# Danfoss Power Solutions

## Predictive Maintenance Analysis Project Report

**Submitted By: Mandar Chaudhari**

---

## 1. Introduction

The purpose of this report is to present the findings and analysis conducted as part of the Predictive Maintenance Project for Danfoss Power Solutions. The focus is on identifying patterns and insights from the maintenance data to enhance predictive maintenance strategies.

## 2. Data Import and Preparation

The dataset was imported from an Excel file named `Maintenance Without Problem and Solution.xlsx`. The following libraries were used for data analysis: pandas, NumPy, seaborn, matplotlib, and calendar.

## 3. Data Cleaning and Transformation

- Renaming Columns

  The columns in the dataset were renamed for better readability:

- `Location Description` to `location`
- `Asset Description` to `asset`
- `Action Taken` to `action`
- `Total Down Time (Mins)` to `down_time_in_mins`
- `Type of Work` to `work_type`
- `Problem` to `problem`

- Converting Day, Month, Year into a Date Column

The `Month` column, initially in textual format (e.g., JAN, FEB), was mapped to numerical values. The `Day`, `Month`, and `Year` columns were then combined into a single `date` column. The original `Day`, `Month`, and `Year` columns were dropped after this transformation.

| | Shift | location | asset | down_time_in_mins | work_type | date |
|---|---|---|---|---|---|---|
| 0 | A | CH02 | Test Rig D1 | 30.0 | B/D | 2024-01-02 |
| 1 | A | PDI line | PDI Assembly & PDI Packing CH01 Line 1 | 60.0 | B/D | 2024-01-02 |
| 2 | A | CH03 | Assembly station 3 | 60.0 | B/D | 2024-01-02 |
| 3 | A | CH06 | Durr Ultrasonic | 30.0 | B/D | 2024-01-02 |
| 4 | A | CH04 | Test Rig F2 | 30.0 | B/D | 2024-01-02 |

## 4. Handling Missing Values

The dataset was inspected for missing values. Any rows with missing values were removed to ensure data integrity. After dropping the missing values, no null entries remained.

## 5. Data Cleaning

- Whitespace was trimmed from string columns to ensure consistency in the data and avoid extra blank space errors.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3091 entries, 0 to 3090
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Shift              3088 non-null   object
 1   location           3088 non-null   object
 2   asset              3088 non-null   object
 3   down_time_in_mins  2993 non-null   float64
 4   work_type          3088 non-null   object
 5   date               3088 non-null   datetime64[ns]
dtypes: datetime64[ns](1), float64(1), object(4)
memory usage: 145.0+ KB
```

- Removal Of Null (Void) Values from the dataset. (Below is null value count)

```
Shift                 3
location              3
asset                 3
down_time_in_mins    98
work_type             3
date                  3
dtype: int64
```

## 6. Data Exploration

- There are a total of 24 Unique Maintenance Locations.
- Below is the list of all the unique maintenance locations in alphabetical order.

```
Total number of unique locations: 24
```

BV LINE
CH01
CH02
CH03
CH04
CH05
CH06
GS01
GS02
GS03
GS04
GS05
Milipore Room
PDI line
PVG
PVLP
PVM
SS01
SS02
SS03
SS04
SS05
Store
Tool Room

- There are a total of 100 Unique Maintenance Assets (Machines).
- Below is the list of all the unique maintenance assets present in alphabetical order.

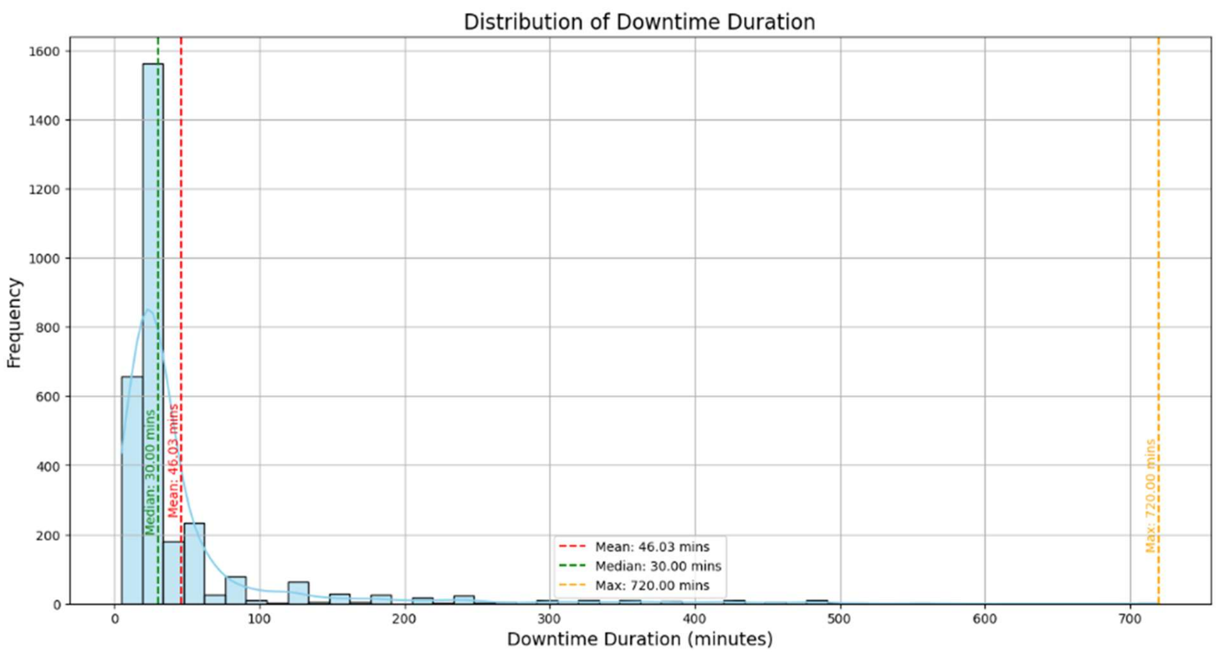
Total number of unique assets: 100

Assembly station 1
Assembly station 2
Assembly station 3
Assembly station 4
Assembly stations of CH01
Assembly stations of CH02
Assembly stations of CH03
Assembly stations of CH04
Assembly stations of CH05
Assembly stations of CH06
Automatic assembly
BVLINE Assembly Station(Dual Stage)
BVLINE Assembly Station(Single Stage)
BVLINE ECO 80 Assembly
BVLINE ECO 80 Test Stand
BVLINE NGCT Big
BVLINE Nitrogen Gas Generator Unit
BVLINE PVG Assembly
BVLINE PVG Test Stand
BVLINE PVM Assembly
BVLINE PVM Test Stand
BVLINE Test Stand A (Single stage)
BVLINE Test Stand A(Dual stage)
BVLINE Test Stand B(Dual stage)
Ball coining
Ball feeder of Assembly station
