

A Naive Bayes Approach for NFL Passing Evaluation using Tracking Data Extracted from Images

Sarah Mallepalle¹, Ron Yurko¹,
Konstantinos Pelechrinis², Samuel L. Ventura¹

¹Carnegie Mellon University, Department of Statistics & Data Science

²University of Pittsburgh, Department of Informatics and Networked Systems

June 11, 2019

Abstract

The NFL collects detailed tracking data capturing the location of all players and the ball during each play. Although the raw form of this data is not publicly available, the NFL releases a set of aggregated statistics via their Next Gen Stats (NGS) platform. They also provide charts that visualize the locations of pass attempts for players throughout a game, encoding their outcome (complete, incomplete, interception, or touchdown). Our work aims to partially close the gap between what data is available privately (to NFL teams) and publicly, and our contribution is twofold. First, we introduce an image processing tool designed specifically for extracting the raw data from the NGS pass chart images. We extract the outcome of the pass, the on-field location, and other metadata. Second, we analyze the resulting dataset and examine NFL passing tendencies and the spatial performance of individual quarterbacks and defenses. We introduce a generalized additive model for completion percentages by field location. We use a Naive Bayes approach for adjusting the 2-D completion percentage surfaces of individual teams and quarterbacks based on the number of their pass attempts. We find that our pass location data matches the NFL’s official ball tracking data.

1 Introduction

Player tracking data captures the position and trajectory of all athletes and objects of interest (e.g. balls, pucks, etc) on the playing surface for a given sport. The importance of this data in analyzing the performances and strategies of players and teams has risen dramatically over the past decade, as organizations look to gain an edge over their opponents in ways that were previously not possible. Publicly, analysis of player tracking data across the four major spots has also increased, but is limited by the availability of such datasets to the public.

Our work aims to bridge the gap between public and private data availability, and to provide an analysis of individual and team passing tendencies in the National Football League (NFL).

1.1 Player Tracking Data in Professional Sports

Major League Baseball (MLB) has been tracking pitch trajectory, location, and speed since 2006 with PITCHf/x (Fast, 2010). In 2015, MLB launched Statcast, which additionally tracks the exit velocity and launch angle of a batted ball along with location and movements of every player during a game (Casella, 2015). The National Basketball Association (NBA) mandated the installation of an optical tracking system in all stadiums in the 2013-14 season (NBA, 2013). This system captures the location of all the players on the court and the ball at a rate of 25 times per second (25 Hz). This data is further annotated with other information, such as event tracking (“play-by-play”), current score, shot clock, time remaining, etc. The National Hockey League (NHL) plans to begin the league-wide use of player- and puck-tracking technology in the 2019-20 season (Wyshynski, 2019). The NFL installed a player tracking system in all of its venues during the 2015 season (NFL, 2019). NFL’s tracking system is RFID-based and records the location of the players and the ball at a frequency of 12.5 Hz.

This type of data have spurred innovation by driving a variety of applications. For example, Cervone et al. (2016a) computed the basketball court’s Voronoi diagram based on the players’ locations and formalized an optimization problem that provides court realty values for different areas of the court. This further allowed the authors to develop new metrics for quantifying the spacing and the positioning of a lineup/team. As another example, *ghosting* models have been developed in basketball and soccer when tracking data is available (Le et al., 2017; Lowe, 2013). The objective of these models is to analyze the players’ movements and identify the optimal locations for the defenders, and consequently evaluate their defensive performance. Other models driven by player spatio-temporal data track the possible outcomes of a possession as it is executed, allowing to evaluate a variety of (offensive and defensive) actions that can contribute to scoring but are not captured in traditional boxscore statistics (Cervone et al., 2016b; Fernández et al., 2019), while Seidl et al. (2018) further used optical tracking data to learn how a defense is likely to react to a specific offensive set in basketball using reinforcement learning. Recently, Burke (2019) developed a model using player tracking data from the NFL to predict the targeted receiver, pass outcome and gained yards. Coming to soccer, Power et al. (2017) define and use a supervised learning approach for the risk and reward of a specific pass that can further quantify offensive and defensive skills of players and teams. For a complete review of sports research with player tracking data, see (Gudmundsson and Horton, 2017).

However, one common trend for player tracking datasets across all sports leagues is its limited availability to the public. For example, there are very limited samples of player tracking data for the NBA, and those that exist are

mostly from the early days of their player-tracking systems¹. Additionally, while pitch-level data is publicly available from the MLB system², the Statcast player and ball location data is not.

1.2 Player Tracking Data in the NFL

In December 2018, the NFL became the first professional sports league to publicly release a substantial subset of its player-tracking data of entire games for its most recent completed season, for two league-run data analysis competitions: (1) the NFL Punt Analytics competition³, and (2) the Big Data Bowl⁴. In total, tracking data from the punt plays from the 2016 and 2017 seasons, as well as all tracking data from the first six weeks of the 2017 regular season were temporarily released to the public for the purposes of these competitions. The Big Data Bowl player tracking data was removed shortly after the completions of this competition.

While this is a tremendous step forward for quantitative football research, the limited scope of the released data limits the conclusions that analysts can draw about player and team performances from the most recent NFL season. The only metrics available to fans and analysts are those provided by the league itself through their Next Gen Stats (NGS) online platform. The NFL’s NGS group uses the league’s tracking data to develop new metrics and present aggregate statistics to the fans. For example, the NGS website presents metrics such as time-to-throw for a QB, completion probability for a pass, passing location charts, and other metrics. However, NGS only provides summaries publicly, while the raw data is not available to analysts or fans. Additionally, any metrics derived from the player tracking data temporarily made available via the NFL’s Big Data Bowl is limited in scope and potentially outdated, as this data only covers the first six weeks of the 2017 season.⁵

1.3 Our Contributions

The objective of our work is twofold:

1. We create open-source image processing software, `next-gen-scrappy`, designed specifically for extracting the underlying data (on-field location of pass attempt relative to the line of scrimmage, pass outcome, and other metadata) from the NGS pass chart images for regular season and postseason pass attempts from the 2017 and 2018 NFL seasons.
2. We analyze the resulting dataset, obtaining a detailed view of league-wide passing tendencies and the spatial performance of individual quarterbacks and defensive units. We use a generalized additive model for modeling

¹<https://github.com/linouk23/NBA-Player-Movements>

²<http://gd2.mlb.com/components/game/mlb>

³<https://www.kaggle.com/c/NFL-Punt-Analytics-Competition>

⁴<https://operations.nfl.com/the-game/big-data-bowl>

⁵<https://twitter.com/StatsbyLopez/status/1133729878933725184>

quarterback completion percentages by location, and an empirical Bayesian approach for estimating the 2-D completion percentage surfaces of individual teams and quarterbacks.

Our work follows in the footsteps of `openWAR` (Baumer et al., 2015), `pitchRx` (Sievert, 2015), `nflscrapR` (Horowitz et al., 2017), `nhlscrapR` (Thomas and Ventura, 2013), `Lahman` (Friendly et al., 2019), `ballr` (Elmore and DeWitt, 2017), and `ncaahoopR` (Benz, 2019), who each promote reproducible sports research by providing open-source software for the collection and processing of data in sports.

With `next-gen-scraPy`, we rely on a variety of image processing and unsupervised learning techniques. The input to `next-gen-scraPy` is a pass chart obtained from NFL’s NGS, similar to the example in Figure 2 (which we detail in the following sections). The output includes the (x, y) coordinates (relative to the line of scrimmage) for the endpoint of each pass (e.g. the point at which the ball is caught or hits the ground) present in the input image, as well as additional metadata such as the game, the opponent, the result of each pass, etc. We then process these data and build a variety of models for evaluating passing performance. In particular, we develop spatial models for the target location of the passes at the league, team defense, and individual quarterback (QB) levels. We use generalized additive models (GAMs) and a 2-D empirical Bayesian approach to estimate completion percentage surfaces (i.e., smoothed surfaces that capture the completion percentage expected in a given location) for individual QBs, team offenses, and team defenses.

The rest of this paper is organized as follows: Section 2 describes the data collected and the processing performed by `next-gen-scraPy`. Section 3 presents the methods used to analyze the raw pass data obtained from `next-gen-scraPy`. Section 4 demonstrates the accuracy of the image processing procedure and presents the results of our analyses of individual QBs and team defenses. Finally, Section 5 concludes our study, presenting future directions and current limitations of `next-gen-scraPy`.

2 Image Processing Methods for Pass Location Data Extraction

In this section, we describe the raw data obtained from NGS, and the processing performed by `next-gen-scraPy`. We further present the output data provided by our software.

