PREDICTING NFL OUTCOMES USING TWITTER DATA

Josh Jiang[*] **Declan McNamara**[*]

1 Introduction

One day in 1906, statistician Francis Galton attended a country fair in the West of England. There, he encountered a contest whereby attendees guess the weight of an ox. Participants wrote their guess on a ticket and submit it to the event organizer. At the end of the contest, Galton borrowed the tickets to perform statistical analysis. To his surprise, he found that while any individual guess could be far from the true weight of the ox, the average of all 787 guesses was within one pound of the mark. What Galton encountered that day is the phenomenon known as the *wisdom of crowds*, first coined by James Surowiecki [11], which is the notion that the judgement of a collective will be more accurate than that of any single individual. Such a phenomenon will serve as the fundamental motivation for this paper.

The NFL (National Football League) is an American sports league consisting of 32 teams playing American football. Each NFL season lasts roughly from September to February, and during this time each of the 32 teams play only 16 regular season games total due to the physicality of the sport, plus a few playoff games for those teams that qualify. Games generally occur on Thursday, Sunday, and Monday (most games occurring on Sunday), but games can also be played on Saturday, and due to the COVID-19 pandemic, games were even played on Wednesday in the 2020-2021 season [2].

Insight into the outcomes of NFL matchups are highly sought after. As the most popular sport in the United States [6], there is an enormous market of sports pundits and amateurs who attempt to predict results. More generally, the betting market for NFL games is ever-growing, with market cap around $1 trillion [7]. Due to the large amount of interest and opinions out there regarding the NFL, we are interested in applying the *wisdom of crowds* theory towards prognosticating the outcome (win/lose) of games. Can the collective opinion of the public be an accurate predictor of the outcome of NFL games? As a proxy for public opinion, we will use tweets.

The goal of this paper is to build a model that predicts the outcome of NFL games relying entirely on tweets regarding the game. Section 2 will elaborate on the data sources used in our analysis. Section 3 will discuss the data gathering and preprocessing steps; namely how we transform a collection of tweets to useful data for analysis. Section 4 will include the results of out analysis following the preprocessing step. In section 5, we offer a semantic interpretation of out results. Section 6 will explore an extension of our project, which is to develop a model to profitably gamble on the outcome of games. Lastly, section 7 will be a discussion on the results of our analysis and suggested improvements that are worth exploring in the future.

2 Data Sources

We used season data from `profootballreference.com` to obtain for all games for each of the 2019-2020 and 2020-2021 NFL seasons the following information: home team, away team, date, time (US/Eastern), home points scored, away points scored, and winner [1, 2]. Hereafter for ease, we refer to the 2019-2020 season and 2020-2021 season as just the 2019 and 2020 seasons, respectively. We choose to focus on regular season games alone (omitting playoff games), resulting in data on 256 games for each season.

Following the example of [8], we try to predict for every matchup whether the home team or away team will win. Such a model is flexible and doesn't need to know information about the teams individually. This setup has a small number of ambiguities, namely the 4 NFL games which were played in London during the 2019 season, but we can easily resolve these using the (seemingly arbitrary) home/away designation provided by the NFL press release for these games,

[*] Both authors contributed equally to the paper.

mirrored in our season data [3, 1]. The only loss of information here is perhaps any true home field advantage that our model may find in games played in the U.S., but we consider this issue small enough to be ignored.

Twitter data for each team and game was extracted in the form of raw tweets using the `snscrape` library [5], of which more details are given below.

For our gambling analysis, historical odds and money lines are accessed via [4] and merged to our other season data. There is no consensus to these quantities in the betting marketplace, but most quotes are very similar to each other, so we take the data given at this source as the typical odds/lines one might see in the marketplace.

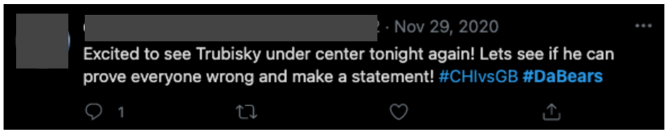
3 Preprocessing

As mentioned, the raw tweets are gathered using the `snscrape` library. For each game and for each team (home/away), we look for tweets that contain hashtags related to the teams involved and are within 3 days before the start of the game. We reasoned that 3 days was an ideal look-back window to ensure that tweets were indeed about the game of interest, rather than the previous week's game (it is conceivable that at worst, a team could play on Monday and then again on Thursday). The list of hashtags associated with each team are provided by [10], with some small tweaks to account for team name changes. To avoid ambiguity, we only used tweets which contained hashtags associated with exactly one team, following the example of [10], and threw away tweets which contained hashtags from 2 or more teams. In total, we used the following amount of tweets in our analysis across the 256 games for each season.

