Following are the results given by Weka after running 10-fold cross-validation test using the **Artificial Neural Network.**

Choice of parameters:

To arrive at the optimal parameter, I first increased the complexity of the model by increasing the number of hidden units and number of epochs. By default weka uses 'a' as number of hidden units which is (no. of features + no. of labels)/2. In my case value of a is 3 so, I went on increasing the number of hidden units from 3. I kept on increasing the complexity (no. of epoch + no. of hidden units) until validation error starts increasing. The point from where the validation error starts increasing is the sweet spot. Following are the results of the values I tried.

Default parameters:

# of hidden units = a

# of epochs = 500

```
Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation:     occupancy_detection_training_dataset-weka.filters.unsupervised.attribute.Remove-R1
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0
    Inputs    Weights
    Threshold    3.2916045588343605
    Node 2    -6.230942820630276
    Node 3    -9.763423325745997
    Node 4    -10.563742208607561
Sigmoid Node 1
    Inputs    Weights
    Threshold    -3.2916045606105917
    Node 2    6.230942821007807
    Node 3    9.763423306848606
    Node 4    10.563742294892165
Sigmoid Node 2
    Inputs    Weights
    Threshold    -18.937963993475027
    Attrib Temperature    -2.6592016676921735
    Attrib Humidity    0.15315038076260587
    Attrib Light    7.071596295150851
    Attrib CO2    -22.373259301674505
    Attrib HumidityRatio    -0.493130003497677
Sigmoid Node 3
    Inputs    Weights
    Threshold    -16.007289225054702
    Attrib Temperature    4.226748817819117
    Attrib Humidity    -2.043190934116511
    Attrib Light    -6.520263359047805
    Attrib CO2    -9.563125107448446
    Attrib HumidityRatio    -1.4848315450180811
Sigmoid Node 4
    Inputs    Weights
    Threshold    -16.09896397056628
    Attrib Temperature    0.7233423287537966
```

```
Sigmoid Node 4
    Inputs    Weights
    Threshold    -16.09896397056628
    Attrib Temperature    0.7233423287537966
    Attrib Humidity    1.2620377971330141
    Attrib Light    -18.53720341331366
    Attrib CO2    -4.003723490637238
    Attrib HumidityRatio    1.4272943478353914
Class Y
    Input
    Node 0
Class N
    Input
    Node 1


Time taken to build model: 5.15 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8047               98.8211 %
Incorrectly Classified Instances        96                1.1789 %
Kappa statistic                          0.9653
Mean absolute error                      0.0198
Root mean squared error                  0.102
Relative absolute error                  5.9087 %
Root relative squared error             24.9529 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.993    0.013    0.953      0.993    0.973      0.966    0.994     0.964     Y
                 0.987    0.007    0.998      0.987    0.992      0.966    0.994     0.999     N
Weighted Avg.    0.988    0.008    0.989      0.988    0.988      0.966    0.994     0.991

=== Confusion Matrix ===

    a    b   <-- classified as
 1717   12 |    a = Y
   84 6330 |    b = N
```

Variation-1:

# of hidden units = 5

# of epochs = 550

```
=== Run information ===

Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.2 -M 0.3 -N 550 -V 0 -S 0 -E 20 -H 5 -R
Relation:     occupancy_detection_training_dataset-weka.filters.unsupervised.attribute.Remove-R1
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0
    Inputs    Weights
    Threshold    9.39500460655446
    Node 2    -10.962512337993358
    Node 3    -4.82262041217414
    Node 4    -5.037913375055205
    Node 5    -9.558051773991592
    Node 6    -8.978214605767382
Sigmoid Node 1
    Inputs    Weights
    Threshold    -9.394718604441103
    Node 2    10.964098925642592
    Node 3    4.820903339378419
    Node 4    5.035013709508026
    Node 5    9.557754155090668
    Node 6    8.978614389019752
Sigmoid Node 2
    Inputs    Weights
    Threshold    -13.715759133994975
    Attrib Temperature    -2.3825614670475557
    Attrib Humidity    1.0394767826314684
    Attrib Light    -19.375524513896305
    Attrib CO2    -0.8321788290912716
    Attrib HumidityRatio    -0.6496340119577556
Sigmoid Node 3
    Inputs    Weights
    Threshold    -7.7898962386624095
    Attrib Temperature    -4.21885170250878
    Attrib Humidity    2.4308320562909445
    Attrib Light    -10.594328718320595
```

```
Sigmoid Node 3
    Inputs    Weights
    Threshold    -7.7898962386624095
    Attrib Temperature    -4.21885170250878
    Attrib Humidity    2.4308320562909445
    Attrib Light    -10.594328718320595
    Attrib CO2    0.018021151660878434
    Attrib HumidityRatio    -0.005178491389300982
Sigmoid Node 4
    Inputs    Weights
    Threshold    -14.37620630678916
    Attrib Temperature    1.188319493217614
    Attrib Humidity    -2.9100799695728536
    Attrib Light    -5.889002408480739
    Attrib CO2    -6.058137247807416
    Attrib HumidityRatio    -2.9322659373452487
Sigmoid Node 5
    Inputs    Weights
    Threshold    -13.957293654050812
    Attrib Temperature    19.286314789525562
    Attrib Humidity    -0.43056192237103885
    Attrib Light    7.389167135886162
    Attrib CO2    -21.01258795370854
    Attrib HumidityRatio    6.8634961310482385
Sigmoid Node 6
    Inputs    Weights
    Threshold    -17.63387907009164
    Attrib Temperature    -3.338800763062421
    Attrib Humidity    -1.7161197100687418
    Attrib Light    -7.309606459508696
    Attrib CO2    -12.529071545840473
    Attrib HumidityRatio    -2.1508962908688924
Class Y
    Input
    Node 0
Class N
    Input
    Node 1


Time taken to build model: 7.33 seconds
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8062               99.0053 %
Incorrectly Classified Instances        81                0.9947 %
Kappa statistic                          0.9705
Mean absolute error                      0.0131
Root mean squared error                  0.0863
Relative absolute error                  3.9151 %
Root relative squared error             21.1112 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.986    0.009    0.968      0.986    0.977      0.971    0.998     0.991     Y
                 0.991    0.014    0.996      0.991    0.994      0.971    0.998     1.000     N
Weighted Avg.    0.990    0.013    0.990      0.990    0.990      0.971    0.998     0.998

=== Confusion Matrix ===

    a    b   <-- classified as
 1704   25 |    a = Y
   56 6358 |    b = N
```

