Orbit Type Classification with Neural Networks

ACTL3143 T2 2023
Brandon Ngo

Contents

1 Executive Summary

This report aims to assess the performance of two neural network architectures in a multiclass classification problem, specifically to classify the orbit types of asteroids. The two architectures evaluated in this study are a simple sequential dense neural network and a wide neural network.

The report begins by defining the problem statement and providing an overview of the dataset used. Subsequently, thorough data processing and exploratory data analysis are conducted to gain deeper insights into the data and better understand the problem at hand. A benchmark model is used as a reference point, along with a detailed vindication of the two neural network architectures employed. Each model is fitted with hyperparameter tuning and their performance is evaluated. Analysis and insights are gained from the models, along with addressing any ethical concerns related to this study.

Overall, this report offers a comprehensive evaluation of neural networks in the context of a multiclass classification problem and the findings presented herein can serve as a valuable resource for those interested in neural networks.

2 Problem Statement & Data Collection

Asteroids are objects that orbit the Sun, like planets, however much smaller, irregular in shape and composed primarily of rock and metal. They exhibit various types or orbits, where the most well-known are the Main-Belt asteroids that contain the majority of all known asteroids and is classified as asteroids that lie between Mars and Jupiter (NASA, 2022). Near-Earth asteroids are asteroids whose orbits bring them relatively close to Earth's orbit and are of interest for scientific study of potential impact hazards. Trojans are an interesting group of asteroids that share the same orbit as Jupiter. These are just some of the types of asteroid orbits and understanding these orbit types is crucial to comprehend the formation and evolution of the solar system and assess potential asteroid impact risks to Earth. This study aims to evaluate the performance of neural networks in classifying asteroid orbit types.

The data was obtained from Kaggle, which was acquired from the database of the Minor Planet Center. It contains approximately 1,000,000 observations of the orbits of all known asteroids till 3rd February 2021 and has 34 features that describe characteristics of an asteroid's orbit. The data presents a **multiclass classification** problem into 11 distinct classes, some of which are the orbit types discussed earlier.

3 Exploratory Data Analysis

This section provides valuable insight into the context of the data and identifies key features relevant to the problem at hand.

Analysis into the proportion of orbit types in Figure 1 below shows majority of classes belonging to 'MBA' which are the Main-Belt asteroids. This presents an imbalanced class distribution problem which will need to be considered when modelling. To ensure a comprehensive and fair analysis of the data, a variety of appropriate performance metrics will be employed. We can consider using resampling techniques such as oversampling the minority class or by using weights to assign greater importance to the minority class to address the class imbalance issue. Notably, the smallest class is 'Atira', containing only 50 of the ~1,000,000 observations. This presents an interesting challenge to assess how neural networks perform with limited training data for a specific class.

Figure 4 from Appendix A showcases the density distribution of the feature 'n', which represents the mean daily motion of an asteroid. As its name suggests, the mean daily motion is the average movement of a body along its orbit, expressed in degrees per day. The density distribution exhibits multiple distinct peaks for some orbit types. These peaks reflect the different mean daily motions associated with specific classes. This suggests potential significance of this feature to distinguish between different orbit types accurately when constructing the neural network. From Figures 3, 4 and 5, a similar conclusion can be made about the following features: Orbital period, Aphelion distance and Perihelion distance. Orbital period is the time taken for an object to complete one full orbit around the Sun, measured in years. Aphelion distance is the point in the orbit of an object where it is furthest from the Sun and Perihelion distance on the other hand, is the point in the orbit of an object where it is closest to the Sun, both measured in astronomical units.

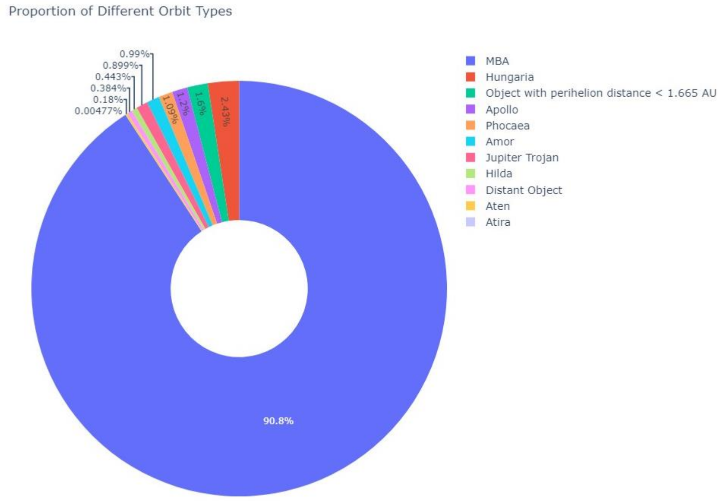

Figure 1: Pie Chart of Orbit Type Proportions

4 Data Preprocessing

This section delves into how the data was cleaned and processed before training the neural network.

