**Predictive Maintenance Model for Electrical Machines**

**Project Report**

**Team No:** 9 - Naveen Rajendran [NUID: 001304569]

INFO 6105 - Data Science Engineering Tools & Methods

Summer 1

**1.) Introduction:**

**1.1 - Background:**

In the current world of smart devices, the role of sensors cannot be neglected. Nearly every electrical & electronic device or equipment is embedded with wide variety of industry grade sensors which provides us continuous linear or non-linear time series data. With the growing number of IoT devices & connected systems, the amount of data generated has multiplied many folds in the past few years. The sensor data can be classified into two types ( Analog & Digital), the magnitude of parameters such as temperature, pressure, vibration, humidity etc. will serve as a meaningful information inorder to take critical decision at the organization level.

**1.2 - Problem Statement:**

A manufacturing companies we have machines deployed at the shop floor such as motor, pump, compressors, blowers, interstage coolers etc. These machines needed to run 24 x 7 throught the year inorder to achieve 100 % productivity in their respective businesses. The 100 % availability of production systems are highly impossible due to unexpected breakdown or damage of these critical electrical machines.

Inorder to have uninterrupted industrial operations there needs to be a proper maintenance of machines installed. Maintenance practice in a factory is classified into condition-based maintenance, preventive maintenance & shutdown maintenance. These maintenance activities are performed only when the machine gets failed or when monthly / yearly based preventive schedule is planned by the engineer. Therefore, the manufacturing company faces an unexpected breakdown of production systems which directly affects the product manufacturing cycle creating losses financially to the company due to increased maintenance cost and slowdown in production of goods.

**1.3 - Goal:**

As per the leading consulting firm Deloitte,

***“Predictive maintenance increases equipment uptime by 10 to 20% while reducing overall maintenance costs by 5 to 10% and maintenance planning time by 20 to 50%”***

Hence, there is a significant need to develop a predictive maintenance model where an engineer can prepare a schedule for maintenance well ahead based on the sensor data collected from the machine. In this project we are trying to resolve the unexpected machine breakdown problem using data science & machine learning techniques. For example, there is a sensor data meant for measuring temperature of the bearing on the motor. Based on the gradual increase in temperature level of the bearing we can predict that our motor is going to fail within few days

**2.) Dataset Description & Source Link:**

**2.1 - Dataset Link:**

**Pump Sensor Data (Kaggle)**

[**https://www.kaggle.com/nphantawee/pump-sensor-data/download**](https://www.kaggle.com/nphantawee/pump-sensor-data/download)

**2.2 - Dataset Description:**

The following data is the set of sensor data collected from an electric pump. There are around **51 sensors** embedded on the machine & there is a series of float data generated from those sensors & towards extreme right there is field named status to determine the status of the machine. The values from the sensor & the corresponding status of machine ( Normal / Failure) are given in the above-mentioned dataset. Sensor readings are accounted a primary data, whereas the **“machine\_status”** is coined as target attribute. Based on the values generated by the 51 sensors, the status of electric pump gets changed

**3.) Tools & Technology Stack:**

**3.1 - Programming Language:**

**3.1.1 - Python 3.8**

A reliable high level multiparadigm , general purpose, object-oriented programming language which is used extensively in web-development, data analysis & machine learning, backend software development. It has multiple libraries associated with respect to data science, machine learning & artificial intelligence

**3.2.Libraries:**

**3.2.2 - Numpy**

Library used for scientific computation that involves larger datasets in multidimensional array and matrices. It has wide variety of mathematical function including algebra, statistics, signal processing etc.

**3.2.3 - Pandas:**

Pandas is a library used for creating dataframe out of the data obtained from the imported datasets. A data frame is a mutable entity that consist of rows, columns & data

**3.2.4 - Scikit:**

A library which primarily comprises of different types of machine-learning algorithms like regression, classification, clustering & dimension reduction

**3.2.5 - Seaborn & Matplotlib**

These two libraries are helpful in visualizing the datasets inform of graphs, attractive statistical plots etc. This powerful visualization libraries can help users getting broader idea of larger datasets in the form of high-level graphical representation. Matplotlib is used for basic-less attractive graph plots whereas, seaborn helps to visualize attractive & dynamic statistical plots

**4.) Method Used:**

For every successful data science model these four mandatory steps needed to be adhered inorder to deliver a meaningful outcome.

* Data Cleaning
* Exploratory Data Analysis & Further Data Processing
* Machine Learning Model Development
* Model Evaluation
* Result & Further Analysis

**Step 1: Data Cleaning & Preprocessing**

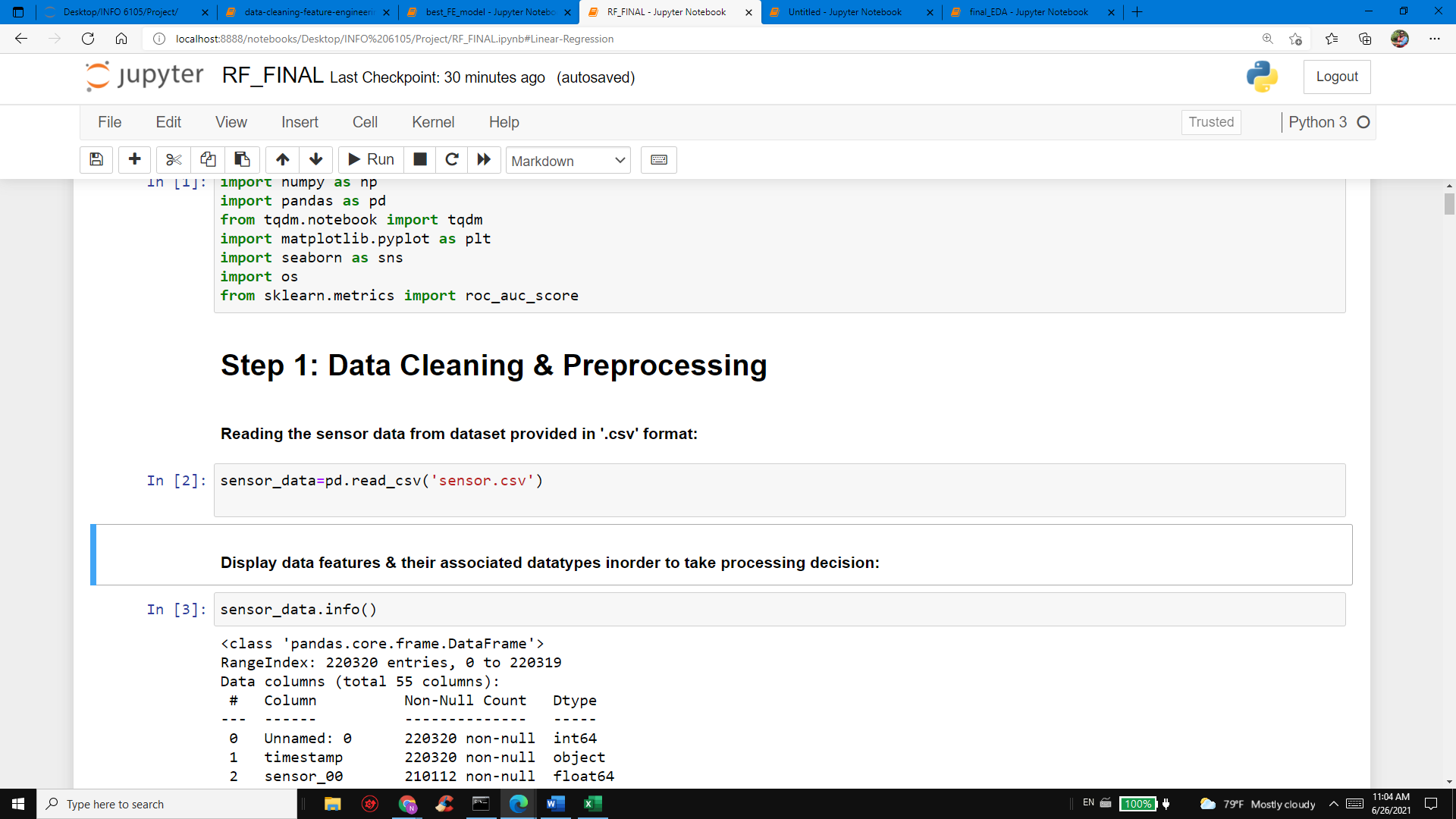
To process any dataset for a data science project the values corresponding to the feature must be clear inorder. Here we have feature set of 51 sensor data along with machine status as a target value. The sensor data is generally a wide range of float values spanning from few hundreds to thousands. Let us consider the values generated by sensor as primary data that need to analyze. We made ‘machine\_status’ column values as the target for the primary dataset of sensors.

**Data** : Float value generated by 51 no’s sensors

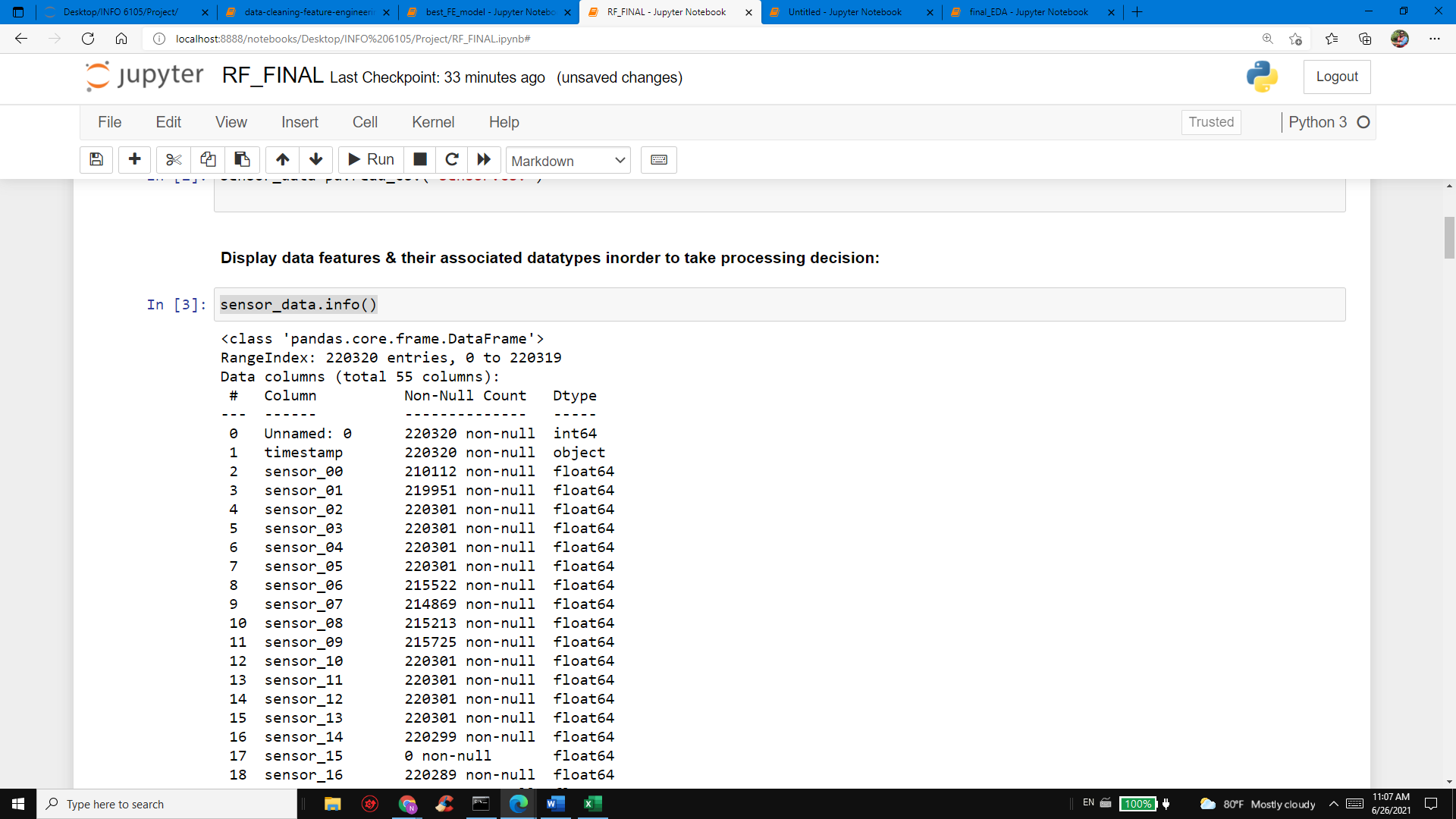
**Target** : Status of machine which is ‘BROKEN’,’NORMAL, RECOVERING’

