Potentially Hazardous Asteroids - Can we predict them?

Daniel Verdon

2023-02-17

Introduction/overview/executive summary

For this final project we will try to analyse the **NASA JPL Asteroid Dataset** pulled from Kaggle.com. It comprises of **958524** rows and **45** columns. We are given the following column descriptions

- **id** Object internal database ID
- **spkid** Object primary SPK-ID
- **full_name** 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

The goal of this project is to predict one of the 2 *flag* variables that we are given **"neo"** and/or **"pha"**. To achieve this we will be performing several operations. In addition to some basic data wrangling, we will perform various data visualisations and understand the saying *a picture can say a thousand words.* After already gaining insights from the visuals, we will perform several machine learning techniques on our models, notably **Generalized linear model; K-Nearest Neighbors Algorithm; Decision Trees**.

We will see that the high accuracy of the decision tree encourages us to dig deeper into the dataset to understand if any of the other variables that were not in our decison tree may also have some explanatory value.

Methods/Analysis

Data cleaning and Exploration

We will first create our dataframe by removing any empty values in the **pha** and **neo** columns. We then would like to decide which of them we would like to try to predict by looking at the table.

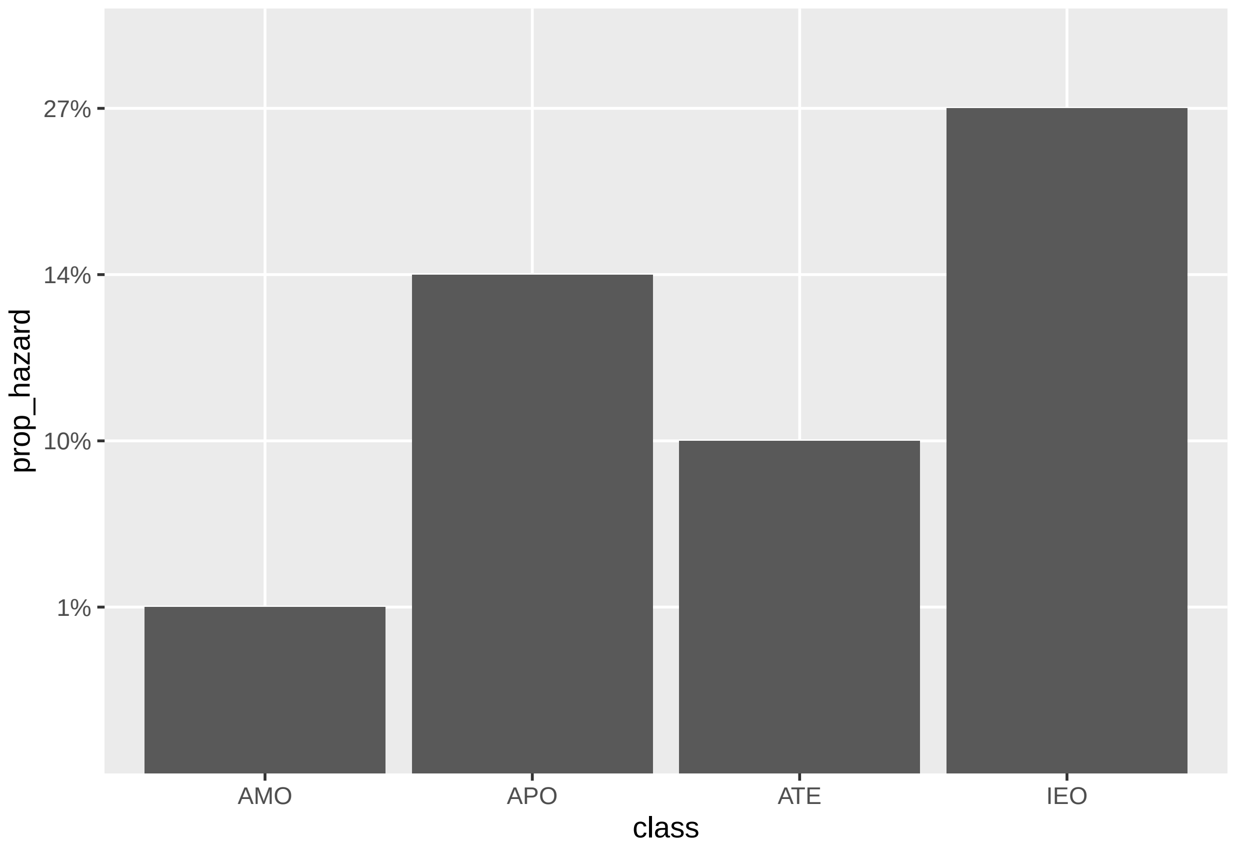
```
df <- readfile |> filter(pha != "" & neo != "") #remove blanks
df |> select(pha,neo) |> table() |> data.frame() |> kable() #explore pha, neo relationship
```

pha	neo	Freq
N	N	915705
Y	N	0
N	Y	20828
Y	Y	2066

We see that all **pha=Y** flags are always also **neo=Y** whereas there are some **pha=N** which are **neo=Y**. This means that **pha** is a better column to try to predict as these asteroids are more hazardous to earth, which is also evident in the column description *Potentially Hazardous Asteroid* compared to *neo: Near-Earth Object*.