Cleaning and data preprocessing was done by exploring variable datatypes and its context. The Boolean features of the data contained either 1 or NaN values, suggesting that zero values were interpreted as NaN from the raw dataset. To address this, NaN values in Boolean features were appropriately converted to zero. Additionally, a new feature was engineered from the variable 'Arc_years,' which originally represented the range between the first and last observations of asteroids in opposition to Earth and the Sun. To make this feature more informative, it was transformed into a new feature representing the integer length between the observations. There were also categorical features which had classes containing one or two observations, causing problems when splitting the data, where these values weren't present in either the training, validation or testing sets. As such, classes with few observations were grouped together. Finally, irrelevant columns pertaining to asteroids' names or identifications were removed from the dataset.

After cleaning, the data was split into training, validation, and testing sets in a 60/20/20 split respectively. Missing values on the numerical features were imputed using median. This choice was made due to the prevalence of right-skewed data. Missing values on the categorical features were imputed using mode. However, one feature presented a unique challenge where it contained both ordinal and nominal values. This feature was the uncertainty parameter, 'U', which is used to quantify the uncertainty when measuring an asteroid's orbit. According to Wikipedia (2023), its values lie in between 0 to 9, with higher values indicating a greater uncertainty. However, this feature also contained letter-codes: 'D', 'E', and 'F'.

'D' – Objects assigned to this letter have insufficient data for orbit determination.

'E' – Objects assigned to this letter denotes orbits for which the eccentricity was assumed, rather than determined.

'F' – Objects assigned to this letter fall in both categories' 'D' and 'E'.

As such, missing values were imputed as 'D' and this feature was treated as a categorical nominal variable. All categorical features were also ordinal encoded and numerical features were standardised for feature scaling. The data transformation processes of imputing, encoding, and scaling were performed after splitting to avoid data leakage from the validation or test sets to inadvertently influence the model during training.

5 Modelling

This section will cover the implementation and evaluation of two different neural network architectures compared to a benchmark in classifying asteroid orbit types. It will discuss the selection of models, their hyperparameters, training process, and performance metrics, providing a comprehensive analysis of each model.

5.1 Benchmark Model

The benchmark model employed in this study was a simple model that assigns all instances of the dataset to the most prevalent class, 'MBA'. This simplistic model essentially captures the baseline performance that any more sophisticated model should aim to surpass. As such, both neural network architectures employed will be compared against this benchmark to gauge their impact to deliver significant results.

5.2 What is a Neural Network?

Neural networks are a type of machine learning algorithm designed to mimic the biological neural networks in the human brain. Simple neural networks can be broken down into three sections:

Input Layer: As with all machine learning algorithms, input data is required to address specific tasks or problems.

Hidden Layer: The input layer is passed through this layer where the data is trained. The hidden layer contains 'neurons' which process and learn from the input. Each neuron in the hidden layer receives the input data where it is multiplied by a weight and a bias term is added. An activation function is then applied to the weighted sum of inputs plus bias term to introduce non-linearity into the output. This neuron output can then be passed on through another hidden layer or onto the output layer. This can be expressed as follows:

$$neuron = f_{activation}\left(\sum \{input_i * weight_i\} + bias\right)$$

Output Layer: This is the desired output of the neural network after being trained through the hidden layers.

What makes neural networks captivating is their ability to learn and improve by adjusting the weights and bias terms to make better predictions. During the training process, the neural network compares its predictions to the output data uses this to update the weights and bias to minimise the difference between predicted and true outputs. This makes them powerful tools for solving complex problems, especially for larger datasets which may have intricate relationships and patterns.

5.3 Sequential Dense Neural Network with Dropout

The first neural network architecture follows a similar structure to the neural network as discussed in Section 5.2. However, an additional component was added to this architecture in the form of a 'dropout layer'. This dropout layer randomly sets a fraction of the neurons to zero during each training iteration, essentially preventing neurons from contributing to subsequent layers. This is a regularisation technique to prevent neural networks from overfitting the data. By introducing randomness during the training process, it reduces the network's dependency on specific neurons to make predictions and allows the neural network to learn generalised representations of the data. Consequently, the neural network is less susceptible to memorising noise in the training data.

Section 3 explores the dataset used in this study and discusses the issues of imbalanced classes in a classification problem. To address this issue, an 'inverse class frequency weighting scheme' was adopted which assigns specific weights to each class. Higher weights are assigned to minority classes and lower weights to the majority class, incentivising the model to balance the impact of different classes and mitigate the issue of class imbalance. The weighting scheme can be expressed as follows:

$$Class\ Weight_i = \frac{Total\ observations}{Observations\ in\ Class\ i * Number\ of\ classes}$$

The output layer used a softmax activation function with 11 (the number of classes) neurons as it converts the output of a neural network into a probability distribution over the output classes, making it suited for classification problems. Additionally, the 'Adam' optimiser was used, coupled with a 'SparseCategoricalCrossentropy' loss function which are suited to multiclass classification problems. Early stopping was employed to prevent overfitting by ending the training process once optimal weights were achieved. Appendix B shows the model architecture.

