Implement Bayes Theorem using Python

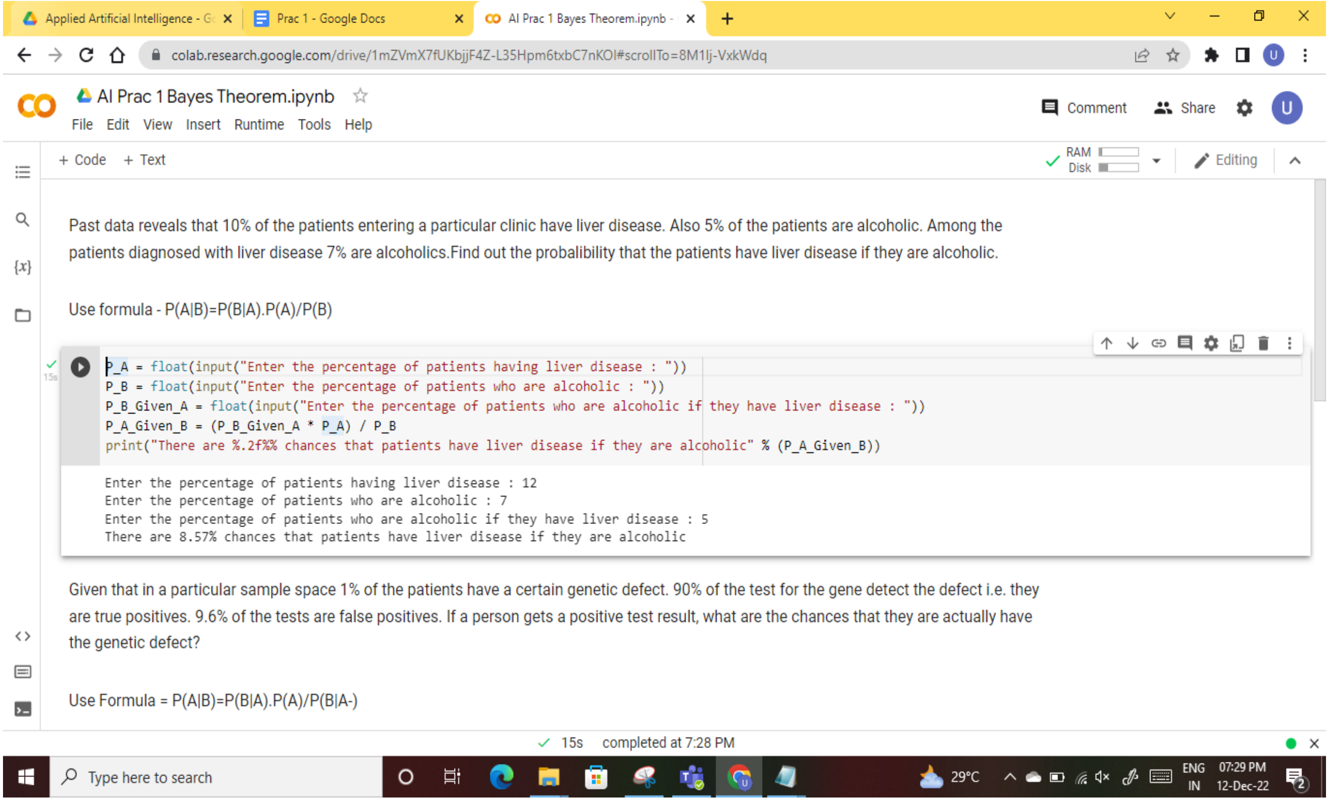
Code :

1) Past data reveals that 10% of the patients entering a particular clinic have liver disease. Also 5% of the patients are alcoholic. Among the patients diagnosed with liver disease 7% are alcoholics.Find out the probability that the patients have liver disease if they are alcoholic.

Use formula - P(A|B)=P(B|A).P(A)/P(B)

```
P_A = float(input("Enter the percentage of patients having liver disease : "))
P_B = float(input("Enter the percentage of patients who are alcoholic : "))
P_B_Given_A = float(input("Enter the percentage of patients who are alcoholic if they
                    have liver disease : "))
P_A_Given_B = (P_B_Given_A * P_A) / P_B
print("There are %.2f%% chances that patients have liver disease if they are alcoholic" %
      (P_A_Given_B))
```

Output :


Applied Artificial Intelligence - G
Prac 1 - Google Docs
AI Prac 1 Bayes Theorem.ipynb -
colab.research.google.com/drive/1mZVmX7fUKbjjF42-L35Hpm6txbC7nKOI#scrollTo=8M1Ij-VxkWdq
AI Prac 1 Bayes Theorem.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing

Past data reveals that 10% of the patients entering a particular clinic have liver disease. Also 5% of the patients are alcoholic. Among the patients diagnosed with liver disease 7% are alcoholics.Find out the probalibility that the patients have liver disease if they are alcoholic.

Use formula - P(A|B)=P(B|A).P(A)/P(B)

```
P_A = float(input("Enter the percentage of patients having liver disease : "))
P_B = float(input("Enter the percentage of patients who are alcoholic : "))
P_B_Given_A = float(input("Enter the percentage of patients who are alcoholic if they have liver disease : "))
P_A_Given_B = (P_B_Given_A * P_A) / P_B
print("There are %.2f%% chances that patients have liver disease if they are alcoholic" % (P_A_Given_B))
```

```
Enter the percentage of patients having liver disease : 12
Enter the percentage of patients who are alcoholic : 7
Enter the percentage of patients who are alcoholic if they have liver disease : 5
There are 8.57% chances that patients have liver disease if they are alcoholic
```

Given that in a particular sample space 1% of the patients have a certain genetic defect. 90% of the test for the gene detect the defect i.e. they are true positives. 9.6% of the tests are false positives. If a person gets a positive test result, what are the chances that they are actually have the genetic defect?

Use Formula = P(A|B)=P(B|A).P(A)/P(B|A-)

2) Given that in a particular sample space 1% of the patients have a certain genetic defect. 90% of the tests for the gene detect the defect i.e. they are true positives. 9.6% of the tests are false positives. If a person gets a positive test result, what are the chances that they actually have the genetic defect?

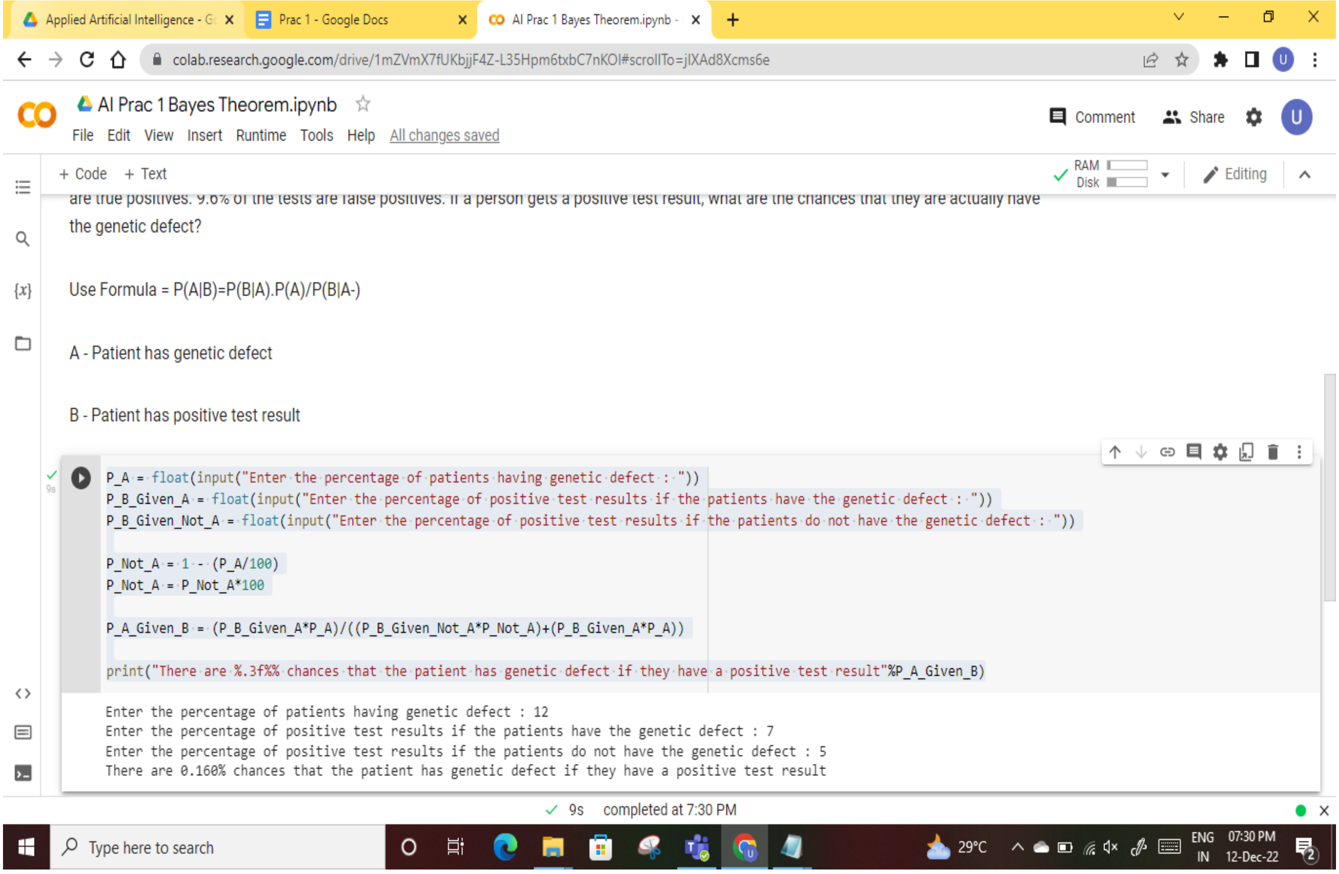
Use Formula = P(A|B)=P(B|A).P(A)/P(B|A-)

A - Patient has genetic defect

B - Patient has positive test result

```
P_A = float(input("Enter the percentage of patients having genetic defect : "))
P_B_Given_A = float(input("Enter the percentage of positive test results if the patients
                        have the genetic defect : "))
P_B_Given_Not_A = float(input("Enter the percentage of positive test results if the
                             patients do not have the genetic defect : "))

P_Not_A = 1 - (P_A/100)
P_Not_A = P_Not_A*100

P_A_Given_B                                                                    =
(P_B_Given_A*P_A)/((P_B_Given_Not_A*P_Not_A)+(P_B_Given_A*P_A))

print("There are %.3f%% chances that the patient has genetic defect if they have a
      positive test result"%P_A_Given_B)
```

Output :


Applied Artificial Intelligence - G
Prac 1 - Google Docs
AI Prac 1 Bayes Theorem.ipynb -
colab.research.google.com/drive/1mZVmX7fUKbjjF4Z-L35Hpm6txbC7nKOl#scrollTo=jlXAd8Xcms6e
AI Prac 1 Bayes Theorem.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
Type here to search
29°C
ENG IN
07:30 PM 12-Dec-22

are true positives. 9.6% of the tests are false positives. If a person gets a positive test result, what are the chances that they are actually have the genetic defect?

Use Formula = P(A|B)=P(B|A).P(A)/P(B|A-)

A - Patient has genetic defect

B - Patient has positive test result

```
P_A = float(input("Enter the percentage of patients having genetic defect : "))
P_B_Given_A = float(input("Enter the percentage of positive test results if the patients have the genetic defect : "))
P_B_Given_Not_A = float(input("Enter the percentage of positive test results if the patients do not have the genetic defect : "))

P_Not_A = 1 - (P_A/100)
P_Not_A = P_Not_A*100

P_A_Given_B = (P_B_Given_A*P_A)/((P_B_Given_Not_A*P_Not_A)+(P_B_Given_A*P_A))

print("There are %.3f%% chances that the patient has genetic defect if they have a positive test result"%P_A_Given_B)
```

```
Enter the percentage of patients having genetic defect : 12
Enter the percentage of positive test results if the patients have the genetic defect : 7
Enter the percentage of positive test results if the patients do not have the genetic defect : 5
There are 0.160% chances that the patient has genetic defect if they have a positive test result
```

✓ 9s completed at 7:30 PM

Implement Conditional Probability and Joint probability using Python

A) Conditional Probability :

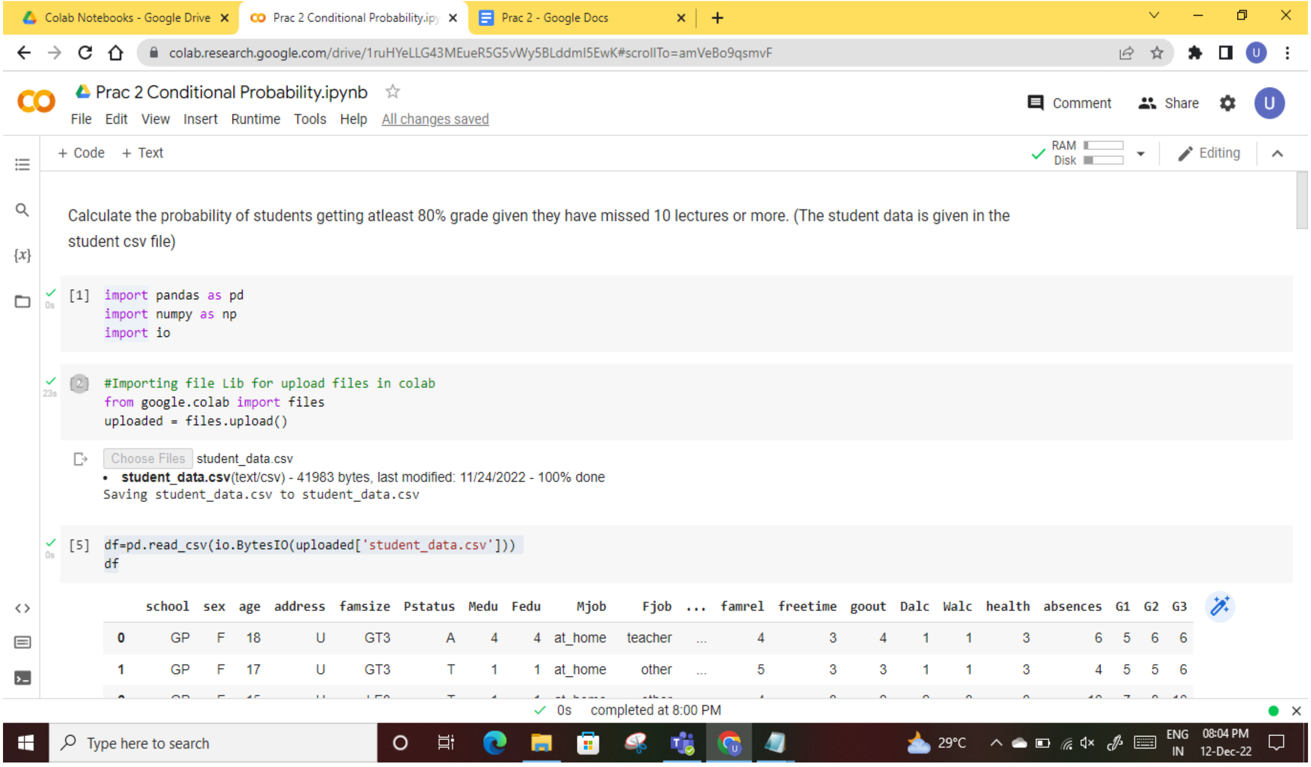
Calculate the probability of students getting at least 80% grade given they have missed 10 lectures or more. (The student data is given in the student csv file)

Code :

```
import pandas as pd
import numpy as np
import io

#Importing file Lib for upload files in colab
from google.colab import files
uploaded = files.upload()

df=pd.read_csv(io.BytesIO(uploaded['student_data.csv']))
len(df)
df['G']= round((df['G1'] + df['G2'] + df['G3']) / 3)

df['Percentage'] = df['G'] * 5
df['Grade_0'] = np.where(df['Percentage'] >= 80, 1, 0)
df.head(10)

df['High_Absentees'] = np.where(df['absences'] >= 10, 1, 1)
df.head(10)

df['Count'] = 1
df.head(10)

df = df[['Grade_0','High_Absentees','Count']]
df.head(10)
```

```
df = df[['Grade_0','High_Absentees','Count']]
df.head(5)

pd.pivot_table(df, values='Count', index='Grade_0', columns='High_Absentees',
                    aggfunc=np.size,fill_value=0)
Total = 283 + 78 + 29 + 5

# P(A) is the probability of getting grade of 80% or more
P_A = (29 + 5) / Total
print(P_A)

# P(B) is the probability of missing 10 lectures or more
P_B = (78 + 5) / Total
print(P_B)

# P(A_Intersection_B ) is the probability of getting grade of 80% or more and missing 10
    lectures or more
P_A_Intersection_B = 5 / Total
print(P_A_Intersection_B)

P_A_Given_B = P_A_Intersection_B / P_B
print(P_A_Given_B)

print('Probability of students getting at least 80% grade given they have missed 10 lectures or
      more is ', round(P_A_Given_B,2))
```

Output :



Calculate the probability of students getting atleast 80% grade given they have missed 10 lectures or more. (The student data is given in the student csv file)

```
[1] import pandas as pd
    import numpy as np
    import io
```

```
#Importing file Lib for upload files in colab
from google.colab import files
uploaded = files.upload()
```

Choose Files student_data.csv

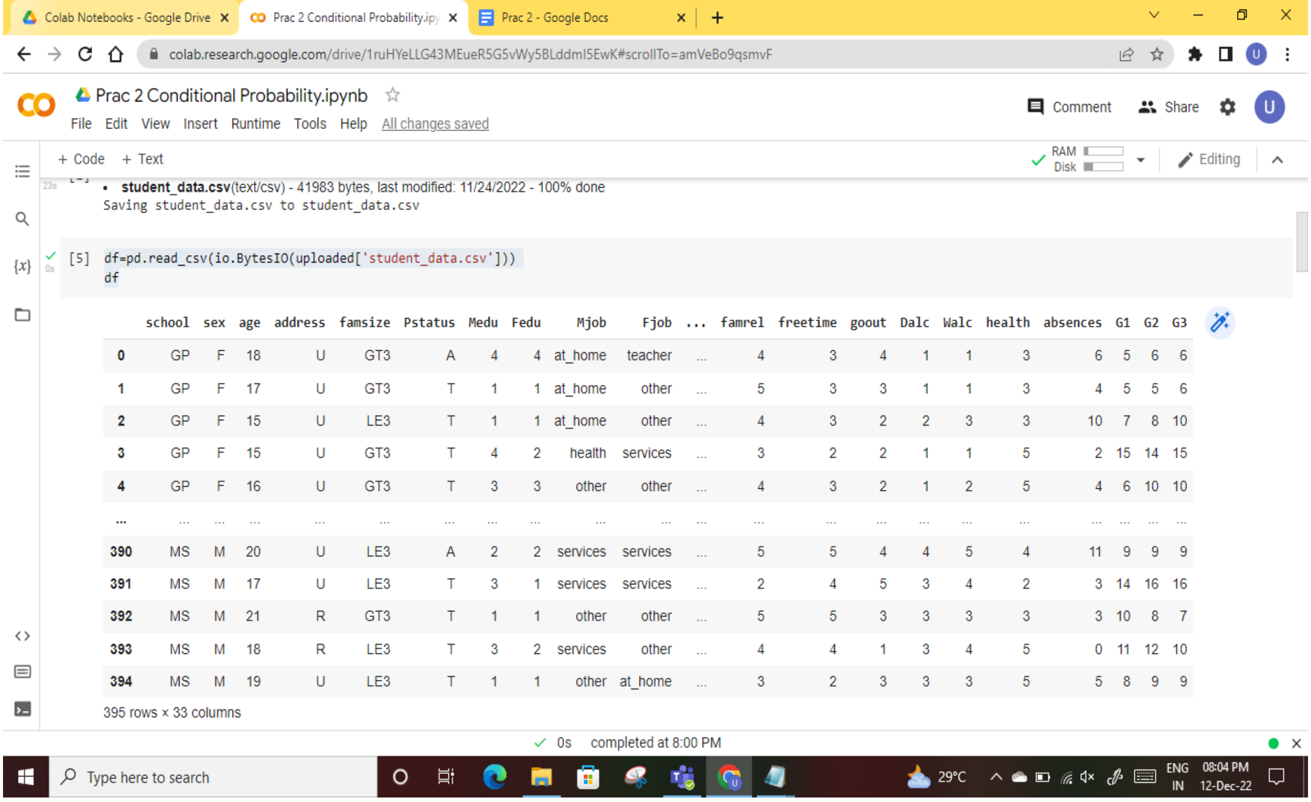
- **student_data.csv**(text/csv) - 41983 bytes, last modified: 11/24/2022 - 100% done

```
Saving student_data.csv to student_data.csv
```

```
[5] df=pd.read_csv(io.BytesIO(uploaded['student_data.csv']))
    df
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6	5	6	6
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4	5	5	6

✓ 0s completed at 8:00 PM



- **student_data.csv**(text/csv) - 41983 bytes, last modified: 11/24/2022 - 100% done

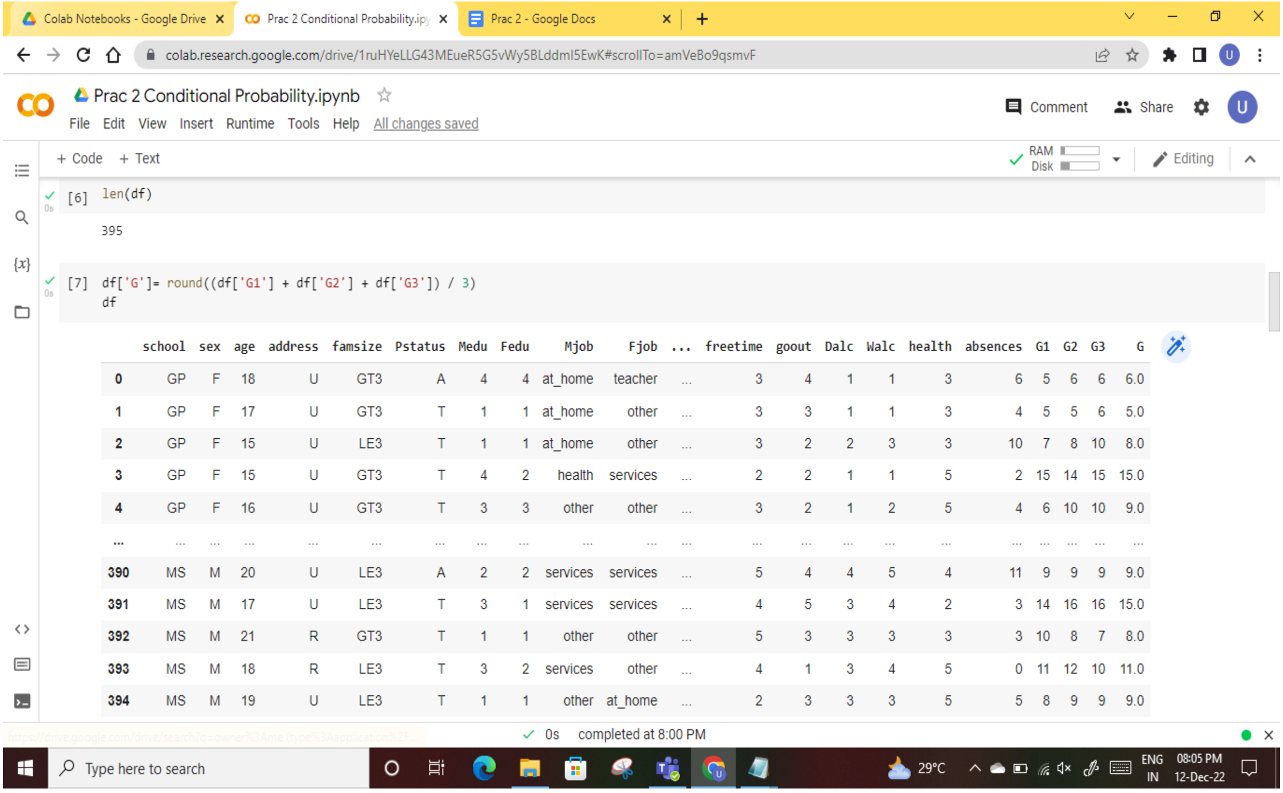
```
Saving student_data.csv to student_data.csv
```

```
[5] df=pd.read_csv(io.BytesIO(uploaded['student_data.csv']))
    df
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6	5	6	6
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4	5	5	6
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	10	7	8	10
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	2	15	14	15
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	4	6	10	10
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	MS	M	20	U	LE3	A	2	2	services	services	...	5	5	4	4	5	4	11	9	9	9
391	MS	M	17	U	LE3	T	3	1	services	services	...	2	4	5	3	4	2	3	14	16	16
392	MS	M	21	R	GT3	T	1	1	other	other	...	5	5	3	3	3	3	3	10	8	7
393	MS	M	18	R	LE3	T	3	2	services	other	...	4	4	1	3	4	5	0	11	12	10
394	MS	M	19	U	LE3	T	1	1	other	at_home	...	3	2	3	3	3	5	5	8	9	9

395 rows × 33 columns

✓ 0s completed at 8:00 PM

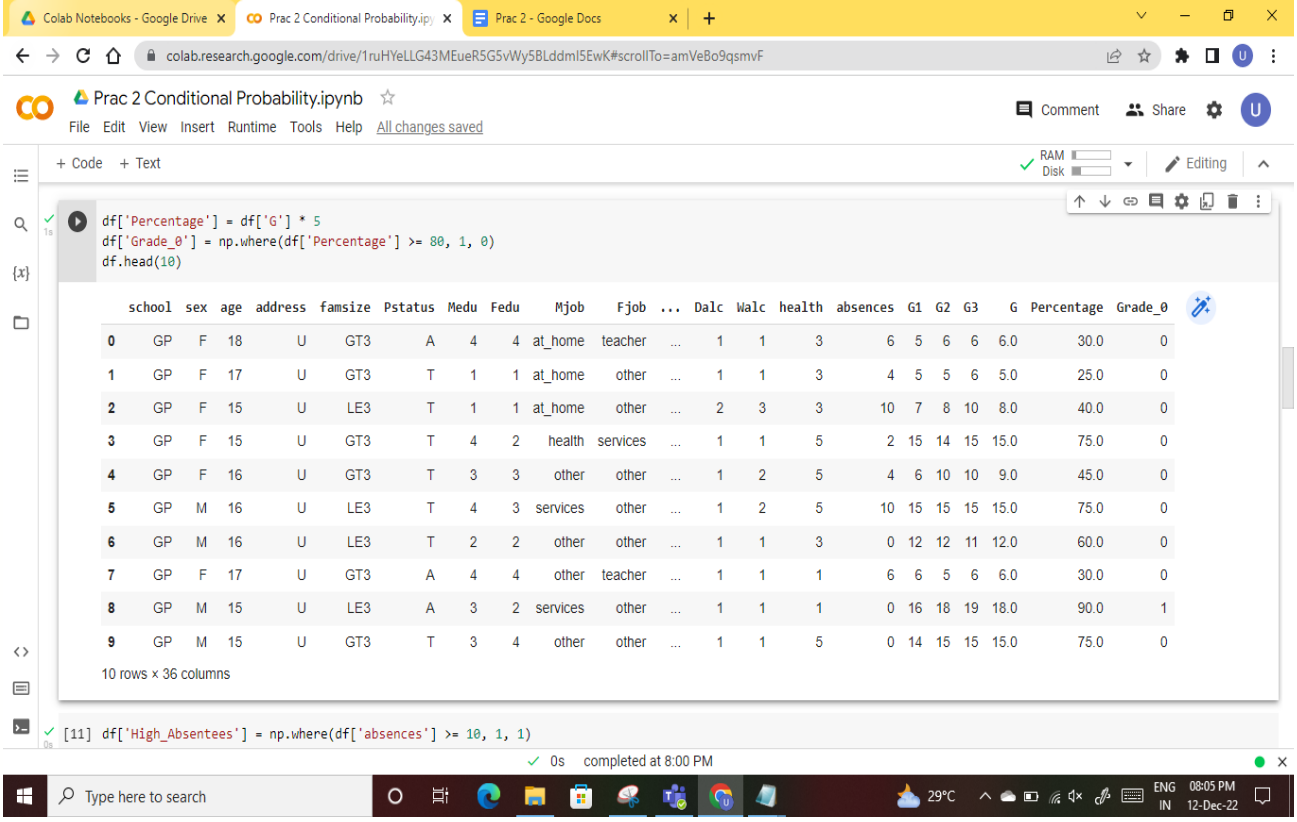

Colab Notebooks - Google Drive
Prac 2 Conditional Probability.ipy
Prac 2 - Google Docs
colab.research.google.com/drive/1ruHYeLLG43MEueR5G5vWy5BLddmI5EwK#scrollTo=amVeBo9qsmvF
Prac 2 Conditional Probability.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
Type here to search
29°C ENG IN 08:05 PM 12-Dec-22

```
[6] len(df)

395
```

```
[7] df['G']= round((df['G1'] + df['G2'] + df['G3']) / 3)
    df
```

```
     school sex age address famsize Pstatus Medu Fedu     Mjob     Fjob ... freetime goout Dalc Walc health absences G1 G2 G3    G
  0      GP   F  18       U     GT3       A    4    4  at_home  teacher ...        3     4    1    1      3        6  5  6  6  6.0
  1      GP   F  17       U     GT3       T    1    1  at_home    other ...        3     3    1    1      3        4  5  5  6  5.0
  2      GP   F  15       U     LE3       T    1    1  at_home    other ...        3     2    2    3      3       10  7  8 10  8.0
  3      GP   F  15       U     GT3       T    4    2   health services ...        2     2    1    1      5        2 15 14 15 15.0
  4      GP   F  16       U     GT3       T    3    3    other    other ...        3     2    1    2      5        4  6 10 10  9.0
...     ... ... ...     ...     ...     ...  ...  ...      ...      ... ...      ...   ...  ...  ...    ...      ... .. .. ..  ...
390      MS   M  20       U     LE3       A    2    2 services services ...        5     4    4    5      4       11  9  9  9  9.0
391      MS   M  17       U     LE3       T    3    1 services services ...        4     5    3    4      2        3 14 16 16 15.0
392      MS   M  21       R     GT3       T    1    1    other    other ...        5     3    3    3      3        3 10  8  7  8.0
393      MS   M  18       R     LE3       T    3    2 services    other ...        4     1    3    4      5        0 11 12 10 11.0
394      MS   M  19       U     LE3       T    1    1    other  at_home ...        2     3    3    3      5        5  8  9  9  9.0
```

