Chandler Kinch

Final Report: Starcraft 2 League Analysis

Problem Statement

Starcraft 2 is a highly competitive Real Time Strategy(RTS) game published by Blizzard Entertainment in 2010. It is a game of resource management, unit manipulation and ever changing strategy. The goal of the game is to build an army from the ground up and use it to outsmart and out maneuver one's enemy to victory. Ever since it's release, millions of RTS lovers have strived to perfect their game and climb in rank on Starcraft 2's ladder system, myself included. This requires hours of gameplay, watching videos, reading strategies and perfecting skills. There are a lot of things to focus on when trying to improve, and my question is 'What matters the most when trying to improve at Starcraft 2?'. I also wanted to make a model that would predict league placement and another model for smurf detection.

The Data

To solve these problems, I explored data provided by a study done at Simon Fraser University in 2013. This data set represents 3,395 games and includes 20 data points for each game, including League Index. Other data points represented things such as Minimap usage, Hotkey usage, Actions per Minute(APM), types of units made, hours played, and data that pertains to the Perception Action Cycle. Understanding this cycle is key to understanding the data, so I will explain it here.

The Perception Action Cycle is visualized by Fig. 1. Each small black tick represents an individual action. The dark and bright yellow sections combine to make a PAC, the black portions are gaps between these PACs and the bright yellow is the latency between the gaps and the first action being taken. In other words, this latency is the time between moving the screen to a new location and taking the first action after this screen move. In Fig. 1 it can be seen that as League increases, so does the frequency of the PAC.

Fortunately for me, the dataset was for the most part cleaned by the group that published it. The only thing that was really out of place were the Age, Hours per Week and Total Hours variables for the 8 League columns were all question marks, signifying missing data. For now, these question marks were simply changed to Not a Number values, and will be handled when modelling. Luckily, these columns didn't seem important when modelling, so this will have little effect on our analysis.

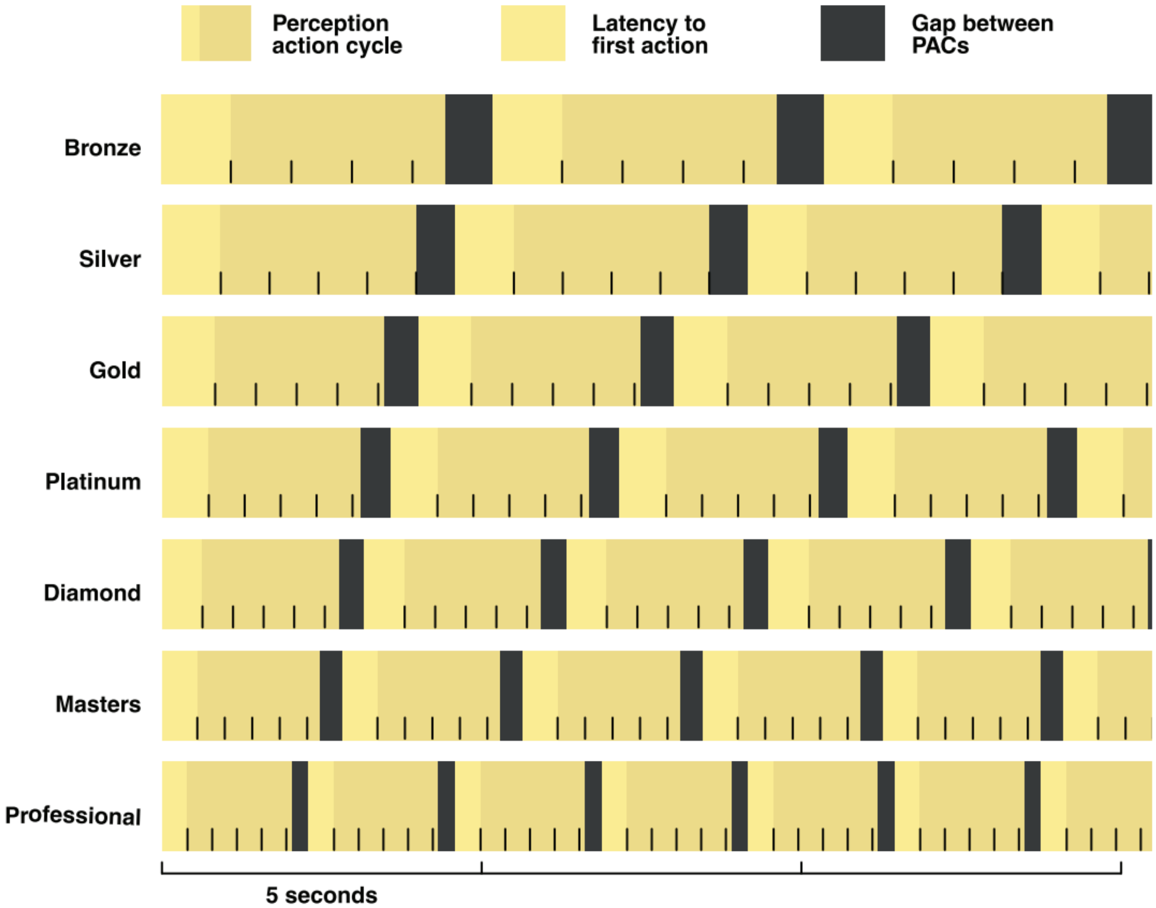

Fig 1. Perception Action Cycles (PACs), actions and attention shifts for a typical StarCraft 2 player, over 15 seconds. Each vertical line tic represents a single action. The y-axis represents league. Notice that most aspects of the PAC become faster with an increase in League[1]

[1] [1]This image comes from https://www.researchgate.net/figure/Perception-Action-Cycles-PACs-Actions-and-attention-shifts-for-a-typical-StarCraft-2_fig16_256932174

Exploratory Data Analysis

Let's begin by getting an idea of how our target variable, League Index, is distributed in Figure 1. Here the numbers 1-8 correspond to Bronze, Silver, Gold, Platinum, Diamond, Master, GrandMaster and Professional Leagues, respectively.

