1 Introduction

This research is aimed at discovering if golf has the same success other sports have with machine learning and fantasy league. Specifically, is it possible to pick a fantasy golf team based on model predictions that will perform as well or better than a human?

Fantasy sports are a type of online game where participants assemble an imaginary team of real players from a professional sport. Points are scored based of performances and statistical analysis of the chosen players. The participant with the most points at the end of the season wins.

Since the boom of the internet, fantasy sports popularity has grown dramatically. It is now estimated to have a $3-$4 Billion annual economic impact across the sports industry [**?**]. With the surge in popularity many leagues offer large prizes for their winners. The Premier League offer VIP hospitality suites to winners, golf leagues such as PGA Tour offer sets of Titleist irons and drivers, while American Football leagues offer prizes from $100 to $1 million, motivating players to pick a successful winning team each week. Various sports have had machine learning based teams compete in them. For example, Dr Sarvapali Ramchurn, of the University of Southampton has used machine learning to choose the best possible team for the Premier League, being able to "outperform 99% of these players" [**?**]. Thus showing that these techniques can be applied to fantasy leagues successfully.

Due to the popularity and prizes offered, Todays Golfer's fantasy league will be the focal point of the research. The following are the goals for the models that will be produced:

- Accurately predict the players finishing position using machine learning algorithms.
- Create a team optimiser to pick the best performing team (using the finishing position predictions) under the given constraints.
- Have a model that performs well enough to win prizes in the league

This area of research consists of multiple computational challenges which will be faced. Predicting the finishing positions of golfers in tournaments has never been done, so many machine learning algorithms must be tested to see which can accurately predict positions using previous data from tournaments. As all of these algorithms have not been used for this specific problem before therefore extensive testing must be done to determine which is most accurate.

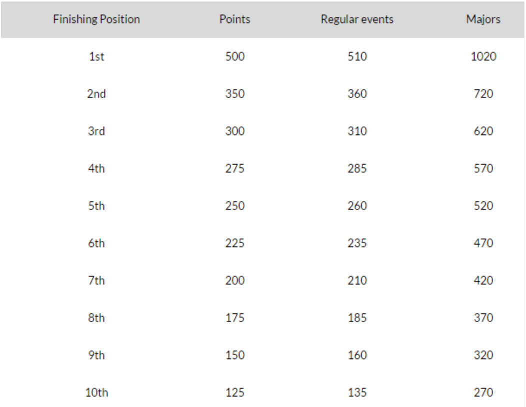
In addition to the models an optimiser must be designed to pick a team each week under the constraints, and simulate all the possible tournament weeks. It is crucial that the algorithm picks the the team which will score as many points as possible, to improve the teams chances of winning the league and the prizes.

2 Background

2.1 Rules For Fantasy Golf

This research will be applied to the fantasy league supplied by Today's Golf. Eight golfers are initially chosen to be the team for the first tournament. From then on each week a possible four transfers are available to be made, where current team players can be switched out for others. Each player has an assigned value, and for all eight players this must not be over $80 million. Additionally, one player from the team is selected to be the captain, who will score double the amount of points he earns while captain. Once a team is selected, it is entered into a league to compete against other competitors. Points are awarded for how well the players in the selected team performs.

Points are given to players based on their finishing positions of the event being played, whether or not they make the cut, and disqualifications. The cut is where after the first two rounds of the tournament, players scoring higher than the 70th lowest scoring professional, are removed from the tournament. The pointing systems can be found on the corresponding fantasy league website. For the Today's Golfer's league it is summarised in the following table:



Finishing Position	Points	Regular events	Majors
1st	500	510	1020
2nd	350	360	720
3rd	300	310	620
4th	275	285	570
5th	250	260	520
6th	225	235	470
7th	200	210	420
8th	175	185	370
9th	150	160	320
10th	125	135	270

Figure 1: *Points Table from 1st-10th*

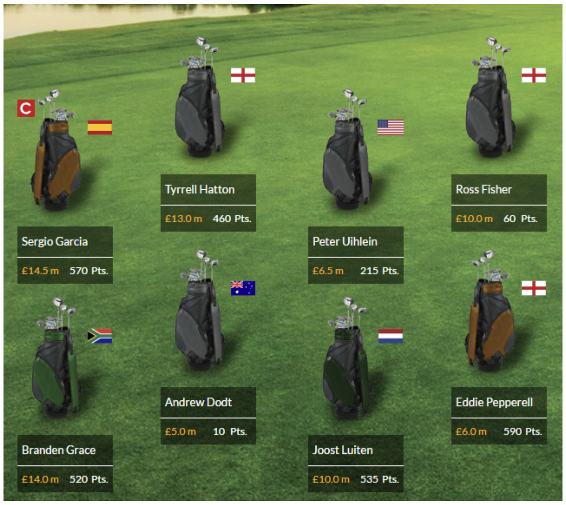
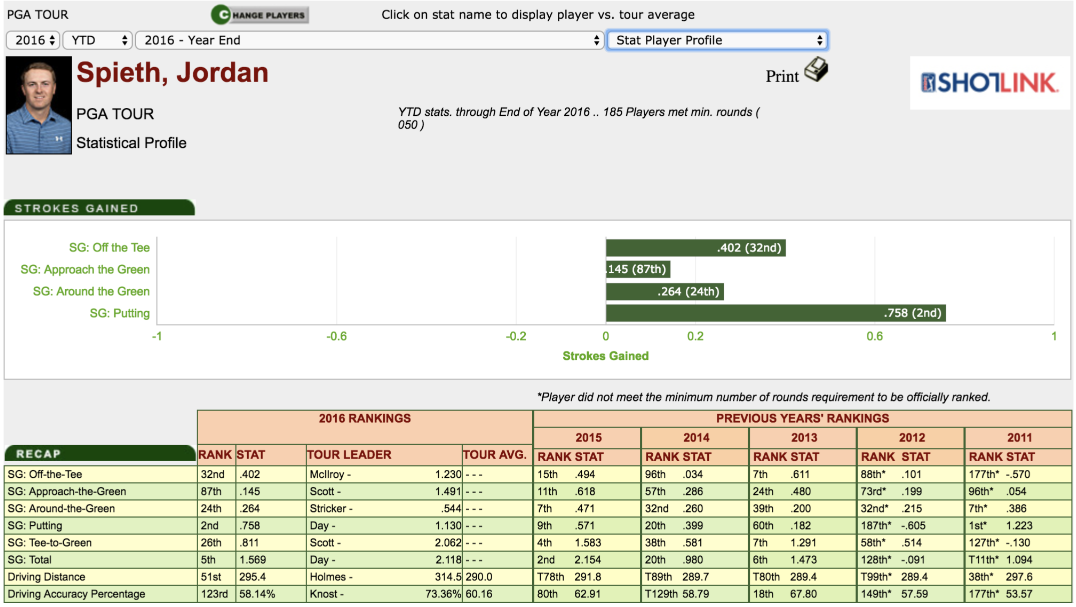


Figure 2: *Today's Golfer Team Selection*

2.2 ShotLink Dataset

ShotLink is the collection and analysis of shot-by-shot data during competition play. The data provides an in-depth view of the players on the PGA Tour, which allows coaches, graduate professors and students to analyse and gain a deeper understanding into the numbers game of golf [?]. Using the ShotLink platform, data has been obtained from 2004-2017 with over 17 million entries.

ShotLink works by mapping the golf course prior to the event, a digital image is used to calculate the exact locations and distances, vastly improving the deficiencies of scoring on paper in previous years [?]. The data went live in 2004 and has been readily available for use in academic papers. With the introduction of the ShotLink Intelligence Powered by CDW Prize, researchers have been trying to find the best new application of ShotLink statistics to golf. Below is an example of the portal where statistical data can be accessed about players.



Click on stat name to display player vs. tour average

Spieth, Jordan

PGA TOUR

Statistical Profile

YTD stats. through End of Year 2016 .. 185 Players met min. rounds (050)

*Player did not meet the minimum number of rounds requirement to be officially ranked.

