Project 4

Popularity Prediction on Twitter

EE239AS Winter 2016

Xin Shen	UID: 704591627
Jiawei Wang	UID: 704592368
Lizhi Zeng	UID: 304593058

Introduction

A useful practice in social network analysis is to predict future popularity of a subject or event. Twitter, with its public discussion model, is a good platform to perform such analysis. With Twitter's topic structure in mind, the problem can be stated as: knowing current (and previous) tweet activity for a hashtag, can we predict its tweet activity in the future? More specifically, can we predict if it will become more popular and if so by how much? In this project, we will try to formulate and solve an instance of such problems.

The available Twitter data is collected by querying popular hashtags related to the 2015 Super Bowl spanning a period starting from 2 weeks before the game to a week after the game. We will use data from some of the related hashtags to train a regression model and then use the model to making predictions for other hashtags. To train the model, you need to prepare training sets out of the data, extract selected features for them, and then fit the regression model on it. The regression model will try to fit a curve through observed values of features and outcomes to create a predictor for new samples. Designing and choosing good features is one of the most important steps in this process and is essential to getting a more accurate system. There are examples of such analysis and useful features in literature[1](You should look into the literature for this). You will be given training data to create the model, and test data to make predictions. The test data consists of tweets containing a hashtag in a specified time window, and you will use your model to predict number of tweets containing the hashtag posted within one hour immediately following the given time window.

Problem 1

Calculate statistics for each hashtag:

To calculate the statistics, we read the data line by line. Because the data is huge, if we load all the data into python to compute, the memory will not be enough and this lead the program to an endless wait. Therefore, we only need to select the information we want from one tweet by another for each hashtag. Since each tweet is a JSON string, we invoke ***json.loads(str)***:

```
hashtag=open("tweet_data/tweet_[#hashtag].txt”,"r")
import json

for line in hashtag:
   tweet=json.loads(line)
```

Now ***tweet*** is a python dictionary and changes every time as we read line by line of ***tweet_[#hashtag].txt***. In another word, ***tweet*** contains the information of each tweet.

To calculate the average number of tweets per hour, we first set up our time period. As we have stated in the Introduction part, the period started from 2 weeks before the game to a week after the game, which is from 15:00 pm 18th Jan, 2015(PST) to 15 pm 8th Feb, 2015(PST). And we transform this time period into timestamp:

```
import time,datetime

begin_time=datetime.datetime(2015,1,18,15,0,0)
mintime=int(time.mktime(begin_time.timetuple()))
end_time=datetime.datetime(2015,2,8,15,0,0)
maxtime=int(time.mktime(end_time.timetuple()))
```

Then the total time(hours) can be calculate as :

(maxtime - mintime) / 3600 = 504

And to select the tweets in this time period, we use ***tweet['firstpost_date']*** to get the first post date of each tweet. If **maxtime>tweet['firstpost_date']>=mintime**, then this tweet is in the time period.

To calculate the average number of followers of users posting the tweets, we get the number of followers of each user of each tweet by ***tweet['author']['followers']***. This attribute means the number of followers of this tweet's author, not the original author if this tweet is a retweet. What's more, we not only need to avoid repeated calculation, but also need to update the number of followers of the user when he/she tweeted again. Because the same user may tweet again or several times in the time period and the number of followers may change as time going. So for a time period, we store the ***user_name*** to classify and the number of followers can be counted as:

```
if tweet['author']['name'] not in user_name:
        user_name.append(tweet['author']['name'])
        number_of_followers.append(tweet['author']['followers'])
else:
        number_of_followers[user_name.index(tweet[‘author']['name'])]
=tweet['author']['followers']
```

Finally, to calculate the number of retweets, we get the data from ***tweet[‘metrics’] [’citations’][’total’]*** for each tweet.

	gohawks.txt	**gopatriots.txt**	**nfl.txt**
Avg. tweet	288.12896825396825	50.051587301587304	467.10714285714283
Avg. followers	1626.2717409862018	1123.176168224299	4256.1993262795495
Avg. retweets	2.015900342246431	1.3815111393007216	1.538951329952171
	patriots.txt	**sb49.txt**	**superbowl.txt**
Avg. tweet	936.2162698412699	1639.6468253968253	2662.2261904761904
Avg. followers	1817.0453996177553	2378.0786156333343	3990.980315927982
Avg. retweets	1.7537792490457833	2.511966620763763	2.3914919337408573

Table.1

From Table.1, we can see that there are more tweets and retweets about supporting Seattle Sea Hawks than New England Patriots. Maybe it is because Hawks is the winner of last year and it has more fans using tweet. Or it may because the fans of Hawks have more followers in tweet than Patriots, so their tweets have more influence. And according to hashtag ‘patriots’, many tweets mentioned this team. Moreover, from the stats data of hashtags ‘superbowl’ and ‘sb49’, we can see that this game is the hit topic in that time period because 2000 plus tweets are talking about it.

Now we plot "number of tweets in hour" over time for #SuperBowl and #NFL. It should be noticed that, we need to reset the counters at every new 1 hour.

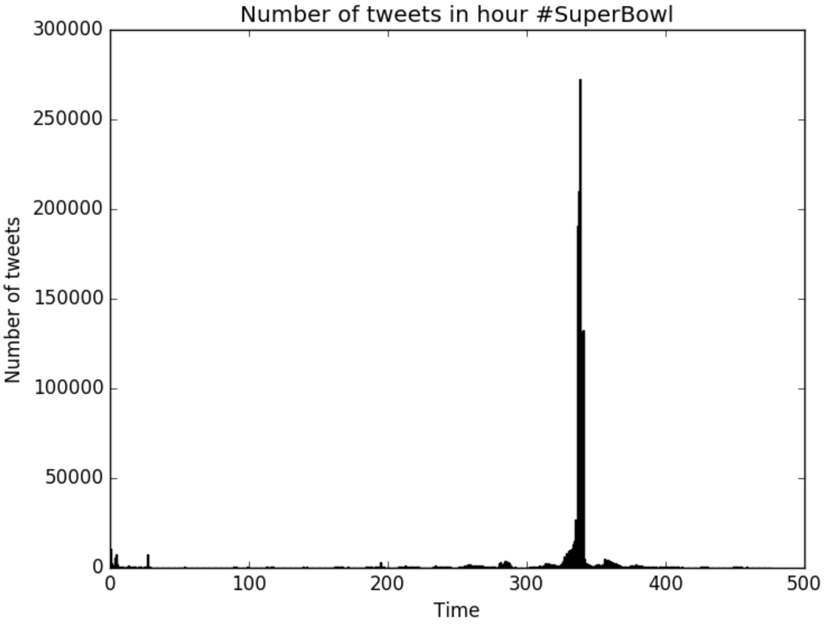


Fig.1

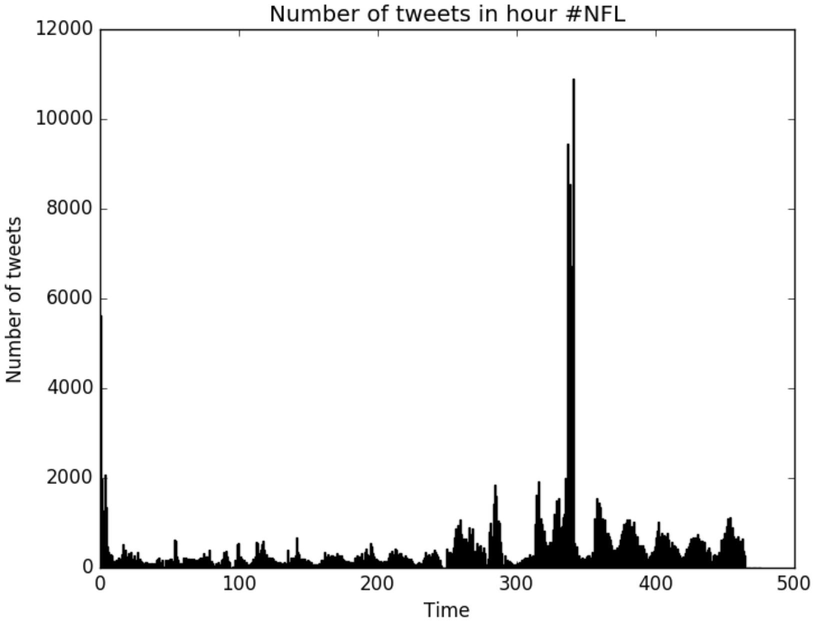


Fig.2

According to Fig.1 and Fig.2, there is a peak at around the 320th~340th hour start from 15:00 pm 18th Jan, 2015(PST), which is the day of the game.

Problem 2

First, we extract the feature data from the dataset by 1-hour-window. This process is just like the process in the second part of problem 1. The only different is that in this problem we need to calculate more features, which are **number of tweets**, **total number of retweets**, **sum of the number of followers of the users posting the hashtag**, **maximum number of followers of the users posting the hashtag**, and **time of the day**, instead of only one feature **number of tweets**. And except for **time of the day**, the other features are calculated as same as in the previous problem.

Consider about the **time of the day**, we should avoid use 0, so we choose 1 to 24 to represent which hour is it of the day. For example, the first hour of our time period starts from 15:00 pm 18th Jan, 2015(PST) to 16:00 pm 18th Jan, 2015(PST), which is the 16th hour of the day. So for this 1-hour-window, the value of **time of the day** is 16.

Since we want to build up a model to predict the number of tweets in the next hour, the train set of the model should be the features in the previous hour and the number of tweets in present hour, and then we can use the features in present hour to make prediction. Therefore, we can split our data set into feature set(data_X) and target set(data_Y) as follow:

```
import numpy as np

data_X=np.array(data_x)
data_X=data_X[:-1]
data_Y=np.array(data_y)
data_Y=data_Y[1:]
```

Where data_X and data_Y are **numpy array**, and each row of them are one-to-one matched as five features and one target.

To build up a linear regression model, we invoke the linear regression model function in ***statsmodels***. And to make our model more precise, we add the constant to reduce the error. Using ***sm.add_constant()*** .

```
import statsmodels.api as sm

data_X=sm.add_constant(data_X)

model = sm.OLS(data_Y,data_X)
results = model.fit()

print (results.summary())
```

Model's training accuracy

To compute model's training accuracy, we input the whole data set of features and target to train our model. After the model is set up, we use the model with features in train set to predict and calculate the average absolute error between the predictive value and the actual value.

However, the training accuracy can only indicate that how well this model is on fitting our train set. It can not represent the accuracy of the model and we shouldn't say the model is good or bad for making prediction in this problem if the training accuracy is small or large. Therefore we use 10-fold cross-validation in ***sklearn*** to test if linear regression model is good on predicting the number of tweets in the next hour with the given five features. And the average absolute error is calculated as:

```
error=abs( model.predict(data_X[testcv])-data_Y[testcv])
results.append(np.array(error).mean())

print(np.array(results).mean())
```

The results are shown as below:

	gohawks	gopatriots	nfl
Avg. Training accuracy	371.68827112111342	87.705423595996564	464.39756560222503
Avg. 10-fold cross-validation	338.96852910536387	79.890989838962298	383.65271596706572
Adj. R-squared	0.634	0.673	0.583
	patriots	**sb49**	**superbowl**
Avg. Training accuracy	1593.067898326291	3317.1506526843741	5265.53647151315
Avg. 10-fold cross-validation	1284.5981683840744	2786.0721793848847	4294.5167295523452
Adj. R-squared	0.708	0.815	0.735

Table.2

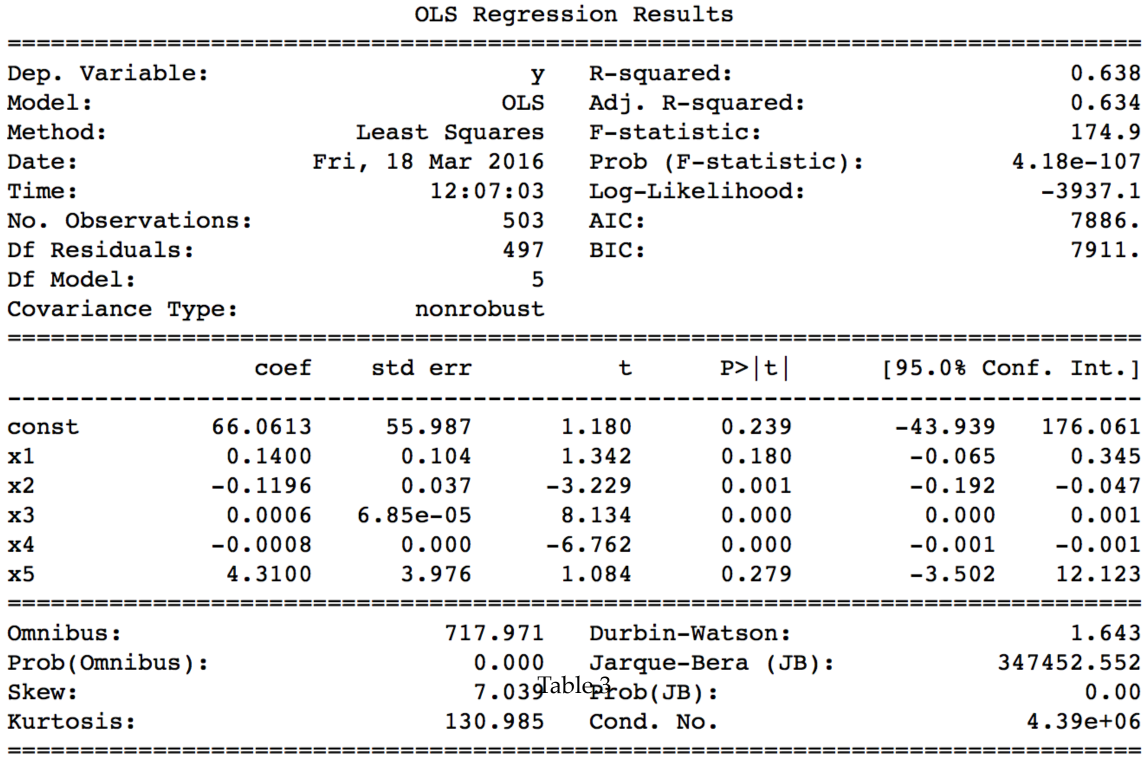
We can see that the training accuracy and 10-fold accuracy are not good by comparing with the value of average tweets in Table.1. And it is interesting that the training accuracy is higher than the 10-fold accuracy. This indicates that our data is not as linear as we expected.

And it can be proved by Fig.1 and Fig.2, since there is a sudden peak. Therefore, in order to do better in making prediction on the number of tweets in the next hour, we should consider about changing features or building up models for different time periods.

