

# Project Title: RCS Space debris size prediction

**Data Set Name: Predicting RCS SIZE of Space Debris**

# Data Set Source: Kaggle

**Data set Link:** [**link**](https://www.kaggle.com/code/kandhalkhandeka/predicting-rcs-size-of-space-debris)

DataSet description :

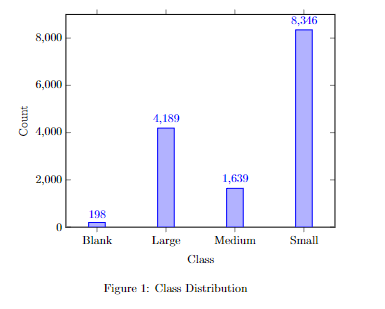
The dataset consists of 14372 records with 40 columns. Each record represents an object and contains various parameters describing its orbital characteristics.

1. **Categorical Variables:**
   * **CCSDS\_OMM\_VERS:** Represents the version of the CCSDS Orbit Data Messages used, categorizing the data into different message versions.
   * **ORIGINATOR:** Indicates the entity or organization responsible for providing the data, categorized into different originators.
   * **CENTER\_NAME:** Specifies the center or institution associated with the reference frame used for describing the object's orbit, categorized into different centers.
   * **REF\_FRAME:** Denotes the reference frame utilized for describing the object's orbit, categorized into various frames of reference.
   * **TIME\_SYSTEM:** Specifies the time system used for temporal data representation, categorized into different time systems.
   * **MEAN\_ELEMENT\_THEORY:** Indicates the mean element theory utilized for calculating orbital elements, categorized into different theories.
   * **RCS\_SIZE:** Represents the Radar Cross Section size category, categorizing the detectability of the object into different size ranges (e.g., SMALL, MEDIUM, LARGE).
   * **COUNTRY\_CODE:** Represents the country code associated with the launch or origin of the object, categorized based on country codes.
   * **SITE:** Specifies the site or location from which the object was launched, categorized based on launch sites.
   * **FILE:** Indicates the name or identifier of the file containing the orbital data, categorized based on file names or identifiers.
   * **TLE\_LINE0, TLE\_LINE1, TLE\_LINE2:** Represents the lines of the Two-Line Element (TLE) data, categorized based on TLE line numbers.
2. **Numerical Variables:**
   * **OBJECT\_ID:** Provides a unique identifier for the space debris object, enabling individual object identification.
   * **LAUNCH\_DATE:** Represents the date of the object's launch, recorded as numerical values for dates.
   * **DECAY\_DATE:** Indicates the date of orbital decay or re-entry into Earth's atmosphere, recorded as numerical values for dates.
   * **GP\_ID:** Provides a unique identifier for the ground point associated with the object, facilitating ground point identification and tracking.
3. **Temporal Variables:**
   * **CREATION\_DATE:** Represents the date and time when the data was created or generated, recorded as temporal values.
4. **Text Variables:**
   * **COMMENT:** Provides additional comments or notes about the dataset, recorded as textual information.
   * **OBJECT\_NAME:** Represents the name or designation of the space debris object, recorded as textual data.
5. **Identifier Variables:**
   * **OBJECT\_ID:** Serves as a unique identifier for the space debris object, facilitating individual object tracking and referencing.
   * **GP\_ID:** Serves as a unique identifier for the ground point associated with the object, enabling ground point tracking and referencing.

NULL value representation of the dataset

|  |  |
| --- | --- |
| Name of column | Number of NA |
| OBJECT\_TYPE" | 0 |
| | "OBJECT\_AGE" | 0 |
| | "INCLINATION" | 0 |
| | "CENT\_FOCUS\_DIST" | 0 |
| | "ECCENTRICITY" | 0 |
| | "PERIOD" | 0 |
| | "SEMIMAJOR\_AXIS" | 39 |
| | "MEAN\_MOTION" | 0 |
| | "APOAPSIS" | 0 |
| | "PERIAPSIS" | 49 |
| | "MEAN\_ANOMALY" | 0 |
| | "ARG\_OF\_PERICENTER" | 0 |
| | "RA\_OF\_ASC\_NODE" | 0 |

Class Distribution of data



Description of work done:

The project was dedicated to predicting the Radar Cross Section (RCS\_size) of space debris, crucial for ensuring the safety and sustainability of space operations. Extensive efforts were invested in collecting and preprocessing data on space debris characteristics and RCS\_size measurements. Various machine learning models, including Random Forest, Gradient Boosting Machine, SVM, Logistic Regression, Naive Bayes, and Decision Tree, were meticulously evaluated for their ability to predict RCS\_size.

After rigorous evaluation, the Random Forest model emerged as the optimal choice, showcasing superior performance across multiple metrics including accuracy, precision, recall, specificity, and F1 score. This finding underscores the pivotal role of machine learning in space debris mitigation efforts, highlighting the significance of accurate RCS\_size prediction for assessing collision risks.

Looking ahead, future directions entail refining the models further, integrating domain expertise to enhance predictive capabilities, and devising robust strategies for managing space debris effectively. This study serves as a testament to the transformative potential of machine learning in addressing complex challenges in the realm of space exploration and safety.

