Indian Institute of Technology Delhi

ELL784 Introduction to Machine Learning: Assignment 1

Mohammed Jawahar Nausheen: 2019CS10371

Himanshu Nigam: 2021EET2828

Deepak: 2019MT10685

Pre-processing – Making the data ready for use

The data provided consists of 7 columns named metaOne, ts, sensorThree, sensorFour, sensorFive, sensorSix and sensorSeven. The columns metaOne and ts were found to contain the time and date of the data recorded. For further analysis they were converted into numerical types using datetime helper methods in pandas.

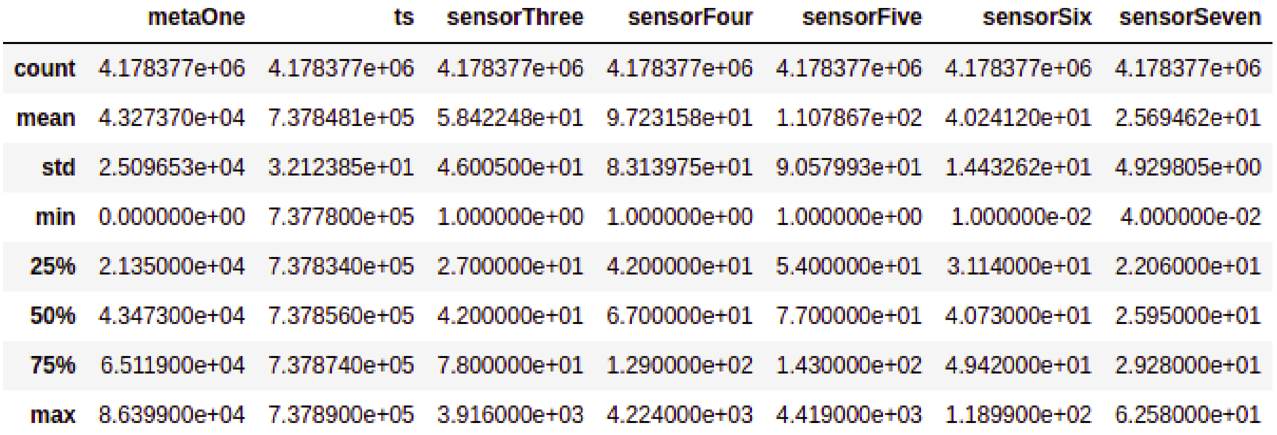
Handling missing values

The data was found to contain several missing values. Missing values are usually dealt with in the following ways.

- Dropping the rows that have missing data. This is the simplest as well as one of the robust ways to deal with missing data.
- Imputing the missing data with column mean. This way, missing data can also be used for training and testing but since this does not account for the variance, it may give worse results when the missing proportion is high.
- Interpolation with time axis. This way is suitable for data that is temporal, i.e, shows dependence with time.

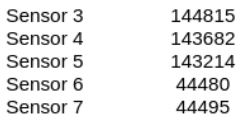
We chose to use the first method since the dataset provided is quite large and dropping 0.04% of it is manageable.

Numerical Summaries



	metaOne	ts	sensorThree	sensorFour	sensorFive	sensorSix	sensorSeven
count	4.178377e+06	4.178377e+06	4.178377e+06	4.178377e+06	4.178377e+06	4.178377e+06	4.178377e+06
mean	4.327370e+04	7.378481e+05	5.842248e+01	9.723158e+01	1.107867e+02	4.024120e+01	2.569462e+01
std	2.509653e+04	3.212385e+01	4.600500e+01	8.313975e+01	9.057993e+01	1.443262e+01	4.929805e+00
min	0.000000e+00	7.377800e+05	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e-02	4.000000e-02
25%	2.135000e+04	7.378340e+05	2.700000e+01	4.200000e+01	5.400000e+01	3.114000e+01	2.206000e+01
50%	4.347300e+04	7.378560e+05	4.200000e+01	6.700000e+01	7.700000e+01	4.073000e+01	2.595000e+01
75%	6.511900e+04	7.378740e+05	7.800000e+01	1.290000e+02	1.430000e+02	4.942000e+01	2.928000e+01
max	8.639900e+04	7.378900e+05	3.916000e+03	4.224000e+03	4.419000e+03	1.189900e+02	6.258000e+01