We notice there is also a **class** variable which seems to classify asteroids. Lets see which class hazardous asteroids fall under

```
  df |> filter(pha == "Y" | neo == "Y") |>
            group_by(class,pha) |>
            summarise(n=n()) |>
            pivot_wider(class,names_from=pha,values_from = n) |>
            mutate(prop_hazard=percent(Y/(N+Y))) |>
            ggplot(aes(x=class,y=prop_hazard))+
            geom_col()
```



We see all hazardous asteroids fall under one of the following classes above. These class definitions also seem to align with the appearance of the hazardous flag.

- Apollos (APO) cross Earth's orbit and have a semi-major axis of more than 1 AU
- Amors (AMO) have orbits strictly outside Earth's orbit
- Atens (ATE) cross Earth's orbit and have a semi-major axis of less than 1 AU
- Atiras (IEO) have orbits strictly inside Earth's orbit

It makes sense to focus on only these classes in order to understand why sometimes it has a **pha** flag Y and sometimes N.

The following code verifies that there are no duplicate asteroids. We get zero which means there are none

```
df_distinct <- df |> distinct(spkid) |> nrow()
df_distinct - nrow(df)
```

```
## [1] 0
```

Lets analyse the *NAs*. In doing so we will only display columns where there is more than one NA

```
df_na <- df |>
  select(everything()) |>
  summarise_all(list(~sum(is.na(.))))

empty_columns <- sapply(df_na, function(x) all(x <= 1 ))
kable(df_na[, !empty_columns])
```

H	diameter	albedo	diameter_sigma
6262	802390	803496	802518

We see there are a lot of missing values for **diameter**, **diameter_sigma** and **albedo**. Given that the variable definition tells us it relates to the geometry of the asteroid we decide it likely wont have a lot of explanatory value on whether the asteroid is hazardous to earth. If we keep these variables then it will also mean removing **85%** of our dataframe.

Looking at the statistics of *diameter* we notice there may also be outliers as the median is *3.972*, the 3rd quartile is *5.7650* but the maximum is much higher at *939.4*.

```
df |> drop_na(diameter) |> select(diameter) |>  summary() |> kable()
```

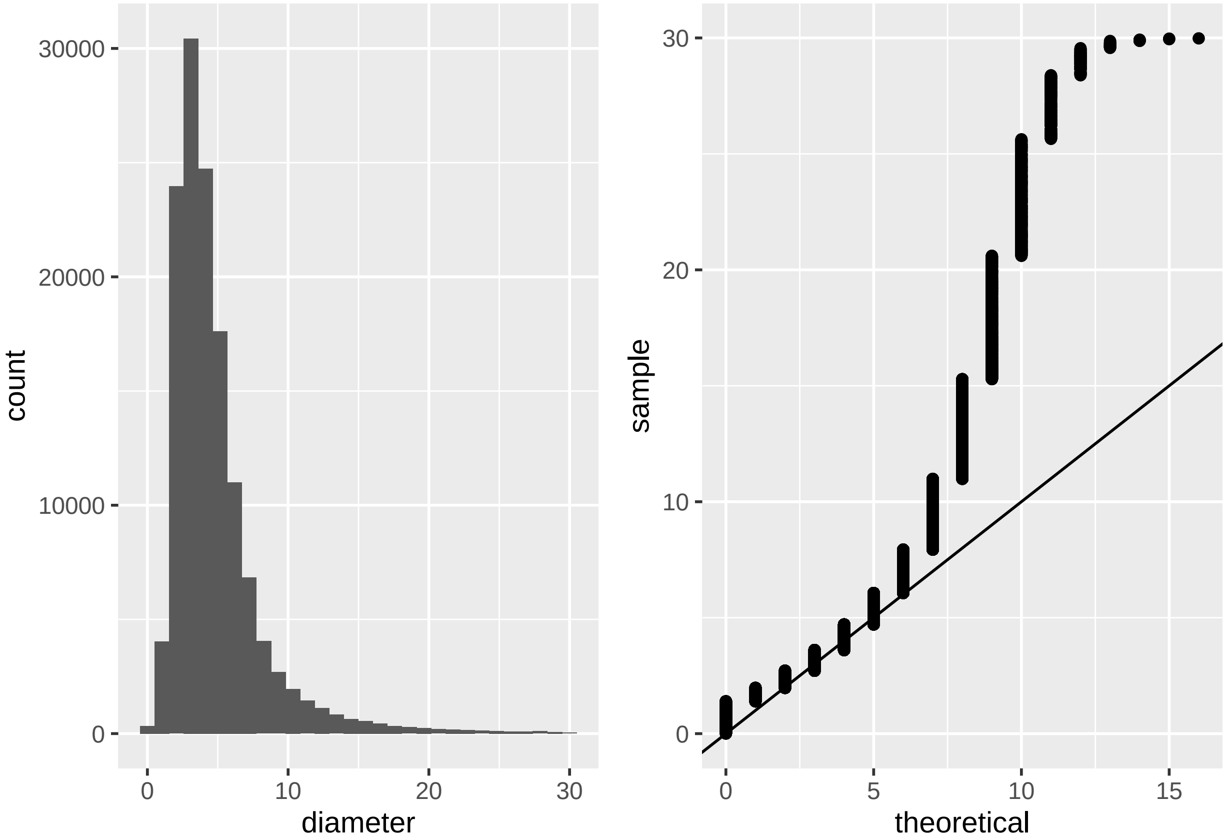
diameter
Min. : 0.0
1st Qu.: 2.8
Median : 4.0
Mean : 5.5
3rd Qu.: 5.8
Max. :939.4

