BRE KING NEWS: THIS ROCK MIGHT BE PROBLEM

Sofia De ngelis, Lima Popal, Leland Russell

1. BSTRACT

Near-Earth Objects (NEOs) are objects which pass within close proximity to Earth. N S has developed a system to classify NEOs as Potentially Hazardous Objects (PHOs), however these methods are not well documented publicly. Using over a century of observational data compiled by N S , we explore the features associated with the PHO classification. We applied both tree-based classification models and multilevel logistic regression to predict N S s classification based on NEOs' luminosity, relative velocity to earth, and miss distance from earth. Results suggest that absolute magnitude is the most significant predictor. Tree models were marginally informative, while logistic regression had high accuracy.

2. Introduction

On November 13, 2020 at 5:20 UTC, an asteroid with a diameter around 7 meters passed within 400 kilometers of the South Pacific Ocean[1]. This asteroid has been named the "2020 VT_4", and is the closest recorded passing of an asteroid to Earth without entering the atmosphere. Objects like the 2020 VT_4 are part of a category of celestial bodies known as "Near Earth Objects" (NEOs)—asteroids or comets who come close to Earth without colliding. Figure 1 shows a computer rendering of an NEO in New York City[2].

Figure 1: There was a concern that the 2018 LF_{16} asteroid could collide with Earth. This shows a (computer rendered) size comparison of the 2018 LF_{16} and New York City.

Fortunately, many NEOs are harmless despite their proximity to Earth. Heavier bodies often pass Earth before their orbits are meaningfully attracted by Earth's gravity. On the other end, lighter bodies pose little threat, even when close to Earth. However, there are many NEOs that warrant caution. N S has developed a system

to categorize the hazard from NEOs by designating certain objects as "Potentially Hazardous Objects" (PHOs). These objects often display particular characteristics such as large size or close approaches that warrant attention.

The method by which an object is classified as hazardous though is unclear. Hazardous in what sense? If an object could enter the atmosphere, but burn in the atmosphere before reaching the surface, is this hazardous? Through this project, we aim to infer which features are associated with N S 's declaration. To do this, we will explore a century's worth of NEOs observations and compare that to N S 's classification of the object as potentially hazardous or not.

3. Data

3.1. **Data cquisition & Introduction.**

The data in this project were obtained from Kaggle[3], which were in turn scraped from N S 's NEO Web Service PI[4]. Details on this are contained in a Github page . The data represent observational records of NEOs from 1910 to 2024, 338,171 observations in total. These were collected and compiled by N S 's Center for Near Earth Object Studies. lso of note is the Jet Propulsion Lab[6] who has a similar dataset, excluding the hazardous classification, but including variable descriptions—we used this source for variable descriptions and details.

The data contain measurements, estimated values, and auxiliary data about the objects. The variables are as follows[6]:

`Neo id` - unique identification number assigned to each asteroid.
`Name` - name give by N S , similar to neo id.
`bsolute Magnitude` - Unitless measure of intrinsic luminosity.
`Estimated Minimum Diameter` (km) - Minimum diameter estimate.
`Estimated Maximum Diameter` (km) - Maximum diameter estimated.
`Orbiting Body` - Which planet that the NEO passes.
`Relative Velocity` (km/h) - Velocity relative to Earth.
`Miss Distance` (km) - Distance missed from center of NEO to center of Earth.
`Is Hazardous` - Indicator of whether an NEO is classified as hazardous.

It's important to note that there is no indication that any of the diameter values were directly measured. When not observed, diameter is estimated using the following formula[7]:

$$\log_{10}(d) = 3\ 1236 \quad 0\ 5 \log_{10} p \quad 0\ 2\mathrm{H}, \tag{3.1}$$

where d is the diameter, p is the geometric albedo (not included in these data), and H is the absolute magnitude.

3.2. **Ethical Considerations.**

The data in this project represent over a century of observational records of NEOs. Unsurprisingly, older observations aren't as detailed because astronomical technology was not as developed in the beginning of the 20th century. This means, for example, that older equipment was not able to identify smaller NEOs. This likely means that smaller objects are underrepresented in earlier data, creating potential gaps by skewing towards bigger objects.

Furthermore, as mentioned earlier, these data only show objects which were near Earth. Objects which actually collided with Earth do not seem to be present in the data. This creates bias in our data since we only can use observations in which there were no collisions. There may be a difference between potentially hazardous objects and truly hazardous objects, and we only have data for the former. Furthermore, the methods by which N S records the data are unclear. There is one observation per instance of the object passing Earth. This is unrealistic though; we would expect that N S would continually monitor objects at least until they are headed away from Earth—these data do not represent this.

We also do not entirely know what it means for an NEO to be potentially hazardous. Misinterpretation of this classifier and catastrophizing the data is quite easy given the relatively large number of PHOs and ambiguity in the term hazardous. It is important to consider the relatively low frequency with which objects collide with Earth in a meaningful way before making conclusions with these data.

3.3. **Data Cleaning.**

For estimated diameter variables, while the Jet Propulsion Lab states that true values are included if known, there are no indications that any observation of diameter is truly known. Further, as figure 2 shows, there is a nearly perfect trend between the diameter variables and absolute magnitude. Considering this, combined with the knowledge of how they were predicted in equation 2.1, we removed the diameter variables on the grounds that they do not provide more information about any particular NEO.

