Assignment 1

Group - CS31

Submitted by

Aditya Srivastava (2201013)

Deepanika Gupta (2201062)

Kartik Saini (2201103)

Q1). What key aspects need to be analyzed regarding the project's feasibility study?

Feasibility Study

Technical Feasibility:

1. **Tools and Frameworks:**
    - Backend: Python (Django)
    - Frontend: HTML,CSS,JS, React.js,
    - NLP Libraries: NLTK, spaCy, TextBlob, Transformers(HuggingFace)
    - Visualization Tools: Matplotlib
    - APIs: API for fetching real-time tweets.
    - Backup and Recovery: GitHub for version control and backup of modularity and models, ensuring easy recovery and collaboration across different team members.
2. **Scalability:**
    - Handling more users using load balancing and scaling
    - Performing newly demanded tasks using modular architecture.
    - Scaling the frontend using techniques like lazy loading, pagination.
    - Adapting to various workloads through efficient data handling and real time analysis.

Schedule Feasibility:

1. **Timeline & Milestones:**
    - Week 1-2: Requirement gathering and feasibility analysis and api setup.
    - Week 3-4: Data Collection, cleaning and preprocessing.
    - Week 5-6: Training sentiment and emotion analysis model along with unit testing.
    - Week 7-8: Implementing toxicity detection module along with unit testing.
    - Week 9-10: Preparing frontend of web app and integration with backend.
    - Week 11: Testing, bug fixing, performance tuning.
    - Week 12: Final presentation and documentation.

Quality Feasibility:

1. Unit, integration, and system testing to ensure correctness.
2. Regular model updates for accuracy and API monitoring for uptime.
3. Integration of new features.

Market Feasibility:

1. **Demand Assessment:**
    - With the rise of social media use, sentiment and emotion analysis is in high demand for marketing, politics, and brand reputation management.
    - Toxicity detection is crucial for content moderation especially for children.

2. **Target Customers:**
    - Marketing teams.
    - Political analysts.
    - Parents and educators seeking content filtering for kids.

Q2). Discuss the project requirements that the customer may provide, and analyze the gathered requirements.

Marketing Teams

- Require tools for real-time sentiment and emotion analysis of tweets to monitor brand perception and customer feedback.
- Need trending topic analysis and exportable sentiment reports for strategic decision-making.

Political Analysts

- Require sentiment and emotion analysis to track public opinion on policies and political figures.
- Require tools to analyze trends over time for better insights.

Parents and Educators

- Seek content filtering features to detect harmful or negative sentiments and ensure safe content for children.
- Require emotion-tuning suggestions to create and share emotionally positive content.

User-Friendly Interface

- The application should feature a simple, intuitive interface accessible to both technical and non-technical users.

Accuracy

- The system must accurately provide correct sentiment, emotion and toxicity classification to meet diverse user needs especially in the case of ambiguous tweets.

Analysis of the gathered requirements

The gathered requirements highlight diverse user needs, including real-time sentiment analysis for marketing teams, sentiment and emotion analysis for political analysts, and content filtering for parents and educators. Marketing teams focus on monitoring brand perception, analyzing trending topics, and generating exportable reports, which are feasible but require robust real-time data processing. Political analysts require tools for sentiment and emotion analysis with trend monitoring over time, which are achievable with accurate time-series analysis and pre-trained emotion models. Parents and educators prioritize content filtering for harmful content and emotion-tuning suggestions, emphasizing safety and positivity, though these features demand high precision in toxicity detection. A user-friendly interface is critical for adoption by technical and non-technical users, requiring intuitive design and visual clarity. Overall, accuracy in sentiment, emotion, and toxicity classification is paramount, particularly for handling ambiguous tweets, requiring fine-tuned models and efficient data handling to meet user expectations effectively.

Requirement	Feasibility	Challenges	Priority
Real-Time Sentiment Analysis	High	Handling large volumes in real-time and api call	High
Exportable Sentiment Reports	High	Generating detailed and visually appealing reports	Medium
Content Filtering	High	Balancing precision and recall	High
User-Friendly Interface	High	Balancing simplicity with functionality	High
Emotion-Tuning Suggestions	Medium	Generating context-appropriate suggestions	Low
Accuracy	High	Handling sarcasm and ambiguous tweets	High

Q3) Which Software Development Life Cycle (SDLC) model will you choose for your software engineering project? Please justify your choice. Write point-wise.

We have decided to use the **Iterative Model** for the development of our project. Below are the justifications for choosing this model:

1. **Incremental Development**
    - The iterative model allows the application to be developed in smaller, manageable iterations. Each iteration focuses on building a specific feature or component of the system, ensuring progress is continuous and measurable.
2. **Flexibility to Incorporate Changes**
    - Requirements for the project may evolve based on feedback from team members, TAs and course instructors. The iterative model provides the flexibility to incorporate these changes in subsequent iterations without affecting the overall project timeline.
3. **Early Risk Identification and Resolution**
    - Integration with external APIs like the Twitter API may pose challenges. The iterative approach enables us to address risks early by testing and resolving issues in smaller increments.
4. **Parallel Development by Multiple Team Members**
    - The iterative model enables parallel development, allowing team members to work on different aspects of a feature, such as sentiment analysis and content suggestion, simultaneously. This overlap improves coordination, ensuring faster feature delivery by allowing concurrent work on the user interface and backend logic, while minimizing bottlenecks.
5. **Continuous Feedback and Improvement**
    - After completing each iteration, the output can be reviewed and tested by team members, TAs and course instructors. This ensures continuous improvement and helps refine and add new features.
6. **Efficient Resource Utilization**
    - By dividing the project into smaller iterations, resources such as time and effort are utilized effectively. For example, the team can first focus on integrating the Twitter API, then move on to implementing user-generated content analysis and so on.
7. **Simplified Testing and Debugging**
    - Each iteration results in a functional module, making it easier to test and debug smaller components. This ensures the reliability of the application at every stage of development.