Golf Machine Learning Project: Predicting Successful Players in the PGA Tour with TidyModels

Thomas Schechter and Ian Bogley

12/10/2021

Golf is a sport with diverse and detailed statistics, making it ideal grounds for prediction and analysis. The purpose of this project will be to use machine learning algorithms to predict whether a player will be successful or not. To do so, we will use data from the PGA 2017 Tour.

In particular, our question will be specified as follows: What players will finish in the top 10 of an event at least once? (Is this accurate phrasing? Is this a good question? May reduce variability of our outcome variable by quite a bit in comparison to predicting number of top 10 finishes. Number of players who never managed to break into the top 10 are 34, while the number who did at some point is 161. This means that with a k-fold cross-validation approach where k=5, the probability of not having a False observation in one of the 5 folds is:[I forgot my probability stuff, but I think we might have a problem here.])

To start, let's begin by reading in the data and install necessary packages.

```r
#Package unloading
library(pacman)
p_load(tidyverse,data.table,tidymodels,stargazer,ggpubr,
       janitor,sandwich,xtable)

#Read in data
pga_data17 <- fread("PGATOUR_data2.csv")
```

Now, we will clean the data, subsetting to remove null and NA values.

```r
#Subset data
pga_data <- pga_data17[1:195,] %>% clean_names() %>%
  mutate(top_ten_finisher = (number_of_top_tens>0))

#check for further na values
lapply(pga_data,FUN = function(x) {sum(is.na(x))}) %>%
  as.data.frame() %>% melt() %>% rename("Var" = "variable","NAs"="value") %>%
  filter(NAs>0) %>% xtable()
```

% latex table generated in R 4.0.3 by xtable 1.8-4 package % Sat Dec 18 13:36:41 2021

	Var	NAs
1	points_behind_lead	1

```r
#There is one missing value in the "points behind lead" column, we will use the mean value to impute
pga_data$points_behind_lead[33] <- mean(pga_data$points_behind_lead,na.rm = T)
```

