COMP 4980

Special Topics: Machine Learning

“Towards Predictive Maintenance: Machine Learning to Detect Failures in Mobile Assets”

November 14th 2023

Ethan Morissette
T00579850

Table of Contents

Data Description

There is a number of different columns that represent operating level/tempratures of specific components in this machinery. These columns will help forecast or predict the failure modes. The `Data Details` from OpenML indicate what levels for each failure mode are required. For example, `HDF` Heat Dissipation: requires a difference between air and process temperature to be below 8.6 [K] and tools rational speed below 1380 [RPM].

The five independent failure modes include tool wear failure (TWF), heat dissipation failure (HDF), power failure (PWF), overstrain failure (OSF), and random failures (RNF). Each failure mode is meticulously defined, providing insights into the conditions under which the manufacturing process may fail. Machine failure is present when one or more failure modes is true.

Each descriptive column provides values necessary to; Create new columns, separate and classify failures/machines, and predict.

All columns are numerical minus the `Product ID`, and ‘Type’ columns which contain L, M, or H. This indicates the likelihood a machine will fail. L being low, M medium, and H for high. This will be interesting after more analysis is done. It can help classify what failure mode is plaguing which asset quality. For this project I will be overlooking these columns.

Although synthetic this dataset does a fantastic job at mirroring real world data, making it a prime candidate for machine learning. It was constructed to emulate authentic industrial scenarios. Building a predictive model that is both accurate and general to the entire dataset may actually prove to be affective in a real world setting.

With 10’000 rows and 14 different columns, it has enough information to be both informative and interesting.

Final Project ML - Analysis.ipynb

Tabular stats take away:

Stat	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]
COUNT	10000	10000	10000	10000	10000
MEAN	300.00493	310.00556	1538.7761	39.98691	107.951
STD	2.000258683	1.483734219	179.2840959	9.968933725	63.65414664
MIN	295.3	305.7	1168	3.8	0
->25%	298.3	308.8	1423	33.2	53
->50%	300.1	310.1	1503	40.1	108
->75%	301.5	311.1	1612	46.8	162
MAX	304.5	313.8	2886	76.6	253

Output from df.describe()

This information may not immediately offer deep insights, but it holds significance for our analysis. Calculating the maximum and minimum values is essential as these extremes can influence the types of failures we anticipate encountering.

Furthermore, examining the mean values is equally as critical. They provide insights into the typical or central values of the various sensor data, helping us understand the base line behavior of our systems.

It's important to note the highs and lows of each feature column as those are the levels at which machines will see failure. More often at a high, like "High Process Temp" or "High Torque". This is where critical components will fail and cause other components to break or fail.

