INTERNSHIP

PROJECT REPORT

FEYNN LABS

PREDICTIVE MAINTENANCE IN MANUFACTURING INDUSTRY

(Using Machine Learning)

By

1] SUYOG PAWAR
2] PABBATHI AKSHAY KUMAR
3] SOURAV RAKSHIT
4] GAYATRI PADMANI

Product/Service Business & Financial Modeling

Problem Statement

Predictive maintenance in manufacturing. This project is to find the early fault detection and diagnosis in the manufacturing industries and industry Manufacturers increasingly collect data from machines in their factories and products and with that observed data we can detect warning signs of expensive failures before they occur. Generally this technology is used in large scale companies but it is possible to use in medium and small scale companies too.

The problem statement attempted in this study may be divided into the following specific objectives.

- Assess the health state of the equipment (State of Health – SoH): Prediction of whether equipment is in its' last 'n' periods of life.
- Predicting the remaining useful life (RUL) of equipment. Prediction of number of cycles remaining before the equipment completely breaks down.
- Outlier Analysis: Finding the anomalies in the data.
- Pareto Alerts: analysis identifies the "vital few" features that contribute the most to plant maintenance and distinguishes them from the "trivial many".

Business Need Assessment

Predictive maintenance is a transformative application of the AI, Machine learning, IIoT with tremendous advantages. Below, we explore five benefits that can serve as differentiators for your organization:

Decreased downtime

Predictive maintenance enables technicians to detect issues in advance and resolve problems before equipment failure can occur, so you can:

- Cut unplanned downtime by as much as up to 30%Schedule multiple service procedures at one time
- Avoid the risk of reputation-damaging outages
- Reduce costly truck rolls required by unexpected downtime

Greater worker productivity

There is no need to disrupt worker productivity for an unexpected malfunction or breakdown. Predictive maintenance plans around workers' schedules, and:

- Enables up to 83% faster service time-to-resolution
- Maximizes uptime and prevents productivity lags
- Increases asset utilization

Reduced field service costs

By anticipating machine maintenance, service departments can generate major cost-savings and increased ROI through:

- Reduction of costly service truck rolls
- Increased first-time fix rates
- Streamlined maintenance costs through reduced labor, equipment, and inventory costs

Improved Product Design

Harnessing the power of IIoT data collected through your machine's sensors, product designers can use this vital information to:

- Extend asset lifespans
- Improve equipment durability and reliability
- Build more efficient machines in the future

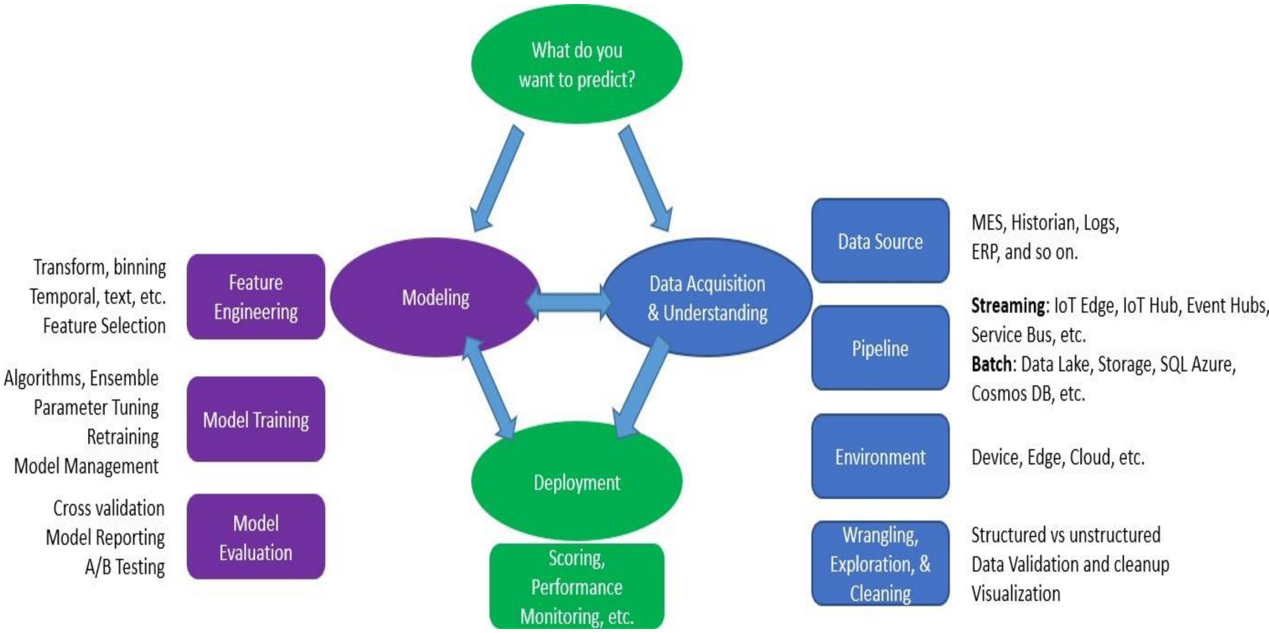
Improved worker safety

An unexpected breakdown or malfunction can lead to hazardous working conditions for your employees. By predicting when a malfunction may occur, you can ensure:

- Employees are nowhere near a machine when it breaks down
- Technicians can carry out service before a machine becomes dangerous

Target Specification and Characteristics

To build a PdM solution, we start with data. Ideally the data shows normal operation and the state of the equipment before, during, and after failures. The data comes from sensors, notes maintained by equipment operators, run information, environmental data, machine specifications, and so on. Systems of record can include historians, manufacturing execution systems, enterprise resource planning (ERP), and so on. The data is made available for analytics in a variety of ways. The following diagram illustrates The Team Data Science Process (TDSP). The process is customized for manufacturing and does an excellent job of explaining the various concerns that one has when building and executing machine learning models.



