Climate Time Series Analysis

CIVE7100 35340 Time Ser/Geospatial Data Sci SEC V30 Spring 2022

Soumyo Dey

Northeastern University

(dey.soum@northeastern.edu)

ABSTRACT

The use of science and technology to predict the state of the atmosphere for a specific location is known as weather forecasting. Observed patterns of events, also known as pattern recognition, were commonly used in ancient weather forecasting methods. However, not all of these predictions turn out to be correct. Weather forecasting is the most difficult problem on the planet. There are a variety of reasons for this, including its experimental values in meteorology, but it's also a common unbiased time series forecasting challenge in scientific study. Various scientists have offered a variety of ways. The goal of research is to improve prediction accuracy. In this project we have used different models namely Arima and LSTM to help predict Delhi weather. First, the data was cleaned using different techniques after which both Arima and LSTM model was used to predict and infer which model gives better result. We will help expand our knowledge in the application of Arima and LSTM model in the domain of weather forecasting.

INTRODUCTION

Delhi's climate is intense in both directions. As a result, the primary goal of this project is to aid in the comprehension of underlying trends or systemic patterns across time. Weather forecasting is important because it assists in predicting future climatic changes. Using data visualizations, users can see seasonal trends and dig deeper into why these trends occur.

When considering the benefits that Arima and LSTM models provide, there is room for improvement in the accuracy of their forecasts. The idea of combining human intelligence with the above-mentioned models can significantly increase weather forecasting accuracy. The characteristics that define weather conditions, such as minimum and maximum temperatures, average rainfall, and so on, change with time, generating time series of each parameter that can be utilized to construct a forecasting model using this data.

Climate time series analysis is particularly difficult because it is very hard to predict the future weather conditions accurately. A number of factors make it difficult to obtain accurate results from climate time series. Some of these factors are: Even with a perfect model and increased observations, there are numerous unpredictable atmospheric fluctuations. Uncertainties or errors in data can amplify as the models run realizations to try and predict the weather conditions.

DATASET

datetime_utc	_conds	_dewptm	_fog	_hail	_heatindexm	_hum	_precipm	_pressurem	_rain	_snow	_tempm	_thunder	_tornado	_vism	_wdird	_wdire
1996-11-01 11:00:00	Smoke	9.0	0	0	NaN	27.0	NaN	1010.0	0	0	30.0	0	0	5.0	280.0	West
1996-11-01 12:00:00	Smoke	10.0	0	0	NaN	32.0	NaN	-9999.0	0	0	28.0	0	0	NaN	0.0	North
1996-11-01 13:00:00	Smoke	11.0	0	0	NaN	44.0	NaN	-9999.0	0	0	24.0	0	0	NaN	0.0	North
1996-11-01 14:00:00	Smoke	10.0	0	0	NaN	41.0	NaN	1010.0	0	0	24.0	0	0	2.0	0.0	North

This dataset contains hourly weather data collected in the city of Delhi from the period of 21 years (from 1996 to 2017). There are in total 20 columns and 1,00,991 rows.
The columns are as follows –

1. datetime_utc – Tells the date(year, month and day) and the time
2. _conds – Tells the condition of the weather
3. _dewptm – Tells the dewpoint in C
4. _fog – Tells whether there is fog or not.
5. _hail – Tells whether there is hail or not.
6. _heatindexm – Tell us the heat index in C.
7. _hum – Tells us the humidity in weight of water vapor per unit weight of air.
8. _precipm – Tells us the precipitation.
9. _pressurem – Tells us the air pressure.
10. _rain – Tells whether there is rain or not.
11. _snow – Tells whether there is snow or not.
12. _tempm – Tells us the temperature in C.
13. _thunder – Tells us whether there is thunder or not.
14. _tornado – Tells us whether there is tornado or not.
15. _vism – Tells us the visibility.
16. _wdird – Tells us the wind angle.
17. _wdire – Tells us the wind direction.
18. _wgustm – Tells us the gust speed.
19. _windchill – Tells us the chilliness in wind.
20. _wspdm – Tells us the wind speed.

