Predicting Machine failure by Data Science

By

SAGOLESEM HEMAT (MECHANICAL ENGINEER)

1. Abstract:

Predictive maintenance is a type of maintenance that directly tracks an asset's health, status, and performance in real time. Predictive maintenance is aimed at reducing costly, unexpected breakdowns and offers the manufacturer the opportunity to plan maintenance around their own production schedule.

1.1 How does predictive maintenance works?

Through a collection of real time data collected through the Industrial Internet of things (IIoT)

Predictive maintenance continuously analyses the conditions of equipment during the normal operations to reduce the likelihood of unexpected machine failure.

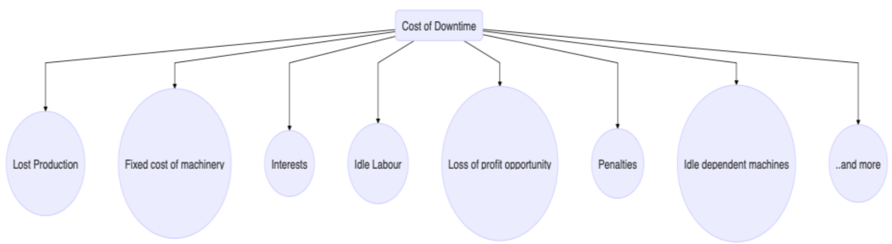
With predictive maintenance we can monitor and test various indicator such as slow bearing speed, temperature or lubrication. Using condition based monitoring and IIoT, these tools detect abnormalities during normal operations and send real time alerts to the machine owner that indicates the potential future failure.

2. Problem statement:

Despite rapid advances in technology, we're still seeing many factory failures and other types of maintenance accidents repeated. Production takes a direct hit because of equipment failures. A great deal of money is lost by the time production restarts. It also impacts OEMs and dealers in terms of lost reputation and business opportunity. Fortunately, these issue can be tackled to a major extent by using data science.

Cost of breakdown is not just the opportunity loss (of potential profit from production), but it also includes fixed cost of the machine. Further, delay in production can attract penalties and lost orders. At times, when other machines also depend on the failed machinery, the cost escalates through the roof. The cost of single breakdown can easily exceed thousands of dollars. The worst part is, this loss can hardly ever be recovered.

While mechanical and technical failures sometimes can't be prevented, most of the time they can be avoided by being spotted earlier enabling recall before a user-facing incident occurs.



3. **Market / Customer/ Business Need Assessment:**

Due to machine's mechanical and technical failures, company faces many issues. So by using this preventive maintenance technologies they had an advantage.

3.1 Decrease downtime:

- Cut unplanned downtime by as much as up to 30% schedule multiple procedure at one time
- Avoid risk of reputation damage outages.
- Reduce trucks roll by unexpected downtime.

3.2 Greater work productivity:

- Enables up to 83% faster service time to resolution.
- Maximize uptime and prevent productivity lags.
- Increase assets utilizations.

3.3 Improved product design:

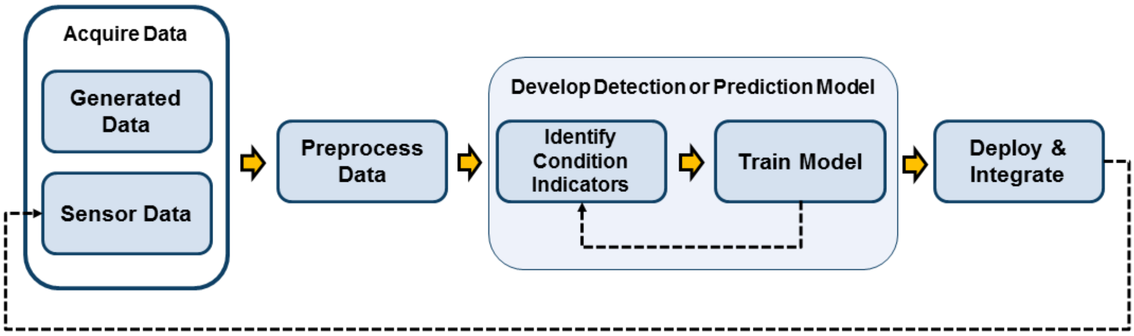
- Extends assets lifespan.
- Improved equipment durability and reliability.
- Build more efficient in future.

3.4 Improved worker safety:

- Employees are now where when machine got breakdown.
- Technician can carry out service before machine got break down.

4. Designing Algorithms for Predictive Maintenance:

Workflows for Algorithm Development-



Sensor data from machine on which algorithm is deployed

A predictive maintenance program uses condition monitoring and prognostics algorithms to analyse data measured from the system in operation. *Condition monitoring* uses data from a machine to assess its current condition and to detect and diagnose faults in the machine. Machine data is data such as temperature, pressure, voltage, noise, or vibration measurements, collected using dedicated sensors.

Prognostics is forecasting when a failure will happen based on the current and past state of the machine. A prognostics algorithm typically estimates the machine's *remaining useful life* (RUL) or time-to-failure by analysing the current state of the machine. Prognostics can use modelling, machine learning, or a combination of both to predict future values of condition indicators.

5. Application of Machine Learning :

5.1 Decision Models for Fault Detection and Diagnosis:

Condition monitoring includes discriminating between faulty and healthy states (fault detection). To design an algorithm for condition monitoring, you use condition indicators extracted from system data to train a decision model that can analyse indicators extracted from test data to determine the current system state.

