Vrije Universiteit Amsterdam

Movie Review Sentiment Analysis

Yasin Aydin (2657725)
İlayda Tukuş (2657227)
Thijs Roozen (2652254)
Humam Syauqi Dawa (2657598)
Xerxes Koehoorn (2528404)

January 8, 2020

Abstract

The aim of this research is to do sentiment analysis by training five different classification models on positive and negative reviews. The five models are Random Forest, Binary Logistic Regression, Naive Bayes, Support Vector Machines, and Naive Bayes in combination with Support Vector Machine. These models were trained and tested using Python and different libraries, where appropriate. The result of this research shows that the combined model (NBSVM) is the most accurate model on our dataset.

1 Introduction

"It has been estimated that as much as 95% of data is unstructured text, images, audio or video" (Bird, Klein, and Loper, 2009), which indicates that it is important to explore methods of making sense of unstructured data. In this report, Sentiment Analysis is used to try to predict the outcome of movie and product reviews. The dataset contains a large amount of Amazon movie and TV reviews and their corresponding rating in a number of stars with irrelevant information that will be weeded out. In this research, classification is chosen over regression since we want our results to reflect the belongingness of reviews in our dataset to certain explicit categories: "positive" and "negative" reviews. Using the Random Forest (RF), Binary Logistic Regression (BLR), Naive Bayes (NB), Support Vector Machines (SVM) classifiers, and the more recently developed combined Naive Bayes and Support Vector Machine (NBSVM) classifier, we will try to find the optimal way to predict whether a movie review-text belongs to a positive or negative assessment of the movie.

1.1 Research question

Wang and Manning (2015) claim that the usefulness of bigram features in a bag-of-features sentiment classification has been underappreciated. In their research, they emphasized the usefulness of the bigrams a lot since the accuracies were higher with the bigrams. This can be also explained in the way English grammar works. To give a small example, in the review of a customer the word 'good' can be mentioned a lot, but if it has 'not' in front of it, the meaning of 'good' is modulated. For that reason, our research question is to find out whether or not the added context-sensitivity of bigram analysis (grouping words in pairs) produces more reliable results than that of unigram analysis.
The amount of words words and bigram in the reviews will result in a very large feature space. We hypothesize that the model that can handle this best for our dataset is the combination of Naive Bayes and Support Vector Machines (NBSVM). As Wang and Manning (2015) concluded, a combination of these models can increase the accuracy of sentiment analysis. So, we will also check if combinations of models increase the robustness the models.

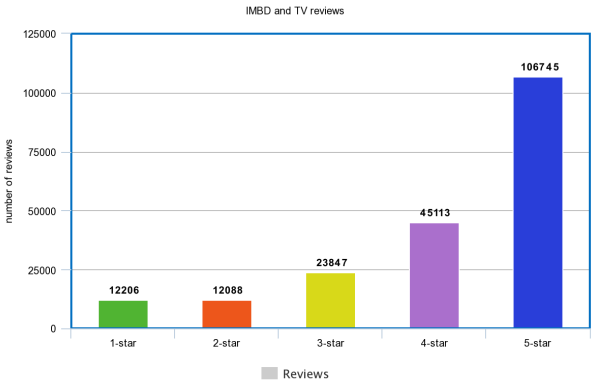

Figure 1: Histogram of 200k dataset review scores

2 Data inspection and preparation

For this research, a dataset of movie and TV Amazon reviews was obtained(McAuley, 2015). With the permission of Julian McAuley, we were granted access to this dataset and could be used for the analysis. The data consists of 9 columns and around 200.000 rows. Everything except the review text and the ratings was discarded. After that, the distribution of the number of reviews per star in our dataset was investigated *(Figure 1)*.

Seeing this, one can conclude that this is a pretty imbalanced dataset. To fix this, we have done the following:

1. Equalizing the number of reviews per star

   Most of the reviews were positive, with an average review-rating of 4.11/5. To make sure our classifiers had a balanced dataset to work with (as to not propagate the positively-biased nature of the scores in our original dataset), we used an undersampling method and balanced the data by having the same amount of reviews for every

rating. In practice, this means that the number of reviews that rated 2 to 5 stars had to be limited to the minimally-represented class of 2 stars with 12.088 reviews.

2. Removing all neutral reviews

   Subsequently we created a dataset from this balanced dataset where all 'neutral' (2-3-4-stars) reviews were removed, and have taken the 1-star reviews as negative and the 5-star reviews as positive reviews. The reasoning behind doing this is that the reviews that have 2-3-4 star ratings are assumed to contain less information pertinent to positive/negative classification.

