Multiclass Classification Using Sentiment Analysis

Simarpreet Singh Ghumman  
Reg No – 21096214

*Abstract* - This project delves into sentiment analysis using the SKlearn implementation of supervised learning algorithms, specifically Decision Tree, Random Forest, and Support Vector Machine (SVM). The study employs a Bag of Words approach, utilizing the TfidfVectorizer to extract feature vectors from Amazon phone reviews. The dataset, centered on user-generated content, encompasses sentiments categorized as positive, negative, or neutral. Through meticulous data preprocessing, feature extraction, and the subsequent training and evaluation of the selected algorithms, the project aims to discern the nuanced performance of these models in sentiment classification. Key metrics such as accuracy, precision, recall, and F1 score are employed to compare and contrast the outcomes of each algorithm. The investigation not only contributes insights into the efficacy of Decision Tree, Random Forest, and SVM in sentiment analysis but also sheds light on their applicability in the context of Amazon phone reviews, thus providing valuable guidance for future endeavors in this domain.

Keywords - *Sentiment Analysis, Decision Trees, Random Forest, SVM, Text Classification.*

# **Introduction**

In the realm of natural language processing, sentiment analysis stands as a critical frontier, deciphering the underlying emotions embedded in textual expressions. This project navigates this dynamic landscape, deploying machine learning algorithms to dissect sentiments within Amazon phone reviews. We focus on three pivotal algorithms - Decision Tree, Random Forest, and Support Vector Machine (SVM), utilizing the SKlearn implementation to unravel the intricate tapestry of user opinions.

The foundation of our approach lies in the Bag of Words methodology. This transformative technique converts textual nuances into numerical feature vectors, paving the way for the application of supervised learning algorithms. As we embark on this journey, it is imperative to draw insights from key research papers that have shaped the discourse in sentiment analysis.

Pang and Lee's [1] [2] extensive review on sentiment analysis methodologies provides a compass for our exploration, offering a nuanced understanding of the evolution of sentiment analysis techniques.

The algorithmic pillars of our study find roots in seminal works such as Leo Breiman's [3][4] exploration of Random Forests, elucidating the power of ensemble learning principles. Likewise, the tutorial by Burges [5] serves as a beacon in comprehending the intricate principles of Classification and Regression fundamentals, a cornerstone of my analysis.

Furthermore, I draw from the foundational documentation of scikit-learn, a library that underpins our implementation, encapsulating a plethora of machine learning algorithms, including the ones central to this study.

# **Literature Review**

## **Supervised Learning**

## Supervised learning is a foundational paradigm within the field of machine learning, characterized by its reliance on labelled training data to make predictions or decisions. In this learning approach, a model is trained on a dataset where each input is paired with the corresponding correct output or target label. The primary objective is for the model to learn the underlying patterns, relationships, or mappings between inputs and outputs, enabling it to generalize and make accurate predictions on unseen data.

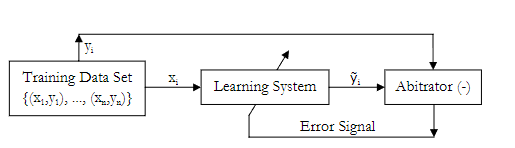


Figure 1. Block Diagram to illustrate Supervised Learning

In the context of this project, which focuses on sentiment analysis of Amazon phone reviews, supervised learning is particularly apt. We leverage labelled data, where each review is tagged with its sentiment label (positive, negative, or neutral). The models, including Decision Tree, Random Forest, and Support Vector Machine (SVM), are trained on this annotated dataset to understand the relationships between textual features and sentiment categories.

Supervised Learning Algorithms are primarily used for two tasks.

**Classification**

In classification tasks, the model predicts discrete class labels. For instance, determining whether an email is spam or not, or classifying images of digits into their respective numbers and this case classifying sentiment of human generated text.

**Regression**

Regression tasks involve predicting continuous values. Examples include predicting house prices based on features like square footage and location or estimating the temperature based on historical data.

## **Decision Trees**

# Decision Trees are powerful and interpretable machine learning algorithms widely employed for both classification and regression tasks. These tree-like structures facilitate decision-making processes by breaking down complex problems into a series of simpler, sequential decisions. This section provides an overview of Decision Trees, their characteristics, and their relevance to the sentiment analysis project.

**1. Key Characteristics of Decision Trees**

**Tree Structure:**

Decision Trees exhibit a hierarchical, tree-like structure, comprising nodes, branches, and leaves.

Each internal node represents a decision based on a feature, each branch signifies the outcome of the decision, and each leaf node denotes the final predicted class or value.

**Decision Nodes:**

Decision nodes evaluate specific features and make binary decisions based on predefined conditions.

In sentiment analysis, decision nodes may assess features derived from text, such as word frequency or sentiment indicators, to determine the sentiment of a given review.

**Leaf Nodes:**

Leaf nodes represent the final output or prediction.

In sentiment analysis, each leaf node may correspond to a sentiment class, such as positive, negative, or neutral.

**Splitting Criteria:**

Decision Trees employ splitting criteria to determine the feature and threshold for dividing the data at each decision node.

Common splitting criteria include Gini impurity for classification tasks and mean squared error for regression tasks.

**2. Decision Trees in Sentiment Analysis:**

1. **Feature Selection:**

In sentiment analysis, Decision Trees can automatically select relevant features from textual data, aiding in the identification of key indicators of sentiment.

1. **Handling Non-Linear Relationships:**

Decision Trees can capture non-linear relationships and interactions between features, making them well-suited for sentiment analysis tasks where sentiment expression is often nuanced.

1. **Interpretability in Sentiment Classification:**

The interpretability of Decision Trees is particularly valuable in sentiment analysis, as it allows stakeholders to understand which features contribute to specific sentiment predictions.

1. **Ensemble Methods:**

Decision Trees can be part of ensemble methods, such as Random Forests, where multiple trees are combined to improve predictive performance.

