CIS434 Final Project - Pin Li

Tweet Sentimental Analysis

Introduction

In this project, we are assigned to identify whether certain tweets are complaint or not, given 1700 tweets of each. This task involves NLP and Machine Learning skills.

Process

To get the result, my main process includes pre-processing, modeling and prediction. For the pre-processing part, I add label to existing data and bind data onto one variable. For modeling part, I compare the accuracy of Bayes regression and SVM model and choose Bayes. Detailed codes are attached.

After predicting using the model chosen, the number of noncomplaint tweets is 1031. When I label each tweet, I apply criteria below:

1. If the tweet is "hate one like another", mark it as noncomplaint;
2. If only asking a question, mark it as noncomplaint;
3. if sarcasm, mark it as complaint.

Conclusion

After manually labeling the tweets, the noncomplaint tweet number decrease to 940. The main conclusions are as follows:

1. Accuracy = 940/1031 = 91.26%;
2. Human beings are complicated.

Since I label the tweets with high tolerant level criteria, the noncomplaint number and accuracy would be higher than another person labels the tweets. Also, human beings are complicated since they use positive words to show their complaints, as known as sarcasm. People also like to mention two companies to compare, making the prediction more ambiguous.

In [1]:

```
import numpy as np
import pandas as pd
from sklearn import svm
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn import metrics
from sklearn.metrics import classification_report,accuracy_score
```

In [2]:

```
#load data
neg = pd.read_csv('/Users/Pin/Desktop/SM Final Project/complaint1700.csv', header = 0, sep = ',')
pos = pd.read_csv('/Users/Pin/Desktop/SM Final Project/noncomplaint1700.csv', header = 0, sep = ',')
```

In [3]:

```
#add label
neg['label'] = -1
pos['label'] = 1
```

In [4]:

```
#bind data
negpos = neg.append(pos)
```

In [5]:

```
#set as vector
X = negpos['tweet']
y = negpos['label']
```

In [6]:

```
#countvectorize
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
```

In [7]:

```
X = cv.fit_transform(X)
```

In [8]:

```
#split test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.2,random_state=33)
```

In [9]:

```
#prepare the moder
from sklearn.naive_bayes import MultinomialNB
nb = MultinomialNB(alpha=0.5)
```

In [10]:

```
#train the bayes model
nb.fit(X_train,y_train)
```

Out[10]:

```
MultinomialNB(alpha=0.5, class_prior=None, fit_prior=True)
```

In [11]:

```
#predict on test set
predictions = nb.predict(X_test)
```

In [12]:

```
#see confusion matrix
from sklearn.metrics import confusion_matrix,classification_report
print(confusion_matrix(y_test,predictions))
print('\n')
print(classification_report(y_test,predictions))
```

```
[[261  71]
 [113 235]]


              precision    recall  f1-score   support

          -1       0.70      0.79      0.74       332
           1       0.77      0.68      0.72       348

    accuracy                           0.73       680
   macro avg       0.73      0.73      0.73       680
weighted avg       0.73      0.73      0.73       680
```

In [13]:

```
#load the real test data and train on the whole training set
test = pd.read_csv('/Users/Pin/Desktop/test_data.csv', header = 0, sep = ',')
tdata = test['tweet']
nb.fit(X,y)
```

Out[13]:

```
MultinomialNB(alpha=0.5, class_prior=None, fit_prior=True)
```

In [14]:

```
tdata = cv.transform(tdata)
```

In [15]:

```
y_pred = nb.predict(tdata)
```

In [16]:

```
test['label'] = y_pred
```

In [17]:

```
nonneg = test[test.label==1]
```

In [18]:

```
#see how many noncomplaint tweets
len(nonneg)
```

Out[18]:

```
1030
```

In [19]:

```
nonneg.to_csv('/Users/Pin/Desktop/SM Final Project/nonneg.csv')
```

In [24]:

```
#SVM
vectorizer = TfidfVectorizer(min_df=5, max_df=10,
                             sublinear_tf=True, use_idf=True)
train_features = vectorizer.fit_transform(negpos['tweet'])
test_features = vectorizer.transform(test['tweet'])
```

In [25]:

```
classifier_liblinear = svm.LinearSVC()
classifier_liblinear.fit(train_features,y)
prediction_liblinear = classifier_liblinear.predict(test_features)
```

In [26]:

```
test['svmlabel'] = prediction_liblinear
```

In [27]:

```
nonneg_svm = test[test.svmlabel==1]
len(nonneg_svm)
```

Out[27]:

```
1210
```

In [ ]:

```
nonneg_svm.to_csv('/Users/Pin/Desktop/SM Final Project/nonneg_svm.csv')
```

In [28]:

```
test.to_csv('/Users/Pin/Desktop/SM Final Project/testres.csv')
```