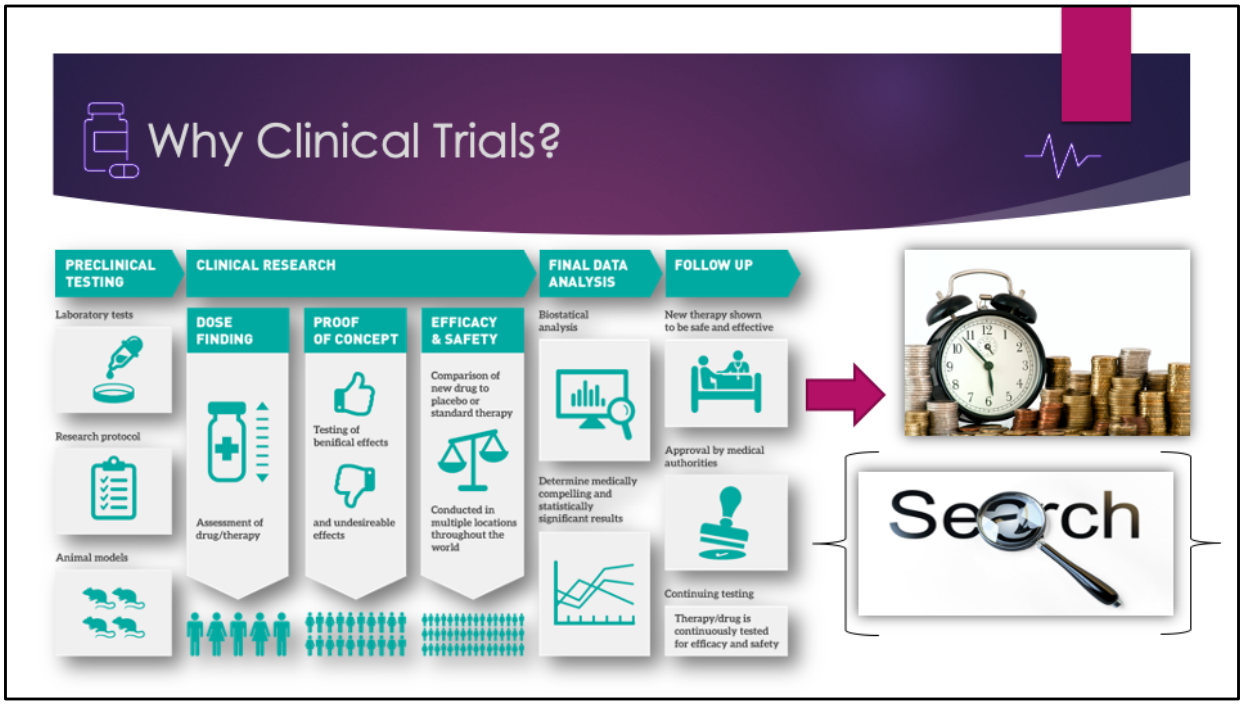
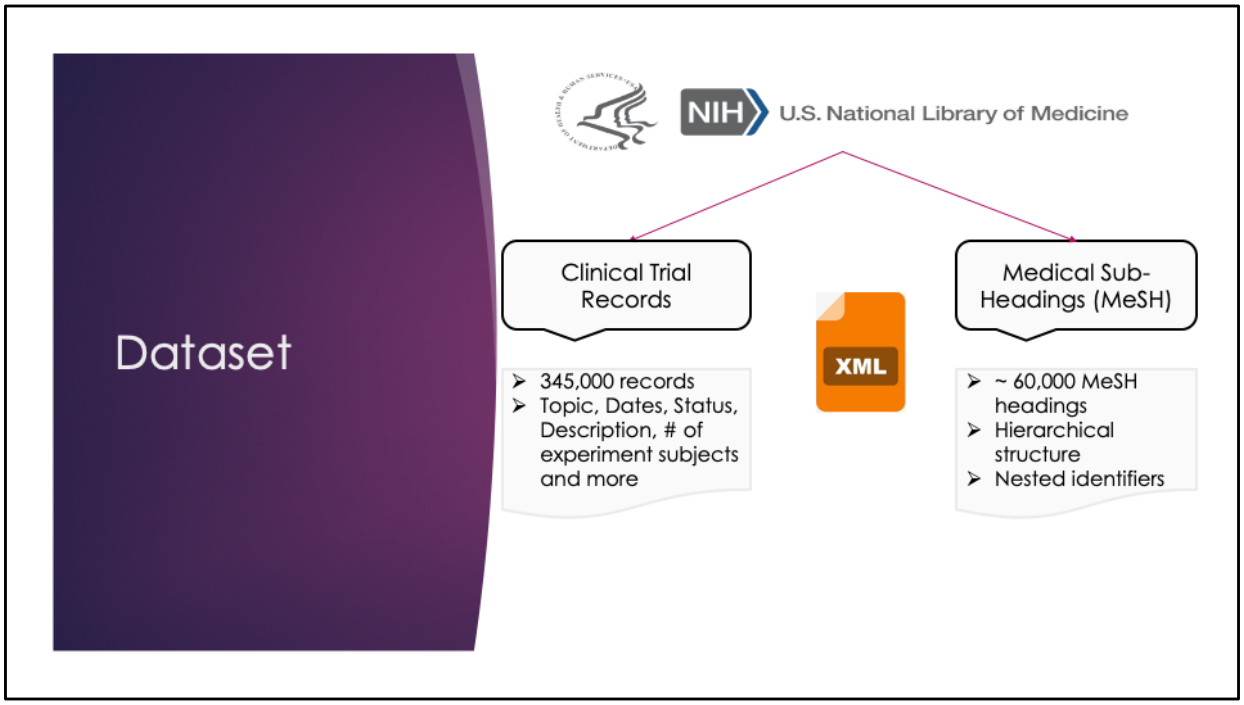

Search Optimization for Clinical Trial Records
APOORVA DORAIVELAN - AVEEK CHOUDHURY - HARSHITA VED - PRAKHAR PATIDAR



Clinical trials include but are not limited to participants receiving specific interventions (drugs, devices, procedures, etc.) according to a research plan or protocol created by investigators. These trials are documented and stored by the government as they are expensive in terms of time and money.

The study of these trials is core to almost all research in the field to gain insights as well as compare, thus the need for a search engine.Considering the sparsity of clinical trials data, often times a user interested in the trials may not be able to key in all keywords relevant to the sought studies in their query, thus resulting in a need for an engine to expand and optimize search to aid the user.



This project makes use of data from two different sources.

First, the publicly released records of global clinical trials/studies which comprises of more than 345,000 records as of August 2020. These records capture various aspects of a study ranging from when the study commenced and concluded, to the number of subjects involved in the same. The attributes also record the overall status and the detailed description of the study. The second dataset is known as the Medical Sub-Headings data (known as MeSH), which is a hierarchical taxonomy of various areas of clinical trials. It comprises of roughly 60,000 topics currently and is characterized by nested identifiers.
Both these data sources are made available publicly by the United States National Institutes of Health and are downloaded in XML format, which was then parsed to extract information. The MeSH data was used to create a Trie structure for efficient usage.

The MeSH terms were parsed through vectorizers to generate term-document matrices with counts and TF-IDF weights for use in the various learning approaches.

Market basket analysis: Apriori

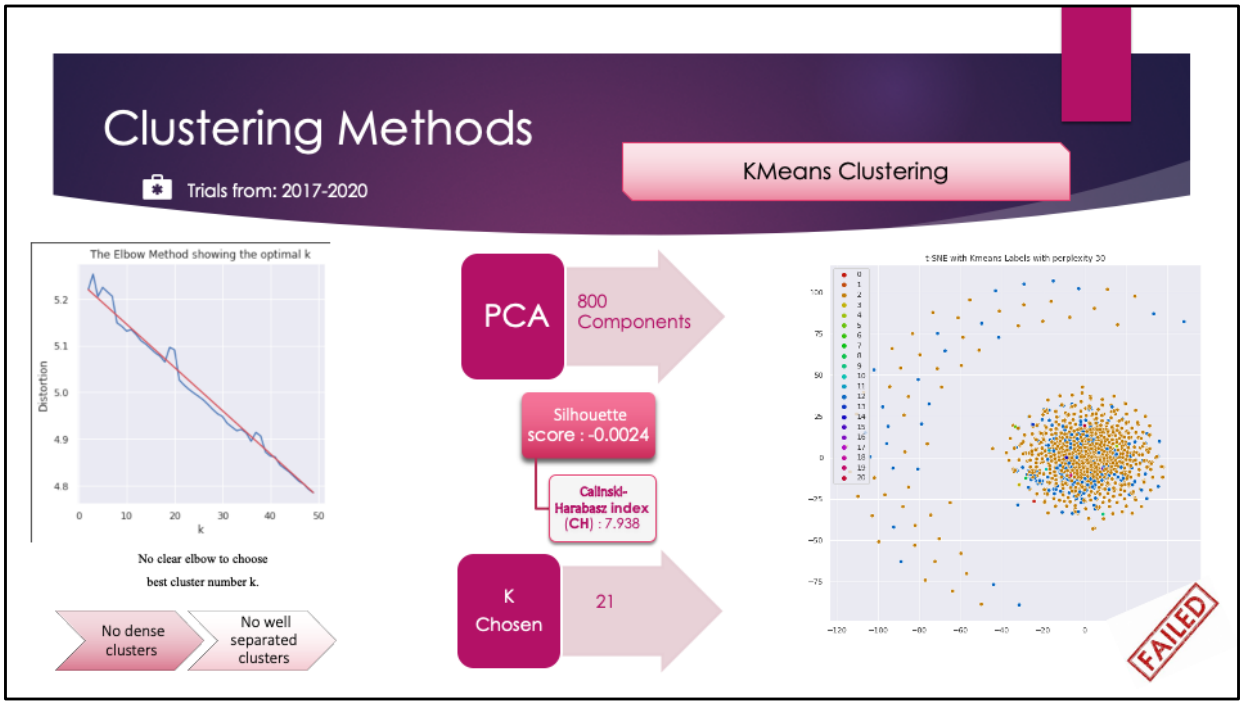
- MBA was used to identify frequently co-occurring keywords in documents.
- No interesting Association Rules found.

FAILED

antecedents	consequents	support	confidence	lift
magnetic', 'imaging'	resonance'	0.020264161	0.968693725	40.149947
resonance'	magnetic', 'imaging'	0.020264161	0.839899111	40.149947
toxicity', 'outline'	objectives', 'absence'	0.020206206	0.836291677	34.566723
'objectives', 'absence'	'toxicity', 'outline'	0.020206206	0.835189843	34.566723
'resonance', 'imaging'	'magnetic'	0.020264161	0.977358491	34.297193
'magnetic'	'resonance', 'imaging'	0.020264161	0.711104332	34.297193
'resonance'	'magnetic'	0.023170633	0.960365121	33.700866
'magnetic'	'resonance'	0.023170633	0.813097417	33.700866
'outline', 'absence'	'toxicity', 'objectives'	0.020206206	0.958487973	32.370966
'toxicity', 'objectives'	'outline', 'absence'	0.020206206	0.682423175	32.370966
'al'	'et'	0.028009922	0.94523763	29.610924
'et'	'al'	0.028009922	0.87745098	29.610924
'objectives', 'cells', 'ii'	'followed', 'outline'	0.020890081	0.805654895	27.852641
'followed', 'outline'	'objectives', 'cells', 'ii'	0.020890081	0.72219996	27.852641
'outline', 'cells'	'followed', 'objectives', 'ii'	0.020890081	0.769206146	27.722913
'followed', 'objectives', 'ii'	'outline', 'cells'	0.020890081	0.752898172	27.722913
'united'	'states'	0.023698028	0.859394704	27.531585
'states'	'united'	0.023698028	0.759190494	27.531585

Market basket analysis is the first method we used to discover keywords that frequently occur together. After the preprocessing of dataset we used the output we got from the count vectorizer to create a transaction database for all the documents.

