MATERIAL SELECTION FOR EV CHASSIS USING SUPERVISED ML ALGORITHMS

(AID-505: Course Project)

Maharshi Gohil
maharshi_pg@mfs.iitr.ac.in
Enrollment No: 23565005

Shitiz Goel
shitiz_g@mfs.iitr.ac.in
Enrollment No: 23566013

Abstract

This project explores the application of multiple machine learning algorithms in the context of material selection for electric vehicle (EV) chassis. The dataset used for this analysis comprises various features related to material properties like Density, UTS, Yield Strength, Modulus of elasticity etc. All the major supervised machine learning models are trained and evaluated for this classification problem. Finally, the results are compared to identify the most effective models for material selection.

1. Main Objectives:

-Predicting the suitability of materials for EV chassis using different ML algorithms.
-Comparing the performance of these models and determining the best models for the given dataset.

2. Introduction:

In the automotive industry, especially for electric vehicles, selecting the right materials for the chassis is critical for performance, safety, and sustainability. This project aims to leverage machine learning algorithms to assist in the decision-making process for material selection. Various algorithms such as Artificial Neural Networks (ANN), Support Vector Machines (SVM), Decision Tree, Bagged Trees, Random Forest, AdaBoost, K Nearest Neighbours (KNN), Naïve Bayes and Logistic Regression are trained and evaluated.

3. Dataset and Pre-processing:

The dataset, available on Kaggle [1], has 1552 rows consisting material properties. Each row has 15 features, but in that data some features had missing values so we will only take appropriate features and create another dataset consisting of only 8 features, which was already created on Kaggle [1].

Feature attributes:
Material - Material name
Su - Ultimate Tensile Strength in MPa
Sy - Yield Strength in MPa
E - Elastic Modulus in MPa
G - Shear Modulus in MPa
mu - Poisson's Ratio in Units of Length
Ro - Density in Kg/m^3
Use - Target

The column 'Use' is the value to be predicted. If it is 'True', we will select that material; otherwise, it will not get selected. Now, non- numerical values must be converted to numerical to train our models. For 'Use', True and False are converted to 1 and 0, respectively.

Pre-processing involves scaling numerical values and handling imbalanced data.

3.1 Scaling the data:

During scaling, we bring all the features under the same scale. Scaling helps our model understand the data. If the feature values are on a similar scale, our models can understand the data better and give better results.
Standard scaling is sensitive to outliers because it relies on the mean and standard deviation but here our dataset has not much outliers so we can use standard scalar for scaling.

3.2 Imbalanced data:

The below chart shows the imbalance in the dataset. Only 8.69% of data belongs to class 'True'. An unbalanced dataset is not good because most ML algorithms try to maximize the accuracy.

