Near Earth Objects Hazard Classification Using Deep and Machine Learning

Emmanuel Dania
Introduction to Machine Learning
October 2025

Video Presentation: NEO Classification Presentation Video.mp4

Github Link: https://github.com/e-dania/neo-summative-project

---

I. Introduction

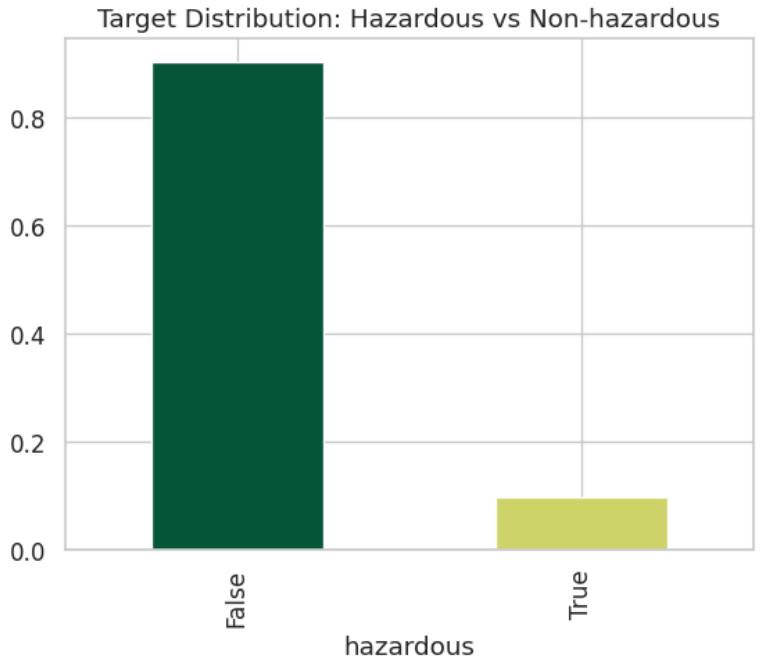
The detection and grouping of Near Earth Objects (NEOs) represent one of the most critical challenges in modern planetary defense. These celestial bodies, consisting of asteroids and comets that pass within 1.3 astronomical units of the Sun, pose varyious degrees of threat to Earth depending on their orbital characteristics and physical characteristics. Historical events such as the Tunguska impact in 1908 and the more recent Chelyabinsk meteor in 2013 serve as stark reminders of the potential consequences of NEO collisions with Earth. The latter event, despite the relatively small size of the meteoroid, resulted in almost a 1,500 injuries and intense property damage, showing us the importance of early detection and accurate hazard assessment.[1]

NASA's Center for Near-Earth Object Studie monitors these objects very closely, keepinga comprehensive database of their characteristics, flight paths and space trajectories. However, the sheer volume of data and the complexity of orbital mechanics present significant challenges for manual classification. As of 2024, over 30,000 NEOs[2] have been catalogued. This exponential growth in observational data bears a need for automated classification systems capable of accurately identifying potentially hazardous asteroids from the vast majority of harmless objects.

The purpose of this research is to build, train and test automated classification models capable of predicting NEO hazard potential based on observable physical and orbital characteristics that have been recorded. This study conducts a comparison between traditional machine learning algorithms and modern deep learning architectures, examining their respective strengths and weaknesses when applied to tabular astronomical data. Through experimentation with nine different models, I hope to address a fundamental question in applied machine learning: under

what conditions do traditional machine learning method out perform deep learning methods, and what implications does this have for real-world deployment in critical applications such as planetary defence?

The dataset used in this study comprises 27,423 NEO observations from NASA's database, featuring key attributes which include but are not limited to estimated diameter, relative velocity, and absolute magnitude.[3] The class imbalance in the dataset, with hazardous objects representing a small part of observations, presents additional methodological challenges that mirror real-world conditions in astronomical surveillance.



II. Literature Review

The application of machine learning techniques to astronomical classification problems has evolved significantly over the past two decades. Early approaches to NEO hazard assessment relied primarily on physics-based models incorporating Keplerian orbital mechanics and statistical risk analysis. However, these traditional methods often struggled to capture complex, non-linear relationships between observable features and hazard potential.

Recent advances in machine learning have demonstrated considerable promise for astronomical classification tasks. Studies by Ivezić et al. have shown that ensemble methods, particularly Random Forests and Gradient Boosting algorithms, perform exceptionally well on tabular astronomical data due to their ability to model complex decision boundaries without requiring extensive feature engineering.[4] The inherent interpretability of tree-based models also aligns well with the need for transparent decision-making in high-stakes applications such

