FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

ICP: Machine Learning Based Regression Analysis and Classification of Unmanned Ground Vehicle (UGV) Designs

Mirajul Islam

Graduate Research Assistant, L3Harris Institute for Assured Information

Computer Science Ph.D. Student

13 December 2024

ICP GitHub URL: https://github.com/islam-mirajul/ICP-Pattern-Recognition

Project Overview

Regression Analysis

Target = ‘Operational Range’
Features = Else All

Clustering and Classification

Custom K-means

Low Performance UGV	2131
Advanced Performance UGV	1656
Standard Performance UGV	564
Moderate Performance UGV	231

Classification

Target = ‘Design Class’
Features = Else All

Fig 1. Variety of Unmanned Ground Vehicles (UGVs)

Extract UVGS Performance Metrics
4582 Designs

Design ID	Chassis Type	Wheel Configuration	Footprint Area	Volume To Weight Ratio	Chassis Weight Percentage	Power Density	Power Efficiency	Energy Efficiency	Motor Power Density	Battery Energy Density	Motor To Weight Ratio	Battery To Weight Ratio	Agility Score	Stability Factor	Speed To Size Ratio	Wheel To Chassis Ratio	Terrain Adaptability (ratio)	Torque To Weight Ratio	Operational Endurance	Operational Range	Control Responsiveness	Control Precision	Sensor Coverage Ratio	Sensor Range Efficiency

Design Class
Low Performance UGV
Moderate Performance UGV
Standard Performance UGV
Advanced Performance UGV

Classification Target

Regression Analysis Target

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Background, Baselines, and Approach

We are exploring new methods to reduce both the cost and time associated with computer-based simulations of physical systems. As part of this effort, we aim to evaluate and compare models for regression and classification tasks, focusing on minimizing errors (MAE, MSE, RMSE, R^2) and maximizing performance metrics (accuracy, precision, recall, F1 score).

This approach involves exploring a variety of machine learning models, including third-party and custom models to predict and optimize simulation parameters, aiming for faster and more cost-effective simulation results.

Baseline Methods

For regression: Mean, Median, Persistence, Zero, and Decile-based predictions.

For classification: Most Frequent Class, Random Classifier, and Stratified Classifier.

Dataset Overview

Table 1. Dataset Overview

Design_ID	Chassis_Type	Wheel_Configuration	Footprint_Area	Volume_to_Weight_Ratio	Chassis_Weight_Percentage	Power_Density	Power_Efficiency	Energy_Efficiency	Motor_Power_Density	Battery_Energy_Density	Motor_to_Weight_Ratio	Battery_to_Weight_Ratio
1	chassis_hub_12WD_mini	12WD	1750	3.038194444	28.93518519	3.485714286	0.001967213	7.119594595	0.032972973	0.164079823	8.472222222	1157.407407
10	chassis_hub_4WD_standard	4WD	600	1.769911504	23.59882006	2.583333333	0.001935484	6.86457529	0.035227273	0.170769231	3.657817109	4129.79351
100	chassis_hub_8WD_heavy_duty	8WD	1350	2.969543147	33.84094755	2.849002849	0.0038	13.60810811	0.025	0.111213873	6.768189509	219.9661591
1000	chassis_hub_8WD_versatile	8WD	1000	2.264150943	28.30188679	4.083333333	0.000612245	1.883207071	0.025789474	0.093176471	7.396226415	415.0943396

Design_ID	Agility_Score	Stability_Factor	Speed_to_Size_Ratio	Wheel_to_Chassis_Ratio	Terrain_Adaptability (ratio)	Torque_to_Weight_Ratio	Operational_Endurance	Operational_Range	Control_Responsiveness	Control_Precision	Sensor_Coverage_Ratio	Sensor_Range_Efficiency
1	0.9	0.875	0.36	0.96	0.48	0.177064514	0.1215	1.0537	3529.786891	0.000118519	6.562047683	8.01
10	0.745355992	1.111111111	0.1	0.8	0.4	0.15042649	0.638761062	2.13351	1882.553009	0.000122499	69.40686205	83.41666667
100	0.924484332	0.882352941	0.422222222	1.107692308	0.553846154	0.132746531	0.043979695	0.26182	4792.666201	0.000113475	46.27124137	55.61111111
1000	0.730296743	0.833333333	0.075	1.2	0.6	0.399667245	0.006520755	0.02983	642.0435415	0.000117647	10.72717855	12.65

