# **Problem Setup**

In this report, I will use a dataset of 10662 comments on movies. Half of the comments are negative, and the other half are positive. Each sample of the dataset contains one or a few sentences.

I will use the dataset to train different classification models to predict that one comment is positive or negative. The algorithms applied include logistic regression, Naïve Bayes and support vector machine. I will also use different NLP tools to process the data, including vectorization, stop words, lemmatization and stemming. The objectives are comparing different models, NLP tools and parameters to find which one perform better in accuracy and generalization capability.

# **Experimental Procedure**

**Step 1 Different Parameters and Models**

Firstly, I will use all the features extracted from the dataset and fit them into different models including logistic regression, Naïve Bayes and SVM. Confusion matrix and ROC will be applied to measure the performances.

Then I will change one of the three key parameters once at a time, which are max features, min df and max df. Vectorized data will be fitted repeatedly into the same three models to measure the influence on the prediction performance.

**Step 2 Different NLP Tools and Models**

From three NLP tools which are stop words, lemmatization and stemming, I will choose one or different combinations once at a time to process the data. The processed data will be fitted into the same three models in step 1 and the same measurement methodology in step 1 will be used to observe the influence of different NLP tools on the prediction performance.

# **Classifier Analysis**

**Step 1**

Firstly, the max feature parameter is set ‘None’. The min df and max df parameters are both 1.

The results including confusion matrix, precision, recall, accuracy and AUC values are shown in appendix. The logistic regression model performs the best and the Naïve Bayes model performs the worst both in accuracy and AUC values. However, I use all the vectorized features to fit the models, so there may be overfitting problem especially in logistic regression model. Another observation is that the ROC curves of the logistic regression model and the SVM model are smooth and close to the shape of a quarter. The ROC curve of the Naïve Bayes model has two sharp ends which shows linear relationship in TPR and FPR, which indicates the algorithm may have drawbacks in calculating extreme values.

Then I experiment with different values in three parameters. By observation in appendix, I find that with the max feature value, max df value decreasing and the min df value increasing, the performance of the logistic regression model gradually decreasing, as well as the support vector machine model, except for the drastic decreasing when max feature value drops from 1500 to 1000. At the same time, the overall performance of the Naïve Bayes model gradually becomes better except for the period that the max feature value drops from 2500 to 1000. When the max df value drops from 1.0 to 0.8, the performances of three models stay the same, which means there may not be any word whose frequency exceeds 80%. Also, I consider that all three models perform the best in this two stages of different parameter values.

**Step 2**

I use one of the best performance parameter combinations to perform the second step of experimentation. The original performance of three models are shown in appendix. From here I started experimenting with different NLP tools.

By applying three different NLP tools respectively including stop words, lemmatization and stemming, the performances of three models are fluctuant and close to the original performance, or even worse. Only when applying stemming, the performances of the logistic regression and support vector machine models become noticeably better. However, when combining two or three tools together, the overall performances of those two models get worse than when applying one of the combined tools, or even all of them. Only the Naïve Bayes model performs stably.

# **Result and Conclusion**

By comparing the performances of three models in different set of vectorization parameters and NLP tools, I find some interesting conclusions and consumptions.

1. The Naïve Bayes model works quite differently with the other two models. The performance of predictions goes the opposite with them in some conditions. It may have strength in predicting on some kinds of data.
2. The parameters of vectorization have great influence in model prediction. It’s important to experiment and choose proper values, which is not too big or too small, to best fit the models.
3. The overuse of NLP technology to process the words may cause worse performance in prediction. It may because of the overfitting problem, or these tools are not reasonable enough yet. One interesting consumption when I find the stemming work, but the stop words don’t is that when people say something positive or negative, they may specifically use some words which we may think are stop words but can be similar to the stems of positive or negative words. So, we should reconsider the categorization of stop words and the application of stop words. Because people may use specific set of stop words in some conditions.

# **Appendix**

A screenshot of a cell phone

Description automatically generated

Classifier Analysis Step 1.1

A screenshot of a computer

Description automatically generated

Classifier Analysis Step 1.2

A close up of a map

Description automatically generatedA close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Classifier Analysis Step 1.3 – ROC of LR, NB and SVM (from left to right)

A screenshot of a computer

Description automatically generated

Classifier Analysis Step 1.3

A screenshot of a cell phone

Description automatically generated

Classifier Analysis Step 2.1

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Classifier Analysis Step 2.2

A screenshot of a computer

Description automatically generated

Classifier Analysis Step 2.3