Monitoring Public Sentiment of NFL Draft Picks via Machine Learning Techniques

Jonathan Wiseman
Elizabethtown College

Honors Senior Thesis
Release Agreement Form

The High Library supports the preservation and dissemination of all papers and projects completed as part of the requirements for the Elizabethtown College Honors Program (Honors Senior Thesis). Your signature on the following form confirms your authorship of this work and your permission for the High Library to make this work available. By agreeing to make it available, you are also agreeing to have this work included in the institutional repository, JayScholar. If you partnered with others in the creation of this work, your signature also confirms that you have obtained their permission to make this work available.

Should any concerns arise regarding making this work available, faculty advisors may contact the Director of the High Library to discuss the available options.

Release Agreement
I, as the author of this work, do hereby grant to Elizabethtown College and the High Library a non-exclusive worldwide license to reproduce and distribute my project, in whole or in part, in all forms of media, including but not limited to electronic media, now or hereafter known, subject to the following terms and conditions:

Copyright
No copyrights are transferred by this agreement, so I, as the author, retain all rights to the work, including but not limited to the right to use in future works (such as articles or books). With this submission, I represent that any third-party content included in the project has been used with permission from the copyright holder(s) or falls within fair use under United States copyright law (http://www.copyright.gov/title17/92chap1.html#107).

Access and Use
The work will be preserved and made available for educational purposes only. Signing this document does not endorse or authorize the commercial use of the content. I do not, however, hold Elizabethtown College or the High Library responsible for third party use of this content.

Term
This agreement will remain in effect unless permission is withdrawn by the author via written request to the High Library.

Signature: Jonathan Wiseman **Date:** 04/30/2020

Monitoring Public Sentiment of NFL Draft Picks via Machine Learning Techniques

By

Jonathan Wiseman

This thesis is submitted in partial fulfillment of the requirements for Honors in the Discipline in Computer Science and the Elizabethtown College Honors Program

May 1, 2020

Thesis Director (signature required)____Peilong Li__________

Department Chair (signature required) Thomas A. Leap

Second Reader__________Jingwen Wang________________

Third Reader____
[only if applicable such as with interdisciplinary theses]

Monitoring Public Sentiment of NFL Draft Picks via Machine Learning Techniques

Jonathan Wiseman
Elizabethtown College
wisemanj@etown.edu

ABSTRACT

Sentiment analysis is a topic in natural language processing that seeks to automatically extract positive and negative polarity from text data. Its applications are diverse, ranging from marketing and sales to forum moderation to gauging public opinion. One particularly interesting application area is found in professional sports: fans share a huge volume of opinions, predictions, and reactions online that can be used to monitor public opinion on specific teams, coaches, and players. This paper explores the application of machine learning based sentiment analysis on a hand-labeled social media dataset focused on reacting to National Football League draft picks. The resulting model, called *DraftSense*, provides information that can be used for future analysis, including attitude towards drafted players, comparison between fan reactions and on-field performance, and comparison between drafted players based on the language used to describe them. Additionally, a labeled dataset for sentiment analysis on professional football will be created for further use.

1. INTRODUCTION

The National Football League is the world's most profitable sports league, achieving over $13 billion in revenue in 2017 alone [4]. The league continues to expand, attracting viewers from around the world, and evolve with regular changes to safety standards, rules and regulations, and even team locations. Accompanying this expansion has been an interest in applying analytics to data generated by NFL players, coaches, and fans. In 2019, the NFL hosted its inaugural Big Data Bowl, challenging college and independent teams to make use of its databases to generate valuable insights about the game and its players [10]. The spirit of the Big Data Bowl reflects a growing interest in using the techniques of data analysis and machine learning to generate insights that stretch across a myriad of sports areas.

The application areas of data analysis in professional football are diverse, ranging from a Sabermetrics-like approach to predicting game and player performances to suggesting rule and safety changes to market analysis of commercial placement and fan engagement. A large media empire has developed around professional football with injury reports, game predictions, and assorted player and coaching news providing constant coverage on all aspects of the game.

Sentiment analysis is a field spanning the disciplines of natural language processing, machine learning, information retrieval, and text mining that seeks to automatically extract the standpoint, view, and mood of an author [14]. Its most common use is to determine the polarity (positive or negative) of a particular sample of text. This can be of great use in marketing research, where companies seek to gauge public opinion of their products; other application areas include monitoring of online forums, automatically assessing product reviews, and as additional input for search engines [2].

There are two primary methods of performing sentiment analysis. The first is a grammatical approach based on the linguistic features of text, such as descriptive adjectives and adverbs, negation words (i.e. "not"), intensifiers (i.e. "very," "extremely"), case, and tense [2]. This approach involves the creation of a carefully crafted lexicon that accurately captures the sentiment of words that are specific or important to a domain; for example, a lexicon crafted for determining sentiment in sports articles would have to assign sentiment to words like "interception" and "fumble." The second approach involves the use of machine learning algorithms to create models that can predict the sentiment of a given text based on labeled data.

