MIT SLOAN
SPORTS ANALYTICS CONFERENCE
MARCH 3 - 4, 2017 HYNES CONVENTION CENTER

American Football Route Identification Using Supervised Machine Learning

Hochstedler, Jeremy & Paul T. Gagnon
Telemetry Sports

Abstract

Football organizations spend significant resources classifying on-field actions. Since 2014, radio-frequency identification (RFID) tracking technology has been used to continuously monitor the on-field locations of NFL players.[1] Using spatiotemporal American football data, this research provides a framework to autonomously classify receiver routes.

1. Introduction

NFL teams spend a significant amount of time each week breaking down film and tagging plays, which enables coaches, scouts, and players to filter plays for further analysis and more efficiently consume video. Traditionally, Quality Control (QC) coaches breakdown film by tagging events, players, actions, and play features. This activity isn't new, as Steve Belichick's *Football Scouting Methods* (1962) outlined many of the roles of a QC coach. However, when it comes to preparing for a game, accuracy (as it relates to getting the data that coaches want) and timing (allowing coaches more time to create a game plan) are key.

With the NFL's available player tracking data, automation and machine learning will enable teams to tag data more quickly and in a more uniform manner. Currently, teams rely on multiple (at times, not standardized) data collection mechanisms. For example, multiple coaches (or third parties) break down game film to tag data. Naturally, these existing methods are susceptible to human error and are time intensive.

2. Data

Since the NFL player tracking data (NextGen Stats) is not available for competitive scouting or public analysis, we have created a dataset using the 2014 regular season Indianapolis Colts base-offensive plays.

Spatial coordinates of all on-field players were captured based on the "All 22" game video at three frames per second to the nearest 0.25 yard for every 2014 base offensive pass play. This yields 231 total plays. Here, the "base offense" is defined as 1st & 2nd down, less than 15-point differential, greater than five minutes remaining in a half, and between the 20 yard lines. Fig. 1 displays the data collection process flow diagram for a given play.

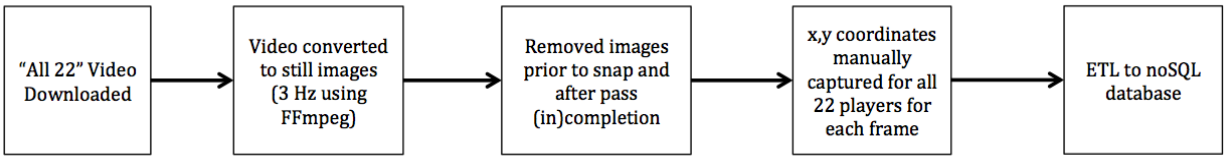


Figure 1. *Data collection process flow.*

From the 231 base offensive pass plays, trained coaches (used here to mimic the work of QC coaches) classified each receiver's route using a simplified route tree as described in Fig. 2. Each route is labeled using a primary tag with secondary labels used to notify the reader that route subtypes are also classified under the primary tag. An industry standard numbering system is also included. The machine learning classifier will use the primary tags associated with each route, therefore we will refer to each of the eleven route types using the primary tag going forward.