✓ 0s completed at 8:00 PM

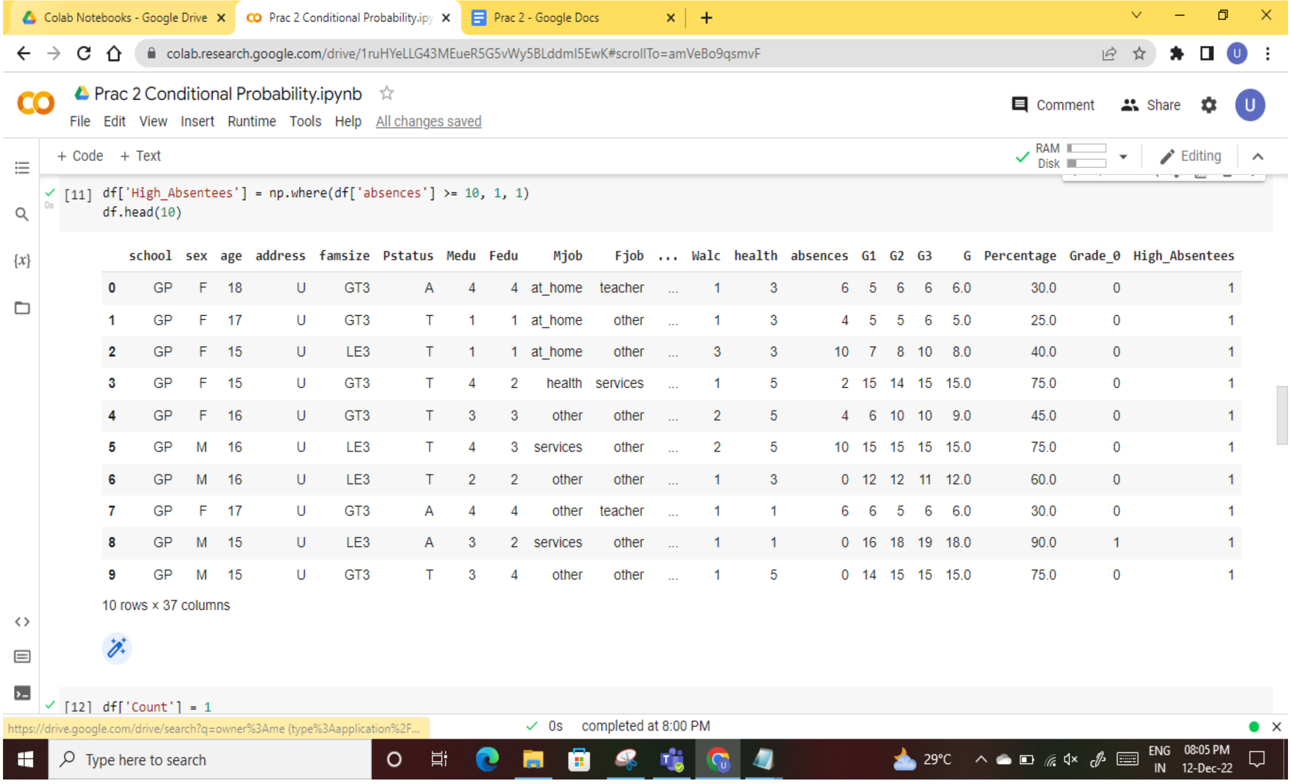

Colab Notebooks - Google Drive
Prac 2 Conditional Probability.ipy
Prac 2 - Google Docs
colab.research.google.com/drive/1ruHYeLLG43MEueR5G5vWy5BLddmI5EwK#scrollTo=amVeBo9qsmvF
Prac 2 Conditional Probability.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
Type here to search
29°C ENG IN 08:05 PM 12-Dec-22

```
df['Percentage'] = df['G'] * 5
df['Grade_0'] = np.where(df['Percentage'] >= 80, 1, 0)
df.head(10)
```

```
  school sex age address famsize Pstatus Medu Fedu     Mjob     Fjob ... Dalc Walc health absences G1 G2 G3    G Percentage Grade_0
0     GP   F  18       U     GT3       A    4    4  at_home  teacher ...    1    1      3        6  5  6  6  6.0       30.0       0
1     GP   F  17       U     GT3       T    1    1  at_home    other ...    1    1      3        4  5  5  6  5.0       25.0       0
2     GP   F  15       U     LE3       T    1    1  at_home    other ...    2    3      3       10  7  8 10  8.0       40.0       0
3     GP   F  15       U     GT3       T    4    2   health services ...    1    1      5        2 15 14 15 15.0       75.0       0
4     GP   F  16       U     GT3       T    3    3    other    other ...    1    2      5        4  6 10 10  9.0       45.0       0
5     GP   M  16       U     LE3       T    4    3 services    other ...    1    2      5       10 15 15 15 15.0       75.0       0
6     GP   M  16       U     LE3       T    2    2    other    other ...    1    1      3        0 12 12 11 12.0       60.0       0
7     GP   F  17       U     GT3       A    4    4    other  teacher ...    1    1      1        6  6  5  6  6.0       30.0       0
8     GP   M  15       U     LE3       A    3    2 services    other ...    1    1      1        0 16 18 19 18.0       90.0       1
9     GP   M  15       U     GT3       T    3    4    other    other ...    1    1      5        0 14 15 15 15.0       75.0       0

10 rows × 36 columns
```

```
[11] df['High_Absentees'] = np.where(df['absences'] >= 10, 1, 1)
```

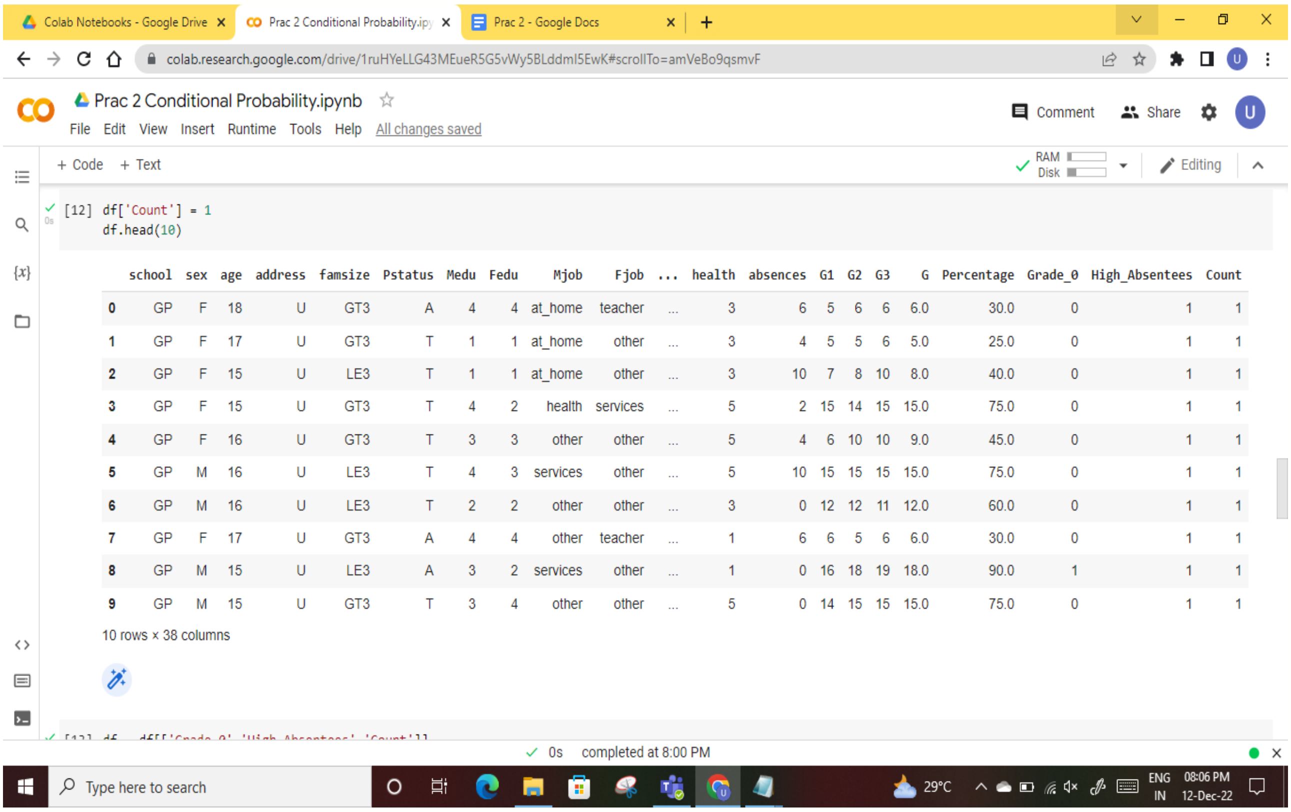
✓ 0s completed at 8:00 PM


Colab Notebooks - Google Drive
Prac 2 Conditional Probability.ipynb
Prac 2 - Google Docs
colab.research.google.com/drive/1ruHYeLLG43MEueR5G5vWy5BLddmI5EwK#scrollTo=amVeBo9qsmvF
Prac 2 Conditional Probability.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
https://drive.google.com/drive/search?q=owner%3Ame (type%3Aapplication%2F...
0s completed at 8:00 PM
Type here to search
29°C
ENG IN 08:05 PM 12-Dec-22

```
[11] df['High_Absentees'] = np.where(df['absences'] >= 10, 1, 1)
     df.head(10)
```

```
  school sex age address famsize Pstatus Medu Fedu     Mjob     Fjob ... Walc health absences G1 G2 G3    G Percentage Grade_0 High_Absentees
0     GP   F  18       U     GT3       A    4    4  at_home  teacher ...    1      3        6  5  6  6  6.0       30.0       0              1
1     GP   F  17       U     GT3       T    1    1  at_home    other ...    1      3        4  5  5  6  5.0       25.0       0              1
2     GP   F  15       U     LE3       T    1    1  at_home    other ...    3      3       10  7  8 10  8.0       40.0       0              1
3     GP   F  15       U     GT3       T    4    2   health services ...    1      5        2 15 14 15 15.0       75.0       0              1
4     GP   F  16       U     GT3       T    3    3    other    other ...    2      5        4  6 10 10  9.0       45.0       0              1
5     GP   M  16       U     LE3       T    4    3 services    other ...    2      5       10 15 15 15 15.0       75.0       0              1
6     GP   M  16       U     LE3       T    2    2    other    other ...    1      3        0 12 12 11 12.0       60.0       0              1
7     GP   F  17       U     GT3       A    4    4    other  teacher ...    1      1        6  6  5  6  6.0       30.0       0              1
8     GP   M  15       U     LE3       A    3    2 services    other ...    1      1        0 16 18 19 18.0       90.0       1              1
9     GP   M  15       U     GT3       T    3    4    other    other ...    1      5        0 14 15 15 15.0       75.0       0              1
10 rows × 37 columns
```

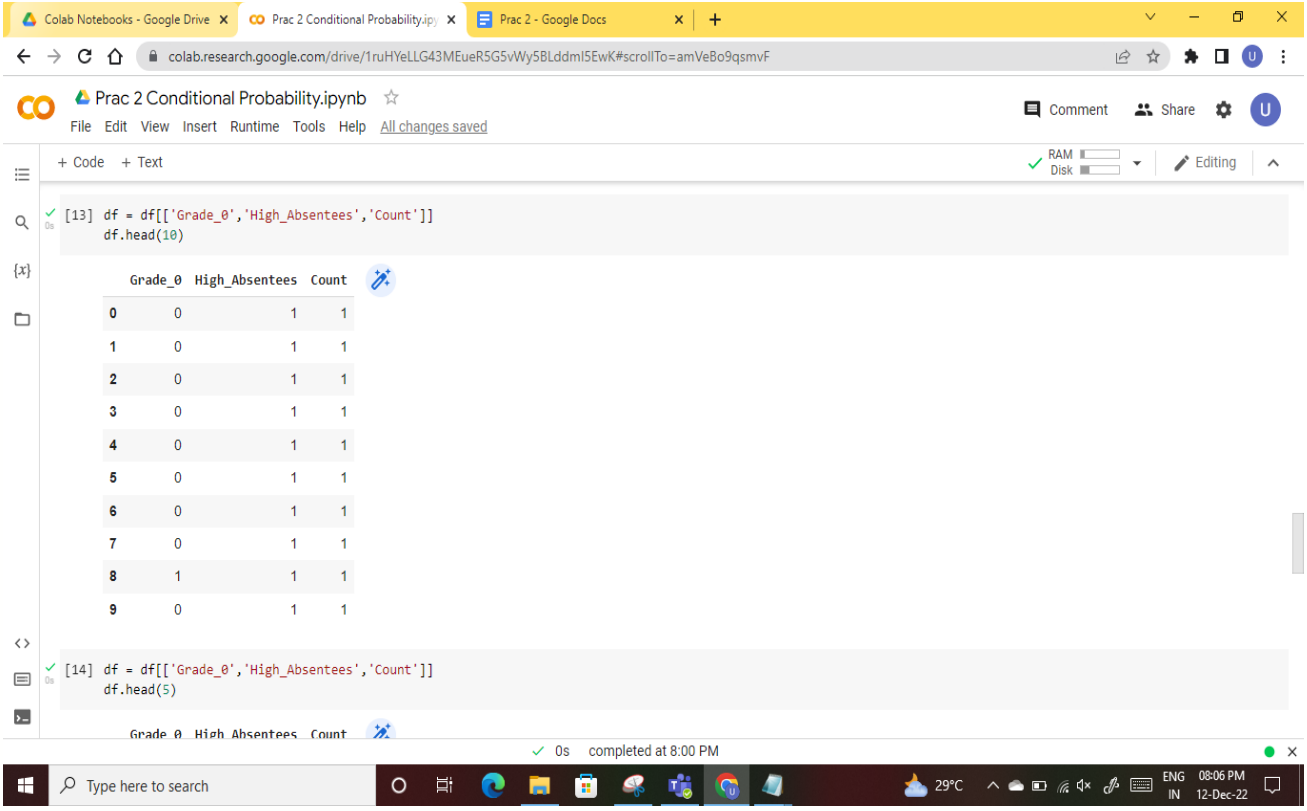
```
[12] df['Count'] = 1
```


Colab Notebooks - Google Drive
Prac 2 Conditional Probability.ipynb
Prac 2 - Google Docs
colab.research.google.com/drive/1ruHYeLLG43MEueR5G5vWy5BLddmI5EwK#scrollTo=amVeBo9qsmvF
Prac 2 Conditional Probability.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
0s completed at 8:00 PM
Type here to search
29°C
ENG IN 08:06 PM 12-Dec-22

```
[12] df['Count'] = 1
     df.head(10)
```

```
  school sex age address famsize Pstatus Medu Fedu     Mjob     Fjob ... health absences G1 G2 G3    G Percentage Grade_0 High_Absentees Count
0     GP   F  18       U     GT3       A    4    4  at_home  teacher ...      3        6  5  6  6  6.0       30.0       0              1     1
1     GP   F  17       U     GT3       T    1    1  at_home    other ...      3        4  5  5  6  5.0       25.0       0              1     1
2     GP   F  15       U     LE3       T    1    1  at_home    other ...      3       10  7  8 10  8.0       40.0       0              1     1
3     GP   F  15       U     GT3       T    4    2   health services ...      5        2 15 14 15 15.0       75.0       0              1     1
4     GP   F  16       U     GT3       T    3    3    other    other ...      5        4  6 10 10  9.0       45.0       0              1     1
5     GP   M  16       U     LE3       T    4    3 services    other ...      5       10 15 15 15 15.0       75.0       0              1     1
6     GP   M  16       U     LE3       T    2    2    other    other ...      3        0 12 12 11 12.0       60.0       0              1     1
7     GP   F  17       U     GT3       A    4    4    other  teacher ...      1        6  6  5  6  6.0       30.0       0              1     1
8     GP   M  15       U     LE3       A    3    2 services    other ...      1        0 16 18 19 18.0       90.0       1              1     1
9     GP   M  15       U     GT3       T    3    4    other    other ...      5        0 14 15 15 15.0       75.0       0              1     1
10 rows × 38 columns
```

```
[13] df = df[['Grade_0','High_Absentees','Count']]
```



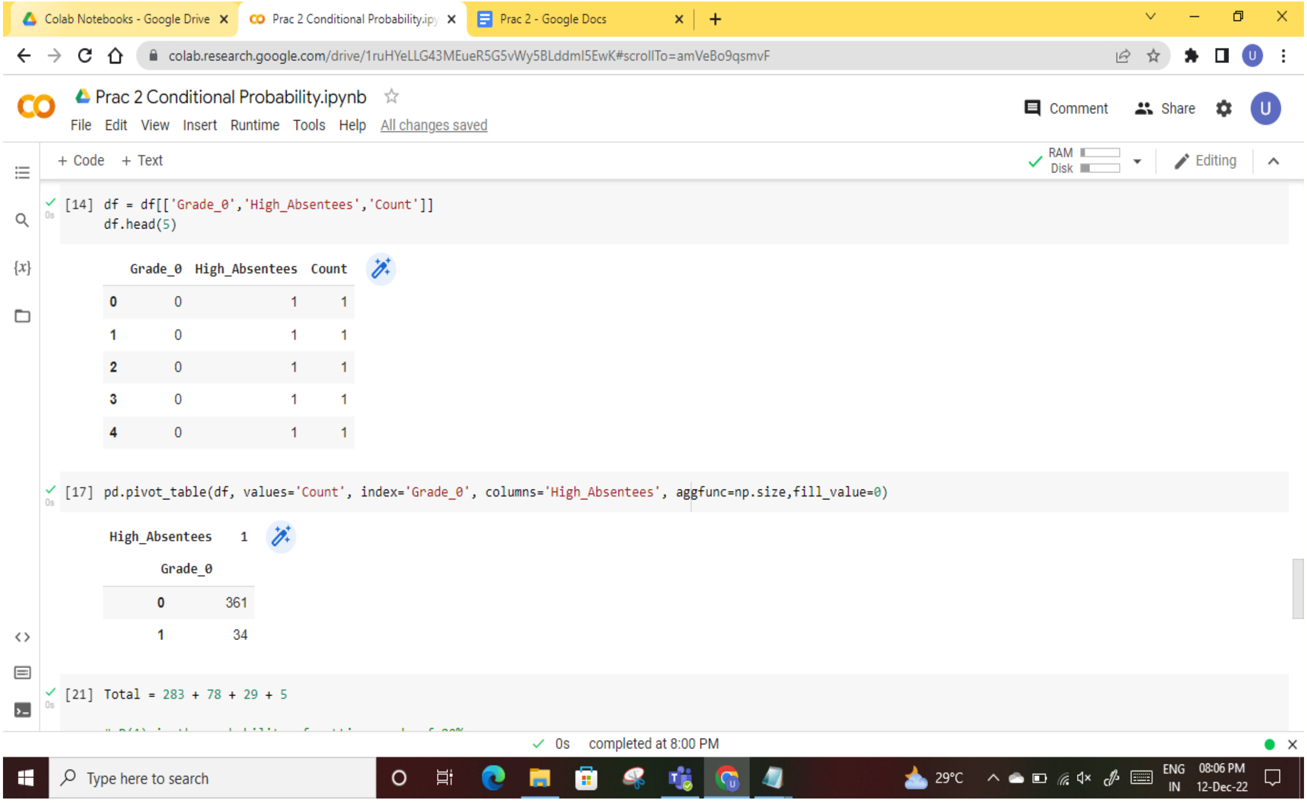
Colab Notebooks - Google Drive
Prac 2 Conditional Probability.ipynb
Prac 2 - Google Docs
colab.research.google.com/drive/1ruHYeLLG43MEueR5G5vWy5BLddmI5EwK#scrollTo=amVeBo9qsmvF
Prac 2 Conditional Probability.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
0s completed at 8:00 PM
Type here to search
29°C
ENG IN 08:06 PM 12-Dec-22

```
[13] df = df[['Grade_0','High_Absentees','Count']]
     df.head(10)
```

```
   Grade_0  High_Absentees  Count
0        0               1      1
1        0               1      1
2        0               1      1
3        0               1      1
4        0               1      1
5        0               1      1
6        0               1      1
7        0               1      1
8        1               1      1
9        0               1      1
```

```
[14] df = df[['Grade_0','High_Absentees','Count']]
     df.head(5)
```

```
   Grade_0  High_Absentees  Count
```



Colab Notebooks - Google Drive
Prac 2 Conditional Probability.ipynb
Prac 2 - Google Docs
colab.research.google.com/drive/1ruHYeLLG43MEueR5G5vWy5BLddmI5EwK#scrollTo=amVeBo9qsmvF
Prac 2 Conditional Probability.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
0s completed at 8:00 PM
Type here to search
29°C
ENG IN 08:06 PM 12-Dec-22

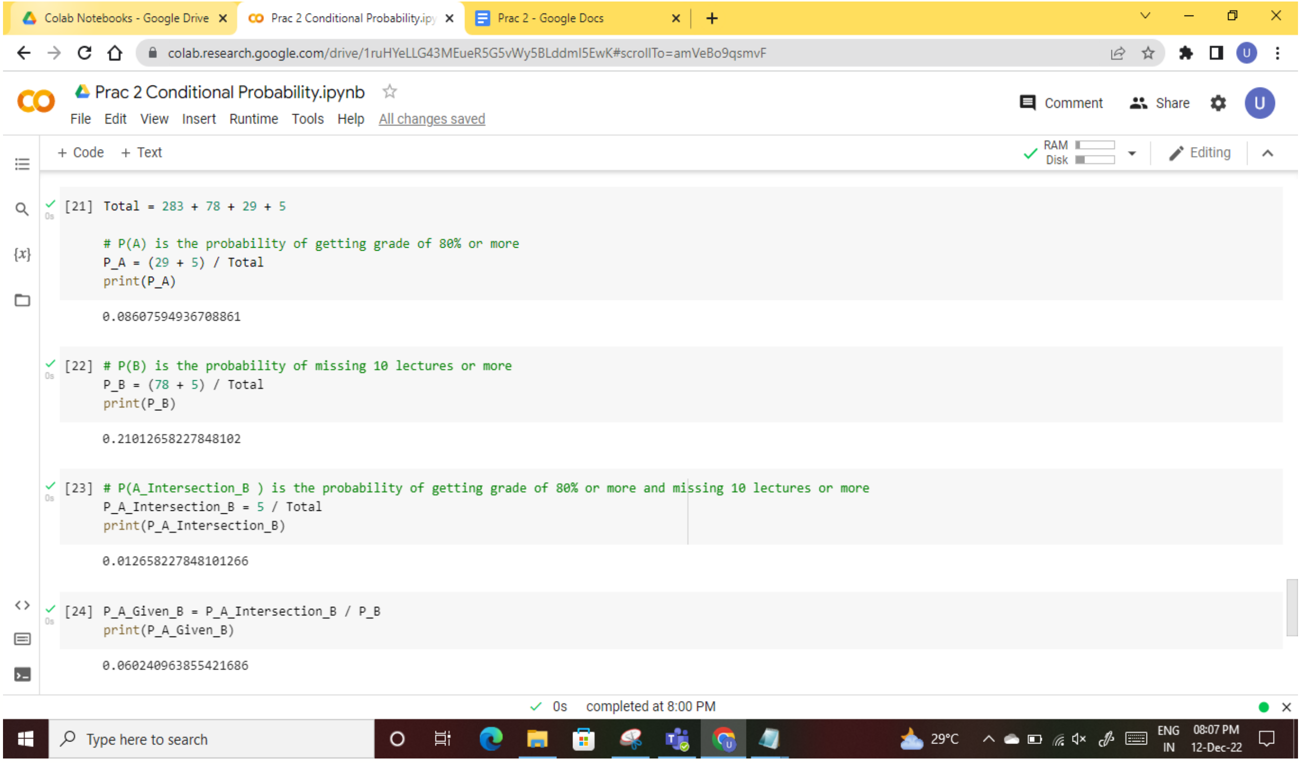
```
[14] df = df[['Grade_0','High_Absentees','Count']]
     df.head(5)
```

```
   Grade_0  High_Absentees  Count
0        0               1      1
1        0               1      1
2        0               1      1
3        0               1      1
4        0               1      1
```

```
[17] pd.pivot_table(df, values='Count', index='Grade_0', columns='High_Absentees', aggfunc=np.size,fill_value=0)
```

```
High_Absentees    1
       Grade_0
             0  361
             1   34
```

```
[21] Total = 283 + 78 + 29 + 5
```

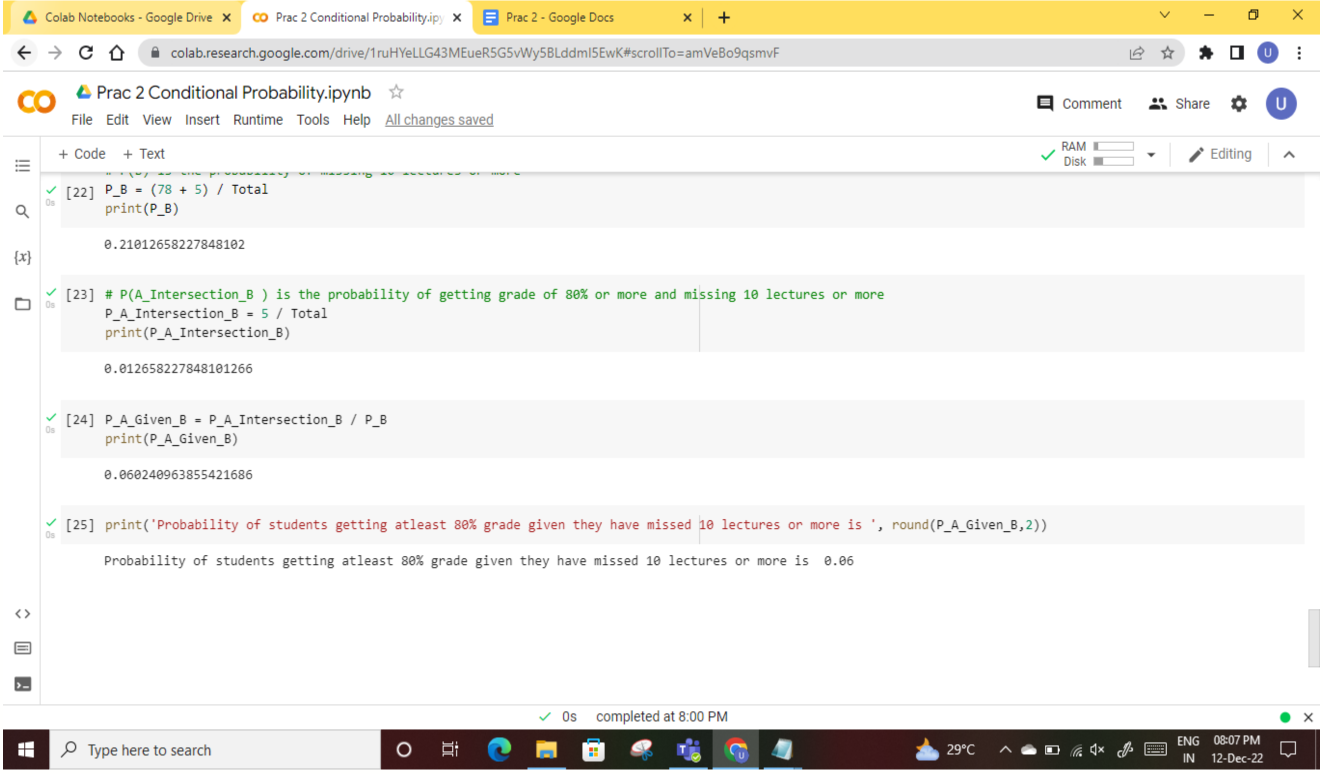

Prac 2 Conditional Probability.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Editing
completed at 8:00 PM

```
[21] Total = 283 + 78 + 29 + 5

     # P(A) is the probability of getting grade of 80% or more
     P_A = (29 + 5) / Total
     print(P_A)

     0.08607594936708861
```

```
[22] # P(B) is the probability of missing 10 lectures or more
     P_B = (78 + 5) / Total
     print(P_B)

     0.21012658227848102
```

```
[23] # P(A_Intersection_B ) is the probability of getting grade of 80% or more and missing 10 lectures or more
     P_A_Intersection_B = 5 / Total
     print(P_A_Intersection_B)

     0.012658227848101266
```

```
[24] P_A_Given_B = P_A_Intersection_B / P_B
     print(P_A_Given_B)

     0.060240963855421686
```


Prac 2 Conditional Probability.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Editing
completed at 8:00 PM

```
[22] P_B = (78 + 5) / Total
     print(P_B)

     0.21012658227848102
```

```
[23] # P(A_Intersection_B ) is the probability of getting grade of 80% or more and missing 10 lectures or more
     P_A_Intersection_B = 5 / Total
     print(P_A_Intersection_B)

     0.012658227848101266
```

```
[24] P_A_Given_B = P_A_Intersection_B / P_B
     print(P_A_Given_B)

     0.060240963855421686
```

```
[25] print('Probability of students getting atleast 80% grade given they have missed 10 lectures or more is ', round(P_A_Given_B,2))

     Probability of students getting atleast 80% grade given they have missed 10 lectures or more is  0.06
```

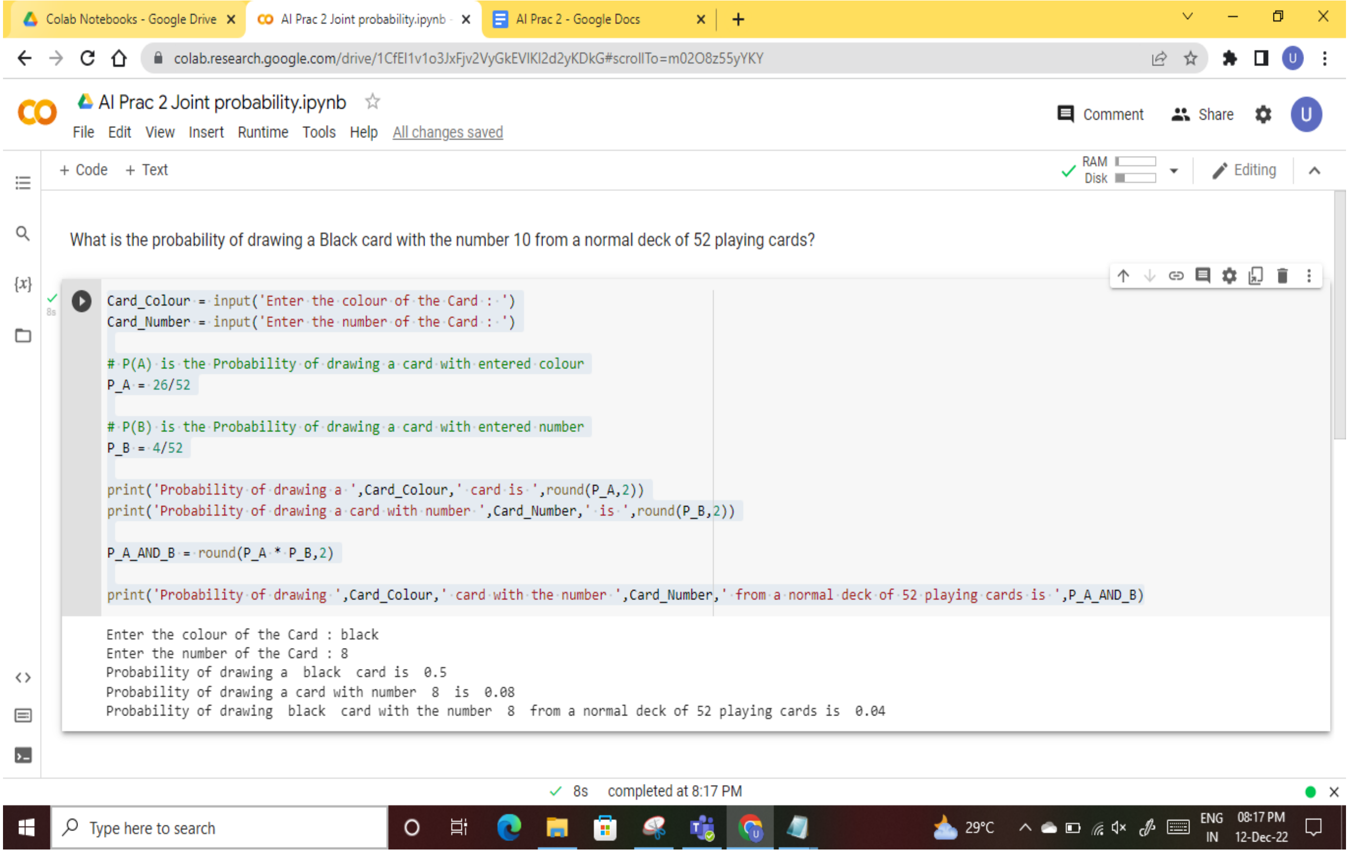
B) Joint probability :

What is the probability of drawing a Black card with the number 10 from a normal deck of 52 playing cards?

