```
In [1]:
         import pandas as pd
         import numpy as np
         import keras
          import tensorflow
          import seaborn as sns
In [2]:
         data = pd.read csv("gas turbines.csv")
In [3]:
         data
                                                          TAT
                                                                        CDP
                                                                                CO
                                                                                     NOX
Out[3]:
                                AH AFDP GTEP
                                                    TIT
                                                                 TEY
            0 6.8594 1007.9 96.799 3.5000 19.663 1059.2 550.00 114.70 10.605 3.1547 82.722
            1 6.7850 1008.4 97.118 3.4998 19.728 1059.3 550.00 114.72 10.598 3.2363 82.776
            2 6.8977 1008.8 95.939 3.4824 19.779 1059.4 549.87 114.71 10.601 3.2012 82.468
            3 7.0569 1009.2 95.249 3.4805 19.792 1059.6 549.99 114.72 10.606 3.1923 82.670
             4 7 3978 1009 7 95 150 3 4976 19 765 1059 7 549 98 114 72 10 612 3 2484 82 311
         15034 9.0301 1005.6 98.460 3.5421 19.164 1049.7 546.21 111.61 10.400 4.5186 79.559
         15035 7.8879 1005.9 99.093 3.5059 19.414 1046.3 543.22 111.78 10.433 4.8470 79.917
         15036 7.2647 1006.3 99.496 3.4770 19.530 1037.7 537.32 110.19 10.483 7.9632 90.912
         15037 7.0060 1006.8 99.008 3.4486 19.377 1043.2 541.24 110.74 10.533 6.2494 93.227
         15038 6.9279 1007.2 97.533 3.4275 19.306 1049.9 545.85 111.58 10.583 4.9816 92.498
        15039 rows × 11 columns
```

## **EDA**

```
In [4]: data.info()
```

RangeIndex: 15039 entries, 0 to 15038 Data columns (total 11 columns): Column Non-Null Count Dtype 0 ΑT 15039 non-null float64 AP 15039 non-null float64 15039 non-null float64 2 AH AFDP 3 15039 non-null float64 4 **GTEP** 15039 non-null float64 TIT 15039 non-null float64 6 TAT 15039 non-null float64 7 TEY 15039 non-null float64 CDP 15039 non-null float64 15039 non-null float64 15039 non-null float64 9 C0 10 NOX

<class 'pandas.core.frame.DataFrame'>

dtypes: float64(11)
memory usage: 1.3 MB

Out[6]:

```
In [6]: data.describe()
```

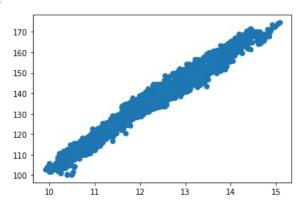
	AT	AP	АН	AFDP	GTEP	TIT	TAT	TEY	CDP	
count	15039.000000	15039.00000	15039.000000	15039.000000	15039.000000	15039.000000	15039.000000	15039.000000	15039.000000	15039.00
mean	17.764381	1013.19924	79.124174	4.200294	25.419061	1083.798770	545.396183	134.188464	12.102353	1.97
std	7.574323	6.41076	13.793439	0.760197	4.173916	16.527806	7.866803	15.829717	1.103196	2.22
min	0.522300	985.85000	30.344000	2.087400	17.878000	1000.800000	512.450000	100.170000	9.904400	0.00
25%	11.408000	1008.90000	69.750000	3.723900	23.294000	1079.600000	542.170000	127.985000	11.622000	0.85
50%	18.186000	1012.80000	82.266000	4.186200	25.082000	1088.700000	549.890000	133.780000	12.025000	1.39
75%	23.862500	1016.90000	90.043500	4.550900	27.184000	1096.000000	550.060000	140.895000	12.578000	2.16
max	34.929000	1034.20000	100.200000	7.610600	37.402000	1100.800000	550.610000	174.610000	15.081000	44.10
4										

```
In [7]: data.shape
```

Out[7]: (15039, 11)

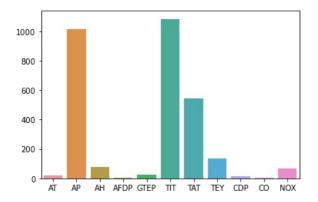
```
import matplotlib.pyplot as plt
plt.scatter(x="CDP",y="TEY",data=data)
```

Out[9]: <matplotlib.collections.PathCollection at 0x1f05b861880>



```
In [10]: sns.barplot(data=data)
```

## Out[10]: <AxesSubplot:>



```
In [11]: from sklearn.preprocessing import StandardScaler
```

In [12]:
 scaler = StandardScaler()
 standardized\_data=scaler.fit\_transform(data)
 df=pd.DataFrame(standardized\_data, columns=data.columns)

```
In [13]: d
```

