Prediction of NFL Scores and Spread Optimization

Ashley Lauren. Mersman

Northwest Missouri State University, Maryville MO 64468, USA
S2571660@nwmissouri.edu

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: sports · betting · machine learning · NFL

1 Introduction

In 2023, approximately 28% of the American public was expected to bet on the NFL. That's 73 million people for an increase of almost 60% from the previous year. Sports betting is legal in 34 states and counting and is a multi-billion dollar industry. [2]

Being able to predict game outcomes and optimize the spread is highly profitable for the industry. The spreads must be close enough to incentivize bettors but also large enough to maximize profits and cover any losses.

Game information (date, season, week, teams, and scores), betting information (favorite to win, spread, over/under), stadium information (stadium, location, elevation, field type, and stadium type) and weather information (temperature, wind speed, and humidity).

1.1 Goals of this Research

The goal of this project is to create a new predictive model for game outcomes and score differentials.

1.2 Process

Data

1. Data Collection
   (a) A curated dataset from Kaggle.com was used that relied on sources for weather, NFL team information, betting information, and game information from the past 50+ years. [4]
2. Data Cleaning

   (a) The data will be cleaned to filter out unnecessary columns leaving the following, game information (date, season, week, teams, and scores), betting information (favorite to win, spread, over/under), stadium information (stadium, location, elevation, field type, and stadium type) and weather information (temperature, wind speed, and humidity).
   (b) Columns will have their data types checked and corrected.
   (c) Check for and correct any duplicates.
3. Data Exploration
   (a) Check for correlations and distributions.
4. Visualization
   (a) Create an interactive map of the common stadiums located in the US with their attributes.

Machine Learning

1. Pre-processing
2. Model Selection and Parameter Tuning
3. Model Analysis

Conclusions

2 Data

2.1 Data Collection

A curated dataset on Kaggle.com by Spreadspoke, a sports data analysis company, was used as the basis of this project. Spreadspoke used a variety of sources, including but not limited to: ESPN, NFL.com, NOAA, NFLweather.com, Pro-Football.com and multiple others referenced on the original dataset page. The dataset if refreshed on a weekly basis. The data is available as multiple csv's and the R file used to curate the dataset. [4]

The dataset time frame is 1966 for NFL game results and 1979 for betting odds data.

The kaggle package was used with the kaggle api to download the dataset. [1]

nfl_stadiums.csv contains data about the 120 stadiums each game has been played. This data goes beyond simply NFL stadiums due to internationally played games and other special games. Variables:

1. stadium_name
2. stadium_location: city, state of the stadium
3. stadium_open: the year the stadium opened
4. stadium_close: the year the stadium closed
5. stadium_type: weather type (indoor, outdoor, retractable)
6. stadium_address
7. stadium_weather_station_zipcode: the zipcode used for weather data collection

8. stadium_weather_type: weather category based on average temperature
9. stadium_capacity: stadium seating maximum
10. stadium_surface: field type (Grass, Turf, FieldTurf, Hellas Matrix Turf, Grass, Turf (1969-1970), Grass, Turf (1970-1971), Grass, Turf (1971-1974))
11. stadium_weather_station: weather station ID for NOAA data
12. stadium_weather_station_name
13. stadium_latitude
14. stadium_longitude
15. stadium_azimuthangle: angle of the stadium from North
16. stadium_elevation

nfl_teams.csv is a datafile containing information about the specific NFL teams. There are 44 entries due to teams moving cities, changing names, and other modifications. Variables:

1. team_name
2. team_name_short
3. team_id
4. team_id_pfr: team id on Profootball-reference.com
5. team_conference
6. team_division
7. team_conference_pre2022
8. team_division_pre2022

The game data in spreadspoke_scores.csv contains game data since the 1966 season totaling 13,788 games. It has game information, weather information, and betting information Variables:

1. schedule_date: date game played
2. schedule_season: game season
3. schedule_week: week of season
4. schedule_playoff: boolean value, is this a playoff game
5. team_home: home team
6. score_home: home team score
7. score_away: away team score
8. team_away: away team
9. team_favorite_id: favorite team to win
10. spread_favorite: spread
11. over_under_line: over under
12. stadium: stadium name
13. stadium_neutral: boolean, is the stadium a neutral site
14. weather_temperature
15. weather_wind_mph
16. weather_humidity
17. weather_detail: precipitation detail

The season time information(schedule_date, schedule_season, schedule_week, and schedule_playoff), scoring and team information (team_home, team_away, score_home, score_away), and conditions (stadium, stadium_neutral, stadium_type, stadium_surface, stadium_elevation, weather_temperature, weather_wind_mph, weather_humidity, weather_detail) will be used to predict the game outcome. The betting information (team_favorite_id, spread_favorite, and over_under_line) will be used to optimize the spread. Winner will be added by comparing the team_home and score_home with the team_away and score_away columns.

