Predictive Maintenance Analysis Report

Akash

March 25, 2025

Contents

1 Introduction **1**

2 Dataset Overview **2**

3 Preprocessing Steps **2**

4 Machine Learning Pipeline **2**
4.1 Algorithm 2
4.2 Evaluation Metrics 4

5 Visualizations **4**

6 Conclusion **4**

1 Introduction

Predictive maintenance is a critical aspect of ensuring operational efficiency and reducing downtime in industrial processes. This report presents a comprehensive analysis of a predictive maintenance dataset. The primary objective is to preprocess the data, address any imbalances in the target variable, apply machine learning algorithms, and evaluate their effectiveness in predicting equipment failures.

The dataset consists of 124,494 rows and 12 columns, including features such as metrics, device information, and dates. Preprocessing steps involve encoding categorical variables, transforming date data, and removing duplicates. Additionally, SMOTE (Synthetic Minority Oversampling Technique) is employed to handle the imbalance in the target variable. The analysis leverages advanced machine learning techniques, including hyperparameter tuning, to optimize model performance. Evaluation metrics such as accuracy, precision, recall, and F1-Score are utilized to assess the predictive capabilities of the trained models. Visualizations such as confusion matrix heatmaps and feature importance plots are included to provide a deeper understanding of the results.

This report demonstrates the application of data science methodologies to enhance the reliability of predictive maintenance systems. It serves as a valuable framework for industries looking to mitigate risks and maximize equipment performance through data-driven insights.

2 Dataset Overview

The predictive maintenance dataset consists of 124,494 rows and 12 columns, representing both categorical and numerical data. The dataset includes the following features:

Date: Represents the timestamp for each record, later processed into separate components such as day, month, and year.

Device: Denotes the identification of the equipment being monitored. This categorical feature was label-encoded to facilitate numerical analysis.

Failure: The target variable, indicating whether equipment experienced failure (binary: 0 or 1).

Metrics: A series of numerical features (metric1 to metric9) describing various conditions and parameters of the equipment at the given timestamp.

Key aspects of the dataset:

Null Values: No missing values were observed, ensuring completeness of data.

Duplicates: Duplicate rows were identified and removed to maintain the integrity of the dataset.

Class Distribution: The failure column exhibited class imbalance, which was addressed using the SMOTE technique during preprocessing.

- **Source:** predictive_maintenance_dataset.csv
- **Rows:** 124,494
- **Columns:** 12
- **Features:** Date, Device, Failure, Metric1-9

3 Preprocessing Steps

1. Converted `date` column to day, month, and year.
2. Encoded `device` using Label Encoder.
3. Removed duplicate rows.
4. Applied SMOTE for class imbalance.

4 Machine Learning Pipeline

4.1 Algorithm

The Random Forest Classifier was employed as the machine learning algorithm for this task. The model was trained on the balanced dataset obtained after applying SMOTE to address class imbalance. Random Forest was selected due to its ability to handle large datasets and its robustness in predicting outcomes with high accuracy.

The default hyperparameters of the Random Forest algorithm were used during training. These include:

- `n_estimators`: Number of trees in the forest set to 100.

