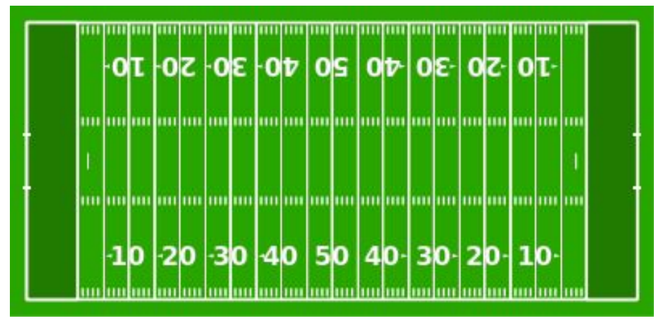
Predicting Yardage gained in a Football Play

Sahil Gupta, Sooraj Kumar, Tiancheng Wu
CS 598PS Final Project (Fall 2019)





Objectives:

- Football is a complex sport. One of the main objectives though is to move with the ball to "gain" yards. Many of these yardage gains come from "run plays".
- Our objective is to predict yards gained on rushing plays given information about the game status at the start of the play, such as player positions, game time, player orientation, etc.
- The dataset consists of player level attributes such as positions, orientation, speed, etc. for each of the 22 players, play level features such as score, quarter, play rusher, etc. and game level features such as location, weather, team information, etc.
- The evaluation metric used by the corresponding Kaggle competition is the Continuous Ranked Probability Score (CRPS).

$$C = \frac{1}{199N} \sum_{m=1}^{N} \sum_{n=-99}^{99} (P(y \le n) - H(n - Y_m))^2$$

So, instead of predicting a yardage value, we predict a CDF function.

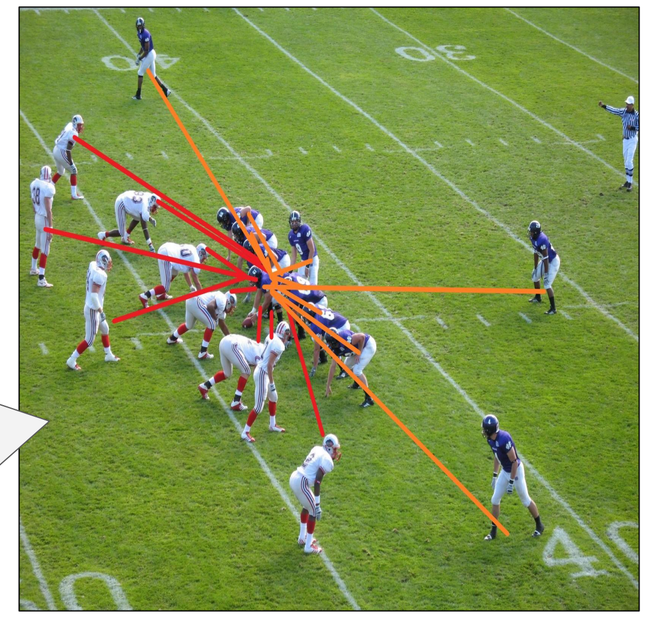
Player position relative to rusher is better than absolute position because: a. Rusher is the central player on the field b. Position can change over time, but it's relatively the same situation. c. Rusher is the main influencer in run plays.

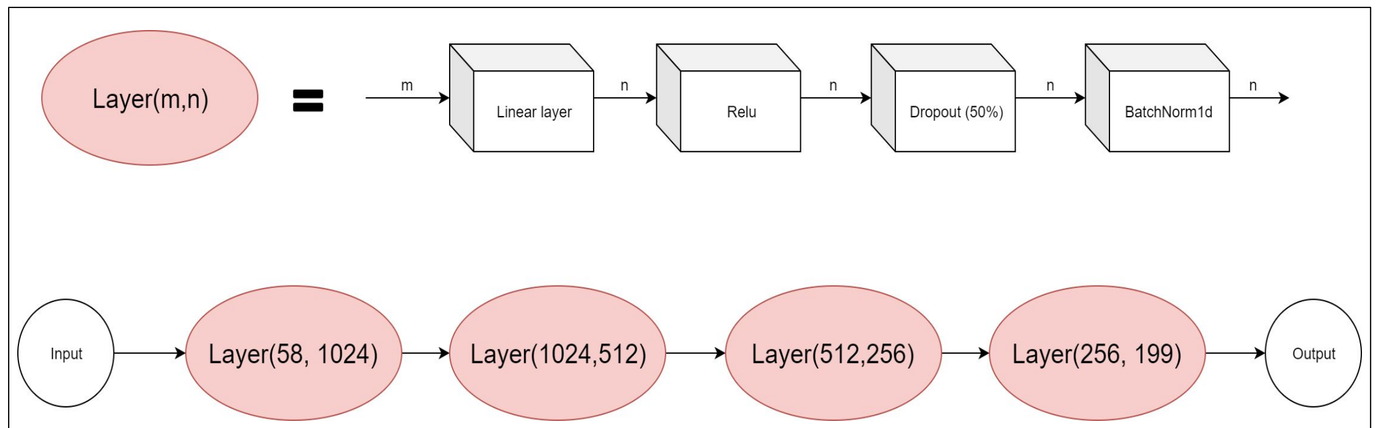
Approach:

1. First, we transform the dataset from a player-level feature set to a play-level feature set.
2. We apply feature engineering to create ~50 meaningful features based on discussions in the football and ML community, some past research into this domain and some of our own intuition. We found that features related to the "rusher" and statistics about other players relative to the rusher made good features.
3. We tried several ML classification and regression techniques to achieve good performance on the yardage prediction task, such as Decision Trees, Linear Regression, Naive Bayes, Deep Neural Nets and various Ensemble methods.

