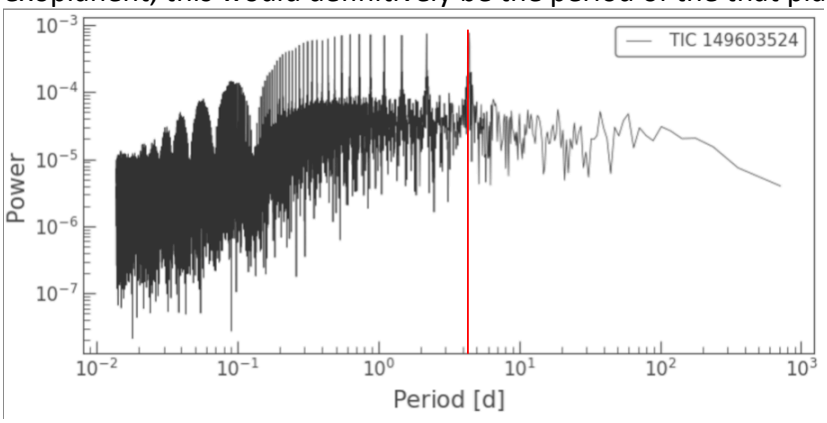
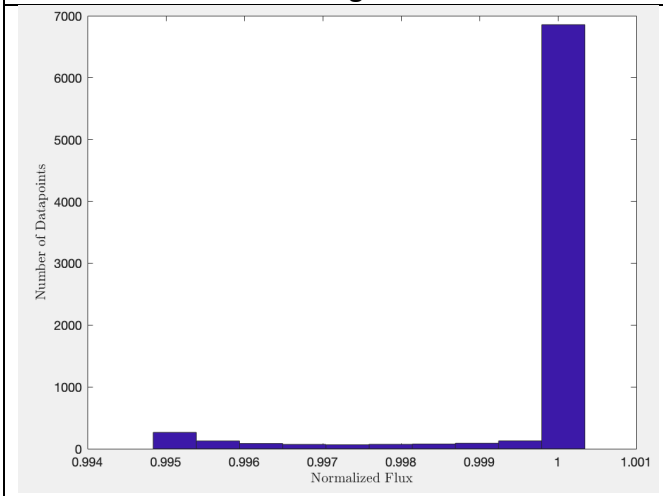
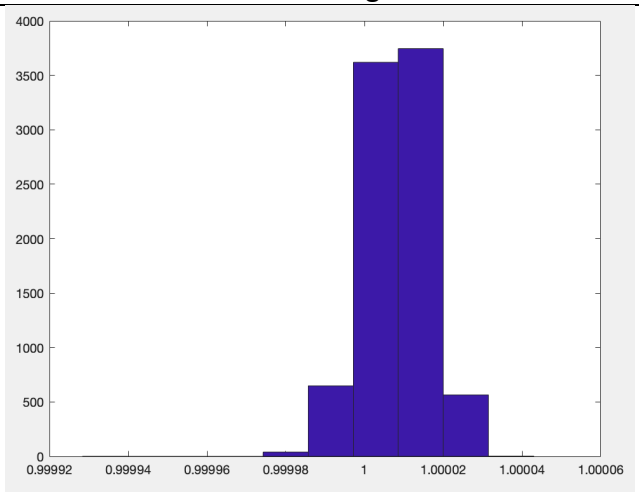
Initial Improvements:

Following the analysis performed in Assignment 6, the following improvements were made to the feature space in an attempt to improve classification performance:

- Periodograms were made for each light curve[1] with which the period with maximum power was determined (henceforth the Peak Power Period). For unoccluded solitary stars with a single exoplanent, this would definitively be the period of the that planet's orbit. Example:



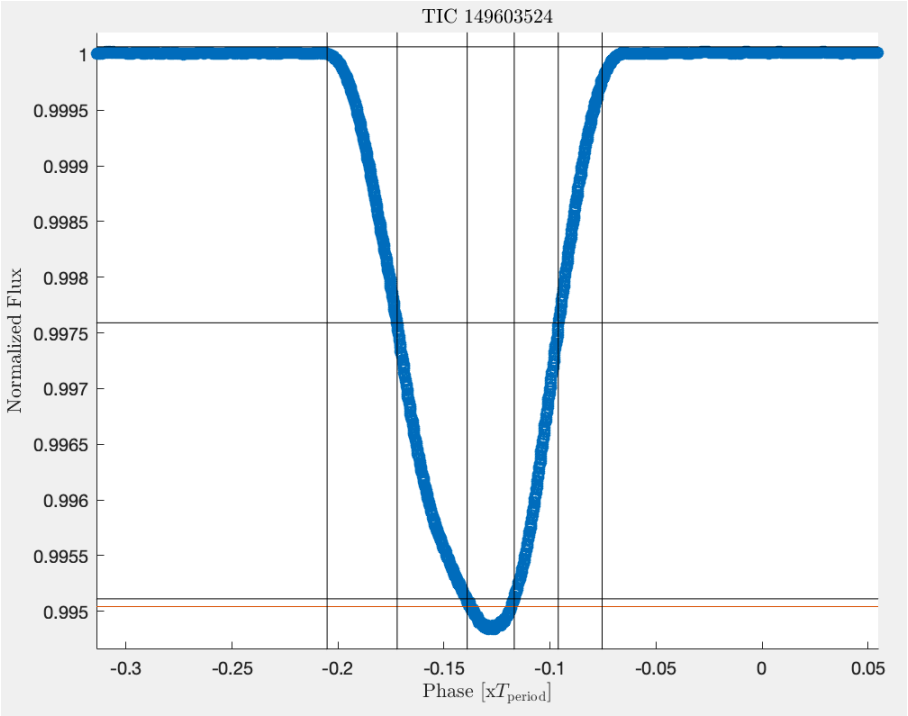
- The light curves were made for each object and folded about the identified Peak Power Period to produce an aggregate snapshot of what the event with the identified period looked like. From this, a histogram was produced of fluxes. If there were multiple peaks, it was assumed that a legitimate transit event had been observed. The histogram of a object which experienced a transit event on the left below shows two peaks: one at the baseline value (1) and the other at the max transit depth; whereas, the one on the right which did not experience such an event shows a much tighter normal distribution about the baseline value (1).

Transiting event	Non-transiting event
	
TIC 149603524	TIC 349191590

For all transit curves were it was assumed that a transit event had been observed, the flux depth of the transit (TransitDepth), duration of the transit (TransitDuration), and average of the 100% rise and fall time (TransitEdgeTime) were computed. If it was not assumed that a transit event was actually observed, these values were set to zero. Below is a folded transit curve with the identified 0%, 50%,

[1] 34.5GB of data!

and 100% drop times and thresholds which were used to determine these features displayed.



