Alec Gomez

Predictive Machine Status Analysis

DATA PREPROCESSING

Introduction - The primary objective of data preprocessing is to prepare the sensor data for exploratory data analysis and the subsequent predictive modeling. Preprocessing data is essential for creating a predictive model for and ensuring model accuracy. To achieve this, methods such as missing value treatment, outlier detection, standardization, and encoding were used. Initial examination of the data revealed that there were, in fact, several outliers, empty columns and unstandardized data which needed to be handled through these methods. The dataset also contained continuous readings from the sensors and timestamp information which could be cleaned up for a more efficient analysis.

Missing Value Treatment - The initial step taken to begin data preprocessing was identifying columns which had float values and contained empty cells. Following the identification of missing values, the reprocess_data function calculated the mean value of a sensor's readings and filled missing values with the mean to maintain data integrity and consistency. Finally, the function identified empty columns, such as *sensor_15*, and dropped them from the dataset. In addition, the predictive models chosen are far more efficient when there are no null or missing values in the dataset.

Outlier Detection - Using the Interquartile Range method, outliers were identified and capped at the maximum and minimum values to prevent large variations in data analysis. This way, data can be more consistent and focused for the standardization phase. I opted for this approach rather than removing outliers because this is a large dataset with potentially many outliers, so avoiding holes in the data was necessary. Maintaining the same dataset size was important, as I wanted to ensure that these models could still perform efficiently on large datasets. In addition, capped 'outliers' can still be visible in the data without distorting the results of the analysis.

Data Standardization - Following the outlier detection and treatment, the sensor data was standardized to have zero mean and variance of one using the Z-score method. The normalization of the data is crucial for ensuring a proper comparison of readings across all sensors, as all of them may have different scales and readings. Once the data was standardized, all readings were rounded to clean up after all of the transformations.

Encoding Machine Status - The dataset contained a string column 'machine_status', which identified whether the machine was BROKEN, NORMAL and RECOVERING. This would later be used as a target variable for predictive modeling, however it had to be encoded to a categorical column of integers 0, 1 and 2 respectively. When the encoding process was finished, the final state of the data was saved as *processed_sensor_data.csv* and accessed via a *processed_df* variable for easy modeling and comparison to the original dataset.

EXPLORATORY DATA ANALYSIS

Correlation Between Sensor Readings and Machine Status - The exploratory analysis began with an examination of the relationships between sensor readings and machine status. A correlation matrix heatmap and sensor-specific box plot was generated to visualize the strength and direction of the relationships between all pairs of sensors, as well as each sensor's relationship with the machine status.

