In [1]:
```
### IMPORTING ALL THE NECESSARY LIBRARIES
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [48]:
```
### READING OUR TRAIN AND TEST DATASET
train=pd.read_csv('C:\Users\sanup\Desktop\titanic_train.csv')
test=pd.read_csv('C:\Users\sanup\Desktop\titanic_test.csv')
```

In [3]:
```
train.head()
```

Out[3]:

	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

In [4]:
```
test.head()
```

Out[4]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

In [5]:
```
train.shape,test.shape
```

Out[5]:
```
((891, 12), (418, 11))
```

Checking For Null Values

In [6]:
```
null_train=train.isnull().sum()/len(train)
null_test=test.isnull().sum()/len(test)
```

In [7]:
```
print('% Null in Train data')
null_train.sort_values(ascending=False)
```

```
% Null in Train data
```

Out[7]:
```
Cabin          0.771044
Age            0.198653
Embarked       0.002245
Fare           0.000000
Ticket         0.000000
Parch          0.000000
SibSp          0.000000
Sex            0.000000
Name           0.000000
Pclass         0.000000
Survived       0.000000
PassengerId    0.000000
dtype: float64
```

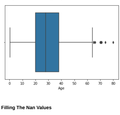
In [8]:
```
print('% Null in Test data')
null_test.sort_values(ascending=False)
```

```
% Null in Test data
```

Out[8]:
```
Cabin          0.782297
Age            0.205742
Fare           0.002392
Embarked       0.000000
Ticket         0.000000
Parch          0.000000
SibSp          0.000000
Sex            0.000000
Name           0.000000
Pclass         0.000000
PassengerId    0.000000
dtype: float64
```

In [9]:
```
### HERE BOXPLOT IS USED TO CHECK WHETHER OUTLIERS ARE PRESENT OR NOT
sns.boxplot(x=train['Age'])
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f4a5c8d520>
```



Filling The Nan Values

In [10]:
```
### AS THERE ARE OUTLIERS IN OUR TRAIN DATASET, SO WE ARE USING MEDIAN NOT MEAN
train['Age']=train['Age'].fillna(train['Age'].median())
train['Age'].isnull().sum()
```

Out[10]:
```
0
```

In [11]:
```
train['Embarked'].value_counts()
```

Out[11]:
```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [12]:
```
train['Embarked']=train['Embarked'].fillna('S')
train['Embarked'].isnull().sum()
```

Out[12]:
```
0
```

In [13]:
```
train.drop('Cabin',axis=1,inplace=True)
```

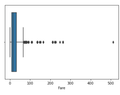
In [14]:
```
train.head()
```

Out[14]:

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

In [15]:
```
sns.boxplot(x=test['Fare'])
```

Out[15]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f4a6441730>
```



In [16]:
```
test['Fare']=test['Fare'].fillna(test['Fare'].median())
test['Fare'].isnull().sum()
```

Out[16]:
```
0
```

In [17]:
```
test['Age']=test['Age'].fillna(test['Age'].median())
test['Age'].isnull().sum()
```

Out[17]:
```
0
```

In [18]:
```
test.drop('Cabin',axis=1,inplace=True)
```

In [19]:
```
train.isnull().sum()
```

Out[19]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

In [20]:
```
test.isnull().sum()
```

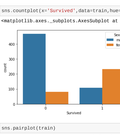
Out[20]:
```
PassengerId    0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

In [21]:
```
### WE ARE USING CORRELATION TO IDENTIFY WHICH ARE THE IMPORTANT FEATURES IN OUR DATASET
plt.figure(figsize=(10,7))
corr = train.corr()
ax = sns.heatmap(
    corr,
    vmin=-1, vmax=1, center=0,
    cmap=sns.diverging_palette(20, 220, n=200),
    square=True,annot=True
)
```

In [22]:
```
### WE CAN SEE FROM ABOVE HEATMAP THAT PCLASS AND FARE ARE MOST CORRELATED FEATURES
```

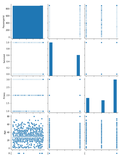
In [23]:
```
sns.countplot(x='Survived',data=train,hue='Pclass')
```

Out[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f4a64c0370>
```



In [24]:
```
sns.countplot(x='Survived',data=train,hue='Sex')
```

Out[24]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f4a67d0e60>
```



In [25]:
```
sns.pairplot(train)
```

Out[25]:
```
<seaborn.axisgrid.PairGrid at 0x1f4a684a80>
```

Converting categorical features into some numeric value

In [26]:
```
from sklearn import preprocessing
```

In [27]:
```
LE=preprocessing.LabelEncoder()
```

In [28]:
```
train['Sex']=LE.fit_transform(train['Sex'])
train['Embarked']=LE.fit_transform(train['Embarked'])
```

In [29]:
```
test['Sex']=LE.fit_transform(test['Sex'])
test['Embarked']=LE.fit_transform(test['Embarked'])
```

Implementing the algorithm

In [30]:
```
from sklearn.linear_model import LogisticRegression
```

In [31]:
```
LR=LogisticRegression()
```

In [32]:
```
test.head()
```

Out[32]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	892	3	Kelly, Mr. James	1	34.5	0	0	330911	7.8292	1
1	893	3	Wilkes, Mrs. James (Ellen Needs)	0	47.0	1	0	363272	7.0000	2
2	894	2	Myles, Mr. Thomas Francis	1	62.0	0	0	240276	9.6875	1
3	895	3	Wirz, Mr. Albert	1	27.0	0	0	315154	8.6625	2
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	0	22.0	1	1	3101298	12.2875	2

In [33]:
```
X_train=train.drop(['Survived','PassengerId','Name','Ticket'],axis=1)
y_train=train['Survived']
X_test=test.drop(['PassengerId','Name','Ticket'],axis=1)
```

In [34]:
```
LR.fit(X_train,y_train)
```

```
C:\Users\sanup\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:762: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[34]:
```
LogisticRegression()
```

In [35]:
```
y_pred=LR.predict(X_test)
```

In [36]:
```
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
```

In [37]:
```
print('Accuracy = {}'.format(LR.score(X_train,y_train)*100))
```

```
Accuracy = 80.02246605611335
```

In [38]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [39]:
```
DC=DecisionTreeClassifier()
```

In [40]:
```
DC.fit(X_train,y_train)
```

Out[40]:
```
DecisionTreeClassifier()
```

In [41]:
```
print('Accuracy = {}'.format(DC.score(X_train,y_train)*100))
```

```
Accuracy = 97.97979797979798
```

In [42]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [43]:
```
RC=RandomForestClassifier()
```

In [44]:
```
RC.fit(X_train,y_train)
```

Out[44]:
```
RandomForestClassifier()
```

In [45]:
```
print('Accuracy = {}'.format(RC.score(X_train,y_train)*100))
```

```
Accuracy = 97.97979797979798
```

In [ ]:
```
```