Utilizing Neural Networks and Linguistic Metadata for Early Detection of Depression Indications in Text Sequences

In this paper author is using reddit social media post which is published as CLEF2017 Dataset to predict early depression by applying neural networks, linguistic metadata features and post messages text sequences. In all existing technologies linguistic metadata features were not available and this features can be used to predict early depression from user’s social media post. In propose paper author is saying user’s language usage in social media post can be used to detect early depression and author is using CNN (convolution neural network) and machine learning models with linguistic metadata features as input. Author added an extra formula called ERDE (Early Risk Depression) to calculate depression risk score.

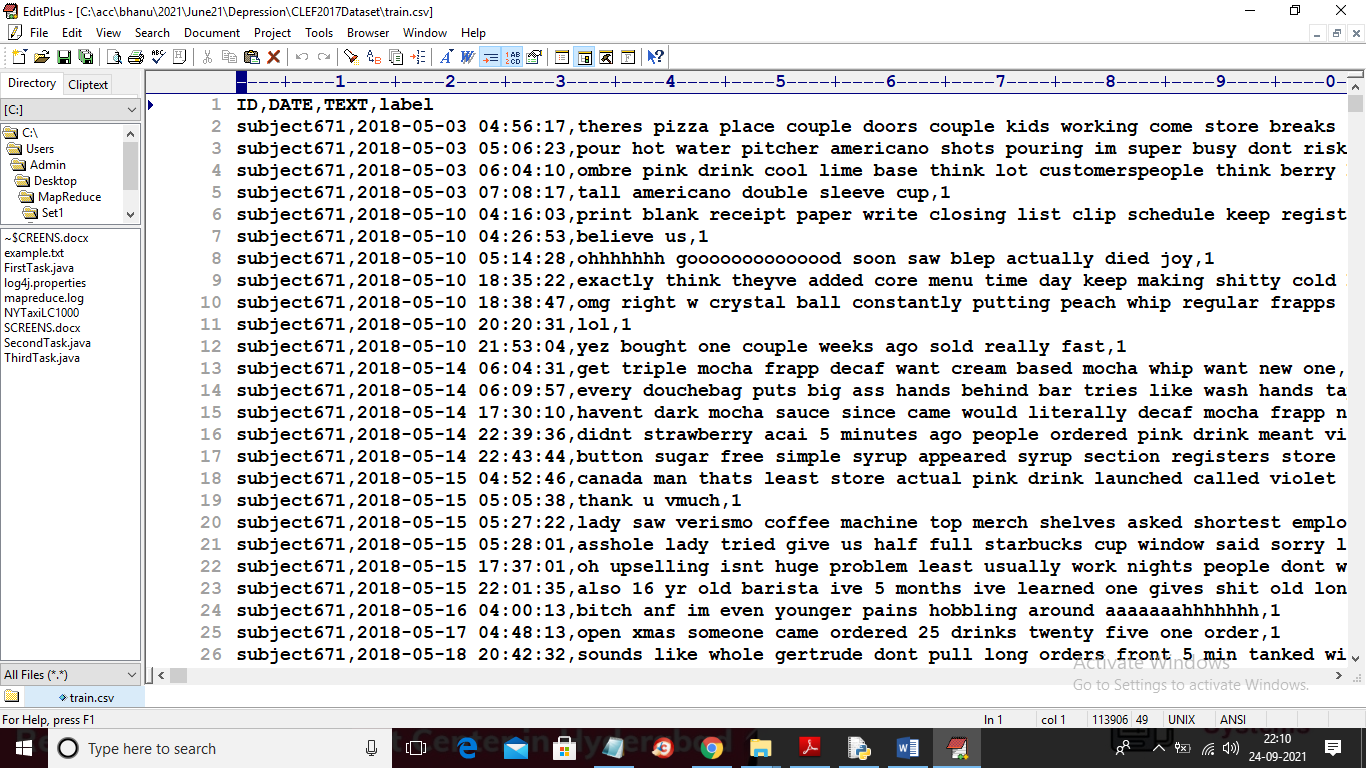
To detect depression author has extracted following linguistic features