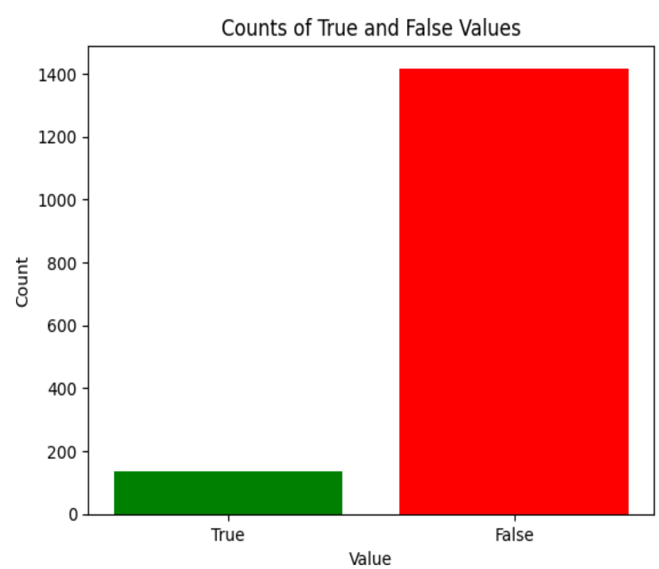


Figure 1: Visualization of imbalance in data

3.3 Visualizing the data in two dimensions:

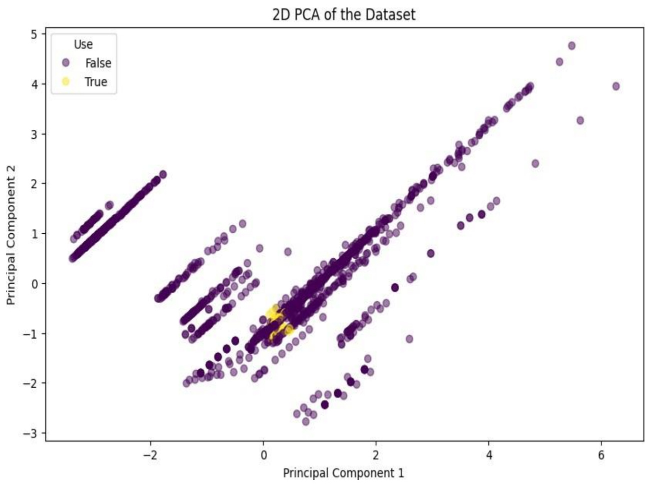


Figure 2: Visualizing the data in two dimensions

3.4 Splits into train and test sets:

	0	1	Instances
Training Data (75%)	1058	106	1164
Testing Data (25%)	359	29	388

4. <u>Machine Learning Algorithms:</u>

1. Artificial Neural Network (ANN):

	precision	recall	f1-score	support
0	1.00	0.99	0.99	359
1	0.90	0.97	0.93	29
accuracy			0.99	388
macro avg	0.95	0.98	0.96	388
weighted avg	0.99	0.99	0.99	388

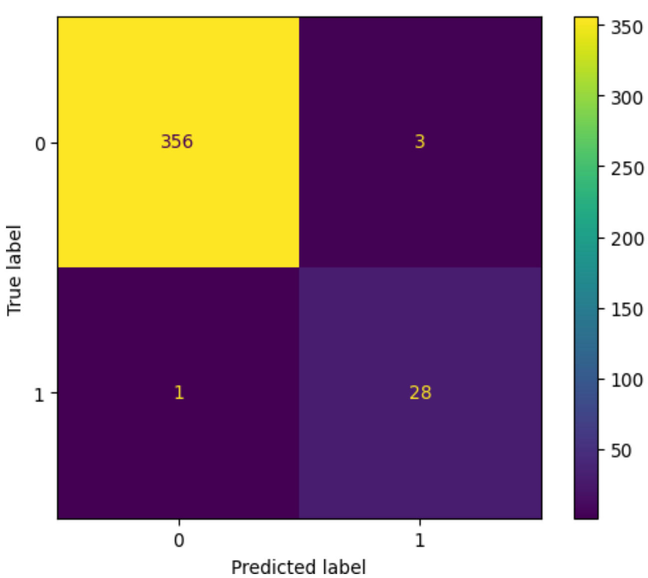


Figure 3: Confusion Matrix for ANN

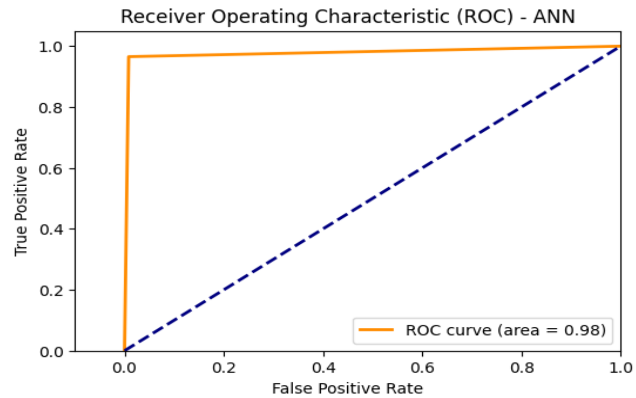


Figure 4: ROC Curve for ANN

2. Support Vector Machine (SVM):

	precision	recall	f1-score	support
0	0.98	0.98	0.98	359
1	0.77	0.69	0.73	29
accuracy			0.96	388
macro avg	0.87	0.84	0.85	388
weighted avg	0.96	0.96	0.96	388

