**Predictive Analytics - Preventive Maintenance Model**

**Raghu Raman Nanduri**

**DSC680 – Fall 2019**

<https://iamnrr.github.io/>

# **Business Objective & Understanding**

In manufacturing industries, downtime for heavy machinery costs lot of dollars both in terms of idle time lost due to repair work and also repair costs. If the businesses can be pro-active and perform regular maintenance tasks proactively along with predicting issues before hand using historical data, it would be a huge boost to their bottom line. In lieu of that industries usually employ IOT (Internet of Things) sensors to monitor and gather readings from various telemetric sensors. By merging telemetry data and failure reports, a predictive model can be built that can predict future fault occurrences of the heavy machinery.

**Business Goal:**

The end goal is to create a proactive maintenance strategy that tries to predict future failures of various components in heavy machines. As mentioned earlier, it benefits the businesses by reducing operational costs, long term maintenance costs and maximizing production hours.

**Approach:**

CRISP – DM Methodology has been used to perform this supervised learning task.

# **Data Understanding**

The following data sources were considered for building this Predictive Maintenance Model.

* Telemetry : Time series data consisting of various measurements like - Voltage, Rotation, Pressure and Vibration readings from various machines.
* Machines : Information about machines.
* Failures : Records of failed components.
* Maintenance : Maintenance historical records of machines involving component replacements due to regular maintenance activity or due to failures.
* Errors : Historical errors thrown by the machines.

Below links has instructions to get data sets for this project. <https://gallery.azure.ai/Experiment/Predictive-Maintenance-Implementation-Guide-Data-Sets-1>

**Feature Variables:** Below is the list of variables in each data set.

Telemetry data set has below variables:

* datetime
* machineID
* volt
* rotate
* pressure
* vibration

Machines:

* machineID
* model
* age

Errors:

* datetime
* machineID
* errorID

Failures:

* datetime
* machineID
* failure

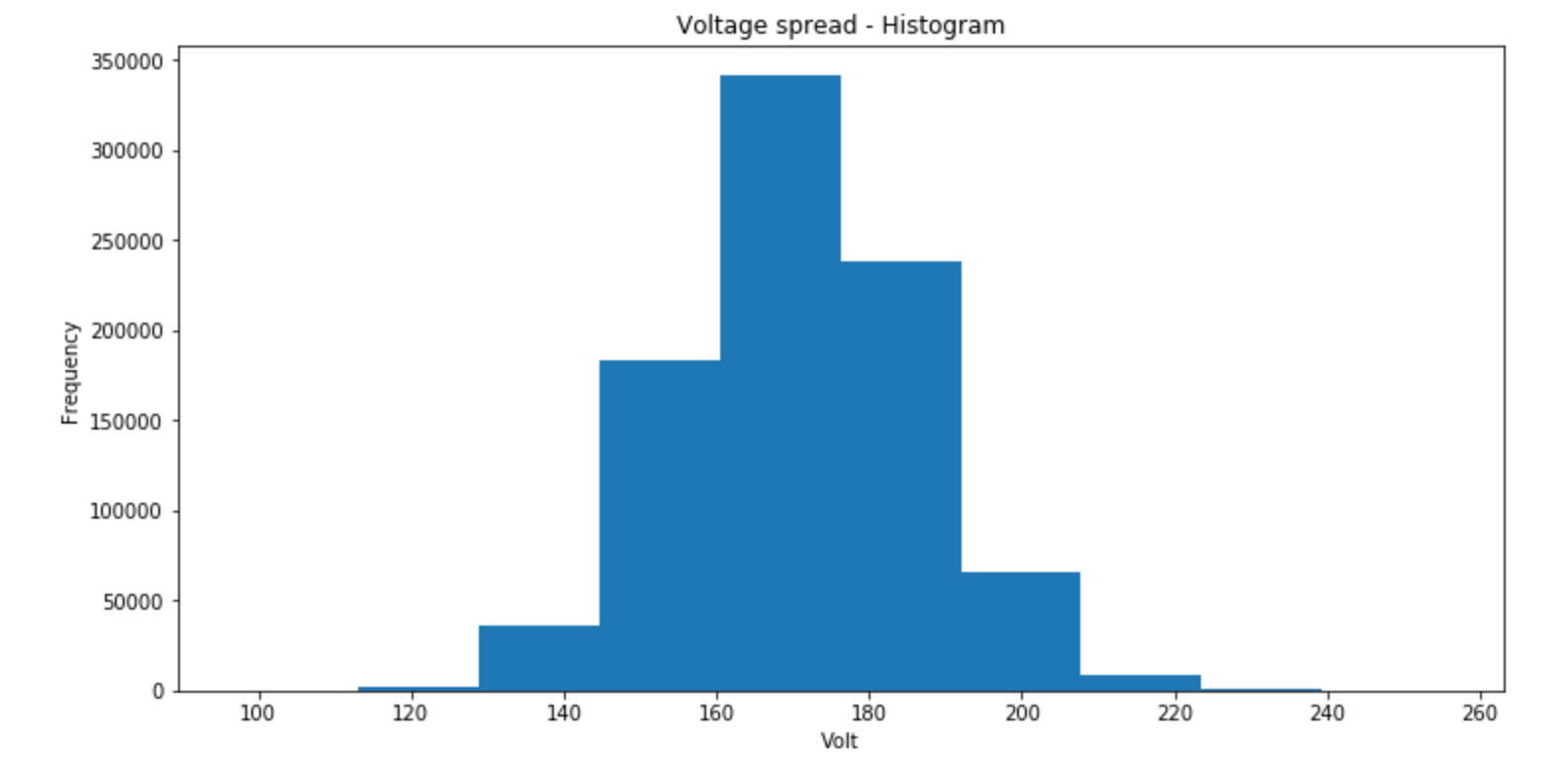
Maintenance:

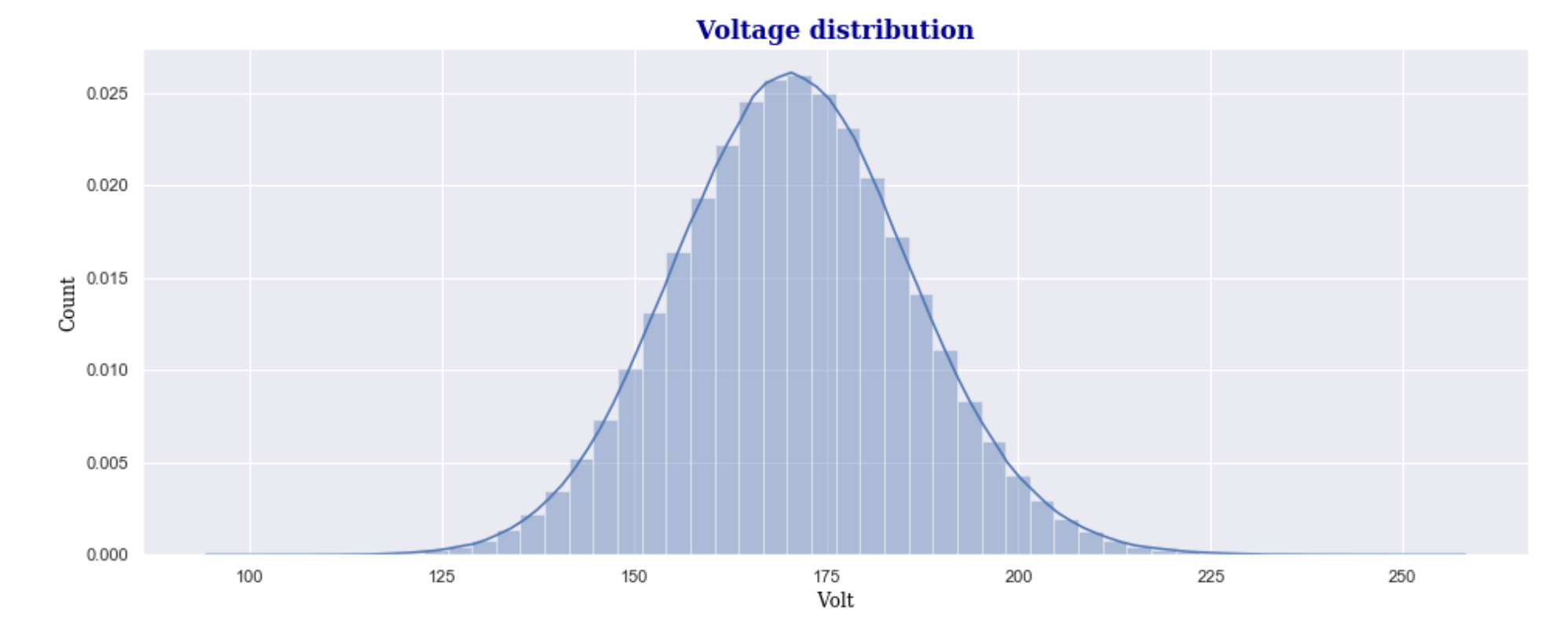
* datetime
* machineID
* comp

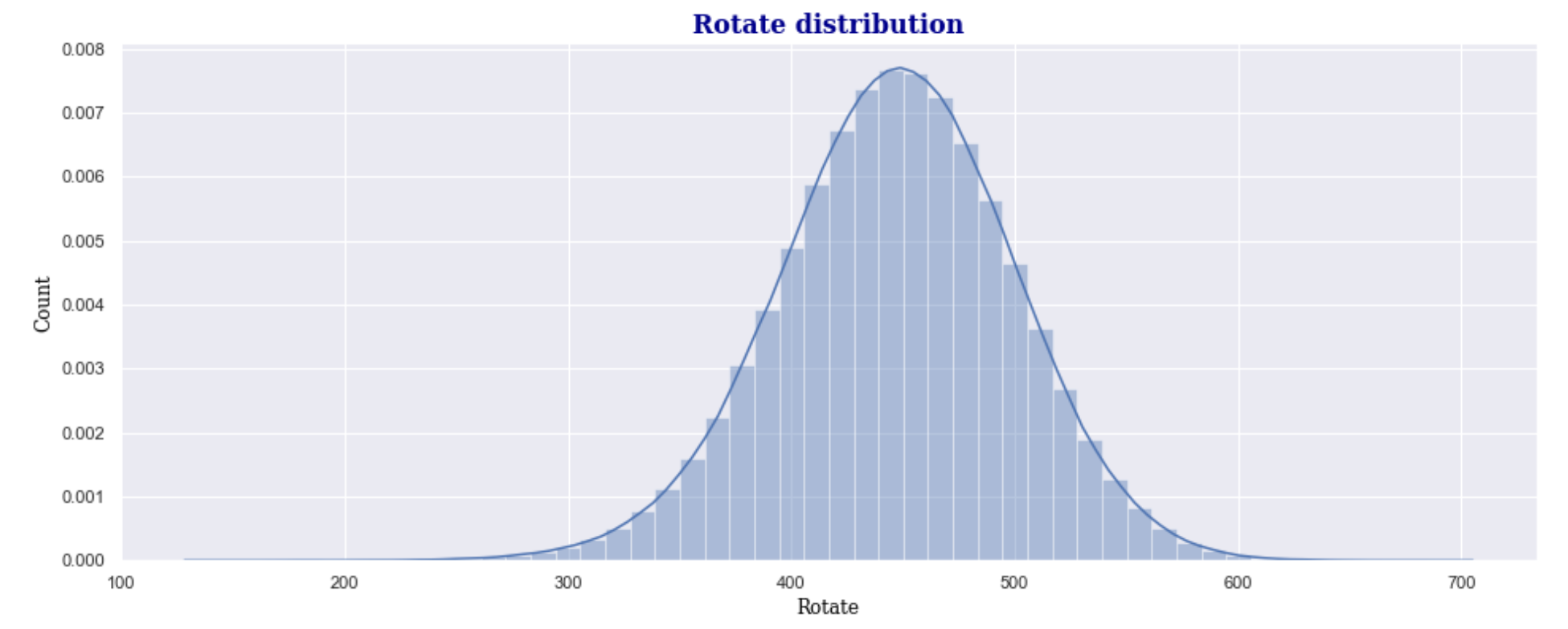
**Exploratory Data Analysis:**

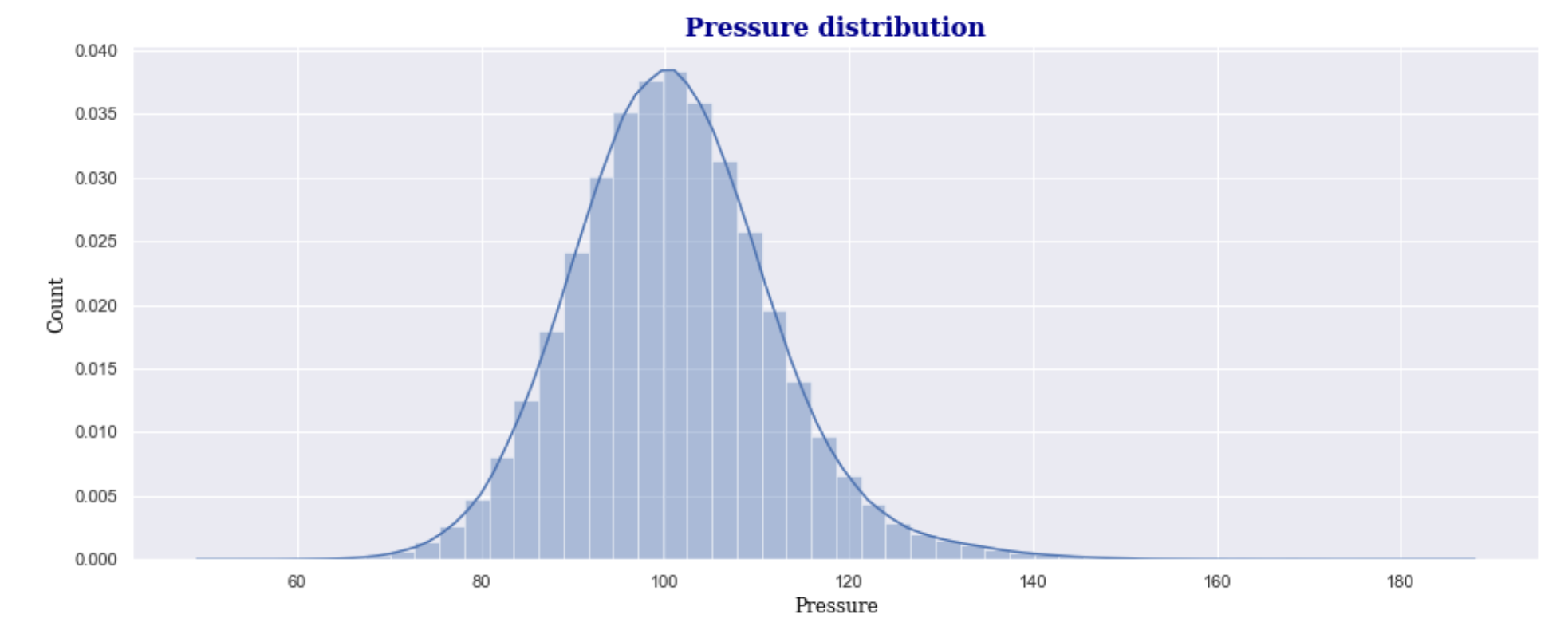
Firstly, I checked for missing values in all the data sets. Luckily, the available data sets are very clean without any missing values.

