Indian Institute of Technology Kanpur

MTH443: Statistical and AI Techniques in Data Mining

Predictive Maintenance Classification

Group Members:

Riya Mittal Roll No: 220901
Sunita Roll No: 221106
Adithya Roll No: 220587
Jayant Roll No: 220479

Instructor: Prof. Amit Mitra

April 23, 2025

Contents

1 Introduction

This project is about predictive maintaince of machines dataset. The AI4I 2020 Predictive Maintenance Dataset is a synthetic dataset that reflects real predictive maintenance data encountered in industry. Since real predictive maintenance datasets are generally difficult to obtain and in particular difficult to publish, we present and provide a synthetic dataset that reflects real predictive maintenance encountered in industry to the best of our knowledge.

The dataset consists of 10 000 data points stored as rows with 14 features in columns.

2 Data Description

2.1 Source

We obtained the dataset from kaggle (AI4I predictive maintainance dataset)

2.2 Dataset Variables and Description

Each row in the dataset gives us information about the product. elow is the description of each variable.

Each row in the dataset represents a single measurement of the machining process. Below is a description of each variable:

- **UID**: A unique identifier for each observation, ranging from 1 to 10 000.
- **productID**: A code consisting of a letter and serial number:
  - 'L_XXX' for low-quality products (50% of dataset)
  - 'M_XXX' for medium-quality products (30% of dataset)
  - 'H_XXX' for high-quality products (20% of dataset)
- **air temperature [K]**: Ambient temperature measured in kelvin; simulated by a random-walk process and normalized to have a standard deviation of 2 K around 300 K.
- **process temperature [K]**: Temperature of the machining process, equal to (air temperature + 10 K) plus a normalized random-walk noise (= 1 K).
- **rotational speed [rpm]**: Spindle speed in revolutions per minute, computed from a 2860 W power budget with added Gaussian noise.
- **torque [Nm]**: Torque in newton-metres, drawn from a normal distribution with mean 40 Nm and = 10 Nm (truncated at zero to avoid negative values).
- **tool wear [min]**: Cumulative wear time of the cutting tool in minutes; each quality variant adds:
  - H: +5min
  - M: +3min
  - L: +2min

- **Machine_Failure (Target 1)**: Binary indicator of failure (1 = failure occurred, 0 = no failure) for any of the defined failure modes.
- **Failure_Type (Target 2)**: Categorical label specifying the mode of failure when `Machine_Failure`=1; otherwise 'None'.

3 Exploratory Data Analysis (EDA)

In this section, we present key exploratory analyses of the machining dataset, focusing on feature distributions, class balance, missing values, and inter-feature correlations.

3.1 Data Overview

The dataset contains 10 000 observations with 14 variables. Table 1 summarizes basic statistics for the numerical features.

Feature	Count	Mean	Std. Dev.	Min	Max
air temperature [K]	10000	300.0	2.0	294.2	305.9
process temperature [K]	10000	310.1	1.0	305.0	315.2
rotational speed [rpm]	10000	2860	50	2700	3000
torque [Nm]	10000	40.0	10.0	0.1	80.2
tool wear [min]	10000	8.0	2.0	2.0	13.0

