Assignment 1

STE6246-002

Time and series prediction and classification of the number of people in a room with sensor data

Abstract

The assignment 1 was basically about the time and series to predict the number of people in a class room using monitored sensor data. There were two sensors mounted on the wall of the classroom which captures data in continuous timeframe. And the log of the people of the room was filled in the excel sheet with other reference data by the students of 5IT.

The K Nearest Neighbor (K-NN), Long and Short-Term Memory (LSTM) recurrent neural network were the two approaches made to analyze the sensor data to predict the actual number of people in the classroom.

Both techniques are robust in working with the data set which is big as it continuously monitors the sensor values.

Introduction

First of all the recording of data with the sensors from Serinus[3] technology was installed in the various rooms at UiT Narvik. They were 3 different wall mounted sensors which captures data like Humidity, Carbon Dioxide (CO_2), Luminance, Noise, Temperature inside the monitored rooms every five minutes.

Then a log of additional data was recorded according to the present situation of the room as we feel how noisy, how well ventilated, or the door opened or closed, lights on or off, and the number of person in the room to an excel sheet. These data were manually created whenever a person goes out or comes into the room. The main point to do this was to prepare the set of data to use for the classification and the prediction of the number of person in the room present at the time with the data from the sensors.

We are familiar with the three approaches for this process the first one is LSTM recurrent and Neural Network and the second one is the K-NN, and CART method with boosting. And need to provide a recommendation after using these models to analyze the data from the sensor and the prediction's accuracy, or usability.

Preparation of Data

Three sensor monitors were installed in different rooms, one was in the classroom where we study. There are about 7/8 person in the room in total, because it was not a big classroom with many people inside, so it maybe a downside that this fact and the experiments doesn't represent the bigger classroom with more people. It is obvious that the data of the sensor will goes up when there is increase in the number of people. The sensors measure indoor temperature in Celsius, CO_2 levels in ppm (parts per million), the relative humidity of the room, noise in decibels, the brightness of the room in Lux and has a binary measurement of detected movement.

On the other side we recorded the data ourselves, so that we can gather information about the context of the room and compare it with the sensor data to predict the real scenario and vice versa. We discussed about what data we need to gather for this purpose and created an excel sheet with data like date and time, number of people in the room, condition of noise, light, ventilation, or the lecture is going on or not, opening and closing of doors and windows, computers that were currently running in the room.

Working with this, principle component analysis (PCA) was used to find out which data will carry significance to apply the algorithm to predict the number of person in the room.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Date	Time	Temp	CO2	Noise	Brightnes	Relative-H	Motion	Nr-People	Lights-on/	External-r	Internal-n	Door-oper	Nr-Windo	Lecture	Nr-Compu	Weekday	Vents-on/off
09-08-17 0:00	10:57	23.99768	463	39.72588	48.42	29.05628	1	4	On	VeryLow	Moderate	No	1	No	5	0	On
09-08-17 0:00	11:02	24.00775	464	40.13122	48.06	28.82432	1	4	On	VeryLow	Moderate	No	1	No	5	0	On
09-08-17 0:00	11:07	24.01782	466	45.10759	47.34	28.95861	1	5	On	VeryLow	High	No	1	No	6	0	On
09-08-17 0:00	11:12	24.03797	466	45.58804	48.06	28.67782	1	5	On	VeryLow	High	No	1	No	6	0	On
09-08-17 0:00	11:17	24.04804	468	40.13122	47.7	28.82432	1	3	On	VeryLow	Moderate	No	1	No	6	0	On
09-08-17 0:00	11:22	24.04804	469	40.13122	47.7	28.62898	1	3	On	VeryLow	Moderate	No	1	No	6	0	On
09-08-17 0:00	11:27	24.04804	471	40.13122	46.98	28.58625	1	3	On	VeryLow	Moderate	No	1	No	6	0	On
09-08-17 0:00	11:32	24.06818	471	40.13122	47.34	28.35429	1	5	On	VeryLow	Low	No	1	No	6	0	On
09-08-17 0:00	11:37	24.06818	470	40.52193	46.98	28.15895	1	7	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	11:42	24.07826	472	52.33219	46.26	28.2261	1	7	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	11:47	24.07826	474	40.13122	46.26	28.28104	1	7	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	11:52	24.06818	475	40.13122	46.62	28.44586	1	7	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	11:57	24.06818	476	44.85834	45.54	28.09791	1	7	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	12:02	24.06818	476	50.58435	45.18	28.13454	1	7	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	12:07	24.04804	476	40.13122	45	27.95141	1	7	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	12:12	24.04804	476	40.13122	46.98	27.97583	1	6	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	12:17	24.06818	476	45.58804	45.36	28.18337	1	6	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	12:22	24.04804	476	65.21848	45.54	28.37871	1	6	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	12:27	24.06818	476	40.13122	45.9	27.85374	1	4	On	VeryLow	Low	No	1	No	8	0	On
09-08-17 0:00	12:32	24.04804	477	39.72588	45.54	27.76828	1	5	On	VeryLow	Low	No	1	No	8	0	On
09-08-17 0:00	12:37	24.04804	477	40.13122	46.44	27.74387	1	7	On	VeryLow	Low	No	1	No	8	0	On
09-08-17 0:00	12:42	24.04804	477	43.50981	45.54	27.81712	1	8	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	12:47	24.08833	477	50.01508	46.26	27.54853	1	8	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	12:52	24.10847	477	54.60776	46.08	27.59736	1	7	On	VeryLow	High	No	1	No	8	0	On
09-08-17 0:00	12:57	24.11854	478	61.88641	45.9	27.3593	1	6	On	VeryLow	High	No	1	No	7	0	On
09-08-17 0:00	1:02	24.12862	479	45.58804	46.08	27.57295	1	7	On	VeryLow	High	No	1	No	7	0	On
09-08-17 0:00	1:07	24.14876	482	49.40782	47.88	27.58515	1	7	On	VeryLow	High	No	1	No	7	0	On
09-08-17 0:00	1:12	24.14876	484	40.13122	50.4	27.58515	1	7	On	VeryLow	High	No	1	No	7	0	On
09-08-17 0:00	1:17	24.14876	484	57.56421	51.84	27.48749	1	8	On	VeryLow	High	No	1	No	7	0	On
09-08-17 0:00	1:22	24.15883	486	67.3007	54.54	27.63399	1	7	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	1:27	24.15883	487	56.11171	47.88	27.53632	1	7	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	1:32	24.15883	487	40.13122	50.58	27.56074	1	7	On	VeryLow	Moderate	No	1	No	8	0	On
09-08-17 0:00	1:37	24.15883	487	40.13122	50.76	27.67061	1	7	On	VeryLow	Moderate	No	1	No	8	0	On

