NFL Quarterback Performance Remains Difficult to Predict

Antonio D'Alessandro

April 23, 2022

1 Introduction

The National Football League is a multi-billion dollar business and one of the most successful entertainment companies in the history of the United States, with an average franchise valuation of approximately 3.5 billion dollars. The value of any individual franchise is dependent on a variety of factors like: location, revenue sharing through TV contracts (negotiated by the NFL as a whole), and merchandise / ticket sales. There are certainly bad teams which are "worth" quite a bit of money, see both New York franchises at the time of this writing. However, there are other teams, particularly those in smaller media markets which have drastically expanded their value in relatively short periods of time.

The Tampa Bay Buccaneer's, for example, saw a 29% rise in their value over the course of one year after they won the Super Bowl vs a league average of 7%. Similarly the Kansas City Chiefs increased their franchise value from 2.1 to 3 billion between 2018 - 2021, +42% vs. a league average of around +20% [Ozanian and Settimi, 2021]. Neither of these results is actually surprising. The Tampa Buccaneers won the Super Bowl in 2020 the same year they acquired Tom Brady, considered by many the greatest quarterback in the history of the game. While the time period $2018 - 2020$ captures the arrival of Patrick Mahommes, who was voted NFL MVP after his first season as a starter. Mahommes incredible quarterback play has catapulted the Chiefs from, at best a wild card contender, to perennial Super Bowl favorite. This argument may seem extreme to some, each NFL team has a roster of 53 players, a dozen coaches or more, plus an army of front office personnel. Is the contribution of one player really this important?

Over the past $10 - 12$ years there has been a surge in the amount of offense being generated during a typical football game. Points scored per game has been trending upwards since 2001 as well as total yards gained by the offense. Much of this trend can be traced to drastic improvements in quarterback play. Overall passing efficiency (the ratio of pass completed vs. attempted), passing yards and passing touchdowns per game have all being improving dramatically [Kelly, 2020]. The increase in passing

touchdowns is of particular interest. Looking back historically, teams with more quarterback passing touchdowns per game won more often and had a higher probability of making it to the playoffs vs. teams which did not. In fact if you were to rank teams by touchdown throws per game over each of the past 10 seasons, approximately 7 of the top 10 in each year had win rates greater than .500 and more than half made the post season [RK]. It is reasonable to believe that if your quarterback is making a lot of touchdown throws, you are scoring more points, and your team is probably winning more games. Winning more games increases the chances of making it to the post season. The value of singular players like Brady and Mahommes is undoubtedly real.

Thus, there is a connection between quarterback skill, winning games, and franchise value. If you are a small-mid market team, finding a star quarterback who can put up points is going to generate wins, increase media exposure and brand recognition nationally, and will get you additional revenue through playoff appearances, and increase ticket & merchandise sales. A natural question to ask, and the focus of this paper, then is: can we reliably make predictions about how many touchdowns an NFL quarterback will throw for in a given game? If so, we could run simulations to determine the offensive game plan which generates the most points, optimize personnel decisions, compute the likelihood of winning particular match-ups and more. We explore this question herein.

In section 2 we will the discuss the definition of phenomena and models. Section 3 will focus on mathematical models and features, while sections 4-7 will outline the process of supervised learning and necessary components. Section 8 will cover the model selection problem, while section 9 will outline some potential uses of our proposed model. Finally in section 10 we will offer concluding thoughts regarding the ability to model our chosen phenomenon accurately.

2 Phenomena and Models

A *phenomena* is anything occurring in the natural world which is of interest. Examples of phenomena may include: flight of an aircraft, physical processes like gravity, or the mortality rate of a disease. In our case the phenomena is NFL quarterback (QB) performance: given any player at this position, how well will they perform in a game? To both understand this phenomenon and predict it in the future, a scientist can work to create a model for it, that is, an approximation of reality. If our chosen phenomena was flight we may construct a model airplane to investigate the underlying physical properties which make flight possible or make predictions about how certain plane designs will fair. Right away we run into something of a problem with our chosen phenomena, what do we mean by "performance" and what could a model of QB performance look like?

As it stands now QB "performance" is rather vague, and a professional athlete seems to be very different from something like the man-made airplane / model airplane example. To move forward we need what are known as metrics and mathematical models. One could imagine a number of ways we could assess how well a quarterback plays in a game. Sportscasters and TV "analysts" frequently use criteria like: "how good he looked in the pocket", "how well he threw the ball" or "how well he read the blitz". However there is a noticeable problem here, these are ambiguous and up to interpretation. Alternatively, we could take a more mathematical approach i.e. we could base our evaluation on total yards thrown, completion percentage, QB rating etc. The advantage here is that these are well defined measurements, they are quantifiable, widely available statistics with no question about what they represent, these are called *metrics*. In general the *response measurement or output metric* of a phenomena is denoted mathematically as y, such that $y \in \mathcal{Y}$ the set of all possible output measurements. Since we already identified a relationship between passing touchdowns and competitiveness of a franchise (and its economic value as a direct consequence) in the introduction, we are going to choose $y =$ number of passing touchdowns as our metric.

To be clear a *passing touchdown* is defined as follows: At the beginning of each offensive play the quarterback is given the ball and he has the option to either "hand off" the ball to another player or throw it through the air to someone on his team running down the field. If the QB chooses to pass to another player, they successfully catch the ball, and that player is in the end-zone or runs it to the end-zone at the opposite end of the field this is a passing touchdown. This metric is unambiguous and can be assessed, the space of for the number of all possible touchdown throws becomes $\mathcal{Y} = \{0, 1, 2, 3 \ldots\}$ i.e. a non-negative integer count. And just like total yards thrown, completion percentage and QB rating, this measurement is widely available and recorded accurately during each game. Next we must construct an approximation of reality which is responsible for generating our y.

3 QB Performance as a Function

It will become useful at this time to employ a common metaphor from mathematics, that of the function machine. Our model can be thought of as a machine which takes in certain inputs, combines them together in some way via a rule, and outputs how many touchdowns a QB will pass for in a game. What forms might this machine take? One possible example is that passing touchdowns y are determined by "how good" the QB looks when passing. Once again, we see a problem similar to the one encountered during the discussion of measurements. This is ambiguous, even if we were able to gather data about "looking good" while throwing the ball that was relevant, we would never be able to construct an experiment to evaluate if this machine reliably outputs the correct number of touchdowns,

because it is fundamentally unclear. There exists a solution to this problem, mathematical models.

Mathematical models have the advantage that just like our output metric, their inputs, are quantifiable metrics. We limit our possible inputs and way we combine them to be only those which can be represented precisely with mathematics. This not only removes the ambiguity but makes it transportable to others who can evaluate if our model is any good at explaining the real world or making predictions. We begin our discussion with the assumption that our phenomenon is deterministic i.e.

$$y = t(z_1, z_2, \ldots, z_q)$$

Translated into words: the number of passing touchdowns is determined by several *casual drivers* (the z_i's) combined via some rule denoted t. Unfortunately limiting our choice of models to mathematical ones does not necessarily make the problem any simpler. To illustrate this point consider some reasonable casual drivers for our phenomena of interest:

$z_1 =$ weather conditions on game day

$z_2 =$ total yards thrown for during game

$z_3 =$ number of times the quarterback is sacked + hurried + pressured during game

$z_4 =$ quarterback sustains injury during the game yes/no

$z_5 =$ number of trips to the red-zone during the game

$z_6 =$ number of passing plays in the playbook

$z_7 =$ decision making ability of the offensive play-caller / head coach

$z_8 =$ "team and scheme" - the strategic make up of the team (run-first, defensive focus, spread offense etc.)