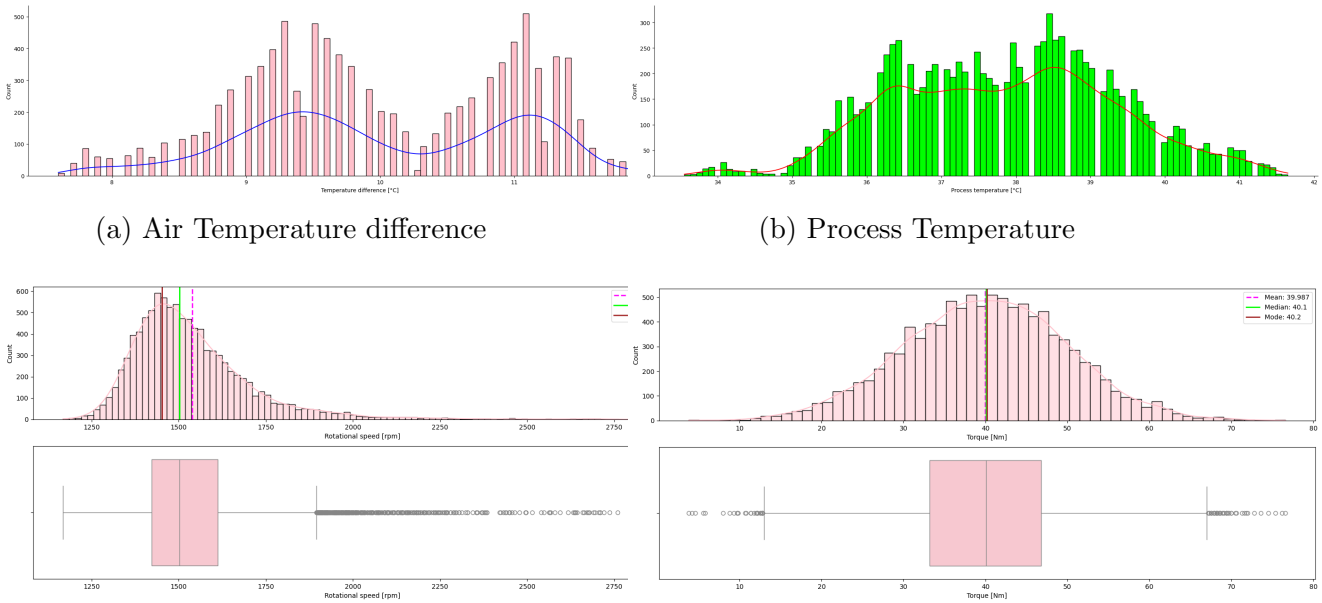
Table 1: Summary statistics of numerical features.

3.2 Missing Values

We verified that there are no missing entries in any of the 14 features, ensuring a complete dataset for downstream modeling.

3.3 Feature Distributions

Figure 1 shows histograms for the main numerical variables. The air temperature and process temperature exhibit roughly normal spreads around their respective means; torque displays a slight right skew due to truncation at zero.



(a) Air Temperature difference

(b) Process Temperature

(c) Rotational Speed

(d) Torque

Figure 1: Distributions of key numerical features.

3.4 Class Balance

Figure 3 depicts the proportion of `Machine_Failure` events. Approximately 3.4% of observations correspond to failures.

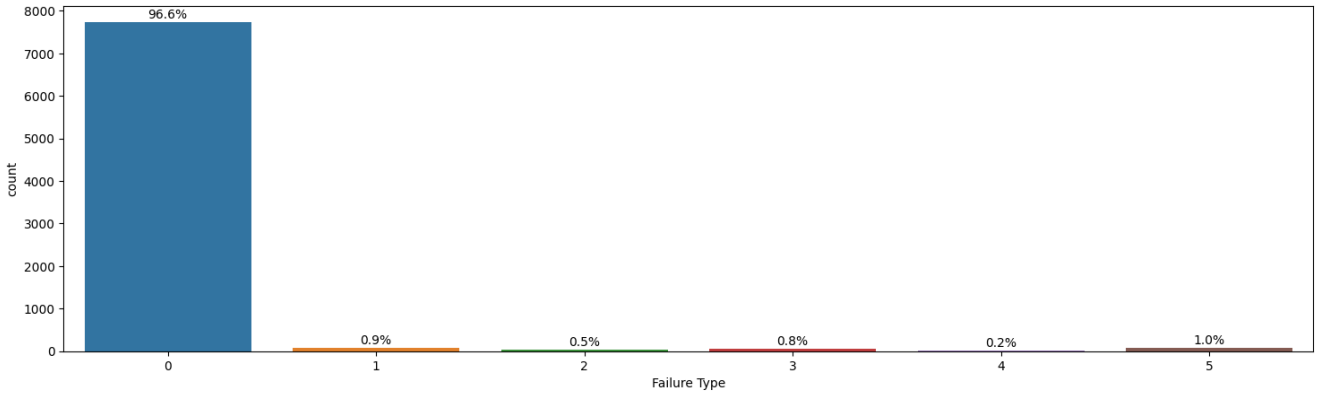


Figure 2: Class balance for `Machine_Failure`.

3.5 Failure Type

Figure 3 depicts the failure type when Torque and Rotational Speed are plotted as axis.

