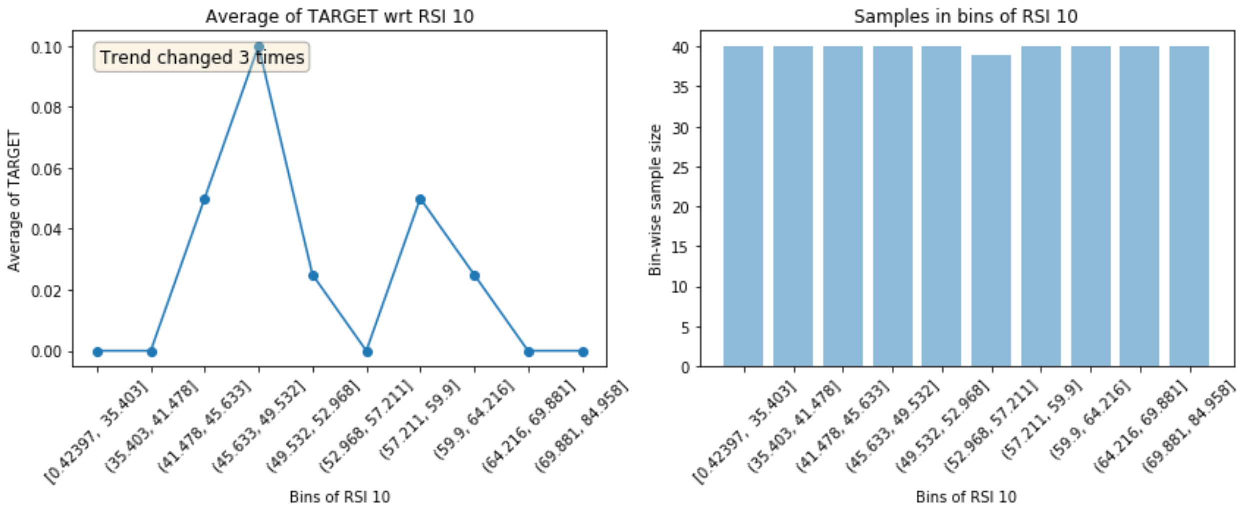
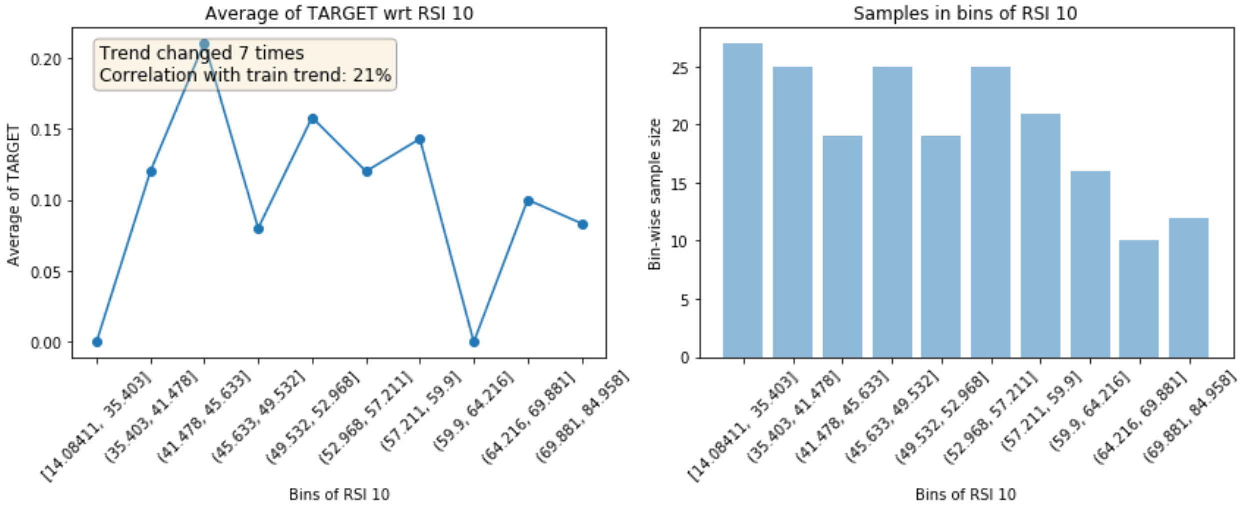
```
stock_symbols = ["SPY"]
read_csv = False
export_csv = True
csv_name = "WITHIN_STOCK_SPY.xlsx"
threshold = .85
days = 1460
percent = 5
days_within = 10
within = True

model, X_train, y_train, X_test, y_test, X_valid, y_valid, select_X_train, se
lect_X_test, select_X_valid, stock_data, feature_bool = model_creation3(stock
_symbols, read_csv, export_csv, csv_name, threshold, days, percent, days_with
in, within)

STEP ONE DONE
BASELINE TRIAL
Mean Accuracy: 90.45226130653276%
Mean Precision: 53.333333333333314%
Mean Recall: 40.0%
Mean AUC: 68.04469273743027%
BASELINE TRIAL w/ THRESHOLDING
Mean Accuracy: 90.45226130653276%
Mean Precision: 54.545454545454625%
Mean Recall: 30.0%
Mean AUC: 63.603351955307296%
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, eval_metric='auc',
              gamma=0, gpu_id=-1, importance_type='gain',
              interaction_constraints='', learning_rate=0.300000012,
              max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
              monotone_constraints='()', n_estimators=100, n_jobs=4,
              num_parallel_tree=1, objective='binary:logistic', random_state=
0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', use_label_encoder=False,
              validate_parameters=1, verbosity=None)
STEP TWO DONE
                                          Plots for RSI 10
                                          Train data plots
```

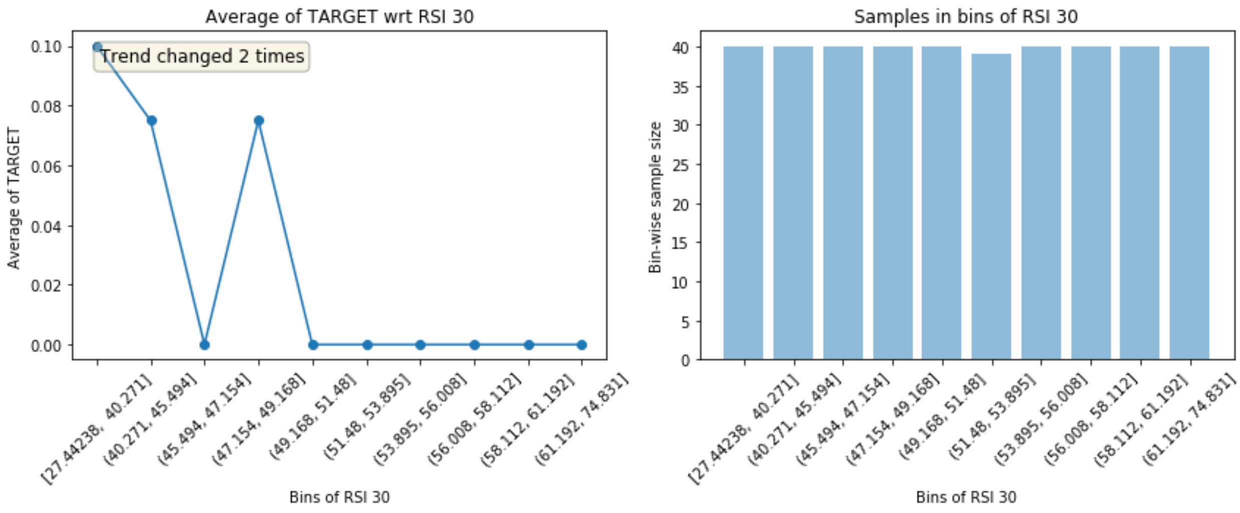


Test data plots

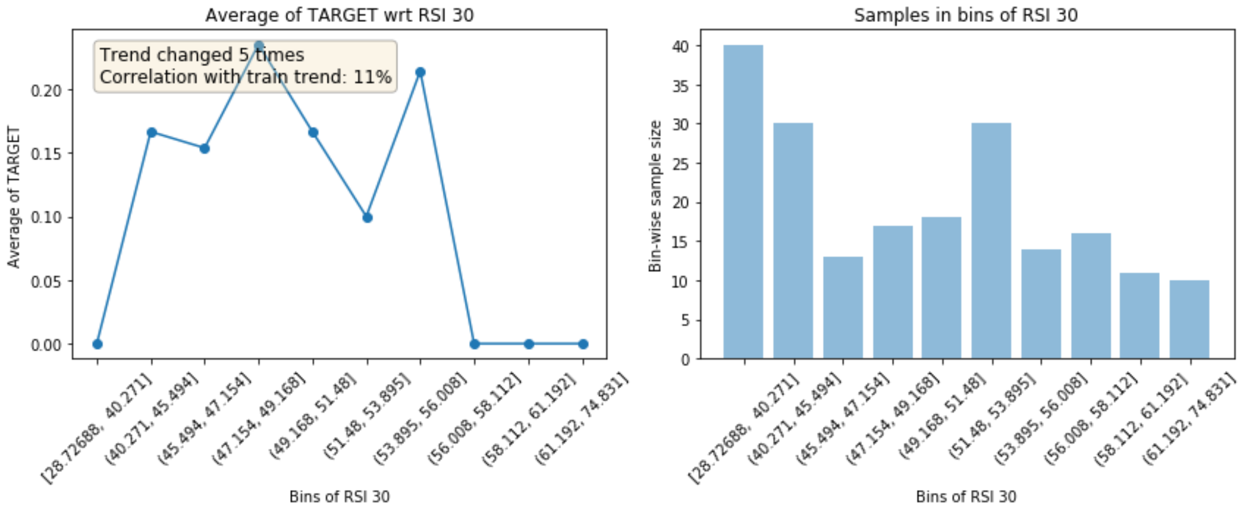


Plots for RSI 30

Train data plots

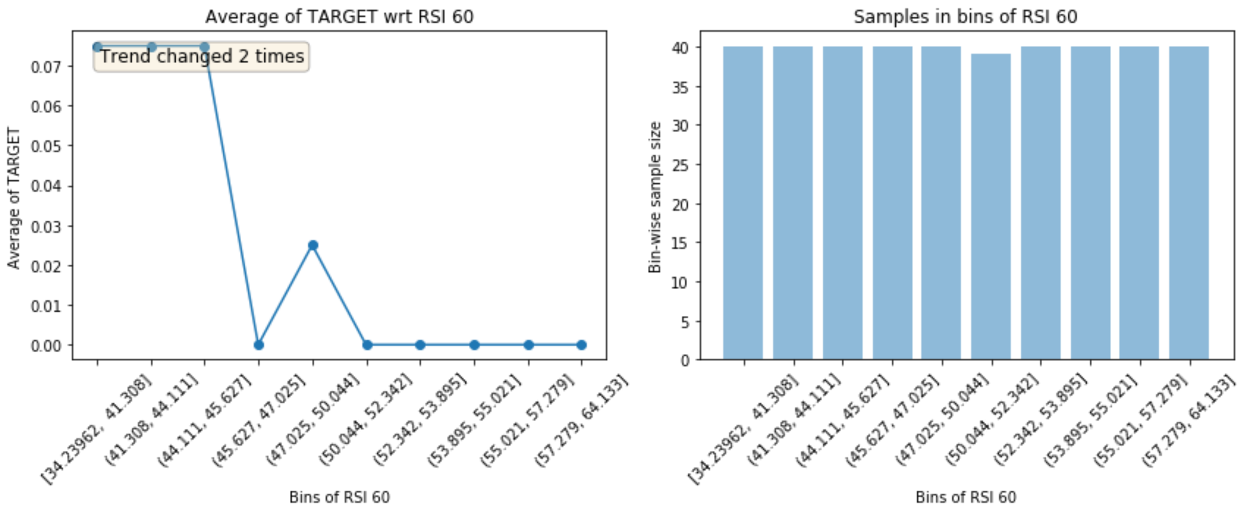


Test data plots

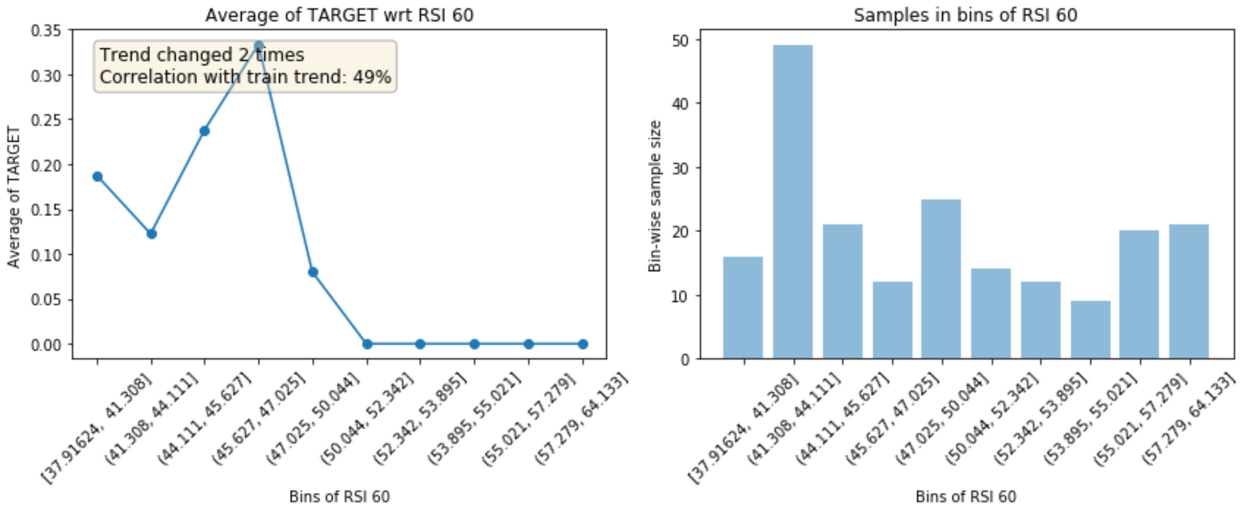


```
------------------------------------------------------------------------------
---------------------------------
```

