

Applications & Tools Demo



Technology

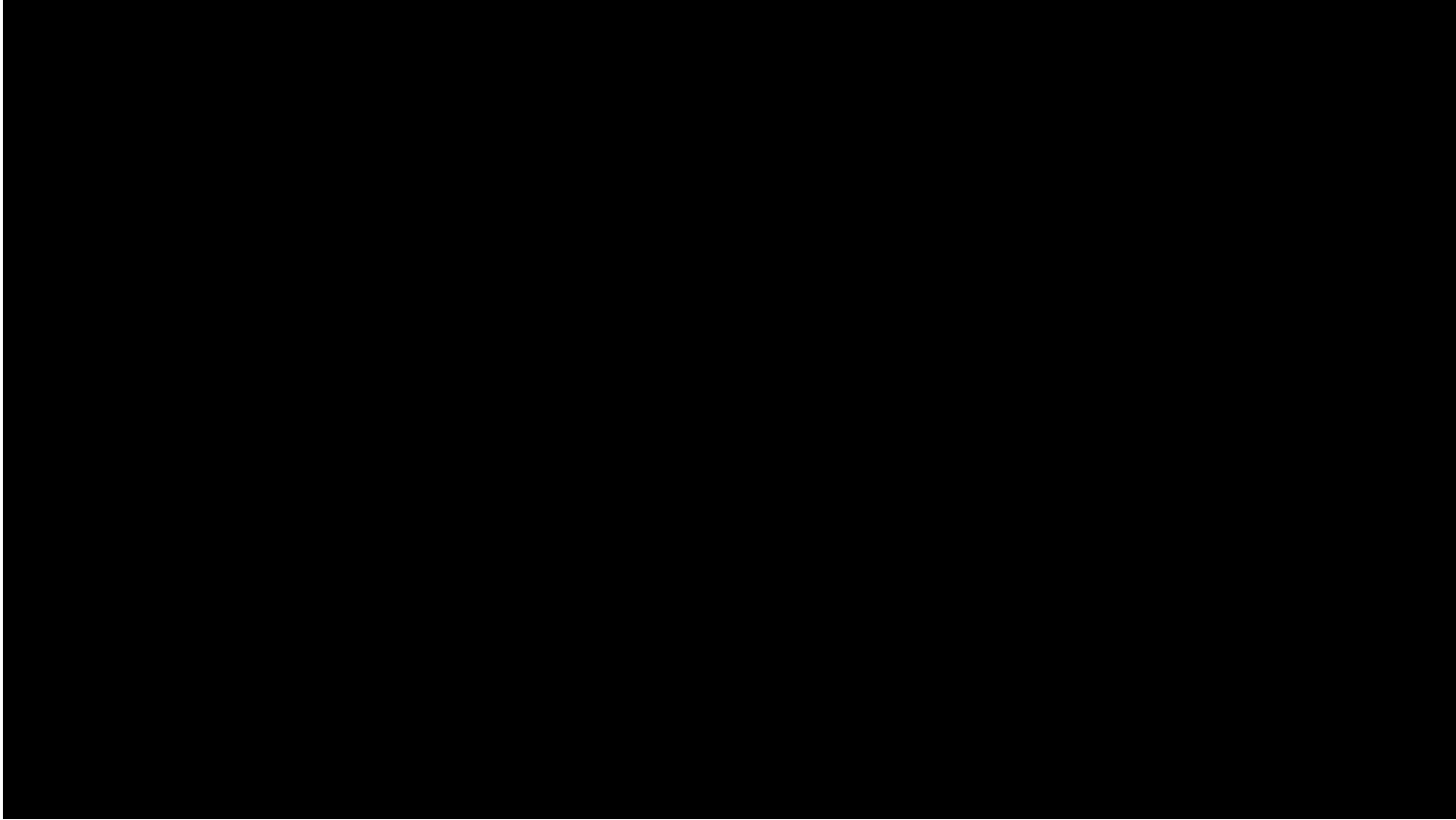
- Open-source, text-mining tool.
- “Machine Learning Made Easy”
(We shall see...)



Technology Applied

- Writing support for students and teachers in the English Language Arts classroom in grades 6-12.
- Automated essay scoring, customized to your content, hosted in the cloud, and embedded in your applications.

