Naive Bayes DS

February 20, 2021

[2]:
```
import pandas as pd
```

[3]:
```
da = pd.read_csv("C:\MTech Data Science\Mini Project\diabetes_data_upload.csv")
```

[4]:
```
da[0:5]
```

[4]:
```
   Age Gender Polyuria Polydipsia sudden weight loss weakness Polyphagia  \
0   40   Male       No        Yes                 No      Yes         No
1   58   Male       No         No                 No      Yes         No
2   41   Male      Yes         No                 No      Yes        Yes
3   45   Male       No         No                Yes      Yes        Yes
4   60   Male      Yes        Yes                Yes      Yes        Yes

  Genital thrush visual blurring Itching Irritability delayed healing  \
0             No              No     Yes           No             Yes
1             No             Yes      No           No              No
2             No              No     Yes           No             Yes
3            Yes              No     Yes           No             Yes
4             No             Yes     Yes          Yes             Yes

  partial paresis muscle stiffness Alopecia Obesity     class
0              No              Yes      Yes     Yes  Positive
1             Yes               No      Yes      No  Positive
2              No              Yes      Yes      No  Positive
3              No               No       No      No  Positive
4             Yes              Yes      Yes     Yes  Positive
```

[5]:
```
data = {'Gender':['Male','Female'], 'Positive':[int(0), int(0)],'Negative':
 ↪[int(0), int(0)]}
df = pd.DataFrame(data, columns=['Gender', 'Positive', 'Negative'])
df
```

[5]:
```
   Gender  Positive  Negative
0    Male         0         0
1  Female         0         0
```

[6]:
```
for i in range(len(da)):
    if da.iloc[i,1] == 'Male':
        if da.iloc[i,-1] == 'Positive':
            df.iloc[0,1] = df.iloc[0,1] + 1
        else:
            df.iloc[0,2] = df.iloc[0,2] + 1
    elif da.iloc[i,1] == 'Female':
        if da.iloc[i,-1] == 'Positive':
            df.iloc[1,1] = df.iloc[1,1] + 1
        else:
            df.iloc[1,2] = df.iloc[1,2] + 1
    else:
        pass
```

[7]:

```
#Frequency table
df
```

[7]:

```
   Gender  Positive  Negative
0    Male       147       181
1  Female       173        19
```

[8]:

```
p = df.iloc[:,1].sum()
n = df.iloc[:,2].sum()
```

[9]:

```
len(df.index)
```

[9]:

```
2
```

[10]:

```
df.loc[len(df.index)] = ['Total', p, n]
```

[11]:

```
#Total Count
df
```

[11]:

```
   Gender  Positive  Negative
0    Male       147       181
1  Female       173        19
2   Total       320       200
```

[12]:

```
s = df.iloc[2,1:].sum()
display(s)
```

```
520
```

[13]:

```
#Build Likelihood table 1
a = {'Gender':['Male','Female','Total'], 'Positive':[float(df.iloc[0,1] / s),␣
 ↪float(df.iloc[1,1] / s), float(df.iloc[2,1] / s)],'Negative':[float(df.
 ↪iloc[0,2] / s), float(df.iloc[1,2] / s), float(df.iloc[2,2] / s)]}
df1 = pd.DataFrame(a, columns=['Gender', 'Positive', 'Negative'])
df1
```

[13]:
```
   Gender  Positive  Negative
0    Male  0.282692  0.348077
1  Female  0.332692  0.036538
2   Total  0.615385  0.384615
```

[14]:
```
#Build Likelihood table 2
b = {'Gender':['Male','Female','Total'], 'Positive':[float(df.iloc[0,1]/df.
 ↪iloc[2,1]), float(df.iloc[1,1]/df.iloc[2,1]),df.iloc[2,1]],'Negative':
 ↪[float(df.iloc[0,2]/df.iloc[2,2]), float(df.iloc[1,2]/df.iloc[2,2]),df.
 ↪iloc[2,2]]}
df2 = pd.DataFrame(b, columns=['Gender', 'Positive', 'Negative'])
df2
```

[14]:
```
   Gender    Positive  Negative
0    Male    0.459375     0.905
1  Female    0.540625     0.095
2   Total  320.000000   200.000
```

