Is it a Planet?: Classifying TESS Observations

Abstract

Exoplanets, planets orbiting stars outside of our solar system, are difficult to detect. Since most exoplanets cannot be detected through direct imaging, they are often detected using the transit method, where astronomers detect exoplanets as they pass in front of their host star. NASA's latest mission to discover exoplanets, the Transiting Exoplanet Survey Satellite (TESS), brings back troves of information about objects of interest (potential exoplanets). We aim to create a model to determine which objects of interest are exoplanets. Our most successful model is a random forest model, where the most important variables were the "light-blocking amount" and planetary radius of the objects of interest. Lastly, we use this model to predict the status of unconfirmed objects of interest.

Background

Exoplanets, planets that exist outside of our Solar System, are almost impossible to discover through direct imaging. One method to detect them is the transit method, which observes potential exoplanets passing in front of their host stars. The Transiting Exoplanet Survey Satellite (TESS) is currently observing many stellar systems, looking for transiting planets, and if there is statistical evidence of one, the object becomes an "object of interest." All TESS objects of interest, or TOI, are classified as being confirmed planets (CP), as having been shown to not actually be a planet (FP), or as awaiting a final decision (PC). Our research objectives are to learn a classifier to distinguish between confirmed planets and false positives and use the best performing model to classify planetary candidates.

Methods

Data Collection

The TESS satellite collects different measurements from potential exoplanets and its host star. The data were downloaded by the class instructor on September 15, 2021 and preprocessed to eliminate uninformative columns. In addition to the predictors (see Appendix I), the potential exoplanets are labeled by their designations. There are 423 confirmed planets, 490 false positives, and 2480 planetary candidates that have yet to be classified as any of the previous two categories. In addition to performing the necessary transformations, we remove 141 observations that are considered as outliers. The resulting data are visualized as histograms on the right.

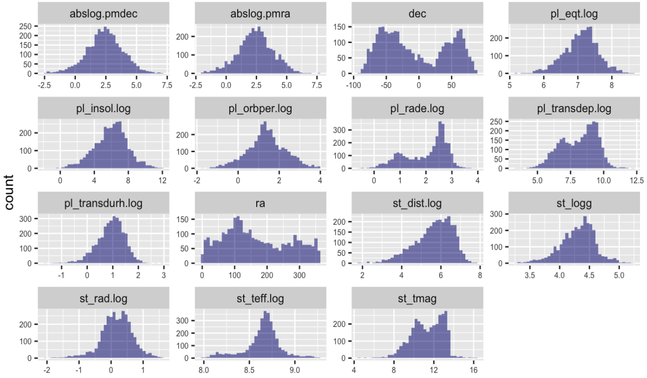


Figure 1: Histogram of predictor variables

Variables

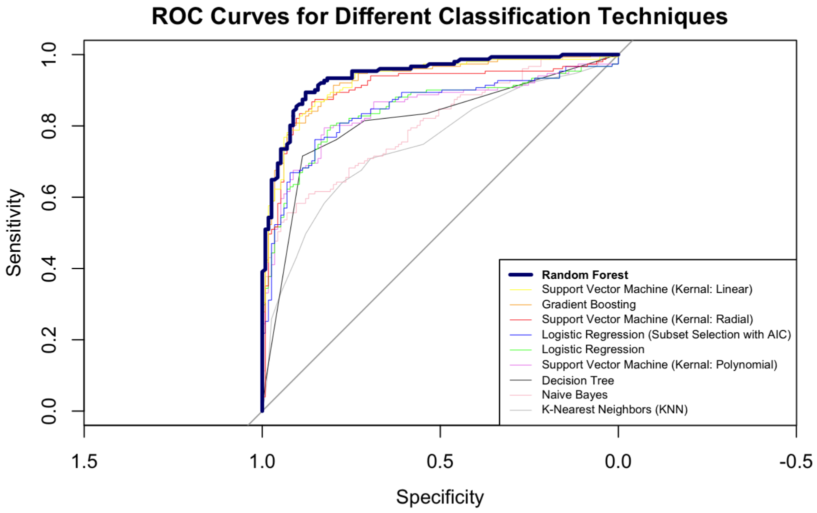
Our primary response variable is a categorical variable, `label`, which indicates what the identity of the planet is. The categories are planetary candidates, false positive or confirmed planets. Also, we do not pay too much attention to the celestial longitude, `ra`, and latitude, `dec`, variables because they do not indicate any more information than where in the sky the TESS satellite found the observations. We perform multiple dimensionality reduction and variable selection processes to determine whether there exists high multicollinearity in the data. We conclude that there is multicollinearity in the data. For example, we notice a near-perfect correlation between the amount of light the planet receives and the planet's surface temperature (see Appendix II). We use principal component analysis (see Appendix II) to determine that we only need to retain 11 principal components. We perform Best Subset Selection (see Appendix III.2) to conclude that 11 out of the 15 variables form the best subset as one of our models. We also attempt lasso and ridge regression (see Appendix III.3) and conclude that given the optimal value of the tuning parameter λ, we do not favor lasso or ridge over logistic regression and would not be removing any variables. Nevertheless, we do not actually need to remove any variables, as the purpose of this analysis is prediction. In not doing so, we ignore the effects of multicollinearity, as intended.

