Western Norway University of Applied Sciences

High Performance Python Lab
Final project

Skoltech
Skolkovo Institute of Science and Technology

ML prediction for thermal performance of vacuum tube solar collectors

Oleg Nikolaev, 2nd year MSc student, PE

Nikita Kuznetsov, 1st year MSc student, ACS

Moscow
2021

energies

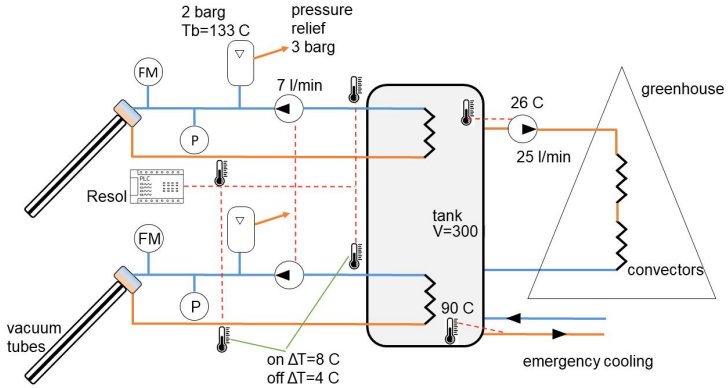
MDPI

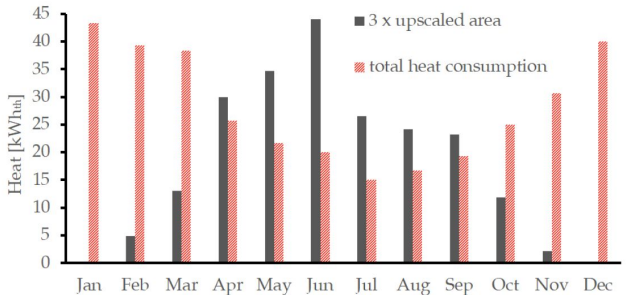
Skoltech

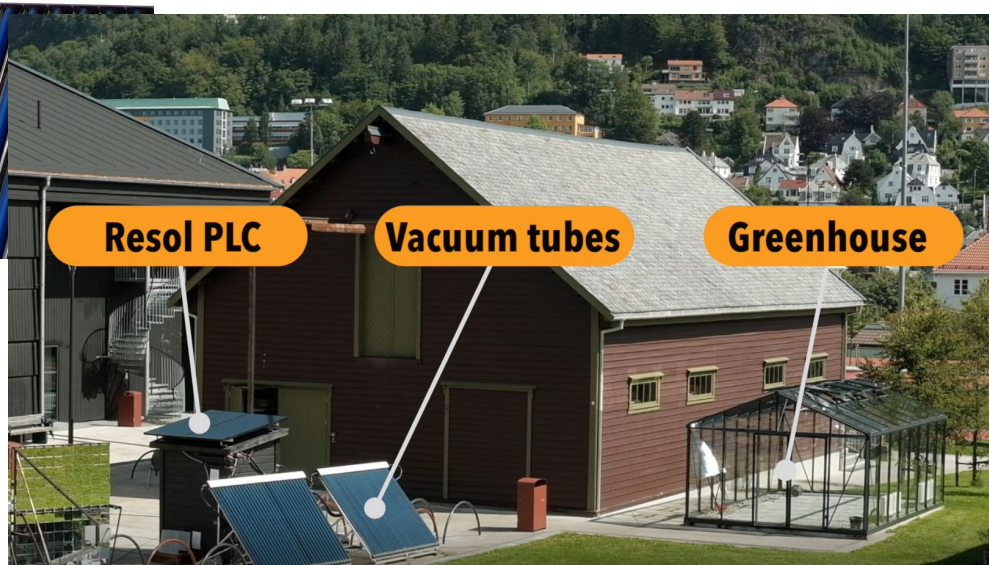
Article

Field Study on the Thermal Performance of Vacuum Tube Solar Collectors in the Climate Conditions of Western Norway

Victoria Popsueva [1], Andrés Franklin Olivares Lopez [1], Anna Kosinska [1] and Oleg Nikolaev [2,3] and Boris V. Balakin [1,2,*]







Data set (df)

23.07.2019 - 17.09.2020 (403 days)

Skoltech

Feature 1 (date)

Target (efficiency)

Feature 2 (solar irradiation *W/m2*)

Feature 3 (ambient temperature *°C*)

