**TED Talk Blue Print for Recommender**

**Approach 1: Content Based Recommendation System**

**Why?**

* **“**A **content-based recommender** looks at the characteristics of the items to determine what to recommend.*”*
* Unlike Collaborate Based Recommendation System you do not need user data rather focus on the similarity bewteen items

**How?**

* Create Count/TF-IDF Vectorizers
* Create LDA or LSA topics - (check out genism for optimal number of topics)
  + Topic Modeling
* Compare Vector “talks” with a distance metric i.e Cosine Similarity, Euclidean Distance, Jaccard Similarity to find similarity within the tf-idf vector
* Feature Engineer for more direction
* Explore Feature Importance's to see what affects the model
  + More Feature Engineering
  + ~~ROC/AUC Curves~~

**Further**

* Create a Hybrid Recommendor
* Create a Flask App
  + Drop down box?
* Presentation

~~Approach 2: Collaborate Based Recommendation System~~

~~Approach 3: Hybrid Based Recommandation System~~

**TED Talk Blue Print for View Predictability**

**Approach: Predicting if a Ted Talk Would be More or Less “Viewed/ Favorable”**

**Why?**

* I want to see if a I can categorize Ted Talks and try to predict if a given Ted Talk will be favorable or not, given how many views they can potentially receive

**How?**

* NLP:
  + Cleaning text
  + Models to consider:
    - Random Forest
    - Naive Bayes
      * Multinomial or Bernoulli
* Views as Target
  + If above average it is “favorable”, if under under mean “Not so favorable”
    - Even go further and say:
      * top 25% - very favorable - most views
      * bottom 25% - least favorable - least views
      * above mean - favorable - above avg views
      * below mean - not so favorable - below avg views
    - ~~Can if be broken apart into Topics?~~
* Other means of finding a target
  + Views to Comment Ratios, Views to Average length of Sentences, Views to Length of transcript, most common words, etc.
  + % of rating categories - flatten

**Further**