Week 6, 11-10-2019

Intelligent Buildings

7LY5M0_Yiwen Shen_Assignment 1

Problem 1: General questions

(1) What is an intelligent building and how does it differ from a building automation system? In your opinion what developments are needed to speed the future implementation of smart buildings?

A building with intelligent automation systems are not only able to consume energy and electricity effectively but also utilise sensors or actuators to collect data from the internal and external sources. (Internal including user data and energy consumption; external sources including weather forecast etc.) The collected data can be analysed by computer systems, optimised by the machine learning algorithms and thus providing insights to satisfy the purpose of improving the building performance, bring sustainabilities into people's lives.

In the recent development of Intelligent building automation systems, designers and engineers have gained interests in **User Experience** aspect (Wen, J. T. (2018). From an Industrial designer perspective, user experience design lies in the intersection of system design and aesthetics interaction (Ross, P. R., & Wensveen, S. A. 2010.). Aesthetic qualities exist not only in the form of building design but also in the interactions between occupants and building. Good interaction design in the context of commercial buildings will offer occupants good experience, maintain their good physical condition, they will also more likely to increase their level of user commitment to the building and thus improve their productivity (Verhaart, J., Li, R., & Zeiler, W. 2018).

For example, user involvement in the design of personal warming/ cooling system is essential (Verhaart, J., Li, R., & Zeiler, W. 2018). Occupants would like to require more thermal comfort in a building, however, thermal comfort is a very difficult aspect to address since it can be defined individually different based on the gender, age, the physical condition of different occupants, as well as humidity and air quality in the building. In order to create an environment with thermal comfort that satisfies the majority of occupants, designers need to study users as well as their behaviours and perceptions.

On the other hand, building managers would like to get more easy access to the data sources, where they can also observe and monitor building performance (Wen, J. T. (2018). For instance, in the control room, where building receives data from different distributed sensors, the obtained data should always be visualized clearly, allowing building managers to control building without extensive training. In my opinion, UI (user interface) design from the monitoring system should be designed simple yet functional to facilitate building manager to perform tasks.

(2) Explain the differences between extrapolation and interpolation. What are their pitfalls for Predictive modelling?

Extrapolation and interpolation are both methods used to predict or estimate hypothetical values for a variable. However, there are some differences between them:

- **Extrapolation** is a method that estimates a value that is beyond the original observation range (Extrapolation. 2019). Which means the value of a variable stands in between known observations. However, by using the extrapolation method, it usually has a higher risk to produce meaningless results that can not deliver higher accuracy in a predictive model.

- **Interpolation** is a method that estimates a value that stays within the range of a set of known data points (Interpolation. 2019). The value of a variable stands between known data points. However, the data points sometimes have likely very similar position or characteristics, For example, in a small amount of datasheet, when using 'ffill' method to perform interpolation function in Jupyter, the new predicted data will use the same value from the previous data point, the result would be very close to known data points, This can also affect the accuracy in a predictive model.

(3) Why is it important to check if a data is stationary or not? Which methods could be used to check the stationarity of a data set?
Stationary is a vital characteristic of time series data. Therefore, stationarity is an important aspect that needs to be addressed when performing data analysis or machine learning (Duke's Fuqua,2019). The goal of machine learning is to generate predictions beyond the training set of the model, stationaried data set is easy to perform prediction. For instance, a stationaried data set allows data analysts to simply predict its statistical properties will be the same in the future as algorithm models have been performed similarly in the past.

We can identify obvious stationary and non-stationary time series by observing line plot. Trends and seasonality (Duke's Fuqua,2019) can be found on stationary data set. There are also many methods can be used to check stationarity in the data set. Summary statistics and Statistics tests are two useful methods.

(4) Project Haystack is an initiative to reduce efforts in data preparation. What other initiatives can you think of that will reduce the workload in the data analytics process?
Altair knowledge hub (Data Governance Tools,2019) also utilizes machine learning algorithms that recommend data analysts relevant data sets and preparation steps to reduce unnecessary workload. Last but not least, Altair reduces thresholds for newcomers when learning about data preparation by designing an intuitive and collaborative user interface, it facilities to bridge the gaps between different stakeholders, designers, researchers and data analysts is able to collaborate same data set remotely at the same time.

Alteryx Designer (Alteryx Designer,2019) is another data preparation cloud platform that aims to reduce the workload in data preparation, integration and analysis. Alteryx designers features 'drag + drop' visual programming in an intuitive user interface to prep, blend and analyse data. User can also perform advanced analytics such as predictive, statistical and spatial analytics in the same workflow.

We can conclude that the trend of performing data analysis seems to be more code-free or code-friendly by utilizing an intuitive user interface design on a cloud platform. As data analytics becomes a popular trend, the threshold for data analytics is getting lower and lower, and data analytics platforms that provide a good user experience take data analytics to a new level.

Problem 2: General questions

(1) The *pd.to_datetime* is used to convert UnixDateTime to DateTimeIndex, to make it clear for data analysis, the UnixDateTime is removed and a new index column named "TimeIndex" is added (see figure 1).

```
In [30]: # converting Unix date to DateTimeIndex
df['TimeIndex'] = pd.to_datetime(df['UnixDateTime'],unit = 's')
df.drop('UnixDateTime', axis =1, inplace=True)
df.set_index('TimeIndex')
```

Out[30]:

TimeIndex	Aggregate [W]	Fridge [W]	Freezer [W]	Washer Dryer [W]	Washing Machine [W]	Toaster [W]	Computer [W]	Television Site [W]	Microwave [W]	Kettle [W]
2013-11-01 22:13:18	358.0	0	0	0	0	0	17	138	2	0
2013-11-01 22:13:31	357.0	0	0	0	0	0	17	138	2	0
2013-11-01 22:13:46	358.0	0	0	0	0	0	17	138	2	0
2013-11-01 22:13:59	357.0	0	0	0	0	0	16	138	2	0
2013-11-01 22:14:14	NaN	0	0	0	0	0	17	139	2	0
2013-11-01 22:14:16	358.0	0	0	0	0	0	17	139	2	0

Figure 1. Data Frame

- *pd.isnull(df)* is used to identify the location of the NaN value.
- *pd.isnull(df).sum()* is used to calculate missing values. In the given datasheet, it seems there are only missing values in the Aggregate, to calculate the ratio of missing data, the following operation is performed:

```
ratio_missing_data = pd.isnull(df['Aggregate [W]']).sum()/len(df['Aggregate [W]'])
```

The ratio of missing data is **0.19999997061285135**

(2) To choose which method will be suitable to fill in the missing values. First of all, the missing values and aggregate values are being plotted. The blue dots indicate the missing values (see figure 2).

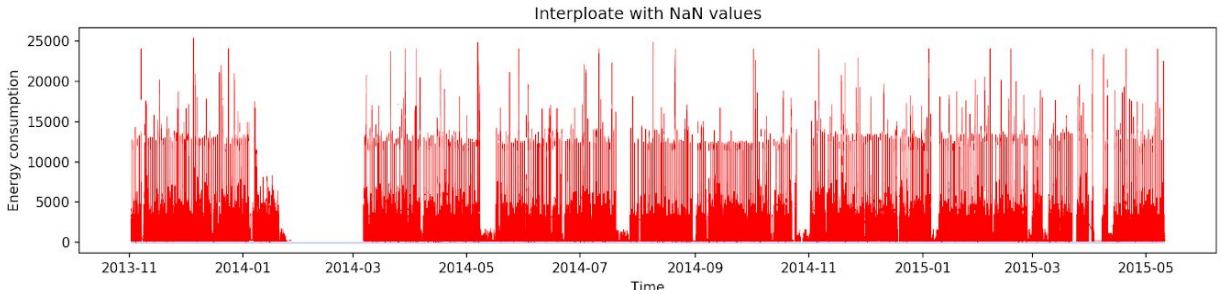


Figure 2. Interpolate NaN values

Three interpolate functions are used to fill in the missing values.

- **Method 1: df1 = df.fillna(0)**, this method will replace all NaN elements with 0s. As we can see from the following figure,the blue dot is covered by red line (see figure 3). However, by using this method will make the mean values look horrible,it will also make the prediction less precious.

