Machine Learning

Course Work

MAHDSE212F-004 S.B.C. Sanjaya

Higher Diploma in Software Engineering 21.2F

Supervisor: Mr. Tharindu Adikari

National Institute of Business Management

2023

“The course work report is submitted in partial fulfilment of the requirement of the Machine Learning subject for Higher Diploma in Software Engineering of National Institute of Business Management.”

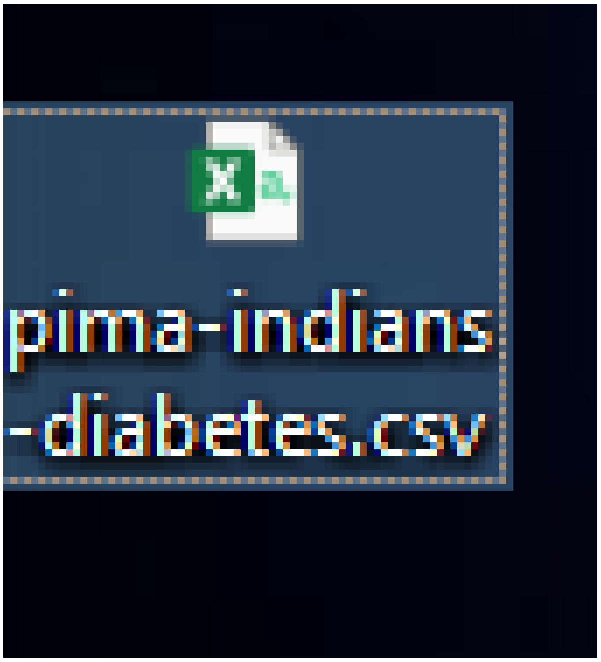
Contents

Introduction

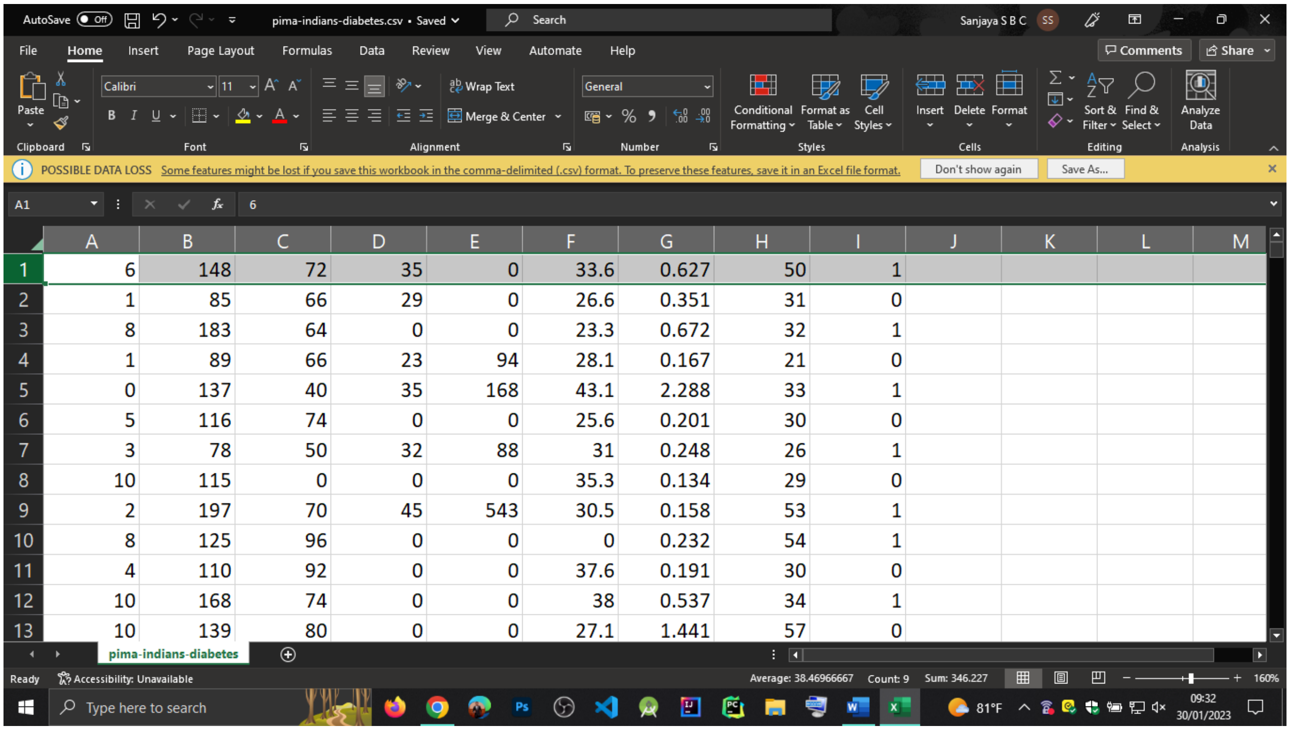
I select a dataset with 1000> records data included one.

Course Work Question Answers

- First I found a dataset with more than 1000 dataset records.



- This is the Prima-Indians_Diabetic.csv file with more than 1000 rows.

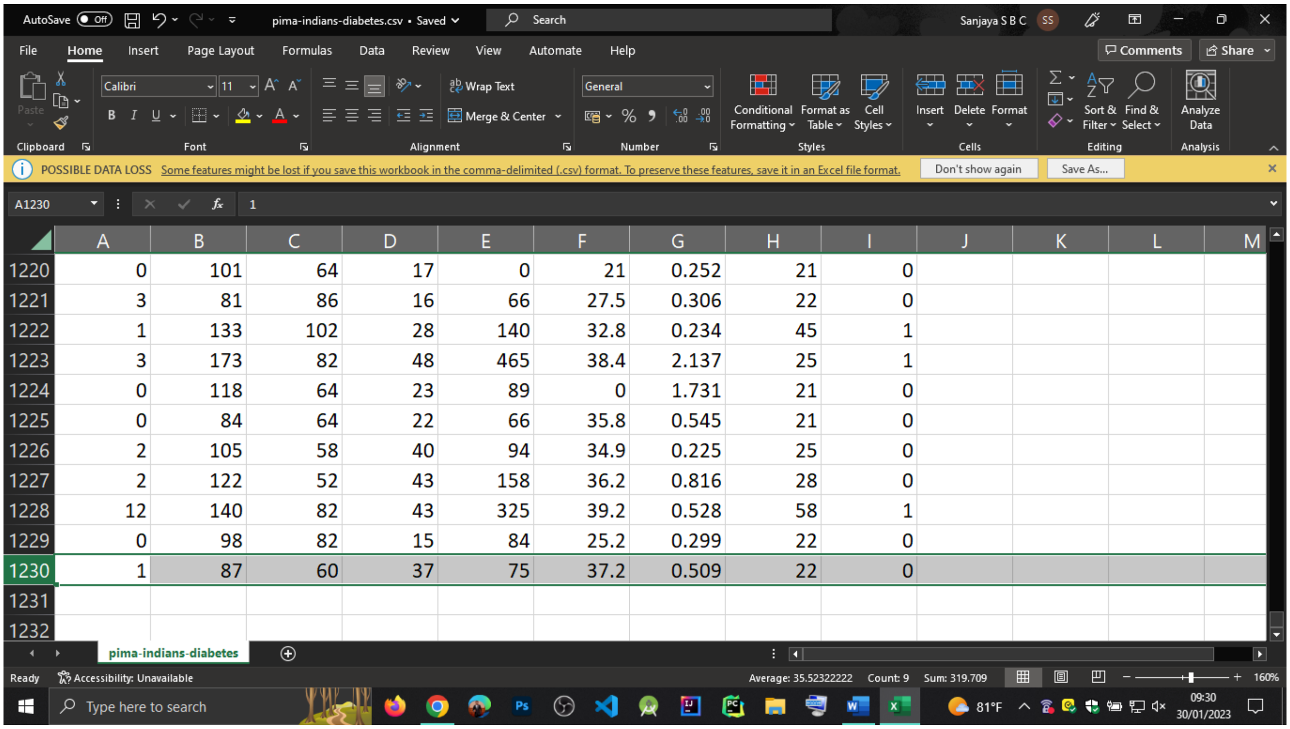


POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

A1 | 6

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	6	148	72	35	0	33.6	0.627	50	1				
2	1	85	66	29	0	26.6	0.351	31	0				
3	8	183	64	0	0	23.3	0.672	32	1				
4	1	89	66	23	94	28.1	0.167	21	0				
5	0	137	40	35	168	43.1	2.288	33	1				
6	5	116	74	0	0	25.6	0.201	30	0				
7	3	78	50	32	88	31	0.248	26	1				
8	10	115	0	0	0	35.3	0.134	29	0				
9	2	197	70	45	543	30.5	0.158	53	1				
10	8	125	96	0	0	0	0.232	54	1				
11	4	110	92	0	0	37.6	0.191	30	0				
12	10	168	74	0	0	38	0.537	34	1				
13	10	139	80	0	0	27.1	1.441	57	0				

Average: 38.46966667 Count: 9 Sum: 346.227



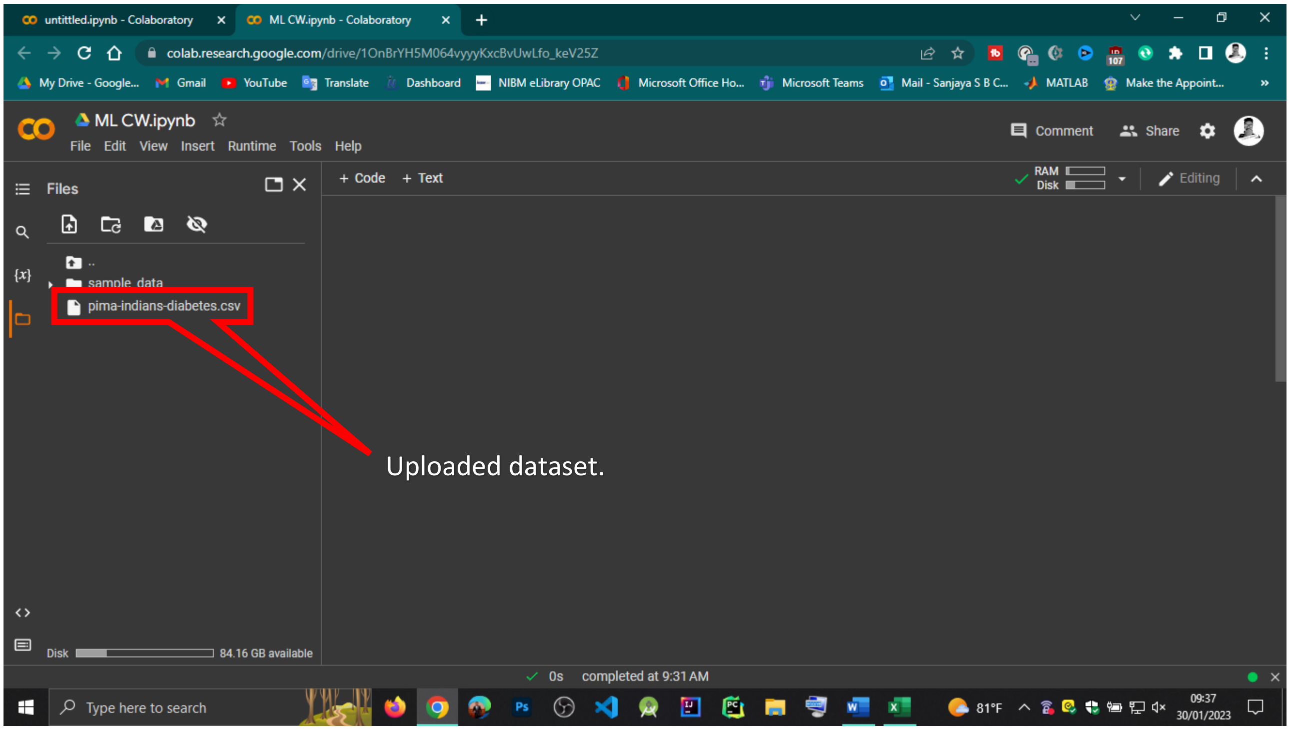
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

