EXPLORING ASTEROID ORBITS: INSIGHTS FROM NEURAL NETWORK MODELING AND DATA DRIVEN ANALYSIS

Aarav Sonthalia
Short Hills, NJ, USA

ABSTRACT

Classifying the orbits of asteroids contributes to research on the Solar System's formation and evolution, making the classification of orbits a fundamental aspect of space exploration. However, accurate orbit determination is often hindered by skewed observational data stemming from uneven and limited sky coverage. In this study, orbital data from the NASA Jet Propulsion Lab was used in the classification of asteroid orbits through a machine-learning approach. Due to imbalances in the dataset, Synthetic Minority Oversampling Technique (SMOTE) was used to compensate for limited observations of certain orbit types. Afterwards, several classification models were implemented using specific orbital features. The most accurate results in this study were produced by a custom Keras neural network, with similar results demonstrated by an MLP neural network and k-nearest neighbors model. The custom neural network was able to effectively distinguish between complex orbital patterns, as demonstrated by its 98.76% validation accuracy and nearly diagonal confusion matrix. The capability of these models not only contributes to our understanding of asteroid orbits but also suggests potential improvements in orbit determination methodologies.

INTRODUCTION

Understanding the orbital dynamics of asteroids is needed for researching planetary formation and evolution. Asteroids, remnants of the early solar system, exhibit orbital patterns influenced by gravitational interactions with planets, stars, and other celestial bodies. These orbital patterns provide information into the origins and evolution of asteroids, shedding light on the processes that shaped our solar system. By categorizing asteroids based on their orbital characteristics, researchers can understand asteroid formation processes and collision histories (1). Furthermore, accurate classification can influence planetary defense efforts by allowing for identification of asteroids that could potentially collide with Earth (2). Asteroid classification leverages observational studies and computational modeling methods. This paper explores the success of different machine-learning models in accurately classifying asteroid orbits.

Asteroid orbits are defined by orbital parameters including but not limited to eccentricity (the shape of the orbit ranging from circular to highly elliptical), semi-major axis length (the average distance from the Sun), inclination (the tilt of the orbit relative to the ecliptic plane), orbital period (the time taken to complete one orbit around the Sun), absolute magnitude (a measure of visual brightness to an observer), diameter (the size), geometric albedo (the reflectivity of the surface), median anomaly (the position of on an orbit), perihelion distance (the closest distance to the Sun), mean motion (the average angular speed), and argument of perihelion (the orientation of the orbit relative to the perihelion direction) (3).

Asteroid orbits are categorized based on their proximity to Earth, orbital characteristics, and location within the solar system. This paper focuses on Apollo asteroids (orbits intersect with Earth's orbit), Amor asteroids (orbits approach but do not cross Earth's orbit), Atira asteroids (orbits are entirely within Earth's orbit), Mars-crossing asteroids (orbits cross the orbit of Mars), Inner Main-belt asteroids (orbits reside in the inner region of the asteroid belt between Mars and Jupiter), Main-belt asteroids (orbits reside in the main region of the asteroid belt between Mars and Jupiter), Outer Main-belt asteroids (orbits reside in the outer region of the asteroid belt between Mars and Jupiter), Jupiter Trojan asteroids (orbits share Jupiter's orbit and reside at the L4 or L5 Lagrange Points), Centaur asteroids (small solar system bodies located between Jupiter and Neptune), Trans-Neptunian Objects (minor planets that orbit at a greater average distance than Neptune), and Hyperbolic asteroids (orbits are hyperbolic in shape) (4). These classifications are essential for understanding the distribution, dynamics, and potential hazards associated with different groups of asteroids.

BACKGROUND

In recent years, advancements in machine learning have transformed multi-class classification, providing tools for analyzing large-scale multi-class datasets (5). Researchers have begun to use machine learning algorithms in astronomy, leveraging the computational capabilities to more efficiently identify complex patterns and relationships within large amounts of orbital data (6). Investigating the application of machine learning techniques in classifying asteroid orbits allows us to analyze the efficacy of different models in discerning between complex patterns.