	2019-2020	2020-2021
Home	353,398	326,344
Away	334,411	322,862

Figure 1: Tweet Volume

After the tweets have been gathered, the content of each tweet is reduced to a *bag of words* and then the frequency of each word is counted in a final dictionary. Figure 1 details that process for one tweet. Note that in reducing a tweet to a *bag of words*, we use several normalizing techniques including enforcing lower case, removing punctuations, and lemmatizing, which is reducing each word to its root word. Furthermore, while the figure below is an example where the content is reduced to unigrams, it can also be extended into n-grams. In our analysis, we used both unigrams and bigrams.



[redacted] [redacted] · Nov 29, 2020

Excited to see Trubisky under center tonight again! Lets see if he can prove everyone wrong and make a statement! #CHIvsGB #DaBears

```
['excited','see','trubisky','center','tonight','let',
'see','prove','everyone','wrong','make','statement',
'chivsgb','dabear']
```

```
{'excited':1,'see':2,'trubisky':1,'center':1,'tonight':1,
'let':1,'prove':1,'everyone':1,'wrong':1,'make':1,
'statement':1,'chivsgb':1,'dabear':1}
```

Figure 2: Preprocessing one tweet

This process is repeated for all tweets for the home and away teams separately and in the end, we obtain a master dictionary of word counts for home and away. Using that, we create a *corpus* of words, H and A, for the home and away teams, by taking the words that have a frequency above a given threshold (in our analysis, we took all words with frequency higher than 0.1% of total number of tweets.)

$$H = \{h_1, \ldots, h_{N_H}\}$$
$$A = \{a_1, \ldots, a_{N_A}\}$$

where h_j is the j^{th} word of the home corpus. Using H and A, we can create a vector representation for each game,

$$G_i = [f^H_{i1}, \dots, f^H_{iN_H}, f^A_{i1}, \dots, f^A_{iN_A}]^T$$

where G_i is the vector representation for the i^{th} game, f^H_{ij} is the the log transform of 1 + frequency of h_j. The frequency of each word is counted by looking at all the tweets gathered for the particular game. Repeating this process for each game, we create the final data matrix M that looks like,

$$M = \left[\begin{array}{ccc|ccc|c} f^H_{11} & f^H_{12} & \cdots & f^A_{11} & f^A_{12} & \cdots & R_1 \\ \vdots & \ddots & & \vdots & \ddots & & \vdots \\ f^H_{N1} & & f^H_{NN_H} & f^A_{N1} & & f^A_{NN_A} & R_N \end{array}\right],$$

where each row $M_i = [G_i, R_i]$, where $R_i \in \{0, 1\}$ is the indicator that the i^{th} game resulted in the home team winning. Now that we have M, this becomes a classification problem where G is the data matrix and R is the response.

This is a very difficult prediction problem for a variety of reasons. Viewed as a classification task, predicting a "Win" or "Loss" (class label 1 or 0) is nondeterministic and is complicated by the randomness of sports outcomes. If we consider market consensus as the "gold standard" of this task, we can see how difficult the problem is. Betting the odds-on favorite for every NFL game in the 2020 results in a classification accuracy of only about 68.4%. Using only Twitter language data to predict, we should not expect our model to perform as well as the market consensus, which presumably makes uses of magnitudes more information than our model.

4 Classification Results

We created data matrix M_{2019} using tweets from the 2019 season and M_{2020} using tweets from the 2020 season. We used a suite of classification tools to train on M_{2019} and test on M_{2020}. We created the corpus using the 2019 season to avoid look-ahead bias.

Figure 2 summarizes the results. It is important to note that many of the methods are sensitive to how their parameters are initialized and will produce different results when rerun. For our figures, we set a seed and report the results of a single run.