Hyperparameter tuning was performed using the validation set to control aspects of the learning algorithm and model architecture to help the neural network learn from the data effectively. Bayesian Optimisation was chosen as the hyperparameter tuning technique as it continually adapts and updates its probabilistic model based on the evaluation results to improve its search efficiency when testing parameters. Table 1 below shows the parameters used in the hyperparameter tuning process.

Hyperparameter	**Hyperparameter Space**
Hidden/Dropout Layers	1, 2, 3
Neurons (each layer can have different number of neurons)	16, 32, 48, 64, 80, 96, 112, 128
Activation Function (each layer can have a different activation function)	Relu, Tanh, Sigmoid
Dropout Rate	0, 0.1, 0.2, 0.3, 0.4, 0.5

Table 1: Table of Hyperparameter Space

5.4 Wide Neural Network

The second neural network architecture adopts more complex architecture, introducing embedding layers and multiple input layers to handle diverse feature types. The features of the dataset are split into three distinct inputs: categorial features, Boolean features, and numerical features. Embedding layers turn discrete data into continuous vectors with multiple dimensions. In the embedding space, the relationships within categorical features are captured based on the distance and direction of vectors. In other words, similar vectors that are output from the embedding layer indicate similar categories based on the context of the data. Its main benefits over one-hot encoding are to reduce dimensionality and has the capacity to learn meaningful relationships between different categories.

In this neural network architecture, the categorical features are processed through an embedding layer, and the resulting embeddings are concatenated with the Boolean features. The combined features are then passed through a single hidden layer with dropout. Similarly, the numerical features are fed into their own hidden layer with dropout. Later, the outputs of both hidden layers are merged by concatenation, and the merged features are passed through a final hidden layer with dropout for further processing. Appendix B shows the model architecture.

The idea behind this architecture is that categorical and numerical data are processed separately to leverage unique characteristics of each data type and allow the network to find relationships between the categorical and numerical features that might not be otherwise possible via a fully connected neural network. Recurrent neural networks (RNNs) and convolutional neural networks (CNNs) were also considered for the second network architecture. They were opted out of as RNNs are known to be more suited to handle sequential data where order of data is essential, and CNNs are more suited to visual recognition tasks (Petersson, 2021).

This model shares the same optimiser, loss function and class weight scheme as Model 1 and hyperparameters tested of this neural network architecture were similar, however the only difference being the hyperparameter for hidden/dropout layers was kept fixed.

6 Results & Discussion

6.1 Hyperparameter Tuning

	Tuned Hyperparameters	
Hyperparameter	**Model 1**	**Model 2**
Hidden/Dropout layers	1	-
Neurons	128	80, 48, 128
Activation	Tanh	Relu, Relu, Tanh
Dropout Rate	0	0.4, 0, 0

Table 2: Table of Tuned Hyperparameters. Note: Model 2 hyperparameters are ordered categorical, numerical, and then all features respectively

The tuned hyperparameters for Model 1 are intriguing as they indicate a hidden/dropout layer of 1, which suggests that the data may involve relatively simpler relationships. Additionally, the optimal dropout rate of 0 suggests that the model is not susceptible to overfitting, and the data may not be highly complex. Another interesting observation in Model 2 is hyperparameters for the final hidden/dropout layer matches that of Model 1.

6.2 Performance on Test Set

	Training Set				**Test Set**			
Model	**Accuracy**	**Macro average f1-score**	**Weighted average f1-score**	**Macro average Precision-Recall AUC**	**Accuracy**	**Macro average f1-score**	**Weighted average f1-score**	**Macro average Precision-Recall AUC**
Benchmark	0.9078	0.0865	0.8639	-	0.9086	0.0866	0.8651	-
Model 1	0.9947	0.9670	0.9949	1	0.9943	0.9440	0.9945	0.9555
Model 2	0.9934	0.9379	0.9936	0.9691	0.9930	0.9182	0.9932	0.9300

Table 1: Table of Model Performance

These models were evaluated on the test set to compare its predictive capability on unseen data. F1-score is the harmonic mean of precision and recall and is useful in dealing with imbalanced classification where it provides a balanced measure of both metrics. Both models clearly outperform the benchmark which shows the ability of neural networks to model non-linear data and learn patterns. In contrast, the limitations of the benchmark model are clear, and its high accuracy is primarily attributed to the imbalanced class distribution. Notably, on metrics such as the macro average f1-score, the benchmark model exhibited poor performance.

