Final Report

Edmond Mui
em656@cornell.edu

Vanshika Bansal
vb273@cornell.edu

Auston Li
al884@cornell.edu

December 4, 2017

1 Introduction

Since 1884, the Hong Kong Jockey Club has created a multi-billion dollar industry from gambling on horse races. Given a field of horses, gamblers must choose from a variety of bet types (e.g. Win: Pick the first place horse, Place: Pick any one of the first to third place horses, Quinella: Pick the first and second place horse in any other, etc.) and depending on the bet type, will receive different payouts if their bets are correct. Gamblers usually place bets off prior information and often rely on various strategies when placing their bets.

2 Problem Statement

We hope to develop a model that outputs the expected position each horse will finish at the end of a race. We believe specific features from the dataset can be manipulated through either classification or regression models to provide us with an output of the horses' expected finishing positions. From the best model, we hope to be able to provide betters with an additional source of information when making their horse racing bets.

3 About the Data Set

The two datasets we used for the project come from all Hong Kong Jockey Club horse races, dating back to 2014 and ending in early 2017. Some features we thought were important from our two datasets, `horse` and `race`, are:

Dataset	Important Features
Race	Race Date Race Course Name; Race Number (the positions of the race during the day) Race ID Race Class (Race Leagues) Track Condition Incident Report
Horse	Horse Name and ID Jockey Trainer Actual Weight (Actual Jockey Weight) Declared Weight (Total Horse Weight) Draw (Horse Lane) Sectional Running Positions Win Odds Race ID; Length Behind Winner

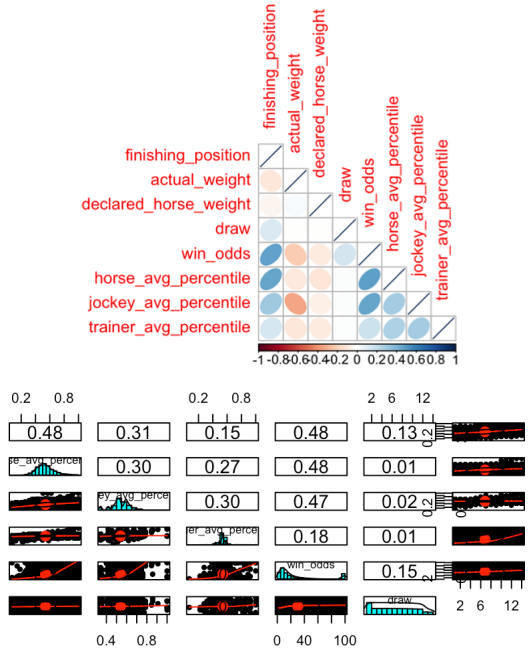
Although many of these features could be important in predicting a horse's race position, we tried to generalize our model to not depend on any specific horses and be adaptable to new data. The main reason for this was because of the limited number of races a horses participate in over their careers, we would not have enough data to create generalized predictive models for specific horses. Generalizing will also allow our model to predict the positions of horses that were not in the training set.

We only train on draw (as the lane a horse runs in can make a difference in an oval field), win odds (to use external predictions to improve our model), actual weight, declared horse weight, and historical statistics about the trainer, horse, and jockey performances. The historical statistics used for the trainer, horse, and jockey performances were the averages of the historical finishing position percentiles (finishing position divided by number of horses in the race for each race). This was also done to ensure that we don't train on specific horses, trainers or jockeys, but rather, we can incorporate new horses, trainers and jockeys if we know their historical performances.

We did not include features such as track condition, race distance, and track name. While intuitively, these features are helpful in predicting the results of a race as certain horses might perform better on certain

tracks or at certain distances, we do not have the means to create a generalized predictive model for specific horses (stated why before).

To start off our analysis, we decided to look into some simple correlations between the horse's features we found important against the horse's finishing position.