**Converting Data from .csv to Pandas data frame for our analysis:**

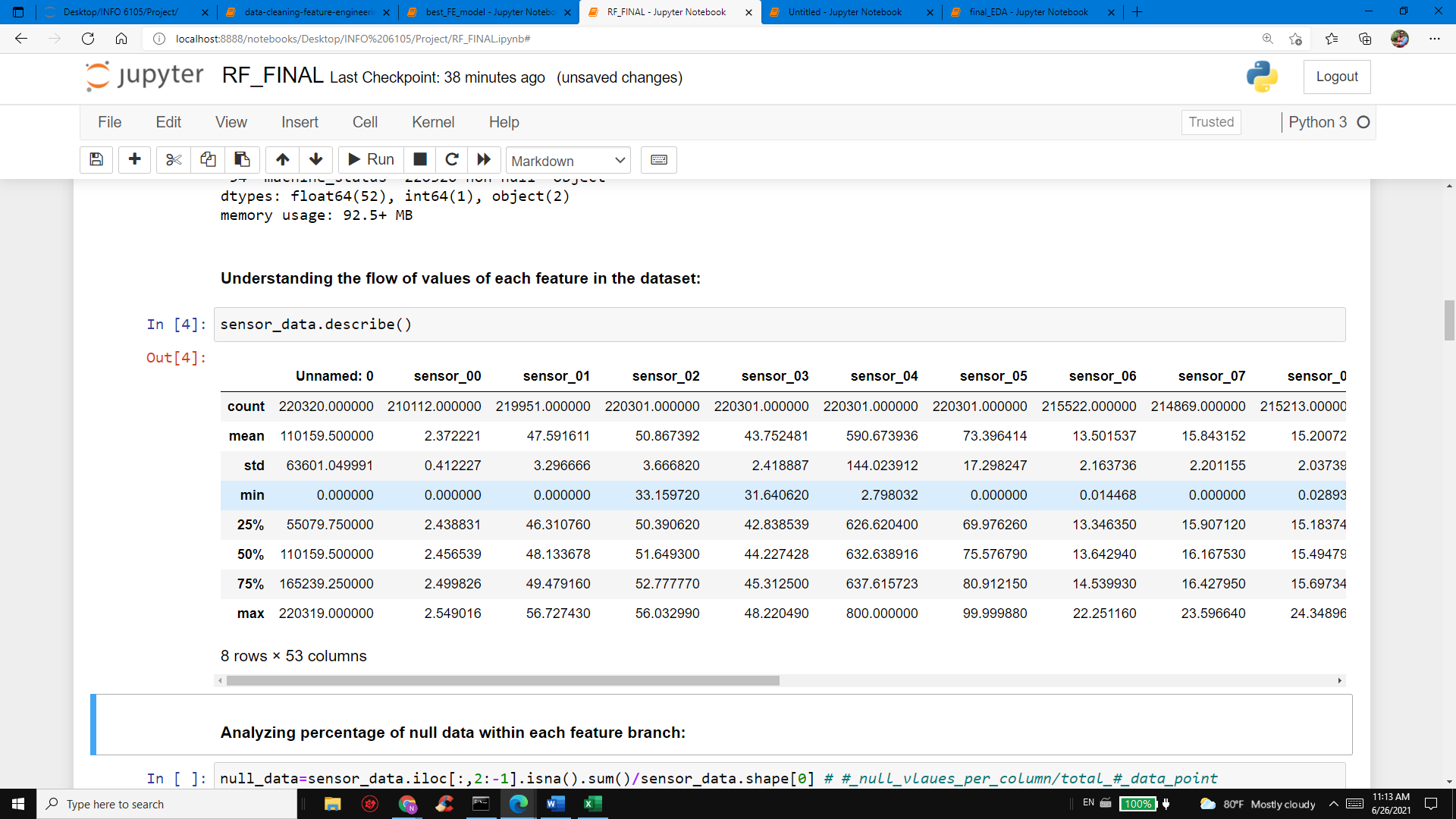
* It is a tedious job to read data everytime from a file & process it. Hence we are converting our dataset into a fram so that we can perform our manipulations on it



* Once we obtain a data frame it is necessary for an analyst to review the column & rows. So that we use info() to extract the detail inorder to make further decisions

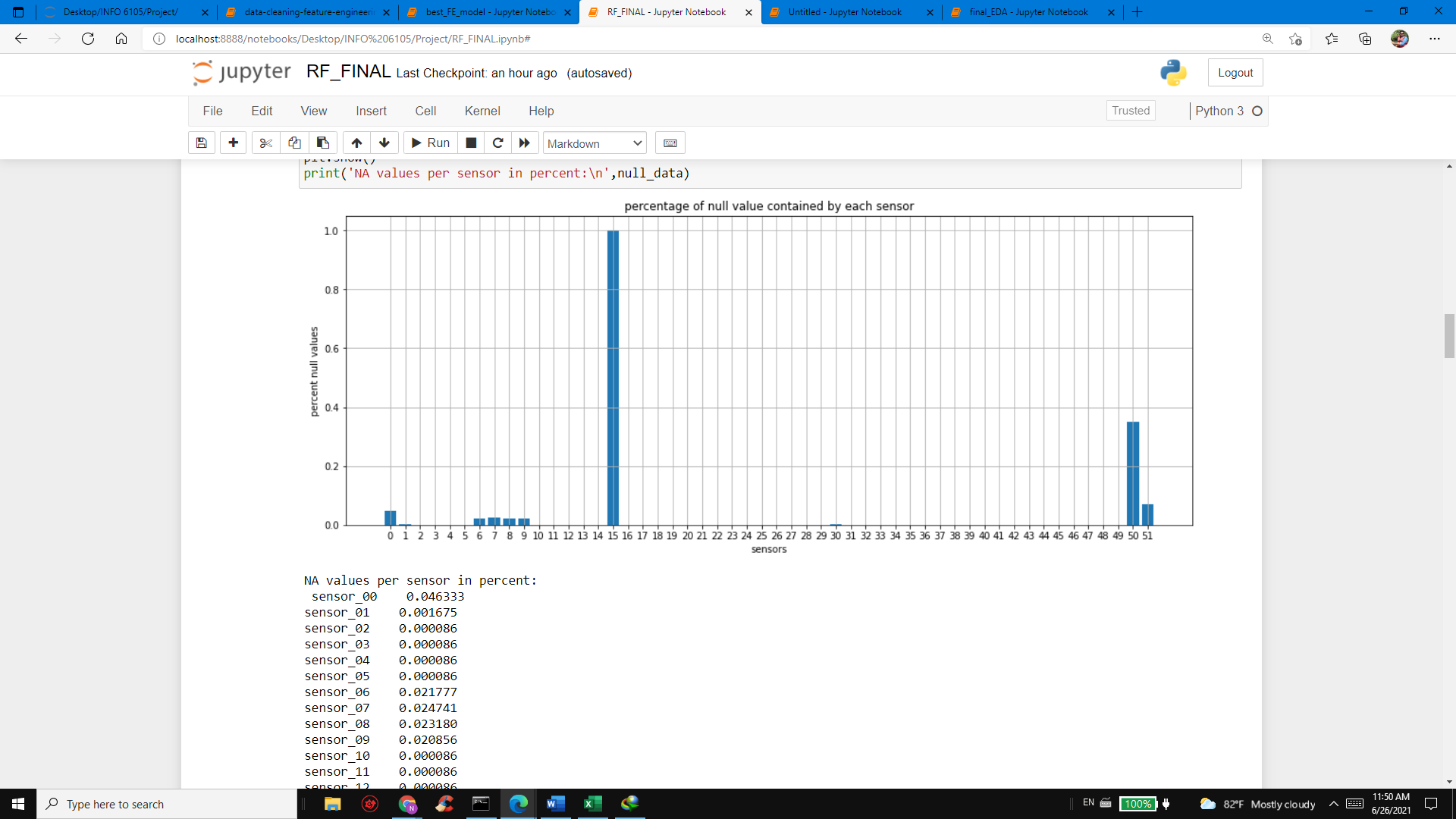


* Describe function on the other hand gives us additional information on the data such as mean, count, median, standard deviation , max & min components



