First, I will explore all algorithms with explorer and will try different parameter settings of algorithms and compare their Accuracy and ROC Area. Finally, selected Algorithms will be highlighted yellow for the comparison in the end.

In subsequent steps/process configuration of each algorithm will be compared with its previous best version.

K-nearest neighbors:

K=1:

```
Scheme:       weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification


Time taken to build model: 0.02 seconds

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

Correctly Classified Instances        8096               99.4228 %
Incorrectly Classified Instances        47                0.5772 %
Kappa statistic                          0.9827
Mean absolute error                      0.0059
Root mean squared error                  0.076
Relative absolute error                  1.7624 %
Root relative squared error             18.5746 %
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    0.004    0.987      0.986    0.986      0.983    0.992     0.978     Y
                 0.996    0.014    0.996      0.996    0.996      0.983    0.992     0.996     N
Weighted Avg.    0.994    0.012    0.994      0.994    0.994      0.983    0.992     0.993

=== Confusion Matrix ===

    a    b   <-- classified as
 1705   24 |    a = Y
   23 6391 |    b = N
```

K = 3:

```
Scheme:       weka.classifiers.lazy.IBk -K 3 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

IB1 instance-based classifier
using 3 nearest neighbour(s) for classification


Time taken to build model: 0.01 seconds

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

Correctly Classified Instances        8093               99.386  %
Incorrectly Classified Instances        50                0.614  %
Kappa statistic                          0.9816
Mean absolute error                      0.0071
Root mean squared error                  0.0682
Relative absolute error                  2.1297 %
Root relative squared error             16.6793 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.985    0.004    0.986      0.985    0.986      0.982    0.997     0.991     Y
                 0.996    0.015    0.996      0.996    0.996      0.982    0.997     0.998     N
Weighted Avg.    0.994    0.013    0.994      0.994    0.994      0.982    0.997     0.997

=== Confusion Matrix ===

    a    b   <-- classified as
 1703   26 |    a = Y
   24 6390 |    b = N
```

K=5:

```
Scheme:       weka.classifiers.lazy.IBk -K 5 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

IB1 instance-based classifier
using 5 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

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

Correctly Classified Instances        8082               99.2509 %
Incorrectly Classified Instances        61                0.7491 %
Kappa statistic                          0.9776
Mean absolute error                      0.0079
Root mean squared error                  0.068
Relative absolute error                  2.3498 %
Root relative squared error             16.6246 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.985    0.005    0.980      0.985    0.982      0.978    0.998     0.996     Y
                 0.995    0.015    0.996      0.995    0.995      0.978    0.998     0.999     N
Weighted Avg.    0.993    0.013    0.993      0.993    0.993      0.978    0.998     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 1703   26 |    a = Y
   35 6379 |    b = N
```

K = 7:

```
Scheme:       weka.classifiers.lazy.IBk -K 7 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

IB1 instance-based classifier
using 7 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

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

Correctly Classified Instances        8092               99.3737 %
Incorrectly Classified Instances        51                0.6263 %
Kappa statistic                          0.9813
Mean absolute error                      0.0082
Root mean squared error                  0.0673
Relative absolute error                  2.4546 %
Root relative squared error             16.4554 %
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    0.005    0.981      0.990    0.985      0.981    0.998     0.997     Y
                 0.995    0.010    0.997      0.995    0.996      0.981    0.998     0.999     N
Weighted Avg.    0.994    0.009    0.994      0.994    0.994      0.981    0.998     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 1711   18 |    a = Y
   33 6381 |    b = N
```

K=10:

```
Scheme:       weka.classifiers.lazy.IBk -K 10 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

IB1 instance-based classifier
using 10 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

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

Correctly Classified Instances        8086               99.3    %
Incorrectly Classified Instances        57                0.7    %
Kappa statistic                          0.9791
Mean absolute error                      0.009
Root mean squared error                  0.0694
Relative absolute error                  2.6977 %
Root relative squared error             16.9736 %
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    0.006    0.979      0.988    0.984      0.979    0.999     0.997     Y
                 0.994    0.012    0.997      0.994    0.996      0.979    0.999     0.999     N
Weighted Avg.    0.993    0.010    0.993      0.993    0.993      0.979    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 1709   20 |    a = Y
   37 6377 |    b = N
```

K=15:

```
Scheme:       weka.classifiers.lazy.IBk -K 15 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

IB1 instance-based classifier
using 15 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

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

Correctly Classified Instances        8084               99.2755 %
Incorrectly Classified Instances        59                0.7245 %
Kappa statistic                          0.9785
Mean absolute error                      0.0104
Root mean squared error                  0.0742
Relative absolute error                  3.1105 %
Root relative squared error             18.1514 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.991    0.007    0.975      0.991    0.983      0.979    0.999     0.997     Y
                 0.993    0.009    0.998      0.993    0.995      0.979    0.999     1.000     N
Weighted Avg.    0.993    0.008    0.993      0.993    0.993      0.979    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 1714   15 |    a = Y
   44 6370 |    b = N
```

Explanation:

I have tried different values of K to achieve best balance between Accuracy and ROC Area. One thing I noted during this process is that, as I increase the value of K accuracy starts to decrease but ROC Area starts to increase. This tells us that there is not much noise in the dataset and dataset is well distributed. If there was a noise in the dataset then upon increasing the value of K the accuracy would have increased. Increase in ROC Area suggests that the distribution is little skewed i.e. we have larger number of training examples of one class label than other and increase in value of K improves ROC Area to deal with the skewed distribution. From above I have chosen value of K = 10 at this value I got the best balance between ROC Area and Accuracy. I will compare KNN with other algorithms by keeping value of K to 10.

Ensembles:

Bagging:

```
Scheme:       weka.classifiers.meta.Bagging -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

Bagging with 10 iterations and base learner

weka.classifiers.trees.J48 -C 0.25 -M 2

Time taken to build model: 0.5 seconds

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

Correctly Classified Instances        8090               99.3491 %
Incorrectly Classified Instances        53                0.6509 %
Kappa statistic                          0.9806
Mean absolute error                      0.009
Root mean squared error                  0.0712
Relative absolute error                  2.6818 %
Root relative squared error             17.4111 %
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    0.005    0.983      0.987    0.985      0.981    0.997     0.992     Y
                 0.995    0.013    0.996      0.995    0.996      0.981    0.997     0.998     N
Weighted Avg.    0.993    0.011    0.994      0.993    0.993      0.981    0.997     0.997

=== Confusion Matrix ===

    a    b   <-- classified as
 1706   23 |    a = Y
   30 6384 |    b = N
```

Explanation:

I have chosen J48 (Decision Tree) classifier for bagging. Because it is the unstable algorithm which responds to random fluctuations in the training data. Bagging reduces the error if a classifier is unstable. Its accuracy is relatively good than KNN (K=10) but its ROC area is lower. Lower ROC area makes sense because, main purpose of bagging is to reduce variance and maintains the bias and we learned from KNN experimentation that there is a skewed distribution of data points and bagging does not resolve that problem as well as KNN (K=10) does. I will use this configuration to compare bagging with other algorithms.

Adaboost:

```
=== Run information ===

Scheme:       weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

AdaBoostM1: Base classifiers and their weights:

J48 pruned tree


Time taken to build model: 1.1 seconds

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

Correctly Classified Instances        8099               99.4597 %
Incorrectly Classified Instances        44                0.5403 %
Kappa statistic                          0.9839
Mean absolute error                      0.0055
Root mean squared error                  0.0721
Relative absolute error                  1.6293 %
Root relative squared error             17.6226 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.989    0.004    0.986      0.989    0.987      0.984    0.999     0.996     Y
                 0.996    0.011    0.997      0.996    0.997      0.984    0.999     1.000     N
Weighted Avg.    0.995    0.009    0.995      0.995    0.995      0.984    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 1710   19 |    a = Y
   25 6389 |    b = N
```

Explanation:

I have used J-48 because of the reasons mentioned in bagging. So far, this is the best balance of accuracy and ROC Area I have achieved for this dataset and why not, as Adaboost focus on wrongly classified examples. But, because of its too much focus on misclassified examples there is a tendency that model built by adaboost will be much more complex in other words there is a slight chance of overfitting,

resulting in poor generalization performance. Adaboost is also susceptible to noise and outliers but, high Accuracy and ROC Area validates our understanding from previous algorithms that there is no noise in the dataset. By focusing on misclassified examples Adaboost considers to solve the problems caused by skewed distribution that is why its accuracy is higher with high ROC Area.