	Test Accuracy - Unigrams	Test Accuracy - Bigrams
Only Home	50.0%	50.0%
Logistic None	57.8%	57.4%
Logistic L1	58.2%	60.5%
Logistic L2	62.5%	61.3%
Adaboost	52.0%	56.3%
Random Forest	55.9%	59.0%
Naive Bayes	55.9%	53.1%
Neural Network	57.8%	59.4%

Figure 3: Results when using full training set

One observation is that $M \in \mathbb{R}^{256\times(N_H+N_A)}$ is very high dimensional as $N_H + N_A$ is in the order of thousands, so overfitting posed a significant problem for us. As a result, we sought to use either penalized methods (e.g. penalized logistic regression, penalized neural network weights) to remain robust to overfitting or aggregation methods (adaboost, random forest) to ensure that undue weight is not put on any single unigram/bigram in the classification task. Our models are implemented in `sklearn` and the penalty parameters for logistic regression were chosen by cross validation. Our neural networks are dense, multilayer networks created using `keras` and tuned accordingly, and the neural network weights at each layer are given an elastic net penalty (L1 + L2) to prevent overfitting. All results are available for viewing on our project GitHub page.

In examining our classification results, we top out at around 60% classification accuracy, a feat that is quite good considering the market can only achieve 68.4% accuracy. Notably, our analysis using Twitter data only performs much better than a similar analysis by [10], who were only able to achieve 52.3% accuracy in predicting the game winner when using only Twitter data. We can perhaps attribute some of this difference to the growth of Twitter as a platform since the years 2010-2012 that the prior work focused on [10].

Given the size of our dataset, one obvious method to try was dimensionality reduction. We used PCA to reduce the columns of M down to 10 and reran our analysis. Figure 4 summarizes those results, which unfortunately demonstrate no significant gain, or perhaps even some loss, in predictive power.

	Test Accuracy - Unigrams	Test Accuracy - Bigrams
Only Home	50.0%	50.0%
Logistic None	60.5%	58.5%
Logistic L1	59.0%	57.8%
Logistic L2	60.9%	61.3%
Adaboost	54.7%	53.5%
Random Forest	61.3%	57.8%
Naive Bayes	52.7%	50.4%
Neural Network	54.7%	58.2%

Figure 4: Results when using training set with dimensionality reduction

5 Semantics

Given our results above, we ought to motivate why our models work, because it may not be immediately obvious to the reader where the signal in our data is coming from, or what grams the model is finding to be a significant predictor of wins/losses. To illustrate, we focus in on our L1 regularized logistic regression model above, seeking some interpretability in the grams chosen by the model.

On the unigram dataset, the L1 regularized logistic regression, with penalty chosen by cross validation, has a coefficient vector with 1958 nonzero coefficients (out of 3033) total, which is not sparse enough to examine all nonzero coefficients. Recalling that our dataset consists of only positive entries, and the fact that via our log transform , covariates are on similar scales, we may look for those coefficients with the largest positive and negative magnitudes as those deemed "most important" by this model. We examine the 20 coefficients with largest positive values, and the 20 with largest negative values, and provide a table of selected unigrams below from each of these groups of 20.

Unigram	Home/Away
’drew’	H
’force’	H
’surgery’	A
’ranking’	H
’cold’	A
’turnover’	A
’michigan’	A
’ravensnation’	H

Figure 5: Selected Unigrams with Largest Positive Coefficients

Unigram	Home/Away
’terrible’	A
’turned’	A
’ravensnation’	A
’lmao’	H
’kcchiefs’	A
’watt’	A
’preseason’	H
’tyler’	A

Figure 6: Selected Unigrams with Largest Negative Coefficients

We see that many of the unigrams have intuitive value, and we can see what types of words the model is recovering from the (noisy) corpus. Largely, the model does a great job at recovering the teams of the league, either via their players or team names. For example, it’s likely that the gram "drew" corresponds to player Drew Brees of the New Orleans Saints, one of the best teams in the league in the 2019 season. Accordingly, high frequencies of the unigram for the home team for a particular game means that the home team for that game is the Saints, and accordingly this contributes to a higher win probability for the home team in the form of a large, positive coefficient. Similarly, the gram "watt" likely corresponds to player J.J. Watt of the Houston Texans, another talented team in the 2019 season, and so accordingly occurrences of this gram with respect to the away team contribute to a lower win probability for the home team in the form of a large negative coefficient.

We see something similar, but even more direct with the names of teams, in particular the Baltimore Ravens (who were the top team in the league in the 2019 season). Accordingly, whenever our model is able to discern the Ravens are playing (either as the home team or away team), the magnitude and sign of the coefficient reflect that the model believes the Ravens will be victorious.

