*Performance Comparison of Classification algorithms on Nearest Earth Objects Dataset*

## Abstract

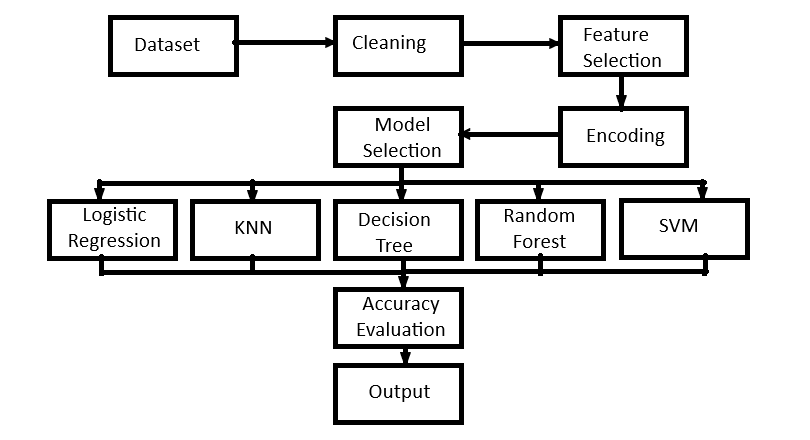
This study examines the speed and accuracy performance of five classification algorithms using a nearest earth objects dataset sourced by NASA. The goal is to determine which algorithm is more effective for classification of the dataset and making predictions. The study focuses on factors such as relative velocity, magnitude of the object, diameter of the object, miss distance from collision etc which influence that the object is hazardous or not. The findings will contribute to better understanding the suitability of different algorithms for this particular task and dataset.

**Keywords**— Logistic Regression, KNN , Decision Tree, Random Forest, SVM

# **Introduction**

In the realm of celestial object detection and hazard assessment, performing classification on datasets featuring parameters like estimated diameter, relative velocity, miss distance, orbiting body, and hazard status is essential. Classification algorithms offer insights that enable the identification of potentially hazardous near-Earth objects (NEOs), aiding space agencies, astronomers, and policymakers in prioritizing monitoring efforts and mitigating threats. Moreover, classification facilitates pattern recognition, trend analysis, and predictive modeling, enhancing our understanding of celestial phenomena and enabling proactive measures to safeguard our planet. By categorizing celestial objects based on their characteristics, such as estimated diameter and absolute magnitude, classification enables the identification of patterns and trends within the dataset. This not only aids scientific research and exploration but also informs risk assessment strategies and planetary defense initiatives. In summary, classification on such datasets is crucial for extracting meaningful information, informing decision-making processes, and advancing our capabilities in understanding and mitigating potential cosmic hazards that could affect Earth and its inhabitants.

**Proposed Methodology**



**Experiment**

To determine the most suitable classifier for the dataset, the experiment employed five different models: Logistic regression, KNN, Decision Tree , Random forest , SVM. The goal was to evaluate their performance in classifying the pre-processed data, training the models, conducting testing, and making predictions using the trained models. The detailed procedure of the experimentation is as follows:

**DATASET DESCRIPTION**

This dataset comprises information pertaining to various celestial objects, primarily near-Earth objects (NEOs), and their associated characteristics relevant to hazard assessment and planetary defense. It encompasses parameters essential for classification and risk analysis, including estimated diameter, relative velocity, miss distance, orbiting body, sentry object status, absolute magnitude, and hazardous classification

**Data Cleaning**

Normalisation of data is important part of data analysis and classification.

Here the nominal data attributes is converted into numeric attributes.

**Data Splitting**

Here splitting the data into 2 halves of different sizes and is used for training and testing in different data mining classification algorithm.

Here 75 - 25 split is used.

**Data Scaling**

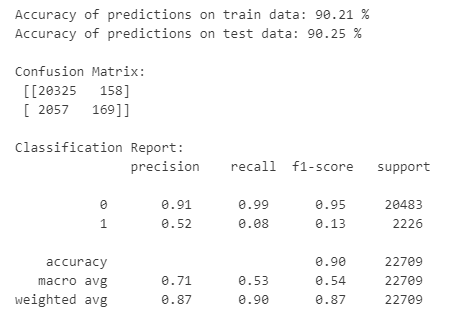
It involves transforming feature values to a common scale to ensure fair comparison and effective model training. Scaling prevents features with larger magnitudes from dominating those with smaller ones, thus improving algorithm convergence and performance. Common scaling techniques include Min-Max scaling, where values are mapped to a predefined range, and Standardization, which centers the data around a mean of zero and a standard deviation of one.

