SIADS 696 Milestone II Project Report - PGA Tour Data Modelling

Predicting PGA Tour Tournament Results and Player Clustering by Play Style

B. Wilson (benwilso@umich.edu), P. Sklamberg (psklam@umich.edu), J. Doherty (jado@umich.edu)

Introduction

The results of golf tournaments are often highly variable, and unlike some team sports where the better team wins most of the time, an individual sport like golf is not nearly as easy to predict. One week, the best golfer in the world might win, and the next week they might miss the cut completely. The game of golf is nuanced in this way, which makes it very difficult to accurately predict the finishing results of a future tournament. We tackled this problem using machine learning techniques to try and predict PGA tournament outcomes, as well as identify clusters of players on the PGA tour with similar playing styles. The anticipated impact of these machine learning models was to enable prediction of likely outcomes / finishing positions of future PGA tournaments (with potential sports betting applications), and classify new players entering the PGA tour based on their playing style and similarity to current tour players.

Our motivation for this work comes from the fact that all three of us are big golf fans. We all consistently watch golf, have favorite players, and have firmly held beliefs around what we believe makes a good golfer. However, intuition is not always correct, and we are curious if the data supports these beliefs. We are very interested in determining *exactly* the differences between a good, mediocre, and bad golfer, and how we might be able to use player data and finishing results from past tournaments to predict outcomes of future tournaments.

From a supervised perspective, we developed models that attempted prediction of finishing positions of all players in a given PGA tournament, comparing the performance of a multinomial logistic regression model, a random forest classifier, a light Gradient Boosted Machine (GBM), and a simple neural network. Following extensive analysis of these models, our main finding is that all models had relatively weak predictive power over tournament finishing positions, performing slightly better than random assignment; causes for this result are discussed in the *Failure Analysis* section of the report. Pivoting to an alternate, simplified approach, two of these models were modified to perform as binary classifiers, predicting whether or not a given player would make the cut in a specific tournament; this work is discussed in detail in Appendix A.

From an unsupervised perspective, we applied dimensionality reduction techniques (PCA and t-SNE) to some of the most important golfer specific statistics (strokes gained) and clustered those golfers using k-means to determine effectiveness of clustering specific playing styles and performance profiles. After analysis of the resultant clusters, our main finding is that the best players generally have the most strokes gained from tee to green, meaning that they are most efficient at getting the ball from the tee box to the green. We found that t-SNE did a slightly better job of preserving local cluster structure than PCA did and allowed for a better overall visual intra-cluster representation of players with similar strengths.

Related Work

Part of the inspiration for this project was Bill Benter's seminal work on horse racing prediction, where he applied multinomial logistic regression to model the likelihood of horse racing outcomes:

- Benter, W. (1994). *Computer based horse race handicapping and wagering systems: A report.* HK Betting Syndicate.

A modern take on this same work was performed by Timothy and Philip Leung, which reproduced portions on Benter's work, using more contemporary horse racing data:

- Leung, T., & Leung, P. (2023, February 1). *Revisiting the algorithm that changed horse race betting*. Acta Machina. https://actamachina.com/posts/annotated-benter-paper

Finally, as our supervised learning model draws inspiration from this work but adapts it to golf (where player performance is influenced by skill-based parameters), a third source was utilized, which looked at the use of machine learning (ML) models to predict outcomes in LPGA events:

- Chae, J. S. (2021). Victory prediction of Ladies Professional Golf Association tournaments using machine learning. *Journal of Sports Analytics, 7*(1), 1–10. https://doi.org/10.3233/JSA-200373

The techniques utilized in this project are simplified approaches to the cited work and exclude many of the considerations included in them (comparison to predictions from public markets, for example), but nonetheless demonstrate a straightforward first-order approach to the problem of using ML to make predictions in golf.

Note that this project is an extension of work performed by this same team in SIADS 593 - Milestone I, which utilized the same dataset for data manipulation & visualization tasks.

Data Source(s)

The primary data source for this project - PGA Tour Golf Data (2015-2022) can be found on Kaggle (https://www.kaggle.com/datasets/robikscube/pga-tour-golf-data-20152022) and can be downloaded in a comma-separated variable (.csv) format. Key variables included in this dataset and used throughout this project included:

- Tournament finishing position ('finish')
- Total strokes ('strokes')
- Course par ('hole_par')
- Course ('course')
- Date ('date')
- Cut result ('made_cut')
- Multiple strokes gained statistics including Strokes gained putting ('sg_putt'), Strokes gained around the green ('sg_arg'), Strokes gained on the approach ('sg_app'), Strokes gained off the tee ('sg_ott'), Strokes gained too to green ('sg_t2g'), Strokes gained total ('sg_total')

This dataset contains a total of 36,864 individual records, with each record representing the performance of a single player at a specific PGA tournament held between 2015 - 2022.

Feature Engineering

Prior to engineering additional features from the dataset, some pre-processing of the raw data was required. This included:

- Conversion of player names to all upper case for consistency
- Conversion of the 'date' column to datetime
- Cleaning up of the 'finish' column to convert ties (i.e.: 'T32') to a numeric finishing position, and removal of players who did not finish the tournament (i.e.: 'CUT', 'WD')

- Players with a finishing position higher than 80 were removed

Following this data cleaning step, the final set of features were created, and are listed below; specific details on exactly how these features were generated, and how missing data was handled can be found in Appendix A. These features fall into 3 general categories:

1) ***Overall player performance features***
   These features represent how well a player has performed over the totality of the dataset, and include lifetime_avg_strokes_to_par, course_avg_strokes_to_par.
2) ***Player performance recency features***
   These features represent recent performance of a player; a player playing well in 2015 may not be playing as well in 2019, and vice-versa. These features include recent finishing positions (AFP_last5, AFP_last 10) and recent strokes gained stats (SG_last 5).
3) ***Course features***
   These features represent the specific course the tournament is being played at, and include course_difficulty and the actual (one-hot encoded) course.

It is important to note that, in order to avoid issues with data leakage, a non-standard approach to performing the test / train split was used. The canonical approach to selecting tournaments at random for the test and training datasets would either cause data leakage (if recency features were calculated prior to the split), or result in non-meaningful features (if calculation of recency features skipped tournaments at random that had been included in the test dataset). Instead, cross-validation was achieved by instead choosing various points in time (end of 2019 season, end of the 2020 season, and end of the 2021 season) to split the dataset into test and train datasets, with each test dataset consisting of one season's worth of tournaments. This way, calculated recency features would be meaningful (and interpretable), while avoiding any data leakage.

The complete list of engineered features is as follows:

1) player_rest
2) course_avg_strokes_to_par
3) lifetime_avg_strokes_to_par
4) course_difficulty
5) Average finish position - last 5 tournaments (AFP_last5)
6) Average finish position - last 10 tournaments (AFP_last10)
7) Strokes gained last 5 tournaments (SG_last5)
8) Course (one-hot encoded)

Part A. Supervised Learning

Methods description

For the prediction of tournament finishing position in PGA tournaments, four different models were developed, from 3 different model families representing distinct learning paradigms: probabilistic linear models, tree-based & ensemble methods, and neural networks. The same feature set was used when tuning all four models, each of which is described below.

