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

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

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

df **=** pd**.**read\_csv('iris.csv')

print("-----------Dataframe Info------------")

print(df**.**info())

print("\n")

print("-----------Dataframe Describe------------")

print(df**.**describe())

print("\n")

print("-----------Dataframe Head------------")

print(df**.**head())

print("\n")

print("----------------Data Preprocessing----------------")

X **=** df**.**iloc[:,0:4]

Y **=** df['Species']**.**values

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

**from** sklearn.preprocessing **import** StandardScaler

X\_train, X\_test, Y\_train, Y\_test **=** train\_test\_split(X, Y, test\_size**=**0.2, random\_state**=**0)

sc\_X **=** StandardScaler()

X\_train **=** sc\_X**.**fit\_transform(X\_train)

X\_test **=** sc\_X**.**transform(X\_test)

print(f'Train Dataset Size - X: {X\_train**.**shape}, Y: {Y\_train**.**shape}')

print(f'Test Dataset Size - X: {X\_test**.**shape}, Y: {Y\_test**.**shape}')

print("\n")

print("---------------Naive Bayes Classifier----------------------")

*# This code fits a Naive Bayes classifier model on the training data, makes predictions on the test data,*

*# maps the predicted species labels to integers, and plots regression lines comparing the predicted labels*

*# to the actual labels for each of the 4 feature columns. It shows how the Naive Bayes model performs at*

*# predicting the species from each individual feature.*

**from** sklearn.naive\_bayes **import** GaussianNB

classifier **=** GaussianNB()

classifier**.**fit(X\_train, Y\_train)

predictions **=** classifier**.**predict(X\_test)

mapper **=** {'setosa': 0, 'versicolor': 1, 'virginica': 2}

predictions\_ **=** [mapper[i] **for** i **in** predictions]

fig, axs **=** plt**.**subplots(2, 2, figsize **=** (12, 10), constrained\_layout **=** **True**);

\_ **=** fig**.**suptitle('Regression Line Tracing')

**for** i **in** range(4):

x, y **=** i **//** 2, i **%** 2

\_ **=** sns**.**regplot(x **=** X\_test[:, i], y **=** predictions\_, ax**=**axs[x, y])

\_ **=** axs[x, y]**.**scatter(X\_test[:, i][::**-**1], Y\_test[::**-**1], marker **=** '+', color**=**"white")

\_ **=** axs[x, y]**.**set\_xlabel(df**.**columns[i **+** 1][:**-**2])

plt**.**show()

print("\n")

print("------------Confusion Matrix-------------")

**from** sklearn.metrics **import** confusion\_matrix

**from** sklearn.metrics **import** classification\_report

cm **=** confusion\_matrix(Y\_test, predictions)

print(f'''Confusion matrix :\n

| Positive Prediction\t| Negative Prediction

---------------+------------------------+----------------------

Positive Class | True Positive (TP) {cm[0, 0]}\t| False Negative (FN) {cm[0, 1]}

---------------+------------------------+----------------------

Negative Class | False Positive (FP) {cm[1, 0]}\t| True Negative (TN) {cm[1, 1]}\n\n''')

cm **=** classification\_report(Y\_test, predictions)

print('Classification report : \n', cm)