Evaluating IPL teams' popularity on Twitter

Sameer Singh

1 Summary

In this project I try to figure out the popularity of various IPL teams on twitter. I collected data from Twitter related to IPL 2020 and trained a Naive Bayes model and a Multinomial Logistic Regression model over the Twitter data. Since, the data collected was unlabeled I manually labeled some of the data I collected and also used some tweets that were similarly categorized into three categories. I got an accuracy of 82% and 87% with the naive Bayes model and the multinomial logistic regression respectively.

2 Introduction

Indian Premier League (IPL) is professional Twenty20 cricket league in India contested annually between 8 teams representing different cities or states in India. The 8 teams that currently participate in IPL are: Mumbai Indians (MI), Chennai Super Kings (CSK), Sunrisers Hyderabad (SRH), Royal Challengers Bangalore (RCB), Kolkata Knight Riders (KKR), Rajasthan Royals (RR), Delhi Capitals (DC) and Kings IX Punjab (KIXP). Though all teams are very popular their popularity vary from region to region and also changes depending on their performance. All these teams maintains a healthy presence on social media networks especially Twitter. They have Twitter handles that keep the fans engaged and updated about the upcoming and ongoing events. In this project, I did a quantitative analysis of the popularity of these teams of Twitter. I collected data from Twitter related to IPL 2020 and performed sentiment analysis. In this project I am trying to answer three main questions:

- Which IPL team generated most "buzz" on twitter during IPL 2020?
- Whether most of the tweets about the team is positive or negative?
- Which team was the favorite on Twitter for winning IPL 2020?

Here by "buzz" I mean the number of tweets, retweets, replies and hashtags generated by each team during IPL 2020. I collect tweet related to each team in IPL and then classify each of those tweet as "positive", "negative" or "neutral" based on various features.

3 Dataset

3.1 Tweets related to IPL

I collected 1000 tweets with each of the following hashtags: #MumbaiIndians, #CSK, #RCB, #KXIP, #RajasthanRoyals, #DelhiCapitals, #KKR and #SRH. The tweets that were collected were

published between between 19 September and 10 November 2020 IST. While collecting tweet we also ensured that none of the tweets are from the official handles of the IPL teams or are endorsed by them. A lot of tweets from Indian subcontinent either on a regional language or contain transliteration of regional languages in English, therefore we simply filter those tweets with such words as I would also be using some other labeled dataset of tweets that contain just English words.

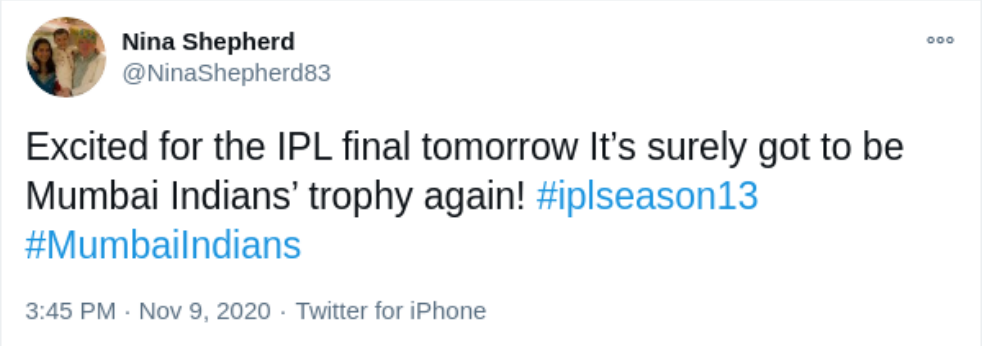

Figure 1: A snapshot of a typical tweet supporting Mumbai Indians

I labeled 800 tweets randomly selected out of the 8000 (1000 tweets per team) tweets collected from Twitter as "positive", "negative", "neutral".

