TIME SERIES FORECASTING

By- Saksham Jain, Raj Aryan, Karan

INTRODUCTION

In this project, we aim to develop a robust time series forecasting model to predict "Total Cloud Cover [%]" for the next 15,25 and 30th minute using historical weather data. We were given 17 columns of features to work on.

The **Training Set** included data for 11 months from which we kept the last **10%** as our testing set to validate our metrics

We were give a weighted metrics of-
Weighted Accuracy - 0.5x(accuracy for 15 mins) + 0.35x (accuracy for 25 mins) + 0.15x(accuracy for 30 mins)

We have taken the R2 score as our weighted metric in our notebooks.

Our Thought Process -

Firstly we took time to analyse the dataset, we found out the data was unordered and there were many missing rows in between which needed to be handled.

- Our primary objective was to sort the data by date and time, i.e. we wanted 1-Jan 00:00, 1 Jan 00:01, and so on
- Second was to Handle missing values

- Then we wanted to do Feature Engineering to enhance the dataset for our models even more
- Adding appropriate visualisations wherever possible to justify our features and results
- Handle how the data will be loaded in a 3d format ie (Batch size, look back, number of columns)
- Custom Loss function to make our model even more robust
- Then using Grid Search we wanted to know which LSTM configuration will work the best for us
- And at last, we applied various our models to show our results

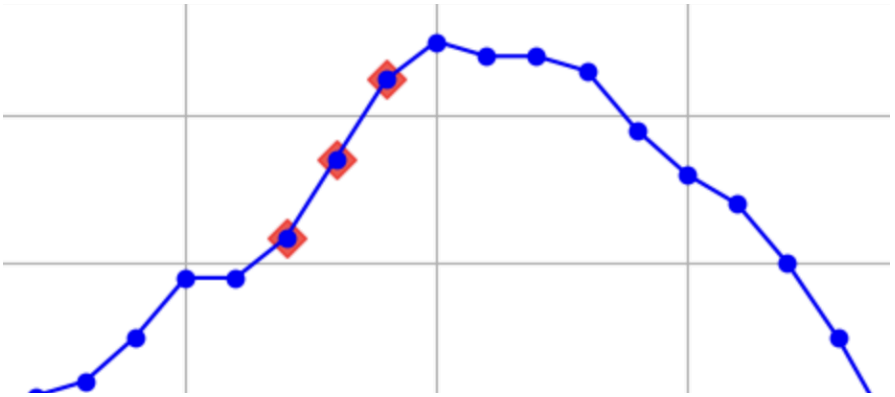
APPROACH -

We can divide our Approaches in two main categories-

1. Pre-Processing
2. Modelling

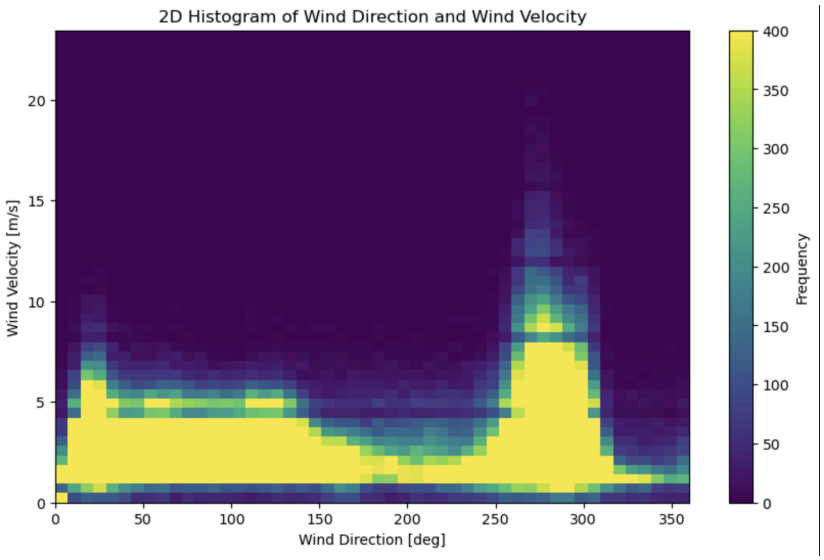
PRE-PROCESSING-

1. First we wanted to sort our data by date and time, but for that all the dates had to be converted into a single format.We did that and made a new column named “Datetime” which had values in format “YYYY-MM-DD Time” to sort our data upon.
2. Then we found only “total cloud cover” had 1400 missing values which we interpolated using cubic method

