Project STANDY
Uzu Uzu Index

Preprocessing
—
Building an ML model
—
Serving

Tang Li Qun

a reference

'Machine Learning Applications for Data Center Optimization' - Jim Gao, Google

some data sources

Japan Meteorological Agency (JMA)

RESAS API

Japan Open Data

Initial Observation

I spoke with my wife,
who is a teacher at an elementary school.

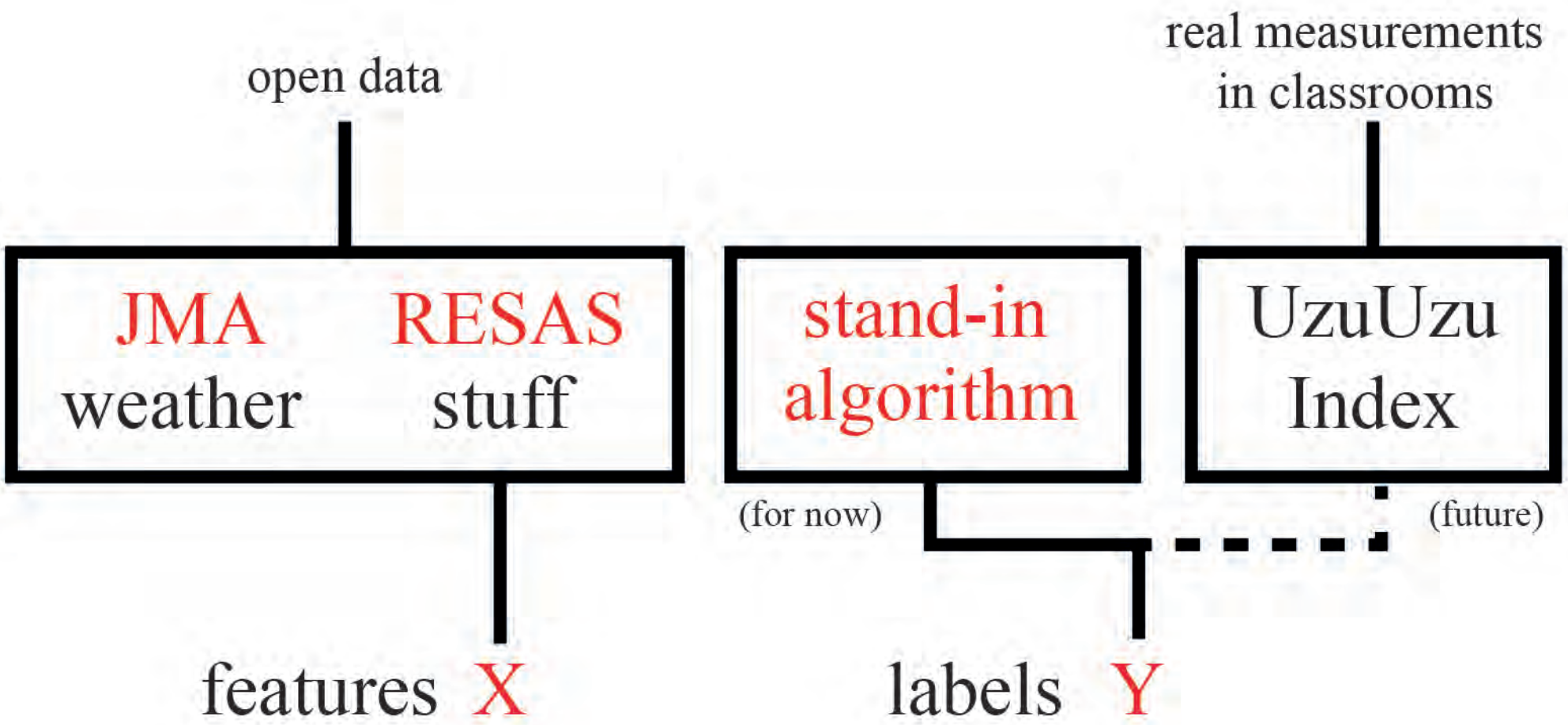
she feels
that some relation between them exists.

For example,
children become very noisy
after rainy days.

We also discussed this idea
with a weather forecaster.
She suggested that
low atmospheric pressure
is also an important key indicator.

We thought about
defining that feeling
as an index, named “**UZU UZU index**”.

"UZU UZU" is a Japanese word
meaning a child’s feeling of raring to go.



train model to predict
UUI as a continuous number

the algorithm

meant to provide our best hypothesis of what an Uzu Uzu Index is

uzuuzu compounded by effects of time

Day	Thu	Fri	Sat	Sun	Mon	Tue
Atm. Pressure	1005.4 hPa	1009 hPa	1011.3 hPa	1011.6 hPa	1009.6 hPa	1004.1 hPa
Daylight	10.1H	4.9H	7.4H	4.5H	9.6H	9.6H

$$\sum_{n=0}^{2} LowPressure(n) * (3-n) * Holiday(n) * Urban\ Index$$

= labels (Y)

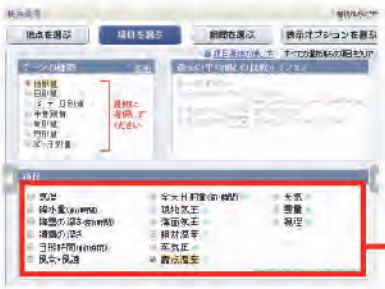
weather

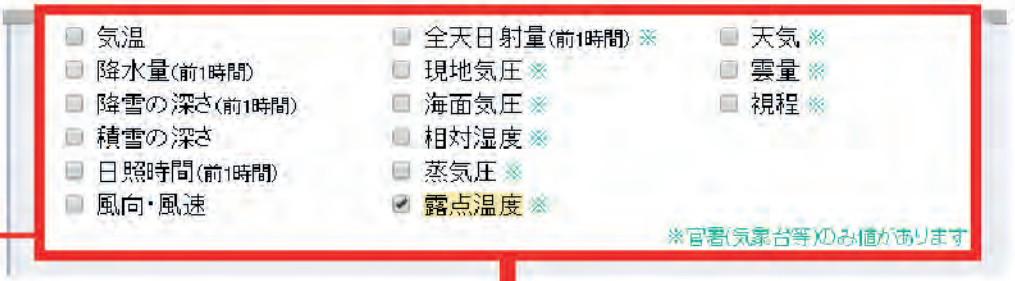
weekend / holidays

num. of places to play nearby

Weather data from JMA

http://www.data.jma.go.jp/gmd/risk/obsdl/index.php





server.js | 1 Asian Care Page(Main Page).pdf | play_area_re

```
1   ダウンロードした時刻:2017/08/05 14:30:12
2
3   ,東京,東京,東京
4   年月日時,現地気圧(hPa),現地気圧(hPa),現地気圧(hPa
5   ,,品質情報,均質番号
6   2016/7/29 1:00:00,1011.2,8,1
7   2016/7/29 2:00:00,1011.2,8,1
8   2016/7/29 3:00:00,1010.9,8,1
9   2016/7/29 4:00:00,1011.0,8,1
10  2016/7/29 5:00:00,1011.2,8,1
11  2016/7/29 6:00:00,1011.4,8,1
12  2016/7/29 7:00:00,1011.6,8,1
13  2016/7/29 8:00:00,1011.8,8,1
14  2016/7/29 9:00:00,1011.5,8,1
15  2016/7/29 10:00:00,1011.3,8,1
16  2016/7/29 11:00:00,1011.4,8,1
17  2016/7/29 12:00:00,1011.0,8,1
18  2016/7/29 13:00:00,1010.4,8,1
19  2016/7/29 14:00:00,1010.2,8,1
20  2016/7/29 15:00:00,1009.7,8,1
21  2016/7/29 16:00:00,1009.8,8,1
22  2016/7/29 17:00:00,1009.8,8,1
23  2016/7/29 18:00:00,1010.1,8,1
24  2016/7/29 19:00:00,1010.7,8,1
```

