Emily Hillenbrand
Metis Project 03 - Classification

Project 03 Summary

31 October 2019

Project Design

I started with this dataset. My initial goal came directly from the "Inspiration" section of the dataset's description: try to predict the cause of a wildfire given its size, location, and date. I wanted to see if I could directly predict the actual cause of a fire out of the 13 available options. However, after getting an accuracy score of 0.34 from my MVP and deciding I wanted to use ROC AUC (which requires a binary classifier) instead of accuracy as my comparison metric, I decided to change my approach. I wanted to see if there was an easy way to rule out any of the causes.

The next thing I tried was further aggregating the fires by their cause. I dropped the rows where the cause was "missing/undefined", and then the rest of them went into one of four categories: Natural, Human/Recreational, Human/Industrial, and Miscellaneous. I was not able to find information on what makes a fire's cause "miscellaneous", so my next step was to see if a fire's cause *could* be predicted; that is, if I could predict whether or not the cause is Miscellaneous. I tried separating the data into "miscellaneous or not", and then tried some test models on it with some rudimentary tuning.

After that, I ran some multi-class "One Vs All" comparisons to see how each category behaves compared to the others, and found that "Natural vs Unnatural" was a clear winner in all cases. The separation is notable in this pairplot.

The next and final step was to re-run the test models and fine-tune the parameters. The final model is shown here with an AUC of 0.95; because this score is so high, I would be very comfortable with using this model to rule out cause=natural.

Even though I didn't end up where I expected, I was able to find a useful application for this analysis.

Tools

1. Python
    a. Data Storage: pickle, pandas

    b. Data Analysis: sklearn, jupyter notebook
    c. Visualization: matplotlib, seaborn
2. SQL
    a. Storage: sqlite database
    b. Queries: sqlite3
3. Google Slides, Google Docs

Data

The data for this project came from Kaggle as a SQLite database.

I used SQL queries through the terminal for initial EDA. Because 1.88 million rows is too unwieldy for the tools at my disposal, I decided to limit my analysis to the western U.S., i.e. Alaska, Washington, Oregon, and California. This brought the length down to about 300,000 rows. I was then able to import that subset to a jupyter notebook and proceed with analysis.

I dropped the rows labeled “missing/undefined” because a “missing” fire cause could easily have belonged to any of the other causes. However, I couldn’t drop “miscellaneous”, because I couldn’t say for sure what that meant and I would have risked providing an incorrect answer.

Most of the data cleaning was already done by the creator of the dataset, so I was essentially able to use it as-is.

Next Time

Break Up With Jupyter Notebooks

Jupyter notebooks are an excellent tool for learning, but they are slow and memory-intensive. Many of the more time-consuming steps of this project could have been done much quicker through an IDE and terminal commands.

Break It Down

In the future, I would like to know if it’s possible to use so few features to predict the cause of a wildfire, whether by a similar “one vs all” comparison or by another approach entirely.

Appendix

Fig 1. MVP scores

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
print("The score for kNN is")
print("{:6.2f}%".format(100*knn.score(X_train, y_train)))
# kNN score is slightly higher than for logreg.
```

executed in 1.74s, finished 17:26:32 2019-10-21

```
The score for kNN is
 34.16%
```

```
acc_train = logreg.score(X_train, y_train) # accuracy (score) of train data
print("The score for Logistic Regression is")
print("%.4f" % (acc_train))
```

executed in 20ms, finished 17:26:28 2019-10-21