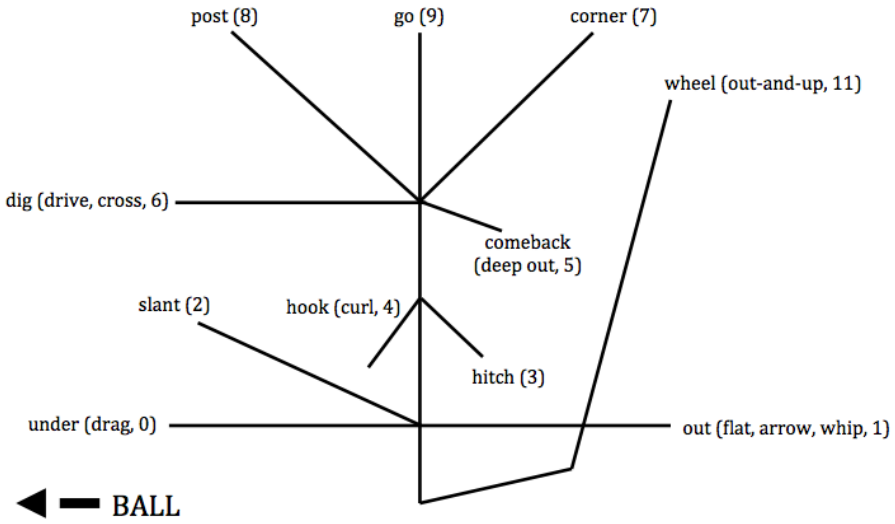


Figure 2. *Simplified route tree.*

After removing screen plays (as most receivers typically block on these plays), we analyzed the routes of all players who lined up as slot receivers or wide receivers (see formation identification), leaving 548 player routes. Table 1 displays the eleven route types, relevant secondary labels, and the number of each route type in the data set.

Route No.	Primary Label	Secondary Labels	Count
0	under	drag, shallow cross	31
1	out	flat, arrow, whip, quick out	70
2	slant	--	30
3	hitch	--	15
4	hook	curl, stop	44
5	comeback	deep out	22

Route No.	Primary Label	Secondary Labels	Count
6	dig	drive, cross	123
7	corner	flag	23
8	post	bang	50
9	go	fly	121
11	wheel	release, out-and-up	19

Table 1. *Route types and associated counts within dataset.*

Route label decisions were made as the coaches interviewed focus more on the route concept as it relates to the play call, not necessarily the specific route run. For example, some routes can be "option" routes as variants will be run based off of defensive alignment and scheme. Specifically, an "out" route can be modified to an "arrow" route if the defender is playing off coverage. In another example, an option "go/9" route can turn into a "post/8" route if the middle of the field is left open (MOFO). Fig. 3 displays the trajectories of all 548 routes, specifically displaying "under" routes in red.

