

AMERICAN FOOTBALL NFL Fantasy Analysis

Nick Rodio

Erin Engle

Katie Beale

Jared Willson

Megan Palma

Jeesu Kwon



Project Overview

Target Audience

Anyone who plays Fantasy Football can utilize our app to help guide them in their decision making process through data analysis.

Motivations

To create a model that accurately predicts the performance of chosen NFL players based on past performance, weather conditions, stadium, years playing, and average yearly fantasy league points scored.

Tools Used:

Linear Regression, Flask, Javascript, Pandas, CSS

30

40

50

40

30

Data Overview

Primary Data Source:

[NFL scores and betting data](#)

[NFL Stats 2012-2023](#)



73 different columns that reference player stats

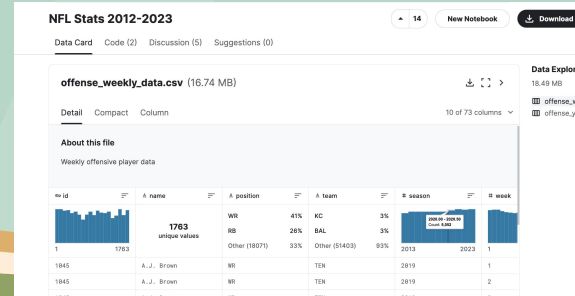
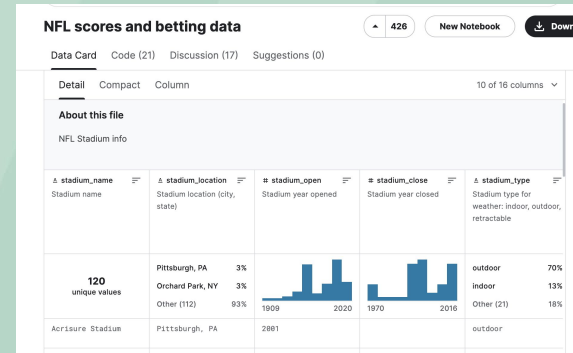


5453 rows



Stadium, weather, & field type data

Site Home Pages



Processing Pipeline

Goal



Ensure data for prior seasons are available, accurate, and consistent across players and seasons including changing team names, stadium names, games per season, etc.



Create a merged dataset from multiple CSV's and spreadsheets

The Kaggle logo, consisting of the word "kaggle" in a blue, lowercase, sans-serif font, is displayed within a white rectangular box.

Process



Downloaded required data from kaggle and updated fields for consistency



Merged stadium and player data



Output new CSV's for model building

30

40

50

40

30

Linear Regression Model



Used many NFL offensive stats to help predict fantasy scores (None that near perfect correlation to fantasy scores)



Back tested model with 2023 model predictions and actual 2023 fantasy results



Predicted 2024 fantasy football scores for upcoming season

```
[10]: lr_model = LinearRegression()
```

```
[11]: lr_model.fit(X_train, y_train)
```

```
[11]: ▾ LinearRegression  
LinearRegression()
```

```
[25]: no_2023_lr_model = LinearRegression()
```

```
[26]: # Training model without 2023 season data  
no_2023_lr_model.fit(X_no_2023, y_no_2023)
```

```
[26]: ▾ LinearRegression  
LinearRegression()
```

	position	games	targets	rec_ypg	ypr	receiving_fumbles	target_share	air_yards_share	carries	rush_ypg	...	overall	stadium_name	stadium_weather_type	stadium_surface
4	WR	17	158	85.65	13.74	2.0	0.30	0.42	0	0.00	...	51.0	Lincoln Financial Field	cold	Grass

30

40

50

40

30

Model Optimization

```
opt_model_all_seasons_data['target_per_game'] = round((opt_model_all_seasons_data['targets']/opt_model_all_seasons_data['games']),2)
opt_model_all_seasons_data['carries_per_game'] = round((opt_model_all_seasons_data['carries']/opt_model_all_seasons_data['games']),2)
opt_model_all_seasons_data['team_off_snaps_per_game'] = round((opt_model_all_seasons_data['teams_offense_snaps']/opt_model_all_seasons_data['games']),2)
opt_model_all_seasons_data['off_snaps_per_game'] = round((opt_model_all_seasons_data['offense_snaps']/opt_model_all_seasons_data['games']),2)
opt_model_all_seasons_data['attempts_per_game'] = round((opt_model_all_seasons_data['attempts']/opt_model_all_seasons_data['games']),2)
opt_model_all_seasons_data['attempts_per_game'] = round((opt_model_all_seasons_data['attempts']/opt_model_all_seasons_data['games']),2)
```



