SUBMITTED BY : JYOTI KHANCHANDANI

ANDREW ID: JKHANCHA

Introduction

The purpose of this project is to predict the Remaining Useful Life (RUL) of equipment based on sensor data. Predicting RUL enables proactive maintenance, minimizing downtime and repair costs. The approach used in this analysis combines data preprocessing, feature scaling, and labeling for supervised learning. An LSTM (Long Short-Term Memory) neural network model is used for time-series forecasting, which is well-suited for handling sequential data like equipment cycles and sensor measurements.

Approach and Methodology

1. **Data Loading and Initial Exploration:**
    - The first step involved loading the data into a DataFrame to inspect the columns and understand the distribution of the values. This exploratory data analysis (EDA) phase enabled the identification of key statistical patterns within the dataset.
    - The statistics were analyzed for each column to assess data quality, missing values, and variability. Columns with little to no variance (i.e., containing only a single value) or a high number of null values were marked for removal.
2. **Column Filtering and Data Cleaning:**
    - Several columns were removed due to low variability or redundancy, which could otherwise introduce noise into the model.
3. The same filtering process was applied to the test dataset to maintain consistency across both training and testing phases.

Feature Scaling:

- After filtering, the remaining numerical columns were normalized using the **MinMaxScaler**.
- **MinMax Scaling** transforms numerical features to a specified range, typically between 0 and 1.
- Scaling ensures that features with larger numeric ranges do not dominate smaller-scale features, ensuring equal contribution to model training.

Cycle Normalization and Label Creation:

- A new normalized column, `cycle_norm`, was created to normalize the equipment's operational cycle across data points.

- A **Remaining Useful Life (RUL)** column was computed by subtracting the current cycle from the equipment's maximum cycle.
- Two binary labels were introduced to facilitate classification-based modeling:
    - **Label 1:** Set to 1 if RUL ≤ 30, otherwise 0.
    - **Label 2:** Set to 1 if RUL ≤ 15, otherwise 0.
- These labels allow the model to classify whether the equipment is approaching a critical point of failure within predefined thresholds (30 and 15 cycles).

Test Data Alignment:

- Similar preprocessing steps were applied to the test dataset, ensuring consistency with the training data. The RUL was recalculated for the test data to match the training framework.
- The truth dataset, containing actual RUL values, was merged with the test dataset to create consistent labels for evaluation.

Correlation Analysis:

- A **correlation matrix** was generated to identify highly correlated features. This step helps determine if additional columns should be excluded to avoid multicollinearity, which can negatively affect model performance.

LSTM Model: Architecture and Hyperparameters

1. Model Architecture

The model is doing a binary prediction on label1 predicting whether the equipment will fail in 30 cycles or not.

The LSTM model was built using a **sequential neural network** structure. Below are the key layers:

- **Layer 1:**
    - LSTM with 16 units
    - Input shape: `(sequence length, number of features)`
    - `return_sequences=True` allows this layer to return the full sequence, making it compatible with subsequent LSTM layers.
    - **Dropout:** A 20% dropout to prevent overfitting by randomly disabling neurons during training.
- **Layer 2:**
    - LSTM with 8 units
    - `return_sequences=False` indicates this is the final LSTM layer, returning only the last output in the sequence.

- **Layer 3:**
    - **Dense Layer** with a sigmoid activation function.
    - Output units match the number of target labels (`nb_out`), which are likely binary labels (label1 and label2).

The **sigmoid activation** function ensures output values between 0 and 1, appropriate for binary classification.

---

2. Model Compilation

- **Loss Function:** Binary Crossentropy
  Used for binary classification, where the model learns by minimizing the difference between predicted probabilities and actual labels.
- **Optimizer:** Adam
  A widely used optimization algorithm for neural networks, especially effective for sparse data and adaptive learning rates.
- **Metrics:** Accuracy
  This metric helps track how often the model correctly predicts the labels during training and validation.

---

3. Training Parameters and Hyperparameters

- **Epochs:** 30
  This determines the number of complete passes through the dataset during training.
- **Batch Size:** 64
  The number of samples processed before the model updates its weights.
- **Validation Split:** 0.2
  20% of the training data is used for validation to monitor the model's performance during training.
- **Early Stopping:**
    - **Monitor:** Validation Loss
    - **Patience:** 5 epochs (training stops if the validation loss does not improve for 5 consecutive epochs).
    - **Min Delta:** 0.001 (a small improvement in loss is required to count as progress).

Evaluation and Results Interpretation

After training, the model is evaluated on the test dataset using the following metrics:

- **Accuracy:** The proportion of correct predictions.
- **Precision:** The ratio of correctly predicted positive observations to the total predicted positives.
- **Recall:** The ratio of correctly predicted positives to the actual positive observations.
- **F1-Score:** The harmonic mean of precision and recall, balancing false positives and false negatives.

  Accuracy: 94%

  Precision: 0.95

  Recall: 0.80

  F1-Score: 0.87

  Confusion Matrix:

  A confusion matrix was printed to provide a detailed view of true vs. predicted labels.

Interpretation of Results

1. **High Accuracy (94%)**:
   The model performs well in correctly classifying the equipment's status across the dataset.
2. **Precision (0.95)**:
   This suggests the model is highly effective in minimizing false positives, meaning it rarely predicts equipment failure when the equipment is functioning correctly.
3. **Recall (0.80)**:
   The recall indicates that the model correctly identifies 80% of the critical cases where equipment is about to fail, but some failures are still missed.
4. **F1-Score (0.87)**:
   A balanced measure of the model's precision and recall, indicating that it handles both metrics well, but some improvements could be made.

Insights from Results

1. **Overfitting Controlled with Dropout and Early Stopping:**
   The use of dropout layers and early stopping likely contributed to preventing overfitting, ensuring the model generalizes well to unseen data.
2. **Balanced Model Performance:**
   While the model shows high precision, the slightly lower recall suggests it might be conservative in predicting failures, prioritizing precision. This behavior is beneficial in scenarios where false alarms are costly but missing critical failures can still pose risks.

3. **Further Improvements:**
    - **Hyperparameter Tuning:** Increasing the LSTM units or adjusting the batch size might further improve recall.
    - **Feature Engineering:** Exploring more interactions between sensor data could reveal hidden trends that improve the model's sensitivity to failures.

Conclusion

The LSTM model shows strong performance with an accuracy of 94% and an F1-score of 0.87, making it a viable solution for predicting the Remaining Useful Life (RUL) of equipment. It demonstrates a good balance between precision and recall, though slightly improving recall could enhance failure detection. This analysis lays a solid foundation for predictive maintenance, with opportunities for future work through hyperparameter tuning and feature engineering.