Can Data Science be Used to Predict and Understand Professional Golf?

July 4th, 2022
Eric Walsh

Project Statement:

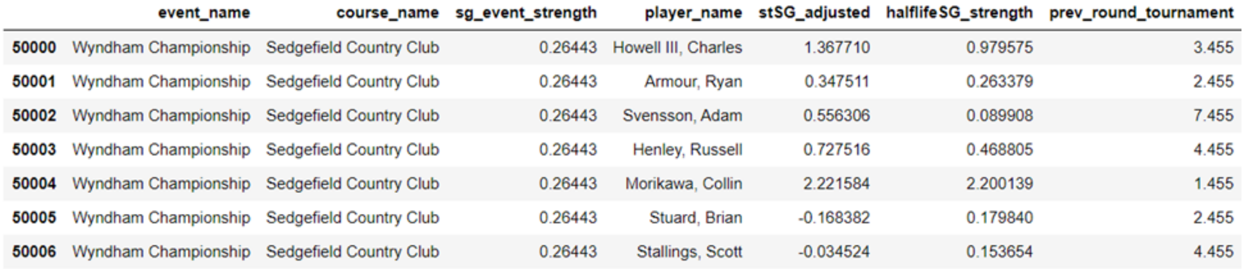
Using historical PGA Tour data, can I create a model that predicts future performance, and in turn be used to create insights into professional golf?

The value added for this project is that the model could be used to create actionable insights for betting, helping professional golfers who are looking to adjust their game, and create a greater general understanding of professional golf.

Golf is a great sport for data- it is a thinking man's game. The golfer himself must make decisions on the fly using yardage numbers, wind numbers, success percentages among a whole range of things. Also, professional golf is an interesting case study for creating models on noisy data. Although round to round performance is somewhat random, professional golf is played over multiple rounds so you get larger sample sets to look at.

Data and Preprocessing

The data for this project was sourced from *Datagolf.com*. The website has a paid API subscription which provides data directly sourced from the PGA tour. The data is round information from all PGA events from 2017 to the current day, along with other quality-of-life tables. Here is a snapshot of the data after it has been accessed from the API:



	event_name	course_name	sg_event_strength	player_name	stSG_adjusted	halflifeSG_strength	prev_round_tournament
50000	Wyndham Championship	Sedgefield Country Club	0.26443	Howell III, Charles	1.367710	0.979575	3.455
50001	Wyndham Championship	Sedgefield Country Club	0.26443	Armour, Ryan	0.347511	0.263379	2.455
50002	Wyndham Championship	Sedgefield Country Club	0.26443	Svensson, Adam	0.556306	0.089908	7.455
50003	Wyndham Championship	Sedgefield Country Club	0.26443	Henley, Russell	0.727516	0.468805	4.455
50004	Wyndham Championship	Sedgefield Country Club	0.26443	Morikawa, Collin	2.221584	2.200139	1.455
50005	Wyndham Championship	Sedgefield Country Club	0.26443	Stuard, Brian	-0.168382	0.179840	2.455
50006	Wyndham Championship	Sedgefield Country Club	0.26443	Stallings, Scott	-0.034524	0.153654	4.455

Figure 1: Snapshot of the ingested data from Datagolf.com. Shows tournament level and player level information.

So, there are a few things to note. The data has two “levels” of information. The first level is the tournament information. This includes the tournament and course metadata, along with the strength of the field. Then there is the second level which has the player information. The information for the players mainly falls into *strokes gained metrics*. Strokes gained is a system developed by the PGA Tour which measures how well a golfer did, in comparison to the rest of the field. So, *strokes gained total* measures how much better/worse a golfer did compared to the average player in that round. Furthermore, strokes gained is broken down into four separate categories: *“off the tee”*, *“approach”*, *“around the green”* and *“putting”*.

Luckily, the dataset itself was very clean. The only null values were for tournaments which did not provide specifics for strokes gained metrics. But when creating factors, an important point is that historical player data was needed to create many of these factors. So, players with a lack of information had to be dealt with. In general, these “*unknown*” players are rookies or fill-in players, so they perform badly compared to the average player. For this reason, and to improve the interpretability of the results, I decided to drop these players from the modeling process.

Factor Engineering

Most of the project work was creating the factors to predict a player’s strokes gained. At a tournament level, there was only 1 main factor that I had to implement. This was *event strength*. Event strength was important because the prediction metric of strokes gained uses the average of the field as a baseline. So, if the field consists of better players, it will be harder to outperform them. The other thing I investigated was *course information*. I was unable to create any simple factors that helped improve model accuracy, so this became one of the “problems” that I investigated later.

The other factors were at a player level. The first and most important one was *historical performance*. I used two metrics to measure this, one which was focused on *long-term performance* (using results from the last two years) and another which focused on the *short-term performance* (using around a month of results). Also, another important factor created here was *skill decompositions,* which used the specific strokes gained categories to create a snapshot of a player's strengths and weaknesses. The creation of these metrics used some normalization and scaling techniques, which the model notebooks go into detail on.

