**Unit 12 – NLP**

**Introduction to NLP in Python – DataCamp**

* Genism is an open source NLP library
  + Top academic models to perform complex tasks
  + Build a word vector
  + Topic identification
* Lowercase, remove punctuation, and stop words are basic pre-processing steps
* SpaCy is similar to genism with different implementations
  + Good for creating pipelines
* Polyglot
  + Uses word vectors for many different languages
  + Over 130; can use direct translation
* Naïve Bayes is a simple classification models for working with text data
  + There is another caveat. What if we see a term that didn't exist in the training data? Instead of using zeros, we add a small negligible value called *alpha t*o each count. This is called Laplace Smoothing.
* A document can be thought of as a vector,
  + Each vector has the same length and each entry "slot" in the vector contains some kind of data about the words that appear in the document such as presence/absence (1/0), count (an integer) or some other statistic. Each vector has the same length because each document shared the same vocabulary across the full collection of documents -- this collection is called a corpus.
  + Bag of words representation because it loses text ordering
* The most common feature selection technique for text mining is the [chi-squared method.](https://nlp.stanford.edu/IR-book/html/htmledition/feature-selectionchi2-feature-selection-1.html)
* Term-Frequency X Inverse Document Frequency
  + **TF-IDF is essentially a measure of term importance, and of how discriminative a word is in a corpus.**

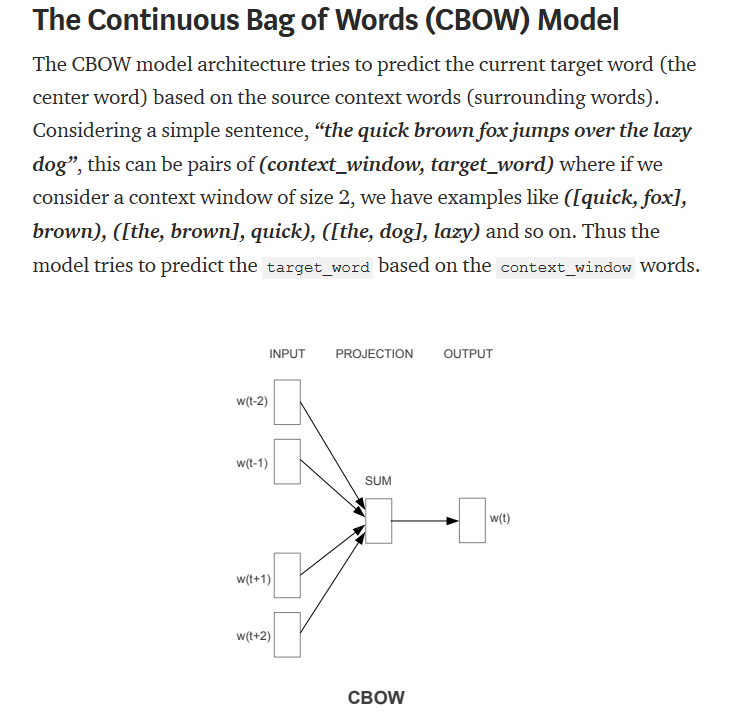
**Machine Learning with Text in Scikit-learn**

* Sparse matrix is great for data that has a lot of zeros, it just has the locations of non-zero numbers; makes computation faster
  + Dense can be better (not many zeros); dense is a normal looking matrix
* CountVectorizer will output spares matrix
* Naïve bayes probabilities are not good to use for interpretation of probabilities
  + Logistic regression is better at predicting probabilities
* CountVectorizer in sklearn:
  + N-gram\_range: tuple (min\_n, max\_n), default=(1,1)
  + 2-grams may not appear very often, so it may create a lot more noise with the signal or may not be valuable

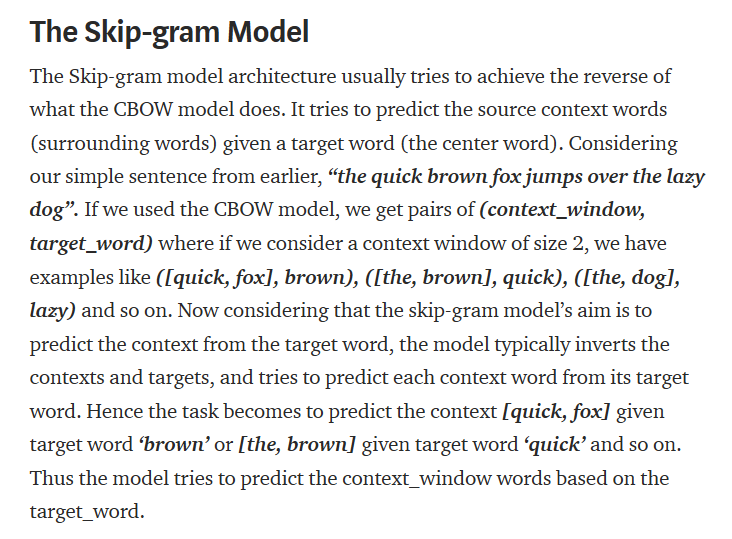
**Modern NLP in Python (Patrick Harrison)**

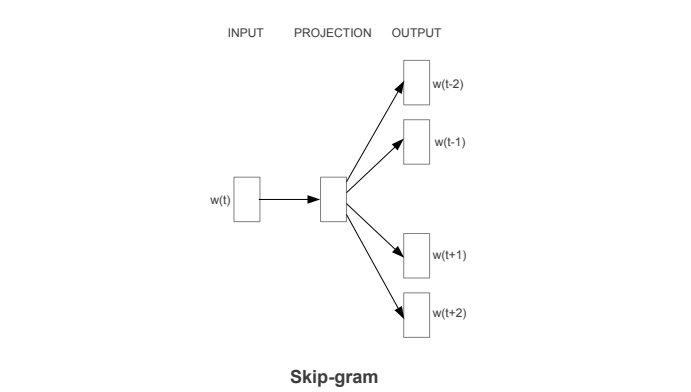
* SpaCy is the first package
* Gensim is the next package
* Gensim can group words together and transform ice cream to “ice\_cream” iterate multiple times “strawberry\_ice\_cream”
* Topic modeling using LDA
  + [Latent Dirichlet Allocation](http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf)
  + Issues: multidimensional, very sparse
  + Fully unsupervised (feed in the number of topics)
  + Assumption is that there are a number of different discernable topics
  + Can look at topics to see what tokens are in there (taco, salsa, chip, Mexican)
* Word2Vec in genism
  + Can find related word or alternate spellings of the same word
  + Can specify number of columns to describe the word and use t-SNE to reduce the dimensionality

**Continuous Bag of Words (CBOW) and the Skip-gram**



* The Word2Vec family of models are unsupervised, but you will still need to leverage a supervised, classification methodology once you have this corpus to get these embeddings.





**Word2Vec**

* We can make more robust Word2Vec models using genism
* Can average the vectors across all the words that appear in each document to get features for each document!
  + Can use this to predict or cluster!

**GloVe Model: Global Vectors for Word Representation**

