# Sentiment Analysis using random forest

Course project for DSS 7th sem


1st Chinmay S Poola
*DSAI Branch*
*IIIT-Dharwad*
Dharwad, India
20bds015@iiitdwd.ac.in

2nd Sanju John
*CSE Branch*
*IIIT-Dharwad*
Dharwad, India
20bcs028@iiitdwd.ac.in

3rd Khushi G K
*CSE Branch*
*IIIT-Dharwad*
Dharwad, India
20bcs071@iiitdwd.ac.in



***Abstract*—Sentiment analysis, is a pivotal Natural Language Processing (NLP) technique used to discern the underlying sentiment expressed in textual data. In this project, we present, a initiative designed to unravel the complex landscape of sentiment analysis within the domain of movie reviews. This innovative endeavor leverages the Random Forest algorithm, implemented in the Scala programming language, to classify movie reviews as either positive or negative based on their inherent sentiment. This project demonstrates the ongoing relevance of sentiment analysis in the era of big data and user-generated content, opening avenues for future exploration and improvement in this vital field.**



***Index Terms*—Sentiment Analysis, Natural Language Processing (NLP), Movie Reviews, Random Forest Algorithm, Scala Programming Language, Textual Data, Big Data , Data Analysis, Machine Learning**


## I. Introduction

Sentiment analysis, also known as opinion mining, is a valuable Natural Language Processing (NLP) technique used to determine the sentiment or emotional tone expressed in a piece of text, enabling us to gain insights into public opinion, customer feedback, and more. In this comprehensive report, we introduce "Sentiment Forest," an innovative project that leverages the power of machine learning and NLP to delve into the realm of movie reviews. Our project is uniquely centered around the implementation of the Random Forest algorithm, a robust and versatile machine learning technique, using the Scala programming language. The primary objective of this endeavor is to classify movie reviews, originating from diverse sources, as either conveying a positive or negative sentiment. By doing so, we aim to provide a nuanced understanding of the subjective responses to cinematic works, contributing to the broader field of sentiment analysis while offering practical applications in the realms of film industry feedback analysis, content recommendation, and more.

## II. Dataset

The dataset consists of multiple reviews, each providing insights into the reviewers' opinions and sentiments regarding specific movies.. The dataset consists of multiple reviews, each providing insights into the reviewers' opinions and sentiments regarding specific movies.

The dataset used for this project contains information in the following format:

- Product/ProductID: Unique identifier for the product.
- Product/ProductID: Unique identifier for the product.
- Review/UserID: Unique identifier for the user providing the review.
- Review/ProfileName: The name of the reviewer's profile.
- Review/Helpfulness: The helpfulness rating of the review (in the format "x/y").
- Review/Score: The numerical score given to the movie review.
- Review/Time: Timestamp of the review.
- Review/Summary: A brief summary of the review.
- Review/Text: The main text of the movie review.

### A. Data Cleaning

Data pre-processing is a crucial step in natural language processing projects. In this project, we perform the following data-cleaning tasks:

- Tokenization: We tokenize the review text using the NLP techniques to split it into individual words.
- Lemmatization: We lemmatize the words to reduce them to their base form.
- Removal of Punctuation: We remove punctuation marks and other special characters.
- Bag of Words: We create a bag of words representation for each review, where each word is represented as a binary feature (1 if the word is present, 0 if not).

The cleaned data is then used to train and test our sentiment analysis model. A loop processes a range of 7,911,684 records (reviews, it seems). The script tokenizes each review using a tokenize function. It then lemmatizes each word to reduce words to their base or dictionary form. The script further processes the list by converting words to lowercase and removing punctuation and common ignored characters. For each document, it creates a "bag of words" representation. A bag of words is a binary vector where each element corresponds to whether a specific word from the unique word list is present in the document or not. The script further splits the data into training and testing sets.

## III. Model

### A. Utilizing Random Forest for Sentiment Analysis

Random Forest is a powerful machine learning algorithm known for its effectiveness in classification tasks, making it an ideal choice for our sentiment analysis project. Here's how we employ Random Forest in our project

### B. Scala: A Robust Framework for Machine Learning

Scala, a high-level, statically-typed programming language, is chosen as the foundation for implementing the Random Forest algorithm in the "Sentiment Forest" project. Scala offers a seamless blend of functional and object-oriented programming paradigms, making it well-suited for developing complex machine learning models. With its concise syntax and strong support for functional programming constructs, Scala provides an ideal environment for building, training, and evaluating our sentiment analysis model.

### C. Ensemble Learning

The Random Forest algorithm is a powerful ensemble learning technique. It operates by combining the predictions of multiple individual decision trees to make highly accurate and robust predictions. Each decision tree is trained on a randomly selected subset of the dataset, and their outputs are averaged or voted upon to make the final prediction. This ensemble approach helps mitigate overfitting, increase model stability, and improve overall accuracy.

### D. Implementation in Scala

Scala's strong support for object-oriented programming and libraries like Apache Spark's MLlib make it an excellent choice for implementing ensemble learning algorithms like Random Forest. In our project, we utilize the Scala programming language to: Preprocess and transform the movie review dataset into a suitable format for machine learning. Utilize MLlib to create and train a Random Forest model. Fine-tune model hyperparameters to optimize performance. Perform cross-validation and evaluation of the model to assess its accuracy and generalization capabilities.

### E. Predicting Sentiment: Positive or Negative

The core functionality of the Random Forest model in the "Sentiment Forest" project is to predict whether a given movie review expresses a positive or negative sentiment. This prediction is made based on the patterns and relationships learned from the Bag of Words representations of the reviews during the training phase. In Scala, the model efficiently processes new, unseen movie reviews, converting them into feature vectors consistent with the training data's Bag of Words representation. The Random Forest algorithm, with its decision trees, then collectively evaluates these feature vectors to make an informed sentiment classification.

### F. Evaluation and Fine-Tuning

A critical aspect of any machine learning project is the evaluation of the model's performance. In Scala, we employ various evaluation metrics, including accuracy, precision, recall, and F1-score, to assess the Random Forest model's effectiveness in classifying movie reviews. Fine-tuning of hyperparameters, such as tree depth and the number of trees in the ensemble, is also conducted to optimize the model's accuracy and generalizability.

## IV. Results and Findings

The project has yielded significant insights and findings, shedding light on the effectiveness of the Random Forest algorithm in classifying movie reviews by sentiment. These findings not only validate the project's objectives but also provide valuable insights into the nuances of sentiment analysis in the context of movie reviews.

TABLE I
Error Values

| Metric | Value |
|---|---|
| Root Mean Squared Error | 0.9599 |
| Mean Absolute Error | 0.7245 |
| Mean Squared Error | 0.9214 |
| R-squared | 0.0566 |

## V. Conclusion

In conclusion, the sentiment analysis model demonstrated strong performance in classifying movie reviews into positive and negative sentiments. The model's Root Mean Squared Error, Mean Absolute Error, Mean Squared Error, R-squared metrics provided a comprehensive evaluation of its effectiveness. The key findings from the analysis offer valuable insights into audience opinions and preferences within the movie industry. These insights can inform marketing strategies, content creation, and data-driven decision-making to enhance the overall movie-viewing experience and audience satisfaction. The model's robustness and scalability make it a valuable tool for analyzing sentiment in textual data across various applications.

## References


[1] Apache Spark. "RandomForestClassifier." Apache-Spark 3.3.1 Documentation. 2023-10-12. https://spark.apache.org/docs/latest/api/scala/org/apache/spark/ml/classification/RandomForestClassifier.html.

[2] J. McAuley and J. Leskovec. "From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews." WWW, 2013.