A general challenge with sentiment analysis is its inability to generalize across domains; for example, a lexicon or model crafted for use in the movie reviews domain will not generalize well to the sports domain [2]. This makes the crafting of specific lexicons time-consuming and requires a significant amount of domain knowledge. The machine learning approach runs into similar problems: supervised classification requires carefully labeled datasets, which are often not publicly available or are based on implicit ratings (for example, movie and product reviews are standardized on a five "star" scale that gives text data implicit ratings). Either approach requires a significant investment in either crafting a lexicon or acquiring a significant dataset that captures the nuances of a given application field.

Any potential use of sentiment analysis on NFL articles must be performed with a specific goal in mind and with a tailored dataset. However, using news articles - which represent structured and proofed text - to predict the outcome of NFL games is problematic. For one, most articles are not specific to one aspect of the game: there are injury reports; news and updates on trades, signings and draft prospects; articles about players' personal lives; and news about retired players and coaches that are no longer active in the game. Each of these areas requires a specific lexicon, and it is doubtful that each is useful in predicting the outcome of a specific football game. Secondly, each article deals with multiple players and topics, such that extracting entity-based sentiment is difficult. For example, one sentence in an article might deal with an offensive and a defensive player at the same time. This makes sentiment analysis difficult, since phrasal extraction is a difficult area of natural language processing [2]; additionally, this requires a model that is capable of orienting sentiment-bearing words to specific players based on that player's context (i.e. an interception is bad for an offensive player but good for a defensive player).

It is clear that any sentiment model based on football text must be directed and purposeful. One potentially useful application is determining public sentiment towards NFL draft picks. The NFL draft is an annual event in which college football players are selected by professional teams for short-term "rookie" contracts [7]; it is the primary mechanism by which college talent enters the NFL. This task is useful for several key reasons. For one, high-valued draft picks (i.e. those selected in the early rounds of the draft) are expected to be polished, capable players. Although rookie contracts are generally inexpensive compared to those for veteran players [7], teams wish to avoid selecting players whose draft stock does not translate well into actual on-field performance. In this way, creating a model to process text data related to draft picks is a useful tool for gauging expert and public opinion towards a player's potential. Secondly, gauging sentiment towards a player is useful from a marketing perspective. The off-field (and sometimes on-field) actions of a player influence fans' perspectives of players and their willingness to engage with the franchises to which they belong. For example, the impact of on-field protests by NFL players such as Colin Kaepernick on NFL revenues is examined in [5]; for an example of a player's actions harming team reputation, see the example of Antonio Brown in [6].

Acquiring a dataset dedicated to the NFL, and to the NFL draft in particular, will require collection of specific and directed material. Social media represents a uniquely vibrant source of material for sentiment analysis. For one, material is widely available and easily collected by making use of existing APIs. One such social media platform that is highly specific is Reddit, a popular news aggregation and content hosting website. Reddit allows its users to form communities, called subreddits, where discussion is focused on a particular topic. For example, the r/NFL subreddit is dedicated to news and events related to the NFL. In this subreddit, users create posts (also called threads) that discuss a particular news story or event. The comments gathered from these threads deal with the particular event in question, and thus represent highly directed reactions to specific events. Thus, Reddit represents a source of reactions that carry sentiment about specific events.

To gauge public reactions to NFL draft picks, I propose *DraftSense*, a machine learning approach to sentiment analysis on text relating to draft picks after they are made. The key design goals of DraftSense are:

Comprehensive: the ability to collect a large volume of data

Specific: collecting data specific to NFL draft picks

Accurate: accurately predict sentiment to summarize the public's reactions to NFL draft picks

2. BACKGROUND

2.1 Related Work

2.1.1 Aggregate Forecasting

It has been consistently observed that the aggregation of a number of individual forecasts leads to better performance over time than relying on a single forecast [16].

In [16], this principle was applied in the sphere of politics and international events by the Good Judgment Project and tested over time in the U.S. Intelligence Advanced Research Projects Activity's Aggregative Contingent Estimation program [16]. The team made probabilistic judgements about specific events (e.g. Greece leaving the Eurozone) by framing them as yes-no questions and presenting them to a poll of 2400 Americans from wide-ranging demographics and professions [16]. The team employed various aggregation techniques ranging from simple averaging to log-odds extremizing of weighted averages [16]. Overall, their methods outperformed U.S. intelligence community predictions by about 30%, even when intelligence officials were given access to classified material [16].

The work carried out by the Good Judgment Project presents several interesting findings. Chief among these is the idea that combining individual predictions (as biased and perhaps ill-informed as they may be) outperforms the singular opinion of an expert. This means that many opinions of perhaps lower quality can be used to obtain a fairly reasonable predictor of future events. It also suggests that social media, where opinions are clear and abundant, might be able to provide a good source of material for making predictions.

Secondly, the attempt to quantify the outcome of events as binary allows one to frame problems as questions of classification. This brings complex events into the realm of prediction, ignoring any potential nuance in favor of a quantifiable outcome. For evaluating NFL draft picks, the question now becomes simple: was the choice to draft player X a good choice?

Finally, the Good Judgment Project utilized a number of different aggregation methods. This makes it possible to break *DraftSense* into two distinct components: one for analyzing sentiment and one for aggregating predictions.

However, the work presented in [16] suffers from a few drawbacks that limit its overall effectiveness. Its primary weakness is its reliance on polls to produce predictions. Sending out a poll for every question that needs answering can be time consuming, expensive, and lead to biased results. Here, the volume of data available on the Internet to be collected by *DraftSense* can help increase speed and scale. Rather than waiting for thousands of individual polls to be answered and returned, *DraftSense* can quickly scrape a high volume of social media posts for predictions focused on specific players.

2.1.2 Sentiment Analysis on NFL Data

There are two existing projects utilizing sentiment analysis to make predictions on NFL games: *Lydia*, developed by Hong and Skiena in [8], and the work of Sinha et al. in [15].

In *Lydia*, a lexical approach to sentiment analysis was applied to text data from news, blog, and other web sources in order to produce a betting paradigm for NFL games. The favorability of a team is derived from its daily positive and negative mentions in the authors' text dataset [8]. Utilizing sentiment alone, the authors achieved 60% prediction accuracy for the 2006-2008 seasons [8]. The authors found that combining sentiment, statistical performance prior to games, and home field advantage produced the most robust model; however, the authors note that the sentiment model only produced significant improvements over the second half of the NFL season, after commentators and fans had developed opinions about teams [8].

Lydia offers a generic framework for how social media data can be used to predict real-world events. The production of raw positive and negative mentions, and their aggregation, is a simple and intuitive approach to deriving general feeling towards a team. However, it suffers from being far too general for practical use. For example, there is no filtering performed on any of the data being scraped. That means that injury reports, coach news, historical articles (e.g. a recap of last week's game), and more are all included in the raw counts. Secondly, *Lydia* was not developed specifically for analyzing sentiment in sports (and not specifically for American football). Its lexicon-based analysis of sports articles is thus questionable. Finally, *Lydia* was used as part of a prediction related to betting lines. This significantly hampers its scope: rather than predicting game outcomes themselves, *Lydia* is used to predict when to bet against the odds.

DraftSense improves upon these limitations by scraping comments from player-specific Reddit threads. For example, all of the comments scraped from the Patrick Mahomes thread are related to Patrick Mahomes and his selection by the Kansas City Chiefs; thus, there is no extraneous information included. Secondly, *DraftSense* is trained on a dataset specifically focused on football. Finally, *DraftSense* avoids predicting betting lines and instead focuses on evaluating public opinion at large.

The work of Sinha et al. in [15] represents another significant inspiration for *DraftSense*. Here, the authors utilize a simple lexicon-based sentiment analysis on Tweets to predict game outcomes. Their approach follows the general logic of *DraftSense*: aggregating social media opinions to produce a forecast. Additionally, the authors combine their text analytics with traditional game statistics (such as a team's win/loss record) to increase accuracy, much like the designers of *Lydia*.

However, the authors of [15] make no attempt to increase scale or speed. There is no component to automatically collect and analyze data, with Tweets needing hand-labeling for effective analysis. One of the major goals of *DraftSense* is its training on a comment dataset so as to be able to automatically analyze a huge volume of data at high speeds. Finally, the authors make no further use of their text data beyond attempting to beat the bookies' over/under line. There is no attempt to track sentiment, trending topics, or compare teams over the course of the season. *DraftSense* will be able to utilize sentence embeddings in order to provide direct comparisons of player similarity. This has applications beyond prediction, such as visualization of the language used to discuss a player or attempting to find a correlation between a player's attributes (e.g. a good arm or fast run speed) and their performance in the league after their draft.

3. DESIGN

Figure 1 displays the overall project outline:

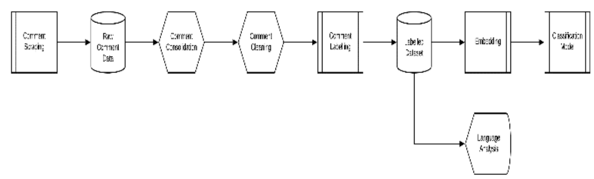

Figure 1: *DraftSense* overview

3.1 Overview

The creation of the *DraftSense* sentiment analysis model can be broken down into five steps:

1. Scraping of comments
2. Data preprocessing
3. Dataset labelling
4. Production of sentence embeddings
5. Training of classification model

The creation of *DraftSense* also resulted in the creation of a free, publicly available dataset. To obtain this dataset, comments were scraped from Reddit via the official PRAW Python library [1]; consult section 3.2 below for a detailed discussion of scraping methodology. Steps four and five utilized the Python libraries Scikit-learn and GenSim [12, 13]. Sentence embeddings were generated from GenSim's implementation of the Sent2Vec algorithm discussed in section 3.5. Sentiment analysis itself was treated as a binary classification task focused on identifying positive and negative polarity. Classification algorithms were limited to Logistic Regression, Naïve Bayes, Random Forest, and Support Vector Machine.

Note that these were the steps utilized to create *DraftSense* from the ground up, which includes a significant portion dedicated to dataset creation and model training. In an application scenario, only the data preprocessing and sentence embedding steps would need to be undertaken.

3.2 Data Collection: Comment Scraping

As discussed above, sentiment analysis is a task requiring labelled datasets in highly specific domains. In order to create *DraftSense*, it was necessary to collect, clean, and label a large corpus of text pertaining specifically to the NFL draft. In order to accomplish this task, it was decided to limit draft picks to quarterbacks selected in the first round; this is assumed to provide a uniformity of language relating to the position. For example, the language used to discuss a quarterback pertains primarily to throwing, running, and managing offenses; this differs from the language used to discuss other offensive positions (which include running, catching, and blocking) and defensive positions (which includes tackling). Additionally, quarterbacks picked in the first round represent significant expenditures of draft capital; in this way, they have a higher expectation put on them and thus it is assumed more emotional, evaluative language will be used to discuss their selection.

In order to understand the scraping methodology, it is necessary to understand the structure of Reddit's r/NFL community. In order to reduce the number of posts reacting to each draft pick, the community has consolidated "megathreads" about draft picks. These are posted by an automatic moderator named u/NFL_Mod. Reddit itself contains a layered comment structure which can be arbitrarily deep. A direct comment on the post is called a top-level comment. A comment that replies to a top-level comment is a second-level comment, and so on. In this way, large reactions in the form of top-level comments and discussion in the form of deeper comments occurs.

With the draft subjects narrowed down to first round quarterbacks, twelve were chosen to create the dataset. The top first- and second-level Reddit comments from each player's draft reaction thread on r/NFL were scraped using PRAW, for a total of 14,434 comments. The twelve players, as well as the number of comments scraped for each player are listed below:

1. Baker Mayfield – 1381
2. Mitchell Trubisky – 1945
3. Daniel Jones – 3165
4. Kyler Murray – 1571
5. Lamar Jackson – 889
6. Dwayne Haskins – 948
7. DeShaun Watson – 910
8. Sam Darnold – 637
9. DeShone Kizer – 589
10. Josh Rosen – 674
11. Josh Allen – 766
12. Patrick Mahomes II – 959

The inequality in comment number for the quarterbacks reflects varying degrees of community interest and factors such as shock, humor, and approval. Quarterbacks whose selection was controversial, such as Mitchell Trubisky, tend to have a higher number of first- and second-level comments. Additionally, some quarterbacks received overwhelmingly positive or negative comments. This certainly influenced the final label distribution and quality of the dataset; for a discussion of its impact, see section 3.4: Dataset Labelling.

3.3 Data Preprocessing

The comment data collected in the first stage in creating *DraftSense* needs cleaning prior to its use in sentiment analysis. There are two major steps required: comment consolidation and comment cleaning.

3.3.1 Comment Consolidation

The raw comment data for each quarterback is stored in a .json file with the following fields:

- comment_id: the unique Reddit comment ID
- post_id: the unique Reddit post ID
- comment: text data of comment

Each quarterback's data is stored in a separate .json file. In order to consolidate the dataset, it was necessary to compile these comments into a larger file. This larger .json file had the following fields:

- subject: quarterback the comment is discussing
- comment: text data of comment
- label: numeric label corresponding to sentiment

With a consolidated .json file, comment cleaning can be performed.

3.3.2 Comment Cleaning

Not all of the text data collected and consolidated is immediately usable. There are a number of characters and patterns that should be removed for natural language processing tasks. For *DraftSense*, the following were scrubbed from the text data: emojis, URLs (including links to other subreddits and users), number signs and hashtags, quotation marks, brackets, parentheses, slashes (forward and back), asterisks, tildes, and newline characters.

Most of these characters and patterns are removed because they contain little detail that can be used to distinguish positive and negative language. For example, URLs may contain links to articles or photos expressing positive or negative polarity but do not constitute single text comments. While utilizing emojis would be very helpful in identifying polarity, many language processing libraries are not yet equipped to handle them. Finally, newline characters were removed primarily for visual clarity in hand labelling.

3.4 Dataset Labelling

With the dataset consolidated and cleaned, hand-labelling was performed. This step was by far the most time consuming and difficult. After an initial trial period, it was decided to split labels into four categories: positive, negative, jokes and memes, and irrelevant comments. Only the positive and negative comments were used for sentiment analysis in *DraftSense*. Joke comments were those that were neither positive nor negative and with humorous intent. For example, Baker Mayfield received hundreds of joke comments relating to his supposed uttering of the phrase "hee hee" during pre-draft interviews with the Browns [3]. Finally, comments that did not directly discuss the draftee in question were deemed irrelevant. Irrelevant comments took many forms, but most commonly discussed trades, other players, coaches, and management officers.

Each of the 14,434 comments were labelled by hand with no external input. This, of course, makes the dataset highly influenced by the subjective opinion of the author. Additionally, it is possible that some comments meant as jokes in the form of exaggeration were included as positive or negative comments.

For an exploration of labels and the resulting dataset in general, see section 4: Exploratory Data Analysis.

3.5 Sentence Embedding

Many machine learning models require a numeric vector as input. As such, the labelled text data must be converted to a numeric vector. There are a number of unique approaches to this task, including a bag-of-words matrix approach, word embedding, and sentence/document embedding.

The bag-of-words approach represents each document (comments in this case) as a column in a matrix, with rows corresponding to words or punctuation. If a word or punctuation is present in the document, then a numeric representation fills the matrix's cell. This can take many forms: a simple binary representation (1 if present, 0 if absent), a kind of weighted representation (i.e. a representation of the word's importance in that document), or a total count of that word's appearance in the document. The bag-of-words approach has scalability issues: as the number of documents and unique words increases, the number of resources required to represent the documents and construct machine learning models also increases. Additionally, the bag-of-words model does not generally account for word order.

To avoid the issues of scalability and loss of the information encoded in word order, an approach known as embedding has emerged. The core aim of embeddings is to represents words and sentences as numeric vectors such that those that are similar to each other are closer in the vector space according to some distance or similarity metric. The simplest form of this is word embedding, in which each word in a text corpus is represented in a vector space such that the words most similar to it are closest. For example, in a text corpus discussing animals, the words "dog" and "cat" would be closer to each other than to the word "elephant." This is primarily accomplished through a predictive task: predicting the next word in a series of words using a neural network.

However, word embeddings fall short when dealing with a corpus with longer documents of varying lengths [9]. An extension of word embeddings was developed in Sent2Vec [11]. Here, documents of varying lengths can be embedded in a vector space to be used as input to machine learning tasks. For *DraftSense*, the implementation of Sent2Vec contained in the Python GenSim library was utilized to produce document embeddings for each comment [13]. Sent2Vec allows the user to specify a document embedding dimension. It is not immediately clear which dimension to use; as such, the embedding dimension was treated as a hyperparameter for model tuning.

3.6 Sentiment Analysis

With the dataset labeled and text data converted into numeric vectors, sentiment analysis may be performed. Four models were trained: Logistic Regression, Naïve Bayes, Random Forest, and a Support Vector Machine.

Only the data classified as positive or negative was used in model training; that is, the classification task was treated as binary. While multi-class classification is possible, it was decided to avoid the joke and irrelevant comments for *DraftSense*. However, this presents a difficulty for any real-world application, since the majority of comments were in fact deemed to be irrelevant or jokes. In a real-world situation, it might be beneficial to have several stages of classification:

1. Distinguish between jokes/irrelevant comments and positive/negative comments
2. Distinguish between positive and negative comments

However, such an arrangement was not tested in the creation of *DraftSense*.

4. EXPLORATORY DATA ANALYSIS

One of the most interesting aspects of *DraftSense* is that it enables a visualization of the language used to discuss the NFL draft on Reddit. For positive, negative, and joke/irrelevant comments there are many interesting trends that reveal a lot about the type of language Redditors used to express their opinions.

The labelling process resulted in 1616 positive, 2652 negative, 3035 joke, and 7131 irrelevant comments. Figure 2 visualizes the distribution of labels:

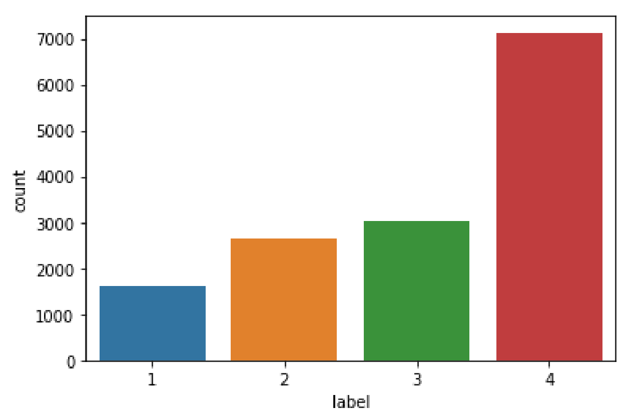

Figure 2: Distribution of Labels

4.1 Positive Comments

Positive comments represented the smallest partition of the dataset. DeShaun Watson and Josh Rosen received the most positive comments, with 250 and 241, respectively. Josh Allen received the fewest positive comments at 55, but not the highest share of negative comments. Figure 3 displays a word cloud representing the most common words and phrases used to discuss draftees positively:

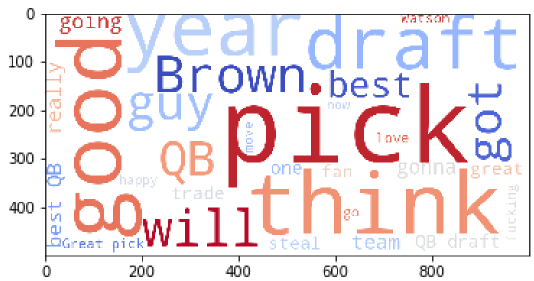

Figure 3: Most Common Positive Words

There are a few interesting trends. For one, words that deal with evaluation are common: "good," "best," "great." There is also a significant number of words that are football- and draft-specific: "QB," "steal," "trade," "great pick." This perhaps reflects the need of a specific dataset that includes these words and phrases prominently. Additionally, there are a number of words that deal with the future: "gonna," "going," "will." This perhaps reflects optimism regarding the future: fans believe that their franchise is in good hands and will succeed.

