University of Haripur.

Assignment #1.

Machine Learning

Title: Naïve Bayes.

To: Ms. Nadia

From: Hamza Sadaat.

Depart: Information Technology.

Program: Computer Science.

Roll No: F18-0501.

Semester: 7th 'B'.

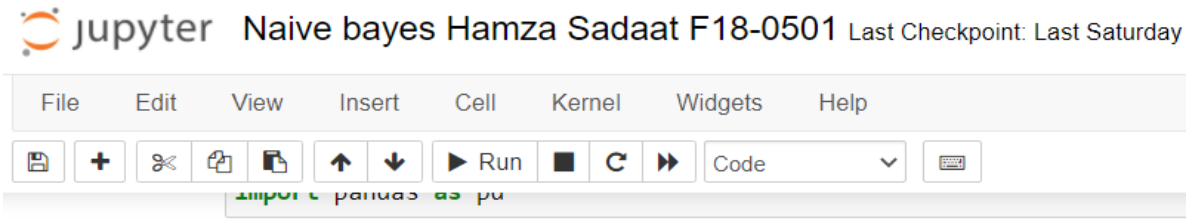
Dated: 15/11/2021.

1. **Recall the equation**

   Posterior Probabilities = Likelihood x Prior Probabilities

   P(A|B) = P(B|A) x P(A)
2. **Calculate Posterior Probability for each class. In your case the number of classes are 2.**
3. **Calculate Prior Probability for each class.**
4. **Calculate the likelihood.**
5. **The dataset given above is training dataset. Split the data into train and validation. For generation of code from scratch consider the last 4 instances (D11, D12,D13,D14) as your validation instances.**
6. **Once the code is generated and its giving accurate answers on validation data. Then test your model on these Unknown instances.**

Creating an NB classifier from scratch:



In [100]:
```
data = pd.DataFrame()
data['outlook'] = ['sunny','overcast','overcast','rain']
data['temperatue'] = ['mild','mild','hot','mild']
data['humidity'] = ['normal','high','normal','high']
data['wind'] = ['strong','strong','weak','strong']
data['play'] = ['yes','yes','yes','no']
data
```

Out[100]:

	outlook	temperatue	humidity	wind	play
0	sunny	mild	normal	strong	yes
1	overcast	mild	high	strong	yes
2	overcast	hot	normal	weak	yes
3	rain	mild	high	strong	no

In [101]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   outlook     4 non-null      object
 1   temperatue  4 non-null      object
 2   humidity    4 non-null      object
 3   wind        4 non-null      object
 4   play        4 non-null      object
dtypes: object(5)
```

Training and Testing Datasets:

jupyter Naive bayes Hamza Sadaat F18-0501 Last Checkpoint: Last Saturday at 10:44 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

In [104]:
```
X = data.drop(['play'],axis='columns')

y = data[['play']]
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)
```

In [105]:
```
X_train = pd.get_dummies(data[['outlook', 'temperatue', 'humidity', 'wind']])
y_train = pd.DataFrame(data['play'])

print(X_train.head())
```

```
   outlook_overcast  outlook_rain  outlook_sunny  temperatue_hot  \
0                 0             0              1               0
1                 1             0              0               0
2                 1             0              0               1
3                 0             1              0               0

   temperatue_mild  humidity_high  humidity_normal  wind_strong  wind_weak
0                1              0                1            1          0
1                1              1                0            1          0
2                0              0                1            0          1
3                1              1                0            1          0
```

In [106]:
```
model = GaussianNB()
model.fit(X_train, y_train)
```

jupyter Naive bayes Hamza Sadaat F18-0501 Last Checkpoint: Last Saturday at 10:44 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