	2016 RANKINGS				PREVIOUS YEARS' RANKINGS									
					2015		2014		2013		2012		2011	
RECAP	RANK	STAT	TOUR LEADER	TOUR AVG.	RANK	STAT	RANK	STAT	RANK	STAT	RANK	STAT	RANK	STAT
SG: Off-the-Tee	32nd	.402	McIlroy - 1.230	- - -	15th	.494	96th	.034	7th	.611	88th*	.101	177th*	-.570
SG: Approach-the-Green	87th	.145	Scott - 1.491	- - -	11th	.618	57th	.286	24th	.480	73rd*	.199	96th*	.054
SG: Around-the-Green	24th	.264	Stricker - .544	- - -	7th	.471	32nd	.260	39th	.200	32nd*	.215	7th*	.386
SG: Putting	2nd	.758	Day - 1.130	- - -	9th	.571	20th	.399	60th	.182	187th*	-.605	1st*	1.223
SG: Tee-to-Green	26th	.811	Scott - 2.062	- - -	4th	1.583	38th	.581	7th	1.291	58th*	.514	127th*	-.130
SG: Total	5th	1.569	Day - 2.118	- - -	2nd	2.154	20th	.980	6th	1.473	128th*	-.091	T11th*	1.094
Driving Distance	51st	295.4	Holmes - 314.5	290.0	T78th	291.8	T89th	289.7	T80th	289.4	T99th*	289.4	38th*	297.6
Driving Accuracy Percentage	123rd	58.14%	Knost - 73.36%	60.16	80th	62.91	T129th	58.79	18th	67.80	149th*	57.59	177th*	53.57

Figure 3: *Example of ShotLink portal*

2.3 Literature Review

2.3.1 Golf and ShotLink

When it comes to predicting the winners of tournaments, various websites such as Golf Digest have web applications which predict the winners of the upcoming tournaments [?] based on the variables you select and their importance. Although they do not explain how it works, the variables they have chosen gave me insight into those which may be important. Carrying on from this, paper [?] aims to understand golf performance, by using studies the author has carried out over the years. It delves into performance indicators that are flawed, such as "greens in regulation", and ones that are useful.

Mark Broadie has released many papers on the analysis of golf data which have been compiled together in his book "Every Shot Counts" [?]. In Broadies' book [?] he recognised the metric "greens in regulation" was defective, as previously mentioned. Thus in his papers he designed a "Strokes Gained" metric which is now used as a key statistical measure on the tour. Presently this statistic is used profusely by pros and amateurs to gain an insight into where they need to improve on their game.

In comparison to Broadie, Sen introduced the "Key Criterion of Success" (KCS) [?] a metric whose goal was to simply help predict a golfer's ranking over a season with greater accuracy than individual statistics. He suggests that the power of each individual golfing statistics is of limited value by itself. The KCS metric is the amalgamation of two existing measures deemed successful, adjusted weighted score and earnings per event. However, Sen appears to describe more limitations with the metric than benefits, throughout his paper.

The single use of machine learning in the field of golf comes from paper [?] where the winning score of players in the PGA Tour are predicted using first round results. In the paper, various machine learning techniques are used: Boosted Decision Tree, Neural Network, Decision Forest, Linear Regression and Bayesian Linear. Using calculated metrics such as Average First Round Score, Wiseman's models produce 67% accuracy to within three shots of the winning score. This demonstrates that there is the possibility to use machine learning in golf and the techniques Wiseman [?] used were then researched in more depth.

2.3.2 Prediction Algorithms

More research led to reading papers and articles around machine learning algorithms. Microsoft offer a "cheat sheet" [?] of algorithms and give a basic explanation of how they work and their limitations and strengths. This brought me to investigate the four algorithms that seemed best suited to my problem, linear regression, decision tree, random forest, and neural networks. Paper [?] demonstrated linear regression models were the most accurate in predicting the winning scores of PGA Tour events, thus motivating me to use them for my research. Due to this, improving a linear regression algorithm was researched further. Upon reading [?] I learned how to analyse the summary of the models to gain insight on which variables are important to the accuracy of the model.

Artificial Neural Networks (ANNs) could be argued to be the most common approach when trying to solve a variety of different problems, including predicting sports outcomes. [?] found that ANNs could be used to predict the finishing times of horses in competitive races to an accuracy of 77%. R provides a neural network toolbox [?] that could be used for the training and testing of the neural network models.

The introductory research of decision trees showed me all their possible uses, not only can they be useful for regression and classification but they have the ability of finding relationships between variables among hundred's of features [?]. Due to the number of variables the ShotLink data has on players, the use of decision trees for finding strong relationships between them will be explored.

Random Forests are "capable of delivering performance that is among the most accurate methods to date" [?] which is why they were investigated to be used for this project. Their ability to find non linear relationships between variables [?] separates them from methods like linear regression. Due to the nature of random forests and their "black-box" interpretation [?], understanding how to interpret them was explored. R, being the common go to for statistical analysis, supplies packages for each of the algorithms that have been stated.

The training and tuning of these models is key to the improvement of their accuracy and robustness. Multiple papers were read to further increase my knowledge in this area. [?] gave a brief overview of the multiple methods for cross validation, including holdout, k-fold and leave-p-out. Each of these methods have their advantages and disadvantages. As discussed in [?], k-fold cross validation was decided to be the better of the options as it gives "accurate performance estimation". In comparison to the other two which produce results with "very large variances" [?]. Kohavi [?] compared multiple approaches including the three aforementioned and recommended stratified 10-fold cross validation as the best model selection method. This can be applied to my models while they're being trained and tested.

2.3.3 Optimising Fantasy teams

Papers were explored that had investigated how to optimise various different fantasy league teams. [?] explains the team selection algorithm is an example of the Knapsack problem, and uses linear programming to solve the team optimisation problem. The Knapsack problem as explored in [?] involves having a set of items with an assigned weight and value. A subset is created which maximises the values within a given limit for the weights, leaving an optimal knapsack. Other papers, such as [?], and [?] both used linear optimisation to pick optimal teams for fantasy league football. This gave an insight into the best and most frequently used algorithms to solve the team selection problem.

2.4 Machine Learning Algorithms

The previously discussed data was then used to feed into the following algorithms, each previous average was used and the models R^2 and RMSE (root mean square error) values were calculated to see how accurate they were. RMSE is a "frequently used measure of the differences between values predicted by a model" [?]. Which is useful as comparing RMSE is comparing how accurate the models were. Descriptions of each algorithm being used follows.

2.4.1 Linear Regression (LR)

As a predictive analysis, the multiple linear regression is used to explain the relationship between one continuous dependent variable and two or more independent variables. Below shows the matrices that are formed to find a multiple linear regression model.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_0 \\ \epsilon 1 \\ \vdots \\ \epsilon p \end{bmatrix}$$

The x matrix is called the design matrix and consists of the observations of each independent variable, the y matrix contains the observations of the dependant variable (finishing position). β matrix contains all the regression coefficients, and ϵ, the error terms.

A number of previous average tournament results data will be used as a feature set and to build a regression model from this. The objective of this is to solve the matrices for the estimated parameters β_i using the following equation:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\vec{y} \tag{1}$$

Where X is the feature set, y is the actual results, and $\hat{\beta}$ are the regression coefficients [?]. Once a regression function has been found it can be used to make predictions. When previous tournaments' data is passed into the model it will output the prediction for the finishing position of the player.

2.4.2 Decision Tree (DT)

Decision tree machine learning will be tested as a way to make predictions. The data being used will again be a set of data from previous tournament's. DT learning involves using decision trees to go from an observation of a item to conclusions about the observation. A diagram of an example decision tree is shown in figure 4.

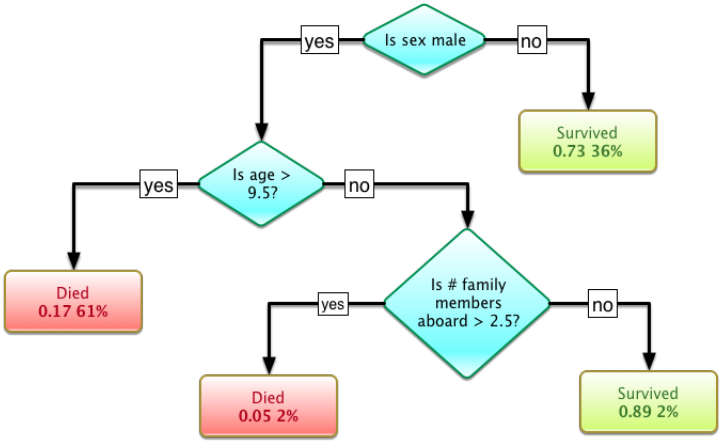


Figure 4: *Decision Tree Example - Titanic Survival*

A decision tree works by initially finding the best attribute and placing that as the root node (the top of the tree). The dataset is then split into subsets where each subset contains data of which an attribute has the same value [**?**]. The process is then repeated until leaf nodes are found for all branches of the tree.