However, applying machine learning on uneven data poses challenges. In large datasets, certain classes may be under or oversampled, leading to biases in the classification process. Uneven data distribution can skew the performance of machine learning algorithms in favor of the majority category, resulting in inaccurate predictions. To solve this problem, researchers can employ Synthetic Minority Oversampling Technique (SMOTE). SMOTE is able to solve the date imbalance by generating synthetic samples of minority classes. By augmenting the dataset with manually created data, SMOTE ensures a more even distribution of categories, thereby enhancing the predictive performance machine learning models in multi-class classification (7,8).

DATASET

We sourced the asteroid dataset utilized in this study from the NASA Jet Propulsion Lab. The dataset consists of 958524 entries and 45 columns, with the columns being significant orbital features. We removed all unnecessary features, including all 1-sigma features and features not directly related to the orbit of asteroids. The remaining features were the absolute magnitude (H), diameter of the asteroid (diameter), geometric albedo (albedo), median anomaly (ma), orbital period (per), eccentricity (e), semi-major axis length (a), perihelion distance (q), inclination (i), mean motion (n), argument of perihelion (w), and the orbital classes (class).

We replaced all null values with the calculated median values of the respective orbital features. We then split the cleaned dataset into the predictor variables (X) and the target variable (y), with the target variable being the orbital class of the asteroids. The dataset was further divided into training and testing datasets using an 80-20 split.

To address an imbalance in the dataset, we employed Synthetic Minority Oversampling Technique (SMOTE) for data augmentation. The SMOTE algorithm, initialized with a k-neighbors value of 3, was applied to the dataset after the removal of all main-belt asteroids (MBA) to significantly reduce runtime, as MBAs constituted 89% of the 950,000+ sample dataset (Figure 1).

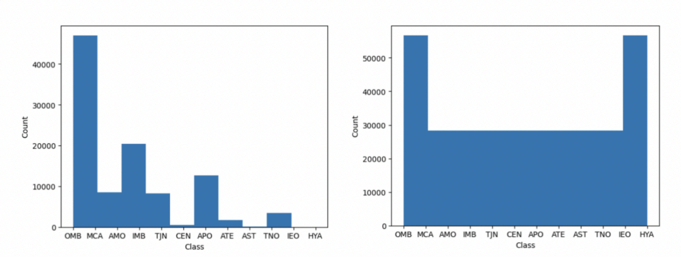

Figure 1. Before and after SMOTE. The distribution of asteroid samples before and after running the dataset through SMOTE for data augmentation. SMOTE was used to augment the data and produce an even number of samples in each asteroid class.

METHODOLOGY/MODELS

We identified four candidate models for classification: logistic regression, random forest, k-nearest neighbors, and neural networks. To prepare the data for the models, we created training and testing datasets with an 80-20 split. Furthermore, before we trained the models, we one-hot encoded the target variable vector to transform the categorical "class" labels into numerical vectors. The one-hot encoded arrays were used during training and later decoded to original labels for model evaluation. We tested the models with various hyperparameter configurations to maximize validation accuracy.

We also created a custom neural network using Keras. The model's architecture consisted of an input layer with 12 nodes (corresponding to the number of features), multiple hidden layers, and an output layer with 12 nodes (representing the orbital classes). The number of hidden layers, nodes per layer, activation function, batch size, and number of epochs were tuned for optimization.

To further optimize the Keras neural networks, we conducted hyperparameter tuning using the Weights and Biases (WandB) platform hyperparameter sweeps. The sweep config consisted of activation functions ('relu', 'tanh', 'sigmoid'), number of layers (2 to 10), number of nodes per layer (2 to 20), number of epochs (100 to 200), and batch size (1000 to 2000). We set up the sweep config with a random search tuning technique to find the maximum validation accuracy. Finally, we set random seeds to improve the reproducibility of the sweeps by removing any randomness from generated neural networks.

We assessed model performance using standard metrics, including accuracy scores and confusion matrices. To help with result interpretation, we reversed the one-hot encoded predictions generated by the models, yielding the original categorical labels in the confusion matrix. We calculated accuracy scores to quantify the overall predictive performance of the models. Additionally, for the Keras neural network models, we plotted the validation and training accuracy over epochs to check for potential overfitting (Figure 2).

