IBM MACHINE LEARNING PROFESSIONAL CERTIFICATE: TIME SERIES AND SURVIVAL ANALYSIS PROJECT.

By Dennis Kweku Botwe.

Main objective

The main objective of the project is to build a time series model that will be able to forecast the climate for the next 12 months. We will be using the LSTM method to find the most accurate model.

About dataset

The Dataset is fully dedicated to the developers who want to train the model on Weather Forecasting for the Indian climate. This dataset provides data from 1st January 2013 to 24th April 2017 in the city of Delhi, India. This is a purely academic dataset and is developed as a part of the Data Analytics course of 2019 at PES University, Bangalore. The dataset contains 1,576 rows and 5 columns

Dataset source: *https://www.kaggle.com/datasets/sumanthvrao/daily-climate-time-series-data?resource=download*

The 5 parameters here are:

date: Date of format YYYY-MM-DD

meantemp: Mean temperature averaged out from multiple 3-hour intervals in a day.

humidity: Humidity value for the day (units are grams of water vapour per cubic meter volume of air).

wind_speed: Wind speed measured in kmph

meanpressure: Pressure reading of weather (measure in atm)

Plan for Data exploration

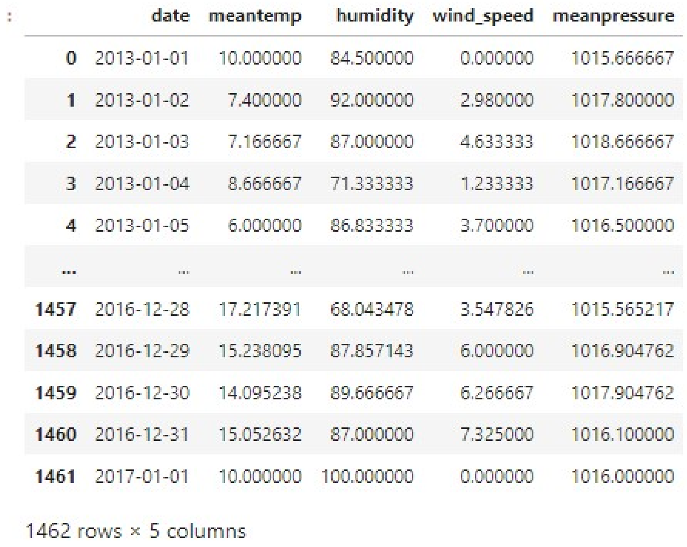
- Data Overview
- Decomposition
- Data Processing and Engineering
- Method of analysis for the model
- Evaluation of Models

- Insights and key findings
- Conclusion
- Suggestions And Next Steps

Data Overview:

- The data is split into Train and Test data frames.

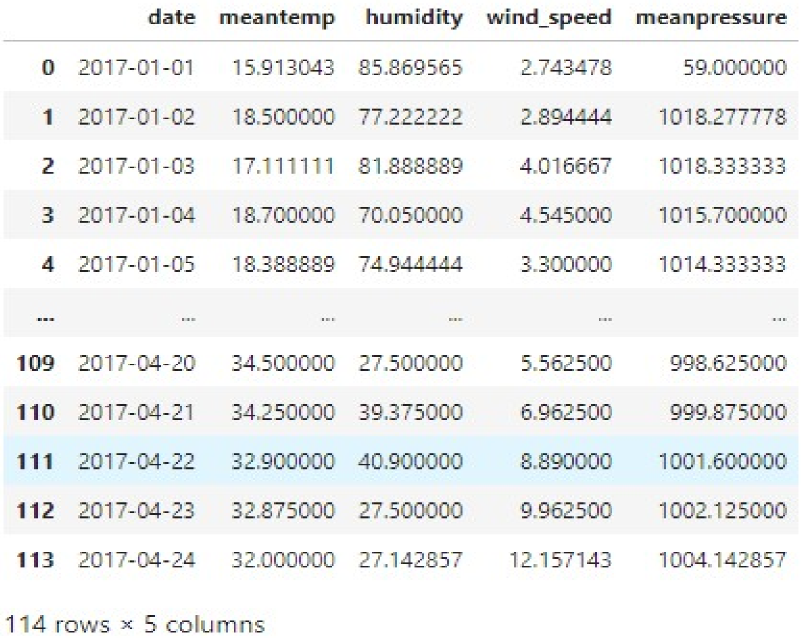
- The Data frame for the training dataset is shown below having 1462 rows and 5 columns:



	date	meantemp	humidity	wind_speed	meanpressure
0	2013-01-01	10.000000	84.500000	0.000000	1015.666667
1	2013-01-02	7.400000	92.000000	2.980000	1017.800000
2	2013-01-03	7.166667	87.000000	4.633333	1018.666667
3	2013-01-04	8.666667	71.333333	1.233333	1017.166667
4	2013-01-05	6.000000	86.833333	3.700000	1016.500000
...	...	...	...	...	...
1457	2016-12-28	17.217391	68.043478	3.547826	1015.565217
1458	2016-12-29	15.238095	87.857143	6.000000	1016.904762
1459	2016-12-30	14.095238	89.666667	6.266667	1017.904762
1460	2016-12-31	15.052632	87.000000	7.325000	1016.100000
1461	2017-01-01	10.000000	100.000000	0.000000	1016.000000

1462 rows × 5 columns

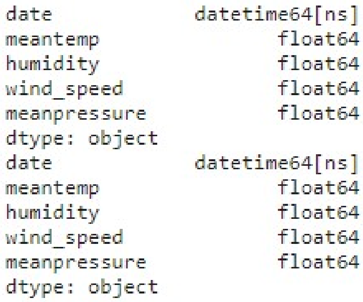
- The Data frame for the training dataset is shown below having 114 rows and 5 columns:



	date	meantemp	humidity	wind_speed	meanpressure
0	2017-01-01	15.913043	85.869565	2.743478	59.000000
1	2017-01-02	18.500000	77.222222	2.894444	1018.277778
2	2017-01-03	17.111111	81.888889	4.016667	1018.333333
3	2017-01-04	18.700000	70.050000	4.545000	1015.700000
4	2017-01-05	18.388889	74.944444	3.300000	1014.333333
...	...	...	...	...	...
109	2017-04-20	34.500000	27.500000	5.562500	998.625000
110	2017-04-21	34.250000	39.375000	6.962500	999.875000
111	2017-04-22	32.900000	40.900000	8.890000	1001.600000
112	2017-04-23	32.875000	27.500000	9.962500	1002.125000
113	2017-04-24	32.000000	27.142857	12.157143	1004.142857

114 rows × 5 columns

- It contains attributes date, meantemp, humidity, wind_speed, and pressure with data types as shown in the image below:

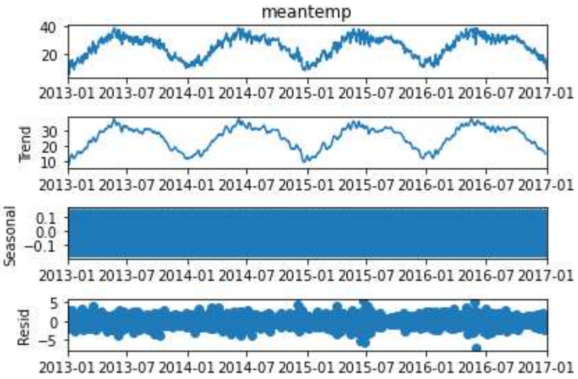


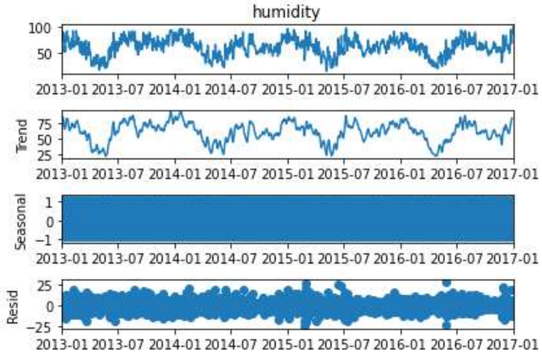
```
date            datetime64[ns]
meantemp               float64
humidity               float64
wind_speed             float64
meanpressure           float64
dtype: object
date            datetime64[ns]
meantemp               float64
humidity               float64
wind_speed             float64
meanpressure           float64
dtype: object
```

