Naive Bayes : Predicting survival from titanic crash

In [28]:

```
import pandas as pd
```

In [29]:

```
df = pd.read_csv("C:/Users/91920/Machine Learning/EDA/titanic.csv")
df.head()
```

Out[29]:

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

In [30]:

```
df.drop(['PassengerId','Name','SibSp','Parch','Ticket','Cabin','Embarked'],axis='columns',i
df.head()
```

Out[30]:

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

In [34]:

```
x = df.drop('Survived',axis='columns')  #x
y = df.Survived  #y

x.head()
```

Out[34]:

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

In [6]:

```
#inputs.Sex = inputs.Sex.map({'male': 1, 'female': 2})
```

In [35]:

```
dummies = pd.get_dummies(x.Sex)
dummies.head(3)
```

Out[35]:

	female	male
0	0	1
1	1	0
2	1	0

In [36]:

```
x = pd.concat([x,dummies],axis='columns')
x.head(3)
```

Out[36]:

	Pclass	Sex	Age	Fare	female	male
0	3	male	22.0	7.2500	0	1
1	1	female	38.0	71.2833	1	0
2	3	female	26.0	7.9250	1	0

I am dropping male column as well because of dummy variable trap theory. One column is enough to repressent male vs female

In [37]:

```
x.drop(['Sex','male'],axis='columns',inplace=True)
x.head(3)
```

Out[37]:

	Pclass	Age	Fare	female
0	3	22.0	7.2500	0
1	1	38.0	71.2833	1
2	3	26.0	7.9250	1

In [38]:

```
x.columns[inputs.isna().any()]
```

Out[38]:

```
Index(['Fare'], dtype='object')
```

In [40]:

```
x.Age[:10]
```

Out[40]:

```
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 [41]:

```
x.Age = x.Age.fillna(x.Age.mean())
x.head()
```

Out[41]:

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

In [13]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(inputs,target,test_size=0.3)
```

In [15]:

```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
```

In [16]:

```
model.fit(X_train,y_train)
```

Out[16]:

```
GaussianNB()
```

In [17]:

```
model.score(X_test,y_test)
```

Out[17]:

```
0.7835820895522388
```

In [18]:

```
X_test[0:10]
```

Out[18]:

	Pclass	Age	Fare	female
667	3	29.699118	7.7750	0
317	2	54.000000	14.0000	0
358	3	29.699118	7.8792	1
457	1	29.699118	51.8625	1
494	3	21.000000	8.0500	0
609	1	40.000000	153.4625	1
837	3	29.699118	8.0500	0
645	1	48.000000	76.7292	0
250	3	29.699118	7.2500	0
356	1	22.000000	55.0000	1

In [19]:

```
y_test[0:10]
```

Out[19]:

```
667    0
317    0
358    1
457    1
494    0
609    1
837    0
645    1
250    0
356    1
Name: Survived, dtype: int64
```

In [21]:

```
model.predict(X_test[0:10])
```

Out[21]:

```
array([0, 0, 1, 1, 0, 1, 0, 1, 0, 1], dtype=int64)
```

In [26]:

```
model.predict_proba(X_test[:10])
```

Out[26]:

```
array([[9.63966132e-01, 3.60338679e-02],
       [9.23917217e-01, 7.60827829e-02],
       [4.12530199e-01, 5.87469801e-01],
       [5.26372453e-02, 9.47362755e-01],
       [9.58022247e-01, 4.19777527e-02],
       [9.01992358e-05, 9.99909801e-01],
       [9.64002889e-01, 3.59971107e-02],
       [4.74505251e-01, 5.25494749e-01],
       [9.63890626e-01, 3.61093742e-02],
       [4.23870804e-02, 9.57612920e-01]])
```

Calculate the score using cross validation

In [27]:

```
from sklearn.model_selection import cross_val_score
cross_val_score(GaussianNB(),X_train, y_train, cv=5)
```

Out[27]:

```
array([0.816     , 0.784     , 0.768     , 0.77419355, 0.74193548])
```

In [ ]: