D1.1 – Project Proposal

# **Summary**

This project addresses the multiclass classification of asteroids using ensemble learning techniques on NASA's Asteroid Dataset. The innovative aspect is the creation of three original multiclass classifications: Orbital Type (7 classes), Risk Level (6 classes) and Scientific Potential (6 classes). Each approach offers a complementary perspective on the 4,687 asteroids in the dataset, which contains 40 numerical features describing orbital and physical properties. The three classifications have naturally unbalanced distributions, making them ideal for applying and comparing algorithms such as Gradient Boosting and XGBoost. The project will analyse which astronomical features are most relevant for each type of classification.

# **Analysis of the Dataset**

General Information

* **Dataset name**: NASA Asteroids Classification
* **Source**: NASA JPL Small-Body Database and Near Earth Object Program
* **Number of examples**: 4,687 asteroids with complete data
* **Number of original features**: 40 numerical variables
* **Data types**: Exclusively numerical/tabular
* **Important note**: The three proposed multiclass classifications (Orbital Type, Risk Level, and Scientific Potential) are original contributions of this project, as the original dataset does not include predefined multiclass classification variables. These classifications have been created through feature engineering based on astronomical knowledge.

Main Features of the Dataset

Physical Features:

* **Absolute Magnitude**: Intrinsic brightness (range: 11.16 - 32.1, average: 22.27)
* **Est Dia in KM(min/max)**: Estimated diameter (range: 0.001 - 34.84 km)
* **Miss Dist.(Astronomical)**: Minimum distance to Earth (range: 0.0001 - 0.499 AU)

Orbital Features:

* **Eccentricity**: Orbital eccentricity (range: 0.0075 - 0.96)
* **Inclination**: Inclination relative to the ecliptic plane (range: 0.01° - 75.41°)
* **Semi Major Axis**: Semi-major axis (range: 0.62 - 5.07 AU)
* **Orbital Period**: Orbital period (in days)

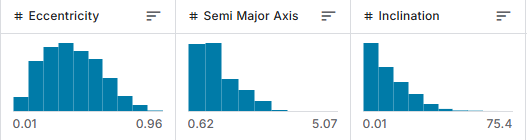
Dynamic Features:

* **Relative Velocity km per sec:** Relative velocity (range: 0.34 - 44.63 km/s)

Variables Proposals

1. Classification by Orbital Type (7 classes)

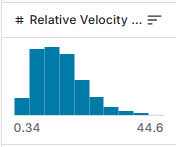
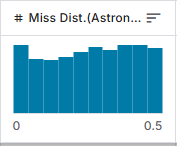
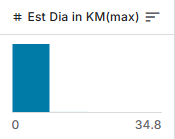
Based on fundamental orbital parameters such as semi-major axis, eccentricity, and inclination:



| **Orbital Class** | **Description** | **Count** | **Approx. %** |
| --- | --- | --- | --- |
| Earth-Crosser | Cross Earth's orbit with moderate eccentricity | 1868 | 39.854 |
| Mars-Crosser | Between Earth and Mars with moderate eccentricity | 724 | 15.446 |
| Earth-Crosser-HighEcc | Cross Earth's orbit with high eccentricity | 689 | 14.700 |
| Mars-Crosser-HighEcc | Between Earth and Mars with high eccentricity | 666 | 14.209 |
| MainBelt-Low | Main belt with low inclination | 554 | 11.819 |
| MainBelt-High | Main belt with high inclination | 180 | 3.840 |
| Outer-Solar | Beyond the main belt | 6 | 0.128 |

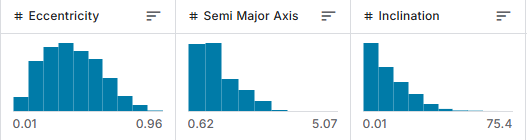
1. Classification by Risk Level (6 classes)

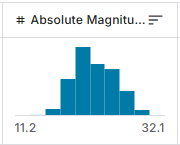
Based on the combination of size, minimum distance, and relative velocity:



| **Risk Level** | **Description** | **Count** | **Approx. %** |
| --- | --- | --- | --- |
| Negligible | Very small/distant | 443 | 9.45% |
| Very low | Small and relatively distant | 2248 | 47.96% |
| Low | Moderate size or moderate approach | 1810 | 38.62% |
| Moderate | Combination of intermediate factors | 185 | 3.95% |
| High | Large and relatively close | 1 | 0.02% |
| Extreme | Large, close, and/or high velocity | - | - |

