Is the Asteroid Hazardous?

Report 02

Brendan D. Cubberly

Explorations in Data Science – MAT 342
11 November 2022

Section 01: Data Exploration

After analyzing the data to observe if the asteroid is hazardous to Earth, I found that there are certain factors of the asteroid that are more dangerous than others. I observed the motion of the asteroid, the magnitude, the minimum diameter of the asteroid, the approaching date, and the minimum orbit intersection. Below are the first two models I observed, which examines the average motion of the asteroid. Based on Figure 1, we can see that there is no real trend in the data. Rather, the data is decreasing as the motion increases, and the hazardous level decreases. However, in Figure 2, we can see that as the motion increases, so does the hazardous level. Figure 2 allows us to see that as the motion of the asteroid increases, so does the hazardous level.

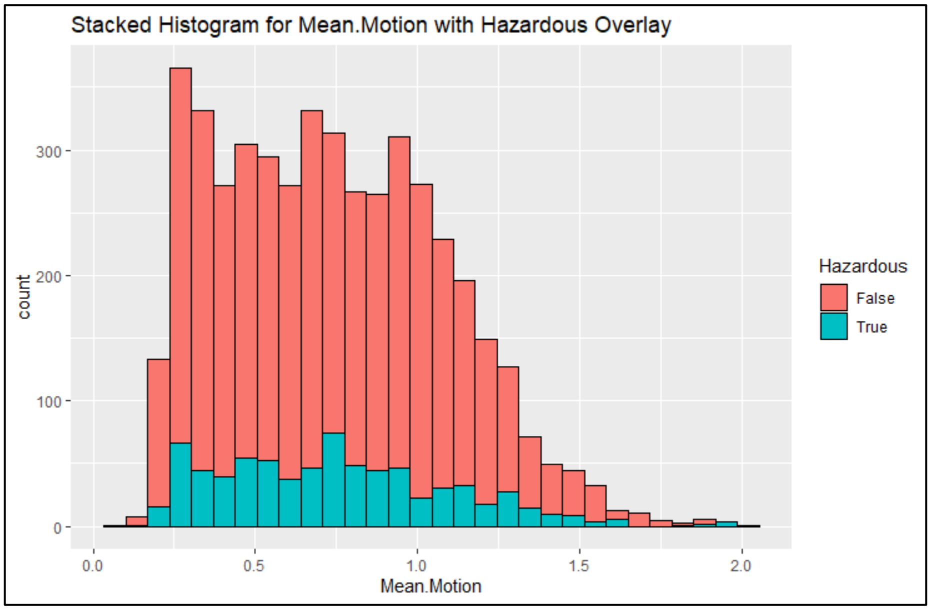


Figure 1: Non-Normalized (Stacked) Histogram for Mean.Motion with Hazardous Overlay

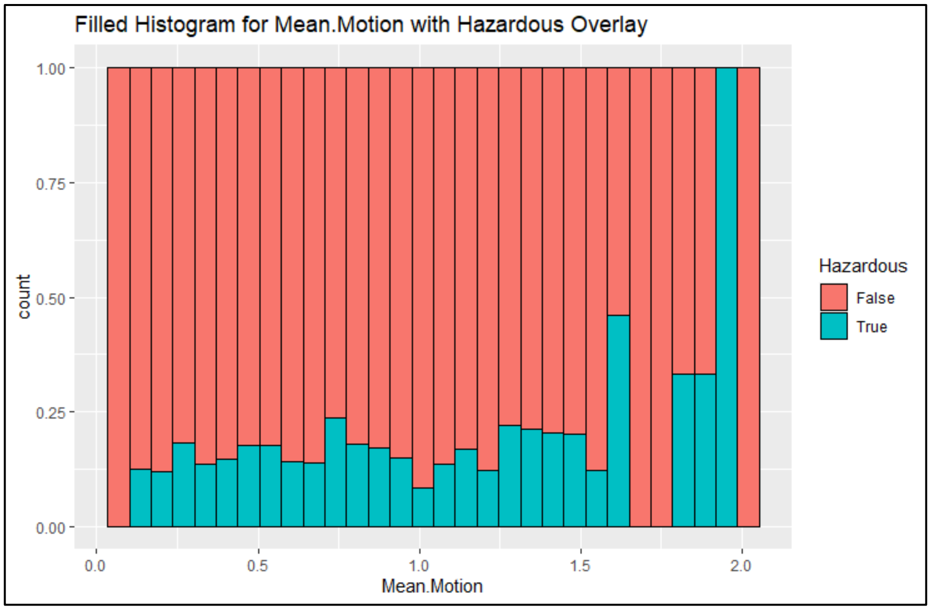


Figure 2: Filled (Normalized) Histogram for Mean.Motion with Hazardous Overlay

As we can see in Figures 3 and 4, the data does not have a particular trend except that the asteroid is most hazardous when the magnitude is around 20. With the similar data being present in the same magnitude level between both figures, we can expect this variable to appear in the model. We can see different instances in Figures 5 through 10.