Code :

```
Card_Colour = input('Enter the colour of the Card : ')
Card_Number = input('Enter the number of the Card : ')

# P(A) is the Probability of drawing a card with entered colour
P_A = 26/52

# P(B) is the Probability of drawing a card with entered number
P_B = 4/52

print('Probability of drawing a ',Card_Colour,' card is ',round(P_A,2))
print('Probability of drawing a card with number ',Card_Number,' is ',round(P_B,2))

P_A_AND_B = round(P_A * P_B,2)

print('Probability of drawing ',Card_Colour,' card with the number ',Card_Number,' from a normal
        deck of 52 playing cards is ',P_A_AND_B)
```

Output :


Colab Notebooks - Google Drive
AI Prac 2 Joint probability.ipynb
AI Prac 2 - Google Docs
colab.research.google.com/drive/1CfEl1v1o3JxFjv2VyGkEVIKl2d2yKDkG#scrollTo=m02O8z55yYKY
AI Prac 2 Joint probability.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing

What is the probability of drawing a Black card with the number 10 from a normal deck of 52 playing cards?

```
Card_Colour = input('Enter the colour of the Card : ')
Card_Number = input('Enter the number of the Card : ')

# P(A) is the Probability of drawing a card with entered colour
P_A = 26/52

# P(B) is the Probability of drawing a card with entered number
P_B = 4/52

print('Probability of drawing a ',Card_Colour,' card is ',round(P_A,2))
print('Probability of drawing a card with number ',Card_Number,' is ',round(P_B,2))

P_A_AND_B = round(P_A * P_B,2)

print('Probability of drawing ',Card_Colour,' card with the number ',Card_Number,' from a normal deck of 52 playing cards is ',P_A_AND_B)
```

```
Enter the colour of the Card : black
Enter the number of the Card : 8
Probability of drawing a  black  card is  0.5
Probability of drawing a card with number  8  is  0.08
Probability of drawing  black  card with the number  8  from a normal deck of 52 playing cards is  0.04
```

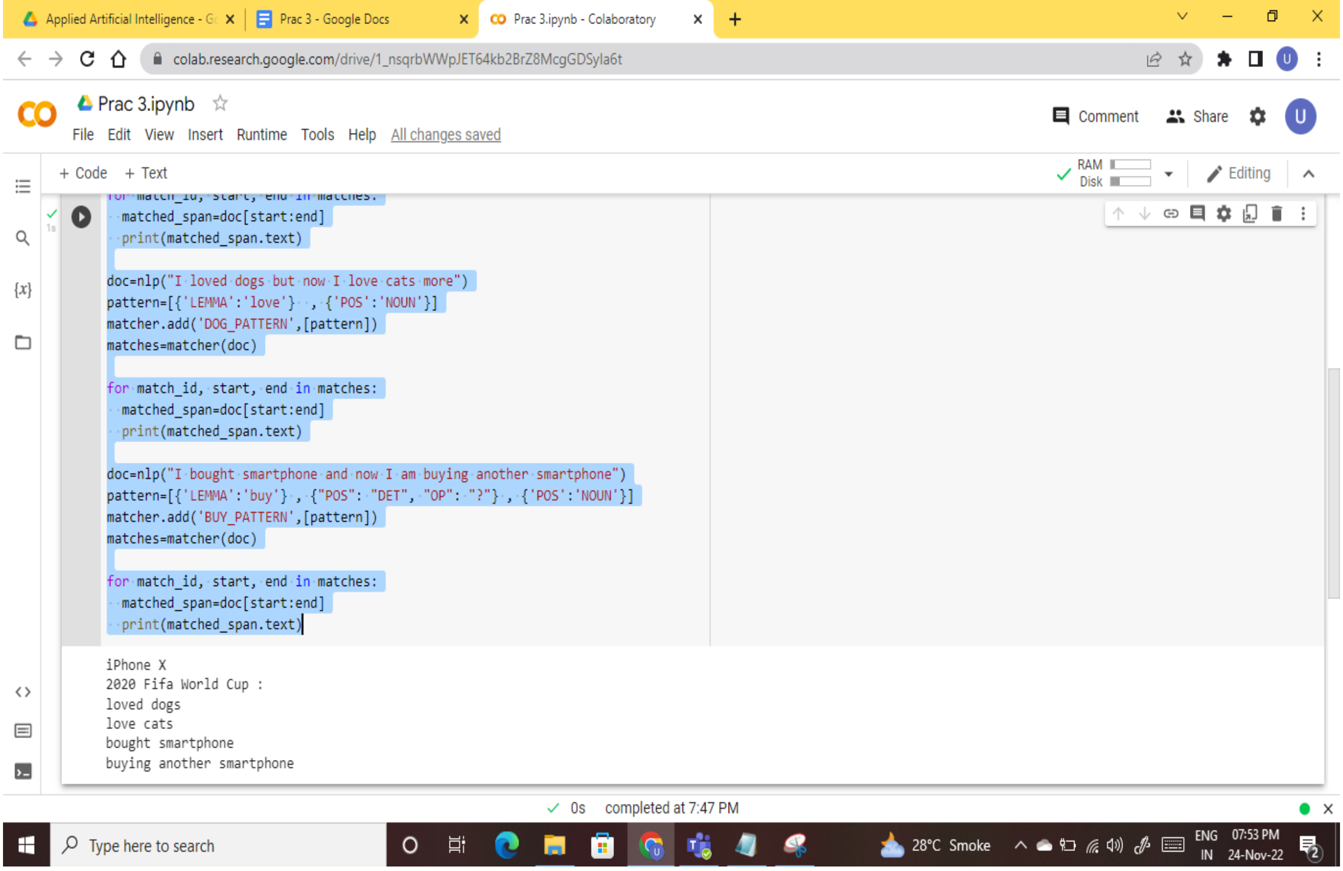
Write a program to implement Rule based system.

Code :

```
import spacy
from spacy.matcher import Matcher
nlp=spacy.load('en_core_web_sm')
matcher=Matcher(nlp.vocab)

doc=nlp("New iPhone X is released")
pattern=[{'ORTH':'iPhone'} , {'ORTH':'X'}]
matcher.add('IPHONE_PATTERN', [pattern])
matches=matcher(doc)

for match_id, start, end in matches:
        matched_span=doc[start:end]
        print(matched_span.text)

doc=nlp("2020 Fifa World Cup : Italy Wins")
pattern=[{'IS_DIGIT':True} , {'LOWER':'fifa'} , {'LOWER':'world'} , {'LOWER':'cup'},
          {'IS_PUNCT':True}]
matcher.add('FIFA_PATTERN',[pattern])
matches=matcher(doc)

for match_id, start, end in matches:
        matched_span=doc[start:end]
        print(matched_span.text)

doc=nlp("I loved dogs but now I love cats more")
pattern=[{'LEMMA':'love'}  , {'POS':'NOUN'}]
matcher.add('DOG_PATTERN',[pattern])
matches=matcher(doc)
```

```
for match_id, start, end in matches:
        matched_span=doc[start:end]
        print(matched_span.text)

doc=nlp("I bought smartphone and now I am buying another smartphone")
pattern=[{'LEMMA':'buy'} , {"POS": "DET", "OP": "?"} , {'POS':'NOUN'}]
matcher.add('BUY_PATTERN',[pattern])
matches=matcher(doc)

for match_id, start, end in matches:
        matched_span=doc[start:end]
        print(matched_span.text)
```

Output :



```
  matched_span=doc[start:end]
  print(matched_span.text)

doc=nlp("I loved dogs but now I love cats more")
pattern=[{'LEMMA':'love'}  , {'POS':'NOUN'}]
matcher.add('DOG_PATTERN',[pattern])
matches=matcher(doc)

for match_id, start, end in matches:
  matched_span=doc[start:end]
  print(matched_span.text)

doc=nlp("I bought smartphone and now I am buying another smartphone")
pattern=[{'LEMMA':'buy'} , {"POS": "DET", "OP": "?"} , {'POS':'NOUN'}]
matcher.add('BUY_PATTERN',[pattern])
matches=matcher(doc)

for match_id, start, end in matches:
  matched_span=doc[start:end]
  print(matched_span.text)
```

```
iPhone X
2020 Fifa World Cup :
loved dogs
love cats
bought smartphone
buying another smartphone
```

Applied Artificial Intelligence - G
Prac 3 - Google Docs
Prac 3.ipynb - Colaboratory
colab.research.google.com/drive/1_nsqrbWWpJET64kb2BrZ8McgGDSyla6t
Prac 3.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0s completed at 7:47 PM
Type here to search
28°C Smoke
ENG IN
07:53 PM
24-Nov-22

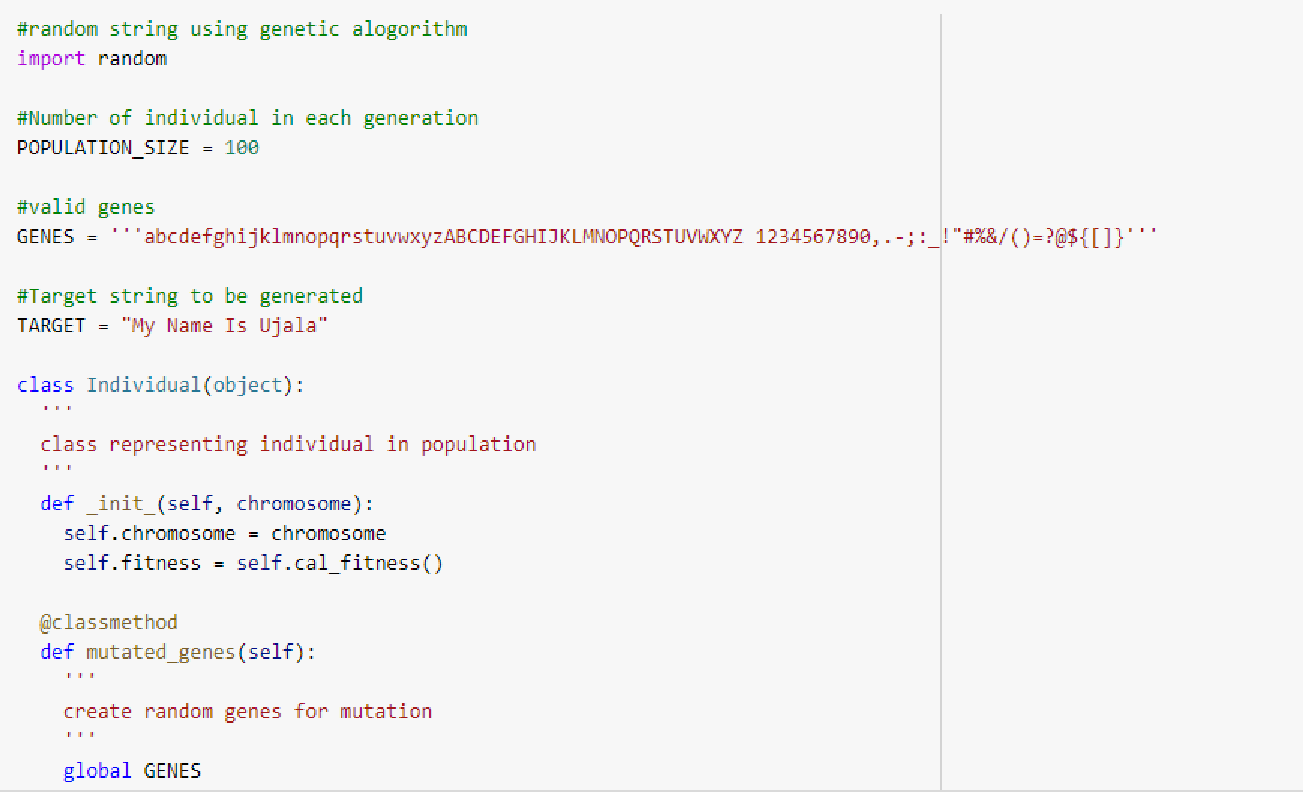
Simulate Genetic Algorithm with suitable example using Python.

Code :

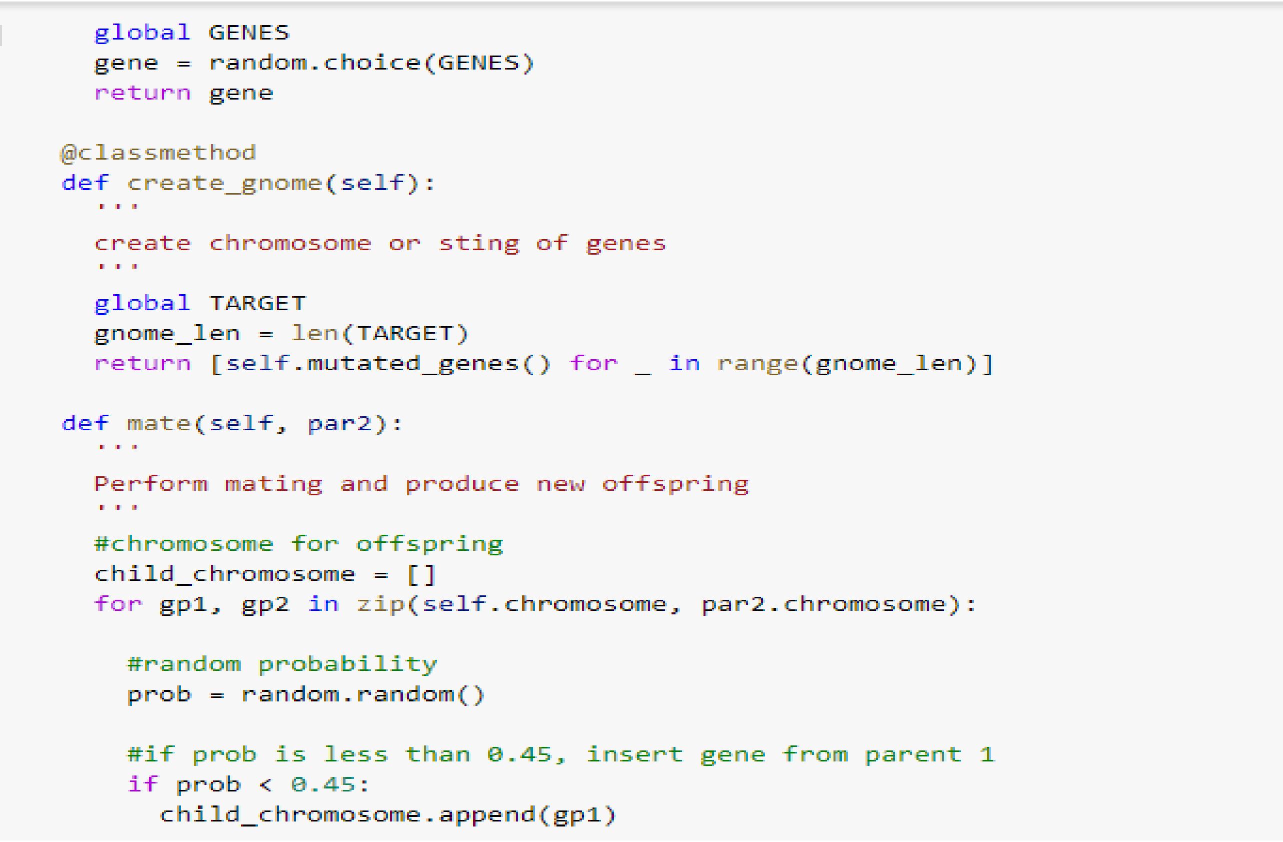
```
#random string using genetic algorithm
import random

#Number of individual in each generation
POPULATION_SIZE = 100

#valid genes
GENES = '''abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
    1234567890,.-;:_!"#%&/()=?@${[]}'''

#Target string to be generated
TARGET = "My Name Is Ujala"

class Individual(object):
    '''
    class representing individual in population
    '''
    def _init_(self, chromosome):
        self.chromosome = chromosome
        self.fitness = self.cal_fitness()

    @classmethod
    def mutated_genes(self):
        '''
        create random genes for mutation
        '''
        global GENES
        gene = random.choice(GENES)
        return gene
    @classmethod
    def create_gnome(self):
        '''
        create chromosome or sting of genes
        '''
        global TARGET
        gnome_len = len(TARGET)
        return [self.mutated_genes() for _ in range(gnome_len)]

    def mate(self, par2):
        '''
        Perform mating and produce new offspring
        '''
        #chromosome for offspring
        child_chromosome = []
        for gp1, gp2 in zip(self.chromosome, par2.chromosome):

        #random probability
        prob = random.random()

        #if prob is less than 0.45, insert gene from parent 1
        if prob < 0.45:
            child_chromosome.append(gp1)
        elif prob < 0.90:
            child_chromosome.append(gp2)
        else :
            child_chromosome.append(self.mutated_genes())

        return Individual(child_chromosome)

    def cal_fitness(self):
        global TARGET
        fitness = 0
        for gs, gt in zip(self.chromosome, TARGET):
                if gs != gt: fitness+=1
                return fitness

#Driver code
def main():
        global POPULATION_SIZE

        #current generation
        generation = 1
        found = False
        population = []

        #create intial population
        for _ in range(POPULATION_SIZE):
                gnome = Individual.create_gnome()
                population.append(Individual(gnome))

        while not found:
                population = sorted(population, key = lambda x:x.fitness)
                if population[0].fitness <=0:
                found = True
                break
                new_generation = []
                s = int((10*POPULATION_SIZE)/100)
                new_generation.extend(population[:s])

                s = int((90*POPULATION_SIZE)/100)
                for _ in range(s):
                parent1 = random.choice(population[:50])
                parent2 = random.choice(population[:50])
                child = parent1.mate(parent2)
                new_generation.append(child)

                population = new_generation
```

```
            print("Generation : {}\tString: {}\tFitness: {}".\
                  format(generation,"".join(population[0].chromosome),
                   population[0].fitness))
            generation += 1

        print("Generation : {}\tString: {}\tFitness: {}".\
              format(generation,"".join(population[0].chromosome),
               population[0].fitness))

if __name__ == '__main__':
  main()
```

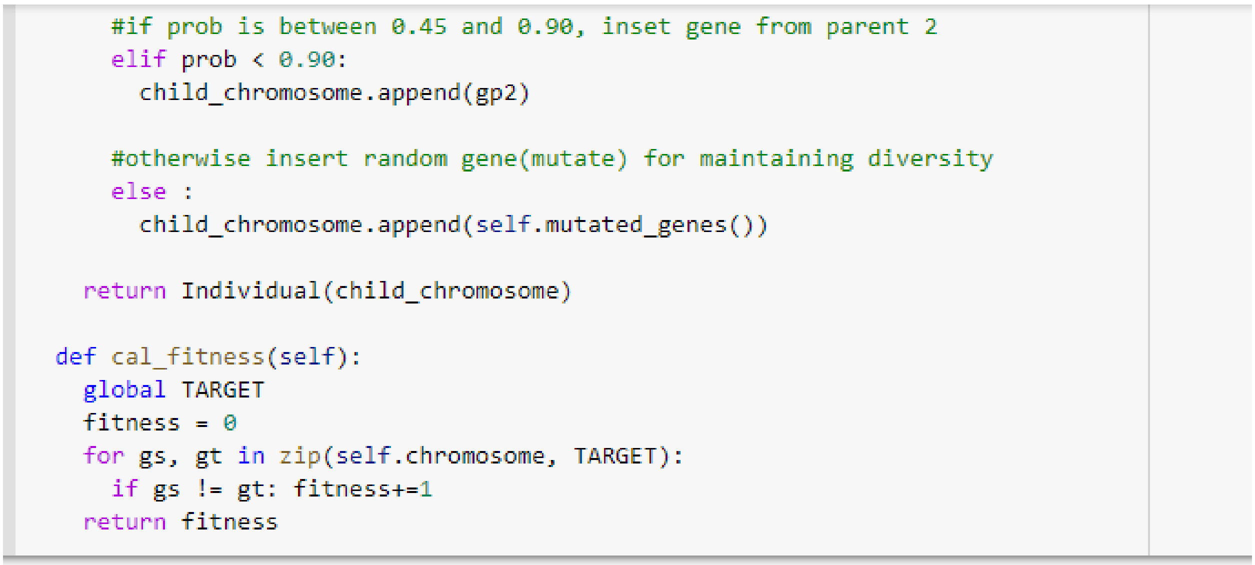
<u>Output</u> :



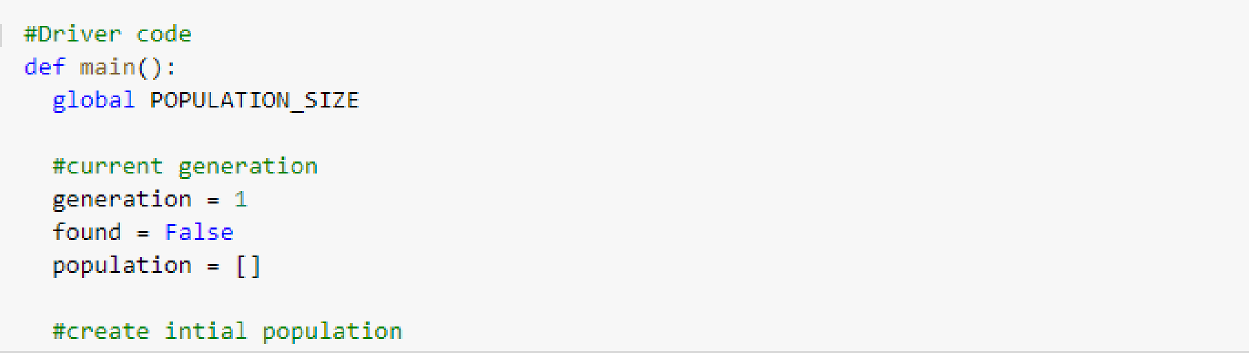
```
#random string using genetic alogorithm
import random

#Number of individual in each generation
POPULATION_SIZE = 100

#valid genes
GENES = '''abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890,.-;:_!"#%&/()=?@${[]}'''

#Target string to be generated
TARGET = "My Name Is Ujala"

class Individual(object):
  '''
  class representing individual in population
  '''
  def _init_(self, chromosome):
    self.chromosome = chromosome
    self.fitness = self.cal_fitness()

  @classmethod
  def mutated_genes(self):
    '''
    create random genes for mutation
    '''
    global GENES
```



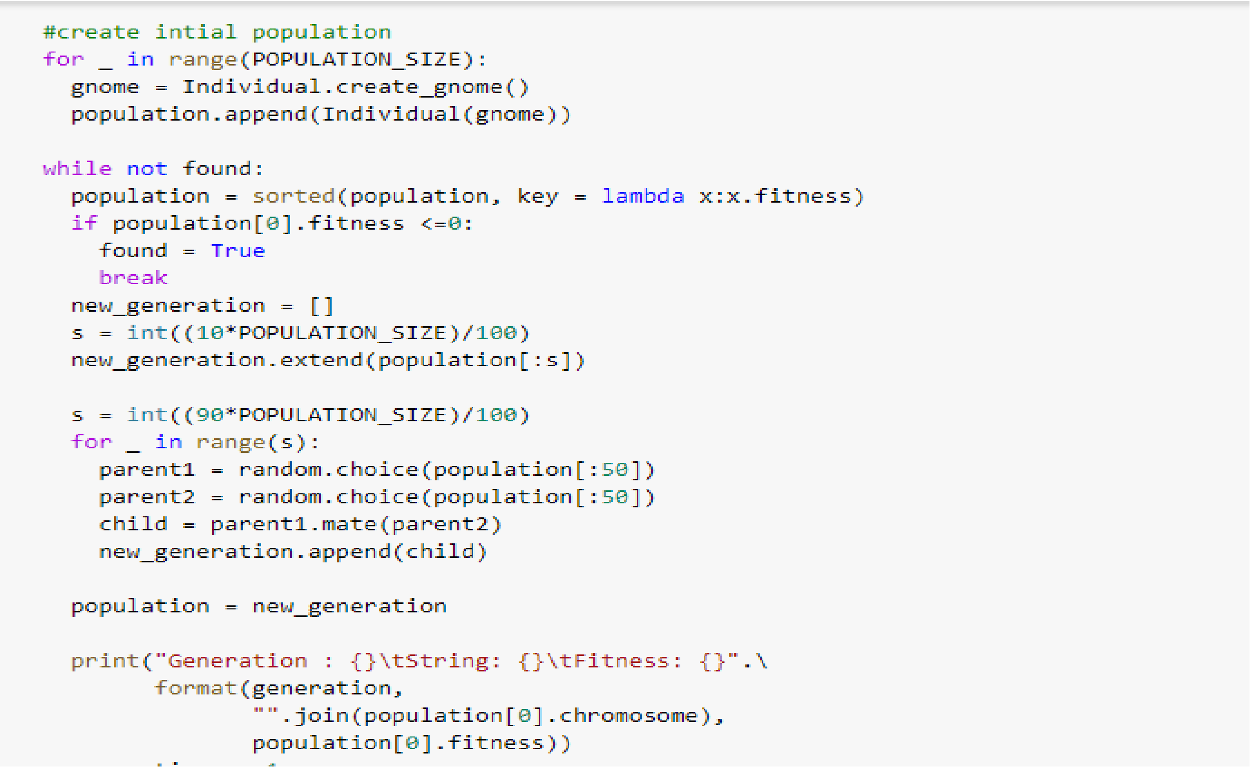
```
    global GENES
    gene = random.choice(GENES)
    return gene

  @classmethod
  def create_gnome(self):
    '''
    create chromosome or sting of genes
    '''
    global TARGET
    gnome_len = len(TARGET)
    return [self.mutated_genes() for _ in range(gnome_len)]

  def mate(self, par2):
    '''
    Perform mating and produce new offspring
    '''
    #chromosome for offspring
    child_chromosome = []
    for gp1, gp2 in zip(self.chromosome, par2.chromosome):

      #random probability
      prob = random.random()

      #if prob is less than 0.45, insert gene from parent 1
      if prob < 0.45:
        child_chromosome.append(gp1)
```



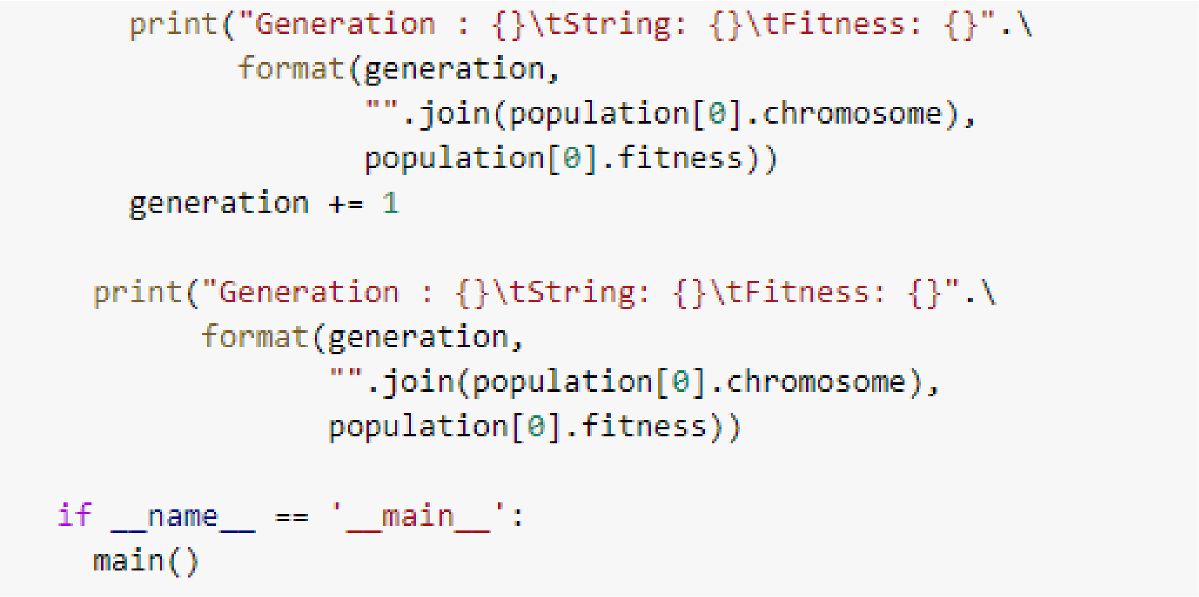
```
      #if prob is between 0.45 and 0.90, inset gene from parent 2
      elif prob < 0.90:
        child_chromosome.append(gp2)

      #otherwise insert random gene(mutate) for maintaining diversity
      else :
        child_chromosome.append(self.mutated_genes())

    return Individual(child_chromosome)

  def cal_fitness(self):
    global TARGET
    fitness = 0
    for gs, gt in zip(self.chromosome, TARGET):
      if gs != gt: fitness+=1
    return fitness
```



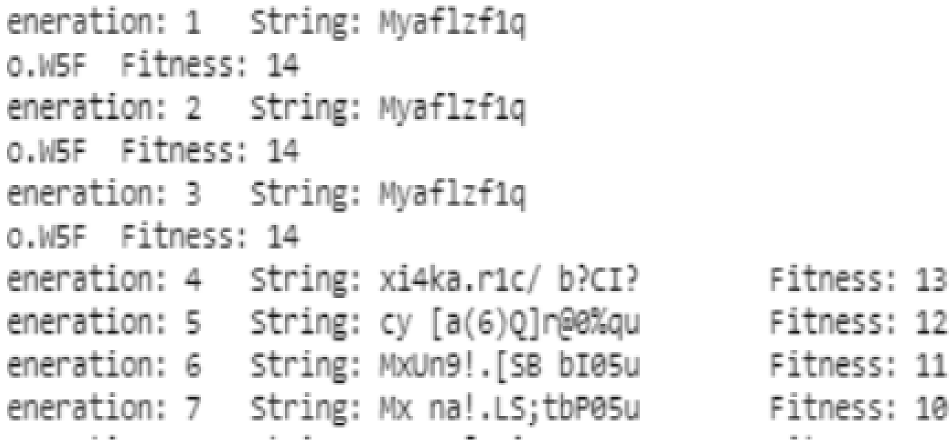
```
#Driver code
def main():
  global POPULATION_SIZE

  #current generation
  generation = 1
  found = False
  population = []

  #create intial population
```



```
#create intial population
for _ in range(POPULATION_SIZE):
  gnome = Individual.create_gnome()
  population.append(Individual(gnome))

while not found:
  population = sorted(population, key = lambda x:x.fitness)
  if population[0].fitness <=0:
    found = True
    break
  new_generation = []
  s = int((10*POPULATION_SIZE)/100)
  new_generation.extend(population[:s])

  s = int((90*POPULATION_SIZE)/100)
  for _ in range(s):
    parent1 = random.choice(population[:50])
    parent2 = random.choice(population[:50])
    child = parent1.mate(parent2)
    new_generation.append(child)

  population = new_generation

  print("Generation : {}\tString: {}\tFitness: {}".\
        format(generation,
               "".join(population[0].chromosome),
               population[0].fitness))
```



```
    print("Generation : {}\tString: {}\tFitness: {}".\
          format(generation,
                 "".join(population[0].chromosome),
                 population[0].fitness))
    generation += 1

  print("Generation : {}\tString: {}\tFitness: {}".\
        format(generation,
               "".join(population[0].chromosome),
               population[0].fitness))

if __name__ == '__main__':
  main()
```



```
eneration: 1   String: Myaflzf1q
o.W5F  Fitness: 14
eneration: 2   String: Myaflzf1q
o.W5F  Fitness: 14
eneration: 3   String: Myaflzf1q
o.W5F  Fitness: 14
eneration: 4   String: xi4ka.r1c/ b?CI?        Fitness: 13
eneration: 5   String: cy [a(6)Q]r@0%qu        Fitness: 12
eneration: 6   String: MxUn9!.[SB bI05u        Fitness: 11
eneration: 7   String: Mx na!.LS;tbP05u        Fitness: 10
```