Data Preprocessing

|  |  |
| --- | --- |
| Rows | 24794 |
| Columns | 14 |

Data Distribution:

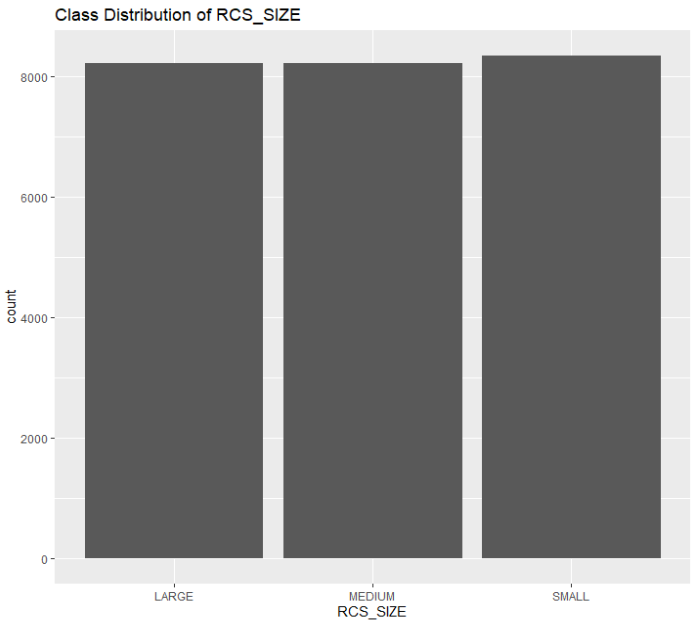
|  |  |
| --- | --- |
| Large | 8224 |
| Medium | 8224 |
| Small | 8326 |

Class imbalance is treated using ROSE:

ROSE, or Random Over-Sampling Examples

In ROSE, synthetic samples are generated for the minority class to balance the class distribution. The algorithm randomly selects examples from the minority class and generates synthetic samples by interpolating between them. This interpolation process helps create new instances that are similar to existing minority class samples but introduce variability to avoid overfitting.

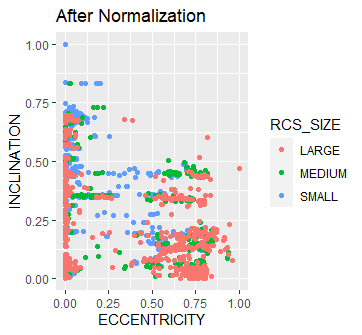
ROSE helps improve the performance of machine learning models by ensuring that each class is adequately represented during training. By balancing the class distribution, models trained on datasets preprocessed with ROSE are less likely to be biased towards the majority class and can better generalize to unseen data.



Normalization:

Normalization is a data preprocessing technique used to scale numerical features to a specific range, typically between 0 and 1 or -1 and 1. It ensures that all features have the same scale, which can improve the performance of machine learning models and algorithms.

In normalization, each feature is adjusted based on its minimum and maximum values in the dataset. The process involves subtracting the minimum value from each data point and then dividing by the range (the difference between the maximum and minimum values). This transformation ensures that the values of each feature are relative to its minimum and maximum values, making them comparable across features.

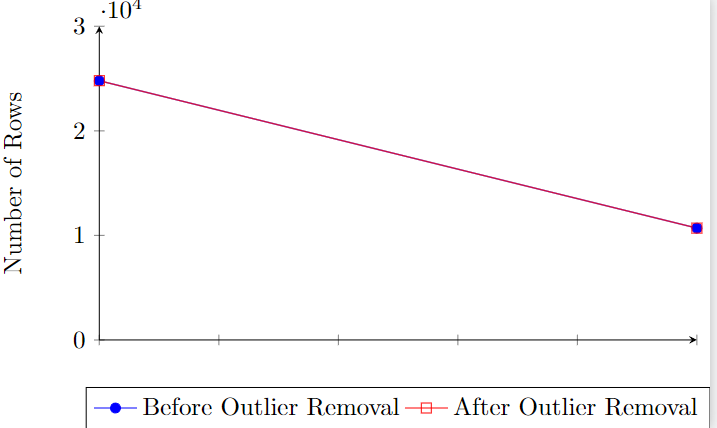


OutLier Removal:

An outlier is an observation in a dataset that significantly deviates from the rest of the data points. It can be an unusually high or low value compared to the majority of the observations. Outliers can distort statistical analyses and machine learning models, leading to inaccurate results.

The interquartile range (IQR) is a measure of statistical dispersion that describes the range of the middle 50% of the data. It is calculated as the difference between the third quartile (Q3) and the first quartile (Q1). The IQR provides a measure of the spread of the central portion of the data, making it resistant to outliers.

To identify outliers using the IQR method, a range is defined as 1−1.5×IQR*Q*1−1.5×IQR to 3+1.5×IQR*Q*3+1.5×IQR. Any data point outside this range is considered an outlier and can be removed or treated accordingly. This method is robust to outliers and provides a systematic approach to handling them in datasets.