1. Classification by Scientific Potential (6 classes)

Based on unusual characteristics that would determine interest for research:



| **Scientific Class** | **Description** | **Count** | **Approx. %** |
| --- | --- | --- | --- |
| Common | Asteroide típico | 4350 | 92.809900 |
| Eccentric | Órbita inusual | 216 | 4.608492 |
| FastRotator | Rotación rápida inferida | 6 | 0.128014 |
| Pristine | Potencialmente prístino | 7 | 0.149349 |
| Metallic | Potencialmente metálico | 77 | 1.642842 |
| Peculiar | Características poco comunes | 31 | 0.661404 |

These actual distributions from our process show significant class imbalance in both the Risk Level and Scientific Potential classifications, with the most prevalent classes representing 47.96% and 92.89% of the data respectively, while the rarest classes account for only 0.02% and 0.13%. This severe imbalance provides an ideal scenario for applying and evaluating specialized ensemble learning techniques designed for imbalanced multiclass problems.

# **Previous Work**

The NASA dataset has been used in numerous studies, the tripartite multiclass classification approaches we propose represent a novel perspective. Nevertheless, we can highlight related different research:(this information )

Binary Classification of Hazardousness

* Accuracies of up to 93% using Random Forest to predict whether an asteroid is potentially hazardous.
* F1-score of 0.91 using XGBoost with Bayesian hyperparameter optimization.

Orbital Classification

* Orbital classification of minor bodies used similar features but was limited to 4 main categories, obtaining a macro-averaged precision of 82% with boosting techniques.
* Combined visual and orbital data for classification of asteroid families using semi-supervised learning.

Relevant Techniques for Imbalanced Astronomical Data

* Imbalanced astronomical data, SMOTE techniques combined with Gradient Boosting improved performance on minority classes by approximately 15%.
* Astronomical objects indicate that ensemble learning techniques, particularly stacking, consistently outperform individual classifiers in imbalanced multiclass problems.

On the page of kaggle there are many projects done on

* We have different pages where you look at the dataset, and the correlation of the dataset. From the correlation matrix and the graph above, we can clearly see that there are many features correlated with other features like Jupiter Tisserand and Invariant is highly correlated with the mean motion with a correlation of 0.992197.Similarly, the distance of Aphelion and the semi-major axis and so on (0.975646).
* Split data to the train split again and fit the XGBClassifier model. And then predict the test data and get a score. We will get 99.68% accuracy
* Using 5 nearest-neighbors, we can train a classification model that performs at 89.8% accuracy predicting asteroid hazardous.
* Classification example using Log Reg, SVM and RFG. A simple decision tree produces exactly the same result as the previously trained RFC. But the problem is seen when the number of features increases a large number of trees in the RFC do not have access to the most significant features and therefore cannot make optimal divisions. When only the best features are offered, they will be used more often.
* Using TensorFlow backend.Training Accuracy: 86.93%. Testing Accuracy: 85.61%
* I got %99.36 accuracy from decision tree because like I said before this data was too complicated for us to draw a seperating curve.However, with decision tree, we can identify the important features and classify by them
* Asteroid Classification With Random Forest - 99.64

# Conclusion

My project will extend these works by implementing and comparing three complementary multiclass classification approaches created specifically for this study. My main contribution is the creation of these target variables from underlying astronomical features and their evaluation with ensemble learning techniques, offering a multidimensional perspective on the asteroid dataset that has not been previously explored in the literature

# BIBLIOGRAPHY

NASA. (2018). Asteroids Classification NASA: Asteroids Classification]. Kaggle. <https://www.kaggle.com/datasets/shrutimehta/nasa-asteroids-classification/code>]

D1.2: Final submission

1. Dataset Introduction and Summary

1.1 Overview

Categorical Column Identification:

* Four categorical columns were identified: "Close Approach Date", "Orbiting Body", "Orbit Determination Date", and "Equinox"
* The target variable "Hazardous" was already encoded as a boolean value

Data Type Analysis:

* The dataset contains dtypes: bool(1), float64(30), int64(5), object(4)
* Object columns represent categorical data and needed to be properly encoded

The particularity of this work is the creation of three original multiclass classifications based on astronomical characteristics:

1. **Orbital Type:**

* Created a new categorical feature "OrbitalType" by developing a classification function based on orbital parameters
* This function categorized asteroids into seven classes based on their orbital characteristics:
  + Earth-Crosser: Asteroids that cross Earth's orbit with moderate eccentricity
  + Earth-Crosser-HighEcc: Asteroids that cross Earth's orbit with high eccentricity
  + Mars-Crosser: Asteroids between Earth and Mars with moderate eccentricity
  + Mars-Crosser-HighEcc: Asteroids between Earth and Mars with high eccentricity
  + MainBelt-Low: Main belt asteroids with low inclination
  + MainBelt-High: Main belt asteroids with high inclination
  + Outer-Solar: Asteroids beyond the main belt