[15]:
```
#P(Positive|Male) = P(Male|Positive) * P(Positive) / P(Male)
prob_pos_men = float(df2.iloc[0,1] * df1.iloc[2,1] / (df1.iloc[0,1] + df1.
 ↪iloc[0,2]))
display(prob_pos_men)
```

```
0.4481707317073171
```

[16]:
```
#P(Positive|Female) = P(Female|Positive) * P(Positive) / P(Female)
prob_pos_female = float(df2.iloc[1,1] * df1.iloc[2,1] / (df1.iloc[1,1] + df1.
 ↪iloc[1,2]))
display(prob_pos_female)
```

```
0.9010416666666669
```

[17]:
```
#Thus it proves that the likelihood of "Females" having diabetes is high
```

[18]:
```
d= {'Age Group':['<=30','31-40','41-50','51-60','61-70','71-80','>80'],␣
 ↪'Positive':[0,0,0,0,0,0,0], 'Negative':[0,0,0,0,0,0,0]}
df3 = pd.DataFrame(d, columns=['Age Group','Positive','Negative'])
```

[19]:
```
for j in range(len(da)):
    if da.iloc[j,0] <= 30:
        if da.iloc[j,-1] == 'Positive':
            df3.iloc[0,1] = df3.iloc[0,1] + 1
        else:
            df3.iloc[0,2] = df3.iloc[0,2] + 1
    elif da.iloc[j,0] >30 and da.iloc[j,0] <=40:
        if da.iloc[j,-1] == 'Positive':
            df3.iloc[1,1] = df3.iloc[1,1] + 1
        else:
            df3.iloc[1,2] = df3.iloc[1,2] + 1
    elif da.iloc[j,0] >40 and da.iloc[j,0] <=50:
        if da.iloc[j,-1] == 'Positive':
            df3.iloc[2,1] = df3.iloc[2,1] + 1
        else:
            df3.iloc[2,2] = df3.iloc[2,2] + 1
    elif da.iloc[j,0] >50 and da.iloc[j,0] <=60:
        if da.iloc[j,-1] == 'Positive':
            df3.iloc[3,1] = df3.iloc[3,1] + 1
        else:
            df3.iloc[3,2] = df3.iloc[3,2] + 1
    elif da.iloc[j,0] >60 and da.iloc[j,0] <=70:
        if da.iloc[j,-1] == 'Positive':
            df3.iloc[4,1] = df3.iloc[4,1] + 1
        else:
            df3.iloc[4,2] = df3.iloc[4,2] + 1
    elif da.iloc[j,0] >70 and da.iloc[j,0] <=80:
        if da.iloc[j,-1] == 'Positive':
            df3.iloc[5,1] = df3.iloc[5,1] + 1
        else:
            df3.iloc[5,2] = df3.iloc[5,2] + 1
    else:
        if da.iloc[j,-1] == 'Positive':
            df3.iloc[6,1] = df3.iloc[6,1] + 1
        else:
            df3.iloc[6,2] = df3.iloc[6,2] + 1
```

[20]:

```
display(df3)
```

```
  Age Group  Positive  Negative
0      <=30        15        30
1     31-40        84        39
2     41-50        87        58
3     51-60        78        49
4     61-70        49        17
5     71-80         3         7
6       >80         4         0
```

