CSE 6242 - Final Paper
Georgia Institute of Technology, OMS Analytics
"QBViz: A modern framework for evaluating Quarterback decision-making and relative performance"

Team 42: Cory Jez, Kyle VanderBush, Alex Furrier, Greg Romrell

April 2019

Motivation

In the world of professional football, players are evaluated based on individual performance metrics such as completion percentage and yards per attempt. We believe that there is an opportunity to improve the overall measurement of players (particularly quarterbacks) by quantifying their decisions. Recently, the NFL implemented on-field player-tracking systems which record the locations and actions of all players on the field. With this data, we believe it is possible to identify and quantify the decisions made by quarterbacks during a game. Quarterback is far and away the most important position in football; no single player possesses more impact on the outcome of the game. As such quantifying their performance as a means for analysis and future performance improvement is a vital task for any team.

Problem Definition

We have taken an opportunity-cost approach to measuring quarterback decision-making and performance above expectation by applying machine learning techniques to NFL player-tracking data. Our model has produced catch probabilities for over 20,000 eligible receivers and we have leveraged this to provide the expected value for each play's outcome. Contrasting this measurement against actual quarterback outcomes has helped identify the most efficient quarterbacks in the NFL. Quarterbacks with the lowest opportunity cost in terms of expected value optimize with regard to decisions and are therefore the most efficient. Similarly, poor QBs will exhibit high opportunity cost with a large amount of expected value left on the field.

Survey

Simple metrics which are commonplace today, such as *Yards per Attempt* lack the ability to capture to context of an individual play. This is why current advanced metrics look to leverage that context with objective functions such as expected win-probability added (Goldner). The NFL themselves has also distributed many new metrics through their use of Next Gen Stats, however they have not leveraged this idea of completion probabilities to grade quarterbacks in terms of opportunity-costs and decision making. Other approaches (Burke) have attempted to quantify similar concepts of decision making, however they have lacked interactive tools which allow users to ask and answer questions relating to this analysis.

List of Innovations

- Given the recency of this NFL player-tracking data, this project is one of the first in research to approach quarterback decision-making.
- Machine learning efforts are sparse in football analysis and we chose the best-in-class model from a variety of approaches, allowing us to have the best accuracy and analysis of the questions we have attempted to answer

- We have created a tool which allows users to interact with our dataset and the results of our analysis. This tool can be leveraged by stakeholders in professional football and media alike. While research on similar topics (e.g. catch probability) has been completed in the past, they have been static presentations in written form. This interactive, dynamic tool will set our project apart from previous research in the field of professional football.
- The metrics we developed (especially Decision Making) will be first of its kind in professional football. With larger datasets they can be expanded upon and added to with the end goal of entering the regular lexicon of QB performance evaluation.

Method

As with any research project, there are several components which will be key to overall project success. We have broken these out into the following modules: feature engineering, model construction and evaluation, analysis, and presentation of results. Below are details about each section as well as information about it's impact on the outcome of our research.

Feature Engineering

The first step towards building a comprehensive model for research is obtaining data and preparing it for modeling and analysis. Our project is no different. From the 4,689 pass plays made publicly available by the NFL, we have engineered the following features for each eligible wide receiver at the time of the quarterback's decision. Most features were included in the dataset obtained by the NFL, so features which were further engineered include information about how that was accomplished:

- Distance from quarterback
- Intended receiver
- Distance to receiver of each of the two closest defenders
- Number of total defenders with a 5-yard radius
- Average defender distance
- Farthest defender distance
- Route Classes
    - We leveraged a k-means algorithm to assign each route to a `route class` based on k=5 and principal components analysis which allowed us to add a feature to the model leveraging higher-order analysis of the routes which were run by each receiver. *Figure 1* shows a visualization of each route class as it would appear on a football field

