Temperature is a very important aspect in our lives that tends to slip under the radar. It can have a huge impact on not only our daily lives but also on all sorts of activity that we don’t even pay attention to. So important is its effect that some may even go as far as to posit that it decides how developed a country is. Ever noticed how most developed countries are usually the ones with cooler weather? Changes in the temperature can change the state of matter as well. At certain temperatures, a solid can turn into liquid or gaseous state and vice-versa. This fact has huge implications in certain industries. For example, a company dealing with storage of hazardous substances may want to keep an eye on the ambient atmospheric temperature as fluctuations in temperature might cause the harmful substance to change state and leak out of the storage facility. Such a substance may be something like petroleum which turns into fumes faster at higher temperature and forms an invisible cloud which can ignite easily with a small spark from cigarette or static electricity damaging private or commercial properties nearby.

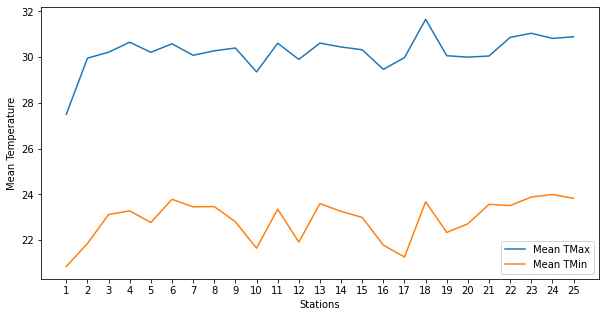
Another area where temperature plays a significant role is agriculture. Extreme temperatures can affect agricultural production and can be the key difference between good or bad yield. For developing countries like India where more the two-third of the population depends on agriculture for their livelihood, an unexpected fluctuation in temperature can be catastrophic. Farmer suicides in India has always been a major issue and a topic of huge debate. This goes to show how important agriculture is for India and hence, the role temperature plays in its economy.

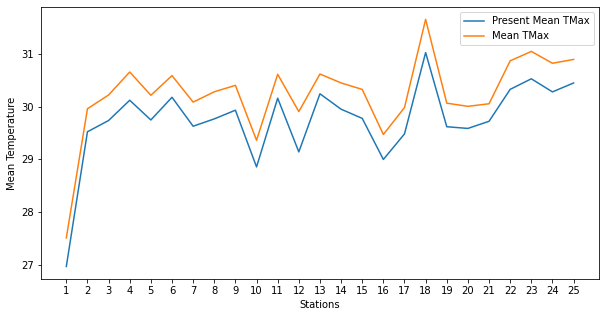
But temperature plays far more role than those already mentioned above. Our very existence depends temperature. As we all are aware, the reason earth can support life is due to the fact that it is precisely at the right distance from the sun. This ensures that the earth’s temperature is just right to support life. The entire ecosystem and all aspects of nature function as an efficient system where temperature plays a significant role. It is responsible for weather phenomenon, the hydrological cycle, etc. Temperature changes can tip this balance. This is why global temperature rise has been a cause of concern in the entire world as if the temperature rises high enough, it may be the cause of the end of mankind.

We can go on and on forever discussing the importance of temperature. To put it briefly, it’s an important aspect of our lives whether we realise it or not. Therefore, being able to predict temperature can be helpful in many ways. Firstly, it may allow for having appropriate measures in place so that a change in temperature does not lead to unpleasant outcomes. Knowing the likely temperature in future can also be a useful tool to monitor weather conditions and issue warnings by the meteorological department for likely weather phenomenon like tornadoes, heavy rainfall, etc which would allow the general public to take necessary precautions. As data scientists, this task of predicting temperature falls on us and we would like to achieve this with the help of machine learning and artificial intelligence. The dataset we have for this example is published by the Korea Meteorological Administration situated at Seoul, South Korea. The dataset is comprising of LDAPS model’s next-day forecasted data, in-setu maximum and minimum temperatures of the current day as well as geographic auxiliary variables like latitude, longitude, station (where the temperature was recorded), etc. The uniqueness of this project lies in the fact that we have to target (also called dependant feature) columns as opposed to one target column which we, as budding data scientists, are used to. While this may look intimidating at first, it is entirely possible with little to no changes in how we do the data pre-processing. We can approach this problem in two ways. First way is to fit two models for each target column. The second approach is to use algorithms that support multiple target variables. This task is a regression task as our target variable is continuous and numeric. For this analysis, we will go with the latter and the algorithms we will try are Linear Regression, Random Forest Regressor, Decision Tree Regressor, K-Neighbors Regressor and MLP regressor.