```
The score for Logistic Regression is
0.3406
```

Fig. 2 Four Categories Pair Plot

0 = Human/Recreational; 1 = Human/Industrial; 2 = Natural; 3 = Miscellaneous

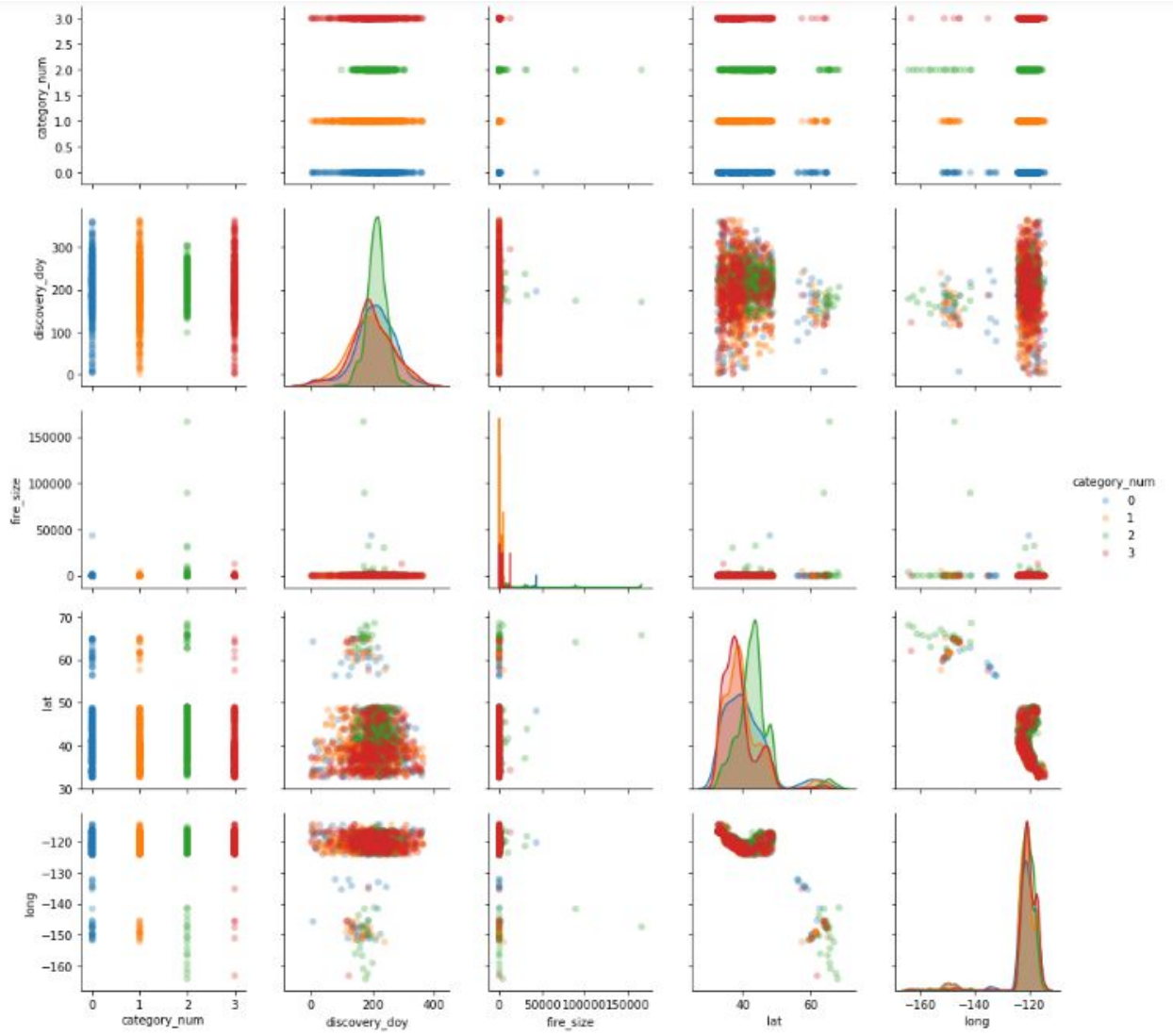


Fig. 3 “Miscellaneous or Not” Pair Plot

Miscellaneous = 0, Not Miscellaneous i.e. All Others = 1

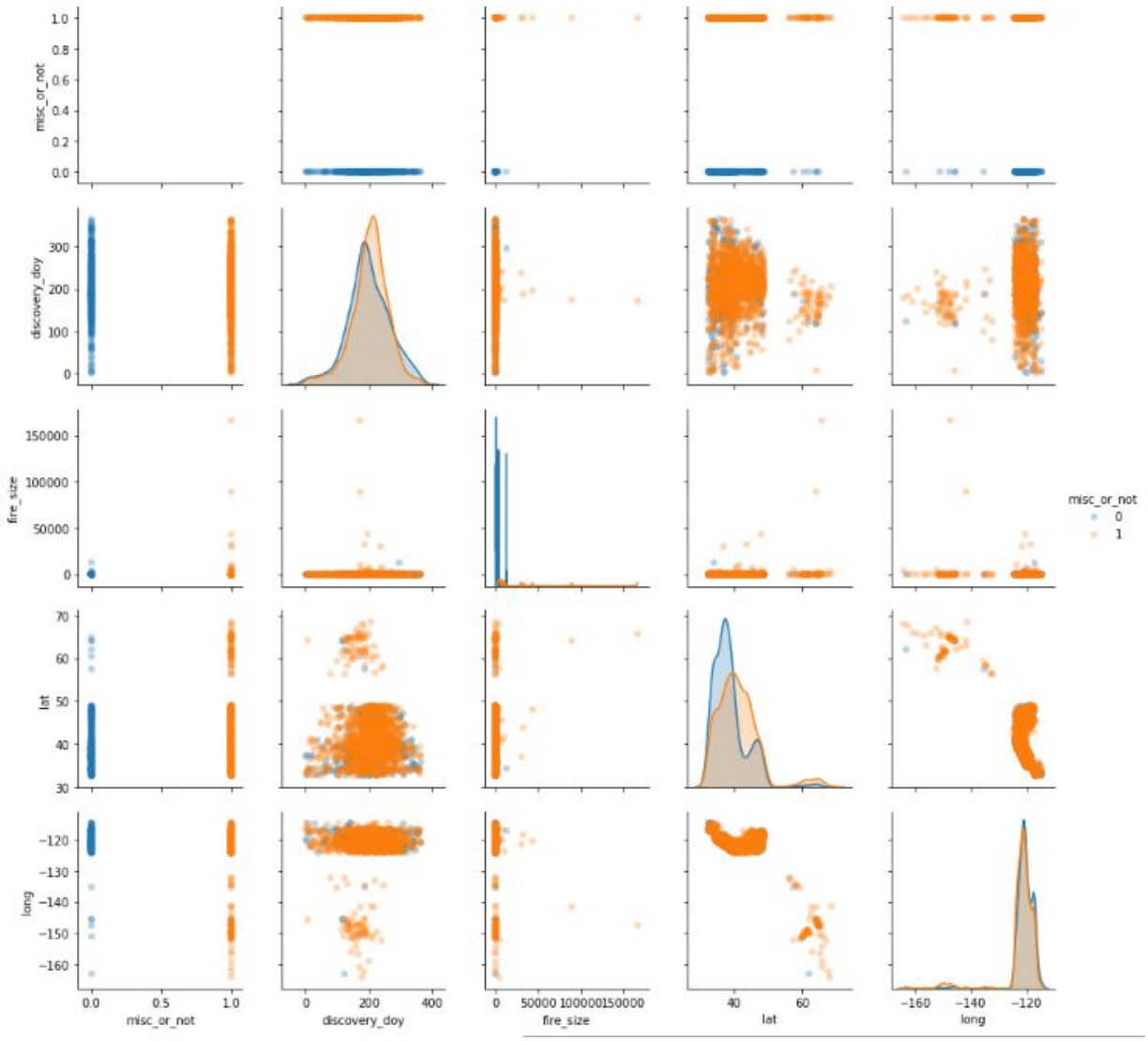