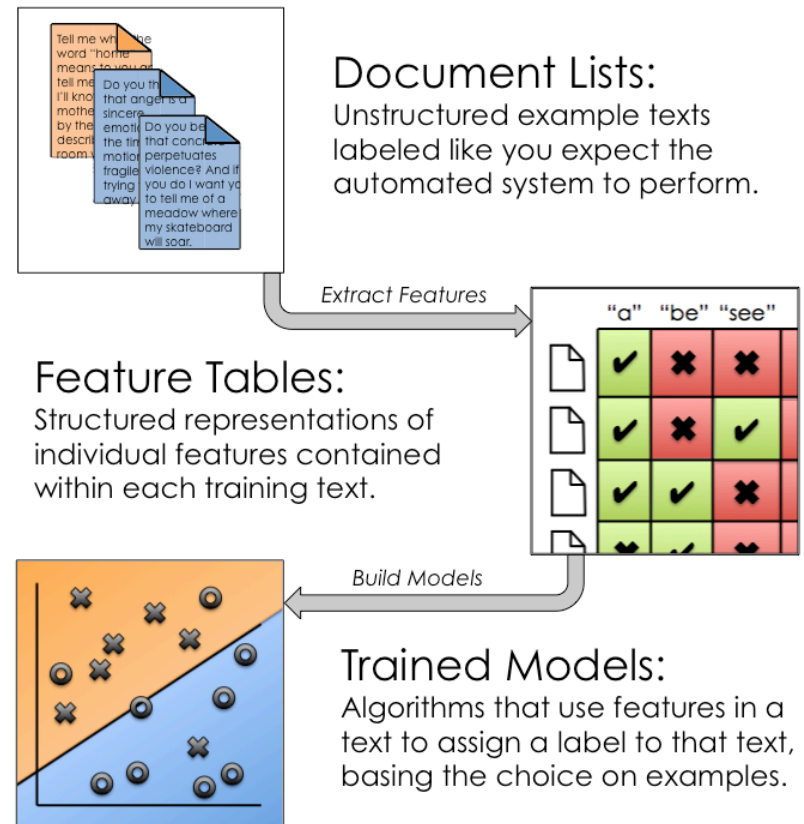




Optimal Scenario for LightSide

1. You're in a situation where text is coming in for your analysis faster than humans can keep up with it.
2. For each text that comes in, you want to assign a single label or number value to that text.
3. You've already defined what your possible set of labels or numbers are, and you've tested to ensure that humans can reliably agree when doing this labeling by hand.
4. Those humans have already sat down and labeled at least several hundred examples, with many examples of each label you're interested in.

Introduction to the LightSIDE Workflow



Help Tiffany and Jenn Get to the LightSide!

Tiffany and Jenn are beginning learning analytics students with little background on mining text and discourse.

They both really like the practical application of the LightSide technology, but are confused and worried how to guide their classmates through a demo.

As a class, help guide them through steps in the interface in order **to get from a set of data to a trained model** before they make a mad dash for the door!



Sentiment Analysis

The dataset we will be using contains about 10,000 example sentences, half of which are positive and half of which are negative, including sentiments that are:

Obvious

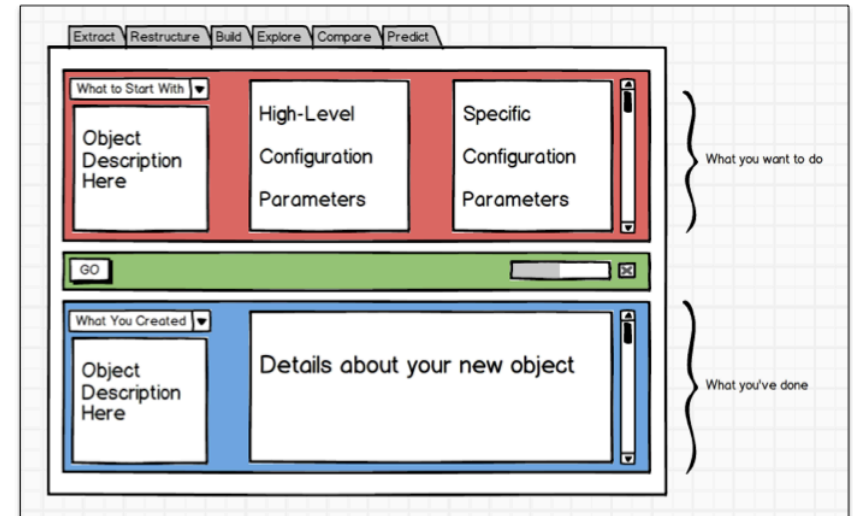
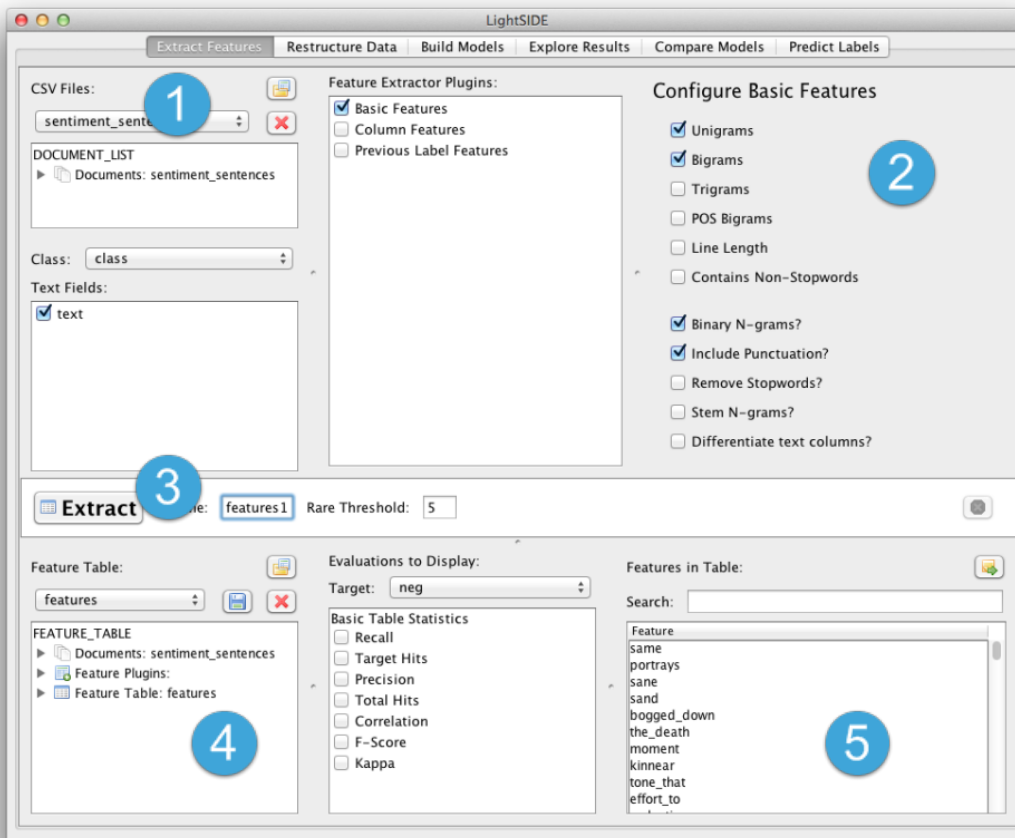
- “This warm and gentle romantic comedy has enough interesting characters to fill several movies, and its ample charms should win over the most hard-hearted cynics.”