Figure 1: Sample Data

Data Correlation

The precision of the accuracy of the data to be predicted on has more than greater significance of some figures rather than all figures and to find out that a principal component analysis PCA was carried out.

Sensor and Logsheet Data Variance

[0.92675258 0.06783797]

<<<<<<<<<----------

Sensor and Logsheet Data Co-rrelation

	Time	Temp	CO2	Noise	Brightness
Time	1.000000	-0.039182	-0.035953	-0.004976	-0.079772
Temp	-0.039182	1.000000	0.188421	-0.166342	0.063991
CO2	-0.035953	0.188421	1.000000	-0.073866	0.151985
Noise	-0.004976	-0.166342	-0.073866	1.000000	0.327752
Brightness	-0.079772	0.063991	0.151985	0.327752	1.000000
Relative-humidity	-0.062399	0.212186	0.088145	-0.243039	-0.242881
Motion	-0.137441	0.066039	0.128456	0.410990	0.704028
Nr-People	-0.144945	0.080503	0.109417	0.493122	0.612977
Weekday	0.010811	0.027236	-0.130568	-0.123749	-0.077930

	Relative-humidity	Motion	Nr-People	Weekday
Time	-0.062399	-0.137441	-0.144945	0.010811
Temp	0.212186	0.066039	0.080503	0.027236
CO2	0.088145	0.128456	0.109417	-0.130568
Noise	-0.243039	0.410990	0.493122	-0.123749
Brightness	-0.242881	0.704028	0.612977	-0.077930
Relative-humidity	1.000000	-0.239696	-0.313332	-0.158490
Motion	-0.239696	1.000000	0.783687	-0.101612
Nr-People	-0.313332	0.783687	1.000000	-0.139882
Weekday	-0.158490	-0.101612	-0.139882	1.000000

<<<<<<<<<<<<----------

<<<<<<<<<<<<----------

Sensor and Logsheet Data Co-variance

	Time	Temp	CO2	Noise	Brightness
Time	12.000999	-0.138985	-10.676634	-0.073626	-6.470972
Temp	-0.138985	1.048420	16.538008	-0.727459	1.534249
CO2	-10.676634	16.538008	7348.057458	-27.044070	305.069529
Noise	-0.073626	-0.727459	-27.044070	18.242261	32.779097
Brightness	-6.470972	1.534249	305.069529	32.779097	548.308593
Relative-humidity	-0.786578	0.790572	27.494033	-3.777223	-20.694840
Motion	-0.204600	0.029057	4.731739	0.754310	7.084062
Nr-People	-0.799018	0.131167	14.925043	3.351496	22.840286
Weekday	0.018727	0.013945	-5.596586	-0.264289	-0.912472