3.2 Tweets unrelated to IPL

Since, only 800 of the IPL tweets I collected from Twitter are annotated, it's not possible to train a good and robust model over them. Therefore, I decided to collect some more data that is already labeled. The other set of data that I used is Niek Sanders' corpus of tweets. The corpus contains over 3000 hand-classified tweets. The tweets are classified to same three categories: "positive", "negative" and "neutral".

After collecting the data, we have 3000 labeled tweets unrelated to IPL, 800 labeled tweets related to IPL and 7200 unlabeled tweets related to IPL.

4 Models

I will use two classifiers for this task: Naive Bayes Classifier and Multinomial Linear Regression.

4.1 Naive Bayes Classifier

Naive Bayes is a probabilistic classifier. So, for a document d, of all the classes $c \in C$, the classifier returns class ĉ which has the maximum posterior probability given the document, where ĉ is the prediction of class c by the model.

$$\hat{c} = \underset{c \in C}{\text{argmax}} P(c|d)$$

Using Bayes' rule and the prior probability of class P(c) and the likelihood of the document $P(d|c)$, we have:

$$\hat{c} = \underset{c \in C}{\text{argmax}} P(d|c)P(c)$$

Without loss of generalization, we can represent a document d as a set of features $f_1, f_2, ..., f_n$:

$$\hat{c} = \underset{c \in C}{\text{argmax}} P(f_1, f_2, ..., f_n|c)P(c)$$

However, the likelihood is too hard to compute, therefore Naive Bayes classifiers make two simplifying assumptions. First, it assumes that the features $f_1, f_2, ..., f_n$ only encode word identity and not position. Second is that the conditional independence assumption that the probabilities $P(f_i|c)$ are independent given the class c. So using chain rule we can write $P(f_1, f_2, ..., f_n|c)$ as:

$$P(f_1, f_2, ..., f_n|c) = P(f_1|c)P(f_2|c)...P(f_n|c)$$

So, finally the equation for Naive Bayes Classifier is:

$$c_{NB} = \underset{c \in C}{\text{argmax}} P(c) \prod_{f \in F} P(f|c)$$

4.2 Multinomial Logistic Regression

Logistic regression is classification method that uses multiple independent variables to predict the probability of category membership on a dependent variable. The multinomial logistic regression is the simple extension of the logistic regression.

5 Experiments

5.1 Features

Now to train classifiers on the Twitter data, I create feature representation of text data. I create two main features using data: unigrams and bigrams. I also plotted the top 5 words in IPL tweets that belongs to each category of data.

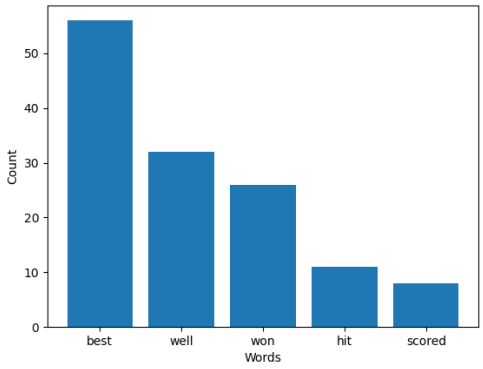

Figure 2: Top 5 words appearing in ”positive” tweets

I found that ”best”, ”well”, ”won”, ”hit”, ”scored” are the 5 most common words in the IPL tweets that were labeled ”positive”.

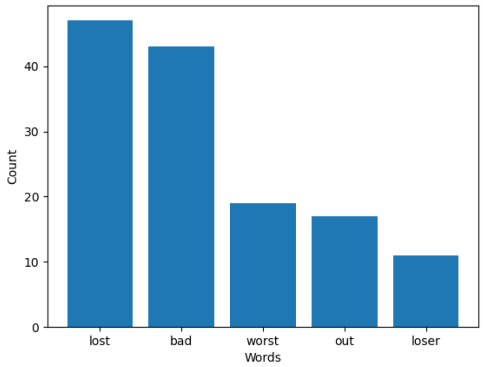

Figure 3: Top 5 words appearing in ”negative” tweets

Also, the 5 most common words in the IPL tweets that were labeled ”positive” are ”lost”, ”bad”, ”worst”, ”out”, ”loser”.

