HW4

Multinomial Naïve Bayes

Bing-Je Wu

IST 736 – Text Mining

Table of Contents

[Introduction 2](#_Toc38661338)

[Analysis and Models 2](#_Toc38661339)

[About the Data 2](#_Toc38661340)

[Data Processing 3](#_Toc38661341)

[Models 4](#_Toc38661342)

[Sentiment 4](#_Toc38661343)

[Authenticity 5](#_Toc38661344)

[Results 7](#_Toc38661345)

[Conclusion 8](#_Toc38661346)

[Reference 8](#_Toc38661347)

# Introduction

Reviews are everywhere on the internet. People tend to write their thought for feeling about something. It could a though of a dish from a restaurant. Or, it could be a feeling about a movie. Even, it can be a complement for a seller from an e-commerce platform. There are tons of reviews produced every day, every second. Among the enormous reviews, it may have fake reviews with intention of destroying someone’s reputation. And, that is the cue for information science comes into play. Having the ability to classify the reviews good or bad, fake or true, becomes a good topic for researcher doing research.

Analysis and Models  
‘MultinomialNB’ (Multinomial Naive Bayes) algorithm and ‘cross\_val\_score’ function from scikit learn library were implemented for building models with train test split approach (holdout approach) and calculating the accuracy rate for k-fold cross-validate approach.

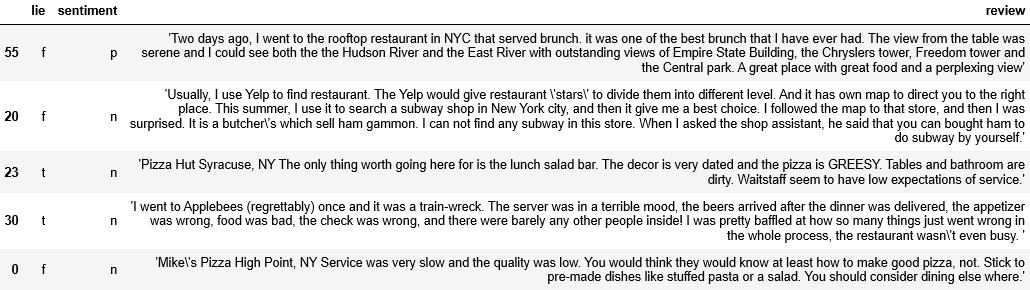
## About the Data

Data were assigned from online learning management system. It is a restaurant review dataset. The dataset contains two labels. One is ‘lie’ label; the other is ‘sentiment’ label. For the ‘lie’ label, it has two classes, ‘t or ‘f’. If a restaurant review is a lie, it will be labeled as ‘f’ (fake). On the other hand, if a review is not a lie, then, it will be labeled as ‘t’ (true). For the ‘sentiment’ label, it contains two classes, ‘n’ or ‘p’. If a review is negative, it will be assigned a label as ‘n’. Otherwise, it will be assigned a label as ‘p’ if it is a positive review.

**Table 1.** *Dataset*

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Data Type |
| lie | Label of authenticity (true or fake, lie detection) | String |
| sentiment | Label of sentiment (negative or positive) | String |
| review | The text of restaurant review | String |

**Figure 1**. *Sample of data before cleaning*

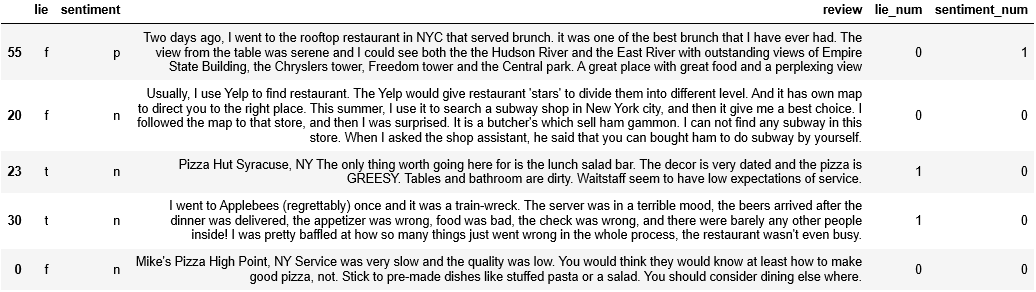
**

Data Processing  
To perform a classification prediction task, several cleaning steps must be done before building a model. By exploring the dataset, it is easy to notice that the ‘review’ column is not standard. Some back slashes are in the text. There are leading and trailing apostrophe in the string as well. It will be a problem when a ‘CountVectorizer’ function is applied to the dataset. Therefore, a customized function, ‘clean\_str()’, was applied to the dataset to remove the trailing and leading apostrophe and redundant black slashes. Two dictionaries, ‘lie\_mapper’ and ‘sentiment\_mapper’, were used as mapper to convert the code to numeric for getting the feature ranking in the later process.

