Prediction of hazardous asteroids

Illinois Institute of Technology

CSP571-Data Preparation and Analysis

Professor: Jawahar Panchal

Niveditha Mangala Venkatesh nmangalavenkatesha@hawk.iit.edu

Kajol Tanesh Shah kshah127@hawk.iit.edu

Saipooja Kandala skandala2@hawk.iit.edu

Sravani Bolisetti sbollisetti@hawk.iit.edu

Surya Thatee sthatee@hawk.iit.edu

Abstract:

Large bits of rock float around in space and are known as Asteroids. These cosmic monoliths are often found in a broad belt between Mars and Jupiter, although some are let free on occasion. When they get close enough to Earth, they are frequently caught in our gravity and shattered by the atmosphere, resulting in hot rock fragments. Most asteroids burn up in the atmosphere and fall to Earth as ash or tiny rocks, which is fortunate. Larger space rocks, on the other hand, are a serious danger. As of August 2021, there were 2,203 potentially harmful asteroids identified (about 8% of the entire near-Earth population), with 159 expected to be larger than one kilometre in diameter. Our goal is to create a model that can determine whether or not asteroids are dangerous and classifying asteroids according to orbit classes and determining which orbit classes contain hazardous asteroids.

Project Overview:

"We're smashing into an asteroid," NASA's Launch Services Program senior launch director Omar Baez said regarding SpaceX rocket Falcon 9, launched on Wednesday to deviate the path of the asteroid DART (Double Asteroid Redirection Test). The whole mission was the first planetary defence test mission from near-Earth objects. Early detection and investigation of possible collisions and close approaches of asteroids with the Earth are necessary to except the asteroid-comet hazard.

Project Objective:

The goal of this project is to apply regression modelling to forecast combinations of orbital characteristics for yet-undiscovered potentially hazardous asteroids (PHAs). The suggested method tries to identify subgroups within all main groupings of near-Earth asteroids (NEAs) that have a high proportion of PHAs, i.e., asteroids are hazardous or not.

Specific Questions:

- What effect do the dataset's other attributes have on determining whether an asteroid is hazardous or not?
- Determining whether an asteroid is dangerous or not?

Data Preparation:

Data Properties

The data that we used in this project are collected from Kaggle, asteroids dataset are found in https://www.kaggle.com/sakhawat18/asteroid-dataset, which is officially maintained by Jet Propulsion Laboratory which is an organization under NASA. We have collected the detailed datasets of all kinds of Data related to Asteroid. The Basic Definitions of the Columns and each data row contains the following definition.

Basic Column Definition

We have collected a detailed dataset of all kinds of data related to Asteroids. The basic description of the columns/features in the dataset is mentioned below,

- **SPK-ID**: Object primary SPK-ID
- **Object ID**: Object internal database ID
- **Object fullname**: Object full name/designation
- **pdes**: Object primary designation
- **name**: Object IAU name
- **NEO:** Near-Earth Object (NEO) flag
- **PHA:** Potentially Hazardous Asteroid (PHA) flag
- **H:** Absolute magnitude parameter
- **Diameter:** object diameter (from equivalent sphere) km Unit
- **Albedo:** Geometric albedo
- **Diameter_sigma:** 1-sigma uncertainty in object diameter km Unit
- **Orbit_id:** Orbit solution ID
- **Epoch:** Epoch of osculation in modified Julian day form
- **Equinox:** Equinox of reference frame
- **e:** Eccentricity
- **a:** Semi-major axis au Unit
- **q:** perihelion distance au Unit
- **i:** inclination; angle with respect to x-y ecliptic plane

- **tp:** Time of perihelion passage TDB Unit
- **moid_ld:** Earth Minimum Orbit Intersection Distance au Unit

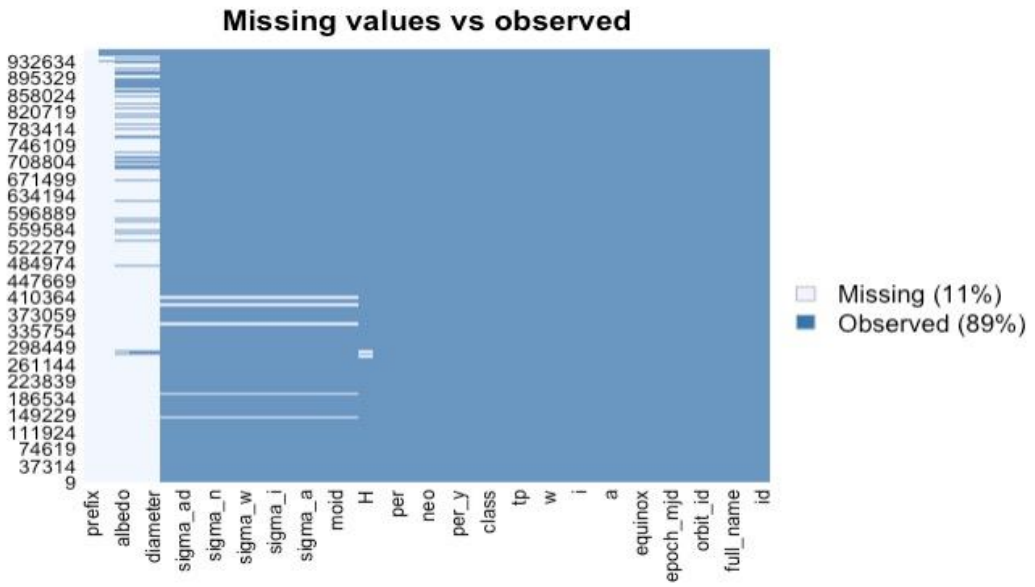
Our dataset is from Kaggle, and it contains both training and test data information. The training dataset has **750,880** rows and the test dataset has **187,719** rows, both data sets have 45 columns.

Data Source: Asteroid Dataset https://ssd.jpl.nasa.gov/tools/sbdb_query.html

Data Cleaning and Wrangling:

We focused on removing inaccurate data from our data set. And we converted the raw data format into readable data which helps in predicting the model.

Null and NA data:



Because we have null values in our data set, we have two methods for dealing with them.

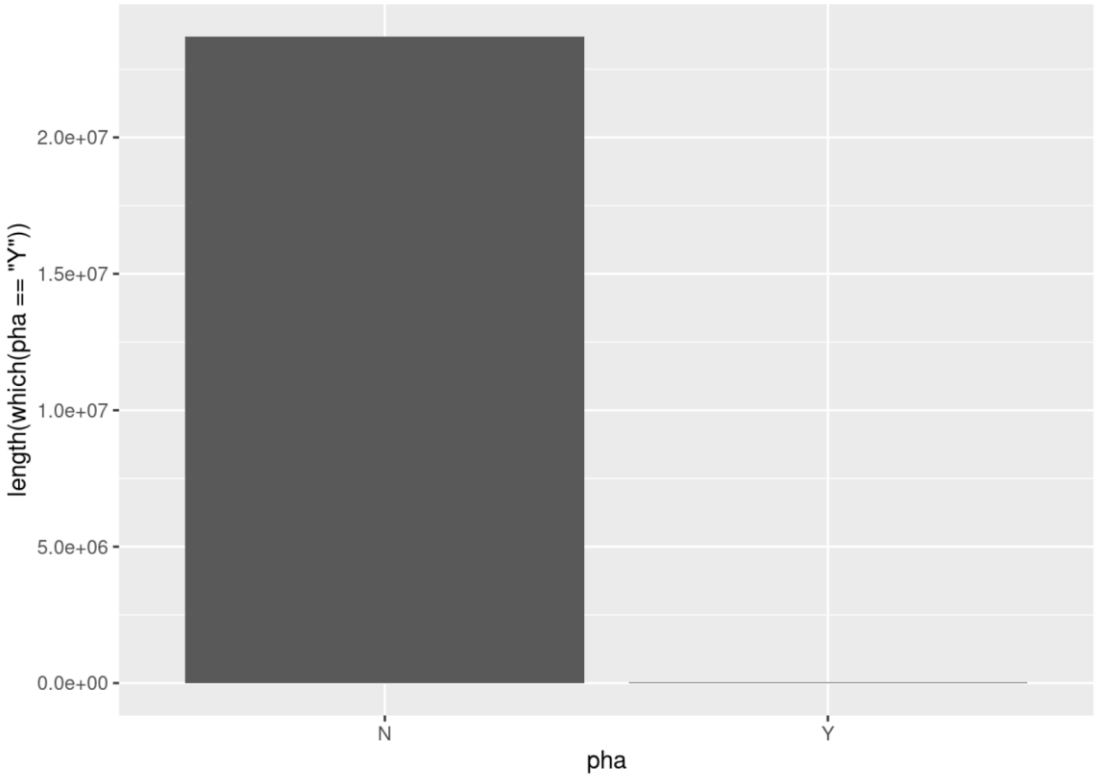
Approach 1: Replacing the missing values with the mean of that respective column in the data set.

Approach 2: Deleting the rows with null/missing values.

We don't know which approach produces superior outcomes, therefore we used both and compared the findings independently.