1. I count: average count of ‘I’ words in user post (author saying depressed users will use More I words)
2. I Count title: average count of ‘I’ words in title
3. Possessive Pronouns: here we will apply NLTK POS tagger on user’s post to find average of Possessive pronouns
4. Personal Pronouns: here we will apply NLTK POS tagger on user’s post to find average of Personal pronouns
5. Past Tense Verbs: here we will apply NLTK POS tagger on user’s post to find average of past tense verbs
6. 4 readable scores: FOG, FRE, LWF and DRC (this formulas apply for text sentences to extract readability scores which indicates level depression in user
7. Month: finding average of months in which user post messages
8. Title length: find length of all posts titles sent by user
9. Text length: find length of all posts messages sent by user
10. Depression: find sum of depression words in all messages sent by user
11. Anxiety: find sum of anxiety words in all messages sent by user
12. Therapist: find sum of therapist words in all messages sent by user
13. Diagnosis: find sum of diagnosis words in all messages sent by user
14. Anti-Depression: find sum of anti-depression words in all messages sent by user

Now author has used many word embedding algorithms such as WORD2VEC, Glove and FASTTEXT. Word embedding algorithms takes users post messages as input and then build a vector model. This vector model contains average frequency of each words available in user post. In this project we are using WORD2VEC and Glove algorithm with logistic regression and CNN.

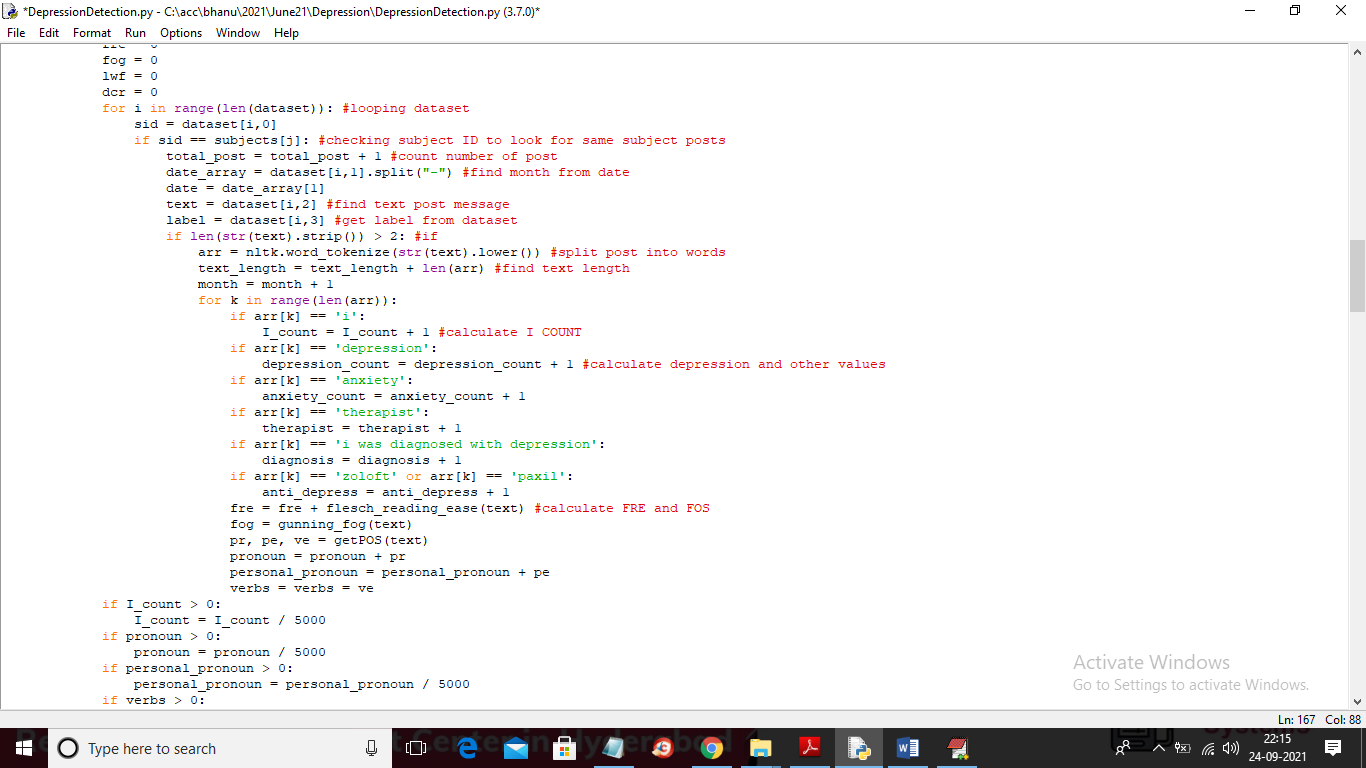
Word embedding vector and Linguistic metadata features will be concatenated and then input to CNN and Logistic regression model to build a depression detection model. This model will be applied on test data to calculate F1SCORE, Precision, Recall and accuracy. In propose model author has calculated ERDE score by diving number depression messages with 5 posts and 50 posts and the output will be consider as depression detection score.

