SENTIMENT ANALYSIS WITH MACHINE LEARNING TECHNIQUES

Abstract

Sentiment Analysis is a branch of Natural Language Processing used to identify attitudes and viewpoints in short text content, such as Twitter posts and reviews (Bhavitha et al., 2017). In this report, we conducted the following steps to investigate the polarity in restaurant reviews: (1) Preliminary selection and comparison of well-known supervised machine learning models for sentiment analysis: Naive Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), K-Nearest Neighbors (KNN), and Logistic Regression (LR). (2) Evaluation of contrasting feature engineering methods based on the selected model, Logistic Regression (LR), and Linear SVM. (3) Implementation of optimized parameters for selected features and hyperparameter tuning for the LR model, resulting in an accuracy of 86%. Model generalization was assessed using random shuffled 10-fold cross-validation. (4) Critical analysis of error in major models, including model bias and variance. Additionally, we optimized the split between the training and validation sets and examined learning curves.

Literature Survey

- Preprocessing text data involves cleaning and formatting unstructured texts to improve the effectiveness of sentiment extraction (Pradha et al., 2019). The author presented a selection of preprocessing techniques, including lowercasing, removal of specific symbols, punctuations, stop words, hash bags, and white spaces, followed by tokenization and lemmatization/stemming. By comparing sentiments before preprocessing and manually preprocessed data trained by multiple ML models, SVM performed the best with 90.3% accuracy, indicating that preprocessing for sentiments improves accuracy and speed.

- Results from various studies (Laoh et al., 2019; Widiyaningtyas et al., 2019; Kaur et al., 2018) demonstrated the effective adoption of n-grams in sentiment classification for short text content like reviews or Twitter posts by many machine learning models, including SVM, KNN, NB, and LR. Research on movie review classification has shown a 1.908% improvement in SVM classifiers using a combination of unigrams and bigrams compared to unigrams alone, and a 2.352% increase in accuracy for the Naïve Bayes model with a combination of unigrams and bigrams compared to using unigrams only (Tripathy et al., 2016).

- In one paper (Dhanani et al., 2018), conceptualized comparisons were offered between traditional word embeddings, including one-hot encoding, bag-of-words, and new word embeddings like Word2Vec, which can be used for sentiment analysis utilizing vectors. In another paper (Katić & Milićević, 2018), authors not **only** provided traditional word embedding methods but also introduced bag-of-n-grams and their Term Frequency-Inverse Document Frequency (TF-IDF) transformation, Word2Vec, and TF-IDF Word2Vec embedding techniques. Comparisons were made between these techniques combined with different classifiers. This research claims that the Bag of n-grams features performed best with the Logistic Regression (LR) model, achieving 0.9231% accuracy compared to LSVM and MNB. On the other hand, a combination of bag-of-n-grams and TF-IDF performed best with SVM, achieving 0.929 accuracy compared to LR and NB. Word2Vec and Word2Vec+TF-IDF achieved only 0.8437 and 0.8222 accuracy, respectively, on the LR model. However, the Word2Vec embedding technique achieved an accuracy of 0.9381 on the Convolutional Neural Network (CNN).

- Feature selection techniques like chi-square improved NB classifiers' performance by 20% in accuracy for sentiment analysis (Nurhayati et al., 2019). In another paper (Setiyaningrum et al., 2019), feature selection increased the efficiency of classification time and cost in the KNN model while maintaining accuracy.

- As several researchers demonstrated (Katić & Milićević, 2018; Dey et al., 2020; Petrescu et al., 2019; Gupta et al., 2019), NB, SVM, and LR are commonly outperformed by simpler machine learning models in sentiment analysis. Other machine learning models like KNN and Decision Trees are sometimes adopted. Advanced models such as CNN, Long Short-Term Memory (LSTM), Random Forest (RF), and ensemble learning methods also excel in sentiment analysis.

Methodology

Preprocessing

To enable the extraction of features from the raw text data, it is essential to perform cleaning and formatting processes on the short text contents. These processes include lowercase conversion, space inference, removal of specific symbols and punctuations, elimination of stop words, handling hash bags, removing white spaces, and, finally, tokenization and lemmatization or stemming.

- ***Inferencing space:*** Some words are connected due to typos or formatting errors. For example, 'Thedinnerwasdelicious!' should have a space in the middle to separate words, like this: 'The dinner was delicious!'

- ***Lowercase text:*** Converting 'Sky' to 'sky' helps unify the text and reduces redundancy of the same word.

- ***Removal of symbols, punctuations, hash bags, and white spaces:*** These elements lack actual meaning as words and should be removed from the text.

- ***Removal of stop words:*** This process reduces common English words like 'a,' 'an,' 'this,' etc., which lack polarity but have a higher frequency and should not appear in the features.

- ***Tokenization:*** Sentences are separated into word strings and stored in lists, which prepares them for counting appearance frequency or behavior as basic features.

- ***Lemmatization:*** This step removes endings like 'ing' or 'ed,' changes 'pen(s)' and 'boxes' to 'box,' and converts 'better' to 'good.' It returns words to their base form to reduce form variations.

Feature engineer

n-grams

Tokenization was employed to obtain unigrams, bigrams, combinations of unigrams and bigrams, combinations of unigrams and trigrams, and combinations of unigrams with tetragrams (Setiyaningrum et al., 2019). The best-selected bag of words as features enhances the classifiers' performance. Examples of unigrams and bigrams include 'this is not bad' -> {'not'}, {'not', 'bad'}.

TF- IDF

Term Frequency and Inverse Document Frequency (TF-IDF):

Term Frequency (TF) = (count of the term) / (total word count in the document)

Inverse Document Frequency (IDF) = log (total number of documents) / (number of documents containing the keyword)

By multiplying TF by IDF, we obtain the weight of the frequency of a word or n-grams in a document, which adjusts for the number of documents containing the word or n-grams.

Word2vec and word2vec tf-idf

Word2vec is a different technique to convert words to vectors compared to traditional word embedding methods like bags of words and one hot-encoding. The encoded vectors retain the connectivity bond from word to word in a sentence. The size is consistent by pre-defining the size, allowing substantial text to be embedded effectively (Katić & Milićević, 2018).

