Dangerous Asteroid Model Report

Tam Nguyen
Student ID
University of Houston
Houston, TX, USA
ttngu454@cougarnet.uh.edu

Eduardo Gamez
Student ID
University of Houston
Houston, TX, USA
edgamez@cougarnet.uh.edu

Merdi Mukenge
Student ID
University of Houston
Houston, TX, USA
mnmukeng@cougarnet.uh.edu

1. INTRODUCTION

There are many asteroids in the vast universe. These asteroids orbit the universe with the help of gravity in 3D space. Some of these asteroids pose a threat to Earth, as they are close or intersects Earth's orbital route. The question is, which asteroids are considered dangerous to Earth? To solve this question, we will be using the Asteroid Dataset provided by NASA Jet Propulsion Laboratory, and edited by Mir Sakhawat Hossain from Kaggle consisting of 958k observations. The attributes we will use will be attributes correlated to the variable PHA. Potentially Hazardous Asteroids is a classification with "Y" or "N". Using this dataset, we can make a model to potentially figure out whether an asteroid is classified as PHA or not based on the attributes of an asteroid. The asteroid dataset contains names, ID, classification variables, and many numerical variables. We will specifically be using numerical variables and classification variables turned numerical. The dataset also contains many NA or NULL values, as these asteroids are newly identified or pose no threat at all. We will ignore these values for the sake of creating our models. The variables used will be listed below, including the response variable PHA with a short explanation.

neo = Near Earth Object flag (Y/N)

pha = Potentially Hazardous Asteroid (Y/N)

h = absolute magnitude (Visual magnitude of asteroid if it was exactly 1 au distance from the observer and 1 au from the Sun at zero angle. More negative means brighter. The Sun is -26.74. Brighter asteroid, means bigger asteroid size, as more surface area to reflect light)

diameter = Object diameter in KM

albedo = ratio of light received by a body to the light reflected by that body (0 is pitch black, 1 is perfect reflector) (albedo with absolute magnitude can help determine size of asteroid)

e = Eccentricity of the orbit (Parameter that describes the orbit's ellipse, 0 is circular, 1 is parabolic, between 0 and 1 is elliptic)

a = Semi-major axis of the orbit (Mean distance of asteroid from the Sun)

q = perihelion distance (An orbit's closest distance to the Sun)

i = inclination of the asteroid's orbit in degrees (respective to the ecliptic plane, the plane we set all plants and Earth to orbit on)

per_y = Orbital period in years

moid = Minimum Object Intersection Distance (minimum distance between asteroids and Earth's orbit)

class = Classification of object's orbit (ex: AMO, APO, ATE, ETC) Each different classification indicates the asteroid's

location in our solar system based on the distance au.

Note: distance au is equal to approximately 150 million Kilometers.

2. METHODOLOGY

In our pursuit to accurately classify a Potentially Hazardous Asteroid (PHA) based on its attributes, we have employed two well-established machine learning algorithms: K-Nearest Neighbors (KNN) and Support Vector Machines (SVM). Both of these methods have gained extensive acknowledgment and application in the field of classification tasks. In the following sections, we explore their strengths while also discussing their respective limitations in the context of our dataset. By doing so, we gain a deeper understanding of how these methods can effectively contribute to our goal of identifying PHAs.

2.1 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a fundamental machine learning algorithm known for its ability to classify observations based on the majority class among its K closest neighbors. This versatile method has been chosen as one of the primary predictive models for our analysis due to its simplicity, flexibility, and efficacy in a wide range of applications. KNN operates on the intuitive principle that objects with similar attributes tend to belong to the same class or category. By using this principle, KNN gives us a tool in our quest for precise classification. In the next section, we explore the inner workings of the KNN algorithm, by discussing its strengths, weaknesses, and how it aligns with our mission to identify Potentially Hazardous Asteroids (PHA) based on their attributes.

2.1.1 KNN Advantages:

I. Simplicity: It is easy to understand and implement. It does not make strong assumptions of the underlying data distribution.
II. Flexibility: It is versatile and can handle various types of data. For example, it can handle binary and multi-classification problems.
III. Interpretability: An observation is classified based on the labels of its K nearest neighbors, therefore predictions made by the model can be easily interpreted.