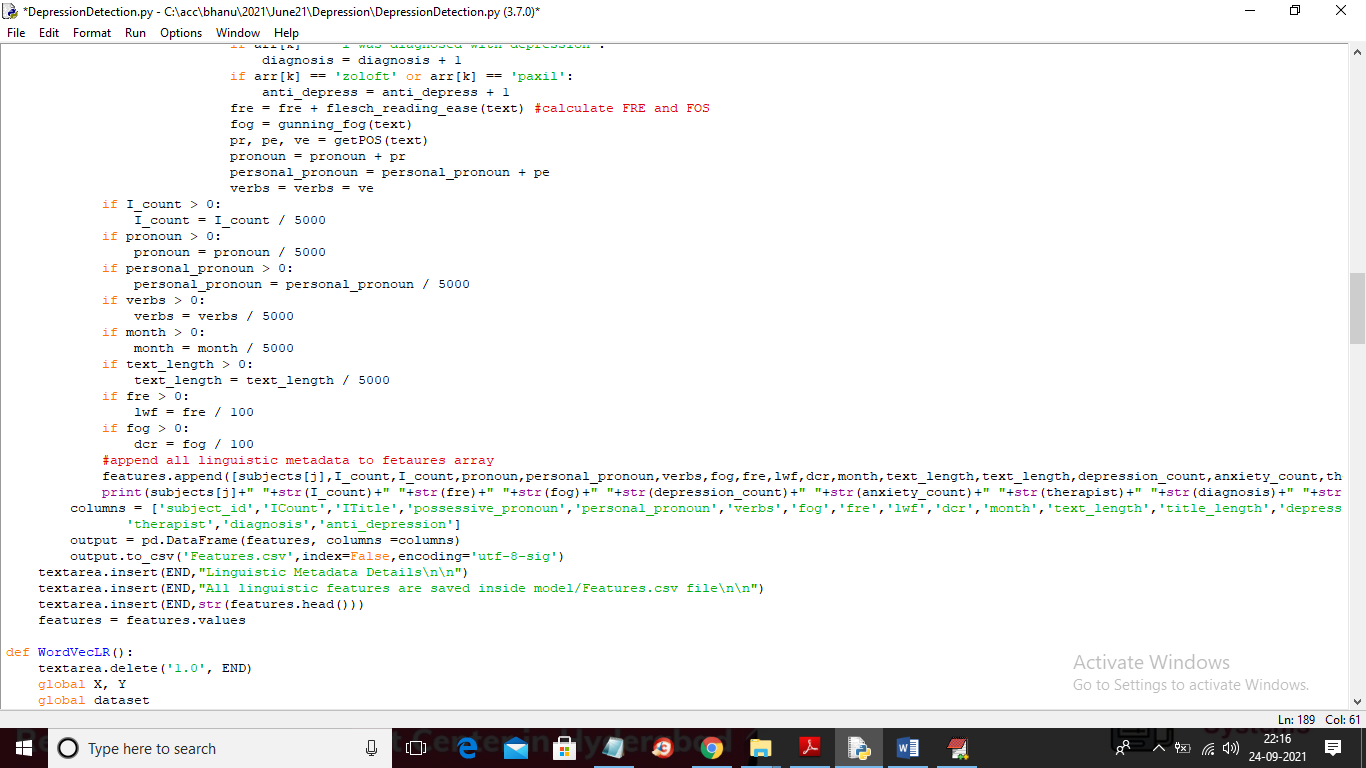
In propose paper author has used CLEF2017 Dataset and we also used same dataset and below screen showing details of dataset



