Automated Anomaly Detection for Predictive Maintenance

Debangana Chatterjee
Data Science Bootcamp, upGrad

Contents

Introduction: Many different industries need predictive maintenance solutions to reduce risks and gain actionable insights through processing data from their equipment.

Although system failure is a very general issue that can occur in any machine, predicting the failure and taking steps to prevent such failure is most important for any machine or software application.

Predictive maintenance evaluates the condition of equipment by performing online monitoring. The goal is to perform maintenance before the equipment degrades or breaks down.

This Capstone project is aimed at predicting the machine breakdown by identifying the anomalies in the data.

Dataset Overview: This project focuses on developing an automated anomaly detection system for predictive maintenance using time-series data. The dataset consists of over 18,000 rows collected over several days, with the binary column 'y' indicating anomalies (1 for anomaly, 0 for normal operation). The rest of the columns are predictors. The primary goal is to build and evaluate a machine learning pipeline capable of detecting anomalies effectively.

Design Choices:

1) Data Preprocessing:

- Handling Missing values
- Checking data types of all features and target class
- Using descriptive statistics we saw that the mean of target class 'y' is 0.0067 i.e. the target class was highly imbalanced and we visualized the class imbalance.
- We handled the target class imbalance with the help of removing outliers and oversampling techniques.
- We have also standardized the features using StandardScalar to improve model performance.

2) Feature Engineering:

- Visualizing Features Correlation plot for better understanding of the relationship among the features.
- Derived metrics like sum, mean, standard deviation, and min, max to capture trends in data.
- Transforming datatype of 'time' feature, added new features 'hour', 'month', 'date' with corrected datatype.
- Used XGBoost's and Random Forest Classifier's built-in feature importance to select the most influential predictors.

3) Model Selection and Training:

- Different classification models were considered for this project including:

  - Logistic Regression
  - Random Forest Classifier
  - Support Vector Machine
  - Gradient Boosting(XGBoost)

The choice of model was based on their performance on the validation dataset, ability to handle class imbalance and ability of reducing overfitting.

- The dataset was split into 3 sets training, validation and testing sets. The training set was used to fit the model, tuning hyperparameters as necessary to improve model performance.

4) Model Validation:

Model Performance was evaluated using various metrics such as:

- Accuracy
- Precision
- Recall
- F1 Score
- ROC-AUC Score
- Confusion Matrix

The evaluation helped to ensure that the model generalizes well to unseen data.

Performance Evaluation:

Upon evaluating various model's performance, it was found that the accuracy score exceeded 75%, meeting the project's success criteria. Key performance metrics demonstrated that all the models were capable of detecting anomalies effectively, balancing both precision and recall.

As all the models showed high accuracy on scaled validation set, we used the models to predict the scaled test data; the AUC scores obtained for all models were 0.99 or 1.00

Random Forest's and XGBoost's built-in attribute Feature importance analysis indicated that certain features significantly influenced the model's predictions.

Future Work:

- Incorporating deep learning techniques like LSTMs for time-series forecasting.
- Including external factors such as weather data or operational logs could improve prediction accuracy.
- Developing a system for continuous monitoring where the model adapts to new data over time.

Source Code:

The source code used in this project includes the following steps:

- Data Loading and Cleaning.
- Exploratory Data Analysis (EDA).
- Feature Engineering.
- Model Training and Evaluation.
- Saving the Model for Deployment.

Conclusion:

This project successfully implemented an anomaly detection pipeline for predictive maintenance. The high performance metrics validate its effectiveness, and proposed enhancements offer a roadmap for further optimization and deployment.