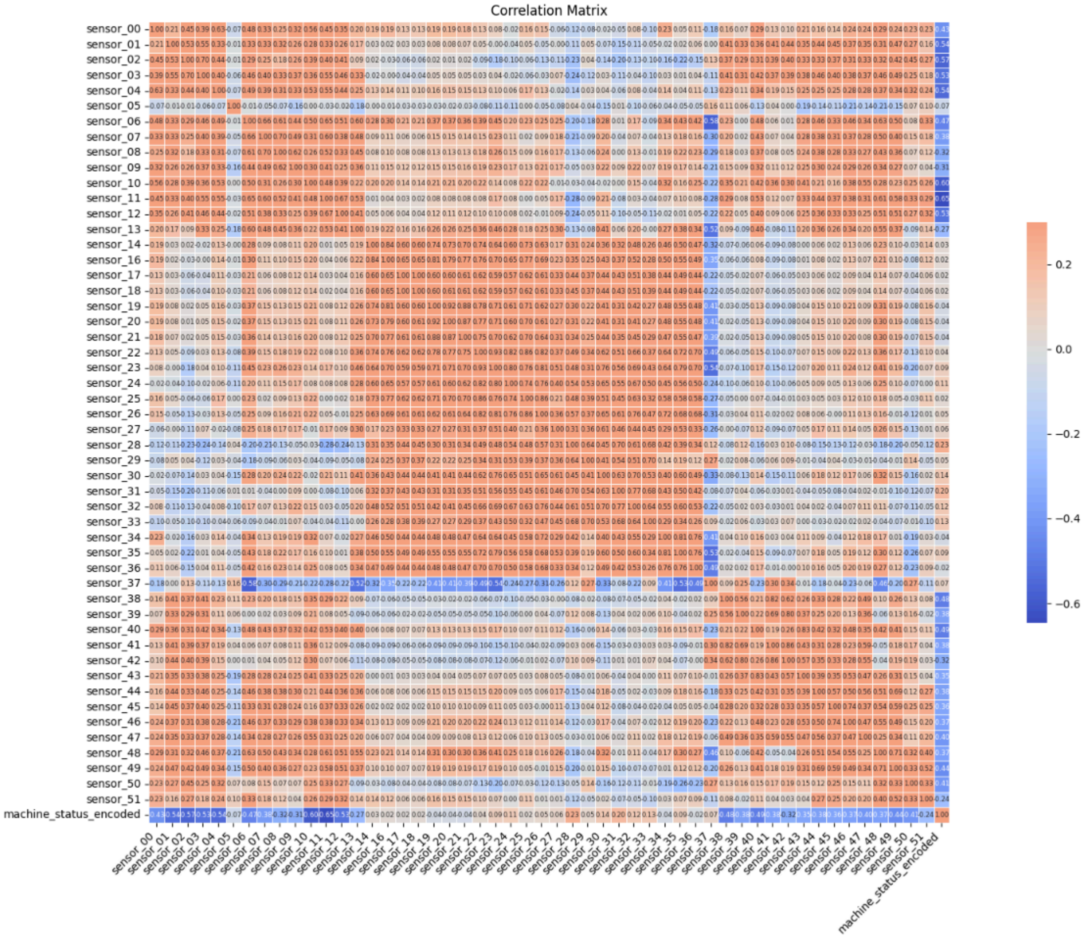


Figure 1

Findings from Correlation Matrix:

- The heat map revealed clusters of sensors with strong intercorrelations, indicating groups of sensors that respond similarly to the machine's operational conditions
- The matrix provided a visual indication of the most relevant sensors to the machine status. Positive correlations (indicated by red hues) suggested a direct relationship, whereas negative correlations (indicated by blue hues) suggested indirect relationships.
- Some sensors which had a particularly poor relationship with machine status were sensors *5* and *37*.

