# Twitter Bot Project Report

Jhagrut Lalwani and Joe Datz

October 9, 2021

**Abstract**

In Sports Statistics, large public sources of information such as *Twitter* are known to have valuable data content but are rarely used systematically due to the need to be able to effectively parse which content provides information a Sports Analyst would value. To advance the usage of Twitter as untapped potential for a source of information, we have produced our own binary labeled dataset of tweets for information containing a report of an athlete receiving an injury and evaluated the effectiveness of both more "classical" machine learning algorithms and state-of-the-art deep learning transformers at classification of these tweets.

## 1 Data.

As of this writing, our dataset for classification consists of 28,387 labeled tweets where 3,561 constitute a statement about the injury of a player in the past, present, or future (labeled as 1), while 24,826 are tweets that can be about any other subject (labeled as 0). These tweets were collected from 2,308 twitter accounts collected from popular sports websites such as *Rotowire* and then gathered directly using python's *twint* library using a Raspberry Pi machine. Some tweet labels were already provided in the form of injury reports with tweets sourced directly from a website, but the overwhelming majority were done by hand.

For comparison purposes, all models are trained on a dataset which contains 8,000 class 0 labels and all class 1 labels. This reduction is done to make sure that our Neural Networks in particular do not suffer from class imbalance issues while still being directly comparable with the standard models.
Of the tweets written by hand, a tweet was labeled as an injury report if:

1. An athlete was moved on or off an injury-based list where they are ineligible to play (excluding for strategic purposes as indicated by the tweet),
2. A direct description of what the injury was for the athlete,
3. A hyperlink to a web page that provides an injury report and a statement in the tweet that indicates such,
4. Any tweet which describes a player in a recovery state such as a rehab outing or exercises to alleviate a condition,
    - Any statement which is not more descriptive than "return to team X" was deemed not enough information because although usually injury, it is not always an injury,
5. Any description of an illness,
6. Hit-By-Pitches in Baseball were included only if:
    - The tweet included statements about the player's reaction as being painful,
    - Any tweet indicating a Hit-By-Pitch to the helmet regardless of the player's reaction.

There are some correction factors to this dataset. For instance, tweets were first labeled sequentially in the order they were gathered, but this might cause sampling bias due to the tweets being labeled all consisting of a similar timeframe and being on a similar set of topics. Tweets then began to be

Table 1:

| Model | Type | Sensitivity | Specificity | Precision | F1 Score | Accuracy |
|---|---|---|---|---|---|---|
| kNN | Boolean | 0.2734 | **0.9965** | **0.9182** | 0.4214 | 0.9058 |
| kNN | TF-IDF | 0.1957 | 0.9942 | 0.8294 | 0.3167 | 0.8941 |
| Bernoulli NB | Boolean | 0.8614 | 0.9486 | 0.7061 | 0.776 | 0.9377 |
| Multinomial NB | Count | 0.8614 | 0.9342 | 0.6525 | 0.7425 | 0.9251 |
| Logistic Regression | TF-IDF | **0.9373** | 0.9787 | 0.8629 | **0.8986** | 0.9735 |
| Random Forest | Boolean | 0.7631 | 0.9719 | 0.7959 | 0.7792 | 0.9458 |
| Random Forest | TF-IDF | 0.8502 | 0.9522 | 0.7184 | 0.7787 | 0.9394 |
| SVM | TF-IDF | 0.8661 | 0.9909 | 0.9315 | 0.8976 | **0.9752** |

randomly sampled from anywhere in our extended dataset of over 1 million tweets to more accurately represent the distribution of sports topics on twitter for labeling purposes. We have also decided to use models we have trained as part of our search for injury statements to intentionally upsample the amount of class 1 labelings, as otherwise we would have a dataset where class 1 labelings would tend towards less than 1% and create a worse performing model.

The choice of injury statements was made because the topic can be very distinct from other topics - i.e, words such as “hamstring” or “ACL” appear here and in virtually no other context, making them easier for models to classify. But in particular, baseball provides a great reference point. The MLB level of baseball has plenty of online reference points such as Fangraph’s injury page to see if our model can keep up with these listings over the course of an entire season, and yet has extremely poor tracking for any league level below MLB despite thousands of players being in them, providing an opportunity for us to supplement baseball knowledge with a much richer dataset for injury analysis.

## 2 Classical Machine Learning Models.

The first models we chose to use were the standard Machine Learning algorithms one would find in a typical beginner course (such as the SVM) to provide a baseline for analysis purposes. Each model had a transformed dataset using the traditional TF-IDF, Boolean, or Count representations for words from the *sklearn* library and word stemming / lemmatization was applied using the *nltk* library to reduce dimensionality. These TF-IDF, boolean, and count matrix representations were done over the entire labeled dataset as opposed to our sample to ensure that the weight values of TF-IDF had converged to an appropriate value. Each model was trained using the same dataset and employed 8-fold Cross Validation to search for each model’s best hyperparameters. Stratified Sampling was also used to ensure that the training and test sets had roughly equal proportions of class 1 and class 0 labelings to work with.

Approxmation methods of PCA such as ICA or Truncated SVD from the sklearn library were also consulted for dimensionality reduction purposes but ultimately failed to fit to our use case. These methods had significantly high runtimes at 15-20 hours per compression for 10,000 tweets, but since we would like to classify incoming twitter information every 24 hours, and in practice this would be over 100,000 tweets, this would be much too expensive to keep up with the flow of information from *Twitter*.

The results of these traditional methods, along with their corresponding representation of the words, are shown in table 1. In our use case we consider the reduction of False Negatives the most significant metric as we would like to avoid any injury reports being lost, which makes sensitivity our most important metric. Here we see that logistic regression has performed the best of all traditional machine learning models.