Now, let's graph potential factors in top ten finishes and look for correlations.

```
#Begin exploring the relationships in the data

#Correlation b/w rounds played and top tens (Is this looking for issues in representation? Players with
model1 <- lm(rounds_played~number_of_top_tens, pga_data)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sat, Dec 18, 2021 - 1:36:41 PM

Table 1:

	Dependent variable:
	rounds_played
number_of_top_tens	1.650***
	(0.443)
Constant	73.985***
	(1.526)
Observations	195
R^2	0.067
Adjusted R^2	0.062
Residual Std. Error	13.596 (df = 193)
F Statistic	13.847*** (df = 1; 193)
Note:	*p<0.1; **p<0.05; ***p<0.01

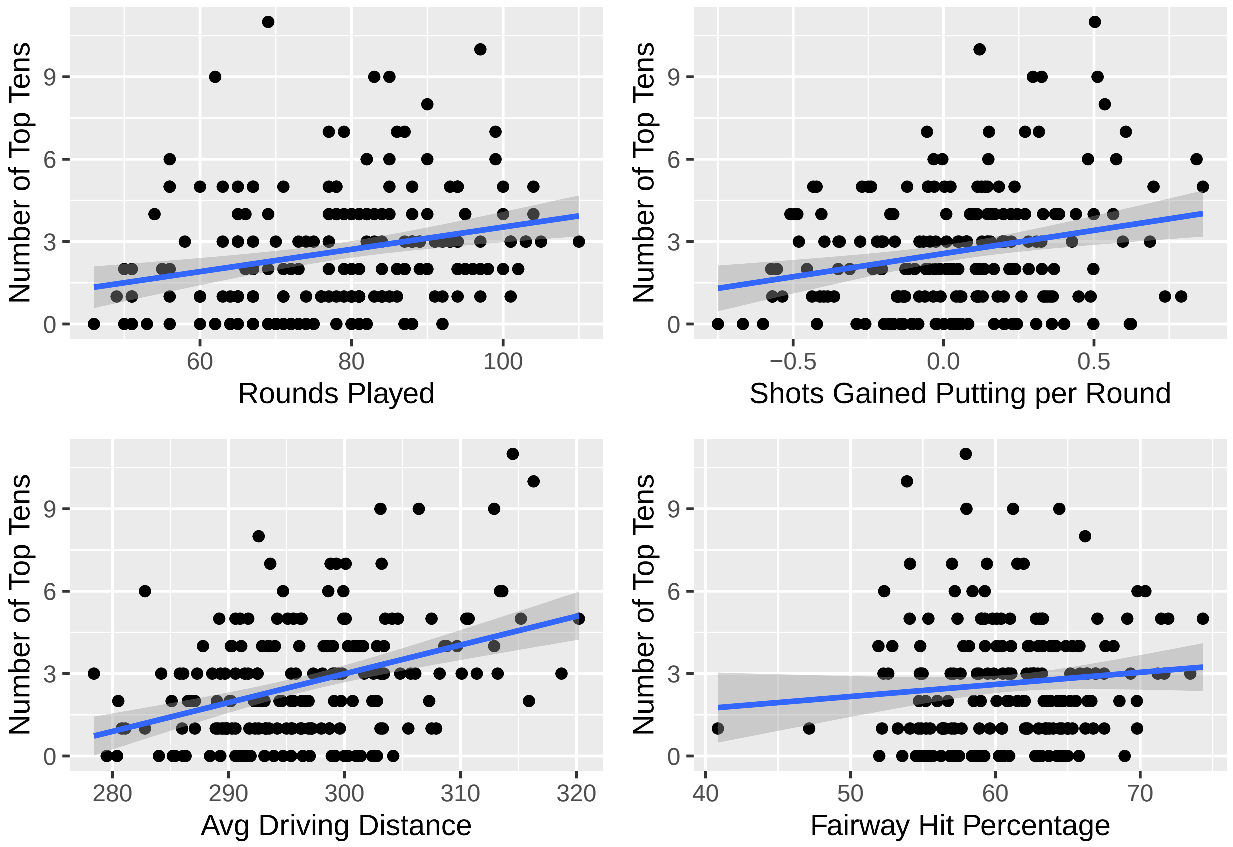
```
#Plot the rounds played relationship
round_plot <-  ggplot(pga_data, aes(rounds_played, number_of_top_tens)) +
  geom_point() +  geom_smooth(method = "lm") +
  labs(x = "Rounds Played",
       y = "Number of Top Tens")

#Plotting strokes gained putting vs. top ten finishes
putt_plot <- ggplot(pga_data, aes(sg_putting_per_round, number_of_top_tens)) +
  geom_point() +  geom_smooth(method = "lm") +
  labs(x = "Shots Gained Putting per Round",
       y = "Number of Top Tens")

#Plotting average drive distance vs. top ten finishes
drive_plot <- ggplot(pga_data, aes(avg_driving_distance,number_of_top_tens)) +
  geom_point() + geom_smooth(method = "lm") +
  labs(x = "Avg Driving Distance",
       y = "Number of Top Tens")

#Plotting Fairway hit percentage vs top ten finishes
fairway_plot <- ggplot(pga_data, aes(x=fairway_hit_percent,y = number_of_top_tens)) +
  geom_point() + geom_smooth(method = "lm")+
  labs(x = "Fairway Hit Percentage",
       y = "Number of Top Tens")
```

```
#Arrange previous plots into a single graphic
ggarrange(round_plot,putt_plot,drive_plot,fairway_plot)
```



Not a significant amount of visual correlation between number of rounds played and number of top ten finishes.

Strokes gained putting seems to have more of an effect on success.

Some relationship appears present in average driving distance, though with more outliers on both ends of the spectrum.

There seems to be even less of a correlation in terms of fairway hit percentage than the previous ones. Even if there is, it is much smaller of an impact than the other factors examined.

Let's examine these relationships as linear regressions, identifying the impact of each and the robustness of the coefficients.