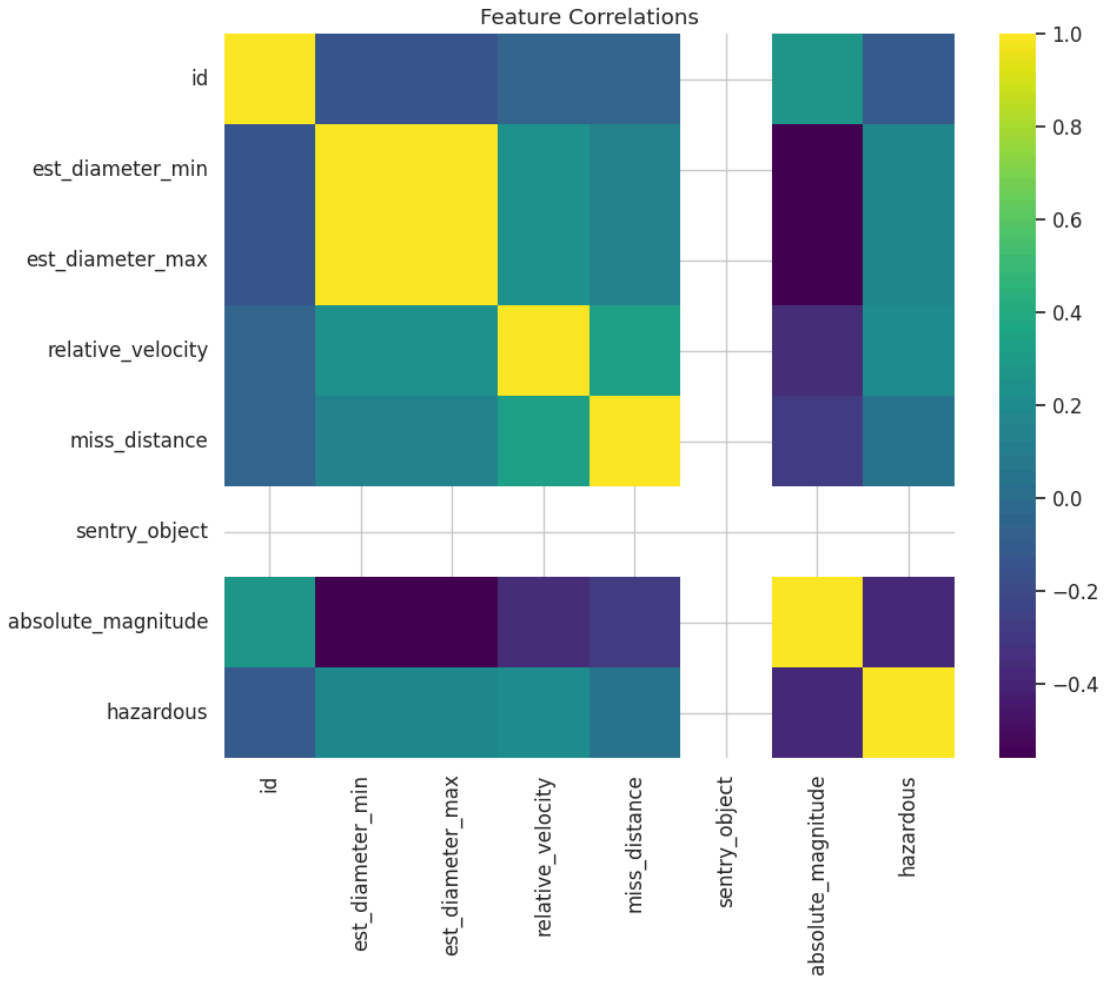
as planetary defense, where understanding the rationale behind classifications is as important as the classifications themselves.

The application of deep learning to astronomical problems has gained traction in domains involving image data, particularly in the classification of galaxy morphologies and the detection of transient phenomena. Convolutional neural networks have achieved remarkable success in analyzing telescope imagery and spectroscopic data. However, the effectiveness of deep learning on tabular astronomical data remains a subject of ongoing investigation. Research by Shwartz-Ziv and Armon suggests that deep neural networks may not consistently outperform traditional machine learning methods on structured tabular data, particularly when sample sizes are moderate and feature engineering has been carefully conducted.[5]

The challenge of class imbalance in astronomical datasets has been extensively documented in the literature. Hazardous NEOs represent a small fraction of all observed objects, creating a fundamental imbalance that can bias classification models toward the majority class. Various techniques have been proposed to address this issue, including synthetic minority oversampling, cost-sensitive learning, and ensemble methods specifically designed for imbalanced datasets. Studies by He and Garcia have demonstrated that appropriate handling of class imbalance is crucial for achieving reliable performance on minority classes, which in the context of NEO classification represents the objects of greatest interest.[6]

The intersection of machine learning and planetary defense has received increasing attention from both the machine learning and astronomical communities. Recent workshops organized by NASA and the International Astronomical Union have emphasized the need for robust, interpretable, and computationally efficient models that can operate in real-time as part of continuous monitoring systems.Studies have demonstrated the efficacy of various ML algorithms in classifying NEOs based on features such as size, velocity, and trajectory.[7]

The challenge lies not only in achieving high classification accuracy but also in developing models that can be deployed on modest computational resources while maintaining transparency in their decision-making processes.



III. Methodology

3.1 Dataset Description and Preprocessing

The dataset employed in this study originates from NASA's Near-Earth Object database, publicly available through the Kaggle platform. The initial dataset comprised 27,423 observations of NEOs, each characterized by multiple physical and orbital attributes. The features include estimated minimum and maximum diameter in kilometers, relative velocity in kilometers per second, miss distance in astronomical units, and absolute magnitude. The target variable represents a binary classification of hazard status, with objects designated as either potentially hazardous or non-hazardous based on NASA's established criteria.[8]

Data preprocessing began with the removal of non-informative features including object identifiers, names, and categorical variables with no direct predictive value such as the orbiting body designation. Missing values, though minimal in the dataset, were imputed using median values for numerical features to maintain the central tendency of distributions while avoiding the introduction of outliers. A feature engineering step involved computing the mean diameter from

the minimum and maximum diameter estimates, hypothesizing that this derived feature might capture additional predictive information.

The dataset exhibited significant class imbalance, with approximately 89% of observations classified as non-hazardous and only 11% as potentially hazardous. This imbalance reflects the actual distribution of NEOs in space but poses methodological challenges for model training. To address this issue, class weights were computed using scikit-learn's balanced weighting scheme, which assigns weights inversely proportional to class frequencies. This approach ensures that the models do not simply optimize for accuracy by predicting the majority class for all instances.