Each row is used as transaction where the row represent a document and each feature is the MeSH keyword. Association rules are generated using Apriori with multiple combinations of minimum support, confidence and lift measures.
Observing the results, we see that we get very few interesting association rules, but even those have a very low support hence we can not classify them as interesting. Optimizing the algo specifically every time for to remove junk is not feasible considering domain specificity and the risk of losing out on important information. Hence, we label it as failure in the interest of a generalized approach.



Moving on, to find intrinsic grouping in a collection of unlabeled data we chose KMeans Clustering Algorithm.

Due to computational limitations, this approach was tested with records from 2017-2020 only.

Vocab V was used to create a *m* x *n* matrix with TF-IDF measures with row as vocab term across all documents and column as document over the vocab terms.

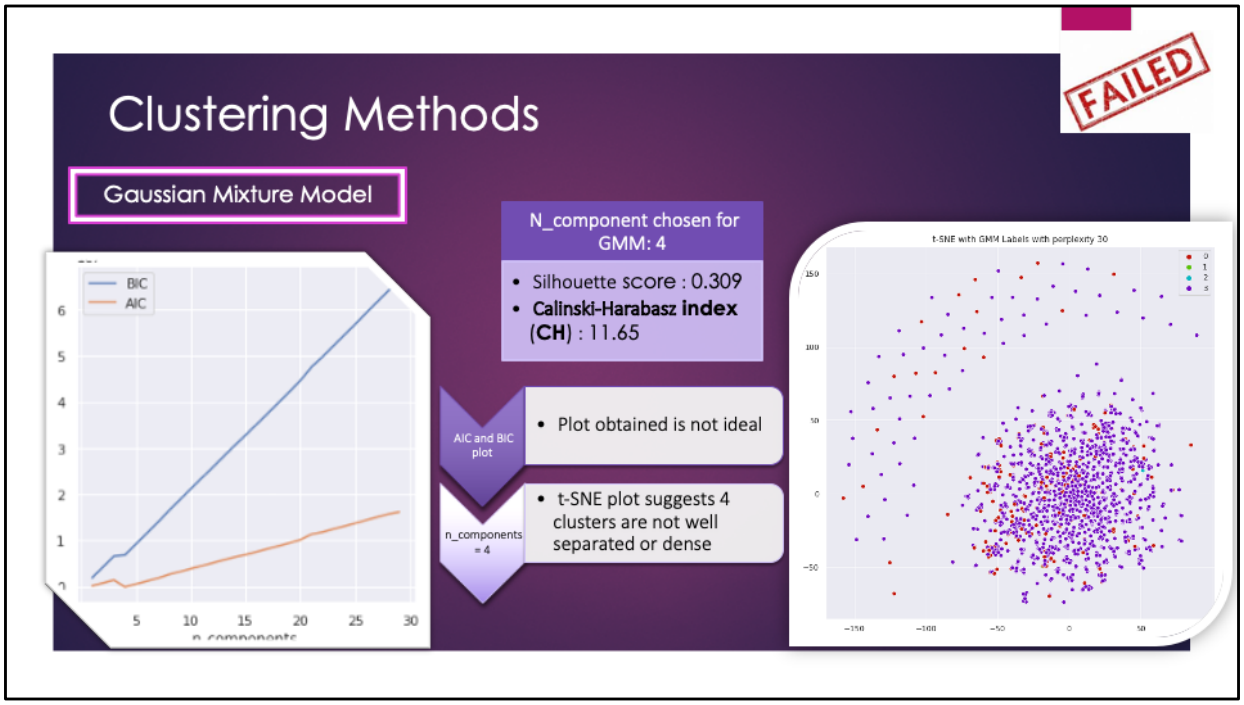
Due to the extreme sparsity of the matrix, Principal Components Analysis (PCA) was performed with the optimal number of components ~800.

The vectorized-reduced MeSH matrix obtained was then used as input for the k-means clustering algorithm. The elbow plot did not give proper elbow, still 21 was selected as optimum looking at mild elbow obtained.

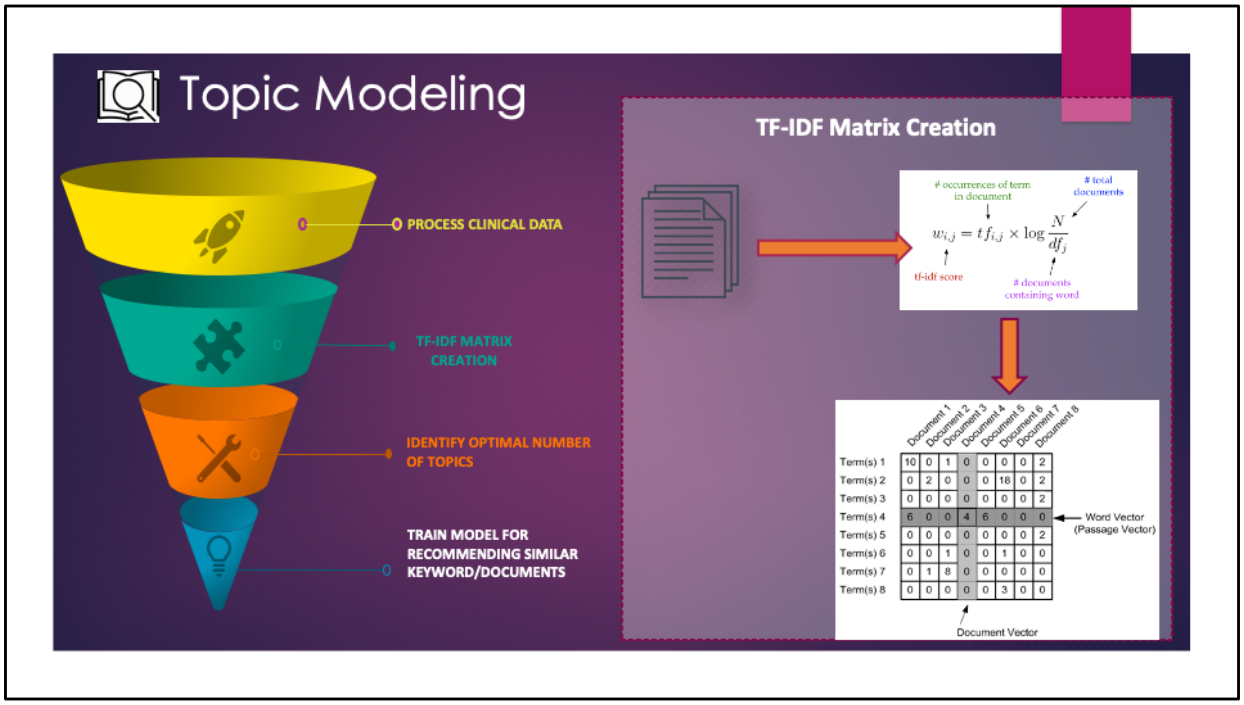
K=21 gave Negative Silhouette score I.e. -0.0024 suggesting points are much closer to its neighbor than to its assigned cluster. Also, Calinski-Harabasz index was 7.938 suggesting clusters are neither dense nor well separated.

At last we visualized the assignments using a t-Distributed Stochastic Neighbor Embedding (t-SNE) plot, we observe that the clusters weren't properly separated.

The reason for k-means algorithm failure could be because clusters are of various size and density, Also, the sparsity and dimensions of input matrix are quiet high.



After failure of KMeans Clustering and to curb its lacking flexibility in cluster shape and probabilistic cluster assignment, we explored Gaussian Mixture Models (GMM).
Analytic criteria, i.e. Akaike information criterion (AIC) and Bayesian information criterion (BIC) were used to determine the optimal number of components as a measure to score a model based on its log-likelihood and complexity.
As seen above the AIC-BIC results were not in line with our general understanding, The value of K=4 gave very low Silhouette score i.e. 0.309 suggesting that points could be in another cluster. Also, Calinski-Harabasz index was 11.65 suggesting no good cluster separation.
t-Distributed Stochastic Neighbor Embedding (t-SNE) plot confirmed that GMM was also not a good algorithm for our data and because of lack of proper clustering we moved on.

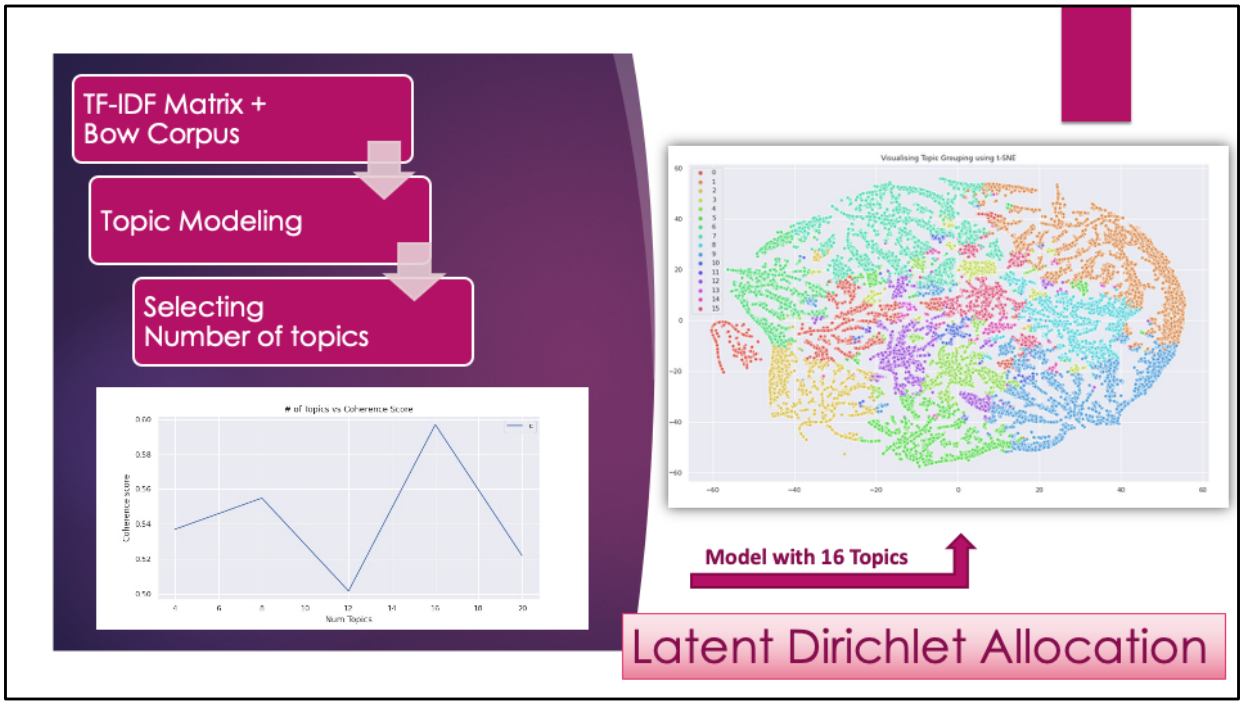


Methods like KMeans and GMM may cluster the articles but they do not label the topics. Hence, we move ahead with Topic Modeling.

