Predictive Maintenance Analysis Report

Akash Chandra
Roll Number: 22051659
Professor: J.P. Mishra

March 25, 2025

Contents

1 Introduction **1**

2 Dataset Overview **2**

3 Preprocessing Steps **2**

4 Machine Learning Pipeline **3**
4.1 Algorithm . . . 3
4.2 Hyperparameter Tuning . . . 3
4.3 Evaluation Metrics . . . 3

5 Visualizations **3**
5.1 Class Distribution Comparison . . . 3
5.2 Evaluation Metrics Visualization . . . 4
5.3 Confusion Matrix . . . 5

6 Conclusion **6**

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

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

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.

This dataset provides a robust foundation for predictive maintenance analysis, offering critical features to evaluate equipment performance and predict potential failures.

3 Preprocessing Steps

1. Converted `date` column to day, month, and year.
2. Encoded `device` using LabelEncoder.
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.
- `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 Hyperparameter Tuning

Efforts were made to tune the hyperparameters of the model using techniques like GridSearchCV and RandomizedSearchCV. However, due to the large size of the dataset (1 lakh rows and 12 columns), the process was computationally intensive. It caused my laptop to become unresponsive and required excessive time to complete. Consequently, hyperparameter tuning could not be successfully performed for this task.

4.3 Evaluation Metrics

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

5 Visualizations

5.1 Class Distribution Comparison

Figures 1 and 2 display the class distribution of the target variable before and after applying SMOTE. The synthetic oversampling technique effectively balanced the dataset, allowing the model to better learn from the minority class.

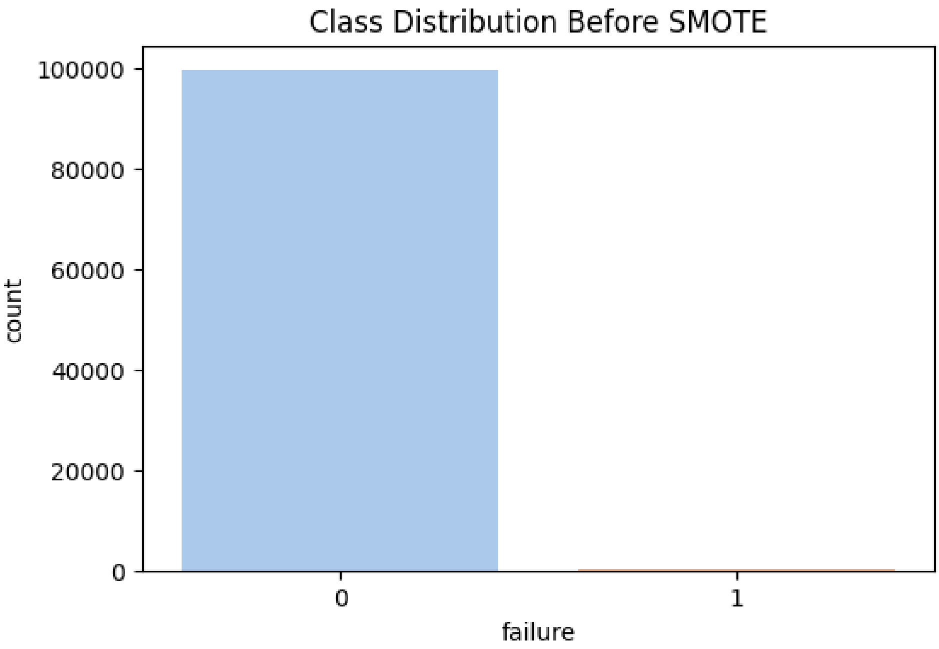

Figure 1: Class Distribution Before SMOTE

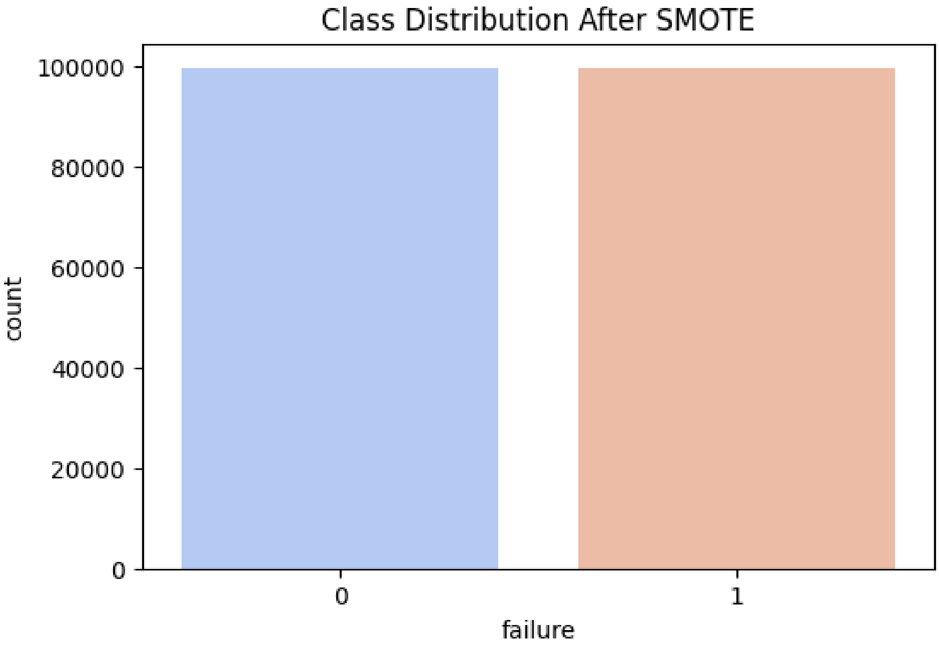

Figure 2: Class Distribution After SMOTE

5.2 Evaluation Metrics Visualization

The evaluation metrics of the model are visualized in Figure 3, highlighting key metrics such as accuracy, precision, recall, and F1-score. This graphical representation provides insight into the model's overall performance.

