Tweets Analysis of “The walking dead” Using Machine Learning

Yixuan Zhao Alaa Ayach

Introduction

The Walking Dead is a show tells the story of a small group of survivors living in the aftermath of a zombie apocalypse. As *The Walking Dead* becomes more and more popular, we came up with an idea to analysis the tweets in sentiments analysis and gender predication by using machine learning techniques and to show whether they have relationship between these two aspects. Also, by using different parameters, we curious whether different machine learning method will effect the accuracies.

Sentiments analysis is becoming one of the most profound research areas for prediction and classification. Automated sentiment analysis of text is used in fields where products and services are reviewed by customers and critics. Here we use different machine learning models to predict the sentiments of the tweets and compare the accuracies.

Also gender prediction always be focused in different fields. Here we use different machine learning models to predict the gender of users. At last, we combine these two parts and get the distribution of different sentiments with different genders.

In our project, we have two assumptions. First, we assume that people who post the tweets are watching or watched the tv show *The Walking Dead.* Second, we assume that the sentiments of the tweets people post are their truly feeling. They doesn’t post fake feeling tweets.

Data

We collect three sets of the data from twitter.com for this project.

The first set, we use *statuses/user_timeline* api to collect the latest tweets from user *The walking fans* as part of the training data in sentiment analysis.

The second set, we collect the tweets have *The walking dead damn* and *the walking dead sucks* as the other part of the training data in sentiment analysis. We keep all the tweets in 4 txt file according to the date of the tweets. We keep the tweets in two different txt files.

The third set, we collect the tweets have *#The walking dead*. We will predict part of this data in sentiment analysis. We use twitter/search api to collet the data in the second and third parts. We keep the tweets in 4 txt files. For all the three sets, we keep all the information related to the tweets.

Methods

The package of all the methods we use in this project is sklearn.

Merge data & Processing data

After we collect the data, we have a lot of txt files. We need to merge the data. We combine the data and also filter data of the third set data. We have two principle to filter the data. First, we only keep the tweets that the language is only English. Second, we only keep the tweets that the user name of the first name in the census list. We keep both the first set and the second of the data in the pickle file named positive and negative. We keep the filter data of the third set in the pickle file named info1

Processing data

After we collect the data, we processing all the data. First, we just keep the language of tweets are English. Second, we just keep the tweets that the first name of the user name in census list. In this way we can label the gender faster. Third, I process 1/3 of the data in the third set and let them fit the model feature we will build later. I finish this step after I build the model.

Sentiment analysis

1.Training data

The first set data and the second set data as our training data. We assume the sentiment of all the tweets in the first set as positive and the tweets in the second set as negative.

2.Tokenize

We use method, sklearn.feature_extraction.text.CountVectorizer to tokenize the tweets. The matrix has 4,663 tweets and 7,770 terms.

3.Machine learning methods

1) Logistic Regression

We use LogisticRegeression in sklearn.linear_model to build the model of logistic regression. We use different value of parameter C and compare the accuracies. The value we use for parameter C are: 0.01,0.1,0.5,1.0,5.0,10.0,100.0,1000.0.

2) SVC

We use svm in sklearn to build the model of svc. We also use the same value of parameter C and compare the accuracies.

3) Gaussian Naive Bayes

We use GaussianNB in sklearn.naive_bayes to build the model. Without the parameter of C, we only can get the accuracy and without any comparison.

4.Cross-validation

We use function KFold to test the accuracy. We use the same standard to test the training model, KFold=5. Then we get an average cross validation score.

5.Predicate

I process part of the data in the third data set. Let the data feature fit the model feature. In this part, we don't use code to process the data. At last, we get 1/3 of the tweet from the third data set and predict their sentiments. We use these three different method to predict the data.

Gender prediction

1.Tokenize

We split the tweets in different part. The parameters we use are: use_descr, lowercase, keep_punctuation, descr_prefix,collapse_urls, collapse_mentions, use_text. We use different token parameters to train the model. The different parameters are user_text and user_descr represent the tweets and the user description. The three token matrix have: 1) both tweets and user description 2) only description 3) only tweets.

2.Parameter

We use different parameters to build the model and compare the accuracies. The parameters are C, minimum document-frequency,

maximum document-frequency and ngrams. Also, we combine the parameters C with other three parameters.

1) C

The values of parameter C are the same with the values in sentiment analysis.

2) minimum document-frequency

The values of the minimum document-frequency are: 2,4,6,8,10.

3) maximum document-frequency

The maximum document-frequency are:.
00001,.00005,.0001,.0005,.001,.005

4) ngrams

The values of ngrams we use are: (1,1), (1,2), (2,2), (1,3), (2, 3), (3,3)

Combine two methods

1.Afinn analysis and census list

1) Distribution of male and female user

We use afinn analysis and census list to label the gender which we predict before and all tweet we collect in the third data set.

2) Distribution of different sentiments with different genders

After we label the data with male and female. We divide every gender into different sentiments — positive and negative.