We eliminate the columns 'id,' 'names,' and 'prefix' from our dataset because they are not required for predicting the outcome.

```
##          spkid      full_name           pdes            neo            pha
##              0              0              0              0              0
##              H       diameter        albedo diameter_sigma       orbit_id
##              0              0              0              0              0
##          epoch     epoch_mjd      epoch_cal       equinox              e
##              0              0              0              0              0
##              a              q              i             om              w
##              0              0              0              0              0
##             ma             ad              n             tp         tp_cal
##              0              0              0              0              0
##            per          per_y          moid        moid_ld       sigma_e
##              0              0              0              0              0
##        sigma_a        sigma_q        sigma_i       sigma_om       sigma_w
##              0              0              0              0              0
##       sigma_ma       sigma_ad       sigma_n       sigma_tp     sigma_per
##              0              0              0              0              0
##          class            rms
##              0              0
```



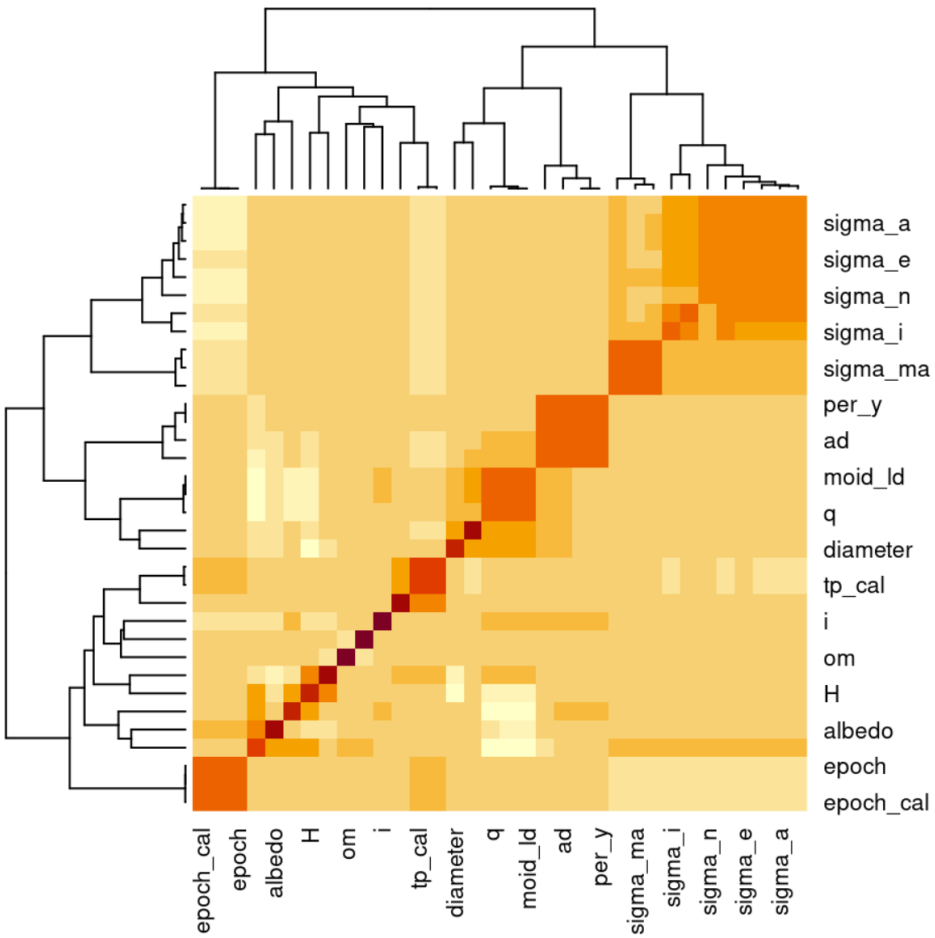
We also remove id, spkid, full_name, orbit_id, and equinox because they are no longer required.

```
##  [1] "pdes"          "neo"           "pha"           "H"
##  [5] "diameter"      "albedo"        "diameter_sigma" "epoch"
##  [9] "epoch_mjd"     "epoch_cal"     "e"             "a"
## [13] "q"             "i"             "om"            "w"
## [17] "ma"            "ad"            "n"             "tp"
## [21] "tp_cal"        "per"           "per_y"         "moid"
## [25] "moid_ld"       "sigma_e"       "sigma_a"       "sigma_q"
## [29] "sigma_i"       "sigma_om"      "sigma_w"       "sigma_ma"
## [33] "sigma_ad"      "sigma_n"       "sigma_tp"      "sigma_per"
## [37] "class"         "rms"
```

Data visualization:

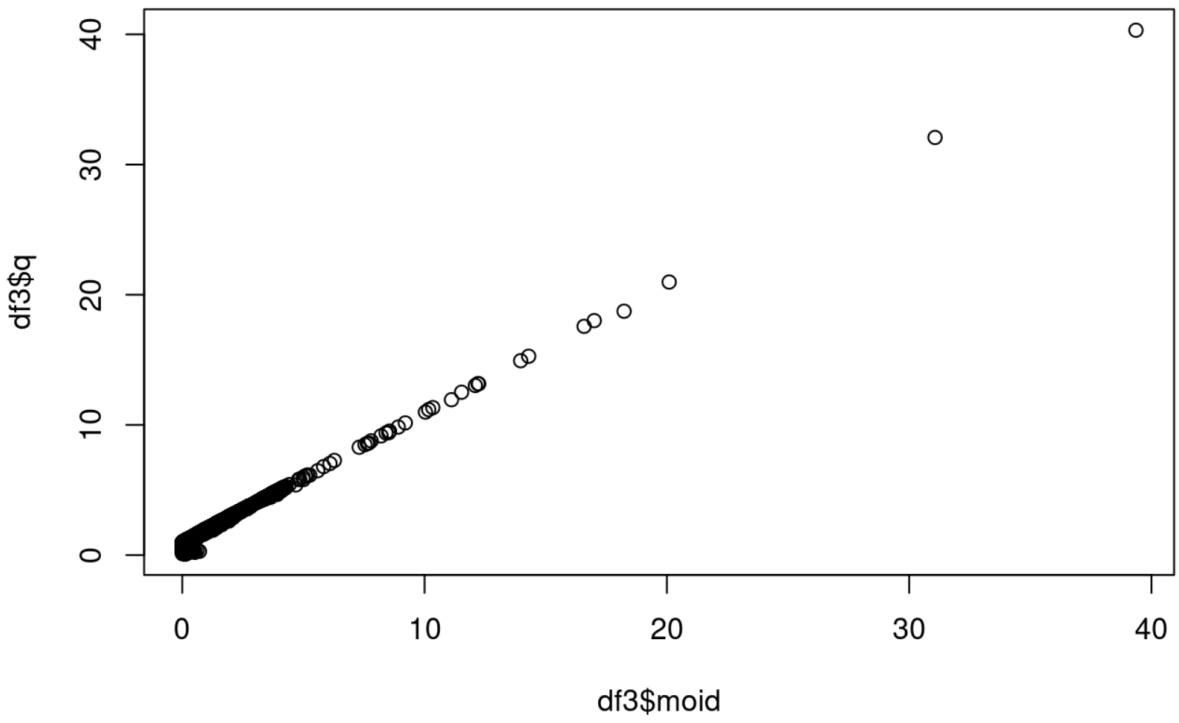
We cleaned and enhanced the dataset from its concept to show several various aspects.

Heat map:



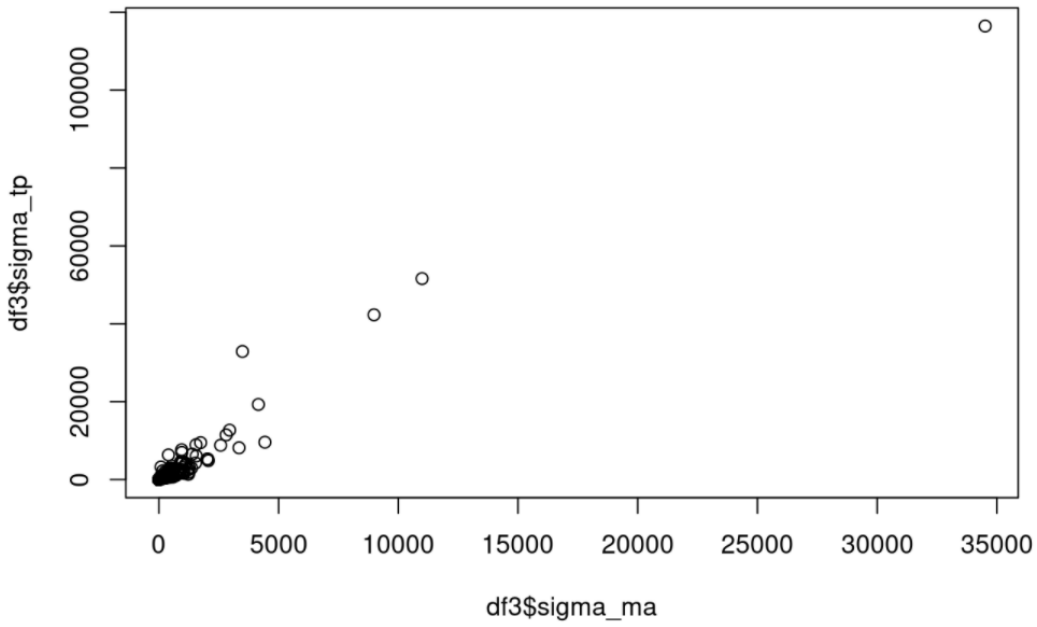
We can see from the heat-map above that some characteristics are correlated with one another, implying that they may be deleted.

epoch_mjd - epoch_cal, p_cal – tp, per - per_y, moid - moid_ld – q, features that are correlated to one another. As a result, per_y, tp_cal, and moid_ld columns were removed.