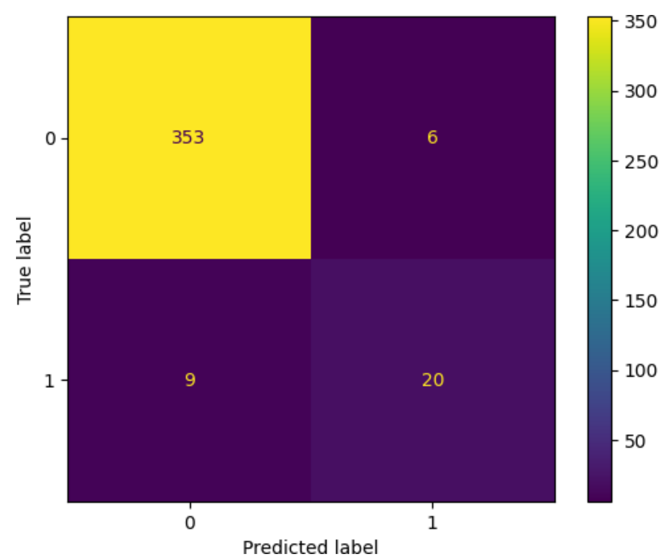


Figure 5: Confusion Matrix for SVM

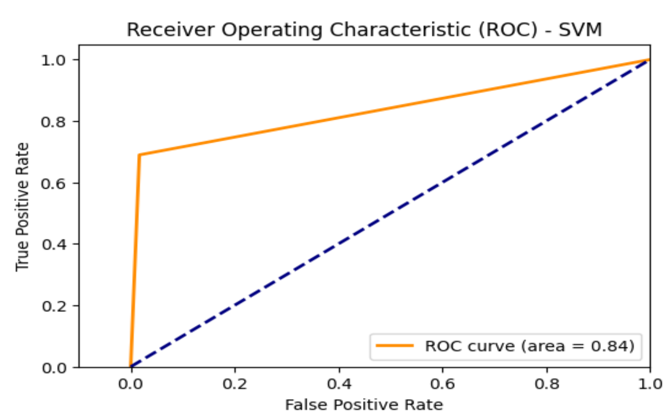


Figure 6: ROC Curve for SVM

3. DECISION TREE:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	359
1	1.00	0.97	0.98	29
accuracy			1.00	388
macro avg	1.00	0.98	0.99	388
weighted avg	1.00	1.00	1.00	388

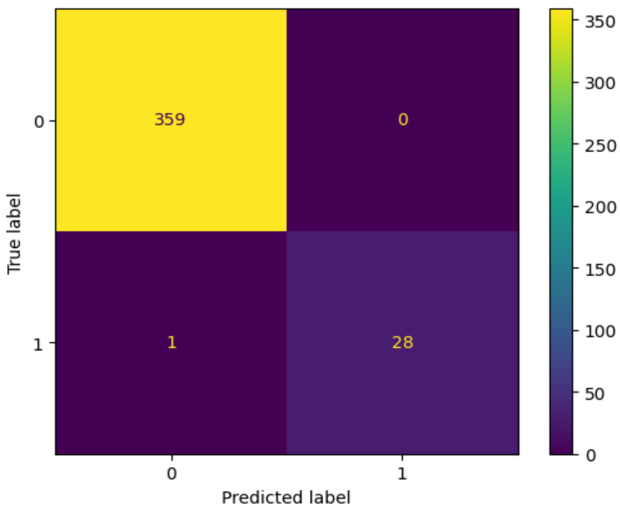


Figure 7: Confusion Matrix for Decision Tree

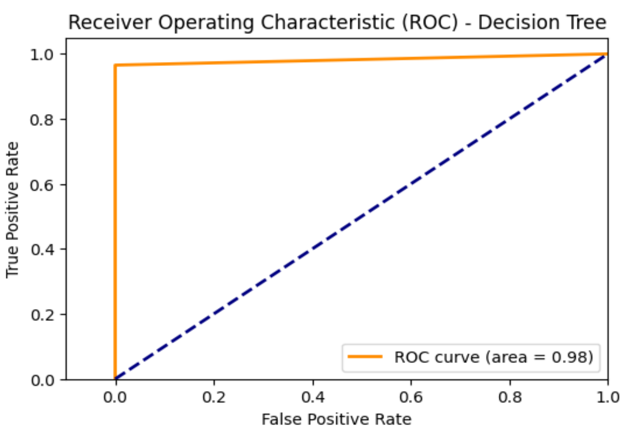


Figure 8: ROC Curve for Decision Tree

4. BAGGED TREES:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	359
1	1.00	0.97	0.98	29
accuracy			1.00	388
macro avg	1.00	0.98	0.99	388
weighted avg	1.00	1.00	1.00	388