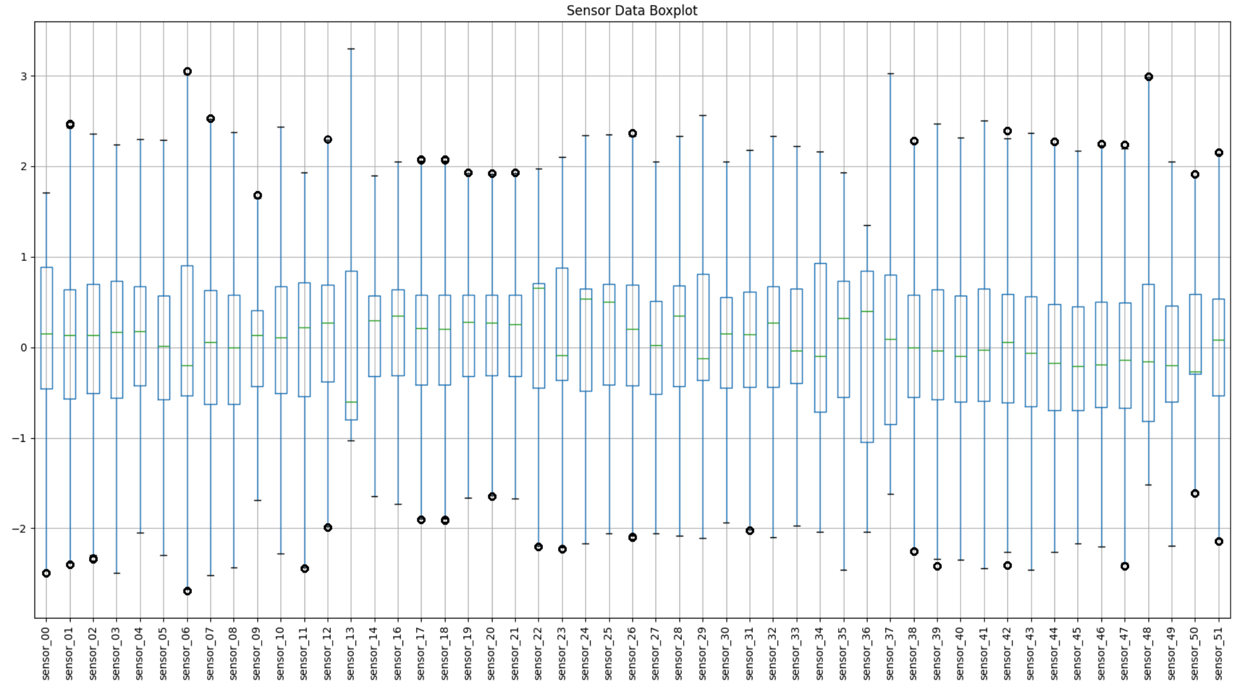


Figure 2

Boxplot Macro-Analysis - Boxplots for each sensor were constructed to investigate their distribution and identify any outliers (*Figure 2)*. These boxplots provided an insight into the central tendency, dispersion, and skew of the sensor data distributions as a whole.

Findings from Box Plot Macro-Analysis:

- The median value, illustrated by the green line within each of the box plots, allowed me to compare the central location of each sensor's readings.
- The spread of the boxes provide a visual summary of the variability and the whiskers provide a visual of the range within each sensor.
- Several sensors display outliers which suggest sporadic readings. This may suggest errors or potential failure.
- Most sensors display a fairly normal distribution of readings, however some sensors have a skewed box which may suggest readings that trend higher or lower than the median.
- Sensors which display abnormal patterns may need to be investigated further.

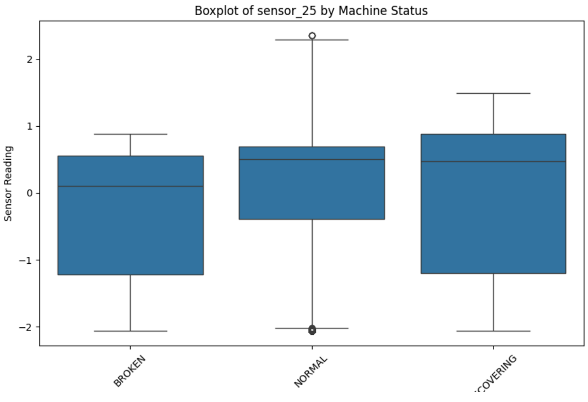


Figure 3

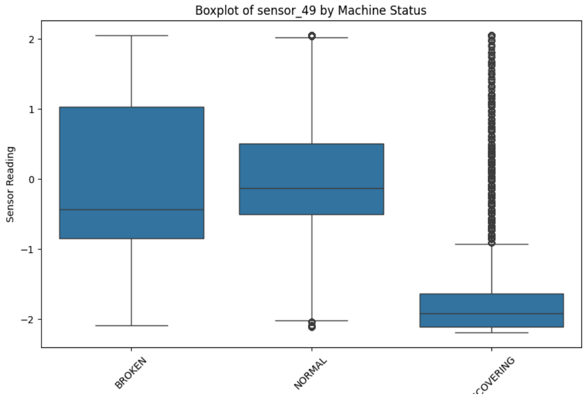
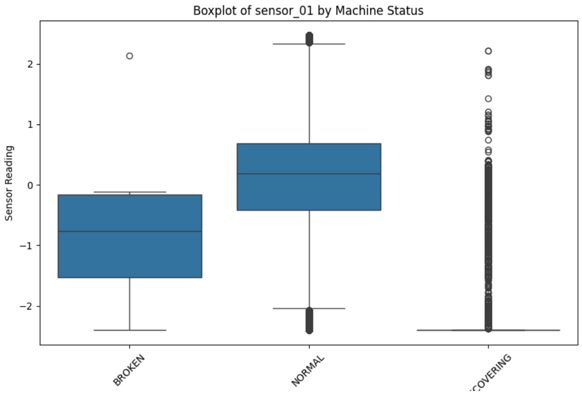