2.2 Data Cleaning

The data cleaning process started by inspecting the datasets using Pandas `pd.info()` The commas were removed from the stadium_capacity variable using `pd.replace()` and the datatype for stadium_capacity, stadium_open, and stadium_close were changed to Int64 using `pd.astype()`.

```
stadiums.stadium_capacity = stadiums.stadium_capacity.replace(',','', regex=True)
stadiums.stadium_capacity = stadiums.stadium_capacity.astype("Int64")
stadiums.stadium_open = stadiums.stadium_open.astype("Int64")
stadiums.stadium_close = stadiums.stadium_close.astype("Int64")
```

A column for "winner" was created in the dataset for scores by iterating through the dataframe and comparing the score_home with the score_away. The team with the larger score was selected as winner and appended to a list of winners. In the case the scores are equal Tie is appended. The list is then converted to a column in dataframe.

```
#establish who won each game
winners = []

#iterate through games compariong scores to determine winners, append the winner to the lis
for i,v in games.score_home.items():
    if games.score_home[i] > games.score_away[i]:
        winners.append(games.team_home[i])

    elif games.score_away[i]>games.score_home[i]:
        winners.append(games.team_away[i])

    else:
        winners.append("Tie")

#convert list to column
games["winner"] = winners
```

The spread is a point differential that quantifies the margin by which a team is expected to win or lose. A spread of +7 for a home team means that the the home team is expected to lose by 7 points. The favored team must win by more

than 7 points in order for the bettor to win a bet for the favored team. [3] The spread $_favoritewassetupconverselytowhatisexpectedbybettinglinesandhadmissingvalues.Therforeanewspre$

The datasets for the score information and stadium information were then merged into one dataframe using `pd.merge()`.

```
merged_df = games.merge(stadiums, left_on="stadium",
                        right_on="stadium_name", suffixes=['_x', '_y'], how="left")
```

The dataframe columns were selected and reordered. Null values were then examined. All rows with missing scores for either score_home or score_away were dropped. The null values under weather_detail were replaced with "No Precip" since the column is used to indicate additional weather information, it is assumed then that the lack of data means lack of precipitation. Lastly, due to the large number of missing values for stadium_surface this column was dropped. The mean value for weather$_humidity, weather_wind_mph, andstadium_elevationwereusedinplaceofthenullvalues.$

```
merged_df = merged_df[['schedule_date', 'schedule_season', 'schedule_week', 'schedule_playoi
                       'team_home', 'team_away', 'score_home', 'score_away', 'winner', 'sco
                       ,'stadium', 'stadium_neutral', 'stadium_type', 'stadium_elevation',
                      'weather_temperature', 'weather_wind_mph', 'weather_detail', 'weather_

merged_df = merged_df.dropna(subset=['score_home'])
merged_df = merged_df.dropna(subset=['stadium_type'])

mean_humidity = merged_df['weather_humidity'].mean()
merged_df["weather_humidity"].fillna(mean_humidity, inplace = True)

mean_temp = merged_df['weather_temperature'].mean()
merged_df["weather_temperature"].fillna(mean_temp, inplace = True)

mean_wind = merged_df['weather_wind_mph'].mean()
merged_df["weather_wind_mph"].fillna(mean_wind, inplace = True)



mean_elevation = merged_df['stadium_elevation'].mean()
merged_df["stadium_elevation"].fillna(mean_elevation, inplace = True)


merged_df['weather_detail'].fillna('No Precip', inplace = True)
```

The data was checked for duplicates using `pd.duplicated().sum()`. No duplicated values were detected.

The cleaned dataset has 13,558 rows and 20 columns.

Finally, the distribution of the columns was reviewed as a final check for outliers, null_values, or strange data that would require further examination. `pd.describe(include='all'`

Table 1. Data Variables and Definitions

Variable	Description	Data Type
schedule_date	Date of the game	Object
schedule_season	Year of Football Season	Integer
schedule_week	Week of season	Object
schedule_playoff	True for Playoff, False for Regular Season	Boolean
team_home	Home Team	Object
team_away	Away Team	Object
score_home	Home Team Score	Float
score_away	Away Team Score	Float
winner	Winning Team	Object
score_difference	Difference Between Away Team Score and Home Team Score	Float
score_total	Total Points Scored	Float
stadium	Stadium Name	Object
stadium_neutral	False for Bias, True for Neutral Site	Boolean
stadium_type	Outdoor, Indoor, Or Covered Stadium	Object
stadium_elevation	Elevation of Stadium	Float
weather_temperature	Temperature at Game, F	Float
weather_wind_mph	Wind Speed at Game, mph	Float
weather_detail	Precipitation at Game	Object
weather_humidity	Humidity at Game, %	Float

The dependent variable will be the score difference

The raw files for this project can be found at The Python notebook and other files used for this project can be found at https://github.com/AMersman/capstone$_NFL$.

2.3 Data Exploration

2.4 Visualization

3 Model Creation

3.1 Pre-Processing

3.2 Model Selection and Parameter Tuning

3.3 Model Analysis

4 Conclusions

References

1. https://www.kaggle.com/docs/api
2. Contessa Brewer, J.G.: A record 73 million americans plan to bet on the nfl this season, survey says (September 2003), https://www.cnbc.com/2023/09/06/nfl-week-1-record-number-of-americans-to-bet-on-nfl-this-season-.html
3. Preciado, D.: What is a spread in sports betting?,
4. Spreadspoke: Nfl scores and betting data, https://www.kaggle.com/datasets/tobycrabtree/nfl-scores-and-betting-data/data