For the plot above, from left to right:
`finishing_position`
`horse_avg_percentile`
`jockey_avg_percentile`
`trainer_avg_percentile`
`win_odds`
`draw`

As shown by the correlation plots, we can see that actual and declared weights have a negative relationship with finishing positions, whereas the draw, win odds, and historical horse, trainer, and jockey percentiles have a positive relationship with finishing positions. This seems to imply that the heavier the horse and jockey, the better the horse will perform and that the better the historical performances of the horses, trainers, and jockeys, the better they are expected to perform. However, we will not solely rely on this simple correlation plot to determine the relationship between our features and finishing positions, but rather, conduct more extensive data analysis before drawing conclusions.

4 Data Cleaning

In terms of actually cleaning the data, we did not have many missing values. The only `NA` values in our data were in races where a horse was withdrawn, injured, or unable to finish the race. If a horse withdrew from a race, we discarded the horse's data since withdrawals are unpredictable and happen before the start of races (when horse bets can still be changed). However, if a horse is unable to complete a race after it begins (e.g. gets disqualified), we assign that horse the last position in the race to penalize the horse (since any bets made on that horse will be automatic losses). In addition, for ties, the dataset sets the tied horses' finishing positions to "[the position they tied at] DH" (for instance `1 DH` in the case of a first place tie). To fix this, we overwrote those values so that when we needed to convert finishing positions to numeric values, tied horses would not have a `NA` finishing position value.

Furthermore, while calculating average percentiles for the horses, trainers, and jockeys, we only used the average race results from the data in the training set. We did this so that horse percentile information from our test set is not taken into account when we train our data. As a result of this, there were some horses, jockeys, and trainers in the test set who did not have historical performance values. To adapt to this, we calculated the average percentile values for all horses, trainers, and jockeys and used those values to represent the "historical performances" of new horses, jockeys, and trainers respectively.

When dividing our training and test sets, we randomly sampled roughly 80% of the races to include in our training set, with the other 20% making up our test set. Then, assuming that different classes of races might have different weights for features that may influence results, we divided the races into five categories based on the races' classes. The first four categories are race classes that have at least one thousand examples and the fifth category is comprised of all of the other remaining race classes.

Category	Race Class
1	Class 2 – 2792 Examples
2	Class 3 – 9567 Examples
3	Class 4 – 10923 Examples
4	Class 5 – 4292 Examples
5	Others – 2024 Examples

5 Data Analysis

Using various models and techniques from class, we wanted to determine the best predictive model. We evaluated a few different models and compared them to find the best model. Given a set of input features, our models predict finishing percentiles for all horses in each race. Then, we turn thes finishing percentiles into finishing positions by multiplying the predicted percentiles by the number of horses in the race (also rounding to the closest integer if necessary). We then took the absolute values of the differences between these predicted finishing positions and the actual finishing positions and divided them by the number of horses in the race to quantify our errors. We divided by the number of horses because we believe being off by two positions in a race with five horses is worse than being off by two positions in a race with twelve horses.

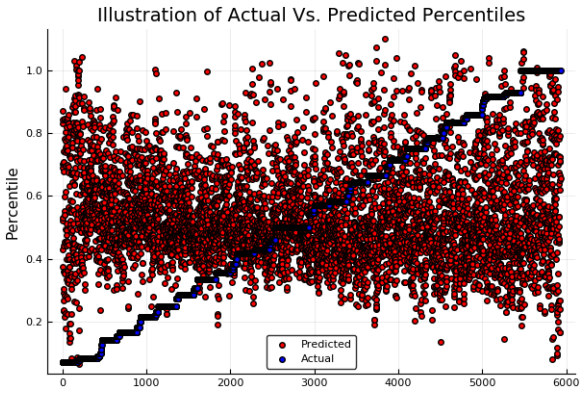
5.1 Regression Models

5.1.1 Simple Least Squares

To run a least squares regression, we separated our cleaned data into an output space of `percentiles` and input space of features:

```
trainer_avg_percentile
horse_avg_percentile
jockey_avg_percentile
declared_horse_weight
actual_weight
win_odds
draw
```

In our regression, we also allowed for an intercept term. In addition, because this was just a preliminary model, we did not separate the training and test set into five categories since we did not expect a standard least squares regression to work well for such a complicated process like horse racing. The results from simply running a least squares regression were, as expected, not promising.