Missing values

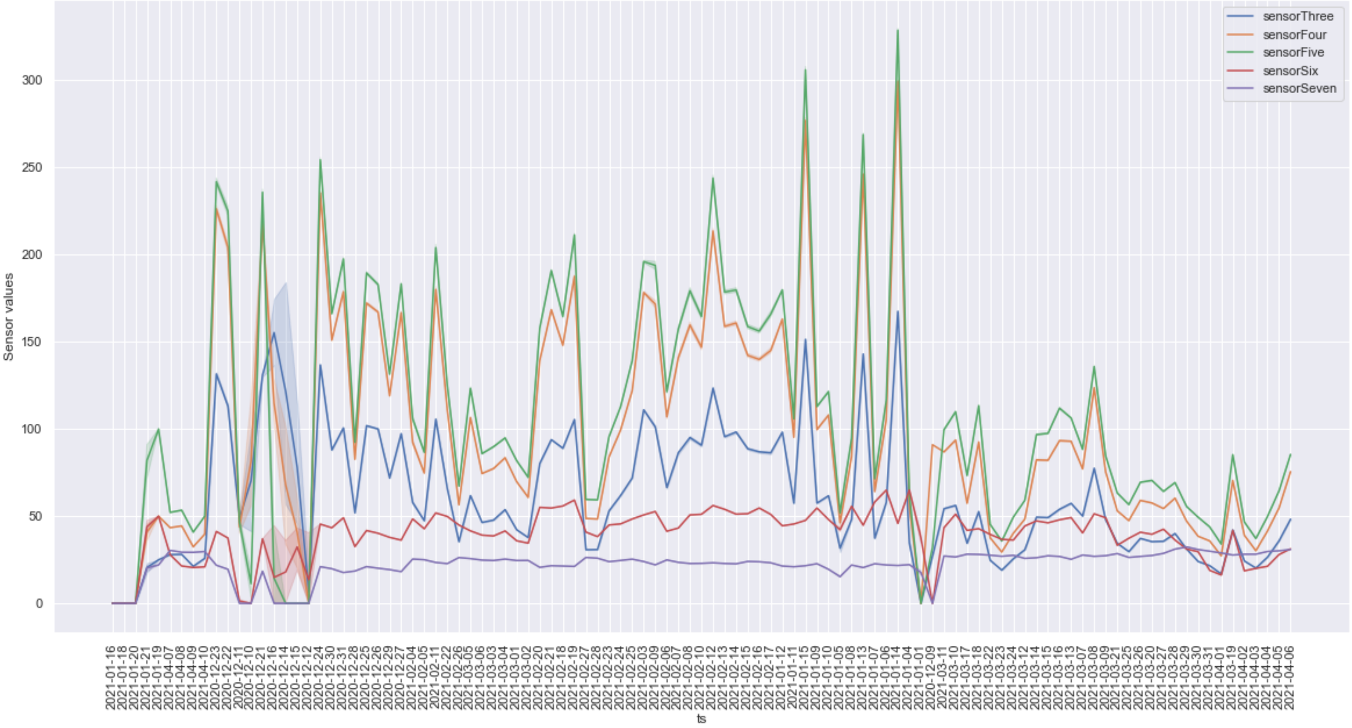


Sensor 3	144815
Sensor 4	143682
Sensor 5	143214
Sensor 6	44480
Sensor 7	44495

Graphical summaries

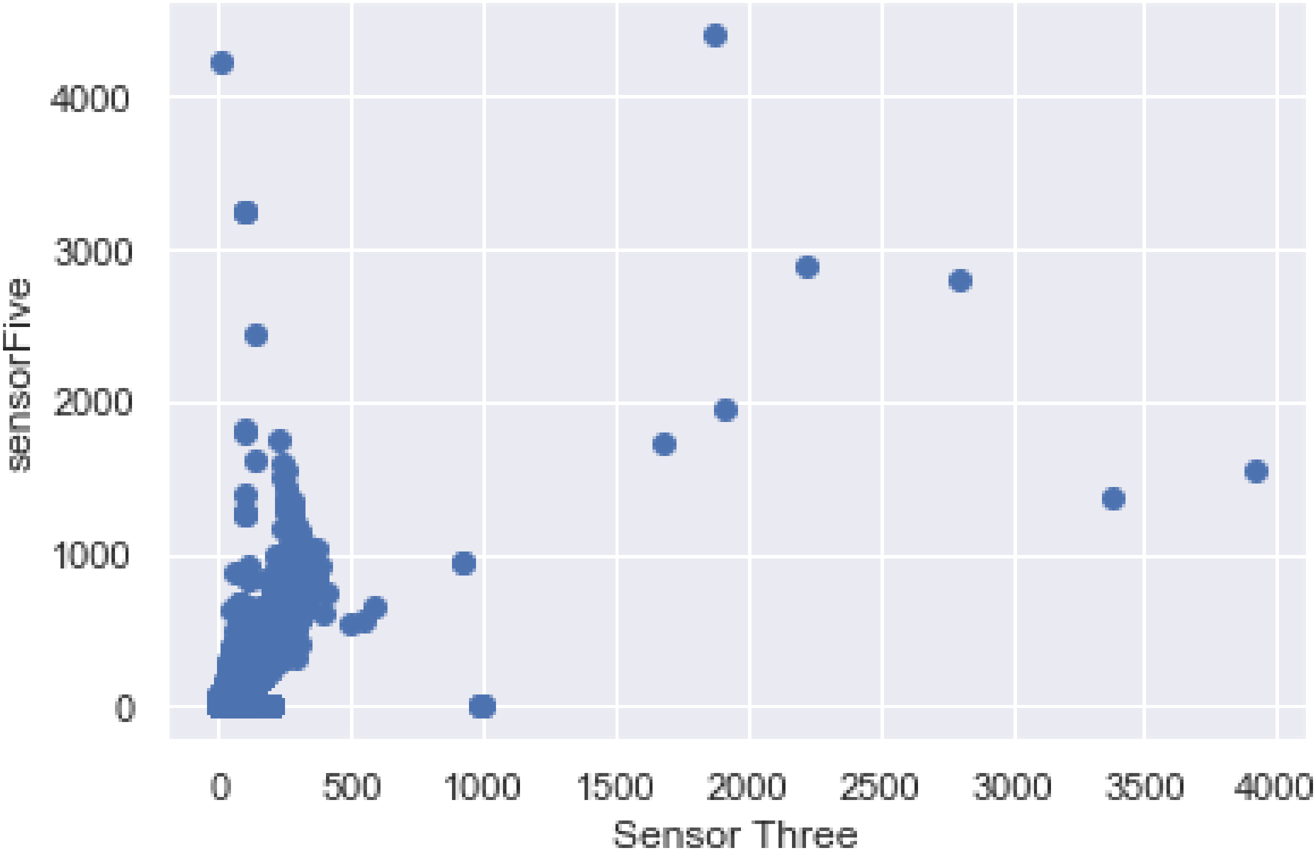
Plots were drawn with each sensor on the y-axis and time on the x-axis. It was observed that there isn't much dependence in the sensor values with time, apparent from the blanket like plots obtained. However there is a strong linearity in sensorThree and sensor Four. From this it is likely that if one of them depends on some feature then the other also shows a similar dependence. Also a positive correlation can be observed in sensor5 vs sensor3 or sensor4 plots.

