Test task
ML modelling

Timoshenko Gala

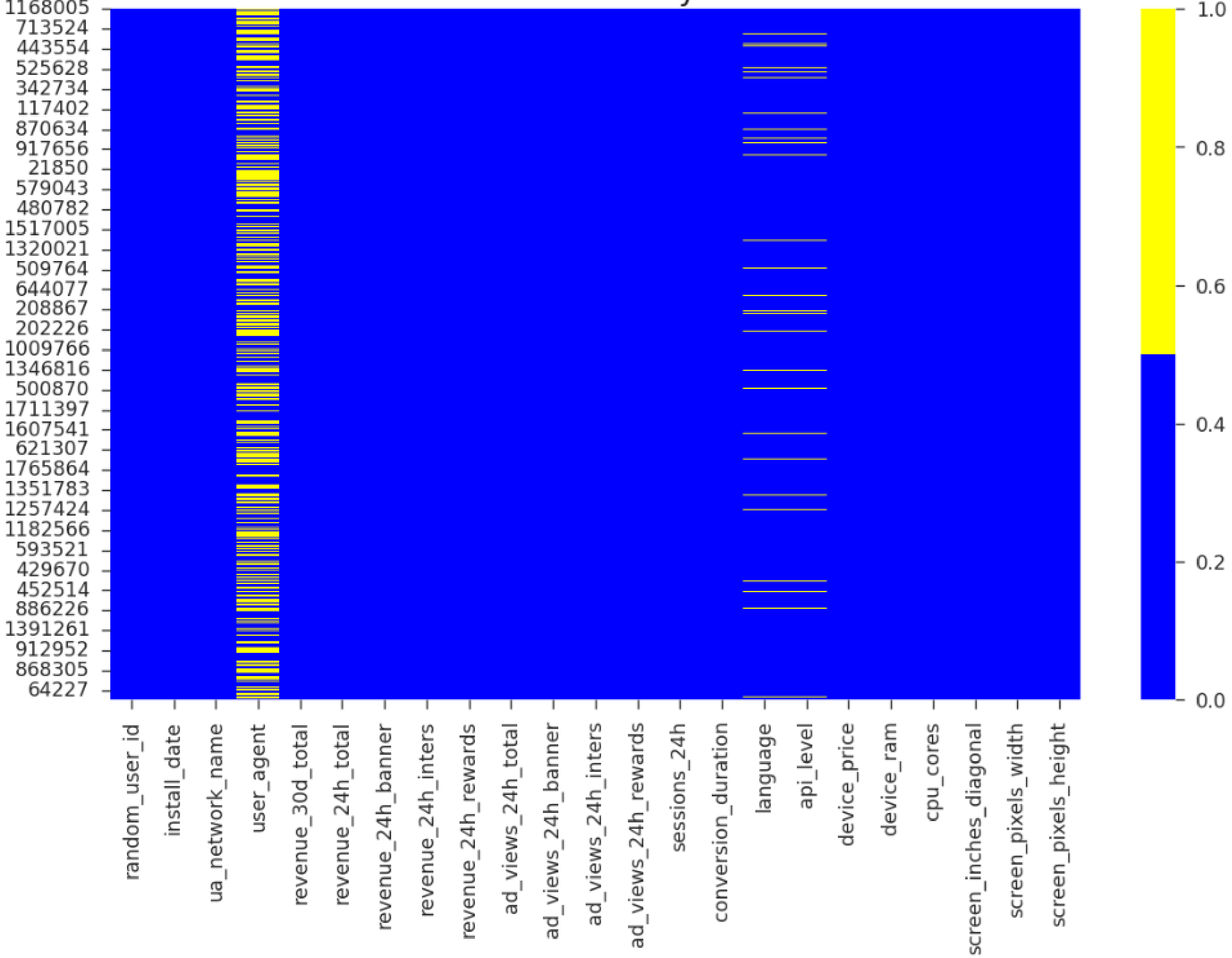
24.04.2023

Pipeline

- Step1.ipynb. EDA(cleaning data from Null, outliers, duplicates, making graphs, using statistics & correlation)
- Step2.ipynb.Baseline model. Linear Regression.
- Step3.ipynb. LGBMRegression
- Step4. Choose the best model.

