**Analysis of Restaurant Reviews**

**Introduction**

Everyone has a favorite restaurant that he or she loves to eat at. A Restaurant provides with all the amenities that one requires for a shift of mood or change in environment to cherish the pleasant atmosphere and enjoy different cuisines to satisfy your hunger. Creating a brand in the restaurant business is a complex thing. It holds so many elements – the service, the food, the ambiance, the interior – but most of all, the PEOPLE.

The term Restaurant has its origins in Paris, where one A. Boulanger started as a soup vendor in 1765. A sign on the door said “restaurant”, referring to the restorative quality of the soups and broths served within. Entrées and main courses joined the menu, and the modern restaurant as we know it took shape. As with so many ideas that evolve over time, restaurants now serve a larger role in society. They have become places of social contact, of discovering new cultures and tastes from far-away lands, of spending an evening with your loved ones, of clinching business deals over a glass of wine, and so forth. All this, of course, in addition to the basic functions of “restoring” people with the help of good food, service and ambience.

During the last few years, reviews have become crucial to the success of a restaurant, as every restaurant owner knows good reviews can boost popularity and profitability, whereas terrible reviews even have the potential of closing businesses down. In a recent research report published by the experts at Website Builder, approximately 61 percent of customers have read online reviews about restaurants.

**Analysis and Models**

**About the Data**

The dataset is a collection of reviews of a popular movie. The reviews are labelled with two labels. One is a sentiment analysis label which tells if the review is a positive or a negative one. The other label is the lie or deception label that tells if the review is a true one or a fake one. There are three columns and 92 rows in this dataset. Now, the challenge with this dataset is the fact that the reviews are spread across multiple columns and is not clubbed in a single column. So, lots of cleaning and pre-processing is to be done on this data before analyzing it.

First, the sentiment and lie labels are separated from the dataset and stored in separate variables. Then, the rest of the columns are simply merged using the ‘+’ symbol into a single column called *review.* Next, any null or nan values are removed by replacing it with a blank string.

To analyze text data, one of the most important concepts that must be applied is to get the data into a matrix format where the rows are the different reviews and the columns are the words and the frequencies of these words in each review make the values inside this matrix. In order to accomplish this, first, the words are tokenized, and unnecessary numbers, punctuations, hyphens and symbols are removed from the text. All the words are also converted into lower case and English stop words are also removed.

Now, we have two data frames – one for the reviews with the sentiment label and another for the reviews with the lie label. The review and lie columns are converted into an array and stored in separate variables for later use. This is just one type or example of vectorization. The Count Vectorization function is used here to vectorize the reviews. Out of the different parameters that can be used, the first model was generated by only removing the stop words. A second model was developed by removing stop words and using lemmatization. A third model was generated with the above two parameters and also removing patterns off the data. These three models were applied to both the sentiment label and lie label data frames.

Results of the first model are displayed in Fig 1.1.