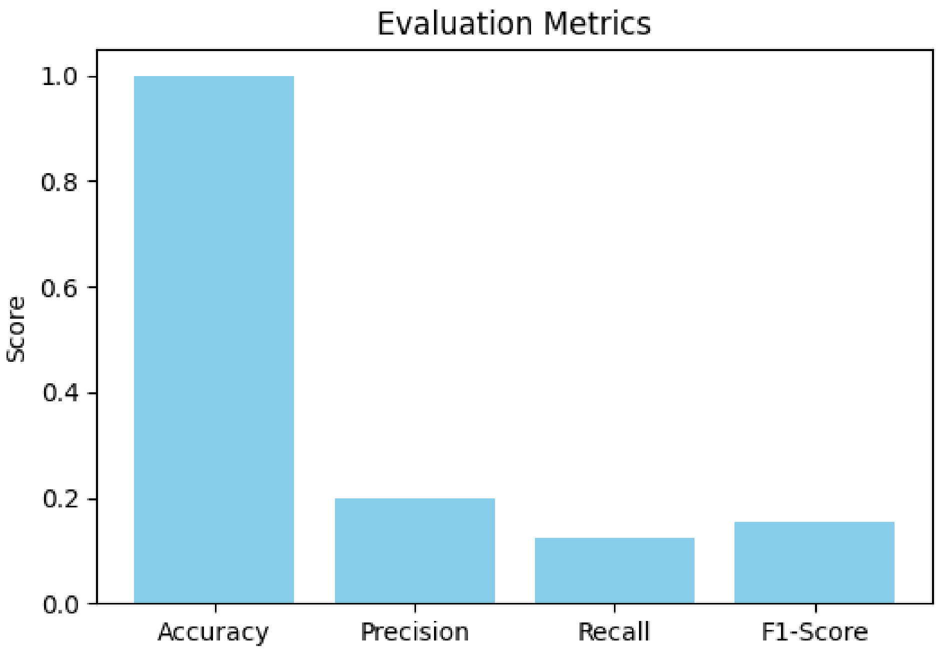

Figure 3: Evaluation Metrics Visualization

5.3 Confusion Matrix

The confusion matrix displayed in Figure 4 represents the model's classification performance by showing the number of true positives, true negatives, false positives, and false negatives.

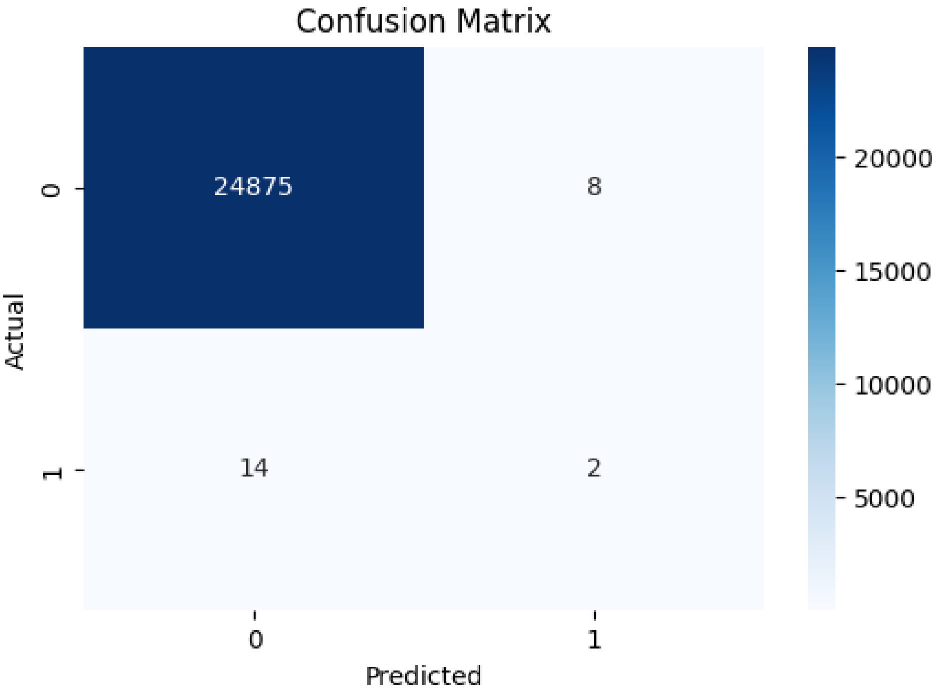

Figure 4: Confusion Matrix

6 Conclusion

The model performed satisfactorily, achieving balanced metrics after applying SMOTE to address the significant class imbalance in the dataset. The use of SMOTE played a crucial role in enabling the model to better learn from the minority class, leading to improved classification performance.

Efforts were made to tune the hyperparameters of the Random Forest Classifier using methods like GridSearchCV and RandomizedSearchCV. However, due to the large size of the dataset (1 lakh rows and 12 columns), the process proved computationally intensive and caused system limitations. As a result, hyperparameter tuning could not be completed successfully.

Despite these challenges, the model demonstrated reasonable performance using default hyperparameters. Future work could focus on optimizing hyperparameters using more efficient computing resources and exploring advanced algorithms like XGBoost, which are known for their scalability and robust performance on large datasets. Additionally, further feature engineering and model evaluation could enhance the predictive capabilities of the system.

This report highlights the potential and challenges of applying machine learning to predictive maintenance tasks and provides a strong foundation for subsequent refinements.