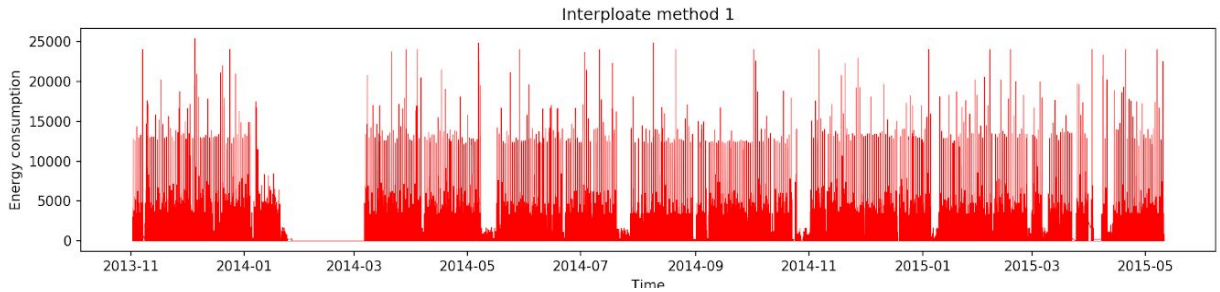


Figure 3. Method 1

- **Method 2: df2 = df.fillna(method = 'ffill')**, this method will replace all NaN elements forward or backward (see figure 4).

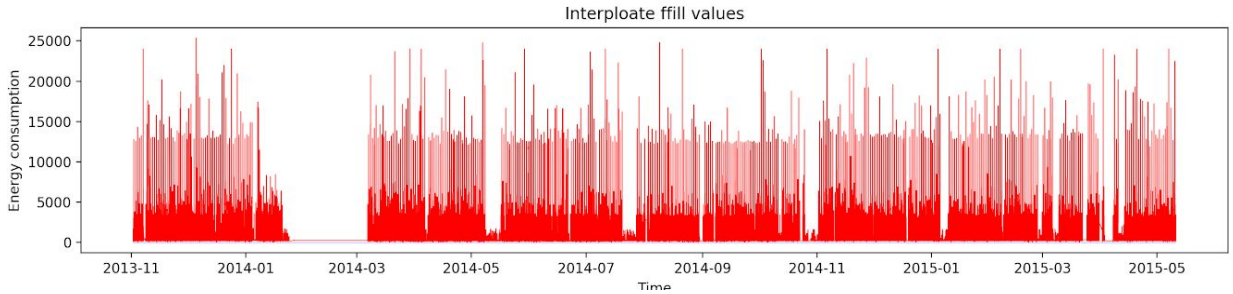


Figure 4. Method 2

- **Method 3: df3 = df.fillna(df['Aggregate'].mean())**, this method will replace all NaN elements with mean value of aggregate. Comparing the missing values and replaced values that fill with mean value of aggregate seems deliver higher accuracy for prediction, therefore method 3 is being used (see figure 5).

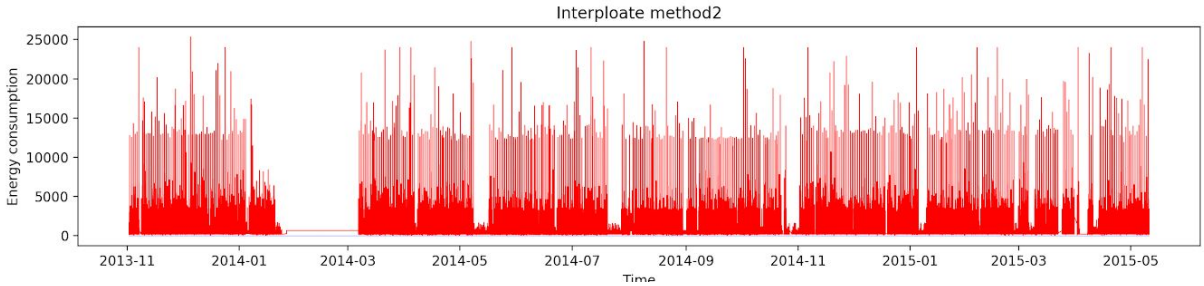


Figure 5. Method 3

(3) Outlier detection

- To find outliers, the ***EllipticEnvelope*** library was used in this data set. The dataframe 3 (missing values were filled with mean value of aggregate) was used. The parameter of outliner fraction is 0.001. The following algorithm was applied to find the amount of outliers:

```
n_outliers = int(outliers_fraction * n_samples)
n_inliers = n_samples - n_outliers
```