Data from the players previous tournament results up to 2016 are used as training data, and the data from the most recent season will be used as testing. It will output the predicted finishing position of each player, where they'll be applied to the fantasy league.

2.4.3 Random Forests (RF)

The third algorithm to be used for predictions is the Random Forest. The algorithm will be fed a set of data with the players previous tournament statistics and their finishing position, which it will be trained on. The model will then be tested on the data from this season and used to predict each players finishing position.

The Random Forest builds on DT's by producing many trees, to aid with more in depth regression analysis. Essentially the algorithm works by picking a subset of features and producing a decision tree. This step is repeated until a set limit is reached, where we are left with a random forest of decision trees. When the testing set is passed to the model, each tree in the forest is used to predict the result. For each prediction target, all the predictions are gathered together and the prediction that has the most votes is returned, this is known as majority voting [?]. The Random Forest modelling will use the randomForest R package [?].

2.4.4 Neural Network (NN)

The final algorithm being tested is the Neural Networks algorithm. As with the other algorithms the neural network will be trained and tested using the same data sets, however the choice of features for each model is likely to be different. The algorithm works by feeding data through a number of nodes which are linked to other nodes. A network is predominantly made up of 3 layers of nodes, an input, hidden and an output [?]. An example of a neural network is shown in Figure 12.

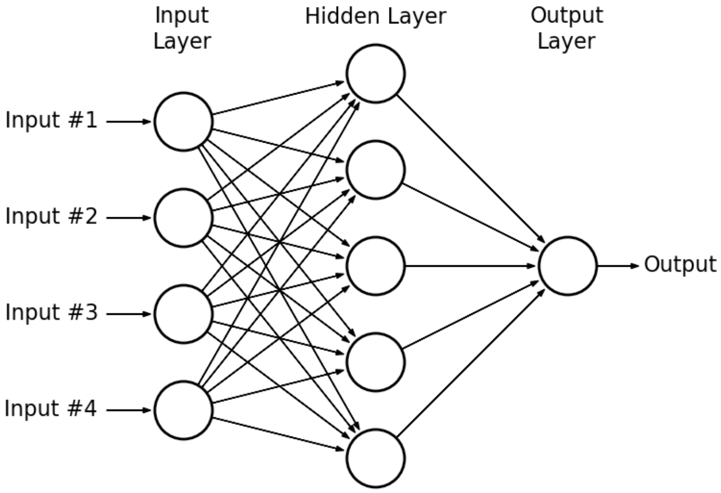


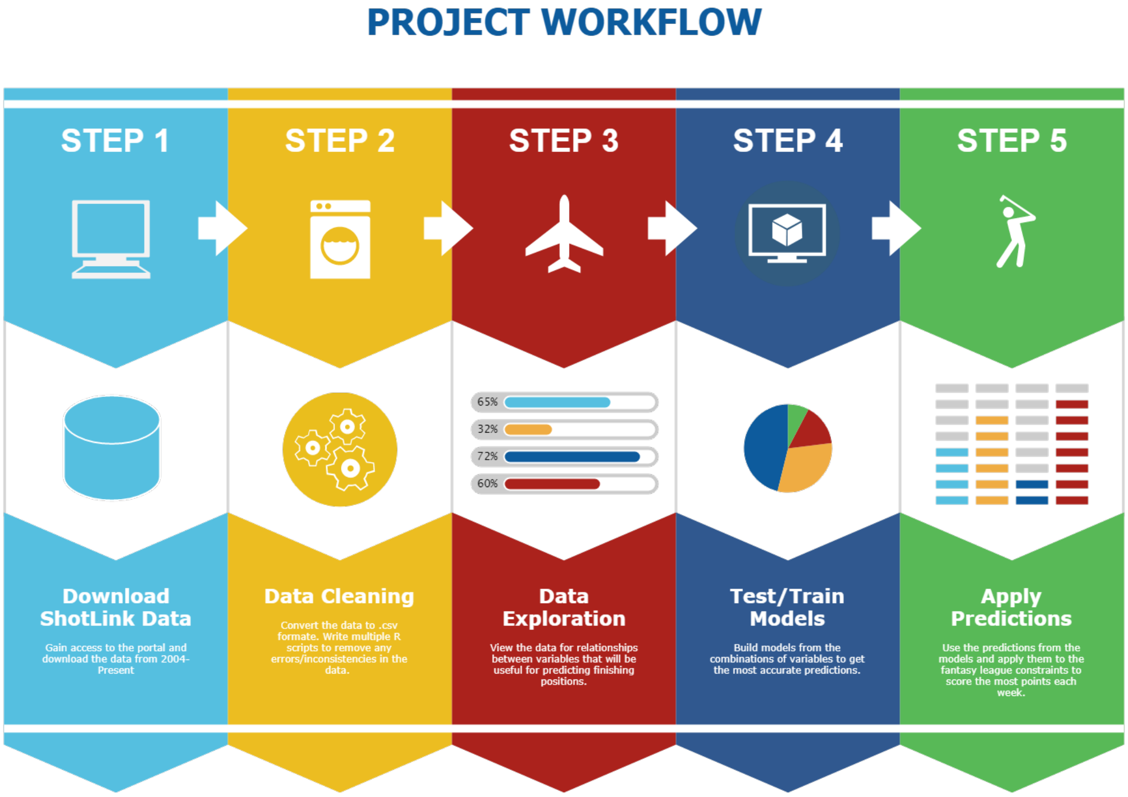
Figure 5: *Neural Network Example*

Each node has a weighted function, usually a logisitic function or softmax, that is predefined. After the input is fed through the hidden layer and reaches the output layer, the output is then compared to the actual value. Using its current prediction it updates the weights of the nodes so that the prediction is more accurate the next time. This is repeated until the model is as accurate as possible, which is known as error-back propagation [?].

Once the predictions are gathered for each entry of the testing data, they will be compared with the actual values to produce an RMSE value which will compared with the previously described models.

3 Data Collection

The work flow for the project is shown below, it has been split into its multiple stages.



Download ShotLink Data: Gain access to the portal and download the data from 2004-Present

Data Cleaning: Convert the data to .csv formate. Write multiple R scripts to remove any errors/inconsistencies in the data.

Data Exploration: View the data for relationships between variables that will be useful for predicting finishing positions.

Test/Train Models: Build models from the combinations of variables to get the most accurate predictions.

Apply Predictions: Use the predictions from the models and apply them to the fantasy league constraints to score the most points each week.

Figure 6: *Work flow of project*

In order to access the ShotLink dataset my supervisor Dr. Age Chapman applied for access, as the PGA Tour wouldn't give direct access to a student. I then wrote a contract (see Appendix D) that stated I'd adhere to all PGA Tour terms and conditions, and in return gain access to the data. The ShotLink platform contains results from 2004 on-wards, at various detail such as event, round, hole, and stroke level. All available ShotLink data from 2004-2017 was downloaded and converted to a .csv file to be used in R.

3.1 Data Cleaning

The data downloaded from the ShotLink portal had many issues with regards to consistency.

- Events: The events that are included in the dataset include stroke play, stableford and team events. For my research I'm only using the PGA events that are used on the fantasy league website.

- NULL values: Many Numeric Fields had NULL values due to this data being non-existent. For example, some "Player Earnings" values had NULL as they had not earned anything for the tournament. It was decided these values were to be set to 0 as this wouldn't affect the data.
- Missing Values: Some values were completely missing from the dataset, thus they were removed as I couldn't have inconsistent data in my dataset, nor could I replace them with other data.
- Incorrectly Calculated Values: In columns such as "Total Score" values were calculated incorrectly. So I wrote an R script to correct it for every entry.

The hole and stroke data sets were disregarded as they are of a much higher scope, but could be looked into for future research.