DATA PREPROCESSING

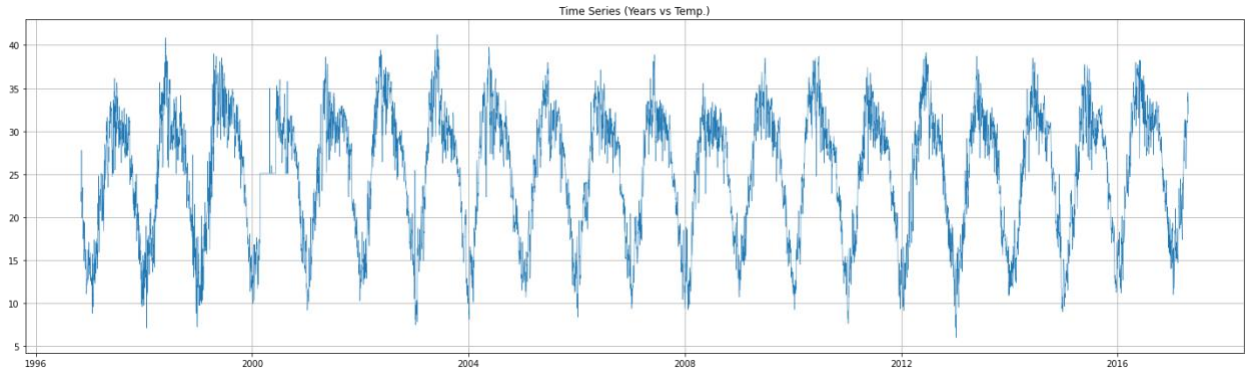
Once we had our data, we had to clean our raw data in order to work with it. We first looked to see if there were any missing values, and although there were very few, we added the mean value in that row. After that, we looked for NAN values and replaced them with 0. Final step was to look for extreme values and remove.

	humidity	temprature
count	100990.000000	100990.000000
mean	57.957422	25.438222
std	23.821218	8.487994
min	4.000000	1.000000
25%	39.000000	19.000000
50%	59.000000	27.000000
75%	78.000000	32.000000
max	243.000000	90.000000

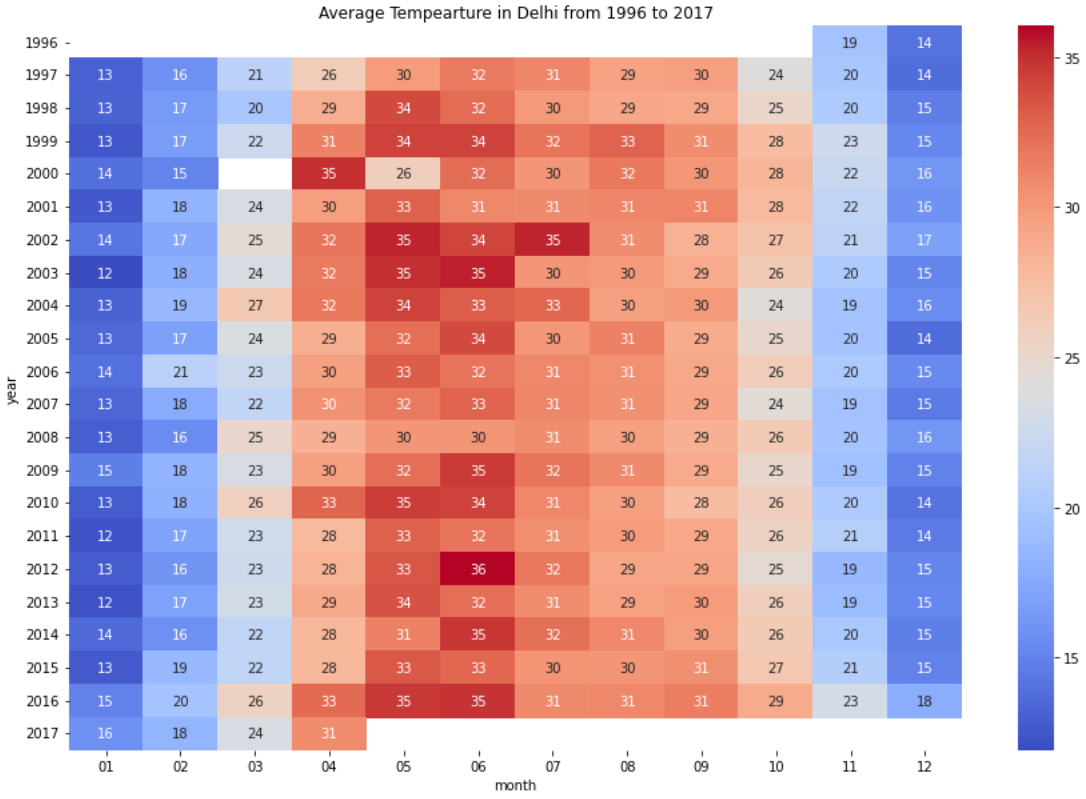
Here we can clearly see that max humidity and temperature is 243 and 90 degrees C respectively, which are extreme values as we cannot have such values in real life. Thus, we removed all the temperature values above 50 degrees C and humidity above 100 to get a more realistic data.

