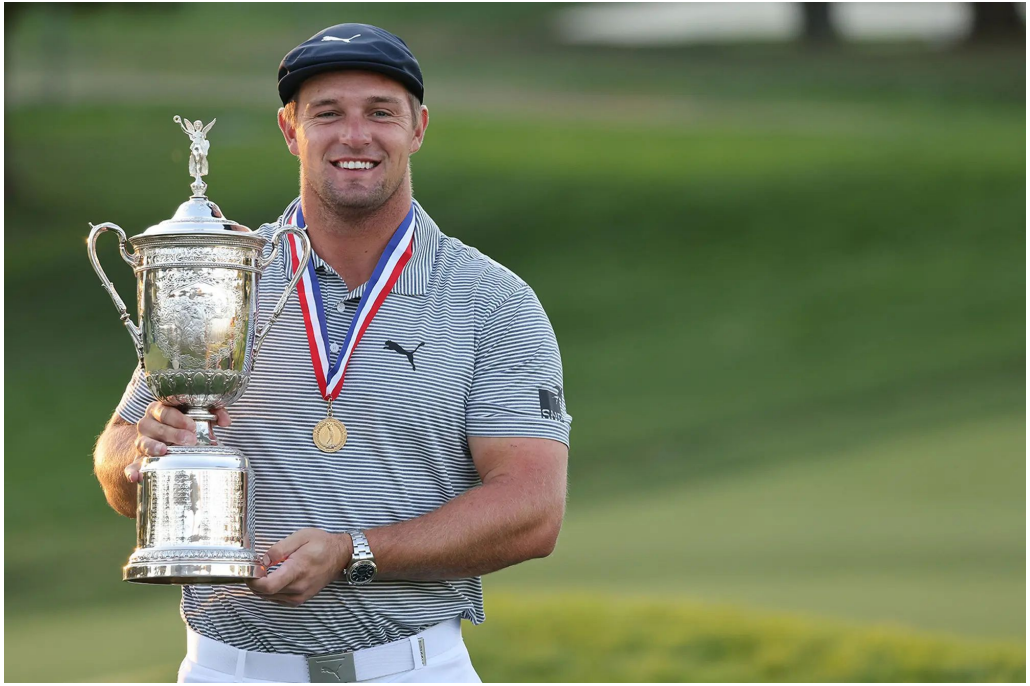
Sunday Red E.3

by Ben

Why Sunday Red E.



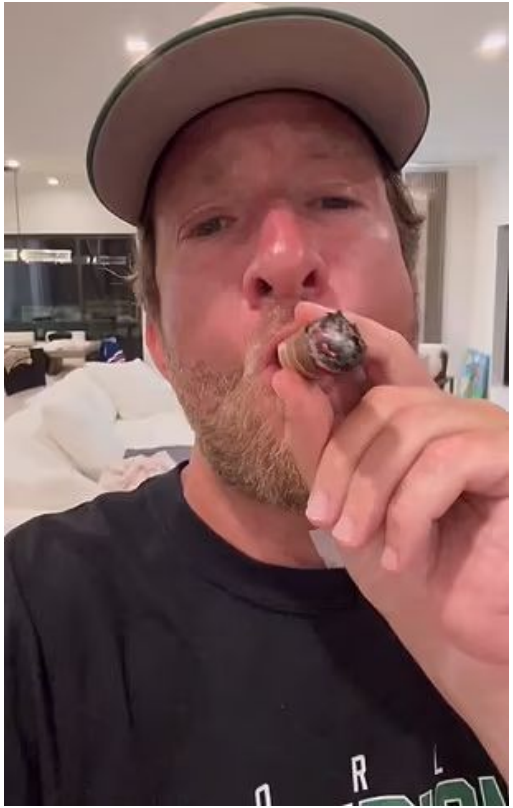
Bryson DeChambeau at the 2024 U.S. Open after winning his second Major (and U.S. Open)

Table of Contents

What am I looking to do

- Identify correlations within the metrics of golf
- See if I can determine a way to predict the leaderboard of a field of players in the Majors (+ Players)
- Figure out if data can be leveraged to find out who makes the weekend (cut) and who will finish at number 1 on Sunday

Who is this for?



Receipt 1 of 1 bets placed

BET PLACED

LIVE **Scottie Scheffler** +180

WINNER

Travelers Championship 2024

Notwithstanding any other house rule(s) and unless the bet is not already settled, in the event there is a reduction in scheduled rounds played, wagers placed on this market will be voided if either: (1) less than 36 holes of the tournament have been completed by all remaining golfers; or (2) the wager(s) are placed after the final shot of the most recently completed round.

Wager Amount:	$180,000.00
Total Payout:	$504,000.00

What tournaments am I analyzing?

