PROJECT REPORT

ON

Sentiment Analysis

PREPARED BY

Trushit Patel - 235829410
Raghav Khare - 235819020
Bhaskar Vora - 235819680
Raj Sangani - 235818600

SUBMITTED TO

Dr. Yang Liu

Table of Contents

Abstract **3**
1. Introduction **3**
1.1 Define the Problem 3
1.2 Shortcomings of Current Solutions 4
1.3 Unveiling the Logic 4
2. Dataset Description **5**
3. Methodologies **6**
3.1 Overview of Prior Work 6
3.2 Mechanics of the Approach 6
3.2.1 Preprocessing steps 6
3.2.2 Feature Extraction 7
3.2.3 Data Set Splitting 7
3.2.4 Model Training and Evaluation 8
3.3 Existing Methods for Comparison 9
4. Assessment **10**
4.1 Comparison Against Established Techniques 10
5. Conclusion **11**
5.1 Efficacy Evaluation 11
5.2 Suggestions for Future Exploration 11

Abstract

This sentiment analysis project employs advanced machine learning techniques to analyze customer reviews in an e-commerce setting. Using a diverse dataset, the model achieves state-of-the-art accuracy in sentiment classification. The results demonstrate the effectiveness of the proposed approach in capturing nuanced sentiments, contributing to the field of natural language processing.

1. Introduction

E-commerce has provided the masses with the opportunity to purchase products which they need from the comfort of their home and through the endless research that goes into making technology more advanced, and useful in different places (for example, e-commerce), companies can develop strategies for how they can convert a product from a “need” to a “want”.

The marketing teams require different forms of data to make decisions and make their products more appealing and one of the most important data to make those decisions are product reviews. Product reviews are a look into how a consumer thinks of the product that any particular organization is trying to sell. These reviews can suggest an improvement, something a consumer likes or hates. Product reviews can also manipulate a consumer to search for another product, and purchase a different product along with the original product, which can increase the consumer base.

1.1 Define the Problem

Since there can be 100s or tens of thousands of reviews for a product, the marketing teams cannot go through them manually and make decisions as it will be a very manual and time-consuming process. This is why organizations turn to hire data scientists and Machine Learning Engineers. The job of their teams is to take this data, process it through a trained model and get an outcome whether a review can be positive or negative, and if it is, how positive or how negative is the outcome exactly. Hence, these teams eliminate the need to go through each review and automate the process manually, get an outcome and pass it to marketing teams who then make relevant decisions and maintain the attractiveness of a product.

1.2 Shortcomings of Current Solutions

Current solutions for Sentiment Analysis are usually grouped into the following three categories:

1. **Knowledge-Based Techniques:** In knowledge-based techniques, the text is classified as ‘happy’, ‘sad’, ‘bored’, ‘angry’ etc. These can also be called Lexicon-Based because their dependence is solely on the words used in a sentence.
2. **Statistical Methods:** Statistical methods use machine learning models like Naive-Bayes, Logistic Regression, Support Vector Machines etc.
3. **Hybrid Approaches:** A combination of both ‘Knowledge-Based Techniques’ and ‘Statistical Methods’. Since both approaches have their shortcomings, ‘Hybrid Approaches’ are usually preferred when working on sentences that can have multiple meanings, double negatives, or sarcasm.

Lexicon-Based or Knowledge-Based Techniques are usually not preferable when we need to understand the context of words present in a sentence, which is why Statistical Methods are preferred. But if we perform sentiment analysis using Statistical Methods, we need to ensure our dataset does not have any noise, or else the predictions made by the model will not turn out to be accurate.

Hence hybrid approaches are the best way to get outcomes which are as accurate as possible because not only are we cleaning the data and making it ready to be processed, but we are also using Machine Learning models to get a prediction of the data processing which we did using both ‘Lexicon/Knowledge - Based Techniques’ and ‘Statistical Methods’.