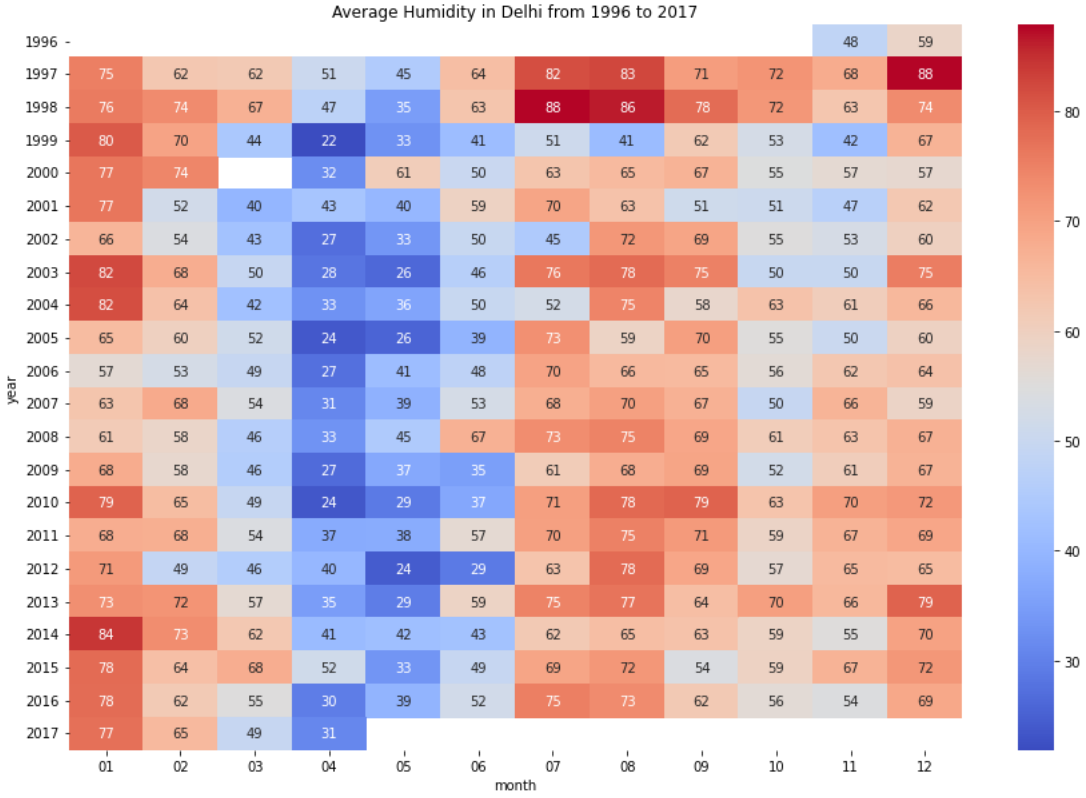
DATA VISUALIZATION

One of the most important aspects of this project is to help visualize our data, look out for trends in time series and finally infer from them.



To start of I find the plot of the time series of Temperature during the time period of 1996 – 2017. We can clearly see that the trend is quite common for all the years.

