IST707 HW6: Naïve Bayes and Decision Tree Comparative Analysis

Name: Akshay Chaurasia Email: ahchaura@syr.edu
SUID: 309898873 Date: 10/14/2019

Now that we have learned two classification algorithms, decision tree and naïve Bayes, let's think further on the question of choosing algorithms for a specific task. Note that there is no silver bullet in terms of algorithm comparison – no algorithm would outperform all other algorithms on all data sets. Therefore, choosing appropriate algorithms is an important decision, and it requires knowledge of both the data set and the candidate algorithms.

Task description:

A telecommunications company is concerned about the number of customers leaving their landline business for cable competitors. They need to understand who is leaving. Imagine that you are an analyst at this company and you have to find out who is leaving and why.

The data set includes information about:

- Customers who left within the last month – the column is called Churn
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies
- Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges
- Demographic info about customers – gender, age range, and if they have partners and dependents

Report structure:

Section 1: Introduction

Briefly describe the classification problem and general data preprocessing. Note that some data preprocessing steps maybe specific to a particular algorithm. Report those steps under each algorithm section.

Section 2: Decision tree

Build a decision tree model. Tune the parameters, such as the pruning options, and report the 3-fold CV accuracy.

Section 3: Naïve Bayes

Build a naïve Bayes model. Tune the parameters, such as the discretization options, to compare results.

Section 4: Algorithm performance comparison

Compare the results from the two algorithms. Which one reached higher accuracy? Which one runs faster? Can you explain why?

Section 5: Model Evaluation

Referencing our learnings on Model Evaluation (ie:Precision,Recall,F-Measure), which accuracy measure do you believe would be best for the Telco company to use and why? Does the use of an alternative evaluation measure effect your decision on which model you would choose? Why or why not?

Solution:

1. Introduction:

The classification problem over here is to find out which Teleco customers are leaving and what is the reason behind it. 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.

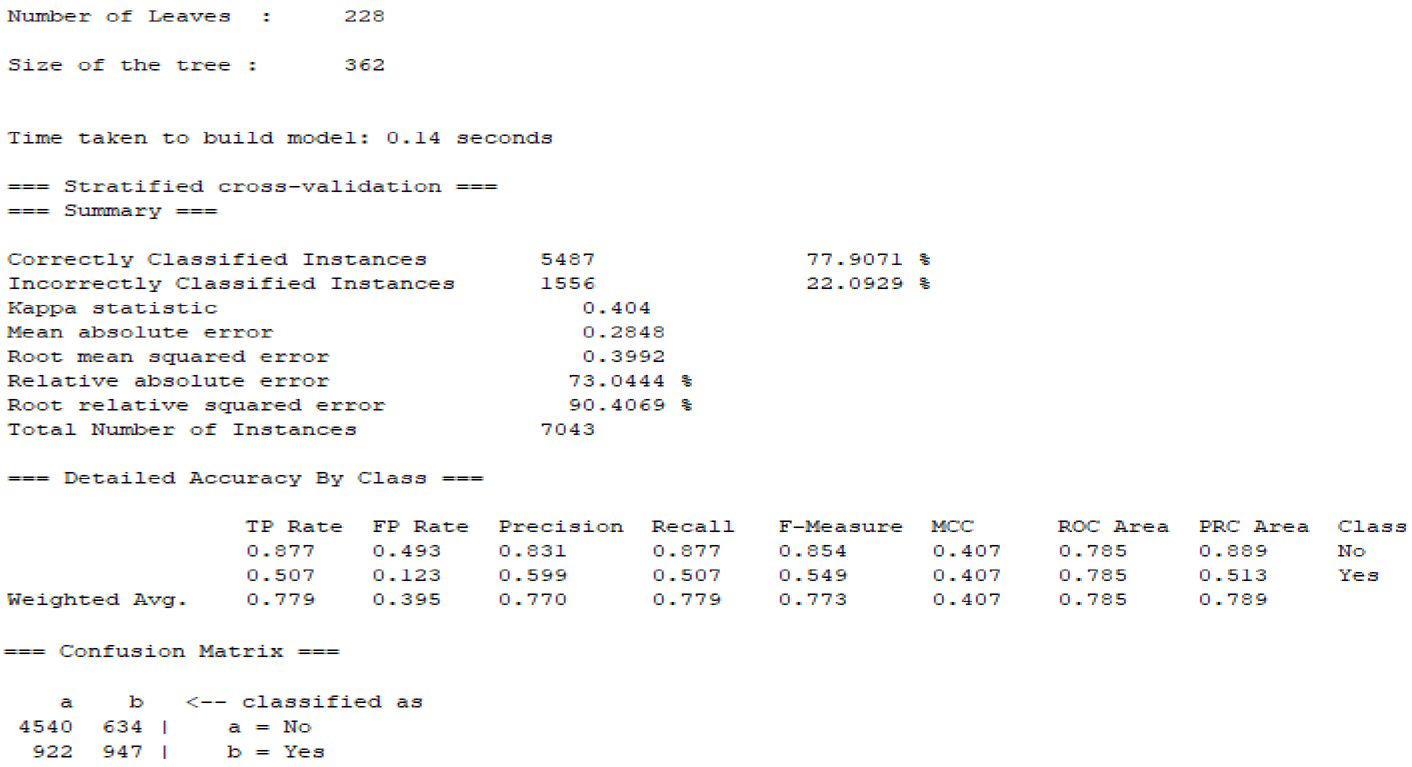
The attribute CustomerID can be eliminated from the model as it is insignificant and won't we helpful in prediction and/or decision making. Also, the attributes SeniorCitizen, tenure, MonthlyCharges, TotalCharges are continuous numeric attributes. Therefore, these attributes need to be discretized.