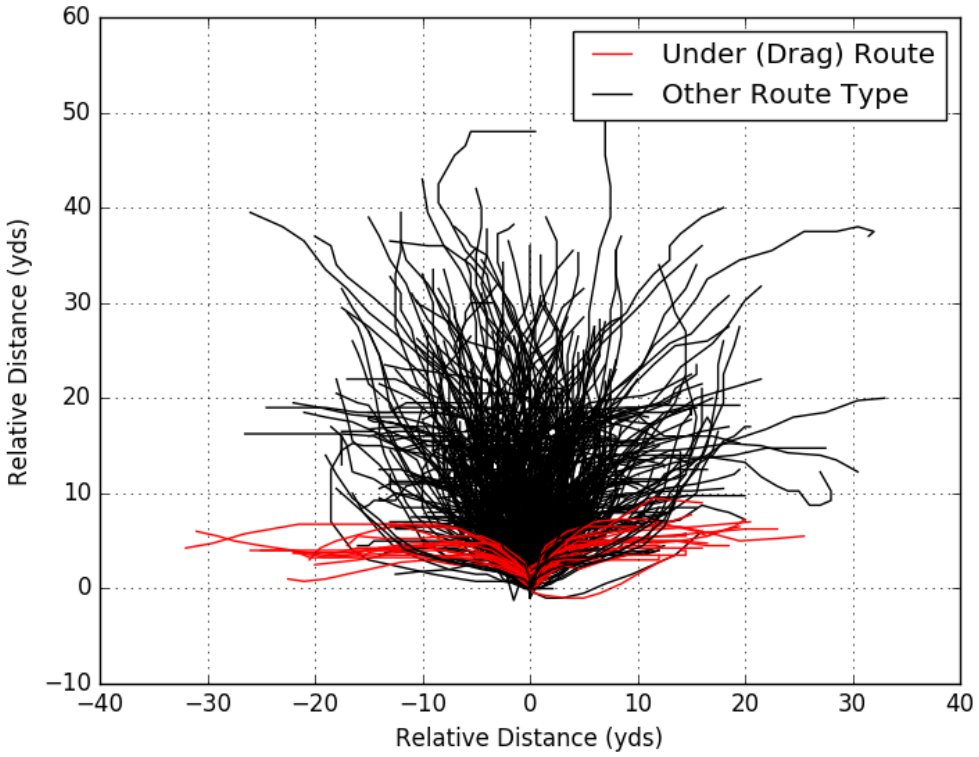


Figure 3. *548 receiver trajectories from the 2014 Indianapolis base offense.*

3. Formation Identification

Individual player starting locations are defined by proximity to teammates, opponents, various fixed field points (e.g. hashes, boundaries, and yard markers), as well as floating points (e.g. line of scrimmage). Logic has been established to classify player alignment as one of ten distinct positions (LT, LG, C, RG, RT, TE, SR, WR, RB, and QB). Additionally, further information is tagged for each individual player. For example, players are identified whether they are lined up on the line of scrimmage or not, in a bunch/stacked formation, their associated skill number (e.g. L1, L2, R1, R2), and more. We will keep to an abbreviated description of the formation and player alignment methods as the focus of this research is to outline a framework for route classification.

4. Route Classification

Upon the start of play execution (the snap), player trajectories are analyzed for every position on the field in order to classify the player action. Because routes are run in relation to ball position, all routes starting from the formation's left side have been mirrored to create x-coordinates as-if they were run from the right side of the formation.

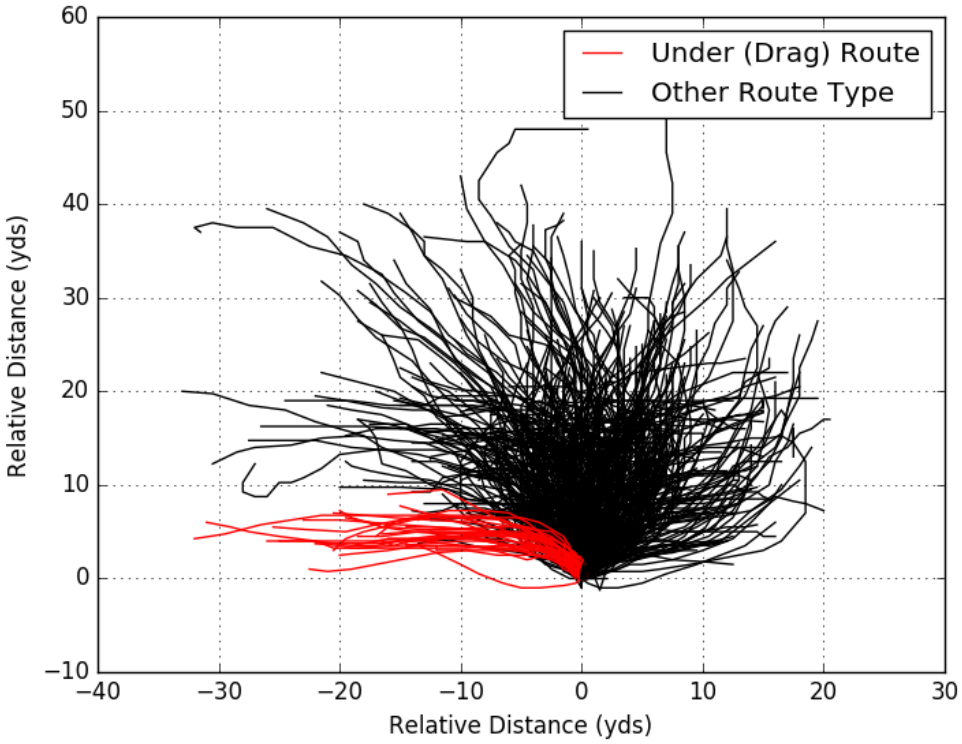


Figure 4. *548 receiver trajectories with skill left paths mirrored to skill right.*

4.1 Feature Extraction

The key to any machine learning algorithm is the selection/extraction of features to train the model. The slot/wide receiver routes are captured until either: a maximum of 15 frames/5 seconds (3 Hz dataset) or forward pass takes place.

