

# Hackathon #1: Where are you from?

**Team 1:** John, Jordan, Mark, Nick

**Team 2:** Alex B., Andrew, Yuriy, Sushant

**Team 3:** Dakota, Larwan, Rachel, Sam

**Team 4:** Alex H., Amy, Leo

**Team 5:** Christine, Piotr, Ricky, Tre

## Step 1: Data

Download the data from DropBox:

<https://www.dropbox.com/s/yhdcw34jjhieym4/hackathon.zip?dl=0>

## Step 2: Extract features

1. Download `helpfulscript` from the Hackathon folder on myCourses.
2. Launch Praat.
3. Open the script from the Praat menu.
4. Change the values of the variables indicated. (Look for comments with all caps.)
5. Run the script.

## Step 3, Option A: Classify using Weka GUI

1. Launch Weka.
2. Click the Explorer button.
3. Click Open file...
4. From the File Format dropdown menu, pick All Files.
5. Navigate to where your output csv file is, and open it.
6. Click the Classify tab.
7. Choose bayes->NaiveBayes.
8. Click Start.

## Step 3, Option B: Classify using sklearn python library

In the Hackathon 1 directory, I've included a little python script that will do some classification with sklearn. It reads in the output file, then classifies using 5-fold cross validation.

## Step 4: Improve classification

Explore the features you just extracted to see which ones are most informative.

In Weka, you can do this in the Preprocess window by selecting and removing attributes. Don't forget that you always need the final "dx" attribute, which is the class label (dementia or control). You can also go to the Select Attributes tab and see what some of the built-in feature-selection methods suggest.

In sklearn, you can just use numpy array slicing to get the features you're interested in. There are a few commented out examples in the code.

# Hackathon #1: Where are you from?

## Step 5: Explore more features, make more improvements

Now that you now how to extract features and do classification, go modify the script to extract different sorts of features. Using the Paste history option is helpful, as is googling Praat scripts to see how other people have done it. Some ideas for features:

F0 (a.k.a. pitch)	length of utterance file
shimmer	MFCC
jitter	silence vs. speech
harmonicity	ratios of features to one another
voicing	range, min, mean, max of any feature

You should also try labeling the audio with phonetic information. You can do this by hand, or to speed things up, by doing forced alignment. Here are a few webpages that allow you do submit files for alignment so you don't have to install an aligner yourself:

<https://phon.wordpress.ncsu.edu/lab-manual/forced-alignment/>  
<http://fave.ling.upenn.edu/FAAValign.html>

You can also try installing forced aligner software yourself. Here are some options:

Penn forced aligner: <http://web.sas.upenn.edu/phonetics-lab/facilities/>  
Prosody Lab: <http://prosodylab.org/tools/aligner/>  
Kaldi: <http://kaldi-asr.org>  
EasyAlign: <http://latlcui.unige.ch/phonetique/easyalign.php>

Helpful: [http://www.fon.hum.uva.nl/praat/manual/Voice\\_6\\_\\_Automating\\_voice\\_analysis\\_with\\_a\\_script.html](http://www.fon.hum.uva.nl/praat/manual/Voice_6__Automating_voice_analysis_with_a_script.html)