BIA660 Web Analytics

Instructor: Rong Liu
Student: Chieh Shih, Ping-Lun Yeh, Ming-Ting Hsieh

Introduction:

Since more and more people like to watch sport games in the US such as NBA, NFL, MLB and NHL, sport industries also believe there are enormous business opportunities that can be explored. According to the survey, sport fans like to bet the results of games no matter watching them in person or via TV live stream. Therefore, data and business analytics skills are necessary and crucial in order to seize these business opportunities.

Motivation:

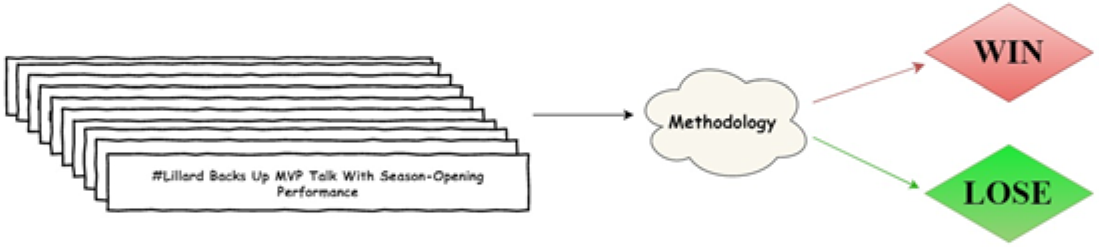


Figure.1 Methodology

As data analytics become more popular, social network analysis related research have received more attention. Normally, Internet users use different kinds of social media to contact with each other, such as Facebook and Twitter. Therefore, I was curious whether we can predict the game results by using these social media because if there were enough information inside, then we may be able to get the predictions with higher accuracy.

Research Overview

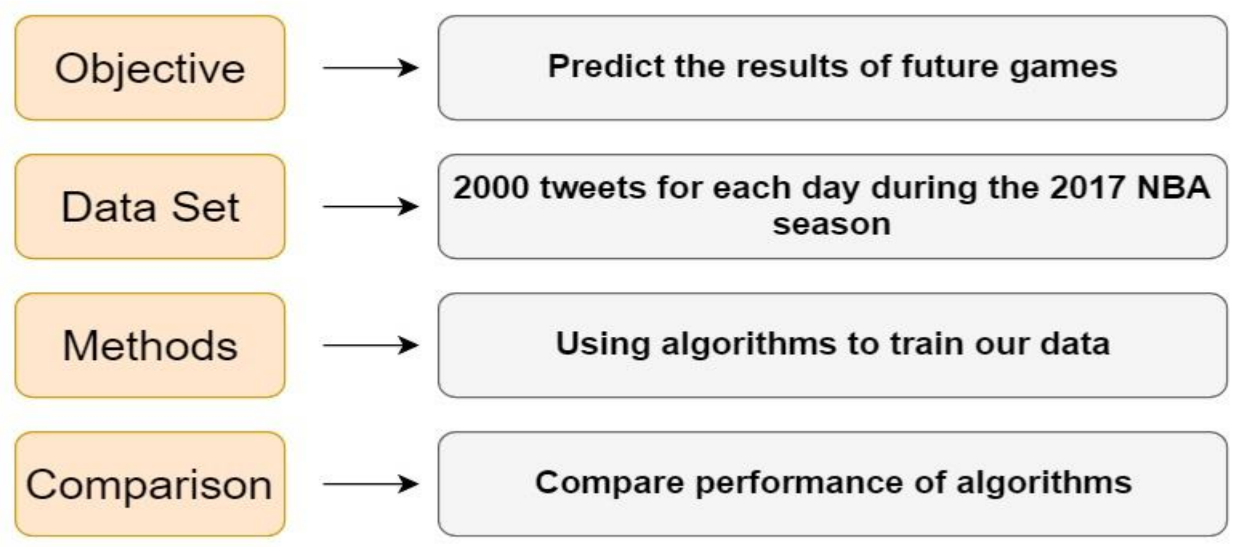

Objective
Predict the results of future games
Data Set
2000 tweets for each day during the 2017 NBA season
Methods
Using algorithms to train our data
Comparison
Compare performance of algorithms

Figure.2 Research Overview

1. Our target is to predict the results of future games
2. There are about 2000 tweets in each day and each team during 2017 NBA season
3. We use five algorithms (Naive Bayes, Support Vector Machine(SVM), K-means, Latent Dirichlet Allocation(LDA), Convolutional Neural Network (CNN)) to train and test our sample dataset (randomly pick teams as our sample dataset)
4. Compare and analyze each result of every algorithm and make conclusions for them.

Workflow

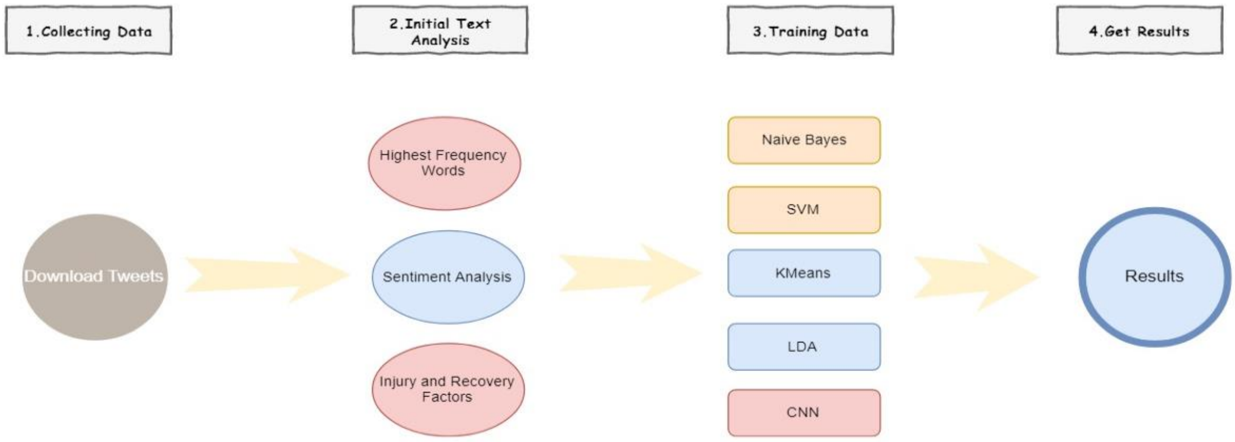


Figure.3 Workflow

Data collection and cleaning

Since the standard Twitter API can not collect the tweets for more than past 30 days, we decide to find another way to get the data. We use Twitter search engine to receive the tweets related to "NBA teams" in 2016. The basic idea is to request directly from tweets and then return them as JSON files.

- ➢ Request example: https://twitter.com/i/search/timeline?f=tweets&q=%20NBA%20since%3A2016-10-25%20until%3A2016-10-26&src=typd&max_position=

- ➢ Reference: https://github.com/Jefferson-Henrique/GetOldTweets-python

- ❖ We collect tweets for each day during the entire 2017 NBA season. The competition started from 2016-10-25 to 2017-04-12 and there are 8 teams as below.

Team Name	Tweets Count
Golden State Warriors	551,248

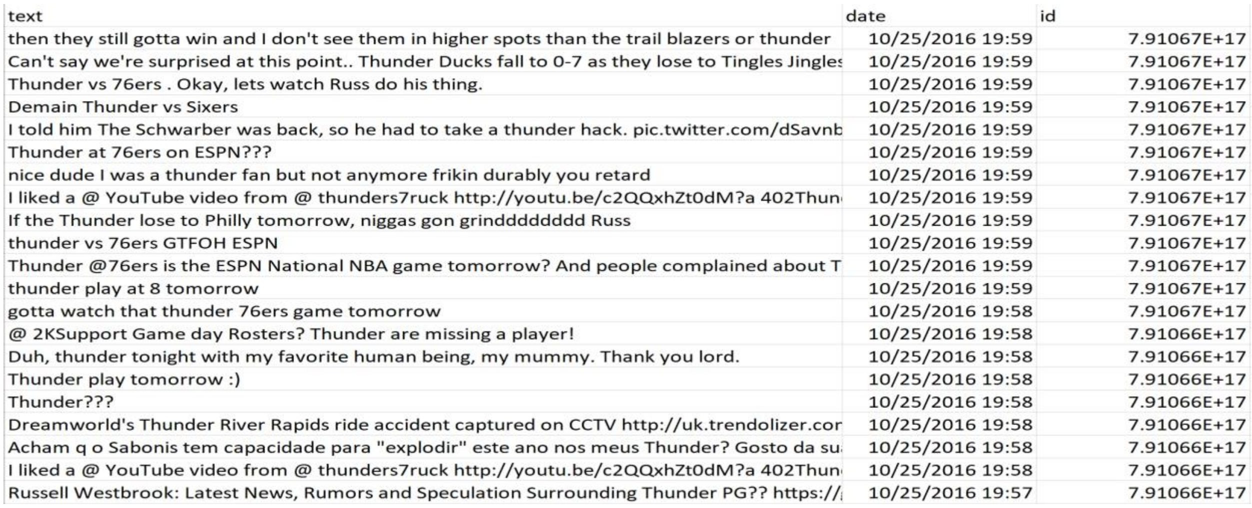
San Antonio Spurs	338,919
Houston Rockets	151,272
Los Angeles Clippers	321,218
Portland Trail Blazers	285,631
Memphis Grizzlies	216,837
Oklahoma City Thunder	332,118
Utah Jazz	248,216

Figure.4 Total tweets : 2,445,459

Python file name: scraping.py

Example:

- ❑ **File format:** Blazers_2016-10-25.csv
- ❑ **File content:** text, date, id



text	date	id
then they still gotta win and I don't see them in higher spots than the trail blazers or thunder	10/25/2016 19:59	7.91067E+17
Can't say we're surprised at this point.. Thunder Ducks fall to 0-7 as they lose to Tingles Jingles	10/25/2016 19:59	7.91067E+17
Thunder vs 76ers . Okay, lets watch Russ do his thing.	10/25/2016 19:59	7.91067E+17
Demain Thunder vs Sixers	10/25/2016 19:59	7.91067E+17
I told him The Schwarber was back, so he had to take a thunder hack. pic.twitter.com/dSavnb	10/25/2016 19:59	7.91067E+17
Thunder at 76ers on ESPN???	10/25/2016 19:59	7.91067E+17
nice dude I was a thunder fan but not anymore frikin durably you retard	10/25/2016 19:59	7.91067E+17
I liked a @ YouTube video from @ thunders7ruck http://youtu.be/c2QQxhZt0dM?a 402Thun	10/25/2016 19:59	7.91067E+17
If the Thunder lose to Philly tomorrow, niggas gon grinddddddd Russ	10/25/2016 19:59	7.91067E+17
thunder vs 76ers GTFOH ESPN	10/25/2016 19:59	7.91067E+17
Thunder @76ers is the ESPN National NBA game tomorrow? And people complained about T	10/25/2016 19:59	7.91067E+17
thunder play at 8 tomorrow	10/25/2016 19:59	7.91067E+17
gotta watch that thunder 76ers game tomorrow	10/25/2016 19:58	7.91067E+17
@ 2KSupport Game day Rosters? Thunder are missing a player!	10/25/2016 19:58	7.91066E+17
Duh, thunder tonight with my favorite human being, my mummy. Thank you lord.	10/25/2016 19:58	7.91066E+17
Thunder play tomorrow :)	10/25/2016 19:58	7.91066E+17
Thunder???	10/25/2016 19:58	7.91066E+17
Dreamworld's Thunder River Rapids ride accident captured on CCTV http://uk.trendolizer.cor	10/25/2016 19:58	7.91066E+17
Acham q o Sabonis tem capacidade para "explodir" este ano nos meus Thunder? Gosto da su	10/25/2016 19:58	7.91066E+17
I liked a @ YouTube video from @ thunders7ruck http://youtu.be/c2QQxhZt0dM?a 402Thun	10/25/2016 19:58	7.91066E+17
Russell Westbrook: Latest News, Rumors and Speculation Surrounding Thunder PG?? https://	10/25/2016 19:57	7.91066E+17

Figure.5 Sample Dataset

Initial text analysis

Since we do not know which methodology has the best performance for our Twitter project, we decide to try three possible ways (Highest Frequency Words, Injury and Recovery Factors and Sentiment Analysis) to test our dataset as our first step and find out that sentiment analysis has the best performance at the end. Therefore, we determine to implement **Sentiment Analysis** with five algorithms.

