**CAPACITY ESTIMATION OF LITHIUM-ION BATTERY**

1. **Business Problem**
   1. **Description:**

Battery is a combination of various electrochemical cells that generate electricity by converting chemical to electrical energy. One such class of batteries is Lithium-ion battery. Lithium-ion batteries have long life, high specific power and high energy density and hence they are widely used in Electric Vehicles, Electronic gadgets etc. But the proper functioning of these batteries warrants extensive monitoring of health so as to avoid any hazards and make them long lasting. Such a technique used in the space of batteries is called Prognostics and Health Management (PHM) of batteries. With PHM it is possible to predict any failure of the batteries well in time so as to take appropriate preventive or corrective measures.

The part of the PHM that monitors all the parameters of the batteries is called the Battery Management System (BMS). One such parameter that is monitored by the BMS is called the Capacity of the battery or the State of Health (SoH). State of Health of a battery is defined as the ratio of the present maximum Charge in Ah after a full charging cycle to the rated initial maximum Charge of the battery in Ah. For example, if the battery had an initial rated maximum Charge of 100 Ah and over a period of time the maximum Charge that can be held by the battery is 80 Ah then the Capacity of the Battery is 80/100 = 0.8. But capacity of a battery cannot be measured directly from its terminals.

* 1. **Problem Formulation:** The task here is to predict the instantaneous capacity of the battery given current, voltage, temperature values sampled at certain intervals from the charge/discharge cycles. In the dataset, there is an impedance cycle which is given but we have not considered it in our modelling because impedance measurement is not possible online in any Electric vehicle or Electronic gadgets etc. as it requires instruments such as electrochemical impedance spectroscope.
  2. **Definitions:**

**Capacity**: How much maximum charge that a battery can hold at any given charging cycle instant.

**State-of-Health (SoH):** The SoH at time ‘t’ of a battery is defined as the ratio of the Capacity of a battery at time ‘t’ to the rated maximum capacity of the battery.

**Charge cycle**: The charging of Lithium-ion battery is carried out at Constant Current Constant Voltage mode. In this mode the battery is charged at constant current till the voltage increases to a particular value and after that the charging is switched to constant voltage mode where the voltage is kept constant at a value and the current is gradually decreased to almost 0A.

**Discharge cycle**: The discharging is done by connecting the battery to a load. During the discharge cycle the current is drawn at a constant value, during this period the terminal voltage of the battery keeps decreasing and the discharge of the battery is stopped when the battery voltage reaches a cut-off value.

1. **Machine Learning Problem:** 
   1. **ML formulation of the business problem:** The capacity of a battery cannot be directly calculated from the terminals of battery but it involves complex calculations to figure out the same. However, we can take help of machine learning techniques to accurately predict the capacity given the various voltage, current and temperature measurements both for the charge and discharge cycles.

In this technique, we calculate certain time duration from the current, voltage and temperature curves from the charge and discharge cycles and try to predict the value of the instantaneous Capacity of a Lithium-ion battery.

The Capacity of a battery is a continuous random variable and hence the current ML problem is a regression problem.

* 1. **Business constraint:**
* There is no sub-second latency constraint however training should not take too long a time to finish.
* Since State of Health is a very critical part of BMS so it should be predicted with as much accuracy as possible.
* Noise and Outliers are part of the dataset which needs to be taken extreme care of in order to improve accuracy of the model.
  1. **Performance metric:** It’s a standard regression problem hence for modelling the loss function is mean squared error however, as a performance metric for the various models I have taken Mean Absolute Error as the metric to compare various models. Since, the target variable is small with values ranging from 1 to 2 so errors are in general very small. So mean squared error value would be very small which won’t give us a good perspective about a model’s performance however, mean absolute error would be able to give a better idea about a model’s performance. Also, since we are considering the mean of the absolute error so it takes into account any anomaly or outliers in the dataset.

1. **About the Dataset:**

The dataset that is taken to model our task of prediction of Capacity is supplied by Prognostics Centre of Excellence under National Aeronautics and Space Administration. Following is the link to the Dataset:

**[https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data](https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-datarepository/publications/#battery)**

**[repository/publications/#battery](https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-datarepository/publications/#battery)**Here a type of Lithium-ion battery called 18650 series is run through many charge/discharge cycles and sampled values of voltage, current and temperature is supplied in the dataset.For our analysis, we have selected four batteries labelled #5, 6, #7 and #18. The batteries were charged at a constant current of 1.5 A until the charging voltage reached 4.2 V and, then, continued to charge at a constant voltage of 4.2 V until the charging current dropped below 20 mA. The batteries were discharged at a constant current of 2 A until the voltage of the battery dropped to 2.7 V, 2.5 V, 2.2 V, and 2.5 V

* 1. **Structure of the Dataset:**

**cycle:** top level structure array containing the charge, discharge and impedance operations

**type:** operation type, can be charge, discharge or impedance

**ambient\_temperature:** ambient temperature (degree C)

**time:** the date and time of the start of the cycle, in MATLAB date vector format

**data:** data structure containing the measurements

**for charge the fields are:**

**Voltage\_measured:** Battery terminal voltage (Volts)

**Current\_measured:** Battery output current (Amps)

**Temperature\_measured:** Battery temperature (degree C)

**Current\_charge:** Current measured at charger (Amps)

**Voltage\_charge:** Voltage measured at charger (Volts)

**Time:** Time vector for the cycle (secs)

**for discharge the fields are:**

**Voltage\_measured:** Battery terminal voltage (Volts)

**Current\_measured:** Battery output current (Amps)

**Temperature\_measured:** Battery temperature (degree C)

**Current\_charge:** Current measured at load (Amps)

**Voltage\_charge:** Voltage measured at load (Volts)

**Time:** Time vector for the cycle (secs)

**Capacity:** Battery capacity (Ahr) for discharge till 2.7V

**for impedance the fields are:**

**Sense\_current:** Current in sense branch (Amps)

**Battery\_current:** Current in battery branch (Amps)

**Current\_ratio:** Ratio of the above currents

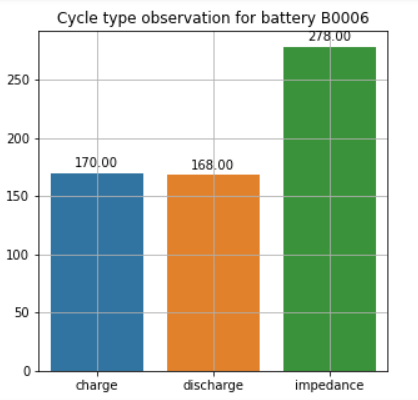
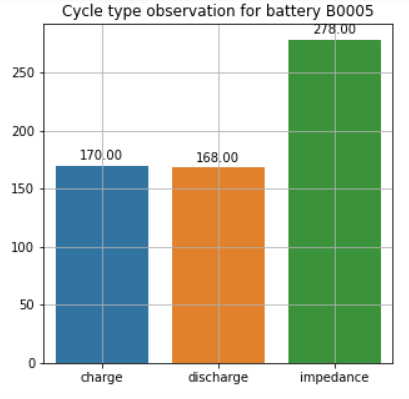
**Battery\_impedance:**Battery impedance computed from raw data

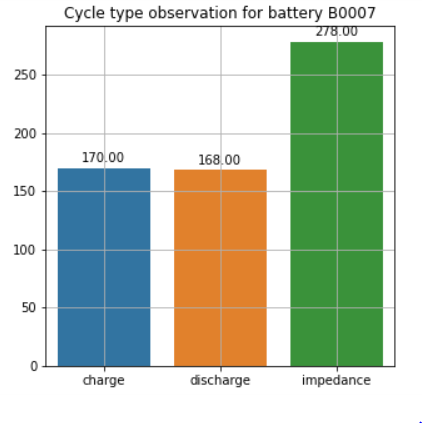
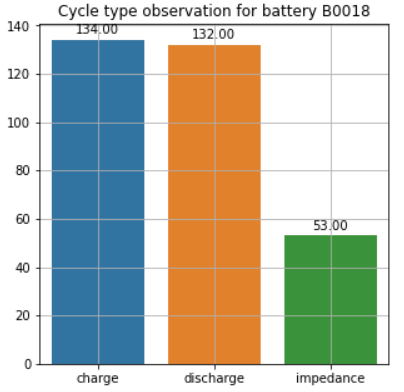
**Rectified\_impedance:** Calibrated and smoothed battery impedance

**Re:** Estimated electrolyte resistance (Ohms)

**Rct:** Estimated charge transfer resistance (Ohms)

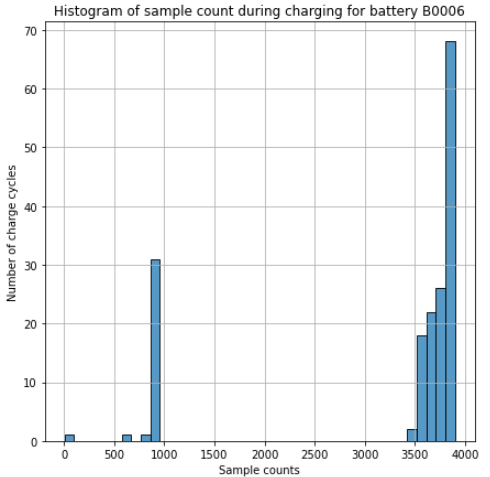
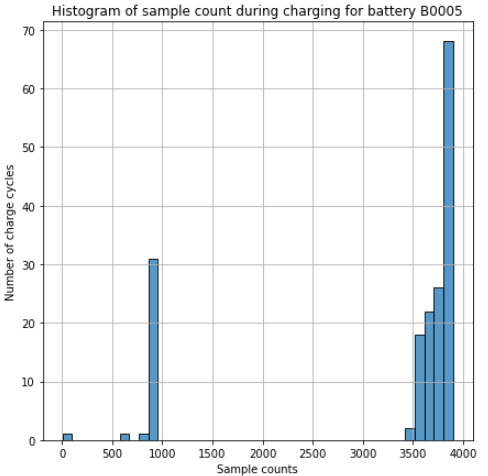
1. **Existing approaches to the problem:** The approaches that are taken so far are mostly Deep learning based where the query point is an array of Voltage or Current data from the charge or/and discharge cycles. The article mentioned in reference section 1 and 2 uses Deep Learning based models such as FNN, CNN and LSTM. Though the DL models have achieved significantly great results of Mean Absolute Error of around 0.25% but they are usually complex in design, requires more compute power and does not give feature importance for analysis and model’s interpretability. The article mentioned in reference section 3 uses some of the classical machine learning based models but the Mean Absolute Error is around 3-4%.
2. **Improvements to the ML approaches:** The approach that I have taken to solving the problem is with traditional ML model looking at the simplicity, quickness in response and model’s interpretability, however with the required engineered features I could achieve Mean Absolute Error of about 0.3% which is almost at par with the DL models.
3. **Exploratory Data Analysis:** 
   1. **Counting the number of charge/discharge cycles:** Every battery in the dataset is made to undergo several charge-discharge-impedance cycles. We want to take a look at the count of number of charge-discharge cycles of the batteries. Following is the count of the number of charge-discharge-impedance cycles represented in bar graph for the various batteries.

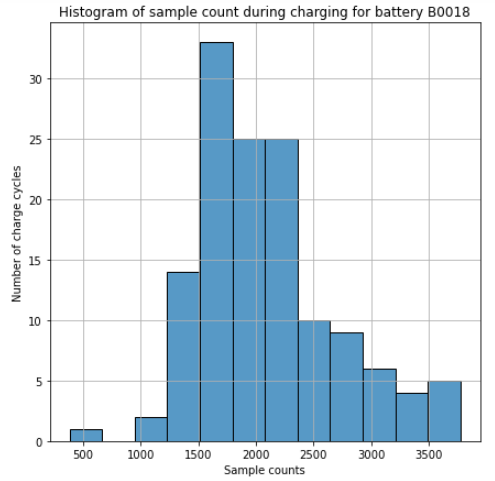
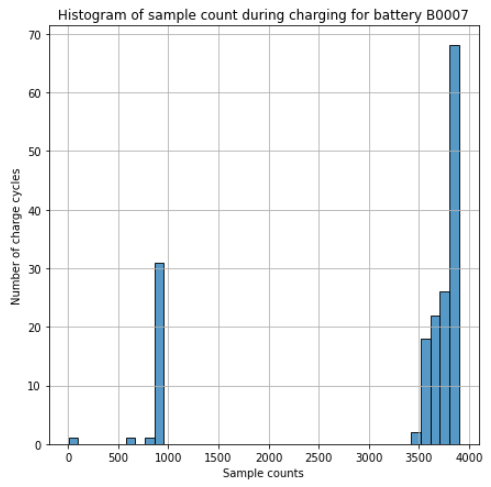




We can see that the charge, discharge and impedance cycles for the batteries are not the same. However, the number of such cycles is the same for battery B0005, B0006 and B0007 but for B0018 the values are different. But in any case, the number of charge and discharge cycles should be the same. There cannot be any charge-charge or discharge-discharge cycles placed consecutively. Hence, there are possibilities of outlier cycles in the dataset which we need to identify and remove from our dataset.

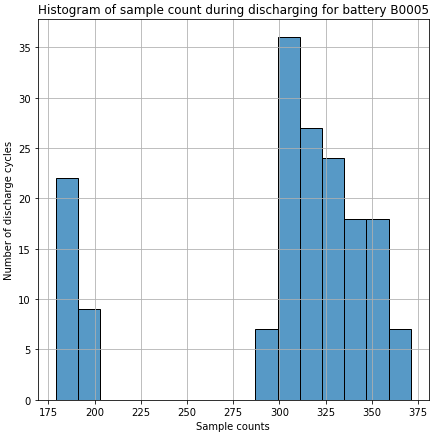
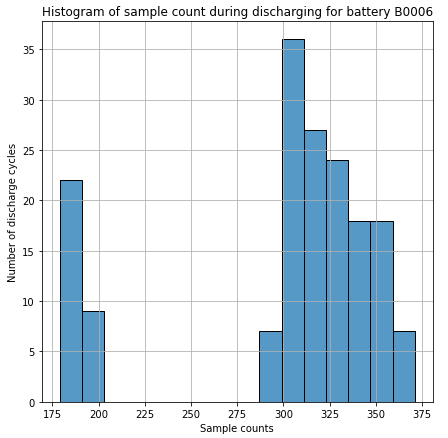
* 1. **Analysis of sample count during charge cycles for the various batteries:** The charge cycles consist of voltage, current and temperature sampled after definite time intervals. I want to analyse the number of sample counts for the various charge cycles and plot the Histogram plot.

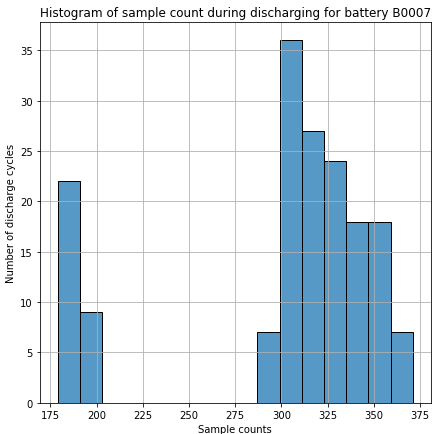
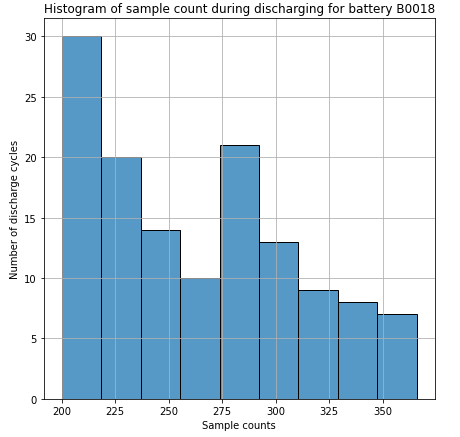




From the plots it is seen that not all charge cycles have the same number of samples. Also, we can see that the distribution of the sample count vs number of charge cycles is identical for battery B0005, B0006 and B0007 but different for battery B0018. For the first three batteries, minimum number of sample count is 5 which is a bit absurd and those cycles are reckoned as an outlier. So, a threshold of 600 is taken, below which all the charge cycles having sample count lesser is taken as an outlier for all the batteries and are removed. It is seen that for batteries 5,6 and 7 there are 2 samples with sample count less that 600 and 1 sample for battery number 18. The cycle numbers with sample size less than 600 for batteries 5,6 and 7 are 84 and 615 however for battery 18 the cycle number 139 has sample size less than 600. And these are the cycles which would be removed for further analysis as they are outlier cycles.