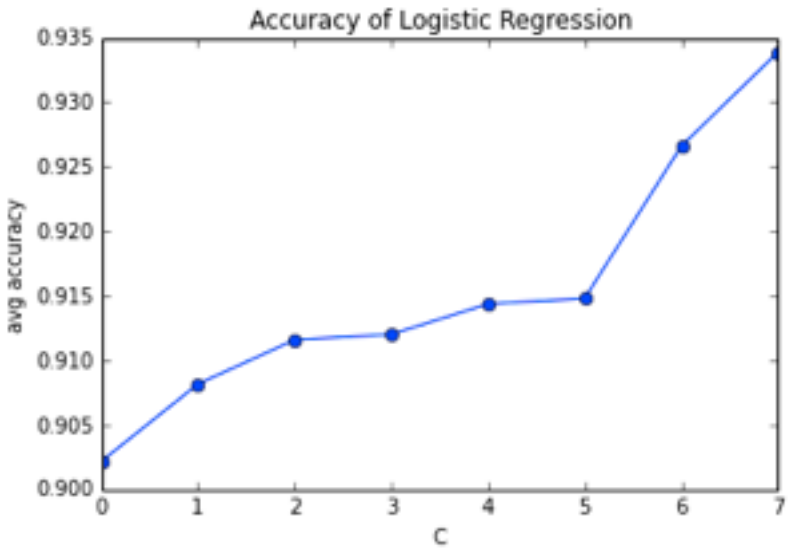
Experiments

Sentiment analysis

1.Comparing different machine learning methods

1)Logistic regression

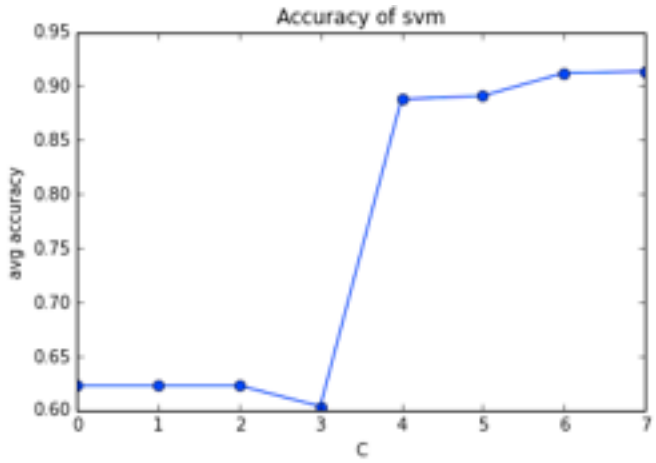
We find that the higher the value C, the higher the accuracy. So when C=0.01, the model has the lowest accuracy and when C=1000.0, the model has the highest accuracy.




```
C: 0.01 avg accuracy 0.01 0.902164092939
C: 0.1 avg accuracy 0.1 0.908160026496
C: 0.5 avg accuracy 0.5 0.911590052855
C: 1.0 avg accuracy 1.0 0.912019237404
C: 5.0 avg accuracy 5.0 0.914379522423
C: 10.0 avg accuracy 10.0 0.914808706972
C: 100.0 avg accuracy 100.0 0.926612202089
C: 1000.0 avg accuracy 1000.0 0.93391040945
```

2)SVC

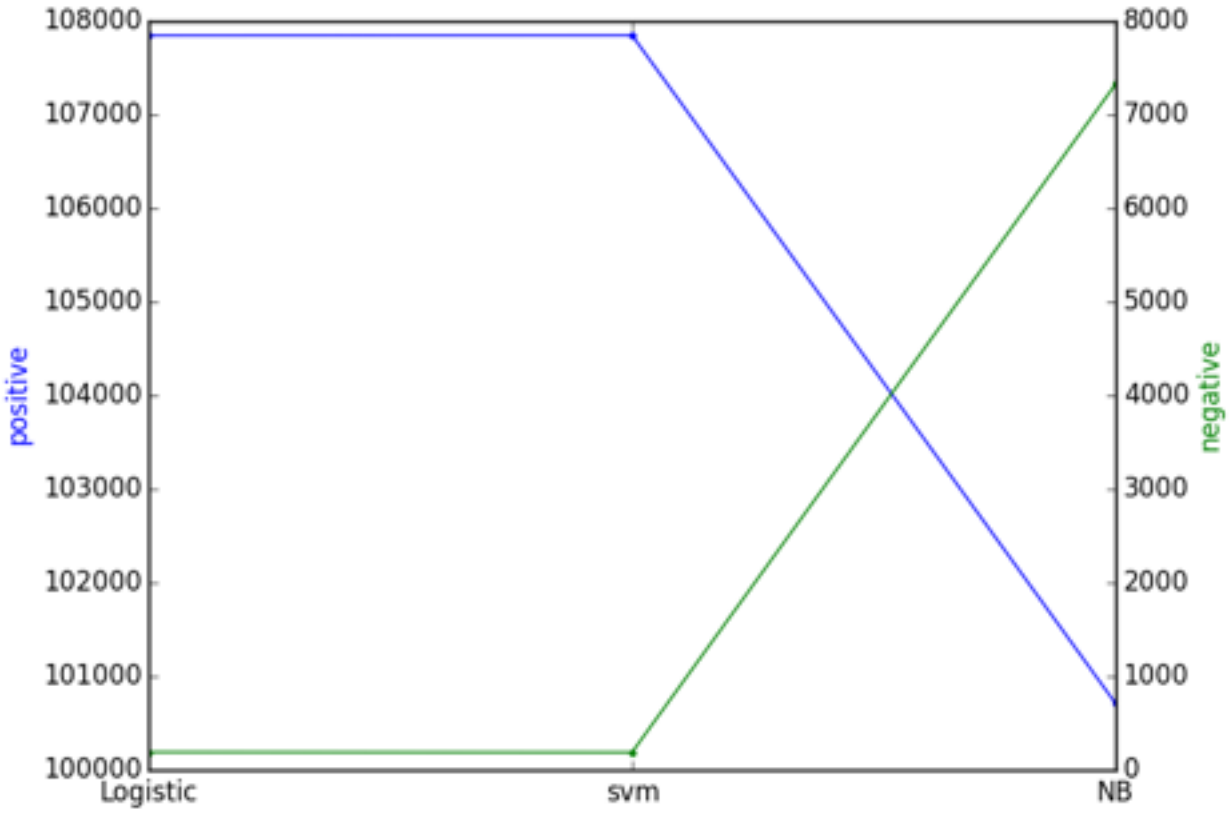
In this figure, we can find that there doesn't have a linear relationship between the value of C and the accuracies. But when C=1000.0, the model has the highest accuracy.




```
avg accuracy 0.01 0.622746781116
avg accuracy 0.1 0.622746781116
avg accuracy 0.5 0.622746781116
avg accuracy 1.0 0.603455326626
avg accuracy 5.0 0.887582628376
avg accuracy 10.0 0.890799672477
avg accuracy 100.0 0.912019237404
avg accuracy 1000.0 0.913520693319
```

3)Gaussian Naive Bayes
There doesn't have any parameter like C in Gaussian Naive Bayes, so we don't have any comparison and the accuracy is 69.22%

2.Predict label
We find that the number of both positive and negative tweets are almost same using Logistic Regression and SVM. But the model Gaussian Naive Bayes has big difference in predicting.



3.Conclusion
The conclusion in this part, we find when C at a certain value, both logistic regression and svm have the highest accuracy and the accuracies are almost the same. So the result using these two methods are almost the same.