Dataset Splits

For Regression Analysis

Total Data Samples 4582
Training set size: 2749 samples - 60%
Validation set size: 916 samples – 20%
Testing set size: 917 samples – 20%

For Classification- Stratified Shuffle Split

Design Class	Training	Validation	Testing	Total
Low Performance UGV	1278	426	427	2131
Advanced Performance UGV	994	331	331	1656
Standard Performance UGV	338	113	113	564
Moderate Performance UGV	139	46	46	231
Total	2749	916	917	4582
Percentage	60.00%	19.99%	20.01%	100%

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Data Analysis for Regression

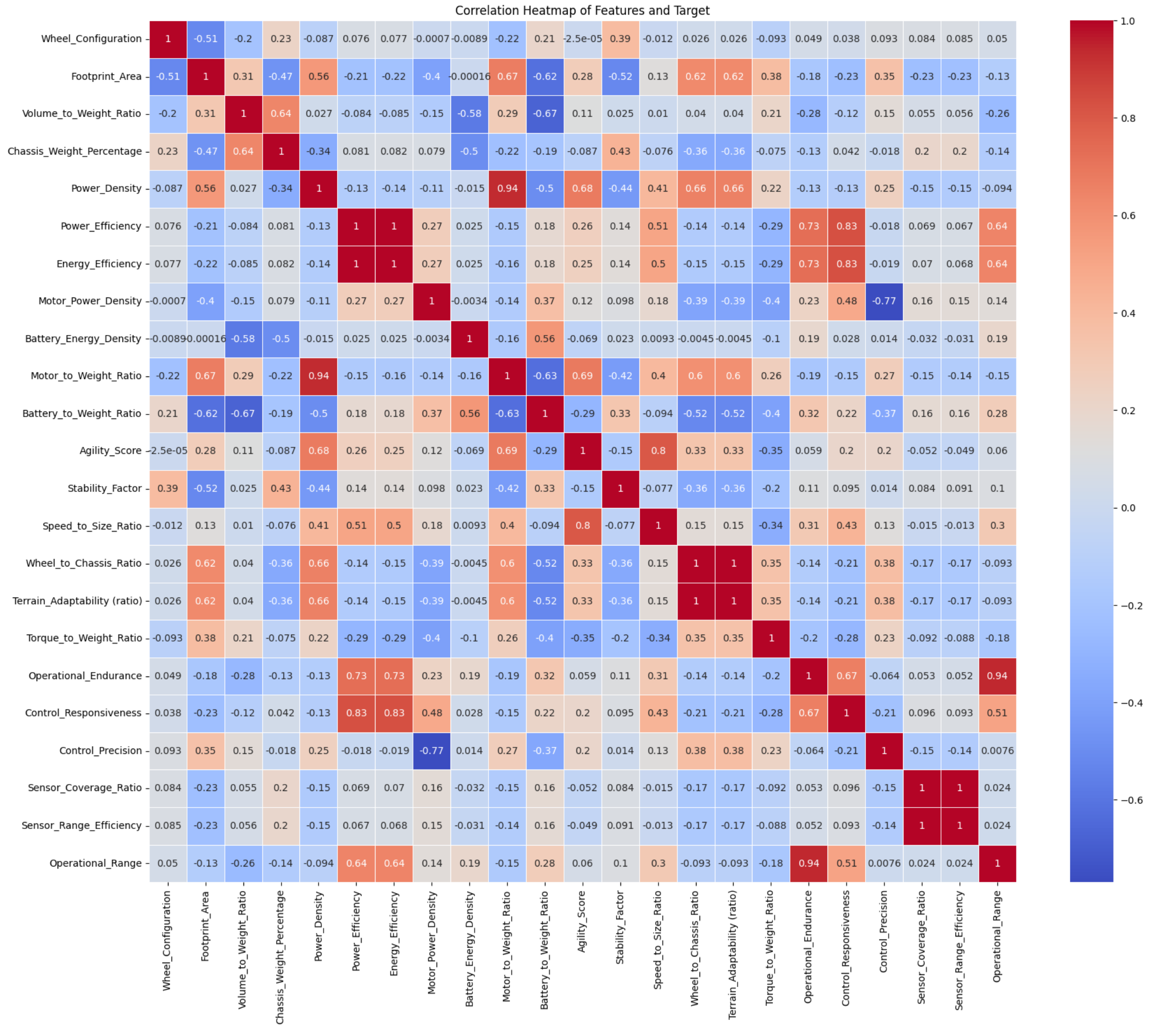

Fig 2. Feature-Target Correlation Heatmap

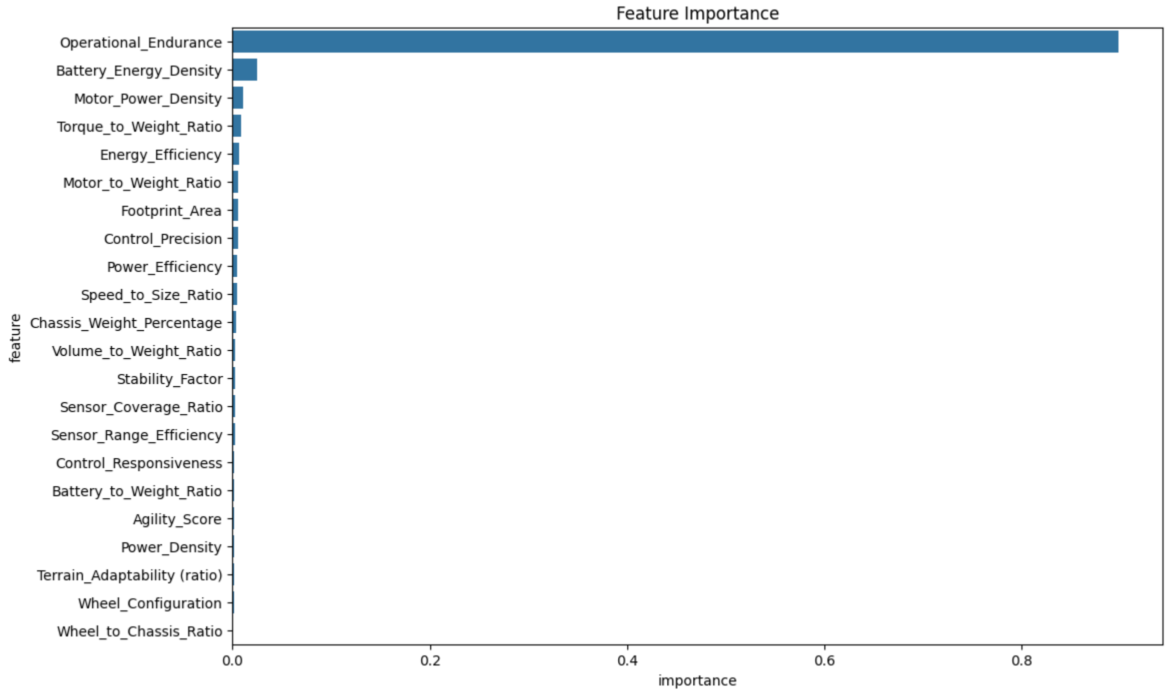

Fig 3. Top Features by Importance using Random Forest Regressor

Methodology - Regression Analysis

Sklearn Models

Support Vector Regression

Random Forest Regressor

Gradient Boosting Regressor

Lasso Regression

Ridge Regression

Custom Model

Custom Bayesian Linear Regression

Baselines Custom Methods

Mean Prediction

Median Prediction

Persistence Model (Last Value Prediction)

Zero Prediction

Decile-based Prediction: This predicts the decile (10% intervals) of the target values