tokyo_cloud_cover.csv | tokyo_weather.csv | tokyo_tem

```
4   年月日時,雲量(10分比),雲量(10分比),雲量(10分比)
5   ,,品質情報,均質番号
6   2016/7/29 1:00:00,,0,1
7   2016/7/29 2:00:00,,0,1
8   2016/7/29 3:00:00,3,8,1
9   2016/7/29 4:00:00,,0,1
10  2016/7/29 5:00:00,,0,1
11  2016/7/29 6:00:00,2,8,1
12  2016/7/29 7:00:00,,0,1
13  2016/7/29 8:00:00,,0,1
14  2016/7/29 9:00:00,7,8,1
15  2016/7/29 10:00:00,,0,1
16  2016/7/29 11:00:00,,0,1
17  2016/7/29 12:00:00,5,8,1
18  2016/7/29 13:00:00,,0,1
19  2016/7/29 14:00:00,,0,1
20  2016/7/29 15:00:00,2,8,1
21  2016/7/29 16:00:00,,0,1
22  2016/7/29 17:00:00,,0,1
23  2016/7/29 18:00:00,1,8,1
24  2016/7/29 19:00:00,,0,1
25  2016/7/29 20:00:00,,0,1
26  2016/7/29 21:00:00,0,8,1
27  2016/7/29 22:00:00,,0,1
```

林野面積（総面積）

指定地域に対しての林野面積（総面積）を年単位で返します。

```
GET  api/v1/forestry/land/forStacked
```

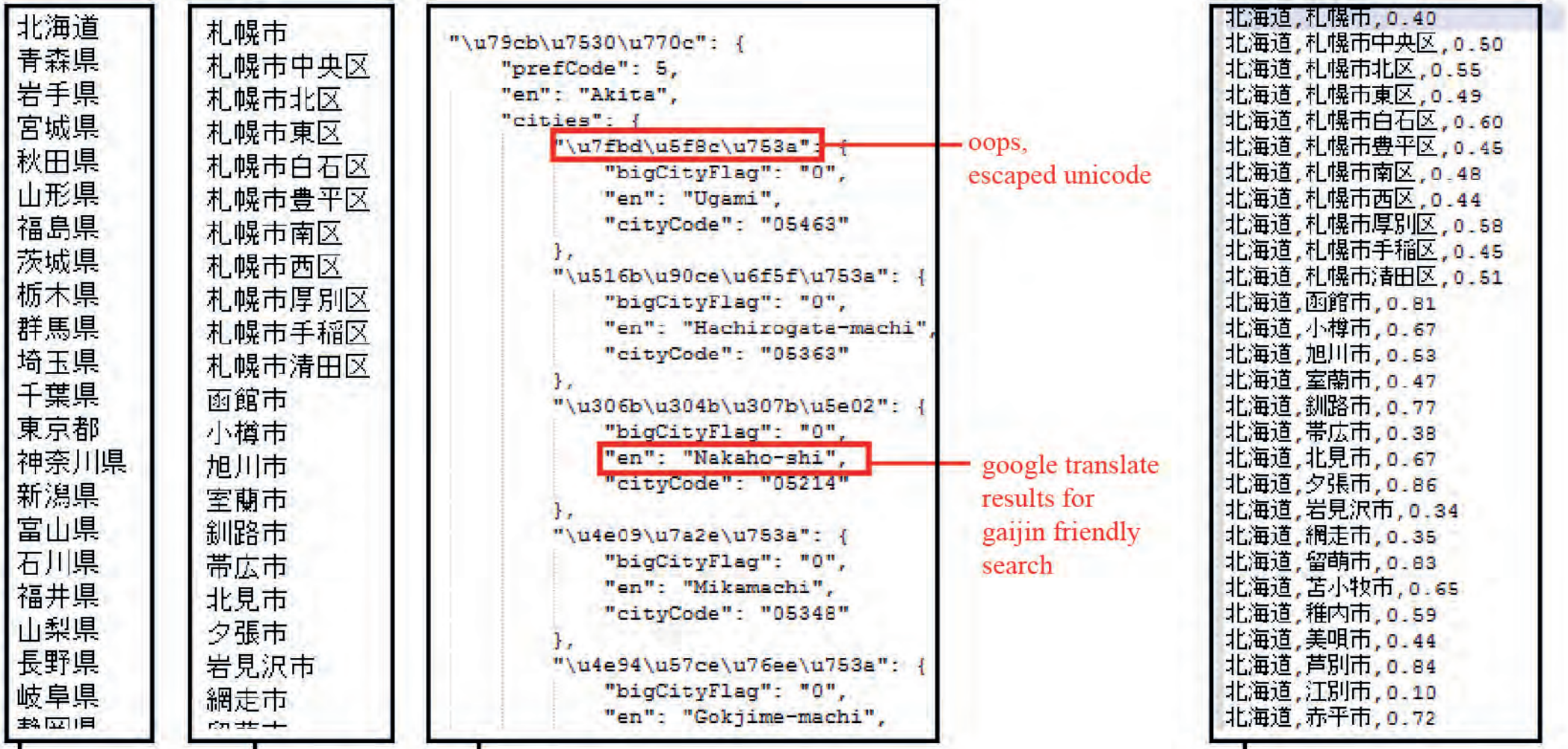
parameters

inputs : prefCode, cityCode

Name	Description
prefCode	都道府県コード
cityCode	市区町村コード 「すべての市町村」を選択する場合は「-」を送ります。
	追加エリアコード 他地域と合算した値を取得する際に使用するパラメータです。「addArea=都道府県コード_市区町村コ

wanted to get the Urban Index for all cities in japan because we were thinking big

didn't want to call the API 3 times to get 1 number



prefecture + cities = prefCode + cityCode json dict

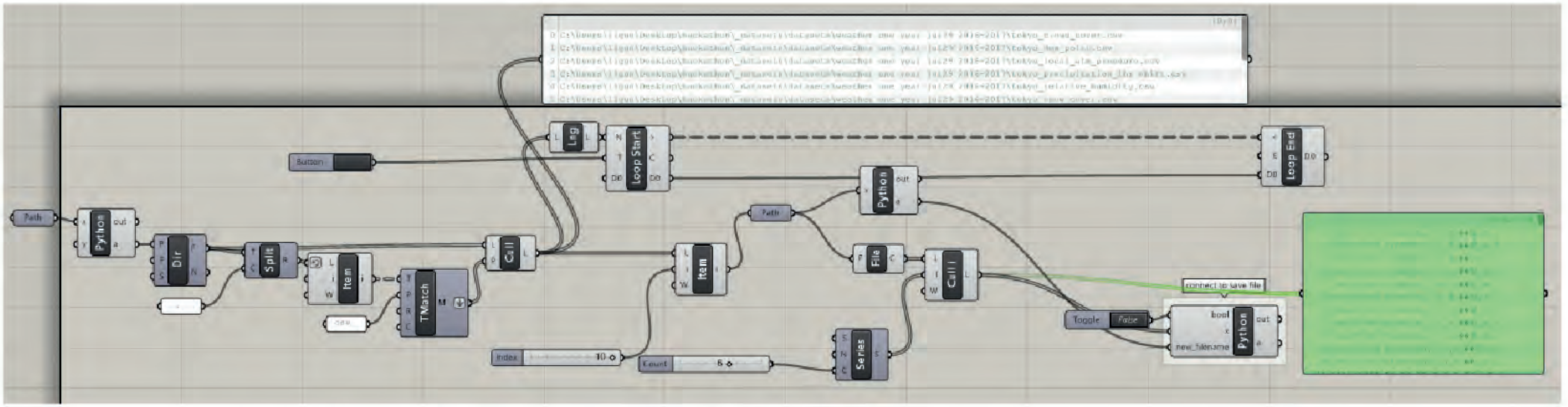
prefCode + cityCode => RESAS API = forest (area)

