A red and white logo

Description automatically generated

**Northeastern University**

ALY6020: Predictive Analytic

Professor: Justin Grosz

Module 4 Project

Methapat Vorakamolpisit

11/23/24

**Introduction**

This report focuses on predicting NFL play types—whether a team will opt for a pass or run—using two distinct machine learning models: XGBoost Classifier (XGBClassifier) and Random Forest Classifier. By analyzing game attributes, the report aims to identify patterns that influence play-calling strategies. The analysis uses two datasets: a training dataset with 33,957 rows and 18 columns and a test dataset with 1,643 rows and 27 columns.

XGBClassifier, a gradient boosting algorithm, is chosen for its ability to handle non-linear relationships and its robustness against overfitting through regularization techniques. On the other hand, Random Forest Classifier, an ensemble learning method, leverages decision trees to provide high accuracy and robustness by averaging the predictions of multiple trees, reducing variance and bias. Both models are compared to assess their performance in identifying key game patterns and providing insights into optimal play-calling strategies.

**Data Preparation**

Null Values

In the training dataset, there are three columns with null values: **DOWN, POINTS** **SCORED BY EITHER TEAM**, and **YARDS GAINED**. For the **DOWN** column, which is categorical data, it is not appropriate to use the mean or median to fill missing values. Instead, the mode, representing the most frequently occurring value, is chosen to fill the 121 null values in the training dataset. This ensures the data remains consistent without introducing unnecessary bias. However, in the test dataset, where there are only 5 null values in the DOWN column, the impact of removing these rows is minimal. Therefore, the decision was made to drop rows with null values in DOWN from the test dataset to maintain data integrity.

Secondly, the **POINTS SCORED BY EITHER TEAM** column has 32,590 null values in the training dataset, which constitutes a significant portion of the data. Retaining this column would introduce considerable noise and reduce model reliability. Since the column has a high level of missingness, the most effective solution is to drop it entirely from both the training and test datasets.

A graph of a number of yards

Description automatically generatedLastly, the **YARDS GAINED** column contains 8,013 null values in the training dataset. Given the skewed distribution of this variable, using the mean to fill missing values could be misleading, as it would be heavily influenced by extreme values. Instead, the median, which represents the middle value and is less affected by outliers, is the most appropriate choice for imputing the null values in this column. This approach preserves the integrity of the dataset while addressing the missing data effectively.

Features Engineering

There are six new variables were created to enhance the analysis: **Remain\_Time\_in\_Seconds, red\_zone, late quarter, TO\_GO\_DOWN\_INTERACTION, previous\_pass\_attempts, and previous\_pass\_attempts** . The **Remain\_Time\_in\_Seconds variable** is derived from the **REMAINING TIME IN THE QUARTER (mm)** column by converting the time into seconds. This is achieved by multiplying the minute value by 60 and adding the remaining seconds, providing a single numerical representation of the remaining time in seconds. The **red\_zone** variable is derived from the **YARD LINE 0-100** column, where values less than or equal to 20, indicating proximity to the end zone, are assigned a value of 1, and all other values are assigned 0. These transformations provide meaningful features for predictive modeling, offering insights into time and scoring opportunities on the field. **Late Quarter** is derive from the **Remain\_Time\_In\_Seconds**, which if the remain time is more than or equal to 120 seconds or two minute the value will be 1 otherwise 0. **TO\_GO\_DOWN\_INTERACTION** comes from To Go multiply by Down. The interaction between TO GO (yards needed for a first down) and DOWN (the current play's down) is critical because it captures the situational context of a play, which can vary significantly based on these two variables. For example, needing 3 yards to achieve a first down has very different implications on 1st down versus 4th down. On 1st down, a team has three more opportunities to achieve the required yardage, so the situation is less urgent. However, on 4th down, failing to gain those 3 yards would typically result in a turnover, making the play far more critical. Lastly, **previous\_pass\_attempts and previous\_pass\_attempts** are derived from the count of run and pass in previous plays of the offensive team, which will count group by the game\_id and the offensive team name.

A graph with a blue rectangular bar

Description automatically generated with medium confidenceA graph with a blue rectangular bar

Description automatically generated with medium confidenceOutlier

For both datasets, the columns selected for inclusion in the model are **QUARTER, DOWN, TO GO, YARD LINE 0-100, Remain\_Time\_in\_Seconds, red\_zone, late quarter, TO\_GO\_DOWN\_INTERACTION, previous\_pass\_attempts, and previous\_pass\_attempts.** These variables were chosen because of their relevance to the target variable, which is the **play type**. Among these variables, **TO GO** was identified as having outliers, which could negatively impact the model's performance by skewing predictions. To ensure accurate and reliable predictions, the outliers in the **TO GO** variable were eliminated, maintaining the integrity of the data for effective modeling.

**Exploratory Data Analysis**

A graph of a number of blue and orange bars

Description automatically generated The hypothesis suggests that in the red zone, teams are more likely to run than pass. However, as illustrated by the bar chart on the left, the number of pass and run plays in the red zone appears nearly equal, indicating that the hypothesis is incorrect. This finding suggests that teams do not have a strong preference for running over passing within this critical scoring area. Additionally, the chart reveals that outside the red zone, teams tend to pass more frequently than run, further challenging the notion that proximity to the end zone significantly alters play-calling tendencies.

**Building Best Model**

XGBoost Classifier and Random Forest Classifier are two powerful tree-based ensemble methods, each with distinct advantages and disadvantages. XGBoost generally tends to have higher accuracy compared to Random Forest, particularly when fine-tuned, because it builds trees sequentially and focuses on correcting the errors of previous trees. This gradient boosting mechanism allows XGBoost to capture subtle patterns and complex interactions in data. However, one downside of XGBoost is its limited interpretability. While it provides feature importance metrics, these can be less intuitive compared to Random Forest, and techniques like SHAP values are often required for a deeper understanding. Additionally, XGBoost requires more computational resources and time to train due to its sequential nature, which can be a bottleneck for very large datasets.

