# IST707 Data Mining: HW7 SVMs, kNN, and Random Forest for Customer Churn

**Name: Akshay Chaurasia** **Email: ahchaura@syr.edu**
**SUID: 309898873** **Date: 10/21/2019**

Now that we have learned about the final three classification algorithms we will study in this class (SVMs, kNN, and Random Forest) let's see which of the 5 classification algorithms we have studied is the best model for the Telecommunications Churn Data Set.

In this homework, you will use SVMs, kNN, and Random Forest algorithms on the Telecommunications Customer Churn data and compare their performance with the naïve Bayes and decision tree models you built for HW6.

Deliverables:

1. Write a report to describe what you did: including the data preparation, transformation, algorithm tuning, and model generation for each model. Describe which model you feel is the best model for this application.
   a. Please review and report on all 4 models evaluation measures (Correctly Classified Instances, Precision, Recall, F-Measure) for each of the 5 models.
   b. Please indicate which evaluation measure you think is best the measure of model accuracy, and why.

**IST707 Data Mining: HW7 SVMs, kNN, and Random Forest for Customer Churn**

**Name: Akshay Chaurasia** **Email: ahchaura@syr.edu**
**SUID: 309898873** **Date: 10/21/2019**

# Solution:

## Data Preparation:

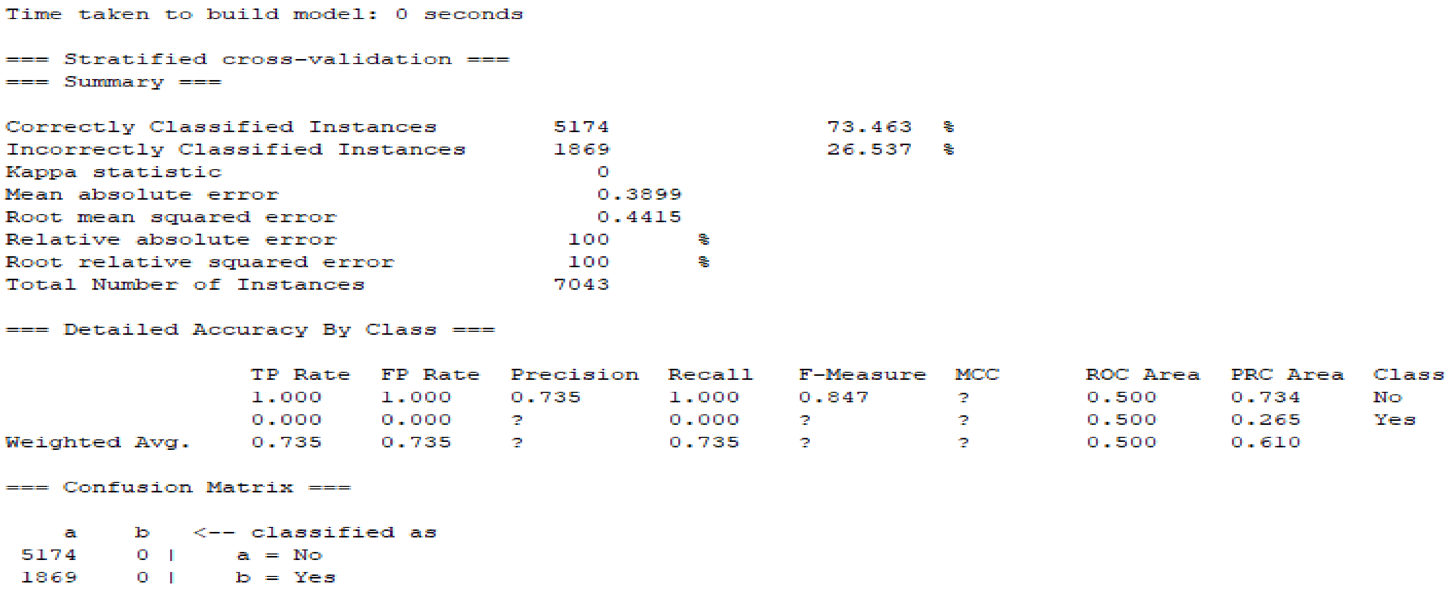
The classification problem over here is to find out which is the best model for the Telecommunications Churn Data Set. This can help identify the customers who are more likely to leave the company in the future and therefore Teleco can target them and provide some special package and make them stay with the Teleco. For this, I have been provided a dataset with 7043 instances and 21 attributes.

The attribute CustomerID can be eliminated from the model as it is insignificant and won't we helpful in prediction and/or decision making. Also, the attributes SeniorCitizen, tenure, MonthlyCharges, TotalCharges are continuous numeric attributes. Therefore, these attributes need to be discretized.

Also, I will be using **10 folds Cross-validation** for all the algorithms.

## 1. Zero Rule Alogorithm

First off, running "Zero Rule Alogorithm" to determine the **baseline accuracy** for the given dataset to get a point of reference which we can use to compare with the other algorithms.

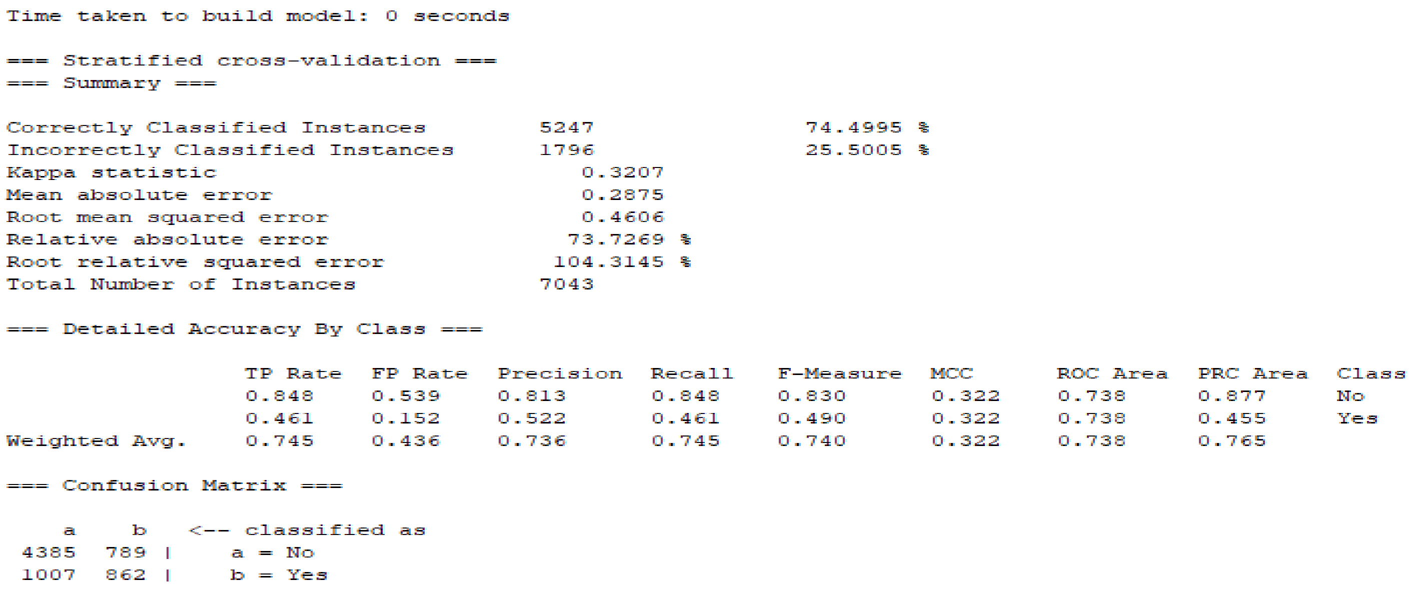


```
Time taken to build model: 0 seconds

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

Correctly Classified Instances        5174               73.463  %
Incorrectly Classified Instances      1869               26.537  %
Kappa statistic                          0
Mean absolute error                      0.3899
Root mean squared error                  0.4415
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.735      1.000    0.847      ?        0.500     0.734     No
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.265     Yes
Weighted Avg.    0.735    0.735    ?          0.735    ?          ?        0.500     0.610

=== Confusion Matrix ===

    a    b   <-- classified as
 5174    0 |    a = No
 1869    0 |    b = Yes
```

