Jonah Lessuk, Aneesh Harwalkar, David Doan, Dillan Hong

FANTASY FOOTBALL DRAFTABILITY MODEL

TABLE OF CONTENTS

01 Introduction

HOME 00

00:00

VISITOR 00

DOWN 00

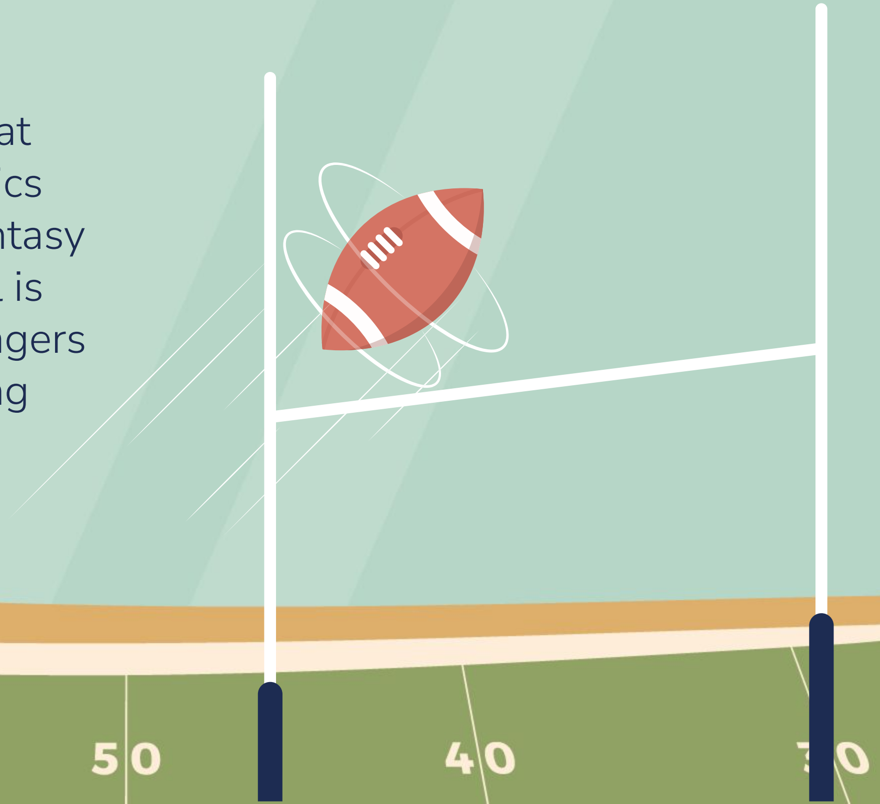
TO GO 00

BALL ON 00

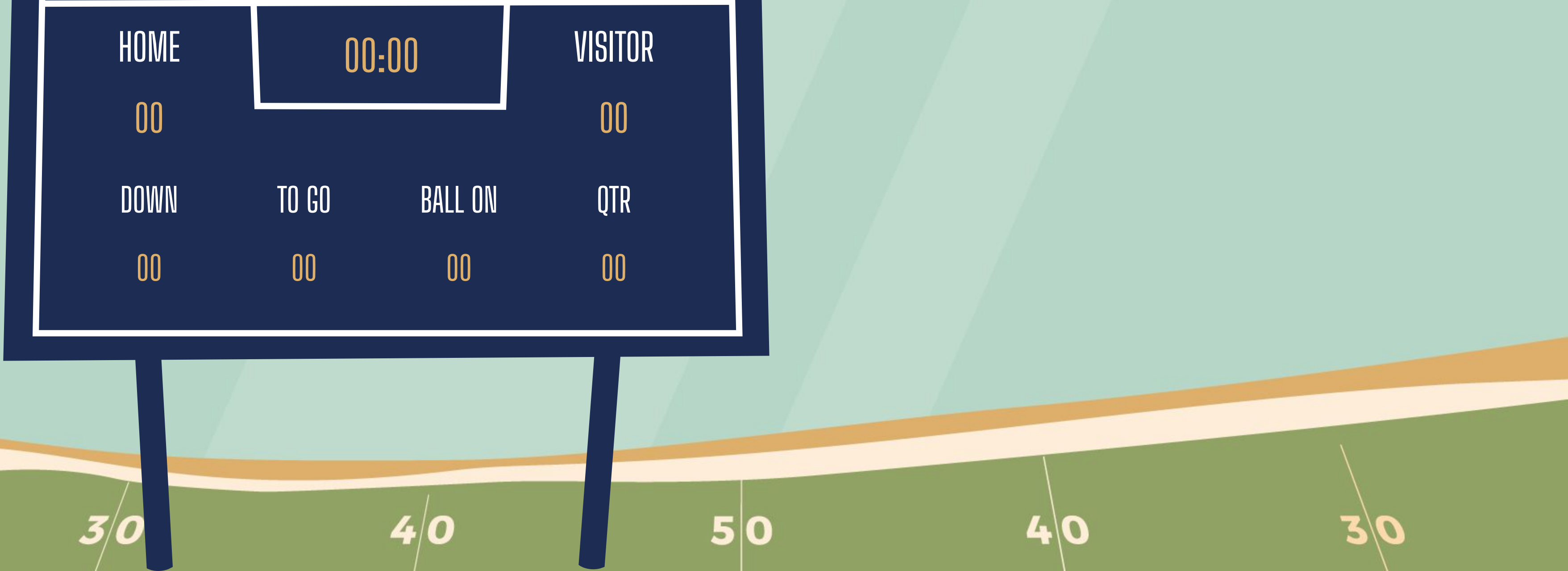
QTR 00

Objective

Create a machine learning model that leverages NFL football player statistics from 2017-2023 to predict players' fantasy rankings given their stats. Our model is designed to help fantasy football managers make smart decisions when selecting players for their rosters.