Feature selection Chi-Square

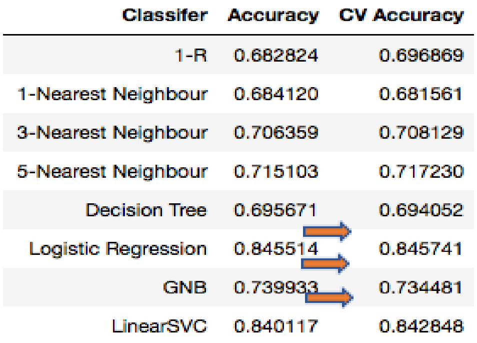
Chi-square is a feature selection method that reduces the dimensions of models and computational time by eliminating non-relevant or less relevant features, thereby enhancing classification performance (Setiyaningrum et al., 2019). The higher the Chi-Square value, the more dependent the feature is on the response..

Classifiers

The preliminary model selection is based on the research presented in the literature survey above, which involves a comparison of the most well-known sentiment analysis supervised machine learning models: Naive Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), Decision Tree (DT), and K-Nearest Neighbors (KNN). The One-R (1-R) classifier was treated as the baseline classifier.

Results evaluation and error analysis

Table 1: Comparisons of different Classifiers



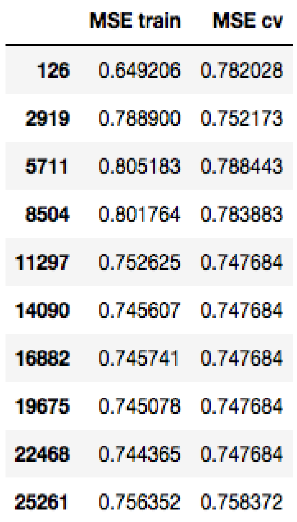
Classifer	Accuracy	CV Accuracy
1-R	0.682824	0.696869
1-Nearest Neighbour	0.684120	0.681561
3-Nearest Neighbour	0.706359	0.708129
5-Nearest Neighbour	0.715103	0.717230
Decision Tree	0.695671	0.694052
Logistic Regression	0.845514	0.845741
GNB	0.739933	0.734481
LinearSVC	0.840117	0.842848

From Table 1, the cleaned and formatted text data has been processed using unigram hot-encoding with chi-square feature selection on the top 1000 features, serving as a baseline for comparing various supervised machine learning models. In 10-fold cross-validation, LR (Logistic Regression), Linear SVC, and NB (Naive Bayes) achieve accuracy scores better than 0.846, 0.843, and 0.734, respectively.

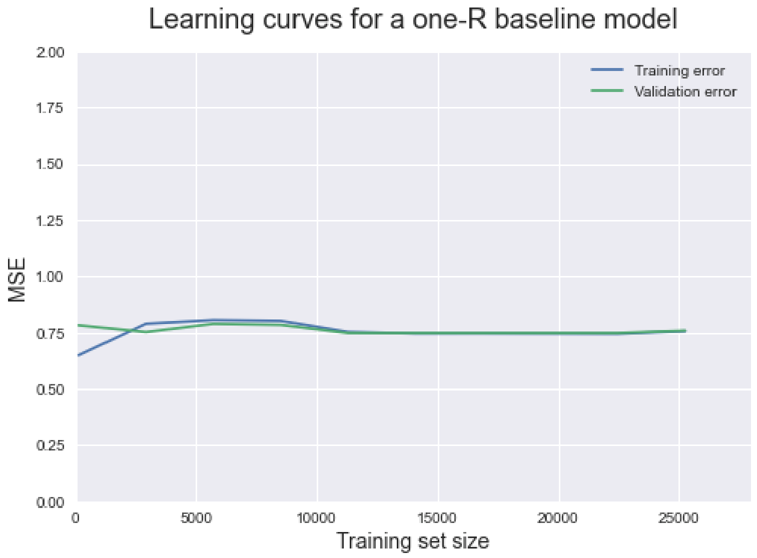
In Table 2 below, we compare the Mean Square Error (MSE) between "Train" and "10-fold cross-validation" for one-R and LR (Logistic Regression). It is evident from both cross-validation and training that the MSE for one-R remains relatively stable. This trend suggests that one-R maintains a consistently high MSE throughout the learning process, indicating that the model does not learn effectively from new instances and may not be suitable for the dataset.

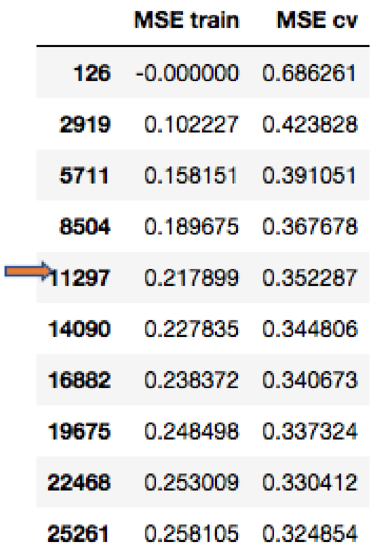
On the other hand, the MSE for LR decreases significantly during cross-validation, and after 11,297 instances, the MSE curve stabilizes. These trends suggest that the LR model has relatively low bias but slightly high variance. Possible solutions to address this include (1) adding more instances, (2) increasing the number of features, (3) changing the feature engineering methods, or (4) performing hyperparameter tuning to further reduce the variance and bias of the LR model.