It is easy to see how we could measure these different casual drivers. However they all suffer from a similar problem, they are fundamentally unknowable to us. We cannot use z_2 = total yards thrown for during the game, because prior to the game actually happening we cannot know the answer. Additionally the function t which combines them can be arbitrarily complex and beyond our reach.

A workaround to this issue is to accept that we can never know t and the best that we can do is come up with another machine or function f which can approximate it. Further we can look for approximations of our casual drivers that are measurable prior to the start of the game. Expressed mathematically, we are going to select $x_1, x_2, \ldots x_p$'s to act as proxies for our $z_1, z_2, \ldots z_n$'s and denote a new rule f which accounts for the fact that t is beyond our understanding.

Using these new ideas we obtain:

$$y = f(x_1, x_2, \ldots, x_n) + \delta \text{ where } \delta = (t - f)$$

Our new collection of proxy inputs $\{x_1, x_2, \ldots, x_n\}$ are commonly referred to as *features, independent variables or predictors* and the δ term is known as "error due to ignorance", reflecting the fact that we simply do not know the true casual drivers $z_1, z_2, \ldots, z_n$ and so there is some information loss happening.

Below are the features we believe could act as proxies for some of the underlying casual drivers of our phenomena:

$x_1 =$ QB Career Completion Rate := total completions / (total attempts + sacks)

$x_2 =$ TD Rate := number of career passing touchdowns / (number of career attempts + sacks)

$x_3 =$ Red Zone Efficiency := career passing touchdowns in the red zone / total red zone trips

$x_4 =$ Rank of Offensive Line

$x_5 =$ Rank of Receiving Corps (Wide Receivers + Tight Ends + Running Backs)

$x_6 =$ Rank of Opposing Pass Rush

Each one of these features is a numerical entity and available before the game takes place. Further it is not hard to see how each could be a factor in the number of touchdowns a quarterback will throw for. If the player in question completes pass attempts at a higher rate than average, it stands to reason they will be better a passing for touchdowns (as TD pass must be a completed pass). If the starting offensive line is poor at protecting during pass plays (represented in the score value) it is unlikely that any chosen pass play will result in success. The rank / score being used in our features are numeric values assigned to players or groups of players on the team that are a function of their effectiveness. Ideally we would construct such functions in-house using widely available historical data on each relevant player. This is not a new idea, their are numerous companies already producing these kinds of scores as well as official NFL surveys of position coaches where active players are assigned a ranking [PF] [FBO].

Features x_1, x_2, x_3 are widely available statistics going back years, there should be no issue with their accuracy, meaning or computation for use in the model. Features x_4, x_5, x_6 are different, these inputs are functions themselves. How they are measured and what they mean is going to be determined by the function definitions used to generate them. Just like our model, all ranking methodologies will be approximations and thus, none will be absolutely "correct". We now offer a potential ranking function.

We would first begin by assembling a small group of experts to "grade" each player or group of players performance on every play from every game, every week, including past recordings going back some number of years. On every play, the player or group of players would be assigned a grade. The grade would be a number between $1 - 10$ with increments of .5. A score of 5 corresponds to "average" or expected performance, 1 a total failure (potential game ending blunder) and 10 an amazing play. Each player or player group would be assigned it's own unique rubric guiding how these scores are determined based on expert opinion, since expectations about what is a failure, average or amazing play will vary based on position and situation. Additionally our analysts will be assessing performance beyond what numbers alone may miss. One may question the subjectivity of this kind of approach, however we believe a solution of this general type is necessary. A grading function based entirely on player statistics will invariably miss the whole story.

Suppose for example we are attempting to grade offensive line performance, the play begins and the line crumbles to a four man rush giving up an awful sack. However a penalty down the field by the defense necessitates a replay of the down. What would have been a sack by the defense is not recorded in the stat line, but the offensive line performance should be penalized since they failed at their job, and only by chance did they have that failure removed from the record. A grade function based purely off on-field statistics may suffer from this kind of event while our expert analysts and rubric will be able to capture a more realistic portrait of performance whether or not a fortunate penalty was called or not. Further, already established reputation or notoriety will not have any effect on the scores being given out, since it is done and a play to play basis. If Tom Brady throws a bad pick six to end the game he will be evaluated the same way any other QB making the same mistake would be. The scores provided by our analyst team could be aggregated at the end of each game and updated every week ultimately becoming feature measurements x_4, x_5, x_6. Admittedly more work can be done in this area to improve how these measurements are made and collected.

There is no universally accepted way to rank / score players in the NFL. Different people have different opinions about what is important, and player performance is constantly in flux. The rankings or player scores will always be based on snapshots from past historical data, consequently the measurements these features provide will have a higher degree of imprecision compared to the more simple measurements like x_1, x_2, x_3. This imprecision will be a contributor to the overall error due to ignorance.

In our context, trying to predict a count for the number of passing touchdowns in a game, we believe error due to ignorance will be significant compared to the other sources which will be described later. One may find this belief surprising, surely there are many more features we could have selected in addition to the original six, and this is true. We could incorporate measurements like: the number

of practice reps for the quarterback of interest, time spent watching film in preparation for the game, is the game being played indoors vs outdoors, other information about the officials, historical play-calling and so on. In fact we could take it a step further, and by adjusting the specified time at which our g is to be run, aspects of the phenomena which were previously realized in the future can be considered known and incorporated (like for instance the weather).

Although we can reliably collect up huge amounts of data relating to player performance and game conditions, there are major causal drivers of our phenomena for which adequate proxies are not known. Earlier during our discussion of casual drivers we identified things like the strategic direction of the team and "intelligence" of the coaches formulating the game-plan and making in-game decisions. The QB and surrounding players on the field are not operating with complete freedom. Their actions, and thus performance is contingent on what plays are being called, and more broadly, what the team managers (coaches, general manager and owners) believe is the best path to victory. It is not uncommon for a QB to transition to new teams with different "schemes", or experience coaching/front office changes, and see a dramatic change in their performance metrics, including touchdown throws (see Ryan Tannehill, Matthew Stafford, Drew Brees). Individuals off the field, specifically the coaches, front office personnel and team ownership are directly influencing what 13 the players can (and cannot) do on the field in each game and across each season. We believe there is an inherent in-ability to account for these casual drivers. This fact will ultimately lead to a model whose predictive accuracy (RMSE) will be too large, and consequently ineffective at making future predictions. We will expand more on what this means in subsequent sections.

4 Learning from Data

We are still left with a significant problem, how do we best combine these features, i.e. what machine/rule/function are we going to use to take these inputs and actually produce the number of touchdowns a quarterback will throw for? To solve for the new function we will take an empirical approach known as supervised learning using historical data.

Supervised learning is defined as process by which we "learn" or find a function which best relates our chosen input measurements $\{x_1, x_2, \dots\ x_p\}$ to our desired output metric y. The desired function is found by examining historical data, sometimes called training data, with an algorithm of our choosing, which will infer the best candidate function based on some pre-determined criteria. Represented in mathematical notation, our model g is the result of running an algorithm $\mathcal{A}$, over a particular set of candidate functions $\mathcal{H}$ and collection of training data $\mathbb{D}$. In other words $g = \mathcal{A}(\mathbb{D}, \mathcal{H})$.