* 1. **Analysis of sample count during discharge cycles for the various batteries:** The similar analysis for the sample count is done for the discharge cycles as well and following are the Histogram plots for the various batteries:

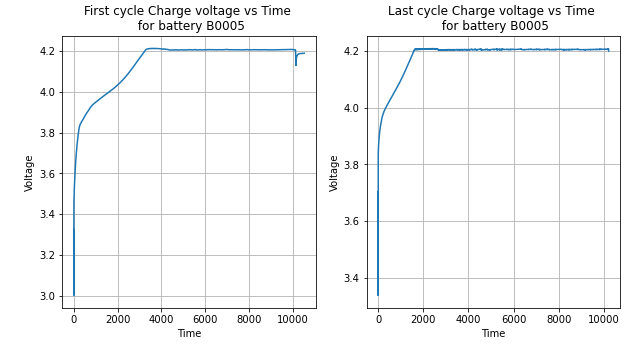
 

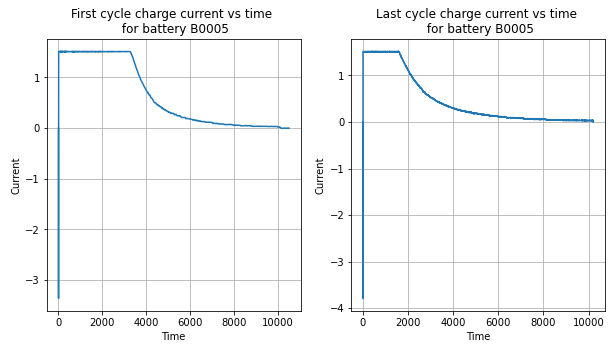
Here we can see that the max value of sample count is around 400 and min value is around 200 for all the batteries. Also, the distribution is same for the first three batteries and different for B0018. As there is no significantly high or low value count compared to the average value, so we can say that there is no outlier datapoints in the discharge cycles with respect to the sample size.

* 1. **The voltage, current and temperature characteristic curves during charge cycle:**

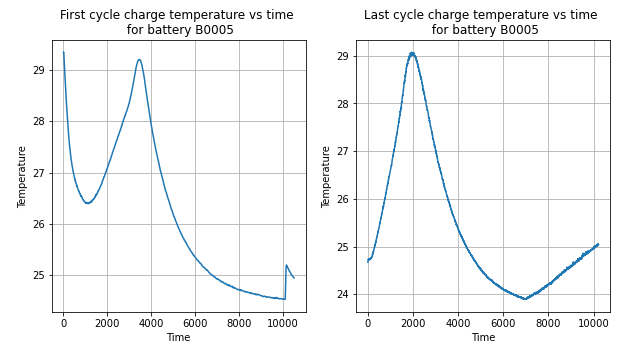
Following are the voltage, current and temperature curves for the battery #5 for the first and the last charge cycle, however similar trend was seen for all the other batteries too:



From the above curve we can observe that during charging the terminal voltage of the battery keeps rising upto 4.2V and after that the battery charges at constant voltage. Now, looking at the voltage curve for the first and the last cycle, we can observe that the time required to reach a voltage of 4.2V decreases as the charging cycle number increases.



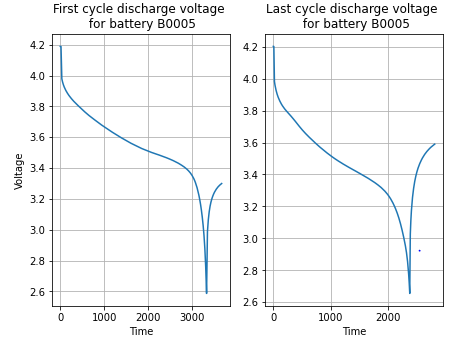
From the above current curves, we can observe that during charging the terminal current of the battery remains constant at 1.5A and after that as the charging mode is switched to constant voltage, the current steadily decreases towards 0A. Now, looking at the current curve for the first and the last cycle, we can observe that the time for which current remains constant decreases as the charging cycle number increases.



From the above temperature curve, we can observe that during charging the temperature of the battery initially decreases and after reaching a local minima, it starts rising towards a local maxima after that the temperature start decreasing. Now, looking at the temperature curve for the first and the last cycle, we can observe that the time at which the temperature reaches the local maxima decreases as the charging cycle number increases.

* 1. **The voltage, current and temperature characteristic curves during discharge cycle:**

Following are the voltage, current and temperature curves for the battery #5 for the first and the last discharge cycle, however similar trend was seen for all the other batteries too:

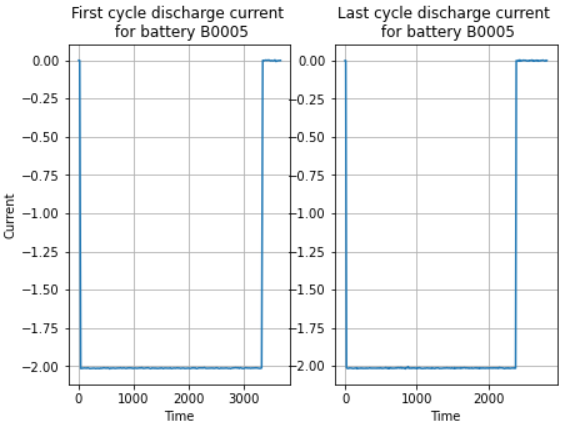


Analysing the Voltage curve during discharge cycles we have:

1. The voltage curve monotonically decreases from a maximum value to the minimum value.

2. The rate of decrease changes with each cycle which is evident from the first and last discharge cycle curves.

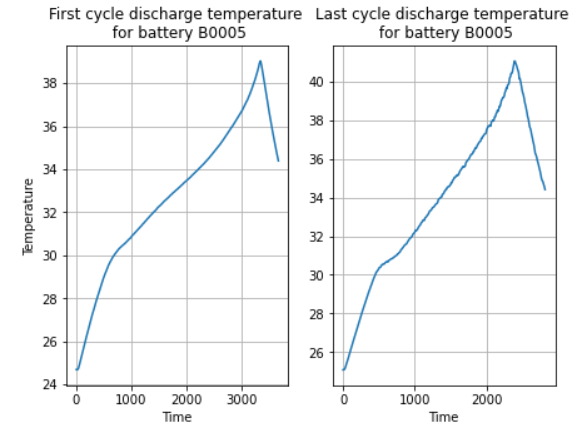
3. The time required to reach the voltage of 2.7V decreases as the number of discharge cycles increases.



Analysing the discharging Current curve:

1. The current remains constant at 2A during discharge of the battery

2. The duration for which the current remains constant at 2A decreases with each cycle which can be seen from the first and last discharge cycle curves.