Table2: LR learning curve with Mean square error (one-R and LR)



	MSE train	MSE cv
126	0.649206	0.782028
2919	0.788900	0.752173
5711	0.805183	0.788443
8504	0.801764	0.783883
11297	0.752625	0.747684
14090	0.745607	0.747684
16882	0.745741	0.747684
19675	0.745078	0.747684
22468	0.744365	0.747684
25261	0.756352	0.758372





	MSE train	MSE cv
126	-0.000000	0.686261
2919	0.102227	0.423828
5711	0.158151	0.391051
8504	0.189675	0.367678
11297	0.217899	0.352287
14090	0.227835	0.344806
16882	0.238372	0.340673
19675	0.248498	0.337324
22468	0.253009	0.330412
25261	0.258105	0.324854

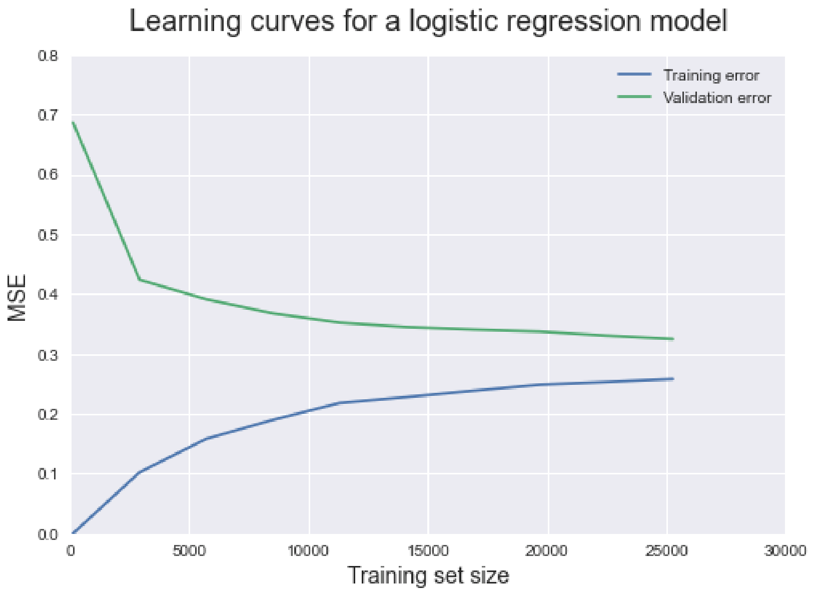
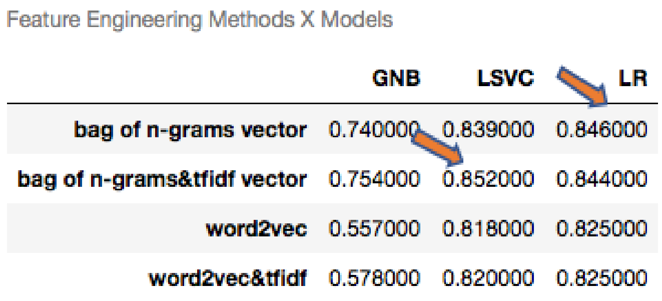


Table 3: (A) Feature engineering methods with top 3 different classifiers. (B,C) Top 2 Feature engineering methods with top2 classifiers.

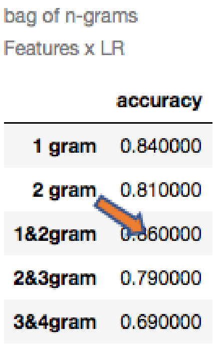
3A



Feature Engineering Methods X Models

	GNB	LSVC	LR
bag of n-grams vector	0.740000	0.839000	0.846000
bag of n-grams&tfidf vector	0.754000	0.852000	0.844000
word2vec	0.557000	0.818000	0.825000
word2vec&tfidf	0.578000	0.820000	0.825000

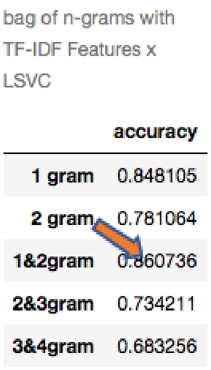
3B



bag of n-grams Features x LR

	accuracy
1 gram	0.840000
2 gram	0.810000
1&2gram	0.860000
2&3gram	0.790000
3&4gram	0.690000

3C



bag of n-grams with TF-IDF Features x LSVC

	accuracy
1 gram	0.848105
2 gram	0.781064
1&2gram	0.860736
2&3gram	0.734211
3&4gram	0.683256

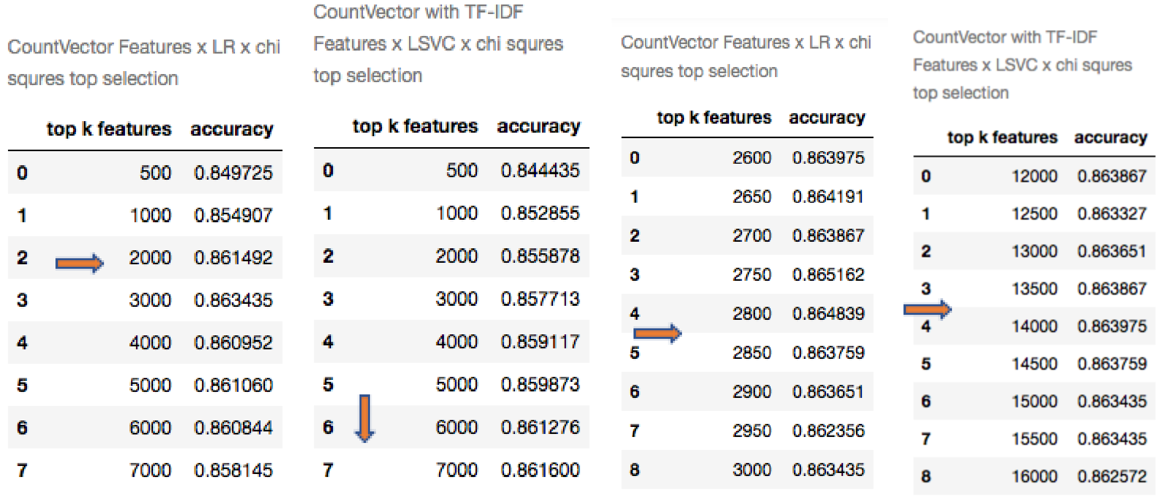
Table 3A cross-combinated with 4 different features methods : { bags of unigram; bag of unigram + tf-idf; word2vec; word2vec + tfidf } with top 3 classifiers: {GNB, LSVC, LR}
As results indicated in the table, Combination with a bag of unigram top 1000 vector+ TF-IDF with LSVC has the best score of 0.852% accuracy. And also, the bag of unigram top 1000 vector with 0.846% accuracy in combination with LR.