# of hidden units = 7

# of epochs = 600

```
Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.2 -M 0.3 -N 600 -V 0 -S 0 -E 20 -H 7 -R
Relation:     occupancy_detection_training_dataset-weka.filters.unsupervised.attribute.Remove-R1
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0
    Inputs    Weights
    Threshold    6.2315827757188265
    Node 2    -6.629944267355875
    Node 3    11.074871518102606
    Node 4    -10.382078443536768
    Node 5    -9.018237371748715
    Node 6    2.443086607832403
    Node 7    -8.523218263762336
    Node 8    -9.240358654730953
Sigmoid Node 1
    Inputs    Weights
    Threshold    -6.231425029764954
    Node 2    6.630092746927222
    Node 3    -11.074790320024503
    Node 4    10.381931418321638
    Node 5    9.023644847803618
    Node 6    -2.4431422075877465
    Node 7    8.517232517010582
    Node 8    9.240195315155887
Sigmoid Node 2
    Inputs    Weights
    Threshold    -7.774974173752215
    Attrib Temperature    -7.95015272846982
    Attrib Humidity    3.3133663730346137
    Attrib Light    -9.904337540998462
    Attrib CO2    2.4238284503979237
    Attrib HumidityRatio    0.494861519629287
```

```
Sigmoid Node 3
    Inputs    Weights
    Threshold    10.465638060129335
    Attrib Temperature    -12.305163273794747
    Attrib Humidity    -0.3752774422419033
    Attrib Light    -0.44161844275452716
    Attrib CO2    17.00703406609774
    Attrib HumidityRatio    -8.88398997275322
Sigmoid Node 4
    Inputs    Weights
    Threshold    -18.037783408665607
    Attrib Temperature    1.2443989839121448
    Attrib Humidity    -0.4805195494485443
    Attrib Light    1.7323165343458777
    Attrib CO2    -20.241916404911176
    Attrib HumidityRatio    -1.293788093950194
Sigmoid Node 5
    Inputs    Weights
    Threshold    -12.20148819678552
    Attrib Temperature    -3.3417160424639447
    Attrib Humidity    1.189828261207414
    Attrib Light    -17.458593386596142
    Attrib CO2    0.23820714795444542
    Attrib HumidityRatio    -0.8255998800958338
Sigmoid Node 6
    Inputs    Weights
    Threshold    -1.482374251778646
    Attrib Temperature    3.1028587093985984
    Attrib Humidity    2.0792572622644414
    Attrib Light    3.9819613872523725
    Attrib CO2    -0.3696952998924179
    Attrib HumidityRatio    2.8053720067445003
Sigmoid Node 7
    Inputs    Weights
    Threshold    -11.829480218581299
    Attrib Temperature    -3.5288878294222146
    Attrib Humidity    1.4368446494483598
    Attrib Light    -16.807573916866556
    Attrib CO2    0.3252281917712792
    Attrib HumidityRatio    -0.7141995143177328
```

```
Sigmoid Node 8
    Inputs    Weights
    Threshold    7.907380898644457
    Attrib Temperature    -2.144055490860973
    Attrib Humidity    -15.566821662808511
    Attrib Light    11.939173154338276
    Attrib CO2    10.433544088364744
    Attrib HumidityRatio    -16.993138222486937
Class Y
    Input
    Node 0
Class N
    Input
    Node 1


Time taken to build model: 10.73 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8082               99.2509 %
Incorrectly Classified Instances        61                0.7491 %
Kappa statistic                          0.9777
Mean absolute error                      0.0112
Root mean squared error                  0.0801
Relative absolute error                  3.3625 %
Root relative squared error             19.582  %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.990    0.007    0.975      0.990    0.982      0.978    0.999     0.995     Y
                 0.993    0.010    0.997      0.993    0.995      0.978    0.999     1.000     N
Weighted Avg.    0.993    0.009    0.993      0.993    0.993      0.978    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 1712   17 |    a = Y
   44 6370 |    b = N
```

