

# George Washington University



**Time series Analysis and Modeling  
DATS 6313  
Final Term Project Report  
10 May 2023**

**Submitted BY: Nusrat J Prithi**  
**Submitted To: Prof. Reza Jafari**

## Table of Contents

Table of figures and tables.....	3
Abstract. ....	4
Objective .....	5
Introduction.....	6
Data Cleaning and Analysis.....	6
Modeling .....	13
ARIMA and SARIMA.....	13
Multi Linear Regression .....	13
Deep Learning (LSTM) .....	17
H-test Predictions Visualization: .....	17
Forecast Function .....	20
Summary and Conclusion: .....	21

## Table of figures and tables

Figure 1: First 5 rows .....	5
Figure 2: Missing Values .....	5
Figure 3: Descriptive Statistics .....	6
Figure 4: Meantemp between 2013-2017 .....	6
Figure 5: Plots of All features .....	7
Figure 6: Meanpressure Outlier fixed .....	8
Figure 7: PACF .....	8
Figure 8: Heatmap .....	9
Figure 9: Pearson Correlation Matrix .....	9
Figure 10: Decomposition .....	10
Figure 11: ADF test output.....	12
Figure 12: Rolling Mean and variance.....	12
Figure 13: ADF test output .....	12
Figure 14: rolling mean and variance.....	12
Figure 15: Feature Importance.....	13
Figure 16: Baseline Models.....	13
Figure 17: ARIMA and SARIMA Evaluations.....	13
Figure 18: Multi-Linear Regression.....	14
Figure 19: 1 Step Prediction.....	14
Figure 20: F test output.....	14
Figure 21: T test output.....	15
Figure 22: Multi-Linear Regression .....	15
Figure 23: ACF Residuals.....	15
Figure 24: Q-Variance.....	16
Figure 25: Variance and Mean residual.....	16
Figure 26: LSTM Summary.....	17
Figure 27: LSTM Evaluation.....	17
Figure 28: H-test Prediction on SARIMA.....	18
Figure 29: Arima h-test.....	19
Figure 30: LSTM h-test.....	19
Figure 31: SARIMA Forecast Function.....	20

**Abstract.**

The main aim for this project is to explore various Machine Learning algorithms with a sole purpose of analyzing and developing models for Indian climate. This project will have the following parts; data loading, data cleaning and visualization, modeling and evaluation. Some of the models we are going to train here are ARIMA, SARIMA, and LSTM.

### Objective

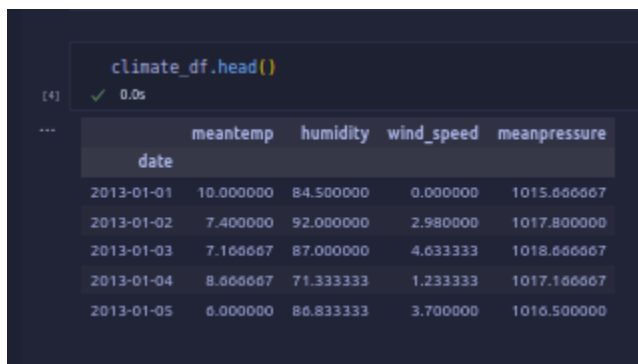
The main objective of this project is to predict future meantemp in Delhi.

## Introduction.

Time series refers to tracing data incrementally over a period of time. Data points in time series are recorded sequentially. Time series applies in every aspect of life since time is everywhere. For instance, an e-commerce firm can record daily sales and use this data in future to predict profits. In our case, we are using time series data to predict Indian climate between 2013 and 2017. This report will contain analysis of time series, cleaning, feature engineering, performing various modeling algorithms such as ARIMA and LSTM, and forecasting.

## Data Cleaning and Analysis.

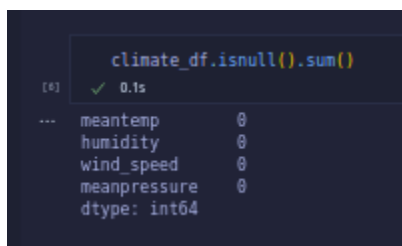
I am using DailyDelhiClimateTrain.csv which has date, meantemp, humidity, wind\_speed, and meanpressure columns. The date is of type object while the other three features of the type float64. While working with this data, we are going to use set date as index and then work with other 4 features. Below is a section of data we will be working with:



date	meantemp	humidity	wind_speed	meanpressure
2013-01-01	10.000000	84.500000	0.000000	1015.666667
2013-01-02	7.400000	92.000000	2.980000	1017.800000
2013-01-03	7.166667	87.000000	4.633333	1018.666667
2013-01-04	8.666667	71.333333	1.233333	1017.166667
2013-01-05	6.000000	86.833333	3.700000	1016.500000

Figure 1: First 5 rows

The csv has no missing values as shown in the figure 2 below.



Column	Count
meantemp	0
humidity	0
wind_speed	0
meanpressure	0
dtype: int64	

Figure 2: Missing Values

When we perform descriptive statics in our data, we observe that the data is well balanced. Figure 3 below shows the descriptive statics;

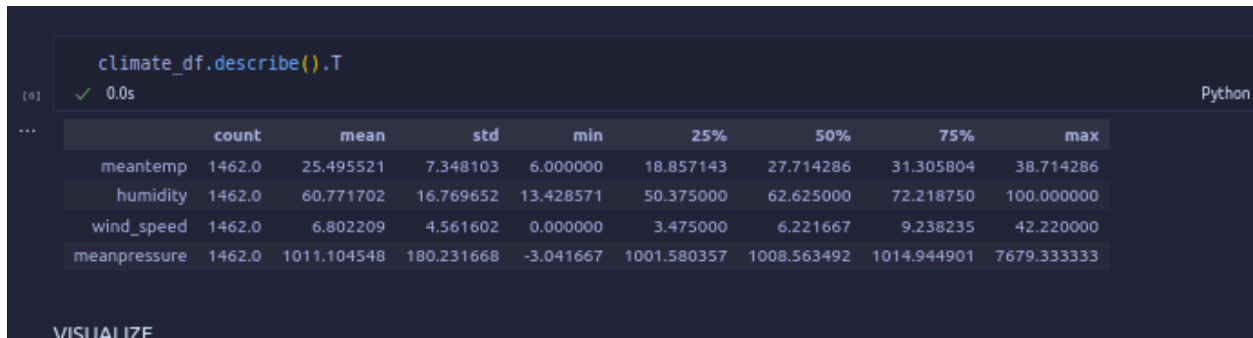


Figure 3: Descriptive Statistics

Let's now do some plots on dependent variables with date.