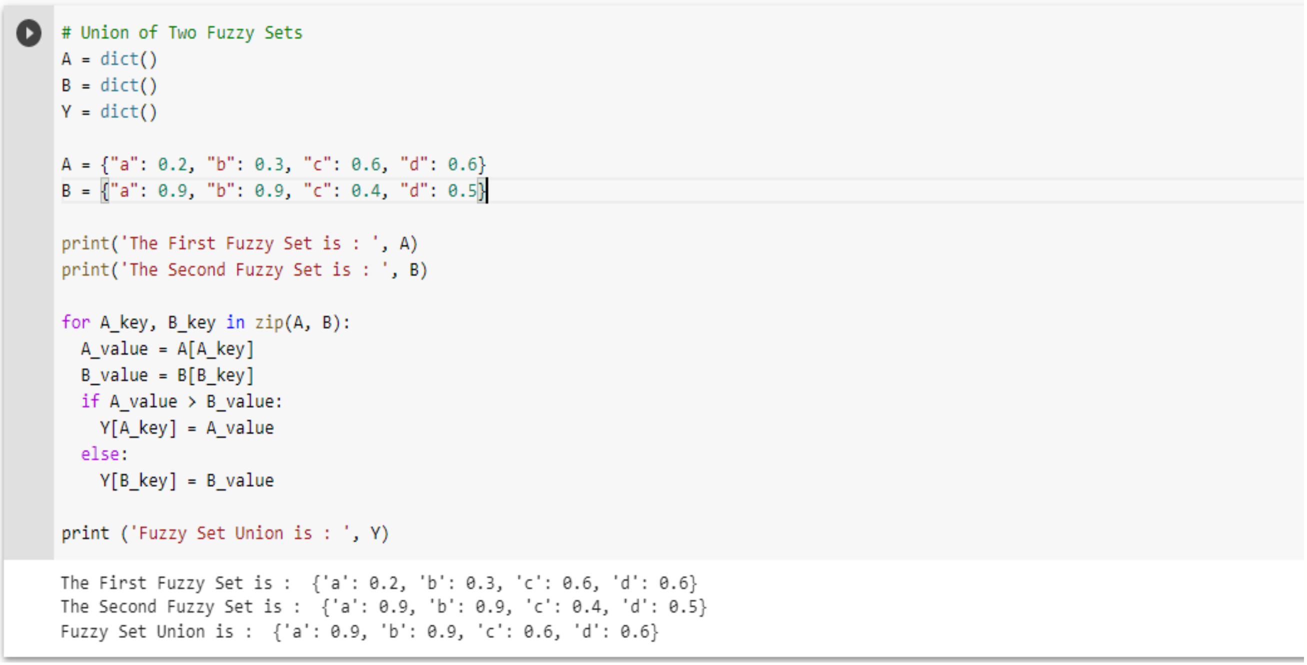
Design a Fuzzy based application using Python

Code :

1) Union of Two Fuzzy Sets :

```
# Union of Two Fuzzy Sets
A = dict()
B = dict()
Y = dict()

A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}
B = {"a": 0.9, "b": 0.9, "c": 0.4, "d": 0.5}

print('The First Fuzzy Set is : ', A)
print('The Second Fuzzy Set is : ', B)

for A_key, B_key in zip(A, B):
        A_value = A[A_key]
        B_value = B[B_key]

        if A_value > B_value:
                Y[A_key] = A_value
        else:
                Y[B_key] = B_value

print ('Fuzzy Set Union is : ', Y)
```

Output :



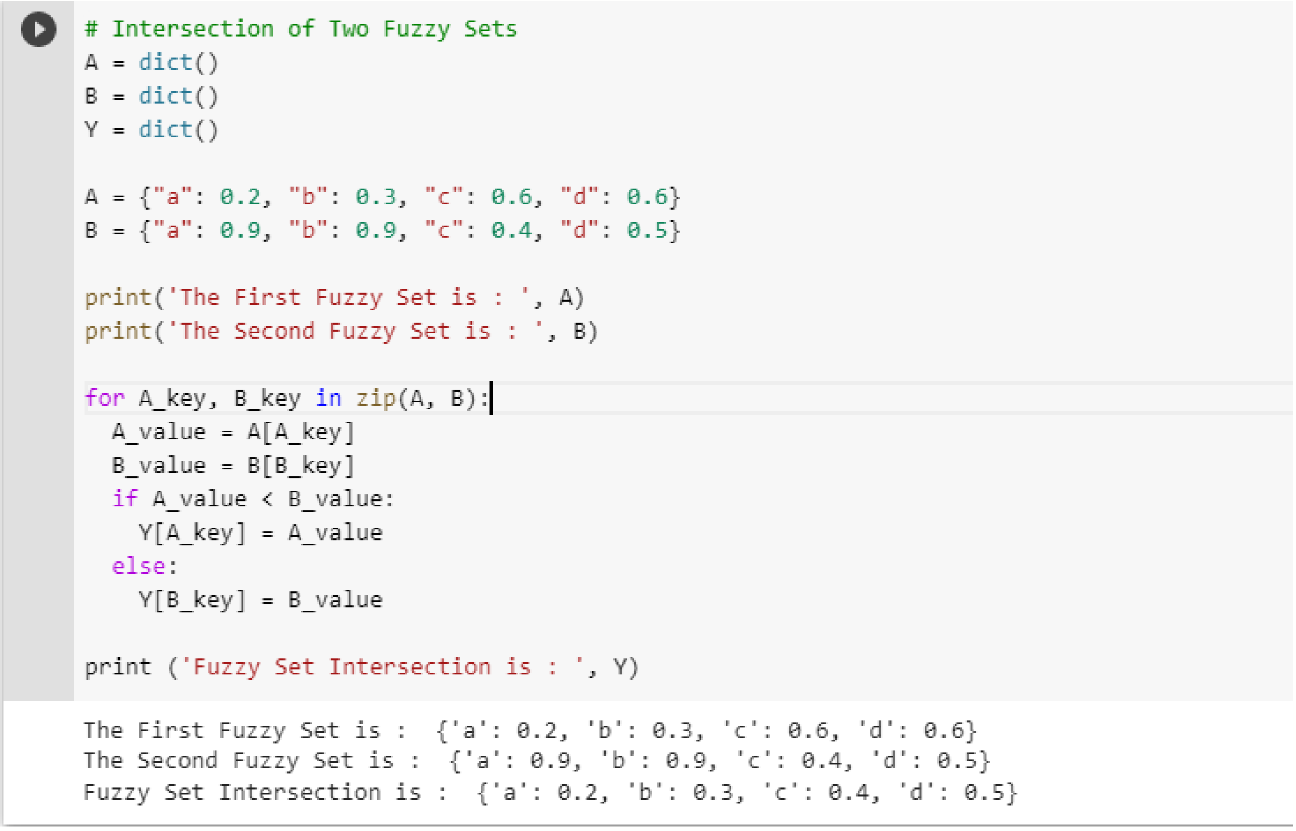
```
# Union of Two Fuzzy Sets
A = dict()
B = dict()
Y = dict()

A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}
B = {"a": 0.9, "b": 0.9, "c": 0.4, "d": 0.5}

print('The First Fuzzy Set is : ', A)
print('The Second Fuzzy Set is : ', B)

for A_key, B_key in zip(A, B):
  A_value = A[A_key]
  B_value = B[B_key]
  if A_value > B_value:
    Y[A_key] = A_value
  else:
    Y[B_key] = B_value

print ('Fuzzy Set Union is : ', Y)
```

```
The First Fuzzy Set is :  {'a': 0.2, 'b': 0.3, 'c': 0.6, 'd': 0.6}
The Second Fuzzy Set is :  {'a': 0.9, 'b': 0.9, 'c': 0.4, 'd': 0.5}
Fuzzy Set Union is :  {'a': 0.9, 'b': 0.9, 'c': 0.6, 'd': 0.6}
```

2) Intersection of Two Fuzzy Sets :

```
# Intersection of Two Fuzzy Sets
A = dict()
B = dict()
Y = dict()

A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}
B = {"a": 0.9, "b": 0.9, "c": 0.4, "d": 0.5}

print('The First Fuzzy Set is : ', A)
print('The Second Fuzzy Set is : ', B)

for A_key, B_key in zip(A, B):
        A_value = A[A_key]
        B_value = B[B_key]

        if A_value < B_value:
                Y[A_key] = A_value
        else:
                Y[B_key] = B_value

print ('Fuzzy Set Intersection is : ', Y)
```

Output :



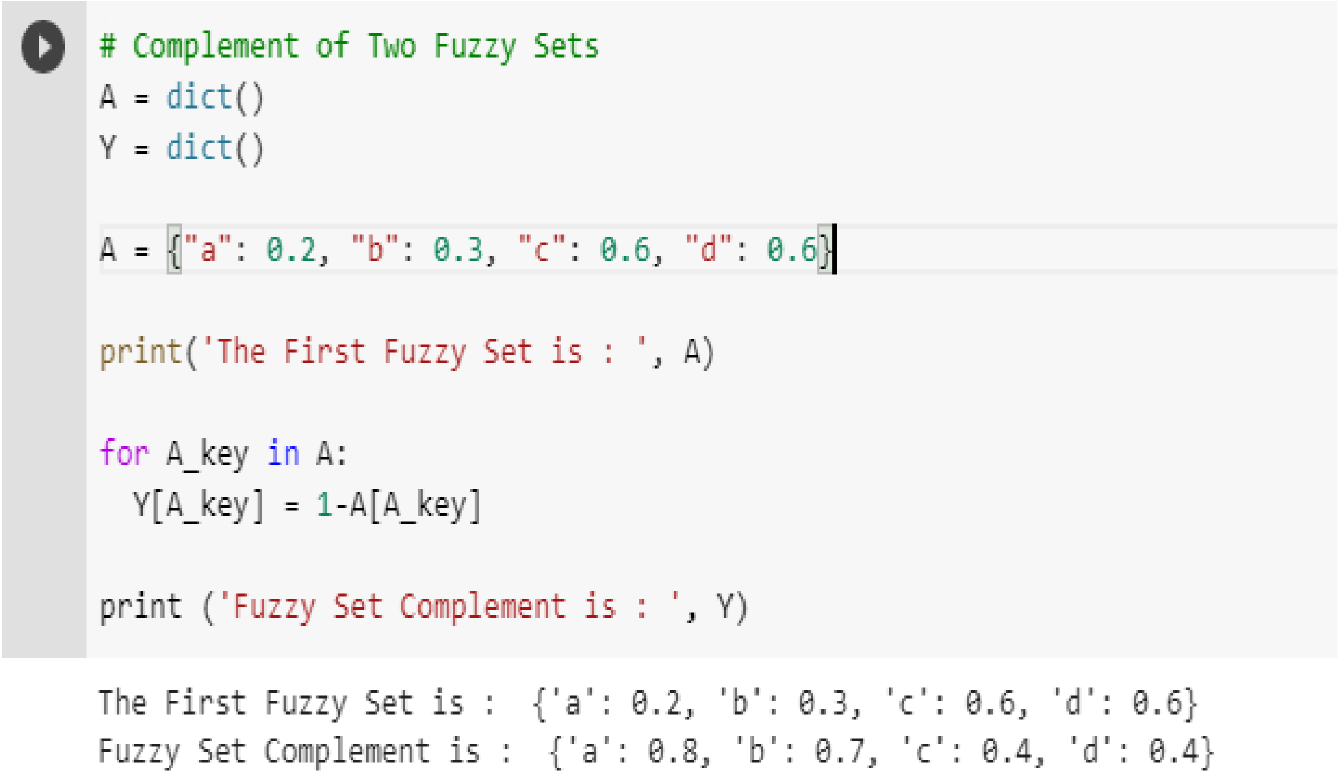
```
# Intersection of Two Fuzzy Sets
A = dict()
B = dict()
Y = dict()

A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}
B = {"a": 0.9, "b": 0.9, "c": 0.4, "d": 0.5}

print('The First Fuzzy Set is : ', A)
print('The Second Fuzzy Set is : ', B)

for A_key, B_key in zip(A, B):
  A_value = A[A_key]
  B_value = B[B_key]
  if A_value < B_value:
    Y[A_key] = A_value
  else:
    Y[B_key] = B_value

print ('Fuzzy Set Intersection is : ', Y)
```

```
The First Fuzzy Set is :  {'a': 0.2, 'b': 0.3, 'c': 0.6, 'd': 0.6}
The Second Fuzzy Set is :  {'a': 0.9, 'b': 0.9, 'c': 0.4, 'd': 0.5}
Fuzzy Set Intersection is :  {'a': 0.2, 'b': 0.3, 'c': 0.4, 'd': 0.5}
```

3) Complement of Two Fuzzy Sets :

```
# Complement of Two Fuzzy Sets
A = dict()
Y = dict()

A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}

print('The First Fuzzy Set is : ', A)

for A_key in A:
        Y[A_key] = 1-A[A_key]

print ('Fuzzy Set Complement is : ', Y)
```

Output :



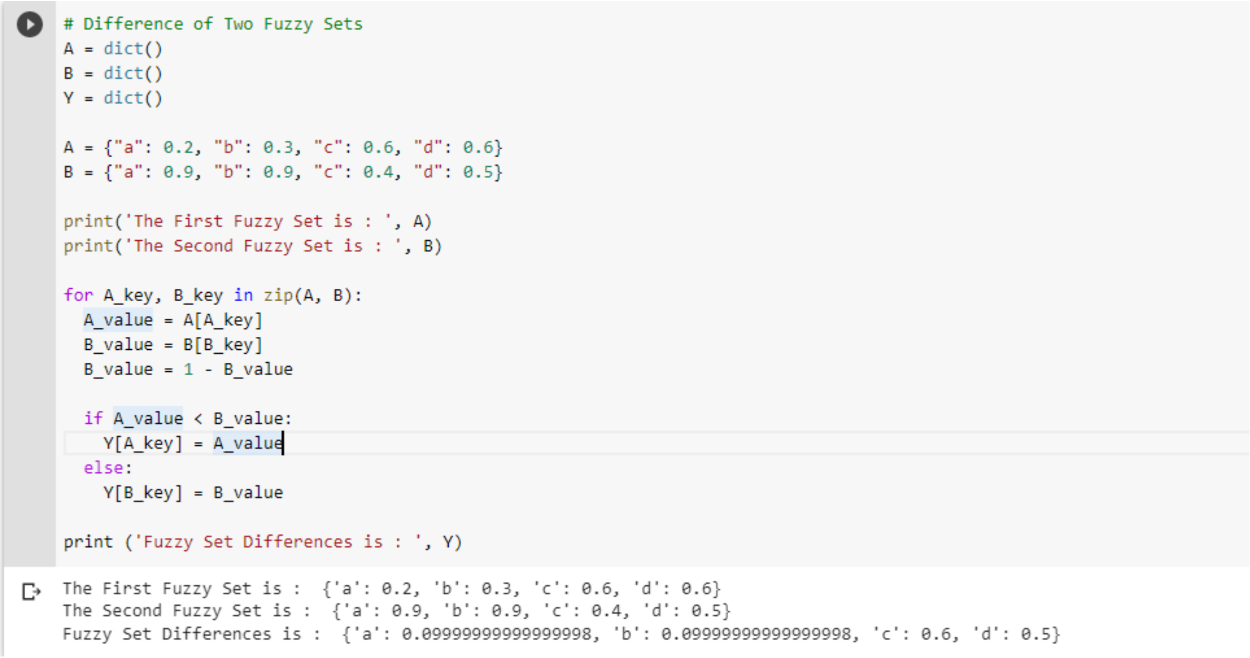
```
# Complement of Two Fuzzy Sets
A = dict()
Y = dict()

A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}

print('The First Fuzzy Set is : ', A)

for A_key in A:
  Y[A_key] = 1-A[A_key]

print ('Fuzzy Set Complement is : ', Y)
```

```
The First Fuzzy Set is :  {'a': 0.2, 'b': 0.3, 'c': 0.6, 'd': 0.6}
Fuzzy Set Complement is :  {'a': 0.8, 'b': 0.7, 'c': 0.4, 'd': 0.4}
```

4) Difference of Two Fuzzy Sets

```
# Difference of Two Fuzzy Sets
A = dict()
B = dict()
Y = dict()

A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}
B = {"a": 0.9, "b": 0.9, "c": 0.4, "d": 0.5}

print('The First Fuzzy Set is : ', A)
print('The Second Fuzzy Set is : ', B)

for A_key, B_key in zip(A, B):
        A_value = A[A_key]
        B_value = B[B_key]
        B_value = 1 - B_value

        if A_value < B_value:
                Y[A_key] = A_value
        else:
                Y[B_key] = B_value

print ('Fuzzy Set Differences is : ', Y)
```

Output :



```
# Difference of Two Fuzzy Sets
A = dict()
B = dict()
Y = dict()

A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}
B = {"a": 0.9, "b": 0.9, "c": 0.4, "d": 0.5}

print('The First Fuzzy Set is : ', A)
print('The Second Fuzzy Set is : ', B)

for A_key, B_key in zip(A, B):
  A_value = A[A_key]
  B_value = B[B_key]
  B_value = 1 - B_value

  if A_value < B_value:
    Y[A_key] = A_value
  else:
    Y[B_key] = B_value

print ('Fuzzy Set Differences is : ', Y)
```

```
The First Fuzzy Set is :  {'a': 0.2, 'b': 0.3, 'c': 0.6, 'd': 0.6}
The Second Fuzzy Set is :  {'a': 0.9, 'b': 0.9, 'c': 0.4, 'd': 0.5}
Fuzzy Set Differences is :  {'a': 0.09999999999999998, 'b': 0.09999999999999998, 'c': 0.6, 'd': 0.5}
```

Write an application to implement supervised and unsupervised learning model

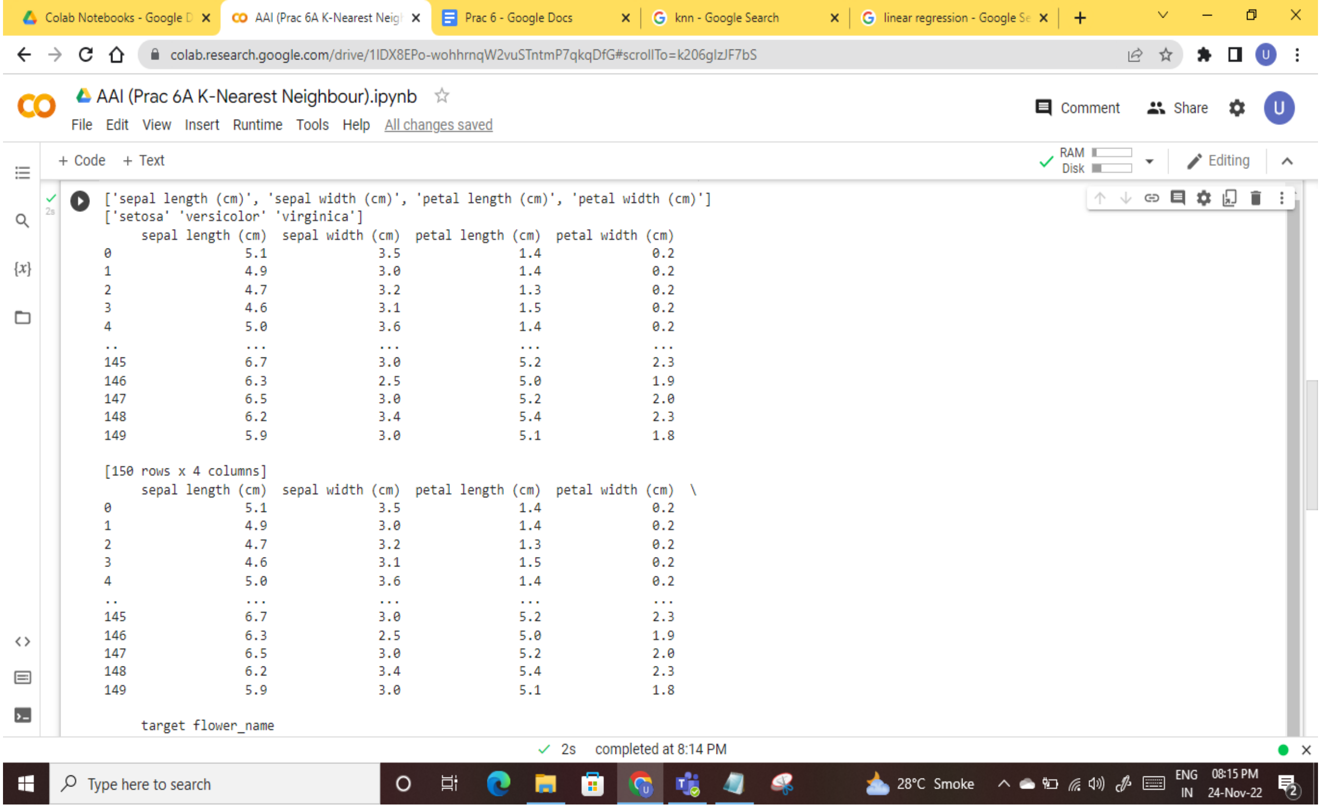
Code :

K-Nearest Neighbour :

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

iris = load_iris()
print(iris.feature_names)
print(iris.target_names)

df = pd.DataFrame(iris.data,columns=iris.feature_names)
print(df)

df['target'] = iris.target
df['flower_name'] =df.target.apply(lambda x: iris.target_names[x])
print(df)

df0 = df[:50]           # setosa
df1 = df[50:100]         # versicolor
df2 = df[100:]          # virginica

# Sepal length vs Sepal Width
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.scatter(df0['sepal length (cm)'], df0['sepal width (cm)'],color="green",marker='+')
plt.scatter(df1['sepal length (cm)'], df1['sepal width (cm)'],color="blue",marker='.')
```

```
# Petal length vs Pepal Width
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.scatter(df0['petal length (cm)'], df0['petal width (cm)'],color="green",marker='+')
plt.scatter(df1['petal length (cm)'], df1['petal width (cm)'],color="blue",marker='.')

X = df.drop(['target','flower_name'], axis='columns')
y = df.target
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2)

knn = KNeighborsClassifier(n_neighbors=10)
knn.fit(X_train, y_train)
knn.score(X_test, y_test)
```

<u>Output</u> :

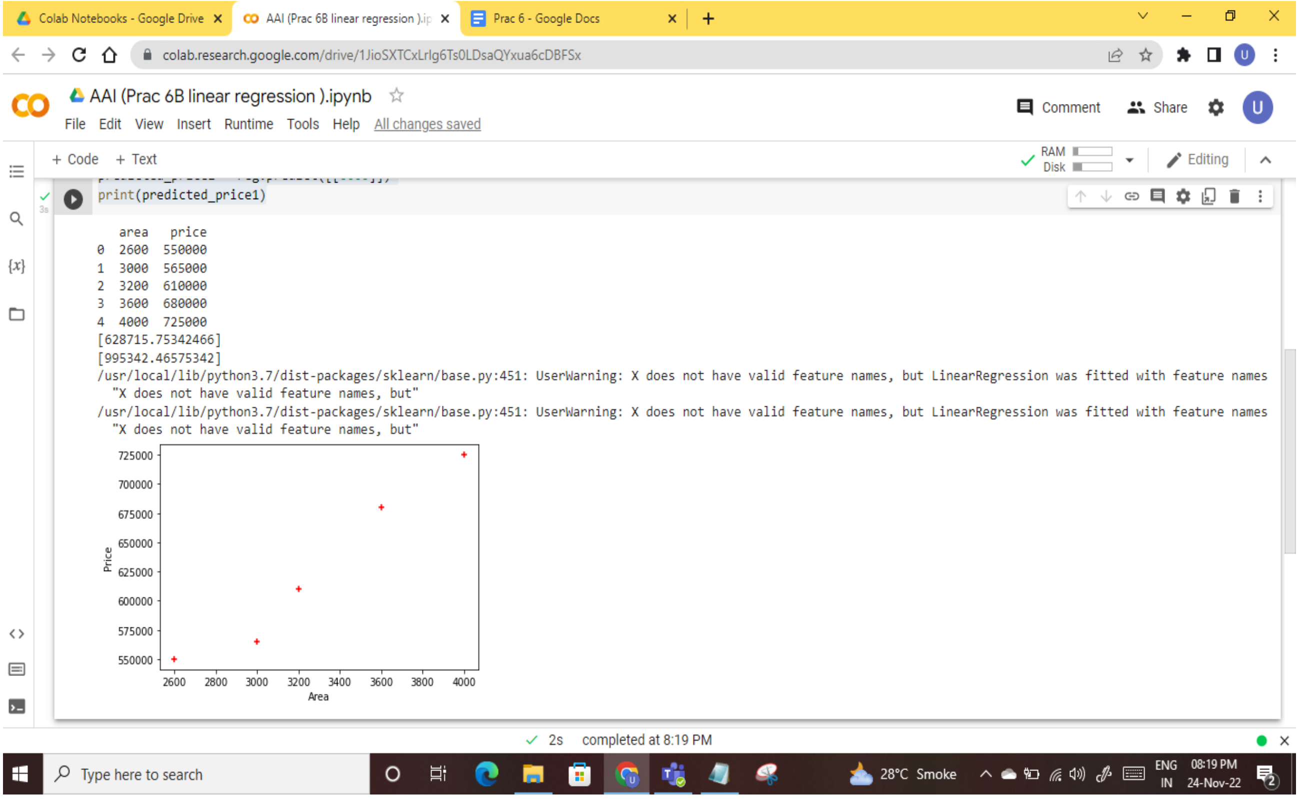

Colab Notebooks - Google D
AAI (Prac 6A K-Nearest Neig
Prac 6 - Google Docs
knn - Google Search
linear regression - Google Se
colab.research.google.com/drive/1IDX8EPo-wohhrnqW2vuSTntmP7qkqDfG#scrollTo=k206glzJF7bS
AAI (Prac 6A K-Nearest Neighbour).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
2s completed at 8:14 PM
Type here to search
28°C Smoke
ENG IN 08:15 PM 24-Nov-22

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
['setosa' 'versicolor' 'virginica']
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0                  5.1               3.5                1.4               0.2
1                  4.9               3.0                1.4               0.2
2                  4.7               3.2                1.3               0.2
3                  4.6               3.1                1.5               0.2
4                  5.0               3.6                1.4               0.2
..                 ...               ...                ...               ...
145                6.7               3.0                5.2               2.3
146                6.3               2.5                5.0               1.9
147                6.5               3.0                5.2               2.0
148                6.2               3.4                5.4               2.3
149                5.9               3.0                5.1               1.8

[150 rows x 4 columns]
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                  5.1               3.5                1.4               0.2
1                  4.9               3.0                1.4               0.2
2                  4.7               3.2                1.3               0.2
3                  4.6               3.1                1.5               0.2
4                  5.0               3.6                1.4               0.2
..                 ...               ...                ...               ...
145                6.7               3.0                5.2               2.3
146                6.3               2.5                5.0               1.9
147                6.5               3.0                5.2               2.0
148                6.2               3.4                5.4               2.3
149                5.9               3.0                5.1               1.8

     target flower_name
```

linear regression :

```
import pandas as pd
import numpy as np
from sklearn import linear_model
import matplotlib.pyplot as plt

df = pd.read_csv('/content/HousePrices.csv')
print(df)

plt.xlabel('Area')
plt.ylabel('Price')
plt.scatter(df.area,df.price,color='red',marker='+')

new_df = df.drop('price',axis='columns')
price = df.price

# Create linear regression object
reg = linear_model.LinearRegression()
reg.fit(new_df,price)

predicted_price = reg.predict([[3300]])
print(predicted_price)

predicted_price1 = reg.predict([[6000]])
print(predicted_price1)
```

Output :


Colab Notebooks - Google Drive
AAI (Prac 6B linear regression).ip
Prac 6 - Google Docs
colab.research.google.com/drive/1JioSXTCxLrlg6Ts0LDsaQYxua6cDBFSx
AAI (Prac 6B linear regression).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
Price
Area
2s completed at 8:19 PM
Type here to search
28°C Smoke
ENG IN
08:19 PM 24-Nov-22

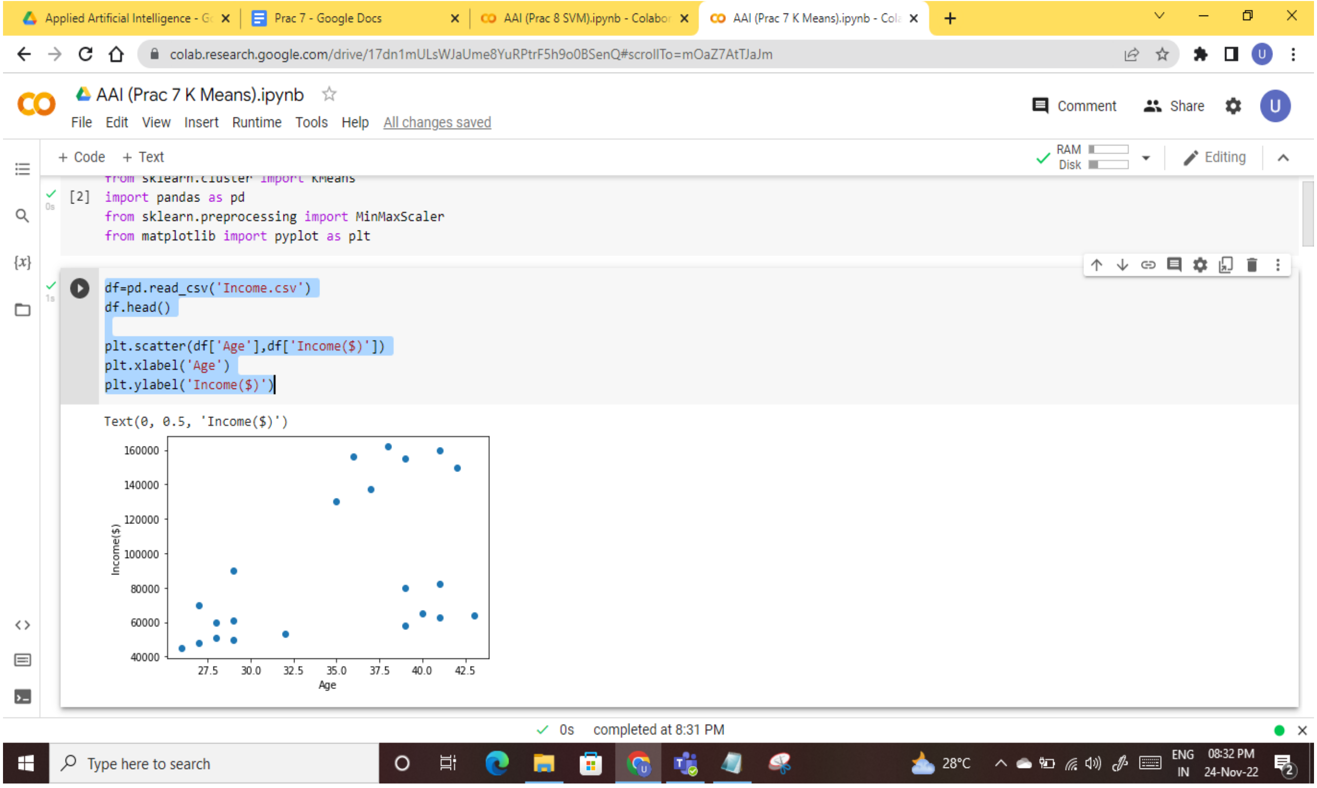
```
print(predicted_price1)
```

```
   area   price
0  2600  550000
1  3000  565000
2  3200  610000
3  3600  680000
4  4000  725000
[628715.75342466]
[995342.46575342]
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  "X does not have valid feature names, but"
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  "X does not have valid feature names, but"
```