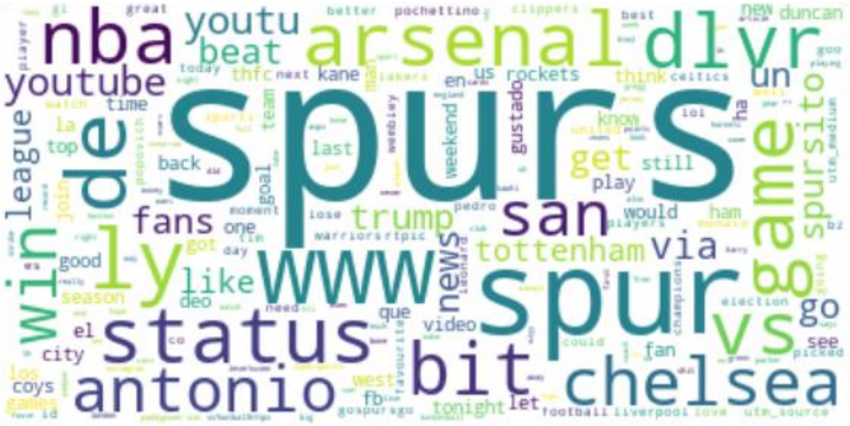


Figure.6 Word Cloud

❑ Keywords Analysis

Target: Try to find out the **Highest Frequency Words** of these tweets.

- **Python File Name:** KeyWord_Analysis.py
- **Data:** Team "Spurs" : 31 days in 2016-11

1. First, we take team "Spurs" and "November" as our demonstration.
2. Separate the entire dataset into three subset datasets based on the time period and they are "3 days", "7 days" and "1 month" individually.
3. Set constraint words such as 'twitter','http','com','pic','ift','tt','https' to clean our dataset.
4. Print out the 20 highest frequency words according to the time periods.
5. After seeing the results of 20 highest frequency words in each time period, we decide to take 5 most meaningful words in the demonstration.

Consequence:

★ 3 days a loop

3 days	**No.1** frequency word (counts)	**No.2** frequency word (counts)	**No.3** frequency word (counts)	**No.4** frequency word (counts)	**No.5** frequency word (counts)
First 3 days	spurs(4480)	arsenal(461)	wembley(405)	game(389)	leverkusen(323)

4~6 days	spurs(4888)	clippers(2913)	arsenal(1354)	bit(492)	win(363)
7~9 days	spurs(4026)	rockets(430)	arsenal(284)	parker(207)	trump(194)
10~12 days	spurs(4165)	trump(727)	election(443)	rockets(318)	popovich(289)
13~15 days	spurs(3853)	antonio(300)	spursito(292)	heat(225)	trump(197)
16~18 days	spurs(4483)	duncan(623)	bit(543)	west(506)	ham(483)
19~21 days	spurs(4224)	west(426)	ham(414)	kane(356)	bit(305)
22~24 days	spurs(4795)	league(470)	champions(339)	chelsea(326)	hornets(307)
25~27 days	spurs(4953)	chelsea(1123)	celtics(446)	leonard(361)	los(344)
28~31 days	spurs(3057)	know(625)	goal(624)	let(612)	weekend(592)

★ 7 days a loop

7 days	**No.1** frequency word (counts)	**No.2** frequency word (counts)	**No.3** frequency word (counts)	**No.4** frequency word (counts)	**No.5** frequency word (counts)
First week	spurs(11161)	arsenal(2000)	game(779)	win(632)	beat(528)
Second week	spurs(9346)	trump(1050)	rockets(803)	status(790)	game(691)
Third week	spurs(10012)	west(963)	ham(919)	bit(914)	win(709)
Last week	spurs(11415)	chelsea(1636)	goal(694)	know(689)	favourite(576)

★ 1 month

1 month	**No.1** frequency word (counts)	**No.2** frequency word (counts)	**No.3** frequency word (counts)	**No.4** frequency word (counts)	**No.5** frequency word (counts)
31 days	spurs(43324)	arsenal(2980)	game(2862)	bit(2618)	win(2188)

Result:

1. We can easily find out that ‘spurs’ has the highest frequency in this demonstration which actually make sense because we take team “Spurs” as our sample dataset.
2. Apparently, small testing dataset (3 days) is more useful than larger dataset (1 month) because there are more information about players’ names, win/lose and other teams’ names.
3. We can tell the condition of team "Spurs" during certain periods based on the keywords:

    a. For instance, we can tell that Spurs fans like to watch soccer games because ‘arsenal’ is a soccer league and it appears thousand times.
    b. In addition, ‘win’ and ‘beat’ represent team "Spurs" get the victory from their opponents ‘clippers’, ’rockets’ and ’hornets’.
    c. Moreover, ‘trump’, ’win’ and ‘election’ shows several times which means Mr.Trump may win the president election in the US.

Exploratory Data Analysis (EDA)

❏ Injury and Recovery Factors:

In this part, we were interested in finding some influential factors which would affect the results of the games. Therefore, we consider if there were injured players or recovered players in the team. We try to find some key words about injury and count the appearances of these words. At the end, we compare these counts with real results of the games and find the relationship between these teams.

- **Python File Name:** Analysis_injury.py
- **Usage:** Gain the injury reports or recovery reports from tweets
- **Data:** Take team "Jazz" dataset from 12/10/2016 to 12/18/2016 for testing.