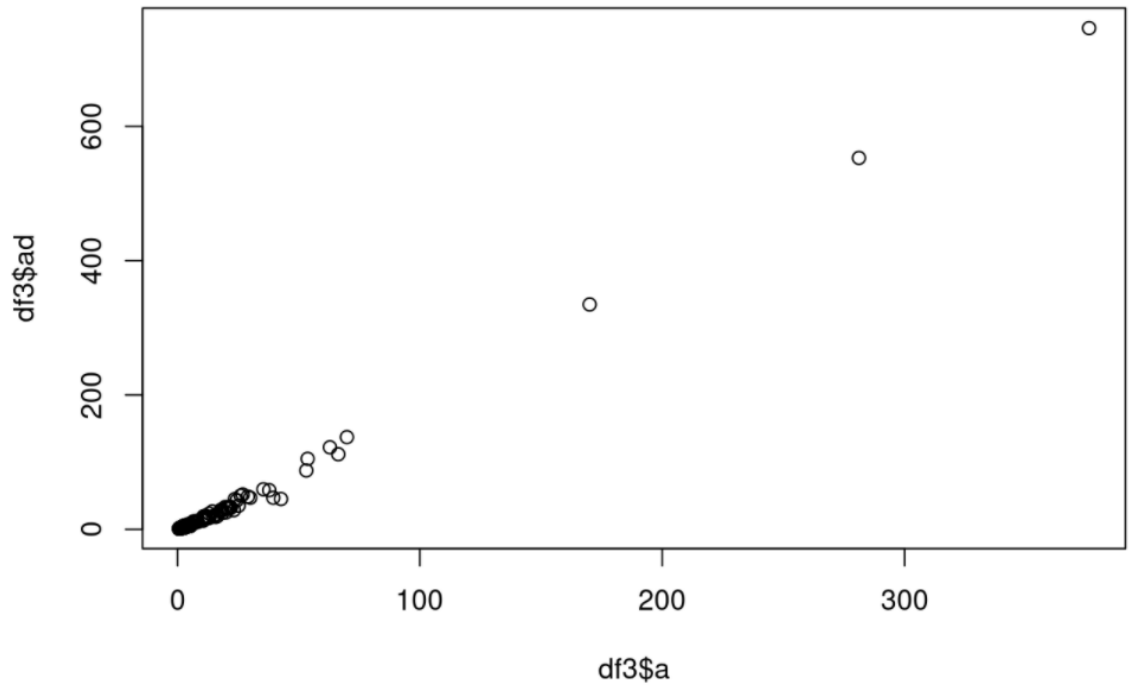


Scatter plot to find relation between sigma_ma and sigma_tp

The relationship between sigma_ma and sigma_tp is not linear, as we can see. As a result, none of them can be dropped.



Scatter plot to find relation between a and ad



The relationship between a and ad is not linear. As a result, none of them will be dropped.

Data summary:

After following the preceding data cleaning methods, we now have the data that is needed to develop our models.

The data's final summary is as follows:

```
##  [1] "H"              "diameter"       "albedo"         "diameter_sigma"
##  [5] "epoch"          "e"              "a"              "i"
##  [9] "om"             "w"              "ma"             "ad"
## [13] "n"              "tp"             "per"            "moid"
## [17] "sigma_e"        "sigma_a"        "sigma_q"        "sigma_i"
## [21] "sigma_om"       "sigma_w"        "sigma_ma"       "sigma_ad"
## [25] "sigma_n"        "sigma_tp"       "sigma_per"      "rms"
```

Modelling Analysis:

Approach 1:

We divide the datasets into train data and test data. There are 750880 training data and 187719 test data sets.

Logistic Regression Model

The logistic model (or logit model) is used in statistics to model the probability of a specific class or event, such as pass/fail, win/lose, alive/dead, or healthy/sick, existing. This can be used to represent a variety of occurrences, such as determining whether an image contains a cat, dog, lion, or other animals. Each detected object in the image would be assigned a probability ranging from 0 to 1, with a total of one.

When we do logistic regression on the datasets of the asteroids for approach 1 with 750880 training data and 187719 test data sets, the summary of the regression data model is,

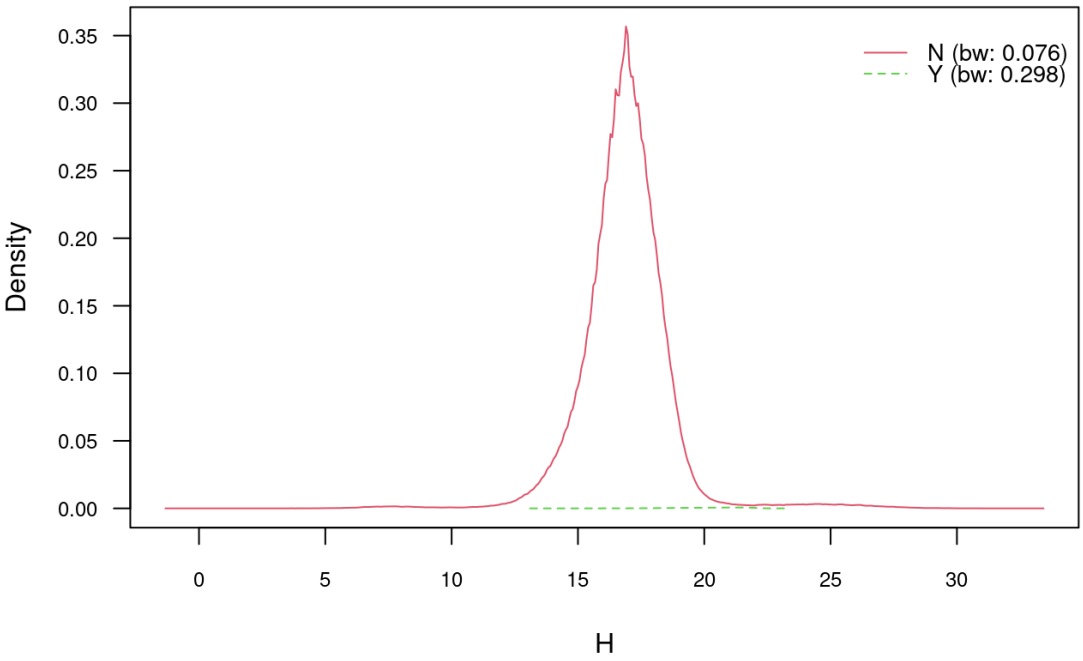
```
##
## Call:
## glm(formula = as.factor(asTraining$pha) ~ ., family = binomial(link = "logit"),
##     data = asTraining)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -8.49    0.00    0.00    0.00    8.49
##
## Coefficients:
##                   Estimate Std. Error    z value Pr(>|z|)
## (Intercept)    -1.162e+17  2.793e+08 -4.159e+08   <2e-16 ***
## H              -8.195e+13  6.320e+04 -1.297e+09   <2e-16 ***
## diameter       -7.603e+12  2.347e+04 -3.239e+08   <2e-16 ***
## albedo         -8.250e+14  1.911e+06 -4.317e+08   <2e-16 ***
## diameter_sigma  6.307e+12  2.934e+05  2.150e+07   <2e-16 ***
## epoch           1.292e+09  1.264e+02  1.022e+07   <2e-16 ***
## e               2.014e+15  9.835e+05  2.048e+09   <2e-16 ***
## a               2.997e+15  4.097e+06  7.315e+08   <2e-16 ***
## i              -6.167e+12  1.310e+04 -4.706e+08   <2e-16 ***
## om             -1.509e+11  7.605e+02 -1.985e+08   <2e-16 ***
## w              -5.877e+10  7.526e+02 -7.809e+07   <2e-16 ***
## ma             -7.014e+10  7.457e+02 -9.406e+07   <2e-16 ***
## ad             -1.501e+15  2.049e+06 -7.324e+08   <2e-16 ***
## n               9.990e+14  1.344e+06  7.432e+08   <2e-16 ***
## tp              4.553e+10  5.554e+01  8.199e+08   <2e-16 ***
## per             6.106e+07  1.752e-01  3.486e+08   <2e-16 ***
## moid           -1.586e+15  2.045e+06 -7.756e+08   <2e-16 ***
## sigma_e         1.717e+11  3.259e+03  5.267e+07   <2e-16 ***
## sigma_a        -1.227e+10  3.669e+02 -3.344e+07   <2e-16 ***
## sigma_q        -5.283e+09  8.791e+01 -6.009e+07   <2e-16 ***
## sigma_i         3.510e+10  1.168e+03  3.007e+07   <2e-16 ***
## sigma_om       -3.967e+09  7.463e+01 -5.316e+07   <2e-16 ***
```