1) Multinomial Logistic Regression

Logistic regression was selected as the baseline model due to the interpretability of the model output (predicted finishing position and associated probabilities), and was the foundation of the horse racing modeling performed by Bill Benter referenced above. The input data was scaled

using StandardScaler, and the output probabilities were used to force rank finishing position (thus precluding any ties) for the tournament. The inverse regularization strength hyperparameter 'C' was tuned over a logarithmic range [0.01, 0.1, 1, 10]. This model served as a good reference point for comparison against the non-linear tree-based and deep learning models.

2) *Random Forest Classifier*

A random forest classifier was chosen in an attempt to capture any non-linear interactions among player, course, and recency features that could be better captured by a tree-based model. This model is also robust to noisy data (as is the case with our dataset). The random forest classifier also does not require any feature scaling or other pre-processing of the data. The number of trees in the forest, represented by the 'n_estimators' hyperparameter, was tuned over [50, 100, 500, 1000, 1500] to find the number of trees yielding the best model performance.

3) *Light Gradient Boosted Machine (GBM)*

In order to improve upon the performance of the random forest classifier, a second tree-based model - a light gradient boosted machine (GBM) - was utilized. This model improves upon the random forest by attempting to minimize error, and as a result should be better able to capture more subtle feature interactions as compared to the random forest classifier. The number of boost rounds (num_boost_rounds = [5, 10, 50, 100, 500]) and learning rate (learning_rate = [0.0005, 0.005, 0.05, 0.5]) hyperparameters were tuned to find the optimal model performance. Like the multinomial logistic regression, the output of the model was force ranked in order to force rank overall finishing position.

4) *Simple Neural Network*

Finally, a simple neural network model was utilized with the goal of capturing complex, non-linear and potentially conditional relationships (recognizing that player performance is often conditional on the specific course being played, for example), beyond those of the previous models. The input data was scaled using StandardScaler, and the hyperparameters of dropout [0.05, 0.1, 0.15, 0.2] and number of training epochs (n_epochs = [25, 50, 100, 200]) were explored to find optimal model performance.

Supervised Evaluation

Overall Results

The two primary metrics used to evaluate these models were as follows:

1) ***Spearman correlation coefficient***
   Spearman's correlation coefficient was chosen as one of the primary evaluation metrics as it is easily interpreted (scores range from -1 for inversely correlated results to +1 for perfectly correlated results). Spearman's correlation coefficient is also well suited for ordinal rankings, such as those that our models are trying to predict, and relatively insensitive to outliers, such that a single bad prediction by the model should not inordinately skew the overall result.

2) ***Mean absolute error (MAE)***
   Mean absolute error (MAE) was chosen as the other primary evaluation metric, as it is also easily interpretable and has an intuitive meaning. In our case, the MAE expresses,

on average, how many positions a given model prediction was off as compared to that player's actual finishing position.

For each of the 4 models evaluated, the results for Spearman's correlation coefficient and MAE are shown in tables #1 and #2, respectively.

Model	***Test / train split #1***	***Test / train split #2***	***Test / train split #3***	***Overall Mean***	***Std. Deviation***
Multinomial logistic regression	0.180	0.234	0.200	0.205	0.0273
Random forest classifier	0.116	0.138	0.114	0.123	0.0133
Light gradient boosted machine (GBM)	0.070	0.136	0.148	0.118	0.0420
Simple neural network	0.180	0.227	0.207	0.205	0.0235

Table #1: Model performance - Spearman's Correlation Coefficient

Model	***Test / train split #1***	***Test / train split #2***	***Test / train split #3***	***Overall Mean***	***Std. Deviation***
Multinomial logistic regression	19.282	19.005	19.707	19.331	0.3535
Random forest classifier	21.963	22.029	22.466	22.153	0.2733
Light gradient boosted machine (GBM)	20.780	20.355	20.385	20.507	0.2372
Simple neural network	18.773	18.519	18.681	18.657	0.1286

Table #2: Model performance - Mean Absolute Error (MAE)

It is important to note that, while the results for both Spearman's correlation coefficient and MAE for these models are not especially impressive, all 4 models do in fact have some predictive

power of overall finishing position. Specifically, in every case Spearman's correlation coefficient > 0, while the correlation coefficient for finishing positions assigned at random would be ≈ 0. Similarly, the MAE for finishing positions assigned at random (1 through 80) would be ≈ 26.66; in every case the calculated MAE for each model was less than this. These results suggest that all 4 models are performing 'slightly better than random assignment'.

This result is evident when plotting actual finishing position vs. predicted finishing position. Figure #1 shows actual finishing position vs. predicted finishing position for the multinomial logistic regression model, and the expected trend from bottom left to top right of the plot is starting to become evident. Similar plots for the other 3 models can be found in Appendix A, and share this same characteristic.

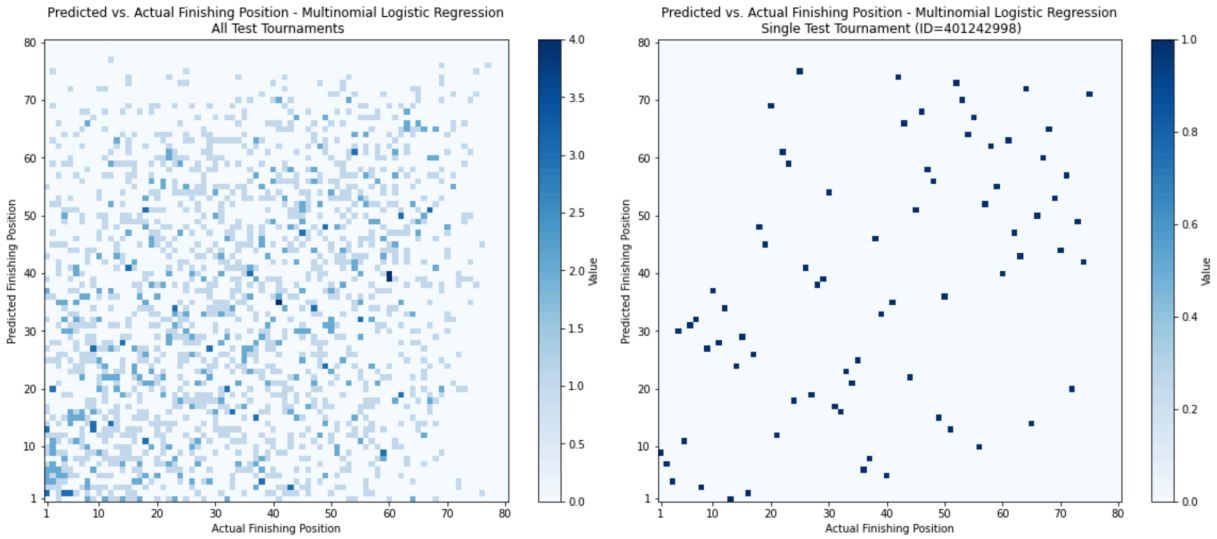

Figure #1: Multinomial logistic regression - all test tournaments and sample test tournament

In-Depth Evaluation

Based on the results discussed above, a more in-depth analysis of the baseline model - multinomial logistic regression - was undertaken to better understand feature importance, and sensitivity to model parameters.