2. Decision Tree

After constructing decision tree using J48 Algorithm and trying different parameter values I found that the following settings outputs the best result:

1. confidenceFactor = 0.25,
2. unpruned = False and
3. Cross-validation Folds = 3

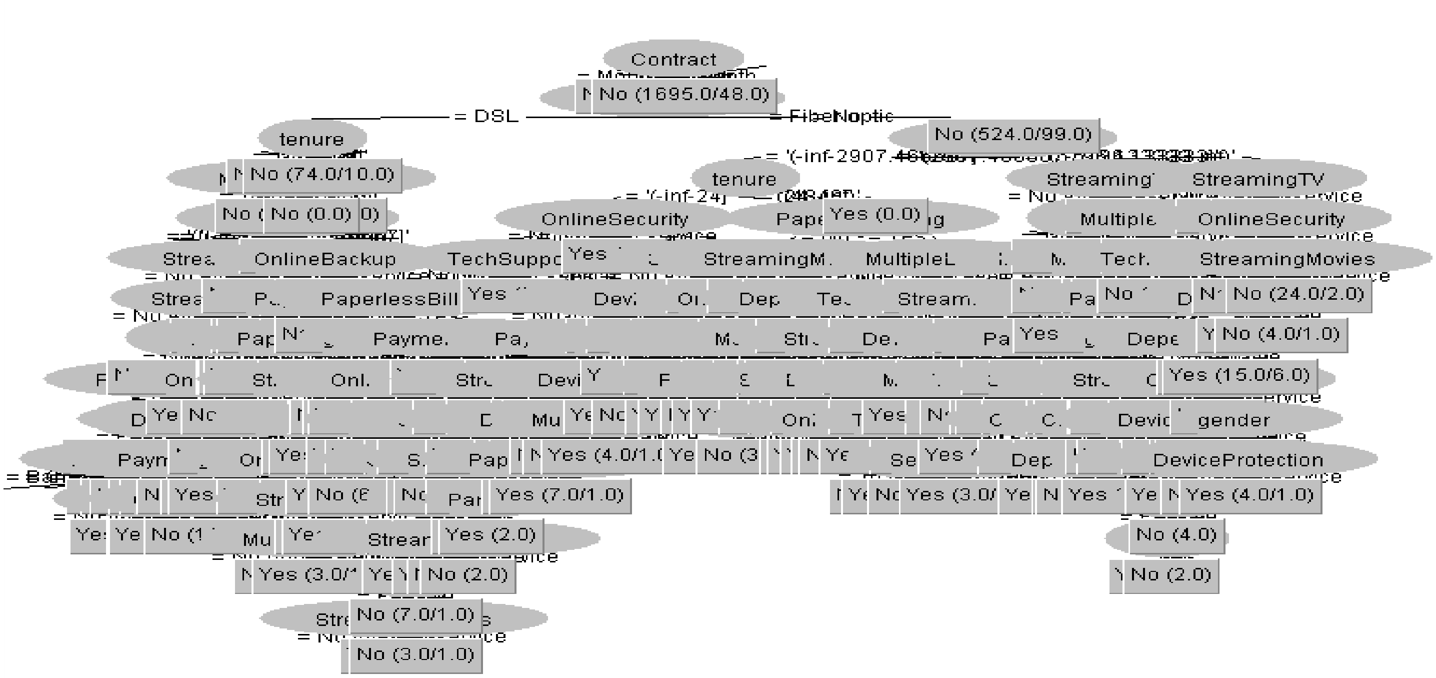
Here, we can see that the number of leaves is 228 and the tree size is 362. Also, the accuracy rate is 77.9071% whereas error rate is only 22.0929%. The tree structure is also pretty symmetrical which I could not get in any other Decision tree model that I created by changing the parameter values.



```
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

=== Confusion Matrix ===

    a    b   <-- classified as
 4540  634 |    a = No
  922  947 |    b = Yes
```

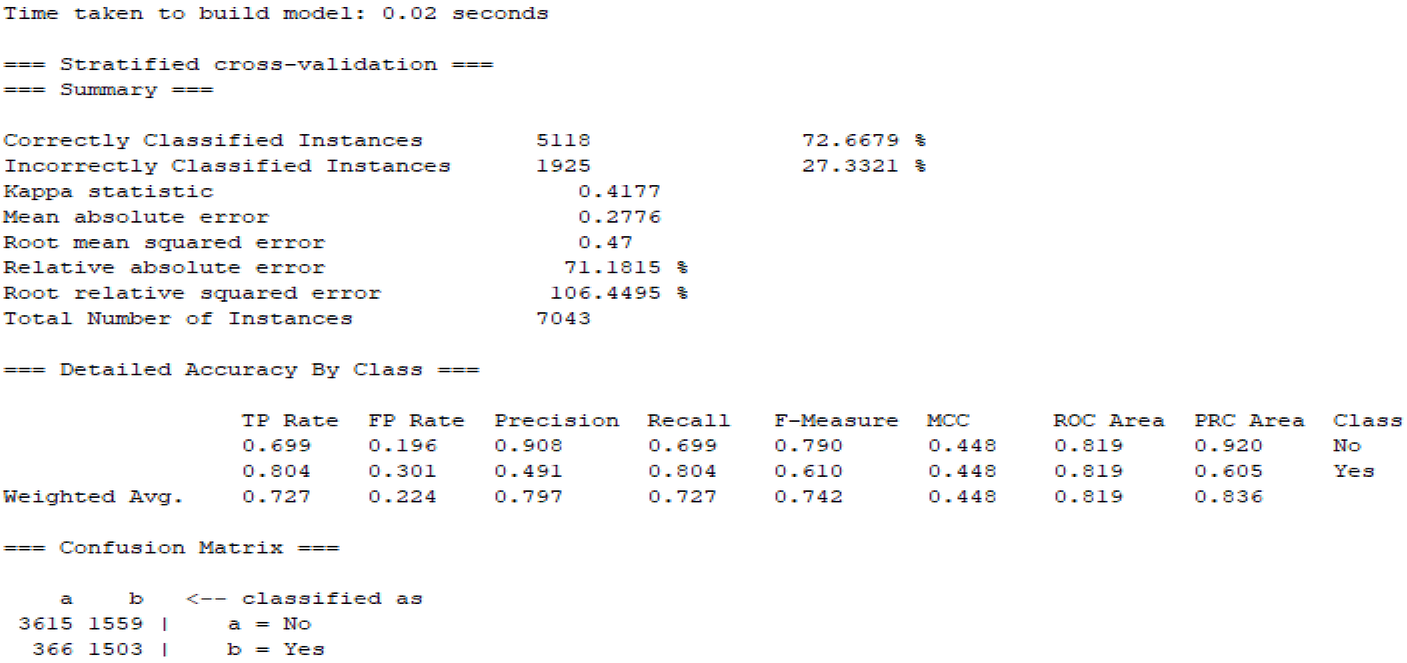
IST707 HW6: Naïve Bayes and Decision Tree Comparative Analysis

Name: Akshay Chaurasia Email: ahchaura@syr.edu
SUID: 309898873 Date: 10/14/2019



3. 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
```

4. Algorithm performance comparison

According to the output that I got after applying the J48 and Naïve Bayes on the given dataset, Decision Tree is a better approach for this dataset because it is giving higher accuracy rate **(77.9071%)** and lower error rate **(22.0929%)**. On the other hand, the accuracy rate of Naïve Bayes is comparatively lower **(72.6679%)**, and the error rate is higher **(27.3321%)**. Let us compare both the algorithms based on Weighted Avg value of Accuracy, Precision, Recall, F-measure and ROC-Area.

Algorithm	Run Time (sec)	Accuracy (%)	Error (%)	Precision (%)	Recall (%)	F-measure (%)
Decision Tree	0.14	77.9071	22.0929	77.0	77.9	77.3
Naïve Bayes	0.02	72.6679	27.3321	79.7	72.7	74.2

From the table above we can see that, the decision Tree has higher accuracy whereas Naïve Bayes has comparatively lower accuracy. On the other hand, Naïve Bayes took **only 0.02 seconds** to analyze the dataset, but the Decision Tree took **0.14 seconds**. Naive Bayes is fast because all it needs are the prior probability values that do not change and can be stored ahead of time. The same probability values are reused in while calculating the posterior. Decision Tree outperforms Naïve Bayes on all parameters but precision **(77.0% vs 79.7%)**.

5. Model Evaluation

Decision Trees are very flexible, easy to understand, and easy to debug. They will work with classification problems and regression problems. Naive Bayes requires you build a classification by hand. There's no way to just toss a bunch of tabular data at it and have it pick the best features it will use to classify.

Decision Trees are very flexible, easy to understand, and easy to debug. It does not require any preprocessing or transformation of features. Even though it is prone to overfitting, you can user pruning or Random forests to avoid that. Naive Bayes work well with small dataset compared to DT which need more data. It is less prone to overfitting and is faster in processing. As such there’s no better classifier, it depends upon problem to problem.