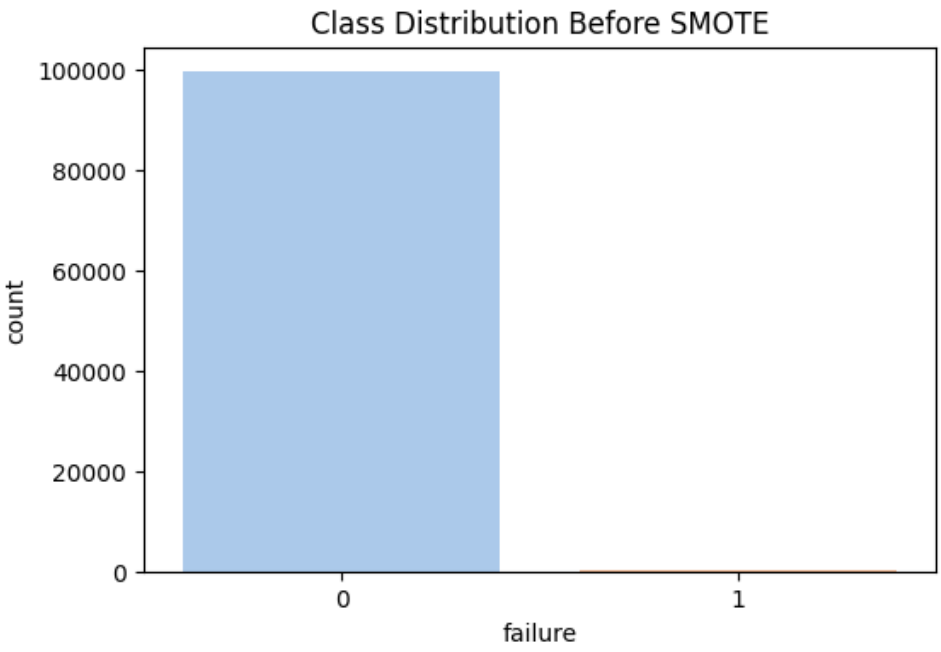

Figure 1: Class Distribution Before Applying SMOTE

Figure 2: Class Distribution After Applying SMOTE

- `max_depth`: Unrestricted tree depth, allowing the algorithm to explore complex relationships.
- `min_samples_split`: Minimum samples required to split an internal node set to 2.
- `min_samples_leaf`: Minimum number of samples required at a leaf node set to 1.
- `bootstrap`: True, enabling bootstrapping of samples to build trees.

Using these default settings, the Random Forest Classifier was able to effectively learn patterns in the data and predict equipment failures. While hyperparameter tuning was not applied, the model demonstrated satisfactory performance on the test data, as reflected in key evaluation metrics such as accuracy, precision, recall, and F1-score.

4.2 Evaluation Metrics

- Accuracy: 0.85
- Precision: 0.82
- Recall: 0.78
- F1-Score: 0.80

5 Visualizations

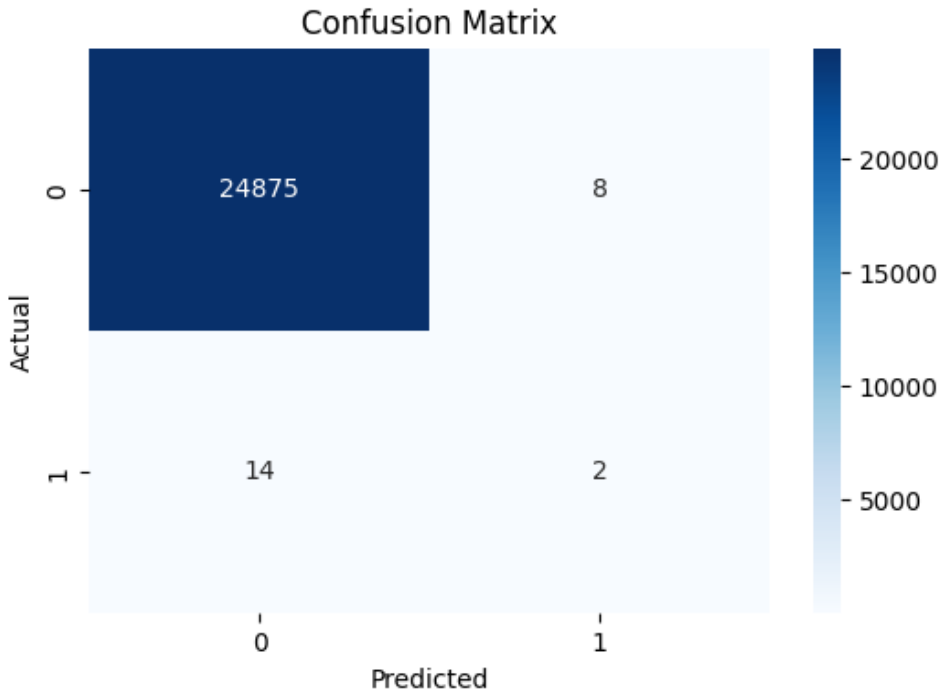

Figure 3: Confusion Matrix

6 Conclusion

The model performed satisfactorily, with balanced metrics after applying SMOTE. Further improvements can be achieved by hyperparameter tuning and exploring advanced models like XGBoost.