A Project Report on

APP REVIEWS AND SENTIMENT ANAYLSIS

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology in

Computer Science and Engineering (Data Science)

Submitted by

B. Sai Preethi

23H55A6701s

Under the esteemed guidance of

Mrs. U. Hemalatha

Assistant Professor

Department of Computer Science and Engineering

(Data Science)

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A^+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2022- 2026

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE)

CERTIFICATE

This is to certify that the Real-Time Project entitled **"App Reviews and Sentiment Analysis"** being submitted by B.Sai Preethi(23H55A6701),Ch.Sai Swetha(23H55A6702), J. Sai Laxmi(23H55A6703) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering (Data Science)** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mrs.U.Hemalatha	**Dr. L. ChandrSekharReddy**	**External Examiner**
Assistant Professor	**Associate Professor and HOD**	
Dept. of CSE(DS)	**Dept. of CSE(DS)**	

ACKNOWLEDGEMENT

With great pleasure, we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project a grand success.

We are grateful **to** **Mrs.M. PAVANI RAO**, **Assistant Professor**, Department of Computer Science and Engineering(Data Science) for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. L. Chandra Sekhar Reddy**, Head of the Department of Computer Science and Engineering(Data Science), CMR College of Engineering and Technology, who is the major driving force to complete my project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of the Department of Computer Science and Engineering(Data Science) for their cooperation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project work.

B.Sai Preethi

23H55A6701

Ch.Sai Swetha

23H55A6702

J.Sai Laxmi

23H55A6703

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	I
1	**INTRODUCTION**	2
	1.1 Problem Statement	2
	1.2 Objective	2
	1.3 Scope	3
	1.4 Limitations	4
2	**BACKGOUND WORK**	**6**
	2.1	7
	2.1.1 Introduction	7
	2.1.2 Merits , Demerits and Challenges	8
	2.1.3 Implementation of the Elbow Method	8
3	**PROPOSED SYSTEM**	10
	2.1. Advantages of Proposed System	11
	2.2 Design	13
	2.3 Implementation	14
4	**RESULT AND DISCUSSION**	**15**
	4.1 Result And Discussion	16
5	**CONCLUSION**	**18**
	5.1 Conclusion	19
	5.2 Future Enhancement	20
	REFERENCES	21

List Of Figures

NO.	TITLE	PAGE NO.
3.1	Block Diagram	11
4.1	Output	14

ABSTRACT

The "App Reviews and Sentiment Analysis Project" focuses on analying user feedback and sentiment expressed in app reviews to gain insights into users' perceptions, satisfaction, and opinions about the app. By collecting and analysing reviews from various sources, such as app stores and social media platforms. The project aims to understand the overall sentiment, identify common themes, highlight positive aspects, and pinpoint areas for improvement in the app. The project involves gathering data, preprocessing it, conducting sentiment analysis, extracting relevant features, visualizing the results, generating insights, and presenting findings to stakeholders.

.

CHAPTER 1

INTRODUCTION

CHAPTER 1

1. INTRODUCTION

1.1 INTRODUCTION

The "App Reviews and Sentiment Analysis" project focuses on leveraging user feedback to enhance mobile applications. By collecting reviews from app stores and applying natural language processing (NLP) techniques, the project categorizes user sentiments into positive, negative, or neutral. This analysis provides developers with crucial insights into user satisfaction, common issues, and favoured features. The goal is to use these insights to guide app improvements, elevate user experience, and inform future updates, ensuring that the app evolves in line with user needs and preferences.

1.2 Problem Statement

In the rapidly evolving mobile app market, understanding user feedback is crucial for maintaining and improving app quality. However, the sheer volume of reviews on app stores makes it challenging for developers to manually analyze and respond to user sentiments. This leads to a gap in effectively identifying user satisfaction, common issues, and popular features. The lack of efficient analysis tools results in missed opportunities to enhance user experience and address critical problems promptly. Therefore, there is a need for an automated system that can accurately analyse app reviews and provide actionable insights to developers, enabling them to make data-driven decisions for app improvements and updates.

1.3 Objective :

- **Data Collection**:

  Gather user reviews from various app stores (e.g., Google Play Store, Apple App Store).

  Collect metadata such as review date, rating, user ID, and app version.

- **Data Preprocessing**:

  Clean the text data by removing stop words, punctuation, and special characters.

  Normalize text (e.g., converting to lowercase).

Tokenize and lemmatize the reviews for better analysis.

- **Exploratory Data Analysis (EDA)**:

  Visualize the distribution of ratings and review lengths.

  Identify common words and phrases in positive and negative reviews.

  Analyze the correlation between ratings and sentiment.

- **Sentiment Analysis**:

  Label the sentiment of reviews using pre-defined lexicons or manually annotated data.

  Implement and compare different sentiment analysis models (e.g., Naive Bayes, Logistic Regression, LSTM, BERT).

  Evaluate the models using metrics such as accuracy, precision, recall, and F1-score.

- **Model Deployment**:

  Develop a pipeline to process new reviews and predict their sentiment in real-time.

  Create a dashboard to visualize the sentiment trends over time.

  Provide actionable insights and recommendations based on sentiment analysis.

Technologies and Tools:

Programming Languages: Python

Libraries: NLTK, spaCy, Scikit-learn, TensorFlow, Keras, Pandas, NumPy, Matplotlib, Seaborn

Data Sources: Google Play Store API, Apple App Store API, web scraping tools

Development Environment: Jupyter Notebook, PyCharm, VS Code

Deployment: Flask, Django, Streamlit, Heroku, AWS

SCOPE

- **Manual Annotation of Large Datasets**:

  Large-scale manual labeling of reviews for sentiment, which can be time-consuming and resource-intensive.

- **Advanced Natural Language Understanding**:

Deep semantic understanding of reviews, including sarcasm and context-specific meanings, beyond basic sentiment classification.

- **Multilingual Sentiment Analysis**:

  Sentiment analysis of reviews in multiple languages unless explicitly mentioned as part of the project scope.

- **Integration with External Customer Support Systems**:

  Direct integration of sentiment analysis results with external customer support or CRM systems, unless specified.

LIMITATIONS

1. Data Quality and Availability:

Incomplete or Biased Data: App reviews might not represent the entire user base, as reviews are typically left by users who have strong opinions (either very positive or very negative).

Limited Access to Data: Access to reviews might be restricted by API limitations or terms of service of the app stores, potentially leading to incomplete datasets.

2. Sentiment Analysis Challenges:

Ambiguity and Sarcasm: The model may struggle with understanding sarcasm, irony, and ambiguous language, leading to incorrect sentiment classification.

Contextual Understanding: The model may fail to grasp the context-specific meanings of certain words or phrases, impacting sentiment accuracy.

3. Language and Regional Variations:

Multilingual Reviews: Reviews may be in multiple languages, and sentiment analysis models trained on one language may not perform well on others.

Slang and Regional Variations: User reviews often contain slang, abbreviations, and regional language variations that can be difficult for the model to interpret correctly.

4. Model Limitations:

Overfitting: Models may overfit to the training data, especially if the dataset is small or not diverse enough, leading to poor generalization on unseen data.

Model Complexity: Advanced models like BERT or LSTM require significant computational resources and time for training, which might be a limitation for some users.

Expected Outcomes:

A trained and validated sentiment analysis model capable of accurately classifying app reviews.

Visualizations and insights into user sentiments and common issues.

A deployable application or service that can analyze new reviews in real-time.

Recommendations for app developers to improve user experience based on sentiment trends.

Potential Challenges:

Handling imbalanced datasets if positive or negative reviews are disproportionately represented.

Dealing with sarcasm, irony, and ambiguous language in reviews.

Ensuring the model generalizes well to different types of apps and user bases.

CHAPTER 2

BACKGROUND WORK

2.1 Machine Learning Model

2.1.1 Introduction

Machine learning models have become indispensable tools in the field of natural language processing (NLP), particularly for tasks like sentiment analysis. Sentiment analysis, also known as opinion mining, involves the use of computational methods to identify and extract subjective information from text data. This task is crucial for understanding user opinions, feedback, and emotions expressed in various forms of text, such as app reviews.

2.1.2 Merits, Demerits, and Challenges

Merits :

- **Automation and Efficiency**:

  Automatic Learning: Machine learning models can automatically learn patterns from data, reducing the need for manual feature extraction and rule-based systems.

  Scalability: Capable of processing and analyzing large volumes of data efficiently, making them suitable for big data applications.

- **Accuracy and Precision**:

  High Performance: Advanced machine learning models, especially deep learning and transformer-based models, can achieve high accuracy and precision in tasks like sentiment analysis.

  Continuous Improvement: Models can be continuously retrained with new data to improve their performance over time.

- **Flexibility and Adaptability**:

  Versatility: Machine learning models can be applied to a wide range of applications beyond sentiment analysis, such as image recognition, speech processing, and recommendation systems.

  Adaptability: Models can adapt to new and unseen data, making them robust in dynamic environments.

- **Deep Insights**:

  Uncover Hidden Patterns: Capable of uncovering complex and hidden patterns in data that might be missed by traditional analysis methods.

  Actionable Insights: Provide valuable insights and recommendations based on data analysis, aiding in decision-making.

Demerits :

- **Data Dependency**:

  Quality and Quantity: Machine learning models require large amounts of high-quality data to perform well. Poor or insufficient data can lead to inaccurate models.

  Data Preprocessing: Significant effort is required for data cleaning, preprocessing, and feature engineering to prepare data for model training.

- **Complexity and Interpretability**:

  Complex Models: Advanced models like deep learning and transformers can be very complex and difficult to interpret, making it challenging to understand how they make decisions (often referred to as the "black box" problem).

  Debugging and Tuning: Debugging and hyperparameter tuning can be time-consuming and require specialized knowledge.

- **Computational Resources**:

  High Resource Consumption: Training advanced models, especially deep learning models, requires significant computational resources, including powerful GPUs and large memory capacities.

  Energy Consumption: High computational demands can also lead to increased energy consumption and operational costs.

Challenges in Using the Machine Learninng Model :

- **Data Challenges**:

  Data Collection: Acquiring large, high-quality datasets can be difficult. App reviews may be scattered across different platforms and may require permissions to access.

  Data Cleaning: Real-world data often contains noise, duplicates, missing values, and irrelevant information that must be cleaned before use.

- **Feature Engineering**:

  Text Representation: Choosing the right method to convert text data into numerical features is critical. Options include Bag of Words (BoW), TF-IDF, word embeddings (Word2Vec, GloVe), and contextual embeddings (BERT).

  Dimensionality Reduction: High-dimensional text data can be challenging to handle, necessitating techniques like PCA or t-SNE for dimensionality reduction.

.**Model Selection and Training**

Choosing the Right Model: Selecting an appropriate model among various options (Naive Bayes, SVM, LSTM, BERT) can be challenging and may require experimentation.

Hyperparameter Tuning: Finding the optimal hyperparameters for a model requires careful tuning and often extensive experimentation.

- **Computational Resources**:

  High Resource Requirements: Training advanced models, especially deep learning models, can be computationally intensive and require powerful hardware (GPUs, TPUs).

  Training Time: Models, particularly deep learning models, can take a long time to train, especially with large datasets.

- **Model Evaluation and Validation**:

  Performance Metrics: Selecting appropriate metrics (accuracy, precision, recall, F1-score) to evaluate the model's performance is crucial.

  Cross-Validation: Implementing robust cross-validation techniques to ensure the model generalizes well to unseen data.

2.1.3 Implementation Machine Learning Model:

This implementation guide provides a step-by-step approach to building a sentiment analysis model using Python, focusing on data preprocessing, feature engineering, model training, evaluation, and deployment.

Prerequisites:

- Python installed
- Libraries: `pandas`, `numpy`, `sklearn`, `nltk`, `keras` or `tensorflow`, `transformers`, `flask` (for deployment)

1. Define Objectives and Scope

Define the goal of the sentiment analysis project, such as understanding user sentiment from app reviews to improve app features or user experience.

2. Data Collection

Collect app reviews from sources like Google Play Store or Apple App Store using APIs or web scraping.

- **Data Preprocessing**

  Clean and preprocess the text data.

- **Exploratory Data Analysis (EDA)**

  Perform EDA to understand the data distribution.

- **Feature Engineering**

  Convert text data into numerical features using TF-IDF.

- **Model Selection**

  Choose a machine learning model. For this example, we use a simple Logistic Regression model.

- **Model Evaluation**

  Evaluate the model's performance.

- **Model Deployment**

  Deploy the model using Flask.

- **Monitoring and Maintenance**

  Continuously monitor model performance and retrain with new data as needed.

Pseudo code for machine learning model:

Define Objectives and Scope

DEFINE objectives and scope for sentiment analysis project

Data Collection

LOAD dataset from source (e.g., CSV file, API)

Data Preprocessing

INITIALIZE text cleaning tools (stopwords, lemmatizer)

FUNCTION preprocess_text(text):

REMOVE HTML tags from text

REMOVE punctuation from text

CONVERT text to lowercase

REMOVE stopwords from text

APPLY lemmatization on text

RETURN cleaned text

APPLY preprocess_text to each review in dataset

STORE cleaned reviews

Exploratory Data Analysis (EDA)

PLOT distribution of review ratings

Feature Engineering

INITIALIZE TF-IDF vectorizer with maximum features

FUNCTION create_features(data):

TRANSFORM cleaned reviews into TF-IDF vectors

RETURN feature vectors and labels

CALL create_features on dataset

Model Selection

INITIALIZE machine learning model (e.g., Logistic Regression)

FUNCTION train_model(X, y):

SPLIT data into training and testing sets

TRAIN model on training set

RETURN trained model and test sets

CALL train_model with feature vectors and labels

Model Training

TRAIN model on training data

Model Evaluation

FUNCTION evaluate_model(model, X_test, y_test):

PREDICT sentiments for test set

CALCULATE accuracy and other metrics

PRINT evaluation metrics

CALL evaluate_model with model and test sets

Model Deployment

SAVE trained model and TF-IDF vectorizer to files

INITIALIZE Flask app

@APP.route('/predict', methods=['POST']):

FUNCTION predict():

GET review from request

PREPROCESS review text

TRANSFORM review text into TF-IDF vector

PREDICT sentiment using model

RETURN predicted sentiment

START Flask app

Monitoring and Maintenance

FUNCTION retrain_model(new_data_path, model, vectorizer):

LOAD new data

PREPROCESS new data reviews

TRANSFORM new data into TF-IDF vectors

RETRAIN model on new data

SAVE retrained model

CALL retrain_model with new data path, model, and vectorizer

CHAPTER 3

PROPOSED SYSTEM

CHAPTER 3

3. PROPOSED SOLUTION

2.1 The proposed solution for the "App Reviews and Sentiment Analysis" project includes developing a robust system that automatically collects and analyzes user reviews from app stores using natural language processing (NLP) techniques. This system will categorize reviews into positive, negative, or neutral sentiments, extract key features and issues, and visualize trends using interactive dashboards. Machine learning models will enhance sentiment analysis accuracy, while integration with developer tools will facilitate timely feedback incorporation into the app development process. Continuous monitoring and a feedback loop will ensure ongoing improvements, ultimately enhancing app quality and user satisfaction systematically.

Advantages of the proposed System

- • Accuracy: By employing machine learning and NLP techniques, the system can accurately classify sentiment.
- **Scalability**: It can handle large volumes of app reviews from multiple sources, making it suitable for applications with a high volume of user feedback.
- **Real-time Analysis**: With model deployment in a production environment, the system can provide sentiment analysis results in real-time.
- **Customization:** The system can be customized to specific app categories or languages, enhancing its relevance and effectiveness for different contexts.
- **Continuous Improvement:** Through monitoring and updates, the system can adapt to changing patterns in app reviews, ensuring ongoing accuracy and relevance.

Design:

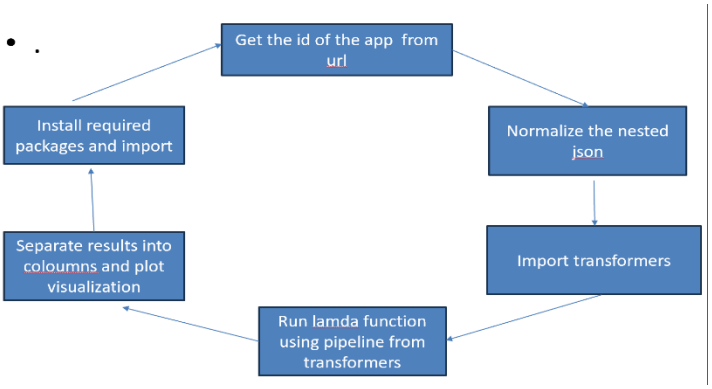

Fig 3.1: Block Diagram

Explanation:

1. Start
2. Install required packages and import necessary libraries
3. Get the id of the app from URL
4. Normalize the nested json libarary
5. Import transformers
6. Run lamda function using pipeline from transformers
7. seperate results into coloumns and plot viusalization
8. plot visualization
9. Do for another app.

