PM2.5 Prediction with Machine Learning

Presented by
Kyi Thin Nu (st124087)
Thongtong Eamsaard (st123300)
Group 7

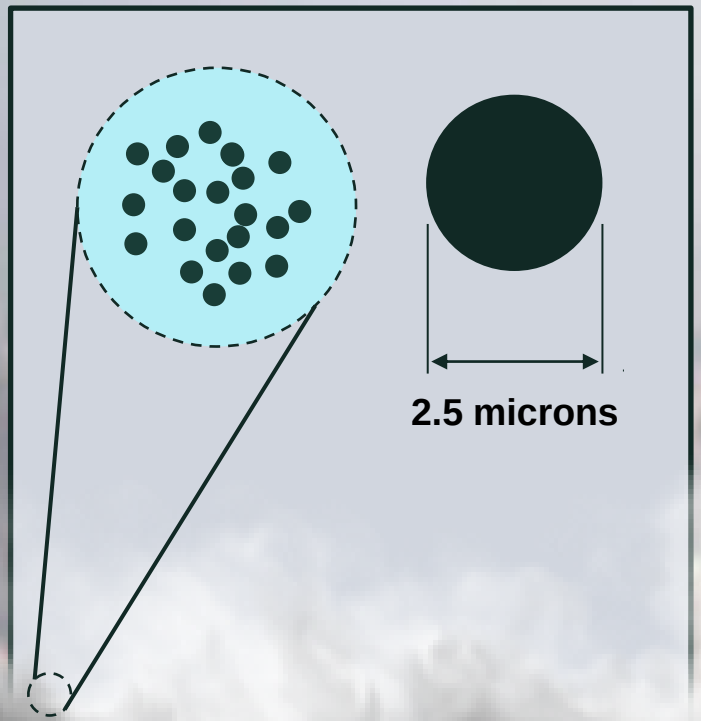
Image sources: https://www.bangkokpost.com/life/social-and-lifestyle/2578139/thailand-clean-air-network-needs-you

Table of Contents

01
02
03
04
05

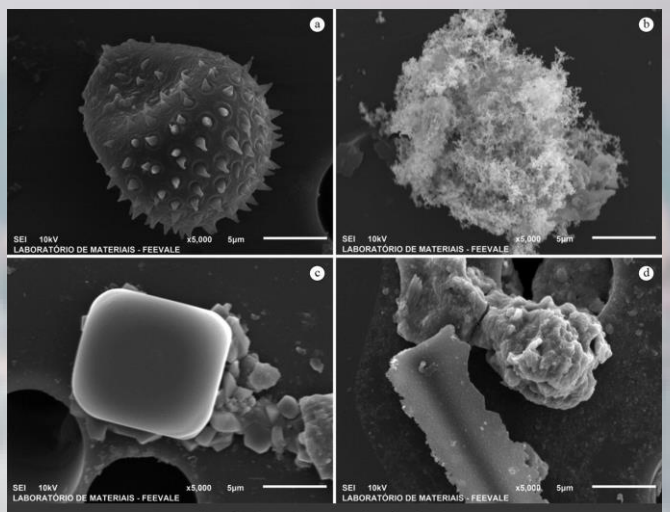
Introduction & Problem Statement

What is PM2.5 ?



Particulate Matter or PM2.5

refers to tiny airborne particles or droplets with a diameter of 2.5 micrometers or smaller.



(a) Particle of biological origin, identified in the PM2.5-10
(b) Soot particle, identified in the PM2.5-10 (b)
(c) Cubic-shaped particle, identified in the PM2.5-10
(d) Particle characteristic of soil resuspension and particle possibly originated from anthropic sources, both identified in the PM2.5-10

Image Source: https://www.devex.com/news/inside-thailand-s-tussle-with-toxic-smog-104836, https://journals.sagepub.com/doi/abs/10.1177/1420326X04059280?journalCode=ibeb

PM2.5 Perspective

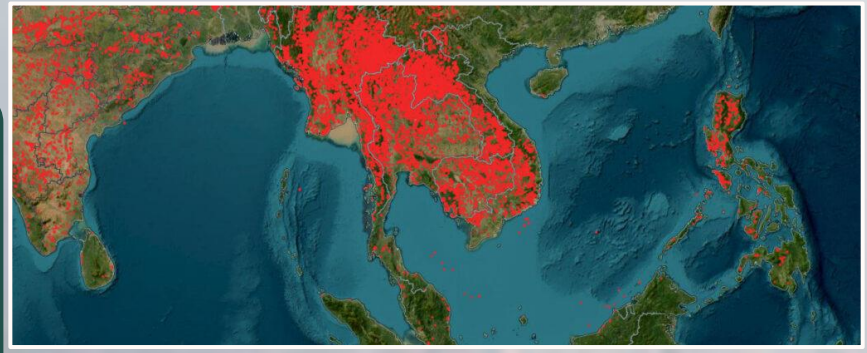
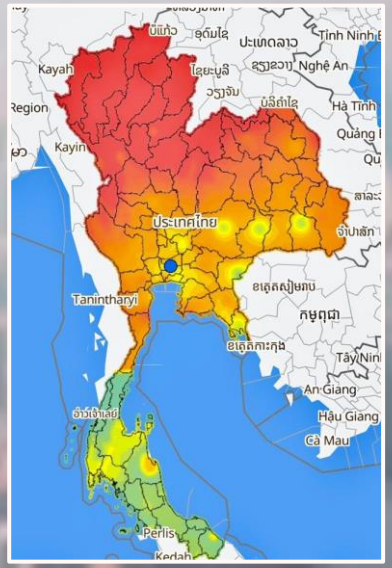


Image Source: https://bkktribune.com/the-hard-lessons-of-pm2-5-haze/

PM2.5: Global Perspective

- helps nations collaborate in addressing transboundary air pollution
- sharing information to mitigate the impact of airborne particles on a global scale.



PM2.5: Local Perspective

- helps individuals make informed decisions about outdoor activities
- it assists policymakers in implementing measures to improve air quality and protect public health.

Image Source: bkktribune.com/the-hard-lessons-of-pm2-5-haze

What is PM2.5 ?

01

02

03

04

05

Why PM2.5 is important ?

Protecting Public Health

Environmental Impact

Policy and Regulation

Awareness and Education

Image Source: https://home.maefahluang.org/pm2_5_neweducatation

Problem Statement

01

02

03

04

05

To predict the PM2.5 values based on given weather conditions and trends of the PM2.5, Correlations between weather parameters and PM2.5, Weather effects (wind speed and temperature) on PM2.5

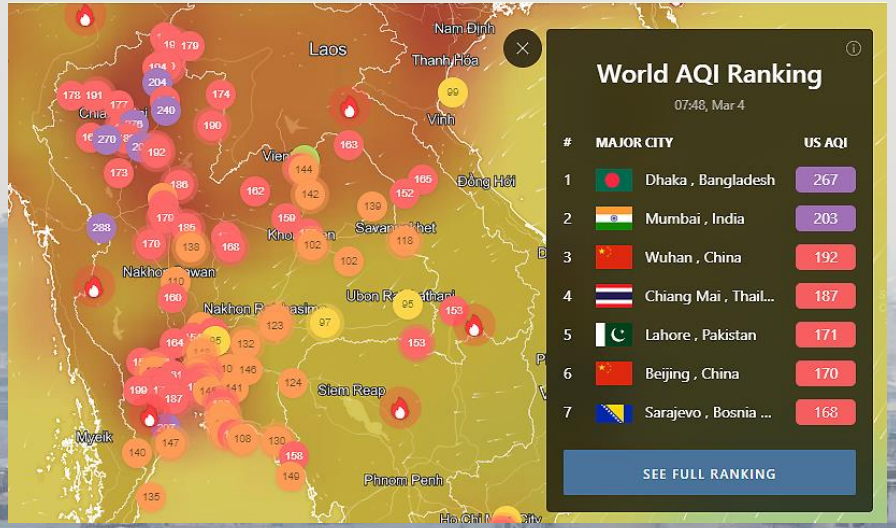
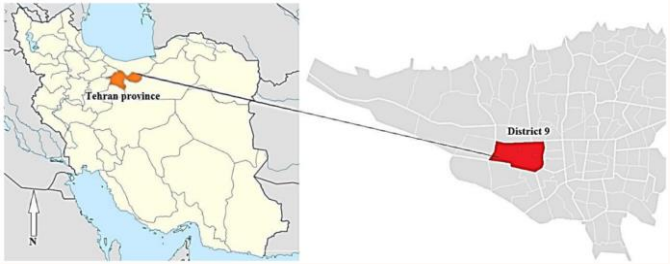