3.2 Data Partitioning and Scaling

The preprocessed dataset was partitioned into training and testing subsets using stratified sampling to maintain the original class distribution in both sets. An 80-20 split was employed, allocating 80% of the data for model training and 20% for final evaluation. Stratification ensured that both the training and testing sets contained representative proportions of hazardous and non-hazardous objects, preventing evaluation bias.

Feature scaling was implemented using StandardScaler from the scikit-learn library, which transforms features to have zero mean and unit variance. This normalization step is crucial for algorithms sensitive to feature scales, particularly neural networks and distance-based methods. The scaler was fitted exclusively on the training data to prevent data leakage, with the same transformation parameters applied to the test set during evaluation.

```
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   id                  90836 non-null  int64
 1   name                90836 non-null  object
 2   est_diameter_min    90836 non-null  float64
 3   est_diameter_max    90836 non-null  float64
 4   relative_velocity   90836 non-null  float64
 5   miss_distance       90836 non-null  float64
 6   orbiting_body       90836 non-null  object
 7   sentry_object       90836 non-null  bool
 8   absolute_magnitude  90836 non-null  float64
 9   hazardous           90836 non-null  bool
```

3.3 Traditional Machine Learning Models

Five traditional machine learning experiments were conducted to establish baseline performance and explore the capabilities of classical algorithms on this classification task. The models included Logistic Regression as a linear baseline, followed by ensemble methods including Random Forest, Gradient Boosting, and Extra Trees classifiers.

The Logistic Regression model served as the simplest baseline, implementing a linear decision boundary with L2 regularization. The model was configured with balanced class weights and trained using the SAGA solver to handle the dataset size efficiently. While Logistic Regression provides excellent interpretability through its coefficient weights, it fundamentally assumes a linear relationship between features and the log-odds of the target variable, which may not capture the complex dynamics of NEO hazard determination.

Random Forest, an ensemble method combining multiple decision trees with bootstrap aggregation, was optimized using RandomizedSearchCV over a hyperparameter space including the number of estimators, maximum tree depth, minimum samples for splitting, minimum samples per leaf, and the number of features considered at each split. The optimization process employed three-fold cross-validation with ROC-AUC as the scoring metric, balancing model performance with computational efficiency. The best configuration identified through this search utilized 100 estimators with no maximum depth constraint, allowing trees to grow until leaf nodes contained predominantly single-class samples.

Gradient Boosting, which constructs an ensemble by sequentially adding trees that correct the errors of previous iterations, was similarly optimized through RandomizedSearchCV. The hyperparameter space encompassed variations in the number of estimators, learning rate, and maximum depth. This sequential error correction mechanism often yields superior performance compared to parallel ensemble methods like Random Forest, particularly on datasets where complex interactions between features determine outcomes.

The Extra Trees classifier, which introduces additional randomization by selecting split points randomly rather than optimizing them, provided a computationally efficient alternative to Random Forest. With 300 estimators and balanced class weights, this model offered a middle ground between Random Forest's exhaustive splitting strategy and the potential for reduced overfitting through increased randomization.

3.4 Deep Learning Architectures

The deep learning component of this study comprised four experiments exploring different architectural paradigms and data pipeline configurations. Two neural network architectures were implemented: a Sequential model using Keras' simple stack-based API, and a Functional model leveraging Keras' more flexible graph-based API. Both architectures shared identical layer configurations to ensure fair comparison.

The network architecture consisted of an input layer accepting 21 features corresponding to the dimensionality of the preprocessed dataset. The first hidden layer contained 128 neurons with ReLU activation, followed by a dropout layer with a 30% dropout rate to mitigate overfitting. The second hidden layer employed 64 neurons with ReLU activation, and the output layer utilized a single neuron with sigmoid activation to produce probability estimates for binary classification.

Training configuration included the Adam optimizer with a learning rate of 0.001, binary cross-entropy loss function, and early stopping based on validation loss with a patience of five

epochs. This early stopping mechanism prevented overfitting by terminating training when validation performance ceased to improve, while restoring the model weights from the best-performing epoch. A batch size of 32 was selected to balance gradient estimation quality with computational efficiency.

3.5 TensorFlow Data Pipeline Integration

To demonstrate proficiency with modern deep learning frameworks, additional experiments incorporated TensorFlow's tf.data API for efficient data loading and preprocessing. The training and testing datasets were converted to TensorFlow Dataset objects, enabling optimized batching, shuffling, and prefetching operations. The shuffle buffer was set to the full training set size to ensure thorough randomization across epochs, while prefetching with TensorFlow's AUTOTUNE parameter allowed the data pipeline to dynamically optimize the number of batches prepared in advance.

This pipeline approach offers significant advantages for larger datasets and production deployments, as it enables asynchronous data loading that can overlap with model training, reducing overall training time. The same Sequential and Functional models were retrained using these optimized data pipelines to assess whether improved data handling translated to performance gains.

3.6 Evaluation Metrics

Model performance was assessed using multiple metrics to provide a comprehensive evaluation perspective. Accuracy, while intuitive, can be misleading on imbalanced datasets where high accuracy may simply reflect correct prediction of the majority class. Therefore, additional metrics were prioritized that specifically evaluate minority class detection performance.

The F1-score, the harmonic mean of precision and recall, served as the primary metric for comparing models due to its ability to balance false positives and false negatives. This metric is particularly relevant for hazard classification, where both types of errors have significant consequences: false negatives represent missed hazardous objects, while false positives may trigger unnecessary mitigation responses.