Comparing the two neural networks, the sequential dense model outperforms the wide model in every metric. As highlighted in Section 6.1, this observation reaffirms the idea that the underlying data may not involve complex relationships, and simpler neural networks have proven to be more effective for capturing relationships in an orbit type classification problem. As such, we can conclude that the simple dense sequential neural network is the best model to accurately classify orbit types.

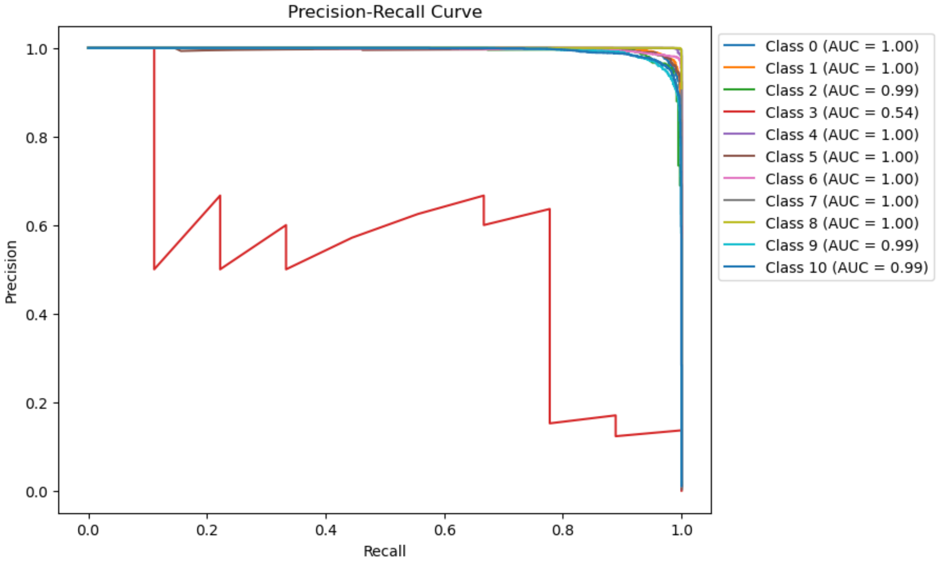

Figure 2: Precision-Recall Curve of Model 1 on Test Set

Despite the great performance of Model 1, a precision-recall curve reveals that it exhibits weaker performance for Class 3 observations, corresponding to 'Atira' as discussed in Section 3. This class comprises the least number of observations in the dataset. The relatively weaker performance on Atira is likely attributed to the limited availability of training observations for this specific class. Neural networks are known to perform better when trained on larger datasets, where ample data allows them to learn more robust representations and patterns.

7 Ethical Concerns & Limitations

Though neural networks have demonstrated significant advancements in various fields, including computer vision and natural language processing, they come with their own set of ethical concerns and limitations, especially more so in this study to predict asteroid orbit types. One primary concern is the difficulty to interpret neural networks, making it challenging to understand the relationships between variables and the decision-making process. Another limitation demonstrated in this study is the need for vast amounts of data to effectively train and make predictions, making neural networks computationally expensive (Donges, 2021).

As discussed in Section 6.2, the underlying data likely does not involve complex relationships. Though the networks performed well, this raises the question of whether other machine learning algorithms, such as decision trees, could achieve similar performance while being more interpretable and computationally efficient. This can be explored in future studies by considering other machine learning techniques to function as the benchmark model. Moreover, the underlying dataset mainly contained features directly related to asteroid orbit types, such as distance from the Sun, orbital period, and mean daily motion (these features were the most important under a permutation importance algorithm, shown in Appendix D). Exploring a dataset with inputs related to non-orbit type variables such as asteroid composition, size, spectral properties, albedo, and spectral features could potentially leverage the depth and complexity of neural networks to find intricate relationships.

Other ethical concerns and limitations included outdated data (data contains all observations up to February 2021), and possible data corruption. Finally, neural networks, more specifically AI, have inherent ethical concerns relating to data privacy, lack of transparency and security risks (Pazzanese, 2020).

8 Conclusion

This study explored the performance of two neural network architectures in classifying asteroid orbit types. These two neural network architectures were a simple dense sequential model, and a wide neural network with multiple inputs. Each model was hyperparameter tuned and performance was compared to a benchmark model, with both neural networks showing significant results over the benchmark. The best performing model for this study was a simple dense sequential model, showing 99.43% accuracy on an unseen test set. Its performance is likely attributed to the nature of the dataset, which likely contains simpler relationships between variables.