	Relative-humidity	Motion	Nr-People	Weekday
Time	-0.786578	-0.204600	-0.799018	0.018727
Temp	0.790572	0.029057	0.131167	0.013945
CO2	27.494033	4.731739	14.925043	-5.596586
Noise	-3.777223	0.754310	3.351496	-0.264289
Brightness	-20.694840	7.084062	22.840286	-0.912472
Relative-humidity	13.240783	-0.374797	-1.814290	-0.288376
Motion	-0.374797	0.184654	0.535879	-0.021834
Nr-People	-1.814290	0.535879	2.532150	-0.111303
Weekday	-0.288376	-0.021834	-0.111303	0.250034

Figure 2: Co-rrelation and Covariance

Figure 3: PCA (Principal Component Analysis) plot

As we can see from the plot there is high correlation between Noise, motion, Time, Relative Humidity, Temperature, Brightness etc. Now it can be easy to focus on the data which are more useful for the analysis.

Methodology

For this assignment 1 a solution for LSTM and K-NN clustering has been created separately using the findings from then PCA. The set of feature vectors are created which gives the results for the analysis using the methods mentioned.

LSTM Neural Network

It is known that the assignment is focused to some instinct around monitored sensor data combined and taken during the certain interval of time series. LSTM is such type of model which I think could be natural for time based series analysis. The data header from the time series are divided into input and output vector to find out the prediction of the number of people according to the input sensor data, Time, Noise, Brightness, Relative-Humidity and Motion.

The collected data is formatted in excel sheet so that we can import the excel sheet to the program in python to build and analyze the relation between data headers. Training set and Test set of the data is prepared which is 30% of the whole data is used for testing.

Python has standard built in libraries and Keras[2] was used to implement the LSTM model of the data to get the prediction from the significant input terms.

```
  set1['Time'] = set1['Time'].astype(float)
   Nr-People   Time      Noise  Brightness  Relative-humidity  Motion
0        4.0  10.57  39.725883   48.419998          29.056282     1.0
1        4.0  11.02  40.131222   48.060001          28.824320     1.0
2        5.0  11.07  45.107590   47.340000          28.958612     1.0
3        5.0  11.12  45.588043   48.060001          28.677816     1.0
4        3.0  11.17  40.131222   47.700001          28.824320     1.0
(5085, 5) (5085,) (2180, 5) (2180,)
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_51 (LSTM)               (None, 80)                27520
_________________________________________________________________
dense_51 (Dense)             (None, 1)                 81
=================================================================
Total params: 27,601
Trainable params: 27,601
Non-trainable params: 0
_________________________________________________________________
```

```
Random 50 data
>> Prediction No. =1  Actual No. =5.0
Random 50 data
>> Prediction No. =0  Actual No. =1.0
>> Prediction No. =0  Actual No. =0.0
>> Prediction No. =4  Actual No. =1.0
>> Prediction No. =0  Actual No. =2.0
>> Prediction No. =1  Actual No. 1.0
>> Prediction No. =0  Actual No. 1.0
Test RMSE: 0.856
<<<<<<<<<<<<Validation>>>>>>>>>>>>:

Predicted No. of people: 2.286
Actual No. of people: 4
ERROR: 1.714 !!!


Predicted No. of people: 2.760
Actual No. of people: 5
ERROR: 2.240 !!!
```

Figure 4: LSTM Model and Prediction

K-NN

K-Nearest Neighbor is well known among as it is the simplest model of machine learning, it is bit slow but can be used process to find out all the Euclidian distance between the objects to get the information in which group or neighbor it belongs to, and according to that principal it is easy to identify and predict data categorically. The value of K plays a significant role in this process, if K=1 then each data object is assigned to its nearest neighbor, or if K>1 then the origin of the data object is determined by majority voting.

K=1 is a basic approach, whereas K>1 requires some preprocessing of the data to prepare the input vector. The same as LSTM input data were used against output vector No of people.

Results

The RMSE Root Mean Square Error of the data set in this LSTM model is 84% approx.

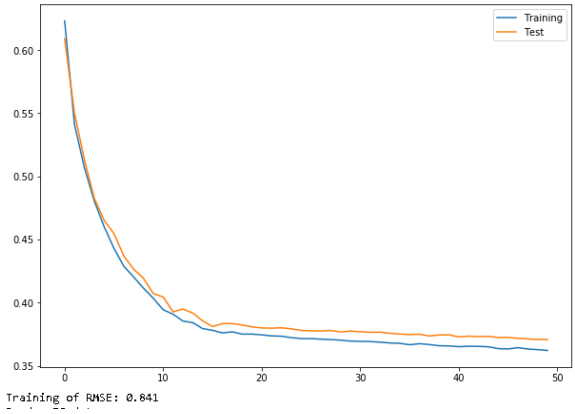

Figure 5: LSTM Plot

There is also a validation set of several input vectors to check whether the prediction falls into the values properly or not.