3.2 Data Exploration

Once I had my cleaned data set I ran a Pearson correlation coefficient test between the 193 numeric values. The Pearson product-moment correlation coefficient measures the strength of a linear relationship between two variables. It can take a value from -1 to +1. A value of 0 indicates no association between the two variables. -1, there is a total negative correlation and +1 and total positive correlation. This enabled me to see which variables directly correlated with Finishing Position. Table 1 shows the variables with the most informative Pearson coefficient.

Variable	R^2 Score
Stroke Average Rank	0.977
Scoring Avg Total Adjustment Rank	0.977
Bogey Avoidance Rank	0.977
Birdies Rank	0.977
Bogeys Rank	0.977
GIR Rank	0.976
Scrambling Rank	0.976
Eagles Rank	0.976
Total Driving Rank	0.976
Driving Accuracy Rank	0.976
Driving Distance Rank	0.976
Sand Save Rank	0.929
Putting Avg GIR Putts	-0.906
Birdie or Better Conv Greens Hit	-0.920
Total Greens in Regulation	-0.920
Pars	-0.931
Overall Putting Avg of Putts	-0.962
Round 4 Score	-0.965
Driving Distance Total Distance	-0.965
Total Strokes	-0.971
Round 3 Score	-0.975
Driving Acc Possible Fairways	-0.976
Driving Distance Total Drives	-0.976
Total Holes Played	-0.978
Total Rounds	-0.978

Table 1: *Variables most correlated with Finish Position*

Many variables were clearly related to finishing positions but not in a explanatory way, for example "Total Rounds" is irrelevant as it will always be four. Likewise "Round 4 Position" will always be the finishing position. This test gave me an insight into which statistics to use.

As well as this I designed and built a tool using the R library Shiny to further help my exploration of the dataset (see Appendix). The application was used to look at graphs of variables plotted against their corresponding finishing position, visualising their relationships helped me understand why the R^2 scores inferred there was an extremely correlated relationship.

The data explorer demonstrated the total positive and negative correlations, and showed how the previously mentioned ranks given to a player when failing to make the cut affected the relationships. Figure 7 shows how these ranks influenced the R^2 values.

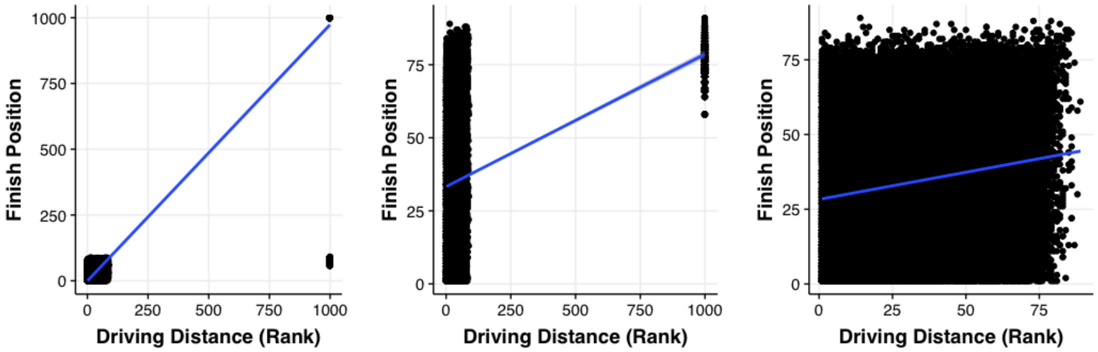


Figure 7: *Affect of 999 Ranks*

Through the progressive removal of the 999 ranks from Finishing Position, and then Driving Distance, it is clear that between these variables there is little correspondence. However, when viewing the PGA Tour website, they do not remove these ranks when presenting players ranks from previous tournaments. For this reason, it would keep the validity of the project if these results were kept in the dataset.

In addition, to explore the chosen variables the Boruta R package [**?**] was used to gain further insight in the importance of each variable. Boruta runs a random forest step-wise algorithm where it recursively gets rid of features in each iteration which didn't perform well in the process. This eventually led to a minimal optimal subset of features, as the method minimizes the error of random forest model. Below shows the graph outputted when run on the dataset. All variables which are green are deemed important, and are ranked in order of importance.

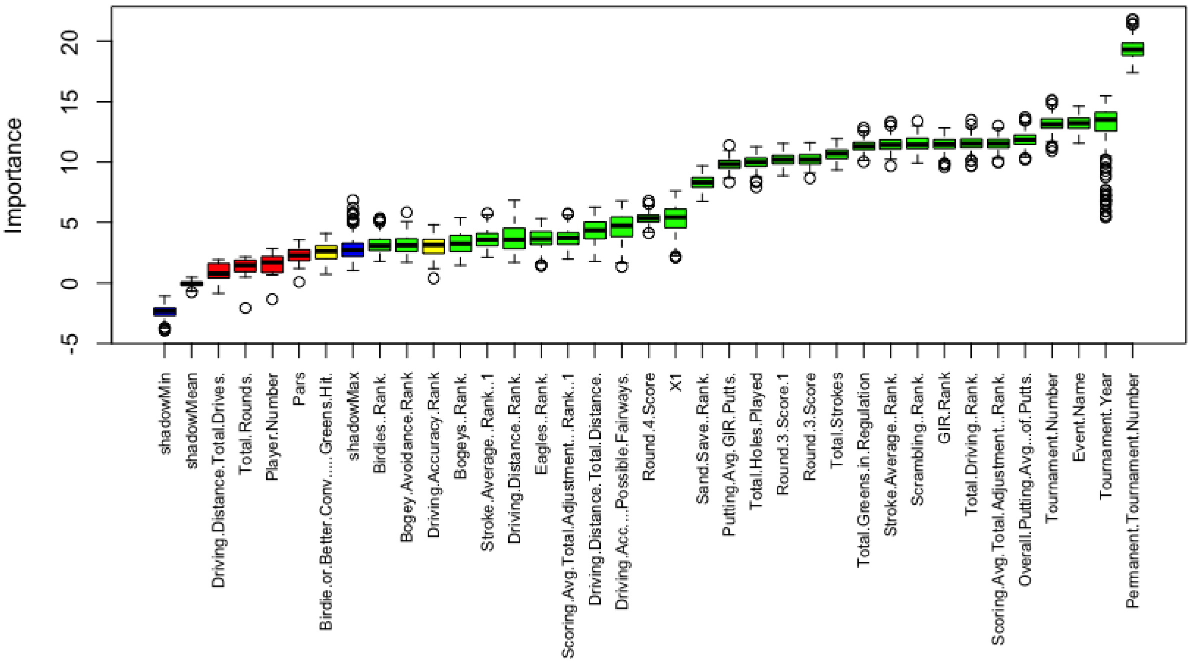


Figure 8: *Important Variables*

Taking the variables that showed to be the most responsive to finishing position I calculated their averages for each player from the previous one to eight tournaments.

4 Finishing Position Predictions

To predict finishing positions, using the variables discussed in the previous sections, each player had their averages computed for the previous one to eight tournaments they had competed in. Each previous one to eight tournaments data was stored in separate csv files making them easy to load into RStudio for predictions. During the testing it was found that using results from the previous tournament produced the most accurate models.

4.1 Algorithm Results

Once the dataset was finished I started applying the various machine learning techniques. The data was split into training and testing data, instead of the usual 70:30 split, results from 2004 to 2016 were used for training and data from the 2017 season for testing. This is due to there only being results from the fantasy league from that season, of which I will be able to compare to. In addition, it was found that using data from the previous tournament yielded the most accurate models. The results from each algorithm will interpreted through their RMSE score and their accuracy of predictions.

4.1.1 Linear Regression

Linear models were trained and tested using all possible combinations of variables to see which models were the most accurate. Once measuring the RMSE values of the regression predictions the best model had a score of 86.06, see table 2 Initially, players predicted positions ranged from -10.96 to 1410.12, which gave an insight to the high RMSE value. When observing the predictions for each tournament, if the data was ordered based on the predictions the outcome was similar to the actual event standings. This led to the development of a work flow to use the initial predictions and attempt to get better results.

The work flow is shown in figure 9.