Figure 4



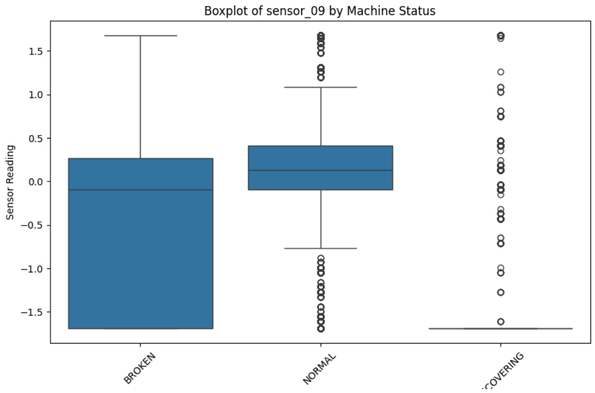


Figure 5

Figure 6

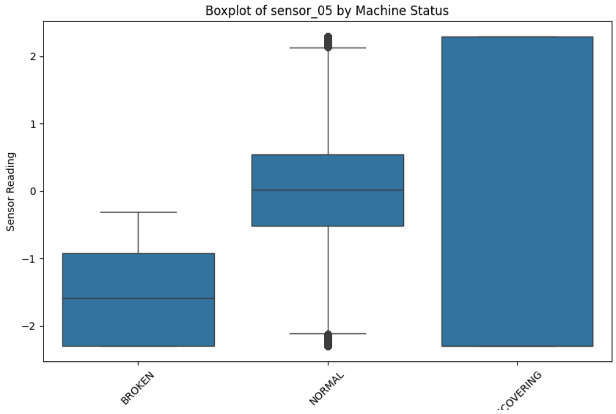


Figure 7

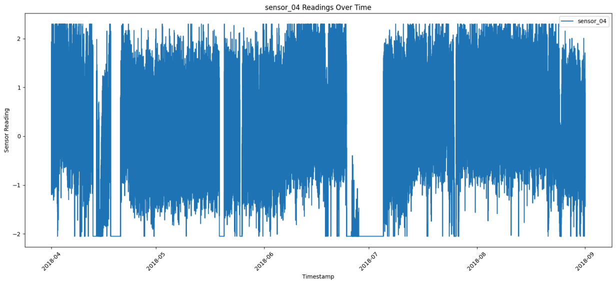
Boxplot Micro-Analysis - Unlike the macro-analysis boxplot, this analysis was a micro-analysis of each individual sensor, where data from a selected sensor can be viewed given its state. This way, individual sensors which stood out as unusual in the macro-analysis can be evaluated individually.

Findings from Box Plot Micro-Analysis:

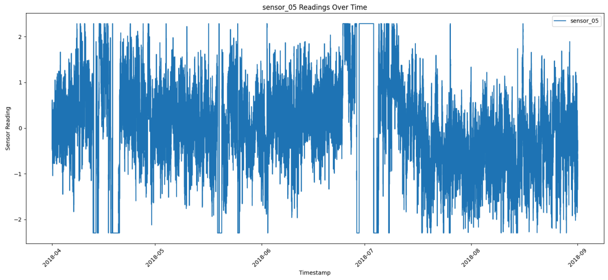
- The vast majority of box plots show the same trends between the three states of sensors. While there were a few anomalies, several trends were constant throughout my analysis.

- BROKEN machines exhibited a significantly higher variance in sensor readings than that of NORMAL machines (*Figure 3* and *Figure 4)*.
- NORMAL machines exhibited mostly stable readings, which is suggested by the lack of or low amount of outliers in the data, as well as significantly lower variance than that of broken machines.
- RECOVERING machines tend to exhibit a high amount of outliers with much lower readings overall, suggesting a consistent shift in the sensor's behavior during the machine's recovery phase. The majority of sensor data suggest that the "RECOVERING" phase is highly volatile with large variations, and, in some cases, there might be no consistent readings around which the data centralizes. This could be due to sporadic and extreme variations in sensor readings during recovery. As shown in *Figure 5*, some sensors appear to have no mean and variation in this state, whereas others have such large variation that there are no identifiable outliers or mean (*Figure 7)*. To reiterate, the RECOVERING state is highly volatile as the machine slowly returns to its NORMAL state.