Feature 4 (theoretical efficiency)

at red days collectors were stopped

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ДАТА	КПД1	КПД2	МОЩ1, [ВТ]	МОЩ2,[ВТ]	Солн изл,	Темп окр	Всего тепла,	Теор КПД1	Теор КПД2	Разница	Разница	День для
2	23.07.2019	0,5802	0,6843	495,6505	699,8281	429,3941	22,1619	2,67E+07	0,6823	0,6772	-0,1021	0,0071	
3	24.07.2019	0,7896	0,8784	1,00E+03	1,30E+03	473,2252	23,8903	7,52E+07	0,6353	0,6245	0,1543	0,2539	
4	25.07.2019	0,8016	0,8881	981,3947	1,28E+03	466,4756	23,4041	7,28E+07	0,6213	0,613	0,1803	0,2751	
5	26.07.2019	0,7721	0,8984	914,9467	1,28E+03	467,5152	29,4243	8,73E+07	0,6159	0,6077	0,1562	0,2907	250719
6	27.07.2019	0,7672	0,7761	675,9699	791,2702	367,7464	27,8901	6,11E+07	0,6658	0,6552	0,1014	0,1209	
7	28.07.2019	0,7763	0,8416	981,3663	1,24E+03	468,582	28,7855	7,38E+07	0,6223	0,6109	0,154	0,2307	
8	29.07.2019	?	?	0	0	413,5342	26,4417	0,00E+00	0,6362	0,6278	0,10842	0,10842	
9	30.07.2019	0,6632	0,8337	934,1723	1,28E+03	450,0032	24,0774	6,48E+07	0,607	0,5958	0,0562	0,2379	
10	31.07.2019	0,3673	0,827	188,9145	485,9243	160,928	17,5694	7,42E+06	0,5113	0,5045	-0,144	0,3225	
11	01.08.2019	?	?	0	0	500,9451	20,1232	2,44E+07	0,676	0,6685	0,10842	0,10842	
12	02.08.2019	0,7007	0,8049	1,02E+03	1,36E+03	493,2808	20,8558	7,21E+07	0,6248	0,6139	0,0759	0,191	
13	03.08.2019	?	?	0	0	279,4534	20,1178	0,00E+00	0,576	0,5724	0,10842	0,10842	
14	04.08.2019	0,3326	0,5215	345,7857	653,0765	415,9342	18,7712	2,90E+07	0,644	0,6372	-0,3114	-0,1157	
15	05.08.2019	0,4554	0,5014	435,5653	586,0323	288,7551	18,4289	2,60E+07	0,6504	0,6395	-0,195	-0,1381	
16	06.08.2019	0,2681	0,3719	197,5304	300,3599	201,6366	18,2341	1,42E+07	0,6152	0,6131	-0,3471	-0,2412	
17	07.08.2019	0,6856	0,772	683,7006	972,7094	363,4836	19,5828	6,03E+07	0,6116	0,5989	0,074	0,1731	70819
18	08.08.2019	0,5181	0,5794	624,5892	854,7184	404,5336	19,722	3,73E+07	0,6301	0,6238	-0,112	-0,0444	
19	09.08.2019	0,7377	0,8694	853,9784	1,17E+03	433,0721	20,9403	6,85E+07	0,6217	0,6114	0,116	0,258	
20	10.08.2019	0,5524	0,7086	468,5227	709,5474	283,2729	21,3922	1,04E+07	0,6289	0,6114	0,0765	0,0973	

$$\eta = \eta_0 - a_1 T_m - a_2 T_m^2 I$$

$$\eta_T = \frac{\rho C_p Q \Delta T}{\cos(\beta_{az}) \cos(45° - \beta_{al}) I A (1 + \alpha)}$$

Main df

(403 rows × 9 columns)

