Joshua Mould, Benjamin Schmid, Noah Swan
Dr. Zhang
STAT8490 - Deep Learning
12 May 2023

Predicting NFL Play Results via Artificial Neural Network

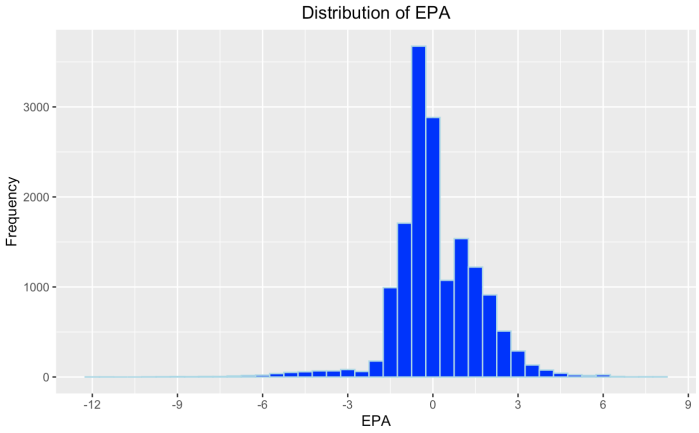
Introduction

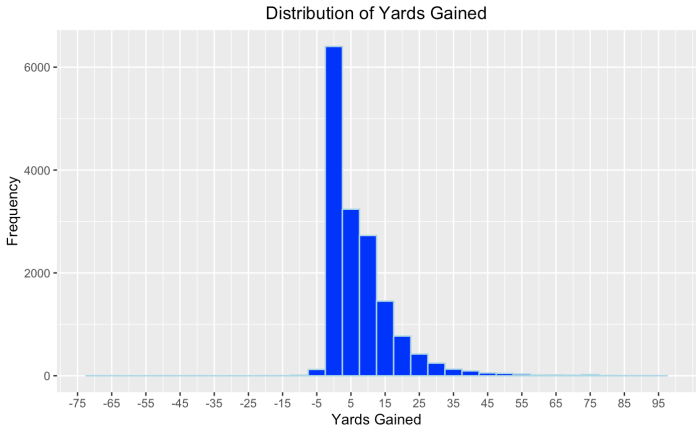
One of the popular fields for predictive modeling, especially due to its difficulty and interest by many fans, is that of sports. Particularly, football is arguably the most popular sport in the nation these days. With advancements in technology, football teams have growing analytics departments hoping to use predictive models to gain a competitive advantage over other teams. Even everyday citizens who participate in fantasy football leagues seek to use data to beat their opponents. Our problem will be one that is looked at by many people who are interested in football: can we use data from the play up until the throw to predict the result of the play?

Explanation of the Data

The dataset that we will be using comes from Kaggle and is part of the NFL Big Data Bowl 2021 analytics competition (plays.csv from https://www.kaggle.com/competitions/nfl-big-data-bowl-2021/data). We will use the dataset that contains the information for all 15,815 passing plays (plays where the ball was thrown to a teammate) that occurred during the 2018 National Football League (NFL) season. There are 32 columns in the dataset, with one being the actual number of yards gained by the play. This is our first response variable, one that can be easily interpreted as our ability to predict the outcome of the play.

Our other response variable will be Expected Points Added (EPA) at the play level. Expected Points Added is calculated based on the change in Expected Points that a play results in. Expected Points are the number of points that a team is expected to score on a given possession based on situational information such as down, yards to go, and field position. Ultimately, the formula involves multiplying the probability of a touchdown given the situation by the number of points that would be scored on a touchdown, which is 7. Then this is added to the probability of a field goal being scored by 3, the number of points scored on a field goal. The Expected Points Added is then the difference between the Expected Points prior to the play and the Expected Points after the play. You can read more about EPA via one of the links in our references section. The distributions of both response variables is below:





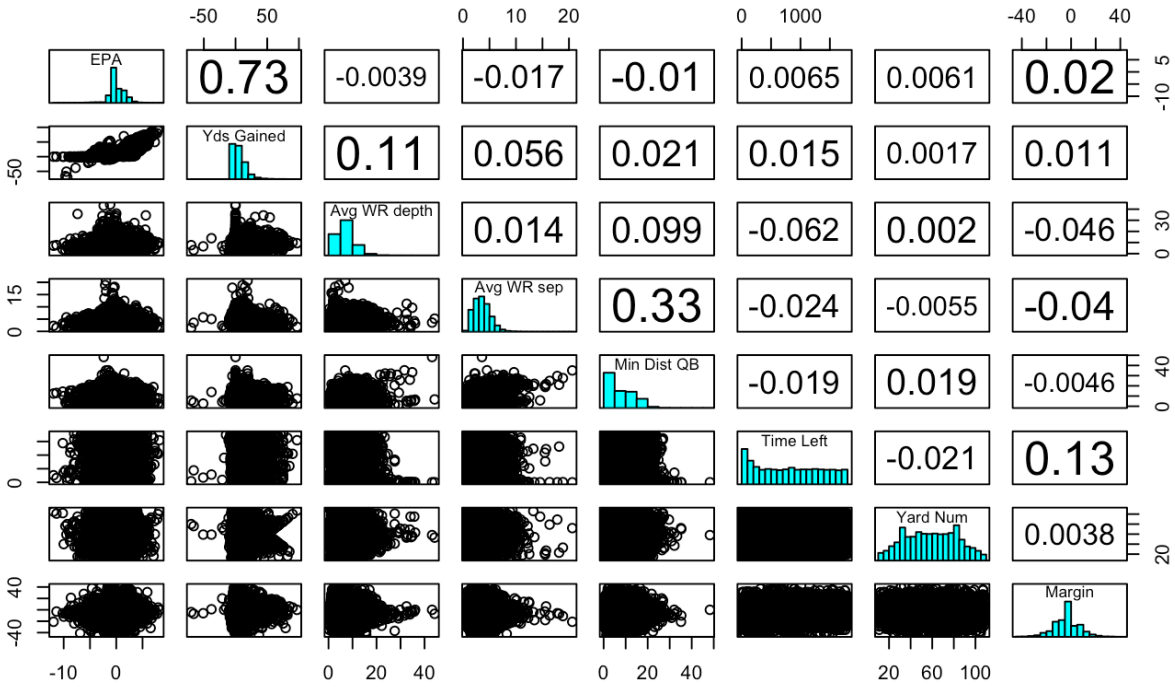
Due to these distributions, we decided to use Mean Absolute Error as our evaluation metric for EPA and Mean Squared Logarithmic Error for Yards Gained. Mean Absolute Error is best used for mostly normal distributions with outliers, which describes EPA well. Mean Squared Logarithmic Error is best used for non-normal distributions that have large values, which describes Yards Gained well. These respective evaluation metrics will be used for all models.

We started with a set of features given in the dataset that captures the play context as described in the table below.

Down	Categorical (1,2,3,4)	Teams have four downs to progress 10 yards, moving to 2nd down if they haven't advanced 10 yards on 1st down, etc.
Yardline	Numeric	Where on the field does the play occur
Time Left in Half	Numeric	Number of seconds remaining in half
Yards to First Down	Numeric	Number of yards until the team reaches the first down mark
Margin	Numeric	Number of points the team in possession of the ball is winning/losing by

Number of Pass Rushers	Numeric	Number of players on the defensive team who are attempting to tackle the quarterback
Offensive Formation	Categorical	Label describing the type of formation for the quarterback and running backs
Dropback Type	Categorical	Label describing if quarterback stays behind offensive line or moves out to either side
Number of Players at each position (x6)	Numeric	Number of running backs, tight ends, wide receivers, defensive linemen, linebackers, and defensive backs on the field

To supplement these variables, we took advantage of the tracking data given in the competition. Denoted by the "week#.csv" files on Kaggle, these files include the locations of all 22 players tracked 10 times per second. We derived three new variables from this data as follows. The variable "mindist_qb" refers to the Euclidean distance of the closest defender to the quarterback when the pass is thrown. The variable "avg_wr_depth" refers to the average horizontal distance between the route runners and the line of scrimmage (where the play starts). We note that route runner refers to any player logged as having run some type of route during the play, which could include wide receivers, tight ends, and running backs. The variable "avg_wr_sep" refers to the average Euclidean distance between the closest defender to each route runner.



The correlation matrix above shows the two response variables and the six predictors with highest rating in variable importance metrics from the random forest model. We see that none of these predictors are highly correlated with either response, and that there are no potential multicolinearity issues.

Our predictors have varying scales. One demonstration of this is the comparison between time left in the half, which goes from 0 to 1800, to yards to first down, which usually is between 0 and 10. In addition, many of our predictors are not expected to have outliers. Due to all of this, we will use standardization for our feature scaling method.