# of hidden units = 9

# of epochs = 700

```
Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.2 -M 0.3 -N 700 -V 0 -S 0 -E 20 -H 9 -R
Relation:     occupancy_detection_training_dataset-weka.filters.unsupervised.attribute.Remove-R1
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0
    Inputs    Weights
    Threshold    5.369437300868006
    Node 2    10.416162495386688
    Node 3    -15.75547706023244
    Node 4    7.387887668769237
    Node 5    -5.51564670215589
    Node 6    -5.841126117201219
    Node 7    -8.807591952531201
    Node 8    -5.155983932952063
    Node 9    -6.448559696204894
    Node 10    -9.591745893162713
Sigmoid Node 1
    Inputs    Weights
    Threshold    -5.370017171393903
    Node 2    -10.413703480501816
    Node 3    15.75133678722431
    Node 4    -7.389109039368038
    Node 5    5.524457055250044
    Node 6    5.858651955360995
    Node 7    8.807517996894342
    Node 8    5.136916263914755
    Node 9    6.440245647256688
    Node 10    9.592545948795234
```

```
Sigmoid Node 2
    Inputs    Weights
    Threshold    3.8628463706974685
    Attrib Temperature    -12.276879084480854
    Attrib Humidity    2.346885147787072
    Attrib Light    -21.72309488457433
    Attrib CO2    19.030101895758637
    Attrib HumidityRatio    -2.20269472464665
Sigmoid Node 3
    Inputs    Weights
    Threshold    -17.06041539552619
    Attrib Temperature    -1.8206805548830853
    Attrib Humidity    0.9041717411754472
    Attrib Light    -22.444329448215704
    Attrib CO2    -2.836122871050121
    Attrib HumidityRatio    -0.5706027779415082
Sigmoid Node 4
    Inputs    Weights
    Threshold    12.724088145676372
    Attrib Temperature    -10.001959896726966
    Attrib Humidity    -5.368070452944032
    Attrib Light    11.471226344826214
    Attrib CO2    14.910463398899905
    Attrib HumidityRatio    -8.949750551878918
Sigmoid Node 5
    Inputs    Weights
    Threshold    -9.92881299961259
    Attrib Temperature    -2.392525482375226
    Attrib Humidity    -1.6717518551348276
    Attrib Light    -1.2300185475635346
    Attrib CO2    -8.107379242761747
    Attrib HumidityRatio    -2.07938831586714
Sigmoid Node 6
    Inputs    Weights
    Threshold    -9.65594165709486
    Attrib Temperature    -7.04606520363934
    Attrib Humidity    0.3809339655824108
    Attrib Light    5.307538962049195
    Attrib CO2    -8.297523463101717
    Attrib HumidityRatio    -0.7921832462577817
```

```
Sigmoid Node 7
    Inputs    Weights
    Threshold    -5.824632865325182
    Attrib Temperature    -5.144073784167161
    Attrib Humidity    5.4081006459606575
    Attrib Light    -12.361571521667907
    Attrib CO2    0.9054193615047683
    Attrib HumidityRatio    3.217172076650094
Sigmoid Node 8
    Inputs    Weights
    Threshold    -9.97135110372233
    Attrib Temperature    -3.710003263687852
    Attrib Humidity    -0.5135483683987686
    Attrib Light    2.27834554210439
    Attrib CO2    -9.201280348356589
    Attrib HumidityRatio    -1.2515851581157942
Sigmoid Node 9
    Inputs    Weights
    Threshold    -8.714267194477163
    Attrib Temperature    -7.553107647495935
    Attrib Humidity    0.584922460286435
    Attrib Light    7.0832572806705425
    Attrib CO2    -8.221551539390292
    Attrib HumidityRatio    -0.5719401471302381
Sigmoid Node 10
    Inputs    Weights
    Threshold    4.405621910539365
    Attrib Temperature    7.160929414946816
    Attrib Humidity    -15.581082159338647
    Attrib Light    15.845385026011092
    Attrib CO2    11.381815817610528
    Attrib HumidityRatio    -14.653465729500091
Class Y
    Input
    Node 0
Class N
    Input
    Node 1
```

```
Time taken to build model: 16.16 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8079               99.214  %
Incorrectly Classified Instances        64                0.786  %
Kappa statistic                          0.9766
Mean absolute error                      0.0107
Root mean squared error                  0.0803
Relative absolute error                  3.2119 %
Root relative squared error             19.6433 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.988    0.007    0.975      0.988    0.982      0.977    0.998     0.992     Y
                 0.993    0.012    0.997      0.993    0.995      0.977    0.998     1.000     N
Weighted Avg.    0.992    0.011    0.992      0.992    0.992      0.977    0.998     0.998

=== Confusion Matrix ===

    a    b   <-- classified as
 1708   21 |    a = Y
   43 6371 |    b = N
```

# of hidden units = 10

# of epochs = 750

```
Class Y
    Input
    Node 0
Class N
    Input
    Node 1


Time taken to build model: 18.72 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8074               99.1526 %
Incorrectly Classified Instances        69                0.8474 %
Kappa statistic                          0.9748
Mean absolute error                      0.0109
Root mean squared error                  0.0805
Relative absolute error                  3.2436 %
Root relative squared error             19.6926 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.986    0.007    0.975      0.986    0.980      0.975    0.999     0.994     Y
                 0.993    0.014    0.996      0.993    0.995      0.975    0.999     1.000     N
Weighted Avg.    0.992    0.013    0.992      0.992    0.992      0.975    0.999     0.998

=== Confusion Matrix ===

    a    b   <-- classified as
 1704   25 |    a = Y
   44 6370 |    b = N
```

As you can see from above that after # of hidden units = 7 and # of epochs = 600 with increase of complexity the accuracy goes down and validation error keeps on rising. So, I will choose # of hidden units to be 7 and epochs to be 600 for achieving maximum accuracy = 99.2509 and lowest validation error = 0.7491. This would be my first variation of ANN. These values also suggest that the dataset has much more variations which cannot be accurately covered by default value of hidden units. Complexity had to be increased to some level in order to achieve the desired accuracy.

