Text Classification of Positive and Negative Reviews for Sephora Skincare Products with Linear Support Vector Machine

Yura Ueno
gusuenoyu@student.gu.se

Abstract

In this report a linear SVM text classifier is presented in order to classify positive and negative reviews for Sephora skincare products. The reviews were scraped and collected from the Sephora online store in March 2023. Using the data, a linear SVM was trained and tuned to classify customer reviews into "recommending" (positive) and "not recommending" (negative). The evaluation of the model on the test data showed good performance with an accuracy of about 0.93 and an AUC of 0.98. Further analysis was also performed on errors that were classified with high confidence and also on the importance of the features. Overall the model performed well but it might be of future interest to extend the sentiment analysis to more than just positive and negative or to use more advanced model or embeddings that are better suited for highly nuanced texts.

1. Introduction

Sentiment analysis has become an essential tool for businesses seeking to gain actionable insights from customer feedback. In the highly competitive beauty and skincare industry, where customer opinions and experiences significantly influence purchasing decisions, understanding consumer sentiment is crucial for staying ahead of market trends. This paper presents a sentiment analysis of positive and negative reviews of Sephora skincare products, leveraging machine learning techniques to classify customer opinions. By analyzing the sentiments expressed in product reviews, Sephora can not only gauge overall customer satisfaction but also identify specific product strengths and weaknesses, which can inform product development, marketing strategies, and customer service improvements. This project offers a valuable approach for businesses like Sephora to better understand customer preferences and enhance their ability to respond effectively to consumer needs by leveraging machine learning techniques.

2. Classifier Development

The learning algorithm used for the sentiment analysis of the reviews in this project is a linear support vector machine. There were several motivations for choosing this algorithm.

Firstly, SVM is one of the most commonly used traditional machine learning algorithms for text classification (Li et al.,2022; Aliwy et al, 2017). Secondly, linear SVM is relatively fast to train and tune, and thus preferred in this case since the timeline for this project was short. Although there have been many research indicating the high performance of deep neural networks for sentiment analyses, training and building a successful deep neural networks requires significantly more time and more computational resources. Since neural networks are sensitive to the initial randomization of weights and prone to converging to a local minimum due to having objective functions with many local minima, training with different initialization is recommended for neural networks, which is one of the reasons why the time required for training and tuning neural networks is very long. On the other hand, the objective function of SVM is convex and guaranteed to converge to a global minimum regardless of their initial configuration, which allows SVMs to have less time required for the optimization of parameters and model training. When comparing between linear SVM and non-linear SVM, non-linear SVM such as SVM with polynomial kernel or RBF kernel also require considerably more time to train and tune, especially with large datasets. Therefore, linear SVM was preferred to deep neural networks and non-linear SVM in this project.

Additionally, several previous research support the use of a linear kernel in text

classification by arguing that most text data are linearly separable (Thoresten, 1998; Soumick, 2019), and that the linear kernel is recommended over non-linear kernel when the number of features is large (Hsu et al., 2003), which is the case in text classification. With these reasons in mind, linear support vector machine is used in this project.

2.1. Data Description

The data used for this project were customer reviews for Sephora skincare products developed by Inky (2023). The data set was obtained from “Sephora Products and Skincare Reviews” on Kaggle. The reviews were collected using a Python scraper in March 2023 from the Sephora online store along with information about all beauty products, including product and brand names, prices, ingredients, ratings, and all features. The dataset also includes sentiment labels of the reviews, indicating if the reviewer recommends the product or not (1 for “recommending" and 0 for “not recommending”).

After removing duplicates, the dataset contains 398,854 reviews in “reviews_0-250.csv”, with 335,431 of them recommending the products while the rest of 634,23 not recommending. Only 20,000 positive reviews and 20,000 negative reviews were used for this project considering that around 40,000 reviews would be sufficient for this kind of sentiment classification and also in order to create a balanced dataset. After selecting those 40,000 reviews, the languages of the reviews were checked using a language detection library “langdetect”. As a result, it was revealed that most of the reviews were in English while there were some reviews written in other languages such as French and Norwegian. As this project did not intend to create multi-lingual sentiment classifier, the reviews in other languages than English were removed from the dataset. After limiting to only English reviews, the number of the reviews were also further limited to be 19,800 for positive and negative reviews each so that we have a perfectly balanced dataset in order to prevent bias towards the dominant class and to ensure that the model learns to recognize patterns for all classes equally. This leads to better generalization when the model encounters new, unseen data, allowing it to accurately classify sentiments across the entire spectrum.