Write an application to implement a clustering algorithm (K Means)

Code :

```
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot as plt

df=pd.read_csv('Income.csv')
df.head()

plt.scatter(df['Age'],df['Income($)'])
plt.xlabel('Age')
plt.ylabel('Income($)')
```



Applied Artificial Intelligence - G
Prac 7 - Google Docs
AAI (Prac 8 SVM).ipynb - Colabor
AAI (Prac 7 K Means).ipynb - Col
colab.research.google.com/drive/17dn1mULsWJaUme8YuRPtrF5h9o0BSenQ#scrollTo=mOaZ7AtTJaJm
AAI (Prac 7 K Means).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing

```
[2] from sklearn.cluster import KMeans
    import pandas as pd
    from sklearn.preprocessing import MinMaxScaler
    from matplotlib import pyplot as plt

    df=pd.read_csv('Income.csv')
    df.head()

    plt.scatter(df['Age'],df['Income($)'])
    plt.xlabel('Age')
    plt.ylabel('Income($)')
```

```
Text(0, 0.5, 'Income($)')
```

160000
140000
120000
100000
80000
60000
40000
Income($)
27.5
30.0
32.5
35.0
37.5
40.0
42.5
Age
0s completed at 8:31 PM
Type here to search
28°C
ENG IN
08:32 PM
24-Nov-22

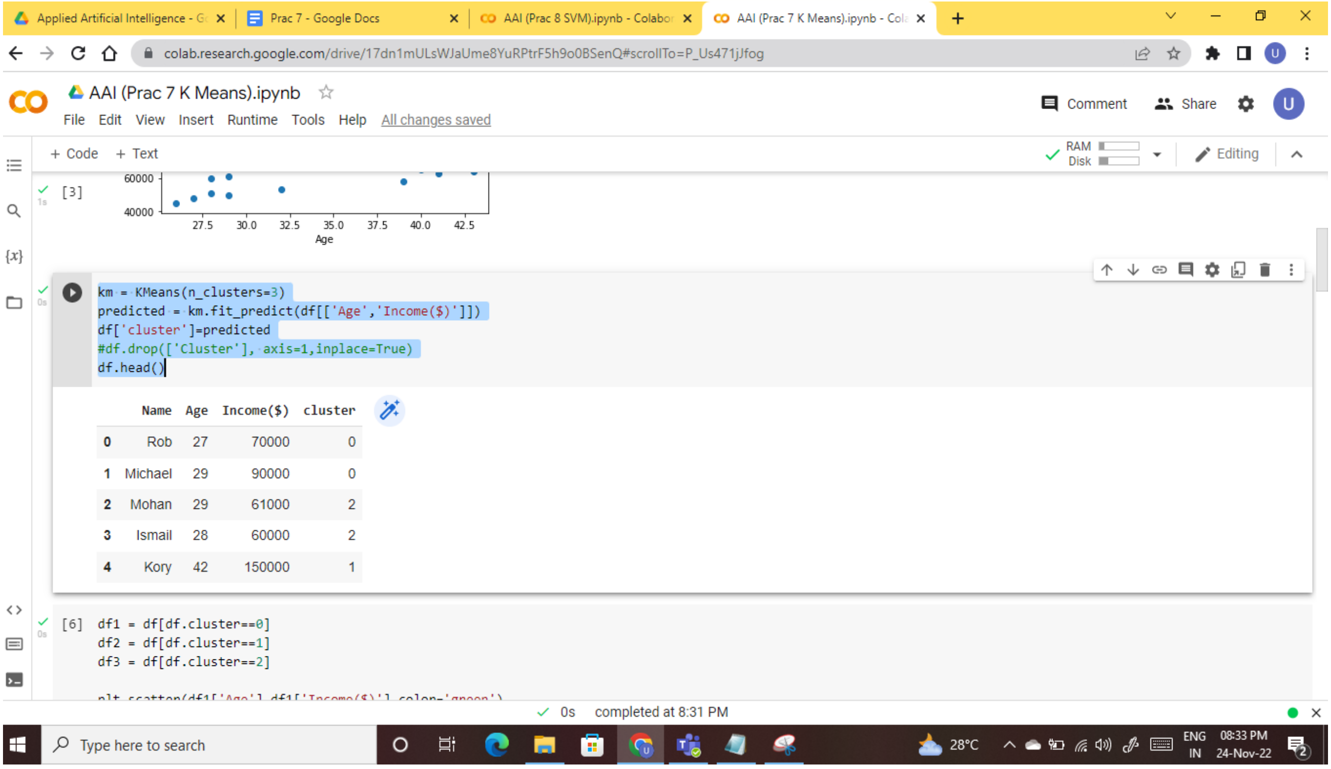
km = KMeans(n_clusters=3)

predicted = km.fit_predict(df[['Age','Income($)']])

df['cluster']=predicted

#df.drop(['Cluster'], axis=1,inplace=True)

df.head()


Applied Artificial Intelligence - G
Prac 7 - Google Docs
AAI (Prac 8 SVM).ipynb - Colabo
AAI (Prac 7 K Means).ipynb - Col
colab.research.google.com/drive/17dn1mULsWJaUme8YuRPtrF5h9o0BSenQ#scrollTo=P_Us471jJfog
AAI (Prac 7 K Means).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
[3]
60000
40000
27.5 30.0 32.5 35.0 37.5 40.0 42.5
Age
0s completed at 8:31 PM
Type here to search
28°C
ENG IN 08:33 PM 24-Nov-22

```
km = KMeans(n_clusters=3)
predicted = km.fit_predict(df[['Age','Income($)']])
df['cluster']=predicted
#df.drop(['Cluster'], axis=1,inplace=True)
df.head()
```

	Name	Age	Income($)	cluster
0	Rob	27	70000	0
1	Michael	29	90000	0
2	Mohan	29	61000	2
3	Ismail	28	60000	2
4	Kory	42	150000	1

```
[6] df1 = df[df.cluster==0]
    df2 = df[df.cluster==1]
    df3 = df[df.cluster==2]
```

df1 = df[df.cluster==0]

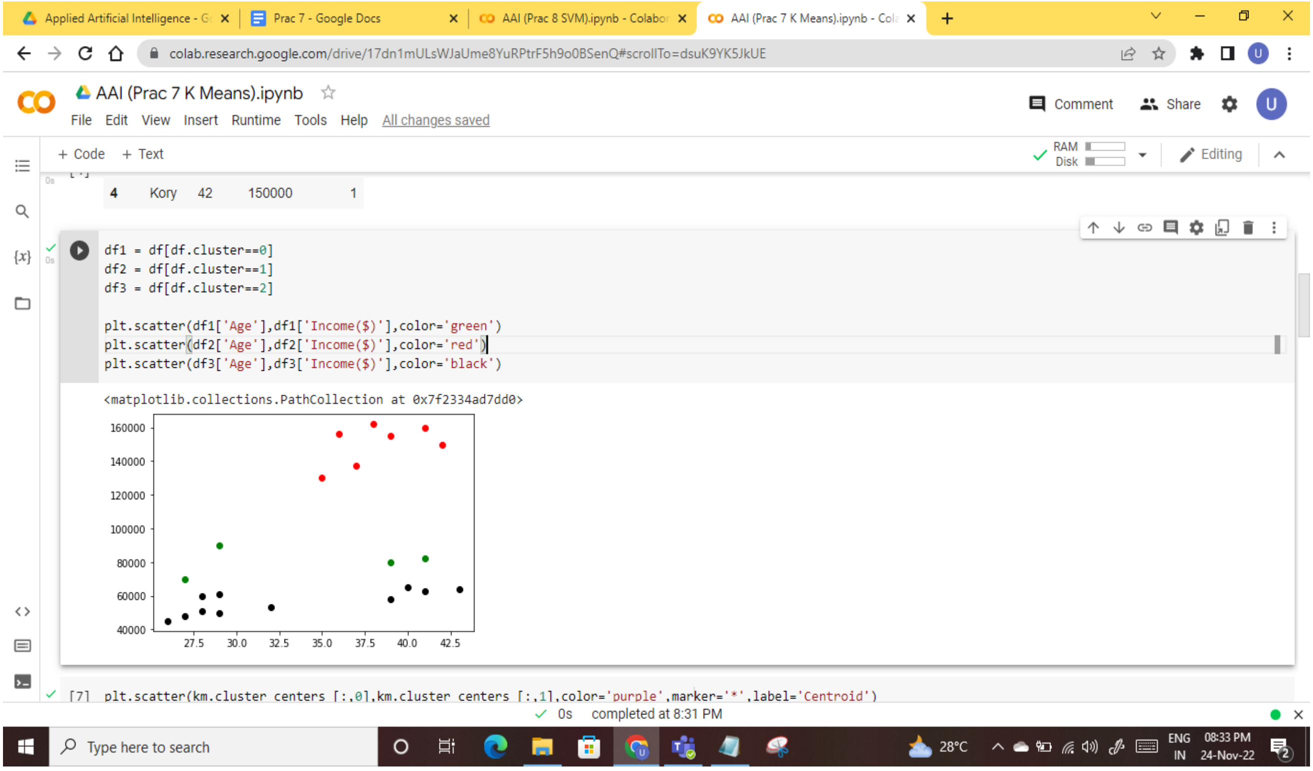
df2 = df[df.cluster==1]

df3 = df[df.cluster==2]

plt.scatter(df1['Age'],df1['Income($)'],color='green')

plt.scatter(df2['Age'],df2['Income($)'],color='red')

plt.scatter(df3['Age'],df3['Income($)'],color='black')


Applied Artificial Intelligence - G
Prac 7 - Google Docs
AAI (Prac 8 SVM).ipynb - Colabor
AAI (Prac 7 K Means).ipynb - Cola
colab.research.google.com/drive/17dn1mULsWJaUme8YuRPtrF5h9o0BSenQ#scrollTo=dsuK9YK5JkUE
AAI (Prac 7 K Means).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
160000
140000
120000
100000
80000
60000
40000
27.5
30.0
32.5
35.0
37.5
40.0
42.5
0s completed at 8:31 PM
Type here to search
28°C
ENG IN
08:33 PM
24-Nov-22

```
4    Kory    42    150000    1
```

```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]

plt.scatter(df1['Age'],df1['Income($)'],color='green')
plt.scatter(df2['Age'],df2['Income($)'],color='red')
plt.scatter(df3['Age'],df3['Income($)'],color='black')
```

```
<matplotlib.collections.PathCollection at 0x7f2334ad7dd0>
```

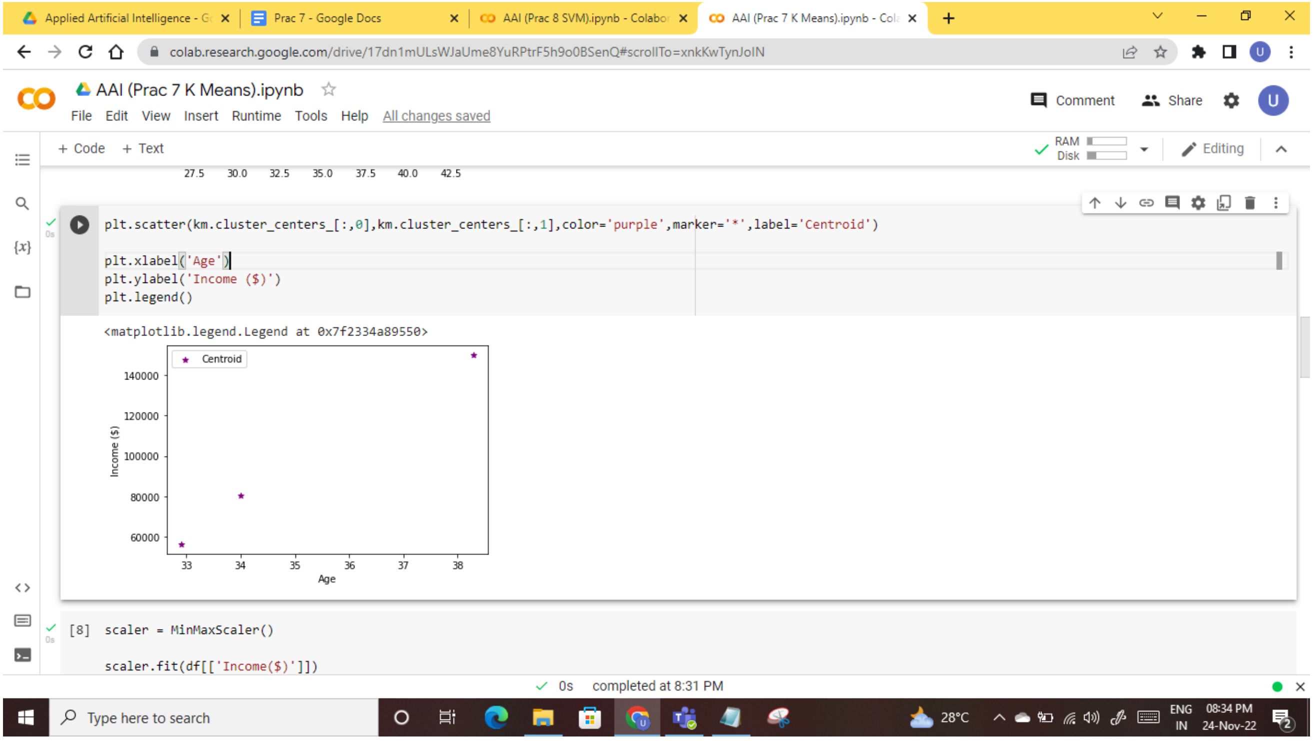
```
[7]  plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',label='Centroid')
```

plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',label='Centroid')

plt.xlabel('Age')

plt.ylabel('Income ($)')

plt.legend()


Applied Artificial Intelligence - G
Prac 7 - Google Docs
AAI (Prac 8 SVM).ipynb - Colabor
AAI (Prac 7 K Means).ipynb - Cola
colab.research.google.com/drive/17dn1mULsWJaUme8YuRPtrF5h9o0BSenQ#scrollTo=xnkKwTynJoIN
AAI (Prac 7 K Means).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
27.5 30.0 32.5 35.0 37.5 40.0 42.5
Centroid
140000
120000
100000
80000
60000
Income ($)
33
34
35
36
37
38
Age
0s completed at 8:31 PM
Type here to search
28°C
ENG IN
08:34 PM
24-Nov-22

```
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',label='Centroid')

plt.xlabel('Age')
plt.ylabel('Income ($)')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f2334a89550>
```

```
[8]  scaler = MinMaxScaler()

     scaler.fit(df[['Income($)']])
```

scaler = MinMaxScaler()

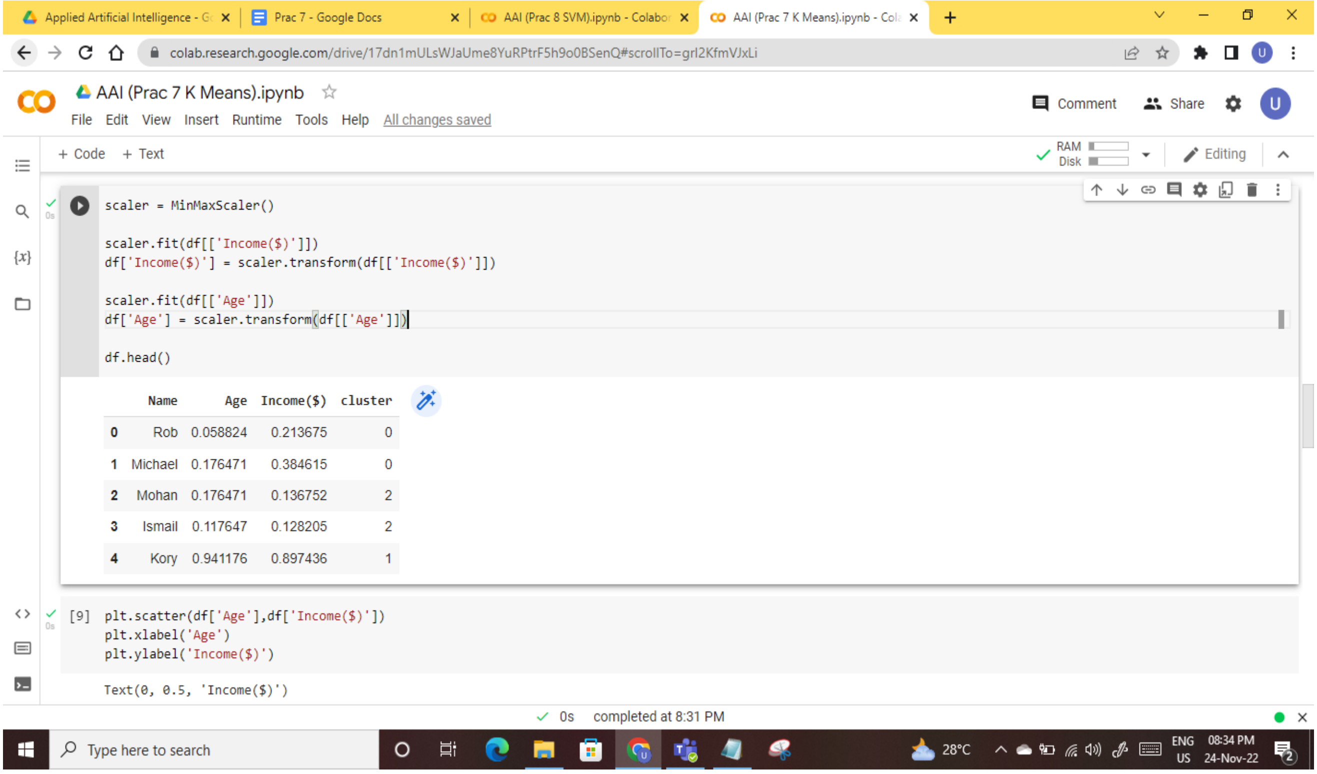
scaler.fit(df[['Income($)']])

df['Income($)'] = scaler.transform(df[['Income($)']])

scaler.fit(df[['Age']])

df['Age'] = scaler.transform(df[['Age']])

df.head()


Applied Artificial Intelligence - G
Prac 7 - Google Docs
AAI (Prac 8 SVM).ipynb - Colabor
AAI (Prac 7 K Means).ipynb - Col
colab.research.google.com/drive/17dn1mULsWJaUme8YuRPtrF5h9o0BSenQ#scrollTo=grI2KfmVJxLi
AAI (Prac 7 K Means).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing

```
scaler = MinMaxScaler()

scaler.fit(df[['Income($)']])
df['Income($)'] = scaler.transform(df[['Income($)']])

scaler.fit(df[['Age']])
df['Age'] = scaler.transform(df[['Age']])

df.head()
```

	Name	Age	Income($)	cluster
0	Rob	0.058824	0.213675	0
1	Michael	0.176471	0.384615	0
2	Mohan	0.176471	0.136752	2
3	Ismail	0.117647	0.128205	2
4	Kory	0.941176	0.897436	1

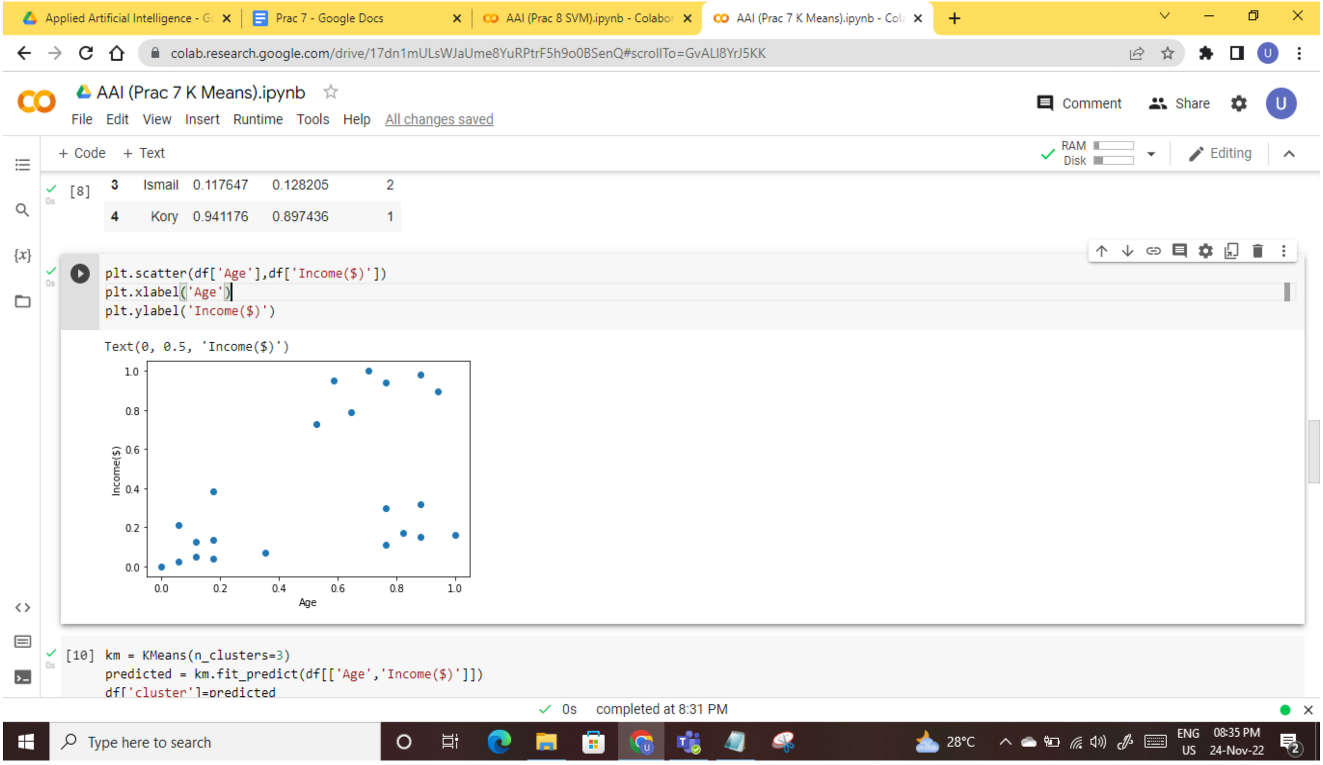
```
[9] plt.scatter(df['Age'],df['Income($)'])
    plt.xlabel('Age')
    plt.ylabel('Income($)')

Text(0, 0.5, 'Income($)')
```

✓ 0s completed at 8:31 PM
Type here to search
28°C
ENG US
08:34 PM 24-Nov-22

plt.scatter(df['Age'],df['Income($)'])

plt.xlabel('Age')

plt.ylabel('Income($)')



	Name	Age	Income($)	cluster
3	Ismail	0.117647	0.128205	2
4	Kory	0.941176	0.897436	1

```
plt.scatter(df['Age'],df['Income($)'])
plt.xlabel('Age')
plt.ylabel('Income($)')
```

```
Text(0, 0.5, 'Income($)')
```

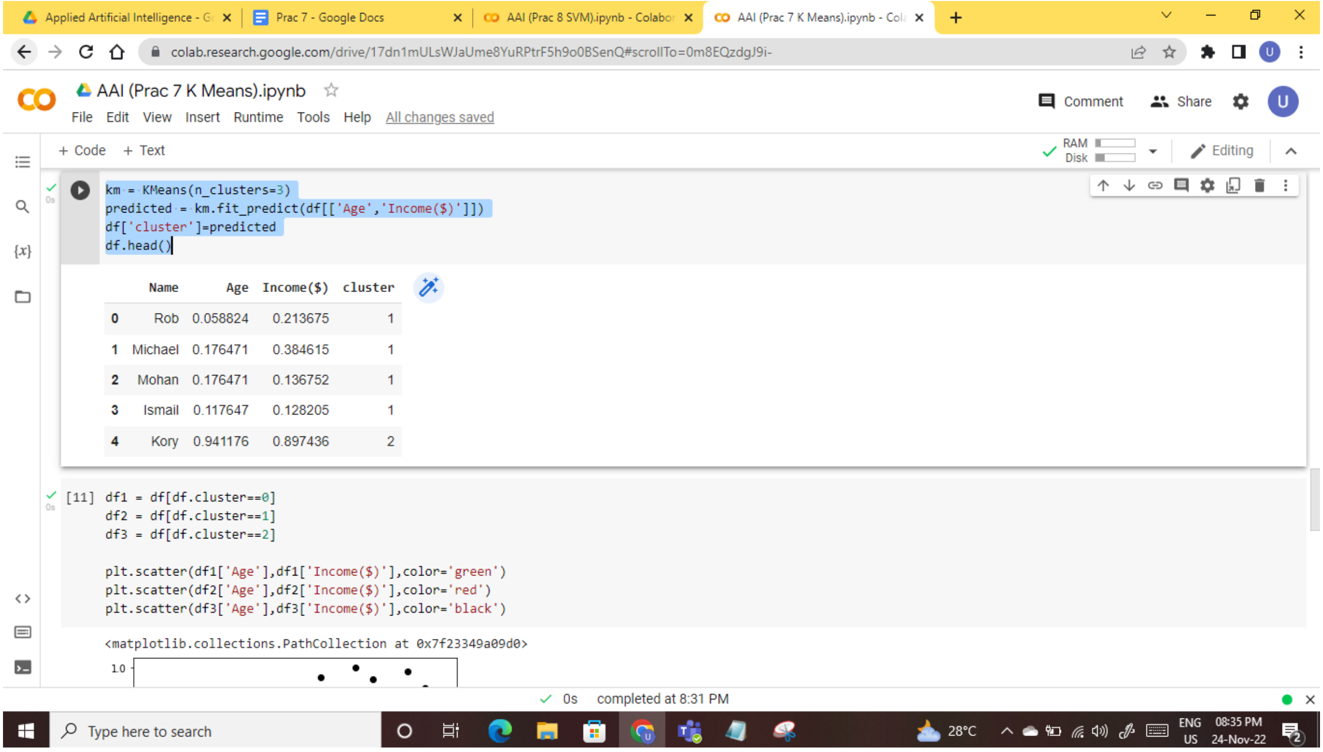
```
[10] km = KMeans(n_clusters=3)
     predicted = km.fit_predict(df[['Age','Income($)']])
     df['cluster']=predicted
```

km = KMeans(n_clusters=3)

predicted = km.fit_predict(df[['Age','Income($)']])

df['cluster']=predicted

df.head()


AAI (Prac 7 K Means).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk Editing
completed at 8:31 PM
Type here to search
28°C ENG US 08:35 PM 24-Nov-22

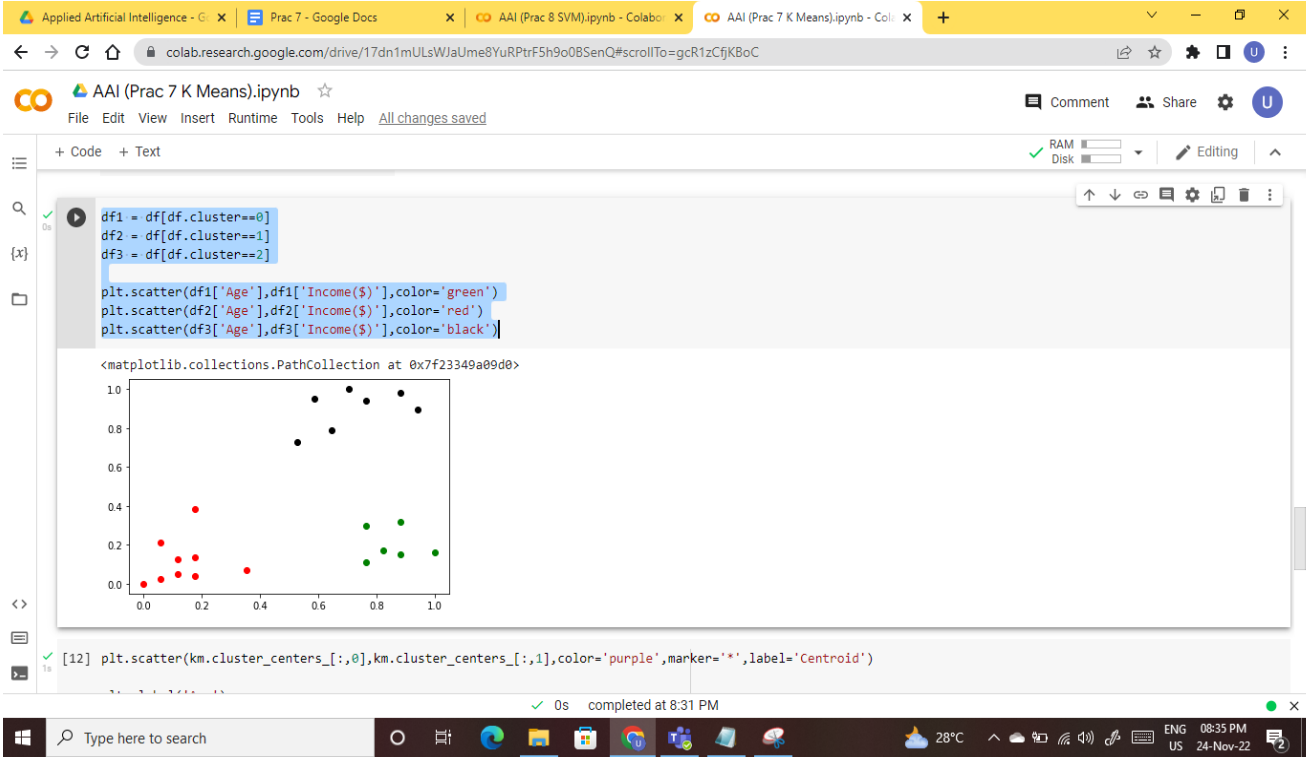
```
km = KMeans(n_clusters=3)
predicted = km.fit_predict(df[['Age','Income($)']])
df['cluster']=predicted
df.head()
```