Statistical Analysis

All analyses are performed in R (version 4.1.0). In order to train and test the classifiers on labeled data, we initially remove the data for planetary candidates (PC), which are unlabeled data. With the remaining data, we use 70% of the data for training and 30% of the data for testing. We then use a number of classification techniques to build classifiers: logistic regression, log-forward subset selection based on Akaike Information Criteria (AIC), decision tree, random forest, extreme gradient boosting, K-nearest neighbors, Naive Bayes, and support vector machines with linear, polynomial, and radial kernels. We then generate final predictions using the best performing model and calculate Youden's J-statistic to determine a threshold for classification and determine the misclassification rate of the resulting model on the labelled data. We finally use the best performing model and the threshold to generate predictions for the unlabelled data.

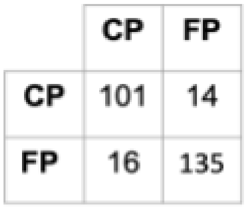
Results

After learning classifiers to categorize TESS objects of interest as confirmed planets (CP) and false positives (FP), we plot the receiver operator characteristics (ROC) curves for each model in the figure below and their area under the ROC curve (AUC) in the table below.



<u>Model Attempted</u>	<u>AUC</u>
Random Forest	**0.943**
Support Vector Machine w/ Linear Kernel	0.926
Extreme Gradient Boosting	0.926
Support Vector Machine w/ Radial Kernel	0.895
Logistic Regression (Subset Selection with AIC)	0.846
Logistic Regression	0.845
Support Vector Machine w/ Polynomial Kernel	0.844
Decision Tree	0.812
Naive Bayes	0.802
K-Nearest Neighbors	0.749

Figure 2/3: ROC Curves and AUC for all models



	CP	FP
CP	101	14
FP	16	135

Figure 4: Confusion Matrix

It appears that the random forest model, with an AUC of 0.943, performs the best out of the models attempted. We calculate Youden's J-statistic, where $J = sensitivity + specificity - 1$, to generate a threshold value that would allow us to identify both confirmed planets and false positives with high accuracy. This results in the confusion matrix to the right. Note that columns of the matrix denote predictions and the labels of observed objects of interest correspond to the row. The resulting misclassification rate (MCR) is 0.113. The specificity, the accuracy in predicting false positives, is 0.894, while the sensitivity, the accuracy in predicting confirmed planets, is 0.878.

Random forest models are an ensemble learning method in which many decision trees are grown deeply and then aggregated together to form one unified model. To learn a random forest model, the algorithm grows a number of deep, unpruned trees from a subset of the variables, and employs a simple majority vote to decide which class a certain observation should be categorized as. It uses a subset of the variables in order to avoid the problem of predictor variables dominating the top couple of splits of the tree each time. Random forest models utilize a majority vote in deciding the classification of some observations. It does better than considering any one of its constituent trees, improving the accuracy when compared to a singular decision tree and reducing overfitting that may come with using decision trees.

