MACHINE LEARNING

TRAFFIC SIGN DOCUMENTATION

Agenda

1- INTRODUCTION

2- *General Information on dataset*

3- Implementation details

4- *Results details*

INTRODUCTION ABOUT PROJECT

The project focuses on the analysis of images related to road conditions, aiming to classify and cluster them into relevant categories. The primary objectives include identifying specific road conditions and implementing unsupervised learning techniques for grouping similar images.

Image Classification:

The project employs logistic regression, a supervised learning algorithm, for image classification. The images, initially stored in the 'Training' directory, are preprocessed, resized, and normalized before being used to train the logistic regression model. The trained model is then evaluated on both a split dataset and a separate test dataset ('Testing' directory), providing insights into its accuracy and predictive capabilities.

Clustering with KMeans

In addition to image classification, the project involves unsupervised learning through KMeans clustering. The 'Training' images are clustered based on their features, and the resulting clusters are analyzed and visualized. The model is further applied to cluster a set of test images, demonstrating its ability to group similar road condition images.

Evaluation and Visualization:

Performance evaluation metrics, such as accuracy scores, confusion matrices, and ROC curves, are utilized to assess the effectiveness of both the image classification and clustering processes. Visualization techniques, including heatmaps and PCA-based scatter plots, are employed to enhance the interpretation of the results.

Hyperparameter Tuning:

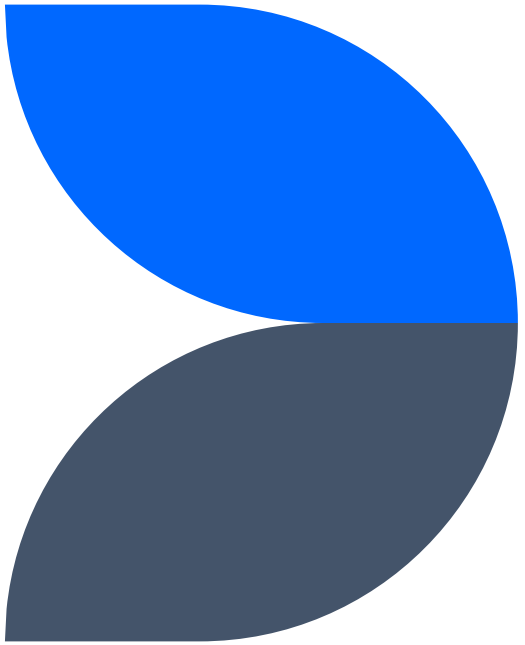
The project involves setting hyperparameters for both logistic regression and KMeans models. Hyperparameters like the regularization strength (C), maximum iterations, and the number of clusters influence the models' learning and clustering behavior.

General Information on dataset

1.1 Dataset Overview

- **Name of Dataset:** Traffic Sign
- **Number of Classes and Labels:** 3 classes - Hump,Give way, Stop
- **Total Number of Samples:** [384]
- **Sample Size (in case of images):** 30x30 pixels
- **Split of Samples:**
  - Training file from website 384
  - Testing file from website 235

2. Implementation Details



2.1 Feature Extraction Phase

- **Number of Features Extracted:** [81 by hug algorithm]

- **Feature Names and Dimension of Resulted Features:**

The features extracted using Histogram of Oriented Gradients (HOG) are essentially descriptors that capture information about the local gradients or edge directions in an image. These descriptors are represented as a one-dimensional array of values, and each value corresponds to a specific feature.

2.2

Hyperparameters

2.2- Hyperparameters

- **The default hyperparameters for logistic regression in scikit-learn include**:

I. **C:** Inverse of regularization strength; smaller values specify stronger regularization.
II. **max_iter:** Maximum number of iterations for optimization.
III. **multi_class:** Method used to handle multiple classes. In your code, 'ovr' (one-vs-rest) is used.

- the hyperparameters used in the logistic regression model:

•**C:** 1.0

•**max_iter:** 1000

•**multi_class:** 'ovr'

For KMeans, the hyperparameters include:

1. the hyperparameters include:

- **n_clusters:** Number of clusters to form.
- **n_init:** Number of times the k-means algorithm will be run with different centroid seeds.