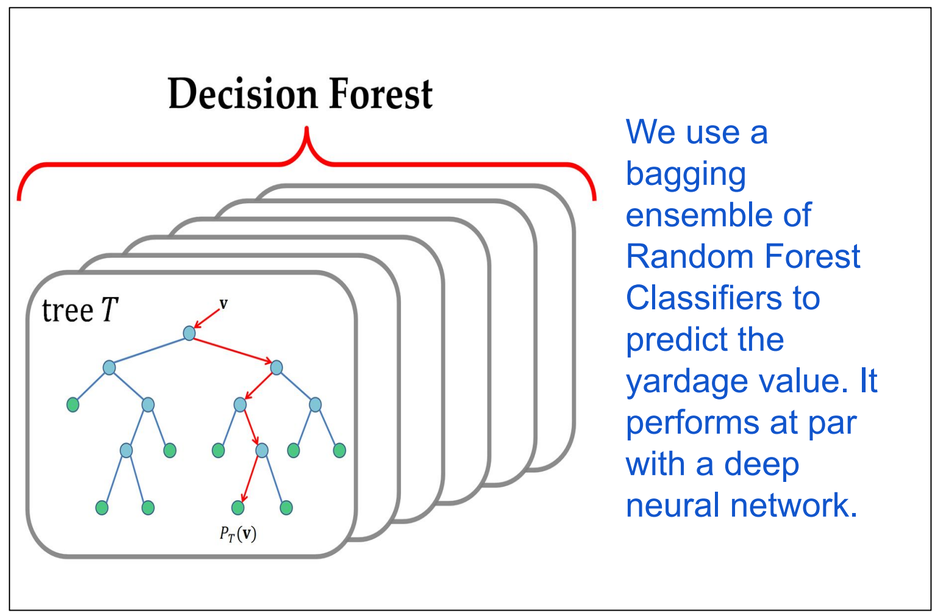
Features:

1. We discovered that utilizing features of the rusher himself provided good performance such as his distance from the line of scrimmage, the mean, min/max and std. dev. of the distance from players on the defensive team and from all players to the rusher, and one-hot encoding the rusher's direction. If too many defenders are close to him, for example, the rusher would probably not gain much yardage as he has a higher chance of being tackled and stopped.
2. We used 'DefendersInTheBox' to identify the ratio of the number of defenders to the number of yards left for a first down. We believed this would influence how motivated defenders were on preventing yardage gained by offense and thus influence the yardage gained on the play
3. We included features like the time delta of the handoff since the faster the running back could receive the ball, the faster the play could be executed and the more yardage could be gained
4. We utilized features like Quarter, Down number and formation (with one-hot encoding) since those could influence what the possession team was attempting to do like punt the ball on the last down.
5. We included the score difference between teams since teams adjust strategies depending on how far they are lagging or leading.





We use a deep neural network with 4 main units, each made up of 4 layers: Linear, Relu, DropOut, and BatchNorm. Neural Nets gave the best overall performance in our testing along with a random forest ensemble.



We use a bagging ensemble of Random Forest Classifiers to predict the yardage value. It performs at par with a deep neural network.

Results:

- We achieved pretty poor performance on the CRPS task using just the simple ML methods such as Naive Bayes and Decision Trees.
- However, ensemble of Decision Tree and Random Forest Classifiers achieved very good performance on the CRPS metric.
- The best performance was achieved using a Neural Net with a fairly standard architecture as shown.
- Despite the simplicity of the RFCs method, their performance was at par with our Neural Net Model.
- On Kaggle, this score would put us in the **top-30** of the ~2000 participants. We must also note that **our training data is 25% smaller** than theirs due to having to carve out a test dataset.
- Performance on the regression task (i.e. trying to directly find the yardage gained) is poor using the several methods we tried, with an L1 error > 3 yards. We suspect that this is because the feature set describing just the state at the start of the play is not sufficient for such a task.
- Features relating to the "rusher" and features of other players relative to the "rusher" seem to do a good job at estimating the distribution of possible yardage gain values.

Method	CRPS Score
Gaussian Naive Bayes	0.162
Decision Tree	0.026
Random Forest Bagging	**0.0131**
Deep Neural Network	**0.0131**
State of the Art (as per Kaggle Leaderboard)	*0.0120*

Potential Future Work:

We foresee the following potential avenues for getting even better performance on this problem:
- Using an ensemble of neural nets. As we mentioned, while Decision Trees by themselves didn't perform well, a bagging ensemble of them performed very well. We see potential in using an ensemble of shallow nets to get even better performance.
- Predicting probability distribution parameters such as mean and standard deviation, instead of the probabilities for each yardage class. The motivation comes from VAEs and the structure of the CRPS metric. Such an approach would also preserve the continuous domain structure of the prediction, which is currently lost when using classification methods. We foresee this being used with a custom differentiable loss function similar to the CRPS metric to train our networks.
- We wish to apply interpretability methods such as LIME to our classifiers to identify the set of features that correlate well with successful plays, to try and explain our model and help coaches better understand the factors leading to successful plays.