Here, we get the accuracy of 73.463% with 5174 number of correctly classified instances.

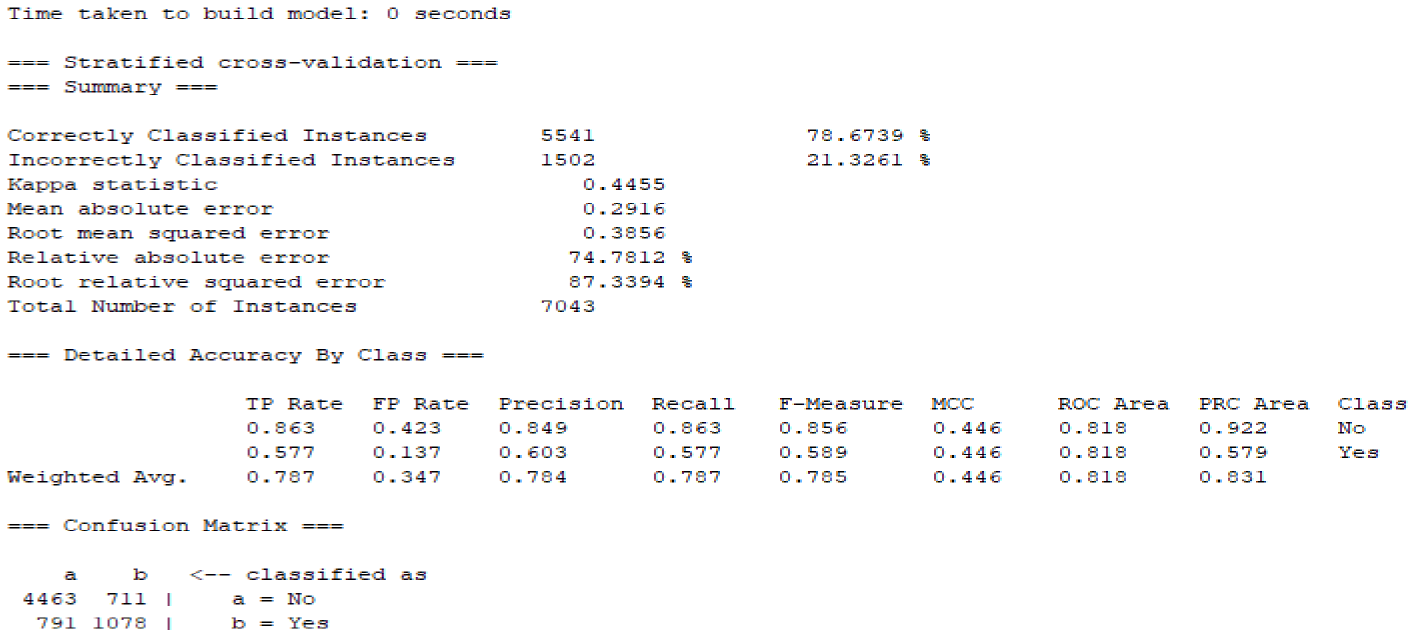
## 2. KNN: IBK

**2.1. For K = 1**, I am getting the accuracy of 74.4995%, which is slightly better than the baseline accuracy of 73.463%. So, I will try to run KNN algorithm on this dataset using some other values of K.



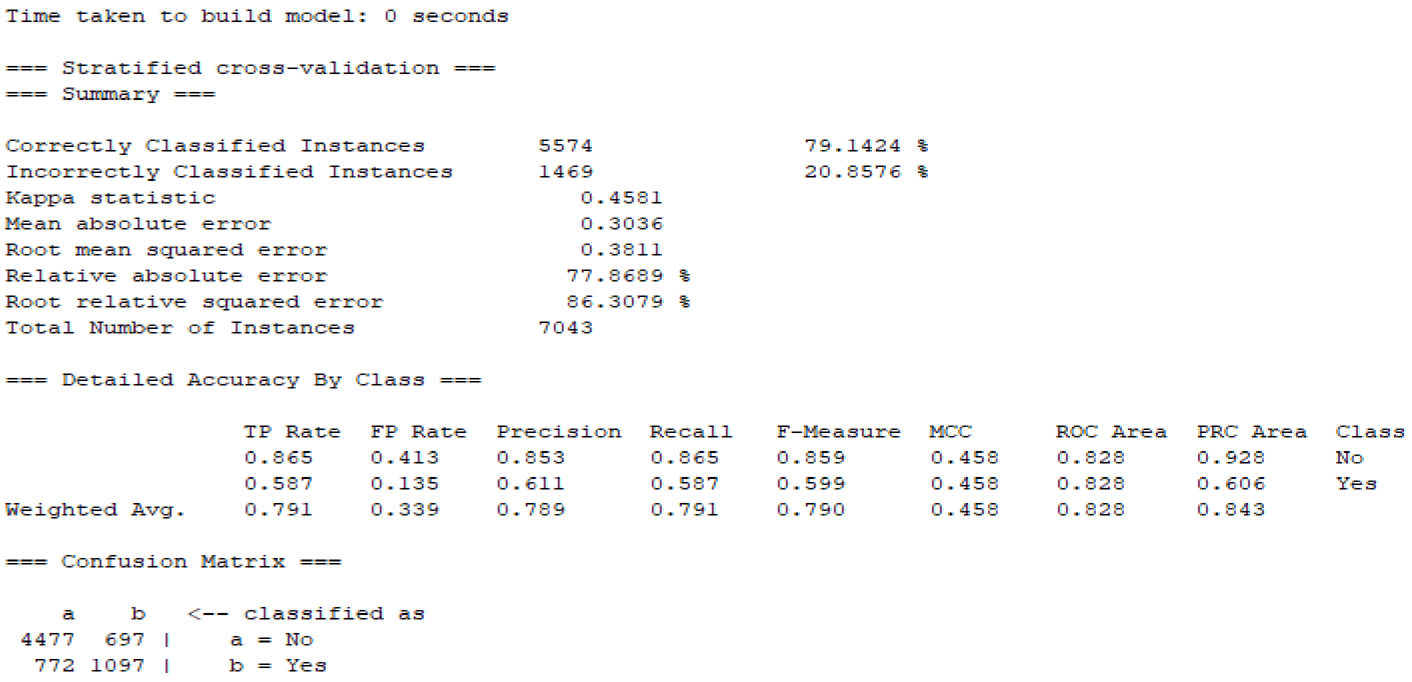
```
Time taken to build model: 0 seconds

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

Correctly Classified Instances        5247               74.4995 %
Incorrectly Classified Instances      1796               25.5005 %
Kappa statistic                          0.3207
Mean absolute error                      0.2875
Root mean squared error                  0.4606
Relative absolute error                 73.7269 %
Root relative squared error            104.3145 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.848    0.539    0.813      0.848    0.830      0.322    0.738     0.877     No
                 0.461    0.152    0.522      0.461    0.490      0.322    0.738     0.455     Yes
Weighted Avg.    0.745    0.436    0.736      0.745    0.740      0.322    0.738     0.765

=== Confusion Matrix ===

    a    b   <-- classified as
 4385  789 |    a = No
 1007  862 |    b = Yes
```