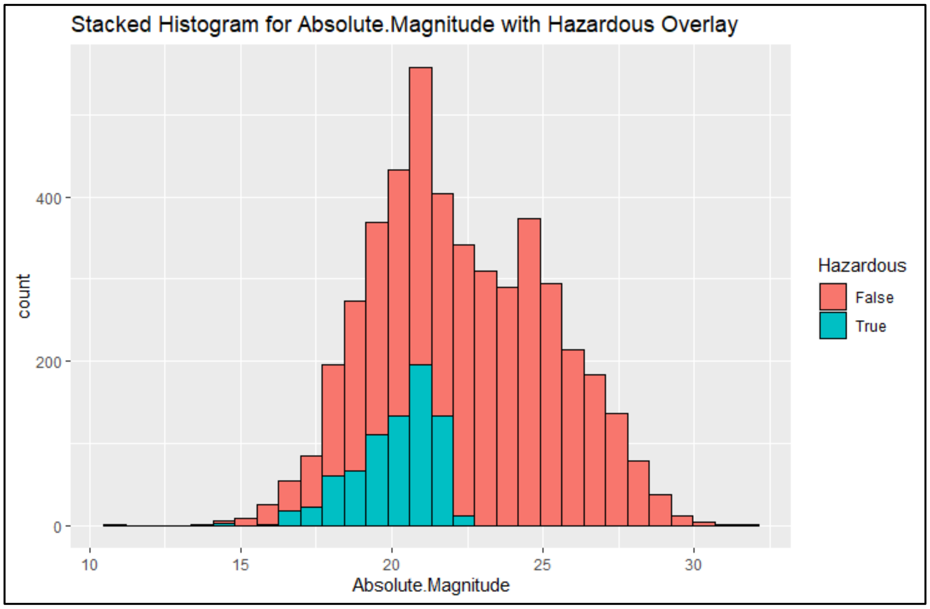


Figure 3: Stacked (Non-Normalized) Histogram for Absolute.Magnitude with Hazardous Overlay

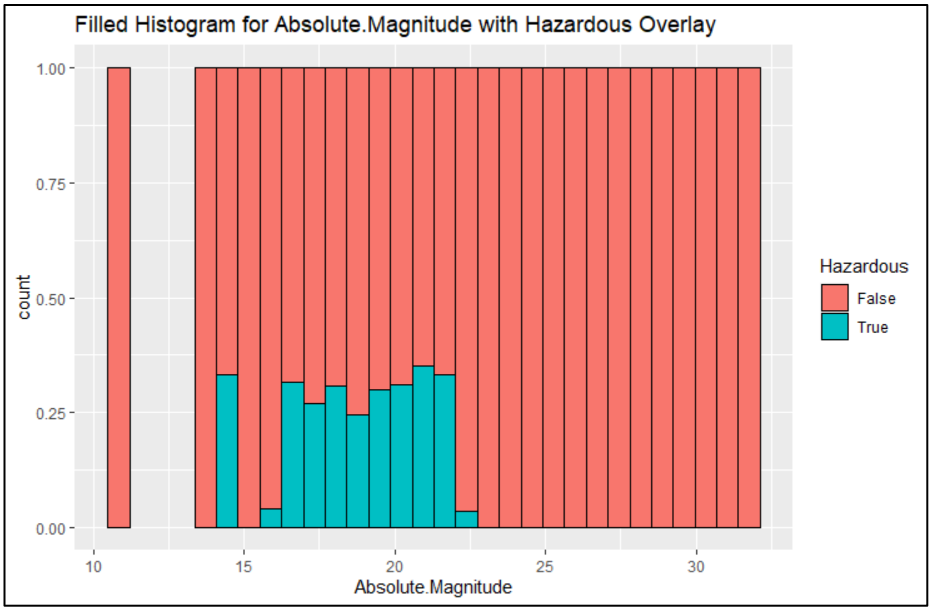


Figure 4: Filled (Normalized) Histogram for Absolute.Magnitude with Hazardous Overlay

Here, the variable observed is the diameter in kilometers, and we can see from Figures 5 and 6 that there is no trend and that there is no point where the data proves to be truly hazardous. We have a large gap in both Figures, and Figure 6 shows the data overall to be non-hazardous.

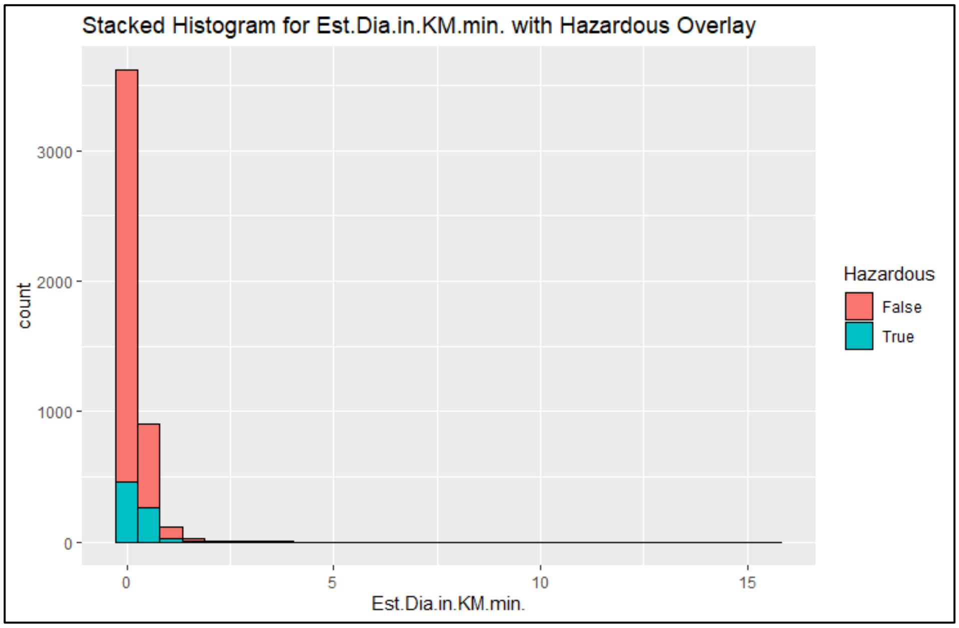


Figure 5: Stacked (Non-Normalized) Histogram for Est.Dia.in.KM.min. with Hazardous Overlay