Your first task is to identify the types of failures you want to predict. With that in mind, you then identify the data sources that have relevant data about that failure type. The pipeline gets the data into the system from your environment. The data scientists use their favorite machine learning tools to prepare the data. At this point, they're ready to create and train models that can identify diverse types of issues. The models answer questions like:

- For the asset, what's the probability that a failure occurs within the next X hours?
- What's the remaining useful life of the asset?
- Is this asset behaving in an unusual way?
- Which asset requires servicing most urgently?

Benchmarking Alternate Product (Comparison with existing product/services)

Predictive maintenance is primarily used to detect upcoming system failures and prevent them using appropriate corrective measures. Using machine learning with predictive maintenance, we can analyze a massive volume of data and detect all possible failures that may lead to various financial and business losses. There are several predictive maintenance applications with machine learning, including manufacturing plants, power plants, railways, aviation, oil & gas industries, logistics & transportation, etc.

- **Manufacturing and IoT:** Predictive maintenance is widely used in manufacturing industries to supervise the production procedure through timely detection of faults and eliminate them before they malfunction using IoT. Hence, it increases the overall efficiency of the manufacturing process.
- **Automotive and Vehicles:** Various technologies connect vehicles to the sensor already enabled in the vehicle by manufacturers or dealers. These sensors collect all information and produce a massive amount of data directly retrieved by manufacturers or dealers, who warn us about any possible failure and corrective measures to prevent them before any malfunction.
- **Utility Suppliers:** Predictive maintenance techniques help utility suppliers to performbetter internal work, such as predicting early traits of supply, demand issues, outage issues, etc.
- **Insurance:** Various banking and financial institutions use predictive maintenance techniques to predict accurate analytics on disastrous weather conditions.

Applicable Patents

S. Matzka, "Explainable Artificial Intelligence for Predictive Maintenance Applications," 2020 Third International Conference on Artificial Intelligence for Industries (AI4I), 2020, pp. 69-74, doi: 10.1109/AI4I49448.2020.00023.

Since real predictive maintenance datasets are generally difficult to obtain and in particular difficult to publish, we present and provide a synthetic dataset that reflects real predictive maintenance data encountered in industry to the best of our knowledge and experience.

Applicable Regulations (Govt and environmental regulations)

"Predictive Maintenance Market by Component (Solutions and Services), Deployment Mode, Organization Size, Vertical (Government and Defense, Manufacturing, Energy and Utilities, Transportation and Logistics), and Region - Global Forecast to 2024"

The global predictive maintenance market size is forecasted to grow from USD 3.0 billion in 2019 to USD 10.7 billion by 2024.

The major growth drivers of the predictive maintenance market include the increasing use of emerging technologies to gain valuable insights. A lack of a skilled workforce may restrain the growth of the predictive maintenance market.

The services segment to grow at a higher CAGR during the forecast period

The predictive maintenance market is segmented on the basis of components, such as solutions and services. The services segment is expected to grow at a rapid pace during the forecast period. An efficient predictive maintenance service helps organizations develop a connected environment by integrating a predictive maintenance solution with their existing IT infrastructure.

The energy and utility segment to grow at the highest CAGR during the forecast period

The predictive maintenance market by vertical has been segmented into government and defense, manufacturing, energy and utilities, transportation and logistics, health and life sciences and others (agriculture, telecom, media, and retail). The energy and utility segment to grow at the highest CAGR during the forecast period, due to the growing demand for automated power-usage analytics applications.

Among regions, Asia Pacific (APAC) to grow at the highest CAGR during the forecast period

APAC is expected to grow at the highest CAGR during the forecast period. The increasing investments by the tech companies in major APAC countries, such as China, and Japan, increasing government regulations and initiatives are expected to drive the growth of the market in APAC.

Market Dynamics

Drivers

- Increasing Use of Emerging Technologies to Gain Valuable Insights
- Growing Need to Reduce Maintenance Cost and Downtime

Restraints

- Lack of Skilled Workforce

Opportunities

- Real-Time Condition Monitoring to Assist in Taking Prompt Actions

Challenges

- Companies' Concern Over Data Security and Privacy Issues
- Frequent Maintenance and Upgradation Requirement to Keep Systems Updated

Regulatory Implications

- General Data Protection Regulation
- Health Insurance Portability and Accountability Act
- Federal Trade Commission
- Federal Communications Commission
- Iso/IEC Standards
- ISO 55000 Standards
- ISO 13374 on Condition Monitoring and Diagnostics of Machines
- ISO/IEC JTC 1
- ISO/IEC JTC 1/SC 42
- ISO/IEC JTC1/SC3 1
- ISO/IEC JTC1/SC2 7
- Industrial Internet Consortium Reference Architecture
- CEN/ISO
- CEN/Cenelec
- National Institute of Standards and Technology
- Eprivacy
- ANSI Tappi Tip 0305-34:2008
- Mimosa

Companies Mentioned

- Asystom
- C3 IoT
- Dingo
- Ecolibrium Energy
- Fiix
- GE
- Hitachi
- IBM
- Microsoft

- OPEX Group
- PTC
- SAP
- SAS
- Schneider Electric
- Sigma Industrial Precision
- Software AG
- Softweb Solutions
- TIBCO
- Uptake

https://www.businesswire.com/news/home/20190627005383/en/Predictive-Maintenance-Market-2024---Focus-on-Government-and-Defense-Manufacturing-Energy-and-Utilities-Transportation-and-Logistics---ResearchAndMarkets.com

Applicable Constraints

Implementing a modern predictive maintenance strategy is no small undertaking.

- **State-of-the-art connectivity via sensors or "smart" equipment:** Connectivity and data communication are the backbone of predictive maintenance. By collecting and monitoring data about intricate aspects of equipment processes — small changes in vibration, temperature and equipment sounds — a system is able to determine that maintenance is required, well before the equipment fails. This data is collected via aftermarket sensors or sometimes, for newer equipment, from sensors built-in to the equipment. The data must also be transmitted to a central repository where it is monitored and analyzed for actionable insights.
- **Data security: Information security should be a primary concern for any "connected" organization:** meaning — in this day and age — any organization doing business featuring any digital aspect. In terms of predictive maintenance, it is critical to guarantee that equipment performance data is not subject to access by outside parties and that outside parties are not able to control predictive maintenance systems. At a more baseline level, it also remains important to protect information such as customer data.
- **Integration:** As mentioned above, data may be collected through aftermarket sensors or built-in, cutting-edge equipment. This data must then be communicated to a central system — often a CMMS (computerized maintenance management system), but sometimes an ERP (enterprise resource planning) system. In any scenario, it is crucial that all hardware, software and equipment can "talk" to one another effectively, and that communication is occurring as quickly as possible

(in near real-time). Implementing predictive maintenance hardware and software is only a first step in the process. Integration is the step that assures everything works together as it should.

- **Higher upfront costs and investment justification:** It is no secret that predictive maintenance involves higher upfront investments than standard preventive maintenance. All of the sensors, equipment and software described above incur costs, as well as the personnel to make sense of the data and drive decisions based on that information. As you will see below, this higher investment yields major benefits short and long term.

Although the challenges discussed in the above points are well worth it for the improved productivity, efficiency, reliability and ROI.

Business Model

Predictive maintenance strategies can help determine the condition of equipment in order to predict when maintenance should be performed. This approach to maintenance can ultimately lead to cost savings over routine preventive maintenance, because maintenance is only performed when warranted.

For those in manufacturing, you know that when machinery breaks, downtime and costly repairs ensue. By implementing predictive maintenance strategies, you can maintain equipment before it breaks down saving on downtime and maintenance costs. This enables optimal asset maintenance and further improves a plant's throughput, efficiency, quality, and safety.

Smart predictive maintenance is a modern maintenance technique that leverages multiple technologies and maintenance approaches, including predictive maintenance. Smart predictive maintenance goes beyond traditional preventive and predictive maintenance in three ways:

- It monitors a network of connected assets
- It automates some maintenance tasks
- It integrates with other maintenance management systems (CMMS, ERP, MES)

Five steps to reach smart predictive maintenance

Although technology can transform your maintenance program, best-in-class maintenance programs aren't generally built in a day. Here's how you can enable smart predictive maintenance by following the steps outlined below:

Step 1: **Start small with a pilot**

A pilot should generally take about three to four weeks on one or two critical assets. This initial effort will include sensor implementation and data streaming connections, as well as initial performance visualization dashboards.

Step 2: **Asset health monitoring**

It takes time to collect performance data, so patience is key. You'll want this information, as well as any asset failure data in order to generate better predictions.

Step 3: **Optimize failure thresholds**

Once data can be reliably connected remotely and an asset has provided enough failure data, the failure thresholds can be optimized.

Step 4: **Leverage data science**

Then, a data scientist can create predictive models, along with machine learning technology to update algorithms—increasing predictive capabilities with each failure until unplanned downtime can be avoided.

