|  |
| --- |
| IST707 text MINING |
|  |

|  |  |
| --- | --- |
| Date Submitted  8/19/2020 | WK7: Scikit-Learn |
| Professor | Dr. Ami Gates |
| Submitted by: | Beverlyn Tucker |

**Analysisand Model**

The datasets are restaurant review provided by Syracuse. This week is about comparing between Naïve Bayes classification and Support Vector Machine Classification. For Naïve Bayes MNB and Bernoulli classification together with count vectorizer and TFIDF vectorizer. After wards applying normalization in each model. Then the same approach for support vector machine: Linear, polynomial, and Radial classification model.

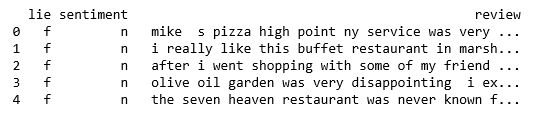
Compare the top 5 words for both positive negative categories. Report the confusion matrix, recall, and precision for both models. With 10 highest and 10 lowest frequency. Both models must have the same parameter features.

**Data prepation**

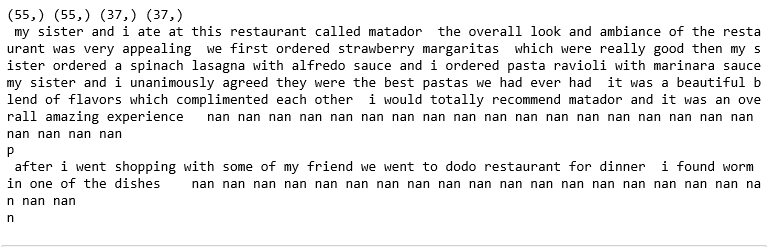
Data restaurant review. Removing special characters, utilizing stop word English. When spitting the data use train\_test\_split from Scikit Learn with .04 test -et the rest if training. Before performing the model pass throw all the words to stemming process

**Results**

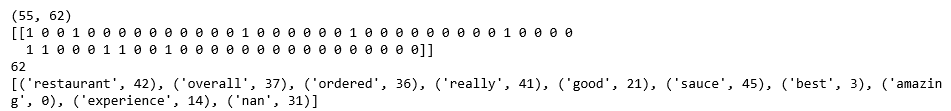
**Fisrt 5 of the data sets after removing special characters**



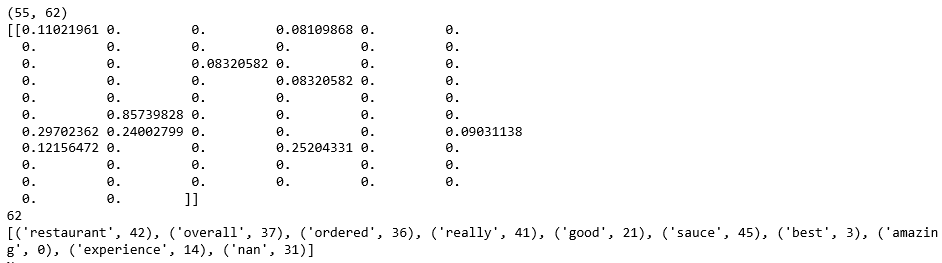
**data split**



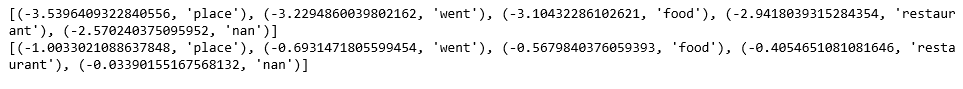
**Train vectorization Results**



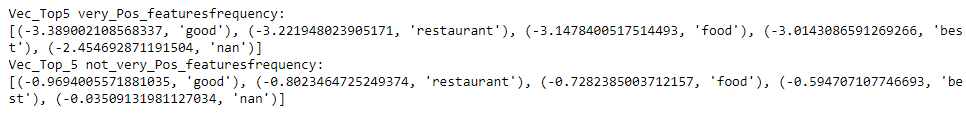
**Train TFIDF results**



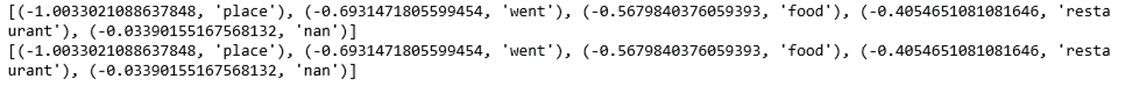
**ComParisson on vectorizer very\_negative\_features**



