Predicting Exoplanet Disposition Using NASA's Kepler Space Observatory Data and Machine Learning

Capstone Project 1 Final Report

Springboard Data Science Career Track
March 26^{th}, 2019

Logan Rudd

1 Introduction

Scientists and researchers in organizations across the globe are currently searching for exoplanets (planets that orbit a star outside our solar system) in the habitable zone of their stars where liquid water might exist on the surface of the planet in order to answer one of the most timeless questions, 'Are we alone?' NASA's Keppler space observatory in particular was developed to survey the region of the Milky Way galaxy closest to us to reveal hundreds of Earth-size and smaller planets in or near the habitable zone so that we can determine the fraction of star systems in our galaxy that might contain such planets.

1.1 About the Dataset

Throughout it's mission duration, which lasted from May 12, 2009 to November 15 2018, the Kepler space observatory has successfully located 9,564 KOIs (Kepler Objects of Interest) for which the data is publicly available online from the NASA Exoplanet Archive.[1] Of the 9,564 KOIs, 2,298 are confirmed exoplanets, 4,841 are false positives, and 2,425 are classified as candidates (not yet confirmed nor shown to be false positives). For each entry in the KOI dataset there are 140 columns/features that organized into the following 8 categories: Identification Columns, Exoplanet Archive Information, Project Disposition Columns, Transit Properties, Threshold-Crossing Event (TCE) Information, Stellar Parameters, KIC (Kepler Instrument Characteristic) Parameters, and Pixel-Based KOI Vetting Statistics. For the purpose of this project our aim is to predict the KOI disposition from using only the last 5 categories: Transit Properties, Threshold-Crossing Event (TCE) Information, Stellar Parameters, KIC (Kepler Instrument Characteristic) Parameters, and Pixel-Based KOI Vetting Statistics.

1.2 Cleaning the Dataset

Before performing any analysis, the dataset was first downloaded from NASAs Exoplanet Archive and imported to the Jupyter Notebook. Missing values were then visualized using MissingNo python library, and columns with all or mostly missing values were deleted (Appendix, Figure 1). Column names were cleaned up by removing the 'KOI_' string before each feature name, and feature engineering was performed to reduce/simplify the columns that describe the error on the measurement. For example, If a numerical measurement has an upper and lower bound error associated with it (eg: columns that end with '_err1' and '_err2'), the two columns were turned into one column that describe the difference in the upper and lower bounds. The resulting dataframe was scanned to find and remove any columns containing the same entry for every KOI, as they didn't supply any useful info for classifying the KOIs. All the KOIs that are dispositioned as 'CANDIDATE' were removed since we're only interested in classifying a KOI as 'FALSE POSITIVE' or 'CONFIRMED'. Missing values were visualized once more as a matrix to see if there are any rows that contain mostly missing values that should be removed (Appendix, Figure 2). The matrix showed that there were relatively few rows that contained mostly missing values, so the dataframe was left as is. The resulting cleaned dataset contains 7,141 rows and 97 columns/features, 77 of which are features in the 5 categories of interest for which will be used to predict KOI disposition. The data was then split 80/20 into training/testing sets so that exploratory data analysis can be performed only on the data that we our training our algorithms to. In this way, predictors are screened from only the training data, and the cross-validated error rate on the models for the test data won't be artificially low due to leaked information from the test/holdout set.[2]

2 Initial Findings

In the data storytelling section, various features were explored to see what kind of relationships exist between the disposition of a KOI – whether it is a confirmed exoplanet or false positive – and its features. In the Kepler Objects of Interest dataset obtained from NASAs Exoplanet Archive, 7 of the columns are non-numeric: fittype, parm_prov, tce_delivname, quarters, datalink_dvr, datalink_dvs, and sparprov. Of these features, the two features that end in 'prov' and the 'tce_delivname' correspond only to the data release number and period that the features for their corresponding category were released, so those two were ignored. Furthermore, the two 'datalink...' features are just the web URLs where the data validation report and summary can be found, so those were ignored as well. That leaves the two features 'fittype' (the statistical fit method used to calculate the planetary parameters) and 'quarters' (a bit string indicating which quarters of Kepler data were searched for transit signatures) to be explored.

First we explored the feature 'fittype' to understand how it plays a role on the disposition of the KOI. A barplot was made with the training data set showing the number of CONFIRMED and FALSE POSITIVE KOIs for each fittype. The bar plot showed that KOIs where the fit type Least Squares was used are all classified as 'FALSE POSITIVE' , and less than 0.47% of KOIs not using any fit and 2.0% of KOIs with using the Markov Chain Monte Carlo fit are classified as 'CONFIRMED' (Apendix, Figure 3). Since the majority of 'CONFIRMED' KOIs fell under the fit type category LS+MCMC, the 'fittype' feature was binarized to engineer a new column 'LS+MCMC' where it is 1 if LS+MCMC is used and 0 if not.

Next, the feature 'quarters' was explored further. First the missing values were filled in with the mode, then the number quarters the KOI was searched for transits were counted by counting the number of ones in the bit string. Another barplot was then made showing the disposition by number of quarters searched (Appendix, Firgure 4). This particular feature showed no strong correlation to the disposition/classification of the KOI.

Histograms were then plotted of several features by disposition, so for each feature there are two histograms, one showing the number of CONFIRMED exoplanets, and the other showing the number of FALSE POSITIVES (Appendix, Figures 5-10). After noticing that several features appear to have a statistically different mean between the two dispositions I performed a 2 sample t-test to see how many features of the 77 have significantly different means between the two classes/dispositions, and in turn help us to identify features that may give us more info on whether or not a KOI is a CONFIRMED or FALSE POSITIVE exoplanet. The null hypothesis for our t-test was that both "CONFIRMED" and "FALSE POSITIVE" KOIs have the same mean for any given numerical feature. The alpha level was set to 0.05, and a list of tuples containing all of the features name and their corresponding p-values was constructed. The results showed that out of 77 features, 68 of them had a p-value less than our alpha level for which we can reject our null hypothesis. The feature importance was then computed using Sklearns ExtraTreesClassifier (Appendix, Figure 11) and compared to the list that we made of the features sorted by p-values. The results were that 8 out 78 features appeared on both lists of top 10 important features. The correlation coefficient between the feature importance from sklearns ExtraTreesClassifier and the p-values for the top 10 features was then calculated to be -0.35 showing a weak correlation between the two.