2.1 Collecting the Raw Image Data

The NFL provides passing charts via their online NGS platform, in the form of JPEG images. Each chart displays the on-field locations (relative to the line of scrimmage) of every pass thrown by a single quarterback in a single game. The points, which represent the ending location of each pass (e.g. the point

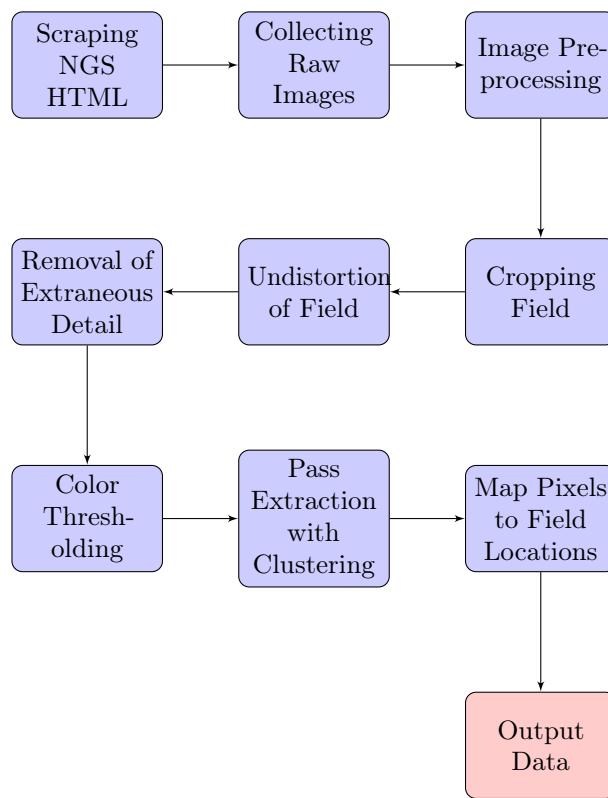


Figure 1: Flow chart of the `next-gen-scrappy` system

at which a ball is caught or hits the ground) are colored by the outcome of the pass (completion, incompletion, interception, or touchdown). Each chart is accompanied by a JavaScript Object Notation (JSON) data structure that details metadata about the quarterback or game represented in that chart. We link each passing chart to the data provided by `nflscrapR` to obtain additional metadata for each game (Horowitz et al., 2017). These charts are available for most games from the 2017 and 2018 NFL regular and post-seasons, though some are missing from the NGS website without explanation. An example passing chart is provided in Figure 2.

In total, there are 402 pass charts for 248 games throughout the 2017 regular and postseason, and 438 pass charts for 253 games throughout the 2018 regular and postseason. Each pass chart image is of 1200×1200 pixels, and is annotated with metadata that includes information such as player name, team, number of total pass attempts, etc. Appendix A provides a detailed list of the metadata necessary to perform data collection and pass detection.

In our analysis, we do not include data from the 2016 season (the first season in which NGS provided these charts), since approximately 65% of the pass charts are missing. Furthermore, the missing charts are not uniformly distributed across teams, biasing potential analyses. Finally, the existing 2016 charts frequently do not have the necessary metadata, introducing additional challenges to the data extraction process that we outline below.

2.2 Image Pre-Processing

Before being able to extract the raw data from the pass charts, the images must be pre-processed. First, the field in the raw image is presented as a trapezoid, rather than a rectangle, which would distort the pass locations extracted from the chart. Second, the images include unnecessary information that needs to be eliminated in order to allow for the streamlined and accurate processing of pass locations.

Removing Distortion. The football field and touchdown pass trajectories as shown in Figure 2 are distorted, because the trapezoidal projection of the field results in a warped representation of the on-field space, such that (for example) a uniform window of pixels contains more square yards towards the top of the image as compared to the bottom of the image. To fix this, we start by cropping each image to contain only the field, while at the same time we extend the sidelines by 10 yards behind the line of scrimmage, as consistent with every pass chart given by NGS. We then project the trapezoidal plane into a rectangular plane that accurately represents the geometric space of a football field. In particular, we calculate the homography matrix \mathbf{H} , such that every point between each point on the trapezoidal plane (x_t, y_t) is mapped to the rectangular plane (x_r, y_r) through the following equation:

$$\begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix},$$

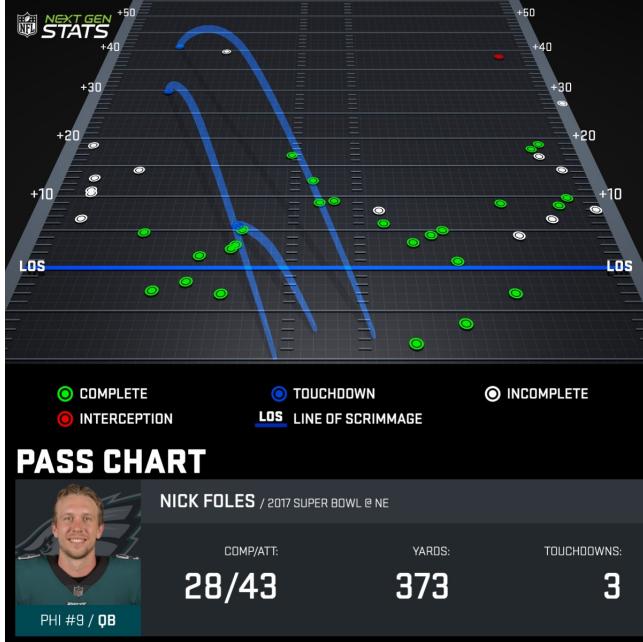


Figure 2: Nick Foles’ Pass Chart from Super Bowl LII - Extra-large (1200×1200) image extracted from the HTML of the Next Gen Stats website, which visualizes the location relative to the line of scrimmage of all complete, incomplete, touchdown, and intercepted passes.

where h_{00}, \dots, h_{21} are the elements of the homography matrix, \mathbf{H} , to obtain the relation between all initial (x_t, y_t) to the corresponding resulting (x_r, y_r) coordinates (Szeliski, 2010). This results in a fully rectangular birds-eye view of the football field, depicting uniform yardage across the height and width of the field.

Eliminating Unnecessary Details from Image. Next, we remove the white sideline numbers of the newly projected field. Each pass chart depicts 10 yards behind the line of scrimmage, and also either 55 or 75 yards beyond the line of scrimmage. Based on the pixel height of the newly transformed rectangular field image, we deduce whether or not the field depicts 55 yards (6 sideline markings) or 75 yards (8 sideline markings), and thus find the approximate locations of the white pixels. We then replace the white pixels with the same shade of grey as the sidelines. Removal of the white sideline numbers allows us to easily use a simple white color threshold to extract the incomplete pass locations, as described in Section 2.3.1. Based on the pixel height and width of the newly transformed rectangular field image, we can calculate the locations of the sideline markings by pixel ratios, and can locate the positions of the sideline markings to remove. The final image of the unwarped, clean, field from which the raw locations of the passes are extracted is shown in Figure 3(a).

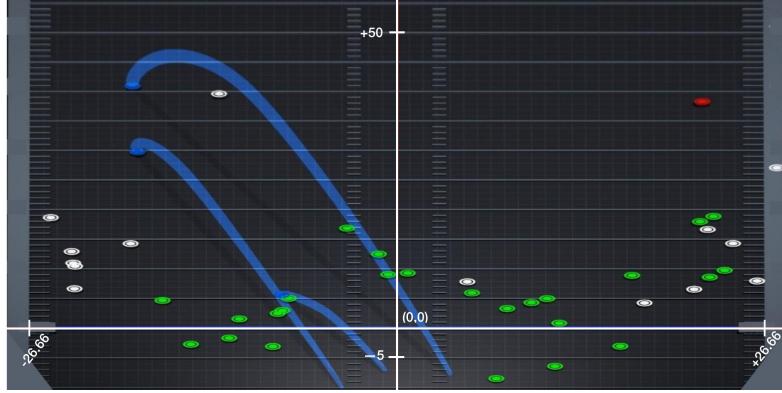


Figure 3: Nick Foles’ Pass Chart from Super Bowl LII after fixing the warped perspective of the field and removing the sidelines. The axes are displayed to show the x-axis falling directly on the line of scrimmage and the y-axis dividing the field vertically in half down the middle.

2.3 Clustering Methods to Extract Pass Locations

Every pass chart shows the locations of four different types of passes, relative to the line of scrimmage: completions (green), incompletions (white), touchdowns (blue), and interceptions (red). The JSON metadata includes the number of each pass type regardless of whether the pass is being depicted on the image or not⁶. There are a number of technical challenges to overcome when extracting pass locations:

- two or more pass locations can overlap,
- the number of passes shown on the image does not always match the number of passes given in the JSON metadata of the image,
- for touchdown passes, the color of the pass location is the same color as the line of scrimmage and pass trajectory path, rendering color thresholding used for the other types of passes ineffective.

To address these issues, `next-gen-scrappy` combines density-based and distance-based clustering methods (DBSCAN and K -means, respectively) with basic image processing techniques to overcome the aforementioned challenges and extract all the pass locations presented on an NGS pass chart.