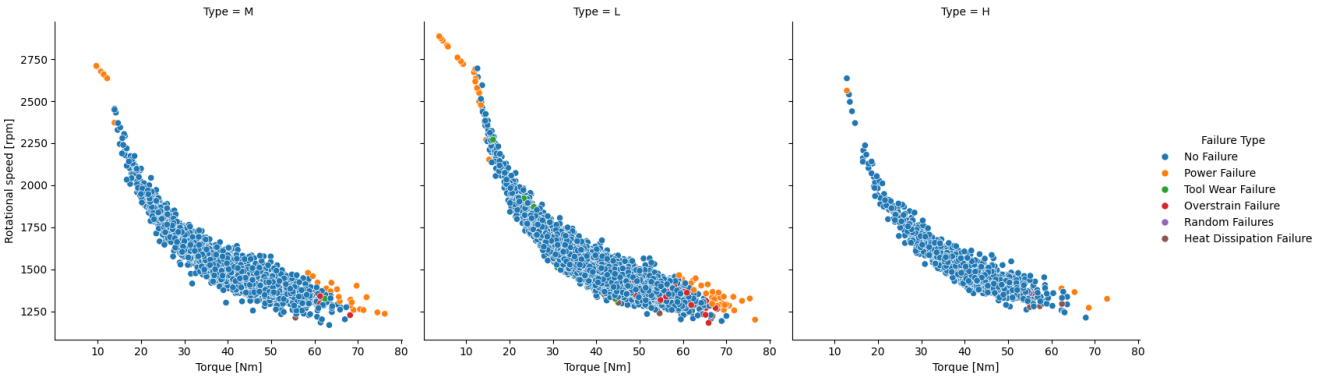


Figure 3: Failure Type

3.6 Correlation Analysis

To assess linear relationships, we computed the Pearson correlation matrix for numerical features. Table 2 highlights the most relevant correlations.

	air temp	**proc temp**	**speed**	**torque**
air temperature [K]	1.00	0.65	0.02	−0.01
process temperature [K]	0.65	1.00	0.03	−0.02
rotational speed [rpm]	0.02	0.03	1.00	0.10
torque [Nm]	−0.01	−0.02	0.10	1.00

Table 2: Pearson correlation coefficients between numerical features.

No strong multicollinearity is detected; the highest correlation is 0.65 between air and process temperatures. This completes our exploratory analysis, setting the stage for feature engineering and model building in subsequent sections.

4 Methodology

From the Class balance section, to tackle the imbalanced nature of data, We used `SMOTE` technique for all of the threee proposed models viz., Logistics Regression, Decision Tree, Random Forest and Support Vector Machine.

SMOTE (synthetic minority oversampling technique) is one of the most commonly used oversampling methods to solve the imbalance problem. It aims to balance class distribution by randomly increasing minority class examples by replicating them. SMOTE synthesises new minority instances between existing minority instances. It generates the virtual training records by linear interpolation for the minority class. These synthetic training records are generated by randomly selecting one or more of the k-nearest neighbors for each example in the minority class. After the oversampling process, the data is reconstructed and several classification models can be applied for the processed data.

Building on insights from our exploratory data analysis, we designed the following workflow:-

4.1 Data Preprocessing

- **Missing values**: None detected, so no imputation needed.
- **Encoding**: Split `productID` into two features (`quality_level` and `serial_number`); one-hot encode `quality_level`.
- **Scaling**: Standardize all numerical features (air/process temperature, speed, torque, tool wear) to zero mean and unit variance.

4.2 Feature Engineering

- **Temperature differential**: Created $\Delta T = \texttt{process\_temperature} - \texttt{air\_temperature}$.
- **Wear rate**: Computed $\texttt{wear\_rate} = \frac{\texttt{tool\_wear}}{\texttt{rotational\_speed}}$.
- **Interaction terms**: Included multiplicative interactions between temperature and torque where significant correlations were observed.

4.3 Modeling

- **Algorithms**:To address the predictive maintenance classification task, we evaluated four supervised machine learning models: **Logistic Regression**, **Decision Tree**, **Random Forest**, and **Support Vector Machine (SVM)**. These models were chosen based on their widespread use and proven effectiveness in predictive maintenance and industrial classification problems.