In contrast, Random Forest excels in simplicity and efficiency. It builds trees independently in parallel, making it faster to train and more suitable for large datasets with straightforward patterns. Random Forest also provides more interpretable feature importance measures, as it calculates the mean decrease in impurity for each feature across all trees. However, it typically achieves slightly lower accuracy than XGBoost, especially for datasets with complex interactions. Furthermore, Random Forest lacks explicit regularization mechanisms like XGBoost, which can make it prone to overfitting in certain scenarios, though this can often be mitigated by tuning parameters such as tree depth or the number of features considered for splits.

For this project, where the primary objective is to achieve the highest possible accuracy, I will use XGBoost Classifier. Its ability to handle complex interactions, optimize through regularization, and focus on minimizing errors makes it the most suitable model for this purpose. While it requires more computational resources and interpretability tools, the potential improvement in accuracy justifies its selection for this project.

XGBoost Classifier

A diagram of a confusion matrix

Description automatically generated

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Precision 0 | Recall 0 | F1score 0 | Precision 1 | Recall 1 | F1score 1 |
| XGB Classifier | 0.665 | 0.72 | 0.72 | 0.72 | 0.58 | 0.58 | 0.58 |

The confusion matrix evaluates the performance of the classification model in predicting two classes: **Run** and **Pass**. The results indicate that the model performs better at correctly identifying "Run" plays, as reflected in the **true positives (689)**, where the model accurately predicted these plays as "Run." However, the model also misclassified **266 plays** as "Run" when they were actually "Pass" (**false positives**), suggesting a tendency to overpredict "Run" in some cases. Similarly, while the model successfully classified **373 plays** as "Pass" (**true negatives**), it failed to predict **269 plays** as "Run," instead misclassifying them as "Pass" (**false negatives**). This demonstrates a comparable level of misclassification for both false positives and false negatives, indicating that the model struggles to distinguish between the two classes, particularly in borderline scenarios.

The evaluation metrics further reveal these challenges. The model achieves an overall accuracy of **66.5%**, meaning about two-thirds of the predictions were correct. For the "Run" class, the **precision, recall, and F1-score** are all **72%**, indicating a consistent and relatively reliable performance for this class. In contrast, the "Pass" class metrics show lower values, with the precision, recall, and F1-score at **58%**, reflecting the model's difficulty in accurately predicting "Pass" plays. This disparity suggests that the model is biased toward predicting "Run," which may stem from class imbalance in the dataset, insufficient features specific to "Pass" plays, or limitations in the current model configuration.

Overall, the confusion matrix and metrics highlight that while the model is moderately effective at predicting "Run" plays, its performance for "Pass" plays needs significant improvement. Addressing these issues could involve techniques such as **class balancing**, where the minority "Pass" class is oversampled or the majority "Run" class is undersampled, or applying **class weights** to account for the imbalance. Additionally, refining features to better capture the nuances of "Pass" plays or introducing new ones could enhance the model's predictive capabilities. Finally, **hyperparameter tuning** of the XGBoost model—such as adjusting scale\_pos\_weight, max\_depth, or learning\_rate—could optimize its performance. With these improvements, the model could achieve better precision and recall for both classes, leading to more reliable and balanced overall predictions.

Lastly, the Variance Inflation Factor (VIF) test has been performed to test multicollinearity among independent variables in a model. VIF of all the variables suggests that there are multiple variables that have moderate multicollinearity, which are Quarter, Down, To Go, and Down To Go Interaction. Among this variable, Quarter has the highest Vif which is 10, following with To Go, Down To Go Interaction, and Down at 9.7, 8.9 and 7.9 respectively.

**Final Insight**

A graph with numbers and text

Description automatically generated

The feature importance analysis from the XGBoost Classifier highlights the key drivers influencing play-calling decisions in NFL games, with the F-score used to measure each feature's contribution. Among the features, Remain\_Time\_in\_Seconds emerges as the most influential (F-score: 1173), underscoring the significant role of remaining time in strategic decision-making, particularly in late-game situations where urgency dictates play-calling. The second most important feature, YARD LINE 0-100 (F-score: 961), reflects the impact of field position, as plays near the end zone or red zone often dictate whether a "Run" or "Pass" is more likely. Historical tendencies also play a crucial role, with previous\_pass\_attempts (F-score: 744) and previous\_run\_attempts (F-score: 699) ranking highly, indicating that patterns from earlier plays strongly influence current decisions.

Additionally, the interaction feature TO\_GO\_DOWN\_INTERACTION (F-score: 416) shows moderate importance, demonstrating that combining down and yards-to-go provides more valuable situational context than their individual components, which have lower F-scores (TO GO: 288, DOWN: 180). Meanwhile, QUARTER (F-score: 169) is the least important feature, suggesting that the quarter of play has minimal influence compared to other situational and temporal variables. Overall, the model prioritizes features that capture the game's situational and temporal dynamics, such as time remaining, field position, and historical play patterns, while features like QUARTER or individual components of interactions contribute less. Moving forward, refining and retaining the highly ranked features while deprioritizing or testing the removal of less influential ones could further optimize the model's performance.

**Appendix**

XGB Classifier

A screenshot of a computer screen

Description automatically generated

A diagram of a confusion matrix

Description automatically generated

A graph with numbers and text

Description automatically generatedRandom Forrest

A screenshot of a computer screen

Description automatically generated

VIF

A screen shot of a computer

Description automatically generated