5.2 Feature Selection:

Feature selection techniques help you reduce large data sets by eliminating features that are irrelevant to the analysis you are trying to perform.

- pca— Perform *principal component analysis*, which finds the linear combination of independent data variables that account for the greatest variation in observed values.
- sequentialfs — For a set of candidate features, identify the features that best distinguish between healthy and faulty conditions, by sequentially selecting features until there is no improvement in discrimination.
- fscnca — Perform feature selection for classification using neighbourhood component analysis.

5.3 Statistical Distribution Fitting:

When you have a table of condition indicator values and corresponding fault states, you can fit the values to a statistical distribution. Comparing validation or test data to the resulting distribution yields the likelihood that the validation or test data corresponds to one or the other fault states

- ksdensity — Estimate a probability density for sample data.
- `histfit` — Generate a histogram from data, and fit it to a normal distribution.
- `ztest` — Test likelihood that data comes from a normal distribution with specified mean and standard deviation.

5.4 Machine Learning:

Statistics and Machine Learning Toolbox includes many functions that you can use to train classifiers.

- `fitcsvm` — Train a binary classification model to distinguish between two states, such as the presence or absence of a fault condition
- `fitcecoc` — Train a classifier to distinguish among multiple states. This function reduces a multiclass classification problem to a set of binary classifiers.
- `fitctree` — Train a multiclass classification model by reducing the problem to a set of binary decision trees.
- `fitclinear` — Train a classifier using high-dimensional training data.

6. Code Implementation:

I'm using dataset from GitHub. Whole code can look up from the link: https://medium.com/swlh/machine-learning-for-equipment-failure-prediction-and-predictive-maintenance-pm-e72b1ce42da1

jupyter Predictive maintenance of machine failure Last Checkpoint: 4 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Markdown

Install all of the relevant Python Libraries

In [2]:
```
#!pip install imblearn --upgrade
#!pip install plotly --upgrade
#!pip install chart-studio --upgrade
```

Import Required Libraries

In [4]:
```
import chart_studio.plotly as py
import plotly.graph_objs as go
import plotly as plotly
import pandas as pd
import numpy as np
from imblearn.over_sampling import SMOTE
from imblearn.over_sampling import SMOTENC
from sklearn import metrics
from sklearn.preprocessing import LabelEncoder
import xgboost as xgb
from xgboost.sklearn import XGBClassifier
import types
import pandas as pd
def __iter__(self): return 0
```

I'm using dataset from GitHub

I'm using dataset from GitHub

In [34]:
```
pd_data_1 = pd.read_csv("Equipment_failure_data_1.csv", sep=",", header=0)
```

In [35]:
```
pd_data_2 = pd.read_csv("Equipment_failure_data_2.csv")
```

In [36]:
```
#concatenate the two data files into one dataframe
pd_data=pd.concat([pd_data_1, pd_data_2])
```

Data Exploration

In [38]:
```
pd_data.head(5)
```

Out[38]:

	ID	DATE	REGION_CLUSTER	MAINTENANCE_VENDOR	MANUFACTURER	WELL_GROUP	S15	S17	S13	S5	S16	S19	S
0	100001	12/2/14	G	O	Y	1	11.088000	145.223448	39.34	3501.0	8.426869	1.9	24.610
1	100001	12/3/14	G	O	Y	1	8.877943	187.573214	39.20	3489.0	6.483714	1.9	24.671
2	100001	12/4/14	G	O	Y	1	8.676444	148.363704	38.87	3459.0	6.159659	2.0	24.733
3	100001	12/5/14	G	O	Y	1	9.988338	133.660000	39.47	3513.0	9.320308	2.0	24.773
4	100001	12/6/14	G	O	Y	1	8.475264	197.181600	40.33	3589.0	8.022960	1.5	24.808

Create a new variable, FAILURE_TARGET. It is equal to 1 if the record proceeds a failure by "failure_window" days or less.

In [60]:
```
pd_data['FAILURE_TARGET'] = np.where(((pd_data.TIME_TO_FAILURE < target_window) & ((pd_data.TIME_TO_FAILURE>=0))), 1, 0)
tips_summed = pd_data.groupby(['FAILURE_TARGET'])['S5'].count()
tips_summed
```

Out[60]:
```
FAILURE_TARGET
0    296011
1     11740
Name: S5, dtype: int64
```

The new field occurs about 4% of the time.

In [61]:
```
pd_data['FAILURE_TARGET'].mean()
```

Out[61]:
```
0.03814772332177637
```

Create the Testing, Training and Validation Groupings

In [62]:
```
#Get a Unique List of All IDs
aa=pd_data
pd_id=aa.drop_duplicates(subset='ID')
pd_id=pd_id[['ID']]
pd_id.shape
```

Out[62]:
```
(421, 1)
```

7. Conclusion:

Here, we can propose to build a machine learning model for predictive maintenance. Since many industries are depends on PLC (programmable Logic Controller) for detecting equipment failure. This is a kind of preventive maintenance. So by applying machine learning model on predictive maintenance, we can save time, money for the company.

8. Reference:

- https://medium.com/swlh/machine-learning-for-equipment-failure-prediction-and-predictive-maintenance-pm-e72b1ce42da1
- **https://analyticsindiamag.com/how-to-predict-machine-failure-using-data-science/**
- **https://www.mathworks.com/help/predmaint/gs/designing-algorithms-for-condition-monitoring-and-predictive-maintenance.html#:~:text=A%20prognostics%20algorithm%20typically%20estimates,current%20state%20of%20the%20machine.**