Hong Kong Car Park

Availability Analysis and Predictive Modelling (Basic)

Group DS- 12

Ng Wing Kei, James

Choi Ka Hou, KZ

Chun Long Hei, Caius

Executive Summary

SITUATION

- Hong Kong is a busy city with heavy car traffic.
- Often times, people have a hard time finding car parks with vacancy.
- There is a use case for using predictive modeling to enable people to find car parks when they need to

KEY QUESTION

How can we provide <u>a prediction of the amount of car park spaces available at a given car park</u> when user provides <u>the full date and time?</u>

DATA WORKFLOW

- Data publicly available through government APIs
- Findings from Exploratory Data Analysis
- Process of developing model and solutions

NEXT STEPS

What can we do to improve the model going forward?

Since 2016, The number of private cars had increased by more than 50 %

Problem:

- Harder and harder for people to find vacant car park spots
- Often times, weekend shoppers who have private cars have to go “car-park” hopping before being able to park their cars
- Frustrating and inefficient processes

Proposed Solutions:

Leveraging public government parking vacancy APIs and data, provide a predictive model that can give users amount of vacancy in a given car park based on an input of date and time

Data Collection

Basic Information of Participating Car Parks

```
basic_df
```

	park_id	name_en	name_tc	name_sc	displayAddress_en	displayAddress_tc	displayAddress_sc	latitude	longitude	district_en	district_tc	district_sc	contactNo	opening_status	height	remark_en
0	tdc1p1	Lee Garden One Car Park	利園一期停車場	利园一期停车场	33 HYSAN AVENUE, Wan Chai District, Hong Kong	香港 灣仔區 希慎道 33號	香港 湾仔区 希慎道 33号	22.278598	114.184793	Wan Chai	灣仔區	湾仔区		open	2.0	Height Limit: \nElectric Vehicle Charging Serv...
1	tdc1p3	Leighton Car Park	禮頓中心停車場	礼顿中心停车场	77 LEIGHTON ROAD, Wan Chai District, Hong Kong	香港 灣仔區 禮頓道 77號	香港 湾仔区 礼顿道 77号	22.277768	114.183100	Wan Chai	灣仔區	湾仔区		open	1.9	Height Limit: \nElectric Vehicle Charging Serv...
2	tdc1p2	Lee Garden Two Car Park	利園二期停車場	利园二期停车场	28 YUN PING ROAD, Wan Chai District, Hong Kong	香港 灣仔區 恩平道 28號	香港 湾仔区 恩平道 28号	22.278252	114.185944	Wan Chai	灣仔區	湾仔区		open	2.1	Height Limit: \nElectric Vehicle Charging Serv...
3	tdc2p1	Car Park 1 (Hourly)	一號停車場(時租)	一号停车场(时租)	CHEONG SHUN ROAD, Islands District, New Terri...	新界 離島區 暢順路 號	新界 离岛区 畅顺路 号	22.313223	113.936656	Islands	離島區	离岛区	2183 4630	open	0.0	Height Limit: \n

Data Collection (Con't)

- Parking Vacancy Data of Participating Car Parks (We know that park_id: tdc1p1 is Lee Garden One Car Park)

