FAMU- FSU College of Engineering

Department of Electrical and Computer Engineering

Fall 2021

EEL-4773 Machine Learning Final Project Report

Title: NFL Football Play Predictor

Name: Brendan Laney

Instructor: Dr. Simon Foo

Date: 7 Dec 2021

Contents

1. Abstract

Football at the professional level is a very strategic of “X’s” and “O’s” considering all players are of the same caliber in comparison to college and high school. To know the outcome of an offensive play before it happens would be extremely useful to the defense trying to defend from it. Machine learning can be used and applied to accurately predict an offensive play.

2. Introduction

The purpose of this project is to predict an NFL football play. More specifically, whether an offensive play is a run or a pass. A few determining factors can be used to predict a play such as: the down of the drive, the quarter of the game, time remaining in the game/quarter, field position, and the personnel the offense lines up in. This problem is both a supervised and classification problem because it will be learning from a data source, and it will be deciding whether it is a run or pass play. Considering there are multiple variables to consider, a decision tree will be used to determine the outcome of the play.

3. Content

A large part of this project, and any machine learning project, is obtaining quality data and interpreting it. A data source was found which contains over 250 details about every single play (over 45,000) during the 2019 season. The data was filtered to remove unnecessary details.

This was accomplished by deleting certain columns in the .csv file. All columns after “score_differential_post” were removed. These columns included scoring propabilities, defensive statistics, and individual players related to the play. As well, columns before yardline_100 were removed. These columns included what team has the ball and play ID information. If the dataset were to analyze play predictions for a particular team, this information may want to be used. But for the focus of this project, it is not necessary. As well the following columns were removed: “game_date”, “game_half”, “quarter_end”, “time”, “yrdln”, “ydsnet”, “desc”, “field_goal_result”, “kick_distance”, “extra_point_result”, “two_point_conv_result”, “timeout_team”, and “td_team”. The remaining columns were then cut and pasted to be aligned next to each other to remove blank columns for viewing purposes inside the .csv file.

Next the filter function from Excel was utilized. For "play_type", all rows besides "run" and "pass" were removed because this takes away unrelated information such as no-plays, special teams, QB kneels, etc. As well, using the filter function again all QB scrambles were removed which were denoted as a "1". This is done because the purpose of this project is to predict designed run and pass plays. QB spikes and QB kneels did not have to altered because they were already removed from filtering "play_type". After this is finished, there should only be 31,304 rows.

Strings such as "pass" and "run" were converted to numeric values because strings cannot be accepted in the decision tree code used. Strings like the aforementioned were converted to numeric values by using the replace function from Excel. "Pass" and "run" are under the column "play_type" and they were converted to "run" = 1 and "pass" = 2. This will also change some columns which contain these strings (ie. pass_length will be converted to 2_length).

As well, values for these columns were changed. The following values were not used in the algorithm, but they may be used to predict the location of a pass or run in future studies. For pass location (2_location) and pass length (2_length); "right" = 1, "left" = 2, "middle" = 3, and "short" = 1, "deep" = 2 respectively. For run location (1_location) and run gap (gap); "right" = 1, "left" = 2, and "tackle" = 1, "guard" = 2, "end" = 3 respectively. "NA" which is information not that is did not happen for a play (ie. there's no pass location if it's a run) were converted to "0".

4. Simulation Results

After converting the data in the file, it was ready to be read into the code. The feature columns used were "down", "qtr", "score_differential", and "ydstogo". The target variable was "play_type" because we wanted to use the feature columns to predict what the play would be. The data was split into a 70% train and 30% test. The data than predicted the response for the dataset which was displayed. The algorithm consistently produced a prediction accuracy of 67%.

5. Analysis/Discussion of the Results

A few changes were modified to get the prediction accuracy to 67%. Originally, “half_seconds_remaining” and “yardline_100” were included along with the aforementioned feature columns but limited the prediction accuracy to 62% to 64.5%. After they were removed, they brought the prediction accuracy to a little over 65%. Then to ensure plays were designed run and pass plays, QB scrambles were removed which then brought the prediction accuracy to 67%. This is pretty good prediction accuracy considering there are a lot of uncertainties to what a football team can do because of different strategies. As long as the data is above 50%, the feature columns used could accurately predict a play call.

6. Conclusion

In conclusion an NFL football play can be predicted with an accuracy of 67%. A large part of this project was obtaining quality data and converting it to be read within a program. Once the data was interpreted, some features of the current game can be used to accurately predict an offensive play.

7. Acknowledgement

The data file was modified using my understanding of football, and removing plays that I thought were unrelated or did not contribute to the play call. The data was aquired from source [2]. The implementation of the algorithm that was used was from source [1]. There were some Github sources used for inspiration; however, they did not contribute to the actual making of the project. The main Github source used for inspiration is source [3].

8. References

[1] “Python decision tree classification with Scikit-Learn Decisiontreeclassifier,” DataCamp Community. [Online]. Available: https://www.datacamp.com/community/tutorials/decision-tree-classification-python. [Accessed: 06-Dec-2021].
[2] Ryurko, “Ryurko/NFLSCRAPR-Data: Data Files (.CSV) accessed with nflscrapr and summarized at the player-level,” GitHub. [Online].

Available: https://github.com/ryurko/nflscrapR-data. [Accessed: 06-Dec-2021].
[3] Naivelogic, "Naivelogic/NFL-smarter-football: A set of analytics and machine learning models with the goal of bring intelligence to the NFL.," GitHub. [Online]. Available: https://github.com/naivelogic/NFL-smarter-football. [Accessed: 06-Dec-2021].

9. Appendix

nfl_play_predictor.py

```python
# Load libraries
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation

path_to_file = "reg_pbp_2019_rev.csv"
play = pd.read_csv(path_to_file)

# removed qb scramble from dataset; spikes and kneels already removed due to play_type

np.nan_to_num(play)

#split dataset in features and target variable
feature_cols=['down', 'qtr', 'score_differential','ydstogo']

X = play[feature_cols] # Features
y = play.play_type # Target variable

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1) # 70% training and 30% test


# Create Decision Tree classifer object
clf = DecisionTreeClassifier()

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)

# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

Python Code

```
In [1]: runfile('C:/Users/Ryan Laney/OneDrive - Florida State Students/EEE-4773 Machine Learning/
Project/NFL Play Predictor/nfl_play_predictor.py', wdir='C:/Users/Ryan Laney/OneDrive - Florida State
Students/EEE-4773 Machine Learning/Project/NFL Play Predictor')
Accuracy: 0.6709615589394101
```

Result