T-test and P-value

T-test is an efficient way to analyze the significance of the features. And from T-test score, we can get a succinct representation of features' significance which is called P-value. To be noticed, T-test is a model based test. The T-test score is calculated by dividing coefficient of feature by std error of feature. This means it might be influenced by the model and other features. So for each hashtag, we calculate the P-value of the five features, and from the results, we will know which features are more significant and which are not.

Usually, if the P-value of the feature is smaller than 0.05 then we can say this feature is significant to the target, otherwise it is not that significant.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.638
Model:                            OLS   Adj. R-squared:                  0.634
Method:                 Least Squares   F-statistic:                     174.9
Date:                Fri, 18 Mar 2016   Prob (F-statistic):          4.18e-107
Time:                        12:07:03   Log-Likelihood:                -3937.1
No. Observations:                 503   AIC:                             7886.
Df Residuals:                     497   BIC:                             7911.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const         66.0613     55.987      1.180      0.239       -43.939   176.061
x1             0.1400      0.104      1.342      0.180        -0.065     0.345
x2            -0.1196      0.037     -3.229      0.001        -0.192    -0.047
x3             0.0006   6.85e-05      8.134      0.000         0.000     0.001
x4            -0.0008      0.000     -6.762      0.000        -0.001    -0.001
x5             4.3100      3.976      1.084      0.279        -3.502    12.123
==============================================================================
Omnibus:                      717.971   Durbin-Watson:                   1.643
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           347452.552
Skew:                           7.039   Prob(JB):                         0.00
Kurtosis:                     130.985   Cond. No.                     4.39e+06
==============================================================================
```

Table.3

Table.3 gives the summary of linear regression model for hashtag **gohawks**. From the this table, we find that the **constant(const)**, **number of tweets(x1)**, and **time of the day(x5)** are not that significant as we expected. And **total numbers of retweets(x2)** can be used, but it is not that better as **number of followers(x3)** and **max number of followers(x4)**. This is weird because **number of tweets** in present hour should be an important feature to predict the number of tweets in the next hour. This happens because we use this five features to build up the model, and if we change the features, we may have a different result. In Problem 3, we will give a further analyze.

To reduce the report length, we stop posting the whole summary of the results. Instead, we build up a Table to post the P-value of each features in different hashtags.

	gohawks	gopatriots	nfl	patriots	sb49	superbowl
constant	0.239	0.558	0.048	0.264	0.661	0.789