type	category	vacancy_type	vacancy	lastupdate	time of api
P	HOURLY	A	129	2021-03-09 8:57:03	2021-03-09 09:00:00
P	HOURLY	A	122	2021-03-09 9:11:03	2021-03-09 09:15:00
P	HOURLY	A	115	2021-03-09 9:27:03	2021-03-09 09:30:00
P	HOURLY	A	116	2021-03-09 9:41:03	2021-03-09 09:45:00
P	HOURLY	A	111	2021-03-09 9:57:03	2021-03-09 10:00:00
P	HOURLY	A	109	2021-03-09 10:11:03	2021-03-09 10:15:00
P	HOURLY	A	101	2021-03-09 10:27:03	2021-03-09 10:30:00
P	HOURLY	A	91	2021-03-09 10:41:06	2021-03-09 10:45:00
P	HOURLY	A	79	2021-03-09 10:57:03	2021-03-09 11:00:00
P	HOURLY	A	74	2021-03-09 11:11:03	2021-03-09 11:15:00
P	HOURLY	A	68	2021-03-09 11:27:03	2021-03-09 11:30:00
P	HOURLY	A	63	2021-03-09 11:41:03	2021-03-09 11:45:00
P	HOURLY	A	50	2021-03-09 11:57:03	2021-03-09 12:00:00
P	HOURLY	A	33	2021-03-09 12:11:03	2021-03-09 12:15:00
P	HOURLY	A	19	2021-03-09 12:27:03	2021-03-09 12:30:00
P	HOURLY	A	13	2021-03-09 12:41:03	2021-03-09 12:45:00
P	HOURLY	A	1	2021-03-09 12:57:03	2021-03-09 13:00:00
P	HOURLY	A	0	2021-03-09 13:13:04	2021-03-09 13:15:00
P	HOURLY	A	0	2021-03-09 13:27:03	2021-03-09 13:30:00
P	HOURLY	A	0	2021-03-09 13:41:03	2021-03-09 13:45:00
P	HOURLY	A	0	2021-03-09 13:57:03	2021-03-09 14:00:00

Predictive Model Development

- Findings from EDA after loading the APIs
- Feature Engineering processes and rationale
- Model Selection and model results
- Model limitations

Exploratory Data Analysis

What does the Data Structure look like?

	type	category	vacancy_type	vacancy	lastupdate	time of api
0	P	HOURLY	A	129	2021-03-09 08:57:03	2021-03-09 09:00:00+00:00
1	P	HOURLY	A	122	2021-03-09 09:11:03	2021-03-09 09:15:00+00:00
2	P	HOURLY	A	115	2021-03-09 09:27:03	2021-03-09 09:30:00+00:00
3	P	HOURLY	A	116	2021-03-09 09:41:03	2021-03-09 09:45:00+00:00
4	P	HOURLY	A	111	2021-03-09 09:57:03	2021-03-09 10:00:00+00:00

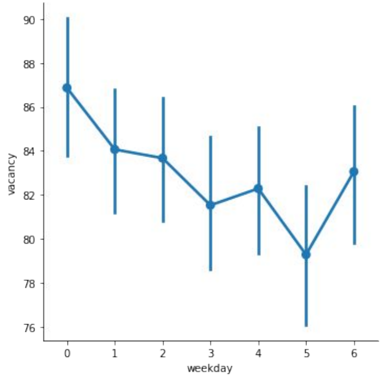
- While downloading the APIs, store an additional column of when the API is created ('time of api')
- Our data only pertains to **Lee Garden One Car Park**
- Vacancy is **what we want to predict**
- We **hypothesis that** vacancy is largely explained by variations in time

Given the data we have, we need to see **how each time variable (hour, minute, month) affect the vacancy** of Lee Garden One Park.

Exploratory Data Analysis

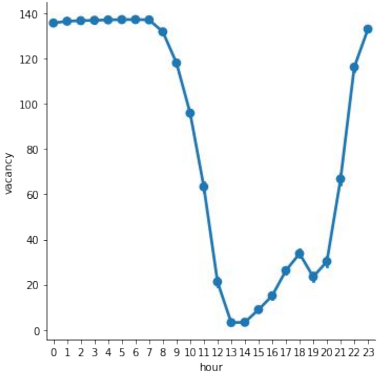
Vacancy breakdown by weekday or hour, graphs generated after grouping vacancy by datetime.weekday() or hour.

Vacancy and weekday



- Variations in weekdays
- *Lower vacancies on weekends, vice versa*

Vacancy and hour of the day

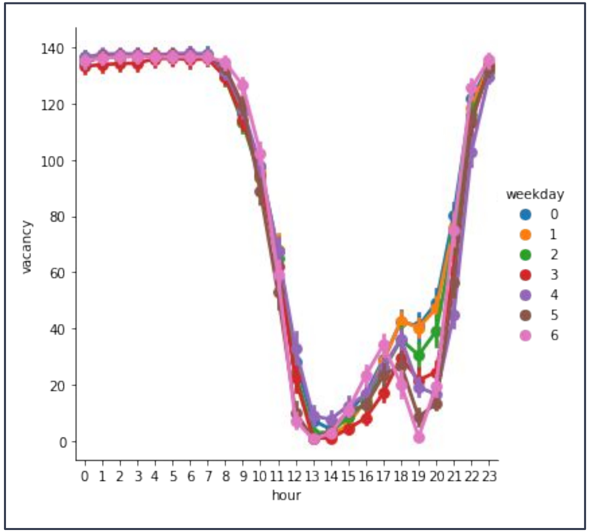


- Variations in hour of the day
- *Lower vacancies at dinner hours, max vacancies at mid night hours*

