Mini Project Report

on

Outlier detection with GMM

Submitted by

Avinash pallapati : 24BEC036

Kesanapalli Madhava Naga Venkata Sai Ram : 24BEC019

Chinmay Pramod Ardalkar : 24BEC007

Nalage Soham Rajendra : 24BEC029

Under the guidance of

DEEPAK K T

Assistant Professor

Associate Dean - Research and Development [R&D]

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING.

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD

Certificate

This is to certify that the project report titled “Outlier Detection with Gaussian Mixture Model (GMM)” is a bona fide work carried out by students whose names are given below in partial fulfillment of the requirements for the,Bachelor of technology during the academic year 2025-26,The work presented in this report is original and has been completed under my supervision,To the best of my knowledge.

Roll No	Names of Students
24BEC036	Avinash pallapati
24BEC019	Sai Ram
24BEC007	Chinmay
24BEC029	Soham

DEEPAK K T
(Project Supervisor)

Contents

1.Introduction

This project focuses on building a Gaussian Mixture Model (GMM)-based classifier for the Statlog Shuttle dataset, a well-known multi-class and highly imbalanced dataset. The system is now fully restructured into a modular and reproducible pipeline with: • Command-line execution • Robust logging • Modular training, prediction, and evaluation functions • Automatic external test data generation • Complete automation using RUN.sh This ensures reproducibility, consistent execution, and easier debugging.

2.Related Work

Gaussian Mixture Models (GMMs) belong to generative probabilistic models capable of capturing multimodal distributions. Traditional work compares GMMs with discriminative classifiers such as SVMs and Random Forests. GMMs remain useful when class-conditional densities are important or when data naturally clusters. However, GMM performance depends heavily on class balance, sample size, and number of components. In this project, GMMs are used in a per-class manner, assigning predictions using maximum log-likelihood across class-specific GMMs.

3.Data and Exploratory Data Analysis (EDA)

The Statlog Shuttle dataset contains 7 numerical features and 7 classes. Class distribution is extremely imbalanced, with Class 1 dominating the dataset. This causes significant challenges in modeling minority classes such as classes 2, 6, and 7. Data preprocessing includes: • Train-test split using stratified sampling • Standardization using StandardScaler • PCA for visualization

4.Methods

The updated MAIN.py includes the following pipeline: 1. Loading dataset using ucimlrepo 2. Train-test split 3. Standardization using StandardScaler 4. Per-class fitting of Gaussian Mixture Models 5. Dynamic number of components: - 1 component if class size < 50 - 2 components otherwise 6. Prediction using maximum per-class log-likelihood 7. Evaluation via confusion matrix, accuracy, and PCA projection The pipeline is fully modular with three main functions: • train_and_save • predict_external • gmm_predict_from_dict

5.Automated Pipeline (RUN.sh)

The RUN.sh script automates the following: 1. Clone or update the GitHub repository 2. Create and activate a virtual environment 3. Install all dependencies 4. Train GMM models 5. Generate external test data 6. Run prediction on external data This ensures a one-command reproducible workflow.

6.Experimental Setup

The system was executed using MAIN.py via command line: • Training: python MAIN.py train • External prediction: python MAIN.py predict --external external/example_matched.csv • max_iter increased to 300 for better EM convergence • Logging enabled for transparency

7. Results

Confusion Matrix and PCA plots are shown below.

