

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

iris_data = pd.read_csv("/content/Iris.csv")

print(iris_data.columns)

Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
      'Species'],
      dtype='object')

# Split the dataset into features and target variable
x = iris_data.drop('Species', axis=1)
y = iris_data['Species']

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=2)

from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
import numpy as np

# Assuming x_train, x_test, y_train, y_test are your training and test sets
# Create and train KNN classifier
knn = KNeighborsClassifier()
knn.fit(x_train, y_train)

# Make predictions on the test set
y_pred = knn.predict(x_test)
# Calculate and print KNN accuracy
print('KNN accuracy:{:.2f}%'.format(knn.score(x_test,y_test)*100))

KNN accuracy:100.00%

from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
import numpy as np

# Assuming x_train, x_test, y_train, y_test are your training and test sets
# Create and train SVM classifier
svm = SVC()
svm.fit(x_train, y_train)

# Make predictions on the test set
y_pred = svm.predict(x_test)
# Calculate and print SVM accuracy
print('SVM accuracy:{:.2f}%'.format(svm.score(x_test,y_test)*100))

SVM accuracy:96.67%

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
import numpy as np

# Assuming x_train, x_test, y_train, y_test are your training and test sets
# Create and train Decision Tree classifier
dtc = DecisionTreeClassifier()
dtc.fit(x_train, y_train)

# Make predictions on the test set
y_pred = dtc.predict(x_test)
# Calculate and print Decision Tree accuracy

```

```
print('Decision Tree accuracy:{:.2f}%'.format(dtc.score(x_test,y_test)*100))
```

```
Decision Tree accuracy:100.00%
```