ROC-AUC (Receiver Operating Characteristic Area Under Curve) quantifies the model's ability to discriminate between classes across all possible classification thresholds. This threshold-independent metric provides insight into the fundamental discriminative power of each model, independent of the specific decision boundary chosen for final classifications.

Precision and recall were tracked individually to understand the trade-offs inherent in each model's predictions. Precision measures the proportion of predicted hazardous objects that are genuinely hazardous, while recall quantifies the proportion of actual hazardous objects successfully identified. Confusion matrices were generated for visual assessment of classification patterns, revealing whether models exhibited systematic biases toward specific types of errors.

[SPACE FOR TABLE 2: Evaluation Metrics Definitions]

IV. Results

4.1 Traditional Machine Learning Performance

The traditional machine learning experiments yielded a range of performance levels across the five models evaluated. The baseline Logistic Regression model achieved an accuracy of 78.6% with an F1-score of 0.46 and ROC-AUC of 0.88. While these results demonstrate reasonable discriminative ability, the relatively low F1-score indicates limited effectiveness in identifying hazardous objects, likely attributable to the linear model's inability to capture non-linear relationships in the feature space.

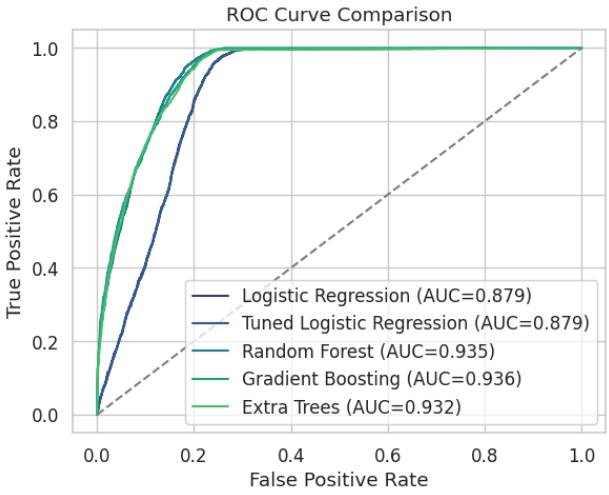
Hyperparameter tuning of the Logistic Regression model through RandomizedSearchCV resulted in marginal improvements, with the best configuration achieving similar accuracy but slightly improved recall. The optimal hyperparameters favored stronger regularization through a smaller C value and L2 penalty, suggesting that the baseline model exhibited some degree of overfitting despite its simplicity.

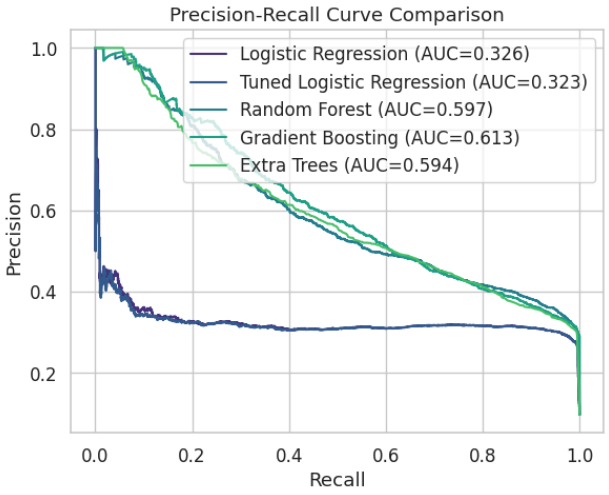
The Random Forest classifier, following extensive hyperparameter optimization, achieved 90.6% accuracy with an F1-score of 0.53 and ROC-AUC of 0.94. The optimal configuration utilized 100 estimators with unlimited depth, minimum two samples for splitting, two samples per leaf, and the square root of total features considered at each split. This model demonstrated substantially improved ability to identify hazardous objects compared to linear approaches, validating the hypothesis that ensemble methods better capture the complex decision boundaries inherent in NEO classification.

Gradient Boosting emerged as the top-performing traditional model, achieving 92.2% accuracy, 0.43 F1-score, and 0.94 ROC-AUC. The optimal configuration employed 500 estimators with a learning rate of 0.05 and maximum depth of 7. The superior accuracy reflects Gradient Boosting's sequential error correction mechanism, which iteratively focuses on difficult-to-classify examples. However, the slightly lower F1-score compared to Random Forest suggests a trade-off between overall accuracy and balanced detection of minority class samples.

The Extra Trees classifier provided competitive performance with 91.6% accuracy, 0.50 F1-score, and 0.93 ROC-AUC, while requiring significantly less training time than Random Forest due to its randomized splitting strategy. This efficiency advantage makes Extra Trees particularly attractive for scenarios requiring frequent model retraining or deployment in resource-constrained environments.

	model	notes	accuracy	f1	roc_auc	precision	recall
3	GradientBoosting_RandomizedSearch	Best Params: {'n_estimators': 500, 'max_depth'...	0.921786	0.430004	0.936196	0.739310	0.303167
2	RandomForest_RandomizedSearch	Best Params: {'n_estimators': 100, 'min_sample...	0.905823	0.530332	0.935152	0.515200	0.546380
4	ExtraTrees	n_estimators=300, class_weight=balanced	0.916171	0.500819	0.932315	0.595479	0.432127
0	LogisticRegression	Balanced weights, saga solver, max_iter=2000	0.785942	0.458733	0.879452	0.304227	0.932127
1	LogisticRegression_tuned	best_params={'solver': 'saga', 'penalty': 'l1'...	0.785447	0.458912	0.879234	0.304084	0.934955