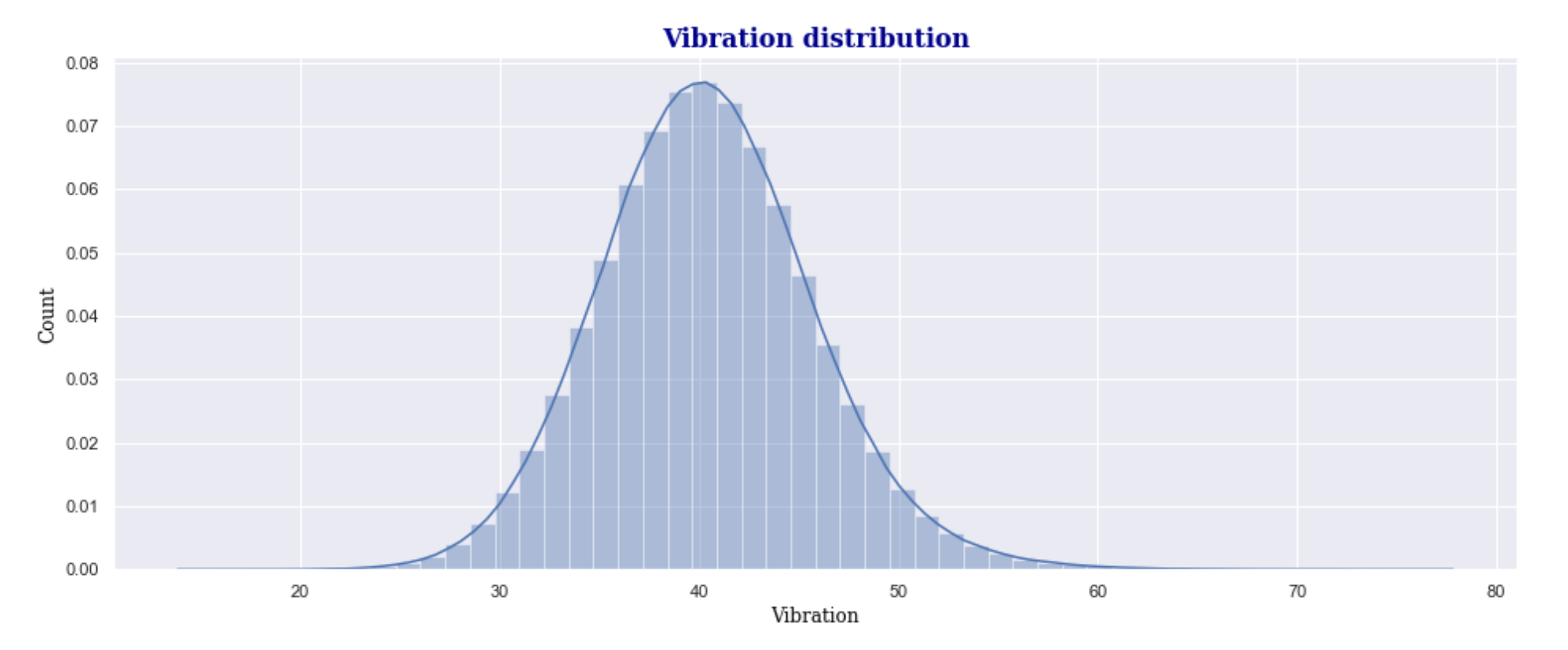
So, let us explore the variables visually.

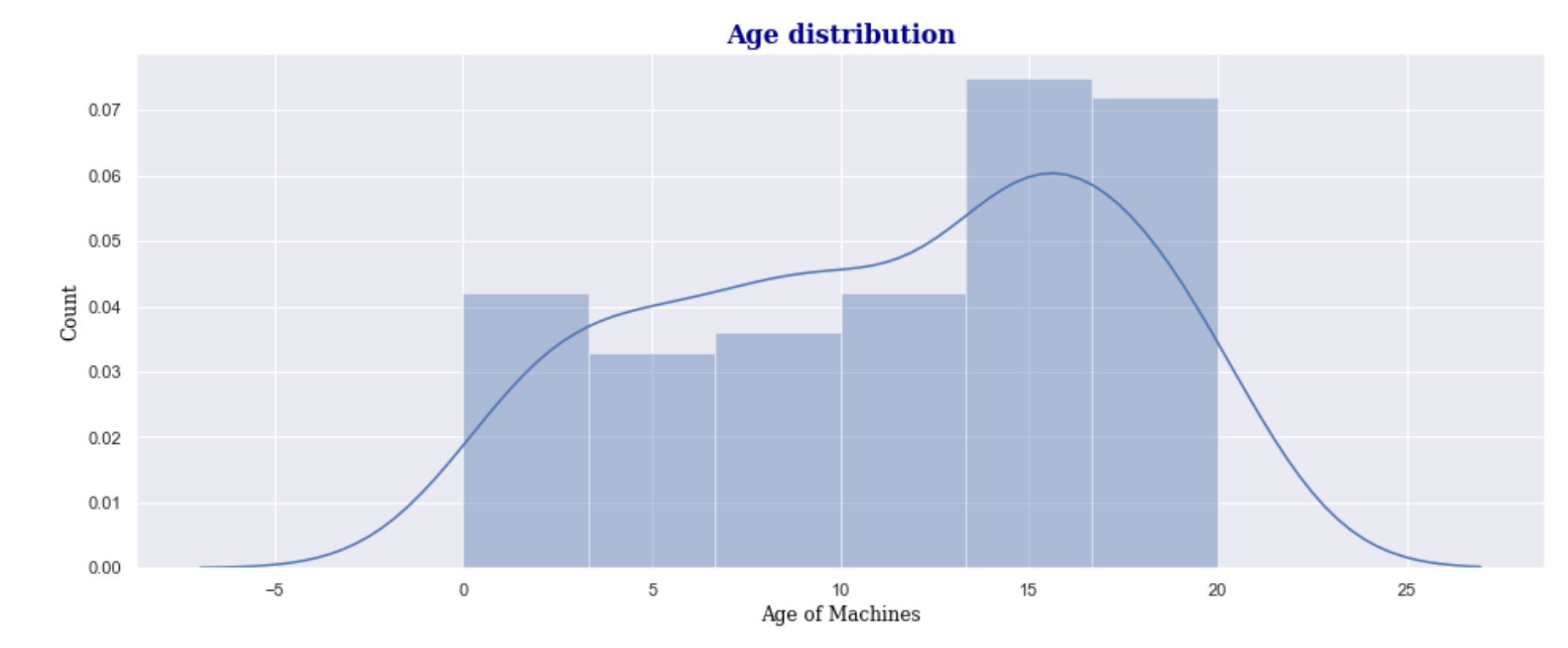
****

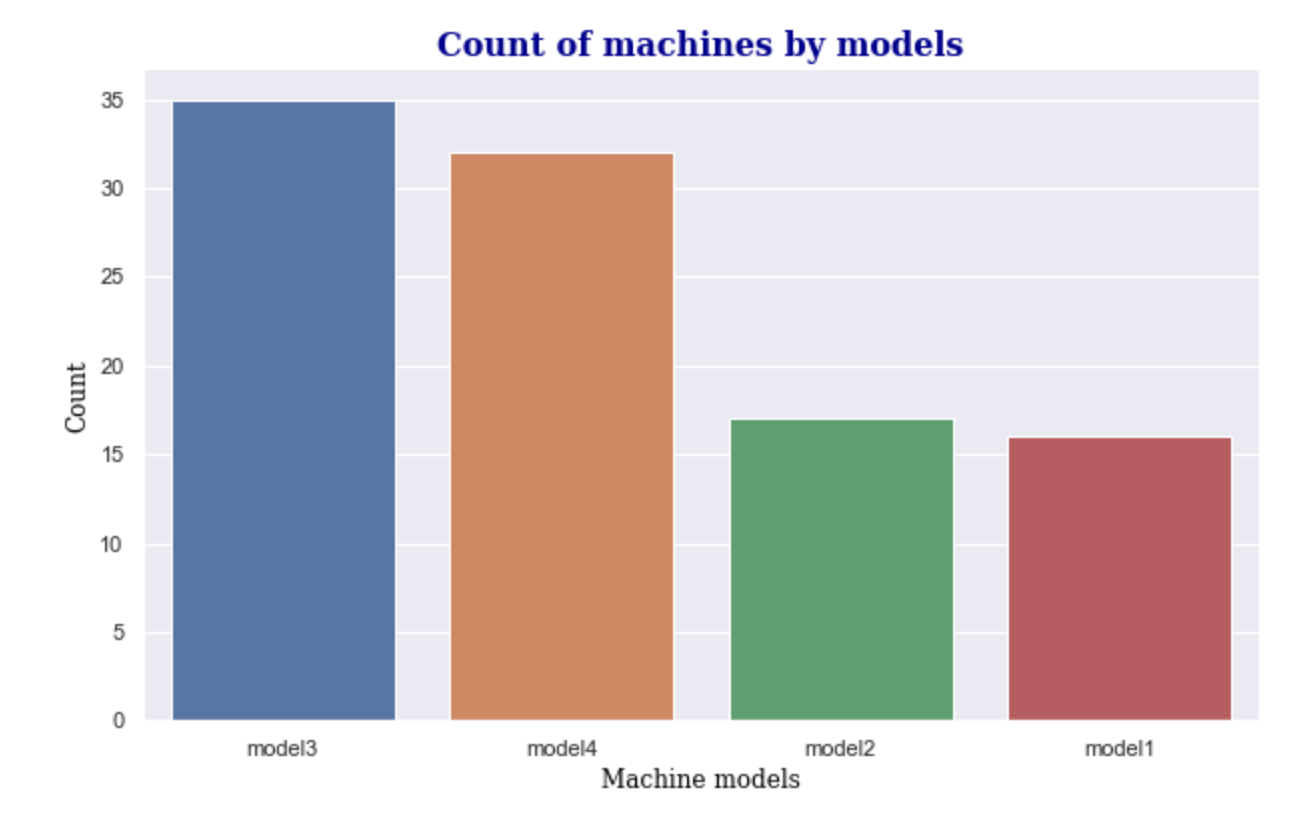
****

****

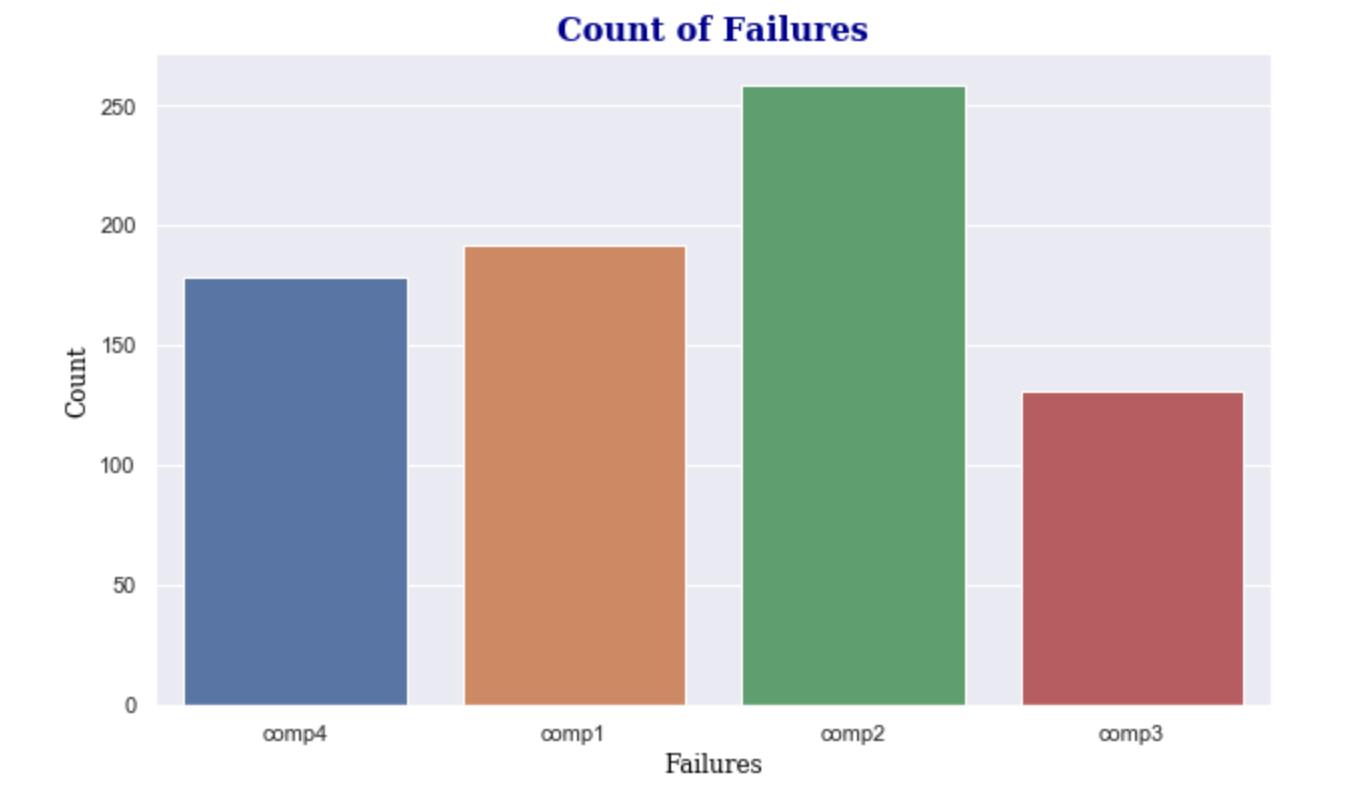
****

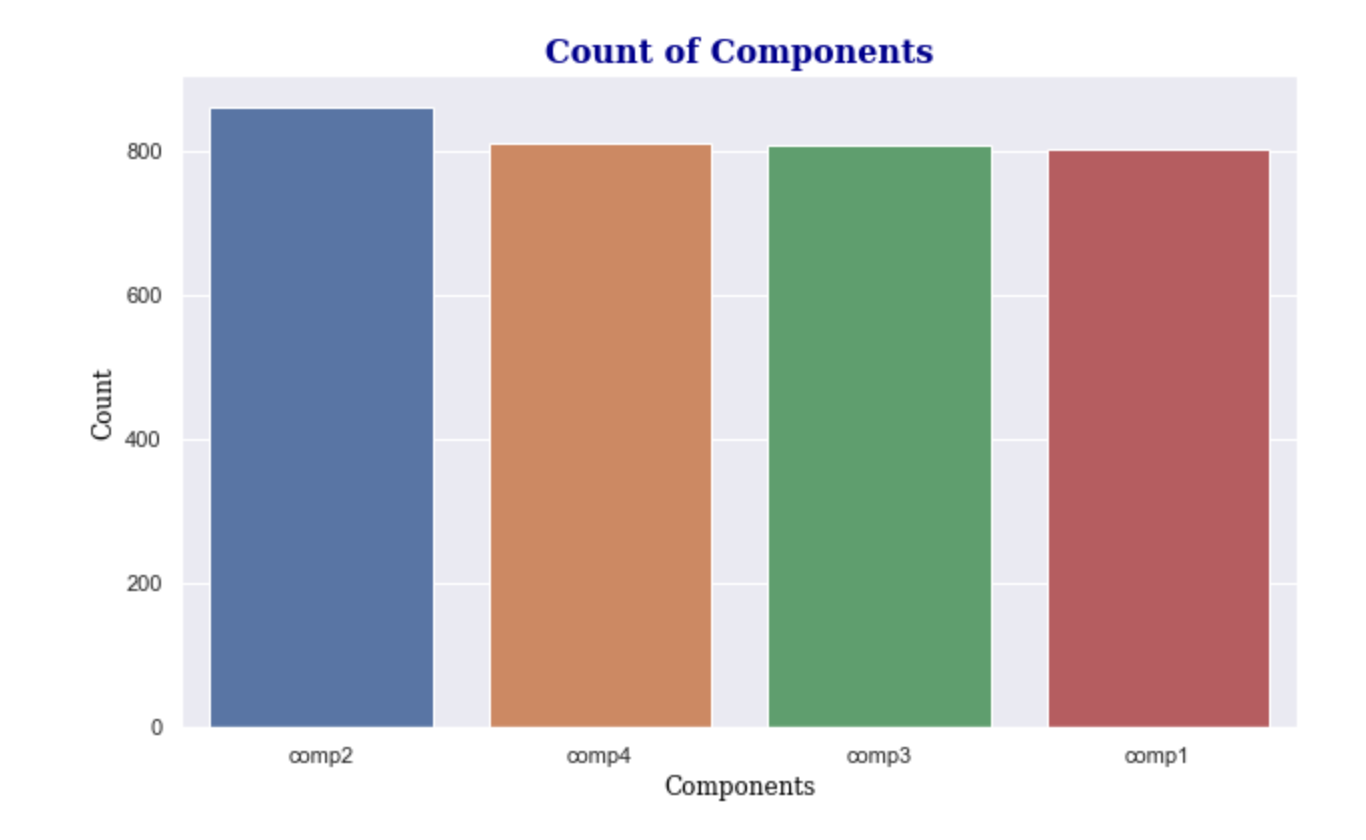
****

****

****

****

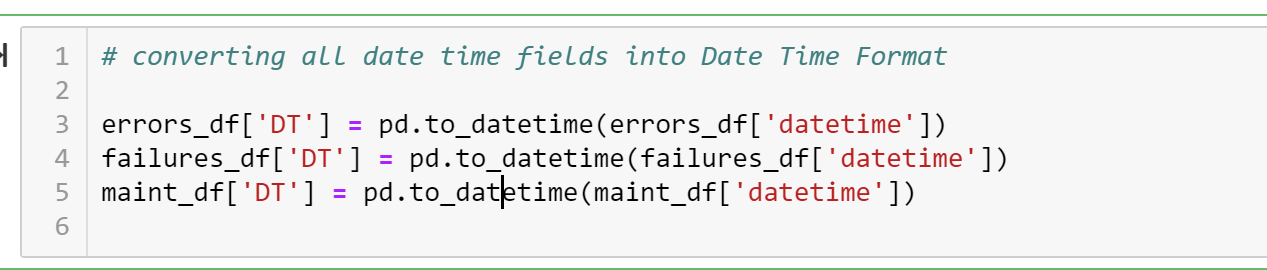