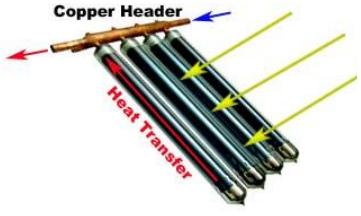
Train and Validation

(229 rows × 7 columns) × 2 collectors

24.07.2019
25.07.2019
26.07.2019
27.07.2019
28.07.2019

	Day	Month	Year	SunRad	Temp	TheorEff1	Eff1
0	23.0	7.0	19.0	429.3941	22.1619	0.6823	0.5802
1	24.0	7.0	19.0	473.2252	23.8903	0.6353	0.7896
2	25.0	7.0	19.0	466.4756	23.4041	0.6213	0.8016
3	26.0	7.0	19.0	467.5152	29.4243	0.6159	0.7721
4	27.0	7.0	19.0	367.7464	27.8901	0.6658	0.7672
...	...	...	...	...	...	...	...
224	1.0	9.0	20.0	287.2277	14.9155	0.6263	0.7633
225	3.0	9.0	20.0	83.2521	14.2076	0.5591	0.7982
226	7.0	9.0	20.0	103.3945	11.7907	0.7153	0.7091
227	9.0	9.0	20.0	165.4577	10.9654	0.6994	0.4034
228	11.0	9.0	20.0	34.0885	11.5876	0.6455	0.9058

Features | Target




```
Day        = 0 NaNs
Month      = 0 NaNs
Year       = 0 NaNs
SunRad     = 0 NaNs
Temp       = 0 NaNs
TheorEff1  = 0 NaNs
TheorEff2  = 0 NaNs
Eff1       = 0 NaNs
Eff2
```

Test

(174 rows × 7 columns) × 2 collectors

29.07.2019 ? 03.08.2019 ?

01.08.2019 ?

	Day	Month	Year	SunRad	Temp	TheorEff1	Eff1
0	29.0	7.0	19.0	413.5342	26.4417	0.6362	?
1	1.0	8.0	19.0	500.9451	20.1232	0.6760	
2	3.0	8.0	19.0	279.4534	20.1178	0.5760	
3	11.0	8.0	19.0	184.7988	17.6170	0.5828	
4	12.0	8.0	19.0	107.8485	14.9016	0.5377	
...	...	...	...	...	...	...	
169	13.0	9.0	20.0	93.7239	11.1725	0.7227	
170	14.0	9.0	20.0	64.6882	14.0804	0.7157	
171	15.0	9.0	20.0	45.3725	13.9169	0.6948	
172	16.0	9.0	20.0	305.1717	12.1398	0.6749	
173	17.0	9.0	20.0	226.1496	12.8769	0.7534	

Features | Target

Model Training

Skoltech

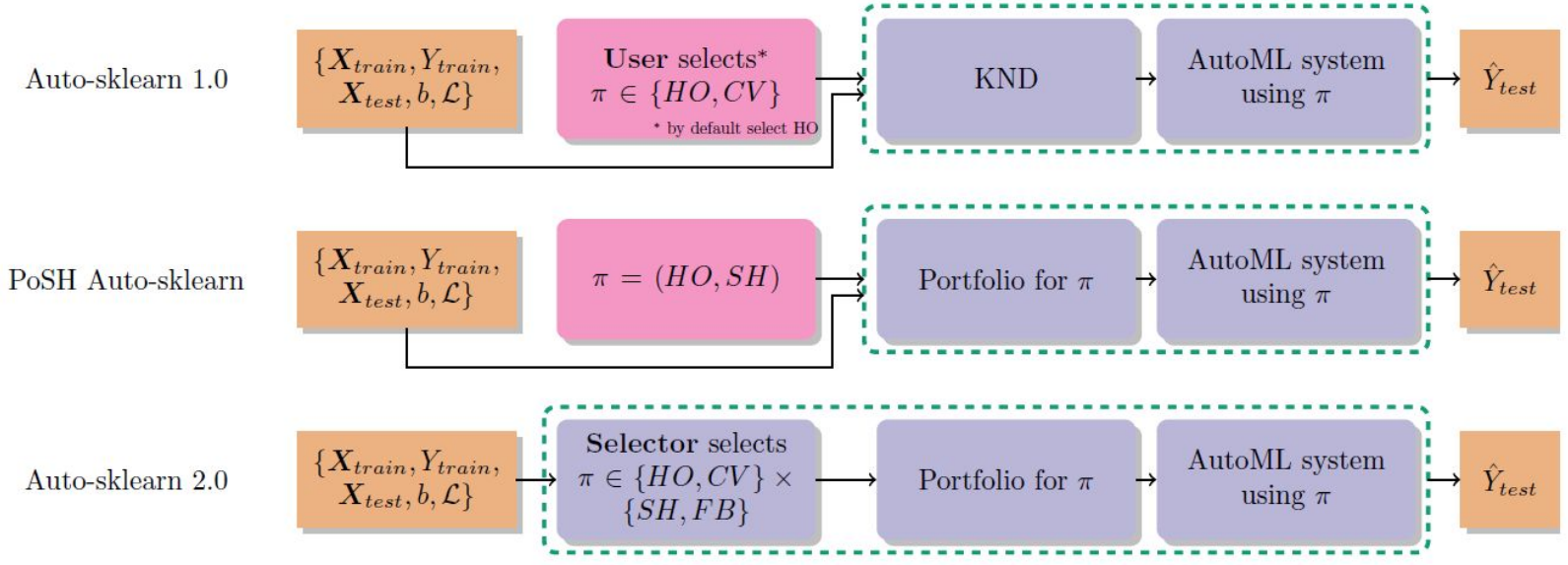
- LinearRegression
- Ridge
- SGDRegressor
- ElasticNet
- Lars
- HuberRegressor
- Lasso
- AdaBoostRegressor
- GradientBoostingRegressor
- RandomForestRegressor
- GaussianProcessRegressor
- SVR
- NuSVR
- DecisionTreeRegressor
- KNeighborsRegressor