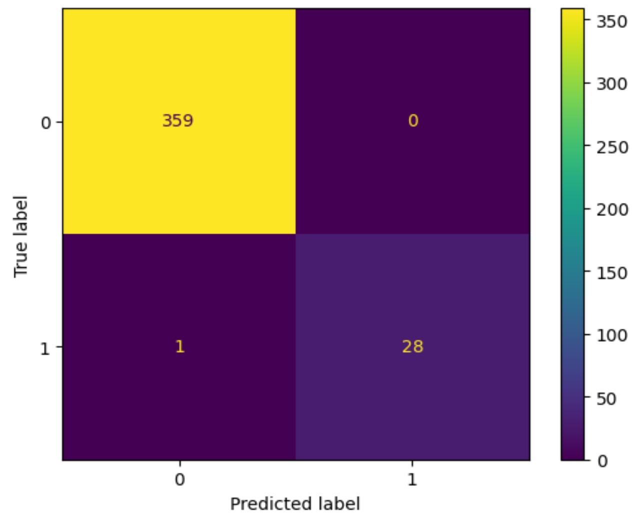


Figure 9: Confusion Matrix for Bagged Tree

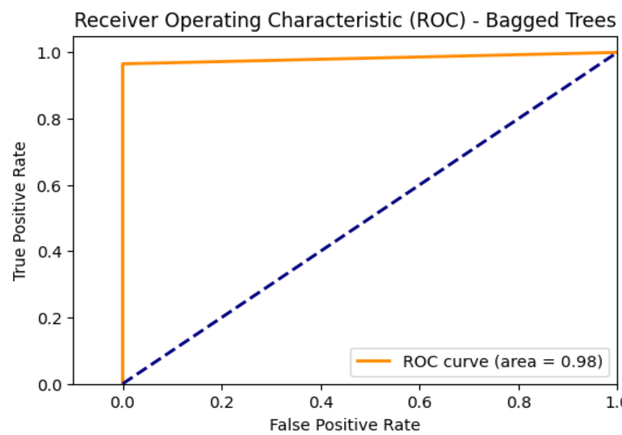


Figure 10: ROC Curve for Bagged Tree

5. RANDOM FOREST:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	359
1	1.00	0.97	0.98	29
accuracy			1.00	388
macro avg	1.00	0.98	0.99	388
weighted avg	1.00	1.00	1.00	388

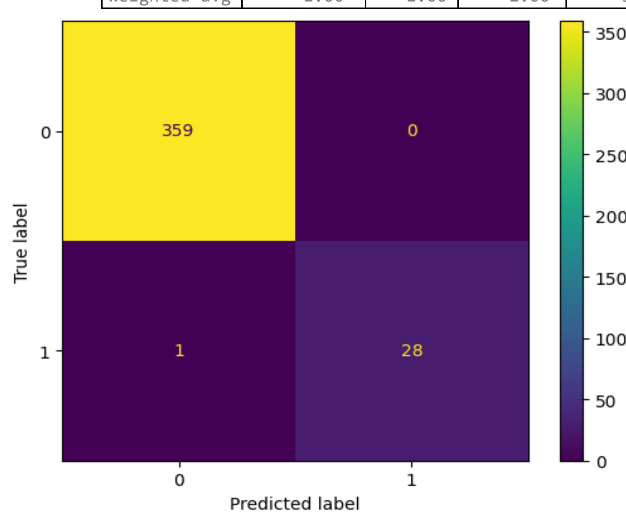


Figure 11: Confusion Matrix for Random Forest

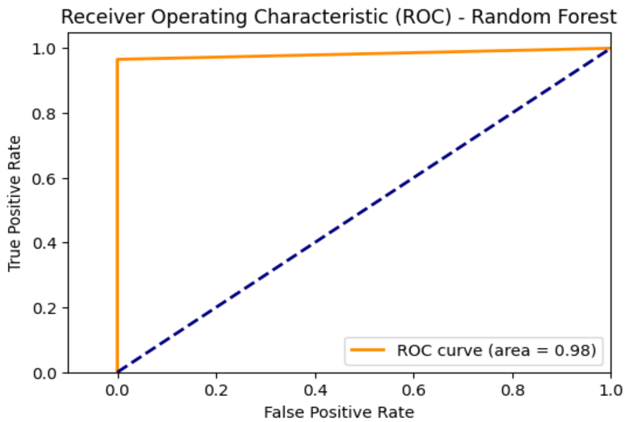


Figure 12: ROC Curve for Random Forest

6. DECISION TREE WITH ADABOOST:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	359
1	1.00	0.97	0.98	29
accuracy			1.00	388
macro avg	1.00	0.98	0.99	388
weighted avg	1.00	1.00	1.00	388

