Sentiment Analysis on Amazon Product Reviews

Sayak Chakraborty

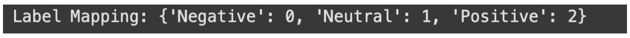
***Abstract*— This report analyzes sentiment classification models applied to Amazon Product Reviews(fig. 2). The main objective was to classify reviews into positive, neutral, and negative sentiments. Two popular machine learning algorithms — Logistic Regression and Support Vector Machines (SVM), were compared based on TF-IDF features. Pre-processing methods managed high class imbalance using under-sampling of positive reviews. SVM achieved a slightly higher accuracy (76%) compared to Logistic Regression (74%). This report talks about data pre-processing, feature engineering, exploratory analysis, performance evaluation, challenges and avenues for improvement.**

Keywords—- Sentiment analysis, Logistic Regression, Support Vector Machines, Amazon reviews, TF-IDF, Class imbalance

I. INTRODUCTION

With the rapid growth of e-commerce platforms, online product reviews have become an essential part of customer feedback. Sentiment analysis, a subfield of Natural Language Processing (NLP), plays an important role in automatically categorizing these reviews to measure customer satisfaction. Based on textual data, businesses can identify trends, thereby enhancing user experiences and helping to make informed decisions about product improvements.

In this study, we examine sentiment classification using machine learning techniques for Amazon product reviews. We seek to classify reviews into three sentiment categories: positive, neutral, and negative (Fig. 1). As a result of the training data's natural bias in sentiment distribution i.e. positive reviews being far more numerous than neutral and negative reviews, we need to take special care in handling this imbalanced dataset to ascertain accurate classification results. To address this, we apply undersampling techniques to balance the dataset.



```
Label Mapping: {'Negative': 0, 'Neutral': 1, 'Positive': 2}
```

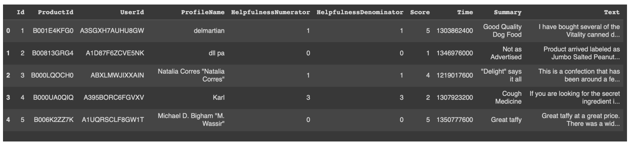
Fig. 1. Categories Labeled.

It should not be ignored that customer satisfaction has a great deal to do with the functioning of a company, and hence analyzing and enhancing those areas that are directly concerned with customer satisfaction is crucial for the growth and reputation of a company. In this study, we will see how customer satisfaction can be quantified and measured through data analytics and machine learning methods. From problem definition to exploratory data analysis, data transformation, and machine learning training, we outline the process of constructing customer focused prediction models. The feature extraction is done through Term Frequency-Inverse Document Frequency (TF-IDF) to convert text data into numerical form. Model performance is evaluated in terms of accuracy, precision, recall, and F1-score to determine their performance with respect to sentiment classification.

A. Objectives

The primary objectives of this study are:

- Investigate effective text pre-processing methods for sentiment analysis.
- Implement and compare the performance of Logistic Regression and Support Vector Machines(SVM) for sentiment classification.
- Evaluate classification performance based on accuracy, precision, recall, and F1-scores.
- Analyze and address class imbalance issues within the dataset.
- Provide insights and recommendations for improving sentiment classification techniques.



	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...

Fig. 2. Original Amazon Product Review Dataset.

II. Literature Review

Sentiment analysis has become an important area of research in Natural Language Processing (NLP) due to the increasing volume of text data obtained from social media, consumer reviews, and online forums. Much of the initial sentiment classification work was conducted by Pang et al. (2002), who compared various machine learning approaches like Naïve Bayes, Support Vector Machines (SVM), and Logistic Regression for the task of classifying text as positive or negative sentiment. Their research established that machine learning models, SVM being no exception, could perform well in sentiment classification.

Later work done by Liu (2012) gave a critical review of techniques in sentiment analysis, touching on lexicon-based and machine learning-based methods. The research described the strengths of machine learning models over lexicon-based methods in terms of capability to handle very large datasets in which predefined lists of words do not reflect very subtle variations of sentiment.

