Midterm Report

Edmond Mui
em656@cornell.edu

Vanshika Bansal
vb273@cornell.edu

Auston Li
al884@cornell.edu

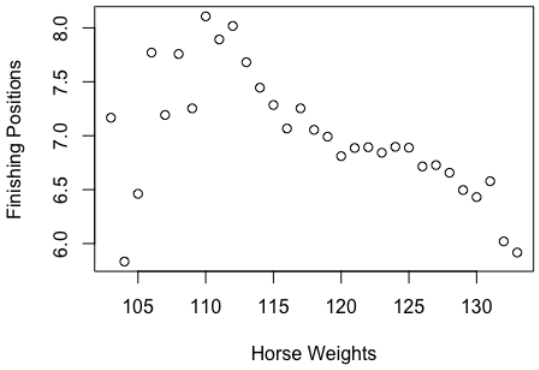
October 27, 2017

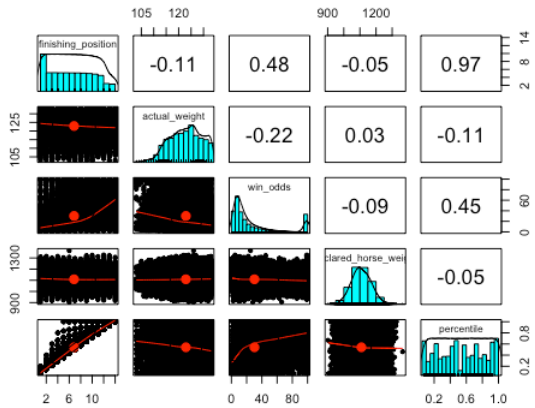
1 Problem Statement

Our project's goal is to determine the winner of a horse race for a specific field of horses, jockeys, trainers, and racing conditions. Given a field of horses and race conditions, we hope to develop a model which outputs the expected position that each horse in the field will be in at the end of the race. In addition, we may try to develop a long-term betting strategy which can hopefully be run over all Hong Kong horse races and return a net profit overtime.

2 About the Data Set

Our project examines Horse Racing in Hong Kong from 2014 to 2016. It is comprised of basic horse information such as horse names, jockey names, trainer names, finishing positions, etc. It also provides more detailed track information such as track conditions, race length, etc. To view some of the basic relationships between the features of our data, we first looked at how the weights of horses affected the horse's finishing position. As shown by the plot on the left, it seems that the lightest and heaviest horses tend to do better on average (1 is the best finishing position value). On the right, we picked features (*finishing_position*, *actual_weight*, *win_odds*, *actual_horse_weight*, *percentile*) that intuitively seem to be correlated with predicting future race finishing positions and plotted a *pair.panels* plot for them.





3 Preliminary Analysis & Validation

3.1 Data Cleaning & Random Forest Model

An important factor to consider while cleaning up the data is what we are cleaning it up for. Different models would benefit the most from different types of data, but in general most benefit from numerical data. Currently, we have cleaned the data up for preliminary analysis with random forest which only trains on numerical data.

Another important factor is what information we want to predict and train on. We considered a few options for these. We can either predict individual positions for every race, or only predict which horses win or lose the race. The latter can be done by either picking a position, for example 3, and assuming every horse at that position and smaller wins while all others lose, or by picking a percentile, for example 10%, and assuming all horses in the top 10% win while all others lose. For both of these predictions, our data would be cleaned and trained on differently. For the first, for instance, our predicted variable would be different positions while for the second, it would be binary (1 for wins, 0 for losses).