Step1:

Using the following list to find the keywords:

➢ injury words(**Negative**):
['hurt','injury','injured','broken','tear','missed','ill','illness']

➢ recovery words(**Positive**):
['recover','recovery','return','health','healthy','heal','back','rehab']

Step2:

1. Count the words in these two lists.
2. If the numbers of recovery words (positive) greater than injury words (negative) which means it is a good expectation for the team "Jazz".

Consequence:

Utah Jazz									
Dec 2016	10	11	12	13	14	15	16	17	18
Recovery Count	**12**	**7**	**15**	**16**	**19**	**37**	**12**	**13**	**7**
Injury Count	**18**	**5**	**3**	**1**	**4**	**7**	**6**	**6**	**1**
Game Consequence	**W**				**W**		**W**		**W**

Figure.7 Injury and Recovery Counts Dataset

1. In this form, we take team "Jazz" as a sample data and its duration is from 12/10/2016 to 12/18/2016.

2. Obviously, we can tell the counts of positive words are **greater** than the counts of negative words in most of these days. In addition, team "Jazz" got 4 straight wins during this period.
3. Therefore, we assume that injury tokens are more related to the victories and defeats, and we will try to do more detail research in order to improve accuracy.

Predict Match Result with Sentiment Analysis:

- **Python File Name:** SentimentAnalysis.py
- **Data:** The tweets that 24 hours before the competition.

❑ **Method 1:**

1. First, we count the amount of positive/negative words by each tweet.
2. Once the amount of positive words greater than negative words we treat it as a good result.
3. If good results are more than bad results 24 hours before the competition, we predict the result of the game is to win.
4. Finally, we compare our forecast model with the actual results of the games and calculate the accuracy.

❑ **Result**:

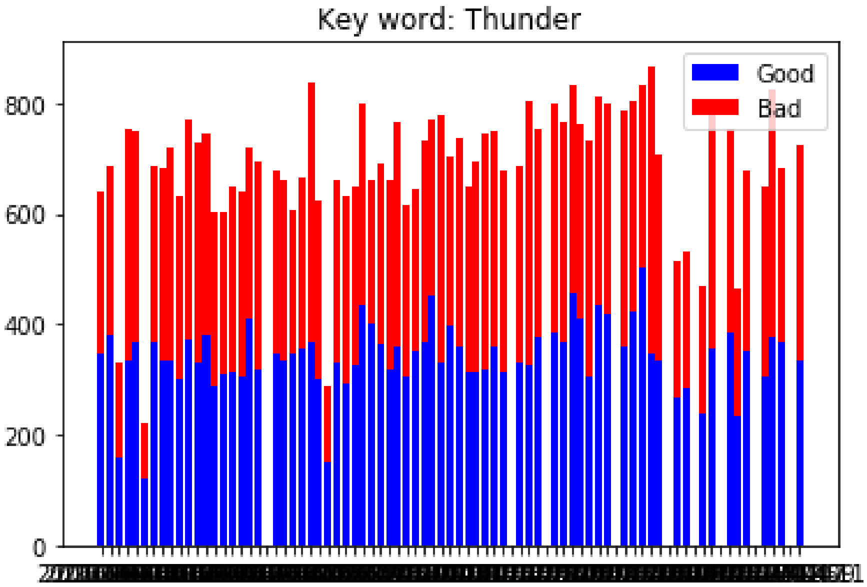


Figure.8 Prediction Result
(Where X-axis is the date of competition, Y-axis is the amounts of good/bad)

➢ Total Predict games: 71

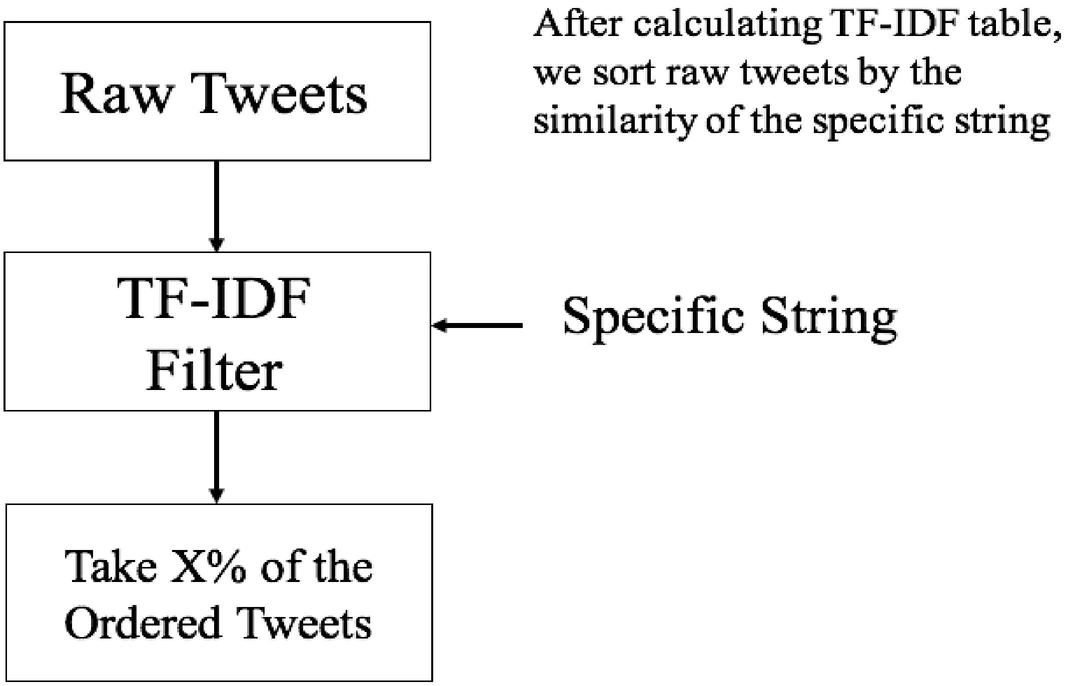
➢ Accuracy: 53.52%

Ratio： Abs(goodCnt – badCnt) / (goodCnt + badCnt)

Predict games	Ratio: Good and Bad	**Accuracy**
4	20%	**75%**
8	15%	**62.5%**
16	10%	**56.25%**

❑ **Method 2:**

1. In order to increase accuracy, we use TF-IDF model to filter some useless twitters.
2. We set the keyword like “NBA” and calculate the similarity table. Then take some part of the ordered tweets to count the sentiment.



