COMP300027 Project 2 - Report

Rating Prediction from Reviews

I. Introduction

The aim of this project is to build a supervised learning system that gives the best prediction accuracy on the ratings from reviews given by reviewers to a business/restaurant, which is represented as one, three, or five stars. This problem is a subset of **Sentiment Analysis**, which is a study on how to classify and interpret the polarity of existing subjective information (e.g. text, document). This has many real-world applications ranging from commercial use up to science and social researches. However, today, there is no best way to represent a text. In this report, we will experiment with different methods of representing text reviews, filtering/selection of features, and different prediction model to achieve the highest prediction accuracy.

II. Experiment Design

a. Vectorization Method

Vectorization is an encoding method to convert text into analyzable features (e.g. numeric). In this experiment, we will experiment with three methods.

<u>Frequency Vectorization</u>

Encodes each text into "Bag of Words" representation. Meaning that each feature will represent the frequency of occurrences of a word per text.

<u>TF-IDF Vectorization</u>

Encodes each text inversely to the frequency approach. It represents how rare the word occurrences which can be interpreted as the importance of the word to the review.

<u>Doc2Vec Representation</u>

Higher level of encoding takes account the structure of each document and generalizes it to numerical representation. Therefore, features will be more correlated to each other compared to the other two methods stated.

b. Feature Selection

As we generate new features, filtering is essential to save training time and prevent overfitting. Feature selection ensures that the feature fitted are the most correlated to class. We experiment with two feature selection methods, both taking the top best 50, 100, and 200 features.

<u>Chi-Squared Filtering</u>

This filtering method is based on Chi-squared residual analysis. It measures the normalized squared difference of observed and expected value pair under independence assumption.

<u>Mutual Information Filtering</u>

This filtering method is based on entropy of the value given the classification.

c. Model Classifier

Given the multi-class nature of this experiment, the chosen classifier model must have a multi-dimensional nature or, if possible, be extended to multi-dimensional. We must also take account the impact of the experiment design, specifically the impact of using different engineered features. Time and space complexity will also one of the driving factors.

<u>Gaussian Naïve Bayes (GNB)</u>

Gaussian Naïve Bayes is chosen to be as our base classifier due to the robust nature of the assumption of independence and Gaussian normal distribution. It is a good benchmark, especially with frequency based vectorizer.

<u>Multinomial Naïve Bayes (MNB)</u>

Multinomial Naïve Bayes is chosen as it is expected to out-perform Gaussian assumption on frequency-based vectorizer. However, does not work with Doc2vec vectorization which will be explained in the report.

<u>One vs Rest Linear Support Vector Machine (SVM)</u>

This classifier works by maximising the margin which is assumed to be linear that separates the hyper plane. As we train our model with supervised encoded features, support vector machine is expected to fit well with this problem.

Multi-Class Logistic Regression (Logistic)

Uses logistic function to estimate the natural probability of the features. This experiment extends the regression into multinomial to allow multiclass prediction.

III. Data Preparation

Two raw data files were provided where one contains the meta features (meta) with the rating class and another that contains the text review of each review. Doc2vec text representation in a data frame structure and frequency vectorized text in a sparse matrix structure is also provided. Rating class from metadata is removed for supervised learning. Then, from these raw data files, we generate new data features to be learned by the predictive classifiers.

Frequency Vectorized Data

Frequency vectorized text is filtered to best 50, 100, and 200 features with both mutual information and chi-squared method.

TF-IDF Vectorized Data

First, transform the frequency vectorized sparse matrix into using TF-IDF transformer. Then filtered the same way.

Doc2vec Data

Doc2vec has top 50, 100, 200 features representation provided. Nothing is changed from this dataset.

Doc2vec + Meta Data

Meta features are pre-processed by removing date in assumption that date does not affect ratings given, and review_id, reviewer_id, business_id in assumption that there is no bias from a reviewer to a specific business. Then, combined with Doc2vec data with expectation of improving accuracy performance.

All engineered data will then be trained with all the predictive classifiers. With exception of Doc2vec data that will not be trained in MNB classifier as MNB can’t process negative values. Prediction accuracy will be from the result of cross-validation of 5 folds.

IV. Error Analysis

The subjective nature of reviews is prone to human emotions, in other words, noise. Noise can heavily affect count based vectorizer methods stated above. Take an example of the word “Goooood” and “Good”. Frequency vectorization will prioritize “good” as it appears more often and TF-IDF will also prioritize the word “Goooood” as it appears less even though both imply the same meaning. The words “and”, ”a” and “the” will also be prioritized in frequency vectorization even though it has no correlation to the class. While human typos will be prioritized by TF-IDF.

V. Result and Discussion

a. Frequency vs TF-IDF

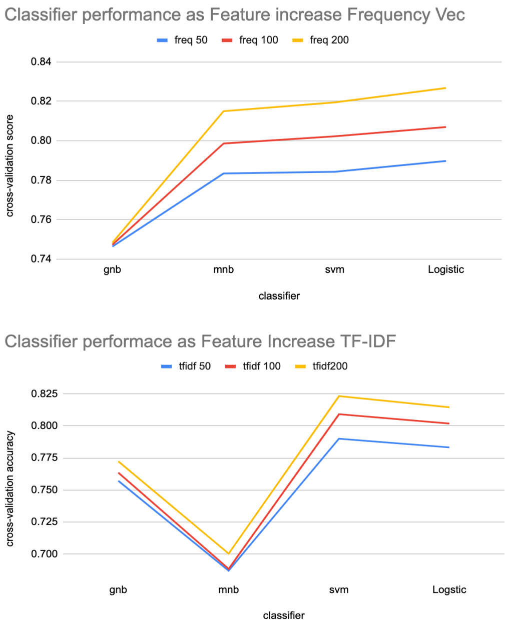


Figure 1- Frequency and TF-IDF performance as feature increase. Both feature selection method performance is averaged.

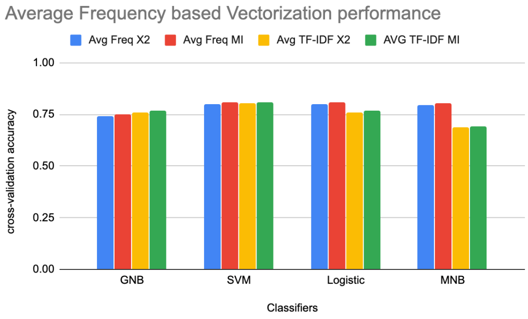
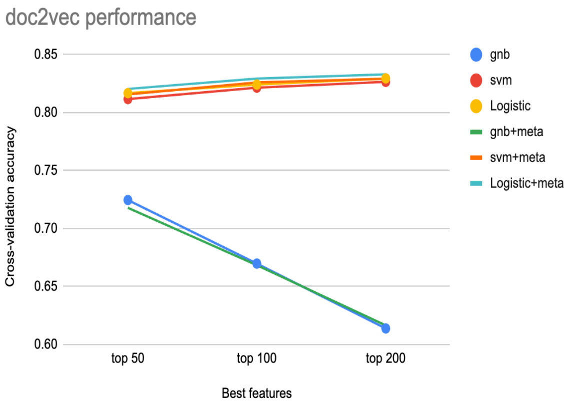
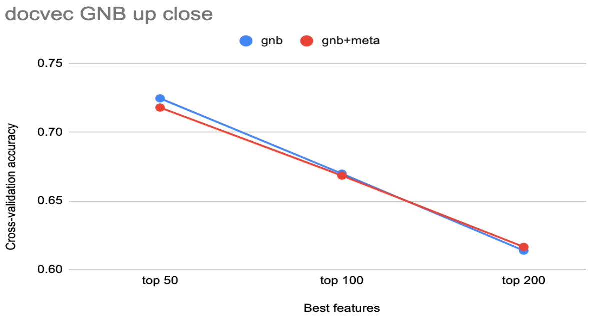


Figure 2- Average classifier performance from all top features per given feature selection method.

- From figure 1 we observed an expected behavior of an increase in accuracy as number of best features chosen increase.
- MNB performs better than GNB with frequency vectorization, opposite of TF-IDF. This may be resulted by the multinomial assumption which gives edge to frequency counts under conditional assumption. TF-IDF instead "undervalues" frequency as it prioritizes rare words. It may also be caused by the reason stated in the error analysis.
- SVM classifier performs the best in TF-IDF. Possibly due to the prioritizing rare words helps distinguish the polarity of the review.
- From figure 2, mutual information feature selection shows to be the better assumption than chi-squared due to taking correlation of features given class instead of relying just on occurrences per given class.

b. Doc2vec vs Doc2vec + meta





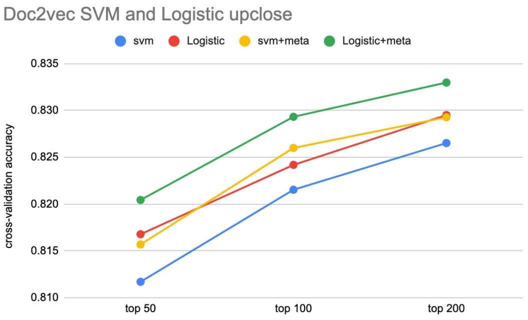


Figure 3- Doc2vec classification performances per number of top feature selection.

