

Program No:	12
Roll No :	1408
Title of Program :	AdaBoost
Objective :	AdaBoost using Titanic Dataset

## Source Code:

To Build a classifier to Predict Survival using Titanic Dataset

```
from google.colab import drive
drive.mount('/content/drive')
```

```
import numpy as np
import pandas as pd
titanic = pd.read_csv("/content/drive/MyDrive/AIML_datasets/titanic.csv")
titanic
```

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
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

## DATA PreProcessing

```
titanic = titanic.drop(['PassengerId', 'Name'], axis=1)
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Survived    891 non-null    int64
1   Pclass      891 non-null    int64
2   Sex         891 non-null    object
3   Age         714 non-null    float64
4   SibSp       891 non-null    int64
5   Parch       891 non-null    int64
6   Ticket      891 non-null    object
7   Fare        891 non-null    float64
8   Cabin       204 non-null    object
9   Embarked    889 non-null    object
dtypes: float64(2), int64(4), object(4)
memory usage: 69.7+ KB
```

### #Count the Null Values

```
print(titanic.isnull().sum())
```

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

```
titanic.describe()
```

Index	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.0	891.0	714.0	891.0	891.0	891.0
mean	0.3838383838383838	2.308641975308642	29.69911764705882	0.5230078563411896	0.38159371492704824	32.204207968574636
std	0.4865924542648575	0.836071240977049	14.526497332334042	1.1027434322934317	0.8060572211299483	49.6934285971809
min	0.0	1.0	0.42	0.0	0.0	0.0
25%	0.0	2.0	20.125	0.0	0.0	7.9104
50%	0.0	3.0	28.0	0.0	0.0	14.4542
75%	1.0	3.0	38.0	1.0	0.0	31.0
max	1.0	3.0	80.0	8.0	6.0	512.3292

### #Unique Values

```
titanic['Sex'].unique()
array(['male', 'female'], dtype=object)
titanic['Pclass'].unique()
array([3, 1, 2])
```

```
titanic['SibSp'].unique()
array([1, 0, 3, 4, 2, 5, 8])
titanic['Parch'].unique()
array([0, 1, 2, 5, 3, 4, 6])
#Encode the data to convert into numeric format
pd.get_dummies(titanic['Pclass'])
```

	1	2	3
0	False	False	True
1	True	False	False
2	False	False	True
3	True	False	False
4	False	False	True
...	...	...	...
886	False	True	False
887	True	False	False
888	False	False	True
889	True	False	False
890	False	False	True

891 rows × 3 columns

### #Gender OneHotEncoding

```
pd.get_dummies(titanic['Sex'])
```

	female	male
0	False	True
1	True	False
2	True	False
3	True	False
4	False	True
...	...	...
886	False	True
887	True	False
888	True	False
889	False	True
890	False	True

891 rows × 2 columns

### #Create a new dataset using encoding

```
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Survived    891 non-null    int64
1   Pclass      891 non-null    int64
2   Sex         891 non-null    object
3   Age        714 non-null    float64
4   SibSp       891 non-null    int64
5   Parch      891 non-null    int64
6   Ticket     891 non-null    object
7   Fare       891 non-null    float64
8   Cabin      204 non-null    object
9   Embarked   889 non-null    object
dtypes: float64(2), int64(4), object(4)
```

```
titanic['Pclass']=titanic['Pclass'].astype(str)
titanic['Parch']=titanic['Parch'].astype(str)
titanic['SibSp']=titanic['SibSp'].apply(str)
dummy_cat =
pd.get_dummies(titanic[['Pclass','SibSp','Parch','Sex','Embarked']],dtype=int)
dummy_cat.head()
```

	Pclass_1	Pclass_2	Pclass_3	SibSp_0	SibSp_1	SibSp_2	SibSp_3	SibSp_4	SibSp_5	SibSp_8	...	Parch_2	Parch_3	Parch_4	Parch_5	Parch_6	Sex_female	Sex_ma
0	0	0	1	0	1	0	0	0	0	0	...	0	0	0	0	0	0	
1	1	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	1	
2	0	0	1	1	0	0	0	0	0	0	...	0	0	0	0	0	1	
3	1	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	1	
4	0	0	1	1	0	0	0	0	0	0	...	0	0	0	0	0	0	

```
dummy_cat.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 22 columns):
```