2.1 Cleaning

A Regex function has been used to get rid of all interpunction, leaving just alphabetical characters. Subsequently, everything was lower-cased.
Table 1 contains the most positive and negative words, according to our analysis.
All words have been stemmed with 'Porter stemming'. Stemming means words are reduced to the part of the word that all grammatical variants of that word have in common (e.g. the stem of terribly is terribl, which is also the stem of terrible). This is very useful for our purpose because this decreases the size of the vocabulary for our bag-of-words (BoW) vectors: without stemming, the two grammatical variants of a word would all have a different entry in our vocabulary. A small vector, which contains the same amount of information (assuming none gets lost through stemming), means that the information density of our feature vectors increases as a result of stemming.

Positive	Rating	Negative	Rating
Excel(-lent)	1.29	Wast(-e)	-2.05
Great	1.27	Worst	-1.77
Favorit(-e)	1.13	Bore	-1.68
Perfect	1.11	Aw(-ful)	-1.50
Love	1.07	Terribl(-e)	-1.41

Table 1: Most common positive and negative word stems

2.2 Feature extraction

After preprocessing, three ways to represent review-text were used as input for the different classifiers: a) a regular 'Bag of Words' model where only the occurrence of a word in a document is noted with a binary variable. b) a BoW model that also keeps track on the frequency per word in a document. c) a BoW model that uses two consecutive words (bi-grams) instead of singular words. With these three kinds of feature extractions, one can use one of them for specific machine learning methods we are going to use.

2.3 Using different representations

"Because not every word in a review is equally important to determine the ranking, applying the 'Term Frequency-Inverse Data Frequency' (TF-IDF) principle to the documents is vital" (Droidhead, 2018). The TF-IDF method can be split into two parts. Firstly, the TF gives us the frequency of each word in a document. Secondly, IDF gives us a weight for rare words in all documents of the corpus. Multiplying these two variables gives a TF-IDF score which implies how important a word is. Also, binary representation will be tested. For instance, in binary logistic regression, each review is represented as a binary vector with a column for every unique word in the corpus, where '1' represents that a given word just exists in the review itself. Having either frequency or binary representations can have different effects on the models. These effects will be compared between models.

2.4 Creating a test set

The dataset that was extracted from the Amazon movie and TV reviews was first randomized, and then the first 75% of the randomized dataset was used for training and validation, and the remaining 25% percentage of the dataset was used as a test set for the end results of this research.

2.5 Implementation

The goal of the research is to eventually determine whether the reviews of Amazon's movie and TV products are positive or negative with the best accuracy possible. The accuracy will be calculated with the true positive and true negative results divided by the number of instances. This is an intuitive measure to use when dealing with binary outcomes. During the implementation part, for each model, different parameters are being tested so that better accuracy could be acquired. Furthermore, other than changing different model parameters, different word representations are used for the reviews as described. These representations work by encoding more information into the vector. As explained, these variables are tested in many ways to get better accuracy on the validation data so that the final result can be optimal. A critical note: In order to use bigrams and hold them in memory, a big amount of memory is needed. In the implementation of this model, some memory errors were faced and handled by lowering the size of data.

3 Models

All of the inputs are based on the BoW model of documents: documents are represented as vectors of word-occurrences and (TF-IDF) frequencies. The BoW model disregards grammar and word-order, although when using bigrams some of these properties are taken into account implicitly.

3.1 Model 1: Random Forest

Breiman (2001) introduced random forests as a robust technique of classification and regression machine learning. A random forest is a set of decision trees where each tree is built from a sample of input data from a bootstrap. In addition, each node in a tree (i.e., each decision) is based on a random subset of features available. By introducing randomness at both data level and model level, random forests have been shown to yield accurate and robust results. Using training and test data from Amazon of product reviews, consisting of 1000 instances, half of which are assigned to positive or negative, Al Amrani, Lazaar, & El Kadirp (2018) found that, 810 instances, among 1000, are correctly classified—giving an accuracy rate of 81%. Furthermore, by employing random forest for Twitter Sentiment Corpus Dataset (TSCD) and Stanford Dataset (SFD) in BoW class of bigram, BoW with random forest yields accuracy rates of 79.24% and 66.57% for TSCD and SFD, respectively (Singh, Tomar, & Sangaiah, 2018). The random forest classifier works by building many decision trees and outputting the class that appears most often. This circumvents one of the main problems that normal (single) decision tree classifiers have, which is overfitting.

For b = 1, ..., B:

1. Sample, with replacement, n training examples from X, Y; call these X_b, Y_b.
2. Train a classification or regression tree f_b on X_b, Y_b.

After training, predictions for unseen samples x can be made by averaging the predictions from all the individual regression trees on x:

$$f(x) = \frac{1}{B}\sum_{b=1}^{B} f_b(x)$$

We implemented this model with 25 trees. Higher tree number would give us better training results but the model would be overfitting. And lower tree numbers were giving too low accuracies so we decided to use 25 tree.