We can make some interpretations about the random forest model. In particular, we consider the mean decrease in accuracy as a metric of variable importance. This metric measures how much accuracy is lost if we remove the variable from the model, thereby measuring a variable's importance. Just like the exploratory data analysis indicated, the "light-blocking amount" of transit (in ppm) and planetary radius (in Earth radii) appear to be the most important variables, as expected. Additionally celestial longitude and celestial latitude, are among the least important, as intended. The variable importance plot displaying these findings appears on the right. For results of other models, refer to Appendix III.

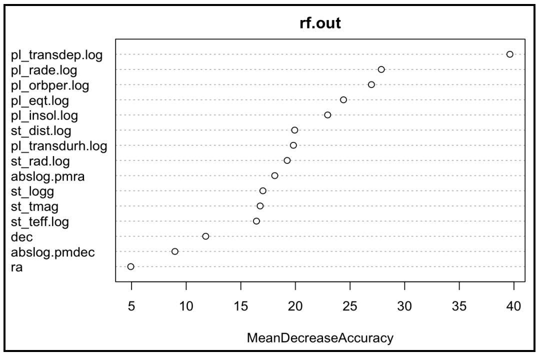


Figure 5: variable importance plot for random forest

We finally apply the random forest model to the unclassified planetary candidates. We run the 2480 observations that have not been classified in our dataset as of yet against our model and determine that 1392 of the planetary candidates will be eventually confirmed as confirmed planets, whereas the other 975 objects of interest will eventually be confirmed as false positives.

Discussion

One key limitation of this study is that if there are any inaccuracies recorded by TESS, it would have impacted our model. Additionally, one of the drawbacks of a random forest model is that we cannot use it for statistical inference because it is a predictive model. Hence, we can only use it to predict if an object of interest is an exoplanet. Another limitation of our study is that we have only used the ROC curve and resulting AUC values to compare the predictive ability of our models. Although it is a widely accepted way to compare predictive models, it would have been a more holistic approach had we used different metrics to compare and contrast the models. In addition, one limitation of using the MCR value is that it only measures whether binary classification was successful or not; the results may be misleading if the given data set is greatly imbalanced. In this case, even with extremely high AUC values, since the classification did not require much effort, we would not know whether a classifier can correctly predict a balanced, random data. Since the main goal of this project was prediction and accurate classification, the effects of multicollinearity between predictor variables were not considered important. However, it is clear that many predictor variables are strongly correlated with each other (see figure 6). Taking into account multicollinearity, we may be able to conduct more statistical inference in order to better understand how each predictor variable relates to each classified class of planets.

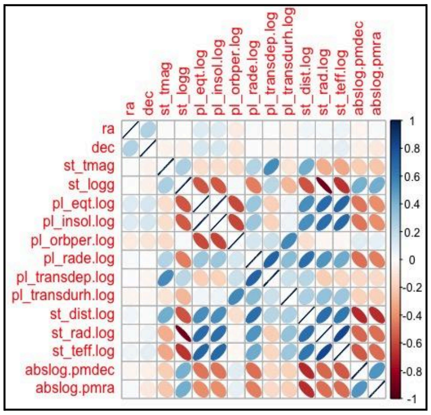


Figure 6: correlation matrix of predictor variables

References

[1]James, Gareth, et al. *An Introduction to Statistical Learning with Applications in R*. Springer, 2017.

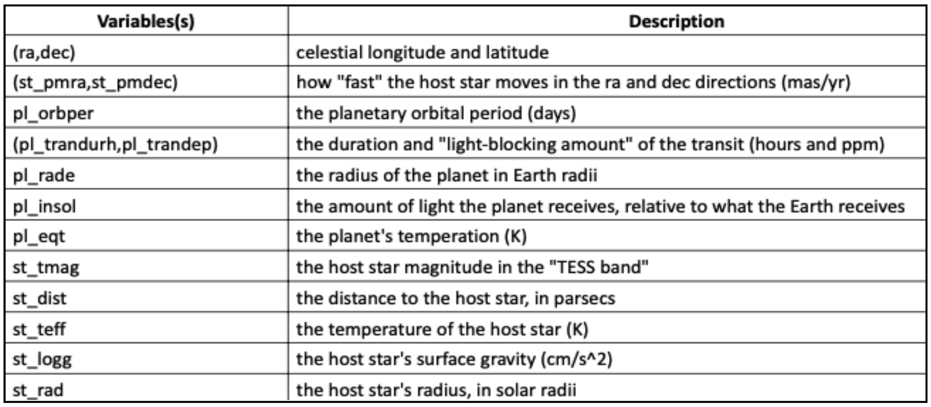
[2]NASA Exoplanet Archive, online at https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=TOI