2.3.1 Image Segmentation by Pass Type

All four different pass types are marked on the pass chart with different colors. Therefore, we examine the Hue-Saturation-Value (HSV) pixel coordinates from

⁶For example, passes that were thrown out-of-bounds are not depicted in pass charts.

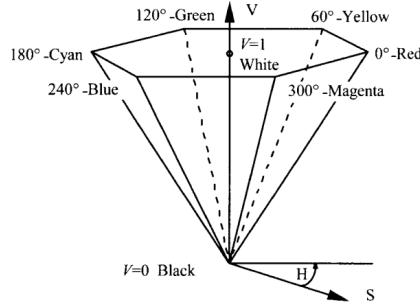


Figure 4: HSV color space representation

Pass Type	Lower Threshold	Upper Threshold
Complete	(80,100,100)	(160,255,255)
Incomplete	(0,0,90)	(0,0,100)
Touchdown	(220,40,40)	(260,100,100)
Interception	(0, 60, 60)	(20,100,100)

Table 1: HSV color thresholds for every pass type

the image to identify parts of the image that fall within a specified HSV color range. In brief, hue characterizes the dominant color contained in the pixel. It is captured by the angular position on the color wheel, with red being the reference color (i.e., $H = 0$ and $H = 360$). Complementary colors are located across of each other on the color wheel and hence, they are 180 degrees apart. Saturation measures the color purity, and is captured by the distance of the color from the center of the color wheel. Value characterizes the brightness of the color (Koschan and Abidi, 2008). This color system is visualized in Figure 4.

Based on the color of the pass type we want to detect, we use a basic thresholding technique to obtain images where all pixels are black except the ones with the pass locations. For example, for completed passes. we will keep the value of pixels within the respective range presented in Table 1 unchanged, and set the value of all other pixels to 0 corresponding to the black color.

The resulting segmented image is shown in Figure 5(a), along with the images obtained for the other pass types. As aforementioned in the technical challenges, for the touchdown passes the color thresholding includes additional noise in the final image, as shown in Figure 5(d).

From the JSON metadata associated with the image, we know the number of pass attempts, n_a , touchdowns, n_{td} , total completions, n_{tot-c} , and interceptions, n_{int} . The number of touchdowns and interceptions on each image is then simply n_{td} and n_{int} , respectively. The number of green non-touchdown completions

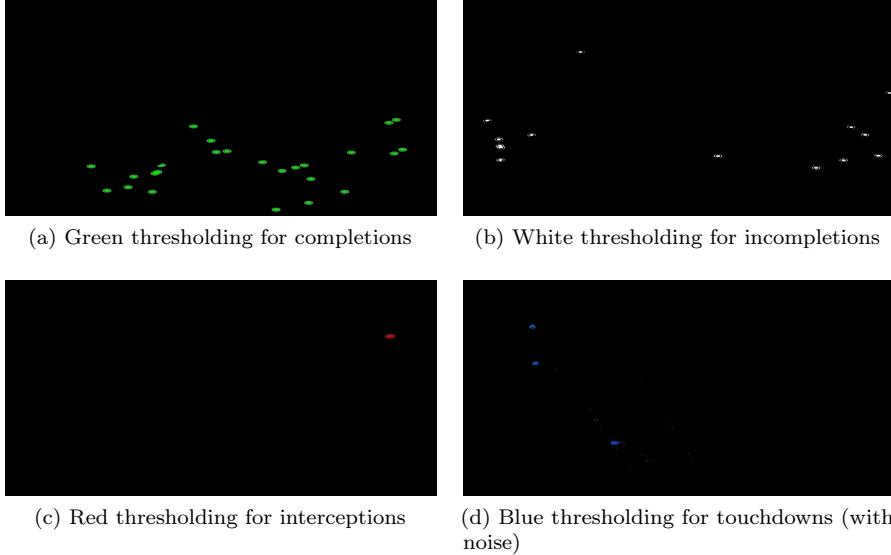


Figure 5: Color thresholding for all pass types. K -means and DBSCAN are subsequently performed on each of these images for pass detection. Noise in the segmented touchdown image results from the pass location having similar color to the line of scrimmage and pass trajectory lines.

n_c presented on the image excludes the touchdown passes and hence, $n_c = n_{tot-c} - n_{td}$, while the number of gray incompletions is $n_{inc} = n_a - n_c - n_{td} - n_{int}$.

Using the segmented images and the number of passes of each type, we detect the pixel locations of each pass type within the respective image (see Sections 2.3.2 and 2.3.3). We then map these locations to the dimensions of a real football field to identify the on-field locations of each pass in (x, y) coordinates, on the coordinate system presented in Figure 3. The $y = 0$ vertical line runs through the center of the field, while the $x = 0$ horizontal line always represents the line of scrimmage.

2.3.2 Identifying Pass Types with K -Means++

K -means is a distance-based clustering method used to define a clustering partition \mathcal{C} of n observations $X_i \in \mathbb{R}^p$, for $i = 1, \dots, n$, into a pre-determined number of clusters K such that the within-cluster variation (in Euclidean space),

$$\sum_{k=1}^K \sum_{\mathcal{C}(i)=k} ||X_i - c_k||^2,$$

with $c_k = \bar{X}_k = \frac{1}{n_k} \sum_{\mathcal{C}(i)=k} X_i$,

is minimized (MacQueen, 1967). The resulting clustering \mathcal{C} from K -means assumes the co-variance structure of the clusters is spherical, which works well for our purposes, as all pass locations are represented by circles in the images.

Rather than searching over all possible partitions, Lloyd's algorithm is the standard approach used for determining K -means partitions \mathcal{C} :

1. choose K random points as starting centers c_1, \dots, c_K ,
2. minimize over \mathcal{C} : assign each point X_1, \dots, X_n to its closest center c_k , $\mathcal{C}(i) = k$,
3. minimize over c_1, \dots, c_K : update the centers to be the average points $c_k = \bar{X}_k$ for each $k = 1, \dots, K$ clusters,
4. repeat steps (2) and (3) until within-cluster variation doesn't change.

Rather than using random initialization for the cluster centers, we use the K -means++ algorithm to choose better starting values (Arthur and Vassilvitskii, 2007). An initial point is randomly selected to be c_1 , initializing the set of centers $C = \{c_1\}$. Then for each remaining center $j = 2, \dots, K$:

1. for each X_i , compute $D(X_i) = \min_{c \in C} ||X_i - c||$,
2. choose a point X_i with probability,

$$p_i = \frac{D^2(X_i)}{\sum_{j=1}^n D^2(X_j)}$$

3. use this point as c_k , update $C = C \cup \{c_k\}$.

Then we proceed to use Lloyd's algorithm from above with the set of starting centers C chosen from a weighted probability distribution. According to this distribution, each point has a probability of being chosen proportional to its squared Euclidean distance from the nearest preceding center (Arthur and Vassilvitskii, 2007).

The K -means++ algorithm is more appropriate to use for pass detection for two reasons. First, by definition, the initialized cluster centers are more likely to be spread further apart from each other in comparison to K -means. This means that the possibility of two cluster centroids falling within the same cluster/single pass location is greatly reduced. Second, even though K -means++ is typically

sensitive to outliers, our data does not present this issue, since all colors on the charts aside from the pass locations (e.g. the line of scrimmage and yardline markers) are already removed, as described above.

To identify complete and intercepted passes, we simply perform K -means++ clustering on the appropriate segmented images with $K = n_c$ and $K = n_{int}$, respectively. While in theory we can do the same for the touchdown passes (i.e. perform K -means++ with $K = n_{td}$), the segmented image sometimes includes outliers, since the line of scrimmage and the touchdown pass trajectories have similar colors to the points representing catch locations for touchdown passes. Thus, we need to further process the corresponding segmented image to remove these unnecessary colored pixels. For this step, we use DBSCAN as we detail in the following section, and apply K -means++ to the resulting image.

One major difficulty in detecting the number of incompletions is that the number of incomplete passes shown on a pass chart image may not match the number given in the JSON data corresponding to an image, n_{inc} . This is most likely because out-of-bounds passes and spikes are not presented in the charts despite being counted as pass attempts. Our first step to solving this issue is performing K -means++ clustering on the segmented image for incompletions, with $K = n_{inc}$. Once we have all n_{inc} cluster centers, we iterate through each cluster center and examine how far away the other cluster centers are. If two cluster centers are distinctly close to each other, one of following two cases is true:

1. Two cluster centers have been mistakenly detected for a single pass location.
This might happen if the metadata specifies that there are 21 incompletions, but the image only shows 20 (e.g. because one pass was out of bounds). In this case, K -means++ will split a single pass location into two, in order to achieve the specified number of clusters K .
2. Two cluster centers have been correctly detected for two pass locations that are close to each other.

We can infer which of the two cases is true by comparing the within-cluster variation of each of these two clusters with the within-cluster variation of a *normal*, single pass location cluster. If the former is significantly smaller than the latter, then case (1) is detected and we reduce the number of incompletions shown in the image by one, otherwise, case (2) is detected and no additional action is required. The result of this iterative process is a newly-adjusted number of incompletions, $n_{inc-adj}$. If $n_{inc-adj} = n_{inc}$ the process is terminated; otherwise we perform K -means++ clustering again with $K = n_{inc-adj}$.

