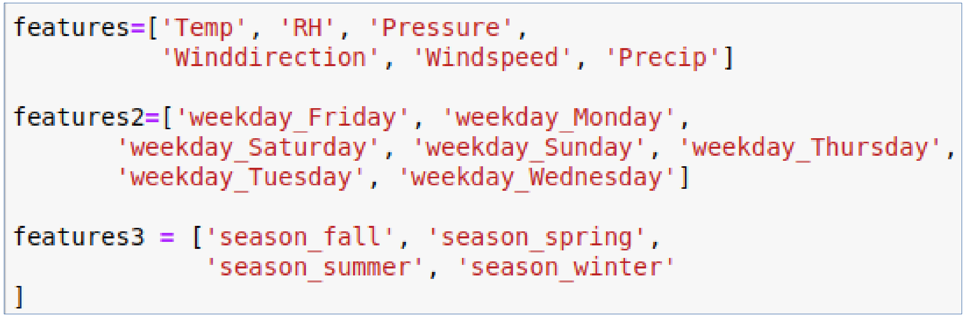
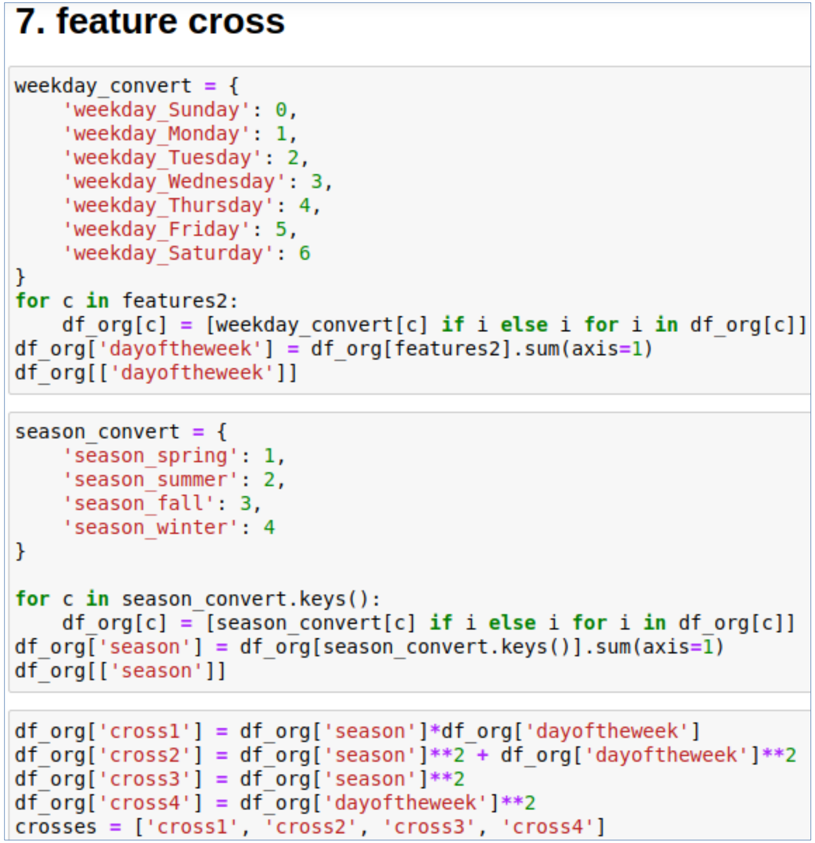
Graz air pollution

Methods

1. Base additive: *f0_add*
2. Base multiplicative: *f0_mul*
3. + features: *f1*
4. + features + features2: *f2*
5. + holidays + features: *f3*
6. + holidays + featues + features2 + features3: *f4*
7. 6. + crossed_features: *f5*
8. 7. + crossed_more_features: *f6*



```
features=['Temp', 'RH', 'Pressure',
          'Winddirection', 'Windspeed', 'Precip']

features2=['weekday_Friday', 'weekday_Monday',
       'weekday_Saturday', 'weekday_Sunday', 'weekday_Thursday',
       'weekday_Tuesday', 'weekday_Wednesday']

features3 = ['season_fall', 'season_spring',
             'season_summer', 'season_winter'
]
```



