precision

April 4, 2017

0.0.1 This report shows applyining statistical tests of the results of Multi armed bandit of pruning the parameters

0.0.2 Here, we are showing two kinds of testing ANOVA test and Nonparametric tests

1 Import needed libraries

1.1 Import libraries for manipulating the data and statistic

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import scipy.stats as stats
        from scipy.stats import ttest_1samp, wilcoxon, ttest_ind, mannwhitneyu
        import scipy.special as special
        import emoji
        from math import pi
        from statsmodels.stats.multicomp import pairwise_tukeyhsd, MultiComparison
        from statsmodels.formula.api import ols
        import statsmodels.stats.api as sms
```

1.2 Import libraries for static ploting

```
In [2]: import matplotlib.pyplot as plt
        import matplotlib.gridspec as gridspec
        %matplotlib inline
        from IPython.display import set_matplotlib_formats
        set_matplotlib_formats('png', 'pdf')
        # some nice colors from http://colorbrewer2.org/
        COLOR1 = '#7fc97f'
        COLOR2 = '#beaed4'
        COLOR3 = '#fdc086'
        COLOR4 = '#ffff99'
        COLOR5 = '#386cb0'
```

1.3 Import libraries for interactive ploting Plotly

```
In [3]: import plotly.plotly as py
        from plotly.graph_objs import *
        import plotly.graph_objs as go
        #from plotly.tools import FigureFactory as FF
        import plotly.figure_factory as FF
        import cufflinks as cf
        cf.go_offline()
```

```
<IPython.core.display.HTML object>
```

1.4 Import libraries for interactive ploting BOKEH

```
In [4]: from bokeh.charts import Bar, Area, defaults, Donut
        from bokeh.layouts import row, gridplot
        from bokeh.charts.attributes import cat, color
        from bokeh.charts.operations import blend
        from bokeh.plotting import figure, output_notebook, show
        from bokeh.models import Legend
        TOOLS = 'box_zoom,box_select,crosshair,resize,reset,lasso_select,pan,save,poly_select,ta
        #defaults.width = 1000
        #defaults.height = 800
        output_notebook()
```

2 Statring the test and visulize the data on small model

2.1 Load the data for pruning the weights using random expoloration

```
In [5]: datafile = "./results/result.xlsx"
        #datafileLeNet = "LecunPruningWeights.csv"
        df_precision = pd.read_excel(datafile, sheetname='accuracy')
        df_precision = pd.read_excel(datafile, sheetname='f1score')
        df_precision = pd.read_excel(datafile, sheetname='precision')
        df_recall    = pd.read_excel(datafile, sheetname='recall')
        #dfLcun = pd.read_csv(datafileLeNet)
        df_precision
```

```
Out[5]:           Methods      SPAM   ABALONE     ADULT    CANCER       CAR  \
        0             Knn  0.899390  0.537541  0.632099  0.875000  0.932718
        1            LSVM  0.907463  0.535859  0.699438  0.900000  0.639023
        2             SVM  0.962382  0.566223  0.767176  0.972973  0.882738
        3         DT_gini  0.907895  0.481021  0.551935  0.900000  0.793132
        4      DT_entorpy  0.945946  0.469148  0.574074  0.894737  0.849928
        5     Bagging Knn  0.952229  0.543804  0.716141  0.947368  0.351484
        6      Bagging DT  0.978571  0.543084  0.698381  0.945946  0.380017
        7   Random Forest  0.952030  0.507469  0.701079  0.972222  0.783861
        8       Ada Boost  0.982759  0.530017  0.693548  0.878049  0.759924
        9              NB  0.675732  0.491034  0.643275  0.660000  0.724146
        10            LDA  0.938111  0.544203  0.730570  0.947368  0.449612
        11            QDA  0.687500  0.539138  0.720339  0.833333  0.797756
```

```
12       Log. Reg.  0.917722  0.541705  0.734694  0.923077  0.640167
13       GP Class.  0.926380  0.547082  0.722983  0.900000  0.877795
14        LightGBM  0.928793  0.528221  0.714968  0.947368  0.765689
15         Xgboost  0.985646  0.560579  0.728125  0.921053  0.839557
16              NN  0.956656  0.563967  0.694181  0.947368  0.868407
17            UCB1  0.957055  0.579791  0.672956  0.947368  0.795608
18        E Greedy  0.948328  0.579791  0.620227  0.947368  0.800424
19    Decay E Gr.   0.951070  0.579791  0.732612  0.945946  0.865544
20         Softmax  0.953846  0.579791  0.735027  0.947368  0.617017
21        Decay SM  0.948328  0.579791  0.750000  0.947368  0.822131
22  Tomp. Sampling  0.942943  0.579791  0.672956  0.947368  0.795608
23           Hedge  0.948171  0.579791  0.672772  0.947368  0.852209
24            EXP3  0.947853  0.579791  0.653409  0.972973  0.580044

       GLASS     HEART      IRIS      PIMA     POKER   TITANIC     VALLY  \
0   0.522917  0.502726  1.000000  0.628571  0.270603  0.671233  0.880734
1   0.715686  0.262193  0.939394  0.500000  0.055692  0.761905  0.817073
2   0.670139  0.760440  1.000000  0.675000  0.425904  0.753623  0.852273
3   0.599673  0.709286  1.000000  0.517857  0.759462  0.576087  0.731481
4   0.604167  0.753468  0.916667  0.578947  0.585722  0.542553  0.828829
5   0.501245  0.541388  0.969697  0.555556  0.277847  0.710145  0.841584
6   0.495833  0.737291  0.897436  0.560000  0.349514  0.750000  0.851852
7   0.618960  0.804077  1.000000  0.714286  0.705821  0.741935  0.848485
8   0.226190  0.311524  0.939394  0.588235  0.057055  0.726027  0.776699
9   0.614496  0.296939  0.939394  0.413793  0.027174  0.728571  0.533945
10  0.690476  0.285772  0.966667  0.588235  0.054348  0.742424  0.696203
11  0.359244  0.603002  0.966667  0.500000  0.124196  0.737500  0.544423
12  0.697829  0.266867  0.966667  0.588235  0.054332  0.753846  0.833333
13  0.683626  0.844444  1.000000  0.656250  0.613514  0.716216  0.877551
14  0.382832  0.530485  0.939394  0.640000  0.234696  0.774194  0.922222
15  0.481481  0.760847  0.939394  0.578947  0.298283  0.796610  0.923077
16  0.490810  0.520258  0.966667  0.621622  0.377618  0.784615  0.802083
17  0.526864  0.557446  1.000000  0.600000  0.346168  0.784615  0.835165
18  0.418831  0.545634  0.939394  0.628571  0.179355  0.784615  0.831325
19  0.511905  0.548575  0.916667  0.666667  0.224014  0.784615  0.810000
20  0.501221  0.385510  0.916667  0.625000  0.368106  0.784615  0.815217
21  0.466435  0.423485  0.491228  0.928571  0.298508  0.784615  0.839080
22  0.526864  0.539286  0.491228  0.600000  0.287830  0.784615  0.835165
23  0.495635  0.464944  0.491228  0.625000  0.399897  0.784615  0.800000
24  0.293981  0.504944  0.453704  0.645161  0.299178  0.784615  0.752381

        WINE  WINE QUILTY      FACE     CHEST
0   0.952381     0.331889  0.625146  0.398237
1   1.000000     0.226500  0.000000  0.081506
2   1.000000     0.244404  0.816381  0.361797
3   0.884127     0.288980  0.249460  0.318427
4   0.878968     0.321229  0.000000  0.360219
5   0.977778     0.398915  0.000000  0.257988
```

```
6   0.893557     0.360275  0.000000  0.162500
7   0.875000     0.380944  0.000000  0.466733
8   0.925926     0.208103  0.000000  0.195703
9   0.910714     0.198334  0.000000  0.156345
10  0.930159     0.233021  0.000000  0.084257
11  1.000000     0.281868  0.000000  0.151025
12  1.000000     0.239299  0.000000  0.087898
13  0.910714     0.507200  0.000000  0.336044
14  0.902778     0.276225  0.000000  0.266481
15  0.910714     0.229662  0.000000  0.474318
16  1.000000     0.304177  0.832348  0.173357
17  1.000000     0.339067  0.839225  0.146184
18  1.000000     0.288260  0.843212  0.150979
19  1.000000     0.211777  0.819550  0.221250
20  1.000000     0.245726  0.841262  0.152046
21  1.000000     0.394014  0.841262  0.218307
22  1.000000     0.263063  0.841262  0.146184
23  1.000000     0.241724  0.841262  0.166196
24  0.977778     0.228358  0.841262  0.192144
```

3 Starting with precision

4 First, All methods

4.1 Visulize the Accuracy of all the models and methods

```
In [6]: p = Bar(df_precision, label= 'Methods',
                 values = blend('SPAM', 'ABALONE', 'ADULT', 'CANCER', 'CAR', 'GLASS',
                               'HEART', 'IRIS', 'PIMA', 'POKER', 'TITANIC', 'VALLY', 'WINE',
                               'WINE QUILTY', 'FACE', 'CHEST'
                               ,name='Scores', labels_name='Score'),
                group=cat(columns='Score', sort=False),
                title="Compare the performance", legend='bottom_center',
                tools=TOOLS, plot_width=3500, plot_height=1600,
                tooltips=[('Score', '@Score'), ('Model', '@Methods')],
                xlabel='List of Models', ylabel='Score')
        p.title.align = "center"
        #p.yaxis.major_label_orientation = "vertical"
        p.xaxis.major_label_orientation = pi/2
        show(p)
```

4.1.1 Only pruning methods

```
In [7]: df_precision_pruning = df_precision.iloc[16:24]
        p = Bar(df_precision_pruning, label= 'Methods',
                 values = blend('SPAM', 'ABALONE', 'ADULT', 'CANCER', 'CAR', 'GLASS',
                               'HEART', 'IRIS', 'PIMA', 'POKER', 'TITANIC', 'VALLY', 'WINE',
                               'WINE QUILTY', 'FACE', 'CHEST'
                               ,name='Scores', labels_name='Score'),
               group=cat(columns='Score', sort=False),
               title="Compare the performance", legend='bottom_center',
               tools=TOOLS, plot_width=2000, plot_height=1500,
               tooltips=[('Score', '@Score'), ('Model', '@Methods')],
               xlabel='List of Models', ylabel='Score')
        p.title.align = "center"
        #p.yaxis.major_label_orientation = "vertical"
        p.xaxis.major_label_orientation = pi/2
        show(p)
```

```
In [8]: df=df_precision.copy()
        df.set_index('Methods', inplace=True)
        py.iplot([{
            'x': df.index,
            'y': df[col],
            'name': col
        }  for col in df.columns])
```

```
Out[8]: <plotly.tools.PlotlyDisplay object>
```

```
In [9]: df_pruning=df_precision_pruning.copy()
        df_pruning.set_index('Methods', inplace=True)
        py.iplot([{
            'x': df_pruning.index,
            'y': df_pruning[col],
            'name': col
        }  for col in df_pruning.columns])
```

```
Out[9]: <plotly.tools.PlotlyDisplay object>
```

```
In [10]: df.iplot(subplots=True, shape=(16,1), shared_xaxes=True, fill=True)
```

```
<IPython.core.display.HTML object>
```

```
In [11]: df.iplot(kind='bar', barmode='stack')
```

```
<IPython.core.display.HTML object>
```

```
In [12]: df.iplot(kind='barh',barmode='stack', bargap=.2)
```

```
<IPython.core.display.HTML object>
```

```
In [13]: df.T.iplot(kind='barh',barmode='stack', bargap=.2)
```

```
<IPython.core.display.HTML object>
```

```
In [14]: df.iplot(kind='box')

<IPython.core.display.HTML object>
```

```
In [15]: df.T.iplot(kind='box')

<IPython.core.display.HTML object>
```

4.1.2 We will use alpha 0.05 to do ANOVA test. The null hypothesis there is no difference between the all methods and the alternative hypothesis there is a difference. According to p-value we see if there is a difference.

```
In [16]: df.T.columns

Out[16]: Index(['Knn', 'LSVM', 'SVM', 'DT_gini', 'DT_entorpy', 'Bagging Knn',
                'Bagging DT', 'Random Forest', 'Ada Boost', 'NB', 'LDA', 'QDA',
                'Log. Reg.', 'GP Class.', 'LightGBM', 'Xgboost', 'NN', 'UCB1',
                'E Greedy', 'Decay E Gr.', 'Softmax', 'Decay SM', 'Tomp. Sampling',
                'Hedge', 'EXP3'],
               dtype='object', name='Methods')
```

```
In [17]: # Perform the ANOVA
         df1 = df.T
         stats.f_oneway(df1['Knn'], df1['LSVM'], df1['SVM'], df1['DT_gini'], df1['DT_entorpy'],
                        df1['Bagging Knn'],df1['Bagging DT'], df1['Random Forest'],
                        df1['Ada Boost'], df1['NB'], df1['LDA'], df1['QDA'],
                        df1['Log. Reg.'], df1['GP Class.'], df1['LightGBM'], df1['Xgboost'],
                        df1['NN'], df1['UCB1'],df1['E Greedy'], df1['Decay E Gr.'],
                        df1['Softmax'], df1['Decay SM'], df1['Tomp. Sampling'],
                        df1['Hedge'], df1['EXP3'])

Out[17]: F_onewayResult(statistic=0.64113414370718658, pvalue=0.90467397956549955)
```

4.1.3 One post-hoc test is to perform a separate t-test for each pair of groups. We can perform a t-test between all pairs using by running each pair through the stats.ttest_ind() we covered in the following to do t-tests:

```
In [18]: # Get all models pairs
         interstModel = ['NN', 'UCB1',
                'E Greedy', 'Decay E Gr.', 'Softmax', 'Decay SM', 'Tomp. Sampling',
                'Hedge', 'EXP3']
         lst = list(df1.columns.values)
         #lst.remove('Methods')
         model_pairs = []

         for m1 in range(len(df1.columns)-1):
             for m2  in range(m1+1,len(df1.columns)):
```

```
        model_pairs.append((lst[m1], lst[m2]))

# Conduct t-test on each pair
pvalueList = []
new_model_pairs = []
for m1, m2 in model_pairs:
    print('\n',m1, m2)
    pvalue = stats.ttest_ind(df1[m1], df1[m2])
    #print(pvalue[1])
    if (m1 in interstModel or m2 in interstModel):
        new_model_pairs.append((m1,m2))
        pvalueList.append(pvalue[1])
    print(pvalue)
```

```
 Knn LSVM
Ttest_indResult(statistic=0.98237674348012849, pvalue=0.33377099083058859)

 Knn SVM
Ttest_indResult(statistic=-0.7986742921137171, pvalue=0.43075707889545034)

 Knn DT_gini
Ttest_indResult(statistic=0.29768002088876128, pvalue=0.76799916430130177)

 Knn DT_entorpy
Ttest_indResult(statistic=0.39722503014495597, pvalue=0.6940136325935834)

 Knn Bagging Knn
Ttest_indResult(statistic=0.74247825982509497, pvalue=0.46357193595913959)

 Knn Bagging DT
Ttest_indResult(statistic=0.70546617456441796, pvalue=0.48596231080919061)

 Knn Random Forest
Ttest_indResult(statistic=-0.29667694176584891, pvalue=0.7687575135851572)

 Knn Ada Boost
Ttest_indResult(statistic=1.1395929389084805, pvalue=0.26347346540515781)

 Knn NB
Ttest_indResult(statistic=1.7673564229994416, pvalue=0.087340685552345365)

 Knn LDA
Ttest_indResult(statistic=1.0897762013885921, pvalue=0.28448830195355101)

 Knn QDA
Ttest_indResult(statistic=1.1920169795191708, pvalue=0.24259893138808686)
```

```
 Knn Log. Reg.
Ttest_indResult(statistic=0.85319465157838159, pvalue=0.40031238070565622)

 Knn GP Class.
Ttest_indResult(statistic=-0.3292225800898439, pvalue=0.74427495454587822)

 Knn LightGBM
Ttest_indResult(statistic=0.59281806750701915, pvalue=0.55774563754979345)

 Knn Xgboost
Ttest_indResult(statistic=0.15561169753055754, pvalue=0.87738145997733674)

 Knn NN
Ttest_indResult(statistic=-0.17526656326870912, pvalue=0.8620477926349559)

 Knn UCB1
Ttest_indResult(statistic=-0.19163411419394286, pvalue=0.84931999534988767)

 Knn E Greedy
Ttest_indResult(statistic=0.10650107964235492, pvalue=0.91589379903652768)

 Knn Decay E Gr.
Ttest_indResult(statistic=-0.09055702498577492, pvalue=0.92844651515541265)

 Knn Softmax
Ttest_indResult(statistic=0.13650852248258069, pvalue=0.89233125601162988)

 Knn Decay SM
Ttest_indResult(statistic=-0.051806766920778825, pvalue=0.95902609113430692)

 Knn Tomp. Sampling
Ttest_indResult(statistic=0.29233103637263735, pvalue=0.77204578836473448)

 Knn Hedge
Ttest_indResult(statistic=0.25380512578704134, pvalue=0.80137755999410276)

 Knn EXP3
Ttest_indResult(statistic=0.6772225247622552, pvalue=0.50345409393670648)

 LSVM SVM
Ttest_indResult(statistic=-1.6232584637649421, pvalue=0.11499720161303838)

 LSVM DT_gini
Ttest_indResult(statistic=-0.74499541058301266, pvalue=0.46207140978502859)

 LSVM DT_entorpy
Ttest_indResult(statistic=-0.6199865135552316, pvalue=0.53994777074113909)
```

```
 LSVM Bagging Knn
Ttest_indResult(statistic=-0.27844437389196219, pvalue=0.78258154295863547)

 LSVM Bagging DT
Ttest_indResult(statistic=-0.31347020865941844, pvalue=0.75609263952453309)

 LSVM Random Forest
Ttest_indResult(statistic=-1.1924255297329571, pvalue=0.24244118748519278)

 LSVM Ada Boost
Ttest_indResult(statistic=0.12704973088171104, pvalue=0.89974880007553604)

 LSVM NB
Ttest_indResult(statistic=0.57285242141526149, pvalue=0.57101364141116728)

 LSVM LDA
Ttest_indResult(statistic=0.0839908408334923, pvalue=0.93362168968467762)

 LSVM QDA
Ttest_indResult(statistic=0.10784008816912569, pvalue=0.91484055689912946)

 LSVM Log. Reg.
Ttest_indResult(statistic=-0.1056211279330372, pvalue=0.91658603921930137)

 LSVM GP Class.
Ttest_indResult(statistic=-1.2168921106162398, pvalue=0.23313220118901429)

 LSVM LightGBM
Ttest_indResult(statistic=-0.39135291943581968, pvalue=0.6983022921179034)

 LSVM Xgboost
Ttest_indResult(statistic=-0.77323317003066772, pvalue=0.44543462686731428)

 LSVM NN
Ttest_indResult(statistic=-1.0941789029421241, pvalue=0.28258432846329629)

 LSVM UCB1
Ttest_indResult(statistic=-1.1060073562480783, pvalue=0.2775140464681321)

 LSVM E Greedy
Ttest_indResult(statistic=-0.83313610571427321, pvalue=0.41135210514805665)

 LSVM Decay E Gr.
Ttest_indResult(statistic=-1.0095898035677608, pvalue=0.32076587867124084)

 LSVM Softmax
Ttest_indResult(statistic=-0.82789060977194362, pvalue=0.41427013690383174)
```

```
 LSVM Decay SM
Ttest_indResult(statistic=-0.9925434917779804, pvalue=0.32887104836756154)

 LSVM Tomp. Sampling
Ttest_indResult(statistic=-0.71022132327574294, pvalue=0.48305167868617216)

 LSVM Hedge
Ttest_indResult(statistic=-0.74763411173890204, pvalue=0.46050148700978699)

 LSVM EXP3
Ttest_indResult(statistic=-0.3871144692914138, pvalue=0.70140412872620539)

 SVM DT_gini
Ttest_indResult(statistic=1.0985058632133058, pvalue=0.28072196880959843)

 SVM DT_entorpy
Ttest_indResult(statistic=1.1504937398545385, pvalue=0.25902888234943855)

 SVM Bagging Knn
Ttest_indResult(statistic=1.4439944868711849, pvalue=0.1591036476136167)

 SVM Bagging DT
Ttest_indResult(statistic=1.4104631706684798, pvalue=0.16868823660159649)

 SVM Random Forest
Ttest_indResult(statistic=0.46164802003945099, pvalue=0.64766388596724223)

 SVM Ada Boost
Ttest_indResult(statistic=1.7866076388553966, pvalue=0.084111287593932355)

 SVM NB
Ttest_indResult(statistic=2.4755065725291692, pvalue=0.019174591424316219)

 SVM LDA
Ttest_indResult(statistic=1.7369924148542102, pvalue=0.09264756178634076)

 SVM QDA
Ttest_indResult(statistic=1.8868646481064495, pvalue=0.068889598202820923)

 SVM Log. Reg.
Ttest_indResult(statistic=1.4896622649816402, pvalue=0.14675220188451671)

 SVM GP Class.
Ttest_indResult(statistic=0.42611495073228323, pvalue=0.67306598663042305)

 SVM LightGBM
Ttest_indResult(statistic=1.2828673248008557, pvalue=0.2093616996279859)
```

```
 SVM Xgboost
Ttest_indResult(statistic=0.85979731289264916, pvalue=0.39671973096440072)

 SVM NN
Ttest_indResult(statistic=0.58433209909222705, pvalue=0.56336563093396996)

 SVM UCB1
Ttest_indResult(statistic=0.56593092654002575, pvalue=0.57564985994240447)

 SVM E Greedy
Ttest_indResult(statistic=0.83123638293846691, pvalue=0.41240742407210762)

 SVM Decay E Gr.
Ttest_indResult(statistic=0.64990706085271366, pvalue=0.52069817406775476)

 SVM Softmax
Ttest_indResult(statistic=0.88307328944914654, pvalue=0.38421856775470831)

 SVM Decay SM
Ttest_indResult(statistic=0.70684828037784275, pvalue=0.48511529597837411)

 SVM Tomp. Sampling
Ttest_indResult(statistic=1.0500831791985155, pvalue=0.30206576540908431)

 SVM Hedge
Ttest_indResult(statistic=1.0179877169971729, pvalue=0.31682367238541087)

 SVM EXP3
Ttest_indResult(statistic=1.4276459428647863, pvalue=0.16372127813985829)

 DT_gini DT_entorpy
Ttest_indResult(statistic=0.11733910545917843, pvalue=0.9073734078181579)

 DT_gini Bagging Knn
Ttest_indResult(statistic=0.48239764753761882, pvalue=0.63302472212114758)

 DT_gini Bagging DT
Ttest_indResult(statistic=0.44403089910955257, pvalue=0.66020671005574261)

 DT_gini Random Forest
Ttest_indResult(statistic=-0.58010013831736318, pvalue=0.56617901053610309)

 DT_gini Ada Boost
Ttest_indResult(statistic=0.90023522765320274, pvalue=0.37516514033631276)

 DT_gini NB
Ttest_indResult(statistic=1.5069547890787938, pvalue=0.14228110349972972)
```

```
 DT_gini LDA
Ttest_indResult(statistic=0.85025850071491016, pvalue=0.40191658278367703)

 DT_gini QDA
Ttest_indResult(statistic=0.93528353312877133, pvalue=0.3571102436824739)

 DT_gini Log. Reg.
Ttest_indResult(statistic=0.61721502774631787, pvalue=0.54174959446718074)

 DT_gini GP Class.
Ttest_indResult(statistic=-0.61160939216643817, pvalue=0.54540359490767998)

 DT_gini LightGBM
Ttest_indResult(statistic=0.33665348539636114, pvalue=0.73872185256024903)

 DT_gini Xgboost
Ttest_indResult(statistic=-0.10666251101319077, pvalue=0.9157668115054155)

 DT_gini NN
Ttest_indResult(statistic=-0.45886991321262821, pvalue=0.64963496499197748)

 DT_gini UCB1
Ttest_indResult(statistic=-0.4745197699511971, pvalue=0.63856533782734193)

 DT_gini E Greedy
Ttest_indResult(statistic=-0.16349648480686652, pvalue=0.87122401429818885)

 DT_gini Decay E Gr.
Ttest_indResult(statistic=-0.3668394067639823, pvalue=0.71631368982616583)

 DT_gini Softmax
Ttest_indResult(statistic=-0.14151183265490461, pvalue=0.88841163579995364)

 DT_gini Decay SM
Ttest_indResult(statistic=-0.33476005667873932, pvalue=0.74013546339139524)

 DT_gini Tomp. Sampling
Ttest_indResult(statistic=0.010540549373320602, pvalue=0.99165979842808549)

 DT_gini Hedge
Ttest_indResult(statistic=-0.030454584630675186, pvalue=0.97590622088155965)

 DT_gini EXP3
Ttest_indResult(statistic=0.39904154839641387, pvalue=0.69268902393432419)

 DT_entorpy Bagging Knn
Ttest_indResult(statistic=0.35587871353442974, pvalue=0.72442182461549554)
```

```
 DT_entorpy Bagging DT
Ttest_indResult(statistic=0.31861957494530024, pvalue=0.7522274097065633)

 DT_entorpy Random Forest
Ttest_indResult(statistic=-0.66079281452632543, pvalue=0.51378794599163946)

 DT_entorpy Ada Boost
Ttest_indResult(statistic=0.76775378732360566, pvalue=0.4486345916204163)

 DT_entorpy NB
Ttest_indResult(statistic=1.3316194901050156, pvalue=0.1930152825861739)

 DT_entorpy LDA
Ttest_indResult(statistic=0.71969504545105079, pvalue=0.47728255743559367)

 DT_entorpy QDA
Ttest_indResult(statistic=0.78967572593862334, pvalue=0.43591457068416817)

 DT_entorpy Log. Reg.
Ttest_indResult(statistic=0.49807235459859411, pvalue=0.62206483970543802)

 DT_entorpy GP Class.
Ttest_indResult(statistic=-0.69044811583261856, pvalue=0.49522001615364353)

 DT_entorpy LightGBM
Ttest_indResult(statistic=0.21884921186037656, pvalue=0.82824958946389338)

 DT_entorpy Xgboost
Ttest_indResult(statistic=-0.20632344414529752, pvalue=0.83793214884470157)

 DT_entorpy NN
Ttest_indResult(statistic=-0.54618212473433092, pvalue=0.5889799454616006)

 DT_entorpy UCB1
Ttest_indResult(statistic=-0.56084581023633773, pvalue=0.57906788245867191)

 DT_entorpy E Greedy
Ttest_indResult(statistic=-0.26283282992225043, pvalue=0.79447609649155637)

 DT_entorpy Decay E Gr.
Ttest_indResult(statistic=-0.45749824392210414, pvalue=0.65060912256840064)

 DT_entorpy Softmax
Ttest_indResult(statistic=-0.24443105790246608, pvalue=0.80856107885583561)

 DT_entorpy Decay SM
Ttest_indResult(statistic=-0.42868045043201913, pvalue=0.67121829480069906)
```

```
 DT_entorpy Tomp. Sampling
Ttest_indResult(statistic=-0.1017644486007573, pvalue=0.9196207874144553)

 DT_entorpy Hedge
Ttest_indResult(statistic=-0.14138973474432828, pvalue=0.88850725426349941)

 DT_entorpy EXP3
Ttest_indResult(statistic=0.26741947246280096, pvalue=0.79097614427349328)

 Bagging Knn Bagging DT
Ttest_indResult(statistic=-0.036703602661116386, pvalue=0.9709644835651019)

 Bagging Knn Random Forest
Ttest_indResult(statistic=-0.97703484737567803, pvalue=0.33636529049258801)

 Bagging Knn Ada Boost
Ttest_indResult(statistic=0.41622404474715197, pvalue=0.68020882940086669)

 Bagging Knn NB
Ttest_indResult(statistic=0.91899331926230321, pvalue=0.36542943379435711)

 Bagging Knn LDA
Ttest_indResult(statistic=0.37035127783809274, pvalue=0.71372283413823667)

 Bagging Knn QDA
Ttest_indResult(statistic=0.41337879291662222, pvalue=0.68226919683815967)

 Bagging Knn Log. Reg.
Ttest_indResult(statistic=0.16431524999931849, pvalue=0.87058508100145748)

 Bagging Knn GP Class.
Ttest_indResult(statistic=-1.0039955137605401, pvalue=0.32341060309788383)

 Bagging Knn LightGBM
Ttest_indResult(statistic=-0.12475158807155594, pvalue=0.90155241049866919)

 Bagging Knn Xgboost
Ttest_indResult(statistic=-0.53226180024222669, pvalue=0.59846500930689039)

 Bagging Knn NN
Ttest_indResult(statistic=-0.87002548351860753, pvalue=0.39119486913525869)

 Bagging Knn UCB1
Ttest_indResult(statistic=-0.88316838009908272, pvalue=0.38416802060433497)

 Bagging Knn E Greedy
Ttest_indResult(statistic=-0.59276398824637999, pvalue=0.55778136164582848)
```

```
 Bagging Knn Decay E Gr.
Ttest_indResult(statistic=-0.78102618867264961, pvalue=0.44090711426680318)

 Bagging Knn Softmax
Ttest_indResult(statistic=-0.58290779865197673, pvalue=0.56431170550665732)

 Bagging Knn Decay SM
Ttest_indResult(statistic=-0.75964786212896396, pvalue=0.45339352138590905)

 Bagging Knn Tomp. Sampling
Ttest_indResult(statistic=-0.45293761310323072, pvalue=0.65385257975825628)

 Bagging Knn Hedge
Ttest_indResult(statistic=-0.49232099527997031, pvalue=0.62607619910569157)

 Bagging Knn EXP3
Ttest_indResult(statistic=-0.10380790620089984, pvalue=0.9180126729843775)

 Bagging DT Random Forest
Ttest_indResult(statistic=-0.94241198838704821, pvalue=0.35350955497776304)

 Bagging DT Ada Boost
Ttest_indResult(statistic=0.45201450724534692, pvalue=0.65450991918728452)

 Bagging DT NB
Ttest_indResult(statistic=0.95966297986454818, pvalue=0.34489596323189398)

 Bagging DT LDA
Ttest_indResult(statistic=0.4059923968316313, pvalue=0.6876295939353545)

 Bagging DT QDA
Ttest_indResult(statistic=0.45141313172042186, pvalue=0.65493830742183201)

 Bagging DT Log. Reg.
Ttest_indResult(statistic=0.19874815519754599, pvalue=0.84380057462343072)

 Bagging DT GP Class.
Ttest_indResult(statistic=-0.96958844439512026, pvalue=0.34000433202605163)

 Bagging DT LightGBM
Ttest_indResult(statistic=-0.08901914301281956, pvalue=0.92965832881107924)

 Bagging DT Xgboost
Ttest_indResult(statistic=-0.49758292699643158, pvalue=0.62240574090055567)

 Bagging DT NN
Ttest_indResult(statistic=-0.834873188213061, pvalue=0.41038861013250105)
```

```
 Bagging DT UCB1
Ttest_indResult(statistic=-0.84814142992294439, pvalue=0.4030757810351917)

 Bagging DT E Greedy
Ttest_indResult(statistic=-0.55758101678389094, pvalue=0.58126761888038181)

 Bagging DT Decay E Gr.
Ttest_indResult(statistic=-0.74610543211818525, pvalue=0.46141060871923856)

 Bagging DT Softmax
Ttest_indResult(statistic=-0.54690311802101221, pvalue=0.58849066442168807)

 Bagging DT Decay SM
Ttest_indResult(statistic=-0.72402262381337423, pvalue=0.47466046859296029)

 Bagging DT Tomp. Sampling
Ttest_indResult(statistic=-0.4159508886993819, pvalue=0.68040652477577834)

 Bagging DT Hedge
Ttest_indResult(statistic=-0.45528312401911863, pvalue=0.65218362190061108)

 Bagging DT EXP3
Ttest_indResult(statistic=-0.065534983896256185, pvalue=0.9481828308458643)

 Random Forest Ada Boost
Ttest_indResult(statistic=1.3459661711910988, pvalue=0.18839730937627105)

 Random Forest NB
Ttest_indResult(statistic=1.963312288636192, pvalue=0.058938621788357126)

 Random Forest LDA
Ttest_indResult(statistic=1.2978604412867998, pvalue=0.20422593053583166)

 Random Forest QDA
Ttest_indResult(statistic=1.407612838133899, pvalue=0.16952354443800005)

 Random Forest Log. Reg.
Ttest_indResult(statistic=1.0662955983315965, pvalue=0.29479655850792819)

 Random Forest GP Class.
Ttest_indResult(statistic=-0.032176055887500707, pvalue=0.97454477215311508)

 Random Forest LightGBM
Ttest_indResult(statistic=0.829955445650849, pvalue=0.41311995114296929)

 Random Forest Xgboost
Ttest_indResult(statistic=0.41403942609168859, pvalue=0.68179058176463125)
```

```
 Random Forest NN
Ttest_indResult(statistic=0.11630894195980045, pvalue=0.90818281315748461)

 Random Forest UCB1
Ttest_indResult(statistic=0.099960201447432243, pvalue=0.92104094077343346)

 Random Forest E Greedy
Ttest_indResult(statistic=0.37371182063949637, pvalue=0.71124686255050762)

 Random Forest Decay E Gr.
Ttest_indResult(statistic=0.19044421651862412, pvalue=0.85024394225218458)

 Random Forest Softmax
Ttest_indResult(statistic=0.40959266945233258, pvalue=0.68501475037356285)

 Random Forest Decay SM
Ttest_indResult(statistic=0.2337860147739787, pvalue=0.81673914267143743)

 Random Forest Tomp. Sampling
Ttest_indResult(statistic=0.56197188347518978, pvalue=0.57831011603196303)

 Random Forest Hedge
Ttest_indResult(statistic=0.52718635777843159, pvalue=0.60194135985631902)

 Random Forest EXP3
Ttest_indResult(statistic=0.92753291422173789, pvalue=0.36105263873654503)

 Ada Boost NB
Ttest_indResult(statistic=0.44094137594195859, pvalue=0.66241686307948877)

 Ada Boost LDA
Ttest_indResult(statistic=-0.043390103800929655, pvalue=0.96567808565145996)

 Ada Boost QDA
Ttest_indResult(statistic=-0.025993251732053215, pvalue=0.97943485905788974)

 Ada Boost Log. Reg.
Ttest_indResult(statistic=-0.23274966061139118, pvalue=0.81753645992354607)

 Ada Boost GP Class.
Ttest_indResult(statistic=-1.3702183278273619, pvalue=0.18078624851473876)

 Ada Boost LightGBM
Ttest_indResult(statistic=-0.52839342576093484, pvalue=0.60111372947541786)

 Ada Boost Xgboost
Ttest_indResult(statistic=-0.91533791101951978, pvalue=0.36731353336659855)
```

```
 Ada Boost NN
Ttest_indResult(statistic=-1.2470250154730247, pvalue=0.22203657336846785)

 Ada Boost UCB1
Ttest_indResult(statistic=-1.2586686691909434, pvalue=0.21785715565388475)

 Ada Boost E Greedy
Ttest_indResult(statistic=-0.97872606485948188, pvalue=0.33554247752820765)

 Ada Boost Decay E Gr.
Ttest_indResult(statistic=-1.1589486093094385, pvalue=0.25561934100744338)

 Ada Boost Softmax
Ttest_indResult(statistic=-0.97654596092862744, pvalue=0.33660339795521454)

 Ada Boost Decay SM
Ttest_indResult(statistic=-1.1443495462119928, pvalue=0.26152729920622753)

 Ada Boost Tomp. Sampling
Ttest_indResult(statistic=-0.85939552687964849, pvalue=0.39693776563804484)

 Ada Boost Hedge
Ttest_indResult(statistic=-0.89801934228852487, pvalue=0.37632624152905847)

 Ada Boost EXP3
Ttest_indResult(statistic=-0.53246481629506681, pvalue=0.59832615567319603)

 NB LDA
Ttest_indResult(statistic=-0.48733870104255722, pvalue=0.62956059302035561)

 NB QDA
Ttest_indResult(statistic=-0.4954509254174197, pvalue=0.62389173669745335)

 NB Log. Reg.
Ttest_indResult(statistic=-0.68281758946090032, pvalue=0.49996143603528487)

 NB GP Class.
Ttest_indResult(statistic=-1.9874291755456015, pvalue=0.056070886869985563)

 NB LightGBM
Ttest_indResult(statistic=-1.0324492601445696, pvalue=0.31011357753126279)

 NB Xgboost
Ttest_indResult(statistic=-1.458154853900431, pvalue=0.15518826973568864)

 NB NN
Ttest_indResult(statistic=-1.8566791996977694, pvalue=0.073201136550242837)
```

```
 NB UCB1
Ttest_indResult(statistic=-1.8678484640510886, pvalue=0.07157946067330194)

 NB E Greedy
Ttest_indResult(statistic=-1.5408971718106503, pvalue=0.13382490207683173)

 NB Decay E Gr.
Ttest_indResult(statistic=-1.7471513791903304, pvalue=0.090842508478082257)

 NB Softmax
Ttest_indResult(statistic=-1.5536904356494097, pvalue=0.13074553833706801)

 NB Decay SM
Ttest_indResult(statistic=-1.7443742328508862, pvalue=0.091332979516465837)

 NB Tomp. Sampling
Ttest_indResult(statistic=-1.434087743798873, pvalue=0.16188928027087141)

 NB Hedge
Ttest_indResult(statistic=-1.480355476686583, pvalue=0.14920487151044853)

 NB EXP3
Ttest_indResult(statistic=-1.0745050112715238, pvalue=0.29116301260414107)

 LDA QDA
Ttest_indResult(statistic=0.019672607741379806, pvalue=0.98443480388746196)

 LDA Log. Reg.
Ttest_indResult(statistic=-0.18993327572107949, pvalue=0.85064075047074106)

 LDA GP Class.
Ttest_indResult(statistic=-1.322245113493338, pvalue=0.19607963189461339)

 LDA LightGBM
Ttest_indResult(statistic=-0.48307655938028249, pvalue=0.63254823920322223)

 LDA Xgboost
Ttest_indResult(statistic=-0.86947310862410776, pvalue=0.39149198152316989)

 LDA NN
Ttest_indResult(statistic=-1.1988371611951032, pvalue=0.23997553531254021)

 LDA UCB1
Ttest_indResult(statistic=-1.2105695319115684, pvalue=0.23551191106889277)

 LDA E Greedy
Ttest_indResult(statistic=-0.931960728183996, pvalue=0.35879690654201768)
```

```
 LDA Decay E Gr.
Ttest_indResult(statistic=-1.1115482908067353, pvalue=0.27516143424652995)

 LDA Softmax
Ttest_indResult(statistic=-0.92883249846214266, pvalue=0.36038960371550854)

 LDA Decay SM
Ttest_indResult(statistic=-1.0961715000664454, pvalue=0.28172560774357647)

 LDA Tomp. Sampling
Ttest_indResult(statistic=-0.811173149330364, pvalue=0.42365543603182987)

 LDA Hedge
Ttest_indResult(statistic=-0.84953792754942448, pvalue=0.40231089522715435)

 LDA EXP3
Ttest_indResult(statistic=-0.48446047759491634, pvalue=0.63157745566889378)

 QDA Log. Reg.
Ttest_indResult(statistic=-0.21903831900350959, pvalue=0.82810361331513782)

 QDA GP Class.
Ttest_indResult(statistic=-1.4330722791558868, pvalue=0.1621769866091399)

 QDA LightGBM
Ttest_indResult(statistic=-0.53184738593054781, pvalue=0.59874849729194424)

 QDA Xgboost
Ttest_indResult(statistic=-0.94276706669170351, pvalue=0.35333083218166617)

 QDA NN
Ttest_indResult(statistic=-1.3020819232662322, pvalue=0.20279740684955241)

 QDA UCB1
Ttest_indResult(statistic=-1.3142455588007629, pvalue=0.19872406927878292)

 QDA E Greedy
Ttest_indResult(statistic=-1.0120174283392591, pvalue=0.31962284051032974)

 QDA Decay E Gr.
Ttest_indResult(statistic=-1.2056308366728303, pvalue=0.23738327853162045)

 QDA Softmax
Ttest_indResult(statistic=-1.0118415573236754, pvalue=0.31970555455844063)

 QDA Decay SM
Ttest_indResult(statistic=-1.1922693120755423, pvalue=0.24250149519088826)
```

```
 QDA Tomp. Sampling
Ttest_indResult(statistic=-0.88784428857996878, pvalue=0.38168773235893938)

 QDA Hedge
Ttest_indResult(statistic=-0.929655494526203, pvalue=0.35997013475565298)

 QDA EXP3
Ttest_indResult(statistic=-0.53851220411109535, pvalue=0.59419708379618141)

 Log. Reg. GP Class.
Ttest_indResult(statistic=-1.0909477884770726, pvalue=0.2839807540273987)

 Log. Reg. LightGBM
Ttest_indResult(statistic=-0.27789263857170832, pvalue=0.78300102397151372)

 Log. Reg. Xgboost
Ttest_indResult(statistic=-0.6559476945377517, pvalue=0.5168574036624789)

 Log. Reg. NN
Ttest_indResult(statistic=-0.96853646655512182, pvalue=0.34052056123760877)

 Log. Reg. UCB1
Ttest_indResult(statistic=-0.98051924577857008, pvalue=0.33467154558521839)

 Log. Reg. E Greedy
Ttest_indResult(statistic=-0.71310181104268766, pvalue=0.4812933769062645)

 Log. Reg. Decay E Gr.
Ttest_indResult(statistic=-0.88666408855057544, pvalue=0.38231278171562644)

 Log. Reg. Softmax
Ttest_indResult(statistic=-0.70540712335018663, pvalue=0.48599851868759969)

 Log. Reg. Decay SM
Ttest_indResult(statistic=-0.86766064577916946, pvalue=0.39246788262354737)

 Log. Reg. Tomp. Sampling
Ttest_indResult(statistic=-0.58724541997566204, pvalue=0.5614329982095807)

 Log. Reg. Hedge
Ttest_indResult(statistic=-0.62371815853394141, pvalue=0.53752670649819234)

 Log. Reg. EXP3
Ttest_indResult(statistic=-0.26697372915008794, pvalue=0.79131608695232525)

 GP Class. LightGBM
Ttest_indResult(statistic=0.85699380913612311, pvalue=0.3982426740468038)
```

```
 GP Class. Xgboost
Ttest_indResult(statistic=0.4428277371929491, pvalue=0.6610670474690763)

 GP Class. NN
Ttest_indResult(statistic=0.14811612219406103, pvalue=0.88324220227418104)

 GP Class. UCB1
Ttest_indResult(statistic=0.13174987657535231, pvalue=0.89606177243517682)

 GP Class. E Greedy
Ttest_indResult(statistic=0.40336304982051585, pvalue=0.68954174881134001)

 GP Class. Decay E Gr.
Ttest_indResult(statistic=0.22128109309172209, pvalue=0.82637284156927215)

 GP Class. Softmax
Ttest_indResult(statistic=0.43989510805755366, pvalue=0.66316603403905106)

 GP Class. Decay SM
Ttest_indResult(statistic=0.26513789749604422, pvalue=0.79271660862095783)

 GP Class. Tomp. Sampling
Ttest_indResult(statistic=0.59212418182365922, pvalue=0.55820409867910459)

 GP Class. Hedge
Ttest_indResult(statistic=0.5576792838392387, pvalue=0.58120134907479459)

 GP Class. EXP3
Ttest_indResult(statistic=0.956168912787559, pvalue=0.34662913716237898)

 LightGBM Xgboost
Ttest_indResult(statistic=-0.40005437681209527, pvalue=0.69195089562483136)

 LightGBM NN
Ttest_indResult(statistic=-0.72435824281960381, pvalue=0.47445746413452261)

 LightGBM UCB1
Ttest_indResult(statistic=-0.73762456676282273, pvalue=0.46647336161486808)

 LightGBM E Greedy
Ttest_indResult(statistic=-0.45656048508152242, pvalue=0.65127547880603243)

 LightGBM Decay E Gr.
Ttest_indResult(statistic=-0.63969263344111615, pvalue=0.52722772159940889)

 LightGBM Softmax
Ttest_indResult(statistic=-0.44352126606233261, pvalue=0.66057107261962511)
```

```
 LightGBM Decay SM
Ttest_indResult(statistic=-0.61579193861328929, pvalue=0.54267600983406017)

 LightGBM Tomp. Sampling
Ttest_indResult(statistic=-0.3138177790956009, pvalue=0.7558312265281264)

 LightGBM Hedge
Ttest_indResult(statistic=-0.3516720690223292, pvalue=0.7275423817388138)

 LightGBM EXP3
Ttest_indResult(statistic=0.029112831118137433, pvalue=0.97696741607653625)

 Xgboost NN
Ttest_indResult(statistic=-0.30591666602016748, pvalue=0.76178097045239601)

 Xgboost UCB1
Ttest_indResult(statistic=-0.32028550568424718, pvalue=0.75097213976644317)

 Xgboost E Greedy
Ttest_indResult(statistic=-0.048270716862544541, pvalue=0.96182042292422421)

 Xgboost Decay E Gr.
Ttest_indResult(statistic=-0.22780269666179409, pvalue=0.82134511736644389)

 Xgboost Softmax
Ttest_indResult(statistic=-0.025420174014263292, pvalue=0.97988816023468628)

 Xgboost Decay SM
Ttest_indResult(statistic=-0.19570029766273139, pvalue=0.84616428221282891)

 Xgboost Tomp. Sampling
Ttest_indResult(statistic=0.1115527986197322, pvalue=0.91192102733277625)

 Xgboost Hedge
Ttest_indResult(statistic=0.075773973735467515, pvalue=0.94010198687354585)

 Xgboost EXP3
Ttest_indResult(statistic=0.45758275530333697, pvalue=0.65054908455057037)

 NN UCB1
Ttest_indResult(statistic=-0.016090338142547938, pvalue=0.9872688563892611)

 NN E Greedy
Ttest_indResult(statistic=0.26264755413119467, pvalue=0.79461756774775738)

 NN Decay E Gr.
Ttest_indResult(statistic=0.076916764404492938, pvalue=0.93920045612897018)
```

```
 NN Softmax
Ttest_indResult(statistic=0.29553608195629255, pvalue=0.76962031063909875)

 NN Decay SM
Ttest_indResult(statistic=0.11778745073392619, pvalue=0.90702117188526632)

 NN Tomp. Sampling
Ttest_indResult(statistic=0.44659572154732885, pvalue=0.65837427449331354)

 NN Hedge
Ttest_indResult(statistic=0.41087231543542685, pvalue=0.68408630799466097)

 NN EXP3
Ttest_indResult(statistic=0.81370818222455565, pvalue=0.42222389713437691)

 UCB1 E Greedy
Ttest_indResult(statistic=0.27747846159164308, pvalue=0.7833159637826892)

 UCB1 Decay E Gr.
Ttest_indResult(statistic=0.092459911433430672, pvalue=0.92694732846058914)

 UCB1 Softmax
Ttest_indResult(statistic=0.31067955915226586, pvalue=0.75819259024466024)

 UCB1 Decay SM
Ttest_indResult(statistic=0.13357357520705196, pvalue=0.89463180288650324)

 UCB1 Tomp. Sampling
Ttest_indResult(statistic=0.46157109130443635, pvalue=0.64771843240403992)

 UCB1 Hedge
Ttest_indResult(statistic=0.42604173065720125, pvalue=0.67311875078463379)

 UCB1 EXP3
Ttest_indResult(statistic=0.8276953573084499, pvalue=0.41437900225642132)

 E Greedy Decay E Gr.
Ttest_indResult(statistic=-0.18327473486897286, pvalue=0.85581552073786515)

 E Greedy Softmax
Ttest_indResult(statistic=0.024631632303294463, pvalue=0.98051190331278504)

 E Greedy Decay SM
Ttest_indResult(statistic=-0.14952239861397326, pvalue=0.88214212144032023)

 E Greedy Tomp. Sampling
Ttest_indResult(statistic=0.16585725369711701, pvalue=0.8693820010902229)
```

```
 E Greedy Hedge
Ttest_indResult(statistic=0.12951918808142254, pvalue=0.8978113464144372)

 E Greedy EXP3
Ttest_indResult(statistic=0.5204318013634659, pvalue=0.60658257502292456)

 Decay E Gr. Softmax
Ttest_indResult(statistic=0.213191294870135, pvalue=0.8326199257630259)

 Decay E Gr. Decay SM
Ttest_indResult(statistic=0.038253180565644182, pvalue=0.9697392473945281)

 Decay E Gr. Tomp. Sampling
Ttest_indResult(statistic=0.35965046597145339, pvalue=0.72162796021578823)

 Decay E Gr. Hedge
Ttest_indResult(statistic=0.32403475059921522, pvalue=0.74816012853507807)

 Decay E Gr. EXP3
Ttest_indResult(statistic=0.71939872374324976, pvalue=0.47746240357037417)

 Softmax Decay SM
Ttest_indResult(statistic=-0.17926308915663908, pvalue=0.85893638567896469)

 Softmax Tomp. Sampling
Ttest_indResult(statistic=0.14475805017259508, pvalue=0.88587005757440984)

 Softmax Hedge
Ttest_indResult(statistic=0.10729202823285981, pvalue=0.91527163297267178)

 Softmax EXP3
Ttest_indResult(statistic=0.50893750097126389, pvalue=0.61451897458160143)

 Decay SM Tomp. Sampling
Ttest_indResult(statistic=0.32856719724887984, pvalue=0.74476539708479117)

 Decay SM Hedge
Ttest_indResult(statistic=0.29197268896011197, pvalue=0.77231712045755474)

 Decay SM EXP3
Ttest_indResult(statistic=0.69631929567434947, pvalue=0.49158905384005269)

 Tomp. Sampling Hedge
Ttest_indResult(statistic=-0.039069676504578377, pvalue=0.96909368199604229)

 Tomp. Sampling EXP3
Ttest_indResult(statistic=0.37024011082824332, pvalue=0.71380479379732176)
```

```
 Hedge EXP3
Ttest_indResult(statistic=0.41172935534195504, pvalue=0.68346476619757235)
```

```
In [19]: for pair, p in zip(new_model_pairs, pvalueList):
             if p < 0.05:
                 print('The pvalue between',pair, 'is', p, '< 0.05 then',
                       emoji.emojize('REJECT the NULL Hypothesis :thumbs_up_sign:'))
             else:
                 print('The pvalue between',pair, 'is', p, '> 0.05 then',
                       emoji.emojize('FAIL to REJECT the NULL Hypothesis :thumbs_down_sign:'))
```

```
The pvalue between ('Knn', 'NN') is 0.862047792635 > 0.05 then FAIL to REJECT the NULL Hypothesi
The pvalue between ('Knn', 'UCB1') is 0.84931999535 > 0.05 then FAIL to REJECT the NULL Hypothes
The pvalue between ('Knn', 'E Greedy') is 0.915893799037 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('Knn', 'Decay E Gr.') is 0.928446515155 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Knn', 'Softmax') is 0.892331256012 > 0.05 then FAIL to REJECT the NULL Hypo
The pvalue between ('Knn', 'Decay SM') is 0.959026091134 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('Knn', 'Tomp. Sampling') is 0.772045788365 > 0.05 then FAIL to REJECT the NU
The pvalue between ('Knn', 'Hedge') is 0.801377559994 > 0.05 then FAIL to REJECT the NULL Hypoth
The pvalue between ('Knn', 'EXP3') is 0.503454093937 > 0.05 then FAIL to REJECT the NULL Hypothe
The pvalue between ('LSVM', 'NN') is 0.282584328463 > 0.05 then FAIL to REJECT the NULL Hypothes
The pvalue between ('LSVM', 'UCB1') is 0.277514046468 > 0.05 then FAIL to REJECT the NULL Hypoth
The pvalue between ('LSVM', 'E Greedy') is 0.411352105148 > 0.05 then FAIL to REJECT the NULL Hy
The pvalue between ('LSVM', 'Decay E Gr.') is 0.320765878671 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('LSVM', 'Softmax') is 0.414270136904 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('LSVM', 'Decay SM') is 0.328871048368 > 0.05 then FAIL to REJECT the NULL Hy
The pvalue between ('LSVM', 'Tomp. Sampling') is 0.483051678686 > 0.05 then FAIL to REJECT the N
The pvalue between ('LSVM', 'Hedge') is 0.46050148701 > 0.05 then FAIL to REJECT the NULL Hypoth
The pvalue between ('LSVM', 'EXP3') is 0.701404128726 > 0.05 then FAIL to REJECT the NULL Hypoth
The pvalue between ('SVM', 'NN') is 0.563365630934 > 0.05 then FAIL to REJECT the NULL Hypothesi
The pvalue between ('SVM', 'UCB1') is 0.575649859942 > 0.05 then FAIL to REJECT the NULL Hypothe
The pvalue between ('SVM', 'E Greedy') is 0.412407424072 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('SVM', 'Decay E Gr.') is 0.520698174068 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('SVM', 'Softmax') is 0.384218567755 > 0.05 then FAIL to REJECT the NULL Hypo
The pvalue between ('SVM', 'Decay SM') is 0.485115295978 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('SVM', 'Tomp. Sampling') is 0.302065765409 > 0.05 then FAIL to REJECT the NU
The pvalue between ('SVM', 'Hedge') is 0.316823672385 > 0.05 then FAIL to REJECT the NULL Hypoth
The pvalue between ('SVM', 'EXP3') is 0.16372127814 > 0.05 then FAIL to REJECT the NULL Hypothes
The pvalue between ('DT_gini', 'NN') is 0.649634964992 > 0.05 then FAIL to REJECT the NULL Hypot
The pvalue between ('DT_gini', 'UCB1') is 0.638565337827 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('DT_gini', 'E Greedy') is 0.871224014298 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('DT_gini', 'Decay E Gr.') is 0.716313689826 > 0.05 then FAIL to REJECT the N
The pvalue between ('DT_gini', 'Softmax') is 0.8884116358 > 0.05 then FAIL to REJECT the NULL Hy
The pvalue between ('DT_gini', 'Decay SM') is 0.740135463391 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('DT_gini', 'Tomp. Sampling') is 0.991659798428 > 0.05 then FAIL to REJECT th
The pvalue between ('DT_gini', 'Hedge') is 0.975906220882 > 0.05 then FAIL to REJECT the NULL Hy
The pvalue between ('DT_gini', 'EXP3') is 0.692689023934 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('DT_entorpy', 'NN') is 0.588979945462 > 0.05 then FAIL to REJECT the NULL Hy
The pvalue between ('DT_entorpy', 'UCB1') is 0.579067882459 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('DT_entorpy', 'E Greedy') is 0.794476096492 > 0.05 then FAIL to REJECT the N
The pvalue between ('DT_entorpy', 'Decay E Gr.') is 0.650609122568 > 0.05 then FAIL to REJECT th
The pvalue between ('DT_entorpy', 'Softmax') is 0.808561078856 > 0.05 then FAIL to REJECT the NU
The pvalue between ('DT_entorpy', 'Decay SM') is 0.671218294801 > 0.05 then FAIL to REJECT the N
The pvalue between ('DT_entorpy', 'Tomp. Sampling') is 0.919620787414 > 0.05 then FAIL to REJECT
The pvalue between ('DT_entorpy', 'Hedge') is 0.888507254263 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('DT_entorpy', 'EXP3') is 0.790976144273 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Bagging Knn', 'NN') is 0.391194869135 > 0.05 then FAIL to REJECT the NULL H
The pvalue between ('Bagging Knn', 'UCB1') is 0.384168020604 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Bagging Knn', 'E Greedy') is 0.557781361646 > 0.05 then FAIL to REJECT the
The pvalue between ('Bagging Knn', 'Decay E Gr.') is 0.440907114267 > 0.05 then FAIL to REJECT t
The pvalue between ('Bagging Knn', 'Softmax') is 0.564311705507 > 0.05 then FAIL to REJECT the N
The pvalue between ('Bagging Knn', 'Decay SM') is 0.453393521386 > 0.05 then FAIL to REJECT the
The pvalue between ('Bagging Knn', 'Tomp. Sampling') is 0.653852579758 > 0.05 then FAIL to REJEC
The pvalue between ('Bagging Knn', 'Hedge') is 0.626076199106 > 0.05 then FAIL to REJECT the NUL
The pvalue between ('Bagging Knn', 'EXP3') is 0.918012672984 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Bagging DT', 'NN') is 0.410388610133 > 0.05 then FAIL to REJECT the NULL Hy
The pvalue between ('Bagging DT', 'UCB1') is 0.403075781035 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Bagging DT', 'E Greedy') is 0.58126761888 > 0.05 then FAIL to REJECT the NU
The pvalue between ('Bagging DT', 'Decay E Gr.') is 0.461410608719 > 0.05 then FAIL to REJECT th
The pvalue between ('Bagging DT', 'Softmax') is 0.588490664422 > 0.05 then FAIL to REJECT the NU
The pvalue between ('Bagging DT', 'Decay SM') is 0.474660468593 > 0.05 then FAIL to REJECT the N
The pvalue between ('Bagging DT', 'Tomp. Sampling') is 0.680406524776 > 0.05 then FAIL to REJECT
The pvalue between ('Bagging DT', 'Hedge') is 0.652183621901 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Bagging DT', 'EXP3') is 0.948182830846 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Random Forest', 'NN') is 0.908182813157 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Random Forest', 'UCB1') is 0.921040940773 > 0.05 then FAIL to REJECT the NU
The pvalue between ('Random Forest', 'E Greedy') is 0.711246862551 > 0.05 then FAIL to REJECT th
The pvalue between ('Random Forest', 'Decay E Gr.') is 0.850243942252 > 0.05 then FAIL to REJECT
The pvalue between ('Random Forest', 'Softmax') is 0.685014750374 > 0.05 then FAIL to REJECT the
The pvalue between ('Random Forest', 'Decay SM') is 0.816739142671 > 0.05 then FAIL to REJECT th
The pvalue between ('Random Forest', 'Tomp. Sampling') is 0.578310116032 > 0.05 then FAIL to REJ
The pvalue between ('Random Forest', 'Hedge') is 0.601941359856 > 0.05 then FAIL to REJECT the N
The pvalue between ('Random Forest', 'EXP3') is 0.361052638737 > 0.05 then FAIL to REJECT the NU
The pvalue between ('Ada Boost', 'NN') is 0.222036573368 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('Ada Boost', 'UCB1') is 0.217857155654 > 0.05 then FAIL to REJECT the NULL H
The pvalue between ('Ada Boost', 'E Greedy') is 0.335542477528 > 0.05 then FAIL to REJECT the NU
The pvalue between ('Ada Boost', 'Decay E Gr.') is 0.255619341007 > 0.05 then FAIL to REJECT the
The pvalue between ('Ada Boost', 'Softmax') is 0.336603397955 > 0.05 then FAIL to REJECT the NUL
The pvalue between ('Ada Boost', 'Decay SM') is 0.261527299206 > 0.05 then FAIL to REJECT the NU
The pvalue between ('Ada Boost', 'Tomp. Sampling') is 0.396937765638 > 0.05 then FAIL to REJECT
The pvalue between ('Ada Boost', 'Hedge') is 0.376326241529 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Ada Boost', 'EXP3') is 0.598326155673 > 0.05 then FAIL to REJECT the NULL H
The pvalue between ('NB', 'NN') is 0.0732011365502 > 0.05 then FAIL to REJECT the NULL Hypothesi
The pvalue between ('NB', 'UCB1') is 0.0715794606733 > 0.05 then FAIL to REJECT the NULL Hypothe
The pvalue between ('NB', 'E Greedy') is 0.133824902077 > 0.05 then FAIL to REJECT the NULL Hypo
The pvalue between ('NB', 'Decay E Gr.') is 0.0908425084781 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('NB', 'Softmax') is 0.130745538337 > 0.05 then FAIL to REJECT the NULL Hypot
The pvalue between ('NB', 'Decay SM') is 0.0913329795165 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('NB', 'Tomp. Sampling') is 0.161889280271 > 0.05 then FAIL to REJECT the NUL
The pvalue between ('NB', 'Hedge') is 0.14920487151 > 0.05 then FAIL to REJECT the NULL Hypothes
The pvalue between ('NB', 'EXP3') is 0.291163012604 > 0.05 then FAIL to REJECT the NULL Hypothes
The pvalue between ('LDA', 'NN') is 0.239975535313 > 0.05 then FAIL to REJECT the NULL Hypothesi
The pvalue between ('LDA', 'UCB1') is 0.235511911069 > 0.05 then FAIL to REJECT the NULL Hypothe
The pvalue between ('LDA', 'E Greedy') is 0.358796906542 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('LDA', 'Decay E Gr.') is 0.275161434247 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('LDA', 'Softmax') is 0.360389603716 > 0.05 then FAIL to REJECT the NULL Hypo
The pvalue between ('LDA', 'Decay SM') is 0.281725607744 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('LDA', 'Tomp. Sampling') is 0.423655436032 > 0.05 then FAIL to REJECT the NU
The pvalue between ('LDA', 'Hedge') is 0.402310895227 > 0.05 then FAIL to REJECT the NULL Hypoth
The pvalue between ('LDA', 'EXP3') is 0.631577455669 > 0.05 then FAIL to REJECT the NULL Hypothe
The pvalue between ('QDA', 'NN') is 0.20279740685 > 0.05 then FAIL to REJECT the NULL Hypothesis
The pvalue between ('QDA', 'UCB1') is 0.198724069279 > 0.05 then FAIL to REJECT the NULL Hypothe
The pvalue between ('QDA', 'E Greedy') is 0.31962284051 > 0.05 then FAIL to REJECT the NULL Hypo
The pvalue between ('QDA', 'Decay E Gr.') is 0.237383278532 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('QDA', 'Softmax') is 0.319705554558 > 0.05 then FAIL to REJECT the NULL Hypo
The pvalue between ('QDA', 'Decay SM') is 0.242501495191 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('QDA', 'Tomp. Sampling') is 0.381687732359 > 0.05 then FAIL to REJECT the NU
The pvalue between ('QDA', 'Hedge') is 0.359970134756 > 0.05 then FAIL to REJECT the NULL Hypoth
The pvalue between ('QDA', 'EXP3') is 0.594197083796 > 0.05 then FAIL to REJECT the NULL Hypothe
The pvalue between ('Log. Reg.', 'NN') is 0.340520561238 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('Log. Reg.', 'UCB1') is 0.334671545585 > 0.05 then FAIL to REJECT the NULL H
The pvalue between ('Log. Reg.', 'E Greedy') is 0.481293376906 > 0.05 then FAIL to REJECT the NU
The pvalue between ('Log. Reg.', 'Decay E Gr.') is 0.382312781716 > 0.05 then FAIL to REJECT the
The pvalue between ('Log. Reg.', 'Softmax') is 0.485998518688 > 0.05 then FAIL to REJECT the NUL
The pvalue between ('Log. Reg.', 'Decay SM') is 0.392467882624 > 0.05 then FAIL to REJECT the NU
The pvalue between ('Log. Reg.', 'Tomp. Sampling') is 0.56143299821 > 0.05 then FAIL to REJECT t
The pvalue between ('Log. Reg.', 'Hedge') is 0.537526706498 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Log. Reg.', 'EXP3') is 0.791316086952 > 0.05 then FAIL to REJECT the NULL H
The pvalue between ('GP Class.', 'NN') is 0.883242202274 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('GP Class.', 'UCB1') is 0.896061772435 > 0.05 then FAIL to REJECT the NULL H
The pvalue between ('GP Class.', 'E Greedy') is 0.689541748811 > 0.05 then FAIL to REJECT the NU
The pvalue between ('GP Class.', 'Decay E Gr.') is 0.826372841569 > 0.05 then FAIL to REJECT the
The pvalue between ('GP Class.', 'Softmax') is 0.663166034039 > 0.05 then FAIL to REJECT the NUL
The pvalue between ('GP Class.', 'Decay SM') is 0.792716608621 > 0.05 then FAIL to REJECT the NU
The pvalue between ('GP Class.', 'Tomp. Sampling') is 0.558204098679 > 0.05 then FAIL to REJECT
The pvalue between ('GP Class.', 'Hedge') is 0.581201349075 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('GP Class.', 'EXP3') is 0.346629137162 > 0.05 then FAIL to REJECT the NULL H
The pvalue between ('LightGBM', 'NN') is 0.474457464135 > 0.05 then FAIL to REJECT the NULL Hypo
The pvalue between ('LightGBM', 'UCB1') is 0.466473361615 > 0.05 then FAIL to REJECT the NULL Hy
The pvalue between ('LightGBM', 'E Greedy') is 0.651275478806 > 0.05 then FAIL to REJECT the NUL
The pvalue between ('LightGBM', 'Decay E Gr.') is 0.527227721599 > 0.05 then FAIL to REJECT the
The pvalue between ('LightGBM', 'Softmax') is 0.66057107262 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('LightGBM', 'Decay SM') is 0.542676009834 > 0.05 then FAIL to REJECT the NUL
The pvalue between ('LightGBM', 'Tomp. Sampling') is 0.755831226528 > 0.05 then FAIL to REJECT t
The pvalue between ('LightGBM', 'Hedge') is 0.727542381739 > 0.05 then FAIL to REJECT the NULL H
The pvalue between ('LightGBM', 'EXP3') is 0.976967416077 > 0.05 then FAIL to REJECT the NULL Hy
The pvalue between ('Xgboost', 'NN') is 0.761780970452 > 0.05 then FAIL to REJECT the NULL Hypot
The pvalue between ('Xgboost', 'UCB1') is 0.750972139766 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('Xgboost', 'E Greedy') is 0.961820422924 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Xgboost', 'Decay E Gr.') is 0.821345117366 > 0.05 then FAIL to REJECT the N
The pvalue between ('Xgboost', 'Softmax') is 0.979888160235 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Xgboost', 'Decay SM') is 0.846164282213 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Xgboost', 'Tomp. Sampling') is 0.911921027333 > 0.05 then FAIL to REJECT th
The pvalue between ('Xgboost', 'Hedge') is 0.940101986874 > 0.05 then FAIL to REJECT the NULL Hy
The pvalue between ('Xgboost', 'EXP3') is 0.650549084551 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('NN', 'UCB1') is 0.987268856389 > 0.05 then FAIL to REJECT the NULL Hypothes
The pvalue between ('NN', 'E Greedy') is 0.794617567748 > 0.05 then FAIL to REJECT the NULL Hypo
The pvalue between ('NN', 'Decay E Gr.') is 0.939200456129 > 0.05 then FAIL to REJECT the NULL H
The pvalue between ('NN', 'Softmax') is 0.769620310639 > 0.05 then FAIL to REJECT the NULL Hypot
The pvalue between ('NN', 'Decay SM') is 0.907021171885 > 0.05 then FAIL to REJECT the NULL Hypo
The pvalue between ('NN', 'Tomp. Sampling') is 0.658374274493 > 0.05 then FAIL to REJECT the NUL
The pvalue between ('NN', 'Hedge') is 0.684086307995 > 0.05 then FAIL to REJECT the NULL Hypothe
The pvalue between ('NN', 'EXP3') is 0.422223897134 > 0.05 then FAIL to REJECT the NULL Hypothes
The pvalue between ('UCB1', 'E Greedy') is 0.783315963783 > 0.05 then FAIL to REJECT the NULL Hy
The pvalue between ('UCB1', 'Decay E Gr.') is 0.926947328461 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('UCB1', 'Softmax') is 0.758192590245 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('UCB1', 'Decay SM') is 0.894631802887 > 0.05 then FAIL to REJECT the NULL Hy
The pvalue between ('UCB1', 'Tomp. Sampling') is 0.647718432404 > 0.05 then FAIL to REJECT the N
The pvalue between ('UCB1', 'Hedge') is 0.673118750785 > 0.05 then FAIL to REJECT the NULL Hypot
The pvalue between ('UCB1', 'EXP3') is 0.414379002256 > 0.05 then FAIL to REJECT the NULL Hypoth
The pvalue between ('E Greedy', 'Decay E Gr.') is 0.855815520738 > 0.05 then FAIL to REJECT the
The pvalue between ('E Greedy', 'Softmax') is 0.980511903313 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('E Greedy', 'Decay SM') is 0.88214212144 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('E Greedy', 'Tomp. Sampling') is 0.86938200109 > 0.05 then FAIL to REJECT th
The pvalue between ('E Greedy', 'Hedge') is 0.897811346414 > 0.05 then FAIL to REJECT the NULL H
The pvalue between ('E Greedy', 'EXP3') is 0.606582575023 > 0.05 then FAIL to REJECT the NULL Hy
The pvalue between ('Decay E Gr.', 'Softmax') is 0.832619925763 > 0.05 then FAIL to REJECT the N
The pvalue between ('Decay E Gr.', 'Decay SM') is 0.969739247395 > 0.05 then FAIL to REJECT the
The pvalue between ('Decay E Gr.', 'Tomp. Sampling') is 0.721627960216 > 0.05 then FAIL to REJEC
The pvalue between ('Decay E Gr.', 'Hedge') is 0.748160128535 > 0.05 then FAIL to REJECT the NUL
The pvalue between ('Decay E Gr.', 'EXP3') is 0.47746240357 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Softmax', 'Decay SM') is 0.858936385679 > 0.05 then FAIL to REJECT the NULL
The pvalue between ('Softmax', 'Tomp. Sampling') is 0.885870057574 > 0.05 then FAIL to REJECT th
The pvalue between ('Softmax', 'Hedge') is 0.915271632973 > 0.05 then FAIL to REJECT the NULL Hy
The pvalue between ('Softmax', 'EXP3') is 0.614518974582 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('Decay SM', 'Tomp. Sampling') is 0.744765397085 > 0.05 then FAIL to REJECT t
The pvalue between ('Decay SM', 'Hedge') is 0.772317120458 > 0.05 then FAIL to REJECT the NULL H
The pvalue between ('Decay SM', 'EXP3') is 0.49158905384 > 0.05 then FAIL to REJECT the NULL Hyp
The pvalue between ('Tomp. Sampling', 'Hedge') is 0.969093681996 > 0.05 then FAIL to REJECT the
The pvalue between ('Tomp. Sampling', 'EXP3') is 0.713804793797 > 0.05 then FAIL to REJECT the N
The pvalue between ('Hedge', 'EXP3') is 0.683464766198 > 0.05 then FAIL to REJECT the NULL Hypot
```