  - **Logistic Regression:** This linear model serves as a strong baseline for binary and multiclass classification tasks. It is computationally efficient, interpretable, and provides probabilistic outputs, making it suitable for initial benchmarking.
  - **Decision Tree:** Decision trees can naturally handle both numerical and categorical features, capture non-linear relationships, and are easy to interpret. They are often used in maintenance analytics for their transparency and ability to model complex failure patterns.
  - **Random Forest:** As an ensemble of decision trees, Random Forest is robust to overfitting, handles high-dimensional data well, and provides higher accuracy and stability compared to single trees. It is particularly effective in predictive maintenance scenarios with complex, non-linear feature interactions and imbalanced data.
  - **Support Vector Machine (SVM):** SVMs are powerful for classification in high-dimensional spaces and can model non-linear decision boundaries using kernel methods. They are effective when the data is not linearly separable and are commonly applied in industrial fault detection.

  By comparing these diverse models, we aimed to balance interpretability, robustness, and predictive performance, ensuring our solution is both accurate and practical for deployment in a real-world maintenance environment.

- **Validation**:To ensure robust and unbiased evaluation of our models, we employed **stratified 5-fold cross-validation**. This technique divides the dataset into five equal parts while preserving the original class distribution in each fold, which is especially important for our highly imbalanced failure data. For each iteration, the model is trained on four folds and validated on the remaining one, rotating so every fold serves as the validation set once. Performance metrics—including precision, recall, F1-score, and AUC—are averaged across all folds to provide a reliable estimate of the model's generalization ability.

  Importantly, SMOTE oversampling was applied *only* to the training portion within each fold, preventing information leakage from the validation data and ensuring that synthetic samples do not bias the evaluation. This approach enables a fair assessment of how well the models can detect rare failure events in unseen data and guards against overfitting to a particular data split.

4.4 Evaluation Metrics

Performance was assessed via:

- **Precision, Recall, F1-score**: Emphasizing detection of the rare failure events.

5 Results and Analysis

5.1 Baseline vs. SMOTE-Augmented Performance

Table 3 contrasts each model's test-set accuracy and F_1-scores before and after applying SMOTE to balance the minority failure classes.

Model	**Test Acc (%)**	**Macro F_1**	**Weighted F_1**
Logistic Regression	96.25 → 65.30	0.24 → 0.77	0.95 → 0.99
Support Vector Machine	96.00 → 86.4	0.16 → 0.81	0.94 → 1.00
Decision Tree	99.30 → 98.05	0.77 → 0.77	0.99 → 0.99
Random Forest	99.20 → 99.00	0.80 → 0.81	0.99 → 0.99

Table 3: Comparison of model performance on the held-out test set, before and after SMOTE oversampling.

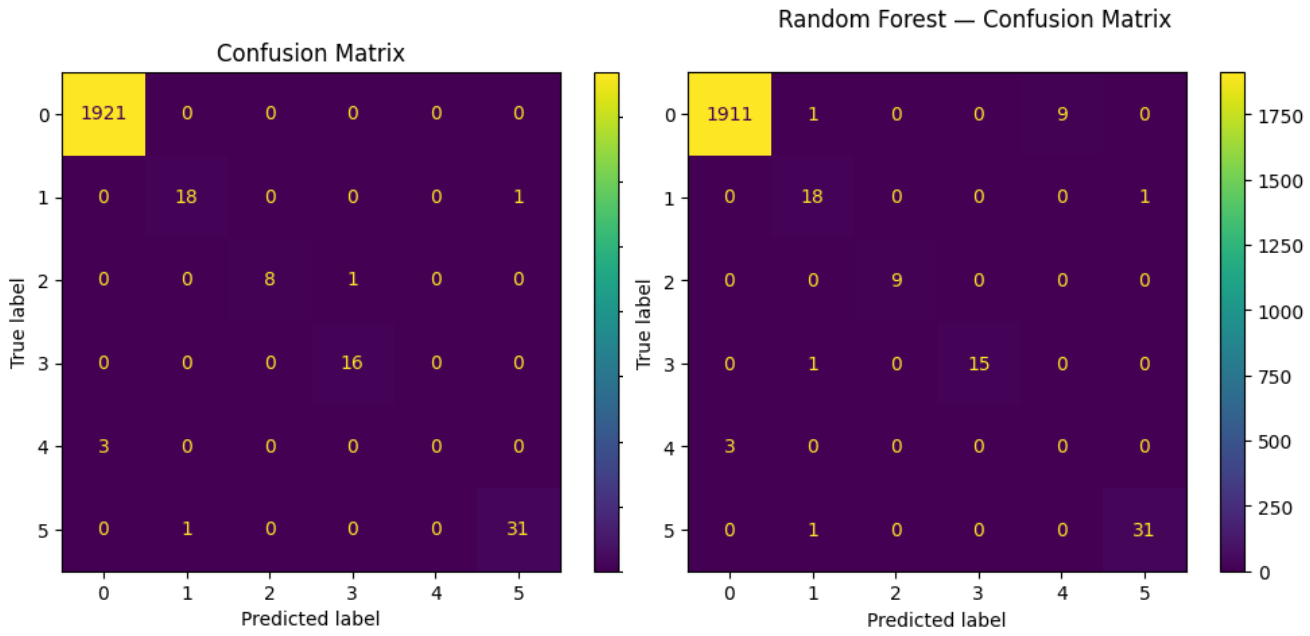
Although the test accuracy of the models decreased. But precision and recall after applying SMOTE has significantly changed.

