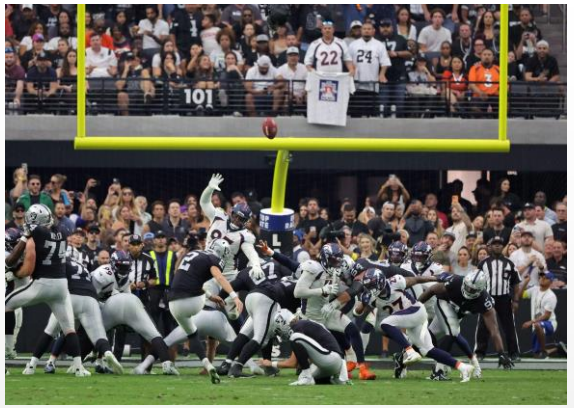


COLLEGE FOOTBALL BETTING:

USING TEAM PERFORMANCE DATA TO PREDICT GAME WINNERS

ALEX BALLI

BACKGROUND

- I graduated from The Ohio State University in 2018 – my time there led me to become interested in college football
- When I moved to Nashville in 2021, it was my first time living in a legal sports betting state
- With my recent training in data science, the next step seemed obvious – could I find useful predictors to build a model to predict college football game winners and potentially even profit from sports betting?

FINDING A DATA SOURCE

- To build a model, I needed data sources that had:
  1. A record of college football winners
  2. Information about the teams that could be used as predictors
  3. Historical betting lines to help measure how well my model performed
- Luckily, I found a website, collegefootballdata.com, that has accurate information about:
  1. College football winners for all conferences dating back to 2004
     a) To simplify this project to fit time constraints, my models focus only Big 10 Conference games (for tOSU)
  2. Abundant team data including, team performance metrics by game, player metrics by game, recruiting rankings, coaching history, and more
     a) To simplify this project to fit time constraints, my models focus only on team performance metrics by game (e.g. yards per game, sacks per game, kicking points per game, etc.)
  3. Historical betting lines dating back to 2013
     a) To simplify this project to fit time constraints, my model performance was only measured against moneyline bets, where a better simply chooses a winner, and the odds determine the magnitude of earnings

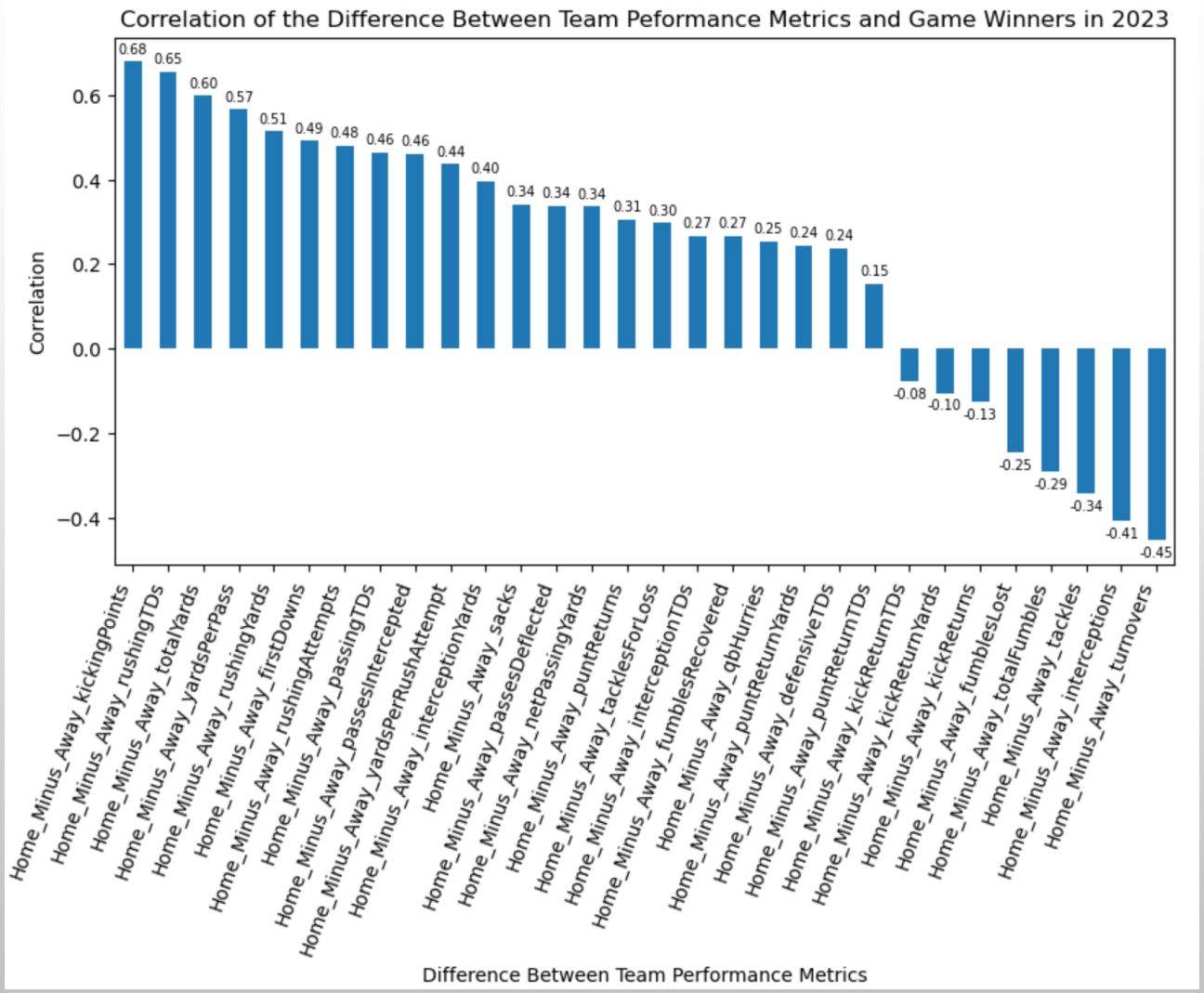
DATA PREPARATION

- Started simple:
  - Used only 2023 data
  - Used only Big 10 games
  - For many different team performance metrics, I calculated:

    Home team metric – Away Team Metric

    and created a correlation matrix with whether or not the home team won

DATA PREPARATION



- Useful for determining important metrics, but there was one big problem:
  - Team performance metrics for a game can’t be used to predict the outcome of the same game!

