

---

# Predicting the Success of NFL Wide Receivers Based on their College Careers Using Classification Models

---

Author: Akira Taniguchi

## **Problem Statement**

The National Football League (NFL) and the National Collegiate Athletic Association (NCAA) are the most popular American football leagues in the USA, a sport that is deeply ingrained in the country's culture and history. We will refer to American football as football for the sake of brevity.

Unlike other professional sports leagues, the NFL does not have a minor league where players can be developed and trained until deemed ready to compete at the highest level. College football fills this role as the second-highest tier of football competition in the US. The overwhelming majority of NFL players compete in the NCAA Division I before being selected in the NFL draft or being signed as undrafted free agents. However, only around 1.6% of NCAA college football players are given the opportunity to play in the NFL [1]. Based on a study of 1996-2016 draft picks, only about 8% of drafted players make a difference in the NFL beyond replacement value, and around 30% see significant play time or make contributions to their teams [2].

In this project, I will be focusing on the wide receiver position in the offense that specializes in catching the ball. I am interested in predicting whether an NFL wide receiver will have a productive season based on data from their college careers. First, I will create a threshold on what constitutes a successful season for a WR in the NFL. Then I will implement and compare seven classification models which are Naïve Bayes, Logistic Regression, Support Vector Machine, Neural Networks, KNN, Decision Tree, and Random Forests to see whether they can accurately make predictions about the success of a wide receiver in the NFL.

## **Data Source**

The data source used in this project is from Pro Football Focus, a sports analytics company that focuses specifically on the NFL and NCAA Division I Football. PFF claims to cover every player on every play of every game at the NFL and major college football level [3]. Their data source was used because of their in-depth analysis and advanced metrics of football that cannot be found elsewhere. PFF began collecting data from NCAA Division I football games starting in 2014, so our dataset starts from that year.

## **Data Preparation**

### **Labelling:**

There are many metrics to evaluate the performance of a wide receiver, but the most common one is how many receiving yards they produce throughout a season. While there is no official measurement of what constitutes a successful season, 1000 yards is often used as a threshold for this. However, it is common for NFL players to take several years in the league to develop into productive players, therefore newer players may be evaluated differently. There are evaluations where rookies producing more than 500 yards [4] and sophomores producing more than 800 yards [5] are considered a success. Thus, I've decided to classify any NFL wide receiver producing 800 yards in any season as a success. Receptions (how many catches they've caught) and touchdowns were not used for labeling as those metrics are much more volatile than receiving yards due to team and game environments and circumstances that may be out of the player's control.

### **Data Cleaning and Feature Selection:**

The datasets originally considered were the NCAA wide receivers' data from the year 2014 to 2022 and NFL wide receivers' data from 2015 – 2023. There are a total of 4804 wide receivers in the NCAA dataset and a total of 630 wide receivers in the NFL dataset. After merging the two, there are a total of 377 wide receivers recorded in both datasets.

Our initial dataset for the project consists of 377 rows and 46 columns. While the NFL dataset was used to label whether the wide receiver had a successful season for classification, the NCAA dataset will be used as the input data for the classification modeling. Within the NCAA dataset, only data from each player's final college season is considered.

Several columns were then removed that were deemed irrelevant to our analysis, such as the year, player name, team name, and various metrics that are not used to evaluate wide receivers such as blocking statistics. This left us with a total of 24 columns that would be considered. Every column consists of numerical data that are comprised of either floats or integers.

Three columns include NaN values which are 'grades\_hands\_drop', 'yards\_per\_reception', and 'yards\_after\_catch\_per\_reception'. The mean value of each column was inputted for all missing values within these columns.

The following table consists of all the columns that are being considered:

Table 1: Features of the dataset

Column/Feature Name	Explanation	Data Type
player_game_count	Number of games in which the player appeared	Integer
avg_depth_of_target	measure of how far downfield a player is typically targeted	Float
avoided_tackles	Missed Tackles Forced after a Reception	Integer
caught_percent	Percentage of targets caught	Float
drops	on-target passes dropped by the receiver	Integer
grades_hands_drop [6]	PFF Grade for HandsDrop (out of 100)	Float
grades_hands_fumble [6]	PFF Grade for HandsFumble (out of 100)	Float
grades_offense [6]	PFF Grade for offense (out of 100)	Float
grades_pass_route [6 ]	PFF Grade for pass routes(out of 100)	Float
route_rate	Measures the percentage of passing plays for which a player ran a route in any given game	Float
routes	How many plays where the player runs to get open for a forward pass	Integer
slot_rate	measures the percentage of passing plays for which a player lined up in the slot	Float
slot_snaps	How many snaps the player was lined up in the slot	Integer
targeted_qb_rating	measure of the performance of the passer	
targets	Number of times a player has had a pass thrown his way	Integer
touchdowns	Touchdowns scored	Integer
wide_rate	measures the percentage of passing plays for which a player lined up out wide	Float
wide_snaps	How many snaps the player was lined up out wide	Integer
receptions	How many successful catches were made	Integer
yards	How many receiving yards	Integer
yards_after_catch (YAC)	Distance gained by a receiver after catching a pass	Integer
yards_after_catch_per_reception	YAC per reception	Float
yards_per_reception (YPR)	Receiving yards gained per reception	Float
yards_per_route_run (YPRR)	Average number of yards that a player makes every time they run a route	Float