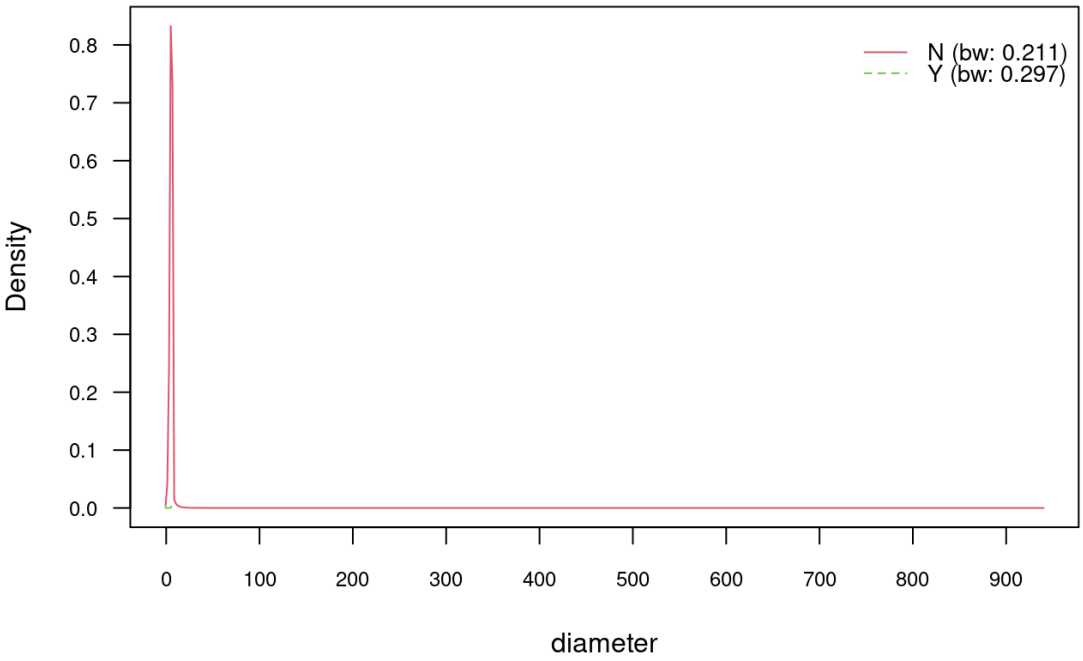
After getting the summary values, we calculated predicted values and error of the model which helps us in calculating the accuracy of the model. Therefore, the accuracy of the model is **“0.997379061256452”.**

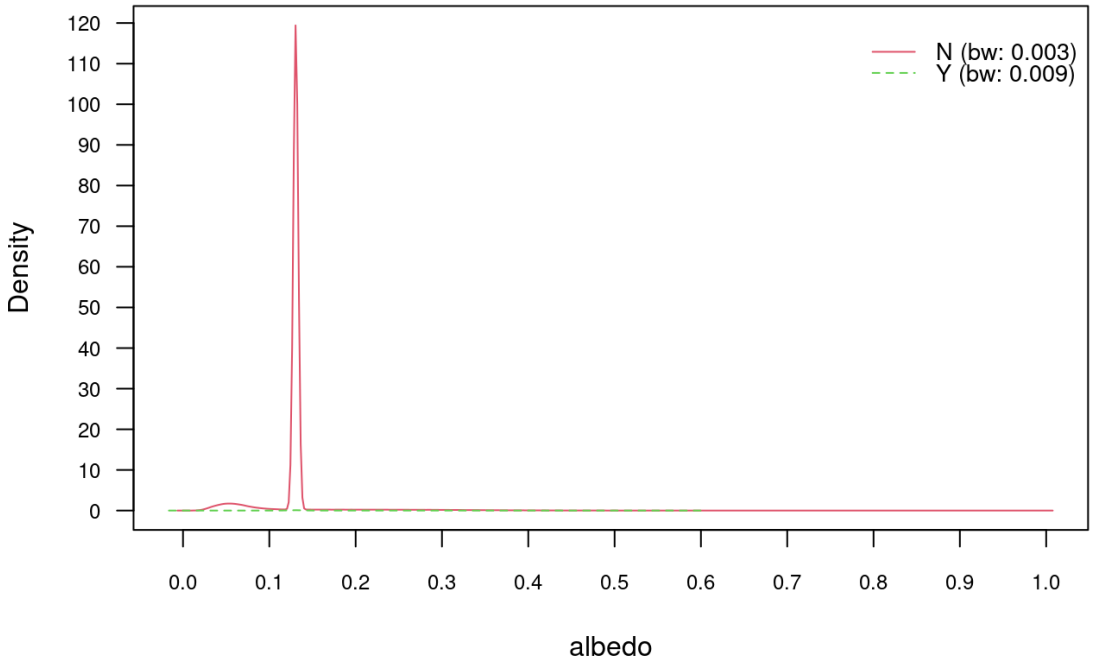
Naive Bayes Model:

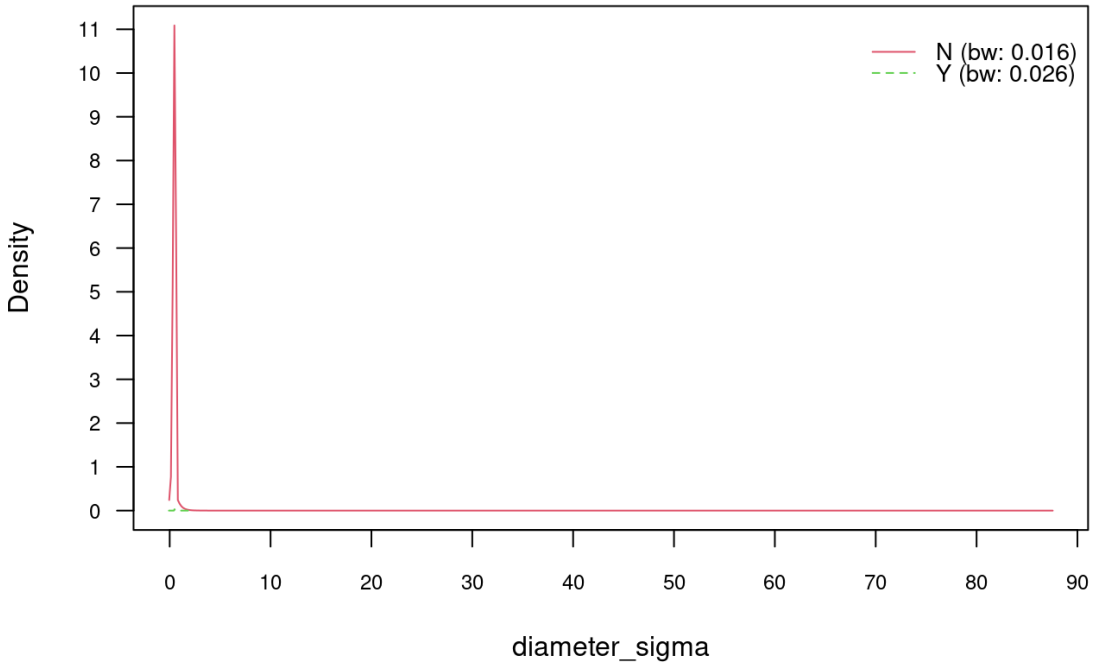
In statistics, naive Bayes classifiers are a subset of "probabilistic classifiers" based on Bayes' theorem and strong (naive) independence assumptions between features (see Bayes classifier). They are among the most basic Bayesian network models, but when combined with kernel density estimation, they may attain greater levels of accuracy.

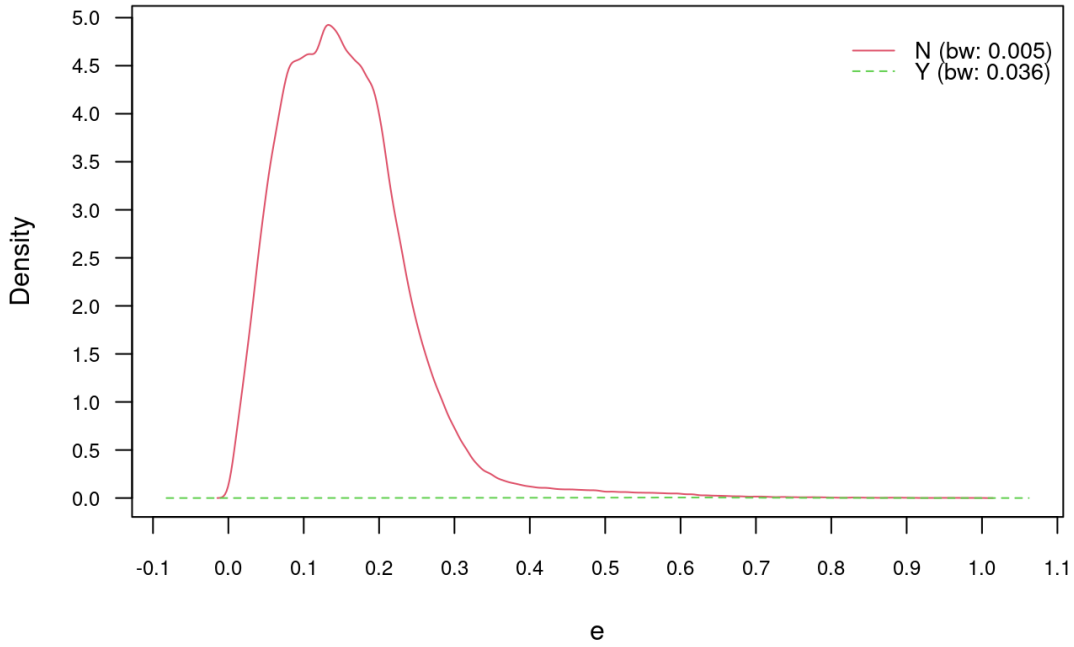
When naive Bayes is applied to the datasets of approach 1, we get the following plots:

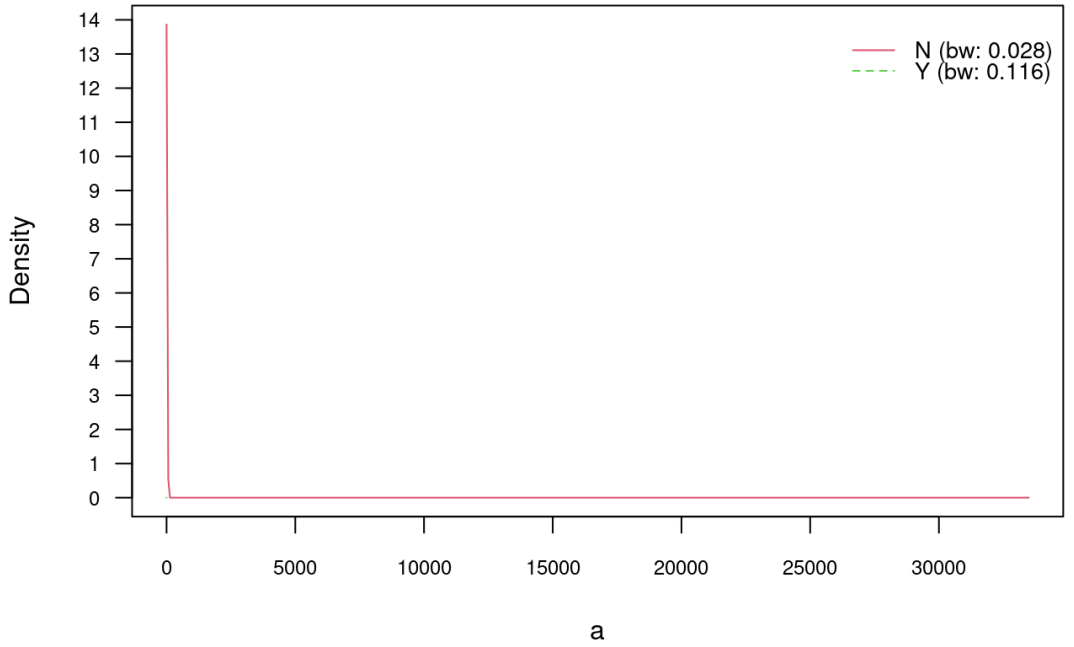

N (bw: 0.076)
Y (bw: 0.298)
Density
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
25
30
H


N (bw: 0.211)
Y (bw: 0.297)
Density
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
100
200
300
400
500
600
700
800
900
diameter


N (bw: 0.003)
Y (bw: 0.009)
Density
120
110
100
90
80
70
60
50
40
30
20
10
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
albedo


N (bw: 0.016)
Y (bw: 0.026)
Density
11
10
9
8
7
6
5
4
3
2
1
0
0
10
20
30
40
50
60
70
80
90
diameter_sigma


N (bw: 0.005)
Y (bw: 0.036)
Density
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
e

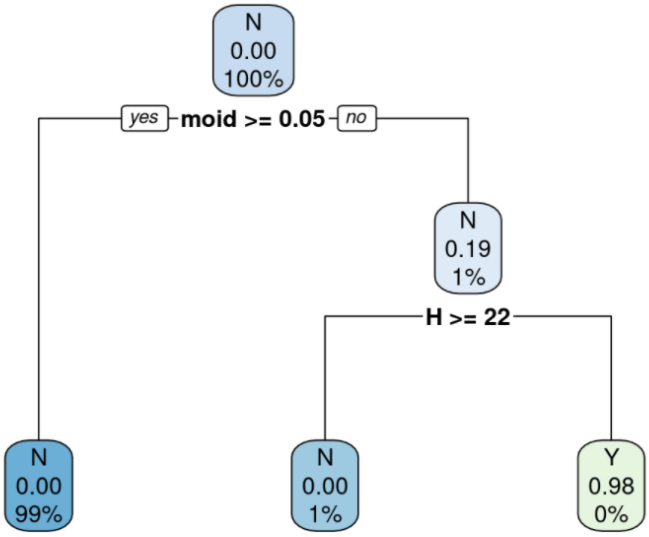

N (bw: 0.028)
Y (bw: 0.116)
Density
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
0
5000
10000
15000
20000
25000
30000
a

The accuracy of the Naïve Bayes model is **“0.997778594601505**”.

Decision tree:

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (for example, whether a coin flip will come up heads or tails), each branch reflects the test's conclusion, and each leaf node represents a class label (decision taken after computing all attributes).

When we apply the decision tree on the asteroid’s dataset, then the plot graph is,



The confusion matrix of the decision tree for test dataset is,

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      N      Y
##          N 187299      4
##          Y      7    409
##
##                Accuracy : 0.9999
##                  95% CI : (0.9999, 1)
##     No Information Rate : 0.9978
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9867
##
##  Mcnemar's Test P-Value : 0.5465
##
##             Sensitivity : 1.0000
##             Specificity : 0.9903
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9832
##              Prevalence : 0.9978
##          Detection Rate : 0.9978
##    Detection Prevalence : 0.9978
##       Balanced Accuracy : 0.9951
##
##        'Positive' Class : N
##
```

The confusion matrix of the decision tree for training dataset is,

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      N      Y
##          N 749197     15
##          Y     30   1638
##
##                Accuracy : 0.9999
##                  95% CI : (0.9999, 1)
##     No Information Rate : 0.9978
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.9864
##
##  Mcnemar's Test P-Value : 0.03689
##
##             Sensitivity : 1.0000
##             Specificity : 0.9909
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9820
##              Prevalence : 0.9978
##          Detection Rate : 0.9978
##    Detection Prevalence : 0.9978
##       Balanced Accuracy : 0.9954
##
##        'Positive' Class : N
##
```

Based on the confusion matrix, we can see the accuracy of the Decision tree model is **“99%”**.

Gradient Boosting Machine Model:

We're going to use GBM after we finish the decision tree model. To obtain the final forecast in this model, we integrated numerous decision trees. Every decision tree node uses a distinct subset of features to choose the optimum split. This means that the individual trees aren't all identical, and they can capture various signals from the data as a result.

Using the predicted values and errors of the model, we were able to determine the model's accuracy as **“0.9798”**.

Support Vector Machine Model:

Support-vector machines (also known as support-vector networks) are supervised learning models that examine data for classification and regression analysis in machine learning. SVMs, which are based on statistical learning frameworks or VC theory, is one of the most robust prediction approaches. An SVM training algorithm produces a model that assigns new examples to one of two categories, making it a non-probabilistic binary linear classifier, given a series of training examples, each marked as belonging to one of two categories (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). SVM assigns points in space to training examples in order to widen the distance between the two categories.

The confusion matrix of the SVM model for test data is given below,

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      N      Y
##          N 187300    412
##          Y      6      1
##
##                Accuracy : 0.9978
##                  95% CI : (0.9975, 0.998)
##     No Information Rate : 0.9978
##     P-Value [Acc > NIR] : 0.6097
##
##                   Kappa : 0.0047
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.999968
##             Specificity : 0.002421
##          Pos Pred Value : 0.997805
##          Neg Pred Value : 0.142857
##              Prevalence : 0.997800
##          Detection Rate : 0.997768
##    Detection Prevalence : 0.999963
##       Balanced Accuracy : 0.501195
##
##        'Positive' Class : N
##
```

The confusion matrix of the SVM model for training data is given below,

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      N      Y
##          N 749204   1653
##          Y     23      0
##
##                Accuracy : 0.9978
##                  95% CI : (0.9977, 0.9979)
##     No Information Rate : 0.9978
##     P-Value [Acc > NIR] : 0.7195
##
##                   Kappa : -1e-04
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.9978
##          Neg Pred Value : 0.0000
##              Prevalence : 0.9978
##          Detection Rate : 0.9978
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : N
##
```

Analysing the data with a Support Vector Machine Model gives an accuracy of “0.9978”.

Approach 2:

We'll divide the data into two parts before using the modeling approaches. Test and Train data.

In this approach we deleted the null values. After the deletion the number of rows reduced drastically.

The size of the train and test data sets are **104914, 26228** respectively.