- Since there are many tournaments/events, my scope is going to be targeting the top 5 events, both in prestige and purse
- The top events include:
  - The Masters - held in April and annually at Augusta National Golf Club in Augusta, Georgia
  - The PGA Championship - held in May at various courses
  - The US Open - held in June at various courses
  - The Open - held in July at various courses
  - The Players - unofficial 5th ‘Major’ held at TPC Sawgrass (since 1982) in Ponte Vedra Beach, Florida

What am I looking at

- Some examples of metrics I am looking at include but not limited to:
  - Tee time - when the golfer begins their round
  - Strokes gained putting - how well they putted compared to the average
  - Strokes gained off the tee - how well they drove the ball
  - Round score - their overall score for the round
  - Round number[1] - the day which they played
  - Event - name of tournament
  - Course[2] - which course was this tournament played at

1 If a player did not make the weekend, that means they were cut and will have NULL values for their finish and no data for rounds 3 and 4

2 The Masters and The Players are played at the same course each year

Looking at correlations

round_score	1.000000
score to par	0.968958
poor_shots	0.659505
prox_fw	0.428922
prox_rgh	0.137786
event_Masters Tournament	0.081056
event_U.S. Open	0.070299
course_par	0.069136
dg_id	0.012091
tee_time_of_day_int	0.008884
teetime_int	0.006309
start_hole	-0.011590
round_num	-0.012804
course_num	-0.024694
event_PGA Championship	-0.034399
year	-0.042288
event_The Open Championship	-0.052022
event_THE PLAYERS Championship	-0.053539
driving_dist	-0.169041
driving_acc	-0.270744
great_shots	-0.367471
sg_arg	-0.427967
sg_ott	-0.445060
scrambling	-0.485022
sg_putt	-0.537982
made_weekend	-0.561754
sg_app	-0.590598
gir	-0.609398
sg_t2g	-0.785877
sg_total	-0.925525

Name: round_score, dtype: float64

- I obtained the correlations of my metrics to round_score which I felt could be a better identifier for correlation (additionally the presence of nulls led to NaN values on my scatter plots)
- The Masters and U.S. Open, the two more competitive tournaments was shown to have a higher correlation with round score as overall scores are higher for these tournaments
- Strokes gained (sg) is a metric that takes into account a golfer's score directly which explains why most of the metrics show strong negative correlations (in golf, a lower score actually is better!)
- Metrics such as driving distance, accuracy, great/poor shots are also closely aligned to a player's performance and can generally help set them up for better positions

Multiple Linear Regression

- I decided to run a multiple linear regression analysis in order to determine whether a prediction on round score was possible
- Some things to note were there are a couple of metrics that were heavily correlated with round score as noted before so these would need to be addressed
- Initial data cleaning included dropping heavily correlated columns and including tournament dummy variables

```
# Step 1: Drop highly correlated features if they exist
columns_to_drop = ['score to par', 'sg_total']
df1_cleaned = df1.drop(columns=[col for col in columns_to_drop if col in df1.columns])

# Step 2: Encode categorical variables
df_encoded = pd.get_dummies(df1_cleaned, drop_first=False)

# Identify expected tournament dummy variables
expected_event_columns = ['event_Masters Tournament', 'event_PGA Championship', 'event_THE PLAYERS Championship',
                          'event_The Open Championship', 'event_U.S. Open']
```

Results

```
Mean Squared Error: 3.919863619862398
R-squared: 0.6858955998135142
```

After running my linear regression, I got an R^2 of 0.69 which means my model can explain 69% of the variability in my model.

Confusion Matrix results

- My model has an accuracy of 73.1% based on the matrix
- There seems to be a good amount positive predictions that were correct but the regression r-squared value of 0.54 means that there is definitely room for improvement
- There is a recall score of 0.77 for predicting players who made the cut however there are incorrect guesses for these as well due to a precision score of 0.72.

```
Classification Accuracy: 0.7305596830113917
Confusion Matrix:
 [[669 306]
 [238 806]]
Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.69      0.71       975
           1       0.72      0.77      0.75      1044

    accuracy                           0.73      2019
   macro avg       0.73      0.73      0.73      2019
weighted avg       0.73      0.73      0.73      2019

Regression Mean Squared Error: 3.638216354028099
Regression R-squared: 0.5408345188569122
```

Re-thinking my approach

- I looked at my model and while going through iterations, was finding very, VERY high accuracies (due to data leakage)
- I realized a column I calculated previously was a factor in this
- Additionally, the way my data was currently presented was not suitable for my end goal of predicting the leaderboard

So what does this mean?

