3_2_MSMWD_cross_validataion_and_training_set2

November 28, 2021

[1]:
```
print('importing modules..')
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import time
#Training and evaluation
from sklearn.model_selection import train_test_split
#from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import log_loss
from sklearn.calibration import CalibratedClassifierCV
import warnings
warnings.filterwarnings("ignore")

from sklearn.feature_selection import SelectKBest, chi2

from xgboost import XGBRFClassifier, XGBClassifier
from hyperopt import hp, STATUS_OK, Trials, fmin, tpe
# https://www.appliedaicourse.com/lecture/11/
↪applied-machine-learning-online-course/3255/
↪models-on-all-features-randomforest-and-xgboost/7/
↪module-6-machine-learning-real-world-case-studies
```

```
importing modules..
```

[2]:
```
print('loading data...')
X = np.load('feature_set2.npy')
Y = np.load('target.npy')
```

```
loading data…
```

[3]:
```
x_train, x_test, y_train, y_test = train_test_split(X,Y,test_size=0.2,␣
↪random_state=11)
x_train, x_cv, y_train, y_cv = train_test_split(x_train,y_train,test_size=0.25,␣
↪random_state=11)
print('Train:cv:test ::',y_train.shape[0],':',y_cv.shape[0],':',y_test.shape[0])
print('Shape of training data', x_train.shape)
```

```
Train:cv:test :: 6520 : 2174 : 2174
Shape of training data (6520, 620)
```

[4]:

```
test_class_count = pd.DataFrame(y_test.flatten())[0].value_counts()
train_class_count = pd.DataFrame(y_train.flatten())[0].value_counts()
cv_class_count = pd.DataFrame(y_cv.flatten())[0].value_counts()
class_count = pd.concat((test_class_count,train_class_count),axis=1)
class_count.columns = ['test_imbalance','train_imbalance']
class_count = pd.concat((class_count,cv_class_count),axis=1)
class_count.columns = ['test_imbalance','train_imbalance','cv_imbalance']

train_test_difference = class_count/class_count.sum()
print("Train test differences - Distribution of classes in train, cv and test␣
↪set")
print(train_test_difference)
```

```
Train test differences - Distribution of classes in train, cv and test set
   test_imbalance  train_imbalance  cv_imbalance
3        0.280129         0.271472      0.258970
2        0.233211         0.228834      0.220331
1        0.137994         0.141258      0.147194
8        0.109476         0.114110      0.113155
9        0.093836         0.091411      0.097976
6        0.064397         0.069325      0.073137
4        0.040478         0.043558      0.047378
7        0.037718         0.035890      0.037718
5        0.002760         0.004141      0.004140
```

[5]:

```
# Hyperparameter tuning: https://www.analyticsvidhya.com/blog/2020/09/
↪alternative-hyperparameter-optimization-technique-you-need-to-know-hyperopt/
# Hyperopt: http://hyperopt.github.io/hyperopt/
# Hyperopt gihub: https://github.com/hyperopt/hyperopt/wiki/fmin
print('Tuning..')

# narrow search on multiple parameters
space = { 'max_depth': hp.choice('max_depth',[1,3,5,7]),
           'learning_rate': hp.choice('learning_rate',[.5,1]),
           'subsample': hp.choice('subsample',[0.4,.5,.6,.7,1]),
         'colsample_bynode':hp.choice('colsample_bynode',[0.4,.5,.6,.7,1]),
           'n_estimators': hp.choice('n_estimators',np.arange(30,400))}

def hyperparameter_tuning(params):
    '''return dict should have train loss with key loss for the fmin to␣
↪minimize. Other params are for analysis'''
    clf = XGBClassifier(eval_metric='mlogloss',**params)
    print(params, end = "=>")
    #tick = time.time(),
```

```
    clf.fit(x_train,y_train.ravel())
    sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
    sig_clf.fit(x_train, y_train.ravel())
    train_loss = log_loss(y_train,sig_clf.predict_proba(x_train))
    loss = log_loss(y_cv,sig_clf.predict_proba(x_cv))
    #print('Time:',time.time() - tick)
    dic = {"loss": loss, "status": STATUS_OK,'train_loss':train_loss}
    dic.update(params)
    print(dic)
    return dic

trials = Trials()
best = fmin(fn=hyperparameter_tuning, space = space, trials=trials,  algo=tpe.
 ↪suggest, max_evals=50)
#TRY THIS: use_label_encoder=False when constructing XGBClassifier

tuned = pd.DataFrame(trials.results).sort_values(by='loss',axis=0)
tuned.to_csv("feature_set2_50_trials.csv")
print(tuned)
print('done')
```

```
Tuning..
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
290, 'subsample': 0.5}
=>
{'loss': 0.023425612423347248, 'status': 'ok', 'train_loss':
0.011407115434723836, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 290, 'subsample': 0.5}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
396, 'subsample': 0.5}
=>
{'loss': 0.02087685188421868, 'status': 'ok', 'train_loss':
0.009884001423295003, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 396, 'subsample': 0.5}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 3, 'n_estimators':
378, 'subsample': 0.4}
=>
{'loss': 0.023025653743884524, 'status': 'ok', 'train_loss':
0.010746776877673454, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
3, 'n_estimators': 378, 'subsample': 0.4}
{'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 5, 'n_estimators': 104,
'subsample': 0.6}
=>
{'loss': 0.02657286743619554, 'status': 'ok', 'train_loss':
0.010455834729584498, 'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 5,
'n_estimators': 104, 'subsample': 0.6}
{'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth': 1, 'n_estimators':
```