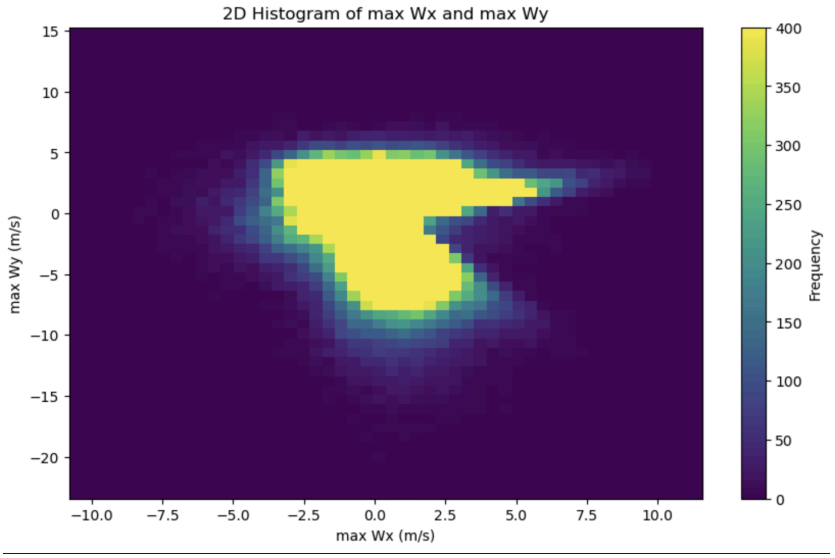


3. The column “ Azimuth Angle” had values in degrees, but the model cannot relate that 0 and 360 degrees are similar , so we converted those angles intro **sin** and **cosines**
4. Similarly, we had two columns for wind speed and wind direction, to relate these values we made new columns that represented a vector where direction was given by wind direction and magnitude of vector by wind speed