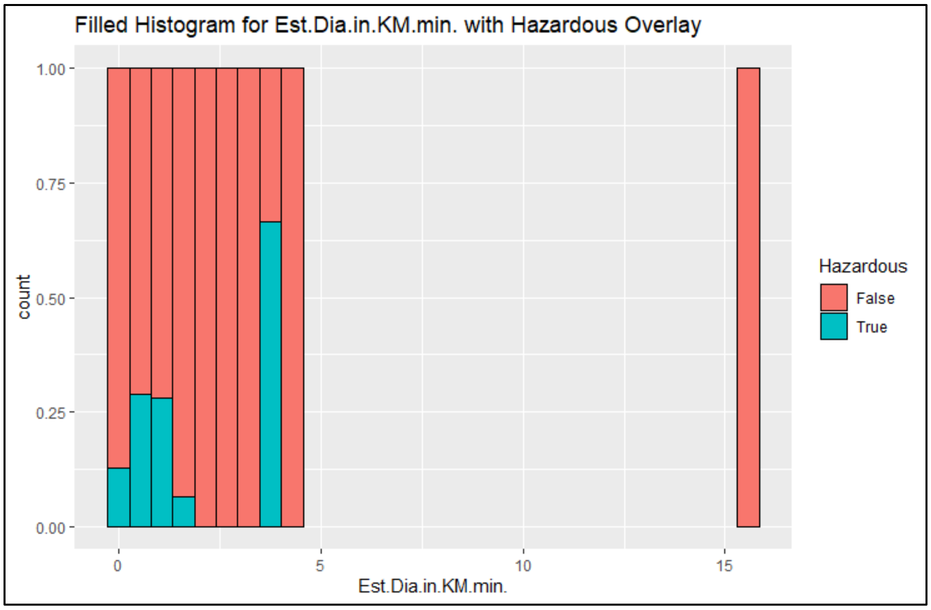


Figure 6: Filled (Normalized) Histogram for Est.Dia.in.KM.min. with Hazardous Overlay

For this variable, we can observe that there is once again no clear trend in the data. Figure 8 even shows that as the date approaches, that the data becomes less hazardous. Figure 8 is filled in with data that shows the variable is not a strong factor to observe if the asteroid is hazardous or not.

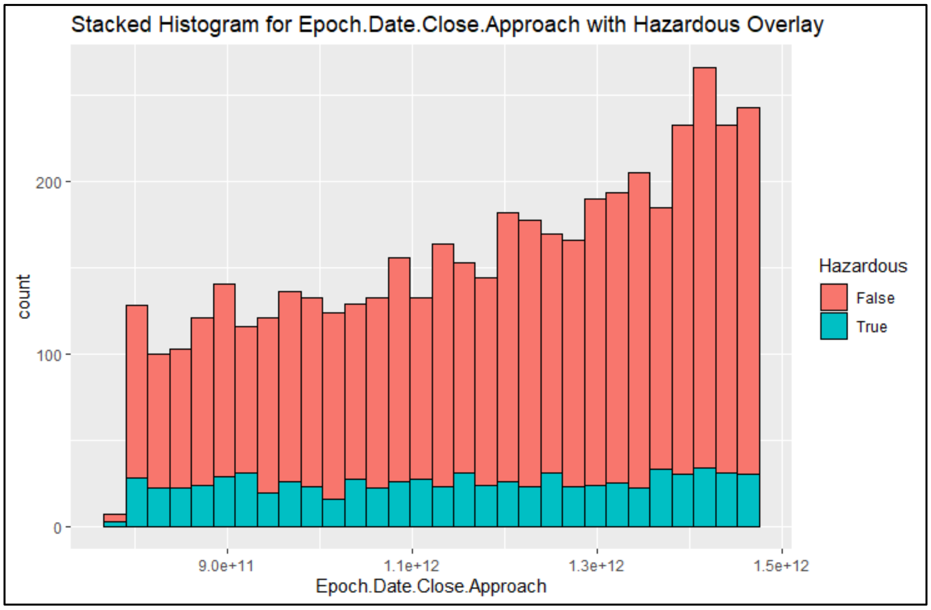


Figure 7: Stacked (Non-Normalized) Histogram for Epoch.Date.Close.Approach with Hazardous Overlay

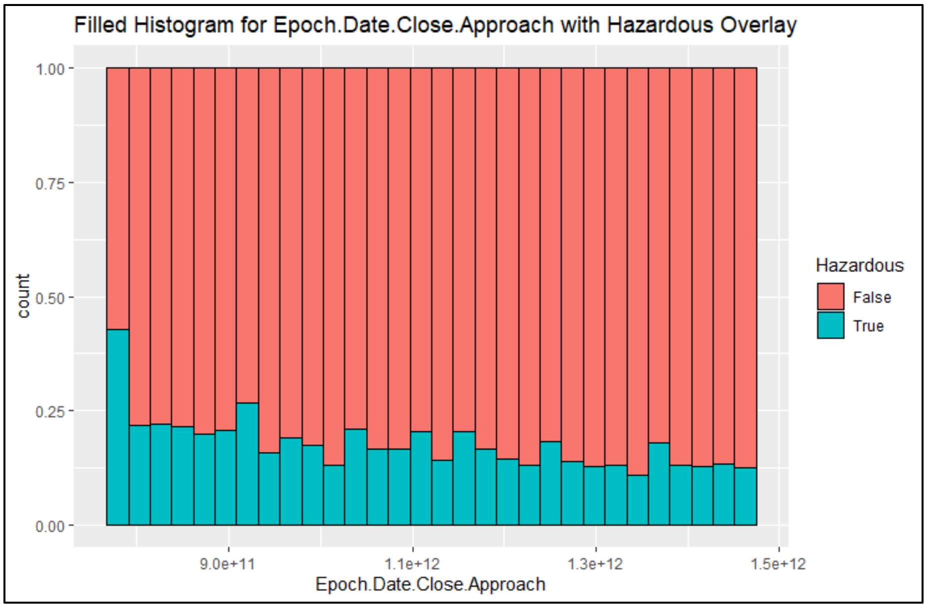


Figure 8: Filled (Normalized) Histogram for Epoch.Date.Close.Approach with Hazardous Overlay

The minimum orbit intersection data presented in Figures 9 and 10 show that as the orbit intersection increases, the asteroid gets to be less hazardous. The data is only true when the orbit intersection is between 0.0 and stops around 0.5. However, like the absolute magnitude, both Figures 9 and 10 have similar data, meaning we can expect to see it in our model.

