For topic modelling, we have used LDA using gemsim and sklearn libraries.

Data-prep: Data was scrubbed for special chars, less than 4 char length and numbers. Data was converted to lower case and tokenized. Stop words and frequently occurring words (freq>3) where removed. Data was them lemmatized using a WordNet lemmatize. Data was then stemmed using a Porter stemmer. This resulted in a dataset of approx. ~ 6500 unique tokens down from approx. 52,000 token words.

For topic creating, we tested various LDA models, using genism and sklearn. Models were also run using unigram and n-gram approaches. Applying n-gram algorithm (2gram) to verify increase in accuracy of topics, but gives much better results. This compromised performance, as it takes % longer to run the LDA model. Based upon perplexity and coherence factors, we chose using the 2gram model with genism.

The scrubbed dataset corpus was fed into the genism LDA algorithm. The model was tuned to look for ranges of topics (between 5 and 30) and number of passes (between 5 and 30) to check for perplexity and concurrence. We ended up using an LDA model with 25 topics and 20 iterations with default thea and beta hyper parameter values. The final model has coherence values of 48% with perplexity of -7.27.

Here are results of top 7 topics and their top terms -

**Topic:21**, '0.082\*"model" + 0.023\*"variabl" + 0.021\*"latent\_variabl" + ' '0.019\*"mixtur\_model" + 0.018\*"mixtur" + 0.017\*"latent" + ' '0.016\*"maximum\_likelihood"

**Topic:24**, '0.042\*"machin\_learn" + 0.032\*"algorithm" + 0.031\*"cluster" + ' '0.019\*"learn" + 0.017\*"machin" + 0.012\*"problem" + ' '0.010\*"method"'),

**Topic:16**, '0.026\*"infer" + 0.024\*"approxim" + 0.020\*"tree" + ' '0.019\*"algorithm" + 0.018\*"variat" + 0.018\*"approxim\_infer" + ' '0.017\*"belief\_propag"'),

**Topic:1**, '0.079\*"network" + 0.016\*"neural\_network" + 0.015\*"neural" + ' '0.014\*"hidden" + 0.014\*"learn" + 0.013\*"dynam\_system" + ' '0.013\*"unit"'),

**Topic:15**, '0.023\*"cost\_function" + 0.015\*"constraint" + ' '0.011\*"much\_faster" + 0.010\*"cost" + 0.010\*"causal" + ' '0.010\*"claus" + 0.010\*"solut"'),

**Topic:18**, '0.020\*"model" + 0.019\*"system" + 0.018\*"neuron" + ' '0.018\*"neural" + 0.010\*"input" + 0.010\*"activ" + ' '0.010\*"respons"'),

**Topic:17**, '0.032\*"algorithm" + 0.028\*"bound" + 0.019\*"function" + ' '0.016\*"lower\_bound" + 0.015\*"loss" + 0.014\*"onlin\_learn" + ' '0.013\*"learn"'),

Most frequent topic terms from the 2gram model:

'algorithm', 'input', 'method', 'network', 'neural', 'neural\_network', 'system', 'approxim', 'visual', 'activ', 'brain', 'converg', 'memori', 'model', 'time', 'tree', 'bayesian', 'data'

Clustering topics by MDS, gives us the following classifications of topics –