02 Dataset


HOME
00:00
VISITOR
00
00
DOWN
TO GO
BALL ON
QTR
00
00
00
00
30
40
50
40
30

Our Dataset and Data Processing

We found our dataset on Kaggle:

- Football statistics by player from 2017-2023
- 27 columns (features), 3,388 rows

```
Printing the skewness of all the numerical features:
Age         1.250696
G          -0.690616
GS          0.696971
Cmp         3.994980
Att         3.915833
Yds         4.033851
TD          4.617195
Int         4.255389
RushAtt     2.971415
RushYds     3.108981
YA          1.862675
RushTD      3.755773
Tgt         1.526440
Rec         1.587321
RecYds      1.905831
YR          0.728506
RecTD       2.273740
Fmb         3.377022
FL          3.270358
PPR         1.323142
dtype: float64
```

Pre-processing:

- Drop columns containing player names, ids, and team
- Establish column PosRk (Position Rank) as our label
- Non-numerical column "FantPos"
  - Transform to list and onehotencoder for four unique positions (WR, RB, TE, QB)
- Scaling:
  - PowerTransformer/StandardScaler used to scale skewed numerical columns

03 Methodology

HOME

00:00

VISITOR

00

00

DOWN

TO GO

BALL ON

QTR

00

00

00

00

30

40

50

40

30

Step-by-step:

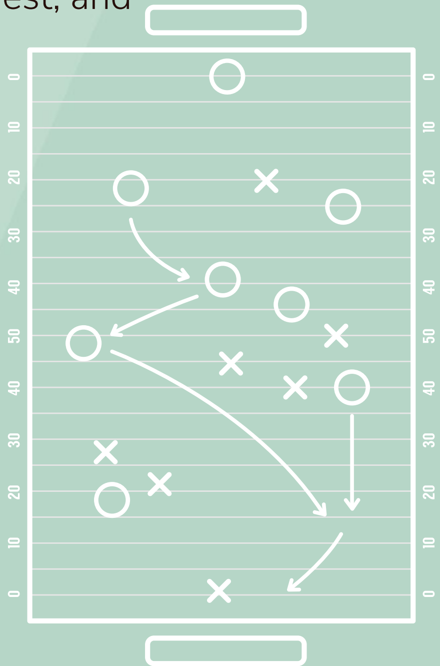
1) Feature selection and pre-processing
2) Feature correlation analysis
3) Train/test split
4) Train multiple models
5) Evaluate Models
    a) Save the best model
6) Test the final model

Our models: Linear Regression, Support Vector Regression, Decision Trees, Random Forest, and Gradient Boosting

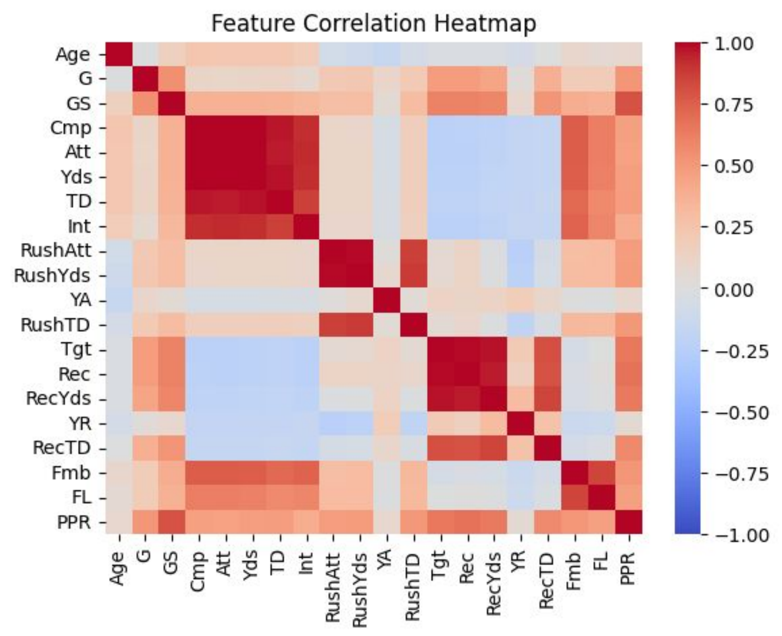
Evaluate using MSE, MAE, and R-squared values

Save the best model using Pickle

One test with randomized inputs is included in the original code. Since our model is saved and can be exported, we can test it more thoroughly in our demo program.



We scaled our data this way because...



