Watch Luis Serrano - Naive Bayes Classifier

In [18]:

```
import pandas as pd
df = pd.read_csv("C:/Users/prasa/Desktop/py codes/ds projects/ML/13 Naive Bayes/titanic.csv")
df.head()
```

Out[18]:

	PassengerId	Name	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
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 [19]:

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

Out[19]:

	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 [20]:

```
target=df.Survived
inputs=df.drop('Survived',axis='columns')
```

In [21]:

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

Out[21]:

	female	male
0	0	1
1	1	0
2	1	0

In [22]:

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

Out[22]:

	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 [23]:
```
inputs.drop('Sex',axis='columns',inplace=True)
inputs.head(3)
```

Out[23]:

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

In [24]:
```
inputs.columns[inputs.isna().any()]
```

Out[24]:
```
Index(['Age'], dtype='object')
```

In [26]:
```
inputs.Age[:10]
```

Out[26]:
```
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 [30]:
```
inputs.Age = inputs.Age.fillna(inputs.Age.mean()) #fill na with mean va
lue
inputs.head(6)
```

Out[30]:

	Pclass	Age	Fare	female	male
0	3	22.000000	7.2500	0	1
1	1	38.000000	71.2833	1	0
2	3	26.000000	7.9250	1	0

	Pclass	Age	Fare	female	male
3	1	35.000000	53.1000	1	0
4	3	35.000000	8.0500	0	1
5	3	29.699118	8.4583	0	1

In [31]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(inputs,target,test_
size=0.2)
```

In [33]:

```
len(X_train)
```

Out[33]:

```
712
```

In [34]:

```
len(X_test)
```

Out[34]:

```
179
```

In [35]:

```
len(inputs)
```

Out[35]:

```
891
```

In [37]:

```
len(X_train)
```

Out[37]:

```
712
```

In [38]:

```
len(inputs)
```

Out[38]:

```
891
```

In [42]:

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

In [43]:

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

```
Out[43]: GaussianNB(priors=None, var_smoothing=1e-09)
```

```
In [44]: model.score(X_test,y_test)
```

```
Out[44]: 0.8435754189944135
```

```
In [45]: X_test[:10]
```

```
Out[45]:
```

	Pclass	Age	Fare	female	male
824	3	2.000000	39.6875	0	1
593	3	29.699118	7.7500	1	0
154	3	29.699118	7.3125	0	1
786	3	18.000000	7.4958	1	0
61	1	38.000000	80.0000	1	0
600	2	24.000000	27.0000	1	0
514	3	24.000000	7.4958	0	1
76	3	29.699118	7.8958	0	1
688	3	18.000000	7.7958	0	1
500	3	17.000000	8.6625	0	1

```
In [46]: y_test[:10]
```

```
Out[46]: 824    0
         593    0
         154    0
         786    1
         61     1
         600    1
         514    0
         76     0
         688    0
```

```
500    0
Name: Survived, dtype: int64
```

```
In [47]: model.predict(X_test[:10])
```

```
Out[47]: array([0, 1, 0, 1, 1, 1, 0, 0, 0, 0], dtype=int64)
```

```
In [48]: model.predict_proba(X_test[:10])
```

```
Out[48]: array([[0.96499346, 0.03500654],
                [0.0979403 , 0.9020597 ],
                [0.98765608, 0.01234392],
                [0.08073053, 0.91926947],
                [0.00194834, 0.99805166],
                [0.04108853, 0.95891147],
                [0.98669108, 0.01330892],
                [0.98770415, 0.01229585],
                [0.98485828, 0.01514172],
                [0.98452849, 0.01547151]])
```

Calculate the score using cross validation

```
In [49]: from sklearn.model_selection import cross_val_score
         cross_val_score(GaussianNB(),X_train, y_train, cv=5)
```

```
Out[49]: array([0.72727273, 0.76223776, 0.79577465, 0.8028169 , 0.75352113])
```