Using Machine Learning to Predict Professional Golf Performance

Submitted in partial fulfillment of the requirement for MATH 4F90 Honours Project

Bachelor of Science in Mathematics and Statistics

Department of Mathematics and Statistics

Faculty of Mathematics and Science Brock University

St. Catharines, Ontario

Brad Klassen

bk15jd@brocku.ca

Prof. William Marshall

December 17, 2019

Contents

1 Introduction

In the age of big data, the field of sports analytics is becoming increasingly transdisciplinary, combining domain specific knowledge from sports management with the statistical and computational tools of data science. Golf is a sport that generates massive amounts of data with no shortage of opportunity for analysis. As statistics becomes more integrated into professional sports, having an analytical edge gives both athletes and teams a competitive advantage. In fact, many PGA Tour champions have credited their success to analysts that give insights on how to succeed at upcoming tournaments. The goal of the project is to use machine learning to predict the performance of professional golfers. This end-to-end machine learning project involves web scraping, data manipulation, exploratory data analysis, model training, model optimization, model evaluation and predictive modelling.

1.1 The Sport of Golf

Golf is a sport in which an individual uses a club to hit a ball into a hole. The objective of the game is to get the ball into the hole in the least amount of hits possible. The player hits the first ball from a tee off area, signalling the beginning of the hole. The hole is placed in a smooth area of short grass that is called the green.[1] When on the green the player uses a club called a putter to lightly hit the ball into the hole. The number of strokes the player takes to hit the ball in the hole is added up and recorded.[2] A typical golf course consists of eighteen holes. At the end of the round the total strokes through the eighteen holes is added up and the golfer with the fewest number of strokes wins.[3] One of the most popular terms used in golf is "par", which refers to the number of strokes an expert golfer is expected to complete an individual hole. The par of each hole is added up over all the holes to determine the total par of a course. The term "birdie" is used when a player hits the ball into the hole in one less stroke than the par. A "bogey" refers to when a player takes one extra stroke than par to get the ball into the hole. Golf is an $84 billion industry that is one of the most popular sports in the world, enjoyed by more than 60 million players.[4]

1.2 PGA Tour

The PGA Tour is the organizer of the main professional golf tournaments and is recognized as the top professional league in the world. The tour has a rich history of professional golfers since its beginning in 1929. Some of its most notable players include, Tiger Woods, Jack Nicklaus, Arnold Palmer, Phil Mickelson, Ben Hogan and many more. The game of golf has changed dramatically

[1] *Sports*. URL: https://www.ducksters.com/sports/golf.php.

[2] *Sports*.

[3] *Sports*.

[4] Erik Matuszewski. *The State Of Golf For 2019 – An Industry Roundtable*. 2019. URL: https://www.forbes.com/sites/erikmatuszewski/2019/05/01/the-state-of-the-golf-industry-for-2019/#48f638c52082.

since 1929 and as a result there are records being broken year-over-year. With the popularity of the sport continuing to grow, the prize pool for tournaments has grown significantly as well. The winner of the 2019 PGA Tour FedEx Cup, Rory McIlroy took home $15,000,000. The amount is nearly three times the amount that Jack Nicklaus won with his 73 wins over his 25-year career. Giving players an analytical advantage is crucial for increasing their chances of winning.

1.3 Statistical Modelling

Famous computer scientist Arthur Samuel defined machine learning as, "The field of study that gives computers the ability to learn without being explicitly programmed."[5] With the amount of data growing exponentially year-over-year, there are plenty of potential applications for machine learning. It is estimated that currently less than 0.5% of available data is being analyzed.[6]

There are multiple target variables that will be used for predictions, depending on the type of problem. When dealing with the historical data, the models will predict the scoring average of an individual throughout the season. The scoring average is the average number of strokes per completed round. Regression models will be implemented due to the feature being continuous. As for the current year, the first target variable is a continuous variable measuring the amount of money earned each week, ranked in ascending order. The weekly earnings rank variable will be used to explore regression techniques. The second target variable is binary and measures whether an athlete will make the cut at an upcoming event. The cut determines the number of players that will make it to play on the weekend of the tournament. The binary variable allows for the exploration of classification models.

[5] URL: `http://www.contrib.andrew.cmu.edu/~mndarwis/ML.html`.

[6] Anthony. *Big Data facts - How much data is out there?* 2019. URL: `https://www.nodegraph.se/big-data-facts/`.

2 Methods

The following section will discuss the methods used in the project.

2.1 Data

For exploration purposes the data has been split into two separate data sets. The first for historical data consisting of statistics from the 2010 to 2018 seasons, and the second for the current year data consisting of statistics from the 2019 season.

2.1.1 PGA Tour Website Data Structure

In order to receive the most accurate tournament predictions, it is important to create data sets with as much information as possible. The more variables to test with the model, the greater the likelihood of an accurate prediction. There are several hundreds of statistics recorded on the PGA Tour website that are used to analyze the performance of each athlete.[7] The Statistics can be divided into the following sub-categories, Off the Tee, Approach the Green, Around the Green, Putting, Scoring, Streaks, Money/Finishes and Points/Rankings. A few of the statistics include, Driving Distance, Driving Accuracy Percentage, Club Head Speed, Ball Speed, Greens in Regulation Percentage, One-Putt Percentage, and many more. Every statistic from each of the sub-categories has been scraped from the website. Within each statistic there are variables that help to indicate the players performance, including the rank of the player in the given statistic, the number of rounds played, and others. There are over 2,000 variables in the 2010-2018 data set, and nearly 1,500 variables in the 2019 data set. The data is updated at the conclusion of every tournament and the statistics are either averaged or summed throughout the entire season.

2.1.2 Web Scraping

There are two web scraping programs that have been created, one to scrape historical data, and the other for current year data. The programs are very similar, however there are slight differences to account for acquiring the historical data. The programming language Python and the BeautifulSoup library were used to create the web-scraping programs. BeautifulSoup is a Python library used for pulling data out of HTML and XML files.[8] The Pandas library was used to manipulate and analyze the data that was retrieved.

2.1.3 Name Correction

When retrieving data from the web, Python is incapable of handling accented characters in the standard utf-8 Unicode format. The name correction code was created to fix athletes names that

[7] *PGA TOUR: Stats Leaders.* URL: https://www.pgatour.com/stats.html.

[8] *Beautiful Soup Documentation¶.* URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

have an accented character. The code takes the Unicode version of a player's name and converts it to a non-accented name. For example, the player Miguel Ángel Jiménez would be displayed on the PGA Tour website properly with the given accents. However, Python would read the data in utf-8 formatting and display the name in the Python data set as "Miguel Angel JimÃ\x83Â©nez". All accents on athlete names have been removed to allow for easier manipulation and the cleanest display of names. Therefore, the name Miguel Ángel Jiménez would be changed to Miguel Angel Jimenez. New accented letters will need to be added to the name correction code manually when discovered.

2.1.4 Historical Data Format

The historical data set contains data from the 2010 to 2018 seasons and consists of the average or sum of the players statistics over the entire season. Below is an example of the format for the historical data set. The example consists of two players and two statistics over two seasons.