After obtaining the (x, y) pixel locations of all cluster centers of a given pass type, we map these coordinates to real field locations relative to the line of scrimmage. For the number of incomplete passes whose locations could not be identified, $n_{inc} - n_{inc-adj}$, we populate these rows in the data with N/A values. Figure 6 shows the result of extracting pass locations using K -means++, and we mark the cluster centers in red. We note that K -means++ is able to detect overlapping pass locations as multiple passes.

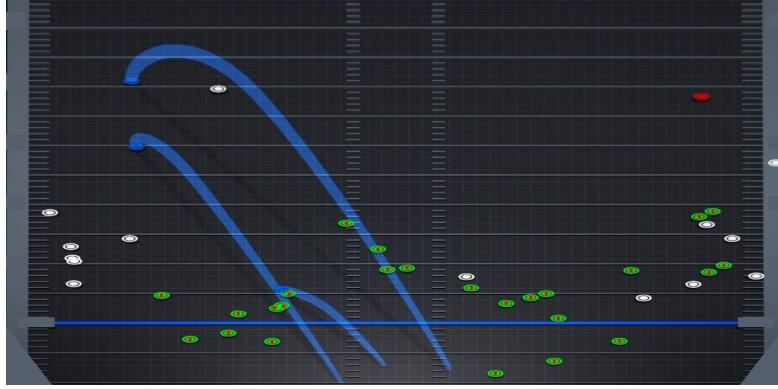


Figure 6: An example result of performing K -means on Nick Foles’ complete passes from Super Bowl LII, with centers of each pass depicted in red.

2.3.3 Removing Noise in Touchdown Images with DBSCAN

Our segmented touchdown images require additional processing after color thresholding in order for K -means++ to correctly identify the pass locations. As mentioned above, this is because line of scrimmage is shown in blue, and because pass trajectories for touchdown passes are included in the image and shown in a similar color. As a result, extraneous blue pixels (“noise”) are often present after image segmentation.

To address this issue, we use DBSCAN, a density-based clustering algorithm that identifies clusters of arbitrary shape for a given set of data points (Ester et al., 1996). DBSCAN is a non-parametric clustering algorithm that identifies clusters as *maximal* sets of density-connected points. Specifically, the DBSCAN algorithm works as follow:

1. Let $X = \{x_1, x_2, \dots, x_n\}$ be a set of observations (“points”) to cluster
2. For each point x_i , compute the ϵ -neighborhood $N(x_i)$; all observations within a distance ϵ are included in $N(x_i)$
3. Two points, x_i and x_j , are merged into a single cluster if $N(x_i)$ overlaps with $N(x_j)$
4. Recompute the ϵ -neighborhood $N(C_k)$ for each cluster C_k
5. Two clusters, C_k and C_l , are merged into a single cluster if $N(C_k)$ overlaps with $N(C_l)$
6. Repeat steps 4-5 until no more clusters overlap
7. If the number of points in a cluster is greater than or equal to a pre-defined threshold τ (i.e. if $|N(p)| \geq \tau$), this cluster is retained

- If the number of points in a cluster is less than a pre-defined threshold τ (i.e. if $|N(p)| < \tau$), this cluster is considered “noise” and discarded

Even though DBSCAN does not require a direct specification of the number of clusters, it should be clear that the choice of ϵ and τ (often referred to as “minPoints”) impacts the number of clusters identified. Figure 7 depicts a high level representation of DBSCAN’s operations, with $\tau = 3$.

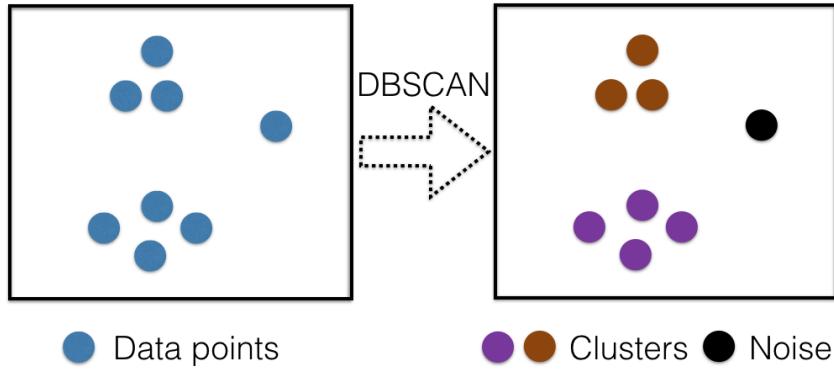


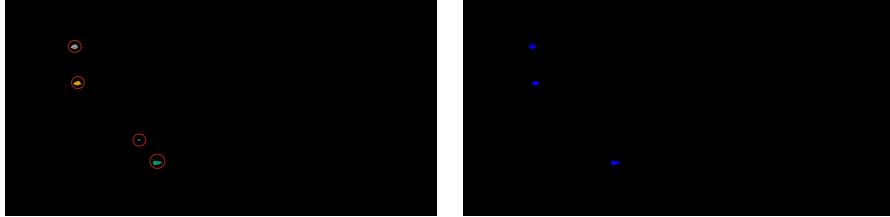
Figure 7: Pictorial representation of DBSCAN’s operations ($\tau=3$).

DBSCAN’s ability to identify *noise* makes it particularly good choice for identifying the locations of touchdown passes. For our purposes, the observations are the individual pixels in the segmented image. To distinguish between actual pass locations and this noise, we use DBSCAN on these observations/pixels to find the n_{td} highest density clusters, with $\epsilon = 10$ and $\tau = n_{td}$. Then we remove from the image any pixels detected as noise or as not belonging to the n_{td} -top density clusters. We finally pass the resulting image to the K -means++ algorithm described above to obtain the raw locations.

Figure 8 shows the output after applying DBSCAN on Figure 5(d) to extract touchdown pass locations. In the original pass chart there are three touchdown passes, while as we see at Figure 8(a) DBSCAN has detected 4 distinct clusters (after removing the points identified by the algorithm as noise). Figure 8(b) shows the selection of the $n_{td} = 3$ most dense clusters that represent the touchdown pass locations. This new version of the segmented image, with noise removed, is then used as input to the K -means++ approach described above.

2.3.4 Output Data

The resulting data from the pass charts cover the 2017 and 2018 regular seasons and postseasons. There are 27,946 rows containing data for 840 pass charts spanning 491 games, with 27,171 rows containing no missing values. Of the cases with missingness, 33 are due to Next Gen Stats not providing pass charts for either team during a game. Appendix A provides an overview of all variables provided by `next-gen-scrappy` and their descriptions, while Table 2 provides a



(a) 4 clusters identified by DBSCAN, circled in red.

(b) $n_{td} = 3$ largest clusters

Figure 8: The result of performing DBSCAN for touchdown pass detection.

breakdown of some basic statistics for the number of pass locations detected by season. Finally, Appendix B contains a subset of the dataset for 10 of Nick Foles’ passes in Super Bowl LII.

2.4 Anecdote Validation

In order to showcase the quality of the data collected by `next-gen-scrappy`, we turned to the tracking data provided by the NFL for the Big Data Bowl competition mentioned earlier. Unfortunately, we cannot compare all the data points collected from `next-gen-scrappy` to the league-provided tracking data. For example, there are ambiguities in the tracking data when it comes to incomplete passes: the tracking data does not specify the point at which the ball hits the ground and is rendered incomplete. Therefore, we focus on completed passes, where the event of completion is clearly annotated in the tracking data. We further transform the tracking data coordinates in reference to the line of scrimmage in order to be directly comparable to the data obtained from `next-gen-scrappy`. We demonstrate the accuracy of our approach, as compared to the tracking data using this procedure, in Section 4.

3 Evaluating and Characterizing Passers and Pass Defenses

In this section, we model pass completion percentage conditional on pass location for the NFL, individual passers, and team defenses. We focus on the field range between 10 yards behind the line of scrimmage to 55 yards in front of the line of scrimmage, since this is where almost all pass locations are located in the two-year span for which we have data. For each group (QB, team defense, or NFL-wide), our model has two main components: a two-dimensional kernel density estimate (KDE) for pass target locations and a generalized additive model (GAM) for completion percentage by location.

3.1 NFL-Wide Models

Below, we describe two models for NFL passing: (1) a league-wide pass target location distribution, estimated using kernel density estimation; and (2) a model for league-wide completion percentage by pass target location, estimated with a generalized additive model.

It is important to recognize that both models use only observational data, and thus should only be used to describe what has happened in the past. The observed data is biased in obvious ways: QBs tend to target open receivers, defenses tend to cover high-leverage areas of the field more closely, and target locations depend greatly on the game situation (e.g. yards to first down). None of this information is available in our dataset, but all of this information will influence on-field decisions that are made by QBs and teams.