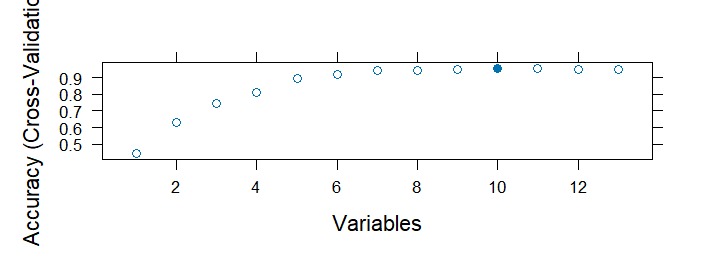
Number of rows

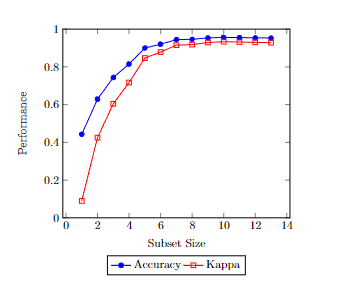
|  |  |
| --- | --- |
| Before | 24794 |
| After | 10696 |

Feature Selection:

**Feature Selection:**

Recursive Features Elimination (RFE):





Mutual Information:

[1] "OBJECT\_TYPE" "OBJECT\_AGE" "INCLINATION" "CENT\_FOCUS\_DIST"

[5] "ECCENTRICITY" "PERIOD" "SEMIMAJOR\_AXIS" "MEAN\_MOTION"

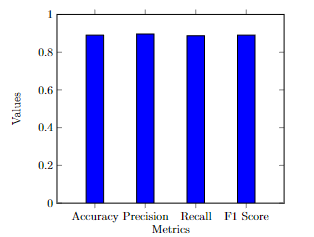
[9] "APOAPSIS" "PERIAPSIS" "MEAN\_ANOMALY" "ARG\_OF\_PERICENTER"

[13] "RA\_OF\_ASC\_NODE"

Model training:

Gradient Boosting Machine: Gradient Boosting Machine (GBM) is an ensemble learning technique that sequentially adds weak learners to minimize the loss function. It employs gradient descent to optimize the model by focusing on the mistakes of the previous learners. GBM commonly uses decision trees as weak learners and incorporates regularization to prevent overfitting. It is renowned for its high predictive accuracy and is widely used in various machine learning tasks, such as classification and regression.

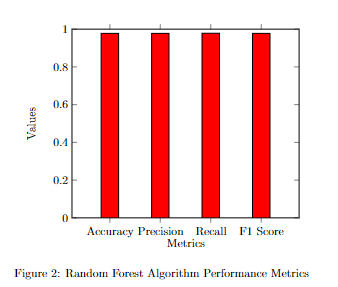
|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Large | Medium | Small |
| Accuracy | 0.890603085553997 | 0.890603085553997 | 0.890603085553997 |
| Precision | 0.960552268244576 | 0.807387862796834 | 0.922196796338673 |
| Recall | 0.855887521968366 | 0.9 | 0.90561797752809 |
| F1 score | 0.905204460966543 | 0.851182197496523 | 0.913832199546485 |
| Specificity | 0.987261146496815 | 0.899931459904044 | 0.945556445156125 |

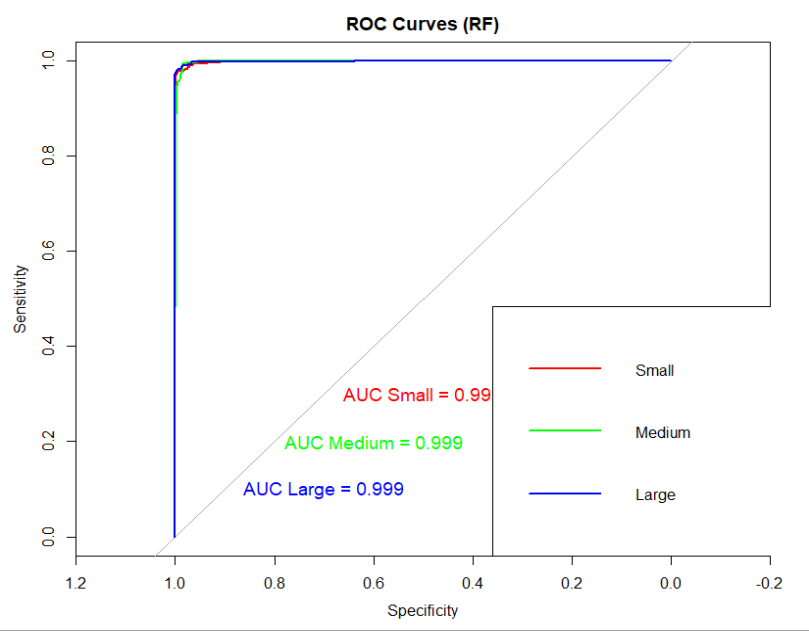


Random Forest:

Random Forest is an ensemble learning method that builds multiple decision trees during training. Each tree in the forest is trained on a random subset of the training data and features, and the final prediction is made by aggregating the predictions of individual trees (via averaging or voting). This technique helps reduce overfitting and improves generalization performance. Random Forest is known for its robustness, scalability, and ability to handle high-dimensional data, making it a popular choice for classification and regression tasks in machine learning.

