# SUMMARY REPORT OF KNN ANALYSIS ALGORITHM

## INTRODUCTION

The goal of this project is to replicate a dataset using the scikit-learn make blobs function, analyze the data using a K-nearest neighbors (KNN) analysis, and assess the model's correctness. Matplotlib can be used to visualize the results.

## DATA SIMULATION

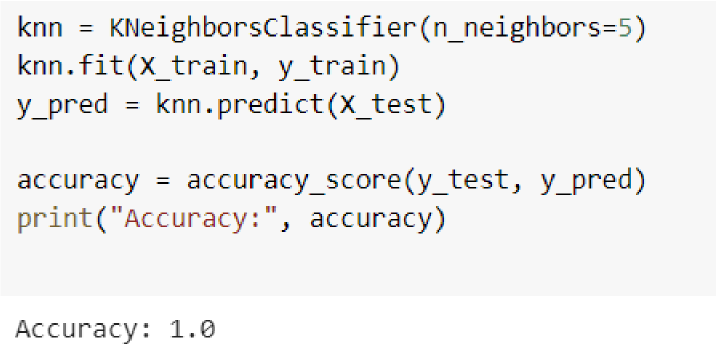
With the help of the make blob’s function, we first simulate a dataset made up of 150 samples and 3 clusters. The cluster centers are [[2, 4], [6, 6], and [1, 9]]. With the help of the label’s variable, we give each sample a label.

## DATA PREPROCESSING

We need to separate the data into training and testing sets to do a KNN analysis. Using the train test split function of scikit-learn, we divide the data by 80/20. For reproducibility, we use a test size of 0.2 and a random state of 1.

## MODEL TRAINING AND EVALUATION

To make predictions, we create a KNeighborsClassifier object with k neighbors=5 and fit it to the training set of data. The test data's labels are predicted using the fitted model, and the accuracy score is computed using both the actual labels and the predicted labels. To calculate the accuracy, we employ the scikit-learn accuracy score function.



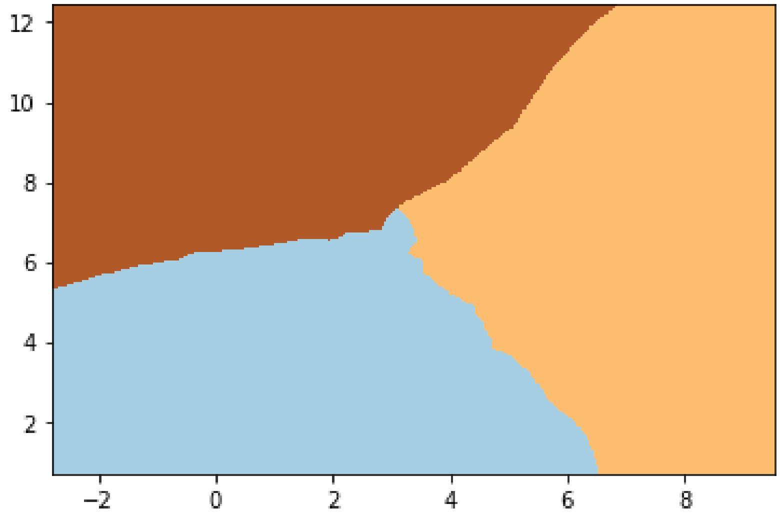
```
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

## RESULTS

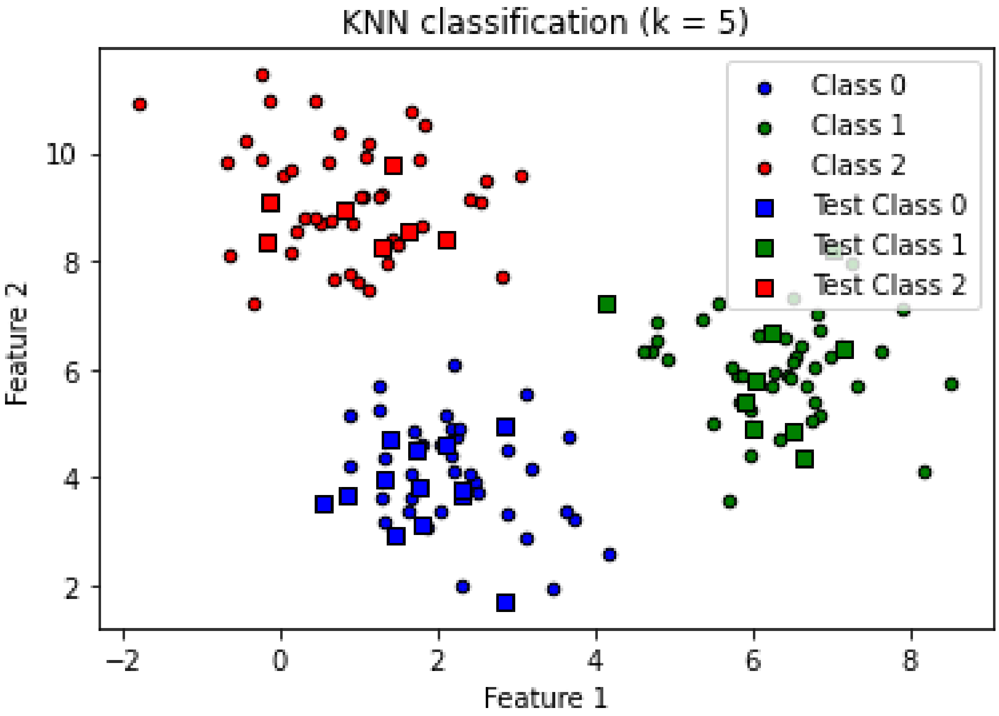
The KNN classifier classified the test data with an accuracy of 1.0 or 100%, showing that the model can do so accurately.

Plotting the data and the KNN classifier's decision limits allows us to see the outcomes. To forecast the class labels of these points, we create a mesh grid of points that spans the entire range of the data. We create a color map of the decision boundaries. Using different markers at the testing points, we also visualize the training and testing points. Using plt.show, we display the plot and add a legend ().



## CONCLUSION

In this project, we simulated a dataset containing three clusters, ran a KNN analysis (k value equals to 5) on the information, and assessed the model's precision. For the test data, we were able to reach an accuracy of 100%, showing that the model can accurately categorize the data. We used Matplotlib to visualize the results.


KNN classification (k = 5)
Class 0
Class 1
Class 2
Test Class 0
Test Class 1
Test Class 2
Feature 2
Feature 1