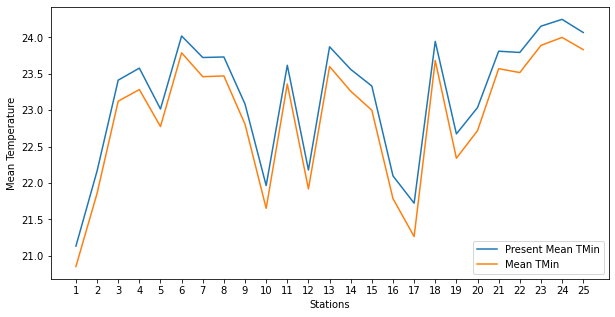
We start by importing the dataset and checking for missing values. There are about 164 rows with missing values out of a total of 7752 rows. This translates to about 2.16% of the data. 2.16% is a low enough data loss, we will drop all rows with missing values. Now that we have taken care of the missing values, lets that a moment to understand the data.

The dataset has 25 columns. Out of these, the target columns for the analysis are Next\_TMax and Next\_TMin, which represents the maximum and minimum temperature for the next time period respectively. For the remainder of this article, we will refer to Next\_TMax and Next\_TMin as simply TMax and TMin. The rest of the columns are feature columns represent maximum and minimum temperatures of present-day, and various geographic auxiliary variables. A date column is also present. While the presence of a data column allows for treating this problem as a multi-variate time series analysis, this would make the prediction problem more complicated as to perform multi-variate time series analysis we need to venture into the realm of deep learning and use LSTM to make predictions. To keep things simple, we will just drop the Date column and treat the analysis as a regression problem.

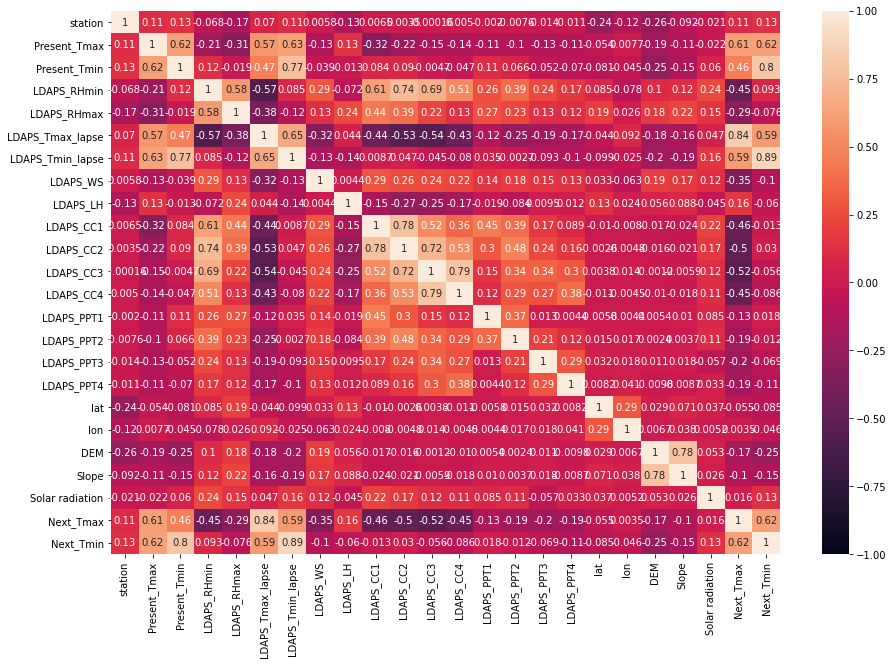
The table below shows the average TMax and TMin for each station. It is evident that the two almost mimic each other. For example, at station 18, we observe a peak in both the TMax and TMin.