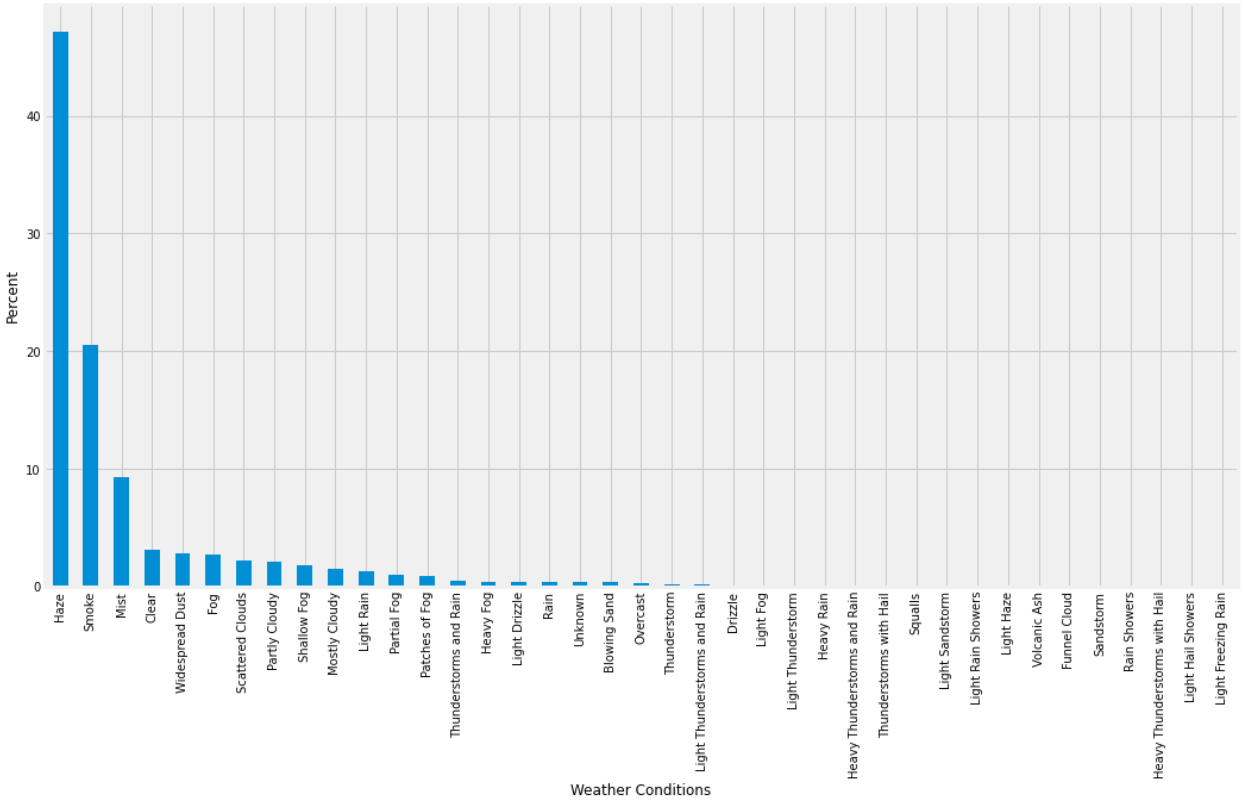

Average Tempearture in Delhi from 1996 to 2017
year
month
1996 19 14
1997 13 16 21 26 30 32 31 29 30 24 20 14
1998 13 17 20 29 34 32 30 29 29 25 20 15
1999 13 17 22 31 34 34 32 33 31 28 23 15
2000 14 15 35 26 32 30 32 30 28 22 16
2001 13 18 24 30 33 31 31 31 31 28 22 16
2002 14 17 25 32 35 34 35 31 28 27 21 17
2003 12 18 24 32 35 35 30 30 29 26 20 15
2004 13 19 27 32 34 33 33 30 30 24 19 16
2005 13 17 24 29 32 34 30 31 29 25 20 14
2006 14 21 23 30 33 32 31 31 29 26 20 15
2007 13 18 22 30 32 33 31 31 29 24 19 15
2008 13 16 25 29 30 30 31 30 29 26 20 16
2009 15 18 23 30 32 35 32 31 29 25 19 15
2010 13 18 26 33 35 34 31 30 28 26 20 14
2011 12 17 23 28 33 32 31 30 29 26 21 14
2012 13 16 23 28 33 36 32 29 29 25 19 15
2013 12 17 23 29 34 32 31 29 30 26 19 15
2014 14 16 22 28 31 35 32 31 30 26 20 15
2015 13 19 22 28 33 33 30 30 31 27 21 15
2016 15 20 26 33 35 35 31 31 31 29 23 18
2017 16 18 24 31
01 02 03 04 05 06 07 08 09 10 11 12
35 30 25 20 15