Lastly, the model does indeed pick up vocabulary more in line with general sentiment, something we had hoped for in this task. Words like "surgery" or "turnover" used with respect to the away team both make sense as positive indicators of a home team win. Worries about the weather (e.g. away team being worried about it being too "cold" in Wisconsin or Chicago) also seem to contribute to higher probability of a home team win, while the presence of derisive language about the home team (e.g. "lmao") seems to contribute to a higher probability of an away team win.

Bigram	Coef. Sign	Home/Away
('super', 'bowl')	+1	H
('pro', 'bowl')	+1	H
('ravensflock', 'ravensnation')	+1	H
('injury', 'report')	-1	A
('next', 'year')	+1	A
('kansa', 'city')	-1	A
('bengal', 'whodey')	+1	A
('russell', 'wilson')	-1	A

Figure 7: Selected Bigrams from L1 Penalized LR

We can do the same process in the bigrams case. Generally, the bigrams are less interpretable than the unigrams, so for illustrative purposes, we use a heavy L1 penalty to induce sparsity and further restrict the corpus for this example to omit bigrams containing "http" as a gram. With these small tweaks, the L1 penalized logistic regression results in a coefficient vector with only 26 nonzero entries out of 15,315, some which are given below along with their signs, and whether the bigram was associated with the home or away team.

We see similar reference to teams and players and get a couple of new ones (Russell Wilson is QB of the Seattle Seahawks, a top team of 2019, while the Cincinnati Bengals had the worst record in the league in 2019). Appropriately, sentiments about Super Bowl hopes are viewed as particularly meaningful for wins, as fanbases presumably only tend to speculate on Super Bowl hopes if their team is a real contender; similarly, speculation or hopes for a team's "next" "year" indicates a write off of the current season, indicative of anticipation of poor results. These examples above demonstrate some of advantages of our Twitter data, but there are disadvantages too. The success of our model depends on the assumption that language carries similar connotations across the 2019 and 2020 seasons, and this assumption may not hold, particularly with respect to language about particular players or teams, which can change drastically between seasons.

Rank	Team	W-L Record
1	Baltimore Ravens	14-2
T2	San Francisco 49ers	13-3
T2	Green Bay Packers	13-3
T2	New Orleans Saints	13-3
T5	Kansas City Chiefs	12-4
T8	Houston Texans	10-6
32	Cincinnati Bengals	2-14

Figure 8: Selected W-L Records for 2019 Season

Rank	Team	W-L Record
1	Kansas City Chiefs	14-2
T2	Green Bay Packers	13-3
T2	Buffalo Bills	13-3
T4	New Orleans Saints	12-4
T21	San Francisco 49ers	6-10
T29	Houston Texans	4-12

Figure 9: Selected W-L Records for 2020 Season

For example, in comparing the 2019 and 2020 seasons, we see that while there are some usual suspects near the top of the league (e.g. Kansas City Chiefs, New Orleans Saints), other teams dropped dramatically in the rankings between these two seasons (e.g. Houston Texans, San Francisco 49ers). This poses a problem to us because language concerning such teams and their players can easily cloud the signal in our model or, at worst, cause us to make completely inaccurate predictions. Nevertheless, our accuracies as shown in the previous section convince us that these problems are minimal overall.

6 Extension: Beating the Market & Gambling Analysis

Given all of our above work, it's natural to question whether or not we can outperform the market if we permit ourselves to use more than just Twitter data, which we have seen above has a limit in the predictive accuracy that it alone can yield. For example, [10] uses betting market data in the form of the point spread and over/under line as inputs to their model, in addition to statistical features such as the average number of points scored/given up by a team in the current season. We would like to similarly augment our design matrix to include some additional data in hopes of increasing our predictive accuracy.

Statistical features like average points scored are not only rolling estimates, but also team and season dependent, and as a result are hard to assemble in the time frame given. As a result, we limit ourselves to including gambling data which are concise and easily accessed for each game and team. This data is readily available prior to each game, so there is no look-ahead bias by including this data as an input to our model.

