Sentiment Analysis Coursework Report

Author: Chibuike Lawrence Orji-Oko

Student ID removed for public version

MSc Data Science & Artificial Intelligence

Table of Contents

1. Introduction

The following document is a report on developing a data analysis software that automatically evaluates sentiments in Amazon reviews for Nine Android apps by three Android Application Development (AAD) companies: AAD_1, AAD_2, and AAD_3. For algorithm development and evaluation, a training dataset of 20,000 reviews and a separate test dataset of 19,998 reviews will be used, ensuring distinct datasets for accuracy. Each review includes the sentiment label (1 - negative, 2 - neutral, and 3 - positive), app ID, and review text. The goal is to make informed investment decisions by identifying the most successful third-party Android Application Development (AAD) company among AAD_1, AAD_2, and AAD_3 based on customer satisfaction with their apps. Existing systems for data analysis will be critically analysed, reviewing their advantages and limitations. This report will highlight and discuss the steps taken to accomplish this task.

2. Literature Review

Sentiment analysis is a natural language processing (NLP) method used to extract emotional tone in a given text data whether it is positive, negative, or neutral. The sentiment analysis system consists of steps to uncover the emotions (positive, negative, or neutral) of the input data (Wankhade, Rao, and Kulkarni, 2022). This section reviews and critically analyses existing systems for sentiment analysis, discussing their advantages and limitations.

The lexicon-based method is a common system for sentiment analysis. This approach uses a predefined sentiment lexicon. A predefined sentiment lexicon is a collection of words or phrases labeled with their corresponding sentiment polarity used to learn the sentiment in a text (Min and Zulkarnain, 2020). The sentiment score of a given text is the aggregation of the sentiment scores of the individual words. Systems such as SentiWordNet and VADER use this technique and these systems' performance depends on the quality and coverage of the sentiment lexicon. The main advantage of lexicon-based methods is their simplicity, ease of implementation, and computationally less expensive. However, they perform poorly with context-dependent sentiment expressions, negation handling, and domain-specific language (Wankhade, Rao, and Kulkarni, 2022).

Machine learning methods are capable of analyzing large datasets and uncovering patterns and relationships within the data. This capability makes them adaptable for sentiment analysis. NLTK (Natural Language Toolkit) and Stanford CoreNLP provide frameworks for building sentiment analysis models using machine learning algorithms (Motitswane, 2023). Machine learning-based approaches involve training an algorithm on a labeled dataset and then using the trained model to make sentiment predictions on unseen data. This process can be expensive and time-consuming when working on a large dataset. Additionally, they often require extensive feature engineering and may not generalize well to different domains or languages (Motitswane, 2023).

Sentiment analysis based on neural network methods is gaining attention. BERT and XLNet systems attained state-of-the-art evaluation scores on various sentiment analysis tasks (Motitswane, 2023). Li, Goh, and Jin (2020) proposed a CNN architecture for sentence-level sentiment classification, achieving an improved performance on benchmark datasets. Deep learning models excel at capturing complex relationships and semantic features from large amounts of unlabeled data. However, these models require large amounts of computational resources for training and inference, making them impractical for resource-constrained environment tasks (Motitswane, 2023).

Bansal and Srivastava (2019) proposed a hybrid system integrating machine learning method with lexicon-based sentiment analysis for product reviews. These hybrid approaches can often outperform individual methods but may suffer from increased complexity and computational overhead (Bansal and Srivastava, 2019).

The choice of a sentiment analysis system is dependent on factors such as application requirements, data, and computational resources. Machine learning sentiment analysis is ideal for large labeled data. Given the requirement and the availability of 20,000 labeled training data and 19,998 labeled test data, machine learning-based sentiment analysis is used for this task.

3. Methodology

K-Nearest Neighbors (KNN)

KNN algorithm excels for both classification and regression tasks (Acito, 2023). In the context of sentiment analysis, KNN classifies a given text using the majority class label of its k nearest neighboring instances of the training dataset (Pamuji, 2021). The algorithm learns by storing the entire training dataset and computing the similarity (e.g., using distance metrics like Euclidean or Cosine) between a new instance and all the stored instances (Pamuji, 2021). The class labels of the k most similar instances are then used to determine the class of the new instance through majority voting or distance-weighted voting. KNN is simple and able to handle non-linear decision boundaries (Acito, 2023).

Support Vector Machine (SVM)

SVM (supervised learning model) finds the optimal hyperplane that maximally divides instances of different classes in a high-dimensional space (Motitswane, 2023). SVMs classify texts as positive, negative, or neutral by mapping the text features (e.g., word frequencies, n-grams) into a high-dimensional feature space and locating the hyperplane that best separates the classes (Sharma and Sabharwal, 2019). SVMs learn by solving an optimization problem that maximizes the margin between the classes' closest instances (support vectors). This makes SVMs effective for high-dimensional data and capable of handling non-linear class boundaries through kernel functions that make the class labels linearly separable (Motitswane, 2023). SVMs are robust to overfitting and achieve high accuracy in sentiment analysis tasks. However, it requires more computation resources for large datasets and is prone to hyperparameters and kernel functions (Sharma and Sabharwal, 2019).

Naive Bayes

Naive Bayes algorithm are built on Bayes' probability theorem and feature independence assumption (Kelly and Johnson, 2021). In sentiment analysis, Naive Bayes algorithm determines the polarity of a text classifiers calculate the probability of a text based on the frequency of words or features in the text and their corresponding probabilities in the training data (Le et al., 2019). Naive Bayes learns through probability estimation of each class and word given the class labels of the training data (Kelly and Johnson, 2021). Naive Bayes algorithm is simple and performs well in sentiment analysis tasks as it can handle high-dimensional datasets (Le et al., 2019). Strengths of Naive Bayes are computational efficiency, interpretability, and robustness to irrelevant features (Le et al., 2019). However, it can struggle with non-linear decision boundaries and may perform poorly when the feature independence assumption is severely violated (Kelly and Johnson, 2021).

Decision Tree

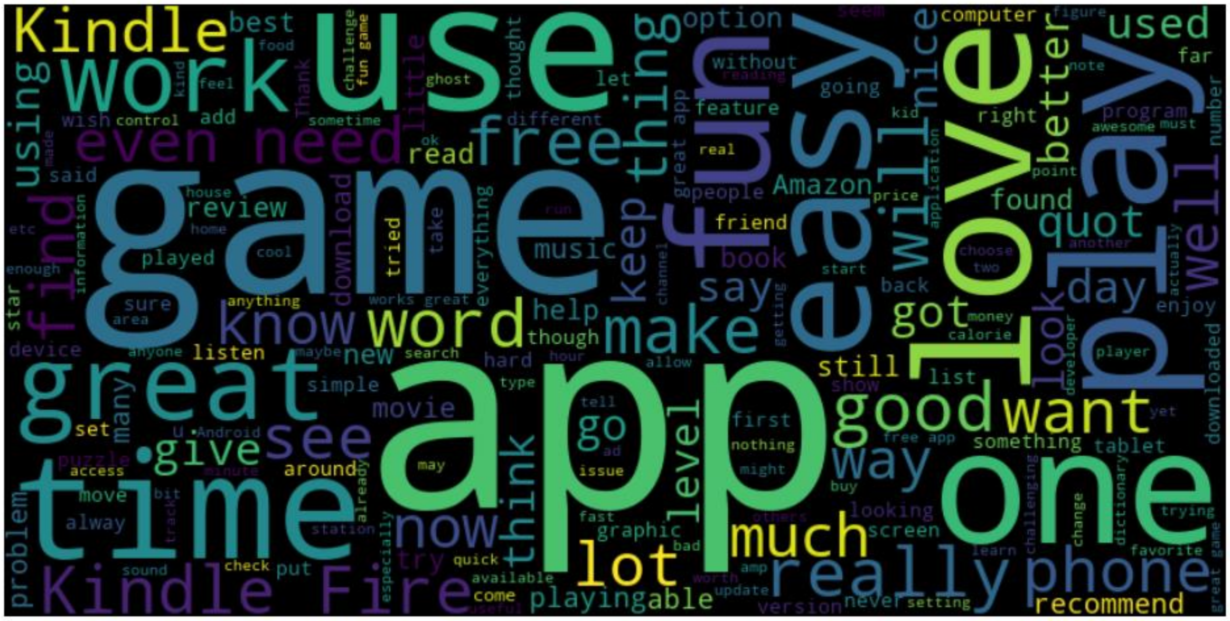
Decision Trees are recursive partitioning algorithms that split the data on the most prominent features, creating a tree-like structure of decisions and their possible consequences (Wankhade, Rao, and Kulkarni, 2022). Decision Trees classify texts into different sentiment classes by learning a series of rules based on certain words being present or absent in the text (Neogi et al., 2021). Decision Trees learn by breaking the data on the feature that attains the most information gain or entropy reduction at each node (Yan et al., 2019). This process goes on and only stops when a given criterion is achieved. Decision Trees are interpretable and perform well on both numerical and categorical datasets (Neogi et al., 2021). However, they are prone to overfitting and sensitive to the choice of splitting criteria and pruning methods (Wankhade, Rao, and Kulkarni, 2022).

Table 1: Strengths and Weaknesses of Sentiment Analysis Algorithms

Algorithm	**Strengths**	**Weaknesses**
K-Nearest Neighbors (KNN)	Simple to understand and implement. No training phase required. Effective non-linear decision boundaries-	Expensive computation resources for large datasets. Prone to K choice and distance metric. Weak that learning data distribution. Sensitive to features that irrelevant.
Support Vector Machine (SVM)	Effective for high-dimensional data. Robust to overfitting. Effective for non-linear class boundaries via kernel tricks.	Expensive computation resources for large datasets. Prone to kernel choice and hyperparameters. Difficult to interpret.
Naive Bayes	Simple and fast to train and classify. Good performance even with naive assumptions. Robust to irrelevant features. Interpretable.	Assumption of feature independence often violated. Struggles with non-linear decision boundaries.
Decision Tree	Easy to interpret and understand. Effective for both numerical and categorical data. No feature scaling required.	Prone to overfitting for high-dimensional data. Small changes in data can lead to very different trees. Performance sensitive to splitting criteria and pruning.

Dataset Statistics:

Figure 1: Word Cloud of Text Reviews



The training dataset used for developing/training the sentiment analysis consists of 20,000 reviews for three Android Application Development (AAD) companies. While, the test dataset used for evaluation consist of 19998 reviews.

Each review of the training and test datasets consists of a) a Class label (with values 1 - negative, 2 - neutral, and 3 - positive reviews), b) the ID code of the corresponding Android application, and the review text. The training and test datasets consist mainly of positive reviews, as shown in Table 2 below.

Table 2: Number of Positive, Negative, and Neutral Labels

Class Label	**Training Data**	**Test Data**
1	2983	3469
2	2205	2087
3	14812	14442

Table 3: Android Application Codes of the 3 AAD Companies

Company Name	**Android Application Code**
AAD_1	B004NWLM8K, B004Q1NH4U, B004LPBTAA
AAD_2	B004S6NAOU, B004R6HTWU, B004N8KDNY
AAD_3	B004KA0RBS, B004NPELDA, B004L26XXQ

4. Experiments

The datasets were preprocessed and transformed into numerical features using the Term Frequency-Inverse Document Frequency (TF-IDF) vectorisation. The algorithms were trained using the training dataset, and their performance was evaluated on the following metrics using the test dataset:

Accuracy measures the proportion of correctly predicted instances out of the total instances evaluated.

Precision is the ratio of instances classified as positive (or negative or neutral) that were indeed correct.

Recall is the fraction of positive (or negative or neutral) instances that were indeed correct.

F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance.

Table 4: Performance Summary of the Algorithms across the Evaluation Metrics

Algorithm	Accuracy	Precision	Recall	F1-score
Naive Bayes	0.72	0.68	0.72	0.61
Support Vector Machine	0.80	0.77	0.80	0.78
K-Nearest Neighbors	0.75	0.70	0.75	0.72
Decision Tree	0.70	0.69	0.70	0.69

Based on the results, the SVM algorithm had the highest overall accuracy of 0.80. The K-Nearest Neighbors (KNN) at 0.75. The Decision Tree and Naive Bayes algorithms had slightly lower accuracies of 0.70 and 0.72, respectively.

For a balanced measure of the model’s performance, the SVM outperformed the other algorithms with a 0.78 F1-score. Decision Tree and KNN algorithms had comparable F1-scores of 0.69 and 0.72 respectively, while Naive Bayes algorithm had the lowest F1-scores of 0.61.

The superior performance of SVM can be credited to its adeptness in managing high-dimensional feature spaces and its resilience against overfitting, a crucial factor in tasks such as text classification (Sharma and Sabharwal, 2019).

The Naive Bayes algorithm, despite its simplicity and computational efficiency, had the lowest overall performance. This could be due to the violation of the feature independence assumption, which is common in text data, where the presence of certain words or phrases can be highly correlated (Kelly and Johnson, 2021).

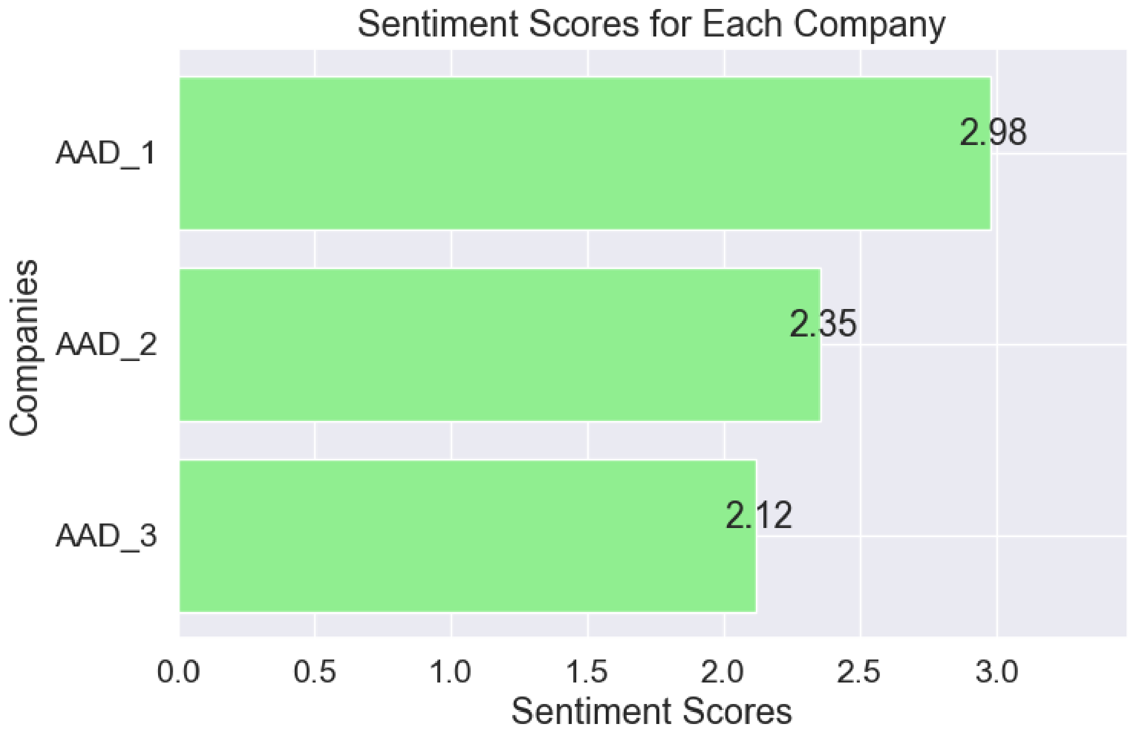
The SVM algorithm is the best sentiment analysis solution among the four algorithms evaluated. It performed better than all the other algorithms on all the metrics evaluated as shown in Table 4 above, indicating its ability to effectively classify sentiment in text data.

5. Analysis Results

Based on the analysis of the Amazon reviews for 9 Android applications, the AAD company likely to be the most successful is AAD_1. This conclusion is from the company average sentiment scores calculated from the predicted sentiments of the reviews.

AAD_1 has the highest average sentiment score of 2.981308, indicating that the reviews for its applications (B004NWLM8K, B004Q1NH4U, B004LPBTAA) were, on average, the most positive compared to the other companies. A higher sentiment score suggests that users had a more positive experience with the applications developed by AAD_1, as reflected in their reviews. Positive user sentiment and satisfaction are crucial factors that contribute to an app's success and adoption.

Figure 2: Average Sentiment Scores of the AAD Companies



Conclusion

This project successfully demonstrated proficiency in combining analysis methods and visualization tools to develop a complete big-data analysis solution for sentiment analysis. Through a comprehensive evaluation of multiple algorithms and their implementation to real-world data, the project critically interpreted and evaluated the results to inform decision-making processes.

