Using the make\_blobs method from sklearn.datasets, The given dataset creates simulated data. Three clusters of data were created, each with a different center coordinate. The number of clusters depends on how many centers are listed. After that by using the 'train\_test\_split' function, I split the data into training (80%) and testing (20%). Then after splitting the data I used to apply the K-nearest neighbors (KNN) classifier to the training data. The KNN model is modified to the training data set to use the fit() function. Based on the testing data, predictions are made using the prediction method. The non-parametric classification method used in machine learning is called K-Nearest Neighbors (KNN). It is a form of instance-based learning where the model identifies new data points based on their close to the training dataset in the classifier after learning from the labeled data. After that to find accuracy the function accuracy score is used to determine the accuracy of the KNN model. By comparing the predicted labels from the training data learning data predicted with the actual labels from the training data train labels to determine the accuracy score. The accuracy score is the percentage of correctly predicted labels or the number of samples that were correctly classified as a proportion of all the samples in the training set. To get better accuracy I change the 'n\_sample' size to 420 then the accuracy score is '0.9824'. Finally, by using the matplotlib.pyplot library, the plot create three different classes. Overall, The code shows how to create random numbers, split them into train and test sets, train and test a KNN classifier, find the accuracy, and show the results using a scatter plot.