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Large | Medium | Small |
| Accuracy | 0.977559607293128 | 0.977559607293128 | 0.977559607293128 |
| Precision | 0.991087344028521 | 0.94413407821229 | 0.996519721577726 |
| Recall | 0.977152899824253 | 0.994117647058824 | 0.965168539325843 |
| F1 score | 0.984070796460177 | 0.968481375358166 | 0.980593607305936 |
| Specificity | 0.996815286624204 | 0.972583961617546 | 0.99759807846277 |

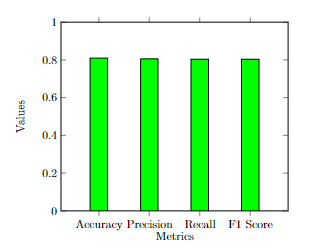


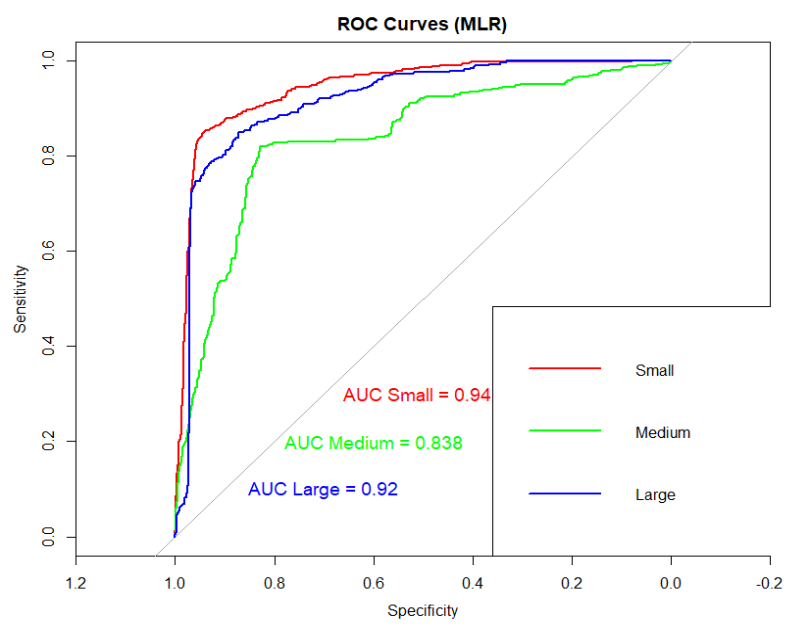


Multinomial Logistic regression :

Multinomial logistic regression is a statistical method used for modeling outcomes with more than two categories. It extends binary logistic regression to handle multiple classes by using a softmax function to compute the probabilities of each class. The model estimates the relationship between the independent variables and the probabilities of the different classes. It is widely used in classification tasks where the target variable has more than two categories, such as predicting the species of a flower based on its features. Multinomial logistic regression provides interpretable coefficients and can handle both numerical and categorical predictors.

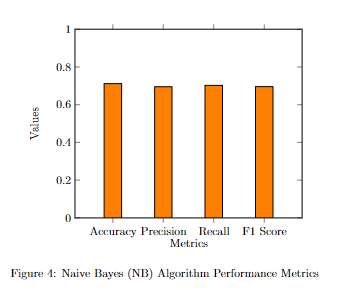
|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Large | Medium | Small |
| Accuracy | 0.809724170172978 | 0.809724170172978 | 0.809724170172978 |
| Precision | 0.817184643510055 | 0.705722070844687 | 0.893939393939394 |
| Recall | 0.785588752196837 | 0.761764705882353 | 0.861797752808989 |
| F1 score | 0.801075268817204 | 0.732673267326733 | 0.877574370709382 |
| Specificity | 0.936305732484076 | 0.85195339273475 | 0.927141713370697 |