Performance metrics are calculated:

Mean Absolute Error (MAE)

Mean Squared Error (MSE)

Root Mean Squared Error (RMSE)

R-squared (R^2)

Performance Visualizations:

Actual vs. Predicted values for each model

Results-Regression Analysis

Table 2: Model Performance Metrics Comparison

Custom **Bayesian Linear Regression** is the best overall performer

Model Group	Model Name	MAE	MSE	RMSE	R^2
Sklearn	SVR	5.5754	1366.6779	36.9686	0.0794
Sklearn	Random Forest	1.3521	173.0158	13.1535	0.8834
Sklearn	Gradient Boosting	1.4401	141.6666	11.9024	0.9046
Sklearn	Ridge Regression	2.6183	64.7935	8.0494	0.9564
Sklearn	Lasso Regression	2.4324	69.5298	8.3385	0.9532
Custom Model	Bayesian Linear Regression	2.5805	64.7743	8.0482	0.9564
Baselines (Custom)	Mean Prediction	10.1794	1487.9680	38.5742	-0.0024
Baselines (Custom)	Median Prediction	6.9908	1532.2448	39.1439	-0.0322
Baselines (Custom)	Persistence Model	7.2899	1521.5258	39.0067	-0.0250
Baselines (Custom)	Zero Prediction	7.0190	1533.7396	39.1630	-0.0332
Baselines (Custom)	Decile-based Prediction	5.4560	1333.9443	36.5232	0.1014

Results-Regression Analysis cont..

