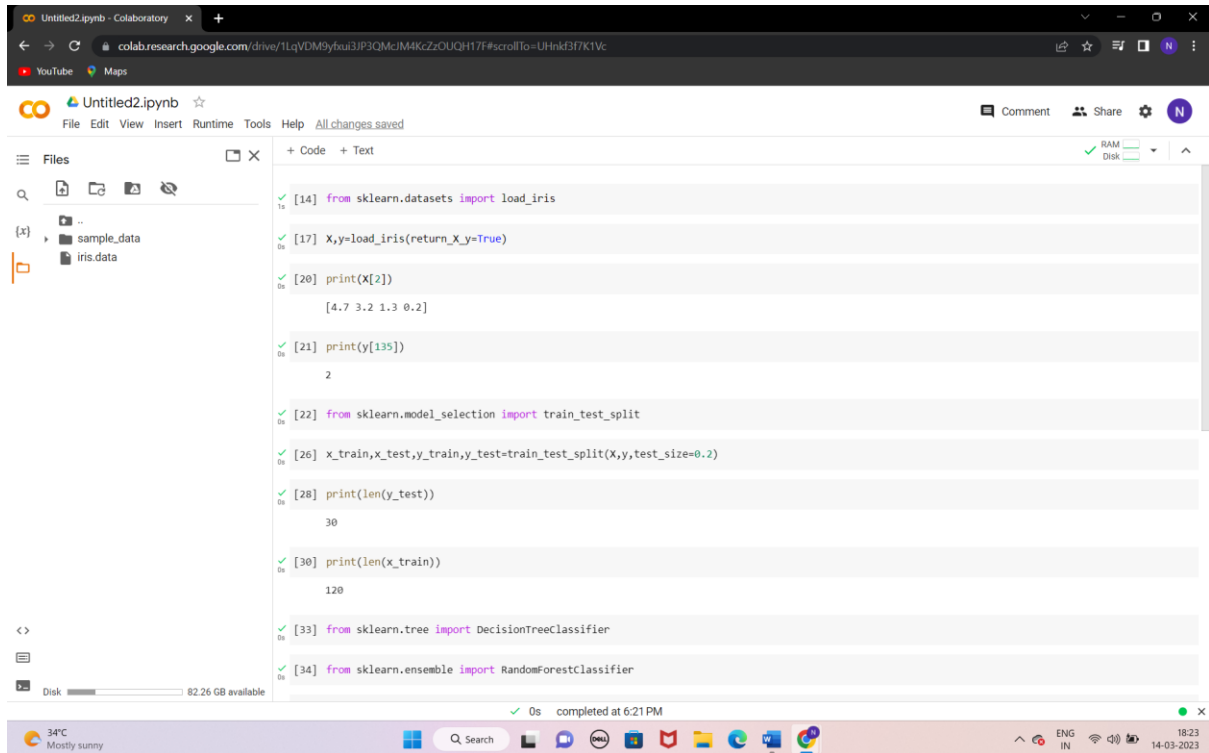


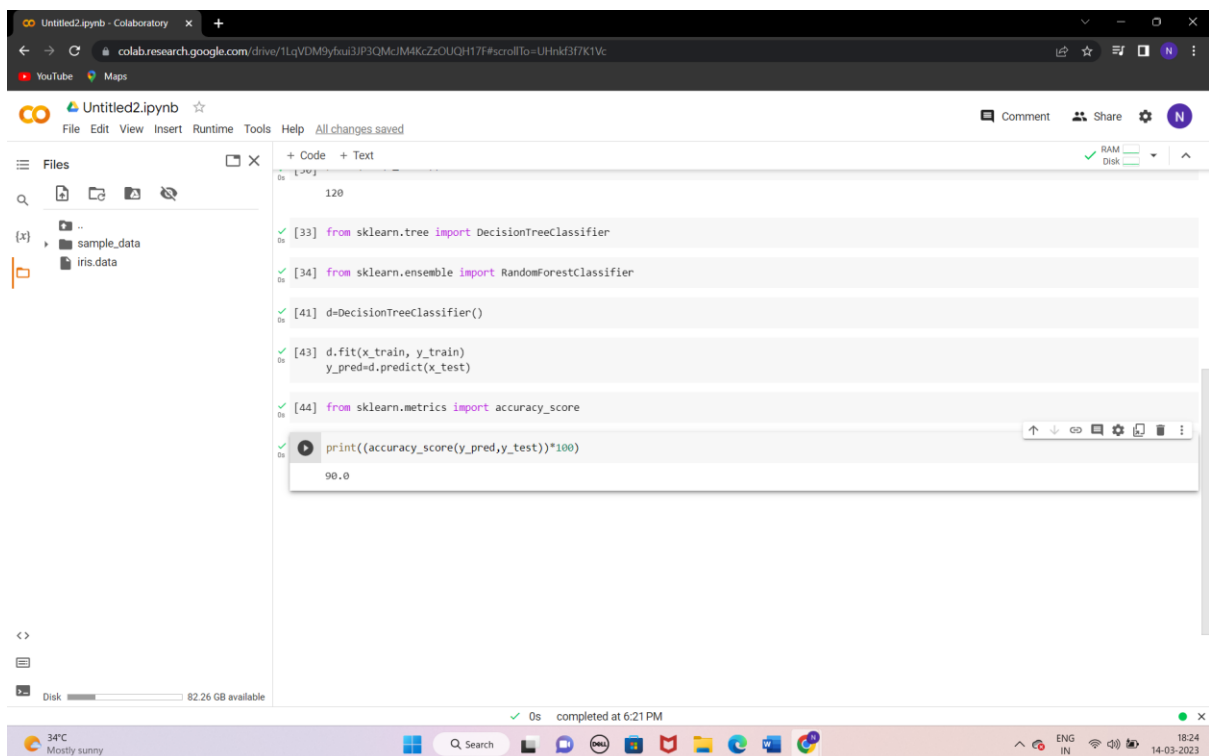
# IRIS FLOWER CLASSIFIER

1.



```
[14] from sklearn.datasets import load_iris
[17] X,y=load_iris(return_X_y=True)
[20] print(X[2])
      [4.7 3.2 1.3 0.2]
[21] print(y[135])
      2
[22] from sklearn.model_selection import train_test_split
[26] x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
[28] print(len(y_test))
      30
[30] print(len(x_train))
      120
[33] from sklearn.tree import DecisionTreeClassifier