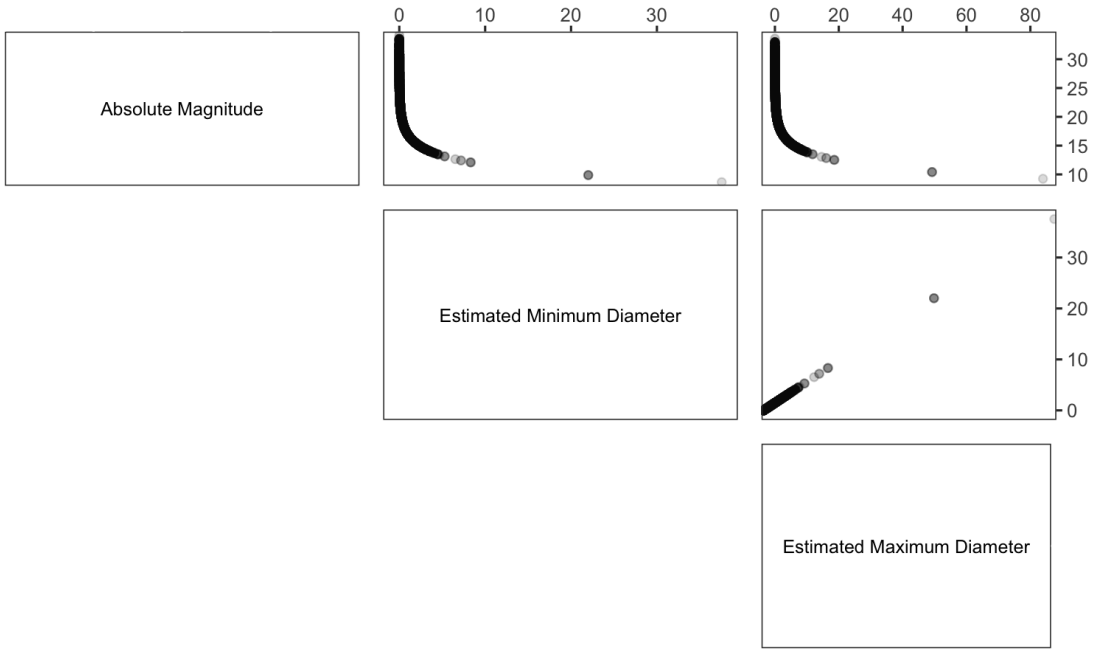

Figure 2: Pairwise scatterplots of the diameter variables and absolute magnitude.

In addition to removing the diameter variables, we also removed the orbiting body variable and all N observations from the data. There was only one value for orbiting

body, that being Earth, so this did not change the data. Interestingly, there were very few—only 28 rows—with N s, so the overall data seem to be largely unchanged in removing these.

4. Exploratory Data nalysis

4.1. **Data Summaries and Visualization.**

Initial investigations of the data show many repeated observations from particular NEOs—that is, particular NEOs were observed near Earth multiple times. Overall, about 328,000 of the roughly 338,000 observations in the dataset were from NEOs repeated at least once in the dataset, and many were observed more than once. lso of note is the larger proportion of nonhazardous objects. Of the some 338,000 observations, around 43,000 were classified as hazardous while the remaining 395,000 observations were not.

Visualizing the data, figure 3 shows pairwise scatterplots of the three quantitative variables remaining in the data. There are certainly interactions between variables, but no clear dependencies like with the diameter variables and magnitude shown in Fig 1. Further, inter-variable correlations were low, so each variable seems to carry different information.

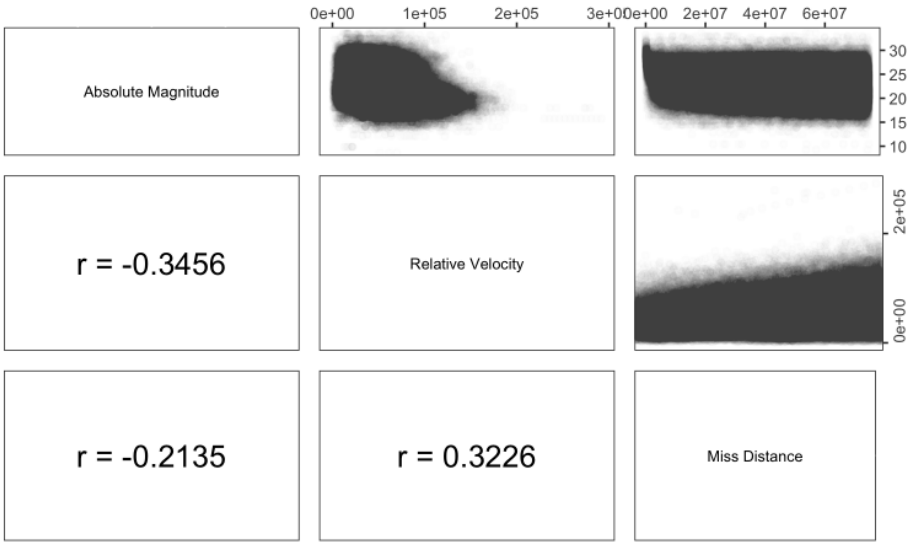

Figure 3: Pairwise scatterplots for three remaining quantitative variables. Correlations between variables are shown mirrored across the diagonal.

There also seems to be a significant relationship between absolute magnitude and the hazardous classification. The histograms in figure 4 show the distribution of the observations' absolute magnitudes based on their classification as hazardous or non-hazardous. There appears to be a clear cut-off where objects with magnitudes above roughly 22km are unlikely to be classified as hazardous. This makes some intuitive sense: as in equation 3.1, larger values of absolute magnitude result in lower sizes, so these are smaller objects. Below this point though, observations may not be as informative due to the presence of similar points in the non-hazardous group.

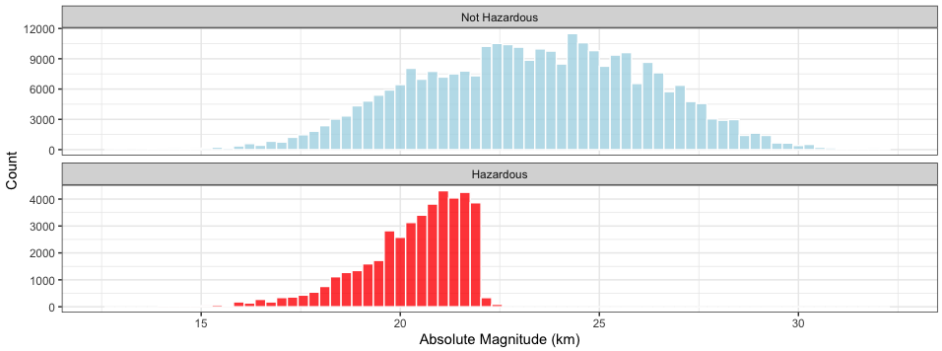

Figure 4: Histogram of bsolute Magnitude observations by hazardous classification