More precisely, using [4] we include exactly 2 additional features to our model: the home team money line and the away team money line. For the unfamiliar reader, money lines are betting quantities that specify precise payouts for betting on a particular team to win. In specifying these payouts, the money line has baked in not only the favorite of a particular matchup, but how heavily the team is favored. For example, in the first week of the 2020 season, one money line pairing was as below:

$$\text{(Home) Kansas City Chiefs:} -450$$
$$\text{(Away) Houston Texans:} +375$$

The numbers above specify payouts as follows: a "+" in the money line denotes the underdog, and the number in this case denotes the amount of profit won by betting \$100; e.g. a \$100 bet on the Texans to win pays out \$375 dollars on top of the original \$100 if the Texans indeed win. On the other hand, a "-" in the money line generally denotes the favorite, and in this case the number denotes the amount one must bet to win \$100 in profit; in this case, a \$450 dollar bet on the Kansas City Chiefs to win only pays out \$100 in profit if they do win. The Chiefs are a heavy favorite in this scenario.

Other matchups are much tighter. For example, another matchup in the first week of the 2020 season had money lines of

$$\text{(Home) Atlanta Falcons:} -105$$
$$\text{(Away) Seattle Seahawks:} -115$$

where profit payouts for either team are less than the original bet. In situations like these, we take the favorite to be the team with the smaller payout, which in this case would be the Atlanta Falcons.

We augment our dataset to include the money line for both home/away teams for each game. After this augmentation, our dataset now looks something like

$$M^* = \left[\begin{array}{ccc|ccc|cc|c} f^H_{11} & f^H_{12} & \cdots & f^A_{11} & f^A_{12} & \cdots & m^H_1 & m^A_1 & R_1 \\ \vdots & \ddots & & \vdots & \ddots & & \vdots & & \vdots \\ f^H_{N1} & & f^H_{NN_H} & f^A_{N1} & & f^A_{NN_A} & m^H_N & m^A_N & R_N \end{array}\right]$$

where m^H_i, m^A_i denote the money lines for the home and away team, respectively, for the the ith game of the season. For our analysis below, we use bigrams exclusively with respect to the twitter features.

It turns out that the naive prediction strategy of picking the betting favorite to win each time results in a prediction accuracy of 68.4%. We are interested in exploring the question of whether or not we can achieve better accuracy than the market consensus by gains in knowledge through our Twitter data and the *wisdom of crowds*. It's not evident that we should achieve any gain at all on the surface, because supposedly the market money line estimates have already baked in the opinions of many. To test this goal, we train the models we did before, now including the home/away money line as inputs, to see the test accuracy we can achieve. Our results are below.

	Test Accuracy	# Underdogs
Only Favorite	68.4%	0
Logistic L1	68.4%	0
Logistic L2	68.4%	8
Adaboost	62.1%	84
Random Forest	64.5%	80
Neural Network	70.3%	31

Figure 10: Test Set Results with Money Lines

In the figure, the "# Underdogs" column denotes the the number of underdogs predicted to win in the 2020 season by our model, in other words the number of deviations from the naive betting strategy. We see that our models might be vastly different from the naive strategy (e.g. Adaboost differs on 84 out of 256 games, nearly a third of total games), or not different at all. In our case, L1 penalized logistic regression collapses down into the naive prediction model because the sparsity imposed by the penalty term results in a coefficient vector with only 2 nonzero entries: those corresponding to the home/away money lines.

In terms of raw predictive accuracy, our models do not really seem to gain much on the naive prediction strategy. In fact, they do significantly worse in some instances, and only somewhat better in 1 instance: our neural network outperforms the market accuracy by about 2% or so. As our test set is so small, though, with only 256 observations, this difference may not indicate much significant gain, although there is some improvement.

However, raw prediction accuracy may not be the best metric to measure information gain due to the noisiness of our data. We want to examine whether or not our model is learning substantively from Twitter beyond the information encapsulated in the money line, and one way to examine this question to see whether our model is finding teams which are undervalued by the market. To test this hypothesis, we pursue a simple betting strategy where based on each model's predictions, we bet $100 on the predicted winner to win the game, and collect payouts accordingly as specified by the money line. For example, betting $100 on the Chiefs to win at a +450 money line as above pays out $22.22 in profit for a correct pick, and pays out a loss of -$100 for an incorrect pick. After computing payouts for all models on the 2020 season, we gather the results below.