Random Forests:

With default settings:

```
Scheme:       weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 1.39 seconds

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

Correctly Classified Instances        8094               99.3983 %
Incorrectly Classified Instances        49                0.6017 %
Kappa statistic                          0.982
Mean absolute error                      0.0079
Root mean squared error                  0.065
Relative absolute error                  2.3747 %
Root relative squared error             15.8822 %
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    0.004    0.984      0.988    0.986      0.982    0.999     0.998     Y
                 0.996    0.012    0.997      0.996    0.996      0.982    0.999     1.000     N
Weighted Avg.    0.994    0.010    0.994      0.994    0.994      0.982    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 1708   21 |    a = Y
   28 6386 |    b = N
```

Number of attributes = 2:

```
Scheme:       weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 2 -M 1.0 -V 0.001 -S 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 2 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 0.78 seconds

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

Correctly Classified Instances        8097               99.4351 %
Incorrectly Classified Instances        46                0.5649 %
Kappa statistic                          0.9831
Mean absolute error                      0.0083
Root mean squared error                  0.0644
Relative absolute error                  2.4944 %
Root relative squared error             15.7584 %
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    0.004    0.985      0.988    0.987      0.983    0.999     0.998     Y
                 0.996    0.012    0.997      0.996    0.996      0.983    0.999     1.000     N
Weighted Avg.    0.994    0.010    0.994      0.994    0.994      0.983    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 1709   20 |    a = Y
   26 6388 |    b = N
```

Number of attributes = 3:

```
Scheme:       weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 3 -M 1.0 -V 0.001 -S 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 3 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 1.06 seconds

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

Correctly Classified Instances        8094               99.3983 %
Incorrectly Classified Instances        49                0.6017 %
Kappa statistic                          0.982
Mean absolute error                      0.0079
Root mean squared error                  0.065
Relative absolute error                  2.3747 %
Root relative squared error             15.8822 %
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    0.004    0.984      0.988    0.986      0.982    0.999     0.998     Y
                 0.996    0.012    0.997      0.996    0.996      0.982    0.999     1.000     N
Weighted Avg.    0.994    0.010    0.994      0.994    0.994      0.982    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 1708   21 |    a = Y
   28 6386 |    b = N
```

Number of attributes = 4:

```
Scheme:       weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 4 -M 1.0 -V 0.001 -S 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 4 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 1.25 seconds

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

Correctly Classified Instances        8094               99.3983 %
Incorrectly Classified Instances        49                0.6017 %
Kappa statistic                          0.982
Mean absolute error                      0.008
Root mean squared error                  0.0665
Relative absolute error                  2.4011 %
Root relative squared error             16.267  %
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    0.004    0.985      0.987    0.986      0.982    0.999     0.998     Y
                 0.996    0.013    0.996      0.996    0.996      0.982    0.999     1.000     N
Weighted Avg.    0.994    0.011    0.994      0.994    0.994      0.982    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 1706   23 |    a = Y
   26 6388 |    b = N
```

Explanation:

Random Forests does a great job in achieving nice trade-off between variance and bias. For all of the settings ROC Area is consistent i.e. 0.999 which is highest but accuracy is maximum with hyper parameter (number of attributes) = 2. If you increase the number of parameters from 2 the accuracy drops down. The Reason for this could be that at value of 2 correlation among the classifiers is low and their strength is maximum. On further increase of number of attributes the correlation between the classifiers starts to increase. Value of 2 for number of attributes is also according to the suggested value that is log2d where d is the total number of attributes so, this value is justified for this dataset. We will proceed with the Random Forests with number of attributes = 2 for comparisons.

Support Vector Machines:

Default Settings (Gamma = 0.0, Kernel = RBF, C=1.0):

```
Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model D:\weka\Weka-3-9 -seed 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 11.11 seconds

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

Correctly Classified Instances        7757               95.2597 %
Incorrectly Classified Instances       386                4.7403 %
Kappa statistic                          0.8685
Mean absolute error                      0.0474
Root mean squared error                  0.2177
Relative absolute error                 14.1697 %
Root relative squared error             53.2383 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.994    0.059    0.821      0.994    0.899      0.875    0.968     0.817     Y
                 0.941    0.006    0.998      0.941    0.969      0.875    0.968     0.986     N
Weighted Avg.    0.953    0.017    0.961      0.953    0.954      0.875    0.968     0.950

=== Confusion Matrix ===

    a    b   <-- classified as
 1719   10 |    a = Y
  376 6038 |    b = N
```

Gamma = 5, Kernel = RBF, C = 1:

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 2 -D 3 -G 0.5 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model D:\weka\Weka-3-9 -seed 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 11.25 seconds

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

Correctly Classified Instances        7684               94.3633 %
Incorrectly Classified Instances       459                5.6367 %
Kappa statistic                          0.8458
Mean absolute error                      0.0564
Root mean squared error                  0.2374
Relative absolute error                 16.8494 %
Root relative squared error             58.0546 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.994    0.070    0.793      0.994    0.882      0.855    0.962     0.790     Y
                 0.930    0.006    0.998      0.930    0.963      0.855    0.962     0.984     N
Weighted Avg.    0.944    0.019    0.955      0.944    0.946      0.855    0.962     0.942

=== Confusion Matrix ===

    a    b   <-- classified as
 1719   10 |    a = Y
  449 5965 |    b = N
```

Gamma = 0, Kernel = RBF, C = 2:

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 2.0 -E 0.001 -P 0.1 -model D:\weka\Weka-3-9 -seed 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 13.17 seconds

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

Correctly Classified Instances        7771               95.4317 %
Incorrectly Classified Instances       372                4.5683 %
Kappa statistic                          0.8728
Mean absolute error                      0.0457
Root mean squared error                  0.2137
Relative absolute error                 13.6557 %
Root relative squared error             52.2639 %
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    0.056    0.827      0.993    0.902      0.879    0.968     0.822     Y
                 0.944    0.007    0.998      0.944    0.970      0.879    0.968     0.986     N
Weighted Avg.    0.954    0.017    0.962      0.954    0.956      0.879    0.968     0.951

=== Confusion Matrix ===

    a    b   <-- classified as
 1717   12 |    a = Y
  360 6054 |    b = N
```

Gamma = 0, Kernel = RBF, C = 3:

```
Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 3.0 -E 0.001 -P 0.1 -model D:\weka\Weka-3-9 -seed 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 11.09 seconds

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

Correctly Classified Instances        7771               95.4317 %
Incorrectly Classified Instances       372                4.5683 %
Kappa statistic                          0.8728
Mean absolute error                      0.0457
Root mean squared error                  0.2137
Relative absolute error                 13.6557 %
Root relative squared error             52.2639 %
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    0.056    0.827      0.993    0.902      0.879    0.968     0.822     Y
                 0.944    0.007    0.998      0.944    0.970      0.879    0.968     0.986     N
Weighted Avg.    0.954    0.017    0.962      0.954    0.956      0.879    0.968     0.951

=== Confusion Matrix ===

    a    b   <-- classified as
 1717   12 |    a = Y
  360 6054 |    b = N
```

Gamma = 0, Kernel = RBF, C = 5:

```
Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 5.0 -E 0.001 -P 0.1 -model D:\weka\Weka-3-9 -seed 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 13.63 seconds

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

Correctly Classified Instances        7771               95.4317 %
Incorrectly Classified Instances       372                4.5683 %
Kappa statistic                          0.8728
Mean absolute error                      0.0457
Root mean squared error                  0.2137
Relative absolute error                 13.6557 %
Root relative squared error             52.2639 %
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    0.056    0.827      0.993    0.902      0.879    0.968     0.822     Y
                 0.944    0.007    0.998      0.944    0.970      0.879    0.968     0.986     N
Weighted Avg.    0.954    0.017    0.962      0.954    0.956      0.879    0.968     0.951

=== Confusion Matrix ===

    a    b   <-- classified as
 1717   12 |    a = Y
  360 6054 |    b = N
```