Bolt torquing (Assembly Station)
Centre Lapping
Cleantech
Cobot of Studer
Cobot of Studer S33
Cobot of kadia
Deburring Machine
Durr Ultrasonic
Dust Seal
ELB China
ELB Oil Filtration Unit
ELB Rim
ELB Wheel
EMAG Rim
EMAG Rim & Wheel
EMAG Wheel
EMAG Wheel conveyor
Jerk Testing
KLEIN
KLEIN 3
Kadia
Kadia 40
Kapp Rim
Kapp Wheel
Klein 1
Klein 2
Knoll Coolant System
Micromatic Grinding
Milipore Room
Nissin
PDI Assembly & PDI Packing CH01 Line 1
PDI Assembly & PDI Packing Line 2
PDI Line Cobot
PV Assembly & Leak Testing
PVG Assembly
PVG Test Stand
PVM Assembly
PVM Test Stand
Piston Coining
Stahli S/F Grinding Machine 1
Stahli S/F Grinding Machine 2
Store Laser Punching Machine 1
Store Laser Punching Machine 2
Store Punching Machine 1
Studer 1 (450)
Studer 2 (451)
Studer 3 (452)
Studer Favorit
Studer S11
Studer S22
Studer S33
Studer S36
TOOL ROOM Tshudin
Teijo
Test Rig A & B
Test Rig A
Test Rig B
Test Rig C
Test Rig D
Test Rig D1
Test Rig D1 & D2
Test Rig D2
Test Rig E1
Test Rig E1 & E2
Test Rig E2
Test Rig F1
Test Rig F1 & F2
Test Rig F2
Test Rig G1
Test Rig G1 & G2