Model Building Process and Hyperparamters

Artificial Neural Network

Our main model for this application project will be using an Artificial Neural Network for each response variable. Our process was to start with the base model that we discussed in class, add complexity, then fine-tune it for the best results. The process and corresponding evaluation metrics are detailed below.

Our model building process for ANN started with the basic model that we worked with in class. We used a downscaling process for hyperparameter tuning. Starting with two hidden layers and a large number of neurons in each, our first step was to scale down the neurons. Next, we added dropout layers, scaling these up. Third, we shifted the number of neurons up and down.

Finally, we added epochs to see if it would further lower loss. The results are shown below with the corresponding evaluation metrics.

For the Yards Gained model, the final model had hyperparameters of 100 epochs, 16 nodes in layer 1, dropout of .2 in layer 1, 8 nodes in layer 2, and dropout of .2 in layer 2. For the EPA model, the final model had hyperparameters of 100 epochs, 16 nodes in layer 1, dropout of .4 in layer 1, 8 nodes in layer 2, and dropout of .4 in layer 2. The error graphs for EPA (left) and

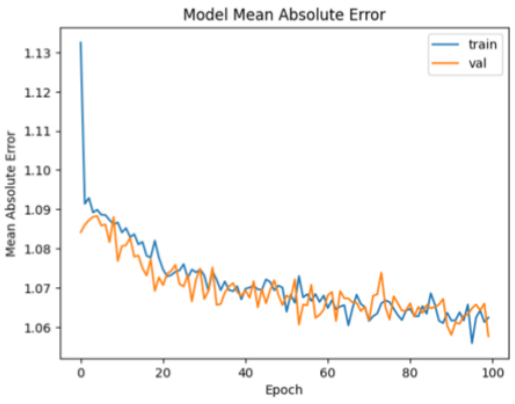
EPA

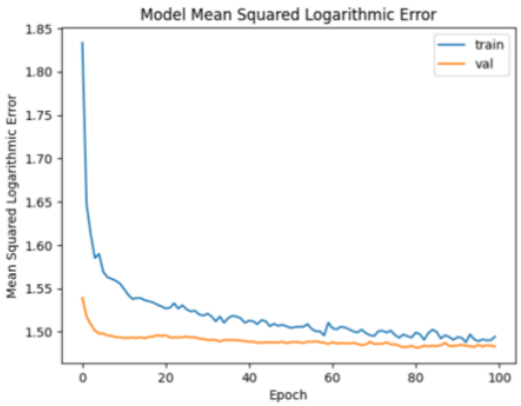
Epochs	Layer1	Dropout?	Layer 2	Dropout?	MAE	MSE	MSLE		
100	32		32		1.162	2.651		Adjust Nodes	
100	32		16		1.139	2.583		Adjust Nodes	
100	16		8		1.083	2.373		Adjust Nodes	
100	16	0.2	8	0.2	1.063	2.263		Add dropout	
100	16	0.3	8	0.3	1.059	2.243		Adjust dropout	
100	16	0.4	8	0.4	1.058	2.232		Adjust dropout	Best model
100	16	0.5	8	0.5	1.06	2.236		Adjust Dropout	
100	16	0.4	12	0.4	1.058	2.245		Adjust nodes	
100	12	0.4	8	0.4	1.063	2.239		Adjust nodes other direction	
150	16	0.4	8	0.4	1.062	2.236		Adjust epochs	

Yards Gained

Epochs	Layer1	Dropout?	Layer 2	Dropout?	MAE	MSE	MSLE		
100	32		32		6.901	98.361	1.953	Adjust Nodes	
100	32		16		6.816	96.735	1.861	Adjust Nodes	
100	16		8		6.742	93.922	1.769	Adjust Nodes	
100	16	0.2	8	0.2	6.664	90.91	1.689	Add dropout	Best model
100	16	0.3	8	0.3	6.695	91.248	1.704	Adjust dropout	
100	16	0.4	8	0.4	6.691	90.616	1.692	Adjust dropout	
100	16	0.5	8	0.5	6.704	90.715	1.7	Adjust Dropout	
100	16	0.2	12	0.2	6.685	90.512	1.698	Adjust nodes	
100	12	0.2	8	0.2			1.696	Adjust nodes other direction	
150	16	0.2	8	0.2			1.7	Adjust epochs	