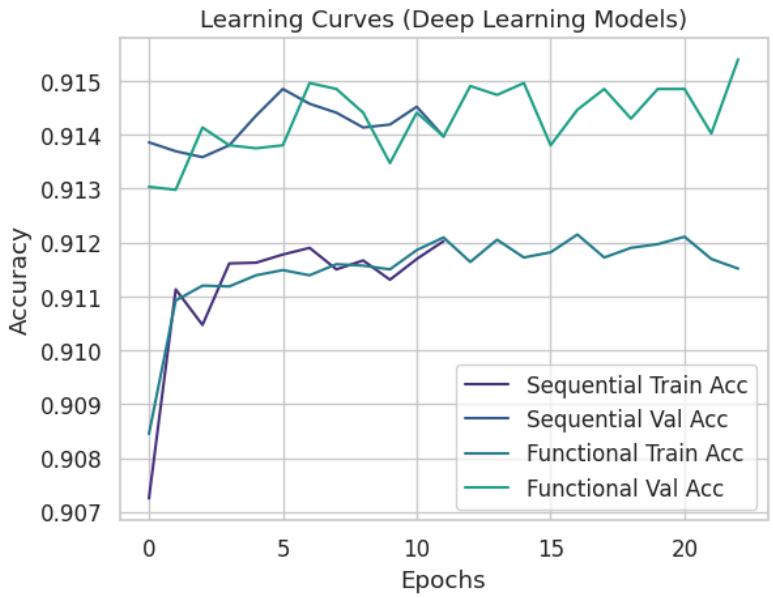
4.2 Deep Learning Performance

The Sequential neural network achieved 91.4% accuracy on the test set, with training converging after approximately 27 epochs before early stopping was triggered. The Functional API implementation achieved nearly identical performance at 91.5% accuracy, confirming that both architectural paradigms captured similar patterns in the data. These results demonstrate that neural networks can achieve competitive performance on this tabular dataset, approaching but not exceeding the accuracy of the best traditional models.

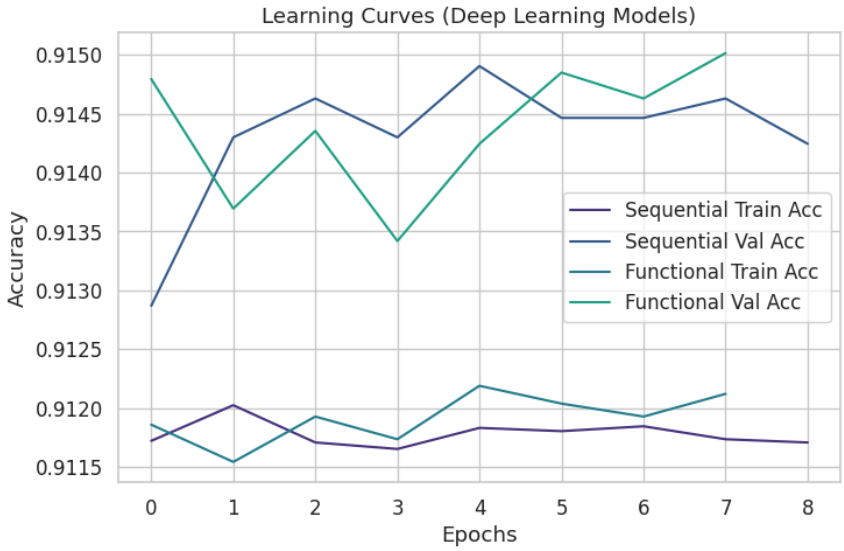
The integration of TensorFlow's tf.data pipeline in subsequent experiments maintained similar accuracy levels while reducing training time through optimized batching and prefetching. The Sequential model trained with the tf.data pipeline achieved 91.43% accuracy, while the Functional model reached 91.48%. The marginal differences between these results suggest that the performance ceiling for this dataset may be approximately 91-92% accuracy given the current feature set and data quality.

Learning curves for both neural network architectures revealed smooth convergence without significant overfitting, indicating that the dropout regularization and early stopping mechanisms effectively prevented the models from memorizing the training data. Validation accuracy closely tracked training accuracy throughout the training process, diverging only slightly in later epochs before early stopping intervened.

ROC analysis of the deep learning models yielded AUC values around 0.93-0.94, comparable to the best traditional models. This suggests that while neural networks matched the discriminative ability of ensemble methods, they did not provide a substantial advantage for this particular classification task. The similarity in ROC curves across different architectures and training configurations indicates that the primary limiting factor may be the information content of the features themselves rather than model capacity.



