Shahjan Khan

Prediction of Survivals on Titanic Using Machine Learning Model

Problem Statement - The RMS Titanic was a British passenger liner that sank in the North Atlantic Ocean in the early morning hours of 15 April 1912, after it collided with an iceberg during its maiden voyage from Southampton to New York City. There were an estimated 2,224 passengers and crew aboard the ship, and more than 1,500 died, making it one of the deadliest commercial peacetime maritime disasters in modern history. The RMS Titanic was the largest ship afloat at the time it entered service and was the second of three Olympic-class ocean liners operated by the White Star Line. The Titanic was built by the Harland and Wolff shipyard in Belfast. Thomas Andrews, her architect, died in the disaster.

In this Project, we will analyze the Titanic data set and make two predictions. One prediction to see which passengers on board the ship would survive and then another prediction to see if we wouldn't survive.

- **Steps to be taken in the Project is sub-divided into the following sections. These are:**
  - Load the necessary libraries such as Numpy , Pandas , sklearn.model etc.
  - Loading the dataset as csv file and showing first ten rows.
  - Drop the unnecessary columns from the data.
  - Calculate statistical values and round them up to 3 decimal places.
  - Checking for null values and return their sum of numbers of true values in each column.
  - Handle the null by mean of all values fill into them.
  - Visualization of Passenger Survival data using Data Visualization with Python.
  - Data preprocessing or (Data cleaning) performed by the one hot encoding in this process we change categorical data into numerical data and the technique is called feature Engineering.
  - Splitting the cleaned data into dependent and independent variables.
  - Splitting the data into train and test sets with train_test_split using sklearn library.
  - Import different kind of Classification Models and Train that model with the help of .fit().
  - Predicting the trained models and then checking their accuracy score and confusion metrics of the model using confusion metrics & accuracy score.

- Then recall the train_test_split and split the data into training and testing set with different models.
- Then predicting the trained models and checking the accuracy of model and print the accuracy difference.
- And finally predict whether the Titanic Survivals classification generated or not.

➢ **Step-1 –** Loading Necessary Libraries used in machine learning.

Import Necessary Libraries.

```
[3] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as snr
    from sklearn.preprocessing import OneHotEncoder
```

➢ **Step-2 -** Loading the dataset as csv file and showing first ten rows.

```
data=pd.read_csv("/content/tested.csv")
data.head(10)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
5	897	0	3	Svensson, Mr. Johan Cervin	male	14.0	0	0	7538	9.2250	NaN	S
6	898	1	3	Connolly, Miss. Kate	female	30.0	0	0	330972	7.6292	NaN	Q
7	899	0	2	Caldwell, Mr. Albert Francis	male	26.0	1	1	248738	29.0000	NaN	S
8	900	1	3	Abrahim, Mrs. Joseph (Sophie Halaut Easu)	female	18.0	0	0	2657	7.2292	NaN	C
9	901	0	3	Davies, Mr. John Samuel	male	21.0	2	0	A/4 48871	24.1500	NaN	S

Shahjan Khan

- **Step-3** - Drop the unnecessary columns from the data.

Drop Unnecessary values

```
data_new=data.drop(['PassengerId','Name','Cabin','Ticket'],axis=1)
data_new.head(10)
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	34.5	0	0	7.8292	Q
1	1	3	female	47.0	1	0	7.0000	S
2	0	2	male	62.0	0	0	9.6875	Q
3	0	3	male	27.0	0	0	8.6625	S
4	1	3	female	22.0	1	1	12.2875	S
5	0	3	male	14.0	0	0	9.2250	S
6	1	3	female	30.0	0	0	7.6292	Q
7	0	2	male	26.0	1	1	29.0000	S
8	1	3	female	18.0	0	0	7.2292	C
9	0	3	male	21.0	2	0	24.1500	S

Connected to Python 3 Google Compute Engine backend

- **Step-4** - Calculate statistical values and round them up to 3 decimal places.

