Equipment Failure Prediction for Predictive Maintenance

Nancy
IGDTUW

Under Mentorship of
Dhairya

IITD-AIA FOUNDATION FOR SMART MANUFACTURING

Project Background

Problem statement:

Equipment failure prediction is a technique that aims to prevent or reduce the occurrence of equipment failures by using data analysis and machine learning methods. Equipment failure prediction can be applied to various domains such as telecommunication, aviation, manufacturing, transportation, and healthcare, etc. The NASA turbofan jet engine data set is a synthetic data set that simulates the run-to-failure degradation of a fleet of turbofan engines. The problem definition is to predict the remaining useful life (RUL) of the engines based on their engine settings and sensor readings.

Unique challenge:

- Data availability and quality: It is essential to collect data for training machine learning algorithms. This information must be readily accessible in sufficient numbers for the algorithms to learn properly. The data also must be accurate, consistent, and relevant to the equipment failure modes.
- Complexity of machine learning algorithms: Algorithms for machine learning can be complicated and challenging to implement, necessitating specific knowledge and skills. The algorithms also must be robust, scalable, and adaptable to different types of equipment and failure scenarios.
- Cost and time of implementation: Predictive maintenance requires investment in sensors, data storage, computing power, and software development. It also takes time to collect enough data, train the algorithms, validate the results, and deploy the solutions

Objective

The Objective of the project was to implement machine learning or deep learning techniques to Predict Equipment Failure for predictive maintenance. Due to the complexity of data, noisy data and excessive data it become difficult to accurately predict the RUL(remaining useful life) of the engine and do the maintenance timely to reduce the risk of failure.

Methodology

1. Dataset Fetching
2. Data Preprocessing: Data cleaning and transformation, handling missing and null values
3. EDA: Exploratory Data Analysis
4. Feature Selection
5. Created class label: Tried to simplify the problem, by converting it to a Classification problem, where the class labels will be of 3 types, i.e., Good Condition represented by 0, Moderate Condition represented by 1 and Warning Condition represented by 2. To defining the class labels, I define the engine's condition with Life_Ratio (LR), which is the ratio between Current Cycle and the End cycle (end_of_life). If LR=0, that means the component has just started its degradation and LR=1 means, it is completely degraded. If LR <= 0.6 - Good Condition (0), if LR <= 0.8 - Moderate Condition (1) and if 0.8 < LR - Warning Condition (2).
6. Balancing the dataset: using SMOTE, SMOTE-tomek techniques
7. Model Implementation: Logistic Regression, Random Forest, Gradient Boosting, KNN, Decision Tree, Extreme Gradient Boosting, Light GBM, Gaussian Naive Bayes, ANN, LSTM models are implemented
8. Model Evaluation: using Accuracy, precision, recall and f-1 score, Mean Squared Error and R2 score.

Implementation

The Logistic Regression, Random Forest, Gradient Boosting, KNN, Decision Tree, Extreme Gradient Boosting, Light GBM, Gaussian Naive Bayes, ANN, LSTM are implemented. First the model are trained without balancing the dataset. Then, Later on the datasets are balanced and again trained on the mentioned algorithms.

Before balancing Random Forest, Decision Tree, XGB gives 100% training accuracy which implies that the models are overfitting.

Implementation

Result of implemented algorithms before balancing

Dataset→	FD001	FD002	FD003	FD004
Model	Test Acc	Test Acc	Test Acc	Test Accu
Logistic Regression	89.70%	99.1 %	88.95%	87.72%
SVM	90.06%	90.6%	89.27%	87.46%
Random Forest	90.74%	99.4%	90.89%	89.33%
KNN	86.14%	98.56%	87.86%	88.67%
Decision Tree	86.14%	99.59%	86%	85.78%
AdaBoost	75.18%	95.10%	81.37%	82.88%
XGB	89.82%	99.70%	90.19%	89.22%
Light GBM	90.28%	99.70%	90.57%	89.3%
Gaussian NB	83.42%	95.56%	83.11%	84.86%

Implementation

Result of implemented algorithms after balancing

Model/Dataset→	FD001	FD002	FD003	FD004
Logistic Regression	83.62%	78.11%	70.11%	76.78%
Random Forest	93.89%	92.75%	93.75%	92.53%
KNN	92.18%	89.87%	91.87%	90.39%
Decision Tree	90.52%	89.9%	90.32%	90%
AdaBoost	83.72%	69.67%	70.67%	82.87%
XGB	92.07%	90.5%	92.73%	90%
Light GBM	91.48%	91.03%	92.23%	89.38%
Gaussian NB	80.64%	83.57%	80.57%	82.83%

Innovation in Implementation

Firstly, I am beginner in this domain and do not had experience to deal with real world dataset, so I tried to simplify the problem by converting it to classification problem, where the class labels will be of 3 types, i.e., Good Condition represented by 0, Moderate Condition represented by 1 and Warning Condition represented by 2. Warning condition means that the maintenance is required and must be done to reduce the risk of failure of engine. If we get 0 or 1 then it means the engine is working fine and no need of maintenance.

- If LR <= 0.6 - Good Condition (0)
- if LR <= 0.8 - Moderate Condition (1)
- if 0.8 < LR - Warning Condition (2)

Then I found that dataset is imbalance, i.e., the one type of class label are more than other class label. If the imbalance datasets is not handled than the implemented models will be biased toward majority class.

Outcome

The Random Forest, Extreme Gradient Boosting and Light GBM shows accuracy, precision value, recall value and f-1 score of more than 90%. These models can be used to predict the equipment failure.

Scalability

By implementing ML models to predict equipment failure for predictive maintenance, one technique is to use historical, available data to foresee when equipment failure is likely to occur and proactively address it with maintenance. We can use a classification or regression techniques to predict the possibility of failure in the next steps or the time left before a system fails based on various types of data such as vibration, temperature, etc. To solve industrial problems using equipment failure prediction, one approach is to use machine learning techniques such as deep neural networks, support vector machines, random forests, etc. that can perform fault detection and prognosis based on various types of data such as vibration, acoustic emission, temperature, etc.

Thank You

nancy045btcse20@igdtuw.ac.in