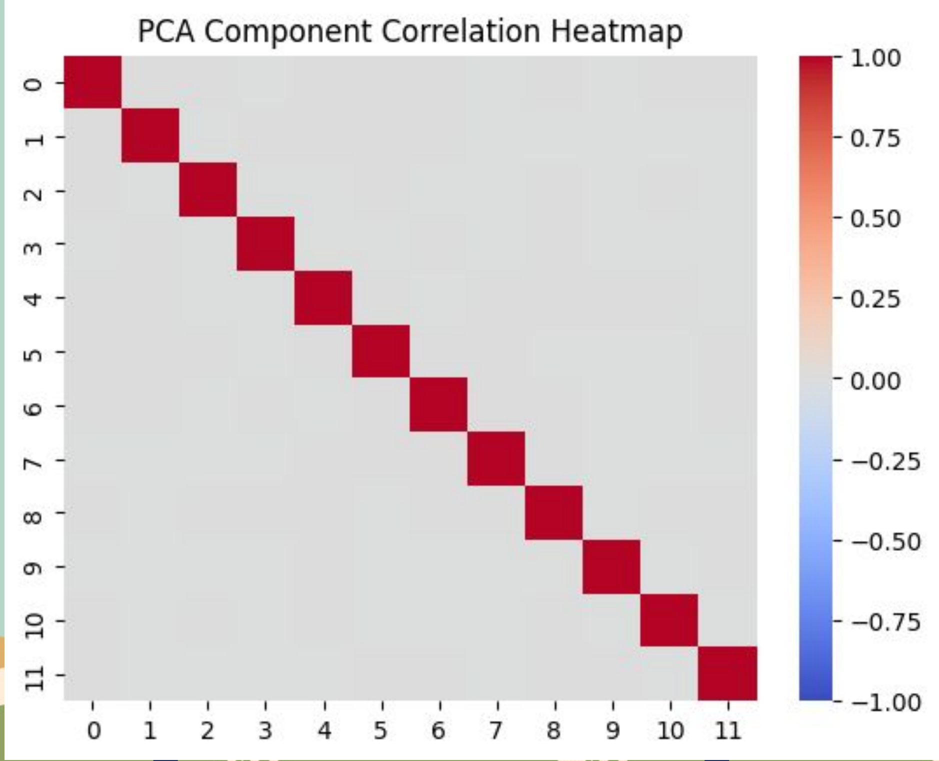

```
Printing all of the strongly correlated feature groups:
['GS', 'PPR']
['Cmp', 'Att', 'Yds', 'TD', 'Int']
['RushAtt', 'RushYds', 'RushTD']
['Tgt', 'Rec', 'RecYds', 'RecTD']
['Fmb', 'FL']
```

Many of these features are positively correlated, creating problems like:

- Redundant information
- Complicates model interpretability
- Can lead to overfitting

How can we fix this?

Principal Component Analysis



What does PCA do?

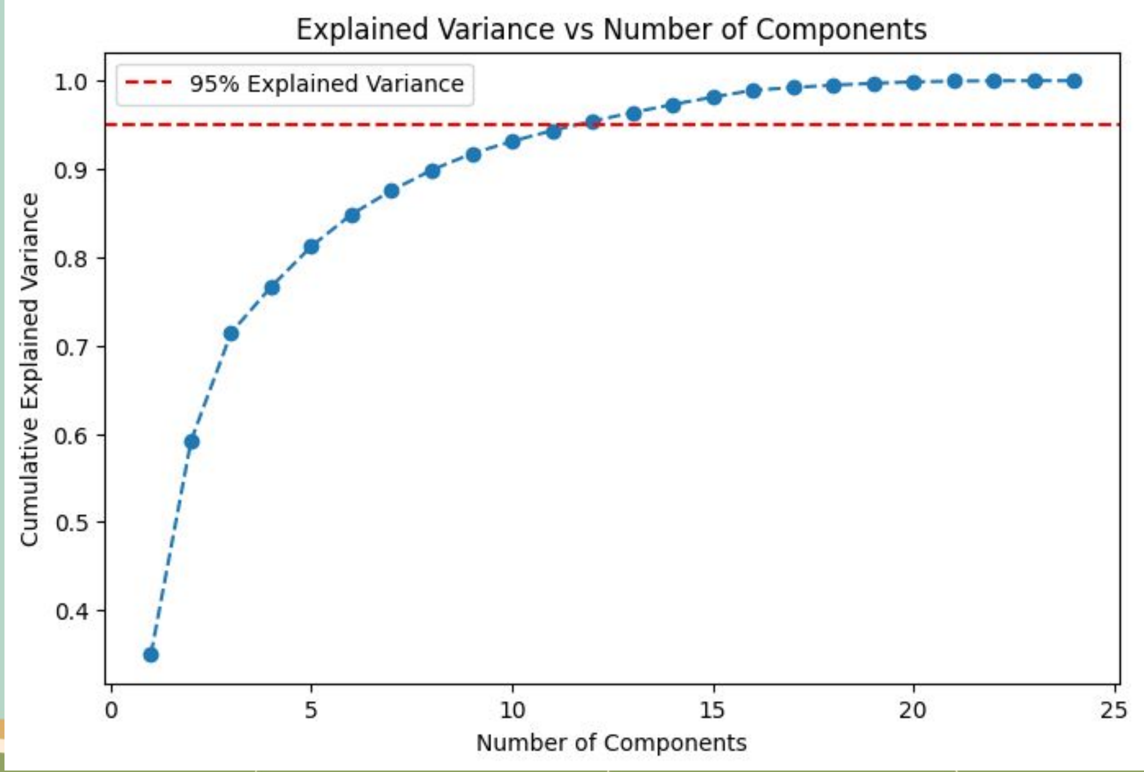
- Reduces dimensionality
- Removes multicollinearity

BUT…

PCA only works on normalized data:

- applied Power Transform to skewed data
- Standard Scaling to normalize data

Principal Component Analysis



Training all the models

Many of our models require the data to be PCA transformed, while others can handle multicollinearity and lose defining information when PCA transformed.