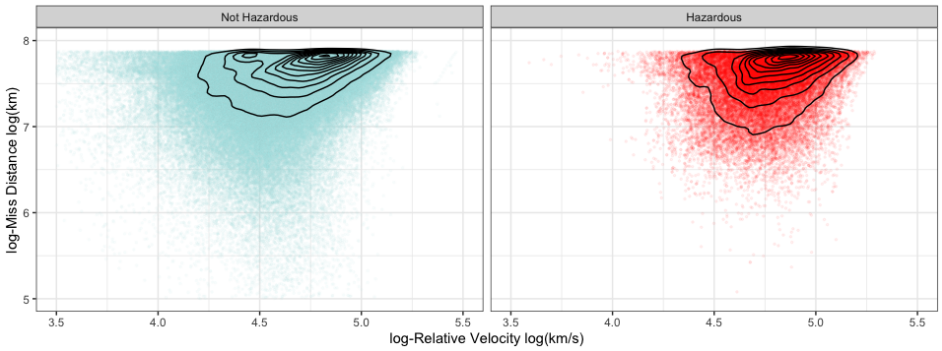

Figure 5: log-miss distance vs log-velocity with hazardous classifier. Black lines show density contours.

Finally, figure 5 shows the spread of hazardous and non-hazardous observations as well as density contours based on their log of relative velocity and log of miss distance. With large disparities in sample size, density contours are able to show the measured distribution better than simple scatter-plots. reas in higher density regions can be thought of as being more likely to contain observations. The log was taken of the two variables to improve the scale of the observations. There appears to be a slight relation between the hazardous classification and these variables. The contours on the hazardous data seem to be slightly more compact in log-relative velocity: whereas non-hazardous data have contours extending further below 4.25 log(km/h), the hazardous data do not. Similarly, the hazardous data seem to extend further down into the log-miss distance. The contour seems to touch 6.75 log(km) for the hazardous, whereas they do not extend past 7 log(km) for nonhazardous. These are slight differences and might typically be safely overlooked, but with tens of thousands of observations for the hazardous data, and hundreds for the nonhazardous data, we are relatively confident that these are real differences.

5. Results & Discussion

5.1. **Methods.**

We made classification trees to predict if objects are hazardous on the basis of the object's absolute magnitude, relative velocity, and distance from the Earth. Generally the log was not used with trees since this is an increasing function and so taking the

log of a variable only serves to move the cutoff point the tree chooses. We built a very complex tree, a pruned tree, and a simple tree. In all trees, the neo id was excluded from our predictors since these trees don't incorporate dependencies. These models were built from the greater part of a 70:30 training-testing split of the full dataset. We used 50-fold cross-validation to regulate overfitting and choose pruning points.

We compared tree models to a simple classifier we called the "stump" tree, which predicted that every NEO was not hazardous (the most common result). This serves as a nice baseline for comparison because this is, in some sense, the worst we might expect our models to do. Small increases in classification rate then indicate an inability to capture the larger structures of the data.

We also built logistic regression models to predict the categorical hazardous response using the three quantitative predictors. We did this because parametric methods generally have more direct inferential power and if the assumptions are satisfied, we often receive many desirable theoretical properties. s discussed earlier, however, there are inter-observation correlations through the neo id. To account for these dependencies, we created logistic regression models with random effects for neo id.

To minimize computation times and to avoid overfitting, we specified only random intercepts for random effects. Furthermore, to help with model convergence we generally specified that we would only consider NEOs for which there were more than some number of observations. s a result, using a traditional training-testing split does not make sense because we would be forced to further split our training set. Realizing this, we chose NEOs from the full dataset. We defined our test set as the complement of the training set—all NEOs for which there were fewer than some number of observations. This means that the random effects of the logistic models were unused since there was no overlap in neo id between the models. This is necessary albeit somewhat unrealistic given that we expect to see NEOs again.

In all cases, we used the classification error on a test set to measure how well the models described the data. For logistic regression, we investigated different classification cutoff points in the training test before computing classification error. We chose cut-off points by minimizing the total classification error rather than the traditional method of minimizing the joint sensitivity and specificity. This is because there are far more NEOs which aren't hazardous and choosing the cut-off point this way would likely result in higher error.

5.2. **Tree Models.**

For trees, as described earlier, we started by making a highly complex foundational tree and then pruned that. We also created a low complexity "sapling" tree, and a "stump"—as described in methods—to compare performance with. Summaries from these models are in Table 1.

We set the foundational tree's starting complexity parameter to an astonishingly low 10^{-6}. This complexity parameter provided minimal penalties for high model complexity, generating a highly complex tree. Through cross-validation, we found that the pruned tree with cp $= 8\,5 \cdot 10$ yielded the lowest error estimate; the tree with this cp is the pruned tree. This pruned tree, contained around 1200 leaves as opposed to the approximate 3600 in the foundational tree. Removing 2/3 of the leaves resulted in a minute increase in accuracy, likely removing bloat or noise from estimations. The sapling, with only 13 leaves, had slightly higher accuracy than

the foundational tree indicating that much of the predictive power of the trees is achieved within the first bunch of leaves. Increasing the leaves, however, leaves room for improvement in the model—there is ample inherent complexity within the data to accommodate many many more leaves, as was demonstrated by the pruned tree having higher accuracy.

	CP	Leaves	Error rate	ccuracy
Foundational Tree	10^{-6}	3632	0.1162	0.8838
Pruned Tree	$8\,5 \cdot 10$	1198	0.1090	0.8910
Sapling	$9 \cdot 10^{-4}$	13	0.1149	0.8851
Stump	1	1	0.1274	0.8726

Table 1: Complexity, represented by the complexity parameter, the number of leaves, and the resulting performance, represented by the classification error rate and the accuracy for each of the three tree models and the stump. The error rate is the classification error, and the accuracy is the complement of the classification error rate.

Figure 6 shows the variable importance for the three nondegenerate tree models. In all three trees, the absolute magnitude had the highest variable importance. The remaining two variables remained somewhat consistently important for all models. The stability in relative variable importance indicates also that increasing the complexity does not change the nature of the trees much.