- An addition feature was also derived from the folded transit curve data: PlanetRadius. Namely, this is the what the expected radius (as a multiple of Earth's radius) of the planet orbiting the observed star would be if it were orbiting a solitary star and the only planet transiting at the time, with either an orbital inclination near zero or negligible limb-darkening on the star. In this case, since the peak transit depth is proportional to the area of the star covered by the planet:

$$\Delta F_{depth} = \frac{A_{planet}}{A_*} = \left(\frac{R_p}{R_*}\right)^2 \rightarrow R_p = R_* \sqrt{\Delta F_{depth}} * \frac{R_\odot}{R_\oplus}$$

- The suggested "StarGap" feature which separates Red Dwarfs from Main Sequence stars was included.

$$StarGap = \left|T_{eff} - 4000K\right|$$

- MAD outlier removal was adjusted to ignore instances with default values for the parameters derived from the folded transit curves before determining the expected distribution.

With the Multilayer Perceptron model created in the previous assignment, the best model correctly classified **91.59% of instances** (in an unevenly distributed dataset) and achieved a **kappa of 0.7184** on the dataset used in that assignment.

New Baseline:

The best model from the previous assignment along with the a couple other potentially well performing models were tested on this new dataset. The best performing of these will serve as the baseline and will be further developed.

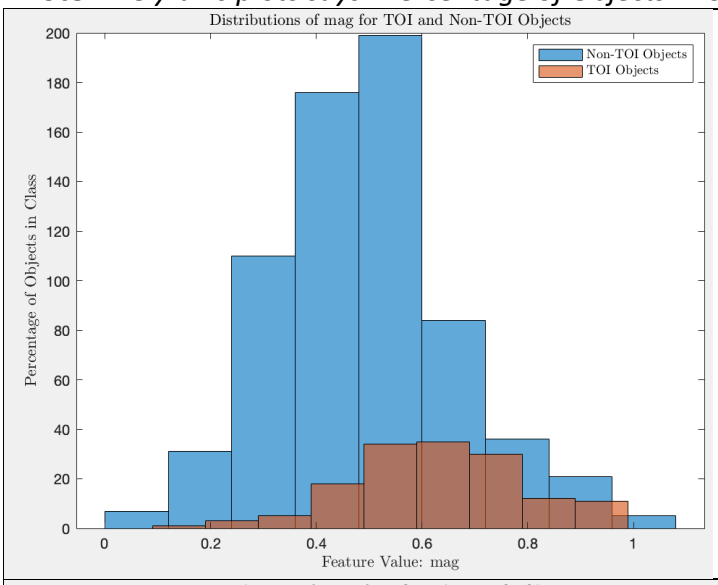
	Percent Correct	Kappa
Multilayer Perceptron (previous assignment but now with 5000 epochs)	**94.8718%**	**0.7795**
Multilayer Perceptron	87.36%	0.1264

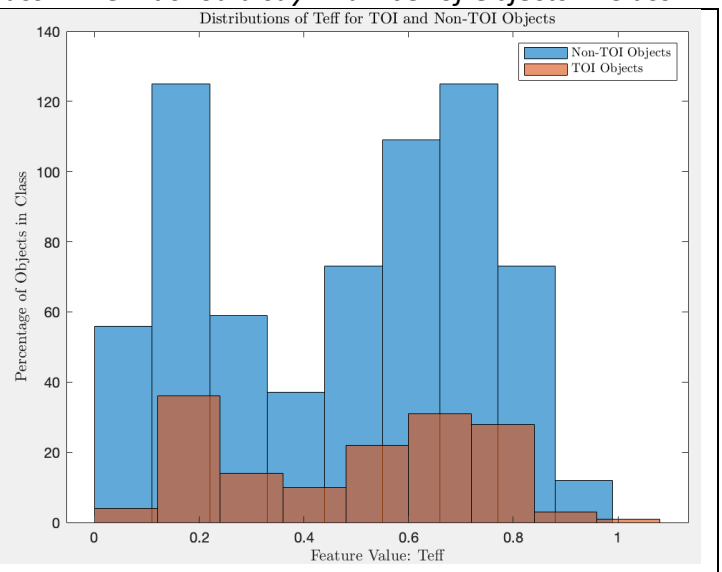
with second 4 layer hidden layer, 5000 epochs.		
LibSVM with PolyKernel	84.43%	0.00
J48 Decision Tree	87.73%	0.48

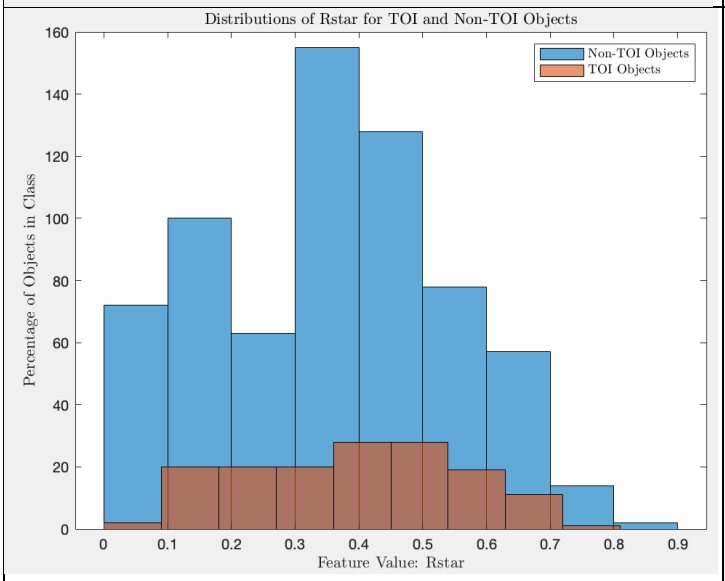
Initial Observations:

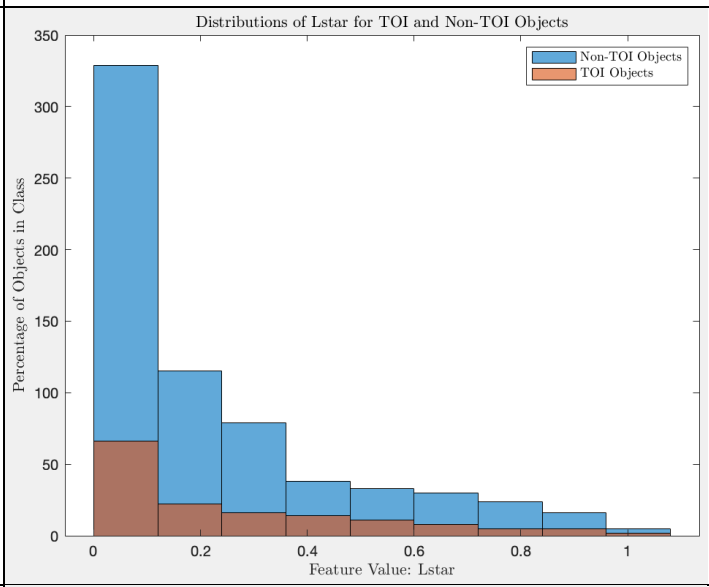
A MATLAB script was written to help with initial observations by plotting the distributions within each feature of the two class values, TOI and non-TOI, producing the following output:

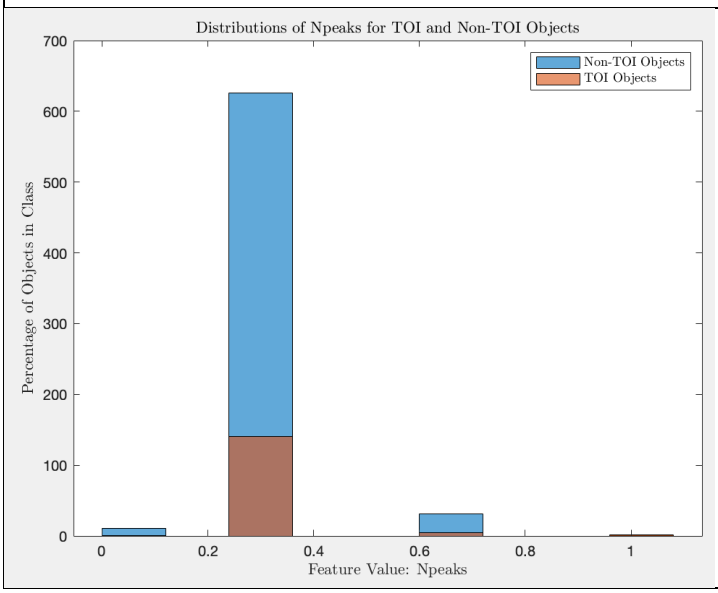
***Note**: *The y-axis plots says "Percentage of Objects in Class" when it should say "Number of Objects in Class".*

