**Starcraft Data Mining Project**

1) Generate your own data for prediction.

2) Explain your the visualization results with at least 100 words for each plot.

* Bar graph best feature - Matplotlib

This graph shows the importance of the features from the gradient boosting algorithm. The most important feature is ProtossReavor. ProtossReavor describes a ground unit that is used mostly as a siege unit. The reavor has a slightly longer range than most defense units. To win in Starcraft you need to kill all of your enemy's units and buildings. This makes the reavor an important unit because it allows you to destroy your enemies bases and defense units while their defense units can’t hit them. The second most important feature is ProtossShields1. ProtossShields1 is an upgrade to protoss units in the game. It increases the amount of damage they can take before they are killed. This is important because the stronger your units are the easier it becomes to eliminate the enemy units without losing your own.

Best feature pie chart - plotly

This chart, like the bar graph above, shows the importance of certain features. It shows the importance of the top 9 features individually and shows the importance of the other 30 features combined. The most import feature appears as the second largest piece of the pie chart due to the grouping of the other 30 features. The grouping of the 30 other features has an importance of 39%. The importance of ProtossReavor is 9.2% and the importance of ProtossShields is 8.9%. It makes sense that these features are the most important, the grouping of 30 makes sense because it is the sum of the importance for 30 features. The ProtossReavor is an important unit in the game due to its range and ProtossShields in an important upgrade in the game.

* Categorical Scatterplot - Seaborn

The Seaborn Categorical Scatterplot shows the relationship of the occurrences of one attribute and its classifications at the midBuild, or halftime, of the game. The x-axis is the classification of strategy taken by the player at the midBuild, or halftime, of the game. The y-axis is the number of occurrences of that attribute. Then overall the scatter plot will show the relationship between the counts of each attribute and its classification. Therefore we can take each attribute and show the occurrences of each classification and look for patterns. The pattern we are looking for is if there are attributes that are most often categorized as a certain strategy or alternatively not very likely to be classified as a certain strategy.

* Dynamic Histogram - Bokeh

The dynamic histograms we plotted show the distribution of the data for a given feature. Every graph has a large column for the 0 value, and a good distribution other than that. The reason that the 0s are still there after preprocessing the data is because they are extremely meaningful 0s. The 0s represent decisions that are made in gameplay. Choosing a different upgrade or unit creates a 0 in certain features. For example if the player chooses to upgrade ProtossGroundArmor1 the ProtossArmor1 feature would recieve a 0. If we were to replace these 0s with the mean for the data valuable information would be lost.

Copy and these plots and explanations in a word file. [Example](https://cmpt300.github.io/project/Explain%20Plots.docx).

3) Answer each of these questions using at least 100 words.

1. Please describe the data of your mode. For example, categorical or continuous data, **distribution of data**, source of data, dimension of data, size of data, etc.

The dataset we chose is a set of continuous data collected from playing the video game StarCraft: Brood War. StarCraft is a real-time strategy game, where you play as one of three races, humans known as Terran, the high-tech aliens Protoss, or the parasitic race Zerg. The game is centered around building a base, gathering minerals, producing troops and destroying your opponent. The source of this data is from Ben Weber and his research.

The dataset has 1140 rows and 56 attributes before cleaning the data. The data is describing choices done by a player playing against another player over many, many games. The dataset does not describe how to play the game, where to move troops, or how to move the units and similar actions. However, it describes the build order and overall strategy of the player.

The attributes used in the dataset describes an important action, such as putting down a building, starting an upgrade or producing a unit. As an example, ProtossPylon describes starting to build a pylon, while ProtossGroundArmor1 describes starting the ground armor upgrade. The classes used are called midBuild and are FastObs, FastLegs, ReaverDrop, Carrier, FastDT, FastExpand, Unknown. Each describes a specific way or strategy to take for the midgame.

2. How do you preprocess the data? Noise? Missing value？ Outlier？

For preprocessing the data we first cut out some of the attributes. We decided to do this because there are some attributes that are simply upgrades of previous attributes and getting rid of these upgrades will allow us to compare attributes more effectively without losing too much information. We removed the outliers of the dataset by using a standard deviation of 1 to trim outliers from the data. Finally we removed the missing values from the dataset, this did not include zero values because those are valid values. We then split up the data into 80% for training data and 20% for testing data.