Plots for RSI 60
Train data plots

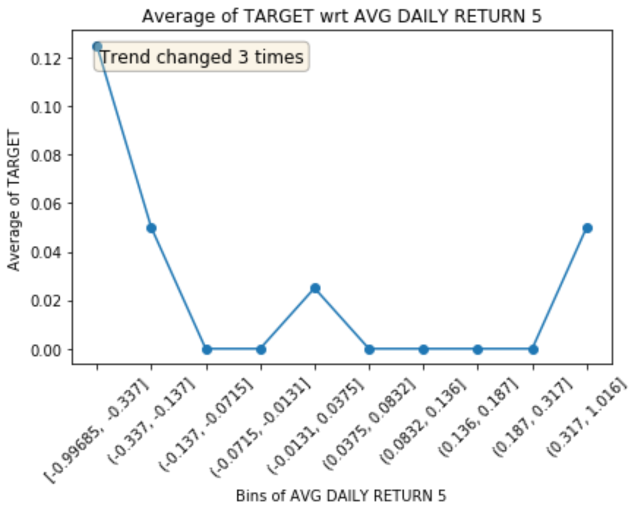


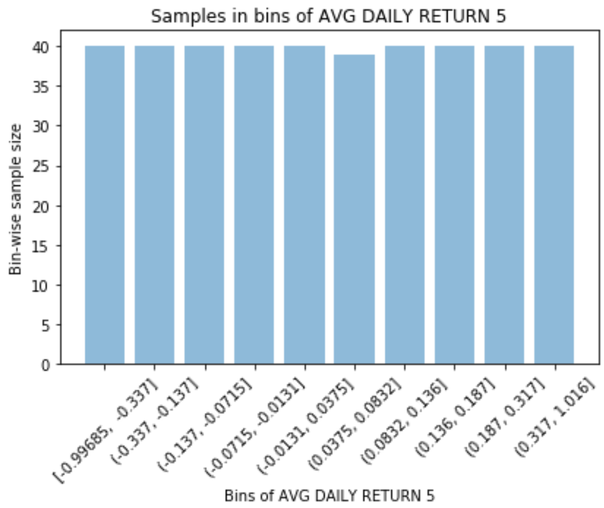
Test data plots



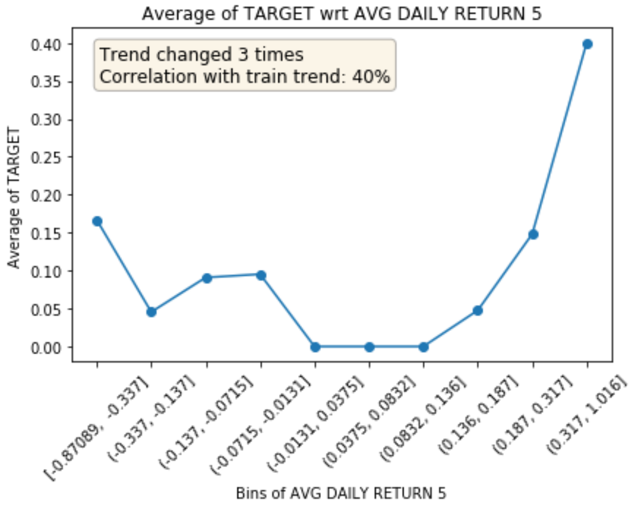
Plots for AVG DAILY RETURN 5

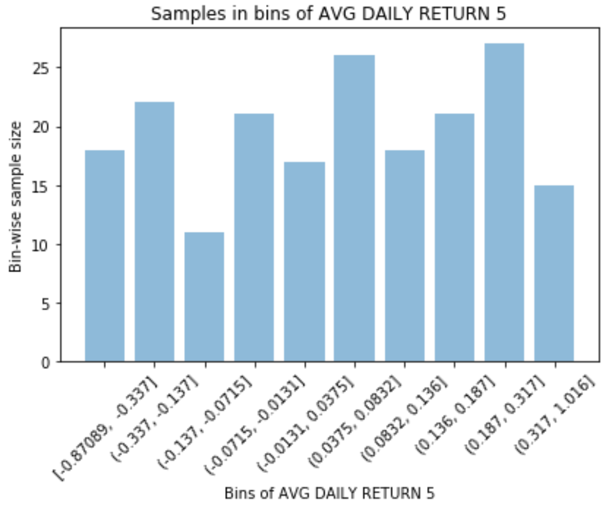
Train data plots


Average of TARGET wrt AVG DAILY RETURN 5

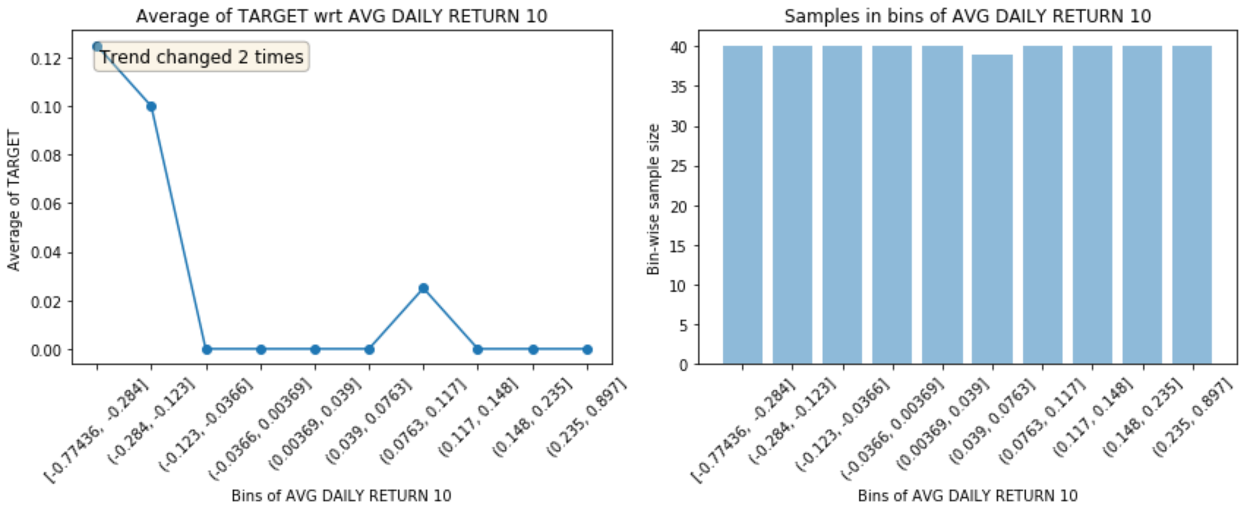

Samples in bins of AVG DAILY RETURN 5

```
Test data plots
```

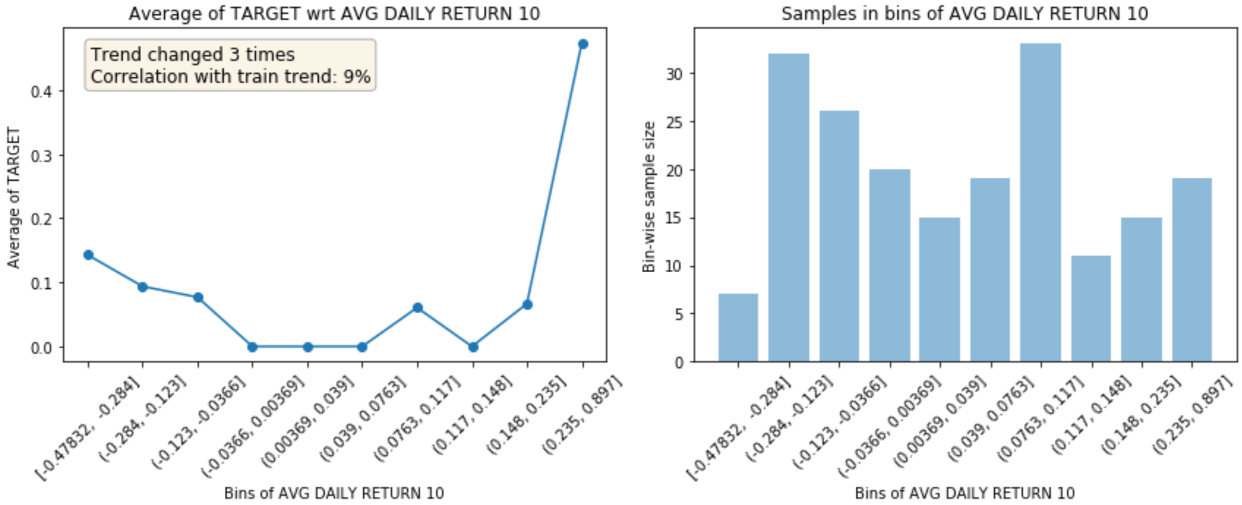

Average of TARGET wrt AVG DAILY RETURN 5


Samples in bins of AVG DAILY RETURN 5

```
------------------------------------------------------------------------------
---------------------------------
                                                                                  
              Plots for AVG DAILY RETURN 10
                   Train data plots
```



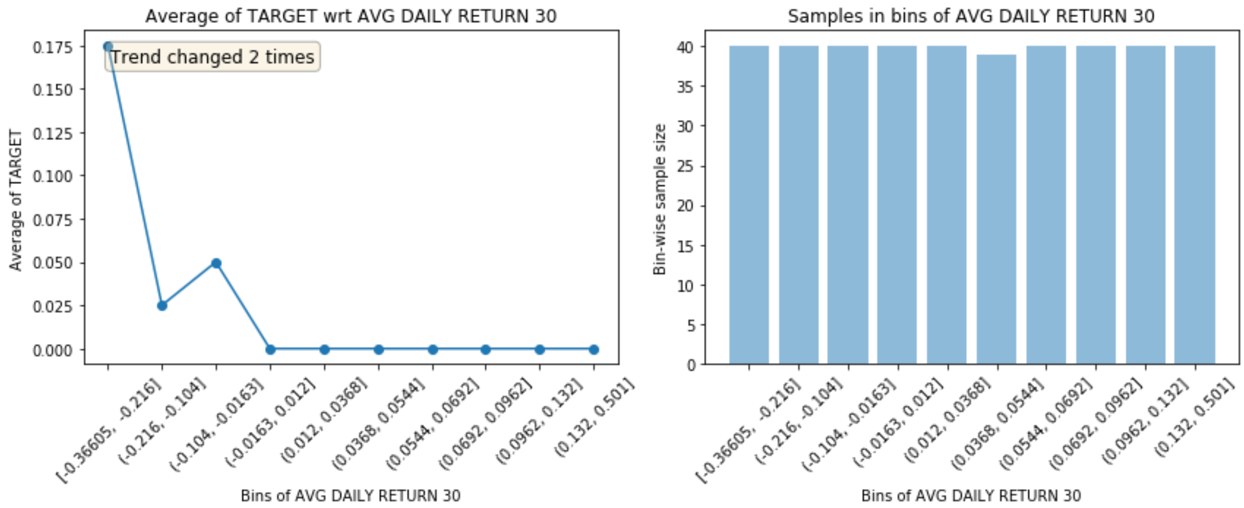
Test data plots



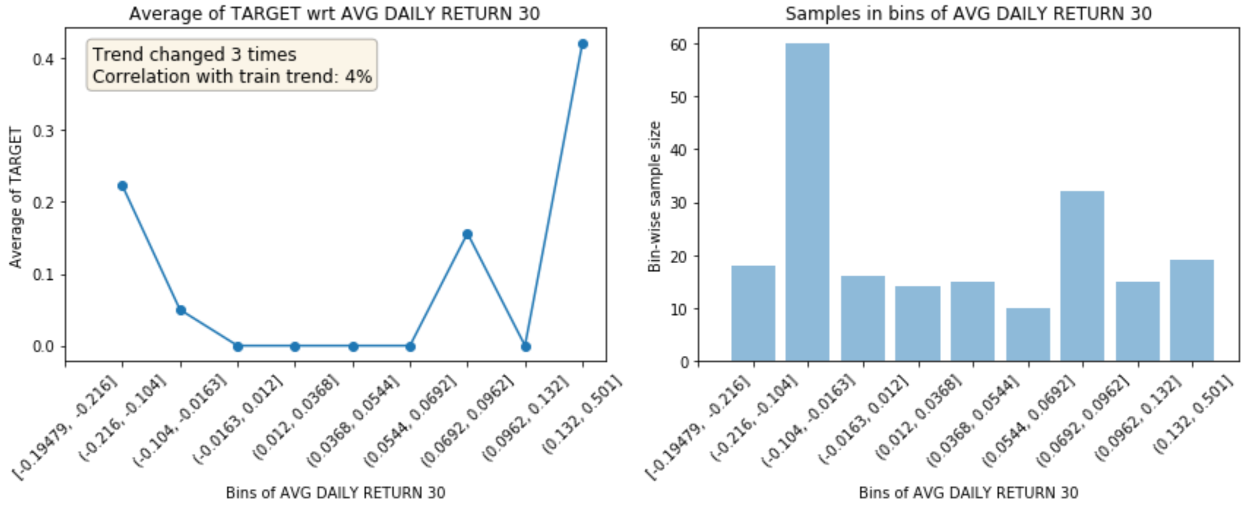
```
------------------------------------------------------------------------------
---------------------------------
```