****

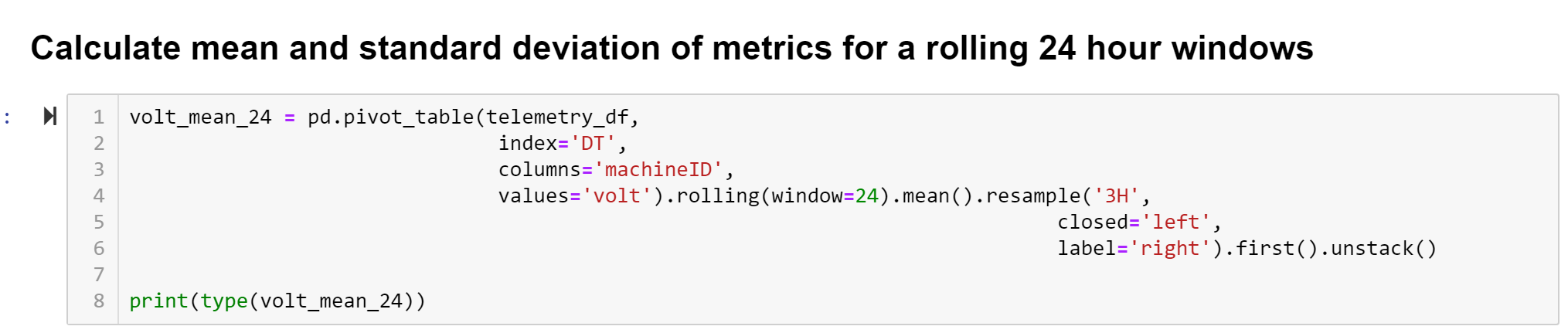
****

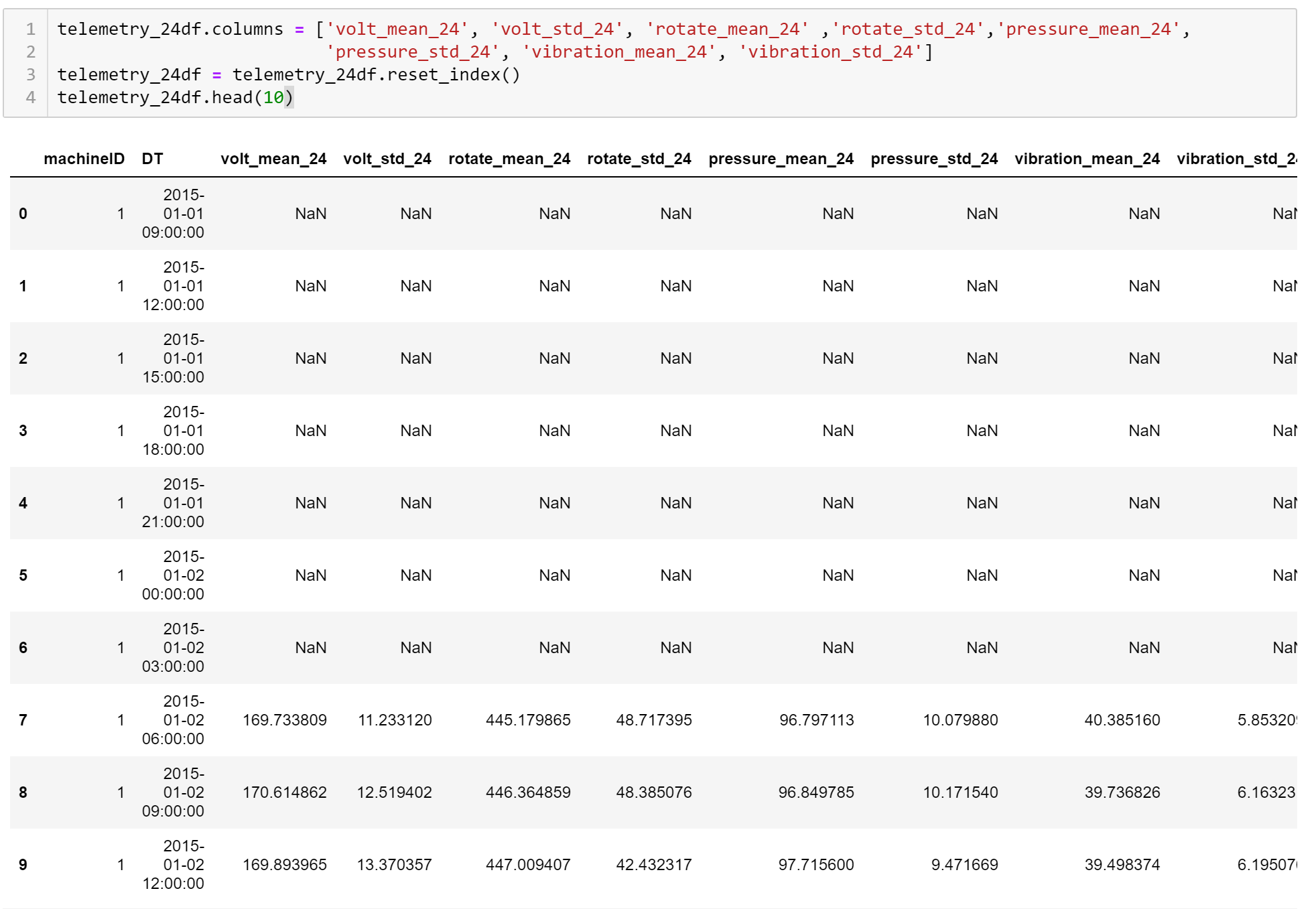
For more detailed exploratory data analysis, please visit my Git Hub portfolio - <https://iamnrr.github.io/>