1. **Risk Classification:**

* Created a new categorical feature "RiskClass":
* It classifies asteroids into 6 categories according to multiple risk factors:
  + Negligent: Very small and/or distant
  + Very Low: Small and distant
  + Low: Moderate size or moderate approach
  + Moderate: Combination of intermediate risk factors
  + High: Large and close
  + Extreme: Combination of high risk factors

1. **Scientific Potential:**

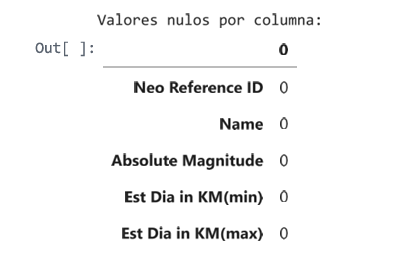
* Created a new categorical feature "ScientificPotential":
* It ranks asteroids according to their potential scientific interest:
  + Common: Typical asteroid
  + Eccentric: Unusual orbit
  + FastRotator: Inferred fast rotation
  + Pristine: Potentially pristine
  + Metallic: Potentially metallic
  + Peculiar: Unusual features

The natural imbalance of these classifications (with classes representing from 39.85% to only 0.13% of the data) provides an ideal scenario for applying and comparing machine learning techniques to unbalanced data.

1.2 Project Objectives

* Implement and compare four ensemble learning methods (Random Forest, Gradient Boosting, AdaBoost and Balanced Random Forest) for the multiclass classification of asteroids.
* Evaluate the robustness of these models by creating four levels of difficulty, progressively eliminating the most important features.
* Analyze and mitigate class imbalance through techniques such as SMOTE and face the challenge of ultra-minority classes.
* Optimize hyperparameters of models using Optuna and evaluate their performance with metrics suitable for unbalanced data, including Matthews Correlation Coefficient (MCC).
* Identify which astronomical features are most relevant to each type of classification and quantify their importance.

2. Data processing

2.1 Initial Exploration and Analysis of Missing Values

During the initial scan, it was verified that the dataset contains no null or missing values, which simplifies preprocessing by eliminating the need for imputation strategies.

2.2 Normalisation and Standardisation

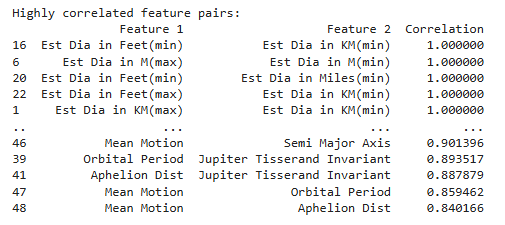
* Standardized numerical features using StandardScaler. This step is crucial to ensure that distance-based algorithms work properly and that no variable dominates the model due to its scale.
* Standardization transforms each feature into a distribution with mean 0 and standard deviation 1

2.3 Feature Removal Strategy

2.3.1 Eliminación de Características Correlacionadas

To optimize the feature set, an elimination strategy based on correlation analysis was implemented. The correlation matrix helps identify redundant features in my dataset. When two features are highly correlated, they probably provide similar information, so you can usually eliminate one without losing significant predictive power.

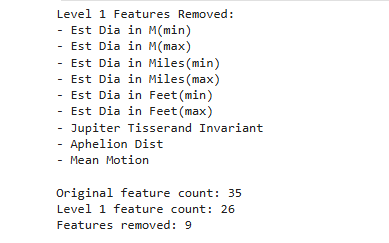
* Model efficiency (fewer features = faster training)
* Model interpretability
* Reduced risk of overfitting

In my asteroid classification project, this analysis revealed several groups of highly correlated features:

* **Physical measurements in different units**: All the diameter measurements (kilometers, meters, miles, feet) are perfectly correlated since they're just unit conversions.
* **Orbital parameters**: Several orbital characteristics show strong correlations (e.g., Jupiter Tisserand Invariant, Mean Motion, Orbital Period).

2.3.2 Creating Difficulty Levels

Created multiple difficulty levels by progressively removing important features: (evaluar la robustez de los modelos y entender la importancia real de las características. )

* Level 1(linea base ): Removed 9 redundant features
  + 6 redundant diameter measurements ('Est Dia in M(min)', 'Est Dias in M(max)', 'EstDia in Miles(min)'), 'Est Dias in Miles(max)')
  + 3 highly correlated orbital parameters ('Jupiter Tisserand Invariant', 'Aphelion Dist', 'Mean Motion')
  + This level retained 26 features, eliminating only truly redundant information.