[21]:

```
p1 = df3.iloc[:,1].sum()
n1 = df3.iloc[:,2].sum()
display(p1, n1)
```

```
320
```

```
200
```

```
[22]: df2
```

```
[22]:    Gender    Positive  Negative
      0    Male    0.459375     0.905
      1  Female    0.540625     0.095
      2   Total  320.000000   200.000
```

```
[23]: d1= {'Age Group':['<=30','31-40','41-50','51-60','61-70','71-80','>80'],
          'Positive':[float(df3.iloc[0,1]/p1), float(df3.iloc[1,1]/p1), float(df3.
       ↪iloc[2,1]/p1), float(df3.iloc[3,1]/p1), float(df3.iloc[4,1]/p1), float(df3.
       ↪iloc[5,1]/p1), float(df3.iloc[6,1]/p1)],
          'Negative':[float(df3.iloc[0,2]/n1), float(df3.iloc[1,2]/n1), float(df3.
       ↪iloc[2,2]/n1), float(df3.iloc[3,2]/n1), float(df3.iloc[4,2]/n1), float(df3.
       ↪iloc[5,2]/n1), float(df3.iloc[6,2]/n1)]}
      df4 = pd.DataFrame(d1, columns=['Age Group','Positive','Negative'])
```

```
[24]: df4
```

```
[24]:   Age Group  Positive  Negative
      0      <=30  0.046875     0.150
      1     31-40  0.262500     0.195
      2     41-50  0.271875     0.290
      3     51-60  0.243750     0.245
      4     61-70  0.153125     0.085
      5     71-80  0.009375     0.035
      6       >80  0.012500     0.000
```

```
[27]: #t = <Positive, Female, 43>
      P_t_positive = df1.iloc[1, 1] * df4.iloc[2, 1]
      display(P_t_positive)
```

```
0.09045072115384614
```

```
[28]: P_t_intersection_positive = P_t_positive * df1.iloc[2, 1]
      P_t_intersection_positive
```

```
[28]: 0.0556619822485207
```

```
[29]: P_t_negative = df1.iloc[1, 2] * df4.iloc[2, 2]
      display(P_t_negative)
```

```
0.010596153846153845
```

```
[30]: P_t_intersection_negative = P_t_negative * df1.iloc[2, 2]
      P_t_intersection_negative
```

[30]:

```
0.004075443786982248
```

[31]:

```
P_t = P_t_intersection_positive + P_t_intersection_negative
display(P_t)
```

```
0.05973742603550295
```

[32]:

```
#P(Positive/t)
res = float(P_t_intersection_positive/P_t)
```

[33]:

```
display(res)
```

```
0.9317773788150808
```

[31]:

```
#Upon manual inspection it was found that female with the age 43 has a positive␣
 ↪class
```

[34]:

```
display(df4) #Probability of the age group being positive or negative
```

```
  Age Group  Positive  Negative
0      <=30  0.046875     0.150
1     31-40  0.262500     0.195
2     41-50  0.271875     0.290
3     51-60  0.243750     0.245
4     61-70  0.153125     0.085
5     71-80  0.009375     0.035
6       >80  0.012500     0.000
```

[35]:

```
display(df1) #Probability of the age group being positive or negative
```

```
   Gender  Positive  Negative
0    Male  0.282692  0.348077
1  Female  0.332692  0.036538
2   Total  0.615385  0.384615
```

[36]:

```
display(df2)
```

```
   Gender    Positive  Negative
0    Male    0.459375     0.905
1  Female    0.540625     0.095
2   Total  320.000000   200.000
```

[38]:

```
#First 100 rows and 2 columns: Age and Gender of the given dataset were␣
 ↪considered for the test dataset
da_test = pd.read_excel("C:\MTech Data Science\Mini Project\Diabetes_Test.xlsx")
```

[39]:

```
da_test[0:5]
```

[39]:

```
   Age Gender  class
0   40   Male      0
1   58   Male      0
2   41   Male      0
3   45   Male      0
4   60   Male      0
```