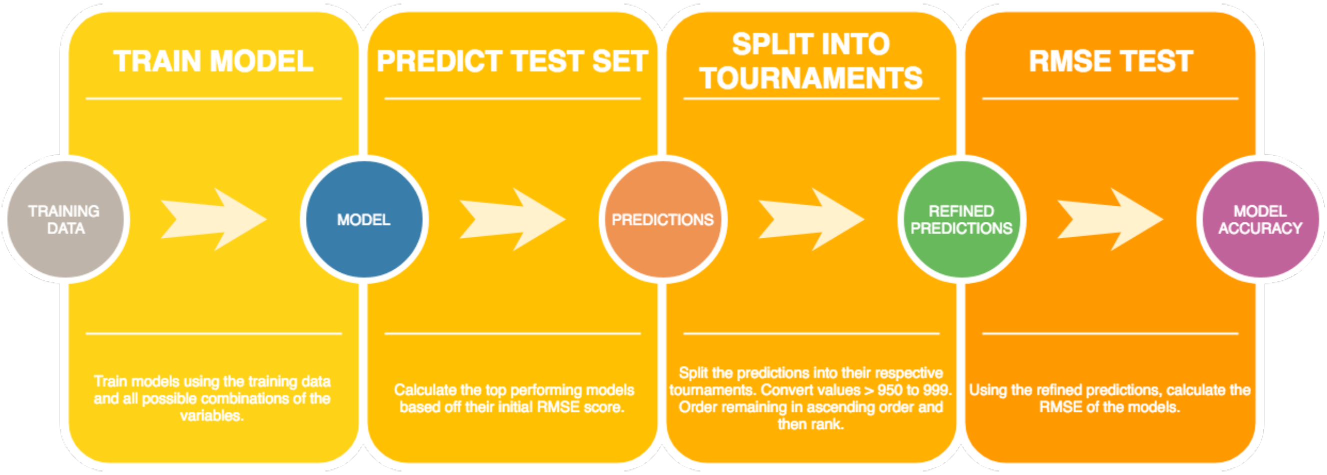

TRAIN MODEL
PREDICT TEST SET
SPLIT INTO TOURNAMENTS
RMSE TEST
TRAINING DATA
MODEL
PREDICTIONS
REFINED PREDICTIONS
MODEL ACCURACY
Train models using the training data and all possible combinations of the variables.
Calculate the top performing models based off their initial RMSE score.
Split the predictions into their respective tournaments. Convert values > 950 to 999. Order remaining in ascending order and then rank.
Using the refined predictions, calculate the RMSE of the models.

Figure 9: *Work-flow for Linear Regression Modelling*

When using the model described in table 2 after the work flow it had an improved RMSE of 68.97.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1,866.379	9.120	204.655	0
Bogey Avoidance Rank	-0.089	0.050	-1.787	0.074
Driving Acc (% Possible Fairways)	0.945	0.354	2.670	0.008
Driving Distance Total Drives	-4.712	2.471	-1.907	0.057
Overall Putting Avg (# of Putts)	2.341	0.113	20.776	0
Pars	-0.187	0.156	-1.200	0.230
Stroke Average (Rank)	0.098	0.050	1.964	0.050
Total Greens in Regulation	-2.506	0.109	-22.975	0
Total Holes Played	-27.661	0.158	-174.760	0

Table 2: *Linear model before step-wise regression*

To have a model that has significant predictors of finishing position, step wise regression was performed on the current linear model to retrieve the subset of current variables that form the optimal model. "Backward elimination step-wise regression" was applied to the model, which entails the following:

- Find variable in model with highest p value > 0.05
- Remove variable from the model
- Refit the model and repeat above steps till all p values < 0.05

After completion of the step-wise regression, our final model had an RMSE value of 59.36, a significant decrease from the original. The models coefficients are shown in table 3.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1,864.757	9.068	205.636	0
Driving Acc (% Possible Fairways)	0.900	0.351	2.562	0.010
Driving Distance Total Drives	−5.247	2.425	−2.164	0.031
Overall Putting Avg (# of Putts)	2.345	0.113	20.831	0
Stroke Average (Rank)	0.009	0.005	2.018	0.044
Total Greens in Regulation	−2.516	0.109	−23.123	0
Total Holes Played	−27.662	0.158	−174.796	0

Table 3: *Linear model after step-wise regression*

To validate if the model would generalise to other data, the model underwent a 10-fold cross validation. This is where the original data is split into 10 sets, then sequentially each set is used as a testing set while the model trains on the remaining nine. The model produced cross validation accuracies of 46.86%, 51.88% to ± 1 place and 56.48% to ± 2 places.

The final model predicted the players' finishing positions with an accuracy of 45.95%, within ± 1 place at 52.57% and ± 2 places at 58.19%. In addition, it successfully predicts players that won't make the cut with an accuracy on 99.9%.

4.1.2 Decision Tree

When modelling the data using a decision tree the rpart R package [?] was used. To test the models all possible combinations of the final variables were used, the best RMSE score obtained was 25.46. This was promising when compared to the previous score from LR as it indicates a better fit between the actual results and the predictions. However when viewing the tree itself it was clear that the DT was not a useful model.

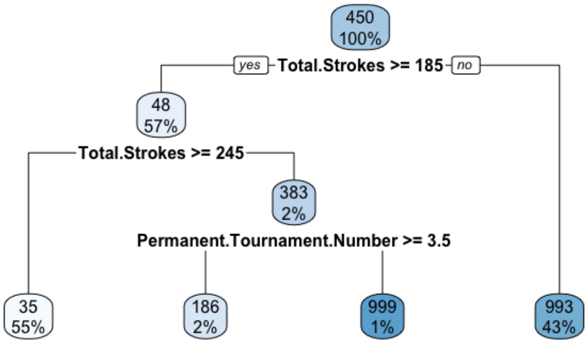


Figure 10: *Decision Tree based on final model*

The tree fails to predict more than five finishing positions, and when viewing the predicted finishing positions based on the test data, only three actual positions were predicted (991.36, 34.96, 427.22). The optimal DT despite having a surprisingly low RMSE would be useless for predicting players finishing positions.

4.1.3 Random Forest

The RF model was trained using all the variables previously chosen from the data exploration. The model consisted of 500 trees and had an RMSE of 38.94. The base accuracy once the predictions had been rounded to 0 decimal places was 40.81%, and 42.27% when given ± 1 place and 47.62% for ± 2 places.

To tune the model to try increase the accuracy, the tuneRF method tries various mtry values to find the one that gives the best out of the box error. Mtry values are the "number of predictors sampled for splitting at each node" [**?**].

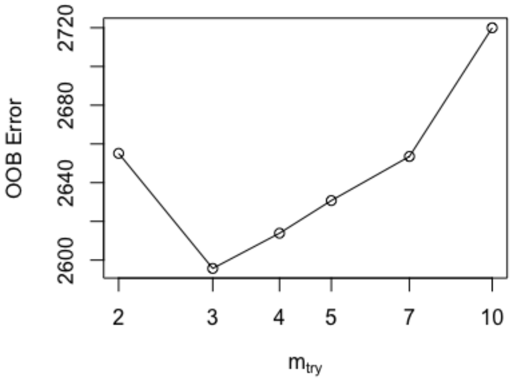


Figure 11: *Optimal mtry value choice*

The best mtry value as shown by the Figure 11 was found to be three. When using the mtry as three, the RMSE decreases to 26.79 but the accuracy dropped to 7.91%. The model failed to predict any finishing position higher than 5^{th} place. The same approach used for linear regression was then taken (see Figure 9). All players who aren't predicted to be cut are ranked in order and those are their finishing positions. This increased the accuracy to 44.76%, 47.53% when given ± 1 place, and 50.27% for ± 2 places. However it also increased the final RMSE to 38.11.

The model was not cross validated as "there is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error. It is estimated internally" [**?**].

4.1.4 Neural Network

When modelling the data using Neural Networks the neuralnet R package [**?**] was used. To test the models all possible combinations of the final variables were used. This also included changing the number of hidden layers and the amount of nodes in each layer, ranging from multiple layers with over 2000 nodes to ones with just 50 nodes. The best RMSE score was 135.07, consisting on one hidden layer with 5 nodes.

During the training stage, the time taken to train each model was significantly larger than the previous methods. To decrease the training time, the training set was cut to only results from the previous year. Once completed the best RMSE score for the model was 25.14, the final model is show in figure 12.