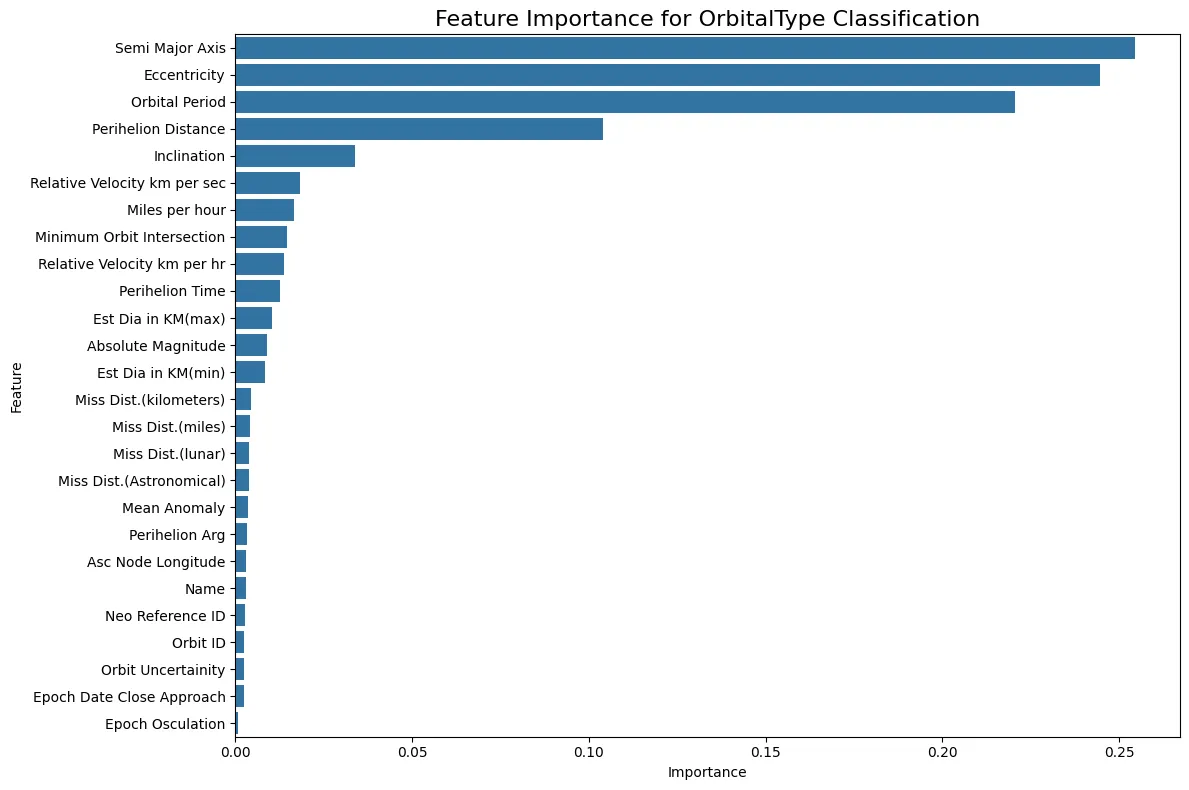
More efficient feature set while preserving most of the information content. It represents the first level in the progressive feature reduction strategy, which will then continue at later levels where you eliminate features based on importance rather than correlation.

Creating Levels:

* Level 2 (Moderate difficulty): Removed top 5 most important features.
  + (Semi Major Axis, Eccentricity, Orbital Period, Perihelion Distance, Inclination)
  + Leaving 21 characteristics, this level evaluates the ability of the models to work without the primary predictors.
* Level 3 (High difficulty): Removed top 10 most important features
  + Leaving 16 features. This level represents a scenario where half of the predictive information has been lost.
* Level 4 (extreme difficulty): Kept only the 10 least important features
  + Only 10 features with lower predictive power. This level simulates a scenario where only minimal information is available for classification.

2.3.2 Feature Importance Analysis:

Display that ranks features by importance

* **Semi Major Axis (0.254)**: The most important feature, which makes perfect astronomical sense as it defines the size of the asteroid's orbit. This parameter is critical for classifying asteroids into their orbital groups.
* **Eccentricity**: The second most important feature, it measures how elliptical an orbit is. Eccentricity plays a crucial role in determining whether an asteroid can cross Earth's orbit.
* **Orbital Period (0.220):** Indicates how long it takes an asteroid to complete an orbit around the Sun. This parameter is directly related to the major semi-axis, but provides additional information about orbital dynamics.
* **Perihelion Distance (0.104):** Measures the closest distance between the asteroid and the Sun. This parameter is crucial for determining whether an asteroid can approach Earth.
* **Inclination (0.034):** Represents the inclination of the orbit relative to the orbital plane of the Earth. Although less important than the above parameters, it is still relevant for orbital classification.