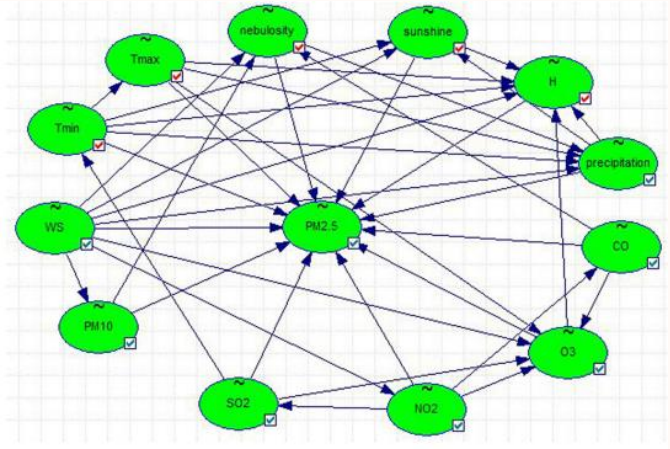
Image Source: https://stats.stackexchange.com/questions/198181/interpreting-temporal-trends-and-selecting-predictors-in-regression-models, https://www.bangkok-travel-ideas.com/weather-in-thailand.html, https://www.bangkokpost.com/thailand/general/2520020/solutions-sought-as-north-ravaged-by-toxic-pm2-5

01
02
03
04
05

Literature Review

Literature Review

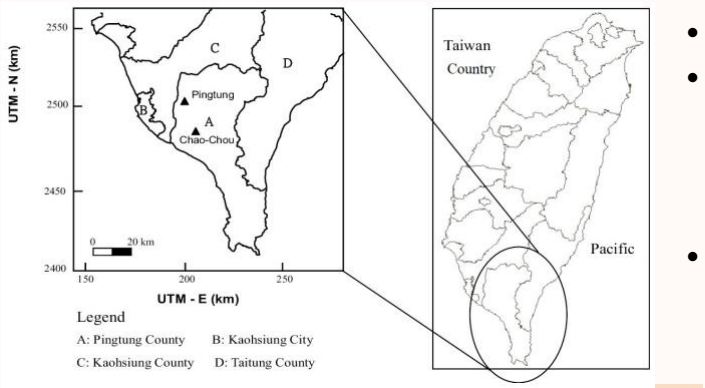




- proposed decision trees (DT), Bayesian Network (BN), and support vector machine (SVM). Using the data for over three periods,
- PM10, NO2, SO2, and O3 are critical factors for PM2.5
- Our work will include location, wind direction and wind speed.

Image Source: Mehdipour V., Stevenson D.S., Memarianfard M. et al, "Comparing different methods for statistical modeling of particulate matter in Tehran, Iran", Air Quality Atmosphere Health 11, pp.1155—1165 (2018), https://doi.org/10.1007/s11869-018-0615-z

Literature Review



- Taiwan Air Quality Monitoring data set.
- The model they used are random forest regressor (RFR), gradient boosting regressor GBR), k neighbors regressor (KNR), MLP regressor (MLPR), and decision tree regressor CART.
- To select the best model, they used cross-validation and determined that gradient boosting regressor model is better in forecasting air pollution in TAQMN data.

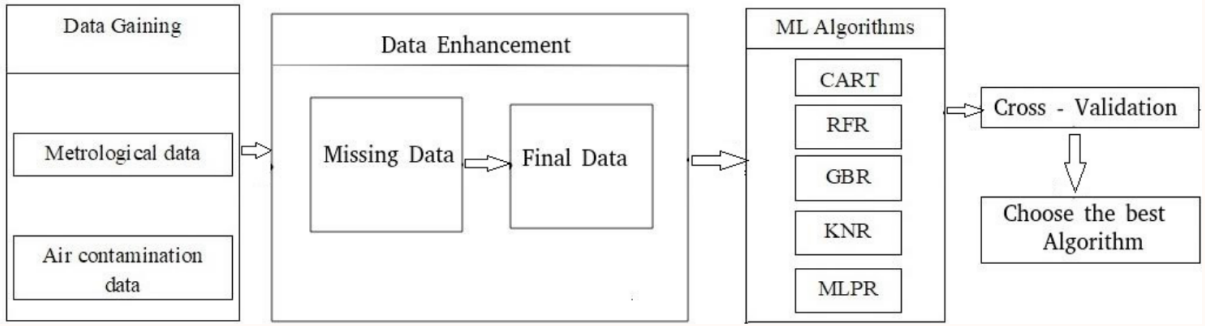


Image Source: Doreswamy, Harishkumar K. S., Yogesh K.M. and Ibrahim Gad. "Forecasting Air Pollution Particulate Matter (PM2.5) Using Machine Learning Regression Models". Procedia Computer Science. vol. 171. 2020. pp. 2057–2066. https://www.sciencedirect.com/science/article/pii/S1877050920312060

Methodology

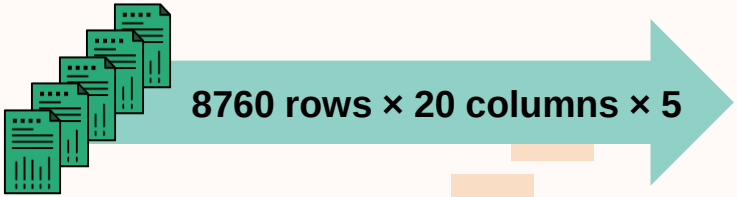
Image Source: https://www.google.co.th › maps

01
02
03
04
05
...

Drop 2nd header not necessary

Rename header + Add Province column because we have 5 stations

Merge all 5 stations -> single DataFrame



	YYMMDD	HR	CO	NO	NO2	NOX	Windspeed	Winddir	Temp	O3	PM10	PM2.5	Province	District	Relhum	Pressure	Rain	SO2	Globrad	Netrad
0	190101	100	1.6	1.0	31.0	32.0	0.1	86.0	22.3	12.0	46	28	Bangkok	Bang Khun Thian	NaN	NaN	NaN	NaN	NaN	NaN
1	190101	200	1.5	1.0	28.0	29.0	0.5	52.0	20.0	15.0	57	28	Bangkok	Bang Khun Thian	NaN	NaN	NaN	NaN	NaN	NaN
2	190101	300	NaN	NaN	NaN	NaN	0.0	28.0	23.4	NaN	52	28	Bangkok	Bang Khun Thian	NaN	NaN	NaN	NaN	NaN	NaN
3	190101	400	1.8	13.0	38.0	52.0	0.1	4.0	23.1	1.0	61	29	Bangkok	Bang Khun Thian	NaN	NaN	NaN	NaN	NaN	NaN
4	190101	500	1.5	5.0	37.0	42.0	0.1	3.0	22.6	1.0	54	32	Bangkok	Bang Khun Thian	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
43795	191231	2000	1.9	138.0	45.0	182.0	0.3	310.0	31.3	4.0	41	27	Bangkok	Din Daeng	48.0	762.0	0.0	NaN	NaN	NaN
43796	191231	2100	2.4	171.0	48.0	219.0	0.2	332.0	31.3	3.0	51	29	Bangkok	Din Daeng	48.0	763.0	0.0	NaN	NaN	NaN
43797	191231	2200	2.4	175.0	52.0	227.0	0.2	325.0	31.0	3.0	58	30	Bangkok	Din Daeng	49.0	763.0	0.0	NaN	NaN	NaN
43798	191231	2300	2.7	207.0	56.0	262.0	0.2	315.0	30.7	2.0	68	32	Bangkok	Din Daeng	51.0	763.0	0.0	NaN	NaN	NaN
43799	191231	2400	2.4	154.0	52.0	206.0	0.4	358.0	30.5	2.0	70	38	Bangkok	Din Daeng	51.0	764.0	0.0	NaN	NaN	NaN