Logistic Regression:

When we perform logistic regression on the datasets of the asteroids for approach 2, the summary of the regression data model is,

```
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -8.49    0.00    0.00    0.00    8.49
##
## Coefficients:
##                  Estimate Std. Error    z value Pr(>|z|)
## (Intercept)    -1.401e+18  4.229e+09 -331331307   <2e-16 ***
## H               5.845e+14  2.527e+05 2312641372   <2e-16 ***
## diameter        1.419e+13  3.134e+04  452972062   <2e-16 ***
## albedo          1.459e+15  2.375e+06  614496977   <2e-16 ***
## diameter_sigma -5.915e+13  3.646e+05 -162208517   <2e-16 ***
## epoch          -3.328e+10  1.782e+03  -18679824   <2e-16 ***
## e               8.884e+14  5.282e+06  168174757   <2e-16 ***
## a              -1.716e+15  1.442e+07 -118998266   <2e-16 ***
## i               2.151e+13  3.771e+04  570464283   <2e-16 ***
## om              6.098e+12  2.037e+03 2993779697   <2e-16 ***
## w              -1.740e+12  2.014e+03 -863904465   <2e-16 ***
## ma             -1.245e+12  2.270e+03 -548219042   <2e-16 ***
## ad              6.276e+14  7.341e+06   85488349   <2e-16 ***
## n              -9.145e+12  6.858e+06   -1333447   <2e-16 ***
## tp              5.979e+11  4.772e+02 1252918660   <2e-16 ***
## per             7.241e+10  1.936e+02  374078675   <2e-16 ***
## moid            1.709e+15  6.842e+06  249735766   <2e-16 ***
## sigma_e        -9.155e+14  7.213e+06 -126924748   <2e-16 ***
## sigma_a        -6.328e+14  1.307e+07  -48423946   <2e-16 ***
## sigma_q         3.979e+13  1.379e+06   28852560   <2e-16 ***
## sigma_i         6.903e+13  1.026e+06   67281230   <2e-16 ***
## sigma_om        7.439e+12  2.279e+05   32644363   <2e-16 ***
## sigma_w        -1.337e+12  3.091e+04  -43243818   <2e-16 ***
## sigma_ma        1.558e+12  3.395e+04   45888707   <2e-16 ***
## sigma_ad        2.418e+14  7.549e+06   32032217   <2e-16 ***
## sigma_n         1.412e+15  1.795e+07   78644310   <2e-16 ***
## sigma_tp       -4.973e+10  3.460e+03  -14372794   <2e-16 ***
## sigma_per       3.744e+11  9.494e+03   39435891   <2e-16 ***
## rms             5.779e+14  4.107e+06  140707130   <2e-16 ***
## ---
```