4.2 Negative Comments

Negative comments were far more prevalent than positive comments. This probably reflects the players chosen to represent the dataset; additionally, high draft picks (particularly quarterbacks) have high expectations attached to them. Fans want to see players who they think are deserving of high picks and who are ready to perform in the NFL without delay. Daniel Jones and Mitchell Trubisky received the most negative comments with 879 and 611, respectively. Josh Rosen received the fewest negative comments at 34. Figure 4 shows a word cloud visualizing negative comments:

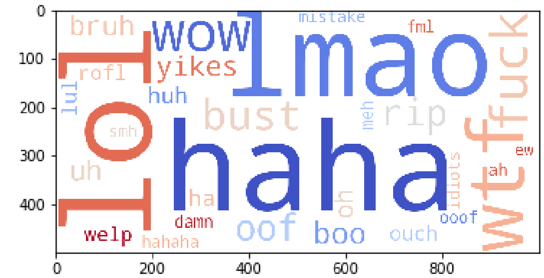

Figure 4: Most Common Negative Words

Most of these words are short, one-word reactions to the draftees. There is also profanity, something that was not particularly common amongst positive comments. Finally, most of these words suggest anger or shock, whereas positive comments dealt with optimism and evaluation.

The length of negative comments is one of their key features. Figure 5 shows the average character length of positive and negative comments:

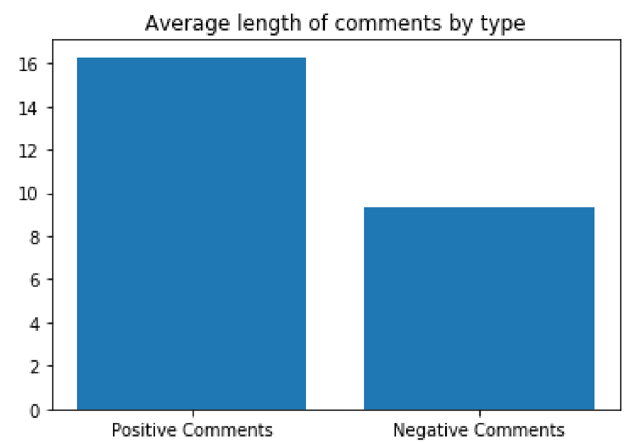

Figure 5: Average Length of Positive and Negative Comments

One can see that, despite making up a larger portion of the total dataset, negative comments are significantly shorter than positive comments. In fact, there are a total of 26305 characters associated with positive comments, while 24718 characters are associated with negative comments. Thus, despite having significantly more data, there is less total negative text.

There is also surprisingly little originality in negative comments. For example, there appear to be 291 unique one-word negative comments. However, after further cleaning the dataset and merging close forms (i.e. "hahaha" and "hahahahahaha" are both forms of "haha"), the number of unique one-word comments drops to 140.

4.3 Joke and Irrelevant Comments

The majority of the dataset belongs to irrelevant comments: comments discussing trades, coaches, other players, or anything else not related to the draftee. Figure 6 displays a word cloud for irrelevant comments:

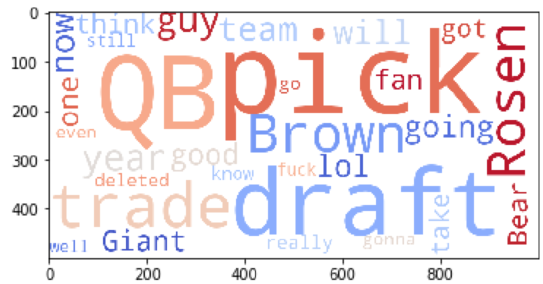

Figure 6: Most Common Irrelevant Comments

There are a few things that are worth noting. First, comments that were deleted were considered irrelevant. Secondly, specific names like "Brown," "Rosen," "Giant," and "Bear" appear with high frequency. This suggests that irrelevant comments were often discussing the actions of other teams or making comparisons with other players. There is also a lot of language related to the draft itself: "draft," "pick," and "trade." This indicates that many irrelevant comments were discussing the draft as a whole as opposed to the player who had just been drafted.

Joke comments tended to be repetitive and dealt with events surrounding particular players. For example, Daniel Jones was often mocked after his selection by the Giants based on his resemblance to Eli Manning, a former Giants quarterback. The Cleveland Browns are often the subject of jokes, as their reputation as a poorly performing team over the past few years makes an easy target. Daniel Jones and Baker Mayfield received the most joke comments with 794 and 421, respectively. DeShone Kizer received the fewest joke comments at 72.

5. SENTENCE EMBEDDING SELECTION

As discussed above, the ideal dimension of sentence embedding was treated as a hyperparameter for model tuning. Figure 7 displays embedding dimension against performance for the four selected machine learning models:

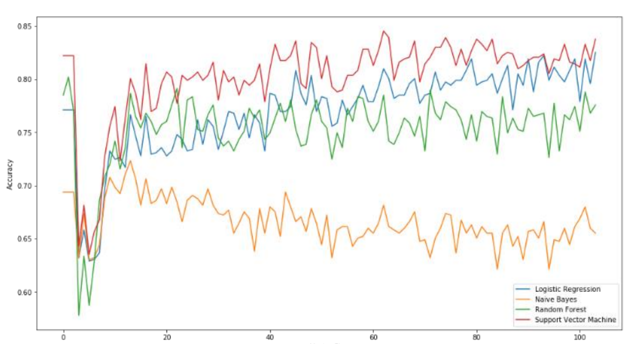

Figure 7: Embedding Dimension vs. Model Performance

There is very little spread in model accuracy as the embedding dimension increases beyond 20. The highest reported accuracy was around 85% on a Support Vector Machine with a vector size of 63.