Added new columns that helped increase performance of model



RMSE increased by 18.59%



R2 increased by 1.37%

PRE-Optimization

```
[32]: import math
      mse_2023 = mean_squared_error(fantasy_2023, fantasy_2023)
      rmse_2023 = math.sqrt(mse_2023)
      rmse_2023

[32]: 17.481317677683556

[33]: r2_2023 = r2_score(fantasy_2023, fantasy_2023)
      r2_2023

[33]: 0.9607669583952572
```

RMSE: 17.48

R2: 96.08

POST-Optimization

```
[37]: # Calculating RMSE for model based on new features
      mse_2023 = mean_squared_error(fantasy_2023, fantasy_2023)
      rmse_2023 = math.sqrt(mse_2023)
      rmse_2023

[37]: 14.229830865136156

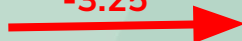
[38]: # Calculating R2 for model based on new features
      r2_2023 = r2_score(fantasy_2023, fantasy_2023)
      r2_2023

[38]: 0.974004203717691
```

RMSE: 14.23

R2: 97.40

-3.25



+1.32



30

40

50

40

30

Model Results



RMSE of 71.00 for 2022 vs 2023 actual fantasy scores proved that our model was not just assuming a players last year performance but including their whole career performance



RMSE => Predictions for 2024 should be expected to be +/-14.23 points different on average



R2 => 97.40% of the variation in predicted fantasy football scores can be explained by the included features or statistics

Official 2024 Season Predictions (Top 10 Players)

	name	team	position	fantasy_2024_score_prediction	fantasy_2024_per_week_score_prediction
0	Josh Allen	BUF	QB	318.60	18.74
1	Puka Nacua	LA	WR	318.36	18.73
2	Justin Herbert	LAC	QB	308.04	18.12
3	Justin Jefferson	MIN	WR	297.64	17.51
4	Lamar Jackson	BAL	QB	293.24	17.25
5	Patrick Mahomes	KC	QB	290.89	17.11
6	Trevor Lawrence	JAX	QB	284.14	16.71
7	CeeDee Lamb	DAL	WR	277.58	16.33
8	Alvin Kamara	NO	RB	274.51	16.15
9	Tyreek Hill	MIA	WR	273.10	16.06

2022 VS 2023 Actual

```
mse_assume = mean_squared_error(fa  
rmse_assume = math.sqrt(mse_assume  
rmse_assume
```

71.00374769451727

Final Model Evaluation

```
[37]: # Calculating RMSE for model bas  
mse_2023 = mean_squared_error(fa  
rmse_2023 = math.sqrt(mse_2023)  
rmse_2023
```

[37]: 14.229830865136156

30

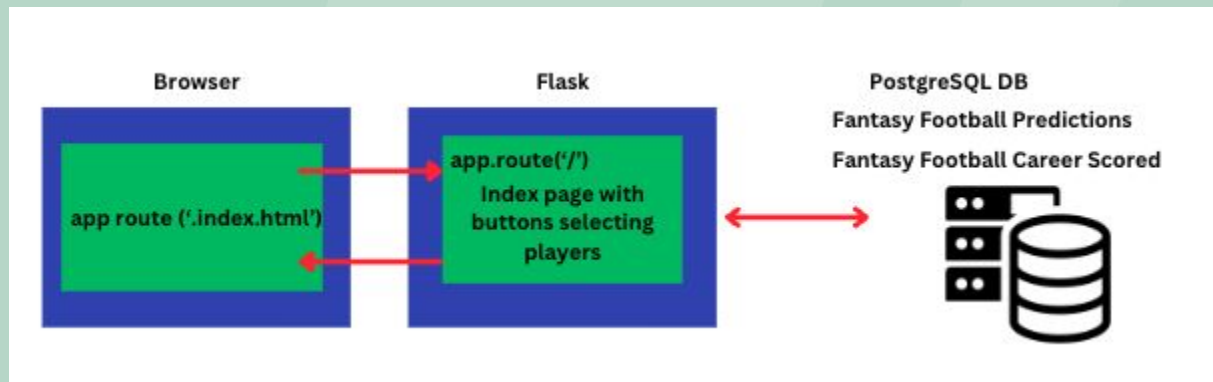
40

50

40

30

Application Architecture



30

40

50

40

30

Demo

30

40

50

40

30

Challenges & Next Steps

Challenges



Weekly data did not match yearly data.



Could not locate and incorporate datasets including weather or date information.

Next Steps



Integrate with the ESPN API to create a live application that accounts for player injuries.



Develop a model to predict week-over-week outcomes.



Add in respective data for a WRs/TEs/RBs - QB stats and vice versa.

30

40

50

40

30

THANK YOU!