Appendix

I. *Predictor variables used in the dataset.*

TABLE 1

Predictor variables used in the data set and their description and units



Variables(s)	Description
(ra,dec)	celestial longitude and latitude
(st_pmra,st_pmdec)	how "fast" the host star moves in the ra and dec directions (mas/yr)
pl_orbper	the planetary orbital period (days)
(pl_trandurh,pl_trandep)	the duration and "light-blocking amount" of the transit (hours and ppm)
pl_rade	the radius of the planet in Earth radii
pl_insol	the amount of light the planet receives, relative to what the Earth receives
pl_eqt	the planet's temperation (K)
st_tmag	the host star magnitude in the "TESS band"
st_dist	the distance to the host star, in parsecs
st_teff	the temperature of the host star (K)
st_logg	the host star's surface gravity (cm/s^2)
st_rad	the host star's radius, in solar radii

II. *Principal Component Analysis (PCA)*

Principal component analysis is used to determine the optimal number of principal components (linear combinations of initial variables) that explains much of the variance, such that another principal component that may be added contributes marginally less information about the variance than any of the previously included principal components. We plot this effect in the principal components versus cumulative variance plot below:

FIGURE 1

Cumulative variance versus the number of principal components

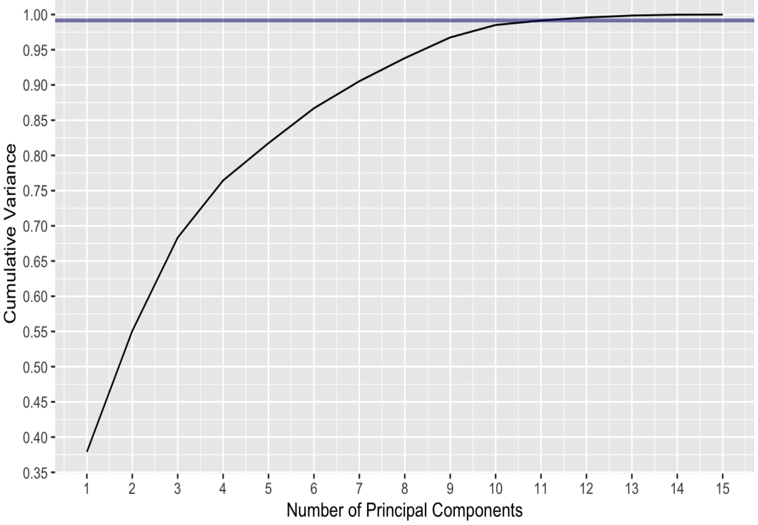


TABLE 2

Specific cumulative variance by number of PC

Number of PC	Cumulative Variance
1	0.3789
2	0.5508
3	0.6834
4	0.7644
5	0.8174
6	0.8669
7	0.9053
8	0.9381
9	0.9675
10	0.9852
11	0.9915
12	0.9958
13	0.9986
14	0.9998
15	1.0000

We typically choose the number of principal components to retain as such that the cumulative variance at that number is more than 0.99. This happens when the number of principal components is 11. However, as stated in the report, we do not remove principal components because our intent in this study is prediction.