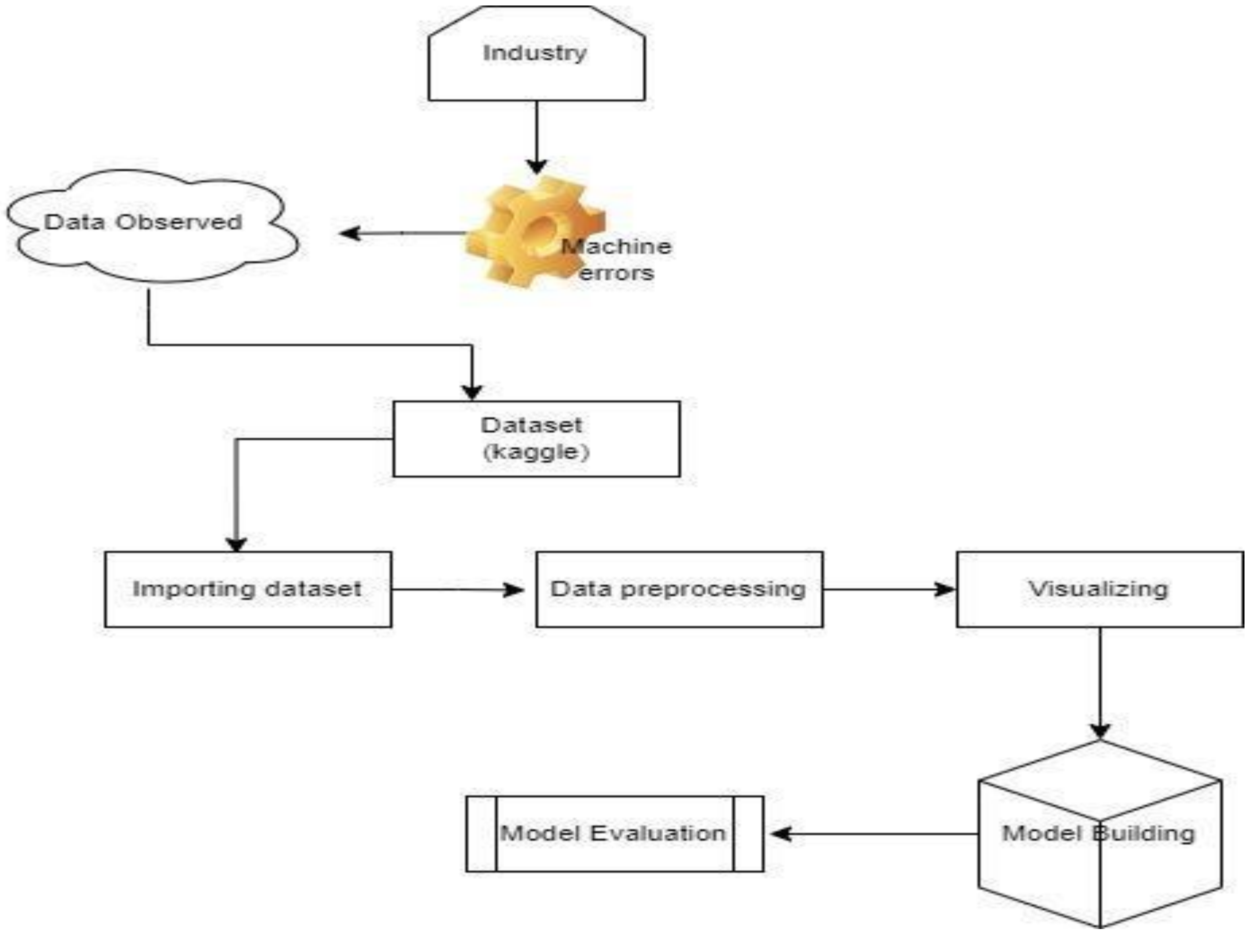
Step 5: **Reaching smart predictive maintenance**

For those with long-term vision, achieving steps 1-4 can lead you to smart predictive maintenance, which can help your business maintain a competitive edge. https://www.ge.com/digital/blog/5-steps-reaching-smart-predictive-maintenance

Predictive Maintenance for Manufacturing Industry Market : Segment Analysis

Predictive Maintenance For Manufacturing Industry Market is Segmented on the basis of Deployment, Verticals, and Geography.

Final Product (abstract) with Schematic Diagram



Financial Model

The Return on Investment (ROI) model is a prominent financial model for predictive maintenance in manufacturing. The financial benefits of implementing predictive maintenance are calculated by comparing the costs of implementation to the financial benefits of averting costly failures.

The ROI model is defined as follows:

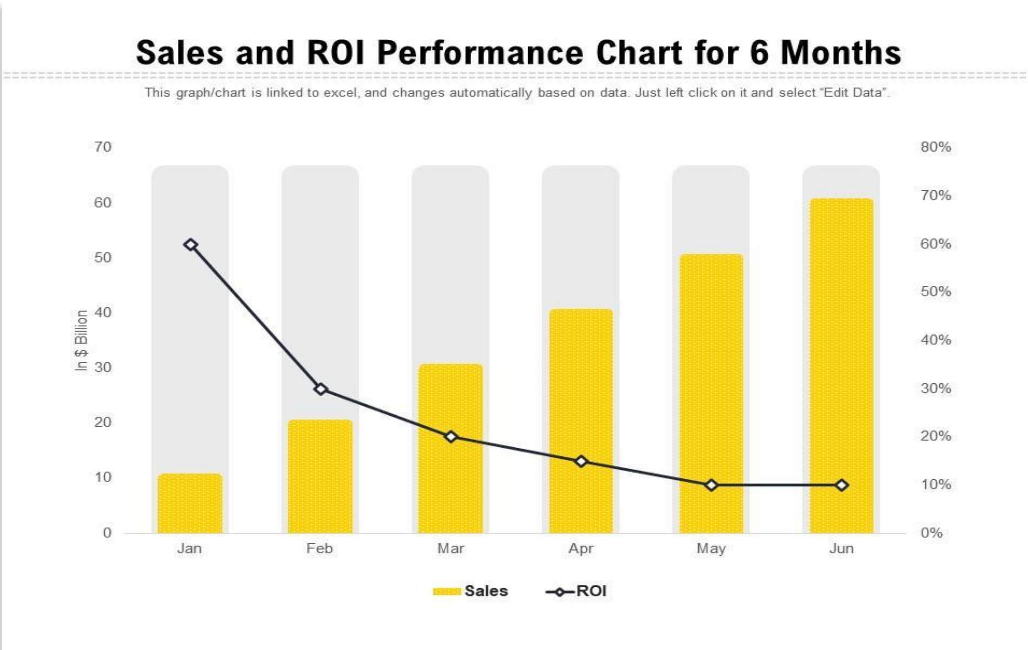
(Financial Benefits - Implementation Costs) / Implementation Costs = ROI

Where:

- Financial Advantages: the amount of money saved by avoiding costly failures as a result of early defect identification and diagnosis. This can include lower maintenance expenses, less downtime, and higher product quality.
- Implementation costs: the costs of installing predictive maintenance, such as sensors, software, and human training.