Topic Modeling is an unsupervised machine learning technique that is capable of scanning a set of documents, detecting words and cluster the words and similar word groups that best characterizes the set of documents.

For model training, we have used 2018 to 2020 Clinical Trials and have pre-processed the data so that the corpus contains only words which are of significance. This gave roughly 75K records from which a bag of words (BOW) was created. Using the BOW and the processed corpus, each word in the corpus was mapped to its corresponding Id in the bag of words and the new corpus would contain only the id for each of its words.

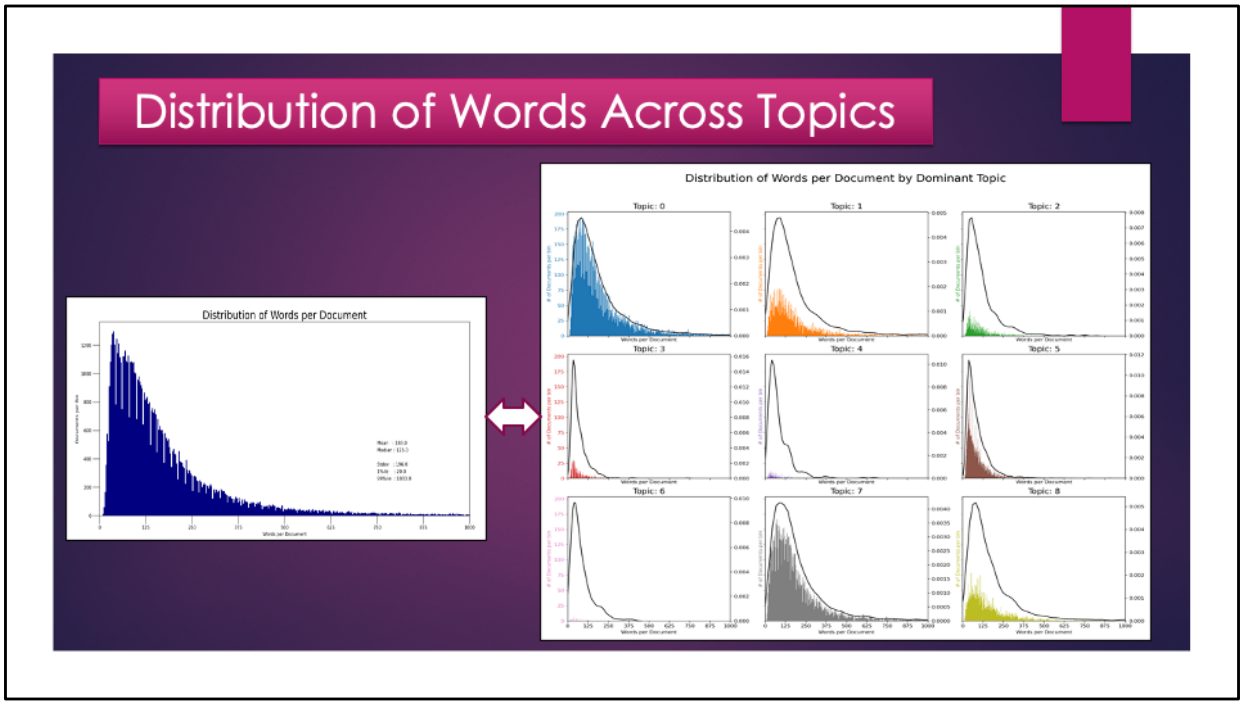
- A Term Frequency- Inverse Document Frequency Matrix, which is a reflection of how important a word is to a document, was created using Gensim's Tf-IDF model, in which each row represents a document and each column represents a keyword.



Training Latent Dirichet Allocation(LDA) model, requires a dictionary of words and a corpus. In our case, the corpus has been converted into a TF-IDF matrix and bag of words was given as an input for Gensim's LDA Multicoremodel.

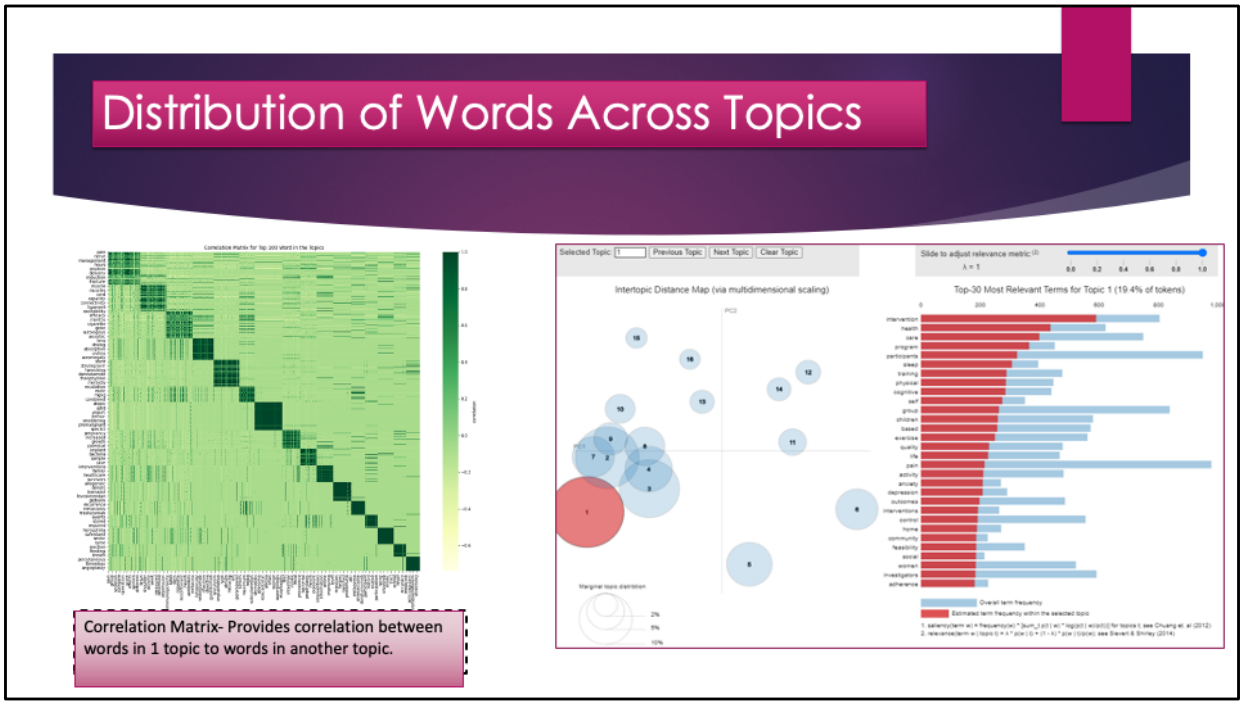
For optimizing the number of topics, LDA model was run and its coherence score (closeness metric) was calculated for each model and plotted as a graph. Here it was found that 16 topics was optimum to generate topic wise clusters.