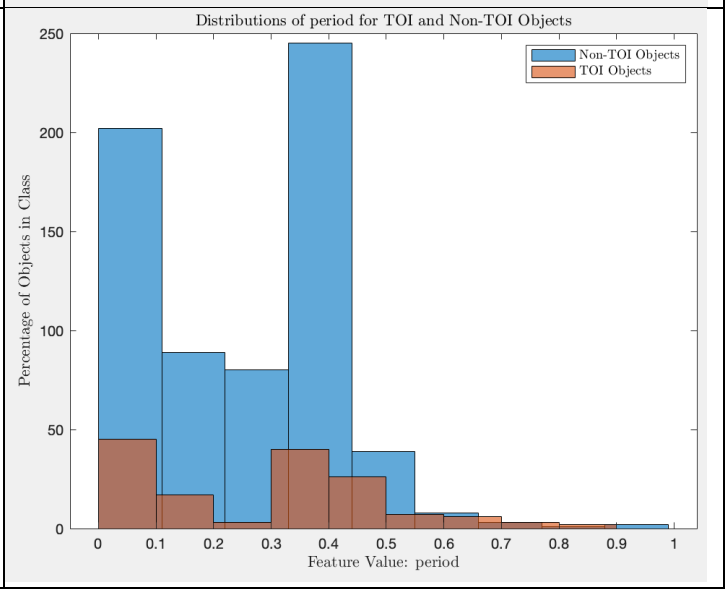


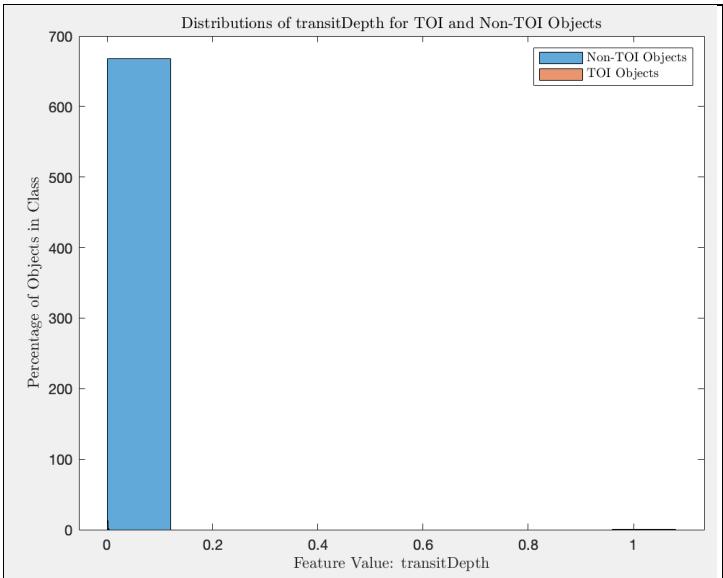


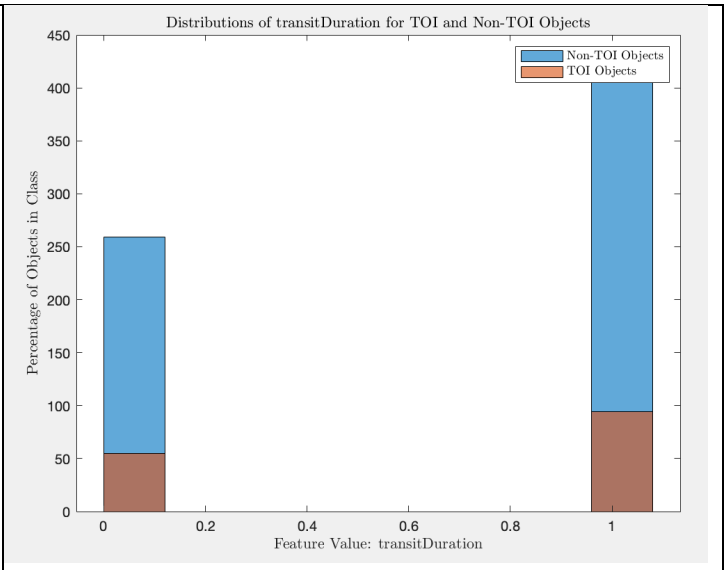


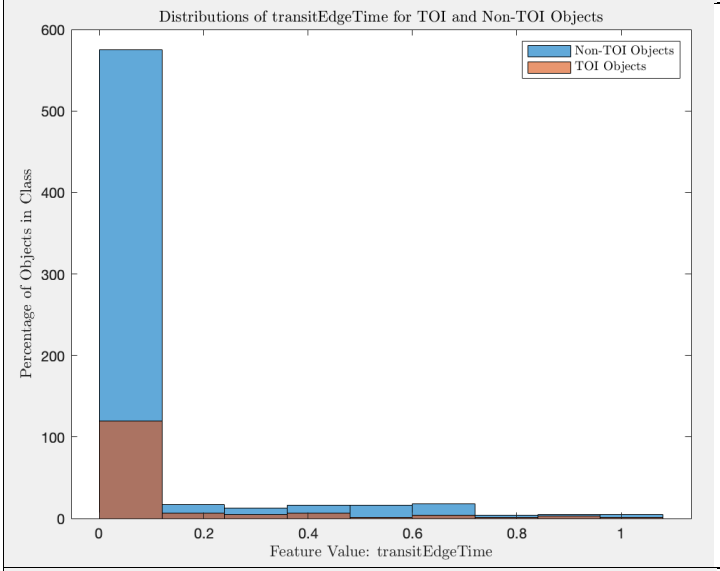


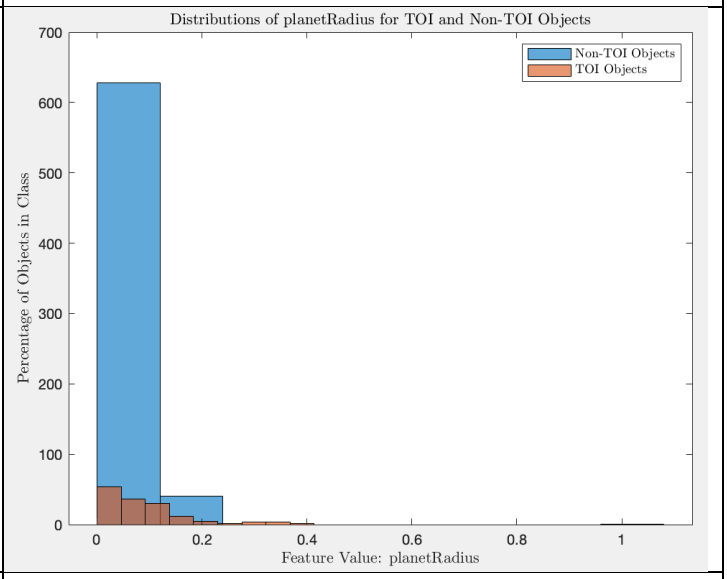


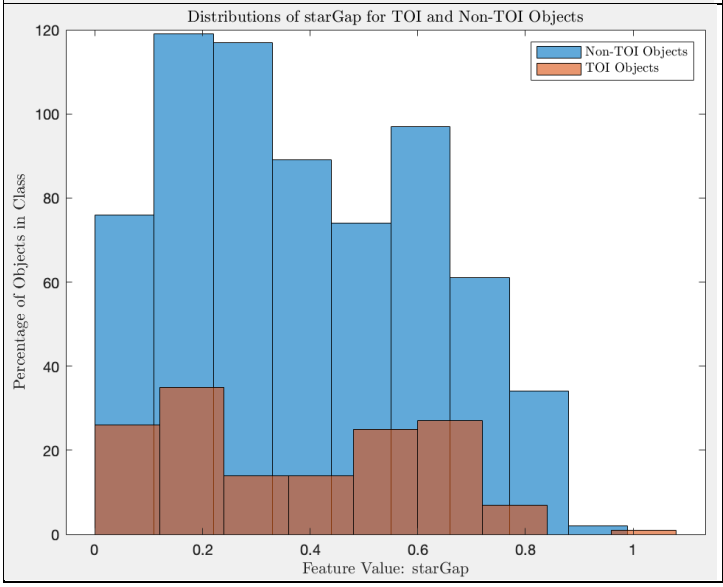



Distributions of transitDepth for TOI and Non-TOI Objects
Non-TOI Objects
TOI Objects
Percentage of Objects in Class
Feature Value: transitDepth


Distributions of transitDuration for TOI and Non-TOI Objects
Non-TOI Objects
TOI Objects
Percentage of Objects in Class
Feature Value: transitDuration


Distributions of transitEdgeTime for TOI and Non-TOI Objects
Non-TOI Objects
TOI Objects
Percentage of Objects in Class
Feature Value: transitEdgeTime

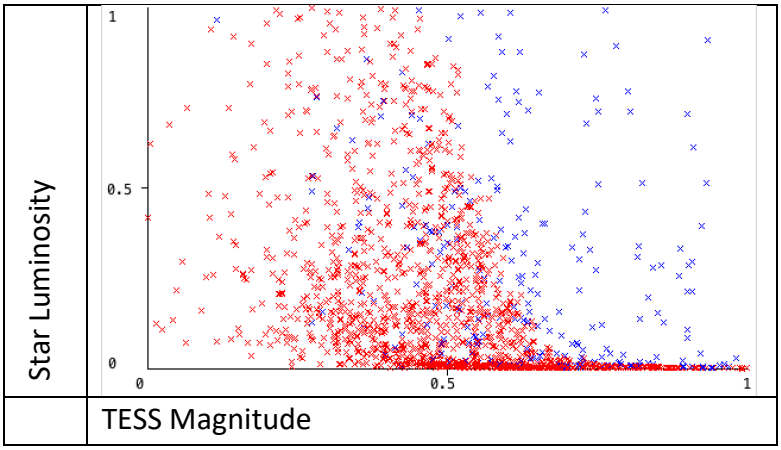

Distributions of planetRadius for TOI and Non-TOI Objects
Non-TOI Objects
TOI Objects
Percentage of Objects in Class
Feature Value: planetRadius

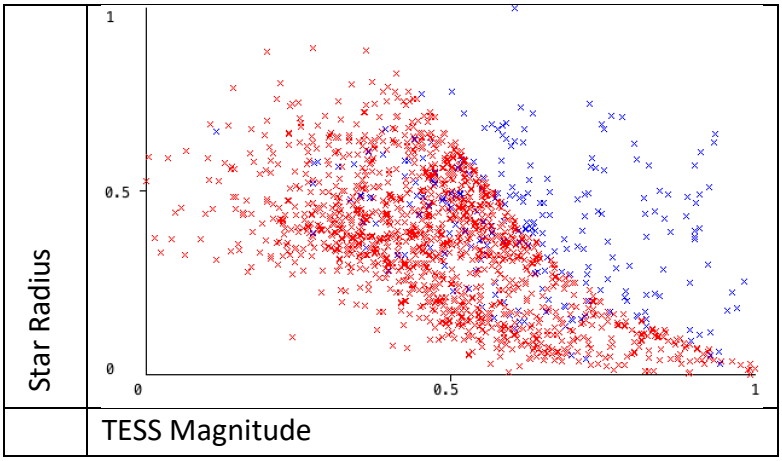

Distributions of starGap for TOI and Non-TOI Objects
Non-TOI Objects
TOI Objects
Percentage of Objects in Class
Feature Value: starGap

