Florida State University Play Picker

CIS 4930: Data Mining

Haleigh Gahan, Kian Haghighi, Rohith Gottiparthim,
Christopher Tucker, & Jonas Wilson
Github: https://github.com/rgottiparthi/FSU-Play-Picker

I. Introduction

FSU Football has seen massive success as of late, but their efforts still left them falling short of the College Football Playoffs. Despite FSU's perfect record, Coach Norvell has made some questionable decisions this season and the team has struggled offensively. This experiment focuses on using deep learning to assist in his decision making.

Data was gathered for every game during the Fall 2023 season and relevant information to guide in play decisions was extracted. Initially, the objective was to create predictions for both offensive and defensive play calls, but there lacked a substantial amount of data. More importantly, this type of analysis is enormously complex, and not feasible to develop extensively given the time constraints of a semester project. For these reasons, it was decided that the model would be geared towards offensive calls. Time, quarter, position in yards, current down, distance to first/goal, and current score were deemed the most critical features, which were analyzed in the extracted datasets. The model employs the Keras library to aid in predicting the play type and optimal outcome value.

II. Literature Survey

Since American Football is historically a relatively statistical sport and is linked to significant investments and capital, the research in this sector is broad. Numerous variations of statistical models have been studied, but it is difficult to draw generalized conclusions as there is a large unpredictable component to the game. Football fans are aware that upsets occur frequently and the predicted game score is rarely accurate, even if the favored opponent comes out on top. Also, players get injured frequently, so the constantly changing roster makes predictions even more challenging. Recent attempts to apply technology to sports predictions include using ML models and AI to not only analyze quantitative statistics, but also film.

An extensive study conducted at the University of Central Florida compared studying football plays to military battles. Both are competitive and require communication and coordination of participants to execute an operation successfully. In a similar vein, one teammate's poor decision or execution can cause the entire operation to fail. Simultaneously, the opponents actions complicate the execution and generate additional considerations.

This study looked at the Rush 2008 football simulator, which was successful in "simulating complex plays, yet sufficiently lightweight to facilitate running machine learning experiments." Researchers used an established playbook, and defined their objectives to be guiding selections via modeling, identifying patterns from historical plays, generating new plays, and using all of this information to develop a model that learns in real-time to make data-driven

& actionable suggestions for the 3 most critical players at the onset of the play. Neglected by this research, and most of its kind, is opponent modeling, as focusing on offensive decisions appears to significantly streamline development and design. Hence on the defensive end, researchers generally seek to investigate identifying patterns & their opponents plans as quickly as possible. However this presents several practical challenges, as modeling human behavior is not always accurate and players need to respond to what is occurring in real time, instead of keeping a mental playbook. In addition to only including 8 players from each team on the field at a time, the Rush 2008 study is limited to control over the quarterback, where the remaining players on the field are controlled autonomously according to the designed execution. There is an obvious vulnerability in this scheme, as errors can very easily and often do occur by others. The receiver is often equally responsible for the result of many passing plays. A skilled defense could also easily prevent execution of the designed play.

Rush 2008 relies on Support Vector Machines (SVM), more specifically a multi-class SVM, to identify defensive actions quickly, hence responding with optimal offensive efforts. As with most ML algorithms, the classification improves with time as more observations become available. The model was trained with 40 case scenarios, using every possible permutation of the 8 offensive & 8 defensive plays outlined in the playbook, along with the 12 possible configurations for an initial formation. The designers applied a cross-validation protocol in order to calibrate the parameters of the training data.

The report states that classification during testing occurs rapidly and begins by choosing the SVM pertinent to the starting formation & time. Following is the generation of an observation vector, which serves to gather data as the game takes place, and add it to the SVM. With this information, the model generates a suggestion that accounts for the opponents defensive decisions and desired point rewards. This fails to consider several critical factors such as the players and their strengths, field position, and game situation (down, time left, etc), but the observed successes and failures can be instrumental in guiding future research.

An alternative study aims to exploit a Bayesian model to make in-play predictions in the sport universally deemed football, but known as soccer in America. Although the games differ in play and principle, the objective remains to move down the field and get the ball though the other team's goal, while protecting your own. The most sports-prediction research is in this area due to soccer being the most popular worldwide and betting practices have prevailed in recent years. Hence, this study is useful and relevant as these methods may be useful in more accurately predicting outcomes in American football. This research is particularly interesting because they separate the complexity of winning a match into two separate categories, pregame and in-play.

Studies regarding pre-match predictions initially pointed to the Negative Binomial model to establish an associate rule. But later research suggests that the Poisson distribution goal-based data analysis is favorable, as it accounts for the offensive and defensive strengths of both opponents. However this model also bears witness to challenges, as it can only be applied in situations where data is time-homogenous and equi-dispersed. Another plausible solution was the count model based on Weibull interarrival times, proposed due to its ability to manage both under and over-dispersed data. When tested, the Weibull model performed the best relative to the other options. More recently, a probabilistic model was introduced that sought to predict whether a team would meet their championship goals based on studying scores from other matches, but most research has been directed at improving the previous methods.

