

Predict survival from titanic crash using Naive bayes

In [51]: `import pandas as pd`

In [52]: `df = pd.read_csv("titanic.csv")
df.head()`

Out[52]:

	PassengerId	Name	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	Braund, Mr. Owen Harris	3	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	Cumings, Mrs. John Bradley (Florence Briggs Th...)	1	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	Heikkinen, Miss. Laina	3	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	1	0	113803	53.1000	C123	
4	5	Allen, Mr. William Henry	3	male	35.0	0	0	373450	8.0500	NaN	

In [53]: `#Deroping the unnecessary columns
df.drop(['PassengerId', 'Name', 'SibSp', 'Parch', 'Ticket', 'Cabin', 'Embarked'], axis=
df.head()`

Out[53]:

	Pclass	Sex	Age	Fare	Survived
0	3	male	22.0	7.2500	0
1	1	female	38.0	71.2833	1
2	3	female	26.0	7.9250	1
3	1	female	35.0	53.1000	1
4	3	male	35.0	8.0500	0

```
In [54]: df.Sex = df.Sex.map({'male': 1, 'female': 2}) # Replacing male with 1 and female
df.head()
```

Out[54]:

	Pclass	Sex	Age	Fare	Survived
0	3	1	22.0	7.2500	0
1	1	2	38.0	71.2833	1
2	3	2	26.0	7.9250	1
3	1	2	35.0	53.1000	1
4	3	1	35.0	8.0500	0

```
In [55]: df.columns[df.isna().any()] # Checking is there is any column which have NaN value
```

Out[55]: Index(['Age'], dtype='object')

```
In [56]: df.Age[:10] # There is NaN value in age
```

Out[56]:

0	22.0
1	38.0
2	26.0
3	35.0
4	35.0
5	NaN
6	54.0
7	2.0
8	27.0
9	14.0

Name: Age, dtype: float64

```
In [57]: df.Age = df.Age.fillna(df.Age.mean()) # fillna function is used to remove the NaN value
df.head()
```

Out[57]:

	Pclass	Sex	Age	Fare	Survived
0	3	1	22.0	7.2500	0
1	1	2	38.0	71.2833	1
2	3	2	26.0	7.9250	1
3	1	2	35.0	53.1000	1
4	3	1	35.0	8.0500	0

```
In [58]: X = df.iloc[:,0:4].values
Y = df.iloc[:,4].values
```

In [59]: X

```
Out[59]: array([[ 3.         ,  1.         , 22.         ,  7.25        ],
 [ 1.         ,  2.         , 38.         , 71.2833       ],
 [ 3.         ,  2.         , 26.         ,  7.925        ],
 ...,
 [ 3.         ,  2.         , 29.69911765, 23.45        ],
 [ 1.         ,  1.         , 26.         , 30.         ],
 [ 3.         ,  1.         , 32.         ,  7.75        ]])
```

```
In [60]: 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)
```

```
In [61]: from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
```

```
In [62]: model.fit(X_train,y_train)
```

```
Out[62]: GaussianNB(priors=None, var_smoothing=1e-09)
```

```
In [63]: y_pred=model.predict(X_test)
```

```
In [64]: from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

```
Out[64]: array([[141,  27],
 [ 29,  71]], dtype=int64)
```

```
In [65]: from sklearn import metrics
metrics.accuracy_score(y_test,y_pred)
```

```
Out[65]: 0.7910447761194029
```