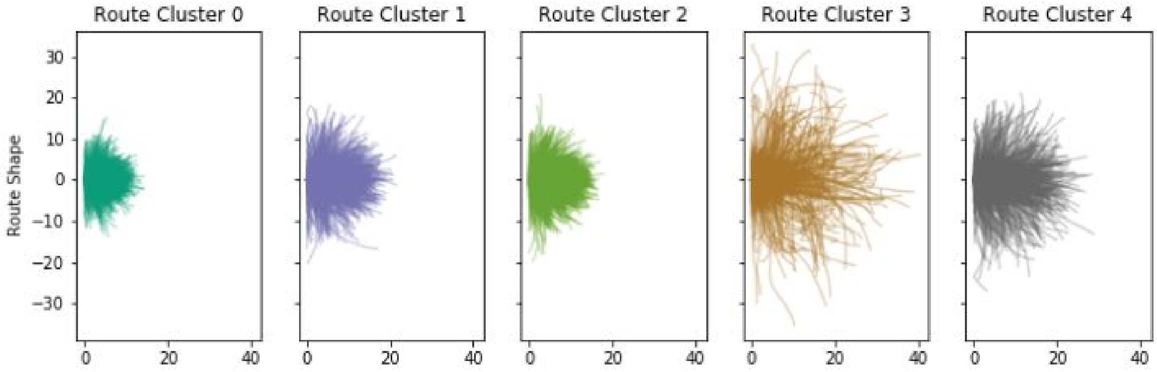


Figure 1: Clusters

- Play metadata
    - Down (1st, 2nd, 3rd, 4th)
    - Distance to First Down
    - Distance to End Zone
    - Score

We also engineered and tested several features which did not improve the out-of-sample accuracy of our models which are discussed below. Those features include:

- Indoor
- Temp
- Wind Speed
- Clutch: is it a clutch situation (boolean). Clutch is less than 5 minutes remaining in the game and less than one score difference (8 points)
- Number of Defensive Backs
- Number in the box: number of defenders in the box at play start
- Number of pass rushers
- Passing Situation: 3rd down and 8 or more yards to go
- Time remaining in game and quarter
- Def_QB1: how many defenders with 1 yard of qb
- Def_QB2: how many defenders with 2 yard of qb
- Def_QB3: how many defenders with 3 yard of qb
- QB Speed at time of release
- Time since snap
- QB Defender Distance
- Out of pocket
- Nearest Defender Speed
- Speed Diff: difference between nearest defenders speed and WR's speed
- WR speed at time of ball release

Given these constructed features we built four different models for comparison and evaluation.

Model Construction

Our team constructed each of the below models evaluate its effectiveness for calculating catch probabilities. For training, the models will learn based solely on the intended receiver. From the trained model we will generate predictions for every eligible receiver not thrown to in order to determine all catch probabilities in the dataset. Post construction, each of the models evaluated using cross-validation to compute an ROC AUC score. From this evaluation of each model, its accuracy and other potential trade-offs we will select our final "production" model. We leveraged random seed functionality from sklearn's train/test/split module to ensure the out of sample evaluation is consistent across models.

The models architectures which we evaluated are below, including information about how each model was tuned or otherwise manipulated to reach its final ROC AUC score.

- Hierarchical Logistic Regression
    - We explored several variations of feature selections, however after exploring all possible feature combinations, our logistic model only performed slightly better than a "random guess" with an AUC score 0.55
- Random Forest
    - First we implemented a Random Forest from sklearn's modules leveraging all of the default parameters, this resulted in an improved AUC score of 0.66
        - Immediately, this tells us there were likely some non-linear relationship between these features which the Logistic model did not pick up on
    - Second, we ran a gridsearchCV on the following parameters of the Random Forest:
        - N_estimators, max_depth, min_samples_leaf, max_features
    - This gridsearchCV produced and AUC score of 0.7225 and *Figure 2* contains the feature importances which were produced from this improved model