4.1 The Set of Candidate Functions $\mathcal{H}$

So far we have assumed that we can represent our phenomena as a mathematical entity:

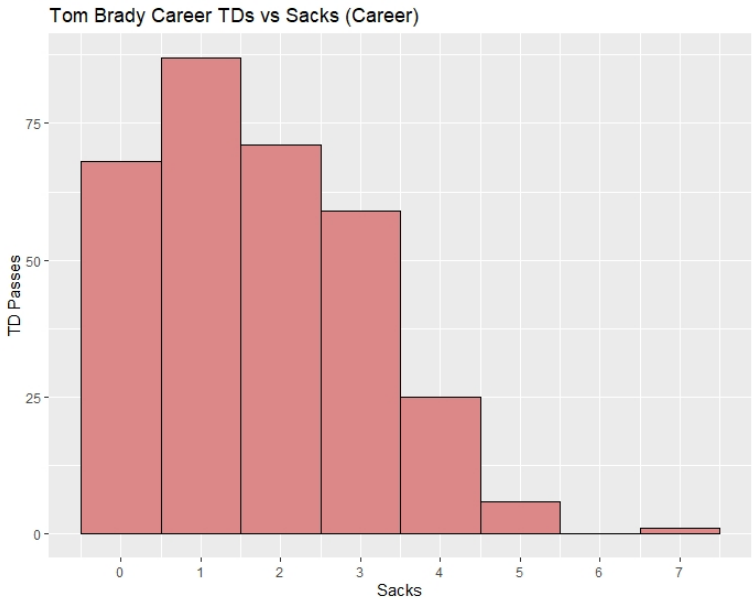
$$y = t(z_1, z_2, \ldots, z_n)$$

but we encountered a problem, the true casual drivers are unknown to us, which necessitated the following substitution:

$$y = f(x_1, x_2, \ldots, x_n) + \delta$$

Unfortunately, f like the original function t, can be arbitrarily complicated. The set of functions containing f is too large and ill-defined to reasonably investigate. Thus it becomes necessary to employ a simplification, that is to reduce the space of possible functions we will examine. This reduced space is the *set of candidate functions* denoted $\mathcal{H}$.

What the set $\mathcal{H}$ looks like will depend on a hypothesized relationship between our features and output metric. During our feature selection phase we supposed a relationship between passing touchdowns and strength of the opposing pass rush. Consider the below graphic. Each bin on the x-axis represents the total number of times Tom Brady was sacked in a game while the y-axis counts the total touchdowns thrown across all games constrained to that number of sacks.



The data seems to be validating our common sense belief in the relationship between touchdowns and pass rush strength. If the opposing pass rush is generating a lot of sacks, Tom Brady doesn't seem to be throwing many touchdowns and vice-versa. Further we might think to use a linear relationship since they are the simplest to start with. And you can also imagine a line tracing through the above histogram being a decent approximation of the relationship.

We could represent such a relationship mathematically as follows:

$$h(x_6) = b_0 + b_6 x_6$$

If we restricted ourselves to this feature x_6 only, our candidate set would be:

$$\mathcal{H} = \{w_0 + w_6 x_6 : w_0, w_6 \in \mathbb{R}\}$$

all possible lines representing the linear relationship between pass rush ranking and passing touchdowns. Furthermore we would suppose the existence of a $h^* \in \mathcal{H}$, the best possible line (or best possible approximation to f given our limited space $\mathcal{H}$) connecting these two variables:

$$h^* = \beta_0 + \beta_6 x_6$$

where b_0 and b_6 are the best "true" values of the coefficients. Our search algorithm (still to be determined) would seek to approximate this h^*, and that approximation would ultimately become our model denoted g.

Recall however, we selected more than one feature for use in our model, naturally we would want to include all of them in the same way. We could generalize our $\mathcal{H}$ to be not one of the best line, but of the best hyper-plane (a fancy math term for a plan in higher dimensions):

$$\mathcal{H} = \{\mathbf{b} \cdot \mathbf{x} : w_0, \ldots, w_6 \in \mathbb{R}\} = \{w_0 + w_1 x_1 + \ldots + w_6 x_6 : w_0, \ldots, w_6 \in \mathbb{R}\}$$

In this case, passing touchdowns is the result of a weighted sum. The value of each feature is "worth" some amount in determining the prediction of passing touchdowns. For example $\hat{b_6}$ the weight of feature x_6 = rank of opposing pass rush, will tell us for every unit change in the rank of opponents pass-rush how much the number of passing touchdowns should change. Supervised learning will search through $\mathcal{H}$ and return a best guess at the value of each $\mathbf{b}$ and return a $\hat{\mathbf{b}}$ which tells us approximately how much changes to each feature measurement contributes generally to our response metric of interest. To run this "search" on our $\mathcal{H}$, we need training data to analyze.

4.2 Training Data $\mathbb{D}$

Let's pretend we are going to compute our own custom stat line for specific quarterback in game we just watched. The stat line is going to consist of our computed feature measurements $x_1, \ldots, x_6$, as well as the number of passing touchdowns our QB in question threw for, y, during the game. We could represent those values in vector format as follows:

$$\left[x_1 = .61 \quad x_2 = .042 \quad x_3 = .77 \quad x_4 = 4 \quad x_5 = 13 \quad x_6 = 22 \quad | \quad y = 3\right]$$

In words, going into the game our QB had: a career 61% completion rate, 4% TD rate, 77% Red Zone efficiency, an offensive line ranked 4th, receivers with a combined rank of 13, played a defense with a pass rush ranked 22, and threw for 3 touchdowns in the game. Further, imagine repeating the above process for every game our QB of interest has played in over their career. Assuming a season is 16 games (ignore the fact that as of 2021 it is now 17 games) and that our quarterback started in all of them, and that the average career of an NFL quarterback is around 3 years, we are looking at a matrix of $n = 3 \times 16 = 48$ rows and 7 columns. This could be organized in the following way:

$$\begin{bmatrix} 1 & x_{1,2} & x_{1,3} & \dots & x_{1,7} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{48,3} & x_{48,3} & \dots & x_{48,7} \end{bmatrix}, \begin{bmatrix} y_1 \\ \vdots \\ y_{48} \end{bmatrix}$$

We add a column of 1's and separate the feature measurements into a matrix $\mathbf{X}$ and a corresponding output vector $\mathbf{y}$, for computational convenience later. Now instead of focusing on one quarterback, imagine looking at each game, every week over the past 10 years, and constructing $\mathbf{X}$ using this process. The columns in our matrix would remain the same, however we would be expanding our row count, denoted n, to be in the neighborhood of $n = 32 \times 16 \times 10 = 5120$. A comment about the feasibility of constructing $\mathbf{X}$ is necessary at this time.

Note that the first 3 columns of $\mathbf{X}$ are individual quarterback statistics, which are available by player and game going back years, in a variety of formats [PFB]. Collecting up the necessary data to fill out the first 3 columns of our matrix should not be difficult. Recall that features x_4, x_5, x_6 are the rankings of particular players or units of players leading up to the game. To compile the data needed to fill these columns, we would need access to the appropriate rankings as they existed at the time leading up to each game which occurred years in the past. If we wished to use our own in-house functions to generate these rankings, this would require retro-actively computing the data needed to fill columns $4, 5, 6$. Provided our ranking functions take accessible historical data as inputs (which they would), these computations should be possible, although the process would be time consuming and potentially costly.

A potential alternative is to pay for access to existing player & team scores. As noted previously there are a number of existing firms already producing this kind of data and have been for years. For example Pro Football Focus has been producing a variety of player and team rankings since 2007. By paying the necessary subscription fees, we could access their historical data and use it to fill out $\mathbf{X}$ as needed. The downside here is that we would not be able to explicitly define how the feature measurements in these cases are being gathered, as the functions behind their generation could be proprietary and unknown to us. Ultimately, regardless of which path is chosen, gathering the necessary data to construct $\mathbf{X}$ is possible, it is just question of how much money and time one has.

The compilation of historical values stored in $\mathbf{X}$ and the corresponding $\mathbf{y}$ we have constructed above are known as *training data*, denoted $\mathbb{D} =< \mathbf{X}, \mathbf{y} >$. With our $\mathbb{D}$ and $\mathcal{H}$ in hand we are almost ready to model our phenomena. One final ingredient is necessary to carry out the supervised learning process, we need an algorithm $\mathcal{A}$.