We ordered the actual percentiles in increasing order to gain a sense of how our predicted percentiles fared. The issue seems to be that the coefficients generated for our features from running a least squares regression are not strong enough to accurately predict the changes in the actual percentiles.

5.1.2 Robust Linear Regression

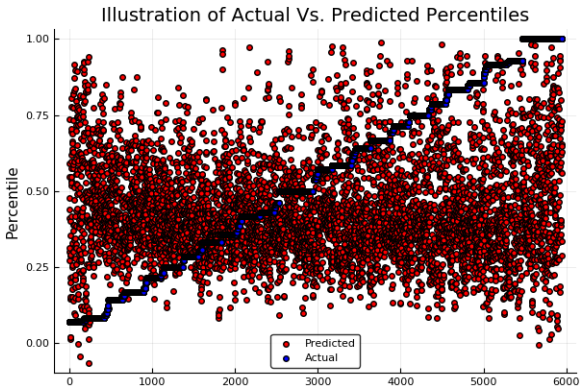
After our initial trial run of a least squares regression, we decided to utilize a robust linear regression to potentially account for any outliers that could have messed up our least squares regression. We also implemented the five categories that we discussed earlier and as a result, created five separate models to predict a horse's finishing percentile from the test set. These predictions were then quantified using the method described in the beginning of the `Data Analysis` section. The results were:

Category	1	2	3	4	5
Training Error	0.190793	0.19629	0.19255	0.205441	0.189731
Test Error	0.200911	0.21132	0.210132	0.218181	0.215063
% Difference	5.30%	7.66%	9.13%	6.20%	13.35%

We then ran a 10-fold cross validation. We noticed that the cross validation for the robust linear regression behaves relatively similarly to the single instance of the robust linear regression. In general, the training errors from the single robust linear model were better than the 10-fold cross validation since the 10-fold cross validation generalizes slightly better.

Category	1	2	3	4	5
Training Error	0.195411	0.196329	0.192228	0.204361	0.193403
Test Error	0.201483	0.211935	0.209833	0.217205	0.216089
% Difference	3.11%	7.95%	9.16%	6.28%	11.73%

To get a better idea of how well the data looks, we plotted how our predicted percentiles fared against the actual percentiles.



Unfortunately, like for the least squares regression, it seems that the coefficients generated for our features are not strong enough to accurately predict the changes in the actual percentiles.

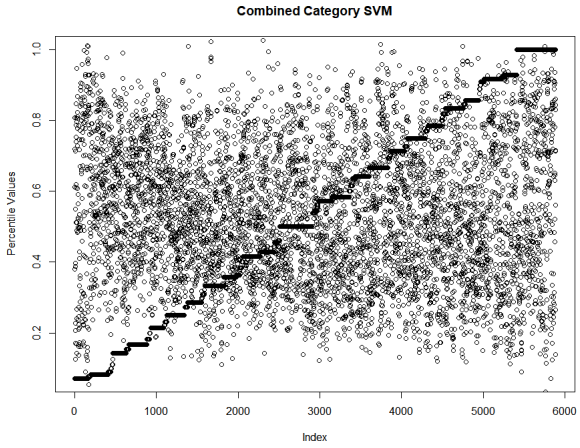
5.1.3 Support Vector Machine Regression

We then decided to choose a Support Vector Machine regression as a potential model because it is better at generalizing and reducing overfitting. It also does not require as large of an amount of training data and will perform markedly better at modeling nonlinear data compared to other linear regression models. The results were:

Category	1	2	3	4	5
Training Error	0.168736	0.180059	0.177489	0.183418	0.164541
Test Error	0.182022	0.203617	0.198869	0.203253	0.194981
% Difference	7.87%	13.08%	12.05%	10.81%	18.50%

Unfortunately, the errors from the Support Vector Machine model show a tendency towards overfitting (significantly larger test errors compared to training errors). However, even after attempting a k-fold cross validation, the resulting model was marginally worse, while also having significantly worse training error. Despite the obvious overfitting, the Support Vector Machine model generally outperforms the the robust linear regression model and its 10-fold variant in the test sets.