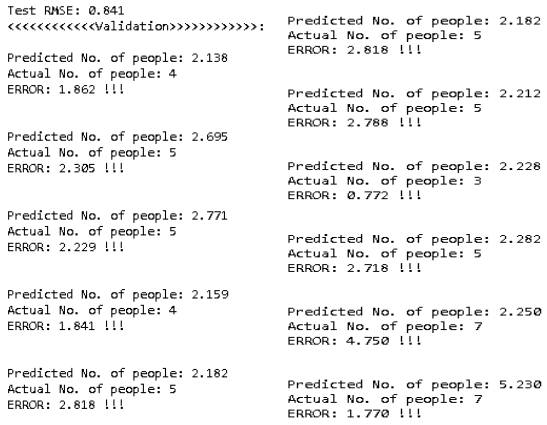

Figure 6: Validation of LSTM model

The same thing was done for the K-NN validation set to find out the difference, between them.

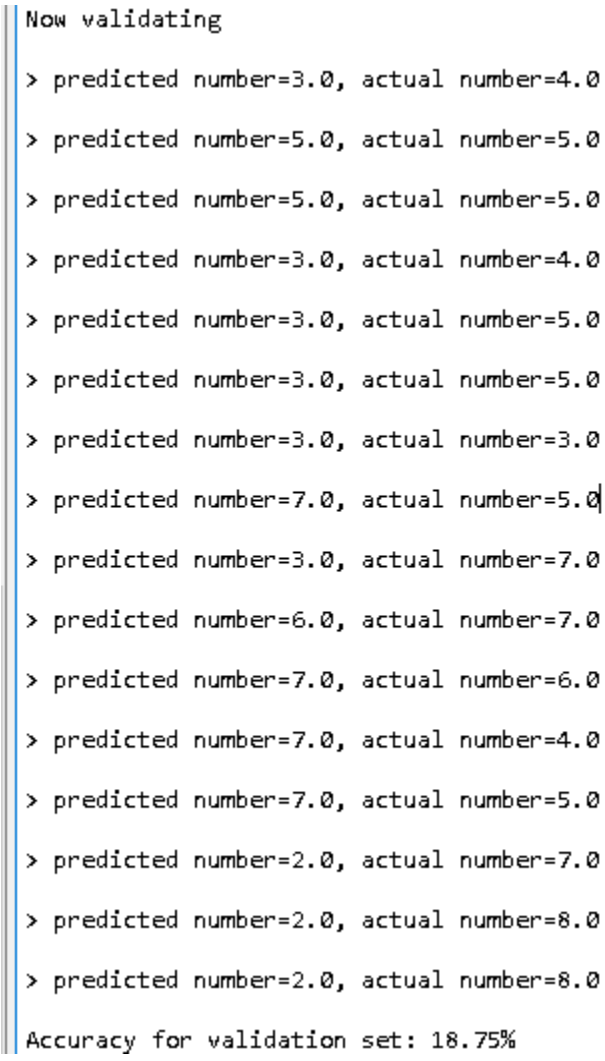

Figure 7: Validation for K-NN

Discussion

The two methods were used for the assignment given both of the methods have accuracy level has some visual after the validation. The LSTM method has some average error of approx. 20-30% for each vector input.

The same is for K-NN model about 18% approximate accuracy, it can be seen clearly that the approximate accuracy rate of the LSTM model is higher to that of K-NN model.

The data which are recorded during the time stamp were manually inserted according to the feelings of the person who is making an entry, the sensor used during this process Serinus are also mounted on the walls and they of course can't produce exact sensing data as well of the whole big classroom.

Conclusion

The task asked for the recommendation of one of the method, so I would definitely go for the LSTM model as it has better accuracy ration than to of K-NN method model. The accuracy of the both method is nearly equivalent though.

I think the data collected and prepared have some human errors as well to be find out exactly it is vary difficult to be precise in this matter. But the recommendation could be LSTM recurrent neural network in this approach.

References

1. Dr. Jason Brownlee. Tutorial to implement k-nearest neighbors in python from scratch. https://machinelearningmastery.com/ tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/, 2014. Online; accessed 22.10.2017.

2. Keras.io. Keras. https://keras.io/, 2017. Online; accessed .02/11.2017.

3. Serinus technology. Optimalt inneklima, arealutnyttelse og energiforbruk — optimal bygningsforvaltning; arbeidsmiljø, læringsmiljø, arealutnyttelse og energiforbruk. https:// serinustechnology.no/, 2017. Online; accessed 25.10.2017.