Instruction for Running Binary Classification -

Preprocessing - Inorder to parse the dataset, you'd first need to run the data_prepration.py which labels the tweet with 0 or 1 (Make sure the dataset is in the same directory).
After that run data_extraction.py which will preprocess the data and save it in the folder named "Dump". Finally run the command data_word2vec.py which will finally convert the tweets into
300 dimension vector and save them in the folder "Data"

Classification - You can run clf.py, which will train all the classifiers mentioned in the report and classify
the data into positive/negative sentiment. To run neural network classifier, run 4_layer_nn.py , make sure keras and tensorflow
are installed in the device.

P.S - You would need to also download the word2vec and emoji2vec pretrained models to the "Dump" folder in the "Preprocessing" directory which can be downloaded using the following links -
1. Word2vec = https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit
2. Emoji2vec = https://github.com/uclmr/emoji2vec/blob/master/pre-trained/emoji2vec.bin

Instruction for Running 3-class classification -

- **Preprocessing:** Run the *create_training_data_text.py* to clean and pre-process the data. The main data file is *myData3.csv*. This script will generate a file called *myCleanData3.csv* which is used for converting tweets to vectors. These data files need to be in the same directory.
- **Converting tweets to vectors:** Run the *create_tweet_vecs.py* to create vectors of tweets. This will require the word2vec and emoji2vec pre-trained models and also the library of emoji2vec. This script will generate *tweet_vecs.txt* and *tweet_labels.txt.* The data files need to be in the same directory.
- The *create_onehot.py* file will create one-hot encoded labels and will save it in the file *tweet_label_onehot.txt.* The data files need to be in the same directory.
- **ELM_classifier:** The *classifier_ELM.py* will run the ELM classifier and will give the classification accuracy. The data files need to be in the same directory.
- **Other classifiers:** The *training_classifiers.py* loads the data, visualizes it, standardize it and then trains a lot of other classifiers and gives their respective accuracy. The data files need to be in the same directory.