	Name	Age	Income($)	cluster
0	Rob	0.058824	0.213675	1
1	Michael	0.176471	0.384615	1
2	Mohan	0.176471	0.136752	1
3	Ismail	0.117647	0.128205	1
4	Kory	0.941176	0.897436	2

```
[11] df1 = df[df.cluster==0]
     df2 = df[df.cluster==1]
     df3 = df[df.cluster==2]

     plt.scatter(df1['Age'],df1['Income($)'],color='green')
     plt.scatter(df2['Age'],df2['Income($)'],color='red')
     plt.scatter(df3['Age'],df3['Income($)'],color='black')
```

```
<matplotlib.collections.PathCollection at 0x7f23349a09d0>
```

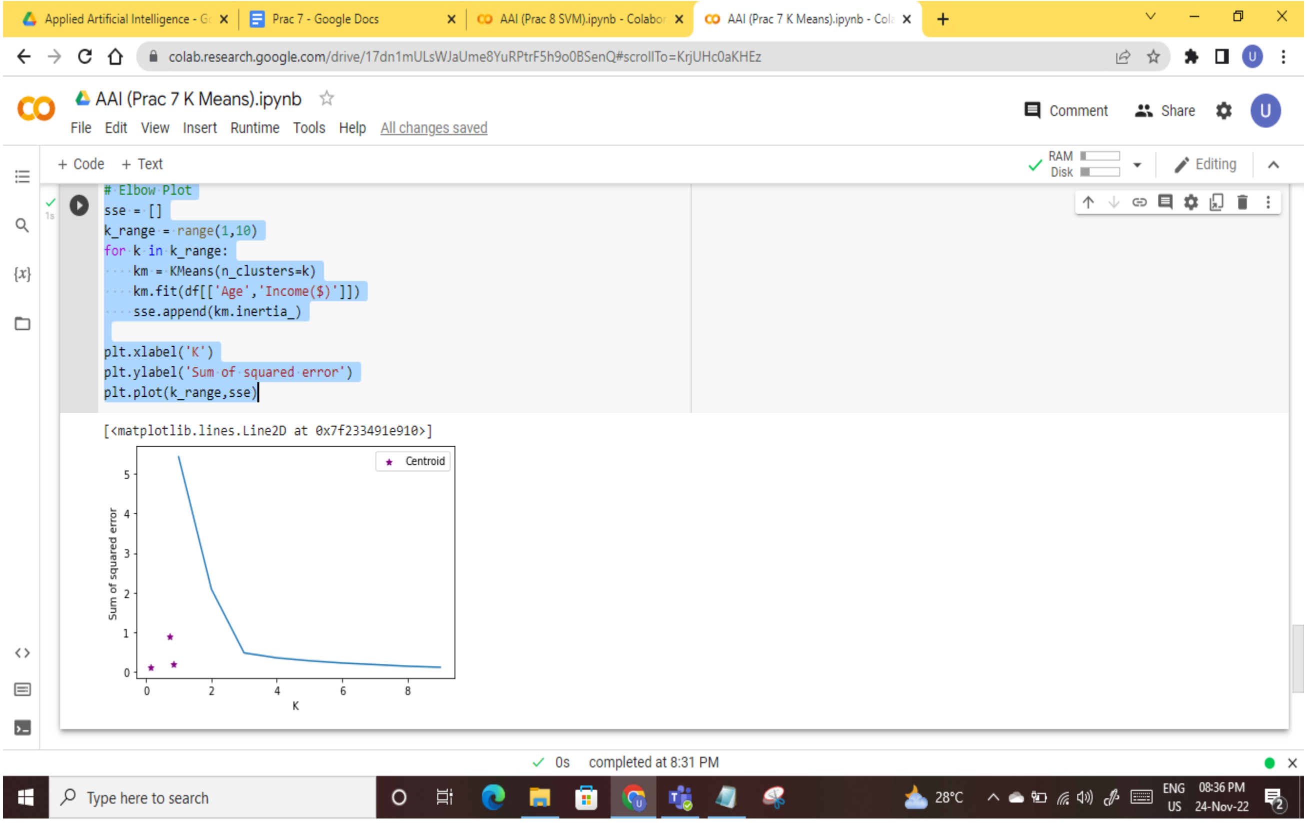
```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]

plt.scatter(df1['Age'],df1['Income($)'],color='green')
plt.scatter(df2['Age'],df2['Income($)'],color='red')
plt.scatter(df3['Age'],df3['Income($)'],color='black')
```


Applied Artificial Intelligence - G
Prac 7 - Google Docs
AAI (Prac 8 SVM).ipynb - Colabo
AAI (Prac 7 K Means).ipynb - Col
colab.research.google.com/drive/17dn1mULsWJaUme8YuRPtrF5h9o0BSenQ#scrollTo=gcR1zCfjKBoC
AAI (Prac 7 K Means).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
1.0
0.8
0.6
0.4
0.2
0.0
0.0 0.2 0.4 0.6 0.8 1.0
Type here to search
28°C
ENG US 08:35 PM 24-Nov-22

```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]

plt.scatter(df1['Age'],df1['Income($)'],color='green')
plt.scatter(df2['Age'],df2['Income($)'],color='red')
plt.scatter(df3['Age'],df3['Income($)'],color='black')

<matplotlib.collections.PathCollection at 0x7f23349a09d0>

[12] plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',label='Centroid')
```

✓ 0s completed at 8:31 PM

```
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',label='Centroid')

plt.xlabel('Age')
plt.ylabel('Income ($)')
plt.legend()

# Elbow Plot
sse = []
k_range = range(1,10)
for k in k_range:
        km = KMeans(n_clusters=k)
        km.fit(df[['Age','Income($)']])
        sse.append(km.inertia_)

plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(k_range,sse)
```


Applied Artificial Intelligence - G
Prac 7 - Google Docs
AAI (Prac 8 SVM).ipynb - Colabo
AAI (Prac 7 K Means).ipynb - Col
colab.research.google.com/drive/17dn1mULsWJaUme8YuRPtrF5h9o0BSenQ#scrollTo=KrjUHc0aKHEz
AAI (Prac 7 K Means).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
Centroid
5
4
3
2
1
0
Sum of squared error
0
2
4
6
8
K
0s completed at 8:31 PM
Type here to search
28°C
ENG US
08:36 PM 24-Nov-22

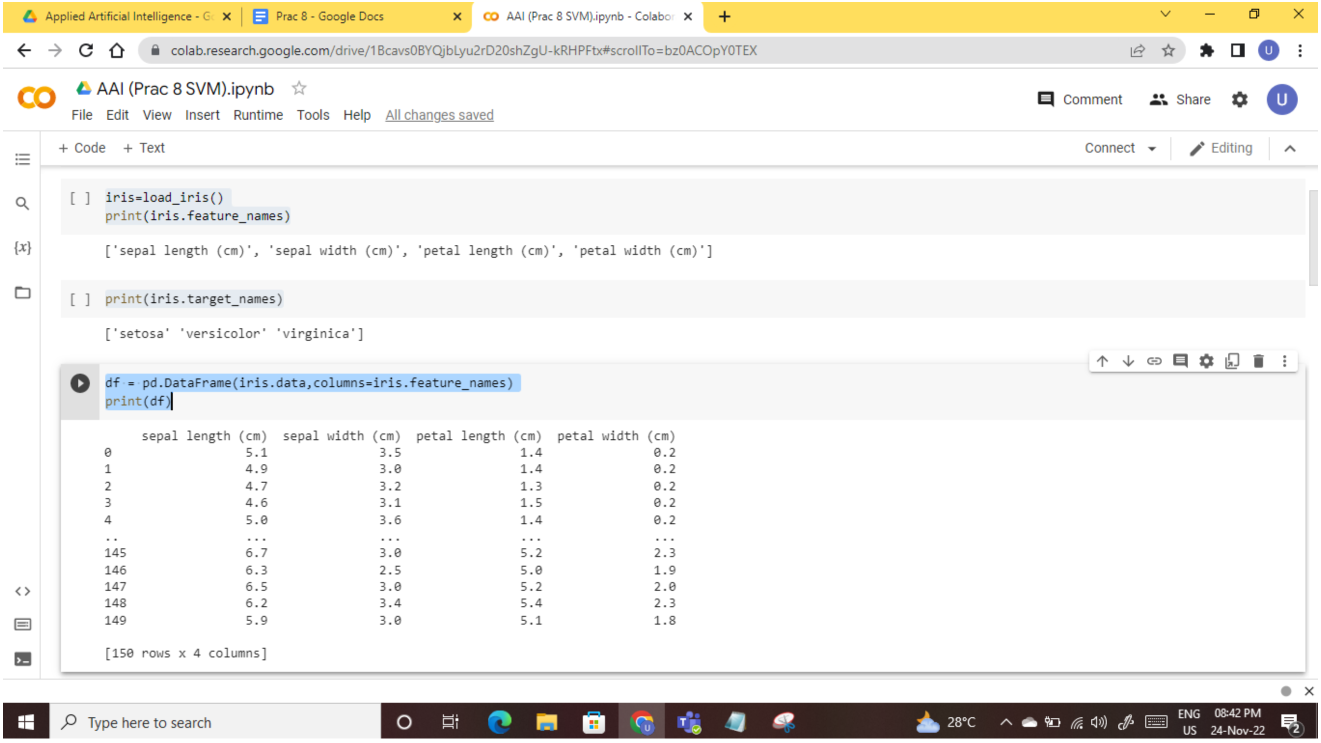
```
# Elbow Plot
sse = []
k_range = range(1,10)
for k in k_range:
    km = KMeans(n_clusters=k)
    km.fit(df[['Age','Income($)']])
    sse.append(km.inertia_)

plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(k_range,sse)
```

```
[<matplotlib.lines.Line2D at 0x7f233491e910>]
```

Write an application to implement a support vector machine algorithm

Code :

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC

iris=load_iris()
print(iris.feature_names)
print(iris.target_names)

df = pd.DataFrame(iris.data,columns=iris.feature_names)
print(df)
```

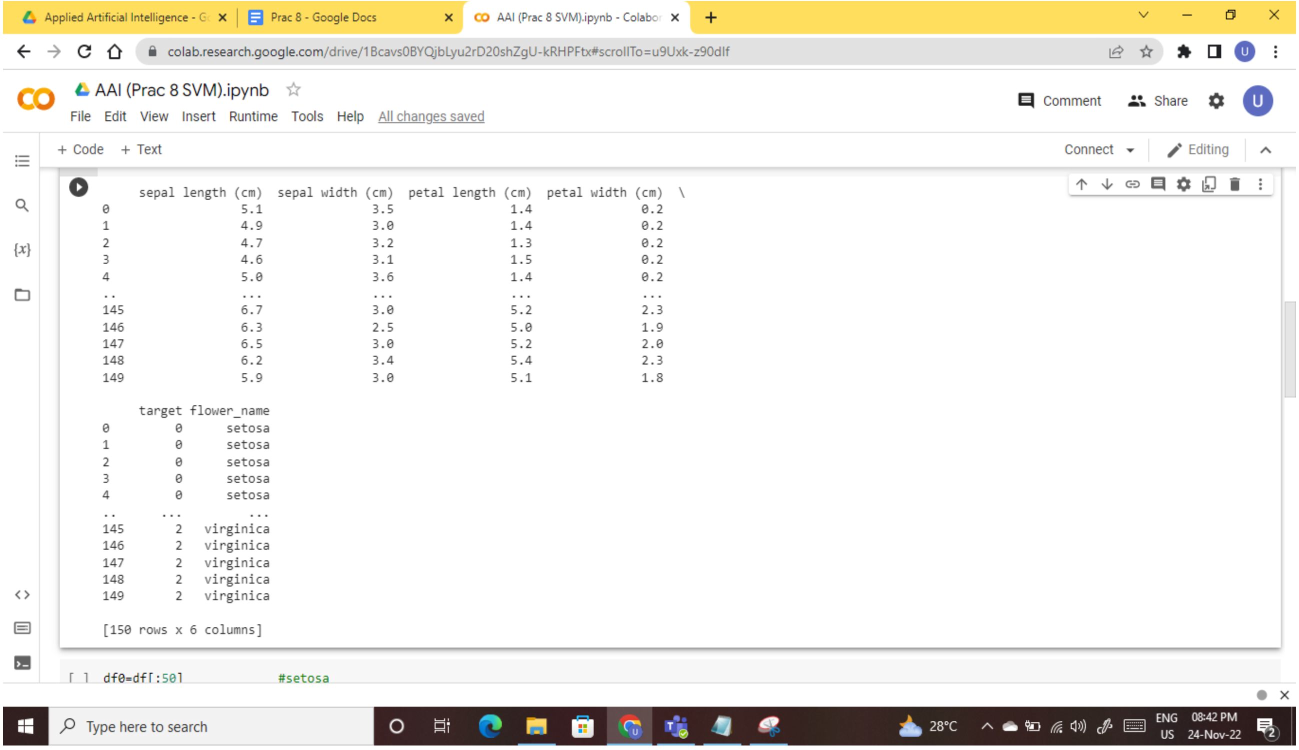

Applied Artificial Intelligence - G
Prac 8 - Google Docs
AAI (Prac 8 SVM).ipynb - Colabo
colab.research.google.com/drive/1Bcavs0BYQjbLyu2rD20shZgU-kRHPFtx#scrollTo=bz0ACOpY0TEX
AAI (Prac 8 SVM).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Connect
Editing
Type here to search
28°C
ENG US
08:42 PM 24-Nov-22

```
[ ] iris=load_iris()
    print(iris.feature_names)

    ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']

[ ] print(iris.target_names)

    ['setosa' 'versicolor' 'virginica']

    df = pd.DataFrame(iris.data,columns=iris.feature_names)
    print(df)

         sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
    0                  5.1               3.5                1.4               0.2
    1                  4.9               3.0                1.4               0.2
    2                  4.7               3.2                1.3               0.2
    3                  4.6               3.1                1.5               0.2
    4                  5.0               3.6                1.4               0.2
    ..                 ...               ...                ...               ...
    145                6.7               3.0                5.2               2.3
    146                6.3               2.5                5.0               1.9
    147                6.5               3.0                5.2               2.0
    148                6.2               3.4                5.4               2.3
    149                5.9               3.0                5.1               1.8

    [150 rows x 4 columns]
```

df['target']=iris.target

df['flower_name'] =df.target.apply(lambda x: iris.target_names[x])

print(df)



Applied Artificial Intelligence - G
Prac 8 - Google Docs
AAI (Prac 8 SVM).ipynb - Colabo
colab.research.google.com/drive/1Bcavs0BYQjbLyu2rD20shZgU-kRHPFtx#scrollTo=u9Uxk-z90dIf
AAI (Prac 8 SVM).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Connect
Editing
Type here to search
28°C
ENG US
08:42 PM 24-Nov-22

```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                  5.1               3.5                1.4               0.2
1                  4.9               3.0                1.4               0.2
2                  4.7               3.2                1.3               0.2
3                  4.6               3.1                1.5               0.2
4                  5.0               3.6                1.4               0.2
..                 ...               ...                ...               ...
145                6.7               3.0                5.2               2.3
146                6.3               2.5                5.0               1.9
147                6.5               3.0                5.2               2.0
148                6.2               3.4                5.4               2.3
149                5.9               3.0                5.1               1.8

     target flower_name
0         0      setosa
1         0      setosa
2         0      setosa
3         0      setosa
4         0      setosa
..      ...         ...
145       2   virginica
146       2   virginica
147       2   virginica
148       2   virginica
149       2   virginica

[150 rows x 6 columns]
```

```
[ ] df0=df[:50]          #setosa
```

df0=df[:50]   #setosa

df1=df[50:100]   #versicolor

df2=df[100:]   #virginica

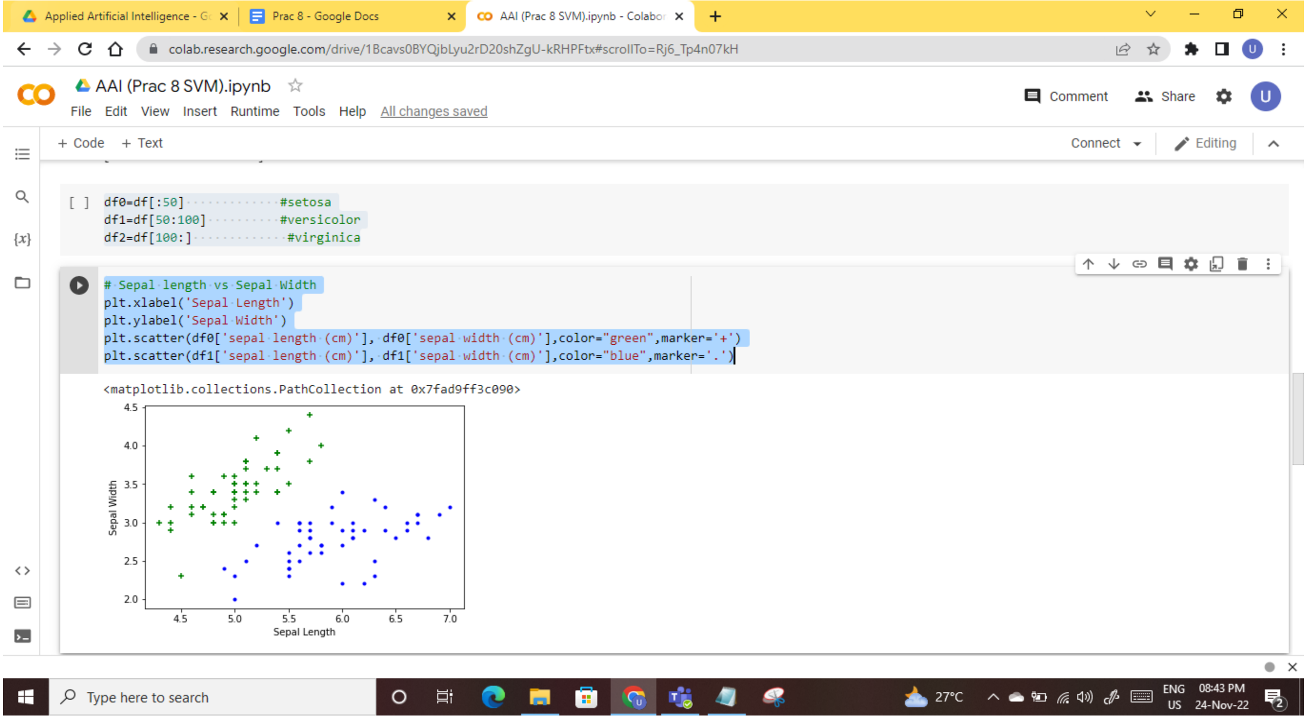
# Sepal length vs Sepal Width

plt.xlabel('Sepal Length')

plt.ylabel('Sepal Width')

plt.scatter(df0['sepal length (cm)'], df0['sepal width (cm)'], color="green",marker='+')

plt.scatter(df1['sepal length (cm)'], df1['sepal width (cm)'],color="blue",marker='.')



Applied Artificial Intelligence - G
Prac 8 - Google Docs
AAI (Prac 8 SVM).ipynb - Colabo
colab.research.google.com/drive/1Bcavs0BYQjbLyu2rD20shZgU-kRHPFtx#scrollTo=Rj6_Tp4n07kH
AAI (Prac 8 SVM).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Connect
Editing
4.5
4.0
3.5
3.0
2.5
2.0
Sepal Width
4.5
5.0
5.5
6.0
6.5
7.0
Sepal Length
Type here to search
27°C
ENG US
08:43 PM
24-Nov-22

```
df0=df[:50]            #setosa
df1=df[50:100]         #versicolor
df2=df[100:]           #virginica
```

```
# Sepal length vs Sepal Width
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.scatter(df0['sepal length (cm)'], df0['sepal width (cm)'],color="green",marker='+')
plt.scatter(df1['sepal length (cm)'], df1['sepal width (cm)'],color="blue",marker='.')
```

```
<matplotlib.collections.PathCollection at 0x7fad9ff3c090>
```

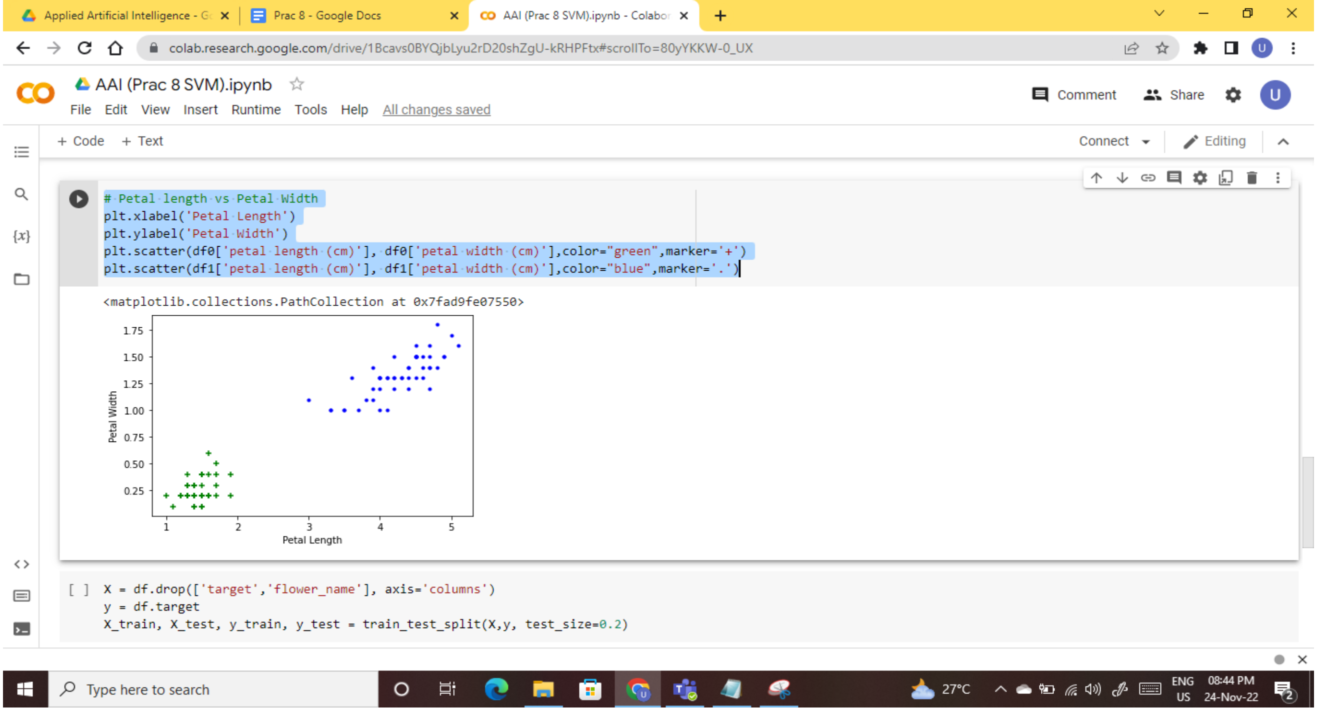
Petal length vs Petal Width

plt.xlabel('Petal Length')

plt.ylabel('Petal Width')

plt.scatter(df0['petal length (cm)'], df0['petal width (cm)'],color="green",marker='+')

plt.scatter(df1['petal length (cm)'], df1['petal width (cm)'],color="blue",marker='.')



Applied Artificial Intelligence - G
Prac 8 - Google Docs
AAI (Prac 8 SVM).ipynb - Colabo
colab.research.google.com/drive/1Bcavs0BYQjbLyu2rD20shZgU-kRHPFtx#scrollTo=80yYKKW-0_UX
AAI (Prac 8 SVM).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Connect
Editing
1.75
1.50
1.25
1.00
0.75
0.50
0.25
Petal Width
1
2
3
4
5
Petal Length
Type here to search
27°C
ENG US
08:44 PM
24-Nov-22

```
# Petal length vs Petal Width
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.scatter(df0['petal length (cm)'], df0['petal width (cm)'],color="green",marker='+')
plt.scatter(df1['petal length (cm)'], df1['petal width (cm)'],color="blue",marker='.')
```

```
<matplotlib.collections.PathCollection at 0x7fad9fe07550>
```

```
X = df.drop(['target','flower_name'], axis='columns')
y = df.target
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2)
```

X = df.drop(['target','flower_name'], axis='columns')

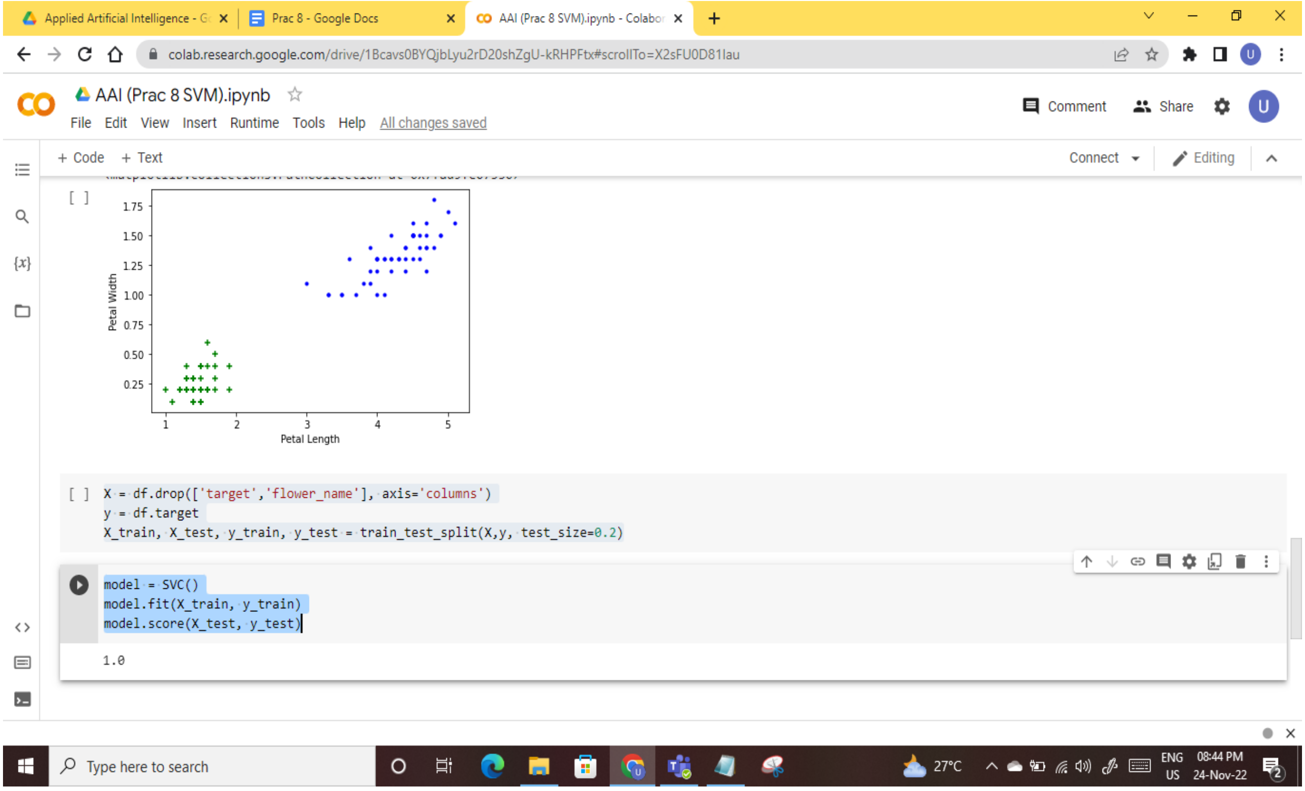
y = df.target

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2)

model = SVC()

model.fit(X_train, y_train)

model.score(X_test, y_test)


Applied Artificial Intelligence - G
Prac 8 - Google Docs
AAI (Prac 8 SVM).ipynb - Colabo
colab.research.google.com/drive/1Bcavs0BYQjbLyu2rD20shZgU-kRHPFtx#scrollTo=X2sFU0D81lau
AAI (Prac 8 SVM).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Connect
Editing
Petal Width
1.75
1.50
1.25
1.00
0.75
0.50
0.25
1
2
3
4
5
Petal Length
Type here to search
27°C
ENG US
08:44 PM
24-Nov-22

```
X = df.drop(['target','flower_name'], axis='columns')
y = df.target
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2)
```

```
model = SVC()
model.fit(X_train, y_train)
model.score(X_test, y_test)
```

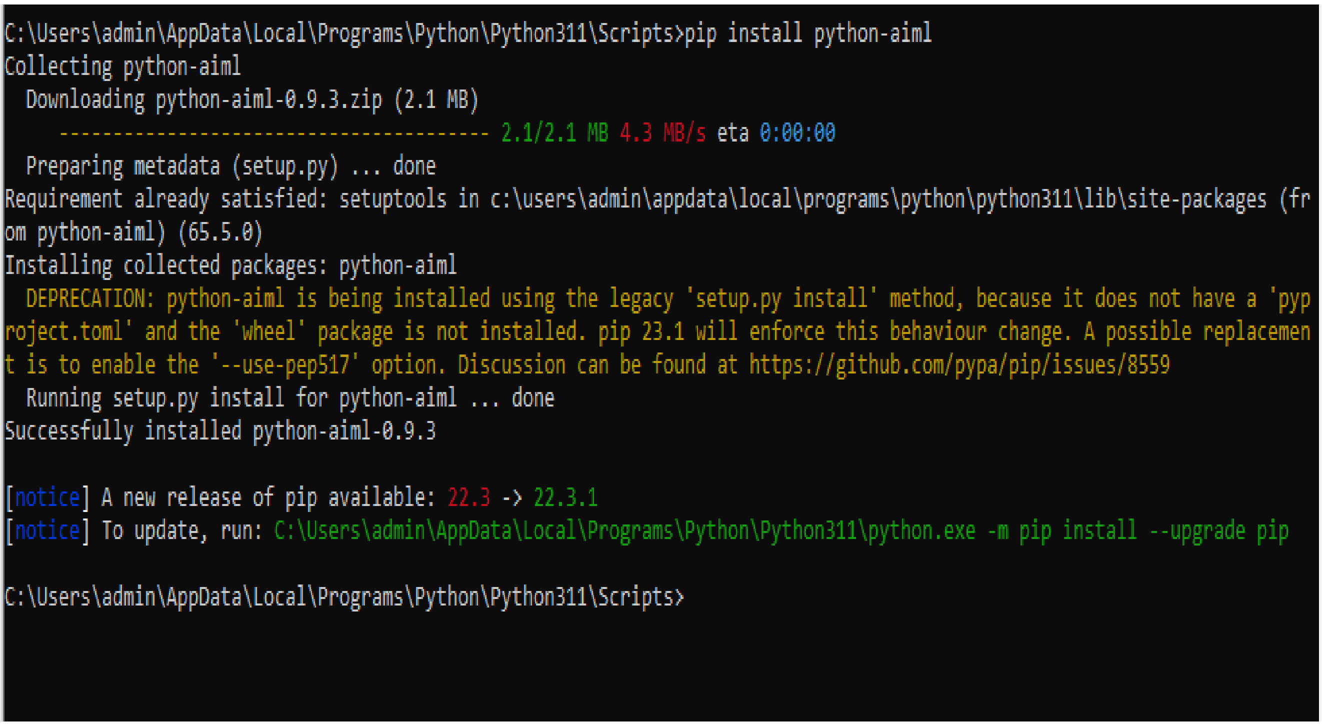
```
1.0
```

Design a bot using AIML

Code :

1) Install python-aiml

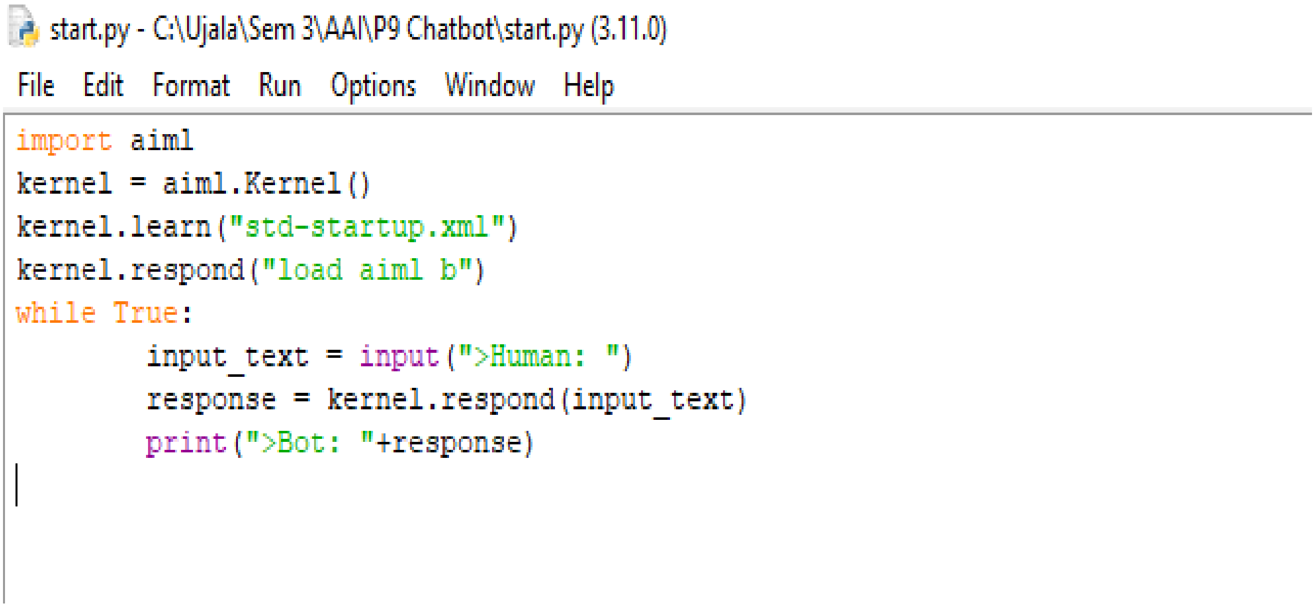
pip install python-aiml



```
C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts>pip install python-aiml
Collecting python-aiml
  Downloading python-aiml-0.9.3.zip (2.1 MB)
     ---------------------------------------- 2.1/2.1 MB 4.3 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: setuptools in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (fr
om python-aiml) (65.5.0)
Installing collected packages: python-aiml
  DEPRECATION: python-aiml is being installed using the legacy 'setup.py install' method, because it does not have a 'pyp
roject.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour change. A possible replacemen
t is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
  Running setup.py install for python-aiml ... done
Successfully installed python-aiml-0.9.3

[notice] A new release of pip available: 22.3 -> 22.3.1
[notice] To update, run: C:\Users\admin\AppData\Local\Programs\Python\Python311\python.exe -m pip install --upgrade pip

C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts>
```

2) Write a code in a python file.