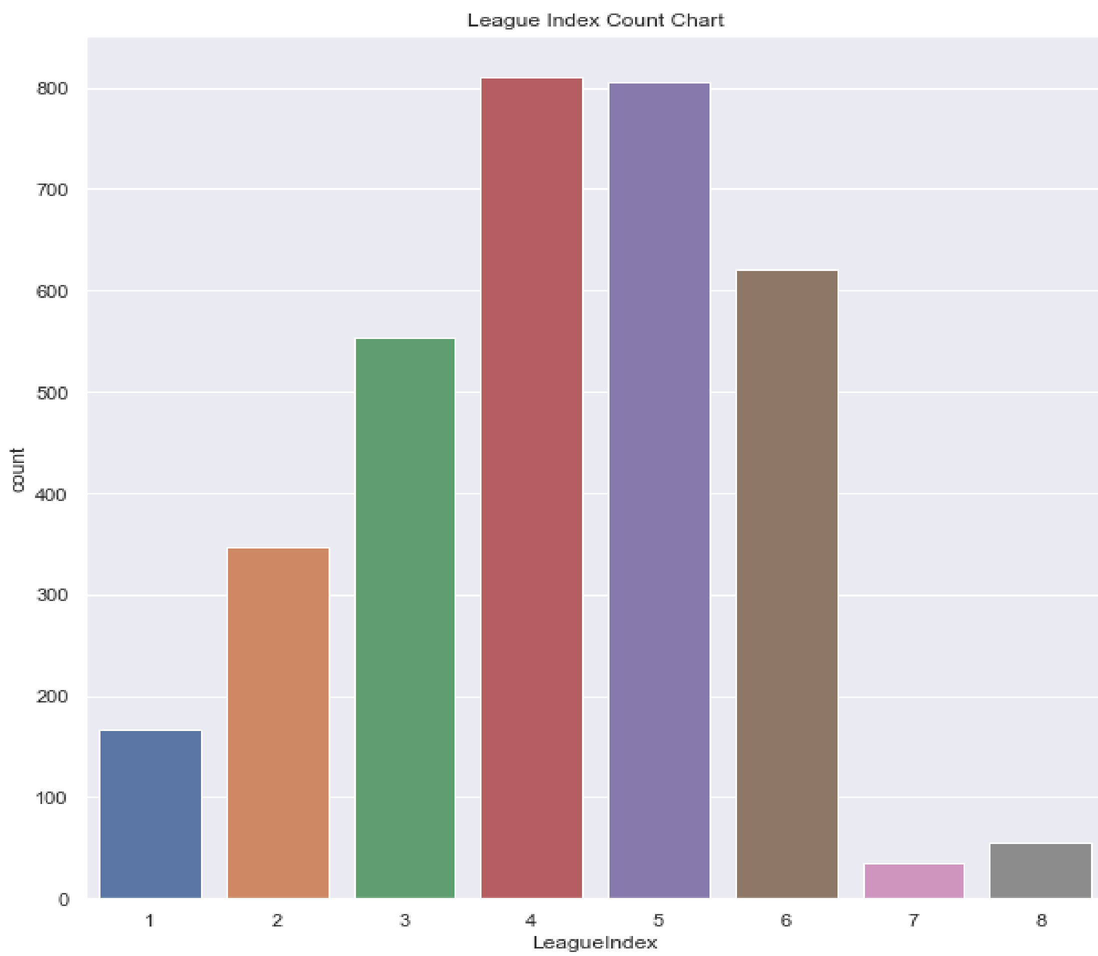

Figure 2 League Index Counts by League

We see that we have a decent spread of players from all the different leagues. The one place where the data struggles is with the GrandMaster and Professional leagues. This makes sense as the GrandMaster league is exclusive to the top 1000 players from each region, and professionals typically belong to this league. We will need to keep this in mind when doing our analysis.

Let's next look at a variable that causes mixed feelings in the Starcraft 2 community, and that is APM. APM causes mixed feelings in the Starcraft 2 community because it is a hard skill to acquire and even when acquired is difficult to use effectively. There are also a number of popular professionals that have relatively low APM and still do well in tournaments.