A little more cryptic, requiring domain knowledge

- “An afterschool special without the courage of its convictions.”

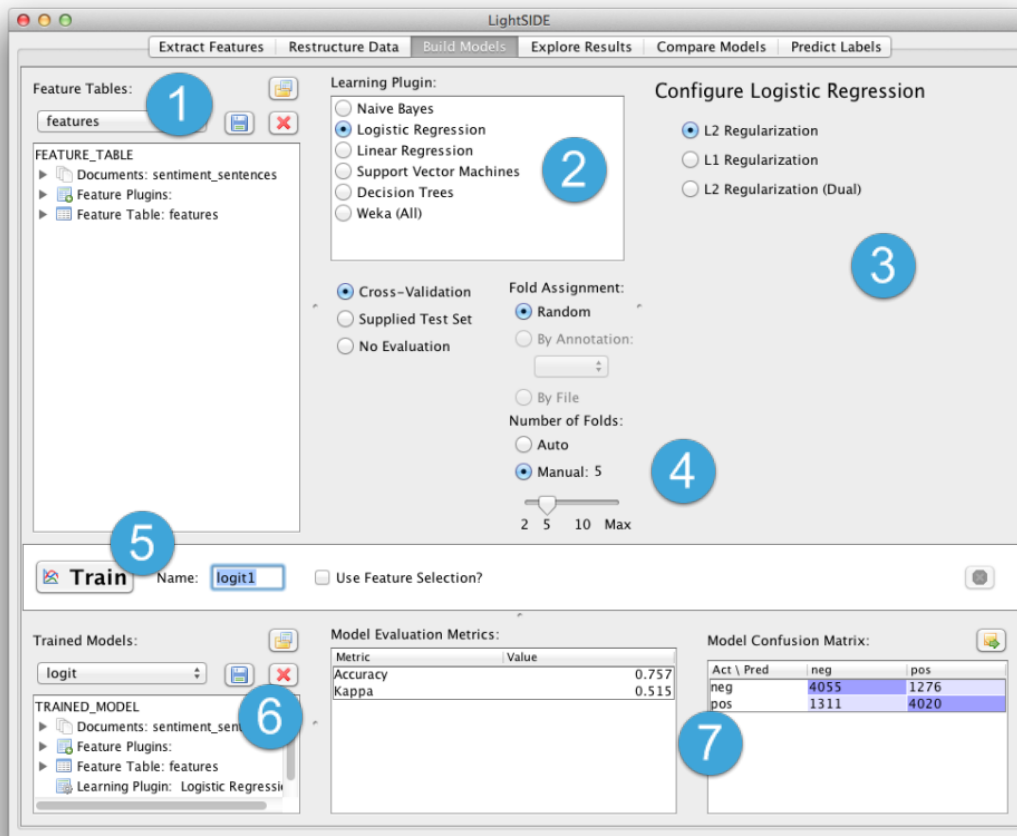
Difficult for even humans to clearly categorize

- “Somewhere short of tremors on the modern b-scene: neither as funny nor as clever, though an agreeably unpretentious way to spend ninety minutes.”