- Resample the dataframe into an array of tuples and the timedate index object to ints before training the model .
- Execute algorithm to identify the outliers, and then check the number of outliers from the prediction match to the amount of outliers, there is one outliner missing, but the result is very close to what we predict.

```
# Check if number of outliers found is corr
y_pred[y_pred == -1].sum()
```

```
-6806
```

```
n_outliers
```

```
6805
```

Figure 6. Outliers check

The plot shows the amount of outliers (see figure 7).

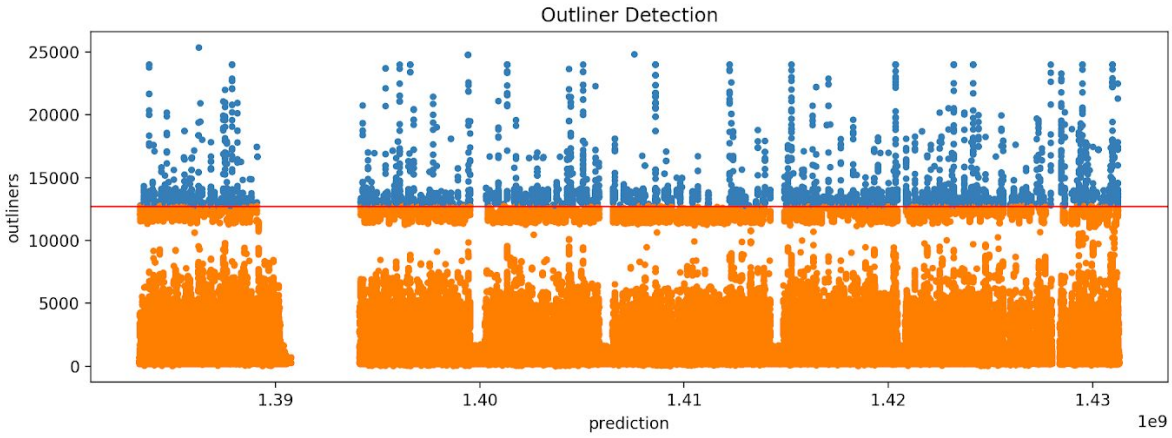


Figure 7. Outliers

(4) Warning function

The warning function is executed by defining a function named 'warning'. Inside of the warning function, a if-statement is used when the consumption of power reaches 200kw. The exact date and time will be printed and stored in a new Excel file named 'new dataframe' (see appendix x). In the exercise only first 10 warnings were printed to demonstrate the warning function was executed correctly. By observing the sum of appliances, we could conclude that the cause of the power surge was due to **none** of the appliance.

(5) After resampled the aggregate power as daily, weekly and monthly by using sum and mean resampling function,we can see a significant difference by comparing both plots.

- In the resample sum value chart, the sum values of daily and weekly aggregate have been relatively stable, although there are significant drops in 2014-01 and 2014 -11. In contrast, the sum of monthly aggregate suffered a severe decline. Particularly from 2014-01 to 2014-03, It is hard to find relationships between these sum values.

