

# Programming Project #2

## Along Party Lines

This problem set is due 11/20.

Since it is an important year for U.S. elections, we'll do a political programming project. In a separate file, we've included the party membership and voting records of various member of congress. We'd like you to use this data to classify congressmen as Republicans or Democrats.

We'd like you to use Bayesian Classifiers and Decision Trees. We've included their votes on a number of issues. The data set is from the 80s, so the issues aren't current (you can envision the congressmen with funny hair if you'd like). And the issues are:

1. Handicapped Infants
2. Water Project Cost Sharing
3. Adopt the Budget Resolution
4. Physician Fee Freeze
5. Aid to El Salvador
6. Religious Groups in Schools
7. Anti Satellite Test Ban
8. Aid to Nicaraguan Contras
9. MX Missile
10. Immigration
11. Synfuels Corporation Cutback
12. Education Spending
13. Superfund Right to Sue
14. Crime
15. Duty Free Exports
16. South Africa Export Administration Act

An individual congressmen is represented by a line in the training file:

```
no,yes,yes,no,no,maybe,yes,yes,yes,yes,yes,no,maybe,yes,yes,yes, democrat
```

Which shows their votes on the issues and their political party. Congressmen who didn't vote during a particular vote are given a "maybe". You may notice that some congressmen have not voted at all.

You have three files:

1. `training-large.csv`: The main training set
2. `training-small.csv`: A smaller training set
3. `testing.csv`: The testing set

## 1 Part 1: Classification

For each part, create a classifier using the suggested python modules. Train the classifier on the small training set and then test using the test data. Collect precision and recall information. Do the same thing for the larger training set.

Please document your code and make the structure clean, readable, and intuitive.

### 1.1 Part 1a: Bayesian Classifier

Use a naïve Bayes classifier to classify the politicians.

You should use the Reverend classifier: <http://divmod.org/trac/wiki/DivmodReverend>. You cannot simply just plug in the classifier; you will have to write a tokenizer to split up the training input appropriately.

### 1.2 Part 1b: Decision Trees

You can find a Python implimentation of the ID3 algorithm here (along with a long tutorial): [http://www.onlamp.com/pub/a/python/2006/02/09/ai\\_decision\\_trees.html](http://www.onlamp.com/pub/a/python/2006/02/09/ai_decision_trees.html)

## 2 Part 2: Discussion

Now that you've classified things, please discuss the different classification techniques. Did different classifiers do better with this data set? How did the size of the training set effect the results? Did different methods work better with different size training sets?

This should definitely be longer than a paragraph, and should be as long as you need to get your ideas across.