Weekdays or hours appear to be **explanatory of the variations in vacancies.**

Exploratory Data Analysis

Vacancy breakdown by weekday <u>and</u> hour, graphs plotted with 'hue = weekday'

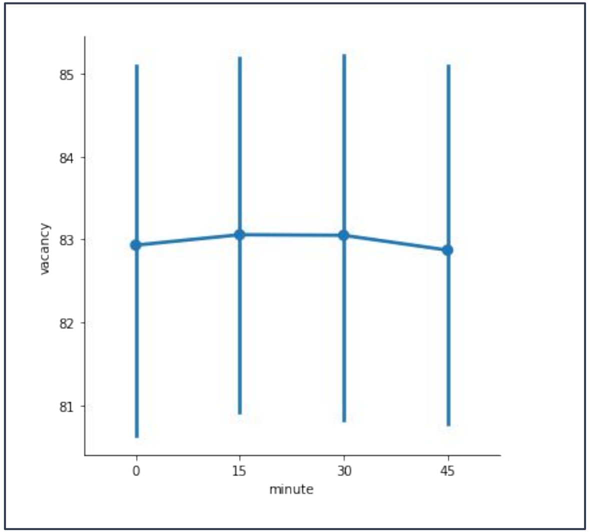


- When considering both weekday and hour as factors:
- *Minimal variations from midnight to morning*
- *Noticeable variations during dinner hours (18:00 - 20:00)*

Weekdays and certain hours in the day appear to be **explanatory of the variations in vacancies.**

Exploratory Data Analysis

Vacancy breakdown by 15-minute interval within an hour

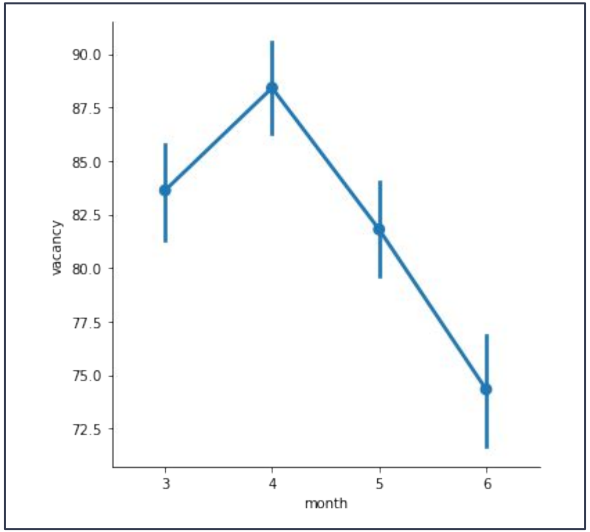


- Not much variation within any 15-minute interval
- Makes sense since Lee Garden One Park's an **hourly parking facility**
- Variation largely driven by difference in hour

15-minute intervals by itself not an explanatory variable for **amount of vacancies.**

Exploratory Data Analysis

Vacancy breakdown by month



- **Ambiguous** significance given government API **only goes back to March**

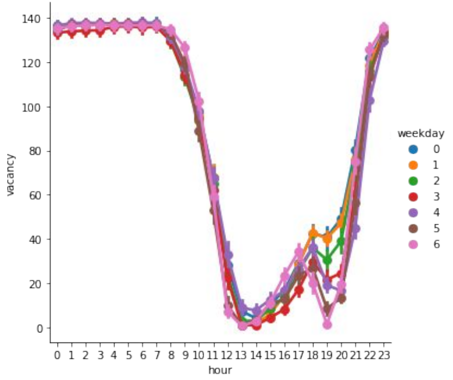
- **Inconclusive** whether it is explanatory or not

There is not enough months worth of data to determine whether month is a significant factor or not.