	Test Accuracy	# Underdogs	Payout	Return
Only Favorite	68.4%	0	-$618.92	-2.4%
Logistic L1	68.4%	0	-$618.92	-2.4%
Logistic L2	68.4%	8	-$593.70	-2.3%
Adaboost	62.1%	84	$346.42	1.4%
Random Forest	64.5%	80	$1621.23	6.3%
Neural Network	70.3%	31	$1511.00	5.9%

Figure 11: Results & Payouts

The results are quite striking. Our more complex models that have significant deviations from the favorite-only prediction strategy (namely Random Forest, Adaboost, and Neural Network) are very profitable, with returns of up to about 6% or so on the total of $100 × 256 = $25,600 wagered. On the contrary, naive strategies which bet the market favorite every time actually lose money, with a return of -2.4%. To us, this provides compelling evidence that our models are finding significant information gain from the Twitter data in the form of teams which are undervalued by the market. The payouts from the underdog picks by models trained on all of this data seem to vastly outweigh the losses from missed underdog picks. These are more than lucky guesses from a few games as well: our model with the best payout, the Random Forest, picks a total of 80 underdogs, far from just a few, and the fact that this model comes out so far ahead is evidence that this model is consistently picking underdogs correctly across the season.

7 Discussion

Our project seeks to answer the question if Twitter data can be an accurate predictor of the outcome of NFL games, thus exemplifying the *wisdom of crowds*. Our analysis demonstrates that there is useful information to be gained in analysing tweets to predict the outcome of NFL games. Our best models were able to achieve prediction accuracies in the low 60% range. Given the *gold standard* of 68.4%, which is achieved by picking the odds-on favorite for each game, our models are certainly not bad. Furthermore, we considered our model performance using another metric - payouts if we were to gamble according to the model. Using that, we showed that we can create a profitable gambling strategy using just Twitter data and money lines. This is perhaps a better metric for determining performance as it adds a unique weight to correctly predicting each game (e.g. correctly picking an +200 underdog counts more than correctly picking a -300 favorite.)

While our procedure works, there are several assumptions we used that do not necessarily hold up. One, as mentioned, is that the vernacular and sentiments of fans remain the same between seasons. This does not hold generally as teams can fluctuate greatly between seasons. The model can be wrong if it trains on a season where a particular team is good and assigns favorable coefficients to words related to it but then those sentiments reverse in the test season. Another assumption is that fan sentiment is same across the NFL. In our method, we are essentially creating a one-size-fits-all model to predict every game. This of course is not the case in reality as each game involves two unique fan bases with their own characteristics.

One improvement to the change between seasons would be to simply include data for the weeks leading up to the game. For example, for a game played in week 8, we can repeat the preprocessing and model training steps using tweets from the training season and weeks 1-7 of the current season. By including data from the current season, the model can stay up to date. With regards to the concern about the model being too general, it is difficult to solve that issue without fundamentally changing our method. Schumaker et al. [9], proposed a method using time series analysis on fan sentiment in the days leading up to the game to predict the outcome of it. A method like that would take into account the unique fluctuations in sentiment for each fan base but is a fundamentally different approach from ours.

References

[1] 2019 NFL Weekly League Schedule. https://www.pro-football-reference.com/years/2019/games.htm.

[2] 2020 NFL Weekly League Schedule. https://www.pro-football-reference.com/years/2020/games.htm.

[3] NFL announces 2019 London Games. https://www.nfl.com/news/nfl-announces-2019-london-games-0ap3000001012397.

[4] NFL Scores and Odds Archive. https://www.sportsbookreviewsonline.com/scoresoddsarchives/nfl/nfloddsarchives.htm.

[5] snscrape GitHub Page. https://github.com/JustAnotherArchivist/snscrape.

[6] Sports in the United States. https://en.wikipedia.org/wiki/Sports_in_the_United_States#American_football.

[7] S. Anderson. NFL Betting Market Efficiency: Finding a Profitable Betting Strategy. *Economics Student Theses and Capstone Projects*, 2019.

[8] S. Kampakis and A. Adamides. Using Twitter to predict football outcomes.

[9] R. P. Schumaker, C. S. Labedz, A. T. Jarmoszko, and L. L. Brown. Prediction from regional angst – a study of nfl sentiment in twitter using technical stock market charting. *Decision Support Systems*, 98:80–88, 2017.

[10] S. Sinha, C. Dyer, K. Gimpel, and N. A. Smith. Predicting the NFL Using Twitter.

[11] J. Surowiecki. *The wisdom of crowds*. Anchor Books, 2004.