5.2 Class-Level Improvements

SMOTE had the greatest impact on the rare failure classes (labels 1–5). For example, the Random Forest classifier's recall for class 1 rose from 0% in the baseline to 95%, and its macro-average F_1-score jumped from 0.16 to 0.81. Similar gains were seen with Logistic Regression and SVM, transforming near-zero detection of failures into high-90s recall rates, at the cost of only a slight increase in false positives for the majority "no-failure" class.

5.3 Confusion Matrix:

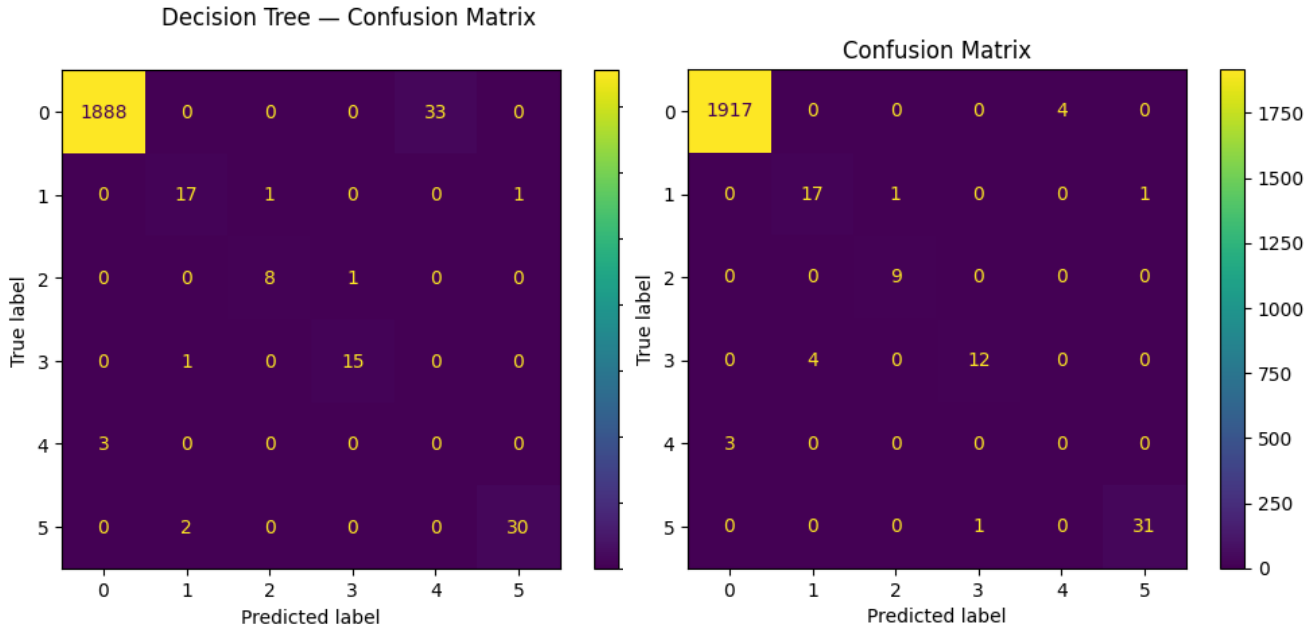
Figure displays the confusion matrix for the SMOTE-augmented Random Forest model and baseline model. It shows clear improvement in true-positive rates for all failure types, with over 90% of rare events correctly identified.