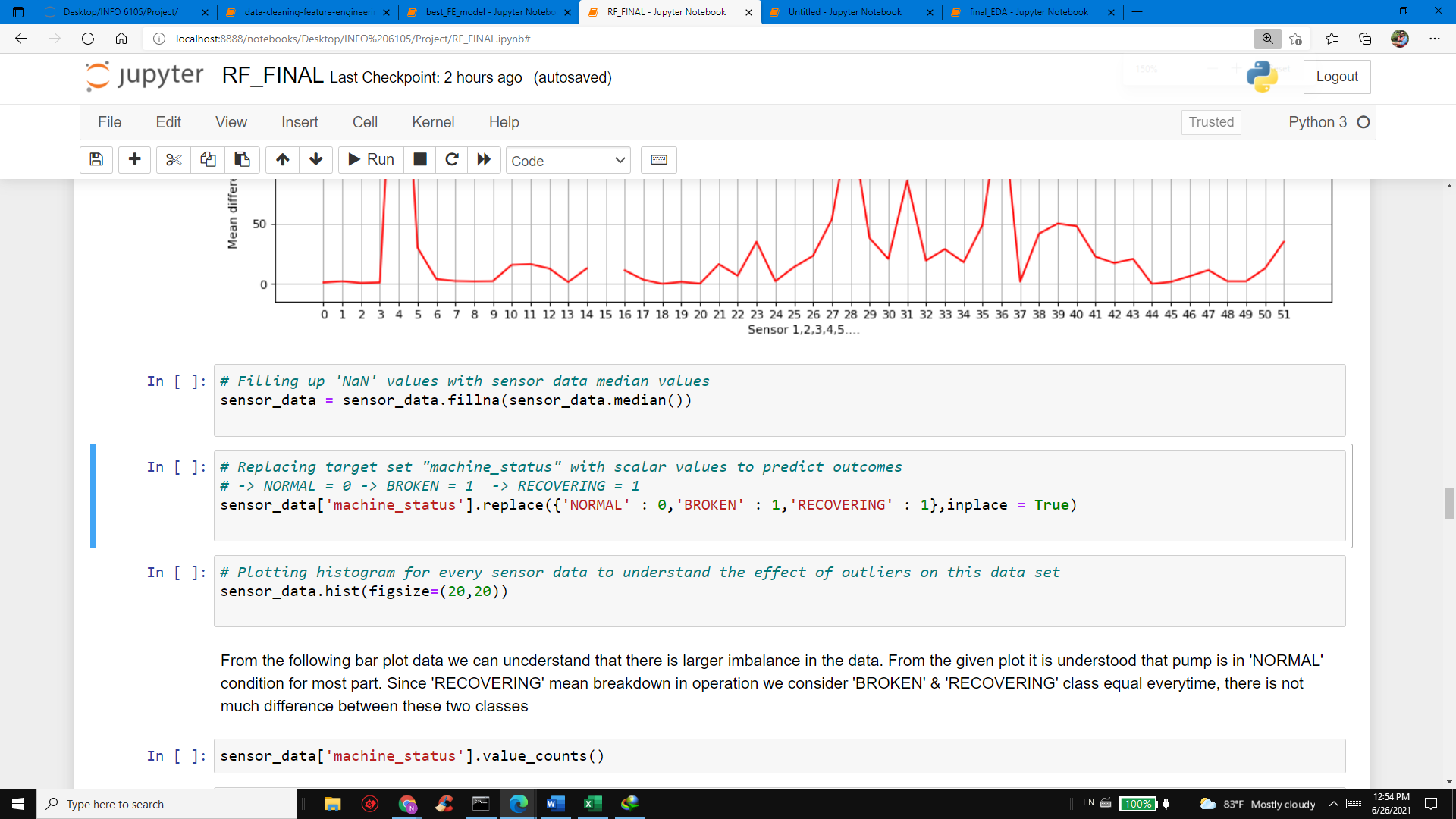
* It is quite important to understand the role of null values / negative values in the dataset,

normally sensors deliver positive float values that will serve as a meaningful data. If there is any discrepancy in the signal between sensor & receiver, there will be a negative value/ zero / ’NaN’ gets recorded in that place. Following graph presents the null value delivered by each sensor deployed on the pump



* From the graph given above its is understood that sensor 15 column has completely no values, hence it needs to be eliminated along with column 0 which is also empty

**Replacing ‘NaN’ with median values:**

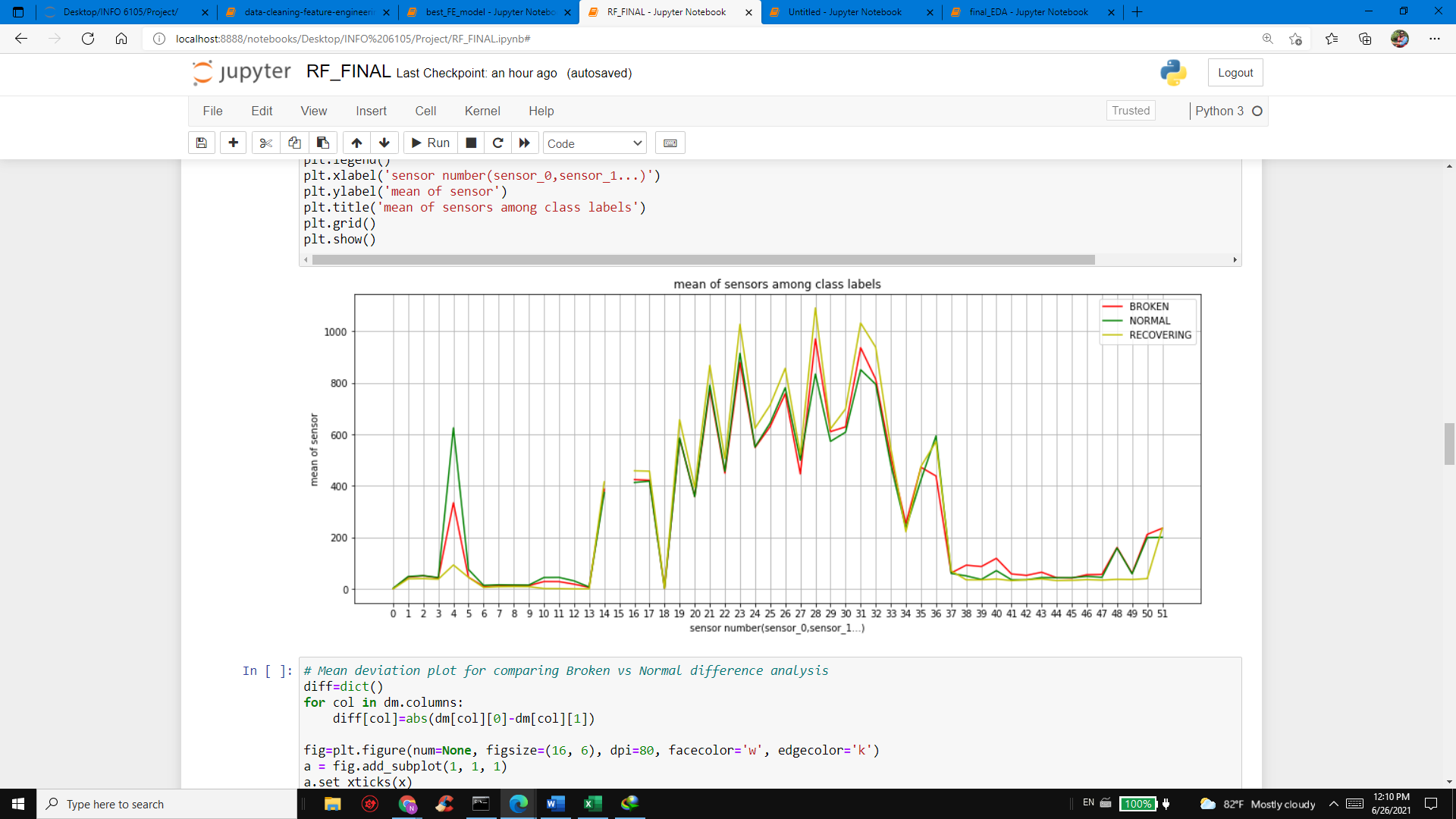


‘NaN’ values get generated when there is a signal distortion & hence these values needed to be voided or filled with some scalar values to further process the dataframe.

**Step 2: Exploratory Data Analysis & Further Processing**

In the previous steps we executed primary operations inorder to eliminate null & improper values, but here we had made our decision to analyze data statistically. Therefore, we are about to compute the mean deviation with respect to target class labels ‘BROKEN’, ‘NORMAL’ & ‘RECOVERING’. This analysis will help us to understand if there are any substantial deviation in the sensor values when compared against the target groups. Based on this we can make our data frame to a level where all the values are consistent enough to perform prediction

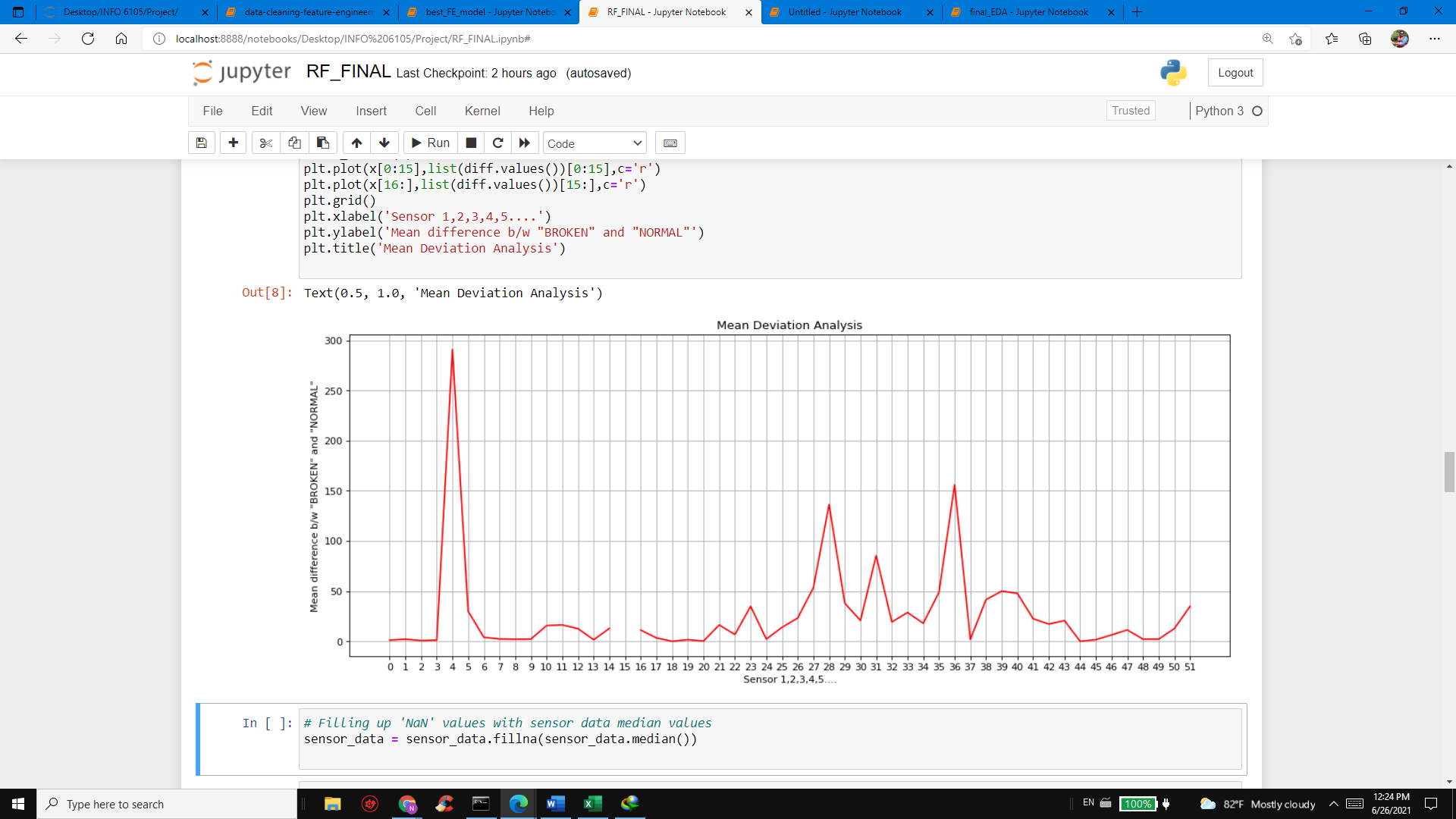
**Sensor label vs Mean of Sensor values**



It is understood from the above graph that the certain sensor values are said to be fluctuation above their mean values. Those are usually considered to be the impact of noise prevailing in the pump environment. Elements such as acoustical noise, external temperature, vibration generated by the pump motor & magnetic field created around the pump could lead to extreme noise generation, thereby causing instability in the data recorded.