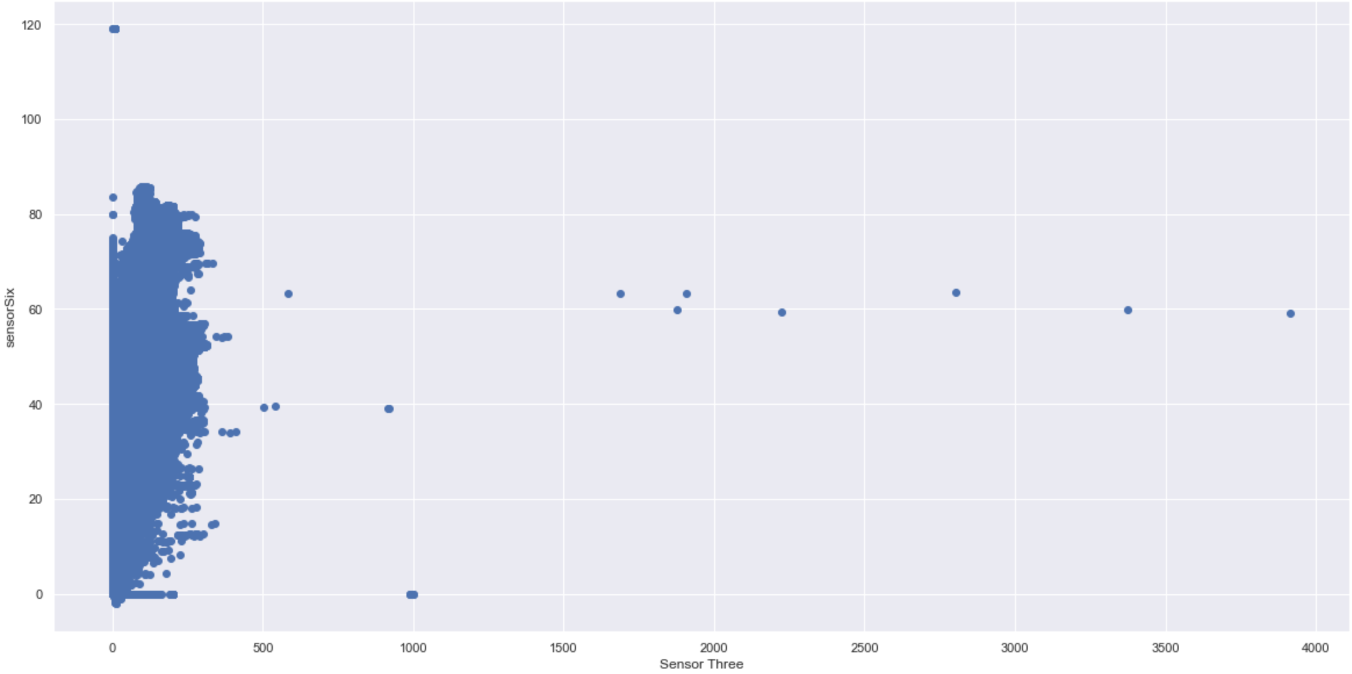
1. Sensors with respect to time interval