```
import aiml
kernel = aiml.Kernel()
kernel.learn("std-startup.xml")
kernel.respond("load aiml b")
while True:
        input_text = input(">Human: ")
        response = kernel.respond(input_text)
        print(">Bot: "+response)
```

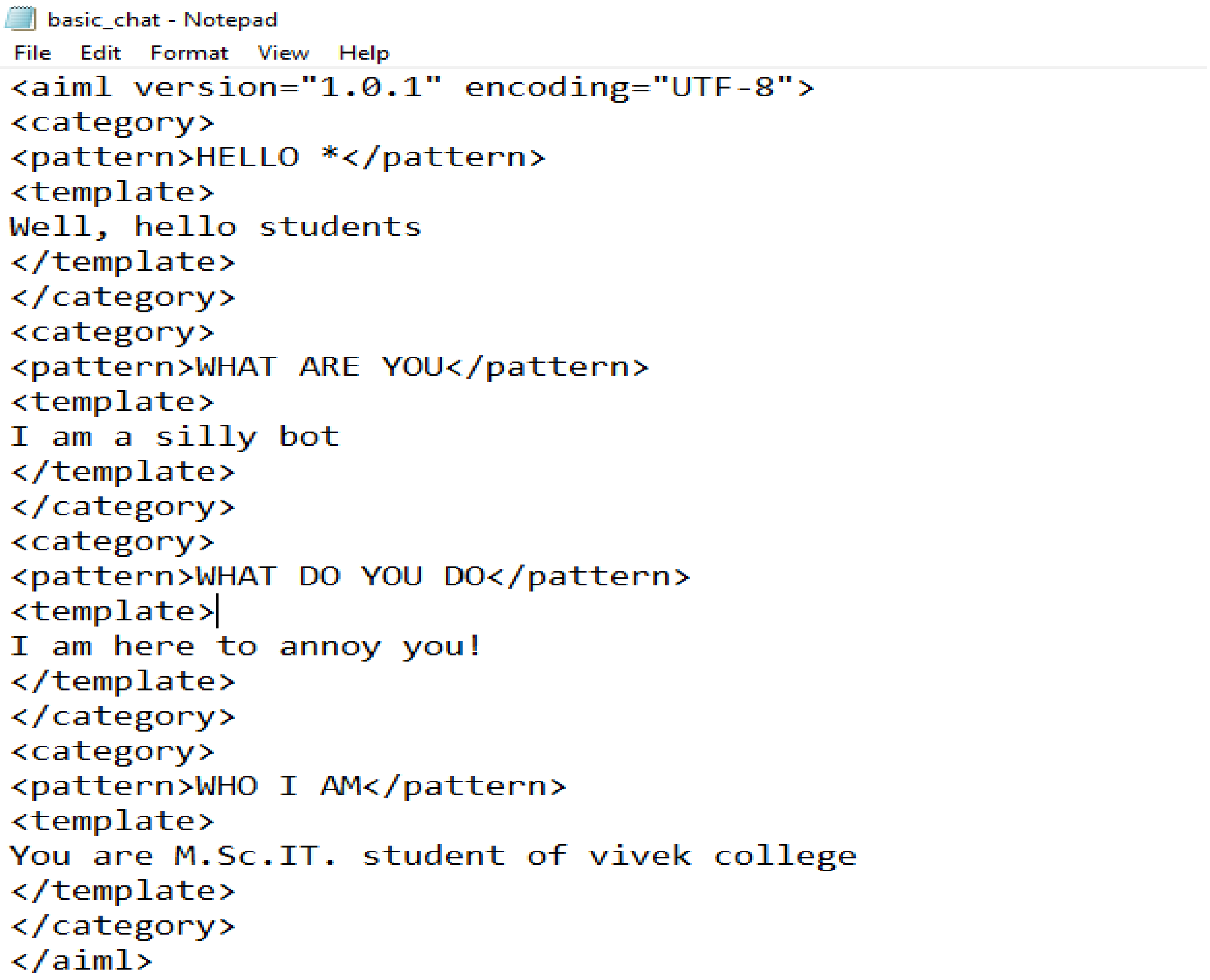

start.py - C:\Ujala\Sem 3\AAI\P9 Chatbot\start.py (3.11.0)
File Edit Format Run Options Window Help

```
import aiml
kernel = aiml.Kernel()
kernel.learn("std-startup.xml")
kernel.respond("load aiml b")
while True:
        input_text = input(">Human: ")
        response = kernel.respond(input_text)
        print(">Bot: "+response)
```

3) Write a code in basic_chat.aiml file :

```
<aiml version="1.0.1" encoding="UTF-8">
<category>
        <pattern>HELLO *</pattern>
        <template>
                Well, hello students
        </template>
</category>
<category>
        <pattern>WHAT ARE YOU</pattern>
        <template>
                I am a silly bot
        </template>
</category>
<category>
        <pattern>WHAT DO YOU DO</pattern>
        <template>
                I am here to annoy you!
        </template>
</category>
```

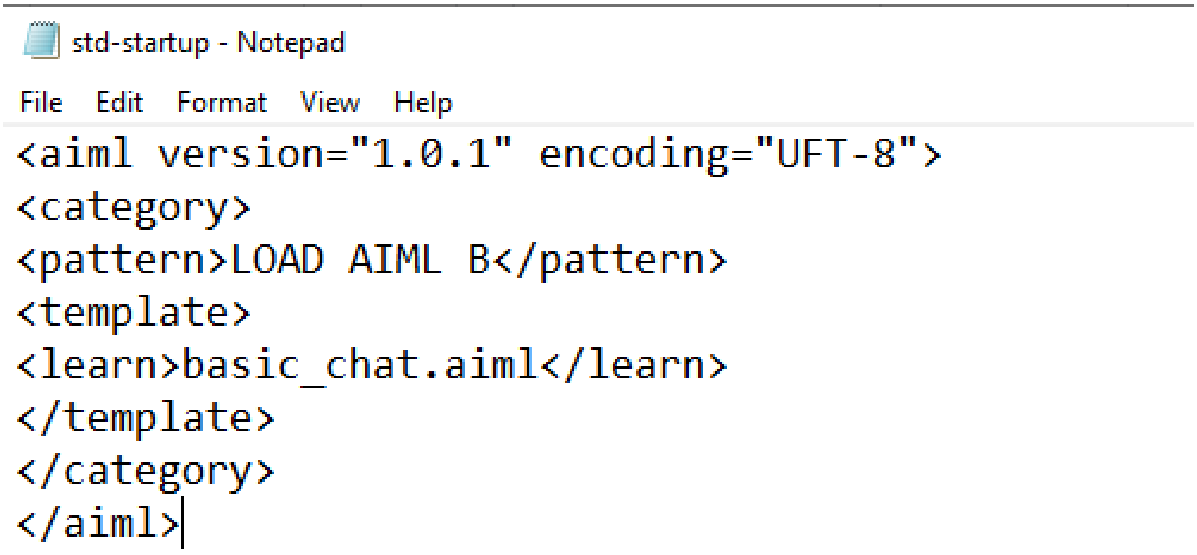
```
<category>
        <pattern>WHO I AM</pattern>
        <template>
                You are M.Sc.IT. student of vivek college
        </template>
</category>


</aiml>
```



```
<aiml version="1.0.1" encoding="UTF-8">
<category>
<pattern>HELLO *</pattern>
<template>
Well, hello students
</template>
</category>
<category>
<pattern>WHAT ARE YOU</pattern>
<template>
I am a silly bot
</template>
</category>
<category>
<pattern>WHAT DO YOU DO</pattern>
<template>
I am here to annoy you!
</template>
</category>
<category>
<pattern>WHO I AM</pattern>
<template>
You are M.Sc.IT. student of vivek college
</template>
</category>
</aiml>
```

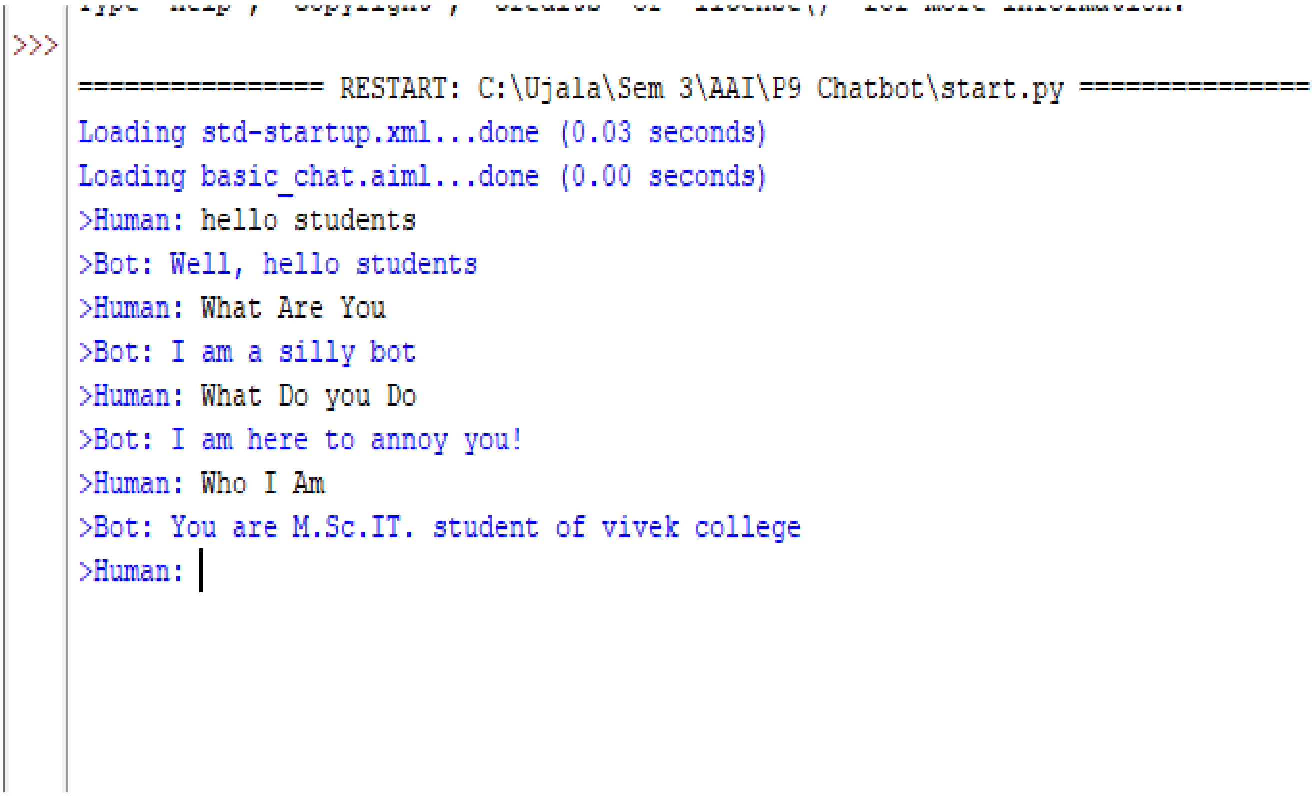
4) Write a code in std_startup.xml file :

```
<aiml version="1.0.1" encoding="UFT-8">
<category>
        <pattern>LOAD AIML B</pattern>
        <template>
                <learn>basic_chat.aiml</learn>
        </template>
</category>
</aiml>
```



```
<aiml version="1.0.1" encoding="UFT-8">
<category>
<pattern>LOAD AIML B</pattern>
<template>
<learn>basic_chat.aiml</learn>
</template>
</category>
</aiml>
```

Output :



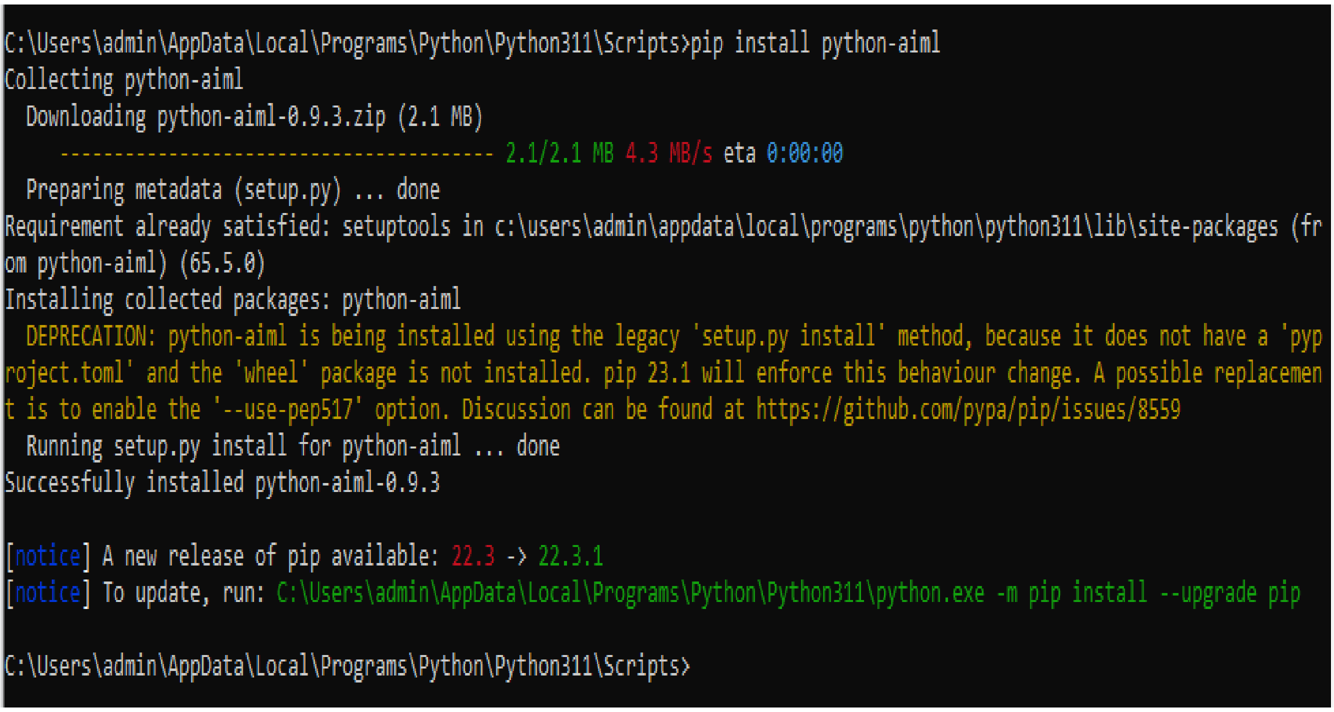
```
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: C:\Ujala\Sem 3\AAI\P9 Chatbot\start.py ===============
Loading std-startup.xml...done (0.03 seconds)
Loading basic_chat.aiml...done (0.00 seconds)
>Human: hello students
>Bot: Well, hello students
>Human: What Are You
>Bot: I am a silly bot
>Human: What Do you Do
>Bot: I am here to annoy you!
>Human: Who I Am
>Bot: You are M.Sc.IT. student of vivek college
>Human: 
```

Design an Expert System using AIML

Code :

1) Install python-aiml

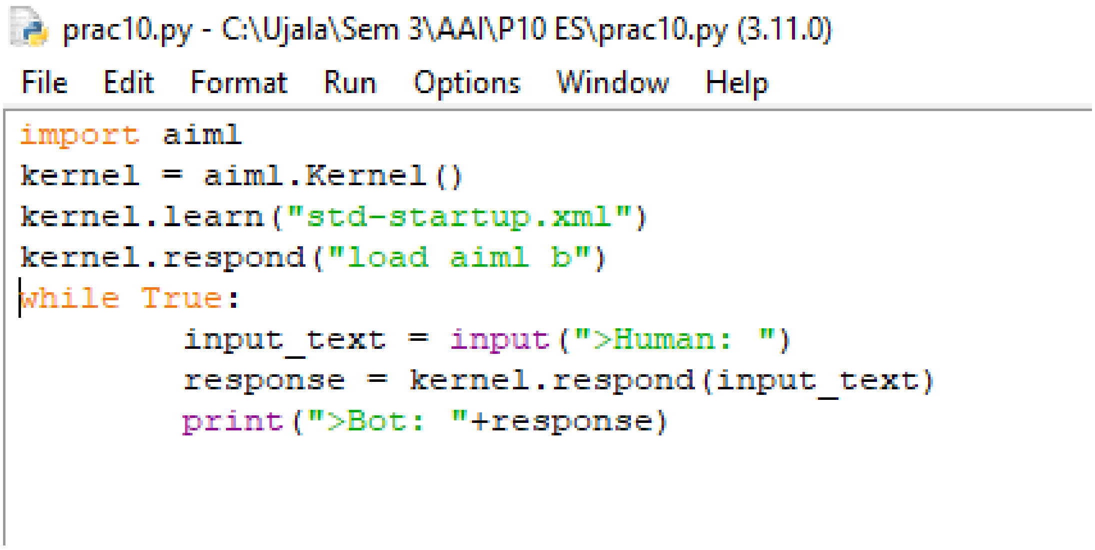
pip install python-aiml



```
C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts>pip install python-aiml
Collecting python-aiml
  Downloading python-aiml-0.9.3.zip (2.1 MB)
     ---------------------------------------- 2.1/2.1 MB 4.3 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: setuptools in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (fr
om python-aiml) (65.5.0)
Installing collected packages: python-aiml
  DEPRECATION: python-aiml is being installed using the legacy 'setup.py install' method, because it does not have a 'pyp
roject.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour change. A possible replacemen
t is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
  Running setup.py install for python-aiml ... done
Successfully installed python-aiml-0.9.3

[notice] A new release of pip available: 22.3 -> 22.3.1
[notice] To update, run: C:\Users\admin\AppData\Local\Programs\Python\Python311\python.exe -m pip install --upgrade pip

C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts>
```

2) Write a code in a python file.

```
import aiml
kernel = aiml.Kernel()
kernel.learn("std-startup.xml")
kernel.respond("load aiml b")
while True:
        input_text = input(">Human: ")
        response = kernel.respond(input_text)
        print(">Bot: "+response)
```



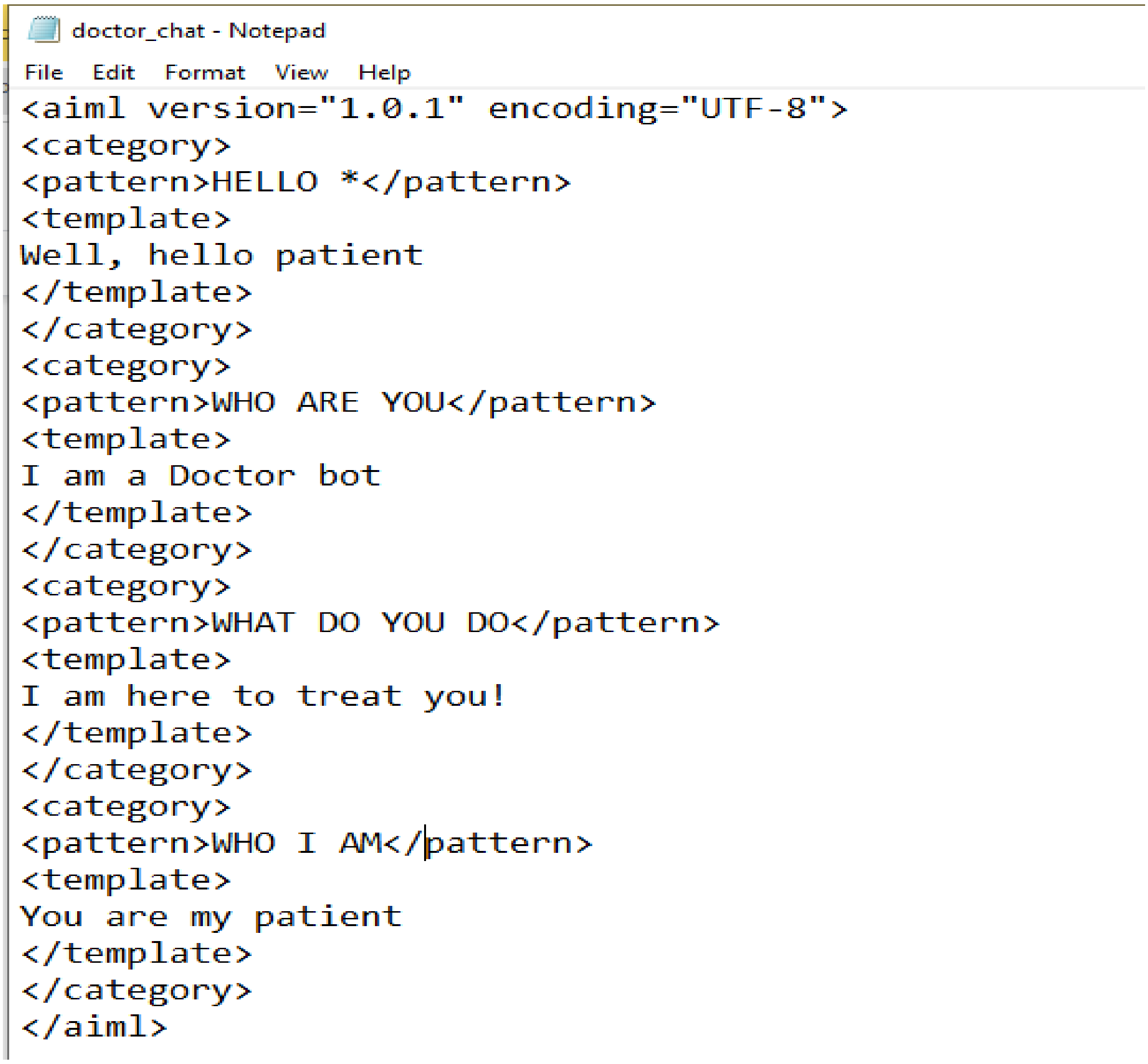
prac10.py - C:\Ujala\Sem 3\AAI\P10 ES\prac10.py (3.11.0)
File Edit Format Run Options Window Help

```
import aiml
kernel = aiml.Kernel()
kernel.learn("std-startup.xml")
kernel.respond("load aiml b")
while True:
        input_text = input(">Human: ")
        response = kernel.respond(input_text)
        print(">Bot: "+response)
```

3) Write a code in doctor_chat.aiml file :

```
<aiml version="1.0.1" encoding="UTF-8">
<category>
        <pattern>HELLO *</pattern>
        <template>
                Well, hello patient
        </template>
</category>
<category>
        <pattern>WHO ARE YOU</pattern>
        <template>
                I am a Doctor bot
        </template>
</category>
<category>
        <pattern>WHAT DO YOU DO</pattern>
        <template>
                I am here to treat you!
        </template>
</category>
```

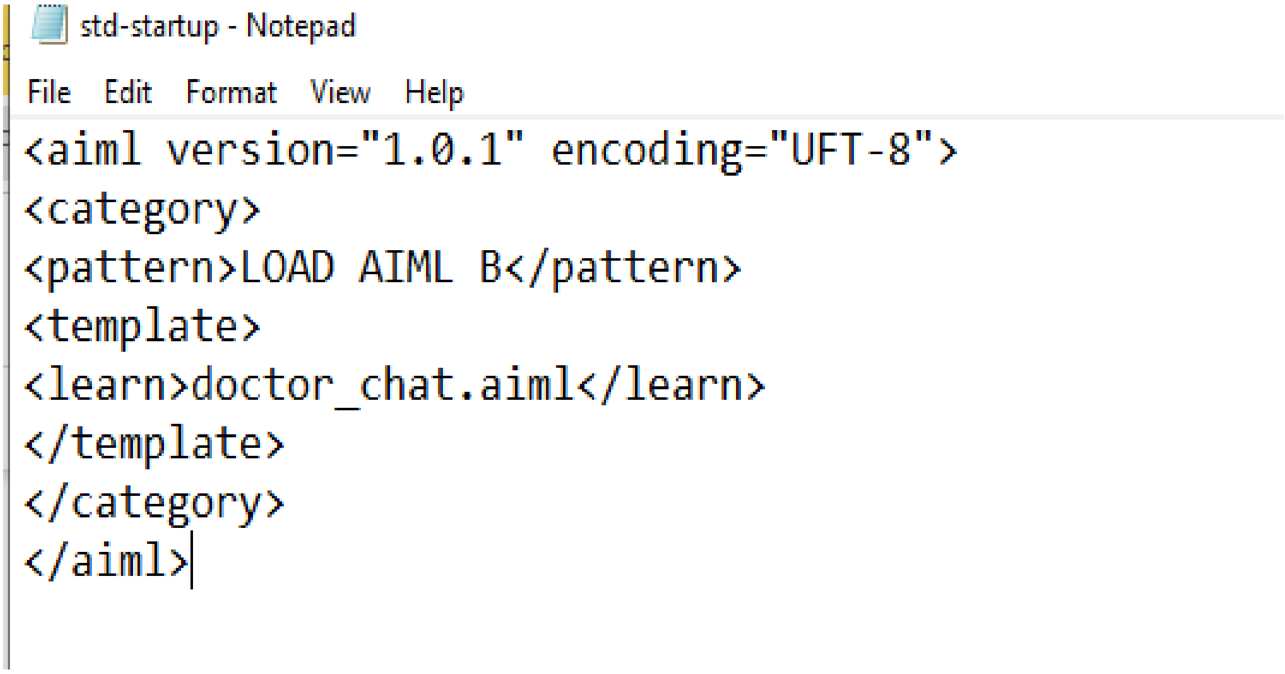
```
<category>
        <pattern>WHO I AM</pattern>
        <template>
                You are my patient
        </template>
</category>
</aiml>
```



```
<aiml version="1.0.1" encoding="UTF-8">
<category>
<pattern>HELLO *</pattern>
<template>
Well, hello patient
</template>
</category>
<category>
<pattern>WHO ARE YOU</pattern>
<template>
I am a Doctor bot
</template>
</category>
<category>
<pattern>WHAT DO YOU DO</pattern>
<template>
I am here to treat you!
</template>
</category>
<category>
<pattern>WHO I AM</pattern>
<template>
You are my patient
</template>
</category>
</aiml>
```

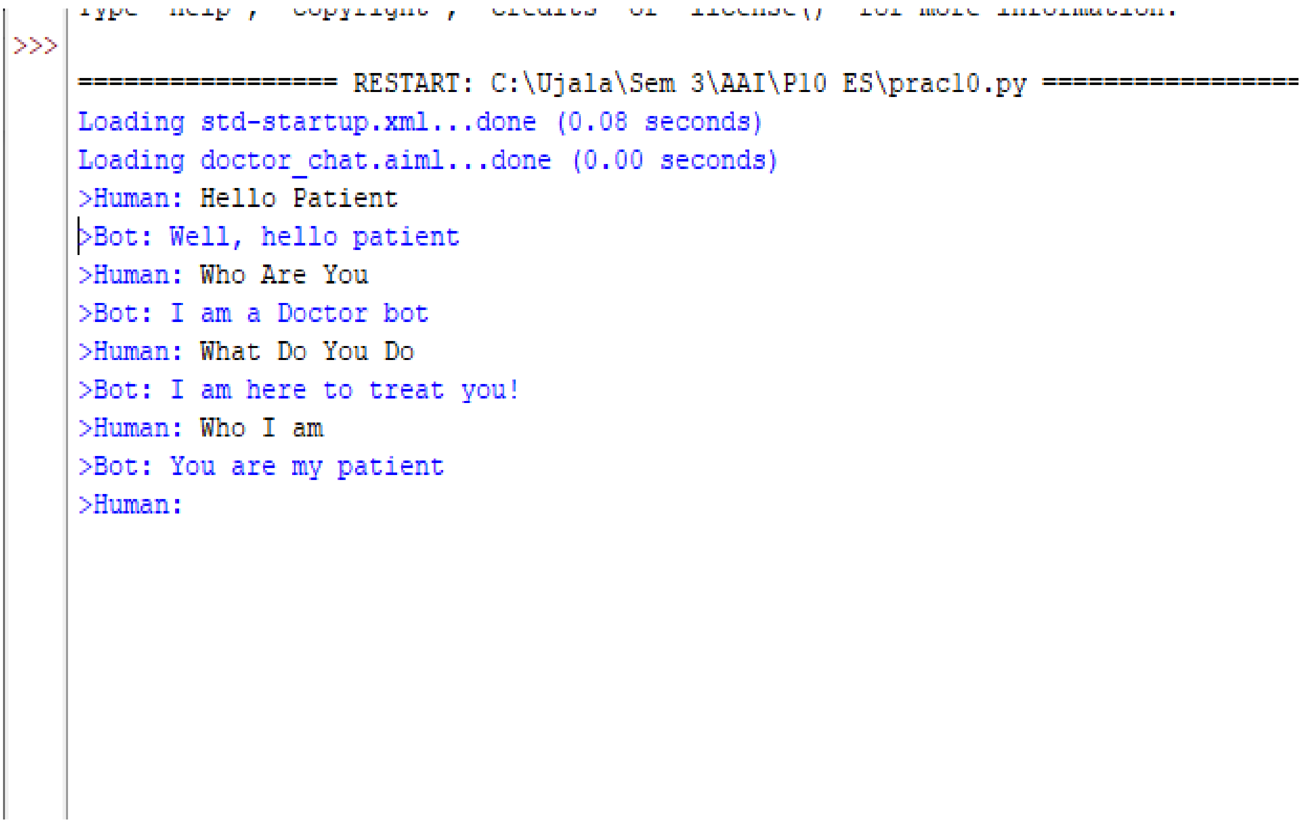
4) Write a code in std_startup.xml file :