Table 3B By adjusting the numbers of n-grams to compare the performances of LR, The combination of 1&2 gram(s) is outperformed with 0.860 accuracy.

Table 3C By adjusting the numbers of n-gram + tf-idf to compare the performances of LSVC, The combination of 1&2 gram(s) is outperformed with 0.861 accuracy.

One potential explanation for this phenomenon could be attributed to a constraint limiting the analysis to a maximum of 1000 features. In this context, the inclusion of both unigrams and 2-grams may confer an advantage over utilizing solely bigrams, as the former not only encompass individual token representations but also capture bi-token combinations within the dataset. It is plausible that the combined 1- and 2-gram features contain unigrams that exhibit superior performance compared to using only bigrams within the restricted feature set. Conversely, it is conceivable that the converse is also true, where the inclusion of bigrams may enhance predictive capabilities beyond relying solely on unigrams when constrained to the 1000-feature limitation.

Table 4: LR Model performs best with 2,850 Features - Count vector with Chi square feature selection.



CountVector Features x LR x chi squres top selection

	top k features	accuracy
0	500	0.849725
1	1000	0.854907
2	2000	0.861492
3	3000	0.863435
4	4000	0.860952
5	5000	0.861060
6	6000	0.860844
7	7000	0.858145

CountVector with TF-IDF Features x LSVC x chi squres top selection

	top k features	accuracy
0	500	0.844435
1	1000	0.852855
2	2000	0.855878
3	3000	0.857713
4	4000	0.859117
5	5000	0.859873
6	6000	0.861276
7	7000	0.861600

CountVector Features x LR x chi squres top selection

	top k features	accuracy
0	2600	0.863975
1	2650	0.864191
2	2700	0.863867
3	2750	0.865162
4	2800	0.864839
5	2850	0.863759
6	2900	0.863651
7	2950	0.862356
8	3000	0.863435

CountVector with TF-IDF Features x LSVC x chi squres top selection

	top k features	accuracy
0	12000	0.863867
1	12500	0.863327
2	13000	0.863651
3	13500	0.863867
4	14000	0.863975
5	14500	0.863759
6	15000	0.863435
7	15500	0.863435
8	16000	0.862572

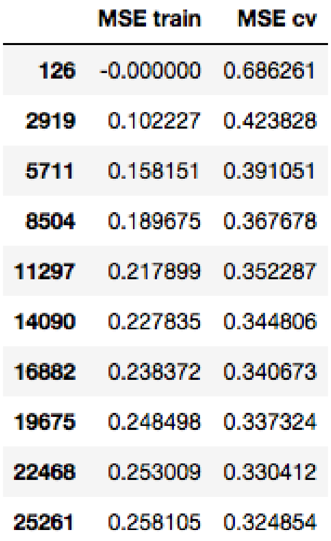
(Count Vector = bag of 1&2grams vector, Count Vector +tf -idf = = bag of 1&2grams vector+tf-idf)*

From Table 4 above, we selected the best top features that maximize the performance of LR and LSVC. The bag of 1&2-grams vector with the top 2,850 features achieved an accuracy of 0.864 for LR, while the bag of 1&2-grams vector combined with TF-IDF with the top 14,000 features achieved an accuracy of 0.864 for LSVC.

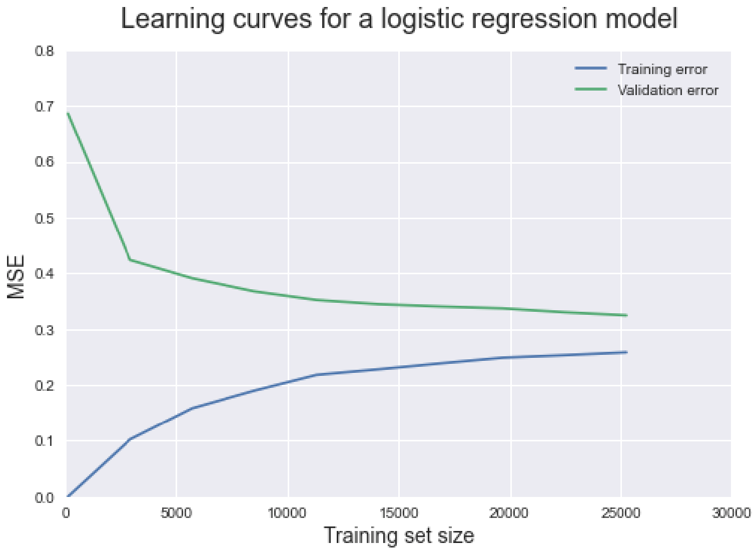
In Table 5 below, we compare the bag of 1&2-grams vector with the top 2,850 features for LR with the earlier baseline features used in LR. As we can see, the Mean Squared Error (MSE) in the selected features from training data reduced to 0.175 compared to the MSE in the baseline features, which was 0.258. Furthermore, the 10-fold cross-validation MSE dropped to 0.29 in the selected features from the baseline of 0.32. This reduction in MSE indicates a decrease in model bias in both the training and cross-validation datasets. However, it's worth noting that the variance of the model increased by 0.04 compared to the baseline, suggesting that simply adding more instances to the training data may not necessarily lead to better-performing models. To improve the current models, it is essential to extract more features from the data or explore more powerful feature engineering methods