7. feature cross

```
weekday_convert = {
    'weekday_Sunday': 0,
    'weekday_Monday': 1,
    'weekday_Tuesday': 2,
    'weekday_Wednesday': 3,
    'weekday_Thursday': 4,
    'weekday_Friday': 5,
    'weekday_Saturday': 6
}
for c in features2:
    df_org[c] = [weekday_convert[c] if i else i for i in df_org[c]]
df_org['dayoftheweek'] = df_org[features2].sum(axis=1)
df_org[['dayoftheweek']]
```

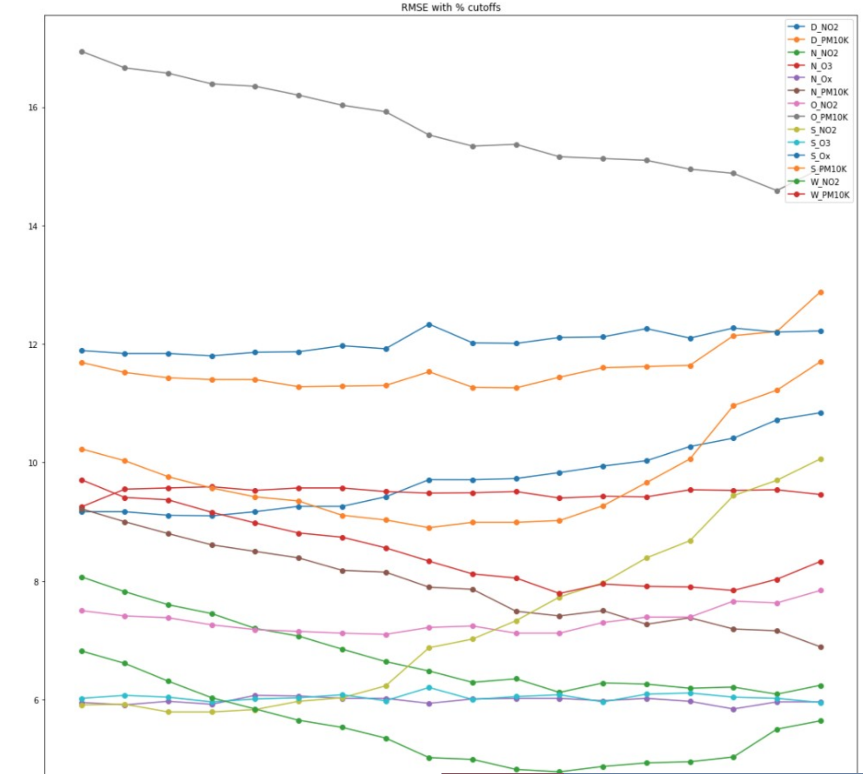
```
season_convert = {
    'season_spring': 1,
    'season_summer': 2,
    'season_fall': 3,
    'season_winter': 4
}

for c in season_convert.keys():
    df_org[c] = [season_convert[c] if i else i for i in df_org[c]]
df_org['season'] = df_org[season_convert.keys()].sum(axis=1)
df_org[['season']]
```

```
df_org['cross1'] = df_org['season']*df_org['dayoftheweek']
df_org['cross2'] = df_org['season']**2 + df_org['dayoftheweek']**2
df_org['cross3'] = df_org['season']**2
df_org['cross4'] = df_org['dayoftheweek']**2
crosses = ['cross1', 'cross2', 'cross3', 'cross4']
```