The ROI model can be used to determine whether predictive maintenance is a reasonable investment. A positive

ROI demonstrates that the financial benefits of implementing predictive maintenance outweigh the costs, indicating that the investment is beneficial. A negative ROI shows that the expenses of implementing predictive maintenance outweigh the financial advantages, rendering the investment unprofitable.
It should be noted that the financial benefits of predictive maintenance can be difficult to calculate precisely because they depend on a range of factors such as the type of equipment, the nature of the manufacturing process, and the cost of downtime. When determining the ROI of predictive maintenance, all of these elements must be carefully considered.



ROI = (Net Profit / Cost of Investment) x 100
Where:

- Net Profit = Total Revenue - Total Expenses
- Cost of Investment = Total Cost of the Project

Product Details

Algorithms, Softwares Used:

Python

Machine learning SVM(support vector machines)

Pandas, Matplotlib, Seaborn, Sklearn

Data Source - Kaggle

Code Implementation/Validation

IMPORTING LIBRARIES & READING DATASET

In [1]:

```python
import pandas as pd
data = pd.read_csv('predictive_maintenance (1).csv')
data
```

Out[1]:

	UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target	Failure Type
0	1	M14860	M	298.1	308.6	1551	42.8	0	0	No Failure
1	2	L47181	L	298.2	308.7	1408	46.3	3	0	No Failure
2	3	L47182	L	298.1	308.5	1498	49.4	5	0	No Failure
3	4	L47183	L	298.2	308.6	1433	39.5	7	0	No Failure
4	5	L47184	L	298.2	308.7	1408	40.0	9	0	No Failure
...	...	...	...	...	...	...	...	...	...	...
9995	9996	M24855	M	298.8	308.4	1604	29.5	14	0	No Failure
9996	9997	H39410	H	298.9	308.4	1632	31.8	17	0	No Failure
9997	9998	M24857	M	299.0	308.6	1645	33.4	22	0	No Failure
9998	9999	H39412	H	299.0	308.7	1408	48.5	25	0	No Failure
9999	10000	M24859	M	299.0	308.7	1500	40.2	30	0	No Failure

10000 rows × 10 columns

PERFORMING DATA PREPROCESSING

In [2]:

```python
data.columns
```

Out[2]:

```
Index(['UDI', 'Product ID', 'Type', 'Air temperature [K]',
       'Process temperature [K]', 'Rotational speed [rpm]', 'Torque [Nm]',
       'Tool wear [min]', 'Target', 'Failure Type'],
      dtype='object')
```

In [3]:

```python
data.isnull().sum()
```

Out[3]:

```
UDI                        0
Product ID                 0
Type                       0
Air temperature [K]        0
Process temperature [K]    0
Rotational speed [rpm]     0
Torque [Nm]                0
Tool wear [min]            0
Target                     0
Failure Type               0
dtype: int64
```

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   UDI                      10000 non-null  int64
 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  int64
 6   Torque [Nm]              10000 non-null  float64
 7   Tool wear [min]          10000 non-null  int64
 8   Target                   10000 non-null  int64
 9   Failure Type             10000 non-null  object
dtypes: float64(3), int64(4), object(3)
memory usage: 781.4+ KB
```

In [5]:
```
data.describe().T
```

Out[5]:

	count	mean	std	min	25%	50%	75%	max
UDI	10000.0	5000.50000	2886.895680	1.0	2500.75	5000.5	7500.25	10000.0
Air temperature [K]	10000.0	300.00493	2.000259	295.3	298.30	300.1	301.50	304.5
Process temperature [K]	10000.0	310.00556	1.483734	305.7	308.80	310.1	311.10	313.8
Rotational speed [rpm]	10000.0	1538.77610	179.284096	1168.0	1423.00	1503.0	1612.00	2886.0
Torque [Nm]	10000.0	39.98691	9.968934	3.8	33.20	40.1	46.80	76.6
Tool wear [min]	10000.0	107.95100	63.654147	0.0	53.00	108.0	162.00	253.0
Target	10000.0	0.03390	0.180981	0.0	0.00	0.0	0.00	1.0

In [6]:
```
df = data.groupby(['Failure Type', 'Type'])
df.first()
```

Out[6]:

		UDI	Product ID	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target
Failure Type	**Type**								
Heat Dissipation Failure	**H**	3830	H33243	302.3	310.9	1366	48.4	130	1
	L	3761	L50940	302.3	310.9	1377	46.8	166	1
	M	3237	M18096	300.8	309.4	1342	62.4	113	1
No Failure	**H**	11	H29424	298.4	308.9	1782	23.9	24	0
	L	2	L47181	298.2	308.7	1408	46.3	3	0
	M	1	M14860	298.1	308.6	1551	42.8	0	0
Overstrain Failure	**H**	5400	H34813	302.8	312.4	1411	53.8	246	1
	L	161	L47340	298.4	308.2	1282	60.7	216	1
	M	3936	M18795	302.6	311.6	1227	68.2	187	1
Power Failure	**H**	1124	H30537	296.6	307.7	1386	62.3	100	1
	L	51	L47230	298.9	309.1	2861	4.6	143	1
	M	195	M15054	298.2	308.5	2678	10.7	86	1
Random Failures	**H**	1749	H31162	298.4	307.7	1626	31.1	166	0
	L	1303	L48482	298.6	309.8	1505	45.7	144	0
	M	1222	M16081	297.0	308.3	1399	46.4	132	0
Tool Wear Failure	**H**	1088	H30501	296.9	307.8	1549	35.8	206	1
	L	78	L47257	298.8	308.9	1455	41.3	208	1
	M	1997	M16856	298.4	308.0	1416	38.2	198	1

In [7]:
```
print(data['Type'].unique())
print(data['Failure Type'].unique())
```

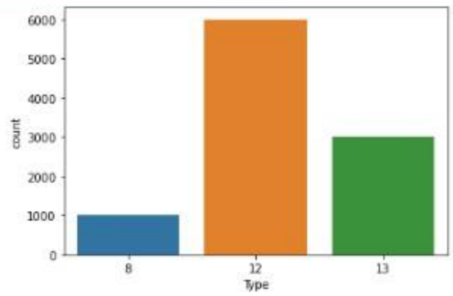