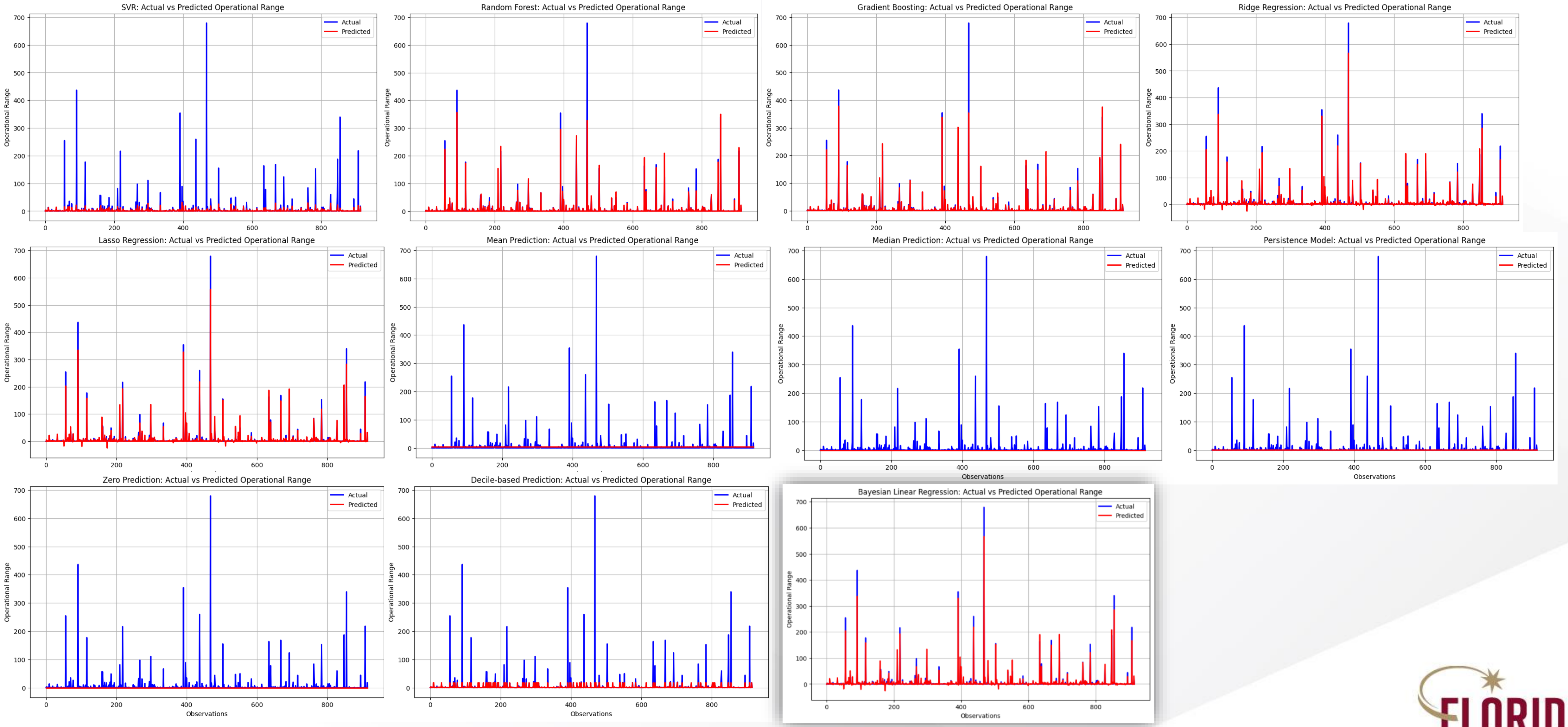

Fig 4. Actual vs Predicted Operational Range

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Clustering & Classification

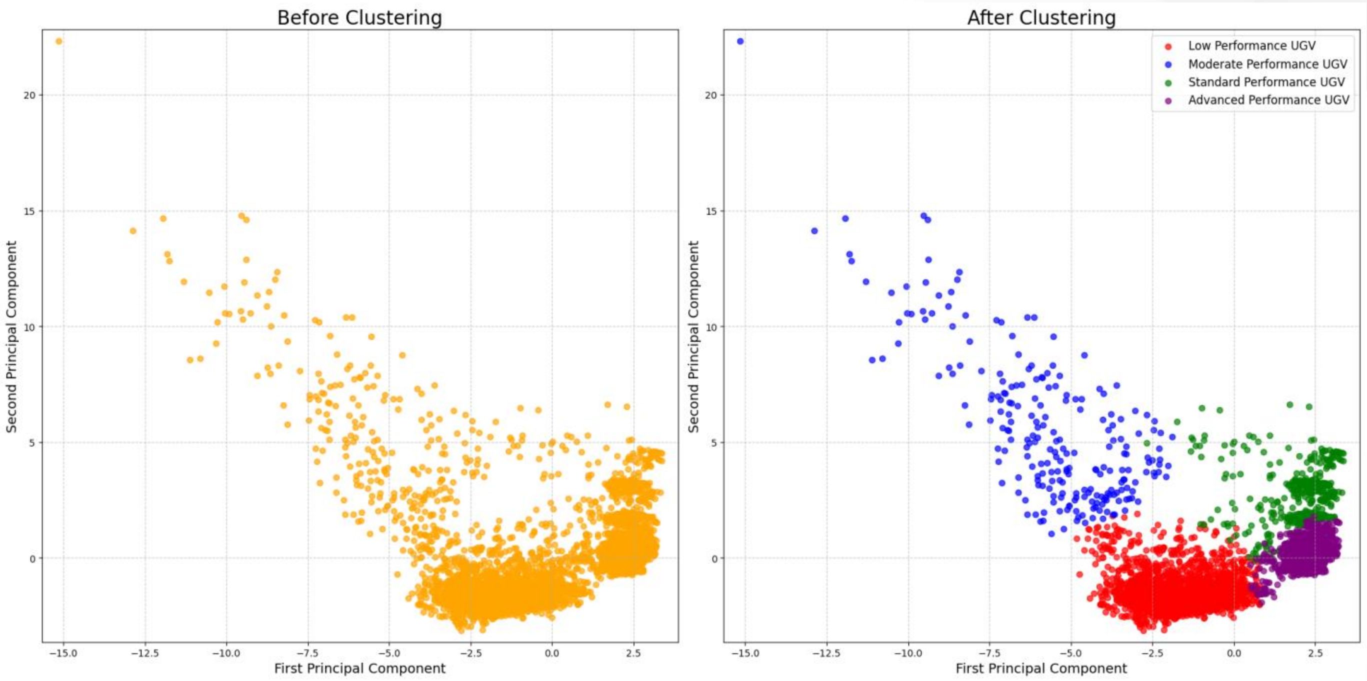

Fig 5. PCA Visualization of UGV Performance Classes: Pre- and Post-Clustering

- PCA (Principal Component Analysis) is used here to reduce the dimensionality of the data, making it easier to visualize and interpret.
- Convex Hulls define the outer boundary of each class, showing the region containing all points. Ellipses visualize the spread and orientation of the class data, reflecting the distribution and variance within the 2D PCA space.