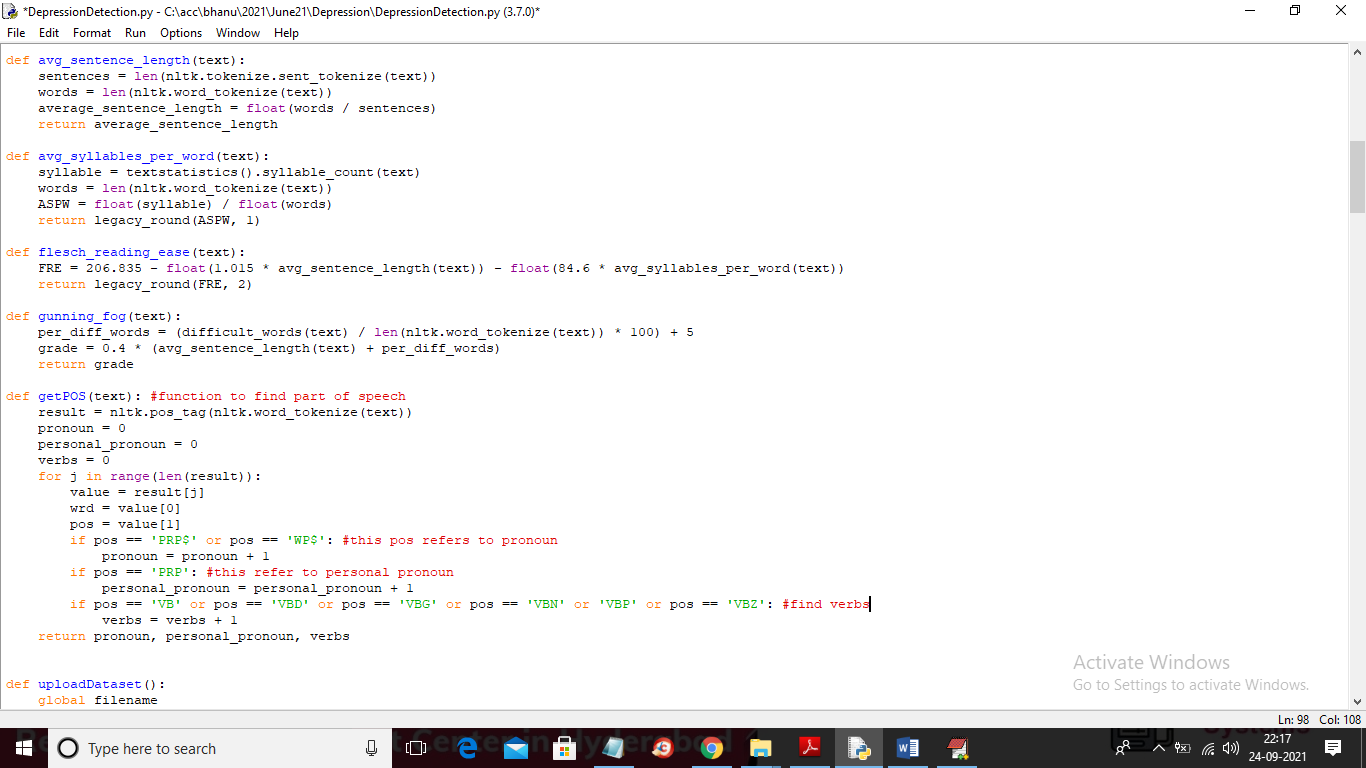
In above screen first row contains dataset columns and remaining rows contains dataset values and in above dataset we can see same person with same ID has sent multiple messages and all this messages are used to calculate Linguistic Metadata. In above dataset we have TEXT messages and class label as 0 or 1 where 0 means post contains no depression and 1 means post contains depression sentences.

Below screen code with red colour comments you can see calculation of Linguistic Metadata

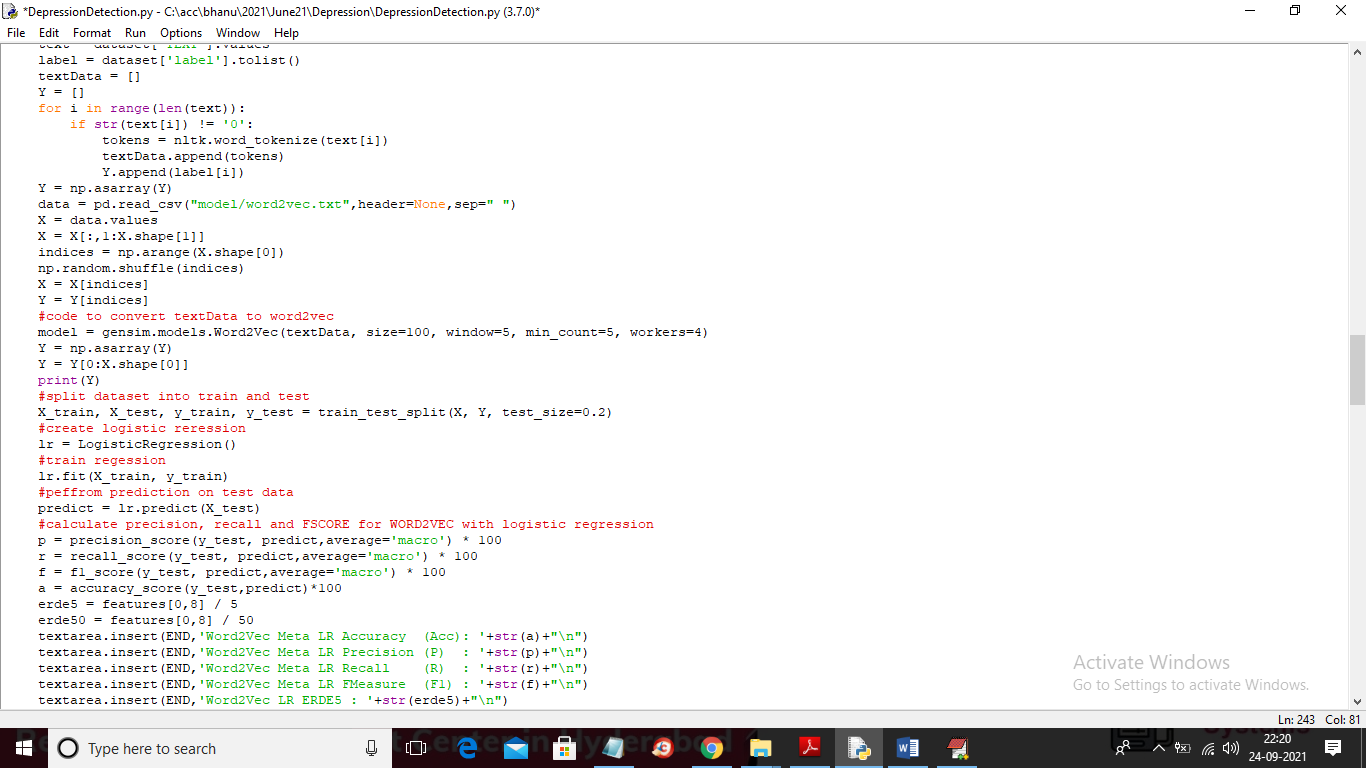




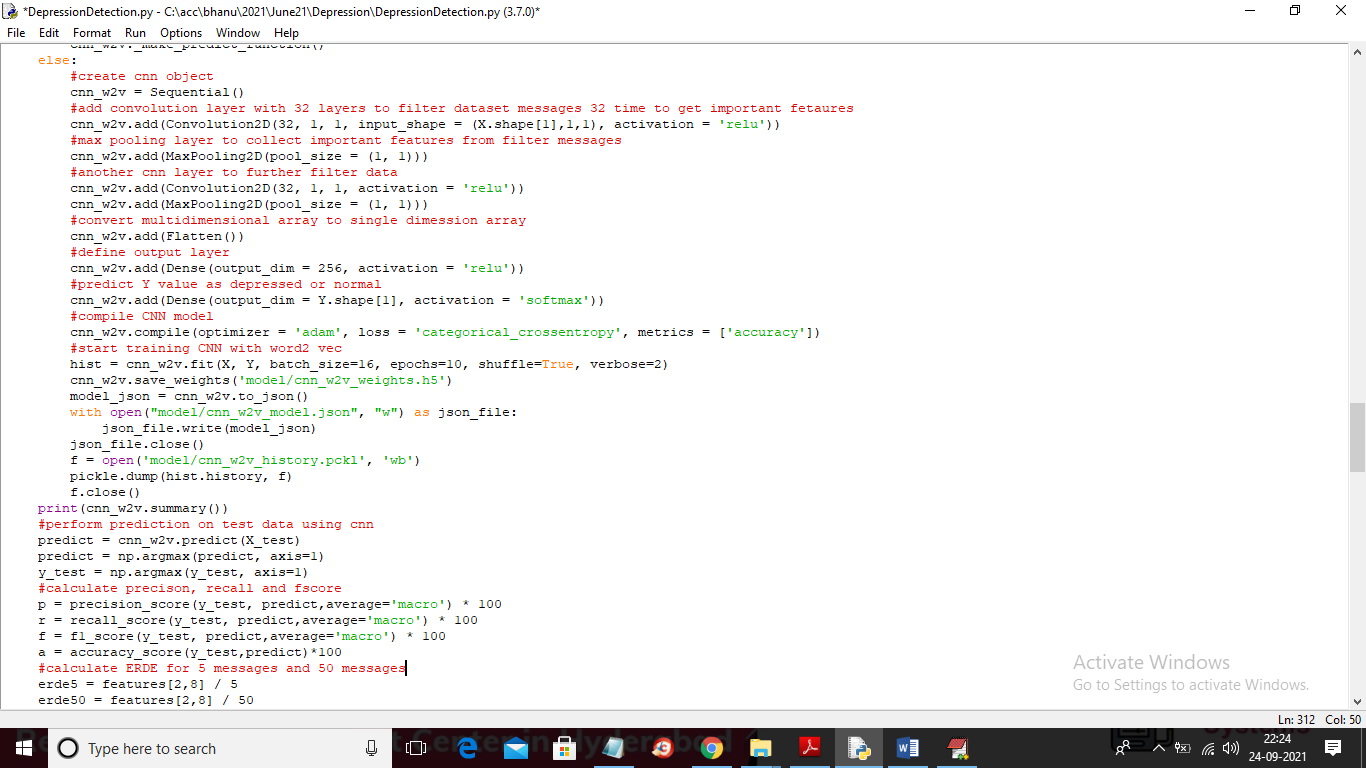
In above two screens you can see Linguistic metadata calculation and in below screen you can see code for part of speech like pronoun and verbs