Test Rig G2
Test Rig H1
Test Rig H1 & H2
Test Rig H2

## 7. Descriptive Statistics and Visualizations

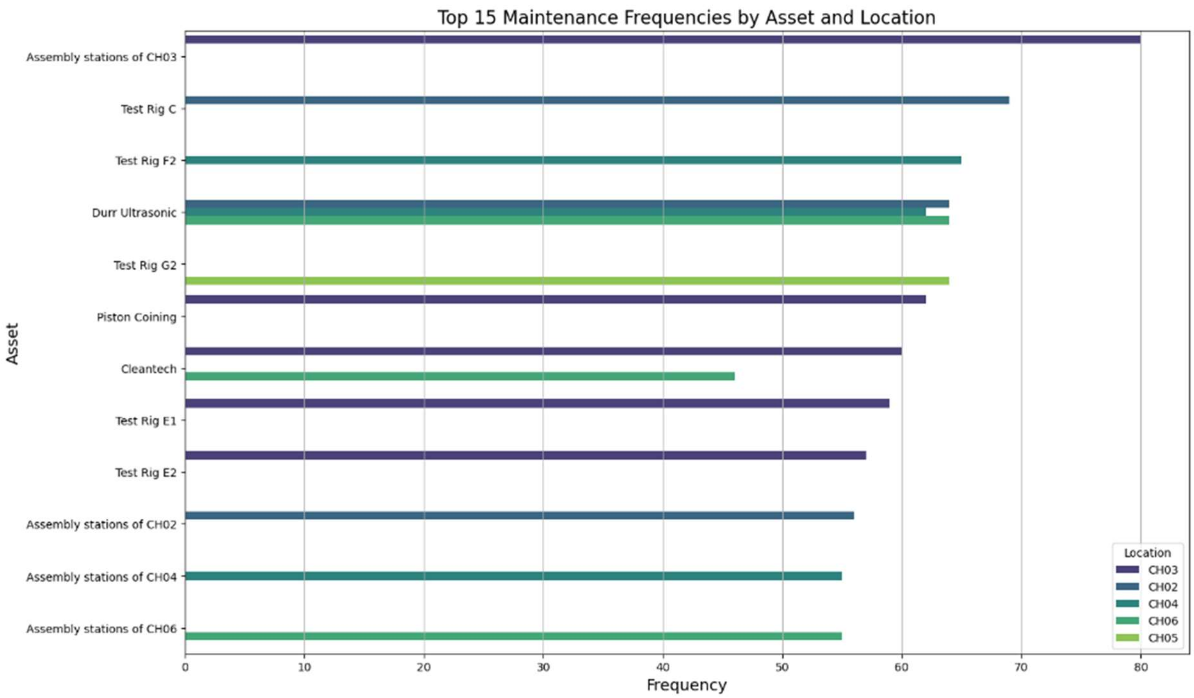
- Descriptive Statistics

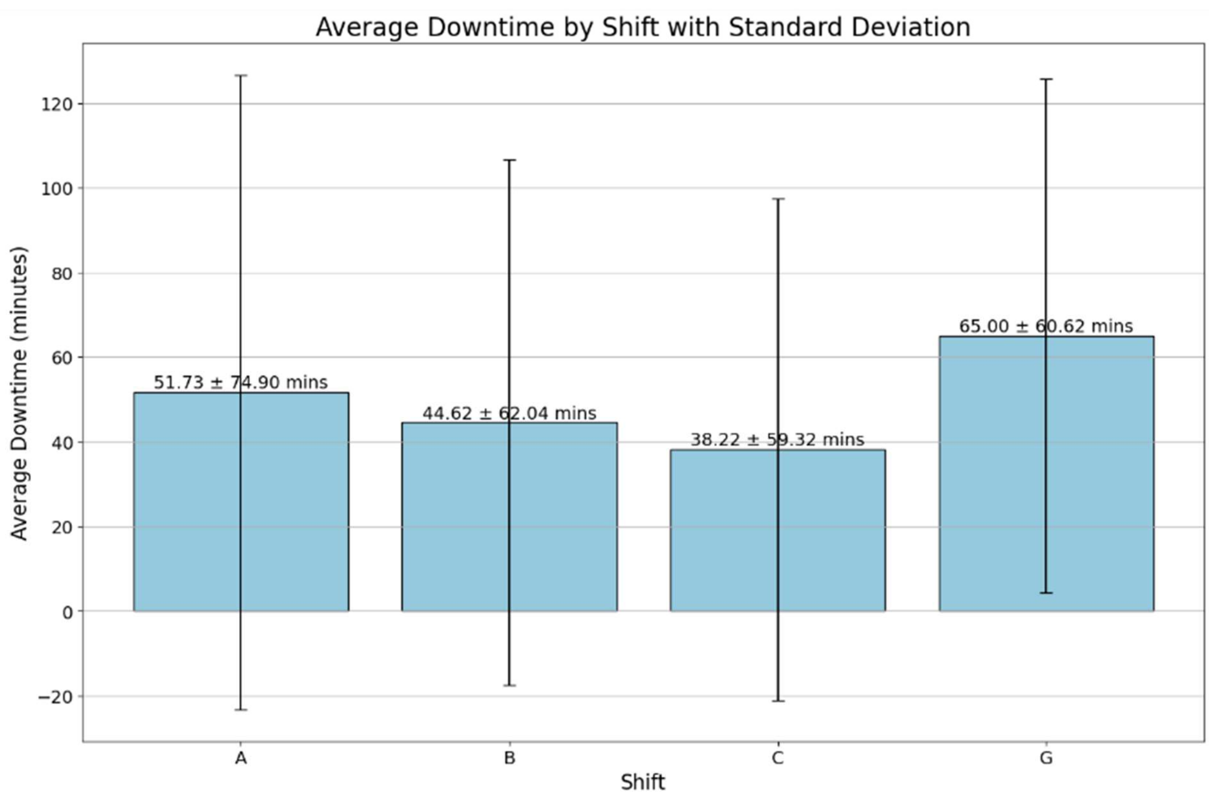
Key statistics of the dataset were computed, providing insights into central tendencies, dispersion, and overall shape of the dataset's distribution with respect to various features which have affected the downtime of machines during maintenance.

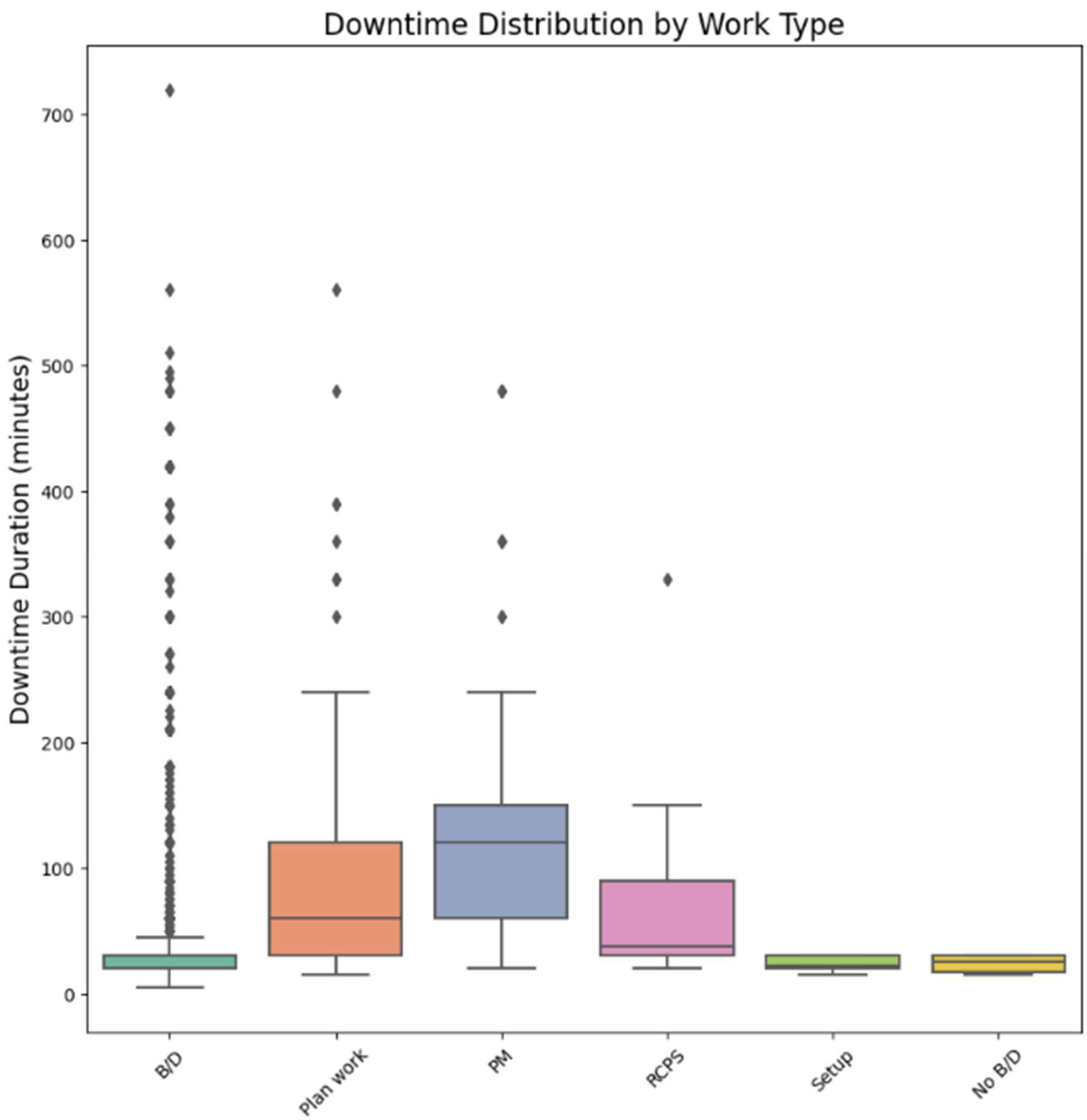



- Maintenance Downtime Patterns

Visual Analysis and Graphs, Plots according to how to maintenance time varies according to various factors such as "SHIFT", "WORK TYPE", "ASSET", "LOCATION", "MONTH WISE DISTRIBUTION".


Top 15 Maintenance Frequencies by Asset and Location
Assembly stations of CH03
Test Rig C
Test Rig F2
Durr Ultrasonic
Test Rig G2
Piston Coining
Cleantech
Test Rig E1
Test Rig E2
Assembly stations of CH02
Assembly stations of CH04
Assembly stations of CH06
Asset
Frequency
Location
CH03
CH02
CH04
CH06
CH05


Average Downtime by Shift with Standard Deviation