Given third quartile value, we will look at the distribution of diameter for values < 30. The reason for setting a limit of 30 is we believe we have a lot of outliers and the number of asteroids bigger than this threshold is only…

```
## [1] "1%"
```

Analysing the distribution…

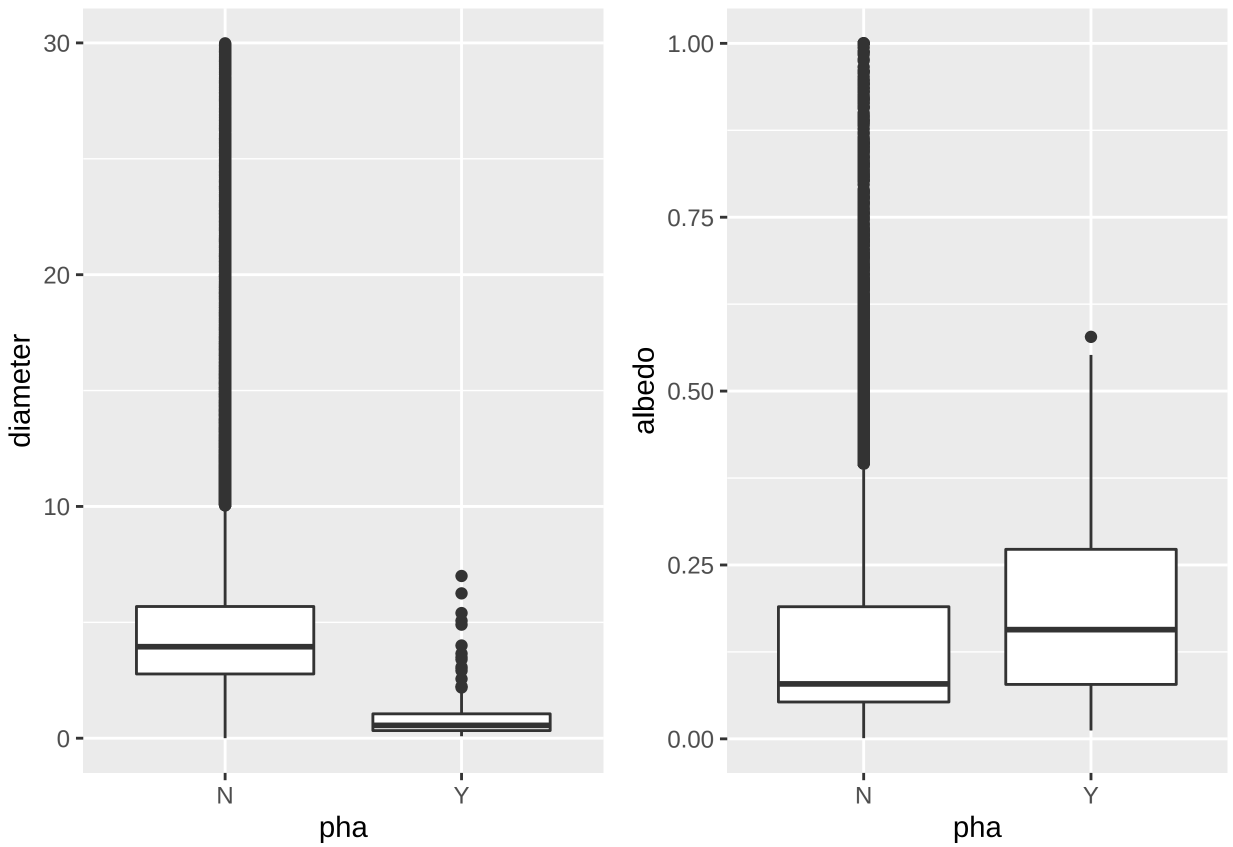
```
p1 <- df |> drop_na(diameter) |> filter(diameter<30) |>  ggplot(aes(x=diameter))+
  geom_histogram()


p2 <- df |> drop_na(diameter) |> filter(diameter<30) |> ggplot(aes(sample = diameter)) +
  stat_qq(distribution = qpois, dparams = list(lambda=4)) +
  geom_abline(intercept = 0, slope = 1)

grid.arrange(p1, p2, ncol = 2)
```