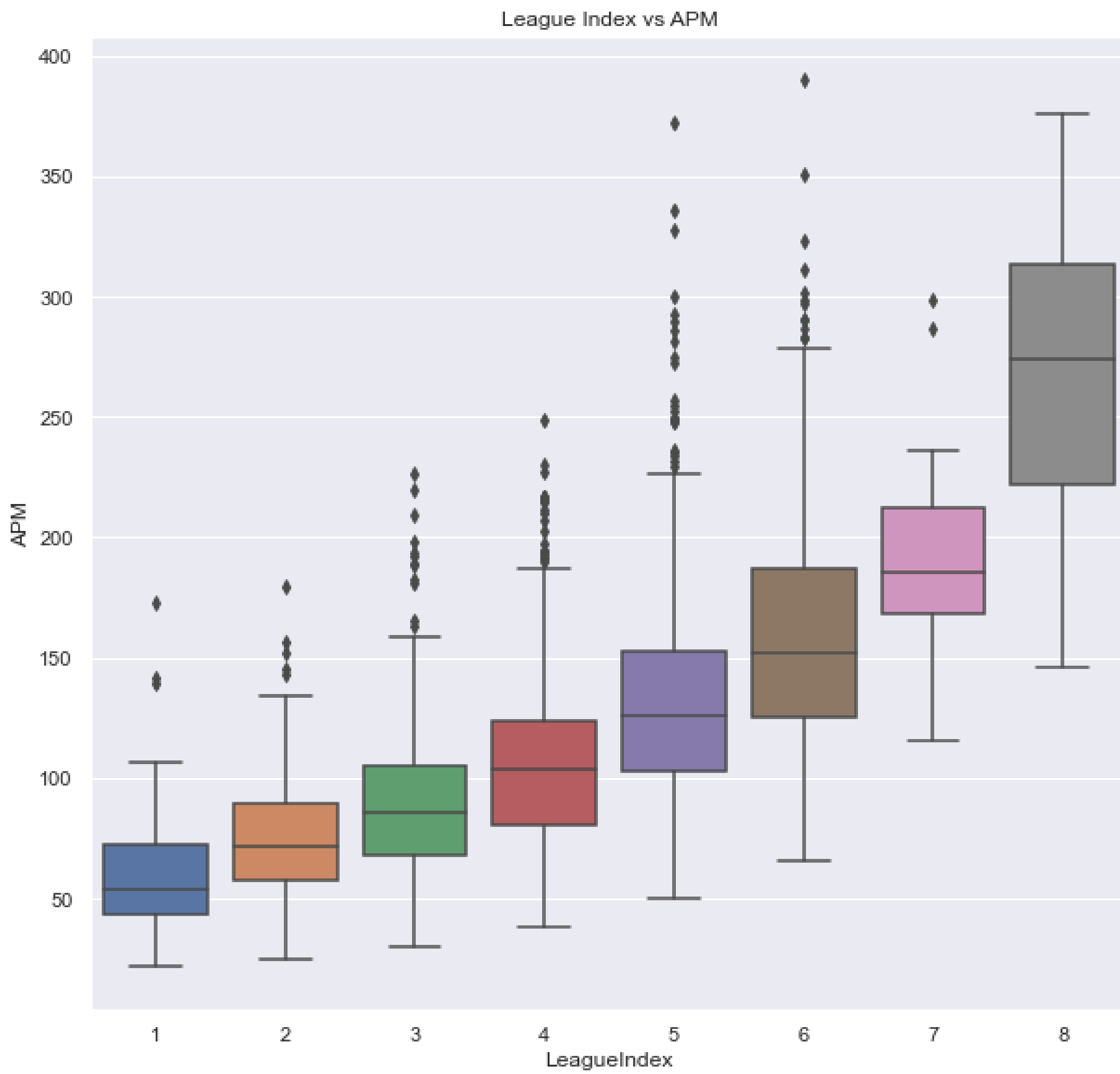

Fig 3. League Index vs Actions Per Minute

This boxplot shows the spread of APM values for each league. There is a steady and strong increase from bronze league to the professional league. As we can see, it is true that it is possible to have high APM and still be stuck in a lower league. It also shows that one can still have a low APM and be in a high league. However this trend is still important and shows an obvious correlation. Since the number of data points for leagues 7 and 8 are so low, we want to check and make sure our results are significant and not due to chance. T-test result for this comparison gave a p-value of $3.31x10^{-10}$, which is pretty close to zero. This shows that this difference is data is not due to chance.

Next let’s get an idea of all of how all the data correlates with the rest of the data. We will do this by using a heatmap in Figure 4.