- Both Doc2Vec and Doc2Vec+ meta representation performs well with SVM and Logistic Regression classifier. This is due to the higher level of encoding that takes account the structure of the document compared to other stated vectorizers.
- It is also expected that the added meta-features improve classification accuracy as more information is obtained about the classifier.
- GNB does not seem to have a good performance. The assumption of independence does not fit the encoding method as Doc2vec takes account of the correlation of word with the document structure.

c. Overall

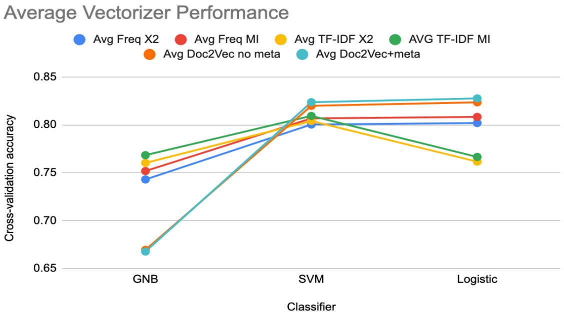


Figure 4- Average performance of all number of top feature selection of the engineered dataset per classifier.

	Best Freq X2	Best Freq MI	Best TF-IDF X2	Best TF-IDF MI	Best Doc2vec no meta	Best Doc2vec+ meta
GNB	0.744	0.753	0.769	0.774	0.725	0.718
SVM	0.817	0.822	0.826	0.830	0.827	0.829
Logistic	0.823	0.830	0.818	0.822	0.830	0.833

Table 1- Best performance of each vectorizer for any number top features of per given classifier.

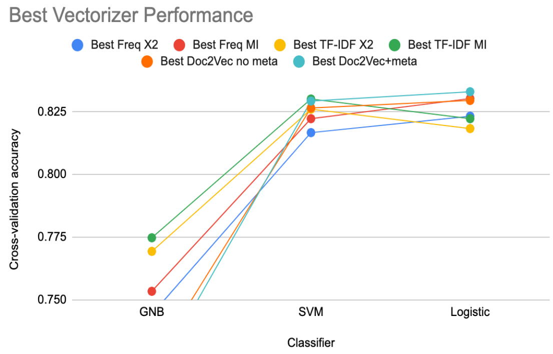


Figure 5- Line graph representation of table 1.

- Figure 4 shows that Doc2vec data averagely performs better than other vectorization in SVM and Logistic Regression with Doc2vec + meta having the highest performance.
- Logistic Regression is a form of probabilistic classification. Which models the data by using features as predictors. Thus, Logistic is observed and expected to have higher performance as we generate new features from the text.
- SVM is observed to be a more consistent performer. This may be due to the data is close to being linearly separable.
- Taking the best performance of each method for each classification shows that Doc2vec with added meta-features with Logistic Regression is the best performance.
- One interesting finding however is that Frequency vectorized with logistic regression data performs second best. This may be caused by outlier result from cross-validation accuracy.

VI. Final Kaggle Result

For Kaggle submission, we use the top four average vectorizer performance. Which are Logistic Regression and Support vector machine with Doc2vec and Doc2vec with added meta. With expectation that Logistic Regression with Doc2vec + meta performs the best.

Filename	System	Accuracy
kaggleresult.csv	Logistic Regression Doc2Vec + Meta	0.83420
kaggleresult3.csv	Logistic Regression Doc2Vec	0.83182
kaggleresult4.csv	SVM Doc2vec	0.84133
kaggleresult5.csv	Svm Doc2vec + Meta	0.84038

Our Kaggle results differ from our experiment. SVM with Doc2vec seems to have the highest prediction. This could be due to the test set being more linearly separable than the train set trained. Overfitting may occur for on doc2vec + meta result which leads to lower accuracy than just doc2vec.

VII. Conclusion

From our investigation, we can conclude that the right chemistry of vectorization technique and feature selection can improve a specific classifier's performance. Our experiment result indicates that Logistic Regression with Doc2vec + metadata performs the best whilst Kaggle result indicates SVM with doc2vec performs better. Future consideration of this experiment includes implementing a stacking system with random forest classifier and other base classification, implement deep or network-based methods, and use a higher number of features and perform wrapper or embedded technique to find an optimal number of features.

VIII. References

Mukherjee, A., Venkataraman, V., Liu, B. & Glance, N. What Yelp fake review filter might be doing? 7th International AAAI Conference on Weblogs and Social Media, 2013.

Rayana, S. & Akoglu, L. Collective opinion spam detection: Bridging review networks and metadata. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015. 985-994.