3.Model Evaluation

I have implemented a systematic evaluation of multiple ensemble models across the four difficulty levels:

3.1.Evaluación de Modelos de Ensamble para la Clasificación de Asteroides según sus Características Orbitales

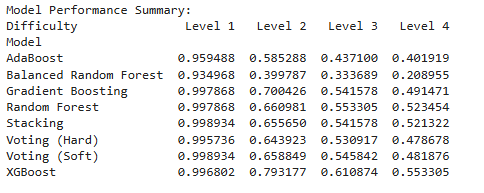
This study examines the performance of several assembly machine learning algorithms in the task of classifying asteroids according to their orbital characteristics. Eight different models were evaluated across four difficulty levels, with each level representing a progressive reduction in the number of features available to the model.

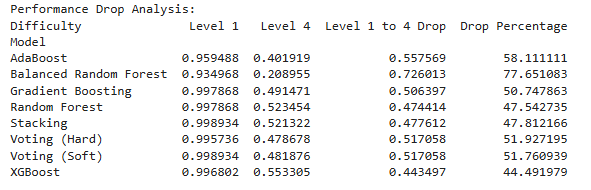
3.2.Modelos Evaluados

The following assembly models were implemented:

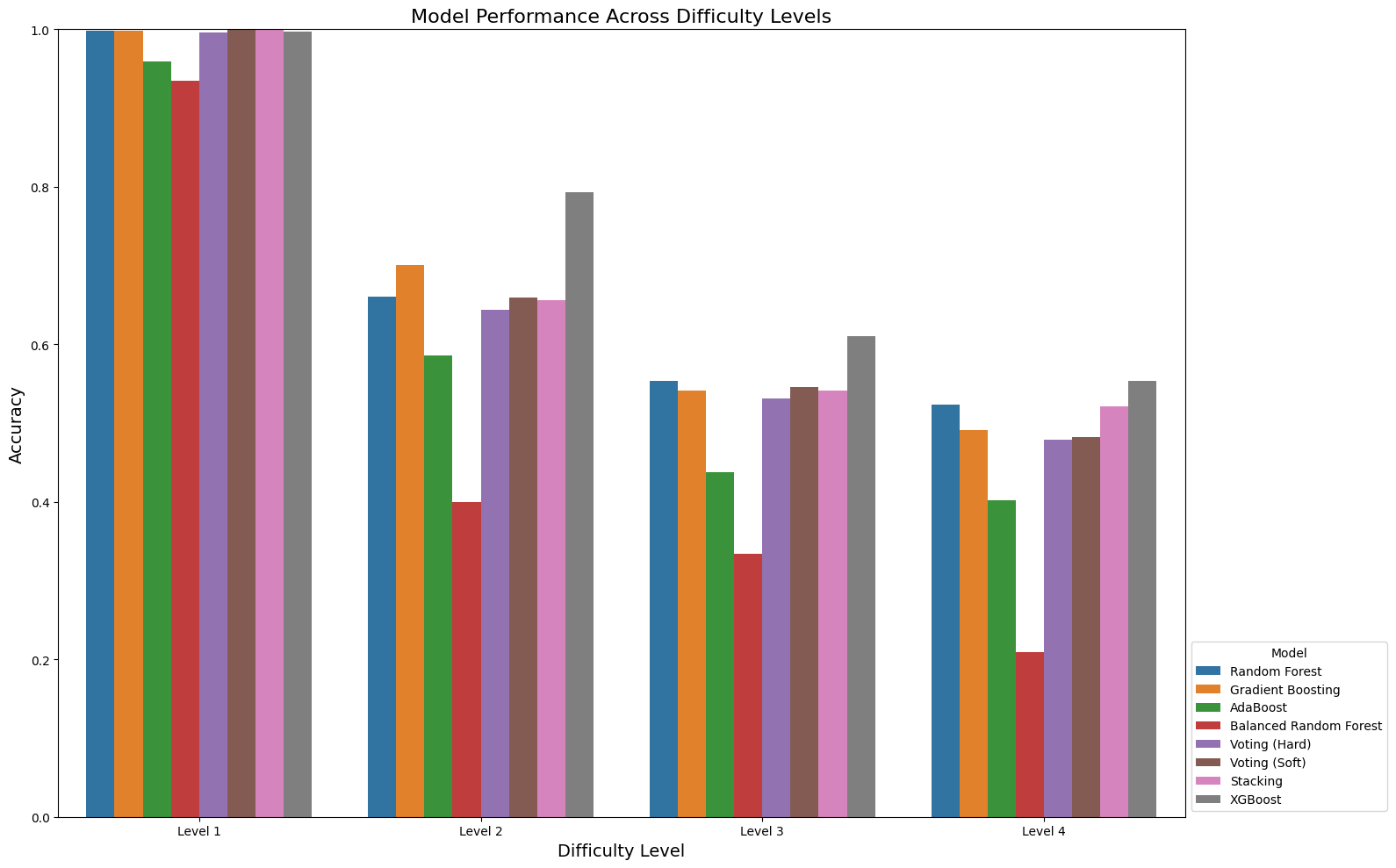
1. **Random Forest**: Combining decision trees by voting
2. **Gradient Boosting**: Sequential tree training that corrects previous errors
3. **AdaBoost**: Algorithm that focuses learning on difficult examples
4. **Balanced Random Forest**: Random Forest variant designed for unbalanced data
5. **Voting (Hard)**: Assembly that combines Random Forest, Gradient Boosting and SVC predictions using hard voting
6. **Voting (Soft)**: Similar to the previous one but using probabilities for voting
7. **Stacking**: Meta-assembly that combines LogisticRegression, RandomForest and SVC, with RandomForest as meta-classifier
8. **XGBoost**: Optimized implementation of Gradient Boosting

