CONTENTS

1. PREDICT ASSET FAILURE

1.1 Introduction

- Python is an interpreted language. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Artificial Intelligence is an approach to make a computer, a robot, or a product to think how smart human think. Artificial Intelligence is a study of how human brain think, learn, decide and work, when it tries to solve problems. And finally this study outputs intelligent software systems. The aim of Artificial Intelligence is to improve computer functions which are related to human knowledge, for example, reasoning, learning, and problem-solving.

The intelligence is intangible. It is composed of

- Reasoning
- Learning
- Problem Solving
- Perception
- Linguistic Intelligence

1.2 Objective Of Research

The main objective of asset failure prediction is used to determine the past working performance of the machine, predict future performance, and assess the capability of generating failure. The objective considers the machine's measures such as temperature, humidity and so on, based on these criteria's prediction is evolved.

It also includes the working condition of the motor whether they can spend more on their necessities or not. The output of this prediction is whether the motor condition becomes failure or not based on the available parameters.

1.3 Problem Statement

Motors working conditions depends on the climatic conditions. It has been witnessed that in many cases failure of the motors occur due to drastic change in weather conditions. Thus we have to take the climatic conditions into account.

So we predict the failure of the motor, based on the following factors.

- Temperature
- Humidity
- Measure 1
- Measure 2
- Measure 3
- Measure 4
- Measure 5

To solve these predictions we must use some algorithms. Here we use Logistic Regression algorithm for our prediction. Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

2. REVIEW OF LITERATURE

Electrical motor reliability and efficiency are directly influenced by the quality of the power supply. Power-quality disturbances such as transients, unbalance and harmonics can harm the windings and decrease motor performance. Voltage surges might induce equipment failure. The financial costs will be significant, particularly when downtime, production losses and premature equipment replacement or repairs are involved.

While surge protectors, voltage regulators and ride-through systems limit the effect of short-term power disturbances on your equipment, ongoing motor analysis and power-quality monitoring increase the chance of detecting degradation before damage is done.

Motor analysis revival: Monitoring motors in their operational state has long been a strategy for efficiency and performance management. Factoring in the influence of electrical supply and distribution extends the benefits of analysis. For instance, the rotor signature is used to determine a motor's load and efficiency, but power quality affects its accuracy. "Systems that take the signature of a motor rotor and use it as an analysis tool are most useful when they calculate efficiency while the equipment is online," says Dennis Bowns, executive director of Green Motors Practices Group. "Other alternatives require shutting the equipment down to perform a resistance test, and then bringing it back online.

3. DATA COLLECTION

Motors are known for their easy failure with the intervention of nature i.e, due to temperature and humidity. Temperature generally causes metals expansion and humidity is responsible for rusting of metals. Taking temperature and humidity into consideration, prediction of motor failure is determined. Here for asset failure prediction parameters of measures are taken with respect to temperature and humidity in order to predict the motor condition.

Temperature: Temperature is the main aspect that has a huge influence on the motor working condition. Metals expand on heating. It might also be a main aspect for the failure of the motor and other reasons may be optional in some cases.

Humidity: Major part of the motor is made of metal. Humidity is a formation of water vapour which has a greater impact on motor failure by rusting of metals.

Measure 1: It gives the first measure of the motor working.

Measure 2: It gives the second measure of the motor working.

Measure 3: It gives the third measure of the motor working.

Measure 4: It gives the fourth measure of the motor working.

Measure 5: It gives the fifth measure of the motor working.

4. METHODOLOGY

4.1 Exploratory Data Analysis

Exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modelling or hypothesis testing task. Exploratory data analysis was promoted by John Tukey to encourage statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. EDA is different from initial data analysis (IDA),[1] which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. EDA encompasses IDA.

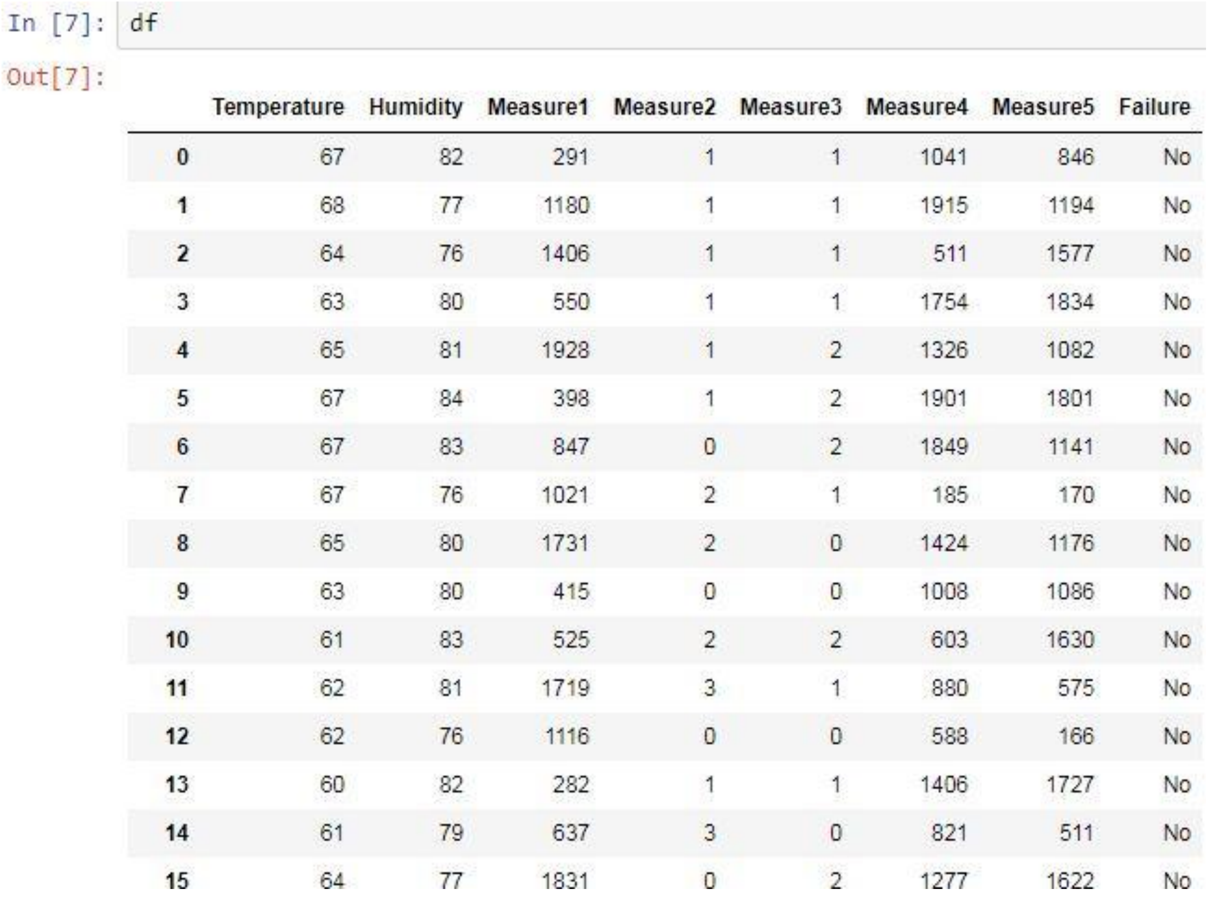
The objectives of EDA are to:

• Suggest hypotheses about the causes of observed phenomena • Assess assumptions on which statistical inference will be based

• Support the selection of appropriate statistical tools and techniques

• Provide a basis for further data collection through surveys or experiments

4.1.1 Figures and Tables

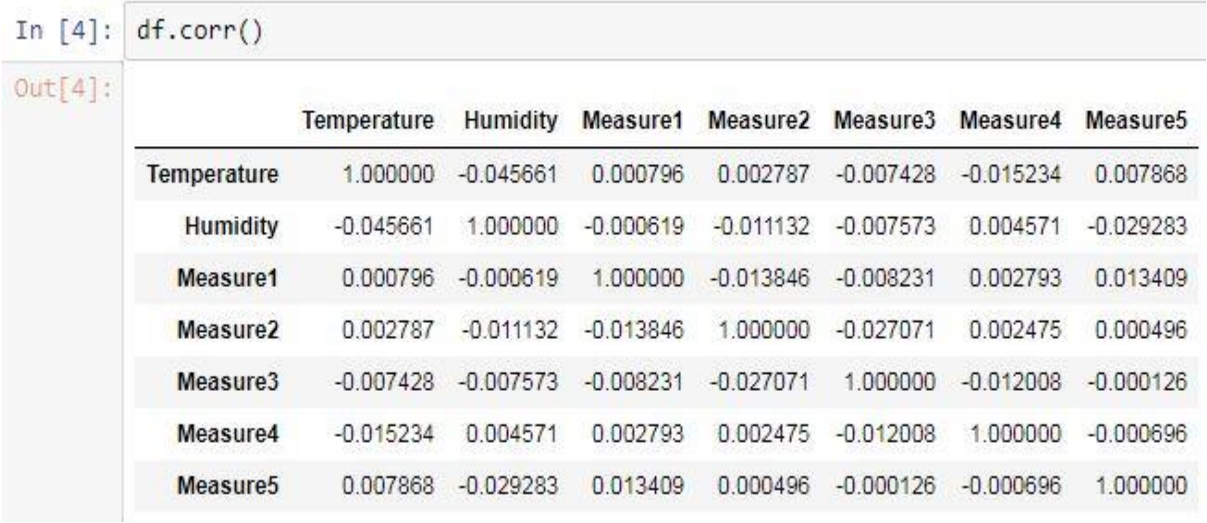


```
In [7]: df
```

Out[7]:

	Temperature	Humidity	Measure1	Measure2	Measure3	Measure4	Measure5	Failure
0	67	82	291	1	1	1041	846	No
1	68	77	1180	1	1	1915	1194	No
2	64	76	1406	1	1	511	1577	No
3	63	80	550	1	1	1754	1834	No
4	65	81	1928	1	2	1326	1082	No
5	67	84	398	1	2	1901	1801	No
6	67	83	847	0	2	1849	1141	No
7	67	76	1021	2	1	185	170	No
8	65	80	1731	2	0	1424	1176	No
9	63	80	415	0	0	1008	1086	No
10	61	83	525	2	2	603	1630	No
11	62	81	1719	3	1	880	575	No
12	62	76	1116	0	0	588	166	No
13	60	82	282	1	1	1406	1727	No
14	61	79	637	3	0	821	511	No
15	64	77	1831	0	2	1277	1622	No

Fig 4.1.1.1: Dataset of motor failure



```
In [4]: df.corr()
```

Out[4]:

	Temperature	Humidity	Measure1	Measure2	Measure3	Measure4	Measure5
Temperature	1.000000	-0.045661	0.000796	0.002787	-0.007428	-0.015234	0.007868
Humidity	-0.045661	1.000000	-0.000619	-0.011132	-0.007573	0.004571	-0.029283
Measure1	0.000796	-0.000619	1.000000	-0.013846	-0.008231	0.002793	0.013409
Measure2	0.002787	-0.011132	-0.013846	1.000000	-0.027071	0.002475	0.000496
Measure3	-0.007428	-0.007573	-0.008231	-0.027071	1.000000	-0.012008	-0.000126
Measure4	-0.015234	0.004571	0.002793	0.002475	-0.012008	1.000000	-0.000696
Measure5	0.007868	-0.029283	0.013409	0.000496	-0.000126	-0.000696	1.000000

Fig 4.1.1.2: Correlation table

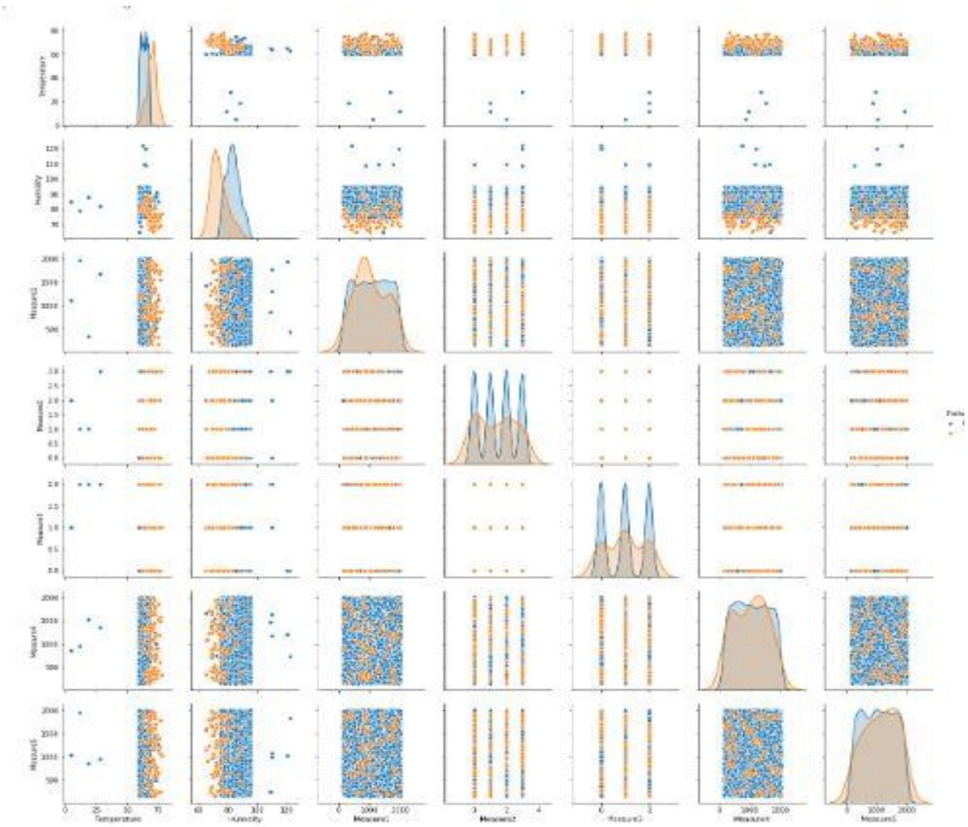
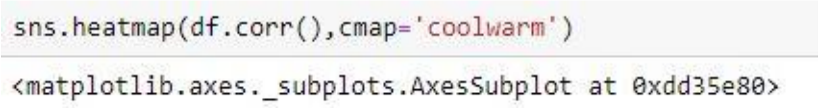