## Data Standardization:

The dataset was standardized as it's a common requirement for many machine learning estimators as they may perform poorly if the individual features do not follow a Normal Distribution [7]. The StandardScaler function from Python's sklearn library was used to perform standardization to transform the features to have a mean of zero and a standard deviation of one.

## Dimensionality Reduction:

As seen in Table 1, many features are bound to be correlated with each other. Features such as routes, targets, receptions, and yards directly affect other features such as grades\_pass\_route, yards\_after\_catch, yards\_per\_reception, yards\_after\_catch\_per\_reception, and yards\_per\_route\_run to name a few. Figure 1 shows the correlation matrix of the dataset where many of the features are moderately and highly correlated.

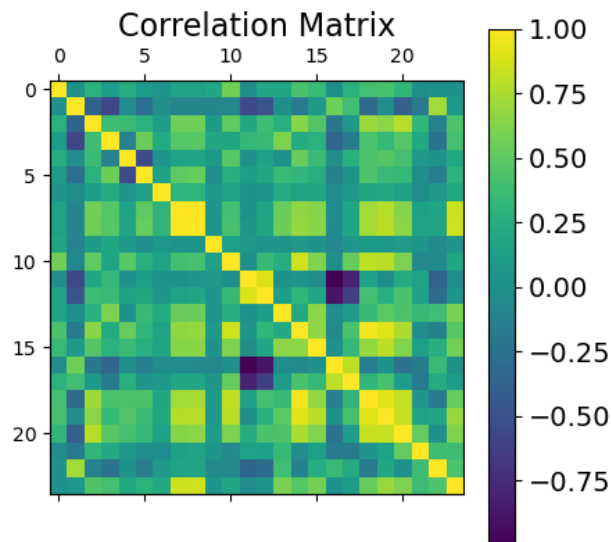


Figure 1: Correlation Matrix of the Variables

Since there are 24 features with many of them correlated, Principal component analysis (PCA) was performed on the dataset. Figure 2 shows how much of the total variance in the original dataset can be explained by each principal component. The first 10 principal components were then chosen as they explain 95.14% of the total variance from the original dataset.

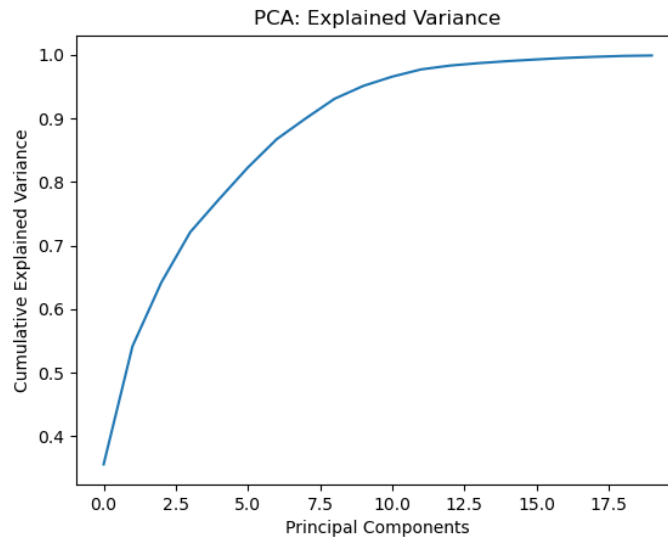


Figure 2: Cumulative Explained Variance Ratio of PCA

### **Train Test Validation Split & Class Imbalance:**

The dataset was split into a training dataset and test dataset at an 80-20 ratio respectively. Within the training dataset, there are 248 players that were labelled as unsuccessful in the NFL and 53 players that were labelled as successful. The classification training data is skewed heavily towards the unsuccessful class. To address this, oversampling was performed on the training dataset using SVMSMOTE, a variant of the SMOTE algorithm which uses an SVM algorithm to detect samples to use for generating new synthetic samples for model training [8]. After oversampling, our training dataset has 496 data points where the classes were balanced with 248 data points each.

