

Contents

Introduction	2
Transformation of variables	2
Splitting into training set, validation set and test set	2
Linear kernel	3
Different kernels	3
Grid search	4
RBF	4
Polynomial kernel	5
Laplacian kernel	5
Exponential kernel	6
Sigmoid kernel	7
Bessel kernel	8
Vizualization	9
Conclusion	10

Introduction

In this project, we aim to classify hazardous and non-hazardous asteroids using Support Vector Machines (SVMs) and kernel SVMs. First, we will evaluate the linearity of the data by performing a linear SVM. We will then use a non-linear SVM and optimize the model's parameters using cross-validation. We will report the number of support vectors, the value of the objective function, and the proportion of correct classifications. To interpret our results, we will visualize them by projecting the data into the first two principal components.

SVMs are a popular classification model in machine learning, which maximizes the margin between two classes by finding the hyperplane that separates them. Kernel SVMs transform the data into a higher-dimensional feature space where it becomes linearly separable. We will use the dataset collected by CNEOS at the JPL of NASA [<https://cneos.jpl.nasa.gov/>], containing 4687 observations and 40 variables that include physical and orbital properties of the asteroids. The target variable “Hazardous” indicates whether an asteroid is potentially hazardous to Earth.

Our aim is to build an accurate model that predicts whether an asteroid is hazardous or not based on its physical and orbital characteristics using SVM and kernel SVM (All variables and their description will be given in annex) . The choice of kernel function is crucial for the performance of the model, and we will experiment with different kernel functions such as linear, polynomial, and RBF kernels to choose the best one. This project will provide insights into the capabilities and limitations of SVMs and kernel SVMs in classification tasks.

Transformation of variables

In the given code, we first read in the NASA dataset using the `read.csv` function. Then we remove some variables that are not necessary because of redundancy (just a change of the measure meter to kilometer to feet etc...) After that, we transform the binary target variable, “Hazardous,” by converting its values to 0 and 1 using the `ifelse` function.

```
path = "~/LDATS2470 Statistical Machine Learning" # Change here the path to the csv file
nasa <- read.csv("nasa.csv")
nasa = subset(nasa, select = -c(Orbiting.Body, Neo.Reference.ID, Equinox,Orbit.Determination.Date,Closeness)

nasa$Hazardous = ifelse(nasa$Hazardous, 1, 0)
```

Splitting into training set, validation set and test set

The code above splits the dataset into training, validation, and test sets. The training set comprises 60% of the data, while the remaining 40% is split equally between the validation and test sets. The function `createDataPartition()` from the `caret` package is used to randomly partition the data while ensuring that the distribution of the target variable (Hazardous) is preserved in each subset. The resulting subsets are stored in the `train`, `test`, and `val` data frames, respectively.

Linear kernel

The purpose here is to determine if the dataset is linearly separable. If the dataset is not linearly separable, it will be necessary to use a non-linear kernel for classification. By performing a linear SVM, we will be able to establish a baseline performance that we can use to compare the performance of the non-linear SVM. We will report the number of support vectors, the value of the objective function, and the proportion of correct classifications as evaluation metrics for the linear SVM.