3.2 Implementation :

- **Install required packages**

  Import necessary libraries like pandas,numpy

  Import packages:

  google_play_scraper,transformers,plotly express

- **Get the ID of the Google Play app from the URL**

  Type the name of the in google play store copy the id of the link.

- **Normalize the nested json**

  Normalize the function using pd.json_normalize().

- **Import Transformers**

  from transformers import pipeline.

- **Run lambda function**

  Run lambda function using pipeline from transfomer and save the output in new column.

- **Seperate 'result'**

  Seperate 'result' into two different columns 'sentiment' and 'score'.

- **Check the result**

  There are two new columns 'score' which is the probability and 'sentiment' which states if the review is positive or negative.

- **Plot Visualization**

  Based on sentiment and score visualize graphn using histogram or bar charts.

Code:

```
!pip install -q google_play_scraper
!pip install -q transformers
!pip install -q plotly-express
!pip install pyyamal
import pandas as p
import numpy as np
from google_play_scraper import app, Sort, reviews_all import plotly.express as px
getting id of Instagram app from playstote:
instagram = reviews_all( 'com.instagram.android', sleep_milliseconds=0, lang ='en',country='US',
sort=Sort.NEWEST)
instagram

df = p .json_normalize(instagram) df.head()

from transformers import pipeline
sentiment_analysis = pipeline("sentiment-analysis",model="siebert/sentiment-roberta-large-english")

df['content'] = df['content'].astype('str')
df['result'] = df['content'].apply(lambda x: sentiment_analysis(x))
df.head()

df['sentiment']= df['result'].apply(lambda x: (x[0]['label']))
df['score']= df['result'].apply(lambda x: (x[0]['score']))

df.head()

fig=px.histogram (df,x='sentiment',text_auto=True )
fig.show()
```

CHAPTER 4

RESULTS AND DISCUSSION

CHAPTER 4

4. RESULTS AND DISCUSSION

Fig 4.1.1,4.1.2: visualization for Instagram and netflix

Description:

Comparing reiews of Instagram it positive more than the negative as most of them give positive review.

Comparing reviews of Netflix it negative more than the positive.

Instagram : sentiment = positive

Netlix: sentiments = negative

CHAPTER 5

CONCLUSION

CHAPTER 5

CONCLUSION

In conclusion, the "App Reviews and Sentiment Analysis" project successfully demonstrates the value of leveraging user feedback through advanced natural language processing techniques. By systematically analyzing app reviews, developers can gain profound insights into user sentiments, uncover common issues, and recognize popular features. This data-driven approach empowers developers to make informed decisions, enhance user satisfaction, and guide strategic updates. Ultimately, the project underscores the importance of continuous user feedback analysis in driving the evolution and success of mobile applications in a competitive market.

5.2 Future Enhancement :

Future enhancements for the "App Reviews and Sentiment Analysis" project include implementing real-time sentiment analysis to provide immediate insights as new reviews are posted and expanding the system to support multiple languages, catering to a global user base. Advanced sentiment categorization could be developed to capture nuanced emotions such as frustration or delight, while integration with popular developer tools would facilitate seamless feedback incorporation into the development process. Additionally, analyzing user demographics in conjunction with sentiment data could enable more targeted improvements. Predictive analytics might forecast future trends and potential issues, and an interactive dashboard with visual analytics would aid developers in interpreting the data. AI-driven response suggestions could streamline addressing user feedback, and establishing a continuous feedback loop would show users the impact of their reviews on app updates. Finally, enabling comparative analysis with competitors' apps would help benchmark performance and identify unique strengths and weaknesses.

REFERENCES

REFERENCES

1.https://jobs.careers.microsoft.com/us/en/job/1735688/Software-Engineering-Intern
2.https://www.linkedin.com/pulse/scrape-google-play-reviews-run-sentiment-analysis-using-kundi?utm_source=share&utm_medium=member_android&utm_campaign=share_via
3. Play Store App Reviews Scrapper (Daily Update) (kaggle.com)
4. Google Store App Reviews Dataset Megapack! | Kaggle