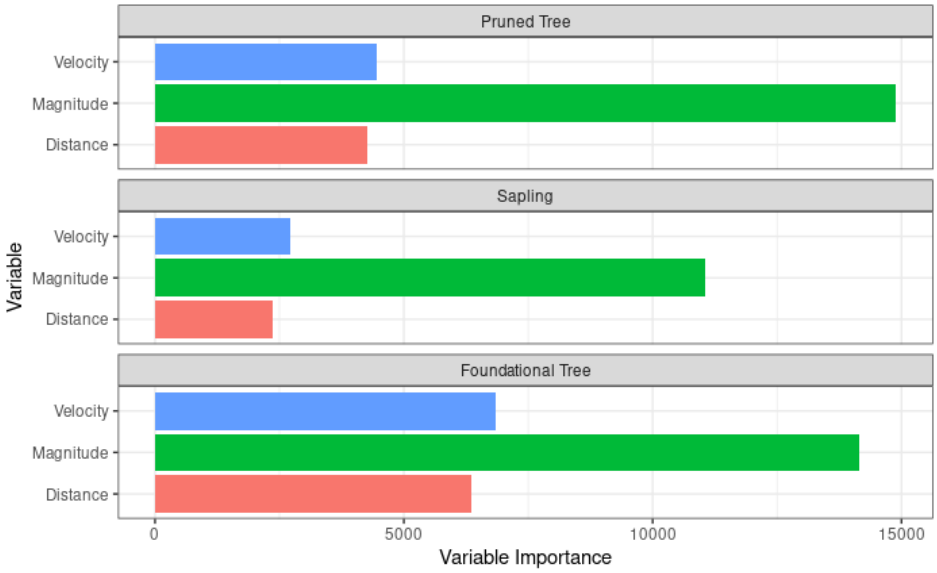

Figure 6: The three tree models' variable importance for bsolute Magnitude (Magnitude), Relative Velocity (Velocity), and Miss Distance (Distance). Variable importance represents the degree to which a variable reduces impurity across splitting. The split impurity was calculated using the Gini index.

The classification error rate and accuracy were calculated to assess the models' performance on new observations. New predictions could be made by sourcing from the testing dataset. The models' predictions were compared to N S 's results (provided by the testing dataset). The classification error rate represents how poorly

each model functioned, while the accuracy highlights how well each model functioned. Due to our goal, we do not explicitly care about classification error rates since we are more interested in the broad trends of how N S classifies these objects. The classification error, however, provides a benchmark for how well the models fit the data. s a holistic, we would trust inferences from a model which perfectly predicts data instead of one which showed essentially random error. Therefore, considering that the error rates are only slightly lower than the stump model we cannot be entirely confident on inferences from this model past statements regarding the significance of bsolute Magnitude. While the high variable importance of the absolute magnitude is likely not a fragment of the models or noise, because of the small increase in performance we hope to verify this trend with a better performing model.

5.3. **Multilevel Logistic Model.**

We fit logistic regression models to predict the hazardous classification of NEOs based on absolute magnitude, relative velocity, miss distance. We fit all of the models with log terms for relative velocity and miss distance; the log transformation helped with model stability by normalizing the scale of the variables. We also fit mixed effects to account for repeated observations of the same NEO by including random intercepts for each neo id.

s discussed in the methods, to aid with model fit for the mixed effects models, we fit all of the models on the entire data trimmed so that there were more than 5 or 10 repeated observations for each NEO. This created a natural test set (albeit smaller than the test set used for trees) consisting of all observations for which the object was observed 5 (or 10) or fewer times.

Summaries of the coefficient estimates for the four models are included in table 2. Based on the random intercepts (RI) model, it seems like absolute magnitude was by far the most important variable, followed by (log-)miss distance, and (log-)relative velocity. For both of the fixed effects (FE) models, there is no indication that any of the variables might be more important than others. This may be because truly all variables are quite important and with such a large sample size, the p-value naturally shrinks to almost zero. lthough it may also be that the FE model is violating assumptions and assumes the data take a particular shape that it does not, and thus arrived at these p-values.

Variable	RI, $n > 5$	FE, $n > 5$	RI, $n > 10$	FE, $n > 10$
Intercept	0.196	$< 2 \cdot 10^{-16}$	0.319	$< 2 \cdot 10^{-16}$
Magnitude	$2\,5 \cdot 10^{-11}$	$< 2 \cdot 10^{-16}$	$1\,1 \cdot 10^{-7}$	$< 2 \cdot 10^{-16}$
Velocity	0.052	$< 2 \cdot 10^{-16}$	0.261	$< 2 \cdot 10^{-16}$
Miss Distance	0.005	$< 2 \cdot 10^{-16}$	0.081	$< 2 \cdot 10^{-16}$

Table 2: p-values of all coefficients in the four models. RI signifies a random intercepts model while FE is for fixed effects. $n > 5$ or $n > 10$ mean that these models were fit with subsets of the data for which there were more than 5 or 10 repeated observations of NEOs. Magnitude is for bsolute Magnitude, Velocity and Miss distance are respectively log-velocity and log-miss distance.

While we had varying levels of significance for different variables with the RI model, the distribution of random intercepts from the model's fit was concerning. One assumption we make when building mixed effects models is that there is some mean zero, normal, "random" effect which is specific to individuals. s shown in figure 7, this assumption is violated. There are two distinct groups within the random effects. Furthermore, neither look fully normal. The group centered above zero seems vaguely normal, but looks less structured. Meanwhile, the group centered just below zero is heavily left skewed. The $n < 10$ model has a similar structure.