3.2 Binary Logistic Regression

Binary logistic regression (BLR) is also known as logit regression, maximum-entropy classification or the log-linear classifier. It is a classification algorithm that learns a model for dichotomous (binary) classification. In this model, the probabilities that are describing the possible outcomes of a single trial are modeled by using a logistic function (sigmoid curve L=1,k=1,x=0 , Figure 2) with the equation:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

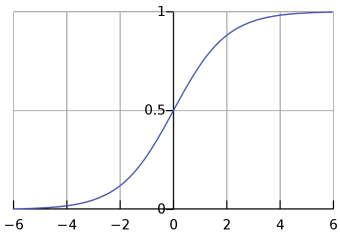

Figure 2: Sigmoid funciton

"Odds of an event are the ratio of probability that an event will occur to the probability that it will not occur."(Park, 2013). So, if the probability of an event occurring is p, the corresponding odds is a value given by $Odds\,of\ \{event\} = \frac{p}{1-p}$

"Since logistic regression calculates the probability of an event occurring over the probability of an event not occurring, the impact of independent variables is usually explained in terms of odds."(Park). The simple logistic model equation is:

$$logit(y) = ln(odds) = ln(\frac{p}{1-p}) = \alpha + \beta^T X$$

The model is used for predicting discrete values like 0/1, yes/no, true/false with the given set of independent variables. According to Park (2013), "Binary logistic regression is typically used when the dependent variable is dichotomous and the independent variables are either continuous or categorical." So, in the model weighted (by Beta) binary (word-presence) / tf-idf (word-frequency) vectors are used as argument for the sigmoid function to determine the probability of a review belonging to the positive (1) or negative (0) class.

In order to decide the optimal accuracy for this classification model on the training set, different C (inverse of regularization strength) values were sampled and the optimal result is found at the C=0.15 value. Even though the classification algorithm is only using binary discrete values, the cross-validation results were around 89.48 range (+/-1.34). The disadvantage of this model is that the number of repeated words in the same review cannot have more influence over classification so that the accuracy could still be improved. Hence, the same model was tested with frequency representation to see if accuracy gets better.

3.3 Model 3: Naive Bayesian

Naive Bayes (NB) classifiers work by assuming that all of the features are independent of each other. This allows for easy calculation of the maximum likelihood. NB is a probabilistic classifier that outputs the class for which a given set of features (a stemmed review, in our case) has the highest probability of belonging to (McCallum & Nigam, 1998). In this research, a multinomial naive Bayes classifier has been used In multinomial NB Classifier case, the likelihood for a given document (frequency vector) appearing in a certain class C_k is determined using the following (Dai, Xue, Yang and Yu (2007):

$$p(\mathbf{x}\,|rating\,) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p_{ki}^{x_i}$$

Where C_k is the class k. However, we are more interested in the probability of a frequency-vector being in a certain class k:

$$p(\,rating\mid \mathbf{x}) \propto log\left(p(rating) * \prod_{i=1}^{n} p_{ki}^{x_i}\right) =$$

$$log\,(p(rating)) + \sum x_i \cdot log(p_{ki}) = \mathbf{b} + \mathbf{w}_k^T \mathbf{x}$$

Taking the log of the likelihood allows for linearization and easier optimization.
There are two important considerations for Naive Bayes when it comes to model-selection:

The first is that a Bernoulli Naive Bayes model has to be used when dealing with binary word-presence vectors, and a Multinomial Naive Bayes model has to be used when using word-frequency vectors.

Furthermore, the smoothing parameter can be chosen. We chose the default (Laplace) smoothing parameter of $\alpha = 1$. Changing this parameter did not give us consistently better performance on the training set.

3.4 Model 4: Support Vector Machine

The basic idea behind the SVM classifier in binary classification is to find a hyperplane, represented by vector w, which not only separates the document vectors in one class from those in the other but for which the separation or margin is as large as possible(Pang, Lee, and Vaithyanathan, 2002, s.82). Although data may not always be easy to separate linearly, SVM will nevertheless be able to find a separating hyperplane by adjusting the value C (Marafino & Davies, 2014). A large value of C means that the margin of the boundaries of the support vectors are closer and the penalty will be given to misclassified data. The other way around, a smaller value of C is increasing the width between the support vectors and giving a less big penalty to misclassified data. In this research, one is dealing with a large vector of possible words in the corpus. Each review has such a vector with the frequency of the number of times a word is mentioned in a review. Although, at first sight, this may not be easy linearly separable, SVM can still produce a fine job because of the above-described algorithm and hyperparameters. To maximize the margin, we need to:

$$minimize : \frac{1}{2} |w|^2$$

$$subject\ to : \ y_i(W^T X_i + b) - 1 \geq 0, : \forall i$$

$$x^{(k)} = \hat{f}^{(k)}$$

which needs to be maximized with respect to α. This is a constrained convex optimization problem with a quadratic objective function and linear constraints. We implicitly assumed that the data is linearly separable, that is, there is a hyperplane which correctly classifies the training data. There have been researches conducted on sentiment analysis using SVM. Using Indonesian online store reviews of Bukalapak, Lutfi et al. (2018) made use of a dataset consisting of 1521 negative reviews and 1656 positive reviews, which then become 3077 reviews altogether after preprocessing. Lutfi et al. (2018) also applied the 10-fold cross-validation technique in order to get 10 iterations for each experiment. The main finding is that SVM provides higher accuracy compared to that of Nave Bayes (NB).

3.5 Model 5: Naive Bayes and Support Vector Machine

Naive Bayes (NB) and Support Vector Machine (SVM) models are often used as baselines for other methods in text categorization and sentiment analysis research (Wang & Manning, 2015). This is the model that combines both known algorithms to get better results in the classification. Wang and Manning (2015) show that NB outperforms SVM on shorter text-lengths, however, they emphasized that SVM is already good at full-length reviews. In this research, the dataset contains lots of long reviews with some short ones. So, in order to increase accuracy, a combination of these models could be used. Also, both validation results of this research and Wang and Manning's results showed that using bigrams increase accuracy and yields more consistent results. The NBSVM model can also deal with bigrams so these factors increase the chance of getting better final results for sentiment analysis. In addition to that, they concluded that when there is a simple model variant where an SVM is built over NB log-count ratios as feature values, the new model can be more strong and robust. So, the new model can perform better for both short and long reviews. Whereas the SVM uses $x^{(k)} = \hat{f}^{(k)}$, NBSVM uses $x^{(k)} = \tilde{f}^{(k)}$,where $\tilde{f}^{(k)} = \hat{r} \circ \hat{f}^{(k)}$. ($\circ$ is the element-wise product) As the model suggested in (Wang & Manning, 2015), the selected parameters are as follows: alpha= 1, C = 1, beta= 0.25 for NBSVM, and C = 0.1 for SVM. Furthermore, the 10-fold cross-validation and standard train/test split with the ratio of 75/25 has been used as they did. The accuracy rates they obtained are 90.41% (1.86) and 90.92% (1.13) for the NBSVM-uni and NBSVM-bi cross validations, respectively. As expected, the bigram model has a higher rate of accuracy and a lower standard deviation than that of unigram.

4 Results

Table 2 and *Table 3* show the accuracy from the different models, using uni- or bigrams and split between cross-validated and final results.

4.1 Model Accuracies

Model	C.V. Results	Test Results
RF-uni	85.21 (±1.30)	85.14
BLR-uni	89.48 (±1.34)	89.64
BLR-bi	89.90 (±1.38)	**90.44**
NB-uni	89.37 (±1.60)	88.40
NB-bi	89.90 (±1.03)	89.33
SVM-uni	89.66 (±1.45)	90.07
SVM-bi	**90.18** (±1.69)	**90.37**
NBSVM-uni	**90.41** (±1.86)	90.10
NBSVM-bi	**90.92** (±1.13)	**91.03**

Table 2: 10-fold cross-validation and results with the presence of words (%). Errors indicate standard deviation. Top 3 methods are in bold and the best is also underlined.

Model	C.V. Results	Test Results
RF-uni	85.51 (±1.92)	86.27
BLR-uni	**89.41** (±1.51)	88.07
BLR-bi	87.14 (±1.63)	**89.58**
NB-uni	88.24 (±1.35)	87.97
NB-bi	87.16 (±1.69	88.77
SVM-uni	88.40 (±2.09	88.80
SVM-bi	86.90 (±1.54)	87.62
NBSVM-uni	**90.96** (±0.99)	**90.04**
NBSVM-bi	**90.81** (±1.12)	**90.73**

Table 3: : 10-fold cross-validation and results with frequencies of words (%). Errors indicate standard deviation. Top 3 methods are in bold and the best is also underlined.

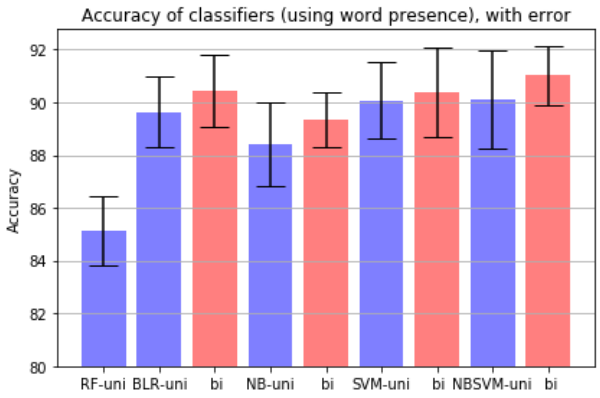

Figure 3: Accuracy of classifiers using word-presence on the training set with error rates(%). Error bars indicate standard deviation.

