3_MSMWD_cross_validataion_and_training_set1

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_set1.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])
```

```
Train:cv:test :: 6520 : 2174 : 2174
```

[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_set1_50_trials.csv")
print(tuned)
print('done')
```

```
Tuning..
{'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 5, 'n_estimators': 143,
'subsample': 0.5}
=>
{'loss': 0.02253795597828581, 'status': 'ok', 'train_loss':
0.011957510006990631, 'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 5,
'n_estimators': 143, 'subsample': 0.5}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
327, 'subsample': 0.4}
=>
{'loss': 0.02087359055641676, 'status': 'ok', 'train_loss':
0.010881264450963456, 'colsample_bynode': 0.6, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 327, 'subsample': 0.4}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 5, 'n_estimators':
117, 'subsample': 0.6}
=>
{'loss': 0.020654722208206876, 'status': 'ok', 'train_loss':
0.010370779039529373, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 5, 'n_estimators': 117, 'subsample': 0.6}
{'colsample_bynode': 1, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
385, 'subsample': 0.5}
=>
{'loss': 0.02073835160303678, 'status': 'ok', 'train_loss':
0.010634715209482632, 'colsample_bynode': 1, 'learning_rate': 0.5, 'max_depth':
3, 'n_estimators': 385, 'subsample': 0.5}
{'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth': 7, 'n_estimators':
111, 'subsample': 0.4}
=>
{'loss': 0.021544133234389584, 'status': 'ok', 'train_loss':
```

```
0.012787377289525196, 'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth':
7, 'n_estimators': 111, 'subsample': 0.4}
{'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
217, 'subsample': 0.6}
=>
{'loss': 0.02363812173941839, 'status': 'ok', 'train_loss':
0.012314685022623824, 'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 217, 'subsample': 0.6}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
141, 'subsample': 1}
=>
{'loss': 0.020710199838869695, 'status': 'ok', 'train_loss':
0.010424487244220848, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 141, 'subsample': 1}
{'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth': 1, 'n_estimators':
287, 'subsample': 0.7}
=>
{'loss': 0.33836525197780964, 'status': 'ok', 'train_loss': 0.31526961256551833,
'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth': 1, 'n_estimators':
287, 'subsample': 0.7}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
146, 'subsample': 0.6}
=>
{'loss': 0.020285067837089277, 'status': 'ok', 'train_loss':
0.010686241180864474, 'colsample_bynode': 0.6, 'learning_rate': 0.5,
'max_depth': 5, 'n_estimators': 223, 'subsample': 0.5}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 1, 'n_estimators':
374, 'subsample': 0.7}
=>
{'loss': 0.023287849073431777, 'status': 'ok', 'train_loss':
0.010382746117335945, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 1, 'n_estimators': 374, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
364, 'subsample': 1}
=>
{'loss': 0.01828638982036347, 'status': 'ok', 'train_loss':
0.010962891430826996, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 364, 'subsample': 1}
{'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
168, 'subsample': 0.6}
=>
{'loss': 0.023661322026297303, 'status': 'ok', 'train_loss':
0.012308895962762302, 'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 168, 'subsample': 0.6}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 1, 'n_estimators':
263, 'subsample': 0.5}
=>
{'loss': 0.022476168605874806, 'status': 'ok', 'train_loss':
```

```
0.010769226369704798, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 1, 'n_estimators': 263, 'subsample': 0.5}
{'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 3, 'n_estimators': 77,
'subsample': 0.4}
=>
{'loss': 0.023573635575668182, 'status': 'ok', 'train_loss':
0.011759942331898011, 'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 3,
'n_estimators': 77, 'subsample': 0.4}
{'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 5, 'n_estimators': 225,
'subsample': 0.5}
=>
{'loss': 0.022626404814395432, 'status': 'ok', 'train_loss':
0.012000583652407512, 'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 5,
'n_estimators': 225, 'subsample': 0.5}
{'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth': 7, 'n_estimators':
293, 'subsample': 0.6}
=>
{'loss': 0.023719186794805218, 'status': 'ok', 'train_loss':
0.012619831783114977, 'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth':
7, 'n_estimators': 293, 'subsample': 0.6}
{'colsample_bynode': 0.5, 'learning_rate': 0.5, 'max_depth': 3, 'n_estimators':
213, 'subsample': 1}
=>
{'loss': 0.022238819314013718, 'status': 'ok', 'train_loss':
0.010030772969645644, 'colsample_bynode': 0.5, 'learning_rate': 0.5,
'max_depth': 3, 'n_estimators': 213, 'subsample': 1}
{'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth': 1, 'n_estimators':
373, 'subsample': 0.6}
=>
{'loss': 0.04621633431558832, 'status': 'ok', 'train_loss': 0.03541373692402954,
'colsample_bynode': 0.6, 'learning_rate': 1, 'max_depth': 1, 'n_estimators':
373, 'subsample': 0.6}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 5, 'n_estimators':
195, 'subsample': 0.4}
=>
{'loss': 0.020613337482772436, 'status': 'ok', 'train_loss':
0.011339821573903489, 'colsample_bynode': 0.6, 'learning_rate': 0.5,
'max_depth': 5, 'n_estimators': 195, 'subsample': 0.4}
{'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth': 7, 'n_estimators':
296, 'subsample': 1}
=>
{'loss': 0.02256677929132009, 'status': 'ok', 'train_loss': 0.01131595731386107,
'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth': 7, 'n_estimators':
296, 'subsample': 1}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
146, 'subsample': 1}
=>
{'loss': 0.02101295974083065, 'status': 'ok', 'train_loss':
```

```
0.010363411404799726, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 7, 'n_estimators': 146, 'subsample': 1}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
177, 'subsample': 1}
=>
{'loss': 0.021012794578706275, 'status': 'ok', 'train_loss':
0.010363118232219191, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 7, 'n_estimators': 177, 'subsample': 1}
{'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth': 7, 'n_estimators':
255, 'subsample': 1}
=>
{'loss': 0.022565738756071454, 'status': 'ok', 'train_loss':
0.011314919434454139, 'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth':
7, 'n_estimators': 255, 'subsample': 1}
{'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth': 7, 'n_estimators':
364, 'subsample': 0.7}
=>
{'loss': 0.020315600164210305, 'status': 'ok', 'train_loss':
0.01105357360706564, 'colsample_bynode': 0.6, 'learning_rate': 0.5, 'max_depth':
7, 'n_estimators': 364, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 7, 'n_estimators':
395, 'subsample': 0.6}
=>
{'loss': 0.023818972680273342, 'status': 'ok', 'train_loss':
0.01153646372770917, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
7, 'n_estimators': 395, 'subsample': 0.6}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
388, 'subsample': 1}
=>
{'loss': 0.01828682945804985, 'status': 'ok', 'train_loss': 0.01096335763337251,
'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
388, 'subsample': 1}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
332, 'subsample': 1}
=>
{'loss': 0.018286772930264004, 'status': 'ok', 'train_loss':