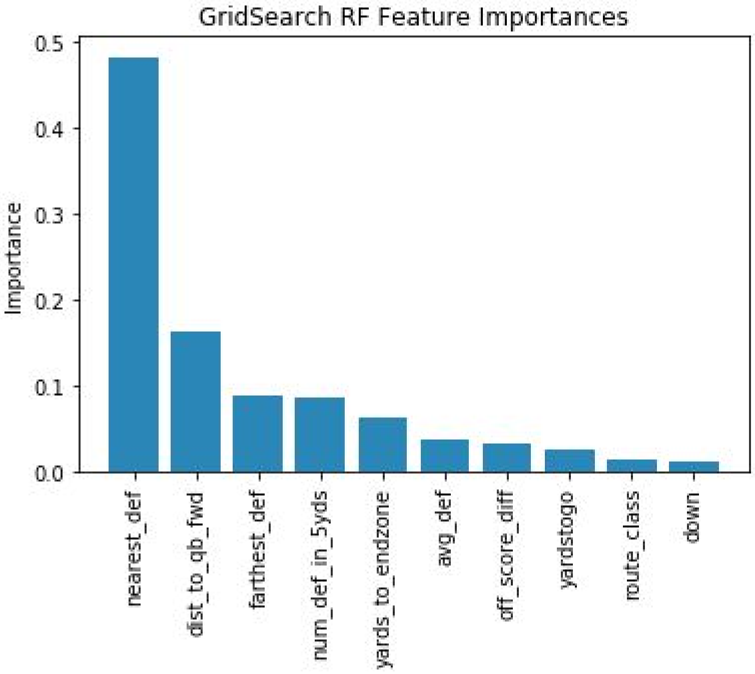
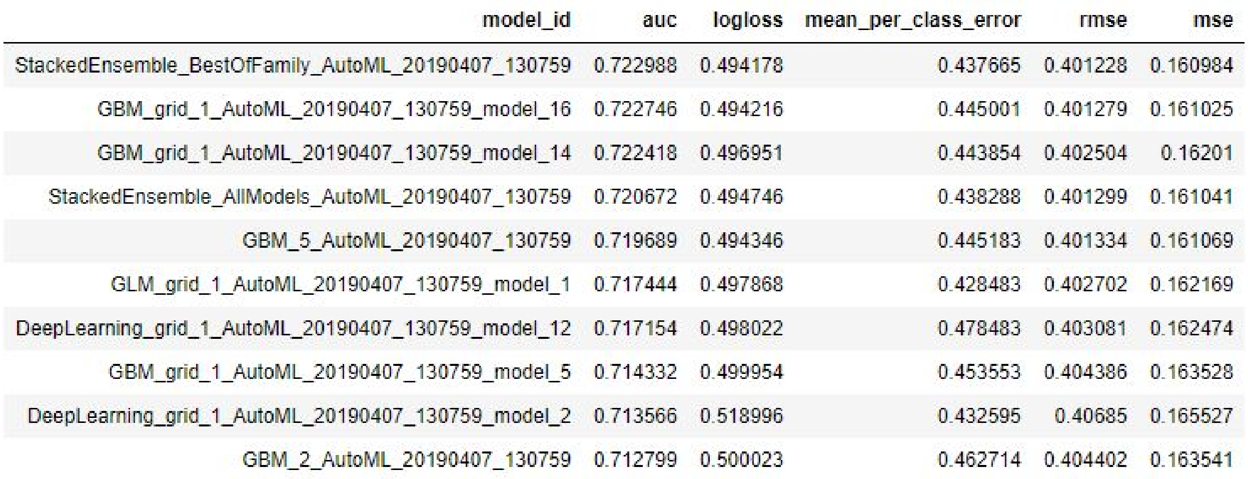


Figure 2: Gridsearched Random Forest, Feature Importances

- H20.ai AutoML
  - H20.ai is an open source machine learning platform which boasts a powerful engine (AutoML) which allows users to train a large number of candidate models and then generate an ensemble of those best-in-class models for a final model output. It has seen much success in both corporate settings and Kaggle competitions.
  - We provided AutoML the same features as our other models, and trained for a 2 hour time period. Below is the leaderboard of top models, and unsurprisingly the stacked ensemble model had the highest AUC score, as seen below in *Figure 3*



model_id	auc	logloss	mean_per_class_error	rmse	mse
StackedEnsemble_BestOfFamily_AutoML_20190407_130759	0.722988	0.494178	0.437665	0.401228	0.160984
GBM_grid_1_AutoML_20190407_130759_model_16	0.722746	0.494216	0.445001	0.401279	0.161025
GBM_grid_1_AutoML_20190407_130759_model_14	0.722418	0.496951	0.443854	0.402504	0.16201
StackedEnsemble_AllModels_AutoML_20190407_130759	0.720672	0.494746	0.438288	0.401299	0.161041
GBM_5_AutoML_20190407_130759	0.719689	0.494346	0.445183	0.401334	0.161069
GLM_grid_1_AutoML_20190407_130759_model_1	0.717444	0.497868	0.428483	0.402702	0.162169
DeepLearning_grid_1_AutoML_20190407_130759_model_12	0.717154	0.498022	0.478483	0.403081	0.162474
GBM_grid_1_AutoML_20190407_130759_model_5	0.714332	0.499954	0.453553	0.404386	0.163528
DeepLearning_grid_1_AutoML_20190407_130759_model_2	0.713566	0.518996	0.432595	0.40685	0.165527
GBM_2_AutoML_20190407_130759	0.712799	0.500023	0.462714	0.404402	0.163541