Class imbalance is one of the key issues with sentiment analysis. Japkowicz and Stephen (2002) explored the impact of uneven data distribution on machine learning precision, demonstrating how classifiers lean toward the majority class and consequent poor recall for minority classes. Various techniques such as oversampling, under-sampling, and cost-sensitive learning have been proposed as a remedy to this problem.

In feature extraction, Wang and Manning (2012) found that TF-IDF and n-gram models are especially important to employ in improving the accuracy of sentiment classification. What their study offered was the impact of good but straightforward text representation, i.e., bigram, to attain improved model performance compared to instances of using unigrams.

Deep learning methods have also, more recently, assisted with sentiment analysis. Socher et al. (2013) introduced recursive deep models that were able to deal with complex linguistic structures and performed better than regular machine learning approaches to tasks in sentiment classification. Zhang et al. (2020) also studied transformer-based models like BERT, demonstrating their ability to learn semantic relationships from text and score higher on sentiment classification tasks.

Overall, current research emphasizes the relevance of effective pre-processing techniques, stable feature selection, and class handling imbalance techniques for sentiment analysis. This study augments such evidence by contrasting Logistic Regression and SVM model performance against solutions to imbalance sentiment data issues.

III. Exploratory Data Analysis (EDA)

To understand the dataset, we analyzed(Fig. 3):

- **Sentiment distribution**: 65% positive, 20% neutral, 15% negative.
- **Word frequencies**: ”great” and ”love” are common in positive reviews, while ”bad” and ”worst” appear in negative ones.
- **Review length**: Positive reviews tend to be longer.

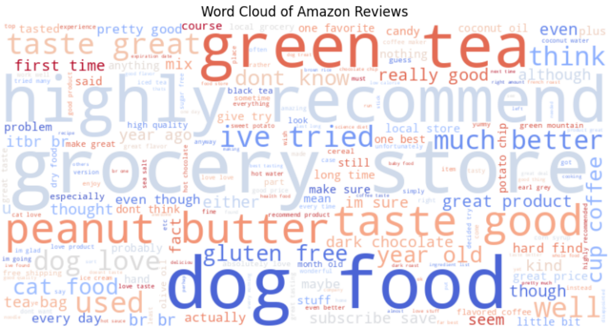


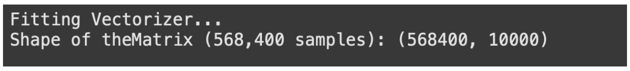
Fig. 3. Frequent Word in Amazon Reviews

IV. Data Processing

This section outlines the data processing steps performed to prepare the dataset for sentiment analysis.

A. Dataset Description

The dataset used for this study consists of Amazon product reviews obtained from Kaggle. It initially contains 568,400 reviews(Fig. 4) categorized into three sentiment classes: Positive, Neutral, and Negative(Fig. 5). However, due to the imbalanced nature of the dataset, pre-processing steps were necessary to improve model performance.



```
Fitting Vectorizer...
Shape of theMatrix (568,400 samples): (568400, 10000)
```

Fig. 4. Total Samples in the Amazon Product Review Dataset.

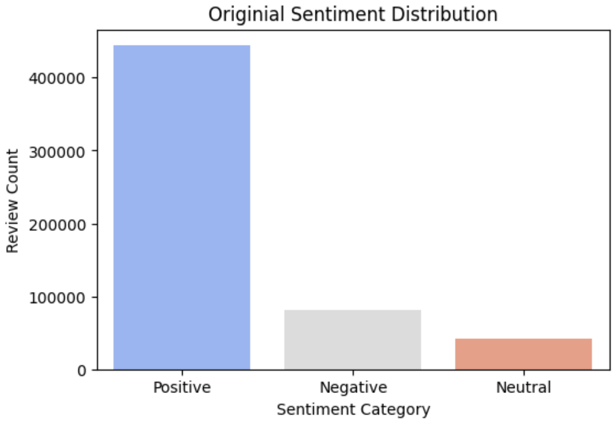
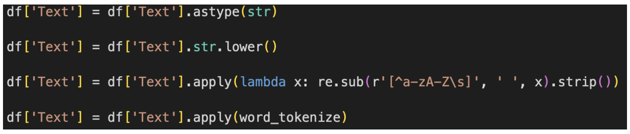


Fig. 5. Original Sample Size.