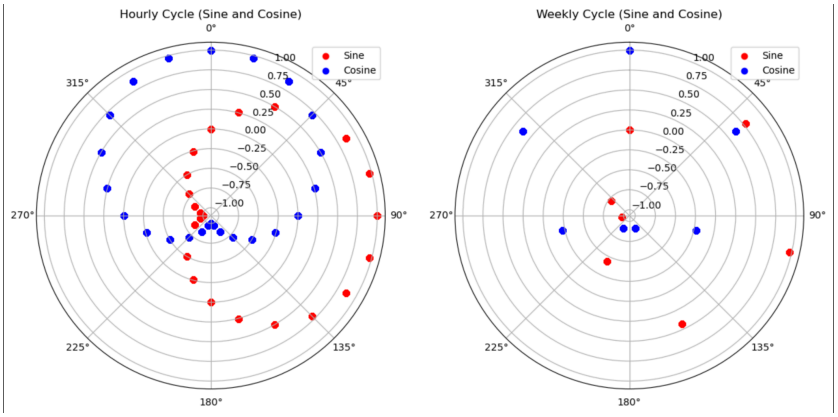
BEFORE-



AFTER-

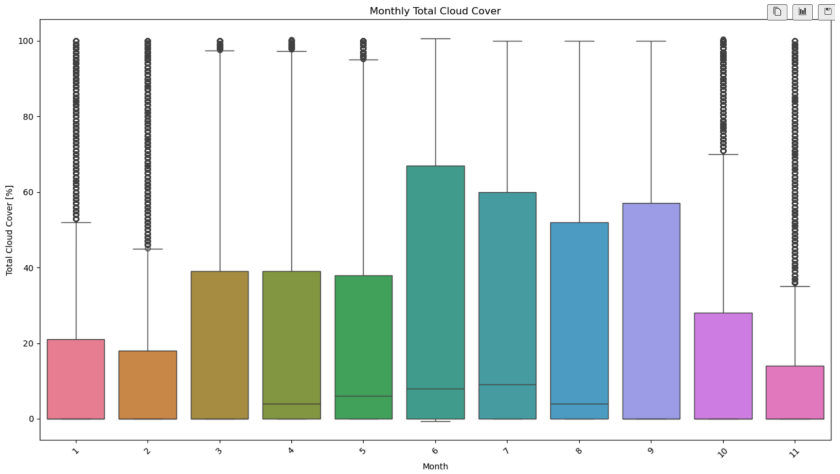


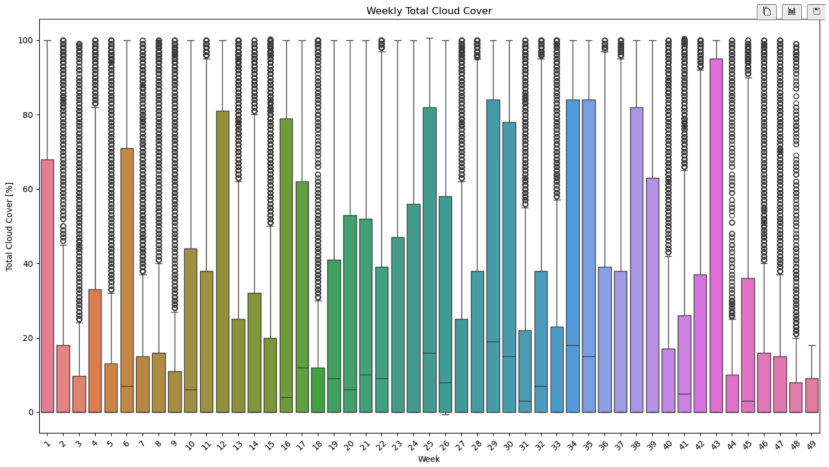
5. Next we made daily and weekly cyclic features so that we can capture “12pm had almost similar weather on 2 different days/weeks”

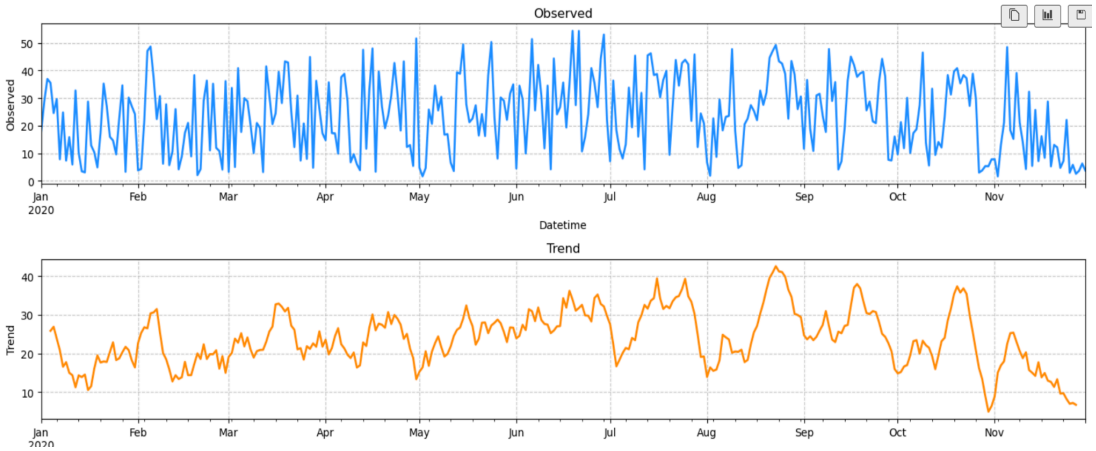


6. At last we found Outliers in the “Snow depth” column, we handled it by replacing the outliers by the maximum values we observed in majority
7. Also added column for lagged features , but the results depreciated.
8. Some more visuals-

9.

