Advertisement CTR Prediction

Shortlist Promising Models

DS5220 / Fall 2023 Semester

Team Members: Liyang Song, Qian Yin

Dec 10, 2023

Introduction

In this part, we trained five different models using standard parameters.

- **Goal:**
  - Minimize false positive instances (Type I error) to lower the rate of sending advertisements to users who are not interested
- **Five models used:**
  - SGD classifier
  - Random forest classifier
  - Decision tree classifier
  - Adaboost classifier
  - Gradient boosting classifier
- **Steps for each model:**
  - Define default model
  - Evaluate model performance on train set
  - Evaluate model performance on validation set
  - Drop the least important attributes based on permutation importance
  - Re-evaluate the model performance on validation set
  - Update the attributes that need to be dropped

Classifier defined

- **SGD Classifier**

```
SGDClassifier(class_weight='balanced', loss='log_loss', max_iter=10000,
              random_state=42)
```

- **Ada Boost Classifier**

```
AdaBoostClassifier(estimator=DecisionTreeClassifier(class_weight='balance
d',
                                                    criterion='log_loss',
                                                    random_state=42),
                   random_state=42)
```

- **Decision Tree Classifier**

```
DecisionTreeClassifier(class_weight='balanced', criterion='log_loss',
                       random_state=42)
```

- **Gradient Boosting Classifier**

```
GradientBoostingClassifier(random_state=42)
```

- **Random forest classifier**

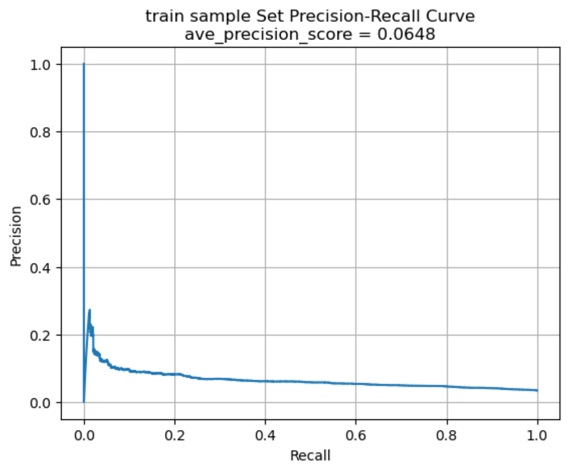
```
DecisionTreeClassifier(class_weight='balanced', random_state=42)
```

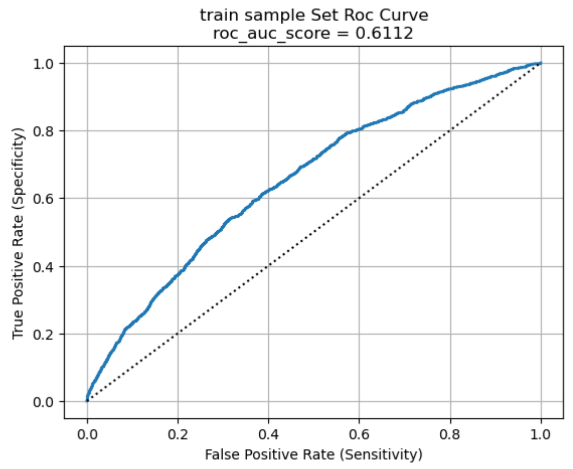
Note: Because the distribution (0/1) is imbalanced, we set ‘class_weight=balanced’ to adjust the weights of the classes.

SGD classifier – Performance on train

```
Check classification report
{'0': {'precision': 0.976314872866597, 'recall': 0.6928900287365201, 'f1-score': 0.8105402035025573, 'support': 32363.0},
'1': {'precision': 0.05800398066533978, 'recall': 0.5294117647058824, 'f1-score': 0.10455283163919024, 'support': 1156.
0}, 'accuracy': 0.687252006324771, 'macro avg': {'precision': 0.5171594267659684, 'recall': 0.6111508967212012, 'f1-scor
e': 0.4575465175708738, 'support': 33519.0}, 'weighted avg': {'precision': 0.9446442564584507, 'recall': 0.68725200632477
1, 'f1-score': 0.7861921799375926, 'support': 33519.0}}
```

```
Check confusion matrix
train sample set confusion matrix:
[[22424  9939]
 [  544   612]]
True Positives =  22424
True Negatives =  612
False Positives(Type I error) =  9939
False Negatives(Type II error) =  544
```



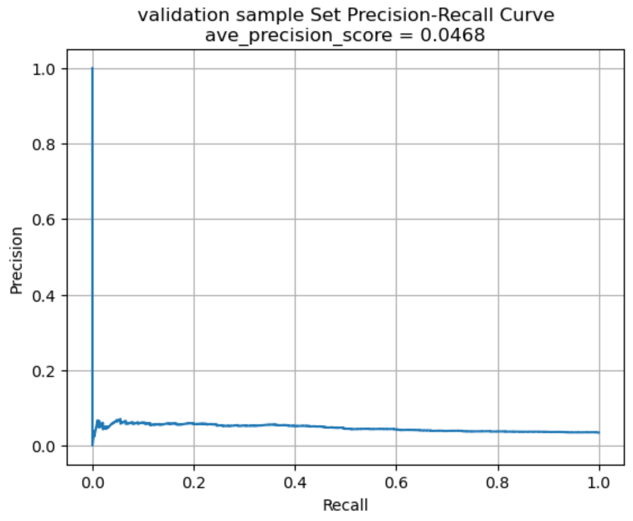


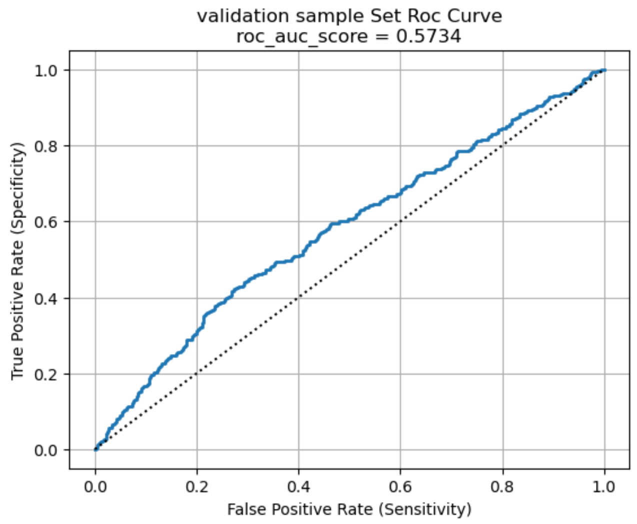
- The low **precision score** and the large number of **false positive** instances indicate that the model is not performing well in correctly identifying positive instances. The **average precision score** is very low.

SGD classifier – Performance on validation

```
Check classification report
{'0': {'precision': 0.9726169844020798, 'recall': 0.6935244686109738, 'f1-score': 0.8096955706247295, 'support': 8092.0},
'1': {'precision': 0.05017234775947912 5, 'recall': 0.4532871972318339, 'f1-score': 0.09034482758620688, 'support': 289.
0}, 'accuracy': 0.6852404247703138, 'macro avg': {'precision': 0.5113946660807794, 'recall': 0.5734058329214039, 'f1-scor
e': 0.4500201991054682, 'support': 8381.0}, 'weighted avg': {'precision': 0.9408085486557831, 'recall': 0.685240424770313
8, 'f1-score': 0.7848903725889185, 'support': 8381.0}}
```

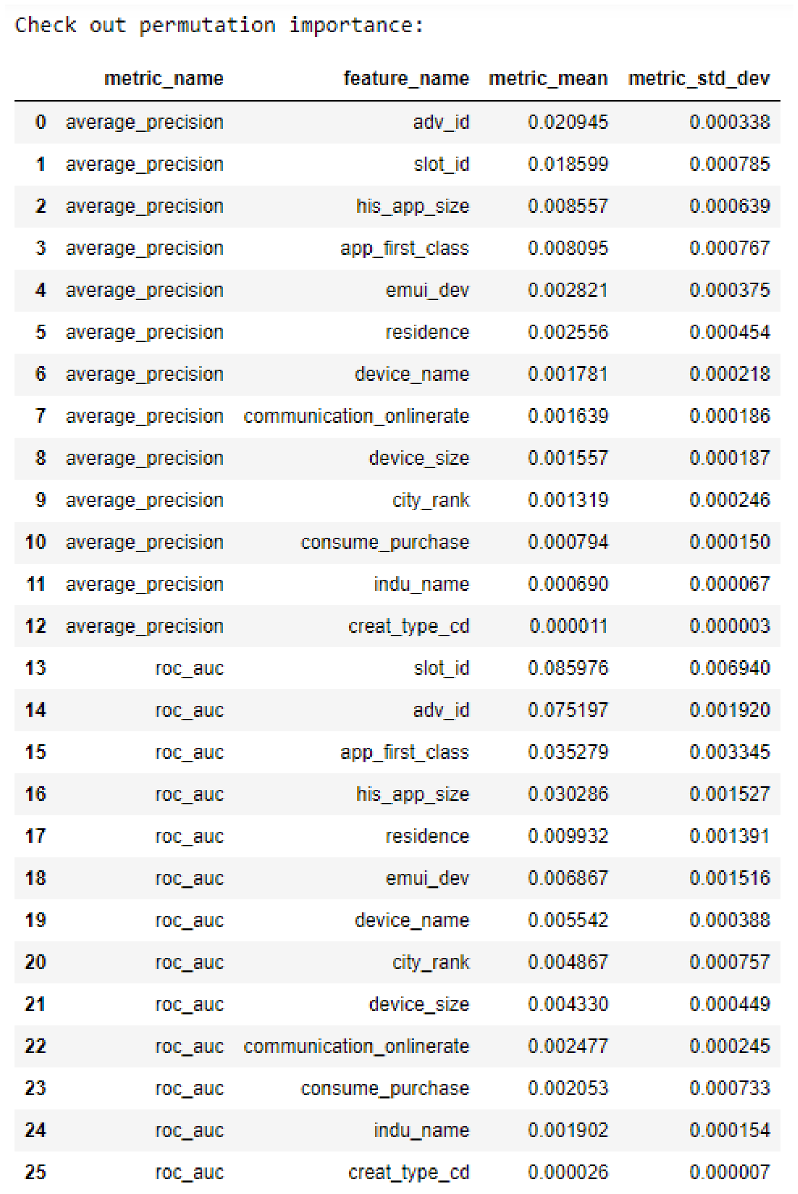
```
Check confusion matrix
validation sample set confusion matrix:
[[5612 2480]
 [ 158  131]]
True Positives =  5612
True Negatives =  131
False Positives(Type I error) =  2480
False Negatives(Type II error) =  158
```





- The performances on train set and validation set are similar. The **average precision score** is slightly lower on validation set.

SGD classifier – Feature selection



```
Check out permutation importance:
```

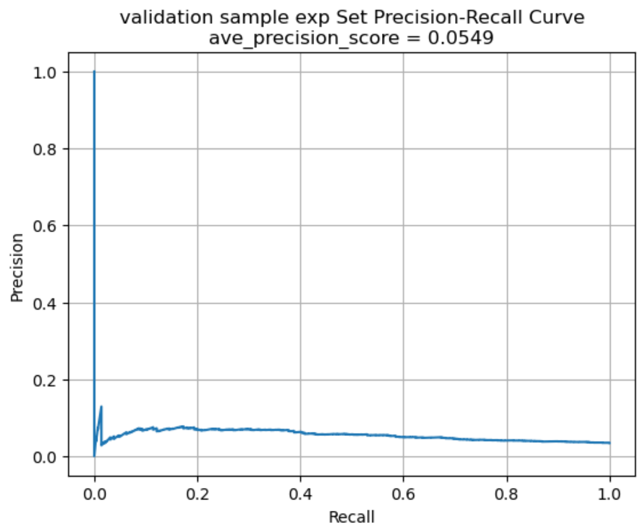
	metric_name	feature_name	metric_mean	metric_std_dev
0	average_precision	adv_id	0.020945	0.000338
1	average_precision	slot_id	0.018599	0.000785
2	average_precision	his_app_size	0.008557	0.000639
3	average_precision	app_first_class	0.008095	0.000767
4	average_precision	emui_dev	0.002821	0.000375
5	average_precision	residence	0.002556	0.000454
6	average_precision	device_name	0.001781	0.000218
7	average_precision	communication_onlinerate	0.001639	0.000186
8	average_precision	device_size	0.001557	0.000187
9	average_precision	city_rank	0.001319	0.000246
10	average_precision	consume_purchase	0.000794	0.000150
11	average_precision	indu_name	0.000690	0.000067
12	average_precision	creat_type_cd	0.000011	0.000003
13	roc_auc	slot_id	0.085976	0.006940
14	roc_auc	adv_id	0.075197	0.001920
15	roc_auc	app_first_class	0.035279	0.003345
16	roc_auc	his_app_size	0.030286	0.001527
17	roc_auc	residence	0.009932	0.001391
18	roc_auc	emui_dev	0.006867	0.001516
19	roc_auc	device_name	0.005542	0.000388
20	roc_auc	city_rank	0.004867	0.000757
21	roc_auc	device_size	0.004330	0.000449
22	roc_auc	communication_onlinerate	0.002477	0.000245
23	roc_auc	consume_purchase	0.002053	0.000733
24	roc_auc	indu_name	0.001902	0.000154
25	roc_auc	creat_type_cd	0.000026	0.000007

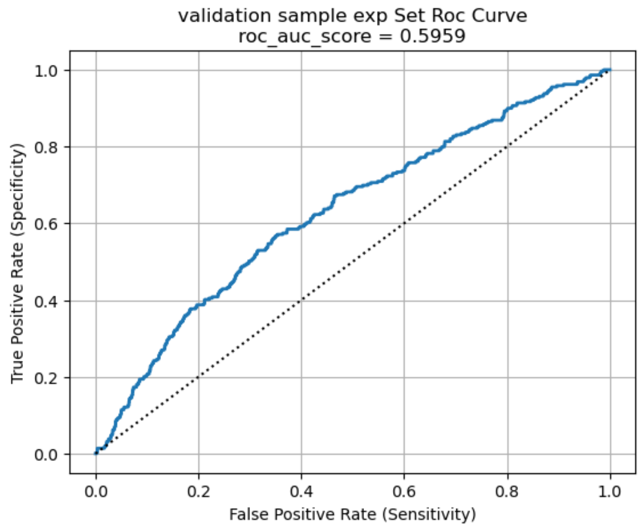
- We use **average_precision** and **roc_auc** to list the most significant features.
- To improve the performance, we conduct a feature selection. Below are additional features that might be dropped
  - 'indu_name',
  - 'creat_type_cd',
  - 'emui_dev',
  - 'residence',
  - 'city_rank',
  - 'communication_onlinerate',
  - 'device_name',
  - 'consume_purchase',
  - 'device_size'

SGD classifier – Performance after feature selection (v)

```
Check classification report
{'0': {'precision': 0.9771590167500543, 'recall': 0.5551161641127039, 'f1-score': 0.7080148159823468, 'support': 8092.0},
'1': {'precision': 0.048625792811839326, 'recall': 0.6366782006920415, 'f1-score': 0.09035109256076602, 'support': 289.0},
'accuracy': 0.5579286481326811, 'macro avg': {'precision': 0.5128924047809469, 'recall': 0.5958971824023727, 'f1-score': 0.3
991829542715564, 'support': 8381.0}, 'weighted avg': {'precision': 0.9451406297177021, 'recall': 0.5579286481326811, 'f1-sco
re': 0.686716066898844, 'support': 8381.0}}
```

```
Check confusion matrix
validation sample exp set confusion matrix:
[[4492 3600]
 [ 105  184]]
True Positives =  4492
True Negatives =  184
False Positives(Type I error) =  3600
False Negatives(Type II error) =  105
```



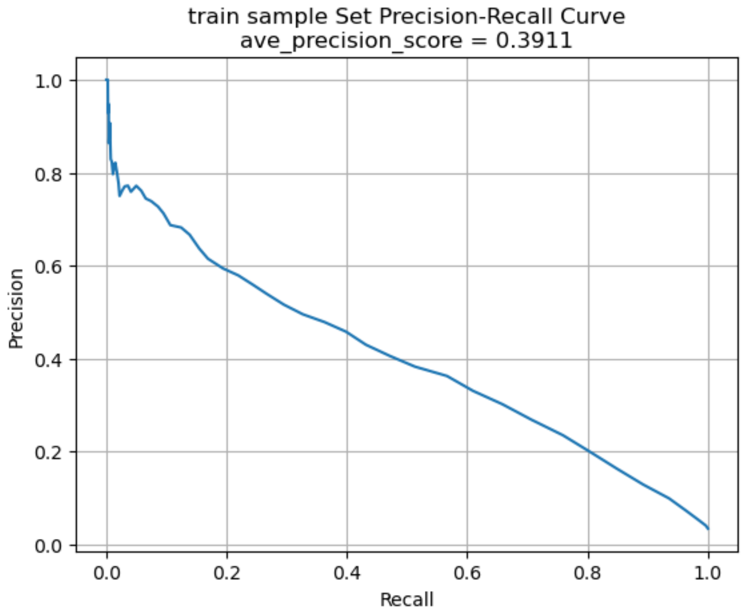


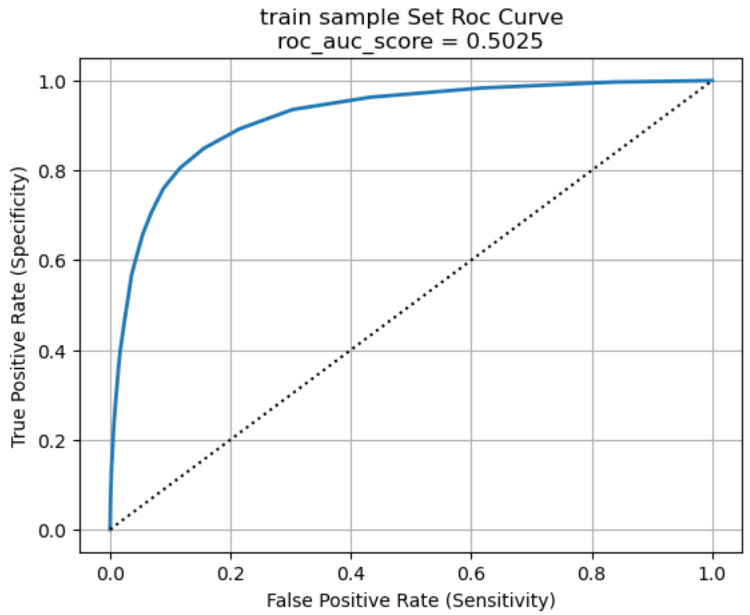
- Above is the performance after dropping those additional features on the validation set. The number of **false positive** instances increases greatly; therefore, we won't drop features at this step.

Random forest classifier – Performance on train

```
Check classification report
{'0': {'precision': 0.965660366090358, 'recall': 0.9999768087261033, 'f1-score': 0.982519036135427, 'support': 129359.0},
'1': {'precision': 0.8846153846153846, 'recall': 0.004975124378109453, 'f1-score': 0.009894600989460099, 'support': 4623.
0}, 'accuracy': 0.965644638832082, 'macro avg': {'precision': 0.9251378753528713, 'recall': 0.5024759665521064, 'f1-scor
e': 0.49620681856244353, 'support': 133982.0}, 'weighted avg': {'precision': 0.9628639385899563, 'recall': 0.965644638832
082, 'f1-score': 0.9489589850563283, 'support': 133982.0}}
```

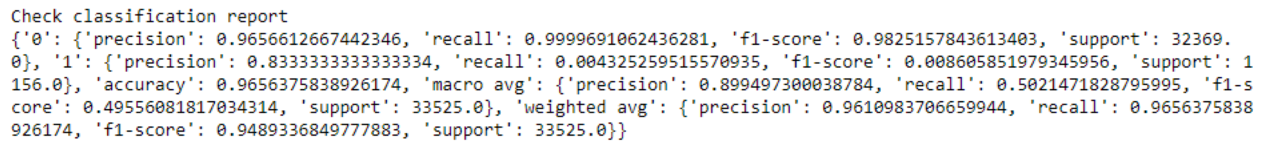
```
Check confusion matrix
train sample set confusion matrix:
[[129356      3]
 [  4600     23]]
True Positives =  129356
True Negatives =  23
False Positives(Type I error) =  3
False Negatives(Type II error) =  4600
```



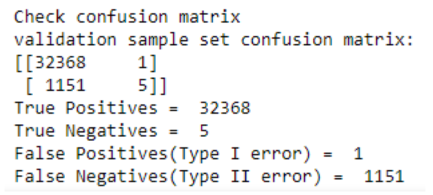


- This model has few **false positive** although the number of **false negatives** is large.

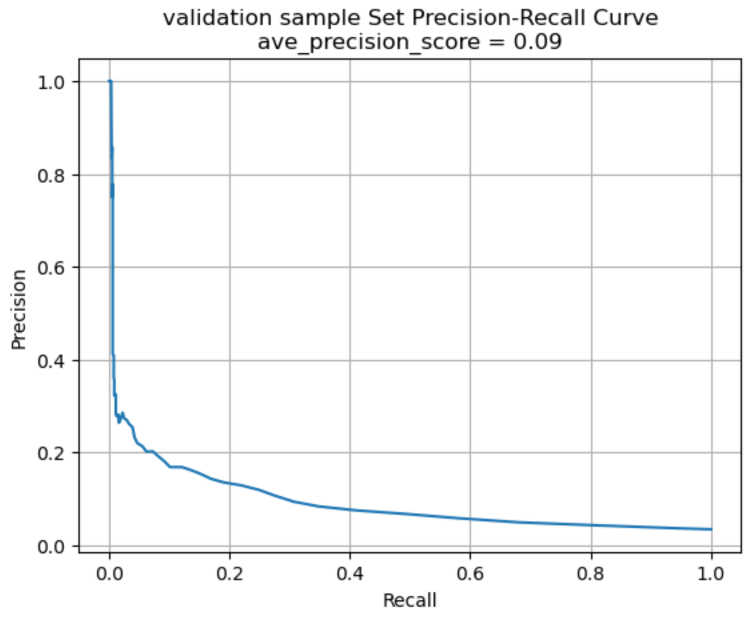
Random forest classifier – Performance on validation

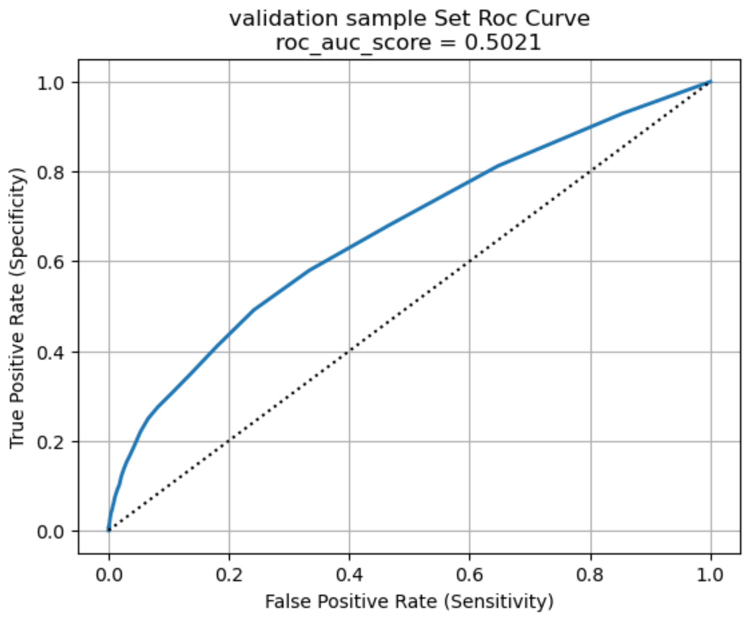


```
Check classification report
{'0': {'precision': 0.9656612667442346, 'recall': 0.9999691062436281, 'f1-score': 0.9825157843613403, 'support': 32369.
0}, '1': {'precision': 0.8333333333333334, 'recall': 0.004325259515570935, 'f1-score': 0.008605851979345956, 'support': 1
156.0}, 'accuracy': 0.9656375838926174, 'macro avg': {'precision': 0.899497300038784, 'recall': 0.5021471828795995, 'f1-s
core': 0.49556081817034314, 'support': 33525.0}, 'weighted avg': {'precision': 0.9610983706659944, 'recall': 0.9656375838
926174, 'f1-score': 0.9489336849777883, 'support': 33525.0}}
```