\*\*\*\*Add point about removing 0’s, The zero’s are crucial to the choices

3. Which five models did you choose to build the model? Why do you choose these five models based on your data?

We used the decision tree, random forest, k nearest neighbors, gradient boosting, and logistic regression algorithms to build our model. We chose the decision tree algorithm because it is a basic starting point for creating a model and it works with almost all data types. We chose to use Random Forest to compare with the decision tree. Next we chose k nearest neighbors to see how accurate this algorithm was in classifying the data. Gradient boosting was chosen next to see if boosting would increase the accuracy of our model. Finally we chose logistic regression to handle the large number of independent variables.

4. Which one is best? Can you explain why this model performs best on your data?

The accuracy of the gradient boosting algorithm is the best of all the algorithms we tried, because of this we chose to use it for our model. Our model has an accuracy of roughly 0.88 when predicting the midbuild. This is the best algorithm because it allows for forward optimization. As the algorithm runs through the build data and it improves itself until it can not improve further. We have the gradient boosting algorithm build 500 trees. Each time it constructs a new tree it improves on the previous tree. This allows the algorithm to create an accurate model.

5. Is it possible that your model is underfit or overfit? Why?

We do not think that our model is under fit. We also do not believe it is underfit. We believe this because we currently have an accuracy score of 0.88. You usually want a model to have an accuracy score of 0.7-0.9. If it is under this range it is considered under fit. If it is above that range it is usually considered overfit. However, there is a little wiggle room for it being overfit. If the model is 100% accurate then it is definitely overfit. Our model’s accuracy score is near the top of this range so we believe that it is a good model.

6. What features are most importances of your data? Why? Can you explain it based on the context of the data?

When we analyzed feature importance the most import features were ProtossReavor and ProtossShields. It makes sense that these features are important in predicting the midbuild due to their importance in actual gameplay. As explained in the description of the graphs, the reavor unit has a long range and is used to destroy defense units. This is important because you need to destroy all enemy units and buildings to win the game, so by destroying the defense units it makes destroying the building much easier. The shields are important because it makes your units harder to kill. Again this helps you kill enemy units while minimizing the risk of losing your own units.

7. What is the most difficult part in the first two steps of this project? Why?

The most difficult part in the first two steps of this project was preprocessing the data, and displaying the plots. Preprocessing the data was difficult because we had to think long and hard about which attributes would make sense to keep while simplifying the dataset. Keeping all the 56 attributes did not make sense since there was so much overlap between the attributes and it was too complex. We also had to make sure that removing the outliers would give our dataset a better distribution. The next difficulty was deciding what plots to use when displaying our data and results. Many of the graphs we tried did not make sense with the data set and at the beginning we had too many overlaps between attributes and could not get any kind of real comparisons.

8. Is there any plots, parameters, or data that you cannot explain it?

There was one plot that we could not explain, a seaborn swarm plot. The seaborn swarmplot showed the features on the y axis and values on the x axis. The points that were plotted were the different midbuilds, and where they take place in relation to the values of the features. This graph was hard to read and didn’t give us any valuable information. It didn’t help us understand the data any further. We couldn’t tell what features were important or at what values certain midbuilds happen. These things led to us deciding to remove this plot from out project.

9. What aspects of your model that you could improve? Why?

One improvement we could make would be choosing an easier dataset. Our data is a small portion of a much larger data set that is published by google’s deepmind. It is a collection of decisions made in the game Starcraft: Brood War. The reason we believe we could have chosen an easier dataset is because understanding the data and what it is representing, is difficult if you have no prior experience with Starcraft. The purpose of the dataset is to determine what the best build to take is at the midpoint of the game. Another issue with our dataset is that there are a lot of meaning 0s in the data. They represent not choosing something in gameplay. This made it difficult to clean the data and make it easier to work with.

10. Why do you think your model work on your data?

We believe our model works well with our data because of our accuracy scores. We can not use a ROC curve to check the accuracy of our model because we have 7 classes and the ROC curve is designed to be used with binary classes. If we were to use the ROC curve it would be skewed because we would be comparing one class to the other 6 showing that our model has a much lower accuracy than it actually does. Due to this we relied on accuracy scores to check our model. Our model currently has an accuracy score of 0.88, which leaves us to believe that we have an accurate model that is neither overfit or underfit.