Plots for AVG DAILY RETURN 30

Train data plots



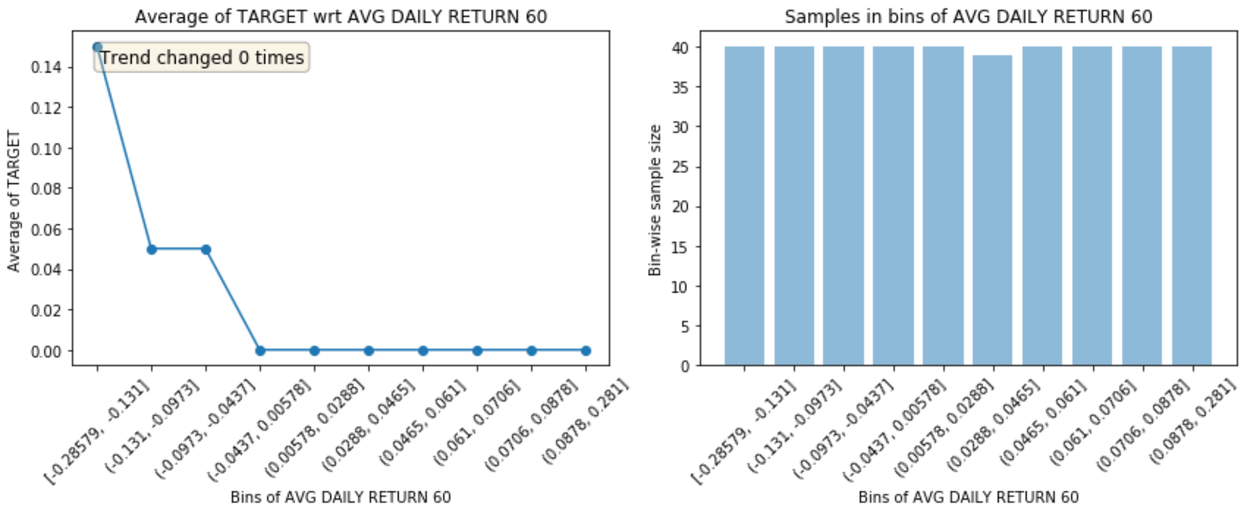
Test data plots



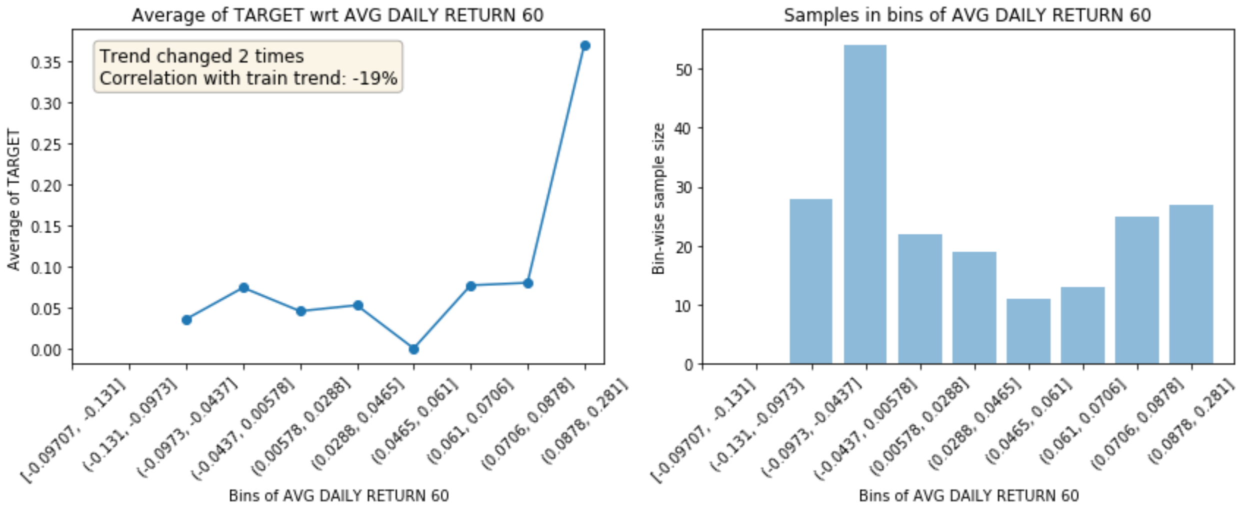
```
------------------------------------------------------------------------------
---------------------------------
```

Plots for AVG DAILY RETURN 60

Train data plots

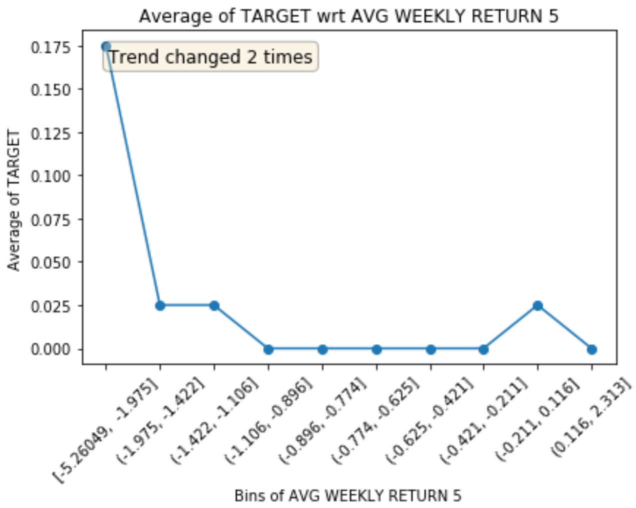


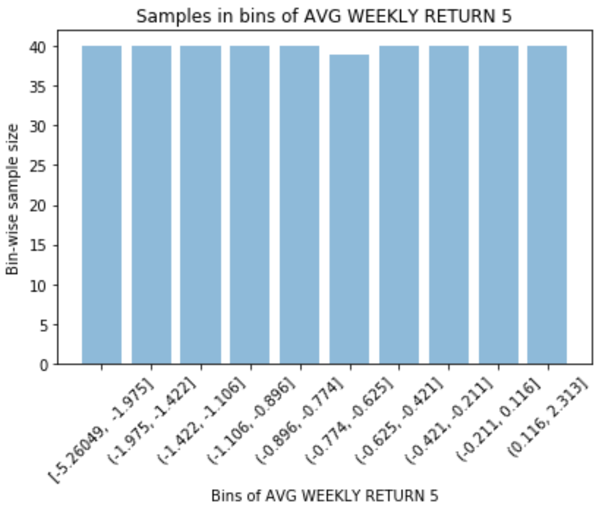
Test data plots



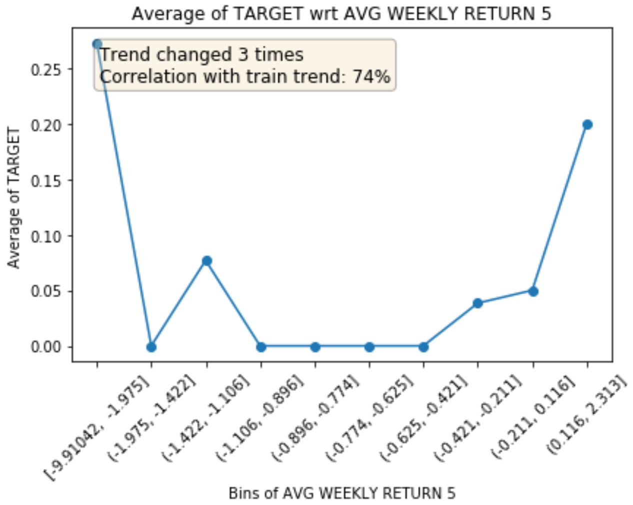
Plots for AVG WEEKLY RETURN 5

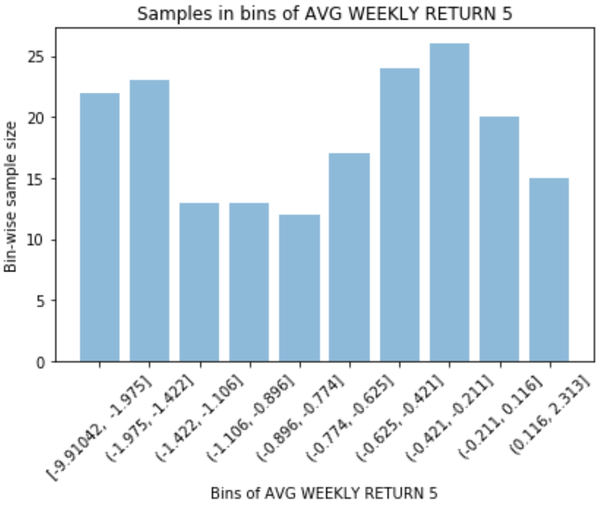
Train data plots





Test data plots

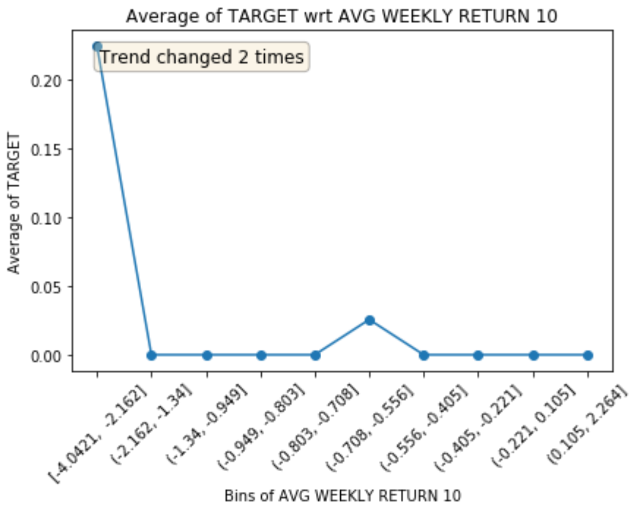


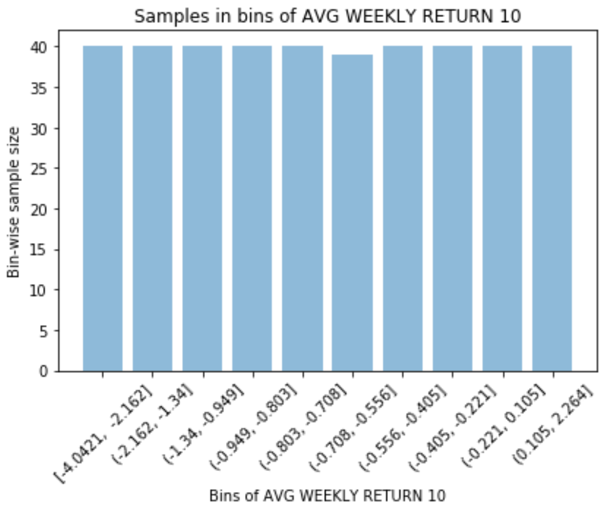


```
--------------------------------------------------------------------------------
----------------------------------
```