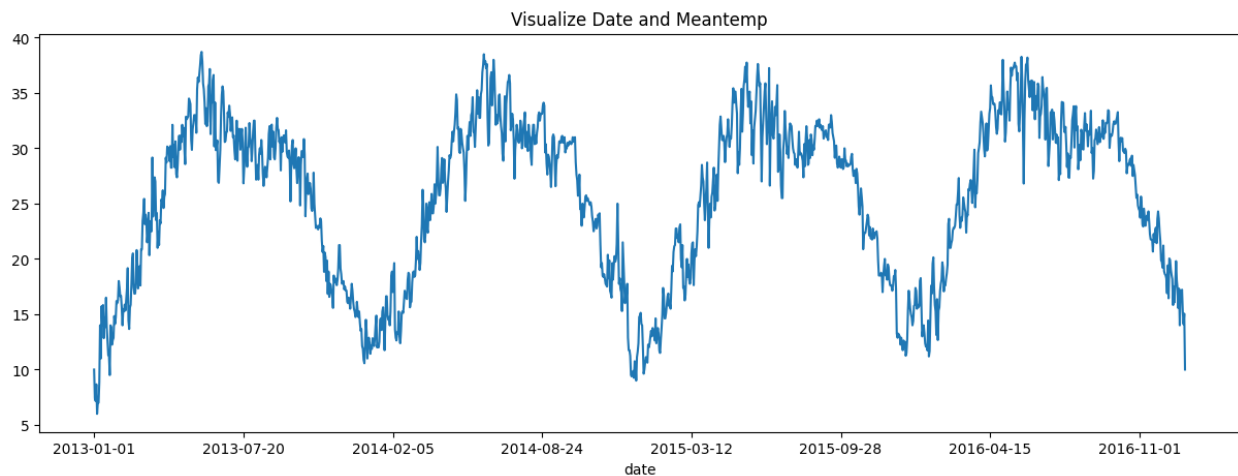
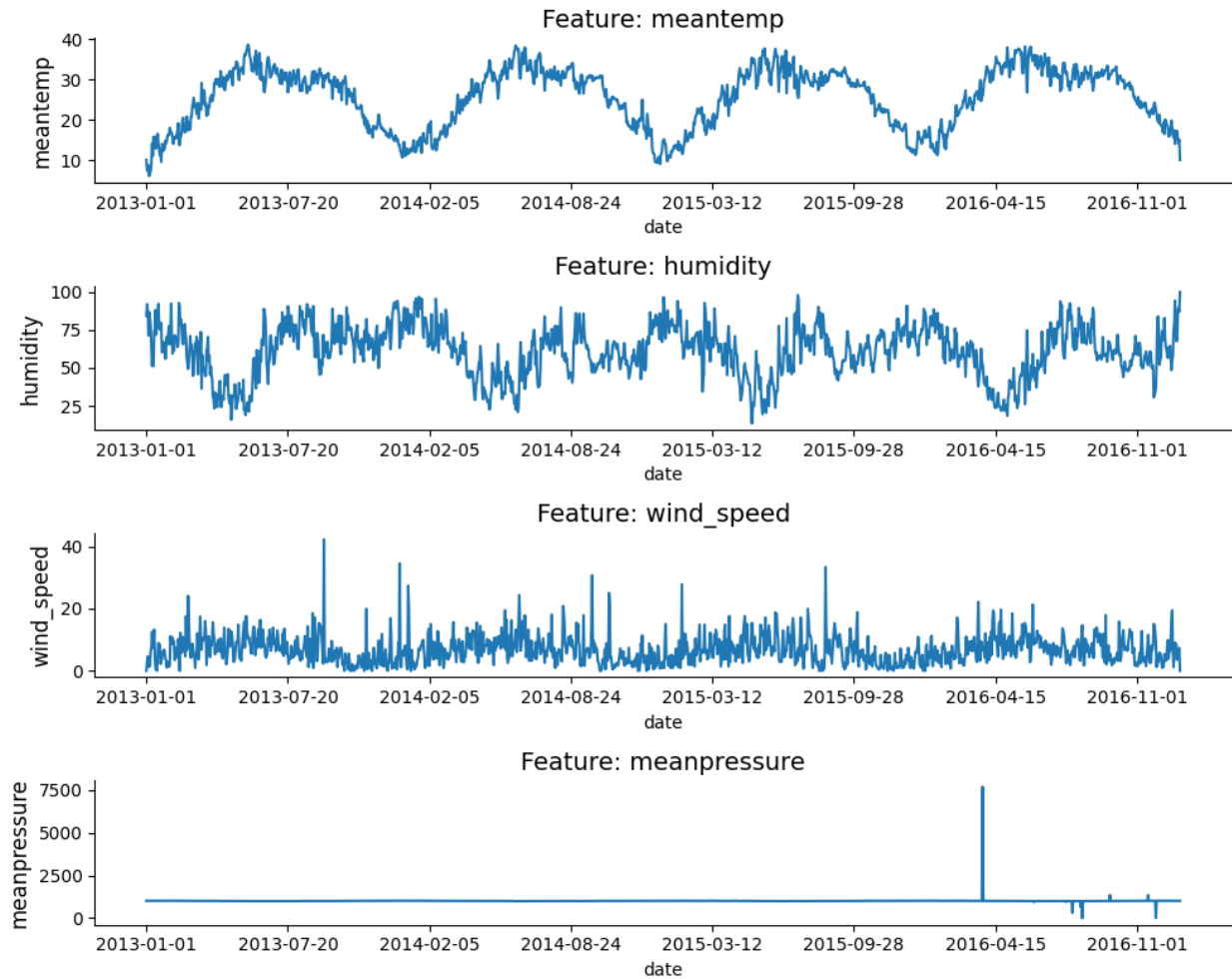


