What’s In a Name?

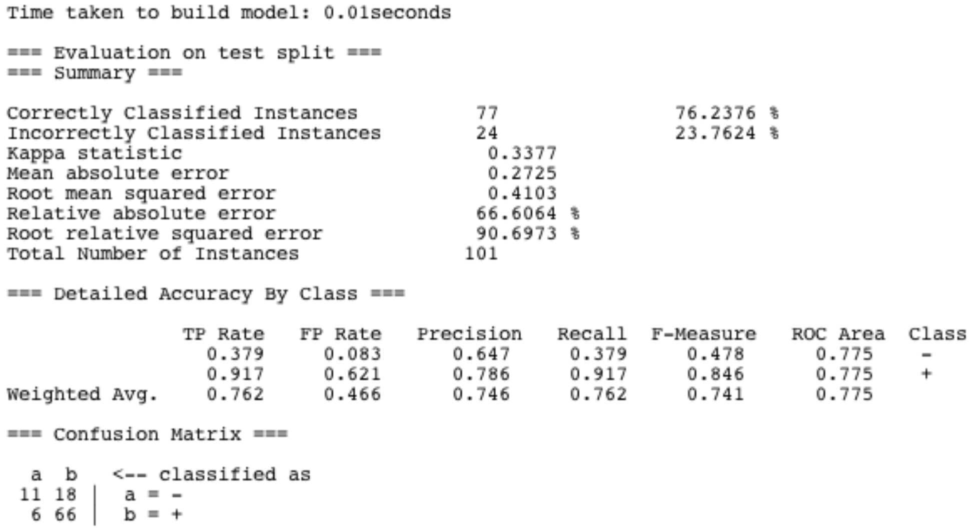
Joe Zeimen

My approach to solving the problem of determining if someone is a winner or loser simply based off of their name followed the following steps.

First I did a quick analysis of the data in a spreadsheet program. I calculated that about 2/3 of the people were winners while only 1/3 were losers. I then separated the data out by sorting on the winning and loosing label to see what patterns might arise that I could visually see. There did not seem to be a discernable pattern. The data was more hidden than that.

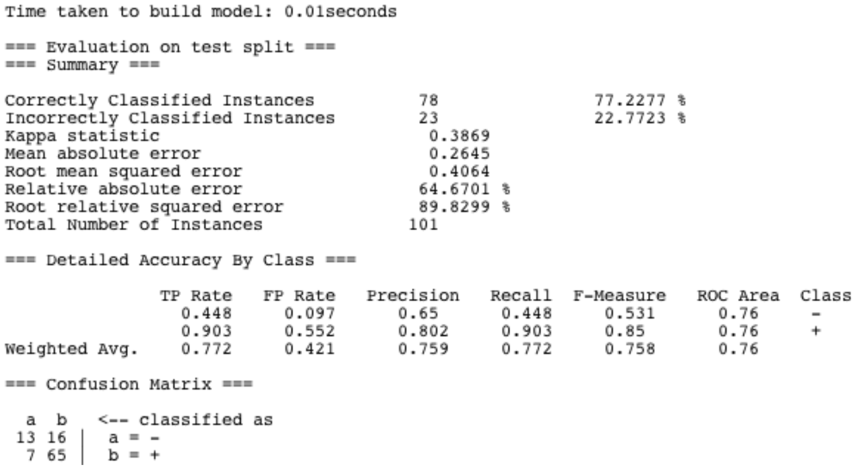
Next I tried sorting the data based on the name. This showed some patterns in the data that started to lead me closer to a solution. I noticed that some of the names that started with certain letters seemed more likely to be almost all losers or almost all winners. I figured this could be one of the possible characteristics to make a classification on. I also noticed something really interesting in that all Thomases were losers while all Toms were winners. This could be name length or how it relates to the last name or some other unknown attribute.

After the initial exploration of the data, I wrote a script that would take in names and extract features from each of the names. One of the first features I extracted was the first letter of the first name followed by number of letters in the names. This script produced a CSV file, which is one of the file formats that Weka can read. I decided to use the J48 algorithm because Weka is able to visually show a tree of the data. Upon running this initial test it is obvious that this does not give Weka enough information to classify with great accuracy the data.



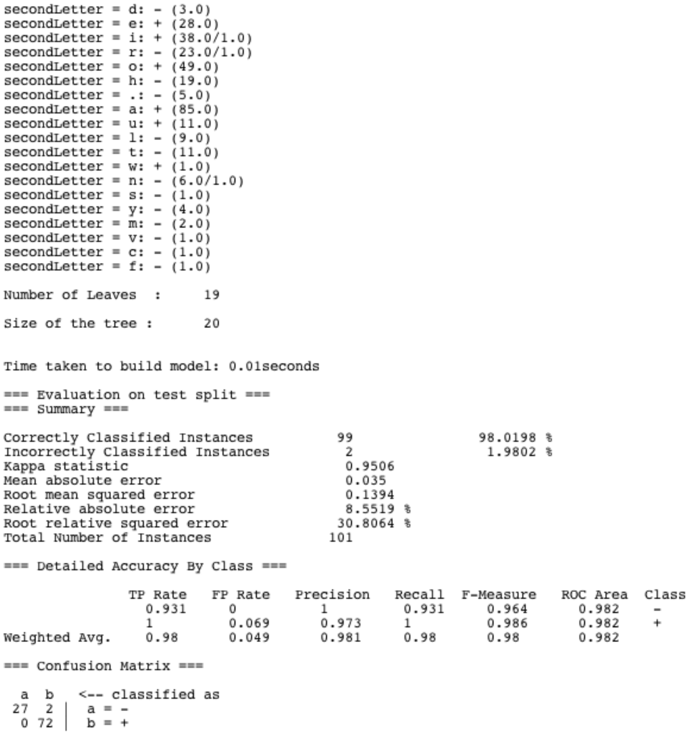
```
Time taken to build model: 0.01seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          77               76.2376 %
Incorrectly Classified Instances        24               23.7624 %
Kappa statistic                          0.3377
Mean absolute error                      0.2725
Root mean squared error                  0.4103
Relative absolute error                 66.6064 %
Root relative squared error             90.6973 %
Total Number of Instances              101

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.379     0.083      0.647     0.379     0.478      0.775    -
                 0.917     0.621      0.786     0.917     0.846      0.775    +
Weighted Avg.    0.762     0.466      0.746     0.762     0.741      0.775

=== Confusion Matrix ===

  a  b   <-- classified as
 11 18 |  a = -
  6 66 |  b = +
```

My next approach was to look at different types of letters. We have consonants and vowels, which can be easily extracted. So I used the counts from that and got about the same results as before in Weka.



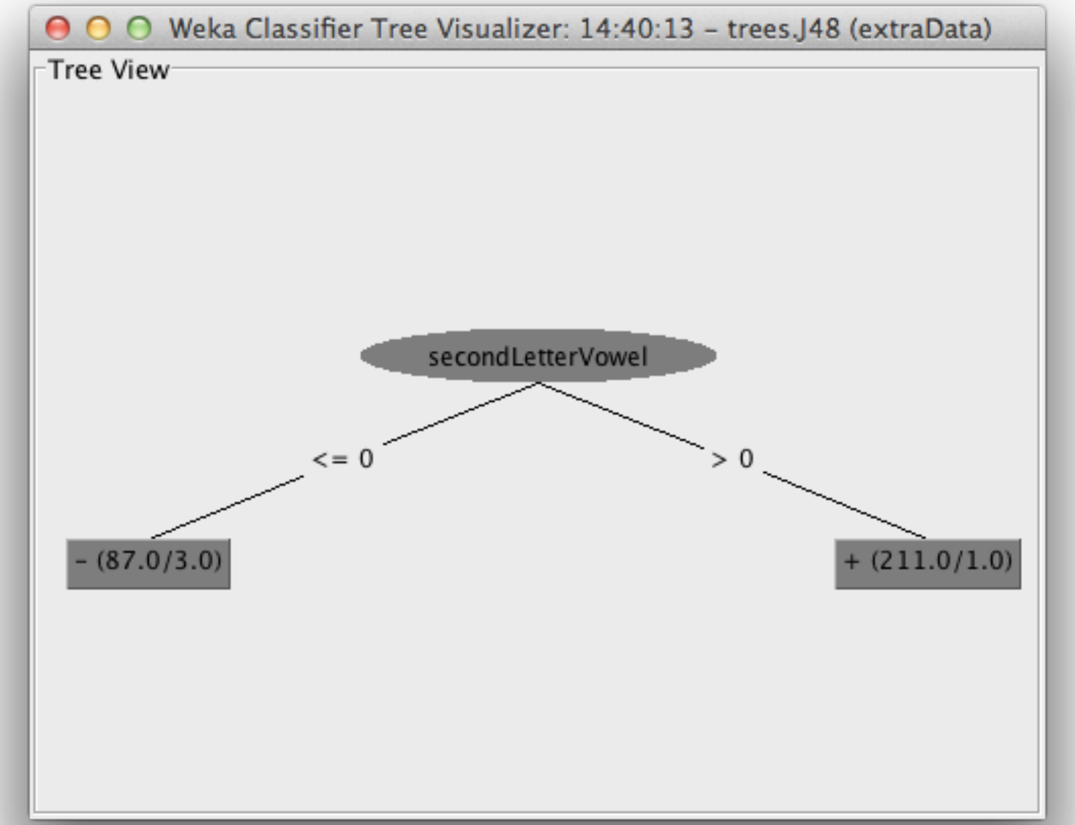
```
Time taken to build model: 0.01seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          78               77.2277 %
Incorrectly Classified Instances        23               22.7723 %
Kappa statistic                          0.3869
Mean absolute error                      0.2645
Root mean squared error                  0.4064
Relative absolute error                 64.6701 %
Root relative squared error             89.8299 %
Total Number of Instances              101

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.448     0.097      0.65      0.448     0.531      0.76     -
                 0.903     0.552      0.802     0.903     0.85       0.76     +
Weighted Avg.    0.772     0.421      0.759     0.772     0.758      0.76

=== Confusion Matrix ===

  a  b   <-- classified as
 13 16 |  a = -
  7 65 |  b = +
```

Looking back at how Thomases are losers while Toms are not, I decided that this is how I will get my foot in the door. Because each of those first names have the exact same label of winner or loser, it makes me think that the middle and last name have no relation to the fact of who is a winner. So, what seperates “Thomas” from “Tom”? The length of the name doesn’t really make sense because there visually those other names don’t show that pattern. How about the other letters in the name? So lets look at the second letter.



```
secondLetter = d: - (3.0)
secondLetter = e: + (28.0)
secondLetter = i: + (38.0/1.0)
secondLetter = r: - (23.0/1.0)
secondLetter = o: + (49.0)
secondLetter = h: - (19.0)
secondLetter = .: - (5.0)
secondLetter = a: + (85.0)
secondLetter = u: + (11.0)
secondLetter = l: - (9.0)
secondLetter = t: - (11.0)
secondLetter = w: + (1.0)
secondLetter = n: - (6.0/1.0)
secondLetter = s: - (1.0)
secondLetter = y: - (4.0)
secondLetter = m: - (2.0)
secondLetter = v: - (1.0)
secondLetter = c: - (1.0)
secondLetter = f: - (1.0)

Number of Leaves  :     19

Size of the tree :      20


Time taken to build model: 0.01seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          99               98.0198 %
Incorrectly Classified Instances         2                1.9802 %
Kappa statistic                          0.9506
Mean absolute error                      0.035
Root mean squared error                  0.1394
Relative absolute error                  8.5519 %
Root relative squared error             30.8064 %
Total Number of Instances              101

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.931     0          1         0.931     0.964      0.982    -
                 1         0.069      0.973     1         0.986      0.982    +
Weighted Avg.    0.98      0.049      0.981     0.98      0.98       0.982

=== Confusion Matrix ===

  a  b   <-- classified as
 27  2 |  a = -
  0 72 |  b = +
```

98% Classification. That's pretty good. Looking at what the second letter classified as winners turns out to be a, e, i, o, or u. Which we can easily recognize are vowels. So after adding an attribute of the second letter being a vowel we find that this is indeed true. We get the same split and a very simple tree:


Weka Classifier Tree Visualizer: 14:40:13 - trees.J48 (extraData)
Tree View
secondLetterVowel
<= 0
> 0
- (87.0/3.0)
+ (211.0/1.0)

The errors the noise in the data indeed are characters in *The Office*.