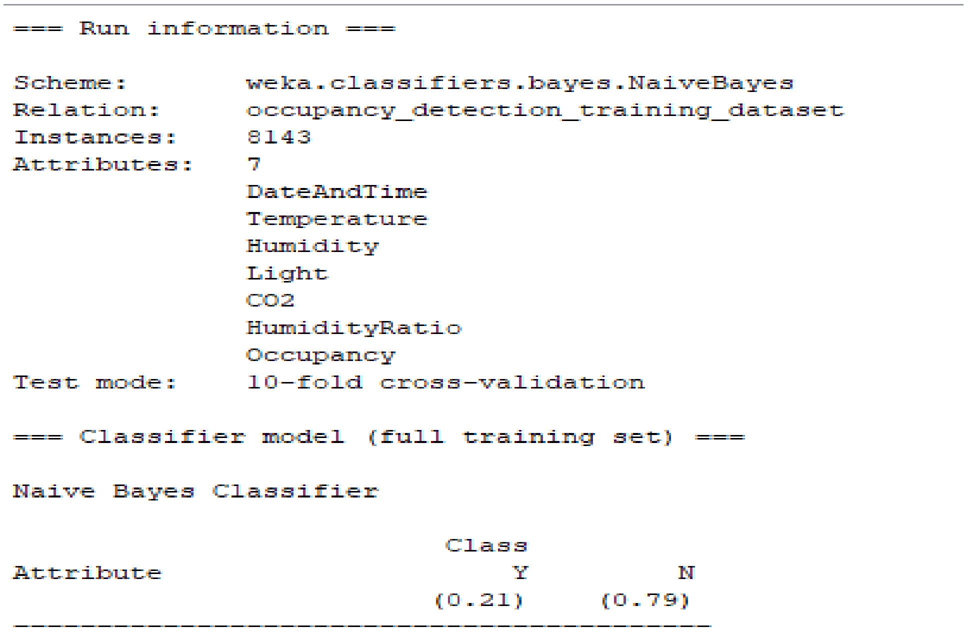
Following is the result given by Weka after running 10-fold cross-validation test using the **Naive Bayes classifier**:

Class Y means Occupancy is there.

Class N means No occupancy.



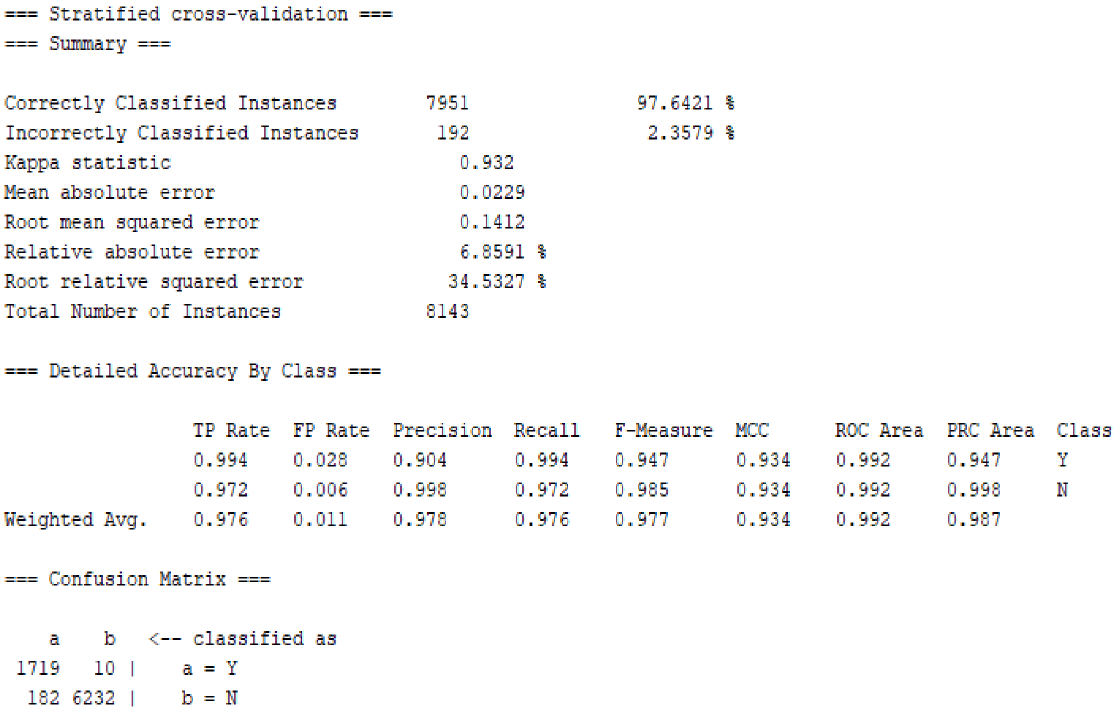
```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     occupancy_detection_training_dataset
Instances:    8143
Attributes:   7
              DateAndTime
              Temperature
              Humidity
              Light
              CO2
              HumidityRatio
              Occupancy
Test mode:    10-fold cross-validation

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

Naive Bayes Classifier

                          Class
Attribute                     Y         N
                         (0.21)    (0.79)
==========================================
```