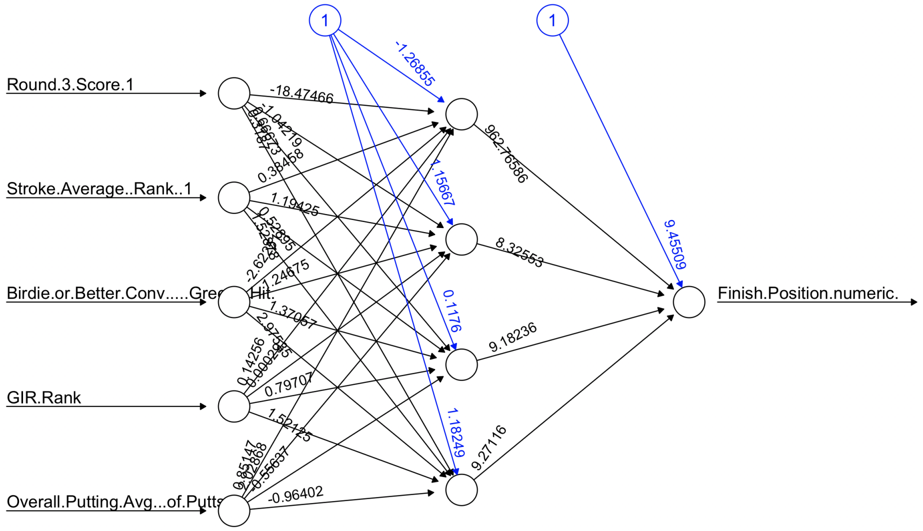


Figure 12: *Final Neural Network Model*

Despite the low RMSE the model had an accuracy of 43.5%, 46.2% when given ± 1 place, and 46.5% for ± 2 places. The results are nearly as accurate as the LM or RF, however the neural network performed similarly to the decision tree in that it only predicted two finishing position values, 999 and 36, rendering it useless for finish position prediction. However, it accurately predicted 99.9% of players who missed the cut and didn't finish the tournament.

4.2 Model Choice

A selection was made based on the results from the previous section shown in these comparison graphs. The graphs show both the accuracy of the models using data from the previous tournament and the RMSE of the models:

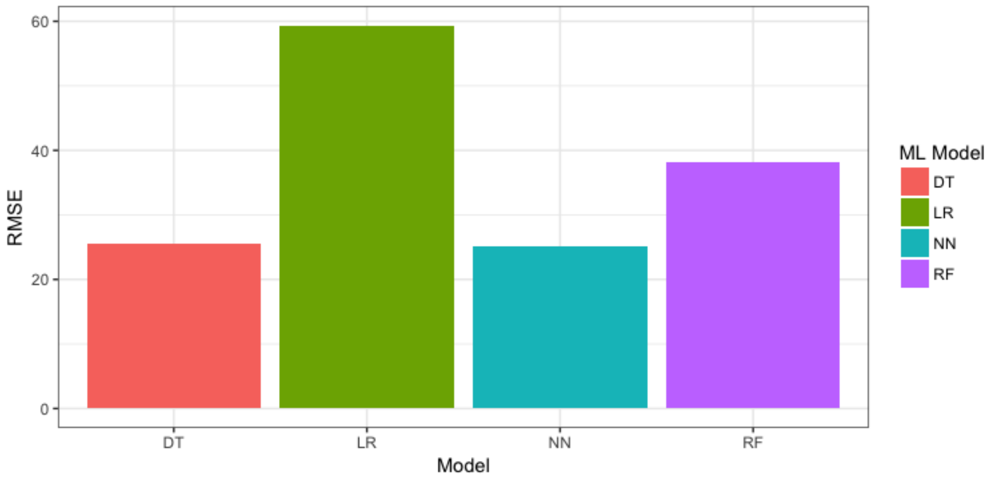


Figure 13: *RMSE Evaluation*

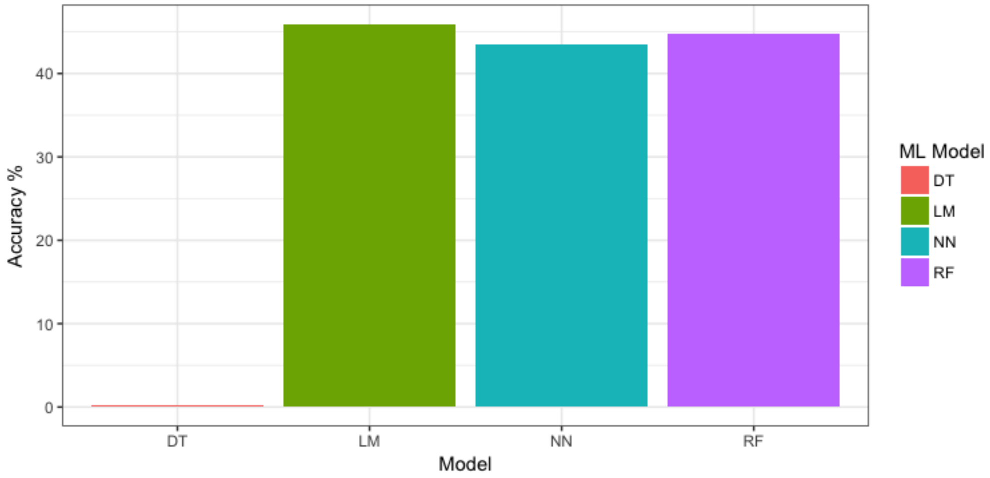


Figure 14: *Accuracy Evaluation*

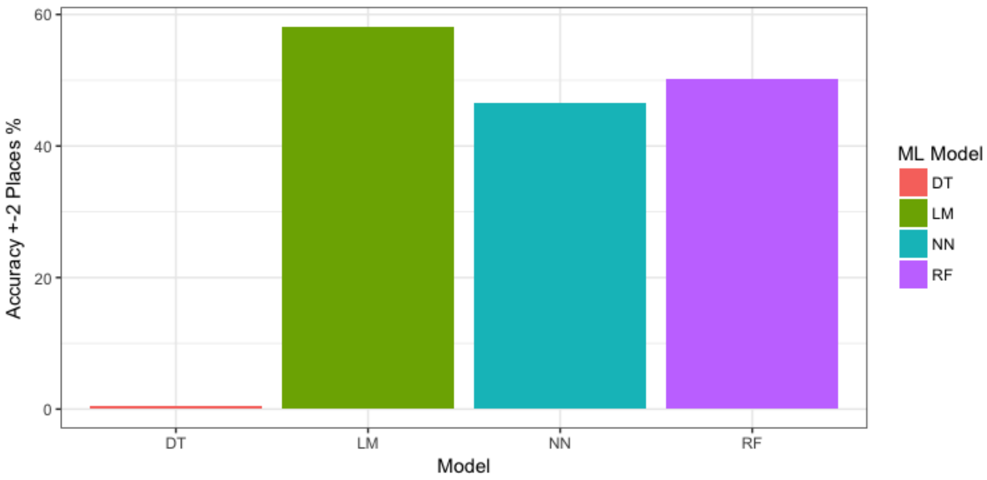


Figure 15: *Accuracy ±2 Evaluation*

For usage with the team optimisation algorithm both the random forest and the linear regression models predictions were opted to be used for the fantasy league team as they had similar accuracies. The linear model is more accurate ±2 places however, so this model is expected to perform better.

5 Team Optimisation

Under the given league constraints, an algorithm needed to be used and tested to produce the highest scoring team possible. The problem is defined as:

$$\begin{aligned} argmax(\sum_{n=1}^{N} points_n) \\ such\ that \sum_{n=1}^{N} n = 8 \\ budget \leq \sum_{n=1}^{N} player\ cost_n \\ transfers\ per\ week \leq 4 \end{aligned} \tag{2}$$

Subject to this problem, algorithms can now be tested. The list of players competing in the tournament, along with their predicted finishing position from the chosen machine learning model will be used to create the most optimal team. Constraints given by Today's Golfer cause this optimisation task to fall under the Knapsack Problem. This problem is described by the following: "Given weights and values of n items, put these items in a knapsack of capacity W to get the maximum total value in the knapsack." [**?**]. In the context of the project, the knapsack is the team, with a capacity of eight. The items are the players, their weights are the fantasy league points and their value is their wage. All while fitting to a constraint of an $80 million budget. The problem can be solved by the use of linear programming, with the help of the lpSolve package [**?**]. To choose the best team there are alternative variables that can be used to pick the team, which will be discussed next.