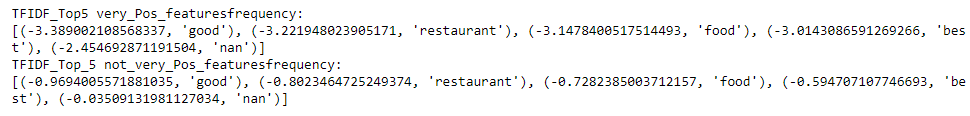
**ComParisson on vectorizer very\_Positive\_features**



**ComParisson on TFIDFvectorizer very\_negative\_features**

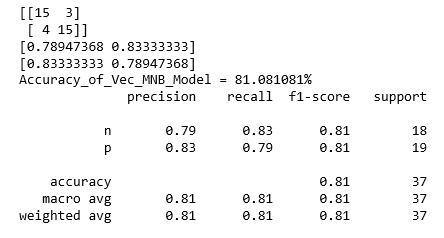
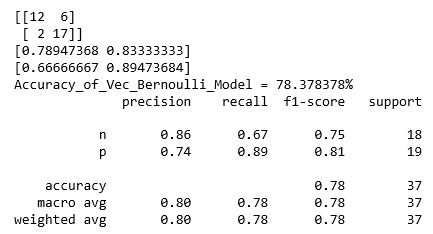
****

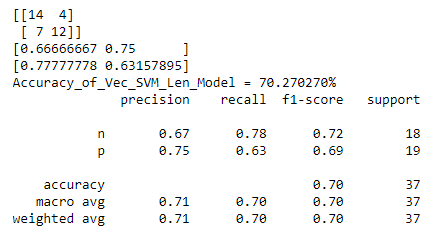
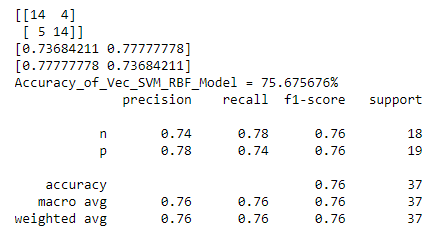
**ComParisson on TFIDFvectorizer very\_positive Features**

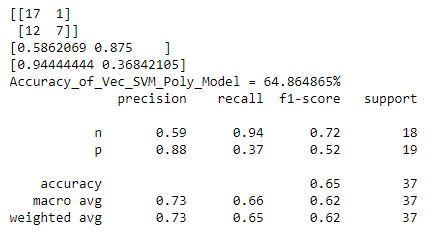


## Model Prediction and Evaluation Results

## Naive Bayes and SVM Count Vectorizer Prediction and Evaluation Results

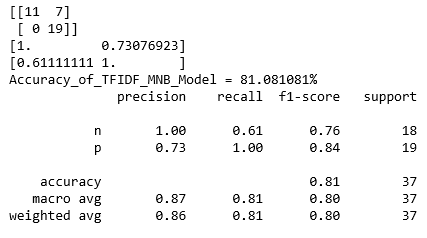
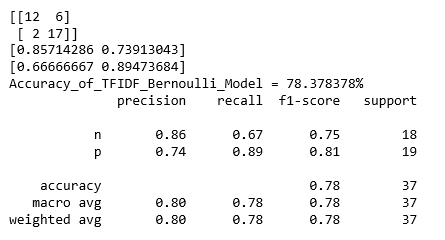