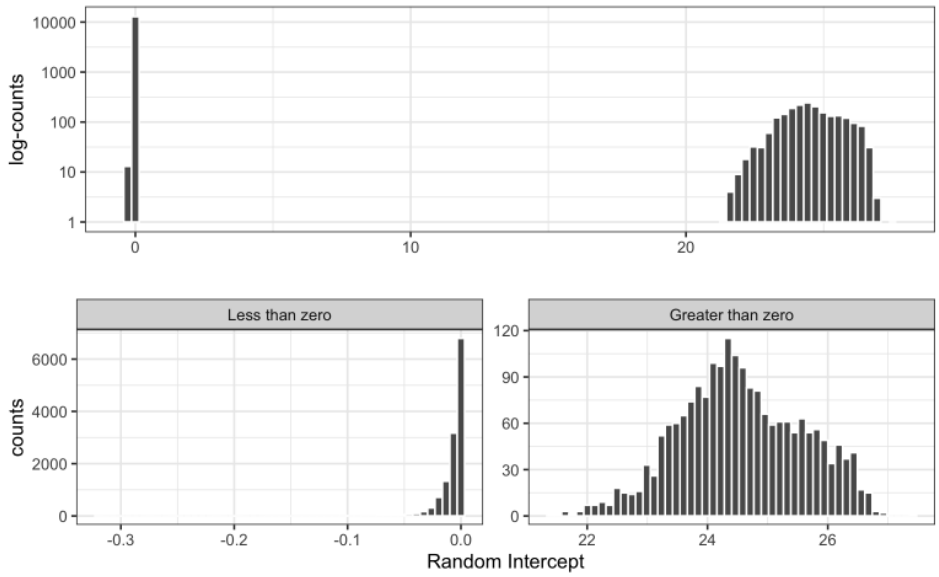

Figure 7: Distribution of RI estimates for the $n < 5$ model. Top plot has log-counts so the random intercepts estimated to be greater than zero are visible. The bottom plot shows individual plots of the RI estimates less than and greater than zero.

The RI models are clearly not perfect, and it seems as if there was likely some other variable we should include as a random effect, or include a random slopes term. However, our data is relatively spare in predictors and repeats of NEOs, and due to model fit time, fitting random slopes is not feasible. We could not explore this route further. Regardless, though, there is a distinct difference between different NEOs indicating that the FE logistic regression model is inappropriate for the data. We will continue our analysis exclusively on the two RI models here.

While not absolutely necessary for inference, considering different cutoff points and test accuracy is a useful metric for measuring how well a model can explain data. For our RI logistic models, we considered different cutoff points, as well as test error. Table 3a summarizes these results, and table 3b shows a confusion matrix constructed with the $n > 5$ model and alternative cutoff point. The confusion matrix for the $n > 10$ model is similar, but has a lower sensitivity. We found the cutoff points by minimizing the total classification error in the training data. New observations in the testing data were treated as independent even if there were other instances of any particular NEO being observed. This is not ideal, but reasonable considering that most observations came from repeated NEOs and the NEOs here were observed at most 5 times, making it difficult to deal with inter-observational correlations. Further, we do not know what

the random effects for new NEOs would look like, so we would have to re-fit the model.

(a)	Optimal Cutoff	Classification Rate
$n < 5$	0.999032	0.9938868
$n < 10$	0.999204	0.9824931

(b)	True	False
True	35869	238
False	0	2825

Table 3: (a) Optimal cutoff (computed as the cutoff which minimizes classification error) and the corresponding classification rate when tested to the test data. (b) Confusion matrix for the $n < 5$ model. The rows indicate the result of the prediction while the columns indicate the truly observed value.

These logistic models have exceptionally high classification rates. It seems like these are certainly capturing the trends of how N S classifies objects as hazardous. Because of this, we may be able to draw some conclusions about the optimal cutoff value we found: if this is indeed similar to how N S models the data, it seems like they are hesitant to classify objects as potentially hazardous. With a cutoff of around 0.999, we generally need overwhelming evidence from the model to predict that an NEO is potentially hazardous. This may also be true for N S indicating that they require overwhelming evidence to have a cause for alarm. This makes sense for two reasons. First, dynamics of space are generally well understood and so in the absence of unforeseen factors (ie perhaps two objects colliding), we can be relatively confident about measurements and predictions of objects' trajectories. Second, many of these objects would be monitored regardless of their status as potentially hazardous, so classifying them as potentially hazardous may be an overreaction.

6. Conclusions

Working with the Near Earth Object data provided by the N S database, we have constructed tree and mixed effect logistic models. Due to the few predictors in the data, these models were generally fit with absolute magnitude, (log-)miss distance and (log-)relative velocity as primary predictors.

Tree models indicated that the absolute magnitude variable seemed to be the most significant. However, they were somewhat uninformative past this. Furthermore, these models achieved marginally higher classification accuracy compared to a simple algorithm classifying each NEO as non-hazardous no matter what. This indicates that they were able to explain the structure of the data somewhat better, but ultimately are unable to fully explain the data, and should thus be relied on in conjunction with more reliable conclusions about the data.

Multilevel Logistic Regression provided a promising alternative: they were able to account for dependencies among NEOs. They can also allow for more inferential properties since they are parametric and thus more interpretable. These models, though, also had their issues. Non-normal distributions of random effect estimates indicated that we were somewhat violating assumptions. Due to the lack of predictors, we were not able to rectify this. However, their ability to add random effects for NEOs is desirable. Despite having odd distributions of random effects, when fitted to new data using only the fixed effect estimates, the models performed exceptionally well, indicating that they were able to successfully capture many features of the

classification algorithm that N S used. From these models, we can safely conclude that absolute magnitude is by far the most important predictor of the three.

It is worth addressing the high classification rate of the logistic regression models. In typical data, we would be concerned by having such high classification rates. Generally, we expect a nontrivial amount of irreducible error, and so having such high success rates might indicate pure luck among the training data or technical errors within the creation of the models or validation on the test data. These are possible, but it is also important to consider the nature of the data: these are curated by N S , and N S itself declares whether or not they deem an object hazardous. Rather than through a stochastic process, these NEOs are likely tagged as hazardous through a largely deterministic process. Having a high classification rate then means that our model is similar to the classifier which N S used. N S likely also has many other measurements which they decided to exclude from these data for one reason or another (ie, continuous monitoring of objects vs a single point). Because of this, we do not quite expect to have perfect accuracy, although the high classification rate indicates that we have likely have built a similar model to what N S uses.

In closing, we have built several models to try to predict the hazardous classification of NEOs. We did this primarily with tree based methods and logistic regression. The tree based methods yielded better-than-average results when compared to a simple classifier when choosing the most likely result. However, the trees did not seem to capture much of the data. Logistic regression, however—particularly with random effects—seemed to capture much of the structure of the data and had high accuracy when predicting classification. Furthermore, in both of these models, absolute magnitude seemed to be the most significant variable. These models could likely be improved by more complete data as discussed earlier, but generally did well as is.