After narrowing the dataset into 39,600 reviews in total, the reviews along with the class labels were shuffled and then split into training and test dataset using sklearn’s train_test_split() function with the ratio of 80% into the training and 20% into the test data. This resulted in 15,840 positive and negative reviews each in the training dataset and 3,960 positive and negative reviews each in the test dataset, ensuring that both the training and the test datasets are balanced.

2.2. Data Representation and Preprocessing

In the dataset, each instance containes a review and its class label (0 for not recommending and 1 for recommending). The reviews were converted to a sparse matrix of TF-IDF features using sklearn's TfidfVectorizer(). TF-IDF (Term Frequency Inverse Document Frequency) is a numeric statistic which is expected to reflect the importance of a term in a document in a collection or a corpus. The calculation of TF-IDF is shown in the formula below, where $tf(t,D)$ is the count of word t in document D and $df(t)$ is the number of documents containing t.

$$TF\text{-}IDF(t,D) = tf(t,D) \cdot log((1+N)/(1+df(t)))$$

As can be seen in the equation, TF-IDF combines two key components: term frequency (TF) and inverse document frequency (IDF). Term frequency measures how often a term appears in a document, while inverse document frequency accounts for how common or rare the term is across all documents. The goal is to highlight terms that are unique to a document by giving higher weights to terms that are frequent within the document but infrequent across the rest of the corpus. The presence of "1+" in the numerator and denominator prevents division by zero and smooths the calculation. TF-IDF ensures that frequently occurring terms across many documents carry little weight (e.g. "the", "a", "is"), while terms that appear in fewer documents but are frequent in a specific one are given higher importance. Additionally, it converts textual data into numerical feature vectors, enabling machine learning models to use the text as input. Using TfidfVectorizer(), each review is converted into a numerical feature vector that indicates the TF-IDF weighted frequency of terms.

2.3. Hyperparameter Tuning

Linear SVC has a hyper parameter C, which is used to set the amount of regularization. This hyper parameter was tuned through grid search

using GridseachCV from sklearn. Since the dataset used in this project is well-balanced between the two classes, and there is no preference between prioritizing correct classification of positive reviews and negative reviews, the C value that yields the highest accuracy is chosen to be the best working C value. After the first round of grid search between 0.01~10,000,000,000., the linear SVC seemed to achieve higher accuracies when C is around 0.1~10. Thus, a second round of grid search was conducted with C set to values between 0.1 and 10. As a result, the linear SVC had higher accuracy when C was between 0.1 and 1.0. Therefore, a final grid search was conducted with C set to values between 0.1 and 1.0. Although there was almost no difference in accuracy among the searched C values, C=0.3 resulted as can be seen in Table 1. Therefore, 0.3 was chosen as the final value of C.

C value	Accuracy	Ranking
0.2	0.92446	3
0.3	0.92541	1
0.4	0.92484	2
0.5	0.92481	4
0.6	0.92434	5
0.7	0.92390	6
0.8	0.92371	7

Table 1: Results of final grid search.

To further test this value as a hyper parameter cross validation was conducted and average accuracy, F1 score, precision, and recall were calculated to observe the overall performance of the model on training data. The results are shown in Table 2 below.

Score	Value
Average accuracy	0.9248
Average F1 score	0.9244
Average precision	0.9299
Average recall	0.9189

Table 2: Average scores from cross validation using C=0.3.

The result of the cross-validation showed that the model had high average accuracy of 0.9248 and high average F1 score of 0.9244. The average precision and recall were approximately the same, indicating that the model classifies positive and negative comments equally well.

3. Results

The tuned and trained model was then evaluated using the test data. This was done first by calculating accuracy on the test dataset as well as checking the ROC curve and AUC of the model.

3.1. Model Evaluation

The metric used to evaluate the performance of the model on the test data was accuracy as the data was well-balanced between the two classes and there was no preference between prioritizing correct classifications of positive and negative reviews in this project. As can be seen in the confusion matrix in Figure 1 below, the accuracy of the model on the test data was about 0.9274, indicating fairly high performance on correctly classifying the test data. Additionally, the confusion matrix shows that the model classified the positive and negative reviews equally well, which also can be seen in Figure 1.