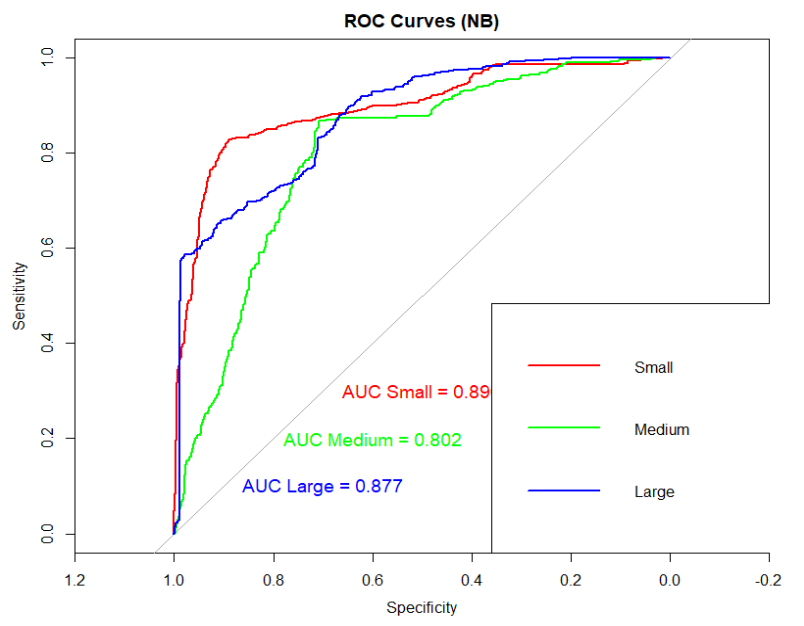




Naïve Bayes: Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem with an assumption of independence between predictors. It calculates the probability of each class given a set of input features and selects the class with the highest probability as the prediction. Despite its simplicity and the "naive" assumption of feature independence, Naive Bayes often performs well in practice, particularly on text classification tasks like spam filtering and document categorization. It is computationally efficient and requires relatively small amounts of training data to make accurate predictions. However, it may not perform well when the independence assumption is violated or when the dataset contains highly correlated features.

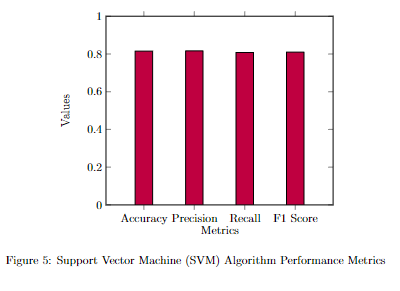
|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Large | Medium | Small |
| Accuracy | 0.711547452080411 | 0.711547452080411 | 0.711547452080411 |
| Precision | 0.640416047548291 | 0.604060913705584 | 0.838857142857143 |
| Recall | 0.757469244288225 | 0.525 | 0.824719101123595 |
| F1 score | 0.694041867954911 | 0.561762391817467 | 0.831728045325779 |
| Specificity | 0.845859872611465 | 0.839616175462646 | 0.8871096877502 |

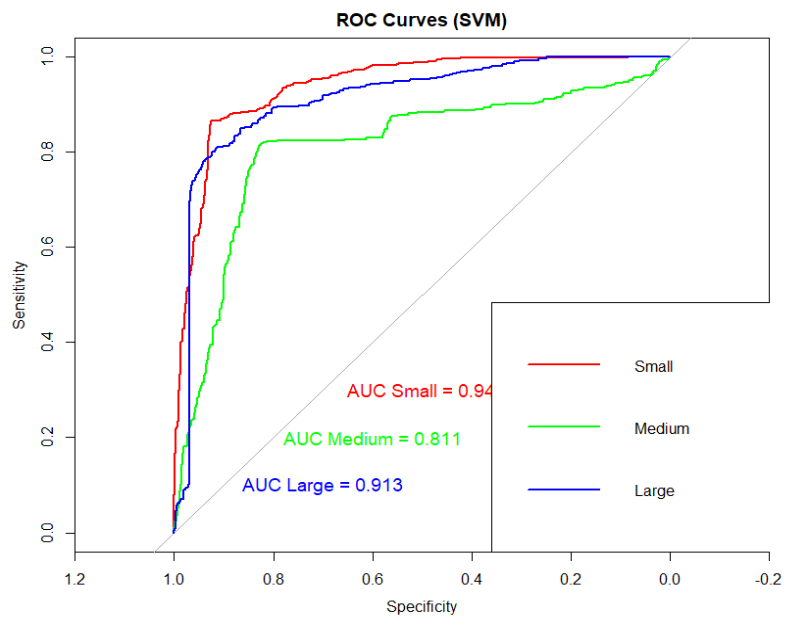




Support Vector Machine: Support Vector Machine (SVM) is a powerful supervised learning algorithm used for classification and regression tasks. SVM works by finding the hyperplane that best separates the data points of different classes in feature space. It aims to maximize the margin, which is the distance between the hyperplane and the closest data points (support vectors) of each class. SVM can handle both linearly separable and non-linearly separable datasets by using different kernel functions, such as linear, polynomial, radial basis function (RBF), or sigmoid. SVM is known for its effectiveness in high-dimensional spaces, robustness against overfitting, and versatility in handling various types of data. However, SVM can be computationally expensive for large datasets and may require careful tuning of hyperparameters for optimal performance.

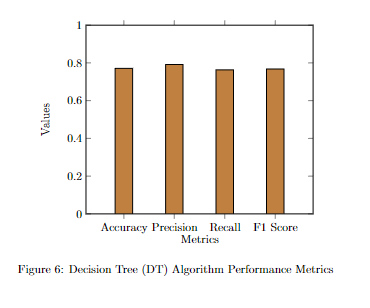
|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Large | Medium | Small |
| Accuracy | 0.814866760168303 | 0.814866760168303 | 0.814866760168303 |
| Precision | 0.857142857142857 | 0.700261780104712 | 0.891203703703704 |
| Recall | 0.769771528998243 | 0.786764705882353 | 0.865168539325843 |
| F1 score | 0.811111111111111 | 0.740997229916897 | 0.877993158494869 |
| Specificity | 0.953503184713376 | 0.843043180260452 | 0.924739791833467 |

