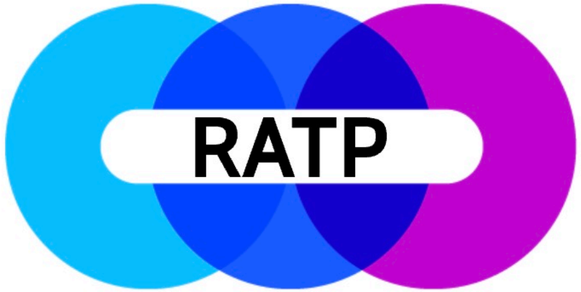
Data transformation

Modelisations

Correlation analysis

Franklin Roosevelt



Chatelet Auber

Predict levels in 3rs station?

Data preparation

Marie BAI, Alex CULPIN, Moussa SIDIBE

Data preparation

```
1 chatelet_air['Year'].value_counts()
```

```
2015    7450
2017    6744
2021    6471
2013    6438
2016    6400
2019    5891
2020    5372
2022    4118
2018     886
2014       1
Name: Year, dtype: int64
```

Chatelet

```
1 auber_air['Year'].value_counts()
```

```
2016    7904
2014    7273
2017    7136
2015    6636
2013    6009
2022    2647
Name: Year, dtype: int64
```

Auber

```
1 roosevelt_air['Year'].value_counts()
```

```
2016    7896
2014    7709
2017    7468
2019    7458
2015    7419
2018    7381
2020    5676
2021    5567
2013    5437
2022    5335
Name: Year, dtype: int64
```

Roosevelt

	DATE/HEURE	NO	NO2	PM10	CO2	TEMP	HUMI
0	2022-10-10T02:00:00+02:00	ND	ND	43	508	18,2	49,1
1	2022-10-10T01:00:00+02:00	ND	ND	45	529	18,5	48,4
2	2022-10-10T00:00:00+02:00	ND	ND	42	547	18,8	47,1
3	2022-10-09T23:00:00+02:00	ND	ND	59	614	19,1	47,2
4	2022-10-09T22:00:00+02:00	ND	ND	65	637	19,4	46,3

	DATE/HEURE	NO	NO2	PM10	PM2.5	CO2	TEMP	HUMI
0	2022-10-10T02:00:00+02:00	28	56	103	35	509	21,7	43,7
1	2022-10-10T01:00:00+02:00	29	58	108	41	518	21,8	43,1
2	2022-10-10T00:00:00+02:00	28	58	140	52	533	22	42,9
3	2022-10-09T23:00:00+02:00	27	57	113	41	570	22,1	43,1
4	2022-10-09T22:00:00+02:00	15	53	138	49	588	22,4	42,2

	date/heure	NO	NO2	PM10	CO2	TEMP	HUMI
0	2022-10-10T02:00:00+02:00	33	58	58	522	21,3	44
1	2022-10-10T01:00:00+02:00	53	63	48	540	21,5	43,3
2	2022-10-10T00:00:00+02:00	31	62	39	551	21,6	42,6
3	2022-10-09T23:00:00+02:00	21	63	31	542	21	44,7
4	2022-10-09T22:00:00+02:00	10	48	30	532	20,8	45,2

Missing values

Feature engineering

Missing values

Chatelet

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 85437 entries, 0 to 85436
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   DATE/HEURE  85437 non-null  datetime64[ns, UTC]
 1   NO          64108 non-null  object
 2   NO2         72046 non-null  object
 3   PM10        72820 non-null  object
 4   CO2         73133 non-null  object
 5   TEMP        74852 non-null  object
 6   HUMI        74863 non-null  object
dtypes: datetime64[ns, UTC](1), object(6)
memory usage: 4.6+ MB
```

Auber

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 84544 entries, 0 to 84543
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   DATE/HEURE  75808 non-null  datetime64[ns, UTC]
 1   NO          75808 non-null  object
 2   NO2         75808 non-null  object
 3   PM10        75808 non-null  object
 4   PM2.5       75808 non-null  object
 5   CO2         75808 non-null  object
 6   TEMP        75808 non-null  object
 7   HUMI        75808 non-null  object
dtypes: datetime64[ns, UTC](1), object(7)
memory usage: 5.2+ MB
```