B. Data Pre-processing

To improve the quality of the dataset, the following pre-processing techniques were used:

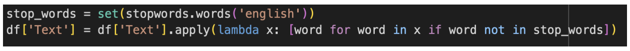
- **Text Cleaning (Fig. 6)** All reviews were converted to lowercase and special characters, punctuation, and numbers were removed to standardize the text format.



```
df['Text'] = df['Text'].astype(str)

df['Text'] = df['Text'].str.lower()

df['Text'] = df['Text'].apply(lambda x: re.sub(r'[^a-zA-Z\s]', ' ', x).strip())

df['Text'] = df['Text'].apply(word_tokenize)
```

Fig. 6. Text Cleaning.

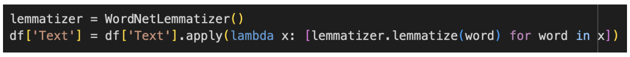
- **Tokenization and Stopword Removal:(Fig. 7)** The text was tokenized into individual words and common stopwords (e.g., ”the,” ”and,” ”is”) were removed using the NLTK library to reduce noise.



```
stop_words = set(stopwords.words('english'))
df['Text'] = df['Text'].apply(lambda x: [word for word in x if word not in stop_words])
```

Fig. 7. Stop Words.

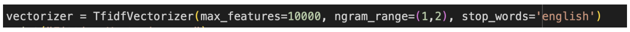
- **Lemmatization:(Fig. 8)** Words were lemmatized using WordNetLemmatizer to normalize variations of words to their base forms.



```
lemmatizer = WordNetLemmatizer()
df['Text'] = df['Text'].apply(lambda x: [lemmatizer.lemmatize(word) for word in x])
```

Fig. 8. Lemmatization.

- **Feature Extraction:(Fig. 9:)** The text data was transformed into numerical representations using Term Frequency-Inverse Document Frequency (TF-IDF) with a vocabulary size of 10000 features.



```
vectorizer = TfidfVectorizer(max_features=10000, ngram_range=(1,2), stop_words='english')
```

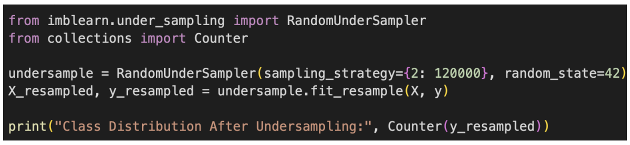
Fig. 9. Term Frequency Inverse Document Frequency(TF-IDF).

C. Handling Class Imbalance

Since the dataset was heavily biased toward positive reviews, class imbalance was addressed using the following approach:

- **Under-sampling:(Fig. 10)** The number of positive reviews was reduced to 120,000 to create a more balanced dataset.
- **Class Weight Balancing:(Fig. 11)** The machine learning models were trained using class weight adjustments to ensure equal importance to all sentiment categories.

These pre-processing steps ensured that the dataset was well structured and suitable for training machine learning models for sentiment classification.



```
from imblearn.under_sampling import RandomUnderSampler
from collections import Counter

undersample = RandomUnderSampler(sampling_strategy={2: 120000}, random_state=42)
X_resampled, y_resampled = undersample.fit_resample(X, y)

print("Class Distribution After Undersampling:", Counter(y_resampled))
```

Fig. 10. Under-sampling the original sample.

