**Objective and Purpose:**

The goal of this research is to predict offensive play types by “run”, “pass”, and “play action pass” (“pa pass”) using only variables that happen pre-snap. The third target of “play action” passes was included because of the implications it has on both sides of the ball, creating opportunities offense while also putting stress on mid-level defenders. While not necessarily being best suited for in-game decision making, the objective is to create actionable insights for members across a Football Ops department to better create a gameplan and enhance their ability to prepare each week. The model will strive for both reliability and interpretability, so that in combination the analyst can trust the data while making it digestible for the end user, most likely football staff.

**Data:**

Data used in this research is from the 2025 Big Data Bowl1 which uses data from the 2022 NFL Season. This includes “play”, “player play”, “game”, “player” and “tracking” data sets. Our model uses all these data sets except “tracking”. “Play” data set is used as the anchor table while the other were joined on “game id” and “possession team”. Data exclusions are as follows:

* Special teams plays excluded
* “kneel” & “spike” plays excluded
* Plays under 2 minutes in the 2nd,4th, and OT periods excluded
* First six offensive plays for each team of each game excluded (equivalent to at least “two full possessions”)
* Outliers, NaN, and null values were all excluded after EDA. Outliers were identified in EDA (see EDA section)

**Feature Engineering:**

* Personnel Feature(str): Joined “player play” and “player” to identify number of number RB & TE on each play and create categorical variable based on “personnel” each play (“21”, ”12”, ”22”, etc.)
* Qtr Minutes (float): Using “gameclock” variable from “play” dataset to get time left in each quarter at the play level
* Game Minutes(float): Using “gameclock” variable from “play” dataset to get time left in the game at the play level
* Possession Team Winning (Binary): Variable that shows whether the possessing team is currently winning the game at the play level
* Possession Team Home (Binary): Variable that shows whether the possessing team is the home team game level
* Dummy Variables: Categorical variables converted to dummy variables with one hot encoding
  + Play down: 1st, 2nd, 3rd, 4th
  + Game Quarter: 1st, 2nd, 3rd, 4th
  + Offensive Formation: All unique formations in “plays” data set
  + WR Alignment: All unique WR alignments in “plays” data set
* Running Game Level Metrics:
  + Average yards per rush running average at the game level
  + Running completion percent at the game level
* Running Season Level Metrics:
  + Average yards per rush running average at the season level
  + Running completion percent at the season level
  + Running play action call percentage at the season level
  + Running pass call to run call ratio at the season level (pass includes pa pass)

**EDA:**

Performed EDA on both original data and engineered features. The EDA included frequency distribution for categorical variables, correlation matrices for continuous variables as well as distribution for continuous variables to help draw out outliers.

A graph of a graph

Description automatically generated with medium confidence

A diagram of different colored shapes

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a graph

Description automatically generated

**Modeling:**

The two models that were chosen to predict our data were the GradientBoostingClassifier from scikitlearn and an XGBClassifier from xgboost. Taking a step back, the data used in both models was split into train and test using a 80/20 split and stratified on our target variable because of the uneven distribution of pa passes compared to pass & run. The training set was then put into a stratified K fold using 5 folds. The data was trained on both base models (no parameters specified) & using parameters after GridSearchCV parameter hypertuning was performed. Below are tables of the parameters used in the Grid Search for each model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameters | Gradient Boost (Best) | | | |
| n\_estimators | 25 | 50 | 75 | 100 |
| max\_depth | 2 | 3 | 5 | None |
| max\_features | n\_features | 7 | 5 | - |
| learn\_rate | .1 | .05 | .01 | - |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PARAMETERs | XGBOOST (Best) | | | |  |
| n\_estimators | 50 | 75 | 100 | 200 | 500 |
| max\_depth | 3 | 5 | 7 | 10 | - |
| LEARN\_RATE | .01 | .05 | .075 | .1 | - |
| subsample | .6 | .8 | 1 | - | - |

After our models have been trained, we look to both understand the reliability and interpretability of each model. We use the accuracy score to understand the reliability of each model. Looking through interpretability lens we draw out the feature importances of each model to show which variables are most impactful & important when predicting play calls. These features can help inform football staff to help enhance their gameplan, film study, etc.

**Results and Interpretation:**

Both models do good job predicting the test with Gradient Boosting seeing a 64% accuracy rate and XGBoost 65%. Both models also generalizae very well with both models having 69% accuracy when predicting train data. We see XGBoost perform better predicting play action passes and runs compared to the Gradient Boost model. While both models predict well, I would defer to the Gradient Boost to use in a real-life scenario because of its easy interpretability as seen below in the “feature importance” views.

A graph of a test

Description automatically generated with medium confidenceA blue squares with numbers and labels

Description automatically generated

A graph with a number of letters and numbers

Description automatically generatedA graph with blue lines

Description automatically generated

**Next Steps:**

Additional feature engineering could be done, including adding opponent defensive metrics. Furthermore, I think more tweaking with the parameters for each model and possibly adding Bayesian optimization when hypertuning would enhance the accuracy. XGBoost could be leveraged to predict more front office topics such as drafting, player development, etc. because of its sound reliability and less of a need for interpretability.

**Sources:**

1. https://www.kaggle.com/competitions/nfl-big-data-bowl-2025/data