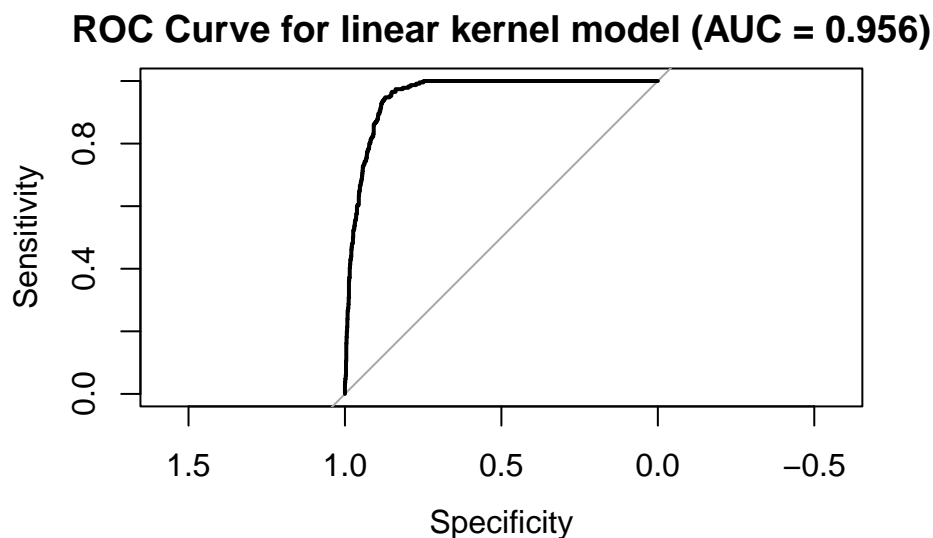
Setting default kernel parameters

```
[1] "the number of support vectors is 1085"
```

```
[1] "the objective function value is -2204.74476464069"
```

	Reference	
Prediction	0	1
0	1174	220
1	1	10

```
[1] "The proportion of correct classifications is 0.843 %"
```



We can see that we have an accuracy of 0.8427046 which is the number of correct classification. Although the data is not linearly separable, the linear kernel still produces reasonable results.

Different kernels

When we use a kernel function in SVM, we map the input data into a higher-dimensional feature space where it becomes linearly separable. This means that instead of working with the original features of the data, we transform them into a new space with potentially infinitely many dimensions. The idea behind this is that it may be easier to find a hyperplane that separates the classes in this new space.

The choice of kernel function determines the shape of the feature space and has a significant impact on the performance of the SVM model. A kernel function is a measure of similarity between two points in the

original space, which is used to compute the inner product between the corresponding transformed points in the feature space. By using different kernel functions, we can transform the data into feature spaces with different properties. All the kernel shape will be put in annex.

We will use a grid search to tune our model and choose the best value for each hyperparameter. The function will be put in the annex.

Grid search

RBF

The RBF kernel is a commonly used in svm classification kernel defined by :

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

the RBF kernel maps the data into a feature space where the distances between points are weighted by a Gaussian function centered on each point. This has the effect of giving more weight to nearby points, which can be useful for modeling non-linear decision boundaries.

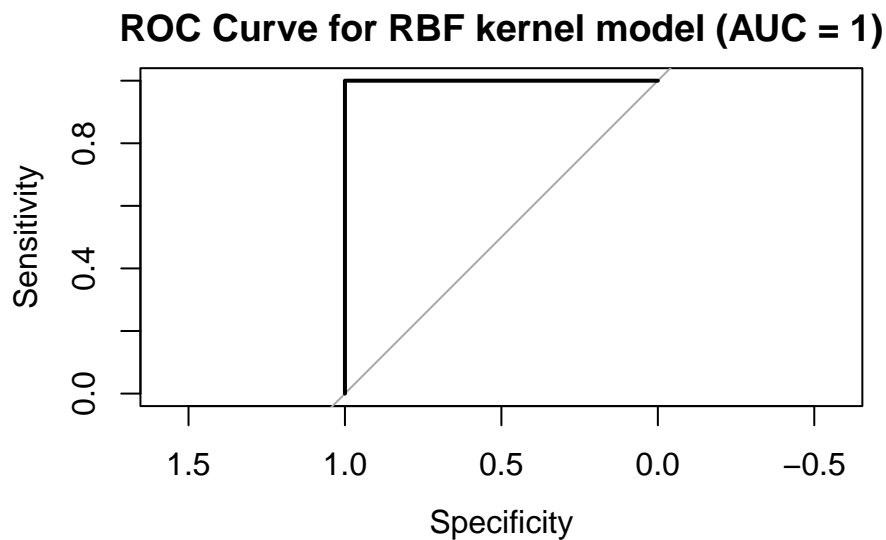
```
[1] "RBF : The proportion of correct classifications is 1 ; Sigma = 0.6"
```

```
[1] "the number of support vectors is 1405"
```

```
[1] "the objective function value is -429.506831249868"
```

	Reference	
Prediction	0	1
0	1175	0
1	0	230

```
[1] "The proportion of correct classifications is 1"
```



Polynomial kernel

The polynomial kernel can be written as $K(x, y) = (x^T y + c)^d$, where x and y are the input data points, d is the degree of the polynomial, c is a constant that is added to the dot product of x and y . The polynomial kernel is particularly useful when the classes are separable by a polynomial function.

