ICS 2312:

Software Systems Development

B.Sc. Computer Science
3.2 (January – May/2024)

BACKGROUND INFORMATION:

This document captures the working of Moodlens, a sentiment analysis application.

Submitted: February, 2025

Proposal for: Moodlens

Authors: Group A7, Version: 1.0

REGISTRATION NUMBER	STUDENT NAME
SCT211-0079/2022	**Joram Kireki**
SCT211-0848/2018	**Jany Muong**
SCT211-0070/2022	**Vincent Ochieng'**
SCT211-0504/2021	**Gatmach Yuol**
SCT211-0535/2022	**Akech Atem**
SCT211-0003/2022	**Josphat Thumi**

A Sentiment Analysis Application.

PART 1: INTRODUCTION

This is what **moodlens** does – it is a sentiment analysis application designed to allow users to analyze the sentiment of text inputs - short texts that take the shape of tweets, in real-time. It provides insights into whether a given text is **positive** or **negative**, offering a user-friendly interface for exploring sentiment trends. The app uses **machine learning** models for sentiment analysis and is built using **Django** for the backend, **PostgreSQL** for database management, and **React** for the frontend. The **React** frontend should consume data from the backend as well as communicate with the backend smoothly.

PART 2: PROBLEM STATEMENT

Social media, for example Twitter, serves as a powerful tool for public discourse, yet analyzing the sentiment of large volumes of tweets remains a challenge. Traditional methods rely on manual interpretation or basic keyword matching, which often leads to inefficiencies and inaccuracies in sentiment classification. We are building a platform that **mimics social media** that serves as central place **for sentiment analysis**.

Challenges Addressed by MoodLens:

- **Scalability:** traditional analysis methods struggle to process millions of tweets efficiently.
- **Subjectivity:** human interpretation is prone to bias, leading to inconsistent sentiment classification.
- **Real-time Insights:** there is a growing need for instant sentiment analysis of trending topics and breaking news.
- **Lack of Contextual Understanding:** simple keyword-based models fail to grasp nuances, sarcasm, and evolving slang within tweets.

MoodLens addresses these challenges by providing a robust AI-driven sentiment analysis tool that is fast, accurate, and scalable for analyzing tweets.

PART 3: JUSTIFICATION FOR PROJECT

Sentiment analysis is becoming increasingly important for various industries, including marketing, politics, finance, and social research. Organizations and researchers can rely on this tool to analyze tuser sentiment dynamically, about products and topics of interests.

Why MoodLens?

- **Accuracy:** utilizes machine learning to achieve high accuracy in classifying tweet sentiment.
- **Efficiency:** processes vast amounts of tweets in real time, reducing manual effort.
- **Scalability:** designed to handle trending hashtags and viral topics efficiently.

By adopting MoodLens, individuals and organizations can derive meaningful insights from Twitter data, enabling data-driven decision-making and sentiment tracking.

PART 4: OBJECTIVES AND CORE FEATURES

We want to come up with an application that satisfies the objectives below – these objectives are closely correlated with how the system functions holistically:

1. **To integrate a machine-learning model seamlessly for use**: we aim to create an application that works correctly to provide feedback to a user of the app on whether text that the model accepts is positive or negative. This should be done with ease by the model.
2. **To develop a User-Friendly Interface**: the interface should be easy to navigate, allowing users to input text and view sentiment predictions quickly.
3. **To track prediction history efficiently**: users can view the last five analyzed texts and their corresponding sentiment scores.
4. **To implement a robust data management application**: we want to store user login data and sentiment analysis results in a secure manner using PostgreSQL for our DBMS.
5. **To create user-specific/personalized sexy visualizations**: display visual representations (like graphs) of the sentiment of recent texts entered by the user – e.g. graphs for the last(eg 5) saved and analyzed texts.

The functionality of the application is described briefly below:

1. **Sentiment Analysis:** we use trained and fine-tuned NLP-based machine learning model specifically for analyzing text.
2. **Create an Interactive Dashboard:** a user-friendly interface displaying sentiment trends
3. **User Management:** support user registration, login, and logout.

4. **Data Storage:** store user login data and sentiment analysis results in a secure manner using PostgreSQL for our DBMS.
5. **Visualization:** display visual representations (e.g., graphs) of the sentiment of recent texts entered by the user.

PART 5: RESOURCES

We will need the resources outlined below for our application:

5.1 Development Environment:

The **moodlens** application will be developed using a development environment with these setup and/or requirements.

- Python 3.11, Django 5.0, TensorFlow 2.17, PostgreSQL 14, Node.js 16.x for React.

5.2 Model Training Process:

We have already started with the model. The sentiment analysis machine learning model was trained using a dataset called **Sentiment140** publicly available on the internet and Kaggle. The **Sentiment140 dataset** has a large number of tweets collected from multiple Twitter users, and using text vectorization and a neural network architecture built around natural language processing to achieve over 80% accuracy. The purpose of this is to get our model to learn enough context on how to get sentiment from text so that it can learn those thought patterns and when it sees new text from a user of the app it can detect emotions with ease.