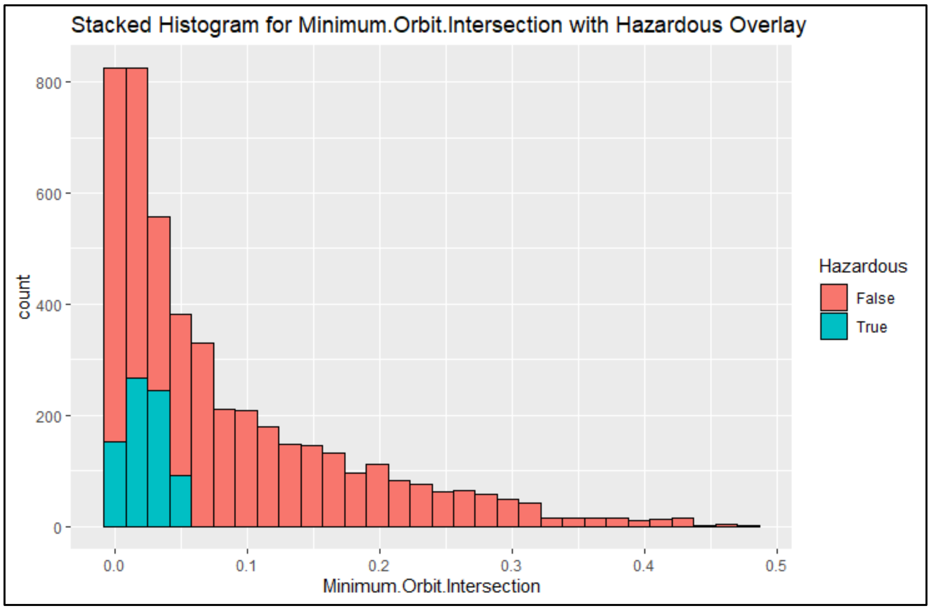


Figure 9: Stacked (Non-Normalized) Histogram for Minimum.Orbit.Intersection with Hazardous Overlay

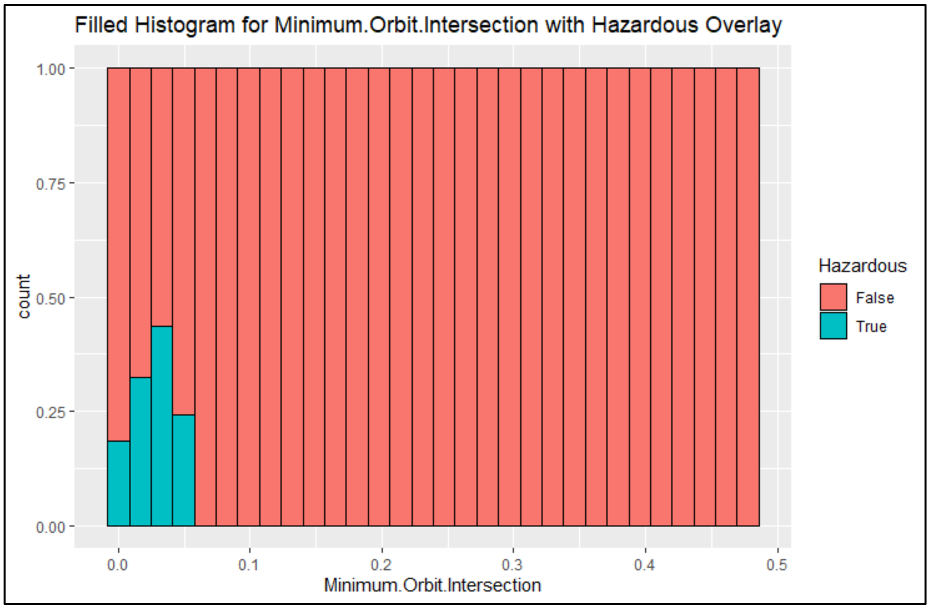


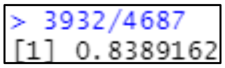
Figure 10: Filled (Normalized) Histogram for Minimum.Orbit.Intersection with Hazardous Overlay

Section 02: Prepare to Model the Data

To ensure that the data is correct, I made multiple models to connect the data together, i.e., validate the partition. I used boxplots for each of the variables that were tested, as well as a CART model (Figure 11) to show that there is a connection between the variables and if they are hazardous. From Figure 11 below, we can see that both absolute magnitude and minimum orbit intersection appeared in the model.

In Figures 12 through 16, we can examine the connection between the variables through the boxplots. If each boxplot in the model appears to be the same to one another, then the connection between the variables is accurate. For example, in Figure 12, we can see that the two boxplots are nearly identical to one another. Therefore, the data is validated. The same applies for the rest of the boxplots, Figures 13-16.

One factor to examine from Figure 11 is the numbers present in the model. Here, we have different segments, or nodes, to help us examine the data. This model is an All-False baseline model which presents all non-hazardous information before the data is split. From here, we can see that the root node is the node labeled with 100%. This root node is especially important because it allows us to see how accurate the model is. Essentially, we look for the baseline accuracy.



```
> 3932/4687
[1] 0.8389162
```

The baseline accuracy of Figure 11 is approximately 83.89%.