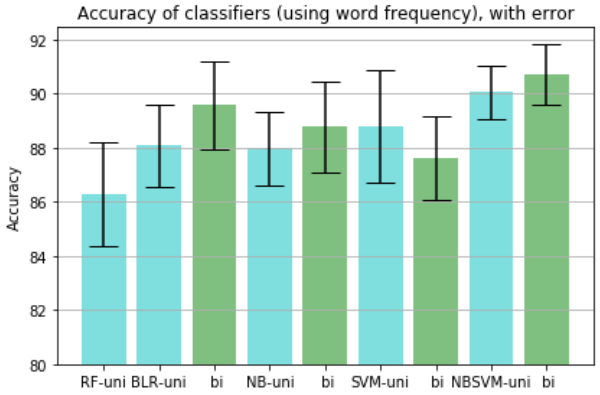

Figure 4: Accuracy of classifiers using tf-idf on the training set with error rates(%). Error bars indicate standard deviation.

4.2 Run times

See Appendix (section 8.1) for run times table

4.3 ROC curves

Appendix 8.3 contains the ROC curves for the logistic regression, Naive Bayes and Random Forest models.

4.4 Random Forest

he Random Forest(RF) model performed the worst results as expected. A reason for this may be that the decision tree model is not compatible with sentiment analysis because it is not possible to get good results without having a good connection between words. Decision trees do form with the knowledge of a word existing in a review or not so it can not have a good way of understanding if two words come side by side might make a sentence negative. The results with TF-IDF is 86.27% (Table 3), which is really close to others, so the power of the RF model should not be underestimated. The results are also parallel with Pouransari and Ghili's (2015) results. With the bag of words and Random Forest classifier, they obtained 84.35% accuracy, which is in around 1.92% error range we obtained with the cross-validation. The difference might be derived from the difference if datasets or the way they cleaned the dataset. Also, with the frequency representation results from Table 3, it can be concluded that the model accuracy increased along with the error rate from the cross-validation So, there is a trade-off between frequency and presence representations. The RF model is very inefficient when the runtime and results are compared (see *Appendix A*, Table 4). The model generation of trees takes lots of time and so does the validation part. Cross-validation time took more than double of all other model's validation times combined. So, when the accuracy and the amount of power considered this model is the most inefficient classifier.

4.5 Binary Logistic Regression

The Binary Logistic Regression model with frequency representation performed around 2% lower then presence representation. The reason for this may be that for probability calculations frequency representation is bad, since the same difference has been seen in the NB model. Both models do calculations with the probabilities and both accuracies were lower in TF-IDF representation. As expected model with bigram outperformed the one with unigram. BLR model is the second best model after NBSVM since the final accuracy is better than others. Also, the error rate in the cross-validation is smaller than SVM, that can be the discriminating reason to decide BLR is better than SVM. With the comparison to SVM and the given dataset, we can conclude that BLR with bigrams outperforms SVM in sentiment analysis. In *figure 6* and *figure 7*, the ROC curve shows that BLR is the best classifier among the three of them. The BLR model is capable of distinguishing and learning classes faster than others.

4.6 Naive Bayesian

NB performed adequately on our dataset. The classification accuracy gain from using bi-grams instead of uni-grams was statistically significant but less than expected. This holds for most classifiers we have tested. It is remarkable that NB performs so well "despite its simplicity and the fact that its conditional independence assumption clearly does not hold in real-world situations " as Pang and Lee (2002) note. As can be seen from the AUC values in the ROC curves, NB performance almost matches to that of the BLR model on the test set.Using word-presence vectors NB-uni got 89.37 (+/-1.60)% accuracy, and for the NB-bigrams, the cross-validation result was 89.90 (+/-1.03)% accuracy (Table 2). By looking at the overlapping error bars (Figure 3) we can not conclude that one is better than other but we can compare these results with representations. Using word-frequency vectors, NB- uni got 88.24 (+/-1.35)% accuracy, and for the NB-bigrams, the cross-validation result was 87.16 (+/-1.69)% accuracy (Table 2). The error bars are still close but since NB-bigram (presence) and NB-bigram (frequency) does not overlap we can conclude that using frequency representation for NB-bigram is worse than the binary representation. Also, by looking at the error rates we can say that overall binary representation gives better accuracy.