Figure 3: AutoML Leaderboard

- XGboost Gradient Boosted Trees
  - Our final model produced was an XG Boost model which optimized for the following parameters: colsample_bylevel, colsample_bytree, interval
  - The model had a 10-fold CV score of 0.7246
  - Below in Figure 4 is a sample of prediction explanation from the *shap* package which allows us to investigate model performance, we can see here that nearest_def and dist_to_qb are the two most important features, which is consistent with what we saw in the Random Forest

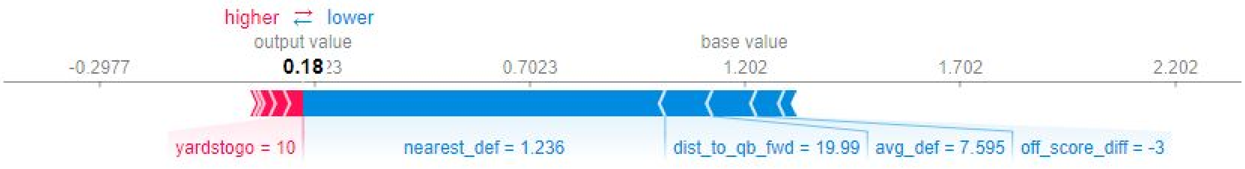


Figure 4: Shap Prediction explanation

Given the performance of the above models and the tradeoffs between training time, interpretability, and application, we have elected to produce our catch probabilities for analysis with our GridSearch Random Forest model, as it has very similar accuracy to the H2o model, but with much faster training time, and interpretability with its feature importances. *Figure 5* below shows a summary of training times and AUC scores of each model.