[34] from sklearn.ensemble import RandomForestClassifier
```

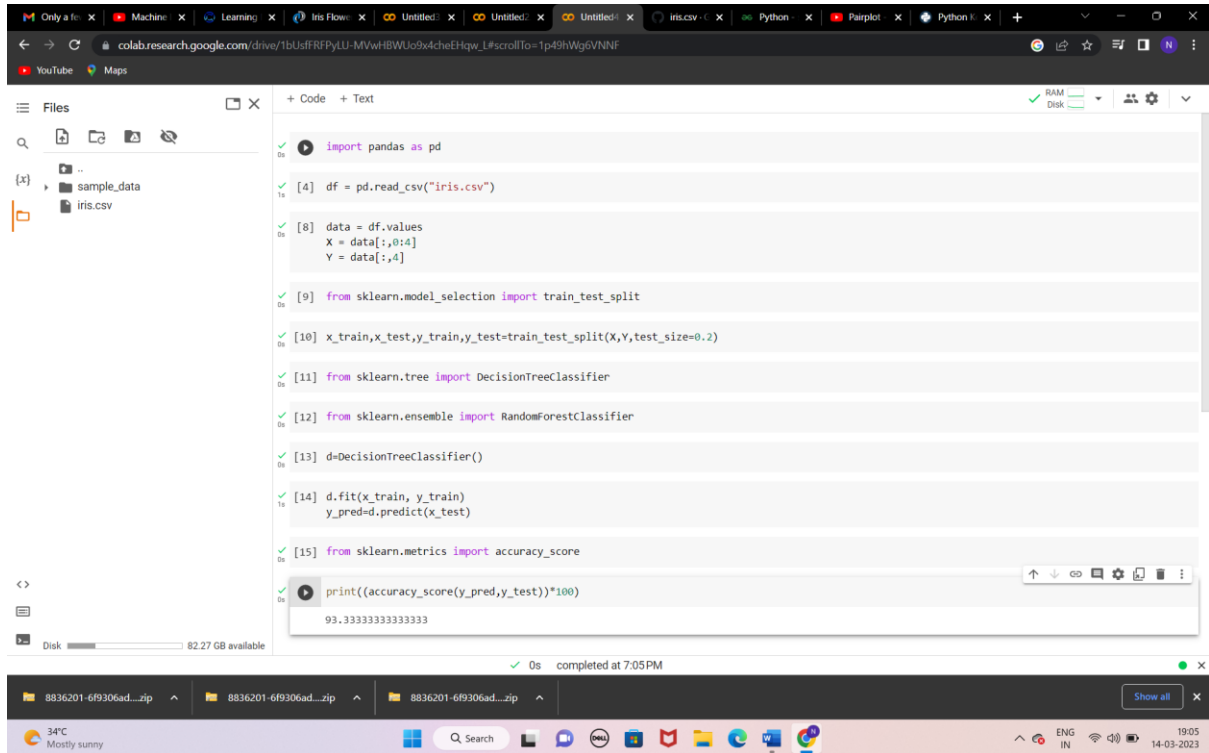
RAM 100% Disk 100% completed at 6:21 PM



```
[33] from sklearn.tree import DecisionTreeClassifier
[34] from sklearn.ensemble import RandomForestClassifier
[41] d=DecisionTreeClassifier()
[43] d.fit(x_train, y_train)
      y_pred=d.predict(x_test)
[44] from sklearn.metrics import accuracy_score
[45] print((accuracy_score(y_pred,y_test))*100)
      90.0
```

