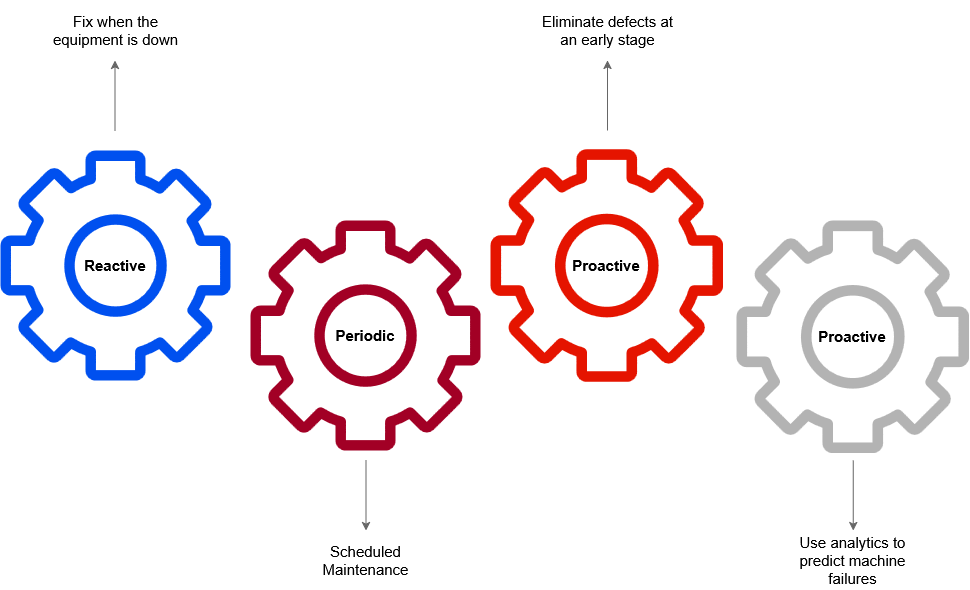
Shashwat Pandey

****

Failure prediction in machines using Ensemble learning and TElemetry analysis

Improving Reliability and Reducing Downtime in Industrial Systems through Predictive Analytics.

****

The maintenance of machines poses significant challenges, leading to frequent breakdowns, decreased productivity, increased downtime, and higher costs. The industry still focuses on REACTIVE, Periodic and PROACTIVE maintenance instead of predictive maintenance.

**What is the Objective of this Project ?**

Our objective is to focus on Predictive maintenance which Leverages historic and real time data to:

* Anticipate and detect potential failure
* Provide early warnings and actionable insights to maintenance teams.
* Reduce the number of unexpected failures

**Description About Dataset :**

**Telemetry Time Series Data:**  It consists of hourly average of voltage, rotation, pressure, vibration collected from 100 machines for the year 2015 and has 8,76,099 rows. Metadata of Machines: it contains Model type & age of the Machines.

**Failures:** Each record represents replacement of a component due to failure. This data is a subset of Maintenance data. This data is rounded to the closest hour since the telemetry data is collected at an hourly rate.