Ablation Analysis

The initial set of features engineered to support model development were as follows, and are described in detail in Appendix A:

1) player_rest
2) Course_avg_strokes_to_par
3) Lifetime_avg_strokes_to_par
4) Course_difficulty (removed)
5) Average finish position - last 5 tournaments (AFP_last5)
6) Average finish position - last 10 tournaments (AFP_last10)
7) Strokes gained last 5 tournaments (SG_last5) (removed)
8) Course (one-hot encoded) (removed)

An ablation analysis was performed to examine model performance following the removal - or ablation - of specific features. This analysis found that the inclusion of some features within the model actually had an adverse impact on model performance (as measured by Spearman's correlation coefficient and MAE), with performance actually improving with the removal of these features. Specifically, 'course_difficulty' and 'SG_last5' were found to adversely affect overall performance. Additionally, the (one-hot encoded) 'course' feature was found to have no influence on model performance, and was therefore also removed in order to reduce overall model complexity. The detailed results of this analysis can be found in Appendix A.

Feature Importance

In order to better understand the importance of specific features and how they contribute to the overall output of the model, a SHAP analysis was performed, which highlights the overall feature importance by showing how each feature influenced individual predictions, and displays the results from all predictions on a single plot; see Figure #2 for the SHAP value plot for the multinomial logistic regression model plot.

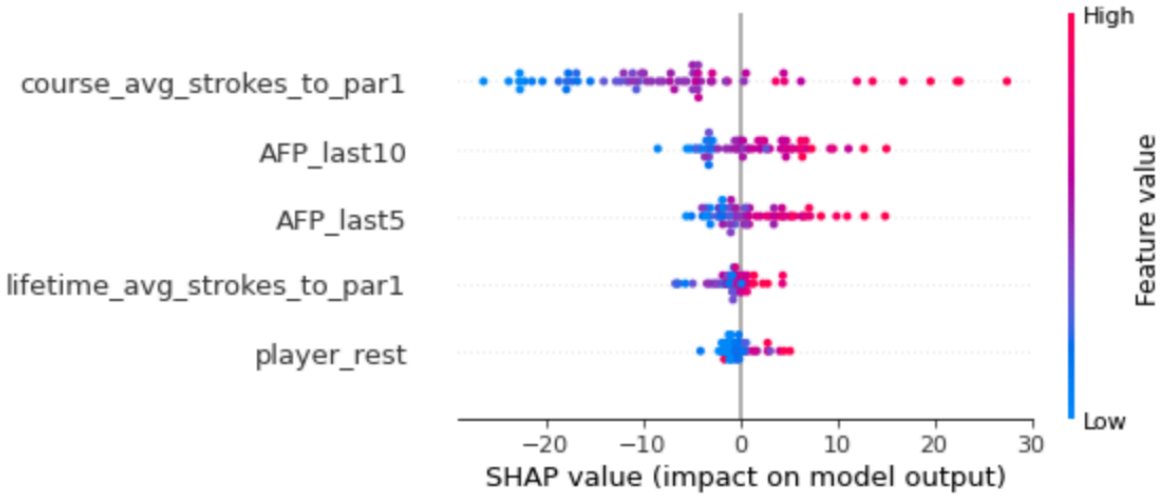

Figure #2: SHAP plot for the multinomial logistic regression model

From this plot, it is evident that higher values for course_average_strokes_to_par, AFP_last5 and AFP_last10 - which would be associated with inferior player performance (lower strokes to par and finishing position would be more desirable than higher values) - have a favorable impact on model output, while lower values of these same features have a slightly negative impact on model performance. The feature player_rest has similar behavior, with higher values (i.e: players being more rested) having a favorable impact on model output, with lower values (i.e.: less rested players) having a negligible impact on model output.

Sensitivity Analysis

In order to understand the influence of the inverse regularization strength 'C' on model performance, a sensitivity analysis for this parameter was performed. As with support vector machines, smaller values of this parameter specify stronger regularization (i.e.: a stronger penalty for overfitting training data). The results of this analysis are shown in Figure #3, and show that model performance improves (larger coefficient of determination) with increasing 'C' (and therefore less regularization).

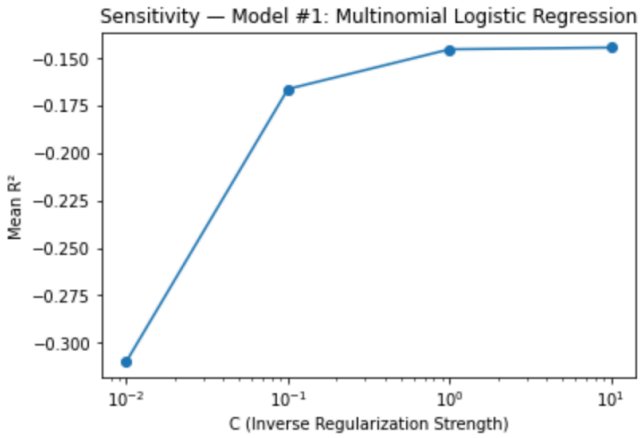

Figure #3: Sensitivity analysis of the inverse regularization strength ('C')

This result makes sense. Owing to the relatively small size of the dataset, it is likely that the 'true signal' in the data is faint, and likely being suppressed by the model regularization; as a result, reducing the regularization penalty (increasing 'C') will improve model performance. A similar improvement in Spearman's correlation coefficient and MAE were observed during hyperparameter tuning during model development.

Tradeoffs

The primary tradeoff identified during evaluation of the multinomial regression model is between model performance (as measured by Spearman's correlation coefficient and MAE) and the size of the training dataset. Examining the values in tables #2 and #3, the worst model performance occurred with test / train split #1, which trained on the smallest dataset (tournaments in 2015 through 2018). Test train splits #2 and #3 both showed significant improvement over these results, and in both cases had larger training datasets. This would suggest that having more training data could potentially improve model performance by improving the amount of signal' in the data.

Failure Analysis

A review of modeling results (and inspection of actual vs. predicted finishing position - see Figure #1) confirms that none of the test tournaments were correctly predicted in their entirety by the multinomial logistic regression model. The failure of the model to achieve even a strong correlation between actual and predicted finishing position can be attributed to three (3) main factors.

1) ***Feature insufficiency***
   The feature set used to train the model was limited to five (5) features, none of which had a strictly positive influence on model output over the entire range of feature values (see Feature Importance above and SHAP plot above). The complex nature of the relationship between features is not fully represented by these 5 features, thereby limiting the predictive power of the model.
2) ***Size of available training data***
   The dataset selected for this project covers PGA tournaments 2015 through 2022. With an average of ~30 tournaments per year in the dataset, the model has relatively few training examples for each finishing position. For example, for the CV test / train split #1 above, the training data is limited to tournaments 2015 through 2018; as a result, the model only has ~120 training examples for each finishing position. With so few training

examples, and a particularly noisy dataset, any model, regardless of type, will struggle to separate 'true signal' from noise.