Roosevelt

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 85437 entries, 0 to 85436
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   NO          80258 non-null  object
 1   NO2         80798 non-null  object
 2   PM10        82514 non-null  object
 3   CO2         80652 non-null  object
 4   TEMP        83535 non-null  object
 5   HUMI        83528 non-null  object
 6   DATE/HEURE  85437 non-null  datetime64[ns, UTC]
dtypes: datetime64[ns, UTC](1), object(6)
memory usage: 4.6+ MB
```

Date engineering

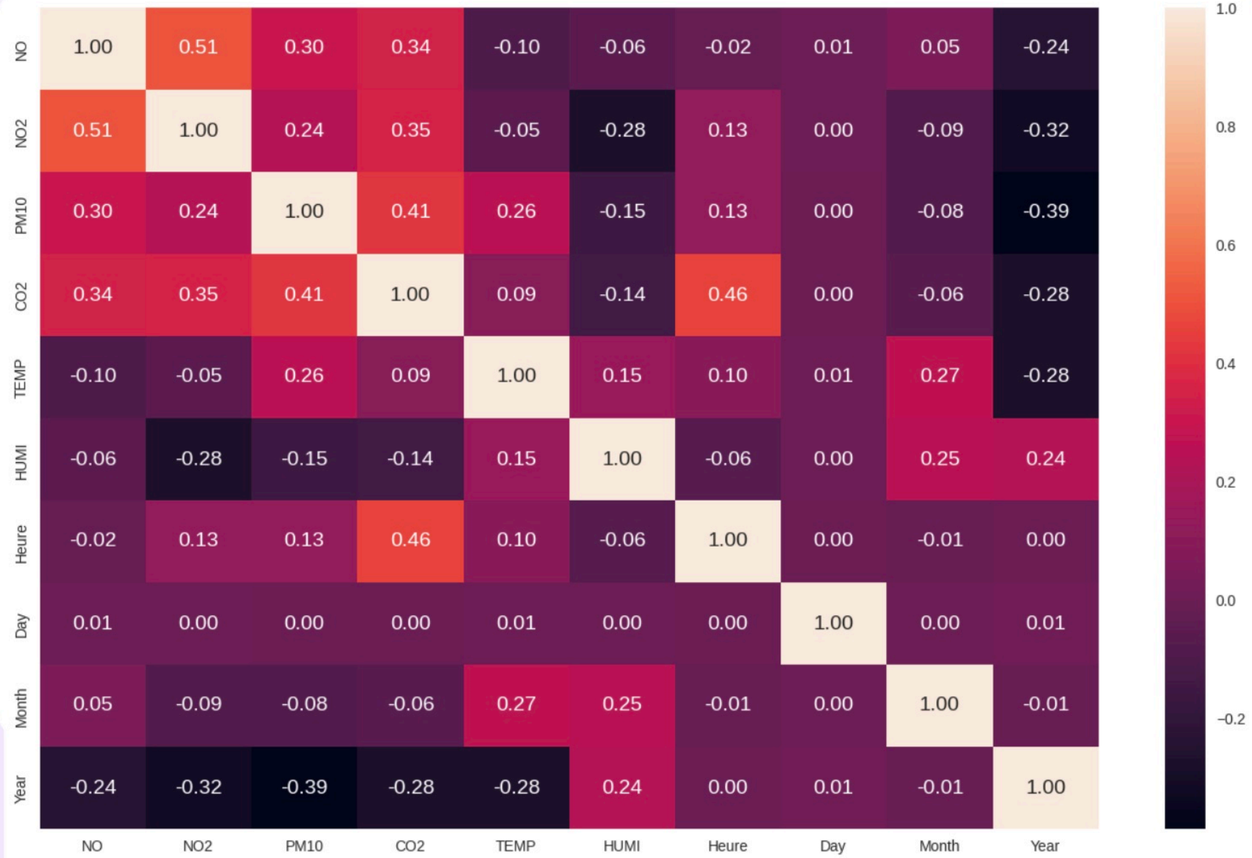
```
1 roosevelt_air.head(2)
```

	NO	NO2	PM10	CO2	TEMP	HUMI	Heure	Day	Month	Year
0	33	58	58	522	21,3	44	0	10	10	2022
1	53	63	48	540	21,5	43,3	23	9	10	2022