Calculates statistical values and rounds them to 3 decimal places.

```
data_new.describe().round(3)
```

	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.000	418.000	332.000	418.000	418.000	417.000
mean	0.364	2.266	30.273	0.447	0.392	35.627
std	0.482	0.842	14.181	0.897	0.981	55.908
min	0.000	1.000	0.170	0.000	0.000	0.000
25%	0.000	1.000	21.000	0.000	0.000	7.896
50%	0.000	3.000	27.000	0.000	0.000	14.454
75%	1.000	3.000	39.000	1.000	0.000	31.500
max	1.000	3.000	76.000	8.000	9.000	512.329

Shahjan Khan

- **Step-5** - Checking for null values and return their sum of numbers of true values in each column.

Mark null values as True and returns sum of number of True values in each column

```
[7] data_new.isnull().sum()
```

```
Survived     0
Pclass       0
Sex          0
Age         86
SibSp        0
Parch        0
Fare         1
Embarked     0
dtype: int64
```

- **Step-6** - Handle the null by mean of all values fill into them.

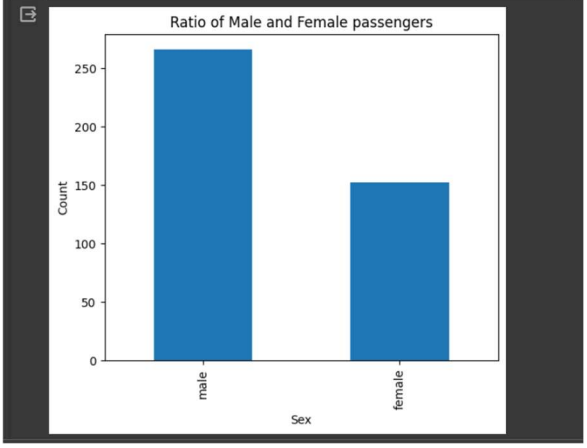
Handling Null Values and Pre Processing the dataset.

```
[ ] data_new['Age']=data_new['Age'].fillna(data_new['Age'].mean())
    data_new['Fare']=data_new['Fare'].fillna(data_new['Fare'].mean())
    data_new.isnull().sum()
```

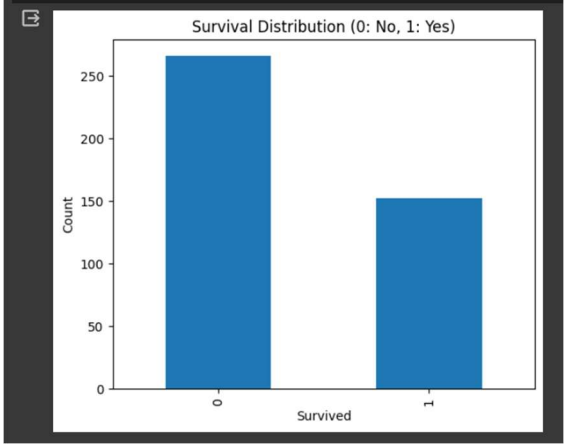
```
Survived    0
Pclass      0
Sex         0
Age         0
SibSp       0
Parch       0
Fare        0
Embarked    0
dtype: int64
```

- **Step-7** - Visualization of Passenger Survival using Data Visualization with Python.

```
ax=data_new['Sex'].value_counts().plot(kind='bar',x = 'Survived',figsize=(6,5))
plt.title("Ratio of Male and Female passengers")
plt.ylabel("Count")
plt.xlabel("Sex")
plt.show()
```

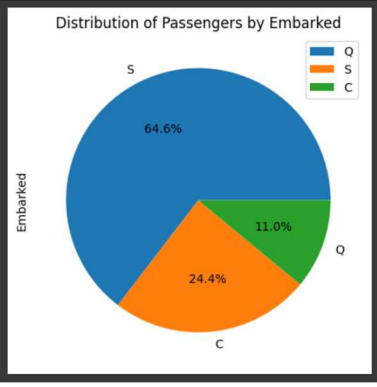



```
ax=data_new['Survived'].value_counts().plot(kind='bar',figsize=(6,5))
plt.title("Survival Distribution (0: No, 1: Yes)")
plt.ylabel("Count")
plt.xlabel("Survived")
plt.show()
```

