Sunday Red E.2

by Ben

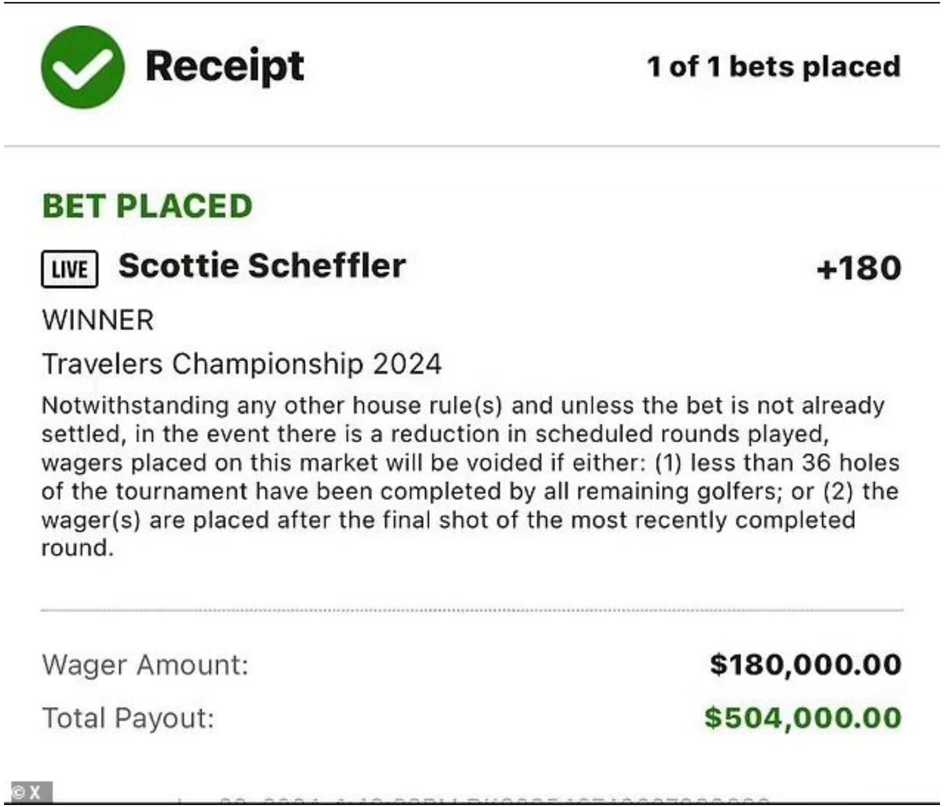
Why Sunday Red E.

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

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**

© X

What is golf?

“Golf is a club-and-ball sport in which players use various clubs to hit a ball into a series of holes on a course in as few strokes as possible.”[1]

[1] Wikipedia contributors. "Golf." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 25 May. 2024. Web. ~~https://en.wikipedia.org/wiki/Golf~~.

What does “make the weekend” mean?

- A standard round of golf consists of 18 holes played with a par (number of strokes) of 72
- In most golf tournaments on the PGA Tour (including the Majors), the standard is to play 4 rounds of golf, generally beginning on Thursday and ending on Sunday
- After the first 2 rounds of golf are completed (Thursday and Friday), the field is evaluated and the top-XX players on the leaderboard are moved onto the weekend
- Anyone that does NOT make the cut is finished for the tournament and does not play out the last 2 rounds

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

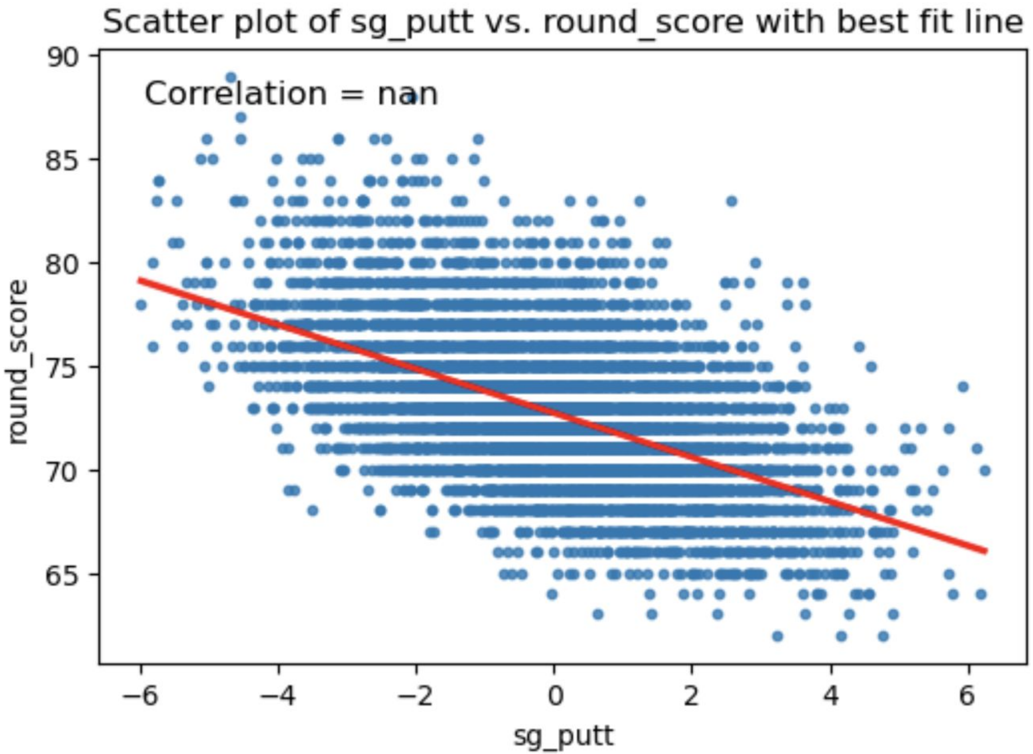
What am I looking at

- 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

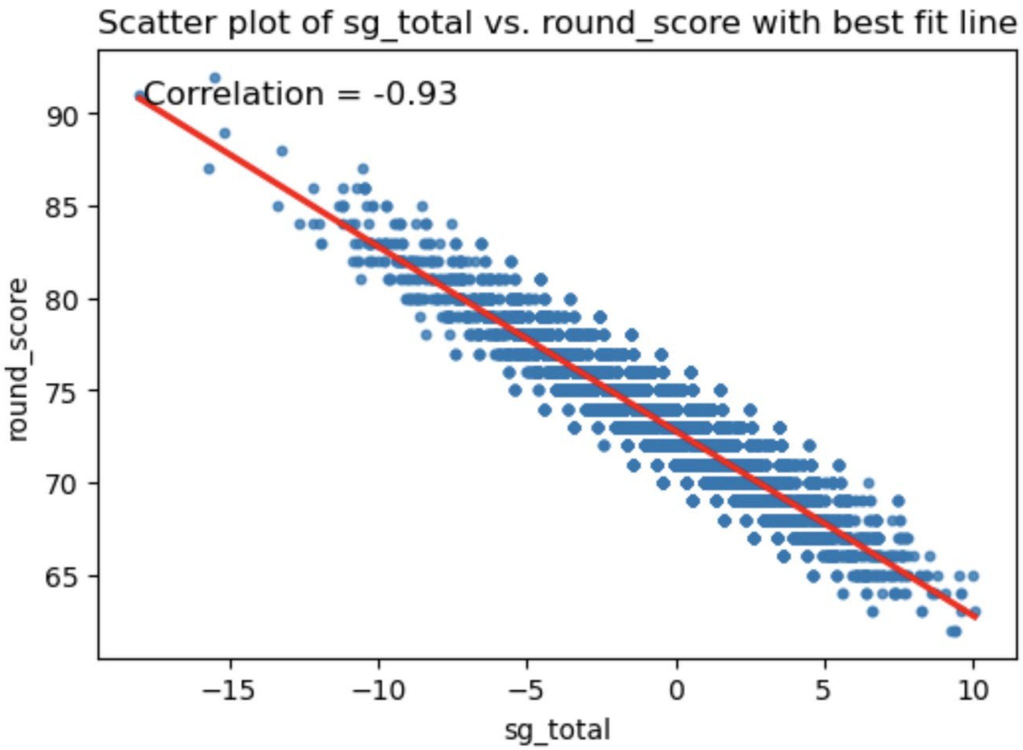
[1] If a player did not make the weekend, that means they were cut and will have NULL values for their finish and no data for rounds 3 and 4

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

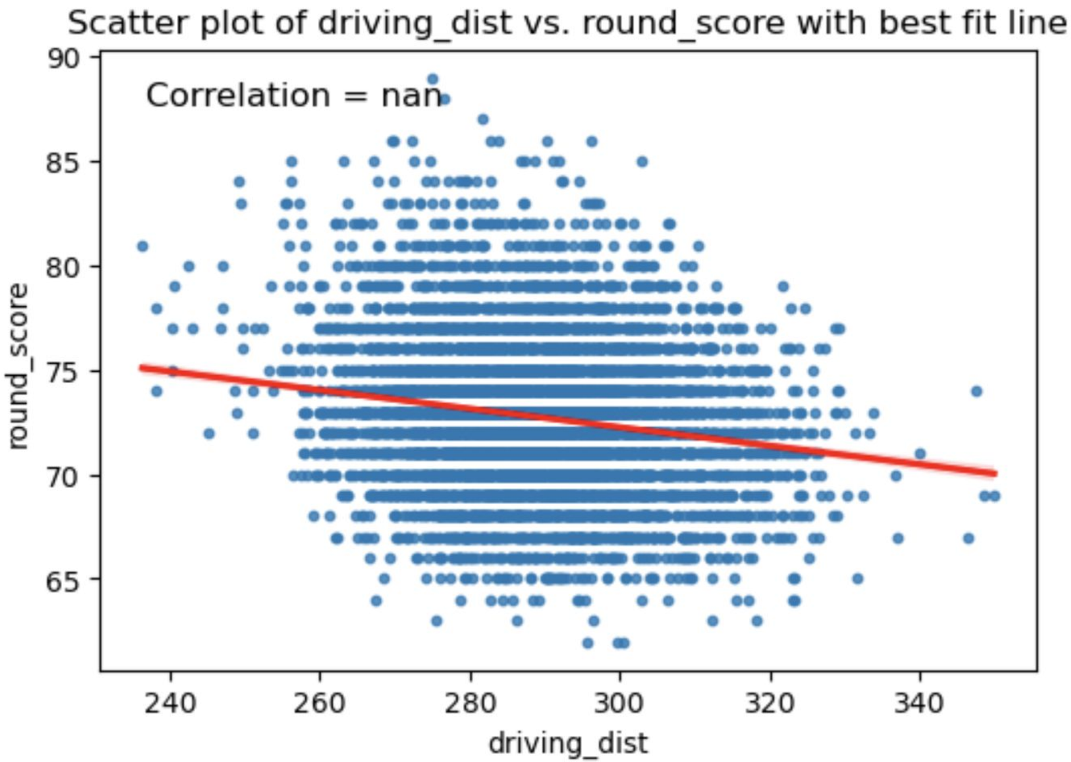
Insights cont.