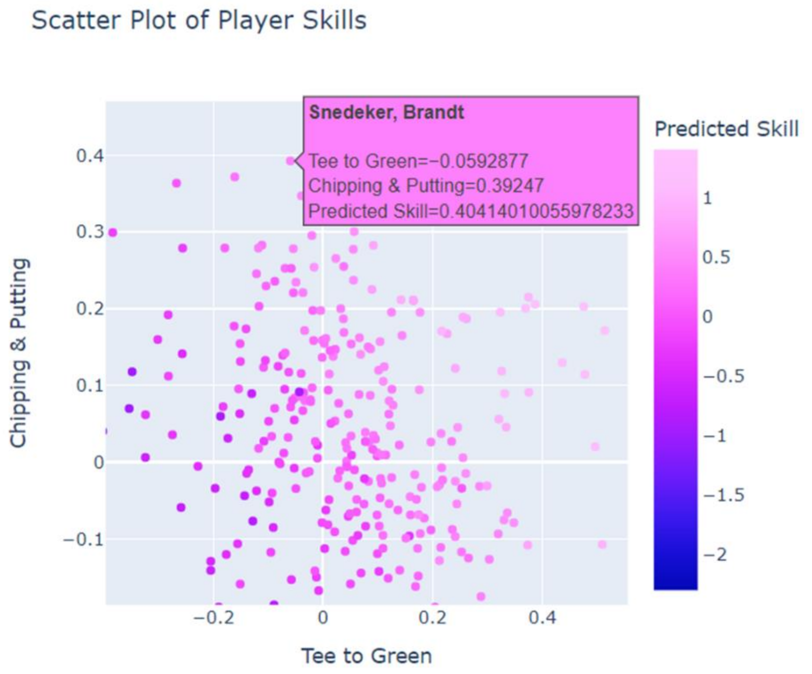
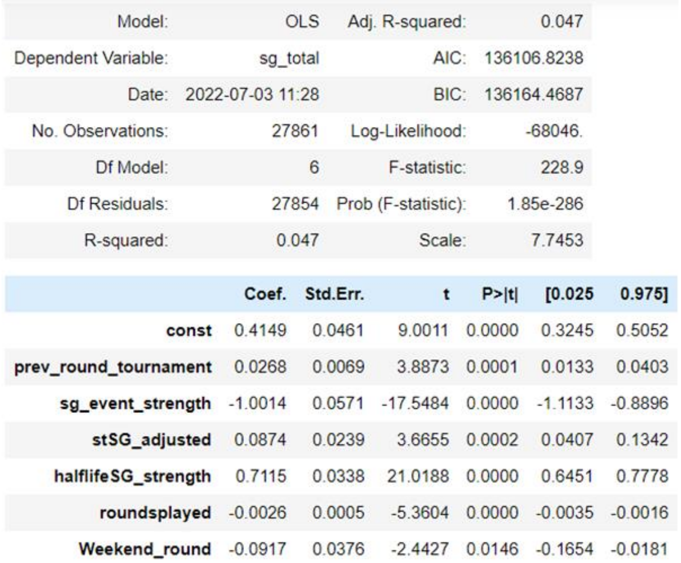


Figure 2 : Plot of player skill decomposition, where each player is a dot on the scatter plot. Tee to green uses "off the tee" and "approach" combined, while Chipping & Putting uses "around the green" and "putting" metrics.

Model Creation and Assessment

Creating the model was an iterative process. The simplest and, in the end, most effective technique was using a linear regression model to predict round performance. For this step, the goal was creating a simple and explainable model. So, the linear regression was tuned by removing factors with low p-values. Then, once the model was simplified down to the most "predictive" elements, it was scored on the test set. Figure 3 shows the model metrics and coefficients.



Model:	OLS	Adj. R-squared:	0.047
Dependent Variable:	sg_total	AIC:	136106.8238
Date:	2022-07-03 11:28	BIC:	136164.4687
No. Observations:	27861	Log-Likelihood:	-68046.
Df Model:	6	F-statistic:	228.9
Df Residuals:	27854	Prob (F-statistic):	1.85e-286
R-squared:	0.047	Scale:	7.7453

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
const	0.4149	0.0461	9.0011	0.0000	0.3245	0.5052
prev_round_tournament	0.0268	0.0069	3.8873	0.0001	0.0133	0.0403
sg_event_strength	-1.0014	0.0571	-17.5484	0.0000	-1.1133	-0.8896
stSG_adjusted	0.0874	0.0239	3.6655	0.0002	0.0407	0.1342
halflifeSG_strength	0.7115	0.0338	21.0188	0.0000	0.6451	0.7778
roundsplayed	-0.0026	0.0005	-5.3604	0.0000	-0.0035	-0.0016
Weekend_round	-0.0917	0.0376	-2.4427	0.0146	-0.1654	-0.0181

Figure 3: Model breakdown from StatsModels module. [stSG_adjusted] is short-term strokes gained, [halflifeSG_strength] is long-term strokes gained. The important metric is R-squared (a measure of the model's accuracy).