Figure 4: Meantemp between 2013-2017

From Figure 4 above, we observe that temperature rises exponentially in the first quarter of the year, remains high during the second quarter, seems to be constant during third quarter, and starts to reduce during the last quarter. This is the case in all yearly between 2013 and 2027.



*Figure 5: Plots of All features*

From figure 5 above, we observe that:

The time series has constant variation and is stationary. Notably, meanpreassure seems to have some abnormal behavior between 2016 and 2017. This abnormality is due to the fact there are outliers in the time series.

The plot below shows the outlier problem fixed from the meanpressure features. This is because the plot seems to have assumed normal behaviour as humidity, meantemp, and wind\_speed.



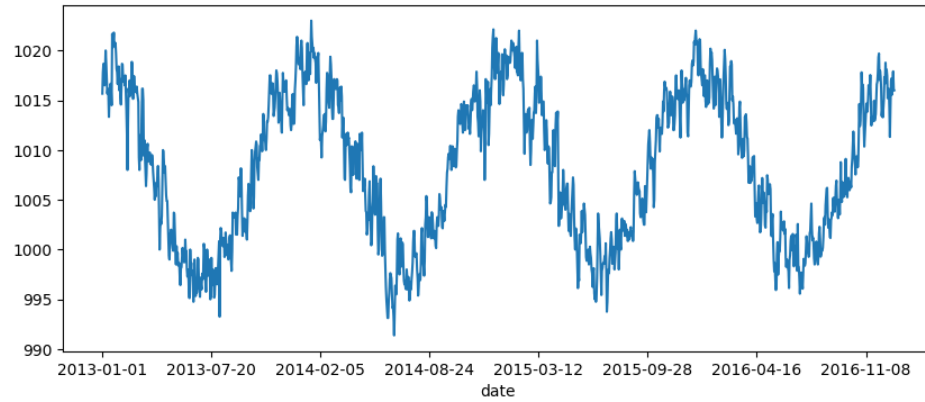


Figure 6: Meanpressure Outlier fixed

PACF of the dependent variable is as shown below.

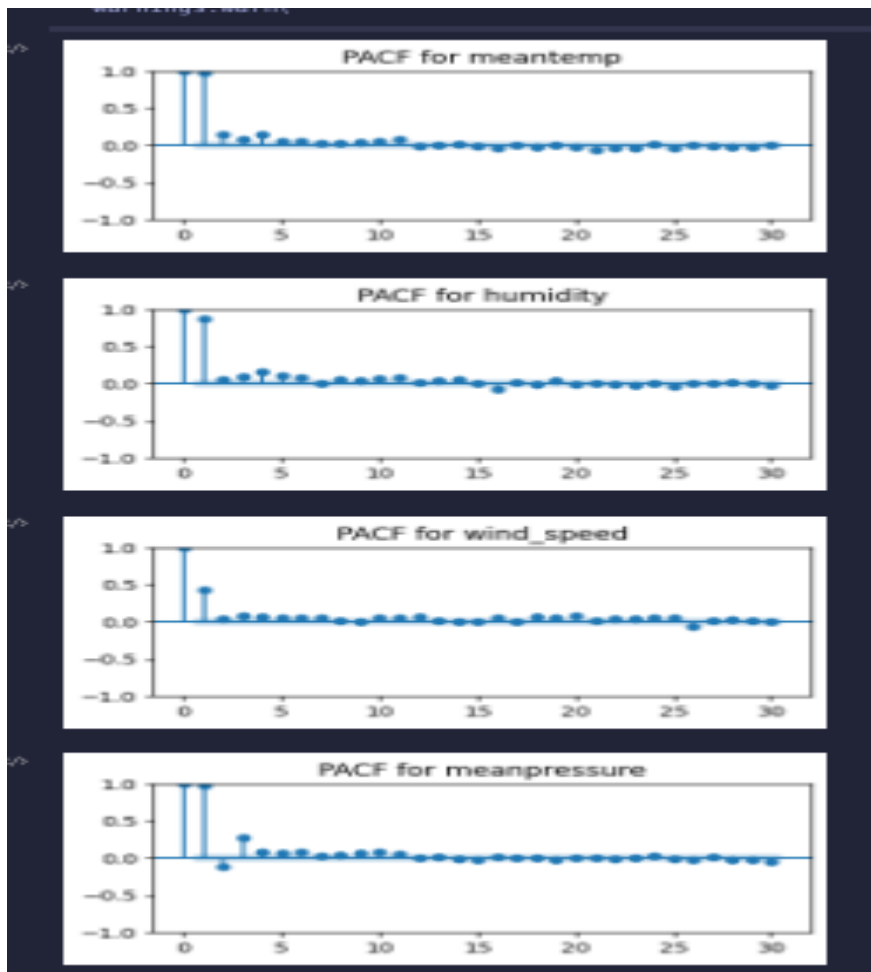


Figure 7: PACF

### PACF observation:

From PACF of the dependent variables above, we can see that, correlation decays to zero and this shows that it is constant and stationery.

