**Task A:**

=======

This is a text classification task where we have 4 classes (class labels: 0,1,2,3). Kindly find attached 3 files:

1. train\_set.txt tab separated file contains training data (text, class label)
2. test\_data.txt test data (only text without the labels)
3. sample\_result.txt for result format (the candidate should send the results in the same format)

It is required to train a classifier on the training data, then test on testing data and send back the results of test data classification in the same format as sample\_result.txt.  (Hint: you should split the given training data into train, validate and test sets)

The required approaches for this text classification task is the traditional approach: Using Python (scikit-learn) and term frequency, doc frequency, bag-of-word features. In few and clear sentences, it is required to elaborate and explain each of the following:

                a) Number of features used, and why

                b) The choice of ngram features (unigram and/or bigram, ...) and why

                c) How the stop words are handled

                d) The choice of min and max of doc frequency and why

                e) How the trained model is evaluated (train, evaluate datasets) (recall, precision, confusion , ...)

 References:

<https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html#sklearn.feature_extraction.text.TfidfVectorizer>

**Task B:**

=======

This is a sentiment analysis task. It is required to train a deep learning model to classify positive and negative sentiment of the well-known IMDB dataset using Deep Learning (Keras or TensorFlow). It is required to elaborate and explain, in few and clear sentences, each of the following:

1. Design the deep network architecture and explain why (for example, lstm, cnn, fully connected, …)
2. Use word embedding and explain why
3. Explain any other hyperparameters (optimizer, learning rate, number of features, ...)
4. How to diagnose and deal with Overfitting
5. Report detailed results on the test data (recall, precision, confusion, ...) and mention how  to enhance the results briefly.
6. Explain why deep learning could perform better than traditional machine learning approaches

In addition to the questions and the results required above, provide the source code for both tasks A and B.