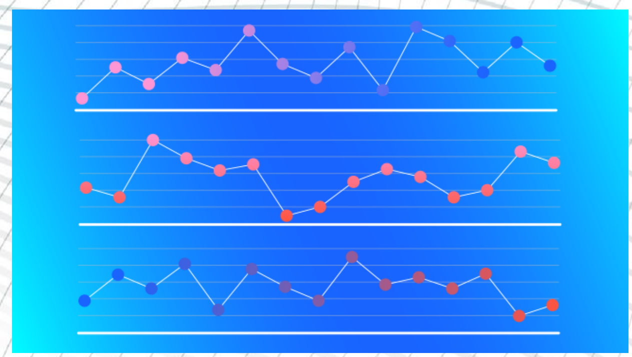
Temperature Forecasting with Machine Learning Algorithms Using Data from IoT Sensor Devices



SCEDT – School of Computing, Engineering and Digital Technologies

Machine Learning **- CIS4035-N**

Module Leader – Dr. Alessandro Di Stefano

Teesside University

TEESSIDE UNIVERSITY
MIDDLESBROUGH
TS1 3BX

Samuel T Ajani

Data Science (M.Sc.)

B1670361

Temperature Forecasting with Machine Learning Algorithms Using Data from IoT Sensor Devices

Samuel Toluwase Ajani
B1670361@live.tees.ac.uk

Department of Data Science
School of Computing, Engineering and Digital Technologies (SCEDT)

ABSTRACT

The advent of IoT devices that are able to inter-network with the environment, humans, and each other has given rise to enormous data being produced. This has led to the need for data scientists to come up with a solution to address the real-time analysis of this data, to give insights and actionable predictions. This paper and the corresponding codes in R look into the analysis of continuous streams of data (Time Series) which reads the temperature of the environment gotten from two IoT sensor devices that are undergoing tests before launching them into the market.

Keywords
IoT, Machine Learning, Time-series, Forecasting.

INTRODUCTION

As signs and warnings are raised because of global warming, the need for some kind of measures to be put in is required. As the earth is warming up fast, the need for scientists to swing to action is required. IoT devices can be developed, reading slight changes in temperature, wind speed, etc., which will trigger a series of actions that can avert devastating occurrences.

ABOUT THE DATA

The dataset worked on, sourced from KAGGLE; with link: https://www.kaggle.com/datasets/atulanandjha/temperature-readings-iot-devices. The data was recorded by the author (Atul, 2018) while testing out an IoT sensor device from 'Limelight IT Research', which was at its alpha testing stage. The author mounted two of these IoT sensor devices in an anonymous room – 'Admin Room', one inside the room and another outside the room. The period of testing was from the 28th of July 2018 to the 8th of December 2018. Between the periods of reading, the devices were shut down or uninstalled at intervals, which gave rise to a few anomalies, inaccurate readings, and outliers in the data recorded. The dataset comprises 5(five) columns (variables) and 97,605(ninety-seven thousand, six hundred and five) rows (observations).

DATA EXPLORATION & FEATURES SELECTION

The dataset in .csv format was downloaded and imported into R. The dataset in its raw state contains columns that have constant values and is not a factor with which the temperature will change. For example, the 'room_id' column does not matter as it was only in a room the experiment was carried out. It was stated by the scientist that carried out the experiment that the devices were shut down at intervals, thereby leading to inconsistency in the pattern. Exploration of the relationship between some relevant variables/column were considered.

The plot below shows the average temperature per day throughout the period of the experiment:

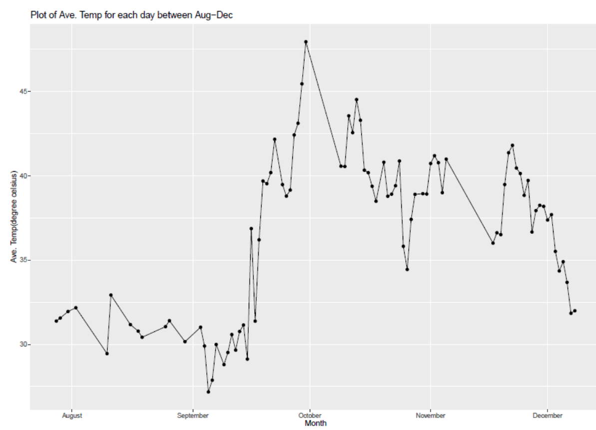


Fig1. Plot of ave. temp for each between Aug –Dec

As the experiment was carried out in India, it is important to know the seasons in which each temperature was taken and the number of specimens there are in each season:

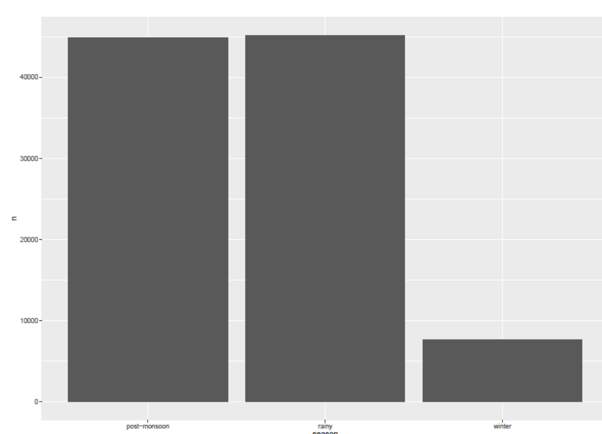


Fig 2. Plot of the number of readings in each season

The box plot shows the distribution of temperature readings based on the location of the IoT sensor devices if it was mounted outside the room or inside.

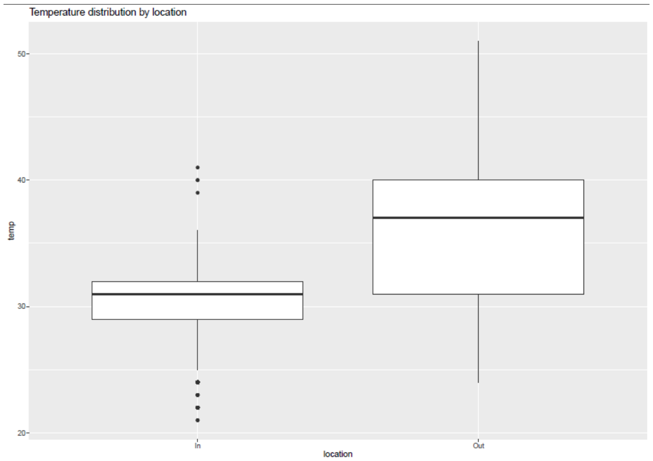


Fig 3. Box-plot showing outliers in temperature readings inside the room

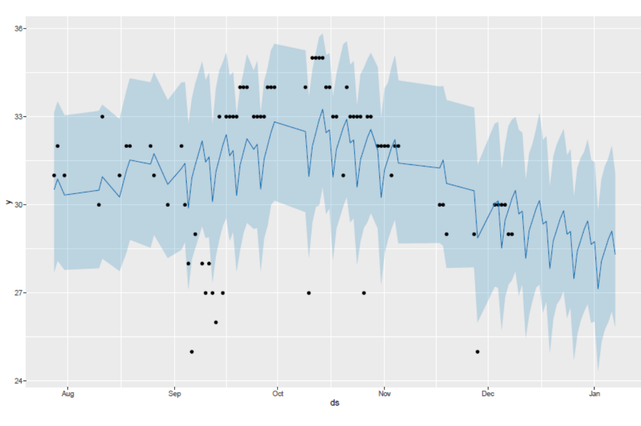
EXPERIMENT

The dataset was cleaned by removing the ‘room_id’ column which was constant throughout the dataset, the ‘noted-date’ and ‘out/in’ column names were changed into ‘time’ and ‘location’ respectively for easier identification. Furthermore, the ‘lubridate’ function was used to make the date column recognizable by the program. When analyzing ‘time-series’ data, the normal and popular machine learning algorithms won’t work, for this experiment, three(3) time-series machine learning algorithms were used:

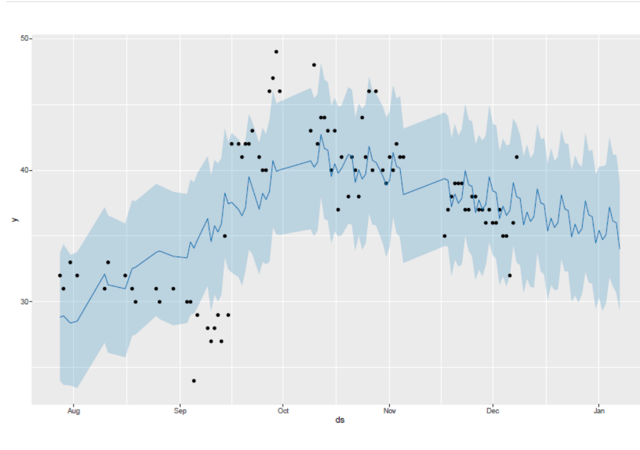
1. Prophet Forecasting Method

The library package ‘Prophet’ was developed by Facebook and has become very popular among data scientists for forecasting using time-series data. The forecast was done in two instances, one for temperature readings taken from Inside and the second from outside.

1.1 The plot shows the ‘Prophet’ plot for temperature readings taken from inside.



1.2 The plot shows the ‘Prophet’ plot for temperature readings taken from outside.



2. Naïve Forecasting Method

Naïve forecasting is a type of time-series forecasting technique that uses the last observation for the next period’s forecast. Therefore, any forecast produced by a Naïve approach has something to do with the last set of observations

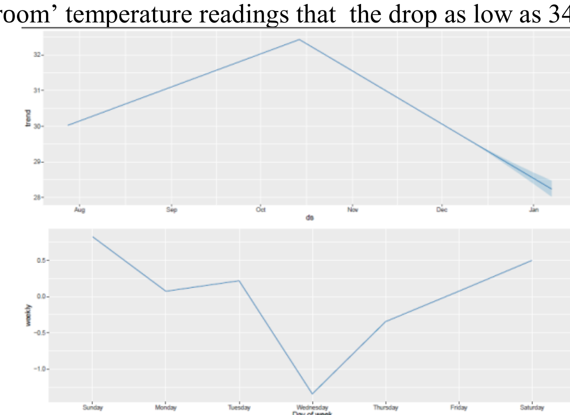
3. Simple Exponential Smoothing

Simple Exponential Smoothing or Single Exponential Smoothing technique is a forecasting technique used to predict the future data of a univariate time series dataset without seasonality or trend.

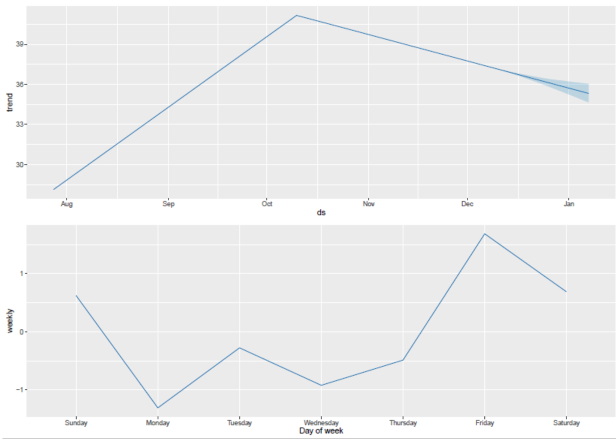
RESULTS

1. Prophet Forecasting Method

The Prophet Forecasting method predicted a downward trend as expected when entering into winter season, thus it predicted for ‘inside-room’ temperature readings that the temperature could drop as low as 27℃, while for ‘outside room’ temperature readings that the drop as low as 34℃.



The above table shows the results of the run of code in R for Temperature readings inside.



The plot above shows the trend for readings of temperature from outside the room.