So, what do these numbers mean? In the end, the model weighted **event strength** and **long-term historical strokes gained performance** the highest. This makes sense because a higher event strength means the average score is going to be better. The average score is what is used to calculate strokes gained, thus making it harder to perform well (in terms of strokes gained). Strong historical strokes gained strength, on the other hand, means a player is better in general and will probably perform better in each round as well.

What about the accuracy of the model? The accuracy (as measured by the R-squared) is very low. This means that the model is **not confident** in its predictions. According to the metrics, the model prediction will be within ~4.4 strokes of the actual, which is not very good. Luckily, a golf tournament is played over 4 rounds. This means that the predictions are going to be more accurate at a **tournament level**. Figure 4 shows the distribution of the predicted strokes compared to the actual strokes gained average as a golfer plays more rounds.

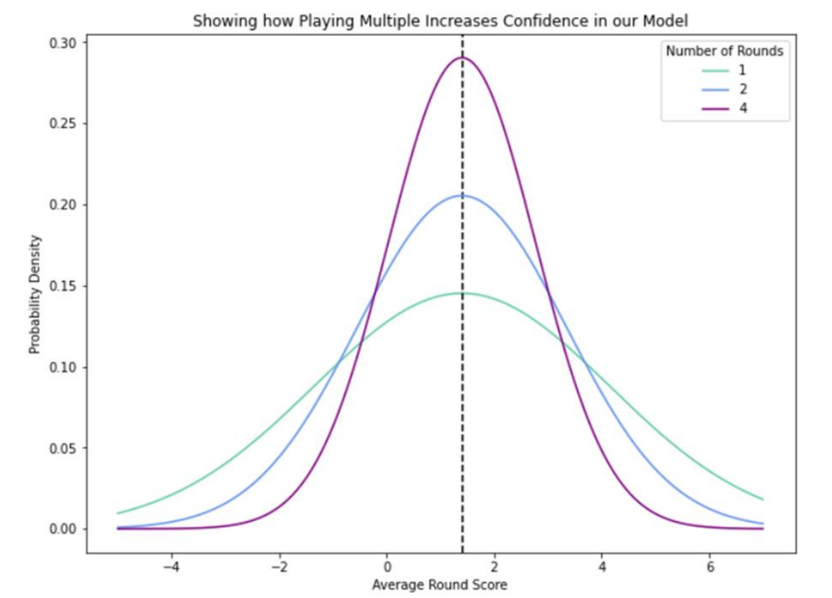


Figure 4: Distribution of average scores, for a golfer with a predicted strokes gained of 1.4 strokes. By the end of the tournament, the strokes gained average is much more likely to be near to the predicted.

Leveraging the Model to Understand Course Fit

Now that we have a baseline model, it is important to show how it can be leveraged to create insights into professional golf. For this example, the model is used to understand “course fit”. **Course fit** is the idea that a certain course favors a certain type of player. For example, you would expect that a longer course would be better for a player who hits the ball further. Unfortunately, the course information is not very specific and often fails to truly assess how a course will play. But, since we have a model that creates predictions, we can look at the predictions for a given tournament, measure how “wrong” the model was (the residuals) and see if there was any correlation between player skills and these residuals. To see this correlation, a linear regression is run on the residuals using the player decomposition factors created earlier. And, often, there is a relationship that is present. So, we know that in some tournaments a certain type of player **performed** better (emphasis on past tense). But is there any way to predict this ahead of time?

Luckily, the PGA Tour returns to the same courses quite often. So, we can use the previous year's course fit to predict the current year's performances. The results are, unsurprisingly, very noisy and not incredibly accurate. Yet, there seems to be some sort of (slight) relationship present.

Conclusion and Further Improvements

In the end, the results from the model were less accurate than I was expecting. After the initial data processing and inspection, I began to understand the results were going to be noisy. The model can be used to create some interesting insights, but it would surely be improved with more data, especially when dealing with course and tournament metrics. Although the course fit work was interesting, I know there is work to be done here. For example, the assumption that round to round variance is the same at all courses is probably not true. I would venture to guess that course has some sort of relationship to score variance.

In terms of other improvements, I would want to have more detailed and specific course information in order to create further insights on the course fit. If it were available, you could do some interesting work with clustering algorithms. Then, you would be able to leverage the results to create some more interesting insights. Also, if more data were present, I would do more inspection on how much better a more complex model would perform (for example, a gradient boosting regression tree). My assumption is that, with enough data, a more complex model would be able to capture ideas like course fit within the predictions. Again, as it stands currently, 4 years of data and limited course and shot level information means that this is not possible.