I “changed” my target variable

Revamped model

[86]:
```
# Aggregating Data
aggregated_df = df_complete.groupby(['dg_id', 'event_name', 'course_num', 'year']).agg({
    'round_score': 'sum',
    'driving_dist': 'mean',
    'driving_acc': 'mean',
    'gir': 'mean',
    'scrambling': 'mean',
    'prox_rgh': 'mean',
    'prox_fw': 'mean',
    'great_shots': 'sum',
    'poor_shots': 'sum',
    'teetime_int': 'mean'
}).reset_index()

# Verify the aggregated DataFrame
print(aggregated_df.head())

# Train-Test Split
X = aggregated_df.drop(columns=['dg_id', 'round_score'])
y = aggregated_df['round_score']

# Convert categorical 'event_name', 'course_num', and 'year' to dummies
X = pd.get_dummies(X, columns=['event_name', 'course_num', 'year'], drop_first=False)

# Fill any remaining NaNs in the feature set with the median value of the respective column
X = X.fillna(X.median())
```

Revamped model (cont.)

```
dtype: int64
0
Test Set Mean Squared Error: 10.727980658646 74
Test Set R-squared: 0.7396326762785792
Training Set Mean Squared Error: 11.17378248737825
Training Set R-squared: 0.757200182654951
```

Revamped model (cont.)

```
def predict_leaderboard(event_name, year):
    # Filter the data for the specific event and year
    event_data = aggregated_df[(aggregated_df['event_name'] == event_name) & (aggregated_df['year'] == year)]

    # Ensure only players who completed all four rounds are included
    event_data = event_data.groupby('dg_id').filter(lambda x: len(x['round_score']) == 1)

    # Check if event_data is empty
    if event_data.empty:
        print(f"No data found for event: {event_name} and year: {year}")
        return None

    # Prepare the features
    event_features = event_data.drop(columns=['dg_id', 'round_score'])
    event_features = pd.get_dummies(event_features, columns=['event_name', 'course_num', 'year'], drop_first=False)
    event_features = event_features.reindex(columns=X.columns, fill_value=0)

    # Fill any remaining NaNs in the feature set with the median value of the respective column
    event_features = event_features.fillna(event_features.median())

    # Predict the scores
    event_data['predicted_score'] = reg.predict(event_features)

    # Sort players by predicted score to simulate a leaderboard
    predicted_leaderboard = event_data.sort_values(by='predicted_score').reset_index(drop=True)

    # Add player names using the player dictionary
    predicted_leaderboard['player_name'] = predicted_leaderboard['dg_id'].map(player_dict)

    # Select relevant columns and show top 10 players
    predicted_leaderboard = predicted_leaderboard[['dg_id', 'player_name', 'event_name', 'year', 'predicted_score']].head(10)

    return predicted_leaderboard

def get_actual_top_10(event_name, year):
    # Filter the data for the specific event and year
    actual_data = df[(df['event_name'] == event_name) & (df['year'] == year)]

    # Ensure only players who completed all four rounds are included
    actual_data = actual_data.groupby('dg_id').filter(lambda x: len(x['round_num'].unique()) == 4)
```

Results

- My resulting model takes inputs of event_name and year in order to predict the top 10 for previous tournaments (sans 2020 Masters which was cut short due to COVID and the 2020 Open which was canceled)

Predicted Leaderboard:

	dg_id	player_name	event_name	year	predicted_score
0	16243	Koepka, Brooks	Masters Tournament	2019	284.589882
1	5321	Woods, Tiger	Masters Tournament	2019	284.848329
2	19895	Schauffele, Xander	Masters Tournament	2019	285.141236
3	12422	Johnson, Dustin	Masters Tournament	2019	285.244615
4	7655	Molinari, Francesco	Masters Tournament	2019	285.261845
5	7672	Oosthuizen, Louis	Masters Tournament	2019	285.339379
6	9771	Day, Jason	Masters Tournament	2019	285.356609
7	19195	Rahm, Jon	Masters Tournament	2019	285.408298
8	12965	Fowler, Rickie	Masters Tournament	2019	285.459988
9	1547	Mickelson, Phil	Masters Tournament	2019	285.589211

Actual Top 10 Leaderboard:

	dg_id	player_name	event_name	year	round_score
3	5321	Woods, Tiger	Masters Tournament	2019	275
33	12422	Johnson, Dustin	Masters Tournament	2019	276
53	16243	Koepka, Brooks	Masters Tournament	2019	276
61	19895	Schauffele, Xander	Masters Tournament	2019	276
28	11676	Finau, Tony	Masters Tournament	2019	277
24	11049	Simpson, Webb	Masters Tournament	2019	277
20	9771	Day, Jason	Masters Tournament	2019	277
16	7655	Molinari, Francesco	Masters Tournament	2019	277
51	15466	Cantlay, Patrick	Masters Tournament	2019	278
36	12965	Fowler, Rickie	Masters Tournament	2019	278

Percentage of Correct Predictions: 70.00%

DEMO

Tiger Woods wins 2000 U.S. Open leading by 15 strokes (second place was at +3)

citation

https://www.dailymail.co.uk/sport/golf/article-13558595/barstool-dave-portnoy-bets-scottie-scheffler-travelers-golf.html

https://www.dailymail.co.uk/sport/golf/article-13558595/barstool-dave-portnoy-bets-scottie-scheffler-travelers-golf.html

https://www.youtube.com/watch?v=bado2QdgD3c&ab_channel=UnitedStatesGolfAssociation%28USGA%29

Next steps aka this weekend

- Iterate further, potentially adding in other tournament data
- Develop model further, debug and incorporate future predictability
- Complete write-ups, finalize for Demo Day

That’s all folks!
Any questions?