In above screen you can see using NLTK we find POS from message and in below screen you can see conversion of post text messages to WORD2VEC.



Below screen showing code with WORD2VEC and CNN



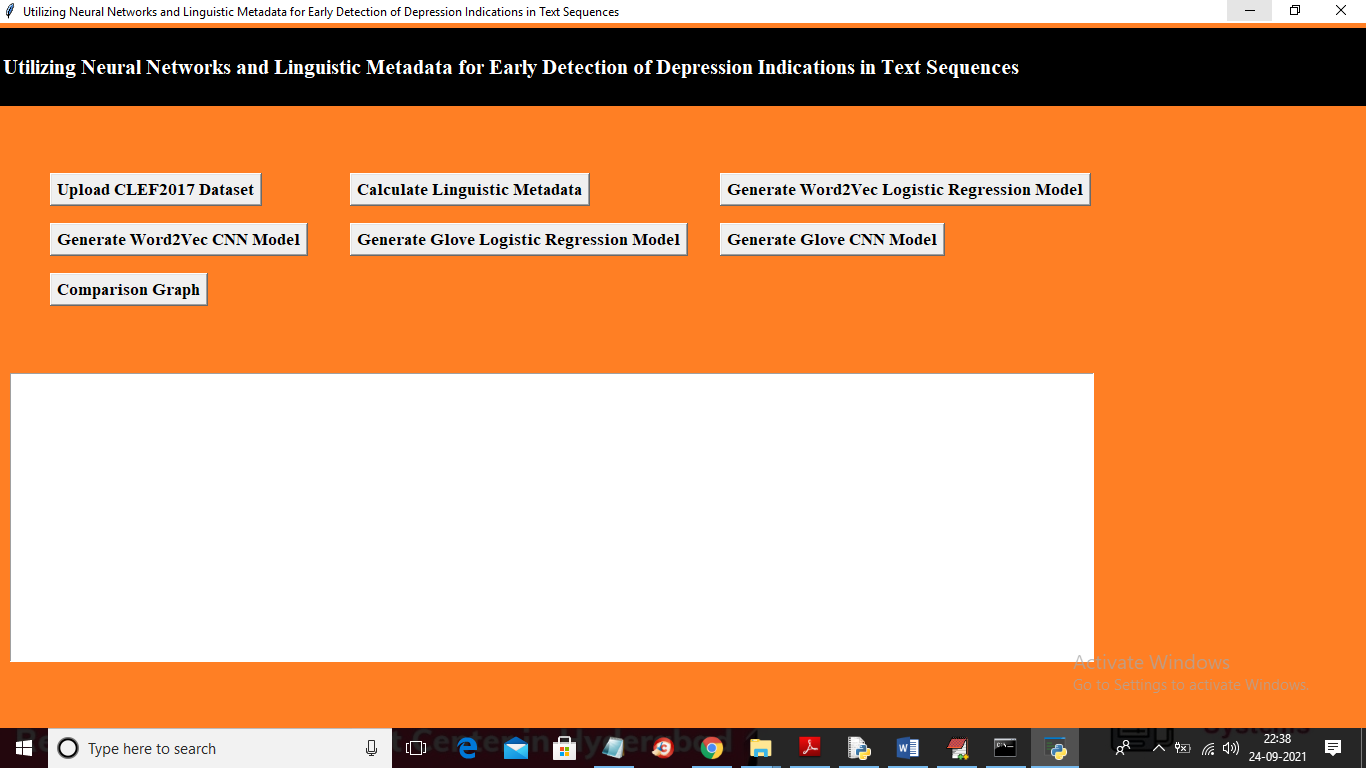
In above screen you can read red colour comments to know WORD2VEC with CNN implementation.