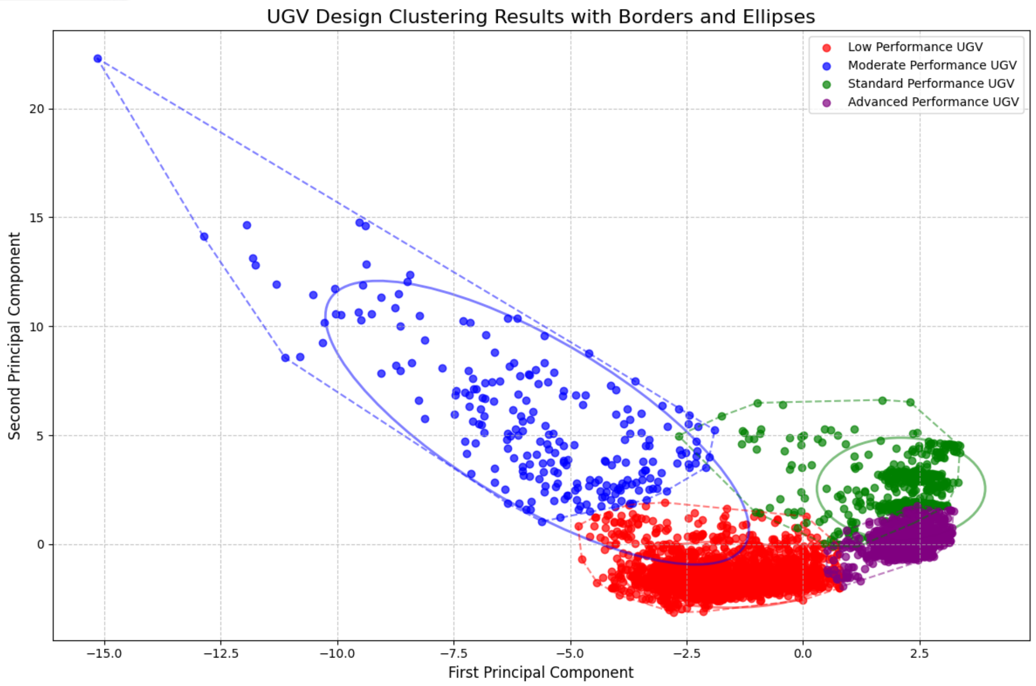

Fig 6. UGV Design Clustering with Convex Hulls and Ellipses in 2D PCA Space

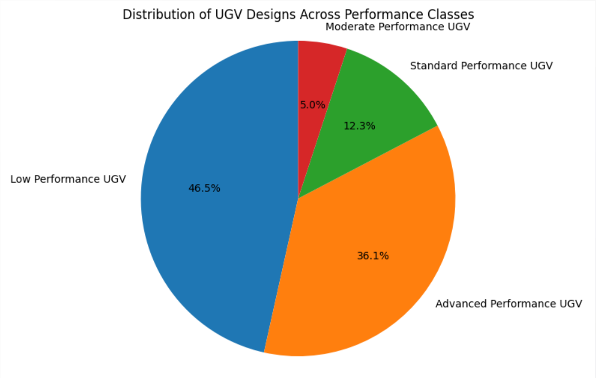

Fig 7. Pir Chart Distribution of UGV Designs

Low Performance UGV	2131
Advanced Performance UGV	1656
Standard Performance UGV	564
Moderate Performance UGV	231

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Methodology - Classification

Custom Models
1. Custom Linear Discriminant Analysis (LDA)
2. Custom Quadratic Discriminant Analysis (QDA)

Sklearn Models
1. Support Vector Machine Classifier
2. Gaussian Process Classifier
3. Random Forest Classifier
4. Gradient Boosting Classifier
5. Fisher's Linear Discriminant
6. Logistic Regression
7. Perceptron
8. Naive Bayes Classifier
9. Stochastic Gradient Descent Classifier

Baseline Sklearn Models
1. Baseline Model (Most Frequent Class)
2. Random Classifier
3. Stratified Classifier

Performance Metrics
Classification Report
1. Accuracy
2. Precision (weighted average)
3. Recall (weighted average)
4. F1 Score (weighted average)

Confusion Matrix

Results-Classification

Table 3. Model's Performance Comparison

Model	Type	Accuracy	Precision	Recall	F1 Score
Custom LDA	Custom Models	1.0000	1.0000	1.0000	1.0000
Custom QDA	Custom Models	1.0000	1.0000	1.0000	1.0000
Baseline Model (Most Frequent Class)	Baseline Sklearn Models	0.4656	0.2168	0.4656	0.2959
Random Classifier	Baseline Sklearn Models	0.2290	0.3385	0.2290	0.2591
Stratified Classifier	Baseline Sklearn Models	0.4057	0.4030	0.4057	0.4043
Support Vector Machine	Sklearn Models	0.9956	0.9956	0.9956	0.9956
Gaussian Process Classifier	Sklearn Models	0.6325	0.5689	0.6325	0.5838
Random Forest Classifier	Sklearn Models	0.9989	0.9989	0.9989	0.9989
Gradient Boosting Classifier	Sklearn Models	0.9989	0.9989	0.9989	0.9989
Fisher's Linear Discriminant	Sklearn Models	0.9673	0.9682	0.9673	0.9667
Logistic Regression	Sklearn Models	0.9117	0.9116	0.9117	0.9106
Perceptron	Sklearn Models	0.7274	0.6703	0.7274	0.6848
Naive Bayes Classifier	Sklearn Models	0.9477	0.9514	0.9477	0.9476
Stochastic Gradient Descent Classifier	Sklearn Models	0.7361	0.6830	0.7361	0.6951