Analysing the Temperature curve during the discharge cycles:

1. The time at which the temperature reaches the peak also decreases with each cycle as well.

* 1. **Checking for any charge-charge or discharge-discharge cycles:**

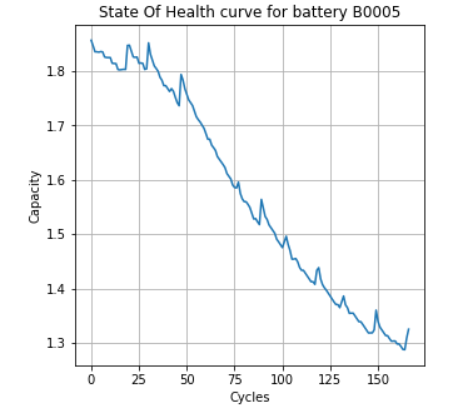
During the various charge and discharge cycles of a battery there cannot be two consecutive charge or discharge cycle because after a full charge, a battery cannot undergo another charge cycle without an intermediate discharge cycle and similar things apply for two consecutive discharge cycles. This type of charge or discharge sequences are considered outlier cycles and need to be filtered from our analysis. Going through the complete dataset for all the four batteries it is seen that for batteries #5,#6 and #7 there is charge-charge sequence at 22-23 and discharge-discharge sequence at 178-179 however for battery #18 there is only charge-charge sequence anomaly at index 90-91.

For removing the outlier cycles, I have removed the first charge cycle as a battery cannot charge twice before discharge and removed the second discharge cycle as a battery cannot discharge twice before charging.

As part of further cleaning of anomalous cycles, I checked if for any battery the first cycle is a discharge cycle and last one is a charge, if yes then I would remove those two cycles but this type of anomaly was not found.

* 1. **Plotting the target variable “Capacity” for various cycles:**

I am here plotting the Capacity vs Discharge Cycle curve for the first battery #5 however the same trend is seen for the other batteries too.



From the above curve it is seen that the capacity of a battery usually decreases as the number of discharge cycles increases with some noise nevertheless. The variance of the capacity is from 1.8 to 1.26 as per the diagram. In general, when the capacity of a battery reaches 70% of the rated value then it is considered to have reached it’s end of life. So, for all the battery when the instantaneous capacity reached at 70% of the rated capacity value i.e. 1.3 approx. then further charging was stopped.

* 1. **Final count of the number of charge-discharge cycles:**

Finally, after removing all the outlier cycles, we are left with only the charge-discharge cycles for the various batteries. For battery numbered #5, #6 and #7 the number is 167 and for the fourth battery #18 the number is 132. Hence, the total number of such cycle pairs for all the batteries is 633. For modelling the problem, we would engineer some features and each of the feature set would come from one charge-discharge cycle. Hence, as there are 633 cycle pairs so there will be 633 feature set or data point for modelling.

1. **First cut approach to the problem:**

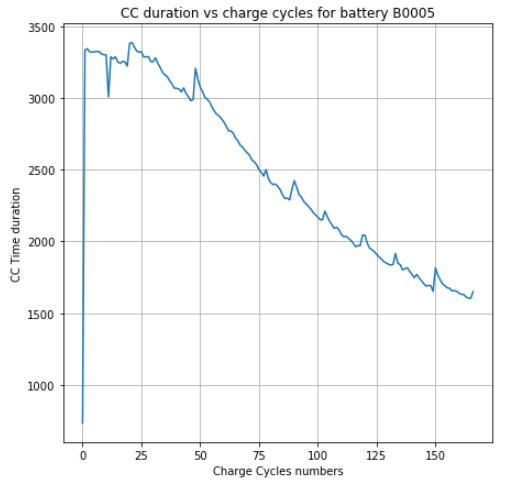
* Firstly, there is no latency constraint however accuracy constraint is higher. So, my focus would be to model with accuracy rather than faster response to query points. I will try to give out the most important features in order to make the model more interpretable
* As read in every research paper the Dataset that is given to us is noisy and not everything is needed for featurization. My first job would be to get rid of the noise. The approach that I will be using is to check if there is proper Charge-Discharge cycle in order and not any Charge-Charge or Discharge-Discharge cycle during cleaning the dataset. Also, I would check if there are any cycles where the number of samples are extremely low and would remove those cycles from our dataset for analysis as was discussed in the EDA section.
* The point in time features would be engineered from the various voltage, current and temperature curves of the battery from both the charge-discharge cycles.
* Not all the features may be useful and hence I would perform correlation analysis to see which features are highly correlated with the target variable i.e. capacity however inter-feature correlation is less.
* If the features become highly correlated then I would perform perturbation test to check for multi-collinearity.
* After the features are ready, I would go with min-max normalization, as it has given better results in most of the research papers that I have gone through.
* Before training with any model, first we I would prepare a baseline model which would be predicting the mean value and get the MSE. This would be used as a benchmark to test the performance of other machine learning models.
* I will try to model with both classical ML and DL models and compare and contrast the results.
* In all of these models, I would try proper hyper parameter tuning with K-folds CV to tweak the model performance.
* The target variable is to estimate the Capacity or SOH which is a continuous variable. Hence the problem is a regression problem as said earlier.
* The performance metric for comparing various models that I would use is mean absolute error.
* At last I would prepare a table to compare the various

1. **Feature Engineering:**

Several aspects of the charge-discharge curve were studied to engineer various features.

* 1. **Duration for which current was constant during charging cycles:**

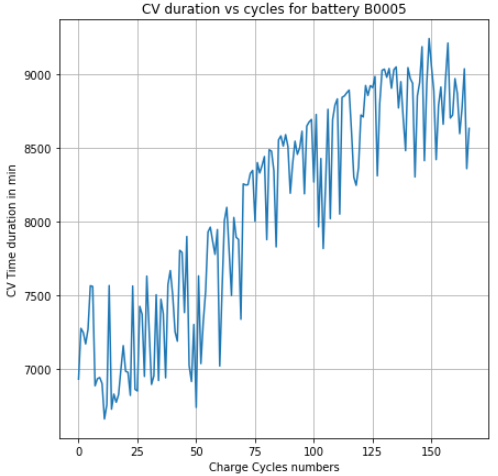
Following is the curve between Constant Current Duration vs Charge Cycle for battery #5 and similar pattern is observed for the other batteries.



From the above curve it can be seen that the constant current duration for a battery during charge cycles decreases almost monotonically and it has great correlation with the target variable i.e. Capacity.

* 1. **Duration for which voltage was constant during charging cycles:**

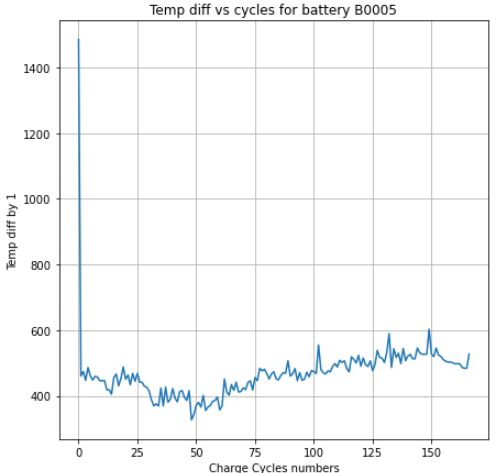
Following is the curve between Constant Voltage Duration vs Charge Cycles for battery #5 and similar observations are seen for the other batteries as well.



From the above curve it seems as if the Constant Voltage duration rises as the charging cycle number increases but it has a lot of noise and hence this is not considered as a feature for modelling.

* 1. **Time required for the temperature to reduce by 1 deg from the local maxima during charge cycle:**

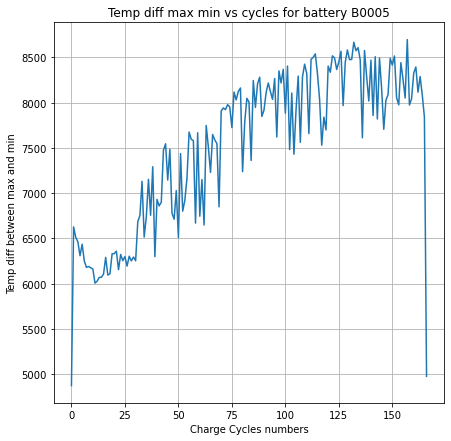
Following is the curve between the time required for the temperature of battery #5 to decrease by 1 degree from the local maxima. Similar characteristics are seen for the other batteries as well.