The graphs first tell us that there are a number of outliers as the plot looks like a *poisson* distribution when looking at values *0 to 30*. The poisson QQplot however tells us although it looks like poisson, the sample values are not following the theoretical values.

Despite not being able to identify the type of distribution we do see through a boxplot that there does seem to be some explanatory value in **diameter**. Interestingly smaller diameter asteroids seem to be more hazardous (this is disproven later). Contrast with **albedo** which does not appear to have much explanatory value.

```r
p1 <- df |> drop_na(diameter) |>
  filter(diameter<30) |>
  ggplot(aes(y=diameter, x=pha))+
  geom_boxplot()

p2 <- df |> drop_na(albedo) |>
  ggplot(aes(y=albedo, x=pha))+
  geom_boxplot()

grid.arrange(p1, p2, ncol = 2)
```



Building the dataframe and test/train splits for Machine Learning

As mentioned for our *primary* dataframe we perform the following actions.

- keep only classes AMO, APO, ATE and IEO
- remove diameter and albedo in order to prioritise data size and not significant explanatory value
- remove neo as we want to predict pha
- remove columns with variations of the asteroid name
- remove Epoch as it relates to the position of the satellite
- remove equinox as it relates to reference frame
- delete all sigma variables as it is the measurement error

Later we will test on a *second* dataframe which includes **diameter** as we saw it may hold some important value for predicting **pha**.

```
df_ast_big <- df |> filter(class %in% c("AMO","APO","ATE","IEO")) |> #Filter classes we want to evaluat
  select(-diameter,-albedo, -neo) |>  #remove diameter and albedo in first models to keep 100% of rows
  select(-class,-id,-full_name,-pdes,-name, -prefix,-orbit_id) |>  #remove names of the asteroid its or
  select(-equinox) |> #remove equinox as by definition it has no impact
  select(-contains("sigma")) |> #these are the error values, we will ignore
  na.omit() #remove NAs
```

We then split the dataframe at random, 20% will be used for testing and 80% for training the model. The reason for choosing 20% is that it accounts for around 4577 rows which is sufficient to test the model.

```
df_ast_big$pha <-  as.factor(df_ast_big$pha)
test_index <- createDataPartition(df_ast_big$spkid, times = 1, p = 0.2, list = FALSE)
test_set <- df_ast_big[test_index, ]
train_set <- df_ast_big[-test_index, ]
```