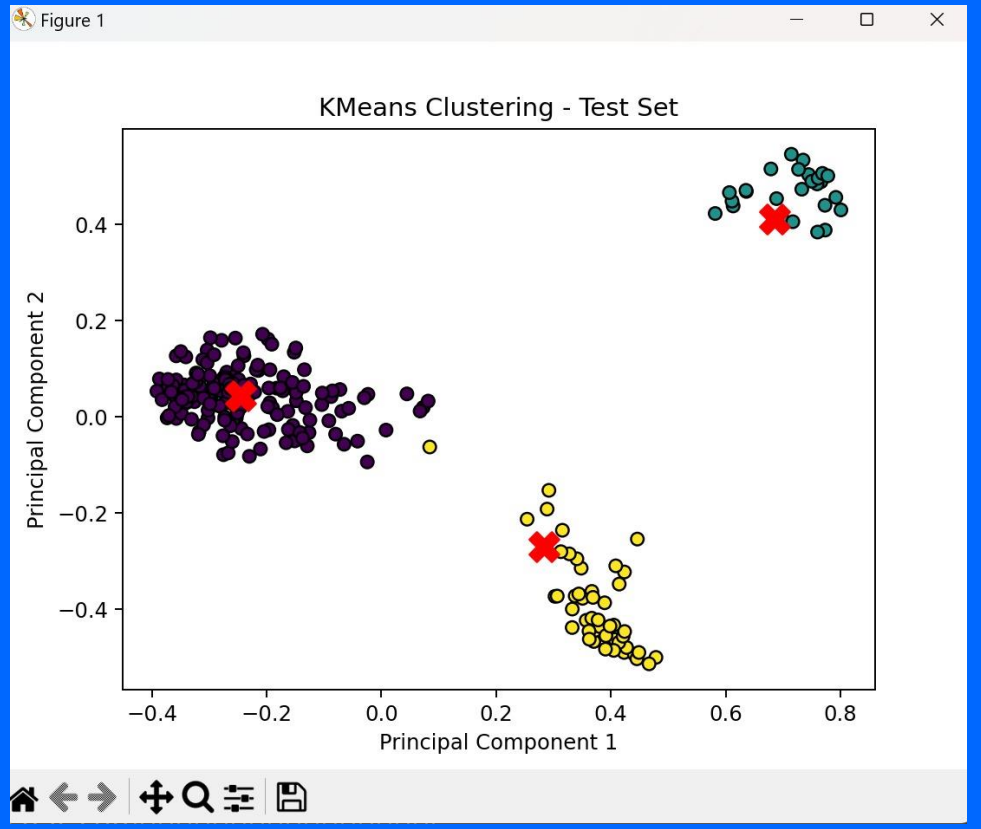
the hyperparameters used in MY KMeans model

- **n_clusters:** 3 (specified in the KMeans instantiation)
- **n_init:** 8 (specified in the KMeans instantiation)
- **random_state:** 30 (specified in the KMeans instantiation)