There is no definite monotonic behaviour seen in the duration required to reduce the temperature of a battery by 1 degree during charging cycle from the local maxima point. Hence, this is not considered as a feature for modelling.

* 1. **Duration in time between the local maxima and the minima value of temperature during charging cycles:**

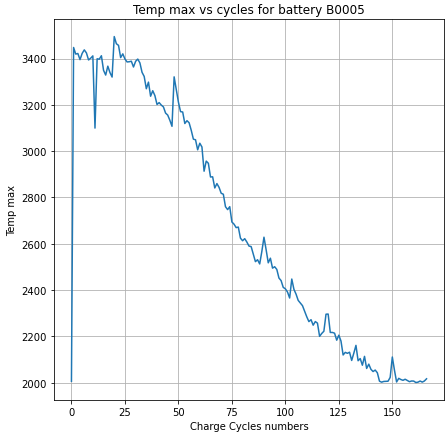
The curve between the time duration between the local maxima and global minima value of temperature during charging cycles for battery #5 vs the charge cycles is as shown in the figure below. Similar characteristics are also seen for the other batteries:



The above is the curve for the time duration between the local max and global min of the temperature curve. This curve seems to be very noisy and hence it is not considered as a feature for modelling.

* 1. **Duration after which temperature has reached the local maxima during the charging cycles:**

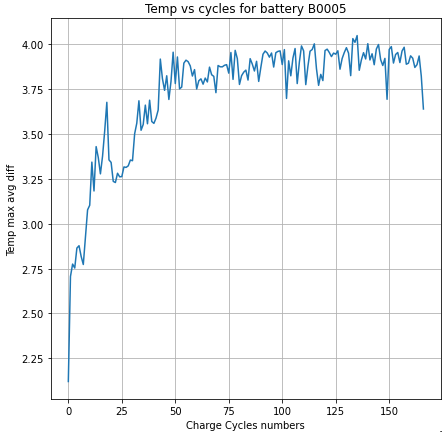
Following is the curve between the time required to reach the local maxima of the temperature curve during charging for battery #5 vs the various charge cycles. Similar characteristics are seen for the other batteries as well.



The feature seems to be monotonically decreasing as a function of the number of charge cycles and this characteristic is in agreement with that of the target variable and hence we can consider this to be a feature for modelling.

* 1. **Difference between the maximum and average value of temperature during charging cycles:**

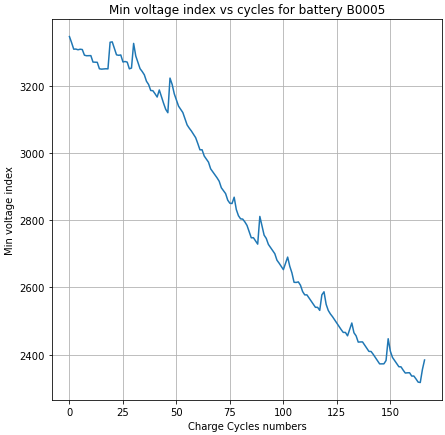
Following is the curve between the difference between the global maximum and average value of temperature for the various charge cycles for battery #5 vs the various charge cycles. Similar sort of trend is seen for the other batteries as well.

****

From the above curve we can conclude that there is no correlation between this feature and the target variable and hence we are not considering this as a feature for modelling.

* 1. **Time required to reach a voltage of 2.7 Volts for the various discharge cycles:**

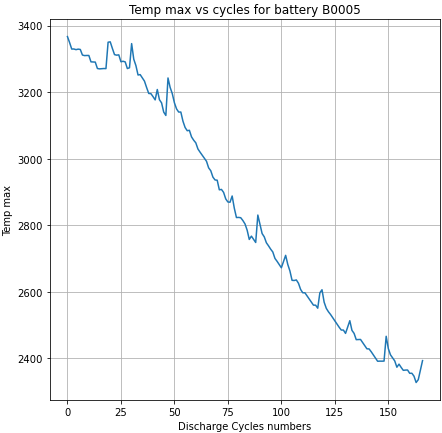
The following is a characteristic of the curve between the time required to reach a voltage of 2.7 volts for battery #5 during discharge vs the various discharge cycles:



From the above characteristics curve there seems to be a great resemblance with the target variable and hence it is considered as a feature for modelling.

* 1. **Duration after which temperature has reached the maximum value during the discharge cycles**

Following is the curve between time after which temperature of the battery #5 has reached maximum value during discharge vs the charge cycles. Similar observations are seen for the other batteries as well:



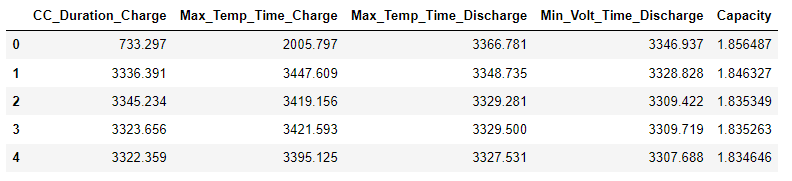
As can be seen the variation of this feature resembles with the target variable and hence this can be taken as a feature for analysis and modelling.

1. **Feature correlation analysis:**

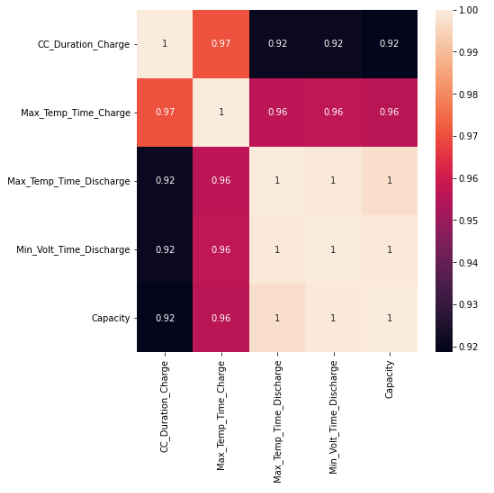
The following are the four point-in-time features which have been selected for modelling:

1. **CC\_Duration\_Charge:** This represents the duration of constant current during charge cycles
2. **Max\_Temp\_Time\_Charge:** This is the duration after which the local maxima was reached during charge cycles.
3. **Max\_Temp\_Time\_Discharge**: This is the duration after which the maximum voltage was reached during discharge cycles.
4. **Min\_Volt\_Time\_Discharge:** This is the duration after which voltage reached 2.7 volts during discharge cycles.

**Following is the dataset after loading it into a Pandas DataFrame:**



**Following is the correlation matrix represented as a heat map:**



We can very clearly see from the correlation heatmap that there is very high correlation between the features and the target variable i.e. Capacity of the batteries.

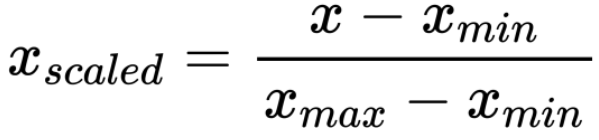
1. **Splitting and scaling:**
   1. **Train-Test split:** The dataset that we have engineered is not a time series data i.e the data points don’t have any temporal nature hence we have done random sampling to select the train and test set in 3:1 ratio. Also I have done the feature engineering before splitting the dataset as the train and test datapoints would essestially undergo the same process without any exception.

**The code for the same is:**

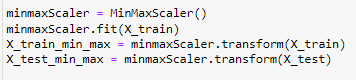


So, for doing the train and test split, Sklearn’s **train\_test\_split** module was used with a test set size of 25% of the original dataset as indicated in the code by test\_size = 0.25. We are here seeding the split with random\_state = 42 so that the same datapoints are obtained in train and test sets across various training cycles.