Basic receiver routes are comprised of three essential components: a stem, pivot, and branch. The stem is the initial segment from starting point (snap, t = 0) to the pivot (re-direction) point. The branch is the final component of the route for which the receiver runs. Using a segmented Euclidean regression, features are distilled from each receiver's trajectory. Fig. 5 highlights the three basic route components.

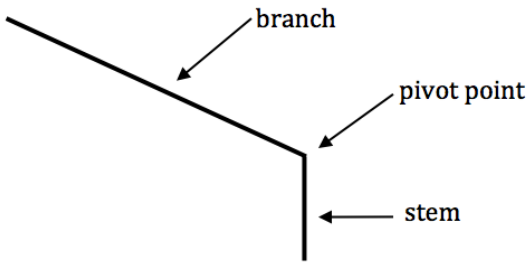


Figure 5. *A typical slant route.*

4.1.1 Segmented Euclidean Regression

While each route possesses unique features such as stem heading and branch length, the most significant features distilled for classification are its stem length and branch heading. These features are used in the training of supervised machine-learning algorithms.

As a first pass in determining the two-segment linear regression, we first use the start point $[x_0,y_0]$ and the route's end point $[x_n,y_n]$ testing each internal node as the pivot point $[x_p,y_p]$. Straight lines are then drawn from $[x_0,y_0]$ to $[x_p,y_p]$ and again from $[x_p,y_p]$ to $[x_n,y_n]$ for each internal data point. The Euclidean distance (error) for all data points is then summed. The point that yields the minimized Euclidean error is denoted as the route's true pivot point.

$$\text{minimize } f(\text{p}) = \sum_{i=0}^{n} \sqrt{\left(x_{ri}^2 - x_{ti}^2\right)^2 + \left(y_{ri}^2 - y_{ti}^2\right)^2}$$

subject to: 0 < p < n

where:

r = regression path
t = true path
p = pivot point

In order to more accurately represent the trajectory data with a three-point path (the regression path), additional pivot points near the route's true pivot point are tested. These points are found by uniformly spreading hypothetical pivot points near the original path's pivot point. Fig. 6 displays an additional 25 (5x5, edges identified by blue arrows) pivot points surrounding the originally minimized pivot point (circled in blue). Note: the spread is increased for illustration purposes. Our algorithm optimizes based on a spread with an order of magnitude smaller and order of magnitude more test points.