Let's now have a look heatmap and Pearson Correlation matrix:

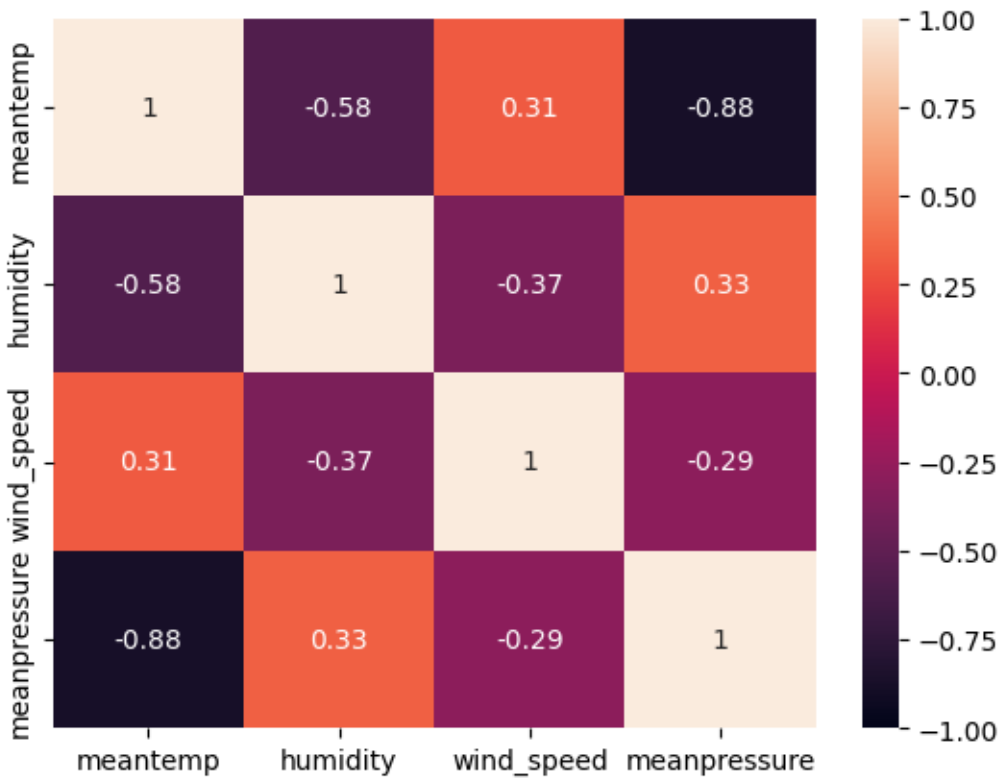


Figure 8: Heat map

```

Pearson Correlation Matrix
meantemp  humidity  wind_speed  meanpressure
meantemp  1.000000 -0.575328  0.307407  -0.878969
humidity  -0.575328  1.000000  -0.374039  0.332819
wind_speed 0.307407 -0.374039  1.000000  -0.294541
meanpressure -0.878969 0.332819 -0.294541  1.000000

```

Figure 9: Pearson Correlation Matrix

Meantemp has a strong negative correlation with meanpressure (-0.88), indicating that as temperature increases, air pressure tends to decrease. Meantemp has a moderate negative correlation with humidity (-0.57), indicating that as temperature increases, humidity tends to decrease. Humidity has a moderate negative correlation with wind\_speed (-0.37), indicating that as humidity increases, wind speed tends to decrease.

There is a weak positive correlation between wind\_speed and meantemp (0.31), indicating that as temperature increases, wind speed tends to increase slightly. There is a weak negative correlation between wind\_speed and meanpressure (-0.29), indicating that as air pressure increases, wind speed tends to decrease slightly. There is no strong correlation between humidity and meanpressure, wind\_speed and meanpressure, or humidity and wind\_speed.