Average Humidity in Delhi from 1996 to 2017
year
month
1996 48 59
1997 75 62 62 51 45 64 82 83 71 72 68 88
1998 76 74 67 47 35 63 88 86 78 72 63 74
1999 80 70 44 22 33 41 51 41 62 53 42 67
2000 77 74 32 61 50 63 65 67 55 57 57
2001 77 52 40 43 40 59 70 63 51 51 47 62
2002 66 54 43 27 33 50 45 72 69 55 53 60
2003 82 68 50 28 26 46 76 78 75 50 50 75
2004 82 64 42 33 36 50 52 75 58 63 61 66
2005 65 60 52 24 26 39 73 59 70 55 50 60
2006 57 53 49 27 41 48 70 66 65 56 62 64
2007 63 68 54 31 39 53 68 70 67 50 66 59
2008 61 58 46 33 45 67 73 75 69 61 63 67
2009 68 58 46 27 37 35 61 68 69 52 61 67
2010 79 65 49 24 29 37 71 78 79 63 70 72
2011 68 68 54 37 38 57 70 75 71 59 67 69
2012 71 49 46 40 24 29 63 78 69 57 65 65
2013 73 72 57 35 29 59 75 77 64 70 66 79
2014 84 73 62 41 42 43 62 65 63 59 55 70
2015 78 64 68 52 33 49 69 72 54 59 67 72
2016 78 62 55 30 39 52 75 73 62 56 54 69
2017 77 65 49 31
01 02 03 04 05 06 07 08 09 10 11 12
80 70 60 50 40 30

The above two heatmaps shows us the variation of average temperature and humidity for all the 12 months (x-axis) during the period of 1996 – 2017 (y-axis).

For the first heatmap, we can clearly see that the summer season (June, July, and August) has had the highest temperature throughout, and that the mean temperature has climbed from 30 degrees C in 1996 to 35 degrees C in 2017, thus illustrating the rise of global warming.

For the second heatmap, we immediately see that the rainy season has the highest humidity, but over also the mean humidity has increased over time indicating rise in air pollution.



The above plot shows the most common weather conditions of Delhi. As ‘Haze’ and ‘Smoke’ dominate the most, it prominently indicates how bad the air pollution level is in Delhi.

MODEL

In this project we have used two different types of models to see which gives us the better prediction.

1. ARIMA MODEL

The ARIMA model (autoregressive integrated moving average) is a generalization of the ARMA model (basic autoregressive moving average). Both of these models are used to anticipate or predict future time-series data points. ARIMA is a type of regression analysis that shows how strong a dependent variable is in relation to other variables that change. It predicts future values based on past values. ARIMA makes use of lagged moving averages to smooth time series data.

According to the name, we can split the model into smaller components as follow:

1.AR - a form of random process represented by an AutoregRegressive model The model's output is linearly related to its own previous value, i.e. the number of lagged data points or past observations.

2.MA - a Moving Average model whose output is linearly dependent on the current and previous observations of a stochastic term.

3.I - The differencing phase to construct stationary time series data, i.e. removing the seasonal and trend components, is referred to as integrated here.

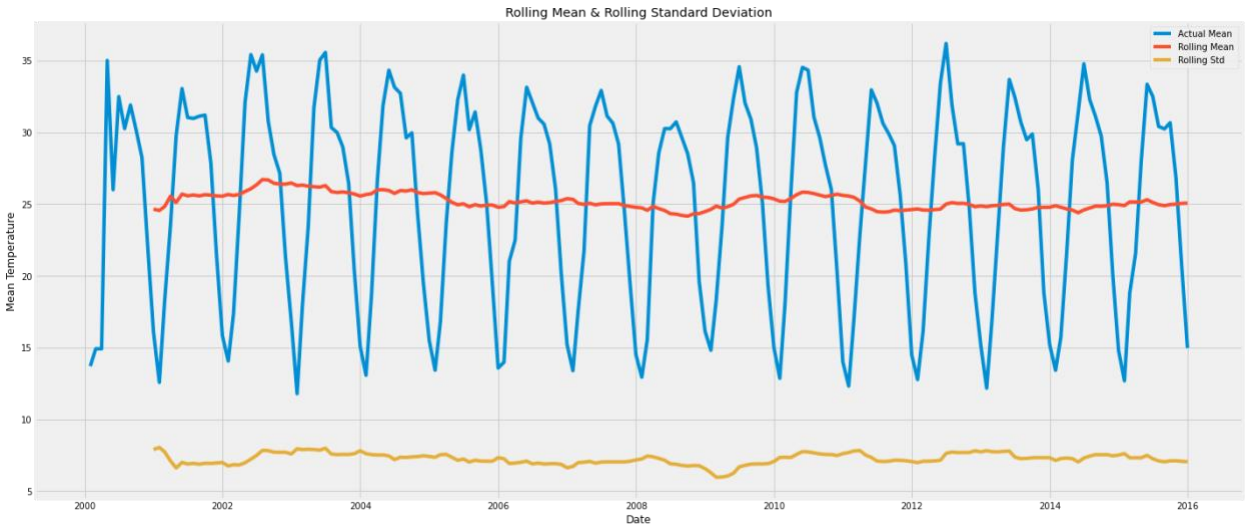
ARIMA model is generally denoted as ARIMA(p, d, q) and parameter p, d, q are defined as follow:

a) p: the lag order or the number of time lag of autoregressive model AR(p)

b) d: degree of differencing or the number of times the data have had subtracted with past value

c) q: the order of moving average model MA(q)