7. References

1. dmin. (2020, November 23). Record breaking close approach of asteroid 2020 VT4. Near-Earth Objects Coordination Center. https://neo.ssa.esa.int/-/record-breaking-close-approach-of-asteroid-2020-vt4

2. Farook, O. (2022, ugust 23). Will the asteroid 2018 LF16 Slam into Earth in 2023?. Black Dot Research. https://blackdotresearch.sg/asteroid-2018lf16/

3. Ivan Sher Bakov. (2024a). N S | Nearest Earth Objects (1910-2024) [Dataset]. https://www.kaggle.com/datasets/ivansher/nasa-nearest-Earth-objects-1910-2024/data

4. N S PI. (n.d.). [Dataset]. https://api.nasa.gov/

5. Ivan Sher Bakov. (2024b). Machine-Learning-Basic-Project [Computer software]. https://github.com/ivansherbakov9/Machine-Learning-Basic-Project/blob/main/creating_dataset%20(1).ipynb

6. Jet Propulsion Laboratory. (1910). NEO Earth Close pproaches [Dataset]. https://cneos.jpl.nasa.gov/ca/

7. Jet Propulsion Laboratory. (n.d.). steroid Size Estimator. https://cneos.jpl.nasa.gov/tools/ast_size_est.html

8. PPENDIX: R CODE

8.1. Data Cleaning & Exploratory Data nalysis.

```
knitr::opts_chunk$set(eval = F, echo = T)
set.seed(243)
library(ggplot2);library(dplyr);library(tidyverse);library(moderndive)
library(magrittr); library(gridExtra)
library(rpart); library(rpart.plot); library(yardstick)
library(glmnet); library(lme4); library(caret)
```

```
NEOs <- read.csv("~/Documents/Math243/Math243_Final_project/
                 Nearest_Earth_Objects.csv") %>%
  mutate(is_hazardous = as.factor(is_hazardous))
```

```
missing = NEOs[!(NEOs$neo_id %in% (NEOs %>% drop_na())$neo_id), ]
head(missing, n = 5) %>% dplyr::select(-neo_id)
missing %>% group_by(neo_id) %>% summarize(number = n())
```

```
pairwise = fun tion(var1, var2, alpha) {
  if(missing(alpha)) {alpha = 0.05} #define  n  lph  if it's missing
  data.frame(x = var1,
             y = var2) %>% #d t fr me with two vectors
    ggplot(aes(x = x, y = y)) + # nd empty plot with these d t
    geom_point(alpha = alpha, color = "black") + #sc tter plot
    theme_bw() + # nd the rest is m nipul ting the themes of the plot
    labs(x = "", y = "") +
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank())
}
```

```
names = fun tion(name, pos) {
  #This is code to m ke plots with n mes in the center. Much of the
  #Code is styling the plot so I will not expl in much of it.
  plot = ggplot() +
    geom_text(data = data.frame(x = paste0(name)),
              aes(x = 0.5, y = 0.5, label = x), size = 3) +
    labs(x = "x", y = "y") +
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.background = element_rect(fill = 'white'),
          panel.border = element_rect(fill = N , colour = "grey20")) +
    geom_point(data = data.frame(x = c(0.5), y = c(0.5)),
            aes(x,y), alpha = 0) +
    scale_x_continuous(position = "top") +
    ylim(0,1) +
```

```
    xlim(0,1) #this sets up  templ te th t  ny position c n use
  if(pos == 1.1) { #depending on position we w nt to style differently
    plot = plot +
      scale_y_discrete() + scale_x_continuous(position = "top") +
      theme(axis.text.x = element_text(color = "white"),
        axis.ticks.x = element_line(color = "white"),
        axis.title.x = element_text(color = "white"),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.title.y = element_blank())
  }
  if(pos == 2.2) {
    plot = plot +
      scale_x_discrete() + scale_y_discrete() +
      theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.title.y = element_blank())
  }
  if(pos == 3.3) {
    plot = plot +
      scale_y_continuous(position = "right", breaks = c(0.5)) +
      theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
        axis.text.y = element_text(color = "white"),
        axis.ticks.y = element_line(color = "white"),
        axis.title.y = element_text(color = "white"))
  }
  return(plot)
}
```

```
#From here, c ll the function with  ll v ri bles. Of note is the
#code included to style e ch plot so it fits in will with grid  rr nge.
#There is no other subst nti l code.
plt1.2 = NEOs %$%
  pairwise(var1 = estimated_diameter_min, var2 = absolute_magnitude) +
  scale_x_continuous(position = "top")  +
  scale_y_discrete() +
  theme(axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.title.y = element_blank())# removes the y  xis m rks

plt2.3 = NEOs %$%
```

```
  pairwise(estimated_diameter_max, estimated_diameter_min) +
  scale_x_discrete() + #removes x  xis m rks
  scale_y_continuous(position = "right") +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank())

plt1.3 = NEOs %$%
  pairwise(estimated_diameter_max, absolute_magnitude) +
  scale_x_continuous(position = "top") +
  scale_y_continuous(position = "right")

plt1.1 = names(" bsolute Magnitude", pos = 1.1)
plt2.2 = names("Estimated Minimum Diameter", 2.2)
plt3.3 = names("Estimated Maximum Diameter", 3.3)

emptyplot = ggplot() +
  theme_void() #Empty sp ce for grid  rr nge
```

```
grid.arrange(plt1.1, plt1.2, plt1.3, # rr nge  ll our plots
             emptyplot, plt2.2, plt2.3,
             emptyplot, emptyplot, plt3.3,
             ncol = 3)
```

```
NEOs = NEOs %>% drop_na()
NEOs = NEOs %>% dplyr::select(-c("name", "orbiting_body",
                                 "estimated_diameter_min",
                                 "estimated_diameter_max"))
```

```
nrepeats = NEOs %>% group_by(neo_id) %>%
  summarize(n = n()) %>%
  arrange(desc(n))
nrepeats %$% mean(n)
repeats = nrepeats %>% filter(n > 1)
repeats %>% head(n = 5)
repeats %$% sum(n)
repeats %>% filter(n > 20) %$% sum(n)
```

```
corplot = fun tion(var1, var2, pos) {
  corr = cor(var1, var2)
  plot = ggplot() +
    geom_text(data = data.frame(x = paste0("r = ", round(corr,4))),
              aes(x = 0.5, y = 0.5, label = x), size = 6) +
    labs(x = "x", y = "y") +
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
```

```
          panel.background = element_rect(fill = 'white'),
          panel.border = element_rect(fill = N , colour = "grey20")) +
    geom_point(data = data.frame(x = c(0.5), y = c(0.5)),
               aes(x,y), alpha = 0) +
    scale_x_continuous(position = "top") +
    ylim(0,1) +
    xlim(0,1)

  if(pos == 2.1) {
    plot = plot +
      scale_x_discrete() + scale_y_discrete() +
      theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.title.y = element_blank())
  }
  if(pos == 3.1) {
    plot = plot +
      scale_y_discrete() +
      theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.title.y = element_blank())
  }
  if(pos == 3.2) {
    plot = plot +
      scale_y_discrete() +
      theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.title.y = element_blank())
  }
  return(plot)
}
```

```
f2plt1.2 = NEOs %$%
  pairwise(var1 = relative_velocity, var2 = absolute_magnitude,
           alpha = 0.0075) +
  scale_x_continuous(position = "top")  +
  scale_y_discrete() +
```

```
  theme(axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.title.y = element_blank())# removes the y  xis m rks
f2plt2.1 = NEOs %$%
  corplot(var1 = relative_velocity, var2 = absolute_magnitude, 2.1)

f2plt1.3 = NEOs %$%
  pairwise(var1 = miss_distance, var2 = absolute_magnitude,
           alpha = 0.0075) +
  scale_x_continuous(position = "top") +
  scale_y_continuous(position = "right")
f2plt3.1 = NEOs %$%
  corplot(var1 = miss_distance, var2 = absolute_magnitude, 3.1)

f2plt2.3 = NEOs %$%
  pairwise(var1 = miss_distance, var2 = relative_velocity,
           alpha = 0.0075) +
  scale_x_discrete() + #removes x  xis m rks
    scale_y_continuous(position = "right",
                       breaks = c(0*10^5, 2 * 10^5)) +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
        axis.text.y = element_text(angle = 90, vjust = 0.5, hjust=1))
f2plt3.2 = NEOs %$%
  corplot(var1 = miss_distance, var2 = relative_velocity, 3.2)


f2plt1.1 = names(" bsolute Magnitude", pos = 1.1)
f2plt2.2 = names("Relative Velocity", pos = 2.2)
f2plt3.3 = names("Miss Distance", pos = 3.3)
```

```
grid.arrange(f2plt1.1, f2plt1.2, f2plt1.3,
             f2plt2.1, f2plt2.2, f2plt2.3,
             f2plt3.1, f2plt3.2, f2plt3.3,
             ncol = 3)
```

```
NEOs %>%
  ggplot(aes(x = absolute_magnitude)) +
  geom_histogram(bins = 100, color = "white",
                 aes(fill = is_hazardous)) +
  facet_wrap(~is_hazardous, ncol = 1, scale = "free_y",
             labeller = as_labeller(c("False" = "Not Hazardous",
                                      "True" = "Hazardous"))) +
  labs(y = "", x = " bsolute magnitude (km)",
       title = "") +
```

```
  scale_fill_manual(values = c("True" = "red", "False" = "lightblue")) +
  theme_bw() + theme(legend.position = "none") +
  xlim(12.5, 32.5)
```

```
NEOs %>%
  ggplot(aes(x = log10(relative_velocity),
             y = log10(miss_distance),
             color = is_hazardous)) +
  geom_point(alpha = 0.01, size = 0.5) +
  labs(x = "log-relative velocity log(km/h)",
       y = "log-miss distance log(km)",
       title = "",
       color = "Is Hazardous") +
  scale_color_manual(values = c("False" = "lightblue",
                                "True" = "red")) +
  theme_bw() +
  guides(color = guide_legend(override.aes = list(alpha=1,size=2))) +
  xlim(3.5,5.5) + ylim(5,8) +
  facet_wrap(~is_hazardous, ncol = 2,
             labeller = as_labeller(c("False" = "Not Hazardous",
                                      "True" = "Hazardous"))) +
  theme(legend.position = "none") +
  geom_density2d(color = "black")
```

