Final Project Report

# Introduction

The algorithm applies machine learning for sorting structured information. There were a number of processes the dataset had to go through before it was prepared for model class comparison and testing of model classifications. One of the top objectives of the project was classifying a single class utilizing alternative features from the input with conventional and neural network approaches. Each model was experimented with in order to discover the most suitable and effective model for the problem.

# Methodology

The following methodology was employed:  
I have embraced the data and examined it with exploratory data analysis (EDA) tools (refer to `generate\_eda\_analysis.py`).

- Preprocessing involved scaling the data to a normal level using `StandardScaler`, and dimensionality reduction with PCA if necessary. Numerous machine learning techniques have been learned and attempted, including:

• Logistic Regression

• Decision Tree Classifier

• Random Forest

• Gradient Boosting

• Support Vector Machines (SVM)

• Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA

• Multi-layer Perceptron (MLP) Neural Network

- The models were saved using joblib for future predictions, and a scaler was persisted for transforming new data consistently.

- Model performances were compared using accuracy score, F1-score, and confusion matrices.

# Results

The models tested were verified with test data. Following is a summary of the performance of the models:

-Random Forest and Gradient Boosting models were better than the other models in terms of accuracy and F1-score metrics.

- Support Vector Machines and MLP Neural Networks also performed well but required more tweaking of their parameters.

-Tree models were more intuitive, but MLP had more flexibility in dealing with non-linear relations.

Each model was saved as a `.pkl` file for reproducibility and future inference.

-The Naive Bayes classifier performed worst, as would be expected with datasets with complex feature interactions.

-K-Nearest Neighbors performed alright, but it's sensitive to feature scaling and the value of 'k'.

- Logistic Regression was a reliable place to start for classification.

- Generally, Random Forest and Gradient Boosting performed better than the rest for F1-score and also for accuracy.

# Discussion and Future Work

All the models performed well, with the top performers being Random Forest and Gradient Boosting because they were correct and consistent in their findings. We can improve in the future by;

- Adjusting by either grid search, or Bayesian optimization

- Adding additional steps in feature engineering for discovering concealed patterns in the data.

- Expanding the dataset or incorporating external datasets to improve model generalization.

- Evaluating model fairness and robustness under different distributions or real-world shifts.

Also, cross-validation and ROC-AUC analysis with more model checking might help further. Naive Bayes, as simple as it is, may perform better with better feature engineering.

KNN might perform better with the correct adjustment of its parameters or feature reduction in order to fix issues with too many features.