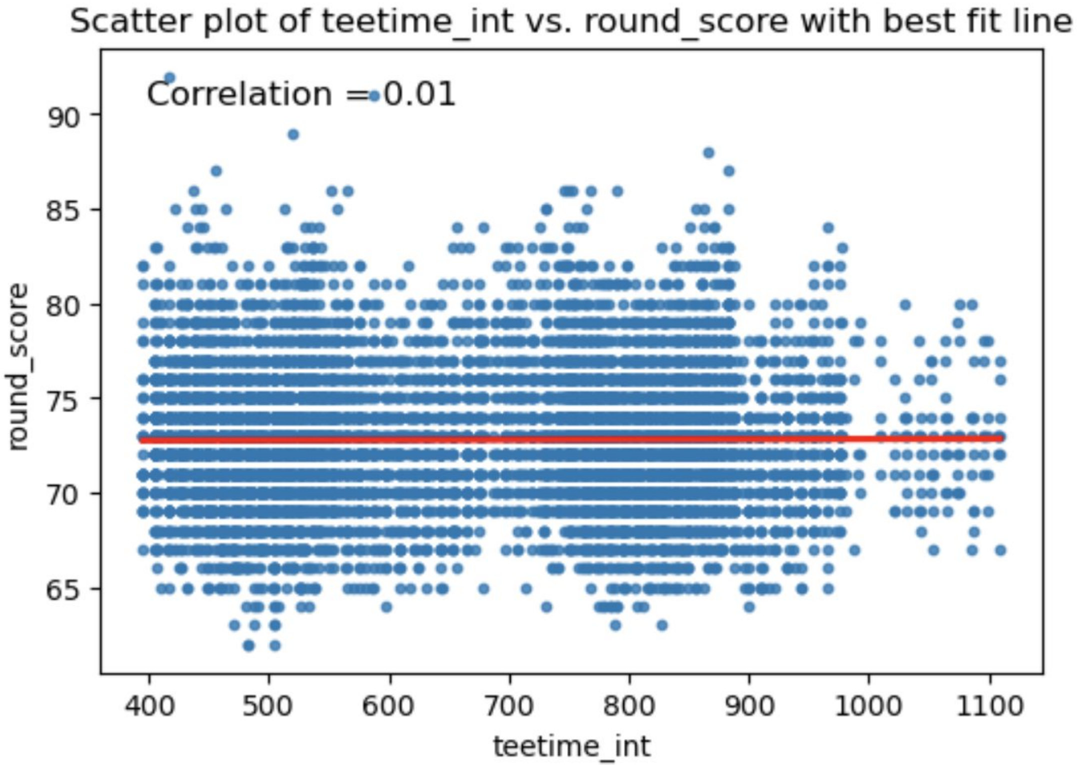
Insights cont.



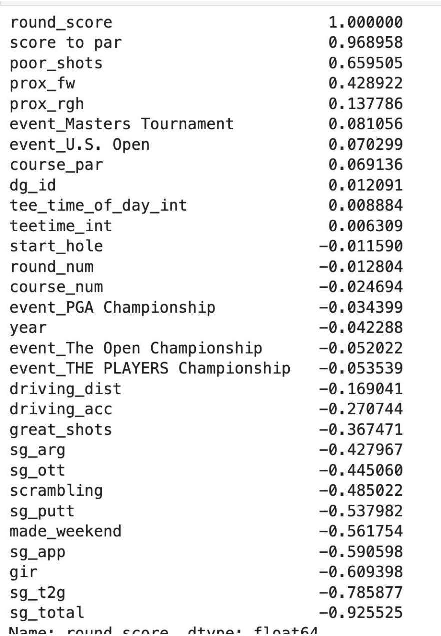
Insights cont.



Insights cont.