A1230 | 1

	A	B	C	D	E	F	G	H	I	J	K	L	M
1220	0	101	64	17	0	21	0.252	21	0				
1221	3	81	86	16	66	27.5	0.306	22	0				
1222	1	133	102	28	140	32.8	0.234	45	1				
1223	3	173	82	48	465	38.4	2.137	25	1				
1224	0	118	64	23	89	0	1.731	21	0				
1225	0	84	64	22	66	35.8	0.545	21	0				
1226	2	105	58	40	94	34.9	0.225	25	0				
1227	2	122	52	43	158	36.2	0.816	28	0				
1228	12	140	82	43	325	39.2	0.528	58	1				
1229	0	98	82	15	84	25.2	0.299	22	0				
1230	1	87	60	37	75	37.2	0.509	22	0				
1231													
1232													

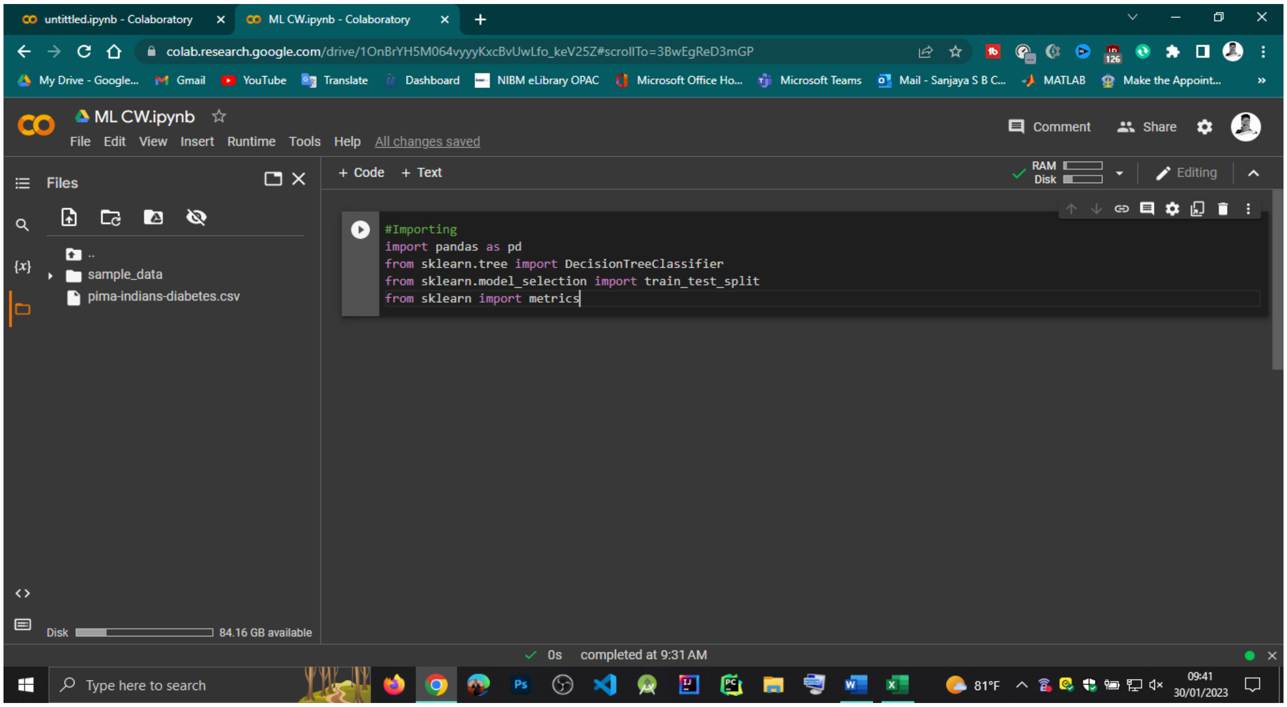
Average: 35.52322222 Count: 9 Sum: 319.709

- First, I go to Google Colab and open a new notebook.
- Then I upload my downloaded Prima-Indian-Diabeties.csv file to the Google Colab.



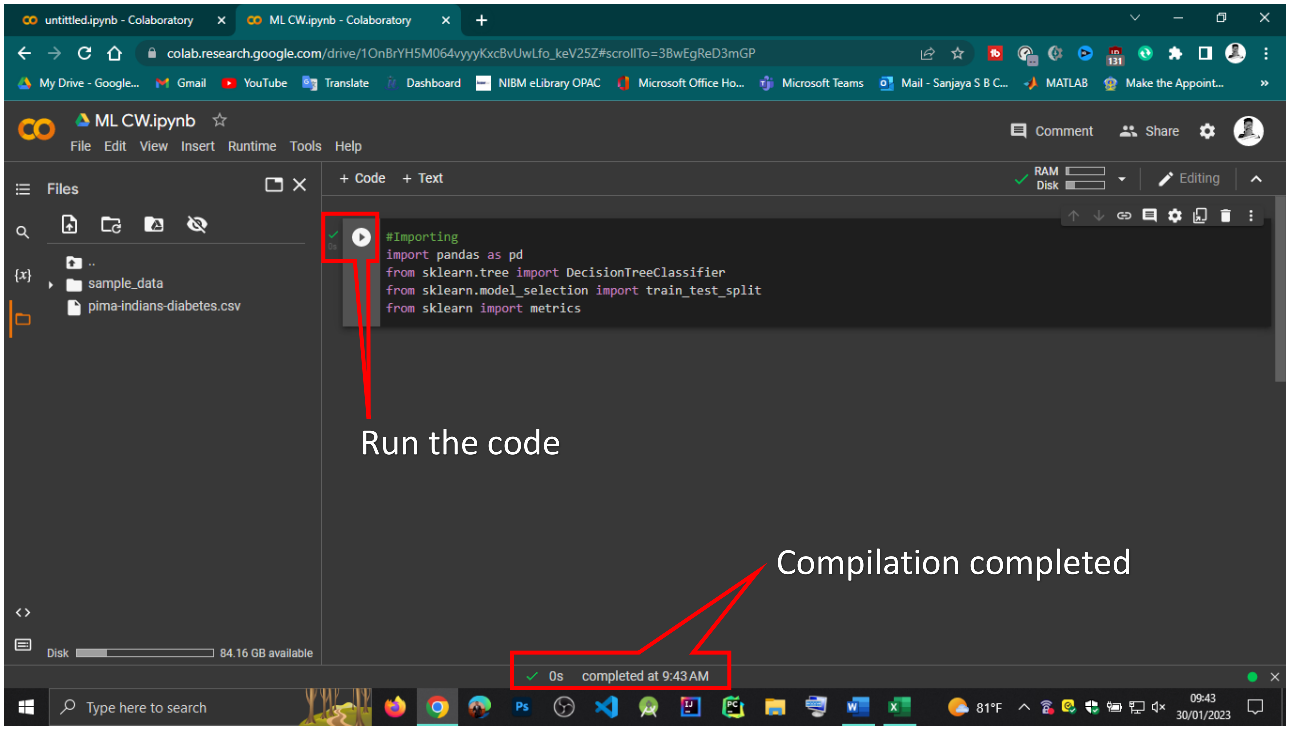
Uploaded dataset.

- Then I open a new code and import pandas as following.



```
#Importing
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

- Then I run this code as below.



```python
#Importing
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

0s completed at 9:43 AM

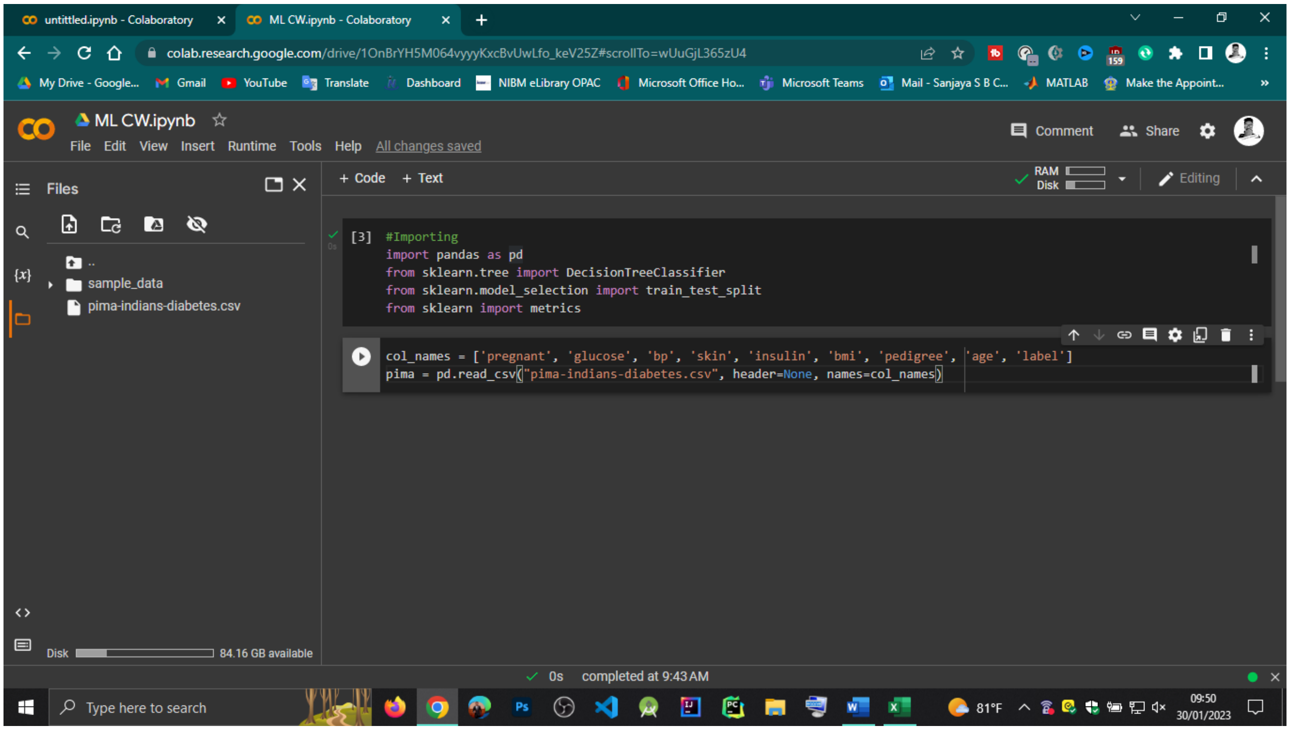
untittled.ipynb - Colaboratory
ML CW.ipynb - Colaboratory
colab.research.google.com/drive/1OnBrYH5M064vyyyKxcBvUwLfo_keV25Z#scrollTo=3BwEgReD3mGP
ML CW.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
Files
+ Code + Text
RAM Disk
Editing
sample_data
pima-indians-diabetes.csv
Disk 84.16 GB available
Run the code
Compilation completed
Type here to search
81°F
09:43
30/01/2023

- Then the compilation of above code completed successfully.