What makes the Logistic Regression (using an L1 penalty) and SVM models (with respect to its F1 score) so much better performing than the Naive Bayes model at reduction of false positives and false negatives is likely both model’s abilities to produce negative weight values and completely eliminating

Table 2:

| Model | Epochs | Sensitivity | Specificity | Precision | F1 Score | Accuracy |
|---|---|---|---|---|---|---|
| LSTM | 10 | 0.9353 | 0.9850 | 0.9541 | 0.9466 | 0.9726 |
| GRU | 10 | 0.9226 | 0.9864 | 0.9577 | 0.9398 | 0.9705 |
| RoBERTa | 5 | 0.9478 | **0.9887** | **0.9664** | 0.9570 | 0.9782 |
| XLM-RoBERTa | 5 | 0.9648 | 0.9855 | 0.9568 | 0.9608 | **0.9803** |
| XLNet | 5 | **0.9691** | 0.9841 | 0.9530 | **0.9609** | **0.9803** |
| DistilBERT | 5 | 0.8706 | 0.9812 | 0.9393 | 0.9036 | 0.9536 |
| DistilBERT FT | 5 | 0.8861 | 0.9789 | 0.9333 | 0.9091 | 0.9557 |

a variable's impact on prediction by setting a weight value of 0. Naive Bayes has no means by which to eliminate or negatively value bad word choices other than the stop words we have intentionally removed, so all word choices, even if poor choices contribute to a prediction choice of class 1. Conversely, an SVM can choose to ignore a word and logistic regression can even produce weights that negatively impact prediction choice of choice 1, so words like "fan" which indicate a tweet is not actually about a injury report can prevent a class 1 labeling.

The Multinomial Naive Bayes failed to perform any better than a Bernoulli Naive Bayes. This is because since both models' effectiveness is based off of keywords and the average tweet is very short, a keyword such as "MCL," "MRI," or "IL" is only ever invoked once. Thus, the additional integer values gained from the usage of a Multinomial Naive Bayes fails to provide any additional value from its additional complexity and the addition of more data to handle it's added complexity value will not change its lower performance rate.

These models tended to perform well at initial classification of injury reports due to distinct contextual keywords as stated previously. However, this did result in false positives and false negatives that we believed were avoidable with more complex models that could incorporate additional contextual information into its inputs. With respect to false positives, due to the amount of injury report tweets that were about players contracting COVID, this lead to COVID being a highly valued keyword in our Naive Bayes models and resulted in many general tweets about COVID, such as keeping track of Arizona's COVID cases, as False Positive labelings. Sarcastic tweets, tweets about injuries to people that were not athletes, and speculative tweets about injuries were also among the common false positives. Conversely, almost every instance of a false negative was a context-based injury statement; a statement where an injury is being referred to but no keywords are being invoked (i.e, "he'll be back when he's close to 100% again," with no revealing phrase like "torn ACL" appearing).

We believe these context-induced instances of False Positives and False Negatives to be permanent failing points for more traditional machine learning methods, as each model assumes independence of the words and thus has no direct mechanism to learn important contextual information. To that end, we have also extended our modeling to LSTM and GRU Neural Networks in the *Keras* python library as well as state-of-the-art Transformer Neural Networks such as XLM-RoBERTa using the *simpletransformers* library.

## 3 Pre-Trained Transformers.

The results of these Neural Network models can be seen in table 2, with the XLNET and XLM-RoBERTa appearing as the best overall models for our use case. For each model we see a modest decrease in the rate of false positives and false negatives being produced. In the future we would expect that with a larger dataset to train on we will achieve even lower false positive and false negative rates.

The two Neural Networks that are not pre-trained, the LSTM and GRU, are comparable with the best traditional models in performance at classification. This is presumably due to their superior

Table 3:

| Model | T/E | Sensitivity | Specificity | Precision | F1 Score | Accuracy |
|---|---|---|---|---|---|---|
| Logistic Regression | TF-IDF | 0.9373 | 0.9787 | 0.8629 | 0.8986 | 0.9735 |
| SVM | TF-IDF | 0.8661 | **0.9909** | 0.9315 | 0.8976 | 0.9752 |
| XLM-RoBERTa | 5 | 0.9648 | 0.9855 | **0.9568** | 0.9608 | **0.9803** |
| XLNet | 5 | **0.9691** | 0.9841 | 0.9530 | **0.9609** | **0.9803** |

methods of retaining contextual information via the *hidden state* framework that both take advantage of. However, both of them are not as effective as the pre-trained Neural Networks. More data will be necessary to determine whether this is a complexity limitation or simply not enough data for complete convergence of parameters.

We'd like to say that RoBERTa's performance issues relative to all other models is due to overfitting, but we cannot say for sure. In a complete reversal from analysis a month ago, the most complex model - XLM-RoBERTa - has became a close 2nd in its sensitivity metric and so it cannot be a simple case of an overly complex model design. Simultaneously, the simplest model - the DistilBERT and DistilBERT FT - performed significantly worse than it did one month ago with more data to train on. We will need more data at this point to reveal which model is the ultimately the best and why. However, we can say the success of XLNet is likely due to it's ability to compute word dependencies for any permutation of words and its performance was consistent with the result in the prior month.

## 4 Final Results and Future Work.

As we see in our two tables, our best traditional model was done with Logistic Regression for its ability to have negative weight values and XLNet was our best model for Neural Networks, which are represented along with the 2nd best models of each category in table 3. In the future we intend to dramatically increase the amount of data we have to work with by continuing to label data at a steady rate, which will further improve both the standard machine learning models up to a certain point of convergence and a Neural Network to a significantly better performing point of convergence. We also will host a website in the future for which we can display the effectiveness of our models with the appropriately found tweets that are found daily.