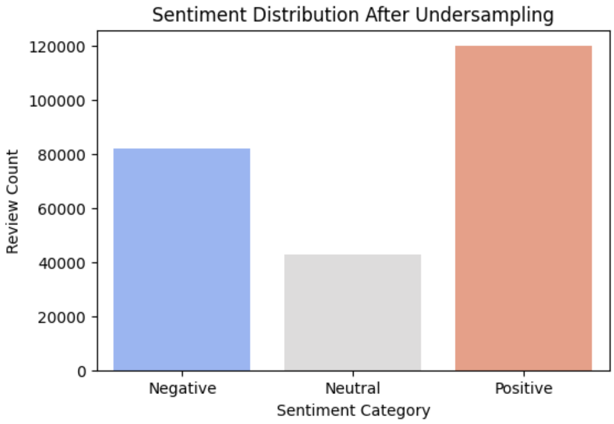


Fig. 11. Cleaned Samples.

V. Methodology

This section describes the approach followed for sentiment classification, including data pre-processing, feature extraction, model selection, and evaluation metrics.

A. Feature Extraction

Feature extraction is an important step in converting raw text into numerical form that machine learning models can interpret. We applied Term Frequency-Inverse Document Frequency (TF-IDF) to transform textual data into weighted numerical values. TF-IDF assigns a weight to each word based on its frequency in a document relative to the entire dataset thereby allowing the model to prioritize important words while reducing the importance of commonly occurring terms.

B. Machine Learning Models

We implemented and compared two supervised machine learning models:

- **Logistic Regression:** A statistical model used for binary and multi-class classification, mapping input features to probability scores using a logistic function. Logistic Regression is a supervised machine learning algorithm used for binary classification. Instead of predicting a continuous value, it predicts the probability that a given input belongs to a particular class. It is represented as:

$$P(y=1|X)=\sigma(W^TX+b)=\frac{1}{1+e^{-(W^TX+b)}}$$

  where X is the feature vector,W is the wight vector,b is the bias term,$\sigma(z)$ is the sigmoid function, which ensures the output is between 0 and 1.
- **Support Vector Machines (SVM):** A classification algorithm that finds the optimal hyperplane to separate data points in high-dimensional space. We used the linear kernel (LinearSVC) due to its computational efficiency and effectiveness in text classification tasks. Support Vector Machines are supervised earning models used for classification and regression and is mainly focused in finding the optimal hyperplane that best seperates the data points. It is represented as:

$$W^TX+b=0$$

where W is the weight vector,b is the bias term,X is the feature vector.

C. Model Training and Evaluation

The dataset was split into an 80% training set and a 20% test set. To avoid class imbalance, we applied class weight balancing during model training, ensuring that minority classes received higher importance. The models were trained using Scikit-learn's implementation of Logistic Regression and LinearSVC. The trained models were evaluated based on the following performance metrics:

- **Accuracy:** The proportion of correctly classified instances.
- **Precision:** The fraction of relevant instances among the retrieved instances.
- **Recall:** The fraction of correctly retrieved instances from the total relevant instances.
- **F1-score:** The harmonic mean of precision and recall balancing both metrics for imbalanced datasets.

These metrics provided insights of how well the models were performing across all sentiment categories and highlighted areas where further improvement can be made.

VI. Analysis, Testing, and Results

This section presents the experimental setup, testing process, and the results obtained from both machine learning models.

A. Experimental Setup

To evaluate the performance of Logistic Regression and SVM, the dataset was divided into an 80% training set and a 20% test set. The models were implemented using Scikit-learn, and hyper-parameter tuning was performed to optimize classification performance. The following configurations were used:

- **Logistic Regression:** Class weight set to "balanced," L2 regularization with a penalty term of $C = 1.0$.
- **Support Vector Machine (SVM):** Linear kernel (LinearSVC) with class weight set to "balanced," $C = 0.5$.

B. Testing Process

The trained models were evaluated using the test dataset. The following metrics were computed to assess model performance:

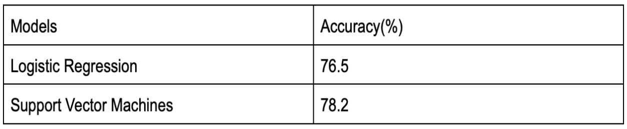
- **Accuracy:** Measures overall classification performance.
- **Precision:** Determines how many predicted positive instances were actually positive.