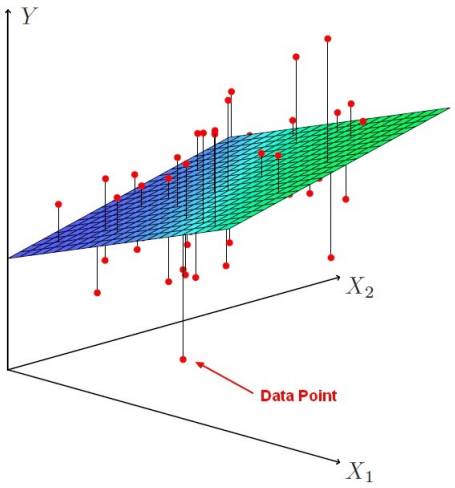
4.3 Algorithms $\mathcal{A}$

With both $\mathcal{H}$ and $\mathbb{D}$ squared away, we only need an algorithm denoted $\mathcal{A}$ to complete the supervised learning process. An *algorithm* is a well defined sequence of finite steps for solving a particular problem. In our case the problem is finding our model g an approximation to h^*, which is an approximation to f, itself a simplification of the true unknown t. Since our candidate set was restricted to be that of all linear relationships $\{w_0 + w_1x_1 + \ldots + w_px_p : w_0, \ldots, w_p \in \mathbb{R}\}$ between our phenomena of interest and selected features, our algorithm will seek to return the best guess $\hat{\mathbf{b}} = \{\hat{b_0}, \ldots, \hat{b_p} \in \mathbb{R}\}$ of the coefficients $\mathbf{b} = \{w_0, \ldots, w_p \in \mathbb{R}\}$. There are many choices of $\mathcal{A}$ such as minimization of mean absolute deviation and ordinary least squares (OLS). Herein, we use OLS which is optimal for estimating the conditional expectation of y given the x's, has a closed form solution and is computationally feasible.

4.3.1 Ordinary Least Squares

To understand the basics behind OLS, we will limit our discussion temporarily to 3-space.Imagine we believe only two features x_1 and x_2 best account for our y. We could graph our training data as points in 3-dimensional space, as seen in the below image. Our specified $\mathcal{H} = \{\text{all linear relationships}\}$ would represent all possible planes in $\mathbb{R}^3$. Our algorithm run on our data set, $\mathcal{A}(\mathbb{D}, \mathcal{H})$, should infer for us the "best" plane. What we mean by "best" is encoded mathematically in what is known as an objective function. An *objective function* is a mathematical relation that our algorithm of choice seeks to minimize (or maximize), the result of this optimization process being the "best" choice given our constraints. In the case of OLS, the objective function is the SSE or sum of squared errors.

To get a better handle on OLS , an example of some of our imagined data as well as a plane in $\mathbb{R}^3$.



Let's breakdown this picture: the red dots correspond to the historical y_i's in our training data and the plane itself is a potential model candidate. The point in the plane below each red-dot would represent the prediction the model would give based on the provided feature measurements, denoted $\hat{y}_i$, and the line connecting the two points is the error $y_i - \hat{y}_i$. Imagine computing this error for every $y_i \in \mathbb{D}$ and associated $\hat{y}_i$ and squaring it, then taking the sum over all the squared errors. We would have computed the sum of the squared errors aka the SSE $= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ for this particular plane.

From the point of view our algorithm, OLS is trying to infer from our data, the plane which has the minimal SSE. In mathematical notation:

$$\hat{\mathbf{b}} = \text{argmin}_{\mathbf{b}} \, ||\mathbf{y} - \hat{\mathbf{y}}||^2$$

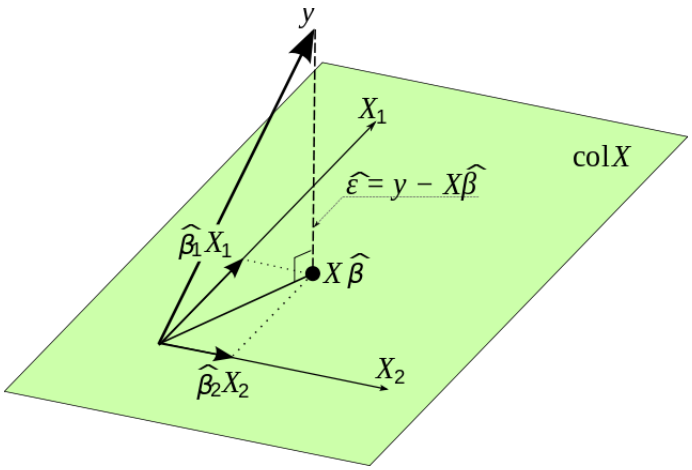
This makes sense, if we are assuming a plane is the best way to model our phenomena, we would like to find such a plane which results in the smallest distance between the predictions it would make based on our feature measurements and what the actual measured response was. In other words, our best model (given our current assumptions about it's form encoded in $\mathcal{H}$) is the one which results in the smallest aggregate of the vertical lines in the above image.

It turns out that the plane resulting from the minimization of SSE, would be one such that each prediction $\hat{y}_i$ is the result of an orthogonal projection of y_i onto a plane defined by a linear combination of our features. In the context of mathematics, orthogonal is just a fancy word to mean of or involving right angles. If each prediction is the result of an orthogonal projection, it means the line representing the error $y_i - \hat{y}_i$, forms a right angle with the plane - and thus forms the shortest path between each y_i and the corresponding prediction $\hat{y}_i$.

Recall, earlier on we chose to represent our $\mathbb{D}$ in terms of a matrix of feature measurements $\mathbf{X}$ and corresponding vector containing the outputs $\mathbf{y}$, this was intentional, not only for organizational

purposes but to leverage the power of linear algebra. OLS makes use of a combination of multivariable calculus and linear algebra, applied to our data, to find our desired orthogonal projection. This procedure generalizes to any space in $\mathbb{R}^n$ where $n \geq 2$.

An example of an orthogonal projection is illustrated in the image below:



Notice, each prediction can be represented algebraically as linear combination of our features :

$$\hat{y}_i = \hat{b}_0 + \hat{b}_1 x_1 + \hat{b}_2 x_2 = \mathbf{x}_{i.}^{\intercal} \hat{\mathbf{b}}$$

Since we have a collection of training data, after this orthogonal projection process we would have a collection of these linear combinations, one for each y_i, which is easily encoded in matrix algebra as follows:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{b}}$$

Our supervised learning process, has returned a model g which is a plane in $\mathbb{R}^{p+1=7}$. Mathematically we have the following:

$$\mathcal{A}(\mathbb{D}, \mathcal{H}) = g \Rightarrow g(\mathbf{X}) = \hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{b}}$$

The final output g is the result of the modeling process; it is our "model" and can be used to predict number of TD's for future games. We will see an explicit example of this operation later.