DATA PREPARATION

- Still using 2023 data only, I created rolling averages by team across each season so that team performance metrics in earlier games could be used to predict outcomes in later games that season.
- Shifted all rolling averages down one game by team so that each game was associated with the season average statistics prior to that game, not including that game.
- Recreated my (*Home team metric – Away Team Metric)* calculations for the season averages
- Tested my ability to use these season average metrics as predictors of whether or not the home team won random 2023 games using logistic regression and random forest models:

Results for the logistic regression model:
The accuracy score is: 0.7333333333333333

	precision	recall	f1-score	support
False	0.80	0.57	0.67	7
True	0.70	0.88	0.78	8
accuracy			0.73	15
macro avg	0.75	0.72	0.72	15
weighted avg	0.75	0.73	0.73	15

Results for the random forest model:
The accuracy score is: 0.6

	precision	recall	f1-score	support
False	0.60	0.43	0.50	7
True	0.60	0.75	0.67	8
accuracy			0.60	15
macro avg	0.60	0.59	0.58	15
weighted avg	0.60	0.60	0.59	15

- I was getting results better than a coin flip!

MODEL SCALE-UP

- It was time to scale up! I brought in the game results and team performance metrics for Big 10 games dating back to 2004 and created the same rolling averages for each performance metric for each team by season.
- I used the 2004-2022 games to train any models while leaving the 2023 games to test the model.
- At this time, 2 models have been used: logistic regression and random forest (max_depth=5):

Results for the logistic regression model:
The accuracy score is: 0.576271186440678

	precision	recall	f1-score	support
False	0.57	0.55	0.56	29
True	0.58	0.60	0.59	30
accuracy			0.58	59
macro avg	0.58	0.58	0.58	59
weighted avg	0.58	0.58	0.58	59

Results for the random forest model:
The accuracy score is: 0.6779661016949152

	precision	recall	f1-score	support
False	0.68	0.66	0.67	29
True	0.68	0.70	0.69	30
accuracy			0.68	59
macro avg	0.68	0.68	0.68	59
weighted avg	0.68	0.68	0.68	59

	variable	importance
0	Home_Minus_Away_kickingPoints_season_avg	0.104457
1	Home_Minus_Away_firstDowns_season_avg	0.089310
2	Home_Minus_Away_totalYards_season_avg	0.066428
3	Home_Minus_Away_yardsPerPass_season_avg	0.065224
4	Home_Minus_Away_sacks_season_avg	0.052905

EVALUATING THE MODEL

- Before improving the models, it was time for the last major step to completing this project: bringing in the moneyline odds!
- Draftkings moneyline odds for 2023 games, as recorded by collegefootballdata.com, were used to evaluate the model.
- Each odds value was converted to a probability:

- Only bet on games where the probability of winning was determined by a given model to be higher than the implied probability by an “extra_confidence_needed” parameter that I could change
- I also set parameters for:
  - a minimum probability to bet (to have a good chance of winning)
  - A maximum probability to bet (to only bet on games that win a significant enough amount of money)

EVALUATING THE MODEL

- Once it was determined what games were worth betting on, one final parameter, the "bet_amount" was used to determine how much would be bet on each game.

Formula to Calculate Bet With American Odds -- Favorites			
Formula	Bet Amount X ((100/(Odds X -1))		
Example	$25 X ((100/(-110 X-1))	To Win:	$22.73

Formula to Calculate Bet With American Odds -- Underdogs			
Formula	Bet Amount X (Odds/100)		
Example	$25 X (150/100)	To Win:	$37.50

- For each game that was predicted correctly, the winnings were added to the profit. Otherwise, the bet amount was subtracted from the profit.
- For example, the results when using the same models from the initial scale-up with:
  - All rolling season average predictors
  - A $100 bet amount
  - An "extra_confidence_needed" = 0.05
  - A minimum predicted probability for betting of 0.1
  - A maximum predicted probability for betting of 0.9

```
Logistic Regression Model Profit with Default Settings:
Total Money Bet: $4100.00
Total Revenue: $5148.04
Total Profit: $1048.04
```

```
Random Forest Model Profit with Default Settings:
Total Money Bet: $4700.00
Total Revenue: $4588.15
Total Profit: $-111.85
```

CONCLUSIONS AND NEXT STEPS

- I was able to create models that successfully use season average team performance statistics to predict college football game winners consistently greater than 50% of the time.
  - Performance statistics like number of first downs, points scored by kicking, total yards, and yards per pass were very useful for predicting winners
- I created a way to use sportsbook moneyline odds to determine if a game is worth betting on, and I created a way to evaluate winnings from betting based on a given model.
- Going forward, there are a number of ways to improve the models to maximize winnings from sportsbetting from here, including:
  - Adding new predictors, possibly outside of team performance statistics (e.g. average player age, number of years head coach has been with the team, etc.) that might be valuable to the model
  - Eliminating unnecessary predictors to avoid overfitting models
  - Expanding to more conferences in addition to the Big 10
  - Adjusting hyperparameters for the models (e.g. max_depth for random forest)
  - Trying new model types (e.g. a gradient boosted model or a neural network)