**Problem and Motivation**

* Jamie: new data scientist paid $$ to solve a problem about market segmentation
  + Her time is valuable.
  + [insert coding problem]
  + Queries stack overflow, reads through responses, but doesn’t seem to find her answer
  + Little did she know that if she had changed her query slightly to [insert], she might have gotten better search results
  + Slide: “stack over flow search is stupid”
    - Compare results resulting from slightly different queries
* To solve this problem: searchSuggester, a tool that improves your stack overflow queries by providing alternative, and often times better, search terms.

**Demo finished product**

* Make sure show to how my suggestions solve Jamie’s problem

**How do I generate my suggestions?**

* **Algorithm overview:** Solved this problem using an NLP algorithm called word2vec
  + A fully trained word2vec model encodes the meanings of words into numerical vectors that have as many as 2 or 300 dimensions.
    - Importantly words that have similar conceptual meanings to each other have similar vectors and are thus closer to each other in vector space.
    - Word2vec learns these word meanings by examining the words that tend to come before and after a target word.
      * Example: the words “create” and “add” often are used interchangeablly in the same contexts, so a word2vec model will end up learning word vectors for these words that put them close together in vector space.
* **Data:** To learn the meanings of words that a person might submit to a stack overflow query search, I trained a word2vec model on 17.5 Million question titles from a stack overflow data dump of user content (this is just a small portion of all the available data).
  + This is a critical training corpus because it contains technical words and phrases that would not be contained in a corpus of new articles, for example.
  + When working with text analysis, pre-processing the text before training your model is critical, and must be specific to your application
  + EDA – show examples of training titles, and connect with preprocessing
    - Coding: robots.txt, pandas.dataframe()
    - Remove periods?
    - Fix spelling mistakes?
    - Lemmatize? Or stem?—no:
      * Removes ‘s’ in pandas, which in some cases significantly changes the results of a query
      * Queries with ‘loop’ vs. ‘looping’ also return different results
      * Since my goal is to broaden search, my suggestions need to include any common variations that could change the results of search (ideally in a good way).
    - Titles often include stop words “how do I find out if ?, but in a search, you don’t want these words, so I removed stop words from my queries
  + Preprocessing
    - Single letters
    - Symbols
    - Stop words
    - **Show example**
* **Generating suggestions**
  + Preprocess your query in same way as training data
  + For each token in query, I find 3 alternative words closest to the original in vector space
    - Essentially finding 3 words that most commonly occur in similar contexts…have similar meanings
  + Next I form all possible combinations of words, retaining the original word order
  + A benefit of word2vec is that it can calculate probabilities of individual words, as well as phrases.
    - So for each of the new combinations of terms, I calculated its probability score…words that tend to go together will have higher scores
    - This has the effect of putting together word-combinations that make the most sense, given the training data
  + Finally, I return to the user the 5 queries with the highest probability scores
* **Validation: So does searchSuggester work?**
  + In short…yes
  + Search engines and search results are hard to validate because you can use different metrics for ‘success’ and search result relevance
  + I built searchSuggestor to broaden users’ scope of search terms in a way that allows them to choose the search path they would like to follow
  + But are my suggestions any good? Do they lead to more search results? Unique results? Results with more votes or answers?
    - Yes….
    - VISUALIZATION