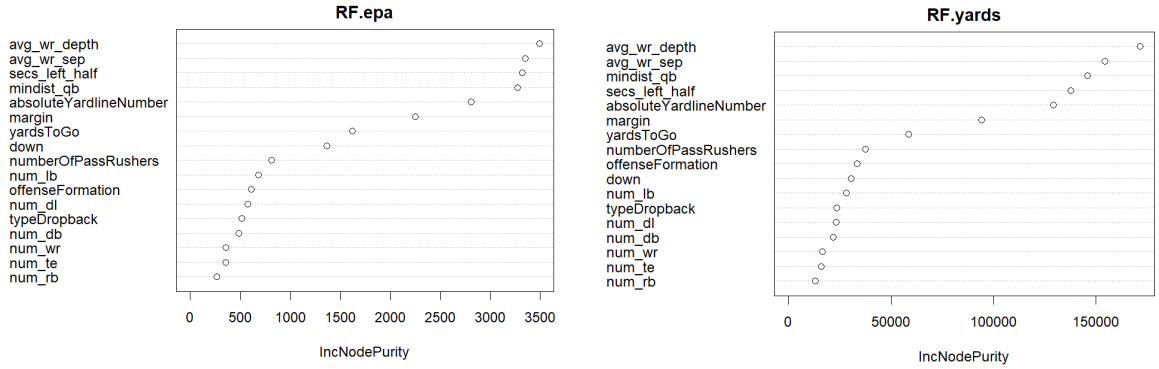
Yards Gained (right) are below.





Random Forest

We built two Random Forest models to compare against the results of the Artificial Neural Network. For these models, we chose 500 for the number of trees and 5 for mtry which we derived from the rule of thumb of taking the number of predictors divided by 3. We tried tuning mtry but found that building successive models took a significant amount of time; as this model is not at the center of our analysis, we believed the rule of thumb to be sufficient. Mtry is a hyperparameter that states how many predictors are included in each tree. This is done to control randomness. We provide the feature importance plots for each of the models here.



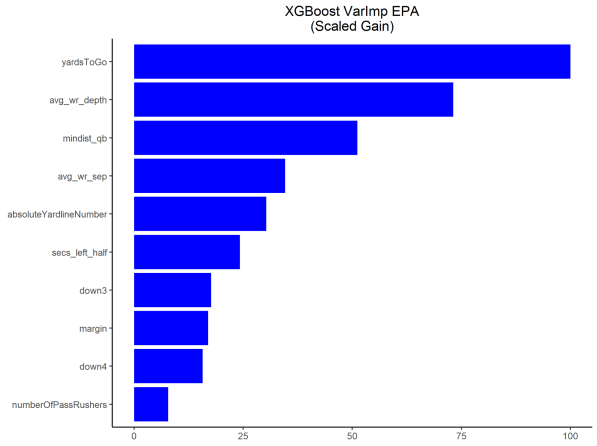
We can see from these plots that in both models, the predictors derived from the tracking data (avg_wr_depth, avg_wr_ep, and mindist_qb) ranked as some of the most important along with secs_left_half and absoluteYardlineNumber. As plays with more yards gained also have higher values for EPA, it is not surprising that there is overlap in variable importance between the two models.

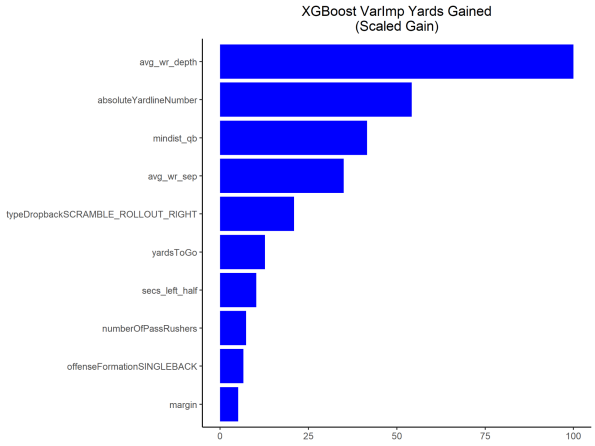
XGBoost

We built two XGBoost Models for comparison as well. Through our research, we found that for many sports predictive modeling projects, people have had great success with a supervised learning model XGBoost, or Extreme Gradient Boosting. Thus, we decided to use it in our project. It is a gradient-boosting decision tree algorithm that combines weak learners sequentially to correct the errors made by the previous models. Models continue to be added until no further improvements can be made. Many people choose to use XGBoost as opposed to other models as it is praised for its high efficiency (it can be 10 times faster than its competitors) and its high accuracy. It is often considered an improvement on random forest.