wikipedia table = total land area

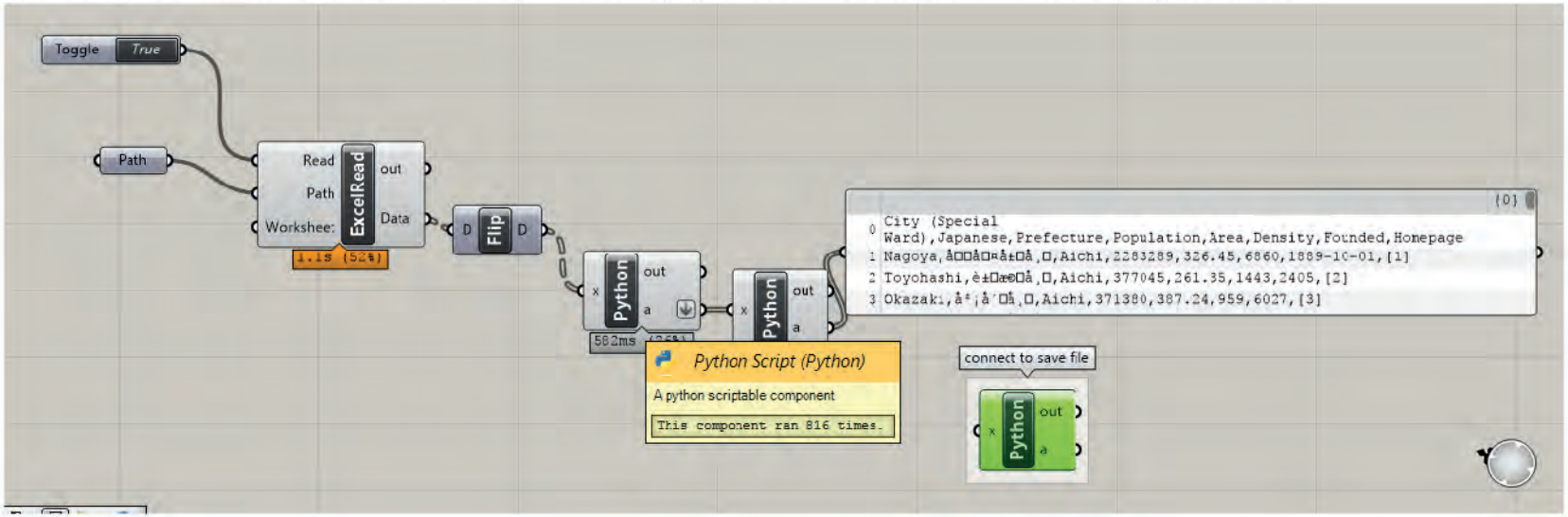
= Urban Index JPN

because everything was in japanese, there were some encoding issues in python that i didn't want to deal with. so I used grasshopper to do some simple cleaning

deleting headers because pandas wasn't happy with japanese characters (i think)

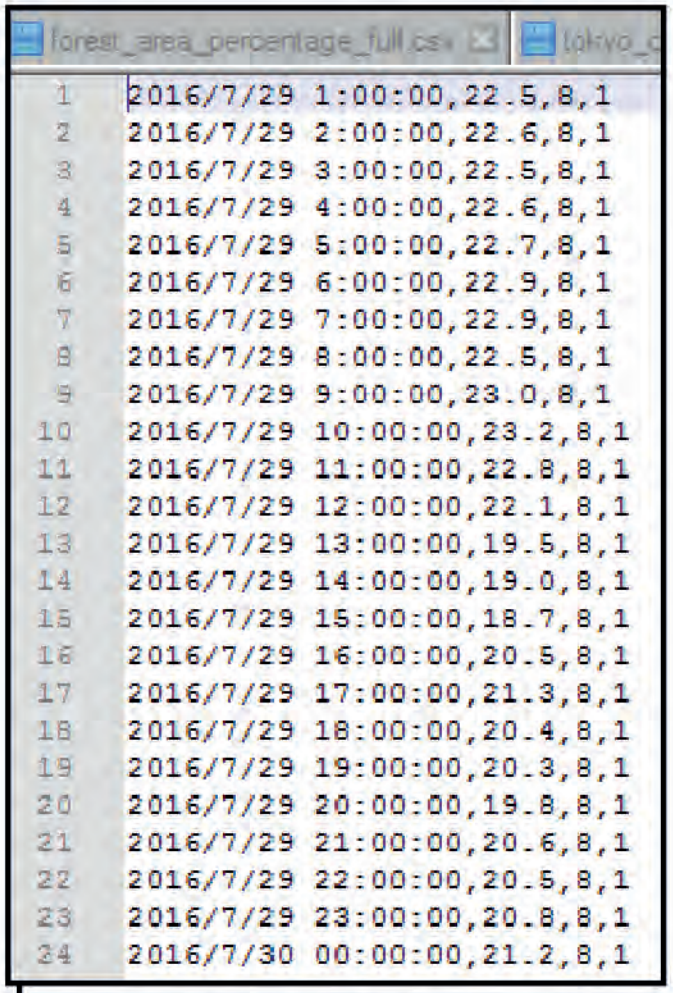


reformatting an excel into csv with japanese character encoding intact



different languages and encodings are..
ちょっと大変です。

weather data * 12 + urban index = features



	forest_area_percentage_full.csv
1	2016/7/29 1:00:00,22.5,8,1
2	2016/7/29 2:00:00,22.6,8,1
3	2016/7/29 3:00:00,22.5,8,1
4	2016/7/29 4:00:00,22.6,8,1
5	2016/7/29 5:00:00,22.7,8,1
6	2016/7/29 6:00:00,22.9,8,1
7	2016/7/29 7:00:00,22.9,8,1
8	2016/7/29 8:00:00,22.5,8,1
9	2016/7/29 9:00:00,23.0,8,1
10	2016/7/29 10:00:00,23.2,8,1
11	2016/7/29 11:00:00,22.8,8,1
12	2016/7/29 12:00:00,22.1,8,1
13	2016/7/29 13:00:00,19.5,8,1
14	2016/7/29 14:00:00,19.0,8,1
15	2016/7/29 15:00:00,18.7,8,1
16	2016/7/29 16:00:00,20.5,8,1
17	2016/7/29 17:00:00,21.3,8,1
18	2016/7/29 18:00:00,20.4,8,1
19	2016/7/29 19:00:00,20.3,8,1
20	2016/7/29 20:00:00,19.8,8,1
21	2016/7/29 21:00:00,20.6,8,1
22	2016/7/29 22:00:00,20.5,8,1
23	2016/7/29 23:00:00,20.8,8,1
24	2016/7/30 00:00:00,21.2,8,1

北海道,札幌市,0.40
北海道,札幌市中央区,0.50
北海道,札幌市北区,0.55
北海道,札幌市東区,0.49
北海道,札幌市白石区,0.60
北海道,札幌市豊平区,0.45
北海道,札幌市南区,0.48
北海道,札幌市西区,0.44
北海道,札幌市厚別区,0.58
北海道,札幌市手稲区,0.45
北海道,札幌市清田区,0.51
北海道,函館市,0.81
北海道,小樽市,0.67
北海道,旭川市,0.53
北海道,室蘭市,0.47
北海道,釧路市,0.77
北海道,帯広市,0.38
北海道,北見市,0.67
北海道,夕張市,0.86
北海道,岩見沢市,0.34
北海道,網走市,0.35
北海道,留萌市,0.83
北海道,苫小牧市,0.65
北海道,稚内市,0.59
北海道,美唄市,0.44
北海道,芦別市,0.84
北海道,江別市,0.10
北海道,赤平市,0.72

weather data * 12 + urban index = features (X)

still pretty dirty....

*note to self : weather data is in general actually quite strongly correlational

delete

turn into binary
0 = weekday
1 = weekend

to clean

sparse data

umm. missing values

the same value everywhere...

	datetime	datetime	atm_pressure	cloud_cover	dew_point	rain_mm	rh	snow_cover	sunlight_hrs	temp_celsius	visibility	weather_rating	forest area coverage	LowPressure	UUI2
62	31/07/2016 15:00	Sun	1007.7	9	23.9	0	68		1.5	30.5	20	4	0.087407407	0.564636292	5.752958056
86	01/08/2016 15:00	Mon	1004.4	8	25.7	0	87		1.05	28.1	20	2	0.087407407	0.715042106	5.353713109
110	02/08/2016 15:00	Tue	1004.1	10-	23.6	0	82		0.86	26.9	25	4	0.087407407	0.727108216	4.740681444
134	03/08/2016 15:00	Wed	1004.8	10-	23.5	0	69		2.26	29.8	15	3	0.087407407	0.698465216	4.264654186
158	04/08/2016 15:00	Thu	1005.6	2	23.1	0	60		2.51	31.8	20	2	0.087407407	0.663738697	4.115254741
182	05/08/2016 15:00	Fri	1006.1	5	24.1	0	62		2.49	32.3	15	2	0.087407407	0.641067406	3.94914483
206	06/08/2016 15:00	Sat	1005.4	2	23.7	0	57		2.47	33.4	20	2	0.087407407	0.672607017	5.981515612
230	07/08/2016 15:00	Sun	1006.2	2	21.5	0	53		2.67	32.3	20	2	0.087407407	0.63645254	7.150210716
254	08/08/2016 15:00	Mon	996.9	10-	22.3	0	56		1.73	32.2	25	3	0.087407407	0.918339745	6.646043429
278	09/08/2016 15:00	Tue	998	6	19.6	0	38		2.47	36.2	20	2	0.087407407	0.900249511	5.810333102
302	10/08/2016 15:00	Wed	1004	10-	22.7	0	61		0.97	31.1	15	4	0.087407407	0.731058579	4.912014502
326	11/08/2016 15:00	Thu	1006.3	10-	18.9	0	51		2.41	30.1	25	3	0.087407407	0.631812418	4.257803921
350	12/08/2016 15:00	Fri	1005.9	6	20.4	0	55		2.29	30.4	15	2	0.087407407	0.650218549	3.94533906
374	13/08/2016 15:00	Sat	1003.8	5	20.8	0	56		1.96	30.6	10	2	0.087407407	0.738850006	6.365349551
398	14/08/2016 15:00	Sun	1001.3	10-	19.8	0	57		0.97	29.2	15	4	0.087407407	0.823464725	8.546406924
422	15/08/2016 15:00	Mon	999.7	10-	22.9	0	71		0.81	28.7	15	4	0.087407407	0.865296948	7.367449757
446	16/08/2016 15:00	Tue	998.5	10-	24.9	0.5	84		0.7	27.8	10	10	0.087407407	0.890903179	6.050232883
470	17/08/2016 15:00	Wed	996.4	7	24.8	0	66		2.52	31.9	10	2	0.087407407	0.925532055	5.42369947
494	18/08/2016 15:00	Thu	1003.7	10	25.8	16	100		0.03	25.8	4	15	0.087407407	0.742690545	4.970038924
518	19/08/2016 15:00	Fri	1003.9	9	23.8	0	64		1.96	31.4	30	4	0.087407407	0.734972599	4.615830944
542	20/08/2016 15:00	Sat	1002	10-	24.2	0	76		1.29	28.8	20	4	0.087407407	0.802183889	7.025739076
566	21/08/2016 15:00	Sun	1003.3	10-	24.9	0	71		1.27	30.8	20	3	0.087407407	0.757679639	8.489785988
590	22/08/2016 15:00	Mon	983.9	10	23.4	3.5	97		0.3	23.9	4	10	0.087407407	0.993438807	7.615402754
614	23/08/2016 15:00	Tue	1001.9	10-	24.9	0	80		0.97	28.7	20	4	0.087407407	0.805338416	5.918252141
638	24/08/2016 15:00	Wed	1005.3	10-	23.4	0	77		0.49	27.8	20	4	0.087407407	0.676995856	4.635103208

runaway column of indices from somewhere

normalize

what is this weather rating?
is it correlational data?

Spent most of Hackathon Day 1

we have no time.
time to get on with the model.

sketches are fast and nice.