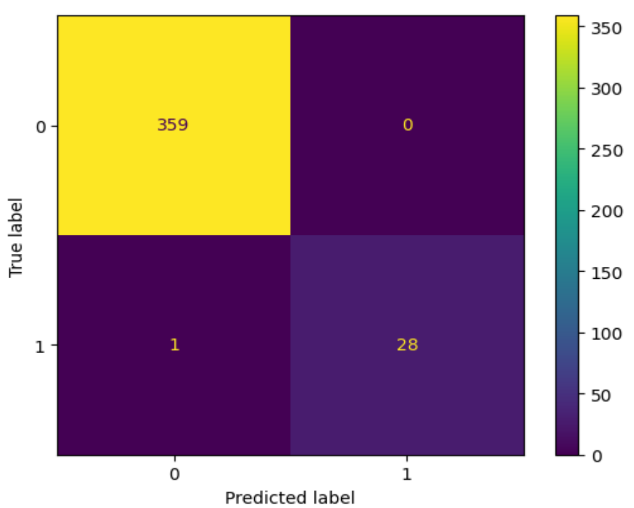


Figure 13: Confusion Matrix for Decision Tree with Adaboost

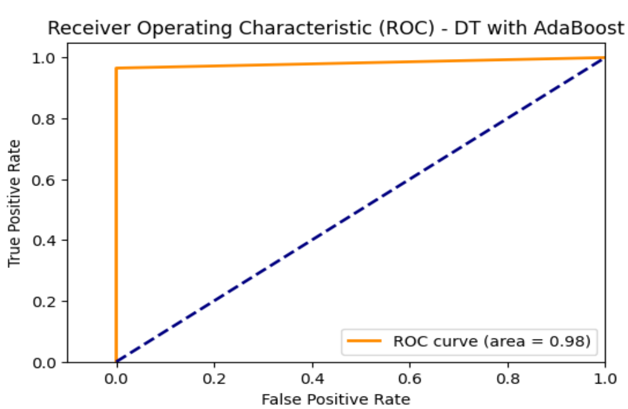


Figure 14: ROC Curve for Decision Tree with Adaboost

7. K-Nearest Neighbour (KNN):

	precision	recall	f1-score	support
0	0.99	0.99	0.99	359
1	0.87	0.90	0.88	29
accuracy			0.98	388
macro avg	0.93	0.94	0.94	388
weighted avg	0.98	0.98	0.98	388

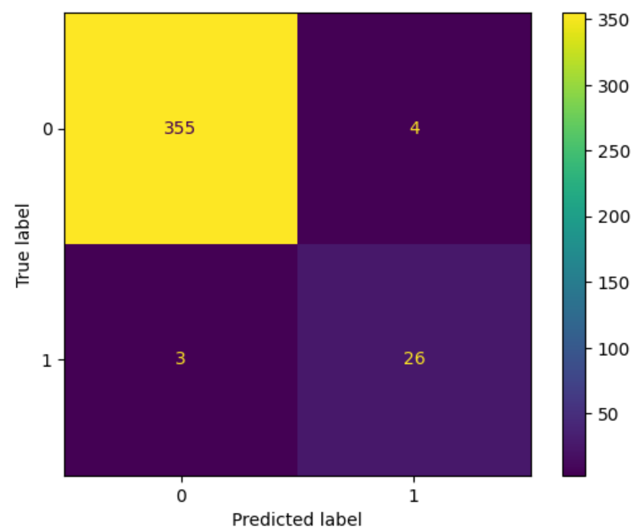


Figure 15: Confusion Matrix for KNN

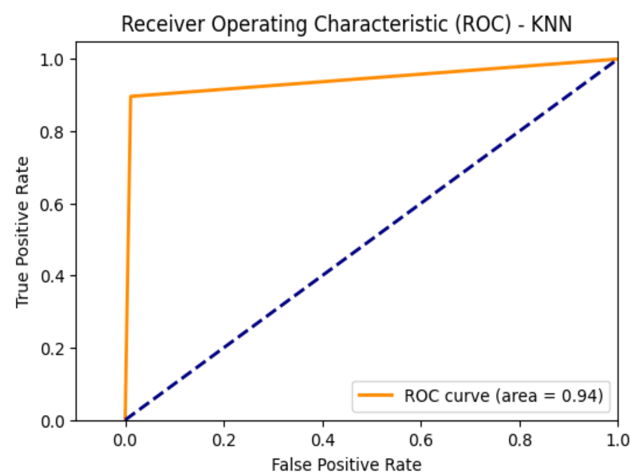


Figure 16: ROC Curve for KNN

8. NAÏVE BAYES:

	precision	recall	f1-score	support
0	0.99	0.84	0.91	359
1	0.31	0.93	0.47	29
accuracy			0.84	388
macro avg	0.65	0.88	0.69	388
weighted avg	0.94	0.84	0.87	388