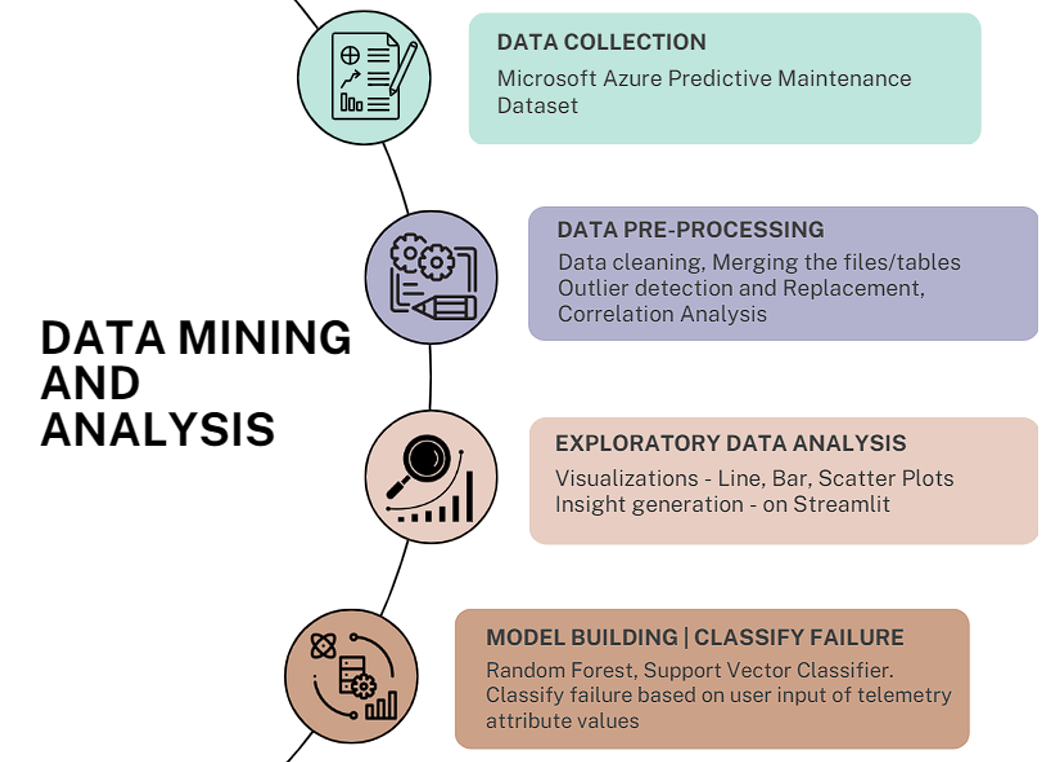
**Errors:** These are errors encountered by the machines while in operating condition. Since, these errors don't shut down the machines, these are not considered as failures. The error date and times are rounded to the closest hour since the telemetry data is collected at an hourly rate.