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

Correctly Classified Instances        7951               97.6421 %
Incorrectly Classified Instances       192                2.3579 %
Kappa statistic                          0.932
Mean absolute error                      0.0229
Root mean squared error                  0.1412
Relative absolute error                  6.8591 %
Root relative squared error             34.5327 %
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    0.028    0.904      0.994    0.947      0.934    0.992     0.947     Y
                 0.972    0.006    0.998      0.972    0.985      0.934    0.992     0.998     N
Weighted Avg.    0.976    0.011    0.978      0.976    0.977      0.934    0.992     0.987

=== Confusion Matrix ===

    a    b   <-- classified as
 1719   10 |    a = Y
  182 6232 |    b = N
```

How good is this result?

This result is perfect for the problem I am trying to solve using this dataset. 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. Accuracy of the Naive Bayes classifier is approaching 98% which will do a good job as amount of energy saved will outweigh the effect of 2.4% of incorrect prediction.

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, precision accuracy of class N is 99.8% which is good.

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, its value is 99.4%.

Would I be satisfied with it in actual use?

Overall I am satisfied, due to reasons mentioned in the answer of previous question where I compared results of accuracy, precision and recall. But, there is some element of doubt in my mind, because of less number of training examples for class Y (Occupancy is there). This doubt does not make me feel confident for deploying this model for actual use. I would love to include more training examples for class Y. So that the dataset becomes much more balanced. If I get the same results after this balancing of the dataset then I will be confident to deploy this model for actual use.

Is the classifier able to predict points with one of the class labels better than the other(s)?