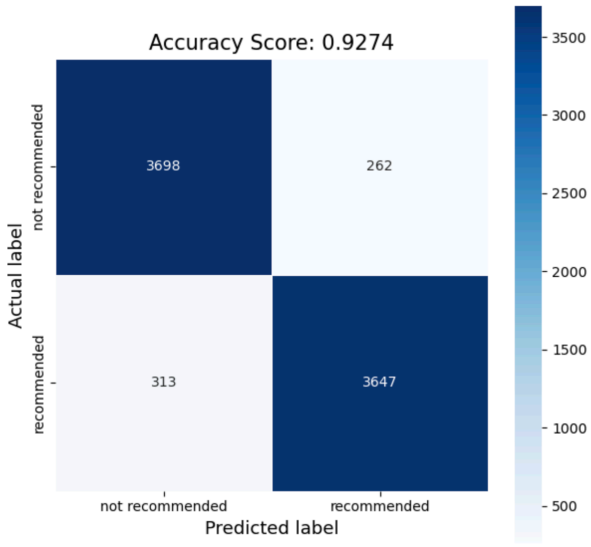

Figure 1: Confusion Matrix

To evaluate the model further, the ROC curve was used. This is a graph of false positive rate

versus true positive rate, where the false positive rate is the proportion of false positives among the negative instances and the true positive rate is the proportion of true positives among the positive instances. The area underneath this curve was then calculated as the AUC, where a perfect model has a value of 1.

The ROC curve and the AUC of the model is shown in Figure 2. The model has an AUC of 0.98, which is very close to 1 and indicates great performance of the model on an unseen data.

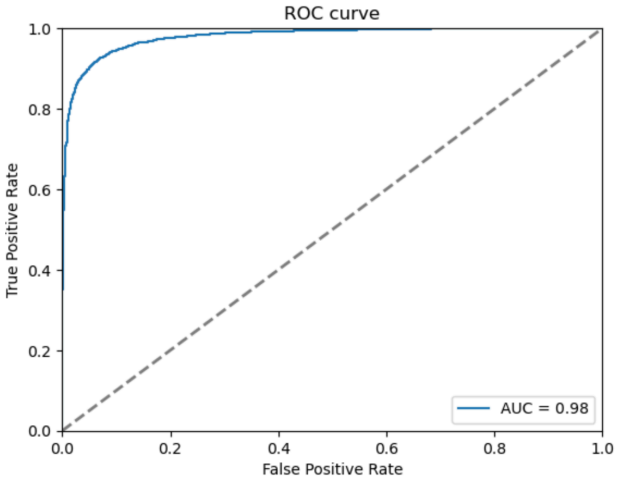

Figure 2: ROC Curve & AUC

3.2. Selected Error Analysis

Another important part of model evaluation is to further investigate its weak points. Although the linear SVM achieved very high accuracy, there were still some reviews that were misclassified as can be seen in the confusion matrix. In order to investigate what could have potentially caused those misclassifications, the five misclassified reviews with the highest confidence score were further analyzed. Table 3 shows those five incorrectly classified comments, followed by evaluations of each review.

#	Comment	Predicted
1	I was really excited to try this product based on the reviews. I loved everything about it - scent, texture etc. Sadly, it caused me to breakout all over my face and I had to stop using it.	0
2	it's really good but so overrated like there's nothing special about it, not worth the money tbh	0
3	This product didn't work for my skin personally, but if you have dry normal skin this would work for you. I have combo acne prone skin and the first couple weeks of use it was amazing but then I started to notice it was clogging the pores on my cheeks and gave me milia bumps near my under eye area. It's a heavier cream, and you don't need that much product to cover your face. It kind of takes a while to sink into the skin, I used this as a nighttime cream for that reason.	0
4	I have super dry lip every single day. my lip is always has peel.try many stuff doesn't work for it. And I watch you tube many people recommend this sleep mask. When you used before you sleep and second day you lip is smooth. But after few hours my lip is still get peel. Used about four mouthed still same never buy it again	1
5	I have never needed chapstick or lip moisturizer. When i use products like these they just dry my lips out after use. This is great for the price but if you have naturally nice lips you won't need this. I put this on my husband every night because he is in love with burt bees so it works for him. He has yet to have a complication.	1

Table 3: Incorrectly classified comments with highest confidence scores and their predicted label.

Review 1: The reviewer expresses enthusiasm for the product's features but ultimately reports a negative experience (breakouts) and stops using it. The true label of 1 might indicate a labeling error, as the overall sentiment skews negative due to the skin reaction. Alternatively, the model might have been confused by the initially positive language and missed the shift in sentiment when the reviewer mentioned the product's adverse effects.

Review 2: The reviewer acknowledges that the product is good but criticizes it for being overrated and not worth the price. This mixed review, while mostly critical, is labeled as 1 (recommendation), possibly suggesting a labeling error.

Review 3: The reviewer provides a detailed account of how the product didn't suit their skin type but could work for others, and thus recommending it for others. The mention of positive effects at first might have led to the true label being 1, but the model likely focused on

the later negative experiences (clogging pores, milia bumps), causing it to predict 0.