```
1 roosevelt_air=convert_object_to_float(roosevelt_air)
2 roosevelt_air.head(2)
```

	NO	NO2	PM10	CO2	TEMP	HUMI	Heure	Day	Month	Year
0	33.0	58.0	58.0	522.0	21.3	44.0	0	10	10	2022
1	53.0	63.0	48.0	540.0	21.5	43.3	23	9	10	2022

Chatelet



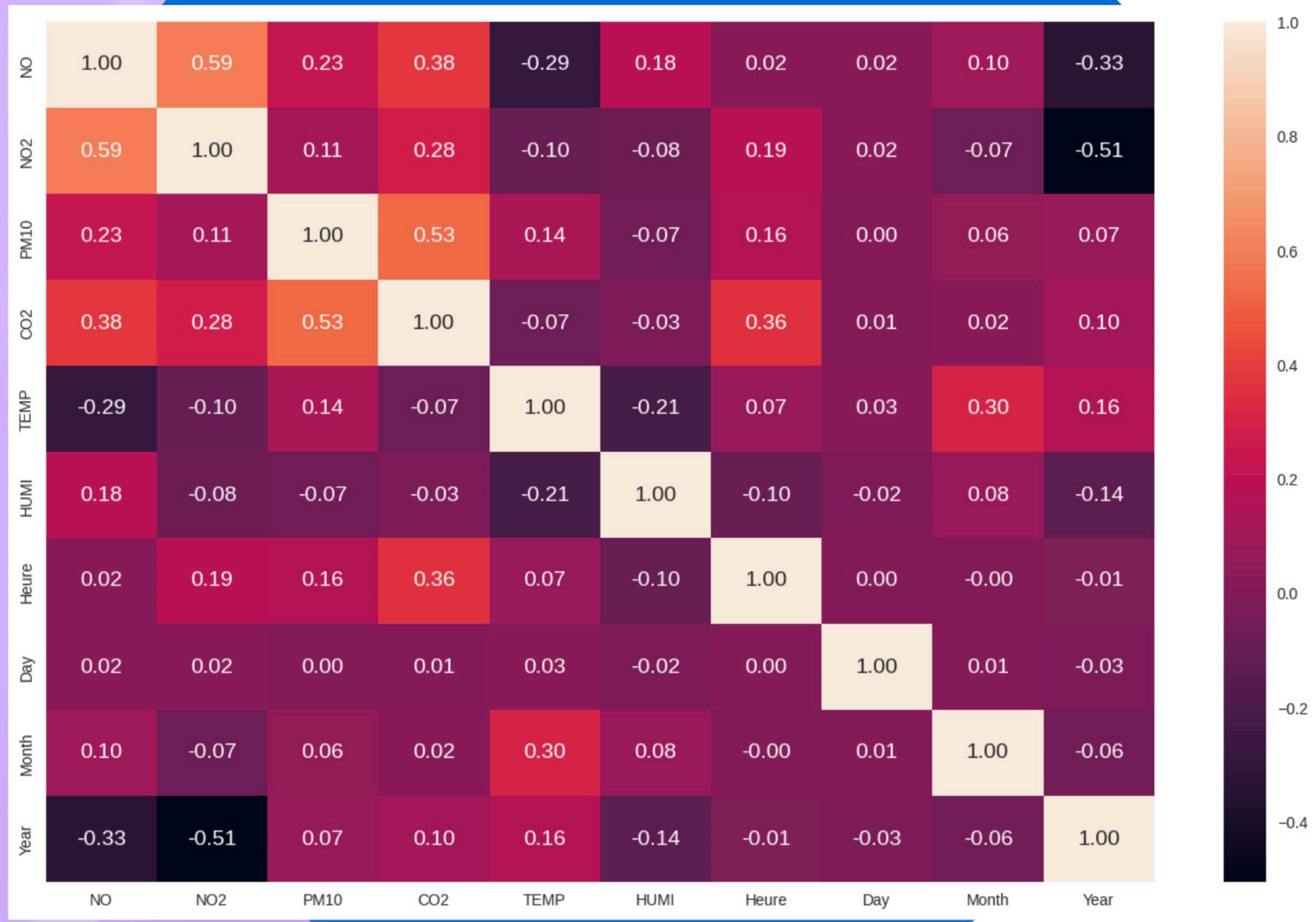
Auber

Roosevelt

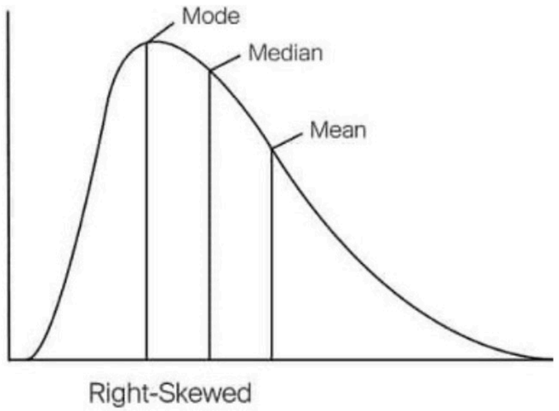
Auber


NO 1.00 0.54 0.11 0.13 0.22 -0.14 0.04 -0.09 0.01 0.06 -0.13
NO2 0.54 1.00 -0.04 -0.03 0.15 0.06 -0.22 0.17 0.01 -0.08 -0.09
PM10 0.11 -0.04 1.00 0.93 0.49 0.17 0.15 0.18 0.00 -0.09 -0.14