Gender prediction

1.cross-validation with different tokens

The figure tells us that when the tokens just have user description, the model has the highest accuracy

```
run_all(test_tweet, use_descr=True, use_text=True)
```

```
113036 unique terms in vocabulary
acc= 0.745353158087
 0.74535315808685498
```

```
run_all(test_tweet, use_descr=True, use_text=False)
```

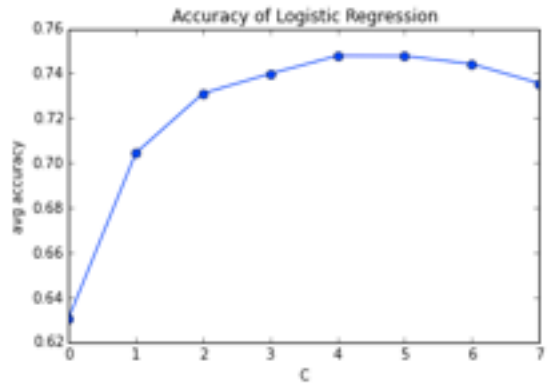
```
84087 unique terms in vocabulary
acc= 0.75453537955
 0.75453537955042105
```

```
run_all(test_tweet, use_descr=False, use_text=True)
```

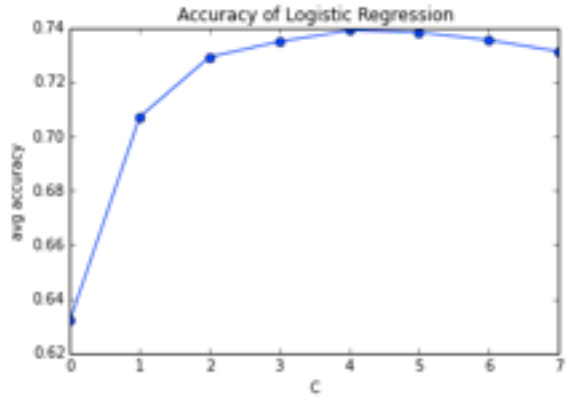
```
41323 unique terms in vocabulary
acc= 0.625624151106
 0.62562415110615821
```

2.cross-validation with different parameters
1)Accuracy of different C with same minimum document-frequency
When C=5.0, the model has the highest accuracy. When C=0.01, the model has the lowest accuracy. Also, no matter what value of C is, the distribution of model accuracy have the same tendency.

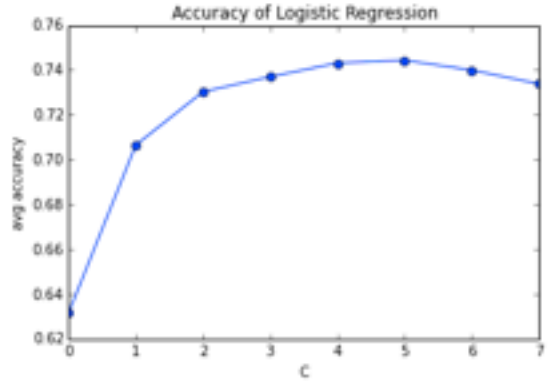
min_df: 2



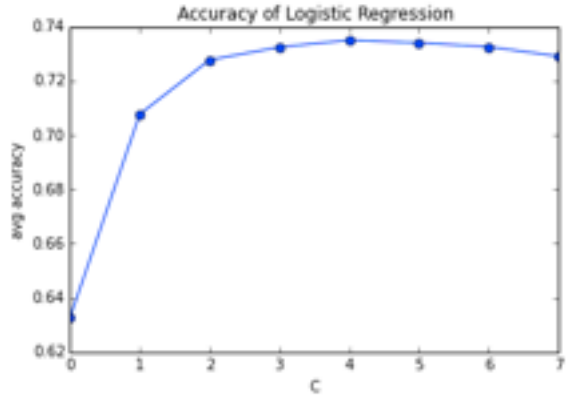
min_df: 6



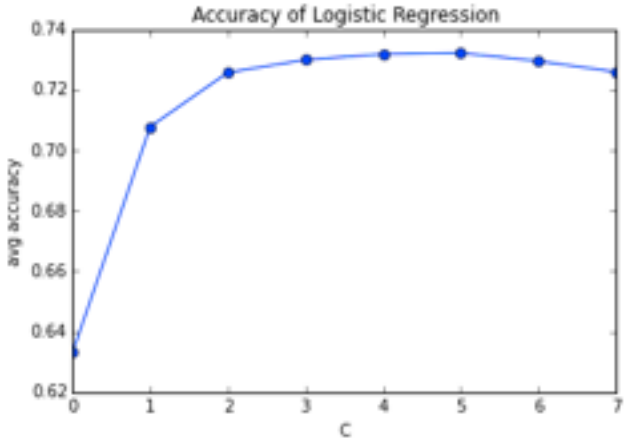
min_df: 4



min_df: 8

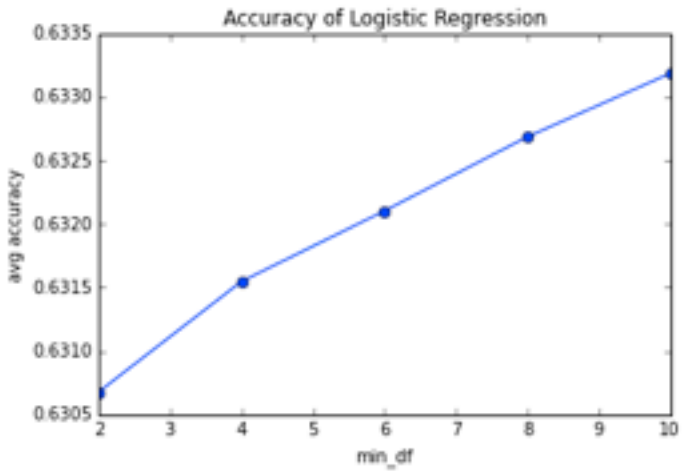


min_df: 10

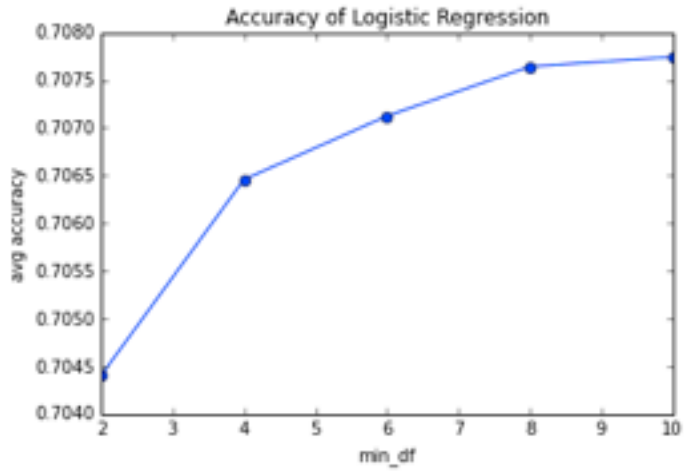


2) Accuracy of different minimum document-frequency with same C
We find that the tendency when C<0.5 and C>0.5 have opposite results.