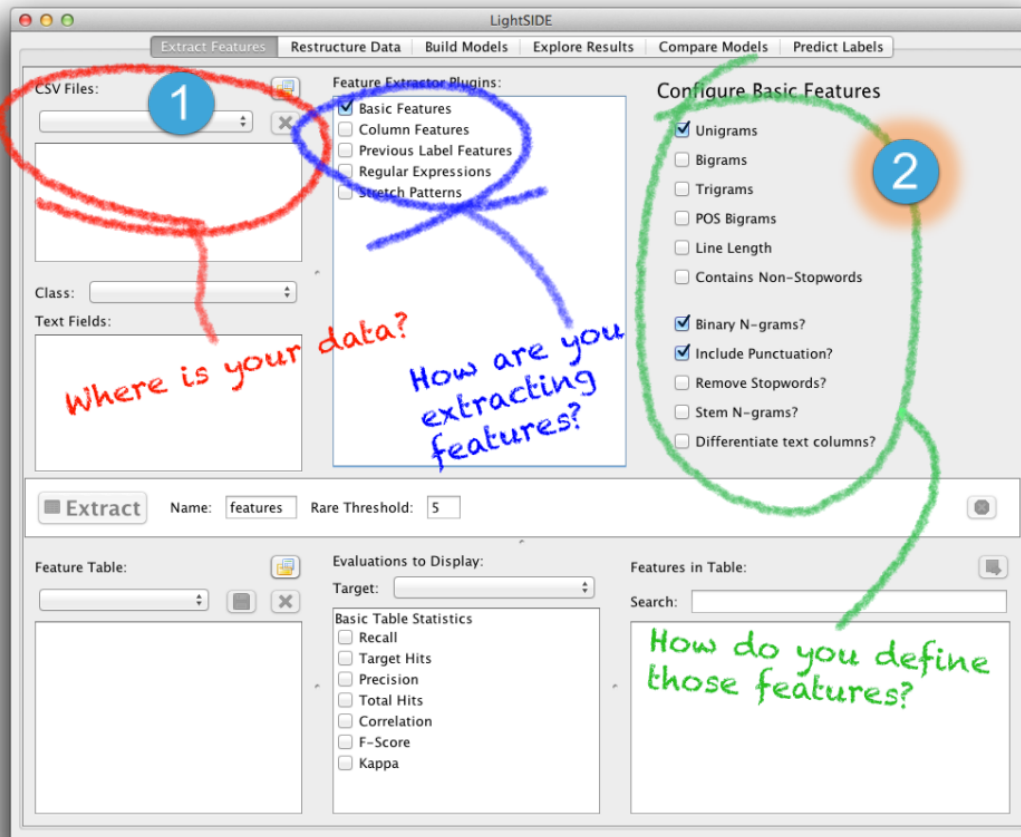
Extract Features Tab Overview

1. Select file
2. Choose features
3. Extract features
4. Table description
5. Feature list



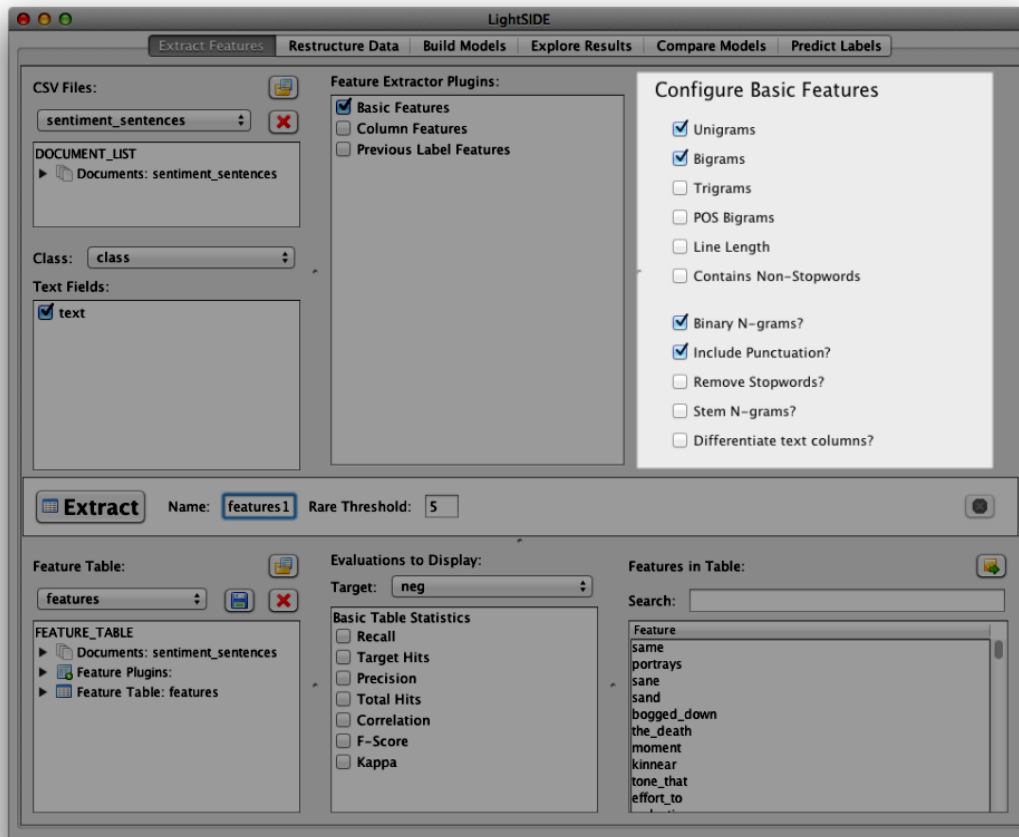
Build Models Tab Overview

1. Feature table selection
2. Choose a learning algorithm
3. Configure a learning algorithm
4. Validate settings
5. Train a model
6. Model description
7. Model performance metrics



Extracting Features

1. Select file
 - Load data (CSV file)
 - Top panel: File data associated with
 - Bottom panel: What our class value and text fields are.
2. Choose features
 - Basic feature plugin
 - Select basic features



