|  |  |
| --- | --- |
| Name Of The Student | Kanak Sahu |
| Internship Project Topic | **“RIO-125: Forecasting System - Project Demand of Products at a Retail Outlet Based on Historical Data.”** |
| Name of the Organization | TCS iON |
| Name of the Industry Mentor | Himalaya Ashish |
| Name of the Institute | Symbiosis University of Applied Sciences |

|  |  |  |
| --- | --- | --- |
| Date | Day # | Hours Spent |
| 20/03/21 | Saturday(Day-17) | 4 |
| Activities done during the day:  **Today I learnt Mean Squared Error and Root Mean Squared Error which is used for accurate prediction.**  **MEAN SQUARED ERROR**  Now consider we are using SSE as our loss function. So if we have a dataset of say 100 points, our SSE is, say, 200. If we increased data points to 500, our SSE would increase as the squared errors will add up for 500 data points now. So let’s say it becomes 800. If we increase the number of data points again, our SSE will further increase. Fair enough? Absolutely not!    The error should decrease as we increase our sample data as the distribution of our data becomes more and more narrower (referring to normal distribution). The more data we have, the less is the error. But in the case of SSE, the complete opposite is happening. Here, finally, comes in our warrior — Mean Squared Error. Its expression is:    We take the average or mean of SSE. So more the data, lesser will be the aggregated error, MSE.    Here as you can see, the error is decreasing as our algorithm is gaining more and more *experience*. The Mean Squared Error is used as a default metric for evaluation of the performance of most regression algorithms be it R, Python or even MATLAB.  **ROOT MEAN SQUARED ERROR**  The only issue with MSE is that the order of loss is more than that of the data. As my data is of order 1 and the loss function, MSE has an order of 2. So we cannot directly correlate data with the error. Hence, we take the root of the MSE — which is the Root Mean Squared Error:    Here, we are not changing the loss function and the solution is still the same. All we have done is reduced the order of the loss function by taking the root. | | |