9 References

Bex, T 2021, *Comprehensive Guide to Multiclass Classification Metrics*, Towards Data Science, accessed 19 June 2023, <https://towardsdatascience.com/comprehensive-guide-on-multiclass-classification-metrics-af94cfb83fbd>

Davis, P & Carney, S 2023, *Asteroids*, The National Aeronautics and Space Administration, accessed 25 July 2023, <https://solarsystem.nasa.gov/asteroids-comets-and-meteors/asteroids/overview/?page=0&per_page=40&order=name+asc&search=&condition_1=101%3Aparent_id&condition_2=asteroid%3Abody_type%3Ailike>

Donges, N 2021, *Pros and Cons of Neural Networks*, Experfy, accessed 25 July 2023, <https://resources.experfy.com/ai-ml/pros-and-cons-of-neural-networks/>

Kafritsas, N 2022, *Tune Deep Neural Networks using Bayesian Optimization*, Towards Data Science, accessed 19 June 2023, <https://towardsdatascience.com/tune-deep-neural-networks-using-bayesian-optimization-c9f6503a049f>

Minor Planet Center 2023, International Astronomical Union, accessed 19 June 2023, <https://minorplanetcenter.net/>

Pazzanese, C 2020, *Ethical concerns mount as AI takes bigger decision-making role in more industries*, Harvard, accessed 25 July 2023, <https://news.harvard.edu/gazette/story/2020/10/ethical-concerns-mount-as-ai-takes-bigger-decision-making-role/>

Petersson, D 2021, *CNN vs. RNN: How are they different?*, accessed 25 July 2023, <https://www.techtarget.com/searchenterpriseai/feature/CNN-vs-RNN-How-they-differ-and-where-they-overlap>

Saifuddin, M 2022, *Stellar Classification: A Machine Learning Approach*, Towards Data Science, accessed 19 June 2023, <https://towardsdatascience.com/stellar-classification-a-machine-learning-approach-5e23eb5cadb1>

Shandilya, S 2021, *Orbit Data for All Known Asteroids in MPC Database*, Kaggle, accessed 19 June 2023, <https://www.kaggle.com/datasets/shivamshandilya/orbit-data-for-all-known-asteroids-in-mpc-database>

Uncertainty parameter 2023, Wikipedia, accessed 10 July 2023, <https://en.wikipedia.org/wiki/Uncertainty_parameter#:~:text=The%20uncertainty%20parameter%20U%20is,mean%20anomaly%20after%2010%20years.>