Ensemble methods enhance the robustness of sentiment analysis models by mitigating overfitting and increasing generalization**.**

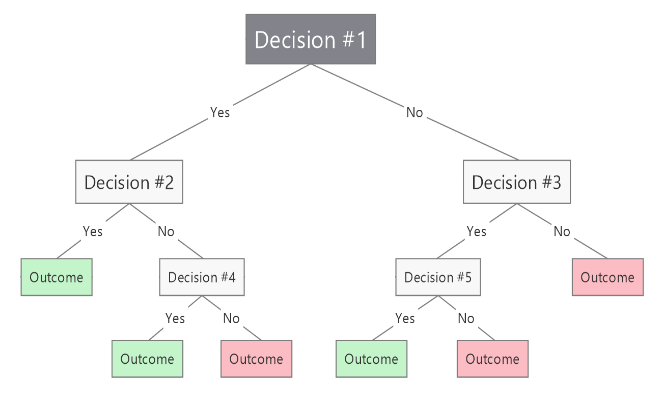


Figure 2. A Typical Decision Tree Structure

## **SVMs (Support Vector Machines)**

Support Vector Machines (SVMs) are powerful and versatile machine learning algorithms used for both classification and regression tasks. Developed in the context of statistical learning theory, SVMs aim to find the optimal hyperplane that maximally separates data points belonging to different classes. This section provides an overview of SVMs, their key characteristics, and their relevance to sentiment analysis in the project.

**1. Key** **Characteristics of Support Vector Machines**

**Hyperplane and Margin:**

SVMs seek to find the hyperplane that maximizes the margin between different classes of data points.

The margin is the distance between the hyperplane and the nearest data point from each class, ensuring a clear separation.

**Support Vectors:**

Support Vectors are the data points that lie closest to the decision boundary (hyperplane).

These points play a crucial role in determining the optimal hyperplane and are pivotal in the SVM training process.

**Kernel Trick:**

SVMs can handle non-linear relationships through the use of kernel functions.

The kernel trick transforms the input features into a higher-dimensional space, allowing SVMs to effectively separate non-linearly separable data.

**C Parameter:**

The regularization parameter C in SVMs controls the trade-off between achieving a smooth decision boundary and correctly classifying training data.

A smaller C encourages a wider margin but allows for more misclassifications, while a larger C enforces a stricter classification at the cost of a narrower margin.

**2. SVMs in Sentiment Analysis:**

1. **Effective in High Dimensional Spaces**

Sentiment analysis often involves high-dimensional feature spaces derived from textual data. SVMs excel in such scenarios, providing effective classification in these complex spaces.

1. **Handling Non-Linear Relationships:**

SVMs, with the use of appropriate kernel functions, can effectively handle non-linear relationships in sentiment expressions, capturing nuanced patterns in user reviews.

1. **Robust Performance:**

SVMs are known for their robust performance, particularly in scenarios where the dataset may be imbalanced or noisy.

This characteristic makes SVMs well-suited for sentiment analysis tasks where the distribution of sentiments may vary.

1. **Tuning for Sensitivity vs. Specificity:**

The C parameter in SVMs allows for tuning the sensitivity and specificity of the model, catering to specific requirements in sentiment analysis applications.

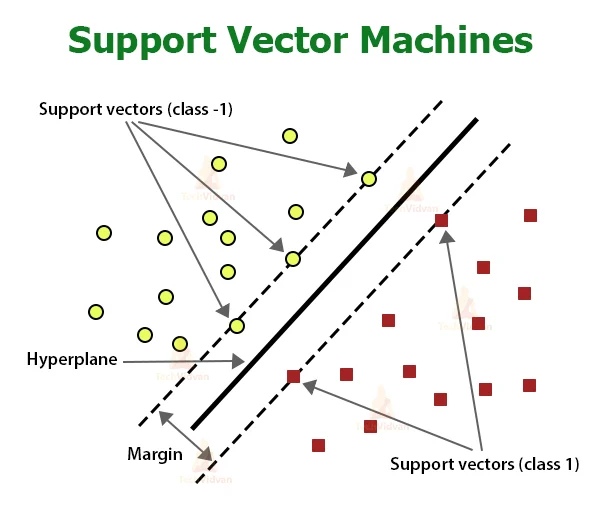


Figure 3. A Visual Representation of SVMs

## **Random Forest**

Random Forest is a powerful ensemble learning algorithm that belongs to the family of decision tree-based methods. It leverages the strength of multiple decision trees to improve predictive accuracy and generalization. This section provides an overview of Random Forest, its key features, and its relevance to sentiment analysis in the project.

**1.**  **Key Characteristics of Random Forest**

**Ensemble Of Decision Trees:**

Random Forest consists of an ensemble of decision trees, each trained on a different subset of the data and features.

The predictions of individual trees are combined through voting or averaging to produce the final prediction.

**Bootstrapping (Bagging):**

Each decision tree in the Random Forest is trained on a bootstrapped sample, which is a random sample with replacement from the original dataset.

This technique, known as bagging (bootstrap aggregating), introduces diversity among the trees, making the ensemble more robust.

**Feature Randomization:**

At each node of a decision tree, a random subset of features is considered for splitting.

This feature randomization further enhances the diversity of the trees and prevents the Random Forest from being overly sensitive to specific features.

**Voting or Averaging:**

For classification tasks, Random Forest combines the predictions of individual trees through majority voting.

For regression tasks, the predictions are averaged, providing a more stable and accurate prediction.

**2. Random Forest in Sentiment Analysis:**

1. **High Predictive Accuracy**

Random Forest is known for its high predictive accuracy, making it suitable for sentiment analysis tasks where accurate classification of sentiments is crucial.

1. **Handling Non-Linear Relationships**

Random Forest can effectively capture non-linear relationships present in textual data, making it adept at discerning the nuanced sentiment expressions in user reviews.

1. **Robustness to Noisy Data:**

The ensemble approach of Random Forest makes it robust to noisy or irrelevant features in the dataset, enhancing its performance in real-world scenarios.

1. **Feature Importance:**

Random Forest provides a measure of feature importance, indicating which features contribute more significantly to the classification. This information can be valuable for interpreting and understanding the key factors influencing sentiment predictions.

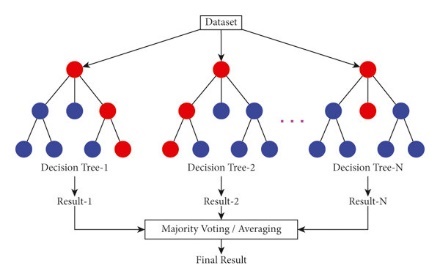


Figure 4. Representation of Random Forest

## **Prominent Research**

*"Opinion Mining and Sentiment Analysis" (Pang and Lee, 2008)*

Pang and Lee's comprehensive review provides a foundational understanding of opinion mining and sentiment analysis methodologies. The paper outlines the evolution of techniques, from lexicon-based approaches to machine learning-based methods, offering valuable insights into the challenges and opportunities in the field.