**2.2. For K = 10,** I am getting the accuracy of 78.6739%



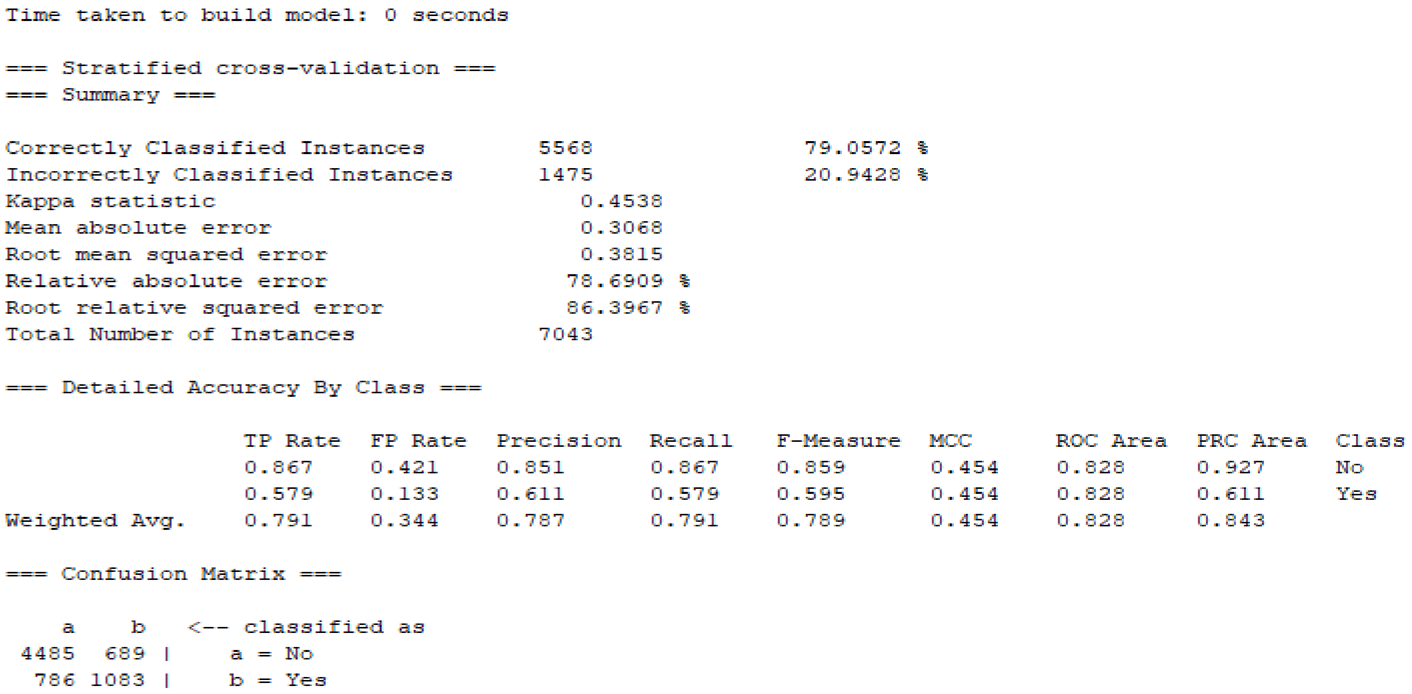
```
Time taken to build model: 0 seconds

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

Correctly Classified Instances        5541               78.6739 %
Incorrectly Classified Instances      1502               21.3261 %
Kappa statistic                          0.4455
Mean absolute error                      0.2916
Root mean squared error                  0.3856
Relative absolute error                 74.7812 %
Root relative squared error             87.3394 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.863    0.423    0.849      0.863    0.856      0.446    0.818     0.922     No
                 0.577    0.137    0.603      0.577    0.589      0.446    0.818     0.579     Yes
Weighted Avg.    0.787    0.347    0.784      0.787    0.785      0.446    0.818     0.831

=== Confusion Matrix ===

    a    b   <-- classified as
 4463  711 |    a = No
  791 1078 |    b = Yes
```