51.73 ± 74.90 mins
44.62 ± 62.04 mins
38.22 ± 59.32 mins
65.00 ± 60.62 mins
Average Downtime (minutes)
Shift
A
B
C
G

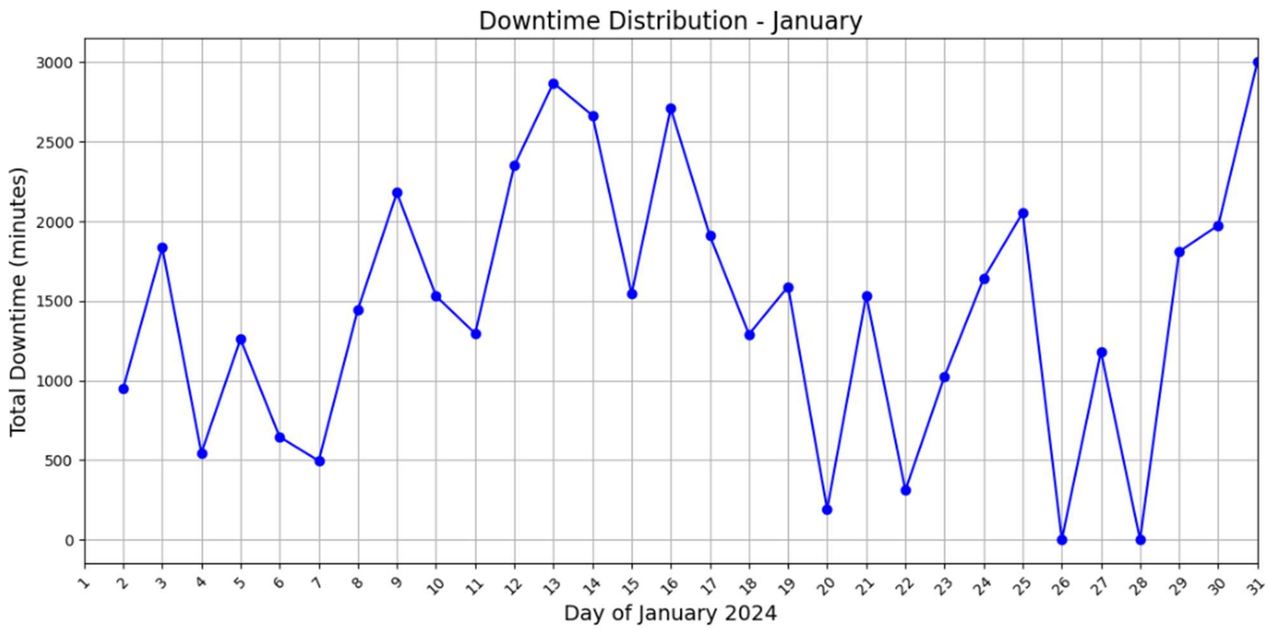



- Maintenance Trends Over Time

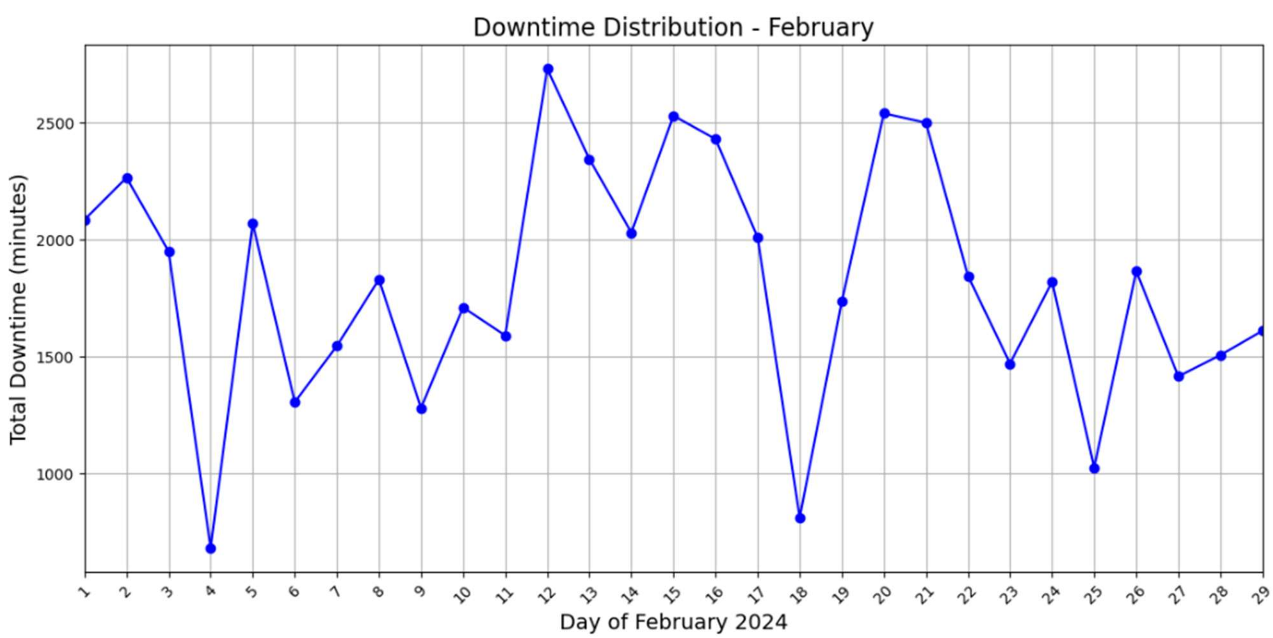
A count plot of maintenance activities over time revealed trends in maintenance patterns. Peaks and troughs in maintenance activities were observed, indicating potential periods of higher maintenance demand.

The graph plots for the months January, February, March, April, (1st Quarter) of the year 2024 are shown below.

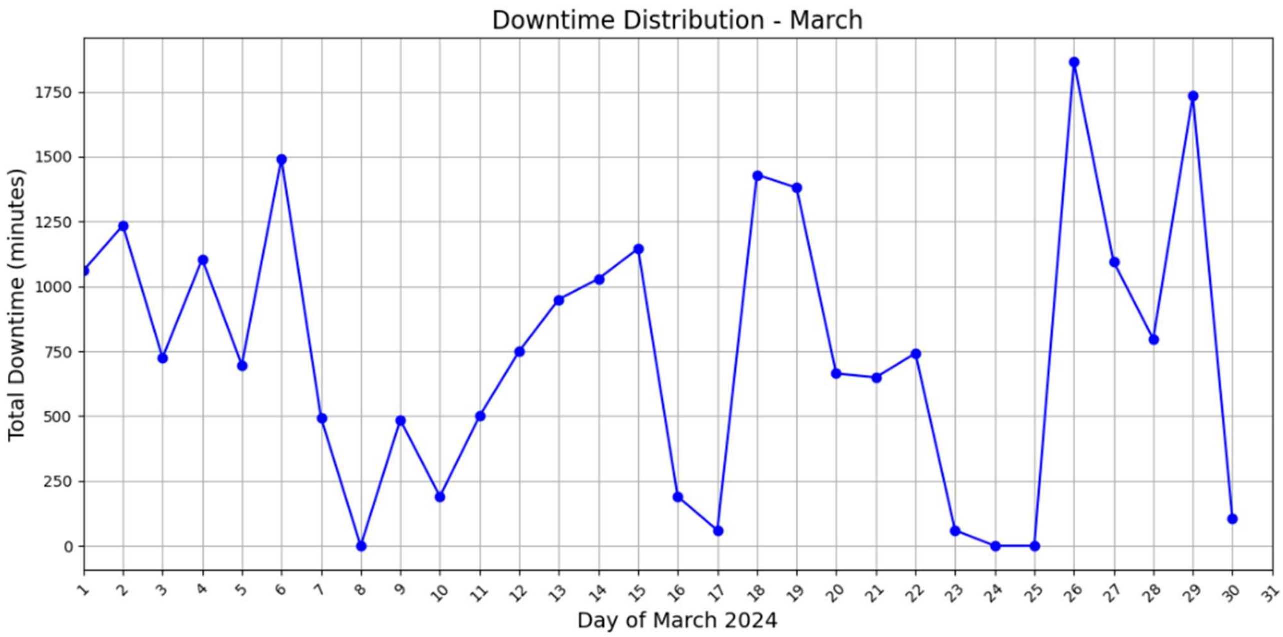
- JANUARY




- FEBRUARY

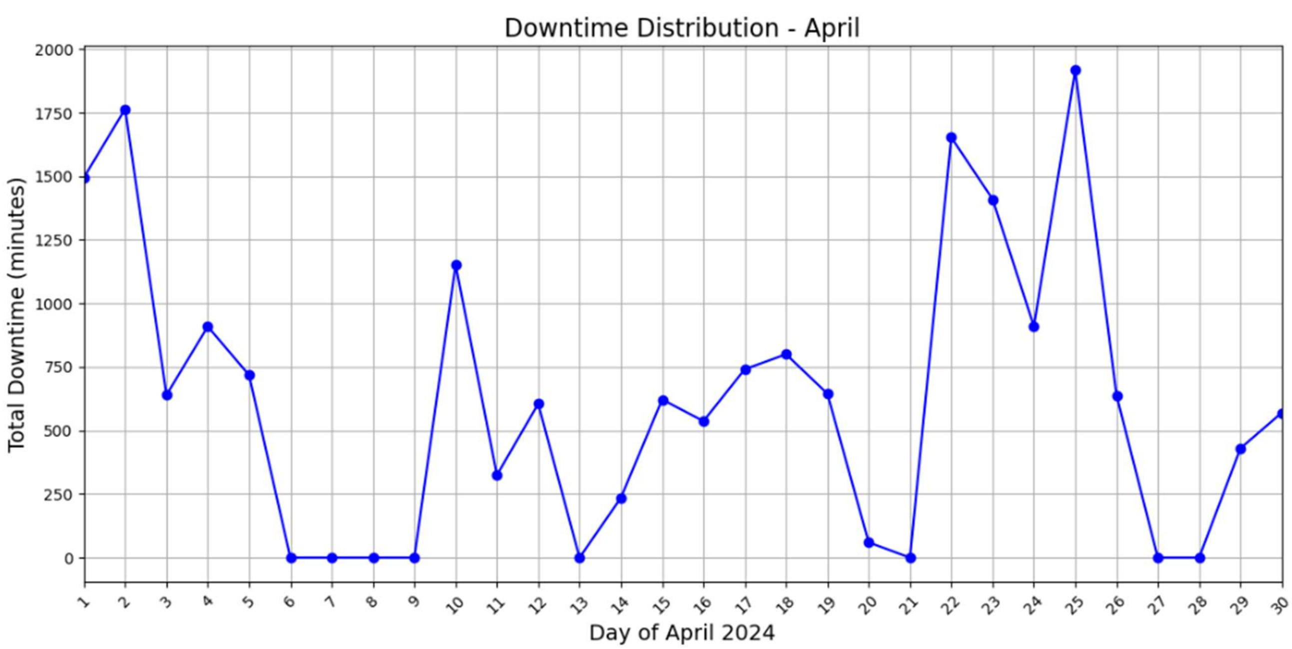


- MARCH



Downtime Distribution - March
Total Downtime (minutes)
0
250
500
750
1000
1250
1500
1750
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Day of March 2024


- APRIL



Downtime Distribution - April
Total Downtime (minutes)
0
250
500
750
1000
1250
1500
1750
2000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Day of April 2024

# 8. Predictive Maintenance Model Building