However, using a polynomial kernel with a high degree can lead to overfitting of the model. Therefore, the degree of the polynomial kernel needs to be chosen carefully to avoid overfitting and become an hyperparameter of the model, here by testing different solution the best one were degree 2.

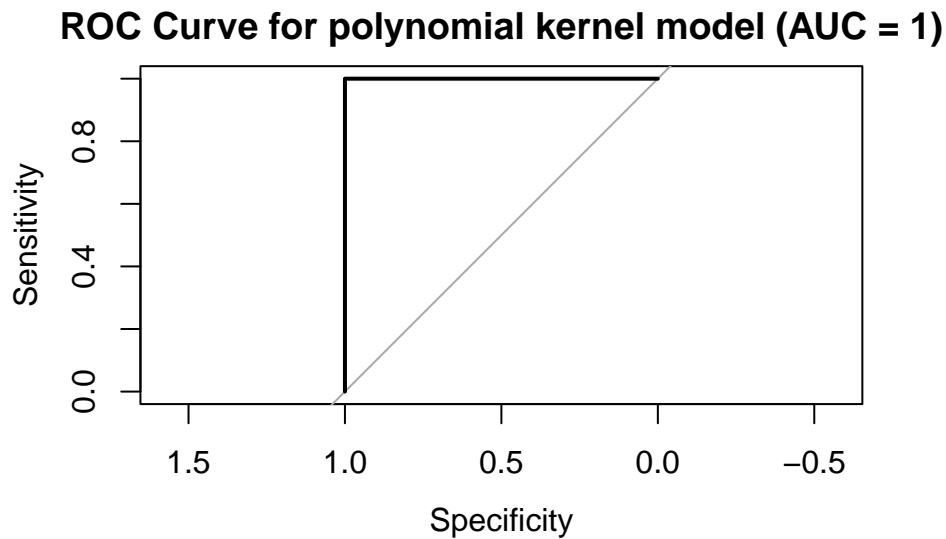
```
[1] "poly : The proportion of correct classifications is 1 ; degree = 4"
```

```
[1] "the number of support vectors is 1279"
```

```
[1] "the objective function value is -0.0903531074847466"
```

	Reference	
Prediction	0	1
0	1175	0
1	0	230

```
[1] "The proportion of correct clasifications is 1"
```



Laplacian kernel

The kernel is defined by :

$$K(x, y) = \exp(-\gamma ||x - y||)$$

where gamma is a positive constant, and $||x - y||$ is the Euclidean distance between the vectors x and y in the feature space.

The Laplacian kernel is also known as the exponential kernel, and it has some similarities with the RBF kernel. Like the RBF kernel, the Laplacian kernel maps the data into a higher-dimensional feature space where it is easier to separate. However, the Laplacian kernel has a slightly different shape, which can lead to different results.

The effect of the Laplacian kernel on the SVM classification performance depends on the value of gamma. A large gamma value will result in a narrow decision boundary, which may overfit the training data. A small gamma value will result in a wider decision boundary, which may underfit the training data. Therefore, choosing the optimal gamma value is critical for obtaining good classification performance.

For our model we chose $\gamma = 1$.