43800 rows × 20 columns

43800 rows × 20 columns

Datetime formatting

(and drop YYMMDD and HR)

	YYMMDD	HR
0	190101	100
1	190101	200
2	190101	300
3	190101	400
4	190101	500

Timestamp
2019-01-01 01:00:00
2019-01-01 02:00:00
2019-01-01 03:00:00
2019-01-01 04:00:00
2019-01-01 05:00:00

Image Source:

01 02 03 04 05 ...

Reorder the columns

making it more readable and more manageable (move target variable to the end)

	CO	NO	NO2	NOX	Windspeed	Winddir	Temp	O3	PM10	PM2.5	Province	District	Relhum	Pressure	Rain	SO2	Globrad	Netrad	Timestamp
0	1.6	1.0	31.0	32.0	0.1	86.0	22.3	12.0	46	28	Bangkok	Bang Khun Thian	NaN	NaN	NaN	NaN	NaN	NaN	2019-01-01 01:00:00
1	1.5	1.0	28.0	29.0	0.5	52.0	20.0	15.0	57	28	Bangkok	Bang Khun Thian	NaN	NaN	NaN	NaN	NaN	NaN	2019-01-01 02:00:00
2	NaN	NaN	NaN	NaN	0.0	28.0	23.4	NaN	52	28	Bangkok	Bang Khun Thian	NaN	NaN	NaN	NaN	NaN	NaN	2019-01-01 03:00:00
3	1.8	13.0	38.0	52.0	0.1	4.0	23.1	1.0	61	29	Bangkok	Bang Khun Thian	NaN	NaN	NaN	NaN	NaN	NaN	2019-01-01 04:00:00
4	1.5	5.0	37.0	42.0	0.1	3.0	22.6	1.0	54	32	Bangkok	Bang Khun Thian	NaN	NaN	NaN	NaN	NaN	NaN	2019-01-01 05:00:00

	Timestamp	Province	District	CO	NO	NO2	NOX	SO2	O3	Windspeed	Winddir	Globrad	Netrad	Temp	Relhum	Rain	PM10	PM2.5
0	2019-01-01 01:00:00	Bangkok	Bang Khun Thian	1.6	1.0	31.0	32.0	NaN	12.0	0.1	86.0	NaN	NaN	22.3	NaN	NaN	46	28
1	2019-01-01 02:00:00	Bangkok	Bang Khun Thian	1.5	1.0	28.0	29.0	NaN	15.0	0.5	52.0	NaN	NaN	20.0	NaN	NaN	57	28
2	2019-01-01 03:00:00	Bangkok	Bang Khun Thian	NaN	NaN	NaN	NaN	NaN	NaN	0.0	28.0	NaN	NaN	23.4	NaN	NaN	52	28
3	2019-01-01 04:00:00	Bangkok	Bang Khun Thian	1.8	13.0	38.0	52.0	NaN	1.0	0.1	4.0	NaN	NaN	23.1	NaN	NaN	61	29
4	2019-01-01 05:00:00	Bangkok	Bang Khun Thian	1.5	5.0	37.0	42.0	NaN	1.0	0.1	3.0	NaN	NaN	22.6	NaN	NaN	54	32

predictor

target

Replace string values in numerical columns

- **'Calm' in Winddir, no direction in degree recorded. So, we replace it as NaN (because same result as sensor can't record data)**
- **'-' in PM10 & PM2.5, no value recorded. So, we replace it as NaN (because same result as sensor can't record data)**

```
, 'Calm'],
```

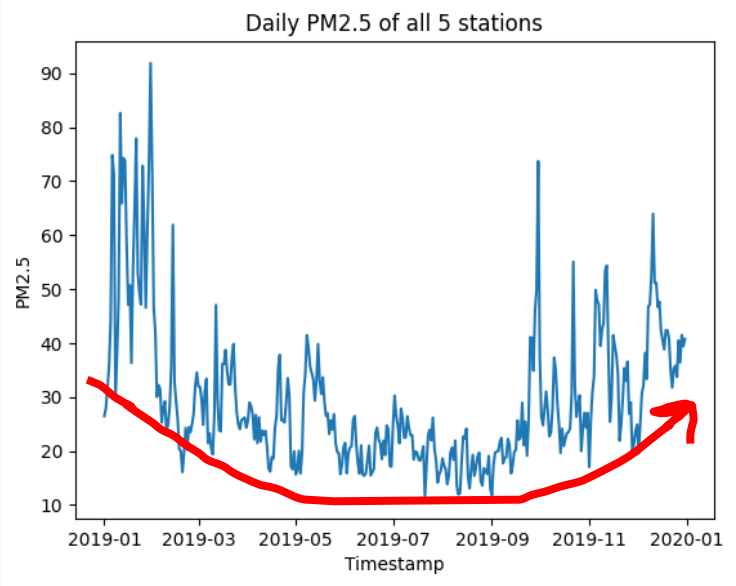
```
'-',
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43800 entries, 0 to 43799
Data columns (total 18 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Timestamp   43800 non-null  datetime64[ns]
 1   Province    43800 non-null  object
 2   District    43800 non-null  object
 3   CO          34833 non-null  float64
 4   NO          34833 non-null  float64
 5   NO2         34833 non-null  float64
 6   NOX         34833 non-null  float64
 7   SO2         34833 non-null  float64
 8   O3          34833 non-null  float64
 9   Windspeed   34833 non-null  float64
 10  Winddir     34791 non-null  object
 11  Globrad     0 non-null      float64
 12  Netrad      0 non-null      float64
 13  Temp        43633 non-null  float64
 14  Relhum      34883 non-null  float64
 15  Rain        34833 non-null  float64
 16  PM10        43800 non-null  object
 17  PM2.5       43800 non-null  object
dtypes: datetime64[ns](1), float64(12), object(5)
memory usage: 6.0+ MB
```

Image Source: https://www.mathsisfun.com/geometry/images/degrees-360.svg

PM2.5 (all stations)

- plotting to see the trend of average PM2.5 daily in all 5 stations.
- PM2.5 is quite high in the first two months. Then it drops dramatically at each its lowest in September. Finally, it starts rising again up until the end of December.