3 Experimentation

In order to determine what features were most effective in classifying Kepler Objects of Interest as a confirmed exoplanet or false positive, four different sets of features were selected to fit the algorithms to. Each set respresents the top 25, 50, 75, and 100% of the most important features as calculated by SKlearn's ExtraTreesClassifier. For each model we first imputed the missing values of the data with the mean, then we scaled the data by replacing the original values with their respective z-scores. Lastly, we ran a hyperparamter search in conjunction with a 5-fold cross validation fit for each model using the Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) as our scoring metric.

All algorithms performed very well overall, with a ROC AUC score above 0.97 using only the top 25% most important features (Appendix, Figure 13). Although Random Forset and XGBoost had nearly identical results and tied in terms of highest ROC AUC score (0.997), XGBoost had a significantly lower runtime, and for that reason XGBoost was selected to predict the test data. For all algorithms, ROC AUC score did not increase with more than the top 75% of features. The following table summarizes ROC AUC score and Total CPU runtime for each of the algorithms used:

Top % of feature importances used:	25%	50%	75%	100%
Support Vector Machine	0.974 (0.003) 52.6s	0.983 (0.002) 1m 23s	0.985 (0.002) 2m 14s	0.985 (0.002) 3m 2s
Logistic Regression	0.983 (0.003) 25.7s	0.989 (0.001) 1m 23s	0.990 (0.002) 1m 23s	0.990 (0.001) 2m 5s
Random Forest	0.994 (0.002) 8m 41s	0.996 (0.002) 14m 28s	0.997 (0.002) 17m 44s	0.997 (0.002) 21m 15s
XGBoost	0.994 (0.002) 7m 4s	0.996 (0.001) 9m 37s	0.997 (0.001) 12m 49s	0.997 (0.001) 16m 41s

4 Results

Prediction of the test data showed an ROC AUC score of 0.996, meaning that there is a 99.6% chance that the model will be able to distinguish between a confirmed exoplanet and a false positive (Appendix, Figure 14). The confusion matrix in Figure 1 below shows that 12 out of 448 'CONFIRMED' exoplanets were predicted as 'FALSE POSITIVES', and 18 out of 981 'FALSE POSITIVES' were predicted as 'CONFIRMED', showing that the model is slightly better at predicting false positives than confirmed exoplanets. This makes sense as there are nearly twice as many more false positives to train the data on as there are confirmed exoplanets.

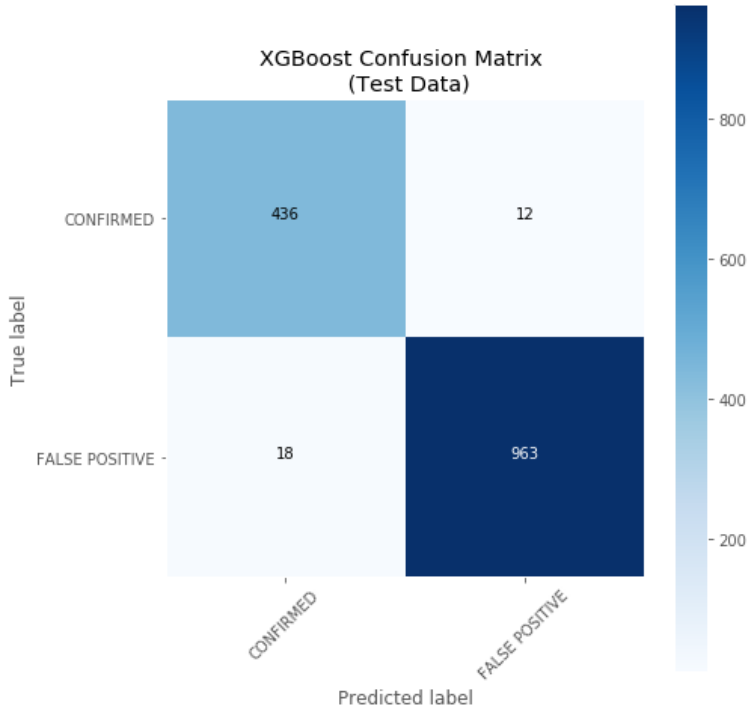

Figure 1: Confusion matrix of XGBoost prediction results on test data

5 Conclusion

A binary classifier was successfully created using data from NASA's Kepler Space Telescope to predict a true exoplanet from a false positive based on features such as transit/stellar properties and instrument characteristic parameters. After analyzing the performance of various machine learning algorithms, XGBoost turned out to be a great machine learning model for this particular binary classification problem.

References

[1] NASA Exoplanet Archive. *Cumulative KOI Data*, 2018 `https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=cumulative`

[2] Hastie, T., Tibshirani R., Friedman, J. (2009). *The Elements of Statistical Learning 2 ed.*. Manhattan, NY: Springer