3) ***Inherent unpredictability of golf***
As discussed in the introduction to this report, golf is an especially unpredictable sport, with individual players having potentially wild swings in performance from one week to the next. This leads to 'noisy' training data, which limits the utility of recency features such as AFP_last5 and AFP_last10 in predicting player performance.

Feature insufficiency could be addressed through enhanced feature engineering and identifying additional features that are less noisy and potentially have a more uniformly positive influence on model output. This would also likely require expanding the dataset to include more raw features. A larger dataset which included more tournaments could also potentially improve model performance by improving the available 'signal' and therefore better fitting the data.

Part B. Unsupervised Learning

Methods description
For the unsupervised portion of our project, we set out to determine which PGA tour golfers are most similar and which are most different by clustering them together based on their average strokes gained stats from 2015-2022. The individual strokes gained stats, including Strokes Gained Putting, Strokes Gained Around the Green, Strokes Gained Approach, Strokes Gained Off The Tee, and Strokes Gained Tee to Green, measure how golfers perform relative to their competitors in different aspects of the game. If a golfer has positive strokes gained stats, it means that they perform relatively better, while negative strokes gained stats mean they perform relatively worse.

For this clustering exercise, we decided to explore two different unsupervised methods: PCA and t-SNE. Below is a table which explains the differences in algorithm, projection, and approach type for these two methods.

	PCA	t-SNE
Algorithm Type	Deterministic	Stochastic
Projection Type	Linear	Non-Linear
Approach Type	Global	Local

Figure #4: PCA and t-SNE Descriptions

1.) PCA (Principal Component Analysis)

We chose to first use PCA for this analysis because of its ability to simply take the overarching strokes gained stats and reduce them into a lower dimensional space which could be easily understood and graphed visually. PCA also has the inherent ability to eliminate multicollinearity. In our case, Strokes Gained Tee to Green is correlated with some of the other strokes gained stats used in this clustering exercise. Eliminating the multicollinearity of these stats using PCA will help us achieve more accurate clustering results overall.

2.) t-SNE (t-Distributed Stochastic Neighbor Embedding)

The second method we chose to use was t-SNE, as the algorithm, projection, and approach are all different from PCA. One of the main benefits of t-SNE over PCA is its ability to project non-linear relationships between features into a lower dimensional space. T-SNE also does a much better job of preserving local clusters while trading off accuracy of inter-cluster distances. An important part of our analysis will include how similar golfers from individual clusters are to one another, which t-SNE will help visualize much more effectively than PCA.

Parameter Considerations and Quantitative Results

When generating the clusters of golfers based off of PCA and t-SNE, there were a few considerations and decisions we needed to make, the most important of which was the imputation strategy. In total, 19% of the strokes gained stats were missing. While this is a somewhat significant portion of the dataset, we decided to use mean imputation for one key reason. When there is lots of missing data, mean imputation can be prone to outliers and create more variance than would be expected. However, the game of golf has a lot of inherent variance. Some weeks, the best golfers play very well, but the next week very poorly. If we imputed the median values for all strokes gained stats, our analysis would be less sensitive to the unpredictable nature of the game of golf. We felt that mean imputation would better capture this variance and give us a better representation of a golfer's overall performance.

After running both PCA and t-SNE on the six strokes gained stats (Strokes Gained Putting, Strokes Gained Around the Green, Strokes Gained Off the Tee, Strokes Gained Approach, Strokes Gained Tee to Green, and Strokes Gained Total) for all of the golfers who played in more than 30 tournaments between 2015 and 2022, we used K-Means Clustering to identify specific clusters within these dimensionally reduced datasets. Since there are 5 main strokes gained stats, we decided to create 5 different clusters to both identify golfers who are of a similar caliber, but also create clusters where players have similar playstyles.

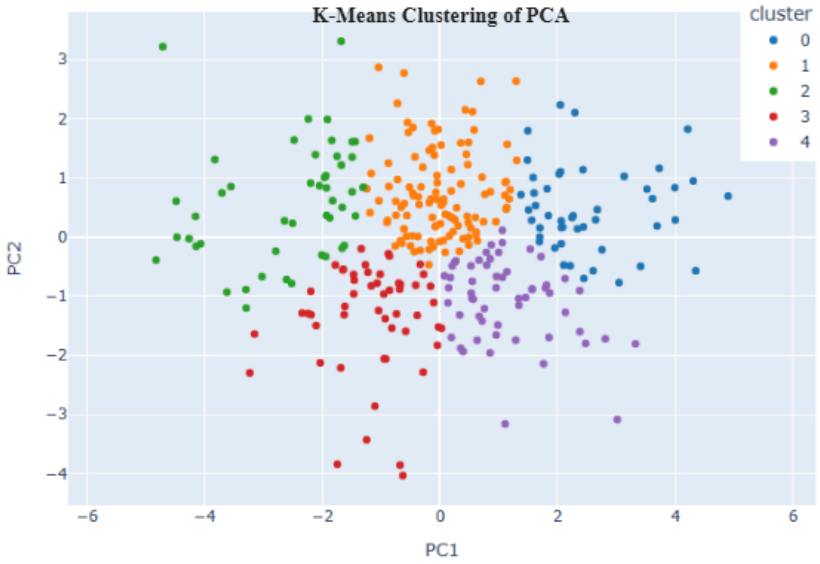

Figure #5: K-Means Clustering of PCA

Figure #6: K-Means Clustering of t-SNE

Figure 5 and Figure 6 above show the K-Means Clustering outputs for both PCA and t-SNE respectively. For t-SNE, we can see a much more sparse global cluster structure, while individual golfers within clusters are more densely clustered. On the other hand, PCA gives us a

much better global structure showing the overall differences between golfers (including outliers) but not necessarily giving us the best local clusters.

Success Metrics

To measure the success of PCA, we calculated the L2 reconstruction error for each player telling us how accurate our reconstructed data would be if we mapped it back to the original dimensional space. Using mean imputation, five clusters, and two principal components, we achieved an L2 reconstruction error of 0.293. Since the scale of our data is generally between -0.5 and 1.5, this reconstruction error is mediocre.

To measure the success of t-SNE, we iterated through a range of values for perplexity (which defines how the model should balance local and global data structures by defining the number of neighbors each point considers) and a range of values for the number of neighbors. At each point, we calculated the trustworthiness score, which calculates how well the local structure of the high dimensional data is preserved when dimensionality reduction is applied. Using this approach, we determined that considering the 5 nearest neighbors with a perplexity value of 10 gave us both consistent local clusters and a trustworthiness score of 0.986 meaning that almost 99% of the time, the 5 nearest neighbors for each point in the high dimensional space were also the 5 nearest neighbors in the dimensionally reduced space. For comparison, if we used a perplexity value of 5 and considered the 20 nearest neighbors, our trustworthiness score would drop to 0.926.

