Sentiment Analysis for Amazon Product Reviews

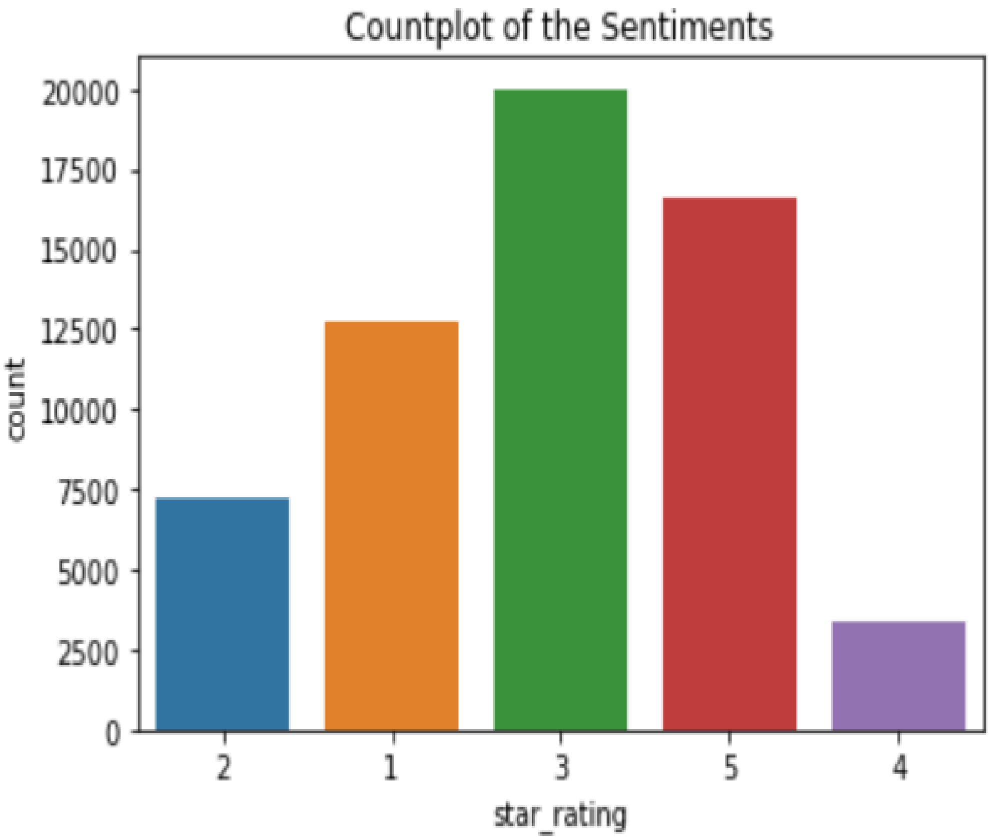
BY:

PRANITHA KOTLA

M15101752

Introduction

- Project Title: Sentiment Analysis for Amazon Product Reviews
- Objective: This project's goal is to create a sentiment analysis model that is especially suited for Amazon product reviews. By analyzing the sentiment stated in the ratings, it is expected to learn important details about consumer experiences, product preferences, and levels of general satisfaction.
- Dataset: The project utilizes an Amazon reviews dataset focused on Amazon products. The dataset has been sourced and downloaded for analysis purposes, providing a comprehensive collection of real customer reviews.
- Data Preparation: The project's initial step was to use the Pandas library to load the dataset. The foundation of the analysis was chosen to be relevant columns, such as star ratings and review content. A subset of 20,000 randomly chosen reviews from each star rating category was chosen in order to create a dataset of 60,000 reviews that would be balanced and would also address computational issues.



The graph shows the number of reviews for each rating respectively:

Reason:

- A random selection of data was made to create a balanced data sample.
- This decision to select a balanced data sample was driven by practical considerations and limitations.
- The dataset used for sentiment analysis consisted of a large number of reviews, and processing the entire dataset on the available system proved to be computationally demanding, leading to slow processing times and system performance issues.

Data Cleaning

- Data Cleaning processes: To enhance the quality and consistency of the collected data, a number of cleaning processes were taken. These activities included deleting non-alphabetical characters, expanding contractions to their full forms, changing all text to lowercase, and eliminating URLs and HTML tags from the review text.
- Average Length of Reviews: Character counts for the average length of the reviews before and after cleaning were calculated as part of the data cleaning procedure. This revealed information about how the cleaning procedures affected the length of the text in general.