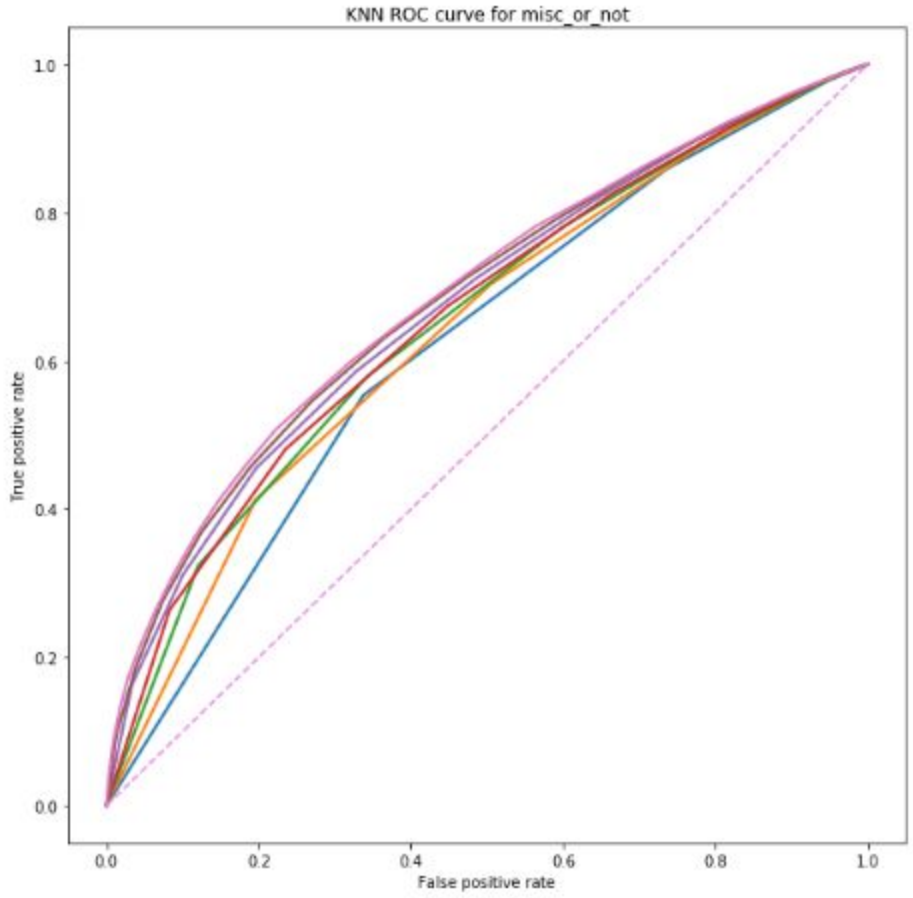


Fig. 4 “Miscellaneous or Not” Models

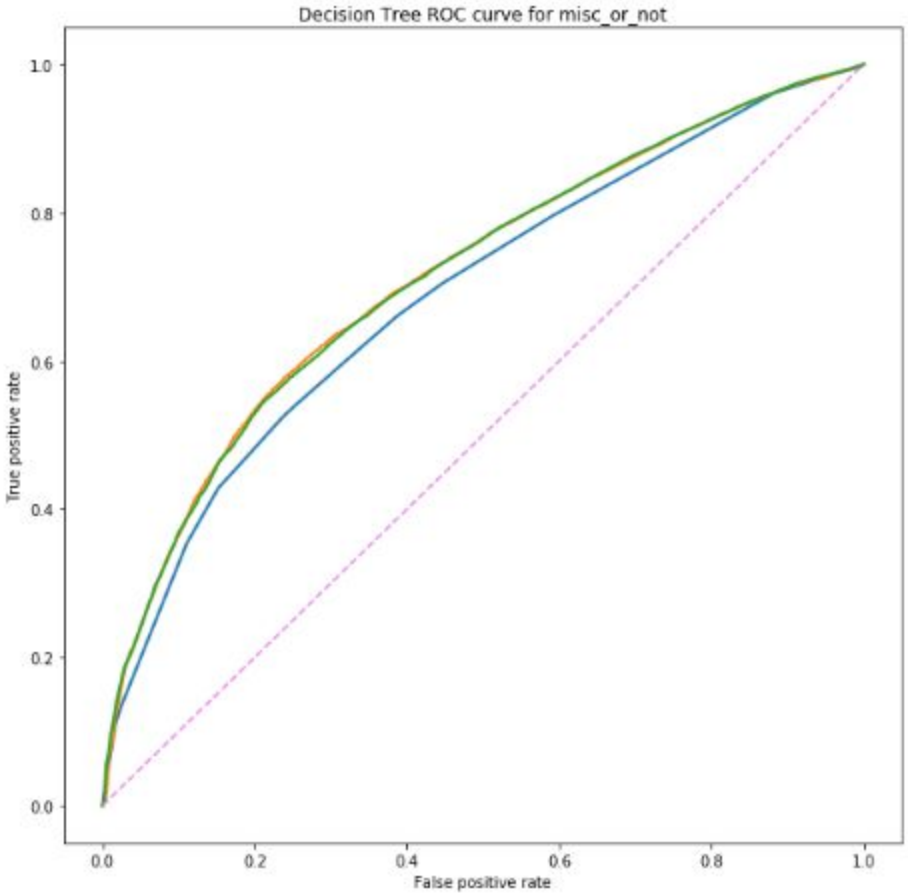
This section contains three figures.