Table 5

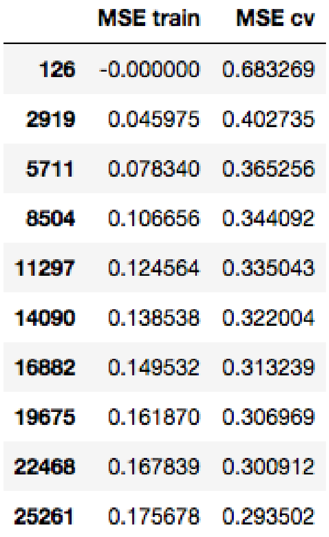
(A) Baseline features with LR



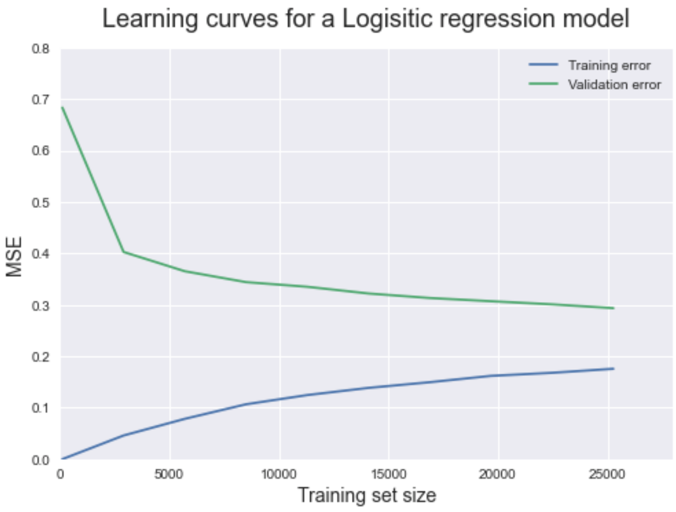
	MSE train	MSE cv
126	-0.000000	0.686261
2919	0.102227	0.423828
5711	0.158151	0.391051
8504	0.189675	0.367678
11297	0.217899	0.352287
14090	0.227835	0.344806
16882	0.238372	0.340673
19675	0.248498	0.337324
22468	0.253009	0.330412
25261	0.258105	0.324854



(B)Bag of 1&2grams vector - top 2850 features with LR



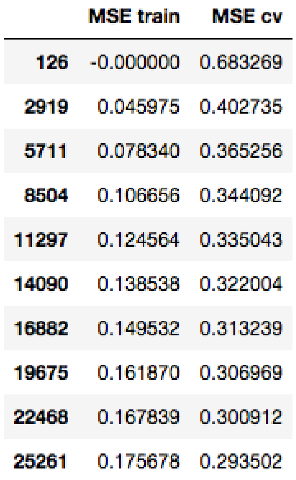
	MSE train	MSE cv
126	-0.000000	0.683269
2919	0.045975	0.402735
5711	0.078340	0.365256
8504	0.106656	0.344092
11297	0.124564	0.335043
14090	0.138538	0.322004
16882	0.149532	0.313239
19675	0.161870	0.306969
22468	0.167839	0.300912
25261	0.175678	0.293502



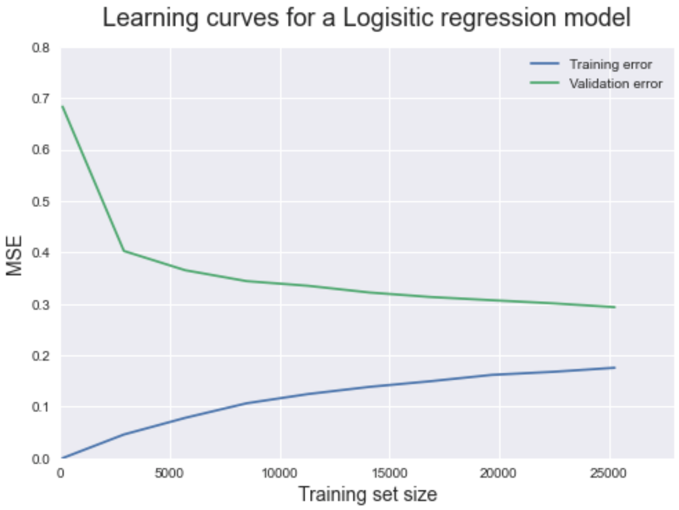
In Tables 6 below, we compare the two best combinations of classification systems. Upon observing the Mean Squared Error (MSE) in both 6(A) and 6(B), it becomes evident that both models exhibit similar bias and variance characteristics. However, in 6(B), performance can be enhanced by incorporating additional features into the models. It's worth noting that augmenting the dataset with more data carries the risk of increasing bias. Consequently, striking a balance between computational resources and the slight performance improvement is a crucial consideration.Similarly, applying the same solution as in 6(A), wherein we compare the top feature sets, with 6(B) containing 14,000 features while 6(A) utilizes only 2,850, indicates that exploring alternative feature extraction methods might further enhance performance.

Table 6 displays the top feature sets, plotted with 2,850 and 14,000 features, using LR and LSVC.

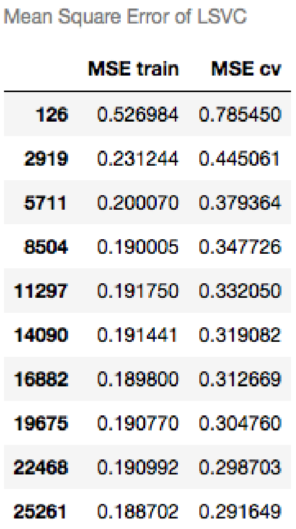
(6A) MSE: bag of 1&2grams vector - top 2850 features with LR



	MSE train	MSE cv
126	-0.000000	0.683269
2919	0.045975	0.402735
5711	0.078340	0.365256
8504	0.106656	0.344092
11297	0.124564	0.335043
14090	0.138538	0.322004
16882	0.149532	0.313239
19675	0.161870	0.306969
22468	0.167839	0.300912
25261	0.175678	0.293502

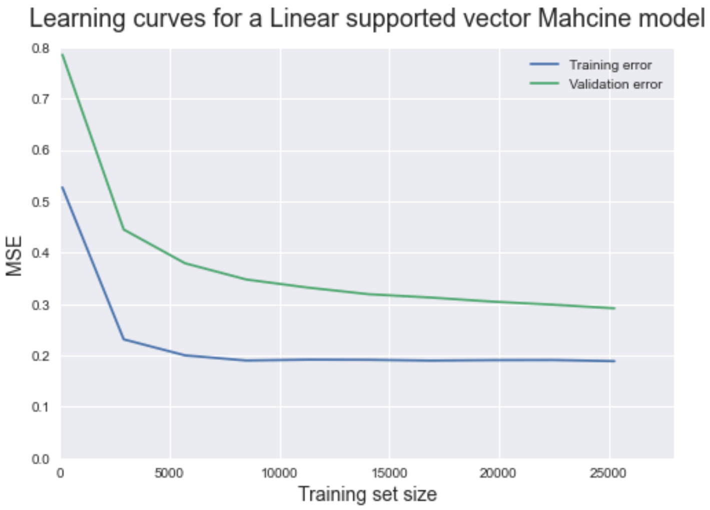


(6B) MSE: bag of 1&2grams vector + tf-idf : top 14000 features with LSVC



Mean Square Error of LSVC

	MSE train	MSE cv
126	0.526984	0.785450
2919	0.231244	0.445061
5711	0.200070	0.379364
8504	0.190005	0.347726
11297	0.191750	0.332050
14090	0.191441	0.319082
16882	0.189800	0.312669
19675	0.190770	0.304760
22468	0.190992	0.298703
25261	0.188702	0.291649