grasshopper for rhino
visual programming tool

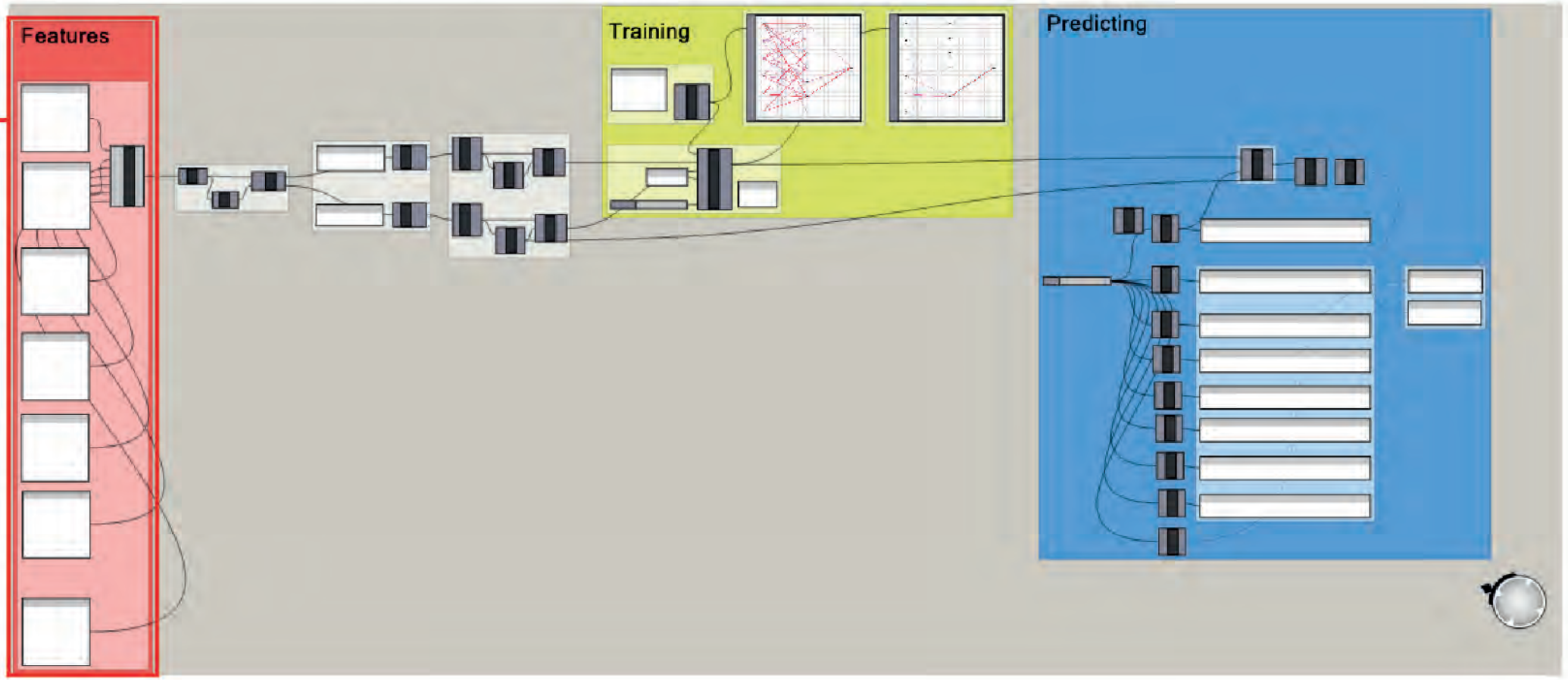
+

owl for grasshopper
machine learning library

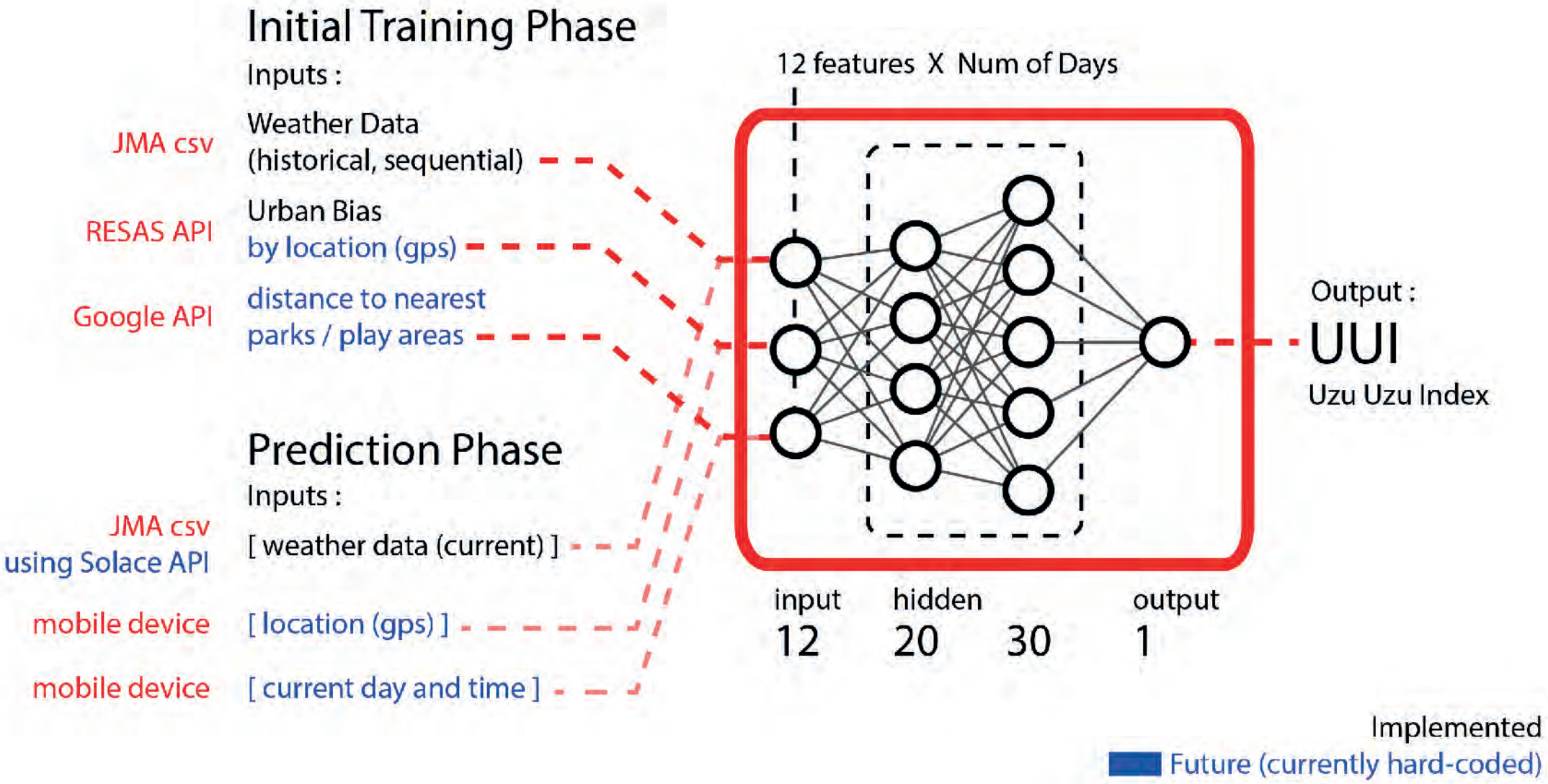
http://www.grasshopper3d.com/
http://www.food4rhino.com/app/owl

draft Neural Network demo

Date	2016.9.1	2016.9.2	2016.9.3	2016.9.4	2016.9.5	2016.9.6	2016.9.7
Day	Thu	Fri	Sat	Sun	Mon	Tue	Wed
Atm. Pressure	1005.4 hPa	1009 hPa	1011.3 hPa	1011.6 hPa	1009.6 hPa	1004.1 hPa	1001.4 hPa
Daylight Time	10.1H	4.9H	7.4H	4.5H	9.6H	9.6H	2.3H
Precipi-tation	0mm	0mm	0mm	3mm	0mm	0mm	4.5mm
Humidity	68%	76%	77%	84%	76%	74%	85%
Ave. Temp	27.0℃	26.7℃	27.2℃	26.7℃	28.4℃	28.8℃	27.1℃



now repeat in Keras / Tensorflow



important part

activation functions,
number of neurons (20,30) and
number of hidden layers (2) were tuned for this dataset

```
def regression_nn():
    # define regression model
    # create model
    model = Sequential()

    # input layer + 20 neuron hidden layer
    model.add(Dense(20, input_dim=feature_len, kernel_initializer='normal'))
    model.add(Activation('relu'))

    # 30 neuron hidden layer
    model.add(Dense(30, kernel_initializer='normal'))
    #model.add(BatchNormalization())
    model.add(Activation('relu'))

    # final nonlinear bit
    model.add(Dense(1, kernel_initializer='normal'))
    model.add(BatchNormalization())
    model.add(Activation('tanh'))
    # compile
    # adam optimizer is super fast compared to sgd
    model.compile(loss='mse', optimizer='adam')
    return model
```

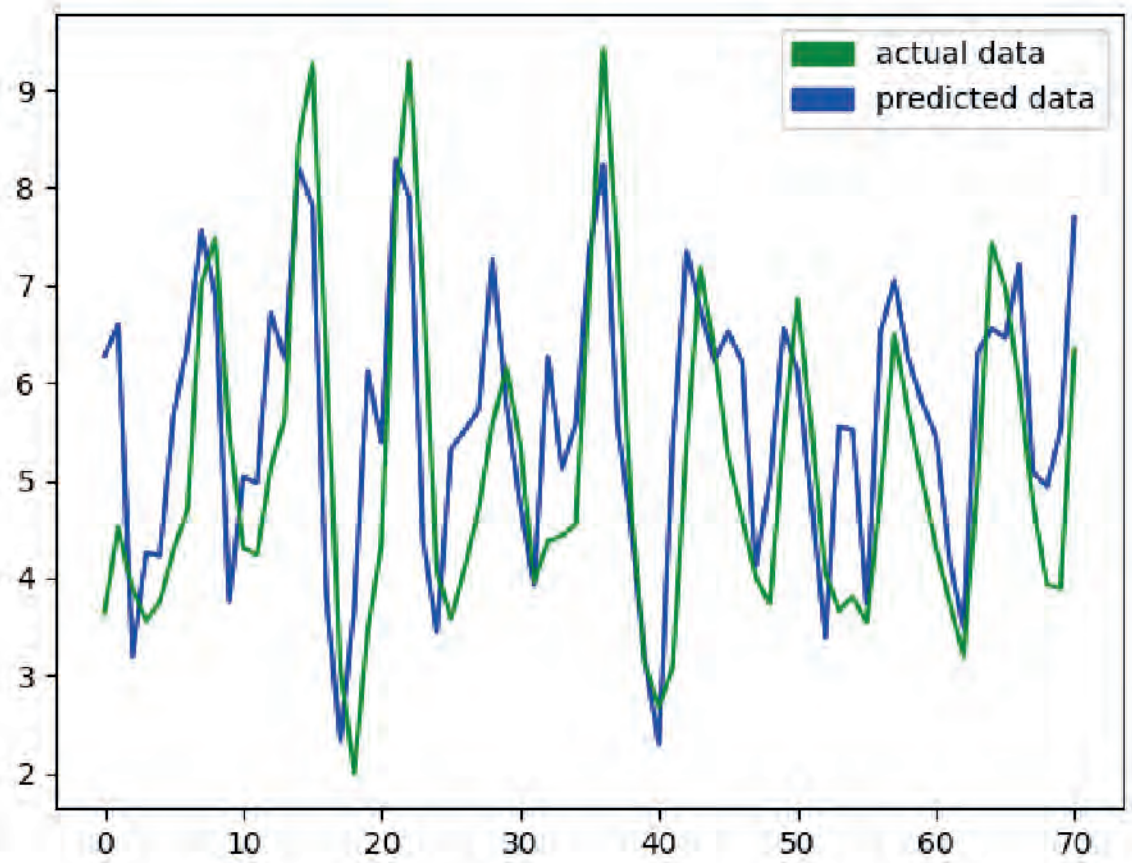
input + full connected (20)