10 Appendix

A - Exploratory Data Analysis

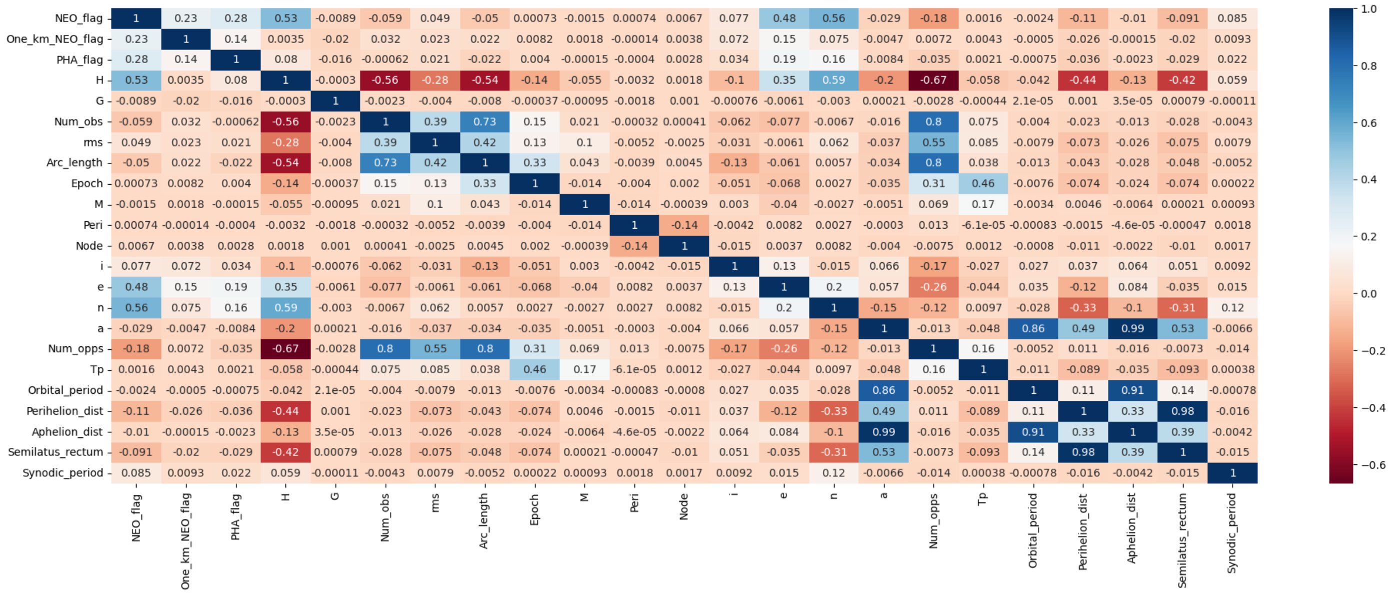

Figure 3: Correlation Heatmap of all numerical features

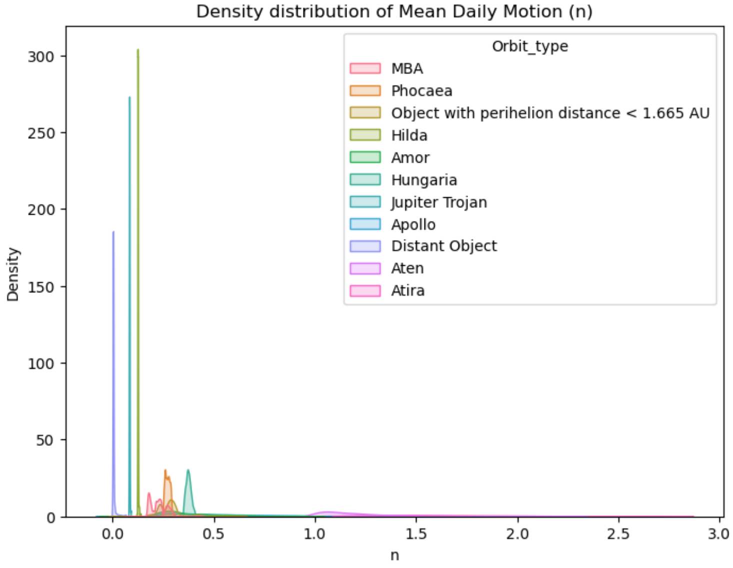

Figure 4: Density Distribution of Mean Daily Motion (feature 'n')