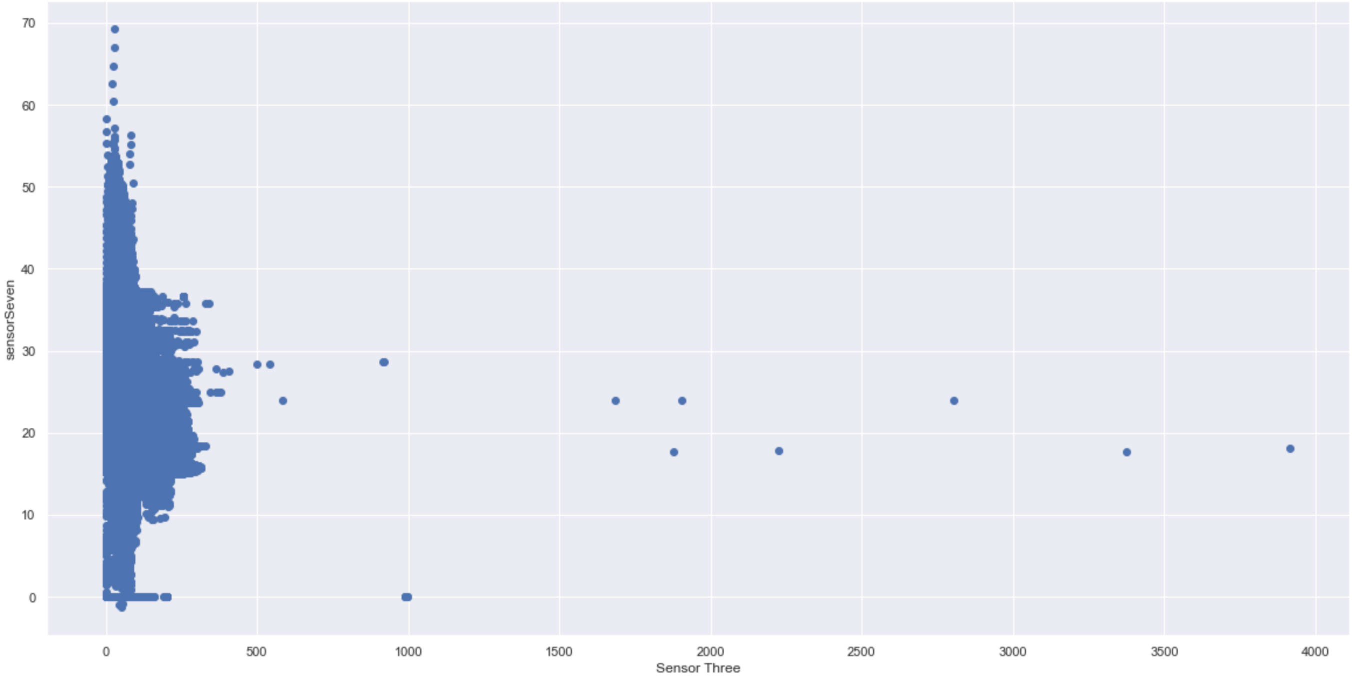


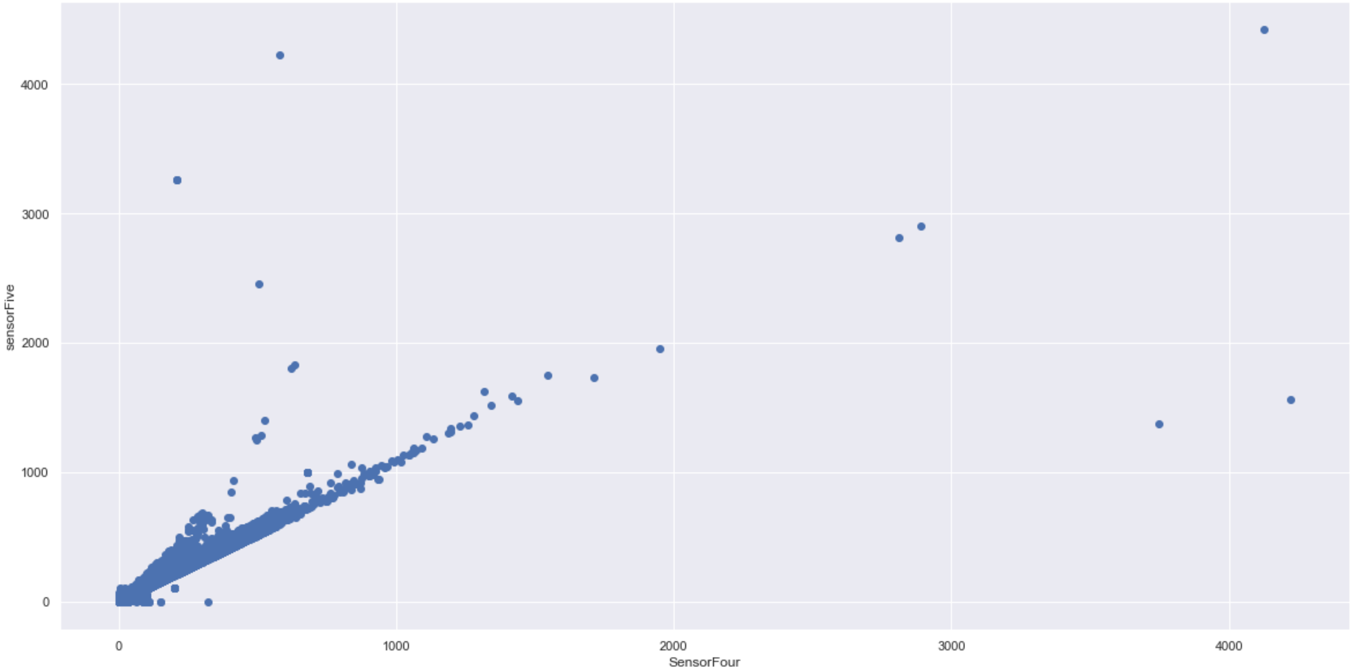
Deductions: The line graph plotted between Time and Sensors clearly shows some similarity in pattern between Sensor three, four and Five whereas no clear pattern between sensor six and seven.