```
Selected features: Index(['Shift', 'location', 'asset', 'work_type'], dtype='object')
```

```python
# Define the deep learning model with early stopping
def create_model(input_shape):
    model = Sequential([
        Dense(128, activation='relu', input_shape=(input_shape,)),
        Dropout(0.5),
        Dense(64, activation='relu'),
        Dropout(0.5),
        Dense(1)  # Output layer for regression
    ])
    model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])
    return model

# Create the model
input_shape = X_train_scaled.shape[1]
dl_model = create_model(input_shape)

# Define early stopping callback
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
```

```
D:\Anaconda\Lib\site-packages\keras\src\layers\core\dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

- Training

```python
# Train the deep learning model and capture the history
history = dl_model.fit(X_train, y_train, epochs=50, batch_size=32, validation_split=0.2)
```

```
Epoch 1/50
60/60 ━━━━━━━━━━━━━━━━━━━━ 2s 8ms/step - loss: 5808.6851 - mae: 40.0828 - val_loss: 4154.1323 - val_mae: 34.2361
Epoch 2/50
60/60 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 5877.7144 - mae: 37.7031 - val_loss: 4094.0129 - val_mae: 33.7551
Epoch 3/50
60/60 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 4910.3149 - mae: 35.9178 - val_loss: 4017.8870 - val_mae: 31.9372
Epoch 4/50
60/60 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 4715.8267 - mae: 33.9758 - val_loss: 3982.9270 - val_mae: 30.8333
Epoch 5/50
60/60 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 4106.5132 - mae: 31.8459 - val_loss: 3960.4275 - val_mae: 30.3946
Epoch 6/50
60/60 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 6007.9951 - mae: 38.4035 - val_loss: 3960.1997 - val_mae: 28.6906
Epoch 7/50
60/60 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 4660.6074 - mae: 33.7291 - val_loss: 3942.9551 - val_mae: 28.6556
Epoch 8/50
60/60 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 5461.2764 - mae: 34.7901 - val_loss: 3940.5090 - val_mae: 28.0625
Epoch 9/50
60/60 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 4516.1509 - mae: 32.0690 - val_loss: 3911.7393 - val_mae: 31.2400
Epoch 10/50
60/60 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 4966.3462 - mae: 35.7385 - val_loss: 3955.0908 - val_mae: 26.8387
Epoch 11/50
60/60 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 5967.5356 - mae: 36.6637 - val_loss: 3881.6309 - val_mae: 30.6890
Epoch 12/50
60/60 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 5116.2417 - mae: 34.3140 - val_loss: 3872.9141 - val_mae: 31.0446
```

- Metrics

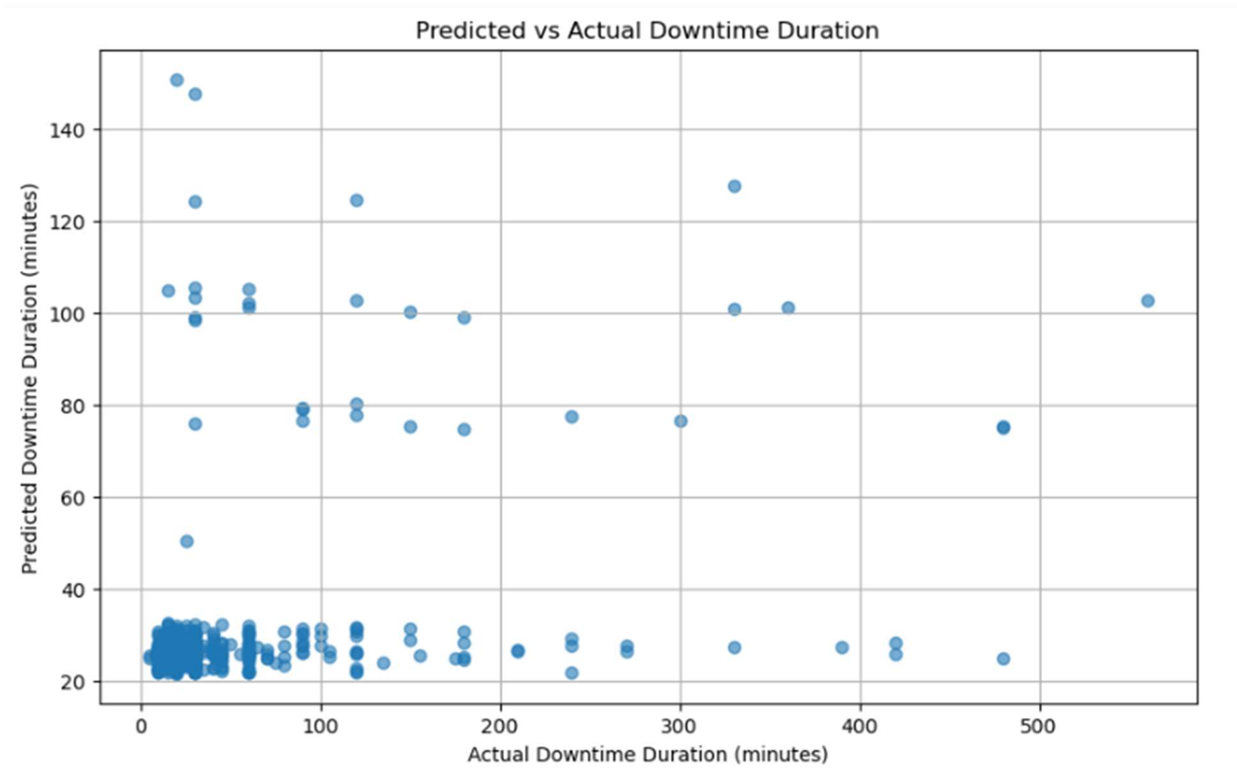
```python
# Predict downtime duration using the trained deep learning model
y_pred_dl = dl_model.predict(X_test_scaled)
```

```
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
```

```python
# Evaluate the deep learning model
mse_dl = mean_squared_error(y_test, y_pred_dl)
print("Mean Squared Error (Deep Learning):", mse_dl)
```

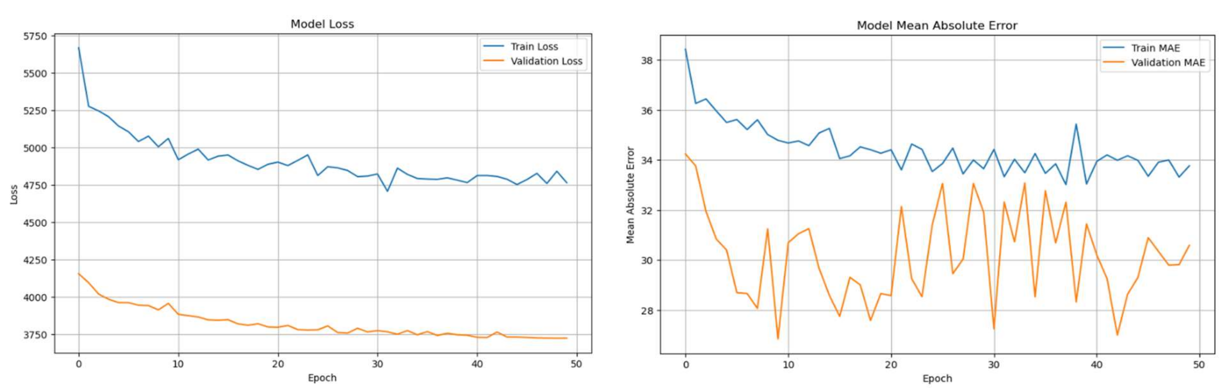
```
Mean Squared Error (Deep Learning): 3947.7153708105457
```