### **Evaluation and Final Results**

The seven classification models that were considered were Naïve Bayes, Logistic Regression, SVM, Neural Network, KNN, Decision Tree, and Random Forest. We trained each model with the training sets after tuning the hyperparameters and predicted the classifiers using the test data sets. The accuracy, precision, recall and F1 scores were calculated for each model's predictions to gauge the performance of their classifications.

As seen in Table 2, Random Forest has the highest accuracy out of all the models with 80.26%, followed by Decision Tree at 77.63%, and Neural Network at 76.32%. However, all the models have a relatively low precision score of less than 0.5. This means that the classifiers have a high number of False positives; predicting that a wide receiver will have a successful season in the NFL despite not doing so. Decision Tree had the highest precision score at 0.2941, followed by Random Forest at 0.2727, and Naïve Bayes at 0.25.

Table 2: Performance of Classifiers

	Accuracy	Precision	Recall	F1 Score
Naive Bayes	0.7105	0.25	0.6	0.3529
Logistic Regression	0.6447	0.16	0.4	0.2289
SVM	0.7105	0.2	0.4	0.2667
Neural Network	0.7632	0.1667	0.2	0.1818
KNN	0.6974	0.2174	0.5	0.303
Decision Tree	0.7763	0.2941	0.5	0.3704
Random Forest	0.8026	0.2727	0.3	0.2857

While all the classification models have a higher recall score than their precision scores, they are still fairly low, hovering around 0.5. A low recall score means that the classifiers have a high number of False negatives; predicting that a wide receiver will not have a successful season in the NFL when they indeed did. Naïve Bayes has the highest recall score at 0.6, followed by KNN and Decision Tree at 0.5.

The F1 score is the harmonic mean of the precision and recall scores where a high F1 score represents high precision as well as high recall [9]. While all the classification models had a fairly low F1 score below 0.5, Decision Tree had the highest F1 score at 0.3704, followed by Naïve Bayes at 0.3529 and KNN at 0.303.

After evaluating all the metrics, I conclude that the Decision Tree model is the best classifier for this problem. Although it has the second highest accuracy behind Random Forest, it has the highest precision and F1 score while having the second highest recall score. While the accuracies of the classification models are relatively high with the best hovering around 80%, future work can delve into improving the precision, recall and F1 scores to better predict whether a wide receiver will have a successful season in the NFL.

### Future Work:

- Increase the amount of data used for analysis such as incorporate all seasons played in the NCAA, not just the final season. A player who was injured in their final season and then goes on to have a successful NFL career may affect the model's predictions. Also consider including NFL combine results that describe physical traits and characteristics such as height, wingspan, weight, strength, and speed.
- Consider other dimensionality reduction techniques such as variants of PCA, LDA, t-SNE, and feature selection techniques such as LASSO, Filter Methods, and Wrapper Methods. Look to combine the dimensionality reduction techniques with the feature selection techniques as well as experiment with using them separately.
- Look into different sampling and resampling techniques when performing train test data split and dealing with class imbalance.

## Works Cited

[2] Ludford, W. (2022, April 27). NFL draft pick bust rate remains very high. Daily Norseman. <https://www.dailynorseman.com/2022/4/26/23042105/nfl-draft-pick-bust-rate-remains-very-high>

[3] Mellor, C. (2019, August 7). PFF Signature Statistics – a glossary. PFF. <https://www.pff.com/news/pro-pff-signature-statistics-a-glossary> NFL. (n.d.).

[5] Moody, K. (2024, March 4). Dynasty Crossroads Episode 267 - Sophomore WR Thresholds. In Dynasty League Football. Dynasty Crossroads Podcast. <https://dynastyleaguefootball.com/dynasty-crossroads-podcast/>

[1] NFL Development Pipeline - College Advisory Committee. NFL Football Operations. Retrieved April 26, 2024, from <https://operations.nfl.com/journey-to-the-nfl/nfl-development-pipeline/college-advisory-committee/>

[6] PFF Player Grades | PFF. (n.d.). PFF. <https://www.pff.com/grades>

[7] Preprocessing data. (n.d.). Scikit-learn. <https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing-scaler>

[9] sklearn.metrics.f1\_score. (n.d.). Scikit-learn. [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html)

[8] SVMSMOTE — Version 0.12.2. (n.d.). [https://imbalanced-learn.org/stable/references/generated/imblearn.over\\_sampling.SVMsmote.html#svm-smote](https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SVMsmote.html#svm-smote)

[4] Using Rookie WR Numbers to Model Sophomore WR Success. (2018). Reddit. Retrieved April 26, 2024, from [https://www.reddit.com/r/DynastyFF/comments/7y02x8/using\\_rookie\\_wr\\_numbers\\_to\\_model\\_sophomore\\_wr/](https://www.reddit.com/r/DynastyFF/comments/7y02x8/using_rookie_wr_numbers_to_model_sophomore_wr/)