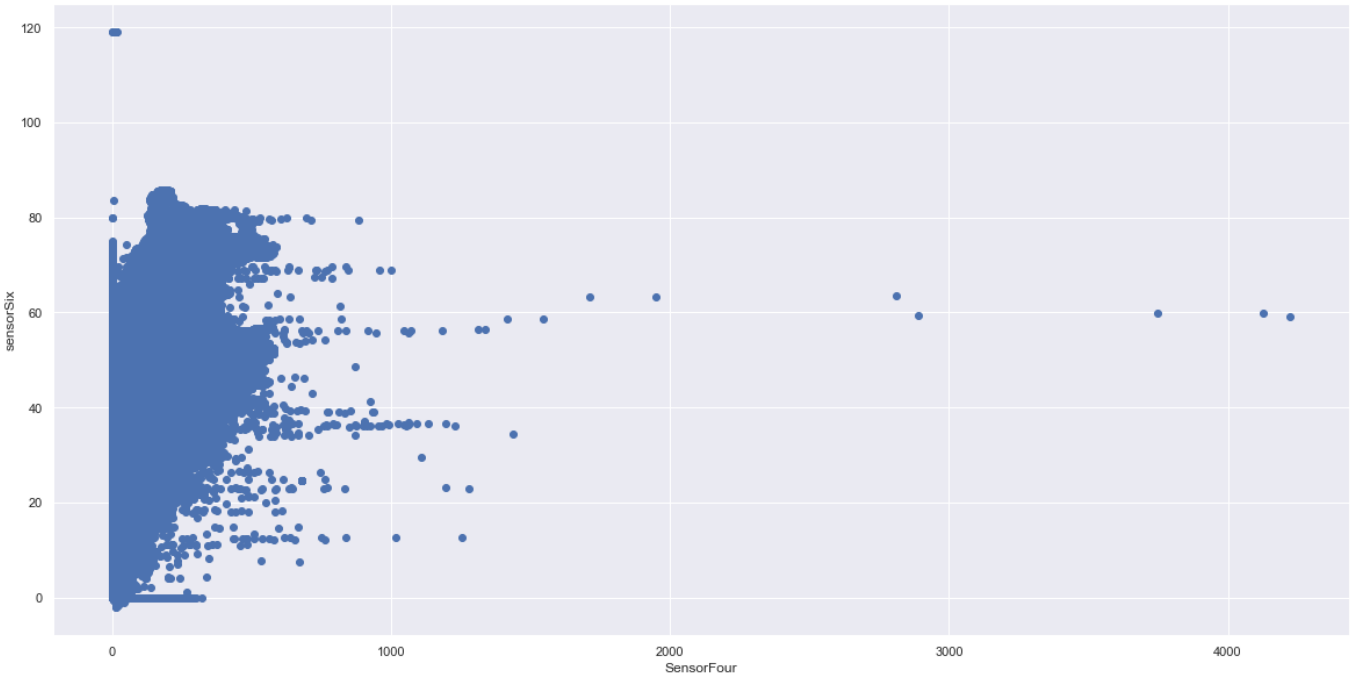
2. Sensors Three and Four will further be plotted against Sensors Five,Six , and Seven and further analysis will be carried out.