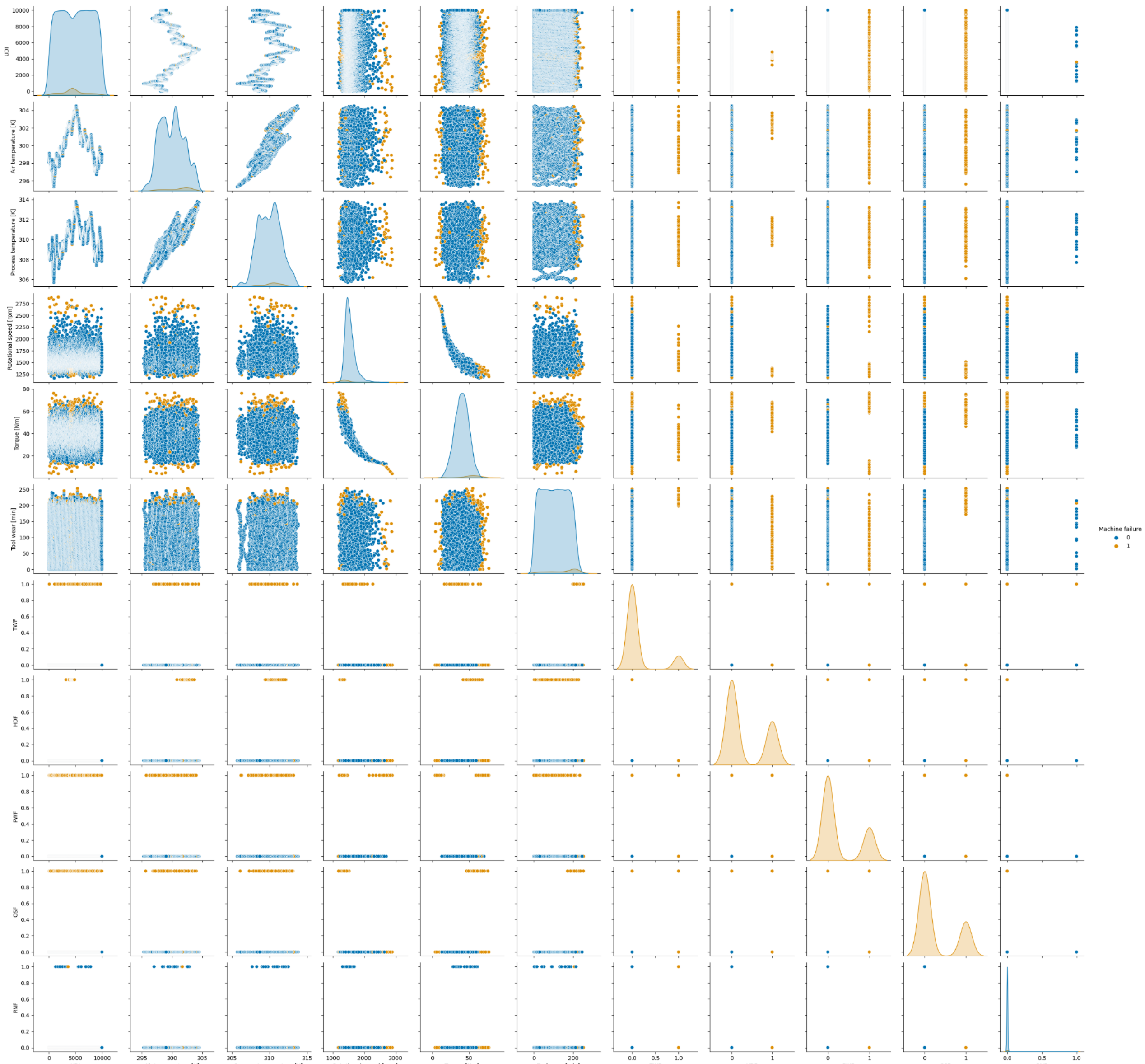
Stat	Machine failure	TWF	HDF	PWF	OSF	RNF
COUNT	10000	10000	10000	10000	10000	10000
MEAN	0.0339	0.0046	0.0115	0.0095	0.0098	0.0019
STD	0.1809808427	0.06767051005	0.1066249825	0.09700871646	0.09851360562	0.043549737 75
MIN	0	0	0	0	0	0
->25%	0	0	0	0	0	0
->50%	0	0	0	0	0	0
->75%	0	0	0	0	0	0
MAX	1	1	1	1	1	1

NOTE: I separated the “describe” statistics to make it easier to read.

Upon reviewing this portion of our descriptive statistics I found that “Machine Failure”, and “HDF” are the most common failures having a larger mean compared to the other modes.

The OpenML page for this dataset states that “HDF” (heat dissipation failure) is the most prevalent failure. “RNF” (random failures) is the least common. This is logical as it is usually uncommon for machinery like this to randomly fail as it undergoes preemptive maintenance constantly.