Review 4: The reviewer describes their persistent lip problem and frustration with the product not resolving the issue, concluding they will never buy it again. The true label of 0 seems accurate, but the model might have been misled by phrases like “recommend this sleep mask” and the initial hopeful tone, resulting in a predicted label of 1.

Review 5: This reviewer explains that they don’t personally need the product but uses it for someone else who loves it. The mixed nature of the review, with neutral-to-positive language about its effectiveness for another person, could have confused the model. The true label of 0 aligns with the reviewer's own indifference, but the model might have focused on the positive remarks about the husband’s experience, leading to a prediction of 1.

Overall, the misclassifications seem to have stemed from mixed or nuanced language where positive and negative aspects are both present, which likely confused the model. In some cases, the true labels might also be inaccurate, especially when the review tone shifts from positive to negative or vice versa, or when the reviewer recommends the product for others despite the product not working for them. Using more advanced models or embeddings like transformer-based models and embeddings (e.g., BERT), which understand the context and nuances of sentences better could help solve these kind of misclassifications.

3.3. Feature Importance

As a final analysis of the model, the feature importance scores were calculated, to see which words are most important in the classifications of the reviews. In Table 4 below, the ten words with highest importance are shown. Observing the ten most important features, most of the words have fairly strong and clear meaning, most likely providing the classifier with clearer information on the tone and sentiment of the reviews (e.g. “love”, "amazing", "best", "skeptical", and “great").

Feature name	Feature importance
love	3.211
amazing	2.624
soft	2.559
best	2.459
skeptical	2.373
gentle	2.190
cucumber	2.162
great	2.133
perfect	2.130

Table 4: The ten features with highest feature importance.

4. Conclusion

This paper presented a linear SVM classifier that classifies reviews as “recommending” and “not recommending” with high accuracy of 0.927. Despite linear SVM being a model that is fairly easy and fast to train, the model was able to successfully classify majority of the reviews, indicating the effectiveness of a linear SVM classifier in this kind of task. That being said, the model failed to classify 575 reviews out of 7,920 reviews in the test dataset. Investigating the five reviews that were misclassified with the highest confidence, the model seemed to have struggled to correctly classify when reviews contained mixed or nuanced language with both positive and negative aspects combined. Using more advanced models or embeddings that can capture the context and nuances of sentences better such as transformer-based models and embeddings (e.g. BERT) could help overcome this challenge of classifying sentiment-mixed reviews. Additionally, the investigation of the misclassified reviews revealed potential labeling errors in the dataset. The mislabeling probably stemed partly from the difficulty of labeling reviews that do not clearly indicate wether they recommend the products or nor, and also partly from the human error caused by the annotators. A potential way to avoid mislabeling and remove reviews that cannot be clearly classified is to have multiple annotators label the same reviews. By doing so, reviews receiving conflicting labels can be further evaluated or removed. Alternatively, developing a model that can classify “neutral” as a label could be considered.

References

Aliwy, A.H., & Ameer, E.H. (2017). Comparative Study of Five Text Classification Algorithms with their Improvements.

Hsu, C.-W., Chang, C.-C. & Lin, C.-J. (2003). A Practical Guide to Support Vector Classification (). Department of Computer Science, National Taiwan University .

Inky, N. (2023). Sephora Products and Skincare Reviews. Kaggle. Available at: https://www.kaggle.com/datasets/nadyinky/sephora-products-and-skincare-reviews/data?select=reviews_0-250.csv.

Li, R., Liu, M., Xu, D., Gao, J., Wu, F., Zhu, L. (2022). A Review of Machine Learning Algorithms for Text Classification. In: Lu, W., Zhang, Y., Wen, W., Yan, H., Li, C. (eds) Cyber Security. CNCERT 2021. Communications in Computer and Information Science, vol 1506. Springer, Singapore. https://doi.org/10.1007/978-981-16-9229-1_14

Soumick Chatterjee, Pramod George Jose, Debabrata Datta, "Text Classification Using SVM Enhanced by Multithreading and CUDA", International Journal of Modern Education and Computer Science(IJMECS), Vol.11, No.1, pp. 11-23, 2019.DOI: 10.5815/ijmecs.2019.01.02

Thorsten Joachims. (1998). Text categorization with Support Vector Machines: learning with many relevant features. In Proceedings of the 10th European Conference on Machine Learning (ECML'98). Springer-Verlag, Berlin, Heidelberg, 137–142. https://doi.org/10.1007/BFb0026683