The Results

Primary Model

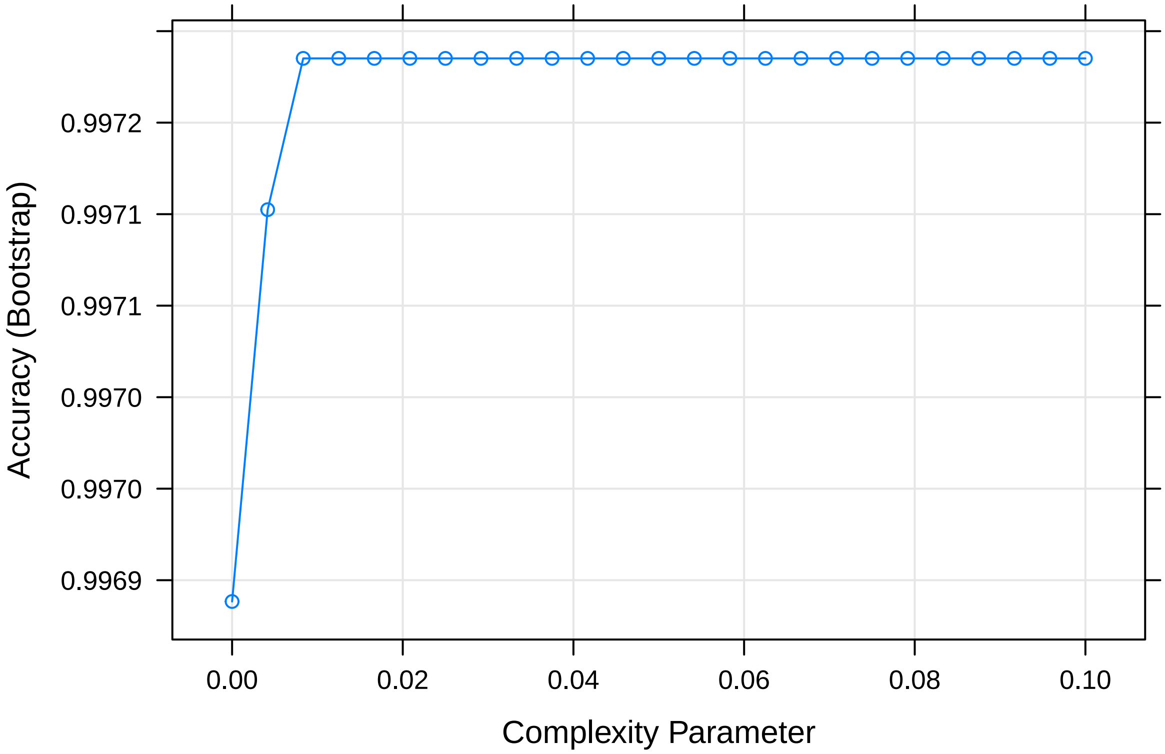
```
###tain GLM, see how well it does
train_glm <- train(pha ~ ., method = "glm", data = train_set)
y_hat_glm <- predict(train_glm, test_set, type = "raw")
mod_glm_bg <- confusionMatrix(y_hat_glm, test_set$pha)$overall[["Accuracy"]]
accuracy_table <- data_frame(Model = "Primary",Method = "GLM", Accuracy = mod_glm_bg)

###train KNN. It will take a while to run. you can skip this seection if you like
train_knn <- train(pha ~ ., method = "knn", data = train_set)
y_hat_knn <- predict(train_knn, test_set, type = "raw")
mod_knn_bg <- confusionMatrix(y_hat_knn, test_set$pha)$overall[["Accuracy"]]

accuracy_table <- bind_rows(accuracy_table,
                          data_frame(Model = "Primary",Method = "KNN", Accuracy = mod_knn_bg))

###train
train_rpart <- train(pha ~ .,
                     method = "rpart",
                     tuneGrid = data.frame(cp = seq(0.0, 0.1, len = 25)),
                     data = train_set)

mod_rpart_bg <- confusionMatrix(predict(train_rpart, test_set), test_set$pha)$overall["Accuracy"]

accuracy_table <- bind_rows(accuracy_table,
                            data_frame(Model = "Primary",Method = "RPART", Accuracy = mod_rpart_bg))
accuracy_table |> knitr::kable()
```

Model	Method	Accuracy
Primary	GLM	0.9609
Primary	KNN	0.9000
Primary	RPART	0.9983

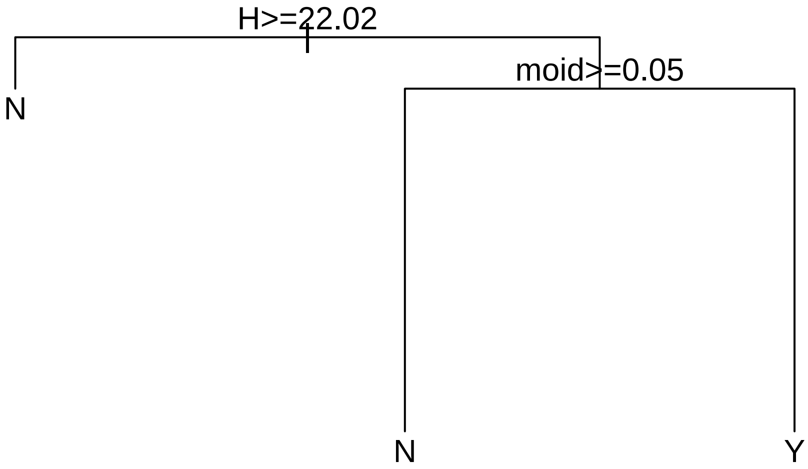
Running the three models: GLM, KNN and decision tree, we are given the following accuracies (Note best *KNN is 9*).

