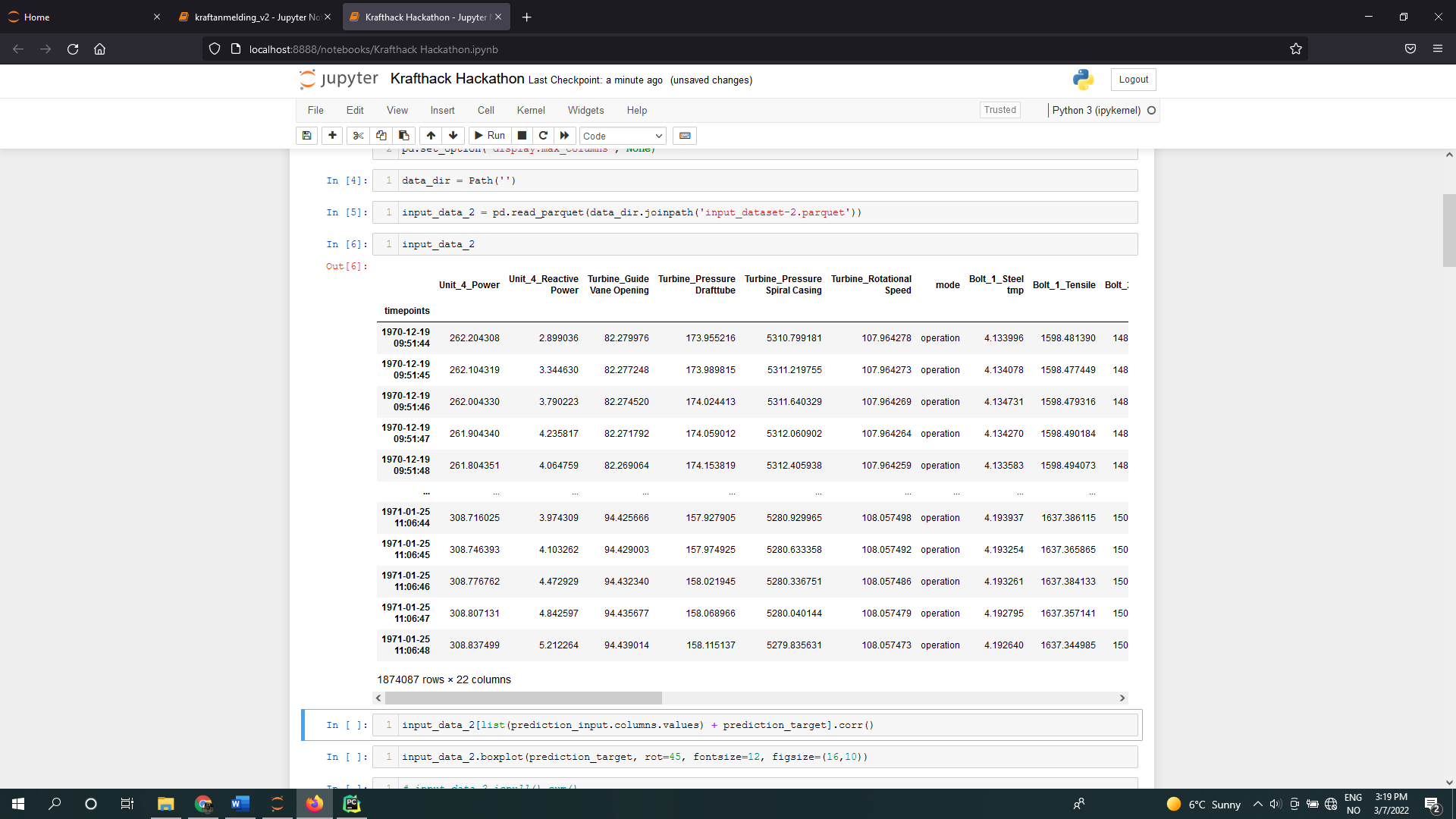
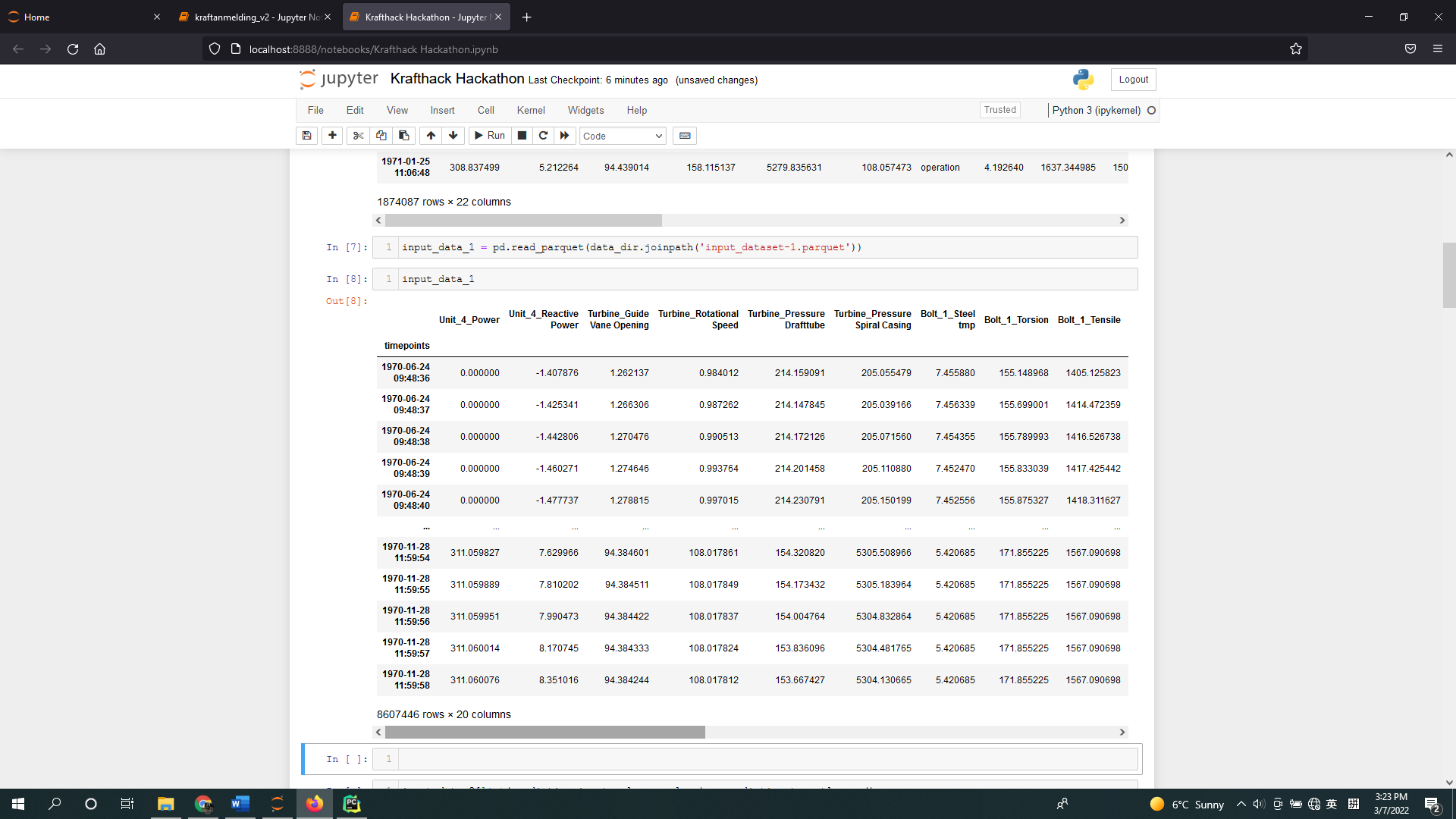
1. Exploratory data analysis