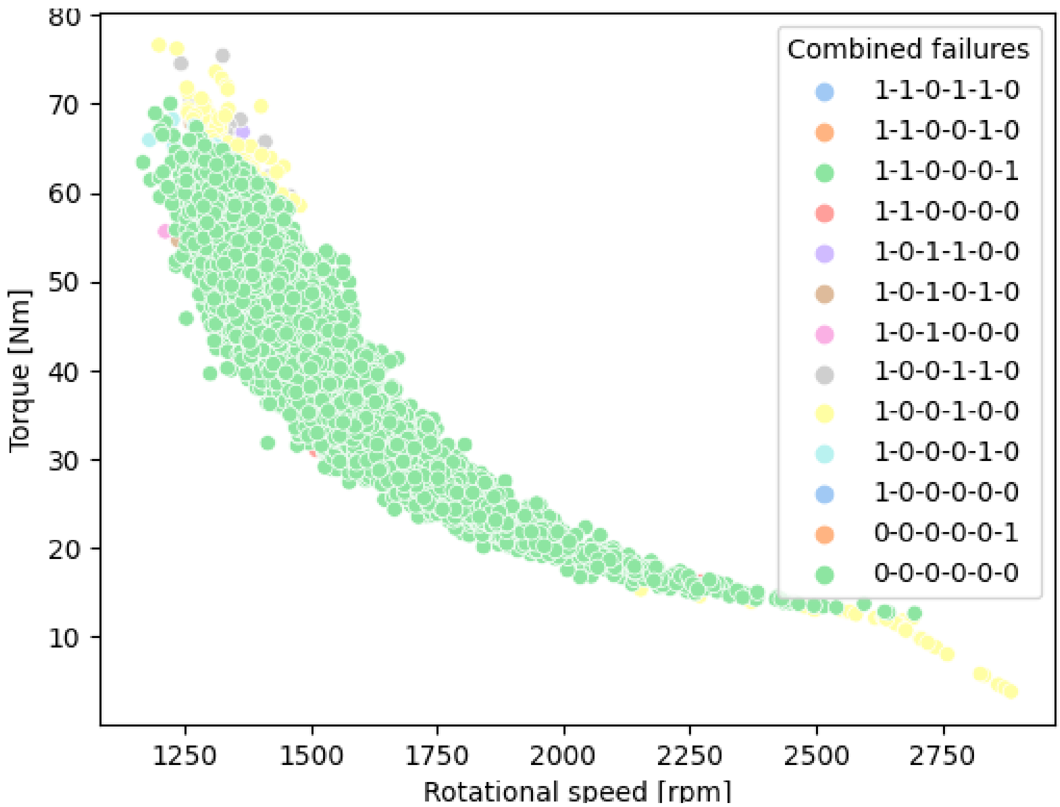
Graphical Statistics



Output from sns.pairplot(eeg, hue='Machine failure', palette='colorblind')

Here we can see all the the possible graph combintaitions of the data prior to an manipulation.

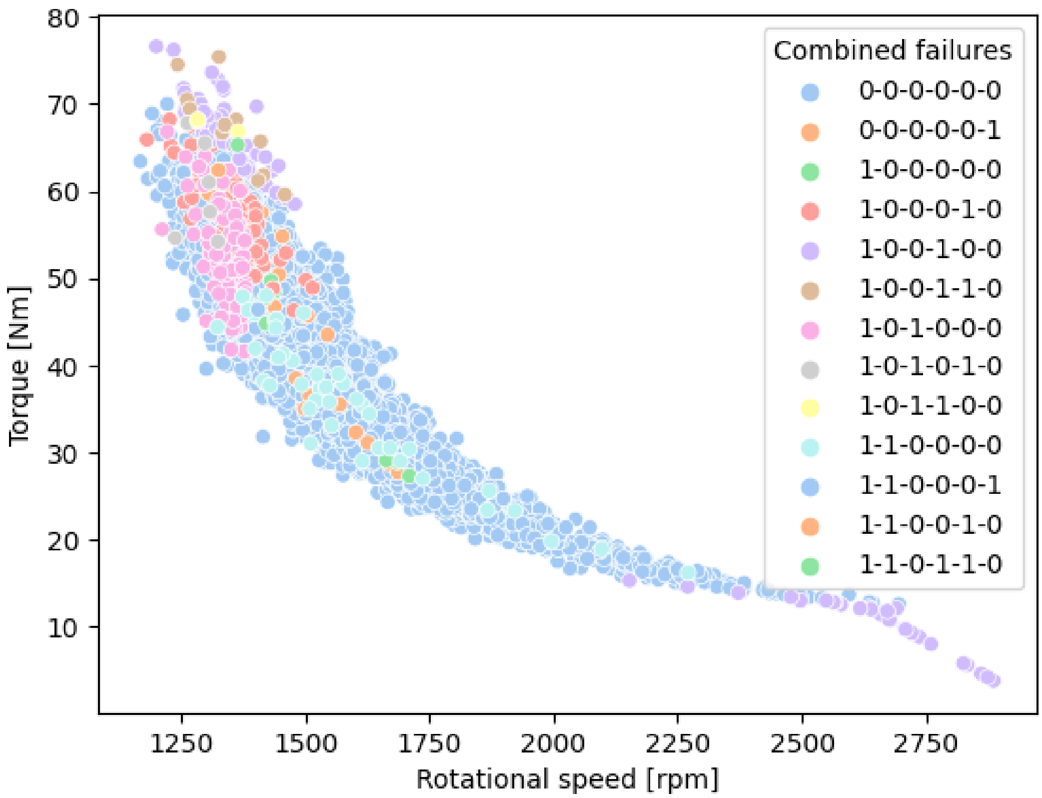
This image is a scatter plot of the columns “Torque [Nm]” and “Rotational speed [rpm]”. I’ve used all the different failure modes as a hue to better understand if these two values cause or can be correlated to failures.



