Linear Regression Project Write-Up

By: Colin Gallagher

Abstract:

A linear regression model was created that predicts a golfer's average score per round that is participating in the Masters tournament. Data was scraped from the PGA Tour website to train, validate, and test various candidate models. Using Python and many of its available packages, the extracted data was able to be cleaned, selected, trained, and reiterated to find better and better models. Clients that would benefit from this model are the golfers to help strategize, gamblers to get an edge, sports analysts to impress their audience, and the PGA to potentially adjust courses to rebalance the game.

Design:

The purpose of this project is to predict a golfer's scoring average at the Master's using past data collected. The goal is to find out what leads to better scoring at the Masters. Is driving performance more important than putting? What about chipping? The model will answer these questions with certain accuracy. Once the model is constructed, the coefficients will reflect the weights of what brings a player's score down. Also by using the equation and statistics of golfers in the present, a score at the Masters can be estimated assuming the golfer performs on average with what their current stats are.

Data:

The raw data that was used for this project was scraped from the PGA Tour website. Scoring averages and many other statistical features were used from the Masters years 2001 to 2022. After scraping and combining all the data into one final main data frame, a total of 1157 rows with 37 features was used to start building models. The target feature being average score per round. The other features were carefully selected from the PGA Tour website and were estimated to have an impact or correlation to a golfer's score based on intuition. Each row represents a player from a certain year with the respective performance stats from that year at the Masters.

Algorithms:

The first objective was to get a baseline model. This was done initially by first throwing all the features from the main data frame into a statsmodel OLS fit model. What was found was three specific features ended up dominating. They were the average scores on par 3, 4, and 5 holes. The coefficients just reflected the number of par 3, 4, and 5 holes on the course, which makes sense but is not what was wanted. Using this info features that either directly reflected a score on a hole, or possibly inferred a score were dropped. The process continued as followed: select initial features that were correlated with the target, fit an OLS model, check for multicollinearity using VIF numbers, remove features that caused multicollinearity issues. Once features showed minimum multicollinearity, the assumptions of linear regression were checked. If all assumptions were met, the model becomes a candidate model. A new set of initial features were selected to build more candidate models. Ridge, Lasso and Elastic Net regression models were also created, but showed no large improvement while also being harder to interpret. Models were compared against each other and the best model was selected. The best model was one that was reasonably accurate (MAE) and easy to interpret. The final selected model:
Scoring Average (pred) = 70.9345 +9.7285*(**Putting Average**) +0.2783*(**3-Putts per Round**) -0.1352*(**Greens in Regulation Percentage**) -0.0713*(**Scrambling**) -0.0137*(**Driving Distance**)
Start with a base score and depending on how a golfer performs by measuring the statistics, they either lose or gain a certain amount of strokes.

Tools:

Tools used for building the linear regression model:

- **Python**: Coding language used to perform actions on the data with its various libraries.
- **Pandas**: Python library used to clean and place data into organized data frames.
- **Matplotlib**: Python library used to visualize the data that was used to evaluate the models.
- **BeautifulSoup**: Python library used to scrape and parse through HTML to get the data.
- **Seaborn**: Python library used to create pair plots that help with feature selection.
- **Statsmodel**: Python library used to fit and evaluate models
- **Scikit-Learn**: Python library used to fit and evaluate models

Communication:

A five minute slide presentation will be given to introduce the target that the model predicts, who it benefits, how it was done, the results, conclusions, any future work that can be done to improve the project, and answer any questions.