Additionally, the vectors produced by Sent2Vec were not normalized. Normalization of embeddings can help to overcome the issue of long sections of text dominating over smaller sections of text. As noted in section 4.2, negative comments tended to be significantly shorter than positive comments. Thus, the differences in text length actually further distinguish positive from negative text.

It is also possible to visualize each comment in a two-dimensional space using PCA reduction on the embedded vectors. Figure 8 displays the resulting comment embeddings color-coded by label:

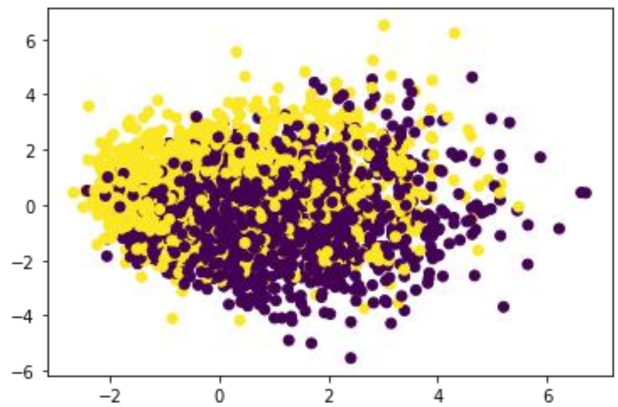

Figure 8: Comment Embeddings Visualized

While there is some overlap between the embeddings for positive and negative comments, there are two clear clusters. This indicates that positive and negative comments were generally distinguishable by their language content.

6. CLASSIFICATION MODELS AND PEFORMANCE

Four machine learning models were trained on the text dataset as embedded by Sent2Vec with a vector size of 62. For negative comments, one-word replies were consolidated using regular expressions. In order to account for the imbalance between positive and negative comments, sampling was performed on negative comments. Classification accuracy was used as the measure of model performance. Table 1 displays the classification accuracy of each model:

Model	Accuracy
Logistic Regression	77%
Naïve Bayes	65%
Random Forest	77%
Support Vector Machine	84%

Table 1: Classification Accuracy for Models

6.1 Logistic Regression

A simple Logistic Regression model was trained on the dataset. The model reached 77.4% accuracy at its highest. The model's confusion matrix suggests that it struggled with false positives more than false negatives: negative comments were labeled positive at a higher frequency than positive comments were labeled negative.

6.2 Naive Bayes

A Naïve Bayes model was trained on the dataset, reaching only 65.3% accuracy (by far the worst of the four). Like the Logistic Regression model, it struggled more with false positives than false negatives.

6.3 Random Forest

The Random Forest model achieved 76.8% accuracy, performing similarly to the Logistic Regression model. However, the Random Forest model struggled almost exclusively with false negatives: positive comments that were labeled negative.

6.4 Support Vector Machine

The Support Vector Machine model achieved 84.5% accuracy, by far the highest of any of the four models. Like the Random Forest, the Support Vector Machine struggled with false negatives over false positives.

A grid search for ideal parameters was not performed for the Support Vector Machine, partly due to speed and time: on average, training the SVM with one set of parameters took twenty minutes. Nonetheless, its relatively high performance suggests room for future improvement in the form of parameter tuning.

7. CONCLUSION

DraftSense represented an attempt at applying sentiment analysis and aggregate forecasting to the domain of professional football, with particular emphasis on monitoring public sentiment towards draft picks. After examining the design, implementation, and performance of *DraftSense*, one can clearly see that its original goals of comprehensiveness, specificity, and accuracy have been largely met.

With regard to comprehensiveness, *DraftSense* is implemented with the ability to scrape, clean, and optionally label large volumes of social media text relating to specific draft picks. For its initial training, *DraftSense* gathered over 14,000 comments describing twelve NFL draftees. Comments were cleaned automatically using well-defined rules and labelled manually to produce a comprehensive dataset. However, the process suffered a bottleneck in dataset generation brought about by manual labelling. Bias by way of subjectivity was introduced by having a single labeler, as was a reduction in production speed. Additionally, the final sentiment analysis model was incapable of detecting joke and irrelevant comments, hindering applications to threads that have not been pre-screened. Finally, performance was perhaps hindered by exploring only the application of Sent2Vec in generating sentence embeddings. Future work to improve the comprehensiveness of *DraftSense* potentially includes the expansion of the text dataset, reduction of label subjectivity by increasing the number of labelers, and introduction of multi-class or a multi-level classification scheme. The first two of these goals can be accomplished by means of crowd-sourced platforms such as Amazon's Mechanical Turk; multi-class or multi-level classification can be accomplished through dataset expansion and schemes such as first detecting positive/negative vs. joke/irrelevant followed by positive/negative classification.