In the combined failures table you can see a count of failures, the can be read as, “Machine failure count”-”TWF count”-”HDF count”-”PWF count”-”OSF count”-”RNF count”. (0-0-0-0-0-0) would suggest no failure, (1-0-0-0-0) suggests “Machine Failure” and so on.

Other than the clusters of failures of either end their are some points that lay in the middle which are strange. This suggests moderate to normal operating levels.I would conclude that this is probably because I have non-related failure modes graphed on Rotational Speed x Torque.

Taking the above graph and sorting the values of the combined failures to show less common points on the top you can see more detail about the data. Machine failures only ever occur when another failure is present except for when it’s a random failure.



You’ll notice that the majority of failure land either in the high range of Rotational Speed and low torque, while the others are on the inverse of that. Mechanical components or machinery don’t generally like to be run in those thresholds, so this partly explains why we see this in the data.

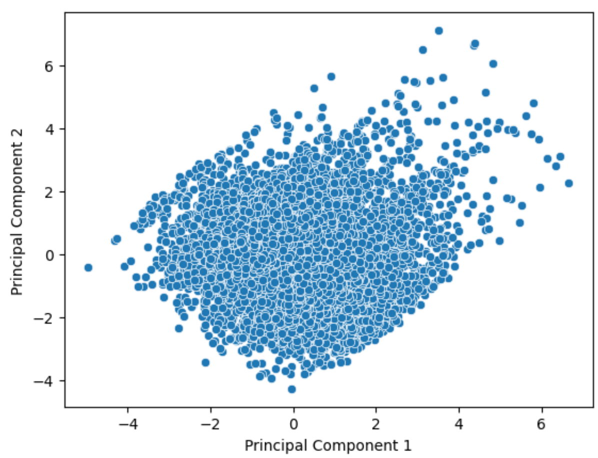
(I) PCA **Final Project ML - PCA.ipynb**

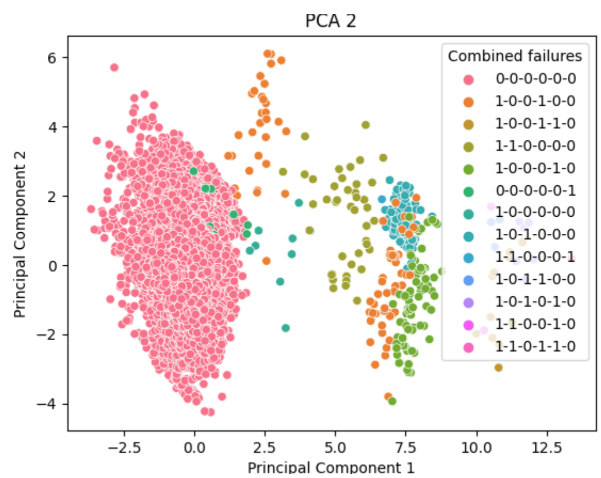
For this portion I squished the following columns into two PC's ('Air temperature [K]', 'Process temperature [K]', 'Rotational speed [rpm]', 'Torque [Nm]', 'Tool wear [min]')

The results from this gave very little insight into my data. The only thing I could really see is that there are no distinct clusters meaning there may not be distinct classes. This makes sense as I didn't include the failure data.

I decided to do the PCA again and I could see three more distinct classes which was interesting. This would point to those failure modes having meaning (we knew this).

Both graphs:





The second graph sets the colours based on the legend. The legend reads as distinctive counts of combined failure modes. These modes are in order from: [Machine failure-TDW-HDF-PWF-OSF-RNF].