Cannot handle multicollinearity:

Linear Regression
SVR with Linear Kernel
SVR with Polynomial Kernel
SVR with RBF Kernel

Can handle multicollinearity:

Linear Ridge Regression
Linear Lasso Regression
Decision Tree
Random Forest
Gradient Boosting

04 Results

HOME 00

00:00

VISITOR 00

DOWN	TO GO	BALL ON	QTR
00	00	00	00

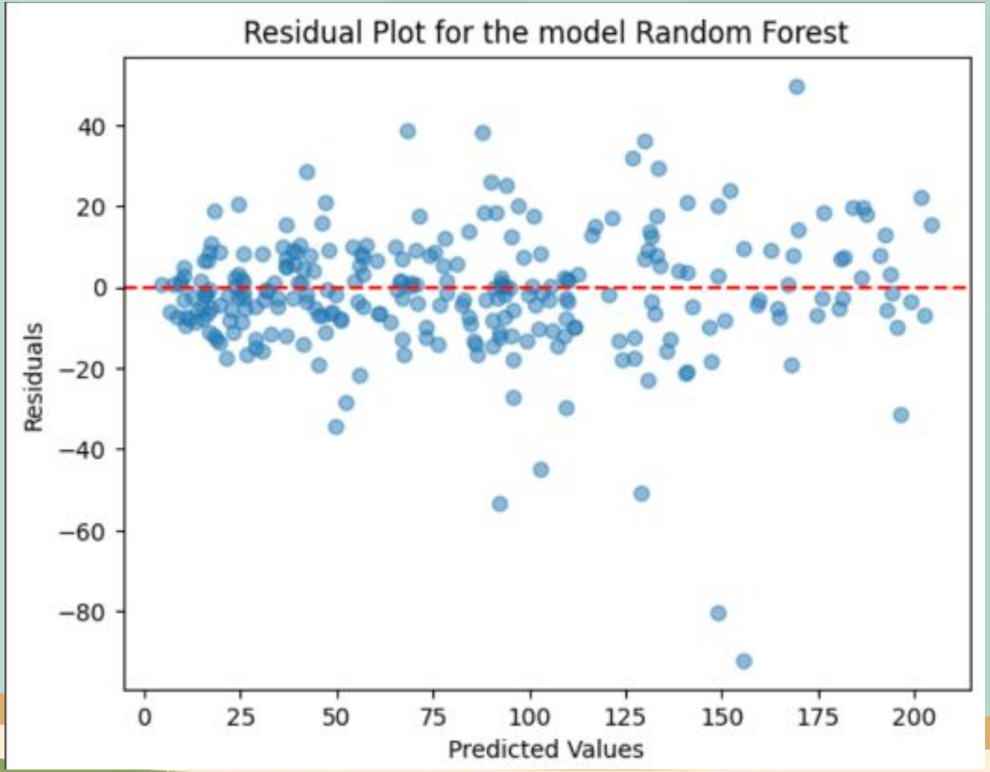
Model Evaluation Metrics (Training and Validation)

- **Random Forest** model achieves lowest MSE (20.91).
- High R^2 confirms strong predictive accuracy.
- Outperformed Decision Tree and Gradient Boosting models.

```
Evaluating   Linear Regression   ...
Average MSE on training / validation data = 447.1080153089607 / 453.76433658888766
Average MAE on training / validation data = 15.997263477422027 / 16.08469825550562
Average R2 on training / validation data = 0.8261596287655715 / 0.823160557367129
-----------------------------------
Evaluating   Linear Ridge Regression   ...
Average MSE on training / validation data = 447.1080242380987 / 453.7630176749615
Average MAE on training / validation data = 15.99740994876357 / 16.08481993616566
Average R2 on training / validation data = 0.8261487227207439 / 0.8231497251017691
-----------------------------------
Evaluating   Linear Lasso Regression   ...
Average MSE on training / validation data = 447.3359294999931 / 453.90376992892925
Average MAE on training / validation data = 16.017554778341676 / 16.103899865642205
Average R2 on training / validation data = 0.8242163622428837 / 0.8211876950990987
-----------------------------------
Evaluating   SVR with linear kernal   ...
Average MSE on training / validation data = 478.75933272646233 / 485.1752946718592
Average MAE on training / validation data = 15.495587395119761 / 15.617817646281605
Average R2 on training / validation data = 0.8269637227941583 / 0.8246552340304124
-----------------------------------
Evaluating   SVR with polynomial kernal   ...
Average MSE on training / validation data = 1045.3557389572652 / 1070.127862487328
Average MAE on training / validation data = 22.570291566711187 / 23.009611786216844
Average R2 on training / validation data = 0.21226634439387798 / 0.17617165178564015
-----------------------------------
Evaluating   SVR with RBF kernal   ...
Average MSE on training / validation data = 127.83003880099815 / 169.25926675073097
Average MAE on training / validation data = 5.9841686090544615 / 7.396215231375605
Average R2 on training / validation data = 0.9535084112074304 / 0.937428564685078
-----------------------------------
Evaluating   Decision Tree   ...
Average MSE on training / validation data = 46.24865388465027 / 51.89197961350787
Average MAE on training / validation data = 5.408853686576789 / 5.760856205163466
Average R2 on training / validation data = 0.9844434528757512 / 0.982403409215632
-----------------------------------
Evaluating   Random Forest   ...
Average MSE on training / validation data = 2.9195638632636327 / 20.918993745387453
Average MAE on training / validation data = 1.2894917999179991 / 3.4677306273062727
Average R2 on training / validation data = 0.9990294557296885 / 0.993006565792425
-----------------------------------
Evaluating   Gradient Boosting   ...
Average MSE on training / validation data = 109.96074776868667 / 194.65878020481972
Average MAE on training / validation data = 7.650847135821311 / 9.771027744321193
Average R2 on training / validation data = 0.9602050545252443 / 0.927538227815664
-----------------------------------
Best model for the task is   Random Forest   which offers the validation MSE of 20.918993745387453
```