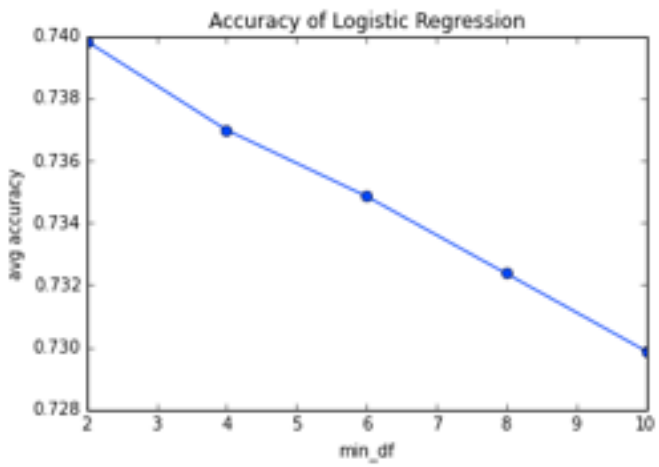
C= 0.01



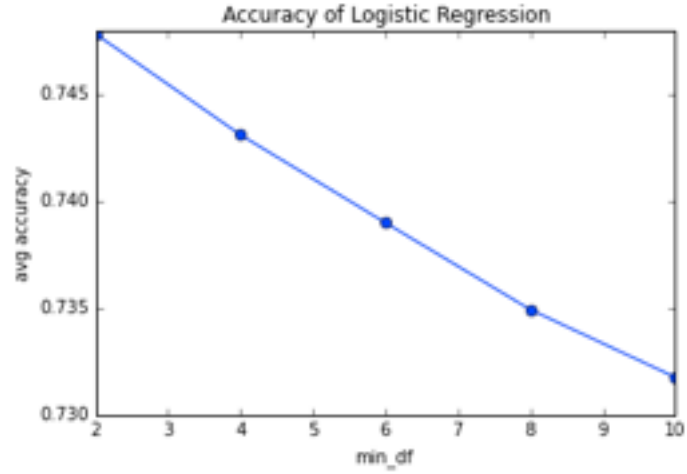
C= 0.1



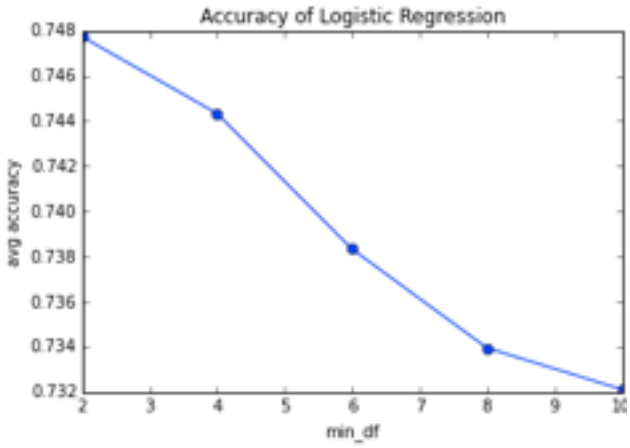
C= 1.0



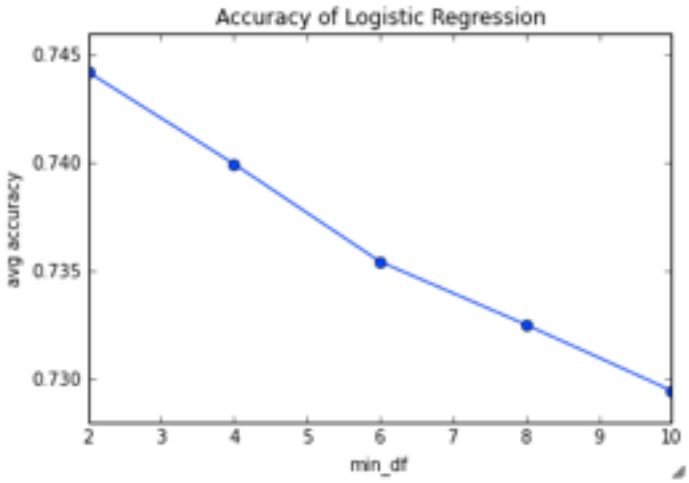
C= 5.0



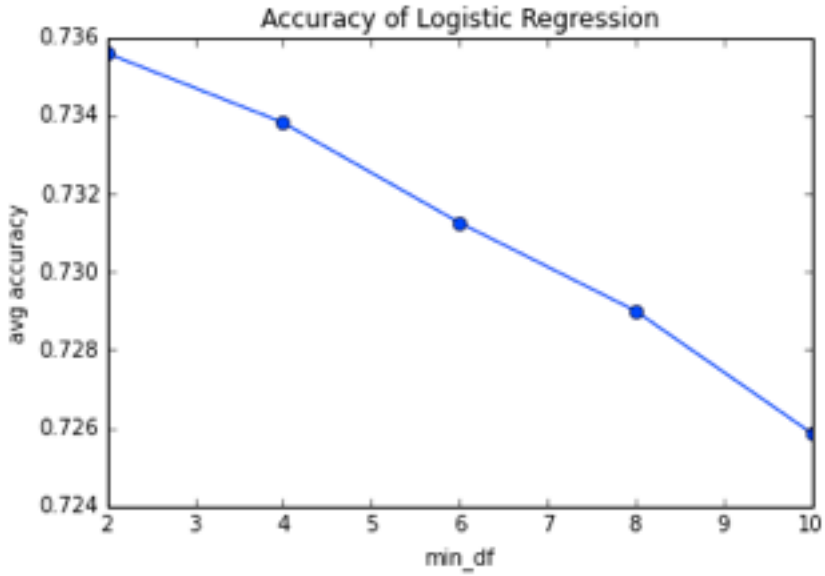
C= 10.0



C= 100.0

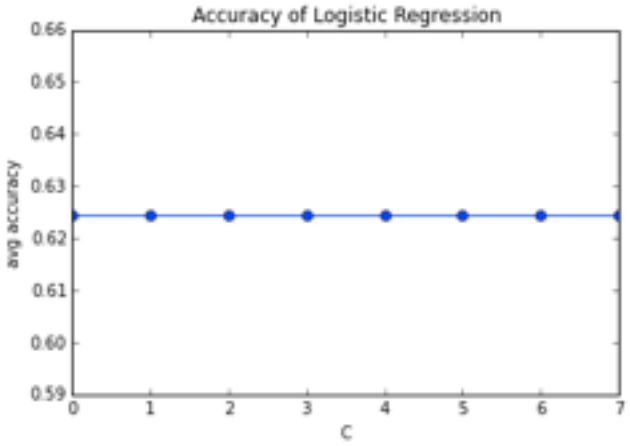


C= 1000.0

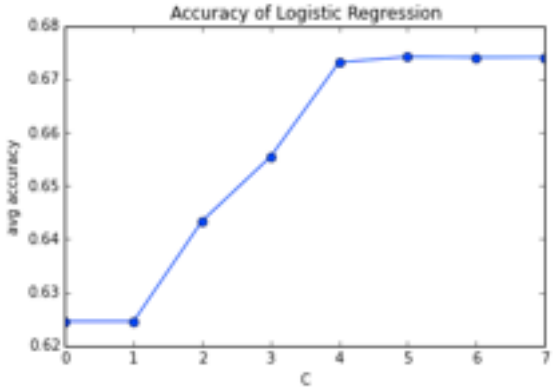


3)Accuracy of different C with same maximum document-frequency
The figure shows that when C=0.01, the model has the lowest accuracy.
The tendency for the tweets almost have the same tendency.