## 9. Cross Validation (Predicted Vs Actual)

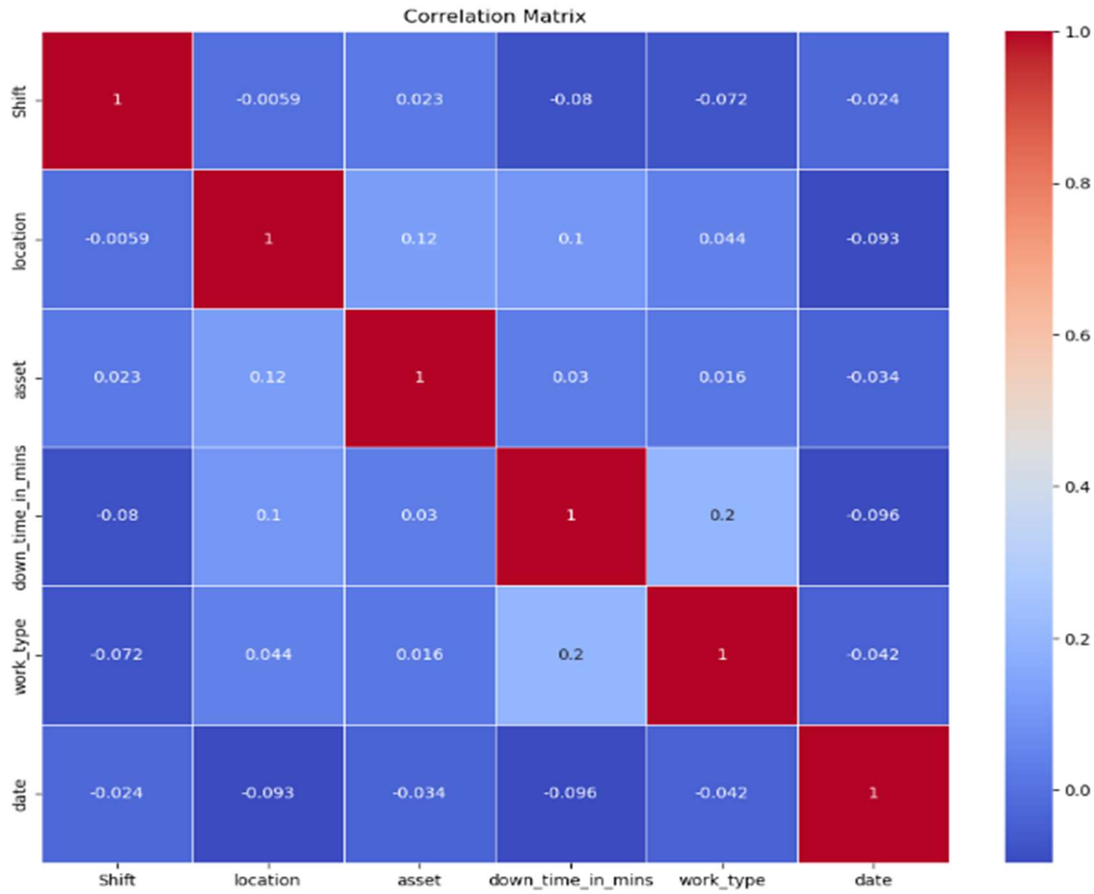



- The overlapped blue dots in the above plot shows the correctly predicted values by the model and the rest of the other outlying values are the wrongly predicted values by the deep learning model.

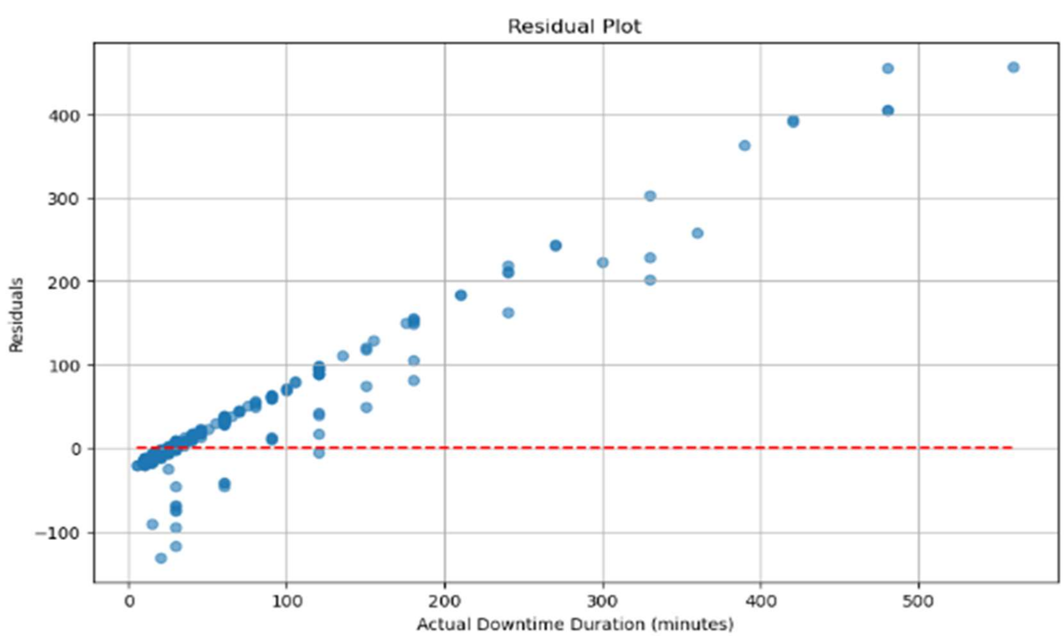
## 10. Plotting of Losses (Training Parameters)