Similarly, if we observe the average of the Present\_TMax and TMax, the lines are completely similar. Same is the case for average Present\_TMin and TMin, as seen below.



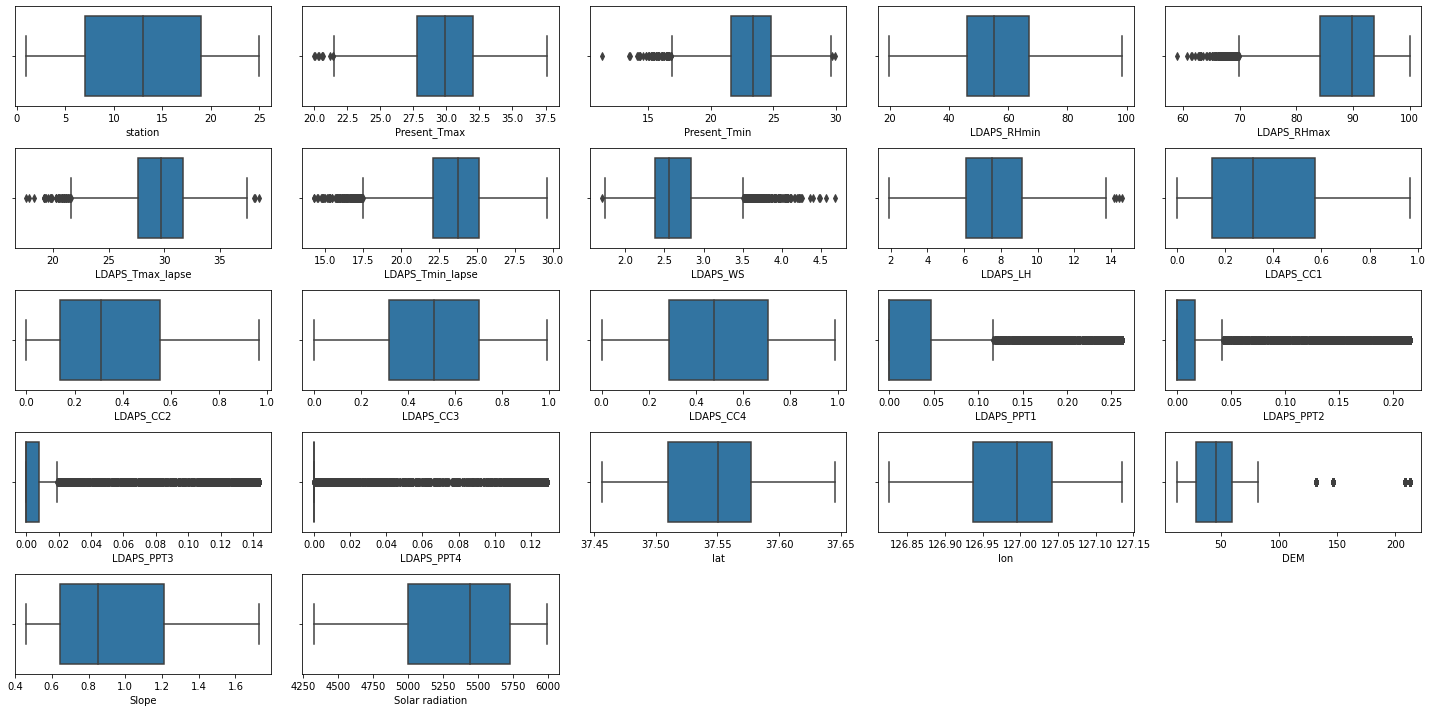
This forces us to suspect that there might be high correlation between these variables. Inspecting the correlation matrix, confirms our suspicion as seen below.