**2.3. For K = 100,** I am getting the accuracy of 79.1424%



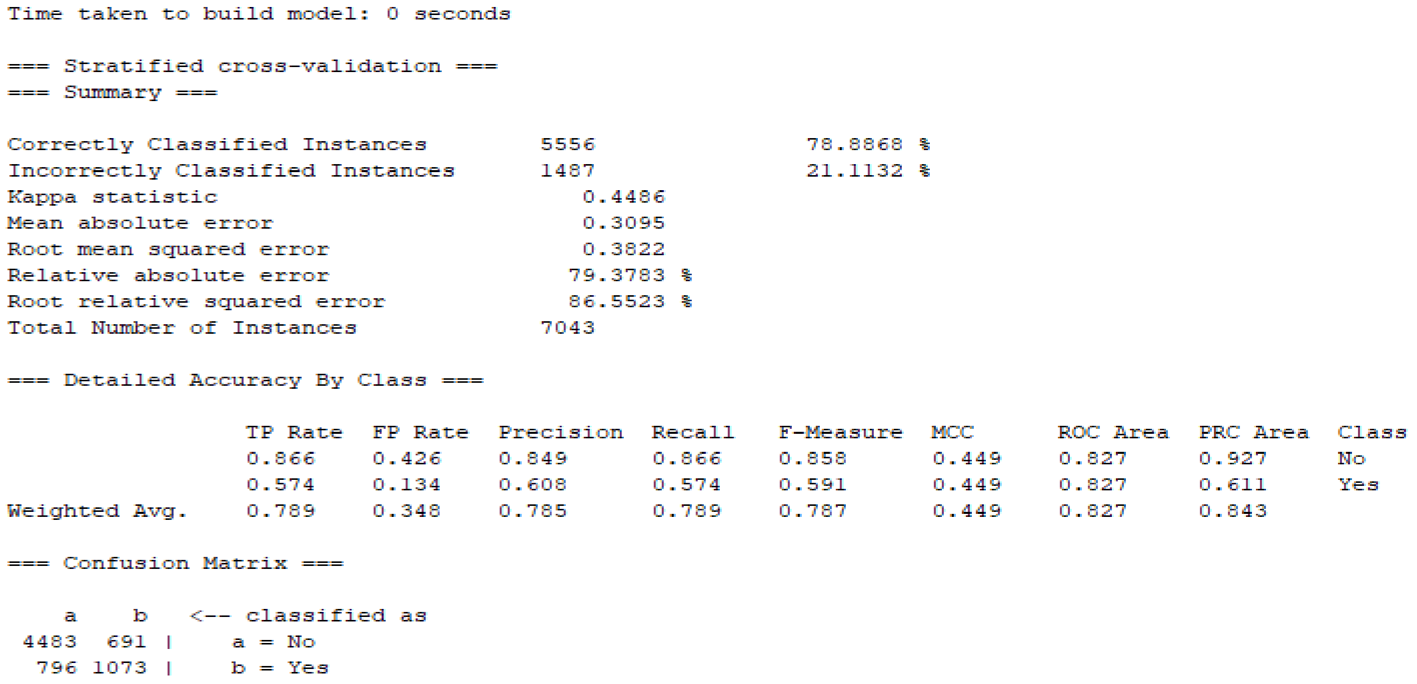
```
Time taken to build model: 0 seconds

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

Correctly Classified Instances        5574               79.1424 %
Incorrectly Classified Instances      1469               20.8576 %
Kappa statistic                          0.4581
Mean absolute error                      0.3036
Root mean squared error                  0.3811
Relative absolute error                 77.8689 %
Root relative squared error             86.3079 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.865    0.413    0.853      0.865    0.859      0.458    0.828     0.928     No
                 0.587    0.135    0.611      0.587    0.599      0.458    0.828     0.606     Yes
Weighted Avg.    0.791    0.339    0.789      0.791    0.790      0.458    0.828     0.843

=== Confusion Matrix ===

    a    b   <-- classified as
 4477  697 |    a = No
  772 1097 |    b = Yes
```

**2.4. For K = 150,** I am getting the accuracy of 79.0572%



```
Time taken to build model: 0 seconds

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

Correctly Classified Instances        5568               79.0572 %
Incorrectly Classified Instances      1475               20.9428 %
Kappa statistic                          0.4538
Mean absolute error                      0.3068
Root mean squared error                  0.3815
Relative absolute error                 78.6909 %
Root relative squared error             86.3967 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.867    0.421    0.851      0.867    0.859      0.454    0.828     0.927     No
                 0.579    0.133    0.611      0.579    0.595      0.454    0.828     0.611     Yes
Weighted Avg.    0.791    0.344    0.787      0.791    0.789      0.454    0.828     0.843

=== Confusion Matrix ===

    a    b   <-- classified as
 4485  689 |    a = No
  786 1083 |    b = Yes
```

**2.5. For K = 200,** I am getting the accuracy of 78.8868%



```
Time taken to build model: 0 seconds

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

Correctly Classified Instances        5556               78.8868 %
Incorrectly Classified Instances      1487               21.1132 %
Kappa statistic                          0.4486
Mean absolute error                      0.3095
Root mean squared error                  0.3822
Relative absolute error                 79.3783 %
Root relative squared error             86.5523 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.866    0.426    0.849      0.866    0.858      0.449    0.827     0.927     No
                 0.574    0.134    0.608      0.574    0.591      0.449    0.827     0.611     Yes
Weighted Avg.    0.789    0.348    0.785      0.789    0.787      0.449    0.827     0.843

=== Confusion Matrix ===

    a    b   <-- classified as
 4483  691 |    a = No
  796 1073 |    b = Yes
```

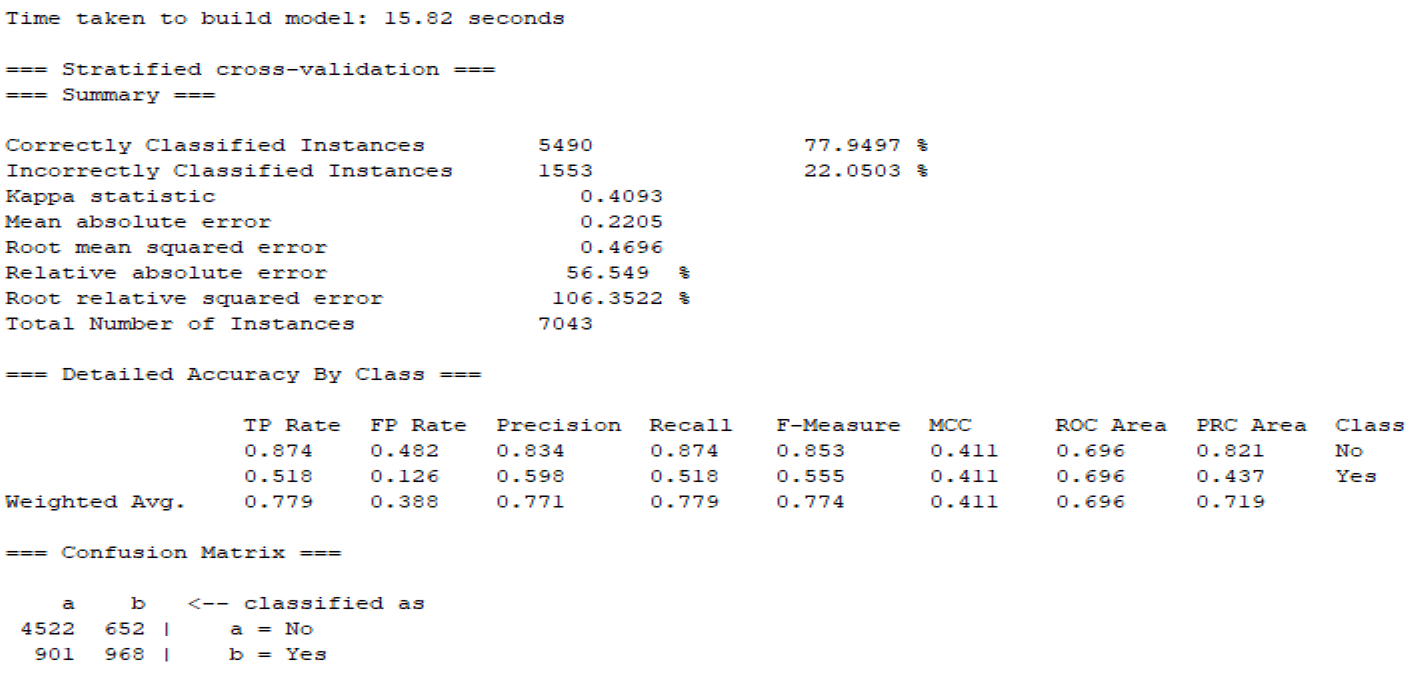
As we can see that for the values of K higher than 100, the accuracy starts decreasing.

| **K** | 1 | 10 | 100 | 150 | 200 |
|---|---|---|---|---|---|
| **Accuracy (%)** | 74.4995 | 78.6739 | 79.1424 | 79.0572 | 78.8868 |

Therefore, I will consider the KNN algorithm with the **K-value of 100** for the comparison with other algorithms.