**Table 2.** *User defined functions for pre-processing*

|  |  |  |
| --- | --- | --- |
| Function | Type | Explain |
| clean\_str() | Function | * Expect: A string with leading and trailing apostrophe and a back slash within the string. ex: " 'I found worm \in one of the dishes.' " * Modify: Remove the trailing and leading apostrophe and back slash * Return: Return a string |
| lie\_mapper | Dictionary | Convert ‘n’ (negative) to 0 and ‘p’ (positive) to 1 |
| sentiment\_mapper | dictionary | Convert ‘t’ (true) to 0 and ‘f’ (fake) to 1 |

**Figure 2**. *Sample of cleaned data after cleaning*



After finishing the data cleaning process, a pre-processing step needs to be performed for model building in text mining. The text data, ‘review’ column, need to be vectorized by applying ‘CountVectorizer()’ from scikit learn package. Two sets of ‘CountVectorizer’ parameter setting were used for model selection. One is in default setting; the other is in default setting but filtering out the stop words. Table 3 shows the detail comparison between two settings in ‘CountVectorizer’.

**Table 3**. *CountVectorizer setting comparison*

|  |  |  |
| --- | --- | --- |
|  | Default (No Stop Words Filter) | With Stop Words Filter |
| Parameters | CountVectorizer(analyzer='word', binary=False, decode\_error='strict', dtype=<class 'numpy.int64'>, encoding='utf-8', input='content', lowercase=True, max\_df=1.0, max\_features=None, min\_df=1, ngram\_range=(1, 1), preprocessor=None, stop\_words=None, strip\_accents=None, token\_pattern='(?u)\\b\\w\\w+\\b', tokenizer=None, vocabulary=None) | CountVectorizer(analyzer='word', binary=False, decode\_error='strict', dtype=<class 'numpy.int64'>, encoding='utf-8', input='content', lowercase=True, max\_df=1.0, max\_features=None, min\_df=1, ngram\_range=(1, 1), preprocessor=None, stop\_words='english', strip\_accents=None, token\_pattern='(?u)\\b\\w\\w+\\b', tokenizer=None, vocabulary=None) |
| Tokens | 1159 | 980 |

Models  
There are two prediction tasks. One is lie detection, predicting a review is true review or fake review. The other task is sentiment prediction, predicting a review is a positive review or negative review. A Naïve Bayes model was used for those tasks by implementing the ‘multinomialNB()’ algorithm from sklearn. The ‘multinomialNB‘ was implemented in default setting with Laplace smoothing. Data were split into training set and test set by implementing the holdout approach with the ‘train\_test\_split()’ function from sklearn package. 75% of the data were split into training set and 25% of the data were assigned into test set. Both Sentiment model and Authenticity model were built based on the same training set and test set with different labels. Sentiment model used ‘sentiment\_num’ column as the output and Authenticity used ‘lie\_num’ column as its label. The predictors are the same using the output vector from applying ‘CountVectorizer’ on ‘review’ column.

Sentiment   
Two candidate models were built for the Sentiment prediction task. One of models was using the document term matrix without the stop words filter applied on ‘CountVectorizer()’ setting. It had 86.96% of accuracy rate. The other model had stop words filter applied on ‘CountVectorizer()’ setting and yield an accuracy rate of 91.3% on the test set.

By comparing with two models through Table 4, it is easy to see that the second model, not containing stop words, has better performance not only in the accuracy but also in other metrics, such as precision, recall and f1-score. Thus, the second model was chosen for our best model and ready to be retrained using entire dataset with k-fold cross-validation approach to build the final model.

From the Table 5 shown below, it seems like the second model was overfitting. The k-fold cross-validation approach has given the penalty for the overfitting and still yielded the accuracy rate at 82.67%. The overall performance was good. Thus the 10-folds model is the best model for the Sentiment prediction task.

**Table 4**. *Models performance comparison for Sentiment task*

|  |  |  |
| --- | --- | --- |
|  | 1. Features Contain Stop Words | 2. Features Not Contain Stop Words |
| Null Model (all positive) | 60.87% | 60.87% |
| Accuracy | 86.96% | 91.3% |
| Confusion Matrix |  |  |
| Classification Report |  |  |

**Table 5**. *Models performance comparison with k-fold cross-validation approach for Sentiment task*

|  |  |  |
| --- | --- | --- |
|  | 5 folds | 10 folds |
| Null Model (all positive) | 50% | 50% |
| Accuracy | 80.35% | 82.67% |
| Recall | 88.89% | 89.0% |

Authenticity  
Two candidate models were built for the lie detection task. One of models used the document term matrix without the stop words filter applied on ‘CountVectorizer()’ setting. It had 34.78% of accuracy rate. The other model had stop words filter applied on ‘CountVectorizer()’ setting and yielded an accuracy rate of 43.48% on the test set.