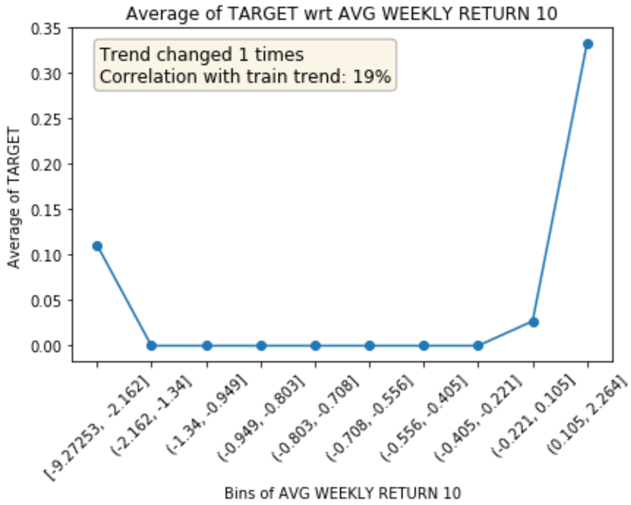
Plots for AVG WEEKLY RETURN 10

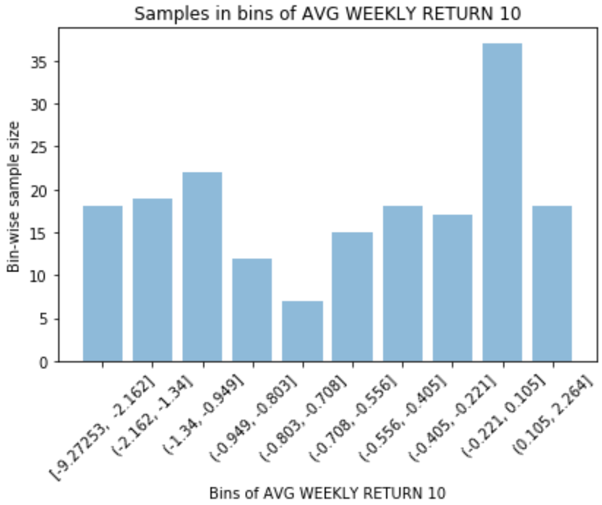
Train data plots





Test data plots

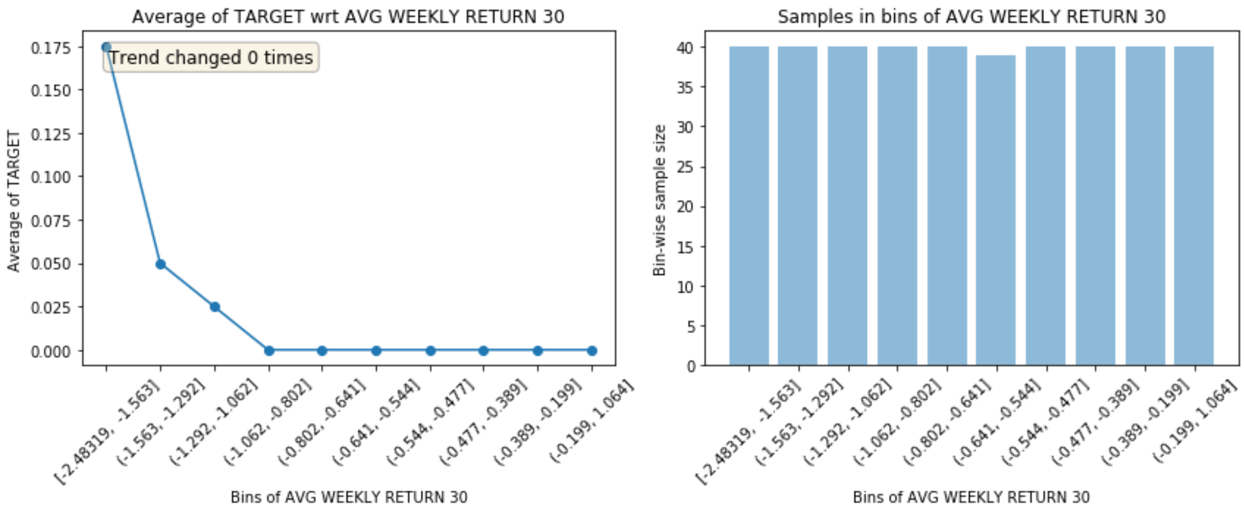




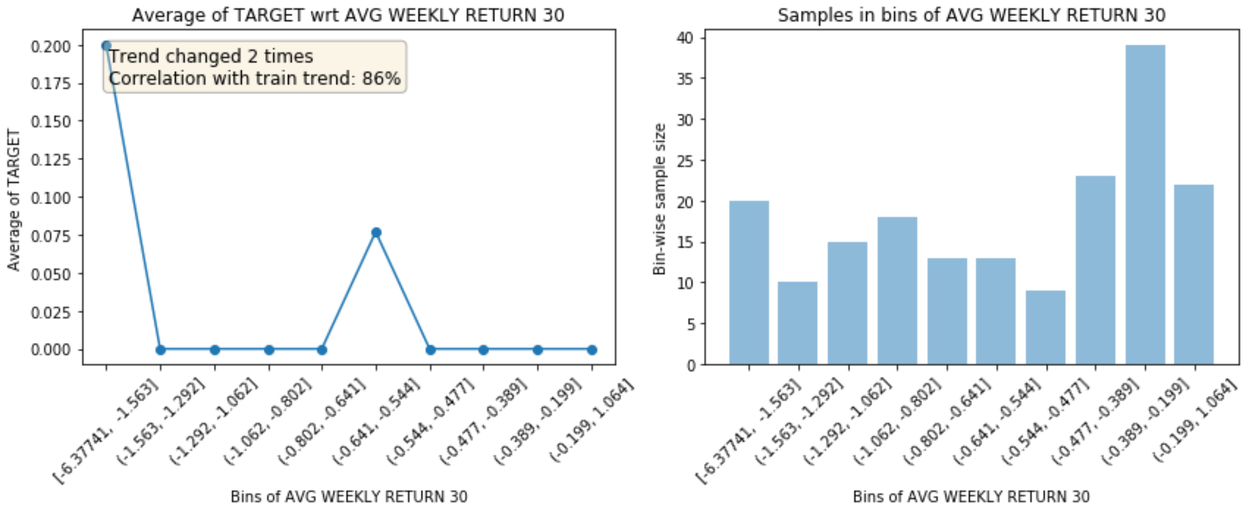
```
------------------------------------------------------------------------------
---------------------------------
```

Plots for AVG WEEKLY RETURN 30

Train data plots



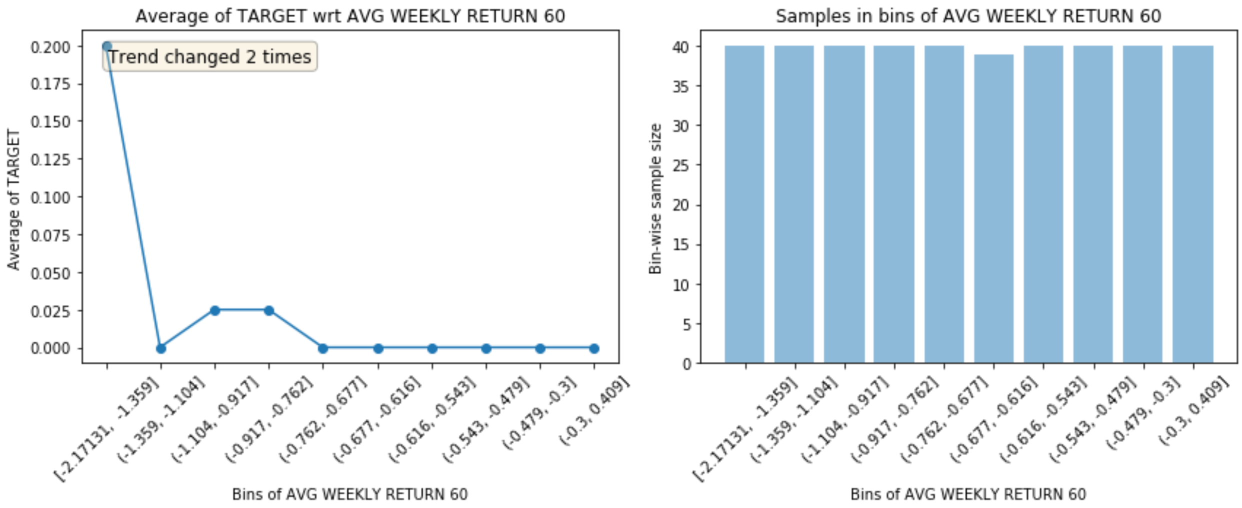
Test data plots



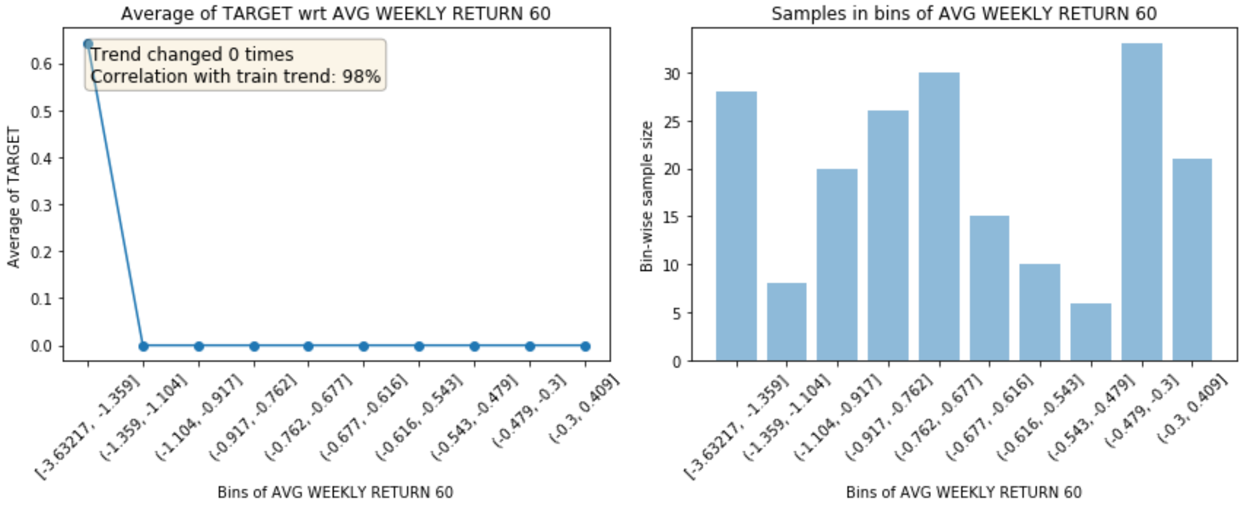
```
------------------------------------------------------------------------------
---------------------------------
```

Plots for AVG WEEKLY RETURN 60

Train data plots



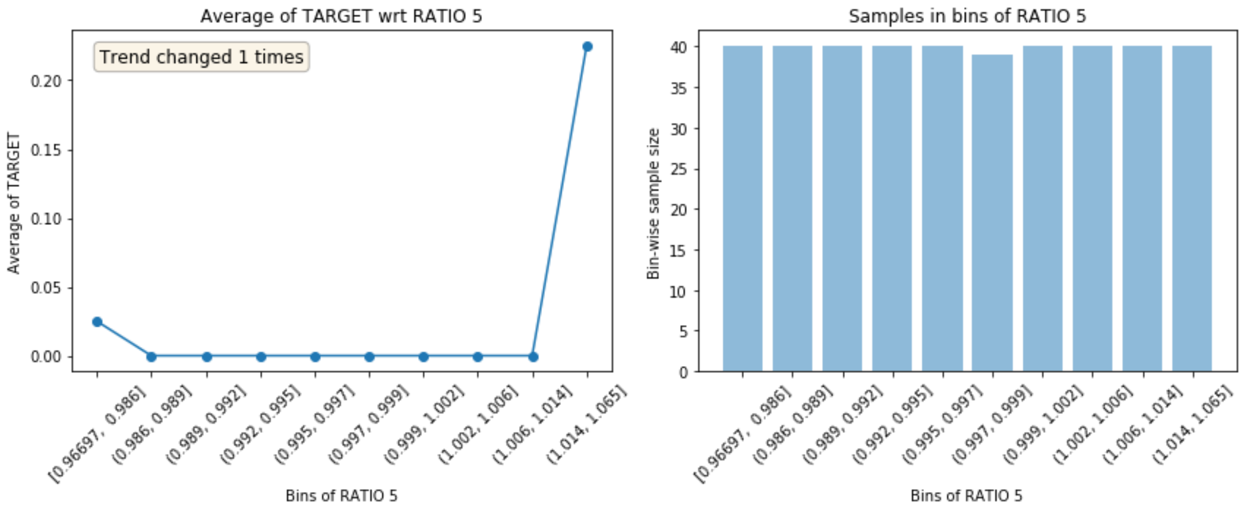
```
                                                                    Test data plots
```



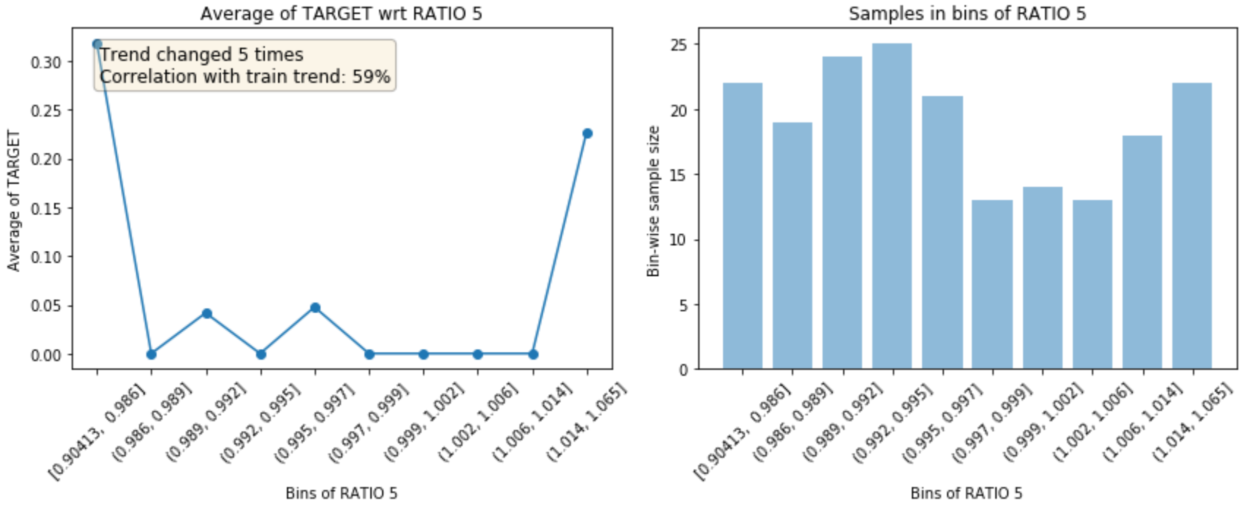
```
--------------------------------------------------------------------------------
----------------------------------



                                                                   Plots for RATIO 5
                                                                    Train data plots
```