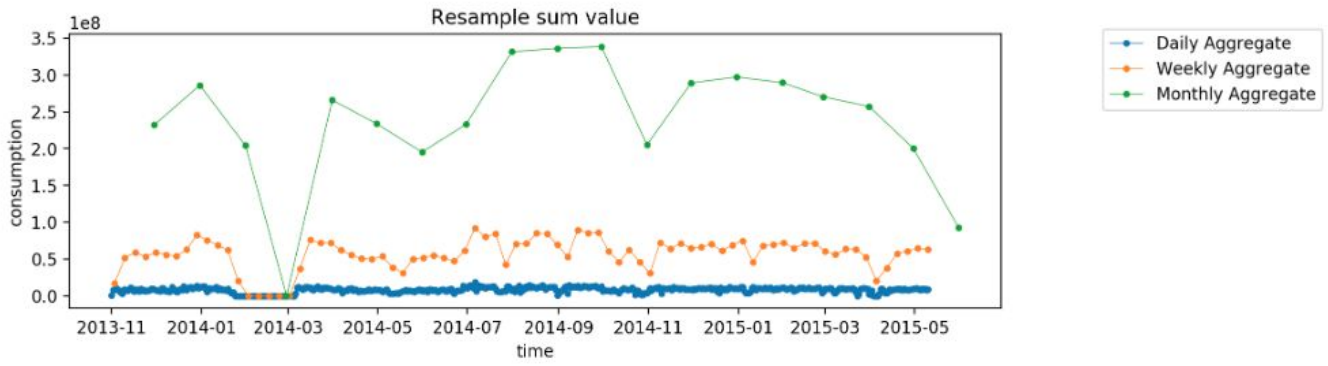


Figure 8. Resample sum value

- In the resample mean value chart, the mean values of weekly aggregate and monthly aggregate have been relatively stable, they have similar amplitude and fluctuation. In contrast, The mean values of daily have been more dynamic.

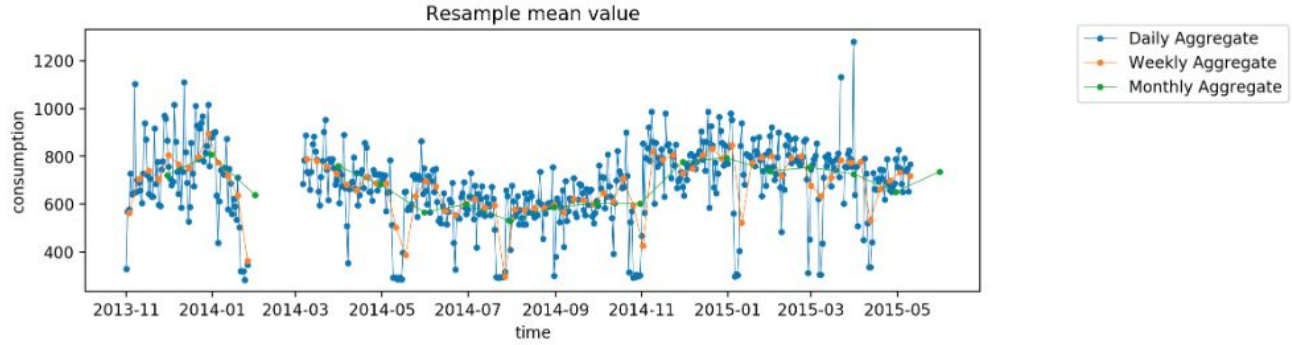


Figure 9. Resample mean values

Problem 3: Extracting trends from observations

(1) Heatmap

To create a heatmap shows the fan speed with hours and days on respectively on Y-axis and X-axis, the ***seaborn library*** was utilized. By using pandas date-time index function, the data sets were able to be split into different periods.

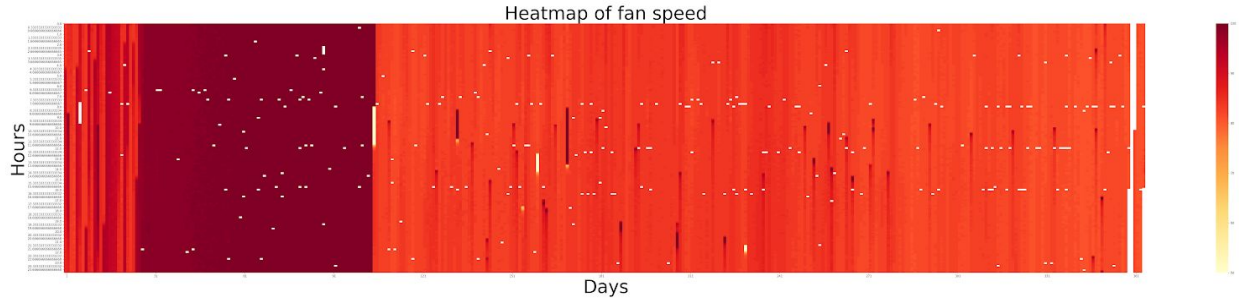


Figure 10. Heatmap

(2-1) Irregular control patterns:

- The white dots and spaces indicate missing data (**NaN** values), means data on that moment of the day were not recorded. For instance, on day 361, data were not recorded in the morning but indeed recorded at night (figure 11 & 12).

```
In [305]: df.loc[df['day_exact']== 361]
```

Out[305]:

System Time	Fan control	Supply Air Temp	Ambient Air Temp	year	month	day	hour	hour_exact	day_exact
2015-12-27 00:00:00	NaN	NaN	NaN	2015	12	27	0	0.000000	361
2015-12-27 00:05:00	NaN	NaN	NaN	2015	12	27	0	0.083333	361
2015-12-27 00:10:00	NaN	NaN	NaN	2015	12	27	0	0.166667	361
2015-12-27 00:15:00	NaN	NaN	NaN	2015	12	27	0	0.250000	361
...	...	...	...	...	...	...	...	...	...
2015-12-27 21:30:00	82.0	20.2000	11.9875	2015	12	27	21	21.500000	361
2015-12-27 21:35:00	82.0	20.2000	12.2000	2015	12	27	21	21.583333	361
2015-12-27 21:40:00	82.0	20.2000	12.4500	2015	12	27	21	21.666667	361

Figure 11 & 12. NaN values

- In the first month day 1 to 25), the performance of the fan does not look normal, it is difficult to make predictions from the information given by the heat map.

(2-2)Operation Schedule:

- From 26th of February to 15th of April (day 26 to 105) the speed of fan reached highest peak (around 100). In the rest of the year (from 16th of April to 31st of December) the speed of fan maintains around 80.
- The fan increases its speed most likely between 8:00 am-9:00 am, 1:00 pm -2:00 pm and 8:00pm - 9:00 pm.

(3) Scatterplot:

To create a scatter plot, the ***matplotlib*** was used, supply air temperature on Y-axis and ambient air temperature is on X-axis (see figure 13).

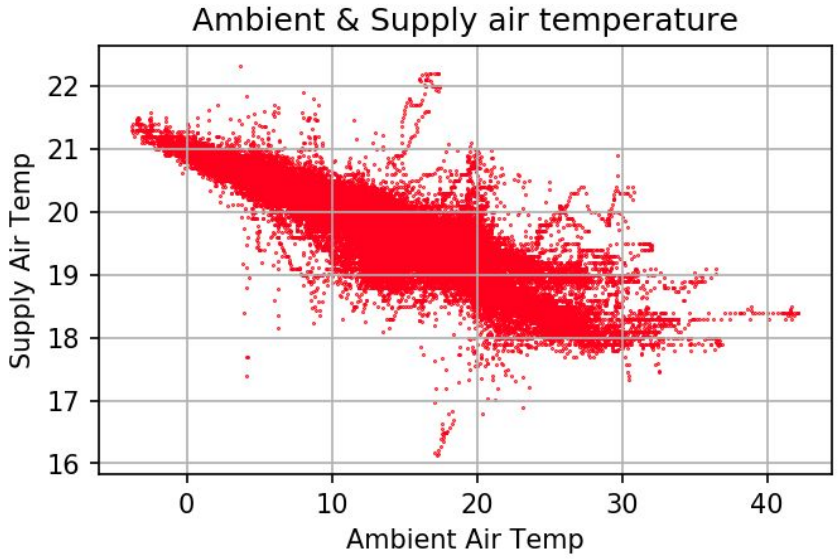


Figure 13. Scatter Plot

(4) Irregular control patterns:
In general, this is a negative correlation and several outliers can be found in the scatter plot. The supply air temperature is sometimes higher or low compared to the rest of the data. For example. some irregular behaviors occur when the ambient air temperature is around 15 degrees.

Control settings
This is a negative correlation, which means the relationship between two variables (x = supply air temperature, y = ambient air temperature) whereby they move in opposite directions. If variables x and y are negatively correlated, as x increases in value, y will decrease; similarly, if x decreases in value, y will increase. Thus, the control settings can be: as the supply air temperature is increased, the ambient air temperature will decrease, similarly, if supply air temperature decreases in value, ambient air temperature will increase.

(5) ACF & PACF
To create ACF and PACF plot, it is necessary to fill the missing data, therefore, the '***ffill***' method was used to fill the missing data. Specifically, the ***Statsmodel library*** (Statsmodels.2019) was used to create ACF and PACF plots.