After calculating TF-IDF table, we sort raw tweets by the similarity of the specific string

Figure.9 TF-IDF

❑ **Example**:

➢ Specific String: ”NBA”

➢ Take 60% of the ordered

❑ **Result:**

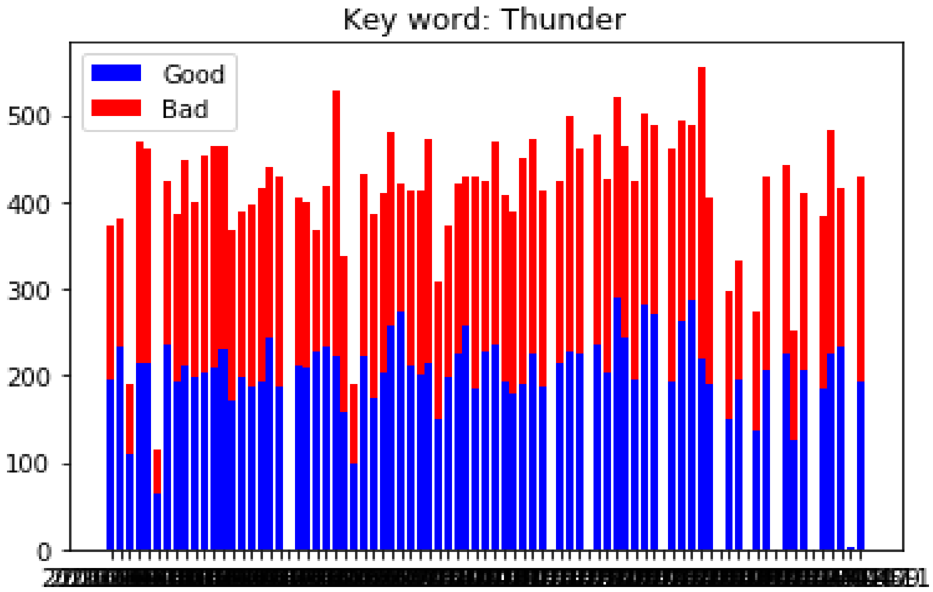


Figure.10 Prediction Result

➢ Total Predict game: 71

➢ Accuracy: 52.11%

Ratio： Abs(goodCnt – badCnt) / (goodCnt + badCnt)

Predict games	Ratio: Good and Bad	**Accuracy**
5	20%	**80%**
11	15%	**72.73%**
23	10%	**65.22%**

Conclusion:

1. Comparing the results of method 1 and 2, we notice something interesting which is if we use TF-IDF model to filter tweets it will reduce the quantity; however, the accuracy increases.
2. In addition, we have better accuracy when the good/bad ratio is high. In other words, if the amount of emotions differs greatly, the forecast will be more accurate and robust.

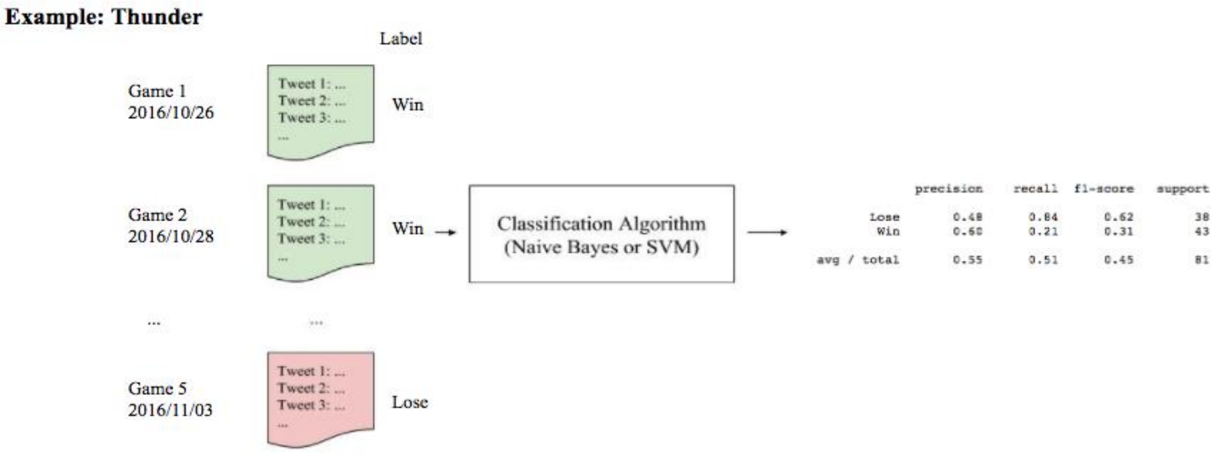
Real NBA Game Predictions:

- **Python File Name:** PredictMethod_CNN.ipynb
- **Data:** The tweets that 24 hours before the competition.

At this part, we implement two supervised and unsupervised algorithms to classify our daily tweets and then make our predictions to check if the results match. We come across an interesting idea during discussing. We believe the emotions of people is a crucial factor which may be able to help us predict the results well. However, it is hard to set the conditions or thresholds for different matches to predict the results. Therefore, we imagine the sentiments of people in the tweets is a big picture and we conduct CNN methodology to recognize the features of this picture and finally it works with high accuracy and it help us to predict the result successfully.

Supervised learning (Naive Bayes, SVM):

We collect tweets before the game 24 hours and label them based on the real game results. We implement Naive Bayes and SVM function provided by sklearn package to calculate f1-scores.