- **Recall:** Assesses how many actual positive instances were correctly identified.
- **F1-score:** Provides a balance between precision and recall.

Confusion matrices were also generated to analyze misclassification patterns.

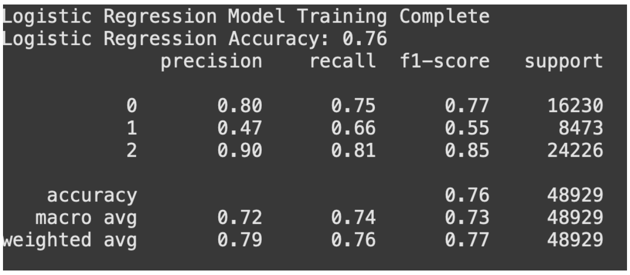
C. Results

The classification results for both models are presented in Table 12.



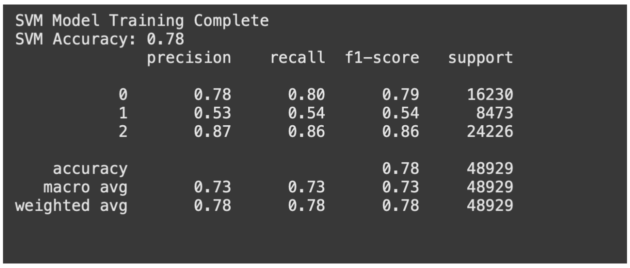
Models	Accuracy(%)
Logistic Regression	76.5
Support Vector Machines	78.2

Fig. 12. Performance Comparison of the Models.



```
ogistic Regression Model Training Complete
ogistic Regression Accuracy: 0.76
              precision    recall  f1-score   support

           0       0.80      0.75      0.77     16230
           1       0.47      0.66      0.55      8473
           2       0.90      0.81      0.85     24226

    accuracy                           0.76     48929
   macro avg       0.72      0.74      0.73     48929
eighted avg       0.79      0.76      0.77     48929
```

Fig. 13. Accuracy of Logistic Regression Model.



```
SVM Model Training Complete
SVM Accuracy: 0.78
              precision    recall  f1-score   support

           0       0.78      0.80      0.79     16230
           1       0.53      0.54      0.54      8473
           2       0.87      0.86      0.86     24226

    accuracy                           0.78     48929
   macro avg       0.73      0.73      0.73     48929
weighted avg       0.78      0.78      0.78     48929
```

Fig. 14. Accuracy of Support Vector Machine Model.

VII. Results Analysis

The results indicate that SVM outperformed Logistic Regression in all evaluation metrics(Fig. 15), particularly in handling neutral sentiment classification. This can be attributed to SVM's ability to find the optimal hyperplane that maximizes class separation, making it more effective for high-dimensional text data. Logistic Regression, while computationally efficient, struggled with the classification of neutral sentiment due to overlapping features between neutral and positive reviews(Fig. 16).

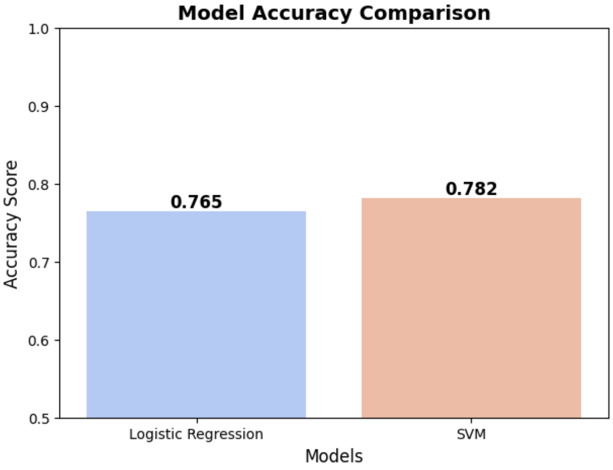


Fig. 15. Accuracy of the Models.