[40]:

```
#Built a Naive Bayes classifier for 2 input features: Age and Gender based on the frequency tables: df1, df2, df4
p_t_pos = float(0.0)
p_t_neg = float(0.0)
for i in range(len(da_test)):
    #Checking the person's gender whether male or female
    if da_test.iloc[i,1] == 'Male':
        #If age is less that or equal to 30
        if da_test.iloc[i,0] <=30:
            p_t_pos = df2.iloc[0,1] * df4.iloc[0,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[0,2] * df4.iloc[0,2] * df1.iloc[2,2]

        #If Age: 31 - 40
        elif da_test.iloc[i,0] >30 and da_test.iloc[i,0] <=40:
            p_t_pos = df2.iloc[0,1] * df4.iloc[1,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[0,2] * df4.iloc[1,2] * df1.iloc[2,2]

        #If Age: 41 - 50
        elif da_test.iloc[i,0] >40 and da_test.iloc[i,0] <=50:
            p_t_pos = df2.iloc[0,1] * df4.iloc[2,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[0,2] * df4.iloc[2,2] * df1.iloc[2,2]

        #If Age: 51 - 60
        elif da_test.iloc[i,0] >50 and da_test.iloc[i,0] <=60:
            p_t_pos = df2.iloc[0,1] * df4.iloc[3,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[0,2] * df4.iloc[3,2] * df1.iloc[2,2]

        #If Age: 61 - 70
        elif da_test.iloc[i,0] >60 and da_test.iloc[i,0] <=70:
            p_t_pos = df2.iloc[0,1] * df4.iloc[4,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[0,2] * df4.iloc[4,2] * df1.iloc[2,2]

        #If Age: 71 - 80
        elif da_test.iloc[i,0] >70 and da_test.iloc[i,0] <=80:
            p_t_pos = df2.iloc[0,1] * df4.iloc[5,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[0,2] * df4.iloc[5,2] * df1.iloc[2,2]

        #If Age > 80
        else:
            p_t_pos = df2.iloc[0,1] * df4.iloc[6,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[0,2] * df4.iloc[6,2] * df1.iloc[2,2]

        #Decision for Positive or Negative for diabetes
        if p_t_pos >= p_t_neg:
            da_test.iloc[i,2] = 'Positive'
        else:
            da_test.iloc[i,2] = 'Negative'

    else:
        #If age is less that or equal to 30
        if da_test.iloc[i,0] <=30:
            p_t_pos = df2.iloc[1,1] * df4.iloc[0,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[1,2] * df4.iloc[0,2] * df1.iloc[2,2]

        #If Age: 31 - 40
        elif da_test.iloc[i,0] >30 and da_test.iloc[i,0] <=40:
            p_t_pos = df2.iloc[1,1] * df4.iloc[1,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[1,2] * df4.iloc[1,2] * df1.iloc[2,2]

        #If Age: 41 - 50
        elif da_test.iloc[i,0] >40 and da_test.iloc[i,0] <=50:
            p_t_pos = df2.iloc[1,1] * df4.iloc[2,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[1,2] * df4.iloc[2,2] * df1.iloc[2,2]

        #If Age: 51 - 60
        elif da_test.iloc[i,0] >50 and da_test.iloc[i,0] <=60:
            p_t_pos = df2.iloc[1,1] * df4.iloc[3,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[1,2] * df4.iloc[3,2] * df1.iloc[2,2]

        #If Age: 61 - 70
        elif da_test.iloc[i,0] >60 and da_test.iloc[i,0] <=70:
            p_t_pos = df2.iloc[1,1] * df4.iloc[4,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[1,2] * df4.iloc[4,2] * df1.iloc[2,2]

        #If Age: 71 - 80
        elif da_test.iloc[i,0] >70 and da_test.iloc[i,0] <=80:
            p_t_pos = df2.iloc[1,1] * df4.iloc[5,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[1,2] * df4.iloc[5,2] * df1.iloc[2,2]

        #If Age > 80
        else:
            p_t_pos = df2.iloc[1,1] * df4.iloc[6,1] * df1.iloc[2,1]
            p_t_neg = df2.iloc[1,2] * df4.iloc[6,2] * df1.iloc[2,2]

        #Decision for Positive or Negative for diabetes
        if p_t_pos >= p_t_neg:
            da_test.iloc[i,2] = 'Positive'
        else:
            da_test.iloc[i,2] = 'Negative'
```

[41]:

```
da_final =da_test
```

[42]:

```
da_final[0:5]
```

[42]:

```
   Age Gender     class
0   40   Male  Positive
1   58   Male  Negative
2   41   Male  Negative
3   45   Male  Negative
4   60   Male  Negative
```

[43]:

```
da_final.to_excel('C:\MTech Data Science\Mini Project\Diabetes_Final.xlsx',␣
 ↪sheet_name='Sheet1')
```

[44]:

```
#Upon manual inspection we found 77 target values were correct out of 100:␣
 ↪Hence accuracy:(0.77) 77% when 2 input features
#considered
```

[46]:

```
data1 = {'Polyuria':['Yes','No'], 'Positive':[int(0), int(0)],'Negative':
 ↪[int(0), int(0)]}
df5 = pd.DataFrame(data1, columns=['Polyuria', 'Positive', 'Negative'])
```