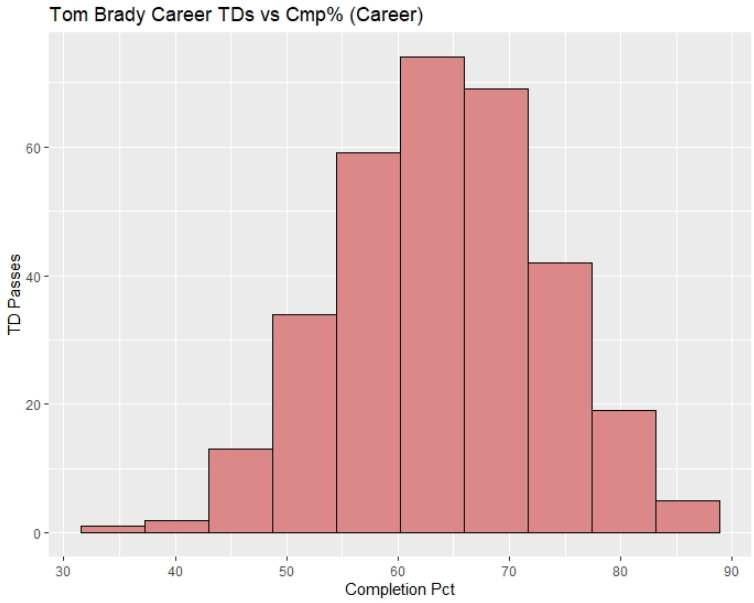
4.4 Two More Sources of Error

After understanding the need for proxies to the true casual drivers, we realized that even the new function f could be too complex for us to handle, and if we wished to have any attempt to find a mathematical model, the space over which we "searched" for one needed to be simplified, enter $\mathcal{H}$. If we selected OLS, $\mathcal{H}$ was the set of linear relationships, which is simpler than the space of functions which contained f. Further we supposed the existence of h^* the best function in $\mathcal{H}$ which could represent f.

It is at this point we incur the second kind of model building error *mis-specification* represented by:

$$y = h^*(x) + \underbrace{(f(x) - h^*(x))}_{\text{misspecification}} + \underbrace{\delta = (t(z) - f(x))}_{\text{ignorance}}$$

Who is to say that our choice for $\mathcal{H}$ is any good? Naturally, choosing a $\mathcal{H}$ that is not rich enough will induce additional error during the construction of our model. To see what we mean let's return to Tom Brady's career stats for a moment and check touchdown throws organized by completion percentage leading into the game:



Unlike the sacks / touchdowns graphic viewed earlier, there appears to be a non-linear relationship between completion percentage and touchdowns. One may imagine trying to use a straight line here would induce significant error in predictions if we regressed on completion percentage alone. A polynomial curve would probably yield a better fit. This would necessitate a change in our $\mathcal{H}$ to include more complicated curves, specifically it would require the inclusion of feature interactions and transformations. Substantial effort has been made to model athletic performance in other areas which could be considered similar to the NFL, the Olympic games being the most notable. Researchers have been making use of a wide variety of different more involved candidate sets to try and model how elite athletes will perform during different events like running & swimming. Selections for $\mathcal{H}$ range from higher order polynomials [Galvan et al., 2018] to linear mixed models [Avalos et al., 2003]. It is unlikely that the simple linear relationship we are assuming can capture the phenomena of professional NFL quarterbacks performance with enough accuracy. The rule which governs the combination of our features is likely more involved, and even though $\mathcal{H}$ is supposed to be a more tractable space of functions, it can still be complicated. Such a situation would require a change to our process, namely we would need to use *machine learning*, supervised learning on a more complex $\mathcal{H}$, to infer our model

g. Due to deficiencies in our chosen $\mathcal{H} = \{\text{all linear relationships}\}$, there is likely to be a significant difference between f and the best candidate model h^*, and thus a difference between our g an h^* as a result.

Finally, there is a third source of model building error *estimation error*. Regardless of features, and selected $\mathcal{H}$, a gap between the output g from $\mathcal{A}(\mathbb{D}, \mathcal{H})$ and the supposed $h^* \in \mathcal{H}$ will always exist. In general the errors due to ignorance and misspecification confuse the algorithm, the amount of historical data available may not be sufficient, and the chosen algorithm to infer our model could be inaccurate or improved upon. Any model g therefore will never match h^* exactly. Specific to our phenomena of interest, this is likely to be another source of error in model, but will likely be overshadowed by the error due to ignorance. Even though we have access to a large amount of historical data, we supposed earlier that our $\mathcal{H}$ was insufficient. It stands to reason that a better selection of a candidate set $\mathcal{H}$ will be more involved and hence require an algorithm that may be more obscure and thus less well understood, or perhaps may not even exist.

We can sum up the final relationship between our model, phenomena and three sources of error mathematically as:

$$y = g(x) + \underbrace{(g(x) - h^*(x))}_{\#3 \text{ est error}} + \underbrace{(f(x) - h^*(x))}_{\#2 \text{ misspecification}} + \underbrace{t(z) - f(x)}_{\#1 \text{ ignorance}}$$

5 Error Metrics

Despite our best efforts, any model we are going to produce will have some associated "noise" or error with it's output. Recall the famous aphorism attributed to Box and Draper: "all models are wrong, but some are useful". The question becomes, by how much is our model wrong? Is there a way we can evaluate our model and check its performance before using it to make future predictions? The answer is to employ prediction error metrics and a process known as model validation.

A *prediction error metric* is a way of quantifying how accurate your models predictions are. In terms of our chosen algorithm OLS, there are a number of error metrics that can be used to evaluate how accurate the model is performing based on data already available to us. The two most often used are: R^2 and RMSE. As mathematical formulas these are represented by:

$$R^2 = \frac{SSR}{SST} = \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$\text{RMSE} = \sqrt{\frac{1}{n-p-1}SSE} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \hat{y_i})^2}$$

RMSE is the simpler of the two to understand from it's mathematical definition. Recall the objective function our algorithm OLS, was using to locate our model - the SSE or sum of squared errors. Our RMSE is the average SSE across all data points, but square rooted. Why the additional

process of taking the square root at the end? It is so the returned error is in the same units as our response metric. To a data scientist this would mean RMSE is "interpretable". RMSE is a bit more intuitive and applicable when compared to R^2 which is "un-interpretable" because the units it reports the error differ from the response metric.

Consider the following example: We fit our model and compute the RMSE which turns out be .211. Suppose further we use our model to make a prediction (more to come on this later) and we get a $\hat{y} = 3.44$, that is we are predicting our quarterback of interest, based on the relevant feature measurements, will throw 3.44 touchdowns in the game in question (naturally we would round this value since our $\mathcal{Y}$ is a subset of the non-negative integers). Using the computed RMSE and Central Limit Theorem we could provide a 95% confidence interval for our prediction, that is the true y or number of passing touchdowns our QB will throw for is in the range of $[3.44 \pm 1.96 \times .211] = [3.03, 3.85]$, so between 3 and 4 touchdowns, not bad. If it were possible to construct a model with this RMSE we are effectively saying it is possible to predict the number of passing touchdowns within an range of 1. As you might imagine a RMSE closer to zero, translates to a lower average error and thus we might expect a more accurate performance. We could also evaluate our model using R^2.

R^2 is a bit different from RMSE in a few ways. As already mentioned the unit's of R^2 are "un-interpretable", so what is it really telling us? In words, it is measuring the proportion of the total variance "explained" by our g fit to the data, or: $1 - \frac{SSE}{SST} = 1 - \frac{\sum(y_i - \hat{y_i})^2}{\sum(y_i - \bar{y})^2}$. The utility in this metric lies in the fact that it is trying to quantify performance of your model vs. the so called null model. What is meant by "null model"? In general the *null model* is the simplest model possible which could be constructed from the data, which depends on context. In the world of OLS, the null model, denoted g_0 is given as $\bar{y}$, the average of our on hand response measurements which is about 2.55. The metric R^2 is capped at 1, and generally is between 0 and 1, however it can go negative. A negative R^2 would mean your fit g is performing worse than the null model, a sure sign you need to re-think your assumptions. Despite some of the utility in R^2, for our purposes we will choose to focus on RMSE.

Even with the availability of these metrics, there is more to do before making predictions. Fitting our model with just the supervised learning process as outlined, and coming out with a g and corresponding RMSE = .211 may lead us to believe we can predict the number of touchdown throws to a margin of ± 1, but we are not the NFL Draft Kings yet. It turns out these error metrics alone are not enough.