III. Other Classification Models Attempted

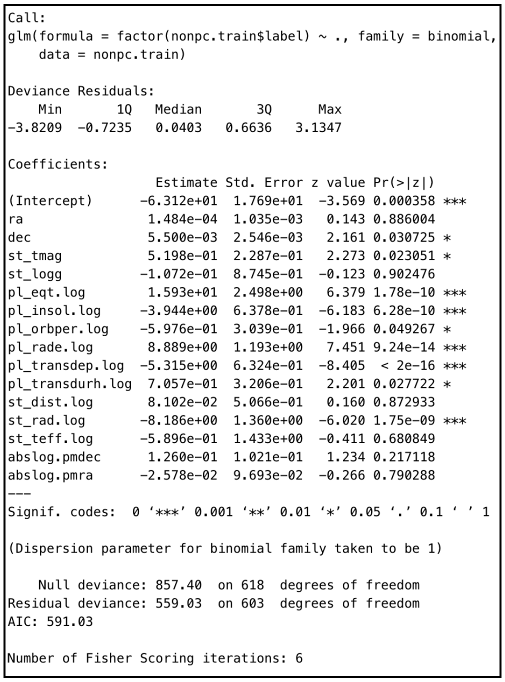
1. *Logistic regression* (see Figure 2) Logistic regression is typically used for datasets with a response variable that can only take on two discrete values that are assumed to map to 0 and 1.
2. *Logistic Regression Using Best Subset Selection with AIC* (see Figure 3) Subset selection is used to increase interpretability and decrease overfitting. AIC, or Akaike Information Criterion, estimates prediction error. It has the following formula:

$$\mathrm{AIC} = \frac{1}{n\hat{\sigma}^2}(\mathrm{RSS} + 2k\hat{\sigma}^2)$$

where RSS is the residual sum of squares, the additive terms penalty terms that increase with k, and $\hat{\sigma}$ an estimate of the variance of the linear regression. Using this metric tends to overfit the data by selecting the model with more variables such that all the important variables are chosen but the list of variables will include some unimportant ones. We utilize forward-stepwise selection incorporating this measure as a metric of bias to greedily search for the variables that yield the optimal subset selection. The model we obtain determines that 11 of the variables (`dec`, `pl_insol.log`, `pl_orbper.log`, `st_rad.log`, `pl_rade.log`, `pl_eqt.log`, `pl_transdep.log`, `pl_transdurh.log`, `st_dist.log`, `st_tmag`, and `st_teff.log`) form the best subset selection. We then use logistic regression on this subset of variables, displaying the result below:

FIGURE 2

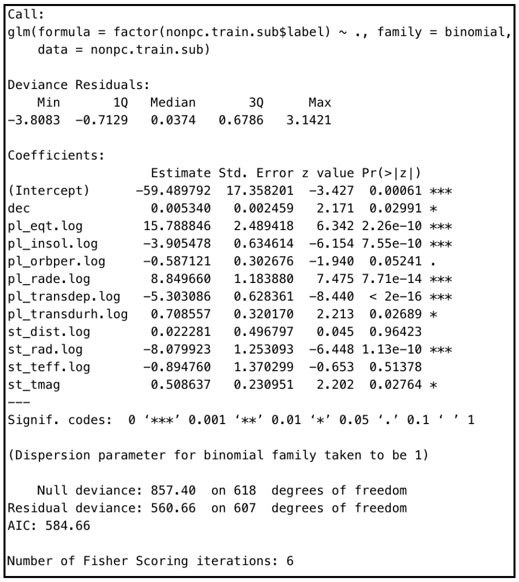
Output for Logistic Regression



```
Call:
glm(formula = factor(nonpc.train$label) ~ ., family = binomial,
    data = nonpc.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.8209  -0.7235   0.0403   0.6636   3.1347

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -6.312e+01  1.769e+01  -3.569 0.000358 ***
ra                1.484e-04  1.035e-03   0.143 0.886004
dec               5.500e-03  2.546e-03   2.161 0.030725 *
st_tmag           5.198e-01  2.287e-01   2.273 0.023051 *
st_logg          -1.072e-01  8.745e-01  -0.123 0.902476
pl_eqt.log        1.593e+01  2.498e+00   6.379 1.78e-10 ***
pl_insol.log     -3.944e+00  6.378e-01  -6.183 6.28e-10 ***
pl_orbper.log    -5.976e-01  3.039e-01  -1.966 0.049267 *
pl_rade.log       8.889e+00  1.193e+00   7.451 9.24e-14 ***
pl_transdep.log  -5.315e+00  6.324e-01  -8.405  < 2e-16 ***
pl_transdurh.log  7.057e-01  3.206e-01   2.201 0.027722 *
st_dist.log       8.102e-02  5.066e-01   0.160 0.872933
st_rad.log       -8.186e+00  1.360e+00  -6.020 1.75e-09 ***
st_teff.log      -5.896e-01  1.433e+00  -0.411 0.680849
abslog.pmdec      1.260e-01  1.021e-01   1.234 0.217118
abslog.pmra      -2.578e-02  9.693e-02  -0.266 0.790288
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 857.40  on 618  degrees of freedom
Residual deviance: 559.03  on 603  degrees of freedom
AIC: 591.03

Number of Fisher Scoring iterations: 6
```