*"Thumbs up? Sentiment Classification using Machine Learning Techniques" (Pang, Lee, and Vaithyanathan, 2002)*

This early work by Pang, Lee, and Vaithyanathan explores the application of machine learning techniques for sentiment classification. The study delves into the challenges of sentiment analysis and introduces the use of machine learning algorithms, laying the groundwork for subsequent research in the field.

*"A Comparative Study on Sentiment Classification of Movie Reviews in Malayalam using Machine Learning Techniques" (Sreekumar and Sreejith, 2017)*

Sreekumar and Sreejith's comparative study focuses on sentiment classification in a specific linguistic context, Malayalam movie reviews. The research contributes insights into the adaptability of machine learning techniques, including Decision Trees, in capturing sentiments in diverse languages and domains.

*"Scikit-learn: Machine Learning in Python" (Pedregosa et al., 2011)*

The scikit-learn library, as detailed by Pedregosa and his colleagues, has become a cornerstone in machine learning research. Its extensive collection of algorithms, including implementations of Decision Trees, Random Forests, and Support Vector Machines, has provided researchers with a powerful toolkit for sentiment analysis experimentation.

*"Twitter mood predicts the stock market" (Bollen, Mao, and Zeng, 2011)*

Bollen, Mao, and Zeng explore the intriguing connection between Twitter sentiment and stock market trends. This pioneering work demonstrates the real-world impact of sentiment analysis, showing how social media sentiment can serve as an indicator for financial market movements.

# **Methodology**

**1. Data Collection:**

The foundation of this sentiment analysis project lies in a dataset extracted from Amazon phone reviews. This dataset comprises a diverse collection of user-generated reviews, capturing a wide spectrum of sentiments expressed towards various phone products. The dataset includes textual content as well as associated user ratings, forming the basis for training and evaluating the sentiment analysis models.

**2.** **Data Preprocessing:**

* **Text Cleaning:**

The raw textual data undergoes a series of preprocessing steps to ensure a clean and standardized input for the machine learning models. This includes the removal of special characters, punctuation, and irrelevant symbols. Text is converted to lowercase to maintain consistency across the dataset.

* **Tokenization and Stopword Removal:**

The reviews are tokenized into individual words, and common stopwords are removed to focus on meaningful content. This step is crucial in reducing the dimensionality of the data and emphasizing keywords relevant to sentiment.

* **TF-IDF Vectorization:**

The preprocessed text is transformed into numerical feature vectors using the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization technique. This process captures the importance of words within each review relative to their occurrence in the entire dataset.

**3**. **Model Training:**

* **Decision Trees:**

A Decision Tree model is trained on the TF-IDF vectors to learn patterns and relationships within the textual features. The interpretability of Decision Trees allows for a clear understanding of the decision-making process in sentiment classification.

* **Random Forest:**

Building upon Decision Trees, a Random Forest ensemble is constructed by training multiple trees on bootstrapped samples of the dataset. The feature randomization inherent in Random Forests enhances model robustness and predictive accuracy.

* **Support Vector Machines (SVMs):**

SVMs are employed for their ability to handle high-dimensional feature spaces and effectively capture non-linear relationships. The SVM model is trained to find the optimal hyperplane that separates different sentiment classes in the TF-IDF space.

**4. Model Evaluation:**

Hyperparameters of the models, such as the depth of Decision Trees or the regularization parameter in SVMs, are fine-tuned using techniques like grid search and cross-validation. This step ensures optimal model performance and generalization to unseen data.

# Experimental setup

**1. Hardware and Software Configurations:**

The experimental setup utilizes Google Colab and its Cloud based computing units, ensuring sufficient computational resources for model training and evaluation. This includes a TPU (Tensor Processing Unit).

The experiments are conducted using the Python programming language and key libraries for machine learning, including scikit-learn, NumPy, pandas, NLTK, bs4 and Matplotlib. The scikit-learn library provides implementations of Decision Trees, Random Forests, and SVMs, streamlining the model development process.

2. **Dataset Partitioning:**

The Amazon phone review dataset is divided into three main partitions: a training set, a validation set, and a test set. The training set, comprising is used for training the models. The validation set aids in hyperparameter tuning. The final evaluation is performed on the test set.

**3.** **Result Assessment and Evaluation**

The performance of each model is evaluated on the test set using a comprehensive set of metrics, including accuracy, precision, recall, and F1 score. The confusion matrix is generated to visualize the model's ability to correctly classify sentiments.

4. **Hyperparameter Tuning**

The hyperparameters of each model are fine-tuned using grid search and cross-validation on the validation set. This process ensures that the models generalize well to unseen data and are not overfitting to the training set.

# **Results**

The following are visualizations and results obtained from the dataset used.