6 Overfitting & Underfitting

Unfortunately, our chosen error metric, the RMSE (as well as the R^2) can be gamed, we can make our models' RMSE as low as we want. In other words our model can be "overfit". To understand how this happens, let us return to our matrix of feature measurements $\mathbf{X}$.

As noted previously, we selected 6 features or predictor variables for our model, whose measurements were organized into matrix $\mathbf{X}$. An individual column vector of $\mathbf{X}$, say $\mathbf{x}_{\cdot 2}$ would be a vertical list of n (approximately 5000) TD % computations, each corresponding to an individual quarterback before a specific game. Now suppose we have some more feature measurements we want to add to our training data, i.e. more column vectors representing different things we believe may influence the number of passing touchdowns. We already identified some within Section 3. Naturally, one would expect the RMSE to go down, after we re-fit (re-run the algorithm) on the new data set with these additions. Our model should become more accurate as we provide it with more relevant information about the phenomena in question. In fact RMSE does go down as the number of features goes up. Here is the problem: we could achieve this same result by adding column vectors full of random numbers to $\mathbf{X}$.

Each additional column (corresponding to a new "fake" predictor) of random numbers will slowly improve whatever error metric we are using to evaluate our model as it currently stands. And as the number of features p approaches the number of data points n we are *overfitting*, that is artificially making our error scores look better. This can happen with any $\mathcal{A}(\mathbb{D}, \mathcal{H})$. Recall in our world of OLS we arrived at our predictions via orthogonal projection. In effect we were "smashing" down our y_i's into a dimension smaller than the one they originally inhabited. This "smaller world" was defined by our predictor variables. By adding columns of random junk to our data set we are effectively stealing from other dimensions to artificially make this smaller world more like the one the y_i's initially belonged. It is like looking over at someone elses' paper during a test and taking their answers (but we don't even need the answers to be correct to make our grade better!). The fact that overfitting is possible, means we have two problems on our hands. The first is that we can potentially be dishonest describing how good our model is given the current model building process. Second, regardless of intentions, an overfit model will suffer from *generalization error*: it will be poor at making future predictions (even though on paper it has a "good score").

Specific to our phenomena and model of quarterback performance, it is unlikely that our model would suffer from overfitting. Our initial estimate of n was somewhere in the neighborhood of $5,000$ and this was a conservative guess. Suppose we increased our feature count from $p = 6$ to some other value. We noted previously that there are very many more predictors we could include in the model, how close can we get to $n = 5000$? Consider the following:

- replace the original 3 individual quarterback features (Completion Pct., TD Pct., and Red Zone Eff.) with a "advanced"[PFB] QB stat line $\rightarrow +33$
- include predictors which capture the physical and mental attributes of the quarterback (height,weight, 40 yard dash, pass release time, individual intelligence and problem solving ability) $+10$
- replace the three aggregate rankings for O-Line, Opp Pass Rush and Receivers with rankings for every individual player on the roster $\rightarrow +32$
- include scores for the coaching staff $\rightarrow +8$
- include measurements from practice leading up to the game (amount of time watching film, time spent in the weight-room, physical therapy etc.) $\rightarrow +10$
- include stadium indoor / outdoor with measurements for weather conditions (temperature, pressure, wind speed, precipitation etc.) $\rightarrow +10$
- include features which capture information about the officiating crew (how many times do the officials in the present game tend to call certain penalties which favor the offense?) $\rightarrow +10$
- include features which capture information about the crowd (total attendance, crowd noise etc.) $\rightarrow +10$
- include features which capture off the field information about key players (how many times did our QB or star wide receiver go out drinking during the week?) $\rightarrow +10$

Even if it was possible to actually measure and collect up this information accurately we are at $p = 133$ features, or $\approx 3\%$ of our assumed n. Our position is that it is far more likely that our model would suffer from *underfitting*, that is we will possess more n than good features, and we won't be learning as much as we could from the data at our disposal. Although an underfit model is not necessarily representative of dishonest practices like an overfit model can be, it is still not desirable. Ultimately we want our approximation of reality to be as accurate as possible so we can make predictions about the future. In the context of throwing touchdowns and winning or losing NFL games, millions of dollars or more can be at stake.

Despite the present challenges with overfitting / underfitting, we can be more honest about evaluating our model, and get a better handle on generalization error. To do this we must alter the supervised learning process slightly to obtain "out of sample" error.

7 Model Validation

The process used to construct our mathematical model to get "honest" error metrics is known formally as *model validation*. Instead of running our algorithm on the entirety of $\mathbb{D}$, as we will did originally, we must partition our historical data set into two distinct sets known as training and testing data denoted $\mathbb{D}_{train}$ and $\mathbb{D}_{test}$. To avoid any underlying patterns associated with time this is done at random, based on a predetermined proportion $\frac{1}{K}$ (K is usually 10 or 5), which represents what percentage of the data will go into the test set. In effect we are going to pretend temporarily that our actual data set $\mathbb{D}$ is smaller, and fit our model based only on the information in $\mathbb{D}_{train}$, call this model $g_{\text{train}} = \mathcal{A}(\mathbb{D}_{train}, \mathcal{H})$. Further we are going to pretend that the information contained in $\mathbb{D}_{test}$ is new from the future. We will process $\mathbb{D}_{test}$ through g_{train} and compute an RMSE based only on how well g_{train} makes predictions on the "new" test data. This RMSE will be an "out of sample" error metric. The term *out of sample* is used because the error is being computed based on data we did not use to build g_{train}. This is in contrast to the original, *in sample* RMSE computation which was based on the model g fit to the entire set $\mathbb{D}$. We can compute an out of sample version of any of the performance metrics previously discussed. Our new RMSE looks like:

$$oosSSE = \sum_{i=1}^{n_*}(y_* - \hat{y_*})^2 \Rightarrow oosRMSE = \sqrt{\frac{1}{n_*} oosSSE}$$

Where n_* is the total number of data points in $\mathbb{D}_{test}$ and y_* / $\hat{y_*}$ represent the response measurement in $\mathbb{D}_{test}$ and the corresponding prediction respectively. These new out of sample error metrics will represent an upper bound on how we will actually do predicting future values based on new data, i.e. they will represent a cap on our generalization error. The "real" out of sample error should be better when the model is used in the field.

When the data $\mathbb{D}$ was partitioned and g_{train} fit in order to get honest error metrics, we essentially threw out some of our data. Remember g_{train} was only fit on a percentage of the entire set $\mathbb{D}$. This was necessary to get the honest errors, but we don't want to leave usable data behind. Therefore, after the out of sample errors are computed, the actual model we will use to predict future touchdown passes, which we can call g_{final}, is actually g fit using the entire (non-partitioned) $\mathbb{D}$ as discussed originally.

It is important to mention that this requires a significant assumption called *stationarity*. This means we believe that as we move through time the underlying casual drivers and t which represent our phenomena will remain the same, ensuring that our proxies with f will also remain representative. This is critical because when we partitioned $\mathbb{D}$ we were essentially conducting a thought experiment which involved assuming a time in-variance - it doesn't matter if we pretend some of our $\mathbb{D}$ is actually "future" data because the factors and rules governing how the phenomena behaves do not change with time. Also, without stationarity the entire modeling enterprise becomes something of a waste of time,

with t and $z_1, z_2, \ldots$ continuously changing you cannot begin the process. In the current context, we believe the phenomenon of QB performance and chosen response metric (passing touchdowns) are in fact stationary. Although we can compute an upper-bound on how well our desired model may perform in the future, there is still more to be done before we can begin making predictions.