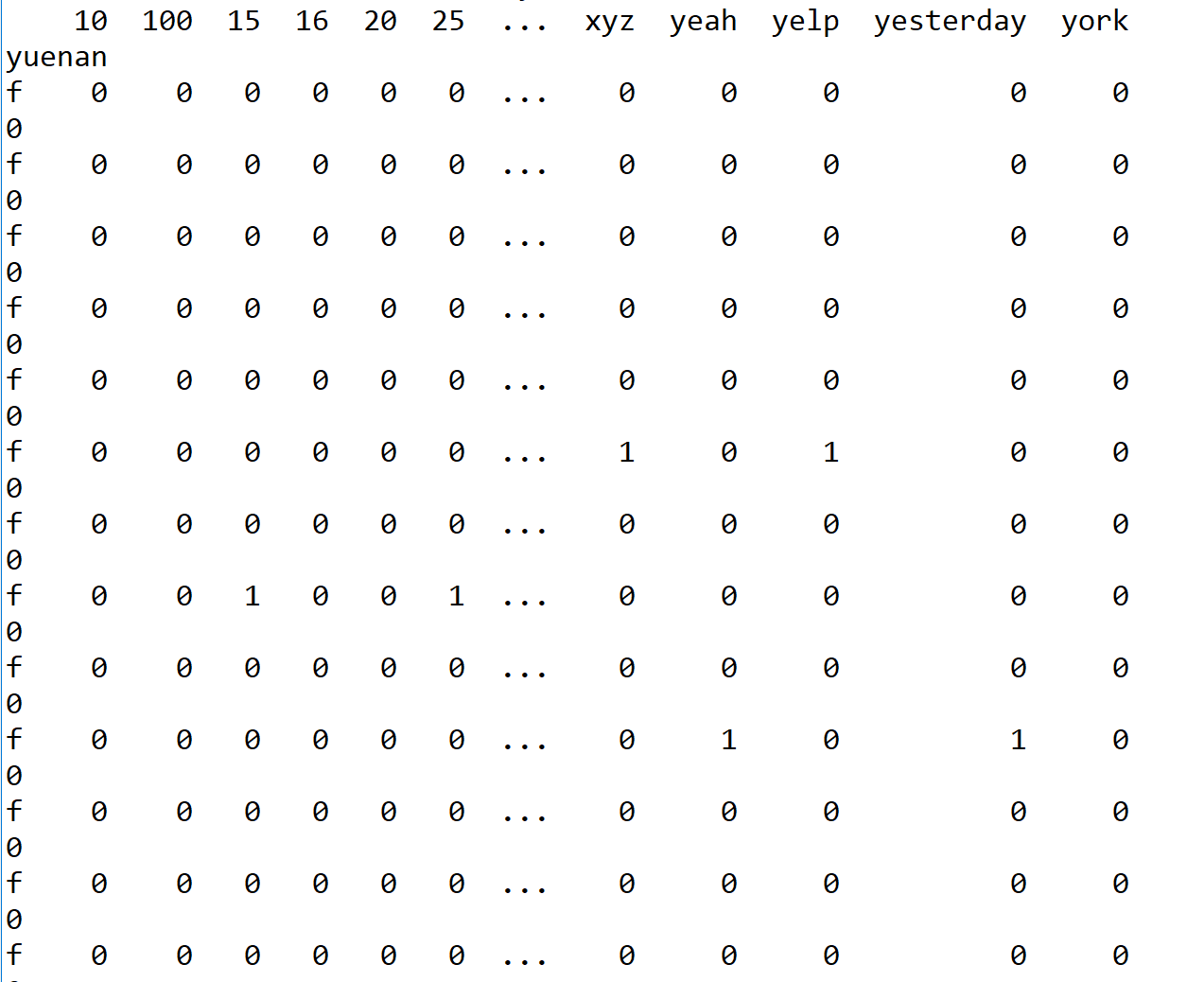


Figure .1 Lie Label Data frame after Vectorization

Further, some exploratory data analysis was done to understand the data structure before deep diving into algorithm application. Fig 2.1 shows the words with the highest frequency and Fig 2.2 shows words with the lowest frequency in the lie label dataset. ‘food’, ‘restaurant’ and ‘place’ are some of the most common words in the dataset. And, words like ‘norm’, ‘course’, ‘number’, ‘cottage’ are the least frequently used in the reviews.