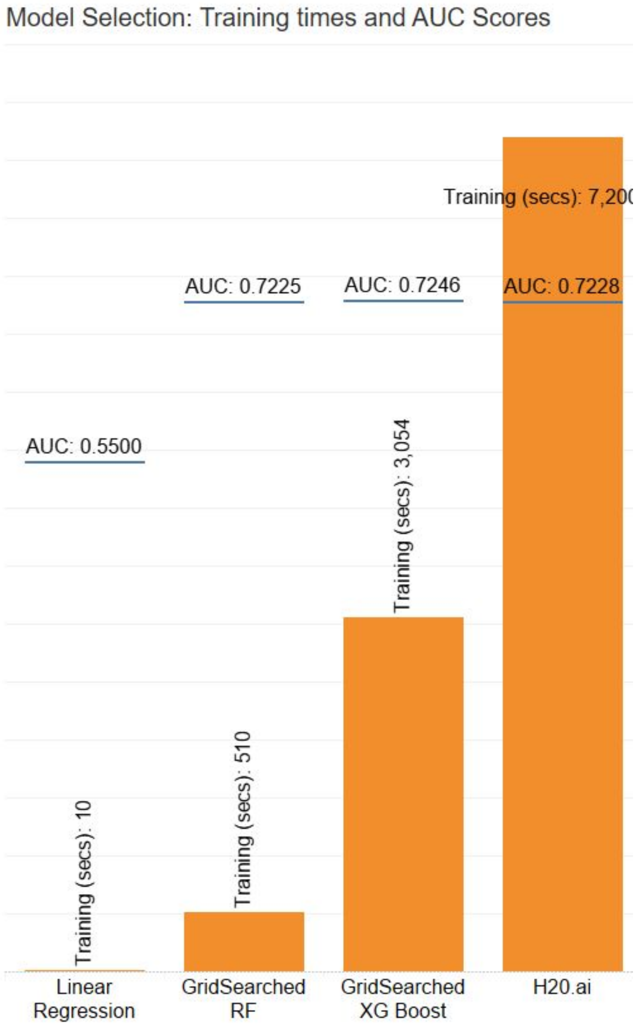
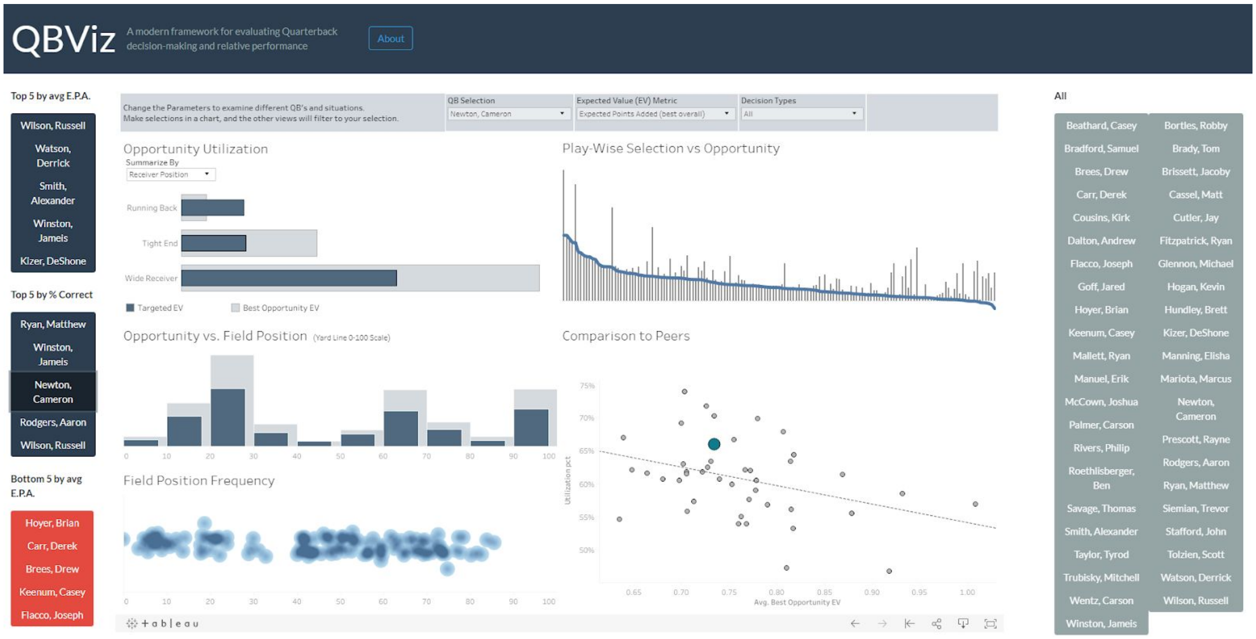


Figure 5: AUC Scores and Training Times

Presentation of Results

We have presented the results of this analysis in a web application which hosts a Tableau dashboard, along with results of our analysis and dynamic selectors. Allowing users to explore several aspects of our analysis. The web app allows users to see the resulting metrics for a given quarterback, but also explore the context of the quarterback's performance. We chose Tableau as a visualization tool as it easily allows users to select a Quarterback, examine their performance, and then "drill-down" into that players' performance within a certain segment. For example, a user can quickly see that Cam Newton overperformed expectations when targeting running backs, as compared to when targeting wide receivers. Figure 6 shows an example of how the user can see this.



Change the Parameters to examine different QB's and situations.
Make selections in a chart, and the other views will filter to your selection.

QB Selection: Newton, Cameron

Expected Value (EV) Metric: Expected Points Added (best overall)

Decision Types: All

Summarize By: Receiver Position

Figure 6: Examining Cam Newton's performance when targeting running backs.

The web app was constructed using html, javascript, and css. Functionality was introduced directly through interactive features in Tableau, as well as leveraging Tableau's javascript API. Other plugins used include jQuery and Bootstrap. By building features around the central visualization, users can quickly find key players and update the dashboard. For example, users can select the top and bottom 5 performers in each of the metrics (described below) via the quick access buttons on the left side of the page. We chose Github pages for hosting, because they allow for a lightweight framework for hosting web applications, minimizing development overhead for the developers.

The QBViz app is available here: https://realkvanderbush.github.io/QBViz/

Analysis and Experiments

Armed with the outputs of our model, and an interactive web application which allows users to interact with out analysis, we took the follow steps to develop metrics which enabled us to ask several different questions.