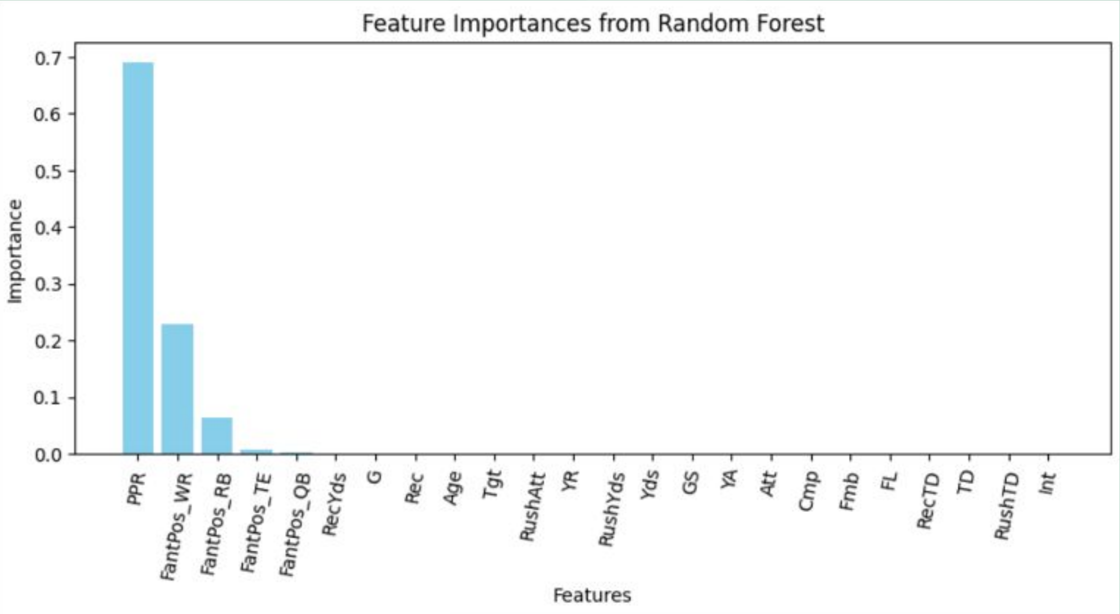
Residual Plot (**Random Forest Model**)

- Visualizes prediction errors (residuals) vs. actual values.
- Random scatter shows minimal bias.
- Confirms the model generalizes well across player ratings.



Feature Importance Bar Chart (**Random Forest Model**)

- Highlights top factors for predicting draftability (e.g., PPR, FantPos_WR).
- Most impactful features likely to align with real-world fantasy strategies.
- Positional data heavily influences draft decisions.



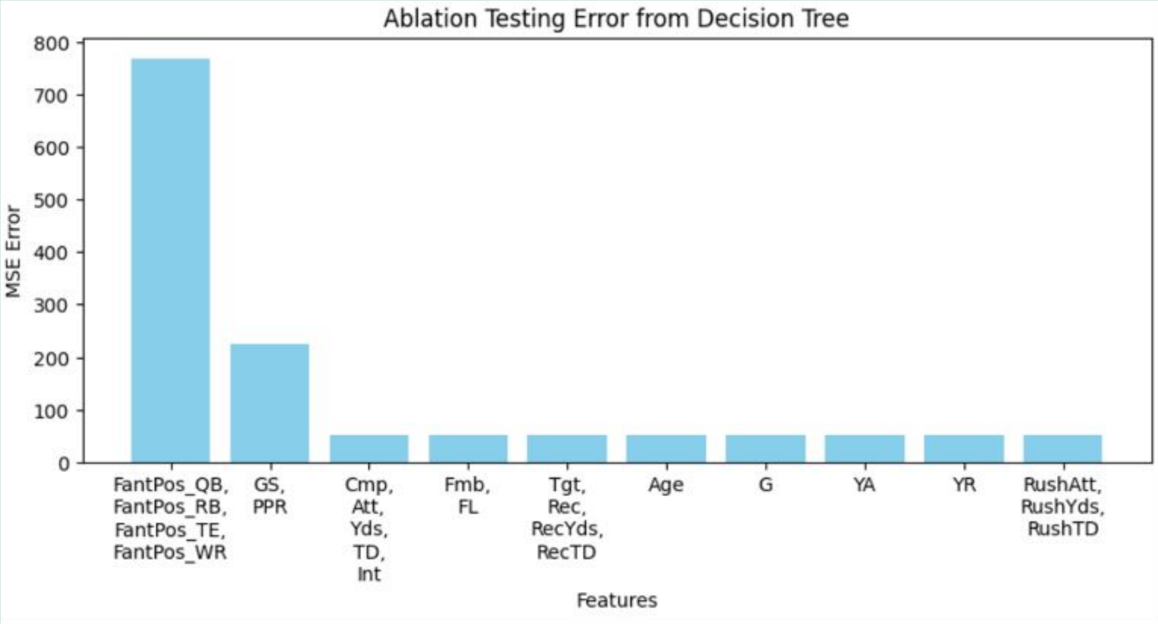
Detailed Feature Importances Table (**Random Forest Model**)

- Numerical ranking of feature importance (e.g., PPR: 0.6, FantPos_WR: 0.2).
- Adds transparency to the model's decision-making process.
- Reinforces the reliability of feature importance results.