Player Name	Season	Statistic	Variable	Value
John Daly	2010	SG: Tee-to-Green	SG: Tee-to-Green - (ROUNDS)	63
John Daly	2010	SG: Tee-to-Green	SG: Tee-to-Green - (AVERAGE)	-0.335
John Daly	2010	SG: Tee-to-Green	SG: Tee-to-Green - (SG:OTT)	0.336
John Daly	2010	SG: Tee-to-Green	SG: Tee-to-Green - (SG:APR)	-0.374
John Daly	2010	SG: Tee-to-Green	SG: Tee-to-Green - (SG:ARG)	-0.298
John Daly	2010	SG: Tee-to-Green	SG: Tee-to-Green - (MEASURED ROUNDS)	47
Tiger Woods	2012	SG: Putting	SG: Putting - (ROUNDS)	69
Tiger Woods	2012	SG: Putting	SG: Putting - (AVERAGE)	0.339
Tiger Woods	2012	SG: Putting	SG: Putting - (TOTAL SG:PUTTING)	16.282
Tiger Woods	2012	SG: Putting	SG: Putting - (MEASURED ROUNDS)	48

The historical data set contains 2,740,403 rows and 5 columns consisting of 3,053 players over the nine-year period. Although a long format was the better storage option for this data set, it is often easier to work with machine learning algorithms when the data is in a wide format. Therefore, the data was transformed before beginning the analysis. Below is a description and example of each column in the historical data set.

Field Name	Data Type	Description	Example
Player Name	Text	Name of the athlete	Tiger Woods
Season	Integer	Season of interest	2010
Statistic	Text	Performance metric	SG: Putting
Variable	Text	Sub-categories within each statistic	SG: Putting - (AVERAGE)
Value	Typically integer, occasionally text	Value of each variable	0.339

2.1.5 Current Year Data Format

The current year data set is updated after the conclusion of every tournament and is the average or sum throughout the entire season. The first tournament recorded in the 2019 data set is the Farmers Insurance Open (January 24 – 27, 2019). The last tournament recorded in the 2019 data set is the Tour Championship (August 22 – 25, 2019). Below is an example of the format for the current year data set. The example consists of two players and two different statistics.

Player Name	Date	Statistic	Variable	Value
Phil Mickelson	2019-03-24	Driving Distance	Driving Distance - (ROUNDS)	26
Phil Mickelson	2019-03-24	Driving Distance	Driving Distance - (AVG.)	304.7
Phil Mickelson	2019-03-24	Driving Distance	Driving Distance - (TOTAL DISTANCE)	15,844
Phil Mickelson	2019-03-24	Driving Distance	Driving Distance - (TOTAL DRIVES)	52
Tiger Woods	2019-03-24	Putting Average	Putting Average - (ROUNDS)	16
Tiger Woods	2019-03-24	Putting Average	Putting Average - (AVG)	1.785
Tiger Woods	2019-03-24	Putting Average	Putting Average - (GIR PUTTS)	382
Tiger Woods	2019-03-24	Putting Average	Putting Average - (GREENS HIT)	214
Tiger Woods	2019-03-24	Putting Average	Putting Average - (BIRDIE CONVERSION)	31.78
Tiger Woods	2019-03-24	Putting Average	Putting Average - (GIR RANK)	3

The weekly tournament data sets for the 2019 season vary in size but are approximately 300,000 rows and 5 columns per tournament. In order to analyze the tournament results, the data must be scraped weekly. The PGA Tour website is updated every Sunday night at the conclusion of each tournament. The results from the weekend are then added onto the data from the current season and averaged or summed over the entire season. By scraping the results weekly, it is possible to see the progression of an athlete over the season by analyzing the week-to-week differences. Below is a description and example of each column in the current year data set.

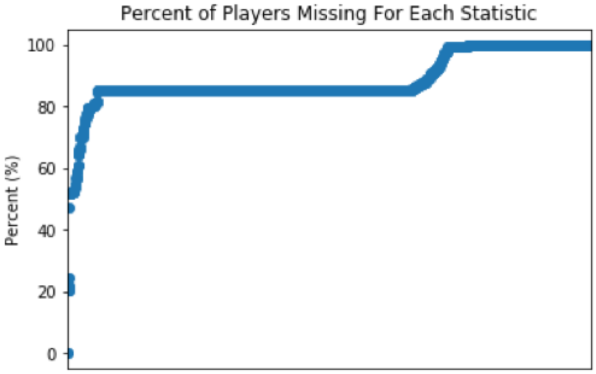
Field Name	Data Type	Description	Example
Player Name	Text	Name of the athlete	Tiger Woods
Date	Date	Last day of the tournament of interest	2019-03-24
Statistic	Text	Performance metric	Putting Average
Variable	Text	Sub-categories within each statistic	Putting Average - (AVG)
Value	Typically integer, occasionally text	Value of each variable	1.785

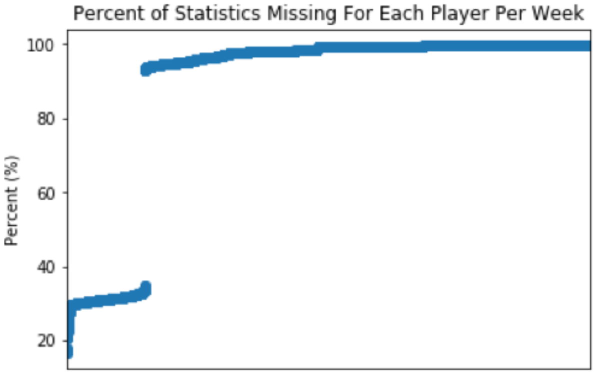
2.1.6 Missing Value Analysis

In order to get a better understanding of the data one must take a further look into the missing values that exist in the data set. Getting a better understanding of the missing values is crucial in order to determine how to deal with them.

2.1.6.1 Historical Data Missing Value Analysis

The initial wide formatted data set consists of 11,223 rows and 2,083 columns. In the wide format, the rows consist of players and the seasons they played on tour. The columns are the variables from the PGA Tour website. Analyzing the data after making the transformation, there appears to be plenty of missing values. Below are graphs showing the percentage of missing data in columns (left) and percent of missing data in rows (right).





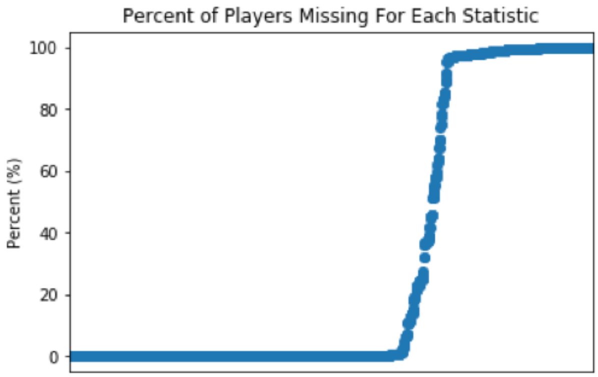
Statistic	Value
Count	2083
Mean	88.3677
Standard Deviation	9.5083
Minimum	0
25%	85.04856
50%	85.04856
75%	99.5812
Maximum	100

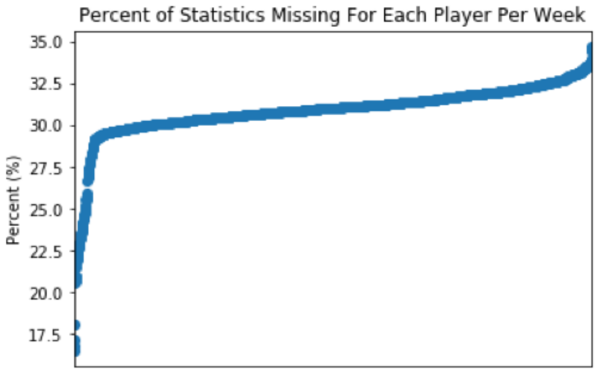
Statistic	Value
Count	11223
Mean	88.3677
Standard Deviation	24.1530
Minimum	16.4666
25%	95.9674
50%	99.2799
75%	99.6159
Maximum	99.8560

As shown above, there appears to be a large amount of missing data. In the statistic “Official World Golf Ranking”, players may be assigned values for the statistic nearly every year regardless of whether they compete or not. A players Official World Golf Ranking (OWGR) points stay on the players tally for two years.[9] Therefore, it is possible that a very dominate player may still be on the OWGR leader board even if they do not play on the PGA Tour for over a year. Since the data is in a wide format, if there is only one variable with a value, then the values of every other variable in the row will be null. Another reason why certain records are nearly entirely null, is due to the “Official World Golf Ranking” statistic taking into consideration 34 professional golf tours. Many of these tours are not as prominent as the PGA Tour and often lack data. As a result, nearly all of the statistics for an athlete for the given year will be null. As shown by the above graph it is obvious that there is a significant jump from approximately 35% missing data to 92% missing data. If there is greater than 90% of the data missing for a certain athlete, the athlete likely did not compete that year but are being assigned values for the “Official World Golf Ranking” statistics.