Appendix

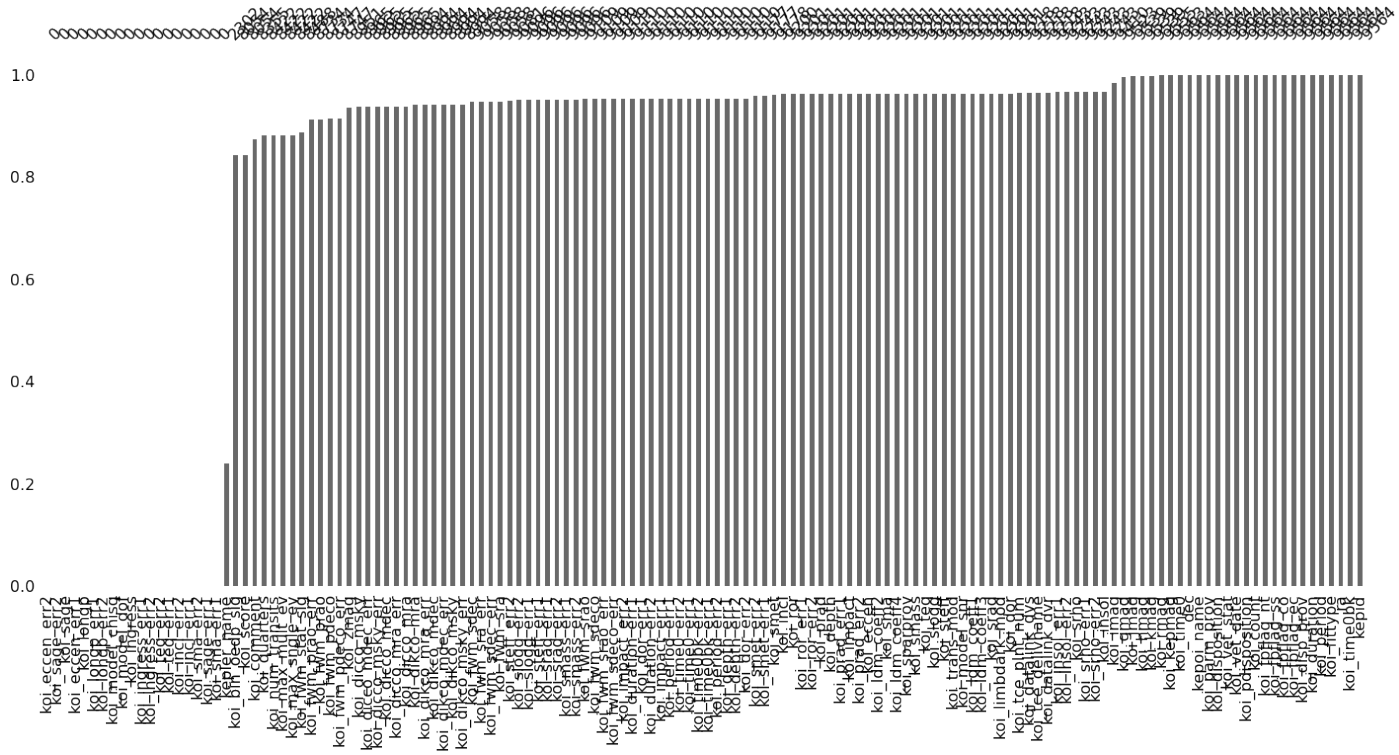

Figure 2: Barplot of missing values by column/feature

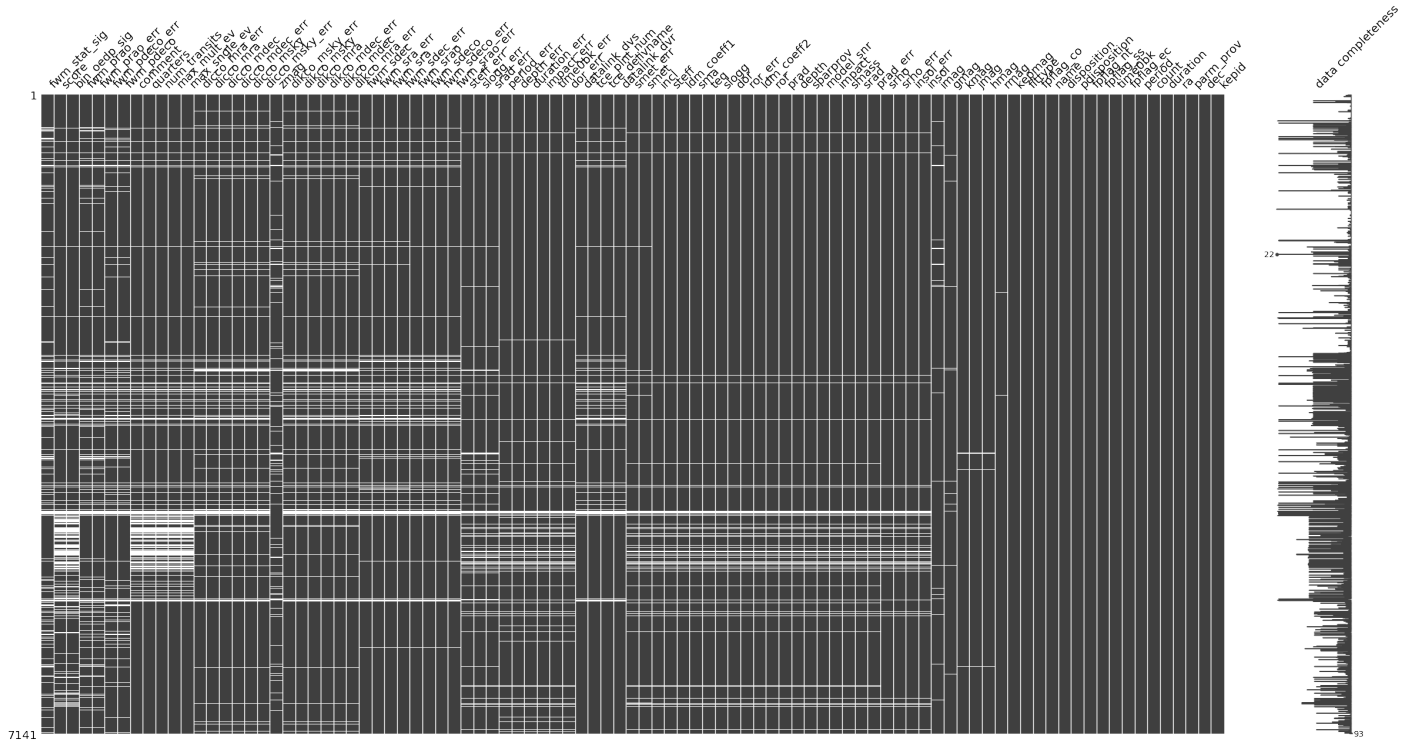