Pretty much models of all possible kinds

scikit learn

For parameters searching **GridSearchCV** was used

Auto ML approach



Auto-WEKA (Thornton et al., 2013)

hyperoptsklearn (Komer et al., 2014)

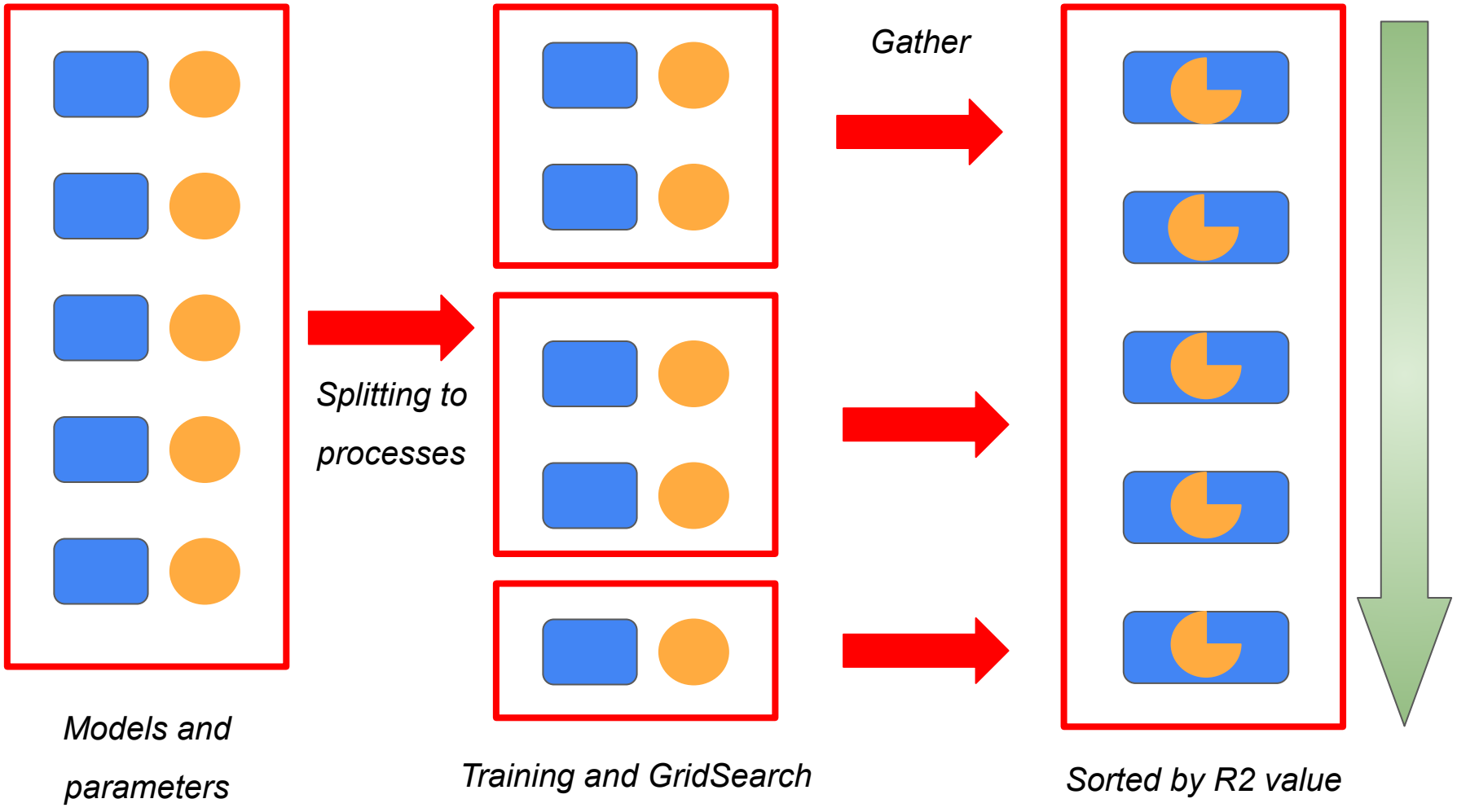
Auto-sklearn (Feurer et al., 2015a)

