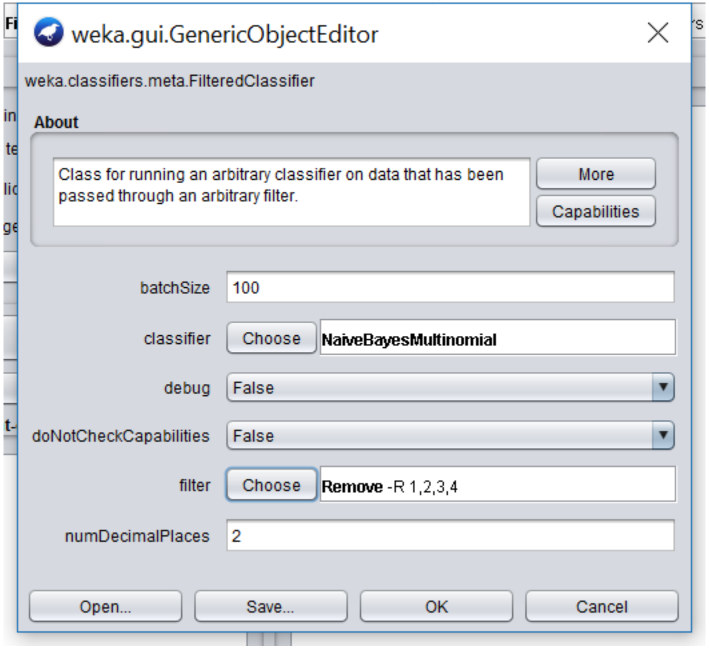
Project in class for LIS 590DT (Data Mining)
Date: 03.07.2017
Team Member: Yingjun Guan, Hui Lyu, Jingdu, Jingfang Zhang

Questions:
Is it possible to predict the gender (Attribute Genni) with the Name (First and/or Last Column)

Our solutions:
1) data preprocessing.
After importing the data (names_ethnea_genni_country_sample.csv), there are six attributes (AUID, Last, First, Ethnea, Genni, PubCountry).
Note: After discussion, we notice that First Name has a most influence on the gender. So instead of analyzing both First name and Last name, here First name is just used as the factor.
a) nominal to string
b) string to word vector

Note: The attribute Genni is filtered with RemoveWithValue (3, unknown).

Note: in the process of classifying, all the irrelevant attributes are filtered.