Immediately we notice the very high accuracy of the decision tree which is also evident in the following plot. The plot after this is the actual decision tree in graphic format.

```
##you can see immediately we hit high accuracy
plot(train_rpart)
```

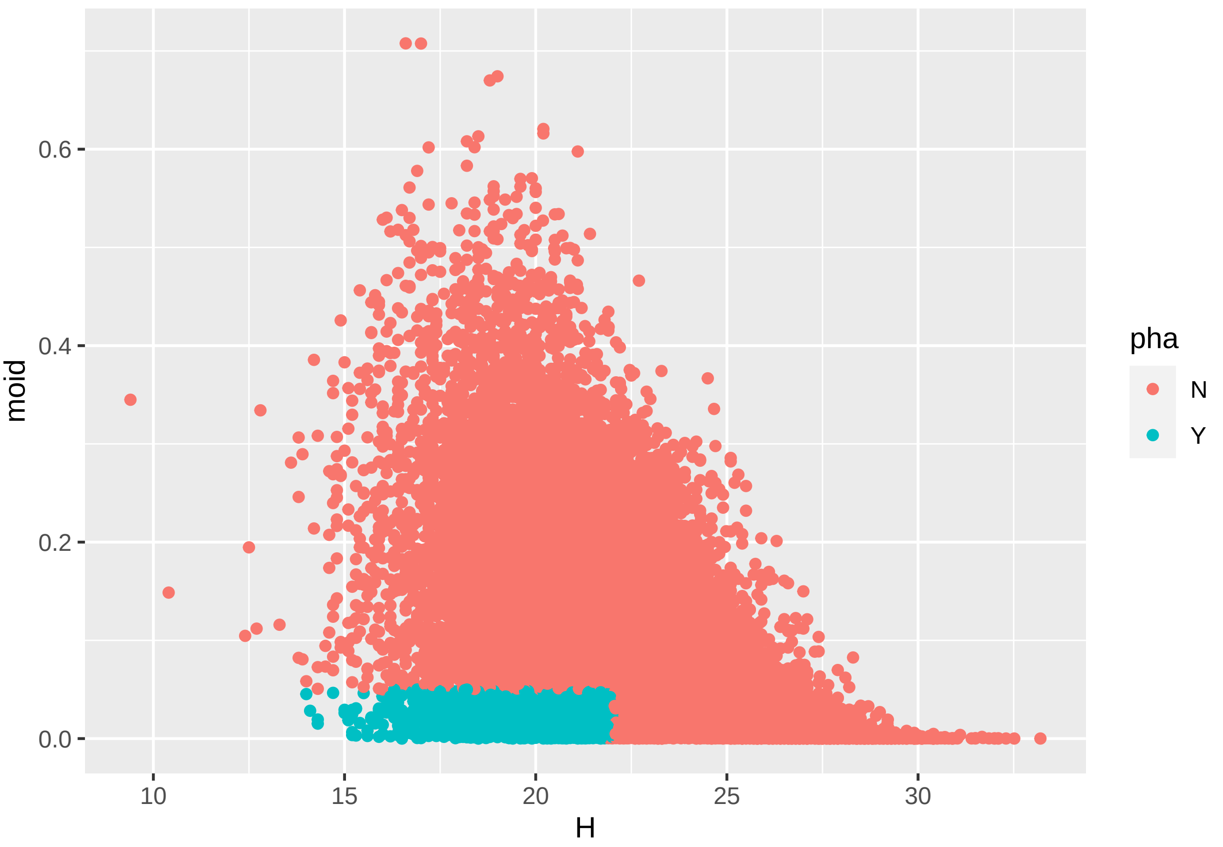


```
# view the final decision tree
plot(train_rpart$finalModel, margin = 0.1) # plot tree structure
text(train_rpart$finalModel) # add text labels
```



Just 2 variables, **MOID** and **H** are enough to reach this accuracy. Lets plot them to visualise this relationship, the color representing the **pha** level.

```
df_ast_big |>
  ggplot(aes(H, moid, color=pha)) +
  geom_point()
```



There is a clear relationship between **H** and **moid** that you could imagine these 2 variable values are ultimately what defines **pha**. Lets test this assumption by training a model using the values from the decision tree where if **H** is less than *22* and **moid** is less than *0.05*, then the asteroid is hazardous.

```
predict_simple <- df |> mutate(predict_val= factor(ifelse(H<22 & moid<0.05,"Y","N")))
mod_simple_bg <- mean(predict_simple$pha==predict_simple$predict_val)

accuracy_table <- bind_rows(accuracy_table,
                            data_frame(Model = "Primary",Method = "SIMPLE", Accuracy = mod_simple_bg))
accuracy_table |>  kable()
```

