DS501

Case Study 3

Report

Group 10:
Congyang Wang, Jiexuan Sun, Zhaoning Su, Chao Xu

1. The Dataset Description

The dataset comes from the url http://www.cs.cornell.edu/people/pabo/movie-review-data. It contains 2000 written reviews of movie divided into positive and negative reviews with cross validation tag. And the original dataset is used in the experiments described in Pang/Lee ACL 2004.

2. The Businesses Intelligence and Motivation

We assume that we are a data science team working for a company providing streaming movies on demand online. Now, the company plans to select some good movies based on movie reviews. So, we need to distinguish those online movie review into negative and positive based on semantic analysis. We decide to use the Scikit-Learn package in python environment to analyze the reviews.

3. The Analysis

STEP1:

We first split the data into training set and test set. 75% (1500 observations) of the data will be the training set. The others will be used to test.

STEP2:

We use Pipeline function from Scikit-Learn to combine 'TfidfVectorizer' function and classification methods (LinearSVC and K-Neighbors Classifier).

a. The Scikit-Learn's 'TfidfVectorizer' function is used to calculate and build the TF-IDF matrix. The TF-IDF statistics, which is the short term of 'term frequency - inverse document frequency', is a widely used in information retrieval and text mining. It's a numerical statistics showing each word's importance to a document in a collection or corpus. Compared with traditional term frequency statistic, it adjusts the effect of those often-used words, which might appear frequently in general, by considering the IDF.

There are three parameters in the 'TfidfVectorizer' function:

min_df (float in range [0.0, 1.0] or int, default=1):
When using integer value, if the document frequency of including the target word smaller than this threshold, we ignore the word. When using float number, if the document propotion of including the target word smaller than this threshold, we ignore the word. It will filter some words which appear in a limited number of documents, when converting to TF-IDF features. See the following examples. We tried min_df = 2, 6, 10.

max_df (float in range [0.0, 1.0] or int, default=1.0):
When using integer value, if the document frequency of including the target word larger than this threshold, we ignore the word. When using float number, if the document frequency of

including the target word larger than this threshold, we ignore the word. It will filter some words which appears in most of the documents, when converting to TF-IDF features. See the following examples. We tried max_df = 0.85, 0.9, 0.95.

ngram_range (tuple (min_n, max_n)):
It limits the boundary of the range of n-values for different n-grams to be extracted. (min_n < extracting n < max_n) See the following examples. We tried (1, 1) and (1, 2).

b. The two classifiers provided by Scikit-Learn we used are LinearSVC and K-Neighbors Classifier.

LinearSVC means the linear support vector classification. And there is a parameter named 'C', which is a penalty parameter of the error term. We tried C = 0.1, 1, 100, 1000.

K-Neighbors Classifier means k-nearest neighbors classification. And there is a parameter named 'n_neighbors', which means the number of neighbors to use. We tried n_neighbors = 3, 5, 7, 9.

STEP3:

We use the GridSearchCV function from Scikit-Learn to find the best combination of parameters for each classification method.
Number of combination of parameters for LinearSVC: 3 * 3 * 2 * 4 = 72
Number of combination of parameters for KNeighbors: 3 * 3 * 2 *4 = 72

STEP4:

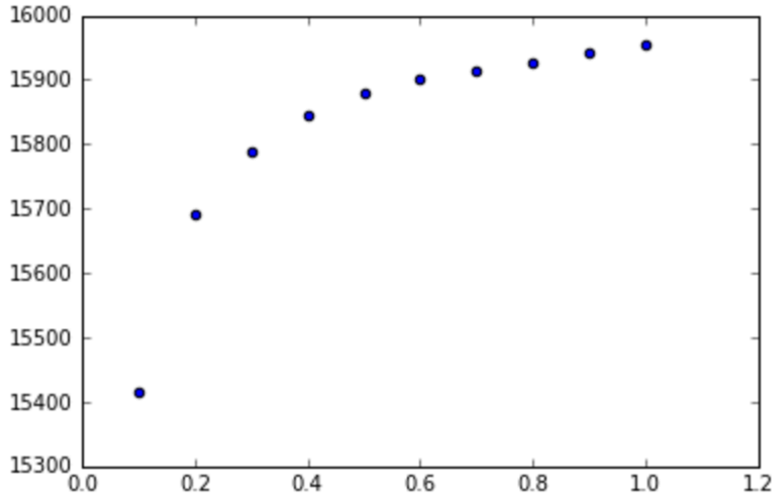
We use the test dataset to test the two classification model with selected parameters. We draw the confusion matrix and compare them. Then, we select the one with more accuracy rate as our final model.

4. The Findings

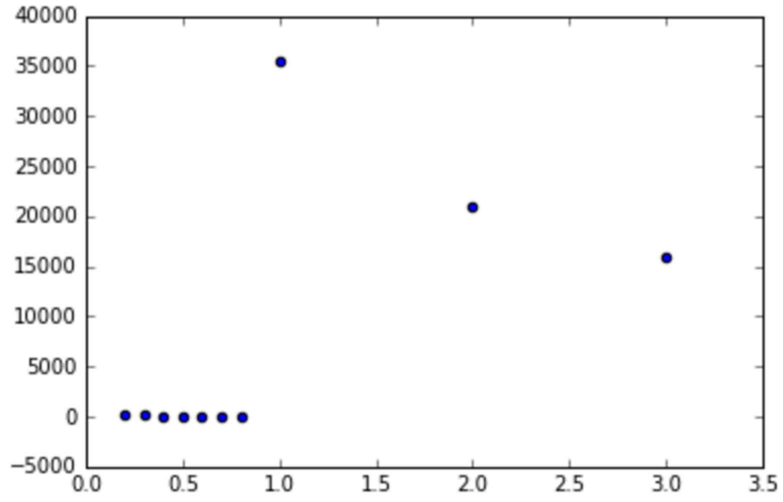
4.1 Different Parameters:

a. max_df & min_df:

```
The number of features under default(max_df=1.0, min_df=1)
(1500, 35429)
The number of features under max_df=0.95, min_df=3
(1500, 15944)
change the max_df we will get different countings of the target words:
```



```
change the min_df we will get different countings of the target words:
```



b. ngram_range:

```
The number of features under default(ngram_range(1,1))
(1500, 35429)
The number of features under ngram_range(1,2)
(1500, 436333)
The number of features under ngram_range(2,2)
(1500, 400904)
```

4.2 Different Classifier:

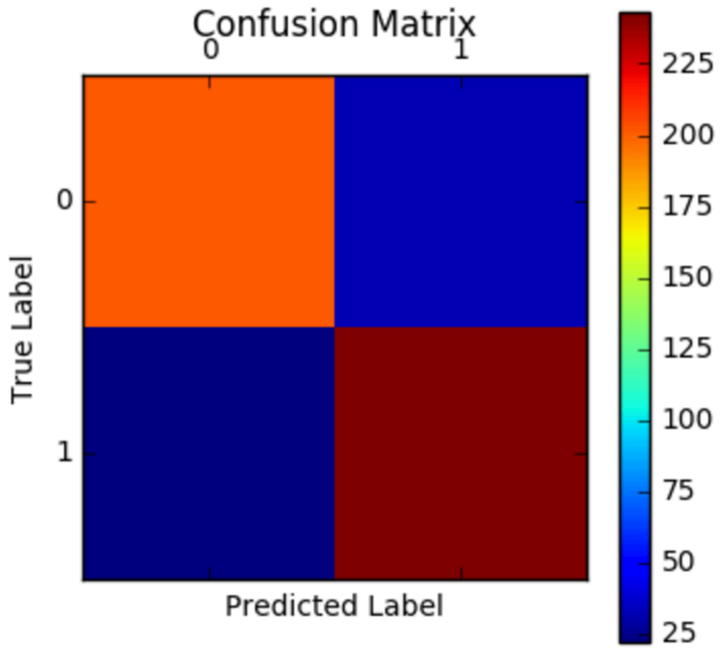
a. LinearSVC:

Best parameter set:

'vect__max_df': 0.85,
'vect__min_df': 6,
'vect__ngram_range': (1, 2),
'clf__C': 1

Confusion Matrix:

```
[[202  33]
 [ 22 243]]
```



Accuracy Rate: 0.89

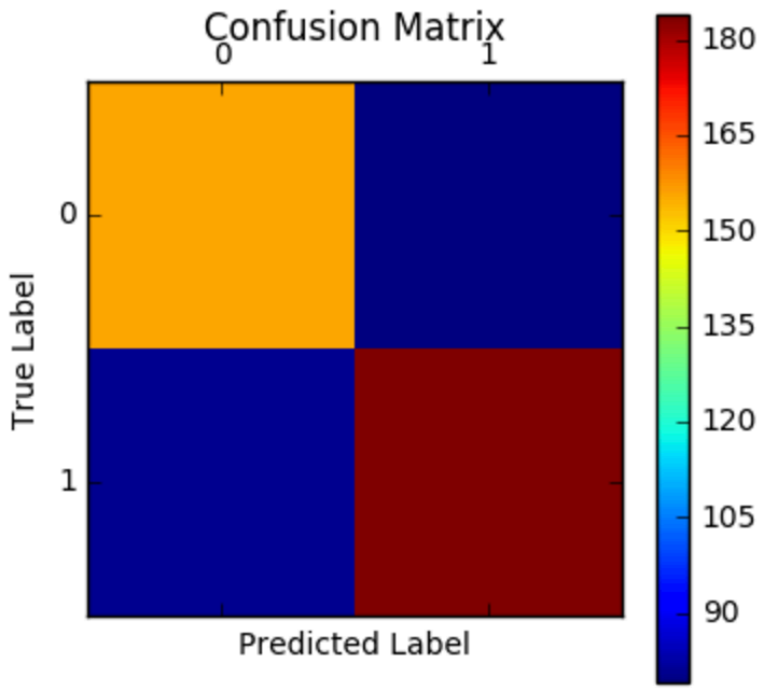
b. KNeighborsClassifier:

Best parameter set:

'vect__max_df': 0.85,
'vect__min_df': 6,
'vect__ngram_range': (1, 1),
'knn__n_neighbors': 7

Confusion Matrix:

```
[[156  79]
 [ 81 184]]
```



Accuracy Rate: 0.68

c. Conclusion:

Based on the above confusion matrix, we choose LinearSVC as our classification method with its best parameter set. Obviously, LinearSVC has higher accuracy rate. We choose LinearSVC as our final model.

5. The Surprise

Under the LinearSVC with its best parameter set {'vect_max_df': 0.85, 'vect_ngram_range': (1, 2), 'clf_C': 1, 'vect_min_df': 10}, we take 2 examples where the prediction was incorrect.

5.1 The first incorrect review(cv978_20929):

We incorrectly classify this review into negative review. Actually, it should be positive. There're two main reasons we think to explain the error: 1. There're too many negative words in this review since it is a horrible movie. Even though the writer want to express an appreciate on this horrible movie, it still need too much negative words to describe the plot like "kill", "crime", "jail" and so on. Because of these negative words, we may get the wrong prediction. 2.There are few positive words used in this review, only one "good" and one "great", the positive words’ frequency is too small.

5.2 The second incorrect review(cv296_12251):

We incorrectly classify this review into negative review. Actually, it should be positive. The main reason is that he uses a few possible words. So that even the author doesn't use many negative words, it still can't be predicted as a possible one easily. Also the words "tear", "just" promote the review predicted as a negative review.

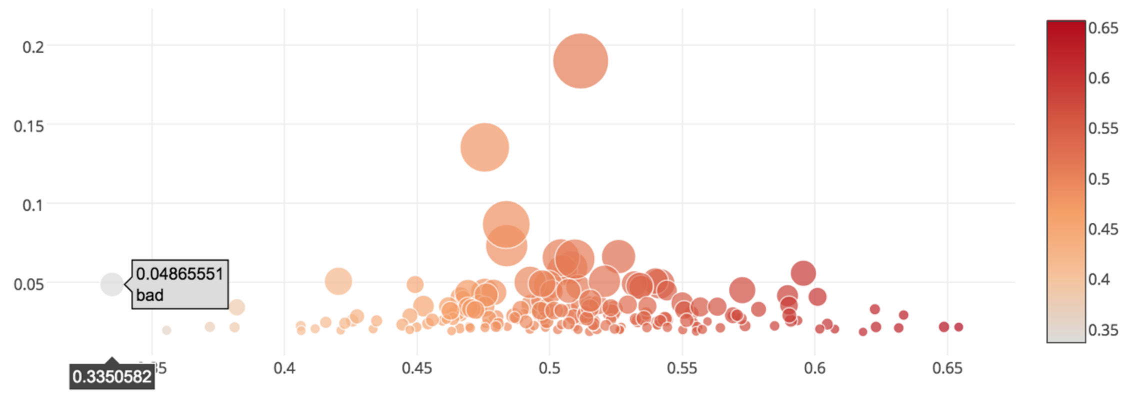
6. The Visualization

6.1 The distribution of Features

Annotation:

- Bubble: Features(Words).
- Bubble Size: The number of reviews contains the feature.
- Bubble Color: The proportion of positive reviews contains the feature.
- X-axis: The proportion of positive reviews contains the feature.
- Y-axis: The average TF-IDF of the feature.

Bubbles, which are on the left with light color, represent features appearing mostly in the negative review. Bubbles, which are on the right with dark red color, represent features appearing mostly in the positive review. What's more, the bubble will be larger, if more review contains the feature. So, large bubbles(features) on the left/right side are the keys to distinguish negative and positive reviews.



6.2 Reviews Clustering (K-Means) under PCA

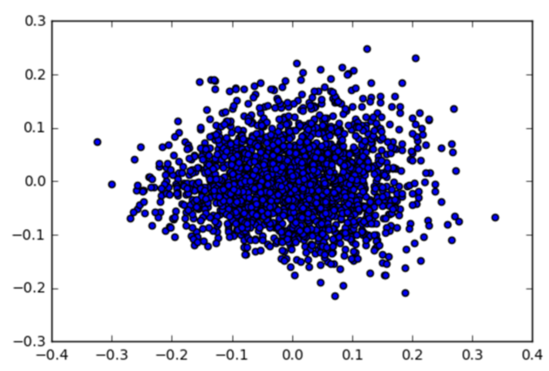
Annotation:

- Dot: Review.
- X-axis: Principal Component.
- Y-axis: Principal Component.
- Color: Cluster.

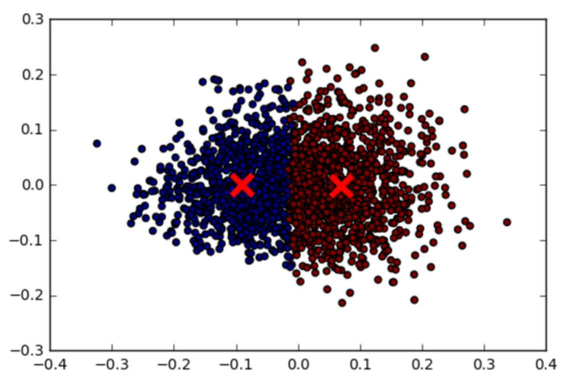
a.

We use PCA to 'derive' the component. Then, we use clustering method - 'K-Means' to cluster each reviews into two groups shown in different color as below. We try to do the PCA to reduce the dimension of the data set first. In this case we use the CountVectorizer() and TfidfTransformer() as two variable who are used as standard to clustering the data. The CountVectorizer() function will get a frequency matrix of the reviews. The TfidfTransformer() function will calculate the tf-idf value of each words. By PCA we can get a 2-dimension data set which used to scatterplot a 2D figure. Then we first perform the PCA's scatterplot, and then use the kmeans method, which is a kind of unsupervised learning method, to do the clustering. And we get the clustering result as the 2nd figure shows. Also we can list that which reviews are labeled as cluster 0 and which are labeled cluster 1. After all of that, we will use the actually table "pos" and "neg" to find the accuracy rate of our prediction.

PCA Figures:

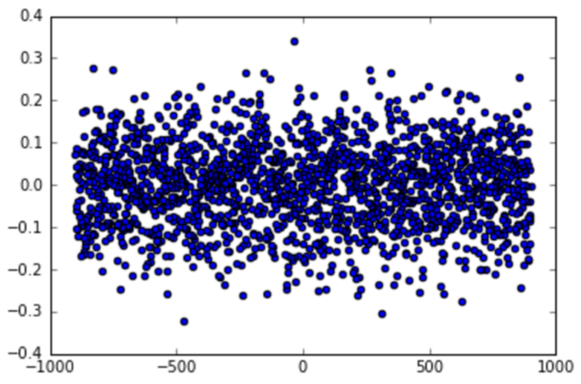


Cluster Figures with Kmeans:

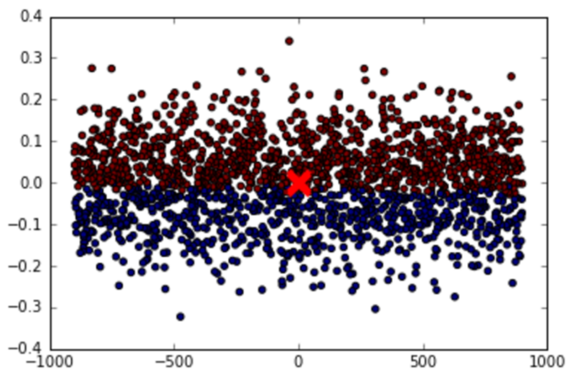


b.
In the first time we use the CountVectorizer() and TfidfTransformer() as two variable who are used as standard to cluster the data. However both of the frequency matrix and tf-idf value have the same weight. Now we assume that the possitive words will be more concentrated also the negative ones. So we add another variable, which reviews are these words come from, to cluster the data. The process is just like the 1st one. We compare the accuracy rate in this case with the first one.

```
pca figures:
```



```
cluster figures with kmeans:
```



7. Businesses Decision:

Finally, we find that LinearSVC with best parameter could distinguish between negative and positive reviews correctly. Also, unsupervised method, like using K-means after PCA analysis, could also distinguish reviews into different groups. After distinguishing those reviews, we would suggest our company to buy (select) those movies with more positive reviews.