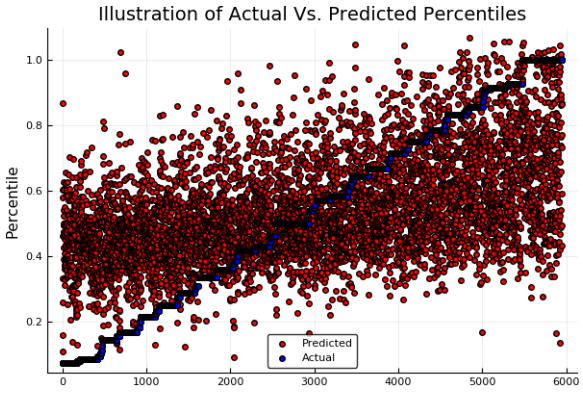
As with all other models, to get a better sense of the model, we plotted how our predicted percentiles fared against the actual percentiles.



As predicted, the Support Vector Machine model tended to overfit the training set and as a result does a poor job predicting on the test set.

5.1.4 Quadratic Loss with L1 Regularizer

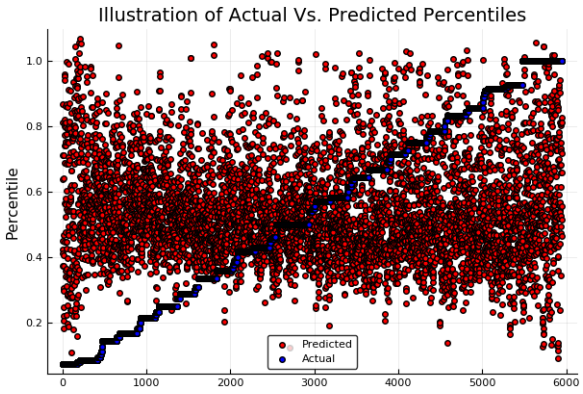
We then decided to try using the proximal gradient method using a simple Quadratic Loss and a L1 regularizer. The results from running this regression were actually quite successful.



Although not a perfect fit with the actual quantiles, there is some indication that this model is able to predict how our input features affect the finishing position of the horses in our test set (slight upward trend in our predictions). We understand that like finance, it will likely be hard to get a perfect model using the limited number of features we had. However, it is still a promising start and once more accurate signals for a horse's finishing position are discovered, this model could potentially be extremely helpful in choosing which horse racing bets to make.

5.1.5 Huber Loss with Non-Negative Regularizer

After having some success with using the proximal gradient method, we decided to explore using the proximal gradient method with a Huber loss function with a non-negative regularizer. The Huber loss function was chosen in hopes that any effects of outliers in a horse's performance would not substantially affect our model. The non-negative regularizer was chosen simply because we are predicting percentiles that are non-negative values. The results, however, were not as promising.



Like the original least squares regression, the issue seems to be that the coefficients generated for our features are not strong enough to accurately predict the changes in the actual percentiles. This is somewhat surprising

because from experiences in class, we would have expected the Huber loss to perform better than a standard Quadratic loss due to the former accounting for outliers.