## Classification

After scaling of data, it is fitted in different models and then observing the accuracy for the same.

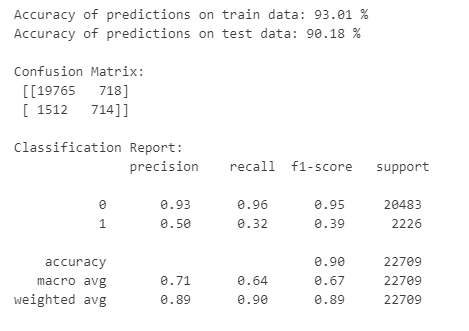
**Logistic Regression**

Logistic Regression is a fundamental supervised learning algorithm used for binary classification tasks, predicting the probability of an event occurrence based on input features. Despite its name, it's primarily employed for classification rather than regression. The algorithm models the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic function. It operates on linear combinations of input features, transformed via the logistic (sigmoid) function to constrain predictions between 0 and 1. Parameters are optimized through techniques like maximum likelihood estimation or gradient descent. Logistic Regression is interpretable, providing insights into feature importance through coefficients. While simple and efficient, it assumes a linear relationship between features and the log-odds of the outcome, limiting its applicability in complex, nonlinear scenarios often addressed by more sophisticated models like decision trees or neural network



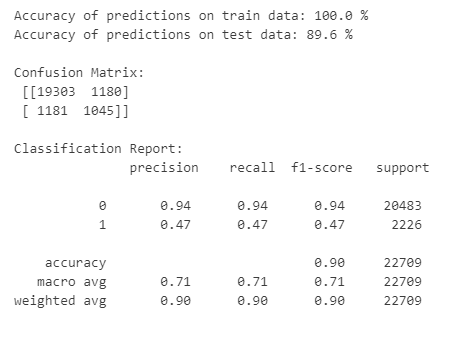
**KNN**

K-Nearest Neighbours (KNN) is a versatile algorithm used for both classification and regression tasks. It operates on the principle of proximity, predicting the label or value of a new data point based on the majority class or average value among its K nearest neighbours in the feature space, determined by a distance metric such as Euclidean or Manhattan distance. While KNN is intuitive and easy to implement, its performance may degrade with high-dimensional data or imbalanced class distributions due to the curse of dimensionality and bias towards the majority class, respectively. To address these challenges, regularization techniques and feature scaling can be applied. Despite its limitations, KNN remains a valuable tool in machine learning, particularly in scenarios where interpretability and simplicity are prioritized, and where dataset sizes are small to moderate



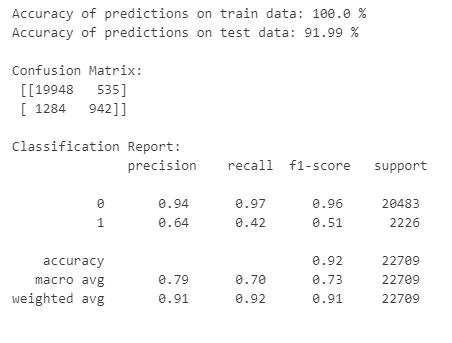
**Decision Tree**

Decision trees are a fundamental tool in machine learning and data analysis, providing a structured approach to decision-making. Decision trees operate by recursively partitioning the input space into smaller regions based on feature values, with the goal of maximizing homogeneity within each partition regarding the target variable. At each step, the algorithm selects the feature that best splits the data into subsets that are as pure as possible, typically measured by metrics like information gain or Gini impurity. This process continues until a stopping criterion is met, such as reaching a maximum depth or when further splitting does not significantly improve purity. Each leaf node then represents a class label or a decision, determined by the majority class of the instances within that partition. During the construction phase, decision trees may overfit the training data, capturing noise and outliers. To address this, techniques like pruning or setting minimum sample requirements per leaf node can be employed to generalize the model. Despite their susceptibility to overfitting, decision trees excel in interpretability and can handle both numerical and categorical data, making them valuable tools in various domains, from healthcare to finance and beyond.



