**Milestone 4:**

**Visualization & Dashboard Development**

**Project Name:** PrognosAI: AI-Driven Predictive Maintenance System Using Time-Series Sensor Data

**Dataset:** NASA Turbofan Jet Engine Data Set

**Prepared by:** Durga Veera Prasad V

## **Objective:**

The goal of this milestone is to build a fully interactive Streamlit-based dashboard to visualize Remaining Useful Life (RUL) predictions for turbofan jet engines using time-series sensor data. The dashboard integrates machine learning model predictions, data processing, and visual analytics for performance monitoring and decision-making.

**Data Conversion (Preprocessing in Jupyter Notebook):**

Before loading data into the dashboard, the original NASA CMAPSS dataset was provided in `.txt` format.

These files were converted into `.csv` format using a preprocessing script in Jupyter Notebook to make them compatible with Pandas and Streamlit.

## **Features Implemented:**

**1. Model Loading and Integration**

- Loads optimized LSTM deep learning model.

- Loads MinMaxScaler for feature scaling.

- Ensures compatibility with scikit-learn version >= 1.4 using feature metadata injection.

**2. Data Preprocessing**

- Automatically reads and processes CMAPSS test data (CSV or TXT).

- Handles missing Engine IDs and incomplete sensor data.

- Renames columns dynamically to maintain consistency across datasets.

**3. Sequence Creation and RUL Prediction**

- Segments data into time-series sequences per engine.

- Applies trained LSTM model to predict Remaining Useful Life (RUL).

- Categorizes predictions into \*\*Normal\*\*, \*\*Warning\*\*, and \*\*Critical\*\* alerts.

**4. Visualization Dashboard (Streamlit)**

- Interactive sidebar controls for sequence length and threshold adjustment.

- Displays detailed summary tables and plots:

- Average, minimum, and maximum RUL per test file.

- Pie chart of alert distribution.

- Scatter plot of Engine vs RUL values.

- Bar chart showing RUL distribution across engines.

- Export option for downloading results as CSV files.

**5. Performance and Usability Enhancements**

- Efficient caching of model and scaler using Streamlit resource caching.

- Robust exception handling for missing files or incompatible input data.

- Fully modularized code for better readability and scalability.

## **Module Usage:**

**Streamlit-** Builds the interactive dashboard interface.

**Pandas -** Handles data loading, transformation, and aggregation.

**NumPy -** Performs numerical operations on time-series data.

**TensorFlow / Keras -** Loads the pre-trained LSTM model and performs predictions.

**Joblib -** Loads the pre-saved MinMaxScaler objects.

**Plotly (Express & Graph Objects) -** Generates dynamic and interactive charts for data visualization.

**OS & Glob -** Manages file paths and batch file processing.

## **Folder and File Structure:**

Milestone-4/

├── CMAPSS/

│ └── converted\_csv/ # Contains test datasets like test\_FD001.csv

│

├── optimized\_lstm\_final.keras # Trained LSTM model

├── scalers\_all.pkl # Saved feature scalers

├── app\_prognosAI\_dashboard.py # Streamlit dashboard script

└── README.md # Project documentation

## **Running the Dashboard:**

**1. Ensure all required libraries are installed:**

**bash**

pip install streamlit pandas numpy tensorflow scikit-learn joblib plotly

**2. Place your test datasets in:**

C:\Users\DELL\OneDrive\Desktop\Milestone-4\CMAPSS\converted\_csv

**3. Launch the Streamlit dashboard:**

**bash**

streamlit run app\_prognosAI\_dashboard.py

```

**4. Open the provided local URL (usually http://localhost:8501) to view the dashboard.**

## **Future Improvements:**

- Add a file upload feature for custom datasets.

- Include trend visualization for RUL degradation over cycles.

- Implement automatic sequence length detection from model input.

- Integrate real-time data monitoring and alert notifications.

## **Conclusion:**

This milestone successfully integrates machine learning predictions, data visualization, and interactive analytics into a unified Streamlit dashboard for predictive maintenance. The system demonstrates how AI can effectively predict the Remaining Useful Life of jet engines using time-series sensor data.