fully connected (30)

output (1)

results

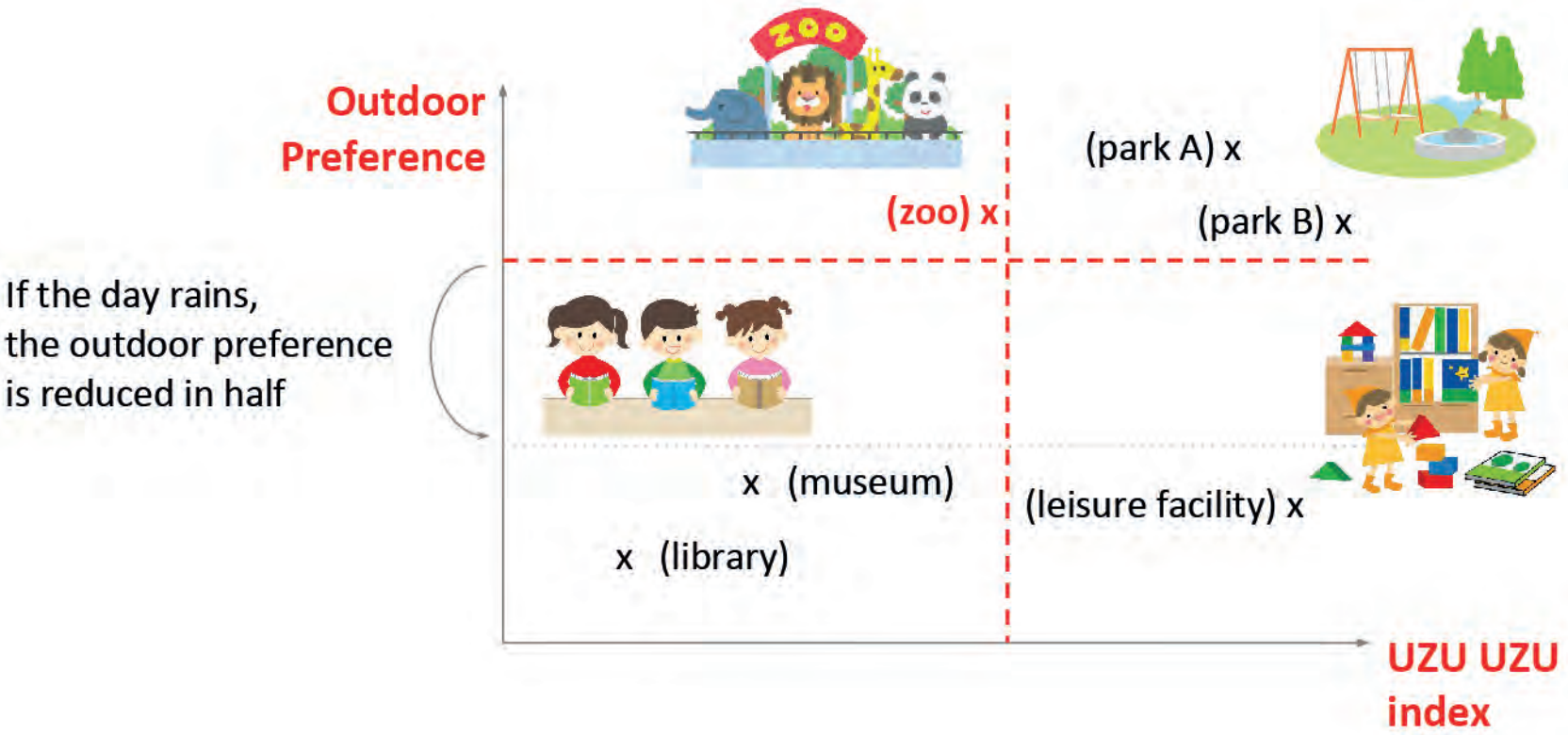
train / test = 80 / 20
test set accuracy = 85 - 88%



essentially the neural network has learnt the algorithm (without us telling it)

now to use the output
to calculate something!

Recommendation of places to play and like-minded friends



serving

```
# sample child
test_input = ['Wed',5,20.8,0.2,56,0,1.96,30.6,10,2,0.087407407,0.655]
outdoor_pref = 0.8
forecast_weather = test_input[3]
```