3.1.1 Estimating the Distribution of NFL Pass Locations

We estimate the league-wide pass location distribution via kernel density estimation (KDE). KDE is a non-parametric approach for estimating a probability distribution given only observational data. In the univariate case, a small probability density function (“kernel”) is placed over each observation x_i , and these kernels are aggregated across the entire dataset x_1, \dots, x_n . Let K be the kernel function, n the number of data points, and h a smoothing parameter. Then, the univariate, empirical density estimate via KDE is: $\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)$.

In our case, we use KDE to estimate the probability of a pass targeting a two-dimensional location (x, y) on the field, relative to the line of scrimmage, $\hat{f}(x, y)$. Density estimation via two-dimensional KDE follows a similar form:

$$\hat{f}_h(x, y) = \frac{1}{nh_x h_y} \sum_{i=1}^n \left\{ K_x\left(\frac{x-x_i}{h_x}\right) K_y\left(\frac{y-y_i}{h_y}\right) \right\}$$

where h_x and h_y are bandwidths for x and y , respectively, and K_x and K_y are the respective kernels. Alternative definitions use joint kernels $K_{x,y}$ or identical kernels K , but for our purposes, the above definition suffices.

We use a bivariate normal kernel K for our estimation (which can be decomposed into independent kernels K_x and K_y), and Scott’s rule of thumb heuristic for the bandwidth (Venables and Ripley, 2002).⁷ We bound the KDE within the rectangular box described above: 10 yards behind the line of scrimmage, 55 yards in front of the line of scrimmage, and between both sidelines. The resulting KDE gives us an empirical estimate of the pass target locations for the entire NFL.

An alternative approach would be to obtain a two-dimensional histogram using a grid over the field and estimating the number of passes that were thrown in each of the grid cells. However, this approach has some limitations, including the histograms’ sensitivity to the anchor point (Silverman, 1986). Furthermore,

⁷ $\hat{h} = 1.06 \times \min(\hat{\sigma}, \frac{IQR}{1.34}) \times n^{-1/5}$, where σ is the standard deviation and IQR is the interquartile range of the values in the corresponding dimension

the estimates obtained from overlaying a grid over the field are essentially a discrete approximation of a continuous surface. Two points (x_1, y_1) and (x_2, y_2) that belong to the same grid, will have the same probability of being targeted with a pass since they belong to the same grid cell. Of course, this problem can be minimized by having the grid cells as small as possible (e.g., 0.5 yards each cell side), but then, sample size concerns limit the interpretability of the resulting density estimate, since we will end up with several empty cells and a noisy estimation. KDE provides a continuous approximation over the surface, smoothing the differences between neighboring points.

3.1.2 Estimating the NFL Completion Percentages Surface

We use generalized additive models (GAMs) to model the probability $P(\text{Completion}|x, y)$ of a pass being completed given the targeted location (x, y) of the field (Hastie and Tibshirani, 1990).

GAMs are similar to generalized linear models (GLMs), except where the response variable is linearly associated with smooth functions of the independent variables (via some link function, e.g. the logit for a binomial response). This is in contrast to typical GLMs (e.g. logistic regression), where the response variable is linearly associated with the independent variables themselves (via some link function). Formally, if y is the dependent variable, and x_i , $i \in \{1, \dots, n\}$ are the independent variables:

$$y = g(\alpha_0 + f_1(x_1) + f_2(x_2) + \dots + f_n(x_n))$$

where g^{-1} is the link function for the model. For binary data, as in our case, the link function is the logit function, $g^{-1}(z) = \log(\frac{z}{1-z})$. The functions f_i can take many forms, but can be thought of as smoothers between the dependent variable y and the independent variable x_i .

For our model, we use two independent variables – the vertical coordinate x and the horizontal coordinate y – and an interaction between them $x \cdot y$. We posit that all three terms are necessary: First, x represents the pass location down the length of the field, and intuitively should have some marginal effect on completion percentage. For example, longer passes are more difficult to complete. Second, y represents the pass location across the width of the field, and intuitively should have some (non-linear) marginal effect on completion probability. For example, passes towards the sidelines are farther away, while passes to the middle of the field are closer, potentially affecting completion probability; target locations towards the middle of the field typically have more defenders in that area, potentially affecting completion probability. Third, we include the interaction term, since we hypothesize that there is some joint non-linear effect on completion probability that is not described by the marginal terms alone. For example, the completion probabilities of passes behind the line of scrimmage likely are likely not aptly described by the marginal terms alone; there is likely some joint relationship between (x, y) and the completion probability. Thus, we have:

$$\log\left(\frac{P(Complete)}{P(Incomplete)}\right) = c_0 + f_x(x) + f_y(y) + f_{xy}(x, y)$$

We choose f_x , f_y and f_{xy} to be tensor product interaction smoothers, which work well in situations where both marginal and interaction effects are present (Wood, 2019). Specifically, this type of smoothing function accounts for the marginal bases from the main effects when estimating the smoothing function for the interaction effect, and it allows for the functional ANOVA decomposition in the model equation above, which is more interpretable than alternative choices. Cross-validation supports our choice of smoother, and visual inspection indicates that this smoother yields interpretable results that make sense in the context of football (see Section 4 for examples).

Alternative models to the GAM are not ideal for this situation. Generalized linear models, for example, will not allow for non-parametric or smooth relationships between the independent variables (distance from line of scrimmage, distance from center of field) and the response variable (completion percentage), and thus will yield a poor fit to the data. Tree-based models (e.g. decision trees, random forests, gradient boosted trees, etc) partition the surface of the field into discrete areas, and will not yield the same smooth surfaces that we obtain from the GAM. In short, the GAM is an ideal choice for the continuous nature of our response variable and the structure of our explanatory data; and it appropriately models and captures the relationship between our explanatory and response variables.

3.2 2-D Naive Bayes for Individual Completion Percentage Surfaces

For individual QBs or team defenses, we can directly apply the methodology used in Section 3.1 to subsets of the data corresponding to these groups. As a result, we can obtain quick estimates of the pass target location for an individual quarterback and the completion percentage surface for an individual quarterback. Similarly, for team defenses, we can take the subset of passes against this team (since opposing defense is included in the metadata described in Section 2), and examine the distribution of pass locations *allowed* by a specific team, and the pass completion percentage surface *allowed* by a specific team.

However, in doing so, we quickly run into sample size issues when estimating the completion percentage model. While we had over 27,000 passes with which to estimate the league-wide models in Section 3.1, a typical NFL team only attempts between 400 and 700 passes in an entire season, and the number of attempts for individual QBs can be even lower than that. Moreover, in specific areas of the field that are less frequently targeted, some QBs may only attempt a handful of passes to that area of the field over the course of an entire season. As such, any model built using these small samples of data for individual QBs and team defenses will likely overfit the data.

In response to this issue, we introduce a two-dimensional naive Bayes approach for estimating the completion percentage surfaces for individual QBs and team

defenses. Our approach regresses the estimates for individual QBs and team defenses towards the league-average completion percentage *in each area of the field*, and adaptively accounts for the sample size of passes in each particular area of the field. Our model is described as follows:

- Let g be the group or player of interest (e.g. an individual QB or a team defense).
- Let $\hat{f}_{NFL}(x, y)$ be the pass target location distribution for the NFL. Let $\hat{f}_g(x, y)$ be the pass target location distribution for group g .
- Let $\hat{P}_{NFL}(\text{Complete}|x, y)$ be the estimated completion percentage surface for the NFL.
- Let $\hat{P}_g(\text{Complete}|x, y)$ be the estimated completion percentage surface for g .
- Let N_{Median} be the median number of pass attempts in the class of group g (for QBs, the median number of passes attempted for all QBs; for team defenses, the median number of pass attempts allowed for all teams). (This is a parameter that can be adjusted to give more or less weight to the league-wide distribution.)
- Let N_g be the number of total passes attempted in the NFL and for g (or against g , in the case of team defenses).
- Finally, let $\hat{P}_G^*(\text{Complete}|x, y)$ be our 2-D naive Bayes estimate of the completion percentage surface for g .

Then,

$$\hat{P}_G^*(\text{Complete}|x, y) = \frac{N_g \times \hat{f}_g(x, y) \times \hat{P}_g(\text{Complete}|x, y) + N_{Median} \times \hat{f}_{NFL}(x, y) \times \hat{P}_{NFL}(\text{Complete}|x, y)}{N_g \times \hat{f}_g(x, y) + N_{Median} \times \hat{f}_{NFL}(x, y)}$$

In words, our approach goes through each location on the completion percentage surface for g and scales it towards the league-average completion percentage surface from Section 3.1, using a scaling factor of N_{Median} . The above approach is Bayesian in nature: The second terms, on the right hand side of the numerator and denominator, can be thought of as the *prior* estimate for the completion percentage. Then, as we accumulate more evidence (i.e. as N_g increases), the first term is given more weight.