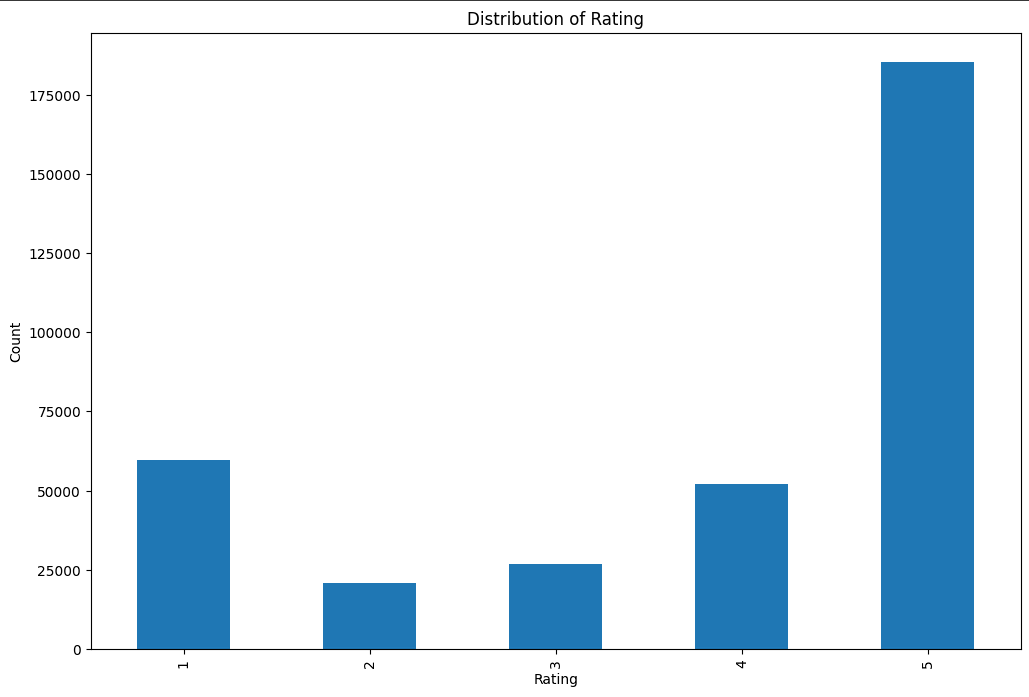


Figure 5. Distributions of Ratings

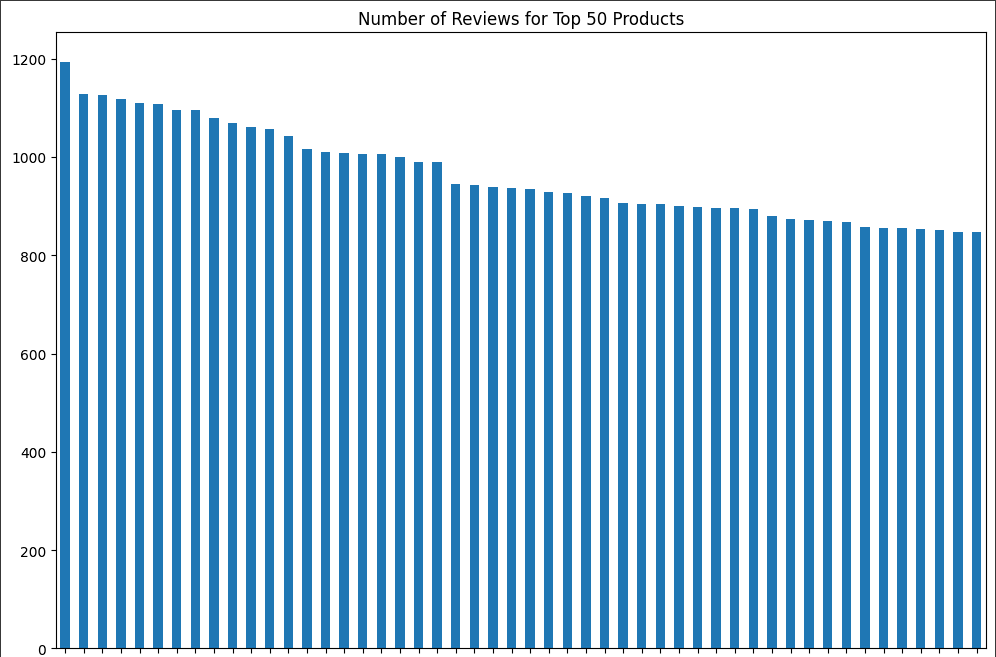


Figure 6. Number of Reviews for Top 50 Products

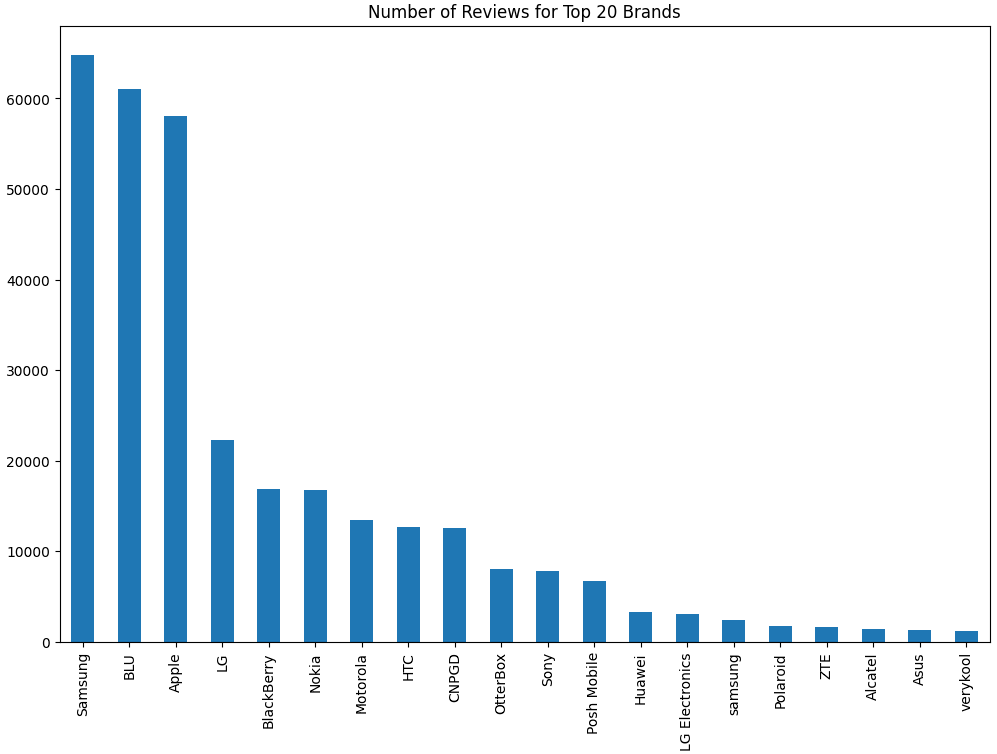


Figure 7. Review count for top 20 Brands

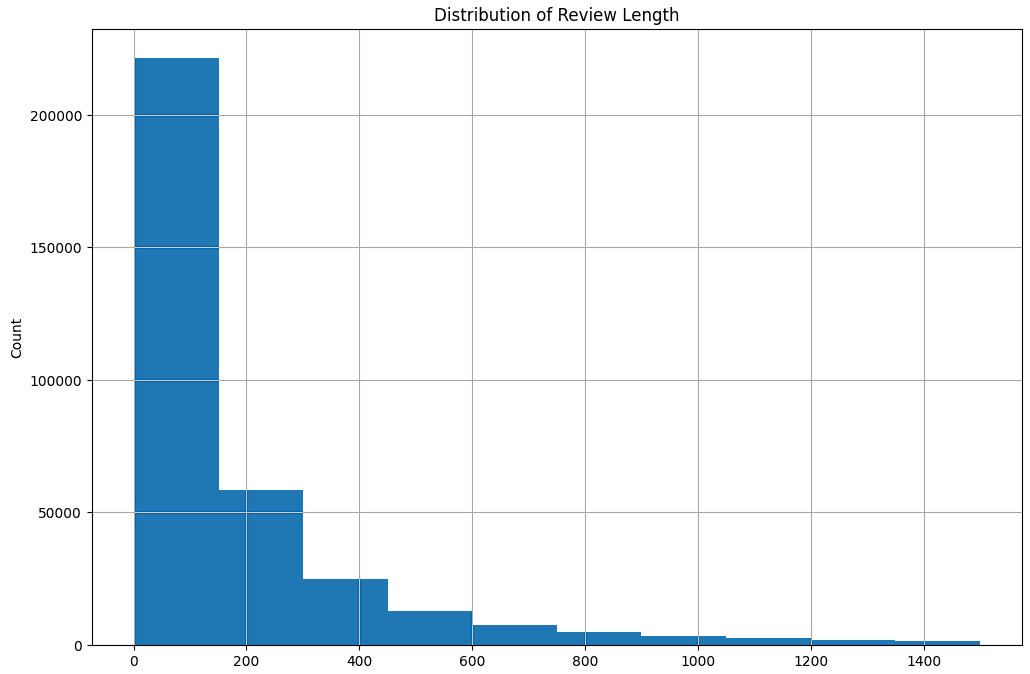