```
ROC AUC score for 3 neighbors =  0.6211005510674262
ROC AUC score for 5 neighbors =  0.640454238179655
ROC AUC score for 7 neighbors =  0.6542530343652012
ROC AUC score for 9 neighbors =  0.6612494847441069
ROC AUC score for 15 neighbors =  0.6740279829985741
ROC AUC score for 25 neighbors =  0.6831874882378449
ROC AUC score for 50 neighbors =  0.6887801757920451
```

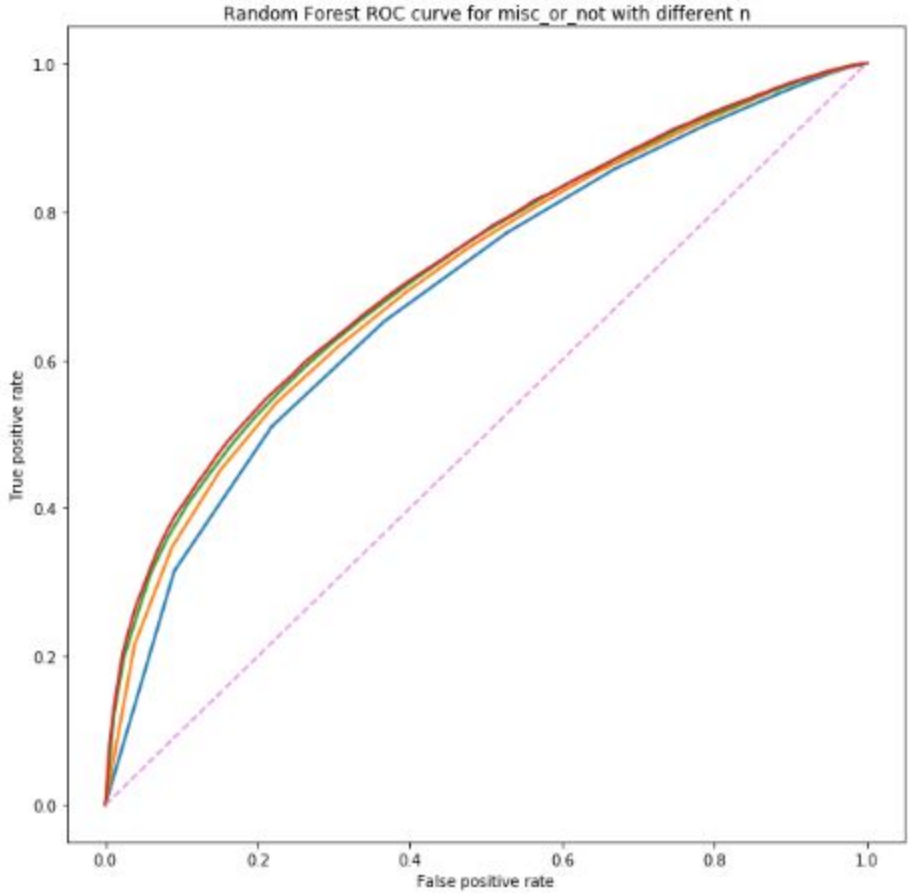


KNN diminishing returns at n_neighbors > 25