Fig 4.1.1.3: Plots against different parameters



```
sns.heatmap(df.corr(),cmap='coolwarm')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0xdd35e80>
```

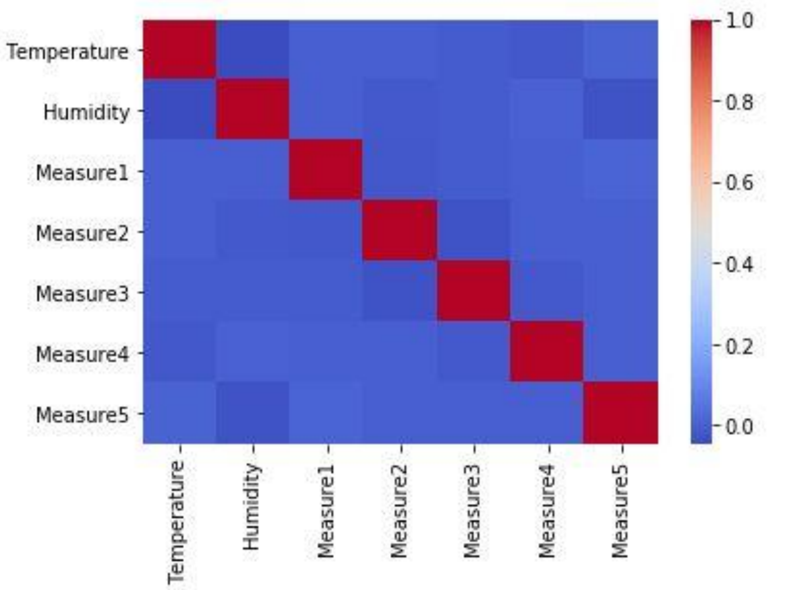


Fig 4.1.1.4: Heatmap for motor failure based on parameters

4.2. Statistical techniques and visualization

Statistics is a collection of tools that you can use to get answers to important questions about data. You can use descriptive statistical methods to transform raw observations into information that you can understand and share. You can use inferential statistical methods to reason from small samples of data to whole domains. Statistics is a pillar of machine learning.

- Data Understanding: Requires the use of summary statistics and data visualization.
- Data Cleaning. Requires the use of outlier detection, imputation and more.
- Data Selection. Requires the use of data sampling and feature selection methods.
- Data Preparation. Requires the use of data transforms, scaling, encoding and much more. Model Evaluation. Requires experimental design and resampling methods.
- Model Configuration. Requires the use of statistical hypothesis tests and estimation statistics. Model Selection. Requires the use of statistical hypothesis tests and estimation statistics. Model Presentation. Requires the use of estimation statistics such as confidence intervals. Model Predictions. Requires the use of estimation statistics such as prediction intervals.

NumPy is a commonly used Python data analysis package. By using NumPy, you can speed up your workflow, and interface with other packages in the Python ecosystem, like scikit-learn, that use NumPy under the hood. NumPy was originally developed in the mid 2000s, and arose from an even older package called Numeric. This longevity means that almost every data analysis or machine learning package for Python leverages NumPy in some way.

