```
#%% md
IMPORTING REQUIRED LIBRARIES
#%%
import pandas as pd
#%%
import numpy as np
#%%
import matplotlib.pyplot as plt
#%%
import seaborn as sns
#%%
train = pd.read_csv("train.csv")
#%%
test = pd.read_csv("test.csv")
#%% md
#Viewing data and different features

#%%
train.head()
#%% md
train.shape
#%%
train.columns
#%%
train["Sex"].value_counts()
#%% raw
Data Visualization
#%%
#visualizing survivals based on gender
train["Died"] = 1 - train["Survived"]
train.groupby("Sex").agg("sum")[["Survived","Died"]].plot(kind="bar",figsize=(10,5),stacked=True)
#%%
#Visualizing survivals based on fare
figure = plt.figure(figsize=(16,7))
plt.hist([train[train["Survived"] == 1]["Fare"],train[train["Survived"] == 0]["Fare"]],stacked=True, bins = 50, label = ['Survived','Dead'])
plt.xlabel("Fare")
plt.ylabel("Number of passengers")
plt.legend()
```

```
#%% md
Processing training data

#%%
#Cleaning the data by removing irrelevant columns
df1 = train.drop(["Name","Ticket","Cabin","PassengerId","Died"],axis=1)
df1.head(100)
#%%
df1.isnull().sum()
#%%
#converting the categorical features "sex" and "Embarked" into numerical values 0 and 1
df1.Sex = df1.Sex.map({"female":0,"male":1})
df1.Embarked=df1.Embarked.map({"S":0,"C":1,"Q":2,"nan":"NaN"})
df1.head()
#%%
#mean age of each sex
mean_age_men = df1[df1["Sex"]==1]["Age"].mean()
mean_age_women = df1[df1["Sex"]==0]["Age"].mean()
#%%
#Filling all the null values in 'Age' with respective mean age
df1.loc[(df1.Age.isnull()) & (df1['Sex']==0),'Age']=mean_age_women
df1.loc[(df1.Age.isnull()) & (df1['Sex']==1),'Age']=mean_age_men
#%%
#let's check for the Null values again now
df1.isnull().sum()
#%%
#since there exist 2 null values in the embarked column, let's drop thhose rows containing null values
df1.dropna(inplace=True)
#%%
df1.isnull().sum()
#%%
#Doing Feature Scaling to standardize the independent features present in the data in a fixed range
df1.Age = (df1.Age-min(df1.Age))/(max(df1.Age)-min(
df1.Age))
df1.Fare = (df1.Fare-min(df1.Fare))/(max(df1.Fare)-min(df1.Fare))
df1.describe()
#%% md
Creating model

#%%
#Splitting the data for training and testing
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    df1.drop(['Survived'], axis=1),
    df1.Survived,
    test_size= 0.2,
    random_state=0,
    stratify=df1.Survived)
#%% md
Logistic Regression
#%%
from sklearn.linear_model import LogisticRegression
lrmod = LogisticRegression()
lrmod.fit(X_train, y_train)
from sklearn.metrics import accuracy_score
y_predict = lrmod.predict(X_test)
accuracy_score(y_test, y_predict)
#%%
#Confusion Matrix
from sklearn.metrics import confusion_matrix
cma=confusion_matrix(y_test, y_predict)
sns.heatmap(cma,annot=True)
#%% md
Processing test data
#%%
#Viewing test data
test.head()
#%%
#Cleaning the data by removing irrelevant columns
df2=test.drop(['PassengerId','Name','Ticket','Cabin'], axis=1)
df2
#%%
```

```
#Converting the categorical features 'Sex' and 'Embarked' into numerical values 0 & 1
df2.Sex=df2.Sex.map({'female':0, 'male':1})
df2.Embarked=df2.Embarked.map({'S':0, 'C':1, 'Q':2,'nan':'NaN'})
df2.head()
#%%
#Let's check for the null values
df2.isnull().sum()
#%%
#Finding mean age
mean_age_men2=df2[df2['Sex']==1]['Age'].mean()
mean_age_women2=df2[df2['Sex']==0]['Age'].mean()
#%%
#Filling all the null values in 'Age' and 'Fare' with respective mean age and mean fare
df2.loc[(df2.Age.isnull()) & (df2['Sex']==0),'Age']=mean_age_women2
df2.loc[(df2.Age.isnull()) & (df2['Sex']==1),'Age']=mean_age_men2
df2['Fare']=df2['Fare'].fillna(df2['Fare'].mean())
#%%
df2.isnull().sum()
#%%
#Doing Feature Scaling to standardize the independent features present in the data in a fixed range
df2.Age = (df2.Age-min(df2.Age))/(max(df2.Age)-min(df2.Age))
df2.Fare = (df2.Fare-min(df2.Fare))/(max(df2.Fare)-min(df2.Fare))
df2.describe()
#%% md
Prediction
#%%
prediction = lrmod.predict(df2)
prediction
#%%
prediction
#%%
submission = pd.DataFrame({"PassengerId": test["
PassengerId"],
                           "Survived": prediction})
submission.to_csv('submission.csv', index=False)
#%%
prediction_df = pd.read_csv('submission.csv')
#%%
sns.countplot(x='Survived', data=prediction_df)
#%%
sns.countplot(x='Survived', data=prediction_df)sns.countplot(x='Survived', data=prediction_df)
#%%

```