**Maintenance:** If a component of a machine is replaced. Components are replaced under two situations:

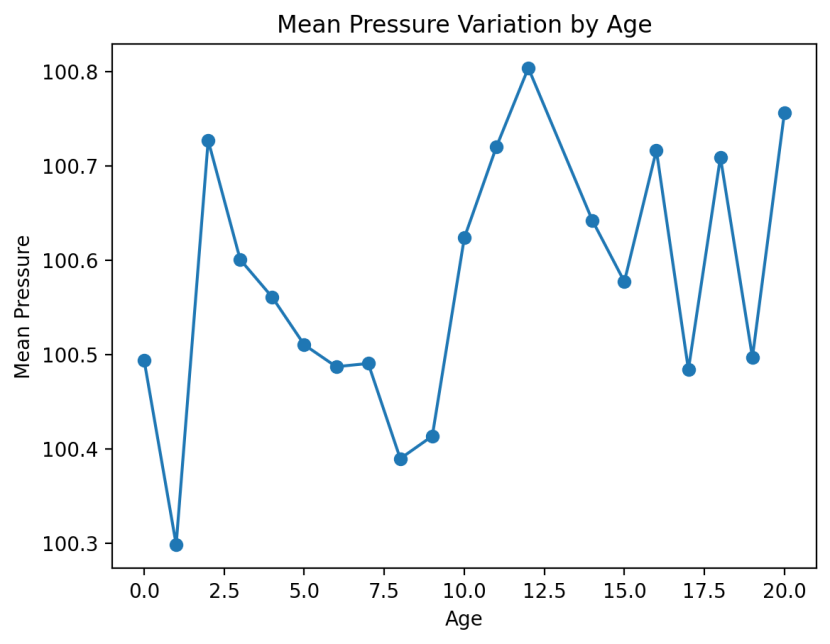
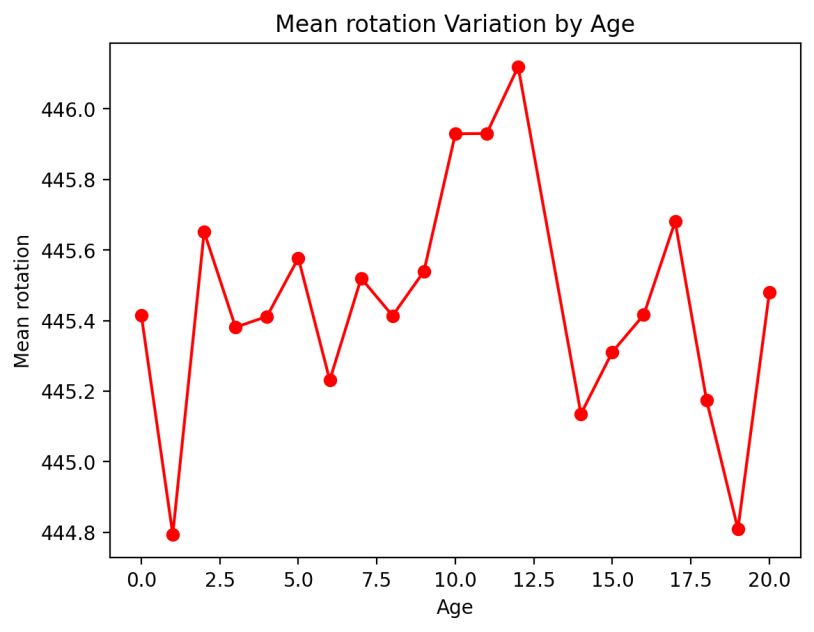
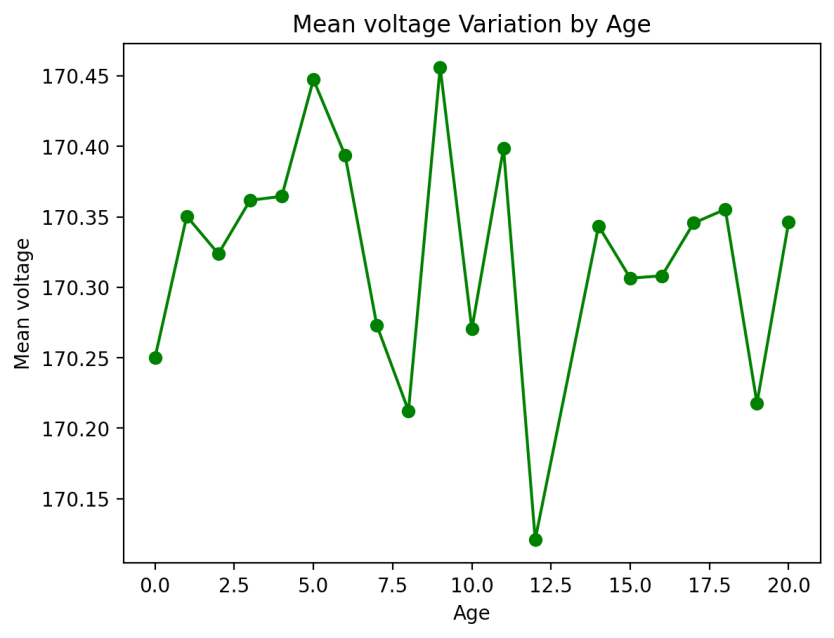
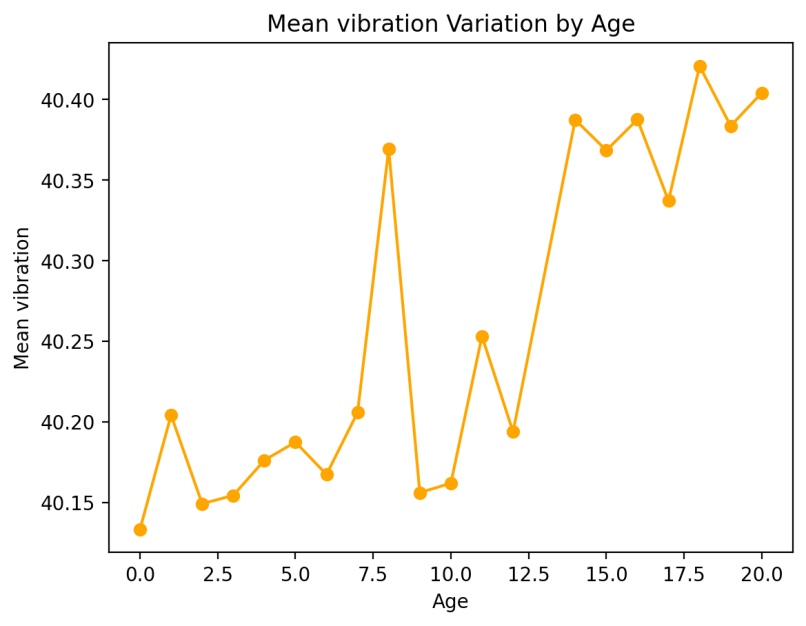
1. During the regular scheduled visit, the technician replaced it (Proactive Maintenance).
2. A component breaks down and then the technician does an unscheduled maintenance to replace the component (Reactive Maintenance). This is considered as a failure and corresponding data is captured under Failures.