5.1 Greedy Team Selection

The initial idea was selecting the players predicted to finish in the top positions, so that they score the most points, while hoping they stay under the set budget. This way the players predicted to finish top of the tournament are guaranteed to be selected. However they may take up a large portion of the budget meaning the remaining players may not score as well in comparison.

5.2 Points Per Million Team Selection

Another approach to take to is to find a team based on the amount of points per million (PPM) a player scores. This shows how valuable their points are against their cost. PPM is defined in equation (3).

$$PPM = \frac{Predicted\ Points}{Cost\ (Per\ million)} \tag{3}$$

The team picked this way will most likely be different to the greedily selected team. This selection method however will select players who score the best for what they're worth. Selecting a team this way guarantees that the budget is used in the most cost effective and efficient way.

5.3 Transfers

As only four transfers are able to be made each week, choosing the best players to swap in/out is optimal. The performance of both the greedy and PPM approach, will be the main basis of which algorithm will be used. The pseudo code of the algorithm for transfers each week is as follows:

Algorithm 1 Transfers Team Selection

```
if $week = 1$ then
    $data = getPredictions()$
    $team = getBestTeam(data, 80000000, 8)$ //Choose team of 8 for first event
else
    if $week > 1$ then
        $newTeam = ()$
        $team = getPreviousTeam()$ //Get prev team and upcoming predictions
        $data = getNextWeekPredictions()$
        $recurringPlayers = checkIfCompeting(team, data)$ //Check for players who're competing in next tournament
        if $recurringPlayers.predictions = 999$ then
            $recurringPlayers.remove(cut)$ //Remove players who will score negative points/predicted to be cut
        end if
        if $recurringPlayers.size$ $between$ $1$ $and$ $4$ then
            $newTeam.append(recurringPlayers)$ //Keep the players who're playing again and aren't being cut
            $remainingCount = 4 - newTeam.size$ //Get players who aren't playing in upcoming tournament but are in previous team and did well last week
            $nonRecurring = team - recurringPlayers$
            $toAdd = getPlayers(nonRecurring, remainingCount)$
            $newTeam.append(toAdd)$
        else if $recurringPlayers > 4$ then //Add the top 4 players who're recurring
            $newTeam.append(top4(recurringPlayers))$
        else
            if $recurringPlayers = 0$ then
                $newTeam.append(top4(team))$ //Keep the 4 best from last week as they're likely to play again and do well
            end if
        end if
        $budget = 80,000,000 - newTeam.getWage()$
        $playersNeeded = 8 - newTeam.size()$ //Add remaining players using the knapsack algorithm with the left over budget
        $newPlayers = getBestTeam(data, budget, playersNeeded)$
        $newTeam = newTeam.append(newPlayers)$
    end if
end if
```

The aforementioned algorithm was implemented, run, and picked teams using both of the previous approaches. The results are discussed in the following section.

6 Evaluation: Team Selection Predictions

After testing the four alternative machine learning methods, the LR and RF model produced the most accurate predictions and were then used for choosing the best fantasy league team. The dataset of predictions was put in chronological order based on the tournaments in the fantasy league. Due to the format of the league certain tournaments fall into the same prediction week as others. The rules state that the team selected will be able to score points for all the tournaments taking place that week. Henceforth these groups of tournaments are joined together so the team selection algorithm has to pick the best team from the conjoined dataset. In addition, the Today's Golfer fantasy league consists of tournaments from both the European Tour and the PGA Tour. Due to the scope of this project, only having gained data from the PGA Tour, I can only put forward a team for the PGA events.

Using both methods of team picking (Greedy and PPM) the following graph shows the weekly scores of the team chosen by the algorithm.

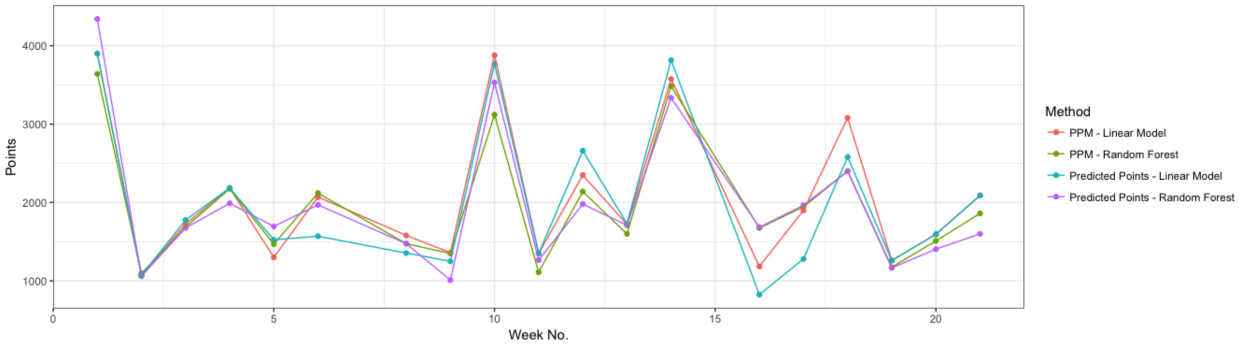


Figure 16: *Points scored per tournament week being competed in*

All the points the scored each week using all approaches follow a similar trend (shown by figure 16).

To place in the Top 20 competitors in the 2017 Today's Golfer league, one would have had to score between 32,805 points $(1^{st} place)$ and 28,200 $(20^{th} place)$. The following graph shows the points scored using both predictions from the linear regression model and random forest.

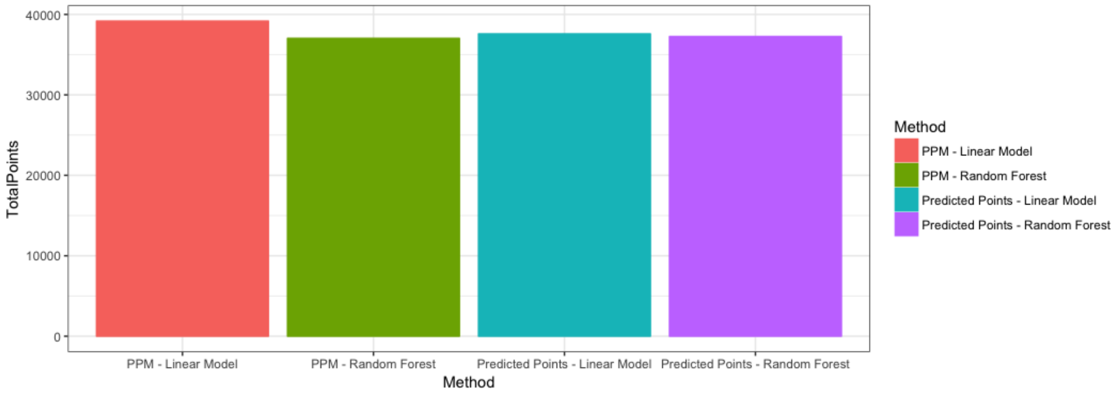
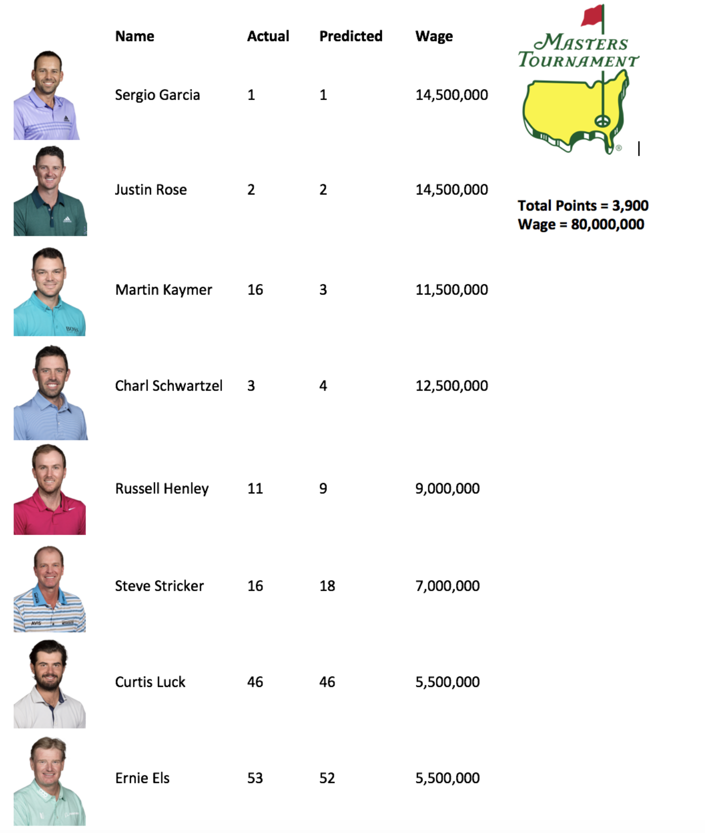