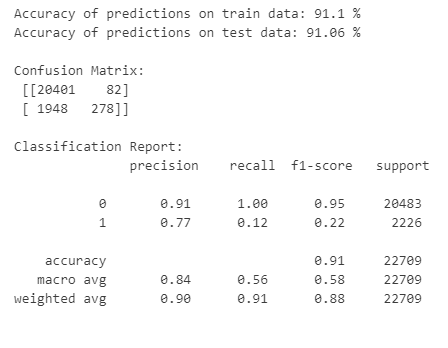
**Random Forest**

Random Forest is an ensemble learning method that operates by constructing multiple decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees. Each tree in the forest is built using a random subset of the training data and a random subset of the features, which introduces diversity among the trees. This diversity helps to reduce overfitting and improve the generalization performance of the model. During the prediction phase, each tree in the forest independently classifies the input data, and the final prediction is determined by aggregating the results from all trees, often through a majority voting scheme for classification or averaging for regression. Random Forests are highly robust to noisy data and outliers and can handle high-dimensional datasets with ease. Additionally, they provide estimates of feature importance, allowing users to identify the most influential features in the prediction process. Random Forests are computationally efficient and can handle large datasets with high dimensionality. These properties make Random Forests a popular choice for various machine learning tasks, including classification, regression, and feature selection.



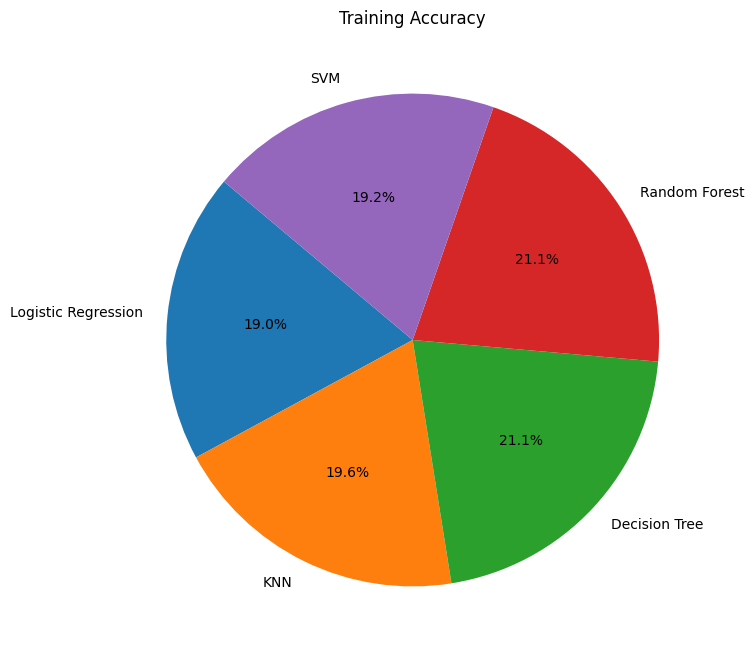
**SVM**

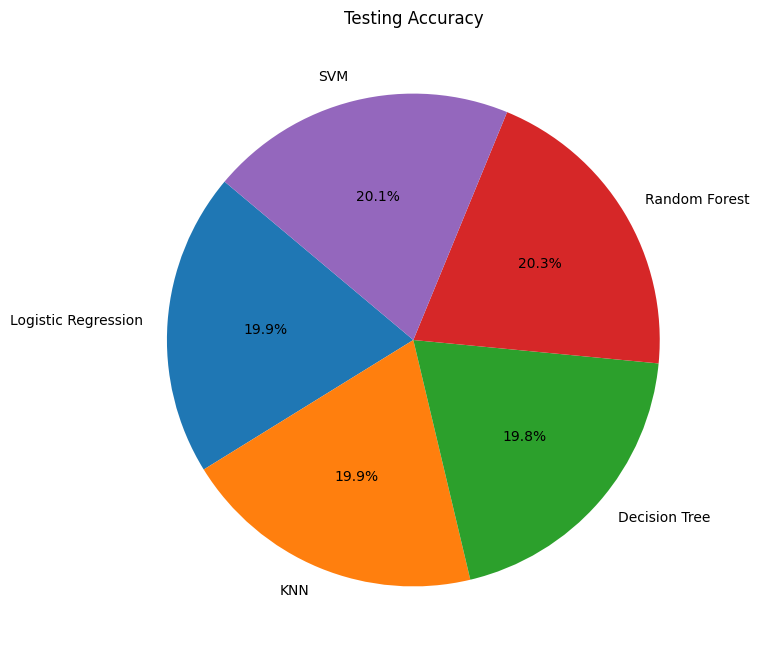
Support Vector Machine (SVM) is a powerful supervised learning algorithm primarily used for classification tasks, though it can be adapted for regression as well. SVM aims to find the hyperplane that best separates the data points into different classes while maximizing the margin between the classes. The hyperplane is determined by support vectors, which are data points lying closest to the decision boundary. SVM seeks to maximize the margin, which is the distance between the support vectors and the hyperplane. This margin optimization leads to a robust classifier that generalizes well to unseen data. SVM is particularly effective in high-dimensional spaces and can handle datasets with complex relationships. Moreover, SVM can incorporate the kernel trick, allowing it to handle nonlinear decision boundaries by transforming the input space into a higher-dimensional feature space. Despite its effectiveness, SVM can be sensitive to the choice of parameters, such as the regularization parameter and the choice of kernel function. Nevertheless, SVM remains a popular choice in various domains, including text classification, image recognition, and bioinformatics.