```
for (i in 1:4) {
  #vector of variable names
  variables <- c("rounds_played","sg_putting_per_round",
                 "avg_driving_distance","fairway_hit_percent")

  #models for each variable
  eval(parse(text = paste("model_",i+1,
                          " <- lm(number_of_top_tens ~ ",variables[i],
                          ",pga_data)",sep = "")))

  #robust se for each variable
  eval(parse(text = paste("robust_se_",i+1,
                          " <- list(sqrt(diag(vcovHC(model_",i+1,"))))",
                          sep ="")))
}
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sat, Dec 18, 2021 - 1:36:43 PM

Table 2:

	Dependent variable:			
	number_of_top_tens			
	(1)	(2)	(3)	(4)
rounds_played	0.041***			
	(0.011)			
sg_putting_per_round		1.689***		
		(0.490)		
avg_driving_distance			0.105***	
			(0.018)	
fairway_hit_percent				0.044
				(0.031)
Constant	−0.528	2.564***	−28.463***	−0.036
	(0.868)	(0.155)	(5.288)	(1.901)
Observations	195	195	195	195
R^2	0.067	0.058	0.152	0.010
Adjusted R^2	0.062	0.053	0.148	0.005
Residual Std. Error (df = 193)	2.132	2.142	2.032	2.196
F Statistic (df = 1; 193)	13.847***	11.874***	34.642***	2.011
Note:		*p<0.1; **p<0.05; ***p<0.01		

We can also go further, collecting the variables with the highest correlation to our dependant variable.

```
cor_coef <- lapply(pga_data[,-c(1,4,5,70)],
                   FUN = function(x) {cor(x=as.numeric(x),
                                          y = pga_data$top_ten_finisher)})

top_var <- cor_coef[order(abs(unlist(cor_coef)),decreasing=T)[1:20]] %>%
  as.data.frame() %>% gather() %>%
  rename(var = "key", cor = "value")
```

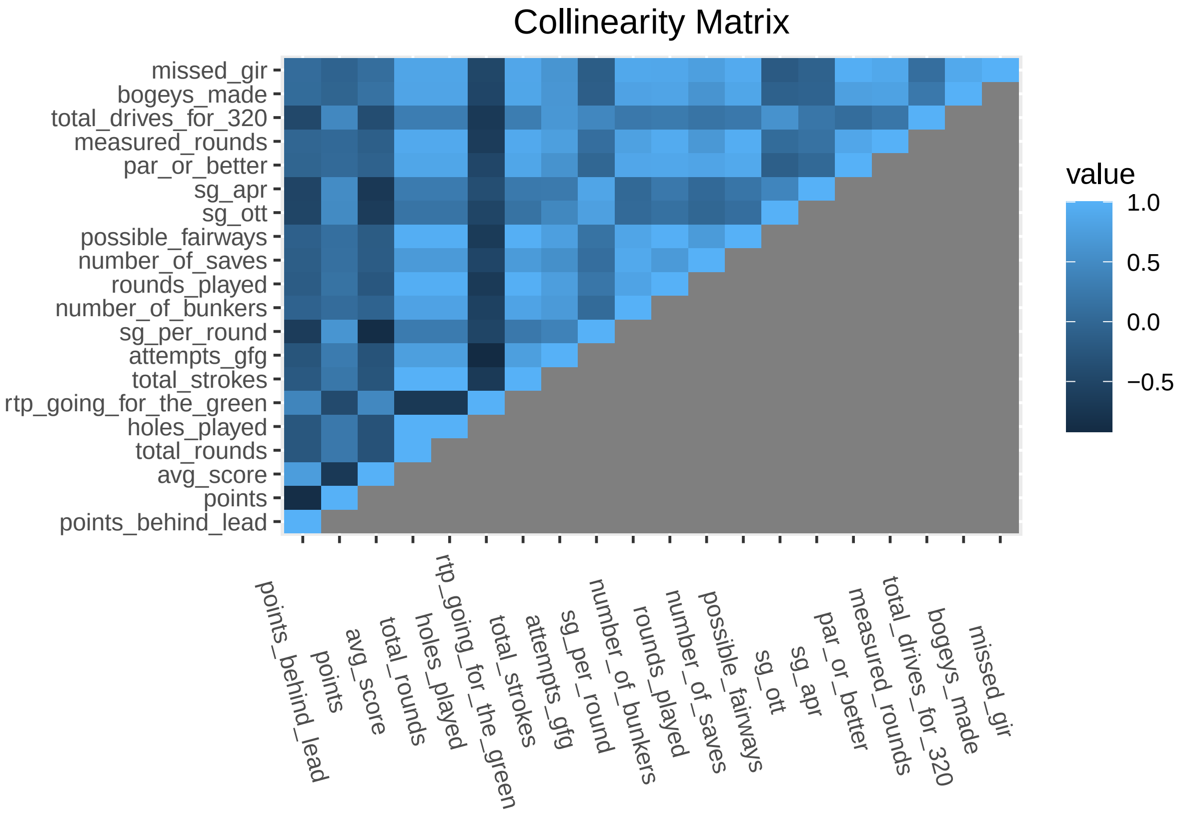
% latex table generated in R 4.0.3 by xtable 1.8-4 package % Sat Dec 18 13:36:43 2021