Qualitative Analysis

However, we also wanted to take a look at the results qualitatively. Based on quantitative results, we know that t-SNE does a good job at preserving cluster accuracy. Qualitatively, we found that the clusters formed by PCA actually perform quite well. Even though this dataset only includes data from tournaments played between 2015 and 2022, let's look at the current number 1 ranked player in the world, Scottie Scheffler, as an example.

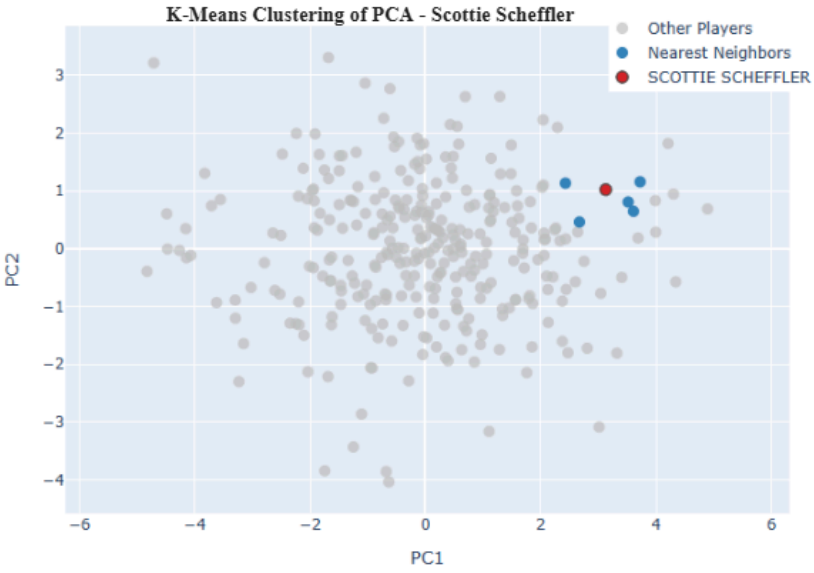

Figure #7: Player Specific PCA

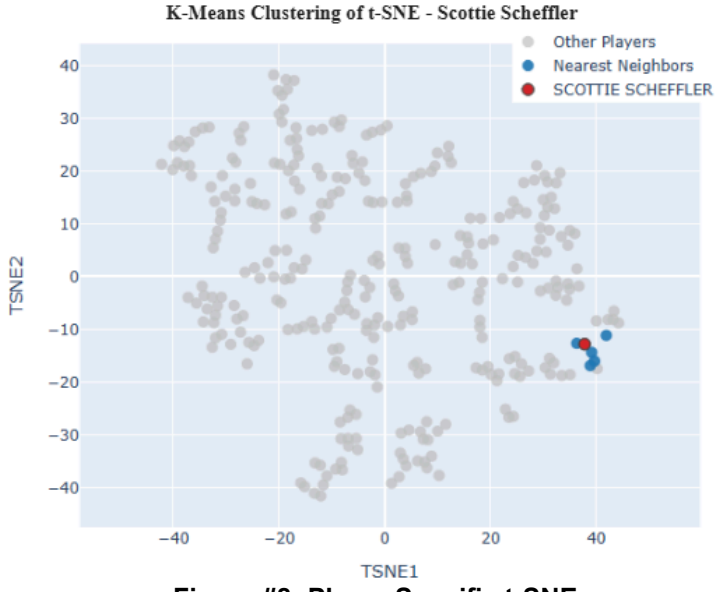

Figure #8: Player Specific t-SNE

Figures 7 and 8 above show where Scottie Scheffler sits in both the PCA and t-SNE clustering outputs respectively. In both instances, he is in the top performing cluster based on strokes gained stats, and his five nearest neighbors are all great players. For the clusters based on

PCA, his five nearest neighbors are Bryson DeChambeau, Henrik Stenson, Hideki Matsuyama, Patrick Cantlay, and Xander Schauffele. For the clusters based on t-SNE, his five nearest neighbors are Dustin Johnson, Jon Rahm, Justin Thomas, Patrick Cantlay, and Tony Finau. Even though only one of Scottie Scheffler's five nearest neighbors are the same across approaches, all of these golfers were ranked in the top during the 2015-2022 period. These 9 unique golfers have won a total of 12 major championships over their careers, while Scottie himself has won 4.

We wanted to further evaluate the clusters by comparing the average of the strokes gained stats for golfers included in each. Figures 9 and 10 below show average strokes gained for players in each cluster, and we can see that both methods give us a clear "best" cluster and a clear "worst" cluster based on sg_total which measures how a golfer performed overall relative to all other golfers.

Average SG Stats per t-SNE Cluster

tsne_cluster	sg_putt	sg_arg	sg_ott	sg_t2g	sg_app	sg_total
0	0.09	0.12	0.35	0.97	0.5	1.06
1	0.38	0.15	0.01	0.46	0.3	0.85
2	0.26	0.03	-0.15	-0.2	-0.08	0.06
3	0.08	0.08	0.17	0.44	0.19	0.53
4	0.5	0.07	-0.07	0.05	0.06	0.55

Figure #9: SG per t-SNE Cluster

Average SG Stats per PCA Cluster

pca_cluster	sg_putt	sg_arg	sg_ott	sg_t2g	sg_app	sg_total
0	0.13	0.17	0.36	1.07	0.55	1.21
1	0.32	0.18	0	0.42	0.24	0.75
2	0.51	0.15	-0.25	-0.25	-0.15	0.26
3	0.17	-0.09	0.09	0.05	0.05	0.22
4	0.02	-0.02	0.31	0.66	0.37	0.69

Figure #10: SG per PCA Cluster

The cluster containing the best players for both PCA and t-SNE is cluster 0, while the cluster containing the worst players for PCA and t-SNE are cluster 3 and cluster 2 respectively. Strokes Gained Tee to Green (t2g) tells us a great deal about how these golfers are achieving their success. This stat measures how efficiently a golfer gets the ball from the tee box to the green.

T-SNE tells us that the golfers from the best cluster gain, on average, 0.97 strokes on the field t2g for each round of golf they play, whereas golfers from the worst cluster lose 0.2 strokes to the field t2g for each round of golf they play. This means that over the span of a four round tournament, golfers from the best cluster would gain almost 4 strokes on the field t2g, while golfers from the worst cluster would lose almost 1 stroke to the field t2g.

PCA tells us that golfers from the best cluster gain, on average, 1.07 strokes on the field t2g for each round of golf they play, whereas golfers from the worst cluster lose 0.25 strokes to the field t2g for each round of golf they play. This means that over the span of a four round tournament, golfers from the best cluster would gain over 4 strokes on the field t2g, while golfers from the worst cluster would lose 1 stroke to the field t2g.

Qualitatively, this follows what we would expect to see as all of the best players are great "ball-strikers" meaning they have great control and distance over both their tee shots and approach shots. It also shows that while the clusters created using both dimensionality reduction techniques differ in terms of the players they group into clusters, they both do a good job in differentiating the best players from the worst.

Finally, we wanted to compare how similar the best and worst clusters were from each dimensionality reduction technique at a high level. To do this, we calculated the Jaccard Similarity between PCA cluster 0 and t-SNE cluster 0 (the best players) as well as PCA cluster 2 and t-SNE cluster 3 (the worst players).

pca_cluster	tsne_cluster	jacc_sim
0	0	0.603
3	2	0.372

Figure #11: Jacc Sim of Best and Worst Clusters

Figure 11 above shows that for the clusters containing the best golfers, there was a Jaccard Similarity of 0.603, which means that ~60% of the unique golfers seen in the union of both clusters were also seen in the intersection of those clusters. In a similar vein, the clusters containing the worst golfers had a Jaccard Similarity of 0.372, meaning that there was only ~37% overlap between the two clusters. From this, we can conclude that these two dimensionality reduction techniques "agreed" more often on who the best golfers were than they did for the worst golfers.

Discussion

Part A. Supervised Learning

The most surprising aspect of the results from this portion of the project was the overall lack of predictive power among all of the models that were developed. Going into the project, the team had a much higher expectation for the ability of the models to predict overall finishing position.

There were 2 main challenges the team encountered. The first of these was related to feature engineering. There was a desire to create some features that captured recent player performance; these features included average finishing position over the last 5 tournaments (AFP_last 5) average finishing position over the last 10 tournaments (AFP_last10), and total strokes gained over the last 5 tournaments (SG_last5); the team referred to these as recency features. The canonical approach to selecting tournaments at random for the test and training datasets would either cause data leakage (if recency features were calculated prior to the split), or result in non-meaningful features (if calculation of recency features skipped tournaments at random that had been included in the test dataset). To address this conundrum, test / train splits were made by choosing various points in time (end of 2019 season, end of the 2020 season, and end of the 2021 season) to split the dataset into test and train datasets, with each test dataset consisting of one season's worth of tournaments. This way, calculated recency features would be meaningful (and interpretable), while avoiding any data leakage.

The second challenge related to the poor predictive performance of the models. As previously discussed, the performance of the supervised models was found to be 'slightly better than random assignment'. In an effort to find patterns in the data with more 'true signal', a second pair of classifiers were developed to predict whether a player would make the cut for a particular tournament. These models performed significantly better in predicting whether a player would make the cut; this work is described in detail in Appendix A.

With additional time and resources, the project team would look to supplement the existing training data with additional training data (representing more tournament results on which to train), and develop new features that could be incorporated into our models. Taken together, both of these would likely increase the predictive power of the supervised models.

Part B. Unsupervised Learning

The most surprising result from this portion of the project was the lack of separation between each of the clusters. Our assumption was that we would generate easily identifiable clusters for golfers who are great putters, great drivers of the golf ball, and great overall. However, in practice these golfers seem to be much more similar to one another than we would have expected. After seeing these results, we considered that the game of golf has extremely high levels of variance. Some of the best golfers can play very poorly in certain weeks, and vice versa. Some players who generally putt very well can putt poorly one week, but play great otherwise. We learned that all of the PGA Tour golfers included in this analysis, while different, are much more similarly matched than we would have imagined.

The main challenge we encountered for the unsupervised learning section was the amount of missing data for the features that we included for use in our dimensionality reduction techniques. We had to decide on the best way to impute the missing values, but this imputation might have caused a reduction in the accuracy of our clusters. We ended up deciding to use mean imputation for both of the techniques as it allowed us a simple, yet inclusive way to calculate the features values for each golfer.

We also ran into challenges with balancing qualitative and quantitative measures of accuracy and success for our clusters. Quantitatively, t-SNE ended up performing significantly better than PCA, but Qualitatively we observed reasonably similar performance. While not explicitly required, we ended up including a qualitative section which described the success of both methods, how they were similar, and how they were different.

If we had more time and resources, we would explore other means of both dimensionality reduction and clustering techniques. While PCA and t-SNE along with K-Means Clustering gave us sufficient results for the purposes of our analysis, there are many intricacies that might have been better identified using other methods. We would also likely have taken a more nuanced approach to sensitivity analysis at scale for both approaches to better understand how changes in specific hyperparameters caused changes in the local and global cluster structures.

Ethical Considerations

As the PGA is overwhelmingly dominated by male players, our team considered the ethical aspects of our topic, and specifically the concern of perpetuating gender-related stereotypes in sports. To mitigate this, we researched PGA rules relating to participation in tournaments, and found that there are in fact no gender requirements to participate in PGA tour events. In fact, women have participated in PGA tour events as recently as 2023.

Similarly, while the PGA has some history of limiting participation based on race, such requirements were officially eliminated in 1961, significantly pre-dating the data to be used in our analyses.

Finally, our dataset contains no data of a personal and potentially sensitive nature for the players, other than their name and national origin, with all fields being otherwise limited to in-game technical performance.

Based on these considerations, our team concluded there were no significant ethical considerations with our choice of data or modeling methods.

Statement of Work (1 point)

Each member of the team made outstanding contributions to this report. We worked as a team to ensure that workload was split up equally and the report sounds like a single voice. However, each team member contributed individually to the work which which is listed below:

Peter Sklamberg

- Unsupervised Learning Section including PCA, t-SNE, Hyperparameter Tuning, Accuracy and Sensitivity Analysis, Cluster Visualizations, Qualitative Analysis.

James Doherty

- Supervised model development & metric selection including creation of the Multinomial logistic regression model, Random forest classifier, Light GBM model, Simple neural network and all related tuning, accuracy, and evaluative work.

Ben Wilson

- Supervised model development & visualizations including creation of the Multinomial logistic regression model, Random forest classifier and related visualizations/tuning, accuracy, and evaluative work as well as creation of the Github repository.

References

- Benter, W. (1994). *Computer based horse race handicapping and wagering systems: A report.* HK Betting Syndicate.

- Leung, T., & Leung, P. (2023, February 1). *Revisiting the algorithm that changed horse race betting*. Acta Machina. https://actamachina.com/posts/annotated-benter-paper

- Chae, J. S. (2021). Victory prediction of Ladies Professional Golf Association tournaments using machine learning. *Journal of Sports Analytics, 7*(1), 1–10. https://doi.org/10.3233/JSA-200373

Appendix A

Detail - Feature Engineering
Provided below is some additional detail relating to engineered features used in Part A of this report.

1. ***player_rest***
   Player rest was calculated as the number of days that had elapsed since that player had previously competed in a tournament. In instances where a player was appearing in the dataset for the first time (and thus it was unknown how much rest that player had), a default value of 28 days was assigned.

2. ***course_avg_strokes_to_par***
   This feature was calculated as the average strokes to par for a given player at a particular course. In instances where a player is playing a particular time for the first time in the test dataset, a default of course_difficulty (see definition below) was assigned for that player.

3. ***lifetime_avg_strokes_to_par***
   This feature was calculated as the average strokes to par of a given player over the entire training dataset, inclusive of all courses.

4. ***course_difficulty***
   Course difficulty was calculated as the average strokes to par of all players who had played that course, over the entire training dataset. Courses that appear for the first time in the test data are removed.

5. ***Average finish position - last 5 tournaments (AFP_last5)***
   This feature was calculated as the average finishing position for a player over their *previous* 5 tournaments (not inclusive of the current tournament). Entries where there was no value (i.e.: that player's first tournament appearance) were removed from the dataset.

6. ***Average finish position - last 10 tournaments (AFP_last10)***
   This feature was calculated as the average finishing position for a player over their *previous* 10 tournaments (not inclusive of the current tournament). Entries where there was no value (i.e.: that player's first tournament appearance) were removed from the dataset.

7. ***Strokes gained last 5 tournaments (SG_last5)***
   This feature was calculated as the average total strokes gained stat for a given player over their *previous* 5 tournaments. Entries where there was no value (i.e.: that player's first tournament appearance) were removed from the dataset.

8. ***Course (one-hot encoded)***
   This 'course' feature in the dataset was one-hot encoded such that this data could be utilized by all the models being evaluated.

Detail - Actual vs. Predicted Finishing Position for Supervised Models

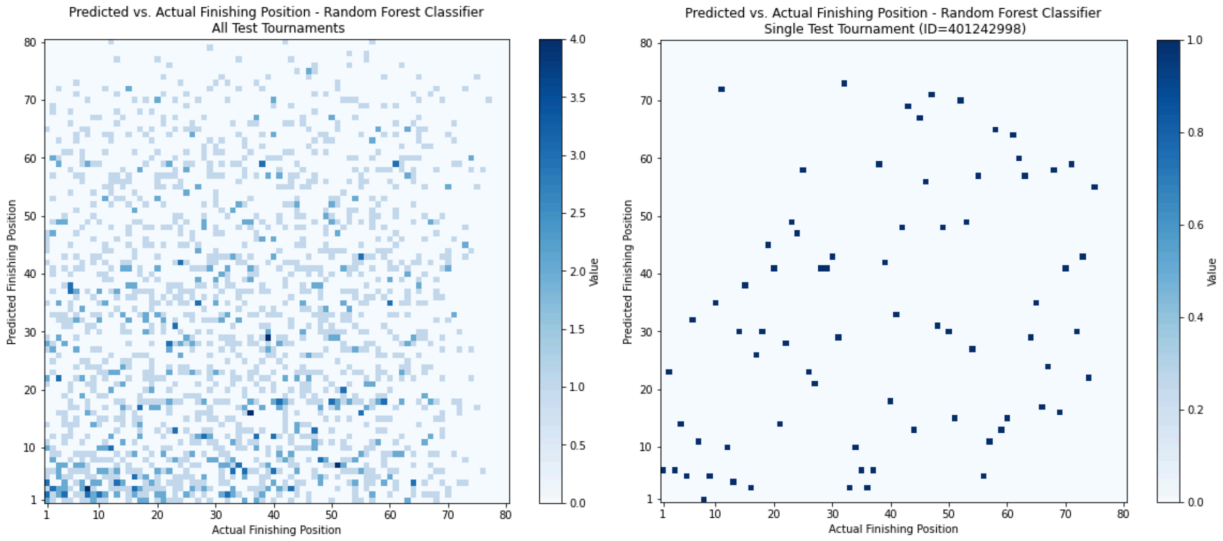

Figure 1A: Random forest classifier results - all test tournaments and sample test tournament

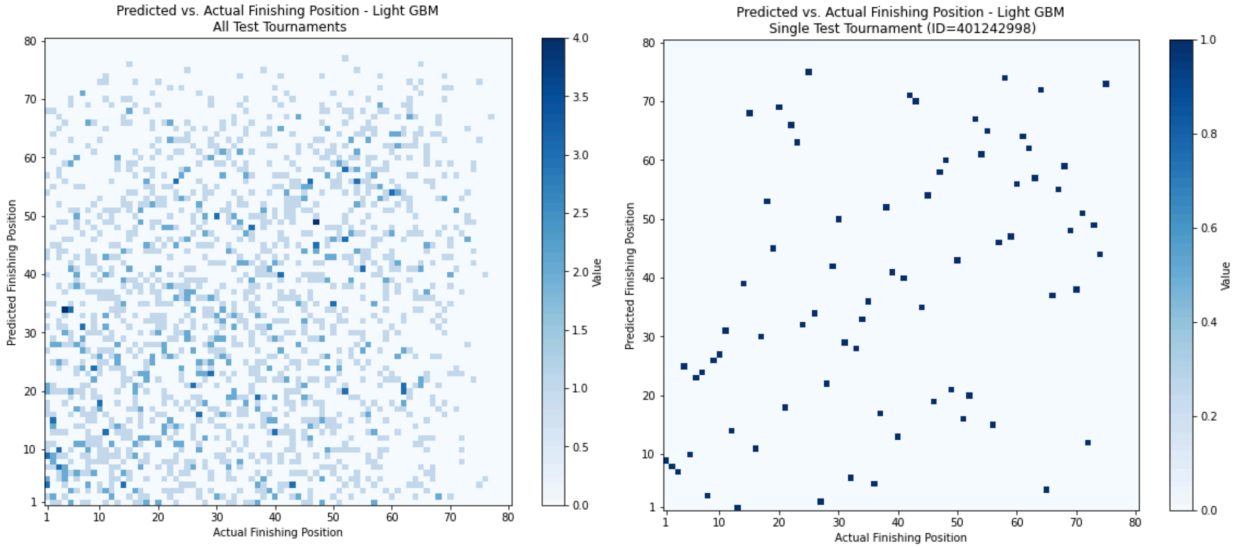

Figure 2A: Light GBM results - all test tournaments and sample test tournament

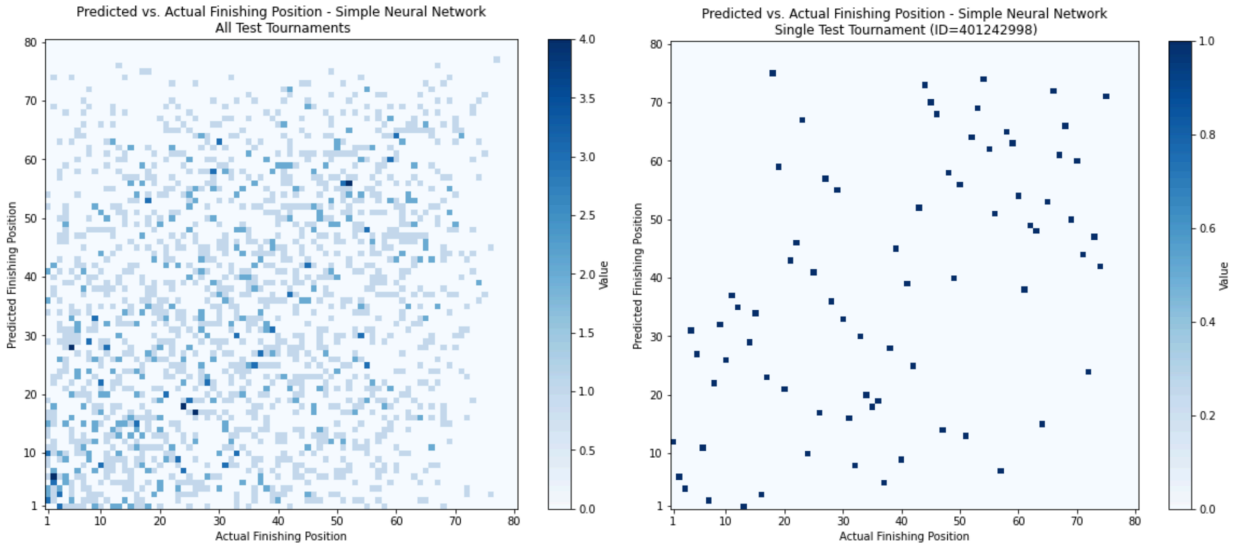

Figure 3A: Simple neural network results - all test tournaments and sample test tournament