Custom LDA and Custom QDA both have perfect scores (Accuracy: 1.0000, Precision: 1.0000, Recall: 1.0000, F1 Score: 1.0000), making them the best performers.

Results-Classification cont..

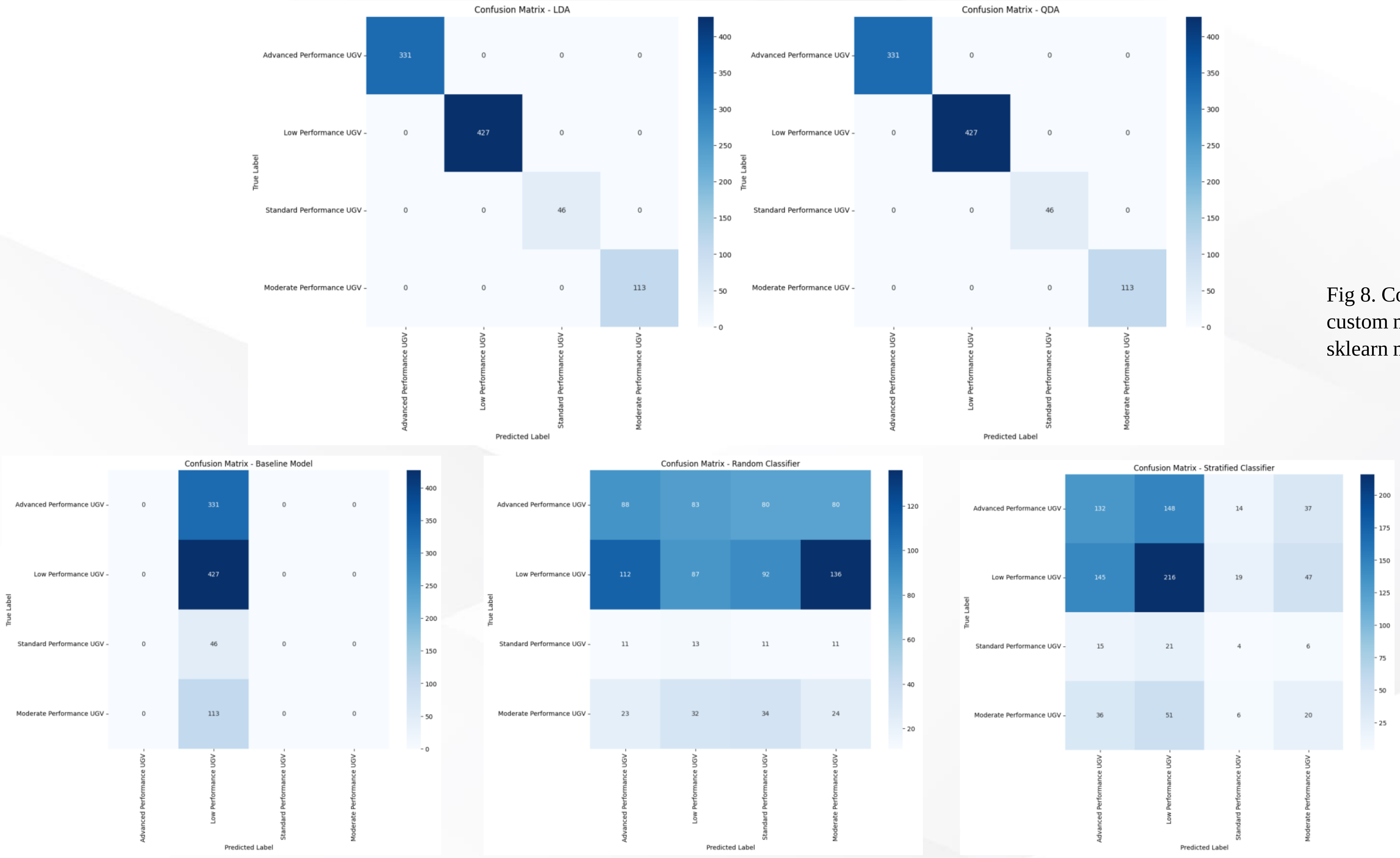

Fig 8. Confusion Matrix for custom models and baseline sklearn models

Results-Classification cont..