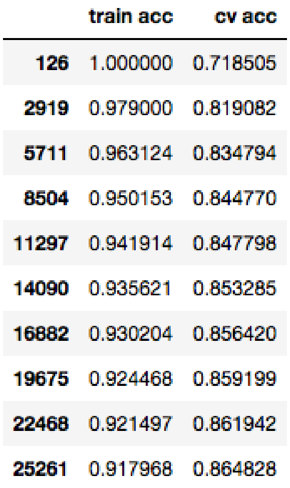


From the Figure 6(D) below, similar to Figure 6(B), as the training data increase, the 10-fold CV accuracy slowly trends upward, while the training accuracy remains stabilized. The tradeoff between computational resources and the marginal improvement in performance becomes apparent once again.

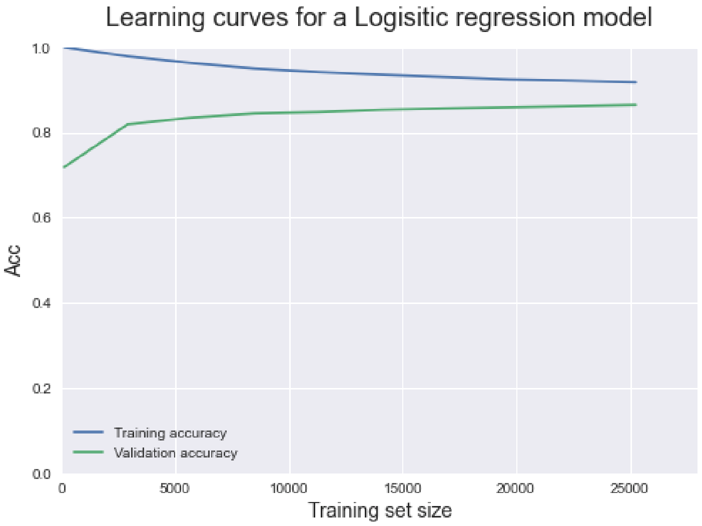
In Figure 6(C), both the training and 10-fold CV accuracy tend to converge as the training data size increases, indicating that the better solution to improve performance is by adding new features or more features rather than increasing the number of training instances.

Regarding splitting sets: The learning curve also shows that the 10-fold CV accuracy keeps increasing. This suggests that using larger portions of the training data for both systems will yield better results whenever possible. For example, using a split like 90% for training and 10% for testing.

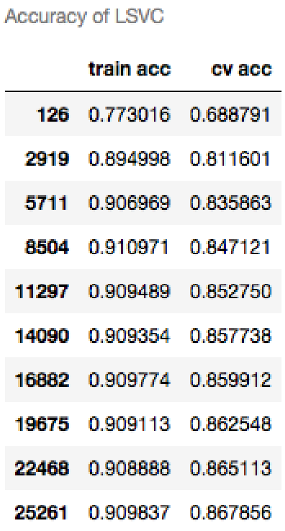
(6C) MSE: bag of 1&2grams vector - top 2850 features with LR



	train acc	cv acc
126	1.000000	0.718505
2919	0.979000	0.819082
5711	0.963124	0.834794
8504	0.950153	0.844770
11297	0.941914	0.847798
14090	0.935621	0.853285
16882	0.930204	0.856420
19675	0.924468	0.859199
22468	0.921497	0.861942
25261	0.917968	0.864828

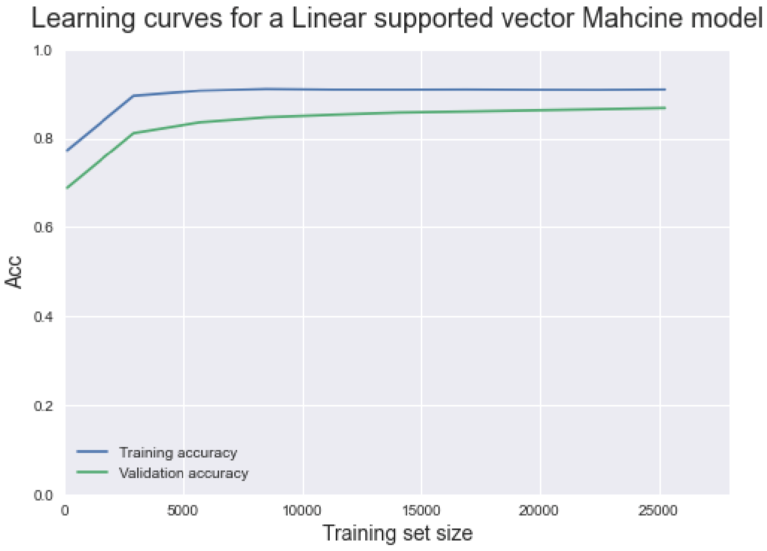


(6D) MSE: bag of 1&2grams vector + tf-idf : top 14000 features with LSVC



Accuracy of LSVC

	train acc	cv acc
126	0.773016	0.688791
2919	0.894998	0.811601
5711	0.906969	0.835863
8504	0.910971	0.847121
11297	0.909489	0.852750
14090	0.909354	0.857738
16882	0.909774	0.859912
19675	0.909113	0.862548
22468	0.908888	0.865113
25261	0.909837	0.867856