Maintenance data has both 2014 and 2015 records. This data is rounded to the closest hour since the telemetry data is collected at an hourly rate.

**Metadata of Machines:** Model type & age of the Machines.



**Variation of Telemetry Attributes w.r.t Age of Machine**

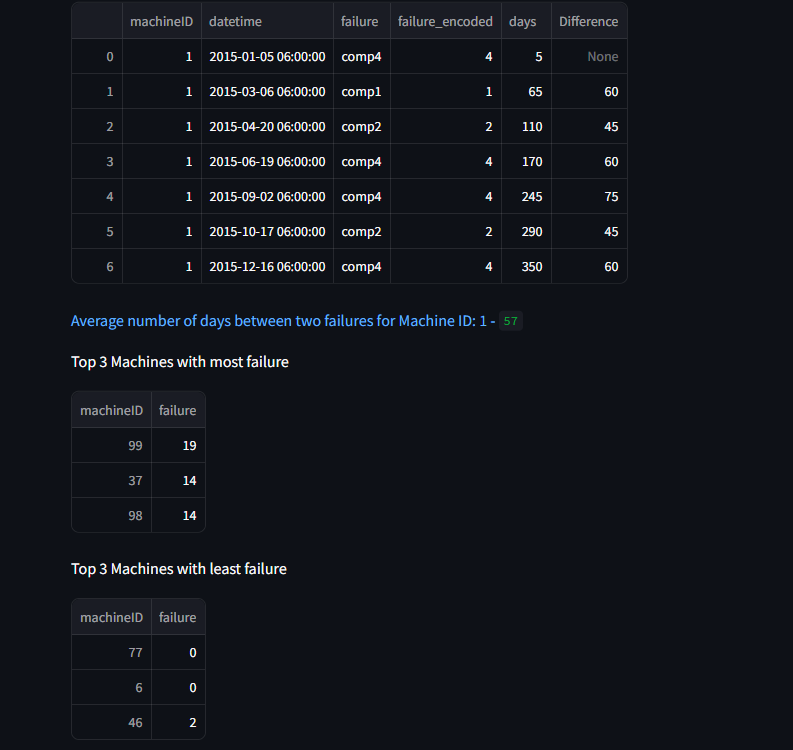


**Machine Failures**

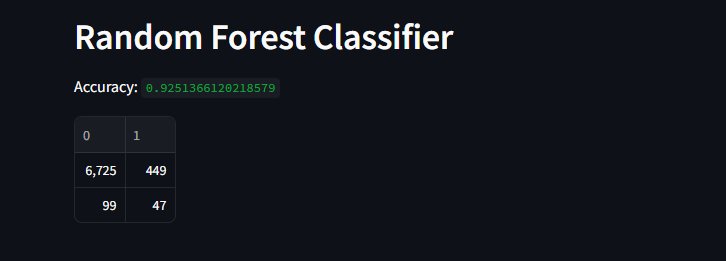
In this analysis, we processed telemetry data and failure logs to study the failure patterns across different machines. The telemetry data, which includes sensor readings such as pressure, voltage, rotation speed, and vibration, was aggregated on a daily basis for each machine. Failures were mapped to binary values and merged with the telemetry data to track when each machine experienced a failure.