Yes, class Y prediction is better than the prediction of class N. This can be clearly seen from the TP rate values of class N and Y. TP rate of class Y is 0.994 and for class N is 0.972. But, this difference is not huge and reason for this difference could be imbalance dataset. I.e. large number of training examples for N than Y (Y: N 21:79).

Which type of error(s) do you think is more costly for your concept, and does the classifier minimize those errors?

Following are the error metrics that are more costly for my concept:

1) Precision of N class error:

In areas where non-sensitive work is being done like in normal office, I want my model to predict N class when actual value is N class, this is the point where HVAC control algorithm will be used for saving energy. It is no harm in predicting N when actual value was Y. This classifier does minimize those errors. Classifier is able to minimize this error as precision error value for class N is very low i.e.

Error = 1 – Accuracy = 1 = 0.998 = 0.002

2) Recall of Y class:

In sensitive buildings, places like hospitals. I want my model to predict Y class when actual value is Y class, because if N is predicted, lights and other energy consuming appliances will be turned off which will be harmful, especially when doctor is performing surgery or is treating the patient. Classifier does minimize these errors as recall error value for class Y is very low i.e.

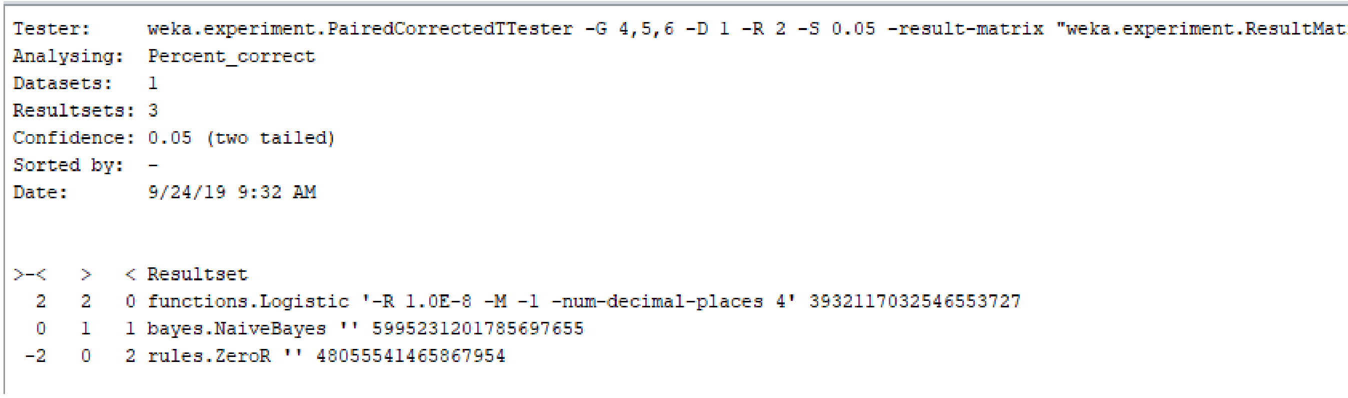
Error = 1 – Accuracy = 1 = 0.96 = 0.06

3) Apparent Error:

It is error on sample used to train the model. This error could mislead us with the results as it will make the model very optimistic in estimating future performance. Naive Bayes classifier does not minimize this error, in fact it hides this error. In order to avoid this error, I want to increase some training examples of Y classes to make this imbalance dataset much more balanced.

Following are different results given by Weka Experimenter:

Ranking:

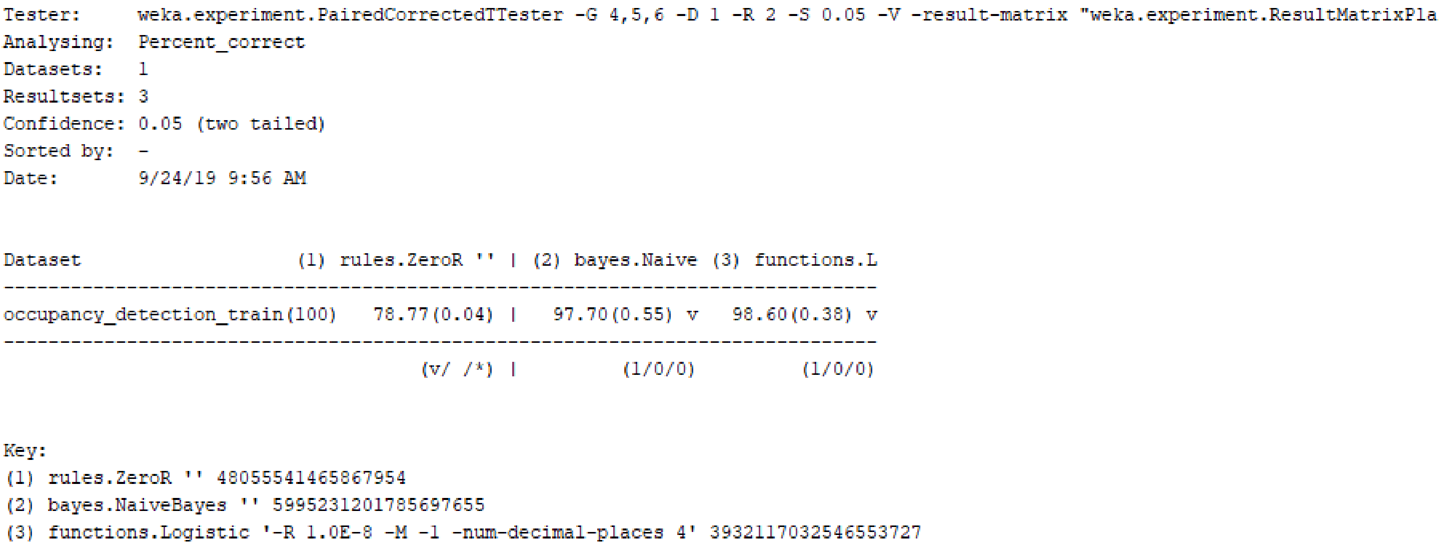


```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMat
Analysing:  Percent_correct
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       9/24/19 9:32 AM