Model	Method	Accuracy
Primary	GLM	0.9609
Primary	KNN	0.9000
Primary	RPART	0.9983
Primary	SIMPLE	0.9999

We reach almost perfect accuracy which supports our theory that **H** and **moid** variables very likely define **pha** flag. The reason for not reaching 100% is likely due to outliers. We see that only *63* out of *938599* do not follow this rule and these outlier asteroids all have a **H** value which is just on the threshold thus supporting idea that they are likely measurement errors. See list of outliers below.

pha	H	moid
Y	22.4	0.0267
Y	22.3	0.0333
Y	22.3	0.0221
Y	22.2	0.0032
Y	22.2	0.0116
Y	22.2	0.0068
Y	22.2	0.0240
Y	22.1	0.0113
Y	22.1	0.0075
Y	22.1	0.0122
Y	22.1	0.0009
Y	22.1	0.0113
Y	22.1	0.0029
Y	22.1	0.0145
Y	22.1	0.0437
Y	22.1	0.0161
Y	22.1	0.0394
Y	22.1	0.0164
Y	22.1	0.0220
Y	22.0	0.0255
Y	22.0	0.0408
Y	22.0	0.0441
Y	22.0	0.0359
Y	22.0	0.0364
Y	22.0	0.0254
Y	22.0	0.0187
Y	22.0	0.0171
Y	22.0	0.0293
Y	22.0	0.0058
Y	22.0	0.0374
Y	22.0	0.0045
Y	22.0	0.0246
Y	22.0	0.0089
Y	22.0	0.0209
Y	22.0	0.0119
Y	22.0	0.0187
Y	22.0	0.0216
Y	22.0	0.0138
Y	22.0	0.0197
Y	22.0	0.0078
Y	22.0	0.0120
Y	22.0	0.0329
Y	22.0	0.0405
Y	22.0	0.0047
Y	22.0	0.0437
Y	22.0	0.0317
Y	22.0	0.0020
Y	22.0	0.0139
Y	22.0	0.0378
Y	22.0	0.0040
Y	22.0	0.0454
Y	22.0	0.0105

pha	H	moid
Y	22.0	0.0210
Y	22.0	0.0169
Y	22.0	0.0248
Y	22.0	0.0363
Y	22.0	0.0220
Y	22.0	0.0068
N	21.9	0.0210
N	21.9	0.0213
N	21.9	0.0005
N	21.9	0.0442
N	21.9	0.0103