* 1. **Doing the Min-Max scaling:** For scaling the data point, the min-max type scaling was done. For mathematical formula for doing the min-max scaling is as follows:



**Code for the min-max scaling is as follows:**

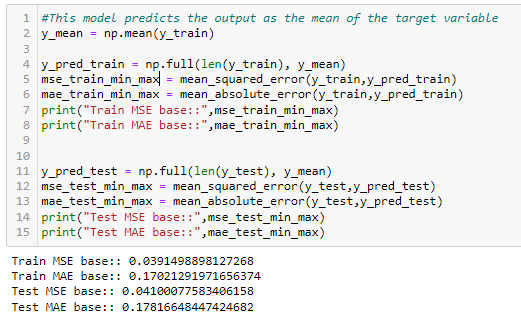


For doing the min-max scaling, the **MinMaxScaler** module of Sklearn was imported. For doing the scaling, only the train data was given to the fit function in order to calculate the minimum and maximum values for each feature and after that scaling of the data points were separately done for the train and test dataset.

1. **Modelling:**
   1. **Base Model:**

Before doing any modelling with the existing ML or DL models, a Baseline Model is selected so as to compare the performance of other models. The simple baseline model is a mean model which predicts all the outputs to the mean value of the target variable and as such I expect any sensible model to have a better performance than the baseline model otherwise the model would be reckoned as a dumb model and rejected.

**The code snippet for the baseline model is:**

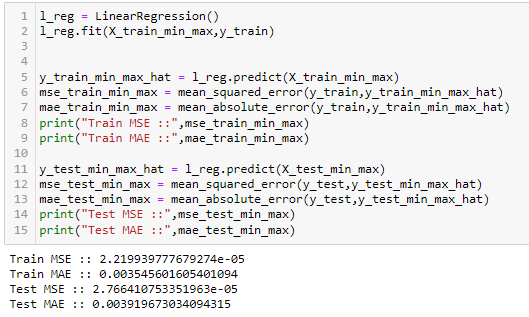


As can be seen from the results, this baseline model has a Mean Absolute Error of around 0.17 and any sensible model should have a value of MAE lesser than this.

* 1. **Linear Regression:**

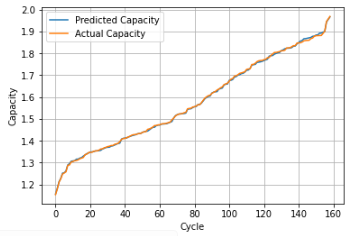
In this technique I am predicting the target variable with simple linear regression without any regularization.

**The code snippet for the same is give below:**



For the linear regression I am using the **LinearRegression** module from Sklearn. In the fit function, the model is trained with the training data and after doing the training, all the weights of the hyperplane which passes through the dataset is calculated. From the model output it is seen that the train and test MAE are 0.0035 and 0.0039 which are very close to each other hence the model is not overfitting and I didn’t choose to go with regularization.

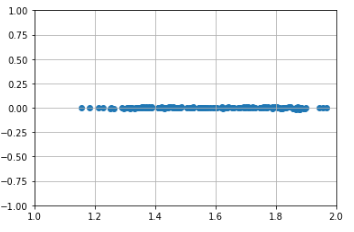
**The plot between the actual test values and the predicted values of the target variable is as shown below:**



From the above curve we can see that the predicted values almost follows the actual target variable.

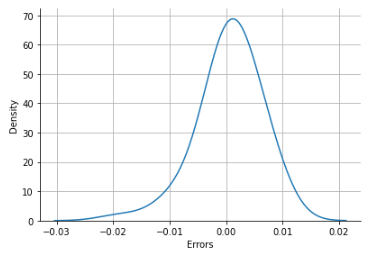
**Checking for Heteroskedasticity:**

The test for heteroskedasticity is mainly to check if there are any outliers in the dataset which results in uneven distribution of the datapoints. Following is the plot for heteroskedasticity which is a plot between the predicted values along the x axis and the difference between the actual and predicted values along the y-axis.



The plot shows that there is no heteroskedasticity in the dataset.

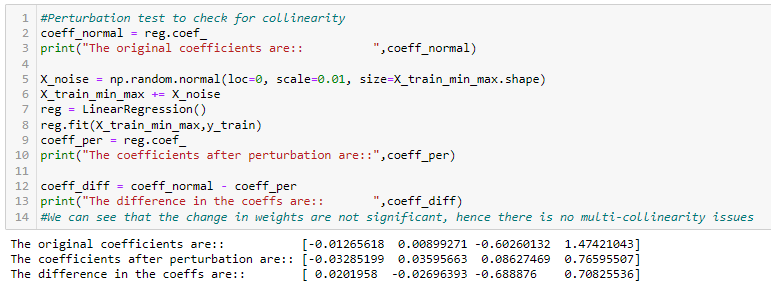
**Following is the error distribution for the predicted values:**



From the predicted value it is seen that the mean of the distribution is at 0.00061 with a standard deviation of 0.0052.

**Perturbation test to check for multicollinearity:**

The code snippet for the perturbation test and the results are as shown in the figure below:

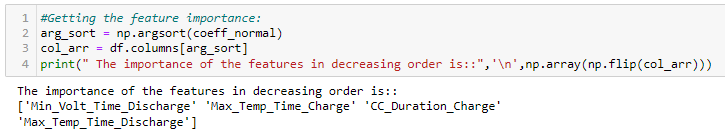


From the above results, it is evident that even when we add a bit of noise to the dataset, the weights of the hyperplane don’t change significantly and hence the feature are not multi-collinear.

**Getting the feature importance:**

Looking into the feature importance which is obtained by taking the weight vector into account we see that the most important feature is the time required to reach a voltage of 2.7V during a discharge cycle.

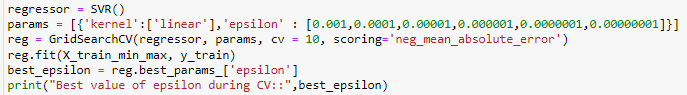
**The code snippet for the same is as follows:**



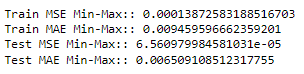
* 1. **Support Vector Regression:**

Since the results with Linear Regression was significantly better that the baseline model hence it’s clear that the linear models may perform better than the non-linear models. As such I am using SVR with linear kernel. For modelling I am using the **SVR** module from Sklearn. Also, I am using simple grid search with 10 folds cross-validation with epsilon as the hyper-parameter for tuning.

**The code snippet for the same is as shown below:**

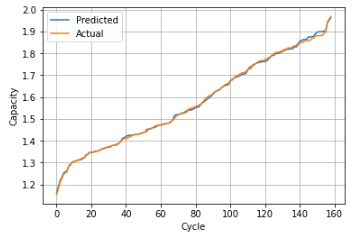


The best value of epsilon during CV is 0.001. Now, after modelling with the best epsilon following are the values of the train and test MSE and MAE.



Looking at the MAE values for both train and test, we can see that there is not a big difference in the values and hence I can conclude that the model is not overfitting.

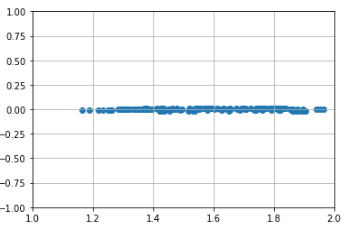
**The plot between the actual test values and the predicted values of the target variable is as shown below:**



From the above curve we can see that the predicted values almost follow the actual target variable with small difference when the capacity is more than 1.8.

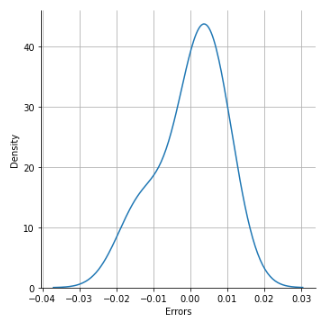
**Checking for Heteroskedasticity:**

Following is the plot for heteroskedasticity which is a plot between the predicted values along the x axis and the difference between the actual and predicted values along the y-axis.