2.1.2 KNN Disadvantages:

I. Computational Cost: Depending on the size of the dataset, KNN can be computationally expensive.
II. Sensitivity: Due to KNN being sensitive to outliers, the calculated distance can be affected consequently affecting the results.
III. Need for Feature Scaling: KNN relies on the distance metric to determine the nearest neighbors. When features have different scales, the ones with larger values will dominate the calculation. Because of this, data scaling is often necessary.

2.2 Support Vector Machine (SVM)

Support Vector Machines (SVM) stands out as a powerful machine learning algorithm. SVM seeks to find an optimal hyperplane that best separates the data into different classes, with the goal of maximizing the margin between the classes. This choice is attributed to SVM's capability to adeptly address intricate data distributions, making it a great tool in high-dimensional spaces. The algorithm's core objective of identifying an optimal decision boundary aligns with our mission of identify Potentially Hazardous Asteroids (PHA) based on their attributes. Similarly to KNN, in the subsequent section, we delve deeper into the strengths,

weaknesses, and how it aligns with our overarching objective.

2.1.1 SVM Advantages:

I. High Dimensions: SVM can handle datasets with large numbers of features. This makes it suitable for complex problems.
II. Robust against Overfitting: SVM is able to control overfitting by using a regularization parameter "C". This enables it to perform effectively on unseen data.
III. Kernel Trick: SVM can work with non linear data by applying kernel functions. Some of these can turn the data into a higher dimensional space.

2.1.2 SVM Disadvantages:

I. Computational Complexity: SVM involves solving a convex optimization problem, so with large datasets the training can be computationally expensive.
II. Sensitivity to hyperparameters: The performance of SVM is sensitive to the choice of hyperparameters. Tuning these is essential for optimal results.
III. Memory Usage: SVM models may require significant memory to store support vectors, especially in high dimensional spaces.

In summary, we have chosen K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) algorithms for our analysis. This choice is based on a careful consideration of our goal of identifying Potentially Hazardous Asteroids (PHA) based on their attributes. These two models together offer a powerful and complementary approach to achieving our mission of asteroid hazard assessment.

3. DATA ANALYSIS

In order to create a model to predict whether an asteroid is PHA or not, we must first clean the dataset. The dataset has a lot of NA or NULL values, and some categorical values. In our KNN and SVM models, they are doing classification on the PHA categorical value. So we will omit the NA/NULL values and convert the other categorical values. This reduces us to 131k observations. The categorical variable, neo, is converted into a binary, while the class variable is mapped from 1 to 10, indicating the asteroid's class. Scaling the data was necessary as some variables were ratios, while others were in thousands.

For unused variables, they are not as relevant to our model in predicting PHA. The following unused variables are as follows:

id, spkid, full_name, pdes, name, prefix, are variables that give the identification to the asteroid. They do not provide meaningful insight to calculations or classification as they are unique to the asteroid.

epoch, epoch_mjd, epoch_cal, equinox, are specific time, date, or reference when the asteroid's orbital path was recorded or calculated based off of.

w, ma, ad, n, tp, tp_cal, per, moid_id, all sigma_values of used/unused variables, are miscellaneous properties that are not as correlated to PHA. sigma_values are measuring uncertainty or deviation of orbital path, they are not as important.

3.1 KNN OPTIMAL TUNING

The cleaned dataset is subdivided into 80% training and 20% testing. The KNN model was first trained against the training set, then tested against the test set for accuracy and error rate. In testing, we tried K values from 1 to 100 with K = 5 and K = 35 performing the best and the same. The accuracy is

0.9988562, and the error rate is 0.001143816.

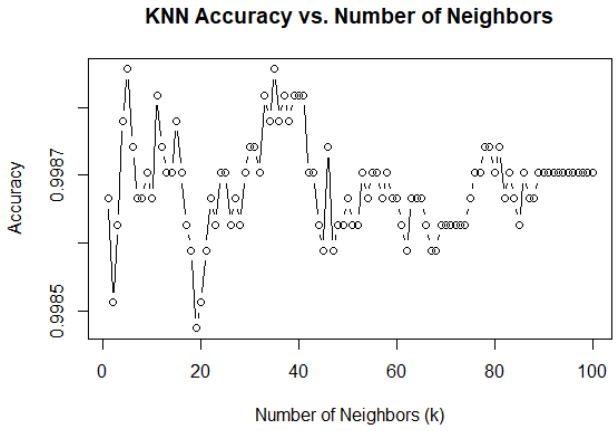

Figure 1: K accuracy vs K neighbors. Accuracy = sum(knn_model == Y_test) / length(Y_test)