Feature Engineering and Selection

Given results of our exploratory data analysis, we think hour and weekday are the most explanatory features.

EDA Results



- Apparent difference between weekdays-hours when it comes to vacancies

Feature Engineering

Saving hour and weekday into new columns

```
df_raw['hour'] = df_raw['time of api'].apply(lambda x:x.hour)

df_raw['weekday'] = df_raw['time of api'].apply(lambda x:x.weekday())
```

Feature Selection

Making the above our X variable and vacancy our y variable after exporting to a new csv file

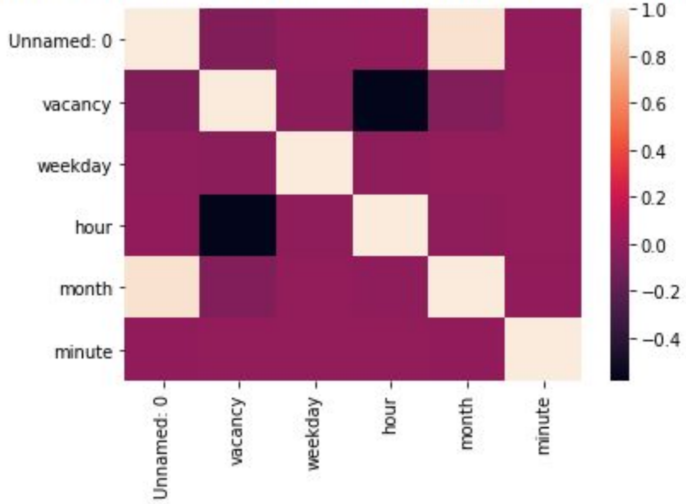
```
#EDA shows these 2 factors seem to be most explanatory


X = df_cleaned[['weekday','hour']]
y = df_cleaned['vacancy']
```

Feature Engineering and Selection

```
sns.heatmap(Lee_Garden.corr())
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe43eb66450>




```
Lee_Garden.corr()
```

	Unnamed: 0	vacancy	weekday	hour	month	minute
Unnamed: 0	1.000000	-0.067122	-0.009557	-0.003387	0.961720	-0.003319
vacancy	-0.067122	1.000000	-0.028712	-0.581245	-0.059549	-0.000379
weekday	-0.009557	-0.028712	1.000000	-0.007323	0.001499	0.001759
hour	-0.003387	-0.581245	-0.007323	1.000000	-0.009702	0.001114
month	0.961720	-0.059549	0.001499	-0.009702	1.000000	-0.003867
minute	-0.003319	-0.000379	0.001759	0.001114	-0.003867	1.000000

Feature Engineering and Selection

1. Since we know that in the “month” column, we only included 3 months for model training and prediction, and actually there is 12 month in every year. So it is not precise to do model training and prediction using “month” feature

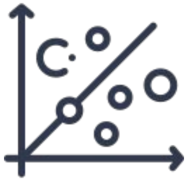
2. We find out the correlation between “vacancy” and “minute” is pretty low. So, we drop out this feature in the model.

3. Finally, we choose “weekday” and “hour” as X variables and “vacancy” as y variable

Model Selection

Given the output aims to predict the ***amount of vacancies***, we considered the following models which are suitable for **continuous value predictions.** Our evaluation metrics are **root mean squared error.**

<u>Linear Regression</u>



- *Linear approach to modelling the relationship between a scalar response and one or more explanatory variables*
- *Great for predicting continuous, numerical values*

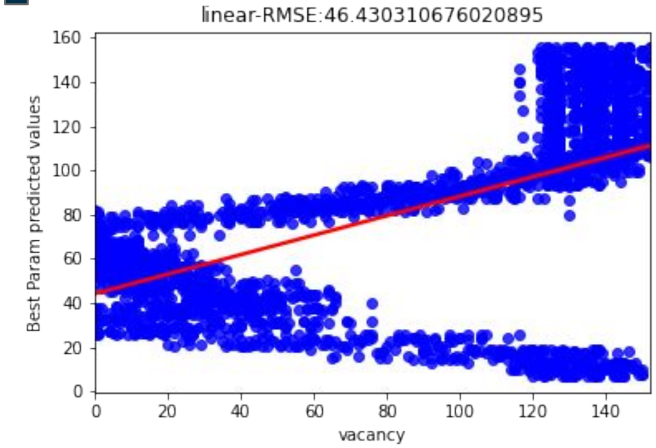
<u>Random Forest Regressor</u>

- *Ensemble approach; combines the predictions from multiple decision trees*
- *Merge different tree predictions together to get a more **accurate** and **stable** prediction rather than relying on individual decision trees.*

We will test the 2 model respectively and pick the best performing ones for our deployment.

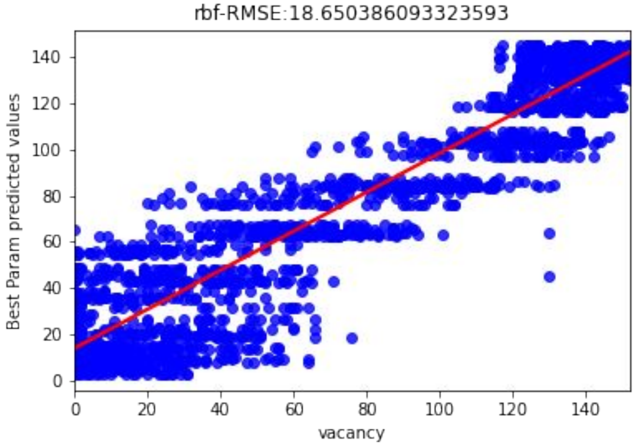
Model Selection – *Regression*

Linear Regression produces mediocre results since our data **does not have a linear relationships**



Use SVR instead

Testing which kernel is the best



- High RMSE from error

- Kernel = 'rbf' yield best results in minimizing error
- Poly RMSE = 56.67
- Sigmoid RMSE = 326.72

Instead of using Linear Regression, rbf regressor is adopted instead. (vacancy mean = 82.97)

Model Selection – SVM(*kernel = 'rbf'*)

After selecting rbf regression as our first regression model, we used **GridSearchCV()** to find best performing **hyperparameters.**

Parameters Grid