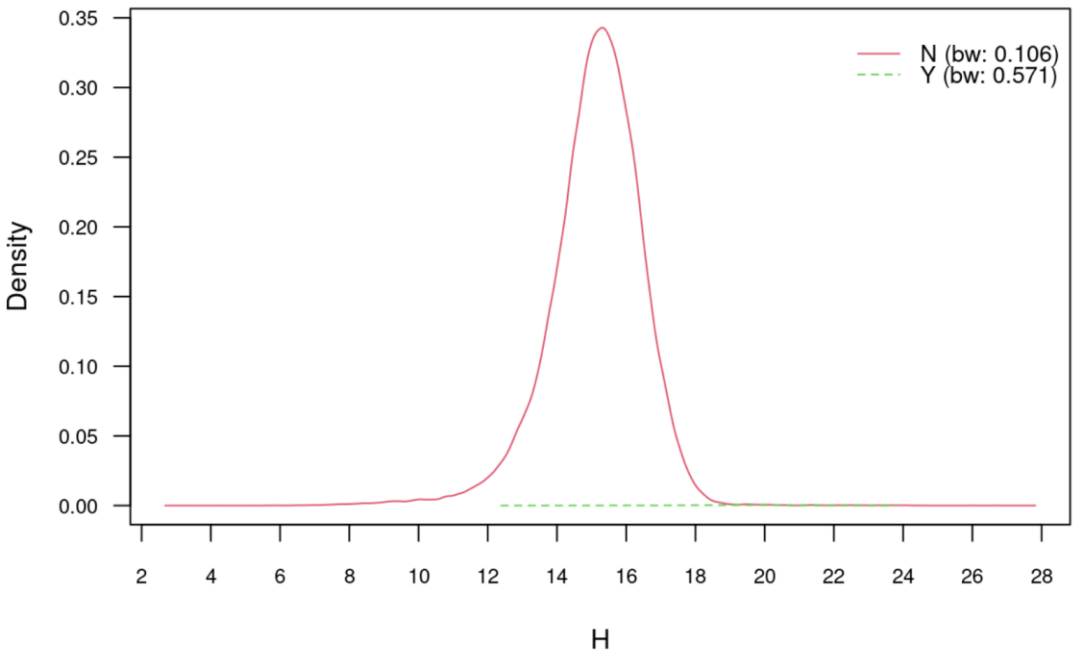
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance:  2199.2  on 104913  degrees of freedom
## Residual deviance: 16435.9  on 104885  degrees of freedom
## AIC: 16494
##
## Number of Fisher Scoring iterations: 24
```

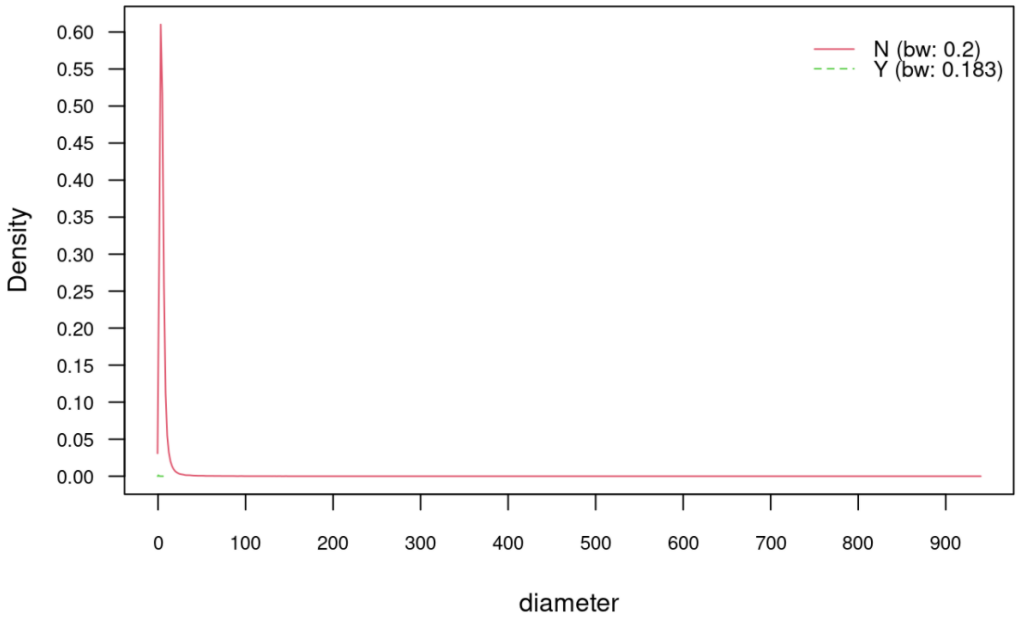
After obtaining the summary data, we estimated the model's predicted values and error, which aids in determining the model's accuracy.

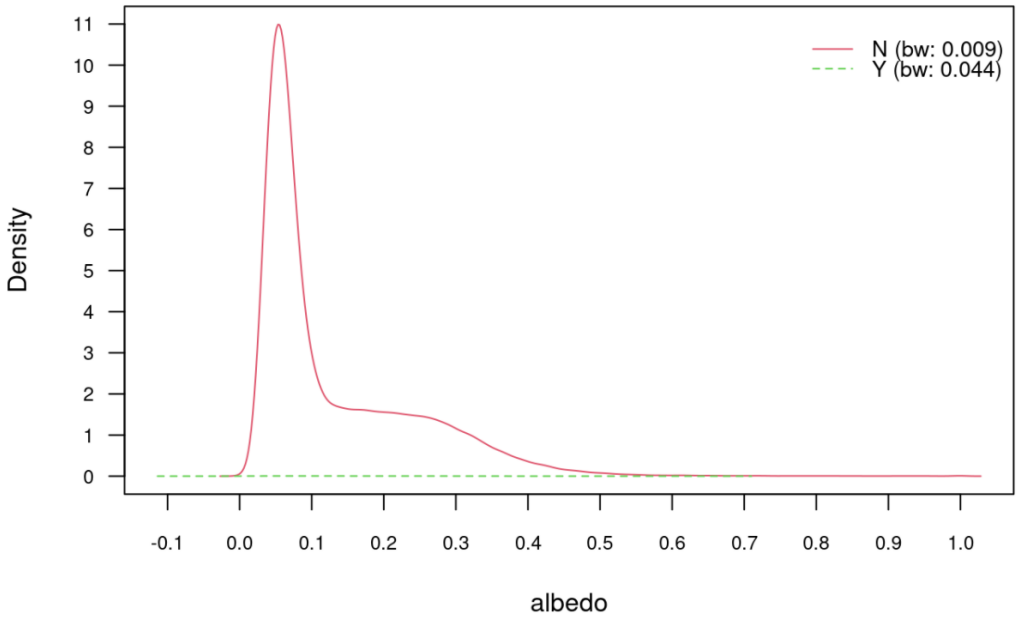
The model's accuracy is **"0.997597986884246".**

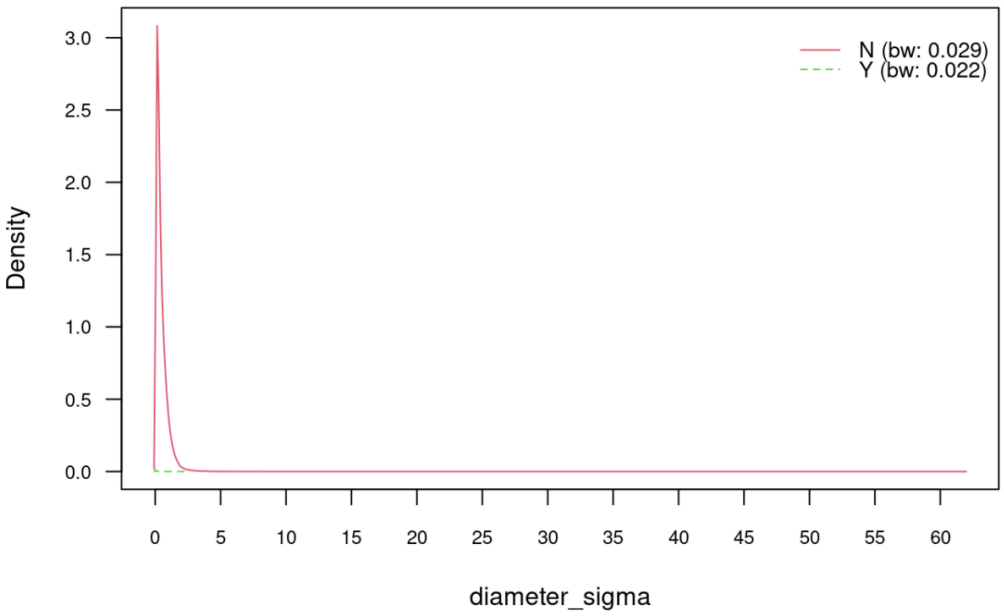
Naïve Bayes model:

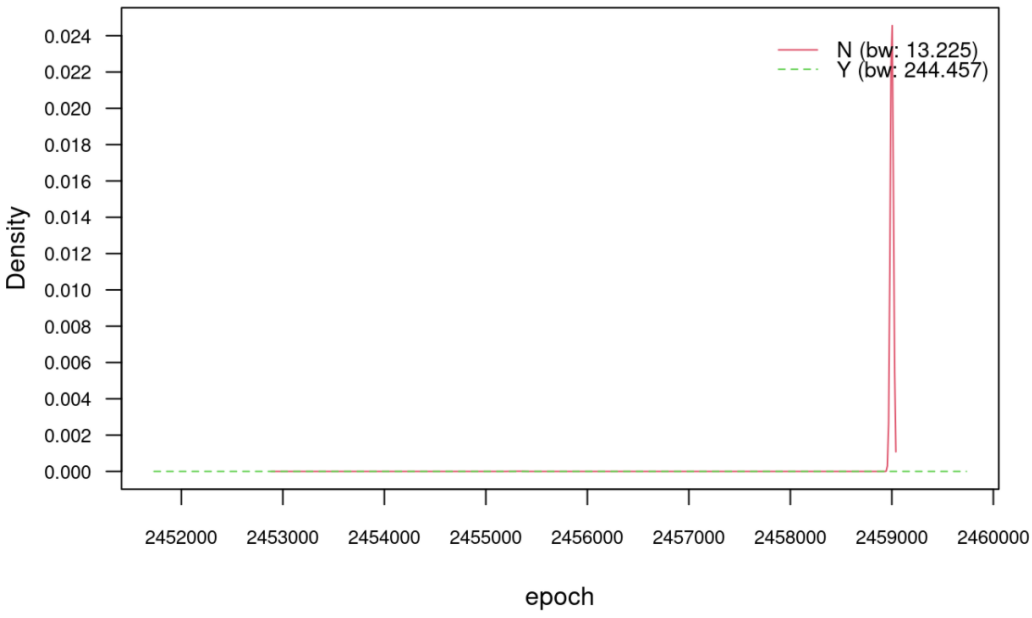
Following figures show the plots of NB Model,

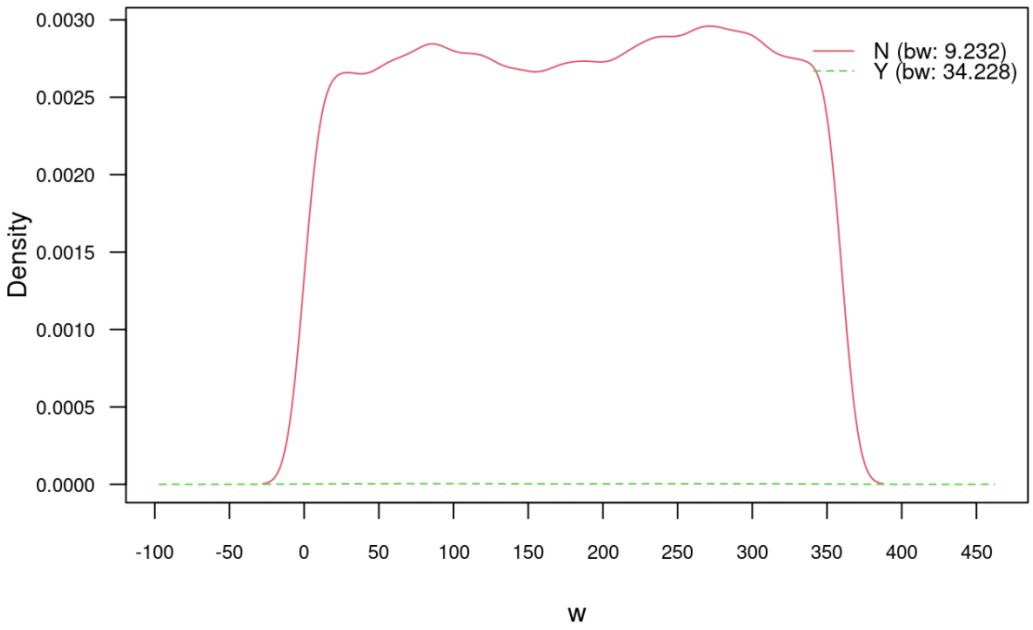


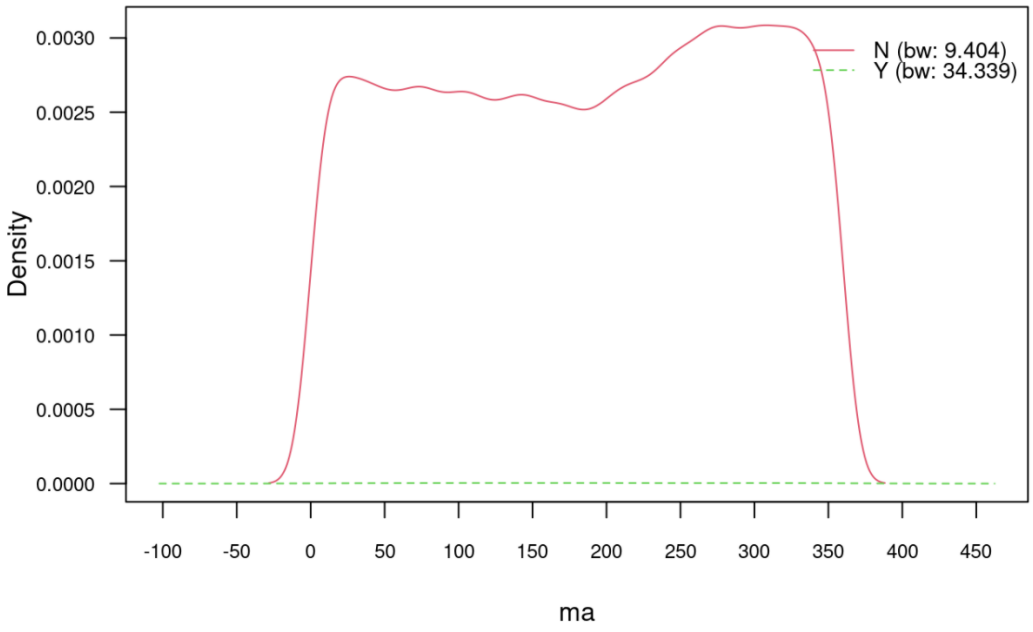

N (bw: 0.2)
Y (bw: 0.183)
Density
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
100
200
300
400
500
600
700
800
900
diameter

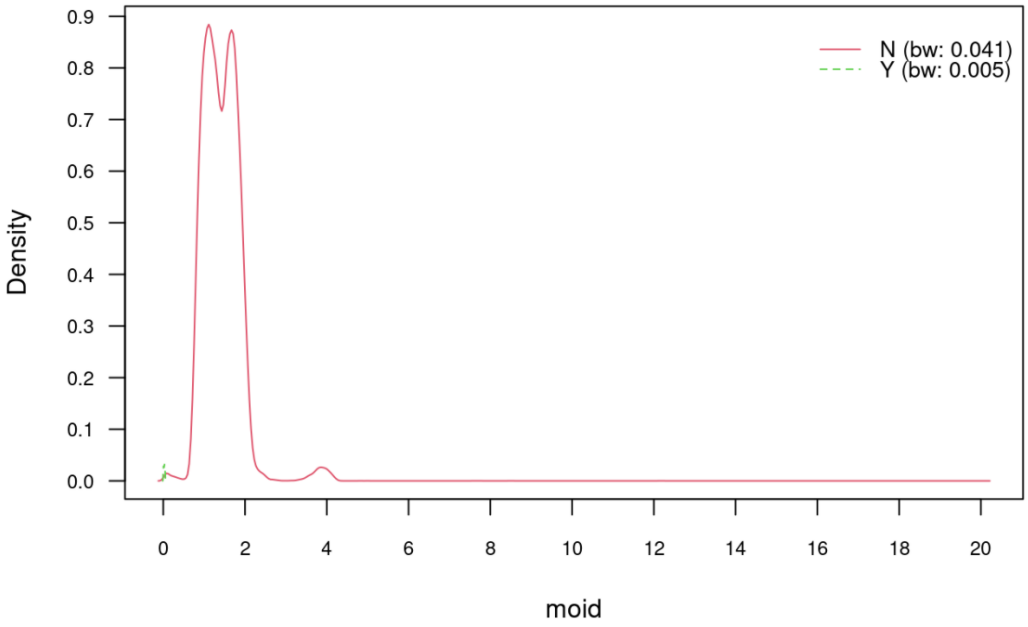

N (bw: 0.009)
Y (bw: 0.044)
Density
11
10
9
8
7
6
5
4
3
2
1
0
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
albedo

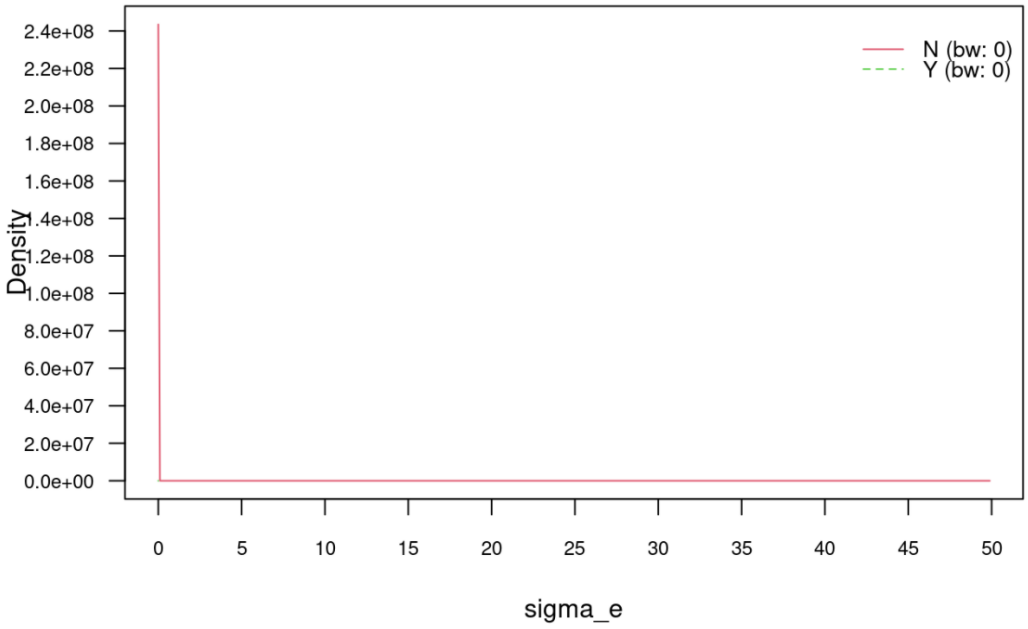

N (bw: 0.029)
Y (bw: 0.022)
Density
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
5
10
15
20
25
30
35
40
45
50
55
60
diameter_sigma


N (bw: 13.225)
Y (bw: 244.457)
Density
0.024
0.022
0.020
0.018
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
2452000
2453000
2454000
2455000
2456000
2457000
2458000
2459000
2460000
epoch


N (bw: 9.232)
Y (bw: 34.228)
Density
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
-100
-50
0
50
100
150
200
250
300
350
400
450
w


N (bw: 9.404)
Y (bw: 34.339)
Density
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
-100
-50
0
50
100
150
200
250
300
350
400
450
ma



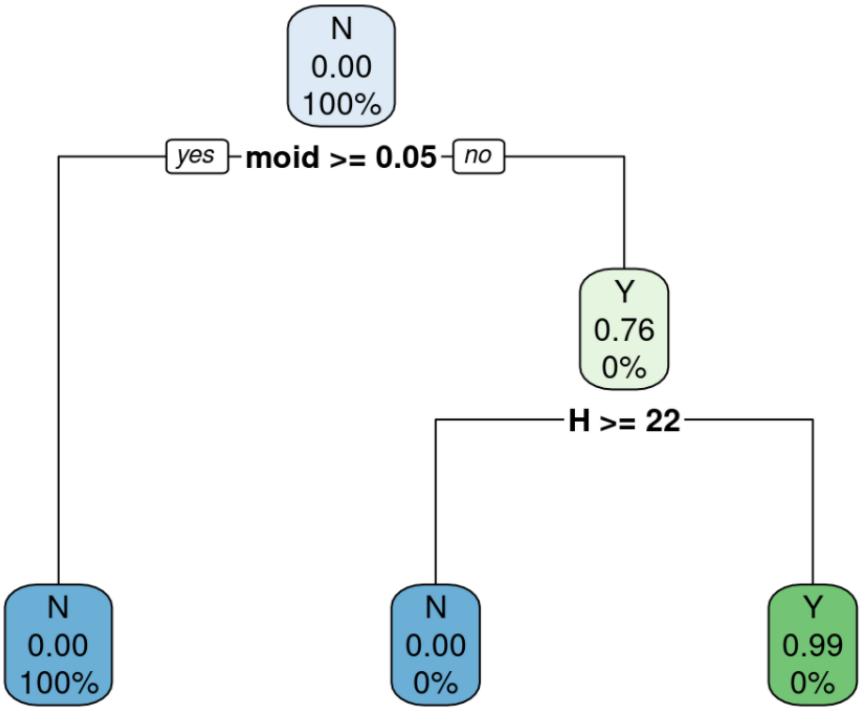


After calculating the NB models for the data the model's accuracy is

"0.997864877230441"

Decision Tree:

We fit the same data to a simple decision tree model which is visualized below.



We estimated the model's predicted values and error, which aids in determining the model's accuracy.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     N     Y
##          N 26192     1
##          Y     0    35
##
##                Accuracy : 1
##                  95% CI : (0.9998, 1)
##     No Information Rate : 0.9986
##     P-Value [Acc > NIR] : 8.384e-15
##
##                   Kappa : 0.9859
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9722
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.9986
##          Detection Rate : 0.9986
##    Detection Prevalence : 0.9987
##       Balanced Accuracy : 0.9861
##
##        'Positive' Class : N
##
```