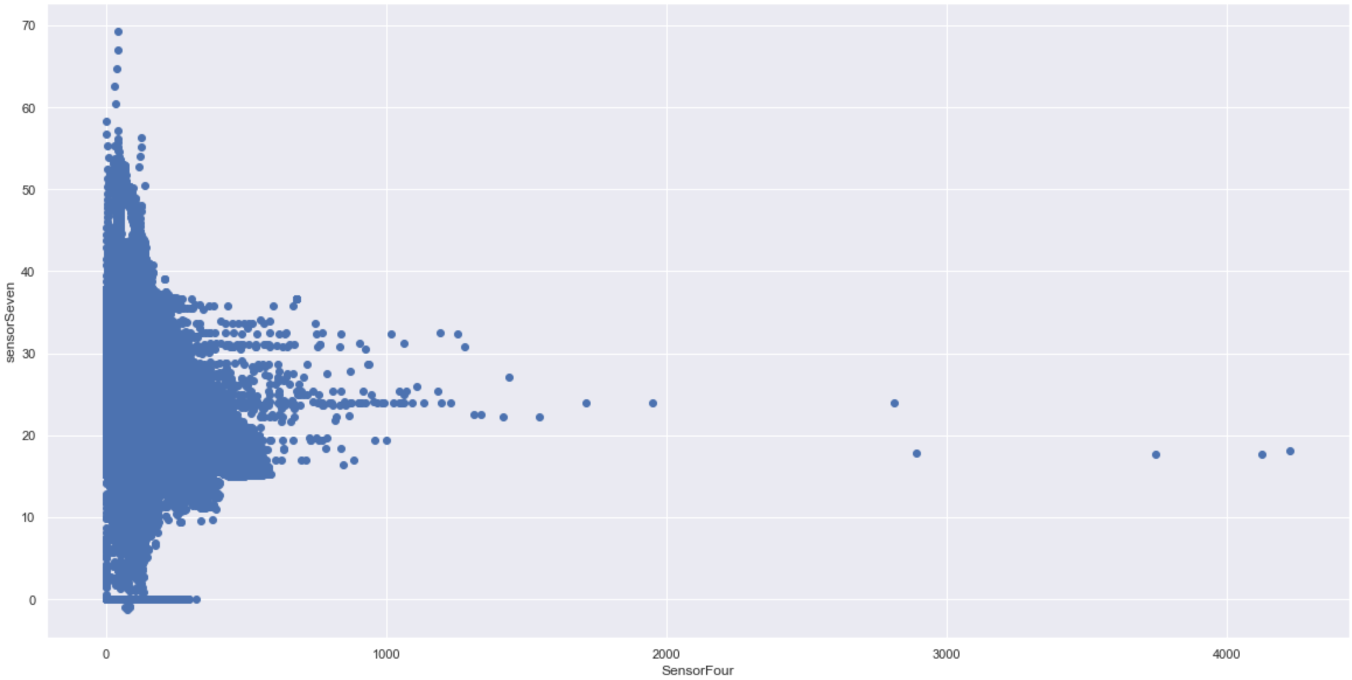

sensorFive
4000
3000
2000
1000
0
0
500
1000
1500
2000
2500
3000
3500
4000
Sensor Three


sensorSix
120
100
80
60
40
20
0
0
500
1000
1500
2000
2500
3000
3500
4000
Sensor Three


sensorSeven
Sensor Three
0
10
20
30
40
50
60
70
0
500
1000
1500
2000
2500
3000
3500
4000


sensorFive
SensorFour
0
1000
2000
3000
4000
0
1000
2000
3000
4000





Deductions Some relationship can be seen between Sensor Three and Sensor Five but no pattern can could be seen betwee sensor three , sensor six and sensor seven.

A strong pattern can be seen between sensor four and sensor five but no such pattern can be seen between sensor four ,sensor six and sensor seven.