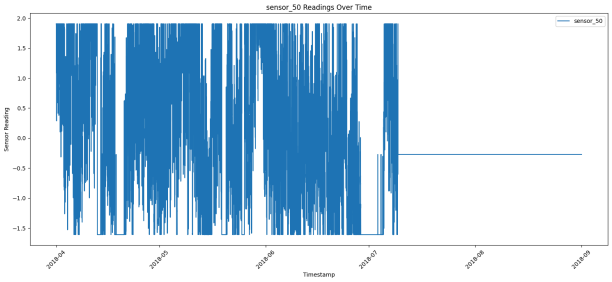
Temporal Analysis of Sensor Readings - To understand how sensor readings evolve over time, series plots for each sensor were generated. This method helps to better understand the trends in sensor readings by comparing the visuals from sensor to sensor.



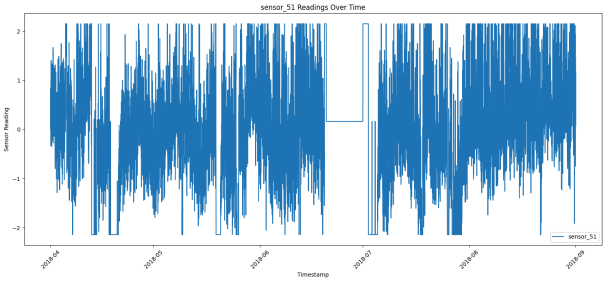
Sensor 04, Figure 8



Sensor 05, Figure 9



Sensor 50, Figure 10



Sensor 51, Figure 11

Findings from Temporal Analysis:

- Some sensors had very high variance over time and appeared to have several outliers which were capped by the standardized data the analysis was performed on (*Figure 8)*.
- Certain sensors demonstrated regular cycles or patterns, suggesting a routine machine operation (*Figure 9*). The analysis suggests that most machines either underwent maintenance or had an outage in 2018-04 and 2018-07.

- Some sensors visibly show irregularities and abrupt changes in behavior (*Figure 11)* suggesting that there are possible breakdowns or irregularities. Sensor 50 (*Figure 10)* stood out as the analysis suggests that it broke down completely and may be susceptible to future malfunction.

FEATURE ENGINEERING

Selecting Sensors to Extract - In order to identify sensors that may be indicative of impending equipment failures, I used sensor data from the EDA as well as histograms (*Figure 12)* to view the distribution of data for each sensor. Sensors which may lead to equipment failures were ones which had very abnormal distributions or, in rare cases, little to no data at all.

