Naive Bayes From Scratch

Kabir Nawani J043

Classifcation problem pertains to finding conditional probablity of some class labels given some data. Bayes Theorem provides a way to find this conditional probablity We could use a probablistic approach where model learns to map certain class labels, given some observation We use the MAP rule to select the label with largest probablity as the classification of the given instance

In [1]:

```
import numpy as np
import pandas as pd
from IPython.display import Image
from IPython.core.display import HTML
```

Wine dataset

The attributes include:

1. **Alcohol**
2. **Malic acid**
3. **Ash**
4. **Alcalinity of ash**
5. **Magnesium**
6. **Total phenols**
7. **Flavanoids**
8. **Nonflavanoid phenols**
9. **Proanthocyanins**
10. **Color intensity**
11. **Hue**
12. **OD280/OD315 of diluted wines**
13. **Proline**

In [2]:

```
columns=['Type','Alcohol','Malic acid','Ash','Alcalinity','Magnesium','Phenols','Flavano
ids','Nonfav','Proanthocyanins','Intensity','Hue','Diluted','Proline']
data=pd.read_csv('wine.csv',names=columns)
```

In [3]:

```
data.head()
```

Out[3]:

	Type	Alcohol	Malic acid	Ash	Alcalinity	Magnesium	Phenols	Flavanoids	Nonfav	Proanthocyanins	Intensity	Hue	Diluted	P
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	

Calculating priors

We have three classes of wine

- **1st class**
- **2nd class**
- **3rd class**

In [32]:

```
#Number of outcomes for class 1
n_outcome1= data['Type'][data['Type']==1].count()

#Number of outcomes for class 2
n_outcome2= data['Type'][data['Type']==2].count()

#Number of outcomes for class 3
n_outcome3= data['Type'][data['Type']==3].count()

#total count
tot_outcomes  = data['Type'].count()
```

In [33]:

```
#Number of outcomes of type1
P_type1= n_outcome1/tot_outcomes

#Number of outcomes of type2
P_type2= n_outcome2/tot_outcomes

#Number of outcomes of type3
P_type3= n_outcome3/tot_outcomes
```

Calculating likelyhood for each feature

In [34]:

```
#Calculating the mean and variance

data_means= data.groupby('Type').mean()
data_means

data_variance=data.groupby('Type').var()
data_variance
```

Out[34]:

	Alcohol	Malic acid	Ash	Alcalinity	Magnesium	Phenols	Flavanoids	Nonfav	Proanthocyanins	Intensity	H
Type											
1	0.213560	0.474100	0.051604	6.483758	110.227937	0.114895	0.158001	0.004907	0.169834	1.534063	0.013
2	0.289406	1.031380	0.099520	11.220962	280.679678	0.297419	0.498014	0.015366	0.362486	0.855494	0.041
3	0.281156	1.183539	0.034110	5.099291	118.602394	0.127428	0.086145	0.015411	0.167147	5.340454	0.013