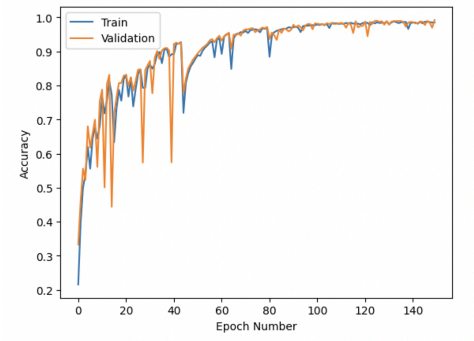

Figure 2. Keras neural network overfitting graph. The graph displaying the training and validation accuracy over epochs of the keras neural network with the highest validation accuracy to check for overfitting. Matplotlib was used to plot the training and validation accuracy over the epochs.

To explore the relationships between the hyperparameters and the validation accuracy, we generated a parallel coordinate plot and correlation table on the Weights and Biases program, as shown in Figure 5. The parallel coordinate plot visualized the performance of different hyperparameter configurations generated by the hyperparameter sweeps, with each colored line representing a trial and each coordinate representing a hyperparameter value. The line color indicates the associated validation accuracy, with warmer colors signifying a higher value. We then created a correlation table to visualize the statistical relationships between the hyperparameters and validation accuracy. A higher value meant a more positive correlation (hyperparameter leads to higher validation accuracy), while a lower value meant the opposite.

RESULTS AND DISCUSSION

We assessed the effectiveness of our models using validation accuracy and confusion matrices.Validation accuracy, calculated as the ratio of the models correctly predicted instances to the known labels of the instances in the validation set, was the primary metric. The confusion matrices provided a detailed breakdown of the true positive, true negative, false positive, and false negative predictions for each class (9). The logistic regression model, initially configured with 200 iterations, achieved a validation accuracy of 52.20%. After increasing the iterations from 200 to 500, the accuracy moderately improved to 56.23%. The corresponding confusion matrix suggested the model struggled with most orbit types, especially Outer Main Belt (OMB), Mars-Crossing (MCA), and Amor (AMO) asteroids (Figure 3A), indicating its inability to understand and analyze complex orbital patterns.

The random forest model was configured with a max depth of 2 and a random state of 0. It yielded a validation accuracy of 94.87%. The corresponding confusion matrix suggested its inability to consistently classify OMB and MCA asteroids (Figure 3B). This difficulty suggests that while the random forest model outperformed the logistic regression model in terms of accuracy, like the logistic regression model, it was unable to correctly identify OMB and MCA asteroid characteristics.

The k-nearest neighbors model, initially configured with one nearest neighbor, yielded a validation accuracy of 96.07%. The confusion matrix suggested that the model struggled with classifying MCA and Apollo (APO) asteroids when compared to the other orbital types (Figure 3C). Increasing the number of nearest neighbors led to a decrease in validation accuracy, with three, nine, and fifteen nearest neighbors yielding validation accuracy scores of 94.04%, 91.80%, and 90.68% respectively. This decrease could indicate a potential trade-off between complexity and accuracy. Unlike the logistic regression and random forest models, however, the k-nearest neighbors model successfully classified OMB asteroids but failed to classify AMO and APO asteroids (Figure 3C).

We then explored multilayer perceptron (MLP) classifiers with varying architectures. We created four networks with configurations of (10,10,10), (3,3,3), (10,10,10,10,10), and (5,5,5,5,5). These networks achieved accuracy scores of 79.00%, 77.29%, 88.95%, and 96.93%, respectively. All four neural networks demonstrated the capability of classifying OMB, TNO, interior earth objects (IEO), and HYA asteroids (Figure 3D). There were no serious flaws in the confusion matrices generated by the MLP neural networks; however, there were still a few minor errors, as shown in the confusion matrix of the highest accuracy model (Figure 3D)

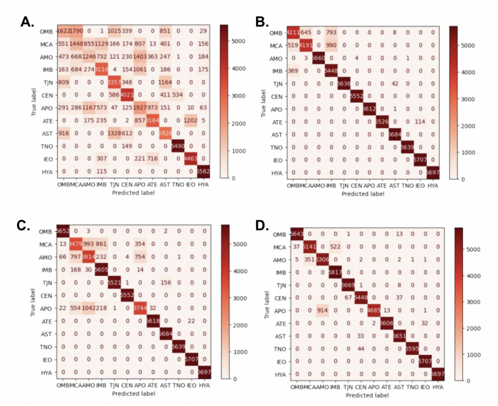

Figure 3. Basic classification model confusion matrices. The confusion matrices displaying the results of the logistic regression (A), random forest (B), k-nearest neighbors (C), and highest validation accuracy yielding MLP neural network (D) models. The matrices were generated by passing the models' respective predictions into a function that displays the true and false positives and negatives