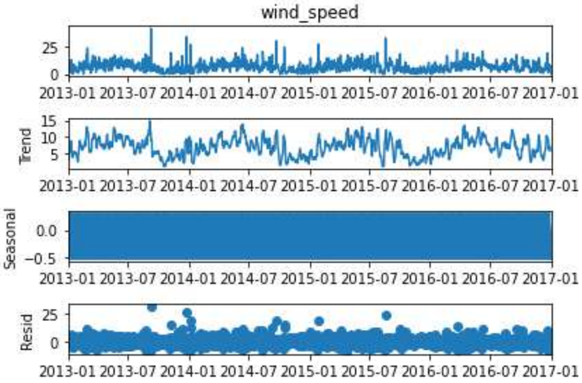
Decomposition

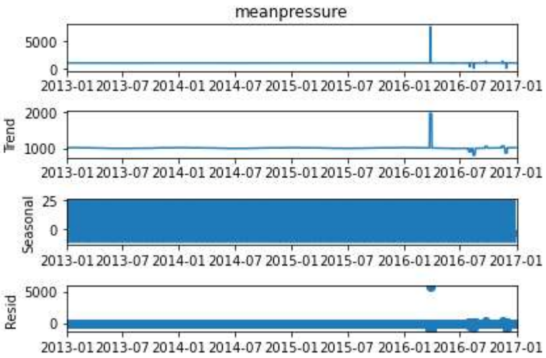
Decomposed the data into Observation, Trend, Seasonality, and Residual through time series decomposition for clarification.

It can be noted that there was trend and seasonality in the dataset as shown below:



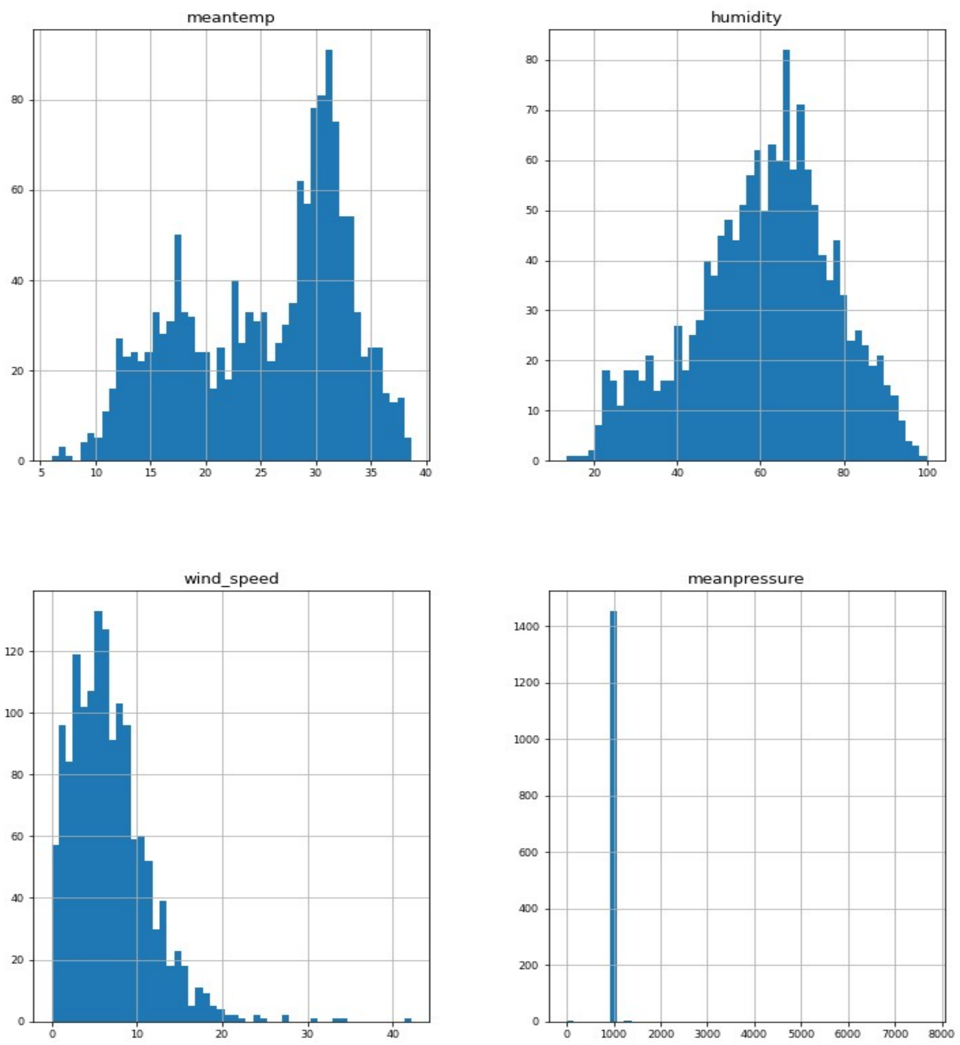






Data Processing and Engineering:

- Histogram of the column values:



- The dataset contains no missing values.
- The data was converted into NumPy arrays and split into train_series, test_series, train_time, and test_time with data shapes of (1462,), (114,), (1462,) and (114,) respectively.

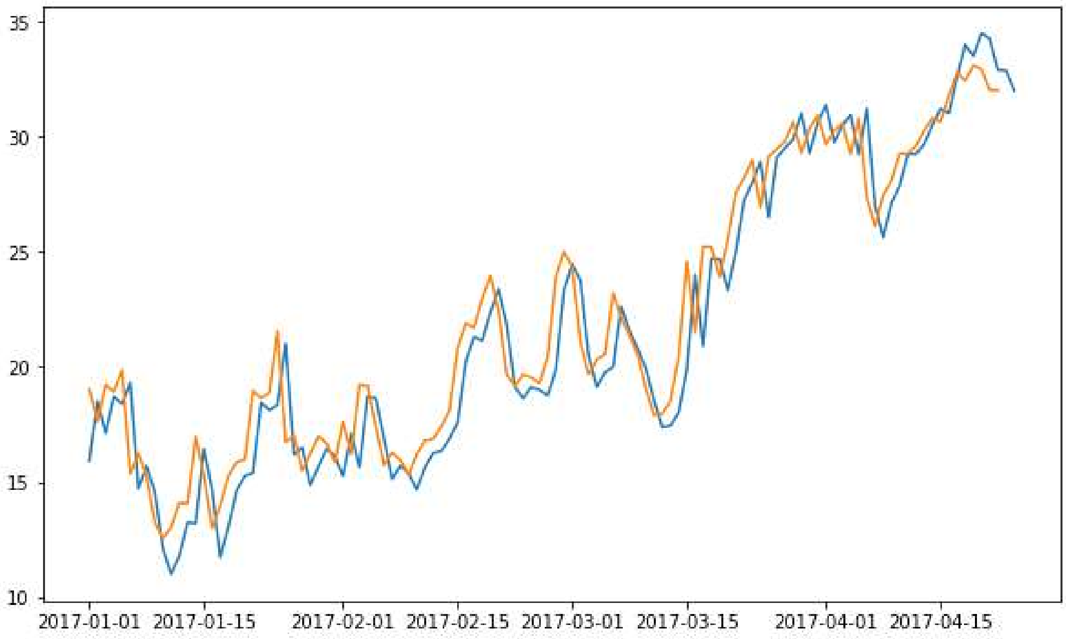
Method of analysis for the model

Long-Short Term Memory is going to be used as the model to predict the forecast since some of the trained values will be stored in memory and then later used as a point of reference to predict the next steps. The use of RNN will be the model since it lacks the capabilities of storing data in memory. The models will be evaluated and the best model will be selected.