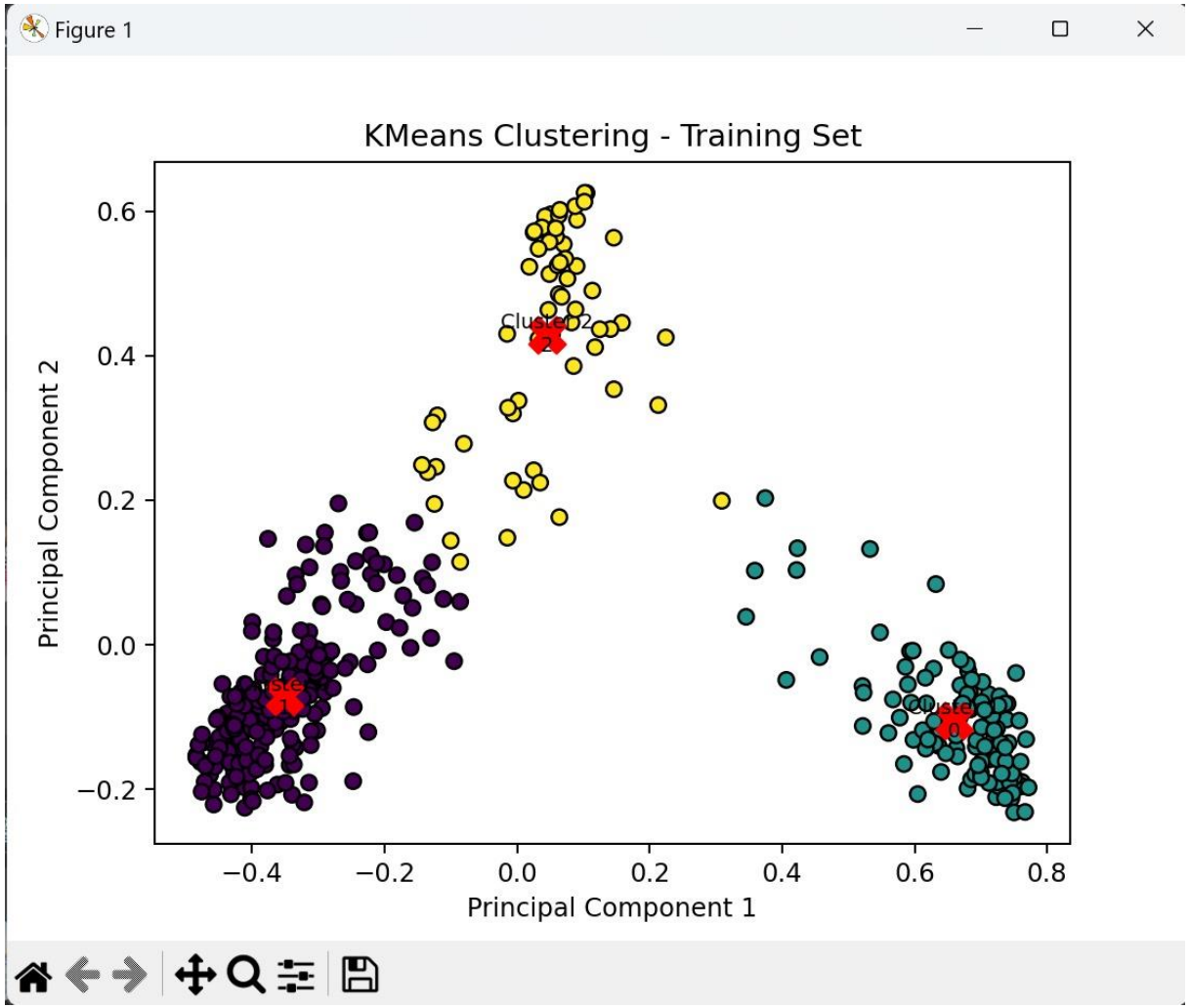
Results details

testing data for kmeans

kkmeans test set plot



kmeans trainset plot



roc for test set

confusion matrix for test set

kmeans accuracy

```
Adjusted Rand Index: 0.8679802278554617
```