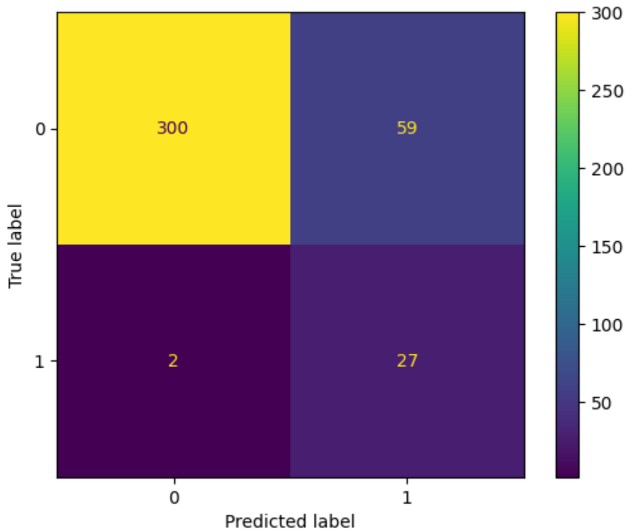


Figure 17: Confusion Matrix for Naïve Bayes

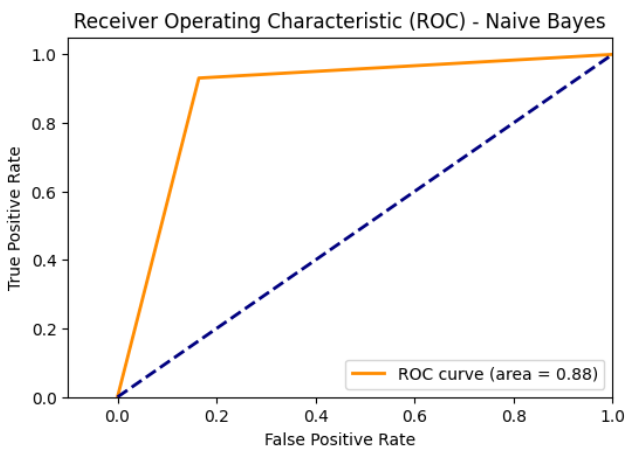


Figure 18: ROC Curve for Naïve Bayes

9. LOGISTIC REGRESSION:

	precision	recall	f1-score	support
0	0.99	0.88	0.93	359
1	0.39	0.93	0.55	29
accuracy			0.88	388
macro avg	0.69	0.91	0.74	388
weighted avg	0.95	0.88	0.90	388