These improvements have contributed to the hierarchical Bayesian Poisson model, defined by scoring rates convexing to combinations of parameters. Match outcomes are predicted using approximations from historical data and "betting odds." Based on this well-developed model, the study in discussion expanded their predictions to account for team strengths and weaknesses using the available player matrix & player data.

The in-play curiosities were motivated by a rise in live betting. The research in this domain is much less extensive and far more recent. The first attempt resulted in a pure birth process model, where goal times of both teams were fed to two nonhomogeneous poisson processes, aligned in a discrete-time and finite-state Markov chain model. A recursive algorithm was used to calculate outcome probabilities. However, practicality challenges were met as it does not consider the in-match data to consistently update information regarding the teams strengths, assuming abilities remain constant throughout the entirety of the game. It estimates the strength parameters of the opponent by estimating on the basis of all other teams in history, and some of this information may not be relevant. On the contrary, it only considers the time at which the play is occurring, enhancing the accuracy by reducing transition time to the final score. Other researchers developed a semiparametric model with a non-parametric baseline intensity. The benefit of this process is that the regression component considers the actual state of the game and strength of the defense. Further investigation into this model may solve some of these challenges.

Ideally, a model would consider both metrics from historical information and in-match information. This study proposes a Bayesian classifier to achieve this estimation and calibrate recommendations based on the team's strengths.

III. Relevant Terminology

Given that this study pertains to American Football, it is important to define any terminology or jargon that will be used in our report or is applicable to the potential situations outlined in the use cases of the model. For the purpose of this experiment, we define a Football game as a 2-team sport where opponents aim to score the highest number of points.

Scoring System

This can be accomplished via catching the ball in or running the ball into the opposing team's endzone for a touchdown (6 points) or kicking the ball within the crossbars (or above, given within the lateral boundaries) of the field goal (3 points). After each touchdown, the scoring team is given the choice of attempting an extra point by kicking from the 3-yard line (1 point) or what is referred to as a 2-point conversion, where the team must once again successfully move the ball over the goal line, but this time in a single-attempt from the 3-yard line (2 points). Additionally, if the quarterback is sacked in the offensive team's endzone, the defensive opponent is given points for what is known as a safety (2 points).

Objective/ Basic Rules

The game is played where the offensive team is given 4 opportunities to move the ball 10 yards. If the threshold is not met, the team moves onto the succeeding "down." If the 10 yards is accomplished, the team gets a fresh set of downs. In the FBS, or Division I, the average amount of yards gained per play is 6.45. For Florida State, the average is 6.64 This type of play continues as the team (theoretically) moves further towards the opposing teams endzone to score, kick a field goal, choose to punt on 4th down (offering the other team possession of the ball), or give up possession at the current position, if 10 yards is not achieved on 4th down. A coin toss is carried out prior to kickoff at every game, and the winning party can decide whether to start the game with possession of the ball or get the ball back at the start of the second half (typically the more popular choice). There are 4 quarters, each consisting of 15 minutes of play. Current possession of the ball is not affected on the transition from the 1st to the 2nd quarter, or the 3rd to the 4th quarter.

Field Layout

The center of the field is denoted by the "shared" 50 yard line, and the line number decreases until it reaches each teams' "0-yard line," also known as the goal line, at the entrance to the endzone. The line of scrimmage is an imaginary line that changes based on the current field position of the ball, but it is the location from which the ball is snapped. The Backfield is the area behind the line of scrimmage, where each play typically begins.

Positions/ Types of Players

Each team is permitted 11 players on the field at a time. The *long snapper* snaps the ball to the *kicker* on punts and kicks. On offense, there are *quarterbacks* who typically get the ball from the center and make decisions about the play. Usually, this involves either handing it off to a *running back* who rushes down the field, run the ball themselves, or pass it to another player on their team. The play is "live" until the ball hits the ground, is thrown out of bounds, or the player with the ball is tackled. The *offensive lineman* are responsible for keeping the opposing defenseman from reaching the *quarterback* or whoever has the ball. *Fullbacks* are a combination of an offensive lineman and a *running back*, and are mainly tasked with protecting the running back, but sometimes carry the ball as well. *Wide receivers* are the targets for passes, tasked with running as far as possible and then receiving a pass from the quarterback. Tight ends start next to the offensive line, and function as a hybrid between a wide receiver and offensive linesman, serving as potential pass catchers but also blocking defenseman.