Gamma = 0, Kernel = Linear, C = 1:

```
Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model D:\weka\Weka-3-9 -seed 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 31.42 seconds

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

Correctly Classified Instances        7926               97.3351 %
Incorrectly Classified Instances       217                2.6649 %
Kappa statistic                          0.9197
Mean absolute error                      0.0266
Root mean squared error                  0.1632
Relative absolute error                  7.9658 %
Root relative squared error             39.9172 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.927    0.014    0.947      0.927    0.937      0.920    0.956     0.893     Y
                 0.986    0.073    0.980      0.986    0.983      0.920    0.956     0.978     N
Weighted Avg.    0.973    0.061    0.973      0.973    0.973      0.920    0.956     0.960

=== Confusion Matrix ===

    a    b   <-- classified as
 1602  127 |    a = Y
   90 6324 |    b = N
```

Gamma = 0, Kernel = Linear, C = 3:

```
Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 3.0 -E 0.001 -P 0.1 -model D:\weka\Weka-3-9 -seed 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 41.65 seconds

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

Correctly Classified Instances        8046               98.8088 %
Incorrectly Classified Instances        97                1.1912 %
Kappa statistic                          0.965
Mean absolute error                      0.0119
Root mean squared error                  0.1091
Relative absolute error                  3.5608 %
Root relative squared error             26.688  %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.996    0.014    0.950      0.996    0.973      0.965    0.991     0.947     Y
                 0.986    0.004    0.999      0.986    0.992      0.965    0.991     0.996     N
Weighted Avg.    0.988    0.006    0.989      0.988    0.988      0.965    0.991     0.986

=== Confusion Matrix ===

    a    b   <-- classified as
 1722    7 |    a = Y
   90 6324 |    b = N
```

Gamma = 0, Kernel = Linear, C = 5:

```
Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 5.0 -E 0.001 -P 0.1 -model D:\weka\Weka-3-9 -seed 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 52.52 seconds

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

Correctly Classified Instances        7227               88.7511 %
Incorrectly Classified Instances       916               11.2489 %
Kappa statistic                          0.7037
Mean absolute error                      0.1125
Root mean squared error                  0.3354
Relative absolute error                 33.6254 %
Root relative squared error             82.0121 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.920    0.121    0.672      0.920    0.776      0.719    0.899     0.635     Y
                 0.879    0.080    0.976      0.879    0.925      0.719    0.899     0.953     N
Weighted Avg.    0.888    0.089    0.911      0.888    0.893      0.719    0.899     0.886

=== Confusion Matrix ===

    a    b   <-- classified as
 1591  138 |    a = Y
  778 5636 |    b = N
```

Gamma = 0.2, Kernel = Linear, C = 3:

```
Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.2 -R 0.0 -N 0.5 -M 40.0 -C 3.0 -E 0.001 -P 0.1 -model D:\weka\Weka-3-9 -seed 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 23.3 seconds

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

Correctly Classified Instances        8046               98.8088 %
Incorrectly Classified Instances        97                1.1912 %
Kappa statistic                          0.965
Mean absolute error                      0.0119
Root mean squared error                  0.1091
Relative absolute error                  3.5608 %
Root relative squared error             26.688  %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.996    0.014    0.950      0.996    0.973      0.965    0.991     0.947     Y
                 0.986    0.004    0.999      0.986    0.992      0.965    0.991     0.996     N
Weighted Avg.    0.988    0.006    0.989      0.988    0.988      0.965    0.991     0.986

=== Confusion Matrix ===

    a    b   <-- classified as
 1722    7 |    a = Y
   90 6324 |    b = N
```

Gamma = 0.4, Kernel = Linear, C = 3:

```
Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.4 -R 0.0 -N 0.5 -M 40.0 -C 3.0 -E 0.001 -P 0.1 -model D:\weka\Weka-3-9 -seed 1
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 22.89 seconds

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

Correctly Classified Instances        8046               98.8088 %
Incorrectly Classified Instances        97                1.1912 %
Kappa statistic                          0.965
Mean absolute error                      0.0119
Root mean squared error                  0.1091
Relative absolute error                  3.5608 %
Root relative squared error             26.688  %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.996    0.014    0.950      0.996    0.973      0.965    0.991     0.947     Y
                 0.986    0.004    0.999      0.986    0.992      0.965    0.991     0.996     N
Weighted Avg.    0.988    0.006    0.989      0.988    0.988      0.965    0.991     0.986

=== Confusion Matrix ===

    a    b   <-- classified as
 1722    7 |    a = Y
   90 6324 |    b = N
```

Explanation:

I have tried various parameters combination in SVM. First in default settings, around 95% accuracy is achieved with ROC Area = 0.968. After increasing the value of gamma in default settings, Accuracy and ROC Area both starts to drop. Which tells us that grouping of data points is much wider and extends away from the plausible separator created by RBF. After this observation, I stopped tuning value of gamma and reset it to 0 for further observations, because If gamma is too large, the radius of the area of influence of the support vectors only includes the support vector itself and no amount of regularization with C will be able to prevent overfitting. It also gave me the notion that the data points are not complex enough. But, I carried on to tune the C parameter, which is to control how many number of misclassifications we can allow. After increasing the value of C to 2 accuracy started to increase but ROC Area remained same as was in when gamma = 0.0 and C=1.0. On further increase of value of C to 3 and 5 statistics remains the same. Which again confirms the first notion that distribution is not complex enough otherwise on increase of complexity with increase in value of C bias would have become much lower and accuracy would have increased. Then I thought if distribution is not complex for RBF then I

should try Linear Kernel. Straightaway I got the better results with Linear Kernel (Gamma = 0.0, C = 1.0) accuracy increased by 2% i.e. 97% with slight decrease in ROC Area to 0.95. This validates our idea that distribution is linearly separable. So I tuned parameter with Linear Kernel on increasing the value of C to 3 the accuracy increased to around 99% and ROC Area to around 0.99 which is a significant increase. This increase suggests that there were some misclassified training examples which were placed close to group of other class label and also despite being linearly separable distribution there is a little complexity and that's why on increasing value of C both accuracy and ROC Area is increased. On further increasing the value of C to 5 the accuracy and ROC Area both plummeted. Accuracy to approx. 88% and ROC Area to 0.89. Which again validate our intuition that there is not much complexity to learn. Then I decided to tune gamma = 0.2 and 0.4 with C = 3, there was no change in accuracy and ROC Area. But time to build model reduces from 42 seconds to 22 seconds. So the best configuration for SVM I got is (Gamma = 0.4, Kernel = Linear, C = 3).

After above analysis I selected following algorithms with configurations for further analysis:

- KNN (K=10)
- Bagging (Classifier = J48)
- Adaboost (Classifier =J48)
- Random Forests (Number of attributes = 2)
- SVM (Gamma = 0.4, Kernel = Linear, C = 3)

Algorithm Comparison with experimenter:

Let's compare above selected algorithms and the best version of algorithms that we got in previous home works i.e. Naive Bayes, Logistic Regression, Decision Tree (pruned, Confidence factor = 0.2), ANN (hidden units = 5, epochs = 550).

Let's put ZeroR as test base for initial comparison.