PM2.5 0.13 -0.03 0.93 1.00 0.50 0.16 0.14 0.18 -0.00 -0.08 -0.14
CO2 0.22 0.15 0.49 0.50 1.00 0.05 0.02 0.33 0.01 -0.03 0.06
TEMP -0.14 0.06 0.17 0.16 0.05 1.00 0.18 0.10 0.03 0.31 0.08
HUMI 0.04 -0.22 0.15 0.14 0.02 0.18 1.00 -0.05 0.00 0.09 -0.05
Heure -0.09 0.17 0.18 0.18 0.33 0.10 -0.05 1.00 -0.00 0.00 -0.01
Day 0.01 0.01 0.00 -0.00 0.01 0.03 0.00 -0.00 1.00 -0.01 -0.01
Month 0.06 -0.08 -0.09 -0.08 -0.03 0.31 0.09 0.00 -0.01 1.00 -0.03
Year -0.13 -0.09 -0.14 -0.14 0.06 0.08 -0.05 -0.01 -0.01 -0.03 1.00
NO NO2 PM10 PM2.5 CO2 TEMP HUMI Heure Day Month Year
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

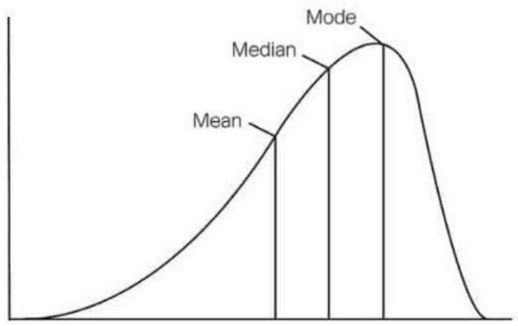
Roosevelt



Skeweness



Right-Skewed



Left-Skewed (Negative Skewness)

Disadvantage

Tail act as outlier

Outlier affect regression based model performance

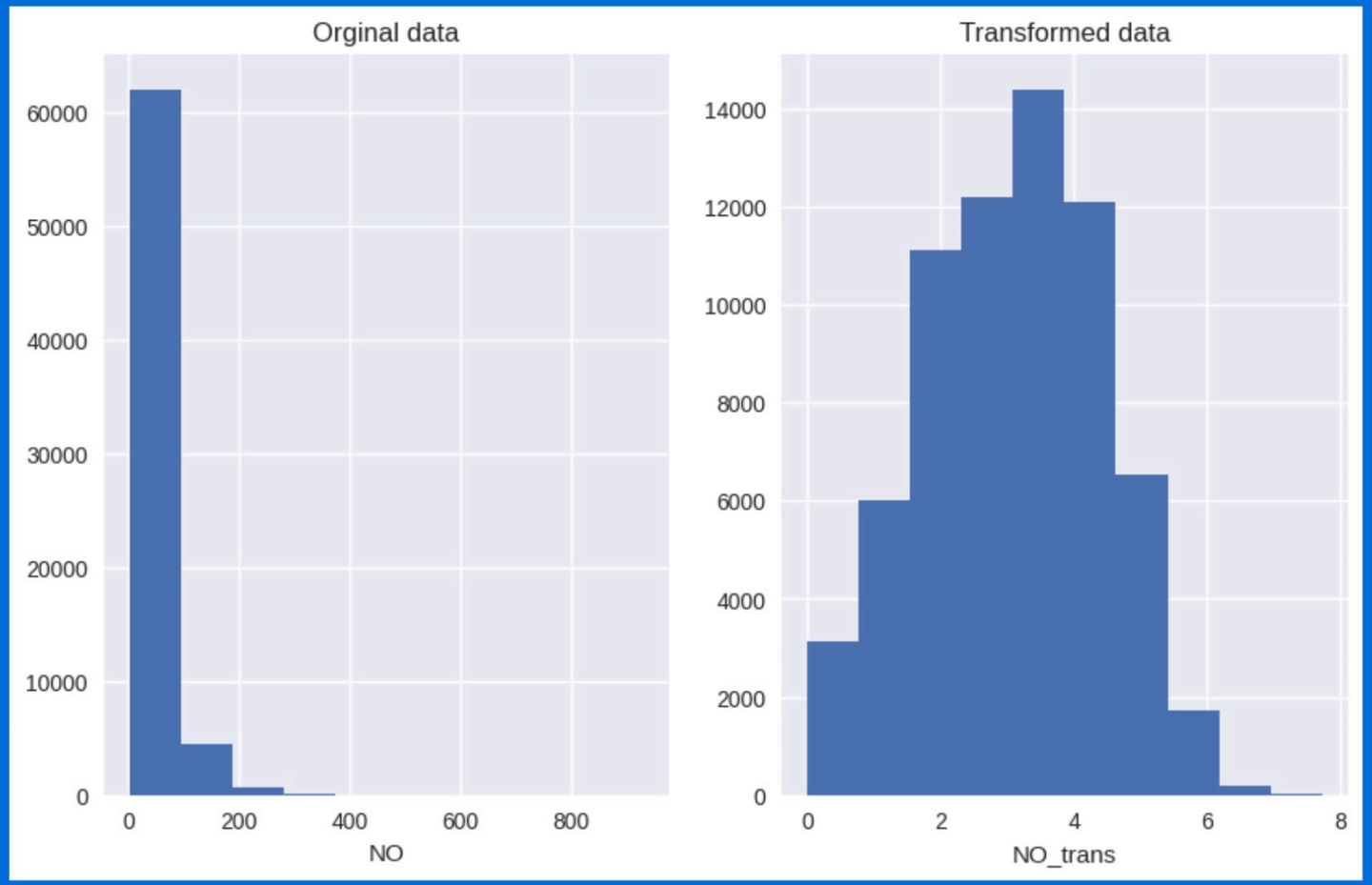
Box-Cox Transformation

Inverse Box-cox Transformation

Box-Cox Transformation

Goal: Transform skewed data to symetric distribution

scipy.special import boxcox



$$y_i^{(\lambda)} = \begin{cases} \dfrac{y_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(y_i) & \text{if } \lambda = 0, \end{cases}$$