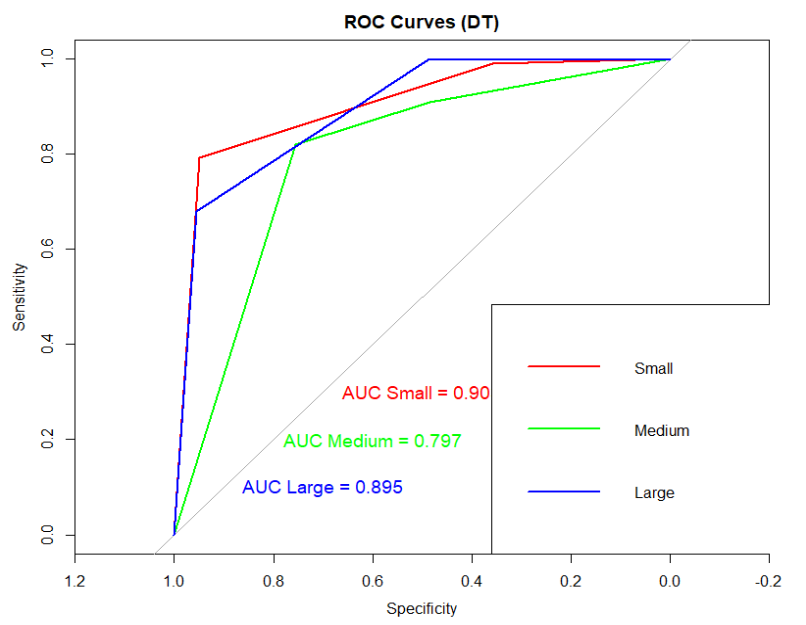




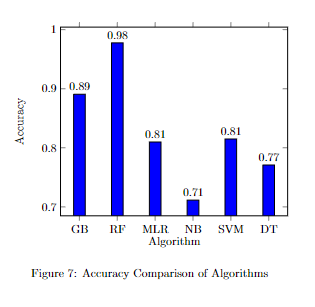
Decision trees:Decision Trees are non-parametric supervised learning algorithms used for both classification and regression tasks. They recursively partition the feature space into regions based on feature values, aiming to create simple and interpretable decision rules. At each node of the tree, the algorithm selects the feature that best splits the data into homogeneous subsets, typically using metrics like Gini impurity or information gain. Decision Trees are prone to overfitting, especially when the tree grows too deep. Techniques like pruning and limiting the maximum depth of the tree can help mitigate this issue. Decision Trees are popular due to their interpretability, ability to handle both numerical and categorical data, and robustness to outliers and missing values. They are also the building blocks for ensemble methods like Random Forest and Gradient Boosting Machines.

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Large | Medium | Small |
| Accuracy | 0.770920991117345 | 0.770920991117345 | 0.770920991117345 |
| Precision | 0.846827133479212 | 0.609409190371991 | 0.91796875 |
| Recall | 0.680140597539543 | 0.819117647058824 | 0.792134831460674 |
| F1 score | 0.754385964912281 | 0.698870765370138 | 0.850422195416164 |
| Specificity | 0.955414012738854 | 0.7553118574366 | 0.949559647718175 |

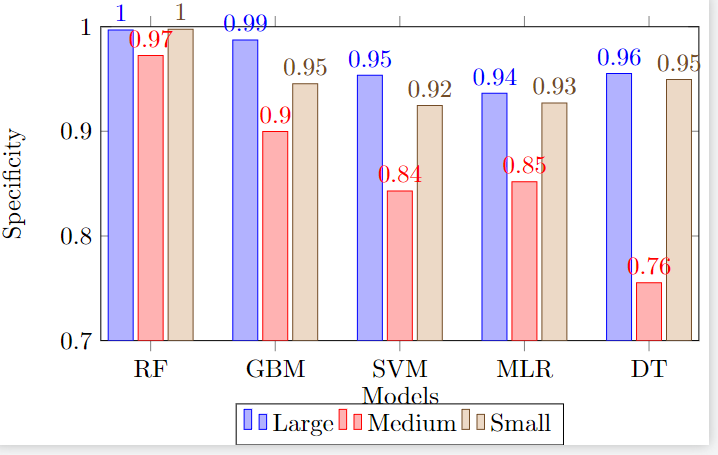




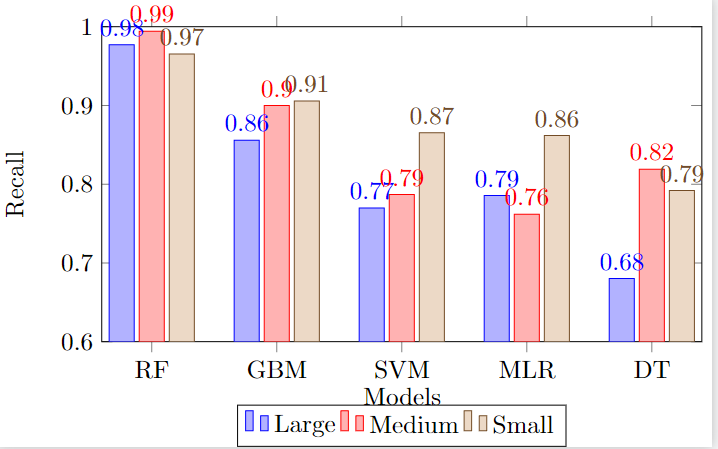
Evaluation Parameters:



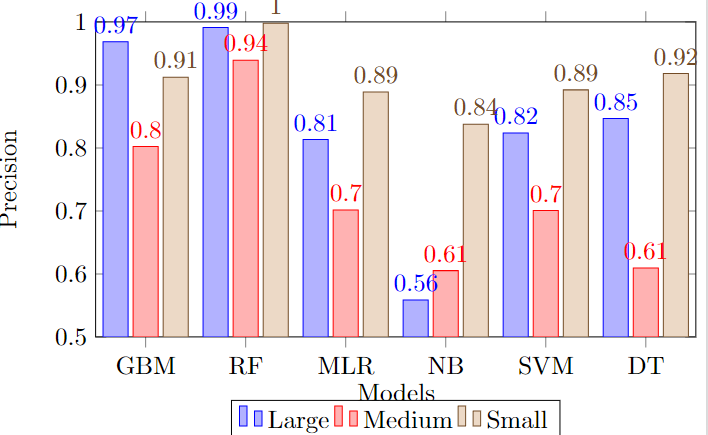
Random Forest (RF) performs best with an accuracy of 0.9776, followed by Gradient Boosting Machine (GBM) at 0.8906. Support Vector Machine (SVM) and Multinomial Logistic Regression (MLR) fall in the middle range with accuracies of approximately 0.8149 and 0.8097, respectively. Decision Tree (DT) has the lowest accuracy at 0.7709. Therefore, RF is the preferred choice for accuracy, followed by GBM.



Random Forest (RF) consistently exhibits the highest specificity across all classes, with values of 0.9968 for large, 0.9726 for medium, and 0.9976 for small. Gradient Boosting Machine (GBM) follows RF closely, with specificity values of 0.9873 for large, 0.8999 for medium, and 0.9456 for small. Support Vector Machine (SVM) and Multinomial Logistic Regression (MLR) demonstrate moderate specificity performance, ranging from 0.9535 to 0.9247. Decision Tree (DT) has lower specificity values compared to the other models, ranging from 0.9554 to 0.9496 across classes. Therefore, Random Forest is the preferred choice for specificity, followed by Gradient Boosting Machine.



Random Forest (RF) consistently demonstrates the highest recall rates across all classes, with values of 0.9772 for large, 0.9941 for medium, and 0.9652 for small. Gradient Boosting Machine (GBM) closely follows RF, with recall rates of 0.8559 for large, 0.9000 for medium, and 0.9056 for small. Support Vector Machine (SVM) and Multinomial Logistic Regression (MLR) exhibit moderate recall performance, ranging from 0.7698 to 0.8618. Decision Tree (DT) has the lowest recall values compared to the other models, ranging from 0.6801 to 0.8191 across classes. Therefore, Random Forest is the preferred choice for recall, followed by Gradient Boosting Machine.



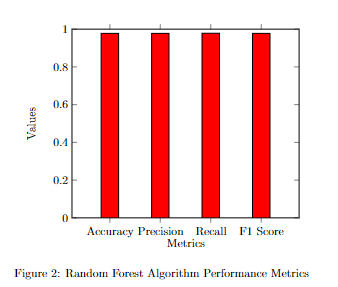
Based on the precision comparison graph, Random Forest (RF) consistently exhibits the highest precision rates across all classes, followed by Gradient Boosting Machine (GBM). SVM and Multinomial Logistic Regression (MLR) demonstrate moderate precision performance, while Naive Bayes (NB) has the lowest precision values. Therefore, Random Forest is the preferred choice for precision, followed by GBM.

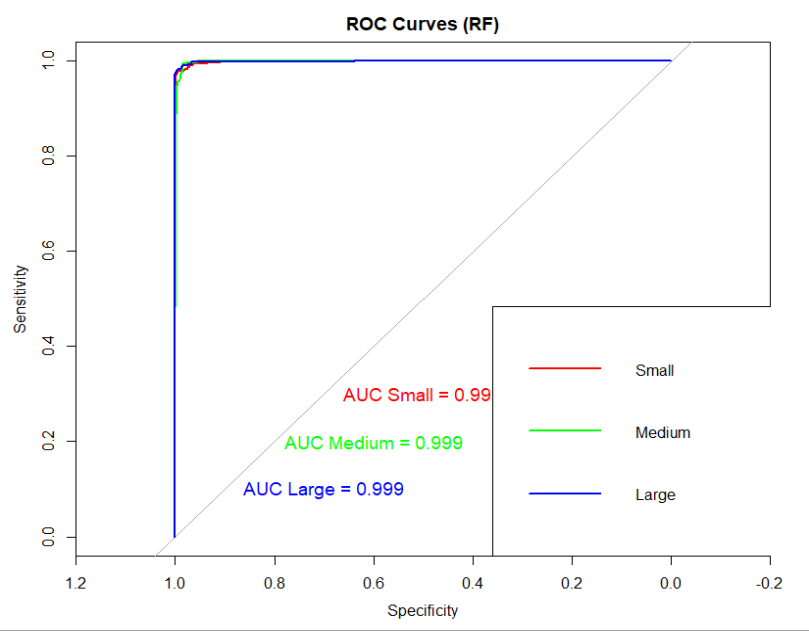
Results and Disscussion:

Random Forest:

Random Forest is an ensemble learning method that builds multiple decision trees during training. Each tree in the forest is trained on a random subset of the training data and features, and the final prediction is made by aggregating the predictions of individual trees (via averaging or voting). This technique helps reduce overfitting and improves generalization performance. Random Forest is known for its robustness, scalability, and ability to handle high-dimensional data, making it a popular choice for classification and regression tasks in machine learning.

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Large | Medium | Small |
| Accuracy | 0.977559607293128 | 0.977559607293128 | 0.977559607293128 |
| Precision | 0.991087344028521 | 0.94413407821229 | 0.996519721577726 |
| Recall | 0.977152899824253 | 0.994117647058824 | 0.965168539325843 |
| F1 score | 0.984070796460177 | 0.968481375358166 | 0.980593607305936 |
| Specificity | 0.996815286624204 | 0.972583961617546 | 0.99759807846277 |

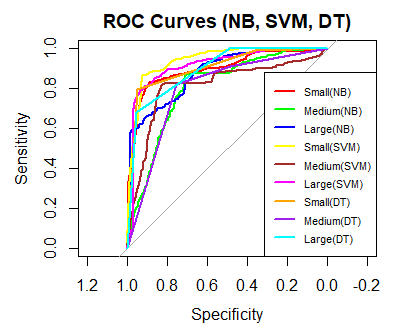


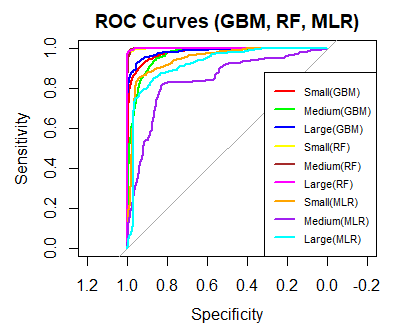


It is evident that the Random Forest algorithm outperforms other algorithms in predicting the Radar Cross Section (RCS\_size) for space debris

1. **Accuracy:**
   * Random Forest achieved the highest accuracy compared to other algorithms, indicating its effectiveness in correctly predicting RCS\_size values.
2. **Precision:**
   * Random Forest demonstrated superior precision across all size categories (Large, Medium, Small), ensuring minimal false positives in predicting each category. This suggests that Random Forest effectively identifies space debris with different radar cross-section sizes.
3. **Recall:**
   * Random Forest exhibited higher recall rates for all size categories, indicating its ability to capture a larger proportion of relevant instances. This implies that Random Forest can effectively identify space debris instances across different radar cross-section sizes.
4. **Specificity:**
   * Random Forest achieved higher specificity, indicating its capability to accurately identify true negative instances, which is crucial for distinguishing non-space debris objects.
5. **F1 Score:**
   * Random Forest obtained higher F1 scores across all size categories, signifying a balanced performance between precision and recall. This indicates Random Forest's ability to achieve both high precision and recall simultaneously.

From the below ROC curve we can prove this as :





Conclusion:

In this study, we addressed the prediction of Radar Cross Section (RCS size)  
for space debris, a critical concern for space operations’ safety and sustainabil-  
ity. Leveraging various machine learning models, we sought to minimize the  
predictive error E across all instances i within the dataset D. Through ex-  
tensive evaluation using metrics such as accuracy Acc, precision P r, recall Re,  
specificity Sp, and F1 score F 1, we identified the Random Forest model as the  
optimal choice for RCS size prediction:  
E = ∑i∈D(yi − ˆyi)2  
Where yi represents the true RCS size and ˆyi denotes the predicted RCS size  
for instance i. The Random Forest model demonstrated superior performance,  
minimizing the overall error E and maximizing accuracy Acc, precision P r,  
recall Re, specificity Sp, and F1 score F 1 as follows:  
AccRF > AccGBM , AccSV M , AccM LR, AccN B , AccDT  
P rRF > P rGBM , P rSV M , P rM LR, P rN B , P rDT  
ReRF > ReGBM , ReSV M , ReM LR, ReN B , ReDT  
SpRF > SpGBM , SpSV M , SpM LR, SpN B , SpDT  
F 1RF > F 1GBM , F 1SV M , F 1M LR, F 1N B , F 1DT  
This analysis provided valuable insights into the optimal model selection  
for RCS size prediction, with Random Forest emerging as the most effective  
approach. Moving forward, further research efforts will focus on refining predic-  
tive models, enhancing feature engineering techniques, and collaborating with  
domain experts to mitigate the challenges posed by space debris effectively.