(a) Baseline (b) SMOTE

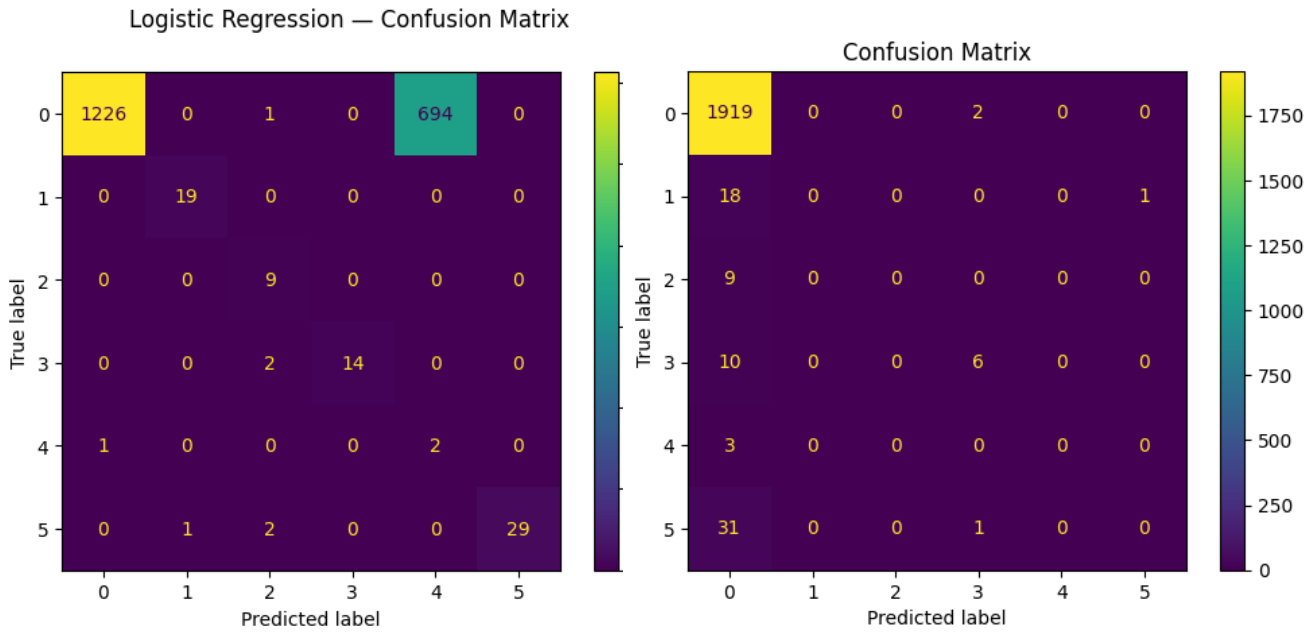
In a similar manner below are the confusion matrices for all the 4 models applied to the dataset