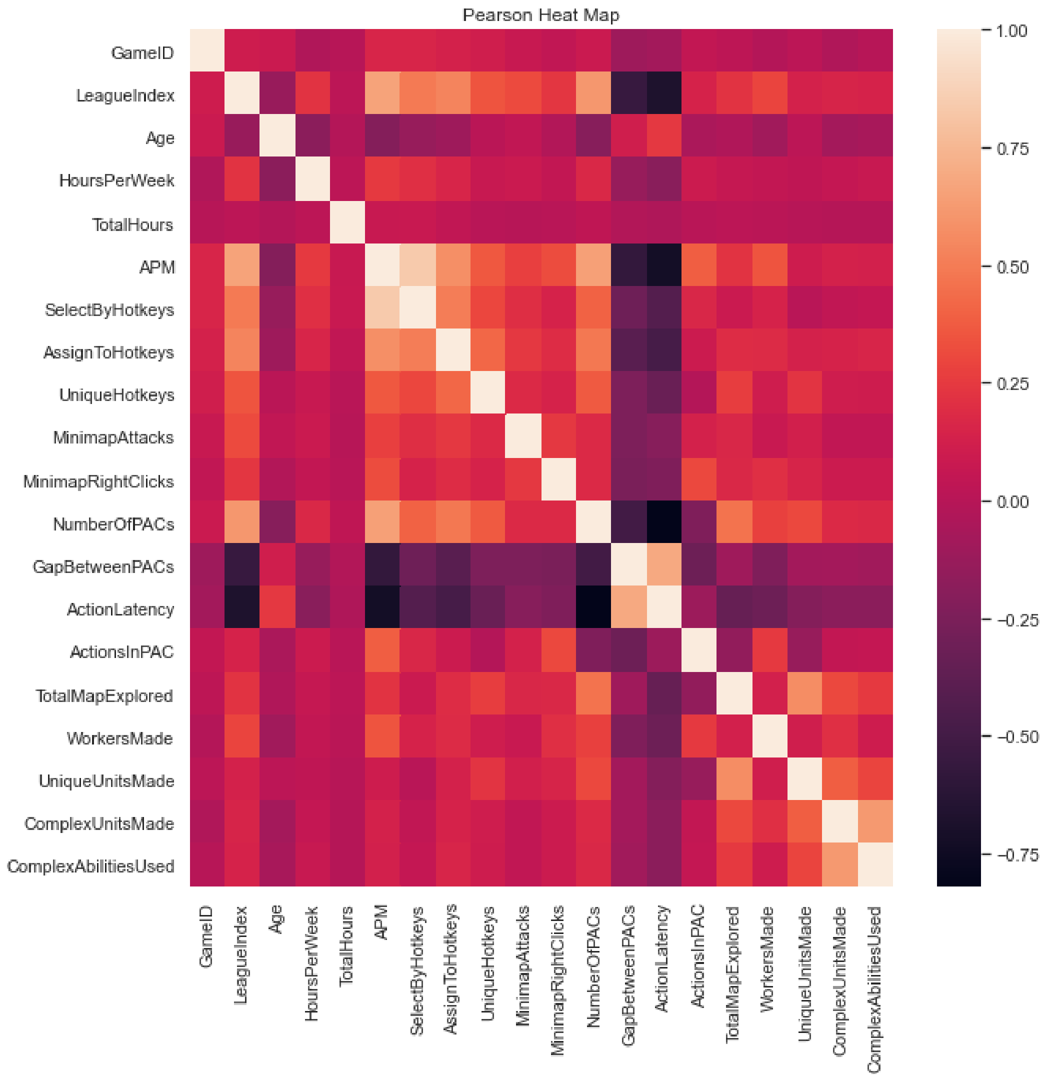

Figure 4

As was expected, APM shows a high correlation with League Index. Other variables that show a high positive correlation are Number of PACs and hotkey usage. Variables that have a strong negative correlation with League Index are Action Latency and Gaps Between PACs.

Feature Importance

Next, let's use a Random Forest Classifier to get an idea for what variables are most important for determining league index. Note, that this algorithm does not handle missing data. Data points were missing for Age, Total Hours and Hours per Week for all the professional league games. The averages for those variables were used to fill in these missing data. This is clearly an imperfect approach as professional players would likely have much higher than the average hours/week for example. However, the total missing data only represented <2% of the data and therefore shouldn't have a large effect on the feature importances shown here.

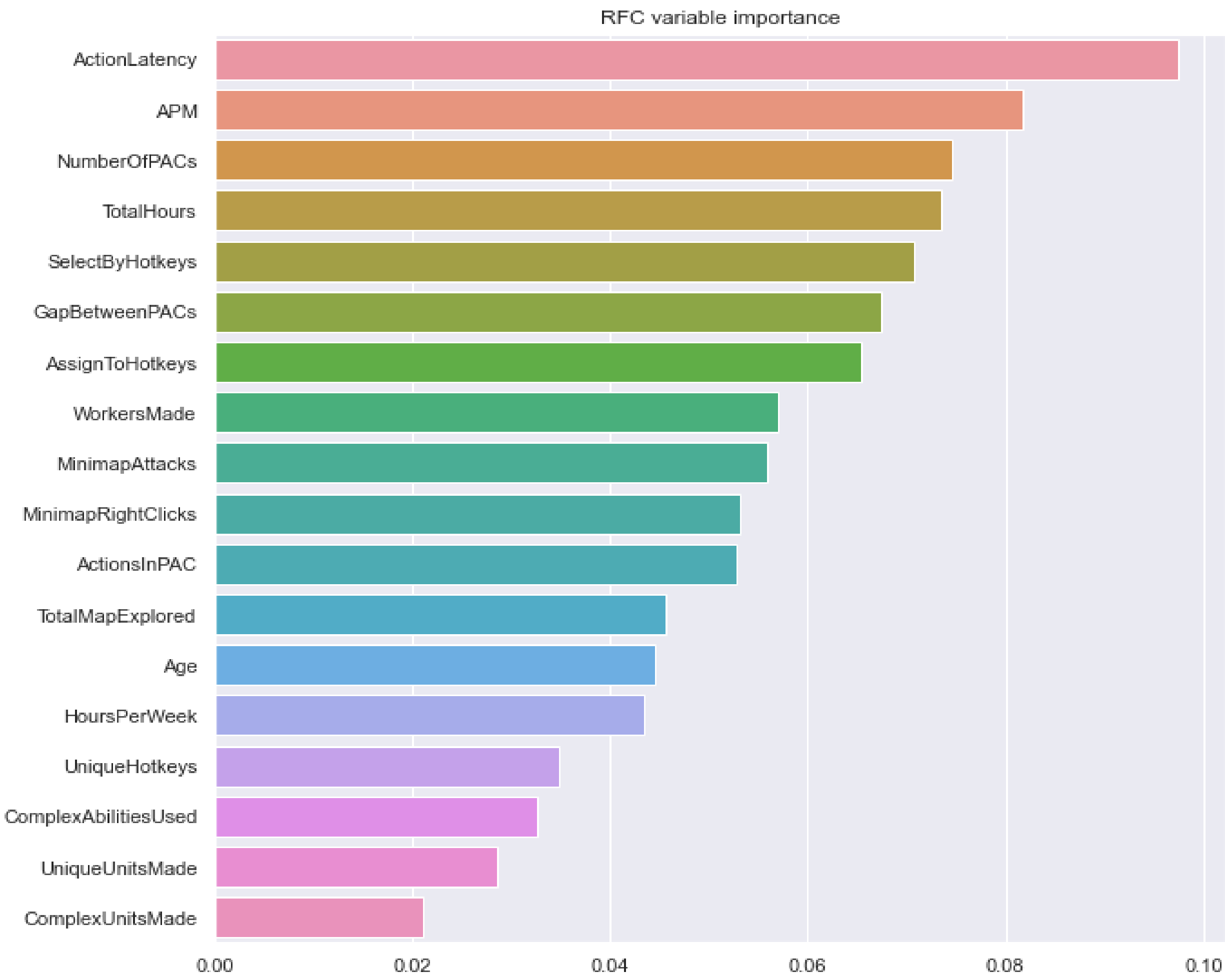

Figure 5

From this graph we can see that Action Latency is the most important feature when predicting League Index, followed closely behind by APM. While this is interesting and tells us a lot about how League Index is being predicted, this next graph gives us a better idea of how the leagues differ from their neighbors.