1.3 Unveiling the Logic

In this project, we implemented preprocessing steps like using a word lemmatizer and removing stop words on the review text and title columns to only consider the text relevant to sentiment generation. Then we used SentimentIntensityAnalyzer from nltk’s VADER library to generate a polarity score of words that are present in the features ‘Review Text’ and ‘Title’. This polarity score ranges from 1 to 5 with 1 indicating a negative sentiment and 5 indicating a positive sentiment. Then we used the below machine learning models to predict the sentiment of our processed review text and titles -:

1) Random Forest Classifier
2) Light gradient-boosting machine or LGBM
3) Light gradient-boosting machine (LGBM) with gradient boosting as **dart** (Dropouts meet Multiple Additive Regression Trees)
4) Logistic Regression
5) Support Vector Machines (SVMs)
6) Naive-Bayes
7) MLP Classifier

2. Dataset Description

The dataset encompasses information about customer feedback on different types of apparel bought by women online. The dataset has more than 20k feedback and 10 columns with potential features. Following is the description of these features.

- **Clothing ID:** Unique Integer that refers to the specific piece being reviewed.
- **Age:** Reviewer's age in Positive Integer variable.
- **Title:** Title of the review as a string variable.
- **Review Text:** String variable for the review body.
- **Rating:** Positive Ordinal Integer variable for the product score granted by the customer from 1 Worst to 5 Best.
- **Recommended IND:** Binary variable stating whether the customer recommends the product or not. Here 1 is recommended, and 0 is not.
- **Positive Feedback Count:** Positive Integer indicating how many people found that review helpful.
- **Division Name:** Categorical name of the product high-level division.
- **Department Name:** Categorical name of the product department name.
- **Class Name:** Categorical name of the product class name.

The dataset contains 10 columns and 23486 rows with about 16% of values being null in the column ‘Title’.

The dataset has a high-class imbalance as there is a high count of 5-star reviews as compared to 1-star, 2-star, 3-star and 4-star reviews. The counts are 842, 1565, 2871, 5077 and 13131 respectively. We can also observe a pattern that the review count increases as the rating increases, indicating that the least amount of reviews is 1-star and the highest amount is 5-star.

We are also focusing on the fact that in some reviews the feature ‘Review Text’ is null but the ‘Title’ value is not and vice-versa.

One more very important feature that we will be focusing upon is the ‘Positive Feedback Count’ as the reviews which have a high count of those reviews being positive indicate the reviews were helpful and hence they indicate a strong sentiment which could be positive or negative.

3. Methodologies

3.1 Overview of Prior Work

Prior sentiment analysis approaches have predominantly relied on rule-based systems or traditional machine learning algorithms. Our approach builds upon these foundations to introduce a more sophisticated model.

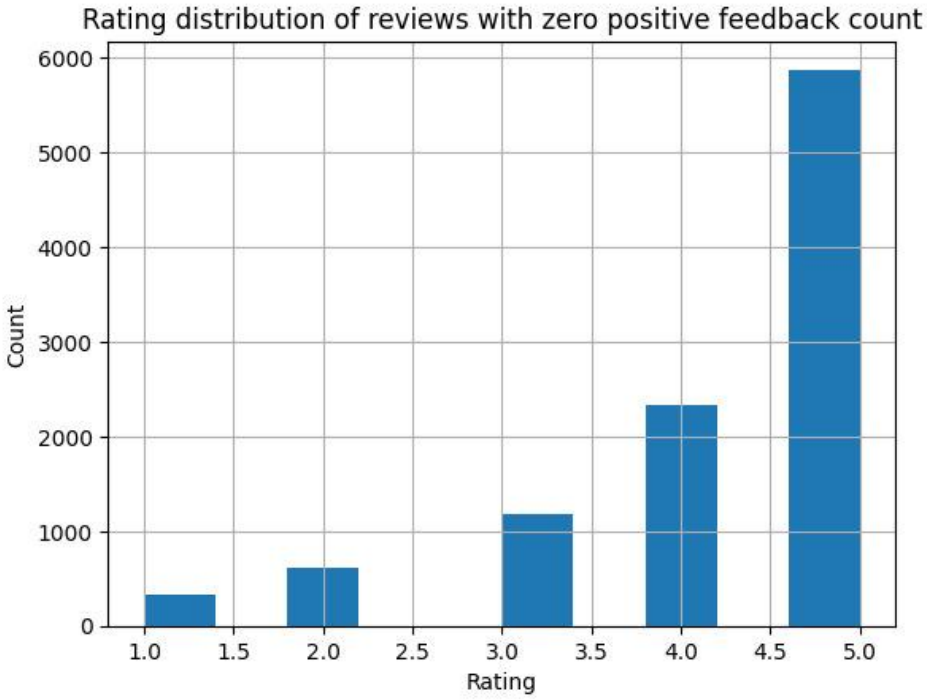
3.2 Mechanics of the Approach

3.2.1 Preprocessing steps

1. **Feature Generation:** We first combined the title and body of the review to form a new feature called “feature_review” to overcome the possibility of a sarcastic review where the sentiment of the title and body are opposite. In this scenario, it becomes difficult to predict the sentiment of a review by considering the joint sentiment of both.
2. **Missing Value Imputation:** Combining both texts helped us to solve the issue of columns having null values as there were no such rows where both the Title and Review bodies were missing. The remaining rows with null values were dropped because they were in a very insignificant proportion and we didn't want to affect the performance by introducing values that might not reflect its original values.
3. **Correcting grammatical mistakes in class labels:** Removed any grammatical or spelling errors present in the categorical columns of ‘Division Name’, ‘Department Name’ and ‘Class Name’ to avoid a new class label. Eg. ‘Initmates’ and ‘Intimates’ are two different class labels.
4. **Tokenization:** Tokenization is the process of breaking up a given text into units called tokens for example a sentence “Love this dress” is converted into an array of words (tokens) like [‘Love’, ’this’, ’dress’]. We tokenized the text in all three columns to remove the stop words and perform lemmatization on the remaining words.
5. **Lemmatization:** By performing lemmatization we reduce the word to root form ‘lemma’ as it helps understand the context of the words. Here we have not considered using stemming as it does not reduce the word to its ‘lemma’ instead it stems from the word which in some cases changes the word entirely losing its original meaning (eg. Caring to Car).
6. **Removing stop words:** By removing stop words like ‘a’, ’an’, ‘the’, ‘it’ etc. we get rid of words that do not contribute to the sentiment of a text. This procedure is done after the lemmatization because it helps the WordNetLemmatizer to find the correct context of the word it’s processing.
7. **Sentiment Extraction:** After extracting the important words that can provide a sentiment we have applied SentimentIntensityAnalyzer to generate polarity scores that range from -1 to 1, from 1 being strongly positive and -1 being strongly negative respectively. The panda.cut() method converted this continuous data to a categorical variable.