This project demonstrated proficiency in combining various analysis methods, such as exploratory data analysis, preprocessing, feature extraction, model training, and evaluation, to build a complete sentiment analysis solution. The critical interpretation and evaluation of the results, including the strengths and weaknesses of each algorithm and the analysis of user sentiment data, provided valuable insights to inform decision-making processes. Identifying the most successful AAD company (AAD_1) based on user sentiment analysis highlights the application of these techniques to solve real-world problems. The project has demonstrated the value of leveraging big data analysis solutions, such as sentiment analysis, to gain actionable insights and make informed business decisions.

References

ACITO, F., 2023. K Nearest Neighbors. In Predictive Analytics with KNIME: Analytics for Citizen Data Scientists (pp. 209-227). *Cham: Springer Nature Switzerland [online].* Available from: https://link.springer.com/chapter/10.1007/978-3-031-45630-5_10 [Accessed 24 April 2024].

BANSAL, B. AND SRIVASTAVA, S., 2019. Hybrid Attribute Based Sentiment Classification of Online Reviews for Consumer Intelligence. *Applied Intelligence [online]*. 49(1). pp.137-149. Available from: https://link.springer.com/article/10.1007/s10489-018-1299-7 [Accessed 22 April 2024].

KELLY, A. AND JOHNSON, M.A., 2021. Investigating the Statistical Assumptions of Naïve Bayes Classifiers. *In 2021 55th Annual Conference on Information Sciences and Systems (CISS)* (pp. 1-6). *IEEE [online]*. Available from: https://doi.org/10.1109/CISS50987.2021.9400215 [Accessed 25 April 2024].

LE, C.C., PRASAD, P.W.C., ALSADOON, A., PHAM, L. AND ELCHOUEMI, A., 2019. Text Classification: Naïve Bayes Classifier with Sentiment Lexicon. *IAENG International journal of computer science [online]*. 46(2). pp.141-148. Available from: https://researchoutput.csu.edu.au/ws/portalfiles/portal/30550232/30550129_Published_article.pdf [Accessed 25 April 2024].

LI, L., GOH, T.T. AND JIN, D., 2020. How Textual Quality of Online Reviews Affect Classification Performance: A Case of Deep Learning Sentiment Analysis. *Neural Computing and Applications [online]*. 32. pp.4387-4415. Available from: https://link.springer.com/article/10.1007/s00521-018-3865-7 [Accessed 20 April 2024].

MIN, W.N.S.W. AND ZULKARNAIN, N.Z., 2020. Comparative Evaluation of Lexicons in Performing Sentiment Analysis. *Journal of Advanced Computing Technology and Application (JACTA) [online]* 2(1). pp.1-8. Available from: https://jacta.utem.edu.my/jacta/article/view/5207 [Accessed 17 April 2024].

MOTITSWANE, O.G., 2023. Machine Learning and Deep Learning Techniques for Natural Language Processing with Application to Audio Recordings (Doctoral Dissertation, North-West University (South Africa)) [online]. Available from: https://repository.nwu.ac.za/handle/10394/42346 [Accessed 21 April 2024].

NEOGI, A.S., GARG, K.A., MISHRA, R.K. AND DWIVEDI, Y.K., 2021. Sentiment Analysis and Classification of Indian Farmers' Protest Using Twitter Data. *International Journal of Information Management Data Insights [online]*. 1(2). p.100019. Available from: https://www.sciencedirect.com/science/article/pii/S2667096821000124 [Accessed 25 April 2024].

PAMUJI, A., 2021. Performance of the K-Nearest Neighbors Method on Analysis of Social Media Sentiment. *Jurnal Informatika dan Sistem Informasi [online*]. 7(1). pp.32-37. Available from: https://journal.uc.ac.id/index.php/JUISI/article/view/2084 [Accessed 24 April 2024].

SHARMA, D. AND SABHARWAL, M., 2019. Sentiment Analysis for Social Media Using SVM Classifier of Machine Learning. *Int J Innov Technol Exploring Eng (IJITEE) [online]*. 8(9). pp.39-47. Available from: https://doi.org/10.35940/ijitee.I1107.0789S419 [Accessed 24 April 2024].

WANKHADE, M., RAO, A.C.S. AND KULKARNI, C., 2022. A Survey on Sentiment Analysis Methods, Applications, and Challenges. *Artificial Intelligence Review [online]*. 55(7). pp.5731-5780. Available from: https://link.springer.com/article/10.1007/s10462-022-10144-1 [Accessed 17 April 2024].