Assigning the means and variance to variables

In [49]:

```
#mean for class 1
Type1_Alc_mean= data_means['Alcohol'][data_means.index==1].values[0]
Type1_Mal_mean= data_means['Malic acid'][data_means.index==1].values[0]
Type1_Ash_mean= data_means['Ash'][data_means.index==1].values[0]
Type1_Alcan_mean= data_means['Alcalinity'][data_means.index==1].values[0]
```

```python
Type1_Mg_mean= data_means['Magnesium'][data_means.index==1].values[0]
Type1_Ph_mean= data_means['Phenols'][data_means.index==1].values[0]
Type1_Flav_mean= data_means['Flavanoids'][data_means.index==1].values[0]
Type1_Nonflav_mean= data_means['Nonfav'][data_means.index==1].values[0]
Type1_Pro_mean= data_means['Proanthocyanins'][data_means.index==1].values[0]
Type1_Intensity_mean= data_means['Intensity'][data_means.index==1].values[0]
Type1_Hue_mean= data_means['Hue'][data_means.index==1].values[0]
Type1_Diluted_mean= data_means['Diluted'][data_means.index==1].values[0]
Type1_Proline_mean= data_means['Proline'][data_means.index==1].values[0]

#variance for class 1
Type1_Alc_var= data_variance['Alcohol'][data_means.index==1].values[0]
Type1_Mal_var= data_variance['Malic acid'][data_means.index==1].values[0]
Type1_Ash_var= data_variance['Ash'][data_means.index==1].values[0]
Type1_Alcan_var= data_variance['Alcalinity'][data_means.index==1].values[0]
Type1_Mg_var= data_variance['Magnesium'][data_means.index==1].values[0]
Type1_Ph_var= data_variance['Phenols'][data_means.index==1].values[0]
Type1_Flav_var= data_variance['Flavanoids'][data_means.index==1].values[0]
Type1_Nonflav_var= data_variance['Nonfav'][data_means.index==1].values[0]
Type1_Pro_var= data_variance['Proanthocyanins'][data_means.index==1].values[0]
Type1_Intensity_var= data_variance['Intensity'][data_means.index==1].values[0]
Type1_Hue_var= data_variance['Hue'][data_means.index==1].values[0]
Type1_Diluted_var= data_variance['Diluted'][data_means.index==1].values[0]
Type1_Proline_var= data_variance['Proline'][data_means.index==1].values[0]

#mean for class 2
Type2_Alc_mean= data_means['Alcohol'][data_means.index==2].values[0]
Type2_Mal_mean= data_means['Malic acid'][data_means.index==2].values[0]
Type2_Ash_mean= data_means['Ash'][data_means.index==2].values[0]
Type2_Alcan_mean= data_means['Alcalinity'][data_means.index==2].values[0]
Type2_Mg_mean= data_means['Magnesium'][data_means.index==2].values[0]
Type2_Ph_mean= data_means['Phenols'][data_means.index==2].values[0]
Type2_Flav_mean= data_means['Flavanoids'][data_means.index==2].values[0]
Type2_Nonflav_mean= data_means['Nonfav'][data_means.index==2].values[0]
Type2_Pro_mean= data_means['Proanthocyanins'][data_means.index==2].values[0]
Type2_Intensity_mean= data_means['Intensity'][data_means.index==2].values[0]
Type2_Hue_mean= data_means['Hue'][data_means.index==2].values[0]
Type2_Diluted_mean= data_means['Diluted'][data_means.index==2].values[0]
Type2_Proline_mean= data_means['Proline'][data_means.index==2].values[0]