8 Model Selection

In the validation process described above there exists several problems. First, there is only one comparison. Suppose it appears our model is overfit i.e. in sample RMSE is much lower than out of sample RMSE. There is nothing we can really do, after the performance is checked on $\mathbb{D}_{test}$, the model cannot be adjusted honestly anymore in the current framework. Second, $\mathbb{D}_{train}$ and $\mathbb{D}_{test}$ are now random variables since we are randomly selecting the data in $\mathbb{D}$ based on the specified K parameter to populate each set. This means it is possible to construct a $\mathbb{D}_{test}$ which could favor (or disfavor) our model unfairly just as a matter of chance.

Finally, there are other algorithms, in place of OLS, we could have run on the data using different combinations of features, including interactions and transformations. It would have been perfectly legal in a modeling sense to consider $log(x_i)$ in place of x_i, a feature transformation, if it appeared that there was a non-linear logarithmic relationship between x_i and passing touchdowns. Additionally, $x_i x_j$, a feature interaction, would also be a possible addition. Allowing for transformations and interactions dramatically increases our set of possible features, but locating those which add predictive value without over-fitting is a difficult problem. Allowing feature interactions and transformations expands our set of possible features beyond what was discussed in previous sections, increasing the number of potential models that could be created using OLS. Furthermore, other algorithms such as K-Nearest Neighbors, Bagged Trees or Random Forest could have also be run in place of OLS, leading to even more model possibilites.

All of the potential models we could have created would have been approximations of reality, and thus no specific one would be "correct", but each could have been potentially useful. Further, our selection process is highly variable due to the randomization of the testing and training sets. How, therefore, is g_{final} actually selected? This is what is known as the "model selection problem". Much has been written about this topic and possible solutions. We will focus on one method in particular: identifying and selecting the model with the lowest out of sample error. To identify our actual g_{final} we will employ a process known as k-fold cross validation. Instead of construction two partitions $\mathbb{D}_{train}$ and $\mathbb{D}_{test}$ we make three: $\mathbb{D}_{subtrain}$, $\mathbb{D}_{select}$, and $\mathbb{D}_{test}$.

Suppose we have several different model candidates $g_1, g_2, g_3, \ldots g_k$ to choose from. To begin we

specify two K parameters, K_{test} and K_{select}. The parameter K_{test} is just like before, it tells us what proportion of the original data will be randomized into $\mathbb{D}_{test}$ while K_{select} is how much will randomly selected for placement into a new split $\mathbb{D}_{select}$. We first create the partition $\mathbb{D}_{test}$ and place it to the side, leaving us with $\mathbb{D}_{subtrain}$. Using $\mathbb{D}_{subtrain}$ and K_{select} we construct $\mathbb{D}_{select}$.

Using $\mathbb{D}_{subtrain}$ we fit each of our candidate models and then assess their errors on $\mathbb{D}_{select}$. We repeat this process k times (the k folds) ensuring that each data point in $\mathbb{D}_{subtrain}$ is represented in $\mathbb{D}_{select}$ one time. At the end of this process the model g_* with the best performance across all the folds is selected as our g_{final}. To get the best estimate of our out of sample error we build g_{final} on $\mathbb{D}_{subtrain} \cup \mathbb{D}_{select}$ and assess the error on $\mathbb{D}_{test}$ which was been in "storage" up until this point.

Regardless of the OLS model we proposed above, we should likely use the Random Forest algorithm which would result in higher predictive accuracy and it would allow us to not worry about specifying transformations and interactions of our proposed features. The one downside of employing this algorithm is that we would lose the ability to explain how our model produces predictions which was possible in our ordinary least squares model proposed herein.

9 Making Predictions

With our out of sample error scores and g_{final} we are ready to make predictions. From section 4.3.1 we know the output of OLS on $\mathbb{D}$ will look like:

$$\mathcal{A}(\mathbb{D}, \mathcal{H}) = g_{\text{final}}$$
$$g_{\text{final}} = \hat{\mathbf{y}} = X\hat{\mathbf{b}}$$

Where $\hat{\mathbf{b}}$ is a column vector of real numbers, the weights for our predictors, which result in the plane with the minimum sum of squared errors.

With the bulk of the work complete (fitting and evaluating the model properly), computing predictions at this point is an exercise in arithmetic. Suppose Aaron Rodgers, the quarterback for the Green Bay packers, is going to play in a big game against the rival Minnesota Vikings this weekend, and we want to know how many touchdowns Aaron will likely pass for. We collect our necessary feature measurements, encoded in the vector $\mathbf{x}_*$ which looks something like:

$$\begin{bmatrix} x_1 = .67 & x_2 = .049 & x_3 = .87 & x_4 = 3 & x_5 = 9 & x_6 = 27 \end{bmatrix}$$

The predicted number of touchdown throws is thus:

$$g_{\text{final}}(\mathbf{x}_*) = \mathbf{x}_*\hat{\mathbf{b}} = \hat{b_0} + \hat{b_1}(.67) + \hat{b_2}(.049) + \ldots + \hat{b_6}(27) = 3.31 \approx 3$$

It is not necessary though to make predictions in this fashion, one at a time. We could have easily collected up the necessary measurements for every QB starting this weekend. Instead of a single row

vector, we would have a collection of row vectors, or a matrix $\mathbf{X}_*$ that could be fed through g_{final}, and instead of an output of a single value, we would have a vector $\hat{\mathbf{y}}_*$ containing the number of predicted passing touchdowns for each quarterback (each row in $\mathbf{X}_*$). If we truly had a model like g_{final} with an oosRMSE that was tight enough, it could be used in multiple different ways.

This model building adventure began based on assumptions about running an NFL franchise and generating wins. We hypothesized a connection between the number of passing touchdowns and wins / playoff appearances. It follows that people involved in running an NFL team, in the front office or part of the coaching staff (who want to win and believe in this theory about where wins come from), could make use of the model. Suppose you are the head coach of a team with three quarterbacks on the active roster, and one of them isn't Tom Brady, it may be unclear who to start. This model would allow you to potentially make a more informed guess as to who should play. Run all three through the model, if the results are $\hat{\mathbf{y}}_* = [1, 1, 4]$ the guy projected to throw for 4 touchdowns might be your best bet. This is not the only use case.

Suppose we wanted to investigate how changes to other players or units on the team affected how many touchdowns are thrown for (and thus our chance to win games). Holding our QB data constant (measurements for features x_1, x_2, x_3) we could make alterations to other predictors and analyze the results. For instance what happens if the rank of our offensive line changes, but our QB and receiver inputs remain generally the same? How would we fair in throwing touchdowns, again assuming no real changes to our quarterback and his receivers, against pass rushes of different ranks?

$$\Big[x_1 = constant \quad x_2 = constant \quad x_3 = constant \quad x_4 = L \quad x_5 = constant \quad x_6 = S\Big]$$

Conducting simulations where we adjust L and S (the rank of our offensive line and rank of opposing pass rush respectively), might provide valuable information as to whether or not we should expend draft picks or money in free agency on improving our offensive line. A similar exercise could be done to investigate how important our receiving corps is, and many more examples are possible. In fact with changes to the predictor variables being used, the simulations could become more focused. Remember in section 6 when discussing how to potentially increase our features, one of the suggestions was to use ranks for every spot on the roster vs. aggregates like offensive line as a unit. If we were to make such a change, simulations using the model could inform the value of adding or removing individual players to our team, which could further inform draft or free agency strategy.

A model like the one described could also find significant use outside of people directly affiliated with an NFL team. Fans of the game, particularly those trying to make money based on the outcomes could make use of the model described. It is impossible to watch an NFL game and not know about Draft Kings or Fan Duel (in additional to several other competitors). Sports betting and fantasy sports

has exploded in recent years, and large amounts of money is exchanged through these websites. A person in possession of our model could clean up in fantasy football.