```
ROC AUC score for max_depth=4:  0.68879692310324
ROC AUC score for max_depth=10:  0.71449457311153
ROC AUC score for max_depth=9:  0.7138353947929971
```



```
ROC AUC score for n=10:  0.692498326816301
ROC AUC score for n=20:  0.711570137197687
ROC AUC score for n=50:  0.7221193747286915
ROC AUC score for n=100:  0.7262343318740032
```



Random Forest diminishing returns at n_estimators > 100

Fig. 5 Example Multi-Class ROC Comparison

Model = Decision Tree, max depth = 9

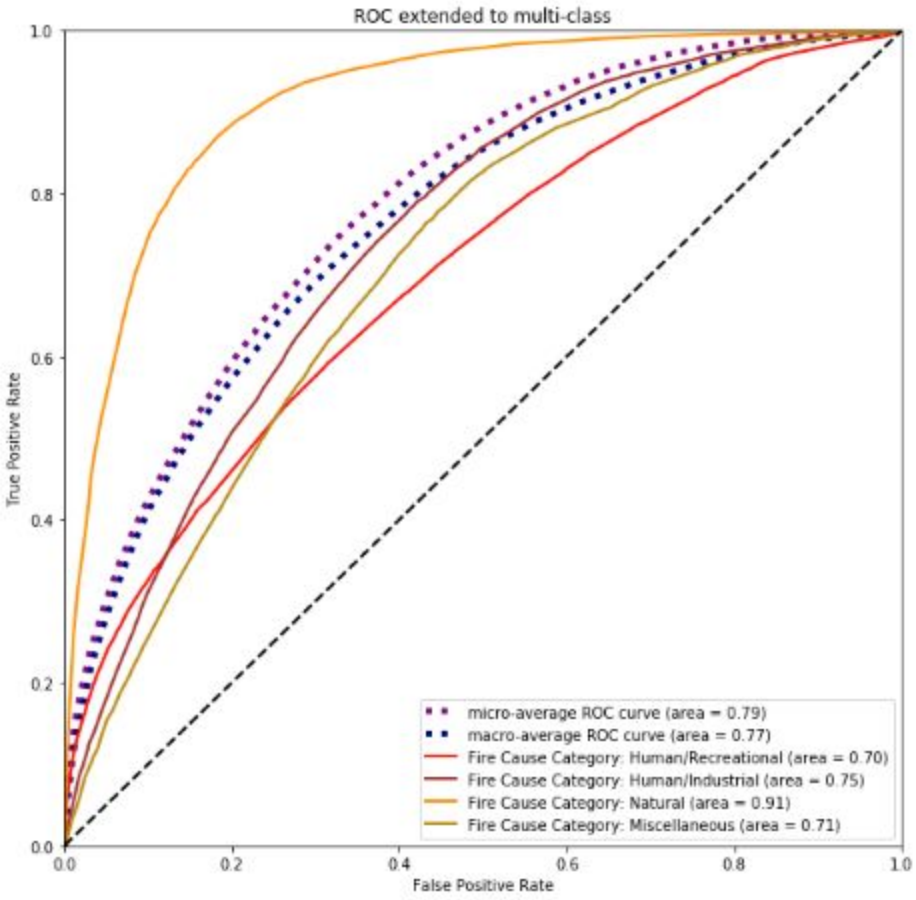


Fig. 6 Natural Vs. Unnatural Pairplot

Notice separation at fire_size:

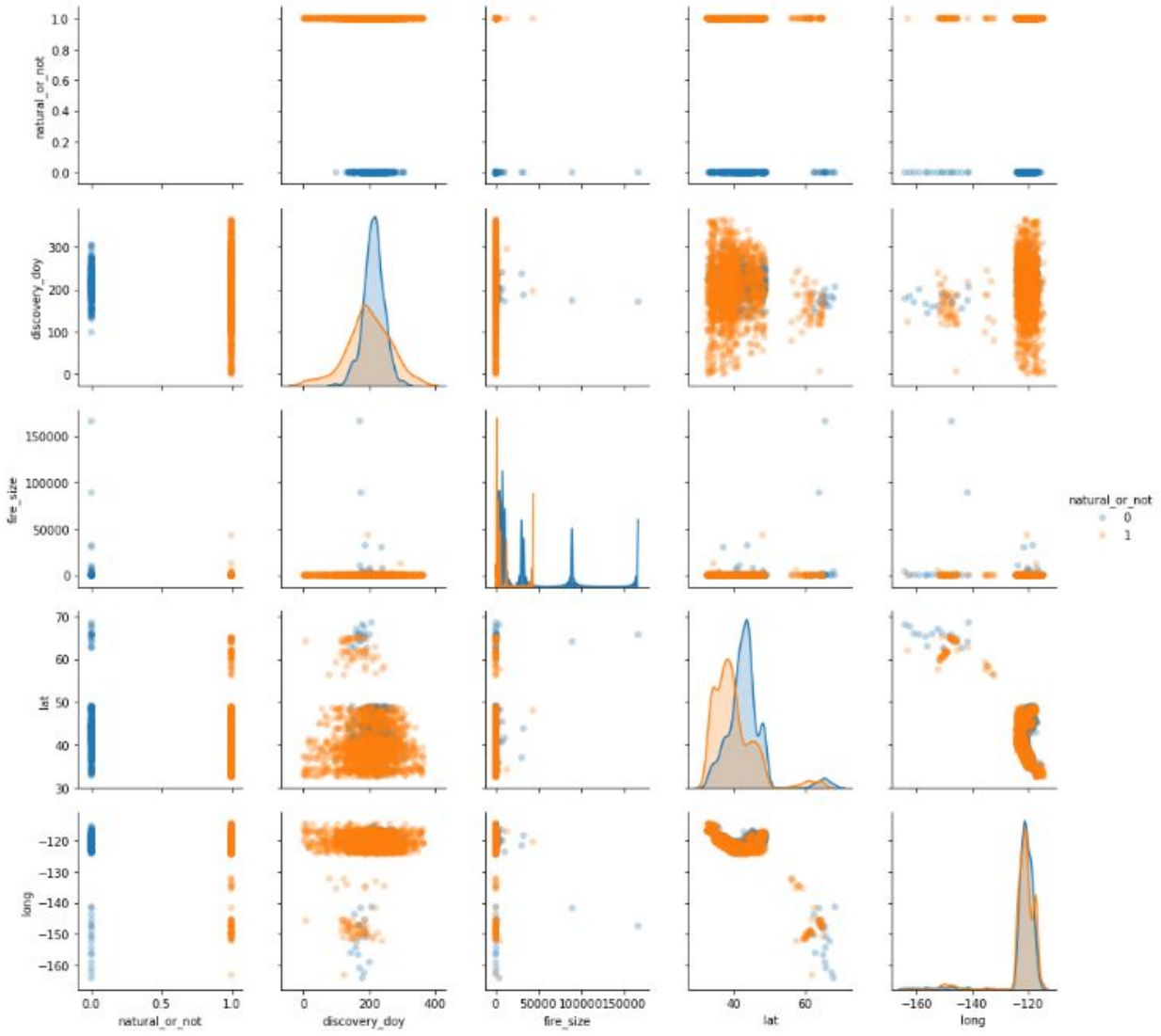


Fig. 7 Final Model: Random Forest

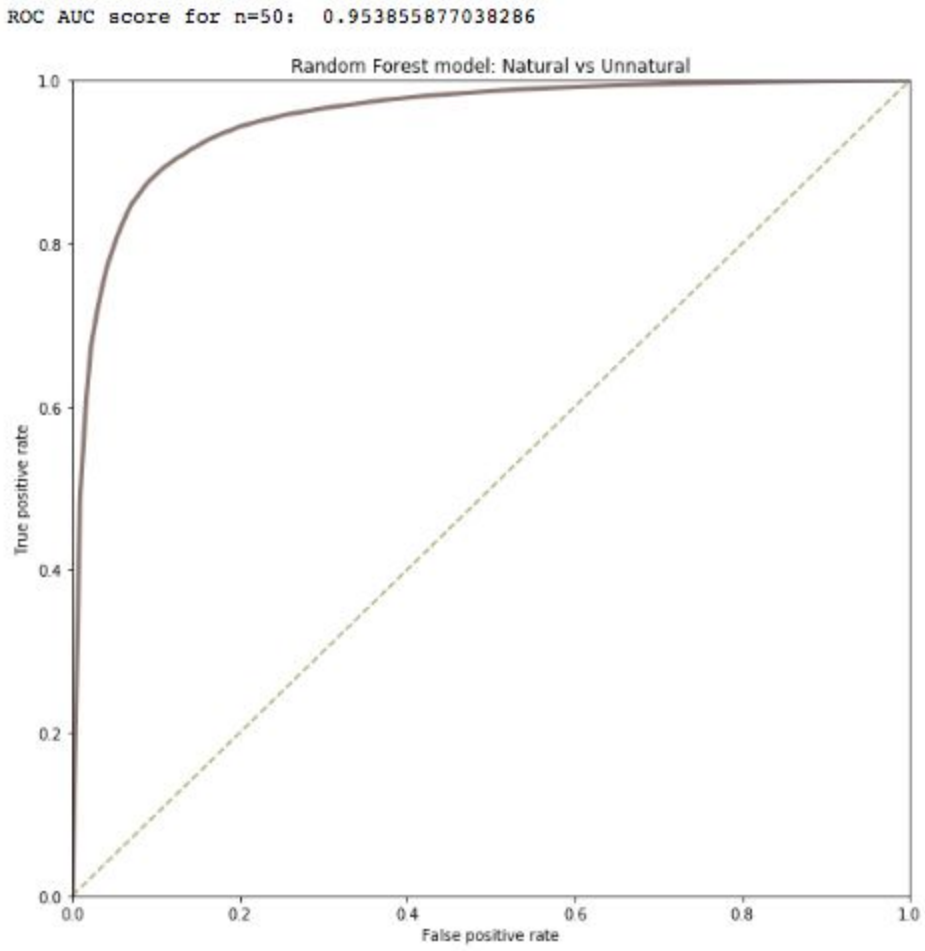


Fig. 8

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.