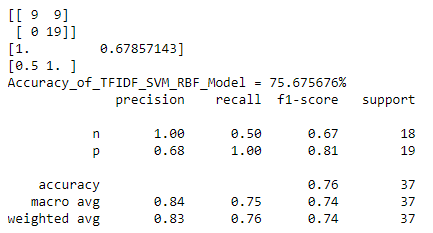
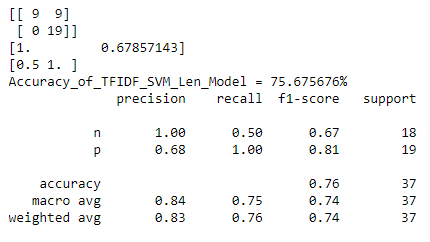
## Model count Vectorization Comparison Results

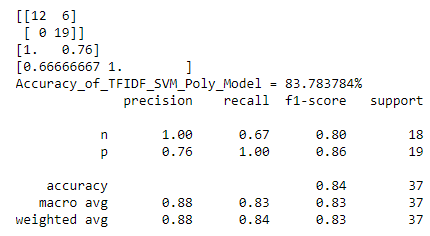
When comparing the Naive Bayes and Support Vector Machine model on vectorizer, MNB has better results, the accuracy of 81%, a weighted average of 81%, weighted precision of 81% and a recall of 81%. SVM Polynomial kernel has the lowest accuracy results of 65% and with weighted precision of 69%

## Model TfidfVectorizerPrediction and Evaluation

## Naive Bayes and SVM TfidfVectorizer Prediction and Evaluation Results





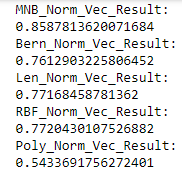
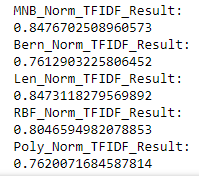
## Model count TFIDFVectorizer Comparison Results

The model results in TfidfVectorizer between 5 models. Naive Bayes model MNB and Bernoulli vs. SVM model with 3 kernels Linear, radial, and polynomial. SVM polynomial took the lead with 84% accuracy with a weighted precision average of 84%, and with a weighted f1-score average of 83%. Followed by MNB with 81%, with a weighted precision average of 86%. Models that were not performing so good are SVM Linear, and radial perform in the same manner across from precision, recall, f1-score even support, the accuracy of 76%, and with a weighted precision average of 83%

**Normalized Results Vectorizer and TFIDF**

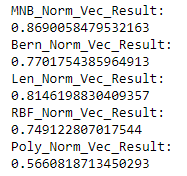
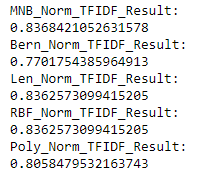
For nomalization use the same approach in all model to see which model perform better. SVM utilize C=1, kernel='poly', verbose=False, degree=3 and Naive Bayes alpha =1

**CV=3 CV=3**

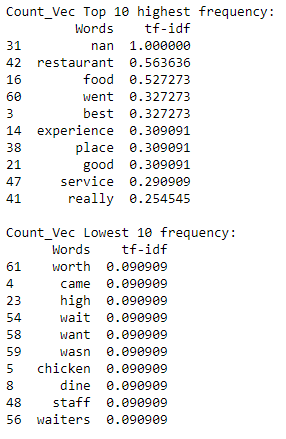
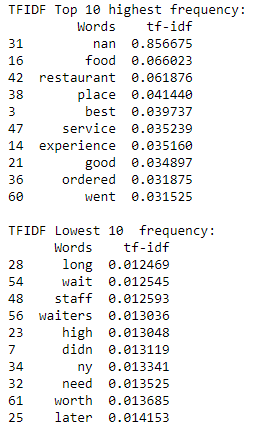
Count vectorizer and TFIDFVectorizer with CV=3 MNB performed very good took the lead in both counvectorizer 86% and 85% in FTIDFvectorizer. SVM linear TFIDF is comparable with MNB. Model performed the least is CVM polynomial with 54%.