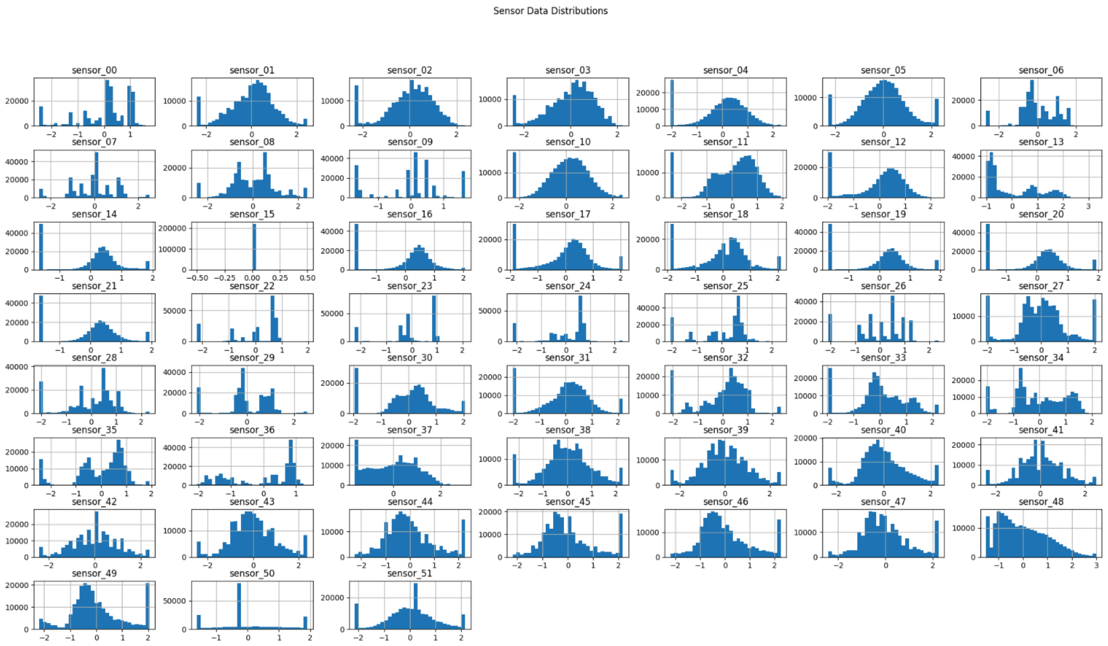


Figure 12

Engineering Data:

Sensors indicative of equipment failures were extracted from the processed dataset and put through an analysis, which calculated the rolling mean, rolling standard deviation, and lagged features of the sensor's readings, in addition to the data which was already present. This data was then saved as *engineered_sensor_data.csv* for viewing and comparison to the processed and original data.

Interpreting Engineered Data:

- Original Sensor Readings represent the standardized (mean of zero and std deviation of one) readings from the sensors at each timestamp.
- Rolling Mean columns represent the rolling average of the sensor readings over time, which smooths out minor fluctuations in order to have more accurate longer-term data trends.
- Rolling Standard Deviation columns are the rolling standard deviation, providing a sense of the variability of the sensor readings over time. Higher values indicate more fluctuation in sensor readings within the timestamp periods.
- Lagged Features columns contain the sensor readings from a previous timestamp. These capture the temporal sequence of readings.
- Mean vs Lagged Features can be compared to determine whether the sensor's current reading is above or below its recent average.
- Changes in lagged features compared to current readings can suggest shifts in the machine's condition.

PREDICTIVE MODELING

Machine Health Modeling - The primary aim of predictive modeling was to anticipate the status of equipment based on readings from various sensors indicative of impending failure. By forecasting equipment conditions, using these models can help prevent failures, plan maintenance effectively, and ensure operational continuity.