Let's start with finding the stationarity

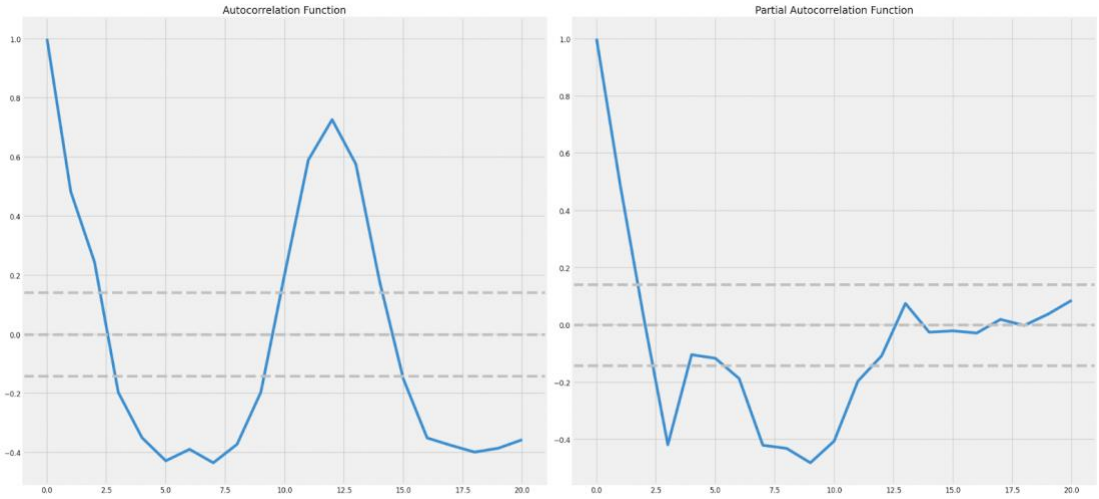


We can clearly see the rolling mean and standard deviation is constant implying the time series is stationary. **Thus the 'd' value is 0**.

Next up we find the autocorrelation and partial autocorrelation

1.ACF: The autocorrelation coefficient function, define how the data points in a time series are related to the preceding data points.

2. PACF: The partial autocorrelation coefficient function, like the autocorrelation function, conveys vital information regarding the dependence structure of a stationary process.



In the autocorrelation graph we see that the first gray line cuts the line graph at value 2. Thus, our **p-value becomes 2**. Similarly in the partial autocorrelation graph we can see that the first gray line cuts the line graph also at 2. Thus, our **q-value also becomes 2**.

Once we have our 'p', 'q' and 'd' value we can start with our ARIMA model.

ARMA Model Results

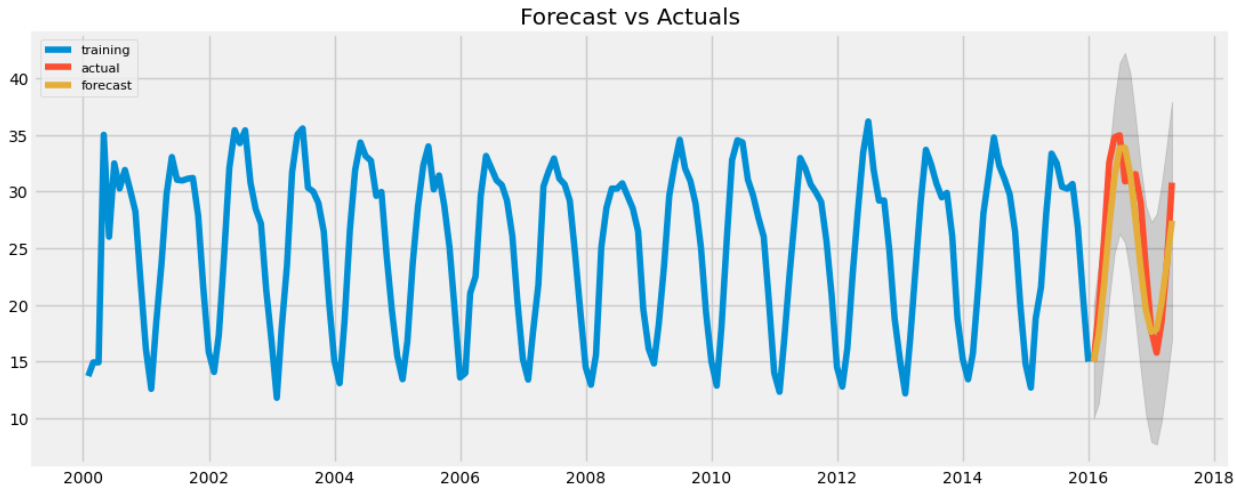
Dep. Variable:	y	No. Observations:	192
Model:	ARMA(2, 2)	Log Likelihood	-454.355
Method:	css-mle	S.D. of innovations	2.552
Date:	Mon, 25 Apr 2022	AIC	920.709
Time:	00:27:58	BIC	940.254
Sample:	0	HQIC	928.625