```
'kernel':['rbf'],
'C':[0.01,0.1,1,10,100],
'gamma':[0.01,0.1,1,10,100],
'epsilon':[0,0.5,2,5,10]
```

GridSearchCV

Results

Best Parameters

```
{'C': 100, 'epsilon': 5,
'gamma': 0.1, 'kernel': 'rbf'}
```

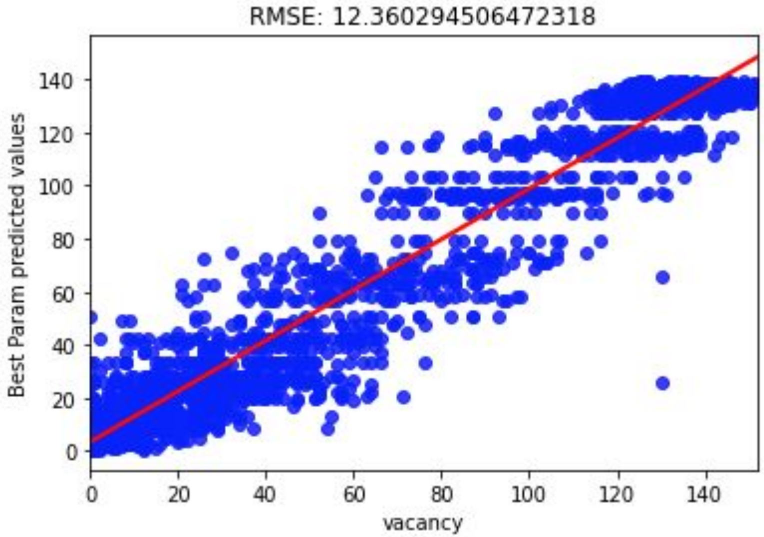
Best Scores