Finally, we created a custom neural network using Keras and employed hyperparameter sweeps to optimize its configuration. The model yielding the highest validation accuracy had 5 hidden layers, 16 nodes per layer, 149 epochs, a batch size of 1000, and the activation function ReLU. It achieved a validation accuracy of 99.18% and had an almost perfectly diagonal confusion matrix (Figure 4). This model was able to accurately classify asteroid orbits and discern between complex orbital patterns.

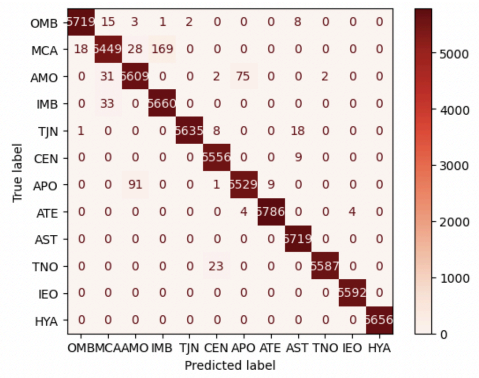

Figure 4. Keras neural network confusion matrix. The confusion matrix displaying the results of the custom Keras neural network. The matrix was generated by passing the model's predictions into a function that displays the true and false positives and negatives.

We also explored the correlation between hyperparameter values generated by the hyperparameter sweeps and validation accuracy to determine which hyperparameters were conducive to high validation accuracy. Correlation is the linear relationship between the hyperparameter and the validation accuracy (10). The higher the correlation, the higher the validation accuracy, and vice versa. The number of nodes demonstrated the highest correlation, suggesting it played a significant role in attaining a high validation accuracy (Table 1).

Hyperparameter	Correlation Value
Number of nodes	0.733
Number of hidden layers	0.267
Runtime	0.267
Activation function (relu)	0.152
Activation function (tanh)	-0.298
Activation function (sigmoid)	0.147
Epochs	-0.080
Batch Size	-0.112

Table 1. Correlation value table. The correlation between the Keras neural network's hyperparameters and validation accuracy. Correlation is calculated as the linear correlation between hyper parameter and validation accuracy.

The parallel coordinate plot generated by hyperparameter sweep software visually plots the relationship between high validation accuracy runs and hyperparameter values, however, it yielded little information due to the scattered high-validation-accuracy run distribution (Figure 5).

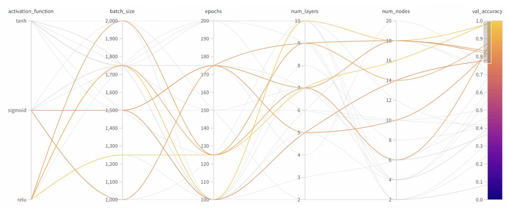

Figure 5. Hyperparameter sweep parallel coordinate plot. The performance of different hyperparameter configurations generated by the hyperparameter sweep. Each run in the sweep was plot and overlayed over the same graph with brighter colors representing higher validation accuracy yielding runs.

CONCLUSION

Our study demonstrates the potential of machine learning techniques paired with SMOTE to classify asteroid orbits despite uneven data and observational bias, contributing to our understanding of orbital dynamics in the solar system. By accurately categorizing asteroid orbits, we can learn about solar system evolution, assess potential hazards, and guide future space exploration efforts.

Despite the success of our models, there are several limitations stemming from the removal of all MBA asteroids. Firstly, main belt asteroids constitute a significant portion of the asteroid population, so our modified dataset may lack a complete representation of orbital dynamics present in the solar system. This incomplete representation could impact the model's ability to classify new orbital types and skew the weights in the neural networks. Features needed for classifying main belt asteroids may be undervalued due to their absence in the dataset, potentially affecting model performance when exposed to unseen data. As a result, the models may not be able to generalize to new data that includes main belt asteroids, limiting their use in real-world applications where they will encounter a broader range of orbits. Reintroducing main belt asteroids could provide a clearer understanding of orbital dynamics but will require faster computational resources.