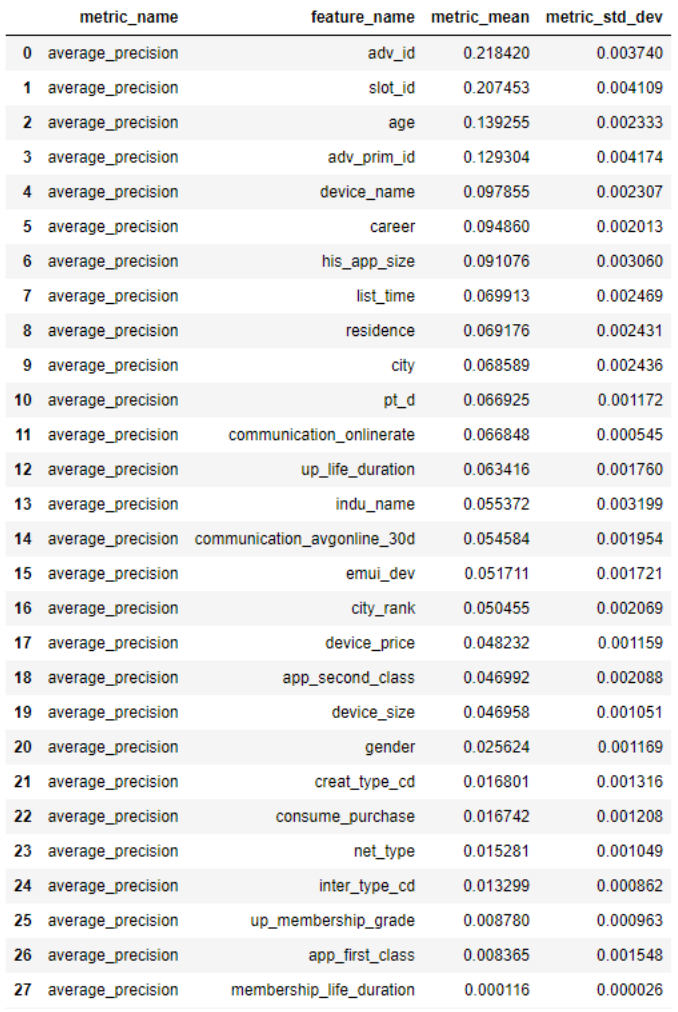
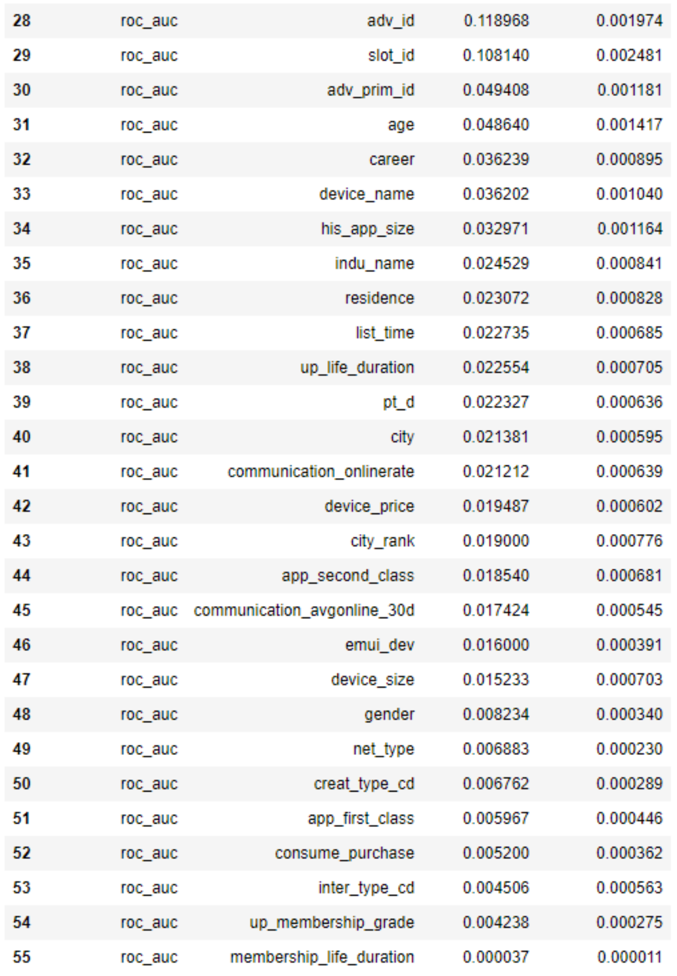
```
Check confusion matrix
validation sample set confusion matrix:
[[32368     1]
 [ 1151     5]]
True Positives =  32368
True Negatives =  5
False Positives(Type I error) =  1
False Negatives(Type II error) =  1151
```





- Compared to the performance on the train set, the **ave_precision_score** decreases significantly

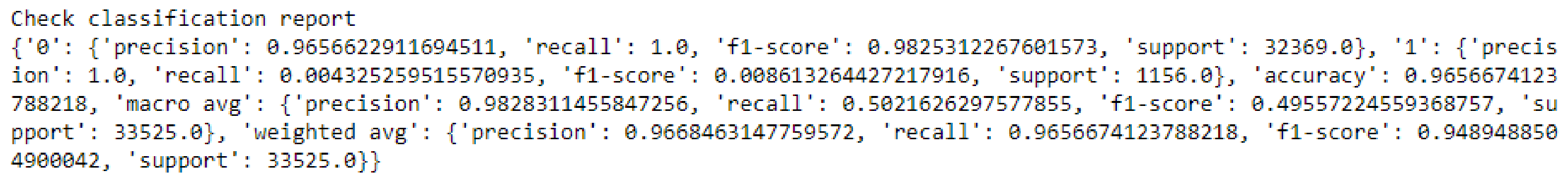
Random forest classifier – Feature selection

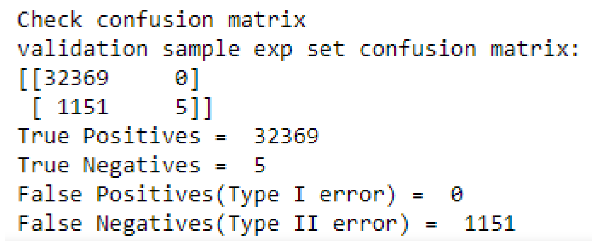
	metric_name	feature_name	metric_mean	metric_std_dev
0	average_precision	adv_id	0.218420	0.003740
1	average_precision	slot_id	0.207453	0.004109
2	average_precision	age	0.139255	0.002333
3	average_precision	adv_prim_id	0.129304	0.004174
4	average_precision	device_name	0.097855	0.002307
5	average_precision	career	0.094860	0.002013
6	average_precision	his_app_size	0.091076	0.003060
7	average_precision	list_time	0.069913	0.002469
8	average_precision	residence	0.069176	0.002431
9	average_precision	city	0.068589	0.002436
10	average_precision	pt_d	0.066925	0.001172
11	average_precision	communication_onlinerate	0.066848	0.000545
12	average_precision	up_life_duration	0.063416	0.001760
13	average_precision	indu_name	0.055372	0.003199
14	average_precision	communication_avgonline_30d	0.054584	0.001954
15	average_precision	emui_dev	0.051711	0.001721
16	average_precision	city_rank	0.050455	0.002069
17	average_precision	device_price	0.048232	0.001159
18	average_precision	app_second_class	0.046992	0.002088
19	average_precision	device_size	0.046958	0.001051
20	average_precision	gender	0.025624	0.001169
21	average_precision	creat_type_cd	0.016801	0.001316
22	average_precision	consume_purchase	0.016742	0.001208
23	average_precision	net_type	0.015281	0.001049
24	average_precision	inter_type_cd	0.013299	0.000862
25	average_precision	up_membership_grade	0.008780	0.000963
26	average_precision	app_first_class	0.008365	0.001548
27	average_precision	membership_life_duration	0.000116	0.000026
28	roc_auc	adv_id	0.118968	0.001974
29	roc_auc	slot_id	0.108140	0.002481
30	roc_auc	adv_prim_id	0.049408	0.001181
31	roc_auc	age	0.048640	0.001417
32	roc_auc	career	0.036239	0.000895
33	roc_auc	device_name	0.036202	0.001040
34	roc_auc	his_app_size	0.032971	0.001164
35	roc_auc	indu_name	0.024529	0.000841
36	roc_auc	residence	0.023072	0.000828
37	roc_auc	list_time	0.022735	0.000685
38	roc_auc	up_life_duration	0.022554	0.000705
39	roc_auc	pt_d	0.022327	0.000636
40	roc_auc	city	0.021381	0.000595
41	roc_auc	communication_onlinerate	0.021212	0.000639
42	roc_auc	device_price	0.019487	0.000602
43	roc_auc	city_rank	0.019000	0.000776
44	roc_auc	app_second_class	0.018540	0.000681
45	roc_auc	communication_avgonline_30d	0.017424	0.000545
46	roc_auc	emui_dev	0.016000	0.000391
47	roc_auc	device_size	0.015233	0.000703
48	roc_auc	gender	0.008234	0.000340
49	roc_auc	net_type	0.006883	0.000230
50	roc_auc	creat_type_cd	0.006762	0.000289
51	roc_auc	app_first_class	0.005967	0.000446
52	roc_auc	consume_purchase	0.005200	0.000362
53	roc_auc	inter_type_cd	0.004506	0.000563
54	roc_auc	up_membership_grade	0.004238	0.000275
55	roc_auc	membership_life_duration	0.000037	0.000011

- We use **average_precision** and **roc_auc** to rank features.
- Features to be dropped
  - 'creat_type_cd',
  - 'up_membership_grade',
  - 'membership_life_duration',
  - 'net_type',
  - 'consume_purchase',
  - 'app_first_class',
  - 'gender',
  - 'inter_type_cd'

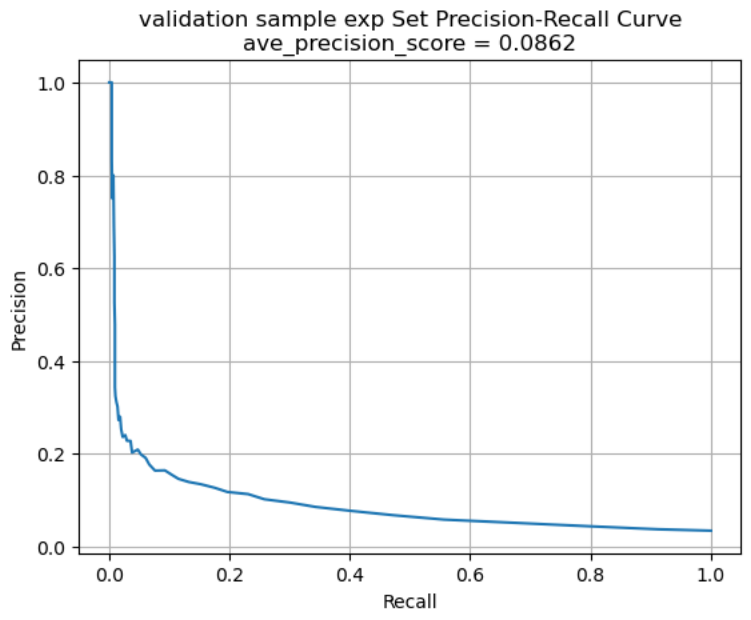
RF classifier – Performance after feature selection (v)

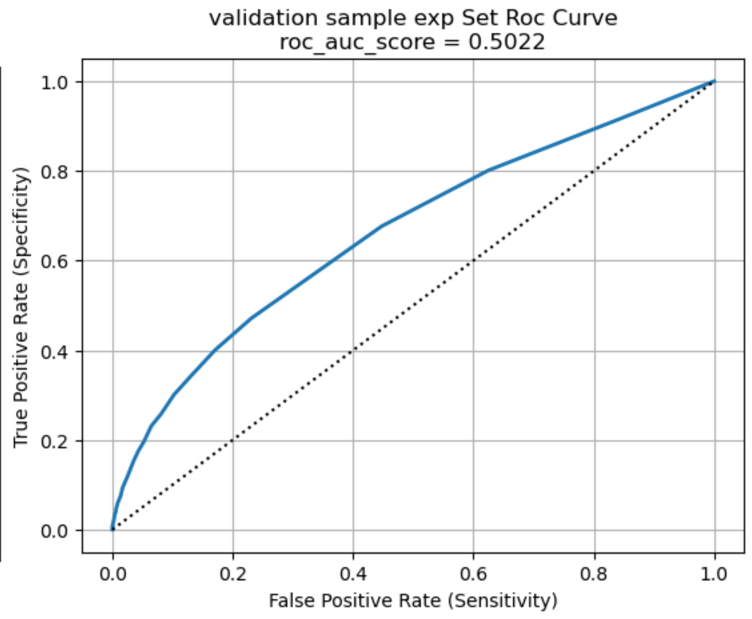


```
Check classification report
{'0': {'precision': 0.9656622911694511, 'recall': 1.0, 'f1-score': 0.9825312267601573, 'support': 32369.0}, '1': {'precis
ion': 1.0, 'recall': 0.004325259515570935, 'f1-score': 0.008613264427217916, 'support': 1156.0}, 'accuracy': 0.9656674123
788218, 'macro avg': {'precision': 0.9828311455847256, 'recall': 0.5021626297577855, 'f1-score': 0.49557224559368757, 'su
pport': 33525.0}, 'weighted avg': {'precision': 0.9668463147759572, 'recall': 0.9656674123788218, 'f1-score': 0.948948850
4900042, 'support': 33525.0}}
```



```
Check confusion matrix
validation sample exp set confusion matrix:
[[32369     0]
 [ 1151     5]]
True Positives =  32369
True Negatives =  5
False Positives(Type I error) =  0
False Negatives(Type II error) =  1151
```



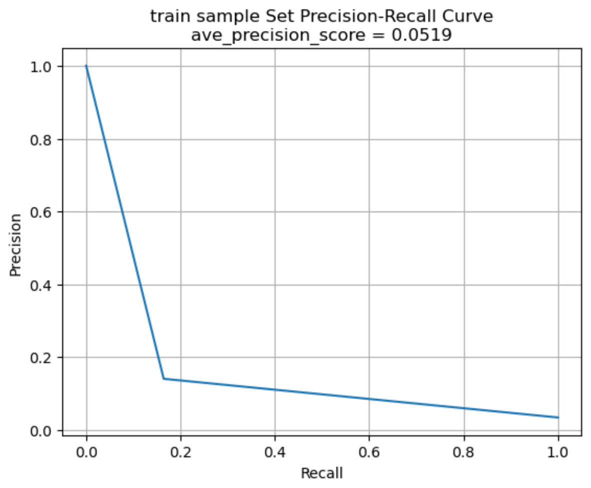


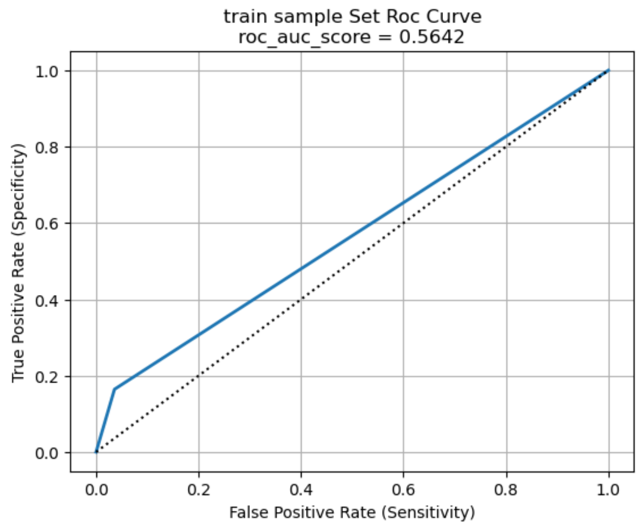
- Because dropping attributes doesn't significantly decrease the performance, we would apply the new drop list in the fine-tuning stage.

Decision Tree Classifier – Performance on train

```
Check classification report
{'0': {'precision': 0.9699701566774435, 'recall': 0.964125699100825, 'f1-score': 0.9670390974880291, 'support': 32363.0},
'1': {'precision': 0.14063656550703182, 'recall': 0.1643598615916955, 'f1-score': 0.1515755883526127, 'support': 1156.0},
'accuracy': 0.9365434529669739, 'macro avg': {'precision': 0.5553033610922377, 'recall': 0.5642427803462603, 'f1-score':
0.5593073429203209, 'support': 33519.0}, 'weighted avg': {'precision': 0.9413681807416162, 'recall': 0.9365434529669739,
'f1-score': 0.9389154715874789, 'support': 33519.0}}
```

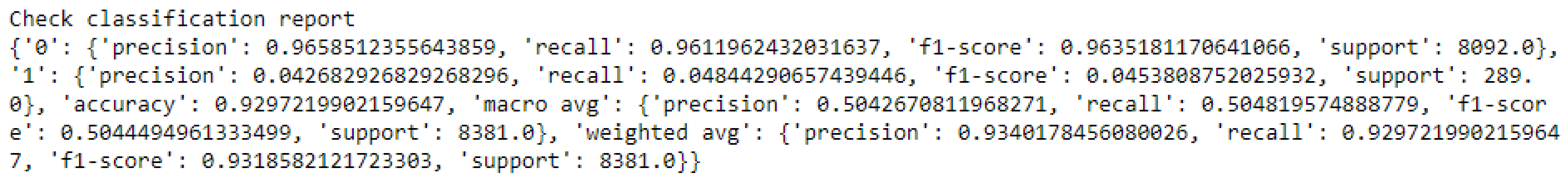
```
Check confusion matrix
train sample set confusion matrix:
[[31202  1161]
 [  966   190]]
True Positives =  31202
True Negatives =  190
False Positives(Type I error) =  1161
False Negatives(Type II error) =  966
```



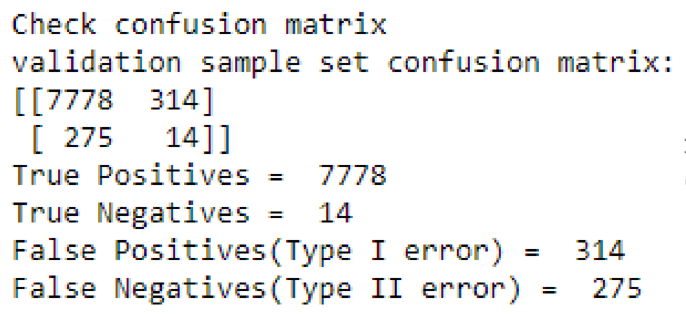


- The low **ave_precision_score** and the number of **false positives** indicate a high rate of Type I error.
- **False negatives** is also high, indicating that the model doesn't perform well in identifying users who actually click on ads.

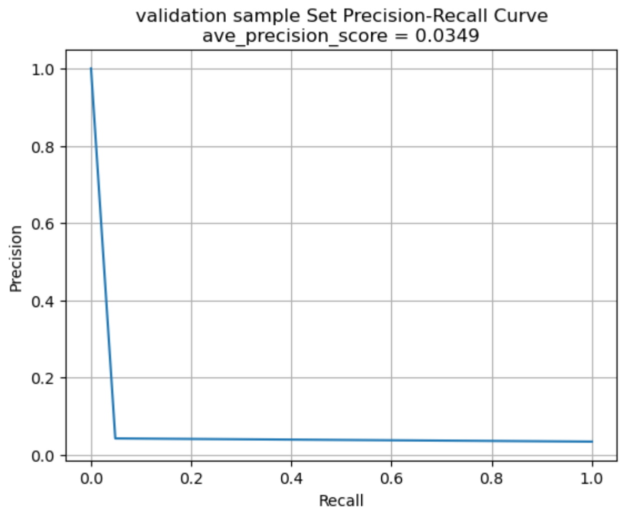
Decision Tree Classifier – Performance on validation

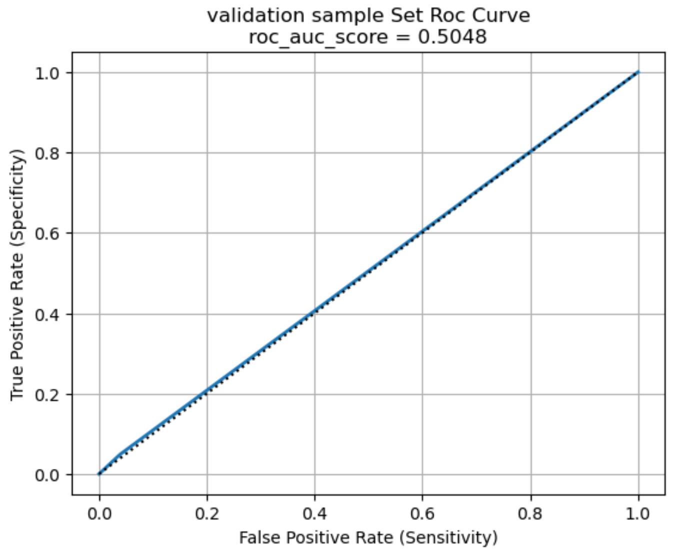


```
Check classification report
{'0': {'precision': 0.9658512355643859, 'recall': 0.9611962432031637, 'f1-score': 0.9635181170641066, 'support': 8092.0},
'1': {'precision': 0.042682926829268296, 'recall': 0.04844290657439446, 'f1-score': 0.0453808752025932, 'support': 289.
0}, 'accuracy': 0.9297219902159647, 'macro avg': {'precision': 0.5042670811968271, 'recall': 0.504819574888779, 'f1-scor
e': 0.5044494961333499, 'support': 8381.0}, 'weighted avg': {'precision': 0.9340178456080026, 'recall': 0.929721990215964
7, 'f1-score': 0.9318582121723303, 'support': 8381.0}}
```