260, 'subsample': 0.7}
=>
{'loss': 0.06698892162191686, 'status': 'ok', 'train_loss': 0.0530363844099486,
'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth': 1, 'n_estimators':
260, 'subsample': 0.7}
{'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth': 1, 'n_estimators':
193, 'subsample': 0.6}
=>
{'loss': 0.02214372259672183, 'status': 'ok', 'train_loss': 0.01015654648630132,
'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth': 1, 'n_estimators':
193, 'subsample': 0.6}
{'colsample_bynode': 0.4, 'learning_rate': 0.5, 'max_depth': 1, 'n_estimators':
94, 'subsample': 0.6}
=>
{'loss': 0.02149463554007085, 'status': 'ok', 'train_loss':
0.009787639447378508, 'colsample_bynode': 0.4, 'learning_rate': 0.5,
'max_depth': 1, 'n_estimators': 94, 'subsample': 0.6}
{'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth': 7, 'n_estimators':
368, 'subsample': 0.7}
=>
{'loss': 0.019427227538402, 'status': 'ok', 'train_loss': 0.009752976862590267,
'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth': 7, 'n_estimators':
368, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 5, 'n_estimators':
179, 'subsample': 1}
=>
{'loss': 0.02184814425666957, 'status': 'ok', 'train_loss':
0.009810508154259268, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 5, 'n_estimators': 179, 'subsample': 1}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
107, 'subsample': 0.6}
=>
{'loss': 0.018458384003673407, 'status': 'ok', 'train_loss':
0.009534720344896873, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 7, 'n_estimators': 107, 'subsample': 0.6}
{'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 5, 'n_estimators': 384,
'subsample': 0.6}
=>
{'loss': 0.026475428759958192, 'status': 'ok', 'train_loss':
0.010508178557739398, 'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 5,
'n_estimators': 384, 'subsample': 0.6}
{'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth': 7, 'n_estimators':
166, 'subsample': 1}
=>
{'loss': 0.02375643349743516, 'status': 'ok', 'train_loss':
0.010279731360041275, 'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth':
7, 'n_estimators': 166, 'subsample': 1}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 1, 'n_estimators':