For instance, focusing on machineID = 1, we calculated the **average number of days between consecutive failures**, which provides insight into the failure frequency and reliability of the machine. This was done by identifying failure events and computing the time intervals between them.

Furthermore, we identified the **top 3 machines with the highest number of failures**, which helps prioritize maintenance and inspection schedules. Similarly, we also highlighted the **3 machines with the least failures**, potentially indicating stable or under-utilized machines. This analysis can support predictive maintenance efforts by identifying machines that require more frequent monitoring and intervention.



**Model Accuracy Analysis**

****

**High Accuracy is Misleading**

The model achieved 92.5% accuracy primarily by correctly predicting the dominant class (class 0).

This makes accuracy a poor metric in this case.

**Poor Detection of Class 1 (Minority Class)**

Precision: Only 9.4% of class 1 predictions are correct.

Recall: Misses about 68% of actual class 1 samples.

F1 Score is very low (0.145), indicating poor balance between precision and recall.

**Severe Class Imbalance**

Class 0 dominates the dataset (98%), so the model is biased toward predicting class 0.

This imbalance results in underfitting for class 1.

****

While this is a significant drop from the Random Forest accuracy (92.5%), we’ll see why this model might actually be better at handling the minority class, which is often more important in real-world applications.

**Lower Accuracy, But Better Recall for Class 1**

The **recall for class 1 (minority)** has improved **from 32.1% (RF) to 69.2% (SVC)**.

This means the SVC model identifies **more of the true class 1 cases**—which is **crucial in applications like fraud detection, disease diagnosis**, etc.

**Poor Precision for Class 1**

Only **4.2%** of the predictions for class 1 were correct — this indicates **a large number of false positives**.

Model is more "liberal" in predicting class 1 to **capture as many true positives as possible**.

**Model Trade-off: Recall vs Precision**

SVC trades **precision for higher recall**.

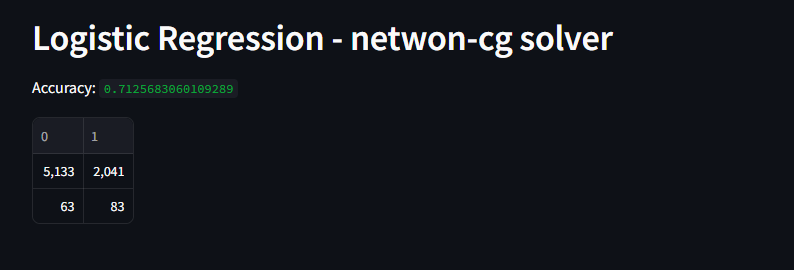
This trade-off may be **desirable or unacceptable depending on context**:

In **medical** scenarios, high recall is preferred (to catch every case).

In **spam filters**, false positives are annoying — so high precision is needed.

**Still Affected by Class Imbalance**

Like with Random Forest, the performance is still skewed by the **imbalanced class ratio**.



**Balanced Trade-off**

**Compared to SVC**, Logistic Regression shows:

**Slightly lower recall** for class 1 (56.8% vs. 69.2%)

**Fewer false positives**, improving **precision** slightly (~4%)

**Higher accuracy** (71.2% vs. 68%)

**Still Poor Precision for Minority Class**

As with SVC, a lot of class 0 samples are being misclassified as class 1 (2041 FPs), leading to a **very low precision (3.9%)**.