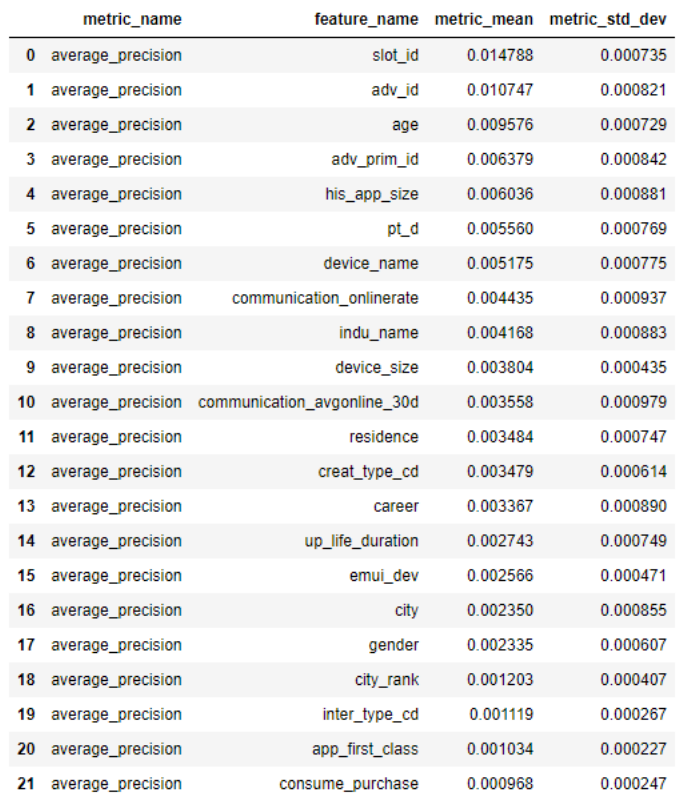
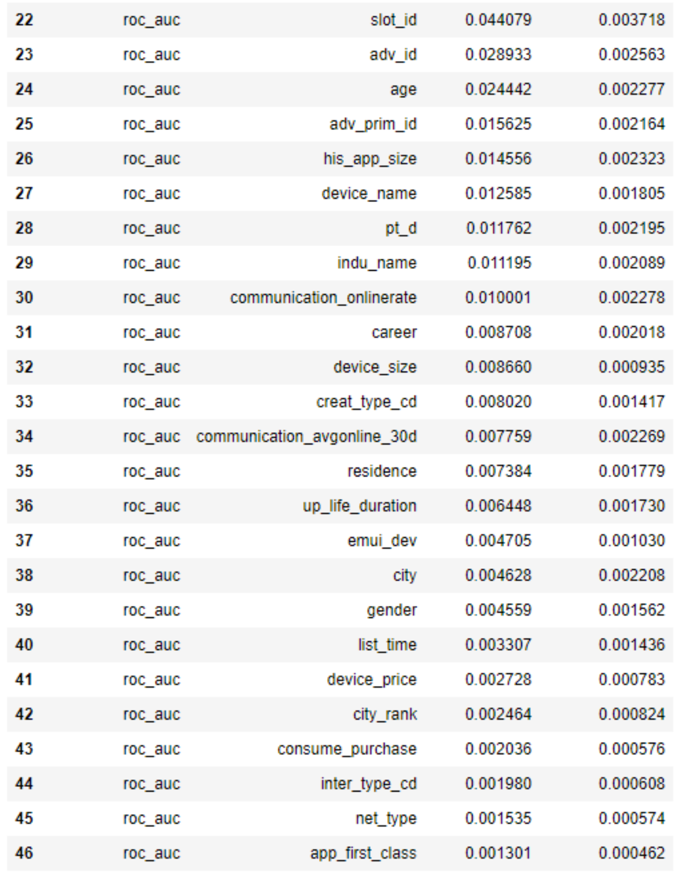
```
Check confusion matrix
validation sample set confusion matrix:
[[7778  314]
 [ 275   14]]
True Positives =  7778
True Negatives =  14
False Positives(Type I error) =  314
False Negatives(Type II error) =  275
```





- Compared to the train set, the **precision score** decreases significantly. Other performance metrics also decrease slightly. The model doesn't generalize to new and unseen data well.

Decision Tree Classifier– Feature selection

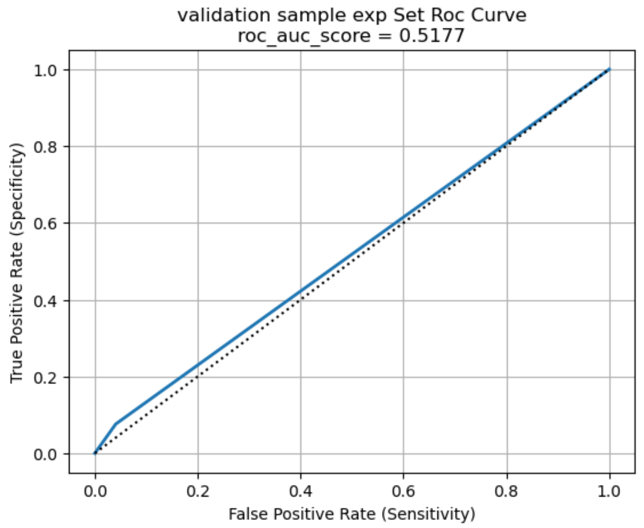
	metric_name	feature_name	metric_mean	metric_std_dev
0	average_precision	slot_id	0.014788	0.000735
1	average_precision	adv_id	0.010747	0.000821
2	average_precision	age	0.009576	0.000729
3	average_precision	adv_prim_id	0.006379	0.000842
4	average_precision	his_app_size	0.006036	0.000881
5	average_precision	pt_d	0.005560	0.000769
6	average_precision	device_name	0.005175	0.000775
7	average_precision	communication_onlinerate	0.004435	0.000937
8	average_precision	indu_name	0.004168	0.000883
9	average_precision	device_size	0.003804	0.000435
10	average_precision	communication_avgonline_30d	0.003558	0.000979
11	average_precision	residence	0.003484	0.000747
12	average_precision	creat_type_cd	0.003479	0.000614
13	average_precision	career	0.003367	0.000890
14	average_precision	up_life_duration	0.002743	0.000749
15	average_precision	emui_dev	0.002566	0.000471
16	average_precision	city	0.002350	0.000855
17	average_precision	gender	0.002335	0.000607
18	average_precision	city_rank	0.001203	0.000407
19	average_precision	inter_type_cd	0.001119	0.000267
20	average_precision	app_first_class	0.001034	0.000227
21	average_precision	consume_purchase	0.000968	0.000247
22	roc_auc	slot_id	0.044079	0.003718
23	roc_auc	adv_id	0.028933	0.002563
24	roc_auc	age	0.024442	0.002277
25	roc_auc	adv_prim_id	0.015625	0.002164
26	roc_auc	his_app_size	0.014556	0.002323
27	roc_auc	device_name	0.012585	0.001805
28	roc_auc	pt_d	0.011762	0.002195
29	roc_auc	indu_name	0.011195	0.002089
30	roc_auc	communication_onlinerate	0.010001	0.002278
31	roc_auc	career	0.008708	0.002018
32	roc_auc	device_size	0.008660	0.000935
33	roc_auc	creat_type_cd	0.008020	0.001417
34	roc_auc	communication_avgonline_30d	0.007759	0.002269
35	roc_auc	residence	0.007384	0.001779
36	roc_auc	up_life_duration	0.006448	0.001730
37	roc_auc	emui_dev	0.004705	0.001030
38	roc_auc	city	0.004628	0.002208
39	roc_auc	gender	0.004559	0.001562
40	roc_auc	list_time	0.003307	0.001436
41	roc_auc	device_price	0.002728	0.000783
42	roc_auc	city_rank	0.002464	0.000824
43	roc_auc	consume_purchase	0.002036	0.000576
44	roc_auc	inter_type_cd	0.001980	0.000608
45	roc_auc	net_type	0.001535	0.000574
46	roc_auc	app_first_class	0.001301	0.000462

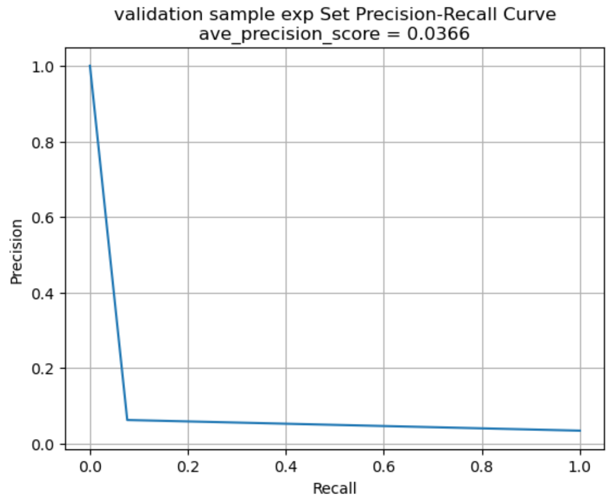
- The use of **average_precision** and **roc_auc** together calculates which feature is important.
- Features to be dropped
  - 'inter_type_cd',
  - 'city_rank',
  - 'app_first_class',
  - 'consume_purchase'

DTree Classifier – After feature selection validation

```
Check classification report
{'0': {'precision': 0.9667496886674969, 'recall': 0.9593425605536332, 'f1-score': 0.9630318819005087, 'support': 8092.0},
'1': {'precision': 0.06267806267806268, 'recall': 0.07612456747404844, 'f1-score': 0.06875, 'support': 289.0}, 'accurac
y': 0.92888676768882, 'macro avg': {'precision': 0.5147138756727798, 'recall': 0.5177335640138409, 'f1-score': 0.51589094
09502543, 'support': 8381.0}, 'weighted avg': {'precision': 0.9355748050126889, 'recall': 0.92888676768882, 'f1-score':
0.9321945756280774, 'support': 8381.0}}
```

```
Check confusion matrix
validation sample exp set confusion matrix:
[[7763  329]
 [ 267   22]]
True Positives =  7763
True Negatives =  22
False Positives(Type I error) =  329
False Negatives(Type II error) =  267
```



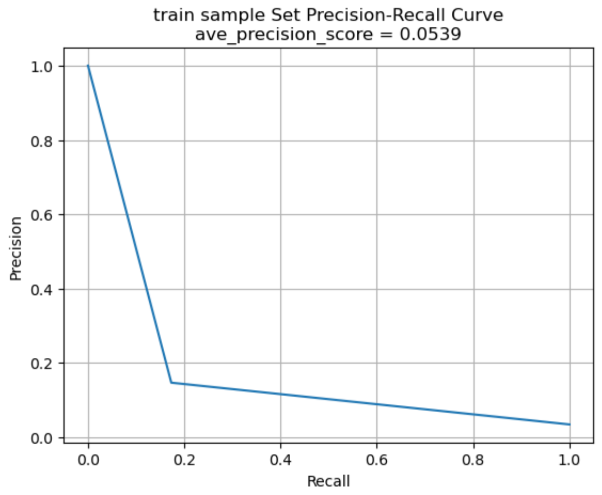


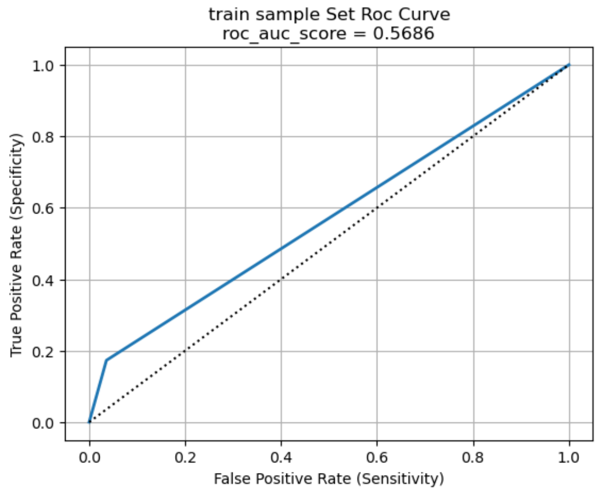
- Because the decrease in **false positives** is subtle and other performance metrics don't change much, we'll consider dropping these features in the fine-tuning stage.