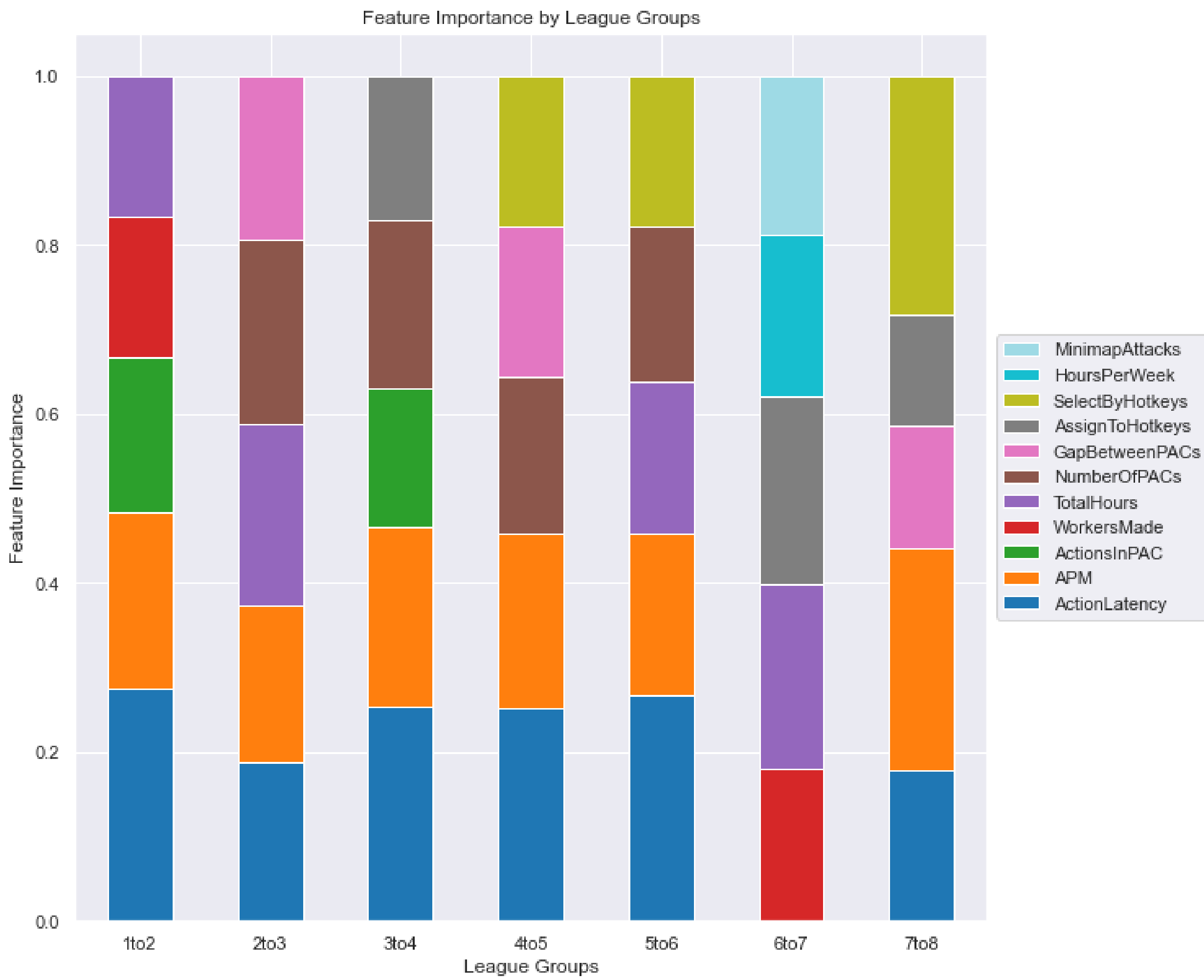

Figure 6[2]

While Fig 5. gave us a good idea of what the most important features are in general, there is an even more interesting question that might be asked - "For each league to league transition, what are the most important features?" For any individual player, trying to graduate from one league to the next, this is the pertinent question as to what skills they need to develop. I addressed this problem by building separate models for each league-to-league transition. For example, the 1 to 2 model used only the data for players in league 1 and 2 and was trained to predict which league they were in. After this, I ran the same feature importance approach used in Fig 5, chose the top 5 features for each transition and represented them in Fig 6. above.

[2] For the "7to8" model Age, Hours per Week, and Total Hours were dropped as these data were missing for the professional leagues

Fig 6 shows us the importance of the top five most important features for predicting League between two different leagues. This chart can help someone pinpoint what technical skills they should work on when they are trying to get to the next level. A few things to note:

- Action Latency and APM appear to remain top features across all league transitions with the exception of the transition from league 6 to league 7. Given the small data in league 7, it's possible that these feature importances
- Actions in PAC appears only twice, once in the 1 to 2 column and again in the 3 to 4 column. Likewise, Workers Made appears only twice. Once in the 1 to 2 column and again in the 6 to 7 column.
- Minimap Attacks and Hours per Week only appear in the 6 to 7 column. This column also seems to be the most unique, as it has these two features that appear only once, one that appears only twice and one that appears only three times. The uniqueness of this column is most likely due to the low number of data points for league 7.
- All variables that deal with the PAC cycle appear in this chart. Only the 6 to 7 column lacks at least one of these variables. All other columns have at least one if not more.

Modeling

League Detection

When players first want to play Starcraft 2 in ranked ladder, they must go through a series of placement games. The purpose of these games is to find a league for the player. Typically they consist of five or ten games with varying difficulty in order to judge the player's skill and to what league they belong. This model hopes to create a structure so that this prediction can be made with good accuracy.