FIGURE 3

Output for Logistic Regression Using Best Subset Selection with AIC



```
Call:
glm(formula = factor(nonpc.train.sub$label) ~ ., family = binomial,
    data = nonpc.train.sub)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.8083  -0.7129   0.0374   0.6786   3.1421

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -59.489792  17.358201  -3.427  0.00061 ***
dec                0.005340   0.002459   2.171  0.02991 *
pl_eqt.log        15.788846   2.489418   6.342 2.26e-10 ***
pl_insol.log      -3.905478   0.634614  -6.154 7.55e-10 ***
pl_orbper.log     -0.587121   0.302676  -1.940  0.05241 .
pl_rade.log        8.849660   1.183880   7.475 7.71e-14 ***
pl_transdep.log   -5.303086   0.628361  -8.440  < 2e-16 ***
pl_transdurh.log   0.708557   0.320170   2.213  0.02689 *
st_dist.log        0.022281   0.496797   0.045  0.96423
st_rad.log        -8.079923   1.253093  -6.448 1.13e-10 ***
st_teff.log       -0.894760   1.370299  -0.653  0.51378
st_tmag            0.508637   0.230951   2.202  0.02764 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 857.40  on 618  degrees of freedom
Residual deviance: 560.66  on 607  degrees of freedom
AIC: 584.66

Number of Fisher Scoring iterations: 6
```

3. *Lasso and Ridge Regression* Lasso (figure 4) and ridge regression (figure 6) are shrinkage methods that are alternatives to best subset selection. They have the following formula:

$$\text{lasso}: \ \text{RSS} + \lambda \sum_{i=1}^{p} |\beta_i|$$

$$\text{ridge}: \ \text{RSS} + \lambda \sum_{i=1}^{p} \beta_i^2$$

where RSS again is the residual sum of squares and the effects of the additive penalty is dictated by the magnitude of the tuning parameter, λ. The closer λ is to zero, it can be said the penalty terms have no effect and would have the same results as a regular linear/logistic regression.

FIGURE 4

Standardized coefficients of predictor variables in the context of lasso regression

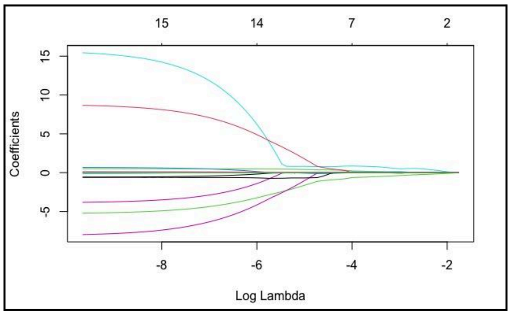
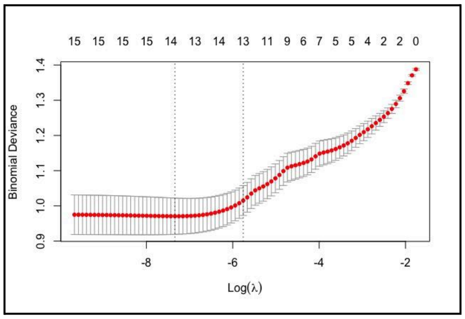


FIGURE 5

Result of k-fold cross validation for logistic regression using lasso regression, where the resulting $log(\lambda)$ value is indicated by the vertical dotted line on the left hand side



In our analysis, figure 5 does not show enough evidence that the binomial deviance chances have significant changes before applying lasso and at the log-lambda value. The minimum $log(\lambda)$ value appears at -7.340, or a λ of 0.000649. Therefore, there is no reason to favor lasso over logistic regression in our case.