Basic Features

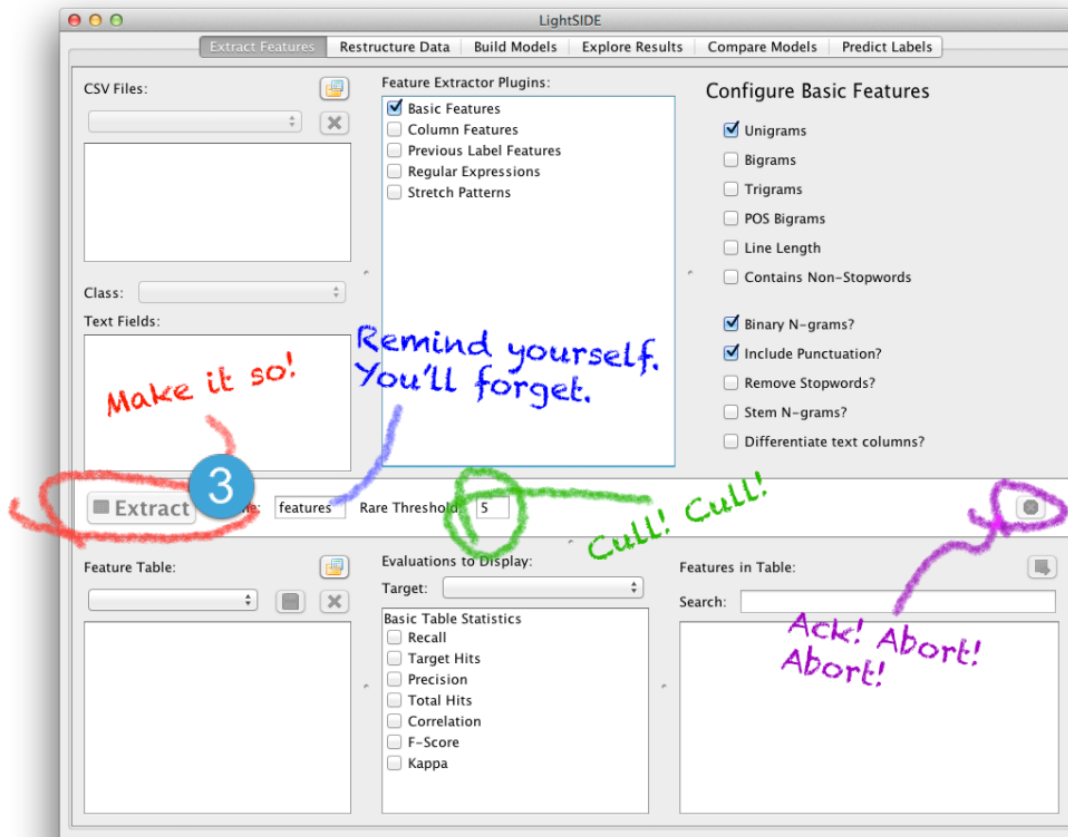
- N-Grams
- POS Bigrams
- Line Length
- Contains Non-Stopwords
- Binary N-grams?
- Include Punctuation
- Stem N-Grams?
- Differentiate Text Columns



Extracting Features

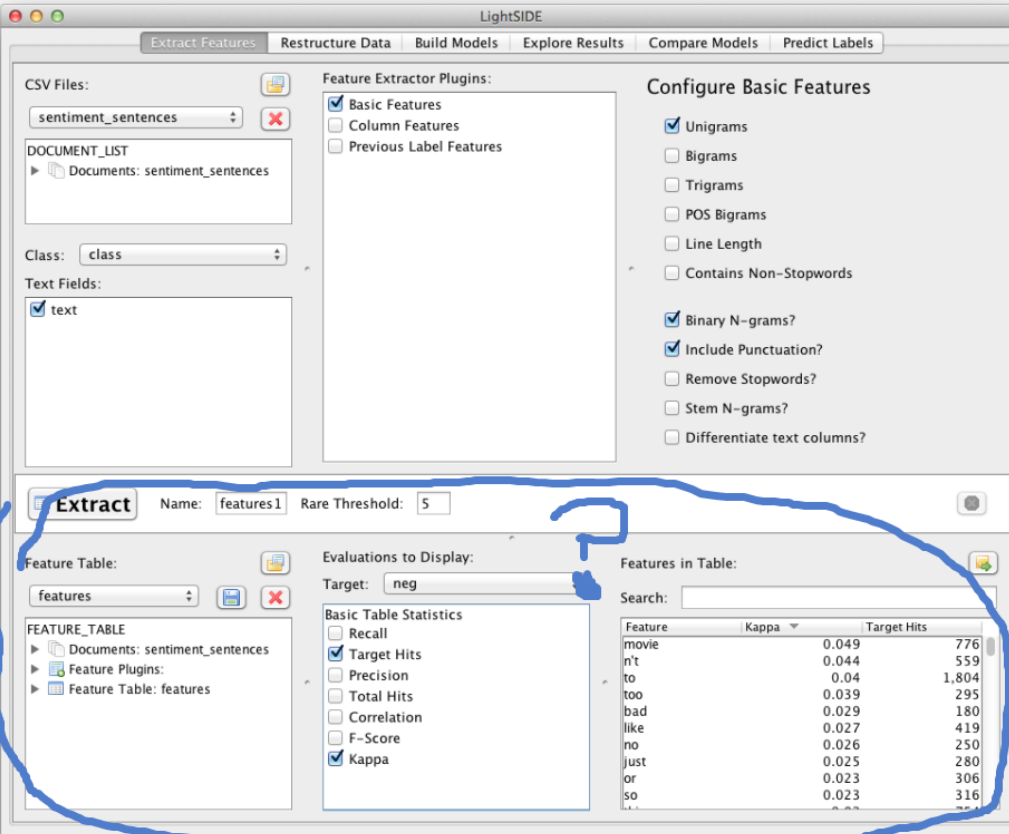
3. Extract features

- Button to make it go!
- Options:
 - Name the settings you choose
 - Rare Threshold: toss out features that don't occur at least a few times
 - Stopwords
 - Obscure words
 - Typos
- Abort mission!



