PRODUCT REVIEWS SENTIMENT ANALYSIS USING LSTM

Dinakar Jonnalagadda
Computer Science(MS) Michigan Technological University,
djonnala@mtu.edu

Nikhil Nandala Nandala
Computer Science(MS) Michigan Technological University,
snandala@mtu.edu

***Abstract*—Sentiment analysis is a technique used to identify the polarity of opinions expressed in text data. Long Short-Term Memory (LSTM) is a type of recurrent neural network that can capture the sequential dependencies in text data. In this project, we build and train an LSTM model for sentiment analysis and evaluate it on a dataset. Our initial approach was to use the random forest, but due to unsatisfactory results we used the LSTM, which outperformed random forest. The model gave best predictions when data is upsampled.**

I. INTRODUCTION

Natural language processing applications such as sentiment analysis are crucial for classifying text data as either positive, negative. Firstly the Random Forests method will be used in this project to create a sentiment analysis model utilizing a dataset of Amazon product reviews. Using the bag- of-words approach, we will preprocess the raw text data by cleaning, tokenizing, eliminating stop words, stemming the words, and turning them into numerical characteristics. To enhance the performance of the model, we will also investigate various feature engineering methods including n-grams and word embeddings.

Next We will be using a Long Short-Term Memory (LSTM) neural network for the sentiment analysis work in addition to Random Forests. A valuable tool for natural language processing tasks like sentiment analysis, LSTMs are a sort of recurrent neural network that can handle sequential input. We may learn more about the efficacy of various machine learning algorithms for sentiment analysis jobs by contrasting the performance of the Random Forests model with the LSTM model. The main goal of this project is to build a highly accurate sentiment analysis model for Amazon product reviews. By carefully analyzing both algorithms' performance. Using accuracy, precision, recall, and F1 score, we will assess the performance of both models.

The dataset for Amazon product reviews that was utilized in this study was downloaded from a source of publicly accessible datasets. Reviews from a range of categories, in cluding books, electronics, toys, and more, are included in the dataset. Each review contains information on the product, the individual who submitted it, and the review's content. There are 60,890 reviews altogether in the dataset. 30,000 reviews will be used to train our models, and 30,890 reviews will be used to evaluate the models. Based on the reviews' associated star ratings, they are classified as either good, negative, or . Before being utilized to train the models, the text data from the reviews is first preprocessed. To do this, non-alphabetic letters must be eliminated, the text must be changed to lowercase, the text must be tokenized into individual words, stop words must be eliminated, and the words must be stemmed. These steps aid in decreasing the dimensionality of the data and enhancing model accuracy.

For sentiment analysis, in addition to the Random Forests technique, we'll use an LSTM neural network. Recurrent neural networks of the LSTM variety are effective at handling sequential data, including text. The preprocessed text data will be used to train the LSTM model, which will then learn to predict the emotion of a particular review based on its input text.

The overall goal of this project is to analyze and choose between Random Forests and LSTM to build a highly accurate sentiment analysis model for Amazon product reviews. We may learn more about the efficiency of various machine learning algorithms for sentiment analysis jobs by analyzing how well these two models perform.

II. BACKGROUND

Sentiment analysis, commonly referred to as opinion mining, is the process of identifying and extracting subjective information from text data using machine learning and natural language processing techniques. Determine whether a piece of text has a good, negative, or emotional tone or attitude by using sentiment analysis. In a variety of industries, including marketing, customer service, and product development, sentiment analysis is helpful. Understanding client preferences, attitudes, and views about a company's goods or services may benefit a firm. Businesses may pinpoint areas for development, create new goods or services, and modify their marketing strategy by studying client feedback. Additionally, political research, brand reputation management, and social media monitoring may all benefit from sentiment analysis. Businesses may assess client sentiment toward their brand

and take the appropriate steps to remedy any bad comments or complaints by studying social media postings and internet reviews. A recurrent neural network (RNN) called LSTM (Long Short-Term Memory) is able to recognize long-term relationships in sequential input, including text. Since LSTM networks can analyze text at the word or sentence level while taking into consideration the context and meaning of the text, they are particularly helpful for sentiment analysis. LSTM networks may be customized for certain purposes like sentiment analysis and are trained on labeled data. Both LSTM and Random Forest have benefits and drawbacks, and some text and sentiment analysis jobs may favor one over the other. When assessing reviews or social media postings or other jobs where the context and meaning of the text matter, LSTM is very helpful. When assessing consumer feedback, for example, if the text data is high-dimensional and noisy, Random Forest is helpful.

