

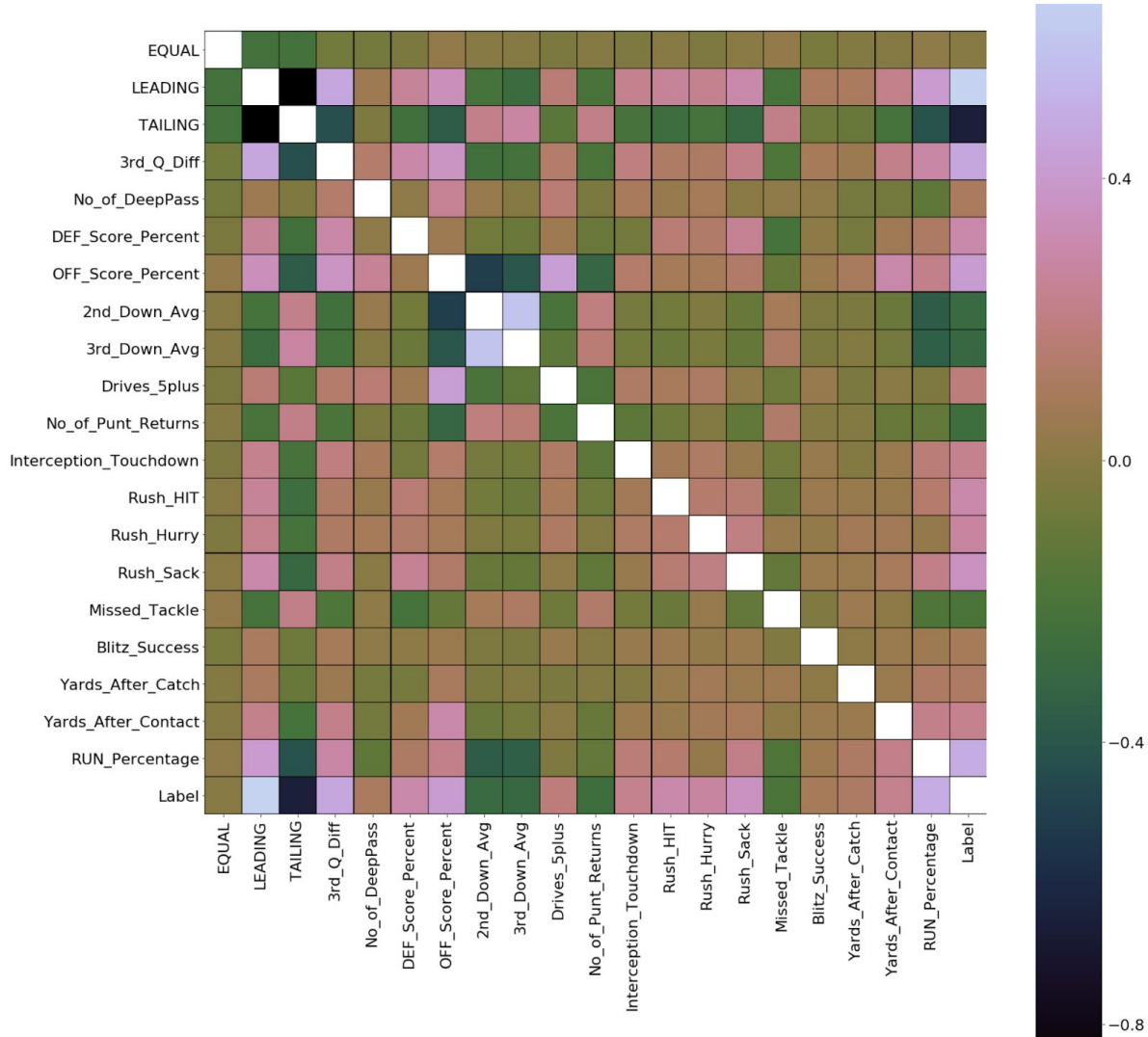
Northwestern Football Analytics Detailed Write-up

Data

The data we sourced from profootballfocus.com was already pretty clean, but since it was at the play-level, significant effort went into aggregating the play-level data into game-level features from the available columns. There are also a lot of columns and possible methods of aggregating (# of instances, sum, average, difference, etc), so we started with a handful of stats Northwestern Football curated for us, and excluded player-level stats and other columns that were exceedingly detailed such as ‘time to snap’, ‘special teams’, ‘stunts’, ‘stops’, etc.

We processed a total of 519 BIG10 game files into one file with 1038 rows (one row per team) and started with 20 features. We visualized the correlations between the features by using a heatmap:

Figure 1: Feature Correlation Heatmap where lighter colors have a higher correlation



And from this, we dropped features 'Equal', 'no_of_DeepPass', and 'Yards_After_Catch' due to them having low correlation values.