8.2. **Tree models.**

```
set.seed(243)
which.train <- sample(1:nrow(NEOs),
                      size = 0.7 * nrow(NEOs))
train <- NEOs[which.train, ]
test <- NEOs[-which.train, ]
rm(which.train)
```

```
rpart(is_hazardous~., data = test,
       ttributeTokcontrol = rpart.control(maxdeth = 0)) -> stump

stump %>% rpart.plot()

stump %<>% prune(cp = 1)

stump %>% rpart.plot()
```

```
set.seed(1)
tree <- rpart(is_hazardous ~ . -neo_id, data = train,
```

```
                control = rpart.control(xval = 50, cp = 0.000001))
rpart.plot(tree)
```

```
plotcp(tree)

sapling <- rpart(is_hazardous ~ . -neo_id, data = train, cp = 0.0009)
rpart.plot(sapling)


best_cp <- tree$cptable[which.min(tree$cptable[, "xerror"]), "CP"]
pruned <- prune(tree, cp = best_cp)
rpart.plot(pruned)

sum(sapling$frame$var == "<leaf>")
sum(pruned$frame$var == "<leaf>") #number of le ves
sum(tree$frame$var == "<leaf>") #number of le ves
```

```
tree$variable.importance %>% as.data.frame(x=)
pruned$variable.importance %>% as.data.frame(x=)
sapling$variable.importance %>% as.data.frame(x=)

importance <- data.frame(
    model = c("Tree", "Tree", "Tree", "Pruned", "Pruned", "Pruned",
              "Sapling", "Sapling", "Sapling"),
    parameter = c("Magnitude", "Velocity", "Distance",
                  "Magnitude", "Velocity", "Distance",
                  "Magnitude", "Velocity", "Distance"),
    importance = c(14142, 6855, 6347, 14891, 4452, 4267,
                   11049, 2711, 2359)
)

ggplot(importance, aes(x=parameter, y=importance, fill = parameter))+
  geom_col()+
  facet_wrap(~model, ncol = 1,
             labeller = as_labeller(c("Pruned" = "Pruned Tree",
                                      "Tree" = "Foundational Tree",
                                      "Sapling" = "Sapling"))) +
  coord_flip() +
  theme_bw() +
  theme(legend.position = "none") +
  labs(x = "Variable", y = "Variable Importance")
```

```
tree_preds <- predict(tree, test, type = "class")

tree_error <- mean(tree_preds!=test$is_hazardous)
```

```
pruned_preds <- predict(pruned, test, type = "class")

pruned_error <- mean(pruned_preds!=test$is_hazardous)

sapling_preds <- predict(sapling, test, type = "class")

sapling_error <- mean(sapling_preds!=test$is_hazardous)

data.frame(
    model = c("Tree", "Pruned", "sapling"),
    cp = c(0.000001, best_cp, 0.0009),
    leaves = c(3632, 1198, 13),
    error_rate = round(c(tree_error, pruned_error, sapling_error), 4),
    accuracy = round(c((1-tree_error),
                       (1-pruned_error),
                       (1-sapling_error)), 4)
)
```