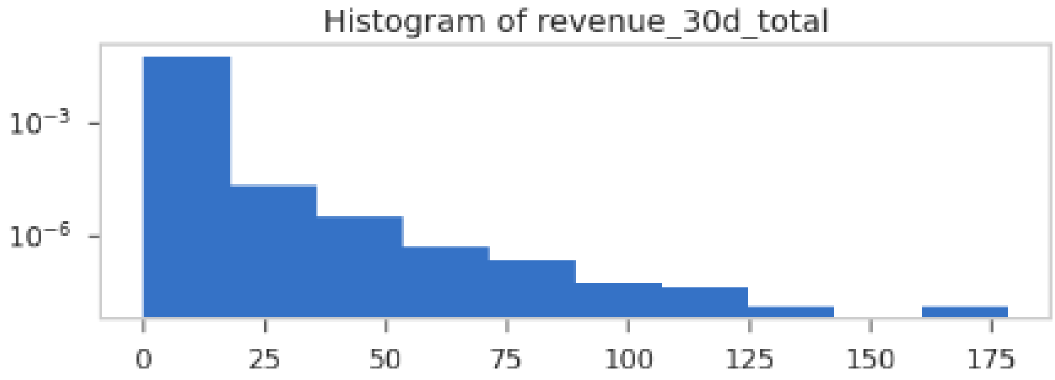
EDA

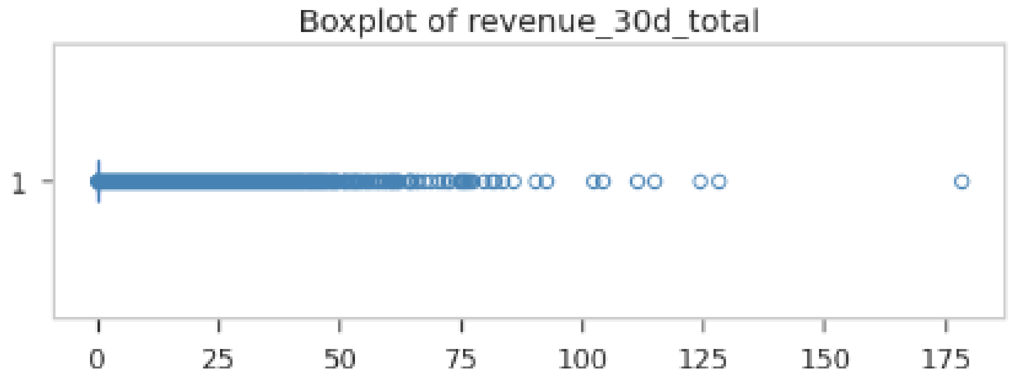
Missed values analysis test



name	% miss values	actions
api_level	2.81	Fill top freq value (30)
language	2.81	Fill top freq value (en)
user_agent	40.77	New feature OS (Android11-top) Linux only
country_code	100	Drop column