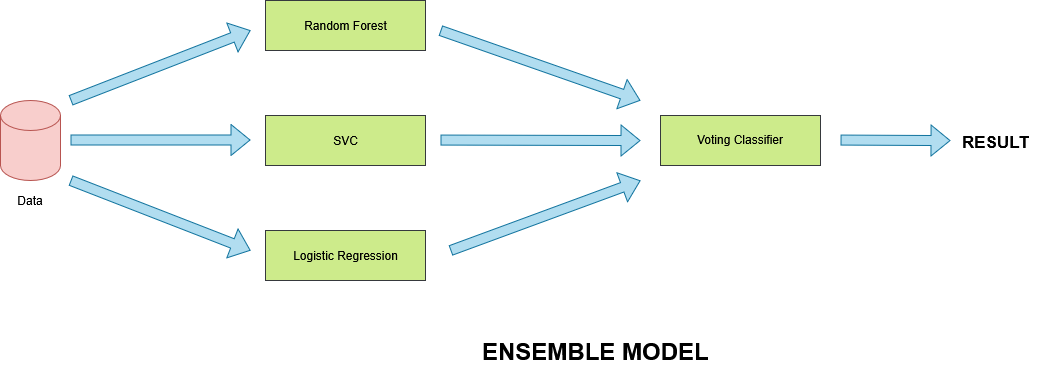
This means many alarms raised by the model for class 1 are **false**.

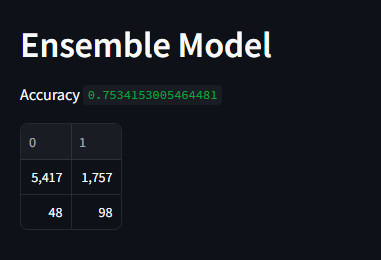
**Class Imbalance Continues to Hurt Precision**

The classifier tries to improve recall on class 1 by **predicting more positives**, but this increases **false positives**.

This results in a **low F1 score (0.074)** for class 1, despite a decent recall.

**ENSEMBLE MODEL**

****

****

This model delivers a strong recall (67.1%) for the minority class while maintaining a moderate level of false positives. Accuracy is highest among non-RF models (75.34%). Class 0 performance remains robust

But compared to other models, it provides the most balanced recall–precision trade-off so far.

The ensemble model smooths out **overfitting** and **underfitting tendencies** of base learners.

The **Ensemble Model** is currently the **best-balanced model** after Random Forest:

* It handles both accuracy and recall well.
* With tuning, it could be **the optimal model for deployment**, especially if interpretability and stability are required.

**Failure Risk Analysis**

The goal of this project is to develop a predictive maintenance dashboard that assesses the risk of machine failure using historical telemetry data and recorded failure events. By applying a machine learning model to recent telemetry readings, we aim to identify machines with a high probability of failure, enabling proactive intervention and reduced downtime.

### ****1. Data Preparation****

Two datasets were used:

* **Telemetry Data (PdM\_telemetry.csv)**: Contains time-series sensor readings (voltage, rotation speed, pressure, and vibration) recorded hourly for each machine.
* **Failure Logs (PdM\_failures.csv)**: Logs the date and type of component failure (e.g., comp1 to comp4) for each machine.

#### ****Processing Steps:****

* **Aggregation**: Telemetry readings were aggregated daily per machine to compute average sensor values.
* **Failure Labelling**: Component-specific failure labels were replaced with a binary indicator (1 = failure, 0 = no failure).
* **Merging**: The daily telemetry and failure datasets were merged on machineID and datetime.
* **Cleaning**: Missing values were handled, and any string '**nan**' in the failure column was replaced with actual **NaN** values before being binarized.

### ****2. Failure Frequency Analysis****

**Using the processed dataset:**

* The **total historical failure count per machine** was computed**.**
* Machines were ranked to identify the **Top 3 machines with the highest number of failures** as well as those with the least failures.
* For example, historical failure analysis for machineID=1 revealed patterns of repeated failure, which can inform maintenance scheduling.

### ****3. Predictive Model****

An **ensemble machine learning model** was used to predict the probability of failure using the following features:

* volt, rotate, pressure, and vibration

#### ****Model Integration:****

* A pre-trained model (ensemble\_model.pkl) was loaded.
* Predictions were generated for the most recent 90 days of telemetry data per selected machine.
* Each record was assigned a **failure probability** between 0 and 1.