Plotted Graphs also shows presence of outliers.

Statistical Significance of predictors

We computed p-values of the predictors using OLS regression from statsmodels.api. The obtained p-values of all predictors except sensorSix is 0 which means that null-hypothesis is rejected and the predictors are statistically significant. The p-value of sensorSix was not zero. However, its value was also less than 0.05 and therefore sensorSix is also statistically significant.

Scaling and normalization

Scaled features can greatly reduce the computation involved in many Machine Learning models. We employed two kinds of scaling methods, Minimax Scaling and Standard Scaling.

- In Minimax scaling, the feature x is converted into (x – min)/ (max - min) thus mapping the feature to [0,1].
- Standard scaling is quite similar, but it uses mean and variance instead of mean and max. A feature x is converted to (x - mean)/variance.

The effect of scaling is more pronounced in the results of non-linear models. However, since we consider linear models for the dataset at hand, there aren't many differences in the evaluations observed later, although scaled features must have reduced the computation made in the fit method.

Normalization is another method that helps scale down features and observe better relationships with other features. In normalization, the input vector is transformed such that it has a unit norm. This brings a quadratic nature to the problem, which is not very suitable if we wish to find linear relationships between data. Indeed, we observed a poor evaluation when normalized features were fed into the model.

Feature selection and dimensionality reduction

When there are multiple dimensions to the input, most machine learning algorithms find it difficult to scale. Thus, we select the few features which give the most contribution in predicting the result. Another reason to do this is that most real-world data has relatively low degrees of independence. Therefore, we evaluate feature significance using some prevalent ways, and select the most significant ones.

- F scores: Using the selectKbest method from sklearn, we evaluate features based on the F scores. F score for a feature is high when their correlation with the target is high. For both sensorThree and sensorFour, the values indicate that there is a strong correlation with sensorFive in comparison to any other feature.
- Recursive Feature Elimination: The least significant feature is pruned until the required number of features is attained.

It was observed that using only the sensorFive data to predict results, resulted in just 13% change in MSE, and almost no change in MAE, R^2 values. Also , from the linear model predicted, coefficients for the features

Models and loss functions

Three different linear models were employed. The simplest being Linear Regression, and the other two related regularised models being Ridge Regression and Lasso.

- Linear Regression: The model estimates w such that the sum of squares in the estimated and predicted values is minimum. This is the simplest model with no regularisation.
- Ridge Regression: The model estimates w such that the sum of squares in the estimated and predicted values along with a L2 regularisation term. It takes a hyper-parameter alpha. Regularisation limits the coefficients from taking high values.

- Lasso : Lasso uses a L1 regularisation term instead of L2 as in Ridge regression. Coefficients obtained using L1 regularisation are sparser than those obtained with L2 regularisation.

Model Evaluation

Multiple metrics can be used to evaluate a model's performance. The dataset is divided into training and test in 70:30 ratio randomly. Then metrics are calculated for both training and test data. Some popular ones that were employed are

- Mean Absolute Error (MAE): Mean of the absolute difference between target and predicted values. The MAE is observed to be about 4.33 in each of the three models.
- Mean Squared Error (MSE): Mean of the squared differences between target and predicted values. The MSE is observed to be about 50.3 implying a RMSE of 7.1.
- Coefficient of determination or R^2 score: The R^2 score is a popular metric calculated as 1 - (Residual sum of squares/ Total sum of squares). The R^2 score falls between 0 and 1. A higher R^2 score implies the variance in target can be well described by the input. The R^2 score is found to be 0.98 which is a very good score.

Cross Validation: The above computations are done after making a train and test split. But evaluating one such split is not fair since other splits may perform better or worse. This is solved by taking numerous train-test splits using cross validation and averaging over them. This provides a more wholesome evaluation of the models.
When averaged over multiple splits using cross validation, the metrics are found as

MAE – 4.33, MSE – 50.1, R^2 score – 0.95

Further it was observed that the test MSE is slightly higher than training MSE. Also, the R^2 score is higher for training than testing.