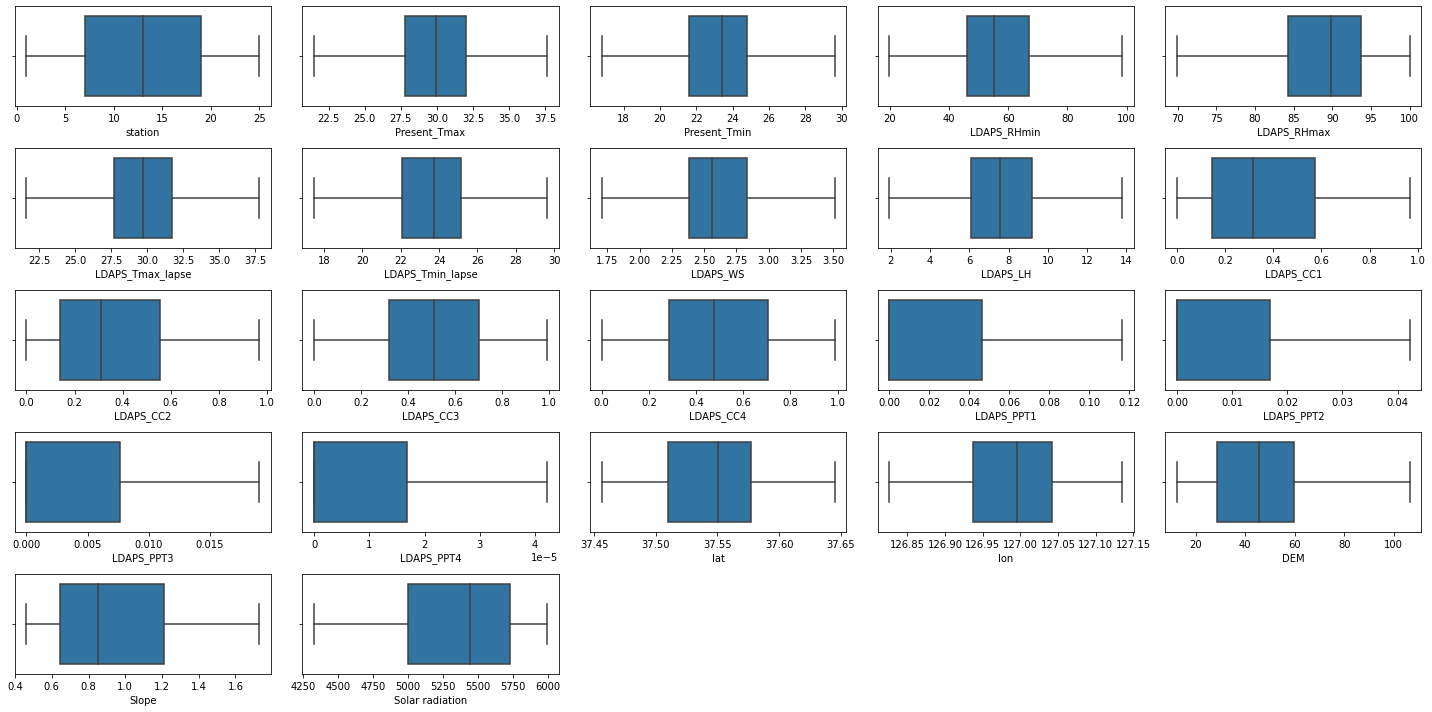
We see that for the TMax (column name: Next\_TMax), Present\_TMax is highly correlated. Same is the case for TMin (column name: Next\_TMin). TMin and Present\_TMin are highly correlated. Lastly, TMax and TMin are also highly correlated with each other. This seems logical as a higher max temperature for a day will also raise the minimum temperature for that day and vice-versa. However, we need to confirm if these high correlations are statistically significant or not. We do this by using Pearson Correlation Test. The test reveals that apart from the column ‘solar radiation’, the rest of the columns have statistically significant relation with the target columns.

Now that we understand our data better, it’s time to move onto the data pre-processing stage. We begin that by first checking for skewness. Any column with a skewness of 0.5 must be transformed to reduce or eliminate skewness. By this criterion, the columns 'LDAPS\_WS', 'LDAPS\_LH', 'LDAPS\_CC3', 'LDAPS\_CC4', 'Slope', 'LDAPS\_PPT1', 'LDAPS\_PPT2', 'LDAPS\_PPT3', and 'LDAPS\_PPT4' have skewness. In order to remove or bring the skewness to acceptable limits, we followed a trial-and-error approach where various transformation techniques were used in each column and then the skewness was checked again. Finally, the transformation which produced the best result was selected. Therefore, for 'LDAPS\_WS', 'LDAPS\_LH', 'LDAPS\_CC3', 'LDAPS\_CC4', we perform square root transformation, for the column 'Slope' we performed cube root transformation and lastly for the columns 'LDAPS\_PPT1', 'LDAPS\_PPT2', 'LDAPS\_PPT3', and 'LDAPS\_PPT4', Yeo-Johnson transformation was performed.