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -V -result-matrix "weka.experiment.ResultMatrixPlainText -mean-prec 2 -stddev-prec 2 -col-name-width 0 -row-name-w
Analysing:  Percent_correct
Datasets:   1
Resultsets: 10
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       11/12/19 12:19 AM


Dataset                   (1) rules.ZeroR '' | (2) bayes.Naive (3) functions.L (4) trees.J48 ' (5) functions.M (6) lazy.IBk '- (7) meta.Baggin (8) meta.AdaBoo (9) trees.Rando (10) functions.
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
occupancy_detection_train (10)   78.77(0.04) |   97.69(0.52) v   98.60(0.27) v   99.39(0.24) v   99.03(0.34) v   99.30(0.25) v   99.35(0.27) v   99.46(0.31) v   99.44(0.27) v   98.81(0.42) v
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
                                    (v/ /*) |         (1/0/0)         (1/0/0)         (1/0/0)         (1/0/0)         (1/0/0)         (1/0/0)         (1/0/0)         (1/0/0)         (1/0/0)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) bayes.NaiveBayes '' 5995231201785697655
(3) functions.Logistic '-R 1.0E-8 -M -1 -num-decimal-places 4' 3932117032546553727
(4) trees.J48 '-C 0.2 -M 2' -217733168393644444
(5) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 550 -V 0 -S 0 -E 20 -H 5' -5990607817048210779
(6) lazy.IBk '-K 10 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -3080186098777067172
(7) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -115879962237199703
(8) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -1178107808933117974
(9) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 2 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(10) functions.LibSVM '-S 0 -K 0 -D 3 -G 0.4 -R 0.0 -N 0.5 -M 40.0 -C 3.0 -E 0.001 -P 0.1 -model D:\\weka\\Weka-3-9 -seed 1' 14172
```

All of these algorithms are significantly better than ZeroR. For the next step lets pick Decision Tree (J-48) as test base. Because this was the best algorithm in the previous home works. Lets see how newly tested algorithms performs in comparison with J48.

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -V -result-matrix "weka.experiment.ResultMatrixPlainText -mean-prec 2 -stddev-prec 2 -col-name-width 0 -row-name-width 25 -
Analysing:  Percent_correct
Datasets:   1
Resultsets: 10
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       11/12/19 12:35 AM


Dataset                   (4) trees.J48 '-C  | (1) rules.ZeroR (2) bayes.Naive (3) functions.L (5) functions.M (6) lazy.IBk '- (7) meta.Baggin (8) meta.AdaBoo (9) trees.Rando (10) functions.
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
occupancy_detection_train (10)   99.39(0.24) |   78.77(0.04) *   97.69(0.52) *   98.60(0.27) *   99.03(0.34) *   99.30(0.25)     99.35(0.27)     99.46(0.31)     99.44(0.27)     98.81(0.42) *
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
                                    (v/ /*) |         (0/0/1)         (0/0/1)         (0/0/1)         (0/0/1)         (0/1/0)         (0/1/0)         (0/1/0)         (0/1/0)         (0/0/1)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) bayes.NaiveBayes '' 5995231201785697655
(3) functions.Logistic '-R 1.0E-8 -M -1 -num-decimal-places 4' 3932117032546553727
(4) trees.J48 '-C 0.2 -M 2' -217733168393644444
(5) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 550 -V 0 -S 0 -E 20 -H 5' -5990607817048210779
(6) lazy.IBk '-K 10 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -3080186098777067172
(7) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -115879962237199703
(8) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -1178107808933117974
(9) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 2 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(10) functions.LibSVM '-S 0 -K 0 -D 3 -G 0.4 -R 0.0 -N 0.5 -M 40.0 -C 3.0 -E 0.001 -P 0.1 -model D:\\weka\\Weka-3-9 -seed 1' 14172
```

Note: function in the last column above is SVM.

No new algorithm performed significantly better than J48. SVM from performed significantly lower than J48. Performance of SVM is not bad itself but, it's the ability of Decision Tree to capture complexity and dependency between attributes in much better way by responding to small perturbation makes it better classifier for this dataset. Now lets pick the one with highest accuracy as test base which is Adaboost.

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrixPlainText -mean-prec 2 -stddev-
Analysing:  Percent_correct
Datasets:   1
Resultsets: 10
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       11/12/19 12:51 AM


Dataset                   (8) meta.Ada | (1) rules (2) bayes (3) funct (4) trees (5) funct (6) lazy. (7) meta. (9) trees (10) func
-----------------------------------------------------------------------------------------------------------------------------------
occupancy_detection_train (10)   99.46 |   78.77 *   97.69 *   98.60 *   99.39     99.03 *   99.30     99.35     99.44     98.81 *
-----------------------------------------------------------------------------------------------------------------------------------
                               (v/ /*) |   (0/0/1)   (0/0/1)   (0/0/1)   (0/1/0)   (0/0/1)   (0/1/0)   (0/1/0)   (0/1/0)   (0/0/1)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) bayes.NaiveBayes '' 5995231201785697655
(3) functions.Logistic '-R 1.0E-8 -M -1 -num-decimal-places 4' 3932117032546553727
(4) trees.J48 '-C 0.2 -M 2' -217733168393644444
(5) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 550 -V 0 -S 0 -E 20 -H 5' -5990607817048210779
(6) lazy.IBk '-K 10 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -3080186098777067172
(7) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -1158799622371997O3
(8) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -1178107808933117974
(9) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 2 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(10) functions.LibSVM '-S 0 -K 0 -D 3 -G 0.4 -R 0.0 -N 0.5 -M 40.0 -C 3.0 -E 0.001 -P 0.1 -model D:\\weka\\Weka-3-9 -seed 1' 14172
```

Note: (4) column is decision trees (6) is KNN (7) is Bagging (9) Random Forests (10) is SVM

Adaboost, Bagging, Decision Trees, KNN and Random Forests needs further comparison in order to pick the best out of them. But, before this comparison lets squeeze as much as accuracy as we can by applying dimensionality reduction and resampling.

Dimensionality reduction:

I will apply following dimensionality reduction techniques to above selected algorithms and other best algorithms that were found for this dataset. In order to remove noisy attributes (if any), reduce overfitting (less complexity), make the model much more interpretable and to improve accuracy and ROC Area.

1) Correlation of attributes with respect to target:

```
Evaluator:     weka.attributeSelection.CfsSubsetEval -P 1 -E 1
Search:        weka.attributeSelection.BestFirst -D 1 -N 5
Relation:      occupancy_detection_training_dataset
Instances:     8143
Attributes:    6
               Temperature
               Humidity
               Light
               CO2
               HumidityRatio
               Occupancy
Evaluation mode:    evaluate on all training data


=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 16
        Merit of best subset found:    0.754

Attribute Subset Evaluator (supervised, Class (nominal): 6 Occupancy):
        CFS Subset Evaluator
        Including locally predictive attributes

Selected attributes: 3,4 : 2
                     Light
                     CO2
```

After applying this technique, Light and CO2 comes out to be the most correlated attribute with target.

Let's apply selected algorithms by reducing dataset to these attributes.

KNN (K=10):

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

Correctly Classified Instances        8053               98.8948 %
Incorrectly Classified Instances        90                1.1052 %
Kappa statistic                          0.9675
Mean absolute error                      0.0163
Root mean squared error                  0.0959
Relative absolute error                  4.8698 %
Root relative squared error             23.4522 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.996    0.013    0.954      0.996    0.975      0.968    0.998     0.986     Y
                 0.987    0.004    0.999      0.987    0.993      0.968    0.998     0.999     N
Weighted Avg.    0.989    0.006    0.989      0.989    0.989      0.968    0.998     0.997

=== Confusion Matrix ===

    a    b   <-- classified as
 1722    7 |    a = Y
   83 6331 |    b = N
```

Both Accuracy and ROC Area reduces. That makes sense as KNN considers weightage of all the attributes to be equal and after removing some of these attributes KNN has lost some information which was correlated enough with target class to achieve high accuracy and ROC Area. So, we will proceed with KNN without Reduced Dimensions according to this filter/ technique.

Bagging (Classifier = J48):

```
Scheme:       weka.classifiers.meta.Bagging -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Relation:     occupancy_detection_training_dataset-weka.filters.unsupervised.attribute.Remove-R1-2,5
Instances:    8143
Attributes:   3
              Light
              CO2
              Occupancy
Test mode:    10-fold cross-validation

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

Bagging with 10 iterations and base learner

weka.classifiers.trees.J48 -C 0.25 -M 2

Time taken to build model: 0.36 seconds

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

Correctly Classified Instances        8065               99.0421 %
Incorrectly Classified Instances        78                0.9579 %
Kappa statistic                          0.9717
Mean absolute error                      0.0175
Root mean squared error                  0.095
Relative absolute error                  5.2174 %
Root relative squared error             23.2311 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.994    0.011    0.962      0.994    0.978      0.972    0.997     0.987     Y
                 0.989    0.006    0.998      0.989    0.994      0.972    0.997     0.999     N
Weighted Avg.    0.990    0.007    0.991      0.990    0.990      0.972    0.997     0.996

=== Confusion Matrix ===

    a    b   <-- classified as
 1719   10 |    a = Y
   68 6346 |    b = N
```

Accuracy reduces, but not that much (0.4%) but ROC Area stays the same. Time to build the model reduces to 0.36 s from 0.5s. We could have afforded the loss of this much accuracy for our use case, if there was significant reduction in time to build the model but, that is not the case so we will proceed with bagging without reduced dimensions.