In order to reduce the number of missing values, a threshold of 40% percent missing data has been set. The threshold is approximately equal to 1,250 non-null values out of a total of 2,083 columns. Meaning if there are more than 833 null values, the row will be excluded from the data set. After the threshold has been set, there remains 1,678 rows and 2,083 columns. On the left is the percentage of missing data in the columns. On the right is the percentage of missing data in the rows.

[9] *Official World Golf Ranking - Home.* URL: `http://www.owgr.com/`.

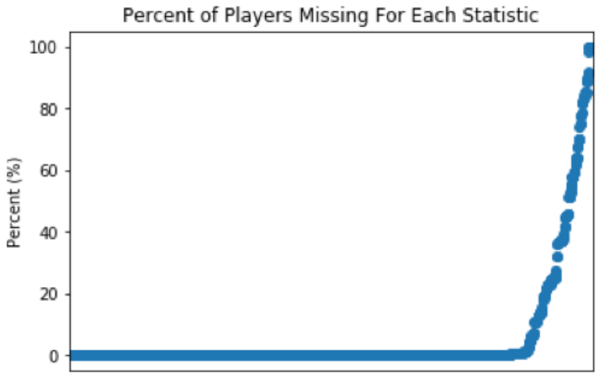


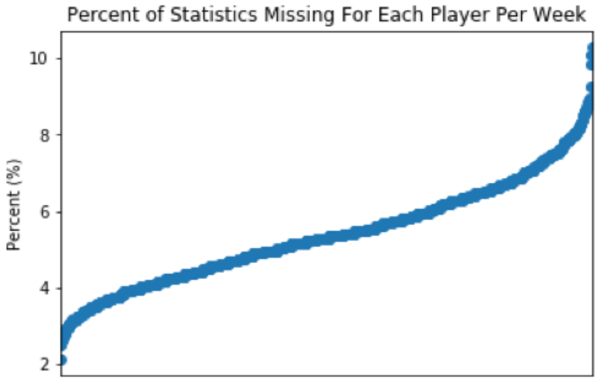


Statistic	Value
Count	2083
Mean	30.8989
Standard Deviation	44.2833
Minimum	0
25%	0
50%	0
75%	97.2586
Maximum	100

Statistic	Value
Count	1678
Mean	30.8989
Standard Deviation	1.6922
Minimum	16.4666
25%	30.3409
50%	31.06097
75%	31.7811
Maximum	34.7576

As shown above in the graph on the left, there is a step increase from approximately 0% missing data to 95% missing data. After analyzing the columns with more than 95% missing data, a significant number of the columns are of the category "Distance Analysis". Distance Analysis was a measure used in 2010 to analyze how far athletes would hit a particular club. Due to the discontinuation of the statistic after 2010, a large majority of the statistic is missing. However, setting a threshold of 95% missing values would remove important statistics above the threshold, including statistics measuring Money Leaders. Therefore, rather than setting a threshold, all Distance Analysis statistics have been removed. After removing the Distance Analysis columns, the data set consists of 1,678 rows and 1,515 columns.





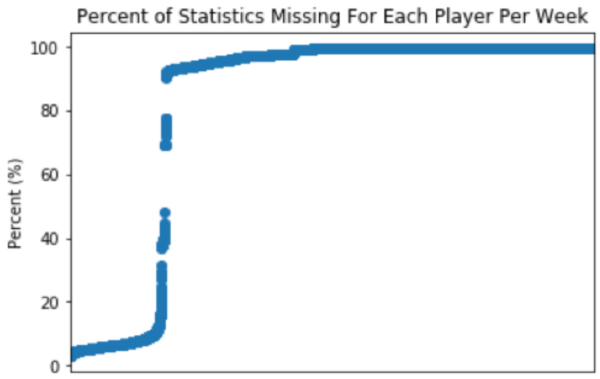
Statistic	Value
Count	1515
Mean	5.3880
Standard Deviation	17.5518
Minimum	0
25%	0
50%	0
75%	0
Maximum	100

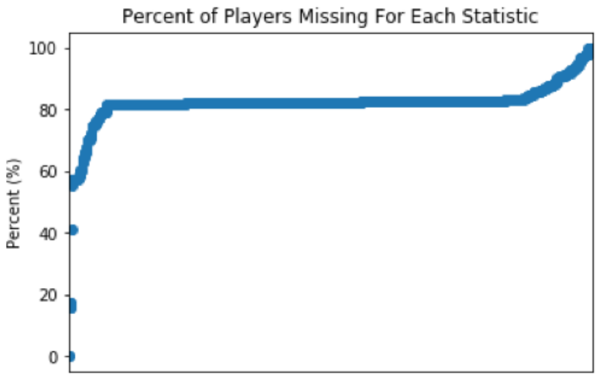
Statistic	Value
Count	1678
Mean	5.3880
Standard Deviation	1.3208
Minimum	2.1122
25%	4.3564
50%	5.2805
75%	6.2706
Maximum	10.2970

The mean of the missing values in the rows is 5.3880%, with a standard deviation of 1.3208%. The mean of the missing values in the columns is 5.3880% with a standard deviation 17.5518%. The final data set now consists of 1,678 rows and 1,515 columns. Where rows are the player in a given season, and columns are the variables. The maximum percent of missing rows is 10.2970%, and the minimum percent of missing rows is 2.1122%. There are now 439 players in the data set. After making the transformations and setting the thresholds for missing values, out of the entire data set, there is a total of 5.3880% missing data. Therefore, nearly 95% of the data set is non-null values.

2.1.6.2 Current Year Data Missing Value Analysis

After combining each file in the 2019 season, it is time for exploratory analysis on the missing values. At the conclusion of the 2019 season on August 25, 2019, a summary of the missing data is as follows. Before setting a threshold or manipulating the data, the mean value of the percent of statistics missing for each player per week is 82.8190% with a standard deviation of 34.8190%.

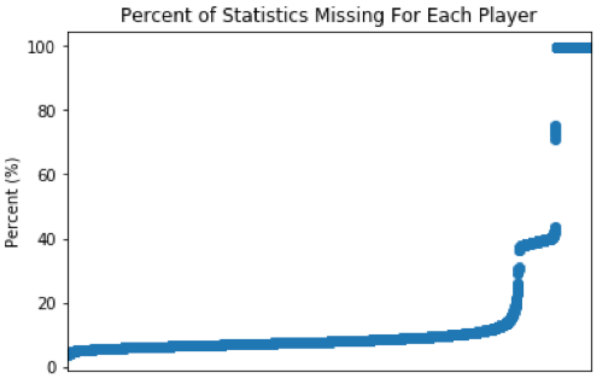


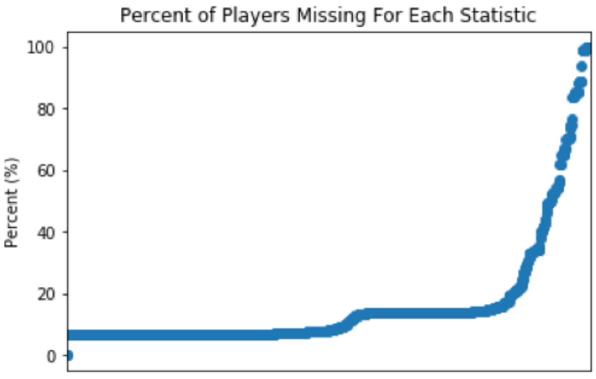


Statistic	Value
Count	3793
Mean	82.8190
Standard Deviation	34.8190
Minimum	2.9373
25%	94.5260
50%	99.4660
75%	99.4660
Maximum	99.8000

Statistic	Value
Count	1498
Mean	82.1077
Standard Deviation	7.2115
Minimum	0
25%	81.9709
50%	82.1026
75%	82.5944
Maximum	100

Some necessary manipulation has been done prior to setting a threshold. If the player did not compete in the given week, the records have been removed. Using the variable "1-Putts per Round - (ROUNDS)" to analyze the difference in rounds from week-to-week. If the difference in rounds is zero, the record has been excluded. In order to reduce complexity, statistics that are an average or percentage throughout the entire season have also been removed. Handling the week-to-week differences in averages or percentages would require a different method for analysis. After completing the necessary manipulation, the missing values are as follows.

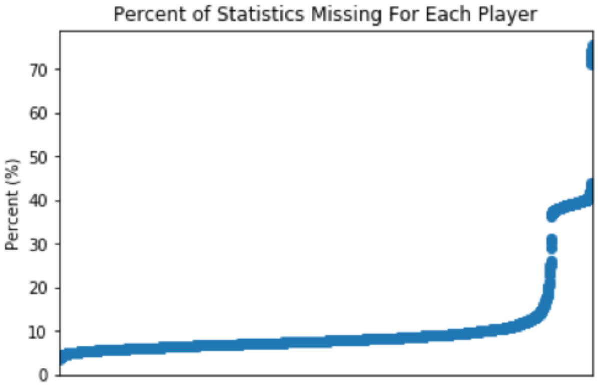


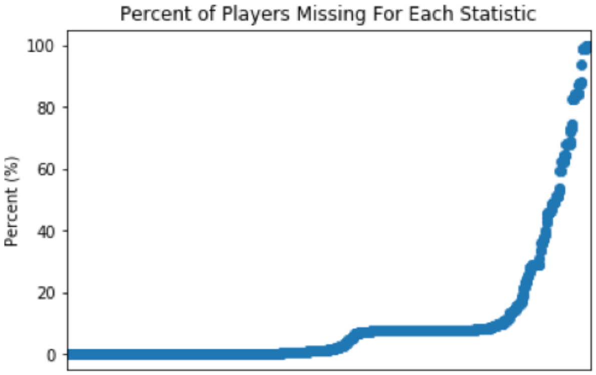


Statistic	Value
Count	3585
Mean	16.4195
Standard Deviation	23.9493
Minimum	3.5120
25%	6.6543
50%	7.8558
75%	10.1664
Maximum	99.8152

Statistic	Value
Count	1082
Mean	16.4195
Standard Deviation	19.1893
Minimum	0
25%	6.8811
50%	8.1861
75%	13.8940
Maximum	100

In order to reduce the number of missing values, a threshold of 99% percent missing data has been set. The threshold is approximately equal to 11 non-null values out of a total of 1,082 columns. Meaning if there are more than 1,071 null values, the row will be excluded from the data set.





Statistic	Value
Count	3539
Mean	10.4340
Standard Deviation	8.9223
Minimum	3.5120
25%	6.5619
50%	7.6710
75%	9.4270
Maximum	75.3235

Statistic	Value
Count	1082
Mean	10.4340
Standard Deviation	20.5575
Minimum	0
25%	0.1978
50%	1.5965
75%	7.7140
Maximum	100

The final data set now consists of 3,539 rows and 1,082 columns. Where rows are the player in a given season, and columns are the variables. There are now 226 unique players in the data set. After making the transformations and setting the thresholds for missing values, out of the entire data set, there is a total of 10.4340% missing data. Therefore, nearly 90% of the data set is non-null values.

2.1.7 Handling Missing Values

Many machine learning models are incapable of handling any missing values in the data. Therefore, a technique must be chosen for handling and filling in the missing values. After evaluating the performance metrics with various methods of filling missing values, a series of two methods led to the highest predictive power. The first step is to group by the player, sort the date in ascending order and fill the missing value with the next non-null value. However, if the last value in a player's record is null, then there is no next non-null value for the player to fill in. Therefore, the remaining missing values were filled in using the mean value for the statistic.

2.1.8 Creating 2019 Target Variables

The target variable used for the historical data is the weighted scoring average. The PGA Tour statistics website has two different variables regarding scoring average. According to the website, the “Scoring Average” variable is defined as, “The weighted scoring average which takes the stroke average of the field into account. It is computed by adding a player’s total strokes to an adjustment and dividing it by the total rounds played. The adjustment is computed by determining the stroke average of the field for each round played. This average is subtracted from par to create an adjustment for each round. A player accumulates these adjustments for each round played.”[10] The second variable regarding scoring average is, “Scoring Average (Actual)” which is defined on the PGA Tour statistics website as the following, “The average number of strokes per completed round.”[11] However, this statistic does not take into consideration the strength of the field in the tournament. Meaning, if a player only competes at courses where it is easier to score well, their scoring average throughout the season may not be a fair comparison to others. The original thought was to use the scoring average of an individual as the target variable for the 2019 season as well. However, the most ideal statistic for analyzing tournament outcomes is actually the “Official Money” statistic. The “Official Money” statistic is defined as, “The total official money a player has earned year-to-date. Note: This statistic is for PGA TOUR members only.”[12] Since the data has been scraped every week beginning in January 2019, it is possible to see the week-to-week progression in “Official Money”. If the week-to-week earnings amount was used as the target variable, the model may place more importance on a tournament where the purse is larger. Therefore, it is important to rank the week-to-week earnings in ascending order, with a rank of one for the individual that earned the most money in the given week. The newly created variable, “Weekly Earnings Rank”, gives information on the tournament results for each week. The “Weekly Earnings Rank” variable will be used as the target variable for all regression models for the 2019 season. As for the classification models, it is possible to determine the players that did not make the cut at a given event by using the week-to-week earnings variable. The cut determines the number of players that will make it to play on the weekend of the tournament.[13] A player that makes the cut will receive a cheque based on their performance, those that do not make the cut will not receive compensation. In the rare occasion that a player is disqualified or withdraws from a tournament, they will also not make money. Therefore, this will be treated similarly to missing the cut at an event. The “Weekly Cut” variable will be used as the target variable for all classification models.

[10] *Stat – Scoring Average.* URL: https://www.pgatour.com/stats/stat.120.html.

[11] *Stat – Scoring Average (Actual).* URL: https://www.pgatour.com/stats/stat.108.html.

[12] *Stat – Official Money.* URL: https://www.pgatour.com/stats/stat.109.html.

[13] Robert Preston. *How Is the Cut Determined in Golf Tournaments?* 2017. URL: https://golftips.golfweek.com/cut-determined-golf-tournaments-1857.html.

2.1.9 Preparing 2019 Data for Machine Learning Model

In order to determine the progression of a player over the season, it is important to analyze the week-to-week difference between variables. The week-to-week differences of statistics gives an indication of how well a player has played over the previous weeks. To determine the optimal number of weeks to take into consideration, the model has been run multiple times with two weeks, three weeks, four weeks and five weeks prior to the week of interest. Three weeks prior to the week of interest led to the optimal models evaluated by the performance metrics. To take into consideration the previous three weeks of data, each variable is displayed as below. This method leads to three times the number of features in the model, yet it is crucial to see the progression of an athlete.

Player Name	Date	SG: Total - (TOTAL SG:T) (Week t-0)	SG: Total - (TOTAL SG:T) (Week t-1)	SG: Total - (TOTAL SG:T) (Week t-2)	SG: Total - (TOTAL SG:T) (Week t-3)
Brooks Koepka	21	14.877	1.088	18.502	13.248
Brooks Koepka	22	-0.620	14.877	1.088	18.502
Brooks Koepka	24	-1.010	-0.620	14.877	1.088

2.2 Machine Learning Techniques

The following section will discuss the machine learning techniques used in the project.

2.2.1 Standardization

Prior to creating and tuning the machine learning models, it is important to ensure the data is in the appropriate format. Standardization is the process of putting different variables on the same scale.[14] This process allows one to interpret the scores between the different variables as they are all in terms of standard deviations from the mean. Z-Score is one of the most popular standardization methods. It converts the features to a common scale with an average of zero and standard deviation of one.[15] Z-Scores are calculated by subtracting the mean and dividing the standard deviation for each value of each feature. In LASSO or ridge regression, due to the penalty placed on the magnitude of the coefficients in a model, one must standardize the data first. The penalty placed on the coefficients will largely depend on the scale of the variables. Coefficients with large variance are small and therefore the penalty is smaller. Therefore, the data must be standardized prior to training the models in order to obtain accurate results.

[14] Jim Frost. *Standardization.* URL: https://statisticsbyjim.com/glossary/standardization/.

[15] *How raw data are normalized - HowTo.ComMetrics.* URL: http://howto.commetrics.com/methodology/statistics/normalization/.

2.2.2 Regularization

Regularization is a common practice in machine learning that helps to reduce the complexity of the model by reducing the contribution of less important features to the model. Typically, regularization leads to higher bias but less variance. The process of attempting to minimize the bias and variance is known as the bias-variance trade-off. In the LASSO and ridge regression models as the lambda value increases, the bias increases, yet the variance decreases. Lambda is a penalty coefficient which regularizes the coefficients. Bias can be defined as the difference between the expected value of the estimator and the true value being estimated.[16] Whereas, variance is the spread or uncertainty of the estimates.[17] Regularization is nearly always beneficial to the predictive performance of the model.[18] L1 regularization and L2 regularization are two of the most popular regularization techniques. LASSO regression performs L1 regularization while ridge regression performs L2 regularization. L1 regularization adds a penalty that is proportional to the square of the magnitude of coefficients.[19] Whereas, L2 regularization adds a penalty that is proportional to the absolute value of the magnitude of coefficients.[20] Both regularization methods minimize the contribution of less important features to the model by adding a penalty term to the loss function.[21] Through the penalty term, L1 regularization encourages the coefficients of the less important features to zero, whereas L2 regularization encourages coefficients to a very small value. Due to L1 regularization entirely removing certain features from the model, the method also can be used as a feature selection tool. With over 4,500 features in the data set, it is ideal to reduce the number of features while retaining a high predictive accuracy.

2.2.3 K-Fold Cross-Validation

In order to evaluate the performance of a machine learning model, the data set is often split into training and testing data. The training data is used to train the machine learning model and the testing data is used to evaluate the performance of the model. It is important to note that data in the testing data cannot also be in the training data. If the training data contains records from the testing data, then the model would be able to learn or know something that the model would not have otherwise known.[22] However, when the model is put into production it will perform very poorly. This is a common problem known as data leakage. This approach may lead to large

[16] URL: http://www.statisticalengineering.com/Weibull/precision-bias.html.

[17] *(Tutorial) Regularization: Ridge, Lasso and Elastic Net.* URL: https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net.

[18] *(Tutorial) Regularization: Ridge, Lasso and Elastic Net.*

[19] Aarshay Jain, Aarshay, and Columbia University. *A Complete Tutorial on Ridge and Lasso Regression in Python.* 2019. URL: https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-ridge-lasso-regression-python/.

[20] Jain, Aarshay, and University, *A Complete Tutorial on Ridge and Lasso Regression in Python.*

[21] Anuja Nagpal. *L1 and L2 Regularization Methods.* 2017. URL: https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c.

[22] Jason Brownlee. *Data Leakage in Machine Learning.* 2019. URL: https://machinelearningmastery.com/data-leakage-machine-learning/.

variance between testing sets. Cross validation is a model validation technique that is used to evaluate models with a limited amount of data.[23] Cross validation protects against overfitting in a predictive model and is particularly important when the amount of data is limited.[24] As described in the book “An Introduction to Statistical Learning”, “This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining k-1 folds.”[25] In simpler terms, of the k folds, k-1 is used for training while the remaining fold is used for testing.[26] The average of the performance metric of the k-folds is found to evaluate the performance of the model.

2.2.4 Gradient Boosting

Gradient Boosting is commonly recognized as one of the most powerful techniques for predictive modeling.[27] Gradient boosting uses an ensemble of weak learners to build predictive regression or classification models.[28] XGBoost (Extreme Gradient Boosting) is a common technique that has quickly become one of the most popular machine learning algorithms. Its speed, performance, and wide variety of tuning parameters has brought a lot of attention to the model. XGBoost can be used for both regression and classification problems, which speaks to its flexibility.

2.2.5 Hyperparameter Optimization

When implementing a machine learning model, it is not typically obvious which parameters within the algorithm allow for optimal performance. The best way to determine the optimal model is by testing various combinations of parameters. Grid search is a common technique that finds the optimal parameters for a given model. Each algorithm has unique parameters to be tuned so a grid search was used on each model. Using a Python machine learning library called Scikit-Learn, the user is able to input a range of values for each parameter and the function will loop through the possible combinations of each parameter to find the model with the best performance metric. The GridSearchCV function from Scikit-Learn can be defined in the following way, “the parameters of the estimator used to apply these methods are optimized by cross-validated grid-search over a parameter grid.”[29]

[23] Jason Brownlee. *A Gentle Introduction to k-fold Cross-Validation.* 2019. URL: https://machinelearningmastery.com/k-fold-cross-validation/.

[24] *What is Cross-Validation? - Definition from Techopedia.* URL: https://www.techopedia.com/definition/32064/cross-validation.

[25] Gareth James et al. *An introduction to statistical learning: with applications in R.* Springer, 2017.

[26] Usman Malik. *Cross Validation and Grid Search for Model Selection in Python.* 2019. URL: https://stackabuse.com/cross-validation-and-grid-search-for-model-selection-in-python.

[27] Jason Brownlee. *A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning.* 2019. URL: https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/.

[28] *(Tutorial) Learn to use XGBoost in Python.* URL: https://www.datacamp.com/community/tutorials/xgboost-in-python#what.

[29] *sklearn.model$_s$election.GridSearchCV–.* URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

2.2.6 Model Evaluation

Each model must be evaluated in order to decide which algorithm should be used for predictions. There are numerous performance metrics that can be used to evaluate the performance of the models. The performance metric that is used to evaluate the performance of regression models in the 2010-2018 seasons is R-squared. R-squared is defined as a statistical measure that represents the proportion of the variance for a dependent variable that is explained by independent variables in a regression model.[30] The performance metric that is used to evaluate the performance of regression models in the 2019 season is root mean square error (RMSE). RMSE is the standard deviation of the residuals, where residuals are a measure of how far the data points are from the regression line.[31] The performance metric that is used to evaluate the performance of classification models is Area under the ROC curve (AUC). The ROC (Receiver Operating Characteristic) Curve shows the performance of a classification model at all thresholds.[32] AUC measures the probability of predicting a real positive will be positive, versus the probability of predicting a real negative will be positive.[33]

2.3 Machine Learning Algorithms

This project utilizes popular machine learning regression algorithms such as ridge regression, LASSO regression and XGBoost regression. The regression methods are used to predict the scoring average of individuals, and the rank of players at upcoming PGA Tour tournaments. In addition to the regression methods, the project implements multiple classification models. Using classification algorithms, it is possible to predict whether an individual will make the cut at an upcoming tournament. The classification models that have been implemented are, logistic regression, decision tree classification and XGBoost classification.