For these two models, we used a tune grid to consider various combinations of hyperparameters. XGBoost models have more parameters so we believed it more important to tune these than the Random Forest models. For the number of trees or boosting rounds, we considered the values 200, 300, 400, and 500. For the learning rate eta, we considered the values 0.025, 0.05, 0.1, and 0.3. For the maximum depth of the trees, we considered the integers

between two and 10 inclusive. We also used three-fold cross-validation to avoid overfitting in the training process. The best model, as measured by root mean squared error, for predicting yards had 400 rounds, a learning rate of 0.025, and a max depth of 2. The best EPA model had 200 rounds, a learning rate of 0.025, and a max depth of 3. The plots below capture the most important variables for each model.





Similarly to the Random Forest models, the predictors from the tracking data were also important for the XGBoost models. Interestingly, yardsToGo ranked as the most important by scaled Gain for the EPA model while not featuring in the top 10 for the yards gained model.

Conclusion

The table below shows the results for each of the three models for both response variables. For both problems, ANN has the best results, although the errors are close for all three models. In addition, we see that our models perform well in a tough area to predict. We see in that our models agree in variable importance, giving us further confidence as to which predictors are more useful.

Model	Yards Gained	EPA
Evaluation Metric	MSLE	MAE
Python ANN	**1.689**	**1.058**
R Random Forest	1.824	1.067
R XGBoost	1.777	1.062

QB	EPA	EPAOE
Patrick Mahomes	.369	.292
Jimmy Garoppolo	.314	.232
Russell Wilson	.291	.208
Drew Brees	.295	.183
Philip Rivers	.268	.174

WR	EPA	EPAOE
Tyler Lockett	.943	.878
OJ Howard	.930	.793
Mark Andrews	.826	.700
Tyrell Williams	.689	.585
Will Fuller	.713	.572

Team Defense	EPA	EPAOE
Buffalo Bills	-.049	-.163
Chicago Bears	-.045	-.149
Minnesota Vikings	.006	-.097
Jacksonville Jaguars	.005	-.089
Los Angeles Chargers	.013	-.086

Above are the top 5 Quarterbacks, Wide Receivers, and Team defenses in terms of the metric we created using our predictions called Expected Points Added Over Expected as denoted below.

$$EPAOE = EPA - EPA\_prediction$$

An average EPAOE that is greater than 0 for offensive players or less than 0 for defensive players means that the player/team overperformed how they were expected to. This metric makes sense of our results by showing us the best quarterbacks at improvising throws and making accurate throws in difficult situations, constantly over performing expectations. Patrick Mahomes is the perfect example of this kind of player. The other quarterbacks are also known for a combination of accuracy and arm strength, leading to smart throws that make the most of the situations they are put in. The top wide receivers tell a similar story. These are all receivers who are good at catching passes thrown in between defenders and do not drop the ball very often. They are able to capitalize on good situations and over perform in bad situations. The top defenses listed for EPAOE, are the defenses that are best at closing in on a play while the ball is in the air and breaking up potentially good offensive opportunities. The EPAOE is negative for defenses because they want to limit the EPA of the offensive team. These defenses do well at deflecting passes, limiting long passes, and intercepting the ball once it is in the air. Using our football knowledge and intuition, we can make sense of these models in context and understand the value they give us in terms of evaluating players.

Future Work and Limitations

We believe that the main reason for such large error metrics is the difficulty of predicting the outcome of a play by using information from just one frame of the play. Using additional features that capture information from the frames before the moment the pass is thrown could improve results. Another limitation of these models and a line of possible work would be the inclusion of some measure of player talent into the models. We would expect that predictions for yards gained or EPA would be higher with a more talented quarterback and more talented wide

receivers. We also found that the predictors engineered from the tracking data tended to be some of the most important, so future work could include the greater utilization of this data set for the addition of features.

References

Dataset (plays.csv) - https://www.kaggle.com/competitions/nfl-big-data-bowl-2021/data
Explanation of EPA - https://www.nfeloapp.com/analysis/expected-points-added-epa-nfl/
Thesis with similar project - https://repository.yu.edu/bitstream/handle/20.500.12202/5659/Avery%20Ennis%20Thesis%20An%20Introduction%20to%20Modeling%20NFL%20Tracking%20Data.pdf?sequence=1&isAllowed=y
Evaluation Metrics - https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/
https://www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/#h-root-mean-squared-logarithmic-error-rmsle