- This is the code I mention above.

```python
#Importing
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

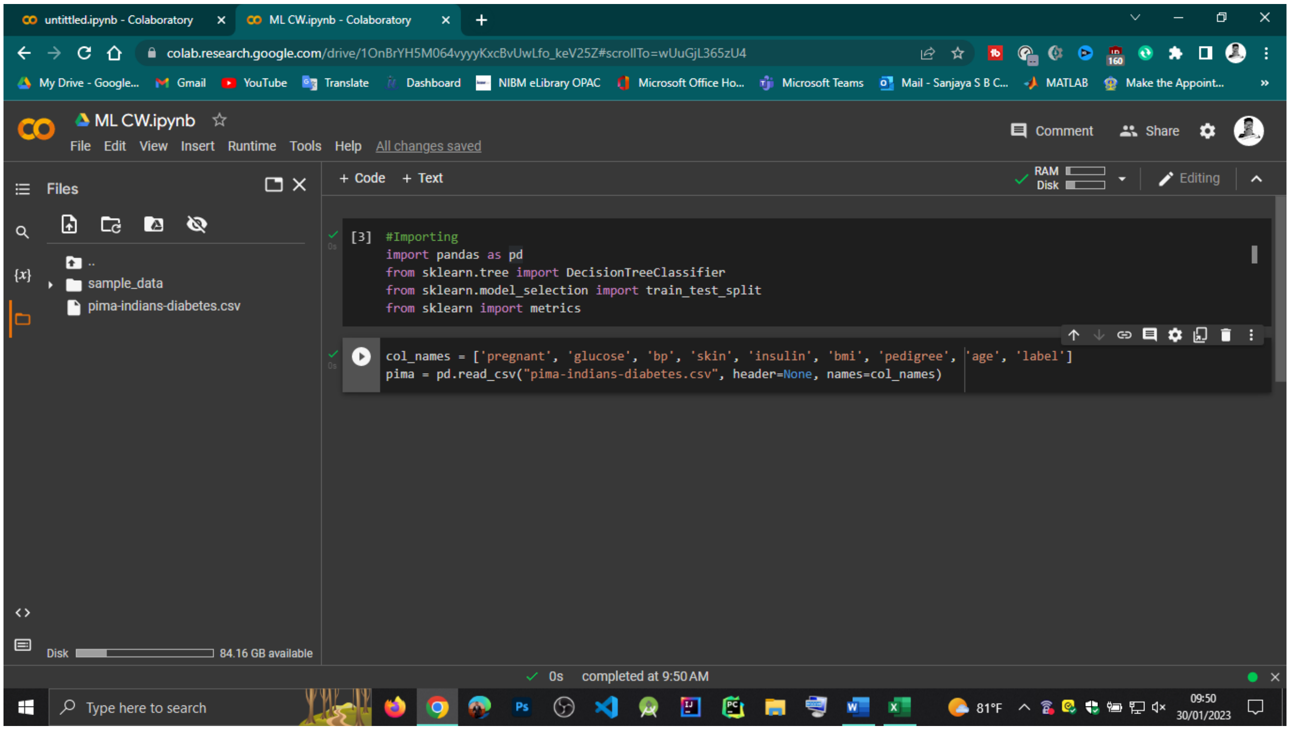
- Then I read and test the file I uploaded. It can be coded by click using a new code + icon.


untittled.ipynb - Colaboratory
ML CW.ipynb - Colaboratory
colab.research.google.com/drive/1OnBrYH5M064vyyyKxcBvUwLfo_keV25Z#scrollTo=wUuGjL365zU4
ML CW.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
+ Code + Text
RAM Disk
Editing
sample_data
pima-indians-diabetes.csv
Disk 84.16 GB available
0s completed at 9:43 AM
Type here to search
81°F
09:50
30/01/2023

```
[3] #Importing
    import pandas as pd
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.model_selection import train_test_split
    from sklearn import metrics
```

```
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']
pima = pd.read_csv("pima-indians-diabetes.csv", header=None, names=col_names)
```

- Then I run this code as below.


untittled.ipynb - Colaboratory
ML CW.ipynb - Colaboratory
colab.research.google.com/drive/1OnBrYH5M064vyyyKxcBvUwLfo_keV25Z#scrollTo=wUuGjL365zU4
ML CW.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
+ Code + Text
RAM Disk
Editing
sample_data
pima-indians-diabetes.csv
Disk 84.16 GB available
0s completed at 9:50 AM
Type here to search
81°F
09:50
30/01/2023

```
[3] #Importing
    import pandas as pd
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.model_selection import train_test_split
    from sklearn import metrics
```

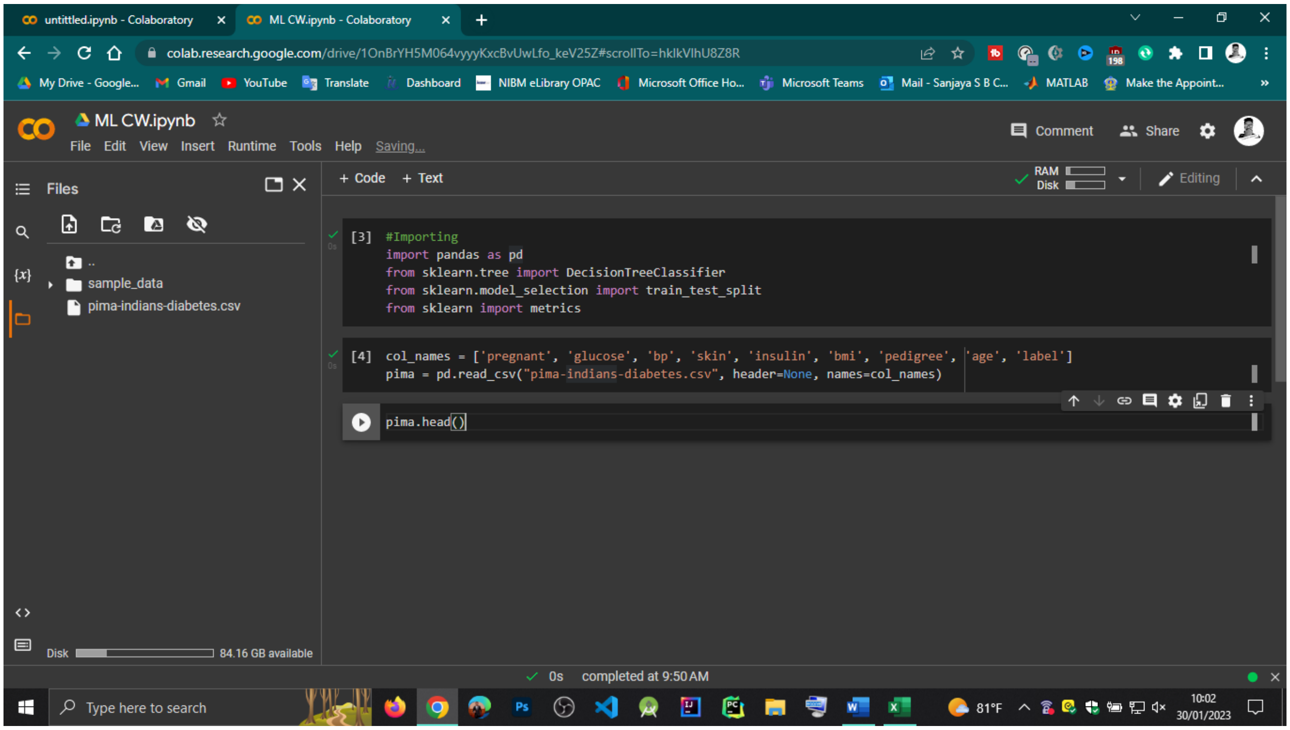
```
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']
pima = pd.read_csv("pima-indians-diabetes.csv", header=None, names=col_names)
```

- Then the compilation of above code completed successfully.

- This is the code I mention above.

```
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']
pima = pd.read_csv("pima-indians-diabetes.csv", header=None, names=col_names)
```

- Then I enter a head and get the details of the table. It can be coded by click using a new code + icon.

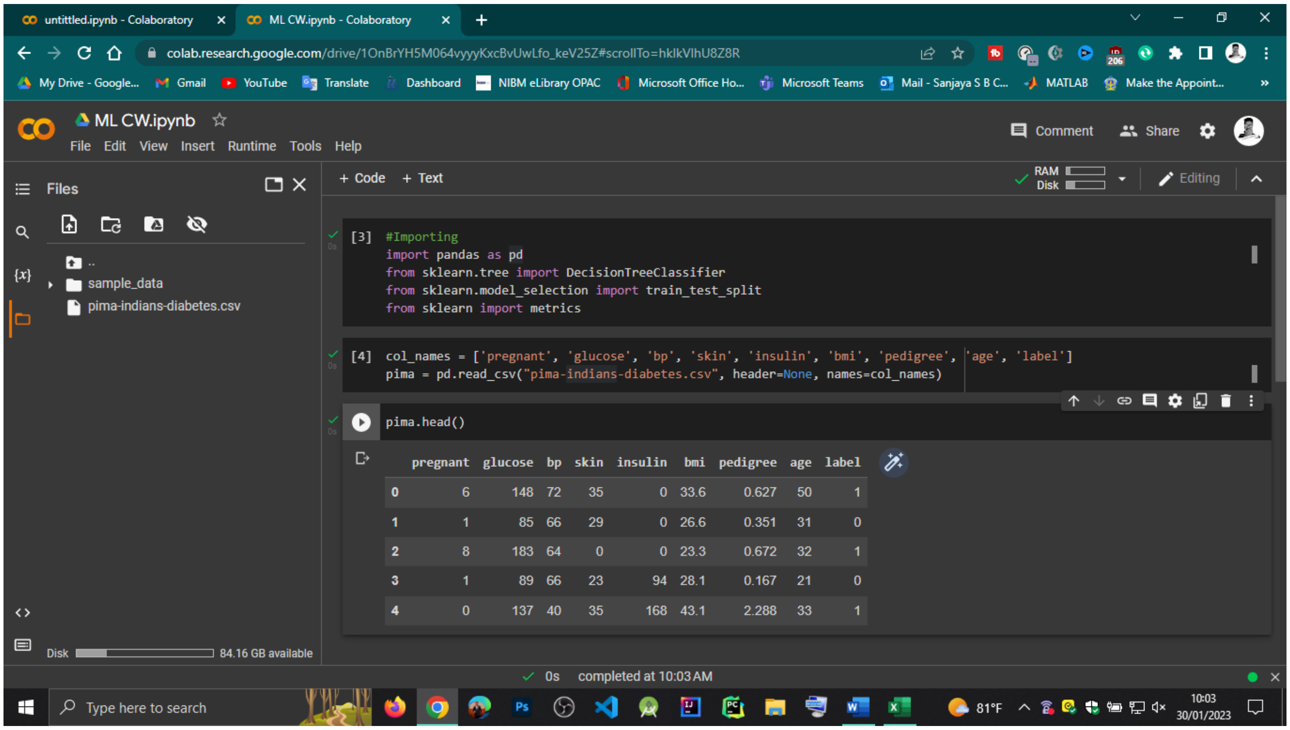


```
[3] #Importing
    import pandas as pd
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.model_selection import train_test_split
    from sklearn import metrics
```

```
[4] col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']
    pima = pd.read_csv("pima-indians-diabetes.csv", header=None, names=col_names)
```

```
pima.head()
```

- Then I run this code as below.


untittled.ipynb - Colaboratory
ML CW.ipynb - Colaboratory
colab.research.google.com/drive/1OnBrYH5M064vyyyKxcBvUwLfo_keV25Z#scrollTo=hkIkVlhU8Z8R
ML CW.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
Files
+ Code
+ Text
RAM
Disk
Editing
sample_data
pima-indians-diabetes.csv
Disk 84.16 GB available
0s completed at 10:03 AM
Type here to search
81°F
10:03
30/01/2023

```
[3] #Importing
    import pandas as pd
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.model_selection import train_test_split
    from sklearn import metrics
```

```
[4] col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']
    pima = pd.read_csv("pima-indians-diabetes.csv", header=None, names=col_names)
```

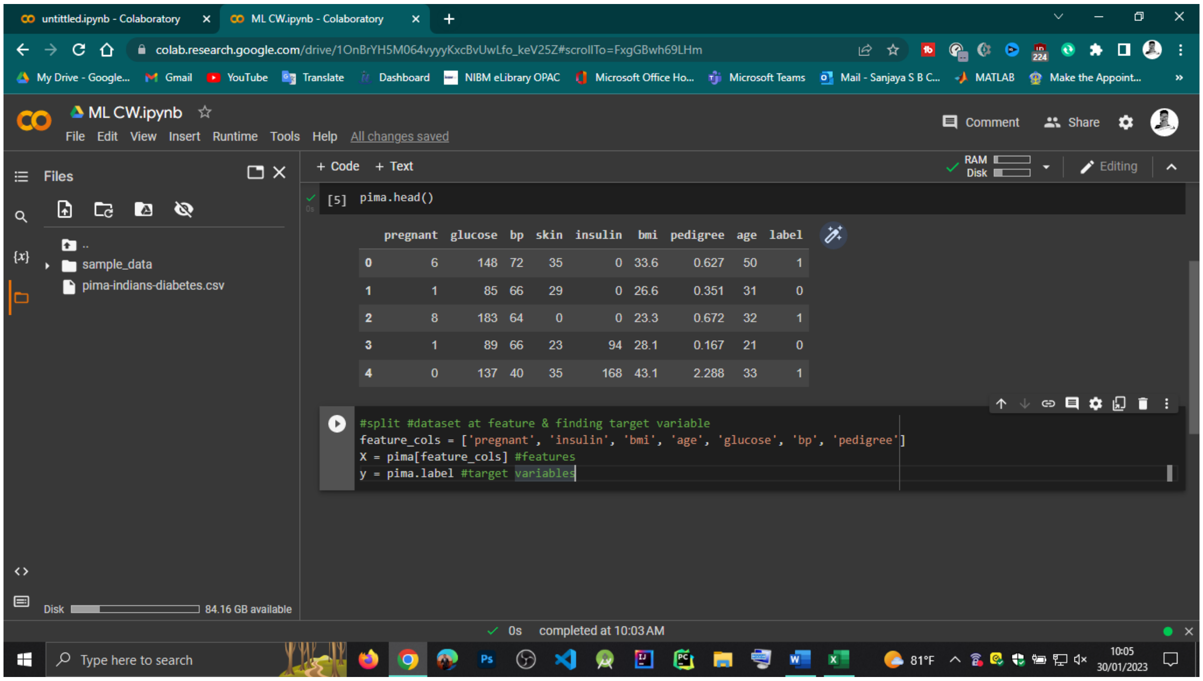
```
pima.head()
```

```
   pregnant  glucose  bp  skin  insulin   bmi  pedigree  age  label
0         6      148  72    35        0  33.6     0.627   50      1
1         1       85  66    29        0  26.6     0.351   31      0
2         8      183  64     0        0  23.3     0.672   32      1
3         1       89  66    23       94  28.1     0.167   21      0
4         0      137  40    35      168  43.1     2.288   33      1
```

- This is the code I mention above.