```
2          0              1               0
3          1              1               0
4          0              0               1
```

In [23]:
```
ct = pd.crosstab(data['outlook'], data['play'], margins = True)
print(ct)
```

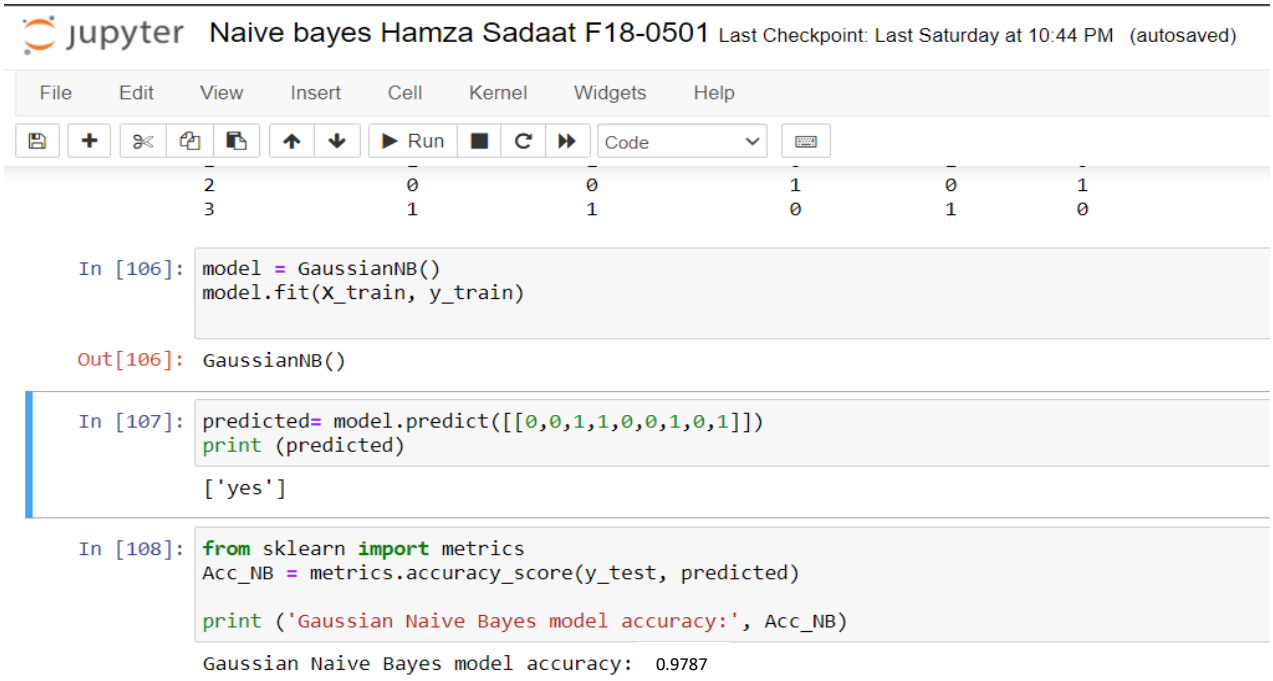
```
play      no  yes  All
outlook
overcast   0    4    4
rainy      2    3    5
sunny      3    2    5
All        5    9   14
```

In [24]:
```
def bayesposterior(prior, likelihood, evidence, string):
      print('Prior=', prior),
      print('Likelihood=', likelihood),
      print('Evidence=', evidence),
      print('Equation =','(Prior*Likelihood)/Evidence')
      print(string, (prior*likelihood)/evidence)
```

In [25]:
```
bayesposterior(prior = ct.iloc[1,1]/ct.iloc[3,1],
               likelihood = ct.iloc[3,1]/ct.iloc[3,2],
               evidence = ct.iloc[1,2]/ct.iloc[3,2],
               string = 'Probability of play tenis =')
```

```
Prior= 0.3333333333333333
Likelihood= 0.6428571428571429
Evidence= 0.35714285714285715
Equation = (Prior*Likelihood)/Evidence
Probability of play tenis = 0.6
```

Prediction with Naïve Bayes:



D15	**Sunny**	**Hot**	**Normal**	**Weak**	**Yes**
D16	**Rain**	**Hot**	**High**	**Strong**	**No**