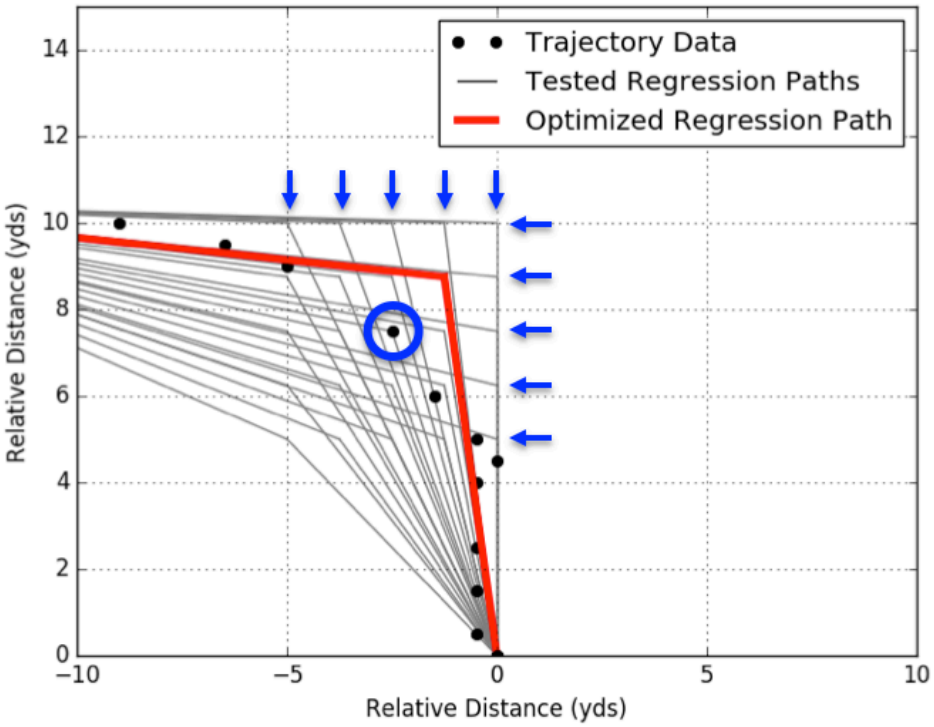
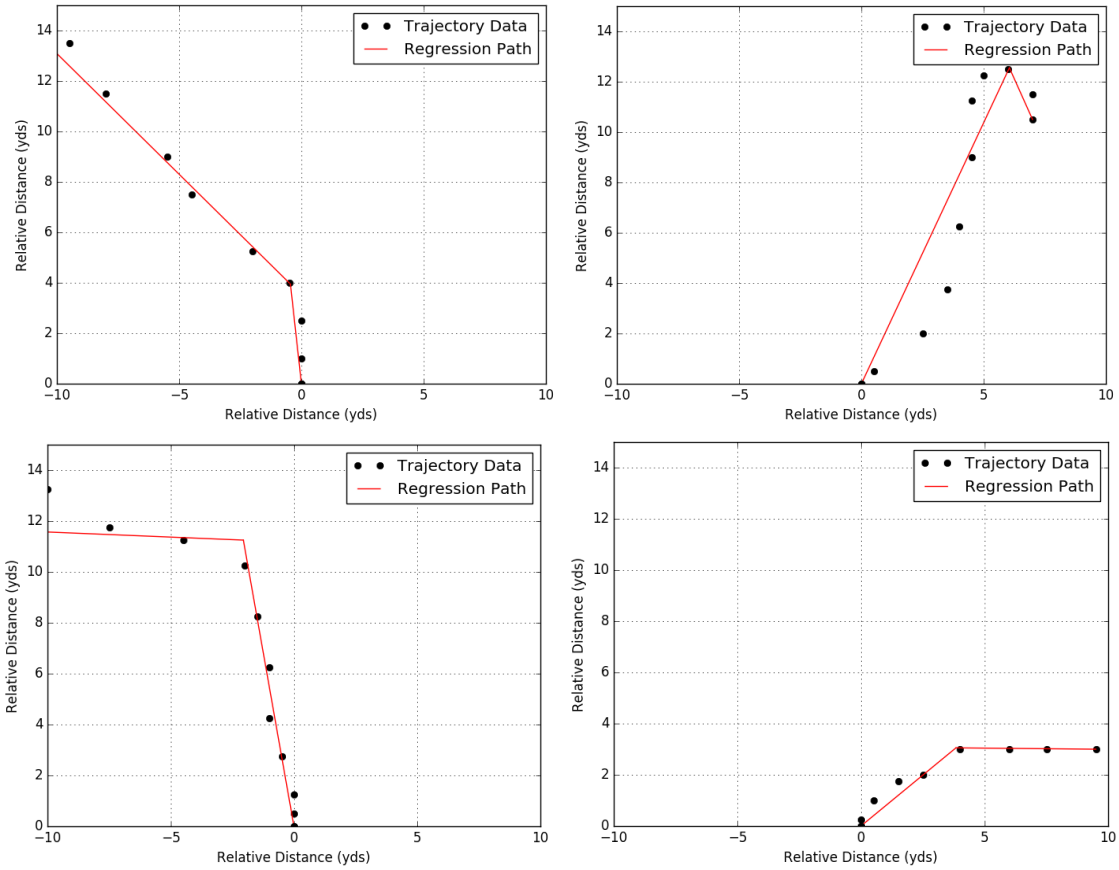


