return 0

Dealing with Missing Values

```
df2 = data.copy() #dataframe copy
```

```
df2.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
int(data['Age'].mean())
```

```
29
```

```
df2['Age'] = df2['Age'].fillna(np.mean(df2['Age']))
df2.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
df2.Embarked.value_counts()
```

```
S    644
C    168
```

```
Q     77
Name: Embarked, dtype: int64
```

```
emabark = df2['Embarked'].dropna()
```

```
df2[df2['Embarked'].isnull()]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
61	62	1	1	Icard, Miss. Amelie	0	38.0	0	0	113572	80.0

```
# while there can be many ways to deal NA values for this column
# we could have drop these NA values by dropping rows as data is less
# on the otther hand we can replace it with mode value
```

```
df2['Embarked'].mode()
```

```
0    S
dtype: object
```

```
df2['Embarked'].fillna(df2['Embarked'].mode()[0], inplace=True)
```

```
df2.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         0
dtype: int64
```

```
df2['Cabin'].value_counts()
```

```
G6             4
B96 B98        4
C23 C25 C27    4
D              3
E101           3
              ..
B4             1
B102           1
A6             1
E10            1
A14            1
Name: Cabin, Length: 147, dtype: int64
```

```
df2['Cabin'].mode()
```

```
0        B96 B98
1    C23 C25 C27
2             G6
dtype: object
```

```
df2['Cabin'].fillna(df2['Cabin'].mode()[0], inplace=True)
```

```
df2.isnull().sum()
```

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

```
df2.corr()
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Pa
PassengerId	1.000000	-0.005007	-0.035144	0.042939	0.033207	-0.057527	-0.001
Survived	-0.005007	1.000000	-0.338481	-0.543351	-0.069809	-0.035322	0.081
Pclass	-0.035144	-0.338481	1.000000	0.131900	-0.331339	0.083081	0.018
Sex	0.042939	-0.543351	0.131900	1.000000	0.084153	-0.114631	-0.245
Age	0.033207	-0.069809	-0.331339	0.084153	1.000000	-0.232625	-0.179
SibSp	-0.057527	-0.035322	0.083081	-0.114631	-0.232625	1.000000	0.414
Parch	-0.001652	0.081629	0.018443	-0.245489	-0.179191	0.414838	1.000
Fare	0.012658	0.257307	-0.549500	-0.182333	0.091566	0.159651	0.216