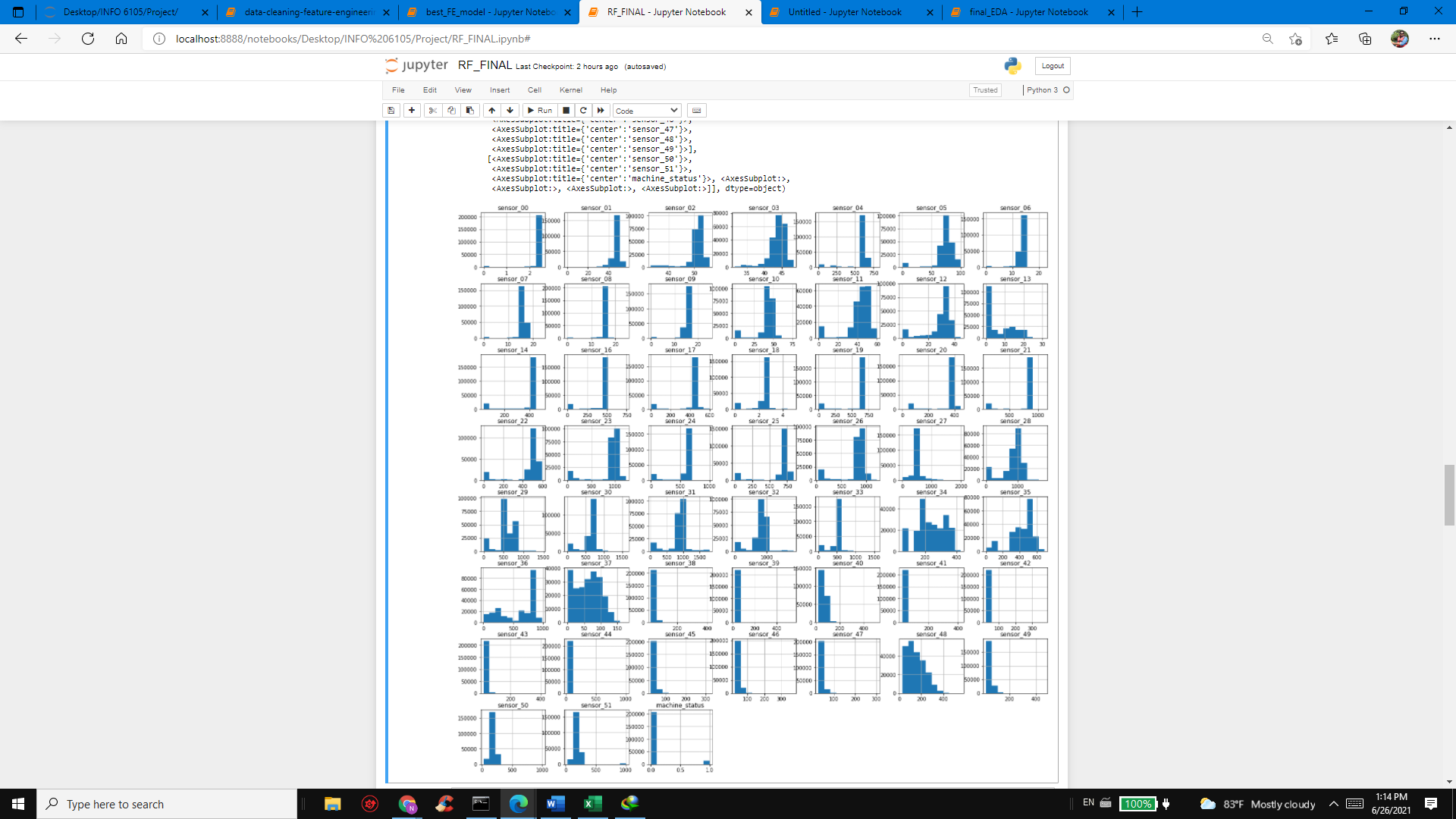
**Broken Vs Normal Label Mean Deviation Analysis**

Since our comparison is against the ‘BROKEN’ & ‘NORMAL’ classes we would compare it inorder to get clear understanding of this data distribution

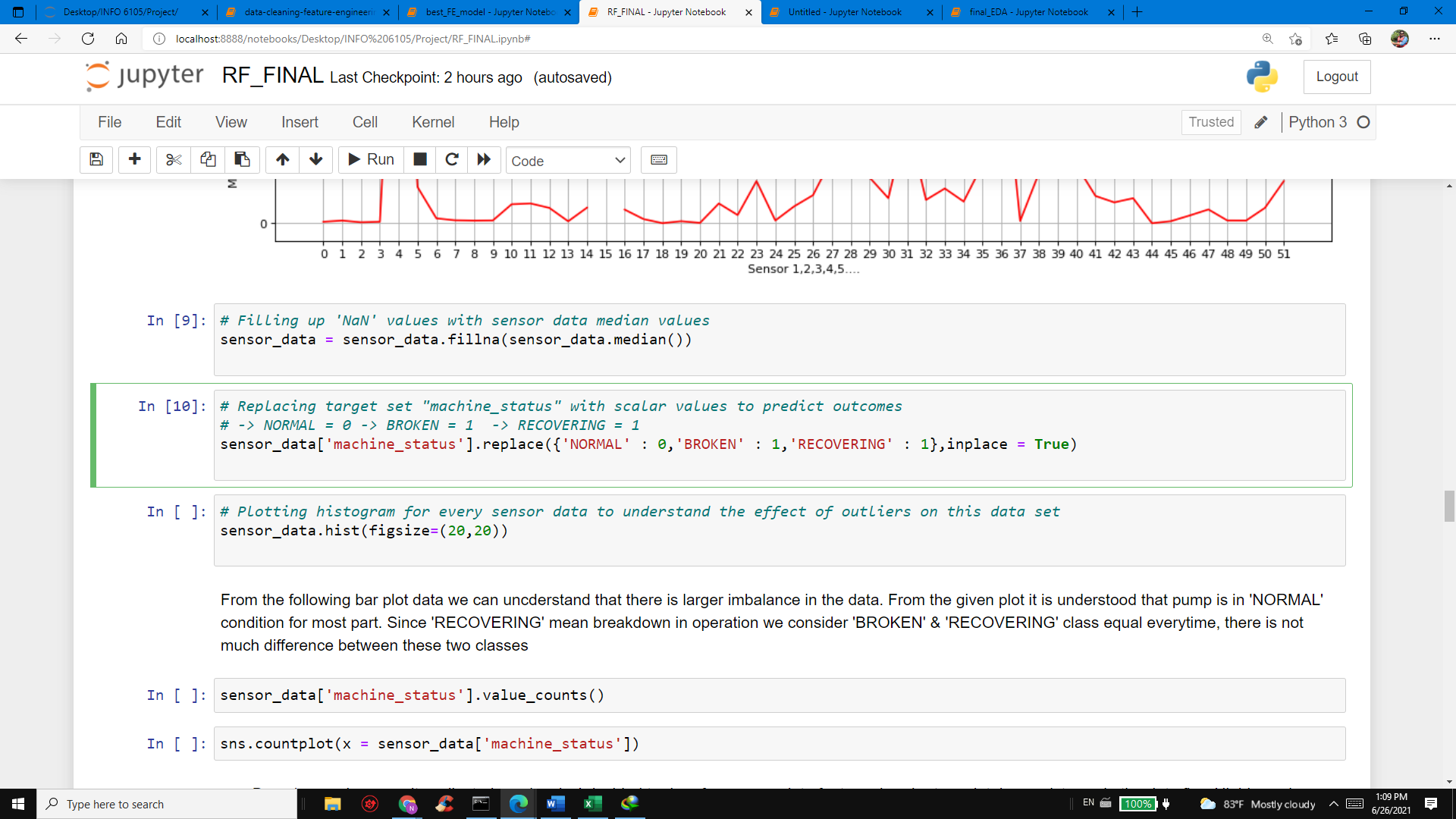


**Outlier Data Analysis:**

By plotting histogram for each sensor data against its mean values, we can understand the sensor value outliers for every sensor employed. Therefore, it allows us to take critical & effective decision on cleaning our datasets



