|  |  |
| --- | --- |
| **Project Title** | **Real-Time Sentiment Analysis for Customer Feedback Using Neural Networks and Streamlit App** |
| **Skills take away From This Project** | **Machine Learning and Deep Learning skills, Web Application Development, Cloud Deployment** |
| **Domain** | **Customer Service, AI & ML, NLP** |

**Objective:**

Develop a system that uses a Neural Network (NN) model to perform sentiment analysis on customer feedback provided through a web application. The application will allow users to input text (e.g., reviews, comments) into a webpage, click a "Predict" button, and receive real-time sentiment predictions (positive, negative, or neutral). The backend will be powered by Streamlit, a lightweight Python framework for building interactive web applications.

This project will enable students to create an end-to-end solution for sentiment analysis, leveraging modern neural network techniques and deploying the application as a standalone web app using Streamlit.

**Project Scope:**

• **End-to-End Development**: The project will involve building a neural network for sentiment analysis, integrating it with a user-friendly Streamlit-based interface, and deploying the application on cloud platforms like AWS or open-source alternatives.

• **Real-World Relevance**: This system can be applied in customer service, e-commerce platforms, or social media monitoring to analyze customer feedback and improve user experience.

**Key Components:**

**1. Neural Network Model Development:**

* Use a dataset like IMDb Movie Reviews or Sentiment140 to train a neural network for sentiment analysis.
* Build a model using techniques such as Embedding layers (e.g., Word2Vec, GloVe, or BERT embeddings).
* Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), or Gated Recurrent Units (GRUs).
* Alternatively, use pre-trained transformer models like BERT for fine-tuning.
* Train the model to classify text into three categories: Positive, Negative, or Neutral.
* Output: Train the model to classify text into three categories: Positive, Negative, or Neutral.
* Evaluation Metrics: Evaluate the model using accuracy, precision, recall, and F1 score.

**2. Streamlit Application Development**

* Develop an interactive web application using Streamlit to serve the trained neural network model.
* Key Features of the Streamlit App:
* A text box where users can input their feedback (e.g., reviews, comments).
* A "Predict" button to trigger sentiment analysis.
* Display the sentiment prediction (Positive, Negative, or Neutral) in real-time.
* Example Workflow:
* User Input: "The product is amazing!"
* Output: "Sentiment: Positive"

**3**. **Frontend Development:**

* Use Streamlit's built-in components to design a simple and intuitive user interface.
* Include features like:
* Input text box.
* Predict button.
* Clear button to reset the input.
* Display area for sentiment results.
* Enhance the UI with Markdown, CSS styling, and visualizations (e.g., progress bars or pie charts for sentiment distribution).

**4.** **Model Deployment:**

* Deploy the Streamlit application using cloud platforms such as:
* AWS: EC2, Lambda, or Elastic Beanstalk.
* Open-source Platforms: Heroku, Render, or Streamlit Community Cloud.

**5. Performance Evaluation and Optimization:**

* Evaluate the neural network model’s performance using metrics such as accuracy, precision, recall, and F1 score.
* Optimize the Streamlit app for low latency and high throughput.
* Explore techniques like model quantization or pruning to reduce inference time.

**Expected Results:**

**1. Fully Functional Web Application:**

* Users can input text and receive real-time sentiment predictions.
* The application will be accessible via a web browser.

**2. Codebase:**

* Well-documented Python scripts for all components, including neural network training, Streamlit app development, and deployment.

**3. Trained Models:**

* Pre-trained and fine-tuned neural network models for sentiment analysis.

**4. Project Report:**

* A comprehensive document covering system architecture, methodology, experimental results, and conclusions.

**5. Ethical Analysis:**

* A discussion on the ethical implications of sentiment analysis, including bias mitigation, user privacy, and interpretability of predictions.

**Tools and Technologies:**

* Programming Language : Python
* Frameworks and Libraries: TensorFlow/Keras or PyTorch (for neural network development).
* Hugging Face Transformers (optional, for pre-trained models like BERT).
* Streamlit (for building the web application).
* Deployment Platforms: AWS (EC2, Lambda, Elastic Beanstalk).
* Open-source platforms like Heroku, Render, or Streamlit Community Cloud.
* Datasets: IMDb Movie Reviews, Sentiment140, or custom datasets for training the model.

**Deliverables:**

**1. Application:**

* A fully functional Streamlit-based web application for sentiment analysis.

**2. Codebase:**

* Python scripts for neural network training, Streamlit app development, and deployment.

**3. Trained Models:**

* Pre-trained and fine-tuned neural network models for sentiment analysis.

**4. Project Report:**

* A detailed report covering system design, implementation, performance analysis, and potential applications.

**5. Ethical Analysis:**

* A discussion on the ethical implications of sentiment analysis, including bias mitigation, user privacy, and interpretability of predictions.

**Project Guidelines:**

• **Independent Work:** You are expected to complete all project components independently, demonstrating proficiency in neural networks, API development, and deployment.

• **Regular Milestones:** The project will be divided into milestones with regular check-ins to ensure steady progress.

• **Final Presentation:** You will present your project to faculty members, discussing the application’s features, challenges faced, and its potential real-world impact.

**Example Workflow:**

**1. User Interaction:**

A user enters the text: "The service was terrible, and I am very disappointed." They click the "Predict" button.

**2. System Processing:**

The Streamlit app processes the input using the neural network model.

The model predicts the sentiment (e.g., Negative).

**3. Output:**

The web page displays: "Sentiment: Negative."

This project combines neural network-based sentiment analysis with Streamlit app development and cloud deployment, offering a complete end-to-end solution.