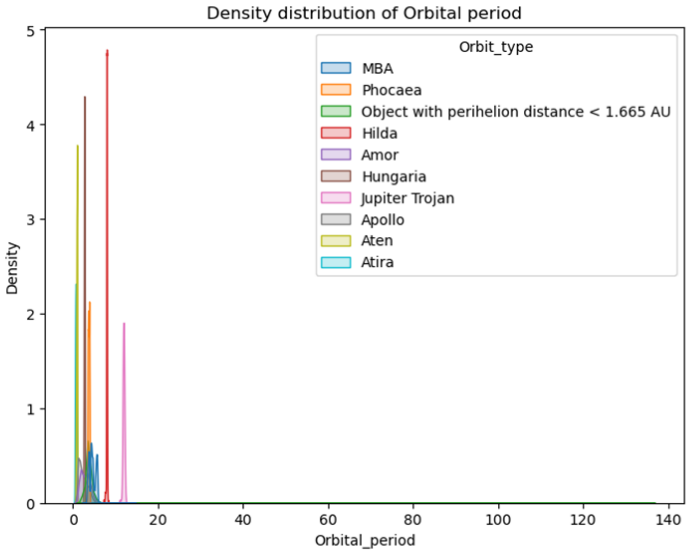

Figure 5: Density Distribution of Orbital Period

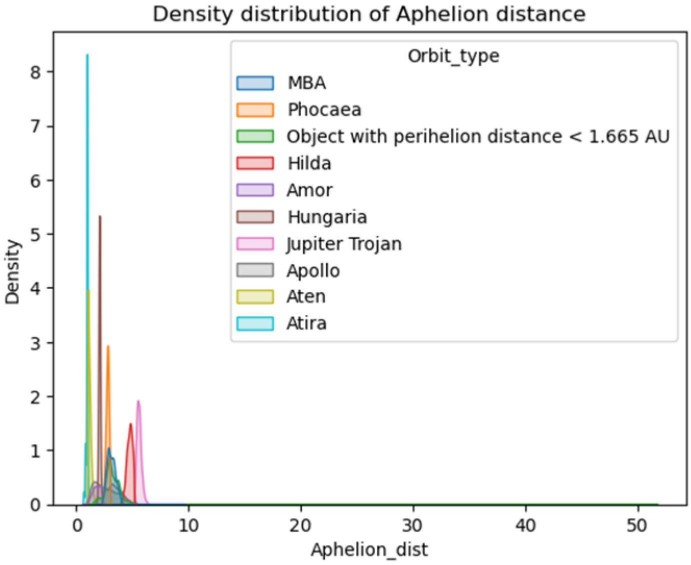

Figure 6: Density Distribution of Aphelion Distance

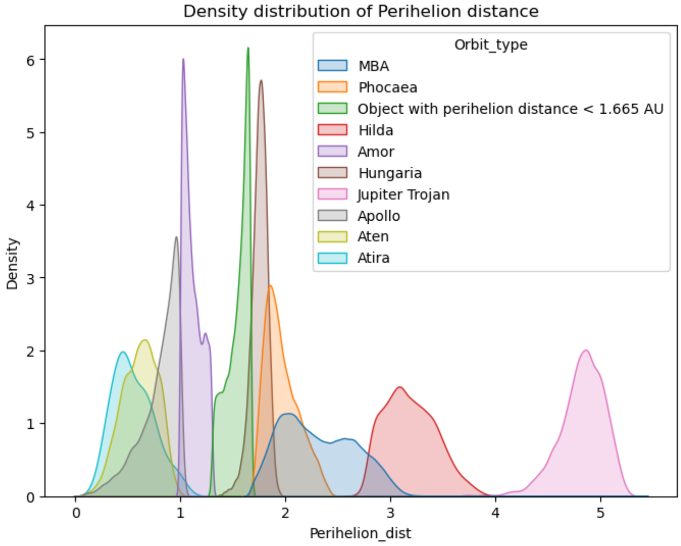

Figure 7: Density Distribution of Perihelion Distance

B – Model Architecture

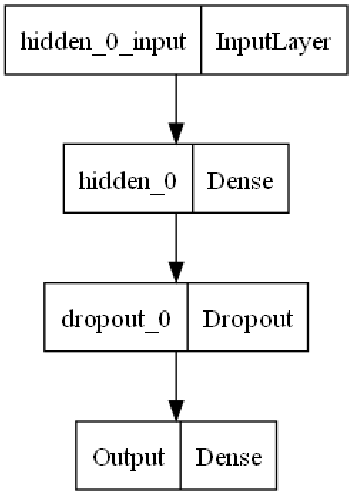

Figure 8: Neural Network Architecture of Model 1

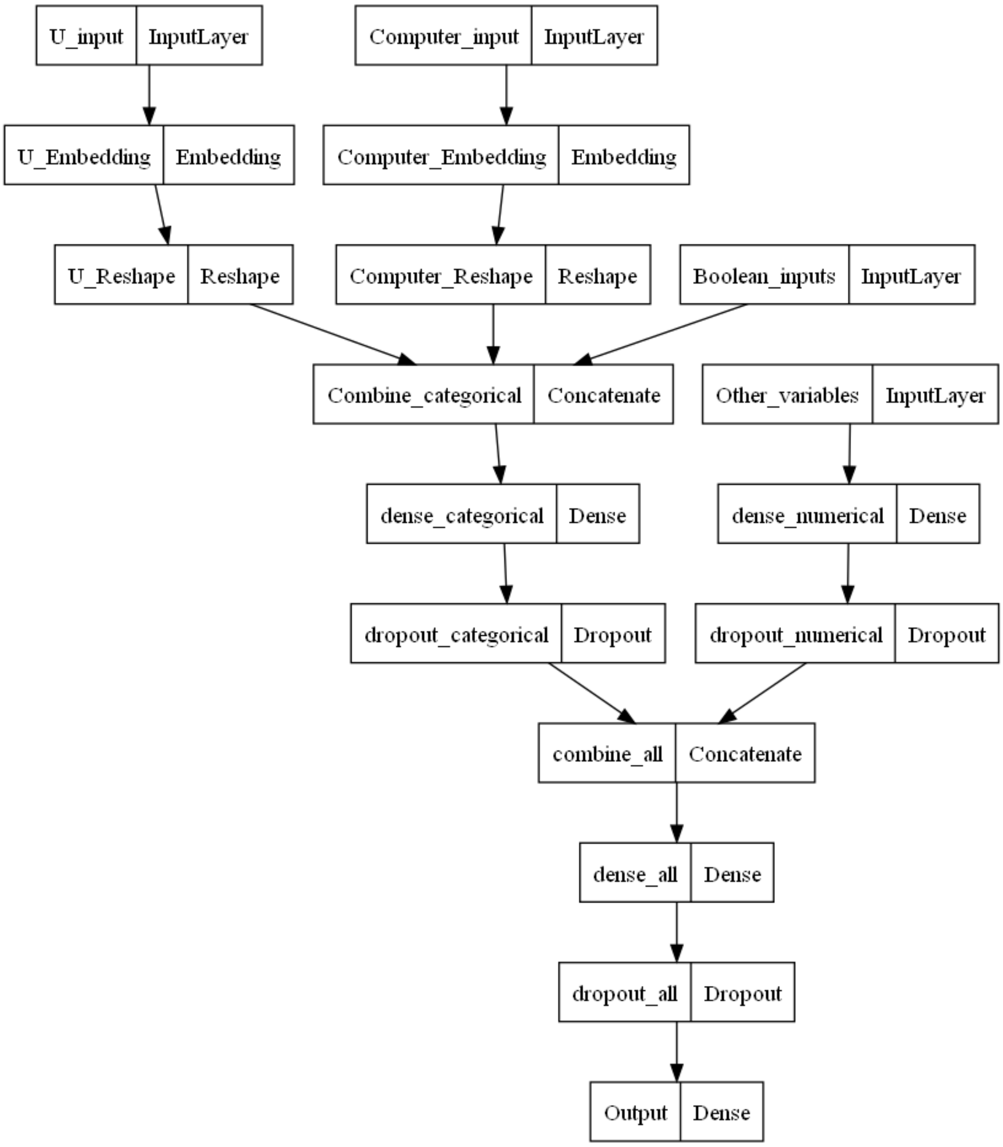
U_input | InputLayer
Computer_input | InputLayer
U_Embedding | Embedding
Computer_Embedding | Embedding
U_Reshape | Reshape
Computer_Reshape | Reshape
Boolean_inputs | InputLayer
Combine_categorical | Concatenate
Other_variables | InputLayer
dense_categorical | Dense
dense_numerical | Dense
dropout_categorical | Dropout
dropout_numerical | Dropout
combine_all | Concatenate
dense_all | Dense
dropout_all | Dropout
Output | Dense