Figure 3: Matrix of missing values by row and column/feature

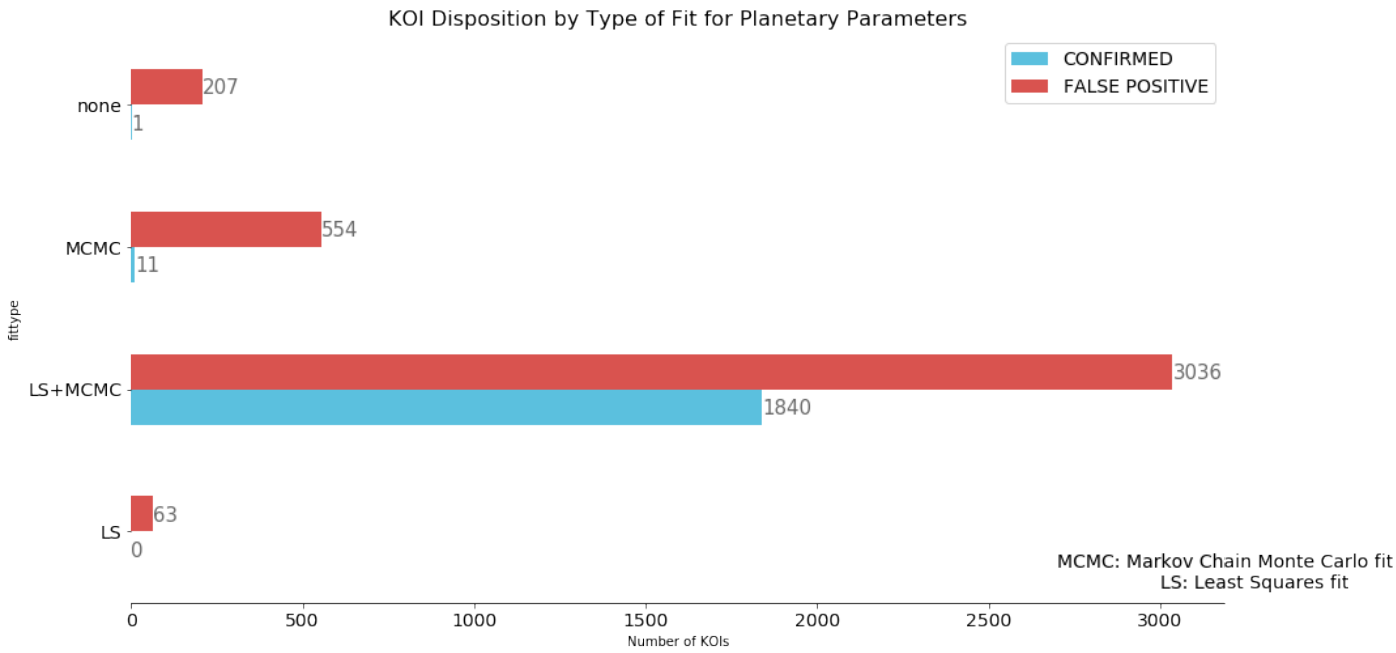

Figure 4: KOI disposition by statistical fit method used to calculate the planetary parameters