```
pima.head()
```

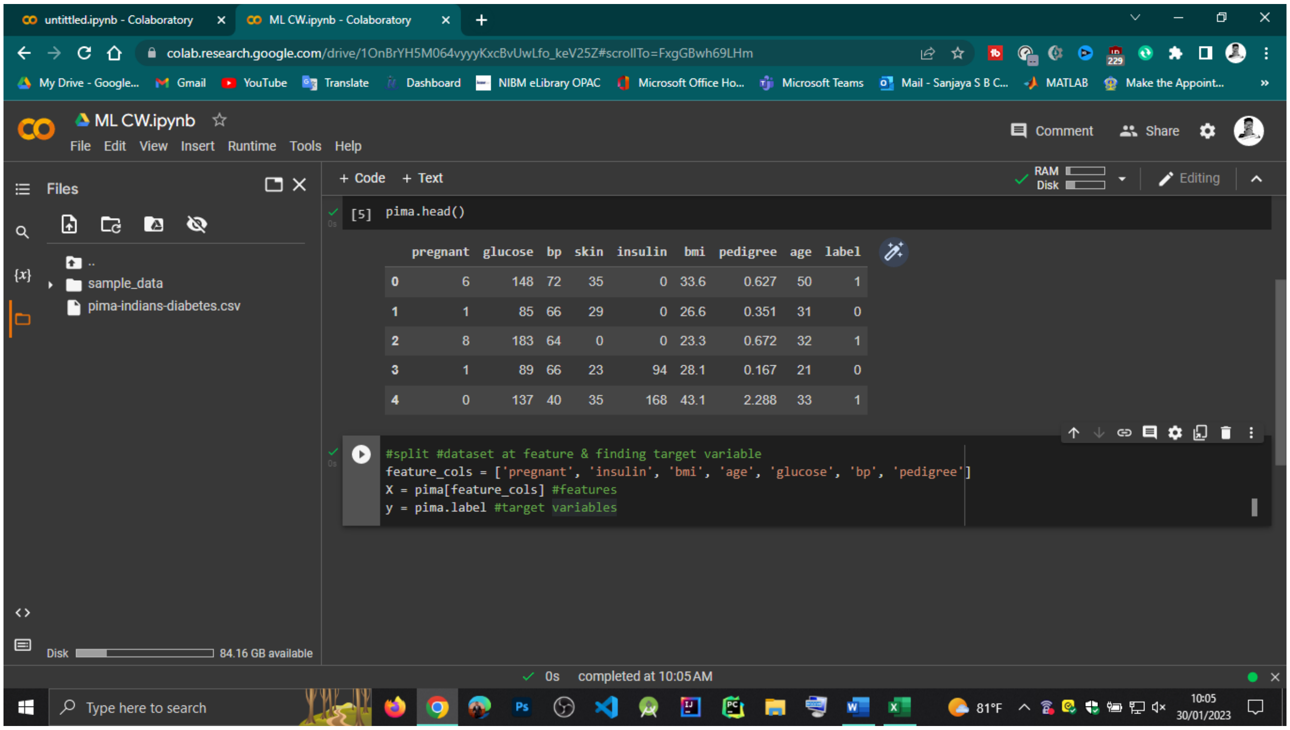
- Then I split dataset in feature & finding target variable. It can be coded by click using a new code + icon.



```
[5] pima.head()

   pregnant  glucose  bp  skin  insulin   bmi  pedigree  age  label
0         6      148  72    35        0  33.6     0.627   50      1
1         1       85  66    29        0  26.6     0.351   31      0
2         8      183  64     0        0  23.3     0.672   32      1
3         1       89  66    23       94  28.1     0.167   21      0
4         0      137  40    35      168  43.1     2.288   33      1
```

```
#split #dataset at feature & finding target variable
feature_cols = ['pregnant', 'insulin', 'bmi', 'age', 'glucose', 'bp', 'pedigree']
X = pima[feature_cols] #features
y = pima.label #target variables
```

- Then I run this code as below.



```
[5] pima.head()

   pregnant  glucose  bp  skin  insulin   bmi  pedigree  age  label
0         6      148  72    35        0  33.6     0.627   50      1
1         1       85  66    29        0  26.6     0.351   31      0
2         8      183  64     0        0  23.3     0.672   32      1
3         1       89  66    23       94  28.1     0.167   21      0
4         0      137  40    35      168  43.1     2.288   33      1
```

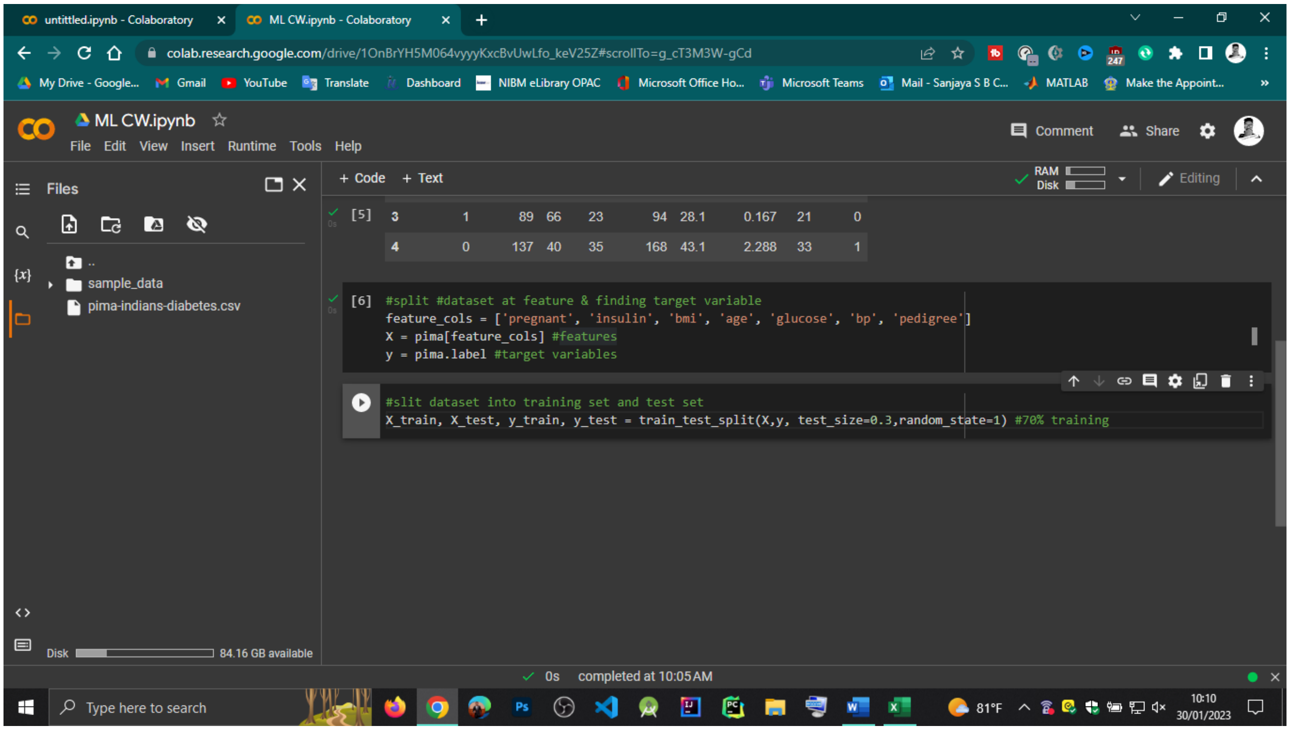
```
#split #dataset at feature & finding target variable
feature_cols = ['pregnant', 'insulin', 'bmi', 'age', 'glucose', 'bp', 'pedigree']
X = pima[feature_cols] #features
y = pima.label #target variables
```

- This is the code I mention above.

```
#split #dataset at feature & finding target variable
```

```
feature_cols = ['pregnant', 'insulin', 'bmi', 'age', 'glucose', 'bp', 'pedigree']
X = pima[feature_cols] #features
y = pima.label #target variables
```

- Then I slit dataset into training set and test set. It can be coded by click using a new code + icon.

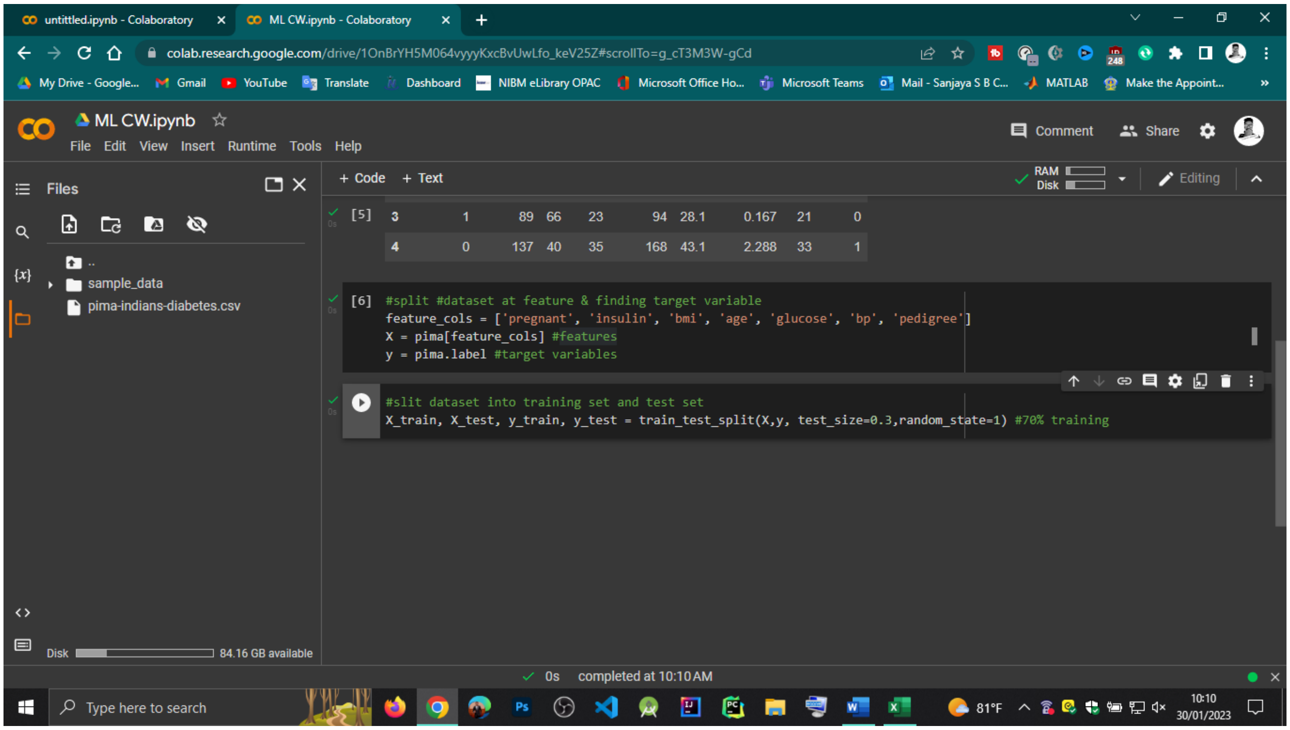


```
[5]  3    1    89   66   23    94   28.1    0.167   21    0
     4    0   137   40   35   168   43.1    2.288   33    1
```

```
[6] #split #dataset at feature & finding target variable
    feature_cols = ['pregnant', 'insulin', 'bmi', 'age', 'glucose', 'bp', 'pedigree']
    X = pima[feature_cols] #features
    y = pima.label #target variables
```

```
#slit dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,random_state=1) #70% training
```

- Then I run this code as below.