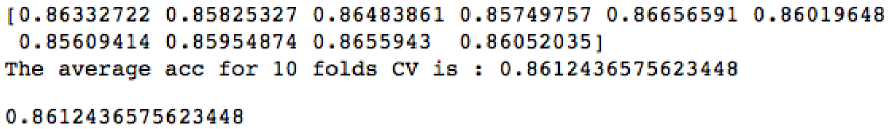
Hyperparameter tuning for logistic Regression models

Table:7

7(A)

LR before hyperparameters tuning 10 folds CV acc = 0.859872

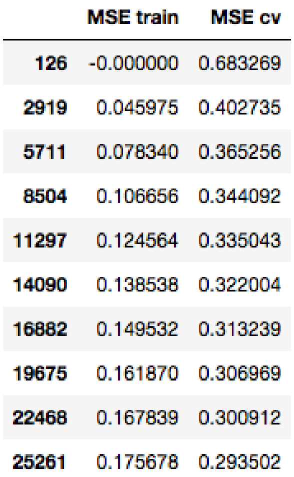
LR with hyperparameters tunning 10 folds CV acc = 0.861243



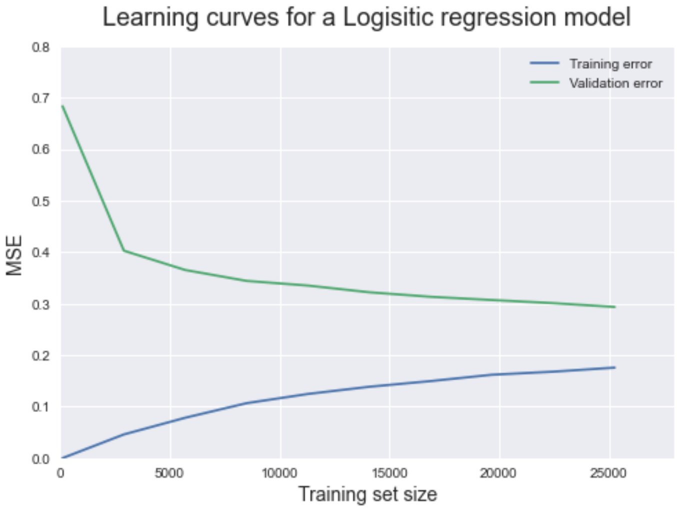
```
[0.86332722 0.85825327 0.86483861 0.85749757 0.86656591 0.86019648
 0.85609414 0.85954874 0.8655943  0.86052035]
The average acc for 10 folds CV is : 0.8612436575623448

0.8612436575623448
```

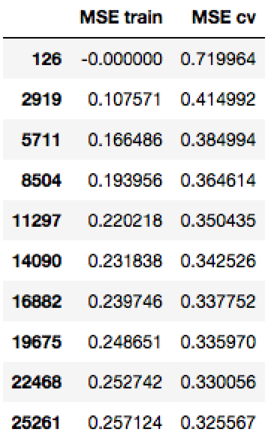
7(B) MSE learning curve of LR before hyper parameter tuning



	MSE train	MSE cv
126	-0.000000	0.683269
2919	0.045975	0.402735
5711	0.078340	0.365256
8504	0.106656	0.344092
11297	0.124564	0.335043
14090	0.138538	0.322004
16882	0.149532	0.313239
19675	0.161870	0.306969
22468	0.167839	0.300912
25261	0.175678	0.293502



7(C) MSE learning curve of LR with Hyperparameters tuning



	MSE train	MSE cv
126	-0.000000	0.719964
2919	0.107571	0.414992
5711	0.166486	0.384994
8504	0.193956	0.364614
11297	0.220218	0.350435
14090	0.231838	0.342526
16882	0.239746	0.337752
19675	0.248651	0.335970
22468	0.252742	0.330056
25261	0.257124	0.325567

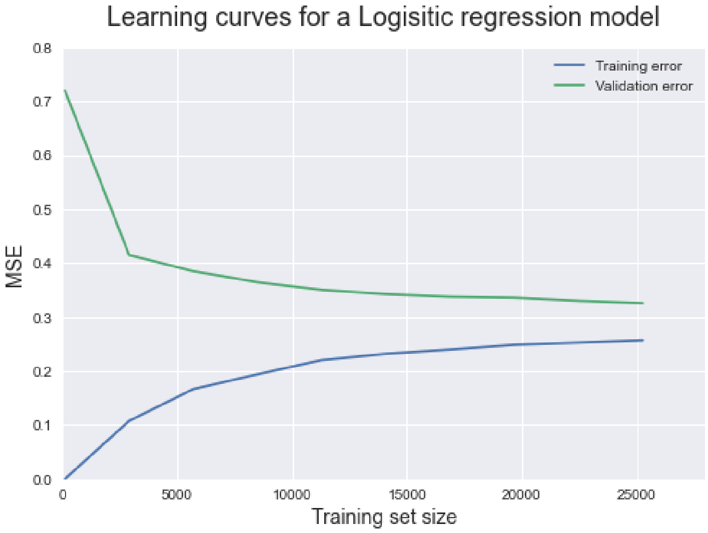
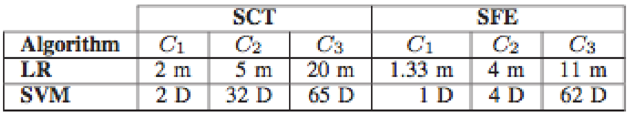


Table 7(A) presents the hyperparameter tuning results for the LR model, showing only a slight increase of 0.14% in 10-fold cross-validation accuracy. Tables 7(B) and 7(C) display figures illustrating the cross-validation mean squared error (CV MSE) scores for the LR model with hyperparameters, which increased by approximately 0.008. This suggests a slight increase in the model's bias. However, the variances for the hyperparameter-tuned LR model decreased by approximately 0.019. These results indicate a trade-off between the model's bias and variance. The LR model with hyperparameters reduced variances while slightly improving its generalization ability to the test set. However, this comes with the risk of increasing errors in both the training and test sets by simplifying the model to some extent.

The computational time required for 10-fold cross-validation with the best parameters, {'l1_ratio': 0.2, 'max_iter': 1000, 'penalty': 'elasticnet'}, consumed several hours to obtain the results. The decision to adopt these hyperparameters in LR depends on the available computational power and time resources. Simultaneously, one must consider the optimization of the model's bias and variance.