Sentiment analysis has various applications in both business and research fields. Some of the major applications are: Customer feedback analysis: Companies can analyze customer feedback from various sources such as social media, customer surveys, and product reviews to understand customer sentiment towards their products or services. Brand monitoring: Sentiment analysis can be used to monitor brand reputation and track the sentiment of the brand across different channels. Companies can use sentiment analysis to understand the strengths and weaknesses of their products and make improvements accordingly.Sentiment analysis can be used to understand the consumer preferences in real-time. Moreover, it can be used in Political context to analyze public opinion on political candidates, issues, and policies. Sentiment analysis can be used to identify and prioritize customer complaints and feedback, and respond to them in effective manner. Sentiment analysis can be used to identify suspicious or fraudulent behavior in online transactions or social media activity. It can also be used to analyze patient feedback and improve healthcare services. Overall, sentiment analysis is a useful tool for companies and researchers alike due to its vast range of applications in several sectors and its ability to enhance decision-making and customer experience.

III. Motivation

The major objective of this project is to create a sentiment analysis model that properly classifies customer evaluations of a product as positive, negative, or utilizing the Random Forests and long short-term memory networks algorithms. A selection of Amazon product reviews for a certain item or group of items will make up the input dataset. To extract useful characteristics from the text data, several preprocessing techniques will be used, such as cleaning, tokenization, stemming, and stop-word removal.

A. PROBLEM STATEMENT

To Create an accessible sentiment analysis application that can be readily incorporated into the company's current workflows and provides decision-making actionable information.

Investigate the use of sentiment analysis in areas other than product reviews, such as social media evaluation, market analysis, and customer support.

B. METHODOLOGY

In this project,, Random Forest and LSTM were used to do sentiment analysis on Amazon product reviews. In order to prepare the data for machine learning algorithms, we first cleaned and transformed the text data into a numerical format. After preprocessing the data, we used the Random Forest method to categorize the reviews into good, negative, and categories. The model's accuracy was unsatisfactory, so we chose to investigate an alternative strategy utilizing LSTM.

LSTM is a deep learning method that can handle sequence data, which makes it appropriate for sentiment analysis and other natural language processing applications. On the basis of the preprocessed data, we trained an LSTM model and outper- formed the Random Forest model in terms of accuracy. Finally, we upsampled the minority classes to solve the issue of class imbalance in the dataset, which enhanced the performance of the LSTM model. Overall, our experiment showed how well deep learning methods, like LSTM, work for sentiment analysis jobs.

IV. Data Description

A total of 60,890 Amazon product reviews, each with five columns, make up the dataset utilized for this study.

Unique id - A unique identification number for each review in the dataset.

category: The classification of the item under assessment (e.g., electronics, apparel, or books).

Review text: The actual text that the consumer wrote while leaving a review.

rating: the customer's star rating of the product, which can range from 1 to 5.

Own rating: an assessment is either positive or negative.

The reviews, which range many different product categories,
were gathered from the Amazon website using web scraping techniques. With about 30% positive, 30% negative, and 40% ratings, the dataset's sentiment labels are balanced.

The dataset was divided into 30,890 reviews for testing and 30,000 reviews for training the Random Forests sentiment analysis algorithm. To ensure that the distribution of sentiment labels is consistent between the training and testing sets, the splitting was done at random.

V. RELATED WORK

A. LITERATURE REVIEW

Numerous research on data mining from diverse sources, such as Twitter, consumer comments, and product evaluations, have been undertaken in recent years.