	coef	std err	z	P>|z|	[0.025	0.975]
const	25.1917	0.119	211.046	0.000	24.958	25.426
ar.L1.y	1.6785	0.024	69.835	0.000	1.631	1.726
ar.L2.y	-0.9519	0.023	-41.164	0.000	-0.997	-0.907
ma.L1.y	-0.9726	0.098	-9.919	0.000	-1.165	-0.780
ma.L2.y	0.1453	0.090	1.618	0.107	-0.031	0.321

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	0.8816	-0.5227j	1.0250	-0.0852
AR.2	0.8816	+0.5227j	1.0250	0.0852
MA.1	1.2685	+0.0000j	1.2685	0.0000
MA.2	5.4264	+0.0000j	5.4264	0.0000

This is our model summary.



This is our final prediction using ARIMA.

Final Result –

Prediction of temperature –

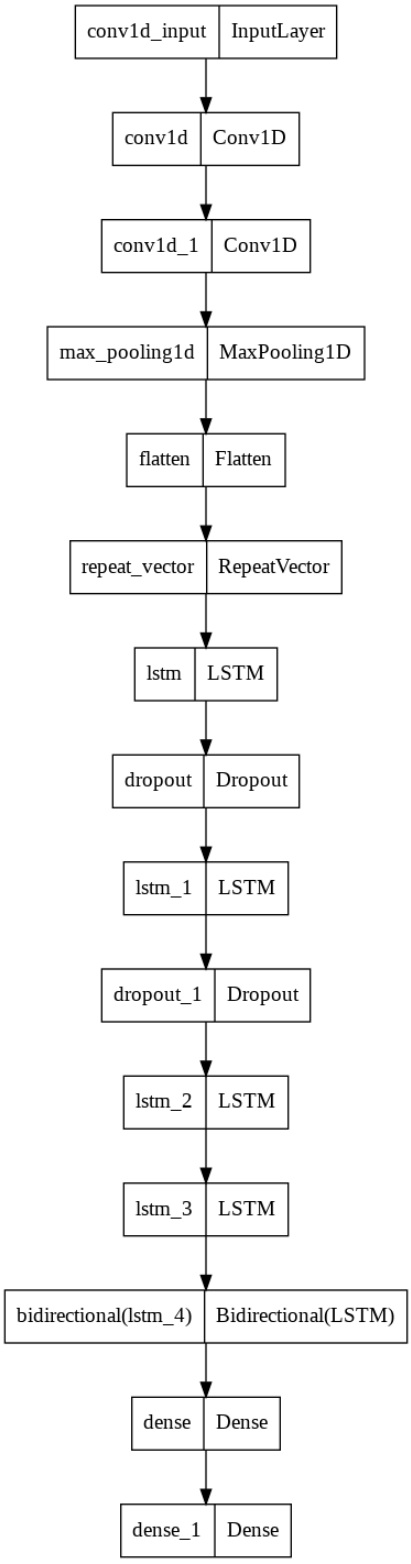
Mean of temperature - 25.438222

RMSE - 3.1057635856013266

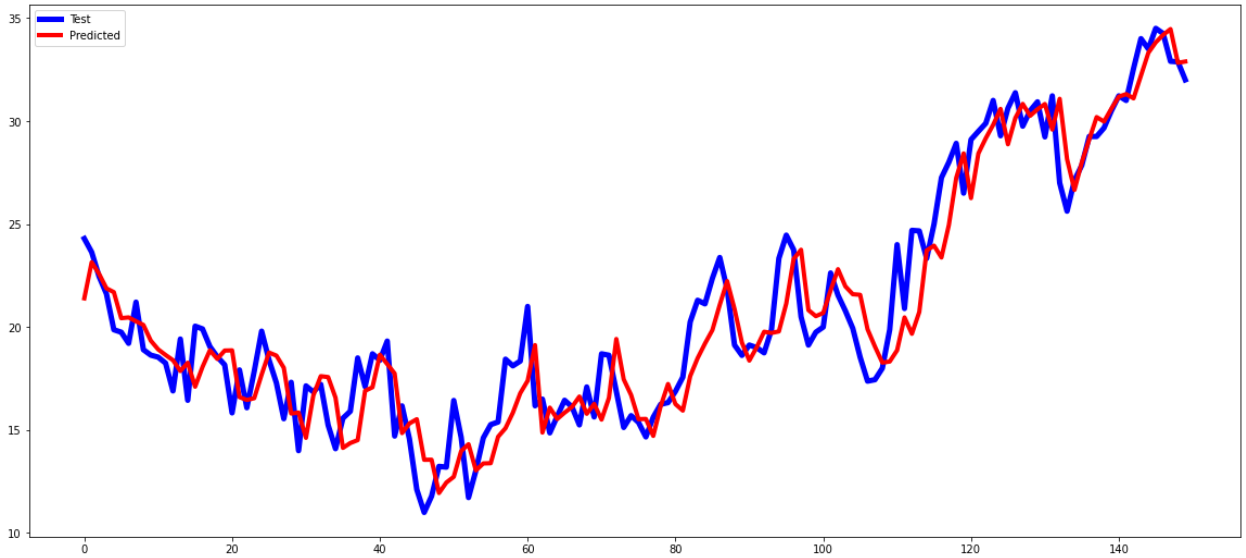
Error - 12.209043484%

We got a 12.2% error, which is quite good for a weather prediction. Let's see if we can improve upon that using LSTM.

2. LSTM MODEL


conv1d_input
InputLayer
conv1d
Conv1D
conv1d_1
Conv1D
max_pooling1d
MaxPooling1D
flatten
Flatten
repeat_vector
RepeatVector
lstm
LSTM
dropout
Dropout
lstm_1
LSTM
dropout_1
Dropout
lstm_2
LSTM
lstm_3
LSTM
bidirectional(lstm_4)
Bidirectional(LSTM)
dense
Dense
dense_1
Dense

This is our LSTM Model framework. This particular framework gave us the best results. We used 3 conv layers, 1 maxpool layer, 1 flatten layer , 1 repeat vector , 2 dropout layers, 3 LSTM layers and finally 2 dense layers.



This is our final prediction using LSTM.

Final Result -

Prediction of temperature -

Mean of temperature - 25.438222

RMSE - 1.847351078574147

Error - 7.26210346%

We got an error of 7.2% which comes out to be better than ARIMA model

CONCLUSION

We can easily observe from the above results that LSTM outperforms ARIMA on the dataset.