**Results**

From the above experiment it shows that the random forest classification is the best data mining technique for the given car data set. The following is the graphical representation for the comparison of the accuracy of the different models.





­­

**Conclusion and future discussion**

In conclusion, by testing all the given models we have conclude from the above given model random forest is best and our objective was to determine the most suitable algorithm for classifying and making predictions on the dataset.

But overall, the choice of the most suitable algorithm depends on the specific requirements of the task and the available computational resources. With this experiment, Researchers and practitioners can use the findings of this study as a basis for selecting the appropriate data mining algorithm when working with car evaluation datasets or similar domains.

Further research could explore the performance of other algorithms, such as perceptron classifier or other ensemble methods and investigate the impact of different feature selection techniques or data preprocessing methods on the results. This would provide a more comprehensive understanding of the best approaches for analysing nearest earth objects data and enable informed decision making in the space industries.

**Reference**

* <https://en.wikipedia.org/wiki/Logistic_regression>
* <https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>
* https://en.wikipedia.org/wiki/Decision \_tree
* <https://en.wikipedia.org/wiki/Random_forest>
* <https://en.wikipedia.org/wiki/Support_vector_machine>
* Ethem Alpaydin. 2020. Introduction to machine learning. The MIT Press.

* Ayşe Eldem. 2018. Comparison of Prediction Success Performances for Classification Methods.

International Journal of Engineering Science Invention, 7(12), Ver. II: 63-65.

* Breiman. L. 1996. Bias, Variance, and Arcing Classifiers. Technical Report 460, Department of Statistics, University of California, Berkeley, CA.
* Breiman, L. 1996a. Bagging predictors. Machine Learning, 24(2):123–140. [4] Callan, R. 1998. Essence of neural networks. Prentice Hall PTR Upper Saddle River, NJ, USA.
* Cortes, C. and Vapnik, V. 1995. Support vector networks, Machine learning, 20(3):273-297. • Govindarajan, M. and Mishra, A. 2014a. Development of a hybrid ensemble approach for Automobile data. International Journal of Engineering Sciences & Research Technology, 3(12):387-392.
* Hua, R. and Dai liankui. 2010. support vector machine classification and regression based hybrid modeling method and its application in raman spectral analysis. Chinese Journal of Scientific Instrument, (11):2440-2446.
* Jiawei Han, Micheline Kamber. 2003. Data Mining – Concepts and Techniques, Elsevier Publications.
* Pelin YJldJrJm, UlaG K. Birant, and Derya Birant. 2019. EBOC: Ensemble-Based Ordinal Classification in Transportation. Journal of Advanced Transportation, Volume 2019:1-17.
* Tom M. Mitchell. 1997. Machine Learning, McGraw-Hill, New York. • Uma R. Salunkhea, Suresh N. Mali. 2016. Classifier Ensemble Design for Imbalanced Data Classification:

A Hybrid Approach. Procedia Computer Science, 85:725 – 732.

* Vapnik, V. Golowich, S. and Smola, A. 1997. Support vector method for function approximation, regression estimation, and signal processing. Advances in neural information processing systems, 281287.