```
RMSE: 12.289104634233755
```

For regression model, we will use rbf regressor with the aforementioned best parameters.

Model Selection – Random Forest

Without any hyperparameter tuning, Random Forest Regressor yields the following results



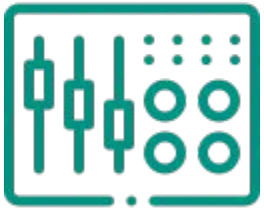
- RMSE = 12.36
- Relatively close to our best tuned SVR model

We will move on to perform hyperparameter tuning to see if it can outperform our SVR model.

Model Selection - GridSearchCV(estimator = rfr)

We used **GridSearchCV()** to find best performing **hyperparameters for our random forest regressor.**

Parameters Grid



```
'n_estimators': [200,400,600,800],
'criterion': ['mse','mae'],
'max_depth': [10,20,40,80],
'min_samples_leaf': [1, 2, 4],
'min_samples_split': [2, 5, 10]
```

GridSearchCV

Results

Best Parameters

```
{'criterion': 'mse',
 'max_depth': 10,
 'min_samples_leaf': 4,
 'min_samples_split': 10,
 'n_estimators': 200}
```

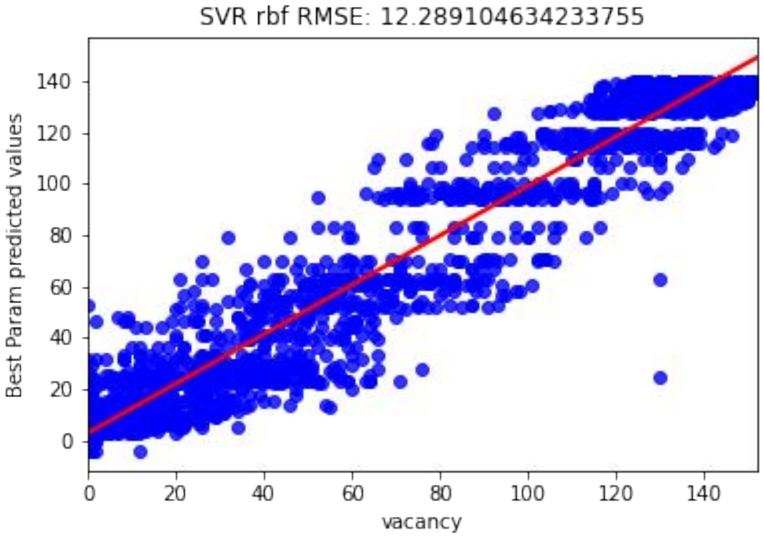
Best Scores

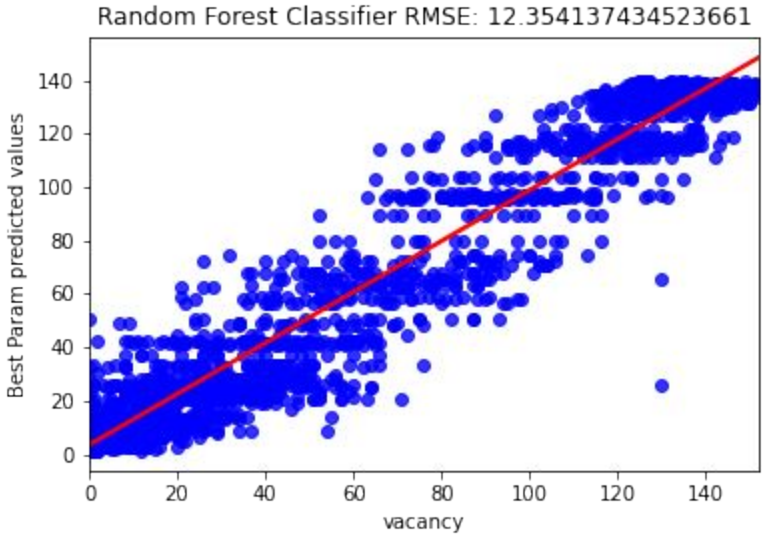
```
RMSE: 12.354137434523661
```

We will then compare the 2 model's results and pick the best performing one for deployment

Model Selection – RFR vs SVM

Comparing the 2 models RMSE results, we have:





Given SVR performs slightly better by our metrics, we will use it going forward. (vacancy mean = 82.97)

Model Prediction and Deployment

For example a user wants to predict the parking space available on **2021-08-01 14:15:00**

Defining Function