Similar alternatives may also be ideal, depending on the context. For example, we could instead shift the weight from the prior to the data in group g instead of using a static weight of N_{Median} . This is easily implemented, and we encourage interested readers to use the open-source code that we provide for these models and try their own approaches.⁸

⁸<https://github.com/sarahmallepalle/next-gen-scrapy>

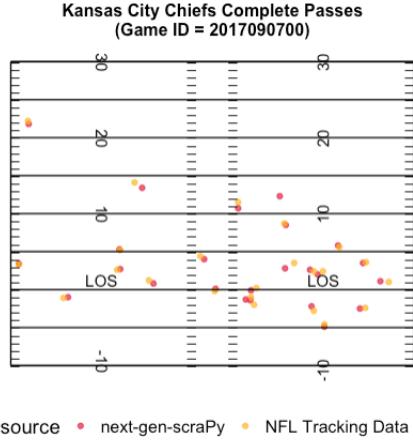


Figure 9: The pass locations obtained from `next-gen-scrappy` closely match those from the player tracking data.

4 Results

In this section, we use the data collected and the modeling framework described in the previous section to analyze the league-wide passing trends and the performances of individual quarterbacks and team defenses passing.

4.1 Accuracy of Data Provided by `next-gen-scrappy`

As described in Section 2.4, we cannot compare *all* of the data provided by `next-gen-scrappy` to league-provided player tracking data for several reasons: (1) the pass locations for incompletions are not marked in the tracking data; (2) the tracking data provides only six weeks of data from 2017 (while we provide data from all weeks from the 2017 and 2018 seasons); and (3) it is impossible to link specific passes from the NGS charts the tracking data, since play identifiers and/or timestamps are not given.

We can, however, provide an example of how our completed pass locations closely match those from the tracking data for an individual game. Figure 9 visualizes the locations of the completed passes for Kansas City from the opening game in the 2017 season against New England. The points are very well-aligned, providing a validating example of the accuracy with which `next-gen-scrappy` obtains raw pass locations by processing the images from Next Gen Stats.

Interestingly, in this example, there is one pass (top-right of chart) that is identified by `next-gen-scrappy` but has no counterpart in the tracking data. Going back to the NGS chart for this game,⁹ this point is clearly present in the chart; it is unclear why there is a discrepancy with the tracking data.

⁹https://charts-cdn-b.nextgenstats.nfl.com/static-charts/900/pass-chart_SMI031126_2017-reg-1_1504847939195.jpeg

Otherwise, the small deviations from the tracking data for individual passes could occur for several reasons. For example, one source may mark the location of the ball, while another source may mark the location of the player catching the ball. Since the Radio Frequency Identification (RFID) chips used to obtain the player tracking data are placed in players' shoulder pads, any catch that involves the receiver reaching out for the ball will have some small deviation between the ball and player locations.

4.2 League-Wide Passing Trends

Figure 10 shows the distribution of pass locations (relative to the line of scrimmage) and the results of the generalized additive model for predicting completion probabilities for the 2017 and 2018 NFL seasons. Comparing the two seasons, we can see that the distributions of pass locations are almost identical, with most passes are relatively short and near the line of scrimmage. Specifically, in 2017, 68.5% of targets were within 10 yards down the field past the line of scrimmage, and 88.8% were within 20 yards. In 2018, 66.5% of targets were within 10 yards, with 89.2% were within 20 yards. For targets further than 20 yards past the line of scrimmage, almost all were towards the sidelines. Only 2.96% of targets past 20 yards in 2017 and 3.11% in 2018 were in the middle of the field (i.e., $-13.33 \leq x \leq 13.33$).

The predicted completion probability follows a similar trend. Passes closer to line of scrimmage have a higher completion percentage. The median predicted completion percentages for the 2017 and 2018 seasons are 73.3% and 76.8% respectively for passes within 10 yards of the line of scrimmage, while these numbers drop to 65.4% and 69.7% for passes within 20 yards. For passes more than 20 yards past the line of scrimmage, the median completion percentages fall drastically to 28.4% and 28.0% respectively. For these deeper passes, the median completion percentage was 30.4% towards the middle and 26.5% towards the sidelines in 2017, and 30.5% towards the middle and 22.7% towards the sidelines in 2018.

This analysis is strictly retrospective, given the observational nature of our dataset, and does not include every game played throughout 2017 and 2018. Potential biases in our dataset may result from the differences in play-calling for specific teams and QBs, differences in decision-making for specific teams and QBs, and the openness receivers and areas of the field to which passes are targeted.

4.3 Team Defense Completion Percentage Allowed Surfaces

Our framework also allows us to explore completion percentage allowed for each team defense by location of the field for an entire season. For this analysis, we compare our model output with Football Outsiders' Defensive Efficiency Ratings, specifically Defense DVOA (Defense-adjusted Value over Average). DVOA uses the idea of "success points" to measure a team's success based on total yards and

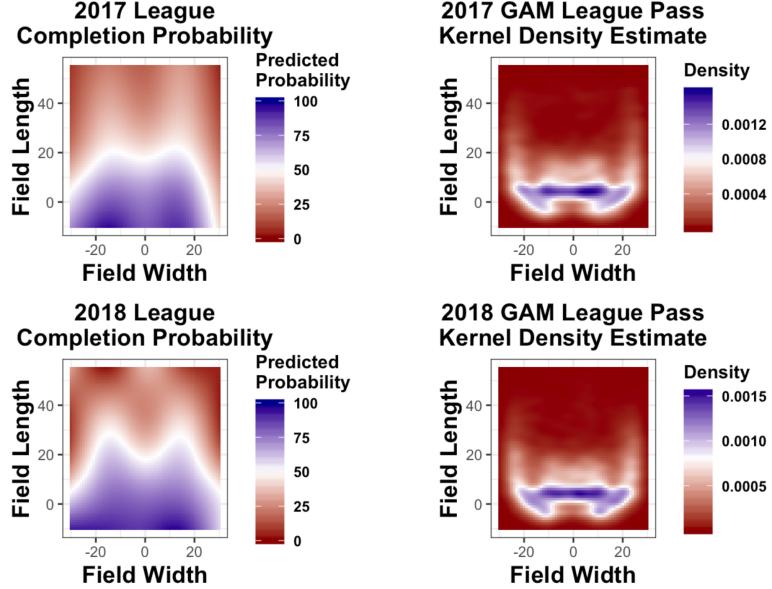


Figure 10: The individual components of our model: kernel density estimates for league-wide pass location probability (first column), and generalized additive models for league-wide completion probability (second column).

yards towards a first down for every play in an NFL season. With the Defense DVOA metric, 0% represents league-average performance, a positive percentage signifies performance in a situation benefitting the opposing offense, and a negative percentage signifies performance benefitting the team’s own defense (Schatz, 2006). In the 2018 regular season, the Chicago Bears and Baltimore Ravens had the first and third lowest Defensive DVOAs of -26.9% and -14.2% , respectively. In Figure 11, we present the predicted completion percentage from our model for the Chicago Bears and Baltimore Ravens. We present both the *raw* predictions (left column), as well as, relative to the league-average (right column). As we can see, in agreement with the Defensive DVOA metric, both teams are projected to allow lower completion percentage over the entire field. Furthermore, Figure 12 depicts the same surfaces for the two defensive teams with the third and fifth highest Defensive DVOAs, namely, the Oakland Raiders at 12.3% and Cincinnati Bengals at 9.0% . Compared to the results for the Bears and the Ravens, we can see that Bengals and Raiders are worse than average in completion percentage allowed over several parts of the field and particularly down the middle of the field.

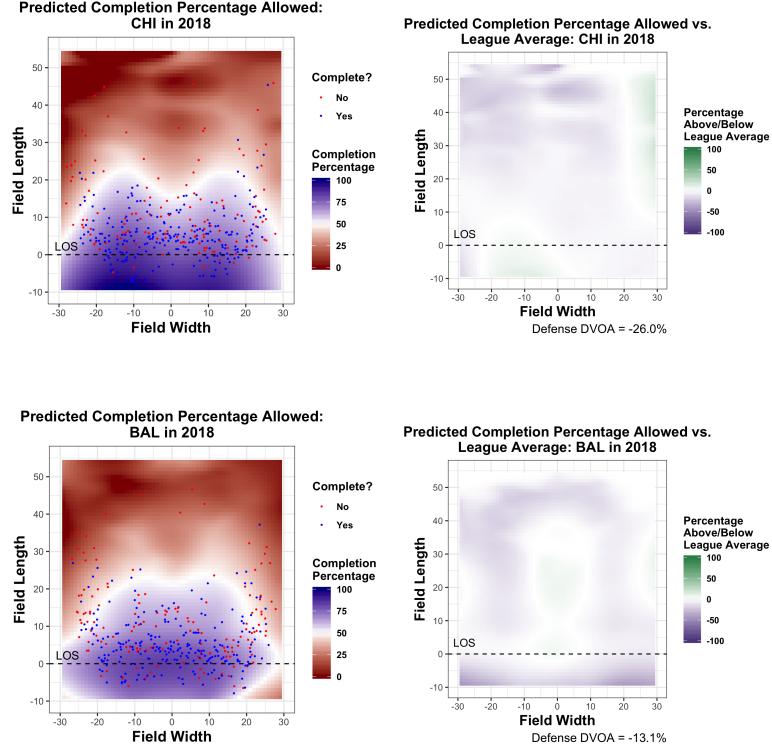


Figure 11: Completion percentage allowed surfaces of the Chicago Bears and Baltimore Ravens defense, who have the highest Expected Points Contributed By All Defense according to Pro Football Reference.