TPOT (Olson et al., 2016a)

Auto-Keras (Jin et al., 2019)

Auto-Sklearn 2.0 (Feurer et al., 2021)

Optimization of training procedure

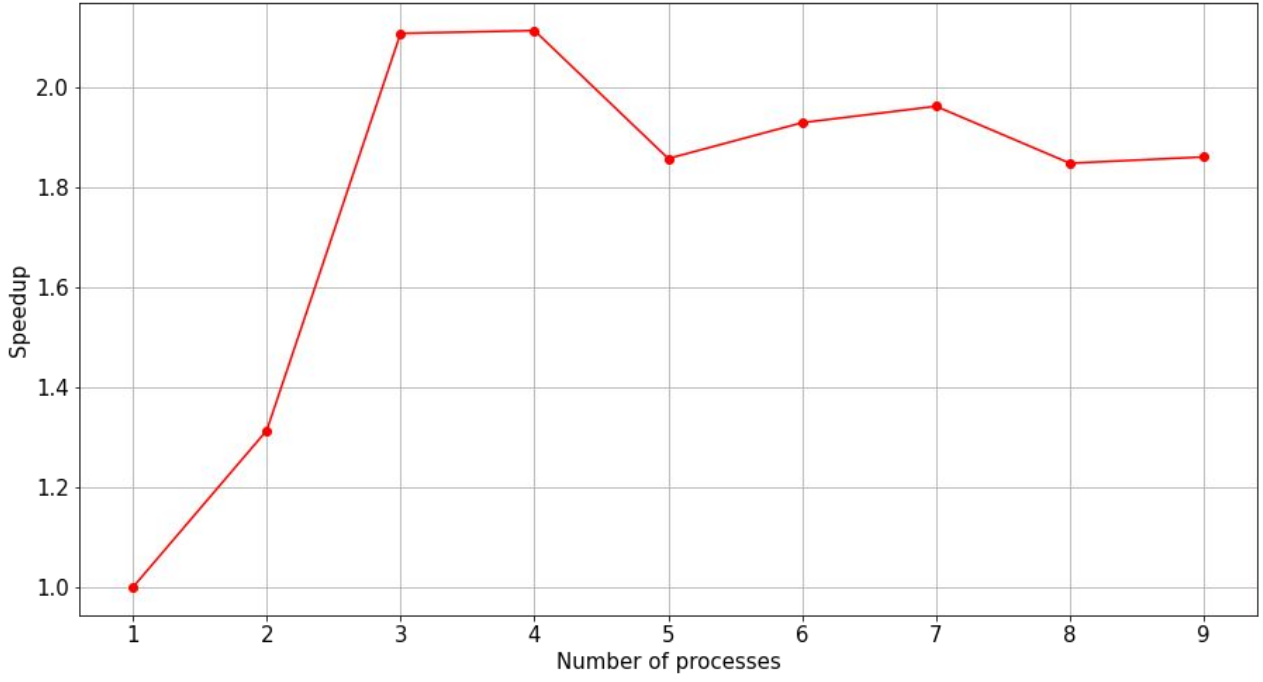


Optimization of training procedure

```
heavy_models = np.array([
    [DecisionTreeRegressor(random_state=42), {
        'max_depth':[1, 3, 5],
        'min_samples_split':[2,3],
        'max_features':['auto', 'log2']}],

    [RandomForestRegressor(random_state=42), {
        'n_estimators':[250, 500],
        'max_depth':[1, 3],
        'min_samples_split':[2,3],
        'max_features':['auto', 'log2']}],

    [AdaBoostRegressor(random_state=42), {
        'n_estimators':[100, 250, 500],
        'loss':['linear', 'square',],
        'learning_rate':[1.0, 0.1, 1.5],
    }],
    [ElasticNet(random_state=42), {
        'alpha':[1.0, 0.1, 1.5],
        'l1_ratio':[0.5, 0.25, 0.75],
    }],

    [Lasso(), {'alpha':[1.0, 0.1, 1.5]}],

    [Ridge(), {'alpha':[1.0, 0.1, 1.5]}],
])
```