```
<aiml version="1.0.1" encoding="UFT-8">
<category>
        <pattern>LOAD AIML B</pattern>
        <template>
                <learn>doctor_chat.aiml</learn>
        </template>
</category>
</aiml>
```



```
<aiml version="1.0.1" encoding="UFT-8">
<category>
<pattern>LOAD AIML B</pattern>
<template>
<learn>doctor_chat.aiml</learn>
</template>
</category>
</aiml>
```

Output :



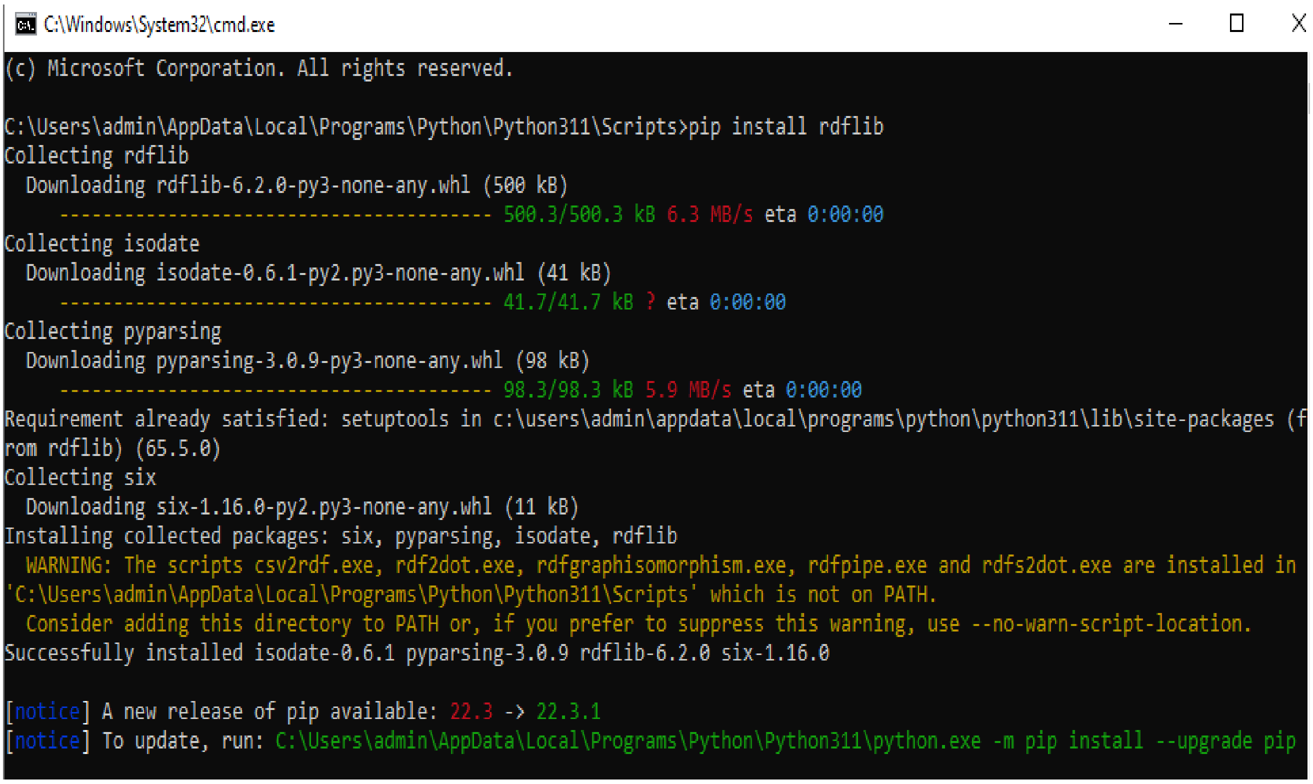
```
Type "help", "copyright", "credits" or "license()" for more information.
>>>
    ================= RESTART: C:\Ujala\Sem 3\AAI\P10 ES\prac10.py =================
    Loading std-startup.xml...done (0.08 seconds)
    Loading doctor_chat.aiml...done (0.00 seconds)
    >Human: Hello Patient
    >Bot: Well, hello patient
    >Human: Who Are You
    >Bot: I am a Doctor bot
    >Human: What Do You Do
    >Bot: I am here to treat you!
    >Human: Who I am
    >Bot: You are my patient
    >Human:
```

Design an application to simulate Semantic Web

Code :

1) Install the rdflib :

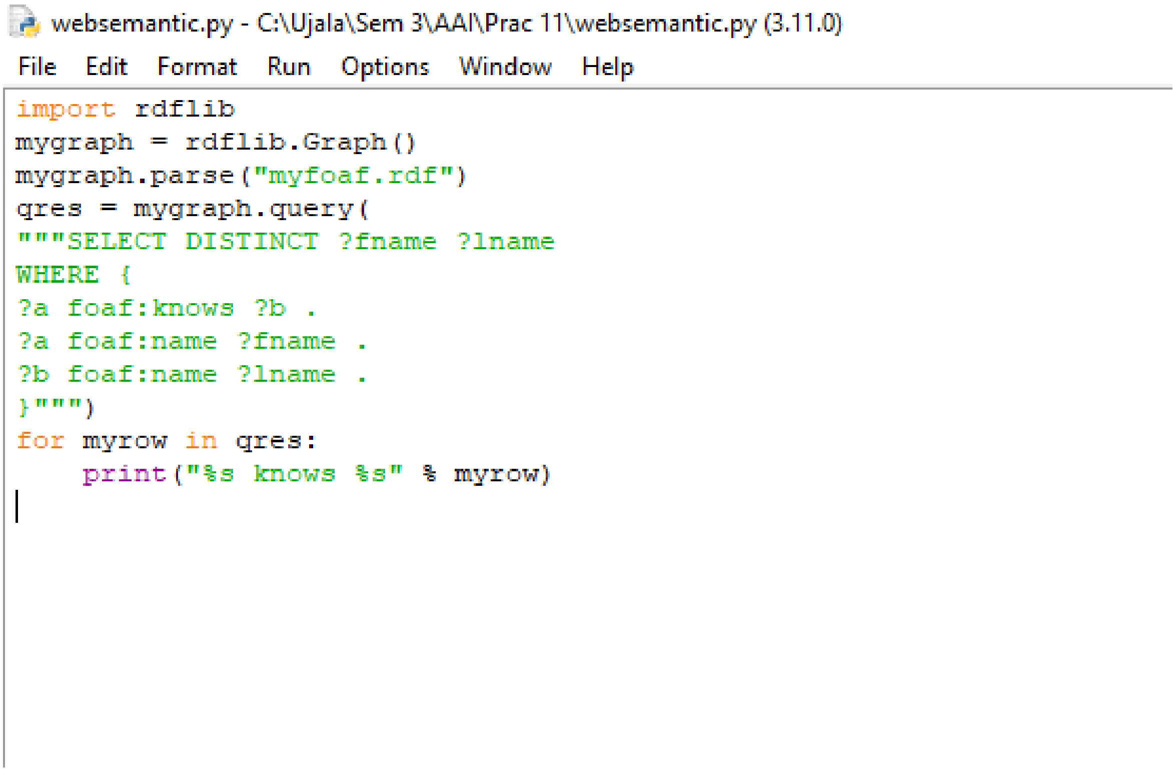
pip install rdflib



```
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts>pip install rdflib
Collecting rdflib
  Downloading rdflib-6.2.0-py3-none-any.whl (500 kB)
     ---------------------------------------- 500.3/500.3 kB 6.3 MB/s eta 0:00:00
Collecting isodate
  Downloading isodate-0.6.1-py2.py3-none-any.whl (41 kB)
     ---------------------------------------- 41.7/41.7 kB ? eta 0:00:00
Collecting pyparsing
  Downloading pyparsing-3.0.9-py3-none-any.whl (98 kB)
     ---------------------------------------- 98.3/98.3 kB 5.9 MB/s eta 0:00:00
Requirement already satisfied: setuptools in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (f
rom rdflib) (65.5.0)
Collecting six
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, pyparsing, isodate, rdflib
  WARNING: The scripts csv2rdf.exe, rdf2dot.exe, rdfgraphisomorphism.exe, rdfpipe.exe and rdfs2dot.exe are installed in
'C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed isodate-0.6.1 pyparsing-3.0.9 rdflib-6.2.0 six-1.16.0

[notice] A new release of pip available: 22.3 -> 22.3.1
[notice] To update, run: C:\Users\admin\AppData\Local\Programs\Python\Python311\python.exe -m pip install --upgrade pip
```

2) Write a code in rdflib.py file.

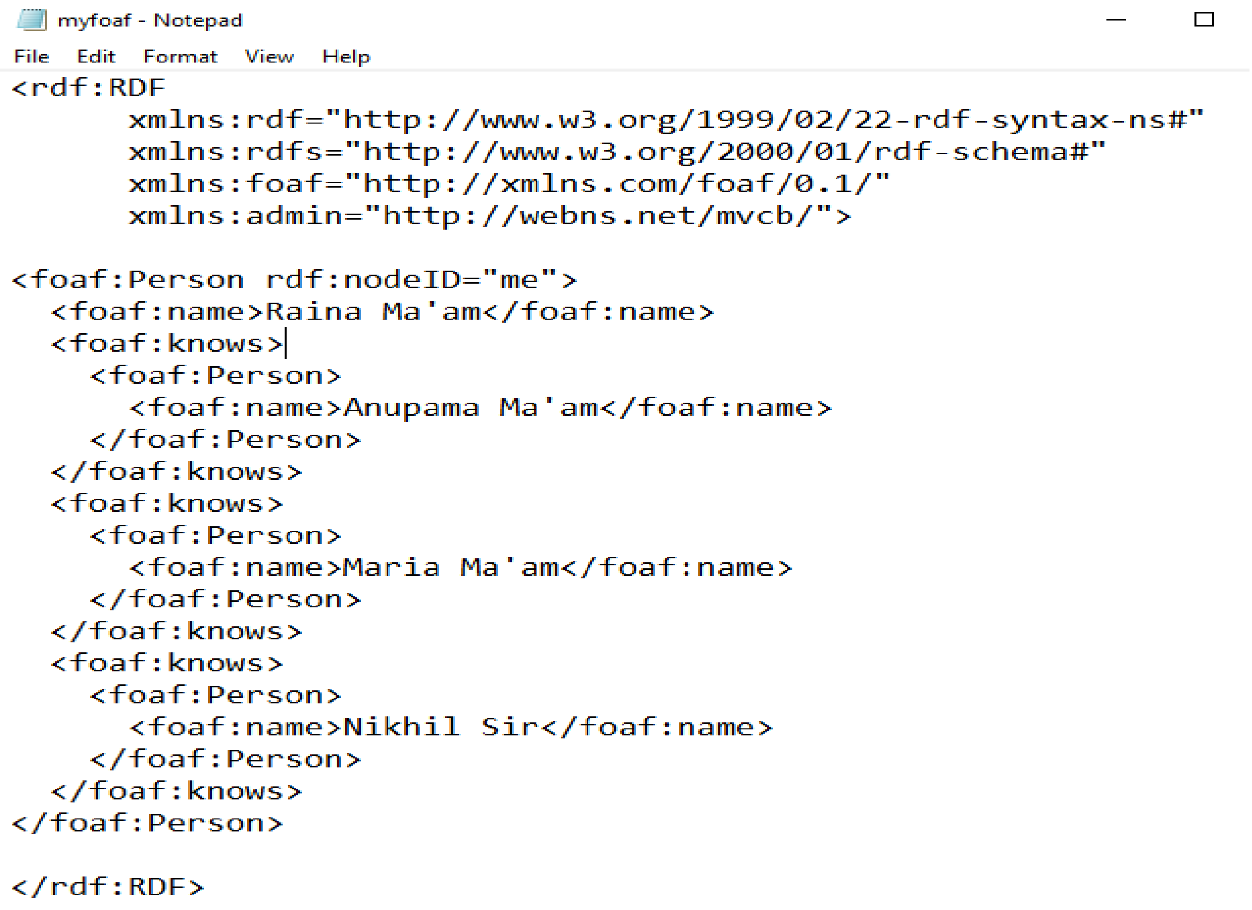
```
import rdflib
mygraph = rdflib.Graph()
mygraph.parse("myfoaf.rdf")
qres = mygraph.query(
"""SELECT DISTINCT ?fname ?lname
WHERE {
?a foaf:knows ?b .
?a foaf:name ?fname .
?b foaf:name ?lname .
}""")
for myrow in qres:
        print("%s knows %s" % myrow)
```


websemantic.py - C:\Ujala\Sem 3\AAI\Prac 11\websemantic.py (3.11.0)
File Edit Format Run Options Window Help

```
import rdflib
mygraph = rdflib.Graph()
mygraph.parse("myfoaf.rdf")
qres = mygraph.query(
"""SELECT DISTINCT ?fname ?lname
WHERE {
?a foaf:knows ?b .
?a foaf:name ?fname .
?b foaf:name ?lname .
}""")
for myrow in qres:
    print("%s knows %s" % myrow)
```

3) Write a code in myfoaf file.

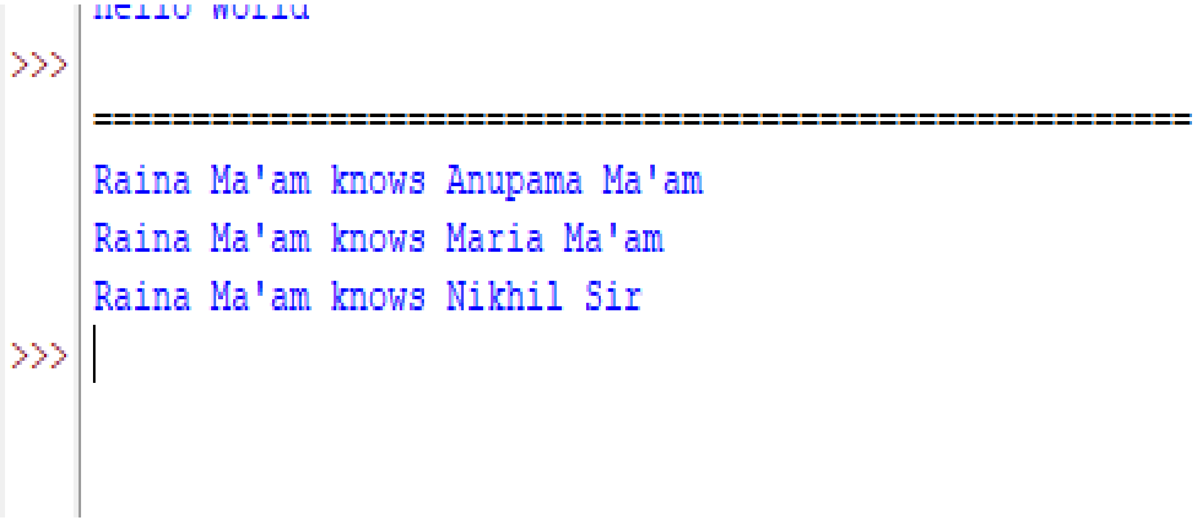
```
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:admin="http://webns.net/mvcb/">

<foaf:Person rdf:nodeID="me">
        <foaf:name>Raina Ma'am</foaf:name>
        <foaf:knows>
                <foaf:Person>
                        <foaf:name>Anupama Ma'am</foaf:name>
                </foaf:Person>
        </foaf:knows>
        <foaf:knows>
                <foaf:Person>
                        <foaf:name>Maria Ma'am</foaf:name>
                </foaf:Person>
        </foaf:knows>
        <foaf:knows>
                <foaf:Person>
                        <foaf:name>Nikhil Sir</foaf:name>
                </foaf:Person>
        </foaf:knows>
</foaf:Person>

</rdf:RDF>
```



```
<rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:foaf="http://xmlns.com/foaf/0.1/"
      xmlns:admin="http://webns.net/mvcb/">

<foaf:Person rdf:nodeID="me">
  <foaf:name>Raina Ma'am</foaf:name>
  <foaf:knows>
    <foaf:Person>
      <foaf:name>Anupama Ma'am</foaf:name>
    </foaf:Person>
  </foaf:knows>
  <foaf:knows>
    <foaf:Person>
      <foaf:name>Maria Ma'am</foaf:name>
    </foaf:Person>
  </foaf:knows>
  <foaf:knows>
    <foaf:Person>
      <foaf:name>Nikhil Sir</foaf:name>
    </foaf:Person>
  </foaf:knows>
</foaf:Person>

</rdf:RDF>
```

Output :

Run the rdflib.py file.



```
    hello world
>>> 
    ================================================================
    Raina Ma'am knows Anupama Ma'am
    Raina Ma'am knows Maria Ma'am
    Raina Ma'am knows Nikhil Sir
>>> 
```

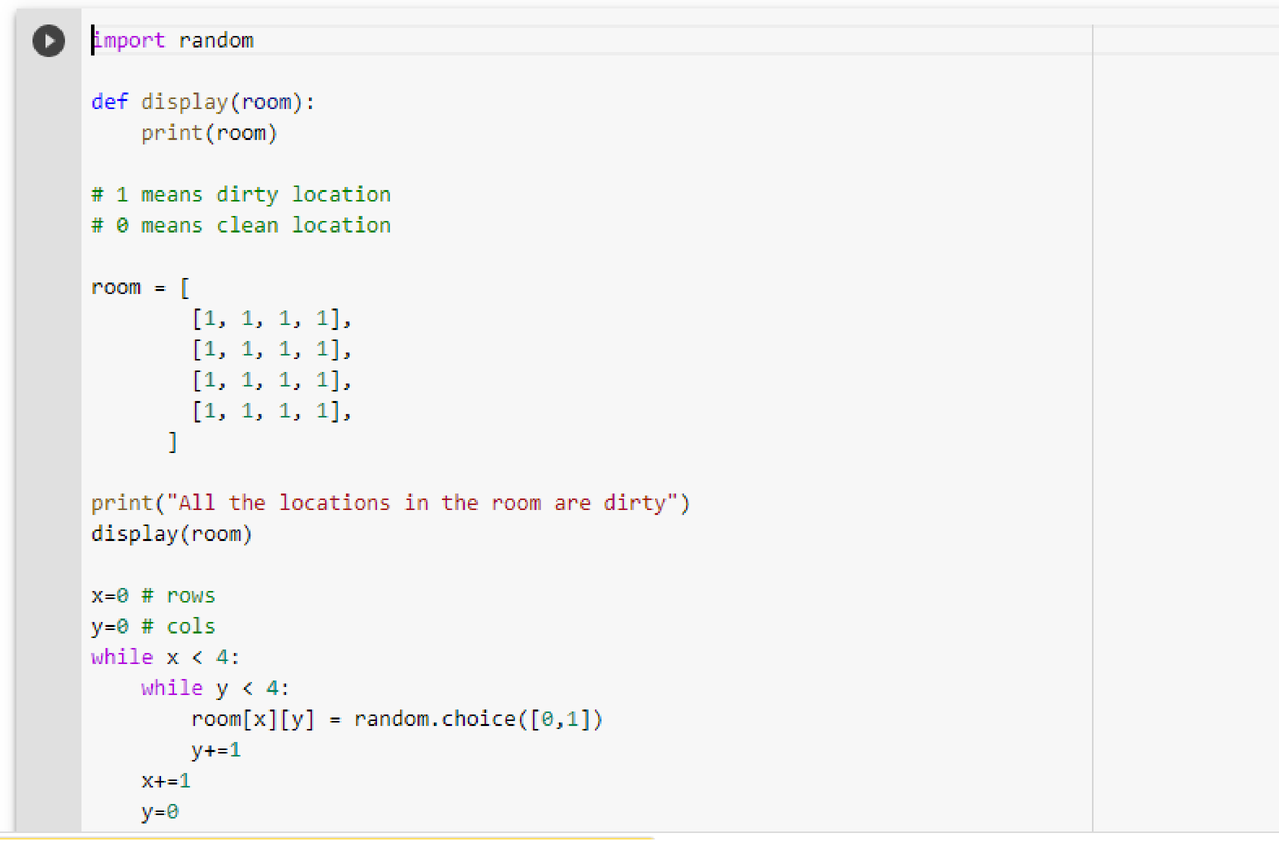
Design an Artificial Intelligence application to implement Intelligent Agent

Code :

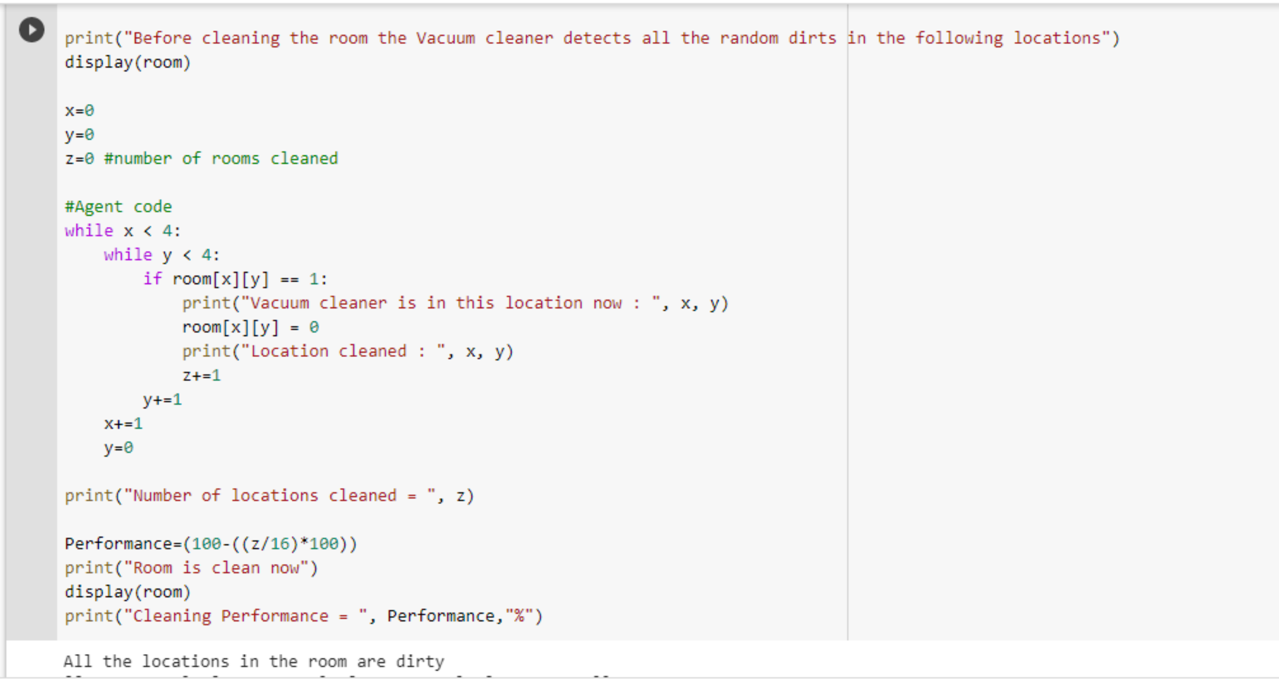
```
import random

def display(room):
        print(room)

# 1 means dirty location
# 0 means clean location

room = [
     [1, 1, 1, 1],
     [1, 1, 1, 1],
     [1, 1, 1, 1],
     [1, 1, 1, 1],
    ]

print("All the locations in the room are dirty")
display(room)

x=0 # rows
y=0 # cols
while x < 4:
        while y < 4:
                room[x][y] = random.choice([0,1])
        y+=1
        x+=1
        y=0

print("Before cleaning the room the Vacuum cleaner detects all the random
    dirts in the following locations")
```

```
display(room)

x=0
y=0
z=0 #number of rooms cleaned

#Agent code
while x < 4:
        while y < 4:
        if room[x][y] == 1:
                print("Vacuum cleaner is in this location now : ", x, y)
                room[x][y] = 0
                print("Location cleaned : ", x, y)
                z+=1
        y+=1
        x+=1
        y=0

print("Number of locations cleaned = ", z)

Performance=(100-((z/16)*100))
print("Room is clean now")
display(room)
print("Cleaning Performance = ", Performance,"%")
```

Output :

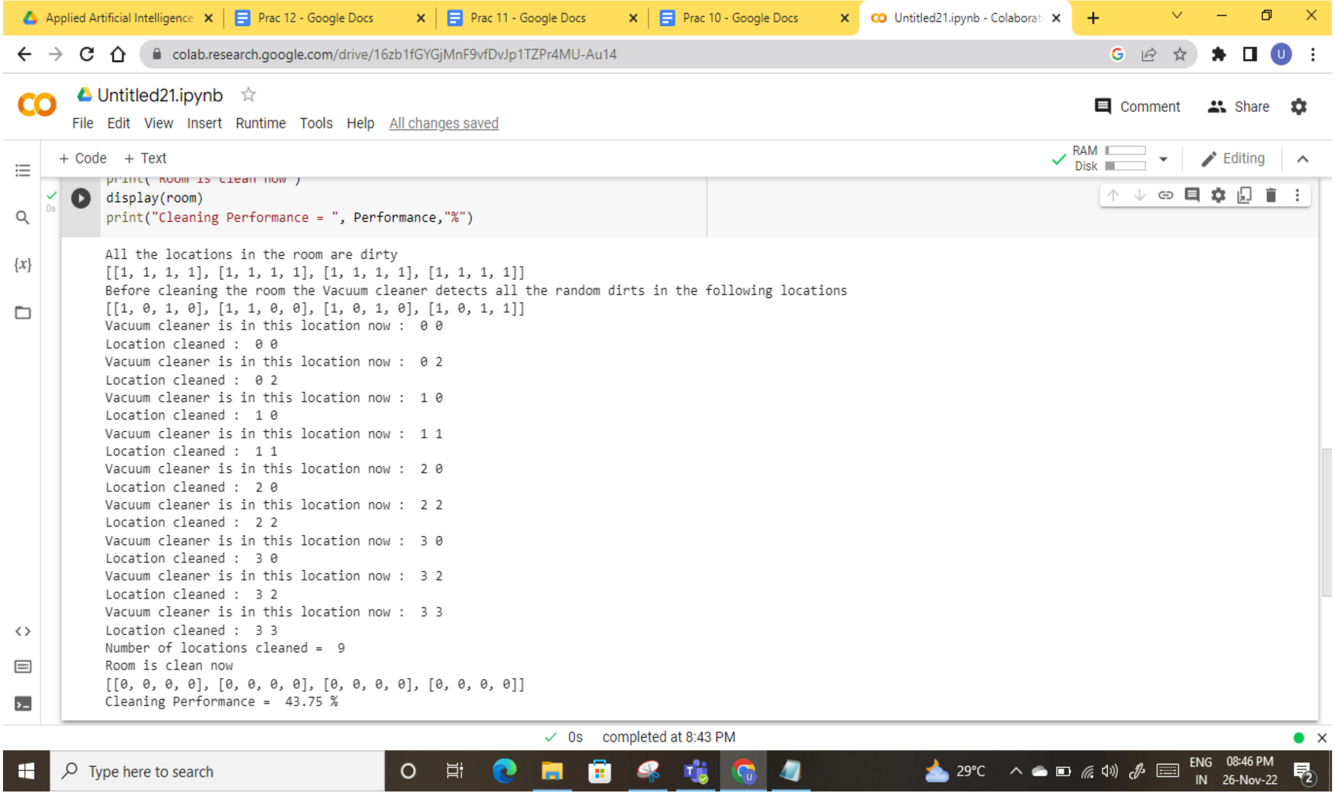


```python
import random

def display(room):
    print(room)

# 1 means dirty location
# 0 means clean location

room = [
        [1, 1, 1, 1],
        [1, 1, 1, 1],
        [1, 1, 1, 1],
        [1, 1, 1, 1],
      ]

print("All the locations in the room are dirty")
display(room)

x=0 # rows
y=0 # cols
while x < 4:
    while y < 4:
        room[x][y] = random.choice([0,1])
        y+=1
    x+=1
    y=0
```



```python
print("Before cleaning the room the Vacuum cleaner detects all the random dirts in the following locations")
display(room)

x=0
y=0
z=0 #number of rooms cleaned

#Agent code
while x < 4:
    while y < 4:
        if room[x][y] == 1:
            print("Vacuum cleaner is in this location now : ", x, y)
            room[x][y] = 0
            print("Location cleaned : ", x, y)
            z+=1
        y+=1
    x+=1
    y=0

print("Number of locations cleaned = ", z)

Performance=(100-((z/16)*100))
print("Room is clean now")
display(room)
print("Cleaning Performance = ", Performance,"%")
```

```
All the locations in the room are dirty
```


Applied Artificial Intelligence
Prac 12 - Google Docs
Prac 11 - Google Docs
Prac 10 - Google Docs
Untitled21.ipynb - Colaborat
colab.research.google.com/drive/16zb1fGYGjMnF9vfDvJp1TZPr4MU-Au14
Untitled21.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM
Disk
Editing

```
print("Room is clean now")
display(room)
print("Cleaning Performance = ", Performance,"%")
```

```
All the locations in the room are dirty
[[1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1]]
Before cleaning the room the Vacuum cleaner detects all the random dirts in the following locations
[[1, 0, 1, 0], [1, 1, 0, 0], [1, 0, 1, 0], [1, 0, 1, 1]]
Vacuum cleaner is in this location now :  0 0
Location cleaned :  0 0
Vacuum cleaner is in this location now :  0 2
Location cleaned :  0 2
Vacuum cleaner is in this location now :  1 0
Location cleaned :  1 0
Vacuum cleaner is in this location now :  1 1
Location cleaned :  1 1
Vacuum cleaner is in this location now :  2 0
Location cleaned :  2 0
Vacuum cleaner is in this location now :  2 2
Location cleaned :  2 2
Vacuum cleaner is in this location now :  3 0
Location cleaned :  3 0
Vacuum cleaner is in this location now :  3 2
Location cleaned :  3 2
Vacuum cleaner is in this location now :  3 3
Location cleaned :  3 3
Number of locations cleaned =  9
Room is clean now
[[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
Cleaning Performance =  43.75 %
```