5.2 Classification Models

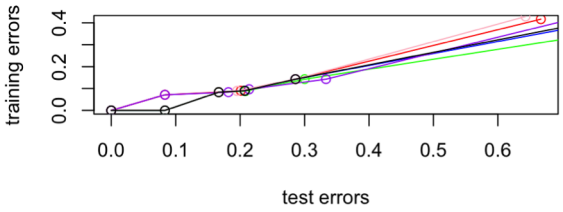
5.2.1 Random Forest

Due to our lack of overwhelming success with regressions, we decided to attempt a Random Forest model using classification trees as it has potential to work well for non-linear data. With the creation of a training and test set as explained in `Data Cleaning`, we ran a Random Forest model on the training data for each of the 5 categories and tested against the each category's test set. Below are the summary statistics for the errors as descried before:

Category	Min	1st Quartile	Median	Mean	3rd Quartile	Max
1	0	0.08333	0.16670	0.20920	0.30000	0.75000
2	0	0.08333	0.16670	0.20750	0.30000	0.83330
3	0	0.08333	0.16670	0.19990	0.28570	0.78570
4	0	0.08333	0.21430	0.21300	0.30000	0.75000
5	0	0.1000	0.1742	0.2027	0.2857	0.7692
All	0	0.08333	0.16670	0.20560	0.28570	0.83330

From these conclusions, we can see that our Random Forest model for Category 4 fits worse than the others. However, since we can see that the summary statistics for the model training and testing on the entire training and testinf sets respectively performs fairly well, we are able to assume that with the features we used, we can predict the finishing position of a horse with one model, regardless of what the category of the race is.

We then decided to compare these error statistics with those of our training error to see whether we fit our data well or not:

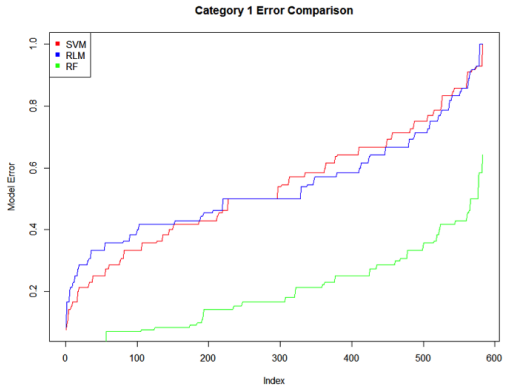


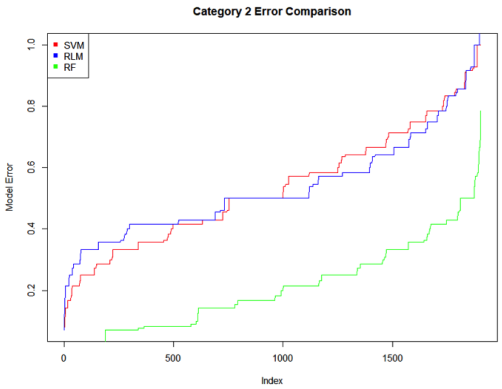
The colors represent each category with:
`Category 1`: Red
`Category 2`: Blue
`Category 3`: Green
`Category 4`: Purple
`Category 5`: Pink
`Training Set`: Black

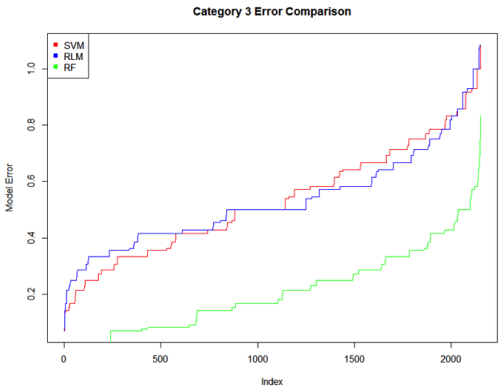
As we can see, the medians for the testing errors are about double those of training errors, but since the values are still pretty low, we can still conclude this Random Forest model fits our data quite well.

