Nick Petty

CAP6776 Homework 2

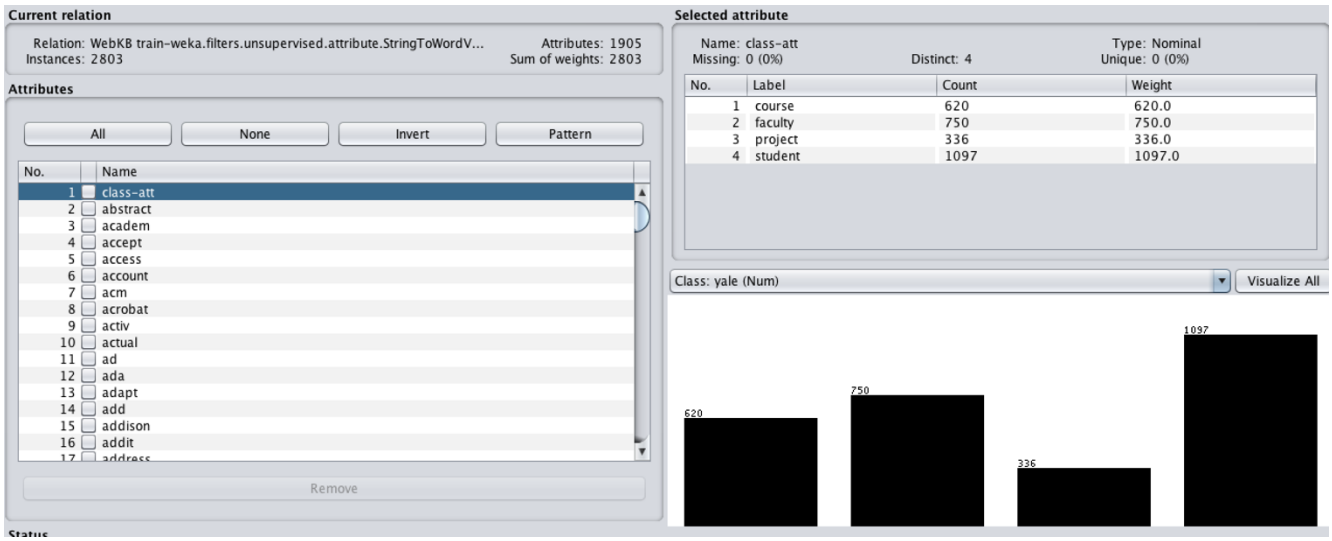
1. Converting .txt files to .arff files
    a. The provided webkb-train-stemmed.txt and webkb-test-stemmed.txt documents cannot be read by Weka.
    b. The attached Python script, TxtToArff.py, converts these files into webkb-train-stemmed.arff and webkb-test-stemmed.arff.
    c. The given files are formatted as <class>\t<document>\n, so they are converted to <.arff header>\n'<document>,<class> in the output files.
    d. Here is the header and a few documents in the training set:

```
1  @relation 'WebKB train'
2
3  @attribute Text string
4  @attribute class-att {course, faculty, project, student}
5
6  @data
7
8  'brian comput scienc depart univers wisconsin dayton street madison offic email wisc offic phone home phone advisor david wood tabl content interest schedul summer public
   interest profession comput architectur oper system compil high speed network distribut parallel system secur account high perform person bicycl walk hike camp travel home
   brew cook comput electron read schedul mondai wwt meet wednesdai meet david cow meet milwauke brian heidi wed madison comput architectur affili meet chicago base public
   journal articl foster perform massiv parallel comput spectral atmospher model atmospher ocean technolog byte drake foster design perform scalabl parallel commun climat
   model parallel comput decemb byte proceed paper foster algorithm comparison benchmark parallel spectral transform water model sixth workshop parallel process meteorolog
   ed world scientif singapor byte drake foster hack williamson adapt scalabl parallel comput proceed global chang symposium american meteorolog societi byte foster load
   balanc algorithm climat model proceed scalabl high perform comput confer ed walker ieee comput societi press byte technic report user guid technic report juli byte foster
   load balanc algorithm parallel commun climat model anl technic report anl mc januari byte poster present foster sutton wagner harrison kendal calcul librari gordon
   research confer high perform comput nation inform infrastructur juli foster sutton wagner calcul librari high perform comput chemistri workshop hilton california august
   earth belong man man belong earth thing connect blood unit man web life strand web chief seattl man sat ground medit life mean accept creatur acknowledg uniti univers
   thing true essenc civil stand bear modifi mon aug cdt',student
9  'denni swanson web page mail pop uki offic hour comput lab offic anderson quadrangl mailbox anderson hall lab resum dilbert comic sport data mine web imag web
   yahoo',student
10 'russel impagliazzo depart comput scienc engin univers california san diego jolla offic appli physic mathemat build apm phone fax email russel ucsd assist professor
   special complex theori research circuit lower bound theori cryptographi comput random cours fall cse algorithm cse algorithm offic hour student prioriti mondai wednesdai
   student prioriti tuesdai thursdai research paper beam cook edmond impagliazzo rel complex search problem beam impagliazzo improv depth lower bound small distanc connect
   beam impagliazzo lower bound hilbert proposit proof impagliazzo reachabl problem finit cellular automata edmond impagliazzo commun complex lower bound circuit depth gupta
   impagliazzo comput planar impagliazzo levin construct pseudo random gener function impagliazzo person view averag case complex impagliazzo distribut hard problem
   impagliazzo effici cryptograph scheme provabl secur subset sum impagliazzo effect random boolean formula impagliazzo paturi size depth trade off threshold circuit
   impagliazzo upper lower bound tree cut plane proof impagliazzo wigderson network algorithm impagliazzo limit provabl consequ permut beam impagliazzo exponenti lower bound
   constant depth proof principl',faculty
```