An LDA model with 16 topics was generated. This optimal model was then visually represented using t-SNE which showed distinct clusters as shown above.



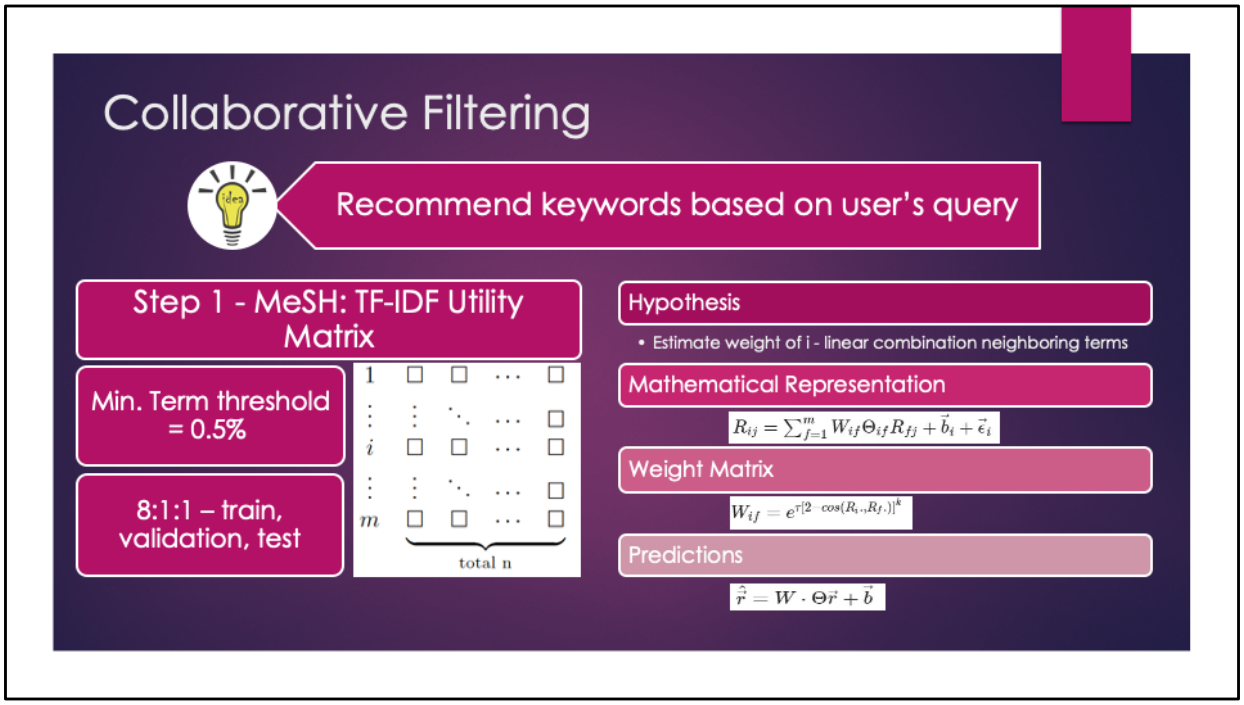
LDA model clusters words to similar topics. With 16 topics, we looked at the distribution of the words in the topic per document. Figure on the left, represents the word per document, this was split into the number of words from a topic present in each document.

Drilling down into the figure on right, Topics 0 and 7 have very high presence in the document whereas Topics 4 and 6 have very less presence.



Top 100 words from each of the topics was taken to evaluate the correlation between the words. A correlation matrix to analysis the closeness of the words within one topic, we then represented the correlation matrix as a heat map to understand how the word in a topic behave. As seen in the figure on the left, words with higher probability of occurrence within each topics are more correlated than words in the same topic with lower probability of occurrence. We can distinctly see 16 dark square which is a representation of the 16 topics in the model.