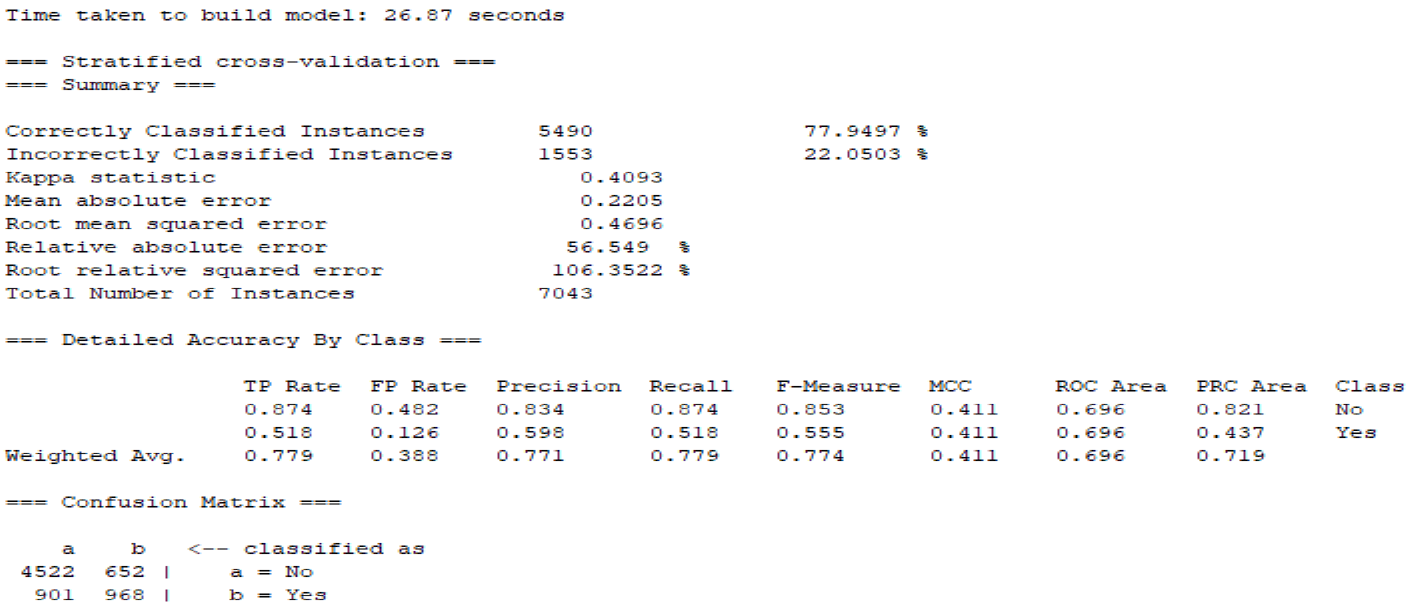
## 3. SVM: SMO

**3.1. For C = 0.5,** I am getting the accuracy of 77.9497%



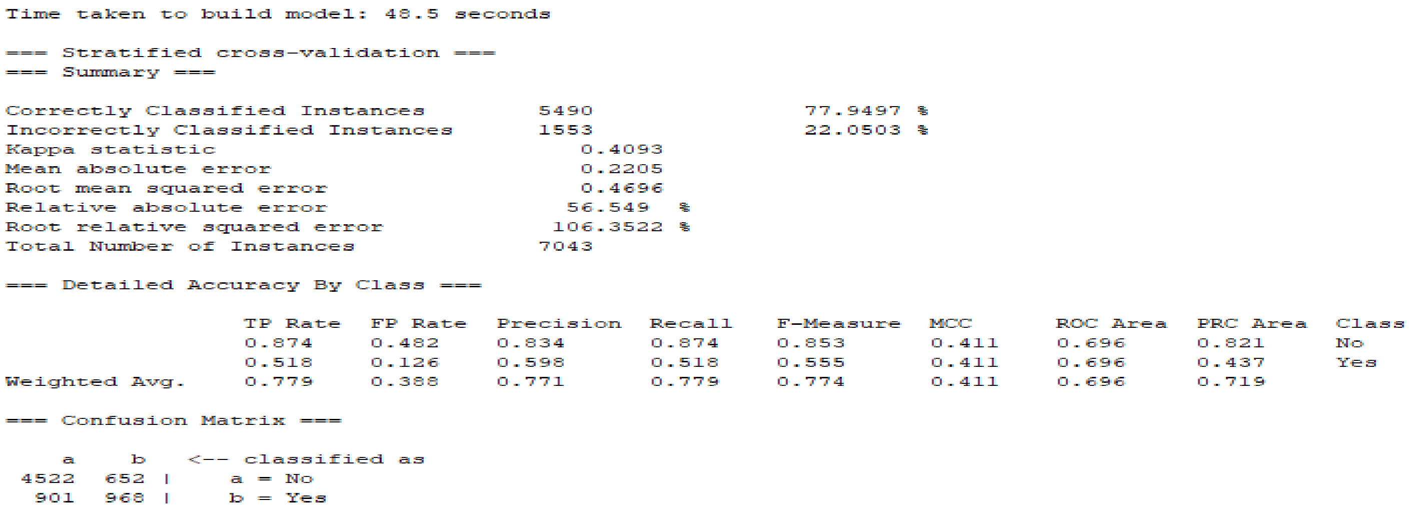
```
Time taken to build model: 15.82 seconds

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

Correctly Classified Instances        5490               77.9497 %
Incorrectly Classified Instances      1553               22.0503 %
Kappa statistic                          0.4093
Mean absolute error                      0.2205
Root mean squared error                  0.4696
Relative absolute error                 56.549  %
Root relative squared error            106.3522 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.874    0.482    0.834      0.874    0.853      0.411    0.696     0.821     No
                 0.518    0.126    0.598      0.518    0.555      0.411    0.696     0.437     Yes
Weighted Avg.    0.779    0.388    0.771      0.779    0.774      0.411    0.696     0.719

=== Confusion Matrix ===

    a    b   <-- classified as
 4522  652 |    a = No
  901  968 |    b = Yes
```

**3.2. C = 1,** I am getting the accuracy of 77.9497%



```
Time taken to build model: 26.87 seconds

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

Correctly Classified Instances        5490               77.9497 %
Incorrectly Classified Instances      1553               22.0503 %
Kappa statistic                          0.4093
Mean absolute error                      0.2205
Root mean squared error                  0.4696
Relative absolute error                 56.549  %
Root relative squared error            106.3522 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.874    0.482    0.834      0.874    0.853      0.411    0.696     0.821     No
                 0.518    0.126    0.598      0.518    0.555      0.411    0.696     0.437     Yes
Weighted Avg.    0.779    0.388    0.771      0.779    0.774      0.411    0.696     0.719

=== Confusion Matrix ===

    a    b   <-- classified as
 4522  652 |    a = No
  901  968 |    b = Yes
```