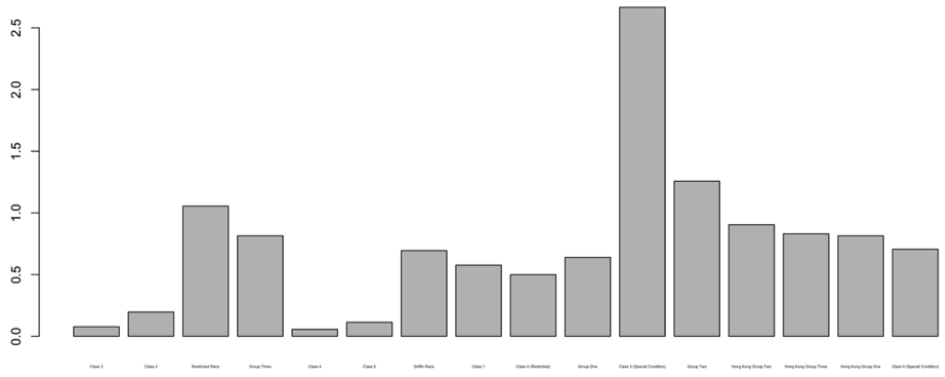
For training the model itself, there are different options to choose from. We can train on how different horses perform in different races with different conditions, like the class of race, race distance, track condition and so on. Another option, the one we chose for our preliminary model, is to train on the features of a horse. This implies, assuming a horse has similar features to those we trained on, for instance weight, average percentile rankings or trainer, we can predict its rank in a race. We chose this so that we can potentially predict the rank of any horse (even ones we haven't seen before) if we have its features. We hope that this will also prevent over fitting, as we aren't biased against new horses and we actually have enough examples to train on. As an average horse runs very few races a year and we only have 3 years of data, training on individual horses would currently over fit our model. Another way to counter over fitting has been using the average percentile ranks for horses, trainers and jockeys in all races so that individual people and horses don't bias our results, only their performances affect the outcome. This way, we can also test new horses, trainers and jockeys if we can get their average percentile estimate one. Another advantage of doing this is to ensure our numeric data actually has meaning; instead of assigning values 1 to the total number of jockeys to different jockeys randomly, we actually metricized their performance.

Fortunately, we didn't have significant data missing. The only *NA* values in our data were in examples where a horse was withdrawn or injured, or somehow unable to finish the race. If a horse is withdrawn from a race, we discarded his example from our training set as withdrawal is unpredictable and will not affect average percentiles ranks significantly. If a horse is unable to complete a race after it begins or is disqualified, we have assigned him a final position of 14 (the last possible position) to penalize the horse while calculating the average percentile ranks. In addition, for ties, the data set had finishing position values such as "1 DH" to indicate a first place tie, so we had to overwrite these values so that when converted to numerics, they would not return *NA*.

Further, we have currently assumed that an actual test set will only take in data from a single race where the race track, track condition etc will be the same for all horses, and therefore can be discarded when training with only the non-specific features of horses. We also removed data that seemed independent of our predicted variable, for instance *race_id*, the day of the race, and the number of the race that day. We also discarded features that cannot meaningfully be converted into numerical, such as the incident reports. In addition, we also removed data that we assume we will not have at the start of a race we are trying to predict results for, for instance the finish times and sectional positions.

Considering that we have different race classes, which differentiate horses into different

performance levels (with a lower number being better and some special categories), we have made separate models for each class to avoid under fitting the data. We then made a model for each class by training it on randomly sampled $\frac{1}{4}$ of the examples in that class, tested it against another randomly sampled $\frac{1}{4}$ of the examples in that class and plotted the average squared errors in our prediction of the ranks.



As these predictions are the squared errors of how incorrectly we are predicting the rank, we believe they look pretty good for a preliminary model. Moving forward, we will try to also build different kinds of models for other possibilities we have mentioned and try to improve our current model with feature engineering. We will use k-fold cross validation, as it intuitively seems like the most accurate way to cross validate and our data set is small enough to make this conceivable.

3.2 Strategy Discovery

Although most of our time was spent on developing the random forest model, we were able to analyze one betting strategy for Show bets. Show bets are won if the gambler picks a horse that comes in first, second, or third place. The strategy that was analyzed was rather basic, first finding which horses placed in the top three the most frequently (for the strategy, we only considered horses that placed in the top three in 75% or more of their races in the training set). Then, the strategy was then tested on a test set of races, where Show bets would be placed on the selected horses (if none of the selected horses were in a race, no bets would be made). To avoid overfitting, multiple random training sets were used (rest of the races were in the test set) and the results were that you would win roughly $\frac{1}{8}$ of the Show bets you placed. Although Show bet payouts vary from race to race (depend on the amount of money on each horse in the race), they hardly give payouts close to 8:1 (payouts are usually around 2:1), meaning that this strategy would result in a loss if played long term.

Moving forward, we hope to analyze various, more data driven strategies to see if they provide gamblers with any long-term arbitrage opportunities. In addition, we plan to use the random forest model's predicted finishing positions for horses in a race to develop a strategy where we place bets on the horses with the best predicted finishing positions. However, the number of horses to place bets on and what type of bets to place will be determined based on how accurate our random forest model is.