Adaboost Classifier – Performance on train

```
Check classification report
{'0': {'precision': 0.9702708585999937, 'recall': 0.9640947996168464, 'f1-score': 0.9671729696218226, 'support': 32363.
0}, '1': {'precision': 0.14684287812041116, 'recall': 0.17301038062283736, 'f1-score': 0.15885623510722796, 'support': 11
56.0}, 'accuracy': 0.9368119573972971, 'macro avg': {'precision': 0.5585568683602025, 'recall': 0.5685525901198418, 'f1-s
core': 0.5630146023645253, 'support': 33519.0}, 'weighted avg': {'precision': 0.941872554789188, 'recall': 0.936811957397
2971, 'f1-score': 0.9392958209867538, 'support': 33519.0}}
```

```
Check confusion matrix
train sample set confusion matrix:
[[31201  1162]
 [  956   200]]
True Positives =  31201
True Negatives =  200
False Positives(Type I error) =  1162
False Negatives(Type II error) =  956
```



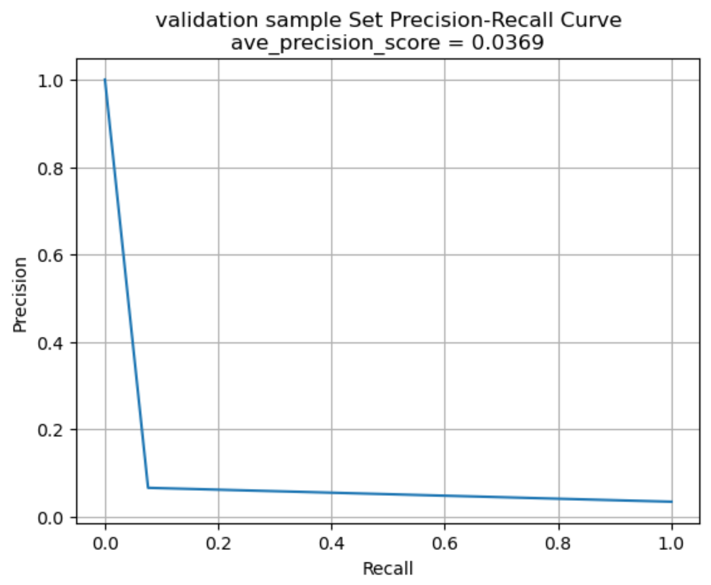


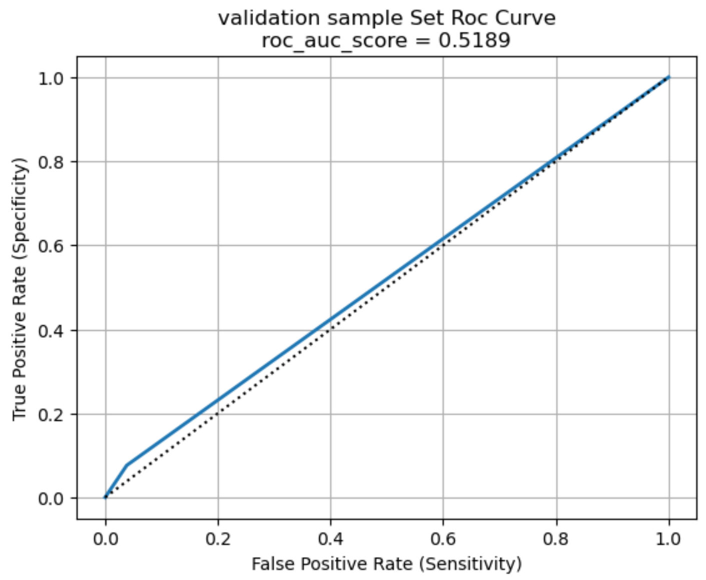
- The default model has many **false positives** and **false negatives**. The model fails to predict the instances well.

Adaboost Classifier – Performance on validation

```
Check classification report
{'0': {'precision': 0.9668281774133433, 'recall': 0.9616905585763718, 'f1-score': 0.9642525246267271, 'support': 8092.0},
'1': {'precision': 0.06626506024096386, 'recall': 0.07612456747404844, 'f1-score': 0.07085346215781, 'support': 289.0},
'accuracy': 0.9311538002624985, 'macro avg': {'precision': 0.5165466188271536, 'recall': 0.5189075630252101, 'f1-score':
0.5175529933922686, 'support': 8381.0}, 'weighted avg': {'precision': 0.9357742768211923, 'recall': 0.9311538002624985,
'f1-score': 0.933445660403661, 'support': 8381.0}}
```

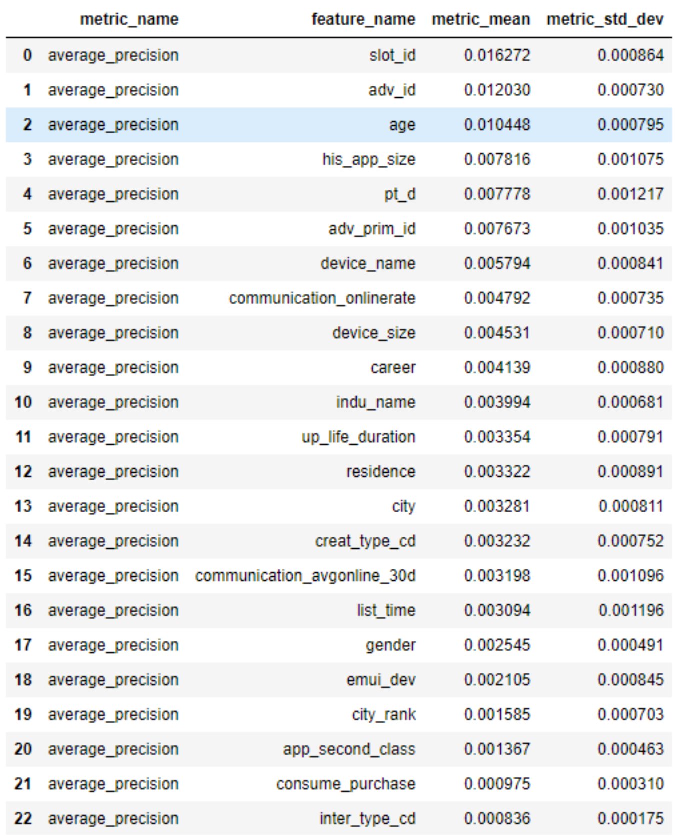
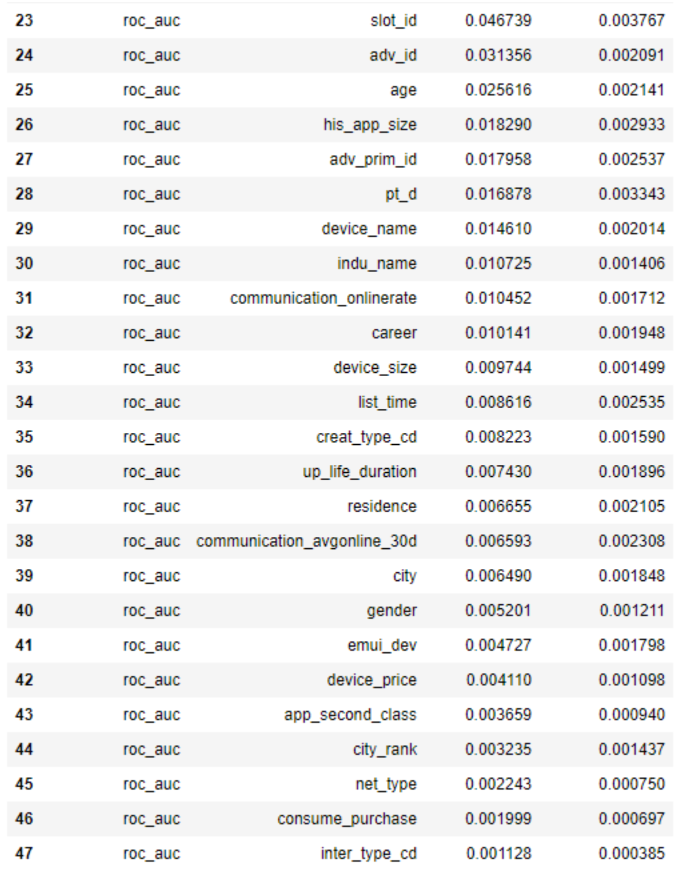
```
Check confusion matrix
validation sample set confusion matrix:
[[7782  310]
 [ 267   22]]
True Positives =  7782
True Negatives =  22
False Positives(Type I error) =  310
False Negatives(Type II error) =  267
```





- Compared to the performance on the train set, the performance on the validation set is worse.

Adaboost Classifier– Feature selection

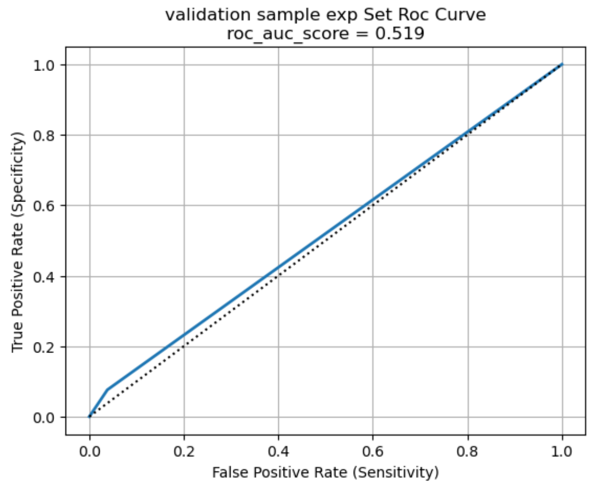
	metric_name	feature_name	metric_mean	metric_std_dev
0	average_precision	slot_id	0.016272	0.000864
1	average_precision	adv_id	0.012030	0.000730
2	average_precision	age	0.010448	0.000795
3	average_precision	his_app_size	0.007816	0.001075
4	average_precision	pt_d	0.007778	0.001217
5	average_precision	adv_prim_id	0.007673	0.001035
6	average_precision	device_name	0.005794	0.000841
7	average_precision	communication_onlinerate	0.004792	0.000735
8	average_precision	device_size	0.004531	0.000710
9	average_precision	career	0.004139	0.000880
10	average_precision	indu_name	0.003994	0.000681
11	average_precision	up_life_duration	0.003354	0.000791
12	average_precision	residence	0.003322	0.000891
13	average_precision	city	0.003281	0.000811
14	average_precision	creat_type_cd	0.003232	0.000752
15	average_precision	communication_avgonline_30d	0.003198	0.001096
16	average_precision	list_time	0.003094	0.001196
17	average_precision	gender	0.002545	0.000491
18	average_precision	emui_dev	0.002105	0.000845
19	average_precision	city_rank	0.001585	0.000703
20	average_precision	app_second_class	0.001367	0.000463
21	average_precision	consume_purchase	0.000975	0.000310
22	average_precision	inter_type_cd	0.000836	0.000175
23	roc_auc	slot_id	0.046739	0.003767
24	roc_auc	adv_id	0.031356	0.002091
25	roc_auc	age	0.025616	0.002141
26	roc_auc	his_app_size	0.018290	0.002933
27	roc_auc	adv_prim_id	0.017958	0.002537
28	roc_auc	pt_d	0.016878	0.003343
29	roc_auc	device_name	0.014610	0.002014
30	roc_auc	indu_name	0.010725	0.001406
31	roc_auc	communication_onlinerate	0.010452	0.001712
32	roc_auc	career	0.010141	0.001948
33	roc_auc	device_size	0.009744	0.001499
34	roc_auc	list_time	0.008616	0.002535
35	roc_auc	creat_type_cd	0.008223	0.001590
36	roc_auc	up_life_duration	0.007430	0.001896
37	roc_auc	residence	0.006655	0.002105
38	roc_auc	communication_avgonline_30d	0.006593	0.002308
39	roc_auc	city	0.006490	0.001848
40	roc_auc	gender	0.005201	0.001211
41	roc_auc	emui_dev	0.004727	0.001798
42	roc_auc	device_price	0.004110	0.001098
43	roc_auc	app_second_class	0.003659	0.000940
44	roc_auc	city_rank	0.003235	0.001437
45	roc_auc	net_type	0.002243	0.000750
46	roc_auc	consume_purchase	0.001999	0.000697
47	roc_auc	inter_type_cd	0.001128	0.000385