**3.2.C = 2,** also I am getting the same accuracy

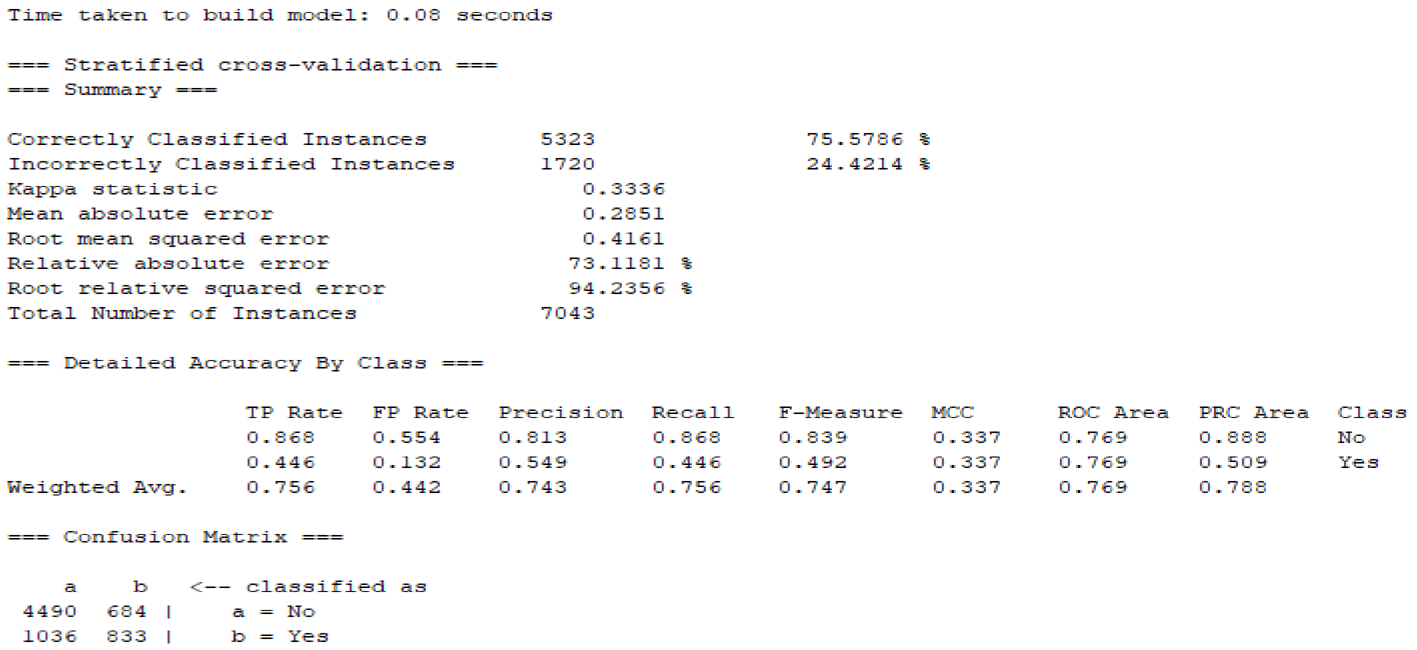


```
Time taken to build model: 48.5 seconds

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

Correctly Classified Instances        5490               77.9497 %
Incorrectly Classified Instances      1553               22.0503 %
Kappa statistic                          0.4093
Mean absolute error                      0.2205
Root mean squared error                  0.4696
Relative absolute error                 56.549  %
Root relative squared error            106.3522 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.874    0.482    0.834      0.874    0.853      0.411    0.696     0.821     No
                 0.518    0.126    0.598      0.518    0.555      0.411    0.696     0.437     Yes
Weighted Avg.    0.779    0.388    0.771      0.779    0.774      0.411    0.696     0.719

=== Confusion Matrix ===

    a    b   <-- classified as
 4522  652 |    a = No
  901  968 |    b = Yes
```