Scenario 1

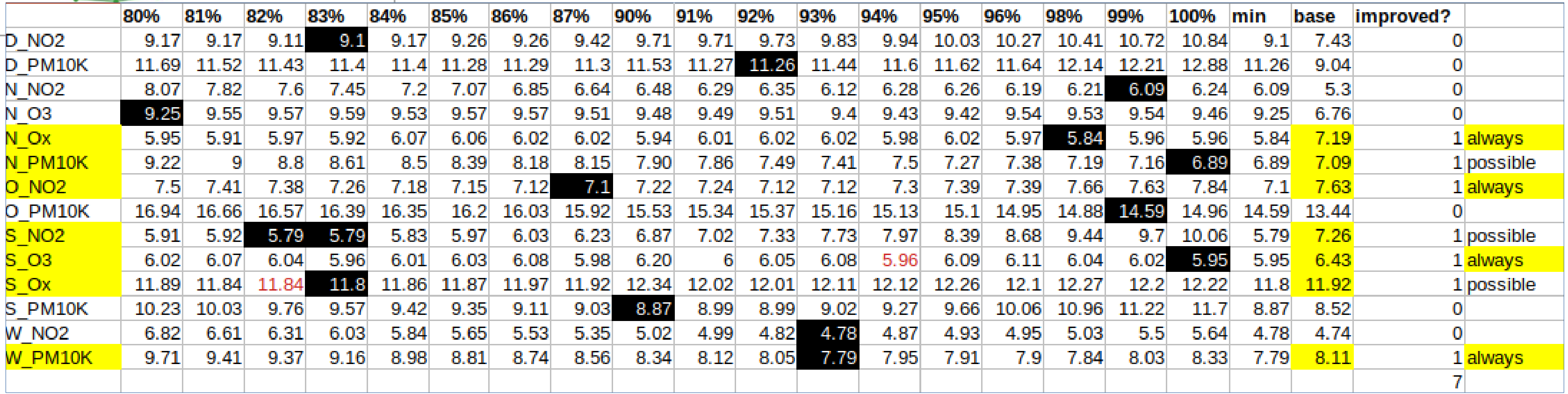
- Train: → end of 2019
- Test: 03 Jan – 10 Mar

% Cutoffs (train set)