Revenue_30d_total target column


Histogram of revenue_30d_total


Boxplot of revenue_30d_total

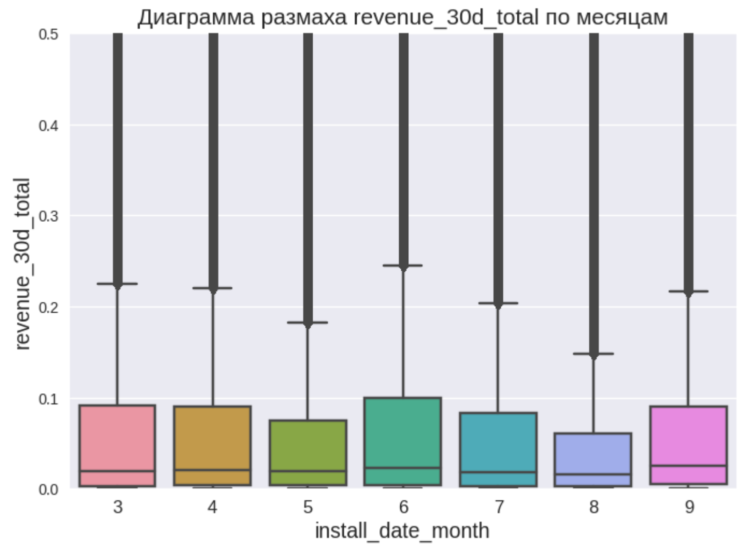


Диаграмма размаха revenue_30d_total по месяцам

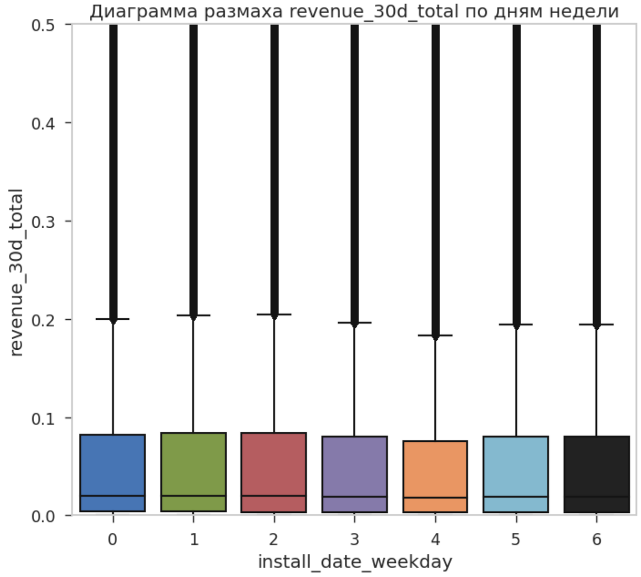
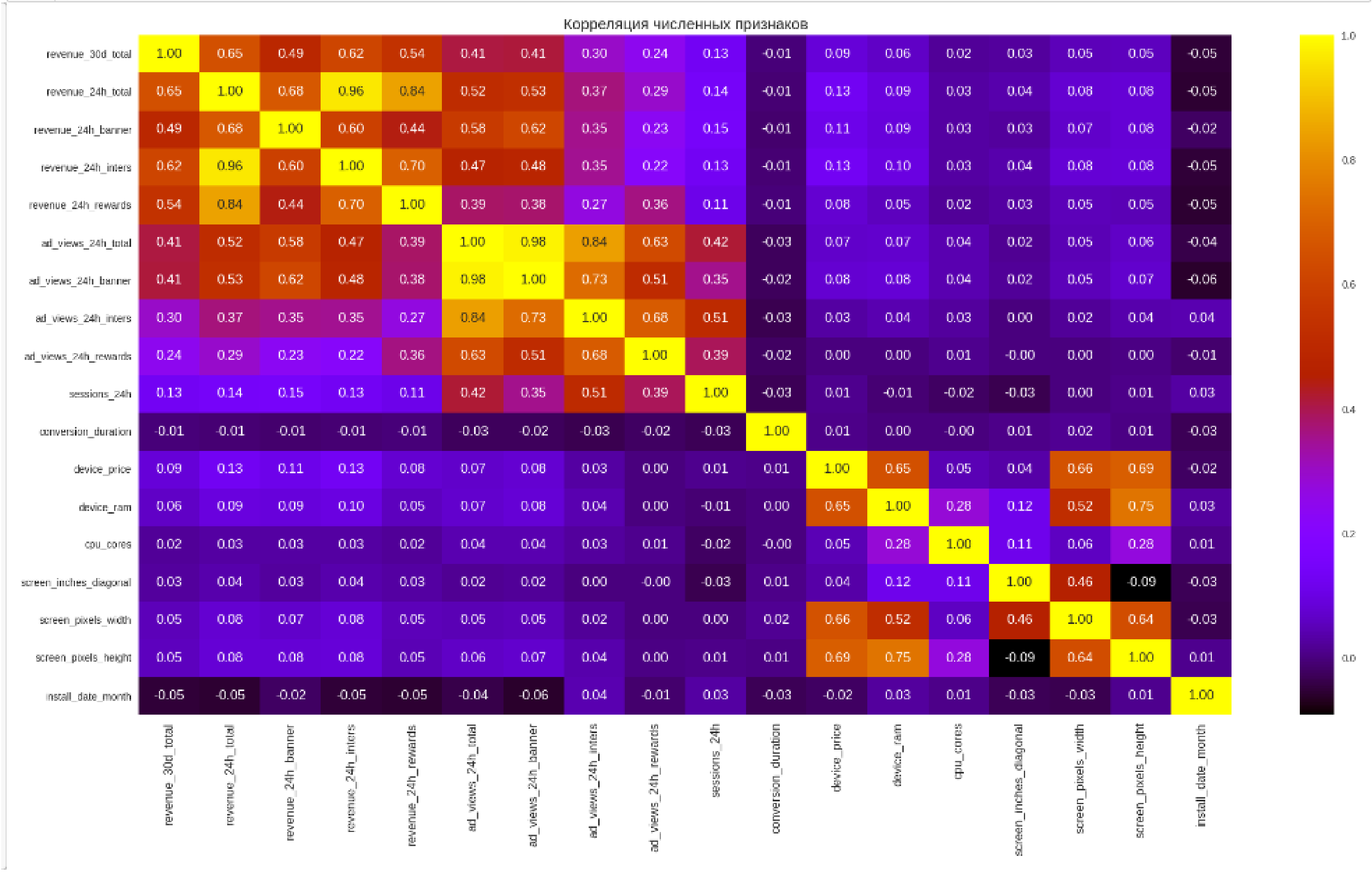