df

Out[13]:		AT	AP	АН	AFDP	GTEP	TIT	TAT	TEY	CDP	со	NOX
	0	-1.439778	-0.826644	1.281436	-0.921232	-1.379101	-1.488376	0.585240	-1.231172	-1.357331	0.532012	1.387845
	1	-1.449601	-0.748647	1.304564	-0.921495	-1.363528	-1.482325	0.585240	-1.229909	-1.363676	0.568733	1.393002
	2	-1.434721	-0.686250	1.219086	-0.944385	-1.351309	-1.476275	0.568715	-1.230541	-1.360957	0.552938	1.363586
	3	-1.413702	-0.623853	1.169060	-0.946884	-1.348194	-1.464173	0.583969	-1.229909	-1.356424	0.548933	1.382878
	4	-1.368693	-0.545857	1.161883	-0.924389	-1.354663	-1.458123	0.582698	-1.229909	-1.350985	0.574179	1.348591
	15034	-1.153182	-1.185428	1.401860	-0.865850	-1.498657	-2.063184	0.103453	-1.426381	-1.543161	1.145792	1.085751
	15035	-1.303986	-1.138630	1.447753	-0.913470	-1.438759	-2.268905	-0.276638	-1.415642	-1.513247	1.293578	1.119943
	15036	-1.386267	-1.076233	1.476971	-0.951488	-1.410967	-2.789257	-1.026650	-1.516089	-1.467922	2.695925	2.170062
	15037	-1.420423	-0.998236	1.441590	-0.988848	-1.447624	-2.456474	-0.528337	-1.481343	-1.422598	1.924683	2.391165
	15038	-1.430734	-0.935839	1.334652	-1.016605	-1.464635	-2.051083	0.057689	-1.428277	-1.377273	1.354150	2.321539

```
In [14]:
           X=df.drop(["TEY"],axis=1)
           y=df["TEY"]
In [15]:
                      ΑT
                                ΑP
                                        AΗ
                                                AFDP
                                                         GTEP
                                                                     TIT
                                                                              TAT
                                                                                       CDP
                                                                                                 CO
                                                                                                         NOX
              0 -1.439778 -0.826644 1.281436 -0.921232 -1.379101 -1.488376 0.585240 -1.357331 0.532012 1.387845
              1 -1.449601 -0.748647 1.304564 -0.921495 -1.363528 -1.482325 0.585240 -1.363676 0.568733 1.393002
              2 -1.434721 -0.686250 1.219086 -0.944385 -1.351309 -1.476275 0.568715 -1.360957 0.552938 1.363586
              3 -1.413702 -0.623853 1.169060 -0.946884 -1.348194 -1.464173 0.583969 -1.356424 0.548933 1.382878
              4 -1.368693 -0.545857 1.161883 -0.924389 -1.354663 -1.458123 0.582698 -1.350985 0.574179 1.348591
          15034 -1.153182 -1.185428 1.401860 -0.865850 -1.498657 -2.063184 0.103453 -1.543161 1.145792 1.085751
          15035 -1.303986 -1.138630 1.447753 -0.913470 -1.438759 -2.268905 -0.276638 -1.513247 1.293578 1.119943
          15036 -1.386267 -1.076233 1.476971 -0.951488 -1.410967 -2.789257 -1.026650 -1.467922 2.695925 2.170062
          15037 -1.420423 -0.998236 1.441590 -0.988848 -1.447624 -2.456474 -0.528337 -1.422598 1.924683 2.391165
          15038 -1.430734 -0.935839 1.334652 -1.016605 -1.464635 -2.051083 0.057689 -1.377273 1.354150 2.321539
         15039 rows × 10 columns
In [16]:
                   -1.231172
Out[16]:
          1
                   -1.229909
                   -1.230541
          2
                   -1.229909
          3
                   -1.229909
          15034
                   -1.426381
          15035
                   -1.415642
          15036
                   -1.516089
          15037
                   -1.481343
          15038
                   -1.428277
          Name: TEY, Length: 15039, dtype: float64
In [17]:
           from sklearn.model_selection import GridSearchCV, KFold
           from keras.models import Sequential
           from keras.layers import Dense
           from keras.wrappers.scikit_learn import KerasClassifier
In [18]:
           from tensorflow.keras.optimizers import Adam, SGD, RMSprop
In [19]:
           n_features =X.shape[1]
           n features
Out[19]: 10
In [20]:
           # create model
           def create model():
               model = Sequential()
                model.add(Dense(12, input_dim=n_features, kernel_initializer='uniform', activation='relu'))
               model.add(Dense(8, kernel_initializer='uniform', activation='relu'))
model.add(Dense(1, kernel_initializer='uniform', activation='sigmoid'))
                optmizer =RMSprop(0.03)
               model.compile(loss='mean squared error', optimizer=optmizer, metrics=['accuracy'])
                return model
In [21]:
           model = KerasClassifier(build fn = create model, verbose = 0)
           batch_size = [10,20,40]
```

```
[CV 3/5; 1/9] START batch size=10, epochs=10......
[CV 3/5; 1/9] END .....batch_size=10, epochs=10;, score=0.000 total time= 9.9s
[CV 4/5; 1/9] START batch_size=10, epochs=10.....
[CV 4/5; 1/9] END .....batch_size=10, epochs=10;, score=0.000 total time= 10.2s
[CV 5/5; 1/9] START batch_size=10, epochs=10......
[CV 5/5; 1/9] END .....batch_size=10, epochs=10;, score=0.000 total time= 11.3s
[CV 1/5; 2/9] START batch size=10, epochs=50.....
[CV 1/5; 2/9] END .....batch_size=10, epochs=50;, score=0.000 total time= 57.9s
[CV 2/5; 2/9] START batch_size=10, epochs=50............
[CV 2/5; 2/9] END .....batch_size=10, epochs=50;, score=0.000 total time= 47.3s
[CV 4/5; 2/9] START batch_size=10, epochs=50......
[CV 4/5; 2/9] END .....batch_size=10, epochs=50;, score=0.000 total time= 48.9s
[CV 5/5; 2/9] START batch size=10, epochs=50......
[CV 5/5; 2/9] END .....batch_size=10, epochs=50;, score=0.000 total time= 55.9s
[CV 1/5; 3/9] START batch_size=10, epochs=100......
[CV 1/5; 3/9] END ....batch size=10, epochs=100;, score=0.000 total time= 1.6min
[CV 2/5; 3/9] END ....batch_size=10, epochs=100;, score=0.000 total time= 1.7min
[CV 3/5; 3/9] START batch size=10, epochs=100......
[CV 3/5; 3/9] END ....batch_size=10, epochs=100;, score=0.000 total time= 1.7min
[CV 4/5; 3/9] START batch_size=10, epochs=100......
[CV 4/5; 3/9] END ....batch_size=10, epochs=100;, score=0.000 total time= 2.0min
[CV 5/5; 3/9] START batch size=10, epochs=100.....
[CV 5/5; 3/9] END ....batch_size=10, epochs=100;, score=0.000 total time= 2.0min
[CV 1/5; 4/9] START batch_size=20, epochs=10.....
[CV 1/5; 4/9] END .....batch size=20, epochs=10;, score=0.000 total time= 8.2s
[CV 2/5; 4/9] START batch_size=20, epochs=10.....
[CV 2/5; 4/9] END .....batch_size=20, epochs=10;, score=0.000 total time= 5.8s
[CV 3/5; 4/9] START batch size=20, epochs=10.....
[CV 3/5; 4/9] END .....batch size=20, epochs=10;, score=0.000 total time= 6.3s
[CV 4/5; 4/9] START batch_size=20, epochs=10............
[CV 4/5; 4/9] END .....batch_size=20, epochs=10;, score=0.000 total time= 6.4s
[CV 5/5; 4/9] START batch size=20, epochs=10.....
[CV 5/5; 4/9] END .....batch_size=20, epochs=10;, score=0.000 total time= 6.9s
[CV 1/5; 5/9] START batch_size=20, epochs=50.....
[CV 1/5; 5/9] END .....batch size=20, epochs=50;, score=0.000 total time= 28.1s
[CV 2/5; 5/9] START batch size=20, epochs=50......
[CV 2/5; 5/9] END .....batch_size=20, epochs=50;, score=0.000 total time= 28.3s
[CV 3/5; 5/9] START batch_size=20, epochs=50.....
[CV 3/5; 5/9] END .....batch_size=20, epochs=50;, score=0.000 total time= 27.7s
[CV 4/5; 5/9] START batch_size=20, epochs=50.....
[CV 4/5; 5/9] END .....batch_size=20, epochs=50;, score=0.000 total time= 47.2s
[CV 5/5; 5/9] END .....batch_size=20, epochs=50;, score=0.000 total time= 47.9s
[CV 1/5; 6/9] START batch_size=20, epochs=100.....
[CV 1/5; 6/9] END ....batch_size=20, epochs=100;, score=0.000 total time= 1.1min
[CV 2/5; 6/9] START batch_size=20, epochs=100......
[CV 2/5; 6/9] END \dotsbatch_size=20, epochs=100;, score=0.000 total time= 1.1min
[CV 3/5; 6/9] START batch_size=20, epochs=100.....
[CV 3/5; 6/9] END ....batch size=20, epochs=100;, score=0.000 total time= 53.3s
[CV 4/5; 6/9] START batch size=20, epochs=100.....
[CV 4/5; 6/9] END ....batch_size=20, epochs=100;, score=0.000 total time= 1.3min
[CV 5/5; 6/9] START batch_size=20, epochs=100.....
[CV 5/5; 6/9] END ....batch_size=20, epochs=100;, score=0.000 total time= 55.7s
[CV 1/5; 7/9] START batch_size=40, epochs=10.....
[CV 1/5; 7/9] END .....batch_size=40, epochs=10;, score=0.000 total time= 3.5s
[CV 2/5; 7/9] START batch_size=40, epochs=10......
[CV 2/5; 7/9] END .....batch_size=40, epochs=10;, score=0.000 total time= 3.7s
[CV 3/5; 7/9] START batch_size=40, epochs=10.....
[CV 3/5; 7/9] END .....batch_size=40, epochs=10;, score=0.000 total time= 3.4s
[CV 4/5; 7/9] START batch size=40, epochs=10.....
[CV 4/5; 7/9] END .....batch_size=40, epochs=10;, score=0.000 total time= 3.4s
[CV 5/5; 7/9] START batch_size=40, epochs=10.....
[CV 5/5; 7/9] END .....batch_size=40, epochs=10;, score=0.000 total time= 3.4s
[CV 1/5; 8/9] START batch size=40, epochs=50......
[CV 1/5; 8/9] END .....batch_size=40, epochs=50;, score=0.000 total time= 15.3s
[CV 2/5; 8/9] START batch_size=40, epochs=50.....
[CV 2/5; 8/9] END .....batch size=40, epochs=50;, score=0.000 total time= 14.5s
[CV 3/5; 8/9] START batch_size=40, epochs=50.....
[CV 3/5; 8/9] END .....batch_size=40, epochs=50;, score=0.000 total time= 16.8s
[CV 4/5; 8/9] START batch_size=40, epochs=50.....
[CV 4/5; 8/9] END .....batch_size=40, epochs=50;, score=0.000 total time= 16.4s
[CV 5/5; 8/9] START batch_size=40, epochs=50.....
[CV 5/5; 8/9] END .....batch_size=40, epochs=50;, score=0.000 total time= 15.7s
```

```
[CV 1/5; 9/9] START batch_size=40, epochs=100......
               [CV 1/5; 9/9] END ....batch_size=40, epochs=100;, score=0.000 total time= 39.4s
                [CV 2/5; 9/9] START batch_size=40, epochs=100.....
               [CV 2/5; 9/9] END ....batch size=40, epochs=100;, score=0.000 total time= 38.0s
               [CV 3/5; 9/9] START batch size=40, epochs=100.....
               [CV 3/5; 9/9] END ....batch_size=40, epochs=100;, score=0.000 total time= 38.4s
               [CV 4/5; 9/9] START batch size=40, epochs=100......
               [CV 4/5; 9/9] END ....batch_size=40, epochs=100;, score=0.000 total time= 30.3s
               [CV 5/5; 9/9] START batch_size=40, epochs=100.....
               [CV 5/5; 9/9] END ....batch_size=40, epochs=100;, score=0.000 total time= 24.8s
In [22]: # Summarize the results
                print('Best : {}, using {}'.format(grid result.best score ,grid result.best params ))
                means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
                params = grid result.cv results ['params']
                for mean, stdev, param in zip(means, stds, params):
    print('{},{} with: {}'.format(mean, stdev, param))
               Best : 0.0, using {'batch_size': 10, 'epochs': 10}
               0.0,0.0 with: {'batch_size': 10, 'epochs': 10}
0.0,0.0 with: {'batch_size': 10, 'epochs': 50}
               0.0,0.0 with: {'batch_size': 10, 'epochs': 100} 0.0,0.0 with: {'batch_size': 20, 'epochs': 10}
               0.0,0.0 with: {'batch_size': 20, 'epochs': 50}
               0.0,0.0 with: {'batch_size': 20, 'epochs': 100}
0.0,0.0 with: {'batch_size': 40, 'epochs': 10}
               0.0,0.0 with: {'batch_size': 40, 'epochs': 50}
               0.0,0.0 with: {'batch_size': 40, 'epochs': 100}
In [23]:
                from keras.layers import Dropout
In [24]:
                def create model(learning rate, dropout rate):
                       model = Sequential()
                       model.add(Dense(8,input\_dim = n\_features,kernel\_initializer = 'normal',activation = 'relu'))
                       model.add(Dropout(dropout_rate))
                       model.add(Dense(4,input dim = n features,kernel initializer = 'normal',activation = 'relu'))
                       model.add(Dropout(dropout_rate))
                       model.add(Dense(1,activation = 'sigmoid'))
                       adam = Adam(lr = learning_rate)
                       model.compile(loss='mean_squared_error',optimizer = adam,metrics = ['accuracy'])
                       return model
In [25]:
                # Create the model
                model = KerasClassifier(build_fn = create_model,verbose = 0,batch_size = 10,epochs = 10)
                learning rate = [0.001, 0.01, 0.1]
                dropout_rate = [0.0, 0.1, 0.2]
                param grids = dict(learning rate = learning rate,dropout rate = dropout rate)
                grid = GridSearchCV(estimator = model,param grid = param_grids,cv = KFold(),verbose = 10)
                grid_result = grid.fit(X,y)
               Fitting 5 folds for each of 9 candidates, totalling 45 fits
               [CV 1/5; 1/9] START dropout_rate=0.0, learning_rate=0.001......
               C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\optimizer v2.py:355: UserWarning: The `lr` argumen
               t is deprecated, use `learning_rate` instead.
               warnings.warn(
               [CV 1/5; 1/9] END dropout_rate=0.0, learning_rate=0.001;, score=0.000 total time=
               [CV 2/5; 1/9] START dropout rate=0.0, learning rate=0.001......
               C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\optimizer v2.py:355: UserWarning: The `lr` argumen
               t is deprecated, use `learning rate` instead.
               warnings.warn(
               [CV 2/5; 1/9] END dropout rate=0.0, learning rate=0.001;, score=0.000 total time=
               [CV 3/5; 1/9] START dropout_rate=0.0, learning_rate=0.001...........
                \verb| C: Users rajesh an a conda 3 \ lib site-packages \ keras optimizer v2 optimizer v2.py: 355: \ User Warning: The `lr` argument rajesh rajesh an a conda 3 \ lib site-packages \ keras optimizer v2 optimizer v2.py: 355: \ User Warning: The `lr` argument rajesh rajes
               t is deprecated, use `learning_rate` instead.
               warnings.warn(
```

```
[CV 3/5; 1/9] END dropout_rate=0.0, learning_rate=0.001;, score=0.000 total time=
[CV 4/5; 1/9] START dropout rate=0.0, learning rate=0.001.....
 \verb|C:\Users| rajesh anaconda \verb|Site-packages| keras| optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 35
t is deprecated, use `learning rate` instead.
   warnings.warn(
 [CV 4/5; 1/9] END dropout_rate=0.0, learning_rate=0.001;, score=0.000 total time=
[CV 5/5; 1/9] START dropout_rate=0.0, learning_rate=0.001......
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer_v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 5/5; 1/9] END dropout rate=0.0, learning rate=0.001;, score=0.000 total time= 12.4s
[CV 1/5; 2/9] START dropout rate=0.0, learning rate=0.01.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\optimizer v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 1/5; 2/9] END dropout_rate=0.0, learning_rate=0.01;, score=0.000 total time=
[CV 2/5; 2/9] START dropout_rate=0.0, learning_rate=0.01.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\optimizer v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 2/5; 2/9] END dropout rate=0.0, learning rate=0.01;, score=0.000 total time=
[CV 3/5; 2/9] START dropout rate=0.0, learning rate=0.01.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 3/5; 2/9] END dropout rate=0.0, learning rate=0.01;, score=0.000 total time= 11.2s
[CV 4/5; 2/9] START dropout rate=0.0, learning rate=0.01.....
 \verb|C:\Users| rajesh anaconda \verb|Site-packages| keras| optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 35
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 4/5; 2/9] END dropout rate=0.0, learning rate=0.01;, score=0.000 total time= 7.7s
[CV 5/5; 2/9] START dropout rate=0.0, learning rate=0.01.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer_v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning rate` instead.
   warnings.warn(
[CV 5/5; 2/9] END dropout_rate=0.0, learning_rate=0.01;, score=0.000 total time=
[CV 1/5; 3/9] START dropout rate=0.0, learning rate=0.1......
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer_v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning rate` instead.
   warnings.warn(
[CV 1/5; 3/9] END dropout_rate=0.0, learning_rate=0.1;, score=0.000 total time= 10.5s
[CV 2/5; 3/9] START dropout_rate=0.0, learning_rate=0.1......
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer_v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 2/5; 3/9] END dropout_rate=0.0, learning_rate=0.1;, score=0.000 total time= 11.4s
[CV 3/5; 3/9] START dropout rate=0.0, learning rate=0.1.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\optimizer v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 3/5; 3/9] END dropout_rate=0.0, learning_rate=0.1;, score=0.000 total time= 11.0s
[CV 4/5; 3/9] START dropout rate=0.0, learning rate=0.1.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer_v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
warnings.warn(
```

[CV 4/5; 3/9] END dropout\_rate=0.0, learning\_rate=0.1;, score=0.000 total time= 9.3s [CV 5/5; 3/9] START dropout\_rate=0.0, learning\_rate=0.1.....

```
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 5/5; 3/9] END dropout_rate=0.0, learning_rate=0.1;, score=0.000 total time=
[CV 1/5; 4/9] START dropout rate=0.1, learning rate=0.001......
 \verb|C:\Users| rajesh anaconda \verb|Site-packages| keras| optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 35
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 1/5; 4/9] END dropout_rate=0.1, learning_rate=0.001;, score=0.000 total time=
[CV 2/5; 4/9] START dropout rate=0.1, learning rate=0.001......
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\optimizer v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning rate` instead.
warnings.warn(
[CV 2/5; 4/9] END dropout_rate=0.1, learning_rate=0.001;, score=0.000 total time=
[CV 3/5; 4/9] START dropout rate=0.1, learning rate=0.001......
 \verb|C:\Users| rajesh anaconda \verb|Site-packages| keras| optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 35
t is deprecated, use `learning_rate` instead.
     warnings.warn(
[CV 3/5; 4/9] END dropout rate=0.1, learning rate=0.001;, score=0.000 total time= 13.3s
[CV 4/5; 4/9] START dropout rate=0.1, learning rate=0.001......
t is deprecated, use `learning rate` instead.
warnings.warn(
[CV 4/5; 4/9] END dropout_rate=0.1, learning_rate=0.001;, score=0.000 total time= 10.0s
[CV 5/5; 4/9] START dropout_rate=0.1, learning_rate=0.001.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer_v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 5/5; 4/9] END dropout_rate=0.1, learning_rate=0.001;, score=0.000 total time= 12.6s
[CV 1/5; 5/9] START dropout_rate=0.1, learning_rate=0.01.....
 \verb|C:\Users| rajesh \ an a conda 3 \ lib \ site-packages \ keras \ optimizer\_v2.py: 355: \ User \ Warning: The `lr` argumen \ lib \ rajesh \ lib \ rajesh 
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 1/5; 5/9] END dropout_rate=0.1, learning_rate=0.01;, score=0.000 total time= 10.4s
[CV 2/5; 5/9] START dropout_rate=0.1, learning_rate=0.01.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer_v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning rate` instead.
   warnings.warn(
[CV 2/5; 5/9] END dropout_rate=0.1, learning_rate=0.01;, score=0.000 total time= 10.9s
[CV 3/5; 5/9] START dropout_rate=0.1, learning_rate=0.01.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\optimizer v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning rate` instead.
warnings.warn(
[CV 3/5; 5/9] END dropout_rate=0.1, learning_rate=0.01;, score=0.000 total time= 9.8s
[CV 4/5; 5/9] START dropout rate=0.1, learning rate=0.01.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 4/5; 5/9] END dropout rate=0.1, learning rate=0.01;, score=0.000 total time=
[CV 5/5; 5/9] START dropout_rate=0.1, learning_rate=0.01.....
 \verb|C:\Users| rajesh anaconda \verb|Site-packages| keras| optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the line of 
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 5/5; 5/9] END dropout_rate=0.1, learning_rate=0.01;, score=0.000 total time=
[CV 1/5; 6/9] START dropout_rate=0.1, learning_rate=0.1.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
```

t is deprecated, use `learning rate` instead.

warnings.warn(

```
[CV 3/5; 8/9] END dropout_rate=0.2, learning_rate=0.01;, score=0.000 total time=
                                                                                                                                                                                                                     9.7s
[CV 4/5; 8/9] START dropout rate=0.2, learning rate=0.01.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\optimizer v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 4/5; 8/9] END dropout_rate=0.2, learning_rate=0.01;, score=0.000 total time=
[CV 5/5; 8/9] START dropout rate=0.2, learning rate=0.01.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer_v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning rate` instead.
warnings.warn(
[CV 5/5; 8/9] END dropout_rate=0.2, learning_rate=0.01;, score=0.000 total time=
[CV 1/5; 9/9] START dropout rate=0.2, learning rate=0.1......
 \verb|C:\Users| rajesh anaconda \verb|Site-packages| keras \verb|Optimizer_v2.py: 355: User \verb|Warning: The `lr` argumen | line | l
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 1/5; 9/9] END dropout_rate=0.2, learning_rate=0.1;, score=0.000 total time=
[CV 2/5; 9/9] START dropout_rate=0.2, learning_rate=0.1...........
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\optimizer v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning rate` instead.
   warnings.warn(
[CV 2/5; 9/9] END dropout_rate=0.2, learning_rate=0.1;, score=0.000 total time=
                                                                                                                                                                                                                  7.7s
[CV 3/5; 9/9] START dropout_rate=0.2, learning_rate=0.1......
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\optimizer v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
    warnings.warn(
[CV 3/5; 9/9] END dropout_rate=0.2, learning_rate=0.1;, score=0.000 total time=
[CV 4/5; 9/9] START dropout rate=0.2, learning rate=0.1.....
C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\optimizer v2.py:355: UserWarning: The `lr` argumen
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 4/5; 9/9] END dropout rate=0.2, learning rate=0.1;, score=0.000 total time= 7.6s
[CV 5/5; 9/9] START dropout rate=0.2, learning rate=0.1.....
 \verb|C:\Users| rajesh anaconda \verb|Site-packages| keras| optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 355: \verb|UserWarning: The `lr` argumen | in the last optimizer_v2.py: 35
t is deprecated, use `learning_rate` instead.
warnings.warn(
[CV 5/5; 9/9] END dropout rate=0.2, learning rate=0.1;, score=0.000 total time=
                                                                                                                                                                                                                  8.0s
```

C:\Users\rajesh\anaconda3\lib\site-packages\keras\optimizer v2\optimizer v2.py:355: UserWarning: The `lr` argumen

t is deprecated, use `learning\_rate` instead.

warnings.warn(

```
In [26]: # Summarize the results
           print('Best : {}, using {}'.format(grid result.best score ,grid result.best params ))
           means = grid_result.cv_results_['mean_test_score']
           stds = grid_result.cv_results_['std_test_score']
           params = grid_result.cv_results_['params']
           for mean, stdev, param in zip(means, stds, params):
    print('{},{} with: {}'.format(mean, stdev, param))
          Best : 0.0, using {'dropout_rate': 0.0, 'learning_rate': 0.001}
          0.0,0.0 with: {'dropout_rate': 0.0, 'learning rate': 0.001}
          0.0,0.0 with: {'dropout rate': 0.0, 'learning rate': 0.01}
          0.0,0.0 with: {'dropout_rate': 0.0, 'learning_rate': 0.1} 0.0,0.0 with: {'dropout_rate': 0.1, 'learning_rate': 0.001}
         0.0,0.0 with: {'dropout_rate': 0.1, 'learning_rate': 0.001}
0.0,0.0 with: {'dropout_rate': 0.1, 'learning_rate': 0.01}
0.0,0.0 with: {'dropout_rate': 0.1, 'learning_rate': 0.1}
0.0,0.0 with: {'dropout_rate': 0.2, 'learning_rate': 0.001}
0.0,0.0 with: {'dropout_rate': 0.2, 'learning_rate': 0.01}
0.0,0.0 with: {'dropout_rate': 0.2, 'learning_rate': 0.1}
In [27]:
           def create model(neuron1, neuron2):
               model = Sequential()
               model.add(Dense(neuron1,input dim = n features,kernel initializer = 'uniform',activation = 'tanh'))
               model.add(Dense(neuron2,input dim = neuron1,kernel initializer = 'uniform',activation = 'tanh'))
               model.add(Dense(1,activation = 'sigmoid'))
               optmizer =RMSprop(0.001)#here,Learning rate is 0.03
               model.compile(loss='binary crossentropy', optimizer=optmizer, metrics=['accuracy'])
               return model
In [28]:
           # Create the model
           model = KerasClassifier(build fn = create model, verbose = 0, batch size = 10, epochs = 10)
           neuron1 = [4,8,16,20,30,40,50]
           neuron2 = [2,4,8,20,30,40,50,60]
           param grids = dict(neuron1 = neuron1.neuron2 = neuron2)
           grid = GridSearchCV(estimator = model,param grid = param grids,cv = KFold(),verbose = 10)
           grid result = grid.fit(X,y)
          Fitting 5 folds for each of 56 candidates, totalling 280 fits
          [CV 1/5; 1/56] START neuron1=4, neuron2=2.....
          [CV 1/5; 1/56] END .....neuron1=4, neuron2=2;, score=0.000 total time= 8.4s
          [CV 2/5; 1/56] START neuron1=4, neuron2=2.....
          [CV 2/5; 1/56] END .....neuron1=4, neuron2=2;, score=0.000 total time= 9.2s
          [CV 3/5; 1/56] START neuron1=4, neuron2=2......
          [CV 3/5; 1/56] END .....neuron1=4, neuron2=2;, score=0.000 total time= 9.8s
          [CV 4/5; 1/56] START neuron1=4, neuron2=2.....
          [CV 4/5; 1/56] END .....neuron1=4, neuron2=2;, score=0.000 total time= 9.3s
          [CV 5/5; 1/56] START neuron1=4, neuron2=2......
          [CV 5/5; 1/56] END .....neuron1=4, neuron2=2;, score=0.000 total time= 9.5s
          [CV 1/5; 2/56] START neuron1=4, neuron2=4.....
```

```
[CV 1/5; 2/56] END .....neuron1=4, neuron2=4;, score=0.000 total time=
[CV 2/5; 2/56] START neuron1=4, neuron2=4.....
[CV 2/5; 2/56] END .....neuron1=4, neuron2=4;, score=0.000 total time=
[CV 3/5; 2/56] START neuron1=4, neuron2=4.....
[CV 3/5; 2/56] END .....neuron1=4, neuron2=4;, score=0.000 total time= 9.0s
[CV 4/5; 2/56]
           START neuron1=4, neuron2=4.....
[CV 4/5; 2/56] END .....neuron1=4, neuron2=4;, score=0.000 total time= 9.8s
[CV 5/5; 2/56] START neuron1=4, neuron2=4.....
[CV 5/5; 2/56] END .....neuron1=4, neuron2=4;, score=0.000 total time=
[CV 1/5; 3/56] START neuron1=4, neuron2=8.....
[CV 1/5; 3/56] END .....neuron1=4, neuron2=8;, score=0.000 total time= 9.9s
           START neuron1=4, neuron2=8.....
[CV 2/5; 3/56]
[CV 2/5; 3/56] END .....neuron1=4, neuron2=8;, score=0.000 total time= 11.2s
[CV 3/5; 3/56]
           START neuron1=4, neuron2=8.....
[CV 3/5; 3/56] END .....neuron1=4, neuron2=8;, score=0.000 total time= 9.7s
[CV 4/5; 3/56]
           START neuron1=4, neuron2=8.....
[CV 4/5; 3/56]
           END .....neuron1=4, neuron2=8;, score=0.000 total time= 14.0s
[CV 5/5; 3/56] START neuron1=4, neuron2=8.....
[CV 5/5; 3/56] END .....neuron1=4, neuron2=8;, score=0.000 total time= 8.1s
[CV 1/5; 4/56]
           START neuron1=4, neuron2=20.....
[CV 1/5; 4/56] END .....neuron1=4, neuron2=20;, score=0.000 total time= 8.1s
[CV 2/5; 4/56]
           START neuron1=4, neuron2=20.....
[CV 2/5; 4/56] END .....neuron1=4, neuron2=20;, score=0.000 total time= 10.5s
           START neuron1=4, neuron2=20.....
[CV 3/5; 4/56]
[CV 3/5; 4/56] END .....neuron1=4, neuron2=20;, score=0.000 total time= 11.2s
[CV 4/5; 4/56] START neuron1=4, neuron2=20.....
[CV 4/5; 4/56] END .....neuron1=4, neuron2=20;, score=0.000 total time= 10.2s
[CV 5/5: 4/56]
           START neuron1=4, neuron2=20.....
[CV 5/5; 4/56] END .....neuron1=4, neuron2=20;, score=0.000 total time= 10.7s
           START neuron1=4, neuron2=30.....
[CV 1/5; 5/56]
[CV 1/5; 5/56]
           END .....neuron1=4, neuron2=30;, score=0.000 total time= 10.7s
[CV 2/5; 5/56] START neuron1=4, neuron2=30.....
[CV 2/5; 5/56] END .....neuron1=4, neuron2=30;, score=0.000 total time= 9.2s
           START neuron1=4, neuron2=30.....
[CV 3/5; 5/56]
[CV 3/5; 5/56] END .....neuron1=4, neuron2=30;, score=0.000 total time= 8.5s
[CV 4/5: 5/56]
           START neuron1=4. neuron2=30.....
[CV 4/5; 5/56] END .....neuron1=4, neuron2=30;, score=0.000 total time= 9.7s
[CV 5/5; 5/56] START neuron1=4, neuron2=30......
[CV 5/5; 5/56] END .....neuron1=4, neuron2=30;, score=0.000 total time= 10.2s
[CV 1/5; 6/56] START neuron1=4, neuron2=40.....
[CV 1/5; 6/56] END .....neuron1=4, neuron2=40;, score=0.000 total time= 10.3s
[CV 2/5: 6/56]
           START neuron1=4, neuron2=40.....
[CV 2/5; 6/56] END .....neuron1=4, neuron2=40;, score=0.000 total time= 9.5s
[CV 3/5; 6/56] START neuron1=4, neuron2=40.....
[CV 3/5; 6/56] END .....neuron1=4, neuron2=40;, score=0.000 total time= 9.0s
[CV 4/5; 6/56] START neuron1=4, neuron2=40.....
[CV 4/5; 6/56] END .....neuron1=4, neuron2=40;, score=0.000 total time= 9.9s
[CV 5/5; 6/56] START neuron1=4, neuron2=40......
[CV 5/5; 6/56] END .....neuron1=4, neuron2=40;, score=0.000 total time= 10.4s
           START neuron1=4, neuron2=50.....
[CV 1/5; 7/56]
[CV 1/5; 7/56] END .....neuron1=4, neuron2=50;, score=0.000 total time= 11.3s
[CV 2/5; 7/56] END .....neuron1=4, neuron2=50;, score=0.000 total time= 8.9s
[CV 3/5; 7/56]
           START neuron1=4, neuron2=50.....
[CV 3/5; 7/56] END .....neuron1=4, neuron2=50;, score=0.000 total time= 10.3s
[CV 4/5; 7/56]
           START neuron1=4, neuron2=50.....
[CV 4/5; 7/56] END .....neuron1=4, neuron2=50;, score=0.000 total time= 10.5s
[CV 5/5; 7/56]
           END .....neuron1=4, neuron2=50;, score=0.000 total time= 9.6s
[CV 1/5; 8/56] START neuron1=4, neuron2=60......
[CV 1/5; 8/56] END .....neuron1=4, neuron2=60;, score=0.000 total time= 8.8s
[CV 2/5: 8/56]
           START neuron1=4, neuron2=60.....
[CV 2/5; 8/56] END .....neuron1=4, neuron2=60;, score=0.000 total time= 11.1s
           START neuron1=4. neuron2=60.....
[CV 3/5; 8/56]
[CV 3/5; 8/56] END .....neuron1=4, neuron2=60;, score=0.000 total time= 12.5s
[CV 4/5; 8/56] START neuron1=4, neuron2=60.....
[CV 4/5; 8/56] END .....neuron1=4, neuron2=60;, score=0.000 total time= 11.8s
[CV 5/5; 8/56]
           START neuron1=4, neuron2=60.....
[CV 5/5; 8/56] END .....neuron1=4, neuron2=60;, score=0.000 total time= 11.8s
[CV 1/5; 9/56]
           START neuron1=8, neuron2=2.....
[CV 1/5; 9/56] END .....neuron1=8, neuron2=2;, score=0.000 total time= 7.2s
[CV 2/5; 9/56] END .....neuron1=8, neuron2=2;, score=0.000 total time= 12.9s
[CV 3/5; 9/56] START neuron1=8, neuron2=2.....
[CV 3/5; 9/56] END .....neuron1=8, neuron2=2;, score=0.000 total time= 9.2s
[CV 4/5; 9/56]
           START neuron1=8, neuron2=2.....
[CV 4/5; 9/56] END .....neuron1=8, neuron2=2;, score=0.000 total time= 9.4s
[CV 5/5; 9/56] START neuron1=8, neuron2=2......
[CV 5/5; 9/56] END .....neuron1=8, neuron2=2;, score=0.000 total time= 12.5s
[CV 1/5; 10/56] START neuron1=8, neuron2=4.....
[CV 1/5; 10/56] END .....neuron1=8, neuron2=4;, score=0.000 total time= 21.4s
[CV 2/5; 10/56] START neuron1=8, neuron2=4.....
[CV 2/5; 10/56] END .....neuron1=8, neuron2=4;, score=0.000 total time= 19.8s
[CV 3/5; 10/56] START neuron1=8, neuron2=4..............
[CV 3/5; 10/56] END .....neuron1=8, neuron2=4;, score=0.000 total time= 17.5s
[CV 4/5; 10/56] START neuron1=8, neuron2=4......
[CV 4/5; 10/56] END .....neuron1=8, neuron2=4;, score=0.000 total time= 10.5s
[CV 5/5; 10/56] START neuron1=8, neuron2=4.....
[CV 5/5; 10/56] END .....neuron1=8, neuron2=4;, score=0.000 total time= 8.3s
```

```
[CV 1/5; 11/56] START neuron1=8, neuron2=8.....
[CV 1/5; 11/56] END .....neuron1=8, neuron2=8;, score=0.000 total time= 8.8s
[CV 2/5; 11/56] START neuron1=8, neuron2=8.....
[CV 2/5; 11/56] END .....neuron1=8, neuron2=8;, score=0.000 total time= 9.6s
[CV 3/5; 11/56] END .....neuron1=8, neuron2=8;, score=0.000 total time= 7.9s
[CV 4/5; 11/56] START neuron1=8, neuron2=8......
[CV 4/5; 11/56] END .....neuron1=8, neuron2=8;, score=0.000 total time= 7.4s
[CV 5/5; 11/56] START neuron1=8, neuron2=8......
[CV 5/5; 11/56] END .....neuron1=8, neuron2=8;, score=0.000 total time=
[CV 1/5; 12/56] START neuron1=8, neuron2=20......
[CV 1/5; 12/56] END .....neuron1=8, neuron2=20;, score=0.000 total time= 6.6s
[CV 2/5; 12/56] START neuron1=8, neuron2=20.....
[CV 2/5; 12/56]
            END .....neuron1=8, neuron2=20;, score=0.000 total time= 6.0s
[CV 3/5; 12/56] START neuron1=8, neuron2=20.....
[CV 3/5; 12/56] END .....neuron1=8, neuron2=20;, score=0.000 total time=
[CV 4/5; 12/56]
            START neuron1=8, neuron2=20.....
[CV 4/5; 12/56] END .....neuron1=8, neuron2=20;, score=0.000 total time= 6.1s
[CV 5/5; 12/56] START neuron1=8, neuron2=20.....
[CV 5/5; 12/56] END .....neuron1=8, neuron2=20;, score=0.000 total time=
[CV 1/5; 13/56] START neuron1=8, neuron2=30.....
[CV 1/5; 13/56] END .....neuron1=8, neuron2=30;, score=0.000 total time= 5.9s
[CV 2/5; 13/56] START neuron1=8, neuron2=30.....
[CV 2/5; 13/56] END .....neuron1=8, neuron2=30;, score=0.000 total time=
[CV 3/5; 13/56] START neuron1=8, neuron2=30......
[CV 3/5; 13/56] END .....neuron1=8, neuron2=30;, score=0.000 total time= 6.4s
[CV 4/5; 13/56] START neuron1=8, neuron2=30.....
[CV 4/5; 13/56] END .....neuron1=8, neuron2=30;, score=0.000 total time= 6.2s
[CV 5/5; 13/56] START neuron1=8, neuron2=30.....
[CV 5/5; 13/56] END .....neuron1=8, neuron2=30;, score=0.000 total time=
[CV 1/5; 14/56]
            START neuron1=8, neuron2=40.....
[CV 1/5; 14/56] END .....neuron1=8, neuron2=40;, score=0.000 total time= 6.9s
[CV 2/5; 14/56] START neuron1=8, neuron2=40............
[CV 2/5; 14/56] END .....neuron1=8, neuron2=40;, score=0.000 total time= 6.2s
[CV 3/5; 14/56] START neuron1=8, neuron2=40.....
[CV 3/5; 14/56] END .....neuron1=8, neuron2=40;, score=0.000 total time= 6.1s
[CV 4/5; 14/56] START neuron1=8, neuron2=40......
[CV 4/5; 14/56] END .....neuron1=8, neuron2=40;, score=0.000 total time=
[CV 5/5; 14/56] START neuron1=8, neuron2=40.....
[CV 5/5; 14/56] END .....neuron1=8, neuron2=40;, score=0.000 total time= 6.0s
[CV 1/5; 15/56] START neuron1=8, neuron2=50.....
[CV 1/5: 15/56]
            END .....neuron1=8, neuron2=50;, score=0.000 total time= 6.2s
[CV 2/5; 15/56] START neuron1=8, neuron2=50.....
[CV 2/5; 15/56] END .....neuron1=8, neuron2=50;, score=0.000 total time=
[CV 3/5; 15/56] START neuron1=8, neuron2=50.....
[CV 3/5; 15/56] END .....neuron1=8, neuron2=50;, score=0.000 total time= 6.1s
[CV 4/5; 15/56] START neuron1=8, neuron2=50.....
[CV 4/5; 15/56] END .....neuron1=8, neuron2=50;, score=0.000 total time=
[CV 5/5; 15/56] START neuron1=8, neuron2=50.....
[CV 5/5; 15/56] END .....neuron1=8, neuron2=50;, score=0.000 total time= 6.2s
[CV 1/5; 16/56] START neuron1=8, neuron2=60......
[CV 1/5; 16/56] END .....neuron1=8, neuron2=60;, score=0.000 total time=
[CV 2/5; 16/56] START neuron1=8, neuron2=60.....
[CV 2/5; 16/56] END .....neuron1=8, neuron2=60;, score=0.000 total time= 6.3s
[CV 3/5; 16/56] START neuron1=8, neuron2=60......
[CV 3/5; 16/56] END .....neuron1=8, neuron2=60;, score=0.000 total time= 6.1s
[CV 4/5; 16/56] START neuron1=8, neuron2=60......
[CV 4/5; 16/56] END .....neuron1=8, neuron2=60;, score=0.000 total time= 6.4s
[CV 5/5; 16/56]
            START neuron1=8, neuron2=60.....
[CV 5/5; 16/56] END .....neuron1=8, neuron2=60;, score=0.000 total time= 6.1s
[CV 1/5; 17/56] START neuron1=16, neuron2=2..................
[CV 1/5; 17/56] END .....neuron1=16, neuron2=2;, score=0.000 total time=
[CV 2/5; 17/56] START neuron1=16, neuron2=2......
[CV 2/5; 17/56] END .....neuron1=16, neuron2=2;, score=0.000 total time= 5.8s
[CV 3/5; 17/56] START neuron1=16, neuron2=2.....
[CV 3/5; 17/56] END .....neuron1=16, neuron2=2;, score=0.000 total time= 5.9s
[CV 4/5; 17/56]
            START neuron1=16, neuron2=2.....
[CV 4/5; 17/56] END .....neuron1=16, neuron2=2;, score=0.000 total time= 6.1s
[CV 5/5; 17/56] START neuron1=16, neuron2=2.....
[CV 5/5; 17/56]
            END .....neuron1=16, neuron2=2;, score=0.000 total time= 6.1s
[CV 1/5; 18/56] START neuron1=16, neuron2=4.....
[CV 1/5; 18/56] END .....neuron1=16, neuron2=4;, score=0.000 total time=
[CV 2/5; 18/56] START neuron1=16, neuron2=4.....
[CV 2/5; 18/56] END .....neuron1=16, neuron2=4;, score=0.000 total time= 6.2s
[CV 3/5; 18/56] START neuron1=16, neuron2=4......
[CV 3/5; 18/56] END .....neuron1=16, neuron2=4;, score=0.000 total time= 5.9s
[CV 4/5; 18/56] START neuron1=16, neuron2=4.....
[CV 4/5; 18/56] END .....neuron1=16, neuron2=4;, score=0.000 total time= 6.0s
[CV 5/5; 18/56] START neuron1=16, neuron2=4.....
[CV 5/5; 18/56] END .....neuron1=16, neuron2=4;, score=0.000 total time=
[CV 1/5; 19/56] START neuron1=16, neuron2=8.....
[CV 1/5; 19/56] END .....neuron1=16, neuron2=8;, score=0.000 total time= 6.1s
[CV 2/5; 19/56] START neuron1=16, neuron2=8......
[CV 2/5; 19/56] END .....neuron1=16, neuron2=8;, score=0.000 total time= 6.7s
[CV 3/5; 19/56] START neuron1=16, neuron2=8......
[CV 3/5; 19/56] END .....neuron1=16, neuron2=8;, score=0.000 total time= 6.4s
[CV 4/5; 19/56] START neuron1=16, neuron2=8.....
[CV 4/5; 19/56] END .....neuron1=16, neuron2=8;, score=0.000 total time= 6.0s
[CV 5/5; 19/56] START neuron1=16, neuron2=8.....
```

```
[CV 5/5; 19/56] END .....neuron1=16, neuron2=8;, score=0.000 total time=
[CV 1/5; 20/56] START neuron1=16, neuron2=20.....
[CV 1/5; 20/56] END .....neuron1=16, neuron2=20;, score=0.000 total time=
[CV 2/5; 20/56] START neuron1=16, neuron2=20.....
[CV 2/5; 20/56] END .....neuron1=16, neuron2=20;, score=0.000 total time= 6.1s
[CV 3/5; 20/56] START neuron1=16, neuron2=20......
[CV 3/5; 20/56] END .....neuron1=16, neuron2=20;, score=0.000 total time=
[CV 4/5; 20/56] START neuron1=16, neuron2=20......
[CV 4/5; 20/56] END .....neuron1=16, neuron2=20;, score=0.000 total time=
[CV 5/5; 20/56] START neuron1=16, neuron2=20......
[CV 5/5; 20/56] END .....neuron1=16, neuron2=20;, score=0.000 total time= 6.1s
[CV 1/5; 21/56] START neuron1=16, neuron2=30.....
[CV 1/5; 21/56] END .....neuron1=16, neuron2=30;, score=0.000 total time=
[CV 2/5; 21/56]
            START neuron1=16, neuron2=30.....
[CV 2/5; 21/56] END .....neuron1=16, neuron2=30;, score=0.000 total time= 6.2s
[CV 3/5; 21/56] START neuron1=16, neuron2=30.....
[CV 3/5; 21/56] END .....neuron1=16, neuron2=30;, score=0.000 total time= 6.4s
[CV 4/5; 21/56] START neuron1=16, neuron2=30......
[CV 4/5; 21/56] END .....neuron1=16, neuron2=30;, score=0.000 total time= 6.2s
[CV 5/5; 21/56] START neuron1=16, neuron2=30.....
[CV 5/5; 21/56] END .....neuron1=16, neuron2=30;, score=0.000 total time= 6.8s
[CV 1/5; 22/56] START neuron1=16, neuron2=40......
[CV 1/5; 22/56] END .....neuron1=16, neuron2=40;, score=0.000 total time=
[CV 2/5; 22/56] START neuron1=16, neuron2=40.....
[CV 2/5; 22/56] END .....neuron1=16, neuron2=40;, score=0.000 total time= 6.2s
[CV 3/5; 22/56] START neuron1=16, neuron2=40.....
[CV 3/5; 22/56] END .....neuron1=16, neuron2=40;, score=0.000 total time=
[CV 4/5: 22/56]
            START neuron1=16, neuron2=40.....
[CV 4/5; 22/56] END .....neuron1=16, neuron2=40;, score=0.000 total time= 6.4s
[CV 5/5; 22/56] START neuron1=16, neuron2=40......
[CV 5/5; 22/56] END .....neuron1=16, neuron2=40;, score=0.000 total time= 6.2s
[CV 1/5; 23/56] START neuron1=16, neuron2=50......
[CV 1/5; 23/56] END .....neuron1=16, neuron2=50;, score=0.000 total time= 6.3s
[CV 2/5; 23/56] START neuron1=16, neuron2=50.....
[CV 2/5; 23/56] END .....neuron1=16, neuron2=50;, score=0.000 total time=
[CV 3/5: 23/56]
            START neuron1=16. neuron2=50.....
[CV 3/5; 23/56] END .....neuron1=16, neuron2=50;, score=0.000 total time= 6.6s
[CV 4/5; 23/56] START neuron1=16, neuron2=50.....
[CV 4/5; 23/56] END .....neuron1=16, neuron2=50;, score=0.000 total time= 6.4s
[CV 5/5; 23/56] START neuron1=16, neuron2=50.....
[CV 5/5; 23/56] END .....neuron1=16, neuron2=50;, score=0.000 total time=
[CV 1/5; 24/56]
            START neuron1=16, neuron2=60.....
[CV 1/5; 24/56] END .....neuron1=16, neuron2=60;, score=0.000 total time= 6.2s
[CV 2/5; 24/56] START neuron1=16, neuron2=60.....
[CV 2/5; 24/56] END .....neuron1=16, neuron2=60;, score=0.000 total time= 6.3s
[CV 3/5; 24/56] START neuron1=16, neuron2=60.....
[CV 3/5; 24/56] END .....neuron1=16, neuron2=60;, score=0.000 total time= 6.2s
[CV 4/5; 24/56] START neuron1=16, neuron2=60.....
[CV 4/5; 24/56] END .....neuron1=16, neuron2=60;, score=0.000 total time= 6.4s
[CV 5/5; 24/56] START neuron1=16, neuron2=60.....
[CV 5/5; 24/56] END .....neuron1=16, neuron2=60;, score=0.000 total time= 6.3s
[CV 1/5; 25/56] END .....neuron1=20, neuron2=2;, score=0.000 total time= 6.1s
[CV 2/5; 25/56] START neuron1=20, neuron2=2.....
[CV 2/5; 25/56] END .....neuron1=20, neuron2=2;, score=0.000 total time=
[CV 3/5; 25/56]
            START neuron1=20, neuron2=2.....
[CV 3/5; 25/56] END .....neuron1=20, neuron2=2;, score=0.000 total time= 6.0s
[CV 4/5; 25/56] START neuron1=20, neuron2=2......
            END .....neuron1=20, neuron2=2;, score=0.000 total time= 6.0s
[CV 4/5; 25/56]
[CV 5/5; 25/56] START neuron1=20, neuron2=2......
[CV 5/5; 25/56] END .....neuron1=20, neuron2=2;, score=0.000 total time= 6.0s
[CV 1/5; 26/56] START neuron1=20, neuron2=4......
[CV 1/5; 26/56] END .....neuron1=20, neuron2=4;, score=0.000 total time= 5.9s
[CV 2/5; 26/56] START neuron1=20, neuron2=4.....
[CV 2/5; 26/56] END .....neuron1=20, neuron2=4;, score=0.000 total time=
[CV 3/5; 26/56] START neuron1=20, neuron2=4.....
[CV 3/5; 26/56] END .....neuron1=20, neuron2=4;, score=0.000 total time= 6.1s
[CV 4/5; 26/56] START neuron1=20, neuron2=4.....
[CV 4/5; 26/56] END .....neuron1=20, neuron2=4;, score=0.000 total time= 6.0s
[CV 5/5; 26/56]
            START neuron1=20, neuron2=4.....
[CV 5/5; 26/56] END .....neuron1=20, neuron2=4;, score=0.000 total time= 6.0s
[CV 1/5; 27/56] START neuron1=20, neuron2=8......
[CV 1/5; 27/56] END .....neuron1=20, neuron2=8;, score=0.000 total time= 6.2s
[CV 2/5; 27/56] START neuron1=20, neuron2=8.....
[CV 2/5; 27/56] END .....neuron1=20, neuron2=8;, score=0.000 total time= 6.1s
[CV 3/5; 27/56]
            START neuron1=20, neuron2=8.....
[CV 3/5; 27/56] END .....neuron1=20, neuron2=8;, score=0.000 total time=
            START neuron1=20, neuron2=8.....
[CV 4/5; 27/56]
[CV 4/5; 27/56] END .....neuron1=20, neuron2=8;, score=0.000 total time= 6.2s
[CV 5/5; 27/56] START neuron1=20, neuron2=8.....
[CV 5/5; 27/56] END .....neuron1=20, neuron2=8;, score=0.000 total time= 6.2s
[CV 1/5; 28/56] START neuron1=20, neuron2=20.....
[CV 1/5; 28/56] END .....neuron1=20, neuron2=20;, score=0.000 total time= 6.3s
[CV 2/5; 28/56]
            START neuron1=20, neuron2=20.....
[CV 2/5; 28/56] END .....neuron1=20, neuron2=20;, score=0.000 total time= 6.1s
[CV 3/5; 28/56] START neuron1=20, neuron2=20.......................
[CV 3/5; 28/56] END .....neuron1=20, neuron2=20;, score=0.000 total time= 6.2s
[CV 4/5; 28/56] START neuron1=20, neuron2=20.....
[CV 4/5; 28/56] END .....neuron1=20, neuron2=20;, score=0.000 total time= 6.2s
```

```
[CV 5/5; 28/56] START neuron1=20, neuron2=20......
[CV 5/5; 28/56] END .....neuron1=20, neuron2=20;, score=0.000 total time= 6.4s
[CV 1/5; 29/56] START neuron1=20, neuron2=30.....
[CV 1/5; 29/56] END .....neuron1=20, neuron2=30;, score=0.000 total time= 6.2s
[CV 2/5; 29/56] END .....neuron1=20, neuron2=30;, score=0.000 total time=
[CV 3/5; 29/56] START neuron1=20, neuron2=30.....
[CV 3/5; 29/56] END .....neuron1=20, neuron2=30;, score=0.000 total time= 6.2s
[CV 4/5; 29/56] START neuron1=20, neuron2=30.....
[CV 4/5; 29/56] END .....neuron1=20, neuron2=30;, score=0.000 total time=
[CV 5/5; 29/56] START neuron1=20, neuron2=30......
[CV 5/5; 29/56] END .....neuron1=20, neuron2=30;, score=0.000 total time= 6.2s
[CV 1/5; 30/56] START neuron1=20, neuron2=40.....
[CV 1/5; 30/56] END .....neuron1=20, neuron2=40;, score=0.000 total time= 6.2s
[CV 2/5; 30/56] START neuron1=20, neuron2=40.....
[CV 2/5; 30/56] END .....neuron1=20, neuron2=40;, score=0.000 total time=
[CV 3/5; 30/56] END .....neuron1=20, neuron2=40;, score=0.000 total time= 6.5s
[CV 4/5; 30/56] START neuron1=20, neuron2=40.....
[CV 4/5; 30/56] END .....neuron1=20, neuron2=40;, score=0.000 total time=
[CV 5/5; 30/56] START neuron1=20, neuron2=40.....
[CV 5/5; 30/56] END .....neuron1=20, neuron2=40;, score=0.000 total time= 6.2s
[CV 1/5; 31/56] START neuron1=20, neuron2=50.....
[CV 1/5; 31/56] END .....neuron1=20, neuron2=50;, score=0.000 total time=
[CV 2/5; 31/56] START neuron1=20, neuron2=50.....
[CV 2/5; 31/56] END .....neuron1=20, neuron2=50;, score=0.000 total time= 6.2s
[CV 3/5; 31/56] START neuron1=20, neuron2=50......
[CV 3/5; 31/56] END .....neuron1=20, neuron2=50;, score=0.000 total time= 6.3s
[CV 4/5; 31/56] START neuron1=20, neuron2=50.....
[CV 4/5; 31/56] END .....neuron1=20, neuron2=50;, score=0.000 total time= 7.0s
[CV 5/5; 31/56] START neuron1=20, neuron2=50.....
[CV 5/5; 31/56] END .....neuron1=20, neuron2=50;, score=0.000 total time= 6.6s
[CV 1/5; 32/56] START neuron1=20, neuron2=60......
[CV 1/5; 32/56] END .....neuron1=20, neuron2=60;, score=0.000 total time= 6.4s
[CV 2/5; 32/56] START neuron1=20, neuron2=60.....
[CV 2/5; 32/56] END .....neuron1=20, neuron2=60;, score=0.000 total time= 9.0s
[CV 3/5; 32/56] START neuron1=20, neuron2=60.....
[CV 3/5; 32/56] END .....neuron1=20, neuron2=60;, score=0.000 total time= 7.8s
[CV 4/5; 32/56] START neuron1=20, neuron2=60.....
[CV 4/5; 32/56] END .....neuron1=20, neuron2=60;, score=0.000 total time= 6.3s
[CV 5/5; 32/56] START neuron1=20, neuron2=60.....
[CV 5/5; 32/56] END .....neuron1=20, neuron2=60;, score=0.000 total time= 6.4s
[CV 1/5; 33/56] START neuron1=30, neuron2=2.....
[CV 1/5; 33/56] END .....neuron1=30, neuron2=2;, score=0.000 total time=
[CV 2/5; 33/56] START neuron1=30, neuron2=2.....
[CV 2/5; 33/56] END .....neuron1=30, neuron2=2;, score=0.000 total time= 5.9s
[CV 3/5; 33/56] START neuron1=30, neuron2=2.....
[CV 3/5; 33/56] END .....neuron1=30, neuron2=2;, score=0.000 total time=
[CV 4/5; 33/56] START neuron1=30, neuron2=2.....
[CV 4/5; 33/56] END .....neuron1=30, neuron2=2;, score=0.000 total time= 6.1s
[CV 5/5; 33/56] START neuron1=30, neuron2=2.....
[CV 5/5; 33/56] END .....neuron1=30, neuron2=2;, score=0.000 total time=
[CV 1/5; 34/56] START neuron1=30, neuron2=4.....
[CV 1/5; 34/56] END .....neuron1=30, neuron2=4;, score=0.000 total time= 6.1s
[CV 2/5; 34/56] START neuron1=30, neuron2=4.....
[CV 2/5; 34/56] END .....neuron1=30, neuron2=4;, score=0.000 total time= 6.4s
[CV 3/5; 34/56] START neuron1=30, neuron2=4.....
[CV 3/5; 34/56] END .....neuron1=30, neuron2=4;, score=0.000 total time= 6.2s
[CV 4/5; 34/56] START neuron1=30, neuron2=4......
[CV 4/5; 34/56] END .....neuron1=30, neuron2=4;, score=0.000 total time= 6.1s
[CV 5/5; 34/56] START neuron1=30, neuron2=4......
[CV 5/5; 34/56] END .....neuron1=30, neuron2=4;, score=0.000 total time= 6.1s
[CV 1/5; 35/56] START neuron1=30, neuron2=8......
[CV 1/5; 35/56] END .....neuron1=30, neuron2=8;, score=0.000 total time= 6.3s
[CV 2/5; 35/56] START neuron1=30, neuron2=8.....
[CV 2/5; 35/56] END .....neuron1=30, neuron2=8;, score=0.000 total time= 6.2s
[CV 3/5; 35/56] END .....neuron1=30, neuron2=8;, score=0.000 total time= 6.2s
[CV 4/5; 35/56] START neuron1=30, neuron2=8.....
[CV 4/5; 35/56] END .....neuron1=30, neuron2=8;, score=0.000 total time= 6.1s
[CV 5/5; 35/56] START neuron1=30, neuron2=8.....
[CV 5/5; 35/56] END .....neuron1=30, neuron2=8;, score=0.000 total time=
[CV 1/5; 36/56] START neuron1=30, neuron2=20.....
[CV 1/5; 36/56] END .....neuron1=30, neuron2=20;, score=0.000 total time= 6.1s
[CV 2/5; 36/56] END .....neuron1=30, neuron2=20;, score=0.000 total time= 6.4s
[CV 3/5; 36/56] START neuron1=30, neuron2=20.....
[CV 3/5; 36/56] END .....neuron1=30, neuron2=20;, score=0.000 total time= 6.2s
[CV 4/5; 36/56] START neuron1=30, neuron2=20.....
[CV 4/5; 36/56] END .....neuron1=30, neuron2=20;, score=0.000 total time=
[CV 5/5; 36/56] START neuron1=30, neuron2=20.....
[CV 5/5; 36/56] END .....neuron1=30, neuron2=20;, score=0.000 total time= 6.2s
[CV 1/5; 37/56] START neuron1=30, neuron2=30......
[CV 1/5; 37/56] END .....neuron1=30, neuron2=30;, score=0.000 total time= 6.3s
[CV 2/5; 37/56] START neuron1=30, neuron2=30.....
[CV 2/5; 37/56] END .....neuron1=30, neuron2=30;, score=0.000 total time= 6.3s
[CV 3/5; 37/56] START neuron1=30, neuron2=30.....
[CV 3/5; 37/56] END .....neuron1=30, neuron2=30;, score=0.000 total time= 6.3s
[CV 4/5; 37/56] START neuron1=30, neuron2=30.....
```

```
[CV 4/5; 37/56] END .....neuron1=30, neuron2=30;, score=0.000 total time=
[CV 5/5; 37/56] START neuron1=30, neuron2=30.....
[CV 5/5; 37/56] END .....neuron1=30, neuron2=30;, score=0.000 total time=
[CV 1/5; 38/56] START neuron1=30, neuron2=40.....
[CV 1/5; 38/56] END .....neuron1=30, neuron2=40;, score=0.000 total time=
[CV 2/5; 38/56] START neuron1=30, neuron2=40.....
[CV 2/5; 38/56] END .....neuron1=30, neuron2=40;, score=0.000 total time=
[CV 3/5; 38/56] START neuron1=30, neuron2=40......
[CV 3/5; 38/56] END .....neuron1=30, neuron2=40;, score=0.000 total time=
[CV 4/5; 38/56] START neuron1=30, neuron2=40.....
[CV 4/5; 38/56] END .....neuron1=30, neuron2=40;, score=0.000 total time= 6.7s
[CV 5/5; 38/56] START neuron1=30, neuron2=40......
[CV 5/5; 38/56] END .....neuron1=30, neuron2=40;, score=0.000 total time=
[CV 1/5: 39/56]
            START neuron1=30, neuron2=50.....
[CV 1/5; 39/56] END .....neuron1=30, neuron2=50;, score=0.000 total time= 7.9s
[CV 2/5; 39/56] START neuron1=30, neuron2=50.....
[CV 2/5; 39/56] END .....neuron1=30, neuron2=50;, score=0.000 total time= 6.8s
[CV 3/5; 39/56] START neuron1=30, neuron2=50.....
[CV 3/5; 39/56] END .....neuron1=30, neuron2=50;, score=0.000 total time= 6.4s
[CV 4/5; 39/56] START neuron1=30, neuron2=50......
[CV 4/5; 39/56] END .....neuron1=30, neuron2=50;, score=0.000 total time= 6.5s
[CV 5/5; 39/56] START neuron1=30, neuron2=50......
[CV 5/5; 39/56] END .....neuron1=30, neuron2=50;, score=0.000 total time=
[CV 1/5; 40/56] START neuron1=30, neuron2=60.....
[CV 1/5; 40/56] END .....neuron1=30, neuron2=60;, score=0.000 total time= 6.4s
[CV 2/5; 40/56] START neuron1=30, neuron2=60.....
[CV 2/5; 40/56] END .....neuron1=30, neuron2=60;, score=0.000 total time= 7.4s
[CV 3/5: 40/56]
            START neuron1=30, neuron2=60.....
[CV 3/5; 40/56] END .....neuron1=30, neuron2=60;, score=0.000 total time= 6.9s
[CV 4/5; 40/56] START neuron1=30, neuron2=60......
[CV 4/5; 40/56] END .....neuron1=30, neuron2=60;, score=0.000 total time= 6.5s
[CV 5/5; 40/56] START neuron1=30, neuron2=60.....
[CV 5/5; 40/56] END .....neuron1=30, neuron2=60;, score=0.000 total time= 6.5s
[CV 1/5; 41/56] START neuron1=40, neuron2=2.....
[CV 1/5; 41/56] END .....neuron1=40, neuron2=2;, score=0.000 total time=
[CV 2/5; 41/56]
            START neuron1=40, neuron2=2.....
[CV 2/5; 41/56] END .....neuron1=40, neuron2=2;, score=0.000 total time= 6.3s
[CV 3/5; 41/56] START neuron1=40, neuron2=2.....
[CV 3/5; 41/56] END .....neuron1=40, neuron2=2;, score=0.000 total time= 6.1s
[CV 4/5; 41/56] START neuron1=40, neuron2=2.....
[CV 4/5; 41/56] END .....neuron1=40, neuron2=2;, score=0.000 total time=
[CV 5/5: 41/56]
            START neuron1=40, neuron2=2.....
[CV 5/5; 41/56] END .....neuron1=40, neuron2=2;, score=0.000 total time= 6.1s
[CV 1/5; 42/56] START neuron1=40, neuron2=4.....
[CV 1/5; 42/56] END .....neuron1=40, neuron2=4;, score=0.000 total time= 6.2s
[CV 2/5; 42/56] START neuron1=40, neuron2=4.....
[CV 2/5; 42/56] END .....neuron1=40, neuron2=4;, score=0.000 total time= 6.1s
[CV 3/5; 42/56] START neuron1=40, neuron2=4.....
[CV 3/5; 42/56] END .....neuron1=40, neuron2=4;, score=0.000 total time= 6.0s
[CV 4/5; 42/56]
            START neuron1=40, neuron2=4.....
[CV 4/5; 42/56] END .....neuron1=40, neuron2=4;, score=0.000 total time= 6.0s
[CV 5/5; 42/56] START neuron1=40, neuron2=4..............
[CV 5/5; 42/56] END .....neuron1=40, neuron2=4;, score=0.000 total time= 6.2s
[CV 1/5; 43/56] START neuron1=40, neuron2=8.....
[CV 1/5; 43/56] END .....neuron1=40, neuron2=8;, score=0.000 total time=
[CV 2/5; 43/56]
            START neuron1=40, neuron2=8.....
[CV 2/5; 43/56] END .....neuron1=40, neuron2=8;, score=0.000 total time= 6.1s
[CV 3/5; 43/56] START neuron1=40, neuron2=8.....
            END .....neuron1=40, neuron2=8;, score=0.000 total time= 6.2s
[CV 3/5; 43/56]
[CV 4/5; 43/56] START neuron1=40, neuron2=8.....
[CV 4/5; 43/56] END .....neuron1=40, neuron2=8;, score=0.000 total time= 6.1s
[CV 5/5: 43/56]
            START neuron1=40, neuron2=8.....
[CV 5/5; 43/56] END .....neuron1=40, neuron2=8;, score=0.000 total time=
[CV 1/5; 44/56]
            START neuron1=40. neuron2=20......
[CV 1/5; 44/56] END .....neuron1=40, neuron2=20;, score=0.000 total time=
[CV 2/5; 44/56] START neuron1=40, neuron2=20......
[CV 2/5; 44/56] END .....neuron1=40, neuron2=20;, score=0.000 total time= 6.7s
[CV 3/5; 44/56] START neuron1=40, neuron2=20.....
[CV 3/5; 44/56] END .....neuron1=40, neuron2=20;, score=0.000 total time= 6.2s
[CV 4/5; 44/56]
            START neuron1=40, neuron2=20.....
[CV 4/5; 44/56] END .....neuron1=40, neuron2=20;, score=0.000 total time= 6.2s
[CV 5/5; 44/56] START neuron1=40, neuron2=20.....
[CV 5/5; 44/56] END .....neuron1=40, neuron2=20;, score=0.000 total time= 6.2s
[CV 1/5; 45/56] START neuron1=40, neuron2=30.....
[CV 1/5; 45/56] END .....neuron1=40, neuron2=30;, score=0.000 total time= 6.4s
[CV 2/5; 45/56] START neuron1=40, neuron2=30.....
[CV 2/5; 45/56] END .....neuron1=40, neuron2=30;, score=0.000 total time=
[CV 3/5; 45/56] START neuron1=40, neuron2=30......
[CV 3/5; 45/56] END .....neuron1=40, neuron2=30;, score=0.000 total time= 6.3s
[CV 4/5; 45/56] START neuron1=40, neuron2=30.....
[CV 4/5; 45/56] END .....neuron1=40, neuron2=30;, score=0.000 total time= 6.2s
[CV 5/5; 45/56] START neuron1=40, neuron2=30......
[CV 5/5; 45/56] END .....neuron1=40, neuron2=30;, score=0.000 total time= 6.2s
[CV 1/5; 46/56]
            START neuron1=40, neuron2=40.....
[CV 1/5; 46/56] END .....neuron1=40, neuron2=40;, score=0.000 total time= 6.4s
[CV 2/5; 46/56] START neuron1=40, neuron2=40.................
[CV 2/5; 46/56] END .....neuron1=40, neuron2=40;, score=0.000 total time= 6.6s
[CV 3/5; 46/56] START neuron1=40, neuron2=40.....
[CV 3/5; 46/56] END .....neuron1=40, neuron2=40;, score=0.000 total time= 6.5s
```

```
[CV 4/5; 46/56] START neuron1=40, neuron2=40.....
[CV 4/5; 46/56] END .....neuron1=40, neuron2=40;, score=0.000 total time= 6.5s
[CV 5/5; 46/56] START neuron1=40, neuron2=40.....
[CV 5/5; 46/56] END .....neuron1=40, neuron2=40;, score=0.000 total time= 6.4s
[CV 1/5; 47/56] START neuron1=40, neuron2=50.....
[CV 1/5; 47/56] END .....neuron1=40, neuron2=50;, score=0.000 total time=
[CV 2/5; 47/56] START neuron1=40, neuron2=50......
[CV 2/5; 47/56] END .....neuron1=40, neuron2=50;, score=0.000 total time= 6.5s
[CV 3/5; 47/56] START neuron1=40, neuron2=50.....
[CV 3/5; 47/56] END .....neuron1=40, neuron2=50;, score=0.000 total time=
[CV 4/5; 47/56] START neuron1=40, neuron2=50......
[CV 4/5; 47/56] END .....neuron1=40, neuron2=50;, score=0.000 total time= 6.4s
[CV 5/5; 47/56] START neuron1=40, neuron2=50.....
[CV 5/5; 47/56]
            END .....neuron1=40, neuron2=50;, score=0.000 total time= 6.5s
[CV 1/5; 48/56] START neuron1=40, neuron2=60.....
[CV 1/5; 48/56] END .....neuron1=40, neuron2=60;, score=0.000 total time=
[CV 2/5; 48/56] START neuron1=40, neuron2=60.....
[CV 2/5; 48/56] END .....neuron1=40, neuron2=60;, score=0.000 total time= 6.5s
[CV 3/5; 48/56] START neuron1=40, neuron2=60.....
[CV 3/5; 48/56] END .....neuron1=40, neuron2=60;, score=0.000 total time= 6.5s
[CV 4/5; 48/56] START neuron1=40, neuron2=60.....
[CV 4/5; 48/56] END .....neuron1=40, neuron2=60;, score=0.000 total time= 6.8s
[CV 5/5; 48/56] START neuron1=40, neuron2=60.....
[CV 5/5; 48/56] END .....neuron1=40, neuron2=60;, score=0.000 total time=
[CV 1/5; 49/56] START neuron1=50, neuron2=2......
[CV 1/5; 49/56] END .....neuron1=50, neuron2=2;, score=0.000 total time= 6.3s
[CV 2/5; 49/56] START neuron1=50, neuron2=2......
[CV 2/5; 49/56] END .....neuron1=50, neuron2=2;, score=0.000 total time= 6.1s
[CV 3/5; 49/56] START neuron1=50, neuron2=2.....
[CV 3/5; 49/56] END .....neuron1=50, neuron2=2;, score=0.000 total time=
[CV 4/5; 49/56] START neuron1=50, neuron2=2.....
[CV 4/5; 49/56] END .....neuron1=50, neuron2=2;, score=0.000 total time= 6.2s
[CV 5/5; 49/56] START neuron1=50, neuron2=2.....
[CV 5/5; 49/56] END .....neuron1=50, neuron2=2;, score=0.000 total time= 6.0s
[CV 1/5; 50/56] START neuron1=50, neuron2=4.....
[CV 1/5; 50/56] END .....neuron1=50, neuron2=4;, score=0.000 total time= 6.3s
[CV 2/5; 50/56] START neuron1=50, neuron2=4.....
[CV 2/5; 50/56] END .....neuron1=50, neuron2=4;, score=0.000 total time= 6.3s
[CV 3/5; 50/56] START neuron1=50, neuron2=4......
[CV 3/5; 50/56] END .....neuron1=50, neuron2=4;, score=0.000 total time= 6.2s
[CV 4/5; 50/56] START neuron1=50, neuron2=4.....
[CV 4/5; 50/56] END .....neuron1=50, neuron2=4;, score=0.000 total time= 6.5s
[CV 5/5; 50/56] START neuron1=50, neuron2=4.....
[CV 5/5; 50/56] END .....neuron1=50, neuron2=4;, score=0.000 total time= 6.0s
[CV 1/5; 51/56] START neuron1=50, neuron2=8.....
[CV 1/5; 51/56] END .....neuron1=50, neuron2=8;, score=0.000 total time= 6.3s
[CV 2/5; 51/56] START neuron1=50, neuron2=8.......
[CV 2/5; 51/56] END .....neuron1=50, neuron2=8;, score=0.000 total time=
[CV 3/5; 51/56] START neuron1=50, neuron2=8.....
[CV 3/5; 51/56] END .....neuron1=50, neuron2=8;, score=0.000 total time= 6.3s