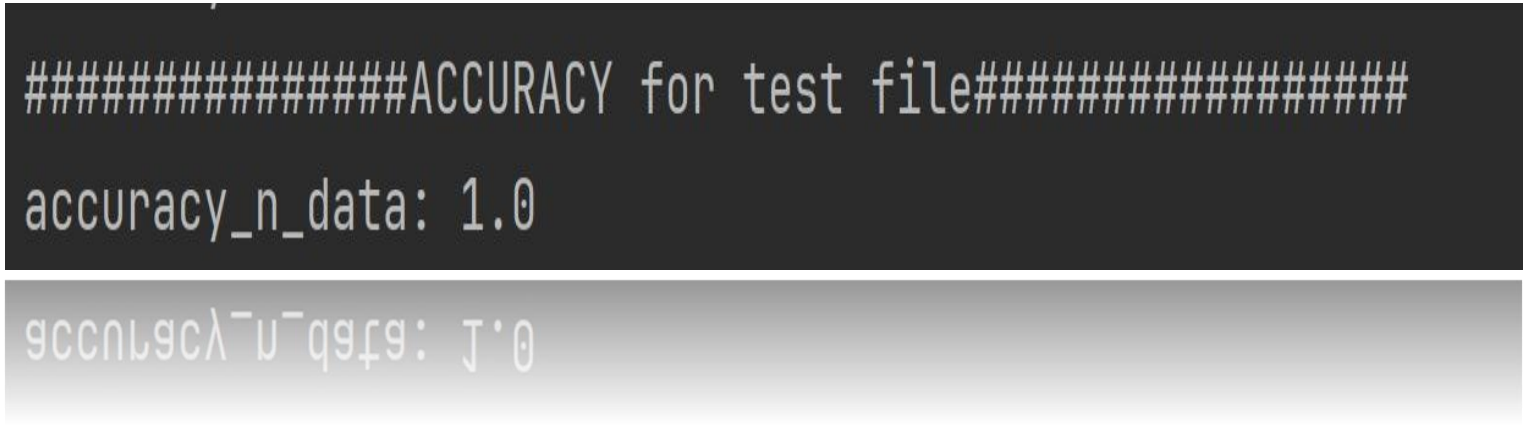
Testind data for logistic regression

split logistic samples 80 train 20 test

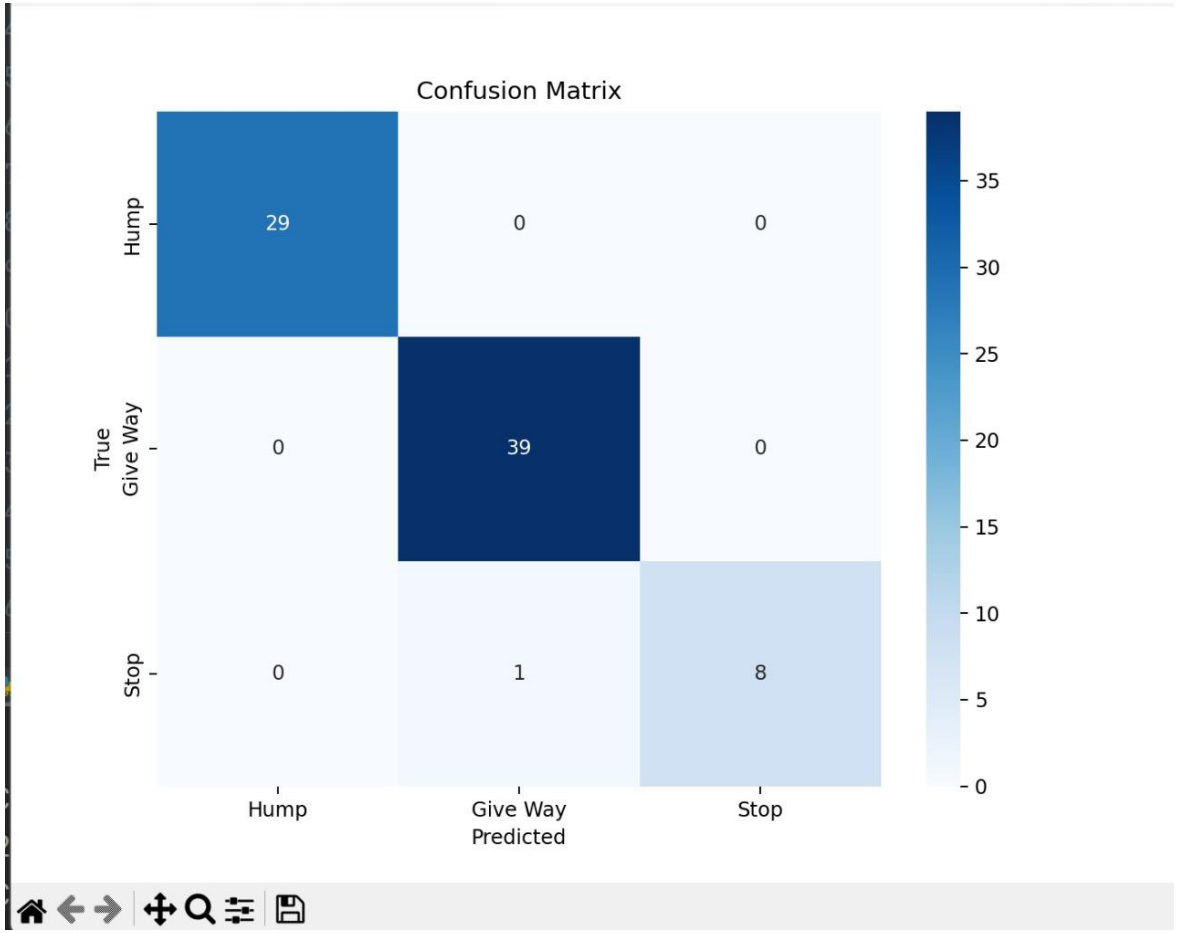
accuracy

```
##################ACCURACY for splited data##################
Accuracy: 0.987012987012987
```