Adaboost (Classifier =J48):

```
Time taken to build model: 0.44 seconds

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

Correctly Classified Instances        8057               98.9439 %
Incorrectly Classified Instances        86                1.0561 %
Kappa statistic                          0.9686
Mean absolute error                      0.0107
Root mean squared error                  0.1016
Relative absolute error                  3.2015 %
Root relative squared error             24.8411 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.982    0.009    0.969      0.982    0.975      0.969    0.998     0.989     Y
                 0.991    0.018    0.995      0.991    0.993      0.969    0.998     0.999     N
Weighted Avg.    0.989    0.016    0.990      0.989    0.989      0.969    0.998     0.997

=== Confusion Matrix ===

    a    b   <-- classified as
 1698   31 |    a = Y
   55 6359 |    b = N
```

Random Forests (Number of attributes = 2):

```
Scheme:       weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 2 -M 1.0 -V 0.001 -S 1
Relation:     occupancy_detection_training_dataset-weka.filters.unsupervised.attribute.Remove-R1-2,5
Instances:    8143
Attributes:   3
              Light
              CO2
              Occupancy
Test mode:    10-fold cross-validation

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

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 2 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 0.5 seconds

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

Correctly Classified Instances        8057               98.9439 %
Incorrectly Classified Instances        86                1.0561 %
Kappa statistic                          0.9686
Mean absolute error                      0.0146
Root mean squared error                  0.092
Relative absolute error                  4.3695 %
Root relative squared error             22.4981 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.984    0.009    0.967      0.984    0.975      0.969    0.997     0.988     Y
                 0.991    0.016    0.996      0.991    0.993      0.969    0.997     0.999     N
Weighted Avg.    0.989    0.015    0.990      0.989    0.989      0.969    0.997     0.997

=== Confusion Matrix ===

    a    b   <-- classified as
 1701   28 |    a = Y
   58 6356 |    b = N
```

Both Accuracy and ROC Area reduces in both of the above algorithms.

SVM (Gamma = 0.4, Kernel = Linear, C = 3):

```
Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.4 -R 0.0 -N 0.5 -M 40.0 -C 3.0 -E 0.001 -P 0.1 -model D:\weka\Weka-3-9 -seed 1
Relation:     occupancy_detection_training_dataset-weka.filters.unsupervised.attribute.Remove-R1-2,5
Instances:    8143
Attributes:   3
              Light
              CO2
              Occupancy
Test mode:    10-fold cross-validation

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

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 14.85 seconds

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

Correctly Classified Instances        8048               98.8334 %
Incorrectly Classified Instances        95                1.1666 %
Kappa statistic                          0.9657
Mean absolute error                      0.0117
Root mean squared error                  0.108
Relative absolute error                  3.4873 %
Root relative squared error             26.4114 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.997    0.014    0.951      0.997    0.973      0.966    0.991     0.948     Y
                 0.986    0.003    0.999      0.986    0.993      0.966    0.991     0.996     N
Weighted Avg.    0.988    0.006    0.989      0.988    0.988      0.966    0.991     0.986

=== Confusion Matrix ===

    a    b   <-- classified as
 1723    6 |    a = Y
   89 6325 |    b = N
```

ROC Area remains the same but, accuracy increases from 98.8088 to 98.334 also, time taken to build the model reduces to 14.85 s from 22 s. So, there is a significant gain in reducing dimensionality in case of SVM. Increase in accuracy tells us that, with C=3 we have introduced some complexity in the model and dimensionality reduction reduces that complexity to give better generalization. So we will definitely proceed with SVM with reduced dimensions according to correlation with target.

Following Algorithms are from previous home works and values/ statistics are compared from the values that were obtained in previous home works.

Naive Bayes:

```
Time taken to build model: 0.02 seconds

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

Correctly Classified Instances        7994               98.1702 %
Incorrectly Classified Instances       149                1.8298 %
Kappa statistic                          0.9469
Mean absolute error                      0.0184
Root mean squared error                  0.1259
Relative absolute error                  5.5069 %
Root relative squared error             30.7855 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    0.023    0.922      0.998    0.959      0.948    0.993     0.949     Y
                 0.977    0.002    0.999      0.977    0.988      0.948    0.993     0.998     N
Weighted Avg.    0.982    0.007    0.983      0.982    0.982      0.948    0.993     0.988

=== Confusion Matrix ===

    a    b   <-- classified as
 1725    4 |    a = Y
  145 6269 |    b = N
```

In this case Accuracy increases from 97.64% to 98.1702% and ROC Area from 0.992 to 0.993. So we will move forward with Naive Bayes with these dimensions.

Logistic Regression:

```
Time taken to build model: 0.14 seconds

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

Correctly Classified Instances        8049               98.8456 %
Incorrectly Classified Instances        94                1.1544 %
Kappa statistic                          0.9661
Mean absolute error                      0.0325
Root mean squared error                  0.1188
Relative absolute error                  9.7087 %
Root relative squared error             29.0434 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.997    0.014    0.951      0.997    0.973      0.966    0.993     0.948     Y
                 0.986    0.003    0.999      0.986    0.993      0.966    0.993     0.998     N
Weighted Avg.    0.988    0.006    0.989      0.988    0.989      0.966    0.993     0.988

=== Confusion Matrix ===

    a    b   <-- classified as
 1723    6 |    a = Y
   88 6326 |    b = N
```

Accuracy increases from 98.6 % to 98.8456% But ROC Area drops to 0.993 from 0.995. Since increase in accuracy is not significant enough and for our use case we cannot afford to lose ROC Area.

Decision Tree (pruning, Confidence factor = 0.2):

```
Time taken to build model: 0.01 seconds

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

Correctly Classified Instances        8058               98.9562 %
Incorrectly Classified Instances        85                1.0438 %
Kappa statistic                          0.9692
Mean absolute error                      0.0179
Root mean squared error                  0.1
Relative absolute error                  5.361  %
Root relative squared error             24.4461 %
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    0.011    0.959      0.993    0.976      0.969    0.991     0.954     Y
                 0.989    0.007    0.998      0.989    0.993      0.969    0.991     0.996     N
Weighted Avg.    0.990    0.008    0.990      0.990    0.990      0.969    0.991     0.987

=== Confusion Matrix ===

    a    b   <-- classified as
 1717   12 |    a = Y
   73 6341 |    b = N
```

Similar to Logistic regression its accuracy increases to 98.96% from 98.6% but ROC Area decreases to 0.991 from 0.995. Since, we cannot afford loss in ROC Area so it is not worth to consider it.

ANN (hidden units = 5, epochs= 550):

```
Time taken to build model: 6.72 seconds

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

Correctly Classified Instances        8042               98.7597 %
Incorrectly Classified Instances       101                1.2403 %
Kappa statistic                          0.9635
Mean absolute error                      0.0193
Root mean squared error                  0.1036
Relative absolute error                  5.7751 %
Root relative squared error             25.3286 %
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    0.014    0.951      0.992    0.971      0.964    0.996     0.968     Y
                 0.986    0.008    0.998      0.986    0.992      0.964    0.996     0.999     N
Weighted Avg.    0.988    0.009    0.988      0.988    0.988      0.964    0.996     0.992

=== Confusion Matrix ===

    a    b   <-- classified as
 1716   13 |    a = Y
   88 6326 |    b = N
```

Both Accuracy and ROC Areas goes down from the values we obtained in previous home works.

2) Principal Component Analysis.