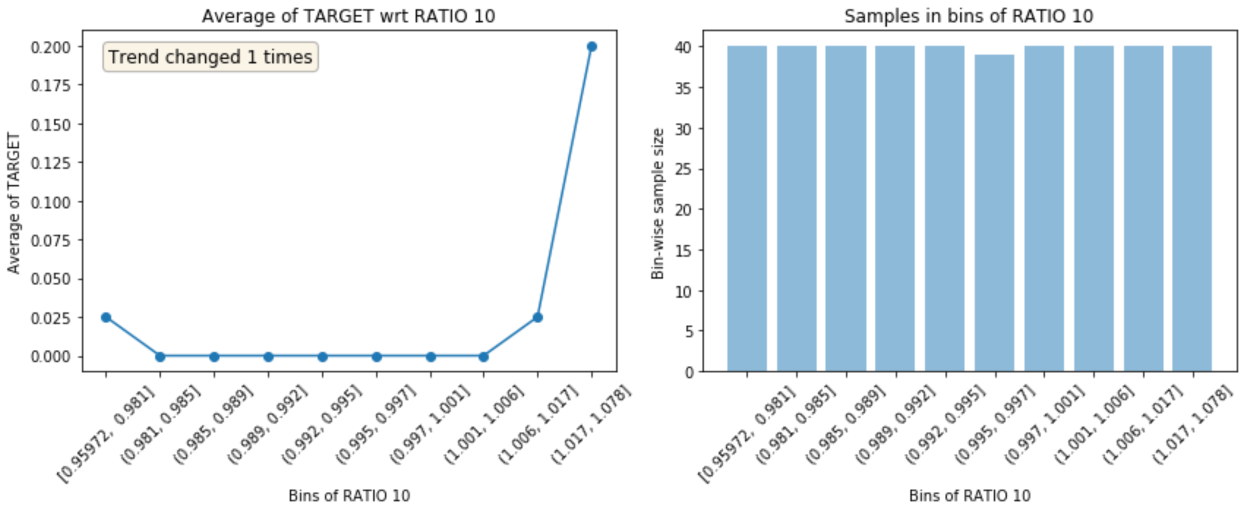
Test data plots



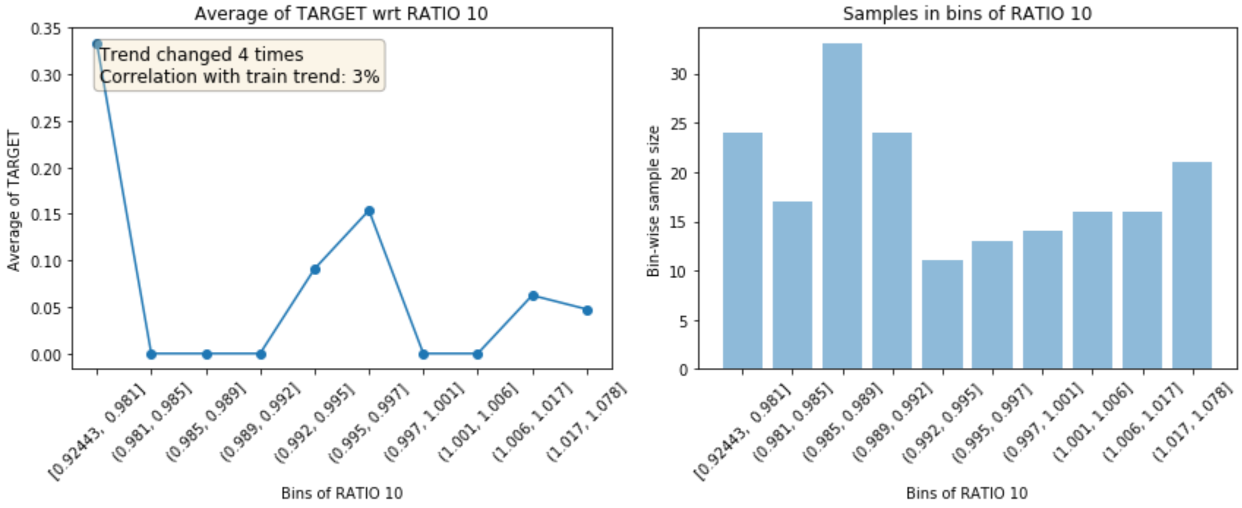
```
--------------------------------------------------------------------------------
----------------------------------
```

Plots for RATIO 10

Train data plots



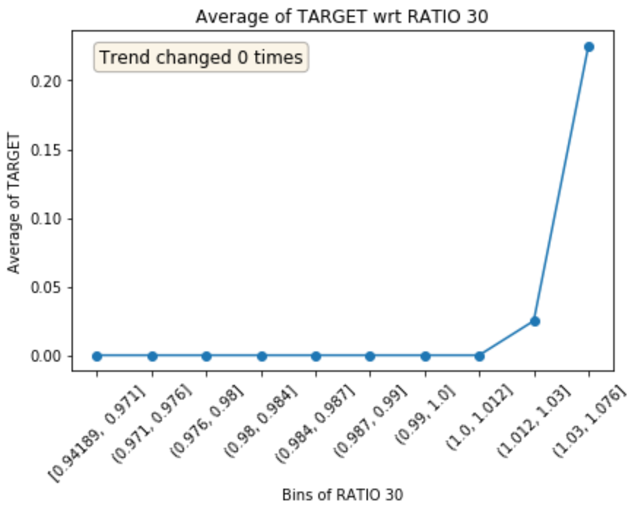
Test data plots

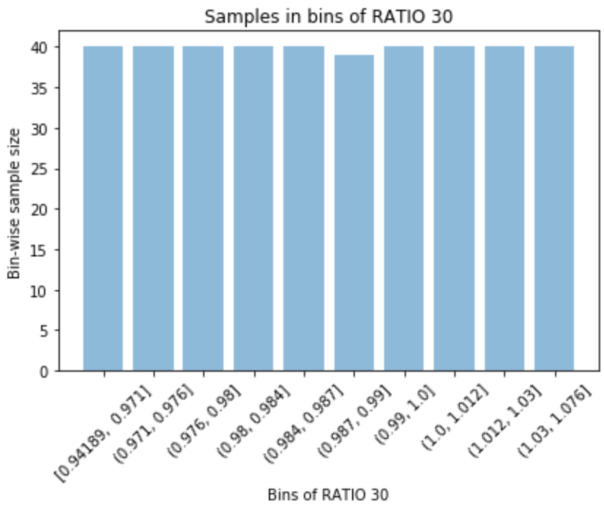


```
----------------------------------------------------------------------------
--------------------------------
```

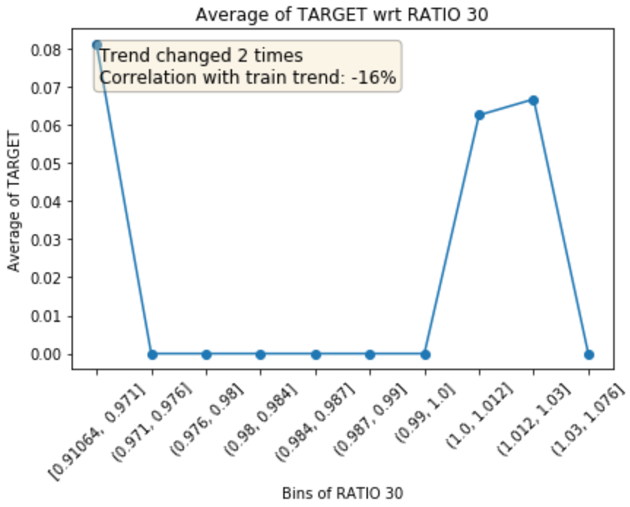
Plots for RATIO 30

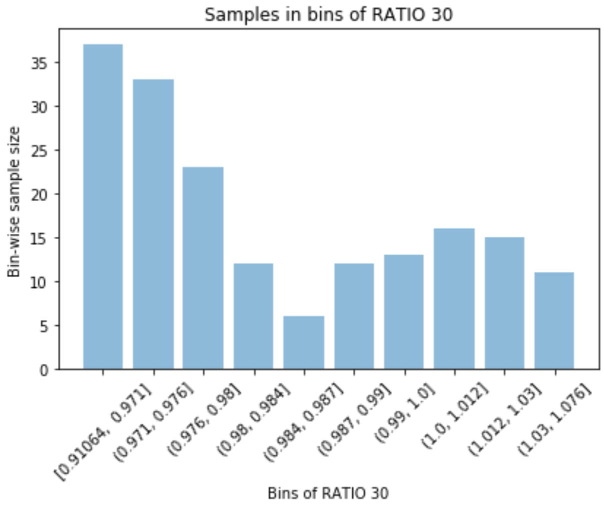
Train data plots





Test data plots

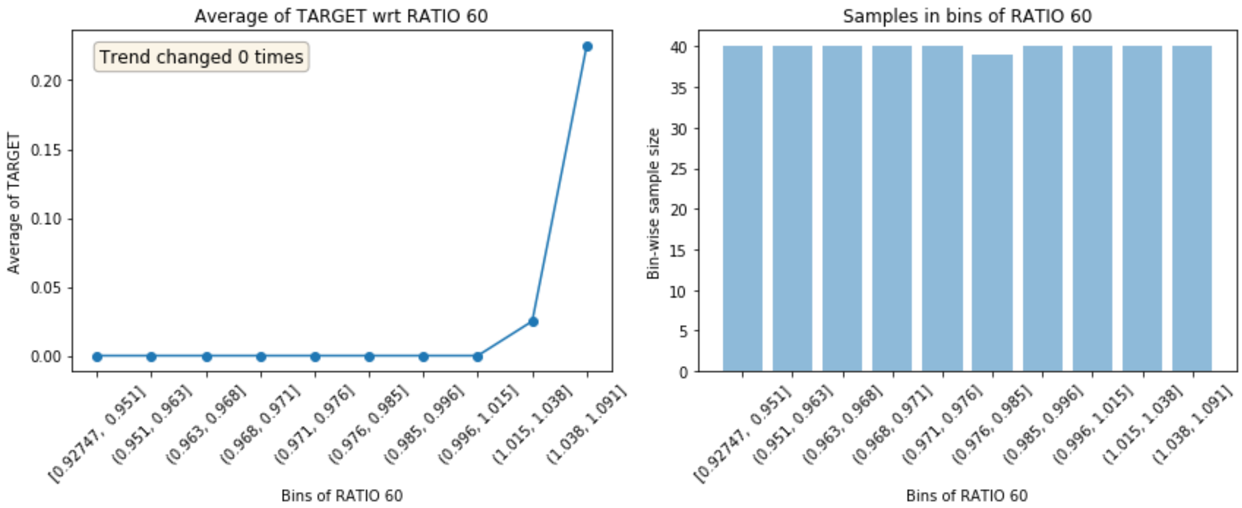




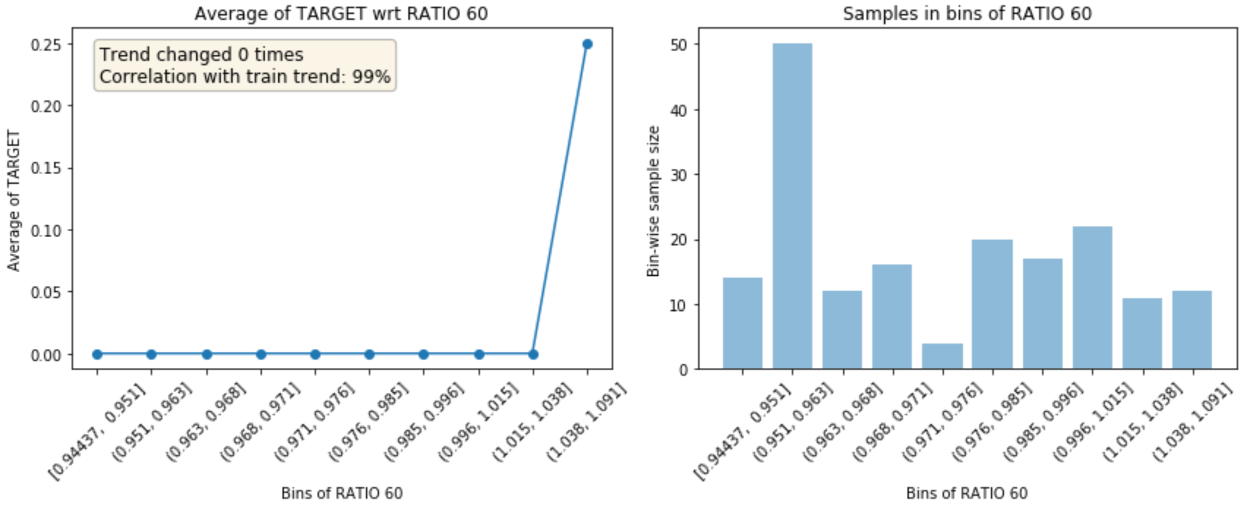
```
----------------------------------------------------------------------------
--------------------------------
```