[CV 4/5; 51/56] START neuron1=50, neuron2=8......
[CV 4/5; 51/56] END .....neuron1=50, neuron2=8;, score=0.000 total time=
[CV 5/5; 51/56] START neuron1=50, neuron2=8.....
[CV 5/5; 51/56] END .....neuron1=50, neuron2=8;, score=0.000 total time= 6.2s
[CV 1/5; 52/56] START neuron1=50, neuron2=20.....
[CV 1/5; 52/56] END .....neuron1=50, neuron2=20;, score=0.000 total time= 6.3s
[CV 2/5; 52/56] START neuron1=50, neuron2=20......
[CV 2/5; 52/56] END .....neuron1=50, neuron2=20;, score=0.000 total time= 6.5s
[CV 3/5; 52/56]
            START neuron1=50, neuron2=20......
[CV 3/5; 52/56] END .....neuron1=50, neuron2=20;, score=0.000 total time= 6.6s
[CV 4/5; 52/56] START neuron1=50, neuron2=20.....
[CV 4/5; 52/56] END .....neuron1=50, neuron2=20;, score=0.000 total time=
[CV 5/5; 52/56] START neuron1=50, neuron2=20.....
[CV 5/5; 52/56] END .....neuron1=50, neuron2=20;, score=0.000 total time= 6.4s
[CV 1/5; 53/56] START neuron1=50, neuron2=30.....
[CV 1/5; 53/56] END .....neuron1=50, neuron2=30;, score=0.000 total time= 6.5s
[CV 2/5; 53/56] START neuron1=50, neuron2=30.....
[CV 2/5; 53/56] END .....neuron1=50, neuron2=30;, score=0.000 total time= 6.4s
[CV 3/5; 53/56] START neuron1=50, neuron2=30.....
[CV 3/5; 53/56] END .....neuron1=50, neuron2=30;, score=0.000 total time= 6.4s
[CV 4/5; 53/56] START neuron1=50, neuron2=30.....
[CV 4/5; 53/56] END .....neuron1=50, neuron2=30;, score=0.000 total time=
[CV 5/5; 53/56] START neuron1=50, neuron2=30.....
[CV 5/5; 53/56] END .....neuron1=50, neuron2=30;, score=0.000 total time= 6.6s
[CV 1/5; 54/56] END .....neuron1=50, neuron2=40;, score=0.000 total time= 6.7s
[CV 2/5; 54/56] START neuron1=50, neuron2=40.....
[CV 2/5; 54/56] END .....neuron1=50, neuron2=40;, score=0.000 total time= 6.6s
[CV 3/5; 54/56] START neuron1=50, neuron2=40.....
[CV 3/5; 54/56] END .....neuron1=50, neuron2=40;, score=0.000 total time=
[CV 4/5; 54/56] START neuron1=50, neuron2=40.....
[CV 4/5; 54/56] END .....neuron1=50, neuron2=40;, score=0.000 total time= 6.5s
[CV 5/5; 54/56] START neuron1=50, neuron2=40.....
[CV 5/5; 54/56] END .....neuron1=50, neuron2=40;, score=0.000 total time= 6.5s
[CV 1/5; 55/56] START neuron1=50, neuron2=50......
[CV 1/5; 55/56] END .....neuron1=50, neuron2=50;, score=0.000 total time= 6.8s
[CV 2/5; 55/56] START neuron1=50, neuron2=50.....
[CV 2/5; 55/56] END .....neuron1=50, neuron2=50;, score=0.000 total time= 6.7s
[CV 3/5; 55/56] START neuron1=50, neuron2=50.....
```

```
[CV 3/5; 55/56] END .....neuron1=50, neuron2=50;, score=0.000 total time=
          [CV 4/5; 55/56] START neuron1=50, neuron2=50......
          [CV 4/5; 55/56] END .....neuron1=50, neuron2=50;, score=0.000 total time=
          [CV 5/5; 55/56] START neuron1=50, neuron2=50.....
          [CV 5/5; 55/56] END .....neuron1=50, neuron2=50;, score=0.000 total time= 6.7s
          [CV 1/5; 56/56] START neuron1=50, neuron2=60......
          [CV 1/5; 56/56] END .....neuron1=50, neuron2=60;, score=0.000 total time=
          [CV 2/5; 56/56] START neuron1=50, neuron2=60.....
          [CV 2/5; 56/56] END .....neuron1=50, neuron2=60;, score=0.000 total time=
          [CV 3/5; 56/56] START neuron1=50, neuron2=60.....
          [CV 3/5; 56/56] END .....neuron1=50, neuron2=60;, score=0.000 total time= 7.2s
          [CV 4/5; 56/56] START neuron1=50, neuron2=60.....
          [CV 4/5; 56/56] END .....neuron1=50, neuron2=60;, score=0.000 total time=
          [CV 5/5; 56/56] START neuron1=50, neuron2=60.....
          [CV 5/5; 56/56] END .....neuron1=50, neuron2=60;, score=0.000 total time= 6.7s
In [29]:
           #Summarize the results
           print('Best : {}, using {}'.format(grid_result.best_score_,grid_result.best_params_))
           means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
           params = grid_result.cv_results_['params']
           for mean, stdev, param in zip(means, stds, params):
    print('{},{} with: {}'.format(mean, stdev, param))
          Best : 0.0, using {'neuron1': 4, 'neuron2': 2}
          0.0,0.0 with: {'neuron1': 4, 'neuron2': 2}
0.0,0.0 with: {'neuron1': 4, 'neuron2': 4}
                                          'neuron2': 2}
          0.0,0.0 with: {'neuron1': 4, 'neuron2': 8}
          0.0,0.0 with: {'neuron1': 4, 'neuron2': 20} 0.0,0.0 with: {'neuron1': 4, 'neuron2': 30}
          0.0,0.0 with: {'neuron1': 4, 'neuron2': 40}
          0.0,0.0 with: {'neuron1': 4, 'neuron2': 50}
          0.0,0.0 with: {'neuron1': 4, 'neuron2': 60}
          0.0,0.0 with: {'neuron1': 8, 'neuron2': 2}
          0.0,0.0 with: {'neuron1': 8, 'neuron2': 4}
          0.0,0.0 with: {'neuron1': 8, 'neuron2': 8}
          0.0,0.0 with: {'neuron1': 8, 'neuron2': 20} 0.0,0.0 with: {'neuron1': 8, 'neuron2': 30}
          0.0,0.0 with: {'neuron1': 8, 'neuron2': 40}
          0.0,0.0 with: {'neuron1': 8, 'neuron2': 50} 0.0,0.0 with: {'neuron1': 8, 'neuron2': 60}
          0.0,0.0 with: {'neuron1': 16, 'neuron2': 2}
          0.0,0.0 with: {'neuron1': 16, 'neuron2': 4}
          0.0,0.0 with: {'neuron1': 16, 'neuron2': 8}
          0.0,0.0 with: {'neuron1': 16, 'neuron2': 20}
          0.0,0.0 with: {'neuron1': 16, 'neuron2': 30}
          0.0,0.0 with: {'neuron1': 16, 'neuron2': 40}
          0.0,0.0 with: {'neuron1': 16, 'neuron2': 50}
0.0,0.0 with: {'neuron1': 16, 'neuron2': 60}
          0.0,0.0 with: {'neuron1': 20, 'neuron2': 2}
          0.0,0.0 with: {'neuron1': 20, 'neuron2': 4}
0.0,0.0 with: {'neuron1': 20, 'neuron2': 8}
          0.0,0.0 with: {'neuron1': 20, 'neuron2': 20}
          0.0,0.0 with: {'neuron1': 20, 'neuron2': 30}
0.0,0.0 with: {'neuron1': 20, 'neuron2': 40}
          0.0,0.0 with: {'neuron1': 20, 'neuron2': 50}
          0.0,0.0 with: {'neuron1': 20,
                                           'neuron2': 60}
          0.0,0.0 with: {'neuron1': 30, 'neuron2': 2}
          0.0,0.0 with: {'neuron1': 30, 'neuron2': 4}
          0.0,0.0 with: {'neuron1': 30, 'neuron2': 8}
          0.0,0.0 with: {'neuron1': 30, 'neuron2': 20}
          0.0,0.0 with: {'neuron1': 30, 'neuron2': 30}
          0.0,0.0 with: {'neuron1': 30, 'neuron2': 40}
          0.0,0.0 with: {'neuron1': 30, 'neuron2': 50}
          0.0,0.0 with: {'neuron1': 30, 'neuron2': 60} 0.0,0.0 with: {'neuron1': 40, 'neuron2': 2}
          0.0,0.0 with: {'neuron1': 40, 'neuron2': 4}
          0.0,0.0 with: {'neuron1': 40, 'neuron2': 8}
          0.0,0.0 with: {'neuron1': 40, 'neuron2': 20}
          0.0,0.0 with: {'neuron1': 40, 'neuron2': 30}
          0.0,0.0 with: {'neuron1': 40,
                                           'neuron2': 40}
          0.0,0.0 with: {'neuron1': 40, 'neuron2': 50}
          0.0,0.0 with: {'neuron1': 40, 'neuron2': 60}
          0.0,0.0 with: {'neuron1': 50, 'neuron2': 2}
          0.0,0.0 with: {'neuron1': 50, 'neuron2': 4}
          0.0,0.0 with: {'neuron1': 50, 'neuron2': 8}
          0.0,0.0 with: {'neuron1': 50, 'neuron2': 20}
          0.0,0.0 with: {'neuron1': 50, 'neuron2': 30}
          0.0,0.0 with: {'neuron1': 50, 'neuron2': 40} 0.0,0.0 with: {'neuron1': 50, 'neuron2': 50} 0.0,0.0 with: {'neuron1': 50, 'neuron2': 60}
```

```
In [31]:
      X train,X test,y train,y test =train test split(X,y,test size =0.3,random state =42)
In [32]:
      X_train.shape,X_test.shape,y_train.shape,y_test.shape
Out[32]: ((10527, 10), (4512, 10), (10527,), (4512,))
In [36]:
      optmizer =RMSprop(0.001)
      model_new=keras.Sequential([
         keras.layers.Dense(4,input_dim =(n_features),activation='relu'),
         keras.layers.Dense(2,activation ='relu')
      1)
      model new.compile(optimizer =optmizer,loss= 'mean squared error',metrics=['accuracy'])
In [37]:
      seed value =42;
      import random
      tensorflow random set seed(seed value)
      model_new.fit(X_train, y_train, epochs=10, batch_size=10, verbose = 1)
      Epoch 1/10
      1053/1053 [=
                         ========] - 1s 698us/step - loss: 0.5495 - accuracy: 0.0000e+00
      Epoch 2/10
      Epoch 3/10
      Epoch 4/10
      1053/1053 [=
                          =======] - 1s 848us/step - loss: 0.4737 - accuracy: 0.0000e+00
      Epoch 5/10
      Epoch 6/10
      1053/1053 [=
                       ===============] - 1s 682us/step - loss: 0.4735 - accuracy: 0.0000e+00
      Epoch 7/10
      Epoch 8/10
      1053/1053 [=
                       ============= ] - 1s 678us/step - loss: 0.4733 - accuracy: 0.0000e+00
      Epoch 9/10
      1053/1053 [==
                       ========] - 1s 700us/step - loss: 0.4733 - accuracy: 0.0000e+00
      Epoch 10/10
      Out[37]: <keras.callbacks.History at 0x1f0651912e0>
In [38]:
      model new.evaluate(X test,y test)
      Out[38]: [0.492971271276474, 0.0]
In [ ]:
```