Each Algorithm will be compared with its best output that it produced in previous steps / configurations, including in process of filtering attributes according to correlation with target.

```
Evaluator:     weka.attributeSelection.PrincipalComponents -R 0.95 -A 5
Search:        weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation:      occupancy_detection_training_dataset
Instances:     8143
Attributes:    6
               Temperature
               Humidity
               Light
               CO2
               HumidityRatio
               Occupancy
Evaluation mode:    evaluate on all training data


=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (unsupervised):
        Principal Components Attribute Transformer

Correlation matrix
  1      -0.14    0.65    0.56    0.15
 -0.14    1       0.04    0.44    0.96
  0.65    0.04    1       0.66    0.23
  0.56    0.44    0.66    1       0.63
  0.15    0.96    0.23    0.63    1


eigenvalue      proportion      cumulative
  2.73659         0.54732         0.54732       -0.55CO2-0.501HumidityRatio-0.414Light-0.396Humidity-0.344Temperature
  1.69948         0.3399          0.88721       0.574Humidity-0.536Temperature-0.445Light+0.414HumidityRatio-0.12CO2
  0.34874         0.06975         0.95696       -0.713Temperature+0.665Light-0.19HumidityRatio+0.111CO2+0.009Humidity

Eigenvectors
 V1      V2      V3
-0.3439 -0.5359 -0.7134 Temperature
-0.3957  0.5741  0.0093 Humidity
-0.4142 -0.4446  0.6654 Light
-0.5501 -0.1201  0.111  CO2
-0.5011  0.4137 -0.1896 HumidityRatio

Ranked attributes:
 0.4527  1 -0.55CO2-0.501HumidityRatio-0.414Light-0.396Humidity-0.344Temperature
 0.1128  2 0.574Humidity-0.536Temperature-0.445Light+0.414HumidityRatio-0.12CO2
 0.043   3 -0.713Temperature+0.665Light-0.19HumidityRatio+0.111CO2+0.009Humidity

Selected attributes: 1,2,3 : 3
```

PCA gives temperature, humidity and light as the most important attributes. Let's try these attributes with our selected algorithms.

KNN (K=10):

```
Scheme:       weka.classifiers.lazy.IBk -K 10 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     occupancy_detection_training_dataset-weka.filters.unsupervised.attribute.Remove-R4-5
Instances:    8143
Attributes:   4
              Temperature
              Humidity
              Light
              Occupancy
Test mode:    10-fold cross-validation

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

IB1 instance-based classifier
using 10 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

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

Correctly Classified Instances        8084               99.2755 %
Incorrectly Classified Instances        59                0.7245 %
Kappa statistic                          0.9784
Mean absolute error                      0.0107
Root mean squared error                  0.0752
Relative absolute error                  3.2022 %
Root relative squared error             18.3931 %
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    0.006    0.977      0.990    0.983      0.978    0.999     0.996     Y
                 0.994    0.010    0.997      0.994    0.995      0.978    0.999     1.000     N
Weighted Avg.    0.993    0.010    0.993      0.993    0.993      0.978    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 1711   18 |    a = Y
   41 6373 |    b = N
```

ROC Area stays the same but accuracy decreases by 0.03%. So we will not proceed with it.

Bagging (Classifier = J48):

```
Scheme:       weka.classifiers.meta.Bagging -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Relation:     occupancy_detection_training_dataset-weka.filters.unsupervised.attribute.Remove-R4-5
Instances:    8143
Attributes:   4
              Temperature
              Humidity
              Light
              Occupancy
Test mode:    10-fold cross-validation

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

Bagging with 10 iterations and base learner

weka.classifiers.trees.J48 -C 0.25 -M 2

Time taken to build model: 0.3 seconds

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

Correctly Classified Instances        8085               99.2877 %
Incorrectly Classified Instances        58                0.7123 %
Kappa statistic                          0.9788
Mean absolute error                      0.0102
Root mean squared error                  0.0758
Relative absolute error                  3.0565 %
Root relative squared error             18.5303 %
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    0.005    0.980      0.987    0.983      0.979    0.998     0.989     Y
                 0.995    0.013    0.996      0.995    0.995      0.979    0.998     0.999     N
Weighted Avg.    0.993    0.012    0.993      0.993    0.993      0.979    0.998     0.997

=== Confusion Matrix ===

    a    b   <-- classified as
 1706   23 |    a = Y
   35 6379 |    b = N
```

Accuracy decreases from 99.35 % to 99.29% but, ROC Area increases to 0.998 from 0.997. We can afford this small decrease in accuracy for increase in ROC Area and also the time to train the model also reduces to 0.3s from 0.5s. So we will select this configuration of SVM with these selected attributes.

Adaboost (Classifier =J48):

```
Time taken to build model: 0.53 seconds

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

Correctly Classified Instances        8084               99.2755 %
Incorrectly Classified Instances        59                0.7245 %
Kappa statistic                          0.9783
Mean absolute error                      0.0071
Root mean squared error                  0.0812
Relative absolute error                  2.1109 %
Root relative squared error             19.8504 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.983    0.005    0.983      0.983    0.983      0.978    0.999     0.994     Y
                 0.995    0.017    0.995      0.995    0.995      0.978    0.999     1.000     N
Weighted Avg.    0.993    0.014    0.993      0.993    0.993      0.978    0.999     0.998

=== Confusion Matrix ===

    a    b   <-- classified as
 1700   29 |    a = Y
   30 6384 |    b = N
```

Accuracy decreases by approx. 0.2% and ROC Area stays the same. So, we will not proceed with these attributes for Adaboost.

Random Forests (Number of attributes = 2):

```
RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 2 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 0.65 seconds

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

Correctly Classified Instances        8089               99.3369 %
Incorrectly Classified Instances        54                0.6631 %
Kappa statistic                          0.9802
Mean absolute error                      0.0086
Root mean squared error                  0.0692
Relative absolute error                  2.5749 %
Root relative squared error             16.9303 %
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    0.005    0.983      0.986    0.984      0.980    0.999     0.997     Y
                 0.995    0.014    0.996      0.995    0.996      0.980    0.999     1.000     N
Weighted Avg.    0.993    0.012    0.993      0.993    0.993      0.980    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 1704   25 |    a = Y
   29 6385 |    b = N
```

Accuracy drops to 99.3369 while ROC Area remains same and reduction in time taken to build model is not significant enough to consider this reduction in dimension.

SVM (Gamma = 0.4, Kernel = Linear, C = 3):

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

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 44.66 seconds

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

Correctly Classified Instances        7406               90.9493 %
Incorrectly Classified Instances       737                9.0507 %
Kappa statistic                          0.7653
Mean absolute error                      0.0905
Root mean squared error                  0.3008
Relative absolute error                 27.0545 %
Root relative squared error             73.5638 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.996    0.114    0.702      0.996    0.824      0.786    0.941     0.700     Y
                 0.886    0.004    0.999      0.886    0.939      0.786    0.941     0.975     N
Weighted Avg.    0.909    0.027    0.936      0.909    0.915      0.786    0.941     0.916

=== Confusion Matrix ===

    a    b   <-- classified as
 1722    7 |    a = Y
  730 5684 |    b = N
```

Both ROC Area and Accuracy drops down significantly by 0.05 and 8% respectively.

Naive Bayes:

```
Time taken to build model: 0.01 seconds

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

Correctly Classified Instances        7931               97.3965 %
Incorrectly Classified Instances       212                2.6035 %
Kappa statistic                          0.9254
Mean absolute error                      0.0257
Root mean squared error                  0.1537
Relative absolute error                  7.6882 %
Root relative squared error             37.5723 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.997    0.032    0.893      0.997    0.942      0.928    0.991     0.909     Y
                 0.968    0.003    0.999      0.968    0.983      0.928    0.990     0.998     N
Weighted Avg.    0.974    0.009    0.977      0.974    0.974      0.928    0.990     0.979

=== Confusion Matrix ===

    a    b   <-- classified as
 1724    5 |    a = Y
  207 6207 |    b = N
```

Both ROC Area and Accuracy drops down as compared to selected Naive Bayes Algorithm with reduced dimensions by filtering attributes with correlation with Target.

Logistic Regression:

```
Time taken to build model: 0.13 seconds

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

Correctly Classified Instances        8038               98.7105 %
Incorrectly Classified Instances       105                1.2895 %
Kappa statistic                          0.9622
Mean absolute error                      0.0314
Root mean squared error                  0.1155
Relative absolute error                  9.3919 %
Root relative squared error             28.2386 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.995    0.015    0.947      0.995    0.970      0.963    0.991     0.925     Y
                 0.985    0.005    0.999      0.985    0.992      0.963    0.991     0.998     N
Weighted Avg.    0.987    0.007    0.988      0.987    0.987      0.963    0.991     0.983