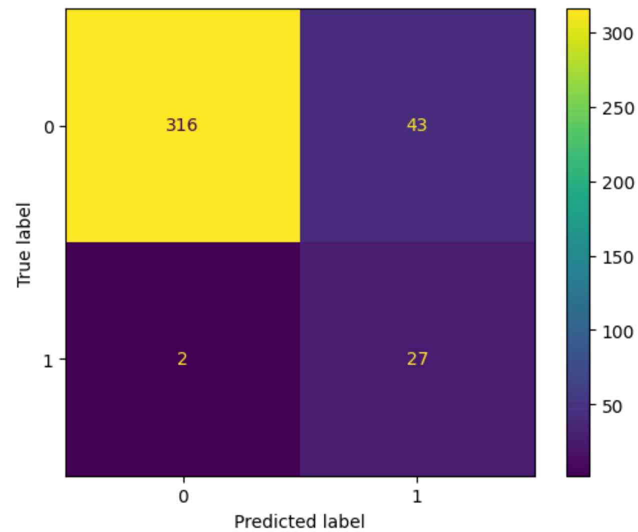


Figure 19: Confusion Matrix for logistic Regression

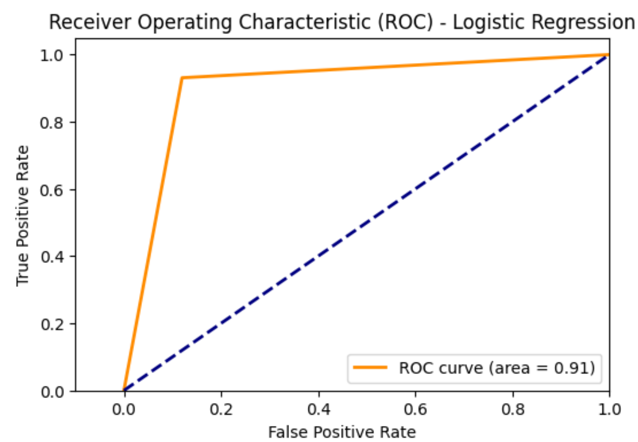


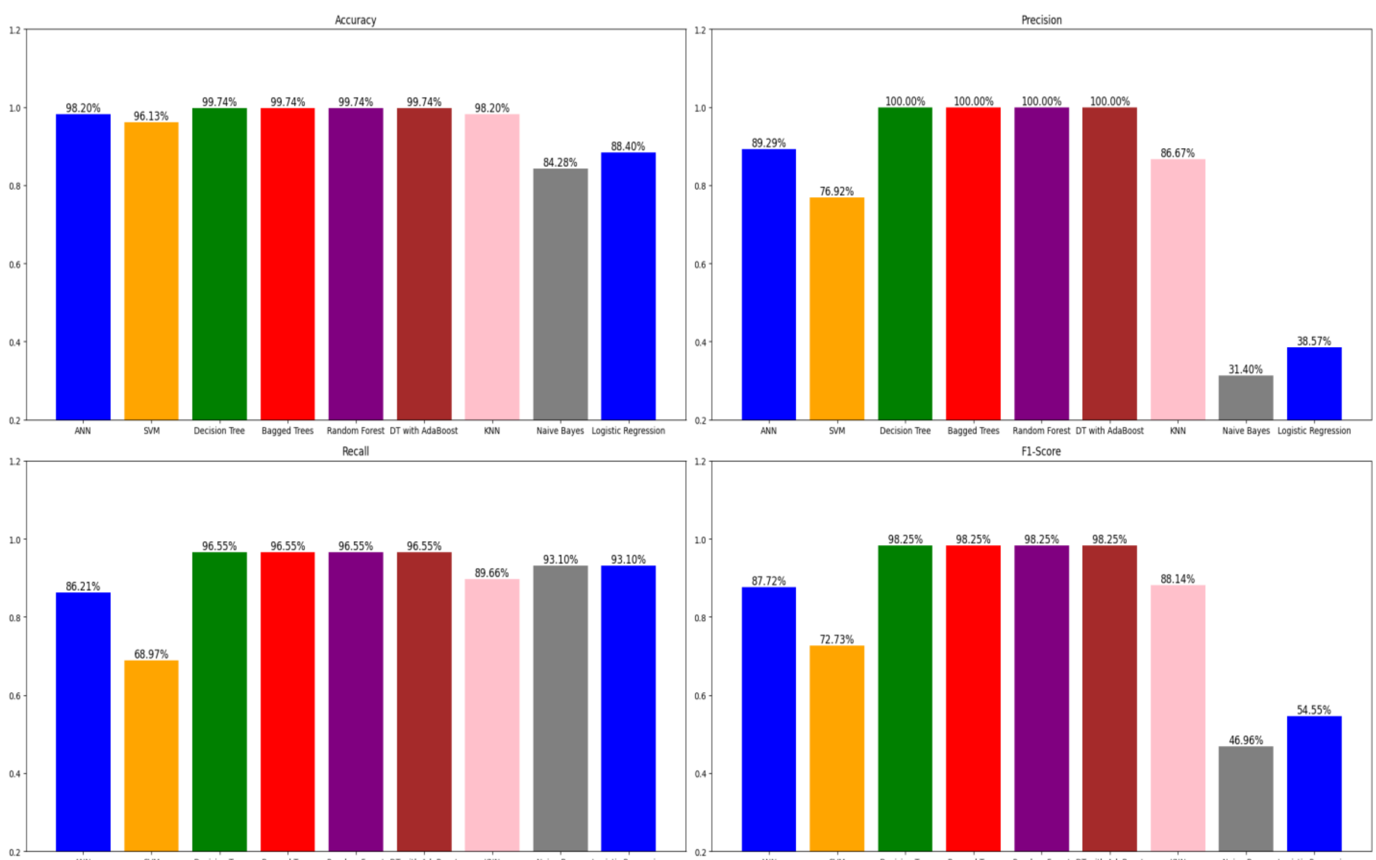
Figure 20: ROC Curve for Logistic Regression

5. Results and Conclusion:

The experiment(test) explored how different supervised machine learning classifiers assessed materials for electric vehicle (EV) chassis based on their properties. Each model was thoroughly tested to see how well it could help decide which materials were suitable for this purpose.