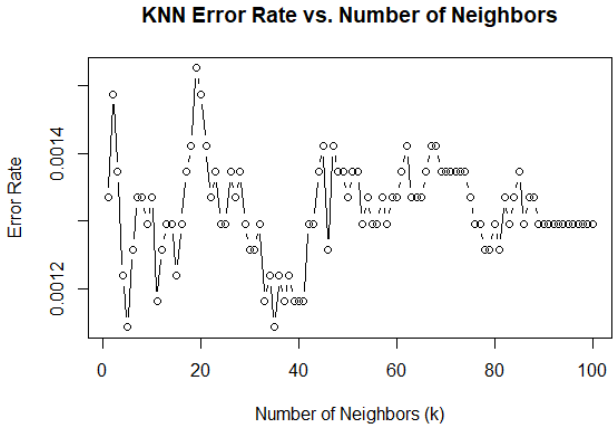

Figure 2: K Error Rate vs K neighbors Error Rate = mean(knn_model != Y_test)

3.2 KNN AND FULL DATASET

With our K values established, we then tested our model against the entire clean dataset. Using K = 35, the result is an accuracy of 0.9988256, and error rate of 0.0011743.

Summary:

```
> print(summary(knn_model_full))
     0      1
131061     75
```

Figure 3: KNN model with full clean dataset.

3.3 SVM OPTIMAL TUNING

Just like KNN, the SVM also uses the same training set and test set splits. The only difference is SVM changing the PHA to a categorical variable of "0" or "1". SVM was tested for 3 kernels, linear, polynomial and radial. Each kernel is provided with their optimal tuning values, and their accuracy and error rate for both train and test.

3.3.1 SVM LINEAR

Support Vector Machine using kernel linear, was tested using costs sequentially ranging from 1 to 100. The best cost was 65. The best accuracy and error rate are as follows:

Accuracy Train: 0.9997140
Accuracy Test: 0.9996187
Error Rate Train: 0.0002859
Error Rate Test: 0.0003812

Testing the SVM linear kernel model to the clean dataset, resulted in accuracy of 0.09996949, and error rate of 0.0000305.

```
> table(pred=train_pred.linear, true=train_data$pha)
    true
pred      0      1
   0 104749     18
   1     12    129
```

Figure 4: SVM Linear model with training dataset.

```
> table(pred=model_pred, true=clean_data$pha)
    true
pred      0      1
   0 130939     24
   1     16    157
```

Figure 5: SVM Linear model with full clean dataset.

3.3.2 SVM POLYNOMIAL

Support Vector Machine using kernel polynomial, was tested using costs 0.01, 0.1, 1, 5, 10, and 100 with degrees 2, 3, 4. The

best cost was 100 with degree 2. The best accuracy and error rate are as follows:

Accuracy Train: 0.9998379
Accuracy Test: 0.9995043
Error Rate Train: 0.0001620
Error Rate Test: 0.00004956

Testing the SVM polynomial kernel model to the clean dataset, resulted in accuracy of 0.9998322, and error rate of 0.0001677.

```
> table(pred=train_pred.poly, true=train_data$pha)
    true
pred      0      1
   0 104754     10
   1      7    137
```

Figure 6: SVM Polynomial model with training dataset.

```
> table(pred=model_pred, true=clean_data$pha)
    true
pred      0      1
   0 130946     13
   1      9    168
```

Figure 7: SVM Polynomial model with full clean dataset.

3.3.3 SVM RADIAL

Support Vector Machine using kernel radial, was tested using only costs 5 and with gamma 0.5. It is computationally expensive. We would've liked to do the hyperparameters in the R file which are cost 1, 5 with gamma 0.1, 0.25, 0.5, 0.75. The best accuracy and error rate are as follows for cost 5 with gamma 0.5:

Accuracy Train: 0.9999618
Accuracy Test: 0.9988561
Error Rate Train: 3.81286e^-05
Error Rate Test: 0.0011438

Testing the SVM radial kernel model to the clean dataset, resulted in accuracy of 0.9999237, and error rate of 7.62567e^-05.

```
> table(pred=train_pred.radial, true=train_data$pha)
    true
pred      0      1
   0 104760      3
   1      1    144
```

Figure 8: SVM Polynomial model with training dataset.

```
> table(pred=model_pred, true=clean_data$pha)
    true
pred      0      1
   0 130953      8
   1      2    173
```

Figure 9: SVM Polynomial model with full clean dataset.