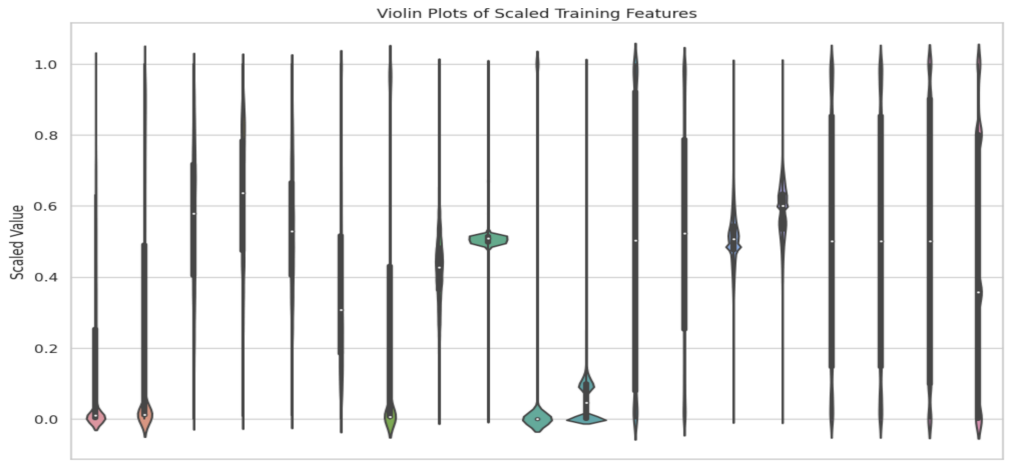






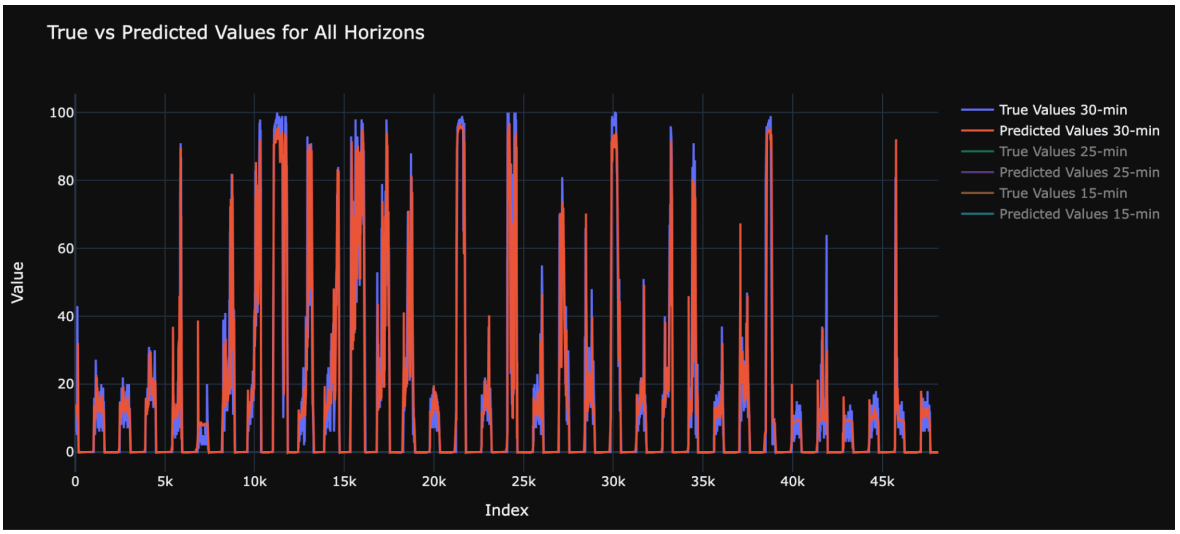
MODELS-

1. We first Scaled our data using MinMaxScaler and plotted a violin plot



2. Then we made a custom class using TimeSeriesGenerator to create datasets for 30,25,15 mins separately
3. We created a custom loss so that model doesn't always predict 0 , which was a majority value in our target column
4. The window size for input sequence was chose to be 75, This was based on our observations plotted using **ACF and PCF** plots, we did also try windows of 20 and 300 but they were not good as expected
5. Then using **Grid Search & For Loop** , tried various configurations for **LSTM** layers, that includes-
    a. Increasing LSTM layers (in number of neurons like 32,64,128 with different PnC)
    b. Decreasing LSTM layers (in number of neurons like 128,64,32 with different PnC)
    c. Increasing then Decreasing LSTM units
    d. Similarly, Increasing & Decreasing Dense Units
6. The **BEST MODEL** was with the one where we used a **LSTM(100)** along with **Dense(32) & Dense(64),** also we noticed that making the model deeper didn't make results better
7. After we found the best configuration, we applied **BILSTM, SIMPLE RNN, GRU, CONV1D-LSTM** to see any improvements
8. We trained a **Transformer** for the task with multihead attention and positional encoding. We modified the compiling process by using optimizers like AdamW.