7/14 are better without parameter tuning

Notes: the rest is still running on the server

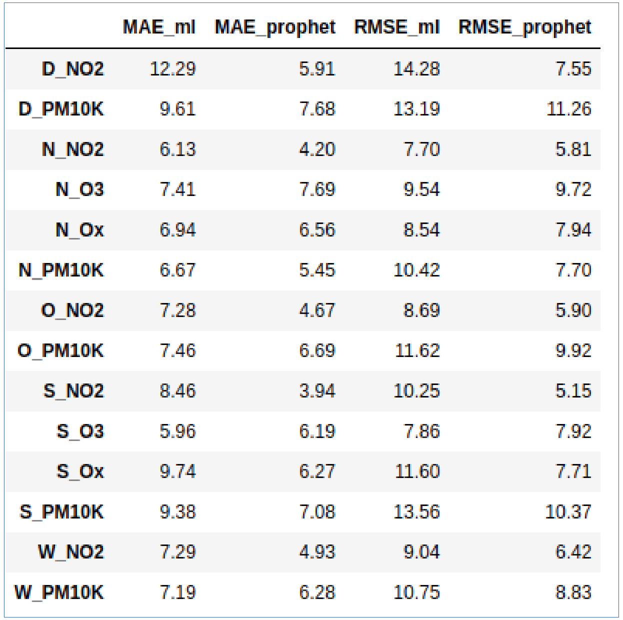


	80%	81%	82%	83%	84%	85%	86%	87%	90%	91%	92%	93%	94%	95%	96%	98%	99%	100%	min	base	improved?	
D_NO2	9.17	9.17	9.11	9.1	9.17	9.26	9.26	9.42	9.71	9.71	9.73	9.83	9.94	10.03	10.27	10.41	10.72	10.84	9.1	7.43	0	
D_PM10K	11.69	11.52	11.43	11.4	11.4	11.28	11.29	11.3	11.53	11.27	11.26	11.44	11.6	11.62	11.64	12.14	12.21	12.88	11.26	9.04	0	
N_NO2	8.07	7.82	7.6	7.45	7.2	7.07	6.85	6.64	6.48	6.29	6.35	6.12	6.28	6.26	6.19	6.21	6.09	6.24	6.09	5.3	0	
N_O3	9.25	9.55	9.57	9.59	9.53	9.57	9.57	9.51	9.48	9.49	9.51	9.4	9.43	9.42	9.54	9.53	9.54	9.46	9.25	6.76	0	
N_Ox	5.95	5.91	5.97	5.92	6.07	6.06	6.02	6.02	5.94	6.01	6.02	6.02	5.98	6.02	5.97	5.84	5.96	5.96	5.84	7.19	1	always
N_PM10K	9.22	9	8.8	8.61	8.5	8.39	8.18	8.15	7.90	7.86	7.49	7.41	7.5	7.27	7.38	7.19	7.16	6.89	6.89	7.09	1	possible
O_NO2	7.5	7.41	7.38	7.26	7.18	7.15	7.12	7.1	7.22	7.24	7.12	7.12	7.3	7.39	7.39	7.66	7.63	7.84	7.1	7.63	1	always
O_PM10K	16.94	16.66	16.57	16.39	16.35	16.2	16.03	15.92	15.53	15.34	15.37	15.16	15.13	15.1	14.95	14.88	14.59	14.96	14.59	13.44	0	
S_NO2	5.91	5.92	5.79	5.79	5.83	5.97	6.03	6.23	6.87	7.02	7.33	7.73	7.97	8.39	8.68	9.44	9.7	10.06	5.79	7.26	1	possible
S_O3	6.02	6.07	6.04	5.96	6.01	6.03	6.08	5.98	6.20	6	6.05	6.08	5.96	6.09	6.11	6.04	6.02	5.95	5.95	6.43	1	always
S_Ox	11.89	11.84	11.84	11.8	11.86	11.87	11.97	11.92	12.34	12.02	12.01	12.11	12.12	12.26	12.1	12.27	12.2	12.22	11.8	11.92	1	possible
S_PM10K	10.23	10.03	9.76	9.57	9.42	9.35	9.11	9.03	8.87	8.99	8.99	9.02	9.27	9.66	10.06	10.96	11.22	11.7	8.87	8.52	0	
W_NO2	6.82	6.61	6.31	6.03	5.84	5.65	5.53	5.35	5.02	4.99	4.82	4.78	4.87	4.93	4.95	5.03	5.5	5.64	4.78	4.74	0	
W_PM10K	9.71	9.41	9.37	9.16	8.98	8.81	8.74	8.56	8.34	8.12	8.05	7.79	7.95	7.91	7.9	7.84	8.03	8.33	7.79	8.11	1	always
																					7	