#	Column	Non-Null Count	Dtype
0	Pclass_1	891 non-null	int64
1	Pclass_2	891 non-null	int64
2	Pclass_3	891 non-null	int64
3	SibSp_0	891 non-null	int64
4	SibSp_1	891 non-null	int64
5	SibSp_2	891 non-null	int64
6	SibSp_3	891 non-null	int64
7	SibSp_4	891 non-null	int64
8	SibSp_5	891 non-null	int64
9	SibSp_8	891 non-null	int64
10	Parch_0	891 non-null	int64
11	Parch_1	891 non-null	int64
12	Parch_2	891 non-null	int64
13	Parch_3	891 non-null	int64
14	Parch_4	891 non-null	int64
15	Parch_5	891 non-null	int64
16	Parch_6	891 non-null	int64
17	Sex_female	891 non-null	int64
18	Sex_male	891 non-null	int64
19	Embarked_C	891 non-null	int64
20	Embarked_Q	891 non-null	int64
21	Embarked_S	891 non-null	int64

```
dtypes: int64(22)
memory usage: 153.3 KB
```

```
final_df = pd.concat([titanic['Age'], dummy_cat], axis=1)
```

```
final_df
```

	Age	Pclass_1	Pclass_2	Pclass_3	SibSp_0	SibSp_1	SibSp_2	SibSp_3	SibSp_4	SibSp_5	...	Parch_2	Parch_3	Parch_4	Parch_5	Parch_6	Sex_female	Sex_male
0	22.0	0	0	1	0	1	0	0	0	0	...	0	0	0	0	0	0	0
1	38.0	1	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	1
2	26.0	0	0	1	1	0	0	0	0	0	...	0	0	0	0	0	0	1
3	35.0	1	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	1
4	35.0	0	0	1	1	0	0	0	0	0	...	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	27.0	0	1	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0
887	19.0	1	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	1
888	NaN	0	0	1	0	1	0	0	0	0	...	1	0	0	0	0	0	1
889	26.0	1	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0
890	32.0	0	0	1	1	0	0	0	0	0	...	0	0	0	0	0	0	0

891 rows x 23 columns

```
final_df.Age = final_df.Age.fillna('0')
```

```
final_df.Age.isnull().sum()
```

```
0
```

## Seperate the data in x and y

```
x = final_df
```

```
y = titanic['Survived']
```

## Split the data into train and test data

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3,
                                                    random_state = 45)
```

## Build the AdaBoost Classifier

```
from sklearn.ensemble import AdaBoostClassifier
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
#Create an instance of adaboost
```

```
clf1 =  
AdaBoostClassifier(n_estimators=100,learning_rate=1.0,random_state=45)
```

## Train the Model

```
model = clf1.fit(x_train,y_train)
```

## Test the Model

```
y_pred1 = model.predict(x_test)
```

## Find the Accuracy of the model

```
from sklearn.metrics import accuracy_score  
print("Accuracy : ",(accuracy_score(y_test,y_pred1)*100))  
Accuracy : 81.34328358208955
```

## Model2

```
#Create an instance of adaboost  
clf2 =  
AdaBoostClassifier(base_estimator=RandomForestClassifier(),random_state=45)  
model2 = clf2.fit(x_train,y_train)  
y_pred1 = model.predict(x_test)  
  
from sklearn.metrics import accuracy_score  
print("Accuracy : ",(accuracy_score(y_test,y_pred1)*100))  
  
Accuracy : 81.34328358208955
```



**MUMBAI EDUCATIONAL TRUST**

**MET Institute of Computer Science**

THE MET LEAGUE OF COLLEGES  
**MET**  
AS SHARP AS YOU CAN GET  
**Bhujbal Knowledge City**