**CV=5 CV=5**

When increasing fold from 3 to 5, in count vectorizer both model increases, the model benefited the most is SVM linear increase more than 4%. from 77% to 81%. On the other hand, on TFIDFVectorizer there is a trade-off. MNB went down at least 2%, also the linear model. The model that performed well with CV=5 is SVM polynomial from 76% to 81%.

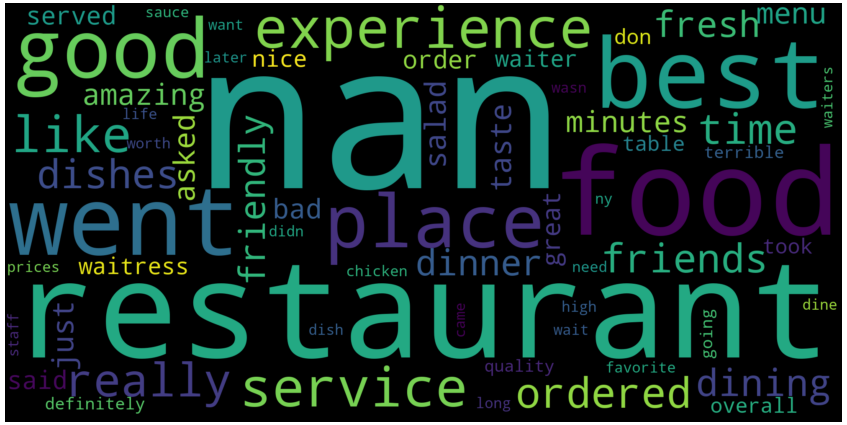
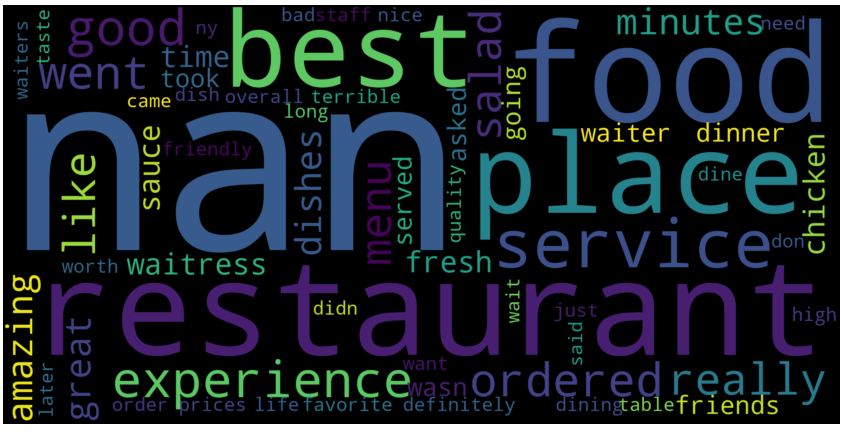
**Normalize top 10 and Lowest word Frequency Results**

The result on the top 10 frequency between vectorizer and TFIDF is interesting. The word restaurant in count vectorizer with 56% compare to 6% in TFIDF. Next is food in Count vectorizer 53% and in TFIDF is 7%.

The count vectorizer has high bias when using the TFIDF it balance the requency.

CountVectorizer word cloud FTIDFVectorizer word Cloud

**Conclusion¶**

The Multinomial Naive Bayes model has been leading across the study. For this dataset, it proved superior to the three models that has created. It is interesting to discover that exploratory analysis indicates.

SVM play an essential role in text classification. Count vectorizer and TFIDFvictorizer have a response differently; the text classification is essential in the bag of words. In the beginning, MNB count vectorizer has better results compared to the SVM models. It is a good start, after taking to account performing inverse frequency, which is imperative to handle the biases. SVM polynomial outperformed the Naive Bayes MNB.

Cross-validation in count vectorizer and TFIDF vectorizer Naive Bayes MNB perform consistently. SVM linear not far behind about the same with MNB TFIDFvectorizer. When performing CV=5 fold cross-validation, Naive Bayes MNB countvectorize took first place with 86%. And then on TFIDF vectorizer MNB, Linear and Polynomial delivering about the same results.