(II) Decision Tree

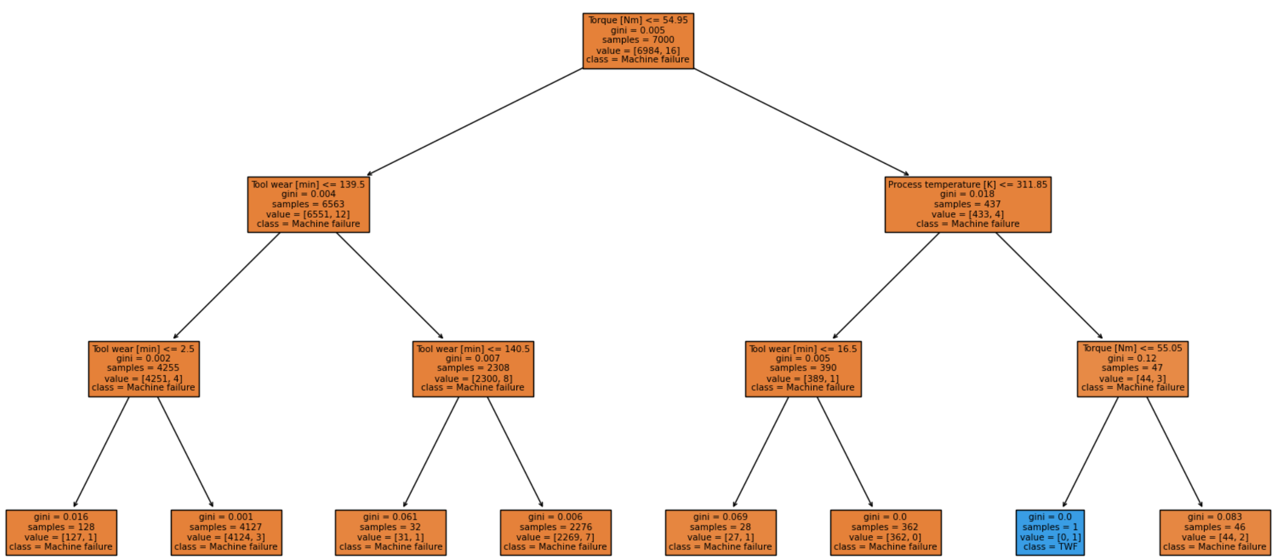
Final Project ML - Decision trees.ipynb

Working through decision trees was an interesting process. I was able to draw out a little information from it that provided insight on features from my data set that have a little more meaning.

Breaking down what I did first. I created a separate decision tree to predict on each type of failure. Then one to predict on all types.

What I found was that “PWF”, and “HDF” were the easiest to predict on using this single tree method. Something that is interesting is that PWF include Torque and HDF includes Tool wearing. These two features seem to be more informative as for most decision trees the first and second splitting questions used Torque then Tool wearing. This could mean that those three features have more weight in deciding a failure or a particular type of failure.

Here is an example tree from my note book:



Experimental Method

Final Project ML - Exploration.ipynb
Final Project ML - Neural Net.ipynb

Goal

With the data I have I want to use Random Forest and a simple Neural network to predict the failure mode of an asset with and accuracy of 85% or higher. This is a fairly realistic goal just based off some techniques I've used to investigate earlier. I would also like to minimize or lower my false negatives. A high recall would be ideal (mid .70s).

Hypothesis

I will be bale to predict at high accuracy (<= 85%) of failure modes of each kind. I will also need to consider recall as a metric, having high false negatives in the context of this data could cost a company millions of dollars.

With some of the failure modes it will be challenging to predict using based on the number of occurrences. This is more true for the decision tree technique, with the Neural Network I should be able to easily predict at 90% or greater accuracy on all failure types. This would be because the Neural Network has a greater ability to adapt to weird shaped data and learn more effectively.

Results

Results have varied. With the the forest I was able to produce high accuracy, and alright recal or minimization of false negatives.

My neural network was challenging as I learnt the weights required some adjusting since the neural network learned to falsely classify failures and increase accuracy. I have a hacky solution to resolve this be setting pre determined weights for 1s and 0s. With recal of this network I have seen as high as .76 and as low as .20.

In the bonus I hope to solve this issue by greeting a bootstrapped dataset with a GAN to equal out the number of failures and non failures thus eliminating the need for changing model weights.

Bonus: I tried using a GAN to bootstrap my data - Work in progress, having issues setting up meta data for my SDV GAN.

Final Project ML - Neural Net v2.ipynb

References

https://keras.io/

https://www.tensorflow.org/

https://scikit-learn.org/stable/

https://sdv.dev/SDV/user_guides/single_table/ctgan.html

OpenAI - ChatGPT (Some coding help/understanding concepts)

DALL-E for the image on the cover page