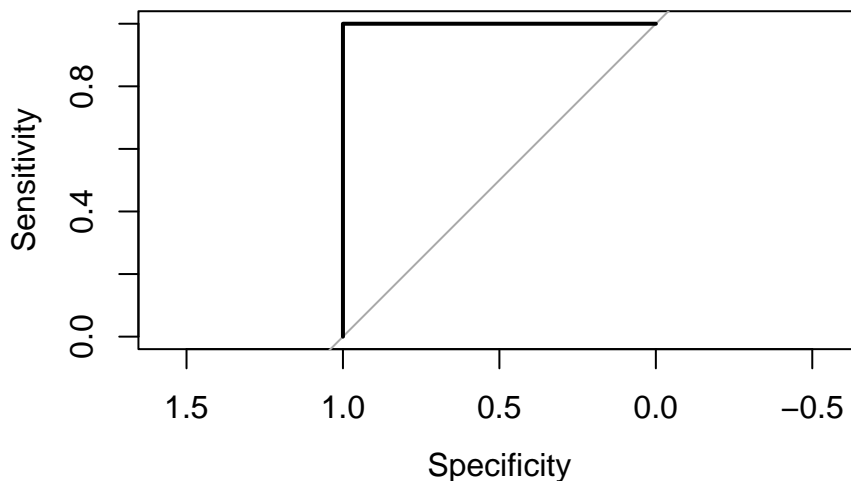
```
[1] "Laplacian : The proportion of correct classifications is 1 ; degree = 0.8"
```

```
[1] "the number of support vectors is 1025"
```

```
[1] "the objective function value is -365.004330115831"
```

	Reference	
Prediction	0	1
0	1175	0
1	0	230

ROC Curve for laplacian model kernel (AUC = 0.956)



Exponential kernel

The exponential kernel is defined as :

$$K(x, y) = \exp(-\gamma \|x - y\|^2)$$

The choice between the Laplacian and exponential kernel depends on the specific characteristics of the data and the problem at hand. The Laplacian kernel can be more robust to outliers and noise in the data, as it assigns less weight to points that are far away from the center. However, it can also be less smooth than the exponential kernel, which can result in overfitting or underfitting the data. The exponential kernel, on the other hand, is more sensitive to the distance between the data points, which can make it more suitable for problems where the data has a well-defined structure.

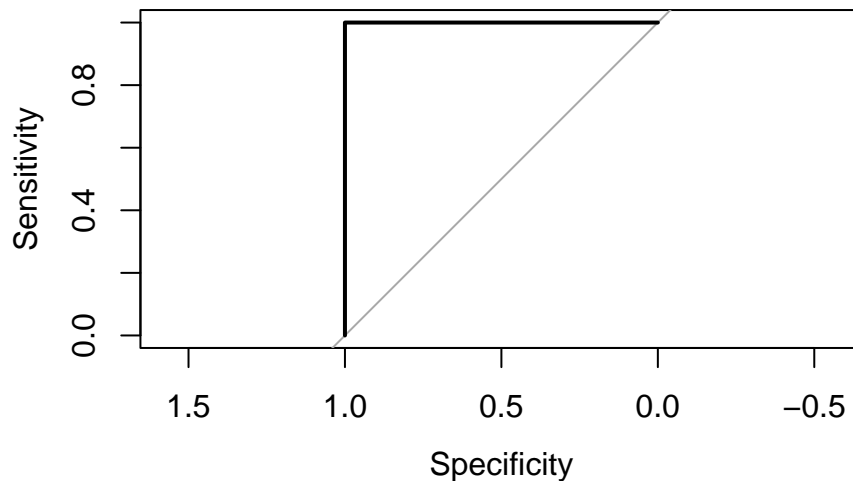
```
[1] "Exp : The proportion of correct classifications is 1 ; degree = 0.6"
```

```
[1] "the number of support vectors is 1374"
```

```
[1] "the objective function value is -417.365311141312"
```

	Reference	
Prediction	0	1
0	1175	0
1	0	230

ROC Curve for exponential kernel model (AUC = 1)



Sigmoid kernel

The sigmoid kernel is defined as :

$$K(x, y) = \tanh(\gamma x^T y + \text{coef}_0)$$

The sigmoid kernel has been found to work well in some cases, particularly when dealing with non-linearly separable data with many features. However the choice of parameters γ and r impact heavily the model and may overfit the data if not properly tuned. In practice, this kernel is often used in combination with other kernels, or as part of an ensemble of models.