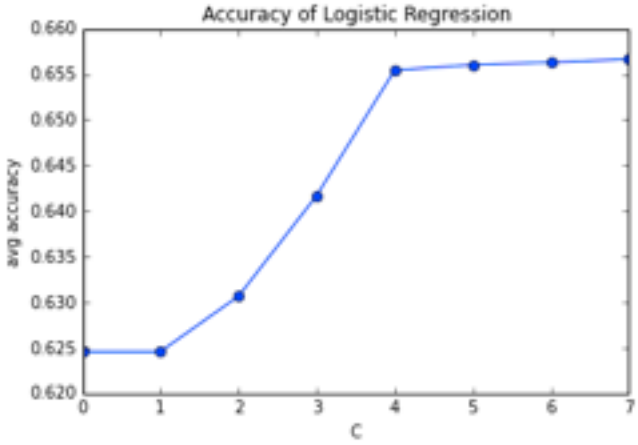
maxdf: 1e-05



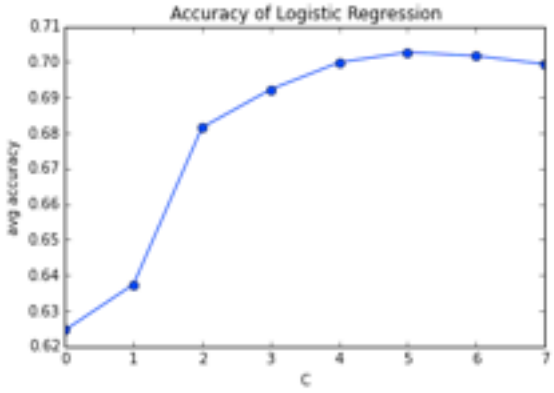
maxdf: 0.0001



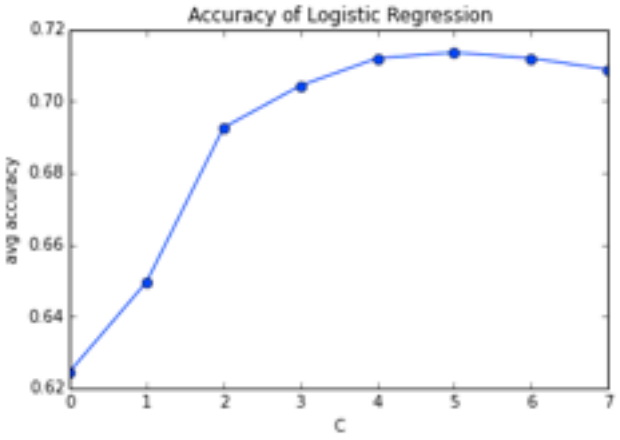
maxdf: 5e-05



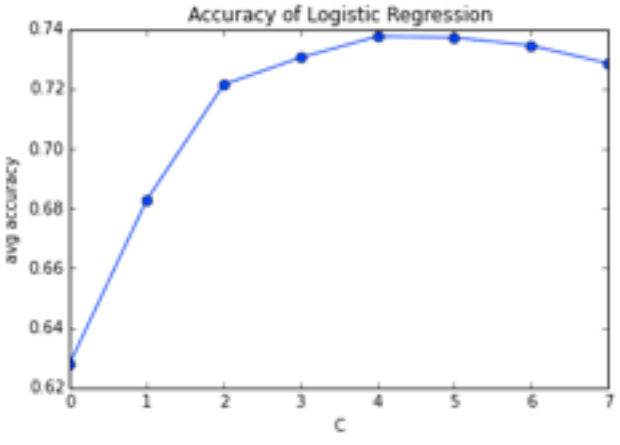
maxdf: 0.0005



maxdf: 0.001

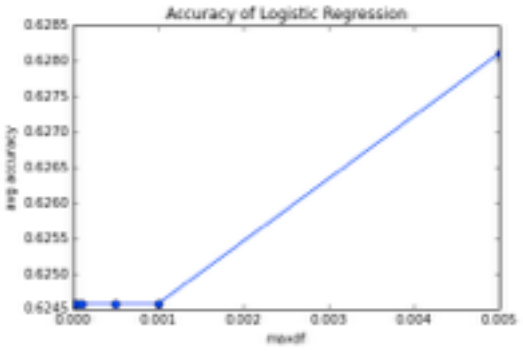


maxdf: 0.005

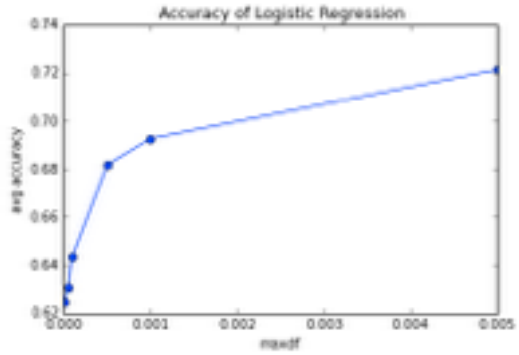


4)Accuracy of different maximum document-frequency with different C
The figure almost have the same tendency. When maximum document-frequency=0.005, the model has the highest accuracy. When maximum document-frequency=0.00001, the model has the lowest accuracy.