To implement this project we have designed following modules

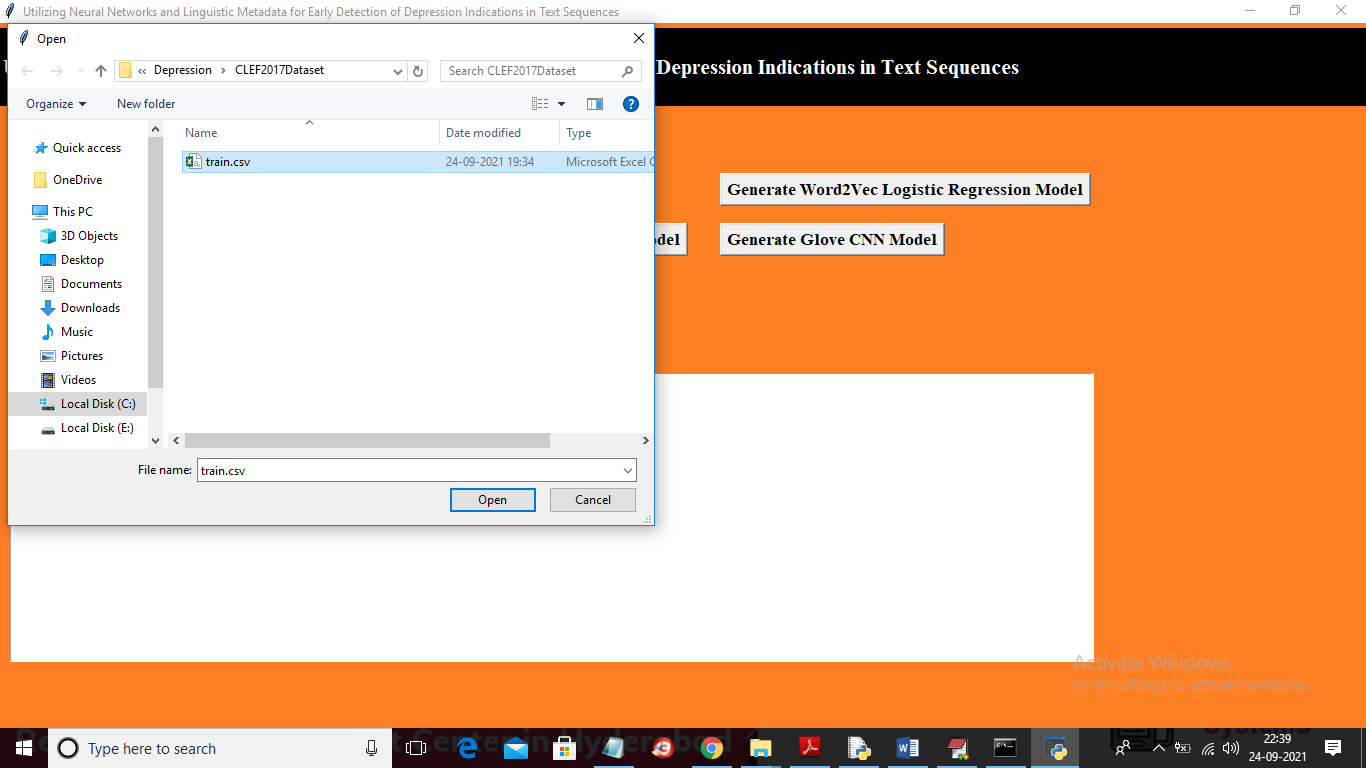
1. Upload CLEF2017 Dataset: using this module we will upload dataset to application
2. Calculate Linguistic Metadata: using this module we will calculate linguistic metadata features
3. Generate Word2Vec Logistic Regression Model: using this module we will convert entire dataset into WORD VECTOR and then train Logistic Regression model
4. Generate Word2Vec CNN Model: using this module we will train CNN with Word Vector and then calculate precision and recall
5. Generate Glove Logistic Regression Model: using this model we will convert entire dataset into Glove vector and then train Logistic Regression model
6. Generate Glove CNN Model: using this model we will train CNN with Glove vector features
7. Comparison Graph: using this module we will plot comparison graph between all algorithms