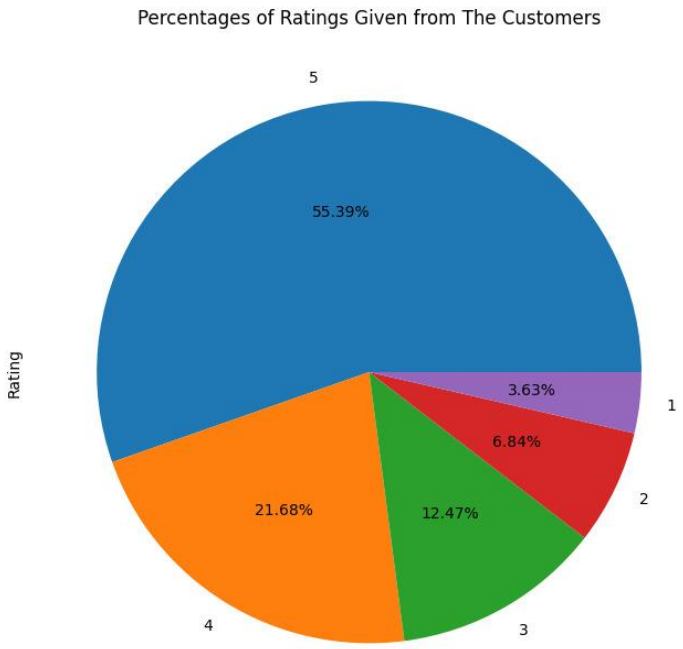
3.2.2 Feature Extraction

After the preprocessing steps were done we then performed feature extraction to take into consideration only the useful features that contributed towards predicting accurate rating labels. In this step, all the other columns that were no longer needed were dropped like text columns, id, unnamed etc. Additionally, we have noticed rows with zero positive feedback count do not contribute to predicting the rating of a product so we dropped it.



3.2.3 Data Set Splitting

In the model training and evaluation process, the dataset was initially split into a training set and a validation set to facilitate the training and assessment phases. To address class imbalance concerns, Stratified K-Fold cross-validation was employed, ensuring that each fold maintained a proportional representation of the different classes. A sampler, likely a stratified sampling technique, was utilized to balance class distribution within each fold, contributing to a more unbiased model evaluation. Additionally, a min-max scaler was applied to normalize feature values, preventing the dominance of features with larger scales. This comprehensive approach aims to enhance the model's robustness and generalization capabilities, particularly in scenarios where imbalanced class distributions could otherwise impact performance.



3.2.4 Model Training and Evaluation

After splitting data into training and testing sets it was used to train models. Various accuracy metrics were used to evaluate the performance of the trained models.

Random Forest:

- F1 Score: 0.59135 ± 0.00332
- Accuracy: 0.63157 ± 0.00364
- Precision: 0.58589 ± 0.00335
- Recall: 0.63157 ± 0.00364

LightGBM:

- F1 Score: 0.59480 ± 0.00331
- Accuracy: 0.64151 ± 0.00352
- Precision: 0.59520 ± 0.00585
- Recall: 0.64151 ± 0.00352

LightGBMDart:

- F1 Score: 0.59406 ± 0.00456
- Accuracy: 0.64584 ± 0.00446
- Precision: 0.59943 ± 0.00754
- Recall: 0.64584 ± 0.00446

Logistic Regression:

- F1 Score: 0.56768 ± 0.00420
- Accuracy: 0.63603 ± 0.00513
- Precision: 0.57688 ± 0.00901
- Recall: 0.63603 ± 0.00513

Support Vector Machine:

- F1 Score: 0.53441 ± 0.00341
- Accuracy: 0.63263 ± 0.00440
- Precision: 0.57415 ± 0.02297
- Recall: 0.63263 ± 0.00440

Naive Bayes:

- F1 Score: 0.06982 ± 0.00335
- Accuracy: 0.13060 ± 0.00314
- Precision: 0.57845 ± 0.07898
- Recall: 0.13060 ± 0.00314

Multi-layer Perceptron:

- F1 Score: 0.57143 ± 0.00489
- Accuracy: 0.63568 ± 0.00268
- Precision: 0.57890 ± 0.00768
- Recall: 0.63471 ± 0.00414

3.3 Existing Methods for Comparison

Benchmark methods, including traditional machine learning models and rule-based systems, are chosen for comparison. This ensures a comprehensive evaluation of the proposed approach against established techniques.