Figure 8. Distribution of Review Length

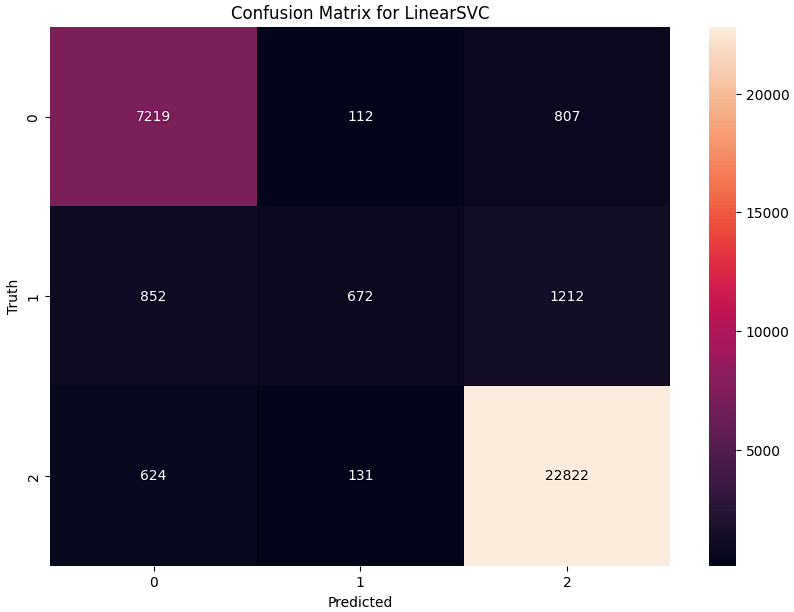


Figure 9. Confusion Matrix for SVM

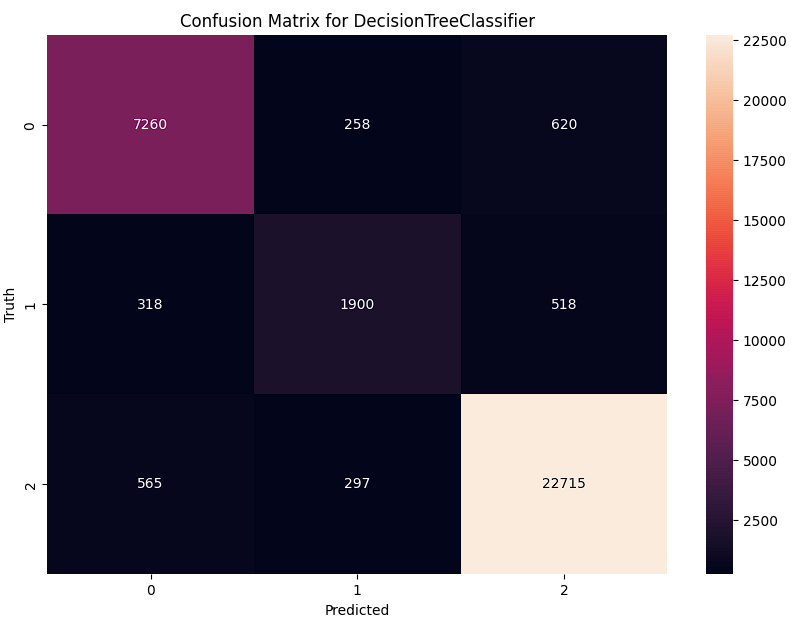


Figure 10. Confusion Matrix for Decision Tree

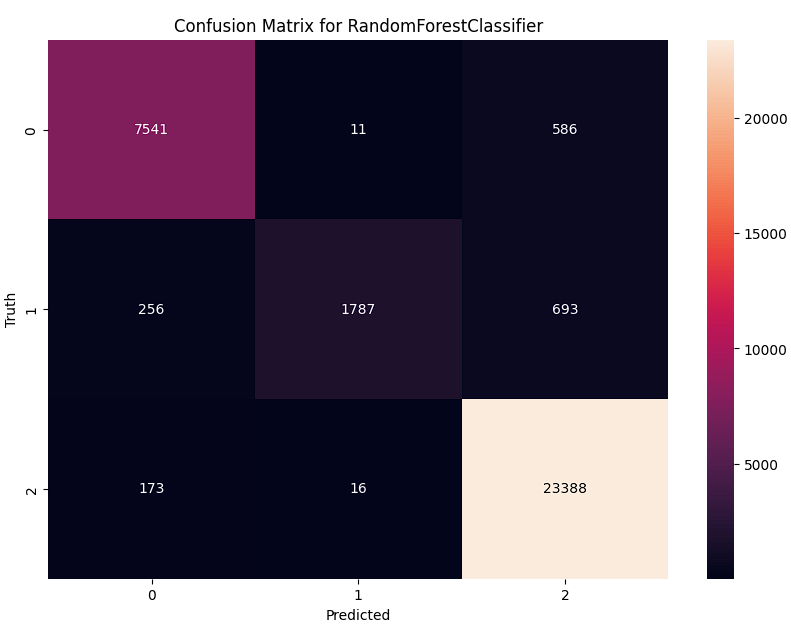


Figure 11. Confusion Matrix for Random Forest