**Target Data Modification:**



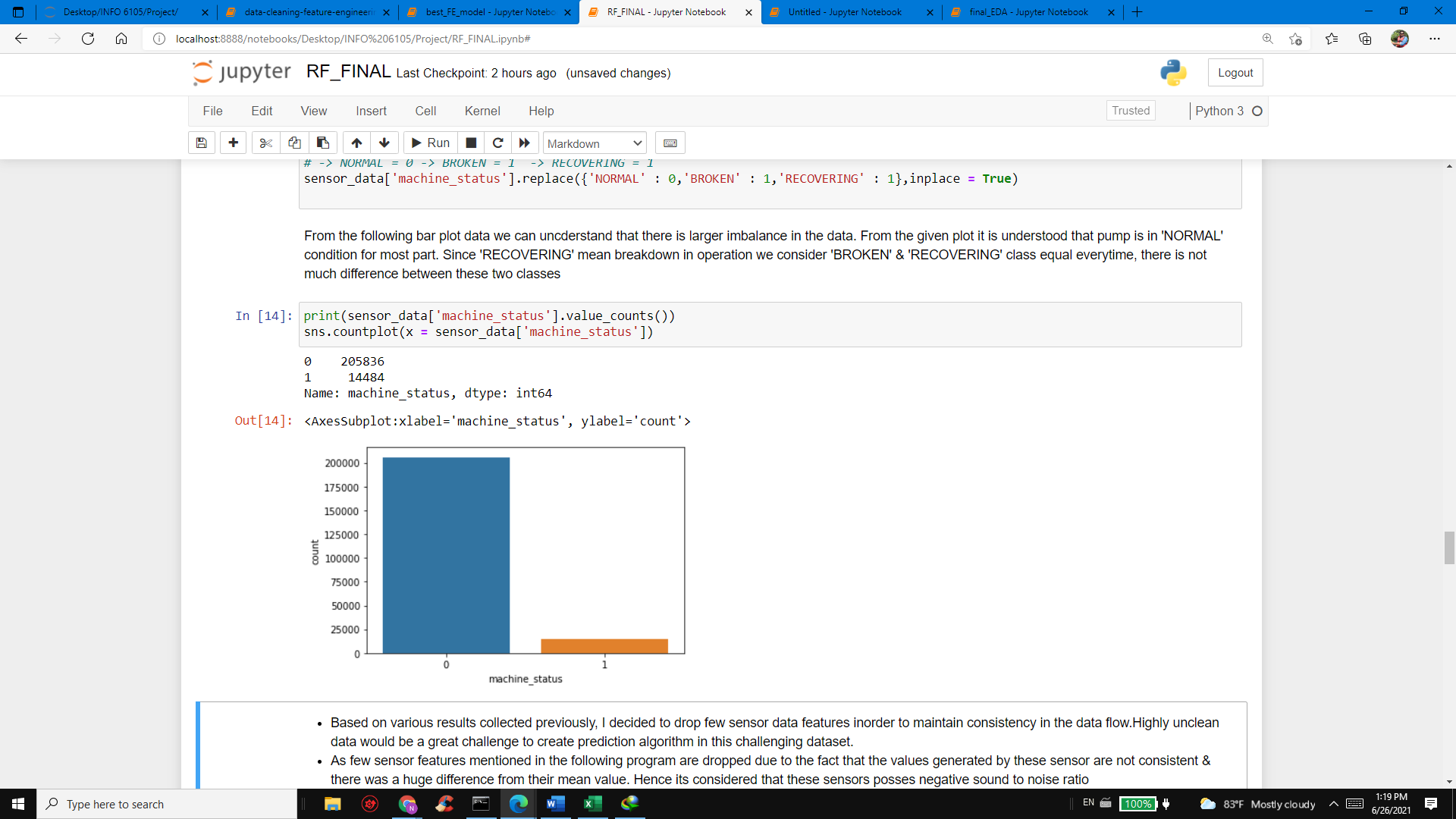
As our target is ‘machine\_status ’ column it is necessary to convert into numerical format for the purpose of machine learning modeling & predictive analytics operations. Hence the following changes are made

* ‘BROKEN’ = 1
* ‘NORMAL’= 0
* ‘RECOVERING’= 1

Since BROKEN & RECOVERING means the same, we consider both to be ‘1’ to make our predictive analysis less complicated

**Target Values Count:**

The values of the target data plays a crucial decision making in selecting machine learning algorithms for data processing. We counted & plotted the data using the predefined functions & observed that the probability that the machine would fail is much lesser, which will make our prediction nearly impossible. From this count it is understood that this is a challenging dataset which requires higher experimentation with different types of ML algorithms used in the industry.



**Machine Failure Analysis:**

As we are aware that machine went to ‘BROKEN’or fail state at 7 times in the toatal time span of four months. We wrote an algorithm to check how long does it take to recover after each subsquent failure event has occurred.

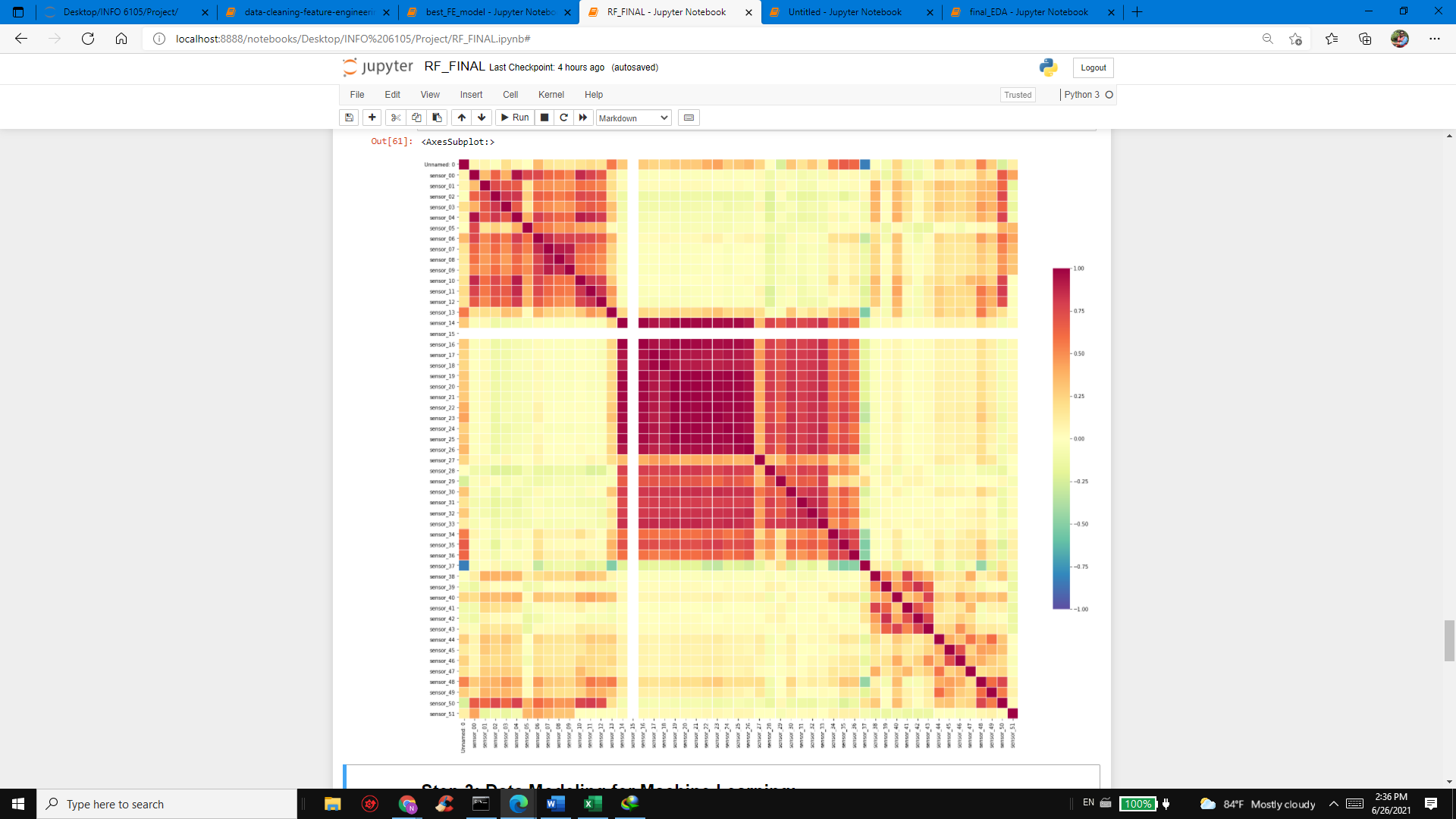


The above graph plot describes the coded algorithm in simpler terms. ‘BROKEN’state is always ‘1’ with whereas ‘RECOVERING’state is depicted as ‘-1’. So whenever the machine fails it hits ‘1’ & drops down to ‘-1’ inorder to go to a recovery state. We have also computed the time it has taken to recover & restore to normalcy for which we have given ‘0’ as the label

**Correlation Matrix:**

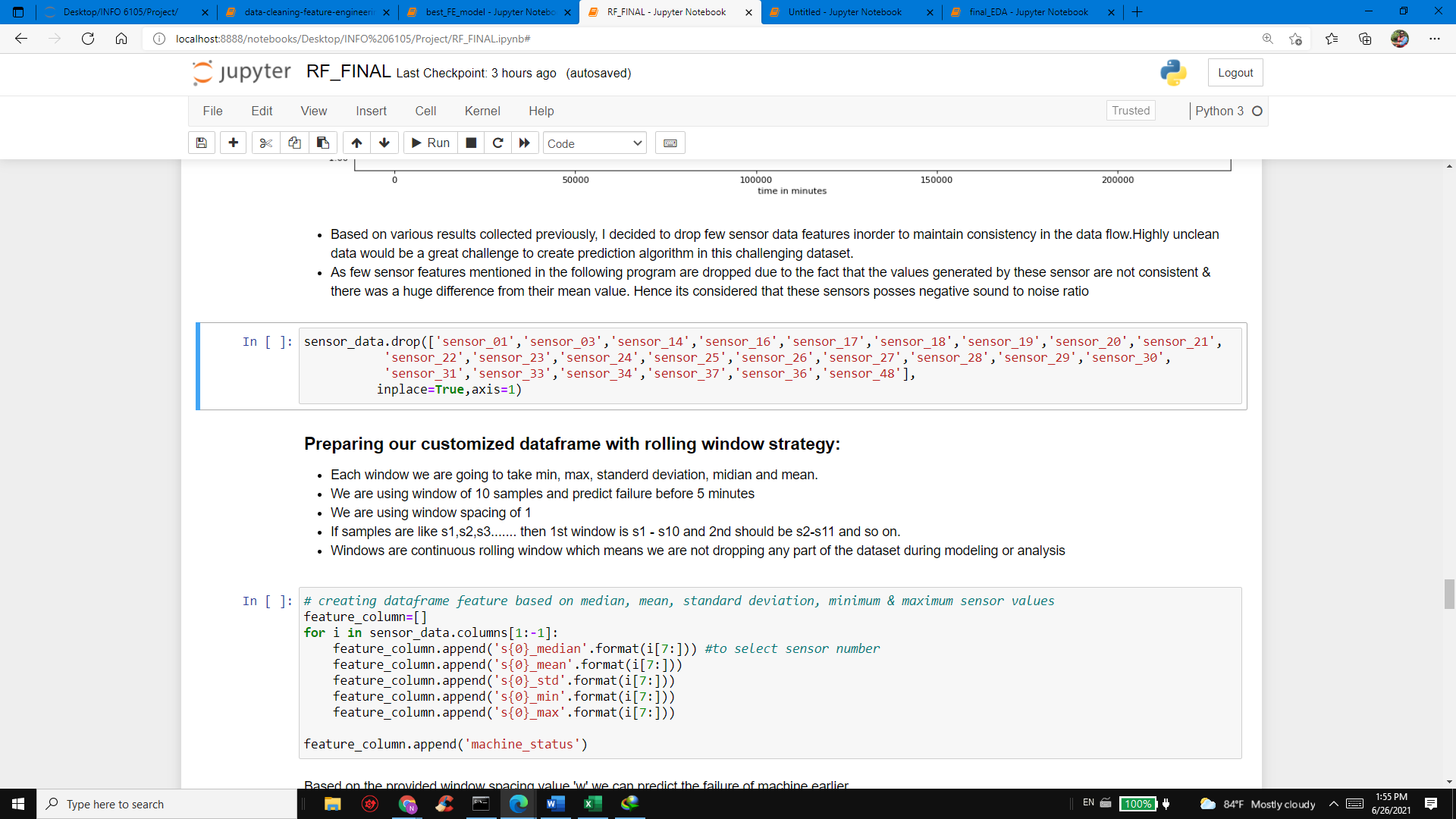
Correlation matrix play a major role in data processing, they determine how much a feature is dependent on another one. Higher the correlation value (greater than 0.7) would be considered as a linear pair. Here we take sensor number 14 has higher correlation with sensor 16 – 36 hence we need to consider it form elimination. Like wise each & every sensor feature must be weighed on their collinearity.