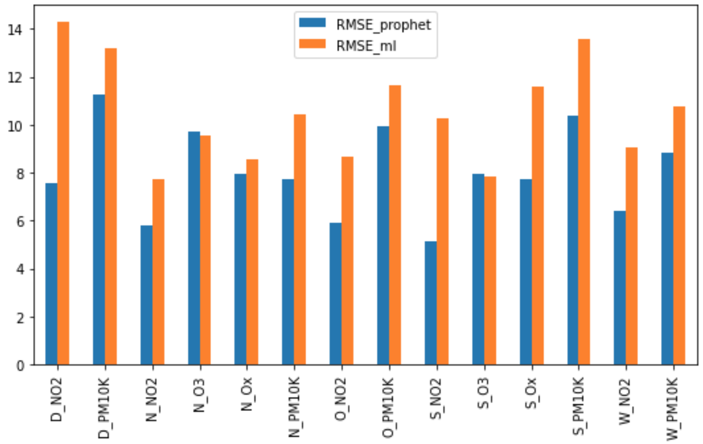
Scenario 2

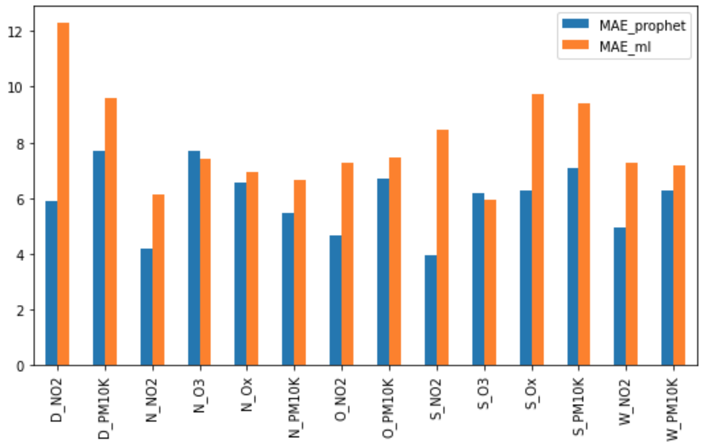
- Train: → 10 Mar 2020
- Test: 11 Mar 2020 – 10 Mar 2021

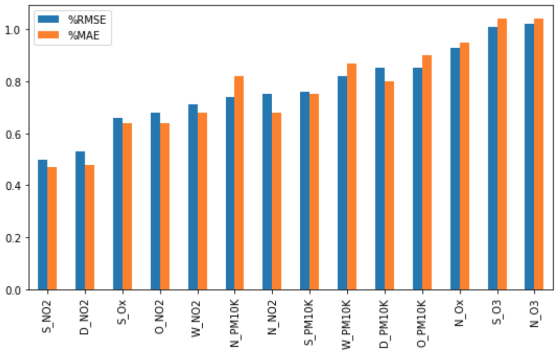
1 year



	MAE_ml	MAE_prophet	RMSE_ml	RMSE_prophet
D_NO2	12.29	5.91	14.28	7.55
D_PM10K	9.61	7.68	13.19	11.26
N_NO2	6.13	4.20	7.70	5.81
N_O3	7.41	7.69	9.54	9.72
N_Ox	6.94	6.56	8.54	7.94
N_PM10K	6.67	5.45	10.42	7.70
O_NO2	7.28	4.67	8.69	5.90
O_PM10K	7.46	6.69	11.62	9.92
S_NO2	8.46	3.94	10.25	5.15
S_O3	5.96	6.19	7.86	7.92
S_Ox	9.74	6.27	11.60	7.71
S_PM10K	9.38	7.08	13.56	10.37
W_NO2	7.29	4.93	9.04	6.42
W_PM10K	7.19	6.28	10.75	8.83