Three different models were built and grid searched for this league placement model. These models consisted of K Nearest Neighbors, Logistic Regression and Random Forest Modeling. The data was scaled using Robust Scaler for all of these models and all features were kept when modeling. The data were also split using sklearn's train-test-split function with 25% percent of the data being kept as test data.

A little more data wrangling was needed to get the data ready for the league placement model. First of all, players can't be placed in Grandmaster or Professional leagues, so they will be dropped. Fortunately, this will also handle the missing values in the professional league. Secondly, this model looks to place new players, so they will have no play time. Consequently, we can drop the hours played column for this reason. Lastly as far as I could tell, Blizzard doesn't track players' ages or their hours played each week. They were provided to the data set by people that submitted their replays. These columns were also dropped.

Each of the mentioned algorithms were trained and evaluated as league predictors, the target metric being accuracy. Since there were so many categories, I wanted the model to make as many accurate predictions as possible. The results were a little disappointing at first, but made more sense with a bit of analysis. The best model was the Logistic Regression with an accuracy of 41%(see Table 1). This seems quite low, but there are a lot of target categories. When evaluating the confusion matrix(Figure7), the low score becomes clearer. The model actually has an accuracy of 86% if accuracy is defined as +/-1 of the actual value. This might be considered an acceptable range as leagues don't differ by a ton, and as long as the player continues to play games their win loss ratio would quickly adjust them to the correct league.

Classifier	Accuracy Score	Hyperparameter Values
K Neighbors	0.36	N_neighbors = 41
Logistic Reg.	0.41	C = 0.046 Penalty = l2
Random Forest	0.39	Max_depth = 9 Max_features = auto n_estimators=500

Table 1 Accuracy scores for each classifier

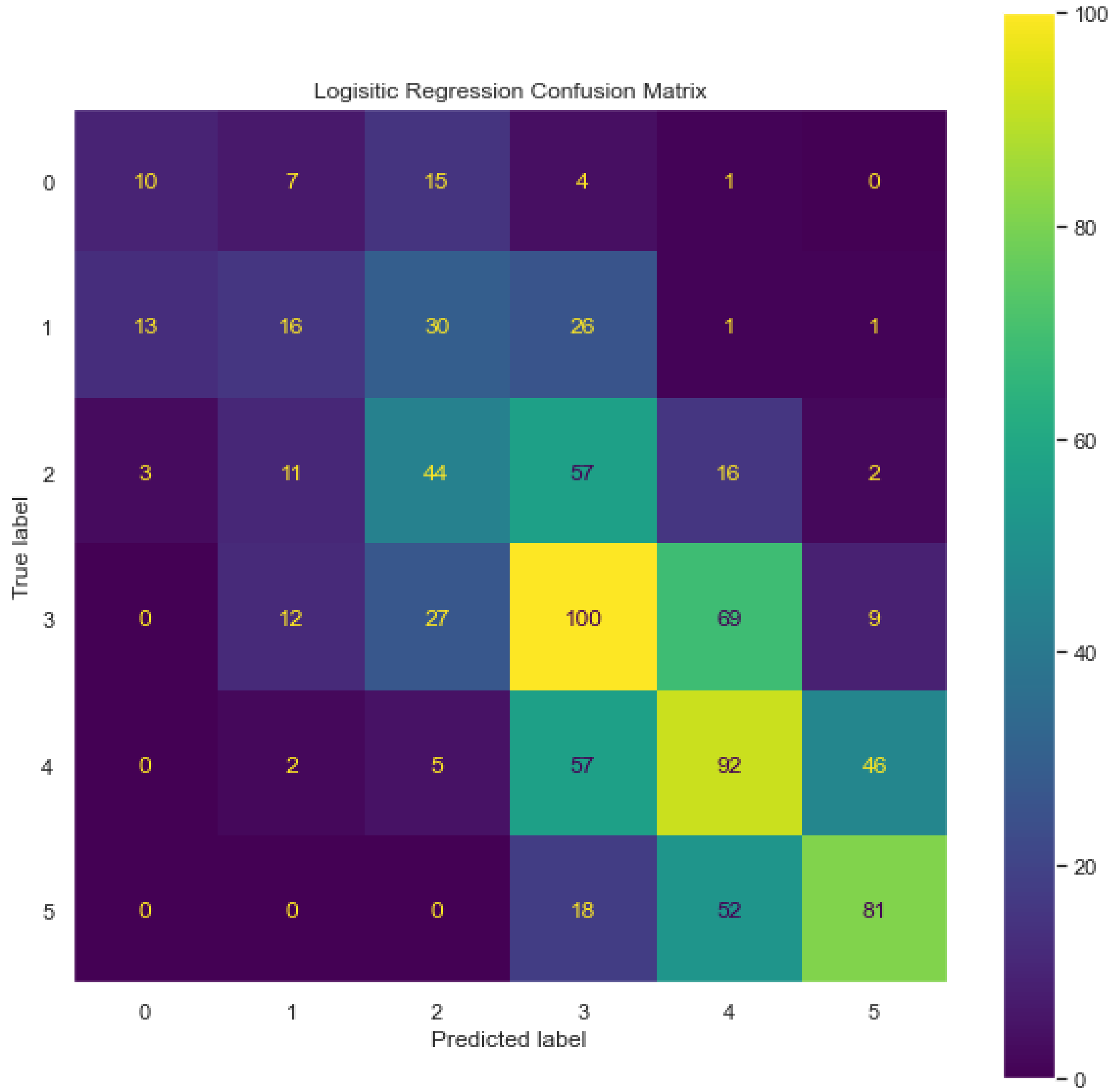

Figure 7

"Smurf" Detection Model

The second model I built was a smurf detector. Smurfing is when a higher league player makes a new account to play in a lower league. Sometimes this is because they want to play with their friends, and sometimes they do this because they want to dominate the people of lower skill. As we can see from the previous charts, there can be quite a big difference in skill between leagues that are two or three leagues apart. Smurfing is a problem because it can ruin the game for the players getting dominated. The game is meant to be played between players of equal skill. This keeps the gameplay fun and interesting for everyone. When smurfs are detected, they will usually receive a large boost in their skill rating so that they will quickly get removed from the lower leagues and into the higher leagues.