Type2_Alc_var= data_variance['Alcohol'][data_means.index==2].values[0]
Type2_Mal_var= data_variance['Malic acid'][data_means.index==2].values[0]
Type2_Ash_var= data_variance['Ash'][data_means.index==2].values[0]
Type2_Alcan_var= data_variance['Alcalinity'][data_means.index==2].values[0]
Type2_Mg_var= data_variance['Magnesium'][data_means.index==2].values[0]
Type2_Ph_var= data_variance['Phenols'][data_means.index==2].values[0]
Type2_Flav_var= data_variance['Flavanoids'][data_means.index==2].values[0]
Type2_Nonflav_var= data_variance['Nonfav'][data_means.index==2].values[0]
Type2_Pro_var= data_variance['Proanthocyanins'][data_means.index==2].values[0]
Type2_Intensity_var= data_variance['Intensity'][data_means.index==2].values[0]
Type2_Hue_var= data_variance['Hue'][data_means.index==2].values[0]
Type2_Diluted_var= data_variance['Diluted'][data_means.index==2].values[0]
Type2_Proline_var= data_variance['Proline'][data_means.index==2].values[0]

#mean for class 3
Type3_Alc_mean= data_means['Alcohol'][data_means.index==3].values[0]
Type3_Mal_mean= data_means['Malic acid'][data_means.index==3].values[0]
Type3_Ash_mean= data_means['Ash'][data_means.index==3].values[0]
Type3_Alcan_mean= data_means['Alcalinity'][data_means.index==3].values[0]
Type3_Mg_mean= data_means['Magnesium'][data_means.index==3].values[0]
Type3_Ph_mean= data_means['Phenols'][data_means.index==3].values[0]
Type3_Flav_mean= data_means['Flavanoids'][data_means.index==3].values[0]
Type3_Nonflav_mean= data_means['Nonfav'][data_means.index==3].values[0]
Type3_Pro_mean= data_means['Proanthocyanins'][data_means.index==3].values[0]
Type3_Intensity_mean= data_means['Intensity'][data_means.index==3].values[0]
Type3_Hue_mean= data_means['Hue'][data_means.index==3].values[0]
Type3_Diluted_mean= data_means['Diluted'][data_means.index==3].values[0]
Type3_Proline_mean= data_means['Proline'][data_means.index==3].values[0]

#variance for class 3
Type3_Alc_var= data_variance['Alcohol'][data_means.index==3].values[0]
Type3_Mal_var= data_variance['Malic acid'][data_means.index==3].values[0]
```

```
Type3_Ash_var= data_variance['Ash'][data_means.index==3].values[0]
Type3_Alcan_var= data_variance['Alcalinity'][data_means.index==3].values[0]
Type3_Mg_var= data_variance['Magnesium'][data_means.index==3].values[0]
Type3_Ph_var= data_variance['Phenols'][data_means.index==3].values[0]
Type3_Flav_var= data_variance['Flavanoids'][data_means.index==3].values[0]
Type3_Nonflav_var= data_variance['Nonfav'][data_means.index==3].values[0]
Type3_Pro_var= data_variance['Proanthocyanins'][data_means.index==3].values[0]
Type3_Intensity_var= data_variance['Intensity'][data_means.index==3].values[0]
Type3_Hue_var= data_variance['Hue'][data_means.index==3].values[0]
Type3_Diluted_var= data_variance['Diluted'][data_means.index==3].values[0]
Type3_Proline_var= data_variance['Proline'][data_means.index==3].values[0]
```