The data is recorded every second, there are approximate 2 millions rows in “input\_dataset-2.parquet” and more than 8 millions rows in “input\_dataset-2.parquet”.





According to “prediction\_input.parquet”, we need to use 'Unit\_4\_Power', 'Unit\_4\_Reactive Power', 'Turbine\_Guide Vane Opening', 'Turbine\_Pressure Drafttube', 'Turbine\_Pressure Spiral Casing', 'Turbine\_Rotational Speed', and 'mode' as inputs for the prediction, make 6 models for the tensile of each bolt ('Bolt\_1\_Tensile' to 'Bolt\_6\_Tensile'), so we merge the two input datasets using the prediction and target columns mentioned above.

A screenshot of a computer

Description automatically generated

The correlation matrix shows that the correlation between the input values and target values are quite low.

Chart, box and whisker chart

Description automatically generated

The box plot indicates that 'Bolt\_2\_Tensile' has quite unique behaviors, 'Bolt\_4\_Tensile', 'Bolt\_5\_Tensile' and 'Bolt\_6\_Tensile' have more outliers than the other three bolts.

1. Model training

There are over 10 millions data in the data set, due to large amount of dataset, the model is too slow to be trained (even with GPU), so we used only “input\_dataset-2.parquet” to train the model

Graphical user interface, text, application, email

Description automatically generated

Because the ‘mode’ is a string, we encoded it into one hot variable (‘mode\_operation’ and ‘mode\_start’ in the above picture).

A screenshot of a computer

Description automatically generated

We further divided the dataset into train and test, then scaled the data, considering the outliers in the dataset, we used RobustScaler.

Graphical user interface, text

Description automatically generated

We tested several different models and parameters, including Xgboost, SVR, neural network etc., even we only use “input\_dataset-2.parquet” to train the model, the neural network is still too slow to be trained (even with GPU), so from reproduction aspect and consider the real life application (in real life scenario, we may need to train many models for many generators, and retrain the model from time to time according to latest data), we finally selected Xgboost as the major model.

1. Production

In considering future production, we implemented a retrain\_model function, this function could loop through all the potential models and parameters, which are reasonable models and parameter ranges we found in step 2, we think these models and parameter ranges are good candidates for future scenario.

The physical conditions of the generator may change in the future, by implementing these candidate models, the users who is a domain expertise but don’t have too much data science knowledge can retrain the model and control the model used in the production easily.

Detailed code about the retrain\_model function is in the hackathon\_train.py file.