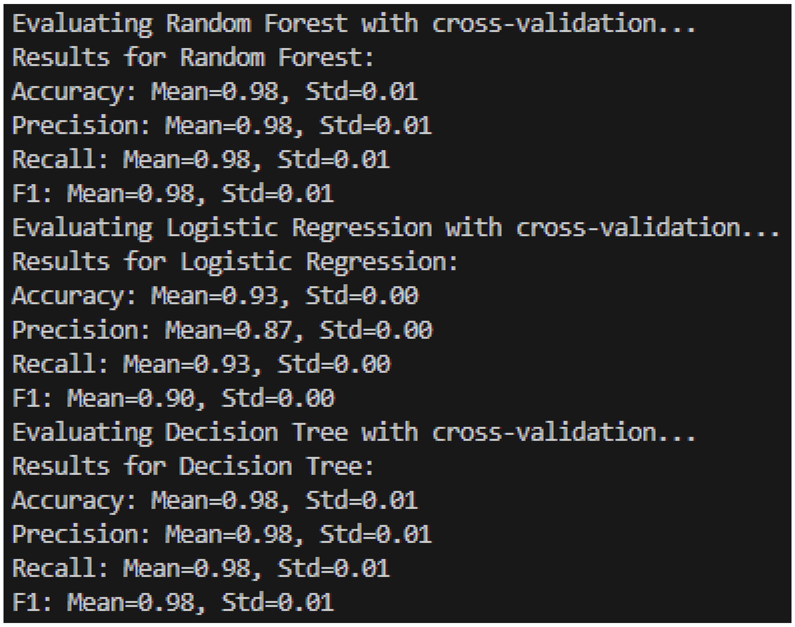
Models Selected for Forecasting

- **Random Forest Classifier -** A group of decision trees that work together to improve accuracy and reduce the chance of overfitting, which occurs when your model learns the present data but has trouble picking up new data. Random Forest Classifier is best used when you have a complex dataset with different data types, which is why this model was the first one chosen.
- **Logistic Regression -** A linear approach which models the probability of a certain class or event existing. Logistic Regression is a very time-efficient model which is very useful for data that isn't too complex. In this case, the data has been cleaned up and standardized, which makes the estimated time efficiency of Logistic Regression much higher than the other models chosen.
- **Decision Tree Classifier -** A decision tree works as a flowchart which makes decisions by splitting the data into branches based on target values. Each branch splits into more branches, from root to leaf. Decision Trees, contrary to logistic regression do not operate linearly, which is why I was curious to see how it would compare to Logistic Regression.

Performance Metrics:

- **Accuracy**: Reflects the overall correctness of the model.
- **Precision**: Indicates the quality of positive predictions.
- **Recall**: Measures the model's ability to capture actual positives.
- **F1 Score**: Balances precision and recall in a single metric, particularly useful when there is an uneven class distribution.

MODEL EVALUATION AND SELECTION



```
Evaluating Random Forest with cross-validation...
Results for Random Forest:
Accuracy: Mean=0.98, Std=0.01
Precision: Mean=0.98, Std=0.01
Recall: Mean=0.98, Std=0.01
F1: Mean=0.98, Std=0.01
Evaluating Logistic Regression with cross-validation...
Results for Logistic Regression:
Accuracy: Mean=0.93, Std=0.00
Precision: Mean=0.87, Std=0.00
Recall: Mean=0.93, Std=0.00
F1: Mean=0.90, Std=0.00
Evaluating Decision Tree with cross-validation...
Results for Decision Tree:
Accuracy: Mean=0.98, Std=0.01
Precision: Mean=0.98, Std=0.01
Recall: Mean=0.98, Std=0.01
F1: Mean=0.98, Std=0.01
```

Figure 13

Random Forest - Exhibited high performance across all metrics with an accuracy, precision, recall, and F1 score consistently around 0.98, showcasing its powerful capability in handling the sensor data for predictive purposes.

Logistic Regression - Showed lower performance than Random Forest with an accuracy of 0.93 and a precision and F1 score slightly lower, suggesting some limitations in handling the complexity of the sensor data.

Decision Tree - Performed on par with the Random Forest, with mean scores of 0.98 across all metrics, implying that for this dataset, the simpler model did not sacrifice accuracy for interpretability.

Model Selection Conclusion - While all three models are very capable of identifying patterns in the sensor data, the Random Forest and Decision Tree models proved to be more highly effective than the Logistic Regression. While Logistic Regression was much more time efficient than the other two, the mean results ranging from 0.87-0.93 do not compete with Random Forest and Decision Tree which yielded means of 0.98 across the board. All factors considered, the superior model was the Random Forest model. Random Forest maintained an evaluation speed between Logistic Regression and Decision Tree, with the accuracy of the latter.

PRACTICAL IMPLICATIONS AND CONCLUSIONS

Maintenance Scheduling and Operational Efficiency - By identifying patterns and outliers in sensor readings that correlate with different machine statuses, these models can help forecast machine breakdowns and allow for more proactive maintenance scheduling. Understanding which sensors are indicative of machine failures allows for sensor-specific monitoring and prevention of breakdowns.

Improved Machine Health Monitoring - Integrating my model and engineered data into real-time monitoring systems can provide immediate alerts when sensor readings suggest a potential failure or sporadic readings deviating from the norm. The use of historical sensor data allows predictive models to assess current sensor data trends and can always be updated to learn as time goes on, enhancing accuracy of future predictions.

Resource Allocation - By predicting which machines are likely to fail, resources can be allocated more effectively, such as maintenance team schedules and time commitment to monitoring specific machines.

Cost Efficiency - Effective predictive maintenance can significantly reduce costs associated with machine failures and downtime. It may be more cost efficient to identify potential failures and take preventive measures rather than repair after a machine has broken down. This way, companies can avoid the higher costs of extensive repairs or replacements.