[47]:

```
for i in range(len(da)):
    if da.iloc[i,2] == 'Yes':
        if da.iloc[i,-1] == 'Positive':
            df5.iloc[0,1] = df5.iloc[0,1] + 1
        else:
            df5.iloc[0,2] = df5.iloc[0,2] + 1
    elif da.iloc[i,2] == 'No':
        if da.iloc[i,-1] == 'Positive':
            df5.iloc[1,1] = df5.iloc[1,1] + 1
        else:
            df5.iloc[1,2] = df5.iloc[1,2] + 1
    else:
        pass
```

[48]:

```
display(df5) #Probability that a person has polyuria given Positive/Negative
```

```
  Polyuria  Positive  Negative
0      Yes       243        15
1       No        77       185
```

[49]:

```
p2 = df5.iloc[:,1].sum()
n2 = df5.iloc[:,2].sum()
display(p2, n2)
```

```
320
```

```
200
```

[50]:

```
df5.iloc[0,1] = float(df5.iloc[0,1]/p2)
df5.iloc[1,1] = float(df5.iloc[1,1]/p2)
df5.iloc[0,2] = float(df5.iloc[0,2]/n2)
df5.iloc[1,2] = float(df5.iloc[1,2]/n2)
```

[51]:

```
#Calculated probabilities for the third input feature Polyuria
display(df5)
```

```
  Polyuria  Positive  Negative
0      Yes  0.759375     0.075
1       No  0.240625     0.925
```

[52]:

```
#First 100 rows and 3 columns: Age,Gender and Polyuria of the given dataset␣
 ↪were considered for the test dataset
da_test1 = pd.read_excel("C:\MTech Data Science\Mini Project\Diabetes_Test1.
 ↪xlsx")
```

[53]:

```
da_test1[0:5]
```

[53]:

```
   Age Gender Polyuria  class
0   40   Male       No      0
1   58   Male       No      0
2   41   Male      Yes      0
3   45   Male       No      0
4   60   Male      Yes      0
```