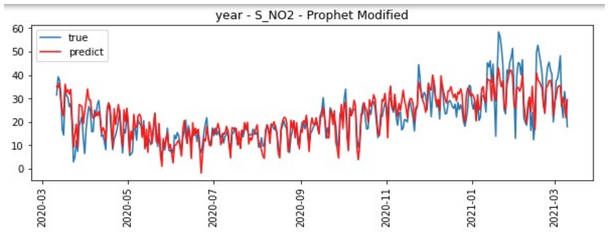


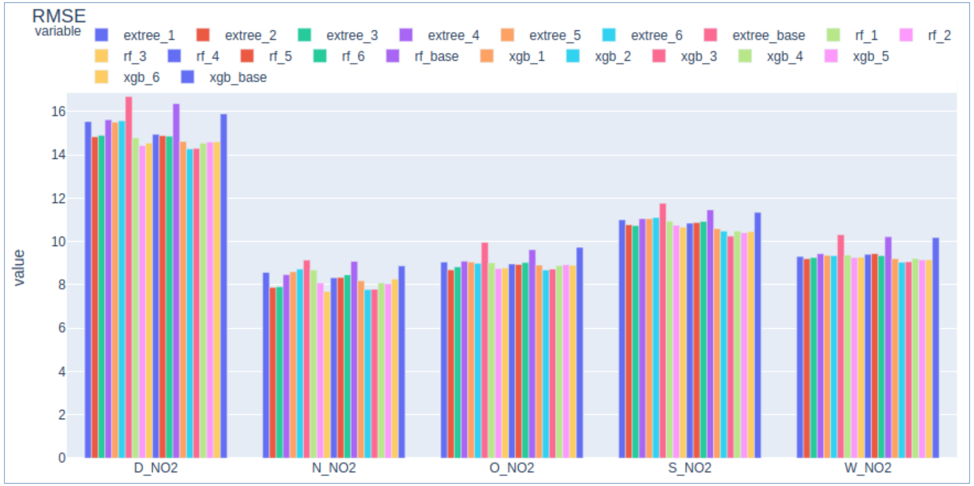


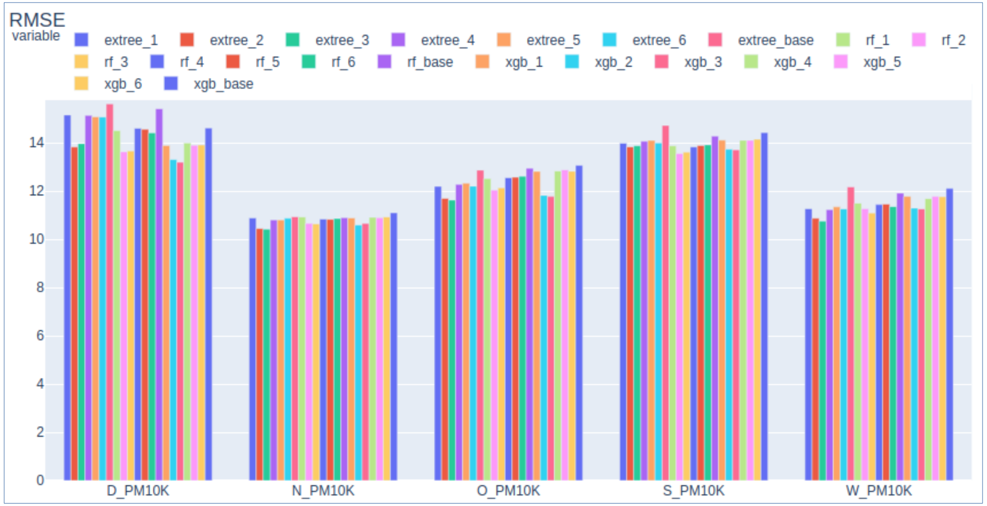


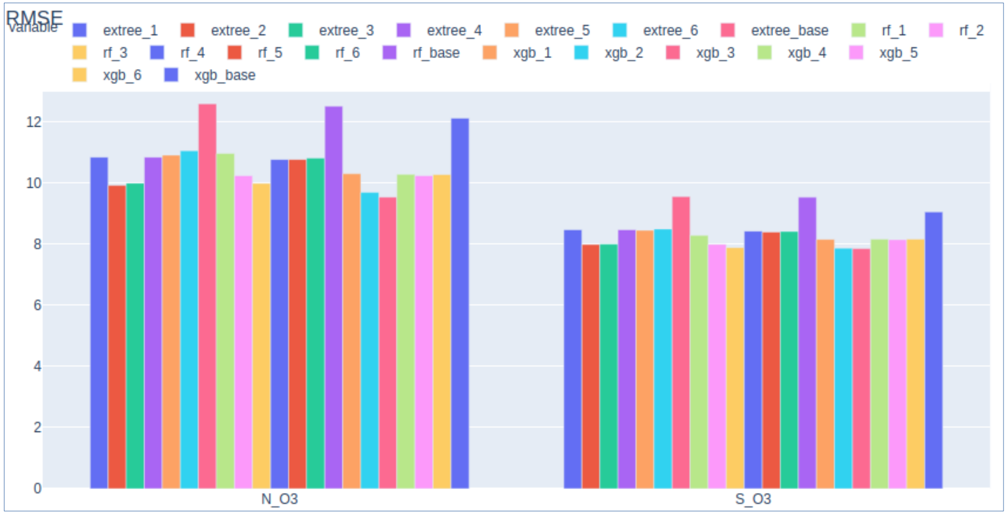
- ML better only at N_O3 and S_O3
- Prophet good at NO2

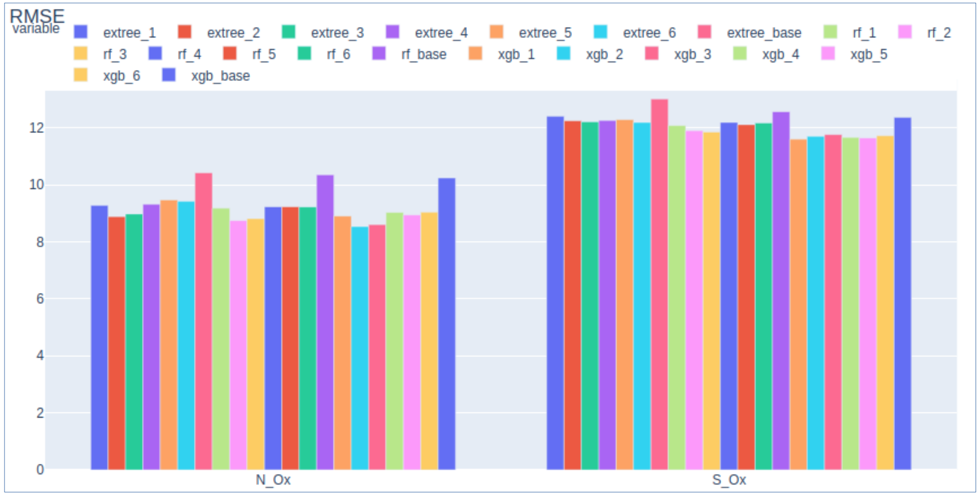
The best




RMSE
variable
extree_1
extree_2
extree_3
extree_4
extree_5
extree_6
extree_base
rf_1
rf_2
rf_3
rf_4
rf_5
rf_6
rf_base
xgb_1
xgb_2
xgb_3
xgb_4
xgb_5
xgb_6
xgb_base
value
0
2
4
6
8
10
12
14
16
D_NO2
N_NO2
O_NO2
S_NO2
W_NO2


RMSE
variable
extree_1
extree_2
extree_3
extree_4
extree_5
extree_6
extree_base
rf_1
rf_2
rf_3
rf_4
rf_5
rf_6
rf_base
xgb_1
xgb_2
xgb_3
xgb_4
xgb_5
xgb_6
xgb_base
0
2
4
6
8
10
12
14
D_PM10K
N_PM10K
O_PM10K
S_PM10K
W_PM10K


RMSE
variable
extree_1
extree_2
extree_3
extree_4
extree_5
extree_6
extree_base
rf_1
rf_2
rf_3
rf_4
rf_5
rf_6
rf_base
xgb_1
xgb_2
xgb_3
xgb_4
xgb_5
xgb_6
xgb_base
0
2
4
6
8
10
12
N_O3
S_O3


RMSE
variable
extree_1
extree_2
extree_3
extree_4
extree_5
extree_6
extree_base
rf_1
rf_2
rf_3
rf_4
rf_5
rf_6
rf_base
xgb_1
xgb_2
xgb_3
xgb_4
xgb_5
xgb_6
xgb_base
0
2
4
6
8
10
12
N_Ox
S_Ox

Thank you!