The plot shows that there is no heteroskedasticity in the dataset.

**Following is the error distribution for the predicted values:**

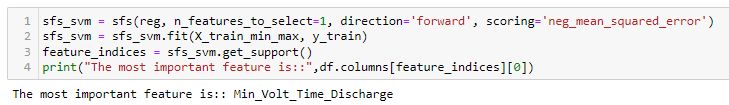


From the predicted value it is seen that the mean of the distribution is at -0.00001 with a standard deviation of 0.008.

**Getting the feature importance:**

Looking into the feature importance which is obtained by doing the forward feature selection, we see that the most important feature is the time required to reach a voltage of 2.7V during a discharge cycle.

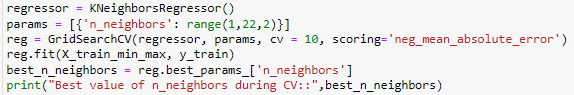
**The code snippet for the same is as given below:**



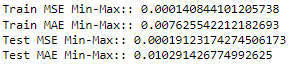
* 1. **K-Nearest Neighbour Regression:**

For modelling with KNN Regressor, I am using the **KNeighborsRegressor** module of Sklearn. Also, I am using simple grid search with 10 folds cross-validation with n\_neighbors as the hyper-parameter for tuning.

**Following is the code snippet for KNN Regressor:**

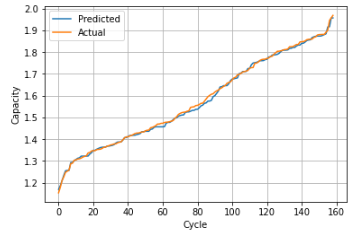


The best value of n\_neighbors during CV is 3. Now after modelling the KNN with n\_neighbors as 3, we get the following results for train and test MSE and MAE:



Now, looking at the values of the MAE for both train and test dataset, we see that there is a huge difference (around 10x times) between this model and the other linear models like Linear Regression and SVR. Hence, linear models are performing better than KNN.

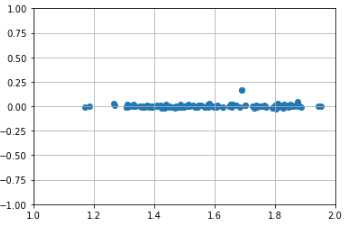
**The plot between the actual test values and the predicted values of the target variable is as shown below:**



From the above curve we can see that the predicted values almost follow the actual target variable with differences when the capacity is more than around 1.5.

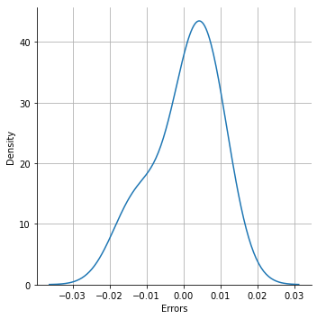
**Checking for Heteroskedasticity:**

Following is the plot for heteroskedasticity which is a plot between the predicted values along the x axis and the difference between the actual and predicted values along the y-axis.



The plot shows that there is no heteroskedasticity in the dataset.

**Following is the error distribution for the predicted values:**



From the predicted value it is seen that the mean of the distribution is at 0.0005 with a standard deviation of 0.0081.

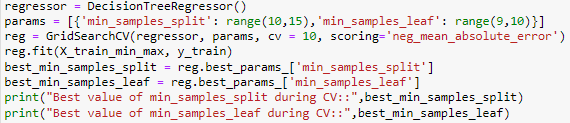
**Getting the feature importance:**

Looking into the feature importance which is obtained by doing the forward feature selection, we see that the most important feature is the time required to reach a voltage of 2.7V during a discharge cycle.

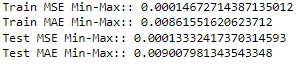
* 1. **Decision Tree Regression:**

For modelling with Decision Tree regression, I am using the **DecisionTreeRegressor** module from Sklearn. I am using simple grid search with 10 folds cross-validation with min\_samples\_split and min\_samples\_leaf as the hyper-parameters for tuning.

**The snippet for the same is as given below:**

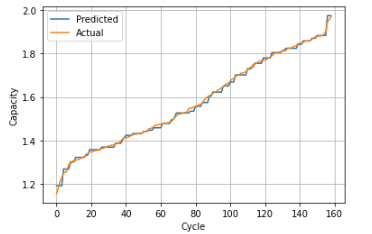


The best value of min\_samples\_split and min\_samples\_leaf is 14 and 9 respectively. Now, modelling with these hyperparameter we get the following values of MSE and MAE for the train and test dataset.



Looking at the MAE for the train and test set, there is a large gap between this model with the linear models.

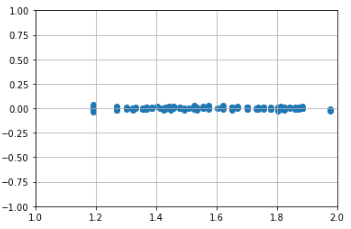
**The plot between the actual test values and the predicted values of the target variable is as shown below:**



From the above curve we can see that the predicted values has small differences with the actual values throughout the range of target variable.

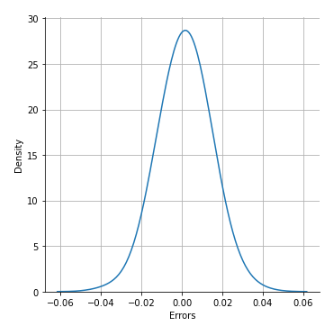
**Checking for Heteroskedasticity:**

Following is the plot for heteroskedasticity which is a plot between the predicted values along the x axis and the difference between the actual and predicted values along the y-axis.



The plot shows that there is no heteroskedasticity in the dataset.

**Following is the error distribution for the predicted values:**



From the predicted value it is seen that the mean of the distribution is at 0.0013 with a standard deviation of 0.0114.

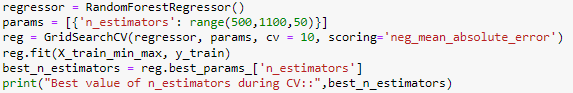
**Getting the feature importance:**

Looking into the feature importance which is obtained by doing the forward feature selection, we see that the most important feature is the time duration for which the current was constant during charging cycles.

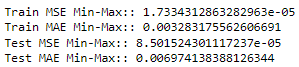
* 1. **Random Forest Regression:**

For modelling with Random Forest regression, I am using the RandomForestRegressor module from Sklearn. I am using simple grid search with 10 folds cross-validation with n\_estimator as the hyper-parameter for tuning.

**The code for the same is as given below:**

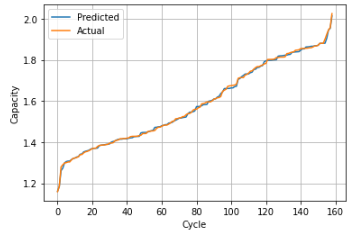


The best value of n\_estimator is 950 and after modelling with this value get the following value of MSE and MAE for the train and test dataset:



Looking at the MAE value for both the train and test dataset, there is a significant improvement from the Decision Tree model after applying Random Forest.

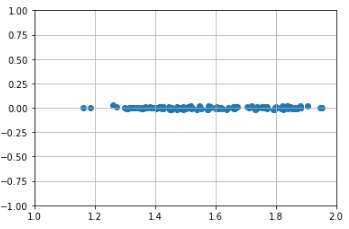
**The plot between the actual test values and the predicted values of the target variable is as shown below:**



From the above plot we we can see that the predicted values has very little differences with the actual values throughout the range of target variable.

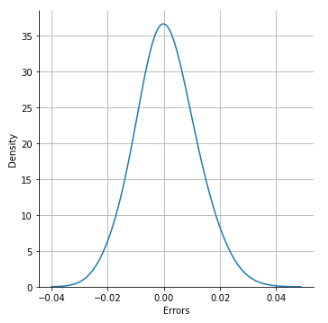
**Checking for Heteroskedasticity:**

Following is the plot for heteroskedasticity which is a plot between the predicted values along the x axis and the difference between the actual and predicted values along the y-axis.