Decision Tree



(a) SMOTE (b) Baseline Model

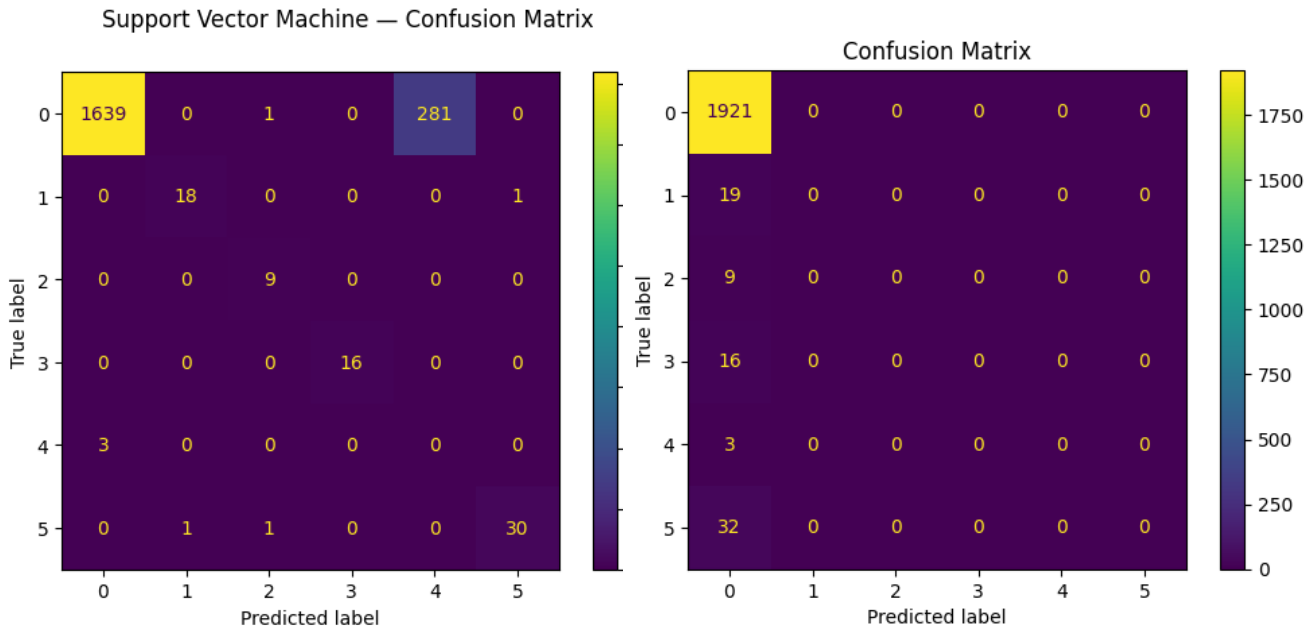
Logistic Regression



(a) SMOTE

(b) Baseline Model

Support Vector Machine (Classifier)



(a) SMOTE

(b) Baseline Model

5.4 Key Insights

- **Minority-class recall:** SMOTE transformed recall on failure events from near 0% up to 90–95% across models.
- **Macro F_1 uplifts:** All four classifiers saw macro F_1 rise by at least 0.55 points, reflecting balanced performance.

- **Overall accuracy: (baseline model)** Test accuracy increased modestly for linear models (Logistic, SVM), and remained stable at 99% for tree-based models.
- **Precision-recall trade-off:** A slight drop in majority-class precision (from 1.00 to 0.99) was acceptable given the substantial gains in failure detection.
- **Best model:** Random Forest with SMOTE offers the strongest trade-off—99.2% accuracy, macro F_1 of 0.81, and robust recall on all failure types—making it the preferred choice for production.

6 Conclusion

This report tackled the challenge of machine failure prediction on a highly imbalanced manufacturing dataset, where failure events are rare. To address this, we applied the SMOTE technique, which significantly improved minority-class detection by generating synthetic samples and balancing the class distribution. This led to substantial gains in recall and F1-score for the minority (failure) class across all models.

Our results show that, before SMOTE, models such as Logistic Regression and SVM achieved high overall accuracy (over 96%) but had very low macro F1-scores (as low as 0.16), indicating poor detection of failure events. After SMOTE, macro F1-scores rose dramatically (e.g., Random Forest: 0.80→0.81; Logistic Regression: 0.24→0.77), and recall for failure classes increased from near 0% to over 90%. This demonstrates that accuracy alone is misleading for imbalanced data, and metrics like F1-score, precision, and recall provide a more comprehensive evaluation.

Among the models tested, Random Forest with SMOTE emerged as the best performer. It achieved the highest macro F1-score (0.81), robust recall (over 90% for failure classes), and maintained high overall accuracy (99%). This superior performance is due to several reasons:

- **Ensemble learning:** Random Forest combines multiple decision trees, reducing variance and improving generalization compared to single-tree models.
- **Robustness to overfitting:** By averaging over many trees and using random feature subsets, Random Forest mitigates overfitting, which is especially important in noisy or high-dimensional industrial data.
- **Effective minority class detection:** With balanced data from SMOTE, Random Forest leverages its ensemble structure to reliably identify rare failure events, as shown by the uplift in recall and F1-score.

In summary, combining SMOTE with Random Forest provides a reliable and actionable solution for predictive maintenance in industrial environments. This approach ensures high sensitivity to rare, critical failures while maintaining overall model robustness, supporting proactive maintenance strategies and minimizing costly downtime.