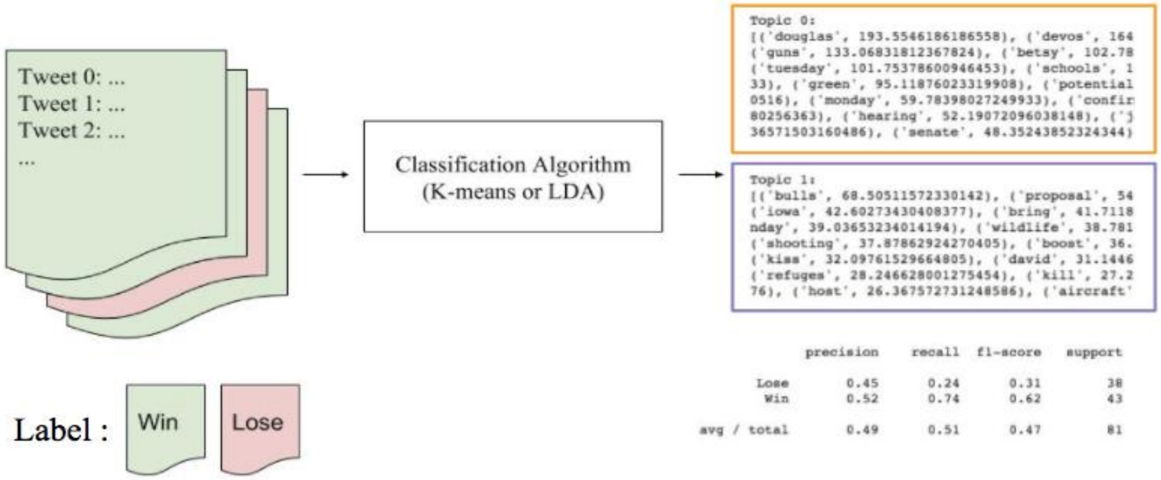


```
             precision    recall  f1-score   support

       Lose       0.48      0.84      0.62        38
        Win       0.60      0.21      0.31        43

avg / total       0.55      0.51      0.45        81
```

Figure.11 Performances of NB and SVM

Unsupervised Learning (K-means, LDA):

We collect tweets before the game 24 hours and label them based on the real game results. The training data is similar to the one in the supervised learning. Then we conduct K-means and LDA function provided by sklearn package to calculate f1-scores. However, we cannot understand the meaning of words and we decide to calculate f1-scores twice as our approach. We assume topic 0 as winning results for the first time and losing results for the second time. At the end, we pick higher f1-score as our result.



```
Topic 0:
[('douglas', 193.5546186186558), ('devos', 164
('guns', 133.06831812367824), ('betsy', 102.78
('tuesday', 101.75378600946453), ('schools', 1
33), ('green', 95.11876023319908), ('potential
0516), ('monday', 59.78398027249933), ('confir
80256363), ('hearing', 52.19072096038148), ('j
36571503160486), ('senate', 48.35243852324344)
```

```
Topic 1:
[('bulls', 68.50511572330142), ('proposal', 54
('iowa', 42.60273430408377), ('bring', 41.7118
nday', 39.03653234014194), ('wildlife', 38.781
('shooting', 37.87862924270405), ('boost', 36.
('kiss', 32.09761529664805), ('david', 31.1446
('refuges', 28.246628001275454), ('kill', 27.2
76), ('host', 26.367572731248586), ('aircraft'
```

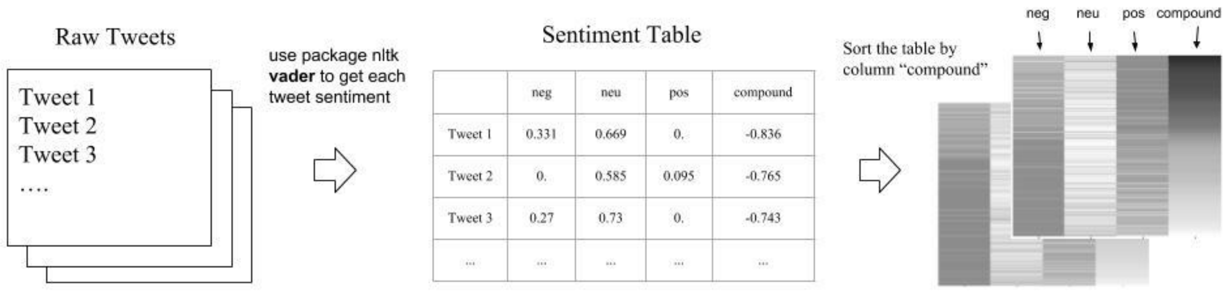
```
             precision    recall  f1-score   support

       Lose       0.45      0.24      0.31        38
        Win       0.52      0.74      0.62        43

avg / total       0.49      0.51      0.47        81
```

Figure.12 Performances of K-means and LDA

Convolutional Neural Network (CNN):

In order to get the good performance of our predictions, we need to deal with thousands of tweets for each day. Our idea is to collect the sentiments from tweets and convert those sentiments to an image and train it with CNN algorithm. As a result, the image can help us recognize the features which stand for a win or a lose.

Step1: We conduct Vader package to get sentiments from each tweet and build the sentiment table


Raw Tweets
Tweet 1
Tweet 2
Tweet 3
….
use package nltk **vader** to get each tweet sentiment
Sentiment Table
Sort the table by column "compound"
neg
neu
pos
compound

	neg	neu	pos	compound
Tweet 1	0.331	0.669	0.	-0.836
Tweet 2	0.	0.585	0.095	-0.765
Tweet 3	0.27	0.73	0.	-0.743
...	...	...	...	...