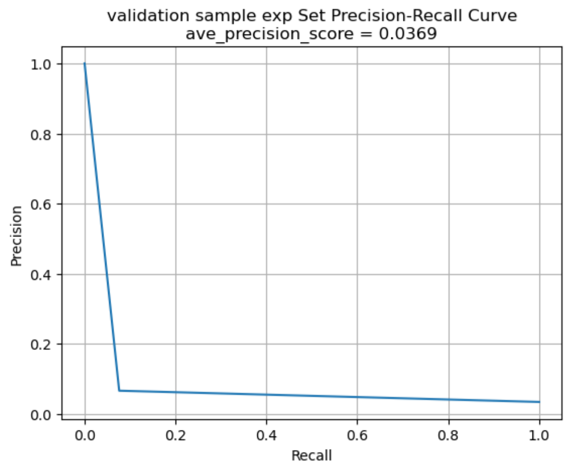
- Features to be dropped
  - 'city_rank',
  - 'consume_purchase',
  - 'emui_dev',
  - 'inter_type_cd',
  - 'app_second_class',
  - 'creat_type_cd',

Adaboost Classifier – After feature selection validation

```
Check classification report
{'0': {'precision': 0.966832298136646, 'recall': 0.9618141374196737, 'f1-score': 0.9643166893817371, 'support': 8092.0},
'1': {'precision': 0.06646525679758308, 'recall': 0.07612456747404844, 'f1-score': 0.07096774193548387, 'support': 289.0},
'accuracy': 0.9312731177663763, 'macro avg': {'precision': 0.5166487774671145, 'recall': 0.518969352446861, 'f1-score': 0.51
76422156586105, 'support': 8381.0}, 'weighted avg': {'precision': 0.9357851587801266, 'recall': 0.9312731177663763, 'f1-scor
e': 0.9335115532629008, 'support': 8381.0}}
```

```
Check confusion matrix
validation sample exp set confusion matrix:
[[7783  309]
 [ 267   22]]
True Positives =  7783
True Negatives =  22
False Positives(Type I error) =  309
False Negatives(Type II error) =  267
```



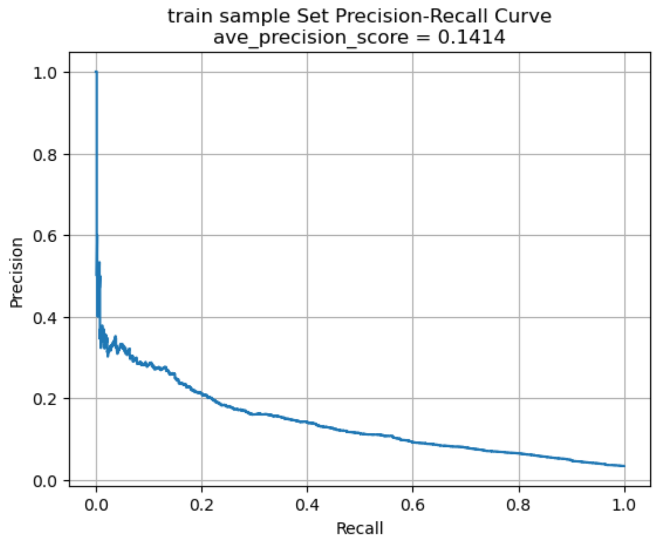


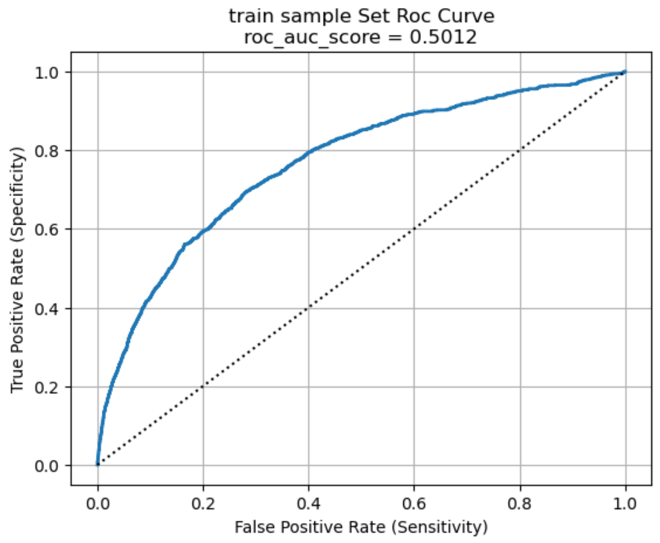
- Because the decrease in **false positives** is subtle and other performance metrics don’t change much, we’ll consider dropping these features in the fine-tuning stage.

Gradient Boosting Classifier – Performance on train

```
Check classification report
{'0': {'precision': 0.9655944139412748, 'recall': 0.9998764020640856, 'f1-score': 0.9824364326375712, 'support': 32363.
0}, '1': {'precision': 0.42857142857142855, 'recall': 0.0025951557093425604, 'f1-score': 0.005159071367153912, 'support':
1156.0}, 'accuracy': 0.9654822637906859, 'macro avg': {'precision': 0.6970829212563516, 'recall': 0.5012357788867141, 'f1
-score': 0.49379775200236253, 'support': 33519.0}, 'weighted avg': {'precision': 0.9470736176440242, 'recall': 0.96548226
37906859, 'f1-score': 0.948732186400255, 'support': 33519.0}}
```

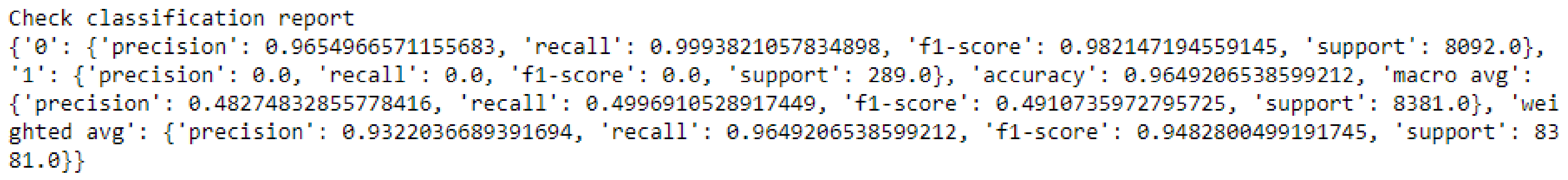
```
Check confusion matrix
train sample set confusion matrix:
[[32359     4]
 [ 1153     3]]
True Positives =  32359
True Negatives =  3
False Positives(Type I error) =  4
False Negatives(Type II error) =  1153
```



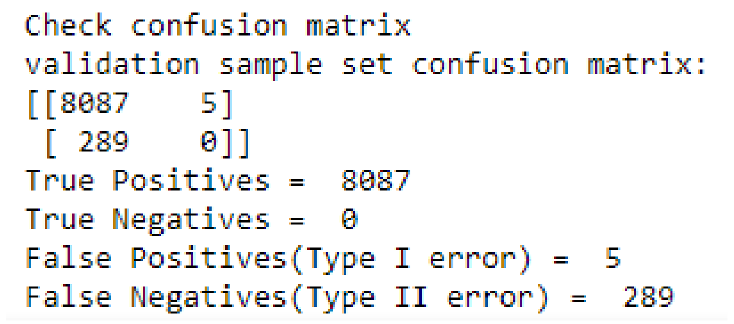


- This model has few **false positive** although the number of **false negatives** is large.

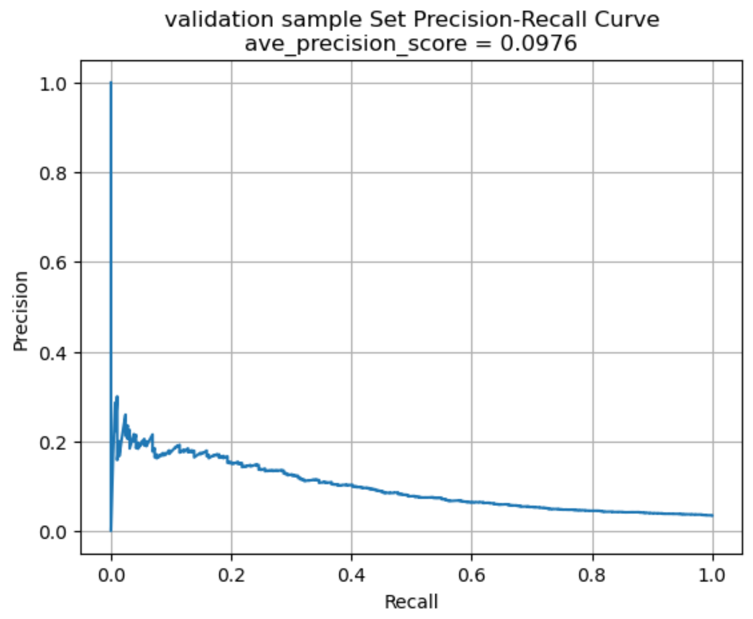
GBoosting Classifier – Performance on validation

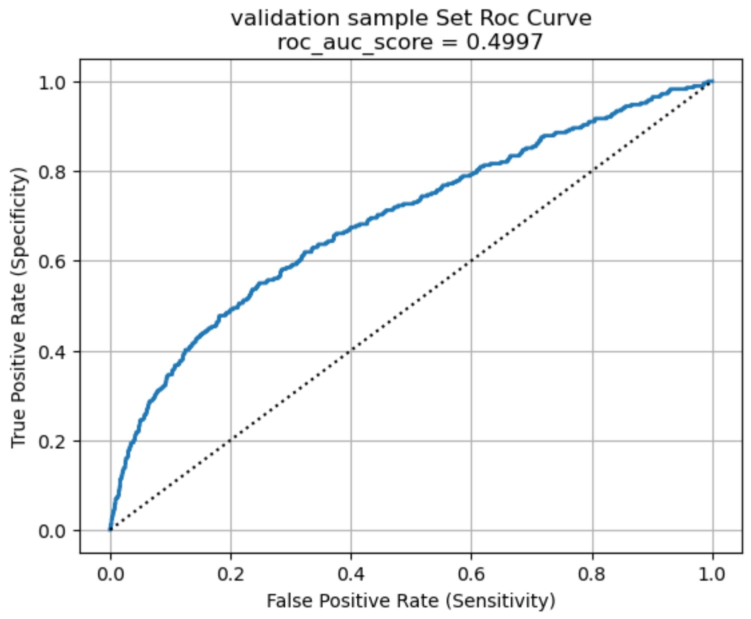


```
Check classification report
{'0': {'precision': 0.9654966571155683, 'recall': 0.9993821057834898, 'f1-score': 0.982147194559145, 'support': 8092.0},
'1': {'precision': 0.0, 'recall': 0.0, 'f1-score': 0.0, 'support': 289.0}, 'accuracy': 0.9649206538599212, 'macro avg':
{'precision': 0.48274832855778416, 'recall': 0.4996910528917449, 'f1-score': 0.4910735972795725, 'support': 8381.0}, 'wei
ghted avg': {'precision': 0.9322036689391694, 'recall': 0.9649206538599212, 'f1-score': 0.9482800499191745, 'support': 83
81.0}}
```