x1	0.180	0.134	0.000	0.000	0.000	0.000
x2	0.001	0.417	0.027	0.000	0.000	0.803
x3	0.000	0.000	0.000	0.000	0.000	0.000
x4	0.000	0.000	0.000	0.041	0.004	0.000
x5	0.279	0.785	0.638	0.913	0.507	0.670

Table.4

According to Table.4, **time of the day(x5)** is not significant for all hashtags, so it is reasonable to remove it from the feature set. **Number of followers(x3)** and **max number of followers(x4)** perform good. They may be related, but they have different impact. In this project we are considering about a countrywide or even worldwide game, so most of the tweets about the game should be retweets. The **number of followers** stands for the average number of followers of each author which may reflect how many people will read and tweet or retweet about this hashtag. But the **max number of followers** indicates that whether there is someone who have a great impact on the public, and people may want to tweet or retweet because of his/her/their influence. For **number of tweets(x1)** and **total numbers of retweets(x2)**, they perform good in some hashtags but not every, so we will have further test and analyze in next section.

Problem 3

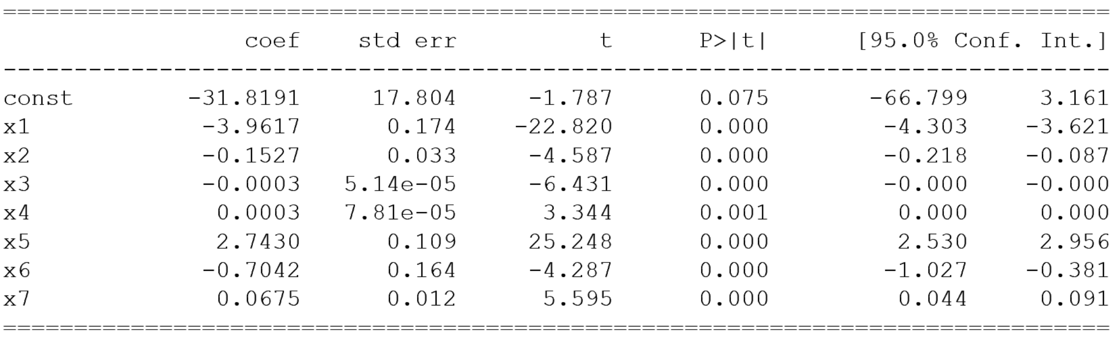
Based on the paper we find and test we have taken, we finally chose 7 features to train the linear regression model, the features are: **number of tweets**, **total number of retweets**, **number of followers**, **max number of followers**, **number of hashtag**, **user mentions**, and **favorite count**.

Here we give a brief explanation about these features. Number of tweets containing a hashtag can represent current popularity of the hashtag. Total number of retweets can show that the level of the content of the tweet has successfully attracted the other users, and the retweeting behavior of a user can also affect his followers. When any followers see a tweet and feel interested, he may retweet or post new tweet using that hashtag, thus the number of followers may become a potential scale of adoption of the hashtag. In a tweet, a user may use several hashtags to show his or her interest in this topic, so the number of hashtag can also reflects the popularity of a topic. User mentions shows if a user was mentioned, and as we know the mentioned people are more likely in taking part into the topic, so he/she may tweet or retweet. Favorite count means that if you like one's tweet, you may click the favorite button to show your interest in that tweet, so the number of favorite a tweet got can show its popularity as well.

The process of setting our data set is similar to the process we have done in problem 2. The only change is we delete the feature **time of the day**, and add 3 new features which are **number of hashtag**, **user mentions**, and **favorite count**.

After we get the new data set, we then can use them to train our model and calculate the P-value to find the top three features of each hashtag.

gohawks:

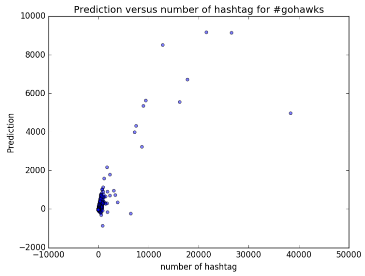


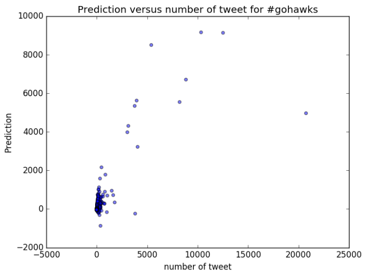
```
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const        -31.8191     17.804     -1.787      0.075       -66.799     3.161
x1            -3.9617      0.174    -22.820      0.000        -4.303    -3.621