Диаграмма размаха revenue_30d_total по дням недели

Correlation



Корреляция численных признаков

	revenue_30d_total	revenue_24h_total	revenue_24h_banner	revenue_24h_inters	revenue_24h_rewards	ad_views_24h_total	ad_views_24h_banner	ad_views_24h_inters	ad_views_24h_rewards	sessions_24h	conversion_duration	device_price	device_ram	cpu_cores	screen_inches_diagonal	screen_pixels_width	screen_pixels_height	install_date_month
revenue_30d_total	1.00	0.65	0.49	0.62	0.54	0.41	0.41	0.30	0.24	0.13	-0.01	0.09	0.06	0.02	0.03	0.05	0.05	-0.05
revenue_24h_total	0.65	1.00	0.68	0.96	0.84	0.52	0.53	0.37	0.29	0.14	-0.01	0.13	0.09	0.03	0.04	0.08	0.08	-0.05
revenue_24h_banner	0.49	0.68	1.00	0.60	0.44	0.58	0.62	0.35	0.23	0.15	-0.01	0.11	0.09	0.03	0.03	0.07	0.08	-0.02
revenue_24h_inters	0.62	0.96	0.60	1.00	0.70	0.47	0.48	0.35	0.22	0.13	-0.01	0.13	0.10	0.03	0.04	0.08	0.08	-0.05
revenue_24h_rewards	0.54	0.84	0.44	0.70	1.00	0.39	0.38	0.27	0.36	0.11	-0.01	0.08	0.05	0.02	0.03	0.05	0.05	-0.05
ad_views_24h_total	0.41	0.52	0.58	0.47	0.39	1.00	0.98	0.84	0.63	0.42	-0.03	0.07	0.07	0.04	0.02	0.05	0.06	-0.04
ad_views_24h_banner	0.41	0.53	0.62	0.48	0.38	0.98	1.00	0.73	0.51	0.35	-0.02	0.08	0.08	0.04	0.02	0.05	0.07	-0.06
ad_views_24h_inters	0.30	0.37	0.35	0.35	0.27	0.84	0.73	1.00	0.68	0.51	-0.03	0.03	0.04	0.03	0.00	0.02	0.04	0.04
ad_views_24h_rewards	0.24	0.29	0.23	0.22	0.36	0.63	0.51	0.68	1.00	0.39	-0.02	0.00	0.00	0.01	-0.00	0.00	0.00	-0.01
sessions_24h	0.13	0.14	0.15	0.13	0.11	0.42	0.35	0.51	0.39	1.00	-0.03	0.01	-0.01	-0.02	-0.03	0.00	0.01	0.03
conversion_duration	-0.01	-0.01	-0.01	-0.01	-0.01	-0.03	-0.02	-0.03	-0.02	-0.03	1.00	0.01	0.00	-0.00	0.01	0.02	0.01	-0.03
device_price	0.09	0.13	0.11	0.13	0.08	0.07	0.08	0.03	0.00	0.01	0.01	1.00	0.65	0.05	0.04	0.66	0.69	-0.02
device_ram	0.06	0.09	0.09	0.10	0.05	0.07	0.08	0.04	0.00	-0.01	0.00	0.65	1.00	0.28	0.12	0.52	0.75	0.03
cpu_cores	0.02	0.03	0.03	0.03	0.02	0.04	0.04	0.03	0.01	-0.02	-0.00	0.05	0.28	1.00	0.11	0.06	0.28	0.01
screen_inches_diagonal	0.03	0.04	0.03	0.04	0.03	0.02	0.02	0.00	-0.00	-0.03	0.01	0.04	0.12	0.11	1.00	0.46	-0.09	-0.03
screen_pixels_width	0.05	0.08	0.07	0.08	0.05	0.05	0.05	0.02	0.00	0.00	0.02	0.66	0.52	0.06	0.46	1.00	0.64	-0.03
screen_pixels_height	0.05	0.08	0.08	0.08	0.05	0.06	0.07	0.04	0.00	0.01	0.01	0.69	0.75	0.28	-0.09	0.64	1.00	0.01
install_date_month	-0.05	-0.05	-0.02	-0.05	-0.05	-0.04	-0.06	0.04	-0.01	0.03	-0.03	-0.02	0.03	0.01	-0.03	-0.03	0.01	1.00

Not bad linear correlation

- revenue_… and ad_views_…
- Device_ram, device_price, screen_height

Feature generation

1. install_date_month
2. date_day
3. user_agent - Android OS type
4. install_date_weekday
5. screen_inches_diagonal
6. screen_pixels_width
7. screen_pixels_height
8. revenue_30d_total_median_per_os
9. revenue_30d_total_median_per_lan
10. square_number columns
11. sqrt_number columns
12. log_number columns