weka.gui.GenericObjectEditor

weka.classifiers.meta.FilteredClassifier

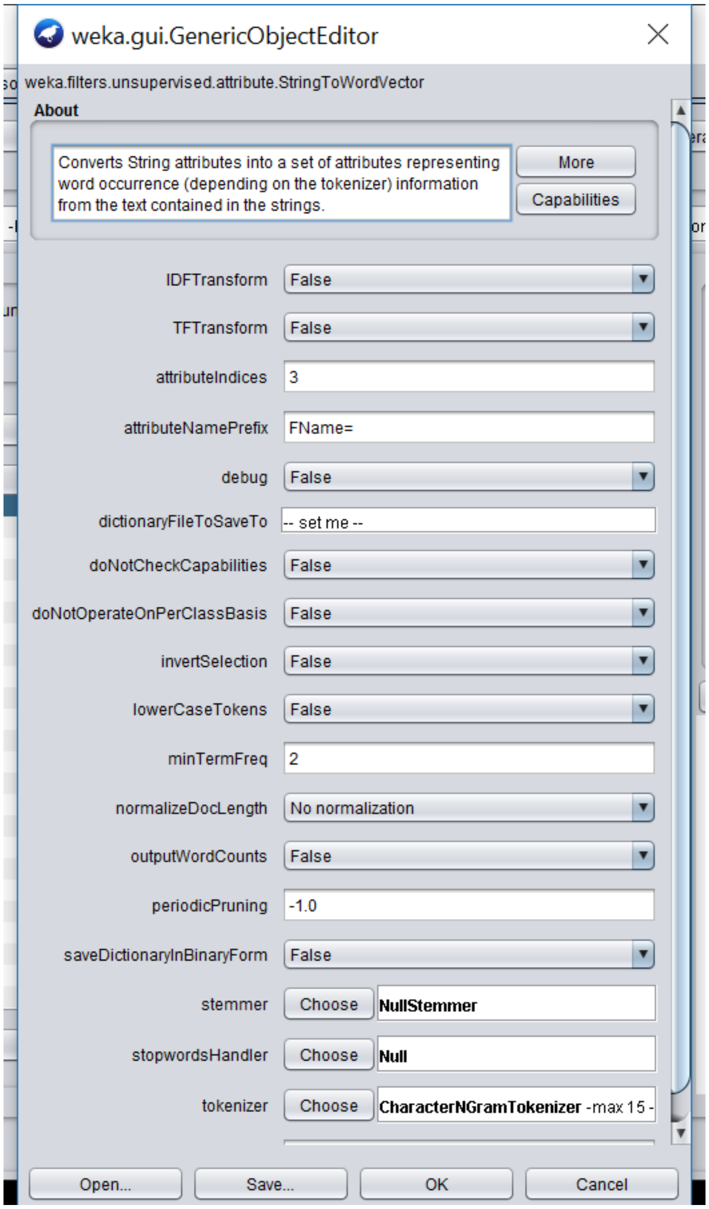
About

Class for running an arbitrary classifier on data that has been passed through an arbitrary filter.

batchSize: 100
classifier: NaiveBayesMultinomial
debug: False
doNotCheckCapabilities: False
filter: Remove -R 1,2,3,4
numDecimalPlaces: 2

Note: the genndi attribute is selected as class.
<Function: Meta/filterselction>

2) word vector settings
For the vector setting, we use the parameters as below:

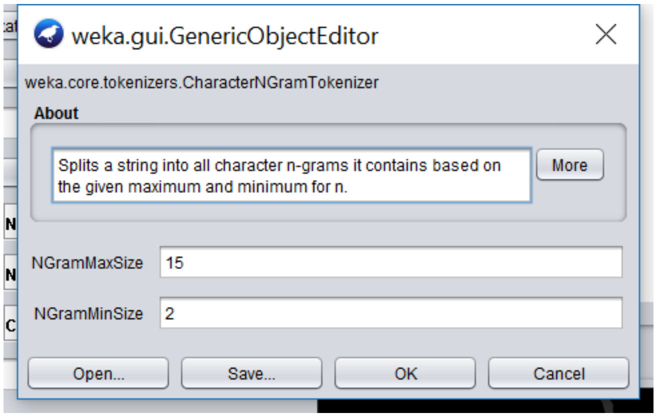


weka.filters.unsupervised.attribute.StringToWordVector

About

Converts String attributes into a set of attributes representing word occurrence (depending on the tokenizer) information from the text contained in the strings.

IDFTransform: False
TFTransform: False
attributeIndices: 3
attributeNamePrefix: FName=
debug: False
dictionaryFileToSaveTo: -- set me --
doNotCheckCapabilities: False
doNotOperateOnPerClassBasis: False
invertSelection: False
lowerCaseTokens: False
minTermFreq: 2
normalizeDocLength: No normalization
outputWordCounts: False
periodicPruning: -1.0
saveDictionaryInBinaryForm: False
stemmer: Choose NullStemmer
stopwordsHandler: Choose Null
tokenizer: Choose CharacterNGramTokenizer -max 15 -



weka.core.tokenizers.CharacterNGramTokenizer

About

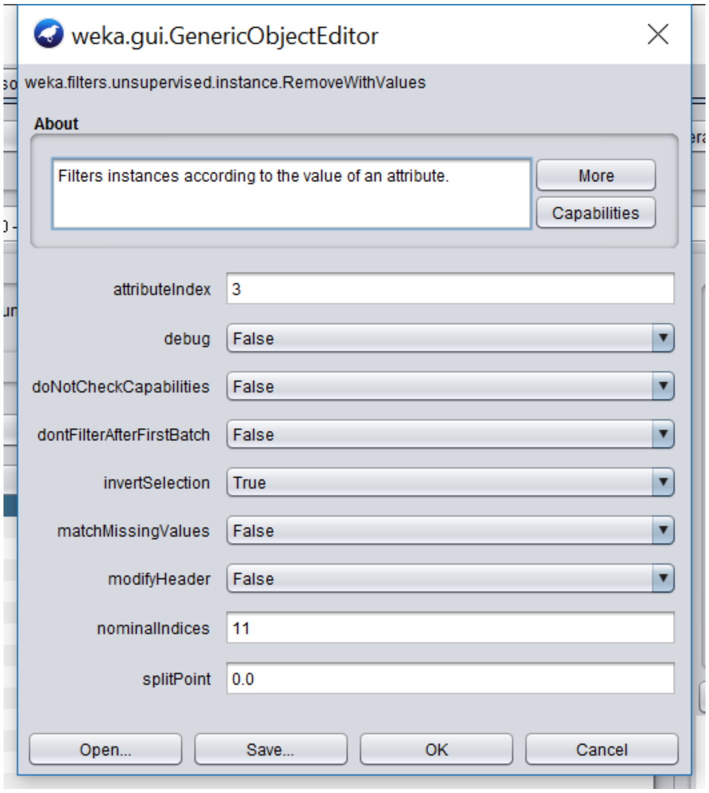
Splits a string into all character n-grams it contains based on the given maximum and minimum for n.