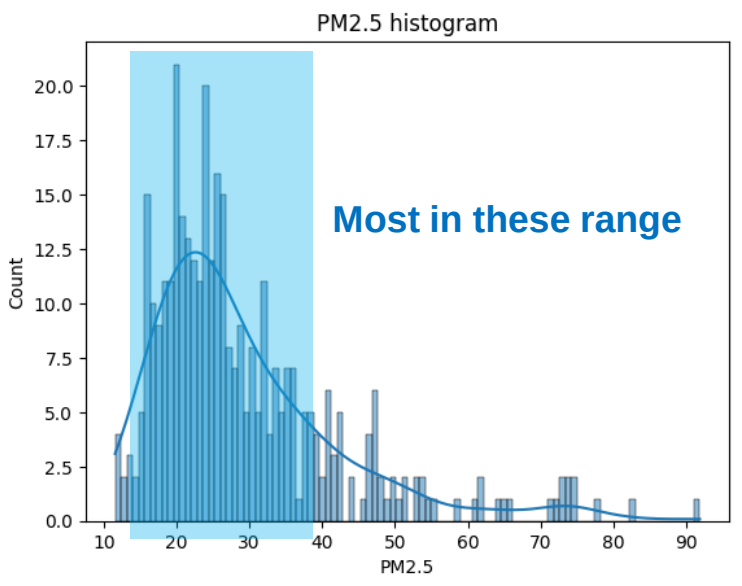


Image Source:

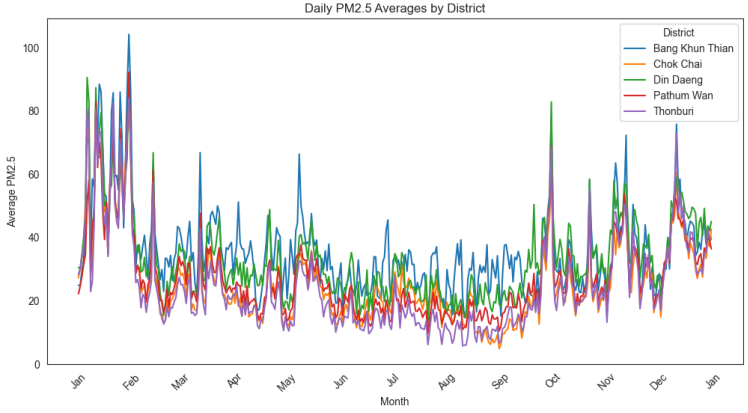
01

02

03

PM2.5 (individual)

- we compared PM2.5 from each station that are relatively in the same province so PM2.5 should be in the same trends.
- However, in kde plot, it shows that stations has 2 different distributions of the PM2.5 values. Bang Khun Thian is similar with Din Daeng. The other group is Chok Chai, Pathum Wan, and Thonburi.



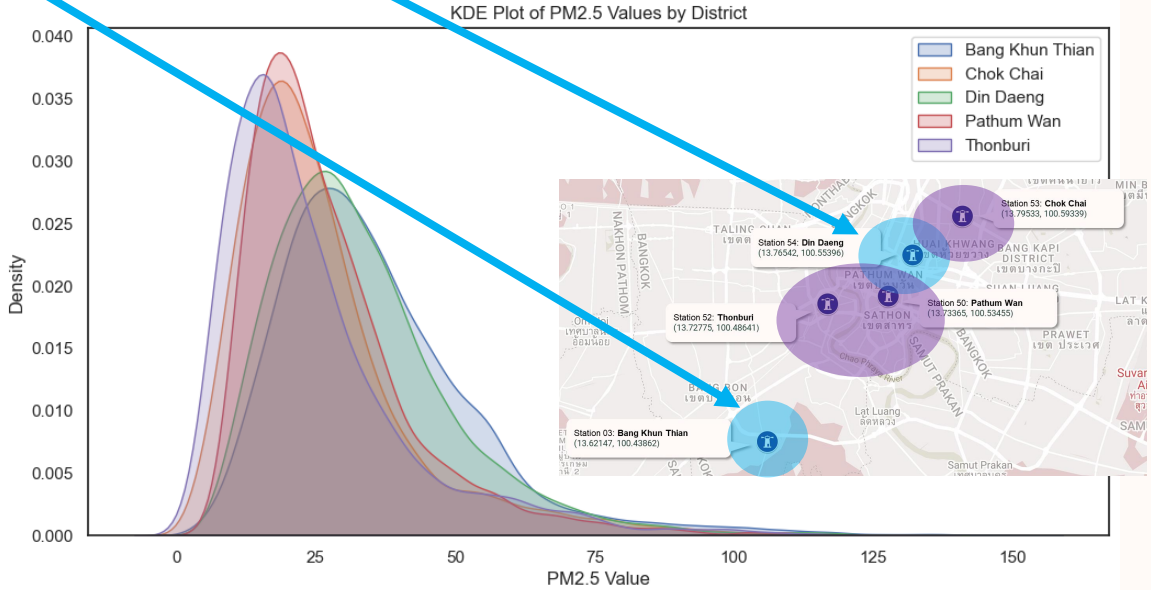
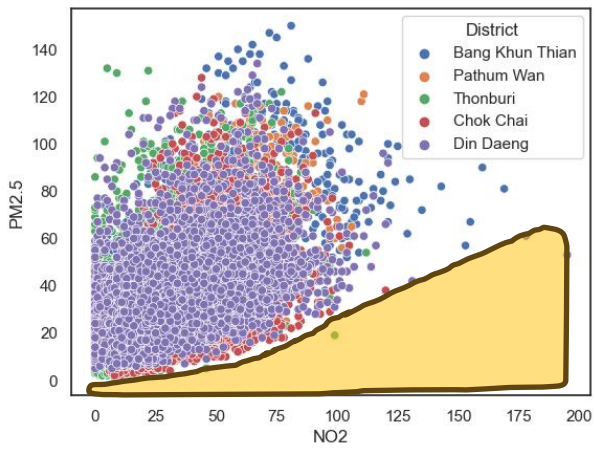
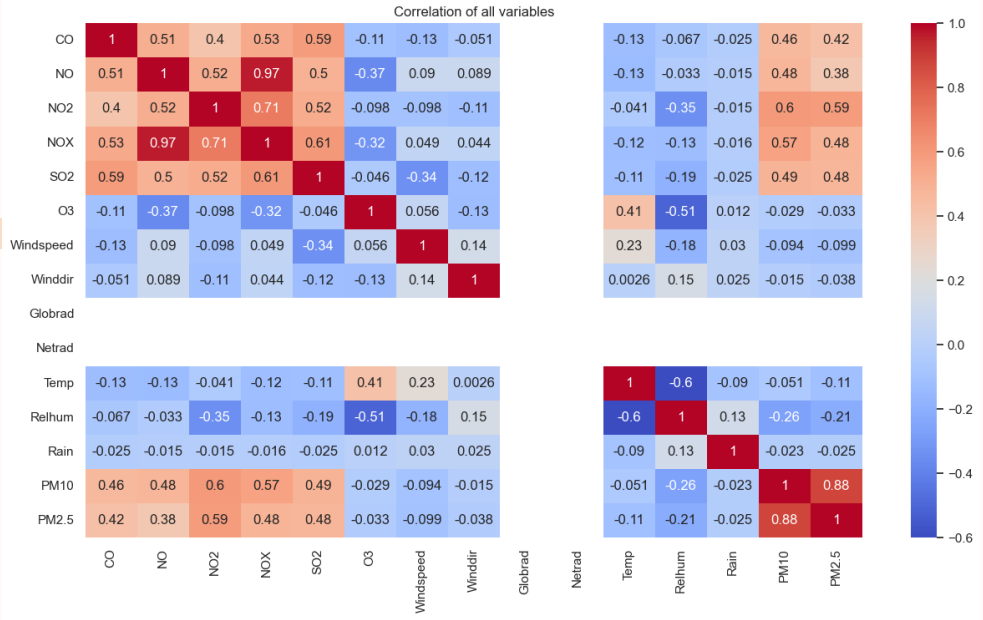


Image Source:



PM2.5 – NO2

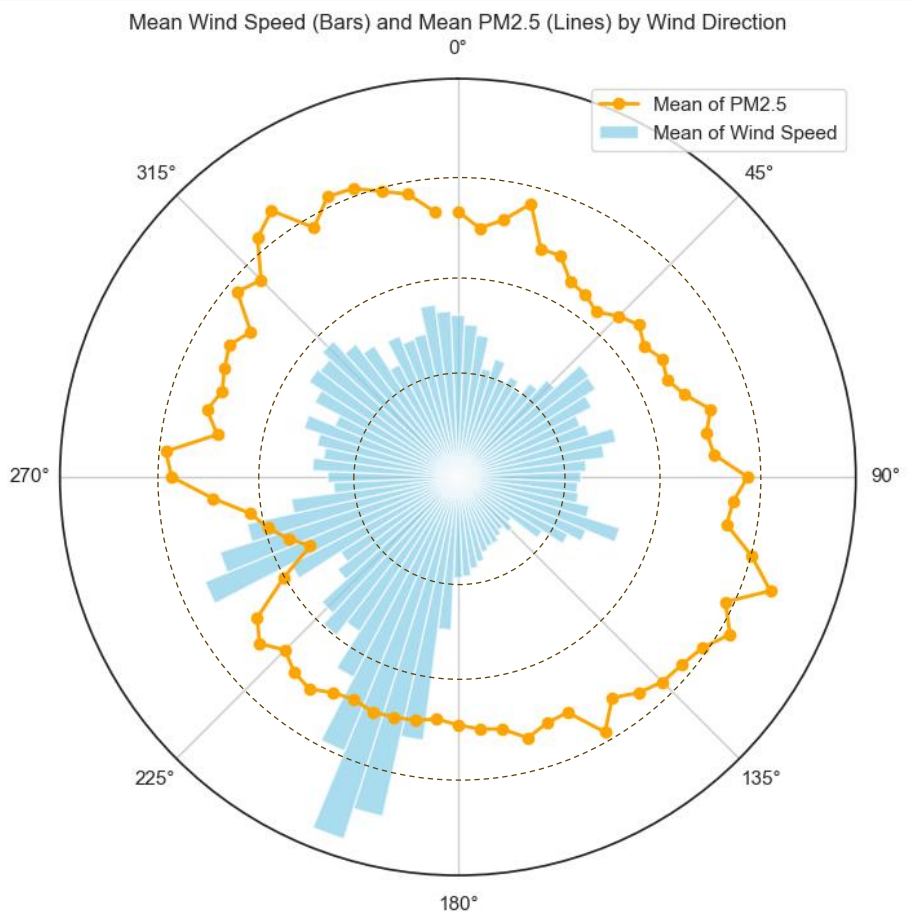
If NO2 increases, it tends to increase the minimum PM2.5 value boundary. This indicates that NO2 has some effects to PM2.5.



Correlation Heat Map

CO, NO, NO2, NOX, SO2, and PM10 have potential to be the predictor for PM2.5 as target variable.

Image Source:



Windspeed – Wind Direction – PM2.5

- Most wind coming from Southwest (SW) direction (180 – 270 degree)
- Wind direction and windspeed not directly main source of PM2.5. (Not the main carrier that bring PM2.5 to Bangkok.

Timestamp	0.000000
Province	0.000000
District	0.000000
CO	5.312785
NO	10.068493
NO2	10.068493
NOX	10.066210
SO2	81.114155
O3	52.506849
Windspeed	20.899543
Winddir	20.586758
Globrad	100.000000
Netrad	100.000000
Temp	0.381279
Relhum	20.358447
Rain	20.472603
PM10	0.867580
PM2.5	0.773973

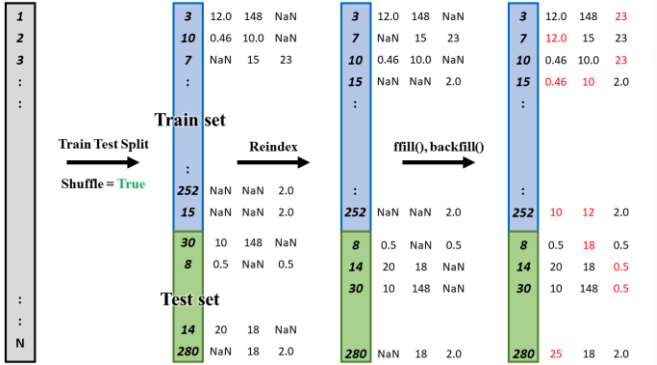
Remove columns

- **Globrad** and **Netrad 100 % are NaN (Not a Number)**
- **SO2 less than 20 % entries**
- **Province single unique value**, Bangkok.
- **Timestamp** will also be dropped **time-independent**.

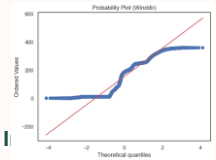
Train-test Split with 80 : 20 train-test ratio

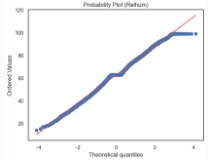
Imputation

Reindex then forward fill and backward fill (sensor data)



Scale and Encode into pipeline





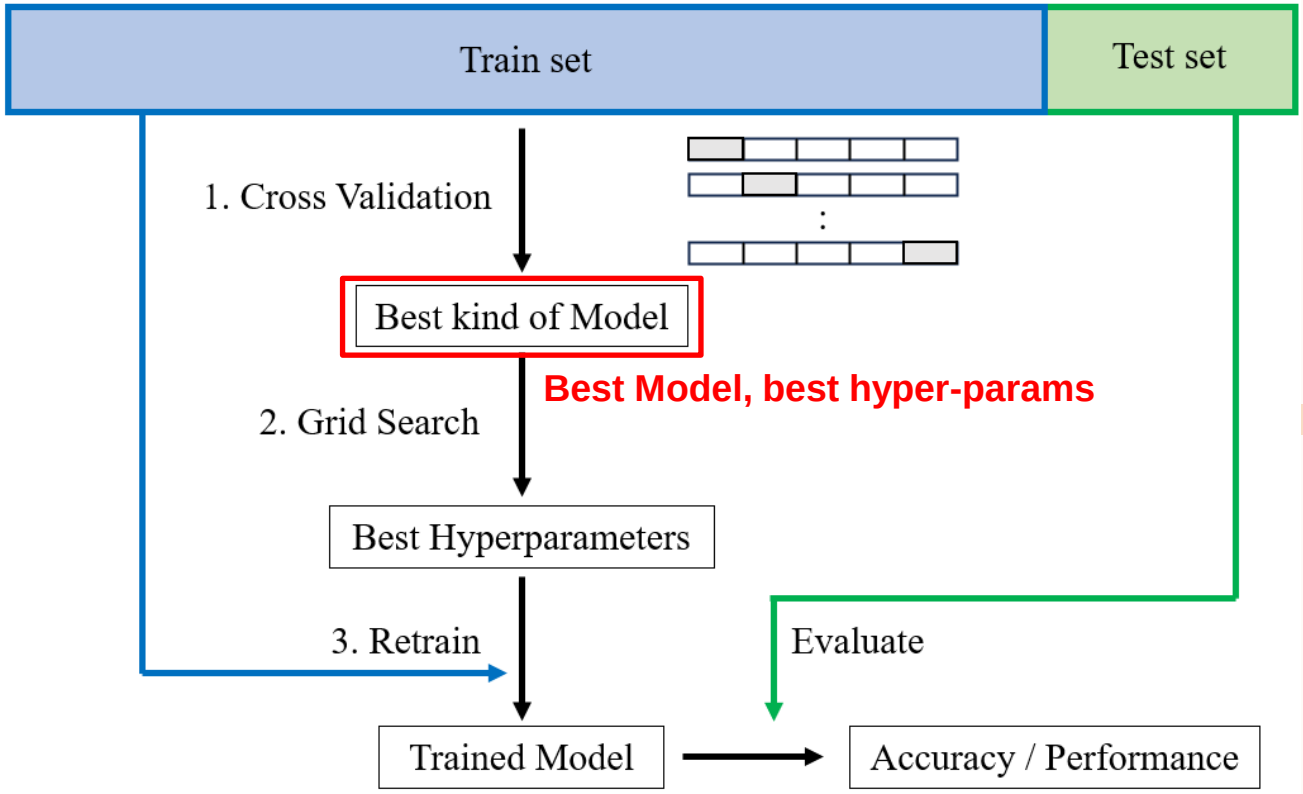
Column Name	Pipeline Step(s)		
	Standard Scale	*MinMax Scale*	*OneHot Encoder*
CO	✓		
NO		✓	
NO2		✓	
NOX		✓	
O3		✓	
Windspeed		✓	
Winddir		✓	
Temp	✓		
Relhum	✓		
Rain	✓		
PM10[a]		✓	
District			✓

[a]PM.10 will be dropped in other scenario.