Figure 9: Neural Network Architecture of Model 2

C - Test Set Results

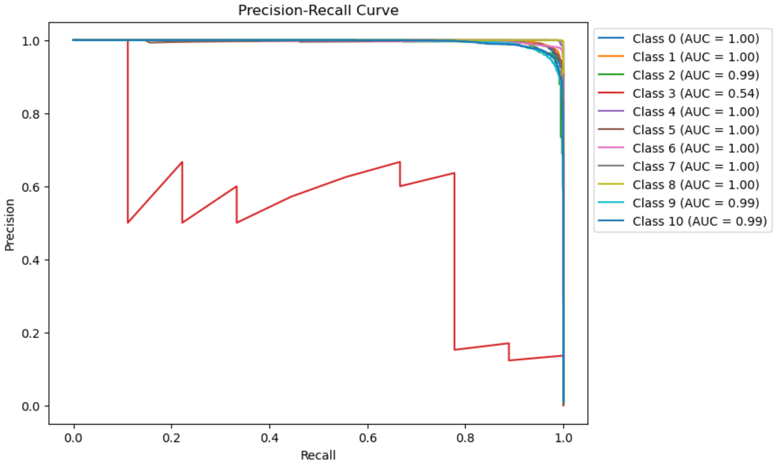

Figure 10: Precision-Recall Curve of Model 2

D – Permutation Importance

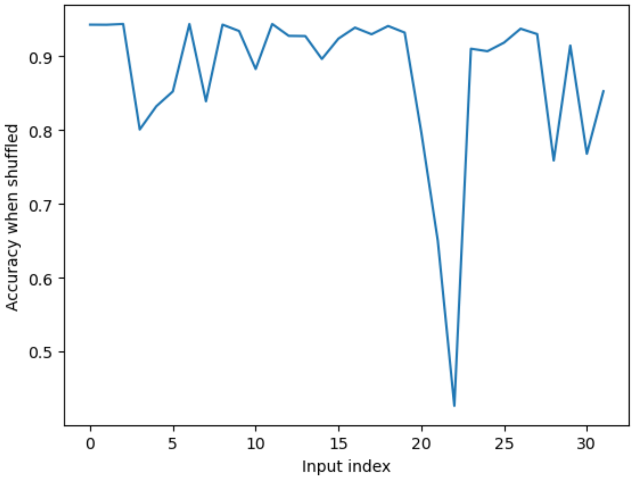

Figure 11: Permutation Importance of Model 1. The five best features were 'n', 'e', 'Perihelion Distance', 'Semilatus_rectum', 'I'.

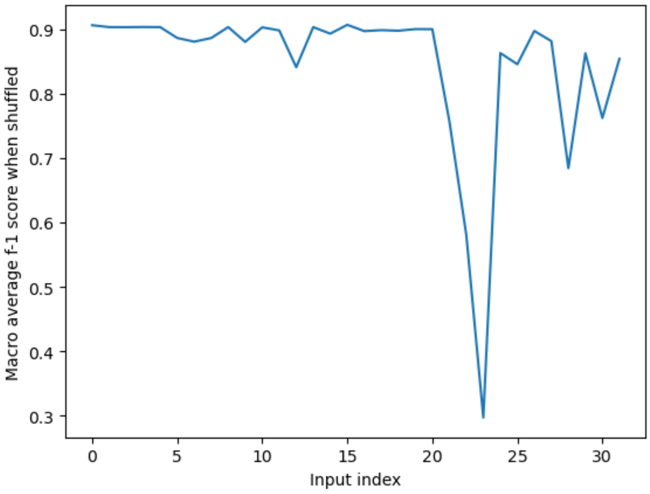

Figure 12: Permutation Importance of Model 2. The five best features were 'n', 'e', 'Perihelion Distance', 'I', 'Semilatus_rectum'.