Kumar et al., concentrated their research ”Opinion mining and sentiment analysis on online customer review” on mining reviews of three Amazon items, including the iPhone 5S, the Samsung J7, and the Redmi Note 3. Other researches have used Amazon reviews for a variety of objectives. For example, Rathor et al. (”Comparative study of machine learning approaches for Amazon reviews”)[2] used the Amazon API to get 21,500 Amazon reviews and randomly picked three thousand reviews for the experiment.

As per ”Sentiment analysis from product reviews using SentiWordNet as lexical resource”[3] a total of three hundred reviews of electronic devices from Amazon website are used, and Rodrigo et.al conducted their research over dst collected in specific products such as GPS, books, and cameras containing 2000 reviews (1000 positives and 1000 negatives) in each dataset (”Document-level sentiment classification: An empirical comparison between SVM and ANN”)[4].

As per ”An approach towards comprehensive sentimental data analysis and opinion mining” Pooja et. al suggest a complete approach for summarizing positive and negative qualities of products, laws, or policies by mining assessments from diverse sources, such as forums and debates. They exhibit the data using a variety of visualization tools. The emphasis in ”An approach towards feature specific opinion mining and sentimental analysis across e-commerce websites”[5] is on automating the process of collecting and assessing online evaluations for opinions expressed about specific features.The authors of ”Semantic Analysis and Implicit Target Extraction of Comments from E-commerce Websites”[6] concentrate on e- commerce websites and the remarks that buyers leave regarding the things they have purchased. They suggest a finergrained approach to concept mining that takes into account both explicit and implicit aspects.

Machine learning approaches and a lexicon-based approach are the two types of sentiment classification methods. In ”Opinion mining and sentiment analysis on online customer review”[1], the Nave Bayes classifier (NB) outperformed Logistic Regression (LR) and SentiWordNet in classifying reviews as positive or negative. The categorization methods’ performance was assessed using Recall, Precision, and F-measure. In “Classifying User Reviews at Sentence and Review Levels Utilizing Naive Bayes”[7], NB was run at two levels of granularity, sentence level and review level, and the relative frequency of each word was calculated using TF-IDF. Accuracy, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) were used to assess the experiment’s outcome at the review level.

B. PREVIOUS RESEARCH

Natural language processing’s well-established subject of sentiment analysis has been extensively used in a range of fields, such as social media, customer feedback, and product evaluations. Support Vector Machines (SVM), Naive Bayes, and Random Forests are just a few of the machine learning algorithms and methods that have been studied for application in sentiment analysis.

From the previous studies on using the Random Forests algorithm for sentiment analysis of Amazon product evaluations. The performance of the Random Forests, SVM, and Naive Bayes algorithms were contrasted using a dataset of over 30,000 reviews from various product categories. According to the study, Random Forests performed better than the other two algorithms with respect to accuracy and F1-score.

Convolutional Neural Networks (CNNs), in particular, were investigated in a different study to analyze sentiment in Amazon product evaluations. The study demonstrated that CNNs can perform better than well-established machine learning algorithms like Naive Bayes and SVM and reach high accuracy.

Several research have looked at the usage of various feature engineering methods to enhance the functionality of sentiment analysis models. To find the most useful characteristics for sentiment analysis of Amazon product evaluations, for instance, In a study the author employed feature selection strategies. The best performance, according to the authors, came from combining bag-of-words and TF-IDF characteristics.

VI. DATA VISUALIZATION

The display of data graphically or visually is known as data visualization. The ability to comprehend complicated data and effectively express conclusions makes it a crucial component of data analysis. Finding patterns, trends, and correlations that can be hard to see in a tabular or numerical format is feasible using data visualization. Data visualization techniques can be done using bar charts, pie charts, circlify, scatter plots, line graphs, heat maps, and histograms, to name a few. Each visualization methodology has advantages and disadvantages of its own, and the technique to be used relies on the data being studied and the insights that need to be conveyed. In order to understand the data and find trends, we employed data visualization tools in our project. To better visualize the class imbalance in the dataset, we displayed the sentiment label distribution. We needed to upsample the minority class, therefore we utilized bar charts to display the frequency of each sentiment category. This guided our feature selection approach and allowed us to determine the key traits for predicting sentiment. We used data visualization tools in our sentiment analysis research to better comprehend our dataset and communicate our results. Bar charts and circlify were the main graphics we utilized.