The following plot is a heat map for determining the collinearity in the given data,



**Final Decision:**

From the inference obtained from charts, histogram & failure data we conclude that the following sensors are deviating lot from their mean values, hence its decided that they need to be eliminated from the data frame



**Eliminated Features:**

sensor\_01','sensor\_03','sensor\_14','sensor\_16','sensor\_17','sensor\_18','sensor\_19','sensor\_20','sensor\_21','sensor\_22','sensor\_23','sensor\_24','sensor\_25','sensor\_26','sensor\_27','sensor\_28','sensor\_29','sensor\_30','sensor\_31','sensor\_33','sensor\_34','sensor\_37','sensor\_36','sensor\_49

**Creating data frame for our ML model**

Since we are trying to predict the failures earlier for the purpose to prevent failure & losses to the electrical assets commissioned in the industry. It is significant to create a data frame filled with meaningful values that can drive an optimal predictive analysis. Hence, we have adopted rolling window strategy to determine the predictions.

As we are predicting the machine status which is a time series data of t, t+1, t+2 …, with help of past data t,t-1,t-2 … its quite complicated to create a window which will be able to predict before 60 minutes. So, we choose a continuous rolling window size of 10 inorder to determine failures atleast by 5 minutes. Our final data frame will consist of rolling datasets which include mean, min, max , standard deviation, median for each qualified sensor data. We also use twice the amount of windows size (w) inorder to predict the failures early.

**Step 3: Machine Learning Model Development**

After performing multiple data analysis & filtering out the odds in the original dataset we are about to perform statistical techniques on the given dataframe created based on continuous rolling window ideology. As we identify this data set as a disproportionate / imbalanced set we need to identify suitable machine learning algorithm to analyze the outcomes.

Before a machine learning algorithm is chosen for analysis the data must be split into test & train factions. In this project we are going to take 50 % of the data for training & remaining 25% data for testing & cross validation. Using cross validation data for this dataset enhances the credibility of the ML model & utilization of cross validation datasets are mandatory here to make the prediction accurate. By this way we can double check the model using cv & test dataset. As we have labelled entities which have a specific target, we will be proceeding with supervised machine learning techniques.

Since our data is continuous time series data, it is quite easier to predict using the regression techniques. Also, at the same time the target output will be either ‘1’ or ‘0’ based on the calculations. So, let us analyze with various types of ML algorithms

1. **Logistic Regression – (Linear Model):**

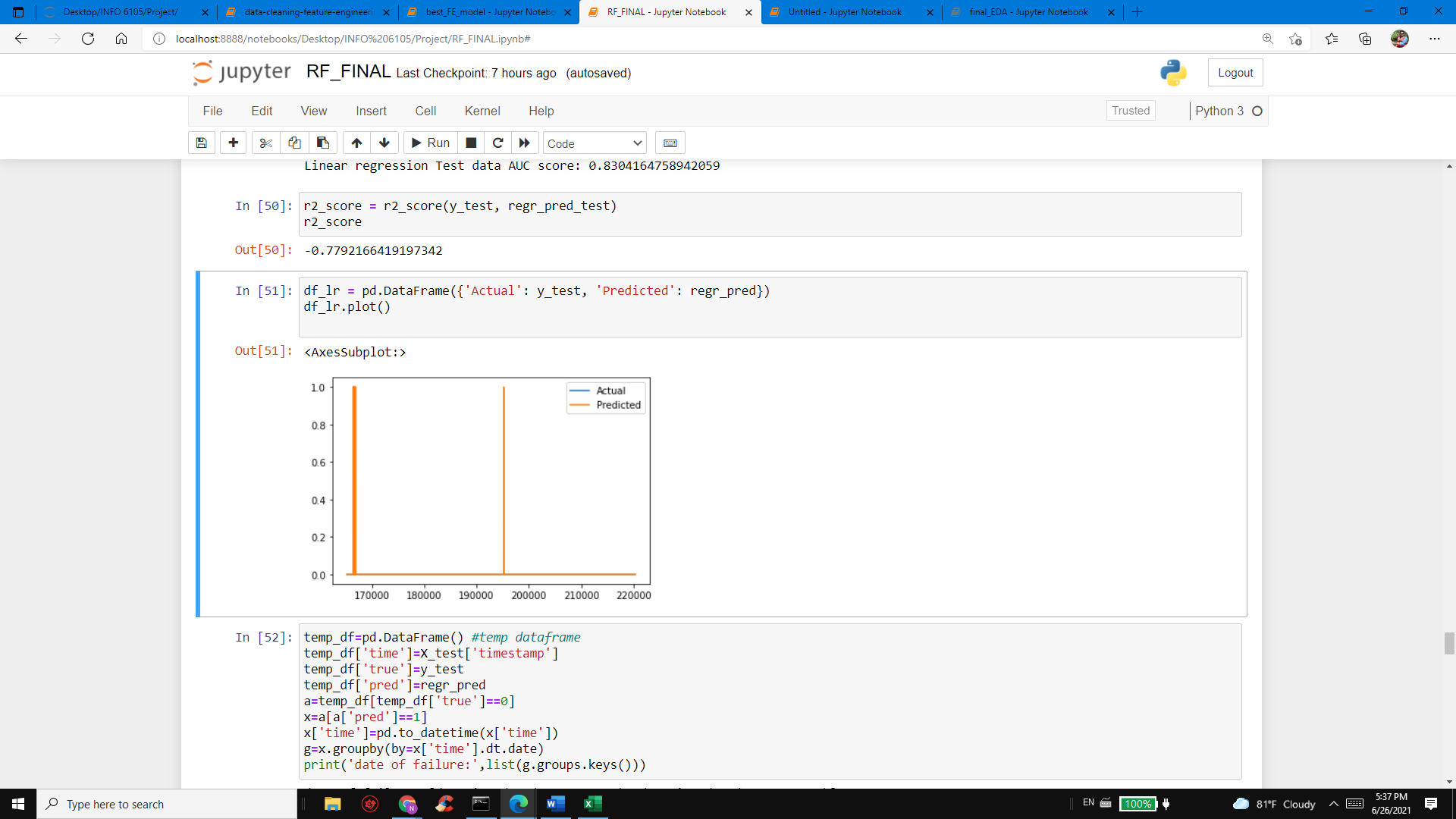
Though the name suggests that it’s a regression method, functionally this algorithm uses probability to predict ‘0’ or ‘1’ based on the independent values. Hence this algorithm falls under classification method. Logistic regression helps to eliminate odd values using simple formula of possibility of event occurring / possibility of even not occurring. This strong mathematical model could help us to segregate our ‘BROKEN’ (0) or ‘NORMAL’ (1) target values.

At first, we trained the model using the 50% train data & 25% for cross validation. And we delivered the following results,

Values Closer to 1, then the ML model accuracy is perfect

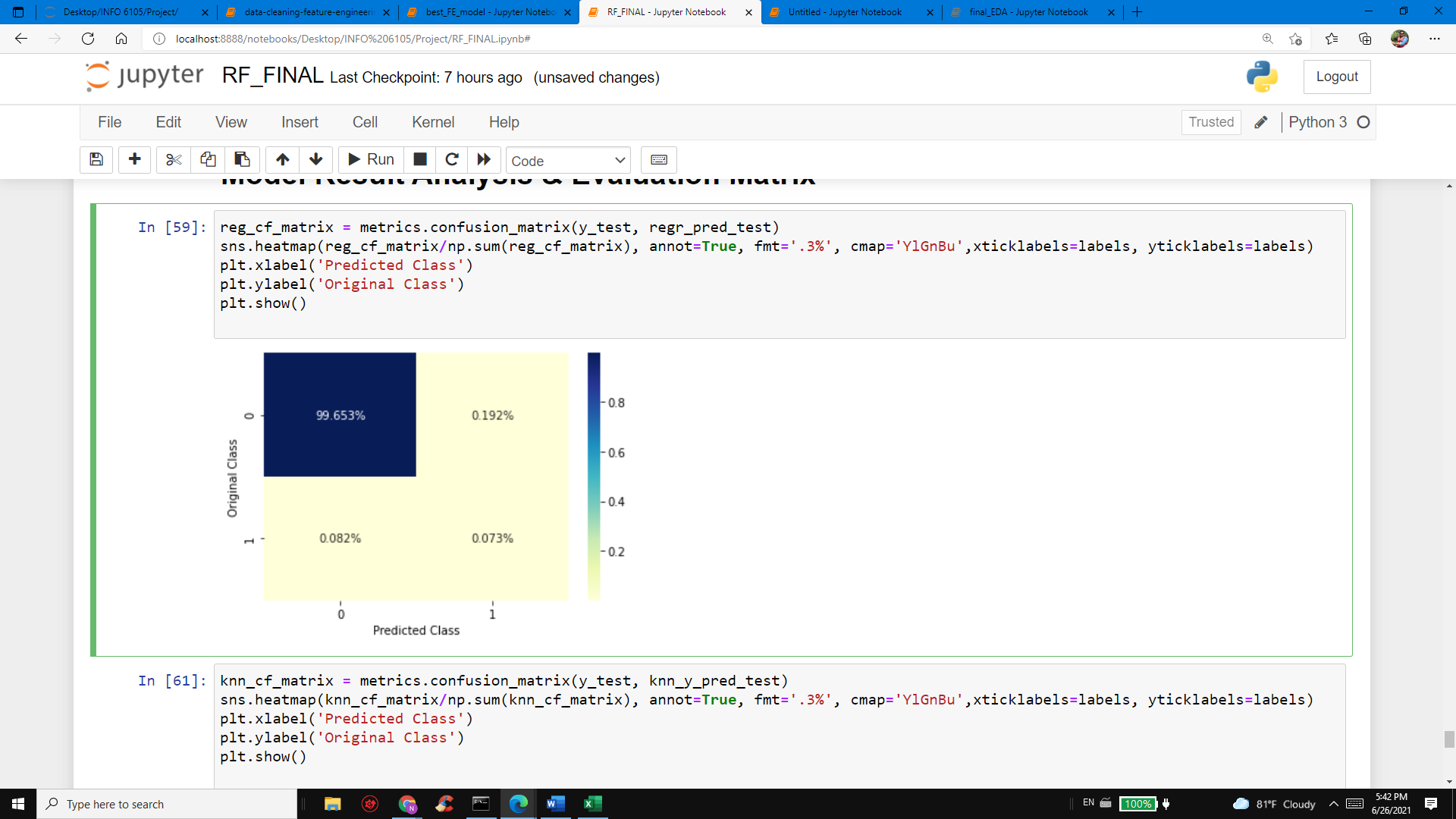
* AUC\_ROC SCORE FOR CV DATA = **0.99**
* AUC\_ROC SCORE FOR TEST DATA = **0.83**

In the following plot one can observe that the model was able to predict the values at high success rate, but inorder to have clear picture confusion matrix is required



**Confusion Matrix:**

Confusion matrix for logistic regression test data vs X\_test data, where we can observe that true positive values have major representation, whereas other parameters are very minimal



1. **K- Nearest Neighbor – (Non-Linear Model):**

K-nearest neighbors is non-parametric way used for classification & regression. It is a form of non-linear method where the decision made here are normally based on major number of closest points to the actual values. For example, we have a cluster of zeros & cluster of ones in a dataset. Now we want to get the prediction for our given data point x,y. The KNN algorithm will compute the closest point for the given target. By increasing its range with the predefined number of neighbors, the algorithm will develop a suitable value with respect to prediction. This is a preliminary classification algorithm which an engineer must before he handles convoluted neural networks & artificial intelligence.