There will be minimal discussion of defense in this experiment, since the model makes recommendations for offensive situations. However, it may be helpful to note that the defensive line is composed of the players that aim to reach the quarterback for a "sack" (tackle before the ball is thrown), or whatever offensive player has possession of the ball to keep them from moving down the field. *Linebackers* aim to stop running backs and tackle receivers, and cornerbacks play behind the defense, trying to block the offense's best receivers from catching the ball.

IV. Defining Methods

In order to conduct this experiment, numerical values had to be attached in order to define the "goodness" or success of a play. A mechanism had to be developed that combined any movement towards the end zone as a positive result, while still taking into consideration the fact that moving the ball to the opponent's 1-yard line with no scoring success was equivalent to no success at all in terms of contributing to a winning result. It was decided that a play value could be approximated by adding the number of yards gained (or lost = negative) to a predetermined weight of the outcome. The outcomes were defined as 210 for a touchdown, 90 for a field goal, 30 for a successful extra point kick, 60 for a 2-point conversion, -60 for a safety, -100 for a turnover or interception. This scale was determined by equating every point to the value 30, and then scaling this value based on the number of game points that the outcome is worth. In the case of a turnover, there is no good measure of how likely it is that the other team scores consequently, or how much this outcome contributed to that score (as opposed to a turnover on downs, punt, etc). Therefore, we define this case scenario as roughly half as "bad" as a touchdown for the other team, and take into account the starting field position for the other team, as this is a large component of the supposed likelihood of the opponent's forthcoming success.

Attribute	Definition	Range
Time	Time left in the quarter	0-15:00
Quarter	Current quarter in the game	1-4
Position in yards	Yards from endzone	1-99
Down	Current down for FSU	1-4
Distance to first down/goal	Yards needed to achieve a first down or goal if in range	0-99
Current Score	Current score in the game(FSU-opponent)	$(0, \infty)$ - $(0, \infty)$

V. Data Acquisition

Given that the goal is to develop a model that makes predictions for Florida State's current team, coaches, and playbook, we restrict the datasets used for both training and testing to the 2022 & 2023 seasons. Play-by-plays were extracted from Seminoles.com, and a python parser was used to extract all relevant information: time left in game, field position, down number, current score, yards to a first down, etc. 70% of the data was randomly assigned to the training set, and we reserved the remaining 30% for testing purposes.

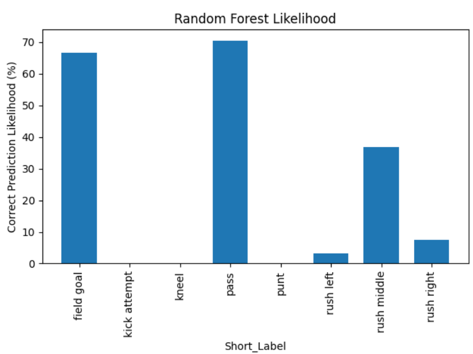
VI. Description of Implementation

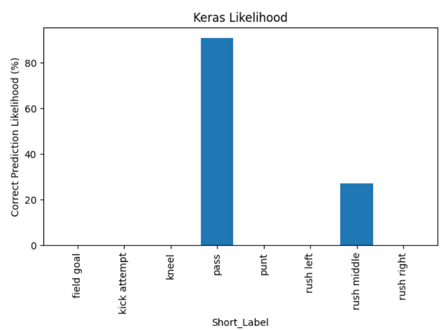
The algorithm was implemented using Tensorflow for the creation of the neural networks for deep learning practices. Keras, an open-source machine package developed by Google, was used to support model creation and training. Sklearn, a Python machine learning library that provides simple and effective tools for data analysis and modeling, aided in preprocessing and loading data from the .txt files. Numpy assisted in basic mathematical computations, and was chosen due to its simplicity, popularity, & ease of use. It is a Python package designed for scientific computing, providing support for large, multi–dimensional arrays and matrices, along with mathematical functions to operate on these arrays. Finally, Pandas aided in manipulation of data and to assign certain weights to the outcomes.