4.4 Quarterback Completion Percentage Surfaces

Our framework also allows us to examine patterns and make league comparisons at the quarterback level. We also compare our quarterback evaluations to ESPN’s Total QBR metric, which describes a quarterback’s contributions to winning in terms of passing, rushing, turnovers, and penalties, as well as accounting for his offense’s performance for every play (Katz and Burke, 2017). In the 2018 regular season, the quarterbacks with the highest Total QBRs were Patrick Mahomes and Drew Brees, with total ratings of 81.8 and 80.8, respectively. Figures 13 and 14 present the predicted from our model completion percentage by field location for the two QBs overlaid by the raw pass charts. When comparing Brees and Mahomes to league average, we find that both quarterbacks perform either equal to or above league average in almost possible target location on the field, indicated by the white (average) and green (above average) areas. On the other end of the spectrum, the quarterbacks with the lowest total ratings

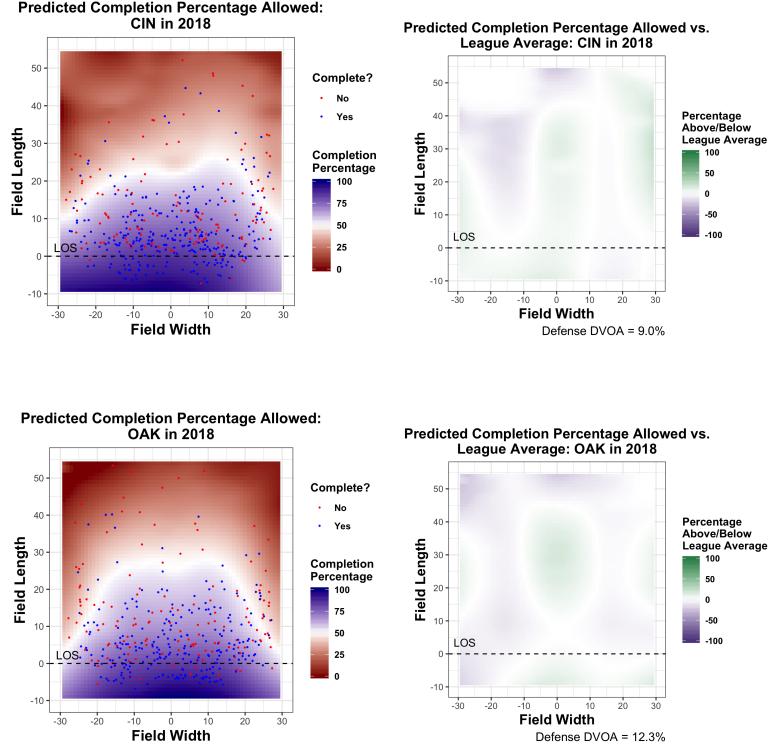


Figure 12: Completion percentage allowed surfaces of the Cincinnati Bengals and Oakland Raiders defense, who have the lowest Expected Points Contributed By All Defense according to Pro Football Reference.

were Joshua Allen, ranked 24th in the league, and Joshua Rosen, ranked last in the league at 33rd, with ratings of 52.2 and 25.9, respectively. Similarly, when comparing Allen and Rosen to league average, we observe that they performed below average (purple) across the field.

5 Discussion and Conclusions

In this paper, we presented `next-gen-scrappy`, an software package that allows football fans and analysts to extract and analyze the underlying data from the pass charts provided by NFL Next Gen Stats via their player tracking technology. With `next-gen-scrappy`, we implement a computer vision module that processes the images (pass charts) provided by Next Gen Stats, a K -Means++ clustering approach to identify passes and their locations on the field (relative to the line of scrimmage), and a DBSCAN clustering approach to remove noise from certain segmented images. The resulting dataset contains all pass locations from the

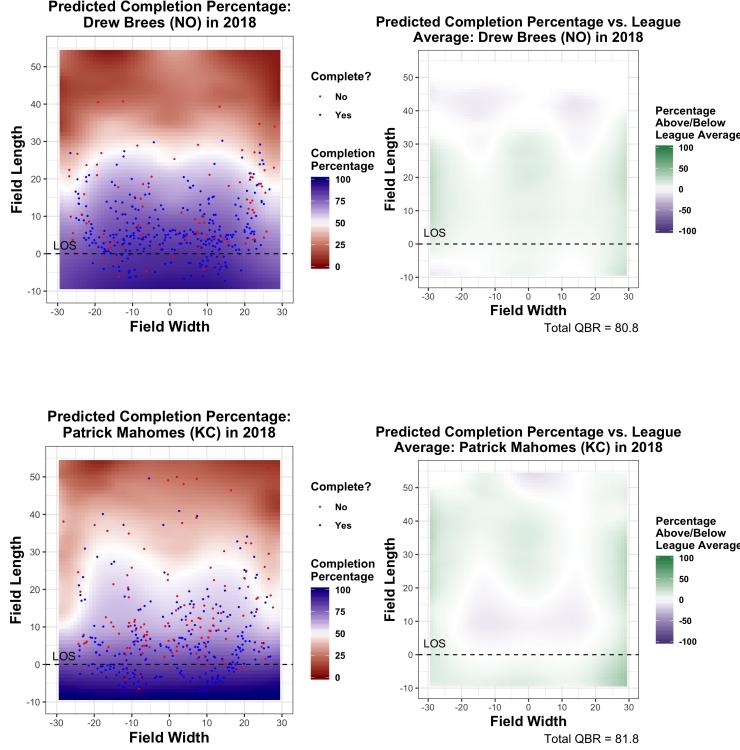


Figure 13: Completion percentage surfaces for Drew Brees and Patrick Mahomes, the quarterbacks with the highest passer ratings in the 2018 regular season, according to Next Gen Stats.

2017 and 2018 seasons that were tracked by the NFL player and ball tracking technology and shown on their Next Gen Stats website. This provides researchers with an abundance of data from 500+ games across two full seasons (including the most recent and relevant season), as compared to the only six weeks of data from 2017 that were temporarily made available by the NFL.

Using the resulting dataset, which we make available publicly, we build statistical models for the completion percentages by location on the field for the NFL, for individual QBs, and for team defenses. To do this, we combine the use of generalized additive models and kernel density estimations via a two-dimensional naive Bayes approach.

While this current work only pertains to pass charts, Next Gen Stats also provides route charts for receivers, and carry charts for running backs. In future work, we hope to extend the functionality of `next-gen-scrappy` to extract this information and provide it to the public. Extracting receiver routes and rusher carry paths is substantially more challenging. We hypothesize that mixture

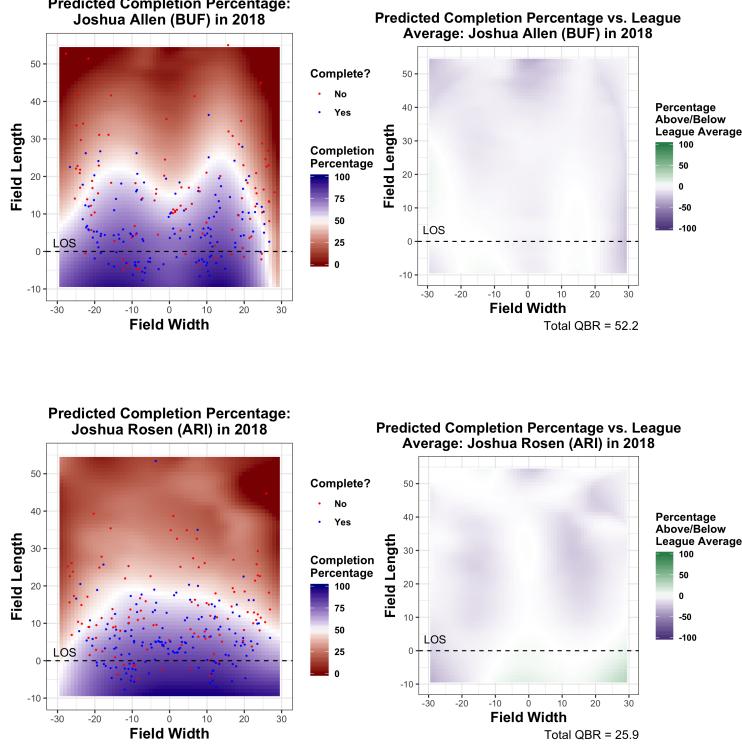


Figure 14: Completion percentage surfaces for Joshua Allen and Joshua Rosen, the quarterbacks with the lowest passer ratings in the 2018 regular season, according to Next Gen Stats.

regression models for detecting the trajectories from the image, but we have only completed very preliminary work on this topic to date.