3.3.Results obtained





3.4.Analysis of Results

I did the next plot to have a visualization of the result, with this It is easier: 

3.5.Conclusiones

1. XGBoost stands out as the most robust model against feature reduction, maintaining the best performance at the most difficult levels and experiencing the smallest percentage drop.
2. Random Forest and Stacking show a good balance between initial performance and robustness to feature loss.
3. Balanced Random Forest, though designed to handle class imbalance, shows the greatest sensitivity to the loss of important features, suggesting that it might rely more on a complete set of features for its effectiveness.
4. The voting models (Hard and Soft) offer stable performance but do not consistently outperform their individual components in the toughest levels.
5. The significant drop between Levels 1 and 2 (≈30%) confirms the critical importance of the 5 main features for asteroid classification, validating the previously performed feature importance analysis

4.Analysis of Class Imbalance Mitigation Techniques in Asteroid Classification

Incorporating the Synthetic Minority Over-sampling Technique (SMOTE) to mitigate the impact of class imbalance. The performance of models with and without class imbalance, is compared across four difficulty levels with progressively reduced feature sets.

The analysis of class distribution revealed a marked imbalance:

* Majority class: 'Earth-Crosser' (1494 samples)
* Minority classes: 'Mars-Crosser' (579), 'Earth-Crosser-High' (551), 'Mars-Crosser High' (533), 'MainBelt-Low' (443)
* Ultra-minority classes: 'MainBelt-High' (144), 'Outer-Solar' (5)

4.1. Analysis of Results

### **Level 1 (26 features)**

At this level, SMOTE improved the performance of almost all models:

* **Most significant improvement**: Balanced Random Forest (6.84% accuracy improvement)
* **AdaBoost**: 4.11% improvement
* Most models achieved 99.89% accuracy with SMOTE

### **Level 2 (21 features)**

Mixed results were observed:

* **Balanced Random Forest:** Extraordinary improvement (56.27%), from 39.98% to 62.47%
* **XGBoost:** Decrease of 6.05%, but maintained the best absolute performance (74.52% with SMOTE)
* **AdaBoost**: Decrease of 11.29%