Table 1: Final Feature List

| Feature Name | Description |
|------------------------|---|
| Leading | if team is leading at halftime |
| Tailing | if team is trailing at halftime |
| 3rd_Q_Diff | score differential at halftime |
| Def_Score_Percent | percentage of successful defensive plays |
| Off_Score_Percent | percentage of successful offensive plays |
| 2nd_Down_Avg | average number of yards left to cover at 2nd downs |
| 3rd_Down_Avg | average number of yards left to cover at 3rd downs |
| Drives_5plus | number of offensive drives that had more than 4 consecutive plays |
| No_of_Punt_Returns | number of punt returns |
| Interception_Touchdown | number of interceptions that resulted in touchdowns |
| Rush_HIT | number of defensive plays that resulted in a hit on the QB |
| Rush_Hurry | number of defensive plays that resulted in the QB having to 'hurry' |
| Rush_Sack | number of defensive plays that resulted in a sack on the QB |
| Missed_Tackle | count of missed tackles |
| Blitz_Success | percentage of blitz that were successful |
| Yards_After_Contact | average yards gained by ballcarrier after first contact |
| Run_Percentage | percentage of offensive plays that were run instead of thrown |

Predictive Models

Because our end goal is to be able to determine which features are the most important for victory, we could not use any black-box models. We also want to provide thresholds for the continuous features that change the outcome of the game. Given these constraints, we started with the following models: ZeroR, Decision Tree, Random Forest, Logistic Regression Classifier, Support Vector Classifier, K-Nearest Neighbor, and Gradient Boosting Classifier. We intentionally excluded Naive Bayes Classifier because a lot of these features are dependent on each other, which is a failure-mode of NBC.

For all models, we used a 80:20 training:testing split on the data and 5-fold cross-validation was used for training accuracies.

Results

We used the scikit-learn package for our models, and obtained the following training accuracies prior to hyperparameter tuning (using defaults):

Table 2: Initial Model Accuracies

| Model | Initial Train Accuracy (%) |
|------------------------------|----------------------------|
| ZeroR | 50.13 |
| Decision Tree | 78.0 |
| Random Forest | 84.0 |
| Logistic Regression | 77.0 |
| Support Vector Classifier | 76.0 |
| K-Nearest Neighbor | 77.0 |
| Gradient Boosting Classifier | 85.0 |

To tune the models, we used the GridSearchCV method that helped iterate through many hyperparameter combinations at once. We found that Random Forest and Gradient Boosting classifiers provided the best training and test accuracies of approximately 85%. The results of all of the models we tried included:

Table 3: Predictive Model Results

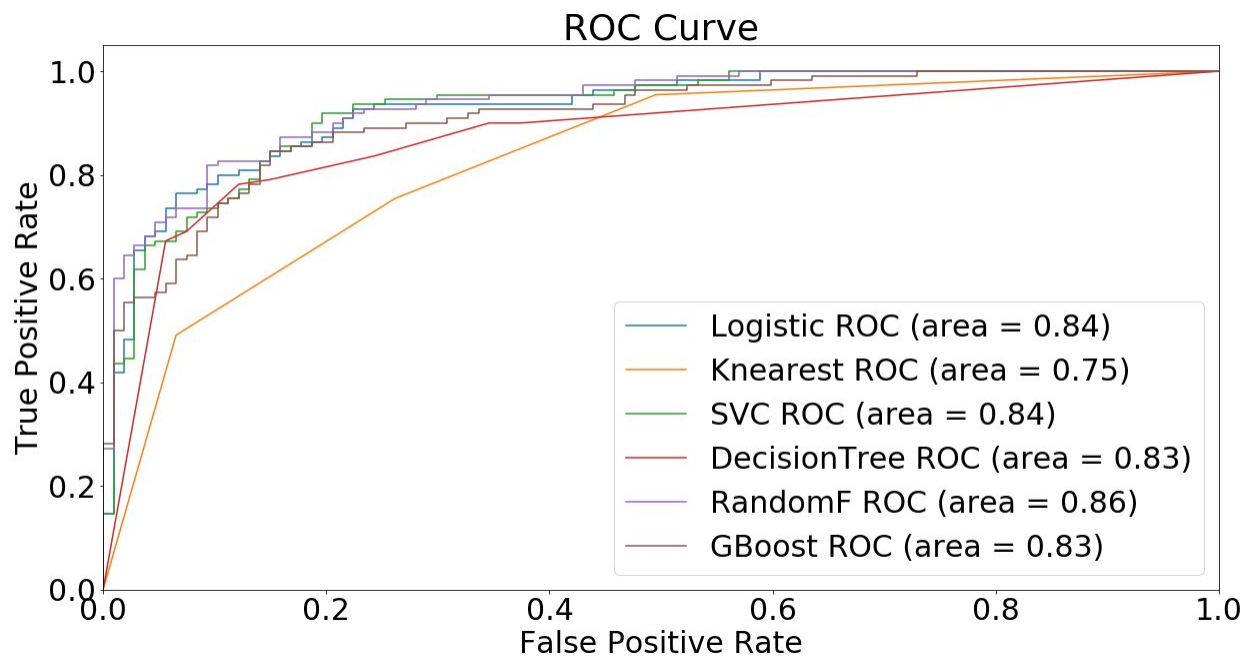
| Model | Train Accuracy | Test Accuracy |
|------------------------------|----------------|---------------|
| ZeroR | 50.13 | 49.76 |
| Decision Tree | 80.34 | 82.94 |
| Random Forest | 85.90 | 85.71 |
| Logistic Regression | 85.43 | 83.87 |
| Support Vector Classifier | 84.73 | 84.33 |
| K-Nearest Neighbor | 76.86 | 74.65 |
| Gradient Boosting Classifier | 85.55 | 82.49 |