85, 'subsample': 0.4}
=>
{'loss': 0.021774449179950313, 'status': 'ok', 'train_loss':
0.010048078677850476, 'colsample_bynode': 0.6, 'learning_rate': 0.5,
'max_depth': 1, 'n_estimators': 85, 'subsample': 0.4}
{'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth': 7, 'n_estimators':
351, 'subsample': 0.5}
=>
{'loss': 0.02168470071614595, 'status': 'ok', 'train_loss':
0.010665176613495309, 'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth':
7, 'n_estimators': 351, 'subsample': 0.5}
{'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 1, 'n_estimators': 260,
'subsample': 1}
=>
{'loss': 0.02923966831275202, 'status': 'ok', 'train_loss':
0.014324814823812407, 'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 1,
'n_estimators': 260, 'subsample': 1}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
312, 'subsample': 0.6}
=>
{'loss': 0.020018561868550822, 'status': 'ok', 'train_loss':
0.009540143792862166, 'colsample_bynode': 0.6, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 312, 'subsample': 0.6}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
131, 'subsample': 0.4}
=>
{'loss': 0.020279786140288903, 'status': 'ok', 'train_loss':
0.011511180332414817, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 131, 'subsample': 0.4}
{'colsample_bynode': 0.5, 'learning_rate': 0.5, 'max_depth': 5, 'n_estimators':
60, 'subsample': 0.5}
=>
{'loss': 0.02024743104937957, 'status': 'ok', 'train_loss': 0.01004122165693674,
'colsample_bynode': 0.5, 'learning_rate': 0.5, 'max_depth': 5, 'n_estimators':
60, 'subsample': 0.5}
{'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth': 7, 'n_estimators':
124, 'subsample': 0.7}
=>
{'loss': 0.018907410272434436, 'status': 'ok', 'train_loss':
0.011322813193230174, 'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth':
7, 'n_estimators': 124, 'subsample': 0.7}
{'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth': 7, 'n_estimators':
331, 'subsample': 0.6}
=>
{'loss': 0.020682398530314776, 'status': 'ok', 'train_loss':
0.010295347879465185, 'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth':
7, 'n_estimators': 331, 'subsample': 0.6}
{'colsample_bynode': 0.4, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':

```
107, 'subsample': 0.7}
=>
{'loss': 0.018726917452867024, 'status': 'ok', 'train_loss':
0.00965618454240924, 'colsample_bynode': 0.4, 'learning_rate': 0.5, 'max_depth':
7, 'n_estimators': 107, 'subsample': 0.7}
{'colsample_bynode': 0.4, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
210, 'subsample': 0.7}
=>
{'loss': 0.01867758137582251, 'status': 'ok', 'train_loss': 0.00964587911121564,
'colsample_bynode': 0.4, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
210, 'subsample': 0.7}
{'colsample_bynode': 0.4, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
100, 'subsample': 0.7}
=>
{'loss': 0.018706401249036327, 'status': 'ok', 'train_loss':
0.009658105958093924, 'colsample_bynode': 0.4, 'learning_rate': 0.5,
'max_depth': 7, 'n_estimators': 100, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
150, 'subsample': 0.7}
=>
{'loss': 0.018391192836998062, 'status': 'ok', 'train_loss':
0.00912838209768504, 'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth':
7, 'n_estimators': 150, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
150, 'subsample': 0.7}
=>
{'loss': 0.018391192836998062, 'status': 'ok', 'train_loss':
0.00912838209768504, 'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth':
7, 'n_estimators': 150, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
150, 'subsample': 0.7}
=>
{'loss': 0.018391192836998062, 'status': 'ok', 'train_loss':
0.00912838209768504, 'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth':
7, 'n_estimators': 150, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
157, 'subsample': 0.7}
=>
{'loss': 0.018429803904164835, 'status': 'ok', 'train_loss':
0.009139900692144381, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 7, 'n_estimators': 157, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
200, 'subsample': 0.7}
=>
{'loss': 0.018371163231389362, 'status': 'ok', 'train_loss':
0.00912676799290517, 'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth':
7, 'n_estimators': 200, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
```