2. Naïve Method Forecasting

```
Forecast method: Naive method

Model Information:
Call: naive(y = dat_ts, h = 4)

Residual sd: 0.6325

Error measures:
             ME      RMSE MAE         MPE     MAPE MASE ACF1
Training set  0 0.6324555 0.4 -0.02298851 1.356322  NaN    0

Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2019             29 28.18948 29.81052 27.76041 30.23959
Feb 2019             29 27.85375 30.14625 27.24695 30.75305
Mar 2019             29 27.59613 30.40387 26.85297 31.14703
Apr 2019             29 27.37895 30.62105 26.52082 31.47918
```

The summary above shows the summary of the forecast for the reading of temperature from inside.

```
Forecast method: Naive method

Model Information:
Call: naive(y = dat_ts, h = 4)

Residual sd: 1.0954

Error measures:
             ME     RMSE MAE         MPE     MAPE MASE       ACF1
Training set  0 1.095445 0.8 -0.04590305 2.208065  NaN -0.3333333

Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2019             36 34.59613 37.40387 33.85297 38.14703
Feb 2019             36 34.01463 37.98537 32.96364 39.03636
Mar 2019             36 33.56843 38.43157 32.28123 39.71877
Apr 2019             36 33.19226 38.80774 31.70593 40.29407
```

The summary above shows the summary of the forecast for the reading of temperature from outside.

3. Simple Exponential Smoothing

```
Forecast method: Simple exponential smoothing

Model Information:
Simple exponential smoothing

Call:
 ses(y = dat_ts, h = 4)

  Smoothing parameters:
    alpha = 0

  Initial states:
    l = 29

  sigma:  1.7321
Error measures:
                     ME     RMSE MAE       MPE     MAPE MASE        ACF1
Training set -0.3333333 1.732051   1 -1.555556 3.777778  NaN -0.2179487

Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2019             29 26.78029 31.21971 25.60524 32.39476
Feb 2019             29 26.78029 31.21971 25.60524 32.39476
Mar 2019             29 26.78029 31.21971 25.60524 32.39476
Apr 2019             29 26.78029 31.21971 25.60524 32.39476
```

The summary above shows the summary of the forecast for the reading of temperature from inside.

```
Forecast method: Simple exponential smoothing

Model Information:
Simple exponential smoothing

Call:
 ses(y = dat_ts, h = 4)

  Smoothing parameters:
    alpha = 1

  Initial states:
    l = 41

  sigma:  2.9155
Error measures:
                     ME     RMSE      MAE       MPE     MAPE MASE       ACF1
Training set -0.8333333 2.915476 2.166667 -2.506614 6.289683  NaN 0.04922894

Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2019             36 32.26367 39.73633 30.28577 41.71423
Feb 2019             36 30.71603 41.28397 27.91886 44.08114
Mar 2019             36 29.52848 42.47152 26.10267 45.89733
Apr 2019             36 28.52733 43.47267 24.57154 47.42846
```

The summary above shows the summary of the forecast for the reading of temperature from outside.

DISCUSSION & CONCLUSION

According to the results of the machine learning algorithms used in this case study, after checking for the confidence interval/ percentage of the error of the models:

1. Prophet forecasting method: Below is the component details from the tail forecasts:

```
            ds    trend additive_terms additive_terms_lower
99  2019-01-02 28.48105    -1.33992523          -1.33992523
100 2019-01-03 28.43165    -0.34513964          -0.34513964
101 2019-01-04 28.38225     0.07308489           0.07308489
102 2019-01-05 28.33284     0.49773610           0.49773610
103 2019-01-06 28.28344     0.82314510           0.82314510
104 2019-01-07 28.23403     0.07350470           0.07350470
    additive_terms_upper      weekly weekly_lower weekly_upper
99           -1.33992523 -1.33992523  -1.33992523  -1.33992523
100          -0.34513964 -0.34513964  -0.34513964  -0.34513964
101           0.07308489  0.07308489   0.07308489   0.07308489
102           0.49773610  0.49773610   0.49773610   0.49773610
103           0.82314510  0.82314510   0.82314510   0.82314510
104           0.07350470  0.07350470   0.07350470   0.07350470
    multiplicative_terms multiplicative_terms_lower
99                     0                          0
100                    0                          0
101                    0                          0
102                    0                          0
103                    0                          0
104                    0                          0
    multiplicative_terms_upper yhat_lower yhat_upper trend_lower
99                           0   24.44826   29.83759    28.30580
100                          0   25.36598   30.92482    28.24535
101                          0   25.91645   31.22417    28.18495
102                          0   25.97648   31.51109    28.12402
103                          0   26.44958   31.86824    28.05943
104                          0   25.41689   31.00945    27.99785
    trend_upper     yhat
99     28.64163 27.14113
100    28.60387 28.08651
101    28.56569 28.45533
102    28.52645 28.83058
103    28.48296 29.10658
104    28.44099 28.30754
```