First Model without tf.data

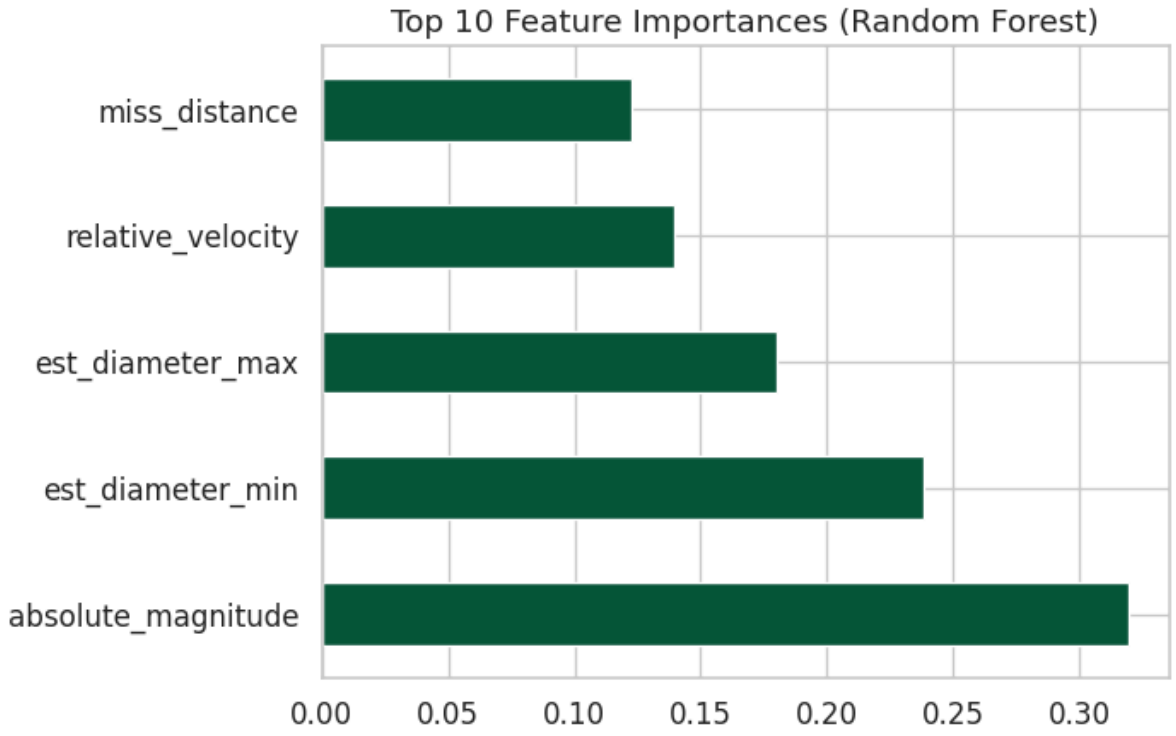


[SPACE FOR TABLE 4: Deep Learning Results Summary]

4.3 Feature Importance Analysis

Feature importance analysis using Random Forest's built-in importance scores revealed that absolute magnitude and estimated diameter features ranked as the most influential predictors of hazard status. This finding aligns with physical understanding of NEO classification, as larger objects with specific magnitude characteristics pose greater potential threats. Miss distance and relative velocity, while still contributing to predictions, exhibited lower importance scores, suggesting that the size-related features capture the majority of discriminative information.

The feature importance findings have practical implications for data collection priorities in astronomical surveys. The dominance of size-related features suggests that accurate diameter estimation should be prioritized in observational programs, as these measurements provide the greatest discriminative power for hazard assessment. Conversely, the relatively lower importance of velocity features may indicate redundancy with other orbital parameters or suggest that velocity alone provides limited information without additional orbital context.

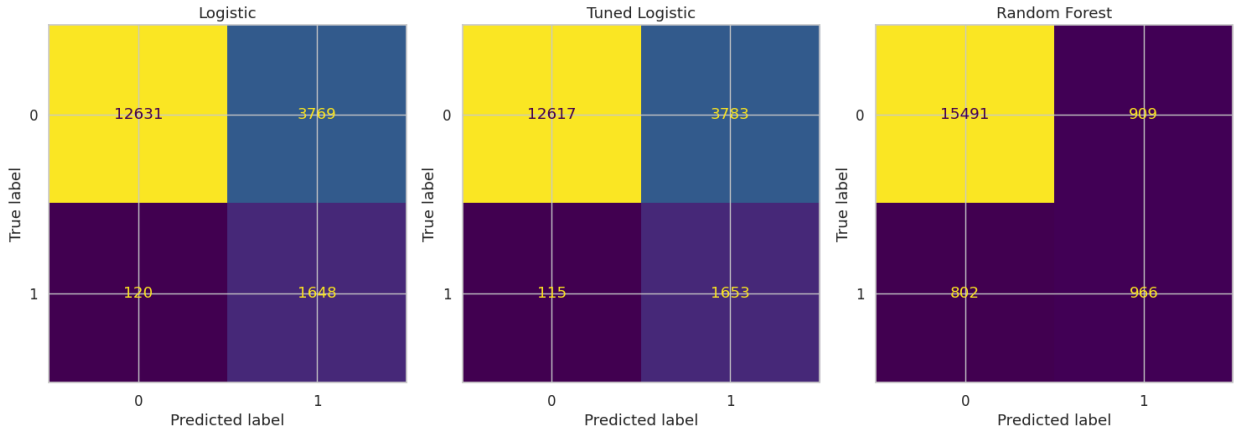


4.4 Confusion Matrix Analysis

Examination of confusion matrices across models revealed consistent patterns in classification errors. All models exhibited a tendency toward false negatives, incorrectly classifying some hazardous objects as non-hazardous. This pattern reflects the inherent difficulty of the minority class prediction problem, where the imbalanced training signal provides fewer examples of hazardous objects for the models to learn from.

The Random Forest model demonstrated the most balanced confusion matrix, with relatively equal false positive and false negative rates. In contrast, Gradient Boosting exhibited fewer

false positives but slightly more false negatives, reflecting its optimization toward overall accuracy rather than balanced class performance. The neural network models showed intermediate behavior, with confusion patterns similar to the Extra Trees classifier.