Features Tables at a Glance

- We extracted the features!
- So what are we looking at and how is this meaningful?



LightSIDE

Extract Features | Restructure Data | Build Models | Explore Results | Compare Models | Predict Labels

CSV Files: sentiment_sentences

DOCUMENT_LIST
Documents: sentiment_sentences

Class: class

Text Fields:
 text

Feature Extractor Plugins:
 Basic Features
 Column Features
 Previous Label Features

Configure Basic Features:
 Unigrams
 Bigrams
 Trigrams
 POS Bigrams
 Line Length
 Contains Non-Stopwords
 Binary N-grams?
 Include Punctuation?
 Remove Stopwords?
 Stem N-grams?
 Differentiate text columns?

Extract Name: features1 Rare Threshold: 5

Feature Table: features

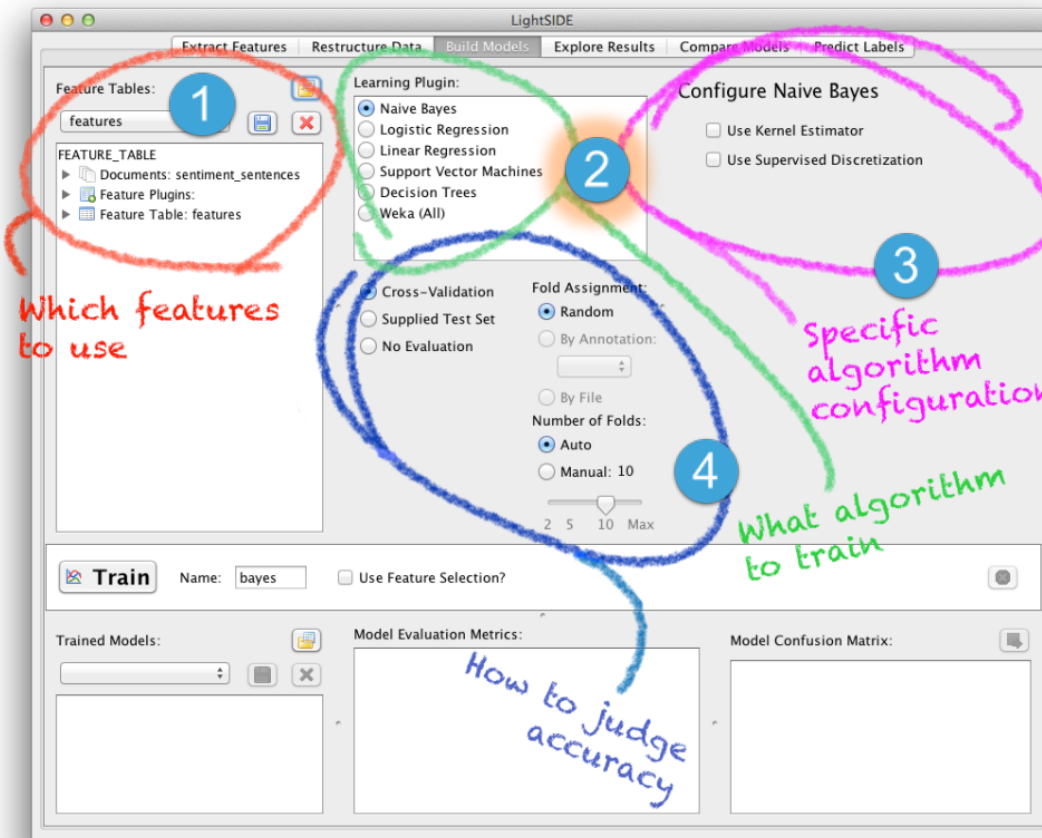
FEATURE_TABLE
Documents: sentiment_sentences
Feature Plugins:
Feature Table: features

Evaluations to Display: Target: neg

Basic Table Statistics:
 Recall
 Target Hits
 Precision
 Total Hits
 Correlation
 F-Score
 Kappa