```
['M' 'L' 'H']
['No Failure' 'Power Failure' 'Tool Wear Failure' 'Overstrain Failure'
 'Random Failures' 'Heat Dissipation Failure']
```

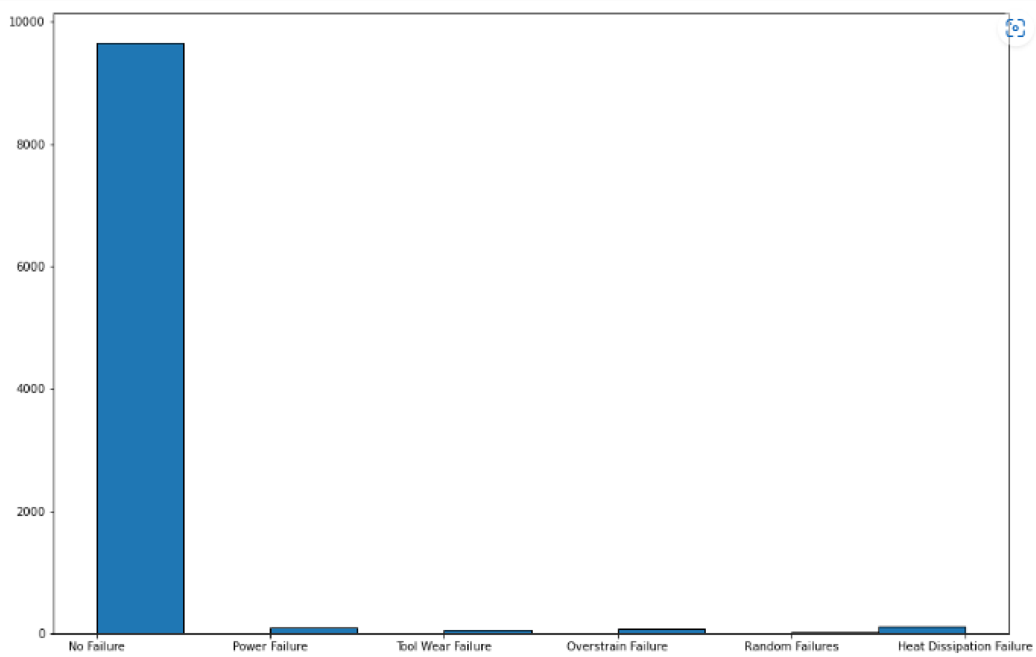
VISUALIZING THE DATASET

In [39]:
```
sns.countplot(data['Type'])
plt.show()
```

```
C:\Users\MY-PC\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

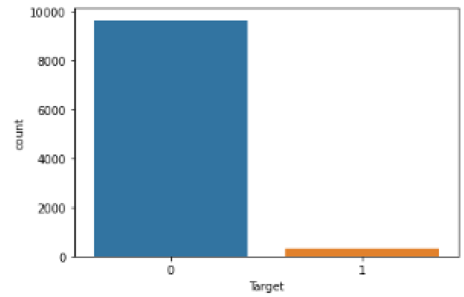


In [8]:
```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize = (15,10))
plt.hist(x = data['Failure Type'], ec = 'black')
plt.show()
```