```
[5]  3    1   89  66  23   94  28.1  0.167  21  0
     4    0  137  40  35  168  43.1  2.288  33  1
```

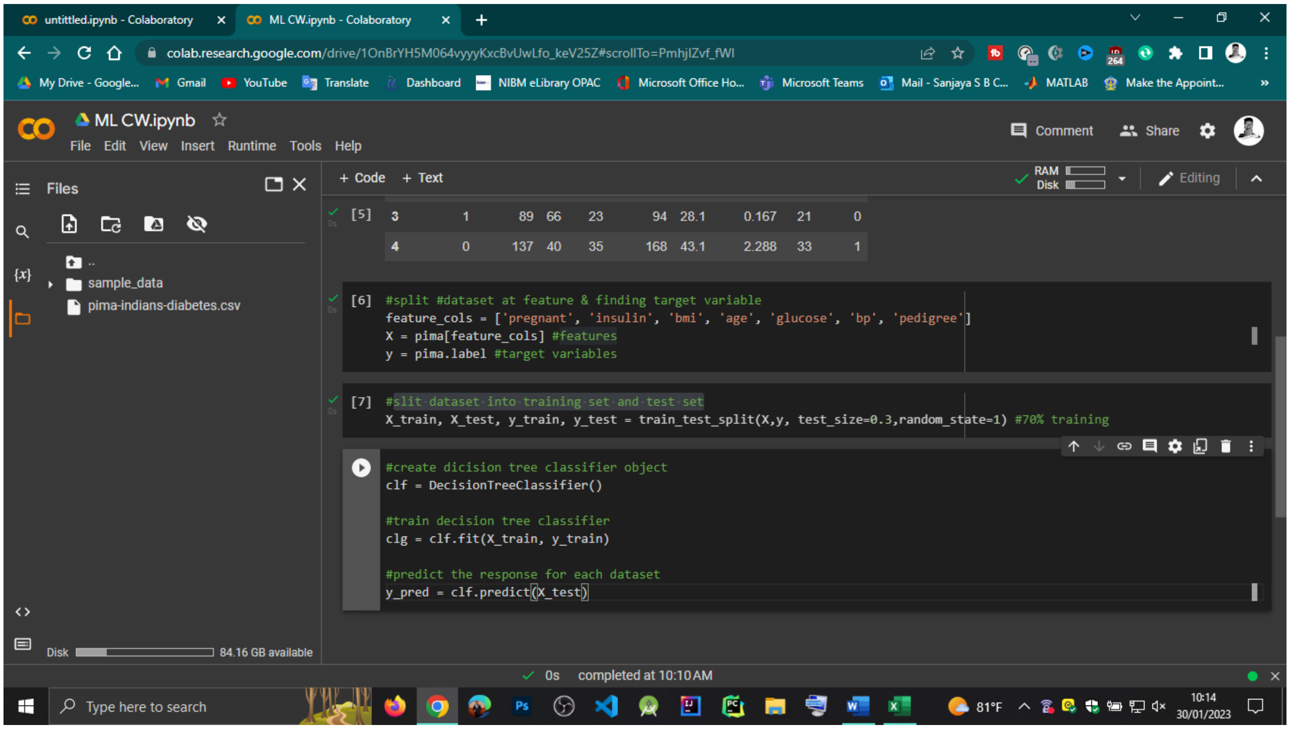
```
[6] #split #dataset at feature & finding target variable
    feature_cols = ['pregnant', 'insulin', 'bmi', 'age', 'glucose', 'bp', 'pedigree']
    X = pima[feature_cols] #features
    y = pima.label #target variables
```

```
#slit dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,random_state=1) #70% training
```

- This is the code I mention above.

```
• #slit dataset into training set and test set
• X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3
  ,random_state=1) #70% training
```

- Then I create decision tree classifier object, train decision tree classifier and predict the response for each dataset. It can be coded by click using a new code + icon.


ML CW.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Files
sample_data
pima-indians-diabetes.csv
Disk 84.16 GB available

```
[5]  3   1   89  66  23   94  28.1  0.167  21  0
     4   0  137  40  35  168  43.1  2.288  33  1
```

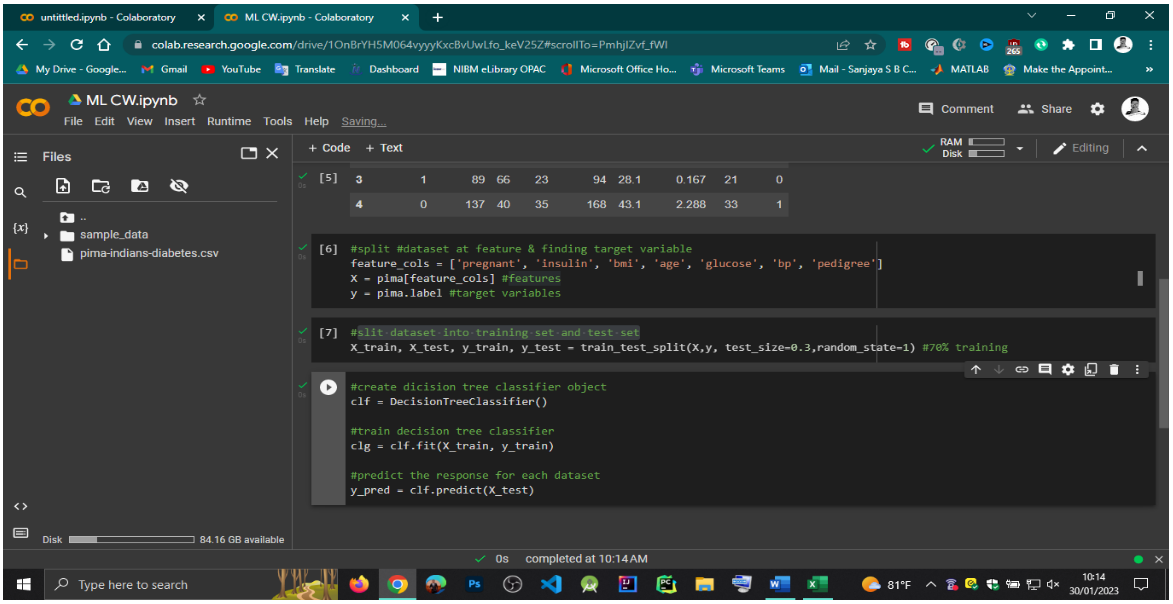
```
[6] #split #dataset at feature & finding target variable
feature_cols = ['pregnant', 'insulin', 'bmi', 'age', 'glucose', 'bp', 'pedigree']
X = pima[feature_cols] #features
y = pima.label #target variables
```

```
[7] #slit dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,random_state=1) #70% training
```

```
#create dicision tree classifier object
clf = DecisionTreeClassifier()

#train decision tree classifier
clg = clf.fit(X_train, y_train)

#predict the response for each dataset
y_pred = clf.predict(X_test)
```

0s completed at 10:10 AM

- Then I run this code as below.


ML CW.ipynb
File Edit View Insert Runtime Tools Help Saving...
+ Code + Text
Files
sample_data
pima-indians-diabetes.csv
Disk 84.16 GB available

```
[5]  3   1   89  66  23   94  28.1  0.167  21  0
     4   0  137  40  35  168  43.1  2.288  33  1
```

```
[6] #split #dataset at feature & finding target variable
feature_cols = ['pregnant', 'insulin', 'bmi', 'age', 'glucose', 'bp', 'pedigree']
X = pima[feature_cols] #features
y = pima.label #target variables
```

```
[7] #slit dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,random_state=1) #70% training
```

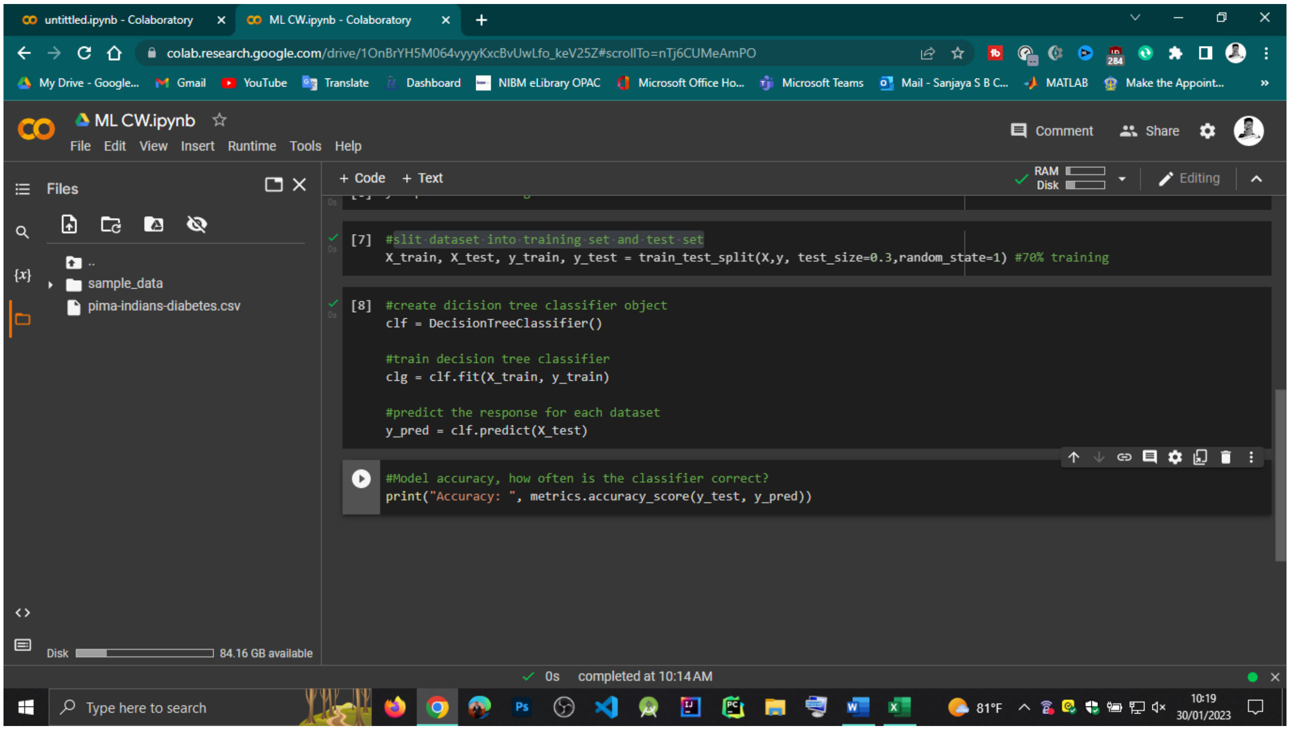
```
#create dicision tree classifier object
clf = DecisionTreeClassifier()

#train decision tree classifier
clg = clf.fit(X_train, y_train)

#predict the response for each dataset
y_pred = clf.predict(X_test)
```

0s completed at 10:14 AM

- This is the code I mention above.