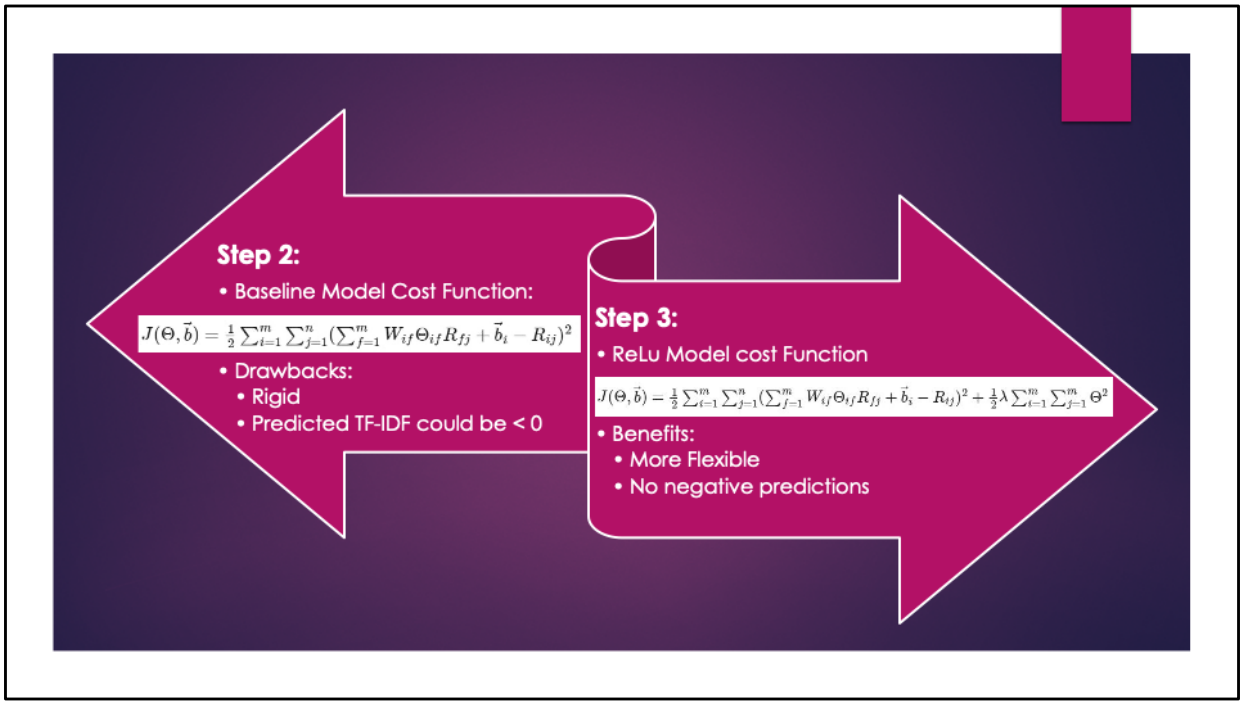
PylDavis based interactive chart was created to represent the topics. It can be observed that Topics (1,2,3,4,7,8,9) are closer compared to the other topics in the model. The bar chart on the left represents the top 30 words from the selected topics(highlighted in red), where the blue bar represents the word frequency in the dataset and red bar represents the words frequency within the topic.



The final approach taken to address the problem of optimizing the search for clinical records is using collaborative filtering. The idea is to be able to recommend associated keywords to a user, which can help enhance their search. As a first step for this approach, a Utility matrix using the MeSH terms is generated of size m x n, where m denotes the number of MeSH terms and n denotes the number of documents. Owing to computational limitation, min. frequency of 0.5% was set as threshold. This matrix is then randomly split in the ratio of 8:1:1 to separate out train, validation and test sets.

The hypothesis is that the TF-IDF measure for a term I in document j can be estimated using the linear combinations of TF-IDF measures of its neighboring MeSH terms in that particular document. This is represented using the first equation, where W_{if} is the neighbor weight matrix measuring the distance between MeSH terms *i* and *f*. This Weight matrix is defined by the second equation, which is basically the transformation of the cosine distance between *i* and *f* to a value between 0 and 1. Theta is the parameter matrix that is learnt using the algorithm.

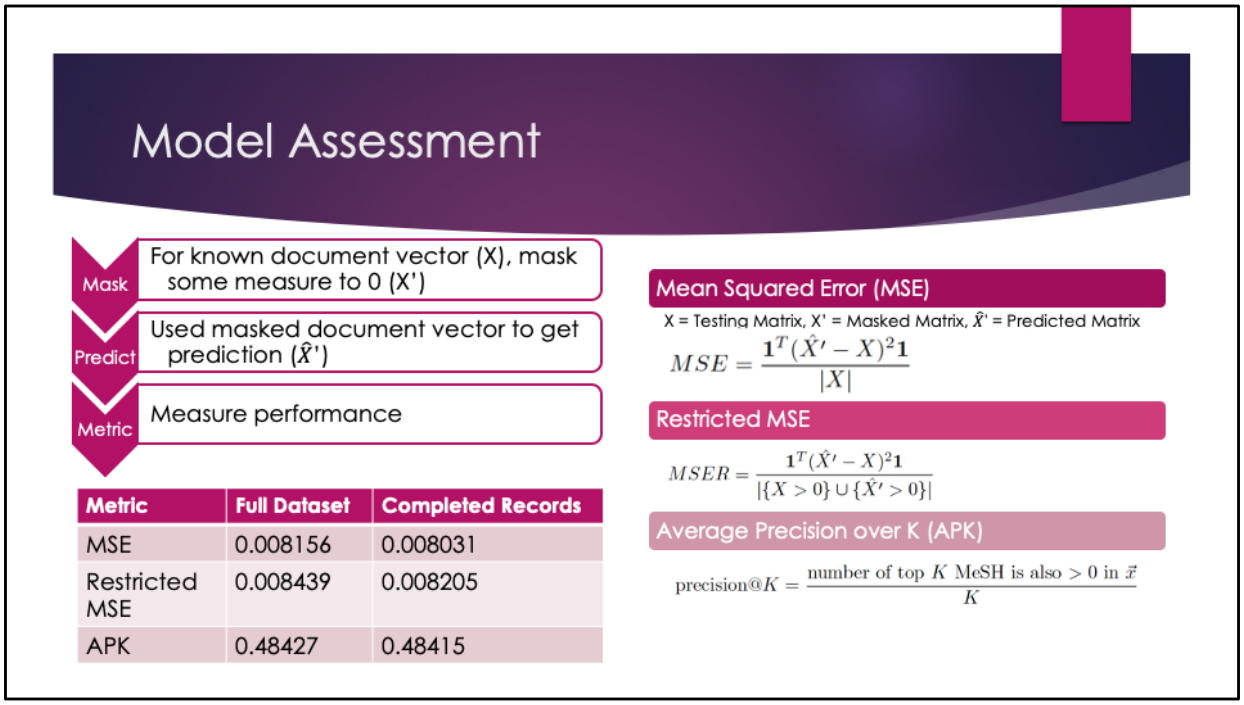
The predictions are made using the third equation that uses the weight matrix, learnt parameter matrix and baseline in their matrix representations.



