import pandas as pd

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

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import classification\_report

#using pandas importing the glass csv file

df = pd.read\_csv('glass.csv')

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(df.drop("Type", axis=1), df["Type"], test\_size=0.2)

# train model using the Naive\_bayes

model = GaussianNB()

model.fit(X\_train, Y\_train)

# evaluating the model

score = model.score(X\_test, Y\_test)

print('Accuracy: %.3f' % score)

# generating the classification report

y\_pred = model.predict(X\_test)

report = classification\_report(y\_true=Y\_test, y\_pred=y\_pred)

print(report)

A screenshot of a computer

Description automatically generated

import pandas as pd

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

from sklearn.svm import LinearSVC

from sklearn.metrics import classification\_report

# using pandas importing the dataset

df = pd.read\_csv('glass.csv')

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(df.drop("Type", axis=1), df["Type"], test\_size=0.2)

# train model using linear support vector machine

model = LinearSVC(dual=False)

model.fit(X\_train, Y\_train)

# evaluating the model

score = model.score(X\_test, Y\_test)

print('Accuracy: %.3f' % score)

# generating the classification report

y\_pred = model.predict(X\_test)

report = classification\_report(y\_true=Y\_test, y\_pred=y\_pred)

print(report)

A screenshot of a computer screen

Description automatically generated