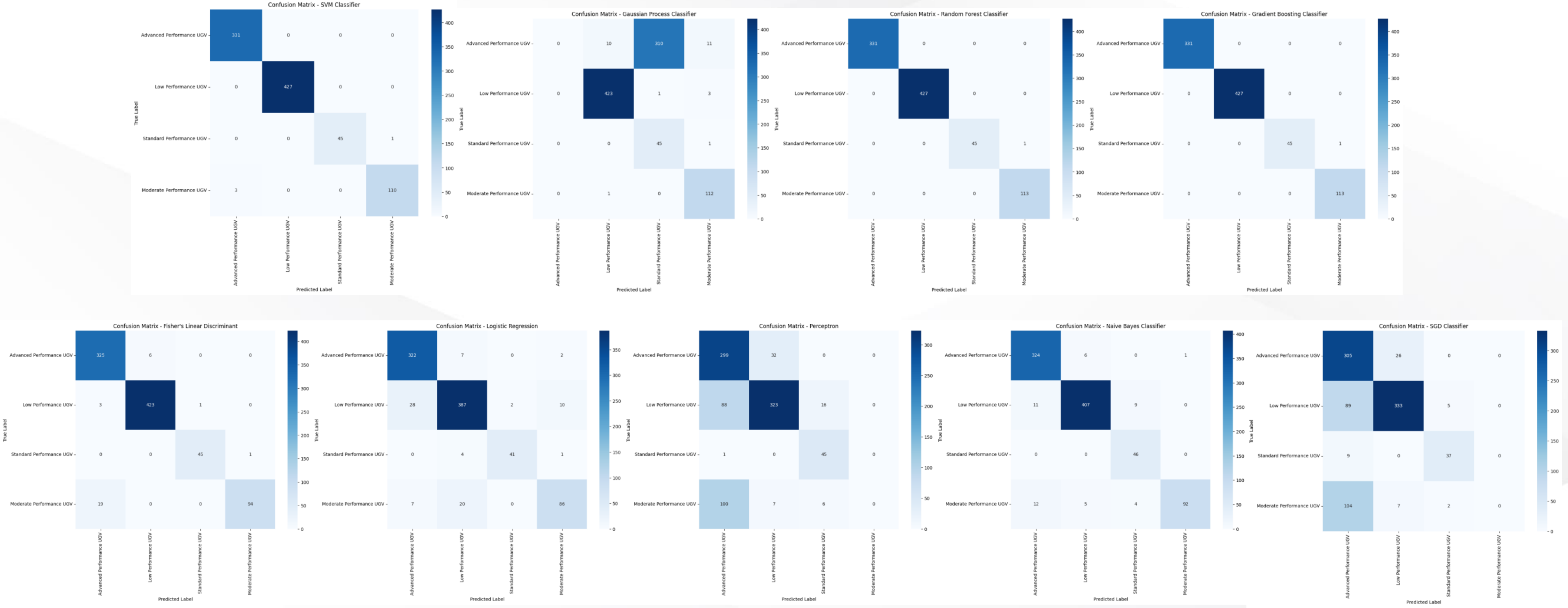

Fig 9. Confusion Matrix for sklearn models

Conclusions

For both regression and classification tasks, custom implementation models performed the best.

Regression Task: The Bayesian Linear Regression model achieved competitive results, with a high $R2$ score of 0.9564 and an RMSE of 8.0482, comparable to Ridge Regression. Among third-party models, Gradient Boosting performed slightly better, but the custom model remains highly effective.

Classification Task: Custom models, particularly Custom LDA and Custom QDA, achieved perfect performance with an accuracy, precision, recall, and F1 score of 1.0000, outperforming all other models. Among third-party models, Random Forest Classifier and Gradient Boosting Classifier achieved near-perfect performance (~99.89%), making them strong contenders.

Acknowledgement: In certain instances, this project employs the scikit-learn library for specific analyses. For further details, please refer to the official scikit-learn website: [https://scikit-learn.org/stable/]

Thank you!