=== Confusion Matrix ===

    a    b   <-- classified as
 1721    8 |    a = Y
   97 6317 |    b = N
```

Same situation as in previous experiment, Accuracy increases but at the expense of ROC Area which is important for our use case. So we will not move forward with it.

Decision Tree (pruned, Confidence factor = 0.2):

```
Time taken to build model: 0.04 seconds

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

Correctly Classified Instances        8087               99.3123 %
Incorrectly Classified Instances        56                0.6877 %
Kappa statistic                          0.9794
Mean absolute error                      0.0097
Root mean squared error                  0.0783
Relative absolute error                  2.9066 %
Root relative squared error             19.1427 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.984    0.005    0.983      0.984    0.984      0.979    0.995     0.979     Y
                 0.995    0.016    0.996      0.995    0.996      0.979    0.995     0.998     N
Weighted Avg.    0.993    0.013    0.993      0.993    0.993      0.979    0.995     0.994

=== Confusion Matrix ===

    a    b   <-- classified as
 1702   27 |    a = Y
   29 6385 |    b = N
```

Its accuracy increases to 99.3123% from 98.6% and ROC Area stays the same i.e. 0.995. So we will select Decision Tree with this configuration.

ANN (hidden units = 5, epochs= 550):

```
Time taken to build model: 8.31 seconds

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

Correctly Classified Instances        8049               98.8456 %
Incorrectly Classified Instances        94                1.1544 %
Kappa statistic                          0.9661
Mean absolute error                      0.0178
Root mean squared error                  0.1022
Relative absolute error                  5.3304 %
Root relative squared error             24.9886 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.996    0.014    0.952      0.996    0.973      0.966    0.995     0.950     Y
                 0.986    0.004    0.999      0.986    0.993      0.966    0.995     0.999     N
Weighted Avg.    0.988    0.006    0.989      0.988    0.989      0.966    0.995     0.988

=== Confusion Matrix ===

    a    b   <-- classified as
 1722    7 |    a = Y
   87 6327 |    b = N
```

Both ROC Area and Accuracy goes down from the values we obtained from previous home works.

General Observation:

Generally, performance of algorithms on attributes selected by PCA is much better than on attributes selected by checking their correlation with target. This is because there is some degree of dependency between the attributes which is not captured by Correlation method. That is why it gives two unrelated attribute like Light and CO2. While PCA chooses related attributes like Temperature, Humidity and Light that is why algorithm performs better after doing PCA.

Following is the updated list of Selected Algorithm after experimenting with dimensionality reduction. I have attached snapshots for ANN and Logistic Regression from previous home works for reference as their statistics did not improve in dimensionality reduction phase.

- KNN (K=10)
- Adaboost (Classifier = J48)
- Random Forests (Number of attributes = 2)
- SVM (Gamma = 0.4, Kernel = Linear, C = 3, attribute selection technique = Correlation with target)
- Naive Bayes (attribute selection technique = Correlation with target)
- Bagging (Classifier = J48, attribute selection technique = PCA)
- Decision Tree (pruned, Confidence Factor = 0.2, attribute selection technique = PCA)
- Logistic Regression

```
Time taken to build model: 0.37 seconds

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

Correctly Classified Instances        8029               98.6    %
Incorrectly Classified Instances       114                1.4    %
Kappa statistic                          0.9587
Mean absolute error                      0.0275
Root mean squared error                  0.1133
Relative absolute error                  8.2223 %
Root relative squared error             27.7134 %
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    0.014    0.949      0.987    0.968      0.959    0.995     0.962     Y
                 0.986    0.013    0.996      0.986    0.991      0.959    0.995     0.999     N
Weighted Avg.    0.986    0.013    0.986      0.986    0.986      0.959    0.995     0.991

=== Confusion Matrix ===

    a    b   <-- classified as
 1706   23 |    a = Y
   91 6323 |    b = N
```

- ANN (hidden units = 5, epochs = 550)

```
=== 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
```

Resampling:

As the dataset contains imbalance class labels (# Y instances = 22%, # N instances = 78) so, I will apply resampling with class balancing option in the Weka to test whether accuracy of above selected algorithms increases or not.

KNN (K=10):

```
Scheme:       weka.classifiers.lazy.IBk -K 10 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   6
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

IB1 instance-based classifier
using 10 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

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

Correctly Classified Instances        8086               99.3    %
Incorrectly Classified Instances        57                0.7    %
Kappa statistic                          0.9791
Mean absolute error                      0.009
Root mean squared error                  0.0694
Relative absolute error                  2.6977 %
Root relative squared error             16.9736 %
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    0.006    0.979      0.988    0.984      0.979    0.999     0.997     Y
                 0.994    0.012    0.997      0.994    0.996      0.979    0.999     0.999     N
Weighted Avg.    0.993    0.010    0.993      0.993    0.993      0.979    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 1709   20 |    a = Y
   37 6377 |    b = N
```

Accuracy increases to 99.4634% (0.1% increase) and ROC Area increase stays the same. This improvement is convincing enough. This improvement makes sense as KNN takes majority vote if number of instances for one class labels will be higher than KNN will most likely to predict that class.

Adaboost (Classifier = J48):

```
Time taken to build model: 0.2 seconds

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

Correctly Classified Instances        8061.6864           99.0014 %
Incorrectly Classified Instances        81.3136            0.9986 %
Kappa statistic                          0.98
Mean absolute error                      0.0136
Root mean squared error                  0.0977
Relative absolute error                  2.7248 %
Root relative squared error             19.5472 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.994     0.014     0.986       0.994   0.990      0.980    0.997     0.997     Y
                 0.986     0.006     0.994       0.986   0.990      0.980    0.997     0.998     N
Weighted Avg.    0.990     0.010     0.990       0.990   0.990      0.980    0.997     0.997

=== Confusion Matrix ===

    a    b   <-- classified as
 4048   24 |    a = Y
   58 4014 |    b = N
```

Both ROC Area and Accuracy decreases. This is probably because Adaboost takes care of skewed distribution problem by increasing the weight of misclassified examples. So if we resample the data, same instances of class (that contained less instances) will appear in sub samples again and again.

Random Forests (Number of attributes = 2):

```
Time taken to build model: 0.57 seconds

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

Correctly Classified Instances        8090.436           99.3545 %
Incorrectly Classified Instances        52.564            0.6455 %
Kappa statistic                          0.9871
Mean absolute error                      0.0102
Root mean squared error                  0.0712
Relative absolute error                  2.0426 %
Root relative squared error             14.2367 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.994    0.007    0.993      0.994    0.994      0.987    1.000     0.999     Y
                 0.993    0.006    0.994      0.993    0.994      0.987    1.000     1.000     N
Weighted Avg.    0.994    0.006    0.994      0.994    0.994      0.987    1.000     1.000

=== Confusion Matrix ===

    a    b   <-- classified as
 4046   26 |    a = Y
   27 4045 |    b = N
```

Balance between ROC Area and Accuracy is more or less the same. Its accuracy goes down by 0.08% but ROC Area increases by 0.001s which is perfect. I will go with this one as when statistics are same you go with the one which takes less take and it takes 0.13 less seconds than previous best and also because of the perfect ROC Area which lead to better generalization.

SVM (Gamma = 0.4, Kernel = Linear, C = 3, attribute selection technique = Correlation with target):

For resampled data LibSVM options fades away in Weka, which is strange. So for further analysis I will use previous selected version of SVM.

Naive Bayes (attribute selection technique = Correlation with target):

```
Time taken to build model: 0.03 seconds

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

Correctly Classified Instances        8018.0501          98.4656 %
Incorrectly Classified Instances       124.9499           1.5344 %
Kappa statistic                          0.9693
Mean absolute error                      0.0146
Root mean squared error                  0.1141
Relative absolute error                  2.9214 %
Root relative squared error             22.817  %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    0.028    0.972      0.998    0.985      0.970    0.993     0.986     Y
                 0.972    0.002    0.998      0.972    0.984      0.970    0.993     0.995     N
Weighted Avg.    0.985    0.015    0.985      0.985    0.985      0.970    0.993     0.990

=== Confusion Matrix ===

    a       b      <-- classified as
 4062.08    9.42 |       a = Y
  115.53 3955.97 |       b = N
```

ROC Area stays the same accuracy increases to a significant extent i.e by 0.3%. So we will select this version of Naive Bayes with resampling.

Bagging (Classifier = J48, attribute selection technique = PCA):

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

Bagging with 10 iterations and base learner

weka.classifiers.trees.REPTree -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0

Time taken to build model: 0.21 seconds

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

Correctly Classified Instances        8080               99.2263 %
Incorrectly Classified Instances        63                0.7737 %
Kappa statistic                          0.9769
Mean absolute error                      0.0116
Root mean squared error                  0.0772
Relative absolute error                  3.4544 %
Root relative squared error             18.8857 %
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    0.006    0.978      0.986    0.982      0.977    0.998     0.990     Y
                 0.994    0.014    0.996      0.994    0.995      0.977    0.998     0.999     N
Weighted Avg.    0.992    0.012    0.992      0.992    0.992      0.977    0.998     0.997

=== Confusion Matrix ===

    a    b   <-- classified as
 1705   24 |    a = Y
   39 6375 |    b = N
```

ROC Area stays same accuracy goes down by to 99.2263%.