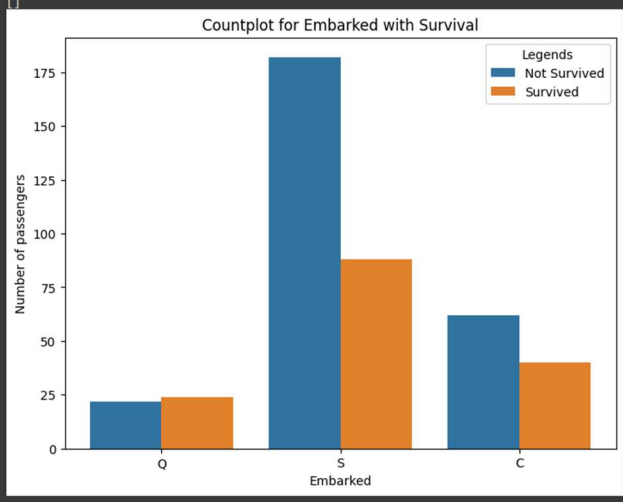


Shahjan Khan

```
ax=data_new['Embarked'].value_counts().plot(kind='pie',figsize=(6,5),autopct='%1.1f%%')
plt.title("Distribution of Passengers by Embarked")
plt.legend(["Q", "S", "C"])
plt.show()
```




```
+ Code + Text
_, ax = plt.subplots(figsize = (8, 6))
snr.countplot(data = data_new, x = "Embarked", hue = "Survived", ax = ax)
ax.set_title("Countplot for Embarked with Survival")
ax.set_xlabel("Embarked")
ax.set_xticklabels(["Q", "S", "C"])
ax.set_ylabel("Number of passengers")
ax.legend(title = "Legends", labels = ["Not Survived", "Survived"])
plt.plot()
```



- **Step-8 -** Data preprocessing or (Data cleaning) performed by the one hot encoding in this process we change categorical data into numerical data and the technique is called feature Engineering.

Change the catagorical data into numerical data by using OneHotEncoding.

```
data_new['Sex']=data_new['Sex'].apply({'male':1,'female':0}.get)
data_new['Embarked']=data_new['Embarked'].apply({'S':1,'Q':2,'C':3}.get)
data_new.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	1	34.5	0	0	7.8292	2
1	1	3	0	47.0	1	0	7.0000	1
2	0	2	1	62.0	0	0	9.6875	2
3	0	3	1	27.0	0	0	8.6625	1
4	1	3	0	22.0	1	1	12.2875	1

- **Step-9** - Splitting the cleaned data into dependent and independent variables.

Deviding the data into Dependent and Independent.

```
[ ]  x=data_new.drop(['Survived'],axis=1)
     y=data_new['Survived']
```

- **Step-10** - Splitting the data into train and test sets with train_test_split using sklearn library.

Deviding the cleaned data into training and testing sets.

```
[ ]  from sklearn.model_selection import train_test_split
     x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.8)
```

- **Step-11** - Import first machine learning model K-Nearest neighbor taking n_neighbor=5.

Creating first machine learning model 'k-nearest neighbour'.

```
[ ]  from sklearn.neighbors import KNeighborsClassifier
     knn=KNeighborsClassifier(n_neighbors=5)
```

- **Step-12** - Train the model using .fit() function.

Train the model

```
[ ]  knn.fit(x_train,y_train)
```

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

- **Step-13** - Predict the trained model using .predict() function.

Make predictions on model

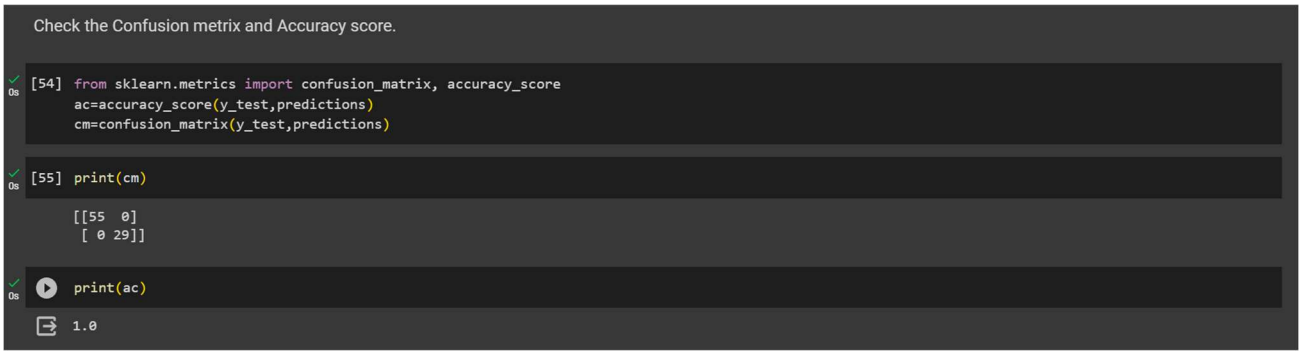
```
[ ]  predictions=knn.predict(x_test)
     print(predictions)
```