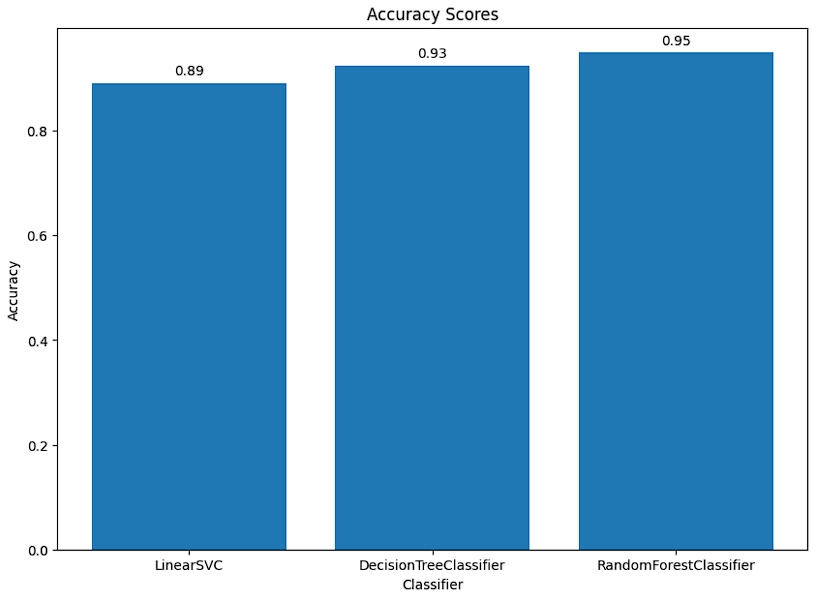


Figure 12. Accuracy Scores for Each Model

##### 

Figure 13. F1 Scores for Each Model (Weighted Average)

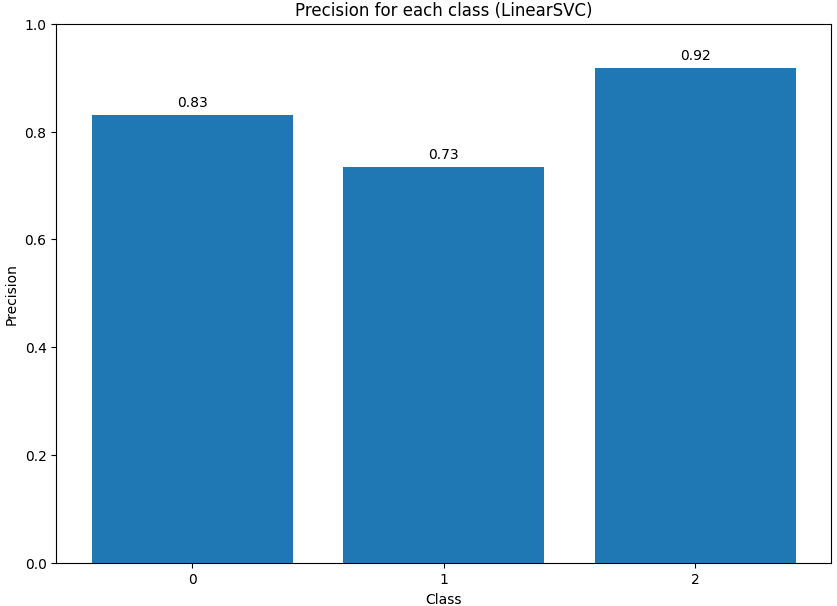


Figure 14. Precision for Each Class (SVM)

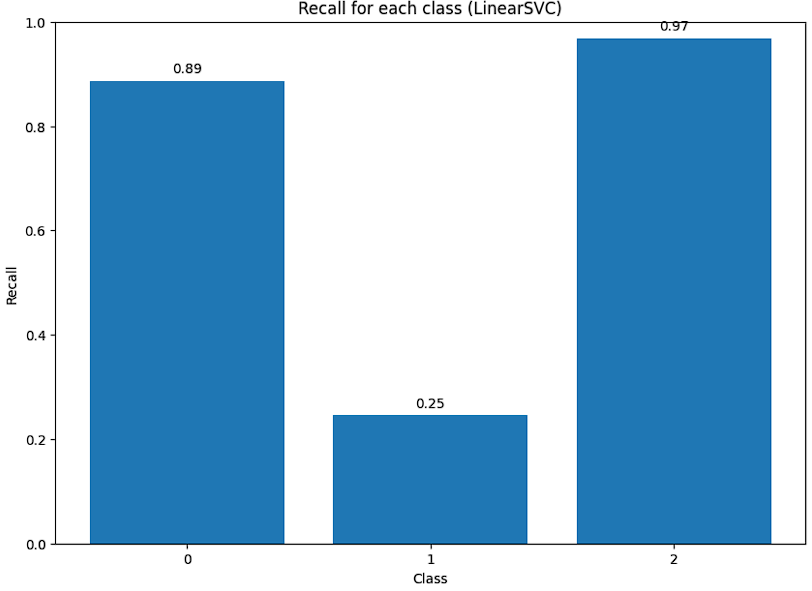


Figure 15. Recall for Each Class (SVM)

