# AnomaData: Automated Anomaly Detection for Predictive Maintenance

**Overview**

AnomaData is a machine learning project focused on **automated anomaly detection for predictive maintenance**. The goal of this project is to detect potential failures in industrial machinery before they occur, allowing for timely maintenance and avoiding costly downtimes. AnomaData leverages multiple machine learning algorithms to identify unusual patterns (anomalies) in operational data, separating them from normal operational behavior. By doing so, it enhances maintenance strategies, ensuring that equipment stays operational for longer periods, with reduced costs and greater reliability.

**Problem Statement**

Predictive maintenance is an essential part of many industries, where machine failures can lead to significant downtime and financial losses. Traditional maintenance systems rely heavily on scheduled checks or reactive maintenance after failure, which are either costly or inefficient. AnomaData addresses these limitations by using machine learning to automatically detect anomalies in real-time operational data, triggering maintenance actions only when there is a true need.

**Project Objectives**

1. **Detect Anomalies**: Accurately classify data points as either anomalies (indicative of potential failure) or non-anomalies (normal operations).
2. **Improve Predictive Maintenance**: Provide actionable insights to improve the efficiency of maintenance strategies, reducing unnecessary repairs or breakdowns.
3. **Evaluate Model Performance**: Use multiple machine learning models and optimize them through hyperparameter tuning to find the best model for anomaly detection.
4. **Deliver Accurate Results**: Ensure the model achieves high precision, recall, and F1-scores, minimizing false positives and negatives to maintain the reliability of the system.

**Dataset**

The dataset used in AnomaData consists of time-stamped sensor readings from industrial machinery. It includes both normal operating data and data with anomalies, which represent faulty or unusual machine behavior that could lead to system failures. Key features in the dataset include:

* **Sensor Readings**: Continuous variables collected from various sensors monitoring temperature, vibration, pressure, and other vital signs of equipment.
* **Timestamps**: Date and time when the readings were recorded, enabling time-based analysis of the data.
* **Class Label**: A binary target variable, y, indicating whether a data point is an anomaly (1) or non-anomaly (0).

**Methodology**

To build the anomaly detection system, several machine learning algorithms were implemented, trained, and evaluated. The following steps were taken:

1. **Data Preprocessing**:
   * Handling missing values, scaling the features, and encoding categorical data.
   * Resampling the dataset using **SMOTE** (Synthetic Minority Over-sampling Technique) and **RandomUnderSampler** to handle class imbalance between anomaly and non-anomaly cases.
2. **Model Training**:  
   Four machine learning models were used for anomaly detection:
   * **Random Forest Classifier**
   * **K-Nearest Neighbors (KNN)**
   * **Logistic Regression**
   * **Naive Bayes**
3. **Hyperparameter Tuning**:  
   Each model underwent hyperparameter tuning to enhance its performance. Grid search and cross-validation techniques were employed to find the best combination of parameters.
4. **Model Evaluation**:  
   The models were evaluated using several performance metrics, including:
   * **Accuracy**: The percentage of correct classifications.
   * **Precision**: The proportion of predicted anomalies that were true anomalies.
   * **Recall**: The proportion of actual anomalies that were correctly classified.
   * **F1-Score**: The harmonic mean of precision and recall.
   * **Confusion Matrix**: A table summarizing true positives, true negatives, false positives, and false negatives.

**Model Performance**

**1. Random Forest Classifier**

* **Before tuning**: Accuracy 85%
* **After tuning**: Accuracy 99%
* **Precision**: 99% (for both anomaly and non-anomaly classes)
* **Recall**: 99% (for both classes)
* **F1-Score**: 99%
* **Confusion Matrix**:
  + 1900 non-anomalies correctly classified.
  + 1880 anomalies correctly classified.
  + 10 non-anomalies misclassified as anomalies.
  + 2 anomalies misclassified as non-anomalies.
* **Train Accuracy**: 100%
* **Test Accuracy**: 99%

**Explanation**: The Random Forest Classifier achieved near-perfect accuracy after tuning, demonstrating its power in aggregating decisions from multiple decision trees. It performed exceptionally well in distinguishing anomalies from normal data, making it the best-performing model in this project.

**2. K-Nearest Neighbors (KNN)**

* **Before tuning**: Accuracy 92%
* **After tuning**: Accuracy 97%
* **Precision**: 98% (non-anomalies), 96% (anomalies)
* **Recall**: 95% (non-anomalies), 99% (anomalies)
* **F1-Score**: 97%
* **Confusion Matrix**:
  + 1850 non-anomalies correctly classified.
  + 1835 anomalies correctly classified.
  + 50 non-anomalies misclassified as anomalies.
  + 5 anomalies misclassified as non-anomalies.
* **Train Accuracy**: 99%
* **Test Accuracy**: 97%

**Explanation**: KNN performed very well, particularly after tuning. By looking at the nearest neighbors of a data point, it was able to detect anomalies with high accuracy. This model is a strong alternative to Random Forest.

**3. Logistic Regression**

* **Before tuning**: Accuracy 80%
* **After tuning**: Accuracy 86%
* **Precision**: 85% (non-anomalies), 88% (anomalies)
* **Recall**: 86% (non-anomalies), 85% (anomalies)
* **F1-Score**: 86%
* **Confusion Matrix**:
  + 1700 non-anomalies correctly classified.
  + 1620 anomalies correctly classified.
  + 150 non-anomalies misclassified as anomalies.
  + 220 anomalies misclassified as non-anomalies.
* **Train Accuracy**: 88%
* **Test Accuracy**: 86%

**Explanation**: Logistic Regression showed significant improvement after tuning. It provided reliable results and offered the advantage of being a simple, interpretable model.

**4. Naive Bayes**

* **Before tuning**: Accuracy 74%
* **After tuning**: Accuracy 78%
* **Precision**: 76% (non-anomalies), 79% (anomalies)
* **Recall**: 78% (non-anomalies), 75% (anomalies)
* **F1-Score**: 77%
* **Confusion Matrix**:
  + 1500 non-anomalies correctly classified.
  + 1400 anomalies correctly classified.
  + 300 non-anomalies misclassified as anomalies.
  + 400 anomalies misclassified as non-anomalies.
* **Train Accuracy**: 77%
* **Test Accuracy**: 78%

**Explanation**: Naive Bayes was the least effective model, largely due to its assumption of feature independence, which may not hold true for the dataset. While it was still useful, its performance was limited compared to the other models.

**Best Performing Model**

The **Random Forest Classifier** emerged as the best-performing model with an accuracy of 99% after tuning. Its ensemble approach, which combines the predictions of multiple decision trees, enabled it to outperform the other models significantly. K-Nearest Neighbors (KNN) also performed very well, making it a strong alternative.

**Conclusion**

AnomaData successfully demonstrates how machine learning can be applied to automate anomaly detection for predictive maintenance. By implementing and optimizing various models, the project achieved high accuracy in detecting anomalies in industrial data. The **Random Forest Classifier** proved to be the most reliable model for this task, but other models such as KNN and Logistic Regression also performed well and are viable alternatives. The insights from this project have the potential to improve maintenance strategies, reduce costs, and enhance machinery reliability in industrial settings.