Looking at correlations

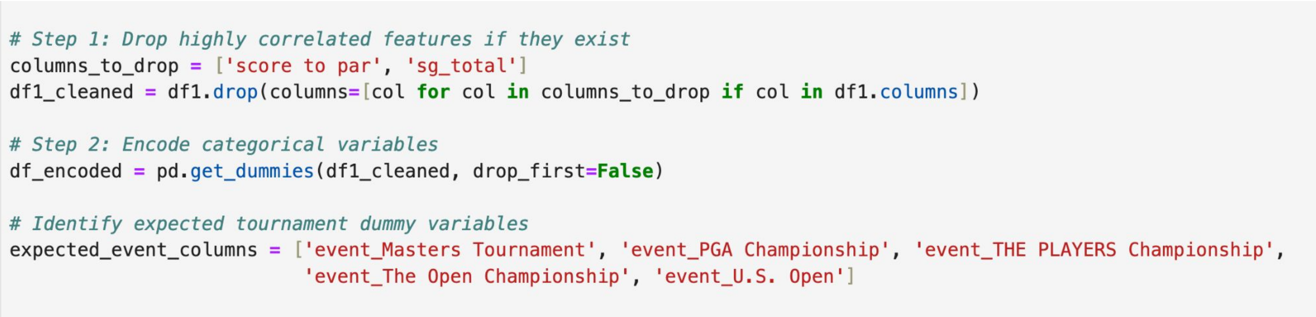


```
round_score                       1.000000
score to par                      0.968958
poor_shots                        0.659505
prox_fw                           0.428922
prox_rgh                          0.137786
event_Masters Tournament          0.081056
event_U.S. Open                   0.070299
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

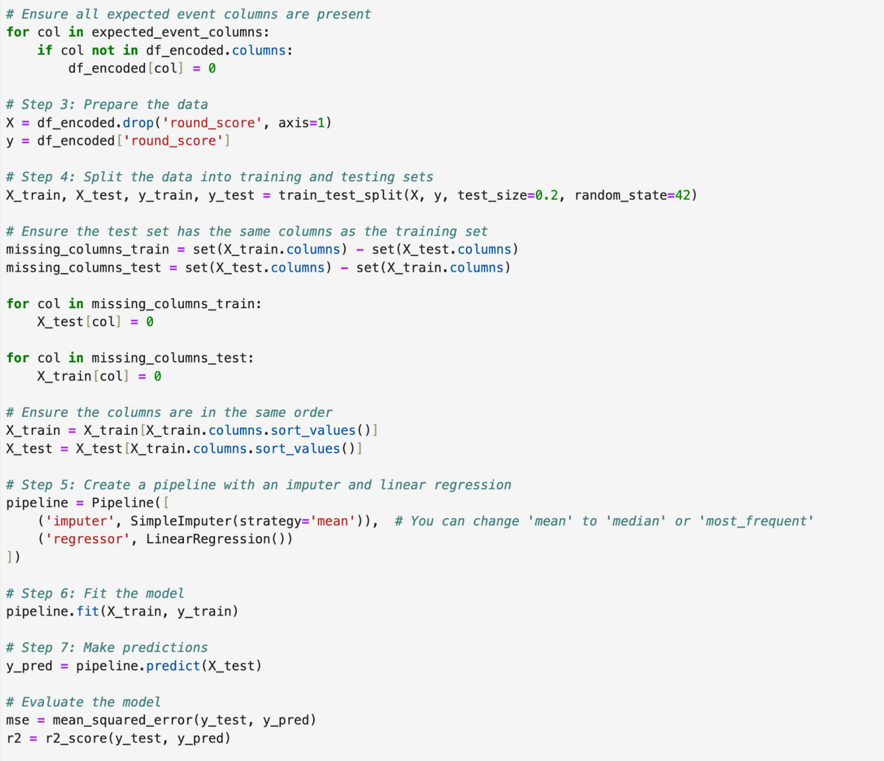
Multiple Linear Regression

- 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



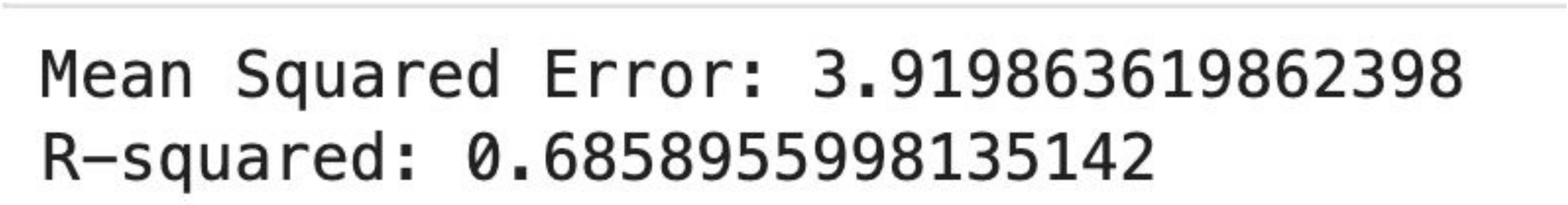
```
# Step 1: Drop highly correlated features if they exist
columns_to_drop = ['score to par', 'sg_total']
df1_cleaned = df1.drop(columns=[col for col in columns_to_drop if col in df1.columns])