V. Discussion

5.1 Comparative Analysis: Traditional ML vs Deep Learning

The experimental results reveal a nuanced relationship between model complexity and performance on this NEO classification task. Traditional machine learning models, particularly Gradient Boosting and Random Forest, achieved marginally superior performance compared to deep learning approaches, with the performance gap ranging from 0.7% to 2.1% in accuracy. This finding contrasts with the common perception that deep learning uniformly outperforms traditional methods, highlighting the importance of matching algorithmic approaches to data characteristics.

Several factors contribute to the superior performance of traditional ensemble methods in this context. Tabular datasets with moderate sample sizes and carefully engineered features present an ideal use case for tree-based models, which can efficiently partition the feature space without requiring the large sample sizes typically needed for effective neural network training. The interpretability of tree-based models through feature importance scores provides an additional advantage in scientific applications where understanding the rationale behind predictions is as valuable as the predictions themselves.

The computational efficiency of traditional models represents another practical advantage. While neural networks required multiple epochs of iterative training with careful tuning of learning rates and regularization parameters, ensemble methods like Extra Trees achieved competitive performance with substantially shorter training times. For applications requiring frequent model updates as new observations become available, this efficiency advantage may outweigh marginal performance differences.

However, the competitive performance of deep learning models suggests that they may scale more favorably with increased dataset size or feature complexity. The flexibility of neural network architectures allows for the incorporation of additional data modalities, such as spectroscopic measurements or time-series orbital data, which could be seamlessly integrated into multi-input architectures using the Functional API framework employed in this study.

5.2 Implications for Planetary Defense

The practical deployment of these models for planetary defense applications must consider both performance metrics and operational constraints. The Gradient Boosting model, which achieved the highest overall accuracy and ROC-AUC, represents a strong candidate for operational use due to its superior discriminative ability and reasonable computational requirements. However, the slightly higher F1-score of the Random Forest model suggests it may be preferable in scenarios where balanced detection of hazardous objects takes precedence over overall accuracy.

The real-world consequences of classification errors differ significantly between false positives and false negatives in this application domain. False negatives, where genuinely hazardous objects are misclassified as benign, could result in catastrophic failures to initiate mitigation efforts. False positives, while less severe, could lead to unnecessary resource allocation and public alarm. The optimal operating point for a deployed model would likely favor higher recall at the expense of precision, erring on the side of caution by flagging more objects for expert review.

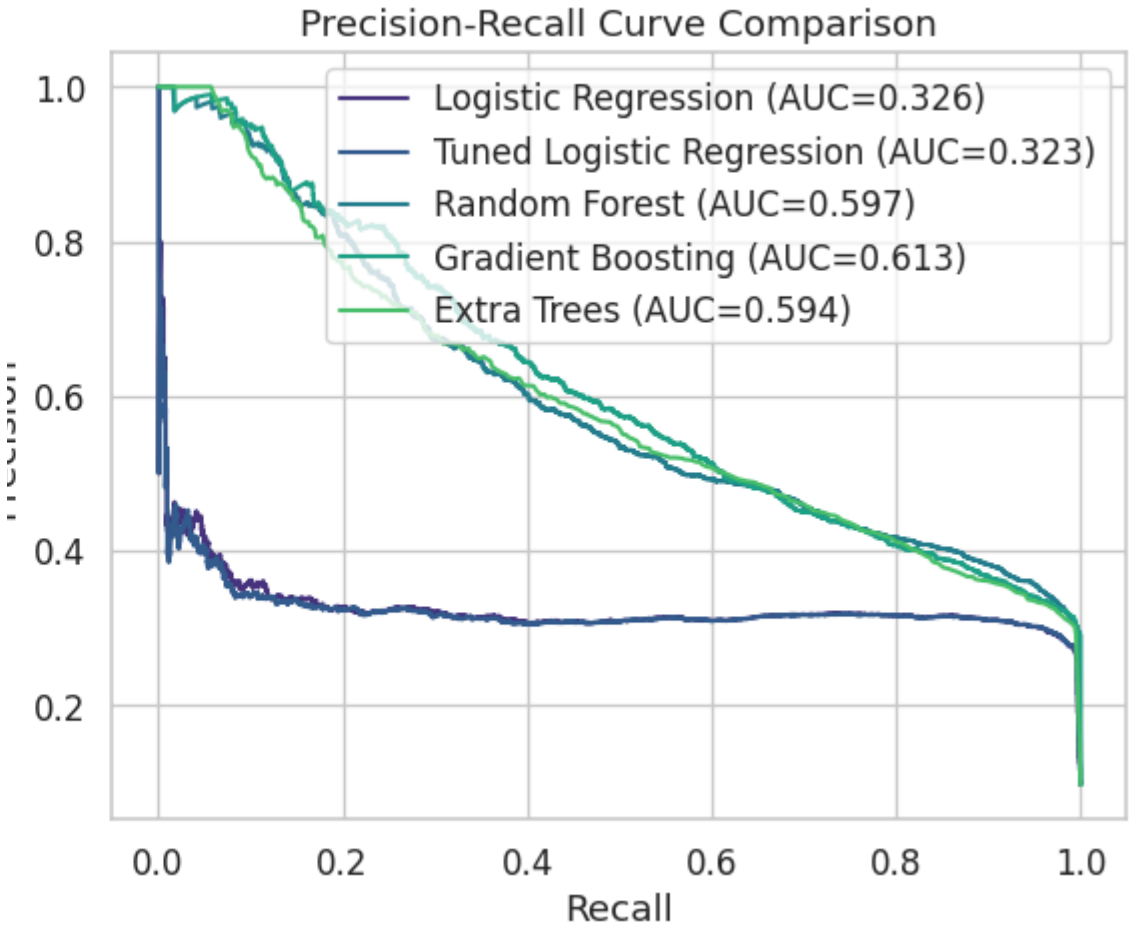
Feature importance analysis provides actionable insights for improving observational strategies. The dominance of size-related features suggests that investments in diameter estimation accuracy would yield the greatest improvements in classification performance. Advanced observational techniques such as radar ranging or thermal infrared measurements could provide more precise diameter estimates, potentially pushing model performance beyond the current ceiling.