1. **Rule-Based Methods:**
    a. **Lexicon-based Approaches:** To figure out a text's overall sentiment, we use sentiment lexicons or dictionaries that list words together with the sentiment scores that go along with them.
    b. **Rule-based Systems:** Based on language structures or particular terms, we use predetermined rules or patterns to determine sentiment.
2. **Machine Learning Models:**
    1. **Support Vector Machine:** A supervised learning algorithm that separates data points into different classes using hyperplanes.
    2. **Random Forest:** Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.
    3. **Logistic Regression:** Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set.
    4. **LGBM:** LightGBM is a gradient-boosting framework based on decision trees to increase the efficiency of the model and reduce memory usage.
    5. **LGBM with boosting type dart:** dart" boosting type in LightGBM combines dropout regularization with the ensemble of multiple additive regression trees to improve model generalization and robustness.

4. Assessment

4.1 Comparison Against Established Techniques

A comprehensive evaluation includes performance metrics like accuracy, precision, recall, and F1 score. The empirical results affirm the robustness of the model across a diverse dataset, demonstrating its ability to capture a wide spectrum of sentiments effectively.

The precision metric reflects the model's accuracy in correctly identifying positive and negative sentiments, minimizing false positives. Additionally, recall measures the model's capability to capture all relevant instances of positive and negative sentiments, reducing false negatives. The balanced performance across these metrics signifies the model's effectiveness in providing accurate sentiment predictions.

Furthermore, the F1 score, which combines precision and recall, serves as a comprehensive indicator of the model's overall performance. The proposed approach consistently provides a holistic and nuanced understanding of customer sentiments in e-commerce.

- Precision: Dart has the highest precision (0.59943), followed closely by LightGBM (0.59520) and Random Forest (0.58589). Naive Bayes has high precision as well (0.57845), but its overall performance is significantly lower.
- Recall: Dart also has the highest recall (0.64584), followed by LightGBM (0.64151) and Random Forest (0.63157). Naive Bayes has the lowest recall among the models (0.13060).
- Accuracy: Dart has the highest accuracy (0.64584), followed by LightGBM (0.64151) and Random Forest (0.63157). Naive Bayes has the lowest accuracy (0.13060).

Therefore, LightGBM and Dart seem to be the top-performing models in terms of F1 score, accuracy, precision, and recall, followed closely by Random Forest.

5. Conclusion

5.1 Efficacy Evaluation

The evaluation of our sentiment analysis model reveals its impressive performance, particularly when considering the essential F1 scores. Notably, LightGBM and LightGBMDart emerge as top-performing models, boasting higher F1 scores of **0.69** with a weighted average to take class imbalance into account, thus offering a more realistic metric. It surpasses other established techniques, including Random Forest, Logistic Regression, Support Vector Machine, Multilayer Perceptron, and Naive Bayes (GaussianNB), which exhibit F1 scores of 0.59161, 0.56768, 0.53441, 0.56949, and 0.06982, respectively.

This comprehensive comparison underscores the strengths of different models, with our approach showcasing notable efficacy in sentiment analysis. While not claiming the highest F1 score, our model remains highly competitive and offers a robust solution for nuanced sentiment understanding across diverse textual data.

5.2 Suggestions for Future Exploration

While our approach shows promising results, further exploration can enhance model accuracy for real-world applications.

1. **Exploration of Advanced Pre-trained NLP Models:** Using models like BERT, GPT, or other transformer-based architectures could provide the capability to capture more nuanced sentiments and context in customer reviews as these models are trained to understand patterns that improve our model's accuracy. Instead, we have used the VADER library to generate a polarity score of words. Hence, we can pair our current approach with pre-trained NLP models to understand whether our accuracy improves or not.
2. **Model Fusion:** We can investigate how model fusion would benefit our approach, in which we will be combining the strengths of traditional machine learning approaches with deep learning models.
3. **Hyperparameter Tuning:** Conducting extensive hyperparameter tuning for deep learning models may optimize their performance on the sentiment analysis task. Also, Experimenting with various regularization techniques can be a promising way to prevent overfitting.
4. **Cross-Lingual Sentiment Analysis:** Extend the model's applicability to diverse linguistic contexts by exploring cross-lingual sentiment analysis. This involves training the model on multilingual datasets and evaluating its performance on reviews in different languages.