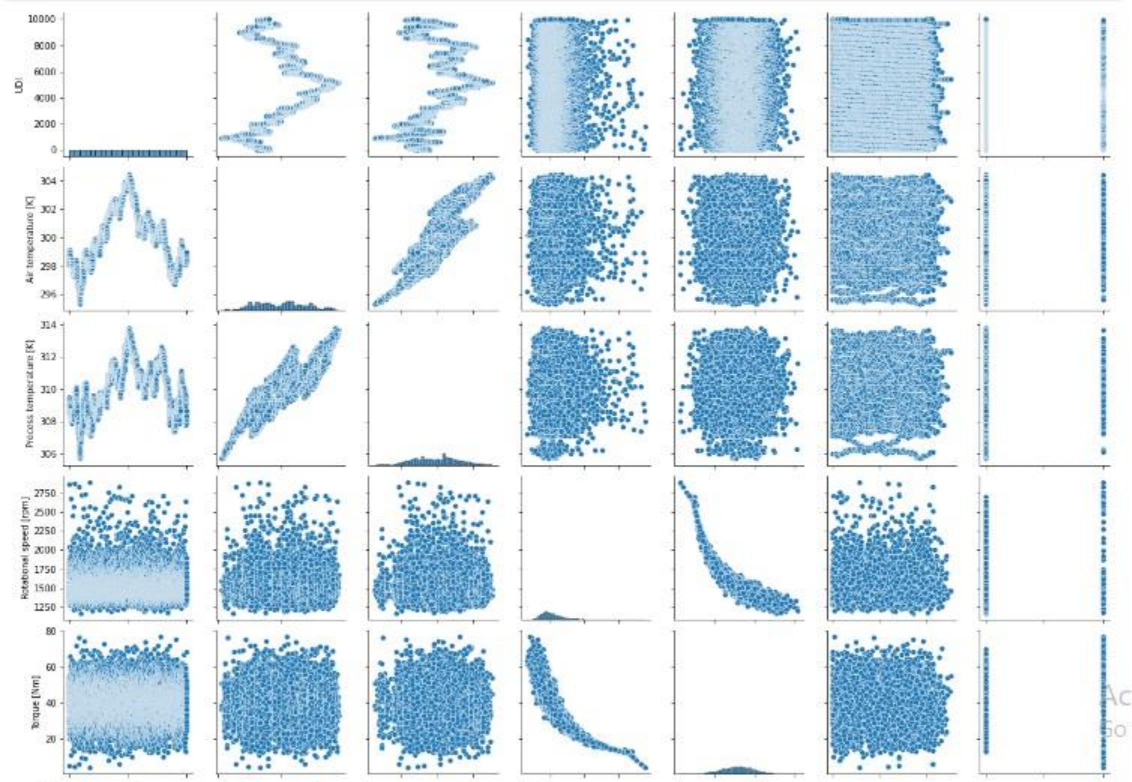


In [9]:
```
sns.countplot(data['Target'])
plt.show()
```

```
C:\Users\MY-PC\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```



In [10]:
```
sns.pairplot(data)
plt.show()
```



In [15]:

```python
print('Target vs Air temperature [K] : ', data['Target'].corr(data['Air temperature [K]'])),
print('Target vs Process temperature [K] : ', data['Target'].corr(data['Process temperature [K]'])),
print('Target vs Rotational speed [rpm] : ', data['Target'].corr(data['Rotational speed [rpm]'])),
print('Target vs Torque [Nm] : ', data['Target'].corr(data['Torque [Nm]']))
```

```
Target vs Air temperature [K] :  0.08255568978323986
Target vs Process temperature [K] :  0.03594597332977692
Target vs Rotational speed [rpm] :  -0.0441875597343754
Target vs Torque [Nm] :  0.19132077505949352
```

In [43]:

```python
data['Failure Type'].replace(['No Failure','Power Failure','Tool Wear Failure','Overstrain Failure','Random Failures','Heat
```

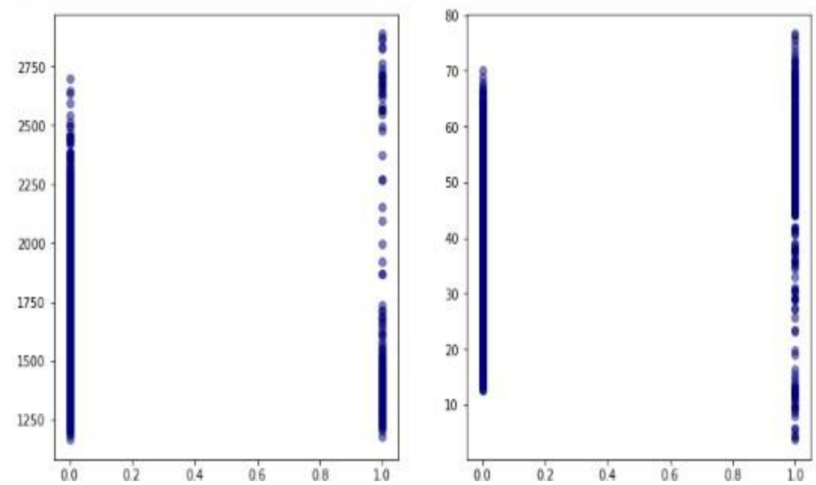
In [44]:

```python
print('Target vs Air temperature [K] : ', data['Failure Type'].corr(data['Air temperature [K]'])),
print('Target vs Process temperature [K] : ', data['Failure Type'].corr(data['Process temperature [K]'])),
print('Target vs Rotational speed [rpm] : ', data['Failure Type'].corr(data['Rotational speed [rpm]'])),
print('Target vs Torque [Nm] : ', data['Failure Type'].corr(data['Torque [Nm]']))
```

```
Target vs Air temperature [K] :  0.11848590323491176
Target vs Process temperature [K] :  0.05557937441930225
Target vs Rotational speed [rpm] :  -0.11967960602784054
Target vs Torque [Nm] :  0.1903461103885093
```

In [17]:

```python
plt.figure(figsize = (12, 6))
plt.subplot(1,2,1)
plt.scatter(data['Target'], data['Rotational speed [rpm]'], color = 'navy', alpha = 0.5)

plt.subplot(1,2,2)
plt.scatter(data['Target'], data['Torque [Nm]'], color = 'navy', alpha = 0.5)

plt.show()
```



BUILDING THE MODEL

In [22]:
```
target = data['Failure Type']
target
```

Out[22]:
```
0       No Failure
1       No Failure
2       No Failure
3       No Failure
4       No Failure
           ...
9995    No Failure
9996    No Failure
9997    No Failure
9998    No Failure
9999    No Failure
Name: Failure Type, Length: 10000, dtype: object
```

In [23]:
```
features = data.drop(['Failure Type', 'Product ID'], axis = 1)
features
```

Out[23]:

	UDI	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target
0	1	13	298.1	308.6	1551	42.8	0	0
1	2	12	298.2	308.7	1408	46.3	3	0
2	3	12	298.1	308.5	1498	49.4	5	0
3	4	12	298.2	308.6	1433	39.5	7	0
4	5	12	298.2	308.7	1408	40.0	9	0
...	...	...	...	...	...	...	...	...
9995	9996	13	298.8	308.4	1604	29.5	14	0
9996	9997	8	298.9	308.4	1632	31.8	17	0
9997	9998	13	299.0	308.6	1645	33.4	22	0
9998	9999	8	299.0	308.7	1408	48.5	25	0
9999	10000	13	299.0	308.7	1500	40.2	30	0

In [24]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=10)
```

In [25]:
```
from sklearn import svm

model = svm.LinearSVC()
model.fit(x_train, y_train)
```

```
C:\Users\MY-PC\anaconda3\lib\site-packages\sklearn\svm\_base.py:985: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
  warnings.warn("Liblinear failed to converge, increase "
```

Out[25]: LinearSVC()

In [26]:
```
model.score(x_train, y_train)
```

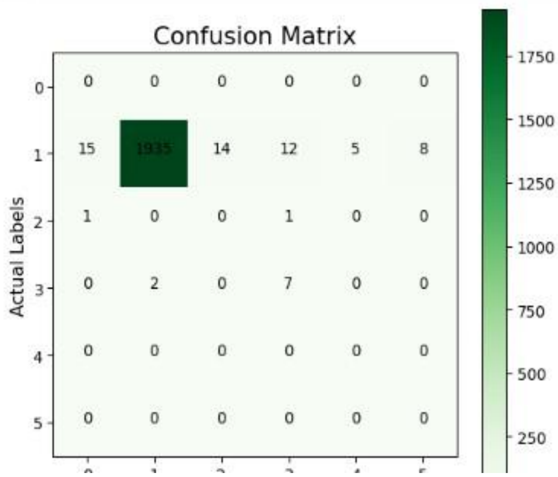
Out[26]: 0.966375

In [27]:
```
model.score(x_test, y_test)
```

Out[27]: 0.971

```
In [33]:
nr_rows = conf_mat.shape[0]
nr_cols = conf_mat.shape[1]
```

```
In [34]:
import itertools
plt.figure(figsize=(6,6), dpi=95)

plt.imshow(conf_mat, cmap = plt.cm.Greens)

plt.title('Confusion Matrix', fontsize=16)
plt.ylabel('Actual Labels', fontsize=12)
plt.xlabel('Predicted Labels', fontsize=12)

for i, j in itertools.product(range(nr_rows), range(nr_cols)):

    plt.text(j,i,conf_mat[i,j],horizontalalignment = 'center')

plt.colorbar()

plt.show()
```




```
In [35]:
# True Positives
import numpy as np
np.diag(conf_mat)

Out[35]: array([   0, 1935,    0,    7,    0,    0], dtype=int64)
```

```
In [36]:
recall = np.diag(conf_mat) / np.sum(conf_mat, axis=1)
recall

C:\Users\MY-PC\AppData\Local\Temp/ipykernel_7004/2268408136.py:1: RuntimeWarning: invalid value encountered in true_divide
  recall = np.diag(conf_mat) / np.sum(conf_mat, axis=1)

Out[36]: array([       nan, 0.97285068, 0.        , 0.77777778,        nan,
                nan])
```

```
In [37]:
precision = np.diag(conf_mat) / np.sum(conf_mat, axis=0)
precision

Out[37]: array([0.        , 0.99896748, 0.        , 0.35      , 0.        ,
       0.        ])
```

```
In [38]:
avg_recall = np.mean(recall)
print(avg_recall)
avg_precision = np.mean(precision)
print(avg_precision)

nan
0.22482791257959045
```

Github - https://github.com/akshaytestgithub/Feynn-Labs-Projects

Conclusion

Predictive maintenance presents you with the best time to work on an asset so that maintenance frequency is minimal and reliability is as high as possible while eliminating unnecessary costs. However, there are few disadvantages to predictive maintenance like high start-up costs and the need for specialized personnel.

Clearly, predictive maintenance is not apt for every company, especially those that have not yet implemented planned maintenance activities. However, larger organizations that have outgrown conventional maintenance practices and have additional budgets should leverage predictive maintenance. Predictive maintenance has been shown to result in a tenfold increase in ROI, 25%-30% reduction in maintenance costs, a 70%-75% decrease in breakdowns, and a 35%-45% reduction in downtime. These statistics are evidence of why predictive maintenance is gaining prominence quickly.

Thank you