Future experiments could focus on refining model hyperparameters, incorporating additional orbital features, reintroducing MBA asteroids, or exploring different machine-learning models and data analysis techniques to further improve the validation accuracy. Furthermore, the models could be adapted to also predict the orbital class of comets, satellites, and free floating objects with the sun as the central body, as all three categories share the same orbital features as asteroids.

ACKNOWLEDGEMENTS

I would like to thank my mentor, Erick Ruiz; the Inspirit AI program; and my parents for their help and support during this project.

REFERENCES

1. "Planetary Science." Oxford Research Encyclopedia of Planetary Science. Oxford University Press. DOI: 10.1093$002facrefore$002f9780190647926.001.0001$002facrefore-9780190647926-e-2
2. NASA. "NASA's Asteroid Redirect Mission." Office of Inspector General Report IG-14-030.
3. Britannica. "Spacecraft Exploration."
4. Britannica. "Asteroids in Unusual Orbits."
5. Montenbruck, O., & Guo, J. (2021). Artificial intelligence: A powerful paradigm for scientific research. The Innovation, 2(2), 100104. www.sciencedirect.com/science/article/pii/S2666675821001041
6. Harvard-Smithsonian Center for Astrophysics. (n.d.). Machine learning. Harvard-Smithsonian Center for Astrophysics. pweb.cfa.harvard.edu/research/topic/machine-learning#:~:text=Machine%20learning%20plays%20a%20huge,out%20features%20in%20galaxy%20clusters.
7. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). "SMOTE: Synthetic minority over-sampling technique." Journal of artificial intelligence research, 16, 321-357.
8. Cheng, Wei-Chao, et al. (2024). "From SMOTE to Mixup for Deep Imbalanced Classification." Arxiv. Retrieved from https://doi.org/10.48550/arXiv.2308.15457.
9. Narkhede, Sarang. (2021). "Understanding Confusion Matrix." Medium, Towards Data Science. Retrieved from towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62.
10. "Parameter Importance." Weights & Biases Documentation. Retrieved from docs.wandb.ai/guides/app/features/panels/parameter-importance?_gl=1%2Akhp7f%2A_ga%2AMTg3NTczMjg2NC4xNzAzMTIyMDU5%2A_ga_JH1SJHJQXJ%2AMTcwNzAwMzA2Ni4yMC4xLjE3MDcwMDMwNjguNTguMC4w.
11. Hossain, Mir Sakhawat, and Md. Akib Zabed. (2023). "Machine Learning Approaches for Classification and Diameter Prediction of Asteroids." Proceedings of International Conference on Information and Communication Technology for Development, pp. 43–55. https://doi.org/10.1007/978-981-19-7528-8_4.
12. Hossain, Mir Sakhawat. (2024). "Asteroid Dataset." Kaggle. Retrieved from www.kaggle.com/datasets/sakhawat18/asteroid-dataset.
13. "Mplot3d Tutorial¶." Mplot3d Tutorial - Matplotlib 2.0.2 Documentation. Retrieved from matplotlib.org/2.0.2/mpl_toolkits/mplot3d/tutorial.html.
14. "SBDB (Filter)." NASA. Retrieved from ssd-api.jpl.nasa.gov/doc/sbdb_filter.html.

15. "Sklearn.Neural_network.MLPClassifier." Scikit. Retrieved from scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html.
16. Tanwar, Sanchit. (2022). "Building Our First Neural Network in Keras." Medium, Towards Data Science. Retrieved from towardsdatascience.com/building-our-first-neural-network-in-keras-bdc8abbc17f5.
17. Team, Keras. "Keras Documentation: Whole Model Saving & Loading." Keras. Retrieved from keras.io/api/models/model_saving_apis/model_saving_and_loading/.
18. Weights & Biases. (2020). "🥕 Integrate Weights & Biases with Keras." YouTube. Retrieved from www.youtube.com/watch?v=Bsudo7jbMow.
19. Weights & Biases. (2020). "🧹 Tune Hyperparameters Easily with W&B Sweeps." YouTube. Retrieved from www.youtube.com/watch?v=9zrmUIlScdY.
20. Wumanandpat. (2023). "Asteroid Dataset - Exploration." Kaggle. Retrieved from www.kaggle.com/code/wumanandpat/asteroid-dataset-exploration