0.010962979138450017, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 332, 'subsample': 1}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
368, 'subsample': 1}
=>
{'loss': 0.01828653118209724, 'status': 'ok', 'train_loss':
0.010963081289536724, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 368, 'subsample': 1}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
318, 'subsample': 1}
=>
{'loss': 0.01828650319458743, 'status': 'ok', 'train_loss':
```

```
0.010963042581933834, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 318, 'subsample': 1}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
127, 'subsample': 1}
=>
{'loss': 0.018285903325850058, 'status': 'ok', 'train_loss':
0.010962410038533104, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 127, 'subsample': 1}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
302, 'subsample': 1}
=>
{'loss': 0.018287054364135923, 'status': 'ok', 'train_loss':
0.010963479364407213, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 302, 'subsample': 1}
{'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 5, 'n_estimators': 304,
'subsample': 1}
=>
{'loss': 0.023808949448387354, 'status': 'ok', 'train_loss':
0.011252303864380083, 'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 5,
'n_estimators': 304, 'subsample': 1}
{'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
291, 'subsample': 0.7}
=>
{'loss': 0.019747565641880512, 'status': 'ok', 'train_loss':
0.01166820681424185, 'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 291, 'subsample': 0.7}
{'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
381, 'subsample': 0.4}
=>
{'loss': 0.024411565698279836, 'status': 'ok', 'train_loss':
0.012963083388168584, 'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 381, 'subsample': 0.4}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 3, 'n_estimators':
196, 'subsample': 1}
=>
{'loss': 0.020668024919943354, 'status': 'ok', 'train_loss':
0.01082404091160321, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
3, 'n_estimators': 196, 'subsample': 1}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
242, 'subsample': 0.5}
=>
{'loss': 0.021345811429222906, 'status': 'ok', 'train_loss':
0.011225829556888027, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 242, 'subsample': 0.5}
{'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 1, 'n_estimators': 350,
'subsample': 1}
=>
{'loss': 0.22980558037052057, 'status': 'ok', 'train_loss': 0.21895007688436174,
```

```
'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 1, 'n_estimators': 350,
'subsample': 1}
{'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
104, 'subsample': 0.4}
=>
{'loss': 0.02040330677317309, 'status': 'ok', 'train_loss':
0.011767552070065603, 'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 104, 'subsample': 0.4}
{'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth': 3, 'n_estimators':
392, 'subsample': 0.7}
=>
{'loss': 0.022921753167528605, 'status': 'ok', 'train_loss':
0.012372889716534963, 'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth':
3, 'n_estimators': 392, 'subsample': 0.7}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
367, 'subsample': 1}
=>
{'loss': 0.018287141877980893, 'status': 'ok', 'train_loss':
0.01096362293842899, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 367, 'subsample': 1}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 1, 'n_estimators':
316, 'subsample': 0.5}
=>
{'loss': 0.032284881643054666, 'status': 'ok', 'train_loss':
0.02036321885663207, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
1, 'n_estimators': 316, 'subsample': 0.5}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
183, 'subsample': 1}
=>
{'loss': 0.01828658542776287, 'status': 'ok', 'train_loss':
0.010962913835020176, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 183, 'subsample': 1}
{'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 3, 'n_estimators': 365,
'subsample': 0.7}
=>
{'loss': 0.022402625566500002, 'status': 'ok', 'train_loss':
0.01173781890235131, 'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 3,
'n_estimators': 365, 'subsample': 0.7}
{'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
326, 'subsample': 0.4}
=>
{'loss': 0.02473264450442829, 'status': 'ok', 'train_loss':
0.012998809304376583, 'colsample_bynode': 0.5, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 326, 'subsample': 0.4}
{'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth': 1, 'n_estimators':
280, 'subsample': 0.5}
=>
{'loss': 0.01876390444252746, 'status': 'ok', 'train_loss':
```

```
0.010998090442452575, 'colsample_bynode': 0.4, 'learning_rate': 1, 'max_depth':
1, 'n_estimators': 280, 'subsample': 0.5}
{'colsample_bynode': 0.7, 'learning_rate': 0.5, 'max_depth': 5, 'n_estimators':
253, 'subsample': 0.6}
=>
{'loss': 0.02062946020435429, 'status': 'ok', 'train_loss':
0.010350400496343226, 'colsample_bynode': 0.7, 'learning_rate': 0.5,
'max_depth': 5, 'n_estimators': 253, 'subsample': 0.6}
{'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth': 5, 'n_estimators':
288, 'subsample': 1}
=>
{'loss': 0.01828602442384233, 'status': 'ok', 'train_loss':
0.010962506222193084, 'colsample_bynode': 0.7, 'learning_rate': 1, 'max_depth':
5, 'n_estimators': 288, 'subsample': 1}
{'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 3, 'n_estimators': 39,
'subsample': 1}
=>
{'loss': 0.02556493171580036, 'status': 'ok', 'train_loss':
0.011074411649318269, 'colsample_bynode': 1, 'learning_rate': 1, 'max_depth': 3,
'n_estimators': 39, 'subsample': 1}
{'colsample_bynode': 0.5, 'learning_rate': 0.5, 'max_depth': 5, 'n_estimators':
383, 'subsample': 0.7}
=>
{'loss': 0.020786294520645187, 'status': 'ok', 'train_loss':
0.01064349351336331, 'colsample_bynode': 0.5, 'learning_rate': 0.5, 'max_depth':
5, 'n_estimators': 383, 'subsample': 0.7}
100%|                    | 50/50 [2:36:50<00:00,
188.20s/trial, best loss: 0.018285903325850058]
        loss status  train_loss  colsample_bynode  learning_rate  max_depth  \
30  0.018286     ok    0.010962               0.7            1.0          5
47  0.018286     ok    0.010963               0.7            1.0          5
11  0.018286     ok    0.010963               0.7            1.0          5
29  0.018287     ok    0.010963               0.7            1.0          5
28  0.018287     ok    0.010963               0.7            1.0          5
42  0.018287     ok    0.010963               0.7            1.0          5
27  0.018287     ok    0.010963               0.7            1.0          5
26  0.018287     ok    0.010963               0.7            1.0          5
31  0.018287     ok    0.010963               0.7            1.0          5
40  0.018287     ok    0.010964               0.7            1.0          5
45  0.018764     ok    0.010998               0.4            1.0          1
33  0.019748     ok    0.011668               0.4            1.0          5
8   0.020115     ok    0.010520               0.6            0.5          7
9   0.020285     ok    0.010686               0.6            0.5          5
24  0.020316     ok    0.011054               0.6            0.5          7
38  0.020403     ok    0.011768               0.4            1.0          5
19  0.020613     ok    0.011340               0.6            0.5          5
46  0.020629     ok    0.010350               0.7            0.5          5
2   0.020655     ok    0.010371               0.7            0.5          5
```