x2            -0.1527      0.033     -4.587      0.000        -0.218    -0.087
x3            -0.0003   5.14e-05     -6.431      0.000        -0.000    -0.000
x4             0.0003   7.81e-05      3.344      0.001         0.000     0.000
x5             2.7430      0.109     25.248      0.000         2.530     2.956
x6            -0.7042      0.164     -4.287      0.000        -1.027    -0.381
x7             0.0675      0.012      5.595      0.000         0.044     0.091
==============================================================================
```

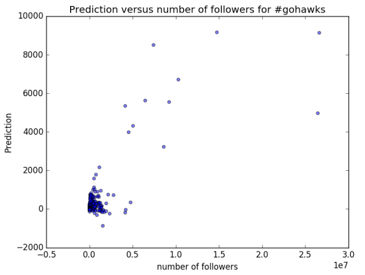
Table.5

The fitting accuracy of the model for *#gohawks* is 88.0%. According to the summary of the results shown in Fig.3, the top three significant features are **number of hashtag**, **number of tweet** and **number of followers**.

We then plot the predicted number of tweets for next hour versus top 3 feature value are shown as below, and we can see the first two features are kind of in a linear pattern.

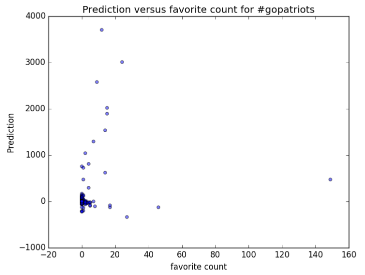


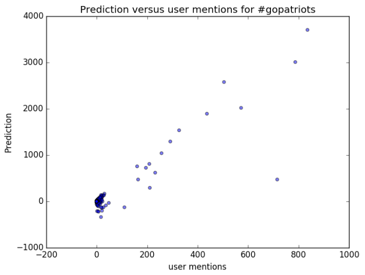


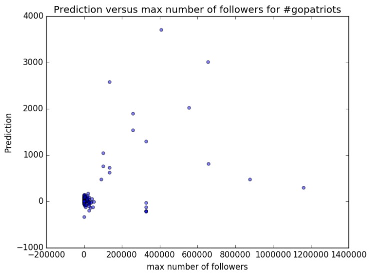


gopatriots:

The fitting accuracy of the model for *#gopatriots* is 83.8%. According to the summary of the results same before, the top three significant features are **favorite count**, **user mentions** and **max number of followers**.



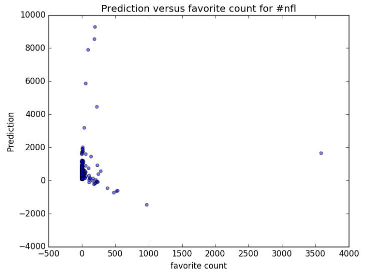


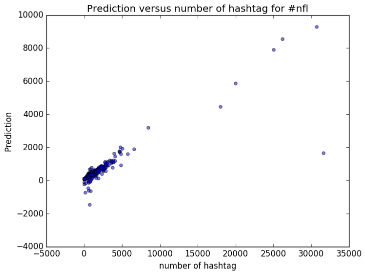


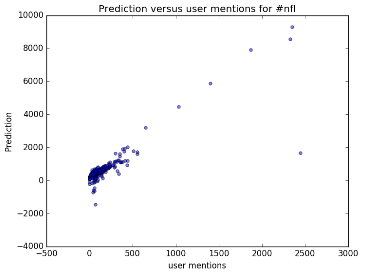
The second feature is vary close to the linear patten. But the first one and the third seems bad. However, we can not draw the conclusion that arbitrary, because there are some huge value points, and we cannot see what the small overlapped area looks like.

nfl:

The fitting accuracy of the model for #nfl is 74.2%. According to the summary of the results, the top three significant features are **favorite count**, **number of hashtag** and **user mentions**.

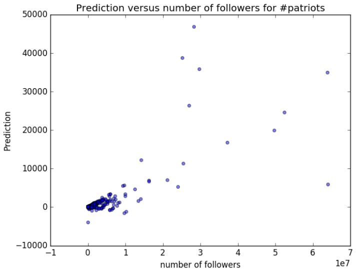


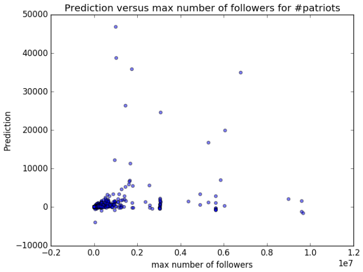


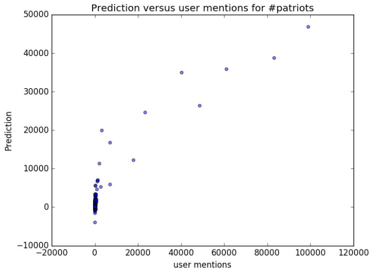


patriots:

The fitting accuracy of the model for #patriots is 75.3%. According to the summary of the results, the top three significant features are **number of followers**, **max number of followers** and **user mentions**.

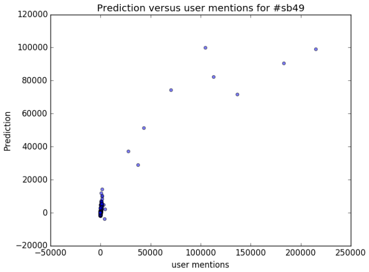


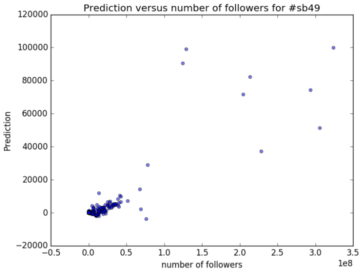


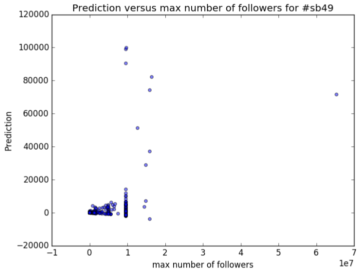


sb49:

The fitting accuracy of the model for #sb49 is 84.7%. According to the summary of the results, the top three significant features are **user mentions**, **number of followers** and **max number of followers**.

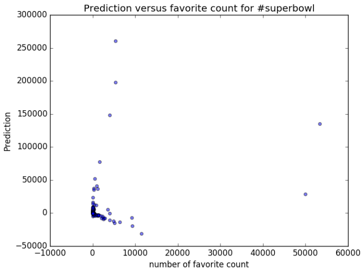


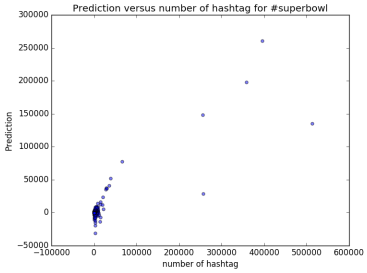


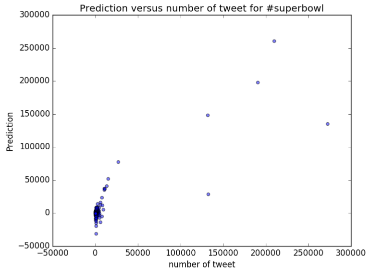


superbowl:

The fitting accuracy of the model for #superbowl is 87.0%. According to the summary of the results shown in Fig.8, the top three significant features are **favorite count**, **number of hastag** and **number of tweet**.







Problem 4

Based on above work, we have 504 hours data set for each hashtag. To separate the data set into the 3 requested time period. We first set up the time point:

```
import time,datetime

begin_window=datetime.datetime(2015,2,1,8,0,0)
min_window=int(time.mktime(begin_window.timetuple()))
end_window=datetime.datetime(2015,2,1,20,0,0)
max_window=int(time.mktime(end_window.timetuple()))
```

Then the time period between our start time 15:00 pm 18th Jan, 2015(PST) and the first time point 8:00 am 1st Jan,2015(PST) is the first time period. The time between our two time points is the second time period. The time between the second time point and the end time is the third period. The operation can be done as follow:

```
# mintime -> 2015,1,18,15,0,0
# maxtime -> 2015,2,8,15,0,0

p1=int((min_window-mintime)/3600)
period1_dataX=data_X[:p1]
period1_dataY=data_Y[:p1]

p2=int((max_window-mintime)/3600)
period2_dataX=data_X[:p2]
period2_dataX=period2_dataX[-12:]
period2_dataY=data_Y[:p2]
period2_dataY=period2_dataY[-12:]

p3=int((max_window-mintime)/3600)
period3_dataX=data_X[p3:]
period3_dataY=data_Y[-p3:]
```

For each time period, we use the selected features which we get in problem 3, and use linear regression method to build up the model.

To use 10-fold cross validation for testing the model's accuracy, we use ***sklearn*** by doing the same process as we have dong in problem 2.

Finally, we represent the accuracy of 10-fold cross validation by calculating the absolute error of each test in the 10-fold. Results are given in Table.6

	gohawks	gopatriots	nfl
Period 1	201.68002176247228	19.731706791	231.313299585
Period 2	2675.3779387031527	903.177428447	3537.10886758
Period 3	117.86571407180718	6.40898583348	170.6181277

	patriots	sb49	superbowl
Period 1	378.69175143121095	144.535843569	554.250482071
Period 2	13661.433096883618	41170.9410132	52528.8839686
Period 3	185.05878438155281	139.276021897	541.243831329

Table.6

From Table.6 , we find that in all hashtags, the accuracy of 10-fold cross validation is bad. This is mainly because the value of target which is the number of tweets in hour change violently during that time period. And it is reasonable because it is the game day, and there must be a extreme peak around 15pm (PST) when the game begin. This explains why the performance of our model is bad in the second period.

What's more, according to the data in period 1 and period 2, the accuracy is low and this indicates that the change in period 1 and 3 is relatively smooth.

Till now, we build up 18 models, 3 models for each hashtag. And these models will be used in the next problem.

Problem 5

Based on the models we have set up in the last section, we load different data sets which are 6 for each hashtag from the same test data file.

To find out the suitable model for the 6 hours test set, we choose one model from the three for each hashtag based on the time period which we can know from the file name.

Then we use the previous 5 features as **tset_X** and last 5 targets as **test_Y**, because to predict the number of tweets in the 7th hour, we only need the features in the 6th hour, so we can use **tset_X** and **test_Y** to calculate which model has the least average absolute errr, and then the least error one is the most suitable one.

For example, for ***sample1_period1*** we select the 7 features we have discussed in problem 3, and calculate the average absolute error of the predict value by using each hashtag's model in period 1. According to the error, we find that the model of **superbowl** performs best. So we then use this model, with the last feature data we select from the data file to make prediction.

The predictions of the 7th over for each 6 hours test data are shown as below:

	Sample 1 P1	Sample 2 P2	Sample 3 P3	Sample 4 P1	Sample 5 P1
Prediction	192	86323	834	245	259
	Sample 6 P2	**Sample 7 P3**	**Sample 8 P1**	**Sample 9 P2**	**Sample 10 P3**
Prediction	31131	110	19	2972	53

Table.7

Problem 6

The dataset in hands is very rich as there is a lot of metadata to a tweet. So in this part, we try to extract some useful information from the dataset rather than just predicting popularity as implemented above. In this part, we show more interests at the information of the individuals among the audience. We analysis the distribution of individual's ethnicity or region an individual might come from.

Why we want to do this?

As a famous sports event, Super Bowl attracts hundreds and thousands of audience every year. However, most of the audience are from United States, and thus the Super Bowl can not be called a world level sports event like Olympic Games. There could be some reasons why it is not well-known among the world. For example, in Chinese culture, American football's athletic performances is too intense, so they might not be interested at it. However, in other case, people may love the sports or events, however, because of the poor propaganda, they have no good opportunities to touch it. So our objection is to find out the potential audience.

How can we do it?

We have three presuppositions:

(1) Depended on the big data, Twitter is a good enough platform to reflect information in real life;

(2) United States is a country full of diversities;

(3) People from certain area have the same culture background.

Considering these three aspects, a good point to represent ethnicity or region is the language that people use. Twitter is a platform that supports multiple languages, which we can make a good use of in our application.

To simplify our process. We focus on the largest dataset, 'tweet_#superbowl'. Among the rich metadata inside, ['tweet']['lang'] metadata can help us classify the language people use. The value inside this parameter is usually in the following two conditions:

(1) A language twitter detected. (for example, 'en' stands for English)

(2) Undefined language, which could be 'mixture' of many languages, only containing emoticons, etc. (coded as 'und')

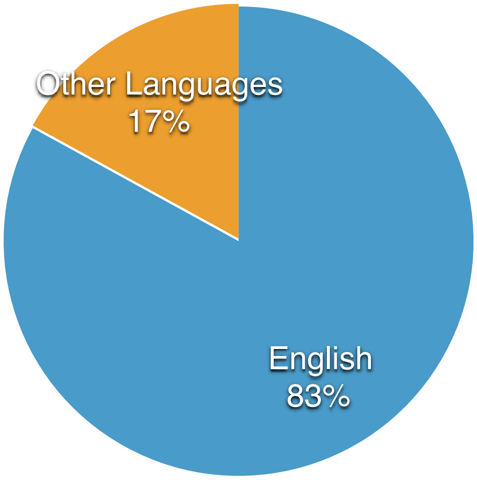
We only consider the first condition as effective data and ignore the tweets that are classified as the second condition. And finally the following table is the statistics we get from #superbowl data, which reports the top 20 languages from twitter users.

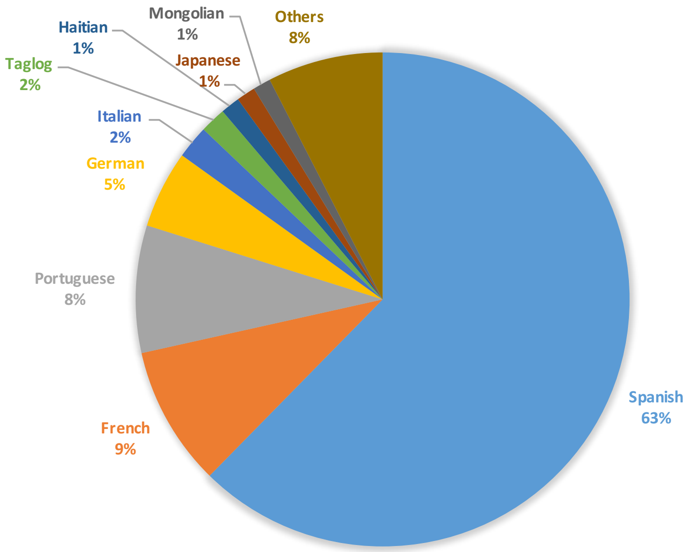
Language Code	Name	Number
en’	English	1055174
‘es’	Spanish	139273
‘fr'	French	20463
‘pt'	Portuguese	18607
‘de'	German	11330
‘it’	Italian	4850
‘tl'	Tagalog	3790
‘ht’	Haitian Creole	2804
‘ja'	Japanese	2769
‘in'	Mongolian	2604
'nl'	Dutch	2089
'lv'	Latvian	1671
‘pl'	Polish	1582
'tr'	Turkish	1498
'et'	Estonian	1392
‘fi’	Estonian	1159
‘sv'	Swedish	1096
'sk'	Slovak	800
‘sl'	Sinhala	750
——	others	4839

Table.8

As we see from the result above, English is the dominant language, not surprisingly. From the following left pie chart, we can clearly see the distribution of English and other languages

About 83% of the twitters are in English, other languages counts only 17%. However, these 17% information is what we care about.





After analyze the distribution in other languages , we got the pie chart as above right. We only take out the top languages among them. The other ones are not big amount and we decide not to consider in this part. The top languages are Spanish, French, Portuguese, German, Italian, Tagalog, Haiti, and Mongolian.

Next, we classify the languages according to the continents and countries where they are spoken as main language:

Europe	Germany, France, Italy, Spain, Portugal
South America	Mexico(Spanish), Colombia(Spanish), Argentina(Spanish)
Asia	Philippine(Tagalog), Japan, Mongolia
Caribbean	Haiti
Others	…

Table.9

Conclusion:

According to our data analysis, we can make a short conclusion that these countries listed above are the most possible ones that Super Bowl would become popular if sponsor of the game can propagate in these regions a little more.

How reliable? It is hard to numerically reflect the accuracy of the result. However, we would point out the possible influences that affect the reliability.

First, we can not rule out the possibility such that people who use twitter is more likely have interested in Super Bowl. In other words, Internet might be different from what the real world is; Second, we supposed that people write certain language on twitter can represent the people that in certain country. This is potentially not true especially if the dataset is limited. But what we can conclude is that, the result can achieve our aim in some degree, which is good enough.

Extended Discussion:

In the discussion above, we come up with a method that using languages in twitter to figure out potential audience of an event. We also put up with some other aspects the twitter data might be used for. However, because of the limited time, we only discuss the basic idea in the last part.

An interesting idea is that we can use the data to access to the supporting ratio of each team. This can be done by analyst the number of twitter of each hash tag, such as #gohawks, #sealhawks stand for Team Hawks, and #gopatriot, #patriot stand for Team Patriot.

Besides the number of twitters for each team, we could also consider the positive or negative attitude of each tweet, which can be down by available developer tool. This kind of gather supporting ratio information is in fact widely applied nowadays.

New research in computer science, sociology and political science shows that data extracted from social media platforms yield accurate measurements of public opinion. In fact the information extract from twitter successfully predict an election. It turns out that what people say on Twitter or Facebook is a very good indicator of how they will vote. Specifically, the study found a correlation between the number of times a candidate for the House of Representatives was mentioned on Twitter in the months before an election and his or her performance in that election. The more a candidate is mentioned on Twitter, the better.