SHAHEED ZULFIKAR ALI BHUTTO
INSTITUTE OF SCIENCE AND TECHNOLOGY

Data Science

Machine Predication Maintenance

Project Report

Instructor:
Muhammad Ahsan
Dr. Imran Amin

Group:
Hammad Anwar [2012119]

Date:
1-3-2024

Contents

1. Executive Summary:

This executive summary provides a brief overview of the machine predictive maintenance report. The introduction emphasizes the revolutionary potential of predictive maintenance against traditional reactive methods in the dynamic industrial world. With a focus on the implementation of modern technologies like the Random Forest algorithm, the report aims to give stakeholders a thorough grasp of the concepts, procedures, and uses of machine predictive maintenance. Proactively addressing the problems associated with reactive maintenance, the research highlights the strategic advantages of predictive maintenance in terms of cost-effectiveness, resource allocation, and overall machinery reliability.

2. Introduction:

In the dynamic world of industrial operations, assuring the dependability and efficiency of machines is critical. This introduction provides the groundwork for exploring the novel idea of predictive maintenance, a strategy that goes beyond traditional reactive techniques. Predictive maintenance aims to identify and prevent problems before they have an impact on operations, as opposed to only addressing them when they happen. The use of advanced tools like the Random Forest model, which can be compared to an incredibly intelligent detective that evaluates vast amounts of data to identify the type of failure that a machine might be experiencing and whether it is occurring at all, is the fundamental component of this transformative approach. Finally, this could improve machine dependability and effectiveness by changing maintenance from reactive modifications to anticipating and preventing problems before they disrupt operations.

As we begin this journey, the narrative unfolds to show how predictive maintenance, powered by tools like the Random Forest model, has the ability to reshape the landscape of machinery care. It represents not simply a methodological change but also a fundamental shift in how we see industrial maintenance. We will discuss how these techniques can help ensure that machines run smoothly, prevent machine breakdowns proactively, and increase maintenance's role as a valuable resource in reaching operational excellence.

3. Objective:

The major goal of this research is to explain the fundamentals of machine predictive maintenance, as well as its capabilities and implications for industries. The report's goal is to give stakeholders the information they need to adopt successful predictive maintenance techniques and achieve operational excellence by offering a thorough overview.

4. Scope:

The basic ideas, practices, and applications of machine predictive maintenance are covered in this report. It includes talks about how to use advanced technologies, such machine learning models, to anticipate and stop possible machine failures. The wider effects of predictive maintenance in terms of cost-cutting, improving overall machinery reliability, and resource allocation are also covered in the research.

5. Literature Review:

In industrial settings, predictive maintenance has developed as a disruptive paradigm with the goal of revolutionizing existing reactive procedures by proactively identifying and treating potential machinery issues. Examining important research and developments in the field, this overview of the literature focuses on predictive maintenance techniques and the use of machine learning models—most particularly the Random Forest model.

- **Predictive Maintenance Strategies:** It makes more sense to anticipate when a machine might break rather than waiting for it to happen and then fixing it. According to research by Li, Zhang, and Wang et al., data can be used to anticipate problems by monitoring machine performance and making necessary adjustments on an as-needed basis rather than merely according to the schedule.
- **Machine Learning in Predictive Maintenance:** With its ability to provide a data-driven method of decision-making, machine learning models have become essential to predictive maintenance. Notably, the ensemble learning algorithm Random Forest has been effective in managing complex datasets and forecasting failures of equipment. Studies conducted by Sun et al. (2019) and Jha et al. (2020) demonstrate the effective use of Random Forest in various industrial contexts.
- **Real-World Applications:** Predictive maintenance has numerous practical uses, as demonstrated by studies like Ramesh et al. (2020), which highlight how it can lower maintenance costs, increase dependability, and improve overall operational efficiency.

When these results are combined, it is clear that predictive maintenance, especially when using the Random Forest model, has enormous potential to change the way industrial machinery is managed. The knowledge gained from these studies shows how industries are dealing with machinery challenges. This will help them achieve increased efficiency, reduced downtime, and long-term operational success.

6. Problem Statement:

The key problem in machine predictive maintenance is transitioning from previous reactive approaches to proactive strategies. The critical requirement is to accurately predict and recognize future failures. Ensuring model correctness, bridging the gap between predicted insights, and navigating data analysis difficulties are all necessary to meet this challenge. The ultimate goal is to increase machine dependability and operating efficiency by combining predictive maintenance approaches in a seamless manner, increasing efficiency, reducing downtime, and achieving long-term operational excellence.