2.3.1 Linear Regression

Regression analysis is a statistical method that is used to evaluate the relationship between two or more variables of interest.[34] Regression analysis analyzes the influence one or more independent variables has on a dependent variable. Where the dependent variable is the factor that you are trying to predict, and the independent variables are the factors that you believe will have an impact on the dependent variable.[35]

[30] Adam Hayes. *R-Squared.* 2019. URL: https://www.investopedia.com/terms/r/r-squared.asp.

[31] Stephanie. *RMSE: Root Mean Square Error.* 2019. URL: https://www.statisticshowto.datasciencecentral.com/rmse/.

[32] *Classification: ROC Curve and AUC — Machine Learning Crash Course.* URL: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.

[33] *What Is an ROC Curve?* 2018. URL: https://www.theanalysisfactor.com/what-is-an-roc-curve/.

[34] Ben Foley. *What is Regression Analysis and Why Should I Use It?: SurveyGizmo Blog.* 2019. URL: https://www.surveygizmo.com/resources/blog/regression-analysis/.

[35] Foley, *What is Regression Analysis and Why Should I Use It?: SurveyGizmo Blog.*

2.3.1.1 LASSO Regression

As mentioned above, LASSO regression is a model that doubles as a feature selection tool to reduce the weight of the coefficients that are not that significant to the model. LASSO regression encourages coefficients to zero. In order to find the optimal model, the value of lambda must be tuned. If the model is fed a lambda value of zero, it will become an ordinary least squares model.[36] This would remove the regularization that was needed. If the model is given a large lambda value, it will place high importance on the features and could lead to underfitting. There is a direct correlation between the lambda value and the number of features in the model. As the lambda value decreases, the model retains a large number of features, however, as the lambda value increases there will be more features given a coefficient of zero and would therefore be excluded from the model. In order to find the optimal lambda values, it is important to evaluate the performance of multiple models with various hyperparameters. The lambda values tested in the model start at 0 with an increment of 0.1, up to the largest value of 2.5.

2.3.1.2 Ridge Regression

A second method for reducing the model complexity is ridge regression. Ridge regression reduces the contribution of less important features by encouraging coefficients to very small values. With ridge regression it is possible to retain all of the features in the original multiple linear regression model while still reducing model complexity. Similar to the LASSO regression model, the lambda value controls the amount of regularization in the model. The lambda values tested in the model start at 0 and go up to the largest value of 1, with finer values closer to the lower and upper bound. contribution of less important features to the model

2.3.1.3 XGBoost Regression

The regression implementation of the XGBoost algorithm consists of hyperparameters for regression problems. XGBoost regression uses the boosting method discussed above to predict the continuous dependent variable. The hyperparameters tuned in the scikit-learn implementation of the XGBoost regression model include, colsample_bytree, n_estimators, and max_depth. Colsample_bytree specifies the fraction of features to choose from at every split in a tree.[37] The value must be between 0 and 1. The colsample_bytree values tested in the model start at 0.1 with an increment of 0.2, up to the largest value of 0.7. The n_estimators hyperparameter refers to the number of trees (or rounds) in the XGBoost model. The n_estimators values start at 25 with an increment of 25, up to the largest value of 50. The max_depth hyperparameter dictates the maximum depth that each tree in a boosting round can grow to.[38] Smaller values lead to shallower

[36] Nagpal, *L1 and L2 Regularization Methods*.

[37] *Tuning colsample$_b$ytree.* URL: `https://campus.datacamp.com/courses/extreme-gradient-boosting-with-xgboost/fine-tuning-your-xgboost-model?ex=8`.

[38] *sklearn.tree.DecisionTreeClassifier¶*. URL: `https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html`.

trees, and larger values to deeper trees.[39] The max_depth values tested in the model start at 3 with an increment of 2, up to the largest value of 11.

2.3.2 Classification

Classification is a type of supervised learning that specifies the class to which data elements belong to.[40] There are two types of classification, binary classification and multi-class classification. Binary classification is used when there are two classes, whereas multi-class classification is used when there are three or more classes.

2.3.2.1 XGBoost Classification

The classification implementation of the XGBoost algorithm consists of hyperparameters optimized for classification. XGBoost classification uses the boosting method discussed above to predict the categorical dependent variable. The hyperparameters that were tuned in the XGBoost classification model for this project are the same as the hyperparameters tuned in the XGBoost regression model for this implementation. The hyperparameters were given the same lower and upper bounds with the same increment as in the XGBoost regression model. There is plenty of opportunity to explore tuning other hyperparameters, but the parameters tuned in this implementation were commonly recognized as some of the most important to the model.

2.3.2.2 Logistic Regression

Binary logistic regression is a classification algorithm that is used to predict the relationship between independent variables and a dependent variable, where the dependent variable is binary. The logistic regression algorithm uses the logistic/sigmoid function to model the binary dependent variable. The standard logistic function is bounded between 0 and 1. The hyperparameters tuned in the scikit-learn implementation of the logistic regression model include, penalty and C. The penalty hyperparameter refers to the method of regularization, whether it is, L1, L2, Elastic Net, or no regularization at all. The penalty hyperparameters given to the logistic regression model were L1 and L2. The C parameter is used in the logistic regression model as a regularization parameter. C is equal to 1/lambda, where lambda is the regularization parameter used to tune the power of the coefficients in the model. Since the C parameter is the inverse of the typical regularization parameter lambda, the regularization in the logistic regression model will be the opposite of usual. A small value of C increases the regularization, resulting in the potential of underfitting the data. Whereas, a large value of C decreases the regularization, leading to increased model complexity, resulting in the potential of overfitting the data. The C values given to the logistic regression model start at 0 with an increment of 0.1, up to the largest value of 2.5.

[39] *sklearn.tree.DecisionTreeClassifier¶*.

[40] *Classification - Machine Learning: Simplilearn.* URL: `https://www.simplilearn.com/classification-machine-learning-tutorial`.

2.3.2.3 Decision Tree Classification

A decision tree is recognized as one of the most popular and powerful classification and regression tools used for predictions.[41] The decision tree algorithm uses a tree like structure to solve the given problem.[42] The data set is broken down into smaller and smaller subsets while incrementally developing a decision tree.[43] The hyperparameters tuned in the scikit-learn implementation of the decision tree classification model include, criterion and max_depth. The criterion parameter refers to the function that measures the quality of a split.[44] The values include gini and entropy. The gini method favours larger partitions and is easy to implement.[45] Whereas, the entropy method favours partitions that have small counts but have many distinct values.[46] The max_depth parameter is the same as the parameter described above in the XGBoost models. The parameter was given the same range of values starting at 3 with an increment of 2, up to the largest value of 11.

3 Results

The following section will discuss the results of the project.

3.1 Performance Evaluation

The following section will discuss the methods used for evaluating the performance of the models.

3.1.1 Historical Performance Evaluation

When working with the historical data predicting scoring average, the overall strongest model measured by the R-squared value is the ridge regression model with a lambda value of 0.001. This model produced an R-squared value of 0.9995. The strongest LASSO regression model produced an R-squared value of 0.9990 with a lambda value of 0.00001. However, the purpose of using ridge regression and LASSO regression was to reduce the complexity of the model. Since the data contains a very large number of variables, it is important to reduce the importance of the features in the model. Therefore, the LASSO model is the most ideal due to its ability to also be used as a feature selection tool. Although the LASSO regression model with a lambda value of 0.00001 was optimal in obtaining a high R-squared value, it was due to the model being fed nearly all of the

[41] *Decision Tree.* 2019. URL: https://www.geeksforgeeks.org/decision-tree/.