From the results, the difference between the trend_upper and lower_trend is not so much and shows that the model is quite precise with the prediction:

2019-01-02:

Trend_upper – Lower_upper

28.63319 – 28.31433 = 0.31886

2019-01-03:

Trend_upper – Lower_upper

28.59425 – 28.25366 = 0.34059

The model's precision is quite high.

```
            ds    trend additive_terms additive_terms_lower
109 2019-01-02 35.64957     -0.9223014           -0.9223014
110 2019-01-03 35.58387     -0.4886217           -0.4886217
111 2019-01-04 35.51817      1.6870192            1.6870192
112 2019-01-05 35.45247      0.6889220            0.6889220
113 2019-01-06 35.38677      0.6228335            0.6228335
114 2019-01-07 35.32107     -1.3107573           -1.3107573
    additive_terms_upper     weekly weekly_lower weekly_upper
109           -0.9223014 -0.9223014   -0.9223014   -0.9223014
110           -0.4886217 -0.4886217   -0.4886217   -0.4886217
111            1.6870192  1.6870192    1.6870192    1.6870192
112            0.6889220  0.6889220    0.6889220    0.6889220
113            0.6228335  0.6228335    0.6228335    0.6228335
114           -1.3107573 -1.3107573   -1.3107573   -1.3107573
    multiplicative_terms multiplicative_terms_lower
109                    0                          0
110                    0                          0
111                    0                          0
112                    0                          0
113                    0                          0
114                    0                          0
    multiplicative_terms_upper yhat_lower yhat_upper trend_lower
109                          0   29.57056   39.81929    35.14495
110                          0   30.08376   39.77135    35.04958
111                          0   31.67042   42.06612    34.93915
112                          0   30.87246   41.30184    34.82829
113                          0   30.85738   41.23398    34.72181
114                          0   28.72087   38.86165    34.62017
    trend_upper     yhat
109    36.13245 34.72727
110    36.09799 35.09525
111    36.07756 37.20519
112    36.04017 36.14139
113    36.01502 36.00960
114    35.99127 34.01031
```

From the results above, the difference between the trend_upper and lower_trend of the outer temperature readings also confirms that the model is quite precise with the prediction but this time around less precise:

2019-01-02:

Trend_upper – Lower_upper

36.13245– 35.14495 = 0.9875

2019-01-03:

Trend_upper – Lower_upper

36.09799 – 35.04958 = 1.04841

In conclusion, the prophet forecasting packaging is quite precise but not much can be said about its accuracy.

2. Naïve Method Forecasting:

```
Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2019             25 22.07761 27.92239 20.53059 29.46941
Feb 2019             25 20.86712 29.13288 18.67931 31.32069
Mar 2019             25 19.93828 30.06172 17.25876 32.74124
Apr 2019             25 19.15523 30.84477 16.06119 33.93881
> #RUN TEST FOR %ERROR
> dat_test$naive = 25  #FROM PREDICTED VALUE IN SUMMARY
>   mape(dat_test$y, dat_test$naive)
[1] 18.91423
```

From the figure shown above, it can be said that the precision of this model is not as sharp as the prophet forecasting method:

Jan 2019 :

At confidence interval 80%

Hi 80 – Lo 80 = 27.92239 - 22.7761 = 5.84478

Feb 2019:

At confidence interval 95%

Hi 95 – Lo 95 = 31.32069 – 18.67931 = 12.64138

From the calculation, we can see that the preciseness of the model is so low, that the difference in temperature as high as 12℃ is not good enough.