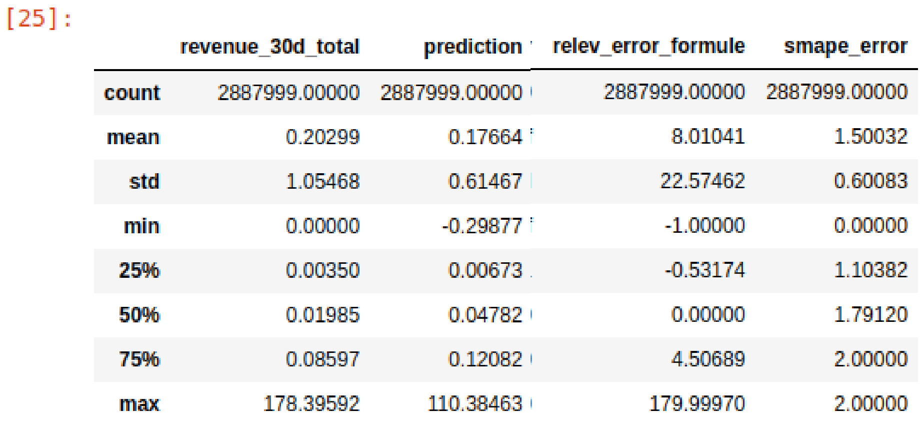
Feature std_scaler

1. revenue_24h_rewards
2. revenue_24h_total
3. revenue_24h_banner
4. revenue_24h_inters
5. api_level
6. sessions_24h
7. screen_inches_diagonal
8. ad_views_24h_reward
9. ad_views_24h_total
10. device_price
11. screen_pixels_width
12. conversion_duration

LinearRegression model

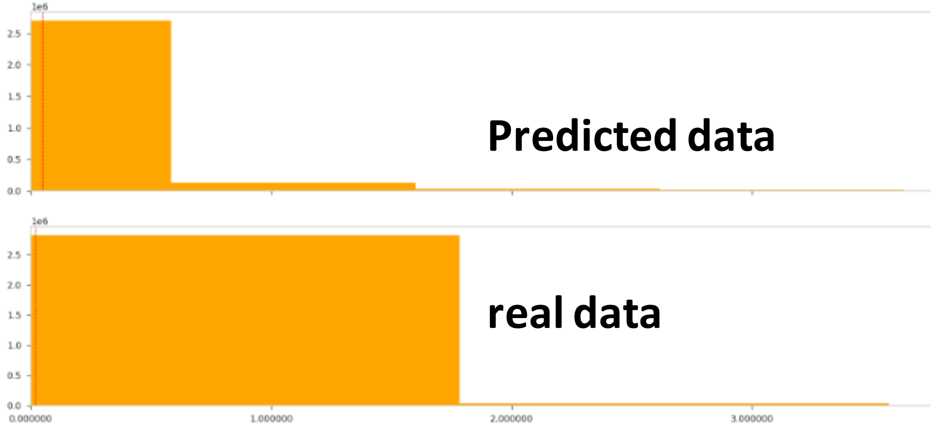
Train: 0.4348, Test: 0.4399
Crossval [0.4201; 0.4328; 0.4273; 0.4726]

Train.csv



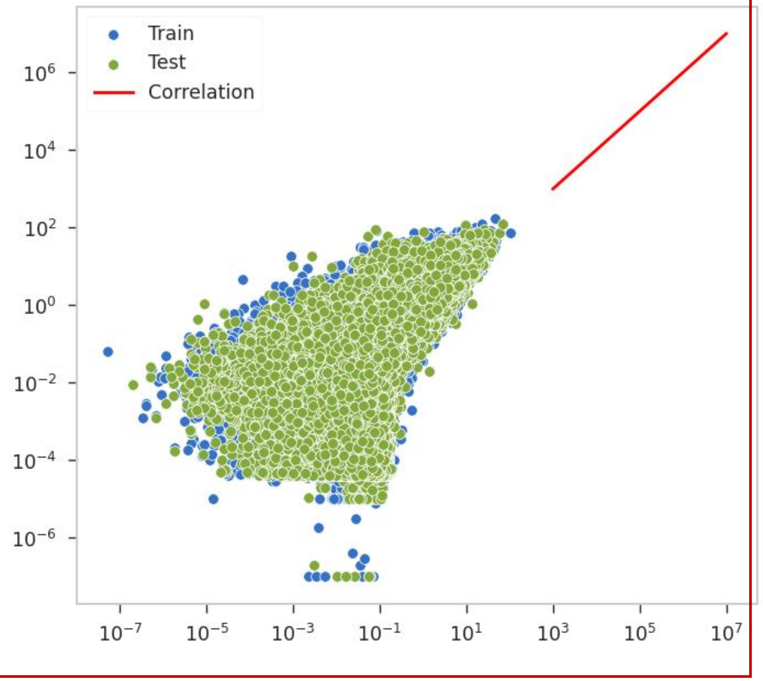
[25]:

	revenue_30d_total	prediction	relev_error_formule	smape_error
count	2887999.00000	2887999.00000	2887999.00000	2887999.00000
mean	0.20299	0.17664	8.01041	1.50032
std	1.05468	0.61467	22.57462	0.60083
min	0.00000	-0.29877	-1.00000	0.00000
25%	0.00350	0.00673	-0.53174	1.10382
50%	0.01985	0.04782	0.00000	1.79120
75%	0.08597	0.12082	4.50689	2.00000
max	178.39592	110.38463	179.99970	2.00000

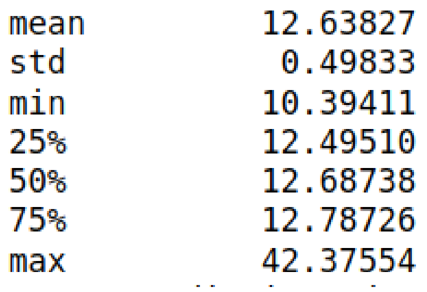


Predicted data

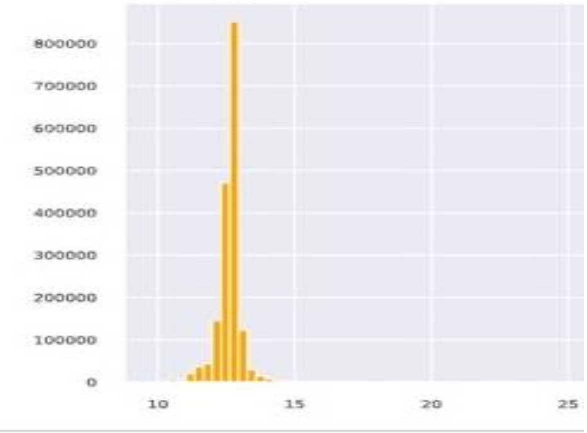
real data



Test.csv



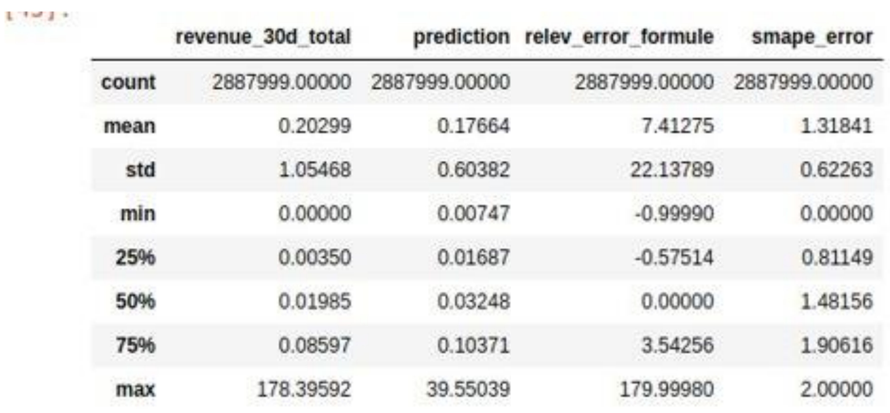
```
mean          12.63827
std            0.49833
min           10.39411
25%           12.49510
50%           12.68738
75%           12.78726
max           42.37554
```



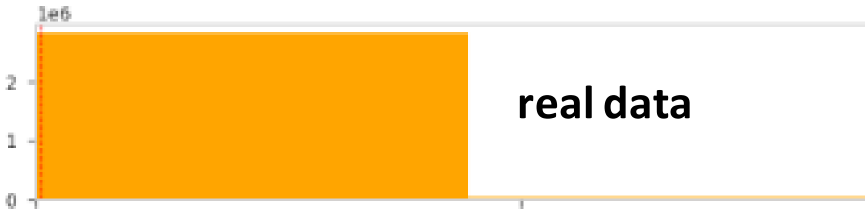
LGBMRegression model

train: 0.508991 test: 0.527053
train: 0.481179 test: 0.502642
train: 0.471994 test: 0.500058
train: 0.466292 test: 0.500101