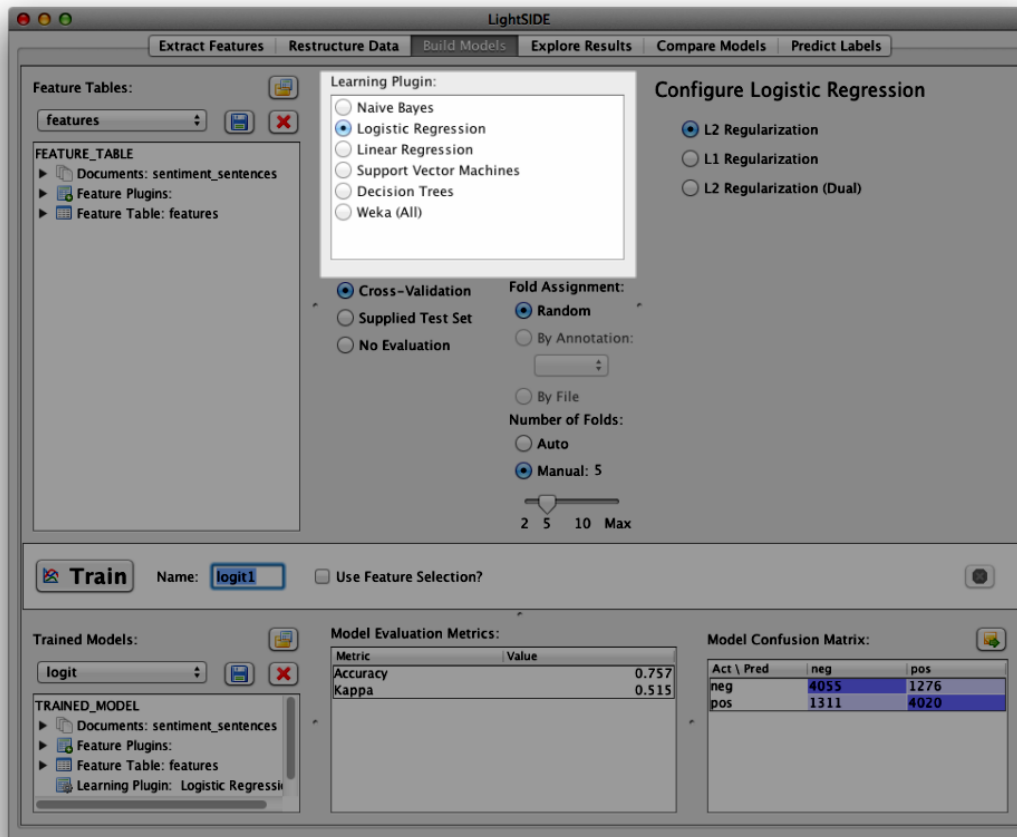
Features in Table:
Search:

Feature	Kappa	Target Hits
movie	0.049	776
n't	0.044	559
to	0.04	1,804
too	0.039	295
bad	0.029	180
like	0.027	419
no	0.026	250
just	0.025	280
or	0.023	306
so	0.023	316



Build Model Inputs

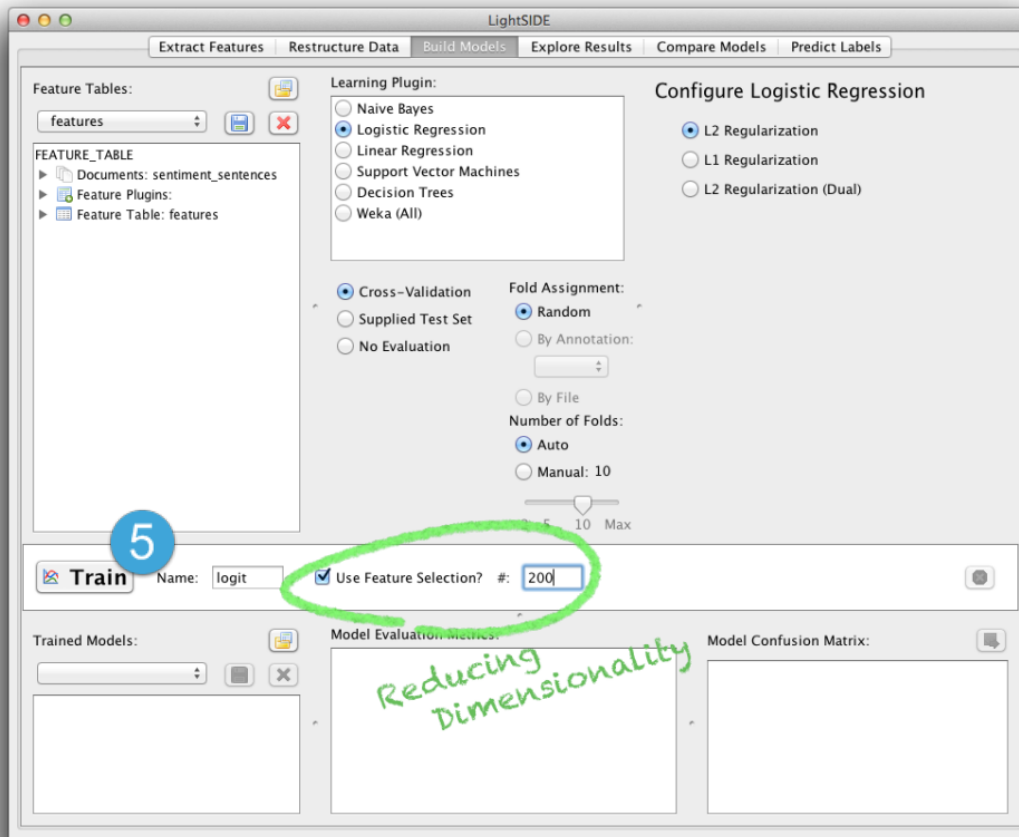
1. Feature table selection
 - Which features?
2. Choose a learning algorithm
 - Which one?
3. Configure a learning algorithm
 - Tweak parameters of algorithm, if necessary
4. Validate settings
 - For most tasks, do a standard 10-fold cross validation



Algorithms

- Naïve Bayes
- Logistic Regression
- Linear Regression
- Support Vector Machines
- Decision Trees