Variation-2:

First variation takes more than 10 seconds to build the model. I have tried to increase the learning rate to make the learning process faster without losing the accuracy and making sure that learning rate shouldn't increase beyond the point where it starts overshooting and never converge at the lowest point. In order to prevent this overshooting I tried to increase the momentum so that my model gets to know direction of next step based on the previous step. After trying different combination of values I decided to use learning rate = 0.3 and momentum = 0.5. Using these values and keeping values of other parameters same as first variation, learning time for the model reduces to around 9 seconds but, with a slight decrease in

accuracy. We will see in experimenter whether this decrease is statistically significant or not. Following are the results shown by Weka

```
Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.5 -N 600 -V 0 -S 0 -E 20 -H 7
Relation:     occupancy_detection_training_dataset-weka.filters.unsupervised.attribute.Remove-R1
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

Time taken to build model: 9.14 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8078               99.2018 %
Incorrectly Classified Instances        65                0.7982 %
Kappa statistic                          0.9762
Mean absolute error                      0.0104
Root mean squared error                  0.0787
Relative absolute error                  3.1235 %
Root relative squared error             19.2446 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.987    0.007    0.976      0.987    0.981      0.976    0.999     0.994     Y
                 0.993    0.013    0.996      0.993    0.995      0.976    0.999     1.000     N
Weighted Avg.    0.992    0.012    0.992      0.992    0.992      0.976    0.999     0.998

=== Confusion Matrix ===

    a    b   <-- classified as
 1706   23 |    a = Y
   42 6372 |    b = N
```

Variation-3:

In order to give much more stability, to reduce complexity of the model I decided appropriate values for validation set size and batch size such that it does not lower the accuracy significantly. Percentage of validation set size is used to terminate the training process by preempting number of epochs. With the help of this complexity of the model is also reduced to avoid overfitting and it gives the best estimate for the test error. I have kept validation set size to 10 percent by keeping in mind the size of the dataset as the dataset does not have large number of records, increasing from 10 percent would not give enough cases to train the model and lowering from 10 percent will be useless as it will not give the best estimate for the test

error. Similarly, larger the batch size greater should be the speed of the learning process but, for this dataset if I increase the batch size from default value of 100, learning process gets lower and lower. Which indicates that for batch size > 100 the model starts to lose its ability to generalize hence, resulting in inaccurate estimate of the test error.
So, for this variation I just set validation set size to 10 with values of other parameters same as in variation-2.

```
Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.5 -N 600 -V 10 -S 0 -E 20 -H 7
Relation:     occupancy_detection_training_dataset-weka.filters.unsupervised.attribute.Remove-R1
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation


Time taken to build model: 4.62 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8065               99.0421 %
Incorrectly Classified Instances        78                0.9579 %
Kappa statistic                          0.9717
Mean absolute error                      0.016
Root mean squared error                  0.094
Relative absolute error                  4.7698 %
Root relative squared error             22.9918 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.992    0.010    0.964      0.992    0.978      0.972    0.997     0.983     Y
                 0.990    0.008    0.998      0.990    0.994      0.972    0.997     0.999     N
Weighted Avg.    0.990    0.008    0.991      0.990    0.990      0.972    0.997     0.996

=== Confusion Matrix ===

    a    b   <-- classified as
 1716   13 |    a = Y
   65 6349 |    b = N
```

Variation-4:

Radial Basis function with default parameters. Main advantage of RBF over MLP is that it takes less time to train. As you can see below it takes only 0.3 seconds to train the model, much faster than the tuned versions of ANN but, with decrease in accuracy. We will see in experimenter whether this decrease is significant or not.

```
Scheme:       weka.classifiers.functions.RBFNetwork -B 2 -S 1 -R 1.0E-8 -M -1 -W 0.1
Relation:     occupancy_detection_training_dataset-weka.filters.unsupervised.attribute.Remove-R1
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Radial basis function network
(Logistic regression applied to K-means clusters as basis functions):

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                   Class
Variable               Y
========================
pCluster_0_0       6.4922
pCluster_0_1       9.1627
pCluster_1_0      -0.7915
pCluster_1_1      -5.3475
Intercept          -3.486


Odds Ratios...
                   Class
Variable               Y
========================
pCluster_0_0     659.9414
pCluster_0_1    9534.2983
pCluster_1_0       0.4532
pCluster_1_1       0.0048


Time taken to build model: 0.3 seconds
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8028               98.5877 %
Incorrectly Classified Instances       115                1.4123 %
Kappa statistic                          0.9581
Mean absolute error                      0.0255
Root mean squared error                  0.1142
Relative absolute error                  7.6144 %
Root relative squared error             27.913  %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.978    0.012     0.956      0.978     0.967      0.958     0.993     0.971     Y
                 0.988    0.022     0.994      0.988     0.991      0.958     0.993     0.998     N
Weighted Avg.    0.986    0.020     0.986      0.986     0.986      0.958     0.993     0.992

=== Confusion Matrix ===

    a    b   <-- classified as
 1691   38 |    a = Y
   77 6337 |    b = N
```

Weka Experimenter:

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -V -result-matrix "weka.experiment.ResultMatrixPla
Analysing:  Percent_correct
Datasets:   1
Resultsets: 5
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       10/24/19 2:43 PM


Dataset                   (1) functions.Mult | (2) functions.M (3) functions.M (4) functions.M (5) functions.R
----------------------------------------------------------------------------------------------------------------
occupancy_detection_train(100)   98.85(0.40) |   99.15(0.32) v   99.12(0.38) v   98.95(0.43)     98.26(0.63) *
----------------------------------------------------------------------------------------------------------------
                                      (v/ /*) |         (1/0/0)         (1/0/0)         (0/1/0)         (0/0/1)


Key:
(1) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -5990607817048210779
(2) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 600 -V 0 -S 0 -E 20 -H 7' -5990607817048210779
(3) functions.MultilayerPerceptron '-L 0.3 -M 0.5 -N 600 -V 0 -S 0 -E 20 -H 7' -5990607817048210779
(4) functions.MultilayerPerceptron '-L 0.3 -M 0.5 -N 600 -V 10 -S 0 -E 20 -H 7' -5990607817048210779
(5) functions.RBFClassifier '-N 2 -R 0.01 -L 1.0E-6 -C 2 -P 1 -E 1 -S 1' -7847475556438394611
```

Note: Order of classifier is same as mentioned above e-g 1st column is with default version 2nd column is the variation-1 third column is the variation-2 and so on.

As you can see from results of the experiment that accuracy of both Variation-1 and Variation-2 is higher than the default version of ANN and that increase in accuracy is also statistically

significant. Which tells us that the model needs to be much more complex as in both Variation-1 and Variation-2 has high number of hidden units as well as the high number of epochs. Variation-3 does provide the better generalization by reducing the complexity and time required for training but, comparative to default ANN configuration its accuracy is neither significantly higher not lower. Radial Basis function accuracy is significantly lower than the default version of ANN which also makes it significantly lower than Variation-1 and Variaton-2. So in terms of accuracy RBF can be ruled out but it has one serious advantage i.e. it is too much fast to train as compared to all the variations of ANN. Now, let's compare values against Variation-1 as the test base.

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -V -result-matrix "weka.experiment.ResultMatrixPla
Analysing:  Percent_correct
Datasets:   1
Resultsets: 5
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       10/24/19 5:43 PM


Dataset                   (2) functions.Mult | (1) functions.M (3) functions.M (4) functions.M (5) functions.R
----------------------------------------------------------------------------------------------------------------
occupancy_detection_train(100)   99.15(0.32) |   98.85(0.40) *   99.12(0.38)     98.95(0.43)     98.26(0.63) *
----------------------------------------------------------------------------------------------------------------
                                       (v/ /*) |         (0/0/1)         (0/1/0)         (0/1/0)         (0/0/1)


Key:
(1) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -5990607817048210779
(2) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 600 -V 0 -S 0 -E 20 -H 7' -5990607817048210779
(3) functions.MultilayerPerceptron '-L 0.3 -M 0.5 -N 600 -V 0 -S 0 -E 20 -H 7' -5990607817048210779
(4) functions.MultilayerPerceptron '-L 0.3 -M 0.5 -N 600 -V 10 -S 0 -E 20 -H 7' -5990607817048210779
(5) functions.RBFClassifier '-N 2 -R 0.01 -L 1.0E-6 -C 2 -P 1 -E 1 -S 1' -7847475556438394611
```