Plots for RATIO 60

Train data plots



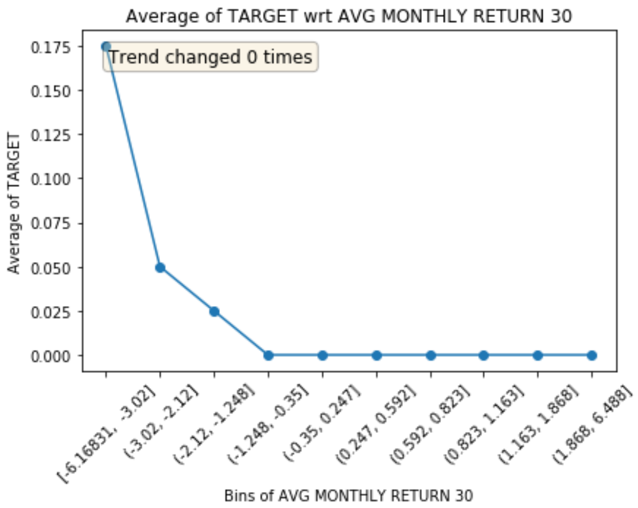
Test data plots

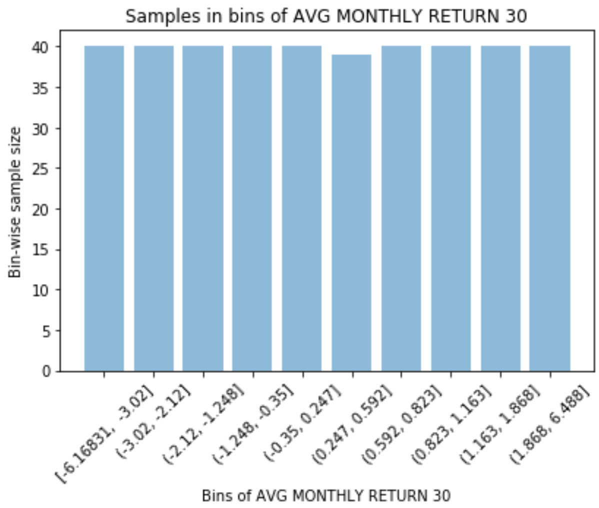


```
--------------------------------------------------------------------------------
----------------------------------
```

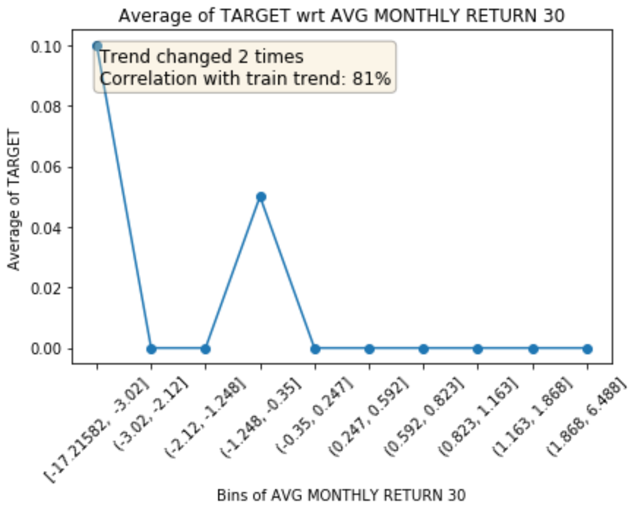
Plots for AVG MONTHLY RETURN 30

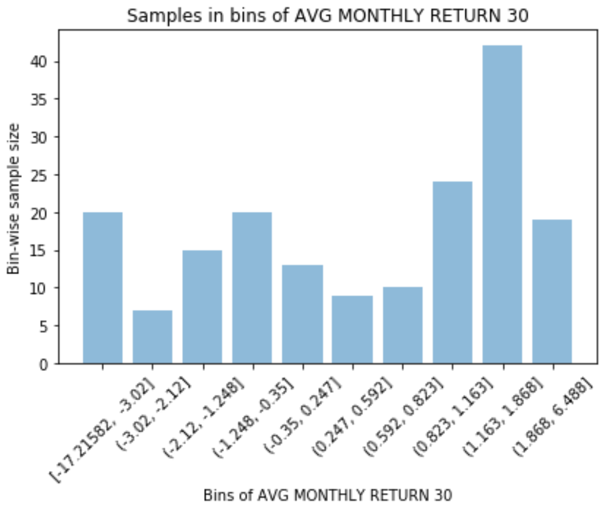
Train data plots





Test data plots

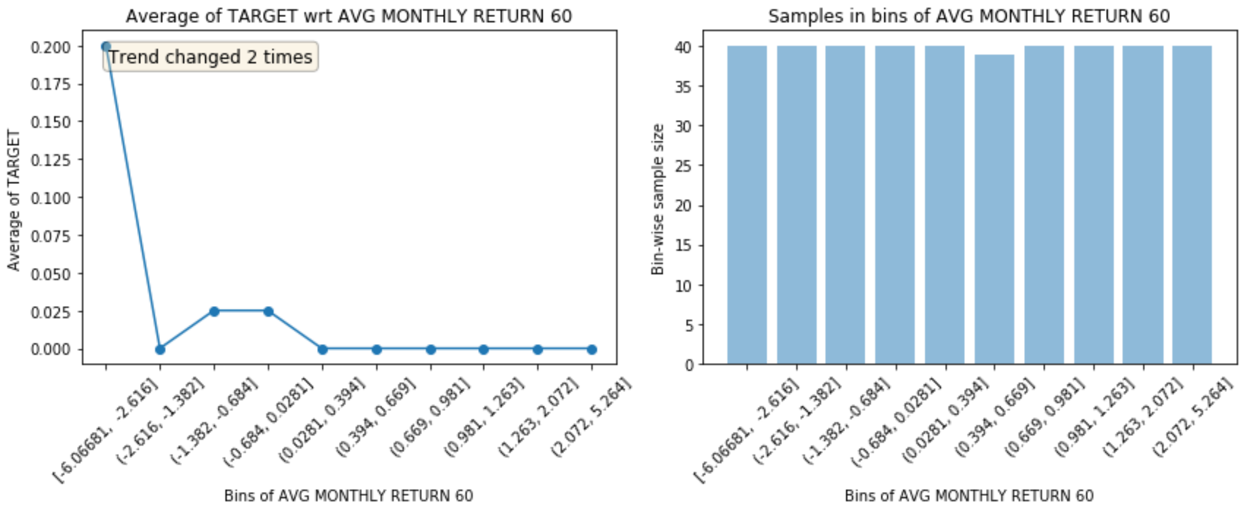




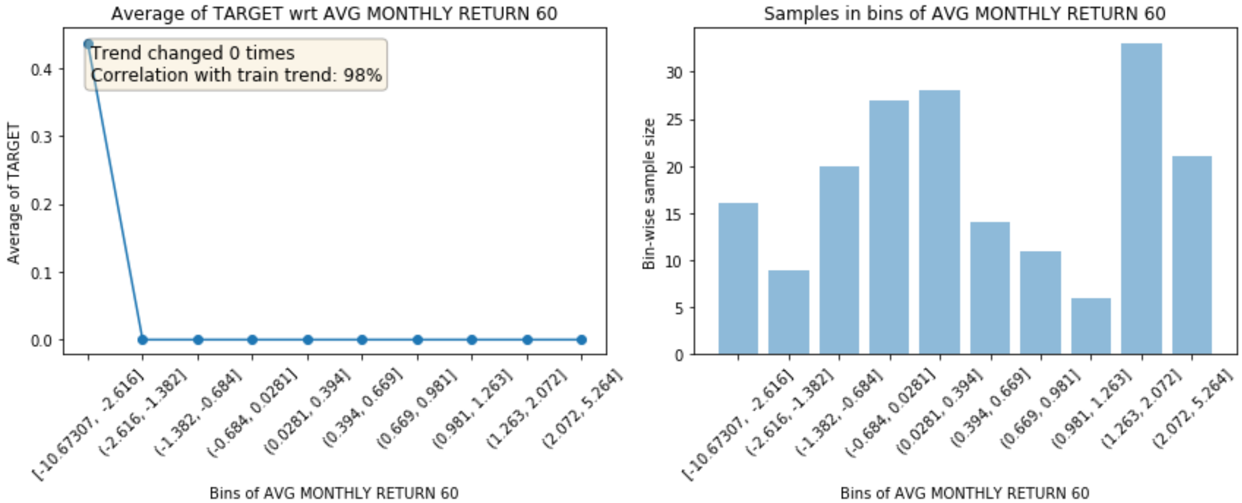
```
--------------------------------------------------------------------------------
----------------------------------
```