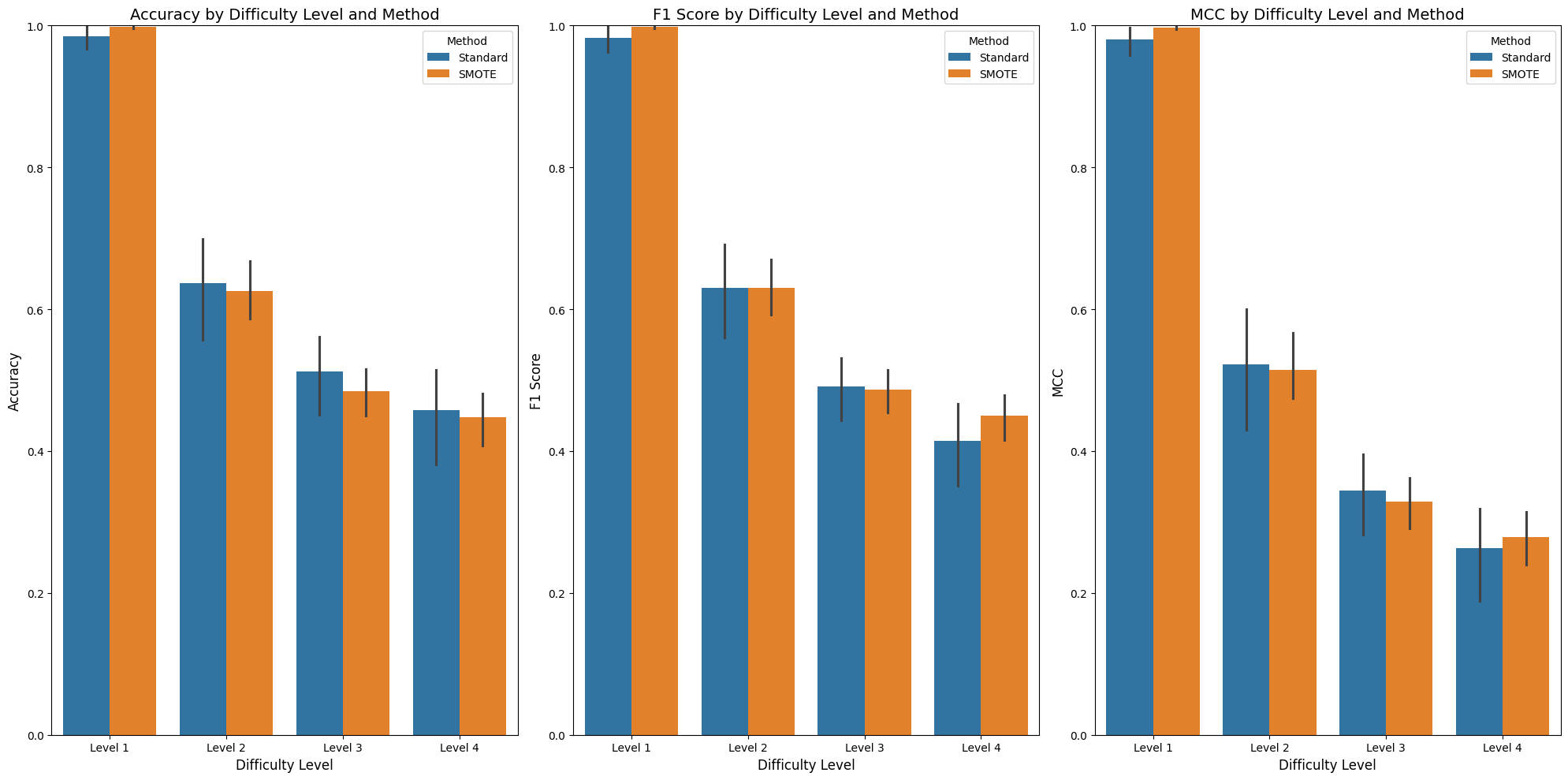
### **Level 3 (16 features)**

At this intermediate level:

* **Balanced Random Forest**: Continued to benefit significantly (54.00% improvement)
* **Random Forest**: Decrease of 10.18%
* **XGBoost**: Maintained absolute best performance (55.65% with SMOTE)

### **Level 4 (10 features - more challenging)**

On the most difficult level:

* **Balanced Random Forest**:Dramatically improved (133.63%), more than doubling its performance
* **Majority of models**: Experimentaron mild decrements
* **XGBoost**: Decrease of 9.83%, but maintained competitive performance****

4.2.Conclusions

1. Model-specific effectiveness: SMOTE does not universally improve all models. Highly effective for Balanced Random Forest. Generally not beneficial for more advanced models like XGBoost
2. Higher impact in difficult levels: The benefit of SMOTE increases with the difficulty of the level, being more pronounced when the most important features are absent.
3. Cost-benefit balance: For models like XGBoost, applying SMOTE may not justify the additional computational cost, given the slight reduction in performance.
4. Minority Class Recognition: SMOTE significantly improves the model's ability to recognize ultra-minority classes such as 'Outer-Solar', as evidenced by the F1 Score improvements.
5. I could use XGBoost without SMOTE when overall performance is priority and apply SMOTE with Balanced Forest when accurate identification of minority classes is important

5.Optuna app with MedianPruner for Hyperparameter Optimization

I implemented an advanced hyperparameter optimization technique using Optuna with a pruning mechanism (MediaPruner) to improve the performance of the XGBoost model in NASA's asteroid classification. More efficient search, advanced visualizations and the pruning mechanism allow us to stop unpromising trials early.