Figure 17: *Points scored per model and approach*

The PPM team selection approach using the predictions from the linear regression model scored 39,175 points, which is 6,370 more points than the winning team. The teams that are selected using the algorithm alongside the predictions, fielded the winning player of the tournament in the team 73.7% of the time, second place 78.9% and third place 57.9%. An example of the team chosen for the 2017 Masters Tournament is shown below:



Name	Actual	Predicted	Wage
Sergio Garcia	1	1	14,500,000
Justin Rose	2	2	14,500,000
Martin Kaymer	16	3	11,500,000
Charl Schwartzel	3	4	12,500,000
Russell Henley	11	9	9,000,000
Steve Stricker	16	18	7,000,000
Curtis Luck	46	46	5,500,000
Ernie Els	53	52	5,500,000

Masters Tournament

Total Points = 3,900
Wage = 80,000,000

Figure 18: *Selected team for The Masters*

7 Discussion

In this project, the models yielded some less than favourable results, their accuracies were particularly low and their RMSE's were high. However, applying the predictions to the fantasy league produced excellent results. Each area this researched focused on had its strengths and weaknesses which are discussed in this section.

Firstly, the ability to have access to this data has been significant in the progress made with the project. If not for the access given by the PGATour and my supervisor Adriane Chapman then this project would have been stuck in the water, as I would've had to find a new source for data.

Secondly, the final results from the models made were not as appealing as one would've liked. However the best model was able to accurately predict to ± 2 places at 58.19% which shows progress. To increase the accuracy of the models there needs to be more in depth variable exploration with relation to previous results and the standings for the tournaments. In addition all the RMSE values calculated for each model were significantly higher than one would've hope for. The lowest being 25.46 for the Decision Tree model which actually performed the worst, producing only three actual position predictions.

Thirdly, as only the results from the single previous tournament were used to produce the models (two - eight failed to perform at the same caliber), they could be improved by using all the results from their previous n tournaments to form a larger matrix. This may increase the accuracy of the models as the average fails to encapsulate the performance each event. For example a player may have played very well for the past two tournaments but the three prior to that played very poorly. An average would hide the success of the better performing tournaments, which in turn could affect the accuracy of the models.

One of the best outcomes of this report is that models (LM and NN) can accurately predict 99.9% of the time who is going to be cut from the tournament, showing that it still has it's uses elsewhere.

Although it didn't have a significant affect on the outcome of the fantasy league, half way through the project the Bunkered league I had planned to use removed all their data on the 2017 season results to make way for the new 2018 season. This was something I hadn't planned for in my risk assessment. This meant a new league was chosen, but due to the limited leagues and their rules, it included both the PGA Tour and the European tour of which I didn't have such rich data for. For this reason I was only able to apply my predictions to 21 tournaments instead of the set 50. In future it would be wiser to write a risk assessment that entails all possible risks, so if they did happen I'd have a better idea of how to react and progress.

Despite this drawback, when using the predictions to select a possible team for each week I could compete in, the algorithm successfully picked teams that in total scored 39,175 points. This was 6,370 points more than the winner of the league who competed in all 50 events. This shows that using the model predictions alongside a fantasy league is extremely beneficial.

The algorithm designed to pick the best possible team was fast and produced reliable and optimal teams. As well as the speed, the algorithm using the PPM approach was able to maximise the number of points scored each week, while sticking to the budget and using the models predictions from each tournament. On the other hand, more time could've been spent on improving the algorithm, looking for players that could've replaced team members that scored higher if there was enough budget and transfers left.

The project had similar aspects as that of paper [**?**], when focusing on the team optimisation. Both leagues have next to identical team selection problems, except for golf there was a maximum number of transfers that could take place. [**?**] had well thought out approaches to base the player selections on which inspired me to do the same. However, Golf fantasy league predictions is significantly different to NFL, which has multiple player positions whereas golf is more focused on the standings at the end of the tournament. Furthermore, my team selection was simulated across the fantasy league tournaments that I was able to predict for, producing a selection of teams that would've won the league, this is something [**?**] was unable to have the chance to do.

Finally, throughout the project I was learning and using R, a language I had previously no experience with. Through tutorials and courses online, I was able to pick up the syntax and gain a vast amount of knowledge on various packages and functions, showing my ability to learn new skills. As well as R, I had to learn to write web scrapers using python to save data from Today's Golfer website, further developing my portfolio of skills.

8 Future Work

After the success of choosing optimal teams for the PGA Tour fantasy league events, I would like to explore the data on the European Tour as well, as this would increase the points scored for each week in the league. Also comparing the models used for the European league and the PGA could be interesting to see, maybe certain variables are more important between the leagues.

The amount of data from previous tournament's supplied to the model could be experimented with to see if variations of the number of previous tournaments would increase the models accuracy.

The models only took into account the variables supplied by ShotLink, to further this work I could explore the effect of external variables such as the weather. Some players may perform better when its wet or when its hot and dry.

It would be useful to improve the models accuracy by exploring previous results of the players deeper and searching for increased positive correlations between the finishing positions and their past performances. This could improve the accuracy of the models and help gain a better insight in what affects the outcomes of golf tournaments. Improving the models accuracies could also lead to increased betting opportunities for companies, as they could predict who would come in certain final positions and update their odds based on that.

Finally, I aim to use the models to compete in the upcoming season fantasy leagues and this year try and win in not just a simulation.

9 Conclusion

In conclusion, looking at the goals that I set at the start of this project and the scope (found in Appendix A/B) the project has met some of the goals but the others need improvement, for example the accuracy of the models.

The main goal of the project was to accurately predict the finishing positions of players for tournaments on the PGA Tour. The best model found was a linear regression model, which predicted this to an accuracy of 49.95% and to within ± 2 places an accuracy of 58.19%. The target of 65% was unfortunately not met, however there is room for improvement and further testing of different variables.

Despite not reaching the accuracy that was intended, the predicted positions were useful when it came to the goal of applying them to the fantasy league and picking the best possible team. The teams that were selected based on the constraints, included the winner of the tournament 68.4% of the time. This led to a score of 39,175 points that my teams selected based off the predictions scored. The score was significantly higher than the winner of the league in 2017 by over 6,000 points. As well as this the teams were selected for only 19 weeks worth of competitions and only 21 PGA Tour tournaments, missing out on an extra 13 weeks and 30 other tournaments. Showing a huge success when applied to the Today's Golfer fantasy league, and completing the goal of coming in the top 10% of players in the league.

Overall, the research demonstrates that the use of linear regression models, and linear programming can predict the standings of PGA tournaments, optimise fantasy golf teams, and beat other players. The simulations shows they would have performed highly when competing in the Today's Golfer league for the 2017 season and could be used to win prizes. Further work could be done to increase the accuracy of these predictions but my models provide a starting point into using machine learning to predict tournament standings within the PGA tour.

10 Acknowledgements

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my thanks to all of them.

I am highly indebted to Age Chapman for their guidance and constant supervision as well as for providing necessary information regarding the project and also helping me with ideas for where my project should turn.

I would like to express my gratitude towards my parents, my girlfriend for her constant support, and the boys for taking playing golf with me when I was struggling for ideas.

I would like to express my special gratitude and thanks to Joseph Lamdin for lending me his guidance and wisdom at my times in need.