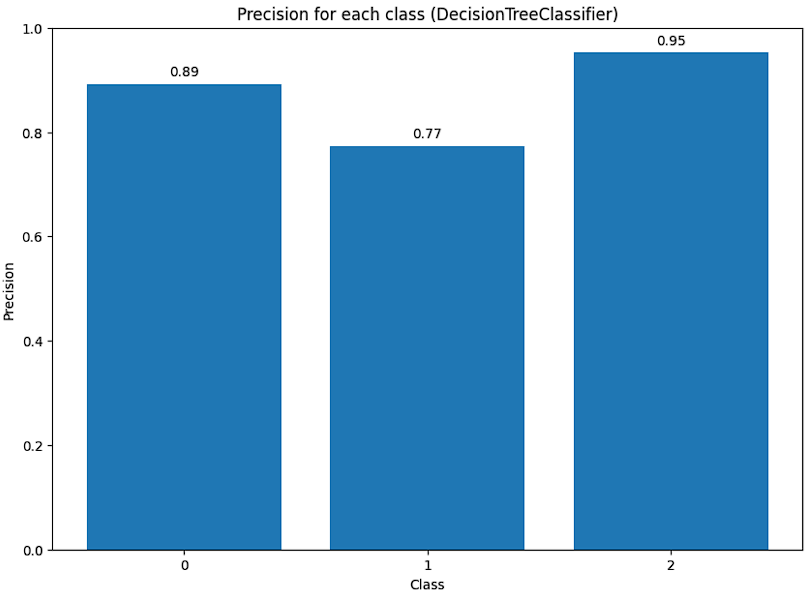
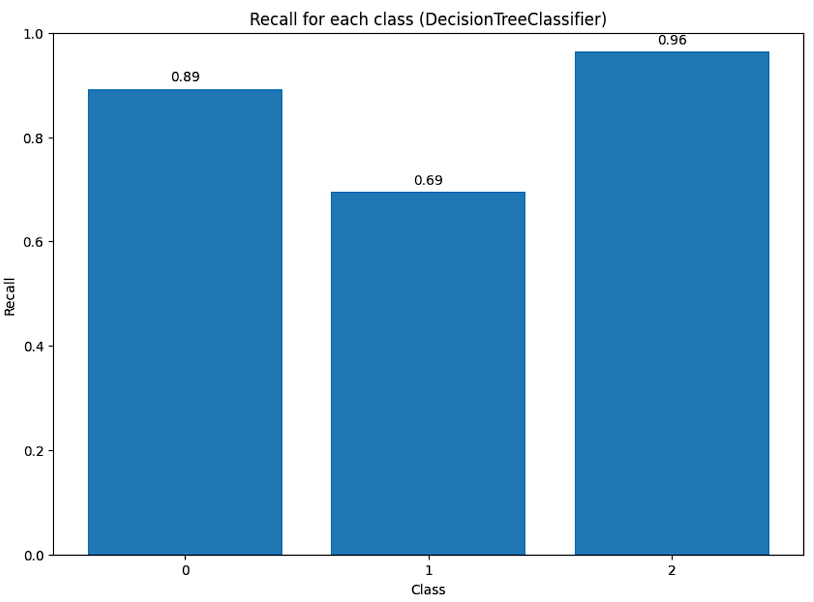
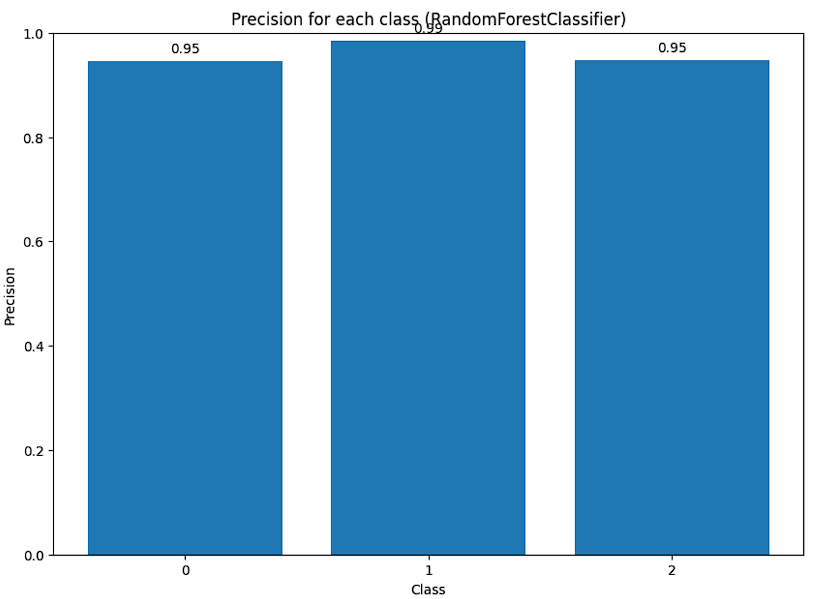


Figure 16. Precision for Each Class (Decision Tree)

 Figure 17. Recall for Each Class (Decision Tree)

 Figure 18. Precision for Each Class (Random Forest)

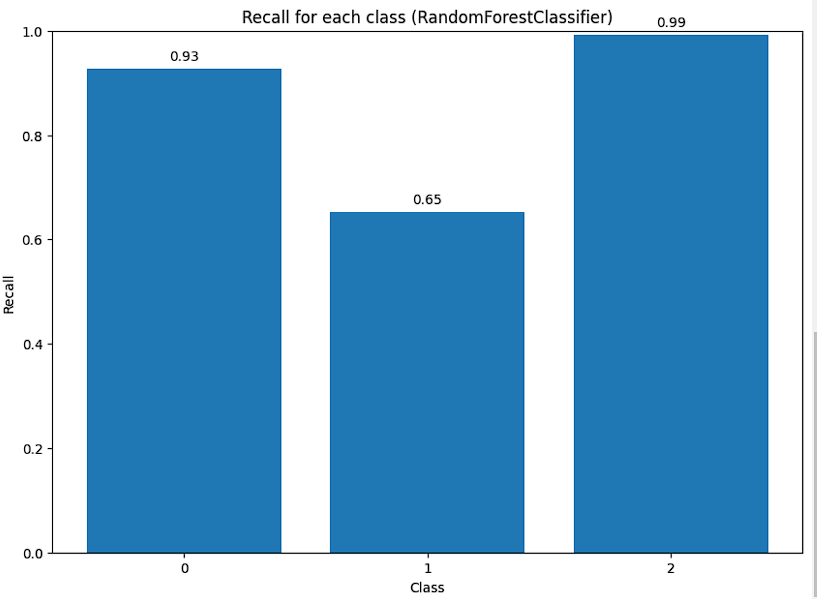


Figure 19. Recall for Each Class (Random Forest)

1. **Random Forest Classifier**

The Random Forest Classifier achieved an accuracy of **0.9496** on the validation set. The precision, recall, and F1-score for each class are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1** |
| 0 | 0.95 | 0.93 | 0.94 |
| 1 | 0.99 | 0.65 | 0.79 |
| 2 | 0.95 | 0.99 | 0.97 |

Class 0: Precision of 0.95, Recall of 0.93, and F1-score of 0.94

Class 1: Precision of 0.99, Recall of 0.65, and F1-score of 0.79

Class 2: Precision of 0.95, Recall of 0.99, and F1-score of 0.97

The confusion matrix shows that the model has a good performance on Class 0 and Class 2, but struggles a bit with Class 1, particularly with false negatives (lower recall).