Plots for AVG MONTHLY RETURN 60

Train data plots

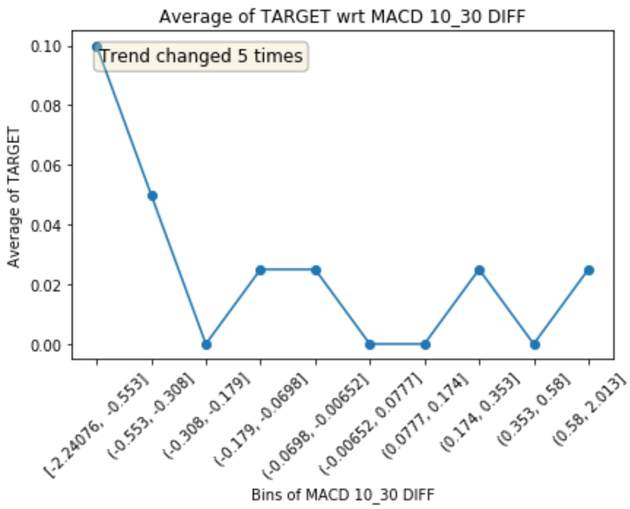


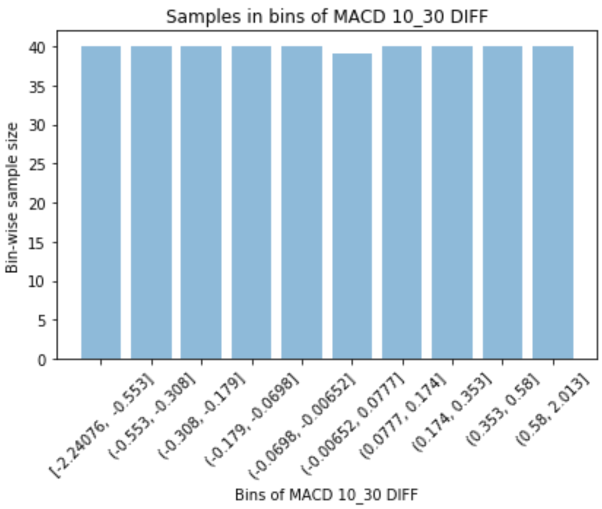
Test data plots



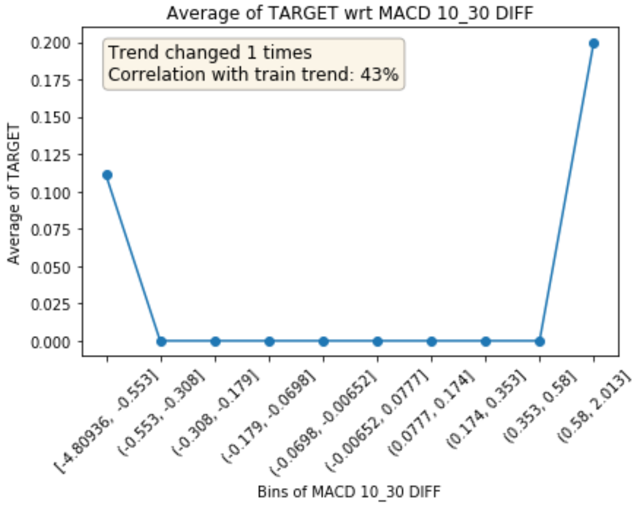
Plots for MACD 10_30 DIFF

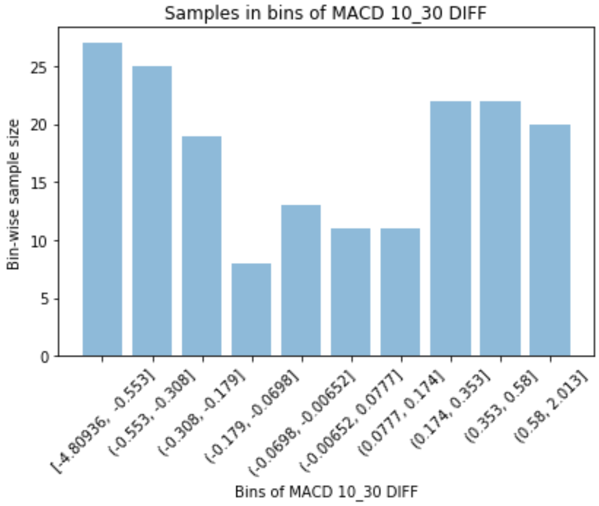
Train data plots





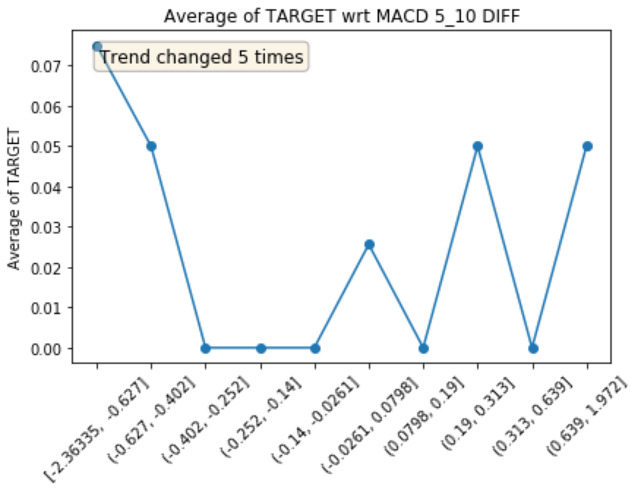
Test data plots

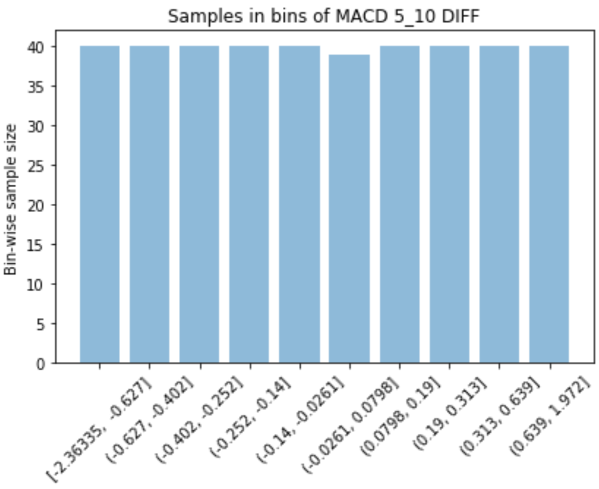




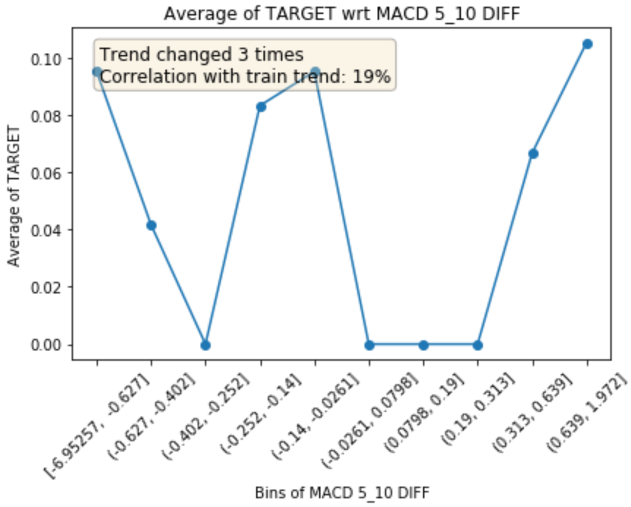
Plots for MACD 5_10 DIFF

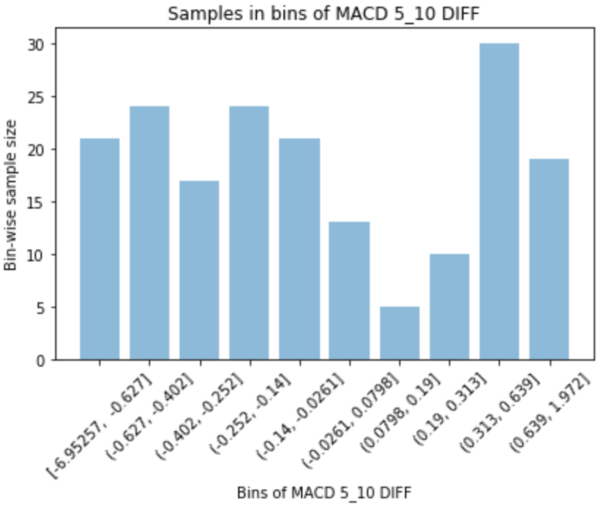
Train data plots



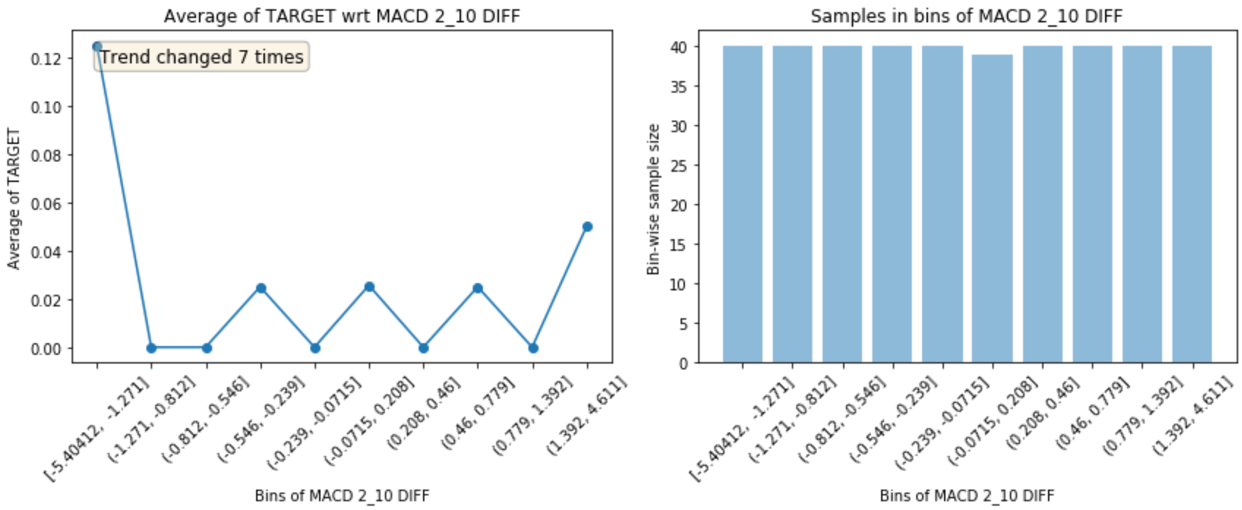


Test data plots

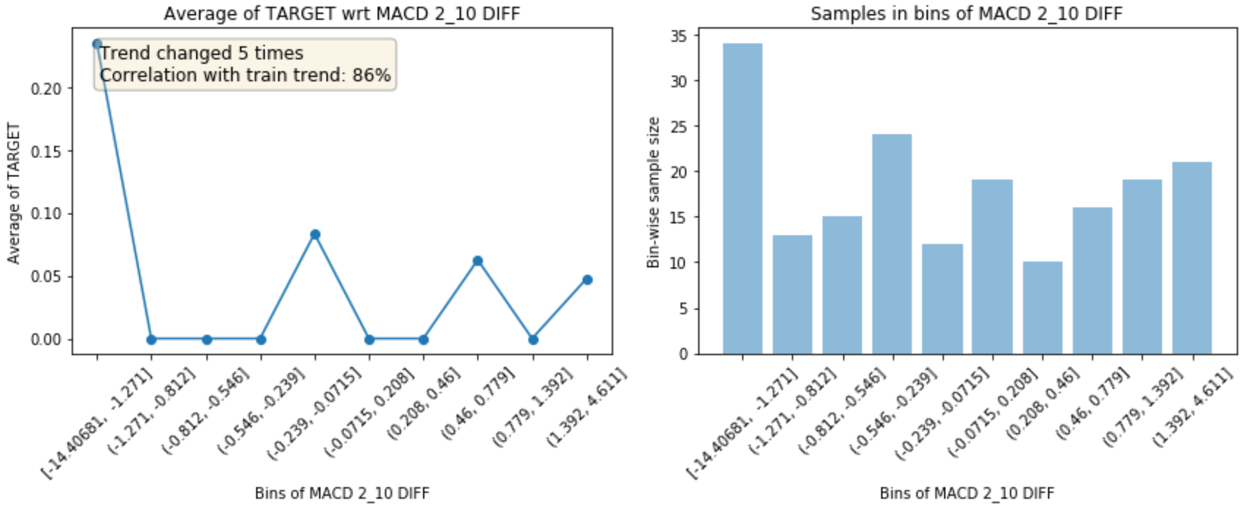




Plots for MACD 2_10 DIFF
Train data plots



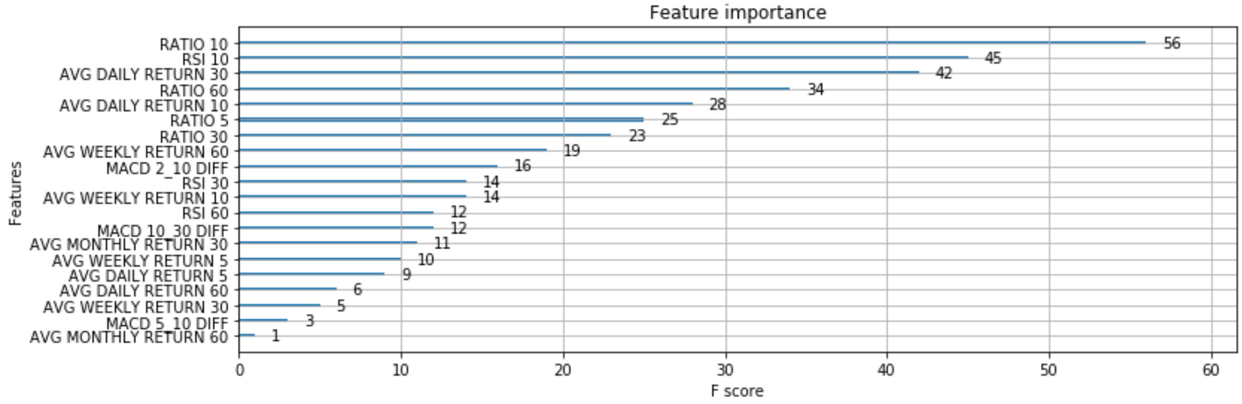
```
                                                         Test data plots
```



```
-----------------------------------------------------------------------------
---------------------------------



STEP THREE DONE
STEP FOUR DONE
```



```
Thresh=0.000, n=20, AUC: 68.04%
Thresh=0.000, n=20, Precision: 53.33%
Thresh=0.005, n=19, AUC: 68.04%
Thresh=0.005, n=19, Precision: 53.33%
Thresh=0.010, n=18, AUC: 68.04%
Thresh=0.010, n=18, Precision: 53.33%
Thresh=0.017, n=17, AUC: 68.04%
Thresh=0.017, n=17, Precision: 53.33%
Thresh=0.020, n=16, AUC: 68.04%
Thresh=0.020, n=16, Precision: 53.33%
Thresh=0.021, n=15, AUC: 70.54%
Thresh=0.021, n=15, Precision: 56.25%
Thresh=0.022, n=14, AUC: 70.54%
Thresh=0.022, n=14, Precision: 56.25%
Thresh=0.022, n=13, AUC: 72.77%
Thresh=0.022, n=13, Precision: 55.56%
Thresh=0.025, n=12, AUC: 72.77%
Thresh=0.025, n=12, Precision: 55.56%
Thresh=0.032, n=11, AUC: 70.54%
Thresh=0.032, n=11, Precision: 56.25%
Thresh=0.035, n=10, AUC: 68.04%
Thresh=0.035, n=10, Precision: 53.33%
Thresh=0.036, n=9, AUC: 68.04%
Thresh=0.036, n=9, Precision: 53.33%
Thresh=0.046, n=8, AUC: 80.54%
Thresh=0.046, n=8, Precision: 65.00%
Thresh=0.053, n=7, AUC: 78.32%
Thresh=0.053, n=7, Precision: 66.67%
Thresh=0.062, n=6, AUC: 78.32%
Thresh=0.062, n=6, Precision: 66.67%
Thresh=0.063, n=5, AUC: 78.32%
Thresh=0.063, n=5, Precision: 66.67%
Thresh=0.077, n=4, AUC: 75.82%
Thresh=0.077, n=4, Precision: 64.71%
Thresh=0.116, n=3, AUC: 92.21%
Thresh=0.116, n=3, Precision: 64.29%
Thresh=0.151, n=2, AUC: 92.21%
Thresh=0.151, n=2, Precision: 64.29%
Thresh=0.184, n=1, AUC: 85.27%
Thresh=0.184, n=1, Precision: 65.22%
-----------------ROUNDING THRESHOLD OF 0.85---------------------
Thresh=0.000, n=20, AUC: 63.60%
Thresh=0.000, n=20, Precision: 54.55%
Thresh=0.005, n=19, AUC: 63.60%
```

```
Thresh=0.005, n=19, Precision: 54.55%
Thresh=0.010, n=18, AUC: 63.60%
Thresh=0.010, n=18, Precision: 54.55%
Thresh=0.017, n=17, AUC: 63.32%
Thresh=0.017, n=17, Precision: 50.00%
Thresh=0.020, n=16, AUC: 63.60%
Thresh=0.020, n=16, Precision: 54.55%
Thresh=0.021, n=15, AUC: 65.54%
Thresh=0.021, n=15, Precision: 50.00%
Thresh=0.022, n=14, AUC: 65.82%
Thresh=0.022, n=14, Precision: 53.85%
Thresh=0.022, n=13, AUC: 65.82%
Thresh=0.022, n=13, Precision: 53.85%
Thresh=0.025, n=12, AUC: 65.82%
Thresh=0.025, n=12, Precision: 53.85%
Thresh=0.032, n=11, AUC: 65.82%
Thresh=0.032, n=11, Precision: 53.85%
Thresh=0.035, n=10, AUC: 63.04%
Thresh=0.035, n=10, Precision: 46.15%
Thresh=0.036, n=9, AUC: 63.32%
Thresh=0.036, n=9, Precision: 50.00%
Thresh=0.046, n=8, AUC: 70.82%
Thresh=0.046, n=8, Precision: 60.00%
Thresh=0.053, n=7, AUC: 70.82%
Thresh=0.053, n=7, Precision: 60.00%
Thresh=0.062, n=6, AUC: 63.60%
Thresh=0.062, n=6, Precision: 54.55%
Thresh=0.063, n=5, AUC: 63.60%
Thresh=0.063, n=5, Precision: 54.55%
Thresh=0.077, n=4, AUC: 51.66%
Thresh=0.077, n=4, Precision: 25.00%
Thresh=0.116, n=3, AUC: 85.54%
Thresh=0.116, n=3, Precision: 68.18%
Thresh=0.151, n=2, AUC: 85.54%
Thresh=0.151, n=2, Precision: 68.18%
Thresh=0.184, n=1, AUC: 85.54%
Thresh=0.184, n=1, Precision: 68.18%
Select feature threshold: .184
STEP SIX DONE
    train-auc-mean  train-auc-std  test-auc-mean  test-auc-std
0         0.934073       0.012742       0.912918      0.056852
1         0.934891       0.012743       0.915482      0.057538
2         0.935107       0.012322       0.914094      0.056306
3         0.935343       0.012613       0.918392      0.060908
```

```
4         0.954882       0.021633       0.917886      0.061265
5         0.955229       0.020675       0.919688      0.063481
6         0.954158       0.021017       0.917886      0.061265
7         0.965381       0.014039       0.920383      0.063993
8         0.965018       0.014204       0.920383      0.063993
9         0.964772       0.013606       0.921771      0.065093
10        0.965235       0.013855       0.921771      0.065093
11        0.965473       0.014078       0.920383      0.063993
12        0.965789       0.014372       0.920383      0.063993
13        0.967129       0.015261       0.920383      0.063993
14        0.967280       0.015442       0.920383      0.063993
15        0.972680       0.014823       0.918362      0.065574
16        0.978318       0.005686       0.943853      0.045434
17        0.978330       0.005547       0.943853      0.045434
18        0.978661       0.005292       0.943853      0.045434
19        0.978999       0.005316       0.942843      0.047227
20        0.979616       0.005253       0.942843      0.047227
21        0.979827       0.004520       0.940882      0.047051
22        0.979737       0.004770       0.940882      0.047051
23        0.980080       0.004343       0.942271      0.047947
24        0.980080       0.004343       0.942271      0.047947
25        0.980344       0.004112       0.942271      0.047947
26        0.981134       0.004106       0.942271      0.047947
27        0.981191       0.003999       0.942271      0.047947
28        0.981322       0.003944       0.942271      0.047947
29        0.981594       0.003989       0.942271      0.047947
30        0.981918       0.004379       0.942271      0.047947
31        0.981918       0.004379       0.942271      0.047947
32        0.981918       0.004379       0.942271      0.047947
33        0.981884       0.004376       0.942271      0.047947
34        0.982232       0.004351       0.944231      0.048063
35        0.982483       0.004142       0.944231      0.048063
36        0.982468       0.003976       0.944231      0.048063
37        0.982050       0.004304       0.945130      0.048999
Ideal n_estimators:  38
TEST EVERYTHING
CV Results:  {'mean_fit_time': array([0.07538342, 0.06593003, 0.07184958, 0.0
628716 , 0.07112079,
       0.06500635, 0.07077556, 0.06449528, 0.07055173, 0.06207399,
       0.06701808, 0.06010919, 0.07063775, 0.05942516, 0.06817594,
       0.06086798, 0.06642413, 0.05788717, 0.06498241, 0.04673333]), 'std_fit
_time': array([0.01317998, 0.00918446, 0.00772216, 0.00633621, 0.0089974 ,
       0.00699166, 0.01244017, 0.00397761, 0.0088887 , 0.00913979,
       0.00909656, 0.00637936, 0.00816238, 0.00665727, 0.00883636,
       0.00510129, 0.00818936, 0.00642921, 0.01225445, 0.00704183]), 'mean_sc
ore_time': array([0.00402837, 0.00357394, 0.00351834, 0.00350928, 0.00328298,
       0.00336828, 0.0032548 , 0.00338974, 0.00329733, 0.00328259,
       0.00325847, 0.00331426, 0.00318909, 0.00327311, 0.00329547,
       0.00322485, 0.00321827, 0.00321159, 0.00316215, 0.00329976]), 'std_sco
re_time': array([0.00076558, 0.00027476, 0.00029726, 0.0003782 , 0.00005155,
       0.00015387, 0.00003926, 0.00021198, 0.00008755, 0.00013374,
       0.00003283, 0.00014609, 0.00003143, 0.00008906, 0.00005253,
       0.00004647, 0.00002661, 0.00004712, 0.00007618, 0.00013041]), 'param_g