Decision Tree (pruned, Confidence Factor = 0.2, attribute selection technique = PCA):

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

Correctly Classified Instances        8087               99.3123 %
Incorrectly Classified Instances        56                0.6877 %
Kappa statistic                          0.9794
Mean absolute error                      0.0097
Root mean squared error                  0.0783
Relative absolute error                  2.9066 %
Root relative squared error             19.1427 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.984    0.005    0.983      0.984    0.984      0.979    0.995     0.979     Y
                 0.995    0.016    0.996      0.995    0.996      0.979    0.995     0.998     N
Weighted Avg.    0.993    0.013    0.993      0.993    0.993      0.979    0.995     0.994

=== Confusion Matrix ===

    a    b   <-- classified as
 1702   27 |    a = Y
   29 6385 |    b = N
```

Both ROC Area and Accuracy stays the same.

Logistic Regression:

```
Time taken to build model: 0.1 seconds

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

Correctly Classified Instances        8078.1702          99.2039 %
Incorrectly Classified Instances        64.8298           0.7961 %
Kappa statistic                          0.9841
Mean absolute error                      0.0206
Root mean squared error                  0.0936
Relative absolute error                  4.1263 %
Root relative squared error             18.7154 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    0.014    0.986      0.998    0.992      0.984    0.994     0.988     Y
                 0.986    0.002    0.998      0.986    0.992      0.984    0.994     0.996     N
Weighted Avg.    0.992    0.008    0.992      0.992    0.992      0.984    0.994     0.992

=== Confusion Matrix ===

    a       b      <-- classified as
 4064.44    7.06 |       a = Y
   57.77 4013.73 |       b = N
```

Accuracy increases by 0.6% and ROC Area decreases by 0.001. Normally, for this case I don't want to loose the ROC Area but, in this case increase in value of accuracy is enough to consider this for further experimentation.

ANN (hidden units = 5, epochs = 550):

```
Time taken to build model: 5.03 seconds

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

Correctly Classified Instances        8075.1806          99.1671 %
Incorrectly Classified Instances        67.8194           0.8329 %
Kappa statistic                          0.9833
Mean absolute error                      0.0092
Root mean squared error                  0.0896
Relative absolute error                  1.8453 %
Root relative squared error             17.9265 %
Total Number of Instances             8143

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    0.014    0.986      0.998    0.992      0.983    0.993     0.987     Y
                 0.986    0.002    0.998      0.986    0.992      0.983    0.993     0.996     N
Weighted Avg.    0.992    0.008    0.992      0.992    0.992      0.983    0.993     0.991

=== Confusion Matrix ===

    a       b      <-- classified as
 4062.08    9.42 |       a = Y
   58.4  4013.1  |       b = N
```

Even though accuracy increases by 0.14% but decrease in ROC Area is significant enough i.e. 0.993 from 0.998 to not proceed with this.

Detailed comparisons of selected algorithms with configurations:

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 (attribute selection technique = Correlation with target, resampled dataset)	**Logistic Regression (resampled dataset)**	**Decision Tree (pruned, Confidence Factor = 0.2, attribute selection technique = PCA)**	**ANN (hidden units = 5, epochs = 550)**	**KNN (K=10, resampled data)**	**Random Forests (Number of attributes = 2, resampled dataset):**	**Adaboost (Classifier = J48)**	**Bagging (Classifier = J48, attribute selection technique = PCA)**	**SVM (Gamma = 0.4, Kernel = Linear, C = 3, attribute selection technique = Correlation with target)**
99.8%	99.8%	99.6%	99.7%	99.8%	99.4%	99.7%	99.6%	99.9%

Here SVM wins the battle and Naive Bayes, Logistic Regression, KNN are lagging behind by 0.1%.

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 (attribute selection technique = Correlation with target, resampled dataset)	Logistic Regression (resampled dataset)	Decision Tree (pruned, Confidence Factor = 0.2, attribute selection technique = PCA)	ANN (hidden units = 5, epochs = 550)	KNN (K=10, resampled data)	Random Forests (Number of attributes = 2, resampled dataset):	Adaboost (Classifier = J48)	Bagging (Classifier = J48, attribute selection technique = PCA)	SVM (Gamma = 0.4, Kernel = Linear, C = 3, attribute selection technique = Correlation with target)
99.8%	99.8%	98.4%	99.0%	99.8%	99.4%	98.9%	98.7%	99.7%

Here Naive Bayes, Logistic Regression and KNN wins the Battle.

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 (attribute selection technique = Correlation with target, resampled dataset)	Logistic Regression (resampled dataset)	Decision Tree (pruned, Confidence Factor = 0.2, attribute selection technique = PCA)	ANN (hidden units = 5, epochs = 550)	KNN (K=10, resampled data)	Random Forests (Number of attributes = 2, resampled dataset):	Adaboost (Classifier = J48)	Bagging (Classifier = J48, attribute selection technique = PCA)	SVM (Gamma = 0.4, Kernel = Linear, C = 3, attribute selection technique = Correlation with target)
99.3%	99.4%	99.5%	99.9%	99.9%	100%	99.9%	99.8%	99.1%

Here Random Forests wins the battle with perfect number and KNN, Adaboost, ANN lags behind by 0.1%

Accuracy:

Naive Bayes (attribute selection technique = Correlation with target, resampled dataset)	Logistic Regression (resampled dataset)	Decision Tree (pruned, Confidence Factor = 0.2, attribute selection technique = PCA)	ANN (hidden units = 5, epochs = 550)	KNN (K=10, resampled data)	Random Forests (Number of attributes = 2, resampled dataset):	Adaboost (Classifier = J48)	Bagging (Classifier = J48, attribute selection technique = PCA)	SVM (Gamma = 0.4, Kernel = Linear, C = 3, attribute selection technique = Correlation with target)
98.47%	99.2%	99.31%	99.15%	99.46%	99.36%	99.46%	99.29%	98.83%

Adaboost and KNN are clear winner here with Random Forest Lagging behind by 0.1%.

Final Verdict:

It was astonishing to see after doing attribute selections and resampling, performance of Naive Bayes increased significantly. In previous home work without these techniques it was unable to compete with ANN and Decision Trees. It performs very well in precision and recall statistics and these are very important for our use case due to reasons mentioned above. But its Accuracy and ROC Area are significantly lower as compared to others despite making much progress. KNN, on the other hand has very high values for ROC Area and Accuracy and very decent/acceptable values for Precision and Recall as well but, problem with KNN is that it does not build the model. It computes the distance again and again for given test data. Occupany detection is a real time system where we want to get prediction as soon as we get data from the sensors, otherwise it will be useless. So we cannot use KNN for this purpose. Performance of ANN and Adaboost are more or less same they both perform poorly in Recall for Y class statistics which is very dangerous if system is deployed in sensitive areas. Even though Adaboost has high value of accuracy and ROC Area but, it is a weak model as underlying elements are nested and iterative. So, they are not parallel independent model they tend to learn a lot about the training data which may lead to poor generalization. On the other hand Random Forest is the one I would go forward with and deploy it in production because, Random Forest achieve a very fine balance between bias and variance. It learns the underlying complexity by not being too much complex, variable selection makes them independent which leads to better generalization. Its values for precision and recall are also very acceptable and that is why its ROC Area is perfect 100%.