The plot shows that there is no heteroskedasticity in the dataset.

**Following is the error distribution for the predicted values:**



From the predicted value it is seen that the mean of the distribution is at 0.00074 with a standard deviation of 0.090.

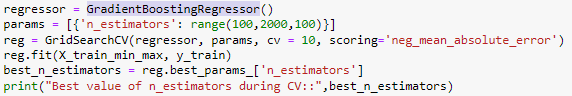
**Getting the feature importance:**

Looking into the feature importance which is obtained by doing the forward feature selection, we see that the most important feature is the time duration for which the current was constant during charging cycles.

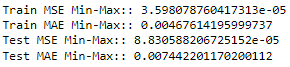
* 1. **Gradient Boosted Regressor:**

For modelling with Gradient Boosted Decision Tree regression, I am using the GradientBoostingRegressor module from Sklearn. I am using simple grid search with 10 folds cross-validation with n\_estimator as the hyper-parameter for tuning.

**The code for the same is as given below:**

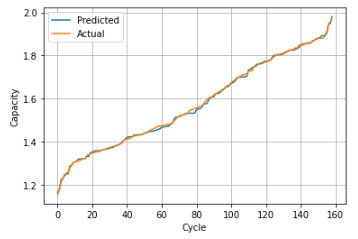


The best value of n\_estimator is 100 and after modelling with this value get the following value of MSE and MAE for the train and test dataset:



Looking at the MAE value for both the train and test dataset, there is a significant improvement from the Decision Tree model as well after Gradient Boosted Regressor model.

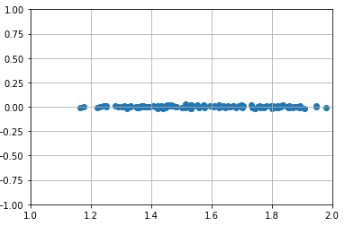
**The plot between the actual test values and the predicted values of the target variable is as shown below:**



From the above plot we can see that the predicted values almost follow the target variable except for the value near 1.5.

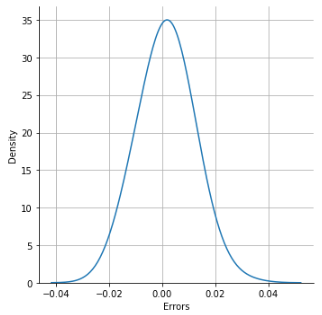
**Checking for Heteroskedasticity:**

Following is the plot for heteroskedasticity which is a plot between the predicted values along the x axis and the difference between the actual and predicted values along the y-axis.



The plot shows that there is no heteroskedasticity in the dataset.

**Following is the error distribution for the predicted values:**



From the predicted value it is seen that the mean of the distribution is at 0.00152 with a standard deviation of 0.0092.

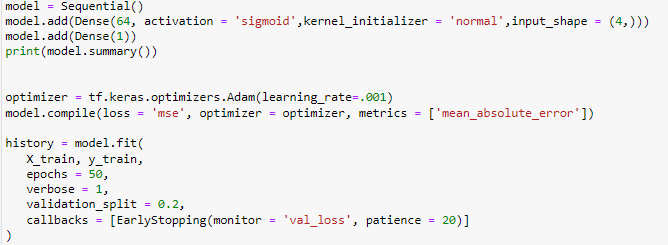
**Getting the feature importance:**

Looking into the feature importance which is obtained by doing the forward feature selection, we see that the most important feature is the time duration for which the current was constant during charging cycles.

* 1. **Multi-Layered Perceptron:**

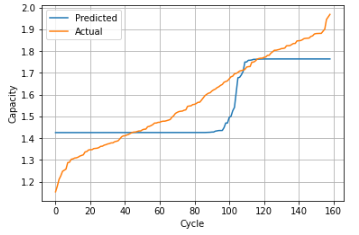
For modelling with Multi-Layered Perceptron for regression, I am using the **Sequential** API from Keras library. I have incorporated only a single hidden layer with 64 sigmoid activation units. The weights are initialised with random normal distribution. The output layer is a single linear activation unit which predicts the target variable which is a continuous random variable. The loss function used is Mean Squared Unit with optimizer as Adam. The metric used for assessing the model’s performance is Mean Absolute Error.

**Following is the code snippet for the same:**



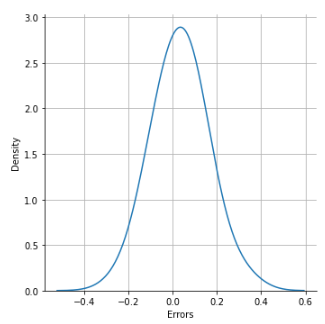
For the above MLP model we could get test MAE of 0.09 which is just better than the baseline mean model.

**The plot between the actual test values and the predicted values of the target variable is as shown below:**



From the above plot we can see that the predicted values don’t have any resemblance with the actual values. Hence, the model is performing very bad compared to the linear models

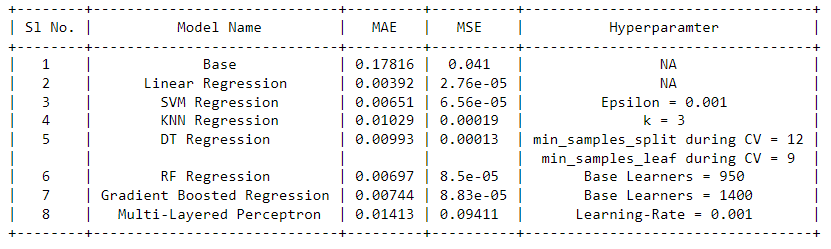
**Following is the error distribution for the predicted values:**



From the predicted value it is seen that the mean of the distribution is at 0.03 with a standard deviation of 0.11.

1. **Comparison and conclusion of the various models in tabular form**

Following is the tabular representation of the performance of the various models:



From the above table, it can be very clearly concluded that the simple Linear Regression has outperformed all the other linear and non-linear models and hence I choose this model for deployment.

1. **Future works:**

The features that I have chosen from the various voltage, current and temperature profiles are the simple point in time values from the curves. In this experiment I have taken the four point-in-time features

1. Duration of the constant current during charging
2. Time at which temperature of the battery reached the local maxima during charging
3. Time at which temperature of the battery reached maximum during discharge
4. Time at which the battery terminal voltage reached the 2.7V mark.

So, anyone with these four parameters available can manually input the values to get the present Capacity of the battery. However, there are still many features that can be tried out so that the model’s performance improves even further.

Some of the features that can be tried are:

1. Area under the voltage curve during discharge
2. Area under the temperature curve during discharge
3. The slope of the voltage curve between the two knee points

Further, since the number of features is small hence the Deep Learning models did not perform well however, more number of features can be engineered and then tried with Deep Learning models for getting better results.

1. References:
2. [Energies | Free Full-Text | Battery State-of-Health Estimation Using Machine Learning and Preprocessing with Relative State-of-Charge (mdpi.com)](https://www.mdpi.com/1996-1073/14/21/7206)
3. [Machine Learning-Based Lithium-Ion Battery Capacity Estimation Exploiting Multi-Channel Charging Profiles | IEEE Journals & Magazine | IEEE Xplore](https://ieeexplore.ieee.org/document/8731962)
4. [State of health prediction of lithium-ion batteries based on machine learning: Advances and perspectives - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S2589004221012347)
5. Links:
6. The link to ipython notebooks is: [abhijitsen712/Case\_Studies (github.com)](https://github.com/abhijitsen712/Case_Studies)
7. The link where the actual production model is running: [ec2-18-212-16-244.compute-1.amazonaws.com:8080/index](http://ec2-18-212-16-244.compute-1.amazonaws.com:8080/index)