**Software Requirements Specification (SRS) Document**

**Sentiment Analysis Web Application**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to define the software requirements for the development of a Sentiment Analysis Web Application using a Neural Network (NN) model. The application will allow users to input text-based feedback through a web interface and receive real-time sentiment predictions (Positive, Negative, or Neutral). The backend will be powered by Streamlit, a lightweight Python framework for interactive web applications.

**1.2 Scope**

The project aims to develop an end-to-end solution for sentiment analysis, including training a neural network model, integrating it with a Streamlit-based user interface, and deploying the application on cloud platforms like AWS or open-source alternatives. The system is applicable in various real-world scenarios, such as customer service, e-commerce, and social media monitoring.

**1.3 Definitions, Acronyms, and Abbreviations**

* **NN (Neural Network)**: A machine learning model used for pattern recognition.
* **NLTK(Natural Language Toolkit)**: leading platform for building Python programs to work with human language data.
* **VADER:** is a sentiment analysis tool specifically designed for social media text and short-form content.
* **Roberta Pretrained Model:** Robustly Optimized BERT Approach. Use a model trained of a large corpus of data. Transformer model accounts for the words but also the context related to other words.
* **BERT (Bidirectional Encoder Representations from Transformers)**: A transformer-based model for natural language processing (NLP) tasks.
* **Streamlit**: A Python framework for building interactive web applications.

**1.4 References**

* Amazon Reviews Dataset

**1.5 Overview**

This document outlines the functional and non-functional requirements of the sentiment analysis web application, along with system features and expected behavior.

**2. Overall Description**

**2.1 Product Perspective**

The system will be a standalone web application, enabling users to analyze textual feedback using an AI-powered sentiment analysis model. The model will be accessible via an interactive web interface, with results displayed in real-time.

**2.2 Product Functions**

* Accepts text input from users
* Processes input using a trained neural network model
* Classifies sentiment into Positive, Negative, or Neutral
* Displays sentiment results on the interface
* Provides accuracy metrics for model evaluation

**2.3 Assumptions and Dependencies**

* Users will provide meaningful text for sentiment analysis.
* The model will be pre-trained and fine-tuned before deployment.
* Streamlit will be used as the primary frontend framework.
* Deployment on AWS or an alternative cloud service.

**3. Specific Requirements**

**3.1 Functional Requirements**

**3.1.1 Neural Network Model Development**

* Train a sentiment analysis model using IMDb Movie Reviews or Sentiment140 dataset.
* Implement embedding technique Robustly Optimized BERT.
* Fine-tune pre-trained models like BERT for sentiment classification.
* Evaluate model performance using accuracy, precision, recall, and F1-score.

**3.1.2 Streamlit Application Development**

* Develop a web-based application using Streamlit.
* Provide a text input field for users to enter feedback.
* Implement a "Predict" button to trigger sentiment analysis.
* Display sentiment results (Positive, Negative, Neutral) in real-time.

**4. System Features**

**4.1 User Input Processing**

* Accepts raw text input from users.
* Preprocesses input before feeding it into the model.

**4.2 Sentiment Prediction**

* Executes model inference to classify input text into sentiment categories.
* Provides real-time results on the interface.

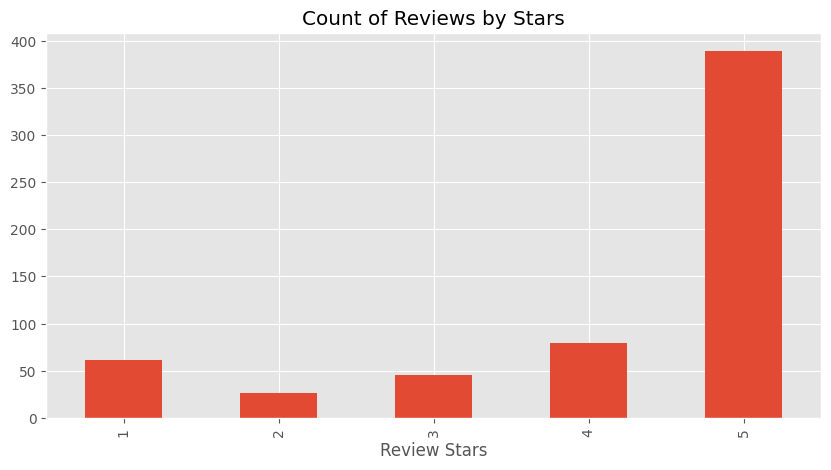
**4.3 Model Evaluation**

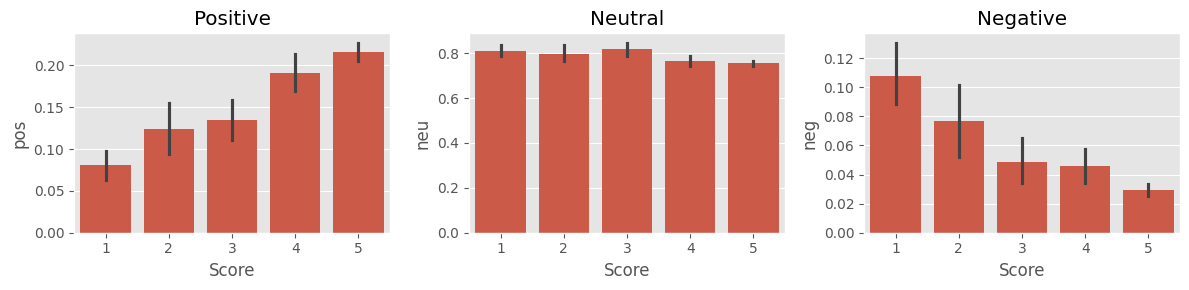
* Displays accuracy metrics for performance assessment.
* Allows retraining of models as needed.

**4.4 Deployment & Maintenance**

* Deploy the application on AWS or an alternative cloud provider.
* Provide regular updates and improvements.
* Implement logging and monitoring features for maintenance.

**5. EDA:**

****

****