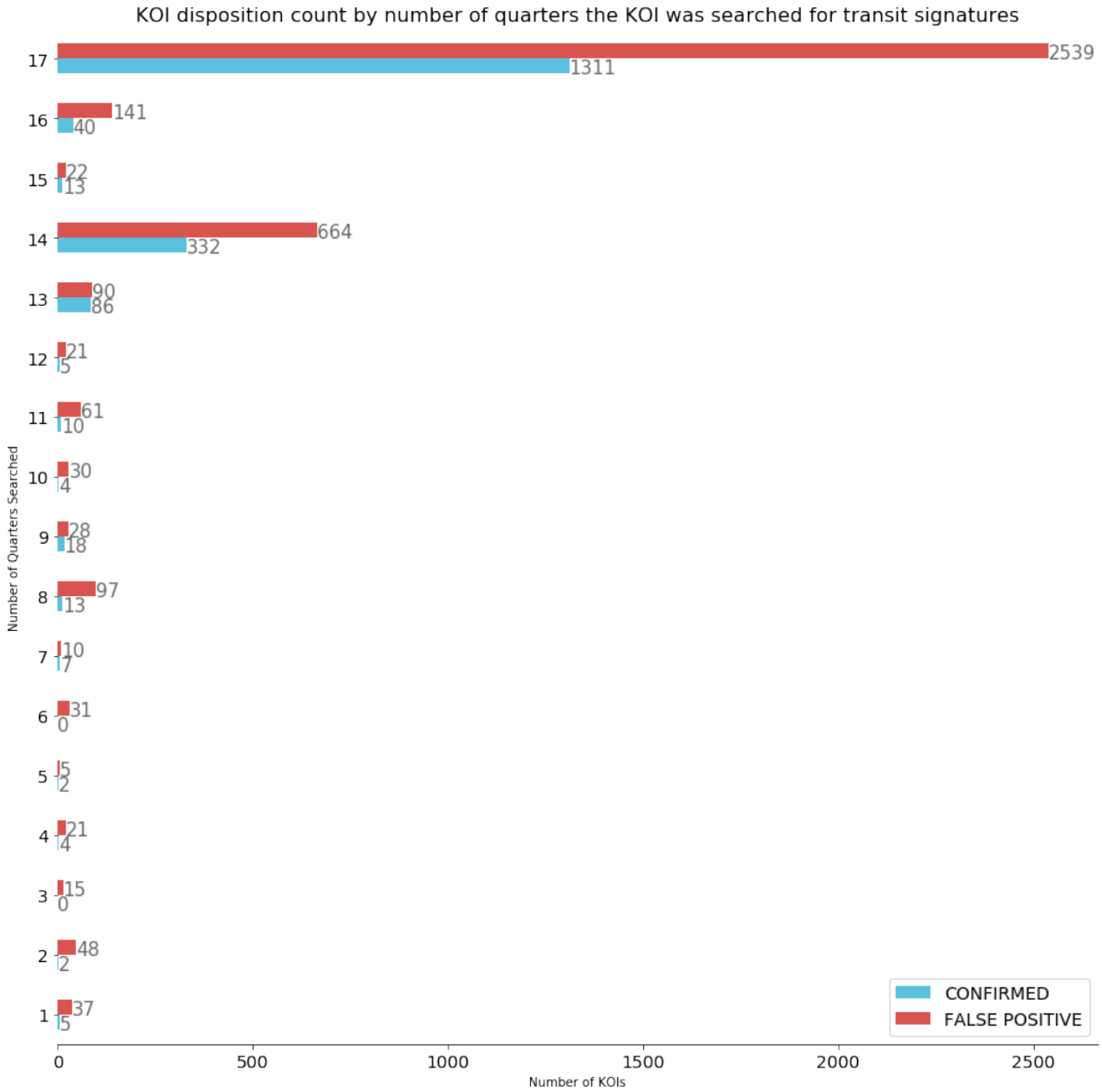

Figure 5: KOI disposition by number of quarters searched for transit signatures

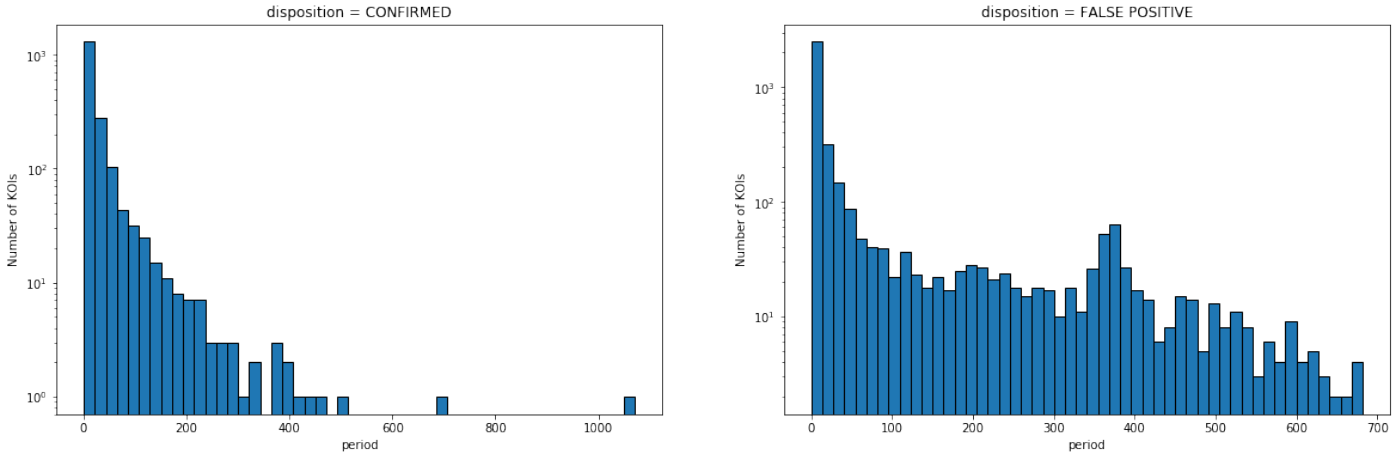

Figure 6: KOI 'period' histograms by disposition

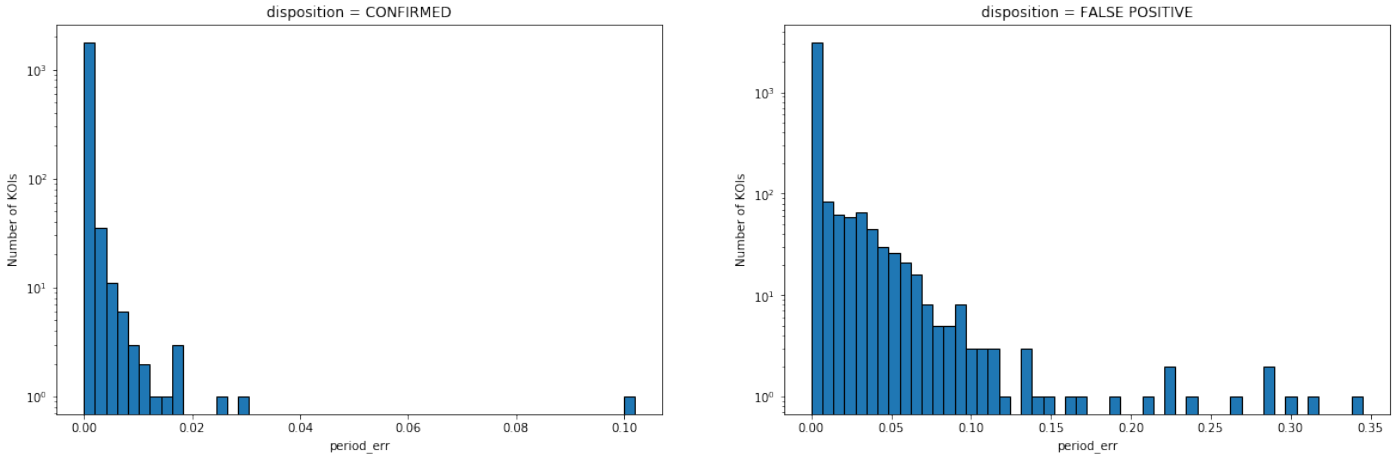