Figure 6. *Illustration of tested regression paths to minimize Euclidean error.*

Fig. 7A-D and Table 2 provide additional examples of how the two-segment Euclidean regression produces stem length and branch heading values for some sample routes.



Figures 7A-D. *Clockwise from top left: slant, comeback, out, and dig routes.*

Route	Stem Length (yds)	Branch Heading (deg. from East)
slant	4.0	136°
comeback	13.9	-63°
out	4.8	0°
dig	10.9	172°

Table 2. *Stem length and branch headings for four routes shown above.*

4.1.2 Average Speed

Average speed throughout the route is calculated using a simple average over the duration of the player's route.

$$\bar{v} = \frac{\sqrt{x_n^2 + y_n^2}}{3f}$$

where:

f = total number of frames
x_n, y_n = final position coordinates

4.1.3 Final Route Location

The two final features used for training include the final position coordinates (x_n, y_n) relative to the player's initial position at snap.

4.2 Model Training

A 1:1 train-to-test ratio was used. Thus, the 548 player routes were split in half, allowing for 274 routes for both training and testing. The training dataset and each route's five features were used to train the classifiers. Specifically, five different supervised learning algorithms were trained and tested: Naive Bayes, k-Nearest Neighbors, Random Forest, Logistic Regression, and Support Vector Machines (SVM).

5. Results

Precision and recall are used to determine overall model accuracy. *Precision* is a measure of the ratio of accurately predicted classifications to the total number predicted for that classification category. *Recall* is a measure of the ratio of accurately predicted classifications to the total number of true classifications for that category.

$$Precision = \frac{t_p}{t_p + f_p}$$

$$Recall = \frac{t_p}{t_p + f_n}$$

where:

t_p = true positives
f_p = false positives
f_n = false negatives

Fig. 8 displays a normalized confusion matrix (of the Naïve Bayes model) to visualize accuracy while Table 3 provides further quantitative details.