Figure.13 CNN Workflow

Step2: Build our own CNN structure (In our CNN structure, we have two main layers to train and recognize the images)

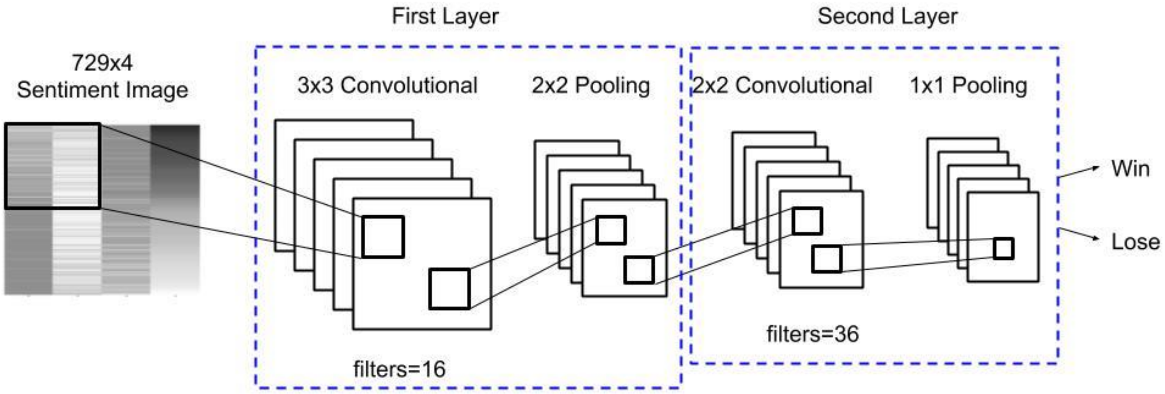


Figure.14 CNN Structure

Result: We have 236 match results and split 90% of them as our training dataset. The accuracy of our CNN model is around **55~68%**

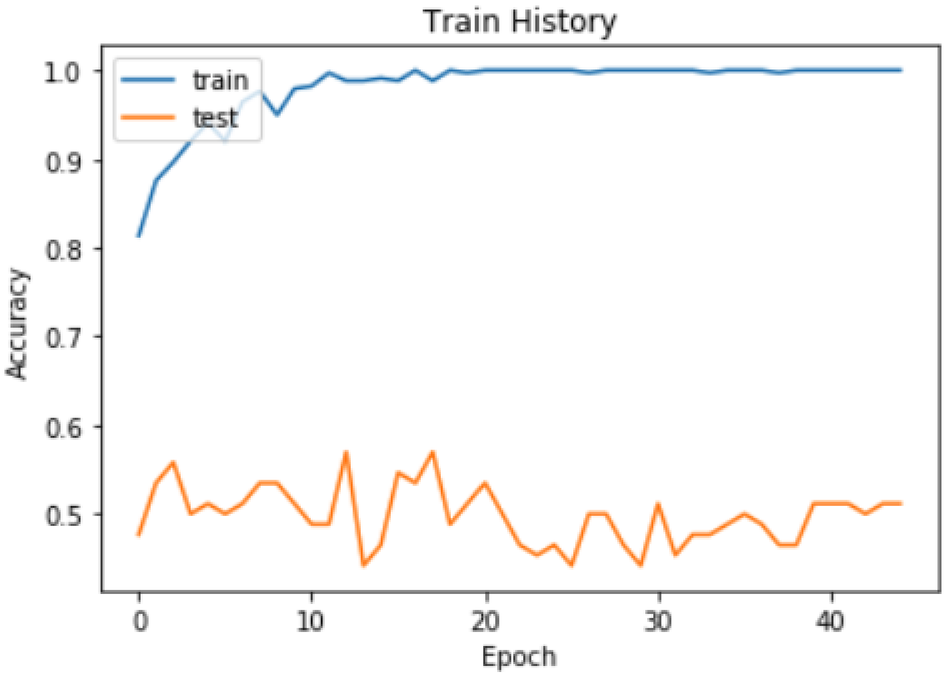


Figure.15 The Result of CNN

Analysis of Experiment Results:

- **Naive Bayes (NB):**

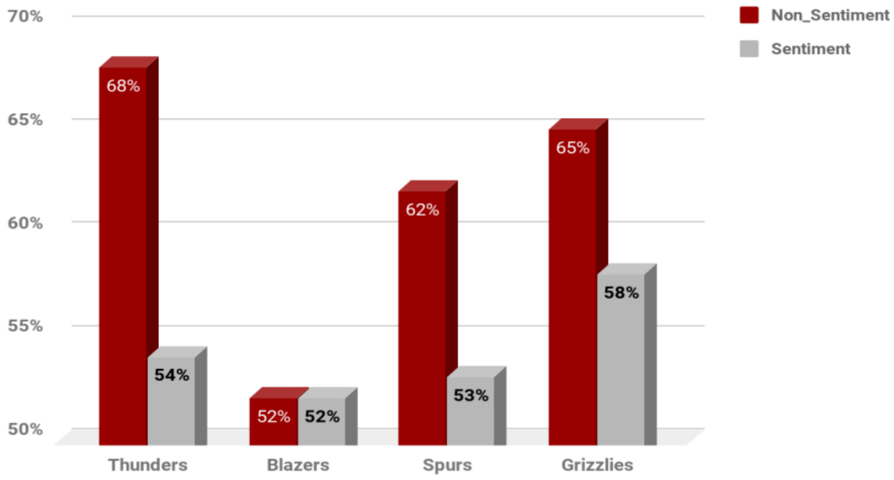


Figure.16 Sentiment Analysis

From this bar chart, we can tell that most performances (accuracies) without sentiment analysis are better than those with sentiment analysis even the accuracy of the lowest one such as team Blazers.