We utilized bar charts to show how the sentiment classifications in our dataset were distributed. On the y-axis, we displayed the number of reviews that fell into each sentiment class (positive, , and negative), and on the x-axis, we plotted the sentiment class label. Each bar's height represented how many reviews there were for that class. With the use of this kind of display, we were able to identify right away how unbalanced our dataset was—positive evaluations outnumbered negative ones by a wide margin. The relative amount of reviews in each sentiment category might also be easily compared thanks to this.

A data visualization package called "Circlify" is used to display hierarchical data in a circular fashion. In our project, Circlify was utilized to visually represent the top 100 goods according to the quantity of client feedback. We choose Circlify because it made it possible for us to present the items in a way that was aesthetically appealing and communicated their relative value and popularity. According to the amount of customer reviews for each product, the size of each circle in the Circlify graphic denotes the quantity of those reviews. The items are placed in a hierarchical manner, with the items with the 100 most reviews with the largest and the items with the fewest reviews followed by it. As we approach further from the center, the circle sizes get smaller, which denotes a decline in the quantity of product reviews.

VII. PREPROCESSING PROCEDURE

Preprocessing data is crucial because raw data may be formatted inconsistently or incompletely. Preprocessing raw data effectively can boost its correctness, which can raise project quality and reliability.

It increases dependability and accuracy. Preprocessing data may increase the correctness and quality of a dataset, making it more dependable by removing missing or inconsistent data values brought on by human or machine mistake. It ensures consistency in data. Data duplication is a possibility while collecting data, and removing them during preprocessing can guarantee that the data values for analysis are consistent, assisting in the production of reliable results. It improves the performance of algorithmic on the data. The quality of the data is improved by preprocessing, which also makes it simpler for machine learning algorithms to read, utilize, and analyze the data.

For the purpose of our project, we performed various preprocessing steps on the 'Review_text and own_rating'. On Review_text we performed various steps like removing punctuation, numbers and special characters. Additionally, we performed steps like removing square brackets, removing stop words as these serve no useful purpose for the sentiment analysis.

For LSTM specific processing we performed pad sequences on the train data and transformed the 'Own Rating' column to numerical values i.e 0's and 1's.(positive=1 and negative=0).

Moreover, on the 'Own_Rating' column we performed steps like removing rows with rating as they are no use for our task.

VIII. EXPERIMENTAL RESULTS

We tested several tree counts and depths for the Random Forest method. The density of the trees dictates how many splits there are in the decision tree, and the number of trees defines how many decision trees there are in the forest. The accuracy of the model may be significantly impacted by these hyperparameters. In the Random Forest algorithm, the depth and number of trees can have a significant impact on the performance of the model. The depth of a decision tree determines the number of splits that can be made, which affects the complexity of the mode either positively or negatively. A tree with more depth can fit the training data more closely, but it may also result in overfitting, which results in poor performance on new, unseen data. On the other hand, a shallower tree may not capture all the relevant features of the data, resulting in underfitting. In the same way, the variance and bias of a Random Forest ensemble are influenced by the number of trees in the ensemble. The model's stability and variance can both be improved by adding more trees, although doing so can cost more to compute. Beyond a certain point, increasing the number of trees may not further improve the performance of the model and may even result in overfitting.

When our random Forest algorithm tested with depth of 15 and number of trees equal to 300. The accuracy is around 81. Upon Decreasing the depth to 10 and increasing the trees to 500 the accuracy was around 83 and showed no further improvement upon tuning the parameters. Additionally, we tried with different normalizations like l1&l2.

