**Required libraries:**

Environment – python3 and jupyter-lab

1. Pandas
2. Matplotlib
3. Seaborn
4. nltk
5. sklearn
6. wordcloud
7. random
8. yaml
9. nlpAUG
10. gensim
11. numpy
12. os

**How to execute the scripts:**

**EDA and dataset split into train test**

1. Create data\_generated folder in working directory where all scripts are there and train, test and author generated test data will be kept. “final\_master\_dataset.csv” must be in the same folder.
2. Change values of parameters in data\_config.yaml file
   1. author\_data\_choice : (bool) to consider author generated test data for testing or not
   2. test\_percent : (int) percentage of test data during train test split of the data (coming from all sources except author)
   3. data\_augmentation : (bool) to consider augmented train data for training or not
3. Run EDA.ipynb to get and visualize different insights on data and generate train, test, author test data split of the dataset.   
   Files gets generated in data\_generated folder : train dataset, test dataset and test data generated by author

**Data Augmentation**

1. If data augmentation in needed, run Data\_Augmentation.ipynb file. In this case, we need to keep data\_augmentation to be True in data\_config.yaml

**Classification using tf-idf**

We are using two classifiers , SVM (Support Vector Machine) classifier and MLP (Multilayer perceptron or simple neural network with n hidden layers)

1. Change values of parameters in SVM\_classifier\_config.yaml file
   1. kernel\_name\_SVM : kernel used to build SVM like linear, poly, rbf, sigmoid, precomputed
   2. balanced\_SVM : True or False , True if we want to keep class weights in order to to balance the classes depending on number of instances of classes present
2. Change values of parameters in MLP\_classifier\_config.yaml file
   1. hidden\_layer\_sizes\_MLP : tuple giving number of neurons in each hidden layer in sequence
   2. learning\_rate\_init\_MLP : learning rate used in algorithm to learn the MLP
   3. max\_iter\_MLP : maximum number of iterations used in learning MLP
3. Change values of parameters in test\_config.yaml file
   1. row\_from\_author\_data : list of rows on which we would like to test our tf-idf based classifiers. It should be between 0 to 248.
   2. test\_question : question given by user to classify using tf-idf classifier
4. Run model\_tfidf\_final.ipynb with all parameters set before. We get train and test accuracy for both the classifiers. If we have kept author\_data\_choice param True then we will also get author test data accuracy. It will also print the predictions of questions given in test\_config.yaml file.

**Classification using word2vec embeddings**

We are using two classifiers , SVM (Support Vector Machine) classifier and MLP (Multilayer perceptron or simple neural network with n hidden layers)

1. Change values of parameters in SVM\_classifier\_config.yaml file
   1. kernel\_name\_SVM : kernel used to build SVM like linear, poly, rbf, sigmoid, precomputed
   2. balanced\_SVM : True or False , True if we want to keep class weights in order to balance the classes depending on number of instances of classes present
2. Change values of parameters in MLP\_classifier\_config.yaml file
   1. hidden\_layer\_sizes\_MLP : tuple giving number of neurons in each hidden layer in sequence
   2. learning\_rate\_init\_MLP : learning rate used in algorithm to learn the MLP
   3. max\_iter\_MLP : maximum number of iterations used in learning MLP
3. Change values of parameters in test\_config.yaml file
   1. row\_from\_author\_data : list of rows on which we would like to test our word2vec based classifiers. It should be between 0 to 248
   2. test\_question : question given by user to classify using word2vec classifier
4. Change values of parameters in word2vec\_config.yaml file
   1. vector\_size : The number of dimensions of the embeddings. Preferred 100 to 800.
   2. window : The maximum distance between a target word and words around the target word. Preferred 3 to 5.
   3. min\_count : The minimum count of words to consider when training the model; words with occurrence less than this count will be ignored. Preferred 3 to 5.
   4. sg : 0 – use CBOW continuous bag of words model, 1 – use skip gram model to train word2vec model
   5. train\_new\_word2vec : if word2vec model needs to be trained again or not. Use false only when it is already trained with required data and vector size and model is saved.
   6. tsne\_word : word for which we need to visualize word2vec similarities using t-SNE plot.
5. Run model\_word2vec\_final.ipynb with all parameters set before. We get train and test accuracy for both the classifiers. If we have kept author\_data\_choice param True then we will also get author test data accuracy. It will also print the predictions of questions given in test\_config.yaml file.