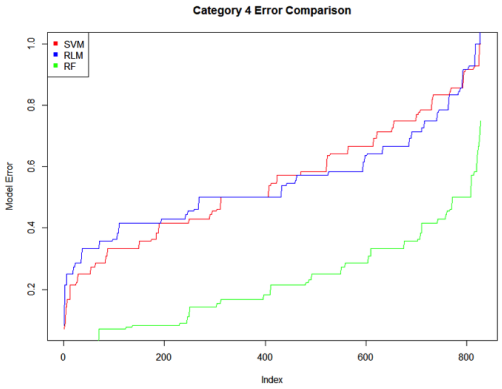
6 Results

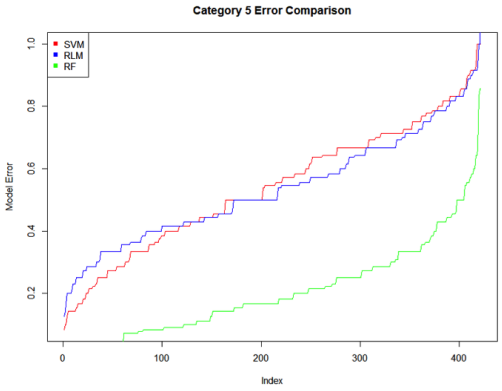
To determine which model works best for predicting the finishing position of horses, we decided to plot error comparison plots for our two best regression models (Support Vector Machine and Robust Logistic Regression; based off average errors) and the Random Forest model. Below are the six error comparison plots for each of the five categories of races in the test set and the entire test set:

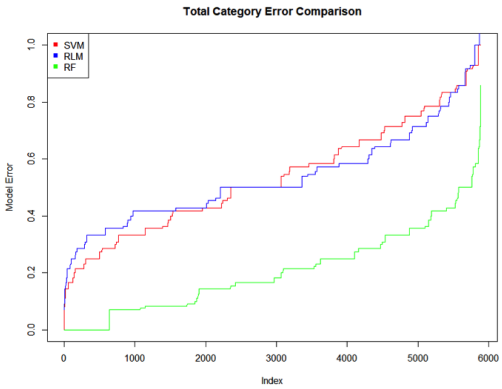












These error graphs compare the relative errors between the Support Vector Machine Regression model, Robust Logistic Regression model, and Random Forest model. Generally speaking, there are several noticeable trends. The Random Forest model has consistently lower errors and it takes on the shape of an exponential curve.

Both the Support Vector Machine Regression and Robust Logistic Regression models have somewhat heavy tailed distributions. Across the board, the Support Vector Machine Regression model performs better at lower index values, while the Robust Logistic Regression model performs better at higher index values. This means that the Support Vector Machine Regression model is the more accurate than the Robust Logistic Regression model when at making predictions when it makes the correct ones, but is the more wrong when it when the predicitons are worse. On the flip side, while the Robust Logistic Regression model does not have as many large error values, it has less accurate predictions even for its smaller errors.

Overall, it is quite clear that the Random Forest model does the best job at predicting the finishing position of horses given the features in our input space. This makes sense since horse racing is likely a non-linear problem and we simply do not have enough fea-

tures to create an accurate predictive model from regressions. However, conducting the regressions did allow us to see that the main drivers in the coefficients for predicting the finishing position of a horse were the historical performances of that horse and jockey (largest coefficient values in all the regressions). Therefore, these are good signals to have if we are to find more feasible features for our regression models.

7 Conclusion

From our analysis, we are fairly confident in how we cleaned our data and the approaches we took for the data analysis. In addition, we are confident that information such as previous horse and jockey performances (represented by the average percentile feature) are the most effective features in attempting to predict finishing positions of various horses. Although our best model, Random Forest, can be used in conjunction with another feasible betting strategies, we do not recommend using it as the sole basis for selecting bets. We believe that we have discovered some initial signals to predict a horse's finishing position, but they are not enough to predict positions at a level of accuracy that we would like (e.g. consistently generates revenue from horse racing bets), as explained by the analysis of our models.

Unfortunately, due to the relatively short careers of race horses, there is not enough data to make individualized position predictions for each horse. Had horses raced at the same frequency and had a career duration as, say, a veteran basketball player, the approach to analyze individual horses may have led to a more successful predictive model. Nonetheless, with our current approach of generalizing horses, we believe that the addition of new potential signals (e.g. Horse Age, Horse Condition, Horse Stamina, etc.) will produce a model that predicts finishing positions at a satisfactory level of accuracy.