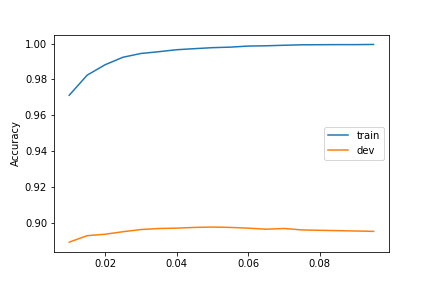
Homework 2

1. Preliminary

Kaggle Username: daniel

Best default accuracy: 88.256% (got from logistic regression with default setting)

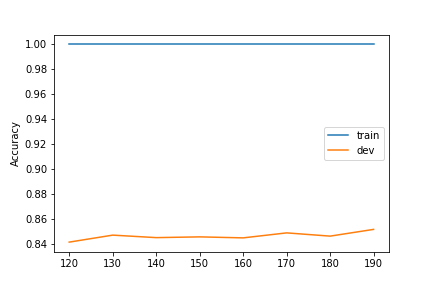
Best custom model accuracy: 89.464% (got from logistic regression with custom features)

1. Default Features
   1. Logistic Regression

For logistic regression, the best accuracy I got from Kaggle is 88.256% and the range I picked for C is (0.01, 0.1, 0.005)

C is the inverse of regularization strength, the larger C means smaller regularization strength which leads to small penalty value and probably overfitting problem.

Firstly, I set the range as (0.1, 1, 0.1) but I found that the trend of the entire dev accuracy line was decease but trend of training accuracy was increase. So, it supposed to the overfitting problem and I reset the range to (0.01, 0.3, 0.01). After that I found there was a peak when C was equal to around 0.05. Therefore, I made the range smaller and closer to 0.05 and found the parameter C which gives the highest test accuracy.

* 1. Random Forest

For Random Forest, the best accuracy I got from Kaggle is 85.235% and the range I picked for n\_estimators is (80, 140, 5)

Parameter n\_estimators decides how many trees will be used in the random forest model and it help to make the prediction more accurate by combining the prediction of many trees.

Firstly, I set range to (50, 300, 10) and I found that the value of dev accuracy was stable with only little fluctuate from n\_estimators =70 to n\_estimators = 140. It means that if I continue increase number of estimators, the accuracy would not increase but the calculation load would be significantly increased. So, I narrowed the range to (80, 140, 5) and found there is a peak when n\_estimators is equal to 110.

1. Custom Features

After adding my own features to the feature function, creating new vector for each review I tuned Parameter C and n\_estimator several times according to the result and finally got the best classifiers for logistic regression model and random forest model.

After submitting to Kaggle, I got that the best accuracy for logistic regression model is 89.464% and 86.06% for random forest model.

As for vector, firstly, I used CountVectorizer(ngram\_range=(1,2)) to build the vector since I believe the sequence of some words is important because there exist some meaningful phrases like “great movie”, “very good” which is highly correlated with positive rating. But if those phrases were split into words, they should be “great”, “movie”, “very” and “good” and they are not that strongly correlated with positive rating.

Regarding to features, I firstly defined a function called neg\_count which count the number of negative words of each review according to sentiment lexicon since the number of positive words and negative words are highly related to the sentiment of the review.

Additionally, I found that some audience gave their rating for the movie in their review. So, I created features like “rate\_1”, “rate\_10” to record whether they gave their rating and what it is.

Also, since some phrases are important like “a good”, I created some bigram features like “good movie” to record whether those phrases appear.

Some marks are also correlated with people’s sentiment. I created features like “!>=3” to record that if there were more than 3 exclamation point appear.

I also created features about those words like “LOL” which show people’s feeling under such an unofficial occasion. For words like “great”, it shows that people are pretty happy if it appears many times. Then I created “great>=3” to record whether people used it many times in their reviews.

1. Analysis

After tuning parameter and creating features, I submitted my prediction to Kaggle and got that the best classifier with default features is logistic regression and logistic regression is also the best classifier with custom feature.

I used “show\_weights” and generated the global importance weights for both classifiers. By comparing them, we could found that they are very similar since they are based on the same mathematical algorithm which is logistic regression.

For example, the first two words which are most importantly correlated with positive review are “excellent” and “perfect” for both these 2 classifiers. And the first two words which are most importantly correlated with negative review is “worst” for both these 2 classifiers.

However, there are also some differences between these 2 classifiers according to the result of global importance weights. The first one is that global weight table of classifier with custom features contain some 2-word phrases like “the worst”, “not worth” which is also helpful for determining the sentiment of review.

Additionally, some important features of classifier with custom features are created by myself like “rate\_3” which recorded audiences’ rating for the movie. Another difference is that the same word in different classifier have different weights. For example, the weight of “worst” in classifier with default feature is -1.15 while it is -0.736 for classifier with custom feature. It is because I add more feature to classifier with custom feature and tuned the regularization strength.

By comparing the and looking into those errors, I found there are some differences between these 2 classifiers. The first one is that classifier with custom feature is much better than default feature in determining the sentiment of reviews which contain many negative adverbs like “can’t”, “don’t”, “wasn’t”. For example, with classifier with default features, “wasn’t disappointed” will be split into two words and both of them will be determined to be negative words. However, with classifier with custom feature, they could be determined to be related with positive sentiment since Bigram was used in this classifier.

Another advantage is that some features I added are pretty useful in determining sentiment like “rate\_10”, “rate\_1” which captures audiences’’ real feeling for this movie. Sometimes they talked a lot about the plot and content of the play which may contain lots of words which are sentiment words in other reviews but means nothing in this review since it is about the plot. In this case, it could be very useful if we could capture people’s rating for the movie.

In the future, I’d like to do more error analysis part since I found that there are still some grammar that may lead to false sentiment determination. so, I may put those features into vector to tune and optimize it.

1. Statement of Collaboration

I discussed with Gloria Gong, her error analysis program cannot find the right path to import data and package like ml\_sentiment. We discussed how to solve the problem and searched on the Internet. We also discussed the reason some problems we met like when we put some features into the vector, the accuracy would do down rather than increase. I also talked with Lingyun Guan. We talked some ideas about how increase the accuracy.