Please have a look below of the process of training. It shows how the model **accuracy** is improving with each training step:

2000/2000 ━━━━━━━━━━━━━━━━━━━━ **6s** 2ms/step - accuracy: 0.7349 - loss: 0.5207 - val_accuracy: 0.7972 - val_loss: 0.4355
Epoch 2/20
2000/2000 ━━━━━━━━━━━━━━━━━━━━ **3s** 2ms/step - accuracy: 0.8310 - loss: 0.3789 - val_accuracy: 0.7994 - val_loss: 0.4429
Epoch 3/20
2000/2000 ━━━━━━━━━━━━━━━━━━━━ **3s** 2ms/step - accuracy: 0.8800 - loss: 0.2836 - val_accuracy: 0.7916 - val_loss: 0.4876
Epoch 4/20
2000/2000 ━━━━━━━━━━━━━━━━━━━━ **3s** 2ms/step - accuracy: 0.9243 - loss: 0.1873 - val_accuracy: 0.7831 - val_loss: 0.6120
Epoch 5/20
2000/2000 ━━━━━━━━━━━━━━━━━━━━ **3s** 2ms/step - accuracy: 0.9487 - loss: 0.1257 - val_accuracy: 0.7749 - val_loss: 0.7757
Epoch 6/20
2000/2000 ━━━━━━━━━━━━━━━━━━━━ **3s** 2ms/step - accuracy: 0.9642 - loss: 0.0896 - val_accuracy: 0.7698 - val_loss: 0.9622
Model saved to keras/sentient73.keras and vocabulary saved to keras/vectorizer_vocab.txt

5.3 Sentiment140 Dataset:

This dataset contains 1,600,000 tweets extracted using the Twitter API. The tweets have been annotated (0 = negative, 4 = positive) and they can be used to detect sentiment.

It is a made up of comma separated values and contains the following 6 fields:

1. **polarity**: the polarity of the tweet (*0* = negative, *2* = neutral, *4* = positive)
2. **ids**: the id of the tweet (*2087*)
3. **date**: the date of the tweet (*Sat May 16 23:58:44 UTC 2009*)
4. **flag**: the query (*lyx*). If there is no query, then this value is NO_QUERY.
5. **user**: the user that tweeted (*robotickilldozr*)
6. **text**: the text of the tweet (*Lyx is cool*)

Snippet:
This is what the dataset looks like

0	1	2	3	4	5
4	1972002925	Sat May 30 08:21:22 PDT 2009	NO_QUERY	noobpwned	Started my morning off with a bloody nose. H...
0	2013837538	Tue Jun 02 23:42:53 PDT 2009	NO_QUERY	missjnz	@lawrencebrown must be a sign you need a holi...
0	1992941765	Mon Jun 01 09:34:24 PDT 2009	NO_QUERY	cupcake147	omg i have to go to summer school!
4	2054702495	Sat Jun 06 07:54:59 PDT 2009	NO_QUERY	heya10	Getting ready to go skydiving. How exciting to...
0	2030659197	Thu Jun 04 08:41:44 PDT 2009	NO_QUERY	Sciteg	As i read a few minutes ago... the visit of Ob...

5.4 Technical Resources:

We will use the following technologies to build the application:

- **Machine Learning Frameworks**: TensorFlow.
- **Development Stack**: Django (backend), React (frontend), PostgreSQL (database).
- **Version Control**: GitHub for collaborative development.

5.5 Human Resources:

- **Data Scientists**: for model training and fine-tuning.
- **Backend Developers**: to develop the API and database management.
- **Frontend Developers**: to design the dashboard interface.
- **Security Experts**: to ensure data privacy compliance.

PART 6: CONCLUSION

The *MoodLens* application represents a thoughtful and innovative experiment on sentiment analysis, aiming to provide users with an understanding and analyzing emotions in text. Through the integration of machine learning techniques, *MoodLens* is able to accurately classify text sentiment empowering users to gain insights into their own **mood** or the emotions conveyed in online content.

The system is designed with scalability and security in mind. The application ensures high performance, responsive design, and easy integration of future features. The PostgreSQL database securely stores user data and sentiment analysis results, allowing users to track their sentiment history over time.

Security is a core tenet of the *MoodLens* application. With authentication mechanisms such as JWT, password hashing, and multi-factor authentication, as well as strong encryption for data at rest and in transit, users can trust that their data is handled securely.

In conclusion, *MoodLens* stands as a promising tool for sentiment analysis, with a solid foundation for future growth. The design and technical architecture are adaptable, allowing the integration of advanced features such as personalized sentiment tracking and expanded visualizations. With continued refinement, *MoodLens* has the potential to evolve into a powerful tool for **mental wellness** and **emotional insight**, benefiting both individual users and larger audiences in **social media**.