3.5 KNN AND SVM COMPARISON

Comparing the best KNN model to the best SVM model, SVM did the best. Despite all models having a very high accuracy of 99%, SVM radial kernel was the only one that performed the best out when it came to testing against the entire dataset.

There could be some improvement in the SVM radial model, like testing more tuning hyperparameters to get the best fit.

3.6 INTERPRETATION

Based on the results, we can say our model did fairly decent in classifying the asteroids. The models generally achieve 98% accuracy. It did predict correctly the majority of asteroids that are PHA Y being ####%. It does solve our research question of creating a model that is able to correctly predict PHA based on an asteroid's properties. Part of the reason why the model has such a high accuracy is because of how the dataset is structured. There are more PHA N asteroids than there are PHA Y asteroids. The reasoning for this is because there are generally more asteroids outside of Earth's orbital range. The outside asteroids classes above class AMO or asteroids not classified as a Near-Earth-Object. There are 7 of these classes that are not in the NEO classification. This is important to note, as we could have specified a certain au distance from Earth, to consider in our data computations. This would've eased

computation times, while sacrificing un-important asteroids that would not hit us at all. With a smaller sample size of PHA N, we could get a more accurate accuracy and error rate in relation to PHA Y.

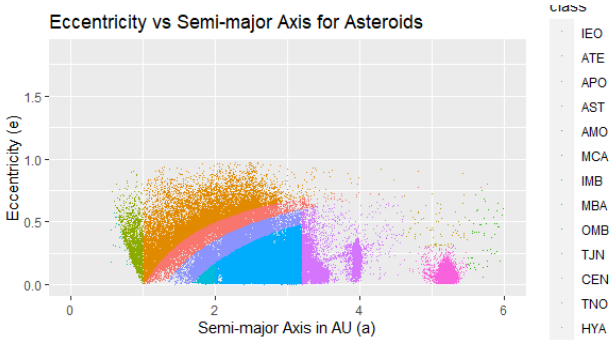

Figure 4: Graph depicting asteroids classes based on mean distance from the sun. Left is closest to the sun, right is farthest.

4. CONCLUSION

As mentioned in interpretation, we accomplished our goal of creating a predictive model that classifies an asteroid as PHA or not based on its attributes. Some difficulties experienced are domain knowledge and understanding of the dataset. Eventually the data set is understood, but it took some time considering we are not astrophysicists. We realized too late that we could have reduced the amount of observations needed which would greatly reduce the computation time needed as mentioned in the interpretation. Especially since the SVM radial kernel takes the longest to compute.

5. REFERENCES

[1] *Mir Sakhawat Hossain. 2023. Asteroid dataset. (December 2023).* Available at: https://www.kaggle.com/datasets/sakhawat18/asteroid-dataset/ (Accessed: 10 September 2023).

[2] Anon. NASA Jet Propulsion Laboratory (JPL) - robotic space exploration. Available at: https://www.jpl.nasa.gov/ (Accessed: 15 November 2023)

[3] Anon. Small-body database query. Available at: https://ssd.jpl.nasa.gov/tools/sbdb_query.html (Accessed: 15 November 2023)

[4] Anon. Diagrams and charts. Available at: https://ssd.jpl.nasa.gov/diagrams/elem_dist.html (Accessed: 15 November 2023)

CONTRIBUTIONS

Tam Nguyen — Creation of Models. Data Analysis. Optimization Hyper Parameter Tuning. Introduction. Conclusion.
Eduardo Gamez - Methodology, assisted with Data Analysis and creation of Models.
Merdi Mukengy - References, assisted with Introduction and Model ideas.

APPENDIX

A.1 Introduction

A.2 Methodology

A.3 Data Analysis

A.3.1 Knn optimal tuning

A.3.3 Svm optimal tuning

A.3.3.1 Svm linear

A.3.3.2 Svm polynomial

A.3.3.3 Svm radial

A.3.5 Knn and Svm comparison

A.3.6 Interpretation

A.4 Conclusion