As a result, the model's accuracy is **“99%”.**

Gradient Boosting Machine Model:

After performing the decision tree model we are implementing GBM. In this model we combined multiple decision trees to generate the final prediction. Following is the confusion matrix for GBM model.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     N     Y
##          N 26191     1
##          Y     3    33
##
##                Accuracy : 0.9998
##                  95% CI : (0.9996, 1)
##     No Information Rate : 0.9987
##     P-Value [Acc > NIR] : 1.059e-10
##
##                   Kappa : 0.9428
##
##  Mcnemar's Test P-Value : 0.6171
##
##             Sensitivity : 0.970588
##             Specificity : 0.999885
##          Pos Pred Value : 0.916667
##          Neg Pred Value : 0.999962
##              Prevalence : 0.001296
##          Detection Rate : 0.001258
##    Detection Prevalence : 0.001373
##       Balanced Accuracy : 0.985237
##
##        'Positive' Class : Y
##
```

Accuracy for this model is **“0.9998”.**

Support Vector Machine Model:

The confusion matrix for the train and test data is given below

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     N     Y
##          N 26192     1
##          Y     0    35
##
##                Accuracy : 1
##                  95% CI : (0.9998, 1)
##     No Information Rate : 0.9986
##     P-Value [Acc > NIR] : 8.384e-15
##
##                   Kappa : 0.9859
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9722
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.9986
##          Detection Rate : 0.9986
##    Detection Prevalence : 0.9987
##       Balanced Accuracy : 0.9861
##
##        'Positive' Class : N
##
```

The accuracy for SVM model is **“0.9978”.**

Conclusion:

Regarding the null values, we used both approaches to see which produced a better model. We found that the models for both approaches have the same accuracy, but the values of true positives in the confusion matrix were low in approach 1 for all the models except the decision tree. This is because 1st approach is getting trained on a large amount of data out of which most of them have negative labels. So, the models are getting trained to predict negative outcomes and have very little data to be trained to predict positive outcomes. Also, this approach is computationally much expensive that the other approach.

We observed that the second strategy, in which we deleted the rows with missing data, produced higher accuracy and precision than the first.

As we can see from the results above, the accuracy of all the models is very close, but we were able to find the best model by using the confusion matrix. When compared to other models, the decision tree model predicted the greatest number of true positives.

Though it is still less, with more data, we can train our model more precisely and predict asteroids that may collide with the Earth.

Data Sources

Kaggle provided us with the data we used in this project, the asteroids dataset can be found at https://www.kaggle.com/sakhawat18/asteroid-dataset

Source Code

We used GitHub for collaboration of our project with our team members,

GitHub Link: https://github.com/kajolshah310/Data_Preparation_Analysis

References

- https://www.researchgate.net/publication/338489667_Identifying_Earth-impacting_asteroids_using_an_artificial_neural_network
- https://www.nasa.gov/sites/default/files/atoms/files/nasem_report_finding_hazardous_asteroids.pdf
- https://www.spacesafetymagazine.com/space-hazards/asteroid-hitting-earth/identifying-potentially-dangerous-asteroids/