DSO530-classification-CV

December 4, 2021

0.1 Pre-processing

[1]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from xgboost.sklearn import XGBClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from xgboost.sklearn import XGBClassifier
```

[51]:
```
df = pd.read_csv('option_train.csv')
```

[3]:
```
df.head()
```

[3]:

	Value	S	K	tau	r	BS
0	21.670404	431.623898	420	0.341270	0.03013	Under
1	0.125000	427.015526	465	0.166667	0.03126	Over
2	20.691244	427.762336	415	0.265873	0.03116	Under
3	1.035002	451.711658	460	0.063492	0.02972	Over
4	39.553020	446.718974	410	0.166667	0.02962	Under

[53]:
```
dic = {'Under':0, 'Over':1}
```

[54]:
```
a = df['BS'].map(lambda x:dic[x])
df['binary_label'] = a
```

[55]:
```
df.head(10)
```

[55]:

	Value	S	K	tau	r	BS	binary_label
0	21.670404	431.623898	420	0.341270	0.03013	Under	0
1	0.125000	427.015526	465	0.166667	0.03126	Over	1
2	20.691244	427.762336	415	0.265873	0.03116	Under	0
3	1.035002	451.711658	460	0.063492	0.02972	Over	1
4	39.553020	446.718974	410	0.166667	0.02962	Under	0
5	2.505002	436.958530	460	0.333333	0.03023	Over	1
6	4.315000	427.015526	435	0.166667	0.03126	Over	1
7	0.345002	428.996368	455	0.154762	0.03116	Over	1

```
8  27.297423  444.186127  420  0.150794  0.02993  Under             0
9   0.190000  429.314292  460  0.150794  0.03085   Over             1
```

[7]:
```
X = df[['S','K','tau','r']].values
y = df['binary_label']
```

[8]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.25,
                                                    random_state = 0,
                                                    stratify = y)
```

[9]:
```
lg = LogisticRegression()
rf = RandomForestClassifier()
```

[10]:
```
# lr on original data
lg.fit(X_train, y_train)
lg.score(X_test, y_test)
```

[10]:
```
0.9095238095238095
```

[20]:
```
# lr on standardized data
from sklearn.preprocessing import StandardScaler
stdsc = StandardScaler()
X_train_std = stdsc.fit_transform(X_train)
X_test_std = stdsc.transform(X_test)

lg.fit(X_train_std, y_train)
lg.score(X_test_std, y_test)
```

[20]:
```
0.9238095238095239
```

[46]:
```
# rf
rf.fit(X_train, y_train)
rf.score(X_test, y_test)
```

[46]:
```
0.9380952380952381
```

0.2 Parameter

0.2.1 Train-Test

[13]:
```
reg = ['l2', 'l2', 'l2', 'l1']
C = [1,0.2,5,1,1,0.2,0.6]
sol = ['lbfgs', 'lbfgs', 'lbfgs', 'liblinear']
```

[14]:
```
d = []
```

```
for i in range(len(reg)):
    lg = LogisticRegression(penalty = reg[i] ,solver=sol[i], C = C[i],␣
 ↪class_weight='balanced', max_iter=200 )
    lg.fit(X_train_std, y_train)
    scoring = lg.score(X_test_std, y_test)
    d.append([reg[i], C[i], sol[i], scoring])
d
```

[14]:
```
[['l2', 1, 'lbfgs', 0.9238095238095239],
 ['l2', 0.2, 'lbfgs', 0.9214285714285714],
 ['l2', 5, 'lbfgs', 0.9238095238095239],
 ['l1', 1, 'liblinear', 0.9238095238095239]]
```

[15]:
```
num = [50, 75, 100, 200, 400, 100, 200]
criterion=['entropy','entropy','entropy','entropy','entropy', 'gini', 'gini']
```

[16]:
```
rf = []
for i in range(len(num)):
    classifier_RF = RandomForestClassifier(n_estimators = num[i],␣
 ↪criterion=criterion[i],
                                           random_state=1,
                                           verbose = 1,
                                           oob_score=True)
    classifier_RF.fit(X_train, y_train)
    scoring = classifier_RF.score(X_test, y_test)
    rf.append([num[i], criterion[i], scoring])

rf
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.5s finished
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
```

[16]:
```
[[50, 'entropy', 0.9333333333333333],
 [75, 'entropy', 0.9357142857142857],
 [100, 'entropy', 0.9357142857142857],
 [200, 'entropy', 0.9380952380952381],
 [400, 'entropy', 0.9333333333333333],
 [100, 'gini', 0.9404761904761905],
 [200, 'gini', 0.9380952380952381]]
```

[17]:
```
num = [100, 400, 400, 100, 100, 100]
eta = [0.1,0.1,0.01,0.1, 0.05, 0.1]
# depth = [6, 10, 10, 10, 10, 12]
```

[19]:
```
f = []
for i in range(len(num)):
    xgb_model = XGBClassifier(objective = 'binary:logistic',
                              learning_rate = eta[i],
                              n_estimators = num[i],
                              # max_depth = depth[i],
                              subsample = 0.8,
                              colsample_bytree = 0.8,
                              # reg_lambda = l2_reg[i] ,
                              random_state = 1,
                              use_label_encoder=False
                              )

    xgb_model.fit(X_train, y_train, eval_set=[(X_test, y_test)],␣
 ↪eval_metric='error', early_stopping_rounds=100)
    scoring = xgb_model.score(X_test, y_test)
    f.append([eta[i], num[i], scoring])
f
```

```
[0]     validation_0-error:0.08571
[1]     validation_0-error:0.07619
[2]     validation_0-error:0.07619
[3]     validation_0-error:0.06905
[4]     validation_0-error:0.06191
[5]     validation_0-error:0.06191
```

```
[6]     validation_0-error:0.06429
[7]     validation_0-error:0.06191
[8]     validation_0-error:0.05714
[9]     validation_0-error:0.06191
[10]    validation_0-error:0.05952
[11]    validation_0-error:0.06191
[12]    validation_0-error:0.06667
[13]    validation_0-error:0.06905
[14]    validation_0-error:0.07143
[15]    validation_0-error:0.06667
[16]    validation_0-error:0.06667
[17]    validation_0-error:0.06429
[18]    validation_0-error:0.06667
[19]    validation_0-error:0.06191
[20]    validation_0-error:0.06429
[21]    validation_0-error:0.06191
[22]    validation_0-error:0.06429
[23]    validation_0-error:0.06667
[24]    validation_0-error:0.06905
[25]    validation_0-error:0.06667
[26]    validation_0-error:0.06429
[27]    validation_0-error:0.06429
[28]    validation_0-error:0.06905
[29]    validation_0-error:0.07143
[30]    validation_0-error:0.06905
[31]    validation_0-error:0.06905
[32]    validation_0-error:0.06905
[33]    validation_0-error:0.06905
[34]    validation_0-error:0.06429
[35]    validation_0-error:0.06905
[36]    validation_0-error:0.06667
[37]    validation_0-error:0.06191
[38]    validation_0-error:0.06191
[39]    validation_0-error:0.06191
[40]    validation_0-error:0.05714
[41]    validation_0-error:0.05714
[42]    validation_0-error:0.05952
[43]    validation_0-error:0.05952
[44]    validation_0-error:0.05952
[45]    validation_0-error:0.05952
[46]    validation_0-error:0.05714
[47]    validation_0-error:0.06191
[48]    validation_0-error:0.06191
[49]    validation_0-error:0.06191
[50]    validation_0-error:0.06191
[51]    validation_0-error:0.06191
[52]    validation_0-error:0.06191
[53]    validation_0-error:0.06191
```

```
[54]    validation_0-error:0.06667
[55]    validation_0-error:0.06429
[56]    validation_0-error:0.06191
[57]    validation_0-error:0.06191
[58]    validation_0-error:0.06905
[59]    validation_0-error:0.06905
[60]    validation_0-error:0.06667
[61]    validation_0-error:0.06905
[62]    validation_0-error:0.06667
[63]    validation_0-error:0.06667
[64]    validation_0-error:0.06667
[65]    validation_0-error:0.06667
[66]    validation_0-error:0.06667
[67]    validation_0-error:0.06667
[68]    validation_0-error:0.06429
[69]    validation_0-error:0.07143
[70]    validation_0-error:0.06667
[71]    validation_0-error:0.06905
[72]    validation_0-error:0.06667
[73]    validation_0-error:0.06667
[74]    validation_0-error:0.06905
[75]    validation_0-error:0.05952
[76]    validation_0-error:0.06191
[77]    validation_0-error:0.06667
[78]    validation_0-error:0.06429
[79]    validation_0-error:0.06667
[80]    validation_0-error:0.06429
[81]    validation_0-error:0.06429
[82]    validation_0-error:0.06667
[83]    validation_0-error:0.06429
[84]    validation_0-error:0.06905
[85]    validation_0-error:0.06905
[86]    validation_0-error:0.06191
[87]    validation_0-error:0.06191
[88]    validation_0-error:0.06191
[89]    validation_0-error:0.05952
[90]    validation_0-error:0.05952
[91]    validation_0-error:0.05952
[92]    validation_0-error:0.05952
[93]    validation_0-error:0.06191
[94]    validation_0-error:0.06191
[95]    validation_0-error:0.06429
[96]    validation_0-error:0.06429
[97]    validation_0-error:0.06429
[98]    validation_0-error:0.06429
[99]    validation_0-error:0.06667
[0]     validation_0-error:0.08571
[1]     validation_0-error:0.07619
```

```
[2]     validation_0-error:0.07619
[3]     validation_0-error:0.06905
[4]     validation_0-error:0.06191
[5]     validation_0-error:0.06191
[6]     validation_0-error:0.06429
[7]     validation_0-error:0.06191
[8]     validation_0-error:0.05714
[9]     validation_0-error:0.06191
[10]    validation_0-error:0.05952
[11]    validation_0-error:0.06191
[12]    validation_0-error:0.06667
[13]    validation_0-error:0.06905
[14]    validation_0-error:0.07143
[15]    validation_0-error:0.06667
[16]    validation_0-error:0.06667
[17]    validation_0-error:0.06429
[18]    validation_0-error:0.06667
[19]    validation_0-error:0.06191
[20]    validation_0-error:0.06429
[21]    validation_0-error:0.06191
[22]    validation_0-error:0.06429
[23]    validation_0-error:0.06667
[24]    validation_0-error:0.06905
[25]    validation_0-error:0.06667
[26]    validation_0-error:0.06429
[27]    validation_0-error:0.06429
[28]    validation_0-error:0.06905
[29]    validation_0-error:0.07143
[30]    validation_0-error:0.06905
[31]    validation_0-error:0.06905
[32]    validation_0-error:0.06905
[33]    validation_0-error:0.06905
[34]    validation_0-error:0.06429
[35]    validation_0-error:0.06905
[36]    validation_0-error:0.06667
[37]    validation_0-error:0.06191
[38]    validation_0-error:0.06191
[39]    validation_0-error:0.06191
[40]    validation_0-error:0.05714
[41]    validation_0-error:0.05714
[42]    validation_0-error:0.05952
[43]    validation_0-error:0.05952
[44]    validation_0-error:0.05952
[45]    validation_0-error:0.05952
[46]    validation_0-error:0.05714
[47]    validation_0-error:0.06191
[48]    validation_0-error:0.06191
[49]    validation_0-error:0.06191
```

```
[50]    validation_0-error:0.06191
[51]    validation_0-error:0.06191
[52]    validation_0-error:0.06191
[53]    validation_0-error:0.06191
[54]    validation_0-error:0.06667
[55]    validation_0-error:0.06429
[56]    validation_0-error:0.06191
[57]    validation_0-error:0.06191
[58]    validation_0-error:0.06905
[59]    validation_0-error:0.06905
[60]    validation_0-error:0.06667
[61]    validation_0-error:0.06905
[62]    validation_0-error:0.06667
[63]    validation_0-error:0.06667
[64]    validation_0-error:0.06667
[65]    validation_0-error:0.06667
[66]    validation_0-error:0.06667
[67]    validation_0-error:0.06667
[68]    validation_0-error:0.06429
[69]    validation_0-error:0.07143
[70]    validation_0-error:0.06667
[71]    validation_0-error:0.06905
[72]    validation_0-error:0.06667
[73]    validation_0-error:0.06667
[74]    validation_0-error:0.06905
[75]    validation_0-error:0.05952
[76]    validation_0-error:0.06191
[77]    validation_0-error:0.06667
[78]    validation_0-error:0.06429
[79]    validation_0-error:0.06667
[80]    validation_0-error:0.06429
[81]    validation_0-error:0.06429
[82]    validation_0-error:0.06667
[83]    validation_0-error:0.06429
[84]    validation_0-error:0.06905
[85]    validation_0-error:0.06905
[86]    validation_0-error:0.06191
[87]    validation_0-error:0.06191
[88]    validation_0-error:0.06191
[89]    validation_0-error:0.05952
[90]    validation_0-error:0.05952
[91]    validation_0-error:0.05952
[92]    validation_0-error:0.05952
[93]    validation_0-error:0.06191
[94]    validation_0-error:0.06191
[95]    validation_0-error:0.06429
[96]    validation_0-error:0.06429
[97]    validation_0-error:0.06429
```

```
[98]    validation_0-error:0.06429
[99]    validation_0-error:0.06667
[100]   validation_0-error:0.06667
[101]   validation_0-error:0.06429
[102]   validation_0-error:0.06429
[103]   validation_0-error:0.06429
[104]   validation_0-error:0.06667
[105]   validation_0-error:0.06191
[106]   validation_0-error:0.06191
[107]   validation_0-error:0.06429
[0]     validation_0-error:0.08571
[1]     validation_0-error:0.08095
[2]     validation_0-error:0.08095
[3]     validation_0-error:0.06905
[4]     validation_0-error:0.06429
[5]     validation_0-error:0.06191
[6]     validation_0-error:0.06429
[7]     validation_0-error:0.06429
[8]     validation_0-error:0.06191
[9]     validation_0-error:0.06191
[10]    validation_0-error:0.05952
[11]    validation_0-error:0.05714
[12]    validation_0-error:0.05714
[13]    validation_0-error:0.06667
[14]    validation_0-error:0.06667
[15]    validation_0-error:0.06905
[16]    validation_0-error:0.07143
[17]    validation_0-error:0.06191
[18]    validation_0-error:0.06191
[19]    validation_0-error:0.06191
[20]    validation_0-error:0.05952
[21]    validation_0-error:0.05952
[22]    validation_0-error:0.05714
[23]    validation_0-error:0.05714
[24]    validation_0-error:0.05714
[25]    validation_0-error:0.05714
[26]    validation_0-error:0.05952
[27]    validation_0-error:0.05952
[28]    validation_0-error:0.05952
[29]    validation_0-error:0.05714
[30]    validation_0-error:0.05714
[31]    validation_0-error:0.05714
[32]    validation_0-error:0.05952
[33]    validation_0-error:0.06191
[34]    validation_0-error:0.06191
[35]    validation_0-error:0.05952
[36]    validation_0-error:0.05952
[37]    validation_0-error:0.06191
```

```
[38]    validation_0-error:0.06429
[39]    validation_0-error:0.06191
[40]    validation_0-error:0.05714
[41]    validation_0-error:0.05714
[42]    validation_0-error:0.05238
[43]    validation_0-error:0.05238
[44]    validation_0-error:0.05238
[45]    validation_0-error:0.05714
[46]    validation_0-error:0.05714
[47]    validation_0-error:0.05714
[48]    validation_0-error:0.05714
[49]    validation_0-error:0.06191
[50]    validation_0-error:0.06191
[51]    validation_0-error:0.05952
[52]    validation_0-error:0.06191
[53]    validation_0-error:0.06191
[54]    validation_0-error:0.06191
[55]    validation_0-error:0.06191
[56]    validation_0-error:0.06429
[57]    validation_0-error:0.06191
[58]    validation_0-error:0.06191
[59]    validation_0-error:0.06429
[60]    validation_0-error:0.06429
[61]    validation_0-error:0.06429
[62]    validation_0-error:0.06429
[63]    validation_0-error:0.06191
[64]    validation_0-error:0.06191
[65]    validation_0-error:0.06191
[66]    validation_0-error:0.06191
[67]    validation_0-error:0.06191
[68]    validation_0-error:0.06429
[69]    validation_0-error:0.06191
[70]    validation_0-error:0.06429
[71]    validation_0-error:0.06429
[72]    validation_0-error:0.06191
[73]    validation_0-error:0.05952
[74]    validation_0-error:0.06429
[75]    validation_0-error:0.06191
[76]    validation_0-error:0.06429
[77]    validation_0-error:0.06429
[78]    validation_0-error:0.06191
[79]    validation_0-error:0.06429
[80]    validation_0-error:0.06429
[81]    validation_0-error:0.06429
[82]    validation_0-error:0.06429
[83]    validation_0-error:0.06429
[84]    validation_0-error:0.06429
[85]    validation_0-error:0.06429
```

```
[86]    validation_0-error:0.06191
[87]    validation_0-error:0.06191
[88]    validation_0-error:0.06191
[89]    validation_0-error:0.06191
[90]    validation_0-error:0.06191
[91]    validation_0-error:0.05952
[92]    validation_0-error:0.06191
[93]    validation_0-error:0.06191
[94]    validation_0-error:0.06191
[95]    validation_0-error:0.06191
[96]    validation_0-error:0.06191
[97]    validation_0-error:0.06191
[98]    validation_0-error:0.06191
[99]    validation_0-error:0.06191
[100]   validation_0-error:0.06191
[101]   validation_0-error:0.06191
[102]   validation_0-error:0.06191
[103]   validation_0-error:0.06191
[104]   validation_0-error:0.06191
[105]   validation_0-error:0.06191
[106]   validation_0-error:0.06191
[107]   validation_0-error:0.06191
[108]   validation_0-error:0.06191
[109]   validation_0-error:0.05952
[110]   validation_0-error:0.05952
[111]   validation_0-error:0.05952
[112]   validation_0-error:0.05952
[113]   validation_0-error:0.05952
[114]   validation_0-error:0.05952
[115]   validation_0-error:0.05952
[116]   validation_0-error:0.05714
[117]   validation_0-error:0.05714
[118]   validation_0-error:0.05714
[119]   validation_0-error:0.05714
[120]   validation_0-error:0.05714
[121]   validation_0-error:0.05714
[122]   validation_0-error:0.05714
[123]   validation_0-error:0.05714
[124]   validation_0-error:0.05714
[125]   validation_0-error:0.05714
[126]   validation_0-error:0.05714
[127]   validation_0-error:0.05714
[128]   validation_0-error:0.05714
[129]   validation_0-error:0.05714
[130]   validation_0-error:0.05714
[131]   validation_0-error:0.05714
[132]   validation_0-error:0.05714
[133]   validation_0-error:0.05714
```

```
[134]   validation_0-error:0.05714
[135]   validation_0-error:0.05714
[136]   validation_0-error:0.05952
[137]   validation_0-error:0.05714
[138]   validation_0-error:0.05714
[139]   validation_0-error:0.05952
[140]   validation_0-error:0.05952
[141]   validation_0-error:0.05952
[0]     validation_0-error:0.08571
[1]     validation_0-error:0.07619
[2]     validation_0-error:0.07619
[3]     validation_0-error:0.06905
[4]     validation_0-error:0.06191
[5]     validation_0-error:0.06191
[6]     validation_0-error:0.06429
[7]     validation_0-error:0.06191
[8]     validation_0-error:0.05714
[9]     validation_0-error:0.06191
[10]    validation_0-error:0.05952
[11]    validation_0-error:0.06191
[12]    validation_0-error:0.06667
[13]    validation_0-error:0.06905
[14]    validation_0-error:0.07143
[15]    validation_0-error:0.06667
[16]    validation_0-error:0.06667
[17]    validation_0-error:0.06429
[18]    validation_0-error:0.06667
[19]    validation_0-error:0.06191
[20]    validation_0-error:0.06429
[21]    validation_0-error:0.06191
[22]    validation_0-error:0.06429
[23]    validation_0-error:0.06667
[24]    validation_0-error:0.06905
[25]    validation_0-error:0.06667
[26]    validation_0-error:0.06429
[27]    validation_0-error:0.06429
[28]    validation_0-error:0.06905
[29]    validation_0-error:0.07143
[30]    validation_0-error:0.06905
[31]    validation_0-error:0.06905
[32]    validation_0-error:0.06905
[33]    validation_0-error:0.06905
[34]    validation_0-error:0.06429
[35]    validation_0-error:0.06905
[36]    validation_0-error:0.06667
[37]    validation_0-error:0.06191
[38]    validation_0-error:0.06191
[39]    validation_0-error:0.06191
```

```
[40]    validation_0-error:0.05714
[41]    validation_0-error:0.05714
[42]    validation_0-error:0.05952
[43]    validation_0-error:0.05952
[44]    validation_0-error:0.05952
[45]    validation_0-error:0.05952
[46]    validation_0-error:0.05714
[47]    validation_0-error:0.06191
[48]    validation_0-error:0.06191
[49]    validation_0-error:0.06191
[50]    validation_0-error:0.06191
[51]    validation_0-error:0.06191
[52]    validation_0-error:0.06191
[53]    validation_0-error:0.06191
[54]    validation_0-error:0.06667
[55]    validation_0-error:0.06429
[56]    validation_0-error:0.06191
[57]    validation_0-error:0.06191
[58]    validation_0-error:0.06905
[59]    validation_0-error:0.06905
[60]    validation_0-error:0.06667
[61]    validation_0-error:0.06905
[62]    validation_0-error:0.06667
[63]    validation_0-error:0.06667
[64]    validation_0-error:0.06667
[65]    validation_0-error:0.06667
[66]    validation_0-error:0.06667
[67]    validation_0-error:0.06667
[68]    validation_0-error:0.06429
[69]    validation_0-error:0.07143
[70]    validation_0-error:0.06667
[71]    validation_0-error:0.06905
[72]    validation_0-error:0.06667
[73]    validation_0-error:0.06667
[74]    validation_0-error:0.06905
[75]    validation_0-error:0.05952
[76]    validation_0-error:0.06191
[77]    validation_0-error:0.06667
[78]    validation_0-error:0.06429
[79]    validation_0-error:0.06667
[80]    validation_0-error:0.06429
[81]    validation_0-error:0.06429
[82]    validation_0-error:0.06667
[83]    validation_0-error:0.06429
[84]    validation_0-error:0.06905
[85]    validation_0-error:0.06905
[86]    validation_0-error:0.06191
[87]    validation_0-error:0.06191
```

```
[88]    validation_0-error:0.06191
[89]    validation_0-error:0.05952
[90]    validation_0-error:0.05952
[91]    validation_0-error:0.05952
[92]    validation_0-error:0.05952
[93]    validation_0-error:0.06191
[94]    validation_0-error:0.06191
[95]    validation_0-error:0.06429
[96]    validation_0-error:0.06429
[97]    validation_0-error:0.06429
[98]    validation_0-error:0.06429
[99]    validation_0-error:0.06667
[0]     validation_0-error:0.08571
[1]     validation_0-error:0.08095
[2]     validation_0-error:0.08095
[3]     validation_0-error:0.06667
[4]     validation_0-error:0.06429
[5]     validation_0-error:0.06191
[6]     validation_0-error:0.06429
[7]     validation_0-error:0.06667
[8]     validation_0-error:0.06191
[9]     validation_0-error:0.06429
[10]    validation_0-error:0.05714
[11]    validation_0-error:0.05476
[12]    validation_0-error:0.05476
[13]    validation_0-error:0.05952
[14]    validation_0-error:0.06191
[15]    validation_0-error:0.06905
[16]    validation_0-error:0.06905
[17]    validation_0-error:0.07143
[18]    validation_0-error:0.06429
[19]    validation_0-error:0.06191
[20]    validation_0-error:0.06191
[21]    validation_0-error:0.05952
[22]    validation_0-error:0.05714
[23]    validation_0-error:0.05714
[24]    validation_0-error:0.05714
[25]    validation_0-error:0.06429
[26]    validation_0-error:0.05952
[27]    validation_0-error:0.06191
[28]    validation_0-error:0.06429
[29]    validation_0-error:0.05952
[30]    validation_0-error:0.05952
[31]    validation_0-error:0.06191
[32]    validation_0-error:0.06429
[33]    validation_0-error:0.05952
[34]    validation_0-error:0.05952
[35]    validation_0-error:0.05952
```

```
[36]    validation_0-error:0.05714
[37]    validation_0-error:0.05952
[38]    validation_0-error:0.05952
[39]    validation_0-error:0.06191
[40]    validation_0-error:0.05952
[41]    validation_0-error:0.05952
[42]    validation_0-error:0.05952
[43]    validation_0-error:0.05952
[44]    validation_0-error:0.05952
[45]    validation_0-error:0.05952
[46]    validation_0-error:0.05952
[47]    validation_0-error:0.05952
[48]    validation_0-error:0.05952
[49]    validation_0-error:0.05952
[50]    validation_0-error:0.05952
[51]    validation_0-error:0.05952
[52]    validation_0-error:0.06191
[53]    validation_0-error:0.06191
[54]    validation_0-error:0.06191
[55]    validation_0-error:0.06191
[56]    validation_0-error:0.06191
[57]    validation_0-error:0.05952
[58]    validation_0-error:0.05952
[59]    validation_0-error:0.06429
[60]    validation_0-error:0.06429
[61]    validation_0-error:0.06191
[62]    validation_0-error:0.06429
[63]    validation_0-error:0.06429
[64]    validation_0-error:0.06429
[65]    validation_0-error:0.06429
[66]    validation_0-error:0.06429
[67]    validation_0-error:0.06429
[68]    validation_0-error:0.06191
[69]    validation_0-error:0.06191
[70]    validation_0-error:0.06429
[71]    validation_0-error:0.06191
[72]    validation_0-error:0.06191
[73]    validation_0-error:0.06191
[74]    validation_0-error:0.06191
[75]    validation_0-error:0.06191
[76]    validation_0-error:0.06191
[77]    validation_0-error:0.06191
[78]    validation_0-error:0.06191
[79]    validation_0-error:0.06191
[80]    validation_0-error:0.06191
[81]    validation_0-error:0.06191
[82]    validation_0-error:0.06191
[83]    validation_0-error:0.06191
```

```
[84]    validation_0-error:0.06191
[85]    validation_0-error:0.06191
[86]    validation_0-error:0.06191
[87]    validation_0-error:0.06191
[88]    validation_0-error:0.06191
[89]    validation_0-error:0.06191
[90]    validation_0-error:0.06191
[91]    validation_0-error:0.06191
[92]    validation_0-error:0.06191
[93]    validation_0-error:0.06191
[94]    validation_0-error:0.06191
[95]    validation_0-error:0.06191
[96]    validation_0-error:0.06191
[97]    validation_0-error:0.06191
[98]    validation_0-error:0.06191
[99]    validation_0-error:0.06191
[0]     validation_0-error:0.08571
[1]     validation_0-error:0.07619
[2]     validation_0-error:0.07619
[3]     validation_0-error:0.06905
[4]     validation_0-error:0.06191
[5]     validation_0-error:0.06191
[6]     validation_0-error:0.06429
[7]     validation_0-error:0.06191
[8]     validation_0-error:0.05714
[9]     validation_0-error:0.06191
[10]    validation_0-error:0.05952
[11]    validation_0-error:0.06191
[12]    validation_0-error:0.06667
[13]    validation_0-error:0.06905
[14]    validation_0-error:0.07143
[15]    validation_0-error:0.06667
[16]    validation_0-error:0.06667
[17]    validation_0-error:0.06429
[18]    validation_0-error:0.06667
[19]    validation_0-error:0.06191
[20]    validation_0-error:0.06429
[21]    validation_0-error:0.06191
[22]    validation_0-error:0.06429
[23]    validation_0-error:0.06667
[24]    validation_0-error:0.06905
[25]    validation_0-error:0.06667
[26]    validation_0-error:0.06429
[27]    validation_0-error:0.06429
[28]    validation_0-error:0.06905
[29]    validation_0-error:0.07143
[30]    validation_0-error:0.06905
[31]    validation_0-error:0.06905
```

```
[32]    validation_0-error:0.06905
[33]    validation_0-error:0.06905
[34]    validation_0-error:0.06429
[35]    validation_0-error:0.06905
[36]    validation_0-error:0.06667
[37]    validation_0-error:0.06191
[38]    validation_0-error:0.06191
[39]    validation_0-error:0.06191
[40]    validation_0-error:0.05714
[41]    validation_0-error:0.05714
[42]    validation_0-error:0.05952
[43]    validation_0-error:0.05952
[44]    validation_0-error:0.05952
[45]    validation_0-error:0.05952
[46]    validation_0-error:0.05714
[47]    validation_0-error:0.06191
[48]    validation_0-error:0.06191
[49]    validation_0-error:0.06191
[50]    validation_0-error:0.06191
[51]    validation_0-error:0.06191
[52]    validation_0-error:0.06191
[53]    validation_0-error:0.06191
[54]    validation_0-error:0.06667
[55]    validation_0-error:0.06429
[56]    validation_0-error:0.06191
[57]    validation_0-error:0.06191
[58]    validation_0-error:0.06905
[59]    validation_0-error:0.06905
[60]    validation_0-error:0.06667
[61]    validation_0-error:0.06905
[62]    validation_0-error:0.06667
[63]    validation_0-error:0.06667
[64]    validation_0-error:0.06667
[65]    validation_0-error:0.06667
[66]    validation_0-error:0.06667
[67]    validation_0-error:0.06667
[68]    validation_0-error:0.06429
[69]    validation_0-error:0.07143
[70]    validation_0-error:0.06667
[71]    validation_0-error:0.06905
[72]    validation_0-error:0.06667
[73]    validation_0-error:0.06667
[74]    validation_0-error:0.06905
[75]    validation_0-error:0.05952
[76]    validation_0-error:0.06191
[77]    validation_0-error:0.06667
[78]    validation_0-error:0.06429
[79]    validation_0-error:0.06667
```

```
[80]    validation_0-error:0.06429
[81]    validation_0-error:0.06429
[82]    validation_0-error:0.06667
[83]    validation_0-error:0.06429
[84]    validation_0-error:0.06905
[85]    validation_0-error:0.06905
[86]    validation_0-error:0.06191
[87]    validation_0-error:0.06191
[88]    validation_0-error:0.06191
[89]    validation_0-error:0.05952
[90]    validation_0-error:0.05952
[91]    validation_0-error:0.05952
[92]    validation_0-error:0.05952
[93]    validation_0-error:0.06191
[94]    validation_0-error:0.06191
[95]    validation_0-error:0.06429
[96]    validation_0-error:0.06429
[97]    validation_0-error:0.06429
[98]    validation_0-error:0.06429
[99]    validation_0-error:0.06667
```

[19]:
```
[[0.1, 100, 0.9428571428571428],
 [0.1, 400, 0.9428571428571428],
 [0.01, 400, 0.9476190476190476],
 [0.1, 100, 0.9428571428571428],
 [0.05, 100, 0.9452380952380952],
 [0.1, 100, 0.9428571428571428]]
```

0.2.2 CV

[21]:
```
from sklearn.model_selection import StratifiedKFold ## recommended for␣
 ↪classification
kfolds = StratifiedKFold(n_splits = 10, random_state = 1, shuffle = True)
```

[22]:
```
reg = ['l2', 'l2', 'l2', 'l1']
C = [1,0.2,5,1,1,0.2,0.6]
sol = ['lbfgs', 'lbfgs', 'lbfgs', 'liblinear']
```

[23]:
```
X_std=stdsc.transform(X)
```

[27]:
```
from sklearn.model_selection import cross_val_score
d = []
for i in range(len(reg)):
    lg = LogisticRegression(penalty = reg[i] ,solver=sol[i], C = C[i],␣
 ↪class_weight='balanced', max_iter=200 )
    lg.fit(X_train_std, y_train)
    scoring = lg.score(X_test_std, y_test)
```

```
    error= cross_val_score(lg, X_std, y, cv=kfolds)
    d.append([reg[i], C[i], sol[i], scoring,np.mean(error)])
d
```

[27]:
```
[['l2', 1, 'lbfgs', 0.9238095238095239, 0.9160714285714286],
 ['l2', 0.2, 'lbfgs', 0.9214285714285714, 0.9166666666666666],
 ['l2', 5, 'lbfgs', 0.9238095238095239, 0.9154761904761903],
 ['l1', 1, 'liblinear', 0.9238095238095239, 0.9160714285714286]]
```

[28]:
```
num = [50, 75, 100, 200, 400, 100, 200]
criterion=['entropy','entropy','entropy','entropy','entropy', 'gini', 'gini']
```

[29]:
```
rf = []
for i in range(len(num)):
    classifier_RF = RandomForestClassifier(n_estimators = num[i],␣
 ↪criterion=criterion[i],
                                           random_state=1,
                                           verbose = 1,
                                           oob_score=True)
    classifier_RF.fit(X_train, y_train)
    scoring = classifier_RF.score(X_test, y_test)
    error= cross_val_score(classifier_RF, X, y, cv=kfolds)
    rf.append([num[i], criterion[i], scoring, np.mean(error)])

rf
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
```

```
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
```

```
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.2s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  75 out of  75 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.2s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.2s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.2s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
```

```
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.2s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.2s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.4s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
```

```
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.4s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.4s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.5s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.6s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.7s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.7s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    1.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.7s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.7s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.6s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.7s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
```

```
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.8s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.8s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 400 out of 400 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.2s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.1s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.2s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.2s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.2s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
```

```
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.2s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.5s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.4s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.4s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.4s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.4s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed:    0.0s finished
```

[29]:
```
[[50, 'entropy', 0.9333333333333333, 0.9345238095238095],
 [75, 'entropy', 0.9357142857142857, 0.931547619047619],
 [100, 'entropy', 0.9357142857142857, 0.9327380952380953],
 [200, 'entropy', 0.9380952380952381, 0.9363095238095237],
 [400, 'entropy', 0.9333333333333333, 0.9351190476190474],
 [100, 'gini', 0.9404761904761905, 0.9357142857142857],
 [200, 'gini', 0.9380952380952381, 0.9363095238095237]]
```

[30]:
```
num = [100, 400, 400, 100, 100, 100]
eta = [0.1,0.1,0.01,0.1, 0.05, 0.1]
# depth = [6, 10, 10, 10, 10, 12]
```

[31]:
```
f = []
for i in range(len(num)):
    xgb_model = XGBClassifier(objective = 'binary:logistic',
                              learning_rate = eta[i],
                              n_estimators = num[i],
                              # max_depth = depth[i],
                              subsample = 0.8,
                              colsample_bytree = 0.8,
                              # reg_lambda = l2_reg[i] ,
                              random_state = 1,
                              use_label_encoder=False
                              )

    xgb_model.fit(X_train, y_train, eval_set=[(X_test, y_test)],␣
 ↪eval_metric='error', early_stopping_rounds=100)
    scoring = xgb_model.score(X_test, y_test)
    error= cross_val_score(xgb_model, X, y, cv=kfolds)

    f.append([eta[i], num[i], scoring, np.mean(error)])
f
```

```
[0]     validation_0-error:0.08571
[1]     validation_0-error:0.07619
[2]     validation_0-error:0.07619
[3]     validation_0-error:0.06905
[4]     validation_0-error:0.06191
[5]     validation_0-error:0.06191
[6]     validation_0-error:0.06429
[7]     validation_0-error:0.06191
[8]     validation_0-error:0.05714
[9]     validation_0-error:0.06191
[10]    validation_0-error:0.05952
[11]    validation_0-error:0.06191
[12]    validation_0-error:0.06667
[13]    validation_0-error:0.06905
```

```
[14]    validation_0-error:0.07143
[15]    validation_0-error:0.06667
[16]    validation_0-error:0.06667
[17]    validation_0-error:0.06429
[18]    validation_0-error:0.06667
[19]    validation_0-error:0.06191
[20]    validation_0-error:0.06429
[21]    validation_0-error:0.06191
[22]    validation_0-error:0.06429
[23]    validation_0-error:0.06667
[24]    validation_0-error:0.06905
[25]    validation_0-error:0.06667
[26]    validation_0-error:0.06429
[27]    validation_0-error:0.06429
[28]    validation_0-error:0.06905
[29]    validation_0-error:0.07143
[30]    validation_0-error:0.06905
[31]    validation_0-error:0.06905
[32]    validation_0-error:0.06905
[33]    validation_0-error:0.06905
[34]    validation_0-error:0.06429
[35]    validation_0-error:0.06905
[36]    validation_0-error:0.06667
[37]    validation_0-error:0.06191
[38]    validation_0-error:0.06191
[39]    validation_0-error:0.06191
[40]    validation_0-error:0.05714
[41]    validation_0-error:0.05714
[42]    validation_0-error:0.05952
[43]    validation_0-error:0.05952
[44]    validation_0-error:0.05952
[45]    validation_0-error:0.05952
[46]    validation_0-error:0.05714
[47]    validation_0-error:0.06191
[48]    validation_0-error:0.06191
[49]    validation_0-error:0.06191
[50]    validation_0-error:0.06191
[51]    validation_0-error:0.06191
[52]    validation_0-error:0.06191
[53]    validation_0-error:0.06191
[54]    validation_0-error:0.06667
[55]    validation_0-error:0.06429
[56]    validation_0-error:0.06191
[57]    validation_0-error:0.06191
[58]    validation_0-error:0.06905
[59]    validation_0-error:0.06905
[60]    validation_0-error:0.06667
[61]    validation_0-error:0.06905
```

```
[62]    validation_0-error:0.06667
[63]    validation_0-error:0.06667
[64]    validation_0-error:0.06667
[65]    validation_0-error:0.06667
[66]    validation_0-error:0.06667
[67]    validation_0-error:0.06667
[68]    validation_0-error:0.06429
[69]    validation_0-error:0.07143
[70]    validation_0-error:0.06667
[71]    validation_0-error:0.06905
[72]    validation_0-error:0.06667
[73]    validation_0-error:0.06667
[74]    validation_0-error:0.06905
[75]    validation_0-error:0.05952
[76]    validation_0-error:0.06191
[77]    validation_0-error:0.06667
[78]    validation_0-error:0.06429
[79]    validation_0-error:0.06667
[80]    validation_0-error:0.06429
[81]    validation_0-error:0.06429
[82]    validation_0-error:0.06667
[83]    validation_0-error:0.06429
[84]    validation_0-error:0.06905
[85]    validation_0-error:0.06905
[86]    validation_0-error:0.06191
[87]    validation_0-error:0.06191
[88]    validation_0-error:0.06191
[89]    validation_0-error:0.05952
[90]    validation_0-error:0.05952
[91]    validation_0-error:0.05952
[92]    validation_0-error:0.05952
[93]    validation_0-error:0.06191
[94]    validation_0-error:0.06191
[95]    validation_0-error:0.06429
[96]    validation_0-error:0.06429
[97]    validation_0-error:0.06429
[98]    validation_0-error:0.06429
[99]    validation_0-error:0.06667
[23:30:58] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:30:58] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:30:58] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
```

```
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:30:58] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:30:58] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:30:59] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:30:59] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:30:59] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:30:59] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:30:59] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[0]	validation_0-error:0.08571
[1]	validation_0-error:0.07619
[2]	validation_0-error:0.07619
[3]	validation_0-error:0.06905
[4]	validation_0-error:0.06191
[5]	validation_0-error:0.06191
[6]	validation_0-error:0.06429
[7]	validation_0-error:0.06191
[8]	validation_0-error:0.05714
[9]	validation_0-error:0.06191
[10]	validation_0-error:0.05952
[11]	validation_0-error:0.06191
[12]	validation_0-error:0.06667
[13]	validation_0-error:0.06905
[14]	validation_0-error:0.07143
[15]	validation_0-error:0.06667
[16]	validation_0-error:0.06667
[17]	validation_0-error:0.06429
```

```
[18]    validation_0-error:0.06667
[19]    validation_0-error:0.06191
[20]    validation_0-error:0.06429
[21]    validation_0-error:0.06191
[22]    validation_0-error:0.06429
[23]    validation_0-error:0.06667
[24]    validation_0-error:0.06905
[25]    validation_0-error:0.06667
[26]    validation_0-error:0.06429
[27]    validation_0-error:0.06429
[28]    validation_0-error:0.06905
[29]    validation_0-error:0.07143
[30]    validation_0-error:0.06905
[31]    validation_0-error:0.06905
[32]    validation_0-error:0.06905
[33]    validation_0-error:0.06905
[34]    validation_0-error:0.06429
[35]    validation_0-error:0.06905
[36]    validation_0-error:0.06667
[37]    validation_0-error:0.06191
[38]    validation_0-error:0.06191
[39]    validation_0-error:0.06191
[40]    validation_0-error:0.05714
[41]    validation_0-error:0.05714
[42]    validation_0-error:0.05952
[43]    validation_0-error:0.05952
[44]    validation_0-error:0.05952
[45]    validation_0-error:0.05952
[46]    validation_0-error:0.05714
[47]    validation_0-error:0.06191
[48]    validation_0-error:0.06191
[49]    validation_0-error:0.06191
[50]    validation_0-error:0.06191
[51]    validation_0-error:0.06191
[52]    validation_0-error:0.06191
[53]    validation_0-error:0.06191
[54]    validation_0-error:0.06667
[55]    validation_0-error:0.06429
[56]    validation_0-error:0.06191
[57]    validation_0-error:0.06191
[58]    validation_0-error:0.06905
[59]    validation_0-error:0.06905
[60]    validation_0-error:0.06667
[61]    validation_0-error:0.06905
[62]    validation_0-error:0.06667
[63]    validation_0-error:0.06667
[64]    validation_0-error:0.06667
[65]    validation_0-error:0.06667
```

```
[66]    validation_0-error:0.06667
[67]    validation_0-error:0.06667
[68]    validation_0-error:0.06429
[69]    validation_0-error:0.07143
[70]    validation_0-error:0.06667
[71]    validation_0-error:0.06905
[72]    validation_0-error:0.06667
[73]    validation_0-error:0.06667
[74]    validation_0-error:0.06905
[75]    validation_0-error:0.05952
[76]    validation_0-error:0.06191
[77]    validation_0-error:0.06667
[78]    validation_0-error:0.06429
[79]    validation_0-error:0.06667
[80]    validation_0-error:0.06429
[81]    validation_0-error:0.06429
[82]    validation_0-error:0.06667
[83]    validation_0-error:0.06429
[84]    validation_0-error:0.06905
[85]    validation_0-error:0.06905
[86]    validation_0-error:0.06191
[87]    validation_0-error:0.06191
[88]    validation_0-error:0.06191
[89]    validation_0-error:0.05952
[90]    validation_0-error:0.05952
[91]    validation_0-error:0.05952
[92]    validation_0-error:0.05952
[93]    validation_0-error:0.06191
[94]    validation_0-error:0.06191
[95]    validation_0-error:0.06429
[96]    validation_0-error:0.06429
[97]    validation_0-error:0.06429
[98]    validation_0-error:0.06429
[99]    validation_0-error:0.06667
[100]   validation_0-error:0.06667
[101]   validation_0-error:0.06429
[102]   validation_0-error:0.06429
[103]   validation_0-error:0.06429
[104]   validation_0-error:0.06667
[105]   validation_0-error:0.06191
[106]   validation_0-error:0.06191
[107]   validation_0-error:0.06429
[23:31:00] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:00] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
```

```
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:01] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:01] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:02] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:02] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:03] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:03] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:03] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:04] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[0]	validation_0-error:0.08571
[1]	validation_0-error:0.08095
[2]	validation_0-error:0.08095
[3]	validation_0-error:0.06905
[4]	validation_0-error:0.06429
[5]	validation_0-error:0.06191
[6]	validation_0-error:0.06429
[7]	validation_0-error:0.06429
[8]	validation_0-error:0.06191
[9]	validation_0-error:0.06191
[10]	validation_0-error:0.05952
[11]	validation_0-error:0.05714
[12]	validation_0-error:0.05714
[13]	validation_0-error:0.06667
```

```
[14]    validation_0-error:0.06667
[15]    validation_0-error:0.06905
[16]    validation_0-error:0.07143
[17]    validation_0-error:0.06191
[18]    validation_0-error:0.06191
[19]    validation_0-error:0.06191
[20]    validation_0-error:0.05952
[21]    validation_0-error:0.05952
[22]    validation_0-error:0.05714
[23]    validation_0-error:0.05714
[24]    validation_0-error:0.05714
[25]    validation_0-error:0.05714
[26]    validation_0-error:0.05952
[27]    validation_0-error:0.05952
[28]    validation_0-error:0.05952
[29]    validation_0-error:0.05714
[30]    validation_0-error:0.05714
[31]    validation_0-error:0.05714
[32]    validation_0-error:0.05952
[33]    validation_0-error:0.06191
[34]    validation_0-error:0.06191
[35]    validation_0-error:0.05952
[36]    validation_0-error:0.05952
[37]    validation_0-error:0.06191
[38]    validation_0-error:0.06429
[39]    validation_0-error:0.06191
[40]    validation_0-error:0.05714
[41]    validation_0-error:0.05714
[42]    validation_0-error:0.05238
[43]    validation_0-error:0.05238
[44]    validation_0-error:0.05238
[45]    validation_0-error:0.05714
[46]    validation_0-error:0.05714
[47]    validation_0-error:0.05714
[48]    validation_0-error:0.05714
[49]    validation_0-error:0.06191
[50]    validation_0-error:0.06191
[51]    validation_0-error:0.05952
[52]    validation_0-error:0.06191
[53]    validation_0-error:0.06191
[54]    validation_0-error:0.06191
[55]    validation_0-error:0.06191
[56]    validation_0-error:0.06429
[57]    validation_0-error:0.06191
[58]    validation_0-error:0.06191
[59]    validation_0-error:0.06429
[60]    validation_0-error:0.06429
[61]    validation_0-error:0.06429
```

```
[62]    validation_0-error:0.06429
[63]    validation_0-error:0.06191
[64]    validation_0-error:0.06191
[65]    validation_0-error:0.06191
[66]    validation_0-error:0.06191
[67]    validation_0-error:0.06191
[68]    validation_0-error:0.06429
[69]    validation_0-error:0.06191
[70]    validation_0-error:0.06429
[71]    validation_0-error:0.06429
[72]    validation_0-error:0.06191
[73]    validation_0-error:0.05952
[74]    validation_0-error:0.06429
[75]    validation_0-error:0.06191
[76]    validation_0-error:0.06429
[77]    validation_0-error:0.06429
[78]    validation_0-error:0.06191
[79]    validation_0-error:0.06429
[80]    validation_0-error:0.06429
[81]    validation_0-error:0.06429
[82]    validation_0-error:0.06429
[83]    validation_0-error:0.06429
[84]    validation_0-error:0.06429
[85]    validation_0-error:0.06429
[86]    validation_0-error:0.06191
[87]    validation_0-error:0.06191
[88]    validation_0-error:0.06191
[89]    validation_0-error:0.06191
[90]    validation_0-error:0.06191
[91]    validation_0-error:0.05952
[92]    validation_0-error:0.06191
[93]    validation_0-error:0.06191
[94]    validation_0-error:0.06191
[95]    validation_0-error:0.06191
[96]    validation_0-error:0.06191
[97]    validation_0-error:0.06191
[98]    validation_0-error:0.06191
[99]    validation_0-error:0.06191
[100]   validation_0-error:0.06191
[101]   validation_0-error:0.06191
[102]   validation_0-error:0.06191
[103]   validation_0-error:0.06191
[104]   validation_0-error:0.06191
[105]   validation_0-error:0.06191
[106]   validation_0-error:0.06191
[107]   validation_0-error:0.06191
[108]   validation_0-error:0.06191
[109]   validation_0-error:0.05952
```

```
[110]   validation_0-error:0.05952
[111]   validation_0-error:0.05952
[112]   validation_0-error:0.05952
[113]   validation_0-error:0.05952
[114]   validation_0-error:0.05952
[115]   validation_0-error:0.05952
[116]   validation_0-error:0.05714
[117]   validation_0-error:0.05714
[118]   validation_0-error:0.05714
[119]   validation_0-error:0.05714
[120]   validation_0-error:0.05714
[121]   validation_0-error:0.05714
[122]   validation_0-error:0.05714
[123]   validation_0-error:0.05714
[124]   validation_0-error:0.05714
[125]   validation_0-error:0.05714
[126]   validation_0-error:0.05714
[127]   validation_0-error:0.05714
[128]   validation_0-error:0.05714
[129]   validation_0-error:0.05714
[130]   validation_0-error:0.05714
[131]   validation_0-error:0.05714
[132]   validation_0-error:0.05714
[133]   validation_0-error:0.05714
[134]   validation_0-error:0.05714
[135]   validation_0-error:0.05714
[136]   validation_0-error:0.05952
[137]   validation_0-error:0.05714
[138]   validation_0-error:0.05714
[139]   validation_0-error:0.05952
[140]   validation_0-error:0.05952
[141]   validation_0-error:0.05952
[23:31:05] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:06] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:06] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:07] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
```

```
[23:31:07] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:07] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:08] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:08] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:09] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:09] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[0]     validation_0-error:0.08571
[1]     validation_0-error:0.07619
[2]     validation_0-error:0.07619
[3]     validation_0-error:0.06905
[4]     validation_0-error:0.06191
[5]     validation_0-error:0.06191
[6]     validation_0-error:0.06429
[7]     validation_0-error:0.06191
[8]     validation_0-error:0.05714
[9]     validation_0-error:0.06191
[10]    validation_0-error:0.05952
[11]    validation_0-error:0.06191
[12]    validation_0-error:0.06667
[13]    validation_0-error:0.06905
[14]    validation_0-error:0.07143
[15]    validation_0-error:0.06667
[16]    validation_0-error:0.06667
[17]    validation_0-error:0.06429
[18]    validation_0-error:0.06667
[19]    validation_0-error:0.06191
[20]    validation_0-error:0.06429
[21]    validation_0-error:0.06191
[22]    validation_0-error:0.06429
[23]    validation_0-error:0.06667
```

```
[24]    validation_0-error:0.06905
[25]    validation_0-error:0.06667
[26]    validation_0-error:0.06429
[27]    validation_0-error:0.06429
[28]    validation_0-error:0.06905
[29]    validation_0-error:0.07143
[30]    validation_0-error:0.06905
[31]    validation_0-error:0.06905
[32]    validation_0-error:0.06905
[33]    validation_0-error:0.06905
[34]    validation_0-error:0.06429
[35]    validation_0-error:0.06905
[36]    validation_0-error:0.06667
[37]    validation_0-error:0.06191
[38]    validation_0-error:0.06191
[39]    validation_0-error:0.06191
[40]    validation_0-error:0.05714
[41]    validation_0-error:0.05714
[42]    validation_0-error:0.05952
[43]    validation_0-error:0.05952
[44]    validation_0-error:0.05952
[45]    validation_0-error:0.05952
[46]    validation_0-error:0.05714
[47]    validation_0-error:0.06191
[48]    validation_0-error:0.06191
[49]    validation_0-error:0.06191
[50]    validation_0-error:0.06191
[51]    validation_0-error:0.06191
[52]    validation_0-error:0.06191
[53]    validation_0-error:0.06191
[54]    validation_0-error:0.06667
[55]    validation_0-error:0.06429
[56]    validation_0-error:0.06191
[57]    validation_0-error:0.06191
[58]    validation_0-error:0.06905
[59]    validation_0-error:0.06905
[60]    validation_0-error:0.06667
[61]    validation_0-error:0.06905
[62]    validation_0-error:0.06667
[63]    validation_0-error:0.06667
[64]    validation_0-error:0.06667
[65]    validation_0-error:0.06667
[66]    validation_0-error:0.06667
[67]    validation_0-error:0.06667
[68]    validation_0-error:0.06429
[69]    validation_0-error:0.07143
[70]    validation_0-error:0.06667
[71]    validation_0-error:0.06905
```

```
[72]	validation_0-error:0.06667
[73]	validation_0-error:0.06667
[74]	validation_0-error:0.06905
[75]	validation_0-error:0.05952
[76]	validation_0-error:0.06191
[77]	validation_0-error:0.06667
[78]	validation_0-error:0.06429
[79]	validation_0-error:0.06667
[80]	validation_0-error:0.06429
[81]	validation_0-error:0.06429
[82]	validation_0-error:0.06667
[83]	validation_0-error:0.06429
[84]	validation_0-error:0.06905
[85]	validation_0-error:0.06905
[86]	validation_0-error:0.06191
[87]	validation_0-error:0.06191
[88]	validation_0-error:0.06191
[89]	validation_0-error:0.05952
[90]	validation_0-error:0.05952
[91]	validation_0-error:0.05952
[92]	validation_0-error:0.05952
[93]	validation_0-error:0.06191
[94]	validation_0-error:0.06191
[95]	validation_0-error:0.06429
[96]	validation_0-error:0.06429
[97]	validation_0-error:0.06429
[98]	validation_0-error:0.06429
[99]	validation_0-error:0.06667
[23:31:10] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:10] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:10] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:10] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:11] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
```

```
[23:31:11] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:11] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:11] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:11] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:11] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[0]	validation_0-error:0.08571
[1]	validation_0-error:0.08095
[2]	validation_0-error:0.08095
[3]	validation_0-error:0.06667
[4]	validation_0-error:0.06429
[5]	validation_0-error:0.06191
[6]	validation_0-error:0.06429
[7]	validation_0-error:0.06667
[8]	validation_0-error:0.06191
[9]	validation_0-error:0.06429
[10]	validation_0-error:0.05714
[11]	validation_0-error:0.05476
[12]	validation_0-error:0.05476
[13]	validation_0-error:0.05952
[14]	validation_0-error:0.06191
[15]	validation_0-error:0.06905
[16]	validation_0-error:0.06905
[17]	validation_0-error:0.07143
[18]	validation_0-error:0.06429
[19]	validation_0-error:0.06191
[20]	validation_0-error:0.06191
[21]	validation_0-error:0.05952
[22]	validation_0-error:0.05714
[23]	validation_0-error:0.05714
[24]	validation_0-error:0.05714
[25]	validation_0-error:0.06429
[26]	validation_0-error:0.05952
[27]	validation_0-error:0.06191
```

```
[28]    validation_0-error:0.06429
[29]    validation_0-error:0.05952
[30]    validation_0-error:0.05952
[31]    validation_0-error:0.06191
[32]    validation_0-error:0.06429
[33]    validation_0-error:0.05952
[34]    validation_0-error:0.05952
[35]    validation_0-error:0.05952
[36]    validation_0-error:0.05714
[37]    validation_0-error:0.05952
[38]    validation_0-error:0.05952
[39]    validation_0-error:0.06191
[40]    validation_0-error:0.05952
[41]    validation_0-error:0.05952
[42]    validation_0-error:0.05952
[43]    validation_0-error:0.05952
[44]    validation_0-error:0.05952
[45]    validation_0-error:0.05952
[46]    validation_0-error:0.05952
[47]    validation_0-error:0.05952
[48]    validation_0-error:0.05952
[49]    validation_0-error:0.05952
[50]    validation_0-error:0.05952
[51]    validation_0-error:0.05952
[52]    validation_0-error:0.06191
[53]    validation_0-error:0.06191
[54]    validation_0-error:0.06191
[55]    validation_0-error:0.06191
[56]    validation_0-error:0.06191
[57]    validation_0-error:0.05952
[58]    validation_0-error:0.05952
[59]    validation_0-error:0.06429
[60]    validation_0-error:0.06429
[61]    validation_0-error:0.06191
[62]    validation_0-error:0.06429
[63]    validation_0-error:0.06429
[64]    validation_0-error:0.06429
[65]    validation_0-error:0.06429
[66]    validation_0-error:0.06429
[67]    validation_0-error:0.06429
[68]    validation_0-error:0.06191
[69]    validation_0-error:0.06191
[70]    validation_0-error:0.06429
[71]    validation_0-error:0.06191
[72]    validation_0-error:0.06191
[73]    validation_0-error:0.06191
[74]    validation_0-error:0.06191
[75]    validation_0-error:0.06191
```

```
[76]	validation_0-error:0.06191
[77]	validation_0-error:0.06191
[78]	validation_0-error:0.06191
[79]	validation_0-error:0.06191
[80]	validation_0-error:0.06191
[81]	validation_0-error:0.06191
[82]	validation_0-error:0.06191
[83]	validation_0-error:0.06191
[84]	validation_0-error:0.06191
[85]	validation_0-error:0.06191
[86]	validation_0-error:0.06191
[87]	validation_0-error:0.06191
[88]	validation_0-error:0.06191
[89]	validation_0-error:0.06191
[90]	validation_0-error:0.06191
[91]	validation_0-error:0.06191
[92]	validation_0-error:0.06191
[93]	validation_0-error:0.06191
[94]	validation_0-error:0.06191
[95]	validation_0-error:0.06191
[96]	validation_0-error:0.06191
[97]	validation_0-error:0.06191
[98]	validation_0-error:0.06191
[99]	validation_0-error:0.06191
[23:31:12] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:12] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:12] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:12] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:12] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:12] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
```

```
[23:31:12] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:13] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:13] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:13] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[0]	validation_0-error:0.08571
[1]	validation_0-error:0.07619
[2]	validation_0-error:0.07619
[3]	validation_0-error:0.06905
[4]	validation_0-error:0.06191
[5]	validation_0-error:0.06191
[6]	validation_0-error:0.06429
[7]	validation_0-error:0.06191
[8]	validation_0-error:0.05714
[9]	validation_0-error:0.06191
[10]	validation_0-error:0.05952
[11]	validation_0-error:0.06191
[12]	validation_0-error:0.06667
[13]	validation_0-error:0.06905
[14]	validation_0-error:0.07143
[15]	validation_0-error:0.06667
[16]	validation_0-error:0.06667
[17]	validation_0-error:0.06429
[18]	validation_0-error:0.06667
[19]	validation_0-error:0.06191
[20]	validation_0-error:0.06429
[21]	validation_0-error:0.06191
[22]	validation_0-error:0.06429
[23]	validation_0-error:0.06667
[24]	validation_0-error:0.06905
[25]	validation_0-error:0.06667
[26]	validation_0-error:0.06429
[27]	validation_0-error:0.06429
[28]	validation_0-error:0.06905
[29]	validation_0-error:0.07143
[30]	validation_0-error:0.06905
[31]	validation_0-error:0.06905
```

```
[32]    validation_0-error:0.06905
[33]    validation_0-error:0.06905
[34]    validation_0-error:0.06429
[35]    validation_0-error:0.06905
[36]    validation_0-error:0.06667
[37]    validation_0-error:0.06191
[38]    validation_0-error:0.06191
[39]    validation_0-error:0.06191
[40]    validation_0-error:0.05714
[41]    validation_0-error:0.05714
[42]    validation_0-error:0.05952
[43]    validation_0-error:0.05952
[44]    validation_0-error:0.05952
[45]    validation_0-error:0.05952
[46]    validation_0-error:0.05714
[47]    validation_0-error:0.06191
[48]    validation_0-error:0.06191
[49]    validation_0-error:0.06191
[50]    validation_0-error:0.06191
[51]    validation_0-error:0.06191
[52]    validation_0-error:0.06191
[53]    validation_0-error:0.06191
[54]    validation_0-error:0.06667
[55]    validation_0-error:0.06429
[56]    validation_0-error:0.06191
[57]    validation_0-error:0.06191
[58]    validation_0-error:0.06905
[59]    validation_0-error:0.06905
[60]    validation_0-error:0.06667
[61]    validation_0-error:0.06905
[62]    validation_0-error:0.06667
[63]    validation_0-error:0.06667
[64]    validation_0-error:0.06667
[65]    validation_0-error:0.06667
[66]    validation_0-error:0.06667
[67]    validation_0-error:0.06667
[68]    validation_0-error:0.06429
[69]    validation_0-error:0.07143
[70]    validation_0-error:0.06667
[71]    validation_0-error:0.06905
[72]    validation_0-error:0.06667
[73]    validation_0-error:0.06667
[74]    validation_0-error:0.06905
[75]    validation_0-error:0.05952
[76]    validation_0-error:0.06191
[77]    validation_0-error:0.06667
[78]    validation_0-error:0.06429
[79]    validation_0-error:0.06667
```

```
[80]	validation_0-error:0.06429
[81]	validation_0-error:0.06429
[82]	validation_0-error:0.06667
[83]	validation_0-error:0.06429
[84]	validation_0-error:0.06905
[85]	validation_0-error:0.06905
[86]	validation_0-error:0.06191
[87]	validation_0-error:0.06191
[88]	validation_0-error:0.06191
[89]	validation_0-error:0.05952
[90]	validation_0-error:0.05952
[91]	validation_0-error:0.05952
[92]	validation_0-error:0.05952
[93]	validation_0-error:0.06191
[94]	validation_0-error:0.06191
[95]	validation_0-error:0.06429
[96]	validation_0-error:0.06429
[97]	validation_0-error:0.06429
[98]	validation_0-error:0.06429
[99]	validation_0-error:0.06667
[23:31:14] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:14] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:14] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:14] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:14] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:15] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:15] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
```

```
[23:31:15] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:15] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
[23:31:15] WARNING: /Users/travis/build/dmlc/xgboost/src/learner.cc:1061:
Starting in XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
```

[31]:
```
[[0.1, 100, 0.9428571428571428, 0.9351190476190476],
 [0.1, 400, 0.9428571428571428, 0.9392857142857143],
 [0.01, 400, 0.9476190476190476, 0.9369047619047619],
 [0.1, 100, 0.9428571428571428, 0.9351190476190476],
 [0.05, 100, 0.9452380952380952, 0.931547619047619],
 [0.1, 100, 0.9428571428571428, 0.9351190476190476]]
```

0.3 NN

[32]:
```python
from sklearn.neural_network import MLPClassifier
```

[36]:
```python
layer = [1,1,1,1,1,1]
node = [10,20,30,40,50,60]
epoch=[200,500,200,500,200,500]
```

[37]:
```python
nn = []
for i in range(len(layer)):
    nnl = MLPClassifier(hidden_layer_sizes=(node[i],),max_iter=epoch[i],activation='relu').fit(X_train, y_train)
    scoring = nnl.score(X_test, y_test)

    nn.append([layer[i], node[i], epoch[i], scoring])
```

[38]:
```python
nn
```

[38]:
```
[[1, 10, 200, 0.9119047619047619],
 [1, 20, 500, 0.9047619047619048],
 [1, 30, 200, 0.9],
 [1, 40, 500, 0.9095238095238095],
 [1, 50, 200, 0.9119047619047619],
 [1, 60, 500, 0.9142857142857143]]
```

0.4 FE

[56]:
```
df.head()
```

[56]:

	Value	S	K	tau	r	BS	binary_label
0	21.670404	431.623898	420	0.341270	0.03013	Under	0
1	0.125000	427.015526	465	0.166667	0.03126	Over	1
2	20.691244	427.762336	415	0.265873	0.03116	Under	0
3	1.035002	451.711658	460	0.063492	0.02972	Over	1
4	39.553020	446.718974	410	0.166667	0.02962	Under	0

[40]:
```
df['v1'] = df['S'] * df['K']
df['v2'] = df['S'] * df['tau']
df['v3'] = df['S'] * df['r']
df['v4'] = df['K'] * df['tau']
df['v5'] = df['K'] * df['r']
df['v6'] = df['r'] * df['tau']
```

[57]:
```
df['v1'] = df['S'] / df['K']
df['v2'] = df['S'] / df['tau']
df['v3'] = df['S'] / df['r']
df['v4'] = df['K'] / df['tau']
df['v5'] = df['K'] / df['r']
df['v6'] = df['r'] / df['tau']
```

[58]:
```
df.head()
```

[58]:

	Value	S	K	tau	r	BS	binary_label	\
0	21.670404	431.623898	420	0.341270	0.03013	Under	0	
1	0.125000	427.015526	465	0.166667	0.03126	Over	1	
2	20.691244	427.762336	415	0.265873	0.03116	Under	0	
3	1.035002	451.711658	460	0.063492	0.02972	Over	1	
4	39.553020	446.718974	410	0.166667	0.02962	Under	0	

	v1	v2	v3	v4	v5	v6
0	1.027676	1264.758401	14325.386601	1230.697675	13939.595088	0.088288
1	0.918313	2562.093150	13660.125589	2789.999994	14875.239923	0.187560
2	1.030753	1608.897145	13727.931207	1560.895522	13318.356868	0.117199
3	0.981982	7114.458665	15198.911770	7245.000056	15477.792732	0.468090
4	1.089558	2680.313841	15081.666928	2459.999995	13841.998650	0.177720

[59]:
```
X = df[['S','K','tau','r','v1','v2','v3','v4','v5','v6']].values
y = df['binary_label']
```

[60]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.25,
                                                    random_state = 0,
```

```
                                                    stratify = y)
```

[61]:
```
lg = LogisticRegression()
rf = RandomForestClassifier()
```

[62]:
```
# lr on original data
lg.fit(X_train, y_train)
lg.score(X_test, y_test)
```

[62]: 0.9095238095238095

[63]:
```
# lr on standardized data
from sklearn.preprocessing import StandardScaler
stdsc = StandardScaler()
X_train_std = stdsc.fit_transform(X_train)
X_test_std = stdsc.transform(X_test)

lg.fit(X_train_std, y_train)
lg.score(X_test_std, y_test)
```

[63]: 0.919047619047619

[64]:
```
# rf
rf.fit(X_train, y_train)
rf.score(X_test, y_test)
```

[64]: 0.9380952380952381

[ ]:
```

```