- **ACF** measures the correlation between the present value of the series with its past values. ACF contains components like trend and seasonality. This allows data analysts to estimate predictions and find patterns in the data. In the plot (figure x), we could find out there is a trend in the data, the autocorrelations tend to have positive values that decrease as the lags increase.

- **PACF** measure the correlation between a variable and a lag of itself that is not explained by correlations at all lower-order lags. In the plot, the partial autocorrelations significant spike only at lag 1, 2,3,4, especially the value at lag 3 is negative. The partial autocorrelations tend to have both positive and negative values. It is impossible to find a global trend is PACF.

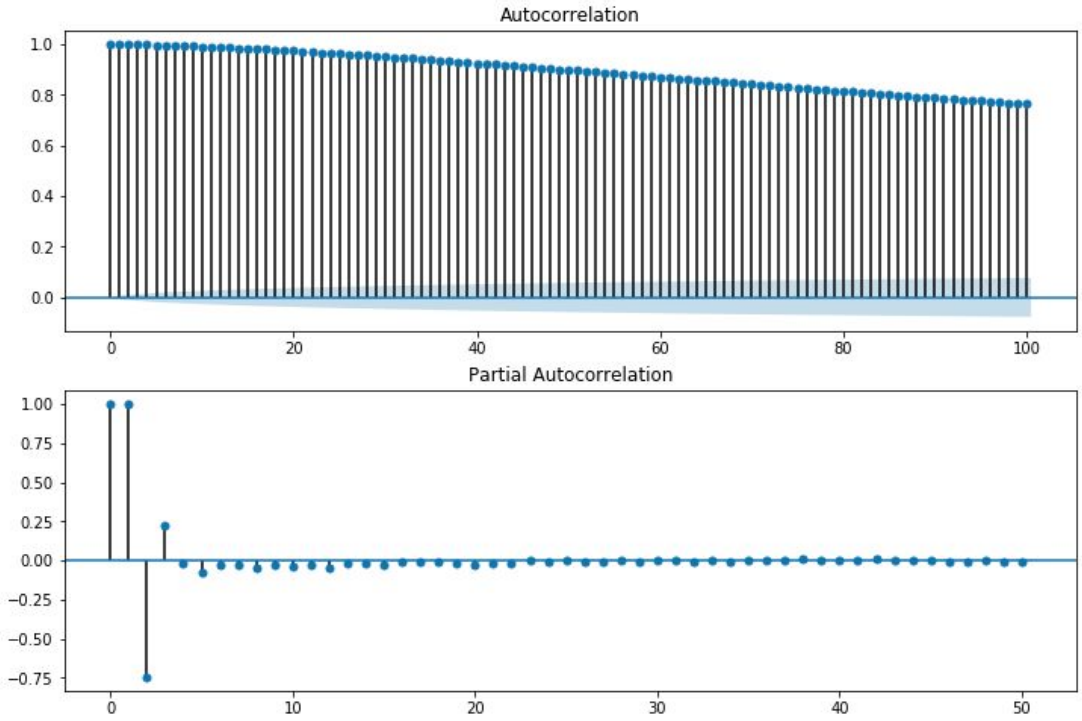


Figure 14. ACF & PACF

(6) Linear Regression

The **linear** regression model was used to fit the regression model and **MSE (mean square error)** metric was used to evaluate the quality of the fitting.

First of all, it is important to assess the performance of model, therefore, the k-folds cross validation (k = 3 was used to access its performance. As we can see the score of the model is around 0.79, which indicates that the performance of model is relatively decent.

```
model = LinearRegression()
scores = []
kfold = KFold(n_splits=3, shuffle=True)
for i, (train, test) in enumerate(kfold.split(X, Y)):
 model.fit(X.iloc[train,:], Y.iloc[train,:])
 scores.append(model.score(X.iloc[test,:], Y.iloc[test,:]))
print(scores)
```

```
[0.7941475425982178, 0.7899141336785844, 0.7934601199424676]
```