7. Methodology

The methodology includes the use of the Random Forest model for predictive maintenance. Preprocessing the data, training the model, and validation are all part of this. The approach incorporates findings from the literature review, as well as methodologies that have proven effective in real-world applications. This ensures that the predictive model is dependable and efficient in predicting machine failures by comparing its performance to historical data.

- **Data Collection:** We have taken the machine predictive maintenance dataset from Kaggle.
- **Data Cleaning**: By deleting null values, unneeded columns, and altering the data types of some columns, we ensured that the data we had was clean and organized.

- **Data Understanding:** In order to identify patterns that can help us predict when a machine might experience problems, we thoroughly analyzed the data to determine how various things are related.
- **Data Visualization:** To have a better understanding of the data, we have used several plots and charts such as histograms, pie charts, scatter plots, heatmaps, and box plots.
- Model Selection: Based on the features of the dataset we chose the Random Forest model for predicting machine failure. Its capacity for group learning and track record of successfully managing complicated datasets matched with our requirements.
- **Model Training:** We have used the Random Forest model to forecast machine issues during the model training phase. The dataset was divided into training and testing sets. After that, the model was trained using the training set, which helped it discover relationships and patterns among different features as well as the probability of machine failures.
- **Evaluation Metrics:** The evaluation phase focused on assessing the model's performance using specific metrics. Accuracy, precision, recall and F1 score were employed followed by the importance of the features.
- **Predictive Analysis:** Predictive analysis was executed using a Random Forest model to anticipate machine failures and gauge their likelihood. Users input air temperature, process temperature, rotational speed, torque, and tool wear minutes. The model then predicted failure occurrence and computed probabilities for each failure type.

8. Conclusion:

In conclusion, this predictive maintenance project offers valuable insights into enhancing industrial machinery reliability. Through meticulous data analysis, model training, and evaluation, we have developed a proactive framework capable of anticipating potential machine failures. The model, tuned to recognize intricate patterns within the dataset, demonstrates proficiency in predicting issues before they escalate. The comprehensive evaluation metrics assure the model's reliability, with considerations for accuracy, precision, recall, and F1 score providing a nuanced understanding of its strengths. The practical recommendations derived from the insights aim to streamline operational procedures and prevent costly downtimes. By fostering a proactive approach to maintenance, we pave the way for improved machinery dependability, reduced operational disruptions, and substantial cost savings

9. Appendices

Data Cleaning:

```
df.isnull()
```

	UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target	Failure Type
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...
10025	False	False	False	False	False	False	False	False	False	False
10026	False	False	False	False	False	False	False	False	False	False
10027	False	False	False	False	False	False	False	False	False	False
10028	False	False	False	False	False	False	False	False	False	False
10029	False	False	False	False	False	False	False	False	False	False

10030 rows × 10 columns

```
df.isnull().values.any()
```

```
True
```

```
cleandf = df.dropna()
```

```
cleandf.isnull().values.any()
```

```
False
```

```
cleandf.isna().values.any()
```

```
False
```

Data Understanding:

```
cleandf.shape
```

```
(10000, 10)
```

```
cleandf.info()
```