- Create an *expected yards added* metric for each potential receiver in each pass play

$$EYA = \{x \epsilon ER \mid (\pi * LOSyards) + NDyards\}$$

Where:

π = catch probability

ER = eligible receivers

LOSyards = yards from line of scrimmage

NDyards = yards to nearest defender

From here, we have the expected yards added by throwing to each receiver on a given play. However, as we know in football, not all yards are created equal. Based on the work by *Yurko, et al.* we know that plays which accrue the same yards, at different parts of the field can be valued very differently. As an extreme example, a 2-yard completion on a team's own 20 yard line on 3rd and 10 is worth far less then a 2-yard completion when an offense has a 3rd and 1, 10 yards away from the endzone. For this reason, we leveraged *Yurko, et al.* and produced an *Expected Points Added* metrics from our EYA values above which allowed us to properly value field position as it relates to quarterback decision-making.

$$EPA = EP(LoS+EYA) - EP(LoS)$$

Where:

EP(LoS) = Exp. Points given LoS as outlined in *Yurko, et al.*

EP(LoS + EYA) = Exp. Points given LoS + EYA for each receiver from above EYA

The difference in EP(LoS) and EP(LoS + EYA) is the incremental EPA for each eligible receiver.

Observations

Now, we are armed with sufficient information to perform an analysis on quarterback performance, as well as produce aforementioned dashboards for users to interact with visualizations in an exploratory manner.

- Rank QBs based on their EPA per play
    - For each eligible receiver in each play, calculate the EPA for that receiver
    - Rank QBs by their average score for this metric
    - Given our approach as outlined above, this should control for time, down, distance, etc of the game
    - Top 5 QBs by EPA are:
        - Russell Wilson
        - Derrick (Deshaun) Watson
        - Alex Smith
        - Jameis Winston
        - DeShone Kizer
    - Bottom 5 QBs by EPA are:
        - Brian Hoyer
        - Derek Carr
        - Drew Brees
        - Case Keenum
        - Joe Flacco
- Rank QBs based on their "decision making"
    - Which quarterbacks make the best decisions (in terms of EPA) the highest percentage of the time?
    - Top 5 Decision-Making QBs:
        - Matt Ryan
        - Jameis Winston
        - Cam Newton

        - Aaron Rodgers
        - Russell Wilson
    - Bottom 5 Decision-Making QBs:
        - Deshone Kizer
        - Joe Flacco
        - Matt Stafford (John)
        - Drew Brees
        - Philip Rivers

- Exploratory Data Analysis & Case Study

An additional feature of our research is our QBViz web application which allows users to ask and answer their own questions about QB performance and dig into certain positions, situations, etc based on their interest on an individual player. This additional feature sets out research apart from others in this similar field (which are usually just presented with static papers).

From these results we can see the variability in quarterback performance, relative to players' decision-making. For example, Kizer is among bottom 5 QBs in decision-making, yet among top 5 in EPA. This indicates that his relatively risky play has paid-off to date, but may not be sustainable moving forward.

We also see a player like Drew Brees, widely considered to be a top QB in the league, ranking in bottom 5 in both categories. In examining Brees in the dashboard, we see that his reliance on WRs and his risky play place him lower in these categories. A possible confounding variable here could be the overall talent of players like Michael Thomas and Alvin Kamara, both regarded as two of the best pass-catchers at their position in the NFL. As can be seen in *Figure 7* below, Kamara's targeted EV outweighs his Best Opp EV, and Thomas has one of the highest Targeted EV values for all WRs in the NFL.