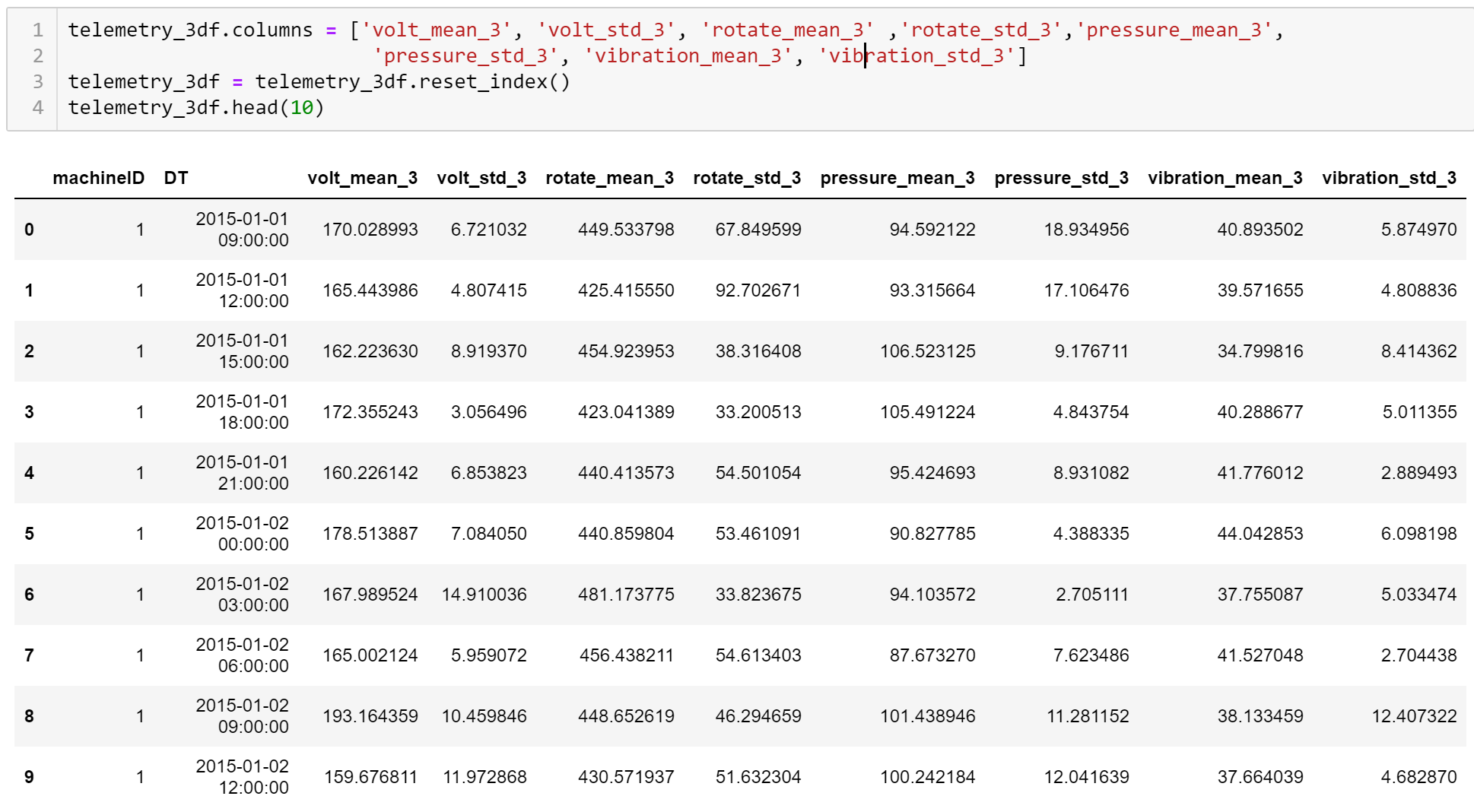
# **Data Preparation**

**Feature Engineering**





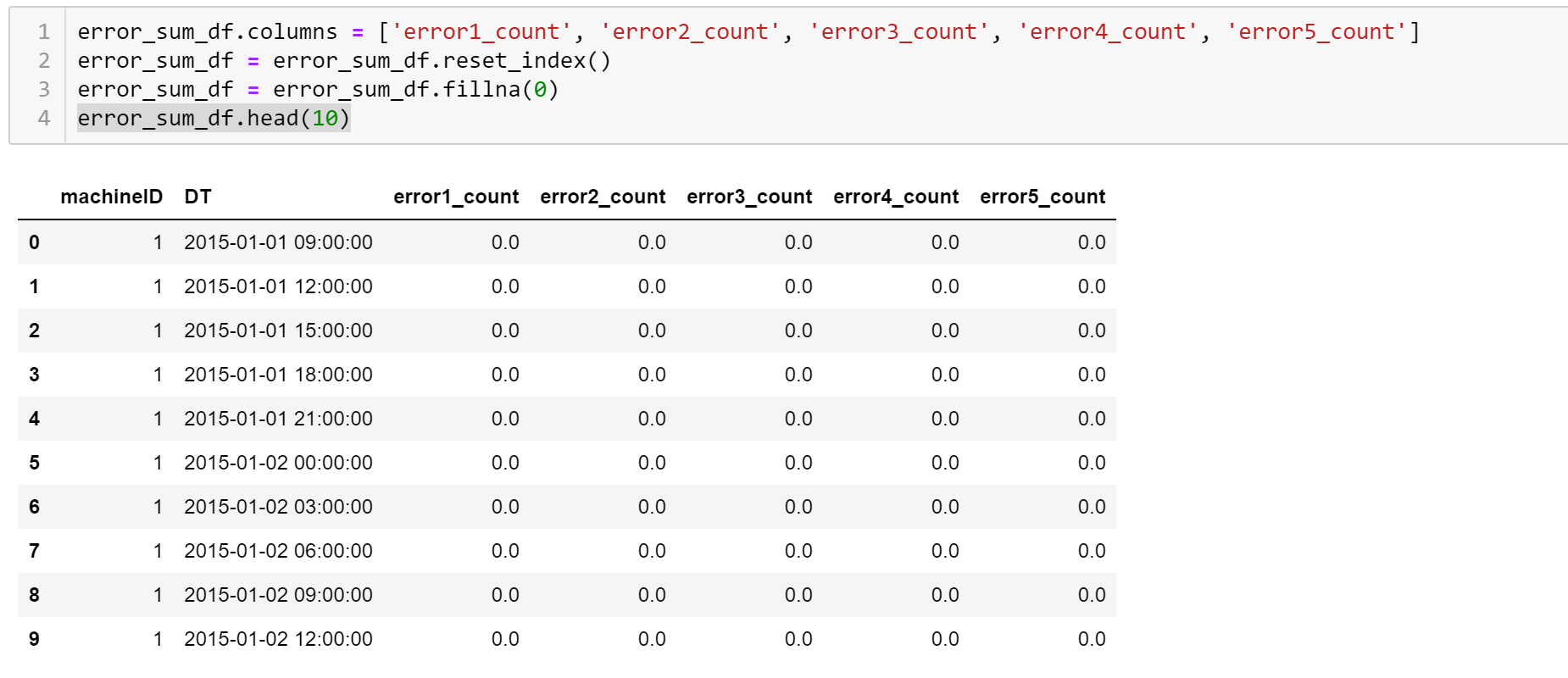


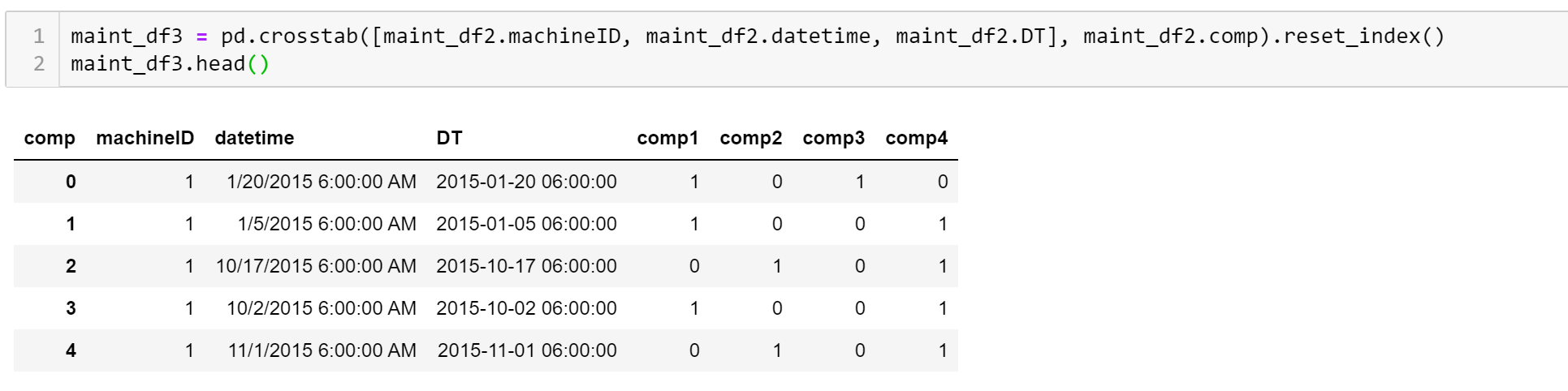


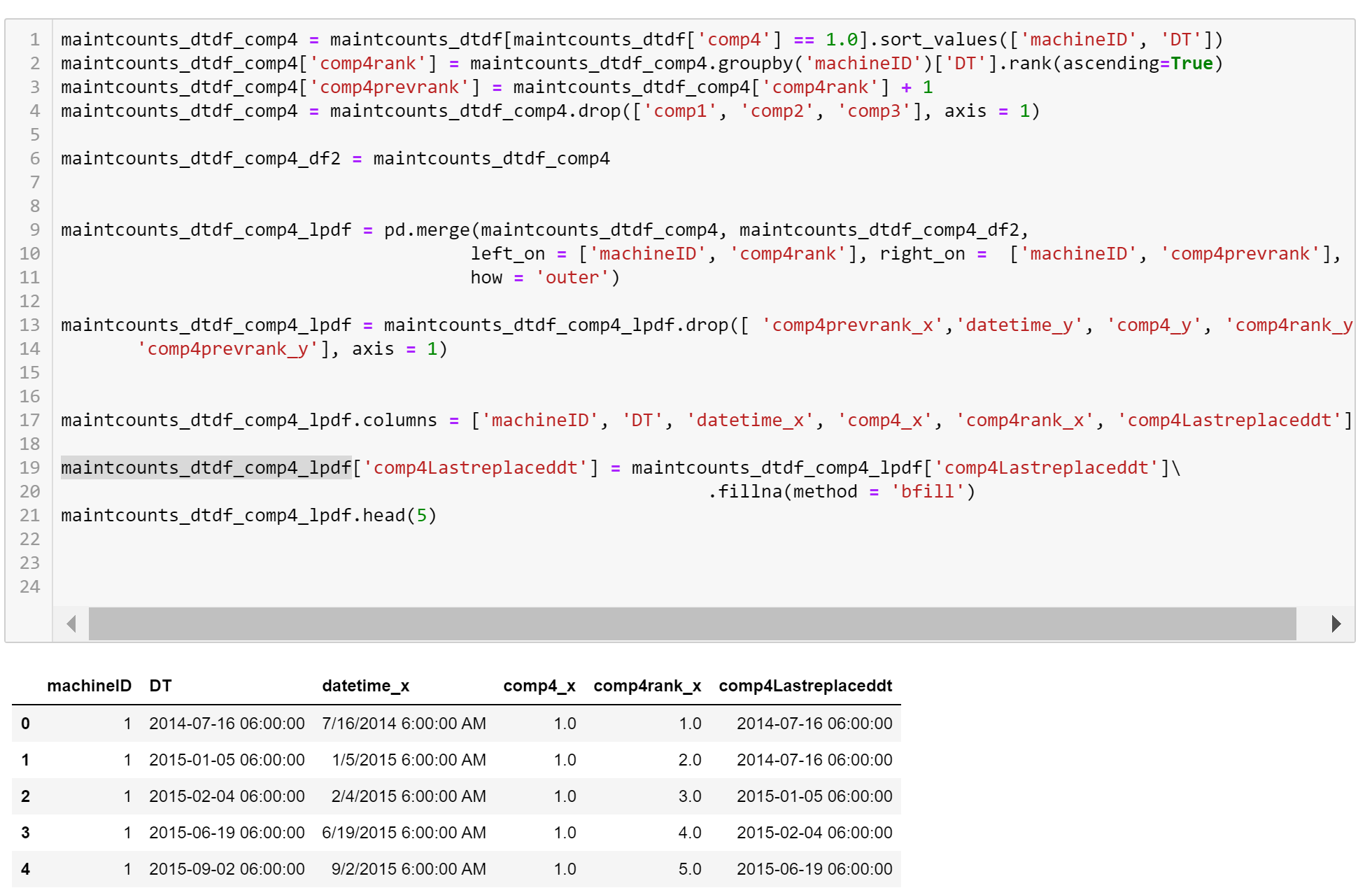
A screenshot of a social media post

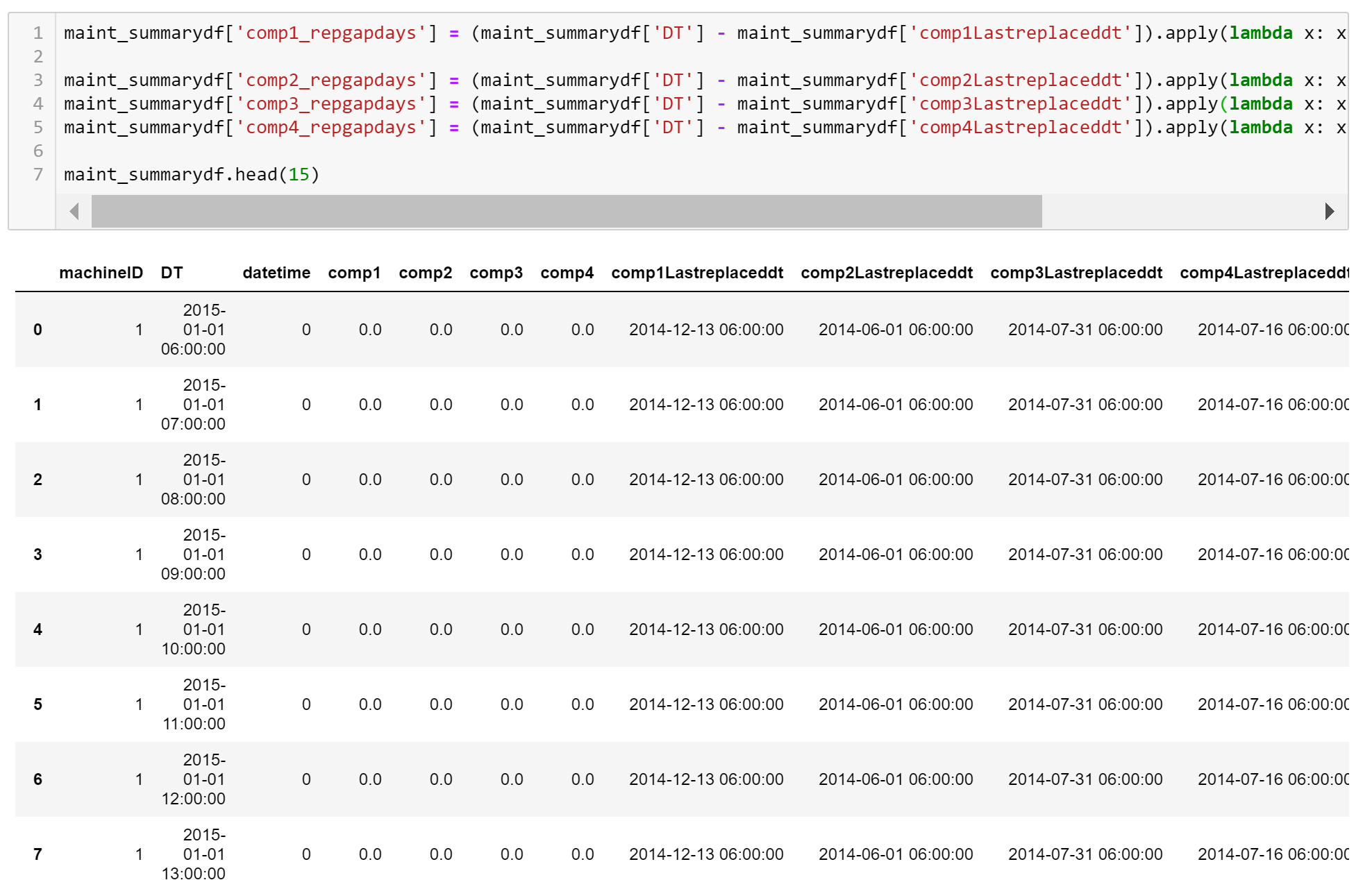
Description automatically generated

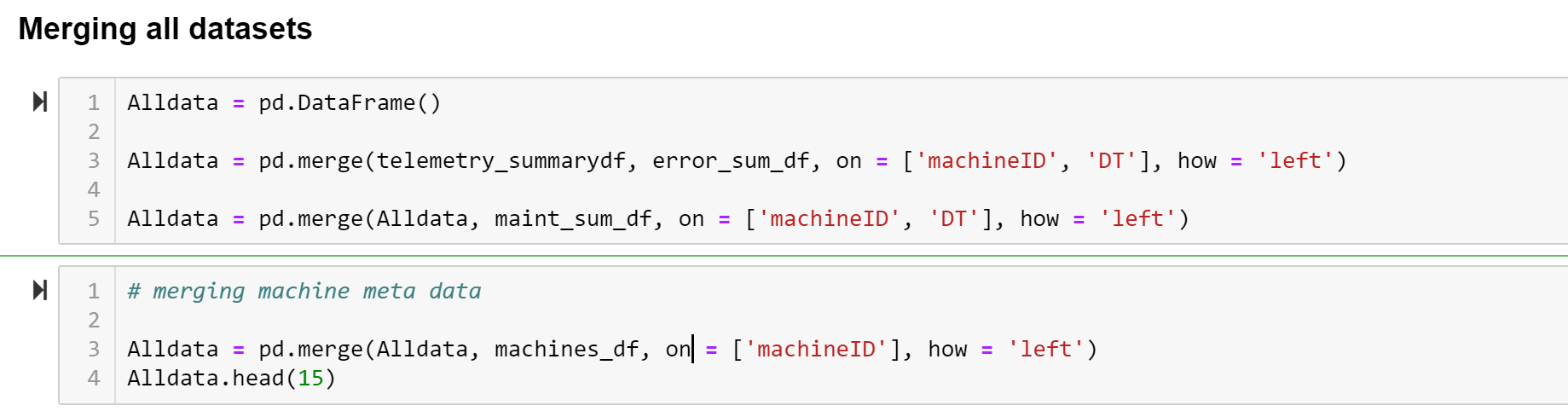


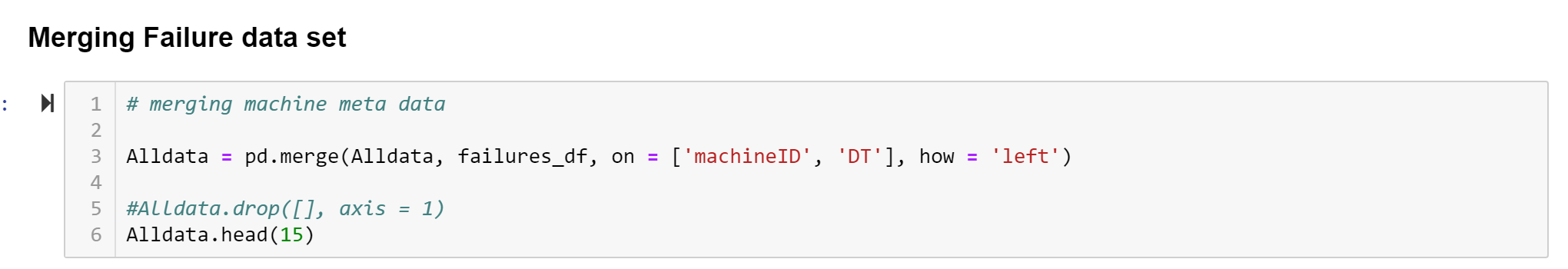








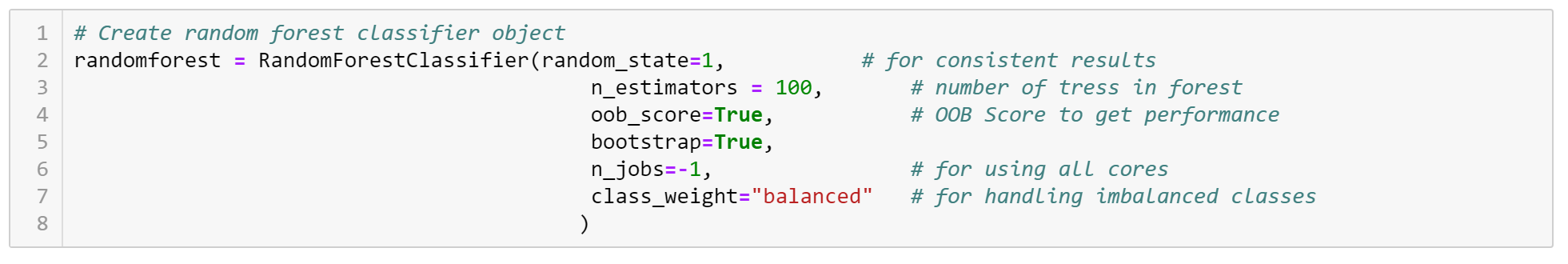




# **Predictive Modelling**

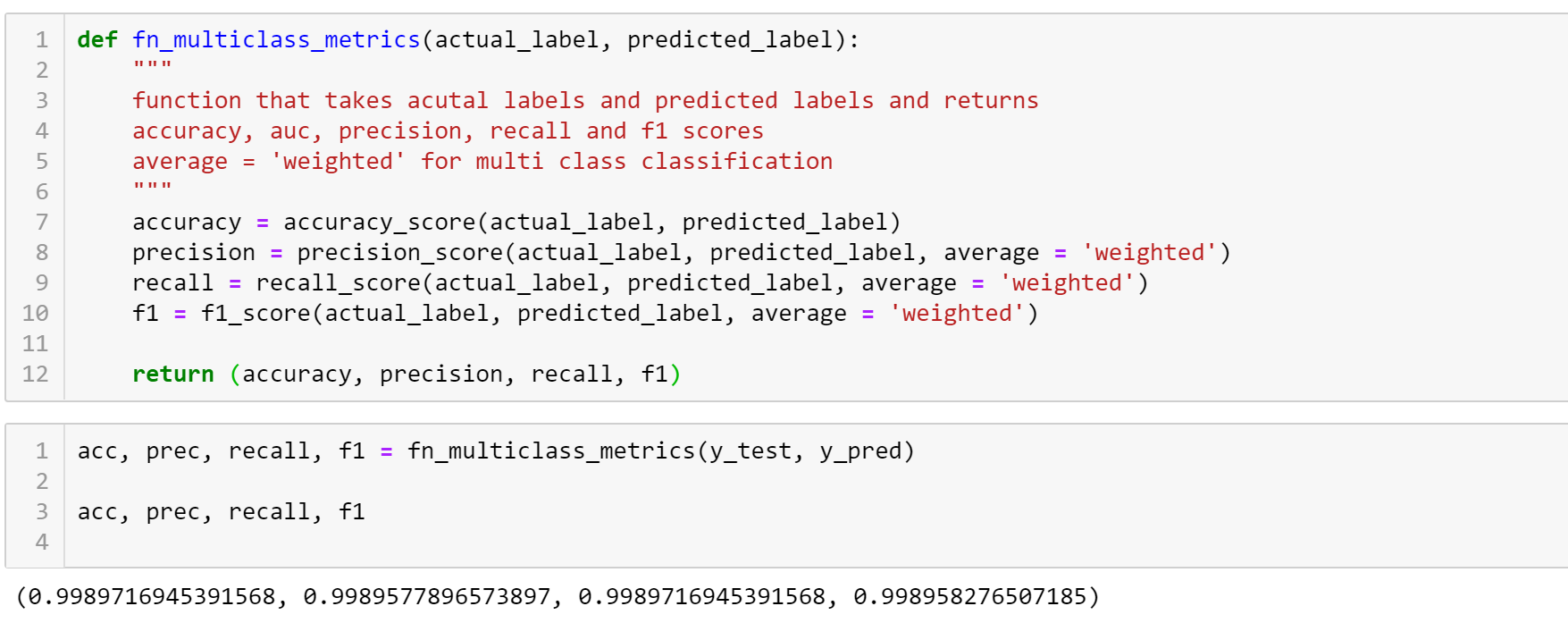
**Model Training**

Random splitting of data for creating training and testing data set does not make sense as failures and errors are time series-based events. So, for this exercise I have split the train data and test data based on the dates.