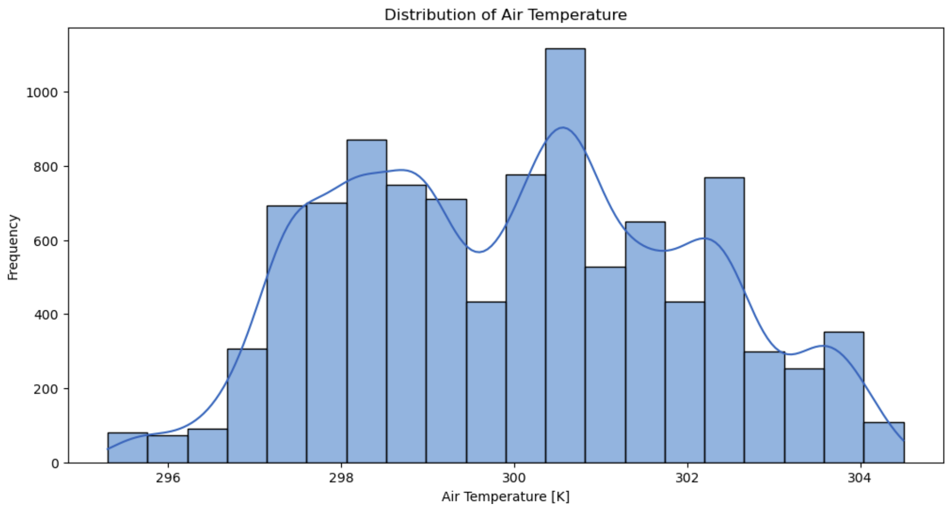
```
<class 'pandas.core.frame.DataFrame'>
Index: 10000 entries, 0 to 10029
Data columns (total 10 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   UDI                      10000 non-null  float64
 1   Product ID               10000 non-null  object
 2   Type                     10000 non-null  object
 3   Air temperature [K]      10000 non-null  float64
 4   Process temperature [K]  10000 non-null  float64
 5   Rotational speed [rpm]   10000 non-null  float64
 6   Torque [Nm]              10000 non-null  float64
 7   Tool wear [min]          10000 non-null  float64
 8   Target                   10000 non-null  float64
 9   Failure Type             10000 non-null  object
dtypes: float64(7), object(3)
memory usage: 859.4+ KB
```

```
cleandf.describe()
```

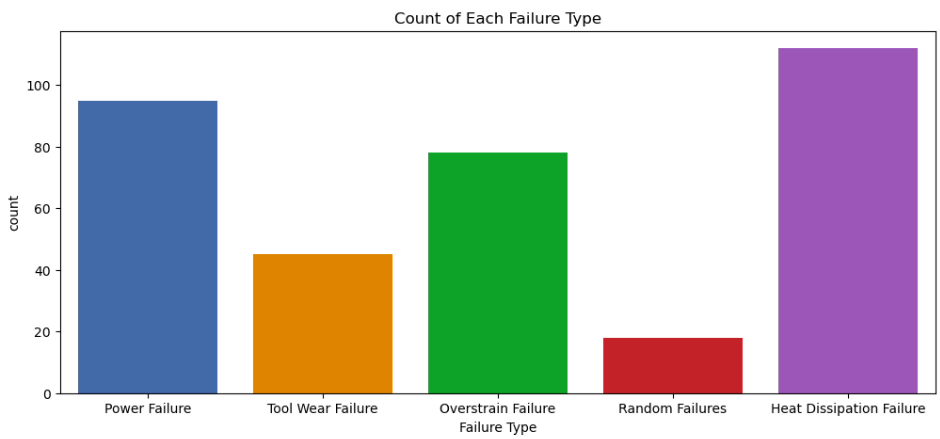
	UDI	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target
count	10000.00000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	5000.50000	300.004930	310.005560	1538.776100	39.986910	107.951000	0.033900
std	2886.89568	2.000259	1.483734	179.284096	9.968934	63.654147	0.180981
min	1.00000	295.300000	305.700000	1168.000000	3.800000	0.000000	0.000000
25%	2500.75000	298.300000	308.800000	1423.000000	33.200000	53.000000	0.000000
50%	5000.50000	300.100000	310.100000	1503.000000	40.100000	108.000000	0.000000
75%	7500.25000	301.500000	311.100000	1612.000000	46.800000	162.000000	0.000000
max	10000.00000	304.500000	313.800000	2886.000000	76.600000	253.000000	1.000000