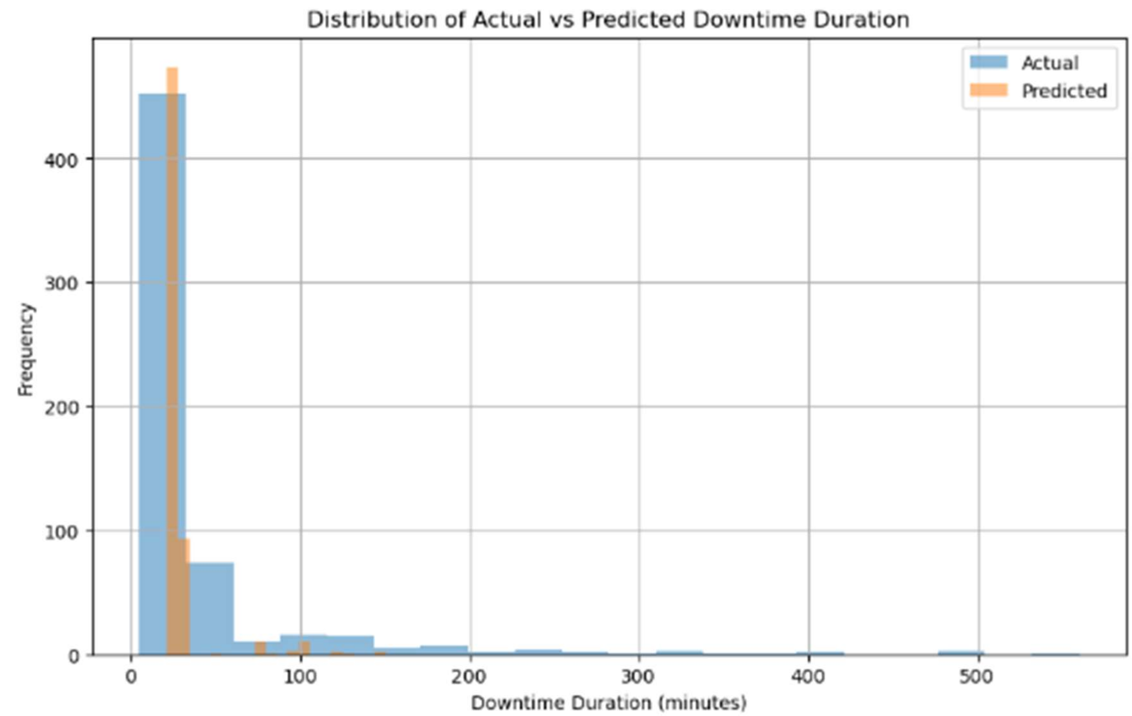
## 11. Correlation Matrix (Correlation Between Features)



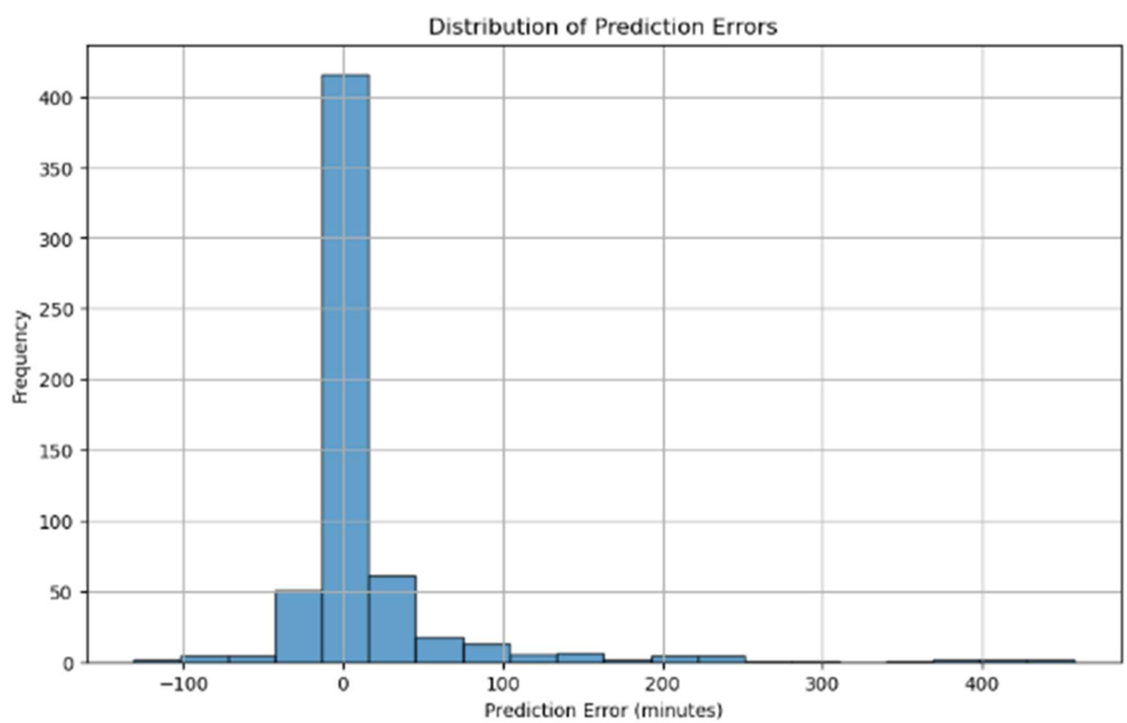

## 12. Residual Plot



## 12. Prediction Distribution Plot




## 12. Model Error Distribution Plot



## 13. Findings

- Maintenance Trends: The analysis revealed specific periods with higher maintenance activities, indicating potential peaks in maintenance demand.
- Predictive Insights: Certain asset types, locations, and work types were identified as key indicators that could predict maintenance needs.

## 14. Recommendations

Based on the analysis, the following recommendations are made to improve maintenance efficiency and effectiveness:

1. Enhanced Scheduling: Implement predictive maintenance schedules based on identified patterns to optimize maintenance timing.
2. Resource Allocation: Allocate resources more efficiently during peak maintenance periods to ensure quick response times.
3. Preventive Measures: Focus on assets and locations with higher downtime for preventive measures to reduce the frequency and impact of maintenance issues.

---

This report summarizes the key aspects of the maintenance analysis for Danfoss Power Solutions. The detailed findings and visualizations support the recommendations provided. By implementing these strategies, Danfoss Power Solutions can enhance its predictive maintenance approach and improve overall operational efficiency.