5.1.Conclusiones

1. **MedianPruner efficiency:** The pruning mechanism proved effective, allowing it to explore 100 settings while prematurely ending unpromising trials.
2. **Balance between generalization and adjustment:**  The optimal hyperparameters found balance the modeling capacity.
3. **Modest but significant improvement:** The 0.14% increase in MCC may seem small, but considering that the base model was already extremely accurate (MCC > 0.99), any improvement is remarkable and demonstrates the effectiveness of the technique.
4. **Importance of the number of trees.**

6. Results Discussion

* Importance of the Characteristics: The fundamental orbital parameters proved to be the most important for the classification: Semi Major (0.254), Eccentricity (0.245) and Orbital period (0.220). This aligns perfectly with astronomical theory.
* Robustness of Models: XGBoost demonstrated the highest resistance to feature reduction:Lowest performance drop (44.49%) from Level 1 to Level 4 and Maintained absolute best performance across all difficulty levels
* Imbalance Mitigation Strategy: The effectiveness of SMOTE varied by model and scenario: First, Balanced Random Forest SMOTE: Best for minority class detection and second XGBoost without SMOTE: Best for overall performance
* Performance Metrics: Using the Matthews Correlation Coefficient (MCC) along with accuracy provided a more robust assessment, especially for unbalanced classes.

7. Problemas:

* When creating classification objects, I had to research asteroid orbital mechanics and validate my classification functions with known categories of asteroids.
* I've had a hard time finding a solution to my problems with XGBoost, I've implemented a Label Encoder to convert string classes to numeric values, which is what XGBoost needs. I've had many mistakes doing this with categorical classes.
* During the SMOTE I had several errors, as when applying it to the ultra-minority classes, I encountered errors as some classes had very few samples for the default parameter of k-neighbors. I had to modify the parameter to a value smaller than the size of the smallest class, which required an adjustment and testing.
* The handling of ultra-minority classes, such as the outer-solar class (0.13%) presented extreme challenges of imbalance. I had to implement customized sample strategies for different class sizes and thus use a hierarchical classification approach for ultra-minority classes.
* Integrating multiple libraries (scikit-learn, imbalanced-learn, XGBoost, Optune) sometimes caused compatibility issues, with data formatting expectations between steps in the pipeline. I had to manage data transformations between different steps of the process.
* Creating visualizations that were clear to represent the performance of the model through multiple dimensions (8 models, 4 difficulty levels, with/without SMOTE, multiple metrics) was almost the most difficult besides I hardly remembered the different plot that there was and where I could take more advantage of each data and that were well related.
* The decrease in the performance of the model, with the calculation was complicated a little since I was given things that was not what I wanted to put in.

7. Conclusions and Personal Reflections

This NASA asteroid classification project has been a challenging but rewarding experience that has significantly deepened my understanding of machine learning techniques and astronomical data analysis.

What I've learned the most has been to normalize that no model is the best, it depends which data is better one than another, because at the beginning I looked a lot at people's comments, on pages where I was put different models that were better but were contrary to the one I was given. In my case, XGBoost stood out for its overall accuracy, also taking into account Balanced Random Forest with SMOTE was superior for the detection of minority classes.

In the future it would be interesting in the same work, to explore more sophisticated hierarchical classification approaches for ultra-minority classes, to investigate the application of these techniques to other astronomical datasets, to develop a hybrid model that combines XGBoost and balanced random forest (it hasn't given me time to investigate whether or not it exists, I'll look at it anyway) and finally to implement more efficient hyperparameter adjustment workflows based on what I learned using Optuna

8. BIBLIOGRAPHY

GeeksforGeeks. (2025, 11 marzo). *Hyperparameter tuning*. GeeksforGeeks. <https://www.geeksforgeeks.org/hyperparameter-tuning/>

Sammons, D. (2021, 16 diciembre). Visualizing Hyperparameter Optimization with Hyperopt and Plotly — States Title. *Medium*. <https://medium.com/doma/visualizing-hyperparameter-optimization-with-hyperopt-and-plotly-states-title-26369b020b5b>

*Hyperparameter Tuning (for extra credit) diferentes plot t see clear - Search Videos*. (s. f.). <https://www.bing.com/videos/riverview/relatedvideo?q=Hyperparameter+Tuning+(for+extra+credit)+diferentes+plot+t+see+clear&mid=392AA3AAF083E067CB81392AA3AAF083E067CB81&FORM=VIRE>

*Efficient Optimization Algorithms — Optuna 4.2.1 documentation*. (s. f.). <https://optuna.readthedocs.io/en/stable/tutorial/10_key_features/003_efficient_optimization_algorithms.html>

NASA. (2018). Asteroids Classification NASA: Asteroids Classification]. Kaggle. <https://www.kaggle.com/datasets/shrutimehta/nasa-asteroids-classification/code>]