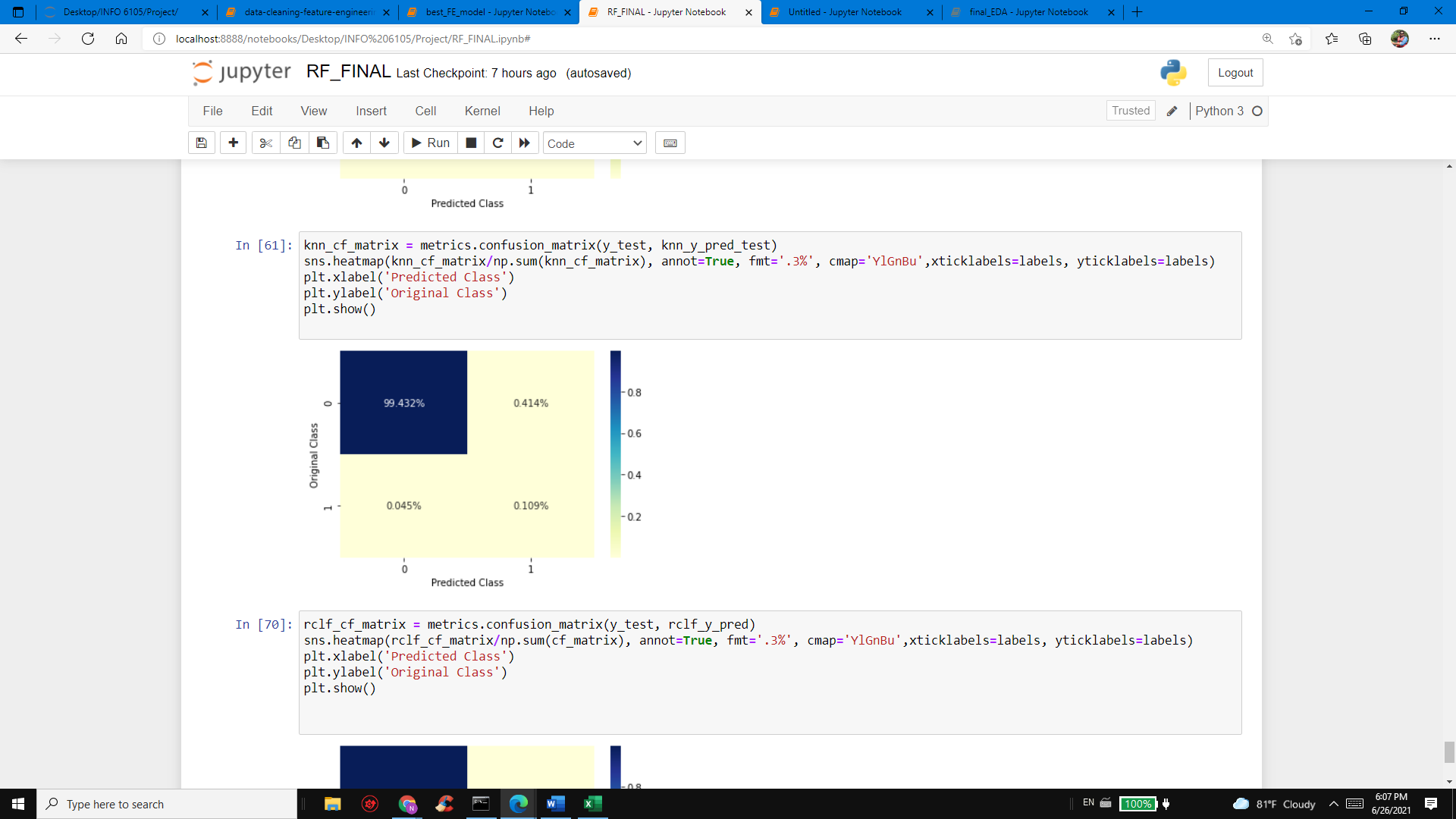
At first, we trained the model using the 50% train data & 25% for cross validation. And we delivered the following results,

Values Closer to 1, then the ML model accuracy is perfect

* AUC\_ROC SCORE FOR CV DATA = **0.90**
* AUC\_ROC SCORE FOR TEST DATA = **0.91**

**Confusion Matrix**

Confusion matrix for knn test data vs X\_test data, where we can observe that true positive values have major representation, whereas other parameters are very minimal



1. **Random forest Classification**

Random forest classifier is an ensemble algorithm which means that it combines one or more algorithms of same kind or different for classifying objects. For example, running prediction over linear regression, naïve bayes & then taking vote of final test object result to decide. This complex algorithm is quite useful for handling our dataset which is quite challenging. Random forest classifier uses multiple decision trees from the training dataset. It then aggregates the votes from these trees to decide final class of the test object.

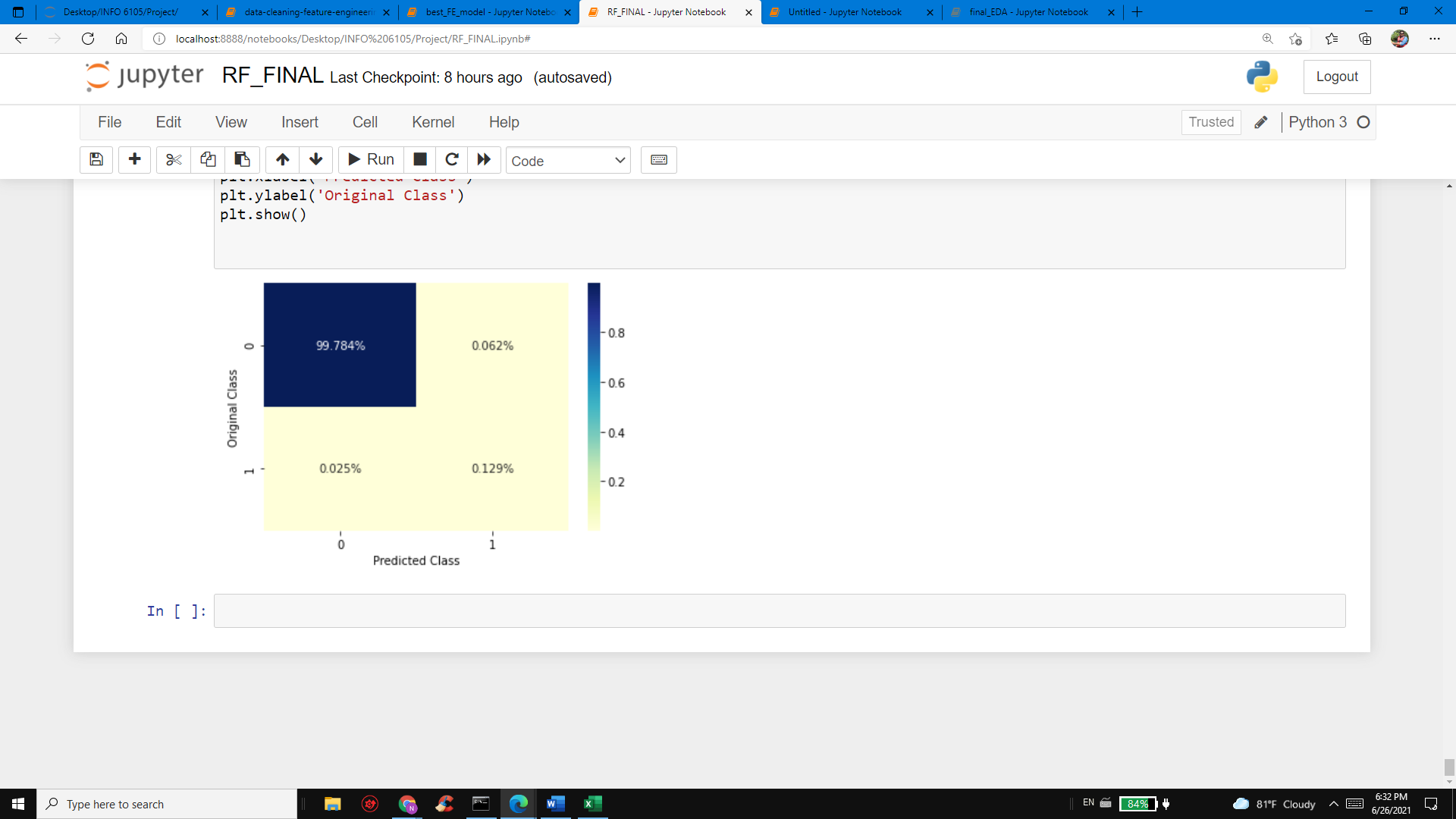
At first, we trained the model using the 50% train data & 25% for cross validation. And we delivered the following results,

Values Closer to 1, then the ML model accuracy is perfect

* AUC\_ROC SCORE FOR CV DATA = **0.99**
* AUC\_ROC SCORE FOR TEST DATA = **0.99**

**Confusion Matrix**

Confusion matrix for knn test data vs X\_test data, where we can observe that true positive values have major representation, whereas other parameters are very minimal



**RESULT & ANALYSIS:**

|  |  |  |
| --- | --- | --- |
| **ML Algorithm** | **AUC\_ROC SCORE** | |
|  | **CV-DATA** | **TEST-DATA** |
| Logistic Regression | 0.99 | 0.83 |
| K Nearest Neighbor | 0.90 | 0.91 |
| Random Forest | 0.99 | 0.99 |

Right from beginning, I quoted several times that this dataset is highly imbalanced & it needs a suitable ML algorithm to handle this problem. Though the previous linear & non-linear algorithm were giving great roc\_auc score it does not mean that the model is perfect. With Random forest classifier we were able to achieve nearest level of accuracy with aoc\_roc score of **0.99.** Hence, I would like to endorse “**Random Forest Classification”** model as the **best model** with low level of deviation.