4.7 Support Vector Machine

The support vector machines model performed very well for both the uni- and bigram versions and both exceed 90% accuracy for presence representation. Like for all models, the bigram results of the model just surpassed the unigram results. By adjusting the

C-value (the width of the support vectors) the margin of making errors could be changed. In this research, the optimal C value was 0.0015. With the final results, it can be concluded that the SVM model is a stable model for long texts since results were not out of error range as it was in the NB model. These unigram and bigram results are also parallel to what Wang and Manning (2015) concluded. However, when the frequency representation has been used the accuracies that are obtained contradicts with the results of Pang and Lee (2002)'s research. In their case, the NB outperforms the SVM model. This contradiction was unexpected but possible with the parameter and implementation differences. On the other hand, the results of bigrams for all three models (NB. BLR, SVM) are almost the same. Considering the runtime of programs and the test set results we can conclude that the SVM-bi is much more efficient and accurate than the NB-bi model (see *Appendix A*, Table 4)

4.8 Naive Bayes and Support Vector Machine

The NBSVM model with bigram outperformed all other models with an accuracy of 91.03% (*Table 3*). Also, with the small +/-1.13% error rate, the model shows that is generally more stable than others. The NBSVM model performs better with bigrams and presence representation. The reason for this may be that the NBSVM is a combination of the NB and the SVM model so it still carries these models properties. Both NB and SVM model had better accuracies with bigrams and presence representation. Also, by looking at error rates it can be concluded that NBSVM is more robust and stable than both models. So, the combination of these model seems to not have a trade-off when these results are considered. The dataset has an average review length of 91 stemmed words. When the average words considered, reviews are not snippet texts. "NBSVM performs well on snippets and longer documents, for sentiment, topic and subjectivity classification, and is often better than previously published results."(Wang and Manning, 2015). The same conclusion can be obtained with this research's results. In this research, as Wang and Manning suggested NBSVM parameters were $\alpha = 1$, $C = 1$, $\beta = 0.25$. It is possible that by changing these parameters and using different datasets, different accuracies may be obtained. This can increase the best accuracy more but even using standard values is highly likely to give better results than other models as seen in *Table 1.*

5 Discussion

5.1 Leaving out the middle ratings

Leaving out the middle (2,3,4) ratings for the training as well as the test set is not without controversy. It is argued by Koppel, M., Schler, J. (2006) that there is no basis for the assumption that "there is less to learn from neutral documents than from documents with clearly defined sentiment.". Their arguments are, however, less applicable to our objectives, since they discus classification into three categories (positive, neutral, negative), whereas we perform binary classification. It does indicate that our original assumptions about the neutral reviews might not have been correct.

5.2 Binary classification versus multiclass classification

Although in our exploratory efforts, good results are obtained on the 1 star and 5 star split dataset, we were not pleased with the performance of our classifiers on the multiclass (ratings 1,2,3,4,5) dataset, and thus we decided to focus only on binary classification. The highest accuracy we achieved on some multiclass training data was around 65% by using the Random Forest model. It yielded a Rank-Loss of more than 1.0 stars. We suspect that one can always expect significant classification-error, because reviewers themselves (when they rate a movie) also perform a classification which has some variance, given the review-text: two people who feel the same way about a movie, might still not give it the same rating.

6 Conclusion

All of the models have been tested and analyzed for getting the best accuracy. For this, different parameters have been tested for each model with different combinations among each other. On the test set, NBSVM model had the highest accuracy, but given the standard deviation of the accuracies in the cross-validation of this model, we cannot conclude that this difference between NBSVM and most other models is statistically significant. Since the Random Forest model's error bars *(Figure 3)* does not overlap with the error bars of other models we can conclude that it is the worst model for sentiment analysis on this dataset. The reason behind this might be that the decision trees might not be compatible with long texts and sentiment analysis overall. On the other hand, Naive Bayes(NB) performance was lower than SVM when the test set results are considered which corresponds with Wang and Manning's (2015) results. The reason Wang and Manning (2015) give for this, is that reviews are generally longer than the 'snippet' length texts at which NB has been previously found to perform adequately. Also NB, results showed that the model is unstable with long texts so that the end results we got from this research deviated from the cross-validation results around 3%.

The unigram and bigram version of every model has been tested. As Wang and Manning(2015) explained "...in sentiment classification there are much bigger gains from bigrams, because they can capture modified verbs and nouns". The results showed that using bigrams is more accurate for sentiment analysis when the test results are considered *(Table 2 and Table3)*. Our research questions cannot be answered conclusively, given that error rates of different model's accuracies overlap on the training data 10-fold cross validation results *(Table 2 and Table 3)*. Overall, the results that have been obtained were quite satisfying with above 91% accuracy. For better results, different parameter combinations can be tested or different model combinations can be implemented as NBSVM.

7 References

7.1 Academic resources

1. Al Amrani, Y., Lazaar, M., El Kadirp, K. E. (2018). Random forest and support vector machine based hybrid approach to sentiment analysis. In *Procedia Computer Science.* https://doi.org/10.1016/j.procs.2018.01.150

2. Bird, S., Klein, E., Loper, E. (2009) Natural Language Processing with Python. Sebastopol, CA: O'Reilly Media.

3. Pang, B., Lee, L., Vaithyanathan., S. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proc. of EMNLP-02.*

4. Dai, W., Xue, G. R., Yang, Q., Yu, Y. (2007, July). Transferring naive bayes classifiers for text classification. In *AAAI (Vol. 7,* pp. 540-545).

5. Gandomi, A. and Haider, M., Beyond the hype: Big data concepts, methods, and analytics, *International Journal of Information Management, vol. 35, no. 2, pp. 137–144*, 2015. doi¿10.1016/j.ijinfomgt.2014.10.007

6. He, R., McAuley, J. Modeling the visual evolution of fashion trends with one-class collaborative filtering. *WWW*, 2016

7. Koppel, M., Schler, J. (2006). The importance of neutral examples for learning sentiment. *Computational Intelligence, 22*(2), 100-109.

8. Lutfi, A. A., Permanasari, A. E., Fauziati, S. (2018). Corrigendum: Sentiment Analysis in the Sales Review of Indonesian Marketplace by Utilizing Support Vector Machine. *Journal of Information Systems Engineering and Business Intelligence, 4*(2), 169. https://doi.org/10.20473/jisebi.4.2.169

9. Marafino, B. J., Davies, J. M., Bardach, N. S., Dean, M. L., Dudley, R. A. (2014). N-gram support vector machines for scalable procedure and diagnosis classification, with applications to clinical free text data from the intensive care unit. *Journal of the American Medical Informatics Association, 21(5)*, 871-875

10. McAuley, J.,Targett, C. Shi, J., van den Hengel, A. . Image-based recommendations on styles and substitutes. *SIGIR*, 2015

11. McCallum, A., Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization (Vol. 752, No. 1*, pp. 41-48).

12. Park, HA. An Introduction to Logistic Regression: From Basic Concepts to Interpretation with Particular Attention to Nursing Domain. *J Korean Acad Nurs. 2013 Apr;43*(2):154-164.

13. Singh, N. K., Tomar, D. S., Sangaiah, A. K. (2018). Sentiment analysis: a review and comparative analysis over social media. *Journal of Ambient Intelligence and Humanized Computing, 0(0), 1–21.* https://doi.org/10.1007/s12652-018-0862-8

14. Wang, S., Manning, C. (2015). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 94305(1), 90–94. https://doi.org/10.1021/jp402392y

7.2 Less-academic resources

1. '1.1.11. Logistic Regression' (Year, Month unknown). Retrieved from https://scikit-learn.org/stable/modules/linear_model .html#logistic-regression

2. Chin, J. (2016) 'NBSVM' Retrieved from https://github.com/Joshua-Chin/nbsvm

3. Droidhead (2018), How to process textual data using TF-IDF in Python. Retrieved from: https://medium.freecodecamp.org/how-to-process-textual-data-using-tf-idf-in-python-cd2bbc0a94a3

4. Generalized Linear Models — scikit-learn 0.20.3 documentation. Retrieved from https://scikit-learn.org/stable/modules/linear_model .html#logistic-regression

5. Kulb, A. (2018) 'Sentiment Analysis with Python (Part 1)' Retrieved from https://towardsdatascience.com/sentiment-analysis-with-python-part-1-5ce197074184

6. Kulb, A. (2018) 'Sentiment Analysis with Python (Part 2)' Retrieved from https://towardsdatascience.com/sentiment-analysis-with-python-part-2-4f71e7bde59a

7. Pouransari, H., Ghili, S. Deep learning for sentiment analysis of movie reviews (2015). Technical report, Stanford University, Retrieved from: https://cs224d.stanford.edu/ reports/PouransariHadi.pdf

8. Shaikh, J. (2017) 'Machine Learning, NLP: Text Classification using scikit-learn, python and NLTK.' Retrieved from https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a

8 Appendix

8.1 Run times

Model	Presence	TF-IDF
RF-uni	706.9 (s)	694.0 (s)
BLR-uni	15.2 (s)	88.07
BLR-bi	29.6 (s)	17.5 (s)
NB-uni	38.0 (s)	22.3 (s)
NB-bi	60.1 (s)	33.5 (s)
SVM-uni	3.1 (s)	2.9 (s)
SVM-bi	9.5 (s)	9.5 (s)
NBSVM-uni	89.5 (s)	88.0 (s)
NBSVM-bi	129.9 (s)	117.1 (s)