Inverse Box-cox

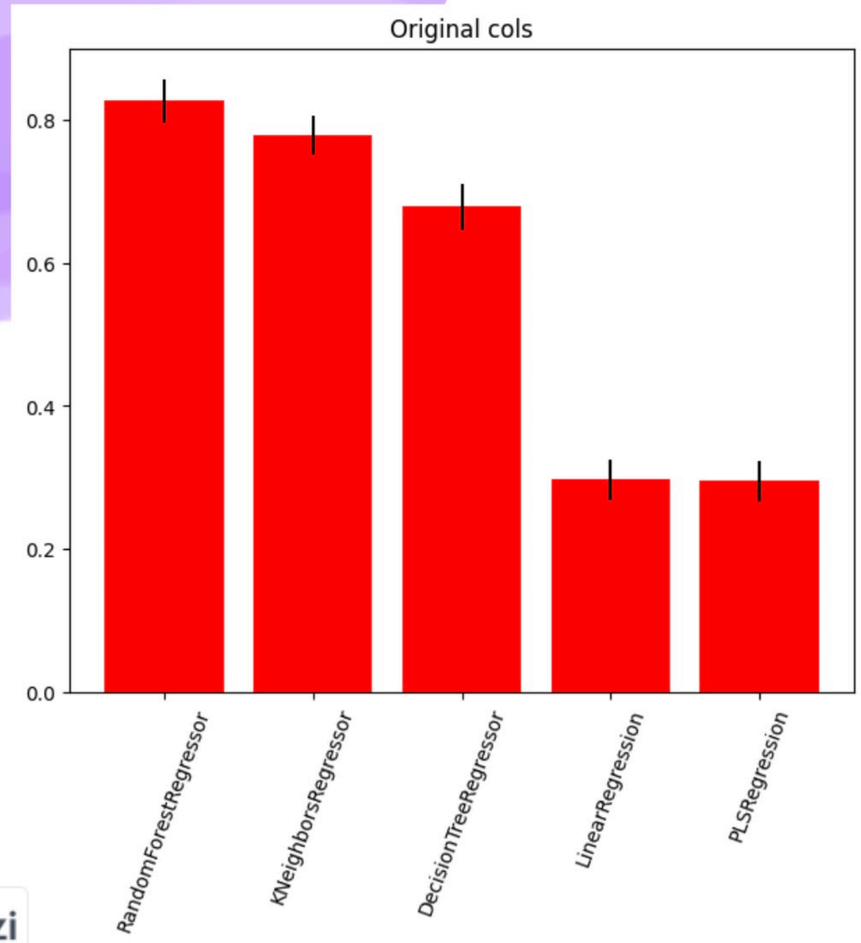
scipy.special import inv_boxcox

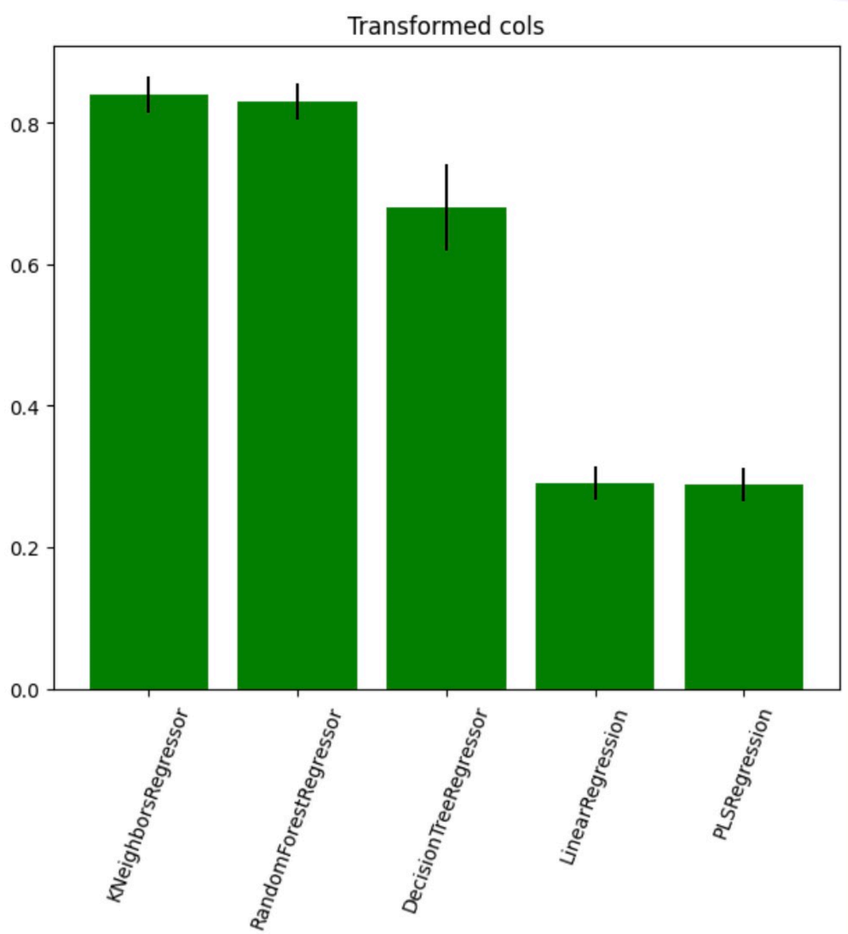
To inverse transformation to the original after prediction

To inverse transformation to the original values after prediction

Chatelet
CO2/NO/NO2/PM10

CV=30, Transformed cols perform better





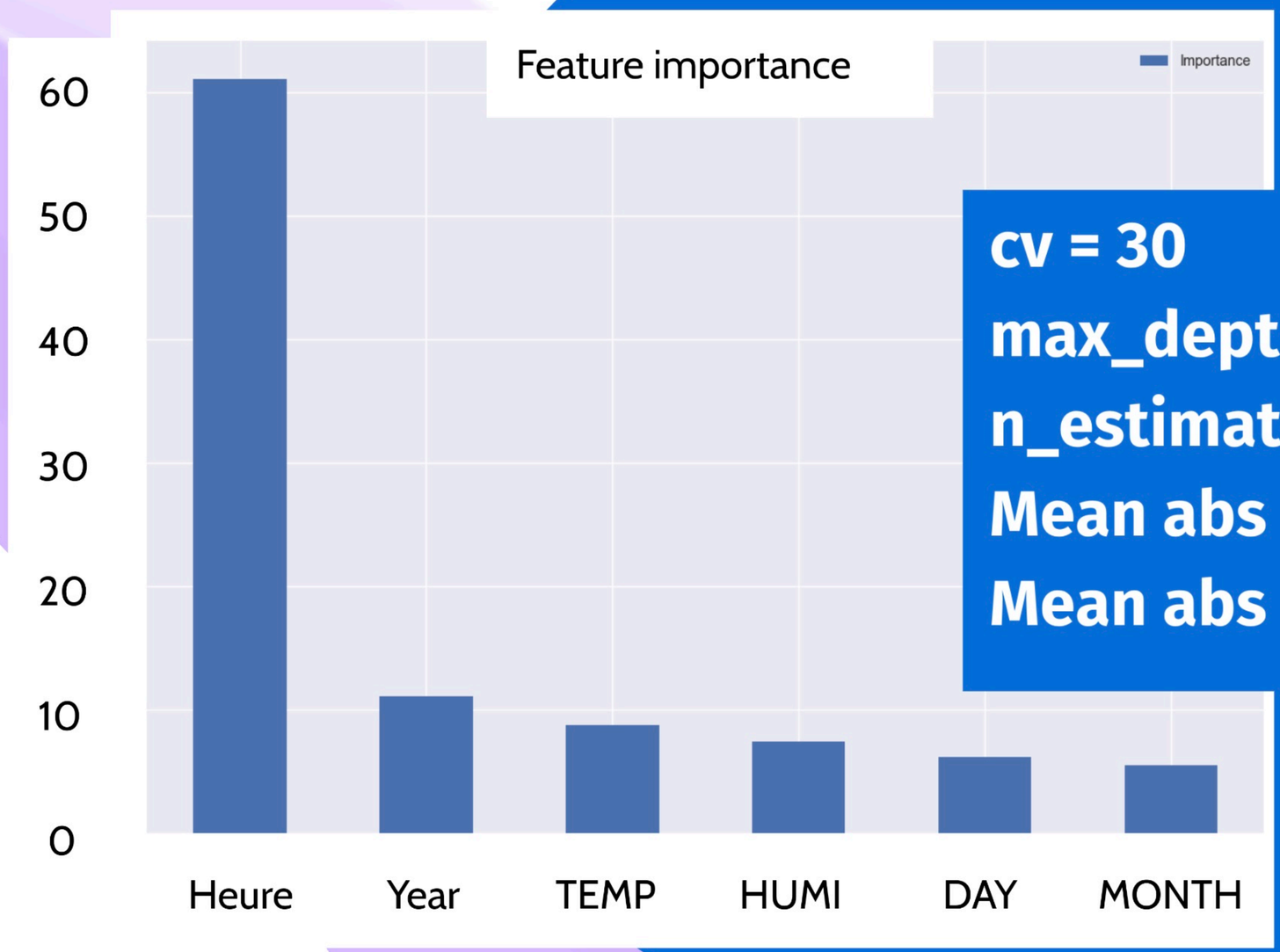
CO2

NO

NO2

PM10

Chatelet CO 2



$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

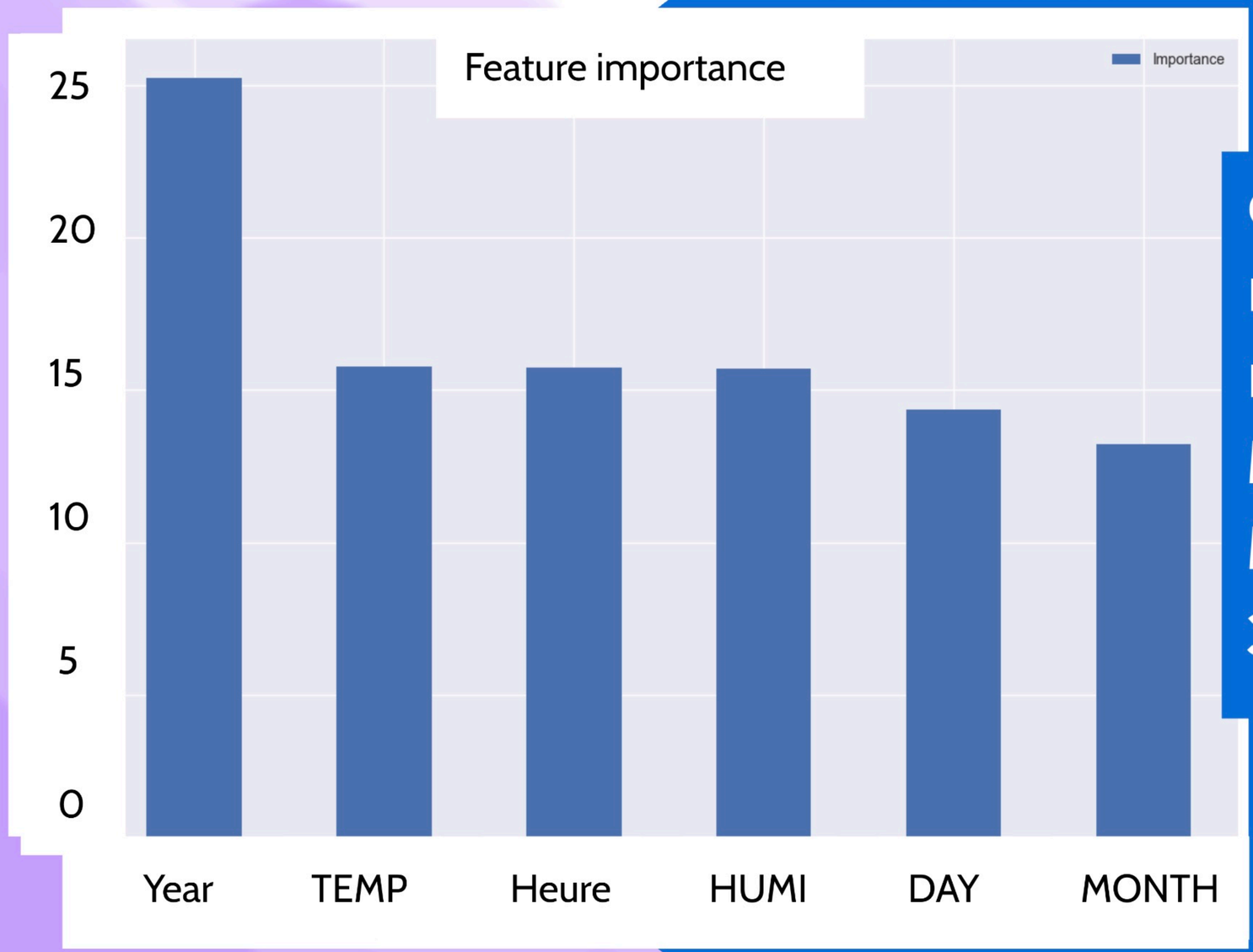
MAE = mean absolute error
y_i = prediction
x_i = true value