FIGURE 6

Standardized coefficients of predictor variables in the context of Ridge Regression

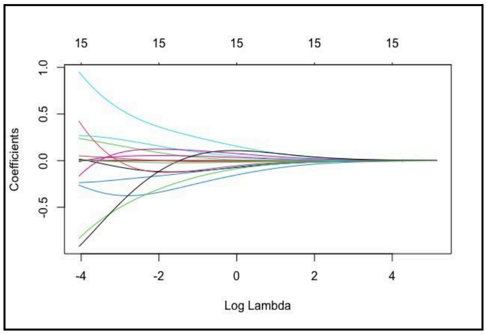
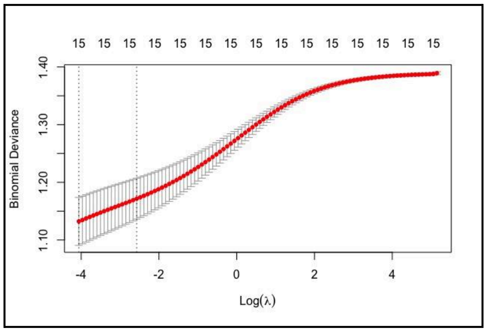


FIGURE 7

Result of k-fold cross validation for logistic regression using Ridge Regression: resulting $log(\lambda)$ value indicated by the vertical dotted line on the left hand side



Similarly to lasso regression, according to figure 7, the minimum $log(\lambda)$ value appears at -4.060, or a λ of 0.0172, so there is no reason to favor ridge over logistic regression as well.

4. *Decision Tree* Recursive binary splitting is used to produce a tree using training data and the splits are based on the reduction of the Gini coefficient, a value that reduces if each node consists of more objects of a single class. Through cost complexity pruning, the sequence of best subtrees is produced (see figure 9). We prune the tree when the complexity parameter corresponding to the leaves in the tree is 0.025 (see figure 8), generating the pruned tree on the right.

FIGURE 8
Resulting plot of cost complexity pruning

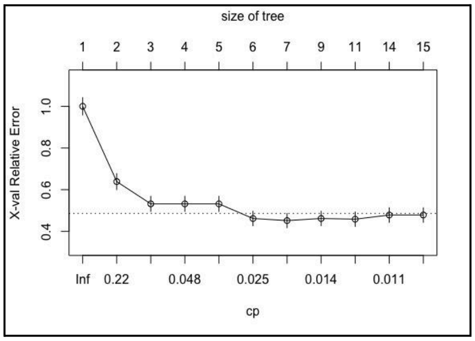
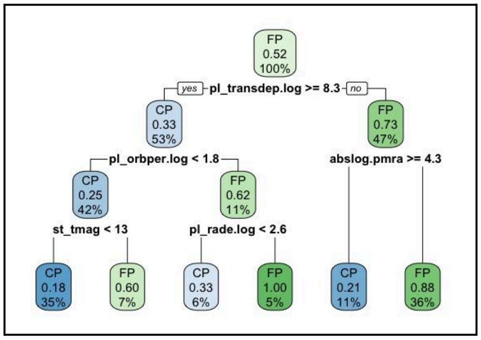
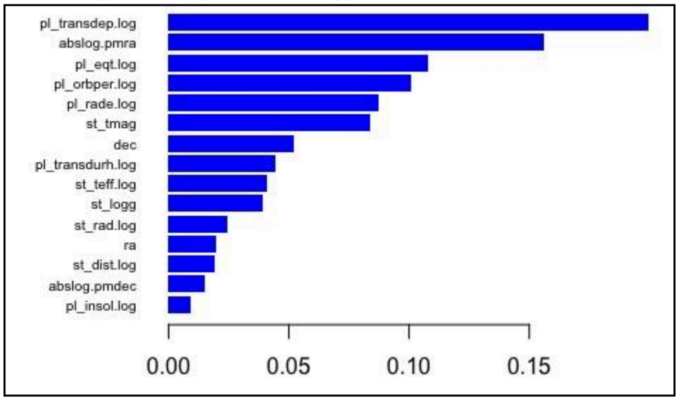