	Feature	Importance
19	PPR	0.690810
23	FantPos_WR	0.228029
21	FantPos_RB	0.064604
22	FantPos_TE	0.006430
20	FantPos_QB	0.003588
14	RecYds	0.000893
1	G	0.000640
13	Rec	0.000618
0	Age	0.000600
12	Tgt	0.000570
8	RushAtt	0.000566
15	YR	0.000519
9	RushYds	0.000358
5	Yds	0.000352
2	GS	0.000308
10	YA	0.000306
4	Att	0.000201
3	Cmp	0.000182
17	Fmb	0.000152
18	FL	0.000091
16	RecTD	0.000087
6	TD	0.000042
11	RushTD	0.000028
7	Int	0.000026

Ablation Testing Bar Chart (**Decision Tree Model**)

- Tests impact of removing features on prediction accuracy.
- Removing FantPos_-, PPR, and GS categories causes the largest error increase.



05 Conclusion

HOME
00
00:00
VISITOR
00

DOWN
00
TO GO
00
BALL ON
00
QTR
00

30 40 50 40 30

Conclusions

Findings:

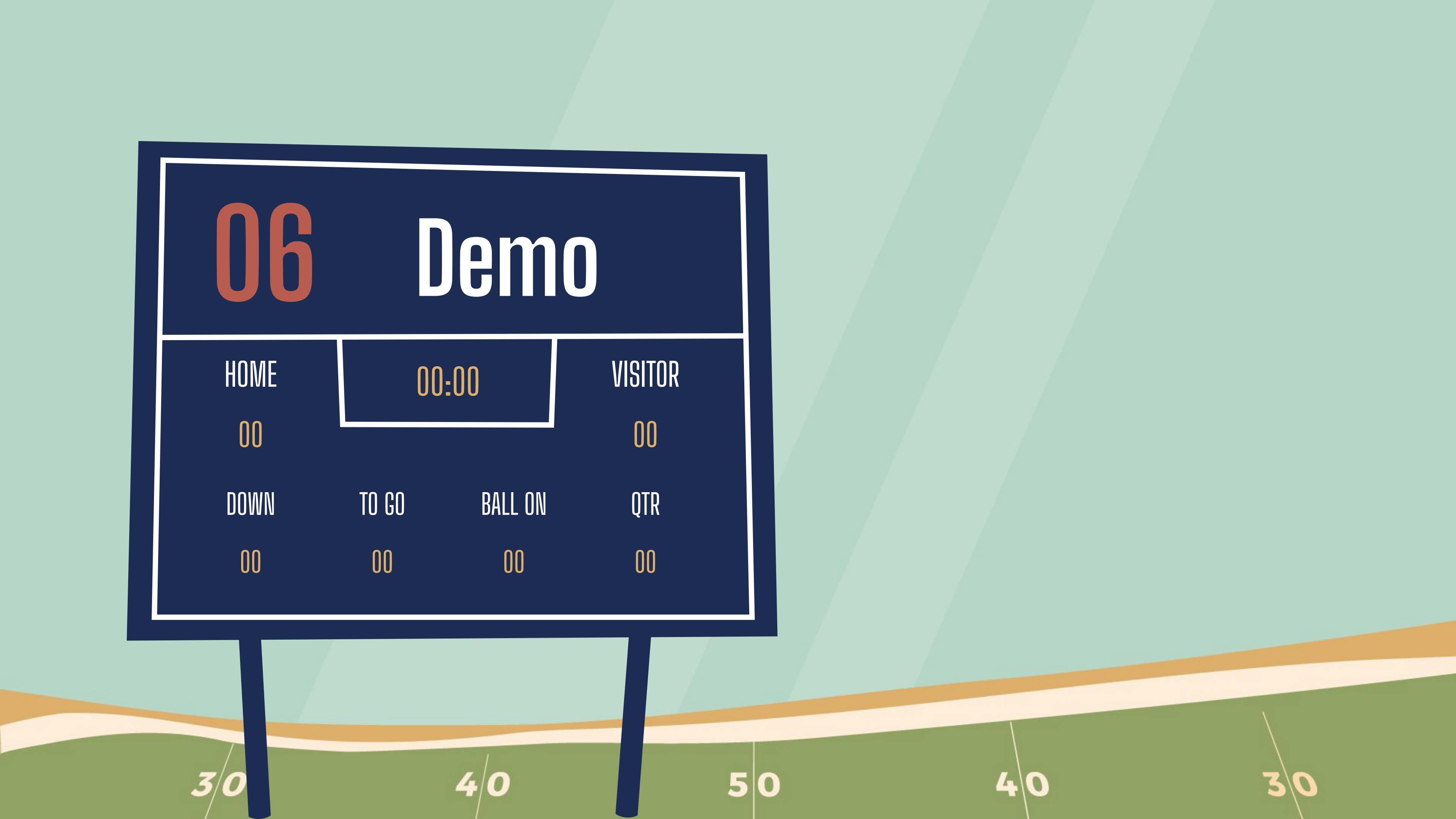
- PPR and positional metrics are key to draftability.
- Random Forest is the most accurate model.