This could be due to the model's limitation, which is that ARIMA can only assess the linear portion. However, the non-linear element of the data may not be white noise, which means that the ARIMA model may miss some information. LSTM is a type of RNN and deep learning application that is designed to learn temporal patterns, capture non-linear dependencies, and preserve useable memory for a longer period of time, resulting in superior results in scenarios when the dataset is large.

BIG PICTURE

Accurate weather prediction can help in various fields -

- To help people take proper precautions to secure themselves and their families in case of unwanted occurrences.
- Organizations can work better with the help of accurate weather predictions and it helps to deliver visual forecasts by various methods that most companies prefer.
- Weather forecasting highly benefits the agriculture sector for buying/selling livestock. It also assists the farmers to decide when to plant crops, pastures, and when to irrigate.
- It provides the business with valuable information that the business can use to make decisions about future business strategies.

Accurate weather forecasting can provide information to people and organizations that can utilize it to reduce weather-related losses and improve societal advantages such as life and property protection, public health and safety, and economic prosperity and quality of life.

CITATIONS

1. Comparison of ARIMA and LSTM for prediction of hemorrhagic fever at different time scales in China - Rui Zhang,Hejia Song,Qiulan Chen,Yu Wang,Songwang Wang ,Yonghong Li (January 14 2022)
2. Data Science for Weather Prediction – The Prerequisite to all Natural Disasters
3. How to Develop LSTM Models for Time Series Forecasting - Jason Brownlee (November 14 2018)
4. How to Create an ARIMA Model for Time Series Forecasting in Python - Jason Brownlee (January 9 2017)