[42] *Decision Tree*.

[43] Chirag Sehra. *Decision Trees Explained Easily.* 2018. URL: https://medium.com/@chiragsehra42/decision-trees-explained-easily-28f23241248.

[44] *sklearn.tree.DecisionTreeClassifier¶*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

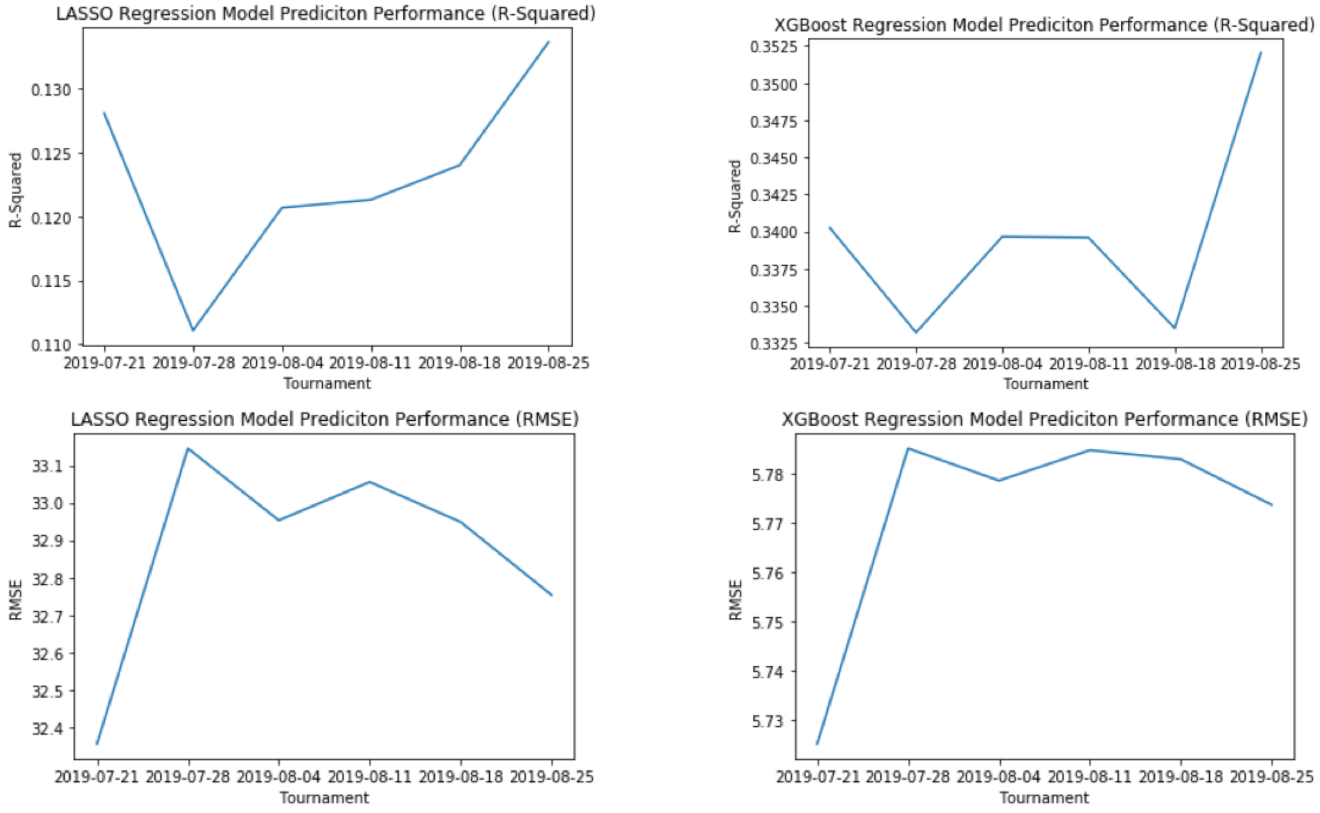
[45] Will. *Learn by Marketing.* 2016. URL: http://www.learnbymarketing.com/481/decision-tree-flavors-gini-info-gain/.

[46] Will, *Learn by Marketing*.

original features. This could lead to the potential of overfitting the data. The model was nearly an OLS model. When taking into consideration the number of features as well as the strength of the model, the LASSO regression model with a lambda value of 0.9 was optimal. With the given lambda value, the number of features was reduced from 1,515 to 24 while still obtaining a high prediction accuracy with an R-squared value of 0.9190. The LASSO regression model reduced the complexity of the model leading to higher predictive power.

3.1.2 Current Year Performance Evaluation

As shown in the plots below, with the six weeks of analysis there is a slight upward trend when using R-squared to evaluate the performance of the regression models. When using RMSE as the performance metric, it appears that the first week of 2019-07-21 performed above average in comparison to the other 5 weeks of interest. Unfortunately, six weeks is a fairly small sample size to truly see the trend of the performance of the model.



The average R-squared value for the last 6 weeks of the 2019 season for the LASSO regression model is 0.1231, and 0.3397 for the XGBoost for the regression model. The average RMSE value for the last 6 weeks of the 2019 season for the LASSO regression model is 32.8680, and 5.7716 for the XGBoost regression model. Using the performance metrics, it is evident that the XGBoost Regression model is performing significantly better than the LASSO regression model. This goes to show the power of gradient boosting predictive models.

The performance of the classification models in the last week of the 2019 season are as follows. The logistic regression model had an AUC of 0.6047, the decision tree classification model had an AUC of 0.5827, and the XGBoost classification model had an AUC of 0.6092. A final model has been created using XGBoost classification where the target variables are shuffled. This gives insights in the performance of the model when it randomly guesses. This model will be the benchmark to see how well the other models are performing compared to random guessing. The AUC achieved through random guessing is 0.4980.

3.2 Predictions

One of the many challenges of predicting PGA Tour tournament outcomes is due to the large number of athletes competing in each tournament. A tournament typically consists of either, 132, 144 or 156 players depending on the event and its location. It is not unreasonable for a poorly ranked player to have an exceptionally well four rounds of golf and become the champion of the tournament. Golf tends to be a very streaky sport and players often go through highs and lows, making the process of predicting performance even more challenging. After acquiring 25 weeks of data during the 2019 season, it was time to create predictions for upcoming events. To evaluate the predictions, the results will be compared with the top 20 as the key indicator of success, as well as top 10, top 5, and correctly predicting the winner as other measures included. Evaluating the performance of the classification model predictions would require the manual input of data for every athlete in each tournament. Therefore, for now the evaluation of the predictions focuses solely on the performance of the regression models. The first prediction was for The Open Championship on July 18 – 21, 2019. The Open Championship is one of the four major tournaments on the tour and is known for its number of elite players that compete. The Open Championship is often one of the most challenging tournaments of the year due to the courses that host the tournament. The challenging conditions often make for an interesting tournament including a wide variety of players performing well to the surprise of many. The XGBoost regression model was able to correctly predict 6 of the top 20, from a field of 156 players. The following week was the World Golf Championships-FedEx St. Jude Invitational, consisting of 64 players. With 46 of the top 50 golfers in the world competing in the tournament, it is yet another tournament with a strong field.[47] The model correctly predicted 13 of the top 20 players. The next tournament was the Wyndham Championship which consisted of 156 players in the field. The model correctly predicted 7 of the top 20. The Wyndham Championship signalled the end of the PGA Tour regular season. The next three weeks of tournaments were playoffs to decide who will win the 2019 FedEx Cup. The first tournament of the playoffs is The Northern Trust, which consisted of 121 players in the field. The model correctly predicted 6 of the top 20. The second week of playoffs is the BMW Championship which consisted of 70 players in the field. The model correctly predicted 10 of the

[47] PGATOUR.COM Staff. *64 players confirmed for the 2019 WGC-FedEx St. Jude Invitational.* 2019. URL: `https://www.pgatour.com/tournaments/wgc-fedex-st-jude-invitational/field/20/final-field-wgc-fedex-stjude-invitational.html`.

top 20 and 4 of the top 10. The final week of playoffs and the tournament deciding the winner of the FedEx Cup is the Tour Championship which consisted of 30 players in the field. The model correctly predicted 17 of the top 20 and 5 of the top 10. The winner of the 2019 FedEx Cup was Rory McIlroy, who took home a prize of $15,000,000.