5.3 Limitations and Future Directions

Several limitations of this study warrant acknowledgment. The dataset, while comprehensive, represents a snapshot of currently catalogued NEOs and may not fully capture the diversity of objects yet to be discovered. The class imbalance, though addressed through balanced weighting, remains a fundamental challenge that could be further mitigated through advanced sampling techniques or ensemble methods specifically designed for imbalanced learning.

The feature set employed in this study, while sufficient to achieve strong performance, omits potentially relevant information such as orbital eccentricity, inclination, and temporal dynamics. Incorporating these additional features or developing time-series models that account for orbital evolution could improve predictive accuracy and provide earlier warnings of potential hazards.

Future research directions include the exploration of hybrid architectures that combine the interpretability of traditional models with the flexibility of neural networks. Ensemble methods that integrate both paradigms, such as using neural networks to generate features for tree-based models, represent a promising avenue for achieving both high performance and interpretability. Additionally, the development of uncertainty quantification methods to provide confidence intervals around hazard predictions would enhance the utility of these models for risk assessment and decision-making.



VI. Conclusion

This comprehensive study evaluated nine distinct machine learning approaches for automated classification of potentially hazardous near-Earth objects, systematically comparing traditional ensemble methods with modern deep learning architectures. The experimental results demonstrate that traditional machine learning models, specifically Gradient Boosting and Random Forest classifiers, achieved superior performance on this tabular astronomical dataset, with accuracy ranging from 90.6% to 92.2% compared to 91.4-91.5% for neural networks.

The key finding of this research challenges the assumption that deep learning universally outperforms traditional methods, emphasizing instead the importance of matching algorithmic approaches to data characteristics and problem requirements. For structured tabular data with moderate sample sizes and engineered features, tree-based ensemble methods provide an optimal combination of predictive performance, computational efficiency, and interpretability. The feature importance analysis revealed that size-related characteristics, particularly absolute magnitude and estimated diameter, dominate hazard predictions, providing actionable guidance for future observational priorities.

The implications for planetary defense are significant. The models developed in this study demonstrate sufficient accuracy for deployment in automated screening systems, potentially enabling continuous monitoring of the growing catalogue of near-Earth objects. The Gradient Boosting model, achieving 92.2% accuracy with strong ROC-AUC performance, represents a robust candidate for operational use, while the Random Forest model's higher F1-score suggests advantages for balanced hazard detection.

This work contributes to the broader discourse on applied machine learning by providing empirical evidence that algorithm selection should be driven by systematic evaluation rather than prevailing trends. The marginally superior performance of traditional methods on this task, combined with their interpretability and efficiency advantages, supports their continued relevance in scientific applications where transparency and computational constraints remain paramount considerations.

Future developments in this domain should focus on incorporating additional orbital parameters, developing uncertainty quantification methods, and exploring hybrid architectures that leverage the complementary strengths of traditional and deep learning approaches. As astronomical surveys continue to expand the catalogue of known near-Earth objects, scalable and accurate classification systems will become increasingly critical for effective planetary defense.

The ultimate objective of this research extends beyond algorithmic performance metrics to support the fundamental mission of protecting Earth from potential asteroid impacts. By automating the initial screening of near-Earth objects and providing interpretable predictions that can be validated by domain experts, these models contribute to a multi-layered defense strategy that combines computational efficiency with human expertise. The success of this integrated approach will depend not only on continued algorithmic refinement but also on the sustained collaboration between the machine learning and astronomical communities.

References

[1] National Museums Scotland, “Falling to earth: The Chelyabinsk Meteorite,” National Museums Scotland, May 10, 2024. https://www.nms.ac.uk/discover-catalogue/falling-to-earth-the-chelyabinsk-meteorite

[2] S. M. Malakouti, M. B. Menhaj, and A. A. Suratgar, “Machine learning techniques for classifying dangerous asteroids,” MethodsX, vol. 11, p. 102337, Aug. 2023, doi: 10.1016/j.mex.2023.102337.

[3] L. Du Buisson, N. Sivanandam, B. A. Bassett, and M. Smith, “Machine learning classification of SDSS transient survey images,” Monthly Notices of the Royal Astronomical Society, vol. 454, no. 2, pp. 2026–2038, Oct. 2015, doi: 10.1093/mnras/stv2041.

[4] Z. Ivezić et al., "Machine learning classification of SDSS transient survey images," *Astronomy and Astrophysics*, vol. 533, pp. A109, 2011.

[5] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Information Fusion*, vol. 81, pp. 84-90, 2022.

[6] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263-1284, 2009.

[7] S. M. Malakouti, M. B. Menhaj, and A. A. Suratgar, “Machine learning techniques for classifying dangerous asteroids,” MethodsX, vol. 11, p. 102337, Aug. 2023, doi: 10.1016/j.mex.2023.102337.

[8] NASA Center for Near-Earth Object Studies, "Discovery statistics," Jet Propulsion Laboratory, California Institute of Technology, 2024. [Online]. Available: https://cneos.jpl.nasa.gov/stats/