PGA Data Analysis

Matthew Solone

5/7/2022

Data Science in Golf has become one of the most important tools for not only analysts for broadcasting systems but also for companies and players. The players and golf companies use data to develop new strategies for themselves to improve their game.

In this post I will go over some simple analytics used to evaluating and predicting performance of the top players in the PGA. I have obtained the raw data from this website; (https://www.advancedsportsanalytics.com/pga-user-guides/#pga-optimizer-app)

The data is from Shotlink data which is the primary data collection application used by the PGA (https://www.pgatour.com/stats/academicdata/shotlink.html) . Shot link uses ’Strokes Gained(SG) statistics which are calculated using mathematical equations that incorporate player performance with the ShotLink® data collected by volunteers at each PGA TOUR tournament. SG is a better measure of performance compared to older gold statistics because it can isolate a players performance and compare to other players in the field. As an example on how SG is calculated, the PGA says this about SG: Putting - The statistic is computed by:

Ex: Average number of putts to hole out from 7 feet, 10 inches is 1.5. If a player one-putts from that distance, he gains 0.5 strokes. If he two-putts, he loses 0.5 strokes. If he three-putts, he loses 1.5 strokes.

There are more SG stats as well:

“The new strokes gained statistics, which were introduced June 1, 2016, break down tee-to-green play into three categories: off-the-tee, approach-the-green and around-the-green. The sum of those three statistics equals strokes gained is tee-to-green.

Off-the-tee + approach-the-green + around-the-green + putting = strokes gained: total

Strokes Gained: Off-the-Tee measures player performance off the tee on all par-4s and par-5s.

Strokes Gained: Approach-the-Green measures player performance on approach shots. Approach shots include all shots that are not from the tee on par-4 and par-5 holes and are not included in strokes gained: around-the-green and strokes gained: putting. Approach shots include tee shots on par-3s.

Strokes Gained: Around-the-Green measures player performance on any shot within 30 yards of the edge of the green. This statistic does not include any shots taken on the putting green”

From these statistics I believe we can accurately predict a players ability to make the cut, ’made_cut‘(1 being made the cut and 0 being did not make the cut), based on the SG statistics. To do this we will fit a logistic regression model to the data and try and find correlation between the variables.

- First lets look at some simple plots of our data and tidy it up a bit. Also remember to load in the librarys we will be using.

```
library(tidymodels)
library(tidyverse)
library(tidyr)
library(readr)
library(broom.mixed)
library(ISLR2)
library(discrim)
library(corrplot)
library(stargazer)
library(discrim)
```

- Fitting our model

```
library(parsnip)
pga_raw$made_cut <- factor(pga_raw$made_cut)

pga_fit <- logistic_reg() %>% set_engine("glm") %>% set_mode("classification") %>% fit(made_cut ~ sg_pu

tidy(pga_fit)
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)    0.487    0.0366      13.3 2.74e- 40
## 2 sg_putt        1.02     0.0398      25.6 3.59e-144
## 3 sg_app         0.977    0.0400      24.4 6.58e-132
## 4 sg_ott         0.874    0.0520      16.8 2.57e- 63
```

- From our initial logistic fit we can see that using the three main strokes gained statistics are all statistically significant and can be used in predictions.
- From here we can predict the outcome and visualize using a confusion matrix.

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
pred_log <- predict(pga_fit, pga_raw)

confusionMatrix(pga_raw$made_cut,pred_log$.pred_class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1479  669
##          1  458 2156
##
##                Accuracy : 0.7633
##                  95% CI : (0.751, 0.7753)
##     No Information Rate : 0.5932
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5179
##
##  Mcnemar's Test P-Value : 3.964e-10
##
##             Sensitivity : 0.7636
##             Specificity : 0.7632
##          Pos Pred Value : 0.6885
##          Neg Pred Value : 0.8248
##              Prevalence : 0.4068
##          Detection Rate : 0.3106
##    Detection Prevalence : 0.4511
##       Balanced Accuracy : 0.7634
##
##        'Positive' Class : 0
##
```

- The confusion matrix shows a a true positive rate of 0.7636 and a true negative rate of 0.7632 which is not too bad but ideally we would be in the 90% range. So in total we are correct about 76.33% of the time. We correctly predicted making the cut about 82.48% of the time in and correctly predicted not making the cut 68.85 % of the time.

Bio -

My name is Matthew Solonem, I am currently an incming Senior at California State University, Chico. Currently pursing a degree in Computational Mathematics and Statistics and a certificate in Data Science. My goal is to use data science to provide meaningful analysis to help drive better buisiness decisions.

LinkedIn - https://www.linkedin.com/in/matthewsolone/

GitHub - https://github.com/mjsolone

Email - mattsolone18@gmail.com