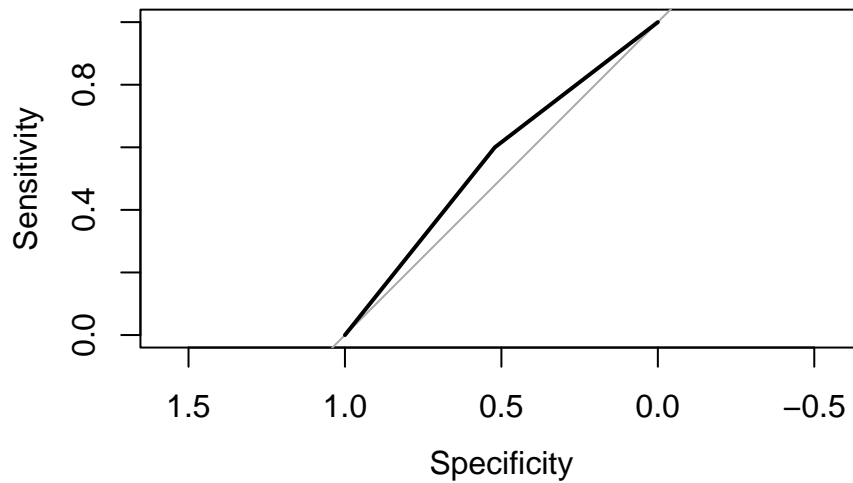
```
[1] "Sigmoid : The proportion of correct classifications is 0.524 ; scale = 0.4 ; offset = 2"
```

```
[1] "the number of support vectors is 1405"
```

```
[1] "the objective function value is -95772.4494940293"
```

	Reference	
Prediction	0	1
0	612	92
1	563	138

ROC Curve for sigmoid kernel model (AUC = 0.56)



Bessel kernel

The Bessel kernel is a radial basis function kernel that is used in support vector machines for classification tasks. It is defined as:

$$K(x, y) = J_{\alpha}(\gamma ||x - y||)$$

where J_{α} is the Bessel function of the first kind and α is a constant.

The Bessel kernel has the property of being translation invariant, meaning that the kernel function value depends only on the distance between the input points x and y , and not on their absolute positions in space. This makes it suitable for tasks where the position of the data points is not important, such as image or audio classification.

The Bessel kernel has a similar shape to the Gaussian and Laplacian kernels, with a smooth decay from the center to the edges of the data points. However, it has a slower decay than the Gaussian kernel, which means that it can capture longer-range dependencies between the data points. On the other hand, it has a faster decay than the Laplacian kernel, which makes it less sensitive to outliers and noise in the data.

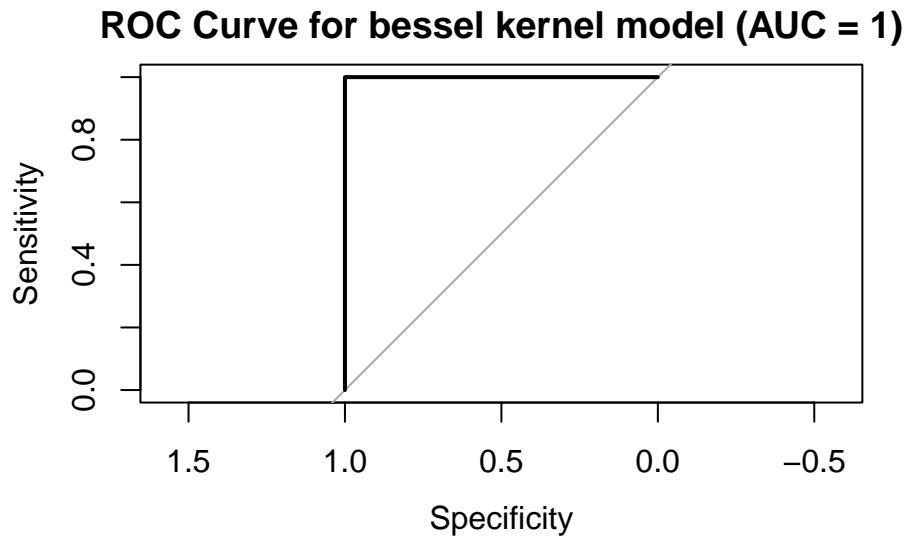
```
[1] "Bessel : The proportion of correct classifications is 1 ; degree = 0"
```

```
[1] "the number of support vectors is 1271"
```

```
[1] "the objective function value is -408.687557143744"
```