Finally, future researchers may attempt to link the passes in our dataset to specific plays, e.g. from the NFL’s player tracking data or to play-by-play data from the `nflscrapR` package Horowitz et al. (2017), using (for example) the air yards information provided by such data sources. Doing so would open up many potential avenues of future research, such as estimating of spatial expected points added (EPA) or win probability added (WPA) surfaces Yurko et al. (2019).

A Data Scrapped from Next Gen Stats

Variable	Description
completions	number of completions thrown
touchdowns	number of touchdowns thrown
attempts	number of passes thrown
interceptions	number of interceptions thrown
extraLargeImg	URL of extra-large-sized image (1200 x 1200)
week	week of game
gameId	10-digit game identification number
season	NFL season
firstName	first name of player
lastName	last name of player
team	team name of player
position	position of player
seasonType	regular ("reg") or postseason ("post")

B Example Subset of Data

game_id	team	week	name	pass_type	x_coord	y_coord	type	home_team	away_team	season
2018020400	PHI	super-bowl	Nick Foles	COMPLETE	-3.6	16.9	post	NE	PHI	2017
2018020400	PHI	super-bowl	Nick Foles	COMPLETE	16.2	-3.0	post	NE	PHI	2017
2018020400	PHI	super-bowl	Nick Foles	COMPLETE	11.5	-6.4	post	NE	PHI	2017
2018020400	PHI	super-bowl	Nick Foles	TOUCHDOWN	-8.5	5.7	post	NE	PHI	2017
2018020400	PHI	super-bowl	Nick Foles	TOUCHDOWN	-18.8	30.1	post	NE	PHI	2017
2018020400	PHI	super-bowl	Nick Foles	TOUCHDOWN	-19.3	41.2	post	NE	PHI	2017
2018020400	PHI	super-bowl	Nick Foles	INTERCEPTION	21.8	37.9	post	NE	PHI	2017
2018020400	PHI	super-bowl	Nick Foles	INCOMPLETE	5.1	7.9	post	NE	PHI	2017
2018020400	PHI	super-bowl	Nick Foles	INCOMPLETE	-12.9	39.6	post	NE	PHI	2017
2018020400	PHI	super-bowl	Nick Foles	INCOMPLETE	26.1	8.0	post	NE	PHI	2017

References

Arthur, D. and S. Vassilvitskii (2007): “K-means++: The advantages of careful seeding,” in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’07, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1027–1035, URL <http://dl.acm.org/citation.cfm?id=1283383.1283494>.

Baumer, B., S. Jensen, and G. Matthews (2015): “openWAR: An open source system for evaluating overall player performance in major league baseball,” *Journal of Quantitative Analysis in Sports*, 11.

Benz, L. (2019): *ncaahoopR: NCAA Men’s Basketball Play-By-Play Functionality*, r package version 1.4.2.

- Burke, B. (2019): “Deepqb: Deep learning with player tracking to quantify quarterback decision-making & performance,” *13th MIT Sloan Sports Analytics Conference*.
- Casella, P. (2015): “Statcast primer: Baseball will never be the same.” URL <https://www.mlb.com/news/statcast-primer-baseball-will-never-be-the-same/c-119234412>.
- Cervone, D., L. Bornn, and K. Goldsberry (2016a): “Nba court realty,” *10th MIT Sloan Sports Analytics Conference*.
- Cervone, D., A. D’Amour, L. Bornn, and K. Goldsberry (2016b): “A multiresolution stochastic process model for predicting basketball possession outcomes,” *Journal of the American Statistical Association*, 111, 585–599.
- Elmore, R. and P. DeWitt (2017): *ballr: Access to Current and Historical Basketball Data*, URL <https://CRAN.R-project.org/package=ballr>, R package version 0.1.1.
- Ester, M., H.-P. Kriegel, J. Sander, and X. Xu (1996): “A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD’96, AAAI Press, 226–231, URL <http://dl.acm.org/citation.cfm?id=3001460.3001507>.
- Fast, M. (2010): “What the heck is pitchf/x?” *The Hardball Times Baseball Annual 2010*, URL <http://baseball.physics.illinois.edu/FastPFXGuide.pdf>.
- Fernández, J., F. Barcelona, L. Bornn, and D. Cervone (2019): “Decomposing the immeasurable sport: A deep learning expected possession value framework for soccer,” *13th Annual MIT Sloan Sports Analytics Conference*.
- Friendly, M., C. Dalzell, M. Monkman, and D. Murphy (2019): *Lahman: Sean ‘Lahman’ Baseball Database*, URL <https://CRAN.R-project.org/package=Lahman>, r package version 7.0-1.
- Gudmundsson, J. and M. Horton (2017): “Spatio-temporal analysis of team sports,” *ACM Computing Surveys (CSUR)*, 50, 22.
- Hastie, T. J. and R. J. Tibshirani (1990): “Generalized additive models,” *Mono-graphs on statistics and applied probability*, 43, 205–208.
- Horowitz, M., R. Yurko, and S. L. Ventura (2017): *nflscrapR: Compiling the NFL play-by-play API for easy use in R*, URL <https://github.com/maksimhorowitz/nflscrapR>, r package version 1.4.0.
- Katz, S. and B. Burke (2017): “How is total QBR calculated? we explain our quarterback rating,” URL http://www.espn.com/blog/statsinfo/post/_/id/123701/how-is-total-qbr-calculated-we-explain-our-quarterback-rating.

- Koschan, A. and M. A. Abidi (2008): *Digital Color Image Processing*, New York, NY, USA: Wiley-Interscience.
- Le, H. M., Y. Yue, P. A. Carr, and P. Lucey (2017): “Coordinated multi-agent imitation learning,” in *Proceedings of the 34th International Conference on International Conference on Machine Learning (ICML)*.
- Lowe, Z. (2013): “Lights, cameras, revolution,” URL <http://grantland.com/features/the-toronto-raptors-sportvu-cameras-nba-analytical-revolution/>.
- MacQueen, J. (1967): “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, Berkeley, Calif.: University of California Press, 281–297, URL <https://projecteuclid.org/euclid.bsmsp/1200512992>.
- NBA (2013): “Nba partners with stats llc for tracking technology.” URL <https://www.nba.com/2013/news/09/05/nba-stats-llc-player-tracking-technology/>.
- NFL (2019): “Nfl operations: Nfl next gen stats,” URL <https://operations.nfl.com/the-game/technology/nfl-next-gen-stats/>.
- Power, P., H. Ruiz, X. Wei, and P. Lucey (2017): “Not all passes are created equal: Objectively measuring the risk and reward of passes in soccer from tracking data,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’17.
- Schatz, A. (2006): “Methods to our madness,” URL <https://www.footballoutsiders.com/info/methods#DVOA>.
- Seidl, T., A. Cherukumudi, A. Hartnett, P. Carr, and P. Lucey (2018): “Bhost-gusters: Realtime interactive play sketching with synthesized nba defenses,” .
- Sievert, C. (2015): *pitchRx: Tools for Harnessing 'MLBAM' 'Gameday' Data and Visualizing 'pitchfx'*, URL <http://cpsievert.github.com/pitchRx>, r package version 1.8.2.
- Silverman, B. (1986): “Density estimation for statistics and data analysis,” Technical report.
- Szeliski, R. (2010): *Computer vision: algorithms and applications*, Springer Science & Business Media.
- Thomas, A. and S. L. Ventura (2013): *nhlscrapr: Compiling the NHL Real Time Scoring System Database for easy use in R*, URL <https://CRAN.R-project.org/package=nhlscrapr>, R package version 1.8.1.

- Venables, W. N. and B. D. Ripley (2002): *Modern Applied Statistics with S*, New York: Springer, fourth edition, URL <http://www.stats.ox.ac.uk/pub/MASS4>, ISBN 0-387-95457-0.
- Wood, S. (2019): *Define tensor product smooths or tensor product interactions in GAM formulae*, URL <https://cran.r-project.org/web/packages/mgcv/index.html>, R package version 1.8-28.
- Wyshynski, G. (2019): “Inside the arrival of nhl player tracking, from microchips to megabets.” URL http://www.espn.com/nhl/story/_/id/25872085/inside-arrival-nhl-player-tracking-microchips-megabets.
- Yurko, R., M. Horowitz, and S. Ventura (2019): “nflwar: A reproducible method for offensive player evaluation in football,” *Journal of Quantitative Analysis in Sports*, Forthcoming.