```
[1 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 1 1 0 0
 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0
 0 0 0 0 0 0 0 0 0 0]
```

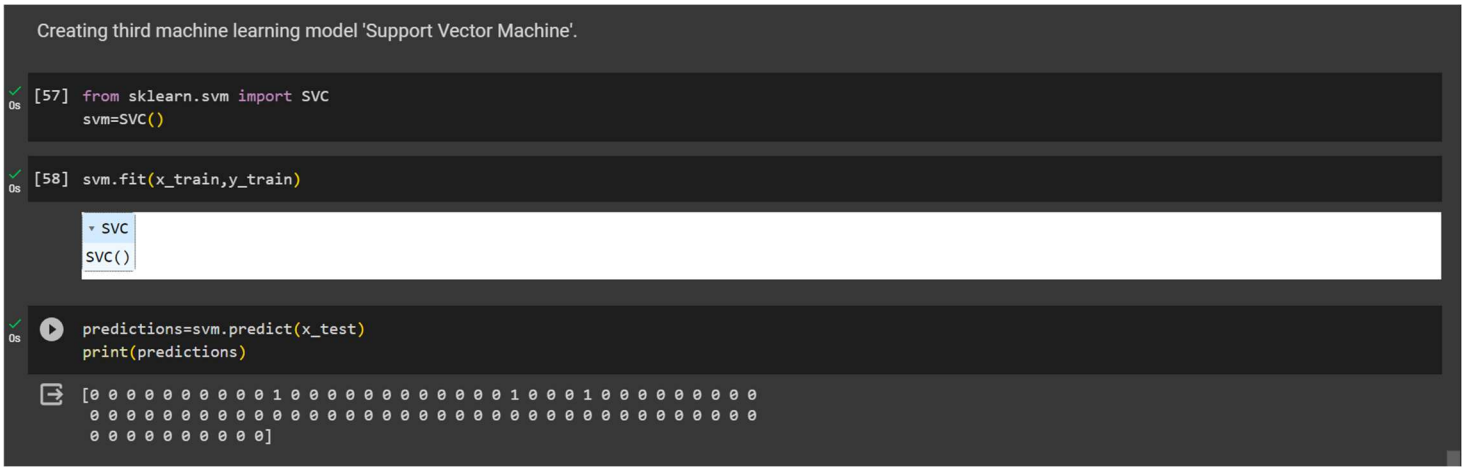
- **Step-14** - Check the accuracy score and print a confusion metrics with confusion metrics & accuracy score.

- **Step-15** – Import the Second Machine Learning Model Decision Tree and train model and then make prediction.

- **Step-16** - Print a confusion metrics and check accuracy score for Decision Tree Model.



- **Step-17** - Import the Third Machine Learning Model Support Vector Machine and train model and then make prediction.



- **Step-18** - Print a confusion metrics and check accuracy score for Support Vector Machine Model.

Conclusion - The purpose of Project is to use the existing features of passengers onboard Titanic as predictors to predict their survival outcome, for 0 being dead and 1 being survived from the tragic ship crash. The K-Nearest neighbor is the is first classification model performed with k=5, and the I use Random Forest Classifier, and then I use the Support Vector Machine (SVM) analysis to improved performance. It is certain through the practice of model improvement, the SVM analysis is better performed than the K-Nearest Neighbor classification analysis and Random Forest analysis is also performed better than K-Nearest Neighbor for prediction accuracy.

However, even from the all three classification model, we can easily see that the Titanic survival outcome is highly depended on several predictors, such as sex, age and passenger class. In particular, Ratio of survived people are more while keeping other predictors conditions constant and lastly, people from a lower class are less likely to survived keeping other predictors conditions constant.