299, 'subsample': 0.7}
=>
{'loss': 0.018671509475577473, 'status': 'ok', 'train_loss':
0.009584000510205958, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 299, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
106, 'subsample': 0.7}
=>
{'loss': 0.01839900542325946, 'status': 'ok', 'train_loss':
0.009148730075024468, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 7, 'n_estimators': 106, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
56, 'subsample': 0.5}
=>
{'loss': 0.021068329168600883, 'status': 'ok', 'train_loss':
0.009964345074784513, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 56, 'subsample': 0.5}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
228, 'subsample': 0.4}
=>
{'loss': 0.019932977688736905, 'status': 'ok', 'train_loss':
0.009879139484826695, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 7, 'n_estimators': 228, 'subsample': 0.4}
{'colsample_bynode': 1, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
250, 'subsample': 0.7}
=>
{'loss': 0.01966170545436672, 'status': 'ok', 'train_loss':
0.009801018449731757, 'colsample_bynode': 1, 'learning_rate': 0.5, 'max_depth':
3, 'n_estimators': 250, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
231, 'subsample': 1}
=>
{'loss': 0.02109872168564039, 'status': 'ok', 'train_loss':
0.009715667950986029, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 7, 'n_estimators': 231, 'subsample': 1}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
273, 'subsample': 0.5}
=>
{'loss': 0.02186130276764457, 'status': 'ok', 'train_loss':
0.010086449629915107, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 7, 'n_estimators': 273, 'subsample': 0.5}
{'colsample_bynode': 1, 'learning_rate': 0.5, 'max_depth': 1, 'n_estimators':
218, 'subsample': 0.7}
=>
{'loss': 0.021369181792280072, 'status': 'ok', 'train_loss':
0.00948657526546517, 'colsample_bynode': 1, 'learning_rate': 0.5, 'max_depth':
1, 'n_estimators': 218, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 5, 'n_estimators':

```
200, 'subsample': 0.4}
=>
{'loss': 0.01984264718390866, 'status': 'ok', 'train_loss':
0.010318000005631553, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 5, 'n_estimators': 200, 'subsample': 0.4}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
175, 'subsample': 0.7}
=>
{'loss': 0.01828051540625518, 'status': 'ok', 'train_loss':
0.009454544003798904, 'colsample_bynode': 0.6, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 175, 'subsample': 0.7}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
42, 'subsample': 0.7}
=>
{'loss': 0.018278892297853042, 'status': 'ok', 'train_loss':
0.009557546018518022, 'colsample_bynode': 0.6, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 42, 'subsample': 0.7}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
42, 'subsample': 1}
=>
{'loss': 0.022183394020881413, 'status': 'ok', 'train_loss':
0.00984623395002492, 'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth':
3, 'n_estimators': 42, 'subsample': 1}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
175, 'subsample': 0.7}
=>
{'loss': 0.01828051540625518, 'status': 'ok', 'train_loss':
0.009454544003798904, 'colsample_bynode': 0.6, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 175, 'subsample': 0.7}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
398, 'subsample': 0.4}
=>
{'loss': 0.01910720949306562, 'status': 'ok', 'train_loss':
0.009773361510059163, 'colsample_bynode': 0.6, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 398, 'subsample': 0.4}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
97, 'subsample': 0.5}
=>
{'loss': 0.021290602900053707, 'status': 'ok', 'train_loss':
0.009680478849334586, 'colsample_bynode': 0.6, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 97, 'subsample': 0.5}
{'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth': 3, 'n_estimators':
247, 'subsample': 1}
=>
{'loss': 0.023248387238679655, 'status': 'ok', 'train_loss':
0.010688454880334503, 'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth':
3, 'n_estimators': 247, 'subsample': 1}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
```

```
334, 'subsample': 0.6}
=>
{'loss': 0.019935754314252943, 'status': 'ok', 'train_loss':
0.009524425427149546, 'colsample_bynode': 0.6, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 334, 'subsample': 0.6}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
112, 'subsample': 0.7}
=>
{'loss': 0.01829093147152248, 'status': 'ok', 'train_loss':
0.009465980169112702, 'colsample_bynode': 0.6, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 112, 'subsample': 0.7}
{'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth': 3, 'n_estimators':
359, 'subsample': 0.7}
=>
{'loss': 0.017919715350828213, 'status': 'ok', 'train_loss':
0.010523712028967356, 'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth':
3, 'n_estimators': 359, 'subsample': 0.7}
{'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth': 1, 'n_estimators':
45, 'subsample': 0.6}
=>
{'loss': 0.16551458300029137, 'status': 'ok', 'train_loss': 0.1558045257692576,
'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth': 1, 'n_estimators': 45,
'subsample': 0.6}
{'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth': 3, 'n_estimators':
115, 'subsample': 0.4}
=>
{'loss': 0.019725126725513572, 'status': 'ok', 'train_loss':
0.011027072182896423, 'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth':
3, 'n_estimators': 115, 'subsample': 0.4}
{'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth': 3, 'n_estimators':
165, 'subsample': 0.5}
=>
{'loss': 0.020131677723017445, 'status': 'ok', 'train_loss':
0.010827717713793853, 'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth':
3, 'n_estimators': 165, 'subsample': 0.5}
100%|                    | 50/50 [23:45<00:00,
28.50s/trial, best loss: 0.017919715350828213]
        loss status  train_loss  colsample_bynode  learning_rate  max_depth  \
46  0.017920     ok    0.010524               0.6            1.0          3
38  0.018279     ok    0.009558               0.6            0.5          3
40  0.018281     ok    0.009455               0.6            0.5          3
37  0.018281     ok    0.009455               0.6            0.5          3
45  0.018291     ok    0.009466               0.6            0.5          3
27  0.018371     ok    0.009127               0.7            0.5          7
24  0.018391     ok    0.009128               0.7            0.5          7
23  0.018391     ok    0.009128               0.7            0.5          7
25  0.018391     ok    0.009128               0.7            0.5          7
29  0.018399     ok    0.009149               0.7            0.5          7
```