Figure 7: KOI 'period_err' histograms by disposition

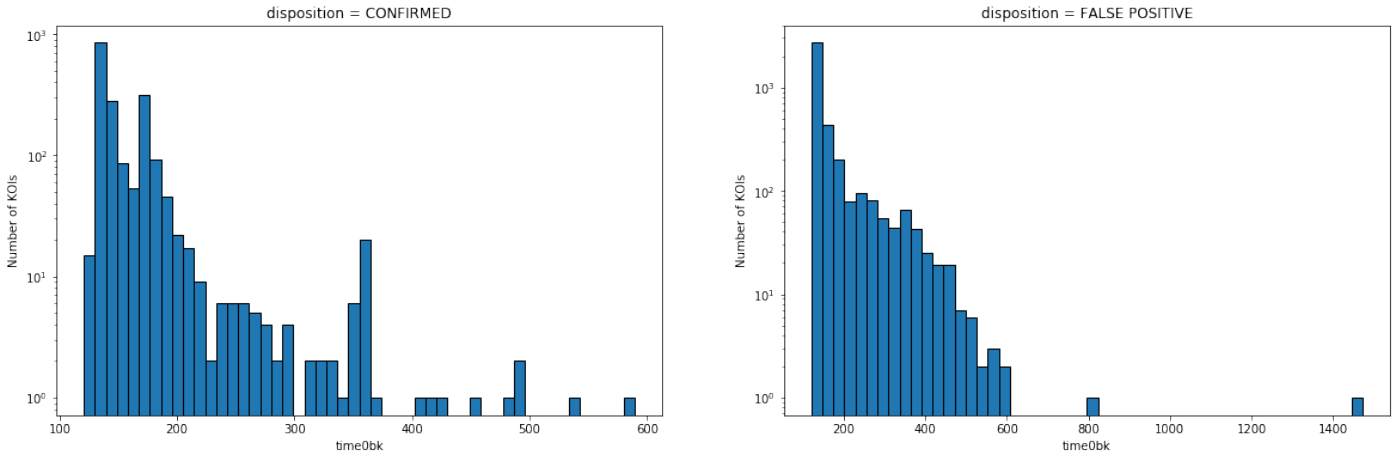

Figure 8: KOI 'time0bk' histograms by disposition

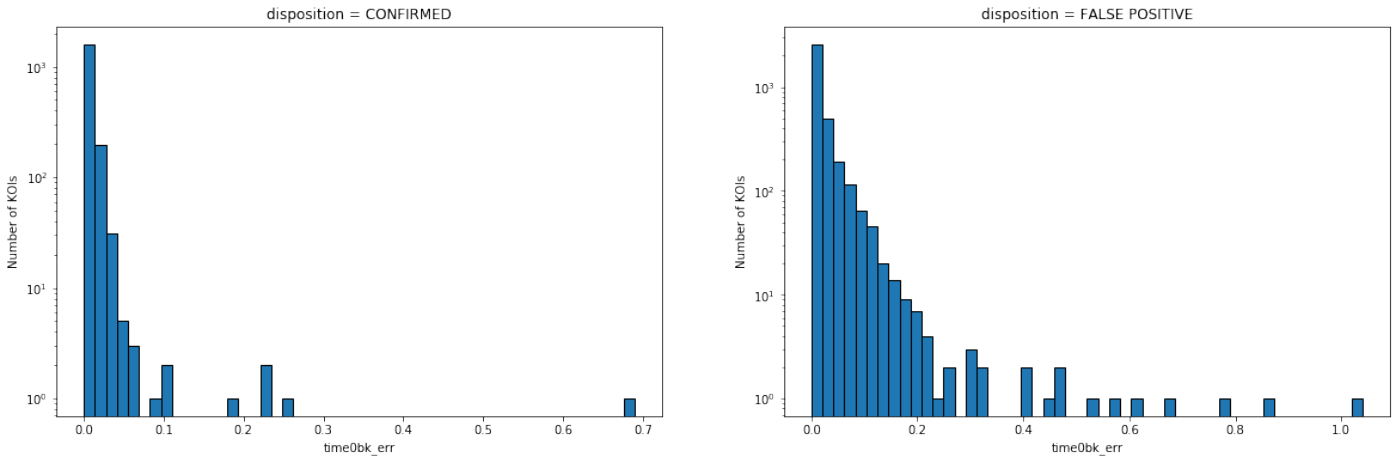

Figure 9: KOI 'time0bk_err' histograms by disposition