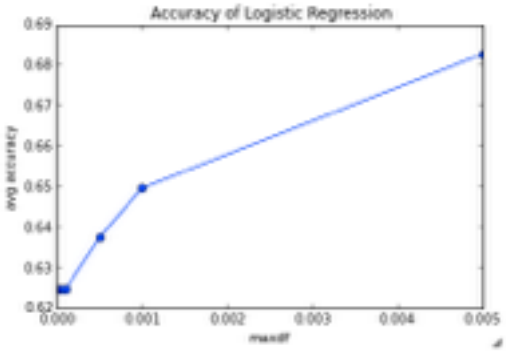
C= 0.01



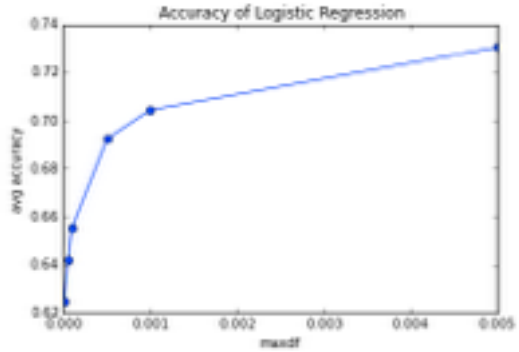
C= 0.5



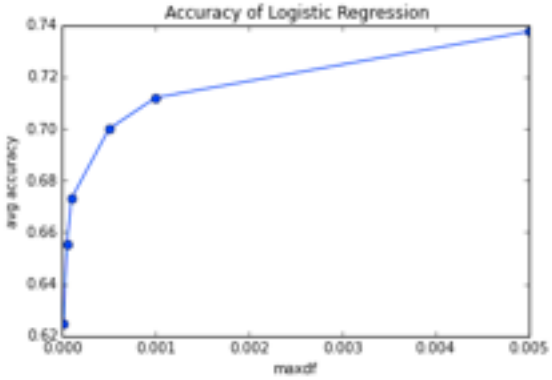
C= 0.1



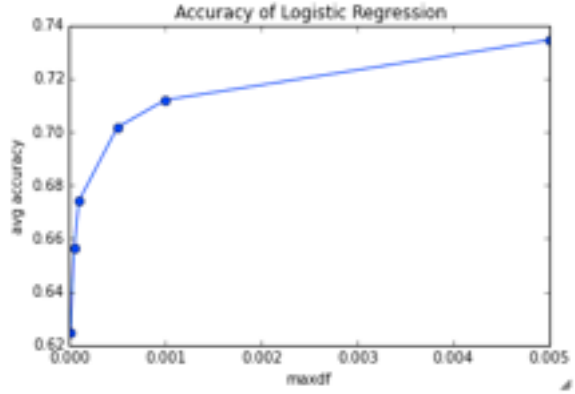
C= 1.0



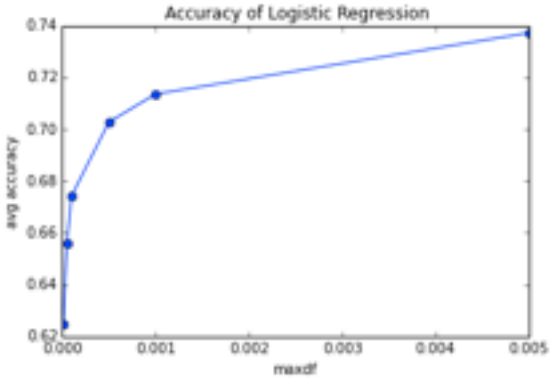
C= 5.0



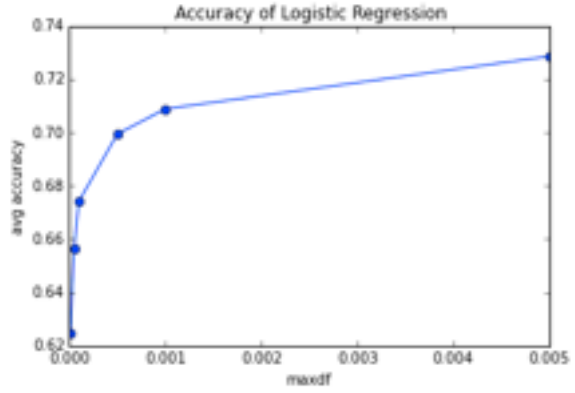
C= 100.0



C= 10.0

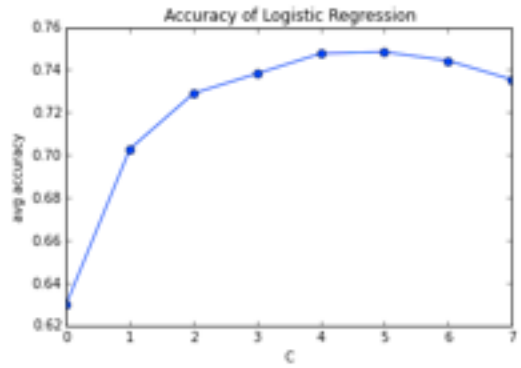


C= 1000.0

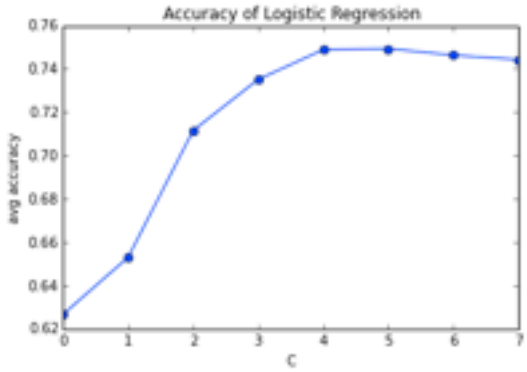


5)Accuracy of different C with same ngram
It doesn't have any tendency we can find. When C=100.0, the model has the highest accuracy. When C=0.01, the model has the lowest accuracy.

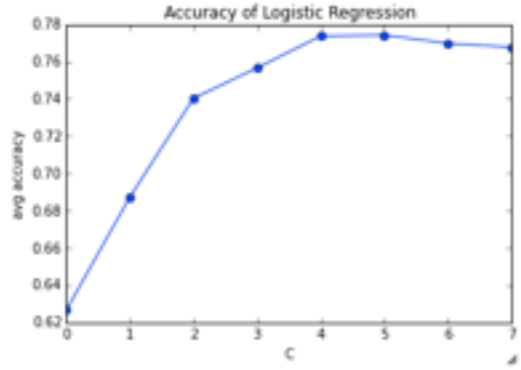
ngram: (1, 1)



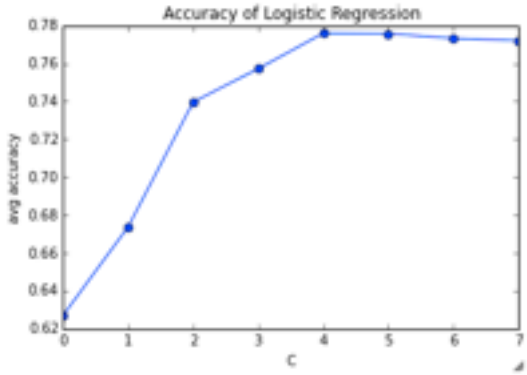
ngram: (2, 2)



