



یادگیری ماشین

تمرین شماره 3

Adaboost NC Algorithms

حسین توکلیان

شماره دانشجویی

9860571

خرداد 1399

دکتر ستار هاشمی

کتابخانه های مورد استفاده:

```
import pandas as pd
import numpy as np
import scipy.stats as ss
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_recall_fscore_support, roc_auc_score
from imblearn.metrics import geometric_mean_score
from imblearn.over_sampling import RandomOverSampler
```

تابع نویز

```
def add_noise(x, r_n, r_c):
    n_c = int(np.floor(r_c * x.shape[0]))
    dv = int(np.floor(x.shape[1] * r_n))
    f_inds = np.random.choice([i for i in range(x.shape[1])], size=dv, replace=False)
    x_c = x.copy()
    a = np.arange(2)
    xU, xL = a + 0.5, a - 0.5
    for f in f_inds:
        probb = ss.norm.cdf(xU, scale=3) - ss.norm.cdf(xL, scale=3)
        probb = probb / probb.sum()
        vals = np.random.choice(a, size=n_c, p=probb)
        s_inds = np.random.choice([i for i in range(x.shape[0])], size=n_c, replace=False)
        x_c[s_inds, f] = vals

    return x_c
```

خواندن فایل

```
def load_data(file_name):
    df = pd.read_csv(file_name)
    df = df.fillna(-1)
    x = df.iloc[:, :-1].values
    y = df.iloc[:, -1].values
    y[y==-1]=0
```

جایگذاری مقادیر خالی با مقدار 1-

```
df = df.fillna(-1)
```

جداسازی:

مقادیر ویژگیها

```
x = df.iloc[:, :-1].values
```

مقادیر کلاس ها

```
y = df.iloc[:, -1].values
```

کلاس 1- را به 0 تغییر دادم مقدارش را چون در کد نویسی اندیس 1- نداشتیم و کار راحت تر است.

```
y[y==1]=0
```

محاسبه ambiguity

```
def abs_ambiguity(trees, alpha, x, y):  
    prds = np.zeros((x.shape[0], len(trees)))
```

درخت، آلفا و x و y را دریافت میکند.

یک آرایه میسازد که قرار است در آن پیشبینی هریک از درختها برای هریک از داده ها ذخیره شود.

```
votes = np.zeros((x.shape[0], 2))
```

آرایه votes داریم که 2 تا ستون داره چو 2 تا کلاس داریم. برای هر کدام از داده ها وقتی که درخت رای میده کلاس 0 یا کلاس 1 $\alpha[i]$ (42) اون درخت اضافه میشه به ستون 1 یا ستون 2 برای اون داده اگر کلاسیفایر رای بده که اون داده از ستون صفر است به ستون اول اضافه میشه در غیر اینصورت به ستون دوم اضافه میشه.

```
for i, tree in enumerate(trees):  
    prds[:, i] = tree.predict(x)
```

در اینجا برای هرکدام از درختها پیشبینی بدست میاد

```
for j in range(x.shape[0]):  
    votes[j, int(prds[j, i])] += alpha[i]
```

و با توجه به اینکه پیشبینی 0 یا 1 بوده اضافه میشه به ستون مربوط به خودش

```
H = votes.argmax(axis=1)  
H_y = np.float64(H == y).reshape((-1, 1))
```

بعد از اون هر کسی که بیشترین رای رو آورده باشه با استفاده از ماکسیم گیری کلاسش تعیین میشه

بعد ملاحظه میکنیم که هر کدام از این پیشبینیها که ensemble برامون انجام داده آیا برابر هستند با اون پیشبینیهای که باید انجام میداده یا نه

```
h_y = np.float64(prds == np.reshape(y, (-1, 1)))
```

و برای هرکدام از کلاسیفایرها به صورت جداگانه همین کار انجام میدیم

```
amb = (H_y - h_y).mean(axis=1)  
return np.abs(amb)
```

طبق فرمول که تو مقاله ذکر شده مقدار amb حساب شده و مقدارش برگردانده میشود.

```
def train_adaboost(x, y, n_trees, lamb):  
    n_samples = x.shape[0]  
    d = np.ones((n_samples,)) / n_samples  
    p = np.ones((n_samples,))
```

تابع adaboost کاری که میکند یک ensemble adaboost را آموزش میدهد که x , y ورودیها و خروجیهای ما هستند و n_tree تعداد درختها و λ مقداری که داخل فایل وجود داشته

d وزن هر کدام از نمونه ها هستند

P پناالتی است که در ابتدا مقدارش را 1 قرار میدیم

```
tree_list = []
alpha = np.zeros((n_trees))
inds = [i for i in range(n_samples)]
```

لیست درختهای ما

Alpha مقادیر ضریب هر یک از درختها است

```
for t in range(n_trees):
    if t > 0:
        amb = abs_ambiguity(tree_list[t], alpha[:t], x, y)
        p = 1 - amb
```

در تکرار

اگر در تکرار اول نباشیم

کاری که انجام میدیم مقدار ambiguity را حساب میکنیم

مقدار پنالتی را حساب میکنیم و کار را جلو میبریم

اما اگر در تکرار اول قرار داشته باشیم یعنی مقدار $t=0$ باشد مقدار amb همان مقدار 1 خواهد بود (55) که قبلا گفته شده

```
sel_inds = np.random.choice(inds, size=n_samples, replace=True, p=d)
```

داده ها به صورت تصادفی با استفاده از توزیع p انتخاب میشوند.

```
tree = DecisionTreeClassifier()
tree.fit(x[sel_inds], y[sel_inds])
l = tree.predict(x)
match = (l == y).astype('float64')
```

یک سری کلاسیفایر با استفاده از داده ها آموزش داده میشود

مقدار خروجی L برای آنها اینجا (68) بدست میاد.

برای هر کدام از این داده ها کلاسیفایر جدیدی که آموزش داده شده درست پیشبینی کرده یا مقدار match نشون میده

```
alpha[t] = 0.5 * np.log((d * p ** lamb * match).sum() / ((d * p ** lamb * np.logical_not(match)).sum()))
tree_list.append(tree)
```

اونایی که درست پیشبینی کرده در صورت با هم جمع میشن و آنهایی که درست پیشبینی نکرده در مخرج با هم قرار میگیرند.

در نهایت درختی که درست شده به لیست درختها اضافه میشود.

```
d = p ** lamb * d * np.exp(-alpha[t] * match)
d /= d.sum()

return tree_list, alpha
```

مقدار d با استفاده از همان فرمول مقاله update شده

و نرمالسازی میشود.

و مقادیر بازگردانده میشود.

```
def predict_adaboost(trees, alpha, x):
```

تابع predict_adaboost خروجی را برابان پیشبینی میکنه مقادیر tree و alpha و x را دریافت میکند.

```
votes = np.zeros((x.shape[0], 2))
```

به همان سبک تابع ambiguity آرایه داریم که 2 ستون دارد.

```
for i, tree in enumerate(trees):
    prd = tree.predict(x)
```

هر کدام از داده هایی که برایش پیشبینی شد (83)

```
for j in range(x.shape[0]):
    votes[j, int(prd[j])] += alpha[i]
```

به ستون 0 یا 1 ، α با توجه به کلاسی که درخت پیشبینی کرده داده شده

```
label = votes.argmax(axis=1)
```

در نهایت با استفاده از ماکسیمم گیری مشخص میشود که متعلق به کدام کلاس است.

```
v_min = votes.min(axis=1).reshape((-1, 1))
v_max = votes.max(axis=1).reshape((-1, 1))
```

ضمن اینکه مقادیر node هایی هم که حساب شده بود بین بازه 0 و 1 انتقال داده میشود

```
votes = (votes - v_min) / (v_max - v_min)
p = votes / votes.sum(axis=1).reshape((-1, 1))
return label, p[:, 1]
```

نرمال سازی میشود

تا در نهایت ستون 1 (94) این node ها یعنی ستون مربوط به کلاس 1 احتمالاتش را به عنوان خروجی برگردانیم در کنار label هایی که پیشبینی شده بود.

```
def report_measures_for_adaboost(x, y, n_trees, trn_ratio, lamb, noise_dst='none', r_n=0, r_c=0):
    trn_x, tst_x, trn_y, tst_y = train_test_split(x, y, train_size = trn_ratio, stratify = y)
```

دریافت داده های ورودی تعداد درخت ها درصد داده آموزشی مقدار لاندا اینکه روی داده train یا test نویز باید اعمال شود و r_n, r_c

تقسیم داده به دو بخش train و test

```
if noise_dst == 'train':
    trn_x = add_noise(trn_x, r_n, r_c)
elif noise_dst == 'test':
    tst_x = add_noise(tst_x, r_n, r_c)
```

بر اساس اینکه نویز روی داده train یا test انجام شود اینکار انجام میشود و در نهایت مقادیر بدست می آید.

```
sampler = RandomOverSampler()
s_x, s_y = sampler.fit_resample(trn_x, trn_y)
```

برای افزایش دقت داده ها

```
prc, recall, fscore, auc, g_mean = (np.zeros(n_itr), np.zeros(n_itr), np.zeros(n_itr), np.zeros(n_itr),
np.zeros(n_itr))
for lamb in np.arange(0.1, 4, 0.5):
    print('lambda: {}'.format(lamb))
    for n_trees in [11, 21, 31, 41, 51]:
        for i in range(n_itr):
            prc[i], recall[i], fscore[i], auc[i], g_mean[i] = report_measures_for_adaboost(x, y, n_trees, trn_ratio,
lamb)

    print(('#Trees: {}, '
        'Precision Mean: {:.2f} STD: {:.2f}, '
        'Recall Mean: {:.2f}, STD: {:.2f}, '
        'F-Score Mean: {:.2f}, STD: {:.2f}'
        'AUC Mean: {:.2f}, STD: {:.2f}'
        'G-Mean: {:.2f}, STD: {:.2f}').format(
            n_trees,
            prc.mean(), prc.std(),
            recall.mean(), recall.std(),
            fscore.mean(), fscore.std(),
            auc.mean(), auc.std(),
            g_mean.mean(), g_mean.std()))
```

```
for noise_dst in ['train', 'test']:
    for r_n in np.arange(0.1, 0.51, 0.1):
        for r_c in np.arange(0.1, 0.51, 0.1):
            for lamb in np.arange(0.1, 4, 0.5):
                print('lambda: {}'.format(lamb))
                for n_trees in [11, 21, 31, 41, 51]:
                    for i in range(n_itr):
                        prc[i], recall[i], fscore[i], auc[i], g_mean[i] = report_measures_for_adaboost(x, y, n_trees, trn_ratio,
                        lamb, noise_dst, r_n, r_c)
                    print(['Noise Added to: {}, Rn: {}, Rc: {}, #Trees: {}, '
                        'Precision Mean: {:.2f} STD: {:.2f}, '
                        'Recall Mean: {:.2f}, STD: {:.2f}, '
                        'F-Score Mean: {:.2f}, STD: {:.2f} '
                        'AUC Mean: {:.2f}, STD: {:.2f} '
                        'G-Mean: {:.2f}, STD: {:.2f}').format(
                        noise_dst, r_n, r_c, n_trees,
                        prc.mean(), prc.std(),
                        recall.mean(), recall.std(),
                        fscore.mean(), fscore.std(),
                        auc.mean(), auc.std(),
                        g_mean.mean(), g_mean.std())])
```

lambda: 0.1

Trees: 11, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.42, STD: 0.10, F-Score Mean: 0.38, STD: #
0.07AUC Mean: 0.68, STD: 0.05G-Mean: 0.62, STD: 0.07

Trees: 21, Precision Mean: 0.40 STD: 0.06, Recall Mean: 0.44, STD: 0.07, F-Score Mean: 0.42, STD: #
0.05AUC Mean: 0.70, STD: 0.03G-Mean: 0.65, STD: 0.05

Trees: 31, Precision Mean: 0.39 STD: 0.06, Recall Mean: 0.42, STD: 0.08, F-Score Mean: 0.40, STD: #
0.06AUC Mean: 0.69, STD: 0.04G-Mean: 0.63, STD: 0.06

Trees: 41, Precision Mean: 0.39 STD: 0.03, Recall Mean: 0.40, STD: 0.08, F-Score Mean: 0.39, STD: #
0.05AUC Mean: 0.68, STD: 0.04G-Mean: 0.62, STD: 0.06

Trees: 51, Precision Mean: 0.38 STD: 0.07, Recall Mean: 0.37, STD: 0.08, F-Score Mean: 0.37, STD: #
0.07AUC Mean: 0.66, STD: 0.04G-Mean: 0.59, STD: 0.07

lambda: 0.6

Trees: 11, Precision Mean: 0.36 STD: 0.05, Recall Mean: 0.40, STD: 0.11, F-Score Mean: 0.38, STD: #
0.07AUC Mean: 0.68, STD: 0.05G-Mean: 0.61, STD: 0.08

Trees: 21, Precision Mean: 0.41 STD: 0.07, Recall Mean: 0.44, STD: 0.05, F-Score Mean: 0.42, STD: #
0.04AUC Mean: 0.70, STD: 0.02G-Mean: 0.65, STD: 0.04

Trees: 31, Precision Mean: 0.43 STD: 0.06, Recall Mean: 0.42, STD: 0.05, F-Score Mean: 0.42, STD: #
0.05AUC Mean: 0.69, STD: 0.03G-Mean: 0.63, STD: 0.04

Trees: 41, Precision Mean: 0.40 STD: 0.05, Recall Mean: 0.40, STD: 0.06, F-Score Mean: 0.40, STD: #
0.05AUC Mean: 0.68, STD: 0.03G-Mean: 0.61, STD: 0.05

Trees: 51, Precision Mean: 0.39 STD: 0.08, Recall Mean: 0.39, STD: 0.09, F-Score Mean: 0.38, STD: #
0.08AUC Mean: 0.67, STD: 0.04G-Mean: 0.60, STD: 0.07

lambda: 1.1

Trees: 11, Precision Mean: 0.35 STD: 0.06, Recall Mean: 0.35, STD: 0.05, F-Score Mean: 0.34, STD: #
0.04AUC Mean: 0.65, STD: 0.02G-Mean: 0.57, STD: 0.04

Trees: 21, Precision Mean: 0.39 STD: 0.08, Recall Mean: 0.44, STD: 0.10, F-Score Mean: 0.41, STD: #
0.08AUC Mean: 0.70, STD: 0.05G-Mean: 0.64, STD: 0.07

Trees: 31, Precision Mean: 0.42 STD: 0.08, Recall Mean: 0.43, STD: 0.08, F-Score Mean: 0.42, STD: #
0.06AUC Mean: 0.69, STD: 0.04G-Mean: 0.64, STD: 0.06

Trees: 41, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.36, STD: 0.10, F-Score Mean: 0.36, STD: #
0.07AUC Mean: 0.66, STD: 0.05G-Mean: 0.58, STD: 0.08

Trees: 51, Precision Mean: 0.39 STD: 0.07, Recall Mean: 0.37, STD: 0.08, F-Score Mean: 0.38, STD: #
0.07AUC Mean: 0.67, STD: 0.04G-Mean: 0.59, STD: 0.06

lambda: 1.6

Trees: 11, Precision Mean: 0.37 STD: 0.03, Recall Mean: 0.42, STD: 0.07, F-Score Mean: 0.39, STD: #
0.04AUC Mean: 0.69, STD: 0.03G-Mean: 0.63, STD: 0.05

Trees: 21, Precision Mean: 0.38 STD: 0.07, Recall Mean: 0.41, STD: 0.09, F-Score Mean: 0.40, STD: #
0.08AUC Mean: 0.68, STD: 0.05G-Mean: 0.62, STD: 0.07

Trees: 31, Precision Mean: 0.39 STD: 0.08, Recall Mean: 0.41, STD: 0.08, F-Score Mean: 0.40, STD: #
0.07AUC Mean: 0.68, STD: 0.04G-Mean: 0.63, STD: 0.06

Trees: 41, Precision Mean: 0.42 STD: 0.07, Recall Mean: 0.41, STD: 0.06, F-Score Mean: 0.41, STD: #
0.05AUC Mean: 0.68, STD: 0.03G-Mean: 0.62, STD: 0.04

Trees: 51, Precision Mean: 0.41 STD: 0.06, Recall Mean: 0.41, STD: 0.08, F-Score Mean: 0.41, STD: #
0.06AUC Mean: 0.68, STD: 0.04G-Mean: 0.62, STD: 0.06

lambda: 2.1

Trees: 11, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.41, STD: 0.10, F-Score Mean: 0.37, STD: #
0.08AUC Mean: 0.68, STD: 0.05G-Mean: 0.62, STD: 0.08

Trees: 21, Precision Mean: 0.39 STD: 0.09, Recall Mean: 0.42, STD: 0.09, F-Score Mean: 0.40, STD: #
0.07AUC Mean: 0.69, STD: 0.04G-Mean: 0.63, STD: 0.06

Trees: 31, Precision Mean: 0.38 STD: 0.04, Recall Mean: 0.41, STD: 0.08, F-Score Mean: 0.39, STD: #
0.05AUC Mean: 0.68, STD: 0.04G-Mean: 0.62, STD: 0.06

Trees: 41, Precision Mean: 0.37 STD: 0.04, Recall Mean: 0.42, STD: 0.09, F-Score Mean: 0.39, STD: #
0.05AUC Mean: 0.69, STD: 0.04G-Mean: 0.63, STD: 0.06

Trees: 51, Precision Mean: 0.42 STD: 0.05, Recall Mean: 0.43, STD: 0.07, F-Score Mean: 0.42, STD: #
0.03AUC Mean: 0.69, STD: 0.03G-Mean: 0.64, STD: 0.05

lambda: 2.6

Trees: 11, Precision Mean: 0.38 STD: 0.06, Recall Mean: 0.43, STD: 0.05, F-Score Mean: 0.40, STD: #
0.05AUC Mean: 0.69, STD: 0.03G-Mean: 0.64, STD: 0.04

Trees: 21, Precision Mean: 0.40 STD: 0.09, Recall Mean: 0.43, STD: 0.08, F-Score Mean: 0.41, STD: #
0.08AUC Mean: 0.69, STD: 0.04G-Mean: 0.64, STD: 0.07

Trees: 31, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.43, STD: 0.10, F-Score Mean: 0.39, STD: #
0.07AUC Mean: 0.69, STD: 0.05G-Mean: 0.63, STD: 0.08

Trees: 41, Precision Mean: 0.39 STD: 0.06, Recall Mean: 0.39, STD: 0.09, F-Score Mean: 0.38, STD: #
0.07AUC Mean: 0.67, STD: 0.05G-Mean: 0.60, STD: 0.07

Trees: 51, Precision Mean: 0.34 STD: 0.04, Recall Mean: 0.39, STD: 0.05, F-Score Mean: 0.36, STD: #
0.04AUC Mean: 0.67, STD: 0.02G-Mean: 0.61, STD: 0.04

lambda: 3.1

Trees: 11, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.46, STD: 0.11, F-Score Mean: 0.39, STD: #
0.08 AUC Mean: 0.70, STD: 0.05 G-Mean: 0.65, STD: 0.08

Trees: 21, Precision Mean: 0.45 STD: 0.08, Recall Mean: 0.49, STD: 0.09, F-Score Mean: 0.47, STD: #
0.08 AUC Mean: 0.72, STD: 0.05 G-Mean: 0.68, STD: 0.07

Trees: 31, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.45, STD: 0.08, F-Score Mean: 0.39, STD: #
0.06 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.65, STD: 0.06

Trees: 41, Precision Mean: 0.34 STD: 0.05, Recall Mean: 0.40, STD: 0.08, F-Score Mean: 0.37, STD: #
0.06 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.62, STD: 0.06

Trees: 51, Precision Mean: 0.37 STD: 0.07, Recall Mean: 0.45, STD: 0.10, F-Score Mean: 0.40, STD: #
0.06 AUC Mean: 0.70, STD: 0.05 G-Mean: 0.65, STD: 0.07

lambda: 3.6

Trees: 11, Precision Mean: 0.40 STD: 0.06, Recall Mean: 0.51, STD: 0.05, F-Score Mean: 0.44, STD: #
0.06 AUC Mean: 0.73, STD: 0.03 G-Mean: 0.69, STD: 0.04

Trees: 21, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.43, STD: 0.06, F-Score Mean: 0.39, STD: #
0.05 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.64, STD: 0.04

Trees: 31, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.42, STD: 0.07, F-Score Mean: 0.39, STD: #
0.05 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.63, STD: 0.05

Trees: 41, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.44, STD: 0.08, F-Score Mean: 0.40, STD: #
0.05 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.64, STD: 0.06

Trees: 51, Precision Mean: 0.38 STD: 0.06, Recall Mean: 0.50, STD: 0.08, F-Score Mean: 0.43, STD: #
0.05 AUC Mean: 0.72, STD: 0.04 G-Mean: 0.69, STD: 0.05

lambda: 0.1

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 11, Precision Mean: 0.38 STD: 0.07, Recall Mean: 0.40,
STD: 0.07, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.62, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 21, Precision Mean: 0.36 STD: 0.07, Recall Mean: 0.41,
STD: 0.10, F-Score Mean: 0.38, STD: 0.08 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.62, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 31, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.37,
STD: 0.05, F-Score Mean: 0.37, STD: 0.05 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.60, STD: 0.04

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 41, Precision Mean: 0.36 STD: 0.07, Recall Mean: 0.39,
STD: 0.11, F-Score Mean: 0.37, STD: 0.08 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.60, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 51, Precision Mean: 0.35 STD: 0.04, Recall Mean: 0.38,
STD: 0.08, F-Score Mean: 0.36, STD: 0.05 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.60, STD: 0.06

lambda: 0.6

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 11, Precision Mean: 0.35 STD: 0.04, Recall Mean: 0.40, STD: 0.09, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.61, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 21, Precision Mean: 0.39 STD: 0.08, Recall Mean: 0.41, STD: 0.07, F-Score Mean: 0.40, STD: 0.07 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.63, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 31, Precision Mean: 0.39 STD: 0.05, Recall Mean: 0.38, STD: 0.07, F-Score Mean: 0.38, STD: 0.05 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.60, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 41, Precision Mean: 0.45 STD: 0.07, Recall Mean: 0.37, STD: 0.07, F-Score Mean: 0.40, STD: 0.06 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.59, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 51, Precision Mean: 0.42 STD: 0.04, Recall Mean: 0.41, STD: 0.06, F-Score Mean: 0.41, STD: 0.04 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.63, STD: 0.05

lambda: 1.1

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 11, Precision Mean: 0.41 STD: 0.05, Recall Mean: 0.42, STD: 0.06, F-Score Mean: 0.41, STD: 0.04 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.64, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 21, Precision Mean: 0.39 STD: 0.06, Recall Mean: 0.44, STD: 0.06, F-Score Mean: 0.41, STD: 0.05 AUC Mean: 0.70, STD: 0.03 G-Mean: 0.64, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 31, Precision Mean: 0.39 STD: 0.07, Recall Mean: 0.38, STD: 0.07, F-Score Mean: 0.38, STD: 0.07 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.60, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 41, Precision Mean: 0.34 STD: 0.08, Recall Mean: 0.32, STD: 0.12, F-Score Mean: 0.33, STD: 0.10 AUC Mean: 0.64, STD: 0.06 G-Mean: 0.55, STD: 0.10

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 51, Precision Mean: 0.40 STD: 0.09, Recall Mean: 0.39, STD: 0.10, F-Score Mean: 0.39, STD: 0.09 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.61, STD: 0.08

lambda: 1.6

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 11, Precision Mean: 0.36 STD: 0.07, Recall Mean: 0.40, STD: 0.06, F-Score Mean: 0.38, STD: 0.06 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.62, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 21, Precision Mean: 0.34 STD: 0.05, Recall Mean: 0.41, STD: 0.10, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.62, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 31, Precision Mean: 0.40 STD: 0.05, Recall Mean: 0.37, STD: 0.06, F-Score Mean: 0.38, STD: 0.05 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.60, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 41, Precision Mean: 0.39 STD: 0.08, Recall Mean: 0.41, STD: 0.08, F-Score Mean: 0.39, STD: 0.08 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.62, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 51, Precision Mean: 0.41 STD: 0.08, Recall Mean: 0.38, STD: 0.10, F-Score Mean: 0.39, STD: 0.09 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.60, STD: 0.09

lambda: 2.1

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 11, Precision Mean: 0.39 STD: 0.07, Recall Mean: 0.42, STD: 0.09, F-Score Mean: 0.40, STD: 0.07 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.63, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 21, Precision Mean: 0.39 STD: 0.07, Recall Mean: 0.43, STD: 0.10, F-Score Mean: 0.40, STD: 0.07 AUC Mean: 0.69, STD: 0.05 G-Mean: 0.63, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 31, Precision Mean: 0.37 STD: 0.07, Recall Mean: 0.40, STD: 0.05, F-Score Mean: 0.38, STD: 0.05 AUC Mean: 0.68, STD: 0.02 G-Mean: 0.62, STD: 0.04

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 41, Precision Mean: 0.36 STD: 0.04, Recall Mean: 0.38, STD: 0.07, F-Score Mean: 0.37, STD: 0.05 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.60, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 51, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.44, STD: 0.08, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.64, STD: 0.06

lambda: 2.6

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 11, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.40, STD: 0.06, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.61, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 21, Precision Mean: 0.35 STD: 0.08, Recall Mean: 0.40, STD: 0.11, F-Score Mean: 0.37, STD: 0.08 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.61, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 31, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.42, STD: 0.09, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.63, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 41, Precision Mean: 0.39 STD: 0.05, Recall Mean: 0.44, STD: 0.07, F-Score Mean: 0.41, STD: 0.05 AUC Mean: 0.70, STD: 0.03 G-Mean: 0.64, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 51, Precision Mean: 0.32 STD: 0.06, Recall Mean: 0.37, STD: 0.10, F-Score Mean: 0.34, STD: 0.07 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.58, STD: 0.08

lambda: 3.1

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 11, Precision Mean: 0.38 STD: 0.06, Recall Mean: 0.45, STD: 0.06, F-Score Mean: 0.41, STD: 0.06 AUC Mean: 0.70, STD: 0.03 G-Mean: 0.65, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 21, Precision Mean: 0.38 STD: 0.06, Recall Mean: 0.46, STD: 0.11, F-Score Mean: 0.41, STD: 0.07 AUC Mean: 0.70, STD: 0.05 G-Mean: 0.65, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 31, Precision Mean: 0.34 STD: 0.06, Recall Mean: 0.44, STD: 0.08, F-Score Mean: 0.38, STD: 0.06 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.64, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 41, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.43, STD: 0.10, F-Score Mean: 0.38, STD: 0.06 AUC Mean: 0.69, STD: 0.05 G-Mean: 0.63, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 51, Precision Mean: 0.36 STD: 0.04, Recall Mean: 0.49, STD: 0.11, F-Score Mean: 0.41, STD: 0.06 AUC Mean: 0.72, STD: 0.05 G-Mean: 0.68, STD: 0.08

lambda: 3.6

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 11, Precision Mean: 0.34 STD: 0.07, Recall Mean: 0.40, STD: 0.07, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.61, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 21, Precision Mean: 0.34 STD: 0.06, Recall Mean: 0.42, STD: 0.08, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.63, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 31, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.45, STD: 0.07, F-Score Mean: 0.40, STD: 0.06 AUC Mean: 0.70, STD: 0.03 G-Mean: 0.65, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 41, Precision Mean: 0.35 STD: 0.04, Recall Mean: 0.44, STD: 0.07, F-Score Mean: 0.39, STD: 0.04 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.64, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 51, Precision Mean: 0.34 STD: 0.06, Recall Mean: 0.44, STD: 0.12, F-Score Mean: 0.38, STD: 0.08 AUC Mean: 0.69, STD: 0.06 G-Mean: 0.64, STD: 0.09

lambda: 0.1

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 11, Precision Mean: 0.35 STD: 0.09, Recall Mean: 0.37, STD: 0.09, F-Score Mean: 0.36, STD: 0.08 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.59, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 21, Precision Mean: 0.36 STD: 0.08, Recall Mean: 0.35, STD: 0.11, F-Score Mean: 0.35, STD: 0.09 AUC Mean: 0.65, STD: 0.05 G-Mean: 0.57, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 31, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.39, STD: 0.10, F-Score Mean: 0.38, STD: 0.07 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.61, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 41, Precision Mean: 0.40 STD: 0.09, Recall Mean: 0.40, STD: 0.06, F-Score Mean: 0.40, STD: 0.07 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.62, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 51, Precision Mean: 0.41 STD: 0.09, Recall Mean: 0.36, STD: 0.06, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.58, STD: 0.05

lambda: 0.6

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 11, Precision Mean: 0.39 STD: 0.07, Recall Mean: 0.46, STD: 0.09, F-Score Mean: 0.41, STD: 0.07 AUC Mean: 0.70, STD: 0.05 G-Mean: 0.66, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 21, Precision Mean: 0.34 STD: 0.10, Recall Mean: 0.35, STD: 0.12, F-Score Mean: 0.34, STD: 0.11 AUC Mean: 0.65, STD: 0.06 G-Mean: 0.57, STD: 0.11

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 31, Precision Mean: 0.38 STD: 0.04, Recall Mean: 0.41, STD: 0.08, F-Score Mean: 0.39, STD: 0.05 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.63, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 41, Precision Mean: 0.38 STD: 0.06, Recall Mean: 0.36, STD: 0.09, F-Score Mean: 0.37, STD: 0.07 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.58, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 51, Precision Mean: 0.37 STD: 0.07, Recall Mean: 0.31, STD: 0.09, F-Score Mean: 0.33, STD: 0.08 AUC Mean: 0.64, STD: 0.04 G-Mean: 0.54, STD: 0.09

lambda: 1.1

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 11, Precision Mean: 0.38 STD: 0.06, Recall Mean: 0.38, STD: 0.06, F-Score Mean: 0.38, STD: 0.06 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.60, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 21, Precision Mean: 0.34 STD: 0.05, Recall Mean: 0.40, STD: 0.09, F-Score Mean: 0.36, STD: 0.07 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.61, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 31, Precision Mean: 0.36 STD: 0.08, Recall Mean: 0.36, STD: 0.08, F-Score Mean: 0.36, STD: 0.07 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.59, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 41, Precision Mean: 0.38 STD: 0.05, Recall Mean: 0.41, STD: 0.08, F-Score Mean: 0.39, STD: 0.05 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.63, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 51, Precision Mean: 0.42 STD: 0.07, Recall Mean: 0.40, STD: 0.08, F-Score Mean: 0.40, STD: 0.06 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.61, STD: 0.06

lambda: 1.6

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 11, Precision Mean: 0.38 STD: 0.05, Recall Mean: 0.42, STD: 0.06, F-Score Mean: 0.40, STD: 0.05 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.63, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 21, Precision Mean: 0.41 STD: 0.10, Recall Mean: 0.40, STD: 0.10, F-Score Mean: 0.40, STD: 0.09 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.62, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 31, Precision Mean: 0.40 STD: 0.10, Recall Mean: 0.41, STD: 0.12, F-Score Mean: 0.40, STD: 0.10 AUC Mean: 0.68, STD: 0.06 G-Mean: 0.62, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 41, Precision Mean: 0.40 STD: 0.10, Recall Mean: 0.36, STD: 0.08, F-Score Mean: 0.37, STD: 0.07 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.58, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 51, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.36, STD: 0.06, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.59, STD: 0.05

lambda: 2.1

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 11, Precision Mean: 0.37 STD: 0.08, Recall Mean: 0.39, STD: 0.10, F-Score Mean: 0.38, STD: 0.08 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.61, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 21, Precision Mean: 0.40 STD: 0.08, Recall Mean: 0.43, STD: 0.07, F-Score Mean: 0.41, STD: 0.07 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.64, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 31, Precision Mean: 0.40 STD: 0.08, Recall Mean: 0.41, STD: 0.09, F-Score Mean: 0.40, STD: 0.08 AUC Mean: 0.69, STD: 0.05 G-Mean: 0.63, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 41, Precision Mean: 0.39 STD: 0.08, Recall Mean: 0.43, STD: 0.12, F-Score Mean: 0.40, STD: 0.09 AUC Mean: 0.69, STD: 0.06 G-Mean: 0.64, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 51, Precision Mean: 0.39 STD: 0.07, Recall Mean: 0.43, STD: 0.06, F-Score Mean: 0.40, STD: 0.06 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.64, STD: 0.05

lambda: 2.6

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 11, Precision Mean: 0.34 STD: 0.04, Recall Mean: 0.42, STD: 0.07, F-Score Mean: 0.38, STD: 0.04 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.63, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 21, Precision Mean: 0.35 STD: 0.08, Recall Mean: 0.39, STD: 0.09, F-Score Mean: 0.37, STD: 0.08 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.61, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 31, Precision Mean: 0.37 STD: 0.05, Recall Mean: 0.42, STD: 0.08, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.63, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 41, Precision Mean: 0.41 STD: 0.07, Recall Mean: 0.47, STD: 0.07, F-Score Mean: 0.44, STD: 0.06 AUC Mean: 0.71, STD: 0.03 G-Mean: 0.67, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 51, Precision Mean: 0.35 STD: 0.05, Recall Mean: 0.41, STD: 0.07, F-Score Mean: 0.37, STD: 0.05 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.62, STD: 0.05

lambda: 3.1

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 11, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.41, STD: 0.07, F-Score Mean: 0.38, STD: 0.05 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.62, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 21, Precision Mean: 0.41 STD: 0.06, Recall Mean: 0.49, STD: 0.11, F-Score Mean: 0.44, STD: 0.07 AUC Mean: 0.72, STD: 0.05 G-Mean: 0.68, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 31, Precision Mean: 0.35 STD: 0.04, Recall Mean: 0.39, STD: 0.08, F-Score Mean: 0.37, STD: 0.05 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.61, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 41, Precision Mean: 0.37 STD: 0.08, Recall Mean: 0.49, STD: 0.10, F-Score Mean: 0.41, STD: 0.07 AUC Mean: 0.71, STD: 0.05 G-Mean: 0.67, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 51, Precision Mean: 0.38 STD: 0.04, Recall Mean: 0.47, STD: 0.07, F-Score Mean: 0.41, STD: 0.03 AUC Mean: 0.71, STD: 0.03 G-Mean: 0.66, STD: 0.04

lambda: 3.6

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 11, Precision Mean: 0.34 STD: 0.07, Recall Mean: 0.41, STD: 0.09, F-Score Mean: 0.37, STD: 0.07 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.62, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 21, Precision Mean: 0.33 STD: 0.03, Recall Mean: 0.42, STD: 0.07, F-Score Mean: 0.37, STD: 0.03 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.62, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 31, Precision Mean: 0.38 STD: 0.05, Recall Mean: 0.48, STD: 0.07, F-Score Mean: 0.42, STD: 0.05 AUC Mean: 0.71, STD: 0.03 G-Mean: 0.67, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 41, Precision Mean: 0.32 STD: 0.04, Recall Mean: 0.45, STD: 0.11, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.69, STD: 0.05 G-Mean: 0.64, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 51, Precision Mean: 0.36 STD: 0.04, Recall Mean: 0.51, STD: 0.09, F-Score Mean: 0.42, STD: 0.05 AUC Mean: 0.73, STD: 0.04 G-Mean: 0.69, STD: 0.06

lambda: 0.1

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.42 STD: 0.09, Recall Mean: 0.37, STD: 0.07, F-Score Mean: 0.39, STD: 0.07 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.59, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.39 STD: 0.05, Recall Mean: 0.37, STD: 0.05, F-Score Mean: 0.37, STD: 0.03 AUC Mean: 0.66, STD: 0.02 G-Mean: 0.59, STD: 0.04

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.39, STD: 0.08, F-Score Mean: 0.36, STD: 0.06 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.61, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.40 STD: 0.06, Recall Mean: 0.40, STD: 0.06, F-Score Mean: 0.39, STD: 0.04 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.62, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.44 STD: 0.06, Recall Mean: 0.41, STD: 0.09, F-Score Mean: 0.42, STD: 0.06 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.62, STD: 0.07

lambda: 0.6

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.39 STD: 0.06, Recall Mean: 0.41, STD: 0.10, F-Score Mean: 0.40, STD: 0.08 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.62, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.37 STD: 0.05, Recall Mean: 0.38, STD: 0.11, F-Score Mean: 0.37, STD: 0.08 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.59, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.35 STD: 0.10, Recall Mean: 0.34, STD: 0.11, F-Score Mean: 0.34, STD: 0.10 AUC Mean: 0.65, STD: 0.05 G-Mean: 0.56, STD: 0.11

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.34 STD: 0.05, Recall Mean: 0.36, STD: 0.11, F-Score Mean: 0.35, STD: 0.08 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.58, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.42 STD: 0.08, Recall Mean: 0.40, STD: 0.14, F-Score Mean: 0.39, STD: 0.09 AUC Mean: 0.68, STD: 0.06 G-Mean: 0.61, STD: 0.11

lambda: 1.1

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.37 STD: 0.08, Recall Mean: 0.35, STD: 0.06, F-Score Mean: 0.36, STD: 0.07 AUC Mean: 0.65, STD: 0.03 G-Mean: 0.57, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.40 STD: 0.07, Recall Mean: 0.42, STD: 0.06, F-Score Mean: 0.40, STD: 0.05 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.63, STD: 0.04

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.40 STD: 0.07, Recall Mean: 0.38, STD: 0.07, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.60, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.41 STD: 0.06, Recall Mean: 0.37, STD: 0.08, F-Score Mean: 0.39, STD: 0.07 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.60, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.38 STD: 0.07, Recall Mean: 0.37, STD: 0.05, F-Score Mean: 0.37, STD: 0.05 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.60, STD: 0.04

lambda: 1.6

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.37 STD: 0.05, Recall Mean: 0.43, STD: 0.07, F-Score Mean: 0.40, STD: 0.04 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.64, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.37 STD: 0.07, Recall Mean: 0.38, STD: 0.08, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.60, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.40 STD: 0.07, Recall Mean: 0.39, STD: 0.09, F-Score Mean: 0.39, STD: 0.07 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.61, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.34 STD: 0.07, Recall Mean: 0.32, STD: 0.11, F-Score Mean: 0.32, STD: 0.09 AUC Mean: 0.64, STD: 0.05 G-Mean: 0.54, STD: 0.10

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.39 STD: 0.09, Recall Mean: 0.36, STD: 0.08, F-Score Mean: 0.37, STD: 0.05 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.59, STD: 0.06

lambda: 2.1

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.38 STD: 0.09, Recall Mean: 0.40, STD: 0.11, F-Score Mean: 0.39, STD: 0.09 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.61, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.36 STD: 0.08, Recall Mean: 0.39, STD: 0.11, F-Score Mean: 0.37, STD: 0.08 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.61, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.38 STD: 0.06, Recall Mean: 0.38, STD: 0.04, F-Score Mean: 0.38, STD: 0.04 AUC Mean: 0.67, STD: 0.02 G-Mean: 0.60, STD: 0.03

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.39, STD: 0.04, F-Score Mean: 0.38, STD: 0.04 AUC Mean: 0.67, STD: 0.02 G-Mean: 0.61, STD: 0.03

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.38 STD: 0.06, Recall Mean: 0.41, STD: 0.09, F-Score Mean: 0.39, STD: 0.07 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.62, STD: 0.08

lambda: 2.6

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.40 STD: 0.09, Recall Mean: 0.41, STD: 0.08, F-Score Mean: 0.40, STD: 0.07 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.63, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.35 STD: 0.08, Recall Mean: 0.36, STD: 0.10, F-Score Mean: 0.35, STD: 0.07 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.58, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.42 STD: 0.07, Recall Mean: 0.52, STD: 0.08, F-Score Mean: 0.46, STD: 0.04 AUC Mean: 0.74, STD: 0.03 G-Mean: 0.70, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.33 STD: 0.04, Recall Mean: 0.40, STD: 0.09, F-Score Mean: 0.36, STD: 0.05 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.61, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.42 STD: 0.10, Recall Mean: 0.45, STD: 0.06, F-Score Mean: 0.42, STD: 0.03 AUC Mean: 0.70, STD: 0.03 G-Mean: 0.65, STD: 0.04

lambda: 3.1

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.34 STD: 0.06, Recall Mean: 0.46, STD: 0.10, F-Score Mean: 0.39, STD: 0.07 AUC Mean: 0.70, STD: 0.05 G-Mean: 0.65, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.43, STD: 0.13, F-Score Mean: 0.39, STD: 0.07 AUC Mean: 0.69, STD: 0.06 G-Mean: 0.63, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.32 STD: 0.05, Recall Mean: 0.43, STD: 0.13, F-Score Mean: 0.36, STD: 0.07 AUC Mean: 0.68, STD: 0.06 G-Mean: 0.63, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.44, STD: 0.07, F-Score Mean: 0.40, STD: 0.05 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.65, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.39 STD: 0.08, Recall Mean: 0.47, STD: 0.12, F-Score Mean: 0.42, STD: 0.09 AUC Mean: 0.71, STD: 0.06 G-Mean: 0.66, STD: 0.09

lambda: 3.6

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.37 STD: 0.05, Recall Mean: 0.50, STD: 0.07, F-Score Mean: 0.42, STD: 0.05 AUC Mean: 0.72, STD: 0.03 G-Mean: 0.68, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.36 STD: 0.05, Recall Mean: 0.42, STD: 0.12, F-Score Mean: 0.38, STD: 0.06 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.62, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.34 STD: 0.03, Recall Mean: 0.46, STD: 0.11, F-Score Mean: 0.39, STD: 0.05 AUC Mean: 0.70, STD: 0.05 G-Mean: 0.65, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.34 STD: 0.04, Recall Mean: 0.45, STD: 0.06, F-Score Mean: 0.39, STD: 0.04 AUC Mean: 0.70, STD: 0.03 G-Mean: 0.65, STD: 0.04

Noise Added to: train, Rn: 0.1, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.36 STD: 0.07, Recall Mean: 0.47, STD: 0.11, F-Score Mean: 0.41, STD: 0.08 AUC Mean: 0.71, STD: 0.06 G-Mean: 0.66, STD: 0.08

lambda: 0.1

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 11, Precision Mean: 0.34 STD: 0.06, Recall Mean: 0.36, STD: 0.07, F-Score Mean: 0.35, STD: 0.05 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.59, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 21, Precision Mean: 0.39 STD: 0.05, Recall Mean: 0.37, STD: 0.09, F-Score Mean: 0.37, STD: 0.05 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.59, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 31, Precision Mean: 0.41 STD: 0.06, Recall Mean: 0.37, STD: 0.08, F-Score Mean: 0.39, STD: 0.07 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.60, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 41, Precision Mean: 0.39 STD: 0.09, Recall Mean: 0.32, STD: 0.07, F-Score Mean: 0.35, STD: 0.07 AUC Mean: 0.64, STD: 0.04 G-Mean: 0.55, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 51, Precision Mean: 0.39 STD: 0.07, Recall Mean: 0.35, STD: 0.09, F-Score Mean: 0.36, STD: 0.07 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.58, STD: 0.07

lambda: 0.6

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 11, Precision Mean: 0.38 STD: 0.12, Recall Mean: 0.37, STD: 0.07, F-Score Mean: 0.36, STD: 0.07 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.59, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 21, Precision Mean: 0.40 STD: 0.04, Recall Mean: 0.41, STD: 0.06, F-Score Mean: 0.40, STD: 0.04 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.63, STD: 0.04

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 31, Precision Mean: 0.42 STD: 0.07, Recall Mean: 0.36, STD: 0.07, F-Score Mean: 0.38, STD: 0.06 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.58, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 41, Precision Mean: 0.35 STD: 0.09, Recall Mean: 0.31, STD: 0.10, F-Score Mean: 0.32, STD: 0.09 AUC Mean: 0.63, STD: 0.05 G-Mean: 0.53, STD: 0.11

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 51, Precision Mean: 0.36 STD: 0.11, Recall Mean: 0.28, STD: 0.10, F-Score Mean: 0.31, STD: 0.10 AUC Mean: 0.63, STD: 0.05 G-Mean: 0.51, STD: 0.12

lambda: 1.1

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 11, Precision Mean: 0.34 STD: 0.07, Recall Mean: 0.35, STD: 0.08, F-Score Mean: 0.34, STD: 0.07 AUC Mean: 0.65, STD: 0.04 G-Mean: 0.57, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 21, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.36, STD: 0.09, F-Score Mean: 0.35, STD: 0.07 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.58, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 31, Precision Mean: 0.35 STD: 0.09, Recall Mean: 0.35, STD: 0.10, F-Score Mean: 0.35, STD: 0.09 AUC Mean: 0.65, STD: 0.05 G-Mean: 0.57, STD: 0.10

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 41, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.30, STD: 0.10, F-Score Mean: 0.32, STD: 0.07 AUC Mean: 0.63, STD: 0.05 G-Mean: 0.53, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 51, Precision Mean: 0.34 STD: 0.08, Recall Mean: 0.34, STD: 0.11, F-Score Mean: 0.34, STD: 0.09 AUC Mean: 0.65, STD: 0.05 G-Mean: 0.56, STD: 0.09

lambda: 1.6

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 11, Precision Mean: 0.34 STD: 0.07, Recall Mean: 0.36, STD: 0.11, F-Score Mean: 0.35, STD: 0.09 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.58, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 21, Precision Mean: 0.38 STD: 0.07, Recall Mean: 0.40, STD: 0.09, F-Score Mean: 0.38, STD: 0.07 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.62, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 31, Precision Mean: 0.41 STD: 0.09, Recall Mean: 0.37, STD: 0.06, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.60, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 41, Precision Mean: 0.37 STD: 0.08, Recall Mean: 0.34, STD: 0.06, F-Score Mean: 0.35, STD: 0.07 AUC Mean: 0.65, STD: 0.03 G-Mean: 0.57, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 51, Precision Mean: 0.38 STD: 0.08, Recall Mean: 0.34, STD: 0.07, F-Score Mean: 0.36, STD: 0.06 AUC Mean: 0.65, STD: 0.03 G-Mean: 0.57, STD: 0.05

lambda: 2.1

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 11, Precision Mean: 0.36 STD: 0.10, Recall Mean: 0.36, STD: 0.11, F-Score Mean: 0.36, STD: 0.09 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.58, STD: 0.10

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 21, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.39, STD: 0.07, F-Score Mean: 0.38, STD: 0.06 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.61, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 31, Precision Mean: 0.40 STD: 0.08, Recall Mean: 0.43, STD: 0.11, F-Score Mean: 0.40, STD: 0.08 AUC Mean: 0.69, STD: 0.05 G-Mean: 0.63, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 41, Precision Mean: 0.36 STD: 0.09, Recall Mean: 0.38, STD: 0.11, F-Score Mean: 0.36, STD: 0.08 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.59, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 51, Precision Mean: 0.35 STD: 0.06, Recall Mean: 0.36, STD: 0.07, F-Score Mean: 0.35, STD: 0.06 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.58, STD: 0.06

lambda: 2.6

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 11, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.39, STD: 0.09, F-Score Mean: 0.37, STD: 0.07 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.60, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 21, Precision Mean: 0.35 STD: 0.04, Recall Mean: 0.41, STD: 0.10, F-Score Mean: 0.37, STD: 0.04 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.62, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 31, Precision Mean: 0.41 STD: 0.07, Recall Mean: 0.42, STD: 0.12, F-Score Mean: 0.41, STD: 0.09 AUC Mean: 0.69, STD: 0.06 G-Mean: 0.63, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 41, Precision Mean: 0.41 STD: 0.08, Recall Mean: 0.42, STD: 0.07, F-Score Mean: 0.41, STD: 0.06 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.63, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 51, Precision Mean: 0.36 STD: 0.05, Recall Mean: 0.45, STD: 0.07, F-Score Mean: 0.40, STD: 0.05 AUC Mean: 0.70, STD: 0.03 G-Mean: 0.65, STD: 0.05

lambda: 3.1

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 11, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.41, STD: 0.12, F-Score Mean: 0.37, STD: 0.08 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.61, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 21, Precision Mean: 0.34 STD: 0.08, Recall Mean: 0.38, STD: 0.11, F-Score Mean: 0.35, STD: 0.08 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.59, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 31, Precision Mean: 0.34 STD: 0.07, Recall Mean: 0.36, STD: 0.10, F-Score Mean: 0.35, STD: 0.08 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.58, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 41, Precision Mean: 0.35 STD: 0.04, Recall Mean: 0.46, STD: 0.09, F-Score Mean: 0.39, STD: 0.05 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.66, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 51, Precision Mean: 0.36 STD: 0.05, Recall Mean: 0.50, STD: 0.10, F-Score Mean: 0.42, STD: 0.05 AUC Mean: 0.72, STD: 0.04 G-Mean: 0.68, STD: 0.06

lambda: 3.6

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 11, Precision Mean: 0.37 STD: 0.09, Recall Mean: 0.46, STD: 0.08, F-Score Mean: 0.41, STD: 0.08 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.66, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 21, Precision Mean: 0.37 STD: 0.07, Recall Mean: 0.47, STD: 0.09, F-Score Mean: 0.41, STD: 0.06 AUC Mean: 0.71, STD: 0.04 G-Mean: 0.67, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 31, Precision Mean: 0.36 STD: 0.05, Recall Mean: 0.43, STD: 0.07, F-Score Mean: 0.39, STD: 0.05 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.64, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 41, Precision Mean: 0.37 STD: 0.05, Recall Mean: 0.46, STD: 0.07, F-Score Mean: 0.40, STD: 0.04 AUC Mean: 0.70, STD: 0.03 G-Mean: 0.66, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 51, Precision Mean: 0.36 STD: 0.07, Recall Mean: 0.52, STD: 0.12, F-Score Mean: 0.42, STD: 0.07 AUC Mean: 0.73, STD: 0.06 G-Mean: 0.69, STD: 0.08

lambda: 0.1

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 11, Precision Mean: 0.43 STD: 0.07, Recall Mean: 0.37, STD: 0.09, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.59, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 21, Precision Mean: 0.38 STD: 0.07, Recall Mean: 0.33, STD: 0.07, F-Score Mean: 0.35, STD: 0.04 AUC Mean: 0.65, STD: 0.03 G-Mean: 0.56, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 31, Precision Mean: 0.36 STD: 0.04, Recall Mean: 0.34, STD: 0.07, F-Score Mean: 0.34, STD: 0.05 AUC Mean: 0.65, STD: 0.03 G-Mean: 0.57, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 41, Precision Mean: 0.40 STD: 0.10, Recall Mean: 0.34, STD: 0.12, F-Score Mean: 0.37, STD: 0.11 AUC Mean: 0.66, STD: 0.06 G-Mean: 0.57, STD: 0.10

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 51, Precision Mean: 0.40 STD: 0.11, Recall Mean: 0.35, STD: 0.08, F-Score Mean: 0.37, STD: 0.09 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.58, STD: 0.07

lambda: 0.6

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 11, Precision Mean: 0.32 STD: 0.06, Recall Mean: 0.29, STD: 0.05, F-Score Mean: 0.30, STD: 0.04 AUC Mean: 0.62, STD: 0.02 G-Mean: 0.53, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 21, Precision Mean: 0.32 STD: 0.07, Recall Mean: 0.35, STD: 0.12, F-Score Mean: 0.33, STD: 0.08 AUC Mean: 0.65, STD: 0.06 G-Mean: 0.57, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 31, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.37, STD: 0.11, F-Score Mean: 0.35, STD: 0.07 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.58, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 41, Precision Mean: 0.40 STD: 0.07, Recall Mean: 0.35, STD: 0.08, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.57, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 51, Precision Mean: 0.40 STD: 0.08, Recall Mean: 0.36, STD: 0.10, F-Score Mean: 0.37, STD: 0.08 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.58, STD: 0.08

lambda: 1.1

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 11, Precision Mean: 0.33 STD: 0.06, Recall Mean: 0.34, STD: 0.09, F-Score Mean: 0.33, STD: 0.07 AUC Mean: 0.65, STD: 0.04 G-Mean: 0.56, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 21, Precision Mean: 0.37 STD: 0.08, Recall Mean: 0.37, STD: 0.08, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.59, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 31, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.35, STD: 0.12, F-Score Mean: 0.35, STD: 0.08 AUC Mean: 0.66, STD: 0.06 G-Mean: 0.57, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 41, Precision Mean: 0.42 STD: 0.06, Recall Mean: 0.36, STD: 0.09, F-Score Mean: 0.38, STD: 0.08 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.58, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 51, Precision Mean: 0.38 STD: 0.08, Recall Mean: 0.34, STD: 0.09, F-Score Mean: 0.35, STD: 0.07 AUC Mean: 0.65, STD: 0.04 G-Mean: 0.56, STD: 0.08

lambda: 1.6

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 11, Precision Mean: 0.36 STD: 0.07, Recall Mean: 0.36, STD: 0.06, F-Score Mean: 0.36, STD: 0.06 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.58, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 21, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.34, STD: 0.07, F-Score Mean: 0.34, STD: 0.06 AUC Mean: 0.65, STD: 0.03 G-Mean: 0.57, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 31, Precision Mean: 0.36 STD: 0.05, Recall Mean: 0.36, STD: 0.09, F-Score Mean: 0.35, STD: 0.06 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.58, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 41, Precision Mean: 0.36 STD: 0.07, Recall Mean: 0.35, STD: 0.09, F-Score Mean: 0.35, STD: 0.08 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.58, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 51, Precision Mean: 0.44 STD: 0.08, Recall Mean: 0.39, STD: 0.10, F-Score Mean: 0.40, STD: 0.08 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.60, STD: 0.08

lambda: 2.1

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 11, Precision Mean: 0.33 STD: 0.08, Recall Mean: 0.38, STD: 0.10, F-Score Mean: 0.35, STD: 0.09 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.60, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 21, Precision Mean: 0.37 STD: 0.04, Recall Mean: 0.38, STD: 0.10, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.59, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 31, Precision Mean: 0.40 STD: 0.06, Recall Mean: 0.38, STD: 0.07, F-Score Mean: 0.39, STD: 0.04 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.60, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 41, Precision Mean: 0.34 STD: 0.06, Recall Mean: 0.32, STD: 0.07, F-Score Mean: 0.33, STD: 0.06 AUC Mean: 0.64, STD: 0.03 G-Mean: 0.55, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 51, Precision Mean: 0.40 STD: 0.10, Recall Mean: 0.41, STD: 0.09, F-Score Mean: 0.40, STD: 0.09 AUC Mean: 0.69, STD: 0.05 G-Mean: 0.63, STD: 0.07

lambda: 2.6

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 11, Precision Mean: 0.37 STD: 0.05, Recall Mean: 0.36, STD: 0.07, F-Score Mean: 0.37, STD: 0.05 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.59, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 21, Precision Mean: 0.33 STD: 0.08, Recall Mean: 0.36, STD: 0.12, F-Score Mean: 0.34, STD: 0.09 AUC Mean: 0.65, STD: 0.06 G-Mean: 0.57, STD: 0.10

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 31, Precision Mean: 0.37 STD: 0.04, Recall Mean: 0.41, STD: 0.09, F-Score Mean: 0.39, STD: 0.05 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.62, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 41, Precision Mean: 0.35 STD: 0.04, Recall Mean: 0.40, STD: 0.12, F-Score Mean: 0.37, STD: 0.07 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.61, STD: 0.09

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 51, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.38, STD: 0.11, F-Score Mean: 0.36, STD: 0.09 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.60, STD: 0.09

lambda: 3.1

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 11, Precision Mean: 0.32 STD: 0.07, Recall Mean: 0.37, STD: 0.09, F-Score Mean: 0.34, STD: 0.07 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.59, STD: 0.07

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 21, Precision Mean: 0.37 STD: 0.07, Recall Mean: 0.43, STD: 0.10, F-Score Mean: 0.39, STD: 0.08 AUC Mean: 0.69, STD: 0.05 G-Mean: 0.63, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 31, Precision Mean: 0.38 STD: 0.08, Recall Mean: 0.48, STD: 0.16, F-Score Mean: 0.41, STD: 0.10 AUC Mean: 0.71, STD: 0.07 G-Mean: 0.66, STD: 0.11

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 41, Precision Mean: 0.37 STD: 0.07, Recall Mean: 0.46, STD: 0.16, F-Score Mean: 0.40, STD: 0.09 AUC Mean: 0.70, STD: 0.07 G-Mean: 0.65, STD: 0.11

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 51, Precision Mean: 0.33 STD: 0.06, Recall Mean: 0.41, STD: 0.11, F-Score Mean: 0.36, STD: 0.07 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.61, STD: 0.08

lambda: 3.6

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 11, Precision Mean: 0.39 STD: 0.06, Recall Mean: 0.45, STD: 0.09, F-Score Mean: 0.41, STD: 0.05 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.65, STD: 0.06

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 21, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.45, STD: 0.11, F-Score Mean: 0.40, STD: 0.07 AUC Mean: 0.70, STD: 0.05 G-Mean: 0.65, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 31, Precision Mean: 0.32 STD: 0.05, Recall Mean: 0.45, STD: 0.12, F-Score Mean: 0.37, STD: 0.07 AUC Mean: 0.69, STD: 0.05 G-Mean: 0.64, STD: 0.08

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 41, Precision Mean: 0.34 STD: 0.06, Recall Mean: 0.43, STD: 0.07, F-Score Mean: 0.38, STD: 0.06 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.63, STD: 0.05

Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 51, Precision Mean: 0.33 STD: 0.06, Recall Mean: 0.50, STD: 0.08, F-Score Mean: 0.40, STD: 0.06 AUC Mean: 0.72, STD: 0.04 G-Mean: 0.68, STD: 0.06

lambda: 0.1

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 11, Precision Mean: 0.36 STD: 0.05, Recall Mean: 0.40, STD: 0.07, F-Score Mean: 0.37, STD: 0.05 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.61, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 21, Precision Mean: 0.35 STD: 0.05, Recall Mean: 0.34, STD: 0.05, F-Score Mean: 0.34, STD: 0.04 AUC Mean: 0.65, STD: 0.02 G-Mean: 0.57, STD: 0.04

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 31, Precision Mean: 0.37 STD: 0.07, Recall Mean: 0.40, STD: 0.09, F-Score Mean: 0.38, STD: 0.06 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.61, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 41, Precision Mean: 0.36 STD: 0.07, Recall Mean: 0.39, STD: 0.10, F-Score Mean: 0.37, STD: 0.07 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.61, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 51, Precision Mean: 0.39 STD: 0.10, Recall Mean: 0.42, STD: 0.11, F-Score Mean: 0.40, STD: 0.10 AUC Mean: 0.69, STD: 0.06 G-Mean: 0.63, STD: 0.10

lambda: 0.6

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 11, Precision Mean: 0.39 STD: 0.09, Recall Mean: 0.38, STD: 0.07, F-Score Mean: 0.38, STD: 0.07 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.60, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 21, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.41, STD: 0.07, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.62, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 31, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.41, STD: 0.13, F-Score Mean: 0.38, STD: 0.09 AUC Mean: 0.68, STD: 0.06 G-Mean: 0.61, STD: 0.12

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 41, Precision Mean: 0.39 STD: 0.06, Recall Mean: 0.46, STD: 0.07, F-Score Mean: 0.42, STD: 0.04 AUC Mean: 0.71, STD: 0.03 G-Mean: 0.66, STD: 0.04

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 51, Precision Mean: 0.38 STD: 0.07, Recall Mean: 0.41, STD: 0.11, F-Score Mean: 0.39, STD: 0.08 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.62, STD: 0.08

lambda: 1.1

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 11, Precision Mean: 0.38 STD: 0.04, Recall Mean: 0.43, STD: 0.08, F-Score Mean: 0.40, STD: 0.05 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.64, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 21, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.40, STD: 0.10, F-Score Mean: 0.38, STD: 0.08 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.61, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 31, Precision Mean: 0.35 STD: 0.06, Recall Mean: 0.38, STD: 0.09, F-Score Mean: 0.36, STD: 0.06 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.60, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 41, Precision Mean: 0.37 STD: 0.05, Recall Mean: 0.41, STD: 0.08, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.62, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 51, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.32, STD: 0.05, F-Score Mean: 0.34, STD: 0.05 AUC Mean: 0.64, STD: 0.03 G-Mean: 0.56, STD: 0.05

lambda: 1.6

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 11, Precision Mean: 0.38 STD: 0.04, Recall Mean: 0.38, STD: 0.07, F-Score Mean: 0.38, STD: 0.04 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.60, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 21, Precision Mean: 0.41 STD: 0.08, Recall Mean: 0.44, STD: 0.07, F-Score Mean: 0.42, STD: 0.07 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.64, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 31, Precision Mean: 0.37 STD: 0.05, Recall Mean: 0.39, STD: 0.06, F-Score Mean: 0.38, STD: 0.05 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.61, STD: 0.04

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 41, Precision Mean: 0.40 STD: 0.06, Recall Mean: 0.40, STD: 0.09, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.61, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 51, Precision Mean: 0.38 STD: 0.04, Recall Mean: 0.39, STD: 0.06, F-Score Mean: 0.38, STD: 0.04 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.61, STD: 0.04

lambda: 2.1

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 11, Precision Mean: 0.38 STD: 0.06, Recall Mean: 0.42, STD: 0.11, F-Score Mean: 0.40, STD: 0.07 AUC Mean: 0.69, STD: 0.05 G-Mean: 0.63, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 21, Precision Mean: 0.38 STD: 0.06, Recall Mean: 0.41, STD: 0.08, F-Score Mean: 0.39, STD: 0.07 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.62, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 31, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.46, STD: 0.06, F-Score Mean: 0.41, STD: 0.04 AUC Mean: 0.70, STD: 0.03 G-Mean: 0.66, STD: 0.04

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 41, Precision Mean: 0.40 STD: 0.04, Recall Mean: 0.42, STD: 0.07, F-Score Mean: 0.41, STD: 0.04 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.63, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 51, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.39, STD: 0.09, F-Score Mean: 0.38, STD: 0.07 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.61, STD: 0.07

lambda: 2.6

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 11, Precision Mean: 0.34 STD: 0.04, Recall Mean: 0.45, STD: 0.09, F-Score Mean: 0.38, STD: 0.05 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.65, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 21, Precision Mean: 0.37 STD: 0.04, Recall Mean: 0.44, STD: 0.09, F-Score Mean: 0.40, STD: 0.05 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.64, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 31, Precision Mean: 0.34 STD: 0.06, Recall Mean: 0.40, STD: 0.06, F-Score Mean: 0.36, STD: 0.05 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.61, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 41, Precision Mean: 0.39 STD: 0.07, Recall Mean: 0.48, STD: 0.07, F-Score Mean: 0.43, STD: 0.06 AUC Mean: 0.71, STD: 0.03 G-Mean: 0.67, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 51, Precision Mean: 0.34 STD: 0.07, Recall Mean: 0.41, STD: 0.12, F-Score Mean: 0.36, STD: 0.09 AUC Mean: 0.68, STD: 0.06 G-Mean: 0.61, STD: 0.09

lambda: 3.1

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 11, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.44, STD: 0.09, F-Score Mean: 0.40, STD: 0.07 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.64, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 21, Precision Mean: 0.38 STD: 0.07, Recall Mean: 0.45, STD: 0.08, F-Score Mean: 0.41, STD: 0.07 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.65, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 31, Precision Mean: 0.41 STD: 0.06, Recall Mean: 0.52, STD: 0.06, F-Score Mean: 0.45, STD: 0.04 AUC Mean: 0.73, STD: 0.03 G-Mean: 0.70, STD: 0.04

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 41, Precision Mean: 0.38 STD: 0.05, Recall Mean: 0.47, STD: 0.07, F-Score Mean: 0.42, STD: 0.05 AUC Mean: 0.71, STD: 0.03 G-Mean: 0.67, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 51, Precision Mean: 0.39 STD: 0.04, Recall Mean: 0.47, STD: 0.06, F-Score Mean: 0.43, STD: 0.04 AUC Mean: 0.71, STD: 0.03 G-Mean: 0.67, STD: 0.04

lambda: 3.6

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 11, Precision Mean: 0.34 STD: 0.07, Recall Mean: 0.41, STD: 0.07, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.62, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 21, Precision Mean: 0.36 STD: 0.07, Recall Mean: 0.48, STD: 0.10, F-Score Mean: 0.41, STD: 0.07 AUC Mean: 0.71, STD: 0.05 G-Mean: 0.67, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 31, Precision Mean: 0.36 STD: 0.07, Recall Mean: 0.55, STD: 0.05, F-Score Mean: 0.43, STD: 0.05 AUC Mean: 0.74, STD: 0.02 G-Mean: 0.71, STD: 0.03

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 41, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.45, STD: 0.09, F-Score Mean: 0.40, STD: 0.06 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.65, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 51, Precision Mean: 0.36 STD: 0.05, Recall Mean: 0.46, STD: 0.08, F-Score Mean: 0.40, STD: 0.06 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.66, STD: 0.06

lambda: 0.1

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 11, Precision Mean: 0.34 STD: 0.06, Recall Mean: 0.35, STD: 0.10, F-Score Mean: 0.34, STD: 0.08 AUC Mean: 0.65, STD: 0.05 G-Mean: 0.57, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 21, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.34, STD: 0.06, F-Score Mean: 0.34, STD: 0.06 AUC Mean: 0.65, STD: 0.03 G-Mean: 0.57, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 31, Precision Mean: 0.37 STD: 0.05, Recall Mean: 0.37, STD: 0.03, F-Score Mean: 0.37, STD: 0.03 AUC Mean: 0.66, STD: 0.01 G-Mean: 0.60, STD: 0.03

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 41, Precision Mean: 0.37 STD: 0.05, Recall Mean: 0.40, STD: 0.13, F-Score Mean: 0.38, STD: 0.08 AUC Mean: 0.68, STD: 0.06 G-Mean: 0.61, STD: 0.10

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 51, Precision Mean: 0.35 STD: 0.05, Recall Mean: 0.35, STD: 0.11, F-Score Mean: 0.34, STD: 0.07 AUC Mean: 0.65, STD: 0.05 G-Mean: 0.57, STD: 0.10

lambda: 0.6

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 11, Precision Mean: 0.37 STD: 0.07, Recall Mean: 0.34, STD: 0.11, F-Score Mean: 0.35, STD: 0.08 AUC Mean: 0.65, STD: 0.05 G-Mean: 0.57, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 21, Precision Mean: 0.37 STD: 0.07, Recall Mean: 0.35, STD: 0.10, F-Score Mean: 0.35, STD: 0.07 AUC Mean: 0.65, STD: 0.05 G-Mean: 0.57, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 31, Precision Mean: 0.38 STD: 0.08, Recall Mean: 0.36, STD: 0.10, F-Score Mean: 0.36, STD: 0.08 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.58, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 41, Precision Mean: 0.38 STD: 0.10, Recall Mean: 0.34, STD: 0.11, F-Score Mean: 0.35, STD: 0.09 AUC Mean: 0.65, STD: 0.05 G-Mean: 0.57, STD: 0.10

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 51, Precision Mean: 0.41 STD: 0.09, Recall Mean: 0.41, STD: 0.07, F-Score Mean: 0.40, STD: 0.06 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.62, STD: 0.05

lambda: 1.1

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 11, Precision Mean: 0.29 STD: 0.07, Recall Mean: 0.31, STD: 0.08, F-Score Mean: 0.29, STD: 0.07 AUC Mean: 0.63, STD: 0.04 G-Mean: 0.53, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 21, Precision Mean: 0.37 STD: 0.08, Recall Mean: 0.38, STD: 0.09, F-Score Mean: 0.37, STD: 0.07 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.60, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 31, Precision Mean: 0.38 STD: 0.07, Recall Mean: 0.36, STD: 0.06, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.58, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 41, Precision Mean: 0.38 STD: 0.11, Recall Mean: 0.34, STD: 0.09, F-Score Mean: 0.36, STD: 0.08 AUC Mean: 0.65, STD: 0.04 G-Mean: 0.57, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 51, Precision Mean: 0.41 STD: 0.06, Recall Mean: 0.35, STD: 0.07, F-Score Mean: 0.37, STD: 0.05 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.58, STD: 0.05

lambda: 1.6

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 11, Precision Mean: 0.42 STD: 0.10, Recall Mean: 0.40, STD: 0.06, F-Score Mean: 0.40, STD: 0.06 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.62, STD: 0.04

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 21, Precision Mean: 0.37 STD: 0.05, Recall Mean: 0.39, STD: 0.09, F-Score Mean: 0.38, STD: 0.06 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.61, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 31, Precision Mean: 0.38 STD: 0.05, Recall Mean: 0.41, STD: 0.11, F-Score Mean: 0.39, STD: 0.07 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.62, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 41, Precision Mean: 0.38 STD: 0.05, Recall Mean: 0.40, STD: 0.08, F-Score Mean: 0.38, STD: 0.05 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.61, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 51, Precision Mean: 0.38 STD: 0.07, Recall Mean: 0.36, STD: 0.07, F-Score Mean: 0.36, STD: 0.06 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.58, STD: 0.05

lambda: 2.1

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 11, Precision Mean: 0.38 STD: 0.07, Recall Mean: 0.42, STD: 0.07, F-Score Mean: 0.40, STD: 0.07 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.63, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 21, Precision Mean: 0.38 STD: 0.05, Recall Mean: 0.42, STD: 0.08, F-Score Mean: 0.40, STD: 0.06 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.63, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 31, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.40, STD: 0.10, F-Score Mean: 0.38, STD: 0.06 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.61, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 41, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.39, STD: 0.11, F-Score Mean: 0.38, STD: 0.07 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.61, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 51, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.42, STD: 0.09, F-Score Mean: 0.39, STD: 0.07 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.63, STD: 0.07

lambda: 2.6

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 11, Precision Mean: 0.39 STD: 0.05, Recall Mean: 0.42, STD: 0.08, F-Score Mean: 0.40, STD: 0.05 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.63, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 21, Precision Mean: 0.39 STD: 0.08, Recall Mean: 0.46, STD: 0.09, F-Score Mean: 0.41, STD: 0.06 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.66, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 31, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.46, STD: 0.07, F-Score Mean: 0.40, STD: 0.05 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.65, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 41, Precision Mean: 0.37 STD: 0.05, Recall Mean: 0.45, STD: 0.11, F-Score Mean: 0.40, STD: 0.07 AUC Mean: 0.70, STD: 0.05 G-Mean: 0.65, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 51, Precision Mean: 0.35 STD: 0.05, Recall Mean: 0.44, STD: 0.08, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.64, STD: 0.06

lambda: 3.1

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 11, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.40, STD: 0.07, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.61, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 21, Precision Mean: 0.31 STD: 0.05, Recall Mean: 0.37, STD: 0.07, F-Score Mean: 0.34, STD: 0.04 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.59, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 31, Precision Mean: 0.37 STD: 0.08, Recall Mean: 0.44, STD: 0.07, F-Score Mean: 0.40, STD: 0.07 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.64, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 41, Precision Mean: 0.35 STD: 0.05, Recall Mean: 0.47, STD: 0.12, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.70, STD: 0.05 G-Mean: 0.66, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 51, Precision Mean: 0.37 STD: 0.08, Recall Mean: 0.47, STD: 0.10, F-Score Mean: 0.41, STD: 0.07 AUC Mean: 0.71, STD: 0.05 G-Mean: 0.66, STD: 0.07

lambda: 3.6

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 11, Precision Mean: 0.34 STD: 0.05, Recall Mean: 0.46, STD: 0.08, F-Score Mean: 0.39, STD: 0.05 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.65, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 21, Precision Mean: 0.31 STD: 0.03, Recall Mean: 0.48, STD: 0.08, F-Score Mean: 0.38, STD: 0.04 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.67, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 31, Precision Mean: 0.35 STD: 0.06, Recall Mean: 0.48, STD: 0.11, F-Score Mean: 0.41, STD: 0.07 AUC Mean: 0.71, STD: 0.05 G-Mean: 0.67, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 41, Precision Mean: 0.38 STD: 0.06, Recall Mean: 0.51, STD: 0.10, F-Score Mean: 0.43, STD: 0.07 AUC Mean: 0.73, STD: 0.05 G-Mean: 0.69, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 51, Precision Mean: 0.35 STD: 0.05, Recall Mean: 0.49, STD: 0.10, F-Score Mean: 0.40, STD: 0.06 AUC Mean: 0.71, STD: 0.05 G-Mean: 0.67, STD: 0.07

lambda: 0.1

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.33 STD: 0.05, Recall Mean: 0.37, STD: 0.08, F-Score Mean: 0.35, STD: 0.06 AUC Mean: 0.66, STD: 0.04 G-Mean: 0.59, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.40, STD: 0.07, F-Score Mean: 0.38, STD: 0.06 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.62, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.42 STD: 0.08, Recall Mean: 0.38, STD: 0.07, F-Score Mean: 0.40, STD: 0.05 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.60, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.39 STD: 0.10, Recall Mean: 0.37, STD: 0.11, F-Score Mean: 0.37, STD: 0.09 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.59, STD: 0.09

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.41 STD: 0.07, Recall Mean: 0.37, STD: 0.05, F-Score Mean: 0.39, STD: 0.05 AUC Mean: 0.67, STD: 0.03 G-Mean: 0.60, STD: 0.04

lambda: 0.6

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.33 STD: 0.07, Recall Mean: 0.33, STD: 0.10, F-Score Mean: 0.33, STD: 0.08 AUC Mean: 0.64, STD: 0.05 G-Mean: 0.56, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.34 STD: 0.04, Recall Mean: 0.33, STD: 0.06, F-Score Mean: 0.33, STD: 0.04 AUC Mean: 0.64, STD: 0.03 G-Mean: 0.56, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.40 STD: 0.05, Recall Mean: 0.40, STD: 0.09, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.61, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.43 STD: 0.09, Recall Mean: 0.34, STD: 0.07, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.57, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.38 STD: 0.09, Recall Mean: 0.31, STD: 0.07, F-Score Mean: 0.33, STD: 0.05 AUC Mean: 0.64, STD: 0.03 G-Mean: 0.54, STD: 0.06

lambda: 1.1

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.33 STD: 0.06, Recall Mean: 0.32, STD: 0.11, F-Score Mean: 0.31, STD: 0.07 AUC Mean: 0.64, STD: 0.05 G-Mean: 0.54, STD: 0.09

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.35 STD: 0.05, Recall Mean: 0.36, STD: 0.07, F-Score Mean: 0.35, STD: 0.05 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.58, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.34 STD: 0.06, Recall Mean: 0.32, STD: 0.07, F-Score Mean: 0.33, STD: 0.06 AUC Mean: 0.64, STD: 0.03 G-Mean: 0.55, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.38 STD: 0.08, Recall Mean: 0.35, STD: 0.08, F-Score Mean: 0.36, STD: 0.06 AUC Mean: 0.65, STD: 0.04 G-Mean: 0.58, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.35, STD: 0.11, F-Score Mean: 0.35, STD: 0.08 AUC Mean: 0.65, STD: 0.05 G-Mean: 0.57, STD: 0.09

lambda: 1.6

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.33 STD: 0.06, Recall Mean: 0.33, STD: 0.09, F-Score Mean: 0.33, STD: 0.07 AUC Mean: 0.64, STD: 0.04 G-Mean: 0.56, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.39 STD: 0.07, Recall Mean: 0.37, STD: 0.09, F-Score Mean: 0.38, STD: 0.07 AUC Mean: 0.67, STD: 0.04 G-Mean: 0.59, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.35 STD: 0.06, Recall Mean: 0.34, STD: 0.08, F-Score Mean: 0.34, STD: 0.06 AUC Mean: 0.65, STD: 0.04 G-Mean: 0.56, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.35 STD: 0.07, Recall Mean: 0.37, STD: 0.09, F-Score Mean: 0.36, STD: 0.07 AUC Mean: 0.66, STD: 0.05 G-Mean: 0.59, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.40 STD: 0.05, Recall Mean: 0.37, STD: 0.07, F-Score Mean: 0.38, STD: 0.05 AUC Mean: 0.66, STD: 0.03 G-Mean: 0.59, STD: 0.06

lambda: 2.1

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.36 STD: 0.06, Recall Mean: 0.42, STD: 0.09, F-Score Mean: 0.39, STD: 0.06 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.63, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.36 STD: 0.08, Recall Mean: 0.37, STD: 0.13, F-Score Mean: 0.35, STD: 0.08 AUC Mean: 0.66, STD: 0.06 G-Mean: 0.58, STD: 0.10

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.39 STD: 0.06, Recall Mean: 0.43, STD: 0.10, F-Score Mean: 0.40, STD: 0.07 AUC Mean: 0.69, STD: 0.05 G-Mean: 0.64, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.41 STD: 0.07, Recall Mean: 0.41, STD: 0.07, F-Score Mean: 0.40, STD: 0.05 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.62, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.38, STD: 0.10, F-Score Mean: 0.37, STD: 0.07 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.60, STD: 0.08

lambda: 2.6

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.38 STD: 0.11, Recall Mean: 0.43, STD: 0.09, F-Score Mean: 0.40, STD: 0.08 AUC Mean: 0.69, STD: 0.04 G-Mean: 0.63, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.38 STD: 0.06, Recall Mean: 0.42, STD: 0.06, F-Score Mean: 0.39, STD: 0.05 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.63, STD: 0.04

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.35 STD: 0.03, Recall Mean: 0.40, STD: 0.07, F-Score Mean: 0.37, STD: 0.03 AUC Mean: 0.68, STD: 0.03 G-Mean: 0.62, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.42 STD: 0.05, Recall Mean: 0.42, STD: 0.07, F-Score Mean: 0.42, STD: 0.05 AUC Mean: 0.69, STD: 0.03 G-Mean: 0.63, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.34 STD: 0.03, Recall Mean: 0.42, STD: 0.10, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.68, STD: 0.05 G-Mean: 0.63, STD: 0.07

lambda: 3.1

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.39 STD: 0.08, Recall Mean: 0.42, STD: 0.10, F-Score Mean: 0.40, STD: 0.08 AUC Mean: 0.69, STD: 0.05 G-Mean: 0.63, STD: 0.07

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.35 STD: 0.06, Recall Mean: 0.43, STD: 0.11, F-Score Mean: 0.38, STD: 0.07 AUC Mean: 0.69, STD: 0.05 G-Mean: 0.63, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.37 STD: 0.07, Recall Mean: 0.51, STD: 0.08, F-Score Mean: 0.42, STD: 0.06 AUC Mean: 0.72, STD: 0.04 G-Mean: 0.69, STD: 0.05

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.34 STD: 0.07, Recall Mean: 0.48, STD: 0.14, F-Score Mean: 0.40, STD: 0.09 AUC Mean: 0.71, STD: 0.07 G-Mean: 0.66, STD: 0.10

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.36 STD: 0.04, Recall Mean: 0.51, STD: 0.07, F-Score Mean: 0.42, STD: 0.03 AUC Mean: 0.72, STD: 0.03 G-Mean: 0.69, STD: 0.04

lambda: 3.6

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 11, Precision Mean: 0.34 STD: 0.04, Recall Mean: 0.42, STD: 0.08, F-Score Mean: 0.37, STD: 0.05 AUC Mean: 0.68, STD: 0.04 G-Mean: 0.63, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 21, Precision Mean: 0.33 STD: 0.06, Recall Mean: 0.48, STD: 0.09, F-Score Mean: 0.38, STD: 0.05 AUC Mean: 0.70, STD: 0.04 G-Mean: 0.66, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 31, Precision Mean: 0.34 STD: 0.07, Recall Mean: 0.48, STD: 0.05, F-Score Mean: 0.39, STD: 0.05 AUC Mean: 0.71, STD: 0.03 G-Mean: 0.67, STD: 0.04

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 41, Precision Mean: 0.33 STD: 0.06, Recall Mean: 0.47, STD: 0.12, F-Score Mean: 0.39, STD: 0.08 AUC Mean: 0.70, STD: 0.06 G-Mean: 0.66, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.30000000000000004, #Trees: 51, Precision Mean: 0.32 STD: 0.06, Recall Mean: 0.51, STD: 0.08, F-Score Mean: 0.39, STD: 0.07 AUC Mean: 0.72, STD: 0.04 G-Mean: 0.68, STD: 0.06

lambda: 0.1

Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 11, Precision Mean: 0.34 STD: 0.07, Recall Mean: 0.33, STD: 0.10, F-Score Mean: 0.32, STD: 0.07 AUC Mean: 0.64, STD: 0.04 G-Mean: 0.55, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 21, Precision Mean: 0.36 STD: 0.07, Recall Mean: 0.31, STD: 0.09, F-Score Mean: 0.33, STD: 0.07 AUC Mean: 0.64, STD: 0.04 G-Mean: 0.54, STD: 0.08

Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 31, Precision Mean: 0.42 STD: 0.09, Recall Mean: 0.34, STD: 0.08, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.65, STD: 0.04 G-Mean: 0.57, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 41, Precision Mean: 0.38 STD: 0.07, Recall Mean: 0.29, STD: 0.07, F-Score Mean: 0.32, STD: 0.05 AUC Mean: 0.63, STD: 0.03 G-Mean: 0.53, STD: 0.06

Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 51, Precision Mean: 0.36 STD: 0.12, Recall Mean: 0.32, STD: 0.14, F-Score Mean: 0.33, STD: 0.12 AUC Mean: 0.64, STD: 0.07 G-Mean: 0.54, STD: 0.14

lambda: 0.6

Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 11, Precision Mean: 0.37 STD: 0.06, Recall Mean: 0.38, STD: 0.11, F-Score Mean: 0.37, STD: 0.06 AUC Mean: 0.67, STD: 0.05 G-Mean: 0.60, STD: 0.08

کتابخانه های مورد استفاده:

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_recall_fscore_support, roc_auc_score
from imblearn.metrics import geometric_mean_score
import numpy as np
from imblearn.over_sampling import SMOTE, RandomOverSampler
from imblearn.under_sampling import RandomUnderSampler
import scipy.stats as ss
```

خواندن فایل

```
def load_data(file_name):
    df = pd.read_csv(file_name)
    df = df.fillna(-1)
    x = df.iloc[:, :-1].values
    y = df.iloc[:, -1].values
```

جایگذاری مقادیر خالی با مقدار 1-

```
df = df.fillna(-1)
```

جداسازی:

مقادیر ویژگیها

```
x = df.iloc[:, :-1].values
```

مقادیر کلاس ها

```
y = df.iloc[:, -1].values
```

کلاس 1- را به 0 تغییر دادم مقدارش را چون در کد نویسی اندیس 1- نداشتیم و کار راحت تر است.

```
y[y==1]=0
```

پایه سازی مربوط به کلاسیفایرهای بخش Adaboost و Bagging

```
def report_measures_for_clf(x, y, clf, sampler):
```

این تابع x و y را دریافت میکند بعد یک کلاسیفایر دریافت میکند و یک Sampler که سَمپلر میتواند smote , under sampler و oversampler باشد.

```
trn_x, tst_x, trn_y, tst_y = train_test_split(x, y, train_size = trn_ratio, stratify = y)
```

با استفاده از تابع `train_test_split` مربوط به کتابخانه `sklearn` داده ها را به `train` و `test` تقسیم میکنم.

با استفاده از `sampler` گفته شده داده ها را نمونه گیری کردم.

```
clf.fit(s_x, s_y)
```

کلاسیفایر را روی داده ها `fit` کردم

```
p = clf.predict_proba(tst_x)
```

در نهایت روی داده های `test` احتمال هر کدام از کلاس ها را بدست آوردم.

```
prd = np.argmax(p, axis=1)
prd[prd == 0] = -1
```

و بعد لیبل کلاسهای `prd` را بدست آوردم و جاهایی که لیبل صفر شده مقدار `-1` قرار میدهم.

```
prc, recall, fscore, _ = precision_recall_fscore_support(tst_y, prd, labels=[1])
```

با استفاده از این تابع مقادیر `precision`, `recall`, `fscore` را بدست می آورم

```
auc = roc_auc_score(tst_y, p[:, 1])
```

مقدار `auc` را بدست می آورم

```
g_mean = geometric_mean_score(tst_y, prd)
```

و در نهایت مقدار `g_mean`

```
clfs = {'NB': GaussianNB(), 'Linear-SVM': SVC(kernel='linear', probability=True),
        'RBF-SVM': SVC(probability=True), '1NN': KNeighborsClassifier(n_neighbors=1)}
```

ساخت یک دیکشنری برای ذخیره کلاسیفایرها

```
samplers = {'SMOTE': SMOTE(), 'UnderSampler': RandomUnderSampler(), 'OverSampler':
            RandomOverSampler()}
```

`sampler` ها با استفاده از کتابخانه `imblearn`

```
file_name = 'Covid-19.csv'
```

```
trn_ratio = 0.7
```

خواندن داده ها به نسبت 70 درصد `train`

```
n_itr = 10
```

تعداد تکرار برای هر کدام از موارد و میانگین بگیریم

```
for clf_name in clfs:
    for sampler_name in samplers:
```

برای هر کدام از کلاسیفایرها و نمونه گیرها کاری که انجام شده این هست که

```
for i in range(n_itr):
    p, r, f, a, g = report_measures_for_clf(x, y, clfs[clf_name], samplers[sampler_name])
```

به اندازه 10 بار (n_itr) تابع report_measures را صدا کردم و ورودی داده ها کل داده ها بوده که تقسیم به train و test شده کلاسیفایر و sampler نیز به آن داده شده و مقادیر p=precision, r=recall, f= fscore, a=auc, g=gmean را حساب کرده و برگردانده.

```
prc += p
recall += r
fscore += f
auc += a
g_mean += g
```

و آن مقادیر را با مقادیر قبلی جمع کرده

```
prc /= n_itr
recall /= n_itr
fscore /= n_itr
auc /= n_itr
g_mean /= n_itr
```

و در نهایت به تعداد تکرار تقسیم کردم تا مقدار میانگین بدست بیاد

```
print('{}, {}, Precision: {:.2f}, Recall: {:.2f}, F-Score: {:.2f}, AUC: {:.2f}, G-Mean: {:.2f}'.format(clf_name,
sampler_name, prc, recall, fscore, auc, g_mean))
```

و در آخر نیز مقدار خروجی چاپ شده

NB, SMOTE, Precision: 0.19, Recall: 0.86, F-Score: 0.31, AUC: 0.81, G-Mean: 0.81

NB, UnderSampler, Precision: 0.18, Recall: 0.90, F-Score: 0.30, AUC: 0.81, G-Mean: 0.80

NB, OverSampler, Precision: 0.08, Recall: 0.94, F-Score: 0.15, AUC: 0.82, G-Mean: 0.47

=====

Linear-SVM, SMOTE, Precision: 0.21, Recall: 0.80, F-Score: 0.33, AUC: 0.86, G-Mean: 0.80

Linear-SVM, UnderSampler, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.88, G-Mean: 0.82

Linear-SVM, OverSampler, Precision: 0.20, Recall: 0.85, F-Score: 0.32, AUC: 0.86, G-Mean: 0.81

=====

RBF-SVM, SMOTE, Precision: 0.28, Recall: 0.69, F-Score: 0.40, AUC: 0.89, G-Mean: 0.78

RBF-SVM, UnderSampler, Precision: 0.19, Recall: 0.94, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

RBF-SVM, OverSampler, Precision: 0.27, Recall: 0.75, F-Score: 0.40, AUC: 0.90, G-Mean: 0.80

=====

NN, SMOTE, Precision: 0.28, Recall: 0.55, F-Score: 0.37, AUC: 0.73, G-Mean: 0.701

NN, UnderSampler, Precision: 0.20, Recall: 0.82, F-Score: 0.32, AUC: 0.80, G-Mean: 0.801

NN, OverSampler, Precision: 0.29, Recall: 0.48, F-Score: 0.36, AUC: 0.70, G-Mean: 0.661

=====

Bagging

```
def get_a_bootstrap(x, y):
```

ساخت bootstrap از داده های کلاس ماینر و کلاس میجر bootstrap میگیریم و کنار هم قرار میدهیم.

```
mjr_x = x[y == -1]
mjr_y = y[y == -1]
```

```
mnr_x = x[y == 1]
mnr_y = y[y == 1]
```

جدا سازی کلاس ماینر از میجر

```
n_mnr = mnr_x.shape[0]
n_mjr = mjr_x.shape[0]
```

بدست آوردن تعداد داده های کلاس ماینر و میجر

```
r_mnr = np.random.randint(0, n_mnr, n_mnr)
r_mjr = np.random.randint(0, n_mjr, n_mnr)
```

به تعداد داده های ماینر و میجر عدد تصادفی تولید کردم.

```
x_bs = np.concatenate([mjr_x[r_mjr], mnr_x[r_mnr]], axis = 0)
y_bs = np.concatenate([mjr_y[r_mjr], mnr_y[r_mnr]], axis = 0)
```

به تعداد مساوی sample گرفته و آنها را به هم چسباندم

```
def train_bagging(n_trees, x, y):
    x, y = get_a_bootstrap(x, y)
    tree_list = []
```

یک bootstrap بگیرد و با استفاده از آن یک درخت را train کند

لیست درخت را داریم که در ابتدا خالی است

```
for _ in range(n_trees):
    tree = DecisionTreeClassifier()
    tree.fit(x, y)
    tree_list.append(tree)
```

و به تعداد درختها درخت میسازم

درخت را روی دادهای جدید ارسال شده به عنوان داده آموزشی fit میکنم

اون درخت جدید را به لیست درختهای قبلی اضافه میکنم.

```
def predict_bagging(tree_list, x):  
    p = 0
```

لیستی از درخت ها و یکسری داد (x) را دریافت میکند

```
p += tree.predict_proba(x)
```

برای هر کدام از درختها احتمال تعلق به هر کلاس را حساب میکند.

```
p /= len(tree_list)  
return p
```

احتمالات را نرمال میکنم عددی بین 0-1 برای همه کلاسها و آن احتمالات را به عنوان خروجی بر میگردانم.

```
def report_measures_for_bagging(x, y, n_trees, trn_ratio):
```

دریافت کل داده ها ، تعداد درختهایی که هر کدام از bagging ها باید داشته باشد و درصد داده ها

```
trn_x, tst_x, trn_y, tst_y = train_test_split(x, y, train_size = trn_ratio, stratify = y)  
bag_trees = train_bagging(n_trees, trn_x, trn_y)  
p = predict_bagging(bag_trees, tst_x)
```

با استفاده از تابع train_test_split مربوط به کتابخانه sklearn داده ها را به test و train تقسیم میکنم.

با استفاده از این داده ها bagging را آموزش میدهد.

و احتمال تعلق به کلاسها رو با استفاده از predict_bagging حساب میکند

```
prd = np.argmax(p, axis=1)  
prd[prd == 0] = -1
```

لیبل ها حساب میشود با استفاده از اینکه احتمال کدام کلاس بیشتر بوده

```
prc, recall, fscore, _ = precision_recall_fscore_support(tst_y, prd, labels=[1])  
auc = roc_auc_score(tst_y, p[:, 1])  
g_mean = geometric_mean_score(tst_y, prd)  
return prc[0], recall[0], fscore[0], auc, g_mean
```

مقادیر گفته شده حساب شده و به عنوان خروجی بر میگردد.

```
file_name = 'Covid-19.csv'  
trn_ratio = 0.7
```

خواندن داده ها به نسبت 70 درصد train

```
n_itr = 10
```

تعداد تکرار برای هر کدام از موارد و میانگین بگیریم

```
for n_trees in [11, 31, 51, 101]:  
    prc, recall, fscore, auc, g_mean = (0, 0, 0, 0, 0)
```

تعداد درختها را این مقادیر قرار دهیم.

```
p, r, f, a, g = report_measures_for_bagging(x, y, n_trees, trn_ratio)
```

تابع report_mesure را صدا میزنیم

Bagging

Noise Added to: train, Rn: 0.1, Rc: 0.1#Trees: 11, Precision: 0.20, Recall: 0.74, F-Score: 0.31, AUC: 0.80, G-Mean: 0.76

Noise Added to: train, Rn: 0.1, Rc: 0.1#Trees: 31, Precision: 0.22, Recall: 0.79, F-Score: 0.34, AUC: 0.83, G-Mean: 0.80

Noise Added to: train, Rn: 0.1, Rc: 0.1#Trees: 51, Precision: 0.22, Recall: 0.80, F-Score: 0.34, AUC: 0.83, G-Mean: 0.80

Noise Added to: train, Rn: 0.1, Rc: 0.1#Trees: 101, Precision: 0.20, Recall: 0.70, F-Score: 0.31, AUC: 0.79, G-Mean: 0.75

Noise Added to: train, Rn: 0.1, Rc: 0.2#Trees: 11, Precision: 0.19, Recall: 0.68, F-Score: 0.30, AUC: 0.77, G-Mean: 0.74

Noise Added to: train, Rn: 0.1, Rc: 0.2#Trees: 31, Precision: 0.21, Recall: 0.73, F-Score: 0.33, AUC: 0.80, G-Mean: 0.77

Noise Added to: train, Rn: 0.1, Rc: 0.2#Trees: 51, Precision: 0.20, Recall: 0.72, F-Score: 0.31, AUC: 0.81, G-Mean: 0.76

Noise Added to: train, Rn: 0.1, Rc: 0.2#Trees: 101, Precision: 0.23, Recall: 0.71, F-Score: 0.35, AUC: 0.80, G-Mean: 0.77

Noise Added to: train, Rn: 0.1, Rc: 0.3#Trees: 11, Precision: 0.22, Recall: 0.72, F-Score: 0.33, AUC: 0.80, G-Mean: 0.76

Noise Added to: train, Rn: 0.1, Rc: 0.3#Trees: 31, Precision: 0.20, Recall: 0.75, F-Score: 0.32, AUC: 0.81, G-Mean: 0.77

Noise Added to: train, Rn: 0.1, Rc: 0.3#Trees: 51, Precision: 0.22, Recall: 0.76, F-Score: 0.34, AUC: 0.81, G-Mean: 0.78

Noise Added to: train, Rn: 0.1, Rc: 0.3#Trees: 101, Precision: 0.21, Recall: 0.69, F-Score: 0.32, AUC: 0.78, G-Mean: 0.75

Noise Added to: train, Rn: 0.1, Rc: 0.4#Trees: 11, Precision: 0.23, Recall: 0.74, F-Score: 0.34, AUC: 0.80, G-Mean: 0.78

Noise Added to: train, Rn: 0.1, Rc: 0.4#Trees: 31, Precision: 0.21, Recall: 0.72, F-Score: 0.33, AUC: 0.81, G-Mean: 0.76

Noise Added to: train, Rn: 0.1, Rc: 0.4#Trees: 51, Precision: 0.21, Recall: 0.72, F-Score: 0.33, AUC: 0.80, G-Mean: 0.77

Noise Added to: train, Rn: 0.1, Rc: 0.4#Trees: 101, Precision: 0.22, Recall: 0.73, F-Score: 0.34, AUC: 0.81, G-Mean: 0.78

Noise Added to: train, Rn: 0.1, Rc: 0.5#Trees: 11, Precision: 0.21, Recall: 0.75, F-Score: 0.33, AUC: 0.80, G-Mean: 0.78

Noise Added to: train, Rn: 0.1, Rc: 0.5#Trees: 31, Precision: 0.22, Recall: 0.71, F-Score: 0.34, AUC: 0.80, G-Mean: 0.77

Noise Added to: train, Rn: 0.1, Rc: 0.5#Trees: 51, Precision: 0.22, Recall: 0.79, F-Score: 0.34, AUC: 0.83, G-Mean: 0.80

Noise Added to: train, Rn: 0.1, Rc: 0.5#Trees: 101, Precision: 0.21, Recall: 0.72, F-Score: 0.32, AUC: 0.80, G-Mean: 0.76

Noise Added to: train, Rn: 0.2, Rc: 0.1#Trees: 11, Precision: 0.20, Recall: 0.68, F-Score: 0.31, AUC: 0.80, G-Mean: 0.74

Noise Added to: train, Rn: 0.2, Rc: 0.1#Trees: 31, Precision: 0.22, Recall: 0.72, F-Score: 0.34, AUC: 0.79, G-Mean: 0.77

Noise Added to: train, Rn: 0.2, Rc: 0.1#Trees: 51, Precision: 0.20, Recall: 0.70, F-Score: 0.31, AUC: 0.80, G-Mean: 0.75

Noise Added to: train, Rn: 0.2, Rc: 0.1#Trees: 101, Precision: 0.21, Recall: 0.71, F-Score: 0.32, AUC: 0.80, G-Mean: 0.76

Noise Added to: train, Rn: 0.2, Rc: 0.2#Trees: 11, Precision: 0.20, Recall: 0.70, F-Score: 0.31, AUC: 0.79, G-Mean: 0.75

Noise Added to: train, Rn: 0.2, Rc: 0.2#Trees: 31, Precision: 0.22, Recall: 0.78, F-Score: 0.34, AUC: 0.83, G-Mean: 0.80

Noise Added to: train, Rn: 0.2, Rc: 0.2#Trees: 51, Precision: 0.22, Recall: 0.74, F-Score: 0.34, AUC: 0.81, G-Mean: 0.78

Noise Added to: train, Rn: 0.2, Rc: 0.2#Trees: 101, Precision: 0.22, Recall: 0.76, F-Score: 0.34, AUC: 0.81, G-Mean: 0.78

Noise Added to: train, Rn: 0.2, Rc: 0.3#Trees: 11, Precision: 0.19, Recall: 0.68, F-Score: 0.29, AUC: 0.77, G-Mean: 0.73

Noise Added to: train, Rn: 0.2, Rc: 0.3#Trees: 31, Precision: 0.22, Recall: 0.71, F-Score: 0.33, AUC: 0.80, G-Mean: 0.76

Noise Added to: train, Rn: 0.2, Rc: 0.3#Trees: 51, Precision: 0.21, Recall: 0.66, F-Score: 0.31, AUC: 0.78, G-Mean: 0.73

Noise Added to: train, Rn: 0.2, Rc: 0.3#Trees: 101, Precision: 0.21, Recall: 0.70, F-Score: 0.31, AUC: 0.79, G-Mean: 0.75

Noise Added to: train, Rn: 0.2, Rc: 0.4#Trees: 11, Precision: 0.20, Recall: 0.79, F-Score: 0.32, AUC: 0.80, G-Mean: 0.78

Noise Added to: train, Rn: 0.2, Rc: 0.4#Trees: 31, Precision: 0.23, Recall: 0.75, F-Score: 0.35, AUC: 0.82, G-Mean: 0.78

Noise Added to: train, Rn: 0.2, Rc: 0.4#Trees: 51, Precision: 0.21, Recall: 0.75, F-Score: 0.32, AUC: 0.80, G-Mean: 0.77

Noise Added to: train, Rn: 0.2, Rc: 0.4#Trees: 101, Precision: 0.21, Recall: 0.82, F-Score: 0.33, AUC: 0.84, G-Mean: 0.80

Noise Added to: train, Rn: 0.2, Rc: 0.5#Trees: 11, Precision: 0.21, Recall: 0.73, F-Score: 0.32, AUC: 0.81, G-Mean: 0.77

Noise Added to: train, Rn: 0.2, Rc: 0.5#Trees: 31, Precision: 0.22, Recall: 0.75, F-Score: 0.34, AUC: 0.82, G-Mean: 0.78

Noise Added to: train, Rn: 0.2, Rc: 0.5#Trees: 51, Precision: 0.20, Recall: 0.75, F-Score: 0.32, AUC: 0.80, G-Mean: 0.77

Noise Added to: train, Rn: 0.2, Rc: 0.5#Trees: 101, Precision: 0.20, Recall: 0.74, F-Score: 0.31, AUC: 0.81, G-Mean: 0.77

Noise Added to: train, Rn: 0.3, Rc: 0.1#Trees: 11, Precision: 0.21, Recall: 0.78, F-Score: 0.33, AUC: 0.82, G-Mean: 0.79

Noise Added to: train, Rn: 0.3, Rc: 0.1#Trees: 31, Precision: 0.23, Recall: 0.78, F-Score: 0.35, AUC: 0.82, G-Mean: 0.80

Noise Added to: train, Rn: 0.3, Rc: 0.1#Trees: 51, Precision: 0.22, Recall: 0.72, F-Score: 0.34, AUC: 0.80, G-Mean: 0.77

Noise Added to: train, Rn: 0.3, Rc: 0.1#Trees: 101, Precision: 0.22, Recall: 0.72, F-Score: 0.33, AUC: 0.80, G-Mean: 0.77

Noise Added to: train, Rn: 0.3, Rc: 0.2#Trees: 11, Precision: 0.22, Recall: 0.77, F-Score: 0.34, AUC: 0.83, G-Mean: 0.79

Noise Added to: train, Rn: 0.3, Rc: 0.2#Trees: 31, Precision: 0.22, Recall: 0.76, F-Score: 0.34, AUC: 0.81, G-Mean: 0.79

Noise Added to: train, Rn: 0.3, Rc: 0.2#Trees: 51, Precision: 0.21, Recall: 0.78, F-Score: 0.33, AUC: 0.82, G-Mean: 0.79

Noise Added to: train, Rn: 0.3, Rc: 0.2#Trees: 101, Precision: 0.21, Recall: 0.71, F-Score: 0.32, AUC: 0.80, G-Mean: 0.75

Noise Added to: train, Rn: 0.3, Rc: 0.3#Trees: 11, Precision: 0.22, Recall: 0.71, F-Score: 0.33, AUC: 0.82, G-Mean: 0.76

Noise Added to: train, Rn: 0.3, Rc: 0.3#Trees: 31, Precision: 0.21, Recall: 0.79, F-Score: 0.33, AUC: 0.83, G-Mean: 0.79

Noise Added to: train, Rn: 0.3, Rc: 0.3#Trees: 51, Precision: 0.21, Recall: 0.74, F-Score: 0.33, AUC: 0.80, G-Mean: 0.77

Noise Added to: train, Rn: 0.3, Rc: 0.3#Trees: 101, Precision: 0.24, Recall: 0.76, F-Score: 0.36, AUC: 0.82, G-Mean: 0.79

Noise Added to: train, Rn: 0.3, Rc: 0.4#Trees: 11, Precision: 0.21, Recall: 0.73, F-Score: 0.33, AUC: 0.80, G-Mean: 0.77

Noise Added to: train, Rn: 0.3, Rc: 0.4#Trees: 31, Precision: 0.22, Recall: 0.76, F-Score: 0.34, AUC: 0.83, G-Mean: 0.78

Noise Added to: train, Rn: 0.3, Rc: 0.4#Trees: 51, Precision: 0.19, Recall: 0.76, F-Score: 0.30, AUC: 0.80, G-Mean: 0.77

Noise Added to: train, Rn: 0.3, Rc: 0.4#Trees: 101, Precision: 0.21, Recall: 0.79, F-Score: 0.33, AUC: 0.84, G-Mean: 0.79

Noise Added to: train, Rn: 0.3, Rc: 0.5#Trees: 11, Precision: 0.22, Recall: 0.82, F-Score: 0.34, AUC: 0.83, G-Mean: 0.81

Noise Added to: train, Rn: 0.3, Rc: 0.5#Trees: 31, Precision: 0.23, Recall: 0.72, F-Score: 0.34, AUC: 0.80, G-Mean: 0.77

Noise Added to: train, Rn: 0.3, Rc: 0.5#Trees: 51, Precision: 0.23, Recall: 0.71, F-Score: 0.32, AUC: 0.80, G-Mean: 0.75

Noise Added to: train, Rn: 0.3, Rc: 0.5#Trees: 101, Precision: 0.21, Recall: 0.67, F-Score: 0.32, AUC: 0.77, G-Mean: 0.74

Noise Added to: train, Rn: 0.4, Rc: 0.1#Trees: 11, Precision: 0.20, Recall: 0.76, F-Score: 0.31, AUC: 0.80, G-Mean: 0.77

Noise Added to: train, Rn: 0.4, Rc: 0.1#Trees: 31, Precision: 0.21, Recall: 0.74, F-Score: 0.32, AUC: 0.81, G-Mean: 0.77

Noise Added to: train, Rn: 0.4, Rc: 0.1#Trees: 51, Precision: 0.19, Recall: 0.74, F-Score: 0.30, AUC: 0.80, G-Mean: 0.76

Noise Added to: train, Rn: 0.4, Rc: 0.1#Trees: 101, Precision: 0.22, Recall: 0.75, F-Score: 0.34, AUC: 0.82, G-Mean: 0.78

Noise Added to: train, Rn: 0.4, Rc: 0.2#Trees: 11, Precision: 0.20, Recall: 0.77, F-Score: 0.31, AUC: 0.79, G-Mean: 0.78

Noise Added to: train, Rn: 0.4, Rc: 0.2#Trees: 31, Precision: 0.21, Recall: 0.77, F-Score: 0.33, AUC: 0.81, G-Mean: 0.79

Noise Added to: train, Rn: 0.4, Rc: 0.2#Trees: 51, Precision: 0.21, Recall: 0.77, F-Score: 0.33, AUC: 0.82, G-Mean: 0.79

Noise Added to: train, Rn: 0.4, Rc: 0.2#Trees: 101, Precision: 0.20, Recall: 0.72, F-Score: 0.31, AUC: 0.81, G-Mean: 0.76

Noise Added to: train, Rn: 0.4, Rc: 0.3#Trees: 11, Precision: 0.20, Recall: 0.81, F-Score: 0.32, AUC: 0.81, G-Mean: 0.79

Noise Added to: train, Rn: 0.4, Rc: 0.3#Trees: 31, Precision: 0.21, Recall: 0.73, F-Score: 0.33, AUC: 0.81, G-Mean: 0.77

Noise Added to: train, Rn: 0.4, Rc: 0.3#Trees: 51, Precision: 0.21, Recall: 0.81, F-Score: 0.34, AUC: 0.82, G-Mean: 0.80

Noise Added to: train, Rn: 0.4, Rc: 0.3#Trees: 101, Precision: 0.22, Recall: 0.69, F-Score: 0.33, AUC: 0.81, G-Mean: 0.75

Noise Added to: train, Rn: 0.4, Rc: 0.4#Trees: 11, Precision: 0.20, Recall: 0.80, F-Score: 0.32, AUC: 0.82, G-Mean: 0.78

Noise Added to: train, Rn: 0.4, Rc: 0.4#Trees: 31, Precision: 0.23, Recall: 0.79, F-Score: 0.36, AUC: 0.82, G-Mean: 0.80

Noise Added to: train, Rn: 0.4, Rc: 0.4#Trees: 51, Precision: 0.21, Recall: 0.76, F-Score: 0.33, AUC: 0.81, G-Mean: 0.78

Noise Added to: train, Rn: 0.4, Rc: 0.4#Trees: 101, Precision: 0.20, Recall: 0.72, F-Score: 0.31, AUC: 0.78, G-Mean: 0.76

Noise Added to: train, Rn: 0.4, Rc: 0.5#Trees: 11, Precision: 0.20, Recall: 0.82, F-Score: 0.32, AUC: 0.82, G-Mean: 0.80

Noise Added to: train, Rn: 0.4, Rc: 0.5#Trees: 31, Precision: 0.20, Recall: 0.80, F-Score: 0.32, AUC: 0.82, G-Mean: 0.79

Noise Added to: train, Rn: 0.4, Rc: 0.5#Trees: 51, Precision: 0.21, Recall: 0.80, F-Score: 0.33, AUC: 0.83, G-Mean: 0.79

Noise Added to: train, Rn: 0.4, Rc: 0.5#Trees: 101, Precision: 0.19, Recall: 0.71, F-Score: 0.30, AUC: 0.78, G-Mean: 0.75

Noise Added to: train, Rn: 0.5, Rc: 0.1#Trees: 11, Precision: 0.20, Recall: 0.76, F-Score: 0.32, AUC: 0.81, G-Mean: 0.78

Noise Added to: train, Rn: 0.5, Rc: 0.1#Trees: 31, Precision: 0.20, Recall: 0.70, F-Score: 0.31, AUC: 0.79, G-Mean: 0.75

Noise Added to: train, Rn: 0.5, Rc: 0.1#Trees: 51, Precision: 0.20, Recall: 0.71, F-Score: 0.31, AUC: 0.79, G-Mean: 0.76

Noise Added to: train, Rn: 0.5, Rc: 0.1#Trees: 101, Precision: 0.20, Recall: 0.80, F-Score: 0.32, AUC: 0.83, G-Mean: 0.79

Noise Added to: train, Rn: 0.5, Rc: 0.2#Trees: 11, Precision: 0.20, Recall: 0.77, F-Score: 0.31, AUC: 0.80, G-Mean: 0.78

Noise Added to: train, Rn: 0.5, Rc: 0.2#Trees: 31, Precision: 0.22, Recall: 0.77, F-Score: 0.34, AUC: 0.81, G-Mean: 0.79

Noise Added to: train, Rn: 0.5, Rc: 0.2#Trees: 51, Precision: 0.20, Recall: 0.70, F-Score: 0.31, AUC: 0.79, G-Mean: 0.75

Noise Added to: train, Rn: 0.5, Rc: 0.2#Trees: 101, Precision: 0.21, Recall: 0.82, F-Score: 0.34, AUC: 0.83, G-Mean: 0.80

Noise Added to: train, Rn: 0.5, Rc: 0.3#Trees: 11, Precision: 0.23, Recall: 0.78, F-Score: 0.35, AUC: 0.82, G-Mean: 0.80

Noise Added to: train, Rn: 0.5, Rc: 0.3#Trees: 31, Precision: 0.22, Recall: 0.68, F-Score: 0.33, AUC: 0.79, G-Mean: 0.75

Noise Added to: train, Rn: 0.5, Rc: 0.3#Trees: 51, Precision: 0.21, Recall: 0.70, F-Score: 0.32, AUC: 0.78, G-Mean: 0.75

Noise Added to: train, Rn: 0.5, Rc: 0.3#Trees: 101, Precision: 0.22, Recall: 0.75, F-Score: 0.33, AUC: 0.80, G-Mean: 0.78

Noise Added to: train, Rn: 0.5, Rc: 0.4#Trees: 11, Precision: 0.22, Recall: 0.77, F-Score: 0.34, AUC: 0.81, G-Mean: 0.79

Noise Added to: train, Rn: 0.5, Rc: 0.4#Trees: 31, Precision: 0.24, Recall: 0.79, F-Score: 0.36, AUC: 0.83, G-Mean: 0.81

Noise Added to: train, Rn: 0.5, Rc: 0.4#Trees: 51, Precision: 0.20, Recall: 0.87, F-Score: 0.32, AUC: 0.83, G-Mean: 0.81

Noise Added to: train, Rn: 0.5, Rc: 0.4#Trees: 101, Precision: 0.20, Recall: 0.79, F-Score: 0.32, AUC: 0.81, G-Mean: 0.78

Noise Added to: train, Rn: 0.5, Rc: 0.5#Trees: 11, Precision: 0.20, Recall: 0.72, F-Score: 0.31, AUC: 0.79, G-Mean: 0.75

Noise Added to: train, Rn: 0.5, Rc: 0.5#Trees: 31, Precision: 0.22, Recall: 0.73, F-Score: 0.33, AUC: 0.82, G-Mean: 0.76

Noise Added to: train, Rn: 0.5, Rc: 0.5#Trees: 51, Precision: 0.22, Recall: 0.73, F-Score: 0.34, AUC: 0.82, G-Mean: 0.78

Noise Added to: train, Rn: 0.5, Rc: 0.5#Trees: 101, Precision: 0.23, Recall: 0.76, F-Score: 0.35, AUC: 0.83, G-Mean: 0.78

Noise Added to: test, Rn: 0.1, Rc: 0.1#Trees: 11, Precision: 0.21, Recall: 0.71, F-Score: 0.33, AUC: 0.79, G-Mean: 0.76

Noise Added to: test, Rn: 0.1, Rc: 0.1#Trees: 31, Precision: 0.19, Recall: 0.71, F-Score: 0.30, AUC: 0.79, G-Mean: 0.75

Noise Added to: test, Rn: 0.1, Rc: 0.1#Trees: 51, Precision: 0.21, Recall: 0.74, F-Score: 0.33, AUC: 0.82, G-Mean: 0.77

Noise Added to: test, Rn: 0.1, Rc: 0.1#Trees: 101, Precision: 0.20, Recall: 0.68, F-Score: 0.30, AUC: 0.79, G-Mean: 0.74

Noise Added to: test, Rn: 0.1, Rc: 0.2#Trees: 11, Precision: 0.21, Recall: 0.74, F-Score: 0.33, AUC: 0.81, G-Mean: 0.78

Noise Added to: test, Rn: 0.1, Rc: 0.2#Trees: 31, Precision: 0.23, Recall: 0.75, F-Score: 0.35, AUC: 0.83, G-Mean: 0.79

Noise Added to: test, Rn: 0.1, Rc: 0.2#Trees: 51, Precision: 0.20, Recall: 0.68, F-Score: 0.31, AUC: 0.78, G-Mean: 0.74

Noise Added to: test, Rn: 0.1, Rc: 0.2#Trees: 101, Precision: 0.21, Recall: 0.70, F-Score: 0.32, AUC: 0.78, G-Mean: 0.75

Noise Added to: test, Rn: 0.1, Rc: 0.3#Trees: 11, Precision: 0.22, Recall: 0.70, F-Score: 0.34, AUC: 0.80, G-Mean: 0.76

Noise Added to: test, Rn: 0.1, Rc: 0.3#Trees: 31, Precision: 0.21, Recall: 0.73, F-Score: 0.32, AUC: 0.78, G-Mean: 0.77

Noise Added to: test, Rn: 0.1, Rc: 0.3#Trees: 51, Precision: 0.23, Recall: 0.72, F-Score: 0.34, AUC: 0.80, G-Mean: 0.77

Noise Added to: test, Rn: 0.1, Rc: 0.3#Trees: 101, Precision: 0.21, Recall: 0.70, F-Score: 0.32, AUC: 0.76, G-Mean: 0.75

Noise Added to: test, Rn: 0.1, Rc: 0.4#Trees: 11, Precision: 0.20, Recall: 0.65, F-Score: 0.31, AUC: 0.76, G-Mean: 0.73

Noise Added to: test, Rn: 0.1, Rc: 0.4#Trees: 31, Precision: 0.19, Recall: 0.66, F-Score: 0.29, AUC: 0.76, G-Mean: 0.73

Noise Added to: test, Rn: 0.1, Rc: 0.4#Trees: 51, Precision: 0.23, Recall: 0.71, F-Score: 0.34, AUC: 0.81, G-Mean: 0.76

Noise Added to: test, Rn: 0.1, Rc: 0.4#Trees: 101, Precision: 0.20, Recall: 0.73, F-Score: 0.31, AUC: 0.79, G-Mean: 0.76

Noise Added to: test, Rn: 0.1, Rc: 0.5#Trees: 11, Precision: 0.21, Recall: 0.66, F-Score: 0.32, AUC: 0.78, G-Mean: 0.74

Noise Added to: test, Rn: 0.1, Rc: 0.5#Trees: 31, Precision: 0.20, Recall: 0.70, F-Score: 0.31, AUC: 0.76, G-Mean: 0.75

Noise Added to: test, Rn: 0.1, Rc: 0.5#Trees: 51, Precision: 0.21, Recall: 0.75, F-Score: 0.33, AUC: 0.79, G-Mean: 0.78

Noise Added to: test, Rn: 0.1, Rc: 0.5#Trees: 101, Precision: 0.22, Recall: 0.69, F-Score: 0.33, AUC: 0.80, G-Mean: 0.76

Noise Added to: test, Rn: 0.2, Rc: 0.1#Trees: 11, Precision: 0.20, Recall: 0.67, F-Score: 0.31, AUC: 0.77, G-Mean: 0.74

Noise Added to: test, Rn: 0.2, Rc: 0.1#Trees: 31, Precision: 0.20, Recall: 0.74, F-Score: 0.32, AUC: 0.79, G-Mean: 0.77

Noise Added to: test, Rn: 0.2, Rc: 0.1#Trees: 51, Precision: 0.23, Recall: 0.71, F-Score: 0.35, AUC: 0.82, G-Mean: 0.77

Noise Added to: test, Rn: 0.2, Rc: 0.1#Trees: 101, Precision: 0.20, Recall: 0.70, F-Score: 0.31, AUC: 0.78, G-Mean: 0.75

Noise Added to: test, Rn: 0.2, Rc: 0.2#Trees: 11, Precision: 0.23, Recall: 0.72, F-Score: 0.34, AUC: 0.80, G-Mean: 0.77

Noise Added to: test, Rn: 0.2, Rc: 0.2#Trees: 31, Precision: 0.23, Recall: 0.76, F-Score: 0.36, AUC: 0.81, G-Mean: 0.79

Noise Added to: test, Rn: 0.2, Rc: 0.2#Trees: 51, Precision: 0.19, Recall: 0.69, F-Score: 0.30, AUC: 0.79, G-Mean: 0.74

Noise Added to: test, Rn: 0.2, Rc: 0.2#Trees: 101, Precision: 0.21, Recall: 0.71, F-Score: 0.32, AUC: 0.78, G-Mean: 0.76

Noise Added to: test, Rn: 0.2, Rc: 0.3#Trees: 11, Precision: 0.20, Recall: 0.66, F-Score: 0.31, AUC: 0.76, G-Mean: 0.73

Noise Added to: test, Rn: 0.2, Rc: 0.3#Trees: 31, Precision: 0.20, Recall: 0.68, F-Score: 0.30, AUC: 0.77, G-Mean: 0.74

Noise Added to: test, Rn: 0.2, Rc: 0.3#Trees: 51, Precision: 0.20, Recall: 0.70, F-Score: 0.31, AUC: 0.77, G-Mean: 0.74

Noise Added to: test, Rn: 0.2, Rc: 0.3#Trees: 101, Precision: 0.21, Recall: 0.66, F-Score: 0.32, AUC: 0.79, G-Mean: 0.74

Noise Added to: test, Rn: 0.2, Rc: 0.4#Trees: 11, Precision: 0.21, Recall: 0.65, F-Score: 0.31, AUC: 0.78, G-Mean: 0.72

Noise Added to: test, Rn: 0.2, Rc: 0.4#Trees: 31, Precision: 0.21, Recall: 0.65, F-Score: 0.32, AUC: 0.76, G-Mean: 0.73

Noise Added to: test, Rn: 0.2, Rc: 0.4#Trees: 51, Precision: 0.20, Recall: 0.66, F-Score: 0.30, AUC: 0.74, G-Mean: 0.72

Noise Added to: test, Rn: 0.2, Rc: 0.4#Trees: 101, Precision: 0.20, Recall: 0.71, F-Score: 0.31, AUC: 0.78, G-Mean: 0.75

Noise Added to: test, Rn: 0.2, Rc: 0.5#Trees: 11, Precision: 0.19, Recall: 0.65, F-Score: 0.29, AUC: 0.75, G-Mean: 0.71

Noise Added to: test, Rn: 0.2, Rc: 0.5#Trees: 31, Precision: 0.20, Recall: 0.61, F-Score: 0.30, AUC: 0.74, G-Mean: 0.71

Noise Added to: test, Rn: 0.2, Rc: 0.5#Trees: 51, Precision: 0.20, Recall: 0.64, F-Score: 0.30, AUC: 0.75, G-Mean: 0.72

Noise Added to: test, Rn: 0.2, Rc: 0.5#Trees: 101, Precision: 0.22, Recall: 0.62, F-Score: 0.31, AUC: 0.77, G-Mean: 0.71

Noise Added to: test, Rn: 0.3, Rc: 0.1#Trees: 11, Precision: 0.20, Recall: 0.70, F-Score: 0.31, AUC: 0.79, G-Mean: 0.75

Noise Added to: test, Rn: 0.3, Rc: 0.1#Trees: 31, Precision: 0.21, Recall: 0.72, F-Score: 0.33, AUC: 0.79, G-Mean: 0.77

Noise Added to: test, Rn: 0.3, Rc: 0.1#Trees: 51, Precision: 0.19, Recall: 0.67, F-Score: 0.30, AUC: 0.78, G-Mean: 0.73

Noise Added to: test, Rn: 0.3, Rc: 0.1#Trees: 101, Precision: 0.19, Recall: 0.68, F-Score: 0.30, AUC: 0.78, G-Mean: 0.74

Noise Added to: test, Rn: 0.3, Rc: 0.2#Trees: 11, Precision: 0.21, Recall: 0.60, F-Score: 0.31, AUC: 0.74, G-Mean: 0.71

Noise Added to: test, Rn: 0.3, Rc: 0.2#Trees: 31, Precision: 0.20, Recall: 0.64, F-Score: 0.31, AUC: 0.76, G-Mean: 0.72

Noise Added to: test, Rn: 0.3, Rc: 0.2#Trees: 51, Precision: 0.20, Recall: 0.65, F-Score: 0.30, AUC: 0.76, G-Mean: 0.73

Noise Added to: test, Rn: 0.3, Rc: 0.2#Trees: 101, Precision: 0.20, Recall: 0.61, F-Score: 0.30, AUC: 0.75, G-Mean: 0.71

Noise Added to: test, Rn: 0.3, Rc: 0.3#Trees: 11, Precision: 0.18, Recall: 0.59, F-Score: 0.27, AUC: 0.73, G-Mean: 0.69

Noise Added to: test, Rn: 0.3, Rc: 0.3#Trees: 31, Precision: 0.21, Recall: 0.67, F-Score: 0.31, AUC: 0.77, G-Mean: 0.74

Noise Added to: test, Rn: 0.3, Rc: 0.3#Trees: 51, Precision: 0.20, Recall: 0.60, F-Score: 0.30, AUC: 0.74, G-Mean: 0.70

Noise Added to: test, Rn: 0.3, Rc: 0.3#Trees: 101, Precision: 0.21, Recall: 0.67, F-Score: 0.31, AUC: 0.77, G-Mean: 0.74

Noise Added to: test, Rn: 0.3, Rc: 0.4#Trees: 11, Precision: 0.18, Recall: 0.61, F-Score: 0.27, AUC: 0.73, G-Mean: 0.69

Noise Added to: test, Rn: 0.3, Rc: 0.4#Trees: 31, Precision: 0.20, Recall: 0.63, F-Score: 0.30, AUC: 0.74, G-Mean: 0.72

Noise Added to: test, Rn: 0.3, Rc: 0.4#Trees: 51, Precision: 0.20, Recall: 0.60, F-Score: 0.29, AUC: 0.73, G-Mean: 0.70

Noise Added to: test, Rn: 0.3, Rc: 0.4#Trees: 101, Precision: 0.21, Recall: 0.68, F-Score: 0.32, AUC: 0.78, G-Mean: 0.75

Noise Added to: test, Rn: 0.3, Rc: 0.5#Trees: 11, Precision: 0.20, Recall: 0.64, F-Score: 0.30, AUC: 0.74, G-Mean: 0.72

Noise Added to: test, Rn: 0.3, Rc: 0.5#Trees: 31, Precision: 0.17, Recall: 0.53, F-Score: 0.25, AUC: 0.68, G-Mean: 0.64

Noise Added to: test, Rn: 0.3, Rc: 0.5#Trees: 51, Precision: 0.18, Recall: 0.62, F-Score: 0.28, AUC: 0.72, G-Mean: 0.70

Noise Added to: test, Rn: 0.3, Rc: 0.5#Trees: 101, Precision: 0.18, Recall: 0.57, F-Score: 0.27, AUC: 0.72, G-Mean: 0.68

Noise Added to: test, Rn: 0.4, Rc: 0.1#Trees: 11, Precision: 0.21, Recall: 0.69, F-Score: 0.32, AUC: 0.79, G-Mean: 0.75

Noise Added to: test, Rn: 0.4, Rc: 0.1#Trees: 31, Precision: 0.20, Recall: 0.65, F-Score: 0.30, AUC: 0.76, G-Mean: 0.72

Noise Added to: test, Rn: 0.4, Rc: 0.1#Trees: 51, Precision: 0.23, Recall: 0.75, F-Score: 0.35, AUC: 0.80, G-Mean: 0.78

Noise Added to: test, Rn: 0.4, Rc: 0.1#Trees: 101, Precision: 0.20, Recall: 0.67, F-Score: 0.31, AUC: 0.77, G-Mean: 0.74

Noise Added to: test, Rn: 0.4, Rc: 0.2#Trees: 11, Precision: 0.19, Recall: 0.63, F-Score: 0.30, AUC: 0.75, G-Mean: 0.72

Noise Added to: test, Rn: 0.4, Rc: 0.2#Trees: 31, Precision: 0.20, Recall: 0.65, F-Score: 0.30, AUC: 0.76, G-Mean: 0.73

Noise Added to: test, Rn: 0.4, Rc: 0.2#Trees: 51, Precision: 0.21, Recall: 0.65, F-Score: 0.31, AUC: 0.76, G-Mean: 0.73

Noise Added to: test, Rn: 0.4, Rc: 0.2#Trees: 101, Precision: 0.20, Recall: 0.63, F-Score: 0.30, AUC: 0.76, G-Mean: 0.72

Noise Added to: test, Rn: 0.4, Rc: 0.3#Trees: 11, Precision: 0.21, Recall: 0.61, F-Score: 0.31, AUC: 0.74, G-Mean: 0.71

Noise Added to: test, Rn: 0.4, Rc: 0.3#Trees: 31, Precision: 0.21, Recall: 0.64, F-Score: 0.31, AUC: 0.76, G-Mean: 0.72

Noise Added to: test, Rn: 0.4, Rc: 0.3#Trees: 51, Precision: 0.19, Recall: 0.61, F-Score: 0.29, AUC: 0.73, G-Mean: 0.70

Noise Added to: test, Rn: 0.4, Rc: 0.3#Trees: 101, Precision: 0.19, Recall: 0.52, F-Score: 0.28, AUC: 0.72, G-Mean: 0.66

Noise Added to: test, Rn: 0.4, Rc: 0.4#Trees: 11, Precision: 0.20, Recall: 0.69, F-Score: 0.30, AUC: 0.76, G-Mean: 0.74

Noise Added to: test, Rn: 0.4, Rc: 0.4#Trees: 31, Precision: 0.19, Recall: 0.53, F-Score: 0.27, AUC: 0.68, G-Mean: 0.66

Noise Added to: test, Rn: 0.4, Rc: 0.4#Trees: 51, Precision: 0.18, Recall: 0.52, F-Score: 0.26, AUC: 0.67, G-Mean: 0.65

Noise Added to: test, Rn: 0.4, Rc: 0.4#Trees: 101, Precision: 0.19, Recall: 0.53, F-Score: 0.27, AUC: 0.68, G-Mean: 0.65

Noise Added to: test, Rn: 0.4, Rc: 0.5#Trees: 11, Precision: 0.17, Recall: 0.45, F-Score: 0.24, AUC: 0.70, G-Mean: 0.59

Noise Added to: test, Rn: 0.4, Rc: 0.5#Trees: 31, Precision: 0.19, Recall: 0.60, F-Score: 0.28, AUC: 0.72, G-Mean: 0.69

Noise Added to: test, Rn: 0.4, Rc: 0.5#Trees: 51, Precision: 0.19, Recall: 0.55, F-Score: 0.28, AUC: 0.69, G-Mean: 0.66

Noise Added to: test, Rn: 0.4, Rc: 0.5#Trees: 101, Precision: 0.17, Recall: 0.52, F-Score: 0.25, AUC: 0.65, G-Mean: 0.64

Noise Added to: test, Rn: 0.5, Rc: 0.1#Trees: 11, Precision: 0.20, Recall: 0.67, F-Score: 0.30, AUC: 0.76, G-Mean: 0.74

Noise Added to: test, Rn: 0.5, Rc: 0.1#Trees: 31, Precision: 0.20, Recall: 0.67, F-Score: 0.30, AUC: 0.77, G-Mean: 0.73

Noise Added to: test, Rn: 0.5, Rc: 0.1#Trees: 51, Precision: 0.21, Recall: 0.69, F-Score: 0.32, AUC: 0.77, G-Mean: 0.75

Noise Added to: test, Rn: 0.5, Rc: 0.1#Trees: 101, Precision: 0.19, Recall: 0.67, F-Score: 0.29, AUC: 0.77, G-Mean: 0.73

Noise Added to: test, Rn: 0.5, Rc: 0.2#Trees: 11, Precision: 0.21, Recall: 0.69, F-Score: 0.32, AUC: 0.76, G-Mean: 0.75

Noise Added to: test, Rn: 0.5, Rc: 0.2#Trees: 31, Precision: 0.21, Recall: 0.66, F-Score: 0.32, AUC: 0.77, G-Mean: 0.74

Noise Added to: test, Rn: 0.5, Rc: 0.2#Trees: 51, Precision: 0.21, Recall: 0.58, F-Score: 0.30, AUC: 0.71, G-Mean: 0.69

Noise Added to: test, Rn: 0.5, Rc: 0.2#Trees: 101, Precision: 0.20, Recall: 0.66, F-Score: 0.30, AUC: 0.77, G-Mean: 0.73

Noise Added to: test, Rn: 0.5, Rc: 0.3#Trees: 11, Precision: 0.19, Recall: 0.60, F-Score: 0.29, AUC: 0.73, G-Mean: 0.69

Noise Added to: test, Rn: 0.5, Rc: 0.3#Trees: 31, Precision: 0.20, Recall: 0.54, F-Score: 0.29, AUC: 0.71, G-Mean: 0.67

Noise Added to: test, Rn: 0.5, Rc: 0.3#Trees: 51, Precision: 0.18, Recall: 0.57, F-Score: 0.27, AUC: 0.69, G-Mean: 0.68

Noise Added to: test, Rn: 0.5, Rc: 0.3#Trees: 101, Precision: 0.20, Recall: 0.55, F-Score: 0.29, AUC: 0.72, G-Mean: 0.68

Noise Added to: test, Rn: 0.5, Rc: 0.4#Trees: 11, Precision: 0.15, Recall: 0.55, F-Score: 0.24, AUC: 0.69, G-Mean: 0.65

Noise Added to: test, Rn: 0.5, Rc: 0.4#Trees: 31, Precision: 0.19, Recall: 0.49, F-Score: 0.26, AUC: 0.68, G-Mean: 0.63

Noise Added to: test, Rn: 0.5, Rc: 0.4#Trees: 51, Precision: 0.20, Recall: 0.64, F-Score: 0.30, AUC: 0.73, G-Mean: 0.73

Noise Added to: test, Rn: 0.5, Rc: 0.4#Trees: 101, Precision: 0.16, Recall: 0.59, F-Score: 0.25, AUC: 0.67, G-Mean: 0.68

Noise Added to: test, Rn: 0.5, Rc: 0.5#Trees: 11, Precision: 0.16, Recall: 0.53, F-Score: 0.23, AUC: 0.69, G-Mean: 0.63

Noise Added to: test, Rn: 0.5, Rc: 0.5#Trees: 31, Precision: 0.18, Recall: 0.48, F-Score: 0.25, AUC: 0.69, G-Mean: 0.62

Noise Added to: test, Rn: 0.5, Rc: 0.5#Trees: 51, Precision: 0.22, Recall: 0.70, F-Score: 0.32, AUC: 0.77, G-Mean: 0.75

Noise Added to: test, Rn: 0.5, Rc: 0.5#Trees: 101, Precision: 0.16, Recall: 0.54, F-Score: 0.23, AUC: 0.66, G-Mean: 0.64

Adaboost

```
def get_a_balanced_set(x, y):  
    mjr_x = x[y == -1]  
    mjr_y = y[y == -1]
```

```
mnr_x = x[y == 1]
mnr_y = y[y == 1]
```

داده‌های مثبت را در نظر گرفته و مثبت‌ها را از منفی‌ها جدا کرده

```
ind = [i for i in range(mjr_x.shape[0])]
r = np.random.choice(ind, size=mnr_x.shape[0], replace=False)
```

اندیس تعدادی از داده‌های ماینر را مساوی تعدادی از داده‌های میجر انتخاب میکند به صورت غیر تکراری

```
x_bs = np.concatenate((mjr_x[r], mnr_x), axis = 0)
y_bs = np.concatenate((mjr_y[r], mnr_y), axis = 0)
```

```
return x_bs, y_bs
```

و آنها را به هم می‌چسباند و داده x, y را برایمان می‌سازد. ورودی x و خروجی y

```
def train_adaboost(x, y, n_trees, max_depth):
```

تابع `train_adaboost` را داریم که فقط یک `adaboost` را برایمان `train` میکند.

تعداد داده‌های آموزشی (x, y) تعداد درخت و عمق درخت

```
n_samples = x.shape[0]
```

تعداد داده‌های آموزشی را بدست آوردم

```
d = np.ones((n_samples,)) / n_samples
```

مقدار اولیه d را ساختم که ضریب هر کدام از داده‌ها را نشان میدهد که در ابتدا همه با هم برابر 1 است

```
tree_list = []
alpha = np.zeros((n_trees))
```

لیست درختها و α ضریب هر درخت

```
inds = [i for i in range(n_samples)]
```

اندیس تمام داده‌ها از 0 تا n

```
for t in range(n_trees):
    sel_inds = np.random.choice(inds, size=n_samples, replace=True, p=d)
```

در طول تکرار ها کاری که انجام میدم به صورت تصادفی داده‌هایی را انتخاب میکنم با توزیع d و اجازه جایگذاری هم وجود داره برای اینکه هر داده‌ای که احتمال بیشتری داشته باشد نمونه‌های بیشتری از آن انتخاب خواهد شد.

```
tree = DecisionTreeClassifier(max_depth=max_depth)
tree.fit(x[sel_inds], y[sel_inds])
```

با استفاده از داده‌های انتخاب شده یک `DecisionTreeClassifier` ایجاد میشود و با استفاده از اون داده‌ها آموزش داده میشود.

```
l = tree.predict(x)
match = (l != y)
err = (d * match).sum()
```

لیبل داده‌ها بدست می‌آید

آنهايي که ليبل يکسان ندارند را بدست مي‌ايريم

ضرب در d کرده و جمع خطا را بدست می‌آوریم.

```
if err > 0.5:
```

```
    return False
```

اگر جمع از 5. بیشتر شد از تابع خارج میشیم که دوباره تابعی که آن را صدا زده باید عمق بیشتری به درخت بدهد تا بتواند کار ادامه یابد. چون خطای بیشتر از نصف کلاسیفایر درست پیشبینی نمیکند.

```
alpha[t] = 0.5 * np.log((1 - err) / err)
```

```
tree_list.append(tree)
```

مقدار α تکرار بدست میاد و درخت به لیت درختها اضافه میشود.

```
d = d * np.exp(-alpha[t] * l * y)
```

```
d /= d.sum()
```

```
return tree_list, alpha
```

update مقدار d بر اساس فرمول در اسلایدها و نرمال شدن برای داشتن حالت توزیع احتمالی و

درنهایت باز گرداندن لیست درخت احتمالی و α

```
def predict_adaboost(tree_list, alpha, x):
```

تابع predict_adaboost لیست درخت ها، α و چیزی که باید برایش پیشبینی ها را انجام دهیم دریافت میکند.

```
s = alpha[0] * tree_list[0].predict(x)
```

```
for i in range(1, len(tree_list)):
```

```
    s += alpha[i] * tree_list[i].predict(x)
```

پیشبینی درختها ضرب در α و جمع با s

```
label = np.sign(s)
```

```
p = 1. / (1 + np.exp(-2 * s))
```

```
return label, p
```

در نهایت لیبل میشود علامت s برای هر کدام از داده ها

و چون احتمال را نیاز داریم با استفاده از تابع لاجستیک رگرسیون تبدیل به مقدار احتمال میکنیم خروجی کلاسیفایر s را.

و داده ها را به عنوان خروجی باز میگردانیم.

```
def report_measures_for_adaboost(x, y, n_trees, n_ensembles, trn_ratio):
```

معیارها را در adaboost حساب میکنیم.

x, y داده های آموزشی n_tree تعداد درخت $n_ensembles$ تعداد adaboost هایی که ساخته میشوند که در صورت تمرین

$n_trees = S$ و $n_ensembles = T$ است و تعداد داده های آموزشی trn_ratio

```
trn_x, tst_x, trn_y, tst_y = train_test_split(x, y, train_size = trn_ratio, stratify = y)
```

```
all_trees = []
```

```
all_alpha = []
```

تقسیم داده ها

باید تمام ensemble ها را ایجاد کرد و درختهای رو کنار هم قرار داد و با استفاده از آن پیشبینی را انجام داد

```
for _ in range(n_ensembles):
```

```
    bs_x, bs_y = get_a_balanced_set(trn_x, trn_y)
```

```
    max_depth = 1
```

بنابر این به تعداد $n_ensembles$ بار کاری که انجام میدم

get_a_balanced از داده ها آموزشی میگیرم

و مقدار عمق را 1 قرار میدهم و سعی میکنم ببینم با این عمق کار به درستی انجام میشود یا نه

```
while True:
    out = train_adaboost(bs_x, bs_y, n_trees, max_depth)
    if out:
        ada_trees, alpha = out
        break
```

یعنی با عمق 1 adaboost با آن داده هایی که بالانس هستند (bs_x, bs_y, ...) آموزش میدم اگر درست درآمد یعنی flase برگشت نشد هم درخت ها و هم آلفا را خواهم داشت

و از حلقه خارج میشوم

```
max_depth += 1
```

وگرنه به مقدار عمق یکی اضافه میکنم و این کار را تکرار میکنم تا عمق کافی باشد و این خطای درختم از 50 درصد بیشتر نشود.

```
if max_depth > 10:
```

اگر عمق را تا مقدار 10 افزایش دادم و اتفاقی نیفتاد

```
bs_x, bs_y = get_a_balanced_set(trn_x, trn_y)
max_depth = 1
```

سعی میکنم داده آموزشی ام را عوض کنم شاید اشکال از آن باشد عمق را 1 قرار میدهم و حلقه را دوباره تکرار میکنم.

```
all_trees.extend(ada_trees)
all_alpha.extend(alpha)
```

در نهایت درخت های ایجاد شده را به لیست درختها اضافه میکنم و alpha را هم اضافه میکنم.

```
prd, p = predict_adaboost(all_trees, all_alpha, tst_x)
```

تابع predict_adaboost را صدا میزنم با تمامی درختهای ensembles ها و آلفاها و داده های تست (tst_x) را بش میدم و مقادیر خروجی را برام حساب میکنه

```
prc, recall, fscore, _ = precision_recall_fscore_support(tst_y, prd, labels=[1])
auc = roc_auc_score(tst_y, p)
g_mean = geometric_mean_score(tst_y, prd)
return prc[0], recall[0], fscore[0], auc, g_mean
```

و این مقادیر (precision, recall, fscore, auc, g_mean) را نیز حساب کرده و باز میگرداند

```
file_name = 'Covid-19.csv'
trn_ratio = 0.7
x, y = load_data(file_name)
n_itr = 10
```

```
for ensemble_size in [10, 15]:
    for n_trees in [11, 31, 51, 101]:
```

اشاره شده بود مقادیر ensemble ها را 10 و 15 قرار دهیم و مقادیر هر کدام 11, 31, 51, 101

```
prc, recall, fscore, auc, g_mean = (0, 0, 0, 0, 0)
for i in range(n_itr):
    p, r, f, a, g = report_measures_for_adaboost(x, y, n_trees, ensemble_size, trn_ratio)
    prc += p
    recall += r
    fscore += f
    auc += a
```

```

g_mean += g

prc /= n_itr
recall /= n_itr
fscore /= n_itr
auc /= n_itr
g_mean /= n_itr

```

```

print('ensemble-size: {}, #Trees: {}, Precision: {:.2f}, Recall: {:.2f}, F-Score: {:.2f}, AUC: {:.2f}, G-Mean: {:.2f}'.format(ensemble_size, n_trees, prc, recall, fscore, auc, g_mean))

```

اندازه ensemble و دیگر مقادیر چاپ میشوند.

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.95, F-Score: 0.31, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 31, Precision: 0.21, Recall: 0.92, F-Score: 0.34, AUC: 0.90, G-Mean: 0.84

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 51, Precision: 0.19, Recall: 0.90, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.92, F-Score: 0.31, AUC: 0.90, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 31, Precision: 0.21, Recall: 0.93, F-Score: 0.34, AUC: 0.89, G-Mean: 0.84

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 51, Precision: 0.21, Recall: 0.95, F-Score: 0.34, AUC: 0.89, G-Mean: 0.85

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

```
p = 1. / (1 + np.exp(-2 * s))
```

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

```
p = 1. / (1 + np.exp(-2 * s))
```

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

```
p = 1. / (1 + np.exp(-2 * s))
```

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

```
p = 1. / (1 + np.exp(-2 * s))
```

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.91, F-Score: 0.32, AUC: 0.86, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.91, F-Score: 0.31, AUC: 0.89, G-Mean: 0.81

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 31, Precision: 0.19, Recall: 0.92, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 51, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 11, Precision: 0.18, Recall: 0.94, F-Score: 0.31, AUC: 0.90, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 31, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 51, Precision: 0.21, Recall: 0.91, F-Score: 0.34, AUC: 0.88, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.94, F-Score: 0.33, AUC: 0.86, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.3, #Trees: 11, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.3, #Trees: 31, Precision: 0.19, Recall: 0.96, F-Score: 0.32, AUC: 0.91, G-Mean: 0.84

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.3, #Trees: 51, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.3, #Trees: 101, Precision: 0.19, Recall: 0.90, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.3, #Trees: 11, Precision: 0.19, Recall: 0.92, F-Score: 0.31, AUC: 0.90, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.3, #Trees: 31, Precision: 0.20, Recall: 0.94, F-Score: 0.33, AUC: 0.90, G-Mean: 0.84

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.3, #Trees: 51, Precision: 0.19, Recall: 0.91, F-Score: 0.31, AUC: 0.87, G-Mean: 0.82

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + \text{np.exp}(-2 * s))$$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.3, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.86, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.97, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 31, Precision: 0.20, Recall: 0.94, F-Score: 0.34, AUC: 0.90, G-Mean: 0.84

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 51, Precision: 0.19, Recall: 0.94, F-Score: 0.31, AUC: 0.87, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.88, F-Score: 0.33, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.93, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 31, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 51, Precision: 0.19, Recall: 0.90, F-Score: 0.31, AUC: 0.87, G-Mean: 0.82

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.88, F-Score: 0.32, AUC: 0.86, G-Mean: 0.81

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 11, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.91, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 31, Precision: 0.20, Recall: 0.94, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 51, Precision: 0.21, Recall: 0.93, F-Score: 0.34, AUC: 0.88, G-Mean: 0.84

ensemble size: 10, Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 11, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.90, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 31, Precision: 0.19, Recall: 0.89, F-Score: 0.31, AUC: 0.88, G-Mean: 0.81

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 51, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + \text{np.exp}(-2 * s))$$

ensemble size: 15, Noise Added to: train, Rn: 0.1, Rc: 0.5, #Trees: 101, Precision: 0.21, Recall: 0.91, F-Score: 0.34, AUC: 0.87, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.91, F-Score: 0.31, AUC: 0.89, G-Mean: 0.81

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 31, Precision: 0.20, Recall: 0.91, F-Score: 0.32, AUC: 0.90, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.89, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 31, Precision: 0.20, Recall: 0.91, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.89, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.87, G-Mean: 0.84

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.93, F-Score: 0.31, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 31, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.90, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 51, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.91, F-Score: 0.32, AUC: 0.90, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 31, Precision: 0.19, Recall: 0.93, F-Score: 0.31, AUC: 0.88, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 51, Precision: 0.21, Recall: 0.92, F-Score: 0.34, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.92, F-Score: 0.32, AUC: 0.87, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.3, #Trees: 11, Precision: 0.19, Recall: 0.93, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.3, #Trees: 31, Precision: 0.19, Recall: 0.90, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.3, #Trees: 51, Precision: 0.21, Recall: 0.89, F-Score: 0.34, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.3, #Trees: 101, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.3, #Trees: 11, Precision: 0.18, Recall: 0.96, F-Score: 0.31, AUC: 0.90, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.3, #Trees: 31, Precision: 0.20, Recall: 0.91, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.3, #Trees: 51, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.3, #Trees: 101, Precision: 0.21, Recall: 0.93, F-Score: 0.34, AUC: 0.87, G-Mean: 0.84

C:/Users/Hosseini/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + \text{np.exp}(-2 * s))$$

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.97, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 31, Precision: 0.20, Recall: 0.94, F-Score: 0.33, AUC: 0.91, G-Mean: 0.84

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 51, Precision: 0.20, Recall: 0.88, F-Score: 0.33, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.94, F-Score: 0.33, AUC: 0.87, G-Mean: 0.84

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.93, F-Score: 0.31, AUC: 0.90, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 31, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 51, Precision: 0.21, Recall: 0.89, F-Score: 0.34, AUC: 0.88, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.88, F-Score: 0.33, AUC: 0.87, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.5, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.31, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.5, #Trees: 31, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.90, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.5, #Trees: 51, Precision: 0.20, Recall: 0.95, F-Score: 0.33, AUC: 0.88, G-Mean: 0.84

ensemble size: 10, Noise Added to: train, Rn: 0.2, Rc: 0.5, #Trees: 101, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.86, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.5, #Trees: 11, Precision: 0.20, Recall: 0.94, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.5, #Trees: 31, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.5, #Trees: 51, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.2, Rc: 0.5, #Trees: 101, Precision: 0.20, Recall: 0.88, F-Score: 0.32, AUC: 0.86, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.1, #Trees: 31, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.87, F-Score: 0.32, AUC: 0.88, G-Mean: 0.81

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.1, #Trees: 31, Precision: 0.20, Recall: 0.91, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.1, #Trees: 101, Precision: 0.21, Recall: 0.91, F-Score: 0.34, AUC: 0.86, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.93, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.2, #Trees: 31, Precision: 0.20, Recall: 0.88, F-Score: 0.33, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.2, #Trees: 51, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.93, F-Score: 0.31, AUC: 0.90, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.2, #Trees: 31, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.2, #Trees: 51, Precision: 0.21, Recall: 0.92, F-Score: 0.34, AUC: 0.88, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.3, #Trees: 11, Precision: 0.18, Recall: 0.92, F-Score: 0.30, AUC: 0.89, G-Mean: 0.81

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.3, #Trees: 31, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.3, #Trees: 51, Precision: 0.21, Recall: 0.94, F-Score: 0.34, AUC: 0.88, G-Mean: 0.84

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.3, #Trees: 101, Precision: 0.21, Recall: 0.87, F-Score: 0.33, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.3, #Trees: 11, Precision: 0.19, Recall: 0.93, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.3, #Trees: 31, Precision: 0.19, Recall: 0.92, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.3, #Trees: 51, Precision: 0.21, Recall: 0.88, F-Score: 0.34, AUC: 0.88, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.3, #Trees: 101, Precision: 0.21, Recall: 0.91, F-Score: 0.34, AUC: 0.87, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.4, #Trees: 11, Precision: 0.20, Recall: 0.94, F-Score: 0.33, AUC: 0.91, G-Mean: 0.84

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.4, #Trees: 31, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.4, #Trees: 51, Precision: 0.20, Recall: 0.87, F-Score: 0.33, AUC: 0.88, G-Mean: 0.81

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.4, #Trees: 11, Precision: 0.18, Recall: 0.94, F-Score: 0.31, AUC: 0.90, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.4, #Trees: 31, Precision: 0.19, Recall: 0.92, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.4, #Trees: 51, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + \text{np.exp}(-2 * s))$$

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.5, #Trees: 11, Precision: 0.19, Recall: 0.89, F-Score: 0.31, AUC: 0.89, G-Mean: 0.81

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.5, #Trees: 31, Precision: 0.19, Recall: 0.94, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.5, #Trees: 51, Precision: 0.20, Recall: 0.88, F-Score: 0.32, AUC: 0.88, G-Mean: 0.81

ensemble size: 10, Noise Added to: train, Rn: 0.3, Rc: 0.5, #Trees: 101, Precision: 0.21, Recall: 0.87, F-Score: 0.34, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.5, #Trees: 11, Precision: 0.19, Recall: 0.91, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.5, #Trees: 31, Precision: 0.20, Recall: 0.94, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.5, #Trees: 51, Precision: 0.21, Recall: 0.86, F-Score: 0.33, AUC: 0.87, G-Mean: 0.81

ensemble size: 15, Noise Added to: train, Rn: 0.3, Rc: 0.5, #Trees: 101, Precision: 0.19, Recall: 0.91, F-Score: 0.31, AUC: 0.86, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.1, #Trees: 11, Precision: 0.20, Recall: 0.94, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.1, #Trees: 31, Precision: 0.19, Recall: 0.95, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.1, #Trees: 51, Precision: 0.21, Recall: 0.93, F-Score: 0.34, AUC: 0.89, G-Mean: 0.84

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.92, F-Score: 0.32, AUC: 0.87, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.1, #Trees: 11, Precision: 0.20, Recall: 0.93, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.1, #Trees: 31, Precision: 0.19, Recall: 0.94, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + \text{np.exp}(-2 * s))$$

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.1, #Trees: 101, Precision: 0.21, Recall: 0.91, F-Score: 0.34, AUC: 0.87, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.2, #Trees: 11, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.2, #Trees: 31, Precision: 0.19, Recall: 0.90, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.2, #Trees: 51, Precision: 0.19, Recall: 0.90, F-Score: 0.31, AUC: 0.88, G-Mean: 0.81

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.2, #Trees: 11, Precision: 0.18, Recall: 0.94, F-Score: 0.30, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.2, #Trees: 31, Precision: 0.20, Recall: 0.95, F-Score: 0.33, AUC: 0.90, G-Mean: 0.84

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.2, #Trees: 51, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + \text{np.exp}(-2 * s))$$

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.3, #Trees: 11, Precision: 0.19, Recall: 0.93, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.3, #Trees: 31, Precision: 0.19, Recall: 0.90, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.3, #Trees: 51, Precision: 0.20, Recall: 0.91, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.3, #Trees: 101, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.3, #Trees: 11, Precision: 0.18, Recall: 0.93, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.3, #Trees: 31, Precision: 0.19, Recall: 0.90, F-Score: 0.31, AUC: 0.89, G-Mean: 0.81

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.3, #Trees: 51, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.3, #Trees: 101, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.4, #Trees: 31, Precision: 0.20, Recall: 0.85, F-Score: 0.32, AUC: 0.89, G-Mean: 0.80

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.4, #Trees: 51, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.4, #Trees: 101, Precision: 0.19, Recall: 0.88, F-Score: 0.32, AUC: 0.86, G-Mean: 0.81

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.92, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.4, #Trees: 31, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.4, #Trees: 51, Precision: 0.20, Recall: 0.89, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + \text{np.exp}(-2 * s))$$

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.93, F-Score: 0.32, AUC: 0.86, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.5, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.5, #Trees: 31, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.5, #Trees: 51, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.4, Rc: 0.5, #Trees: 101, Precision: 0.20, Recall: 0.89, F-Score: 0.32, AUC: 0.86, G-Mean: 0.81

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.5, #Trees: 11, Precision: 0.19, Recall: 0.92, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.5, #Trees: 31, Precision: 0.20, Recall: 0.87, F-Score: 0.32, AUC: 0.88, G-Mean: 0.80

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.5, #Trees: 51, Precision: 0.21, Recall: 0.93, F-Score: 0.34, AUC: 0.89, G-Mean: 0.84

ensemble size: 15, Noise Added to: train, Rn: 0.4, Rc: 0.5, #Trees: 101, Precision: 0.20, Recall: 0.94, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.1, #Trees: 11, Precision: 0.18, Recall: 0.91, F-Score: 0.31, AUC: 0.89, G-Mean: 0.81

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.1, #Trees: 31, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.93, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.1, #Trees: 31, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.92, F-Score: 0.32, AUC: 0.87, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + \text{np.exp}(-2 * s))$$

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.95, F-Score: 0.33, AUC: 0.87, G-Mean: 0.84

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.92, F-Score: 0.32, AUC: 0.90, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.2, #Trees: 31, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.2, #Trees: 51, Precision: 0.19, Recall: 0.95, F-Score: 0.32, AUC: 0.89, G-Mean: 0.84

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.2, #Trees: 11, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.2, #Trees: 31, Precision: 0.19, Recall: 0.92, F-Score: 0.32, AUC: 0.90, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.2, #Trees: 51, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.86, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.3, #Trees: 11, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.3, #Trees: 31, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.3, #Trees: 51, Precision: 0.21, Recall: 0.88, F-Score: 0.34, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.3, #Trees: 101, Precision: 0.20, Recall: 0.95, F-Score: 0.33, AUC: 0.87, G-Mean: 0.84

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.3, #Trees: 11, Precision: 0.19, Recall: 0.91, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.3, #Trees: 31, Precision: 0.21, Recall: 0.92, F-Score: 0.34, AUC: 0.90, G-Mean: 0.84

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.3, #Trees: 51, Precision: 0.21, Recall: 0.89, F-Score: 0.34, AUC: 0.88, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.3, #Trees: 101, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.87, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.32, AUC: 0.90, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.4, #Trees: 31, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.4, #Trees: 51, Precision: 0.21, Recall: 0.89, F-Score: 0.34, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.95, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.4, #Trees: 31, Precision: 0.21, Recall: 0.93, F-Score: 0.34, AUC: 0.89, G-Mean: 0.84

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.4, #Trees: 51, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.85, F-Score: 0.33, AUC: 0.87, G-Mean: 0.81

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.5, #Trees: 11, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.5, #Trees: 31, Precision: 0.19, Recall: 0.91, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.5, #Trees: 51, Precision: 0.20, Recall: 0.86, F-Score: 0.33, AUC: 0.88, G-Mean: 0.81

ensemble size: 10, Noise Added to: train, Rn: 0.5, Rc: 0.5, #Trees: 101, Precision: 0.21, Recall: 0.84, F-Score: 0.34, AUC: 0.88, G-Mean: 0.81

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.5, #Trees: 11, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.88, G-Mean: 0.81

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.5, #Trees: 31, Precision: 0.20, Recall: 0.88, F-Score: 0.32, AUC: 0.89, G-Mean: 0.81

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.5, #Trees: 51, Precision: 0.19, Recall: 0.91, F-Score: 0.31, AUC: 0.86, G-Mean: 0.82

ensemble size: 15, Noise Added to: train, Rn: 0.5, Rc: 0.5, #Trees: 101, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.1, #Trees: 11, Precision: 0.18, Recall: 0.93, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.1, #Trees: 31, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.1, #Trees: 31, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + \text{np.exp}(-2 * s))$$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.93, F-Score: 0.34, AUC: 0.86, G-Mean: 0.84

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.92, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.2, #Trees: 31, Precision: 0.20, Recall: 0.92, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.2, #Trees: 51, Precision: 0.19, Recall: 0.91, F-Score: 0.31, AUC: 0.86, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.86, F-Score: 0.32, AUC: 0.87, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.90, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.2, #Trees: 31, Precision: 0.20, Recall: 0.94, F-Score: 0.33, AUC: 0.89, G-Mean: 0.84

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.2, #Trees: 51, Precision: 0.20, Recall: 0.94, F-Score: 0.33, AUC: 0.87, G-Mean: 0.84

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.3, #Trees: 11, Precision: 0.20, Recall: 0.96, F-Score: 0.33, AUC: 0.90, G-Mean: 0.84

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.3, #Trees: 31, Precision: 0.21, Recall: 0.91, F-Score: 0.33, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.3, #Trees: 51, Precision: 0.20, Recall: 0.87, F-Score: 0.33, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.3, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.3, #Trees: 11, Precision: 0.19, Recall: 0.95, F-Score: 0.31, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.3, #Trees: 31, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.3, #Trees: 51, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.3, #Trees: 101, Precision: 0.21, Recall: 0.92, F-Score: 0.34, AUC: 0.87, G-Mean: 0.84

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.96, F-Score: 0.31, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.4, #Trees: 31, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.4, #Trees: 51, Precision: 0.21, Recall: 0.90, F-Score: 0.34, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.92, F-Score: 0.32, AUC: 0.86, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.97, F-Score: 0.31, AUC: 0.90, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.4, #Trees: 31, Precision: 0.21, Recall: 0.93, F-Score: 0.34, AUC: 0.90, G-Mean: 0.84

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.4, #Trees: 51, Precision: 0.20, Recall: 0.92, F-Score: 0.32, AUC: 0.87, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.88, F-Score: 0.32, AUC: 0.86, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.5, #Trees: 11, Precision: 0.18, Recall: 0.94, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.5, #Trees: 31, Precision: 0.21, Recall: 0.86, F-Score: 0.33, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.5, #Trees: 51, Precision: 0.20, Recall: 0.87, F-Score: 0.33, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.1, Rc: 0.5, #Trees: 101, Precision: 0.20, Recall: 0.86, F-Score: 0.33, AUC: 0.87, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.5, #Trees: 11, Precision: 0.19, Recall: 0.95, F-Score: 0.31, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.5, #Trees: 31, Precision: 0.19, Recall: 0.90, F-Score: 0.31, AUC: 0.88, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.5, #Trees: 51, Precision: 0.19, Recall: 0.90, F-Score: 0.31, AUC: 0.87, G-Mean: 0.82

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.1, Rc: 0.5, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.86, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.31, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.1, #Trees: 31, Precision: 0.19, Recall: 0.92, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.1, #Trees: 51, Precision: 0.21, Recall: 0.89, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.90, F-Score: 0.31, AUC: 0.89, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.1, #Trees: 31, Precision: 0.20, Recall: 0.91, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.1, #Trees: 51, Precision: 0.19, Recall: 0.93, F-Score: 0.32, AUC: 0.87, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.86, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.95, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.2, #Trees: 31, Precision: 0.21, Recall: 0.89, F-Score: 0.34, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.2, #Trees: 51, Precision: 0.20, Recall: 0.94, F-Score: 0.33, AUC: 0.89, G-Mean: 0.84

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.2, #Trees: 31, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.2, #Trees: 51, Precision: 0.21, Recall: 0.92, F-Score: 0.34, AUC: 0.88, G-Mean: 0.84

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.86, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.3, #Trees: 11, Precision: 0.19, Recall: 0.96, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.3, #Trees: 31, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.3, #Trees: 51, Precision: 0.21, Recall: 0.91, F-Score: 0.34, AUC: 0.90, G-Mean: 0.84

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.3, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.86, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.3, #Trees: 11, Precision: 0.19, Recall: 0.90, F-Score: 0.31, AUC: 0.89, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.3, #Trees: 31, Precision: 0.20, Recall: 0.91, F-Score: 0.32, AUC: 0.88, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.3, #Trees: 51, Precision: 0.20, Recall: 0.91, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.3, #Trees: 101, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.86, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.92, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.4, #Trees: 31, Precision: 0.19, Recall: 0.90, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.4, #Trees: 51, Precision: 0.19, Recall: 0.90, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.96, F-Score: 0.32, AUC: 0.90, G-Mean: 0.84

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.4, #Trees: 31, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.4, #Trees: 51, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + np.exp(-2 * s))$$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + np.exp(-2 * s))$$

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.86, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.5, #Trees: 11, Precision: 0.18, Recall: 0.93, F-Score: 0.31, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.5, #Trees: 31, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.5, #Trees: 51, Precision: 0.19, Recall: 0.83, F-Score: 0.31, AUC: 0.87, G-Mean: 0.79

ensemble size: 10, Noise Added to: test, Rn: 0.2, Rc: 0.5, #Trees: 101, Precision: 0.20, Recall: 0.87, F-Score: 0.32, AUC: 0.87, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.5, #Trees: 11, Precision: 0.19, Recall: 0.96, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.5, #Trees: 31, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.5, #Trees: 51, Precision: 0.21, Recall: 0.80, F-Score: 0.34, AUC: 0.87, G-Mean: 0.80

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.2, Rc: 0.5, #Trees: 101, Precision: 0.20, Recall: 0.82, F-Score: 0.33, AUC: 0.85, G-Mean: 0.80

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.93, F-Score: 0.31, AUC: 0.90, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.1, #Trees: 31, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.1, #Trees: 101, Precision: 0.21, Recall: 0.90, F-Score: 0.34, AUC: 0.87, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.92, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.1, #Trees: 31, Precision: 0.19, Recall: 0.91, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.92, F-Score: 0.32, AUC: 0.88, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.94, F-Score: 0.33, AUC: 0.87, G-Mean: 0.84

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.2, #Trees: 31, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.2, #Trees: 51, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.86, F-Score: 0.32, AUC: 0.86, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.2, #Trees: 11, Precision: 0.20, Recall: 0.94, F-Score: 0.33, AUC: 0.90, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.2, #Trees: 31, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.2, #Trees: 51, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.2, #Trees: 101, Precision: 0.21, Recall: 0.87, F-Score: 0.33, AUC: 0.86, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.3, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.3, #Trees: 31, Precision: 0.21, Recall: 0.87, F-Score: 0.33, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.3, #Trees: 51, Precision: 0.20, Recall: 0.88, F-Score: 0.32, AUC: 0.87, G-Mean: 0.81

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.3, #Trees: 101, Precision: 0.19, Recall: 0.87, F-Score: 0.31, AUC: 0.85, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.3, #Trees: 11, Precision: 0.19, Recall: 0.95, F-Score: 0.32, AUC: 0.90, G-Mean: 0.84

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.3, #Trees: 31, Precision: 0.19, Recall: 0.93, F-Score: 0.31, AUC: 0.88, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.3, #Trees: 51, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.3, #Trees: 101, Precision: 0.19, Recall: 0.88, F-Score: 0.32, AUC: 0.85, G-Mean: 0.81

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.93, F-Score: 0.31, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.4, #Trees: 31, Precision: 0.19, Recall: 0.91, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.4, #Trees: 51, Precision: 0.19, Recall: 0.87, F-Score: 0.31, AUC: 0.86, G-Mean: 0.80

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.86, F-Score: 0.32, AUC: 0.86, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.4, #Trees: 11, Precision: 0.19, Recall: 0.96, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.4, #Trees: 31, Precision: 0.19, Recall: 0.89, F-Score: 0.32, AUC: 0.87, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.4, #Trees: 51, Precision: 0.20, Recall: 0.91, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.4, #Trees: 101, Precision: 0.20, Recall: 0.84, F-Score: 0.32, AUC: 0.86, G-Mean: 0.80

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.5, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.31, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.5, #Trees: 31, Precision: 0.21, Recall: 0.79, F-Score: 0.33, AUC: 0.88, G-Mean: 0.79

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.5, #Trees: 51, Precision: 0.19, Recall: 0.89, F-Score: 0.32, AUC: 0.87, G-Mean: 0.81

ensemble size: 10, Noise Added to: test, Rn: 0.3, Rc: 0.5, #Trees: 101, Precision: 0.20, Recall: 0.86, F-Score: 0.32, AUC: 0.86, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.5, #Trees: 11, Precision: 0.18, Recall: 0.97, F-Score: 0.31, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.5, #Trees: 31, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.5, #Trees: 51, Precision: 0.20, Recall: 0.88, F-Score: 0.33, AUC: 0.88, G-Mean: 0.82

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.3, Rc: 0.5, #Trees: 101, Precision: 0.19, Recall: 0.81, F-Score: 0.31, AUC: 0.85, G-Mean: 0.79

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.1, #Trees: 11, Precision: 0.20, Recall: 0.94, F-Score: 0.33, AUC: 0.90, G-Mean: 0.84

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.1, #Trees: 31, Precision: 0.20, Recall: 0.91, F-Score: 0.32, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.1, #Trees: 51, Precision: 0.21, Recall: 0.90, F-Score: 0.34, AUC: 0.88, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.1, #Trees: 101, Precision: 0.19, Recall: 0.90, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.1, #Trees: 11, Precision: 0.18, Recall: 0.94, F-Score: 0.31, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.1, #Trees: 31, Precision: 0.19, Recall: 0.91, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.88, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.33, AUC: 0.86, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.95, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.2, #Trees: 31, Precision: 0.19, Recall: 0.89, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.2, #Trees: 51, Precision: 0.21, Recall: 0.90, F-Score: 0.34, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.86, F-Score: 0.33, AUC: 0.86, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.32, AUC: 0.90, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.2, #Trees: 31, Precision: 0.19, Recall: 0.94, F-Score: 0.32, AUC: 0.88, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.2, #Trees: 51, Precision: 0.19, Recall: 0.92, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.92, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.3, #Trees: 11, Precision: 0.19, Recall: 0.93, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.3, #Trees: 31, Precision: 0.20, Recall: 0.86, F-Score: 0.32, AUC: 0.88, G-Mean: 0.81

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.3, #Trees: 51, Precision: 0.20, Recall: 0.88, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.3, #Trees: 101, Precision: 0.21, Recall: 0.85, F-Score: 0.33, AUC: 0.86, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.3, #Trees: 11, Precision: 0.18, Recall: 0.93, F-Score: 0.30, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.3, #Trees: 31, Precision: 0.19, Recall: 0.86, F-Score: 0.32, AUC: 0.87, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.3, #Trees: 51, Precision: 0.19, Recall: 0.89, F-Score: 0.32, AUC: 0.87, G-Mean: 0.82

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.3, #Trees: 101, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.86, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.4, #Trees: 11, Precision: 0.18, Recall: 0.96, F-Score: 0.30, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.4, #Trees: 31, Precision: 0.20, Recall: 0.89, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.4, #Trees: 51, Precision: 0.20, Recall: 0.88, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.4, #Trees: 101, Precision: 0.19, Recall: 0.81, F-Score: 0.30, AUC: 0.85, G-Mean: 0.78

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.4, #Trees: 11, Precision: 0.18, Recall: 0.96, F-Score: 0.30, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.4, #Trees: 31, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.88, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.4, #Trees: 51, Precision: 0.19, Recall: 0.82, F-Score: 0.31, AUC: 0.86, G-Mean: 0.79

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.4, #Trees: 101, Precision: 0.19, Recall: 0.88, F-Score: 0.31, AUC: 0.85, G-Mean: 0.81

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.5, #Trees: 11, Precision: 0.19, Recall: 0.92, F-Score: 0.31, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.5, #Trees: 31, Precision: 0.19, Recall: 0.83, F-Score: 0.31, AUC: 0.87, G-Mean: 0.79

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.5, #Trees: 51, Precision: 0.20, Recall: 0.85, F-Score: 0.32, AUC: 0.88, G-Mean: 0.80

ensemble size: 10, Noise Added to: test, Rn: 0.4, Rc: 0.5, #Trees: 101, Precision: 0.19, Recall: 0.82, F-Score: 0.31, AUC: 0.84, G-Mean: 0.79

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.5, #Trees: 11, Precision: 0.18, Recall: 0.94, F-Score: 0.30, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.5, #Trees: 31, Precision: 0.18, Recall: 0.86, F-Score: 0.30, AUC: 0.88, G-Mean: 0.80

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.5, #Trees: 51, Precision: 0.21, Recall: 0.86, F-Score: 0.33, AUC: 0.88, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.4, Rc: 0.5, #Trees: 101, Precision: 0.20, Recall: 0.87, F-Score: 0.32, AUC: 0.86, G-Mean: 0.81

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.94, F-Score: 0.32, AUC: 0.89, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.1, #Trees: 31, Precision: 0.20, Recall: 0.87, F-Score: 0.32, AUC: 0.89, G-Mean: 0.81

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.1, #Trees: 101, Precision: 0.20, Recall: 0.87, F-Score: 0.32, AUC: 0.87, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.1, #Trees: 11, Precision: 0.19, Recall: 0.97, F-Score: 0.31, AUC: 0.90, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.1, #Trees: 31, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.90, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.1, #Trees: 51, Precision: 0.20, Recall: 0.94, F-Score: 0.33, AUC: 0.87, G-Mean: 0.83

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + \text{np.exp}(-2 * s))$$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + \text{np.exp}(-2 * s))$$

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.1, #Trees: 101, Precision: 0.19, Recall: 0.88, F-Score: 0.32, AUC: 0.86, G-Mean: 0.81

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.2, #Trees: 11, Precision: 0.18, Recall: 0.93, F-Score: 0.30, AUC: 0.89, G-Mean: 0.81

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.2, #Trees: 31, Precision: 0.21, Recall: 0.91, F-Score: 0.34, AUC: 0.90, G-Mean: 0.84

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.2, #Trees: 51, Precision: 0.20, Recall: 0.88, F-Score: 0.33, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.90, F-Score: 0.32, AUC: 0.86, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.2, #Trees: 11, Precision: 0.19, Recall: 0.96, F-Score: 0.31, AUC: 0.89, G-Mean: 0.83

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.2, #Trees: 31, Precision: 0.20, Recall: 0.93, F-Score: 0.33, AUC: 0.89, G-Mean: 0.84

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.2, #Trees: 51, Precision: 0.20, Recall: 0.82, F-Score: 0.32, AUC: 0.86, G-Mean: 0.80

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.2, #Trees: 101, Precision: 0.20, Recall: 0.89, F-Score: 0.32, AUC: 0.85, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.3, #Trees: 11, Precision: 0.18, Recall: 0.96, F-Score: 0.30, AUC: 0.89, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.3, #Trees: 31, Precision: 0.20, Recall: 0.87, F-Score: 0.33, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.3, #Trees: 51, Precision: 0.20, Recall: 0.89, F-Score: 0.33, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.3, #Trees: 101, Precision: 0.20, Recall: 0.84, F-Score: 0.32, AUC: 0.85, G-Mean: 0.80

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.3, #Trees: 11, Precision: 0.18, Recall: 0.91, F-Score: 0.30, AUC: 0.88, G-Mean: 0.81

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.3, #Trees: 31, Precision: 0.19, Recall: 0.85, F-Score: 0.31, AUC: 0.87, G-Mean: 0.80

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.3, #Trees: 51, Precision: 0.19, Recall: 0.87, F-Score: 0.32, AUC: 0.87, G-Mean: 0.81

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$p = 1. / (1 + \text{np.exp}(-2 * s))$

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.3, #Trees: 101, Precision: 0.20, Recall: 0.89, F-Score: 0.32, AUC: 0.86, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.4, #Trees: 11, Precision: 0.17, Recall: 0.93, F-Score: 0.29, AUC: 0.89, G-Mean: 0.81

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.4, #Trees: 31, Precision: 0.20, Recall: 0.89, F-Score: 0.32, AUC: 0.88, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.4, #Trees: 51, Precision: 0.19, Recall: 0.91, F-Score: 0.31, AUC: 0.87, G-Mean: 0.82

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.4, #Trees: 101, Precision: 0.19, Recall: 0.81, F-Score: 0.30, AUC: 0.84, G-Mean: 0.78

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.4, #Trees: 11, Precision: 0.18, Recall: 0.97, F-Score: 0.30, AUC: 0.89, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.4, #Trees: 31, Precision: 0.19, Recall: 0.85, F-Score: 0.31, AUC: 0.86, G-Mean: 0.80

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.4, #Trees: 51, Precision: 0.19, Recall: 0.84, F-Score: 0.31, AUC: 0.86, G-Mean: 0.80

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + np.exp(-2 * s))$$

C:/Users/Hossein/Downloads/Uni/Shiraz Uni/Machine Learning/Hashemi/HW/ML Assignment#3_98-99/adaboost.py:84: RuntimeWarning: overflow encountered in exp

$$p = 1. / (1 + np.exp(-2 * s))$$

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.4, #Trees: 101, Precision: 0.19, Recall: 0.80, F-Score: 0.30, AUC: 0.84, G-Mean: 0.78

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.5, #Trees: 11, Precision: 0.18, Recall: 0.97, F-Score: 0.31, AUC: 0.88, G-Mean: 0.83

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.5, #Trees: 31, Precision: 0.18, Recall: 0.77, F-Score: 0.29, AUC: 0.85, G-Mean: 0.76

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.5, #Trees: 51, Precision: 0.18, Recall: 0.83, F-Score: 0.29, AUC: 0.85, G-Mean: 0.78

ensemble size: 10, Noise Added to: test, Rn: 0.5, Rc: 0.5, #Trees: 101, Precision: 0.17, Recall: 0.72, F-Score: 0.28, AUC: 0.82, G-Mean: 0.74

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.5, #Trees: 11, Precision: 0.18, Recall: 0.96, F-Score: 0.30, AUC: 0.88, G-Mean: 0.82

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.5, #Trees: 31, Precision: 0.18, Recall: 0.83, F-Score: 0.30, AUC: 0.86, G-Mean: 0.79

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.5, #Trees: 51, Precision: 0.19, Recall: 0.79, F-Score: 0.30, AUC: 0.84, G-Mean: 0.78

ensemble size: 15, Noise Added to: test, Rn: 0.5, Rc: 0.5, #Trees: 101, Precision: 0.19, Recall: 0.80, F-Score: 0.30, AUC: 0.84, G-Mean: 0.78

Questions:

❓ Why should we set `max_depth` parameter in AdaBoost.M1 so that the base classifiers become a little better than random guess?

Because if the depth of tree > 1 then we do not consider it as the weak classifier.

❓ What do we mean by stable, unstable, and weak classifier?

Stable learning algorithm is one for which the prediction does not change much when the training data is modified slightly(not sensitive to the noise). But unstable don't act like this. Stable classifiers do not change much over replicates of T .

Some classification and regression methods are unstable in the sense that small perturbations in their training sets or in construction may result in large changes in the constructed predictor. Subset selection methods in regression, decision trees in regression and classification, and neural nets are unstable.

When we used AdaBoost, my weak classifiers were basically thresholds for each data attribute. Those thresholds need to have a performance of more than 50%, if not it would be totally random. A weak classifier is simply a classifier that performs poorly, but performs better than a random guessing.

❓ What kind of classifiers should be used in Bagging? How about AdaBoost.M1? Why?

We can bagging with any base learner but they should tend to have a low bias and, consequently, high variance.

AdaBoost is best used to boost the performance of decision trees on binary classification problems.

They are simpler in computation.

Questions:

Why is AdaBoost.NC algorithm suitable for imbalanced data classification?

It has good generalization ability under two-class imbalance scenarios by exploiting ensemble diversity

Diversity is important in ensemble learning. How does AdaBoost.NC incorporate diversity in its model?

that combines the strength of negative correlation learning and boosting. It emphasizes ensemble diversity explicitly during training and shows very encouraging empirical results in both effectiveness and efficiency in comparison with the conventional AdaBoost and other NCL methods in general cases.

❓ What are the differences between Precision, Recall, F-measure, AUC, and G-mean measures?

Please read more about the definitions of these measures.

Recall, precision, and F-measure are widely discussed single-class measures for two-class problems, which are still applicable to multiclass problems. Recall is a measure of completeness; precision is a measure of exactness. F-measure incorporates both to express their tradeoff. For the overall performance, G-mean and AUC are often used in the literature, but they are originally designed for two class problems. Therefore, they have to be adapted to multiclass scenarios: an extended G-mean is defined as the geometric mean of recall values of all classes; a commonly accepted extension of AUC is called M measure or MAUC, the average AUC of all pairs of classes.

❓ How does noise influence the classification results? Are AdaBoost.NC, AdaBoost, and Bagging robust to the added noise?

The presence of noise in the data may affect the intrinsic characteristics of a classification problem, since these corruptions could introduce new properties in the problem domain. For example, noise can lead to the creation of small clusters of examples of a particular class in areas of the domain corresponding to another class, or it can cause the disappearance of examples located in key areas within a specific class. The boundaries of the classes and the overlapping between them are also factors that can be affected as a consequence of noise. All these alterations difficult the knowledge extraction from the data and spoil the models obtained using that noisy data when they are compared to the models learned from clean data, which represent the real implicit knowledge of the problem.

AdaBoost can be sensitive to outliers / label **noise** because it is fitting a classification model (an additive model) to an exponential loss function, and the exponential loss function is sensitive to outliers/label **noise**

bagging is much better than boosting, and sometimes better than randomization.