Table 4: Runtimes of models for both presence and frequency representation (seconds), (10-fold cross validation time included) System specifications: (model-MSI-GV62 8RE), i7-8750H @2.20GHz(up to 3.9), 16GB RAM, NVDIA GTX1060

8.2 code

https://drive.google.com/open?id=1sqXsDO2qAc0fVO8wio1ItoPcWc

8.3 Barchart

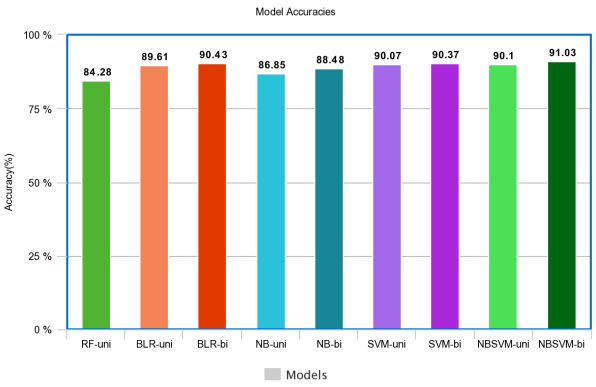

Figure 5: Accuracy of classifiers on the test set. (%)

8.4 ROC curves

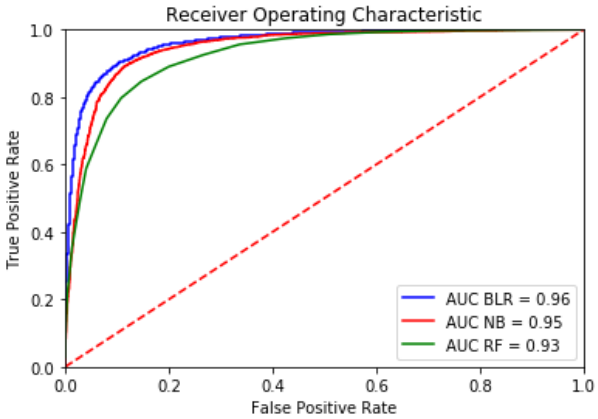

Figure 6: ROC curves using word presence

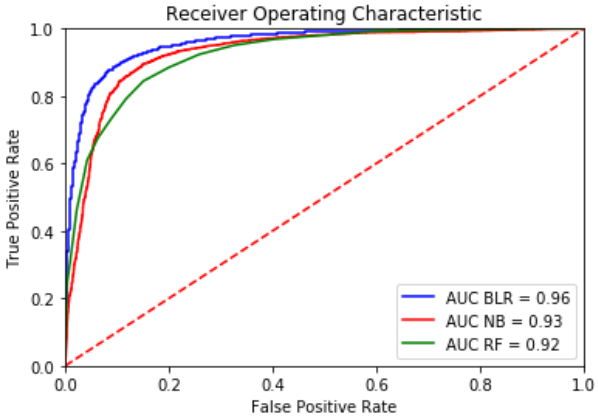

Figure 7: ROC curves using tf-idf

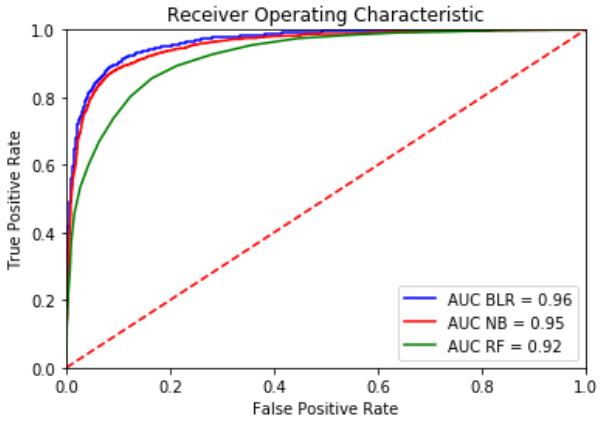

Figure 8: ROC curves using word presence and bigrams

8.5 Contribution table

Name	Contribution
İlayda Tukuş	Models (2,5), Results (1,2), Conclusion
Xerxes Koehoorn	Programming model(3) NB and ROC graphs, Preparation of the dataset, Introduction, Models(3,4), LaTeX.
Yasin Aydın	Programming models(1,2,3,4,5), Preparation of dataset, Abstract, Introduction, Models (2,5), Results(1,2,4,5), Conclusion.
Thijs Roozen	Introduction, Models (1,4), data inspection and preparation, Results(2,4)
Humam Syauqi Dawa	Models (1,4), grammar and syntax Check