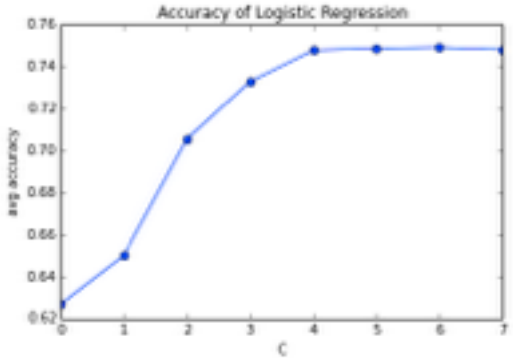
ngram: (1, 2)



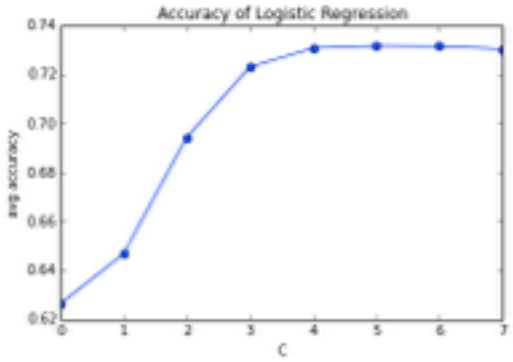
ngram: (1, 3)



ngram: (2, 3)

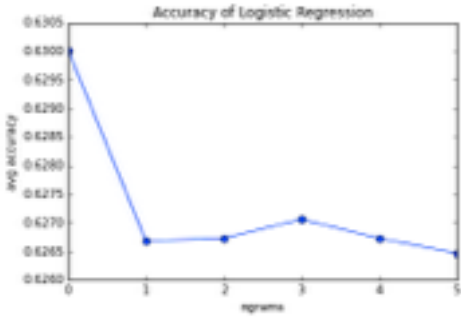


ngram: (3, 3)

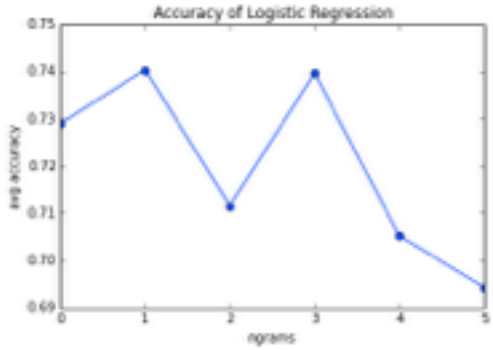


6)Accuracy of different ngram with same C
From the figure, we find that the tendency between C<0.5 and C>0.5 have the opposite results.

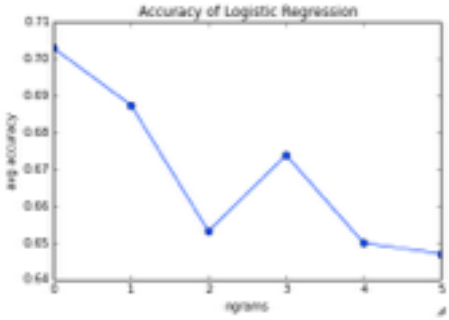
C= 0.01



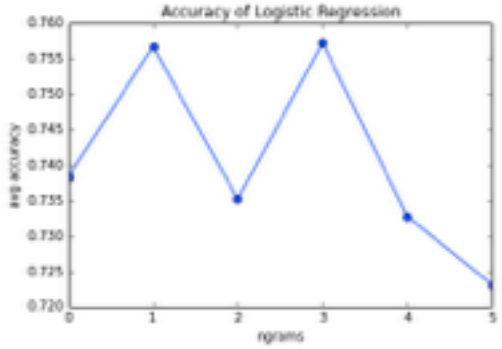
C= 0.5



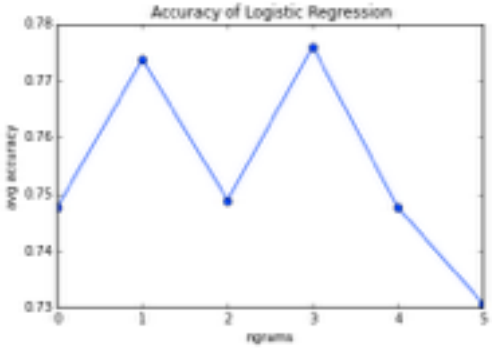
C= 0.1



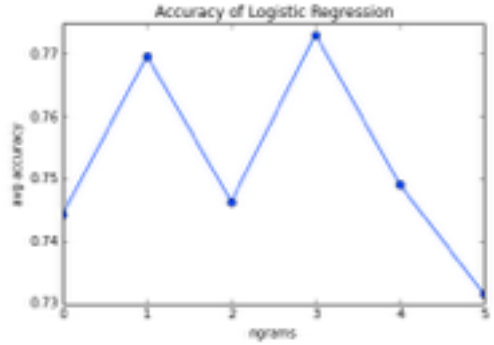
C= 1.0



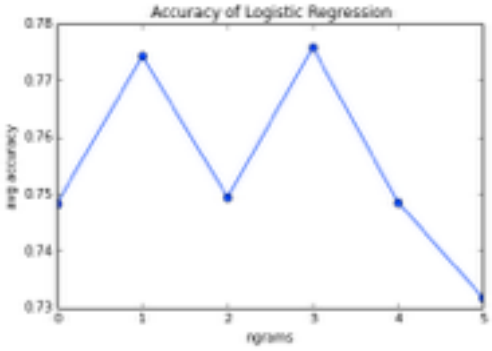
C= 5.0



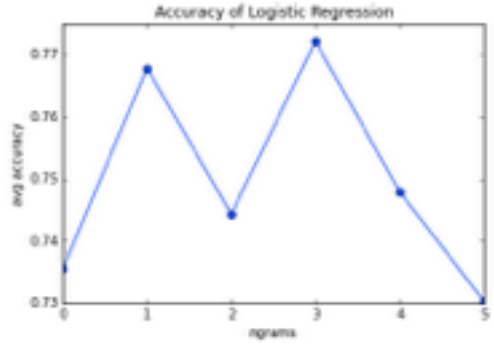
C= 100.0



C= 10.0



C= 1000.0



3.Conclusion

1)No matter using what parameter, we always can get a conclusion that when C=0.01, the model has the lowest accuracy.