Therefore, I will be considering **"3.2 C=1"**

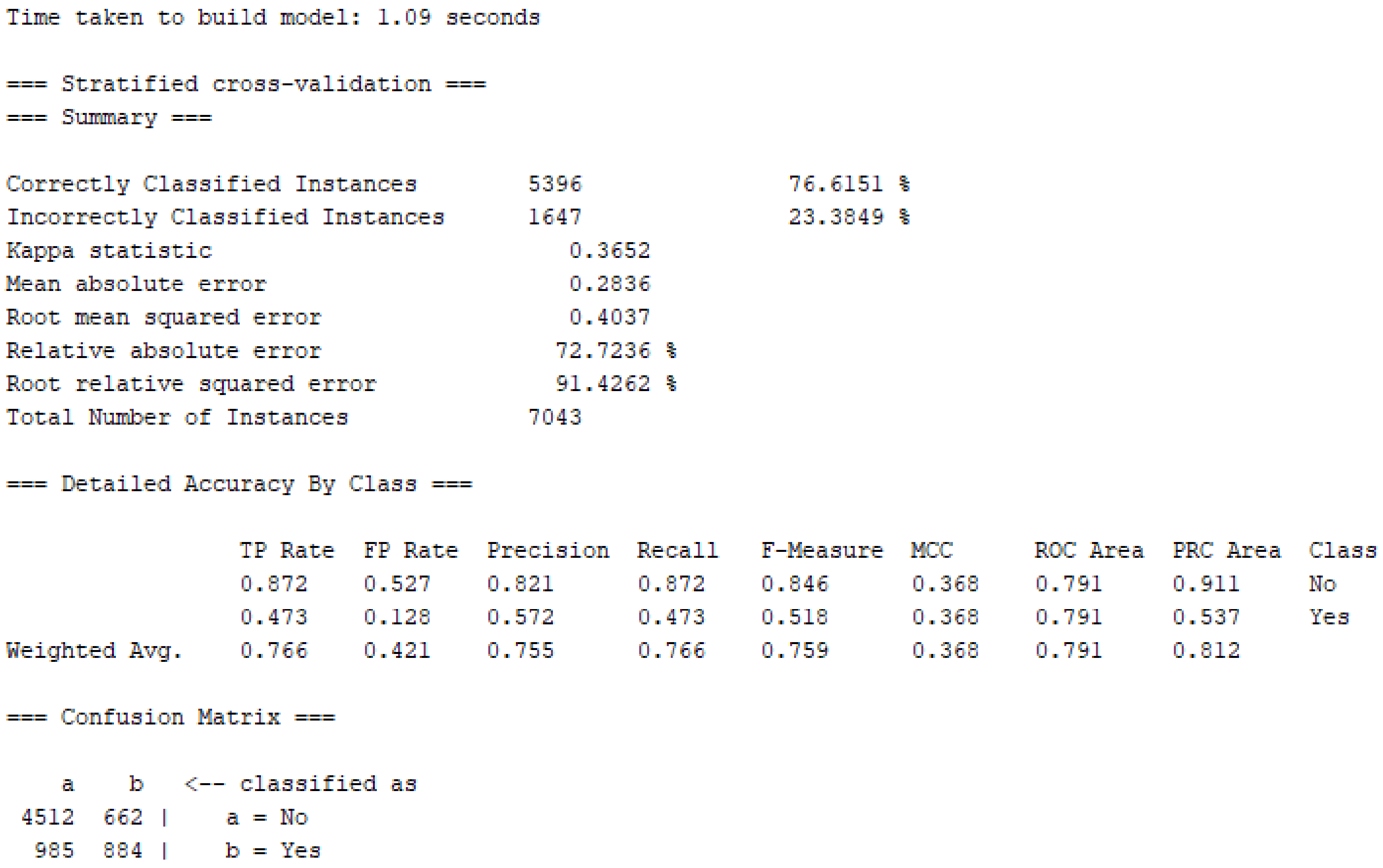
## 4. Random Forest

**4.1. For Num of iterations = 10,** I am getting the accuracy of 75.5786%



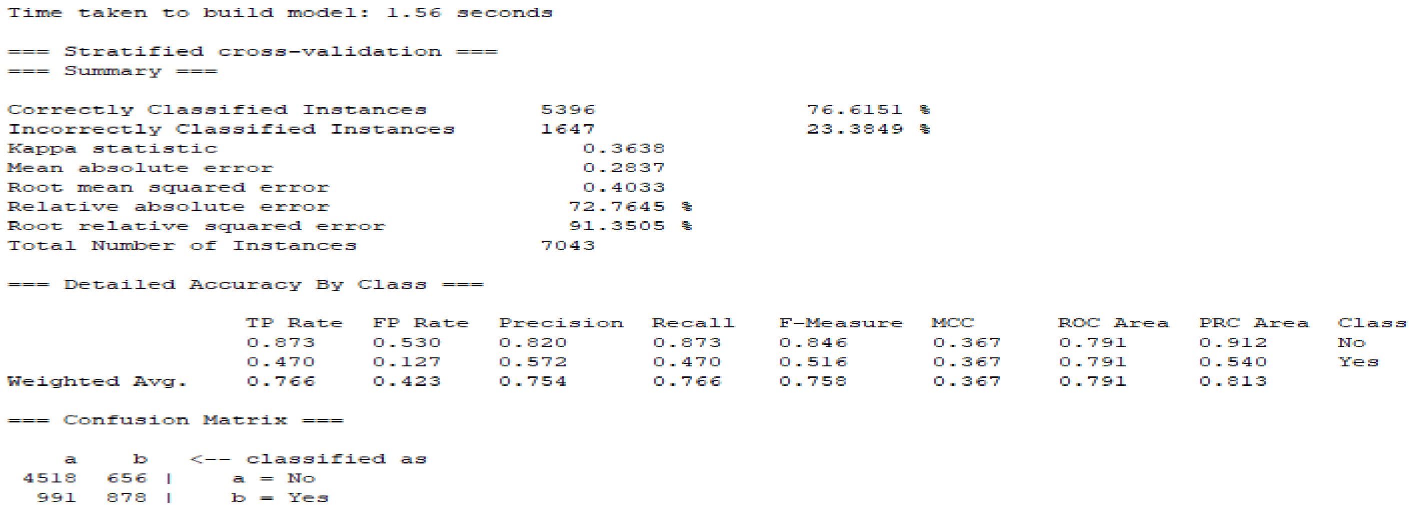
```
Time taken to build model: 0.08 seconds

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

Correctly Classified Instances        5323               75.5786 %
Incorrectly Classified Instances      1720               24.4214 %
Kappa statistic                          0.3336
Mean absolute error                      0.2851
Root mean squared error                  0.4161
Relative absolute error                 73.1181 %
Root relative squared error             94.2356 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.868    0.554    0.813      0.868    0.839      0.337    0.769     0.888     No
                 0.446    0.132    0.549      0.446    0.492      0.337    0.769     0.509     Yes
Weighted Avg.    0.756    0.442    0.743      0.756    0.747      0.337    0.769     0.788

=== Confusion Matrix ===

    a    b   <-- classified as
 4490  684 |    a = No
 1036  833 |    b = Yes
```

**4.2. For Num of iterations = 100,** I am getting the accuracy of 76.6151%



```
Time taken to build model: 1.09 seconds

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

Correctly Classified Instances        5396               76.6151 %
Incorrectly Classified Instances      1647               23.3849 %
Kappa statistic                          0.3652
Mean absolute error                      0.2836
Root mean squared error                  0.4037
Relative absolute error                 72.7236 %
Root relative squared error             91.4262 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.872    0.527    0.821      0.872    0.846      0.368    0.791     0.911     No
                 0.473    0.128    0.572      0.473    0.518      0.368    0.791     0.537     Yes
Weighted Avg.    0.766    0.421    0.755      0.766    0.759      0.368    0.791     0.812

=== Confusion Matrix ===

    a    b   <-- classified as
 4512  662 |    a = No
  985  884 |    b = Yes
```

**4.3. For Number of iterations = 200,** I am getting the accuracy of 76.6151%



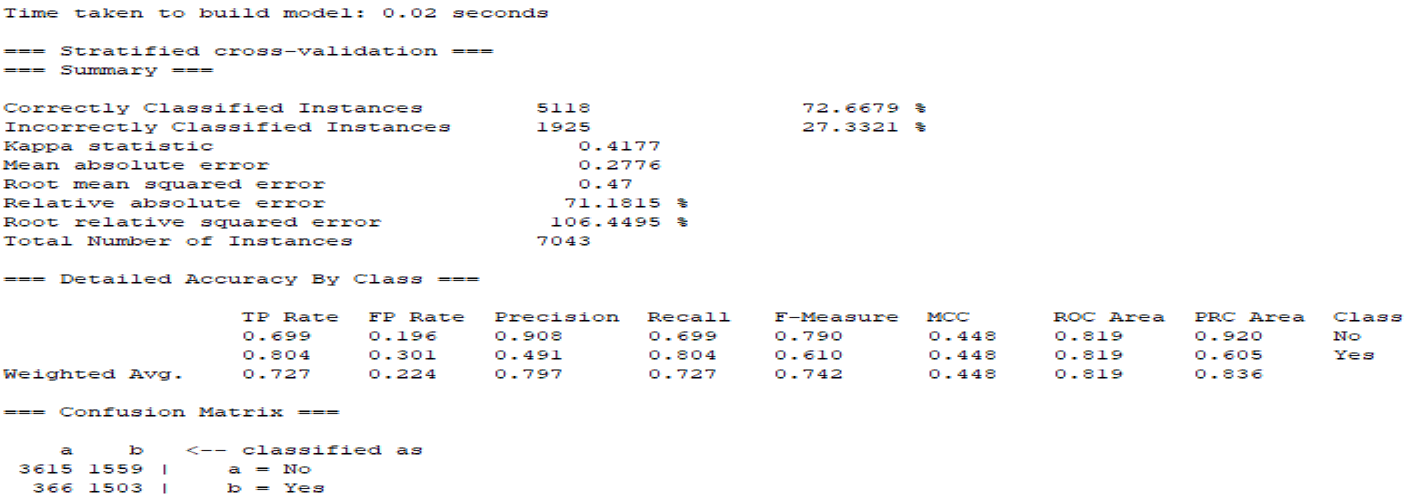
```
Time taken to build model: 1.56 seconds

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

Correctly Classified Instances        5396               76.6151 %
Incorrectly Classified Instances      1647               23.3849 %
Kappa statistic                          0.3638
Mean absolute error                      0.2837
Root mean squared error                  0.4033
Relative absolute error                 72.7645 %
Root relative squared error             91.3505 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.873    0.530    0.820      0.873    0.846      0.367    0.791     0.912     No
                 0.470    0.127    0.572      0.470    0.516      0.367    0.791     0.540     Yes
Weighted Avg.    0.766    0.423    0.754      0.766    0.758      0.367    0.791     0.813

=== Confusion Matrix ===

    a    b   <-- classified as
 4518  656 |    a = No
  991  878 |    b = Yes
```

