**1. Introduction**

This document outlines the architecture, components, and deployment strategy for an AI/ML project utilizing HTML, SCSS, JavaScript, Flask for the frontend and backend, Python for the ML model, and Excel datasets for input.

**2. System Overview**

The AI/ML system consists of the following key components:

1. **Frontend** - A user interface built using SCSS, HTML, Flash & Java Script for data interaction and visualization.
2. **Backend** - API layer for processing requests and serving ML predictions.
3. **Machine Learning Model** - Python-based model for training and inference.
4. **Dataset** - Excel files used for training and inference.

**3. Architecture**

**3.1 System Components**

**Frontend (HTML, SCSS, JavaScript, Flask)**

* Users upload Excel datasets.
* Communicates with the backend via REST API.
* Displays predictions and insights using Chart.js / D3.js.

**Backend (Node.js & Express.js API)**

* Handles API requests from the frontend.
* Processes and validates Excel data.
* Calls the ML model API for predictions.
* Returns responses to the frontend.

**Machine Learning Model (Python & FastAPI/Flask)**

* Reads and preprocesses Excel files (Pandas, OpenPyXL).
* Trains ML models (Scikit-learn, TensorFlow).
* Exposes REST API for inference.

**Dataset & Storage**

* Excel files stored locally or in cloud storage (Google Drive, AWS S3).
* Preprocessed and structured before use.

**4. Workflow**

1. **User uploads Excel file via frontend.**
2. **Backend validates and preprocesses data.**
3. **Backend sends structured data to ML API.**
4. **ML model performs inference and returns predictions.**
5. **Backend forwards response to frontend.**
6. **User views predictions and insights in UI.**

**5. Technology Stack**

|  |  |
| --- | --- |
| **Component** | **Technology** |
| Frontend | Node.js, Express.js, React (optional) |
| Backend | Node.js, Express.js, FastAPI/Flask |
| ML Model | Python, Pandas, Scikit-learn, TensorFlow |
| Storage | Local file system, Google Drive, AWS S3 |
| API Communication | RESTful APIs |
| Deployment | Docker, Kubernetes, AWS/GCP/Azure |

**6. Deployment & Scaling**

* **Containerization**: Docker for packaging applications.
* **Orchestration**: Kubernetes for managing deployments.
* **Cloud Hosting**: AWS/GCP/Azure for scalability.
* **Monitoring**: Prometheus, Grafana for performance tracking.

**7. Security Considerations**

* **Authentication**: OAuth2, JWT for secure API access.
* **Data Encryption**: Secure storage and transmission of sensitive data.
* **Rate Limiting**: Prevent abuse of API endpoints.

**Model Performance and Results**

**8.1 Key Performance Metrics**

* **Accuracy: Measures the percentage of correct predictions.**
* **Precision & Recall: Evaluates the model’s effectiveness in classifying sentiment correctly.**
* **F1 Score: Balances precision and recall.**
* **Mean Squared Error (MSE) / Root Mean Squared Error (RMSE): Measures prediction error for regression tasks.**
* **ROC-AUC Score: Measures the classification performance.**

**8.2 Comparison Against Baselines**

* **Baseline Model: A simple logistic regression or random forest model for comparison.**
* **Improvement Over Baseline: Measure performance gains with advanced ML models like LSTM, CNN, or ensemble learning.**
* **Benchmarking with Industry Standards: Compare results against publicly available datasets and benchmarks.**

**8.3 Visualizations**

* **Confusion Matrix: Displays the performance of classification models.**
* **Precision-Recall Curves: Illustrates trade-offs between precision and recall.**
* **Stock Price Prediction vs. Actual Prices: Time-series plots comparing predictions and real market data.**
* **Sentiment Distribution: Pie charts or bar graphs showing sentiment categories.**
* **Feature Importance Graphs: Highlights influential features in predictions.**

**9. Conclusion**

**This technical documentation provides a comprehensive overview of the AI/ML system architecture, workflow, technology stack, and deployment considerations. Future enhancements may include improved ML models, real-time streaming data support, and AI-driven analytics.**