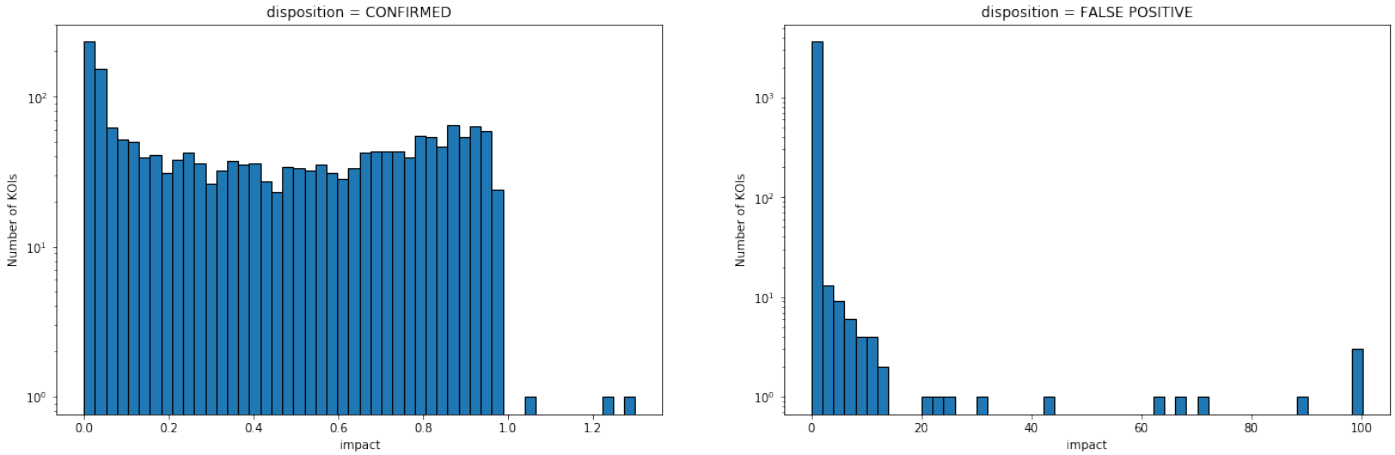

Figure 10: KOI 'impact' histograms by disposition

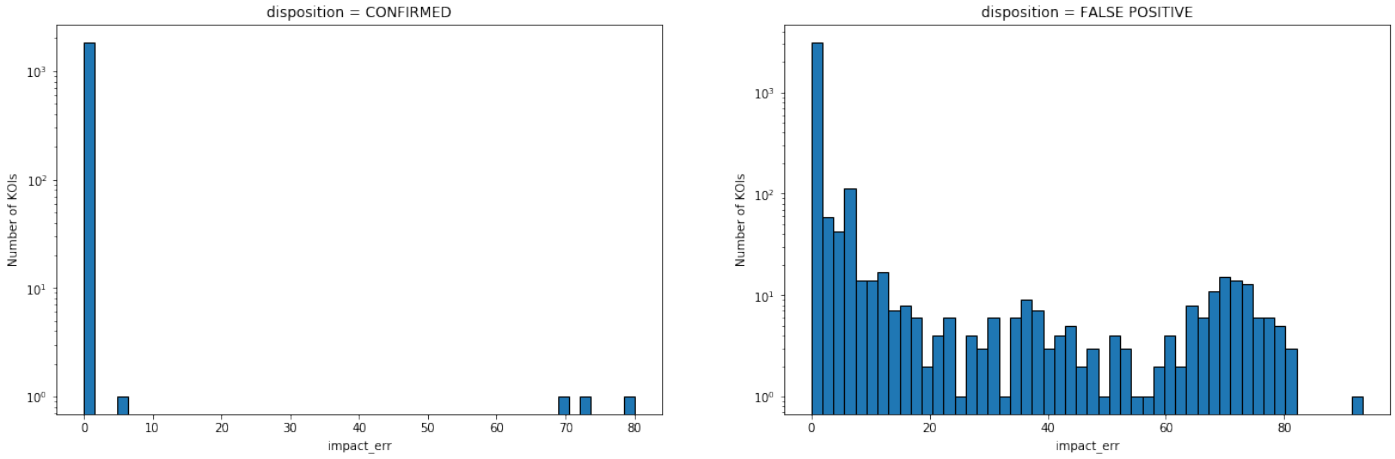

Figure 11: KOI 'impact_err' histograms by disposition

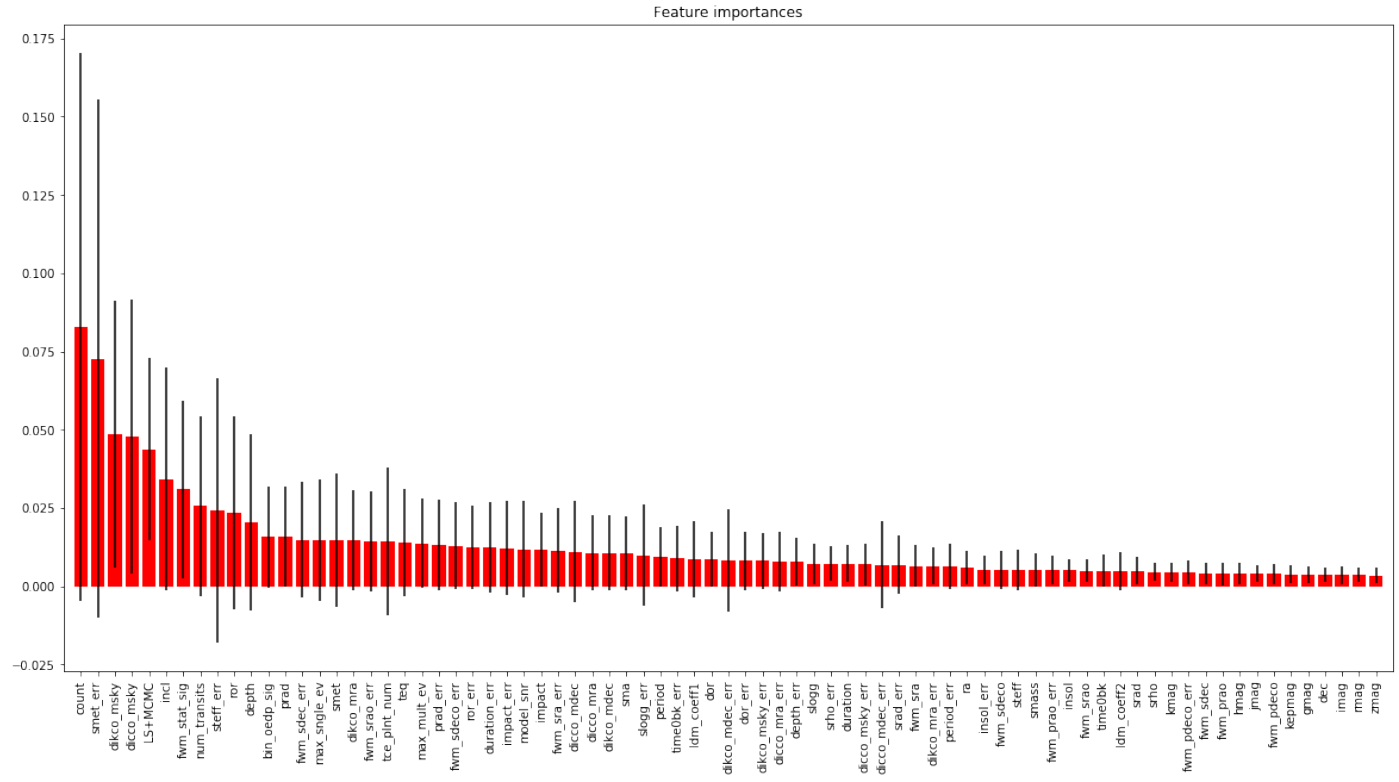

Figure 12: Scikit-learn's ExtraTreesClassifier feature importance

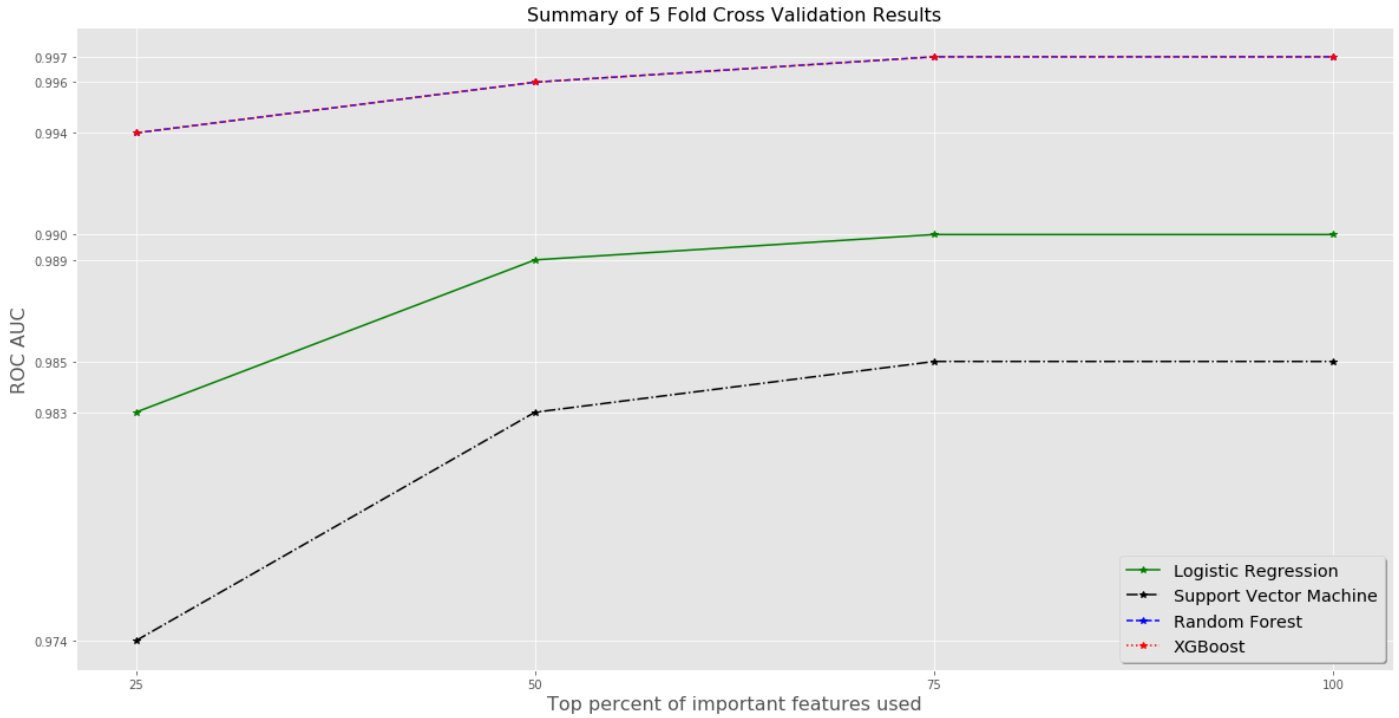

Figure 13: Summary of 5-fold cross validation results

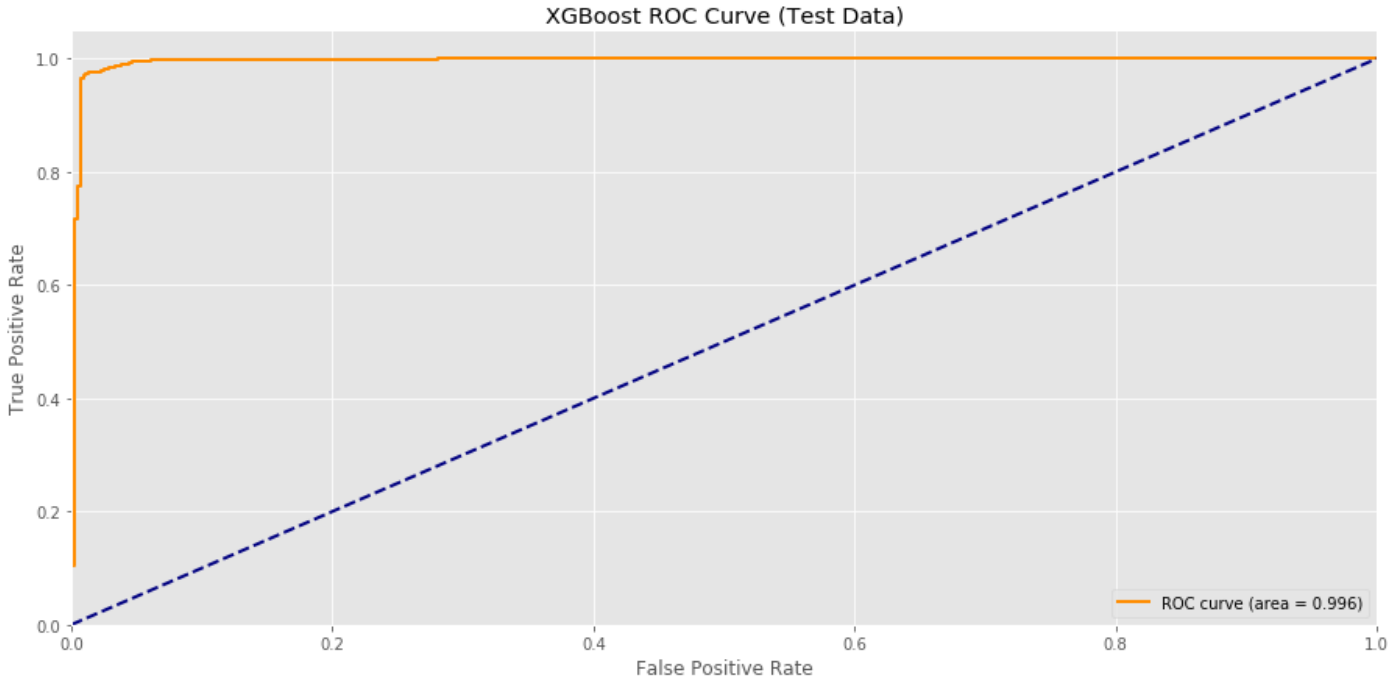

Figure 14: Receiver operating characteristic curve of XGBoost on test data