>-<   >   < Resultset
  2   2   0 functions.Logistic '-R 1.0E-8 -M -1 -num-decimal-places 4' 3932117032546553727
  0   1   1 bayes.NaiveBayes '' 5995231201785697655
 -2   0   2 rules.ZeroR '' 48055541465867954
```

Logistic Regression tops the list in Weka Experimenter ranking results. Logistic regression won 2 and lost 0, Naive Bayes won 1 and lost 1, ZeroR lost 2. After seeing these rankings looks like Logistic regression should be the automatic choice? No, we now should compare average accuracy, standard deviation and look for statistical significance. For this comparison let’s make ZeroR as Test base.

ZeroR as Test Base:



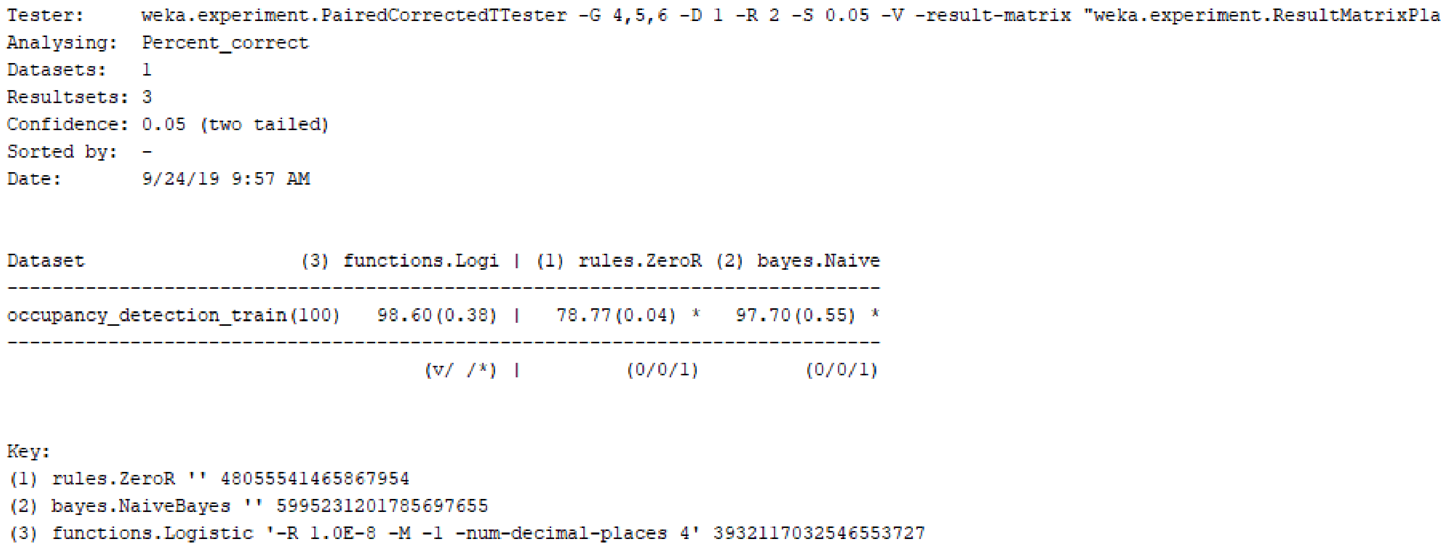
```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -V -result-matrix "weka.experiment.ResultMatrixPla
Analysing:  Percent_correct
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       9/24/19 9:56 AM


Dataset                   (1) rules.ZeroR '' | (2) bayes.Naive (3) functions.L
------------------------------------------------------------------------------
occupancy_detection_train(100)   78.77(0.04) |   97.70(0.55) v   98.60(0.38) v
------------------------------------------------------------------------------
                                     (v/ /*) |         (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
```

As we can see that average accuracy of both Naive Bayes and Logistic Regression is greater than ZeroR and there is ‘v’ attached next to their results, which indicates that difference in the accuracy of their results as compared to ZeroR is statistically significant. So we can say that these two algorithms achieved a statistically significantly better result than the ZeroR baseline. Since, the accuracy for Logistic Regression is higher than the accuracy for Naive Bayes, so next we want to see if the difference between these two accuracy scores is significant. For this purpose we make Logistic Regression a Test base.

Logistic Regression as Test Base:



```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -V -result-matrix "weka.experiment.ResultMatrixPla
Analysing:  Percent_correct
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       9/24/19 9:57 AM


Dataset                   (3) functions.Logi | (1) rules.ZeroR (2) bayes.Naive
------------------------------------------------------------------------------
occupancy_detection_train(100)   98.60(0.38) |   78.77(0.04) *   97.70(0.55) *
------------------------------------------------------------------------------
                                     (v/ /*) |         (0/0/1)         (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
```

In this case accuracy score of both ZeroR and Naive Bayes is lower than that of Logistic Regression and also there is a '*' next to the results of Naive Bayes and Logistic Regression which means that result is statistically different.

Conclusion:

It is clearly evident that Logistic Regression is the best algorithm for this dataset. Because difference in accuracy is statistically different and accuracy of Logistic Regression is the highest. So, we can confidently move forward and use Logistic Regression Algorithm to make predictions.