35	0.020668	ok	0.010824	0.7	1.0	3
6	0.020710	ok	0.010424	0.7	0.5	3
3	0.020738	ok	0.010635	1.0	0.5	3
49	0.020786	ok	0.010643	0.5	0.5	5
1	0.020874	ok	0.010881	0.6	0.5	3
22	0.021013	ok	0.010363	0.7	0.5	7
21	0.021013	ok	0.010363	0.7	0.5	7
36	0.021346	ok	0.011226	0.7	1.0	5
4	0.021544	ok	0.012787	0.4	1.0	7
17	0.022239	ok	0.010031	0.5	0.5	3
43	0.022403	ok	0.011738	1.0	1.0	3
13	0.022476	ok	0.010769	0.7	0.5	1
0	0.022538	ok	0.011958	1.0	1.0	5
23	0.022566	ok	0.011315	0.4	1.0	7
20	0.022567	ok	0.011316	0.4	1.0	7
15	0.022626	ok	0.012001	1.0	1.0	5
39	0.022922	ok	0.012373	0.5	1.0	3
10	0.023288	ok	0.010383	0.7	0.5	1
14	0.023574	ok	0.011760	1.0	1.0	3
5	0.023638	ok	0.012315	0.5	1.0	5
12	0.023661	ok	0.012309	0.5	1.0	5
16	0.023719	ok	0.012620	0.6	1.0	7
32	0.023809	ok	0.011252	1.0	1.0	5
25	0.023819	ok	0.011536	0.7	1.0	7
34	0.024412	ok	0.012963	0.5	1.0	5
44	0.024733	ok	0.012999	0.5	1.0	5
48	0.025565	ok	0.011074	1.0	1.0	3
41	0.032285	ok	0.020363	0.7	1.0	1
18	0.046216	ok	0.035414	0.6	1.0	1
37	0.229806	ok	0.218950	1.0	1.0	1
7	0.338365	ok	0.315270	0.6	1.0	1

	n_estimators	subsample
30	127	1.0
47	288	1.0
11	364	1.0
29	318	1.0
28	368	1.0
42	183	1.0
27	332	1.0
26	388	1.0
31	302	1.0
40	367	1.0
45	280	0.5
33	291	0.7
8	146	0.6
9	223	0.5
24	364	0.7

```
38              104         0.4
19              195         0.4
46              253         0.6
2               117         0.6
35              196         1.0
6               141         1.0
3               385         0.5
49              383         0.7
1               327         0.4
22              177         1.0
21              146         1.0
36              242         0.5
4               111         0.4
17              213         1.0
43              365         0.7
13              263         0.5
0               143         0.5
23              255         1.0
20              296         1.0
15              225         0.5
39              392         0.7
10              374         0.7
14               77         0.4
5               217         0.6
12              168         0.6
16              293         0.6
32              304         1.0
25              395         0.6
34              381         0.4
44              326         0.4
48               39         1.0
41              316         0.5
18              373         0.6
37              350         1.0
7               287         0.7
done
```

[1]:
```python
import pandas as pd
pd.read_csv("feature_set1_50_trials.csv")
```

[1]:
```
    Unnamed: 0      loss status  train_loss  colsample_bynode  learning_rate  \
0           30  0.018286     ok    0.010962               0.7            1.0
1           47  0.018286     ok    0.010963               0.7            1.0
2           11  0.018286     ok    0.010963               0.7            1.0
3           29  0.018287     ok    0.010963               0.7            1.0
4           28  0.018287     ok    0.010963               0.7            1.0
5           42  0.018287     ok    0.010963               0.7            1.0
```

6	27	0.018287	ok	0.010963	0.7	1.0
7	26	0.018287	ok	0.010963	0.7	1.0
8	31	0.018287	ok	0.010963	0.7	1.0
9	40	0.018287	ok	0.010964	0.7	1.0
10	45	0.018764	ok	0.010998	0.4	1.0
11	33	0.019748	ok	0.011668	0.4	1.0
12	8	0.020115	ok	0.010520	0.6	0.5
13	9	0.020285	ok	0.010686	0.6	0.5
14	24	0.020316	ok	0.011054	0.6	0.5
15	38	0.020403	ok	0.011768	0.4	1.0
16	19	0.020613	ok	0.011340	0.6	0.5
17	46	0.020629	ok	0.010350	0.7	0.5
18	2	0.020655	ok	0.010371	0.7	0.5
19	35	0.020668	ok	0.010824	0.7	1.0
20	6	0.020710	ok	0.010424	0.7	0.5
21	3	0.020738	ok	0.010635	1.0	0.5
22	49	0.020786	ok	0.010643	0.5	0.5
23	1	0.020874	ok	0.010881	0.6	0.5
24	22	0.021013	ok	0.010363	0.7	0.5
25	21	0.021013	ok	0.010363	0.7	0.5
26	36	0.021346	ok	0.011226	0.7	1.0
27	4	0.021544	ok	0.012787	0.4	1.0
28	17	0.022239	ok	0.010031	0.5	0.5
29	43	0.022403	ok	0.011738	1.0	1.0
30	13	0.022476	ok	0.010769	0.7	0.5
31	0	0.022538	ok	0.011958	1.0	1.0
32	23	0.022566	ok	0.011315	0.4	1.0
33	20	0.022567	ok	0.011316	0.4	1.0
34	15	0.022626	ok	0.012001	1.0	1.0
35	39	0.022922	ok	0.012373	0.5	1.0
36	10	0.023288	ok	0.010383	0.7	0.5
37	14	0.023574	ok	0.011760	1.0	1.0
38	5	0.023638	ok	0.012315	0.5	1.0
39	12	0.023661	ok	0.012309	0.5	1.0
40	16	0.023719	ok	0.012620	0.6	1.0
41	32	0.023809	ok	0.011252	1.0	1.0
42	25	0.023819	ok	0.011536	0.7	1.0
43	34	0.024412	ok	0.012963	0.5	1.0
44	44	0.024733	ok	0.012999	0.5	1.0
45	48	0.025565	ok	0.011074	1.0	1.0
46	41	0.032285	ok	0.020363	0.7	1.0
47	18	0.046216	ok	0.035414	0.6	1.0
48	37	0.229806	ok	0.218950	1.0	1.0
49	7	0.338365	ok	0.315270	0.6	1.0

	max_depth	n_estimators	subsample
0	5	127	1.0

1	5	288	1.0
2	5	364	1.0
3	5	318	1.0
4	5	368	1.0
5	5	183	1.0
6	5	332	1.0
7	5	388	1.0
8	5	302	1.0
9	5	367	1.0
10	1	280	0.5
11	5	291	0.7
12	7	146	0.6
13	5	223	0.5
14	7	364	0.7
15	5	104	0.4
16	5	195	0.4
17	5	253	0.6
18	5	117	0.6
19	3	196	1.0
20	3	141	1.0
21	3	385	0.5
22	5	383	0.7
23	3	327	0.4
24	7	177	1.0
25	7	146	1.0
26	5	242	0.5
27	7	111	0.4
28	3	213	1.0
29	3	365	0.7
30	1	263	0.5
31	5	143	0.5
32	7	255	1.0
33	7	296	1.0
34	5	225	0.5
35	3	392	0.7
36	1	374	0.7
37	3	77	0.4
38	5	217	0.6
39	5	168	0.6
40	7	293	0.6
41	5	304	1.0
42	7	395	0.6
43	5	381	0.4
44	5	326	0.4
45	3	39	1.0
46	1	316	0.5
47	1	373	0.6

48	1	350	1.0
49	1	287	0.7