Limitations:

- Temporal changes in player trends not captured.
- Lack of real-time data reduces immediate applicability.

Future Applications:

- Real-time draft tools.
- Extend to other fantasy sports.


06
Demo
HOME
00:00
VISITOR
00
00
DOWN
TO GO
BALL ON
QTR
00
00
00
00
30
40
50
40
30

Xavier Worthy Chiefs

Overview Stats Games Splits Bio

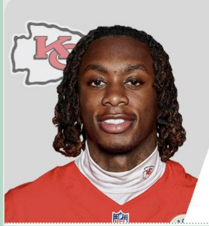
Scrimmage

Regular Season

Season	GP	REC	TRG	REC/G	YDS	AVG	YPG	TD	LNG	ATT	ATT/G	YDS	YPC	YPG	TD	LNG	TOT YDS	TOT TD	FUM
2024	12	33	61	2.8	407	12.3	33.9	4	54	11	0.9	49	4.5	4.1	2	21	456	6	0
Career	12	33	61	2.8	407	12.3	33.9	4	54	11	0.9	49	4.5	4.1	2	21	456	6	0

Xavier Worthy has collected 114.6 PPR fantasy points this season.

https://www.statmuse.com/nfl/player/xavier-worthy-31096/career-stats



#1

Xavier
Worthy

Kansas City Chiefs

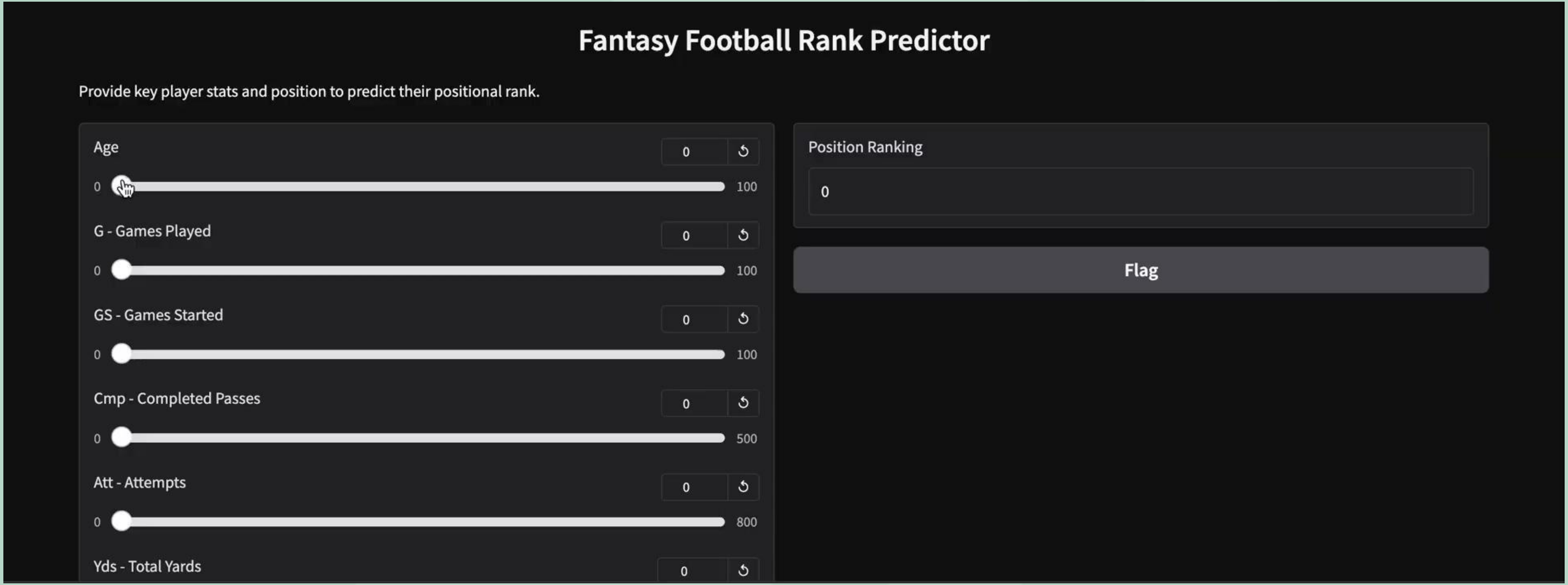
ELIG WR
MANAGER Waivers
STATUS Healthy

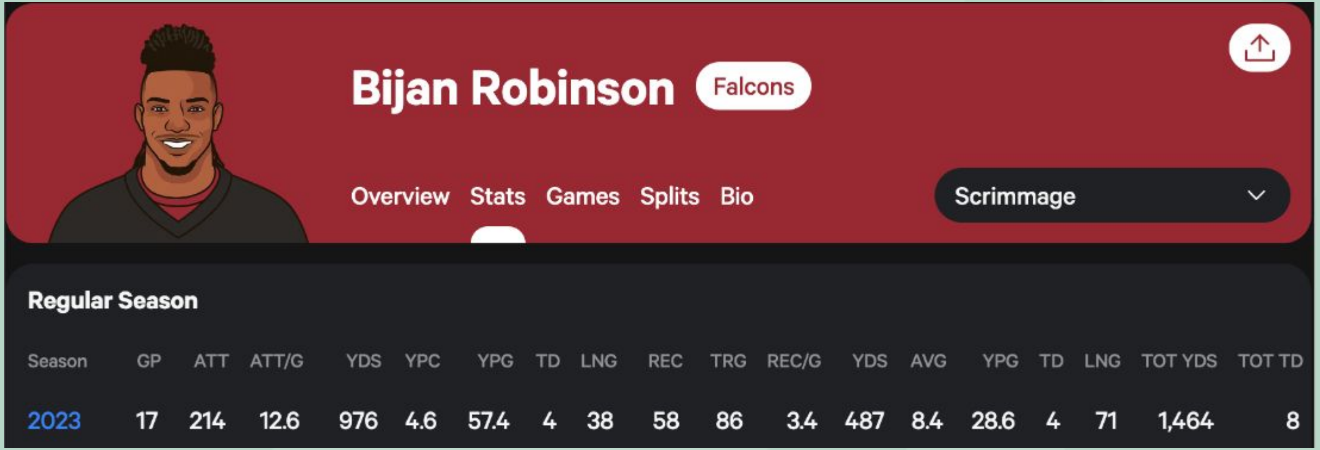
POSITION RANK
40

AVERAGE POINTS
9.5

% ROSTERED
80.8 (-0.4)

https://fantasy.espn.com/football/players/add

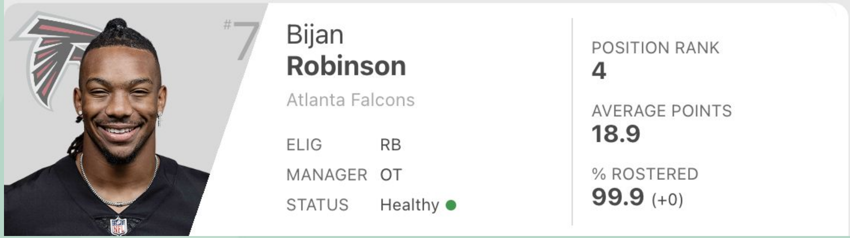

Fantasy Football Rank Predictor
Provide key player stats and position to predict their positional rank.
Age
0
0
100
G - Games Played
0
0
100
GS - Games Started
0
0
100
Cmp - Completed Passes
0
0
500
Att - Attempts
0
0
800
Yds - Total Yards
0
Position Ranking
0
Flag



Regular Season

Season	GP	ATT	ATT/G	YDS	YPC	YPG	TD	LNG	REC	TRG	REC/G	YDS	AVG	YPG	TD	LNG	TOT YDS	TOT TD
2023	17	214	12.6	976	4.6	57.4	4	38	58	86	3.4	487	8.4	28.6	4	71	1,464	8

https://www.statmuse.com/nfl/ask/bijan-2023-ppr



https://fantasy.espn.com/football/players/add

Fantasy Football Rank Predictor

Provide key player stats and position to predict their positional rank.

Age

0

0 100

G - Games Played

0

0 100

GS - Games Started

0

0 100

Cmp - Completed Passes

0

0 500

Att - Attempts

0

0 800

Yds - Total Yards

0

Position Ranking

Flag

30 40 50 40 30

Thanks for listening!

Q & A

What was the rationale behind selecting Position Ranking as the target value for prediction, rather than Overall Ranking?

We chose to select Position Ranking for our label instead of the Overall Ranking because in fantasy football, scoring differs depending on the specific position of the player, whether it's QB, RB, WR, TE. Some positions will get more points than others for the same statistic so we decided that delivering the user a position rank might be more beneficial and accurate.

What factors led to the decision to use regression models for prediction rather than other machine learning algorithms?

We used regression analysis because it provides more detailed and precise predictions among players. Given our skewed data, where many stats are clustered on the lower end, regression allows us to rank players with similar stats more effectively.

Could you provide a detailed description of the mechanics behind Power Transformation in data preprocessing?

See the following slide...

Power Transformation

At its basic description, Power Transformation changes skewed data to fit a normal distribution (Gaussian). And there are two methods:

- **Box-Cox**: Applicable only to datasets with positive values (excluding 0)
- **Yeo-Johnson**: Applicable to datasets with positive and negative values

For our project, we used **Yeo-Johnson** transformation as certain stats aren't applied to all player positions. For instance, quarterbacks don't have stats in receptions.

How does Yeo-Johnson transformation work technically?

$$\psi(\lambda, y) = \begin{cases} ((y+1)^\lambda - 1)/\lambda & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0 \\ -[(-y+1)^{2-\lambda} - 1)]/(2-\lambda) & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y+1) & \text{if } \lambda = 2, y < 0 \end{cases}$$

For all the positive values, the Yeo-Johnson transformation applies a formula similar to Box-Cox, where y is the original data value and λ is a parameter that determines the specific equation to be applied. This λ parameter is chosen to maximize the likelihood of achieving a Gaussian distribution, generally iterating through possible values until the highest likelihood is achieved.

References and Project Link

Link to the GitHub repository:
https://github.com/aneeshharwalkar/FantasyFootballDraftabilityModel

Link to the Kaggle dataset:
"Fantasy Football Data 2017 - 2023" by Gary Bolduc
https://www.kaggle.com/datasets/gbolduc/fantasy-football-data-2017-2023?select=fantasy_merged_7_17.csv

References:

Yeo, In-Kwon, and Richard A. Johnson. "A new family of power transformations to improve normality or symmetry." *Biometrika*, vol. 87, no. 4, 1 Dec. 2000, pp. 954–959, https://doi.org/10.1093/biomet/87.4.954.