[55]:

```
#Built a Naive Bayes classifier for 3 input features: Age,Gender and Polyuria␣
 ↪based on the frequency tables: df1, df2, df4, df5
p_pos = float(0.0)
p_neg = float(0.0)
for i in range(len(da_test1)):
    #Checking the person's gender whether male or female
    if da_test1.iloc[i,1] == 'Male':

        #If Age <=30
        if da_test1.iloc[i,0] <=30:
            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[0,1] * df4.iloc[0,1] * df1.iloc[2,1] * df5.
↪iloc[0,1]
                p_neg = df2.iloc[0,2] * df4.iloc[0,2] * df1.iloc[2,2] * df5.
↪iloc[0,2]
            else:
                p_pos = df2.iloc[0,1] * df4.iloc[0,1] * df1.iloc[2,1] * df5.
↪iloc[1,1]
                p_neg = df2.iloc[0,2] * df4.iloc[0,2] * df1.iloc[2,2] * df5.
↪iloc[1,2]

        #If Age 31 - 40
        elif da_test1.iloc[i,0] >30 and da_test.iloc[i,0] <=40:

            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[0,1] * df4.iloc[1,1] * df1.iloc[2,1] * df5.
↪iloc[0,1]
                p_neg = df2.iloc[0,2] * df4.iloc[1,2] * df1.iloc[2,2] * df5.
↪iloc[0,2]
            else:
                p_pos = df2.iloc[0,1] * df4.iloc[1,1] * df1.iloc[2,1] * df5.
↪iloc[1,1]
                p_neg = df2.iloc[0,2] * df4.iloc[1,2] * df1.iloc[2,2] * df5.
↪iloc[1,2]

        #If Age 41 - 50
        elif da_test1.iloc[i,0] >40 and da_test1.iloc[i,0] <=50:

            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[0,1] * df4.iloc[2,1] * df1.iloc[2,1] * df5.
↪iloc[0,1]
                p_neg = df2.iloc[0,2] * df4.iloc[2,2] * df1.iloc[2,2] * df5.
↪iloc[0,2]
            else:
                p_pos = df2.iloc[0,1] * df4.iloc[2,1] * df1.iloc[2,1] * df5.
↪iloc[1,1]
                p_neg = df2.iloc[0,2] * df4.iloc[2,2] * df1.iloc[2,2] * df5.
↪iloc[1,2]

        #If Age 51 - 60
        elif da_test1.iloc[i,0] >50 and da_test1.iloc[i,0] <=60:

            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[0,1] * df4.iloc[3,1] * df1.iloc[2,1] * df5.
↪iloc[0,1]
                p_neg = df2.iloc[0,2] * df4.iloc[3,2] * df1.iloc[2,2] * df5.
↪iloc[0,2]
            else:
                p_pos = df2.iloc[0,1] * df4.iloc[3,1] * df1.iloc[2,1] * df5.
↪iloc[1,1]
                p_neg = df2.iloc[0,2] * df4.iloc[3,2] * df1.iloc[2,2] * df5.
↪iloc[1,2]

        #If Age 61 - 70
        elif da_test1.iloc[i,0] >60 and da_test1.iloc[i,0] <=70:

            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[0,1] * df4.iloc[4,1] * df1.iloc[2,1] * df5.
↪iloc[0,1]
                p_neg = df2.iloc[0,2] * df4.iloc[4,2] * df1.iloc[2,2] * df5.
↪iloc[0,2]
            else:
                p_pos = df2.iloc[0,1] * df4.iloc[4,1] * df1.iloc[2,1] * df5.
↪iloc[1,1]
                p_neg = df2.iloc[0,2] * df4.iloc[4,2] * df1.iloc[2,2] * df5.
↪iloc[1,2]

        #If Age 71 - 80
        elif da_test1.iloc[i,0] >70 and da_test1.iloc[i,0] <=80:

            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[0,1] * df4.iloc[5,1] * df1.iloc[2,1] * df5.
↪iloc[0,1]
                p_neg = df2.iloc[0,2] * df4.iloc[5,2] * df1.iloc[2,2] * df5.
↪iloc[0,2]
            else:
                p_pos = df2.iloc[0,1] * df4.iloc[5,1] * df1.iloc[2,1] * df5.
↪iloc[1,1]
                p_neg = df2.iloc[0,2] * df4.iloc[5,2] * df1.iloc[2,2] * df5.
↪iloc[1,2]

        #If Age >= 80
        else:

            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[0,1] * df4.iloc[6,1] * df1.iloc[2,1] * df5.
↪iloc[0,1]
                p_neg = df2.iloc[0,2] * df4.iloc[6,2] * df1.iloc[2,2] * df5.
↪iloc[0,2]
            else:
                p_pos = df2.iloc[0,1] * df4.iloc[6,1] * df1.iloc[2,1] * df5.
↪iloc[1,1]
                p_neg = df2.iloc[0,2] * df4.iloc[6,2] * df1.iloc[2,2] * df5.
↪iloc[1,2]

        #Decision for Positive or Negative for diabetes
        if p_pos >= p_neg:
            da_test1.iloc[i,3] = 'Positive'
        else:
            da_test1.iloc[i,3] = 'Negative'

    else:
        #If Age <= 30
        if da_test1.iloc[i,0] <=30:

            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[1,1] * df4.iloc[0,1] * df1.iloc[2,1] * df5.