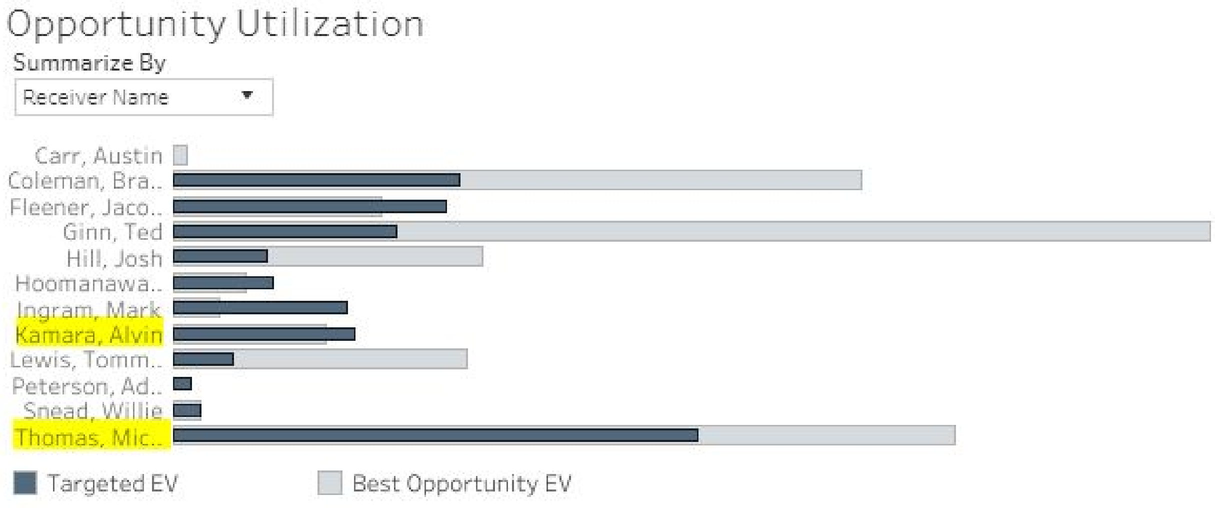


Figure 7: Examining Michael Thomas and Alvin Kamara

It is possible that our model may be pessimistic about opportunities to these receivers, but the WR talent level more than makes up for that pessimism. In future research, we will look to further incorporate receiver talent into the model in order to account for this phenomenon.

Future Development

QBViz provides a base for future analysis and development in the realm of quarterback decision making. Currently we have limited our analysis to the frame where the QB throws the ball, while it is possible that the timing of the throw is just as important as the receiver targeted. In the future we would like to apply our catch probability model and subsequent EPA to all frames in a single play. This would allow for analysis not only of the optimal receiver to target, but the optimal point to throw. Comparing the actual targeted receiver's EPA against the max EPA at any point in the play could further distinguish the decision making ability of quarterbacks.

Long-term, we may look to formalize this research and submit it to either the Journal of Quantitative Analysis in Sports or the Journal of Sports Analytics.

Conclusions and Discussion

In review of this project, we are extremely pleased with the goals we set out to accomplish and the subsequent results of this research. We leveraged the NFL's player tracking data and engineered features which were then used to develop a series of predictive models to predict catch probabilities of over 20K eligible receivers in our dataset. From there, we selected a gridsearchCV Random Forest model to produce a catch probability metric which then fed into a series of analyses on those data. We leveraged previous research from *Yurko, et al.* to transform those catch probabilities into EPA - a metric which controls for all factors of gameplay to objectively evaluate quarterback decision making and performance. We completed a series of experiments (outlined above) which revealed several interesting outcomes, such as:

- QB "risk-taking" can pay off as seen in DeShone Kizer, but questions whether or not their decision-making is sustainable in the long-term
- QBs who favor their RBs and TEs tend to perform well in both the EPA and Decision-Making metrics (Winston, Newton). This could be an indicator of a league-wide inefficiency with the current market value placed on WRs.
- Successful QBs such as Drew Brees may not grade out well in these metrics, given the above-expectation talent of their WRs and other pass-catchers

It should be noted that the work of the team was distributed evenly throughout the course of this project, across all phases.

Citations

Goldner, K. (2012). A Markov Model of Football: Using Stochastic Processes to Model a Football Drive. Journal of Quantitative Analysis in Sports, 8(1), pp. -. Retrieved 18 Apr. 2019, from doi:10.1515/1559-0410.1400

Burke, B. (2019). DeepQB: Deep Learning with Player Tracking to Quantify Quarterback Decision-Making & Performance. *Sloan Sports Analytics Conference*

Yurko, R., Ventura, S. & Horowitz, M. (2019). nflWAR: a reproducible method for offensive player evaluation in football. Journal of Quantitative Analysis in Sports, 0(0), pp. -. Retrieved 18 Apr. 2019, from doi:10.1515/jqas-2018-0010