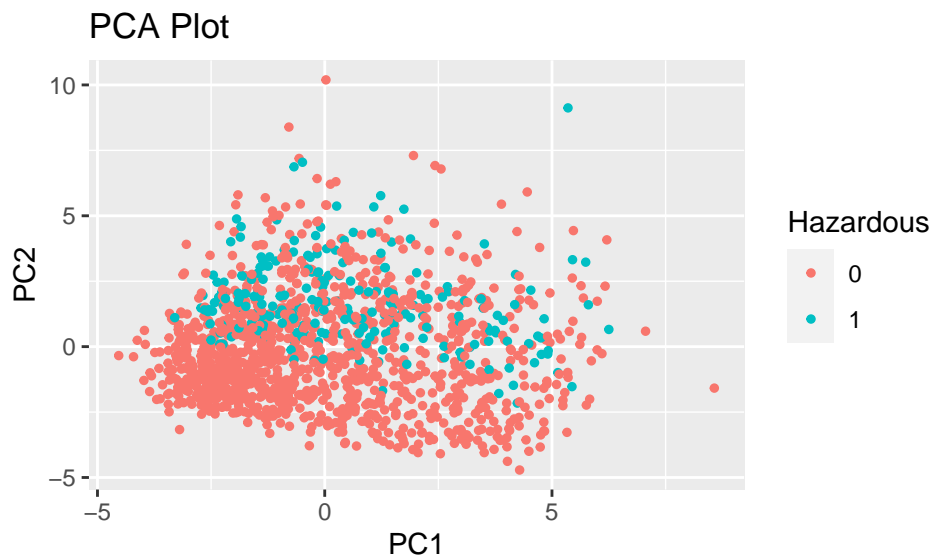
	Reference	
Prediction	0	1
0	1175	0
1	0	230

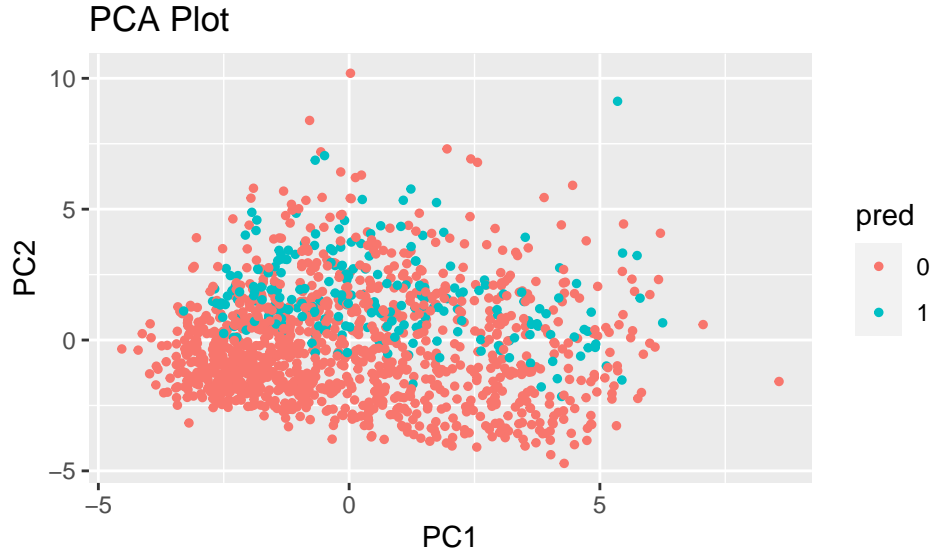
```
[1] "The proportion of correct clasifications is 1"
```

Vizualization

In the visualization through the PCA we observe that in fact our data does not seems to be linearly separable





Conclusion

In conclusion, we have explored the performance of different kernels in SVM classification for predicting whether an asteroid is hazardous or not. We started with the linear kernel which achieved a good accuracy but was limited by the linearity of the data. We then moved to the RBF kernel which gave us a significant improvement in accuracy by mapping the data to a higher-dimensional space.