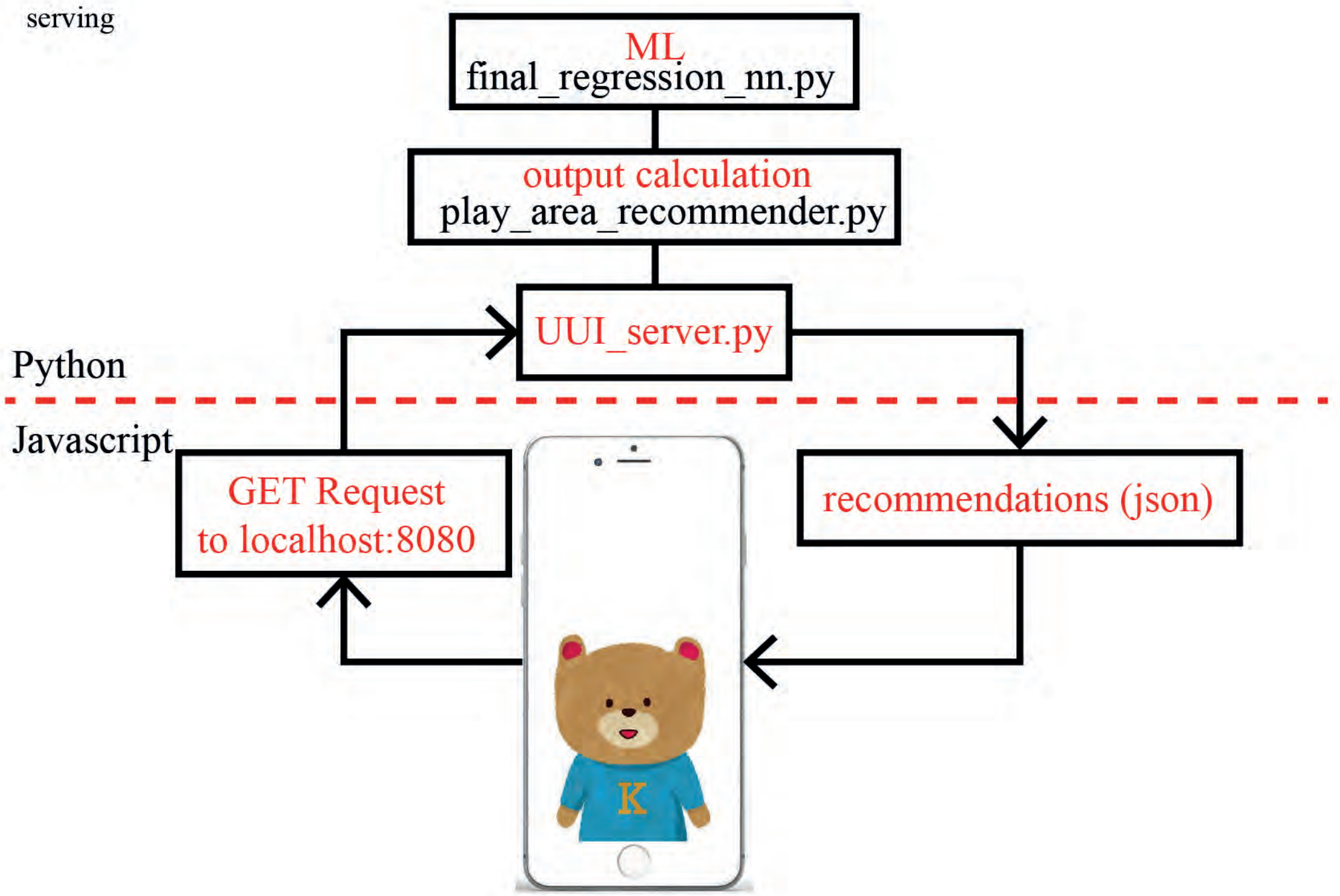
Angsty child A

```
216/292 [=====================>.........] - ETA: 0s - loss: 0.0048
292/292 [==============================] - 0s - loss: 0.0059
test mse :  0.0255443790663
C:\Users\liqun\AppData\Local\Programs\Python\Python35\lib\site-package
average accuracy 87.55%
  warnings.warn(DEPRECATION_MSG_1D, DeprecationWarning)
Saved model to disk
0.4882217347621918
recommendations : ['leisure_facility', 'park_b']
```

recommendations

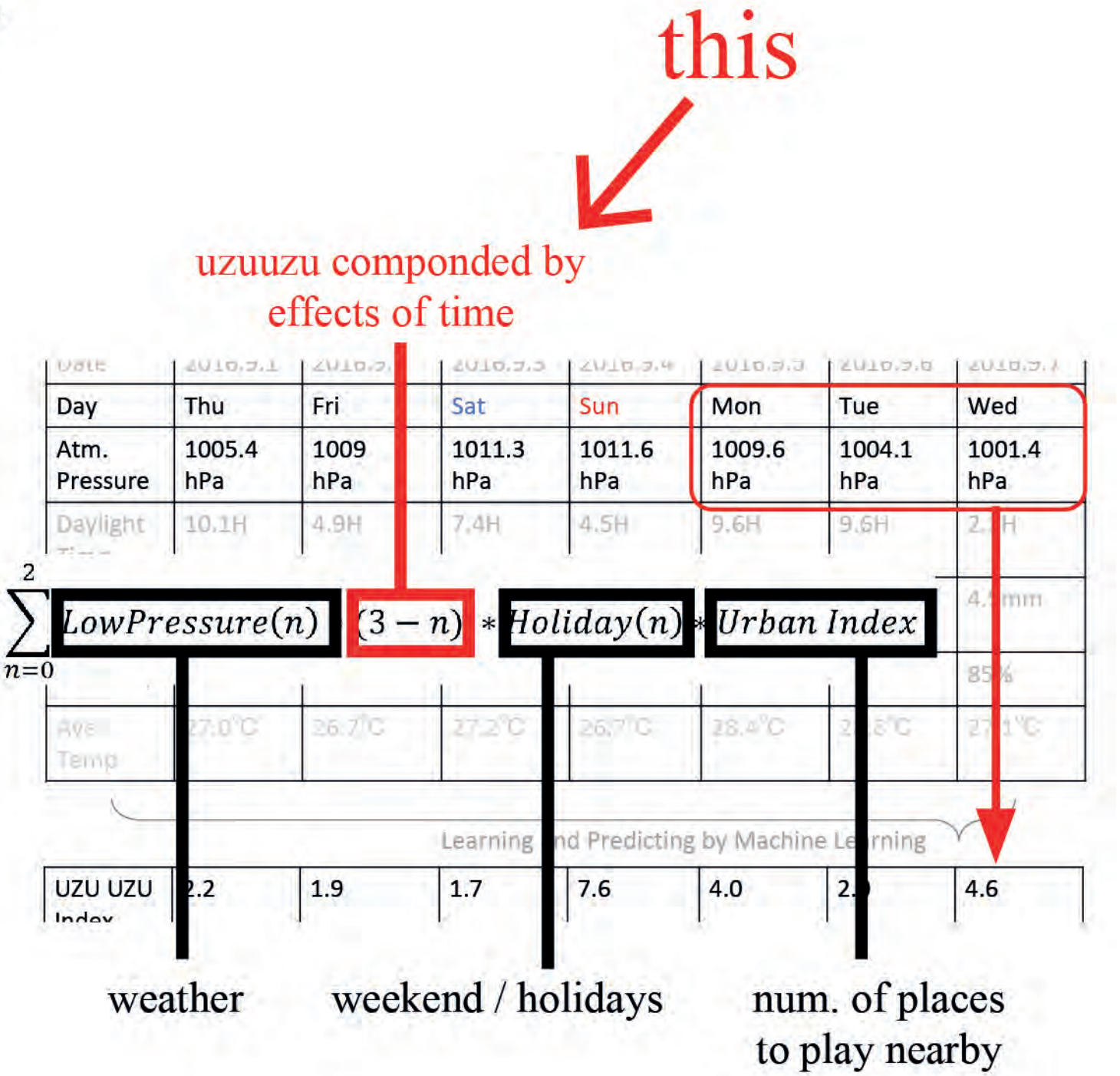
```
C:\WINDOWS\system32\cmd.exe

C:\Users\liqun\Desktop\hackathon\__backend_dev\UzuUzuApi>python3 uui_ser
please wait while python loads all modules...
Using TensorFlow backend.
./modules/uzuuzuindex_nn\final_regression_nn.py
Starting server at http://192.168.100.59:8080
```

wrap in an API
and make a server


serving
ML
final_regression_nn.py
output calculation
play_area_recommender.py
UUI_server.py
Python
Javascript
GET Request
to localhost:8080
recommendations (json)