Select Model → Cross Validation (K-Fold = 5) → Tune Hyper-params → GridSearchCV (K-Fold = 10) → Best Model, Best hyper-params

- *Linear Regression*
- *Ridge Regression*
- *Lasso Regression*
- *Gradient Boosting Regression*
- *Histogram-based Gradient Boosting Regression Tree*
- *Support Vector Regression (SVR)*
- *K Neighbor Regression*
- *Decision Tree Regression*
- *Random Forest Regression*
- *AdaBoost Regression*

Image Source: https://www.shutterstock.com/search/tightening-bolts,



Use R2 score

Image Source:

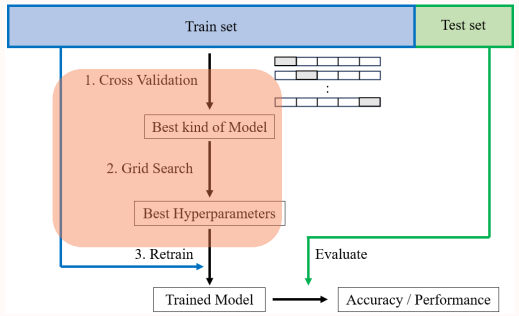
Create	Model
Pickle	Model
Create	Web Application
Deploy	AWS

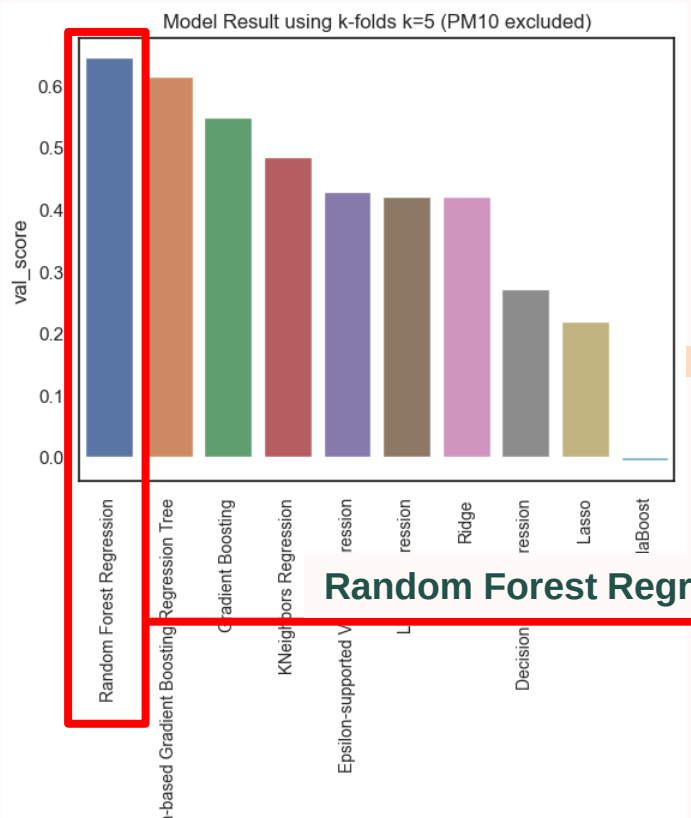
Image Source:

Results

Results

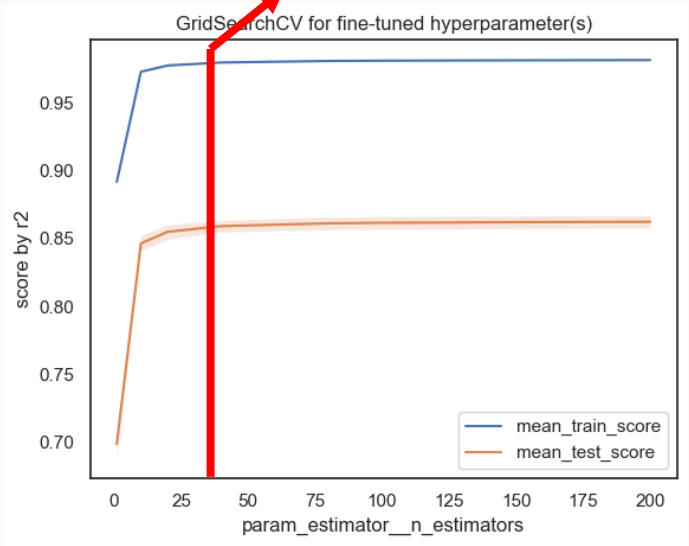
Select Model and Hyper-parameter(s) Tuning





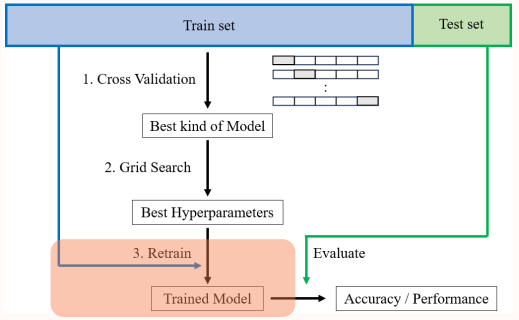
Model Selection

Hyper-parameter tuning (n_estimators = 40)



Random Forest Regression

Results



Create Pipeline

Pipeline

preprocessor: ColumnTransformer

- StandardScaler: CO Temp Relhum Rain
- MinMaxScaler: NO NO2 NOX O3 Windspeed Winddir
- OneHotEncoder: district
- remainder: passthrough

RandomForestRegressor

Image Source:

Results

Train and Evaluation

```
# create new pipeline to fit
random_forest = RandomForestRegressor(random_state=0,
                                      n_estimators=200)

deploy_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                                  ('model', random_forest)])
```

✓ 0.0s

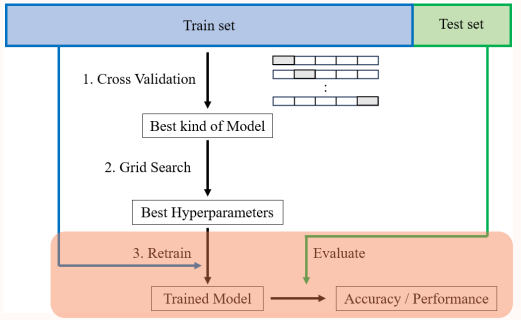
```
# training fit and transform (already include both fit and transform for train set)
# now this time, no seperation between train and val set like in GridSearchCV
deploy_pipeline.fit(X=X_train, y=y_train)
```

✓ 1m 29.3s

```
# No information leakage
yhat = deploy_pipeline.predict(X=X_test) # transform (no fit) and then predict
print(mean_squared_error(y_true=y_test, y_pred=yhat))
print(r2_score(y_true=y_test, y_pred=yhat))
```

```
162.97067793338206
0.4425220697072084
```

Image Source:



Training

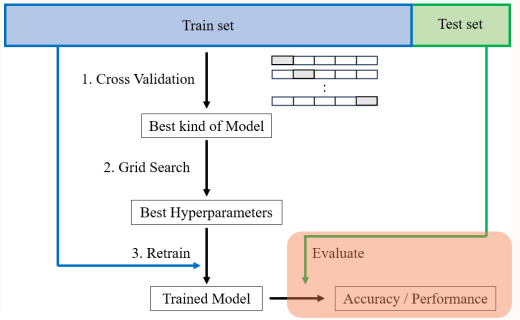
- *Val R2 score = 0.648*
(64.8 % that model can explained)