Classifier	Parameters Used before Tuning
ANN	Hidden_layer_sizes=(5), max_iter=500, activation='tanh', solver='lbfgs'
SVM	C=10, kernel='rbf'
Decision Tree	Criterion='gini'
Bagged Trees	Estimator = DecisionTreeClassifier()
Random Forest	Default
Adaboost	Default
KNN	n_neighbors=5
Naïve Bayes	GaussianNB()
Logistic Regression	Penalty=None, solver='lbfgs', max_iter=200, class_weight='balanced'

Classifier	Best Parameters Using GridSearchCV	Accuracy before Tuning	Accuracy after Tuning
ANN	'activation': 'tanh', 'alpha': 0.001, 'hidden_layer_sizes': (50,), 'learning_rate': 'adaptive', 'solver': 'lbfgs'	0.9896	0.9974
SVM	'C': 100, 'degree': 2, 'gamma': 1, 'kernel': 'rbf'	0.9613	0.9871
Decision Tree	'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2	0.9974	0.9510
Bagged Trees	'base_estimator__criterion': 'gini', 'base_estimator__max_depth': 20, 'base_estimator__min_samples_leaf': 1, 'base_estimator__min_samples_split': 2, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 100	0.9974	0.9639
Random Forest	'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 50	0.9974	0.9974
Adaboost	'base_estimator__criterion': 'gini', 'base_estimator__max_depth': 30, 'base_estimator__min_samples_leaf': 4, 'base_estimator__min_samples_split': 5, 'learning_rate': 1, 'n_estimators': 50	0.9974	0.9690
KNN	'n_neighbors': 7, 'p': 1, 'weights': 'distance'	0.9819	0.9896
Naïve Bayes	'var_smoothing': 1e-09	0.8427	0.9123
Logistic Regression	'C': 0.001, 'class_weight': None, 'max_iter': 100, 'penalty': 'l1', 'solver': 'liblinear'	0.8840	0.9252



Model Performance Overview:

- **Decision Tree and Ensemble Methods (Bagged Trees, Random Forest, AdaBoost):**

Exceptional performance across precision, recall, and F1-score metrics.
Robustly classify suitable materials with high accuracy and minimal misclassification.

- **Artificial Neural Network (ANN):**

Demonstrated remarkable accuracy and precision but slightly lower recall compared to decision tree-based models.

- **Support Vector Machine (SVM):**

Showcased balanced performance in precision and recall, providing reliable classification with notable accuracy.

- **K-Nearest Neighbours (KNN):**

Offered high accuracy and reasonable recall, though with a slight compromise in precision.
Provided a fast and effective approach to material selection.

- **Logistic Regression:**

Provided reasonably good precision and recall, though it showed some limitations in correctly identifying suitable materials.

- **Naïve Bayes:**

While achieving high precision for suitable materials, it struggled with lower recall, possibly due to an inherent assumption of feature independence.

Imbalanced Data and Performance:
The analysis was performed on a significantly imbalanced dataset (with only 8.69% of data labelled as 'True' materials). Decision tree-based methods, along with ensemble methods like Random Forest and AdaBoost, exhibited robust performance despite this imbalance, showcasing their resilience to skewed class distributions.

Conclusion:
In conclusion, the tuning process has generally improved the performance of most classifiers. Decision tree-based methods, especially ensemble methods, continue to demonstrate robust performance. The Artificial Neural Network (ANN) significantly improved its accuracy after tuning. Support Vector Machine (SVM) and K-Nearest Neighbours (KNN) maintained their balanced performance. Logistic Regression and Naïve Bayes showed improvements but may still have limitations in correctly identifying suitable materials, especially in recall. The decision tree-based methods remain the most promising approaches for material selection in EV chassis design.

6. References:

1. Dataset: https://www.kaggle.com/datasets/purushottamnawale/materials