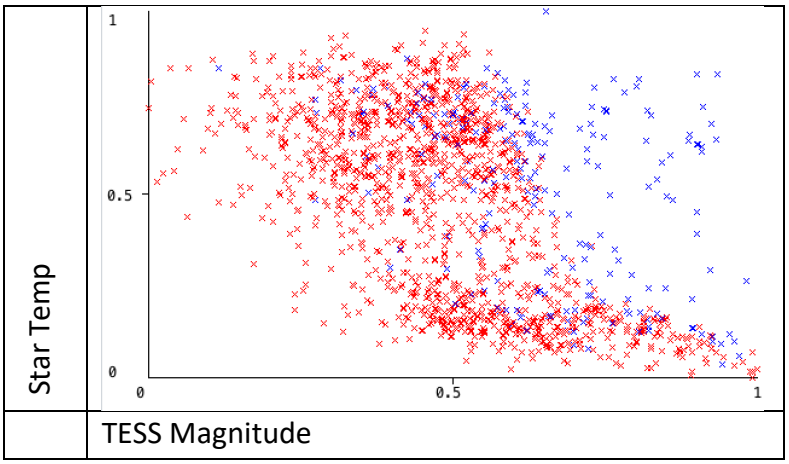
```
The avg. mag for Non-TOI objects was 0.491 but 0.627 for TOI objects (percent difference: 24.3%)
The avg. Teff for Non-TOI objects was 0.472 but 0.487 for TOI objects (percent difference: 3.2%)
The avg. Rstar for Non-TOI objects was 0.355 but 0.387 for TOI objects (percent difference: 8.6%)
The avg. Lstar for Non-TOI objects was 0.214 but 0.255 for TOI objects (percent difference: 17.4%)
The avg. Npeaks for Non-TOI objects was 0.344 but 0.351 for TOI objects (percent difference: 2.0%)
The avg. period for Non-TOI objects was 0.247 but 0.280 for TOI objects (percent difference: 12.4%)
The avg. transitDepth for Non-TOI objects was 0.002 but 0.000 for TOI objects (percent difference: 123.8%)
The avg. transitDuration for Non-TOI objects was 0.611 but 0.628 for TOI objects (percent difference: 2.8%)
The avg. transitEdgeTime for Non-TOI objects was 0.075 but 0.097 for TOI objects (percent difference: 26.1%)
The avg. planetRadius for Non-TOI objects was 0.056 but 0.086 for TOI objects (percent difference: 42.5%)
The avg. starGap for Non-TOI objects was 0.385 but 0.371 for TOI objects (percent difference: 3.8%)
```

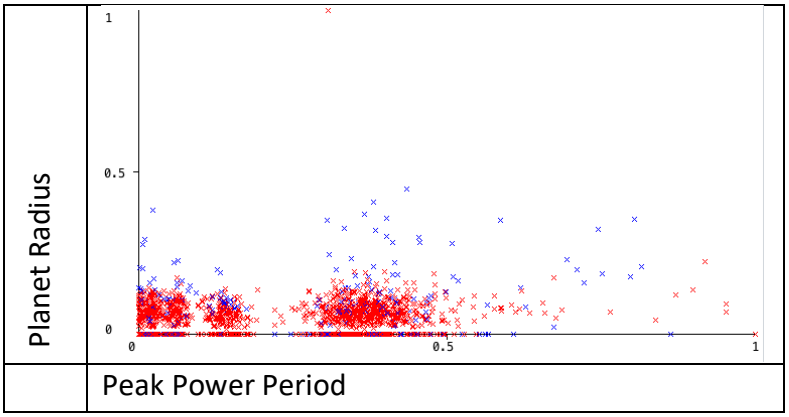
The most clear observation that comes from these data is that there is no feature which contains a clear partition between the two classes within itself. If possible, it would be great to add more features (such as transit curve goodness-of-fit) which have such a distinction. That said, from this early analysis, the very distinguished difference in the means for the newly added TransitDepth feature indicates that it may serve as a useful feature for later partitioning (perhaps through a clustering algorithm).

To gain further insight into whether there exist obvious partitions between the two classes in the feature space, the Weka Visualize workspace was used. Distinct partitions between the TOI and Non-TOI classes were observed in the following 2D slices of the feature space (where blue is the TOI class and red is the Non-TOI class). Many of these observations are similar to those observed in the previous assignment (at least for the relations between features which existed in that iteration) but are greatly ameliorated by the significant increase in the number of processed instances in the dataset. One notable observation as a result of this increase is the quite obvious hard border on the left side the of the Star-Radius v. Magnitude distribution within the Non-TOI class but the lack of such a well-defined border in the TOI class.

