**from** sklearn.datasets **import** load\_iris

iris\_dataset**=**load\_iris()

print("keys of iris\_dataset:\n{}"**.**format(iris\_dataset**.**keys()))

print(iris\_dataset['DESCR'][:193] **+** "\n...")

print("Target names: {}"**.**format(iris\_dataset['target\_names']))

print("Feature names: \n{}"**.**format(iris\_dataset['feature\_names']))

print("Type of data: {}"**.**format(type(iris\_dataset['data'])))

print("Shape of data: {}"**.**format(iris\_dataset['data']**.**shape))

print("First five rows of data:\n{}"**.**format(iris\_dataset['data'][:5]))

print("Type of target: {}"**.**format(type(iris\_dataset['target'])))

print("Shape of target: {}"**.**format(iris\_dataset['target']**.**shape))

print("Target:\n{}"**.**format(iris\_dataset['target']))

**from** sklearn.model\_selection **import** train\_test\_split

x\_train,x\_test,y\_train,y\_test**=**train\_test\_split(iris\_dataset['data'],iris\_dataset['target'],random\_state**=**0)

print("x\_train shape: {}"**.**format(x\_train**.**shape))

print("y\_train shape: {}"**.**format(y\_train**.**shape))

print("x\_test shape: {}"**.**format(x\_test**.**shape))

print("y\_test shape: {}"**.**format(y\_test**.**shape))

**%pip** install mglearn

**import** pandas **as** pd

**import** mglearn

iris\_dataframe**=**pd**.**DataFrame(x\_train,columns**=**iris\_dataset**.**feature\_names)

pd**.**plotting**.**scatter\_matrix(iris\_dataframe,c**=**y\_train,figsize**=**(15,15),alpha**=**.8,cmap**=**mglearn**.**cm3)

**from** sklearn.neighbors **import** KNeighborsClassifier

knn**=**KNeighborsClassifier(n\_neighbors**=**1)

knn**.**fit(x\_train,y\_train)

**import** numpy **as** np

x\_new**=**np**.**array([[5,2.9,1,0.2]])

print("x\_new.shape: {}"**.**format(x\_new**.**shape))

prediction**=**knn**.**predict(x\_new)

print("Prediction: {}"**.**format(prediction))

print("predicted target name: {}"**.**format(iris\_dataset['target\_names'][prediction]))

y\_pred**=**knn**.**predict(x\_test)

print("Test set predictions:\n {}"**.**format(y\_pred))

print("Test score: {:.2f}"**.**format(np**.**mean(y\_pred **==** y\_test)))

print("Test set score: {:.2f}"**.**format(knn**.**score(x\_test,y\_test)))

🌺 Iris Flowers Classification Using Machine Learning 🌺  
  
Excited to share my recent project where I applied machine learning techniques to classify Iris flowers! 🌸 Utilizing a dataset with features like sepal length, sepal width, petal length, and petal width, I implemented a robust model that accurately distinguishes between three different Iris species: Setosa, Versicolor, and Virginica.  
  
🔍 Technologies Used: Python, Scikit-learn, Pandas, Matplotlib & Seaborn  
  
Achieved impressive accuracy in classifying Iris flowers, showcasing the potential of machine learning in solving real-world classification problems. This project not only enhances my skills in machine learning but also demonstrates my ability to translate data into actionable insights.  
<https://lnkd.in/du3VDV3U> 👈 this a link to the GitHub repository where you can access the code, dataset, and detailed documentation.  
[Bharat Intern](https://www.linkedin.com/company/bharat-intern/)  
[#machinelearning](https://www.linkedin.com/feed/hashtag/?keywords=machinelearning&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7117180102869684225)