n = total number of data points

cv = 30
max_depth: 25
n_estimators: 800
Mean abs error: 30.87
Mean abs error percentage: 5.18%

$$M = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

M = mean absolute percentage error
n = number of times the summation iteration happens
A_t = actual value
F_t = forecast value

Chatelet NO



cv = 30

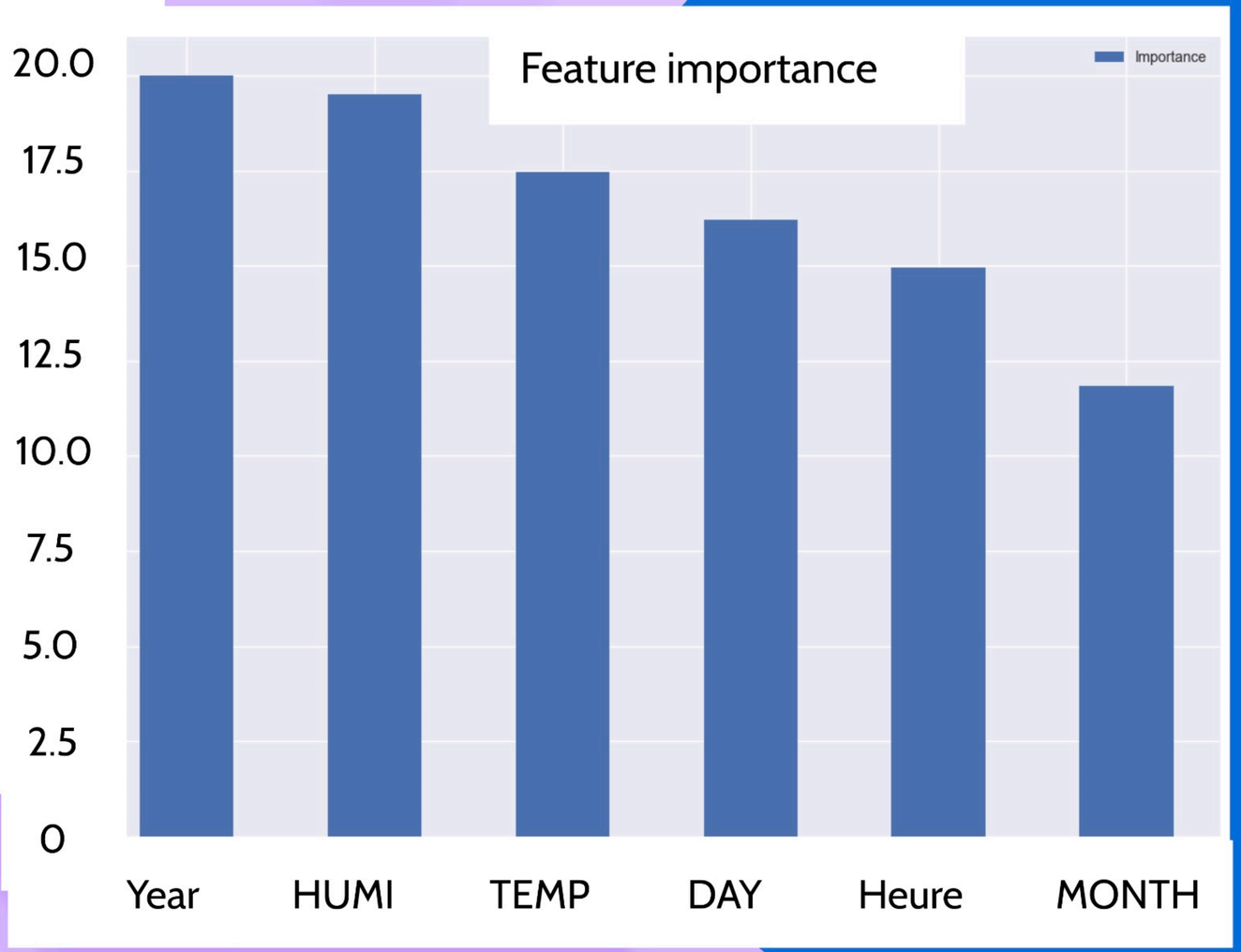
max_depth: 25

n_estimators: 800

Mean abs error: 26.06

Mean abs error percentage: 33.98%

Chatelet NO 2



cv = 30

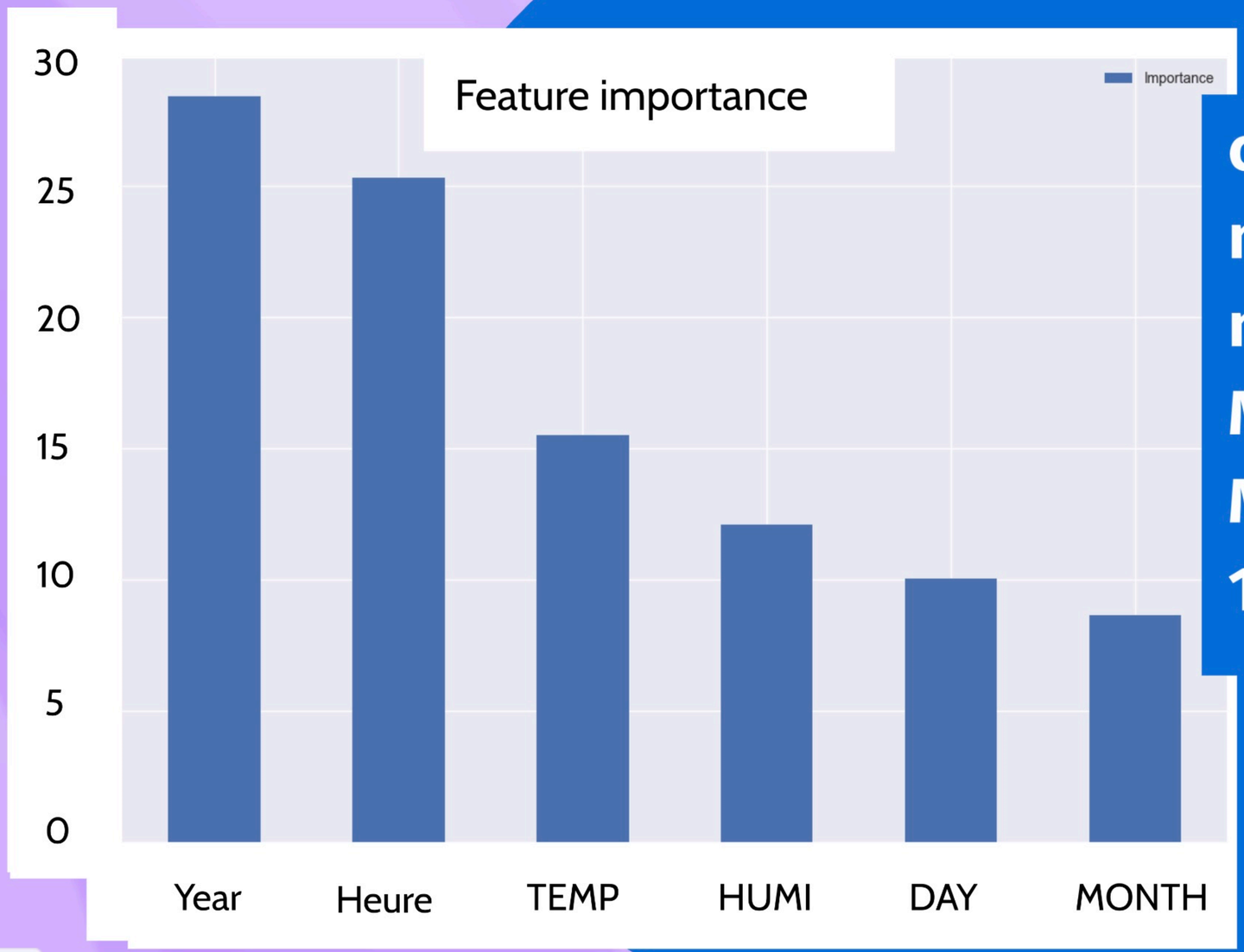
max_depth: 25