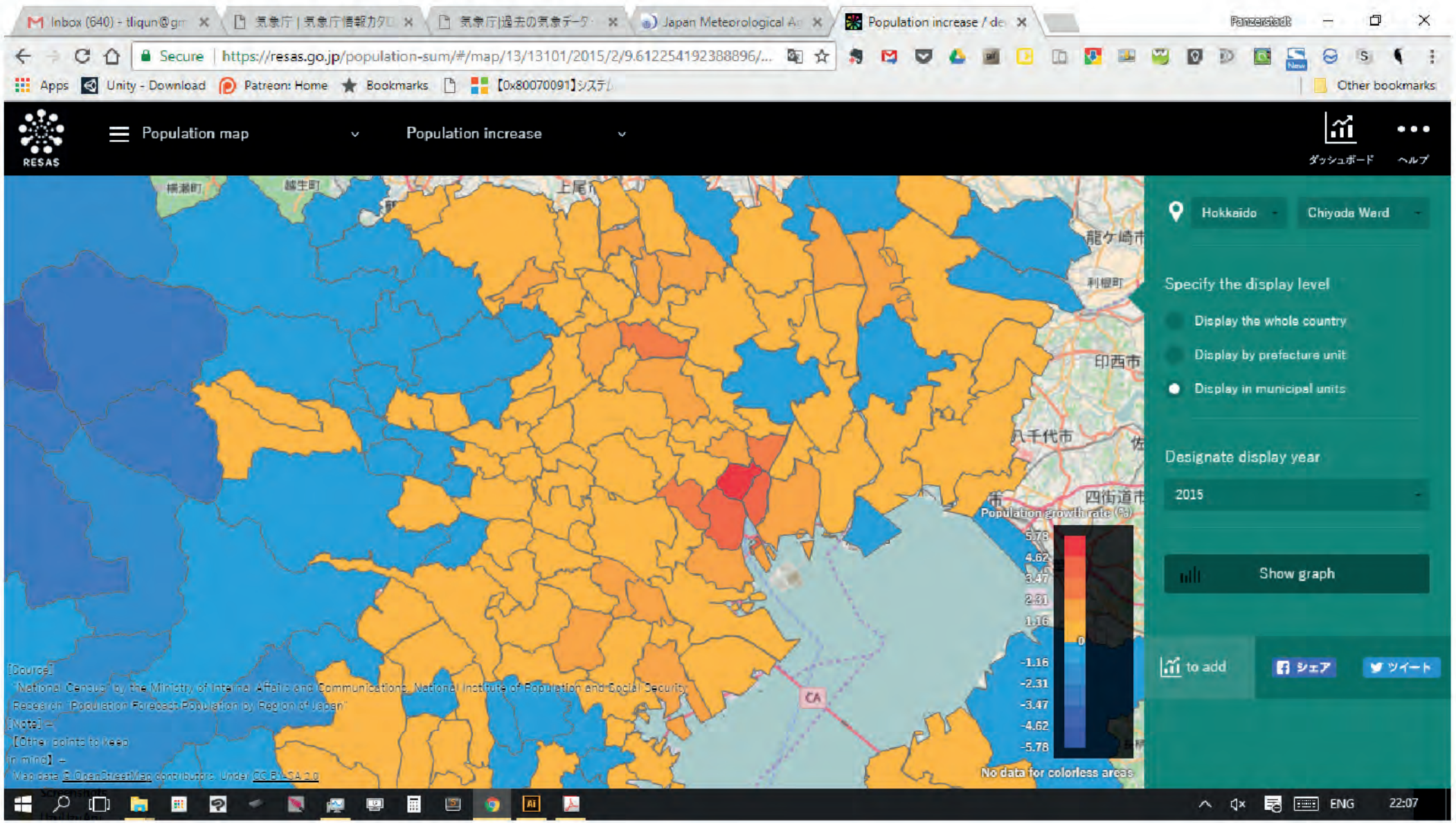
next steps.

todo


this
uzuuzu componded by
effects of time
Day
Thu
Fri
Sat
Sun
Mon
Tue
Wed
Atm. Pressure
1005.4 hPa
1009 hPa
1011.3 hPa
1011.6 hPa
1009.6 hPa
1004.1 hPa
1001.4 hPa
Daylight
10.1H
4.9H
7.4H
4.5H
9.6H
9.6H
85%
Learning and Predicting by Machine Learning
UZU UZU Index
2.2
1.9
1.7
7.6
4.0
4.6
$\sum_{n=0}^{2} LowPressure(n) \quad (3-n) * Holiday(n) * Urban\ Index$
weather
weekend / holidays
num. of places
to play nearby

todo

Economic / population data from RESAS https://resas.go.jp/#/13/13101



todo

more goverment data not found in RESAS

http://www.data.go.jp/data/dataset

e site list | Examples of public data utilization | communication | Information for developers

データセットを検索...

Use Hint: Multiple keyword search function can be used for keyword search by "AND", "OR", "NOT".
Example: Administrative AND Environment NOT White Paper → A data set containing "administrative" and "environment" and not including "white paper" will be searched.

Download metadata

Relevance | desce | 20 ca

19,368 data sets found

Criminal statistical data _ Heisei 28 (January - December) 【fixed value】

Criminal Investigation Support Office of Criminal Investigation Office of Criminal Investigation of National Police Agency.
January-December Edition of Crime Statistics Materials Created by Analysis Officials 【Final Value】

XLS PDF
Release date: 2017-02-10
Metadata update date: 2017-07-25

Food Safety Committee Quarterly magazine "Food Safety"

Food safety committee quarterly magazine "Food Safety"

PDF
Release date:
Metadata update date: 2017-07-12

第３表　着工建築物：用途別、構造別（建築物の数、床面積の合計、工事費予定額）

平成28年計分

用途		構造	総計			木造			鉄骨鉄筋コンクリート造			鉄
			建築物の数（棟）	床面積の合計（㎡）	工事費予定額（万円）	建築物の数（棟）	床面積の合計（㎡）	工事費予定額（万円）	建築物の数（棟）	床面積の合計（㎡）	工事費予定額（万円）	建築物の数（棟）
全国計												
		全建築物計	609,333	132,962,092	2,631,501,828	449,580	56,579,193	939,064,967	647	2,289,491	70,599,016	17,227
A		居住専用住宅	516,160	77,469,638	1,449,494,088	421,962	51,999,983	856,327,708	266	502,546	6,773,017	11,110
B		居住専用準住宅	2,645	954,298	19,210,965	1,517	361,521	6,158,697	2	6,001	120,350	149
C		居住産業併用建築物	6,574	3,791,604	100,691,802	3,807	640,591	11,006,830	32	192,422	5,179,090	1,051
	11	居住農林水産業併用	237	32,805	525,515	162	24,051	347,060	0	0	0	2
	12	居住鉱業，採石業，砂利採取業，建設業併用	379	92,334	1,620,300	244	39,167	617,580	5	2,857	52,600	34
	13	居住製造業併用	222	77,797	1,540,562	125	20,666	351,853	1	14	200	20
	14	居住電気・ガス・熱供給・水道業併用	113	34,215	598,817	65	10,732	178,621	0	0	0	10
	15	居住情報通信業併用	39	34,697	1,559,061	19	3,225	49,977	0	0	0	5
	16	居住運輸業併用	53	16,764	384,363	23	3,793	64,841	0	0	0	4
	17	居住卸売業，小売業併用	1,388	1,089,847	28,380,961	546	89,077	1,540,714	10	92,897	2,657,700	377
	18	居住金融業，保険業併用	57	76,352	1,752,531	20	2,907	50,302	1	4,833	120,000	14
	19	居住不動産業併用	348	434,199	11,446,069	125	20,729	348,362	5	4,074	86,400	118
	20	居住宿泊業，飲食サービス業併用	741	513,843	15,045,045	482	81,121	1,482,705	2	338	6,300	132
	21	居住医療，福祉併用	652	447,649	10,748,053	351	77,680	1,459,692	2	739	20,390	111
	22	居住教育，学習支援業併用	145	55,173	1,630,198	97	15,228	281,520	0	0	0	14
	23	居住その他のサービス業併用	1,731	846,161	16,056,355	1,254	200,126	3,422,739	7	80,929	1,945,500	161
	24	居住公務併用	71	44,905	1,103,215	50	6,122	112,666	1	5,741	290,000	8
	25	他に分類されない居住産業併用	398	250,985	8,506,147	244	15,967	758,232	0	0	0	41
D		農林水産業用建築物	7,321	2,165,981	21,289,176	3,105	698,267	4,923,787	5	438	3,400	78
F		鉱業，採石業，砂利採取業，建設業用建築物	4,102	1,107,916	17,602,764	1,474	201,466	2,755,672	4	535	5,700	63

todo

measure UzuUzuIndex
from classrooms.

Thank you.

https://github.com/panzerstadt/UzuUzuIndex