Numpy 2-Dimensional Arrays With NumPy, we work with multidimensional arrays. It dive into all of the possible types of multidimensional arrays later on, but for now, this project focus on 2-dimensional arrays. A 2-dimensional array is also known as a matrix, and is something you should be familiar with.

Creating A NumPy Array : We can create a NumPy array using the numpy.array function. If we pass in a list of lists, it will automatically create a NumPy array with the same number of rows and columns. In the below code, we:

- Import the NumPy package.
- Pass the list of lists wines into the array function, which converts it into a NumPy array.

- Exclude the header row with list slicing.
- Specify the keyword argument dtype to make sure each element is converted to a float.

Pandas is an open source python library that is built on top of NumPy. It allows you do fast analysis as well as data cleaning and preparation. Pandas is hands down one of the best libraries of python. It supports reading and writing excel spreadsheets, CVS's and a whole lot of manipulation. It is more like a mandatory library you need to know if you're dealing with datasets from excel files and CSV files. i.e for Machine learning and data science.

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib. Matplotlib was originally written by John D. Hunter, has an active development community, and is distributed under a BSD-style license. Matplotlib is designed to be as usable as MATLAB, with the ability to use Python, and the advantage of being free and open-source. Several toolkits are available which extend Matplotlib functionality. Some are separate downloads, others ship with the Matplotlib source code but have external dependencies.

- Basemap: map plotting with various map projections, coastlines, and political boundaries.
- Cartopy: a mapping library featuring object-oriented map projection definitions, and arbitrary point, line, polygon and image transformation capabilities. (Matplotlib v1.2 and above)
- Excel tools: utilities for exchanging data with Microsoft Excel • GTK tools: interface to the GTK+ library.

Visualization with Matplotlib - One of Matplotlib's most important features is its ability to play well with many operating systems and graphics backend. Matplotlib supports dozens of backends and output types, which means you can count on it to work regardless of which operating system you are using or which output format you wish. This cross-platform, everything-to-everyone approach has been one of the great strengths of Matplotlib. It has led to a large user base, which in turn has led to an active developer base and Matplotlib's powerful tools and ubiquity within the scientific Python world.

Importing Matplotlib - Just as we use the np shorthand for NumPy and the pd shorthand for Pandas, we will use some standard shorthands for Matplotlib imports.

Plotting from a script - If you are using Matplotlib from within a script, the function plt.show() is your friend. plt.show() starts an event loop, looks for all currently active figure objects, and opens one or more interactive windows that display your figure or figures.

The plt.show() command does a lot under the hood, as it must interact with your system's interactive graphical backend. The details of this operation can vary greatly from system to system and even installation to installation, but matplotlib does its best to hide all these details from you.

4.3 Data Modelling and visualization:

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways. Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range of nodes in the palette. Flows can be then deployed to the runtime in a single-click.

JavaScript functions can be created within the editor using a rich text editor.

A built-in library allows you to save useful functions, templates or flows for re-use.

The light-weight runtime is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.

With over 225,000 modules in Node's package repository, it is easy to extend the range of palette nodes to add new capabilities

The flows created in Node-RED are stored using JSON which can be easily imported and exported for sharing with others.

An online flow library allows you to share your best flows with the world.