Detail - Ablation Analysis Results

Feature Set	***Spearman Correlation***	***Mean Absolute Error***	***Notes***
Baseline (all features): - player_rest - course_avg_strokes_to_par - lifetime_avg_strokes_to_par - course_difficulty - AFP_last5 - AFP_last10 - SG_last5 - Courses (one-hot encoded)	0.1283	20.5605	Baseline performance using all available features.
Features used: - player_rest - course_avg_strokes_to_par - lifetime_avg_strokes_to_par - ~~course_difficulty~~ - AFP_last5 - AFP_last10 - SG_last5 - Courses (one-hot encoded)	0.1978	19.7674	Performance improvement.
Features used: - player_rest - course_avg_strokes_to_par - lifetime_avg_strokes_to_par - ~~course_difficulty~~	0.1998	19.7069	Performance improvement.

- AFP_last5 - AFP_last10 - ~~SG_last5~~ - Courses (one-hot encoded)			
Features used: - player_rest - course_avg_strokes_to_par - lifetime_avg_strokes_to_par - ~~course_difficulty~~ - AFP_last5 - AFP_last10 - ~~SG_last5~~ - ~~Courses (one-hot encoded)~~	0.1998	19.7069	No change in performance; removal of course encoding reduces model complexity.
Features used: - ~~player_rest~~ - course_avg_strokes_to_par - lifetime_avg_strokes_to_par - ~~course_difficulty~~ - AFP_last5 - AFP_last10 - ~~SG_last5~~ - ~~Courses (one-hot encoded)~~	0.1956	19.7890	Performance reduction.
Features used: - player_rest - course_avg_strokes_to_par - lifetime_avg_strokes_to_par - ~~course_difficulty~~ - AFP_last5 - ~~AFP_last10~~ - ~~SG_last5~~ - ~~Courses (one-hot encoded)~~	0.1905	19.8385	Performance reduction.
Features used: - player_rest - course_avg_strokes_to_par - lifetime_avg_strokes_to_par - ~~course_difficulty~~ - ~~AFP_last5~~ - AFP_last10 - ~~SG_last5~~ - ~~Courses (one-hot encoded)~~	0.1980	19.795	Performance reduction.

Table 1A: Ablation analysis results

Detail - Binary Classifier Performance (Cut / Made the Cut)

Following the development of the initial models for predicting finishing position, the best 2 performing models were modified to predict the binary target of whether or not a given player will make the cut of a specific tournament. Initial versions of these models (further tuning of hyperparameters was not performed) were found to be significantly better performing at this task; see results below.

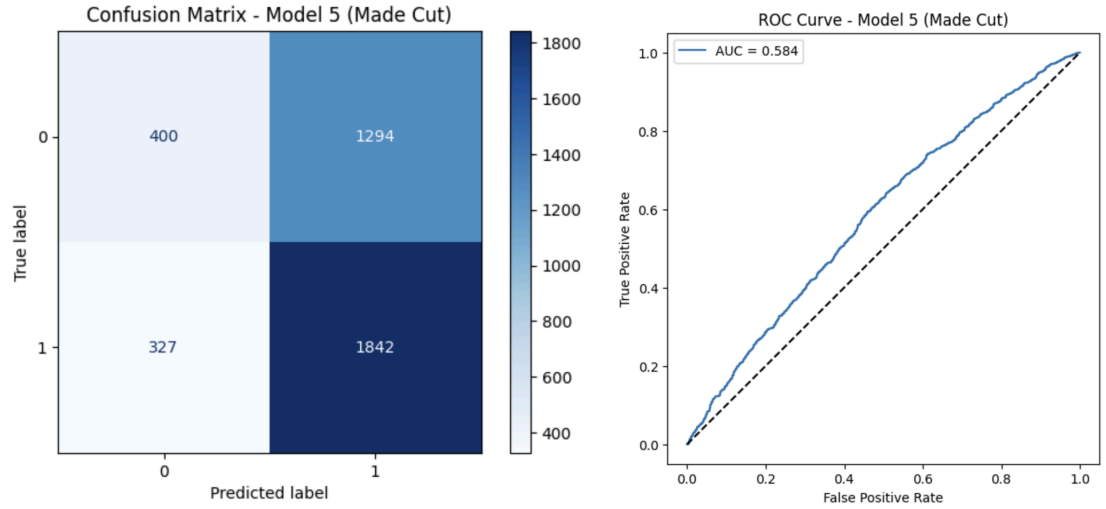

Table 4A: Logistic regression binary classification results

Average Precision: 0.628998862138165
Average Recall: 0.8514979923544082
Average f1 Score: 0.7145765571400606

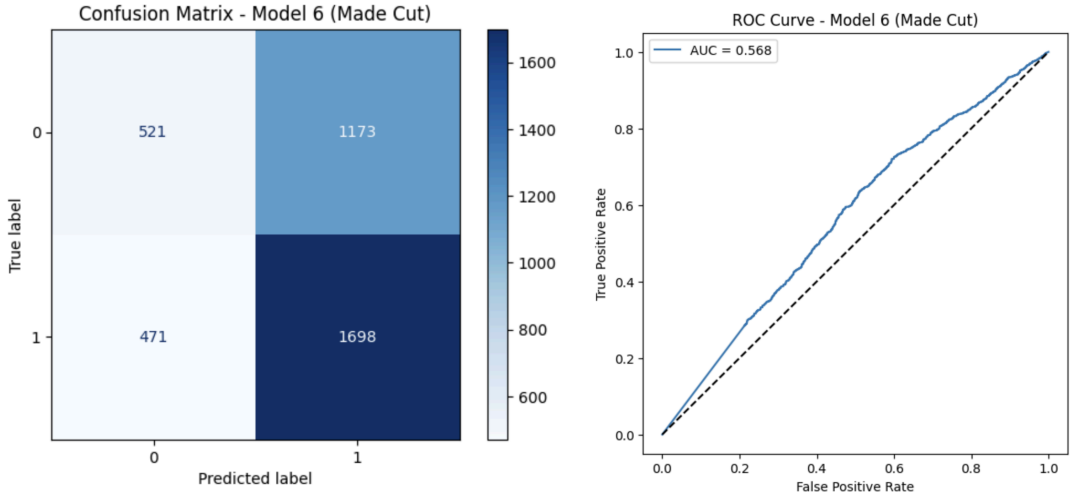

Table 5A: Simple neural network binary classification results

Average Precision: 0.6302637951845135
Average Recall: 0.7908150622211583
Average f1 Score: 0.693980519768441