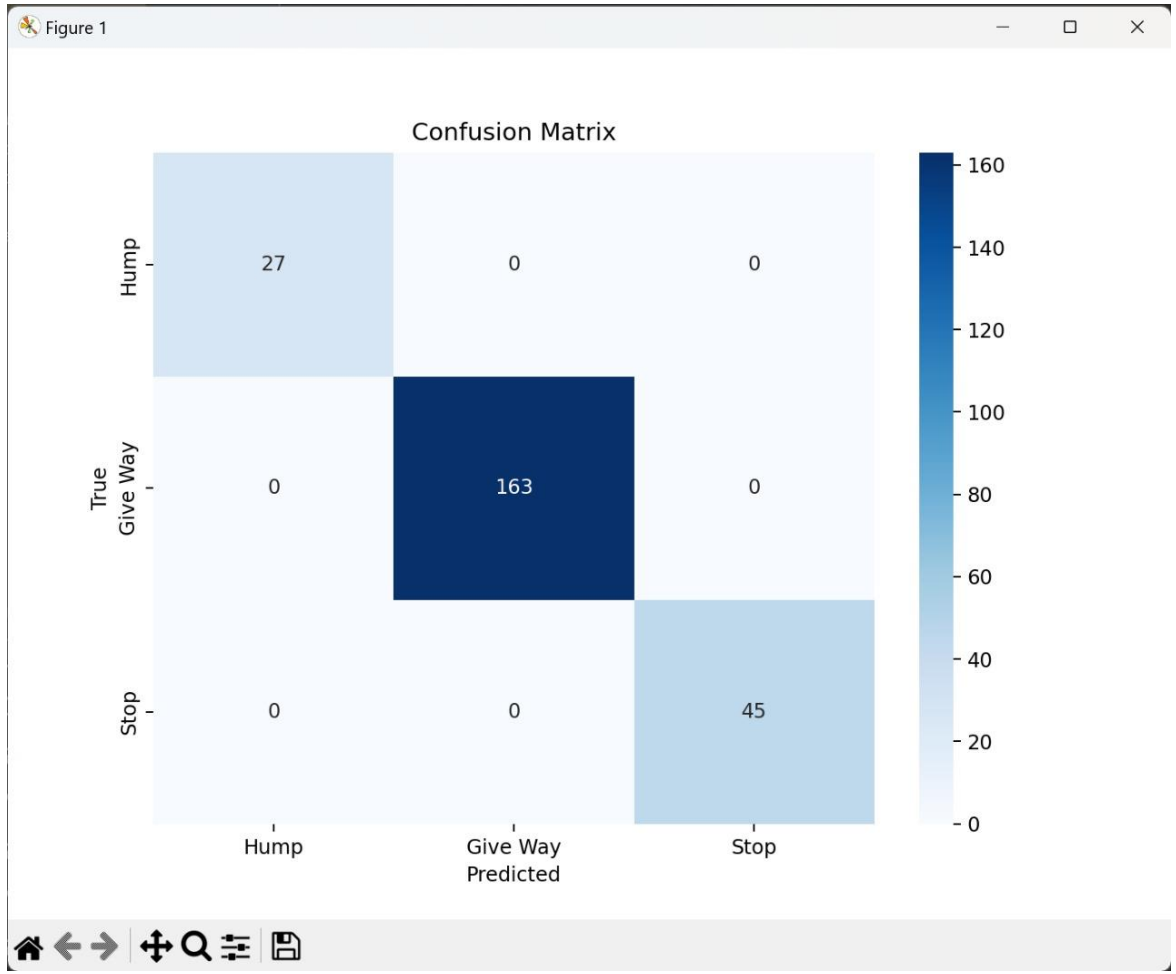
accuracy for test file



split file conf matrix



test file conf matrix



roc for test file