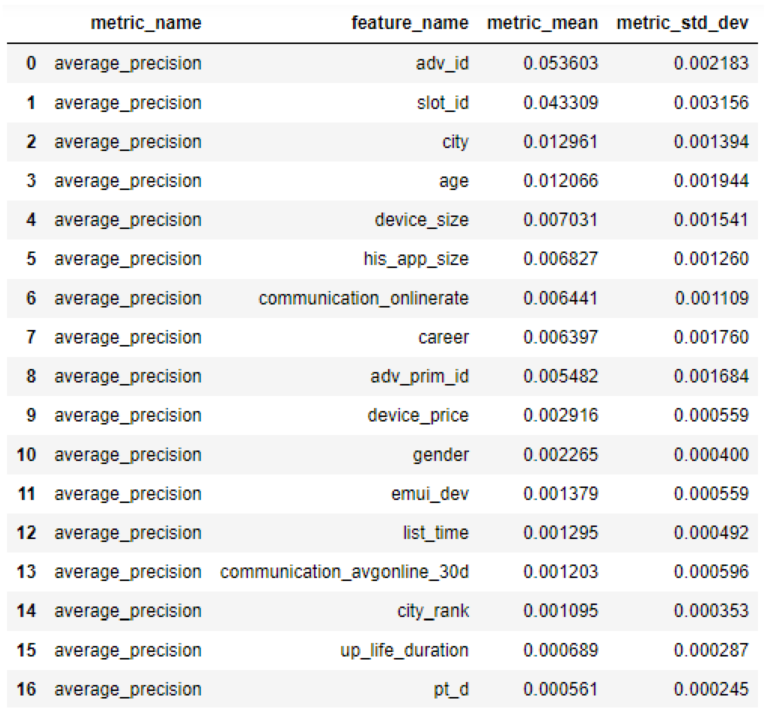
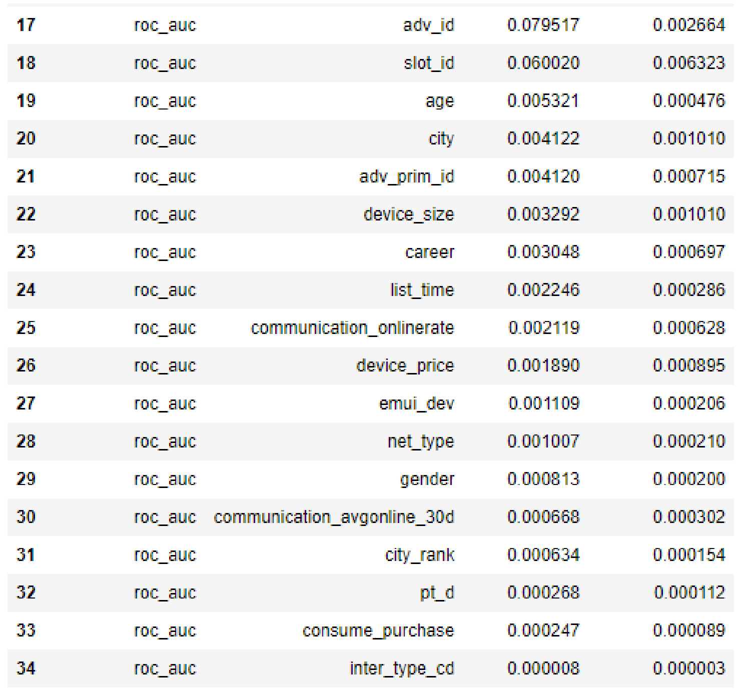
```
Check confusion matrix
validation sample set confusion matrix:
[[8087    5]
 [ 289    0]]
True Positives =  8087
True Negatives =  0
False Positives(Type I error) =  5
False Negatives(Type II error) =  289
```





- Compared to the train set, the validation set has much lower **precision score** and **ave_precision_score**. Notably, **precision score** for class 1 is 0.

GBoosting Classifier – Feature selection

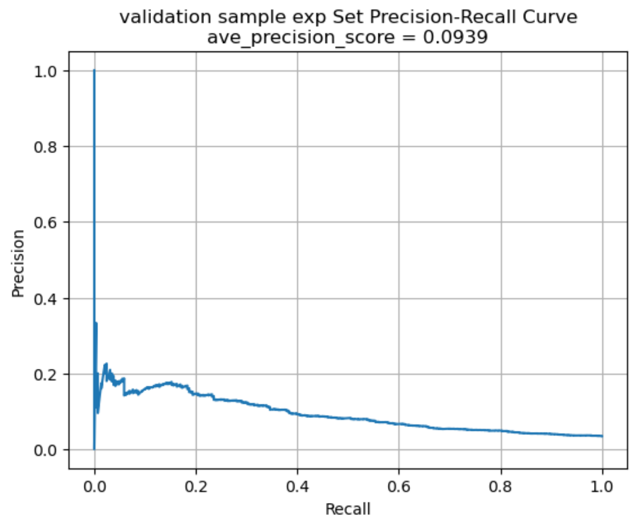
	metric_name	feature_name	metric_mean	metric_std_dev
0	average_precision	adv_id	0.053603	0.002183
1	average_precision	slot_id	0.043309	0.003156
2	average_precision	city	0.012961	0.001394
3	average_precision	age	0.012066	0.001944
4	average_precision	device_size	0.007031	0.001541
5	average_precision	his_app_size	0.006827	0.001260
6	average_precision	communication_onlinerate	0.006441	0.001109
7	average_precision	career	0.006397	0.001760
8	average_precision	adv_prim_id	0.005482	0.001684
9	average_precision	device_price	0.002916	0.000559
10	average_precision	gender	0.002265	0.000400
11	average_precision	emui_dev	0.001379	0.000559
12	average_precision	list_time	0.001295	0.000492
13	average_precision	communication_avgonline_30d	0.001203	0.000596
14	average_precision	city_rank	0.001095	0.000353
15	average_precision	up_life_duration	0.000689	0.000287
16	average_precision	pt_d	0.000561	0.000245
17	roc_auc	adv_id	0.079517	0.002664
18	roc_auc	slot_id	0.060020	0.006323
19	roc_auc	age	0.005321	0.000476
20	roc_auc	city	0.004122	0.001010
21	roc_auc	adv_prim_id	0.004120	0.000715
22	roc_auc	device_size	0.003292	0.001010
23	roc_auc	career	0.003048	0.000697
24	roc_auc	list_time	0.002246	0.000286
25	roc_auc	communication_onlinerate	0.002119	0.000628
26	roc_auc	device_price	0.001890	0.000895
27	roc_auc	emui_dev	0.001109	0.000206
28	roc_auc	net_type	0.001007	0.000210
29	roc_auc	gender	0.000813	0.000200
30	roc_auc	communication_avgonline_30d	0.000668	0.000302
31	roc_auc	city_rank	0.000634	0.000154
32	roc_auc	pt_d	0.000268	0.000112
33	roc_auc	consume_purchase	0.000247	0.000089
34	roc_auc	inter_type_cd	0.000008	0.000003

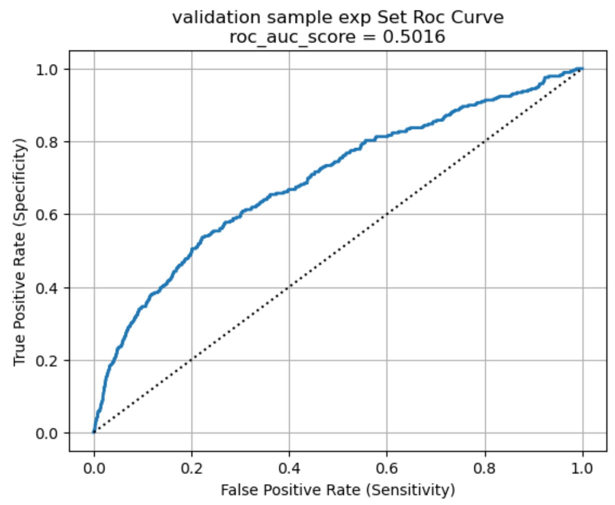
- Features to be dropped
  - 'device_size',
  - 'pt_d',
  - 'career',
  - 'emui_dev',
  - 'communication_avgonline_30d',
  - 'adv_prim_id',
  - 'gender',
  - 'list_time',
  - 'city_rank',
  - 'communication_onlinerate',
  - 'device_price'

GBoosting Classifier – After feature selection validation

```
Check classification report
{'0': {'precision': 0.9656242539985677, 'recall': 0.9997528423133959, 'f1-score': 0.9823922282938676, 'support': 8092.0},
'1': {'precision': 0.3333333333333333, 'recall': 0.0034602076124567475, 'f1-score': 0.006849315068493151, 'support': 289.0},
'accuracy': 0.9653979238754326, 'macro avg': {'precision': 0.6494787936659505, 'recall': 0.5016065249629263, 'f1-score': 0.4
946207716811804, 'support': 8381.0}, 'weighted avg': {'precision': 0.9438211188032147, 'recall': 0.9653979238754326, 'f1-sco
re': 0.9487528174929927, 'support': 8381.0}}
```

```
Check confusion matrix
validation sample exp set confusion matrix:
[[8090    2]
 [ 288    1]]
True Positives =  8090
True Negatives =  1
False Positives(Type I error) =  2
False Negatives(Type II error) =  288
```





- The model performances before and after dropping some common least important features in permutation importance test are similar. We'll drop these features in the fine-tuning stage.

Classifiers – Comparison

SGD Classifier

	stage	accuracy	precision	recall	cv_mean_accuracy	cv_mean_precision	cv_mean_recall	cv_mean_f1	roc_auc_score
0	train sample	0.6873	0.058004	0.529412	0.5468	0.0457	0.6064	0.0849	0.6112
0	validation sample	0.6852	0.050172	0.453287	0.5531	0.0444	0.5746	0.0823	0.5734
0	train sample exp	0.5561	0.053088	0.705017	0.5464	0.0474	0.6229	0.0880	0.6279
0	validation sample exp	0.5579	0.048626	0.636678	0.5890	0.0431	0.5160	0.0792	0.5959

Decision Tree Classifier

	stage	accuracy	precision	recall	cv_mean_accuracy	cv_mean_precision	cv_mean_recall	cv_mean_f1	roc_auc_score
0	train sample	0.9365	0.140637	0.164360	0.9265	0.0547	0.0701	0.0614	0.5642
0	validation sample	0.9297	0.042683	0.048443	0.9264	0.0723	0.0898	0.0795	0.5048
0	train sample exp	0.9350	0.150273	0.190311	0.9246	0.0599	0.0796	0.0680	0.5759
0	validation sample exp	0.9289	0.062678	0.076125	0.9281	0.0638	0.0762	0.0691	0.5177

Random Forest Classifier

	stage	accuracy	precision	recall	cv_mean_accuracy	cv_mean_precision	cv_mean_recall	cv_mean_f1	roc_auc_score
0	train sample	0.9656	0.884615	0.004975	0.9655	0.3467	0.0013	0.0026	0.5025
0	validation sample	0.9656	0.833333	0.004325	0.9655	0.1333	0.0017	0.0034	0.5021
0	train sample exp	0.9657	0.958333	0.004975	0.9655	0.4333	0.0011	0.0022	0.5025
0	validation sample exp	0.9657	1.000000	0.004325	0.9655	0.0000	0.0000	0.0000	0.5022

Gradient Boosting Classifier

	stage	accuracy	precision	recall	cv_mean_accuracy	cv_mean_precision	cv_mean_recall	cv_mean_f1	roc_auc_score
0	train sample	0.9655	0.428571	0.002595	0.9654	0.00	0.0000	0.0000	0.5012
0	validation sample	0.9649	0.000000	0.000000	0.9650	0.10	0.0034	0.0067	0.4997
0	train sample exp	0.9655	0.600000	0.002595	0.9652	0.05	0.0009	0.0017	0.5013
0	validation sample exp	0.9654	0.333333	0.003460	0.9644	0.25	0.0070	0.0133	0.5016

Adaboost Classifier

	stage	accuracy	precision	recall	cv_mean_accuracy	cv_mean_precision	cv_mean_recall	cv_mean_f1	roc_auc_score
0	train sample	0.9368	0.146843	0.173010	0.9265	0.0552	0.0701	0.0617	0.5686
0	validation sample	0.9312	0.066265	0.076125	0.9260	0.0674	0.0831	0.0738	0.5189
0	train sample exp	0.9370	0.162429	0.198962	0.9241	0.0577	0.0796	0.0666	0.5812
0	validation sample exp	0.9313	0.066465	0.076125	0.9267	0.0560	0.0691	0.0612	0.5190

- This is a summary table for all models. All models need to be improved based on the performance metrics.