1. **Decision Tree Classifier**

The Decision Tree Classifier achieved an accuracy of **0.9252** on the validation set. The precision, recall, and F1-score for each class are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1** |
| 0 | 0.89 | 0.89 | 0.89 |
| 1 | 0.77 | 0.69 | 0.73 |
| 2 | 0.95 | 0.96 | 0.96 |

Class 0: Precision of 0.89, Recall of 0.89, and F1-score of 0.89

Class 1: Precision of 0.77, Recall of 0.69, and F1-score of 0.73

Class 2: Precision of 0.95, Recall of 0.96, and F1-score of 0.96

The confusion matrix shows that the model performs well on Class 0 and Class 2, but has some difficulty with Class 1, with a lower recall indicating a higher number of false negatives.

1. **SVM**

The SVM achieved an accuracy of **0.8915** on the validation set. The precision, recall, and F1-score for each class are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1** |
| 0 | 0.83 | 0.89 | 0.86 |
| 1 | 0.73 | 0.25 | 0.37 |
| 2 | 0.92 | 0.97 | 0.94 |

Class 0: Precision of 0.83, Recall of 0.89, and F1-score of 0.86

Class 1: Precision of 0.73, Recall of 0.25, and F1-score of 0.37

Class 2: Precision of 0.92, Recall of 0.97, and F1-score of 0.94

The confusion matrix shows that the SVM performs well on Class 0 and Class 2, but struggles significantly with Class 1, particularly with false negatives (lower recall).

In summary, all three models perform well on Class 0 and Class 2, but have some difficulty with Class 1. The Random Forest Classifier has the highest overall accuracy, but there may be room for improvement in its handling of Class 1. The SVM, despite having the lowest overall accuracy, performs comparably to the other models on Class 0 and Class 2, but its performance on Class 1 is significantly worse. This suggests that Class 1 may be more difficult to predict accurately, or that it may be underrepresented in the training data. Further investigation and potential model tuning would be beneficial to improve these results.

# **Disscussion and Future Scope**

The Random Forest Classifier achieved the highest accuracy among the three models, with an accuracy of 0.9496 on the validation set. This suggests that the Random Forest model was able to correctly classify a large proportion of the instances in the validation set. However, the performance across different classes varied. While the model performed well on Class 0 and Class 2, it struggled with Class 1, particularly with false negatives as indicated by the lower recall. This could be due to various factors such as Class 1 being underrepresented in the training data, or the features not being as discriminative for Class 1. It would be worth investigating this further, perhaps by looking at the distribution of classes in the training data or examining the feature importance in the Random Forest model. It’s also important to note that while the Random Forest Classifier had the best performance, it also took the longest to train. This is a common trade-off in machine learning - more complex models that can capture intricate patterns in the data often require more computational resources and time to train.

The Decision Tree Classifier, while not as accurate as the Random Forest, still achieved a respectable accuracy of 0.9252 on the validation set. Similar to the Random Forest model, the Decision Tree model performed well on Class 0 and Class 2, but had some difficulty with Class 1. This again suggests that Class 1 may be more challenging to predict accurately. One possible reason could be that the decision boundaries for Class 1 are not as clear-cut, making it harder for the Decision Tree to correctly classify these instances. Further analysis could involve examining the decision tree structure to understand how the model is making its predictions.

The SVM had the lowest overall accuracy of 0.8915 on the validation set. Despite this, it performed comparably to the other models on Class 0 and Class 2. However, its performance on Class 1 was significantly worse, particularly with a large number of false negatives leading to a low recall. This suggests that the SVM’s decision boundary for Class 1 may not be optimal. Tuning the parameters of the SVM, such as the regularization parameter C or the kernel parameters, could potentially improve its performance on Class 1.

In conclusion, while all three models showed promising results, there is room for improvement, particularly in their ability to accurately predict Class 1. Future work could involve further analysis to understand why Class 1 is more challenging to predict, as well as experimenting with different model parameters, feature engineering techniques, or even trying different machine learning models. It’s also important to remember that model selection should not be based solely on accuracy, but also consider other factors such as interpretability, training and prediction time, and the specific requirements of the task at hand.

##### **Acknowledgment**

##### I extend my gratitude to the various indirect and direct sources of assistance that have contributed to the development of this research. My understanding of the intricate landscape of ensemble learning methods and supervised learning algorithms was significantly shaped by the wealth of knowledge available through IEEE publications. These platforms have been indispensable in providing me with the latest research findings and historical perspectives which underpin the theoretical foundation of my work. Special thanks are due to the open-source software and tools that have facilitated my experimental processes. The availability of these resources has allowed me to stand on the shoulders of giants, leveraging well-established codebases and libraries to bring my project and its results into empirical testing. Furthermore, I am grateful for the support systems that have indirectly contributed to this work. While not directly involved in the research, their role has been essential in providing an environment conducive to rigorous scientific inquiry.

##### **References**

1. “Thumbs up? Sentiment Classification using Machine Learning Techniques”: B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment Classification using Machine Learning Techniques,” in Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), 2002, pp. 79–86
2. “Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales”: B. Pang and L. Lee, “Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales,” in Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05), 2005, pp. 115–124
3. “Random Forests”: L. Breiman, “Random Forests,” Machine Learning, vol. 45, pp. 5–32, 2001
4. “Using Random Forest to Learn Imbalanced Data”: C. Chen, A. Liaw, and L. Breiman, “Using Random Forest to Learn Imbalanced Data,” University of California, Berkeley, Tech. Rep. 666, 2004
5. “From RankNet to LambdaRank to LambdaMART: An Overview”: C. J.C. Burges, “From RankNet to LambdaRank to LambdaMART: An Overview,” Microsoft Research, Tech. Rep. MSR-TR-2010-82, 2010
6. "A Comparative Study on Sentiment Classification of Movie Reviews in Malayalam using Machine Learning Techniques" (Sreekumar and Sreejith, 2017)
7. "Twitter mood predicts the stock market" (Bollen, Mao, and Zeng, 2011)
8. *"Opinion Mining and Sentiment Analysis" (Pang and Lee, 2008)*
9. "Scikit-learn: Machine Learning in Python" (Pedregosa et al., 2011)
10. "A Comparative Study on Sentiment Classification of Movie Reviews in Malayalam using Machine Learning Techniques" (Sreekumar and Sreejith, 2017)