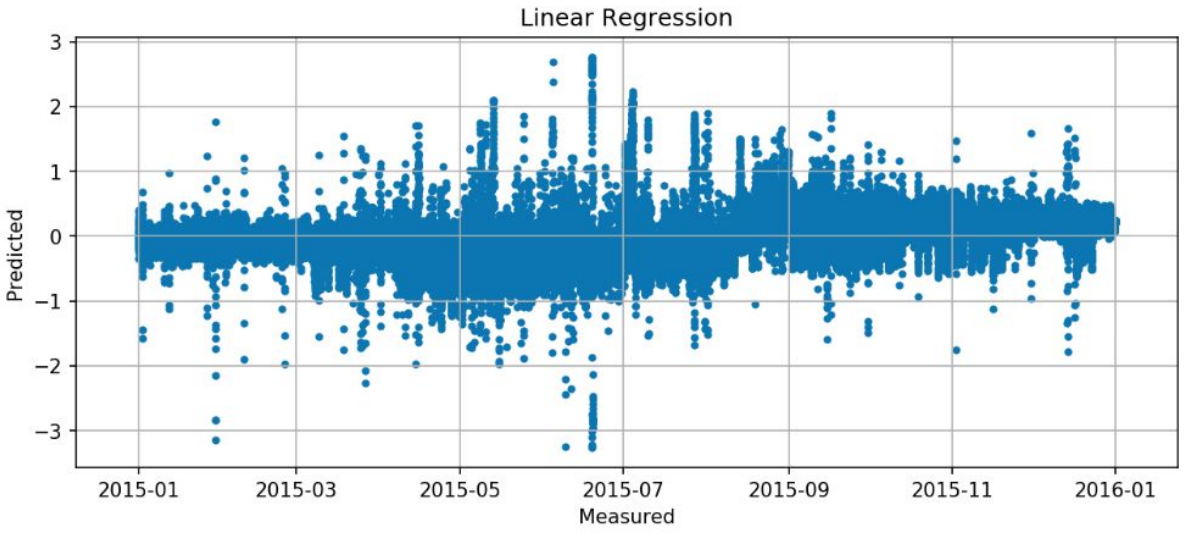
```
plt.figure(figsize = (10,4), dpi = 150)
plt.scatter(X.index,Y-predictions, marker='.')
mean_square_error_metric = sum((Y-predictions)['Supply Air Temp']**2)/len(Y)

plt.title('Linear Regression')
plt.xlabel('Measured')
plt.ylabel('Predicted')
plt.savefig('Interploate fillin values')

plt.grid(True)
plt.show()
print(mean_square_error_metric)
```

Figure 15. Scatter Plot

MSE is the mean of the squared difference between estimate and the data. This is an identical calculation to the calculation of the variance of a statistic, where the estimate is the mean (Mean squared error 2019). Ideally the MSE should be 0, however, in reality it is impossible to achieve 0 in a model, therefore, smaller MSE generally indicates a better estimate. In this model, the MSE is 0.091, this indicates that we have obtained score.




```
0.09146927211112549
```

Figure 16. Scatter Plot

Reference

1. Wen, J. T. (2018). *Intelligent building control systems: a survey of modern building control and sensing strategies*. Cham, Switzerland: Springer.
2. Ross, P. R., & Wensveen, S. A. (2010). *Designing aesthetics of behavior in interaction: Using aesthetic experience as a mechanism for design*. International Journal of Design, 4(2), 3-13.
3. Verhaart, J., Li, R., & Zeiler, W. (2018). *User interaction patterns of a personal cooling system: A measurement study*. Science and Technology for the Built Environment, 24(1), 57-72.
4. Extrapolation. (2019, October 9). Retrieved from https://en.wikipedia.org/wiki/Extrapolation.
5. Interpolation. (2019, October 9). Retrieved from https://en.wikipedia.org/wiki/Interpolation.
6. Duke's Fuqua . (n.d.). Retrieved from https://www.fuqua.duke.edu/.
7. Knowledge Hub. (2019). Retrieved from https://www.datawatch.com/in-action/knowledge-hub/.
8. Alteryx Designer . (2019). Retrieved from https://www.alteryx.com/
9. Means squared error.(2019.) Retrieve from https://en.wikipedia.org/wiki/Mean_squared_error.
10. Stasmodels.(2019). Retrieved from https://www.statsmodels.org/stable/index.html.

Appendix

1. Warning function excel