One of the primary reasons for the better performance of SVM(Fig. 14) is its ability to manage non-linearly separable data points using margin optimization. Text data, especially sentiment-based classification often exhibits complex decision boundaries where linear classifiers like Logistic Regression may not be sufficient.

The confusion matrix analysis revealed that Logistic Regression(Fig. 13) misclassified neutral reviews more frequently, labeling them as positive. This suggests that Logistic Regression is sensitive to class imbalances despite applying class-weight balancing techniques. On the other hand, SVM's margin-based approach handled neutral reviews more effectively, leading to improved recall and precision for this class.

Another contributing factor to SVM's superior performance is its robustness to high-dimensional sparse feature spaces as generated by TF-IDF. Textual data often contains a large number of irrelevant or weakly significant features, and SVM efficiently prioritizes the most critical features for classification.

While SVM provided better results, sentiment analysis remains challenging due to the nature and contextual ambiguity of human language. Future improvements can include attempting non-linear kernels, ensemble models, or deep learning-based models such as transformers to enhance accuracy and robustness.

In general, the analysis verifies that SVM is a better model for sentiment classification especially when handling imbalanced datasets where it is important to differentiate between neutral and positive sentiment.

VIII. Conclusion

The present study contrasted sentiment classification models of Amazon review products using Logistic Regression and Support Vector Machines (SVM). It has been observed from the results that SVM performs better compared to Logistic Regression on all the performance measurements, particularly identifying neutral sentiments. The greater performance of SVM is due to its ability to deal with high-dimensional feature spaces effectively and find the best decision boundaries.

The study identifies class imbalance problems of sentiment classification and demonstrates how methods of undersampling may be employed to render models fairer. However, the classification of neutral sentiment is a problem since the linguistic patterns that are shared across multiple sentiment classes provide a challenging feature.

Future research can explore the application of deep learning architectures, such as transformers (BERT), to further improve sentiment classification accuracy. Alternative feature extraction techniques, such as word embeddings, can be applied to augment text representations to improve classifications.

In conclusion, SVM is one of the best choices for sentiment classification applications, particularly when dealing with skewed datasets where distinguishing between neutral and positive sentiment is crucial. The findings of this study provide meaningful implications for firms that would like to apply sentiment analysis for the assessment of customer feedback and decision-making.