# 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']
```

Multiple Linear Regression (cont.)



```
# Ensure all expected event columns are present
for col in expected_event_columns:
    if col not in df_encoded.columns:
        df_encoded[col] = 0

# Step 3: Prepare the data
X = df_encoded.drop('round_score', axis=1)
y = df_encoded['round_score']

# Step 4: Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Ensure the test set has the same columns as the training set
missing_columns_train = set(X_train.columns) - set(X_test.columns)
missing_columns_test = set(X_test.columns) - set(X_train.columns)

for col in missing_columns_train:
    X_test[col] = 0

for col in missing_columns_test:
    X_train[col] = 0

# Ensure the columns are in the same order
X_train = X_train[X_train.columns.sort_values()]
X_test = X_test[X_train.columns.sort_values()]

# Step 5: Create a pipeline with an imputer and linear regression
pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy='mean')),  # You can change 'mean' to 'median' or 'most_frequent'
    ('regressor', LinearRegression())
])

# Step 6: Fit the model
pipeline.fit(X_train, y_train)

# Step 7: Make predictions
y_pred = pipeline.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

Results

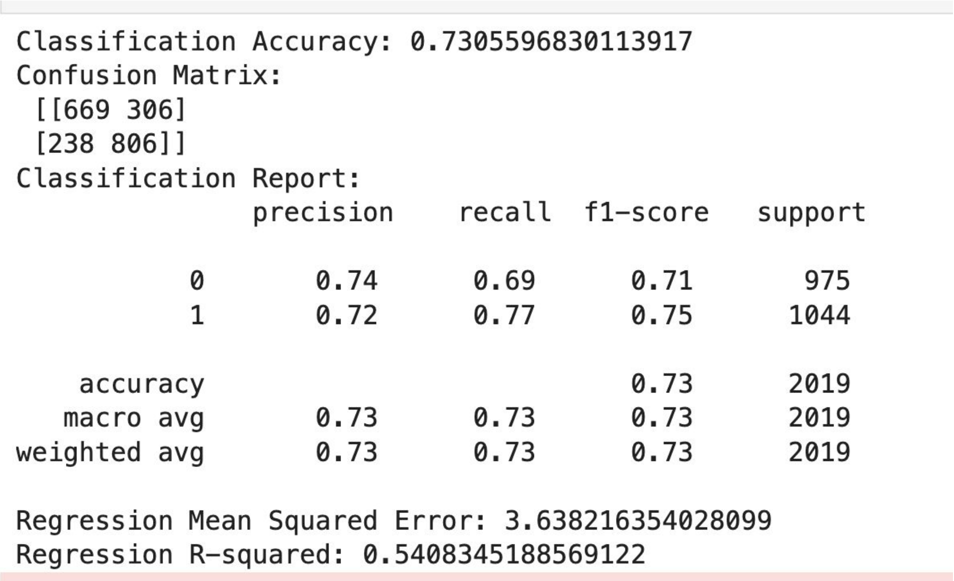


```
Mean Squared Error: 3.919863619862398
R-squared: 0.6858955998135142
```

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
```

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

- I realized an aggregation of my score to par in an altered dataframe would be more optimal for my end goal of being able to predict the leaderboard (and at minimum the winner)
- By creating this altered dataset, in addition to enhancing my original model, I think I could create a good predictor and since my original model is already at 69%, there seems to be promise!

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

That’s all folks!
Any questions?