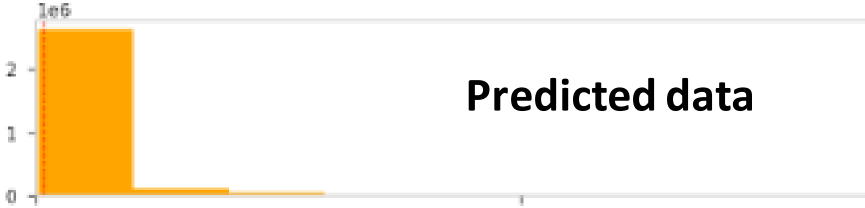
Train.csv



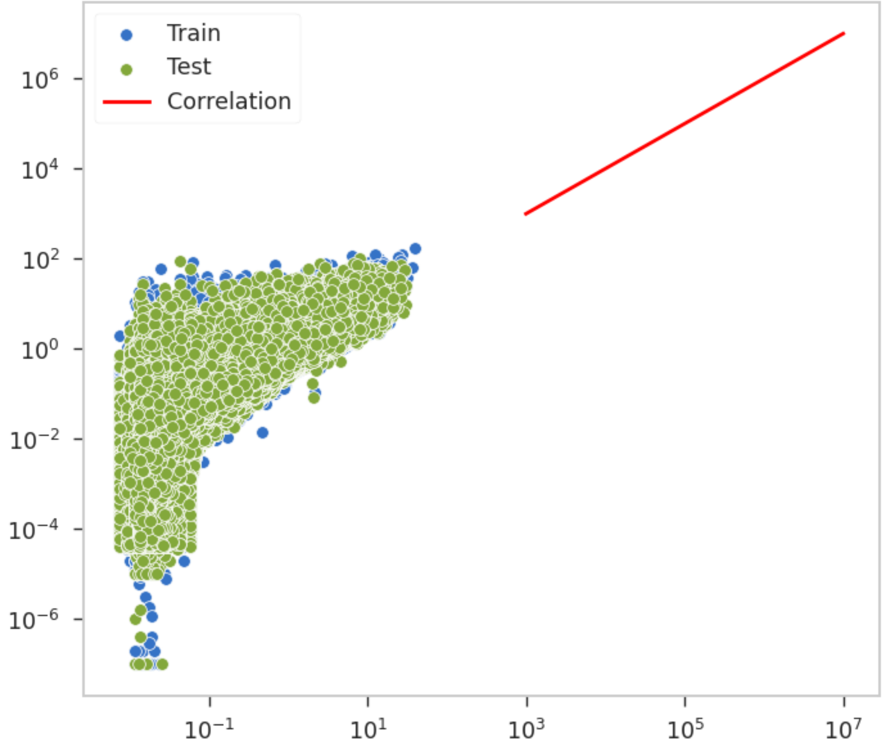
	revenue_30d_total	prediction	relev_error_formule	smape_error
count	2887999.00000	2887999.00000	2887999.00000	2887999.00000
mean	0.20299	0.17664	7.41275	1.31841
std	1.05468	0.60382	22.13789	0.62263
min	0.00000	0.00747	-0.99990	0.00000
25%	0.00350	0.01687	-0.57514	0.81149
50%	0.01985	0.03248	0.00000	1.48156
75%	0.08597	0.10371	3.54256	1.90616
max	178.39592	39.55039	179.99980	2.00000