Data Visualization

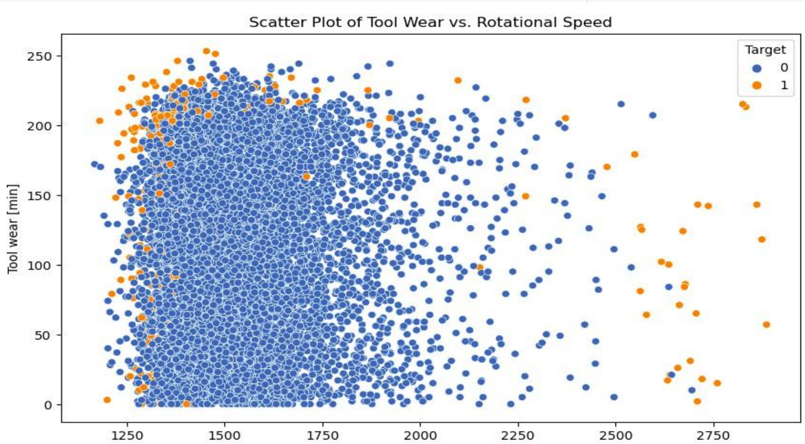
```
[ ] # Histogram for a numerical variable
    plt.figure(figsize=(12, 6))
    sns.histplot(cleandf['Air temperature [K]'], bins=20, kde=True)
    plt.title("Distribution of Air Temperature")
    plt.xlabel("Air Temperature [K]")
    plt.ylabel("Frequency")
    plt.show()
```




```
[ ] # Creating a count plot of failure types
    plt.figure(figsize=(12, 5))
    sns.countplot(x='Failure Type',  data=cleandf[cleandf['Failure Type'] != 'No Failure'])
    plt.title("Count of Each Failure Type")
    plt.show()
```




```
# Scatter Plot of Tool Wear vs. Rotational Speed
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Rotational speed [rpm]', y='Tool wear [min]', data=cleandf, hue='Target')
plt.title("Scatter Plot of Tool Wear vs. Rotational Speed")
plt.show()
```



Machine Learning Model:

```
[ ]  # For Modling we use first Random Forest Classifier
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import accuracy_score, classification_report

     X = cleandf.drop(['Target', 'Failure Type', 'Product ID', 'Type'], axis=1)  # Features (excluding target variables)
     y = cleandf['Target']  # Target variable

     # Assuming X contains your features and y contains your target variable
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

     # Initialize the Random Forest classifier
     rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

     # Train the model
     rf_classifier.fit(X_train, y_train)
```

```
        RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
# Make predictions
y_pred = rf_classifier.predict(X_test)

# Evaluate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")

# Classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.98
Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      1939
           1       0.82      0.59      0.69        61

    accuracy                           0.98      2000
   macro avg       0.90      0.79      0.84      2000
weighted avg       0.98      0.98      0.98      2000
```

```
# In this we show which column has more importance in accuracy
feature_importances = rf_classifier.feature_importances_

feature_importance_dict = dict(zip(X.columns, feature_importances))

sorted_features = sorted(feature_importance_dict.items(), key=lambda x: x[1], reverse=True)

for feature, importance in sorted_features:
    print(f"{feature}: {importance:.4f}")
```

```
Torque [Nm]: 0.3210
Rotational speed [rpm]: 0.2147
Air temperature [K]: 0.1616
Tool wear [min]: 0.1539
Process temperature [K]: 0.1488
```

Predictive Analysis:

```
[ ] failure_types = ['No Failure', 'Failure']

    # Input for new data point
    new_data_point = {'Air temperature [K]': float(input('Enter Air temperature [K]: ')),
                      'Process temperature [K]': float(input('Enter Process temperature [K]: ')),
                      'Rotational speed [rpm]': float(input('Enter Rotational speed [rpm]: ')),
                      'Torque [Nm]': float(input('Enter Torque [Nm]: ')),
                      'Tool wear [min]': float(input('Enter Tool wear [min]: '))}

    # Create a DataFrame from the new data point
    new_data_df = pd.DataFrame([new_data_point])

    # Use the model to make predictions
    predicted_failure = rf_classifier.predict(new_data_df)[0]

    # Use the model to predict the probabilities
    failure_probabilities = rf_classifier.predict_proba(new_data_df)[0]

    # Print the prediction and probabilities
    print(f'Predicted Failure: {failure_types[predicted_failure]}')
    print(f'Chances of Failure: {failure_probabilities[1]*100:.2f}%')

    # Print the predicted failure type and probabilities for each type
    print('Probabilities of Each Target:')
    for i, probability in enumerate(failure_probabilities):
        print(f'{failure_types[i]}: {probability * 100:.2f}%')
```

```
Enter Air temperature [K]: 297.5
Enter Process temperature [K]: 308.3
Enter Rotational speed [rpm]: 2564
Enter Torque [Nm]: 12.8
Enter Tool wear [min]: 127
Predicted Failure: Failure
Chances of Failure: 92.00%
Probabilities of Each Target:
No Failure: 8.00%
Failure: 92.00%
```