RAM 100% Disk 100% completed at 6:21 PM

2.

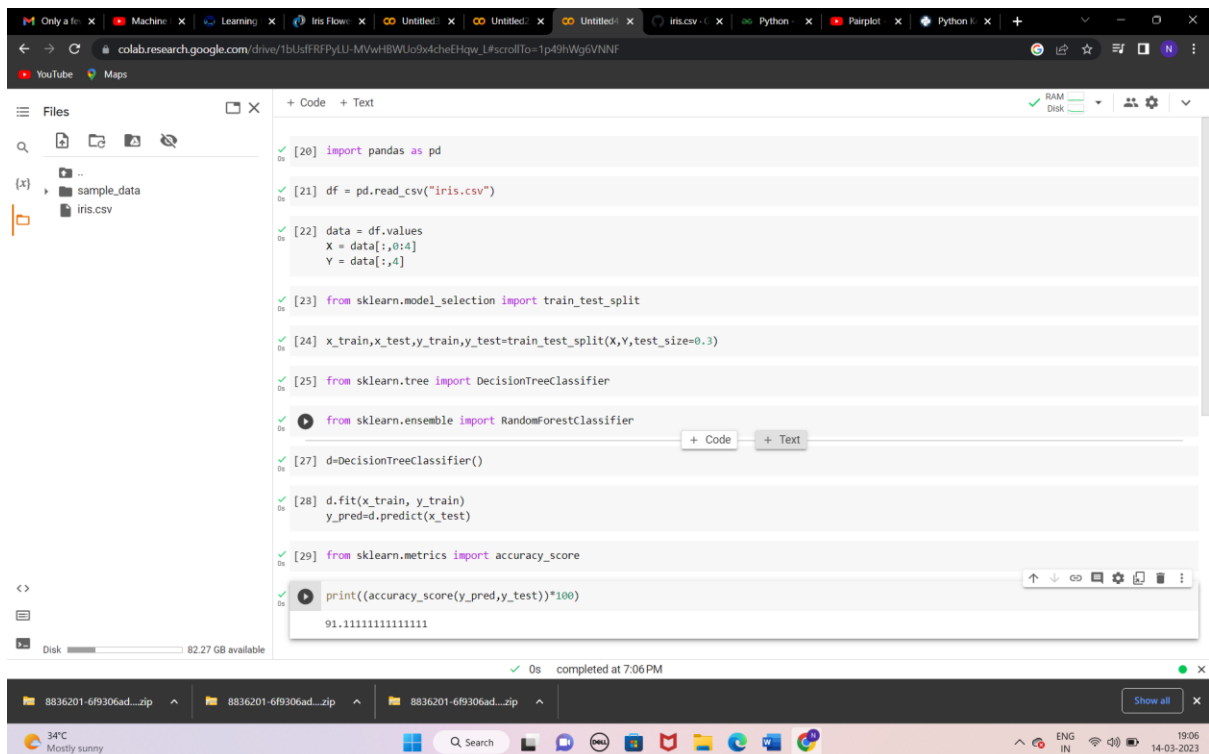


The screenshot displays a Google Colab notebook with the following code and output:

```
import pandas as pd
df = pd.read_csv("iris.csv")
data = df.values
X = data[:,0:4]
Y = data[:,4]
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2)
from sklearn.tree import DecisionTreeClassifier
d=DecisionTreeClassifier()
d.fit(x_train, y_train)
y_pred=d.predict(x_test)
from sklearn.metrics import accuracy_score
print((accuracy_score(y_pred,y_test))*100)
```

The output of the notebook is 93.33333333333333.

3.