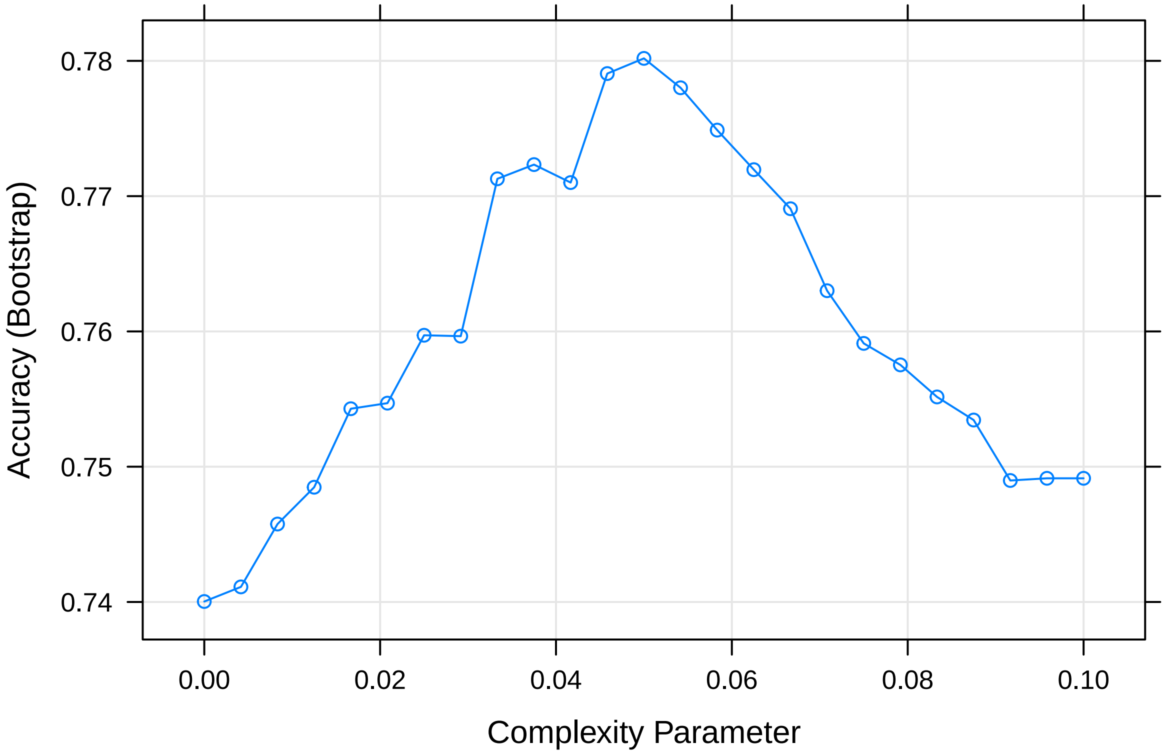
Second Model

Given that we conclude **H** and **moid** essentially define **pha**, it doesn't leave room to see the explanatory value of the other variables. For this second model we will exclude **H** and **moid** from the dataframe and include **diameter** as well **albedo**. The methods will be the same. We get the following results

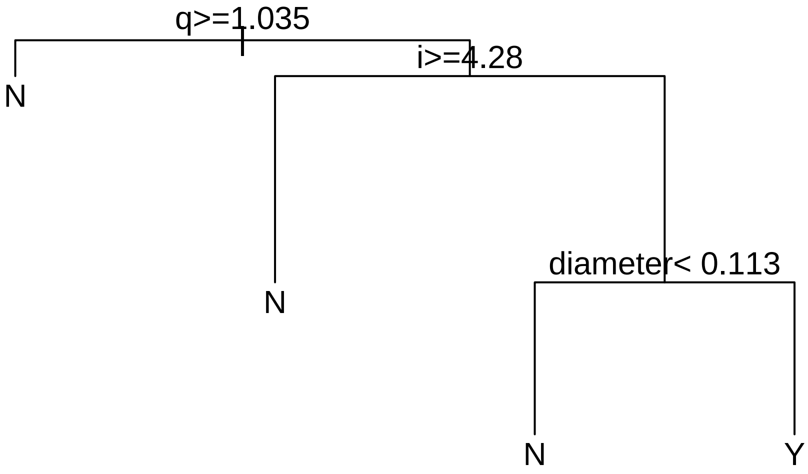
Model	Method	Accuracy
Primary	GLM	0.9609
Primary	KNN	0.9000
Primary	RPART	0.9983
Primary	SIMPLE	0.9999
Secondary	GLM	0.7250
Secondary	KNN	0.7438
Secondary	RPART	0.8062

We see RPART gives us the best accuracy although not very high. Below we plot the accuracy chart and the decision trees

```
plot(train_rpart)
```



```
plot(train_rpart$finalModel, margin = 0.1) # plot tree structure
text(train_rpart$finalModel) # add text labels
```



Following the decision tree, lets try to describe the results in layman terms. If the asteroid's orbit's closest point to the sun is less than *1.035 AU* (**q**), the orbit inclination is not greater than *4.292 degrees* (**i**) and the asteroid **diameter** is greater than *113 metres* then there is an **0.8062** chance the asteroid is hazardous.

This makes sense because when googling, we find earth's **q** to be *0.9832899 AU* which is close to the decision tree **q**, and you imagine the lower the inclination to earths orbit **i**, the more chance it could hit earth. For **diameter** you can imagine if the asteroid is very small (remember median diamater was *4000 metres* and 1st quartile was *2800 metres*) then the asteroid would just burn up in earths atmosphere (note, i have no physics background so i could be very wrong on this last assumption).

Conclusion

In this project we first gave our reasons for wanting to predict the **pha** flag over the **neo** flag. Through several visualisations we saw how the **diameter** and **albedo** contained a lot of *NAs* however the former variable did show it may have some predictive power. This helped us to decide to run ML methods on two models, the *primary* containing the most rows and the *secondary* without **H** and **moid** but including **diameter** and **albedo**. In the *primary model* we discovered that **H** and **moid** are enough to predict **pha** at 100% accuracy if it was not for assumed outliers. In the *secondary model* we discover that other variables can predict the correct answer 81% of the time.

If our assumption is that **H** and **moid** are the only variables used to determine **pha** then it might cloud the understanding by astrophysicists that there are also other variables that could be used should **H** and **moid** not be available. The limitations in this project is that apart from the description of the dataset, this data analysis was done by someone with no formal astrophysics education and not in consultation with some who has this background so there may be some incorrect assumptions.

My recommendation is for the astrophysics community to look to see if **diameter** could be used in partially determining if an asteroid is hazardous as currently it seems only asteroid magnitude from earth **H** and it's orbit's closest point to earth **moid** is being used.

Thank you for the attention. This was fun!