First, the developed parser must be used to extract the relevant data from the Play-By-Play datasets and then writing this information into a .csv file. Then the implementation begins by importing the libraries discussed above and loading in the data from the .csv file, 'plays.csv,' into a Pandas DataFrame. Next, the features (x) and the target variable which represents the "best" or suggested play (y) are extracted from the dataset. The categorical labels are encoded using the sklearn preprocessing LabelEncoder function, assigning targets with a value between 0 and the number of classes-1. At this point, the data can be split into training and test sets using the train_test_split method with the stratify parameter set to y, to ensure distribution of play types is maintained in both sets. It was decided that 70% should be used for training an accurate model, and 30% should be reserved for testing. This distribution and assignment is done through random selection, and in this case uses the train_test_split method, a random state of 42 to ensure consistency during testing, and stratify=y tin order to prevent class imbalance and provide an even distribution amongst sets. The features are standardized, if necessary, using the StandardScaler and the transform (on test set) and transform_fit (on training set), which are sklearn methods. This standardization brings the features to a common scale, thereby enhancing model performance. The neural network model is ready to be built with the Kerbas library and compiled, which will consist of input and output layers, with a hidden layer in between. To improve convergence, we also define a learning rate scheduler (magnitude of change to update model weights during training) using the Adam optimizer. Adam was selected based on its updating characteristic of separation for each parameter and weight. The model is also compiled with sparse categorical crossentropy loss. The training occurs using sample weights. Finally, we can evaluate the model by calculating loss and accuracy, and use these metrics to make predictions regarding test data. Once these predictions are established, predictions are converted back to class labels to generate an output DataFrame (output_df) of actual and predicted play types, along with associated outcome weights. We experimented with using Drop Out to avoid overfitting our model, but it did not seem to improve the accuracy by a statistically significant amount. Since drop outs randomly set a fraction of input units to zero during training, perhaps this function would be of use if we were working with a larger dataset.

A randomized model, Random Forest, was also developed to evaluate if training a model using past results had an impact on successful outcomes over picking plays at pure random. The first steps of this implementation were the same, from loading and preparing data, extracting features and encoding class labels, splitting the dataset, and standardizing features. However when it came to the model building phase, the scikit-learn library was used to develop a classification model with 100 decision trees (n_estimators=100) and a specified random state for reproducibility. Similarly, the Random Forest model was evaluated on the test set and the accuracy was computed using sckit-learn's accuracy_score function. The trained model made predictions on the test set, which were converted back to class labels to be outputted using a DataFrame (output_df). The accuracy of the play predictions are analyzed and compared to that of the model trained with the actual algorithm.

VII. Results





Our keras model only successfully predicted rush middle and run. Pass was predicted correctly over 80% of the time and rush middle was predicted correctly roughly 20% of the time. The keras model predicts plays accurately 43%, despite only predicting pass and rush middle. Our random forest model predicted plays 40.83% correctly. Field goal attempts and pass plays had the highest correct prediction, both over 60%. Rush plays performed similarly, with rush middle as the highest correctly predicted output. However, our random forest model was not able to successfully predict kick attempts, kneels, and punts.

VIII. Conclusions

Our keras model performed slightly better than our random forest model, despite our keras model only predicting 2 out of the 9 play types. Both of models would have performed significantly better with more data.

When we began the project, we aimed to create a play predictor for both offense and defense. However, there lacked enough data in order to create a model that would predict defensive plays.

In hindsight, it is evident that our models could have potentially yielded better results with a more extensive dataset. The limited range of play types predicted by the models suggests that an expanded dataset could enhance the models' predictive capabilities. Moving forward, acquiring additional data and refining our models could be instrumental in achieving more comprehensive and accurate play predictions.

IX. Future Research

There are several mechanisms by which this experiment could be expanded in the future to be more useful and real-world applicable. On a basic level, the second layer could be changed to include player data for future teams, so recommendations could be made based on the current roster and player availability. If we were to take this next step, it would be favorable to expand our data from other seasons for both training and testing. Additionally, we could further develop our analysis and recommendation scheme to include defensive plays. Studying the opponent and their respective strengths could be helpful in predicting success on both offense and defense. Given unlimited time and resources, this model could account for various other factors that affect performance and weigh into decision making on the field: location (home vs away, time zone travel), stadium conditions (noise, weather), and upcoming games & opponents. In some situations, it is valuable to look beyond the scope of the current game, and make coaching decisions that may not be the most favorable to scoring points in this exact moment. These include, but are not limited to potential injuries, saving starting athletes for more competitive future games, giving opportunities to rising players, letting seniors play in their final games, or protecting the ball from mishaps that result in unintentional change of possession or points for the opposing team.

References

College Football Yards Per Play Statistics (2023). The Power Rank. https://thepowerrank.com/cfb-yards-per-play-stats/

Goyal, U. (2020). Leveraging Machine Learning to Predict Play Calling Tendencies in the NFL. Massachusetts Institute of Technology. https://dspace.mit.edu/bitstream/handle/1721.1/129909/1237411720-MIT.pdf?sequence=1&isAllowed=y

Laviersa, K. R., Sukthankar, G. (2014). Using Opponent Modeling to Adapt Team Play in American Football. *Plan, Activity, and Intent Recognition, Elsevier.* https://www.academia.edu/download/78422086/Sukthankar-PAIRBook2014.pdf

Presto, G. A Beginners Guide to American Football Positions (2023, February 13). Nike. https://www.nike.com/a/football-positions

Zou Q., Song K., & Shi J. (2020). A Bayesian In-Play Prediction Model for Association Football Outcomes. Applied Sciences. 10(8):2904. https://doi.org/10.3390/app10082904