It is important to note that most of the predictions being made in a real world setting should be *interpolation*. That is the predicted number of touchdown passes are ultimately constrained to be within the combined domain of existing historical data points, $g(\mathbf{x}_*) \in [min(x_1), max(x_1)] \times [min(x_2), max(x_2)] \times \ldots \times [min(x_6), max(x_6)]$. It is unlikely that a quarterback will come along with individual statistics that stray far from existing minimums and maximums historically. It would also not be difficult to check for this beforehand. This is in contrast to *extrapolation* where the model would seek to make predictions using input data beyond the range of the historical on-hand $\mathbb{D}$. Such a model would be mercurial and unreliable.

This entire discussion about the different use cases for g_{final} pre-supposes a model with a certain level of accuracy, that is a oosRMSE within a certain range. We already spoke about the pitfalls of constructing this type of model: we probably won't learn as much as we can (underfitting), there will be issues with selecting the best candidate set $\mathcal{H}$ - reflected in significant misspecification error. It is our belief that oosRMSE will always be large. This would translate to confidence intervals which are too large to be informative.

Remember the example from section 7, and suppose our oosRMSE (instead of in sample RMSE) is computed to be .211, with a $\hat{y} = 3.44$, and 95% CI = $[3.03, 3.85]$. Our prediction range is thus approximately $3 - 4$ touchdowns. This is useful knowledge in reality, the range is essentially within one touchdown and we know that oosRMSE is conservative, so we are likely to perform even better. However, suppose we alter the example and say that our oosRMSE is something like .89, our range expands to $[3.44 \pm 1.75] = [1.69, 5.19]$. This is much less useful, and unfortunately it is our belief we will more often than not have oosRMSE's which are too great for practical use.

10 Conclusions

In section 3 we mentioned a lack of proxies for certain casual drivers which are key to our phenomena, namely those associated with strategic decision making inside and outside an individual game. We concluded this would lead to increased error due to ignorance, and combined with misspecification error, a model with an oosRMSE too large to be effective in a real world setting. We would like to close out now with a concrete example which hopefully illustrates this point.

Prior to the start of the 2018 season the Seattle Seahawks hired Brian Schottenheimer as their offensive coordinator (the coach responsible for the offensive strategy and play-calling). Schottenheimer, completely re-vamped the teams offense, and as a result the Seahawks finished with an offense ranked

6 and 8 respectively, in terms of points scored, at the end of the 2018 and 2019 seasons. By the start of 2020 things were looking even better. Quarterback Russell Wilson was having a career season. In particular, over the first 8 games, he passed for a total of 28 touchdowns. To put this is context, he was on pace to break the record for number of touchdown throws in a single season, 56, held by hall-of-famer Peyton Manning. Wilson's other key stats by mid-season 2020 were also at career highs. The Seahawks as team, despite having one of the worst defenses in the NFL, seemed unstoppable and poised for a Super Bowl run. However something strange happened. In the back half of the season, Russell Wilson's performance took a dramatic turn. In the final 8 games he threw for only 12 more touchdowns, and his other passing statistics also took a dive. The Seahawks experienced no significant injuries to the offensive starters, the coaches were the same, and their opponents where not significantly better. All the available evidence would have one predict Russel Wilson was going to close out the season with ≈ 60 touchdown passes and a play-off bye. Instead Wilson ultimately passed for 40 touchdowns and the Seahawks were eliminated in a wild card game [RWP]. So what happened, and more importantly, what does this mean in our modeling context?

Around week 8 the Seattle head coach, Pete Carroll met with OC Brian Schottenheiemer and forced him to change the offensive strategy of the team. Carroll was not satisfied with the aggressive pass-heavy attack that was fueling the Seahawks success, and wanted a different approach to match what he perceived as "tougher" opponents down the stretch. Carroll forced a change, despite all the available evidence showing the existing game-plan was effective. The Seattle offense sputtered, capped off with an embarrassing loss in a wild card game they should have never been in, and Brian Schottenheimer was (scapegoated) fired. This example illustrates why the modeling of quarterback passing touchdowns will be flawed for the foreseeable future [Condotta, 2021] [Press, 2021].

Pete Carroll's judgement had a direct and dramatic impact on Russell Wilsons' performance as well as the Seahawks as a whole. What feature could have been used as part of our model to accurately account for this? It is our position that no such feature exists. This is a fundamentally unknowable casual driver with no sufficient proxy. Consequently, any model like the one discussed in this paper would have encountered serious error when attempting to predict the number of Wilson's future touchdown throws, in large part because the model would have been missing key information relevant to his success. What happened with Russell Wilson, is dramatic but not unique, and is happening in one way or another in every NFL game and across each season.

Unfortunately each NFL game, and each individual season for every team is an example of a dynamic system. Everything happening over the course of a game affects everything else, and the relationships governing the eventual outcome are most likely non-linear and complicated. This makes it so that extremely small changes at one point in time, can have massive unforeseen consequences

later [Silver, 2012, p. 194-196]. The rationale for Pete Carroll's decision to change the offense after 2.5 years, if taken at face value, was based on his perception of the Seahawks' upcoming schedule (among other things). In other words his own predictions of what opponents in the near future would due to try and stop him. He, like our model, and the rest of the 319 coaches and 32 general managers in the NFL, was attempting to operate with data whose value is continually changing and uncertain. Additionally his mistakes, which will be numerous as a consequence of the ever changing environment, compound and have strange consequences affecting all players including the quarterback under his direction. Individual NFL games (and each season as a whole) represent systems which are over-connected and in a constant state of flux. Any attempt to model non-trivial aspects of the game will suffer from high variability, i.e. RMSE and resulting prediction ranges too large to be of use. It follows that quarterback performance will remain difficult to predict.

References

Football outsiders. URL https://www.footballoutsiders.com/.

Pro football focus. URL https://www.pff.com/.

Pro football reference. URL https://www.pro-football-reference.com/.

Nfl team passing touchdowns per game. URL https://www.teamrankings.com/nfl/stat/passing-touchdowns-per-game?date=2012-02-05.

Russell wilson game log 2020. URL https://www.pro-football-reference.com/players/W/WilsRu00/gamelog/2020/.

Marta Avalos, Philippe Hellard, and Jean-Claude Chatard. Modeling the training-performance relationship using a mixed model in elite swimmers. *Medicine and Science in Sports and Exercise*, 35 (5):838–846, May 2003.

Bob Condotta. Pete carroll's solution to seahawks' second-half offensive woes? 'we have to run it more', 2021. URL https://www.seattletimes.com/sports/seahawks/pete-carrolls-solution-to-seahawks-second-half-offensive-woes-we-have-to-run-it-more/.

Marylu Galvan, Arnulfo Rojas, Sandra Chavarria, Manuel Elizondo, Jose Mendoza, Arturo Borrego, Gabriel Mancilla, and Jorge Lundez. Mathematical models at the olympic games to predict road events. *international Journal of Innovative Computing, Information and Control*, 14(5):1905–1915, October 2018.

Danny Kelly. The nfl has never seen an offensive boom like this, 2020. URL https://www.theringer.com/2020/11/12/21561876/nfl-scoring-boom-offense.

Mike Ozanian and Christina Settimi. The nfls most valuable teams 2021, 2021. URL https://www.forbes.com/sites/mikeozanian/2021/08/05.

Associated Press. Seahawks fire offensive coordinator brian schottenheimer, 2021. URL https://www.king5.com/article/sports/nfl/seahawks/seahawks-fire-offensive-coordinator-brian-schottenheimer/281-15c625cd-b3d6-4e1d-a4b3-76c1cb1da4b4.

Nate Silver. *The Signal and the Noise*. Penguin Group, 2012.