Using the defined hypothesis we designed a baseline\naive model where we define cost function for the problem using the hypothesis and adding a vector ‘b’ as the baseline of the i-th mesh term and noise represented by epsilon.

We used a matrix implementation of batch gradient descent to minimize the loss and provide us with final theta values which could be used for predictions. Testing it on different datasets and going through the results it was observed that some of the predicted TF-IDF values sometimes came out to be negative which defies the definition of TF-IDF itself and also would tend to increase the loss instead of minimizing it hence making it more rigid.

To solve this problem, we added a regularization parameter to our updated ReLu model which helped us make the model flexible by constraining the magnitude of theta. Batch gradient descent was found to be very computationally expensive hence to overcome that Mini-batch gradient descent was used to train the model.

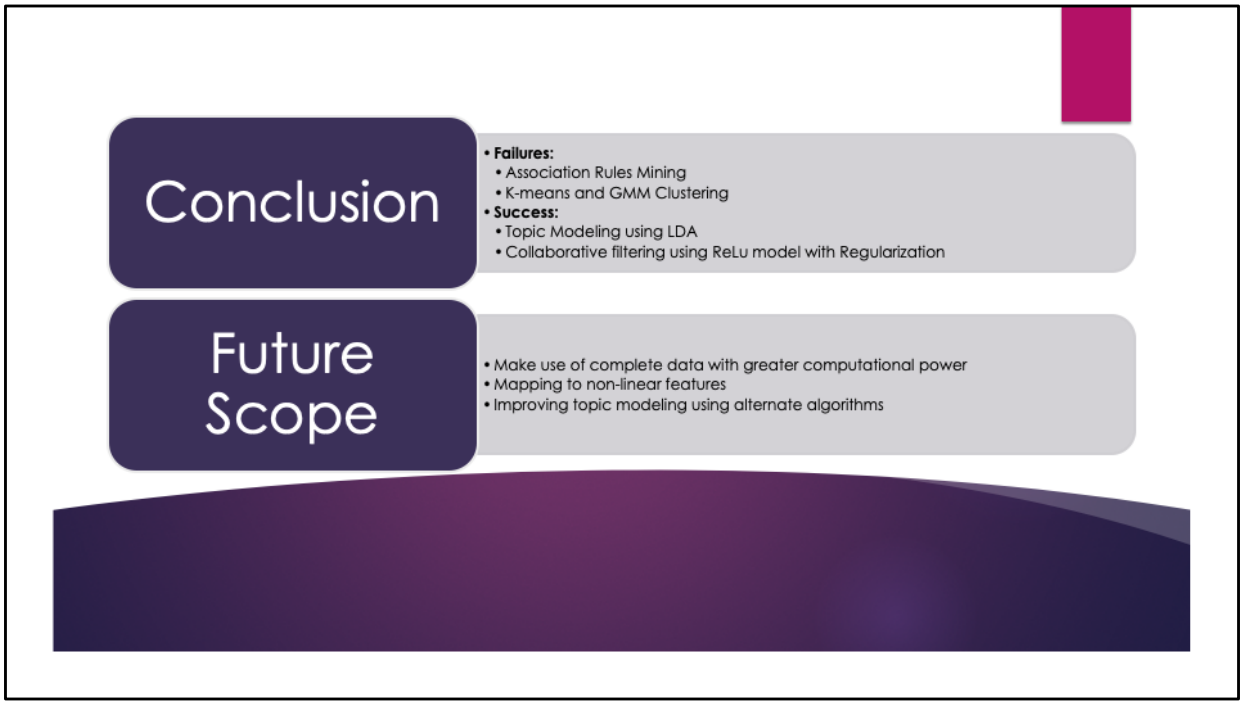


In order to validate the model, the first step is to mask a known document vector, X. The masking is done by randomly switching the TF-IDF measures of certain MeSH term with weight > 0 to 0. This masked vector, X' is then used to get predictions from the model, denoted by X^'.

The predicted vector is then evaluated using 3 metrics -

1. First– Mean Squared Error (MSE) is calculated over the total number of elements in the testing matrix. For a sparse matrix like ours, this value is usually a small number and might not reflect the actual performance.
2. This leads to the second metric – Restricted MSE or MSER which takes in the denominator elements that are not 0 either in X or X^', hence a better resolution.
3. The third metric – Average Precision at K is computed as the ratio of top K terms that were masked as 0 but were predicted > 0 and were actually > 0 in the test matrix to the value of K selected. In this case, K was chosen as 3.

The algorithm was trained on 2 compositions of the original data – one using all the records while the other using only records of studies that had status as Completed. The model performance is shown in the table and was considerably stable across the train and test. The APK score indicates that the model performs reasonably well



Through the course of this project, we had the opportunity to put to use a lot of techniques learnt in class and had first-hand experience of witnessing certain algorithms failing and some succeeding. Given the nature of our dataset, we certainly improved on our skill of handling and processing complex text data. While the algorithms like association rule mining and k-means clustering failed to provide meaningful results, topic modeling and collaborative filtering performed reasonably well. The predictions received from both this approaches can be used to make recommendations to the user in order to enhance their record search experience.

As future work, we would like to explore the usage of the complete dataset, given we have computational freedom and attempt techniques to map non-linear features. We would also like to explore on improving results from topic modeling using algorithms like LSA and matrix factorization.

Thank you for your time.