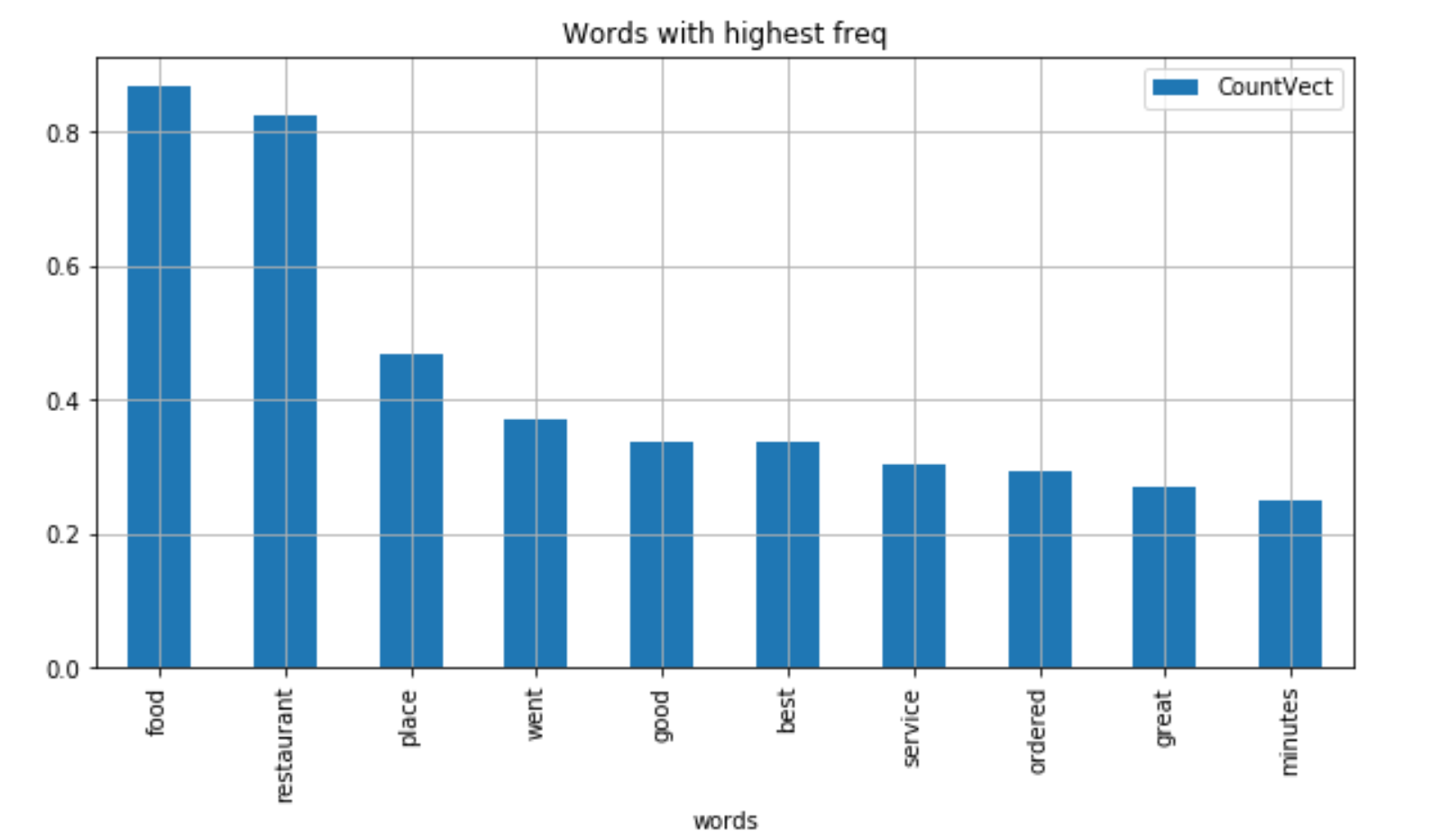


Figure 2.1 Words with high Frequency

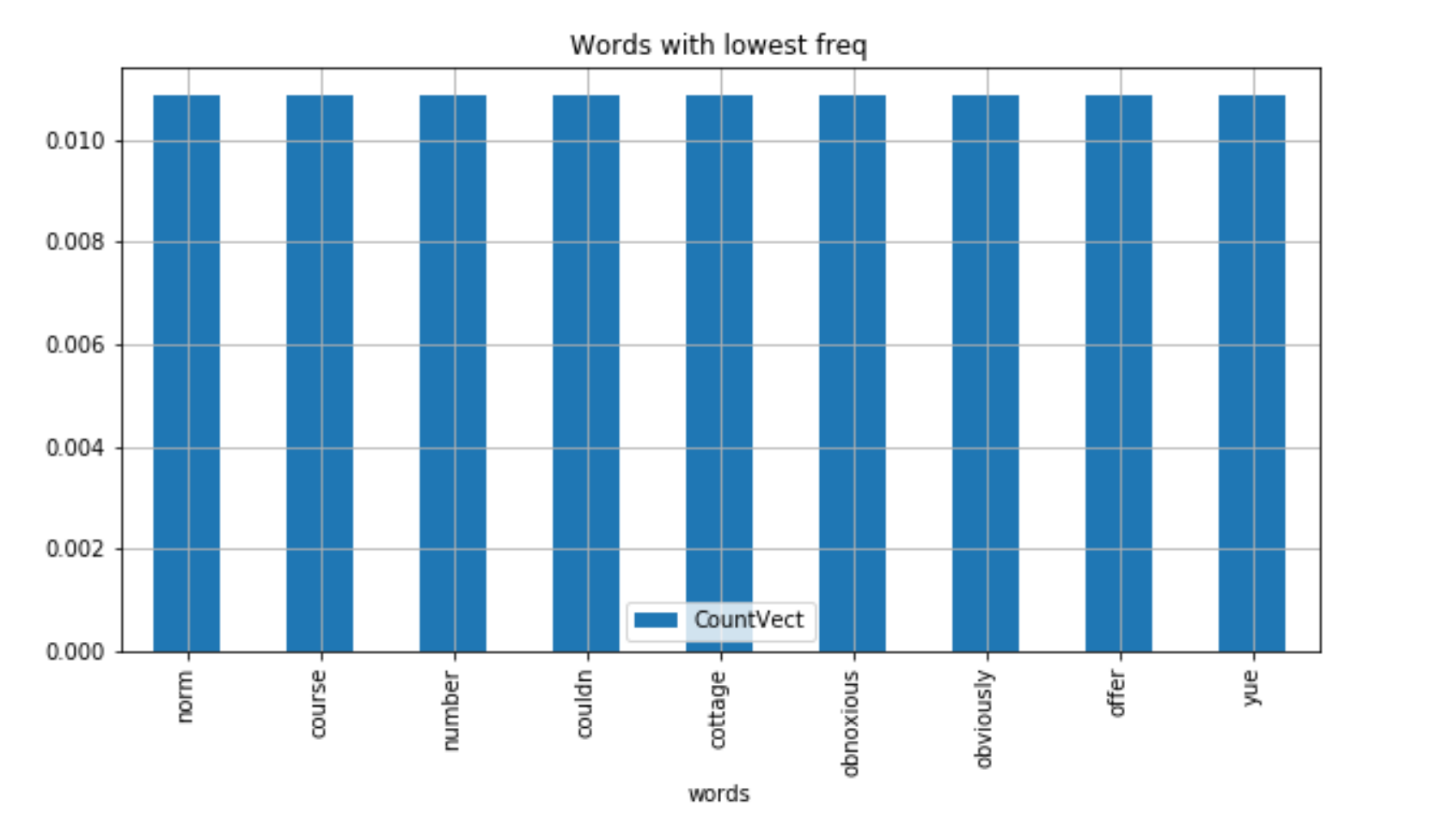


Figure .2 Words with low Frequency

To make the visualization more appealing, a word cloud was generated (Fig 3.1) to see the most and least common words in the data frame. The below word cloud shows the results from model 1 lie label dataset.