NGramMaxSize: 15
NGramMinSize: 2

3) country selection

We choose three senarios: all the countries, just English, Just Chinese. With the function of Meta/filtered classifier

The figure below shows how to filter just the English Ethenea



weka.filters.unsupervised.instance.RemoveWithValues

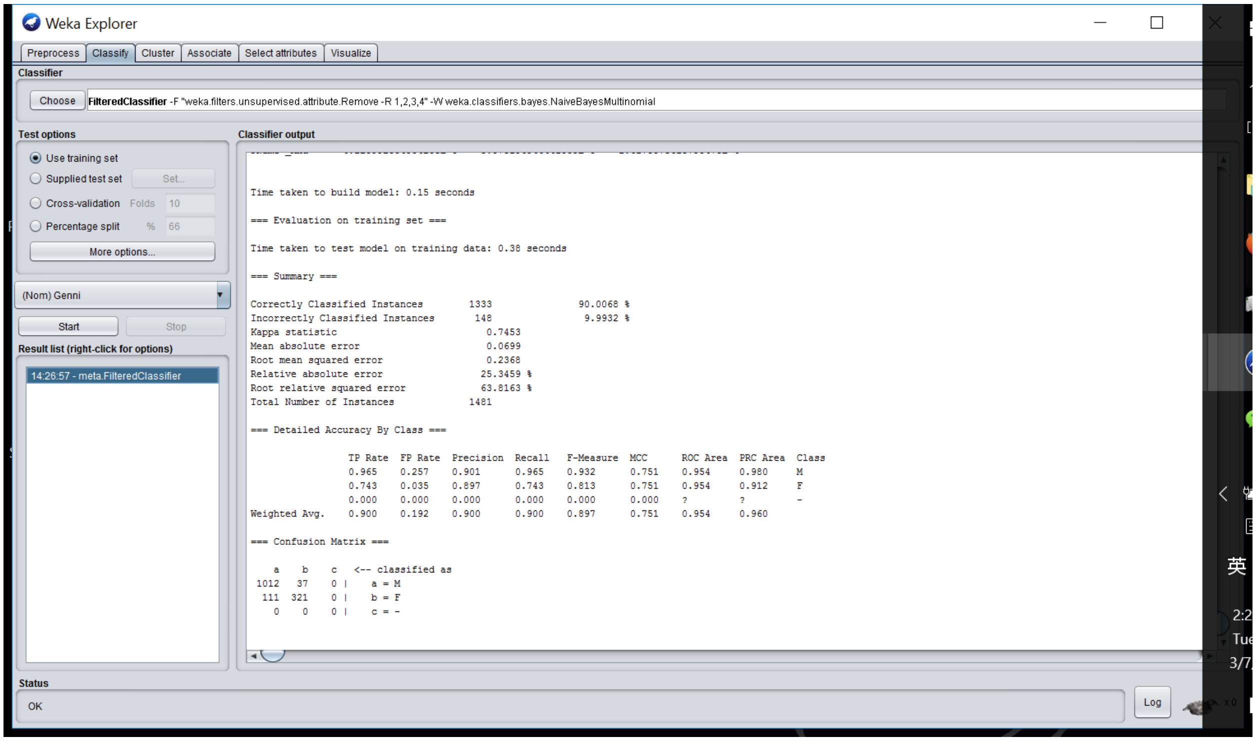
Filters instances according to the value of an attribute.