### Seasonal Decomposition:

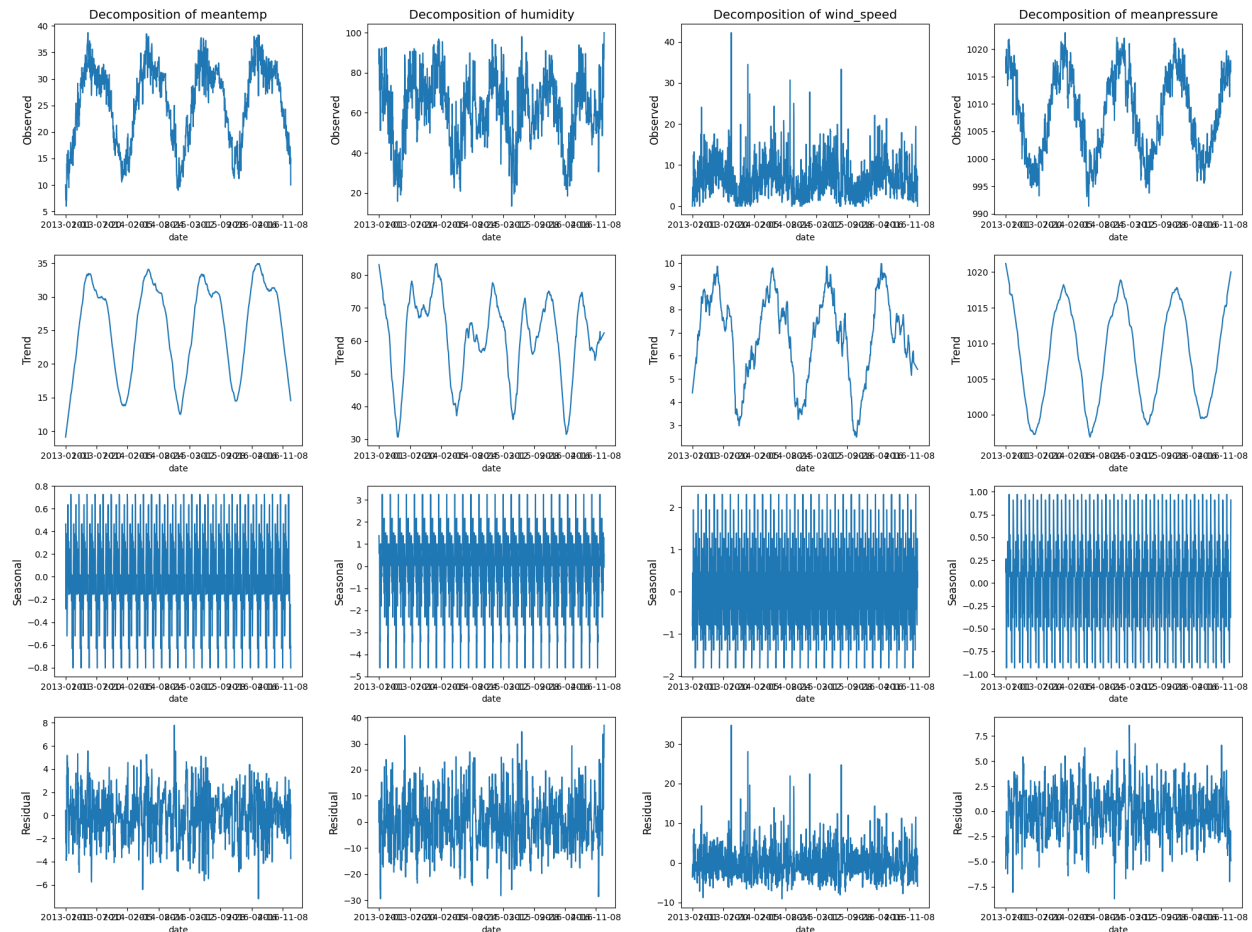


Figure 10: Decomposition

### Stationary:

### ADF Test of Stationary:

```

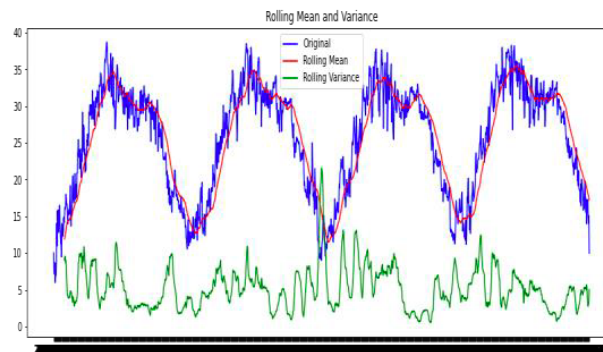
ADF Statistic: -2.021069055920673
p-value: 0.2774121372301602
Critical Values:
  1%: -3.4348647527922824
  5%: -2.863533960720434
 10%: -2.567831568508802

```

*Figure 11: ADF test output*

The data is not stationary as the ADF statistic value (-2.010274418658002) is greater than the critical values at all levels of significance (1%, 5%, and 10%). Also, the p-value (0.2820863747671887) is greater than the significance level (0.05). Therefore, we cannot reject the null hypothesis that the data is non-stationary.

### **Rolling Mean and Variance:**



*Figure 12: Rolling Mean and variance*

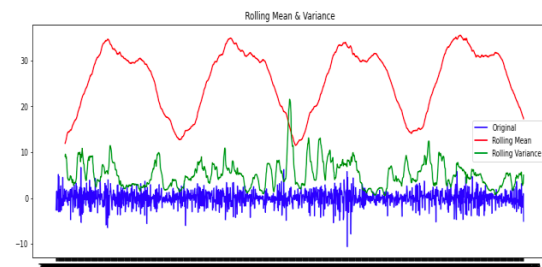
The rolling and variance of figure above are not constant. Means non stationary.

### **Transform The data into stationary:**

To transform the data stationary to non stationary first order differentiation has been performed

```

ADF Statistic: -16.352650052172265
p-value: 2.922236116679744e-29
Critical Values:
  1%: -3.4348929812602784
  5%: -2.863546418485167
 10%: -2.5678382024888378
  
```



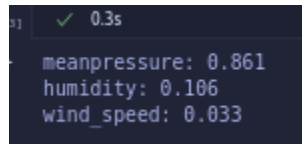
*Figure 13: ADF test output and variance*

*Figure 14: rolling mean*

After 1st order differentiation the data become stationary. Where p value is less than 0.05. Rolling mean and variance also become stable. Data is now stationary.

### Feature Selection:

When we check feature importance for, meanpressure and humidity seems to have more effect on the target variable.



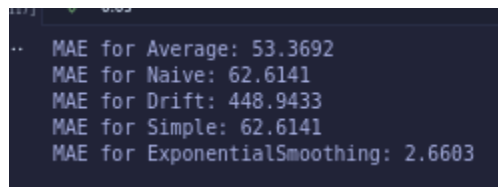
```

111 | ✓ 0.3s
    |
    | meanpressure: 0.861
    | humidity: 0.106
    | wind_speed: 0.033
  
```

*Figure 15: Feature Importance*

### Modeling

In this part, I did base models, ARIMA and SARIMA, and LSTM. For base models, I fit the data in the average, naïve, drift, and simple algorithms. I then evaluated the respective models using Mean Squared Error (MSE) and the following were the results.