26	0.018430	ok	0.009140	0.7	0.5	7
9	0.018458	ok	0.009535	0.7	0.5	7
28	0.018672	ok	0.009584	0.7	0.5	3
21	0.018678	ok	0.009646	0.4	0.5	7
22	0.018706	ok	0.009658	0.4	0.5	7
20	0.018727	ok	0.009656	0.4	0.5	7
18	0.018907	ok	0.011323	0.4	1.0	7
41	0.019107	ok	0.009773	0.6	0.5	3
7	0.019427	ok	0.009753	0.5	1.0	7
32	0.019662	ok	0.009801	1.0	0.5	3
48	0.019725	ok	0.011027	0.5	1.0	3
36	0.019843	ok	0.010318	0.7	0.5	5
31	0.019933	ok	0.009879	0.7	0.5	7
44	0.019936	ok	0.009524	0.6	0.5	3
15	0.020019	ok	0.009540	0.6	0.5	3
49	0.020132	ok	0.010828	0.6	1.0	3
17	0.020247	ok	0.010041	0.5	0.5	5
16	0.020280	ok	0.011511	0.7	1.0	5
19	0.020682	ok	0.010295	0.5	1.0	7
1	0.020877	ok	0.009884	0.7	0.5	3
30	0.021068	ok	0.009964	0.7	0.5	3
33	0.021099	ok	0.009716	0.7	0.5	7
42	0.021291	ok	0.009680	0.6	0.5	3
35	0.021369	ok	0.009487	1.0	0.5	1
6	0.021495	ok	0.009788	0.4	0.5	1
13	0.021685	ok	0.010665	0.6	1.0	7
12	0.021774	ok	0.010048	0.6	0.5	1
8	0.021848	ok	0.009811	0.7	0.5	5
34	0.021861	ok	0.010086	0.7	0.5	7
5	0.022144	ok	0.010157	0.4	1.0	1
39	0.022183	ok	0.009846	0.6	0.5	3
2	0.023026	ok	0.010747	0.7	1.0	3
43	0.023248	ok	0.010688	0.6	1.0	3
0	0.023426	ok	0.011407	0.7	1.0	5
11	0.023756	ok	0.010280	0.5	1.0	7
10	0.026475	ok	0.010508	1.0	1.0	5
3	0.026573	ok	0.010456	1.0	1.0	5
14	0.029240	ok	0.014325	1.0	1.0	1
4	0.066989	ok	0.053036	0.6	1.0	1
47	0.165515	ok	0.155805	0.6	1.0	1

	n_estimators	subsample
46	359	0.7
38	42	0.7
40	175	0.7
37	175	0.7
45	112	0.7
27	200	0.7

```
24              150          0.7
23              150          0.7
25              150          0.7
29              106          0.7
26              157          0.7
9               107          0.6
28              299          0.7
21              210          0.7
22              100          0.7
20              107          0.7
18              124          0.7
41              398          0.4
7               368          0.7
32              250          0.7
48              115          0.4
36              200          0.4
31              228          0.4
44              334          0.6
15              312          0.6
49              165          0.5
17               60          0.5
16              131          0.4
19              331          0.6
1               396          0.5
30               56          0.5
33              231          1.0
42               97          0.5
35              218          0.7
6                94          0.6
13              351          0.5
12               85          0.4
8               179          1.0
34              273          0.5
5               193          0.6
39               42          1.0
2               378          0.4
43              247          1.0
0               290          0.5
11              166          1.0
10              384          0.6
3               104          0.6
14              260          1.0
4               260          0.7
47               45          0.6
done
```

```
[1]: import pandas as pd
     pd.read_csv("feature_set2_50_trials.csv")
```

[1]:

	Unnamed: 0	loss	status	train_loss	colsample_bynode	learning_rate	\
0	46	0.017920	ok	0.010524	0.6	1.0	
1	38	0.018279	ok	0.009558	0.6	0.5	
2	40	0.018281	ok	0.009455	0.6	0.5	
3	37	0.018281	ok	0.009455	0.6	0.5	
4	45	0.018291	ok	0.009466	0.6	0.5	
5	27	0.018371	ok	0.009127	0.7	0.5	
6	24	0.018391	ok	0.009128	0.7	0.5	
7	23	0.018391	ok	0.009128	0.7	0.5	
8	25	0.018391	ok	0.009128	0.7	0.5	
9	29	0.018399	ok	0.009149	0.7	0.5	
10	26	0.018430	ok	0.009140	0.7	0.5	
11	9	0.018458	ok	0.009535	0.7	0.5	
12	28	0.018672	ok	0.009584	0.7	0.5	
13	21	0.018678	ok	0.009646	0.4	0.5	
14	22	0.018706	ok	0.009658	0.4	0.5	
15	20	0.018727	ok	0.009656	0.4	0.5	
16	18	0.018907	ok	0.011323	0.4	1.0	
17	41	0.019107	ok	0.009773	0.6	0.5	
18	7	0.019427	ok	0.009753	0.5	1.0	
19	32	0.019662	ok	0.009801	1.0	0.5	
20	48	0.019725	ok	0.011027	0.5	1.0	
21	36	0.019843	ok	0.010318	0.7	0.5	
22	31	0.019933	ok	0.009879	0.7	0.5	
23	44	0.019936	ok	0.009524	0.6	0.5	
24	15	0.020019	ok	0.009540	0.6	0.5	
25	49	0.020132	ok	0.010828	0.6	1.0	
26	17	0.020247	ok	0.010041	0.5	0.5	
27	16	0.020280	ok	0.011511	0.7	1.0	
28	19	0.020682	ok	0.010295	0.5	1.0	
29	1	0.020877	ok	0.009884	0.7	0.5	
30	30	0.021068	ok	0.009964	0.7	0.5	
31	33	0.021099	ok	0.009716	0.7	0.5	
32	42	0.021291	ok	0.009680	0.6	0.5	
33	35	0.021369	ok	0.009487	1.0	0.5	
34	6	0.021495	ok	0.009788	0.4	0.5	
35	13	0.021685	ok	0.010665	0.6	1.0	
36	12	0.021774	ok	0.010048	0.6	0.5	
37	8	0.021848	ok	0.009811	0.7	0.5	
38	34	0.021861	ok	0.010086	0.7	0.5	
39	5	0.022144	ok	0.010157	0.4	1.0	
40	39	0.022183	ok	0.009846	0.6	0.5	
41	2	0.023026	ok	0.010747	0.7	1.0	
42	43	0.023248	ok	0.010688	0.6	1.0	

43	0	0.023426	ok	0.011407	0.7	1.0
44	11	0.023756	ok	0.010280	0.5	1.0
45	10	0.026475	ok	0.010508	1.0	1.0
46	3	0.026573	ok	0.010456	1.0	1.0
47	14	0.029240	ok	0.014325	1.0	1.0
48	4	0.066989	ok	0.053036	0.6	1.0
49	47	0.165515	ok	0.155805	0.6	1.0

	max_depth	n_estimators	subsample
0	3	359	0.7
1	3	42	0.7
2	3	175	0.7
3	3	175	0.7
4	3	112	0.7
5	7	200	0.7
6	7	150	0.7
7	7	150	0.7
8	7	150	0.7
9	7	106	0.7
10	7	157	0.7
11	7	107	0.6
12	3	299	0.7
13	7	210	0.7
14	7	100	0.7
15	7	107	0.7
16	7	124	0.7
17	3	398	0.4
18	7	368	0.7
19	3	250	0.7
20	3	115	0.4
21	5	200	0.4
22	7	228	0.4
23	3	334	0.6
24	3	312	0.6
25	3	165	0.5
26	5	60	0.5
27	5	131	0.4
28	7	331	0.6
29	3	396	0.5
30	3	56	0.5
31	7	231	1.0
32	3	97	0.5
33	1	218	0.7
34	1	94	0.6
35	7	351	0.5
36	1	85	0.4
37	5	179	1.0

38	7	273	0.5
39	1	193	0.6
40	3	42	1.0
41	3	378	0.4
42	3	247	1.0
43	5	290	0.5
44	7	166	1.0
45	5	384	0.6
46	5	104	0.6
47	1	260	1.0
48	1	260	0.7
49	1	45	0.6