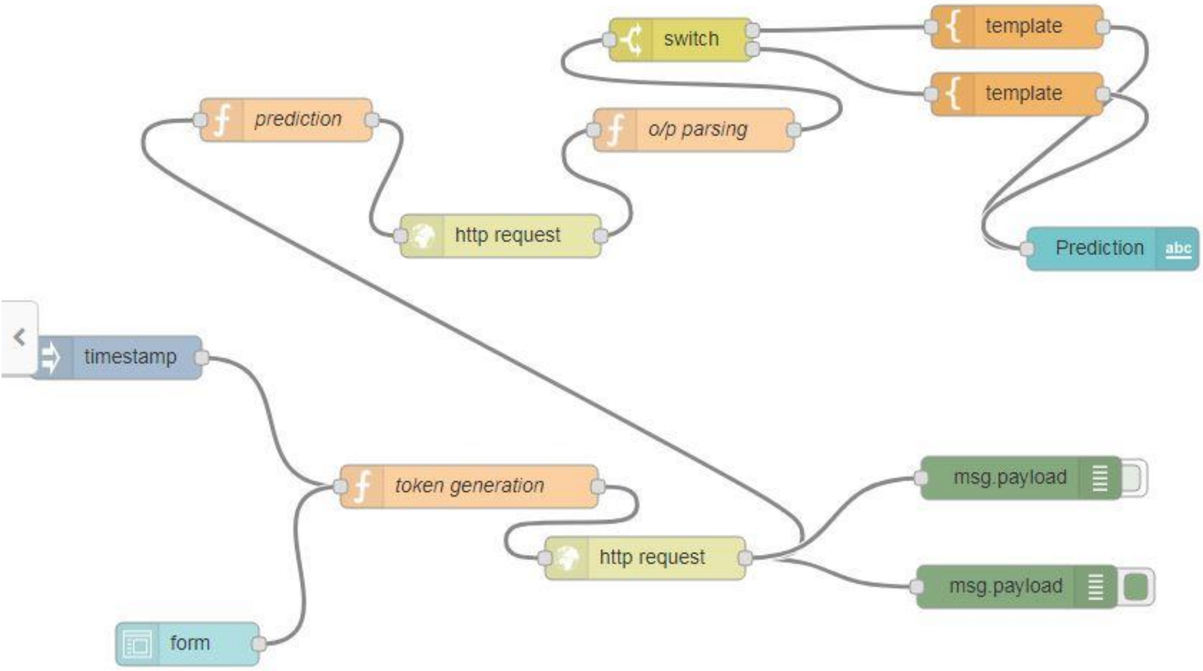
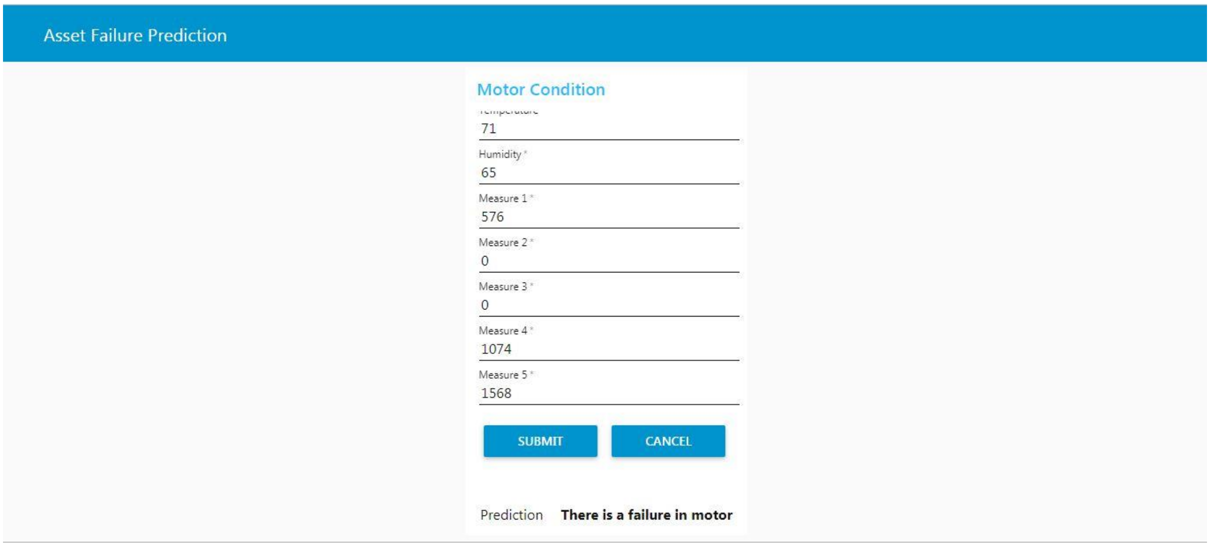


Fig: Flow modules using Node-RED



Asset Failure Prediction

Motor Condition

Temperature
71

Humidity *
65

Measure 1 *
576

Measure 2 *
0

Measure 3 *
0

Measure 4 *
1074

Measure 5 *
1568

Prediction **There is a failure in motor**

Fig: Asset failure prediction output

5.REFERENCES

https://www.kaggle.com/

https://www.analyticsvidhya.com/

https://www.r-project.org/

https://www.youtube.com/

https://www.wikipedia.org/

6.CONCLUSION

With the help of our work, we will be predicting the failure of the motor and analyzing the factors which strongly effect the failure. This prediction will give suggestions based on the result obtained which level of attribute can lead to a failure. This intimates the customer to respond on the motor's condition before anything goes wrong.