```
#create dicision tree classifier object
clf = DecisionTreeClassifier()

#train decision tree classifier
clg = clf.fit(X_train, y_train)

#predict the response for each dataset
y_pred = clf.predict(X_test)
```

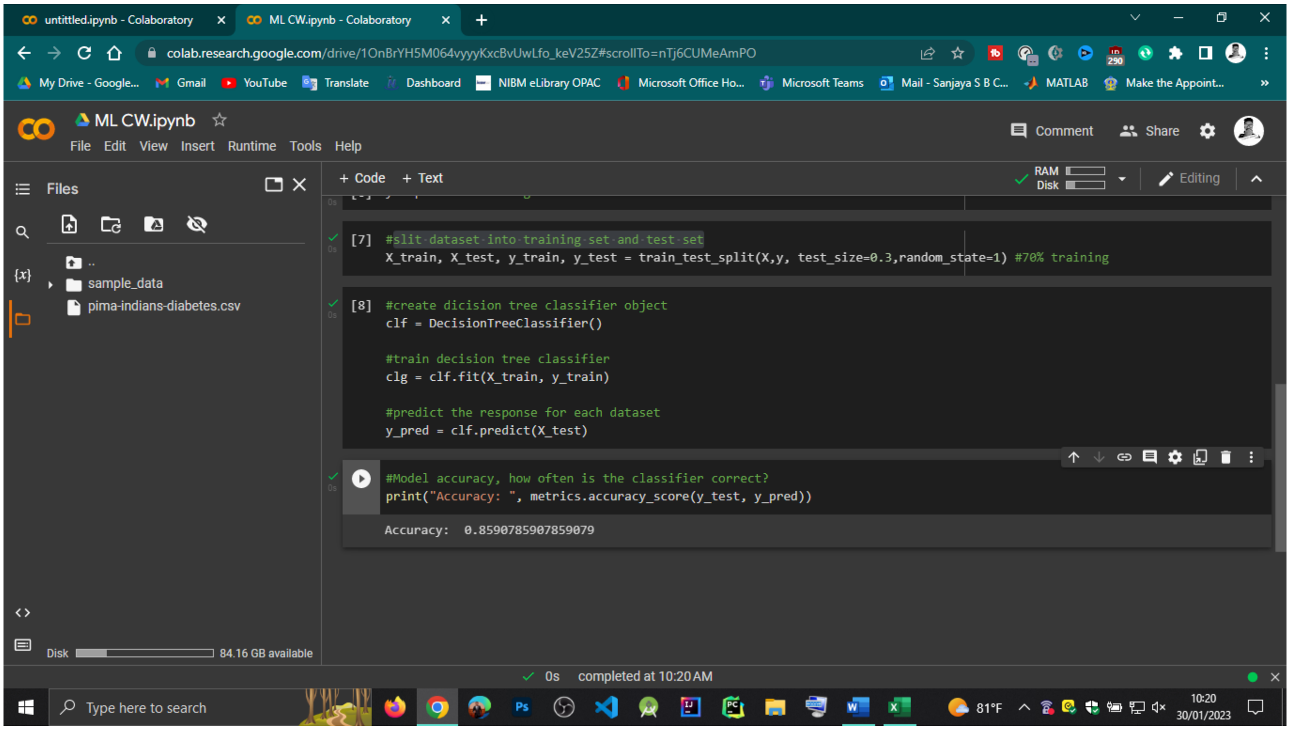
- Then I Model accuracy, how often is the classifier correct. It can be coded by click using a new code + icon.



untittled.ipynb - Colaboratory
ML CW.ipynb - Colaboratory
colab.research.google.com/drive/1OnBrYH5M064vyyyKxcBvUwLfo_keV25Z#scrollTo=nTj6CUMeAmPO
ML CW.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
Files
sample_data
pima-indians-diabetes.csv
+ Code + Text
RAM
Disk
Editing
Disk 84.16 GB available
0s completed at 10:14 AM
Type here to search
81°F
10:19
30/01/2023

```
[7] #slit dataset into training set and test set
    X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,random_state=1) #70% training

[8] #create dicision tree classifier object
    clf = DecisionTreeClassifier()

    #train decision tree classifier
    clg = clf.fit(X_train, y_train)

    #predict the response for each dataset
    y_pred = clf.predict(X_test)

    #Model accuracy, how often is the classifier correct?
    print("Accuracy: ", metrics.accuracy_score(y_test, y_pred))
```

- Then I run this code.



```
[7] #slit dataset into training set and test set
    X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,random_state=1) #70% training

[8] #create dicision tree classifier object
    clf = DecisionTreeClassifier()

    #train decision tree classifier
    clg = clf.fit(X_train, y_train)

    #predict the response for each dataset
    y_pred = clf.predict(X_test)

    #Model accuracy, how often is the classifier correct?
    print("Accuracy: ", metrics.accuracy_score(y_test, y_pred))

    Accuracy:  0.8590785907859079
```

- This is the code I mention above.

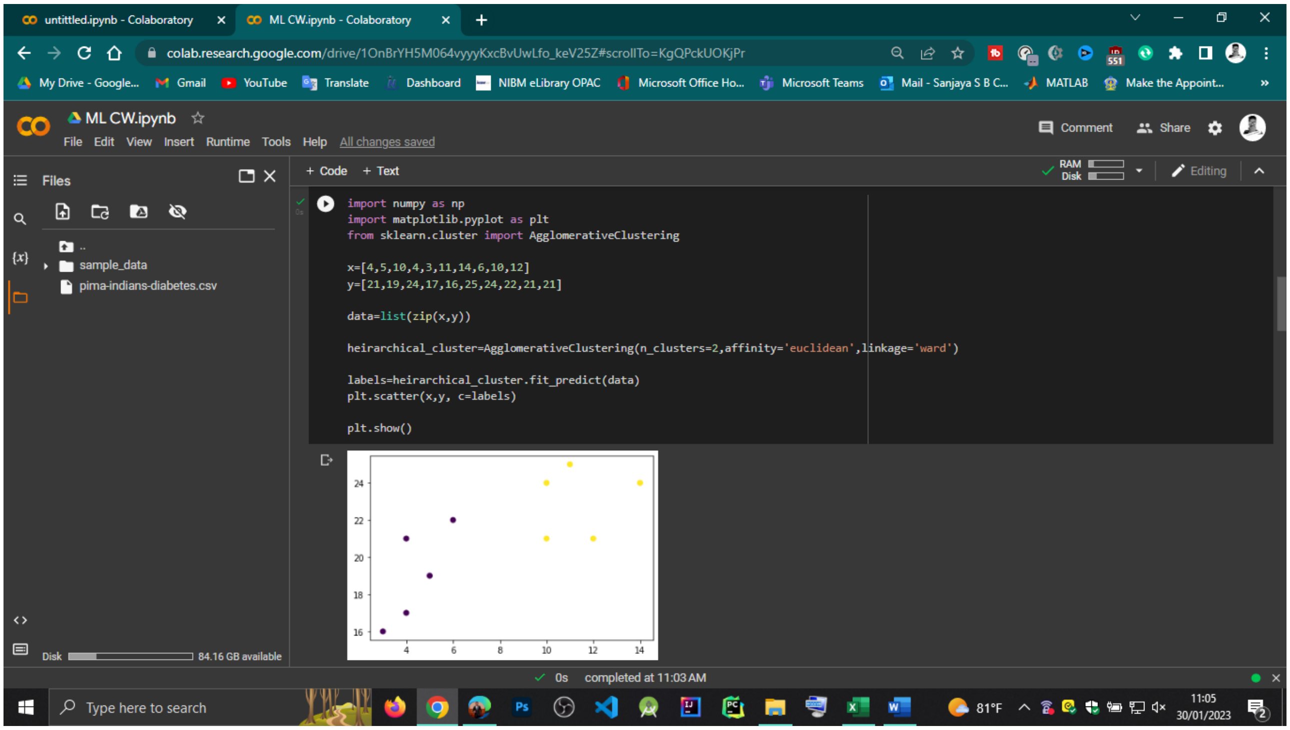
```
#Model accuracy, how often is the classifier correct?
print("Accuracy: ", metrics.accuracy_score(y_test, y_pred))
```

- Finally, I got the accuracy value of this dataset as below.

```
Accuracy:  0.8590785907859079
```

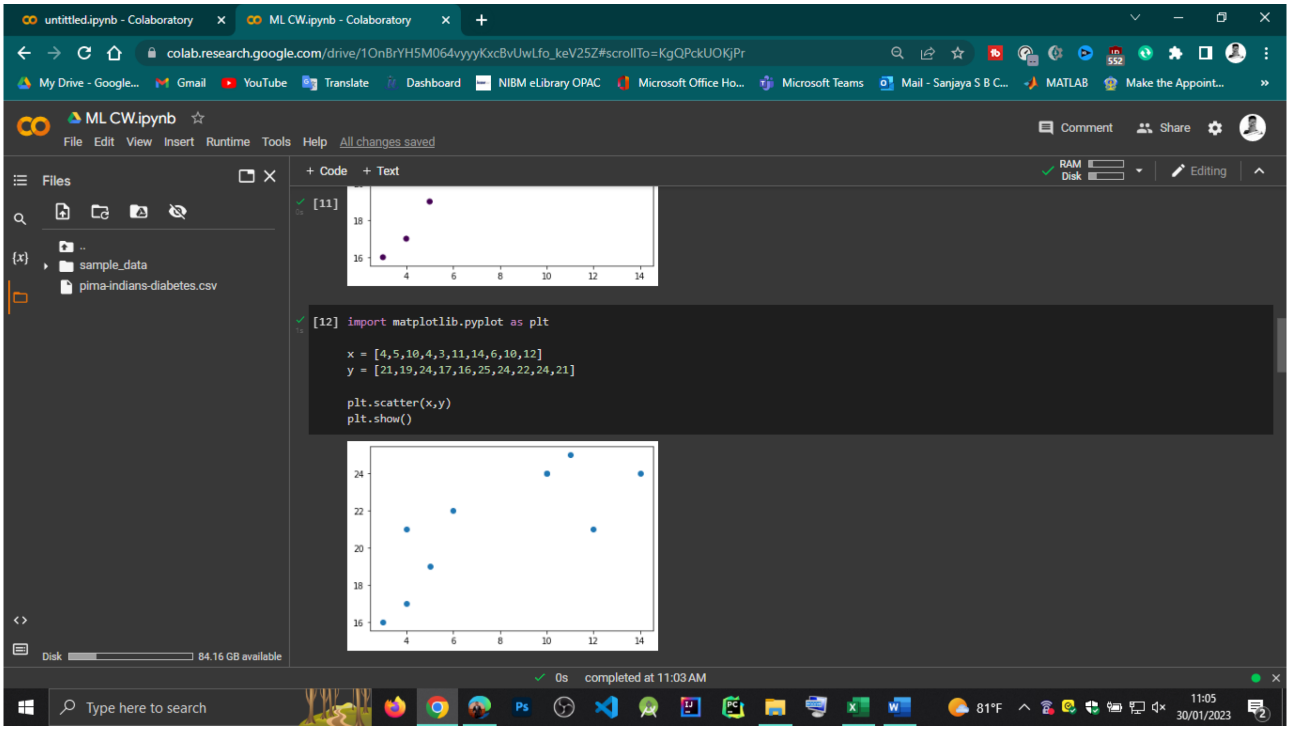
- And I did some Machine Learning Algorithms as additional.

- Then I run codes as follows and did some Machine Learning Algorithms.



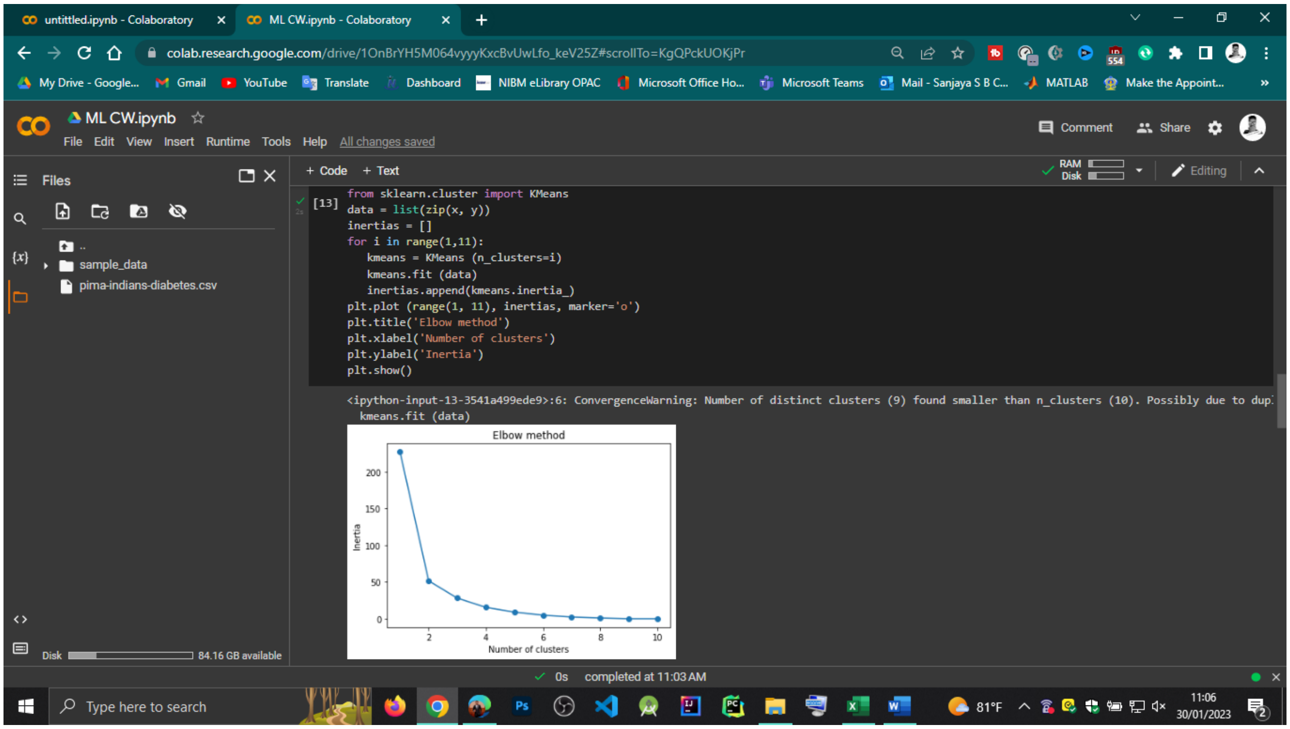
untittled.ipynb - Colaboratory
ML CW.ipynb - Colaboratory
colab.research.google.com/drive/1OnBrYH5M064vyyyKxcBvUwLfo_keV25Z#scrollTo=KgQPckUOKjPr
ML CW.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Editing
Files
sample_data
pima-indians-diabetes.csv
Disk 84.16 GB available
24
22
20
18
16
4
6
8
10
12
14
0s completed at 11:03 AM
Type here to search
81°F
11:05
30/01/2023

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import AgglomerativeClustering

x=[4,5,10,4,3,11,14,6,10,12]
y=[21,19,24,17,16,25,24,22,21,21]

data=list(zip(x,y))

heirarchical_cluster=AgglomerativeClustering(n_clusters=2,affinity='euclidean',linkage='ward')

labels=heirarchical_cluster.fit_predict(data)
plt.scatter(x,y, c=labels)

plt.show()
```



