TY B.TECH **A.Y : 2023-2024** **MACHINE LEARNING**

GOOGLE PLAY STORE SENTIMENT ANALYSIS USING NLP

Abstract

This sentiment analysis project utilized Natural Language Processing (NLP) techniques to classify app store reviews as positive or negative based on their sentiment. With over 12,000 reviews, rated from 1 to 5, initial preprocessing steps involved removing irrelevant columns, handling missing values, and filtering reviews based on scores to focus on positive (scores 4 and 5) and negative (scores 1 to 3) sentiments. Text preprocessing techniques, including punctuation and stopwords removal, were applied to clean the textual data. For feature extraction, CountVectorizer converted text data into numerical form, and the 'score' column was transformed into binary labels.Several classification models, such as Naive Bayes, SVM, Random Forest, and Logistic Regression, were trained and evaluated for sentiment classification. Additionally, NLTK's VADER Sentiment Intensity Analyzer provided insights into customer sentiments, aiding businesses in decision-making.

Dataset

The dataset contains over 12,000 app store reviews with 12 features. Essential attributes such as 'reviewId', 'userName', and 'content' are included, alongside the review score, rated on a scale of 1 to 5. Additional information encompasses 'thumbsUpCount', 'reviewCreatedVersion', and 'appId', which denotes the reviewed application. Some reviews have corresponding replies, evident from 'replyContent' and 'repliedAt' columns.

Objectives

- Efficiently classify app store reviews into positive and negative sentiments using NLP techniques.
- Extract valuable insights into customer sentiments towards various applications.
- Ensure robust sentiment analysis with diverse classification models and tools.
- Demonstrate scalability in handling large textual datasets with over 12,000 reviews.

Data Visualization and Processing

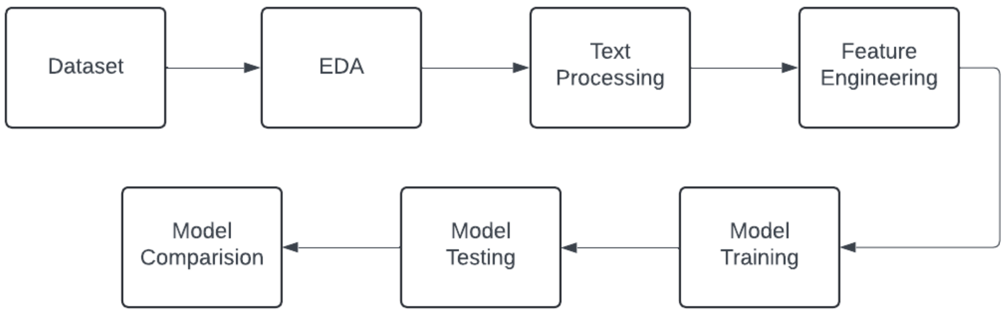
1) Removed 'replyContent' and 'repliedAt' columns, visualized null values with a heatmap, checked null value counts, filtered dataframe based on scores 1 and 5, added a new 'length' column based on the length of 'content', and created a new dataframe with selected columns.

2) Applied CountVectorizer to tokenize text data and convert it into a matrix of token counts.

3) Converted the 'score' column into binary labels, where scores 1 to 3 are considered as negative (label 0) and scores 4 to 5 are considered as positive (label 1), as part of feature engineering and data preprocessing.

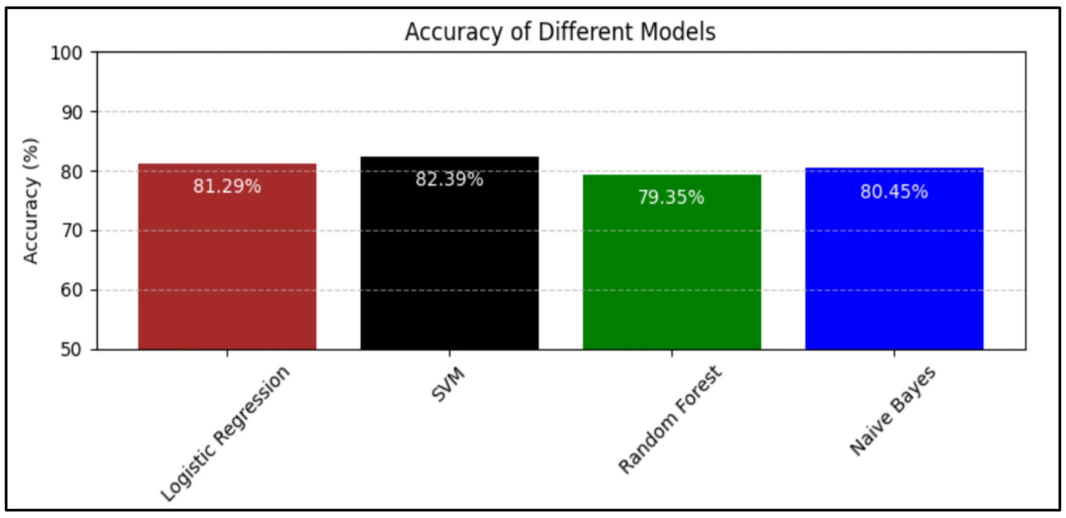
4) Added a new column 'binary_label' to df_new which represents binary labels for good reviews (4-5) as 1 and negative reviews (1-3) as 0.

System Architecture

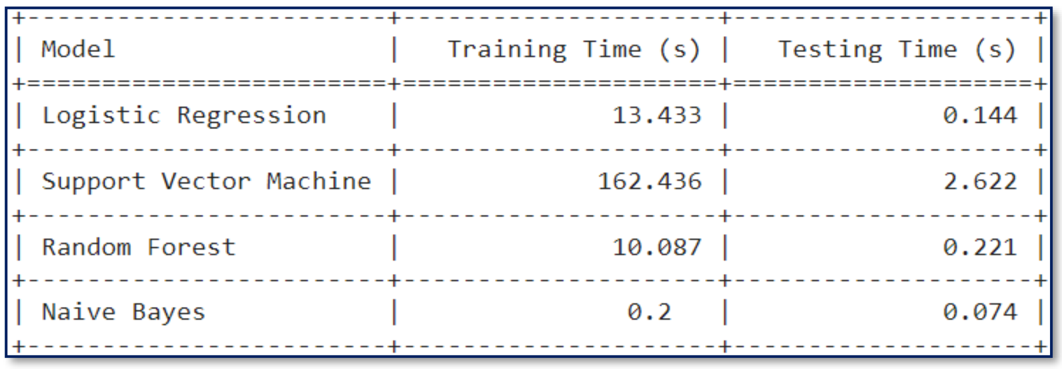

Dataset
EDA
Text Processing
Feature Engineering
Model Training
Model Testing
Model Comparison

Results of ML Model Implementation

The SVM model achieved the highest accuracy of 82.38% among all models, likely due to its ability to capture complex relationships in the data using a linear kernel.



The two models with the minimum time complexity are Naive Bayes and Logistic Regression. Naive Bayes achieves this due to its simple probabilistic model, which requires minimal computation during both training and prediction. Similarly, Logistic Regression's linear nature and convex optimization lead to efficient training and inference times, making it suitable for large-scale datasets and real-time applications.



```
+------------------------+---------------------+--------------------+
| Model                  |   Training Time (s) |   Testing Time (s) |
+========================+=====================+====================+
| Logistic Regression    |              13.433 |              0.144 |
+------------------------+---------------------+--------------------+
| Support Vector Machine |             162.436 |              2.622 |
+------------------------+---------------------+--------------------+
| Random Forest          |              10.087 |              0.221 |
+------------------------+---------------------+--------------------+
| Naive Bayes            |               0.2   |              0.074 |
+------------------------+---------------------+--------------------+
```

Conclusion

The sentiment analysis project successfully utilized NLP techniques to classify app store reviews into positive and negative sentiments, focusing on scores 4 and 5 for positive and scores 1 to 3 for negative sentiments. The SVM model achieved the highest accuracy of 82.38%, demonstrating its ability to capture complex relationships in the data using a linear kernel. Additionally, Naive Bayes and Logistic Regression exhibited minimum time complexity, making them efficient choices for real-time sentiment analysis tasks. Overall, the project demonstrated scalability in handling large textual datasets and provided valuable insights into customer sentiments towards various applications on the Google Play Store.

Group Members :

1.Vruksheeka Deshmukh	202101070058
2. Swastik Gaikwad	202101070062
3. Sanika Thakare	202101070120