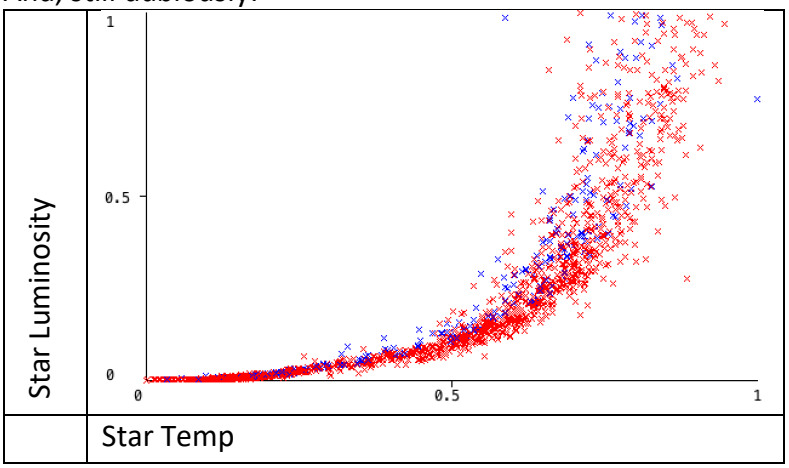






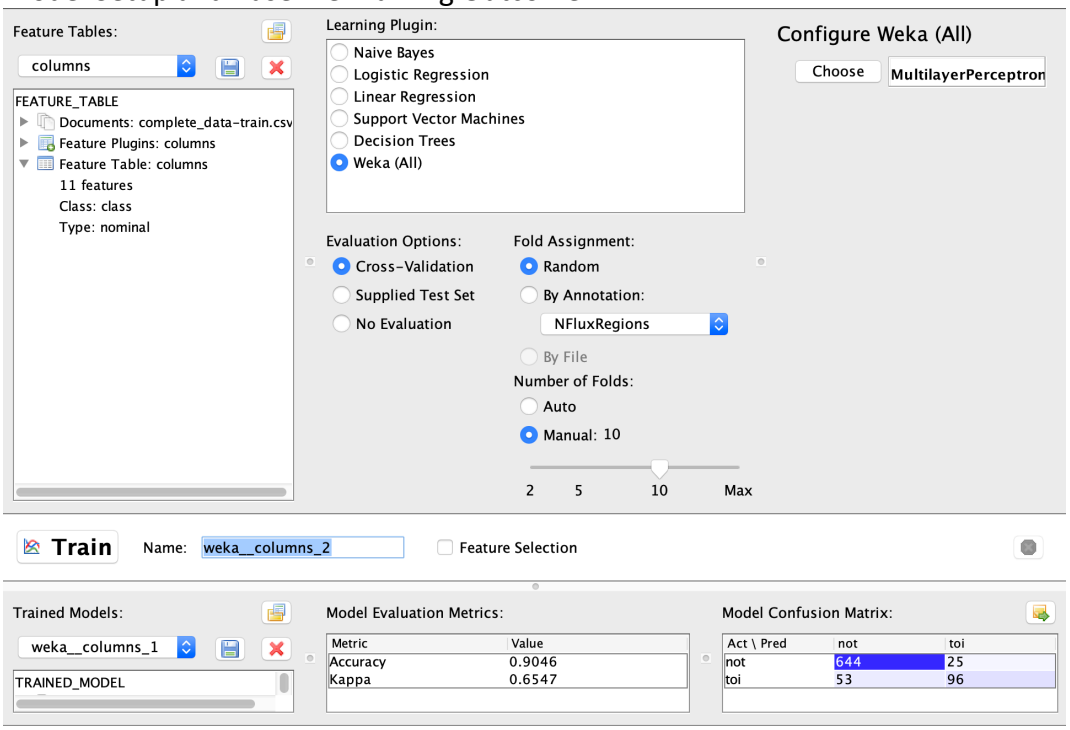


And, still dubiously:



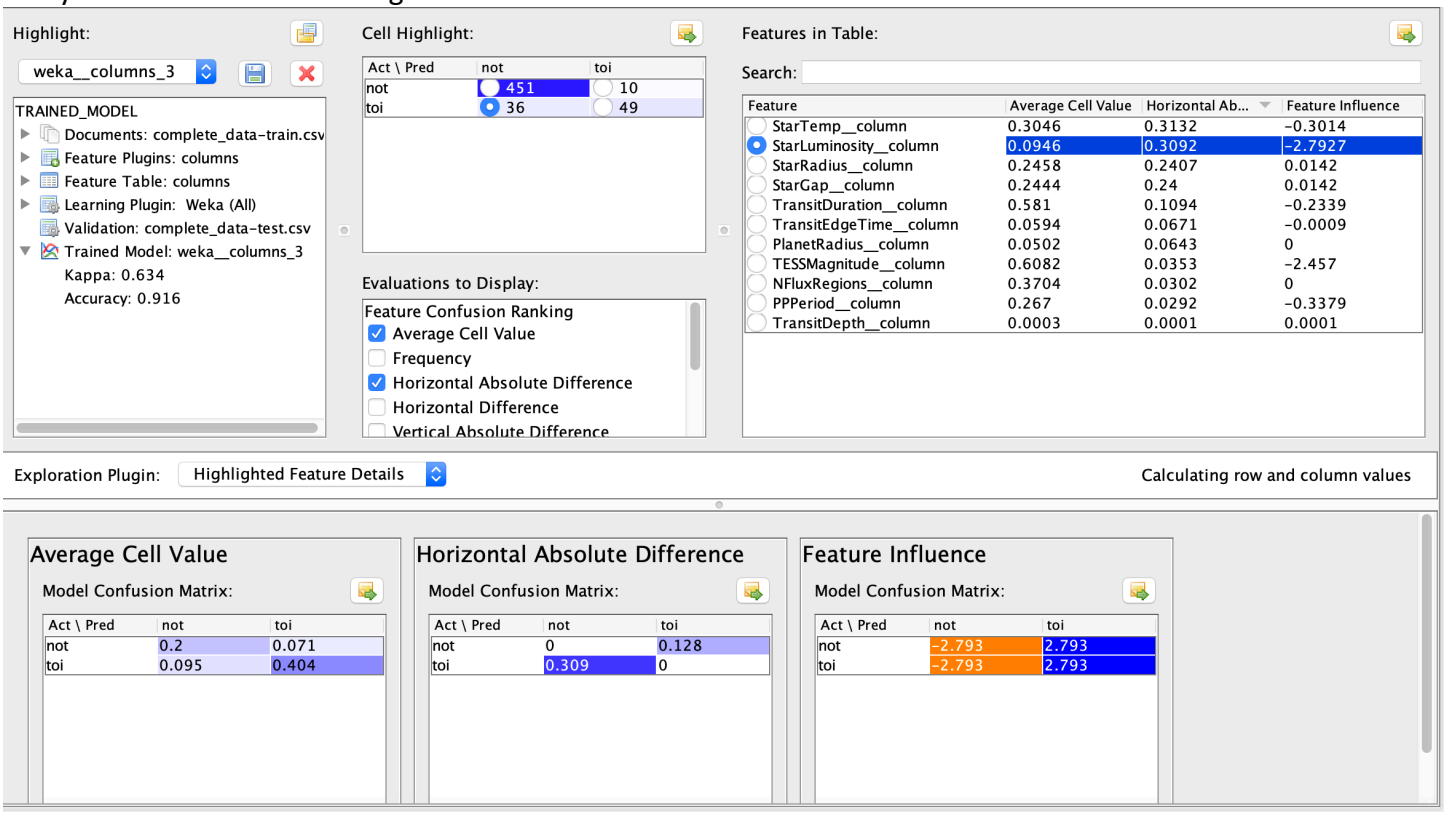
Error Analysis:

Model Setup and Baseline Training Outcome:



Horizontal Comparisons:

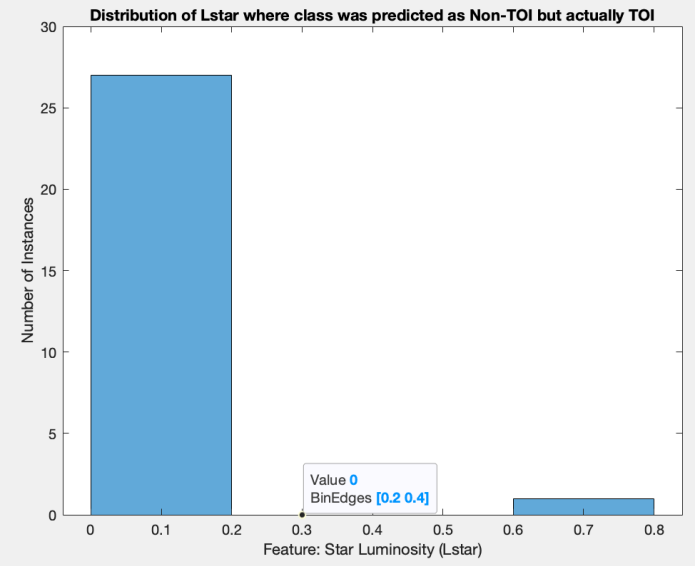
The error cell in which class was predicted to be Non-TOI but was actually TOI was chosen to perform error analysis on since it has the largest value of the two error cells.