attributeIndex: 3
debug: False
doNotCheckCapabilities: False
dontFilterAfterFirstBatch: False
invertSelection: True
matchMissingValues: False
modifyHeader: False
nominalIndices: 11
splitPoint: 0.0

4) results

For all three scenarios, we have all the results. (Due to the memory problem, Test on training rather than cross-validation)

CHINESE:



FilteredClassifier -F "weka.filters.unsupervised.attribute.Remove -R 1,2,3,4" -W weka.classifiers.bayes.NaiveBayesMultinomial

● Use training set

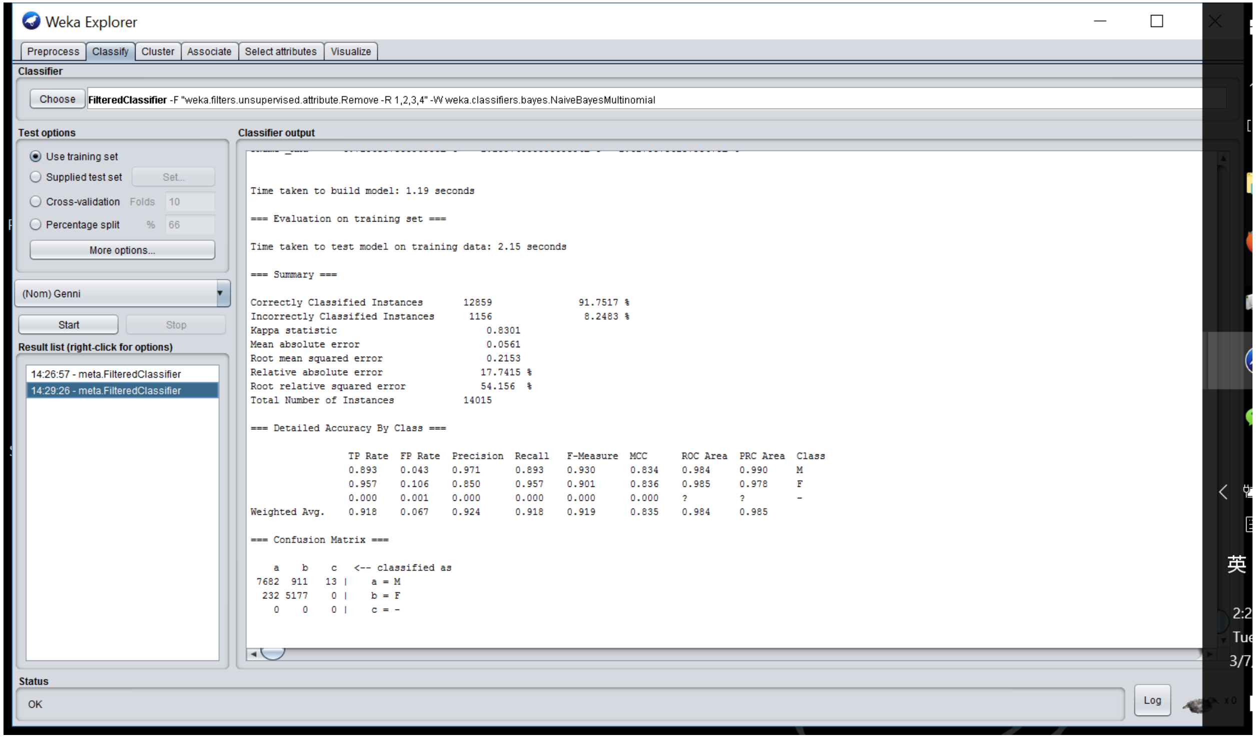
(Nom) Genni

14:26:57 - meta.FilteredClassifier

```
Time taken to build model: 0.15 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.38 seconds

=== Summary ===

Correctly Classified Instances        1333               90.0068 %
Incorrectly Classified Instances       148                9.9932 %
Kappa statistic                          0.7453
Mean absolute error                      0.0699
Root mean squared error                  0.2368
Relative absolute error                 25.3459 %
Root relative squared error             63.8163 %
Total Number of Instances             1481

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.965    0.257    0.901      0.965    0.932      0.751    0.954     0.980     M
                 0.743    0.035    0.897      0.743    0.813      0.751    0.954     0.912     F
                 0.000    0.000    0.000      0.000    0.000      0.000    ?         ?         -
Weighted Avg.    0.900    0.192    0.900      0.900    0.897      0.751    0.954     0.960

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1012   37    0 |    a = M
  111  321    0 |    b = F
    0    0    0 |    c = -
```