Figure 1: Confusion Matrix

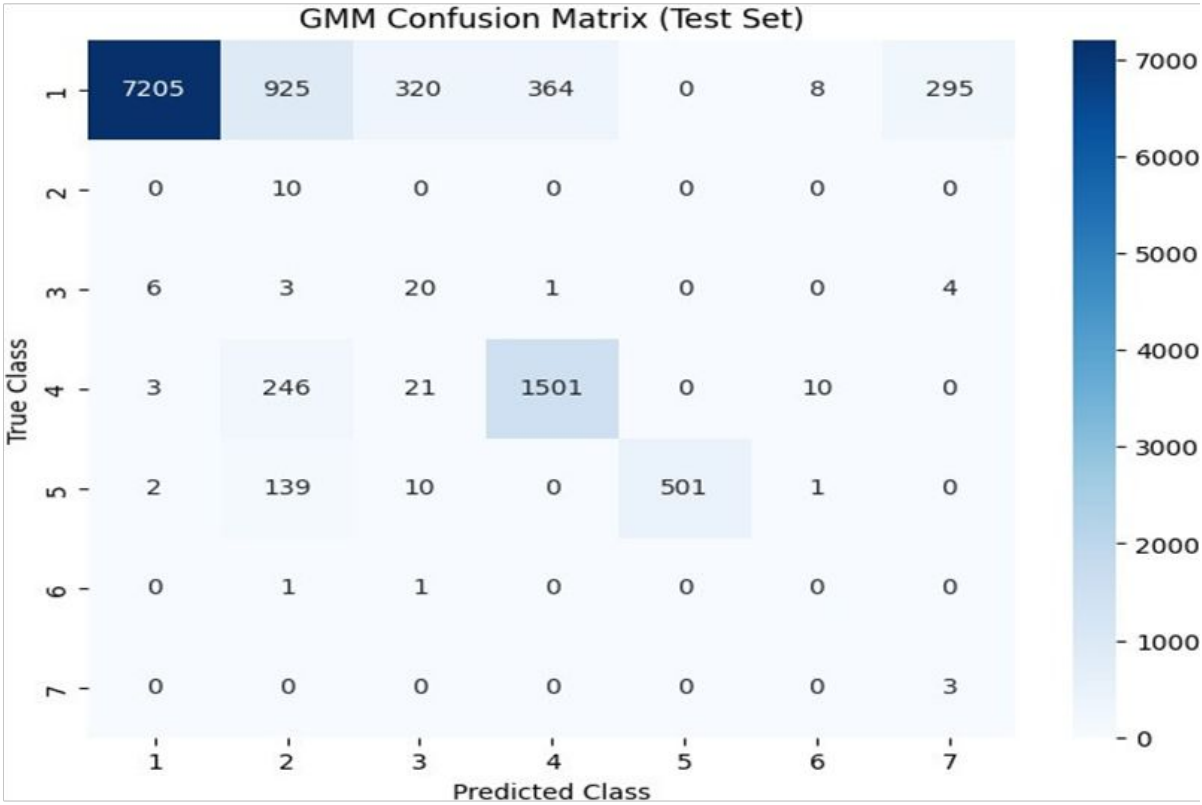
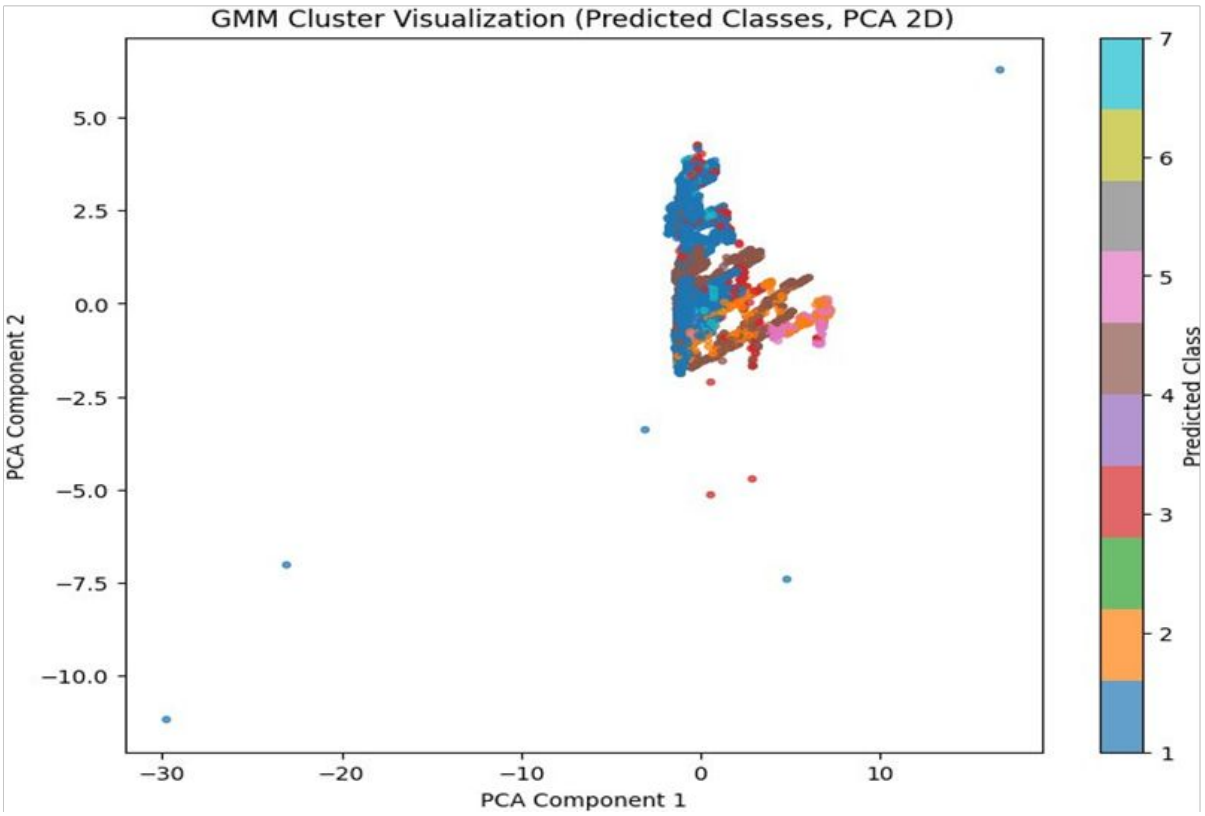


Figure 2: PCA 2D Projection



8.Error Analysis

Key observations: • Class imbalance strongly affects minority class performance • Classes 2, 6, and 7 have too few samples for stable GMM estimation • PCA visualization shows class overlap, especially among Classes 1, 3, 4, and 5 • Increasing the number of components or using discriminative classifiers may help Error sources: 1. Sparse minority class samples 2. Overlapping distributions 3. Limited mixture complexity

9.Conclusion and Future Work

The updated GMM pipeline performs reliably for major classes but struggles for minority classes due to imbalance. The modularized and automated workflow improves reproducibility and debugging. Future extensions may include: • SMOTE or oversampling • ROC/PR curve analysis • Using Bayesian GMMs • Hybrid discriminative–generative models • Ensemble techniques

10.Appendix: Code & Artifacts

The project includes the following files: • MAIN.py — main training and prediction pipeline • Test_Data_gen.py — automatic external dataset generator • RUN.sh — one-command reproducible execution • output/scaler_gmm_shuttle.joblib • output/gmm_class_model_shuttle.joblib • output/training_columns_shuttle.joblib • external/example_matched.csv