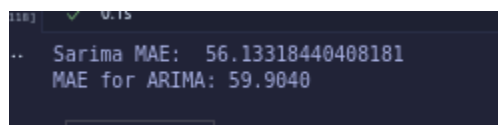
```

112 | ✓ 0.0s
    |
    | .. MAE for Average: 53.3692
    |    MAE for Naive: 62.6141
    |    MAE for Drift: 448.9433
    |    MAE for Simple: 62.6141
    |    MAE for ExponentialSmoothing: 2.6603
  
```

*Figure 16: Baseline Models*

### ARIMA and SARIMA

The following figure shows evaluation of ARIMA and SARIMA.



```

113 | ✓ 0.1s
    |
    | .. Sarima MAE: 56.13318440408181
    |    MAE for ARIMA: 59.9040
  
```

*Figure 17: ARIMA and SARIMA Evaluations*

When we compare evaluated values between Baseline models and SARIMA, we find out that, the only model that performs better than SARIMA is Average and Exponential Smoothing.

### Multi Linear Regression

I did a multi-linear modeling on a time series dataset and got an accuracy of 91.74% as shown in the figure below:

```
print(f"Multilinear Regression: {accuracy_mtl:.2f} %")
✓ 0.0s
Multilinear Regression: 91.74 %
```

*Figure 18: Multi-Linear Regression*

I performed 1 step prediction and the following were the results:

```
print(f"True value: {y_true_one_step}, predicted value: ")
[122] ✓ 0.0s
... True value: 10.0, predicted value: [15.51054496]
```

*Figure 19: 1 Step Prediction*

When we compare the true value and predicted value in Figure 15 above, we observe that there is a difference of 5 which is not very large.

## Hypothesis Test:

### F-Test:

```
✓ # Perform the F-test on the test set ...
F-value: [1071.47695909 1039.21620434 1135.62043608 1589.00237373 2901.53024912]
P-value: [2.18611058e-098 6.70773977e-097 3.06009576e-101 2.40489568e-118
4.11552784e-151]
```

*Figure 20: F test output*

The hypothesis test indicates strong evidence to reject the null hypothesis. Large F-values suggest the regression model fits the data well and there is a significant relationship between the variables. Small p-values indicate low probability of observing such large F-values by chance, providing strong evidence against the null hypothesis. Overall, the results suggest the regression model is a good fit for the data.

### T-Test:

The output of the T-test indicates that there is strong evidence to reject the null hypothesis. The t-values are large and the p-values are very small, indicating that each independent variable has a significant effect on the dependent variable. Overall, the results suggest that the independent variables are important predictors of the dependent variable.



### Q-Variance results:

The Q variance Lagrange multiplier statistic has a value of 79.05. The p-value associated with the Q statistic is  $2.77\text{e-}16$ , indicating strong evidence against the null hypothesis. The F-statistic has a value of 21.47. The p-value associated with the F-statistic is  $3.45\text{e-}18$ , which is very small and indicates strong evidence against the null hypothesis.

```
... Lagrange multiplier statistic: 79.04935789256066
    P-value: 2.76929752471959e-16
    F-statistic: 21.46682836217308
    F p-value: 3.449810172558677e-18
```

*Figure 24: Q-Variance*

From the Q-variance, we can see that p-value is greater than 0.05, implying that our time series is stationary.

### Variance and Mean Residual Results:

The mean of the residuals is 0.001859, which indicates that on average, the residuals are close to zero. The residual variance is 2.68, which indicates the spread or dispersion of the residuals around the regression line. A low residual variance indicates that the predicted values are close to the actual values.

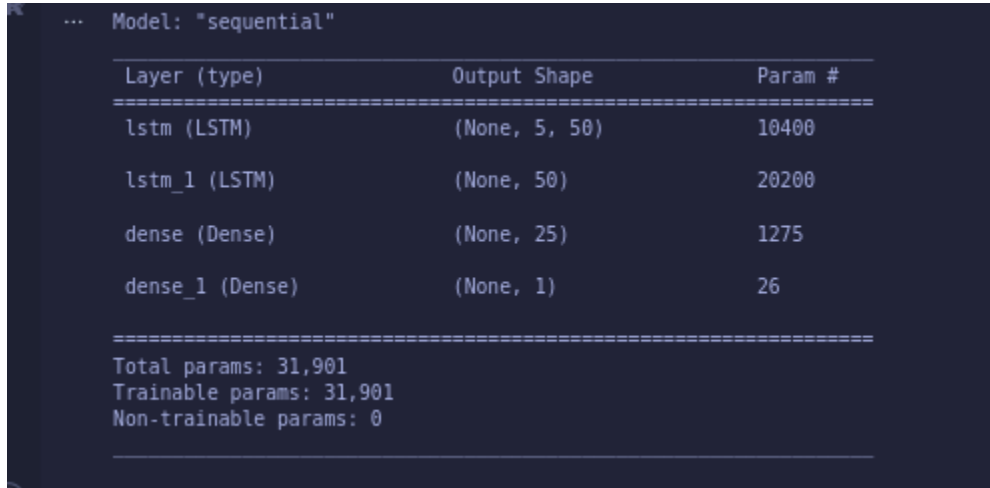
```
print('Residual variance: ', residual_variance)
[85] ✓ 0.0s
... Residual mean: 0.001859406429704669
    Residual variance: 2.6815871310316632
```

*Figure 25: Variance and Mean residual*



## Deep Learning (LSTM)

I designed an LSTM network with 2 LSTM Layers of 50 neurons each, a 1 Dense Layer of 25 neurons and 1 Dense Layer of 1 neuron. The figure below shows a summary of the LSTM network:

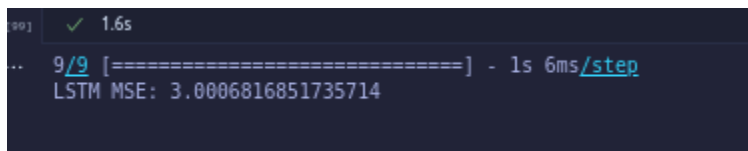


Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 5, 50)	10400
lstm_1 (LSTM)	(None, 50)	20200
dense (Dense)	(None, 25)	1275
dense_1 (Dense)	(None, 1)	26

=====  
 Total params: 31,901  
 Trainable params: 31,901  
 Non-trainable params: 0

Figure 26: LSTM Summary

Evaluation of LSTM model is as shown in the Figure below:



9/9 [=====] - 1s 6ms/step
LSTM MSE: 3.0006816851735714

Figure 27: LSTM Evaluation

## H-test Predictions Visualization:

### SARIMA Future Prediction:

The predicted values for the next 30 time steps or periods. The first predicted value is 23.65598521 and the last predicted value is 24.10967946. The predicted values are likely based on a SARIMA model that was trained on historical friction data. The predicted values could be used to make informed decisions about the future of the sarcoma, such as planning for potential maintenance or repairs.