Regarding the LinearSVC model, as noted by researchers (Petrescu et al., 2019), the improvement achieved through hyperparameter tuning with LSVC is minimal when applied to sentiment analysis conducted on social network content. The authors argued that LR is much more efficient than SVM, as shown in Table VII. While there is a slight improvement in LSVC evaluation compared to LR, the time required to build the LSVC model is not feasible. Several experiments were conducted to fine-tune the hyperparameters of LinearSVC, but errors arose, indicating either insufficient memory or segmentation faults after running overnight with the top 14,000 features. It is important to note that a trial run with 1,000 instances did yield the best hyperparameter results.

Table VII: SVM vs. LR runtime comparison



	SCT			SFE		
Algorithm	C_1	C_2	C_3	C_1	C_2	C_3
LR	2 m	5 m	20 m	1.33 m	4 m	11 m
SVM	2 D	32 D	65 D	1 D	4 D	62 D

(Petrescu et al., 2019)

Conclusion

This report attempts to address sentiment analysis on raw restaurant review text. The results from this report are consistent with the literature surveys, including the finding that the combination of unigrams and bigrams as features in bags of words improves the performance of both Logistic Regression (LR) and Linear Support Vector Classification (LSVC). The bag of 1&2-grams vectors performs best with the Logistic Regression model, achieving an accuracy value of 0.861243. Additionally, the bag of 1&2-grams vectors combined with TF-IDF performs best with the LinearSVC (SVM) model, achieving an accuracy of 0.863. The Chi-square feature selection method slightly improves performance by adjusting the number of selected features (k features). Hyperparameter tuning also results in a slight accuracy improvement of 0.14%.

Simpler models like Logistic Regression and Linear SVM outperform more resource-intensive models, requiring fewer computing resources.

In future work, we will compare the trade-off between expense and performance of models such as CNN, LSTM, and ensemble learning to evaluate their potential as classification systems compared to simpler traditional models. We will also explore other feature engineering methods, such as the lexicon approach, to further enhance our analysis

Reference:

Arjun Mukherjee, Vivek Venkataraman, Bing Liu, & Natalie Glance. (2013). What Yelp Fake Review Filter Might Be Doing? *International AAAI Conference on Web and Social Media; Seventh International AAAI Conference on Weblogs and Social Media*. https://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/view/6006

Bhavitha, B. K., Rodrigues, A. P., & Chiplunkar, N. N. (2017). Comparative study of machine learning techniques in sentimental analysis. *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 216–221. https://doi.org/10.1109/ICICCT.2017.7975191

Dey, S., Wasif, S., Tonmoy, D. S., Sultana, S., Sarkar, J., & Dey, M. (2020). A Comparative Study of Support Vector Machine and Naive Bayes Classifier for Sentiment Analysis on Amazon Product Reviews. *2020 International Conference on Contemporary Computing and Applications (IC3A)*, 217–220. https://doi.org/10.1109/IC3A48958.2020.233300

Dhanani, J., Mehta, R., Rana, D., & Tidke, B. (2018). Sentiment Analysis using Novel Distributed Word Embedding for Movie Reviews. *2018 Tenth International Conference on Advanced Computing (ICoAC)*, 138–145. https://doi.org/10.1109/ICoAC44903.2018.8939104

Gupta, A., Singh, A., Pandita, I., & Parashar, H. (2019). Sentiment Analysis of Twitter Posts using Machine Learning Algorithms. *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, 980–983.

Katić, T., & Milićević, N. (2018). Comparing Sentiment Analysis and Document Representation Methods of Amazon Reviews. *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, 000283–000286. https://doi.org/10.1109/SISY.2018.8524814

Kaur, S., Sikka, G., & Awasthi, L. K. (2018). Sentiment Analysis Approach Based on N-gram and KNN Classifier. *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 1–4. https://doi.org/10.1109/ICSCCC.2018.8703350

Laoh, E., Surjandari, I., & Prabaningtyas, N. I. (2019). Enhancing Hospitality Sentiment Reviews Analysis Performance using SVM N-Grams Method. *2019 16th International Conference on Service Systems and Service Management (ICSSSM)*, 1–5. https://doi.org/10.1109/ICSSSM.2019.8887662

Nurhayati, Putra, A. E., Wardhani, L. K., & Busman. (2019). Chi-Square Feature Selection Effect On Naive Bayes Classifier Algorithm Performance For Sentiment Analysis Document. *2019 7th International Conference on Cyber and IT Service Management (CITSM)*, *7*, 1–7. https://doi.org/10.1109/CITSM47753.2019.8965332

Petrescu, A., Truică, C.-O., & Apostol, E.-S. (2019). Sentiment Analysis of Events in Social Media. *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 143–149. https://doi.org/10.1109/ICCP48234.2019.8959677

Pradha, S., Halgamuge, M. N., & Tran Quoc Vinh, N. (2019). Effective Text Data Preprocessing Technique for Sentiment Analysis in Social Media Data. *2019 11th*

International Conference on Knowledge and Systems Engineering (KSE), 1–8. https://doi.org/10.1109/KSE.2019.8919368

Rayana, S., & Akoglu, L. (2015). Collective Opinion Spam Detection: Bridging Review Networks and Metadata. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 985–994. https://doi.org/10.1145/2783258.2783370

Setiyaningrum, Y. D., Herdajanti, A. F., Supriyanto, C., & Muljono. (2019). Classification of Twitter Contents using Chi-Square and K-Nearest Neighbour Algorithm. *2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 1–4. https://doi.org/10.1109/ISEMANTIC.2019.8884290

Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems With Applications*, *57*, 117–126. https://doi.org/10.1016/j.eswa.2016.03.028

Widiyaningtyas, T., Elbaith Zaeni, I. A., & Farisi, R. A. (2019). Sentiment Analysis Of Hotel Review Using N-Gram And Naive Bayes Methods. *2019 Fourth International Conference on Informatics and Computing (ICIC)*, 1–5. https://doi.org/10.1109/ICIC47613.2019.8985946