Optimization of training procedure

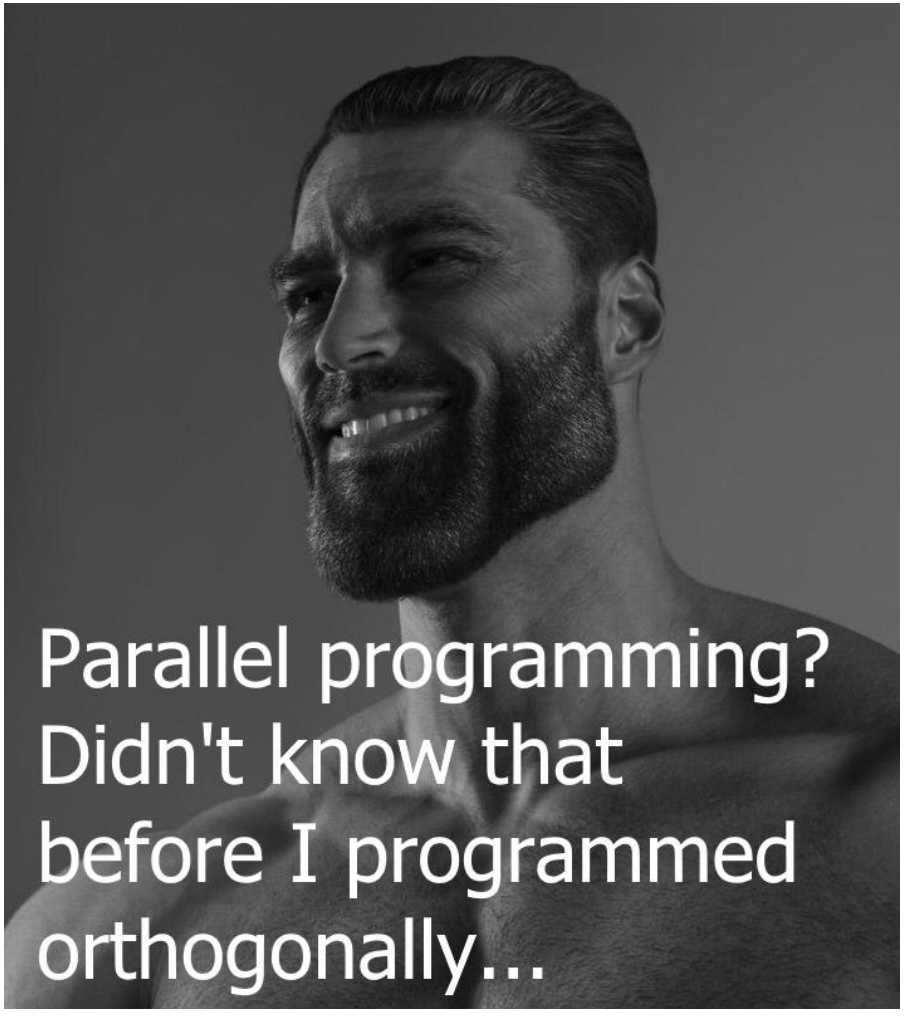
$$Speedup = \frac{T_{Serial}}{T_{Parallel}} = \frac{T(N=1)}{T(N)}$$




```
0.28763    Ridge()
0.001517   RandomForestRegressor(max_depth=3, n_estimators=500, random_state=42)
-0.014861  ElasticNet(random_state=42)
-0.014861  Lasso()
-0.12794   AdaBoostRegressor(learning_rate=0.1, n_estimators=100, random_state=42)
-0.36028   DecisionTreeRegressor(max_depth=3, max_features='auto', random_state=42)
```

Future Vision

- Implementation of more interactive solution
- Optimize parallelism
- Hyper parallelism
- Memory allocation issues



The end!

Thank you for attention

Oleg Nikolaev, 2nd year MSc student, PE

Nikita Kuznetsov, 1st year MSc student, ACS