4 Discussion

The following section addresses use cases of the project and future applications.

4.1 Public Presence

The existing PGA tour data set has been posted on Kaggle, a popular data science website. The data set has been very popular with over 50,000 views, 9,000 downloads and 30 posted analyses. The data set has even reached and remained at the number one spot as the "hottest" data set for numerous weeks, ahead of the 24,000 plus data sets on the website. Its online presence has allowed numerous people to share their unique analyses and draw insights and conclusion on various statistics. The success of this data set goes to show the lack of golf data that is posted online for public use.

In addition to the data being used for personal exploration, there has also been multiple use cases for academic purposes at various universities. An economics student at the University of Leeds has used the data for his dissertation, where he is analyzing different aspects of the golf game and the skills that contribute to winning. A group of MBA students from the University of Washington have used the data for a major project in their studies to predict world golf rankings.

4.2 Future Applications

There is nearly an endless amount of analysis that can be done using the PGA Tour golf data. Which means there is no shortage for future applications. Unlike other sports, the location of events plays an integral role on how certain athletes will perform. Something that seems to be as insignificant as the type of grass used at the golf course can significantly affect how players perform, especially on the greens. The altitude of a course is very important due to the air density. As the altitude increases, the air density decreases which leads to further ball flight. Humidity and wind also play a very important part in driving distance and accuracy. Having data specific to the course and location of the tournament would help to discover the athletes that perform best in certain conditions.

The model currently takes into consideration the three previous tournaments when making predictions for an upcoming event. However, the model does not know which of the three weeks is the most recent. Implementing an auto-regressive model will give insights on the trend of athletes which in return will increase prediction accuracy.

Increasing the number of data sources will add new features to the model improving accuracy further. The Official World Golf Ranking website has data on 34 professional golf tours. By combining the OWGR data with the PGA Tour data, there will be more information on the players that do not primarily compete on the PGA Tour. This will be important for predicting performance of players that are new to the PGA Tour but have been competing on other professional tours. A web scraping program has recently been created to acquire this data weekly. A second data source that would be beneficial to the project is ShotLink data. The ShotLink System is a platform for collecting and disseminating scoring and statistical data on every shot by every player in real-time.[48] This unique data source would provide the opportunity to create live models that will be capable of making predictions and calculating win probabilities as the tournament progresses. Overall there are plenty of future applications that can be implemented to increase prediction power.

[48] Ron Bryant. URL: http://www.shotlink.com/.

References

URL: http://www.contrib.andrew.cmu.edu/~mndarwis/ML.html.

URL: http://www.statisticalengineering.com/Weibull/precision-bias.html.

Anthony. *Big Data facts - How much data is out there?* 2019. URL: https://www.nodegraph.se/big-data-facts/.

Beautiful Soup Documentation¶. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

Brownlee, Jason. *A Gentle Introduction to k-fold Cross-Validation.* 2019. URL: https://machinelearningmast com/k-fold-cross-validation/.

— *A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning.* 2019. URL: https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/.

— *Data Leakage in Machine Learning.* 2019. URL: https://machinelearningmastery.com/data-leakage-machine-learning/.

Bryant, Ron. URL: http://www.shotlink.com/.

Classification - Machine Learning: Simplilearn. URL: https://www.simplilearn.com/classification-machine-learning-tutorial.

Classification: ROC Curve and AUC — Machine Learning Crash Course. URL: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.

Decision Tree. 2019. URL: https://www.geeksforgeeks.org/decision-tree/.

Foley, Ben. *What is Regression Analysis and Why Should I Use It?: SurveyGizmo Blog.* 2019. URL: https://www.surveygizmo.com/resources/blog/regression-analysis/.

Frost, Jim. *Standardization.* URL: https://statisticsbyjim.com/glossary/standardization/.

Hayes, Adam. *R-Squared.* 2019. URL: https://www.investopedia.com/terms/r/r-squared.asp.

How raw data are normalized - HowTo.ComMetrics. URL: http://howto.commetrics.com/methodology/statistics/normalization/.

Jain, Aarshay, Aarshay, and Columbia University. *A Complete Tutorial on Ridge and Lasso Regression in Python.* 2019. URL: https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-ridge-lasso-regression-python/.

James, Gareth et al. *An introduction to statistical learning: with applications in R.* Springer, 2017.

Malik, Usman. *Cross Validation and Grid Search for Model Selection in Python.* 2019. URL: https://stackabuse.com/cross-validation-and-grid-search-for-model-selection-in-python.

Matuszewski, Erik. *The State Of Golf For 2019 – An Industry Roundtable.* 2019. URL: https://www.forbes.com/sites/erikmatuszewski/2019/05/01/the-state-of-the-golf-industry-for-2019/#48f638c52082.

Nagpal, Anuja. *L1 and L2 Regularization Methods*. 2017. URL: https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c.

Official World Golf Ranking - Home. URL: http://www.owgr.com/.

PGA TOUR: Stats Leaders. URL: https://www.pgatour.com/stats.html.

Preston, Robert. *How Is the Cut Determined in Golf Tournaments?* 2017. URL: https://golftips.golfweek.com/cut-determined-golf-tournaments-1857.html.

Sehra, Chirag. *Decision Trees Explained Easily*. 2018. URL: https://medium.com/@chiragsehra42/decision-trees-explained-easily-28f23241248.

sklearn.model$_s$election.GridSearchCV–. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

sklearn.tree.DecisionTreeClassifier¶. URL: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

sklearn.tree.DecisionTreeClassifier¶. URL: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

Sports. URL: https://www.ducksters.com/sports/golf.php.

Staff, PGATOUR.COM. *64 players confirmed for the 2019 WGC-FedEx St. Jude Invitational*. 2019. URL: https://www.pgatour.com/tournaments/wgc-fedex-st-jude-invitational/field/20/final-field-wgc-fedex-stjude-invitational.html.

Stat – Official Money. URL: https://www.pgatour.com/stats/stat.109.html.

Stat – Scoring Average. URL: https://www.pgatour.com/stats/stat.120.html.

Stat – Scoring Average (Actual). URL: https://www.pgatour.com/stats/stat.108.html.

Stephanie. *RMSE: Root Mean Square Error*. 2019. URL: https://www.statisticshowto.datasciencecentral.com/rmse/.

Tuning colsample$_b$ytree. URL: https://campus.datacamp.com/courses/extreme-gradient-boosting-with-xgboost/fine-tuning-your-xgboost-model?ex=8.

(Tutorial) Learn to use XGBoost in Python. URL: https://www.datacamp.com/community/tutorials/xgboost-in-python#what.

(Tutorial) Regularization: Ridge, Lasso and Elastic Net. URL: https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net.

What Is an ROC Curve? 2018. URL: https://www.theanalysisfactor.com/what-is-an-roc-curve/.

What is Cross-Validation? - Definition from Techopedia. URL: https://www.techopedia.com/definition/32064/cross-validation.

Will. *Learn by Marketing*. 2016. URL: http://www.learnbymarketing.com/481/decision-tree-flavors-gini-info-gain/.