Figure .1 Word cloud for Model 1 lie dataset



Figure .1 Word cloud for Model 2 lie dataset

Figure 4.1 shows the word cloud results from model 2 where lemmatization was done in addition to stop word removal. Although most of the results are similar, there are some new words with higher frequency like ‘wa’, ‘ordered’ and ‘plate’.



Figure .1 Word cloud for Model 3 lie dataset

Fig 5.1 shows the results from model 3 where token pattern was set to eliminate some tokens apart from stop word removal and lemmatization. The results of both model 2 and model 3 are quite similar with no major differences.

**Classification Techniques**

As per the definition provided in the book ‘Introduction to Data Mining’, classification is the task of learning a target function *f* that maps each attribute set *x* to one of the predefined class labels *y*. Classification techniques are most suited for predicting or describing datasets with binary or nominal categories. Different classification models can be built from an input data set using classification techniques. Some of the examples include decision-tree classifiers, neural networks, random forests, kNN nearest neighbor classifier and support vector machines. The classification techniques that are going to be used for the analysis of this text data is the Multinomial Naïve Bayes model.

There are two ways in which the dataset is divided into training and test sets. The first method is to just use the train\_test\_split function that is explained below.

Before applying this model, the dataset is first divided into training and test sets. The model learns with the training data and applies the learned algorithm on the test data to get good predictions. Using the train\_test\_split function, the arrays builder (an array of all the words and its frequencies) and lie (the corresponding lie label – fake or true review) is divided into training and test sets. The resulting training dataset has 64 rows and the test dataset has 28 rows.

The second method is using cross-validation method using 10-folds to separate the training and test sets randomly. In K-Folds Cross Validation we split our data into k different subsets (or folds). We use k-10 subsets to train our data and leave the last subset (or the last fold) as test data. We then average the model against each of the folds and then finalize our model. After that we test it against the test set.

Two different algorithms were applied to the three models developed. One is the multinomial Naïve Bayes algorithm and the other one is the Bernoulli Naïve Bayes algorithm. And, every model used both cross validation methods and the simple train test split method to split the training and testing sets and the results were compared.

**Multinomial Naïve Bayes Classifier Algorithm**

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values. It is called naive Bayes because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable. Rather than attempting to calculate the values of each attribute value, they are assumed to be conditionally independent given the target value. This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not hold.

For the review lie and sentiment label classification dataset, the Multinomial naive Bayes function from the library sklearn package is used by holding the label which in this case is the column Lie as the Y variable and use all other words in the training data to create a model. This model is then used to predict the labels of the test data. The results are then compared with the labels of the test data that we have stored in a separate variable.

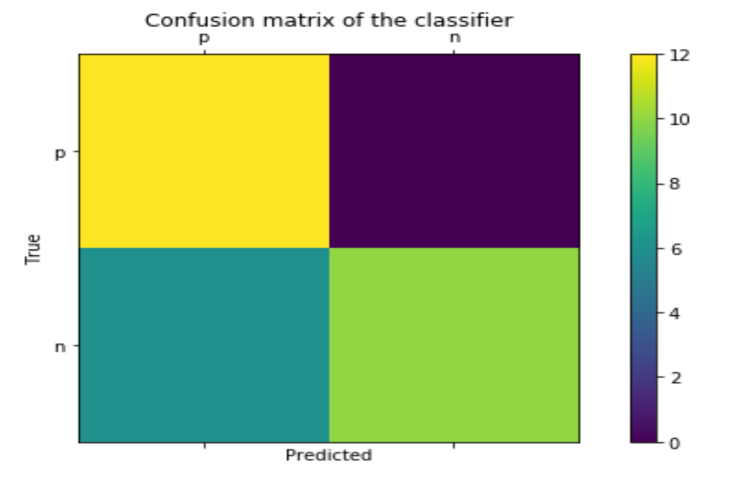
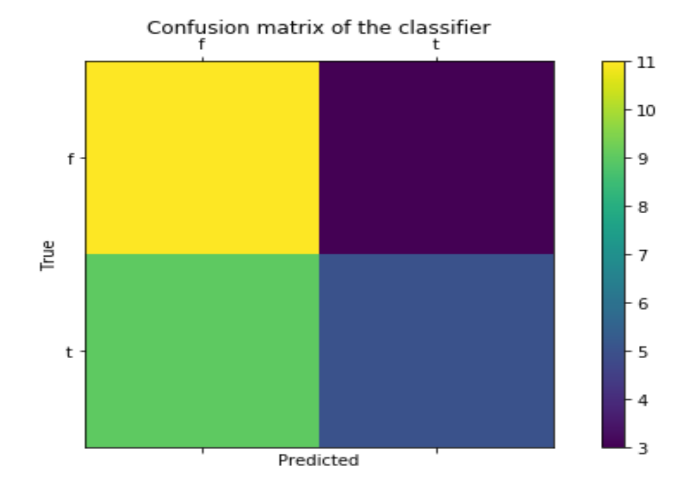


Figure .1 Lie prediction Model 1 Figure 6.2 Sentiment prediction Model 1

Fig 6.1 shows the number of records that the model 1 correctly and incorrectly classified as a true or fake review. Fig 6.2 shows the shows the number of records that the model 1 correctly and incorrectly classified as a positive or negative review. Model one’s performance on the lie dataset is that it was able to classify more fake reviews accurately than true reviews. But the incorrect classification of a true review as a fake review is high.

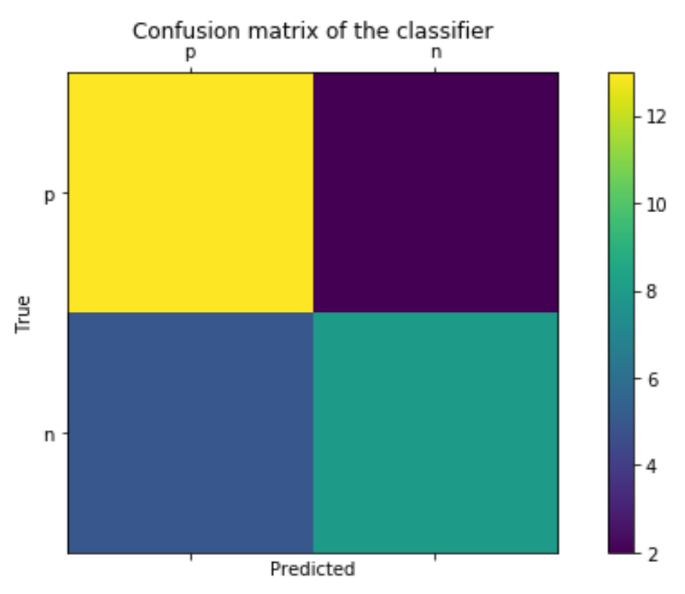
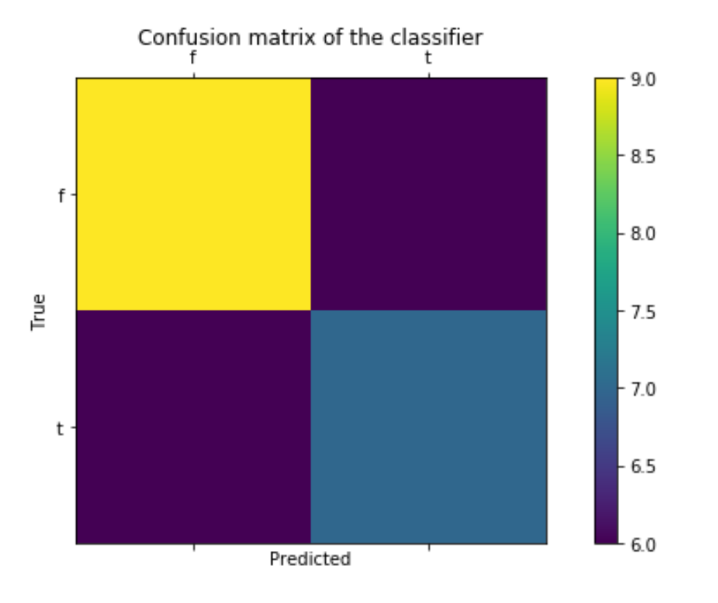


Figure 7.1 Lie prediction Model 2 Figure 7.2 Sentiment prediction Model 2

Fig 7.1 shows the number of records that the model 2 correctly and incorrectly classified as a true or fake review. Fig 7.2 shows the shows the number of records that the model 2 correctly and incorrectly classified as a positive or negative review. The darker purple colors on the diagonal shows that the number of records incorrectly classified on the lie label is lesser than that of model one whereas Model two’s accuracy of predictions is lesser compared to model one.

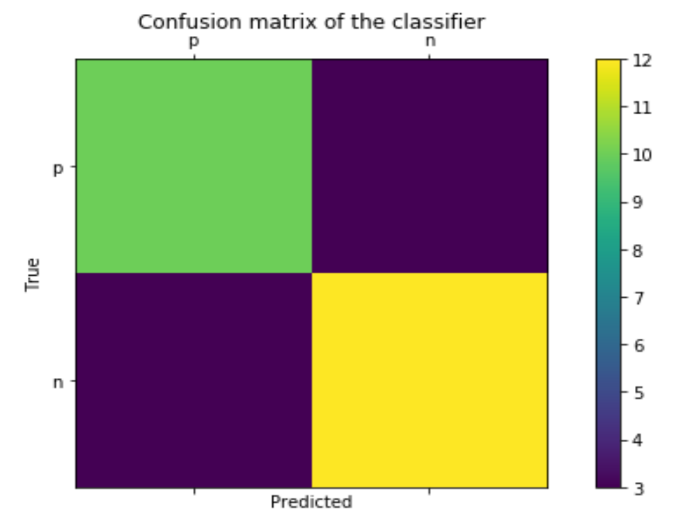
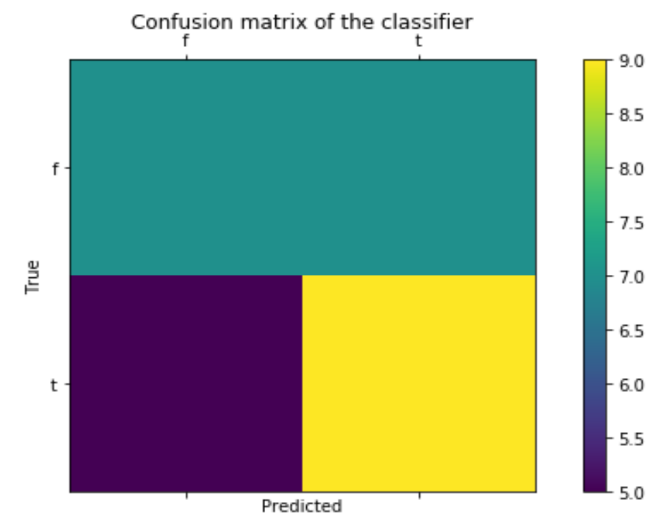


Figure 8.1 Lie prediction Model 3 Figure 8.2 Sentiment prediction Model 3

Model three’s performance on the lie dataset is worse than the first two models. It was able to detect majority of the true reviews accurately but fails in predicting fake reviews. Whereas on the sentiment dataset, model three performs exceptionally well. Fig 8.2 visualizes this by showing purple diagonal from the right to the left.

**Bernoulli Naïve Bayes Classifier Algorithm**

The Bernoulli naive Bayes classifier assumes that all our features are binary such that they take only two values (e.g. a nominal categorical feature that has been one-hot encoded). In the multivariate Bernoulli event model, features are independent Booleans(binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks, where binary term occurrence features are used rather than term frequencies.

For the review lie and sentiment label classification dataset, the Bernoulli naive Bayes function from the library sklearn package is used by holding the label which in this case is the column Lie as the Y variable and use all other words in the training data to create a model. This model is then used to predict the labels of the test data. The results are then compared with the labels of the test data that we have stored in a separate variable.

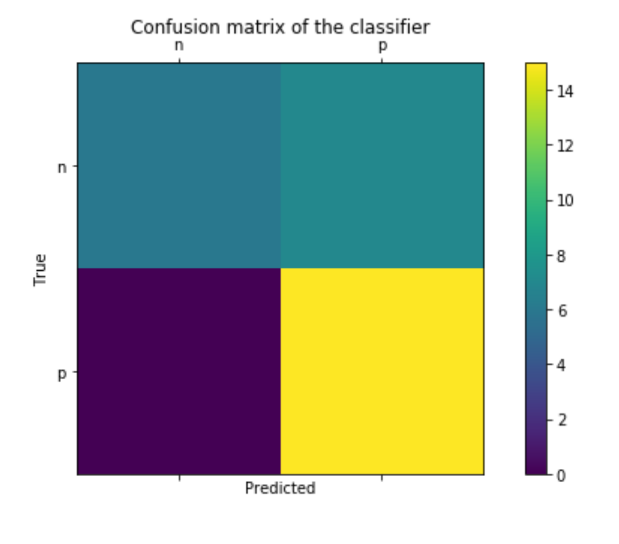
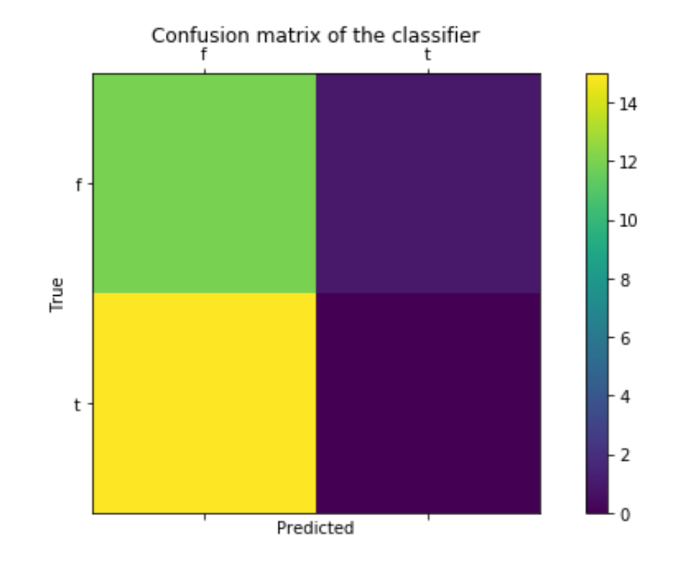


Figure 9.1 Lie Prediction Model 1 Figure 9.2 Sentiment Prediction Model 1

Fig 9.1 shows that the model was able to predict true reviews accurately but incorrectly classified many true reviews as fake reviews. Fig 9.2 shows that the model was able to predict many positive reviews accurately but classified some negative reviews as positive reviews.

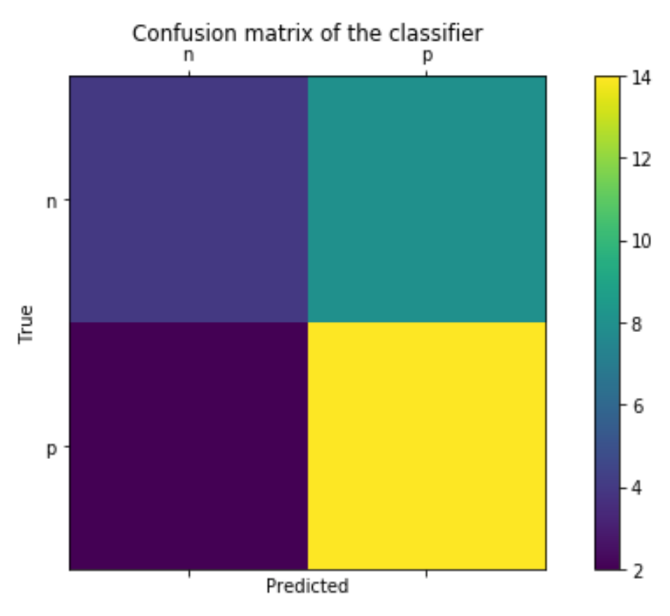
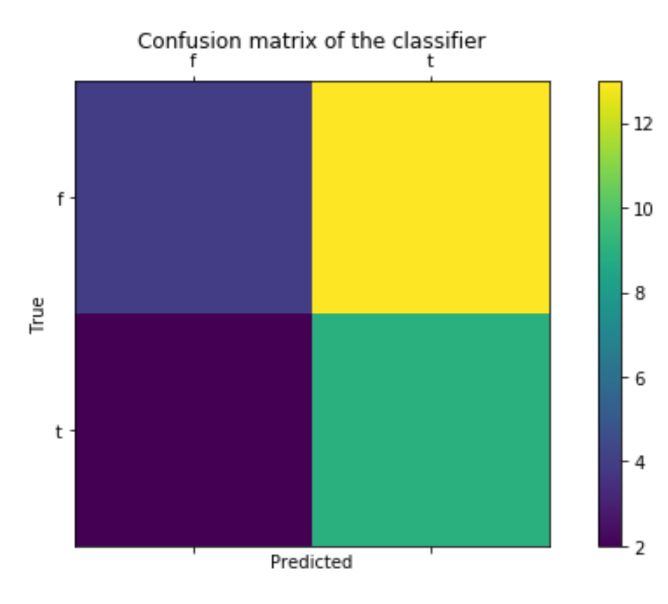


Figure 10.1 Lie Prediction Model 2 Figure 10.2 Sentiment Prediction Model 2

Whereas in Model 2, Fig 10.1 shows that the model was able to predict true reviews accurately but incorrectly classified many fake reviews as true reviews. Fig 10.2 shows that the model was able to predict many positive reviews accurately but classified some negative reviews as positive reviews.

The difference between model 1 and model 2 results in sentiment label review is that model two’s performance worsened in predicting negative reviews accurately than model one.

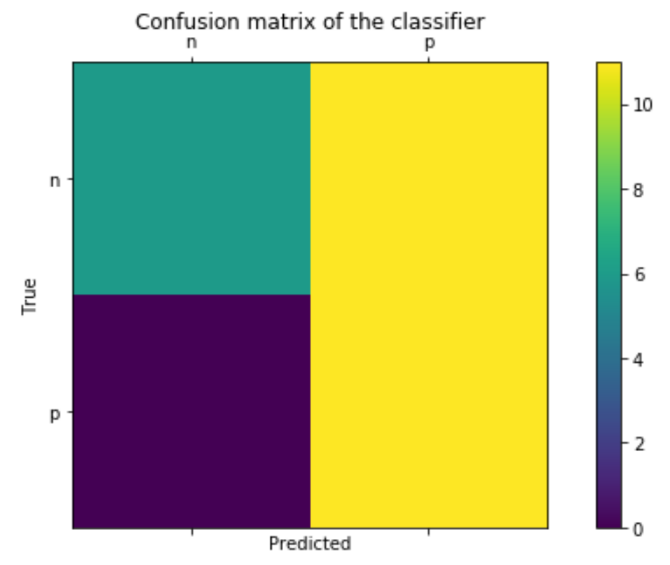
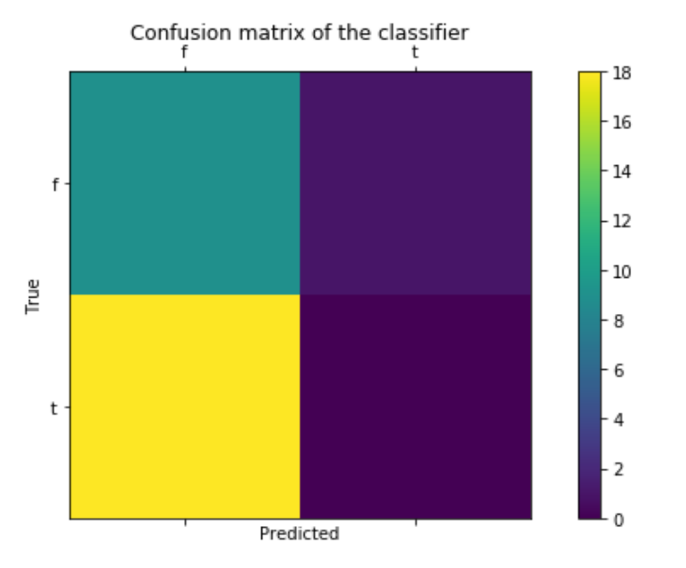


Figure 11.1 Lie Prediction Model 3 Figure 11.2 Sentiment Prediction Model 3

Model three’s performance on the lie dataset worsened even more because it predicted many true reviews as fake reviews. Model three’s performance on the sentiment dataset is also poor in the fact that it predicted many negative reviews as positive reviews apart from predicting a high number of positive reviews accurately. Fig 11.1 and 11.2 depict the above explained results.

**Results**

**Multinomial Naïve Bayes Classifier Algorithm**

The accuracy score of the NB model on the lie label dataset was only 54% whereas the 10-fold cross validation score went up to 65%. Subsequent models 2 and 3 (with lemmatization and token pattern removal) did not yield in better results. They, in fact, stayed in the same range without much variation. But, the best model for the lie dataset was Model 1 with a 10-fold cross validation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Lie Dataset** | | | |
| **Multinomial NB** | | | |
|
|  | Model 1 | Model 2 | Model 3 |
| Accuracy Score | 54% | 50% | 57% |
| CV Score - 10-fold | 65% | 62% | 62% |

Figure 2.1 Multinomial NB Lie Dataset results

For the sentiment label dataset, the accuracy score of Model 1 and Model 3 were similar with 79% accuracy rate. But the scores improved with 10-fold cross validation to 84% and this didn’t change much for the other two models.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sentiment Dataset** | | | |
| **Multinomial NB** | | | |
|
|  | Model 1 | Model 2 | Model 3 |
| Accuracy Score | 79% | 75% | 79% |
| CV Score - 10-fold | 84% | 83% | 83% |

Figure 12.2 Multinomial NB Sentiment Dataset results

From the above results, one can conclude that lie detection is far more a difficult process than sentiment classification of reviews. And, so far, 10-fold cross validation seems to yield better accuracy results than changing the parameters of the algorithm model itself.

**Bernoulli Naïve Bayes Classifier Algorithm**

For Bernoulli, the accuracy score of the model on the lie label dataset was only 43% whereas the 10-fold cross validation score went up to 67%. Subsequent models 2 and 3 (with lemmatization and token pattern removal) did not yield better results. All three models resulted in the same accuracy rate of 67% with 10-fold cross validation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Bernoulli** | | | |
|
|  | Model 1 | Model 2 | Model 3 |
| Accuracy Score | 43% | 46% | 32% |
| CV Score - 10 fold | **67%** | 67% | 67% |

Figure 13.1 Bernoulli NB Lie Dataset results

For the sentiment label dataset, the accuracy score of Model 1 and Model 2 were similar with 75% accuracy rate. But the scores improved with 10-fold cross validation to 80% and this didn’t change much for the other two models.

|  |  |  |  |
| --- | --- | --- | --- |
| **Bernoulli** | | | |
|
|  | Model 1 | Model 2 | Model 3 |
| Accuracy Score | 75% | 75% | 61% |
| CV Score - 10 fold | 80% | 80% | 80% |

Figure 13.1 Bernoulli Sentiment Dataset results

So far, comparing the two algorithms on 3 different models, the Bernoulli algorithm fared better on both the lie and sentiment label dataset than the multinomial NB algorithm. And, in terms of which training split model worked, the 10-fold cross validation method yields the best results on any of the algorithms.

**Conclusion**

While a significant number of people tend to read online reviews prior to visiting a restaurant to dine or hosting an event, it is also worth pointing out that around 34 percent of diners currently choose restaurants based solely on information offered on peer review websites. This means that most diners disregard the restaurant’s website or social media pages, preferring to rely on data present on review sites, further increasing their importance and influence on the market. Another interesting fact is that approximately 53 percent of the coveted 18 to 34-year-old demographic reported that online reviews play an important role into their dining decisions.

Hence, with people trusting on reviews to make their next decision which directly impacts the restaurant business, it’s imperative that these reviews are true and not fake. Only 27% of people trust reviews that they believe are authentic. Negative reviews on google search can cause a restaurant to lose about 70% of its potential customers. Hence, it’s very important to detect fake reviews and take them off to sustain and bring more customers.

**Sources:**

<https://www.modernrestaurantmanagement.com/the-impact-of-reviews-on-the-restaurant-market-infographic/>

<https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6><https://www.machinelearningplus.com/nlp/lemmatization-examples-python/>