2. Filtering the .arff files
    a. Using the .arff files as-is, without filtering will result in incompatible classifiers.
    b. Online tutorials recommend using the command line to apply filters to the .arff files, as shown:

```
> java weka.filters.unsupervised.attribute.StringToWordVector -b -i
/Users/Nick/OneDrive/Documents/CAP6776_Info_Retr/hw2/TxtToArff/arff/webkb-train-stemmed.arff -o
/Users/Nick/OneDrive/Documents/CAP6776_Info_Retr/hw2/TxtToArff/arff/webkb-train-stemmed-vector.arff -c last -r
/Users/Nick/OneDrive/Documents/CAP6776_Info_Retr/hw2/TxtToArff/arff/webkb-test-stemmed.arff -s
/Users/Nick/OneDrive/Documents/CAP6776_Info_Retr/hw2/TxtToArff/arff/webkb-test-stemmed-vector.arff -C
```

    This filter applies the word count parameter with the '-C' argument.
    c. Loading the training set into Weka:



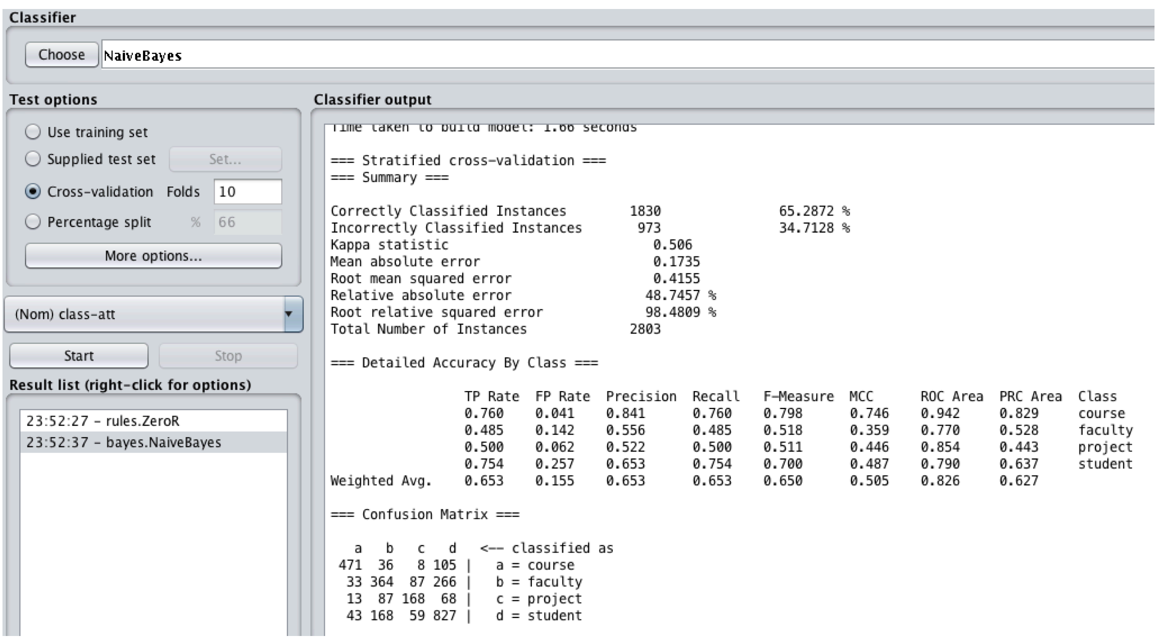
Nick Petty
CAP6776 Homework 2

d. Setting the class attribute as class gives a document-word matrix, which is partially shown here:

No.	1: abstract Numeric	2: academ Numeric	3: accept Numeric	4: access Numeric	5: account Numeric	6: acm Numeric	7: acrobat Numeric	8: activ Numeric	9: actual Numeric	10: ad Numeric	11: ada Numeric
1	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0

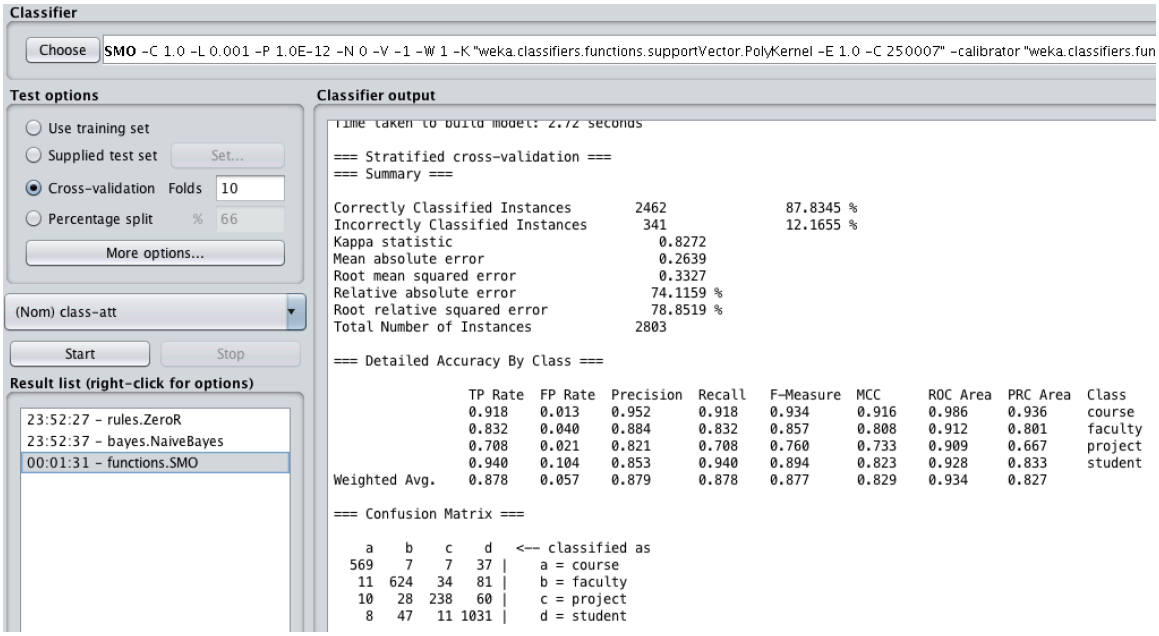
3. Building the classifiers

a. Using the training set with Naïve Bayes and 10-fold cross validation gives the following results:



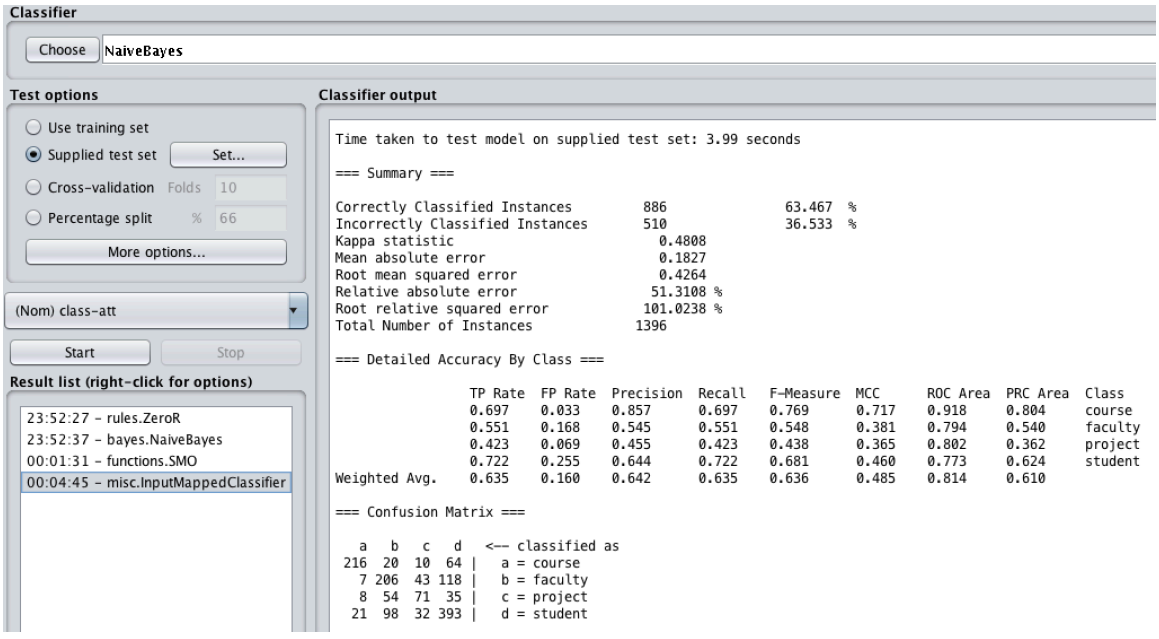
Nick Petty
CAP6776 Homework 2

b. The same set with SVM:



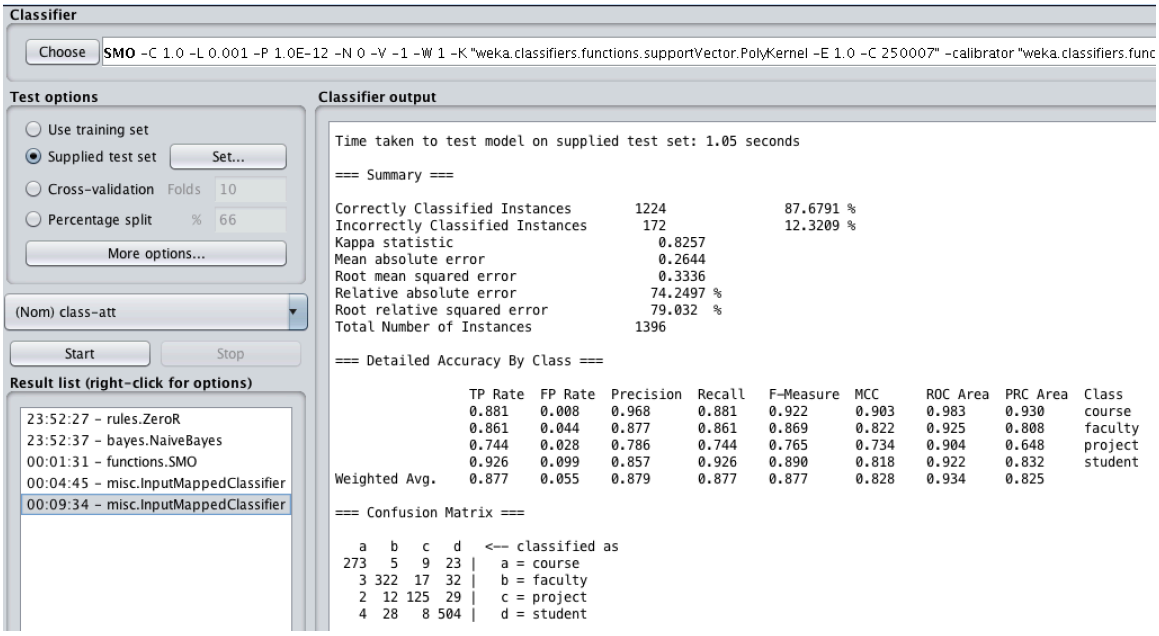
c. This Naïve Bayes classifier was about 65% accurate, while the Support Vector Machine classifier was about 88% accurate.
d. For this training set, the SVM classifier has much better results.

4. Testing the classifiers
   a. The training set is run against the test set. This is with the Naïve Bayes classifier on default parameters:



Nick Petty
CAP6776 Homework 2

b. The training set is run against the test set with SVM and default parameters:



c. Again, the Naïve Bayes classifier was much less accurate than the Support Vector Machine, at 64% accuracy to 88% accuracy.
d. With the provided training and test data sets, Weka shows the SVM classification is better than Naïve Bayes at determining which documents refer to courses, students, projects, or faculty.