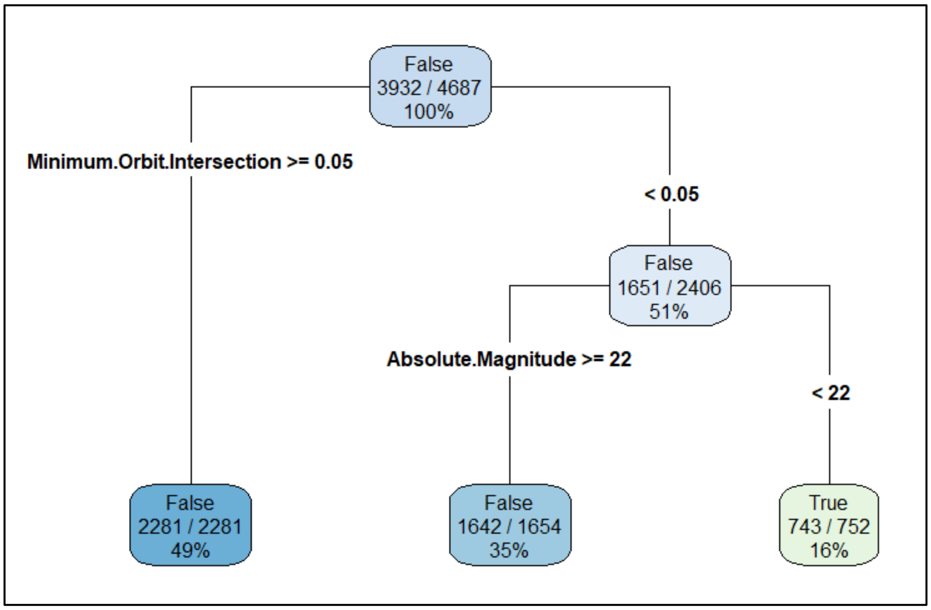


Figure 11: CART Model to Validate Partition of Hazardous Target Variable

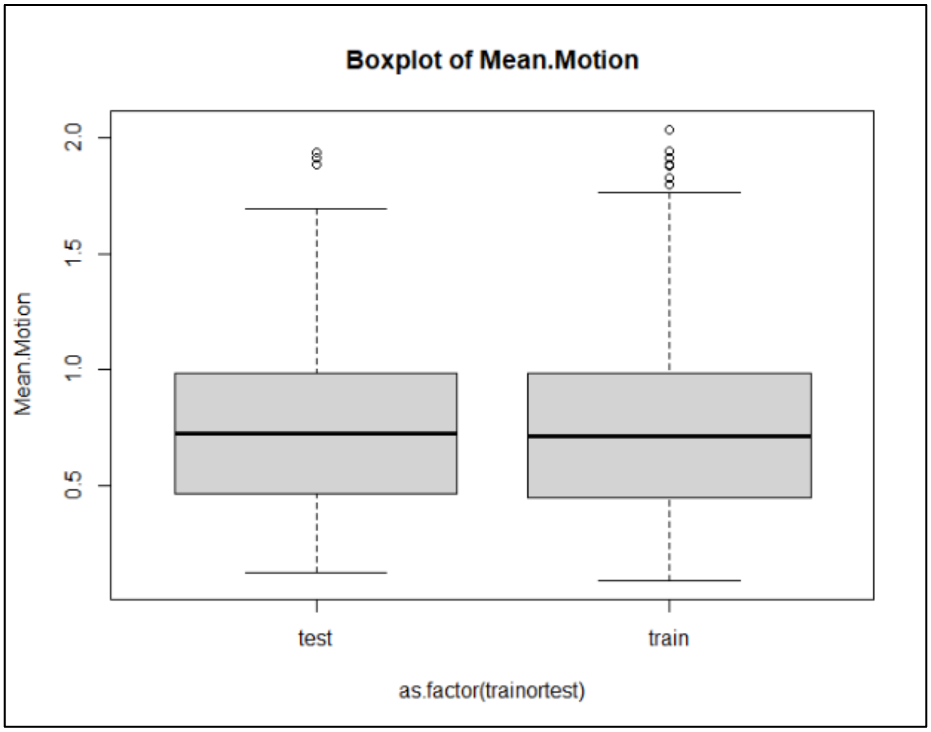


Figure 12: Boxplot of Mean.Motion for Train or Test Data

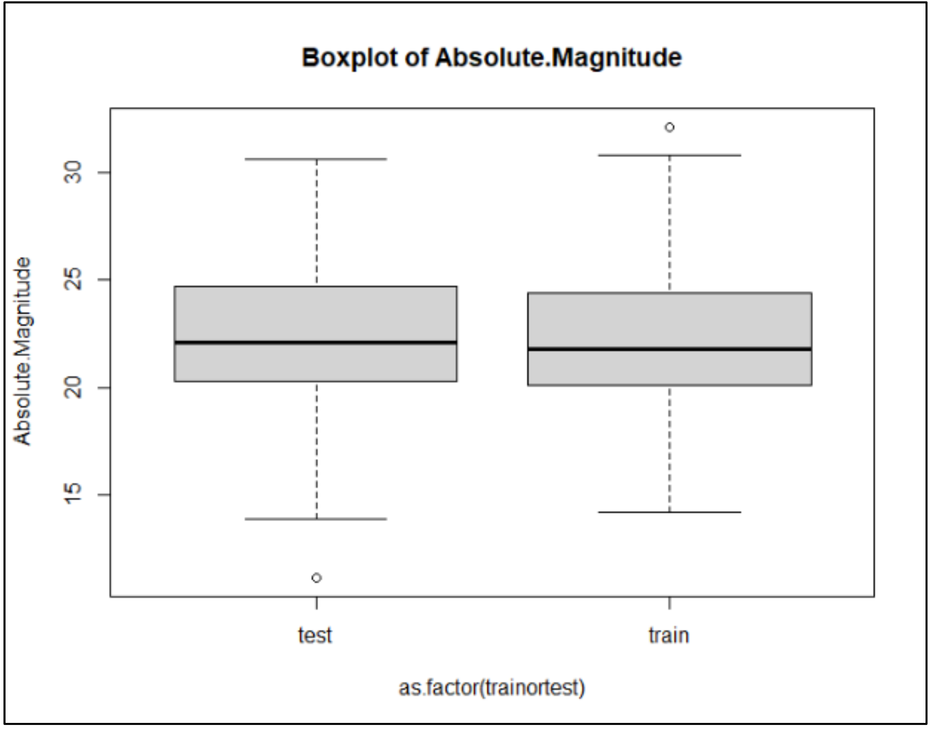


Figure 13: Boxplot of Absolute.Magnitude for Train or Test Data

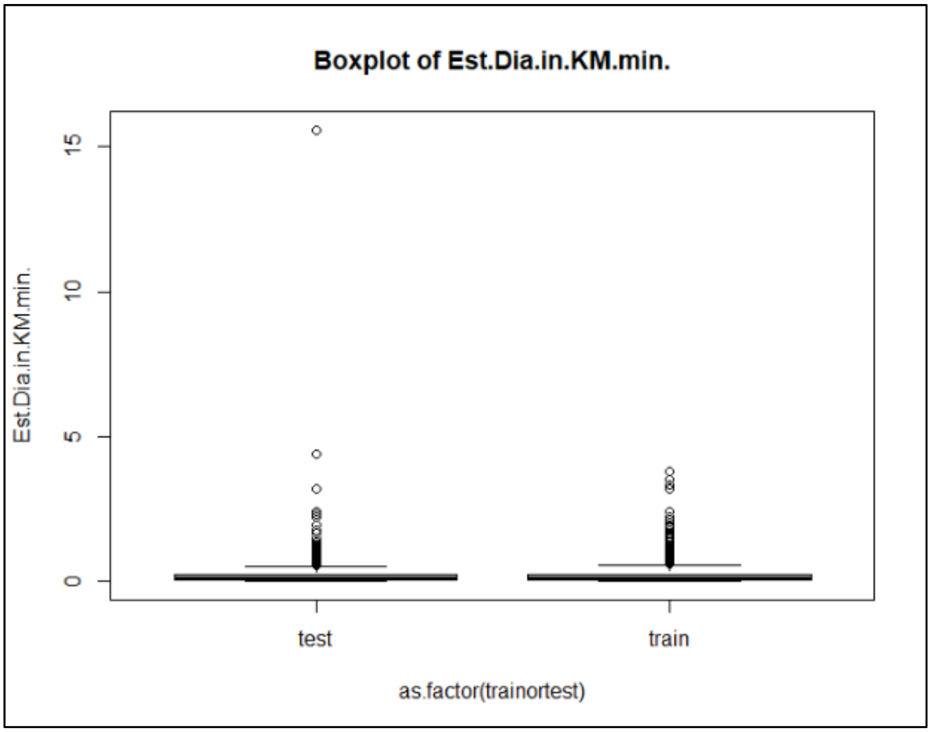


Figure 14: Boxplot of Est.Dia.in.KM.min. for Train or Test Data

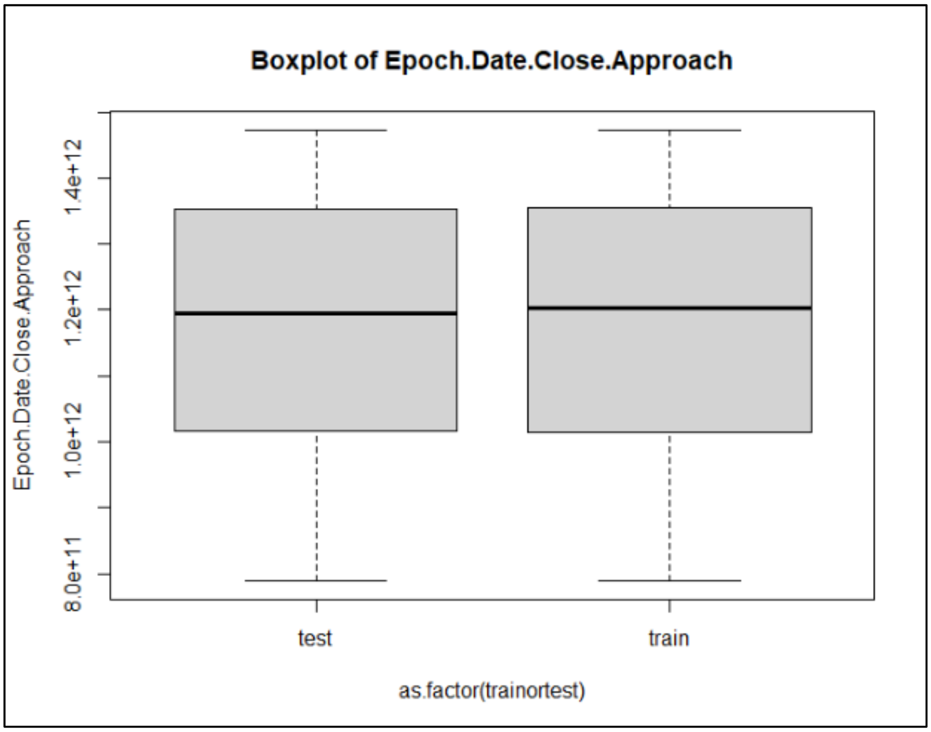


Figure 15: Boxplot of Epoch.Date.Close.Approach for Train or Test Data

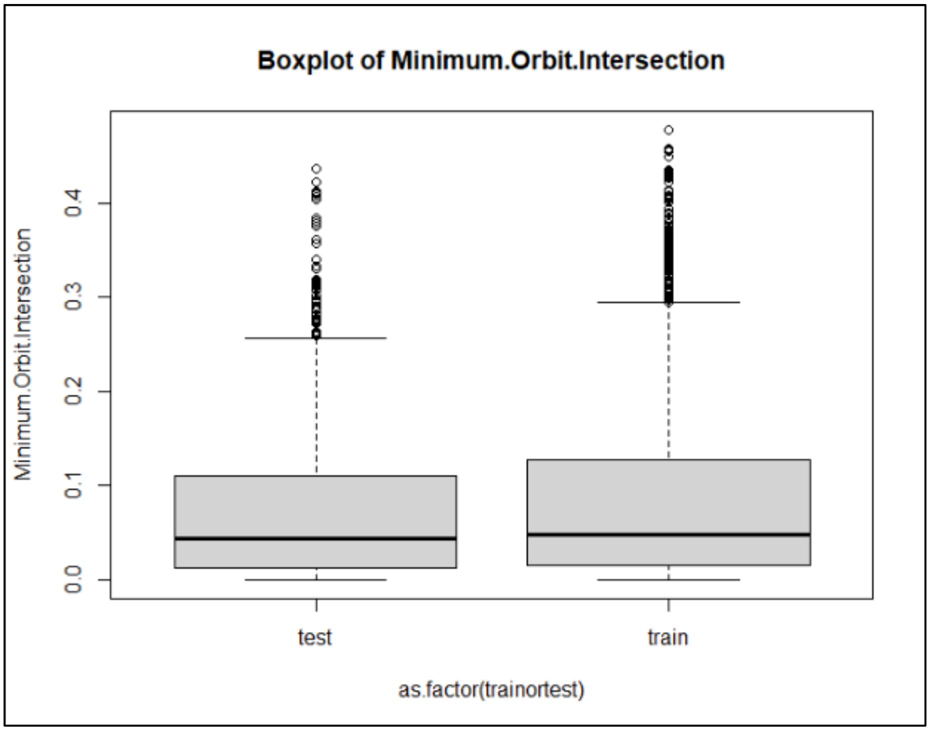


Figure 16: Boxplot of Minimum.Orbit.Intersection for Train or Test Data

Section 03: CART Models

To examine the way the 10-fold CART model splits, we can look at the decision rules created by the tree. Essentially, decision trees split based on the information given in the data, and when we use cross-validation we can further examine the reasoning behind the splits. In Figure 17, we can see that the model is an All-False model, and the root node is split between .84 and .16. Essentially, 84% of the minimum.orbit.intersection data is greater than .05, is not hazardous, while 16% of the data is less than or equal to. Since it is an all-false model, and by our Figures in section 1, most of the data in the models will not be hazardous. This model then splits for the variable absolute.magnitude, where the hazardousness level is true 16%, and false 35%. By looking at the bottom nodes, we can see that all the nodes added together sum to 100%.

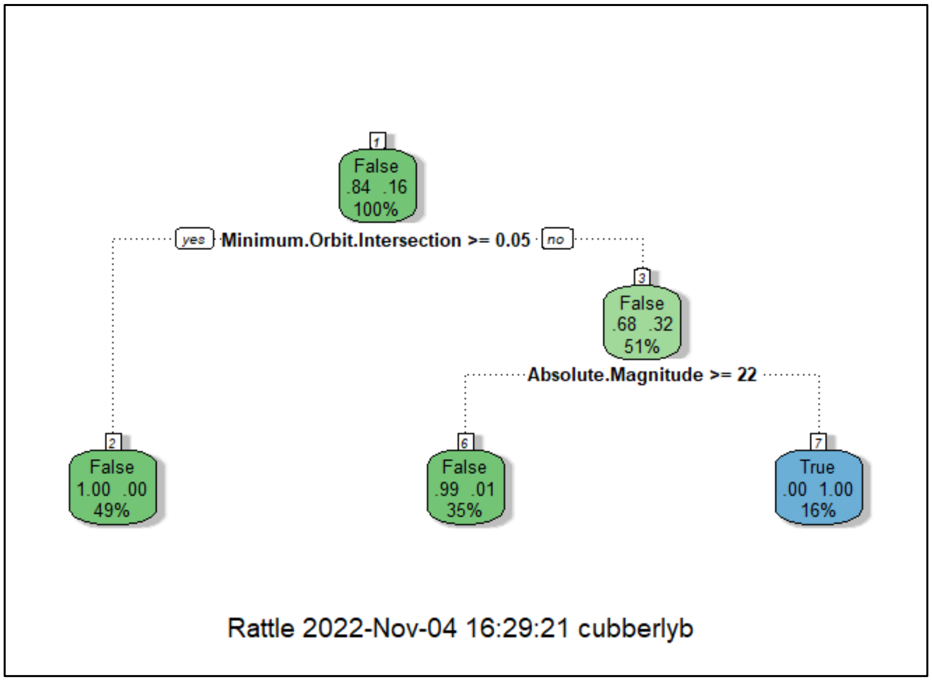


Figure 17: 10-Fold Cross Validation CART Model of Hazardousness

It is also important to view the accuracies of each model, both the training data set and the testing data set. Through coding, I was able to obtain an accuracy of 99.52% of the training data set before the 10-fold cross validation. I also obtained an accuracy of 99.66% of the test data before cross validation. These two accuracies are extremely close, and they are also extremely accurate. However, when I calculated accuracies on the 10-fold cross validation training set, I once again obtained an accuracy of 99.52%. However, in the test data set, my accuracy was 99.57%. Once again, these accuracies are extremely close and accurate. One thing to note is that in the baseline training data set and the 10-fold cross validation set, the accuracies are the exact same. This could be an indicator of overfitting for the data set. However, for the test data sets, the accuracies were close but different overall. However, I am pleased to say that the test data for the 10-fold cross validation model had a slightly larger accuracy than the baseline model, even though both models were very similar.

Section 04: Bringing it all Together

The 10-fold cross validation model had a better accuracy overall, but it was only by .05%. This is not a drastic difference in the data, which helps explain that either the baseline model or the 10-fold CART model would be an acceptable model to use to evaluate the hazardousness of the asteroid based on the data. However, this means that the k-fold model did better meet my expectations because of the slightly higher accuracy.