OK

English



FilteredClassifier -F "weka.filters.unsupervised.attribute.Remove -R 1,2,3,4" -W weka.classifiers.bayes.NaiveBayesMultinomial

● Use training set

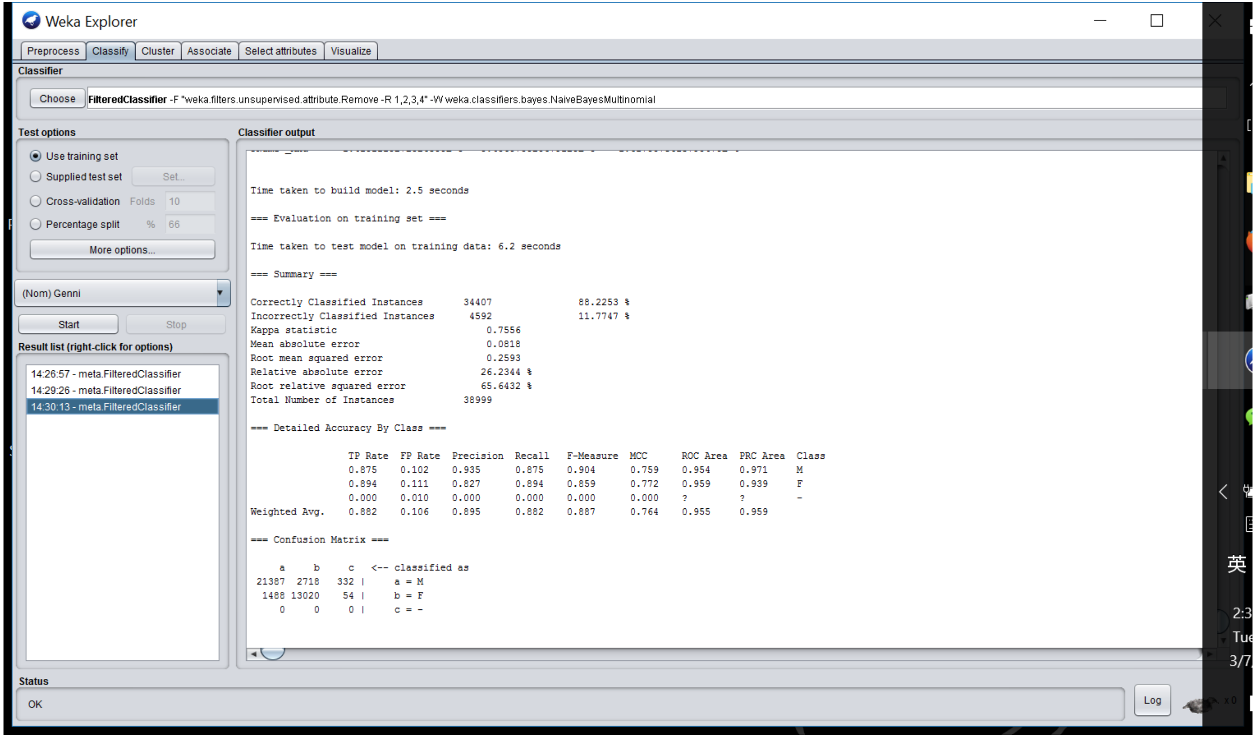
(Nom) Genni

14:26:57 - meta.FilteredClassifier
14:29:26 - meta.FilteredClassifier

```
Time taken to build model: 1.19 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 2.15 seconds

=== Summary ===

Correctly Classified Instances       12859               91.7517 %
Incorrectly Classified Instances      1156                8.2483 %
Kappa statistic                          0.8301
Mean absolute error                      0.0561
Root mean squared error                  0.2153
Relative absolute error                 17.7415 %
Root relative squared error             54.156  %
Total Number of Instances            14015

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.893    0.043    0.971      0.893    0.930      0.834    0.984     0.990     M
                 0.957    0.106    0.850      0.957    0.901      0.836    0.985     0.978     F
                 0.000    0.001    0.000      0.000    0.000      0.000    ?         ?         -
Weighted Avg.    0.918    0.067    0.924      0.918    0.919      0.835    0.984     0.985

=== Confusion Matrix ===

    a    b    c   <-- classified as
 7682  911   13 |    a = M
  232 5177    0 |    b = F
    0    0    0 |    c = -
```

