

# *Product reviews sentiment analysis using random forest.*

Dinakar Jonnalagadda  
Computer Science(MS)  
Michigan Technological  
University

Nihil Nandala Nandala  
Computer science(MS)  
Michigan Technological  
University

**Keywords**—*Preprocessing, class imbalance, logistic regression, random forest, Support Vector Machine, confusion matrix*

## I. PROJECT GOAL

The objective of this project is to create a sentiment analysis model that uses logistic regression to categorize Amazon product reviews as positive, negative, or neutral. An input dataset of Amazon product reviews will be used for the project, and the text data will be preprocessed to extract features that may be applied to classification. A labeled subset of the data will be used to train the logistic regression model, and a holdout test set will be used to assess the model's performance. The performance of the model, including measures like accuracy, precision, recall, and F1 score, will also be examined as part of the study. The final output will be a sentiment analysis model that can accurately categorize Amazon product reviews as positive, negative, or neutral.

## II. BACKGROUND DESCRIPTION

The project "sentiment analysis using logistic regression" seeks to identify and categorize people's reviews as they are reflected in text data. The motivation behind this research stems from the growing significance of comprehending and interpreting the sentiment underlying huge volumes of text data in a variety of contexts, including social media, consumer feedback, and product evaluations.

Based on the sentiment or emotion expressed in the text, sentiment analysis divides text data into many groups. This designation may be neutral, negative, or positive. Random forest is one of the methods for sentiment analysis that is most frequently employed. A popular machine learning model for classification issues is random forest, which makes it a good option for sentiment analysis jobs.

## III. DATASET DESCRIPTION

The dataset contains a total of 60890 amazon product reviews, which will be split into 30000 sets for training the model and

30890 sets for testing the model. The dataset has 5 columns namely unique\_id, category, review\_text, rating, own\_rating

## IV. SUCCESS EVALUATION CRITERIA

The evaluation criteria for the sentiment analysis on amazon product reviews involves accuracy notation, confusion matrix, precision.

The accuracy denotes the percentage of correctly predicted instances. Since, accuracy alone isn't enough to evaluate the model, especially in case of class imbalance, we use precision. Precision is used to calculate ratio between true positives and total number of true positive predictions. On the other hand we also use confusion matrix, which shows the number of true positives, true negatives, false positives, false negatives.

## V. PAPERS TO READ

In order to familiarize more with the problem, we take for this project and its objectives one can refer to the following papers:

1) "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. Proceedings of the 12th International Conference on World Wide Web, 519-528". In this paper researchers proposed a Machine learning approach to classify product reviews as positive, negative using support vector machines(SVM).

2) "Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 168-177". In this paper researchers provided a useful

approach to summarize customer reviews using text mining techniques.

In order to get to know the problem of class imbalance the following papers can be helpful.

1) "Addressing the curse of imbalanced training sets: One-sided selection. Proceedings of the 14th International Conference on Machine Learning, 179-186".

## VI MACHINE LEARNING ALGORITHM TO USE

For the proposed project the task is to train the machine learning model on the text data, the reviews of various products, and correctly classify the review as positive, negative or neutral. In order to achieve this task there are various algorithms such as Naïve bayes classifier, logistic regression, support vector machines, random forest.

Which one to use? From the available algorithms one that finds feasible is the "random forest algorithm". The naïve bayes algorithm basically predicts on assumptions and doesn't perform well on large datasets and non linear data and also in the situation of class imbalance the classifier will be more inclined to the majority class and can lead to misclassification. Logistic regression, basically binary classification, is not well suited for handling class imbalance. The random forest works well for class imbalance and can handle class imbalance. Hence, the Random forest algorithm is feasible for this task

## VII.POSSIBLE DELIVERABLES

The deliverables for the proposed project will be percentage of accuracy the ratio between true positives plus true negatives and total number of predictions multiplied by 100. A confusion matrix to give a better overview of the true positives, true negatives, false positives, false negatives. And the class label for the given input review such as, positive, negative, neutral.

## VIII.TEAM MEMBERS AND WORK DIVISION

Team Members: Dinakar Jonnalagadda, [djonнала@mtu.edu](mailto:djonнала@mtu.edu)

Nikhil Nandala Nandala, [snandala@mtu.edu](mailto:snandala@mtu.edu)

### *Work Assignment:*

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 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: Using stratified sampling 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.

## IX. MILE STONES

1)Data Collection

2)Preprocessing

3)Model training, parameter tuning

4) prediction

5)Model evaluation

6)Model improvement.

7)Project presentation