9. At last we even tried **RESIDUAL MODELLING,** where we took results from LSTM and fed the Residuals (predicted-actual values) into another **LSTM, Linear Unit, XGBOOST** but the results were similar with little to no improvements
10. We also tried Residual Modelling with **ARIMA** but first we accidentally overfit the model by showing ARIMA our test values, this was realised and fixed but the training for ARIMA took **40 HOURS** to train, so the idea was discarded and we sticked to our previous results

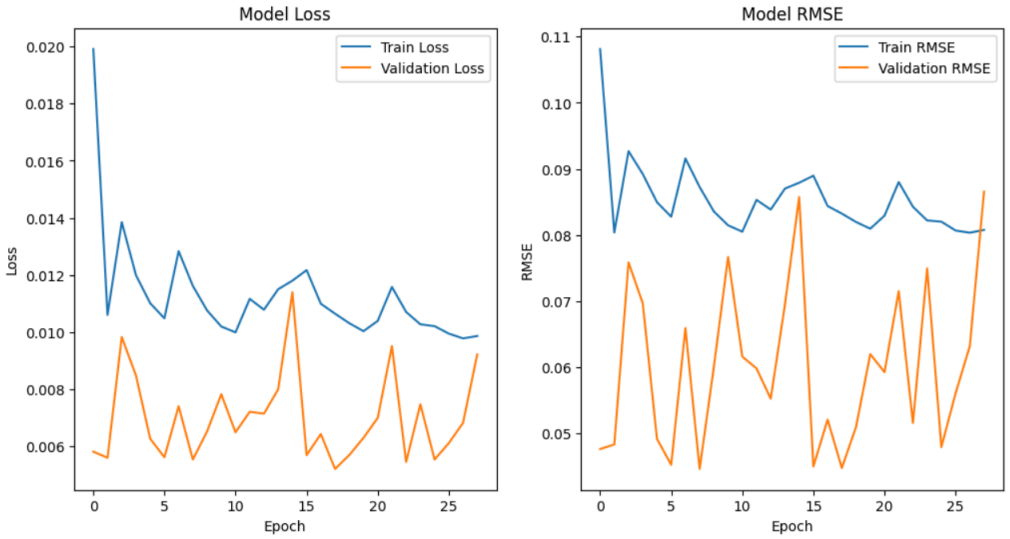


NOTE - WE HAVE MADE ALL PREDICTED INTERACTIVE GRAPHS AVAILABLE IN OUR NOTEBOOKS

RESULTS-

MODEL	WEIGHTED R2
LSTM (100)	0.913039910678336
RNN(100)	0.896709974203813 2

GRU(100)	**0.9104270234828695**
BILSTM	**0.9051736393252218**
CONV1D-LSTM	**0.9088082839366532**
LSTM + LINEAR UNIT	**0.9116687493190487**
LSTM + XGBOOST	**0.9104049360647528**
TRANSFORMER	**0.898098091875525 1**
RNN(only 30 min pred)	**0.9160694187042883**



Plotting History for Basic RNN

BLOCKERS -

- The dates were in different formats , some were in format “Jan-1” while some were in format “1-Jan”, which took time to handle
- Confusion on how to handle outliers
- Creating datasets for different timestamps, initially made data for predicted the next minute, but was resolved by custom class
- Loading the model while using custom metrics caused error and had to be fixed by using register_keras_serializable method
- While using backend from keras, faced a None Value attribute error. Resolved it by using functions directly from tensorflow.

FUTURE PROSPECTS -

- Ensembling with LSTM and ARIMA
- Some even more Feature Engineering like heatmap
- Hyperparameter Tuning (optimisers alternatives)