↪iloc[0,1]
                p_neg = df2.iloc[1,2] * df4.iloc[0,2] * df1.iloc[2,2] * df5.
↪iloc[0,2]
            else:
                p_pos = df2.iloc[1,1] * df4.iloc[0,1] * df1.iloc[2,1] * df5.
↪iloc[1,1]
                p_neg = df2.iloc[1,2] * df4.iloc[0,2] * df1.iloc[2,2] * df5.
↪iloc[1,2]

        #If Age 31 - 40
        elif da_test1.iloc[i,0] >30 and da_test1.iloc[i,0] <=40:

            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[1,1] * df4.iloc[1,1] * df1.iloc[2,1] * df5.
↪iloc[0,1]
                p_neg = df2.iloc[1,2] * df4.iloc[1,2] * df1.iloc[2,2] * df5.
↪iloc[0,2]
            else:
                p_pos = df2.iloc[1,1] * df4.iloc[1,1] * df1.iloc[2,1] * df5.
↪iloc[1,1]
                p_neg = df2.iloc[1,2] * df4.iloc[1,2] * df1.iloc[2,2] * df5.
↪iloc[1,2]
```

```
        #If Age 41 - 50
        elif da_test1.iloc[i,0] >40 and da_test1.iloc[i,0] <=50:

            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[1,1] * df4.iloc[2,1] * df1.iloc[2,1] * df5.
↪iloc[0,1]
                p_neg = df2.iloc[1,2] * df4.iloc[2,2] * df1.iloc[2,2] * df5.
↪iloc[0,2]
            else:
                p_pos = df2.iloc[1,1] * df4.iloc[2,1] * df1.iloc[2,1] * df5.
↪iloc[1,1]
                p_neg = df2.iloc[1,2] * df4.iloc[2,2] * df1.iloc[2,2] * df5.
↪iloc[1,2]

        #If Age 51 - 60
        elif da_test1.iloc[i,0] >50 and da_test1.iloc[i,0] <=60:

            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[1,1] * df4.iloc[3,1] * df1.iloc[2,1] * df5.
↪iloc[0,1]
                p_neg = df2.iloc[1,2] * df4.iloc[3,2] * df1.iloc[2,2] * df5.
↪iloc[0,2]
            else:
                p_pos = df2.iloc[1,1] * df4.iloc[3,1] * df1.iloc[2,1] * df5.
↪iloc[1,1]
                p_neg = df2.iloc[1,2] * df4.iloc[3,2] * df1.iloc[2,2] * df5.
↪iloc[1,2]

        #If Age 61 - 70
        elif da_test1.iloc[i,0] >60 and da_test1.iloc[i,0] <=70:

            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[1,1] * df4.iloc[4,1] * df1.iloc[2,1] * df5.
↪iloc[0,1]
                p_neg = df2.iloc[1,2] * df4.iloc[4,2] * df1.iloc[2,2] * df5.
↪iloc[0,2]
            else:
                p_pos = df2.iloc[1,1] * df4.iloc[4,1] * df1.iloc[2,1] * df5.
↪iloc[1,1]
                p_neg = df2.iloc[1,2] * df4.iloc[4,2] * df1.iloc[2,2] * df5.
↪iloc[1,2]
```

```
        #If Age 71 - 80
        elif da_test1.iloc[i,0] >70 and da_test1.iloc[i,0] <=80:

            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[1,1] * df4.iloc[5,1] * df1.iloc[2,1] * df5.iloc[0,1]
                p_neg = df2.iloc[1,2] * df4.iloc[5,2] * df1.iloc[2,2] * df5.iloc[0,2]
            else:
                p_pos = df2.iloc[1,1] * df4.iloc[5,1] * df1.iloc[2,1] * df5.iloc[1,1]
                p_neg = df2.iloc[1,2] * df4.iloc[5,2] * df1.iloc[2,2] * df5.iloc[1,2]

        #If Age >= 80
        else:

            #If person is suffering from polyuria
            if da_test1.iloc[i,2] == 'Yes':
                p_pos = df2.iloc[1,1] * df4.iloc[6,1] * df1.iloc[2,1] * df5.iloc[0,1]
                p_neg = df2.iloc[1,2] * df4.iloc[6,2] * df1.iloc[2,2] * df5.iloc[0,2]
            else:
                p_pos = df2.iloc[1,1] * df4.iloc[6,1] * df1.iloc[2,1] * df5.iloc[1,1]
                p_neg = df2.iloc[1,2] * df4.iloc[6,2] * df1.iloc[2,2] * df5.iloc[1,2]

        #Decision for Positive or Negative for diabetes
        if p_pos >= p_neg:
            da_test1.iloc[i,3] = 'Positive'
        else:
            da_test1.iloc[i,3] = 'Negative'
```

[56]:

```
da_final1 =da_test1
```

[57]:

```
da_final1[0:5]
```

[57]:

```
   Age Gender Polyuria     class
0   40   Male       No  Negative
1   58   Male       No  Negative
2   41   Male      Yes  Positive
3   45   Male       No  Negative
4   60   Male      Yes  Positive
```

[58]:

```python
da_final1.to_excel('C:\MTech Data Science\Mini Project\Diabetes_Final1.xlsx', sheet_name='Sheet1')
```

[ ]:

```python
#Upon Manual inspection we found that 88 target values out of 100 are correct, thus accuracy: (0.88) 88%
```