Appendix (R Code)

```
d <- nasa
d <- d[,-c(1, 2, 12, 21:24, 39) ]
d$Hazardous <- as.factor(d$Hazardous)
install.packages("caret")
library(caret)
#Question 1
summary(nasa)
ggplot(nasa, aes(Mean.Motion)) +
  geom_histogram(aes(fill = Hazardous),
                 color = "black", position = "stack") + ggtitle("Stacked Histogram for Mean.Motion
with Hazardous Overlay")
ggplot(nasa, aes(Absolute.Magnitude)) +
  geom_histogram(aes(fill = Hazardous),
                 color = "black", position = "stack") + ggtitle("Stacked Histogram for
Absolute.Magnitude with Hazardous Overlay")
ggplot(nasa, aes(Est.Dia.in.KM.min.)) +
  geom_histogram(aes(fill = Hazardous),
                 color = "black", position = "stack") + ggtitle("Stacked Histogram for
Est.Dia.in.KM.min. with Hazardous Overlay")
ggplot(nasa, aes(Epoch.Date.Close.Approach)) +
  geom_histogram(aes(fill = Hazardous),
                 color = "black", position = "stack") + ggtitle("Stacked Histogram for
Epoch.Date.Close.Approach with Hazardous Overlay")
ggplot(nasa, aes(Minimum.Orbit.Intersection)) +
  geom_histogram(aes(fill = Hazardous),
                 color = "black", position = "stack") + ggtitle("Stacked Histogram for
Minimum.Orbit.Intersection with Hazardous Overlay")


ggplot(nasa, aes(Mean.Motion)) +
  geom_histogram(aes(fill = Hazardous),
                 color = "black", position = "fill") + ggtitle("Filled Histogram for Mean.Motion with
Hazardous Overlay")
ggplot(nasa, aes(Absolute.Magnitude)) +
  geom_histogram(aes(fill = Hazardous),
                 color = "black", position = "fill") + ggtitle("Filled Histogram for Absolute.Magnitude
with Hazardous Overlay")
ggplot(nasa, aes(Est.Dia.in.KM.min.)) +
  geom_histogram(aes(fill = Hazardous),
                 color = "black", position = "fill") + ggtitle("Filled Histogram for Est.Dia.in.KM.min.
with Hazardous Overlay")
```

```
ggplot(nasa, aes(Epoch.Date.Close.Approach)) +
  geom_histogram(aes(fill = Hazardous),
                 color = "black", position = "fill") + ggtitle("Filled Histogram for
Epoch.Date.Close.Approach with Hazardous Overlay")
ggplot(nasa, aes(Minimum.Orbit.Intersection)) +
  geom_histogram(aes(fill = Hazardous),
                 color = "black", position = "fill") + ggtitle("Filled Histogram for
Minimum.Orbit.Intersection with Hazardous Overlay")

#Question 3
inTraining <- createDataPartition(y= d$Hazardous,
                                  p = .75,
                                  list = FALSE)
#training data
d.train <- d[ inTraining ,]
dim(d.train)
# testing data
d.test <- d[-inTraining,]

d.train$trainortest <-
  rep("train", nrow(d.train))
d.test$trainortest <-
  rep("test", nrow(d.test))
d.all <- rbind(d.train, d.test)
#Target variable / Hazardous
boxplot(Hazardous ~ as.factor(trainortest), data = d.all)

kruskal.test(Hazardous ~ as.factor(trainortest), data = d.all)$p.value

#Mean.Motion

boxplot(Mean.Motion ~ as.factor(trainortest), data = d.all, main = "Boxplot of Mean.Motion")

kruskal.test(Mean.Motion ~ as.factor(trainortest), data = d.all)$p.value

#Absolute.Magnitude

boxplot(Absolute.Magnitude ~ as.factor(trainortest), data = d.all, main = "Boxplot of
Absolute.Magnitude")

kruskal.test(Absolute.Magnitude ~ as.factor(trainortest), data = d.all)$p.value

#Est.Dia.in.KM.min.
```

```
boxplot(Est.Dia.in.KM.min. ~ as.factor(trainortest), data = d.all, main = "Boxplot of Est.Dia.in.KM.min.")

kruskal.test(Est.Dia.in.KM.min. ~ as.factor(trainortest), data = d.all)$p.value

#Epoch.Date.Close.Approach

boxplot(Epoch.Date.Close.Approach ~ as.factor(trainortest), data = d.all, main = "Boxplot of Epoch.Date.Close.Approach")

kruskal.test(Epoch.Date.Close.Approach ~ as.factor(trainortest), data = d.all)$p.value

#Minimum.Orbit.Intersection

boxplot(Minimum.Orbit.Intersection ~ as.factor(trainortest), data = d.all, main = "Boxplot of Minimum.Orbit.Intersection")

kruskal.test(Minimum.Orbit.Intersection ~ as.factor(trainortest), data = d.all)$p.value

#4
install.packages("rpart")
library(rpart)
cart01 <- rpart(Hazardous ~., data = d, method = "class")
cart01
library(rpart.plot)
rpart.plot(cart01, type = 4, extra = 102)

3932/4687
#7
#training
cart.train <- rpart(Hazardous ~., data = d.train, method = "class")
rpart.plot(cart.train, type = 4, extra = 102)

pred.cart <- predict(object = cart.train, newdata = d.train,
                    type = "class")
table(d.train$Hazardous, pred.cart)
2948 +551

dim(d.train)[1]
3499/3516

#test
```

```
cart.test <- rpart(Hazardous ~., data = d.test, method = "class")

pred.cart.test1 <- predict(object = cart.test, newdata = d.test, type = "class")

table(d.test$Hazardous, pred.cart.test1)
(980 + 187)/1171

#5
install.packages("rattle")
library(rattle)
library(caret)
set.seed(700)
train.control <- trainControl(method = "cv",
                              number = 10)
d.train.cca <- na.omit(d.train)
model <- train(Hazardous ~., data = d.train.cca,
               method = "rpart",
               trControl = train.control)
model
plot(model$finalModel)
text(model$finalModel)
fancyRpartPlot(model$finalModel, cex = .8)
pred.kfold.train <- rpart.predict(object = model, newdata = d.train)
table(d.train$Hazardous, pred.kfold.train)
(2948+551)/3516
pre.kfold.test <- rpart.predict(object = model, newdata = d.test)
table(d.test$Hazardous, pre.kfold.test)
(983+183)/1171
```