**AnomaData: Automated Anomaly Detection for Predictive Maintenance**

**Objective**

In many industries, equipment failure can cause costly downtime and safety risks. Predictive maintenance helps prevent this by continuously monitoring equipment data to detect signs of potential failures early on. Our project, AnomaData, aims to predict machine breakdowns by identifying unusual patterns (anomalies) in the data. This can help industries proactively avoid breakdowns, saving time, and money, and improving safety.

1. Import Libraries

We start by loading the necessary tools (libraries) to work with data, build models, and create visualizations. This is like gathering the tools before starting a task.

2. Data Preparation

Before we build our model, we need to clean and organize the data. This ensures the model will be accurate and reliable.

* 2.1 Understand the Data: We first take a look at the data. It contains 18,000+ rows, with each row representing equipment data. We have a column y that tells us if an anomaly (issue) occurred (1 for yes, 0 for no).
* 2.2 Missing Value Treatment: Sometimes, parts of the data are missing. We fill or remove these gaps so the model can work properly.
* 2.3 Check Duplicate Rows: We check if there are any duplicate entries in the data that could skew the results.
* 2.4 Exploratory Data Analysis (EDA): We visualize the data to understand patterns and relationships. This gives us insights into how different factors might influence anomalies.
* 2.5 Data Cleaning and Feature Engineering: We clean the data and create new useful features (characteristics) that the model can use to better detect anomalies.
* 2.6 Outlier Detection: Outliers are data points that are far outside the usual range and can confuse the model. We identify these.
* 2.7 Outlier Treatment: We either remove or adjust outliers to make the data more consistent.
* 2.8 Balancing Dataset: If we have too many examples of one type (e.g., too many non-anomalies), the model might learn to ignore anomalies. We balance the data so the model can learn from both anomalies and normal data.
* 2.9 Scaling: We adjust the scale of the data to make sure that no single feature (like equipment temperature) dominates the model. This helps the model treat all features fairly.

3. Splitting Data into Train and Test

We split the data into two parts: one for training the model (where it learns patterns) and one for testing (to see how well the model performs on new, unseen data).

4. Random Forest Classifier Model

This model works like a collection of decision trees. It makes predictions by "voting" across many trees, which makes it very accurate and stable.

* 4.1 Evaluate the Model: We measure the model’s performance using metrics like accuracy, precision, and recall to see how well it detects anomalies.
* 4.2 Hyperparameter Tuning: We fine-tune the model's settings to improve its performance.
* 4.3 Performance After Tuning: After tuning, we test the model again, and its performance improves significantly.
* 4.4 Visualize Feature Importances: We create a chart to show which features are most important for detecting anomalies.

5. K Nearest Neighbors (KNN) Model

This model finds similar past examples to make predictions. If most similar examples show an anomaly, it will predict an anomaly.

* 5.1 Evaluate the Model: We check how well KNN predicts anomalies using similar metrics.
* 5.2 Hyperparameter Tuning: We adjust settings to optimize the model’s performance.
* 5.3 Performance After Tuning: After fine-tuning, the model shows improved performance.
* 5.4 Feature Importance for KNN: We explain which features are important for the KNN model’s decisions.

6. Naive Bayes Model

This model makes predictions based on probabilities, assuming all features work independently.

* 6.1 Evaluate the Model: We measure how well this model detects anomalies.
* 6.2 Hyperparameter Tuning: Adjustments are made to improve the model.
* 6.3 Performance After Tuning: We see improvement after tuning.

7. Logistic Regression Model

This model predicts anomalies based on a mathematical relationship between the features and the outcome.

* 7.1 Evaluate the Model: We assess how well logistic regression performs.
* 7.2 Hyperparameter Tuning: We fine-tune to boost its performance.
* 7.3 Performance After Tuning: Performance improves with tuning.
* 7.4 Feature Importance for LR: We explain which features have the most impact on the predictions.

8. Conclusion

Each model helps us identify equipment failures early on, but some models (like Random Forest and KNN) perform better than others. This approach can be applied across different industries, helping businesses prevent costly machine breakdowns by catching anomalies before they lead to failure. Predictive maintenance through anomaly detection ensures efficient operations, reduces risk, and gives companies actionable insights to improve their processes.