FIGURE 9
Pruned tree resulting from cost complexity pruning(figure 4)



5. *Extreme Gradient Boosting* A model is learned by first fitting a tree to the data. The residuals are updated and the data that is used to fit the next tree would be the set of residuals. Therefore, the model would be able to resolve the issues of fitting by using the residual values as the new data. Extreme gradient boosting allows measurement of variable importance by measuring the gain, which is the contribution of each variable to the model based on the total gain of the feature's splits (see figure 6).

FIGURE 10
Variable importance plot resulting from Extreme Gradient Boosting



6. Naive Bayes Naive Bayes attempts to predict the probability of a class given the predictor variables using Bayes' Rule, where we have

$$p(C_k|\mathbf{x}) \propto p(C_k)\prod_{i=1}^{n} p(x_i|C_k)$$

and one fundamental simplifying assumption (that gives this model its "naive" characterization) that there is mutual independence between the predictor variables. $p(C_k)$ is known as the prior probability and determines the probability that a class is represented in the data, while $p(x_i|C_k)$ estimate the probability of the existence of a value given the presence of a certain class (known as the likelihood probability).

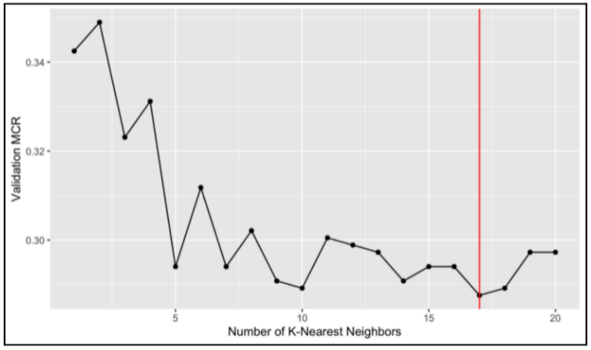
7. *K-Nearest Neighbors* K-nearest neighbors, or KNN, is another statistical learning technique for our purposes. It examines the K nearest neighbors given its quantitative explanatory variables, and determines the proportion of its neighbors that are of a certain class. In mathematical notation,

$$P[Y = j|\mathbf{x}] = \frac{1}{k}\sum_{i=1}^{k} \mathbb{I}(Y_i = j)$$

where $\mathbb{1}(\cdot)$ is an indicator function that returns 1 if the argument is true and 0 otherwise. To obtain k, we used cross-validation to tune k based on the mis-classifcation rate. The results are displayed in the graph below in figure 11. The ideal k is one that has the minimum validation misclassification rate, which in this case, is 17.

FIGURE 11

Number of K-nearest neighbors and the resulting validation misclassification rates



8. *Support Vector Machines* Support vector machines, or SVMs, determine boundaries that allow overlap and misclassification, doing so in the native space of predictor data. The overlap is controlled by tuning through cross validation a cost parameter C that becomes more tolerant for misclassification as it increases. It uses the kernel-trick by transforming variables from the native space to a higher dimensional space without actually performing those transformations. We use several kernels including linear, polynomial, and radial kernels. Linear kernels produce a hyperplane as the boundary, which actually does not employ the kernel trick, and we determine an optimum cost C of 3.981. Polynomial kernels use the same idea of cost but also further requires tuning another constant d, the degree of the polynomial kernel. We obtain an optimum cost C of 562.341 and a degree d of 1. For the purposes of understanding the machine learning technique, if the degree were 2, then it transforms bivariate explanatory data containing variables X and Y to a 5-dimensional space with axes X, Y, XY, X^2, and Y^2, in effect axes incorporating different combinations of the variable. Lastly, radial kernels again have a cost parameter C of 1000 but it also has another parameter γ of 0.00316. The higher the parameter of γ, the higher the influence from observations that are farther away than the boundary obtained.