```python
from datetime import datetime

def user_prediction(date_string):
    #date_string = str(input('Enter date(yyyy-mm-dd hh:mm): '))
    date_datetime = datetime.strptime(date_string, "%Y-%m-%d %H:%M")

    #extracting datetime variables for prediction model
    input_dict = {}
    input_dict['weekday'] = date_datetime.weekday()
    input_dict['hour'] = date_datetime.hour

    for_pred = pd.DataFrame(input_dict,index=[0])

    return print(f'On {date_string}, the amount of vacany predicted in Lee Garden One Park is {int(best_param_svr.predict(for_pred)[0])}')
```

- A function for converting user input string to Datetime
- Extracting weekday and hour for our prediction model
- Output prediction results

User Input

```python
date_string = str(input('Enter date(yyyy-mm-dd hh:mm): '))
user_prediction(date_string)
```

```
2021-08-01 14:15
Enter date(yyyy-mm-dd hh:mm):
```

- Python input function to take in user string
- User input date string

Output: 'On 2021-08-01 14:15, the amount of vacancy predicted in Lee Garden One Park is 4.'

Model limitations

Limited data

- Data only available **from 2021-03-09**
- We can only collect **three months of data**
- **No proxy for busy seasons and holidays**

Limited X variables

- From the dataframe, we only chose **“Weekday”** and **“hour”** as the X variables
- With **more data**, there may be **more explanatory features** that we can uncover

One Car Park Only

- So far we have only tackled the prediction problem **for 1 carpark only (Lee Garden)**
- In the following, we will talk about **improvements** and **problems with more car parks in a prediction model.**

The major limitation comes from lack of comprehensive and representative data.

What can be improved going forward?

- Following the API and increasing data counts as more data comes.