In the case of LSTM, we experimented with different numbers of layers. The number of layers determines the number of LSTM cells in the network. Increasing the number of layers can allow for the model to learn more complex representations of the input data, but it can also increase the risk of overfitting. We also experimented with and without regularization in the LSTM model. Regularization is a technique used to prevent overfitting by adding a penalty term to the loss function. In our case, we used an l1 regularization term. We found that the model performed better with the regularization term included. It was found that LSTM produced superior outcomes versus Random Forest. This could be explained by the fact that the LSTM deep learning algorithm was created primarily to handle sequential data, such as text. By using its internal memory system, LSTM has the capacity to record the sequential dependencies in the text data. As a result, it can develop the ability to recognize the context and connections among the words in a phrase, which is crucial for correctly anticipating sentiment.

Without upsampling LSTM is overfitting the data, but with upsampling LSTM is able to predict the review correctly.

The standard machine learning method Random Forest, on the other hand, works by building a lot of decision trees and merging the results. It can also handle text data, however because it uses a bag-of-words representation of the text, it might not be as good at capturing the sequential relationships in the data as LSTM. Hyperparameters like the number of trees and depth of the trees can also affect how well Random Forest performs. It might be difficult and time- consuming to find the ideal hyperparameters. The optimization process could be made simpler by the fact that LSTM needs setting fewer hyperparameters, such as the number of layers and the learning rate.

IX ALGORITHMS AND PROCEDURES

The initial idea of our implementation is to build the sentiment prediction model using randomforest. But upon training the model and getting the accuracy, which is not satisfactory, we implemented model using LSTM algorithm and got the satisfactory results. In the following paragraphs we discuss the approach we used like with and without upsampling and various parameters tuning along with a brief on data upsampling.

When dealing with unbalanced datasets, upsampling is a strategy used in data analysis. When there are considerably more instances of one class than the other, the dataset is said to be unbalanced. In such cases, machine learning algorithms may be biased in favor of the majority class, resulting in poor performance for the minority class. By raising the proportion of instances in the minority class, upsampling is a strategy used to overcome this problem and provide a balanced dataset. To raise the minority class's representation in the dataset, the procedure entails reproducing existing instances of the minority class or creating new syn- thetic examples. Upsampling can be done in a variety of ways, such as randomly, using SMOTE (Synthetic Minority Over- sampling Technique), or using ADASYN (Adaptive Synthetic Sampling). In order to create a balanced dataset, random oversampling replicates instances of the minority class at random. ADASYN concentrates on areas of the feature space where the class distribution is sparser and produces synthetic examples of the minority class. The method creates new instances in lower- density areas by computing the density distribution of the minority class instances. This method is very beneficial for datasets with intricate class distributions. Upsampling can enhance the efficiency of machine learning algorithms on unbalanced datasets and lessen bias toward the majority class, among other advantages. Additionally, the method can aid in locating examples of the minority class that are particularly instructive, improving generalization and performance on untested data. It is crucial to remember that upsampling can occasionally have disadvantages. The risk of overfitting is one of the key problems, especially when utilizing random oversampling. Using more advanced techniques, like SMOTE or ADASYN, which create synthetic instances based on the minority class's underlying distribution, can help to overcome this.

In our project, we Upsampled the minority class i.e the negative instances and shown the weights of both positive and negative instances. Both the classes have the same weight.

In our study, we analyzed the performance of two well-known algorithms, Random Forest and LSTM, to do sentiment analysis on customer reviews of a product. The most precise classification of the evaluations as good, negative was what we aimed for. First, we tested several tree counts and depths in Random Forest, but we were unable to get the needed accuracy. With upsampled data, we were able to achieve an accuracy of up to 87%, but not with non-upsampled data. The depth and number of trees in the ensemble have a significant impact on the accuracy of the Random Forest method, which builds an ensemble of decision trees. A more accurate model is often produced by an ensemble of multiple trees, but this also increases computing complexity and time. Similar to decreasing the depth of the trees, increasing it can improve accuracy but also increase the risk of overfitting.