Not only based on the auc\_roc score I give priority to random forest classifier, we have the following live model analysis data to prove my argument. Since our project meant for predicting the failures beforehand, we created a temporary data frame to hold false positive values & observed that at one failure point we were able to predict the machine status as broken before 5 minutes. There is another example where we were able to predict machine failure before 6 hours, that is due to fact with recovering time taken by the pump. That means, higher the time to recover the pump operation – earlier the prediction of failure.

In the next page I have compared the original data with predicted data which proves my argument of predictive analytics,

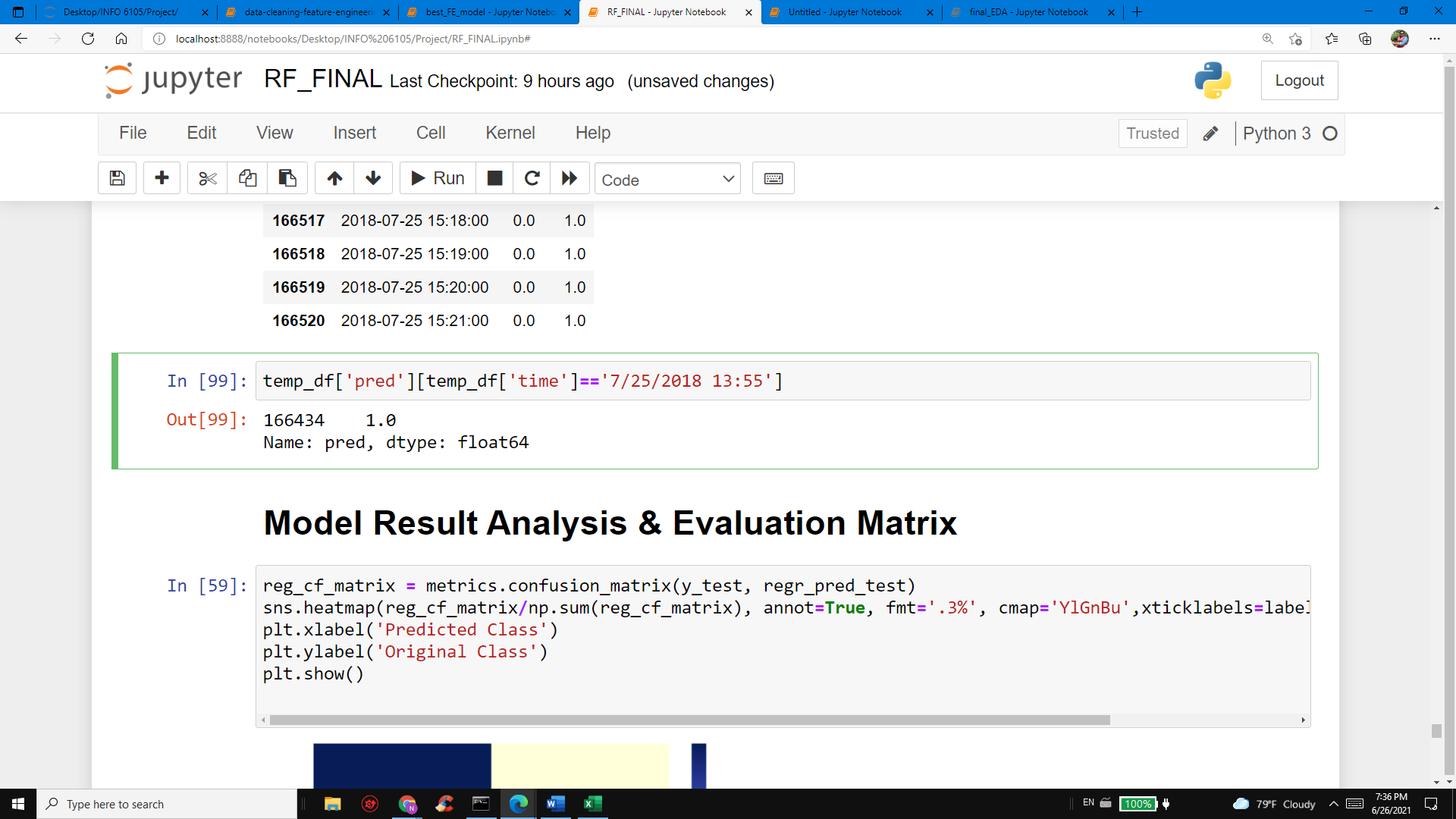
**Example 1:**

Here is the example of prediction before 5 mins,

* We had pump failure on 7/25/2018 @ 14:00 p.m.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 166435 | 7/25/2018 13:55 | | NORMAL | |
| 166436 | 7/25/2018 13:56 | | NORMAL | |
| 166437 | 7/25/2018 13:57 | | NORMAL | |
| 166438 | 7/25/2018 13:58 | | NORMAL | |
| 166439 | 7/25/2018 13:59 | | NORMAL | |
| 166440 | 7/25/2018 14:00 | | BROKEN | |
| 166441 | 7/25/2018 14:01 | | RECOVERING | |
|  |  | | |  | |

* Our prediction using random forest classifier algorithm we were able to see the predicted value ‘1’ which means pump is ‘BROKEN’ 5 minutes earlier



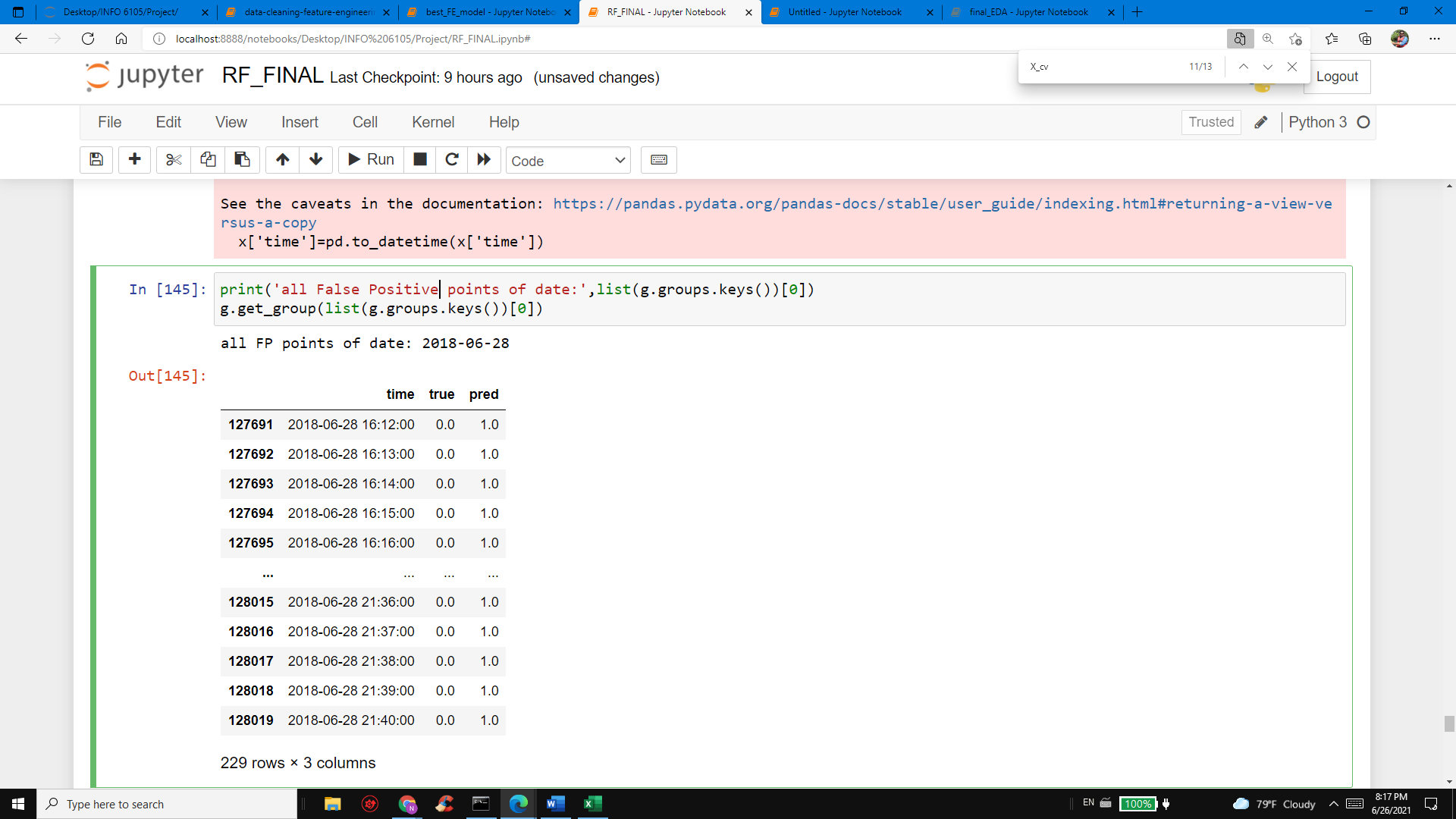
**Example 2:**

Here is the example of prediction before 6 hours,

* We had pump failure on 06/28/2018 @ 22:00:00 HRS,

|  |  |  |
| --- | --- | --- |
|  | 6/28/2018 21:58 | NORMAL |
| 128039 | 6/28/2018 21:59 | NORMAL |
| 128040 | 6/28/2018 22:00 | BROKEN |
| 128041 | 6/28/2018 22:01 | RECOVERING |
| 128042 | 6/28/2018 22:02 | RECOVERING |
| 128043 | 6/28/2018 22:03 | RECOVERING |
|  |  |  |

* Our prediction using random forest classifier algorithm we were able to see the predicted value ‘1’ which means pump is ‘BROKEN’ 6 hours earlier (i.e) from 16:12 HRS same day. Since the severity of damage is high, we were able to predict much earlier



**Conclusion:**

Though the results were satisfactory using various famous machine learning algortithms, we were able to predict the failures only before few minutes to hours. This will not be good enough in actual use case scenarios. So inorder to have clear forecast of failures, we need to adopt deep learning & advanced ML techniques to deliver realtime promising results. These kind of scientific advances will benefit companies only if they are successfully deployed in the market, so better alternatives for a machine learning model should be always present. If one model is 99% successful , engineers must work further to achieve 100% resiliently without stopping. Even ignoring smaller portion error makes any ML model worthless.

**References:**

1. <https://www.kaggle.com/adithyajere/pump-dataset>
2. <https://www.kaggle.com/artgor/simple-eda-and-models>
3. <https://machinelearningmastery.com/random-forest-for-time-series-forecasting/>
4. <https://neurospace.io/blog/2019/05/predicting-machinery-breakdown-on-a-water-pump/>
5. <https://medium.com/codex/machine-learning-logistic-regression-with-python-5ed4ded9d146>
6. <https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea>