**WEATHER TIME SERIES FORCASTING**

**REPORT**

In the end, there were total of 14 models which I made to cater to the time-series prediction problem. I started with a simple non-machine-learning model, as a benchmark, it led to a Mean Accuracy Error of 2.62. Following that, I utilized a neural network model in the form of a dense layer for which I obtained the slight scratch off the MAE with the value of 2.63. Nevertheless, this model faced problems because it did not have any theoretical base to masculate time series, thus the temporal characteristic got sacrificed. Later on, when I went for convolutional model, it too didn’t solve my problem. Convolutional models face a stiff challenge in that they apply the same treatment to all data segments by pooling that breaks the chronological order and losing important links in the process. Moving to the relevancy of time value conservation, I applied Recurrent Neural Networks or RNNs, whereas its structure basing on handling of time series data.

RNNs possess a particular power in using information from the past in the present decision-making process thereby capturing complex patterns within sequential data. The internal state of the RNN is equivalent to the memory of past inputs and therefore, the model can deal with sequences of different lengths. On the other hand, although Simple RNN preserves the information from all time steps in theory, it runs into issues in practice. Importantly, it tackles the vanishing gradient issue that the training of deep networks becomes difficult. Besides the graph, Simple RNN is the one having the worst performance among all the models. In order to solve this problem, I employed the LSTM and the GRU RNNs inside the Keras framework.

The Simple Gated Recurrent Unit (GRU) stands out as the most accurate with a MAE of 2.47 trumping all the other models in the consumer goods segment. It is highly effective in identifying trends that evolves and over time and also consumes less resources than the LSTM models. What followed next was my attempt in creating LSTM models which have a proven excellence in working over time-series data. To this end, I have constructed a group of six LSTM models which differ in their architecture depending on the number of units recursively stacked, from 8 to 32. Following this testing, variation 8 with 8 units was demonstrated the most promising and it reached the MAE value of 2.55. I tried different methods including recurrent dropout and bidirectional data to overcome the model overfitting which reduced the mean error value. Bidirectional data representation allows different perspectives to recurrent network. From these enhancements came comparable MAE values among all models, surpassing the common-sense baseline model every time. Similar pattern occurred in the evaluation graph of the mean absolute error as well. Furthermore, I experimented with RNN and 1D convolution hybrid model which turned out to be inadequate with MAE of 3.81. This poor performance could merely indicate that convolution only destroys the order of the data.

The outcomes analysis wited me that RNN and 1D Convultional architectures are most likely to work but GRU and LSTM models perform better compared to that. On the contrary, of all the models mentioned, LSTM is given by experts, however, my attempts show that GRU is better. The correct hyperparameter values related to the number of the units stacked in recurrent layers, the recurrent dropout ratio and 1/0 directional data utilization are all the important steps toward GRU optimization.