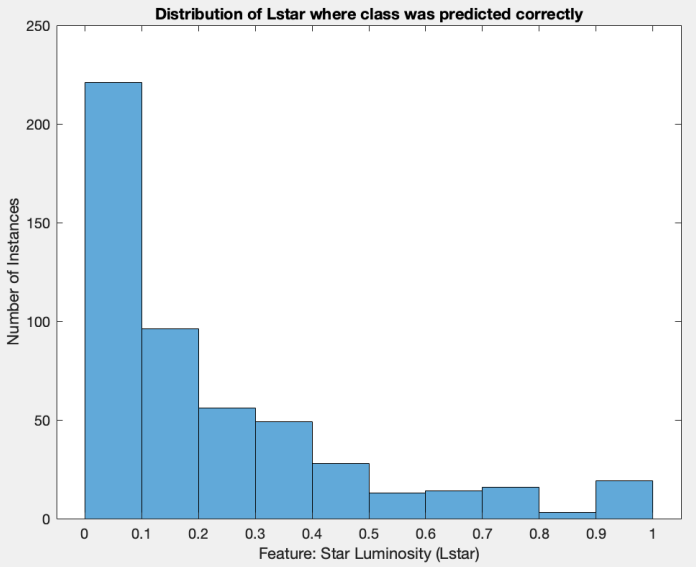


The first problematic feature chosen was the "StarLuminosity" column since it had one of the highest horizontal absolute differences and also had a large feature influence magnitude (feature weight wasn't an option), and non-zero average cell value.

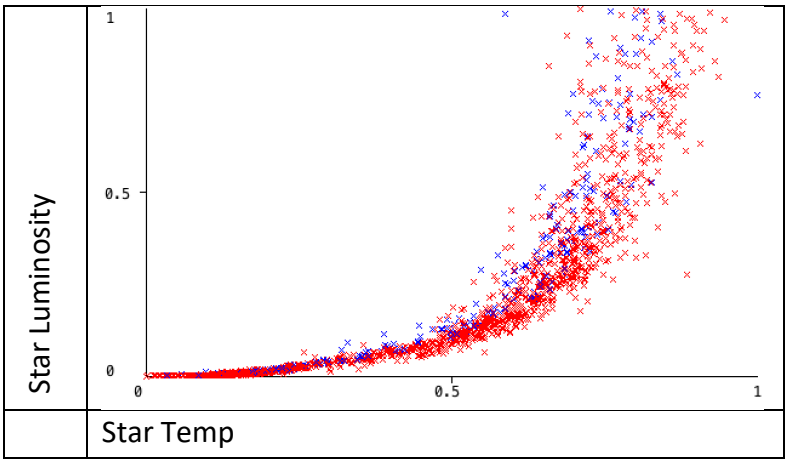
The selected data shows that the model increases the likelihood of TOI as StarLuminosity increases and that lots of the instances which are actually classified as TOI, get classified as Not-TOI.

To gain further insight on this, the dataset including the predicted labels were exported from the Predict Labels pane. Feeding this data into a basic MATLAB helper script produces the attached histograms which show that for all instances where class was predicted to be Not-TOI but actually was TOI, "Lstar" was, in the vast majority of cases, less than 0.2. Unfortunately, this also appears to loosely mirror the distribution for correctly classified instances, most of which are below 0.2.





However, knowing that this feature is especially problematic for low luminosities, in combination with the distinct but oddly shaped border observed between the classes in the Lstar vs Teff plot in the initial observations:



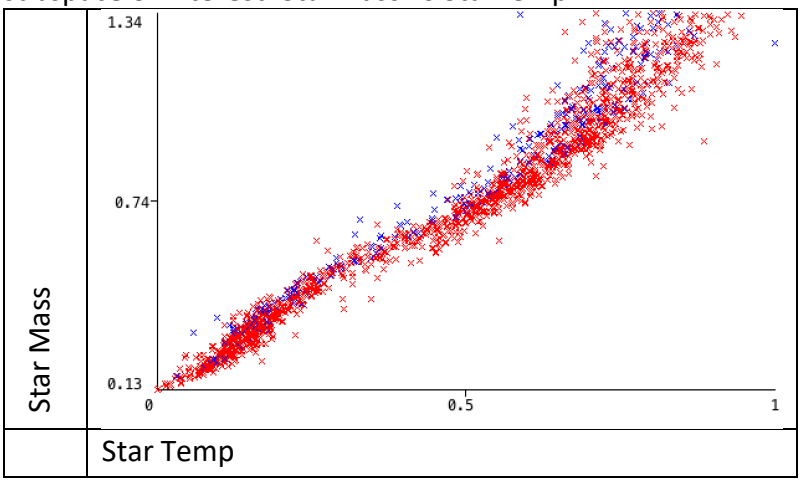
The true distinction could lie in the stellar mass which is known to be $\frac{M_*}{M_\odot} \approx \left(\frac{L_*}{L_\odot}\right)^{\frac{1}{3.5}}$ for main-sequence stars.

The addition of such a feature would provide more contrast between the classes at lower stellar luminosities.

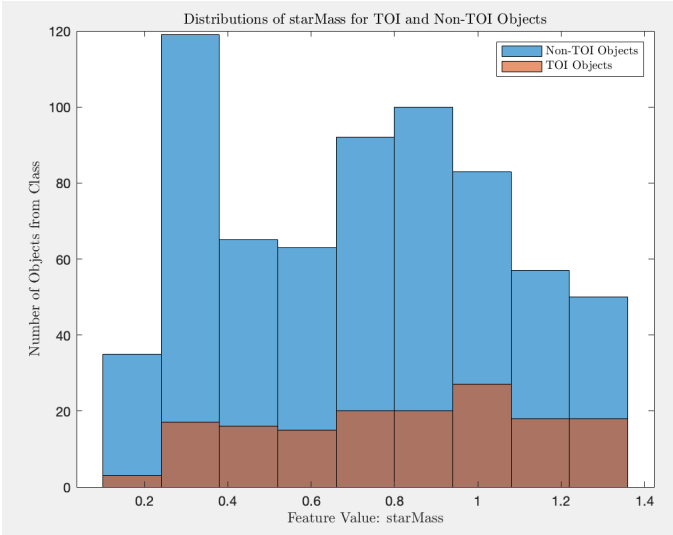
Model Improvement:

The **StarMass** parameter proposed in the horizontal comparison error analysis was added to the feature space and the model was retrained.

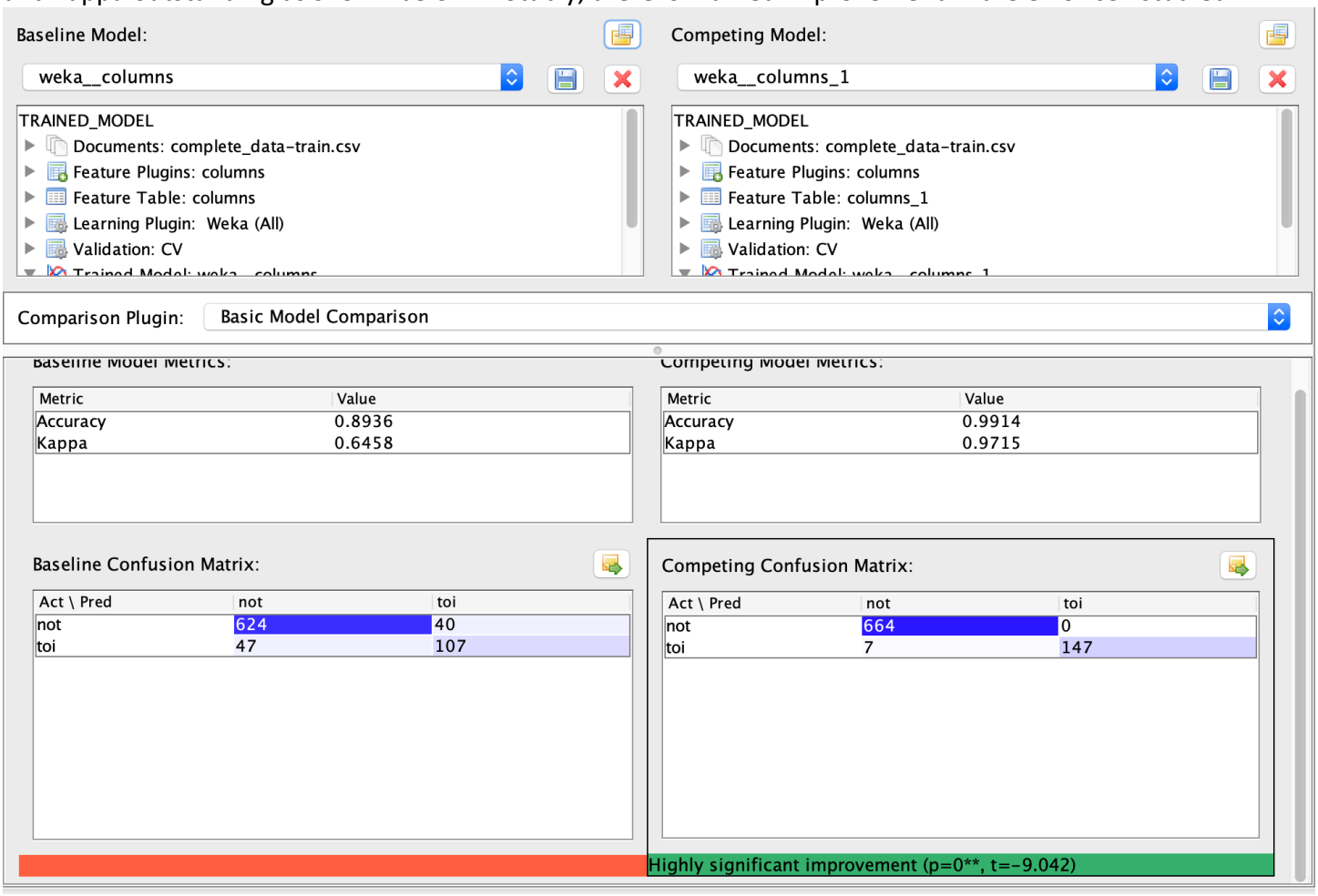
In Weka Visualize, the addition of this feature appears to have created a clearer distinction within the 2D subspace of interest: StarMass vs StarTemp:



In addition, as shown below, there are no particularly salient differences in the distributions of StarMass for TOI and Non-TOI Objects, aside from that the StarMass distribution of TOI objects appears to be shifted slightly to the higher mass end.



In comparison to the baseline, this model offers a highly significant improvement, making both correctness and kappa outstanding as shown below. Notably, there is marked improvement in the error cell studied.



Feature Space Revisions

Much of the delay in this assignment has been in attempting to fit an analytical model of a transit curve to the folded transit curve which would allow for a more precise TransitDuration parameter as well as the addition of a "Goodness-of-fit" parameter. The difficultly has simply been in implementing the model (Mandel'02, shown below) in such a way that it can be efficiently performed over the ~6GB of folded transit curves.

$$F(p,\ z) = \left[\int_0^1 dr\ 2rI(r)\right]^{-1} \int_0^1 dr\ I(r)\frac{d[F^e(p/r,\ z/r)r^2]}{dr}$$

where:

$F^e(p,z) = 1 - \lambda^e(p,z)$

$$\lambda^e(p,\ z) = \begin{cases} 0, & 1+p<z, \\ \frac{1}{\pi}\left[p^2\kappa_0 + \kappa_1 - \sqrt{\frac{4z^2-(1+z^2-p^2)^2}{4}}\right], & |1-p|<z\le 1+p, \\ p^2, & z\le 1-p, \\ 1, & z\le p-1, \end{cases}$$

$\kappa_1 = \cos^{-1}[(1-p^2+z^2)/2z], \kappa_0 = \cos^{-1}[(p^2+z^2-1)/2pz]$

$I(r) = 1 - \gamma_1(1-\mu) - \gamma_2(1-\mu)^2$, where $\gamma_1 + \gamma_2 < 1$ [2]

Identified Flaws for Next Revision:

- Folded Transit Curve "Goodness-of-fit" (R^2)
- Evaluate adding sgCDPP Noise metric.
- Perform parameter tuning on hidden layer(s) structure. Currently using the default, unlikely this is actually the best.

[2] Mandel and Agol, 2002 https://iopscience.iop.org/article/10.1086/345520/fulltext/