Evaluation

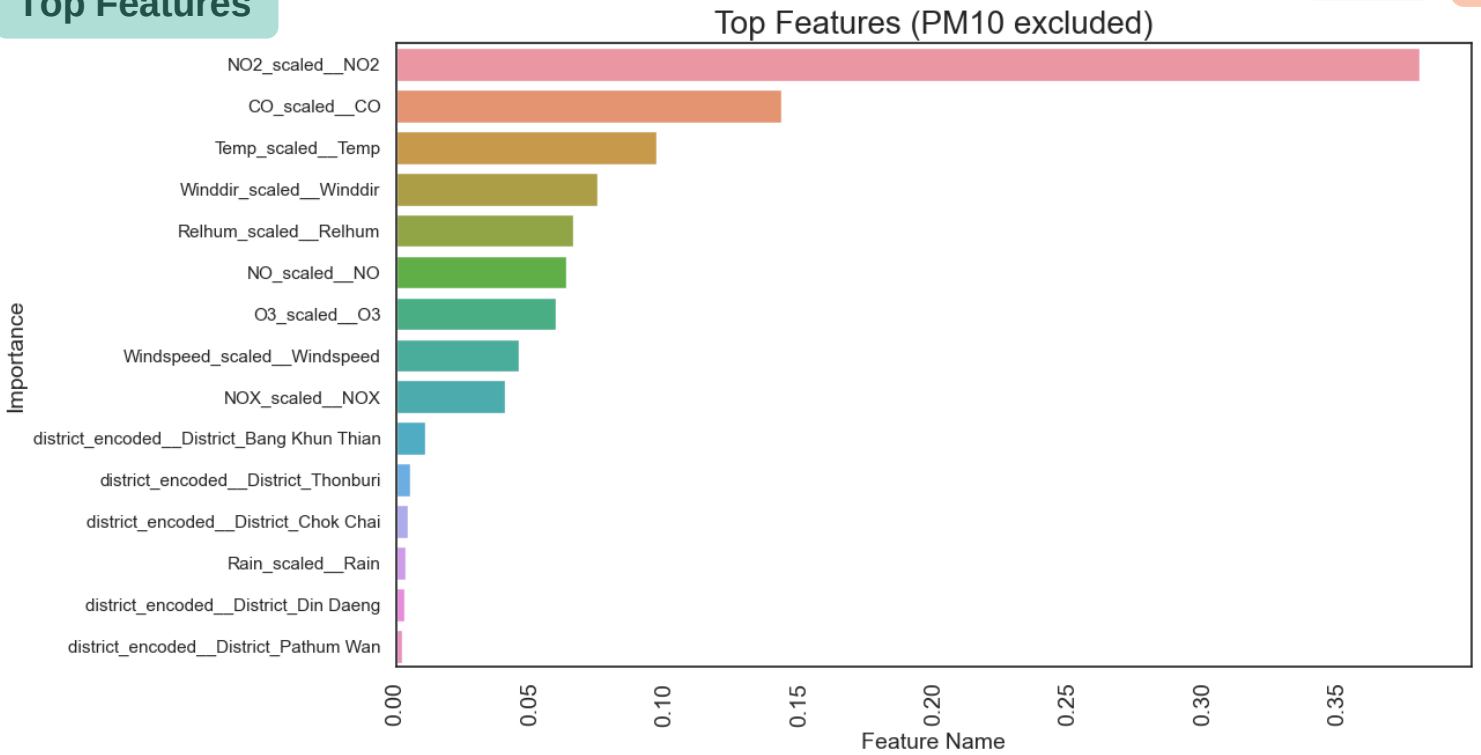
- *Test R2 score = 0.443*

Results

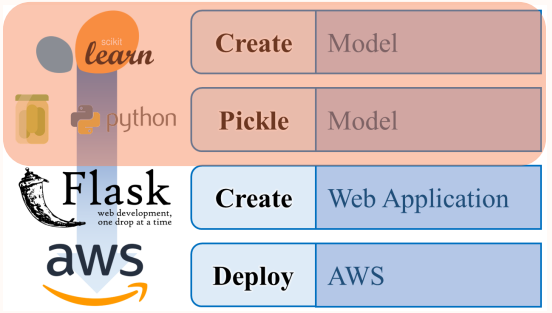
Evaluation



Top Features



Results



Create and Pickle the Model

```
import pickle

# save the model to disk
filename = 'model/pm25-prediction20231110_noPM10_v2_chronoffill.model'
pickle.dump(deploy_pipeline, open(filename, 'wb'))


# load the model from disk
loaded_model = pickle.load(open(filename, 'rb'))


predicted_pm25 = loaded_model.predict(sample)
predicted_pm25
```

```
array([38.235, 40.9  , 46.825, 41.015, 41.015, 40.4  ])
```

Sample DataFrame

	District	CO	NO	NO2	NOX	O3	Windspeed	Winddir	Temp	Relhum	Rain
0	Pathum Wan	24	25	34	10	25	20	120	32	20	0.2
1	Pathum Wan	30	30	30	20	5	2	0	30	1	0.8
2	Bang Khun Thian	30	30	30	20	5	2	0	30	1	0.8
3	Chok Chai	30	30	30	20	5	2	0	30	1	0.8
4	Thonburi	30	30	30	20	5	2	0	30	1	0.8
5	Din Daeng	30	30	30	20	5	2	0	30	1	0.8

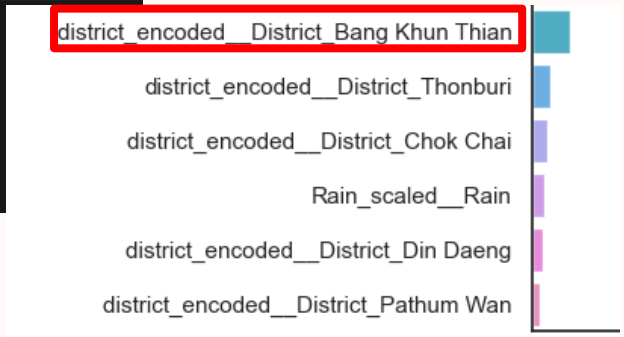


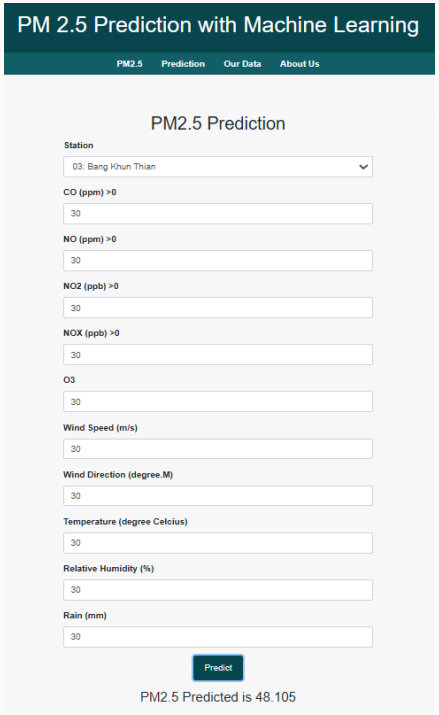
Image Source:

Results

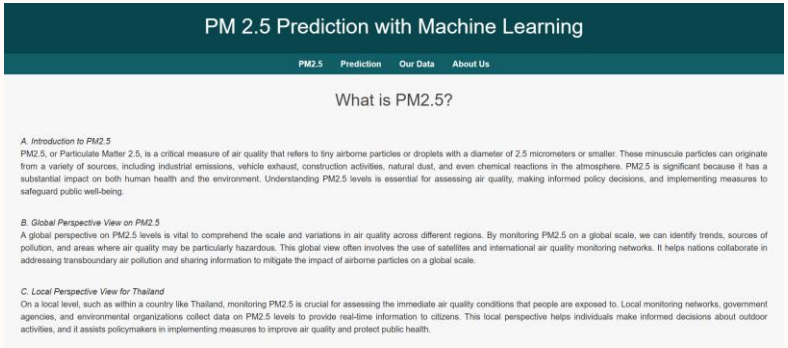
Web Application in local host

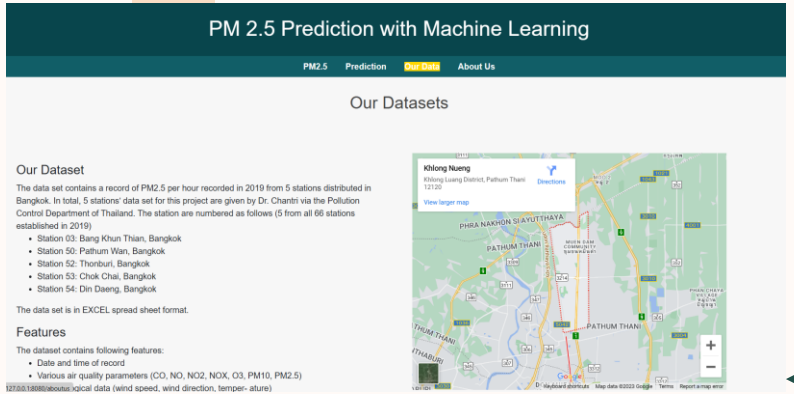
Create	Model
Pickle	Model
Create	Web Application
Deploy	AWS

Prediction



PM2.5 Info





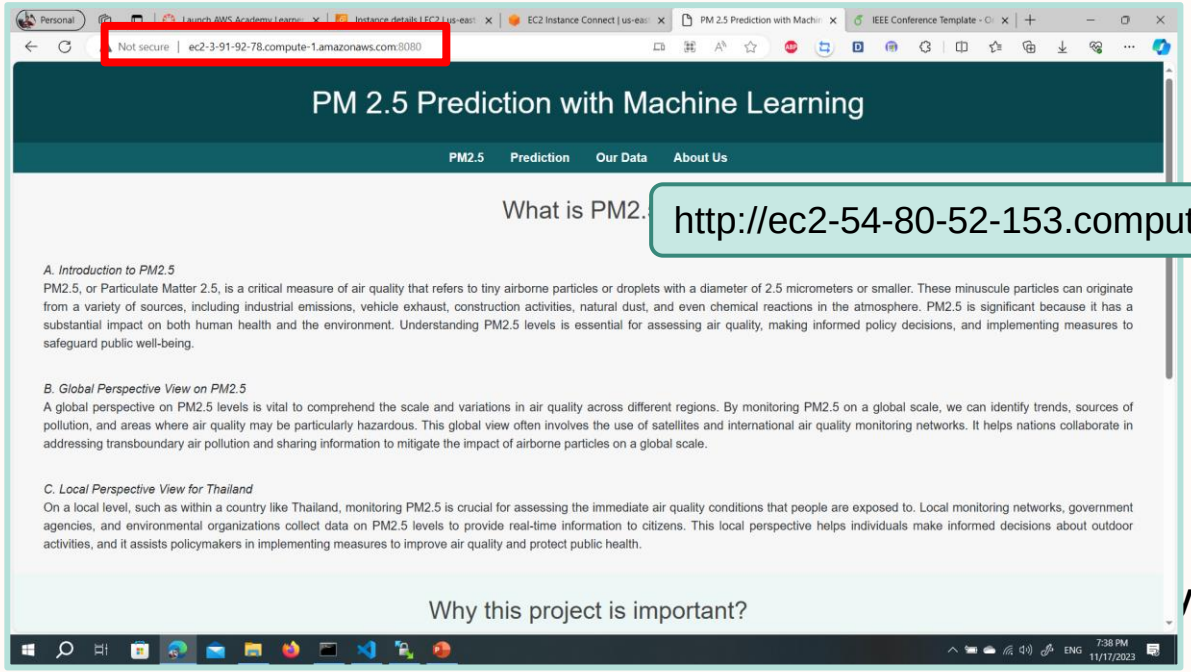
Project Info

Image Source:

Results

Deployment in aws

Create	Model
Pickle	Model
Create	Web Application
Deploy	AWS



http://ec2-54-80-52-153.compute-1.amazonaws.com:8080/

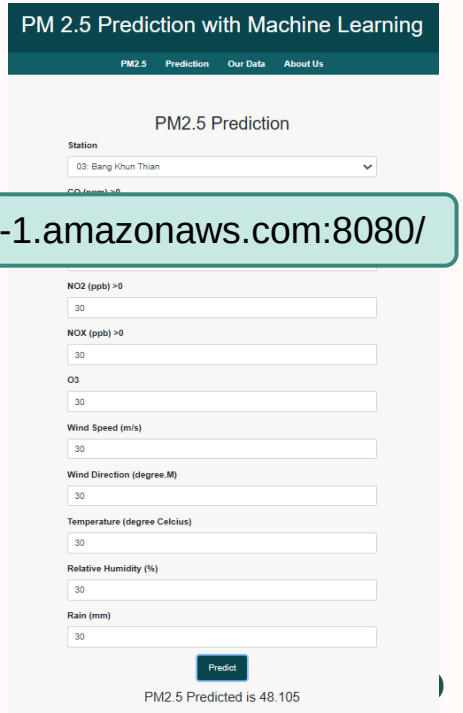


Image Source:

Conclusion & Future Works

Conclusion

01

02

03

04

05

- The main benefit of this project is predicting the PM2.5 value based on given weather conditions, specializing scikit-learn library to create a machine learning model to achieve that task and deploying it to the website for end-users to use.

- This model can also be used with other station data too, to further enhance the performance of the model, making it able to predict various data from other places.

Image Source:

Future Works

01
02
03
04
05

- **Time-series Analysis**

since the problem can be involved forecast with time-series.

- **Air Quality Index**

giving the users know more on how should they act and prepared (Decision-Making)

- **Include Weather data**

From other stations would give more in-depth details in PM2.5 data in Thailand will strengthen the model further in predicting PM2.5 in more diverged places

	US AQI Level	PM2.5 (μg/m³)	Health Recommendation (for 24hr exposure)
	Good 0-50	0-12.0	Air quality is satisfactory and poses little or no risk.
	Moderate 51-100	12.1-35.4	Sensitive individuals should avoid outdoor activity as they may experience respiratory symptoms.
	Unhealthy for sensitive groups 101-150	35.5-55.4	General public and sensitive individuals in particular are at risk to experience irritation and respiratory problems.
	Unhealthy 151-200	55.5-150.4	Increased likelihood of adverse effects and aggravation to the heart and lungs among general public.
	Very Unhealthy 201-300	150.5-250.4	General public will be noticeably affected. Sensitive groups should restrict outdoor activities.
	Hazardous 301+	250.5+	General public at high risk to experience strong irritations and adverse health effects. Everyone should avoid outdoor activities.

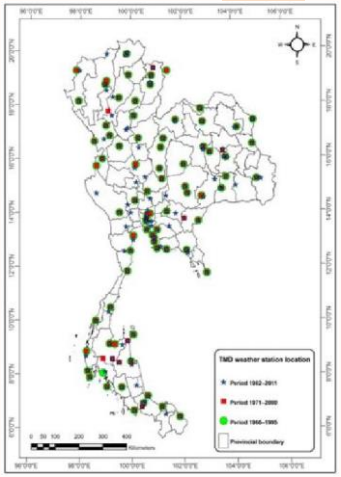


Image Source: https://www.researchgate.net/publication/294670317_Comparison_and_Recalibration_of_Equations_for_Estimating_Reference_Crop_Evapotranspiration_in_Thailand/figures?lo=1


THANK
YOU