real data

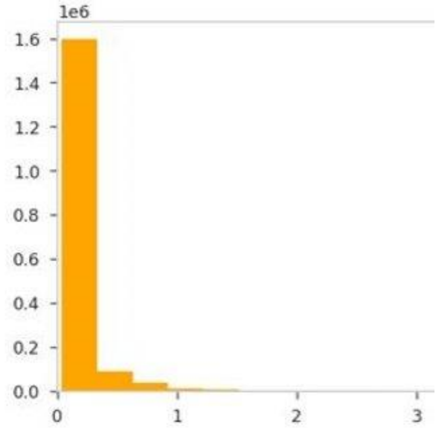


Predicted data

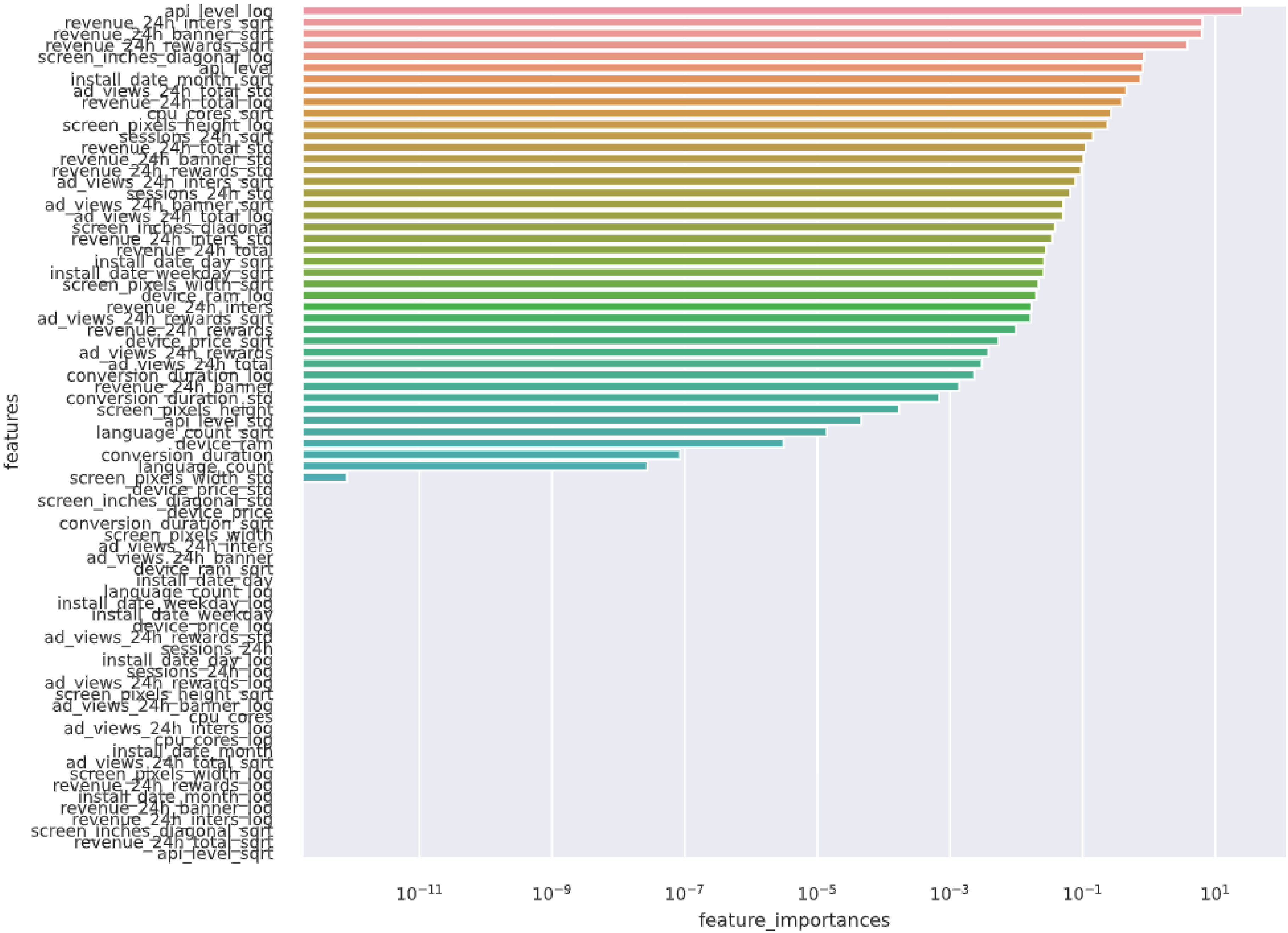


Test.csv

```
mean           0.17228
std            0.41612
min            0.03277
25%            0.06011
50%            0.07221
75%            0.13221
max           29.75212
```



Feature importance



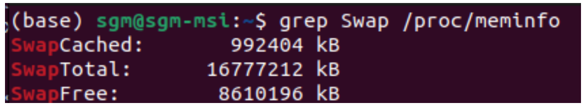
Choose the best model

Linear regression	LGBMRegression
bad in prediction 0 min-values	bad in prediction max-values

Set metric Rel_err can occur inf values (division to 0), better to choose SMAPE=0.6,
Rel_err_std=20%
	SMAPE=0.6,
	Rel_err_std=22%

Predicted range on test[min=10; max=42]	Predicted range on test[min=0.03; max=29]
Predicted range on train[min=0; max=110]	Predicted range on train[min=0.07; max=39]
Cross_val Train: 0.4348, Test: 0.4399	Cross_val train: 0.508991 test: 0.527053

- Choosing between Linear regression & LGBMRegression is hard, cause the value-range is diff in both case.
- I've tried to run XGBoost, CatBoost, MLP (NeuralNetwor) but Kernel was dead quickly, I swapped all the memory, but it didn’t work



```
(base) sgm@sgm-msi:~$ grep Swap /proc/meminfo
SwapCached:       992404 kB
SwapTotal:      16777212 kB
SwapFree:        8610196 kB
```

- If you could provide computing power I'll try extra runs
- Rel_err is smaller in Linear Regr, but I'm confused about the set range in test.csv
- So I suggest to choose LGBMRegression at the moment and continue on looking the best version