The screenshot displays a Google Colab notebook with the following code and output:

```
import pandas as pd
df = pd.read_csv("iris.csv")
data = df.values
X = data[:,0:4]
Y = data[:,4]
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.3)
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
d=DecisionTreeClassifier()
d.fit(x_train, y_train)
y_pred=d.predict(x_test)
from sklearn.metrics import accuracy_score
print((accuracy_score(y_pred,y_test))*100)
```

The output of the notebook is 91.11111111111111.

colab.research.google.com/drive/1bUsFRFPyLU-MVwHBWUo9x4cheEHqW\_L#scrollTo=1p49hWg6VNNF

Files

- sample\_data
- iris.csv

```
[34] import pandas as pd
[35] df = pd.read_csv("iris.csv")
[36] data = df.values
      X = data[:,0:4]
      Y = data[:,4]
[37] from sklearn.model_selection import train_test_split
[38] x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.25)
[39] from sklearn.tree import DecisionTreeClassifier
[40] from sklearn.ensemble import RandomForestClassifier
[41] d=DecisionTreeClassifier()
[42] d.fit(x_train, y_train)
      y_pred=d.predict(x_test)
[43] from sklearn.metrics import accuracy_score
[44] print((accuracy_score(y_pred,y_test))*100)
      92.10526315789474
```

completed at 7:08 PM

8836201-6f9306ad...zip

34°C Mostly sunny

Search

ENG IN 19:08 14-03-2023

colab.research.google.com/drive/1bUsFRFPyLU-MVwHBWUo9x4cheEHqW\_L#scrollTo=1p49hWg6VNNF

Files

- sample\_data
- iris.csv

```
[34] import pandas as pd
[45] df = pd.read_csv("iris.csv")
[46] data = df.values
      X = data[:,0:4]
      Y = data[:,4]
[47] from sklearn.model_selection import train_test_split
[48] x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.1)
[49] from sklearn.tree import DecisionTreeClassifier
[50] from sklearn.ensemble import RandomForestClassifier
[51] d=DecisionTreeClassifier()
[52] d.fit(x_train, y_train)
      y_pred=d.predict(x_test)
[53] from sklearn.metrics import accuracy_score
[54] print((accuracy_score(y_pred,y_test))*100)
      100.0
```

completed at 7:08 PM

8836201-6f9306ad...zip

34°C Mostly sunny

Search

ENG IN 19:08 14-03-2023

4.

The screenshot shows a Google Colab notebook titled 'Untitled3.ipynb'. The left sidebar displays the file explorer with a folder named 'sample\_data' containing a file 'iris.csv'. The main code area contains the following code cells:

```
[2] import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

[16] df = pd.read_csv("iris.csv")

[17] df.describe()
```

The output of the `df.describe()` cell is a summary statistics table:

	sepal.length	sepal.width	petal.length	petal.width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Below the table, the following code cell is visible:

```
[18] # Separate features and target
data = df.values
X = data[:,0:4]
Y = data[:,4]
```

The notebook status bar at the bottom indicates 'completed at 6:55 PM'.

The screenshot shows the same Google Colab notebook with additional code cells for data splitting, model training, and prediction:

```
[18] # Separate features and target
data = df.values
X = data[:,0:4]
Y = data[:,4]

[20] # Split the data to train and test dataset.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)

# Support vector machine algorithm
from sklearn.svm import SVC
svm = SVC()
svm.fit(X_train, y_train)

# Predict from the test dataset
predictions = svm.predict(X_test)

# Calculate the accuracy
from sklearn.metrics import accuracy_score
accuracy_score(y_test, predictions)

0.9666666666666667
```

The notebook status bar at the bottom indicates 'completed at 6:55 PM'.

## IMAGE RECOGNITION USING HAAR CASCADE .

```
+ Code + Text
import cv2
from google.colab.patches import cv2_imshow

face_cascade=cv2.CascadeClassifier(cv2.data.harcascades + "haarcascade_frontalface_default.xml")

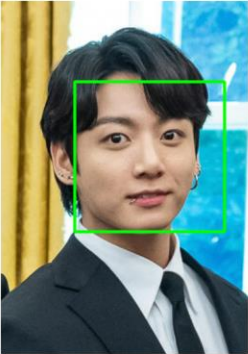
img = cv2.imread('image.jpg')

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)

cv2_imshow(img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



0s completed at 9:16 PM

## BOSTON HOUSING CLASSIFIER

```
+ Code + Text
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

# Load the Boston Housing dataset
boston = pd.read_csv('Boston.csv')

# Separate the features (X) and target (y) variables
X = boston.drop(columns=['medv'])
y = boston['medv']

#d.fit_transform(X_train, y_train)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Train the Decision Tree Regressor on the training data
d = DecisionTreeRegressor()
d.fit(X_train, y_train)

# Use the trained model to make predictions on the testing data
y_pred = d.predict(X_test)

# Calculate the mean squared error of the model's predictions
score = mean_squared_error(y_test, y_pred)
score
```

13.76

0s completed at 10:44 PM