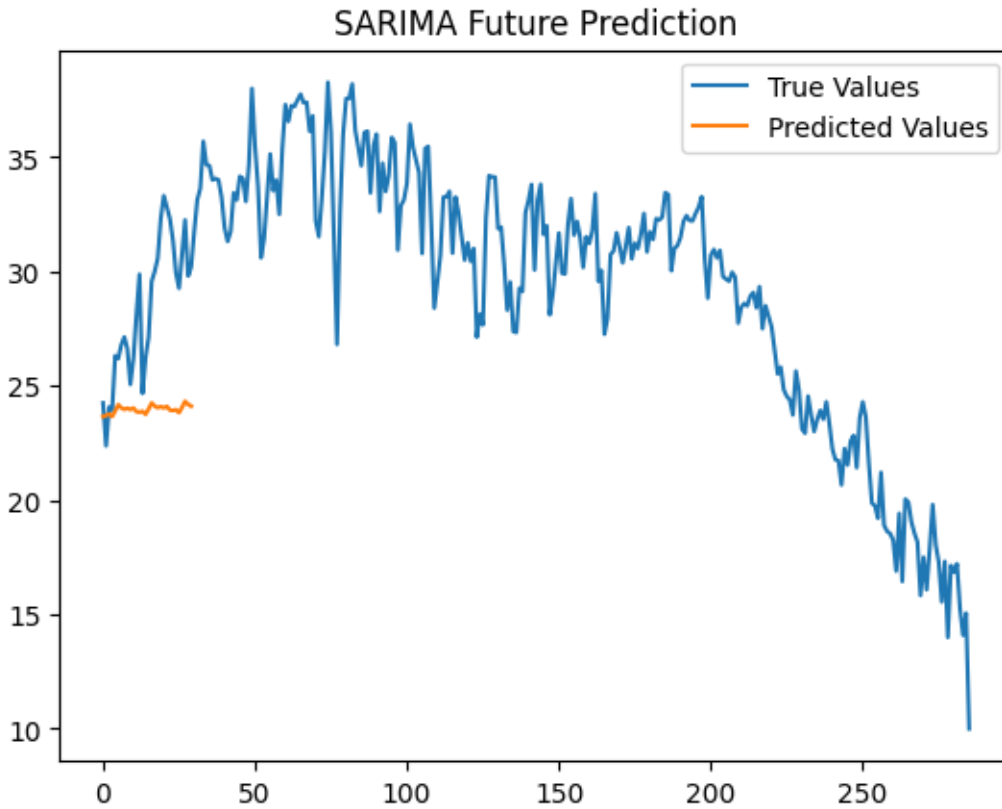


Figure 28: *H*-test Prediction on SARIMA

### ARIMA Future Prediction:

The future prediction of ARIMA model consists of 30 values. The predicted values range from 23.72173325 to 23.87457385. The difference between the predicted values is relatively small, indicating a relatively stable trend.

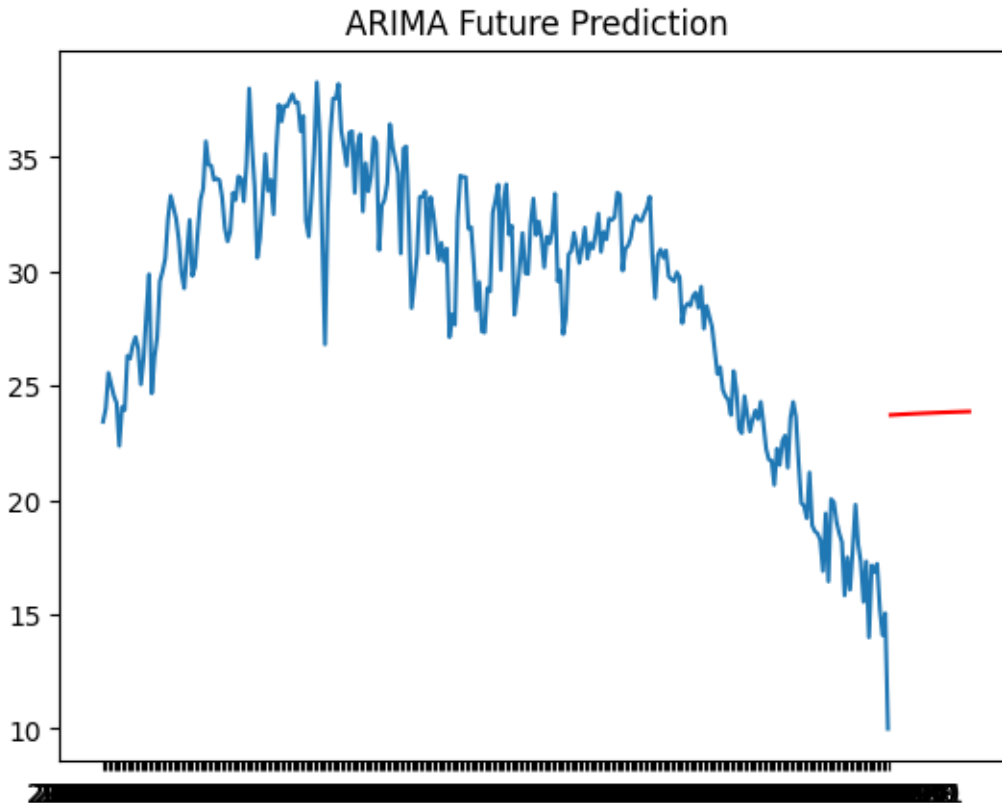
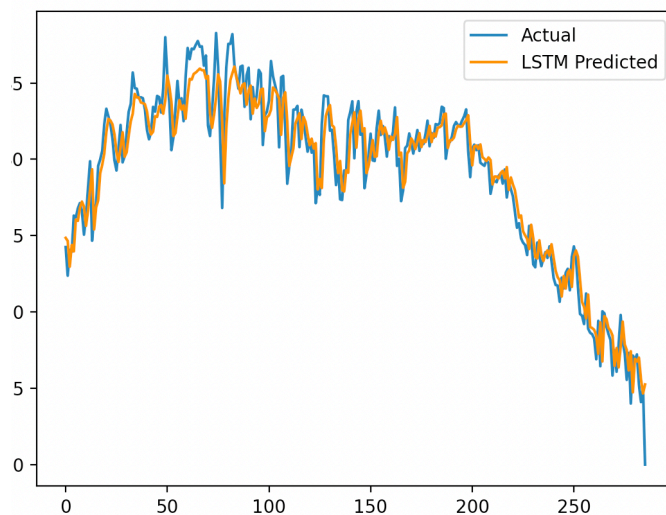