Note: order is preserved just position of default and Variation-1 of ANN is switched

It can be observed that there is no significance difference between Variation-1 and Variation-2. So we can choose any one of these models. Variation-2 would be ideal if we are short on time for training as it takes less time to train.

Statistical test comparison to previous methods:

Previous methods I used were Naive Bayes Classifier, Logistic Regression, and Decision Trees (J48).
Before diving into the comparison I would like to explain the problem briefly and which statistical tests are important for the use case.

Problem is to minimize the energy consumption by accurate determination of occupancy detection in buildings. It has been estimated that if we are able to predict the occupancy 100% correctly, then we can save 30% to 42% of the energy. Other than the overall accuracy, the accuracy of precision and recall values also matters. Precision of the N (No occupancy) class suggests me that this result is good. Because, in order to save the energy, you do not want to predict Y (Occupancy is there) when actual result is N (No occupancy), in this case chance to save the energy will be lost. So here is the comparison.

Precision of the N class:

Naive Bayes	Logistic Regression	Decision Tree (J48) without pruning	Decision Tree (J48) with pruning	ANN(Variation-1)	ANN (Variation-2)
99.8%	99.6%	99.7	99.7	99.7	99.6

For this metric, values of all of the models are sufficient for our use use case.

Recall of Y class:

Harm caused by wrong prediction depends on the type of building/place in which we are detecting occupancy. If we are detecting occupancy in some type of office, hotel etc. then precision of N class matters. But, in sensitive places like hospitals and other places where sensitive work is being done, where we don't want to predict N class when actual result is Y class then recall of Y class is a good measure to judge.

Naive Bayes	Logistic Regression	Decision Tree (J48) without pruning	Decision Tree (J48) with pruning	ANN(Variation-1)	ANN (Variation-2)
99.4%	98.7%	98.8	98.8	99.0	98.7

Naive Bayes wins the battle in this case.

ROC Area:

It is a good measure for comparing different classifiers as it tells how well the model will do in different thresholds by achieving nice trade-off between sensitivity and specificity.

Naive Bayes	Logistic Regression	Decision Tree (J48) without pruning	Decision Tree (J48) with pruning	ANN(Variation-1)	ANN (Variation-2)
99.2%	98.7%	99.3	99.4	99.9	99.9

Variations of ANN are doing extremely well in this case.

Accuracy:

Naive Bayes	Logistic Regression	Decision Tree (J48) without pruning	Decision Tree (J48) with pruning	ANN(Variation-1)	ANN (Variation-2)
97.64%	98.6%	99.37%	99.38	99.15	99.12

Variations of ANN are doing extremely well in this case.

Decision Tree versions performs extremely well in this case. Lower accuracy of Naive Bayes tells us that there is some degree of dependency among the features which is not captured by Naive Bayes and that is why variations of both ANN and Decision Trees are performing better as they capture that dependency in terms of higher complexity.

Final Verdict:

I would personally go with Decision Tree for this use case. Because it takes much lesser time to train than ANN and in terms of accuracy it outperforms everyone. It does not have the best values for other metrics but its values are very much closer to the best values. In order to remove problem of overfitting the, pruned version will do the job.