Data Preprocessing and Feature extraction

- Preprocessing Steps: The Natural Language Toolkit (NLTK) was used for additional data preprocessing. Stop words, which are frequently used terms with minimal meaning for sentiment analysis, were eliminated as part of this process.
- Word2Vec: Word embeddings were produced using the pre-trained Word2Vec model in order to capture the semantic meaning of terms in the reviews. This model offers 300-dimensional word embeddings that may be used to represent the words in the reviews. It was trained on a sizable corpus of news articles.
- TF-IDF Feature Extraction: From the preprocessed text data, features were extracted using the TF-IDF method. Each word's relevance in the reviews is determined by the TF-IDF algorithm by taking into account both its frequency within a given review and across the entire dataset. This produced a numerical representation of the textual data that was ready for the sentiment analysis models to be trained.

Solution Narrative

Model Training and Evaluation

- Model Selection: Several models were trained and evaluated for sentiment analysis, including Perceptron, SVM, and RNN.
- Performance Metrics: The models were evaluated using various performance metrics, including accuracy, precision, recall, and F1-score. These metrics provide insights into the effectiveness of the models in accurately classifying the sentiment of Amazon product reviews.
- Comparison: A comparison was made between the different models used in the project (Perceptron, LinearSVC, RNN) to determine their respective accuracies and identify the most effective approach for sentiment analysis in this context.

Performance Metrics

	Precision	**Recall**	**F1- Score**
Perceptron	word2Vec- 67.08	word2VeC- 63.76	Word2VeC- 64.14
	TFIDF- 71.31	TFIDF- 71.54	TFIDF- 71.38
SVM	Word2VeC- 66.93	Word2VeC- 67.09	Word2VeC- 66.96
	TFIDF- 74.17	TFIDF- 74.38	TFIDF- 74.26

•The results clearly demonstrate that TF-IDF features outperform Word2Vec features by a significant margin.

•As a result, we can say that TF-IDF features are better suited for sentiment analysis as they take into account the frequency of occurrence of each word, whereas Word2Vec features average the vectors and may cause important words to lose their prominence, resulting in lower accuracy.

Model Comparison

Model	Accuracy
Perceptron using word2vec Perceptron using TFIDF	63.75 71.54
SVM using word2vec SVM using TFIDF	67.08 74.38
RNN	60.79

Conclusion

- With a precision of 74.17%, recall of 74.38%, and F1 score of 74.27%, SVM utilizing TF-IDF obtained the greatest accuracy of 74.38%. With balanced precision, recall, and F1 score, this model exhibits the highest overall performance in properly classifying feelings.
- Since the RNN model performs comparatively lower than the other models in this particular sentiment analysis job, it is possible that the sequential pattern of the reviews does not greatly affect sentiment categorization.
- In conclusion, the findings show how crucial feature extraction methods and classification algorithms are for sentiment analysis. The Word2Vec-based models and RNN fall short in reliably predicting sentiment, whereas the TF-IDF-based models (Perceptron and SVM) did better.

References/Bibliography

- **Steven Brownfield & Junxiu Zhou -Sentiment Analysis of Amazon Product Reviews.** **Software Engineering Perspectives in Intelligent Systems** pp 739–750
  - **Benchmarking my Results:** This research, similar to my project focuses on both traditional machine learning methods, such Support Vector Machines, as well as deep learning models such as Multilayer Perceptron (MLP) and Recurrent Neural Network (RNN). They are tested and compared in terms of their effectiveness.
  - For feature extraction, the research paper used bag of words and GloVe whereas I have used TFIDF and Word2Vec
- (https://radimrehurek.com/gensim/auto_examples/tutorials/run_word2vec.html)
- https://pytorch.org/tutorials/intermediate/char_rnn_classification_tutorial.html
- (https://pytorch.org/tutorials/intermediate/char_rnn_classification_tutorial.html)
- (https://www.youtube.com/watch?v=0_PgWWmauHk&ab_channel=PatrickLoeber)
- https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn
- (https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn)
- https://www.analyticsvidhya.com/blog/2022/01/tutorial-on-rnn-lstm-gru-with-implementation/
- (https://www.analyticsvidhya.com/blog/2022/01/tutorial-on-rnn-lstm-gru-with-implementation/)

Thank You!