The LSTM, a kind of recurrent neural network that can simulate sequential input, was the subject of our next experiment. A 2-layer LSTM model with l1 regularization achieved the greatest accuracy, at about 90%, out of all the layer counts and regularization techniques we examined. Our final model comprised two levels, with 32 and 64 units, respectively, on each level. In contrast to Random Forest, LSTM discovers patterns in the data by processing information across a number of layers rather than relying on a predetermined number of decision trees. But, because of the class imbalance though the accuracy is higher the model overfitted and given the wrong output.

Additionally, we tried out upsampling the minority class in our dataset, which was made up of reviews with a bad attitude. Negative reviews are frequently less frequent than positive which creates a class imbalance problem in sentiment analysis. We discovered that upsampling the negative class increased the precision of our LSTM model, yielding a precision of over 92%. In order to achieve a balance between the proportion of positive, negative in the dataset, upsampling entails replicating cases of the minority class. In conclusion, we discovered that in our sentiment analysis study, LSTM performed better than Random Forest. In comparison to Random Forest, LSTM proved more accurate in modeling sequential patterns in the data. Additionally, we discovered that increasing the sample size of the minority class in our dataset addressed difficulties with class inequality and enhanced the precision of our LSTM model.

The various versions of algorithms and outputs are as follows:

Without Upsampling :

Random Forest: f1_score :83.3

Input: bad bad maintenance, bad food, good atmosphere.

Output:Positive; Desired output: Negative .

LSTM: f1_score:90.96

Input: Bad quality, Bad Service

Ouput: Positive, Desired Output: Negative (Overfitted).

With Upsampling :

RandomForest: f1_score: 86

Input: good maintenance, bad food, Output: Positive.

Desired Output: Negative

LSTM : f1_score : on positives :0.97, on negatives: 0.86

Input:Good look , bad quality, Output: Negative

Desired Output: Negative.

X. Contributions

Member 1: Dinakar Jonnalagadda
Member 2: Nikhil Nandala.

Member 1:

1) Data Preparation: handling missing values and duplication, feature engineering, and dataset collection and cleaning managing imbalances, tokenization.
2) Model Training: setting up the hyperparameters, implementing the Random Forest algorithm with Scikit-Learn and lstm algorithm, and training the model with the training data.
3) Model Evaluation: Evaluating the performance of the trained model using evaluation metrics such as accuracy, precision, recall, F1 score, and confusion matrix.

Member 2:

1) Data Splitting: To divide the dataset into training and testing sets will ensure that the class distribution is retained in both sets.
2) Model Tuning: Fine-tuning the hyperparameters of the Random Forest algorithm using grid search or randomized search to improve the performance of the model.
3) Prediction: Making predictions on the test set using the trained and tuned model and computing the evaluation metrics on the predictions.

REFERENCES

1) K. S. a. D. J. a. M. J. Kumar, “Opinion mining and sentiment analysis on online customer review,” in IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Chennai, 2016.

2) A. S. a. A. A. a. D. P. Rathor, “Comparative study of machine learning approaches for Amazon reviews,” Procedia computer science, vol. 132, pp. 1552–1561, 2018.

3) A. a. S. V. a. M. B. ernian, “Sentiment analysis from product reviews using SentiWordNet as lexical resource,” in 2015 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, 2015.

4) J. F. V. W. P. G. N. Rodrigo Moraes, “Document-level sentiment classification: An empirical comparison between SVM and ANN,” Expert Systems with Applications, vol. 40, pp. 621–633, 2013.

5) Prashast Kumar, ArjitSachdeva, “An approach towards feature specific opinion mining and sentimental analysis across e-commerce websites”. 5th International Conference- Confluence The Next Generation Information Technology Summit (Confluence), 2014.

6) Hui Song, Jianfeng Chu, Yun Hu, Xiaoqiang Liu, “Semantic Analysis and Implicit Target Extraction of Comments from E-commerce Websites”. Fourth World Congress on Soft- ware Engineering, 2013.

7) Y. a. K. V. Saito, “Classifying User Reviews at Sentence and Review Levels Utilizing Na¨ıve Bayes,” in 21st Inter- national Conference on Advanced Communication Technol- ogy (ICACT), PyeongChang Kwangwoon Do, Korea (South), 2019.