# **Evaluation**

In preventive maintenance prediction, the most important metric to evaluate the model is recall, which conveys the actual number of failures predicted by the model. Here in the model built. it is around 99.8%. I suspect this could be due to large portion of failure = 'none'. I am sure, model could be further tweaked to nullify this bias with the help of domain experts.



# **Model Tuning**

While I did accuracy and recall of above 99%, I think the model is slightly biased. The model can be further improved considering the technicalities in generating different feature variables in the preventive maintenance. Different kind of classifying algorithm like Gradient Boosting Classifier or Deep Neural Networks can be used to improve the model further.

# **Deployment**



# **Conclusion**

The code for this project can be obtained from my Git Hub repository [here](https://github.com/iamnrr/DSC680-Projects/blob/master/Customer%20Behaviour%20Prediction/README.md).

Please visit [my Git Hub Portfolio](https://iamnrr.github.io) to look at my other projects.

# **Assumptions**

# **10. Techniques**

Used the below modules in python to accomplish this supervised classification task.

* pandas
* numpy
* sklearn
* matplotlib
* seaborn
* joblib
* os

# **11. References**

1. <https://gallery.azure.ai/Experiment/Predictive-Maintenance-Implementation-Guide-Data-Sets-1>
2. <https://limblecmms.com/blog/predictive-maintenance/>