Figure 29: Arima h-test

### LSTM Future Prediction:

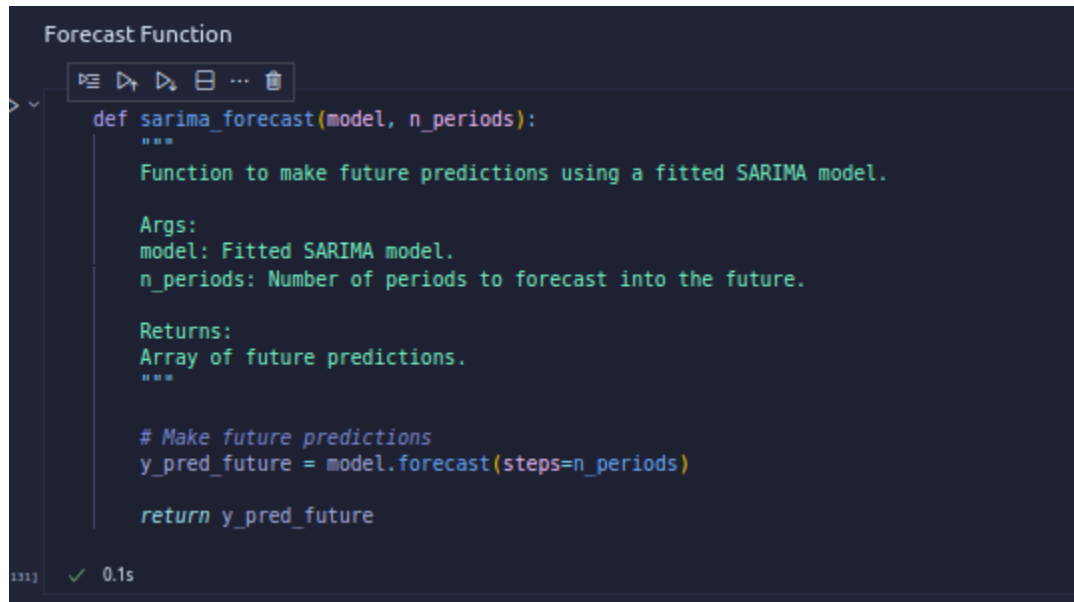
The mean squared error (MSE) of the LSTM model is 3. This indicates that the model's predictions on average differ from the actual values by 3 units. A low MSE indicates that the model's predictions are accurate and have a small amount of error.



*Figure 30: LSTM h-test*

### Forecast Function

The following is a SARIMA forecast function to predict future meantemp.



```

Forecast Function
def sarima_forecast(model, n_periods):
    """
    Function to make future predictions using a fitted SARIMA model.

    Args:
    model: Fitted SARIMA model.
    n_periods: Number of periods to forecast into the future.

    Returns:
    Array of future predictions.
    """

    # Make future predictions
    y_pred_future = model.forecast(steps=n_periods)

    return y_pred_future
  
```

*Figure 31: SARIMA Forecast Function*

After applying the above function, I got the following results with 24 steps

```
[23.65598622 23.69852869 23.76537511 23.65500745 23.90114782 24.1694459
24.03721306 23.96640614 24.01917101 23.95806399 24.02882253 23.8666022
23.84941897 23.88193965 23.75184421 23.98664657 24.24842843 24.11245056
24.0394913 24.09101917 24.02920121 24.09955116 23.937096 23.91977781]
```

### Result:

When we compare all models trained, I would choose LSTM model since it performs better than both SARIMA and ARIMA. This is as evident in the Mean Squared Error where by LSTM has an MSE of 3, ARIMA has MSE of 59, and SARIMA 56. The smaller the error the better the model and thus LSTM is by far the best.

When we compare ARIMA and SARIMA, obviously I will choose SARIMA since it has a lower Mean Squared Error.

**Summary and Conclusion:**

The SARIMA model has limitations in handling complex variances, non-stationary time series data, and outliers. To address these limitations in the future, it may be best to use a deep learning model such as LSTM, which has shown superior performance compared to both SARIMA and ARIMA models in our analysis.