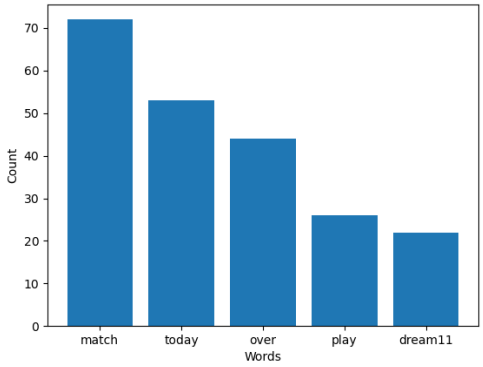

Figure 4: Top 5 words appearing in ”neutral” tweets

Similarly, the 5 most common words in the IPL tweets that were labeled ”neutral” are ”match”, ”today”, ”over”, ”play”, ”dream11”. Note that Dream11 was the sponsor of the IPL 2020.

I used all these most common 15 words from each category as features as well. I also, used the fact that each of these documents are tweets so they have some metadata attached to it. Therefore I used ”liked” and ”retweets” (which are two additional features of Twitter) as features for our models as well.

5.2 Training and Results

I trained the Naive Bayes Classifier over 3500 tweets(3000 non IPL and 500 IPL tweets) of training data and got an accuracy of 81.56% over the 300 tweets(all IPL tweets) of test data.

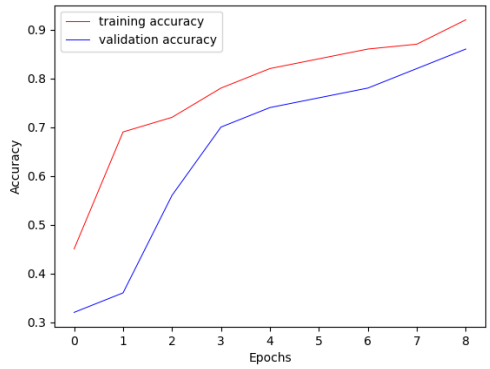

Figure 5: Plot of Logistic Regression model accuracy on Train and Validation Datasets

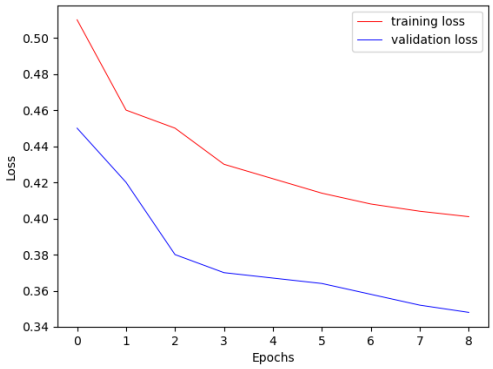

Figure 6: Plot of Logistic Regression model loss on Train and Validation Datasets

However, when I trained a Multinomial Logistic Regression over the same data the accuracy jumped to 87.30%.

6 Conclusion

Working on this problem made me realize that collecting data can be an enormous task in itself. Once you have clean, labeled data training a model is less challenging. When I started working on this problem I wanted to train a few state of the art machine learning models, but as it turn out it took me lot of time to collect and annotate data. But at the same time time I observed that even the simpler models like Naive Bayes and Logistic Regression can give great results if trained over clean and sufficient data. I also believe that given the limited amount of data and computation, neural networks would not have performed as well as these models. Nonetheless, I am pretty satisfied with 87% accuracy that I got with my model given that I also collected my own data for this project.

7 Future Work

I would be annotate more tweets for this task and also apply some other models as I plan to keep woring on this project this winter.

References

[1] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, USA, 1st edition, 2000.

[2] Niek Sanders. Twitter sentiment corpus. sanders analytics.