n_estimators: 1000

Mean abs error: 10.19

Mean abs error percentage: 43.53%

Chatelet PM 10



cv = 30

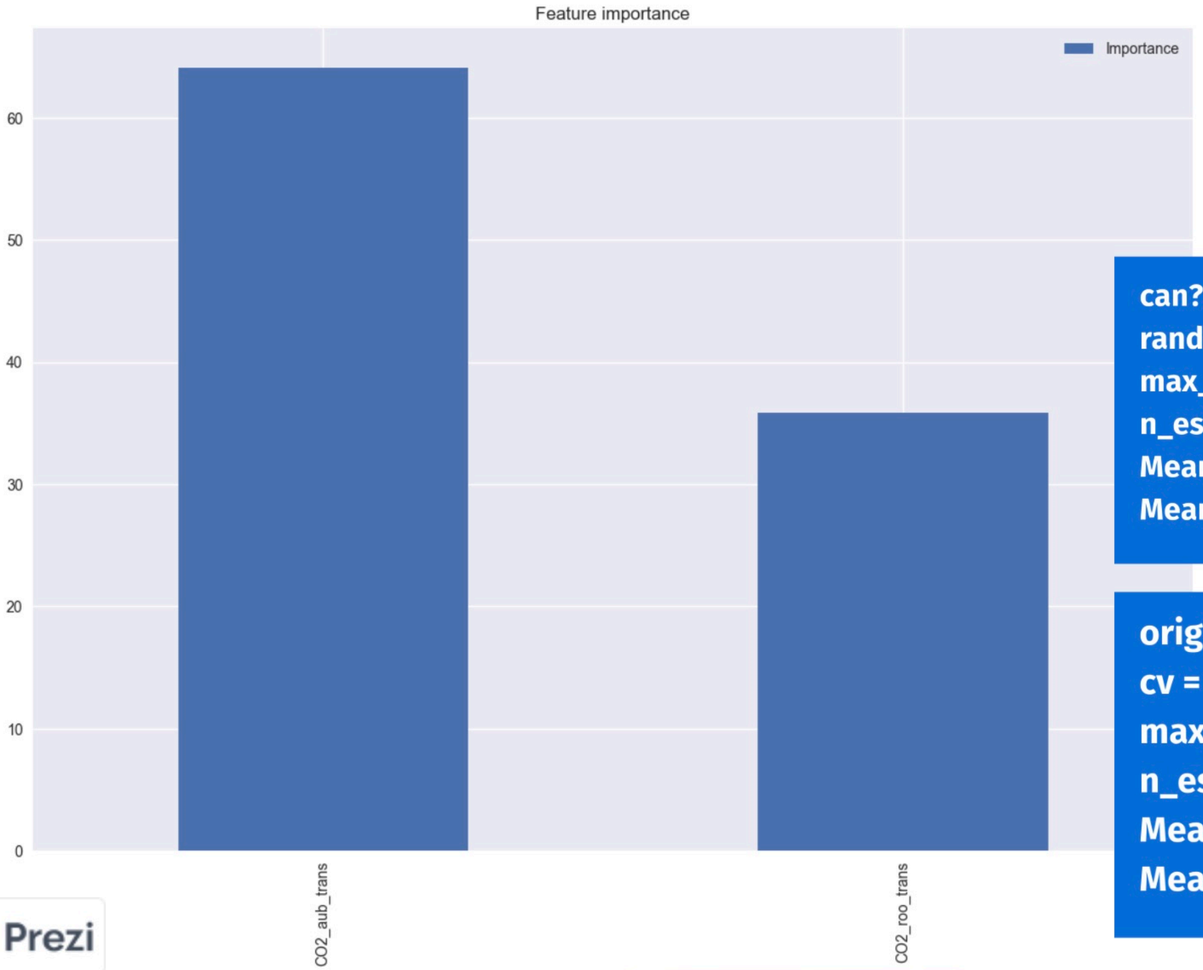
max_depth: 25

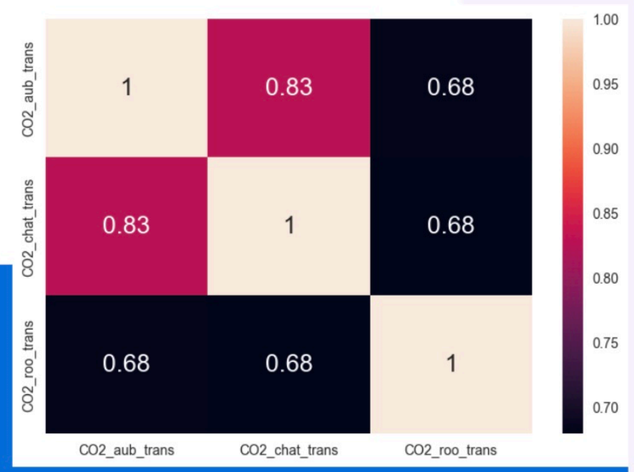
n_estimators: 700

Mean abs error: 5.30

Mean abs error percentage: 17.73%

Can current levels in two stations help predict levels in third station?





can?
random_state = 5
max_depth = 25
n_estimators = 800
Mean abs error: 59.42
Mean abs error percentage: 9.55%

original:
cv = 30
max_depth: 25
n_estimators: 800
Mean abs error: 30.87
Mean abs error percentage: 5.18%