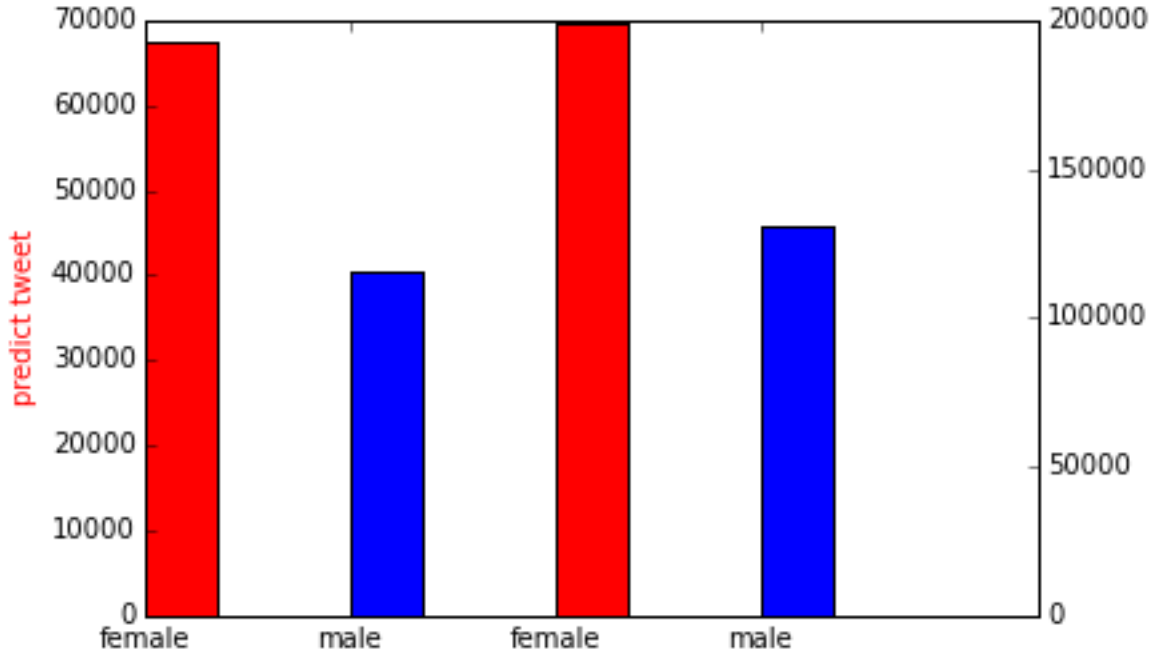
2)When the model using parameter maximum document-frequency, the accuracies doesn't have any relationship with value of C.

3)When the model using parameter minimum document-frequency and ngrams, the tendency of accuracies will change depends on C.

Combine methods

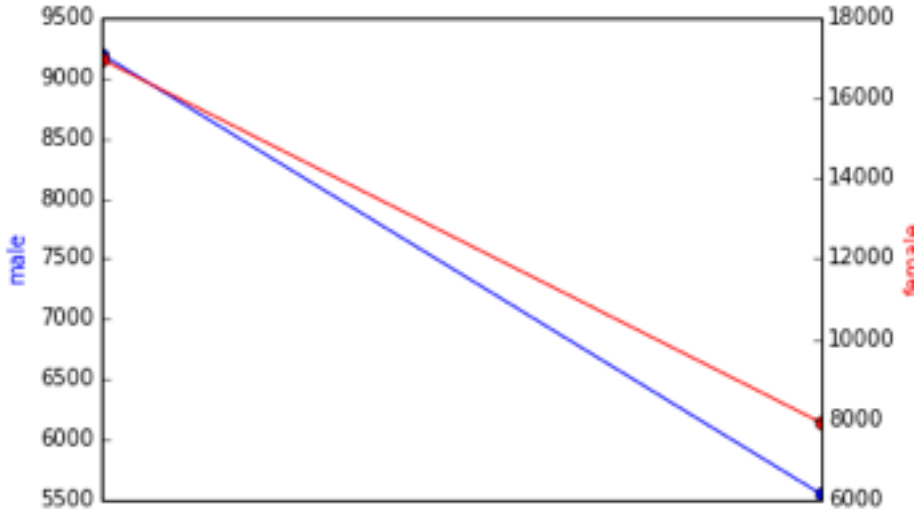
1.Distribution of different genders

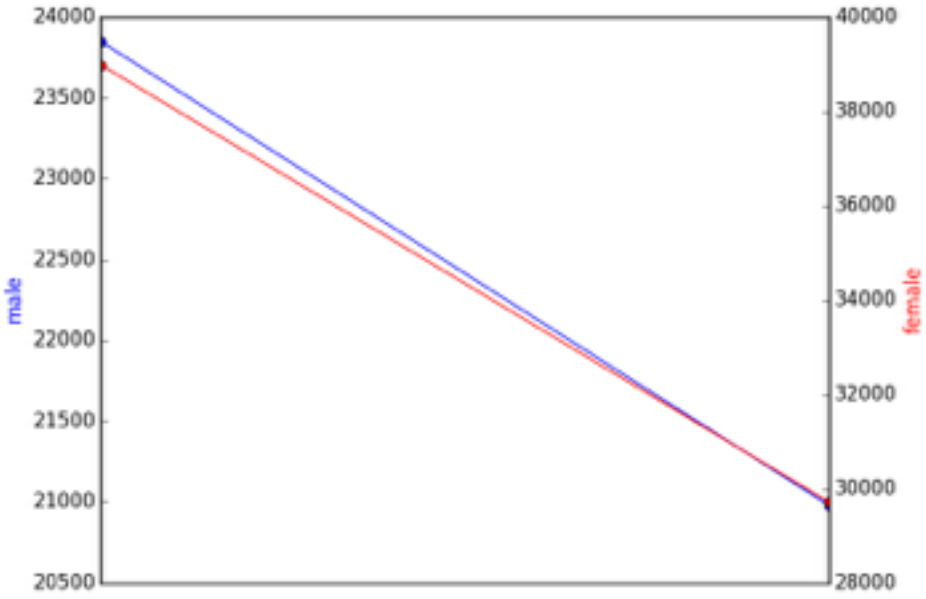
As we can see from the figure, analysis both test tweets and all tweets, we find that the tweets from female are more than the tweets from male. The tweets from female are about 1.5 times as the tweets from male.



2.Distribution of different sentiments with different genders.

Both female and male, they post positive tweets more than negative tweets.





Related work

- Machine Learning and Feature Based Approaches to Gender Classification of Facebook Statuses
- A Machine Learning Based Approach for Predicting Undisclosed Attributes in Social Networks

Conclusions and Future Work

1.Add a label as neutral in the training data.
2.Collect more training data and filter more precisely.
3.Compare accuracy between machine learning methods and census list and afinn sentiments.
4.Use more methods to compare the accuracy.