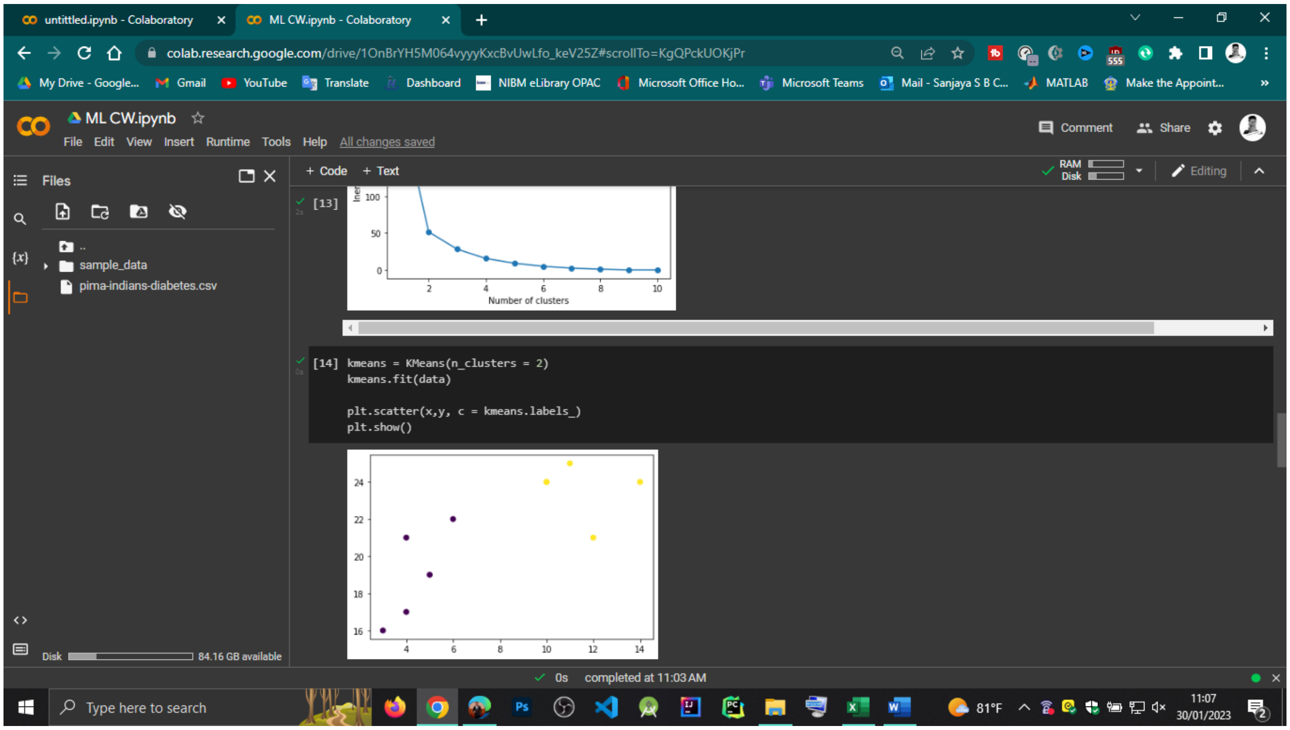
untittled.ipynb - Colaboratory
ML CW.ipynb - Colaboratory
colab.research.google.com/drive/1OnBrYH5M064vyyyKxcBvUwLfo_keV25Z#scrollTo=KgQPckUOKjPr
ML CW.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Editing
Files
sample_data
pima-indians-diabetes.csv
Disk 84.16 GB available
[11]
18
16
4
6
8
10
12
14
24
22
20
18
16
4
6
8
10
12
14
0s completed at 11:03 AM
Type here to search
81°F
11:05
30/01/2023

```
[12] import matplotlib.pyplot as plt

     x = [4,5,10,4,3,11,14,6,10,12]
     y = [21,19,24,17,16,25,24,22,24,21]

     plt.scatter(x,y)
     plt.show()
```

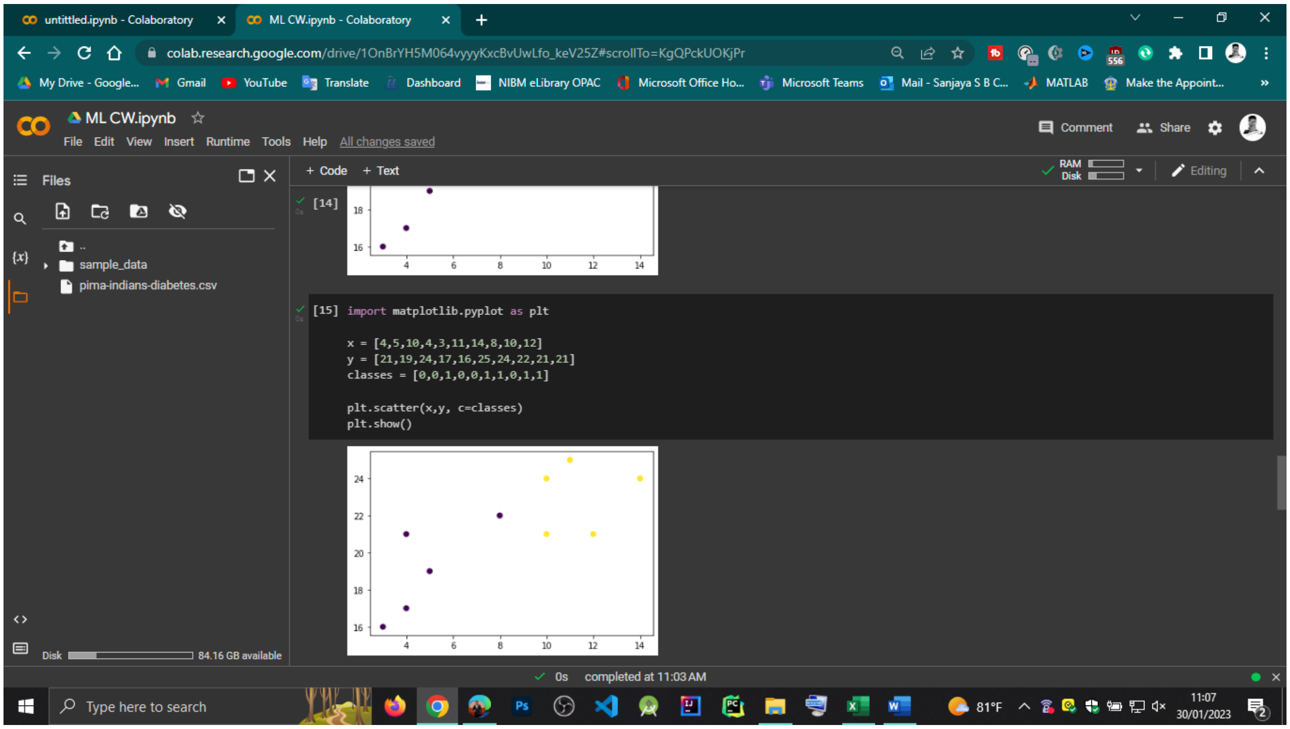


```
[13]
from sklearn.cluster import KMeans
data = list(zip(x, y))
inertias = []
for i in range(1,11):
   kmeans = KMeans (n_clusters=i)
   kmeans.fit (data)
   inertias.append(kmeans.inertia_)
plt.plot (range(1, 11), inertias, marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```

```
<ipython-input-13-3541a499ede9>:6: ConvergenceWarning: Number of distinct clusters (9) found smaller than n_clusters (10). Possibly due to dup:
  kmeans.fit (data)
```

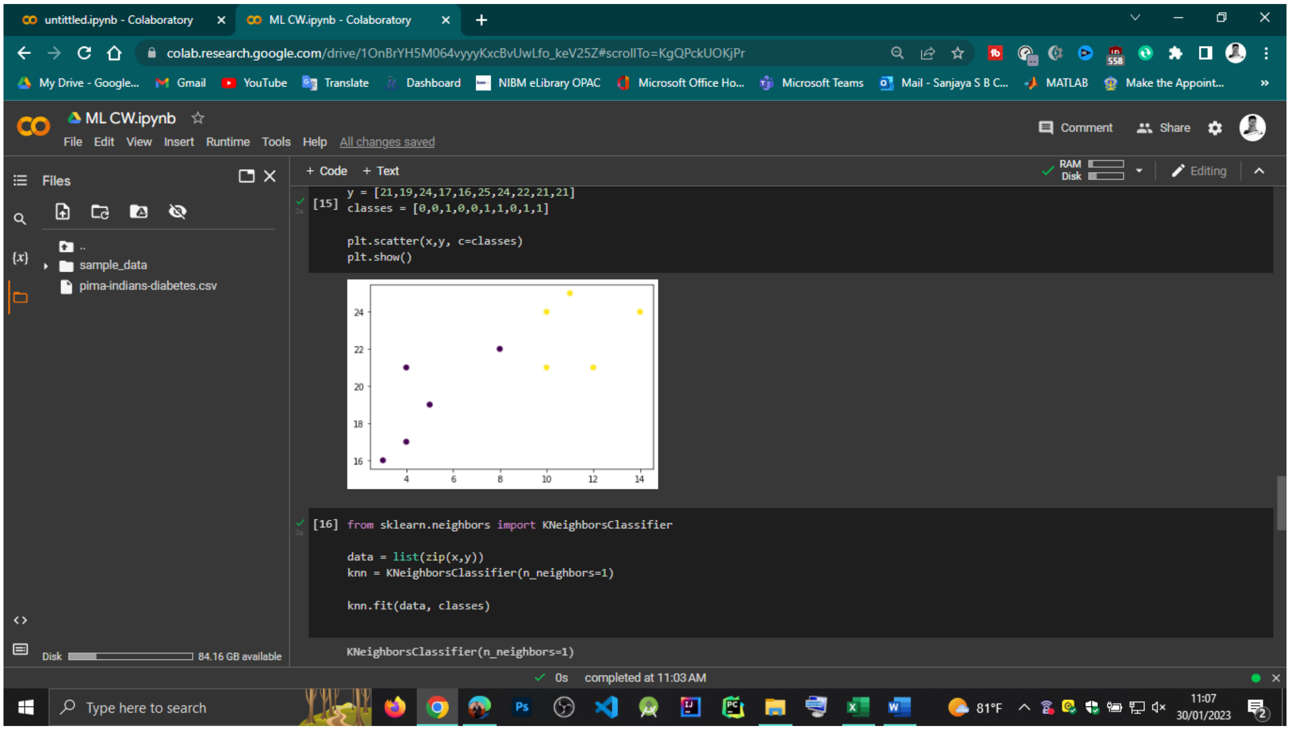