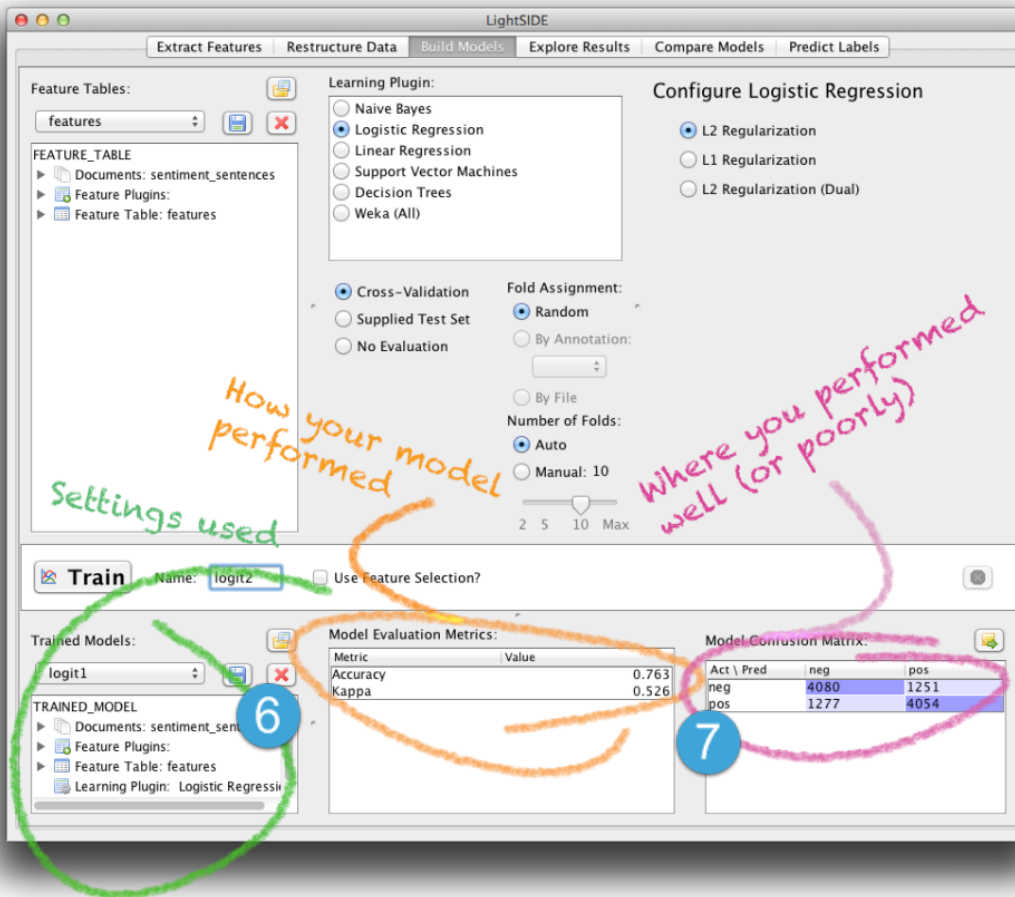




Building a Model

5. Use Feature Selection?

- This option will perform feature selection on your data by measuring each feature's chi-squared statistic against the class you're attempting to automatically recognize.
- Features below your threshold count will simply be discarded before machine learning is performed.



The screenshot shows the LightSIDE interface with the following sections:

- Feature Tables:** features
- Learning Plugin:**
 - Logistic Regression
 - Naive Bayes
 - Linear Regression
 - Support Vector Machines
 - Decision Trees
 - Weka (All)
- Configure Logistic Regression:**
 - L2 Regularization
 - L1 Regularization
 - L2 Regularization (Dual)
- Cross-Validation:**
 - Cross-Validation
 - Supplied Test Set
 - No Evaluation
- Fold Assignment:**
 - Random
 - By Annotation:
 - By File
- Number of Folds:**
 - Auto
 - Manual: 10
- Train:** Name: logit2, Use Feature Selection?
- Trained Models:** logit1 (circled with a green circle and number 6)
- Model Evaluation Metrics:**

Metric	Value
Accuracy	0.763
Kappa	0.526
- Model Confusion Matrix:**

Act \ Pred	neg	pos
neg	4080	1251
pos	1277	4054

Reading the Model Performance Summary

6. Model description

- Series of steps that got us from a set of documents to this model, for our own reference.

7. Model performance metrics

- Middle box: Summary statistics of how well the model reproduced the input labels in your testing data.
- Right box: confusion matrix.
 - number of instances that have been classified in each possible combination of actual and predicted label.
 - First bird's-eye view of error analysis.

Way to go class!

That's it! We've now created a model, based on the example data, which is able to classify new data using the labels we've selected.

We can see that the model is expected to perform at about 75.7% accuracy, which is about halfway between random guessing – a reasonable start, but certainly not quite what we'd want from an end product.

What would be next? Error Analysis. Let's not burst Tiffany and Jenn's bubble quite yet...





Error Analysis Process Assumptions

You care about specific types of mistakes.

Confusion matrices provide a coarse but effective way of finding those mistakes.

Features are the most important cause of error.

“Confusing” features are those that disproportionately appear in misclassified documents.

- Relative ranking of confusing features is more important than an absolute number

You must look at the data to understand the data.

- For the most daring individuals – go explore results in LightSide!

From the KF Post:

- [Download the tool - researcher's workbench version 2.3.1 \(Nov. 2014\)-Comes with test data](#)
- [Tutorial: Installing and Running LightSide](#)
- [Tutorial: Quick Start Guide to LightSide](#)
- [All Tutorials on LightSide and Machine Learning](#)
- [Manual - LightSide Researcher's Workbench User Manual](#)
- [Open source test data \(go to source on left menu\)](#)
- [Open source plug in repository \(go to source on left menu\)](#)