

FNC Methods & Packages

Two-step classification

Step 1: Classify related/unrelated

Step 2: Classify agree/disagree/discuss

What methods could you try for each one? Discuss!

Two-step classification

Step 1: Classify related/unrelated

Possible methods:

Two-step classification

Step 2: Classify agree/disagree/discuss

Possible methods:

Packages Recap

- A bunch of useful Python scripts bundled together for your convenience
- “Black box” = details of implementation are hidden
- Have to rely on documentation
- Installation method?

Why use packages for NLP?

- Saves you from doing statistical calculations by hand!
- Saves time on NLP-specific repeated tasks

```
>>> import nltk
>>> sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
>>> tokens = nltk.word_tokenize(sentence)
>>> tokens
['At', 'eight', 'o'clock', 'on', 'Thursday', 'morning',
'Arthur', 'did', 'n't', 'feel', 'very', 'good', '.']
```

- Fast and easy to try and compare different methods on the same data

Why is it still important to know what a package does?

- Understand which methods are the best fit for your data
- Understand what the parameters are

```
class sklearn.linear_model. LogisticRegression (penalty='l2', dual=False, tol=0.0001, C=1.0,  
fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='liblinear', max_iter=100,  
multi_class='ovr', verbose=0, warm_start=False, n_jobs=1)
```

[\[source\]](#)

- Customize the parameters to get better results
- If something goes wrong, you will notice and can troubleshoot

scikit-learn (sklrm)

- Machine learning package in Python
- Methods for (among others):
 - Classification (SVM, Nearest Neighbor, Naïve Bayes, Decision Trees, etc.)
 - Clustering (K-Means, etc.)
 - Pre-processing (normalize data, obtain best features, etc.)

nlTK

- Natural language processing toolkit
- Provides access to data like corpora (bodies of example text for statistical analysis)
- Methods for:
 - Classification
 - Sentence parsing
 - Tokenization
 - Most natural language applications!