Next, we check for outliers. From the boxplot below, it is evident many of the columns have outliers.



We treat these outliers by replacing them with values that are 1.5 times the interquartile range on either side of the 25th and 75th Percentiles. After this the resulting boxplots are much better as evident below.



After this, we separate the target and feature columns. We should keep in mind that we have two target columns for this analysis. Therefore, while separating the feature columns and the target columns, the ‘y’ variable will contain the two target columns, namely, Next\_TMax and Next\_TMin and the independent features will be contained in ‘x’. We, then, move onto OneHotEncoding any categorical variables. Note that although the column ‘station’ holds numeric values, it is actually categorical variable as each number represents a weather station. Therefore, we will dummy encode this column. Once this is done, we move on to standardizing the numeric columns. This concludes our pre-processing pipeline and we can finally start fitting models to this data.

Before fitting the model, we fit an arbitrarily chosen model multiple times with different random states during train-test split to find the best random state where accuracy is the highest. In our case, that random state was 199. We then use this random state to build all our models after train-test split.

As mentioned in the beginning of the article, we will fit five models. Then judge the R2-scores of each model after which we will perform cross validation on each of the five models. Once this is done, the model with the least difference between R2-score and cross validation R2-score will be selected for final stage. Once the models were fitted, the R2-scores for Linear Regression, Random Forest Regressor, Decision Tree Regressor, K-Neighbors Regressor and MLP regressor were 1.0, 0.9988302, 0.995929, 0.812036 and 0.9969 respectively. While these scores are already high and performing cross validation might seem unnecessary, the high scores may be a result of overfitting. With cross validation, we can ensure best fit and eliminate chances of overfitting or underfitting. After cross validation, the scores in the same order as before were 1.0, 0.99, 0.99, 0.79 and 0.99. Out of all the five models, Linear Regression had the least score difference of 0 and hence was chosen as the final model.

This leads us to the final stage of our project which is hyper-parameter tuning. Although, for our final model (Linear Regression), the R2 score is perfect, we have to remind ourselves that this impressive score is one existing data only and might not hold if presented with new unseen data. Therefore, to ensure that the model is as robust as possible, it is a good practise to perform hyper parameter tuning. We will do so using GridSearchCV which, after fitting the data, gives the following hyper parameter settings: {'fit\_intercept': True, 'normalize': True, 'positive': True}. Fitting a model with these parameters concludes our model building process. Finally, we serialise our model for production and future use. This allows us to make predictions without fitting the model in future.

While it has been an interesting experience to work on this project, there is still room for improvement. There were two columns containing the geographical coordinates. This information may have been used to find the location and could have been added into the dataframe which may have contributed to a even better and robust model. This was not done in this analysis as the geo-encoder library, at the time to making this model, was not functioning properly. Also, as mentioned before, we had dropped the ‘Date’ column and treated this project as a simple regression problem instead of a time series analysis. Performing a time series on this data set might have resulted in an even better model and needs to be explored as usually time series data have factors like autocorrelation (that is, when a value is correlated with itself from a previous period of time), trend and seasonality which can be accounted for using time series analysis like Seasonal Autoregressive Integrated Moving Average or SARIMA. Furthermore, multivariate time series can also be performed using deep learning techniques like neural networks (Long Short Term Memory – LSTM which is a special type of Recurrent Neural Network). Lastly, data science is a continuous process which involves constantly monitoring model performance and re-tweaking the algorithms to correct performance issues which maybe be encountered once the model is deployed. Thus, one should explore ways to how the model can be trained at regular intervals with new data as they become available. Furthermore, one should aim to achieve this with minimal human intervention and make this process as automatic as possible.