Therefore, I will be considering the **"4.2 Number of Iterations = 100"**

**To compare these kNN, SVM and Random Forest with Naïve Bayes and Decision Tree, I will use the result of my last assignment (HW6), where I evaluated Naïve Bayes and Decision Tree.**

## 5. Naïve Bayes:

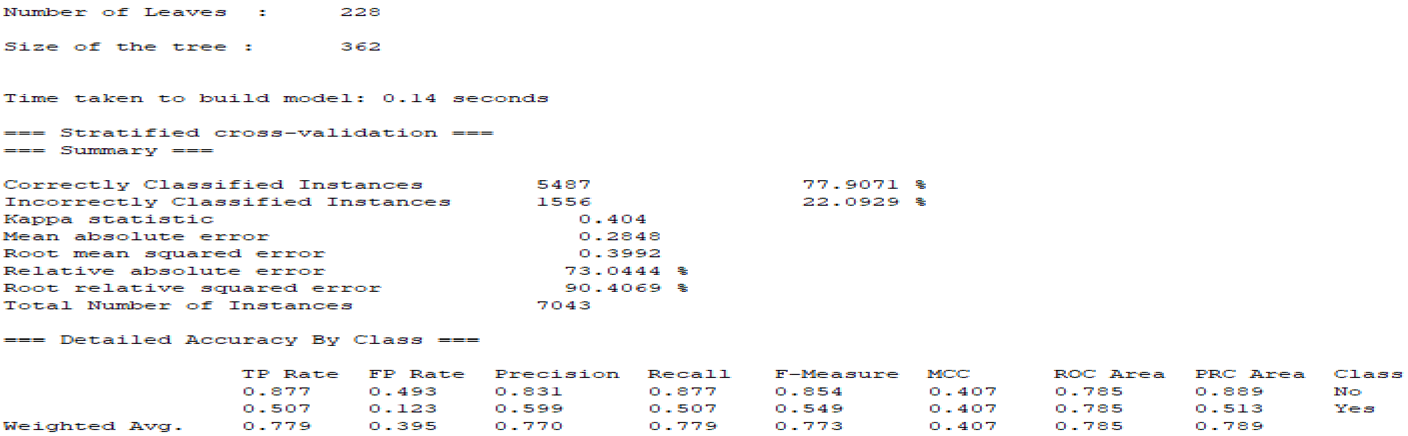
After running the Naïve Bayes algorithm on the given dataset with Cross-validation Folds value of 3, I am getting the Accuracy rate of 72.6679% and Error rate of 27.3321%



```
Time taken to build model: 0.02 seconds

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

Correctly Classified Instances        5118               72.6679 %
Incorrectly Classified Instances      1925               27.3321 %
Kappa statistic                          0.4177
Mean absolute error                      0.2776
Root mean squared error                  0.47
Relative absolute error                 71.1815 %
Root relative squared error            106.4495 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.699    0.196    0.908      0.699    0.790      0.448    0.819     0.920     No
                 0.804    0.301    0.491      0.804    0.610      0.448    0.819     0.605     Yes
Weighted Avg.    0.727    0.224    0.797      0.727    0.742      0.448    0.819     0.836

=== Confusion Matrix ===

    a    b   <-- classified as
 3615 1559 |    a = No
  366 1503 |    b = Yes
```

## 6. Decision Tree

Here, we can see that the number of leaves is 228 and the tree size is 362. Also, the accuracy rate is 77.9071% whereas error rate is only 22.0929%. The tree structure is also pretty symmetrical which I could not get in any other Decision tree model that I created by changing the parameter values.



```
Number of Leaves  :     228

Size of the tree :      362


Time taken to build model: 0.14 seconds

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

Correctly Classified Instances        5487               77.9071 %
Incorrectly Classified Instances      1556               22.0929 %
Kappa statistic                          0.404
Mean absolute error                      0.2848
Root mean squared error                  0.3992
Relative absolute error                 73.0444 %
Root relative squared error             90.4069 %
Total Number of Instances             7043

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.877    0.493    0.831      0.877    0.854      0.407    0.785     0.889     No
                 0.507    0.123    0.599      0.507    0.549      0.407    0.785     0.513     Yes
Weighted Avg.    0.779    0.395    0.770      0.779    0.773      0.407    0.785     0.789
```

a. Please review and report on all 4 models evaluation measures (Correctly Classified Instances, Precision, Recall, F-Measure) for each of the 5 models.

| Model | Accuracy (%) | Correctly Classified Instances | Precision (%) | Recall (%) | F-Measure (%) |
|---|---|---|---|---|---|
| **Naïve Bayes** | 77.9071 | 5118 | 77.0 | 77.9 | 77.3 |
| **Decision Tree** | 72.6679 | 5487 | 79.7 | 72.7 | 74.2 |
| **kNN** | 79.1424 | 5574 | 78.9 | 79.1 | 79 |
| **SVM** | 77.9497 | 5490 | 77.1 | 77.9 | 77.4 |
| **Random Forest** | 76.5725 | 5393 | 75.4 | 76.6 | 75.8 |

b. Please indicate which evaluation measure you think is best the measure of model accuracy, and why.

After comparing the 5 models mentioned above, we can see that kNN isway better than the other 4 models. It has the highest **Accuracy** (79.1424 %), with the most number of **correctly classified instances** (5574). It also has the best **Recall** (79.1%) and best **F-Measure** (79%). Also, kNN's **ROC Area value** (82.8%) is higher than the other algorithms. Thus, I would say that **kNN is the best classification algorithm for the given dataset**.

Accuracy alone cannot be trusted to select a well-performing model due to Accuracy Paradox. It can be misleading. Sometimes it may be desirable to select a model with a lower accuracy because it has a greater predictive power on the problem. In my opinion, **F1 score** which is nothing but harmonic mean of precision and recall, should be given more impotance as these metrics also take false negatives and false positives into account.