With the above problem in mind, I also built a smurf detection model. This also required a bit of data wrangling and additional labelling. Leagues 1-3 were labelled as 'Low' and leagues 5-6 were labelled as 'High'. These labels were chosen so that players that were smurfing from the 'High' leagues could be detected when playing the 'Low' leagues and could be handled. Leagues 7-8 were excluded for the same reasoning as for the previous model. League 4 was dropped as it would simplify the model and this isn't a common league either for smurfs to come from or go to.

Again, the aforementioned algorithms were trained on the data. Each model was gridsearched to optimize hyperparameters for the ROC AUC score. This metric was chosen because it is threshold independent and I planned to threshold after hyperparameter optimization. The ROC curves for the three models can be seen in Fig. 8. Their scores and hyperparameters can be seen in Table 2.

Classifier	**ROC AUC Scores**	**Best Hyperparameters**
K Neighbors	0.92	N_neighbors = 47
Logistic Regression	0.93	C = 0.359 Penalty = l2
Random Forest	0.93	Max_depth = 9 Max_features =auto N_estimators = 500

Table 2 ROC scores for smurf detection models[3]

[3] Logistic Regression had a score of 0.9276 while Random Forest had a score of 0.9278. These round to the same thing, but I choose to go with the slightly higher value.

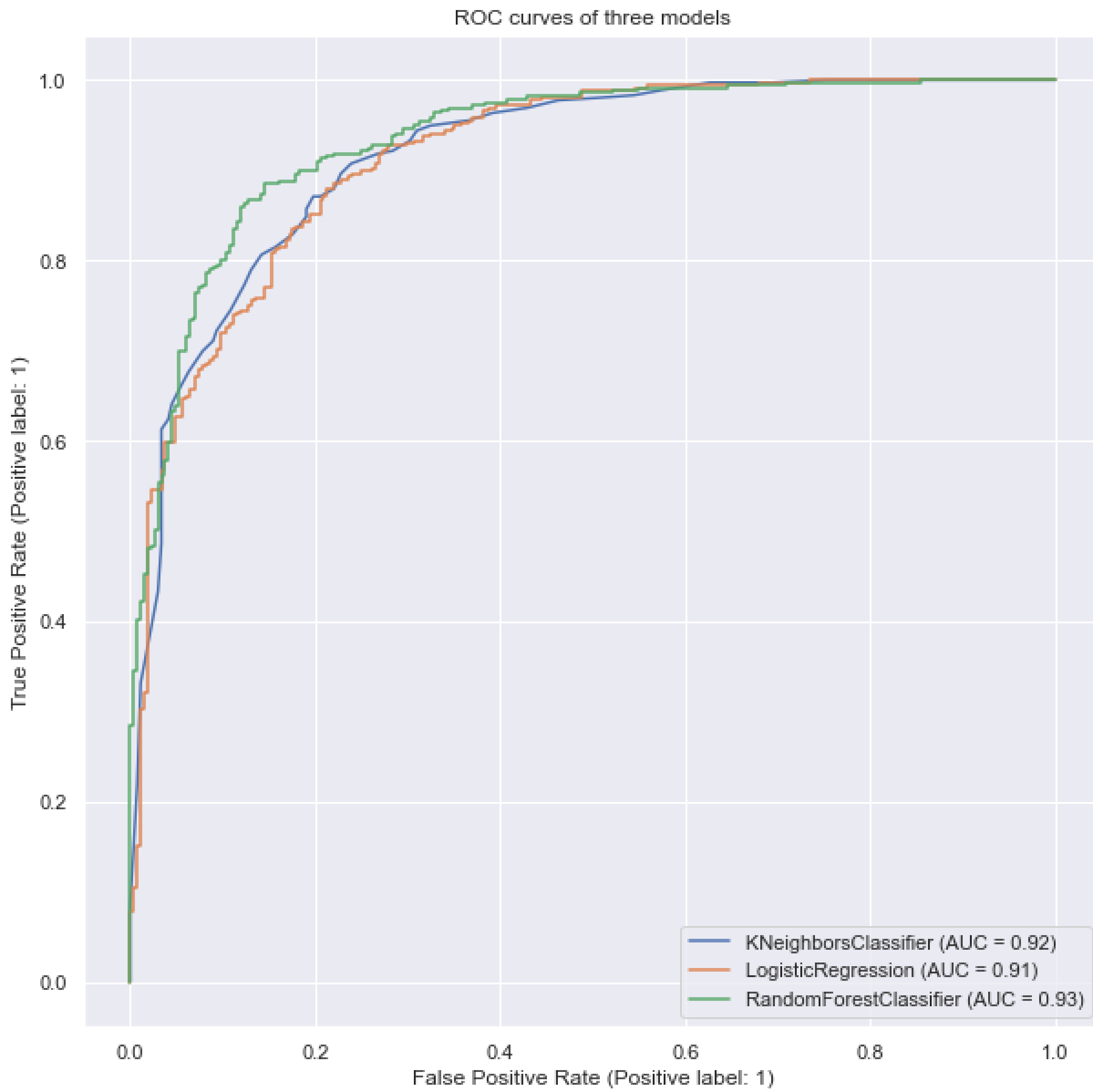

Fig. 8

The model that had the highest ROC AUC score was the Random Forest model with an ROC of 0.93. The next step was to perform thresholding to optimize for the business case. In this case, I wanted to prioritize recall because I wanted as many smurfs as possible to be detected and labelling a few non-smurfs as smurfs isn't a big deal. These players would simply get a league boost and no one complains about that unless they're smurfing. In Fig 9 below, I plotted out the Precision and Recall at every threshold to view the possible options here. I determined that a threshold of 0.23 would be the best option which yields the following results:

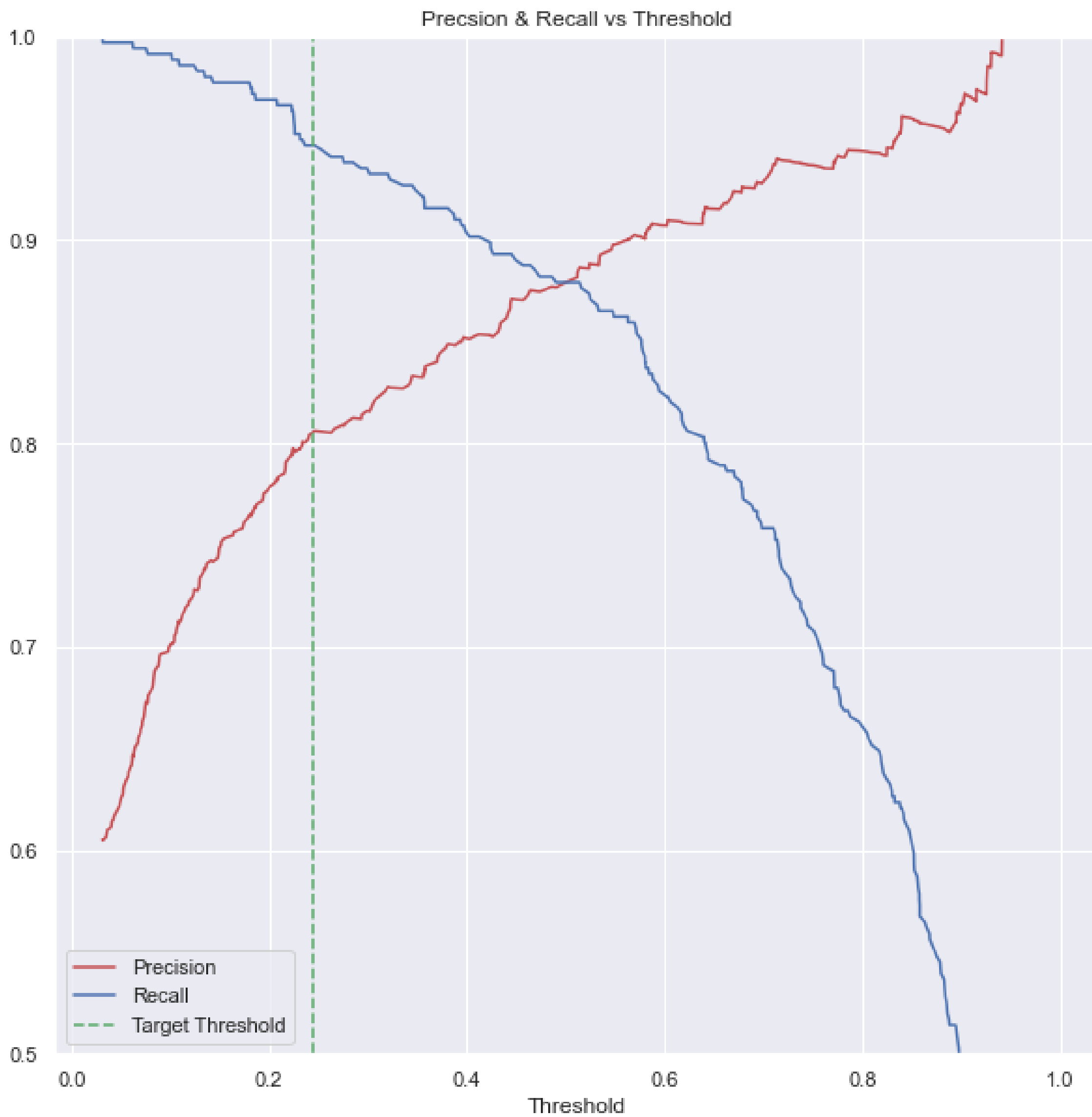

Fig. 9 Precision and Recall graph for Random Forest smurf detector

Class	Precision	Recall
Non-smurfs	0.90	0.67
Smurfs	0.79	0.95

Table 3 Classification Report for Random Forest Model at Threshold 0.23

At a threshold of .23, our model predicts smurfs with a recall of 95%. Now, this dropped the recall for non-smurfs to 67% percent, and while that seems worthwhile to me, it would be worth speaking with stakeholders to determine whether or not this tradeoff makes sense to them from a business perspective or if they'd prefer to have the threshold adjusted.

Conclusion

In the realm of Starcraft 2, these models can be incredibly useful. Having fair and fun games is essential to having a thriving community, and making sure players are participating in games of equal skill is vital to this. The league placement model can help place players in the league they should be in and the smurf detection model will help ensure that players can't cheat the league system to command lower level games.

Fascinating insights can also be derived from the feature importances. To answer the question I posed in the introduction, 'What matters the most when trying to improve at Starcraft 2?', it seems that Action Latency, APM and Number of PACs are the most important skills overall. This does imply that one's speed has a lot to do with their skill, but this excludes one's ability to make decisions. So while these skills are most important overall, we saw that it can be more nuanced from league to league and this can help to show how decision making has an impact on league placement.

While these models can be very useful, there is always room for improvement. The first area for the smurf model is getting a real smurf data set. In this analysis, I looked at players playing in their own leagues. It would be interesting to see data from actual smurfs and see how that differs from the data in this analysis. Also, this data is about 8 years old. Modern data could be different and would be interesting to analyze to see if results could be improved. Another thing that could be done is to examine the misclassifications and see if patterns could be detected and adjust the models accordingly.