**Table 6**. *Models performance comparison for Authenticity task*

|  |  |  |
| --- | --- | --- |
|  | 1. Features Contain Stop Words | 2. Features Not Contain Stop Words |
| Null Model (all true) | 60.87% | 60.87% |
| Accuracy | 34.78% | 43.48% |
| Confusion Matrix |  |  |
| Classification Report |  |  |

As the classification report shown above, both of models were not performing well. The null model, base model, having all predictions as ‘true’ review, has 60.87%. However, the task is to predict if a review is a lie. Therefore, accuracy is not the ultimate metric to look at. In the lie detection, knowing someone is telling a lie is important. Thus, the metric, ‘recall’, should be considered. Here, the second model has the recall at 55.56%. It is better than guessing. The model is worth to be retain using the entire dataset with k-fold cross-validation approach to see if its performance can be improved.

**Table 7**. *Models performance comparison with k-fold cross-validation approach for Authenticity task*

|  |  |  |
| --- | --- | --- |
|  | 5 folds | 10 folds |
| Null Model (all fake) | 50% | 50% |
| Accuracy | 61.93% | 56.44% |
| Recall | 65.33% | 61.5% |

Clearly, the overall performance has been improved in terms of accuracy rate and recall. The recalls of both models are higher than the base model, having all review predicted as fake review. By comparing with two cross-validated models, 5-folds and 10-folds, so far, it is obvious to say that 5-folds model is the best model for Authenticity task.

# Results

With the best models built, the next step to perform analysis on what words are those indicative words that can be helping prediction. A Naïve Bayes approach has been implemented to calculate the condition probability for each of words in different condition/class. A add-one smoothing (Laplace smoothing) approach was applied when calculating the condition probabilities. According to the sklearn documentation for multinomial Naïve Bayes algorithm, the add-one smoothing approach is to add ‘1’ on the nominator and add ‘number of vocabularies learned during the training’ on the denominator. Two odds ratios were also calculated in aid of finding the most indicative words. For the sentiment prediction, a ‘positive ratio’ was primary used to determine the most representative for the ‘pos’ class. It was calculated by having the condition probability of words in ‘pos’ class divided by the condition probability of words in ‘neg’ class. For the lie detection task, a ‘fake ratio’ was calculated by having the condition probability of words in ‘fake’ class divided by the condition probability of words in ‘neg’ class.

**Table 8**. *Feature Ranking in MultinomiaNB for both tasks*

|  |  |
| --- | --- |
| Sentiment | Authenticity |
|  |  |

By looking at the Table 8, it shows the top 20 indicative words for both Sentiment prediction and Authenticity prediction. For the sentiment prediction, it seems like some of the positive words, ‘amazing’, ‘best’, ‘love’, ‘friend’, ‘awesome’, ‘great’, ‘liked’, were listed on the list. It is consistent with the best model since the model has pretty good overall performance.

However, for the lie detection, it is not persuasive. There are some reasons could explain why. First, the best model was not very good although it did yield the satisfying result but still have room for improvement. Second, the size of the data was not big enough for an algorithm to learn the pattern of writing a fake review. Third, there are other factors need to be considered when one wants to perform the fake review detection. It could be the length of the review, the number of words used in a review, time writing a review…etc. Last, the ‘lie’ label could be not the ground truth. With the reasons mentioned above, one can say that the model did not learn the pattern very well. Other techniques should be considered for doing the Authenticity prediction task.

# Conclusion

Reviews can be a tool or an indicator for people to decide whether buy a product or not, whether go to a restaurant for eating or not. People tend to rely on reviews often and often. Many companies also provide the review function for customers to leave their thought and feeling about a product, a transaction, or a shopping experience. However, reviews can be deceiving. When it comes to business, companies will try to find a way to compete their competitors. In that case, some companies may take the unethical route to undermine others’ reputation. And, that hurts the company who are working their business normally and legally. With the technique of Authenticity detection, it helps those company come up smelling of roses. Therefore, the better Authenticity detection can balance the fairness of the market, disclose the truth to the public and unveil the harmful intentions.

# Reference

MultinomialNB, from scikit-learn [Web Link](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html) ; 2 cross\_val\_score, from scikit-learn. [Web Link](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html?highlight=cross_val#sklearn.model_selection.cross_val_score)  
3 Metrics and scoring, from scikit-learn. [Web Link](https://scikit-learn.org/stable/modules/model_evaluation.html#scoring-parameter) ; 4 Confusion matrix, from Wiki. [Web Link](https://en.wikipedia.org/wiki/Confusion_matrix)  
5 Simple guide to confusion matrix terminology. [Web Link](https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/) ; 6 Additive smoothing, from wiki. [Web Link](https://en.wikipedia.org/wiki/Additive_smoothing#Pseudocount)