❑ **Support Vector Machine (SVM):**

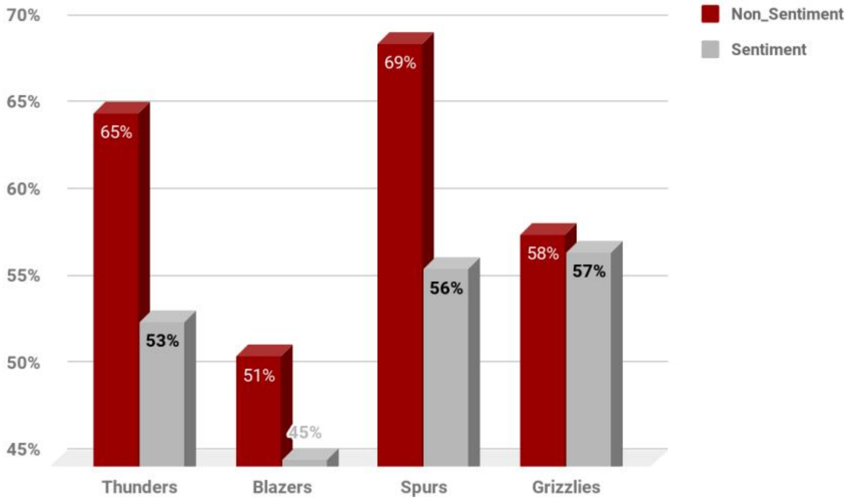


Figure.17 Sentiment Analysis

From our SVM experiment, we can get the similar results with Naive Bayes. The performance (accuracies) without sentiment analysis are significantly higher than those with sentiment analysis.

- **Comparison with each algorithm:**

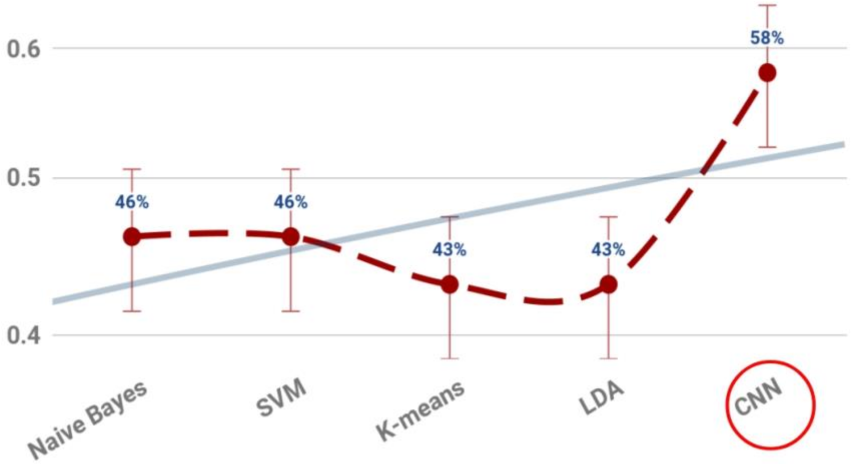


Figure.18 Comparison with each algorithm

This is the figure of comparison with five different algorithms. We can tell that supervised learning algorithms (Naive Bayes and SVM) have better performance than unsupervised learning algorithms (K-means & LDA). However, CNN still has the highest accuracy more than other algorithms.

- **Business Insights with our analysis:**

If we can successfully build a concrete model with high accuracy, we can borden our model not only in NBA but also in other sport games such as NFL, MLB and NHL. In addition, there will be more business opportunities for sport fans and industries to create these great situations (win win). For instance, more and more companies will focus on the combination of game predictions and sports lotteries. In the meantime, sport fans can enjoy the advantages of our analysis.

Future Plan

- **More Effective Keyword**

To be honest, we do not get too much useful information from High Frequency Words and Injury/Recovery Words. Thus, in our future work, we want to improve the process from these two ways.

For high frequency words, we consider to clean more meaningless words such as ‘youtube’ or ‘ham’.

For injury and recovery words, instead of doing more words cleaning, we intend to use superstar players as our keywords. If we find one of them is in the injury list, we label this team as negative level. In the best case, we hope that we can combine all of these three text analysis, then put them into algorithms. We assume that the performance will be more precise than the performance now.

❑ Improve Unsupervised Learning Accuracy

For unsupervised learning, we just simply add all tweets before the game 24 hours as a document. We found that there are lots of same words in the two topics. For further research, we can remove the same words in our original documents and this may help us classify the topic more accurately.

❑ More Parameters for CNN Training

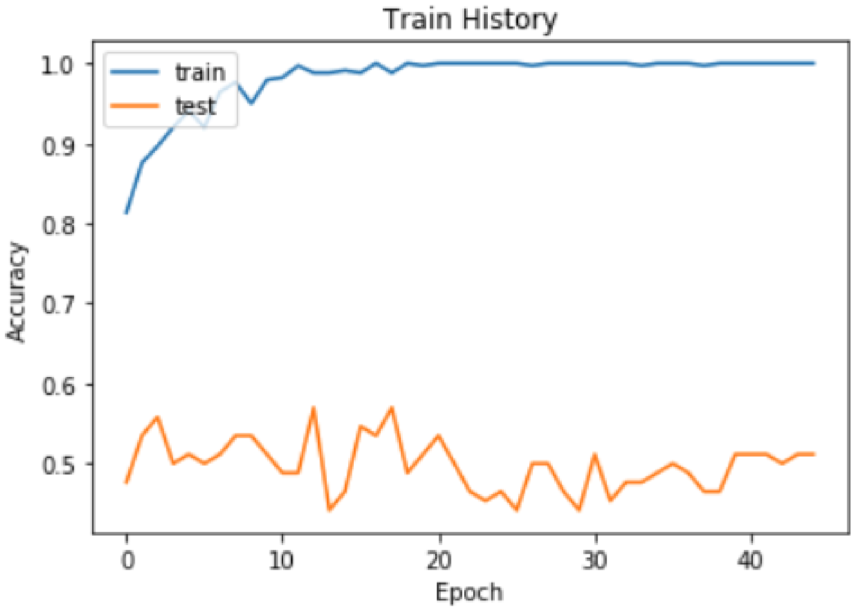


Figure.19 CNN Model

In our CNN model, we can tell that the best performance (accuracy) of our training and testing model is around 50~60% which means the sentiments of people is random in each game. To get better accuracy, we need to find other ways to improve the accuracy. We think we can add more parameters to help us train our CNN model. In other words, the parameters we used to train CNN must relate to the match results.