Now we have a set of variables which are highly correlated with our explanatory variable. Now let's check for colinearity between these variables.

	var	cor
1	points_behind_lead	-0.41
2	points	0.37
3	avg_score	-0.35
4	total_rounds	0.34
5	holes_played	0.33
6	rtp_going_for_the_green	-0.33
7	total_strokes	0.33
8	attempts_gfg	0.32
9	sg_per_round	0.32
10	number_of_bunkers	0.32
11	rounds_played	0.31
12	number_of_saves	0.31
13	possible_fairways	0.29
14	sg_ott	0.29
15	sg_apr	0.27
16	par_or_better	0.26
17	measured_rounds	0.25
18	total_drives_for_320	0.24
19	bogeys_made	0.23
20	missed_gir	0.22

```r
top_var_data <- pga_data %>% select(top_var$var)
top_var_cormat <- round(cor(top_var_data),2)
top_var_cormat[lower.tri(top_var_cormat)] <- NA

top_var_cormat %>% melt() %>%
  ggplot(aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() + theme(axis.text.x= element_text(angle = -75),
                      plot.title=element_text(hjust = .5)) +
  labs(title = "Collinearity Matrix") + xlab("") + ylab("")
```



We can see that there are quite a few variable pairings that may present a danger. Let's take the mean absolute correlation coefficient, then take a look at all the pairs of explanatory variables which have an absolute correlation coefficient above it. We will use this as a cutoff, meaning that these variable pairings are ones we want to avoid in our models.

```
top_var_avoid <- top_var_cormat %>% melt() %>%
  filter((abs(value)>abs(mean(value,na.rm=T)))&!value==1) %>%
  arrange(-abs(value))
```

For presentation purposes, we will only show the first 10 pairs.

% latex table generated in R 4.0.3 by xtable 1.8-4 package % Sat Dec 18 13:36:44 2021

	Var1	Var2	value
1	rounds_played	possible_fairways	0.98
2	total_strokes	rounds_played	0.97
3	total_strokes	possible_fairways	0.97
4	total_rounds	rounds_played	0.96
5	holes_played	rounds_played	0.96
6	total_rounds	possible_fairways	0.96
7	holes_played	possible_fairways	0.96
8	par_or_better	missed_gir	0.96
9	possible_fairways	measured_rounds	0.95
10	rtp_going_for_the_green	attempts_gfg	-0.92

When creating our models, we will need to ensure that no two variables in the same row are included in the same model. To do so, let's create a list of variable pairings that cannot be put together.

```
#top_var_avoid[,-3]
```

For our explanatory variable, we have a classification problem. We have decided to make it a binary variable detailing whether a player is ever able to make it into the top 10 of an event.