8.3. **Logistic Regression.**

```
NEOs <- NEOs %>%
  mutate(
    log_velocity = log10(relative_velocity),
    log_miss = log10(miss_distance)
  )

NEOs_filtered_ng5 <- NEOs %>%
  group_by(neo_id) %>%
  filter(n() > 5) %>%
  ungroup() %>%
  mutate(is_hazardous = case_when(is_hazardous == "True" ~ T,
                                  .default = F))

NEOs_filtered_ng10 <- NEOs %>%
  group_by(neo_id) %>%
  filter(n() > 10) %>%
  ungroup() %>%
  mutate(is_hazardous = case_when(is_hazardous == "True" ~ T,
                                  .default = F))
```

```
log_glmm_ng5 <- glmer(
  is_hazardous ~ absolute_magnitude+log_velocity+log_miss+(1|neo_id),
  data = NEOs_filtered_ng5,
  family = binomial)

log_glmm_ng10 <- glmer(
```

```
  is_hazardous ~ absolute_magnitude+log_velocity+log_miss+(1|neo_id),
  data = NEOs_filtered_ng10,
  family = binomial)

log_glm_ng5 <- glm(
  is_hazardous ~ absolute_magnitude + log_velocity + log_miss,
  data = NEOs_filtered_ng5,
  family = binomial)

log_glm_ng10 = glm(
  is_hazardous ~ absolute_magnitude + log_velocity + log_miss,
  data = NEOs_filtered_ng10,
  family = binomial)
```

```
summary(log_glmm_ng5)

summary(log_glmm_ng10)

summary(log_glm_ng5)

summary(log_glm_ng10)
```

```
plotranef = fun tion(glmm) {
  total_ranef = glmm %>%
    ranef() %$%
    neo_id %>%
    ggplot(aes(`(Intercept)`)) +
    geom_histogram(color = "white", bins = 100) +
    theme_bw() +
    labs(x = "", y = "log-counts") +
    scale_y_log10()

  prep_fig6 = glmm %>%
    ranef() %$%
    neo_id %>%
    mutate(location = case_when(
        `(Intercept)` > 0 ~ "Greater than zero",
        .default = "Less than zero"
    ))#to segment d t

  split_ranef = prep_fig6 %>%
    ggplot(aes(`(Intercept)`)) +
    geom_histogram(color = "white", bins = 50) +
    theme_bw() +
    labs(x = "Random Intercept", y = "counts") +
    facet_wrap(~factor(location, levels = c("Less than zero",
```

```
                                          "Greater than zero")),
                        scales = "free")

  final_plot = grid.arrange(total_ranef, split_ranef, ncol = 1)
  return(final_plot)
}

plotranef(glmm = log_glmm_ng5)
plotranef(glmm = log_glmm_ng10)
```

```
roc = fun tion(glmm, train_data) {
  # we w nt to first determine optim l cutoff for tr ining d t
  #  nd then check how this performs on the test d t

  fitted_logodds = predict(log_glmm_ng5, train_data)
  fitted_probs = exp(fitted_logodds)/(1+exp(fitted_logodds))

  results = data.frame(truth = as.factor(train_data$is_hazardous),
                       probs = fitted_probs) %>%
    mutate(preds = ifelse(probs >= 0.5, T,F), #st nd rd cutoff
           preds = as.factor(preds))

  roc_data = roc_curve(data = results, truth = truth,
                       probs, event_level = "second") %>%
    # Get number of correct f lses  nd combine for error
    mutate(ntruth = specificity * (results %>%
                                     filter(truth == "F LSE") %>%
                                     nrow()),
           nfalse = sensitivity * (results %>%
                                     filter(truth == "TRUE") %>%
                                     nrow()),
           propcorrect = (ntruth + nfalse)/(nrow(results))) %>%
    arrange(-propcorrect) # rr nge to get cutoff

  roc_data %>%
    .[[1,1]] -> c #s ve the cutoff v lue  s c

  test = NEOs %>%  # M ke testing set
    .[not(rownames(.) %in% rownames(train_data)),]


  newpreds = predict(glmm, test, type = "response",
                     allow.new.levels = T)
  new_results = data.frame(truth = test$is_hazardous,
                           probs = newpreds) %>%
    mutate(preds = factor(ifelse(probs >= c, "True", "False")))
```

```
  conf_mat = base::table(Preds = new_results$preds,
                         Truth = new_results$truth)
  test_accuracy = (conf_mat[1,1] + conf_mat[2,2])/(nrow(test))
  return(list(cutoff = c, test_accuracy = test_accuracy,
              matrix = conf_mat))
}
roc(log_glmm_ng5, NEOs_filtered_ng5)
roc(log_glmm_ng10, NEOs_filtered_ng10)
```