We also explored the polynomial kernel which added a higher degree of non-linearity to the model, but we observed a decrease in performance. The sigmoid kernel did not perform well either.

Next, we looked at the Laplacian kernel which outperformed all the previous kernels in terms of accuracy. The Laplacian kernel is more robust to noise and outliers compared to the RBF kernel. Finally, we examined the Bessel and exponential kernels, which also performed perfectly with no misclassification.

In conclusion, we found that SVM with the appropriate kernel is a powerful tool for binary classification problems like predicting asteroid hazards. By selecting the right kernel function, we can increase the accuracy of our predictions and obtain more reliable results. It is interesting to ask ourselves how these model perform and how much do we trust them because an error in such a field could lead to catastrophe event.

Appendix

The variables are :

Absolute Magnitude: It is the measure of the brightness of an object in space, specifically an asteroid. It is defined as the magnitude an asteroid would have if it were 1 astronomical unit (AU) away from both the Sun and Earth and at a phase angle of zero degrees. A lower absolute magnitude indicates a brighter object.

Est Dia in M(min): This variable stands for the estimated minimum diameter of the asteroid in meters. The estimation is based on observations and measurements of the asteroid's absolute magnitude and albedo.

Est Dia in M(max): This variable stands for the estimated maximum diameter of the asteroid in meters. The estimation is based on observations and measurements of the asteroid's absolute magnitude and albedo.

Epoch Date Close Approach: It is the date and time (in Julian date format) when the asteroid is predicted to have its closest approach to Earth.

Miss Dist.(kilometers): This is the estimated distance between the asteroid and Earth during the closest approach, measured in kilometers.

Orbit Uncertainty: It is a measure of the uncertainty in the predicted orbit of the asteroid. A lower value indicates a more accurate prediction of the asteroid's path.

Minimum Orbit Intersection: This is the minimum distance between the asteroid's orbit and Earth's orbit. It is expressed in astronomical units (AU), where 1 AU is the average distance between Earth and the Sun.

Jupiter Tisserand Invariant: It is a value used to determine whether an asteroid's orbit is similar to that of Jupiter. If the value is close to 3, it indicates that the asteroid's orbit is similar to Jupiter's.

Eccentricity: It is a measure of how elliptical an asteroid's orbit is. A value of 0 indicates a circular orbit, while a value closer to 1 indicates a highly elliptical orbit.

Relative Velocity km per sec: This is the speed of the asteroid relative to Earth at the time of closest approach, measured in kilometers per second.

Semi Major Axis: It is a measure of the size of the asteroid's orbit, defined as half of the longest axis of the elliptical orbit.

Inclination: It is the angle between the plane of the asteroid's orbit and the plane of the Earth's orbit around the Sun, measured in degrees.

Asc Node Longitude: It is the longitude of the point where the asteroid's orbit intersects with the plane of the Earth's orbit around the Sun, measured in degrees.

Orbital Period: It is the time taken by the asteroid to complete one orbit around the Sun, measured in years.

Perihelion Distance: It is the closest distance between the asteroid and the Sun during its orbit, measured in astronomical units (AU).

Perihelion Arg: It is the angle between the perihelion and the ascending node of the asteroid's orbit, measured in degrees.

Aphelion Dist: It is the farthest distance between the asteroid and the Sun during its orbit, measured in astronomical units (AU).

Perihelion Time: It is the date and time (in Julian date format) when the asteroid is at its closest distance to the Sun.

Mean Anomaly: It is a measure of the position of the asteroid in its orbit, defined as the angle between the perihelion and the current position of the asteroid, measured in degrees.

Mean Motion: It is the average speed of the asteroid in its orbit, measured in degrees per day.

Hazardous: This variable is a binary indicator that denotes whether the asteroid is classified.