With regard to specificity and accuracy, the production of *DraftSense* has also resulted in the production of a publicly available, NFL-draft specific, labelled text dataset. The sentiment analysis model was trained on text that dealt specifically with the NFL draft, helping it to avoid the problem of domain specificity that generally plagues sentiment analysis. As a measure of success, model performance reached 84%. Future improvements possibly include expansion of the training dataset as discussed above and parameter tuning of specific models to increase performance.

In addition to achieving the goals that guided *DraftSense*, the project also resulted in key insights about the language that characterizes the discussion of NFL draft picks on social media. These include insights about language patterns in positive and negative comments such as emotion and tone, tense, length, and repetition.

Finally, *DraftSense* represents an attempt to quantify emotion and opens the path to harnessing this data to make verifiable predictions about the future. With the ability to assess a quarterback's selection as positive or negative, one can make a prediction about that quarterback's future performance. *DraftSense* enables the thousands of individual predictions that are made about the future to be combined according to the principles of aggregate forecasting. Along with the dataset that accompanies *DraftSense*, this provides a platform that other students and researchers may utilize and expand.

8. REFERENCES

[1] Bryce Boe. PRAW Documentation. (January 27, 2020). Retrieved January 27, 2020 from https://praw.readthedocs.io/en/latest/index.html.

[2] Erik Boiy, Pieter Hens, Koen Deschacht, and Marie-Francine Moens. 2007. Automatic Sentiment Analysis in On-line Text. In *Proceedings ELPUB 2007 Conference on Electronic Publishing*, 13 - 15 June 2007, Vienna, Austria. 349-360.

[3] Christian D'Andrea. 2018. 'HEE HEE' is the new rallying cry for Baker Mayfield and the Browns. (Aril 26, 2018). Retrieved April 26, 2020 from https://www.sbnation.com/2018/4/26/17288584/browns-baker-mayfield-pied-piper-hee-hee-hue-jackson.

[4] Forbes. 2017. Total revenue of all National Football League teams from 2001 to 2018 (I in billion U.S. Dollars). Retrieved January 23, 2020 from https://www.statista.com/statistics/193457/total-league-revenue-of-the-nfl-since-2005/.

[5] Ahiza Garcia. 2018. Will the NFL Ban on Kneeling Boost Ratings? (May 23, 2018). Retrieved January 27, 2020 from

https://money.cnn.com/2018/05/23/news/nfl-ratings-anthem-protest/index.html.

[6] Paul Gutierrez and Mike Reiss. 2019. Timeline of Antonio Brown's Season. (Sep. 20, 2019). Retrieved January 27, 2020 from https://www.espn.com/nfl/story/_/id/27544508/line-antonio-brown-season.

[7] Tadd Haislop. 2019. How does the NFL Draft Work? Rules, Rounds, Eligibility, and More. (April 27, 2019). Retrieved January 27, 2020 from https://www.sportingnews.com/us/nfl/news/how-does-the-nfl-draft-work-rules-rounds-eligibility-and-more/o431yshp0l431e7543pcrzpg1.

[8] Yancheng Hong and Steven Skiena. 2010. The Wisdom of Bookies? Sentiment Analysis Versus the NFL Point Spread. In *Fourth International AAAI Conference on Weblogs and Social Media*, May 2010, AAAI Publications.

[9] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning*, 21 June 2014, Journal of Machine Learning Research, 1189-1196. arXiv: 1405.4053. Retrieved from https://arxiv.org/abs/1405.4053.

[10] NFL OPS. 2019. The NFL's Inaugural Big Data Bowl. (February 2019). Retrieved January 23, 2020 from https://operations.nfl.com/the-game/big-data-bowl/?utm_source=Direct.

[11] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *Proceedings of NAACL-HLT 2018*, June 1 – 6 2018, New Orleans, Louisiana. Association for Computational Linguistics, 528-540. https://doi.org/10.18653/v1/N18-1049.

[12] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine Learning in Python. In *Journal of Machine Learning Research*, 2011, vol 12, 2825 – 2830.

[13] Radim Rehurek. Doc2vec Paragraph Embeddings. (Nov. 1, 2019). Retrieved January 27, 2020 from https://radimrehurek.com/gensim/models/doc2vec.html.

[14] Anuj Sharma and Shubhamoy Dey. 2012. A Comparative Study of Feature Selection and Machine Learning Techniques for Sentiment Analysis. In *Proceedings of the 2012 ACM Research in Applied Computation Symposium*, 23 – 26 October 2012, San Antonio, Texas. ACM Inc., New York, NY, 1-7. https: doi.org/10.1145/2401603.

[15] Shiladitya Sinha, Chris Dyer, Kevin Gimpel, and Noah Smith. 2013. Predicting the NFL using Twitter. In *Proceedings of the ECML/PKDD Workshop on Machine Learning and Data Mining for Sports Analytics*, Prague, Czech Republic.

[16] Philip Tetlock, Barbara Mellers, and Peter Scoblic. 2017. Bringing Probability Judgments into Policy Debates via Forecasting Tournaments. *Science* 355, 6324 (February 3, 2017), 481-483. DOI: https://doi.org/10.1126/science.aal3147.