```
[14]
kmeans = KMeans(n_clusters = 2)
kmeans.fit(data)

plt.scatter(x,y, c = kmeans.labels_)
plt.show()
```


untittled.ipynb - Colaboratory
ML CW.ipynb - Colaboratory
colab.research.google.com/drive/1OnBrYH5M064vyyyKxcBvUwLfo_keV25Z#scrollTo=KgQPckUOKjPr
ML CW.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
sample_data
pima-indians-diabetes.csv
Disk 84.16 GB available
+ Code + Text
RAM Disk
Editing
18
16
4 6 8 10 12 14
24
22
20
18
16
4 6 8 10 12 14
0s completed at 11:03 AM
Type here to search
81°F
11:07
30/01/2023

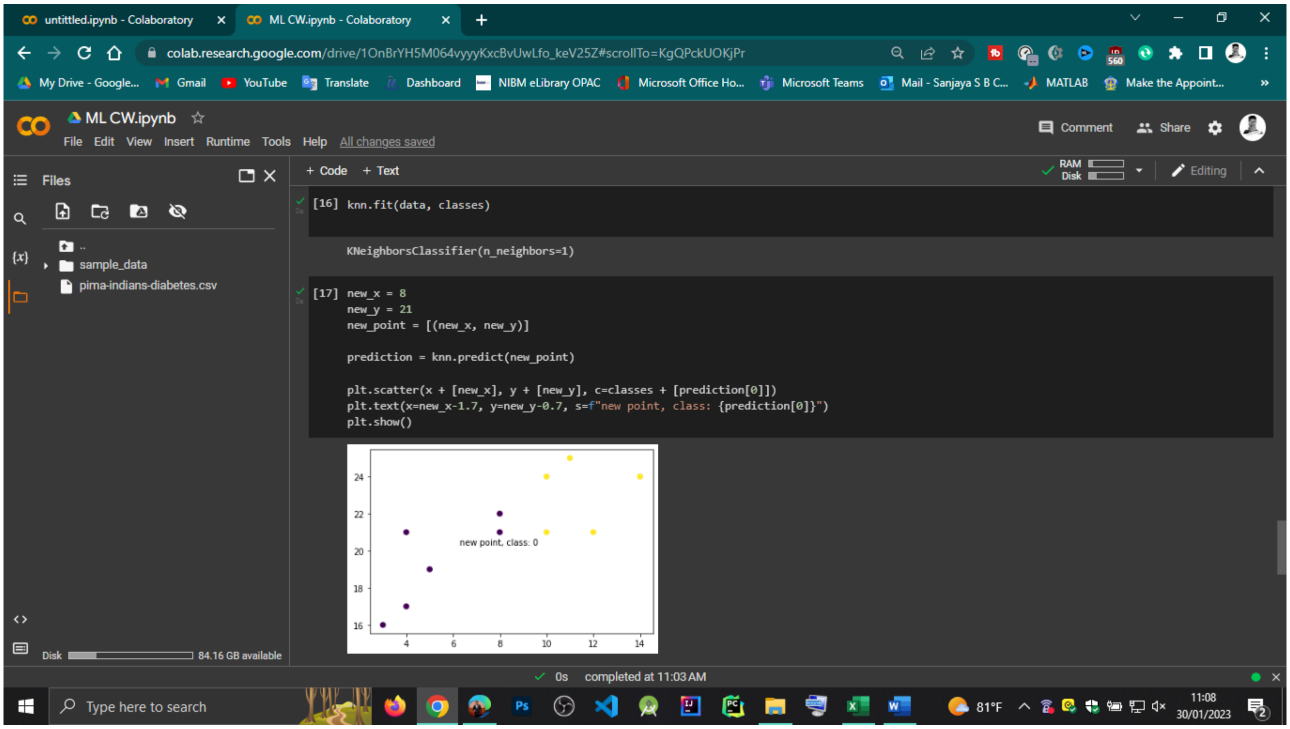
```
[14]
```

```
[15] import matplotlib.pyplot as plt

     x = [4,5,10,4,3,11,14,8,10,12]
     y = [21,19,24,17,16,25,24,22,21,21]
     classes = [0,0,1,0,0,1,1,0,1,1]

     plt.scatter(x,y, c=classes)
     plt.show()
```

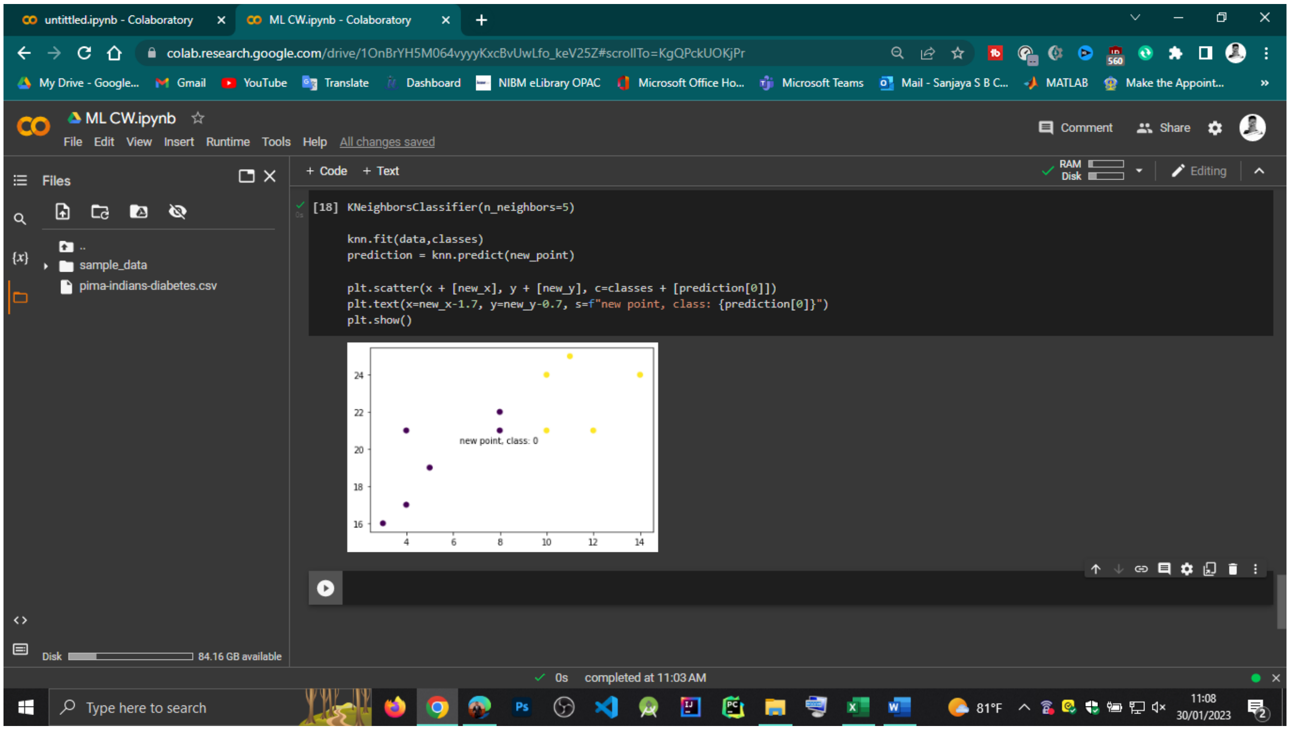

untittled.ipynb - Colaboratory
ML CW.ipynb - Colaboratory
colab.research.google.com/drive/1OnBrYH5M064vyyyKxcBvUwLfo_keV25Z#scrollTo=KgQPckUOKjPr
ML CW.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
sample_data
pima-indians-diabetes.csv
Disk 84.16 GB available
+ Code + Text
RAM Disk
Editing
24
22
20
18
16
4 6 8 10 12 14
0s completed at 11:03 AM
Type here to search
81°F
11:07
30/01/2023

```
     y = [21,19,24,17,16,25,24,22,21,21]
[15] classes = [0,0,1,0,0,1,1,0,1,1]

     plt.scatter(x,y, c=classes)
     plt.show()
```

```
[16] from sklearn.neighbors import KNeighborsClassifier

     data = list(zip(x,y))
     knn = KNeighborsClassifier(n_neighbors=1)

     knn.fit(data, classes)

     KNeighborsClassifier(n_neighbors=1)
```



```
[16] knn.fit(data, classes)

     KNeighborsClassifier(n_neighbors=1)

[17] new_x = 8
     new_y = 21
     new_point = [(new_x, new_y)]

     prediction = knn.predict(new_point)

     plt.scatter(x + [new_x], y + [new_y], c=classes + [prediction[0]])
     plt.text(x=new_x-1.7, y=new_y-0.7, s=f"new point, class: {prediction[0]}")
     plt.show()
```



```
[18] KNeighborsClassifier(n_neighbors=5)

     knn.fit(data,classes)
     prediction = knn.predict(new_point)

     plt.scatter(x + [new_x], y + [new_y], c=classes + [prediction[0]])
     plt.text(x=new_x-1.7, y=new_y-0.7, s=f"new point, class: {prediction[0]}")
     plt.show()
```

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import AgglomerativeClustering

x=[4,5,10,4,3,11,14,6,10,12]
y=[21,19,24,17,16,25,24,22,21,21]

data=list(zip(x,y))

heirarchical_cluster=AgglomerativeClustering(n_clusters=2,affinity='euclidea
n',linkage='ward')

labels=heirarchical_cluster.fit_predict(data)
plt.scatter(x,y, c=labels)

plt.show()
```

```
import matplotlib.pyplot as plt

x = [4,5,10,4,3,11,14,6,10,12]
y = [21,19,24,17,16,25,24,22,24,21]

plt.scatter(x,y)
plt.show()
```

```
from sklearn.cluster import KMeans
data = list(zip(x, y))
inertias = []
for i in range(1,11):
   kmeans = KMeans (n_clusters=i)
   kmeans.fit (data)
   inertias.append(kmeans.inertia_)
plt.plot (range(1, 11), inertias, marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```

```
kmeans = KMeans(n_clusters = 2)
kmeans.fit(data)

plt.scatter(x,y, c = kmeans.labels_)
plt.show()
```

```
import matplotlib.pyplot as plt

x = [4,5,10,4,3,11,14,8,10,12]
y = [21,19,24,17,16,25,24,22,21,21]
classes = [0,0,1,0,0,1,1,0,1,1]

plt.scatter(x,y, c=classes)
plt.show()
```

```
from sklearn.neighbors import KNeighborsClassifier

data = list(zip(x,y))
knn = KNeighborsClassifier(n_neighbors=1)

knn.fit(data, classes)
```

```
new_x = 8
new_y = 21
new_point = [(new_x, new_y)]

prediction = knn.predict(new_point)

plt.scatter(x + [new_x], y + [new_y], c=classes + [prediction[0]])
plt.text(x=new_x-1.7, y=new_y-0.7, s=f"new point, class: {prediction[0]}")
plt.show()
```

```
KNeighborsClassifier(n_neighbors=5)

knn.fit(data,classes)
prediction = knn.predict(new_point)

plt.scatter(x + [new_x], y + [new_y], c=classes + [prediction[0]])
plt.text(x=new_x-1.7, y=new_y-0.7, s=f"new point, class: {prediction[0]}")
plt.show()
```

Source Code

https://colab.research.google.com/drive/1OnBrYH5M064vyyyKxcBvUwLfo_keV25Z?usp=sharing

Dataset File

https://drive.google.com/file/d/1GyTkel27AyonSxZHriUZyu05t9cbiJ8k/view?usp=share_link

Conclusion

The conclusion is, I train some machine learning models and get output by finding the Accuracy value of the Prima-Indians-Diabestes.csv dataset file.

References

Kaggle (2022). *Kaggle: Your Home for Data Science*. [online] Kaggle.com. Available at: https://www.kaggle.com/.

Google (2019). *Google Colaboratory*. [online] Google.com. Available at: https://colab.research.google.com/.