From the calculation of the % error, it can be said that the percentage is low(18.91423%) which indicates some level of accuracy.

```
Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2019             37 32.78839 41.21161 30.55890 43.44110
Feb 2019             37 31.04389 42.95611 27.89091 46.10909
Mar 2019             37 29.70528 44.29472 25.84369 48.15631
Apr 2019             37 28.57678 45.42322 24.11780 49.88220
> #RUN TEST FOR %ERROR
> dat_test$naive = 37 #FROM PREDICTED VALUE IN SUMMARY
>   mape(dat_test$y, dat_test$naive)
[1] 12.88045
> |
```

From the above results for temperature measured outside the room, the predicted temperature for the proceeding months was given to be 37℃, with the same problem faced as in the results from temperature readings taken inside the room, precision is low and the accuracy is quite high:

Jan 2019:

At confidence interval 80%

Hi 80 – Lo 80 = 41.21161 – 32.78839 = 8.42322

Feb 2019:

At confidence interval 95%

Hi 95 – Lo 95 = 46.10909 – 27.89091 = 18.21818

% error = 12.88045%

3. Simple Exponential Smoothing:

```
Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2019             29 28.26010 29.73990 27.86841 30.13159
Feb 2019             29 27.95362 30.04638 27.39970 30.60030
Mar 2019             29 27.71845 30.28155 27.04004 30.95996
Apr 2019             29 27.52019 30.47981 26.73683 31.26317
> #%ERROR
> dat_test$simplexp = 29
>   mape(dat_test$y, dat_test$simplexp)
[1] 9.497833
> |
```

From the above figure shown above, the Simple/Single Exponential Smoothing demonstrates a low % error of 9.497833% with somewhat higher precision than the Naïve method:

Jan 2019:

At confidence interval 80%

Hi 80 – Lo 80 = 29.73990 – 28.26010 = 1.4798

Feb 2019:

At confidence interval 95%

Hi 95 – Lo 95 = 30.13159 – 27.86841 = 2.26318

```
Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2019             36 32.11991 39.88009 30.06591 41.93409
Feb 2019             36 30.51272 41.48728 27.60794 44.39206
Mar 2019             36 29.27949 42.72051 25.72186 46.27814
Apr 2019             36 28.23982 43.76018 24.13183 47.86817
> #%ERROR
> dat_test$simplexp = 36
>   mape(dat_test$y, dat_test$simplexp)
[1] 12.39331
> |
```

Jan 2019:

At confidence interval 80%

Hi 80 – Lo 80 = 39.88009 – 32.11991 = 7.76018

Feb 2019:

At confidence interval 95%

Hi 95 – Lo 95 = 44.39206 – 27.60794 = 16.78412

From the temperature readings from the IoT sensor device mounted outside, the results, show that even if the performance of the Simple Exponential smoothing is high for the IoT device mounted inside the room, the performance for the one mounted is quite poor, therefore we can assume:

That there are other factors to be considered when taking the temperature taken outside into account, e.g. non-environmental heating around the device, human interaction, etc.

RECOMMENDATIONS

From the results and inferences from the experiment:

1. The limitation to temperature readings as data used were gotten from August through to December, I recommend that the models would perform better if data can be provided for at least a whole year (January – December).

2. According to the author, the IoT sensor devices were turned off at intervals, I would recommend that the IoT be left alone to keep collecting data with fewer interruptions and if the whole experiment could be isolated from fewer external or human interference.

REFERENCES:

Avril, Coghlan (2018). *"A Little Book of R for Time Series"*.

Atul, Anand (2020). *"Temperature readings: IoT Devices"*. URL: https://www.kaggle.com/datasets/atulanandjha/temperature-readings-iot-devices

Kohei, Mu (2020). *"IoT Temperature Forecasting"*. URL: https://www.kaggle.com/code/koheimuramatsu/iot-temperature-forecasting

https://www.msystechnologies.com/blog/fundamentals-of-forecasting-and-linear-regression-in-r/

https://www.simplilearn.com/tutorials/data-science-tutorial/time-series-forecasting-in-r#:~:text=Time%20series%20forecasting%20is%20the,as%20a%20time%20series%20data.

https://www.pluralsight.com/guides/time-series-forecasting-using-r