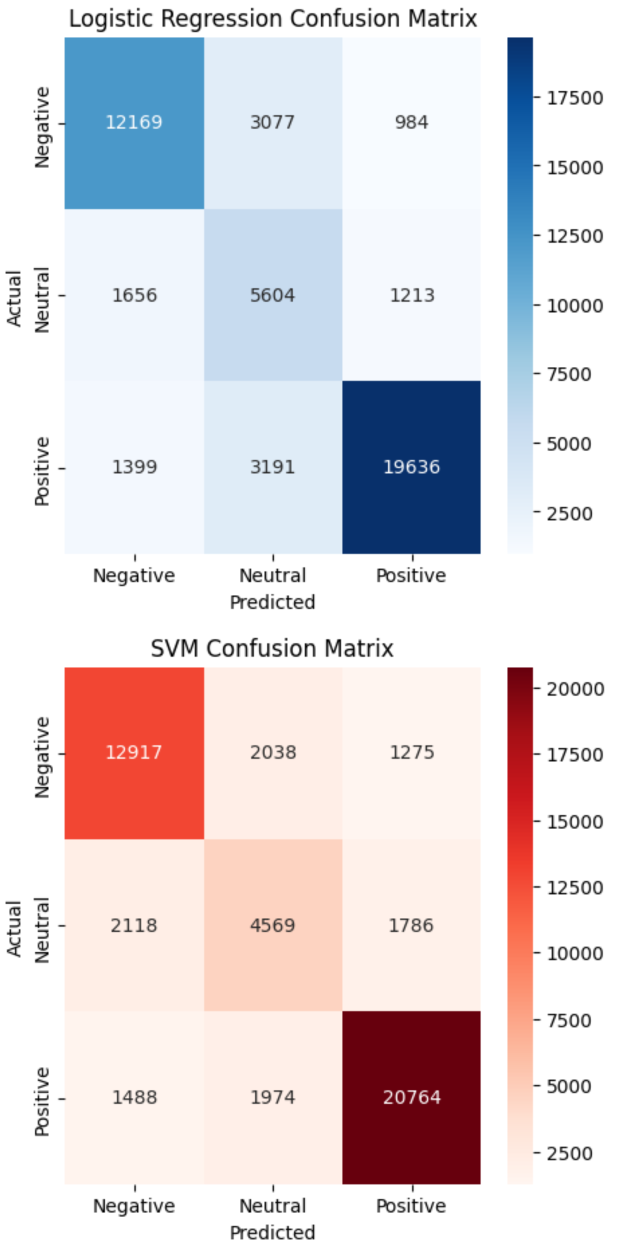


Fig. 16. Comparison of Logistic Regression and SVM wrt labels

References

[1] Pang, B., Lee, L., & Vaithyanathan, S. (2002). ”Thumbs up? Sentiment classification using machine learning techniques.” Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 79-86.

[2] Liu, B. (2012). ”Sentiment Analysis and Opinion Mining.” Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers.

[3] Japkowicz, N., & Stephen, S. (2002). ”The class imbalance problem: A systematic study.” Intelligent Data Analysis, vol. 6, no. 5, pp. 429-449.

[4] Wang, S., & Manning, C. D. (2012). ”Baselines and bigrams: Simple, good sentiment and topic classification.” Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 90-94.

[5] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). ”Recursive deep models for semantic compositionality over a sentiment treebank.” Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1631-1642.

[6] Zhang, Y., Zhang, J., & Zhao, H. (2020). ”BERT-based text classification: A survey.” Applied Sciences.

[7] Kaggle. ”Amazon Product Reviews Dataset.” URL:`https://www.kaggle.com/datasets/arhamrumi/amazon-product-reviews`

[8] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E. (2011). ”Scikit-learn: Machine Learning in Python.” Journal of Machine Learning Research, vol. 12, pp. 2825-2830. URL: `https://scikit-learn.org`

[9] Bird, S., Klein, E., & Loper, E. (2009). ”Natural Language Processing with Python.” O’Reilly Media. URL: `https://www.nltk.org`

[10] Mueller, A., ”WordCloud for Python.” URL: `https://github.com/amueller/word_cloud`

[11] Rehurek, R., & Sojka, P. (2010). ”Software Framework for Topic Modelling with Large Corpora.” Proceedings of LREC 2010 Workshop. URL: `https://`

radimrehurek.com/gensim/

[12] Chollet, F. (2015). "Keras: Deep Learning Library for Theano and TensorFlow." URL: https://keras.io/

[13] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., & Zheng, X. (2016). "TensorFlow: A System for Large-Scale Machine Learning." OSDI 2016. URL: https://www.tensorflow.org/

[14] Van Rossum, G., & Drake, F. L. (2009). "The Python Language Reference Manual." Python Software Foundation. URL: https://www.python.org/

[15] Honnibal, M., & Montani, I. (2017). "spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks, and Incremental Parsing." URL: https://spacy.io

[16] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). "SMOTE: Synthetic Minority Over-sampling Technique." Journal of Artificial Intelligence Research, 16, 321-357.

[17] Ramos, J. (2003). "Using TF-IDF to Determine Word Relevance in Document Queries." Proceedings of the First International Conference on Machine Learning.

[18] Powers, D. M. W. (2011). "Evaluation: From Precision, Recall, and F-Measure to ROC, Informedness, Markedness & Correlation." Journal of Machine Learning Technologies.

[19] McKinney, W. (2010). "Data Structures for Statistical Computing in Python." Proceedings of the 9th Python in Science Conference, pp. 51-56. URL: https://pandas.pydata.org