Test data

In [54]:

```
#creating empty dataframe for prediction
wine= pd.DataFrame()

#creating a feature for a single row
wine['Alcohol']= [13.64]
wine['Malic acid']= [3.1]
wine['Ash']=[2.56]
wine['Alcalinity']= [15.2]
wine['Magnesium']= [116]
wine['Phenols']= [2.7]
wine['Flavanoids']= [3.03]
wine['Nonfav']= [0.17]
wine['Proanthocyanins']= [1.66]
wine['Intensity']= [5.1]
wine['Hue']= [0.96]
wine['Diluted']= [3.36]
wine['Proline']= [845]

wine
```

Out[54]:

	Alcohol	Malic acid	Ash	Alcalinity	Magnesium	Phenols	Flavanoids	Nonfav	Proanthocyanins	Intensity	Hue	Diluted	Proline
0	13.64	3.1	2.56	15.2	116	2.7	3.03	0.17	1.66	5.1	0.96	3.36	845

In [51]:

```
#Create a function to calc the P(x\y)
def p_x_given_y(x, y_mean,y_var):

    #Using probablity density fucntion
    p= 1/(np.sqrt(2*np.pi*y_var))* np.exp((-(x-y_mean)**2)/(2*y_var))

    return p
```

In [55]:

```
out1= P_type1 * \
p_x_given_y(wine['Alcohol'][0], Type1_Alc_mean,Type1_Alc_var) *\
p_x_given_y(wine['Malic acid'][0], Type1_Mal_mean,Type1_Mal_var) *\
p_x_given_y(wine['Ash'][0], Type1_Alc_mean,Type1_Ash_var) *\
p_x_given_y(wine['Alcalinity'][0], Type1_Alcan_mean,Type1_Alcan_var) *\
p_x_given_y(wine['Magnesium'][0], Type1_Mg_mean,Type1_Mg_var) *\
p_x_given_y(wine['Phenols'][0], Type1_Ph_mean,Type1_Alc_var) *\
p_x_given_y(wine['Flavanoids'][0], Type1_Flav_mean,Type1_Flav_var) *\
p_x_given_y(wine['Nonfav'][0], Type1_Nonflav_mean,Type1_Nonflav_var) *\
p_x_given_y(wine['Proanthocyanins'][0], Type1_Pro_mean,Type1_Pro_var) *\
p_x_given_y(wine['Intensity'][0], Type1_Intensity_mean,Type1_Intensity_var) *\
p_x_given_y(wine['Hue'][0], Type1_Hue_mean,Type1_Hue_var) *\
p_x_given_y(wine['Diluted'][0], Type1_Diluted_mean,Type1_Diluted_var) *\
p_x_given_y(wine['Proline'][0], Type1_Proline_mean,Type1_Proline_var)

out2= P_type2 * \
p_x_given_y(wine['Alcohol'][0], Type2_Alc_mean,Type2_Alc_var) *\
p_x_given_y(wine['Malic acid'][0], Type2_Mal_mean,Type2_Mal_var) *\
p_x_given_y(wine['Ash'][0], Type2_Alc_mean,Type2_Ash_var) *\
p_x_given_y(wine['Alcalinity'][0], Type2_Alcan_mean,Type2_Alcan_var) *\
p_x_given_y(wine['Magnesium'][0], Type2_Mg_mean,Type2_Mg_var) *\
p_x_given_y(wine['Phenols'][0], Type2_Ph_mean,Type2_Alc_var) *\
p_x_given_y(wine['Flavanoids'][0], Type2_Flav_mean,Type2_Flav_var) *\
p_x_given_y(wine['Nonfav'][0], Type2_Nonflav_mean,Type2_Nonflav_var) *\
p_x_given_y(wine['Proanthocyanins'][0], Type2_Pro_mean,Type2_Pro_var) *\
p_x_given_y(wine['Intensity'][0], Type2_Intensity_mean,Type2_Intensity_var) *\
p_x_given_y(wine['Hue'][0], Type2_Hue_mean,Type2_Hue_var) *\
p_x_given_y(wine['Diluted'][0], Type2_Diluted_mean,Type2_Diluted_var) *\
p_x_given_y(wine['Proline'][0], Type2_Proline_mean,Type2_Proline_var)


out3= P_type3 * \
p_x_given_y(wine['Alcohol'][0], Type3_Alc_mean,Type3_Alc_var) *\
p_x_given_y(wine['Malic acid'][0], Type3_Mal_mean,Type3_Mal_var) *\
p_x_given_y(wine['Ash'][0], Type3_Alc_mean,Type3_Ash_var) *\
p_x_given_y(wine['Alcalinity'][0], Type3_Alcan_mean,Type3_Alcan_var) *\
p_x_given_y(wine['Magnesium'][0], Type3_Mg_mean,Type3_Mg_var) *\
p_x_given_y(wine['Phenols'][0], Type3_Ph_mean,Type3_Alc_var) *\
p_x_given_y(wine['Flavanoids'][0], Type3_Flav_mean,Type3_Flav_var) *\
p_x_given_y(wine['Nonfav'][0], Type3_Nonflav_mean,Type3_Nonflav_var) *\
p_x_given_y(wine['Proanthocyanins'][0], Type3_Pro_mean,Type3_Pro_var) *\
p_x_given_y(wine['Intensity'][0], Type3_Intensity_mean,Type3_Intensity_var) *\
p_x_given_y(wine['Hue'][0], Type3_Hue_mean,Type3_Hue_var) *\
p_x_given_y(wine['Diluted'][0], Type3_Diluted_mean,Type3_Diluted_var) *\
p_x_given_y(wine['Proline'][0], Type3_Proline_mean,Type3_Proline_var)
```

Final prediction

In [56]:

```
if(out1<out2):
    if(out2<out3):
        print('It is type 3')
    else:
        print('It is type 2')
else:
    print('It is type 1')
```

```
It is type 2
```