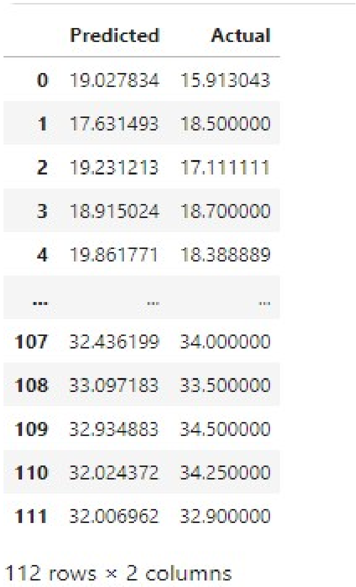
Model Evaluation

Model 1:

The first LSTM model trained used a 1D CNN (often used in time series analysis), with depth and kernel size set to 32 and 3 respectively, and ReLU as an activation function. The model had two sets of 32 nodes in each hidden layer and 1 dense output layer with return sequences set to true. Also, the SGD Optimizer was used with a learning rate of 1e-4 and momentum of 0.9 after running 100 epochs. After running the summary, the model had a total and trainable parameters of 16,801



It can be observed that the results fitted very well as seen above and the predicted value can be seen in the table below with a Mean Absolute Error of 1.3984083

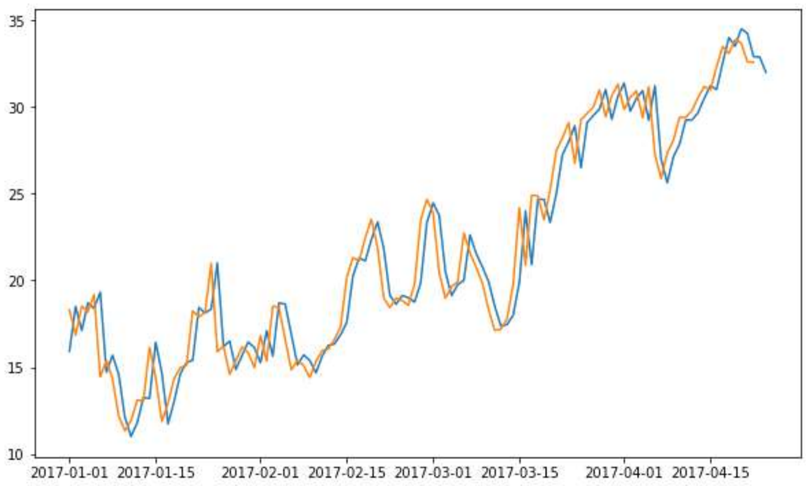


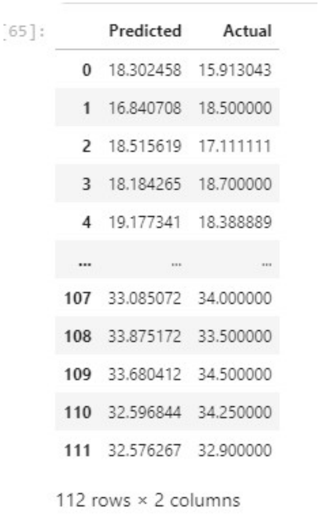
	Predicted	Actual
0	19.027834	15.913043
1	17.631493	18.500000
2	19.231213	17.111111
3	18.915024	18.700000
4	19.861771	18.388889
...	...	...
107	32.436199	34.000000
108	33.097183	33.500000
109	32.934883	34.500000
110	32.024372	34.250000
111	32.006962	32.900000

112 rows × 2 columns

Model 2:

The second model is no different from the first but this time with some slight tweaks. The depth of the 1D CNN was adjusted to 16 and the two sets of 16 nodes in the hidden layers. After running the summary, the model had a total and trainable parameters of 4,305. It can be observed that the results fitted very well and the predicted value can be seen in the table below with a Mean Absolute Error of 1.337945.