Analysis

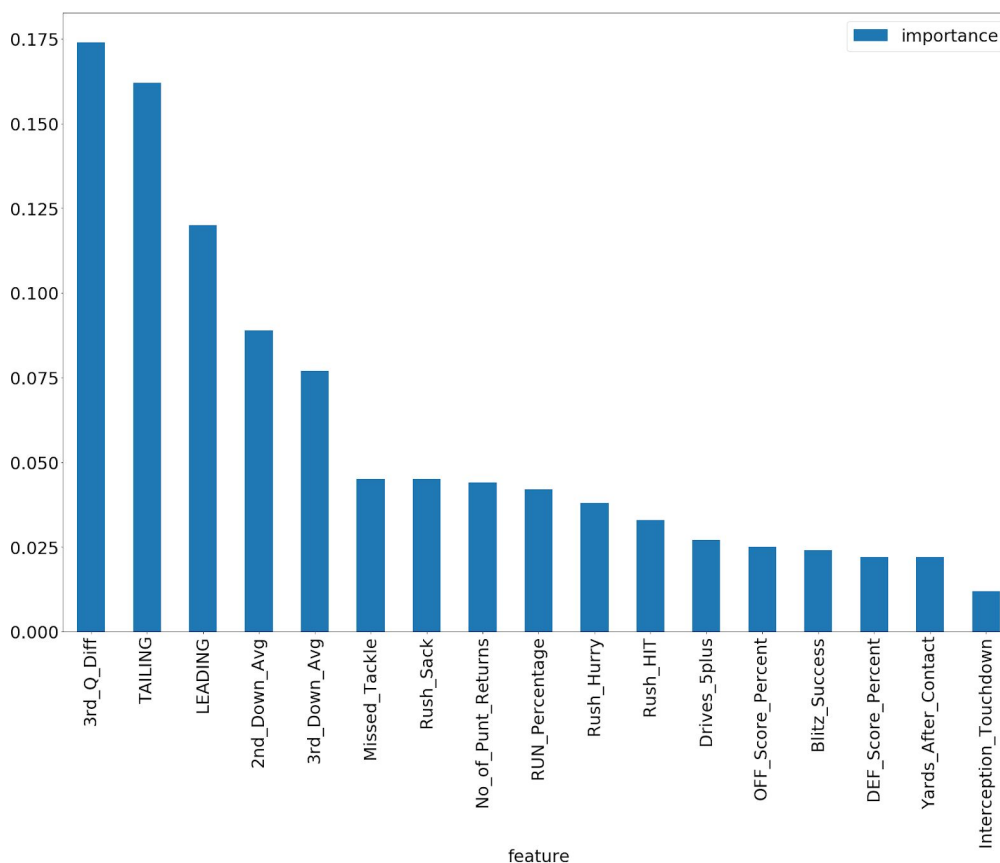
As expected, the Random Forest model out-performed the Decision Tree model as it is an ensemble method of decision trees. The K-Nearest Neighbor accuracy lacked quite a bit in accuracy in the training set, and we attribute that to not having enough data to compensate for the curse of dimensionality of having 17 features.

On top of just looking at training and test accuracies, an additional way we evaluated the models included the Receiver Operating Characteristic (ROC) metric that shows the true positive rate across the false positive rate. With a higher area under the curve meaning there is higher sensitivity and specificity, we can conclude that Random Forest is the best model based off of this metric.

Figure 2: ROC Plot of all models



We extracted the feature importances from the Random Forest model in order of importance:

Figure 3: Random Forest Feature Importances

Conclusion & Future Plans

We were satisfied with the accuracies we were able to obtain from the models we tuned, and the Random Forest model performed the best for our purposes. From this model, we were able to identify '3rd_Q_Diff', 'Tailing', 'Leading', '2nd_Down_Avg' and '3rd_Down_Avg' as the most important features contributing to victory.

Moving forward, we want to be able to find threshold values for each of these features. However, since the threshold for a given feature that will change the outcome of the game from a loss to a win varies depending on the values of all of the other features, we aren't able to get a static threshold for each of the features that will ensure victory. Instead, we plan on building out a dynamic analysis tool that takes in values for each feature, predicts the win %, and provides the cutoff thresholds for each feature given the input values. We would also like to be able to built out models specific to each BIG10 team. That way, Northwestern can see what attributes are the most important in order to beat a specific team.

Team Contribution: Vamsi led the charge for the bulk of the data aggregation, and coding of the models. Eric focused on high-level strategies, coordination with NU Football and analysis write-up. Noah built out the website