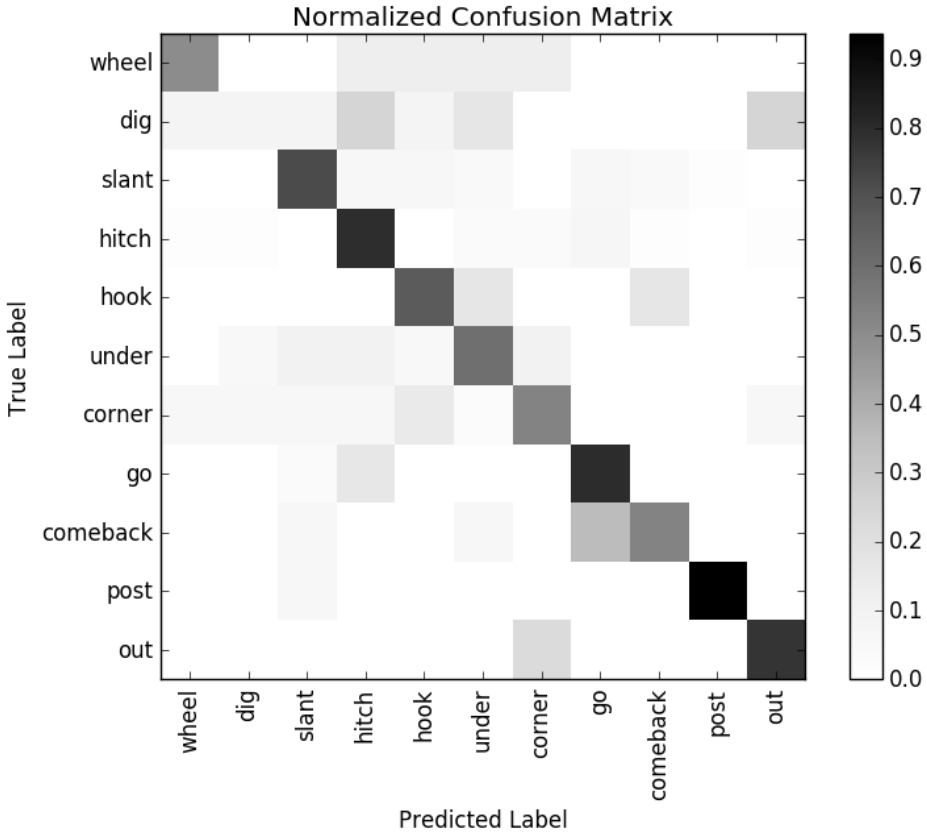


Figure 8. *Normalized confusion matrix of a Naïve Bayes model.*

Route	Precision	Recall	Count
comeback	0.36	0.45	11
corner	0.33	0.08	13
dig	0.85	0.65	68
go	0.80	0.71	68
hitch	0.25	0.57	7
hook	0.56	0.50	18
out	0.71	0.77	31
post	0.59	0.79	28
slant	0.65	1.00	11
under	0.83	0.83	12
wheel	0.54	1.00	7
total / avg	**0.70**	**0.68**	**274**

Table 3. *Precision details displaying an overall precision of 0.70.*

The maximum recorded precision and recall for each of the five learning algorithms is outlined in Table 4.

Classifier	Precision	Recall
Naïve Bayes	0.70	0.68
kNN	0.68	0.68
Random Forest	0.66	0.64
Logistic Regression	0.57	0.62
SVM	0.45	0.29

Table 4. *Maximum total precision and recall for each classification method.*

6. Conclusion

6.1 Summary

While this framework shows that the Naïve Bayes classifier provides the most accurate (yet mediocre) solution, it alone is not the main takeaway from this analysis. Instead, the primary focus shall remain in the *process* outlined, which includes feature selection and extraction.

The classification process outlined here will significantly improve the efficiency of NFL QC coaches. Additionally, these (and other future) methods will enhance game plan development by more accurately and more quickly identifying opponent tendencies, keys, and strategies. These methods do not replace the need for QC coaches; rather they enable the entire coaching staff to dive deeper into the data, tendencies, and video. By enabling them to focus their time and efforts on areas that are less mundane and providing these tags earlier each week, coaches can spend more time using the data and less time generating it.

6.2 Limitations

Because the number of training samples used here are relatively small, we expect increased accuracy with more training data. Because the RFID data possesses significantly better resolution (10 Hz vs. 3 Hz here), we also expect an accuracy increase. As is true with supervised learning applications, models improve with additional training data. In this case the manual data creation limits the training set and the results. Once the NextGen Stats data is available to teams for competitive scouting purposes, the value and accuracy of these methods will grow.

6.3 Future work

With the current data set, we plan to further investigate how route combinations contribute to play outcomes. Additional work can be done using synthesized route datasets to train the machine learning algorithm. As NextGen Stats becomes available, we plan to work with teams to focus automation and auto-classification efforts on QC coach activities that are mundane and error prone, in addition to areas where it is difficult for QC coaches to "see" the underlying data based on video alone.

References

[1] Zebra Technologies. Accessed December 10, 2015. https://www.zebra.com/us/en/nfl.html

[2] Belichick, Steve. *Football Scouting Methods.* 1962. Martino Publishing. Mansfield Centre, CT.