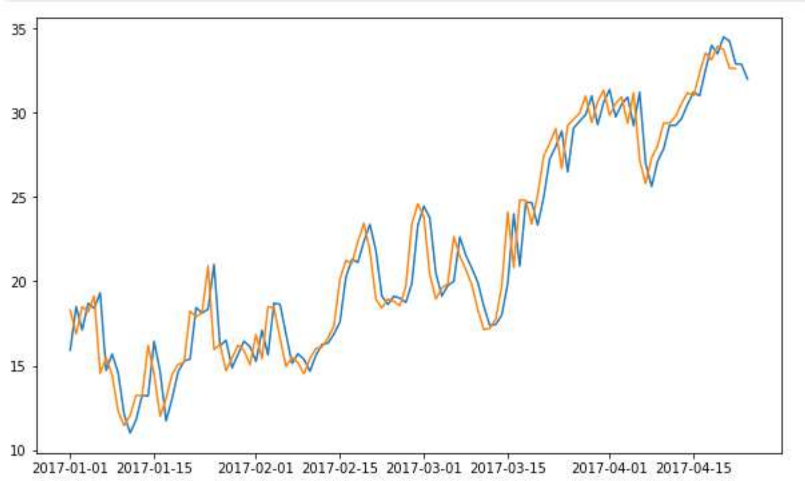


	Predicted	Actual
0	18.302458	15.913043
1	16.840708	18.500000
2	18.515619	17.111111
3	18.184265	18.700000
4	19.177341	18.388889
...	...	...
107	33.085072	34.000000
108	33.875172	33.500000
109	33.680412	34.500000
110	32.596844	34.250000
111	32.576267	32.900000

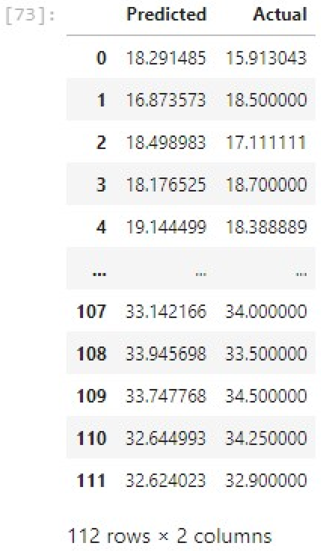
112 rows × 2 columns

Model 3:

For the final LSTM model to be trained, a 1D CNN was used with depth and kernel size set to 16 and 3 respectively, and ReLU was used for the activation function. There were two hidden layers with 16 nodes each and one dense output layer with true return sequences. This time, the Adam Optimizer was used with a learning rate of 1e-4 for running 100 epochs. After running the summary, the model had a total and trainable parameters of 4,305. As shown in the plots below it can be observed there are no major changes as it fits the train set but it slightly performed better in terms of Mean Absolute Error, getting a rate of 1.3289496.



And checking from the prediction table it can be seen that the model made some significant changes.



[73]:

	Predicted	Actual
0	18.291485	15.913043
1	16.873573	18.500000
2	18.498983	17.111111
3	18.176525	18.700000
4	19.144499	18.388889
...	...	...
107	33.142166	34.000000
108	33.945698	33.500000
109	33.747768	34.500000
110	32.644993	34.250000
111	32.624023	32.900000

112 rows × 2 columns

Model 3, the Chosen Model

With Mean Absolute Error being the evaluation matrix and also keeping in mind that the model how well the model fits the data, the final model is selected as the best one out of the three that were trained. As said, in the final LSTM model to be trained, a 1D CNN was used with depth and kernel size set to 16 and 3 respectively, and ReLU was used for the activation function. There were two hidden layers with 16 nodes each and one dense output layer with true return sequences. This is a fairly simple model with MAE of 1.3289496 and made a very close prediction compared to Model 1 and 2 with MAE of 1.3984083 and 1.337945 respectively.

Key Findings and Insights

All the models did very well but taking into consideration the Mean Absolute Error and how they fit the data, it can be said model 1 seems a little off and it had the highest MAE which is bad since the higher the MAE, the lower the performance of the model. For models 2 and 3, there wasn't any major difference but then again Model 3 outperformed the models based on MAE and was selected as the ideal model. In general, the predictions of all 3 models are not accurate enough to build a business model in their current forms.

Conclusion

Three models were evaluated carefully and as said model 3 was selected due to its performance based on Mean Absolute Error and the model's ability to fit well with the data.

Suggestions and Next Steps

Many regularization techniques can be used to improve the accuracy of the model. I suggest Gated Recurrent Unit (GRU) since it trains quicker and produces about the same results as LSTM models. And also since the dataset is small, GRU will do very well.