OK

All the languages



FilteredClassifier -F "weka.filters.unsupervised.attribute.Remove -R 1,2,3,4" -W weka.classifiers.bayes.NaiveBayesMultinomial

● Use training set
○ Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) Genni

Result list (right-click for options)

14:26:57 - meta.FilteredClassifier
14:29:26 - meta.FilteredClassifier
14:30:13 - meta.FilteredClassifier

```
Time taken to build model: 2.5 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 6.2 seconds

=== Summary ===

Correctly Classified Instances       34407               88.2253 %
Incorrectly Classified Instances      4592               11.7747 %
Kappa statistic                          0.7556
Mean absolute error                      0.0818
Root mean squared error                  0.2593
Relative absolute error                 26.2344 %
Root relative squared error             65.6432 %
Total Number of Instances            38999

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.875    0.102    0.935      0.875    0.904      0.759    0.954     0.971     M
                 0.894    0.111    0.827      0.894    0.859      0.772    0.959     0.939     F
                 0.000    0.010    0.000      0.000    0.000      0.000    ?         ?         -
Weighted Avg.    0.882    0.106    0.895      0.882    0.887      0.764    0.955     0.959

=== Confusion Matrix ===

     a     b     c   <-- classified as
 21387  2718   332 |     a = M
  1488 13020    54 |     b = F
     0     0     0 |     c = -
```

OK

5) Conclusion

For the prediction accuracy,

Language	Prediction accuracy
All countries	88%
Chinese	90%
English	91%

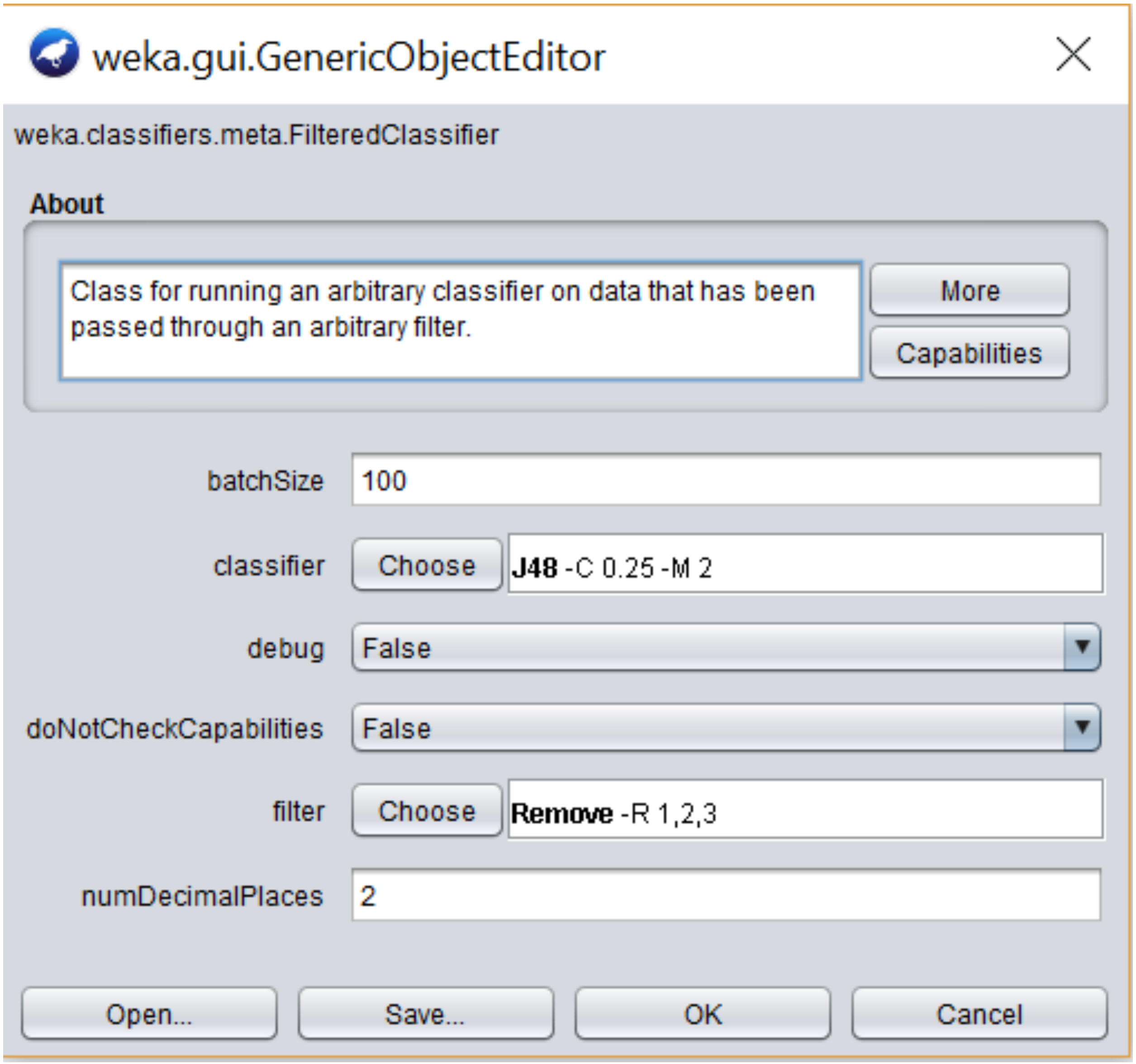
As a conclusion, we can use the name (First name) to predict the gender with a good accuracy! For different athnea, English behaves better than Chinese, than the overall performance.

6) Other efforts.

We tried establishing a CHINESE only document, and has it analyzed, the result is similar to the answer above.

We also tried using the decision tree to analyze the results.

Modify classifier to J48, for Chinese only instances



weka.gui.GenericObjectEditor

weka.classifiers.meta.FilteredClassifier

About

Class for running an arbitrary classifier on data that has been passed through an arbitrary filter.

batchSize 100

classifier **J48** -C 0.25 -M 2

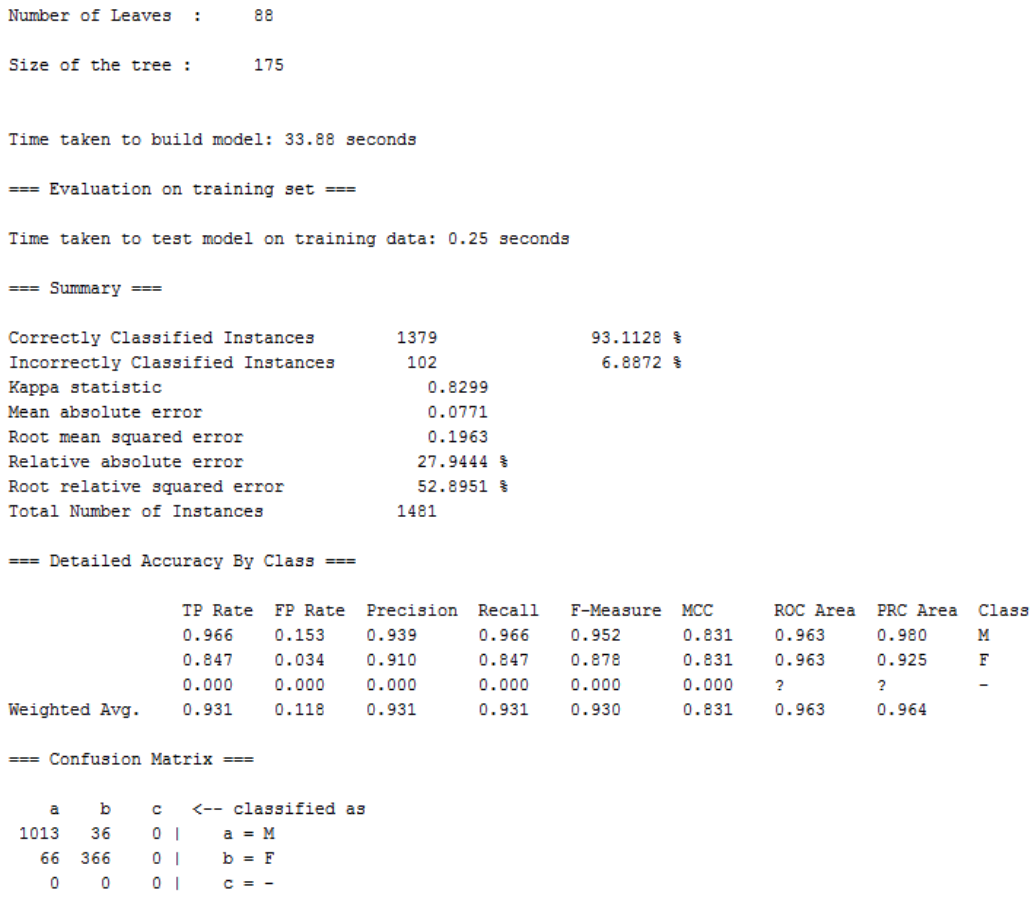
debug False

doNotCheckCapabilities False

filter **Remove** -R 1,2,3

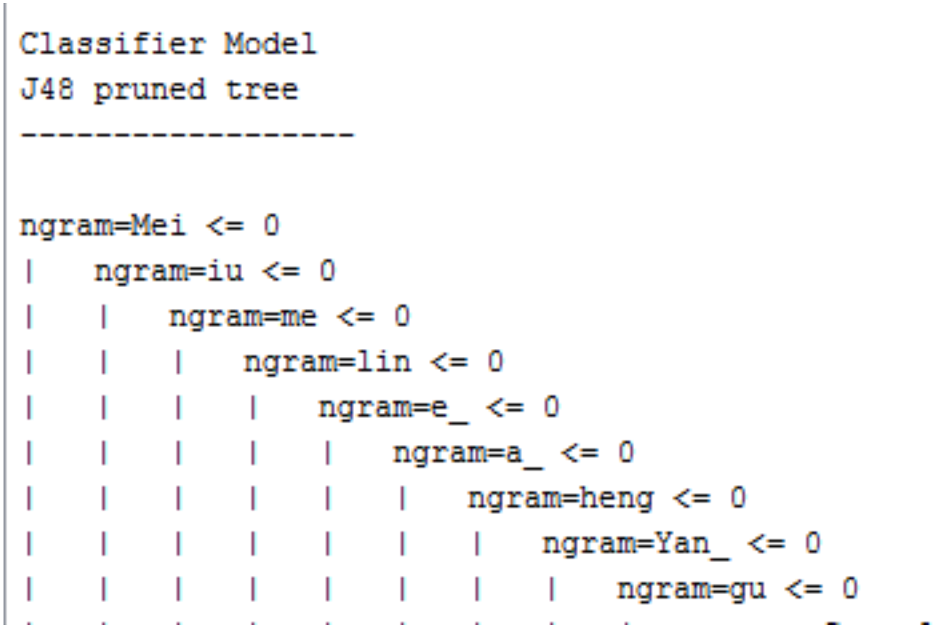
numDecimalPlaces 2

The correct rate raises up about 3%, to 93%.



```
Number of Leaves  :     88

Size of the tree :      175


Time taken to build model: 33.88 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.25 seconds

=== Summary ===

Correctly Classified Instances        1379               93.1128 %
Incorrectly Classified Instances       102                6.8872 %
Kappa statistic                          0.8299
Mean absolute error                      0.0771
Root mean squared error                  0.1963
Relative absolute error                 27.9444 %
Root relative squared error             52.8951 %
Total Number of Instances             1481

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.966    0.153    0.939      0.966    0.952      0.831    0.963     0.980     M
                 0.847    0.034    0.910      0.847    0.878      0.831    0.963     0.925     F
                 0.000    0.000    0.000      0.000    0.000      0.000    ?         ?         -
Weighted Avg.    0.931    0.118    0.931      0.931    0.930      0.831    0.963     0.964

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1013   36    0 |    a = M
   66  366    0 |    b = F
    0    0    0 |    c = -
```

This is the visualized tree. The top gram is “Mei”.



```
Classifier Model
J48 pruned tree
------------------


ngram=Mei <= 0
|   ngram=iu <= 0
|   |   ngram=me <= 0
|   |   |   ngram=lin <= 0
|   |   |   |   ngram=e_ <= 0
|   |   |   |   |   ngram=a_ <= 0
|   |   |   |   |   |   ngram=heng <= 0
|   |   |   |   |   |   |   ngram=Yan_ <= 0
|   |   |   |   |   |   |   |   ngram=gu <= 0
```