amma': masked_array(data=[0.0, 0.0, 0.0, 0.0, 0.1, 0.1, 0.1, 0.1, 0.2, 0.2, 0
.2,
                   0.2, 0.3, 0.3, 0.3, 0.3, 0.4, 0.4, 0.4, 0.4],
             mask=[False, False, False, False, False, False, False, False,
                   False, False, False, False, False, False, False, False,
                   False, False, False, False],
       fill_value='?',
            dtype=object), 'param_max_depth': masked_array(data=[9, 9, 10, 10
, 9, 9, 10, 10, 9, 9, 10, 10, 9, 9, 10, 10,
                   9, 9, 10, 10],
             mask=[False, False, False, False, False, False, False, False,
                   False, False, False, False, False, False, False, False,
                   False, False, False, False],
       fill_value='?',
            dtype=object), 'param_min_child_weight': masked_array(data=[1, 2,
1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2,
                   1, 2],
             mask=[False, False, False, False, False, False, False, False,
                   False, False, False, False, False, False, False, False,
                   False, False, False, False],
       fill_value='?',
            dtype=object), 'params': [{'gamma': 0.0, 'max_depth': 9, 'min_chi
ld_weight': 1}, {'gamma': 0.0, 'max_depth': 9, 'min_child_weight': 2}, {'gamm
a': 0.0, 'max_depth': 10, 'min_child_weight': 1}, {'gamma': 0.0, 'max_depth':
10, 'min_child_weight': 2}, {'gamma': 0.1, 'max_depth': 9, 'min_child_weight'
: 1}, {'gamma': 0.1, 'max_depth': 9, 'min_child_weight': 2}, {'gamma': 0.1, '
max_depth': 10, 'min_child_weight': 1}, {'gamma': 0.1, 'max_depth': 10, 'min_
child_weight': 2}, {'gamma': 0.2, 'max_depth': 9, 'min_child_weight': 1}, {'g
amma': 0.2, 'max_depth': 9, 'min_child_weight': 2}, {'gamma': 0.2, 'max_depth
': 10, 'min_child_weight': 1}, {'gamma': 0.2, 'max_depth': 10, 'min_child_wei
ght': 2}, {'gamma': 0.3, 'max_depth': 9, 'min_child_weight': 1}, {'gamma': 0.
3, 'max_depth': 9, 'min_child_weight': 2}, {'gamma': 0.3, 'max_depth': 10, 'm
in_child_weight': 1}, {'gamma': 0.3, 'max_depth': 10, 'min_child_weight': 2},
{'gamma': 0.4, 'max_depth': 9, 'min_child_weight': 1}, {'gamma': 0.4, 'max_de
pth': 9, 'min_child_weight': 2}, {'gamma': 0.4, 'max_depth': 10, 'min_child_w
```

```
eight': 1}, {'gamma': 0.4, 'max_depth': 10, 'min_child_weight': 2}], 'split0_
test_score': array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1
., 1., 1.,
       1., 1., 1.]), 'split1_test_score': array([1., 1., 1., 1., 1., 1., 1.,
1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1.]), 'split2_test_score': array([0.98611111, 0.99305556, 0.98
611111, 0.99305556, 0.98611111,
       0.99652778, 0.98611111, 0.99652778, 0.98611111, 0.99652778,
       0.98611111, 0.99652778, 0.98611111, 0.99652778, 0.98611111,
       0.99652778, 0.98611111, 0.99652778, 0.98611111, 0.99652778]), 'split3_
test_score': array([1.        , 0.98611111, 1.        , 0.98611111, 1.
,
       0.98611111, 1.        , 0.98611111, 1.        , 0.98611111,
       1.        , 0.98611111, 1.        , 0.98611111, 1.        ,
       0.98611111, 1.        , 0.98611111, 1.        , 0.98611111]), 'split4_
test_score': array([0.32857143, 0.32857143, 0.32857143, 0.32857143, 0.3285714
3,
       0.32857143, 0.32857143, 0.32857143, 0.32857143, 0.32857143,
       0.32857143, 0.32857143, 0.32857143, 0.32857143, 0.32857143,
       0.32857143, 0.32857143, 0.32857143, 0.32857143, 0.32857143]), 'mean_te
st_score': array([0.86293651, 0.86154762, 0.86293651, 0.86154762, 0.86293651,
       0.86224206, 0.86293651, 0.86224206, 0.86293651, 0.86224206,
       0.86293651, 0.86224206, 0.86293651, 0.86224206, 0.86293651,
       0.86224206, 0.86293651, 0.86224206, 0.86293651, 0.86224206]), 'std_tes
t_score': array([0.26723668, 0.26653786, 0.26723668, 0.26653786, 0.26723668,
       0.26688388, 0.26723668, 0.26688388, 0.26723668, 0.26688388,
       0.26723668, 0.26688388, 0.26723668, 0.26688388, 0.26723668,
       0.26688388, 0.26723668, 0.26688388, 0.26723668, 0.26688388]), 'rank_te
st_score': array([ 1, 19,  1, 19,  1, 11,  1, 11,  1, 11,  1, 11,  1, 11,  1,
11,  1,
       11,  1, 11], dtype=int32)}
Best Params:  {'gamma': 0.0, 'max_depth': 9, 'min_child_weight': 1}
Best Score:  0.862936507936508
    train-auc-mean  train-auc-std  test-auc-mean  test-auc-std
0         0.934073       0.012742       0.912918      0.056852
1         0.934891       0.012743       0.915482      0.057538
2         0.935107       0.012322       0.914094      0.056306
3         0.935343       0.012613       0.918392      0.060908
4         0.954882       0.021633       0.917886      0.061265
5         0.955229       0.020675       0.919688      0.063481
6         0.954158       0.021017       0.917886      0.061265
7         0.965381       0.014039       0.920383      0.063993
8         0.965018       0.014204       0.920383      0.063993
9         0.964772       0.013606       0.921771      0.065093
```

```
10        0.965235       0.013855       0.921771      0.065093
11        0.965473       0.014078       0.920383      0.063993
12        0.965789       0.014372       0.920383      0.063993
13        0.967129       0.015261       0.920383      0.063993
14        0.967280       0.015442       0.920383      0.063993
15        0.972680       0.014823       0.918362      0.065574
16        0.978318       0.005686       0.943853      0.045434
17        0.978330       0.005547       0.943853      0.045434
18        0.978661       0.005292       0.943853      0.045434
19        0.978999       0.005316       0.942843      0.047227
20        0.979616       0.005253       0.942843      0.047227
21        0.979827       0.004520       0.940882      0.047051
22        0.979737       0.004770       0.940882      0.047051
23        0.980080       0.004343       0.942271      0.047947
24        0.980080       0.004343       0.942271      0.047947
25        0.980344       0.004112       0.942271      0.047947
26        0.981134       0.004106       0.942271      0.047947
27        0.981191       0.003999       0.942271      0.047947
28        0.981322       0.003944       0.942271      0.047947
29        0.981594       0.003989       0.942271      0.047947
30        0.981918       0.004379       0.942271      0.047947
31        0.981918       0.004379       0.942271      0.047947
32        0.981918       0.004379       0.942271      0.047947
33        0.981884       0.004376       0.942271      0.047947
34        0.982232       0.004351       0.944231      0.048063
35        0.982483       0.004142       0.944231      0.048063
36        0.982468       0.003976       0.944231      0.048063
37        0.982050       0.004304       0.945130      0.048999
Ideal n_estimators:  38
select model name: trial2
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.8, eval_metric='auc',
              gamma=0.0, gpu_id=-1, importance_type='gain',
              interaction_constraints='', learning_rate=0.1, max_delta_step=0
,
              max_depth=9, min_child_weight=1, missing=nan,
              monotone_constraints='()', n_estimators=38, n_jobs=4, nthread=4
,
              num_parallel_tree=1, objective='binary:logistic', random_state=
27,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=27,
              subsample=0.8, tree_method='exact', use_label_encoder=False,
              validate_parameters=1, ...)
STEP SEVEN DONE
```

```
AUC: 49.6987951807229
Precision: 0.0
Accuracy: 82.5
ROUNDING THRESHOLD OF 0.85
AUC: 49.6987951807229
Precision: 0.0
Accuracy: 82.5
STEP EIGHT DONE
```