- Attempted prediction model with more than 1 car park.
- Problems with our attempt and why we opted for 1 car park version for now.

Advanced Model Attempt

Data Preparation

- Data manipulation of about **~9000 files with each ~350 park data**
- Total of **~3,000,000 vacancy data**
- Due to huge amount of data, have **tried to use C# language as the data manipulation tool**
- **Flattening ~9000 JSON files into dataframes**
- The flattened JSON files is uploaded to GitHub for future development efficiency

Feature Engineering

- **Categorize the 9000 * 350 data with different parks**
- For better calculation of parks within a district
- Use day of **weeks, hours, minutes** as parameter to predict
- After **limited amount of EDA, these three features is thought to be most relevant**

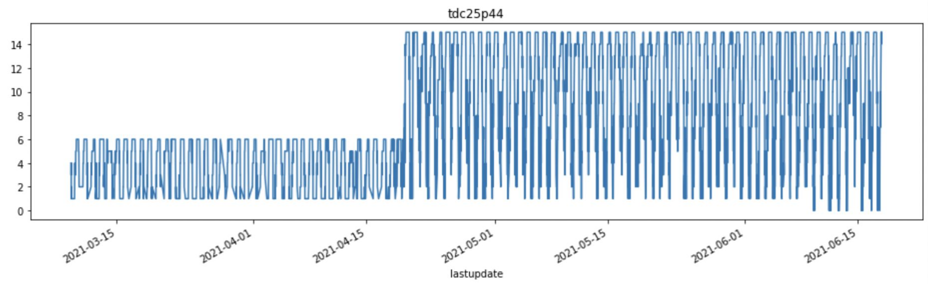
Prediction

- Use random forest regression to predict the vacancy of car parks in a desired district
- Tried to evaluate from three regression models: l**inear, decision tree, random forest**
- **Random Forest predictions turns out to be the best**

Improvements for >1 car park model

There are some **major problems with the data when more car parks are considered**, which require **significantly more time to tackle.**

Car Park Size Problem



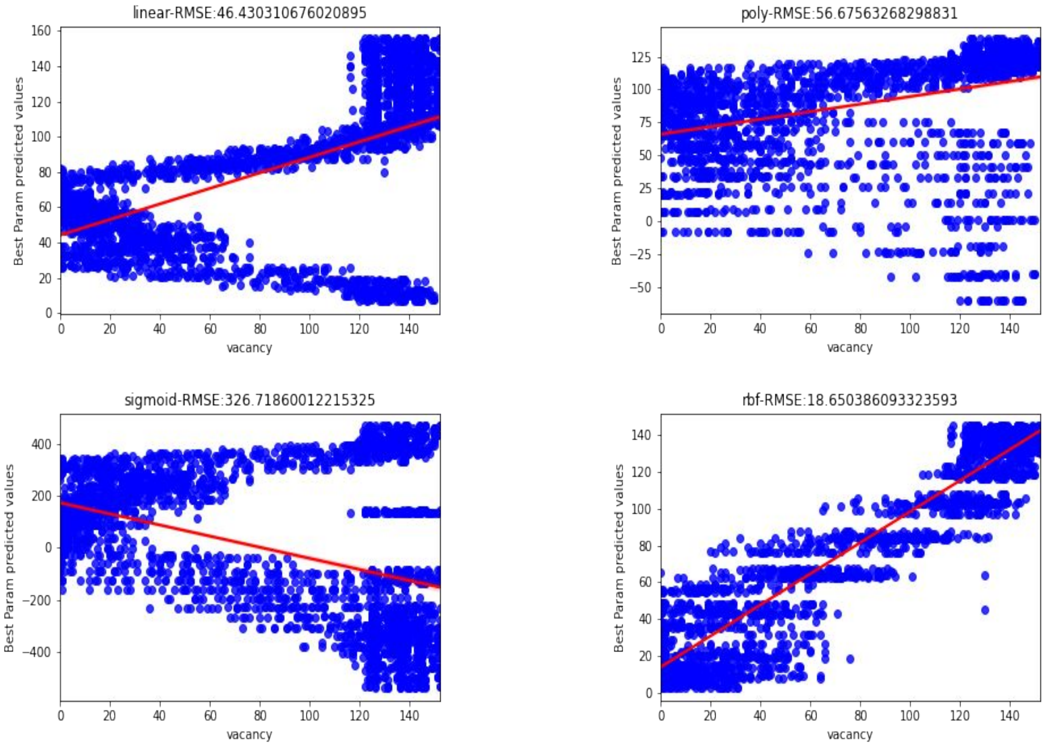
- Car park with id “tdc25p44”
- Might be having expansion around end of April this year.
- Sudden jump in vacancy

Service Category Problem

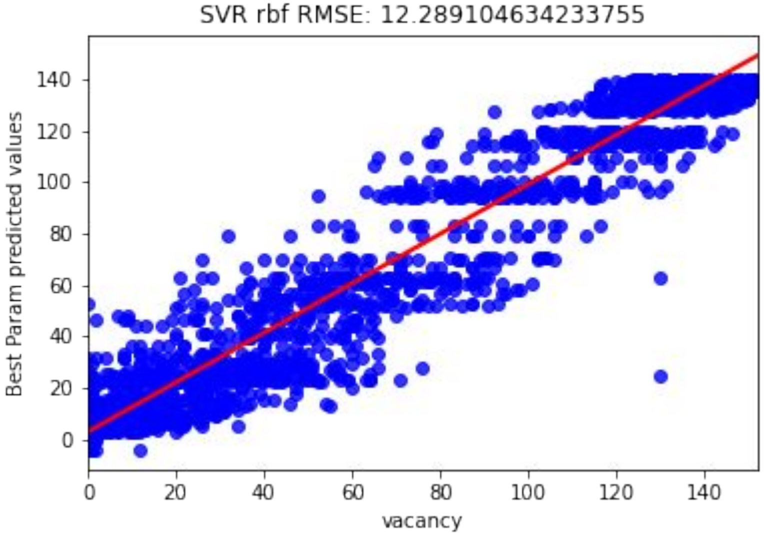
```
Service category: [HOURLY,
DAILY, MONTHLY]
```

- Could potentially explain variations in vacancy as well
- Need further data cleaning

Appendix 1 – kernel tuning results



Appendix 2 – best parameters SVM



Reference

Parking vacancy Data| DATA.GOV.HK:
https://data.gov.hk/tc-data/dataset/hk-td-tis_5-real-time-parking-vacancy-data