SCREEN SHOTS

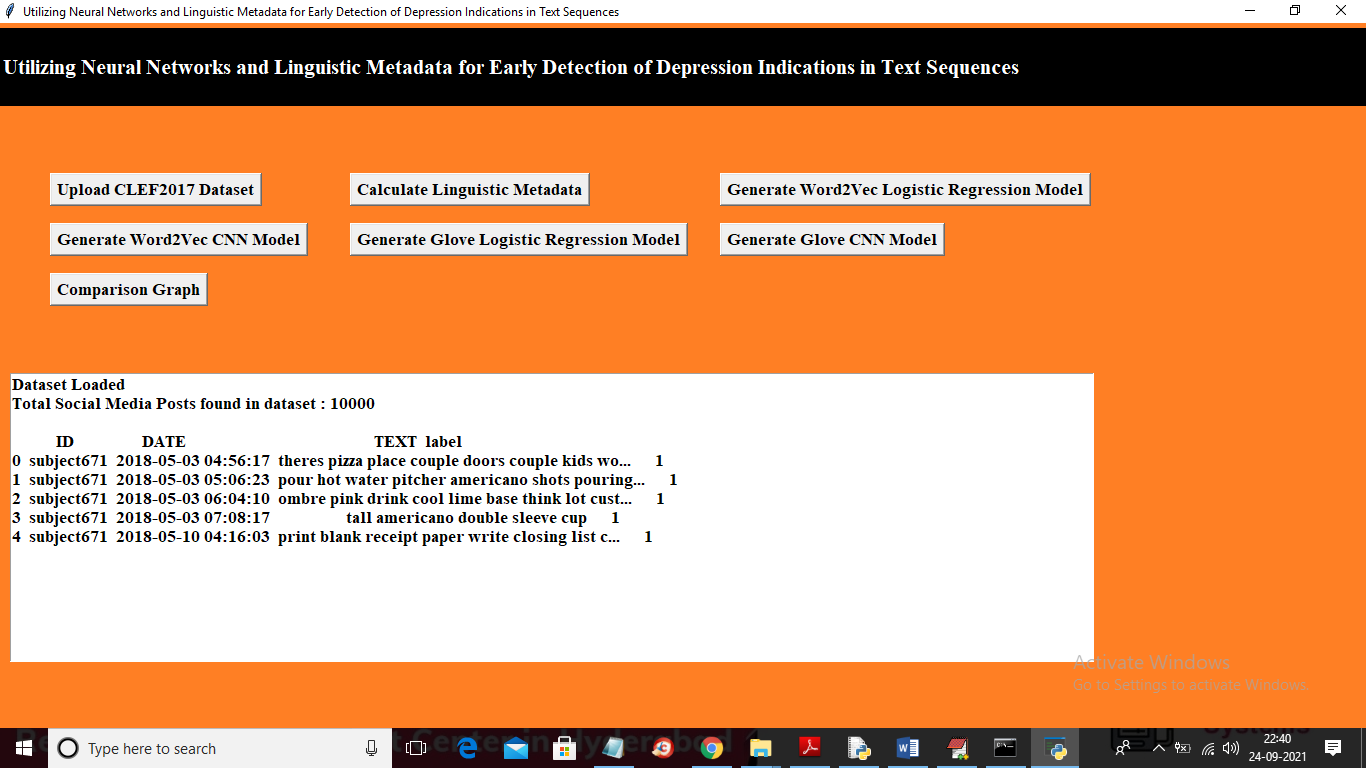
To run project double click on ‘run.bat’ file to get below screen



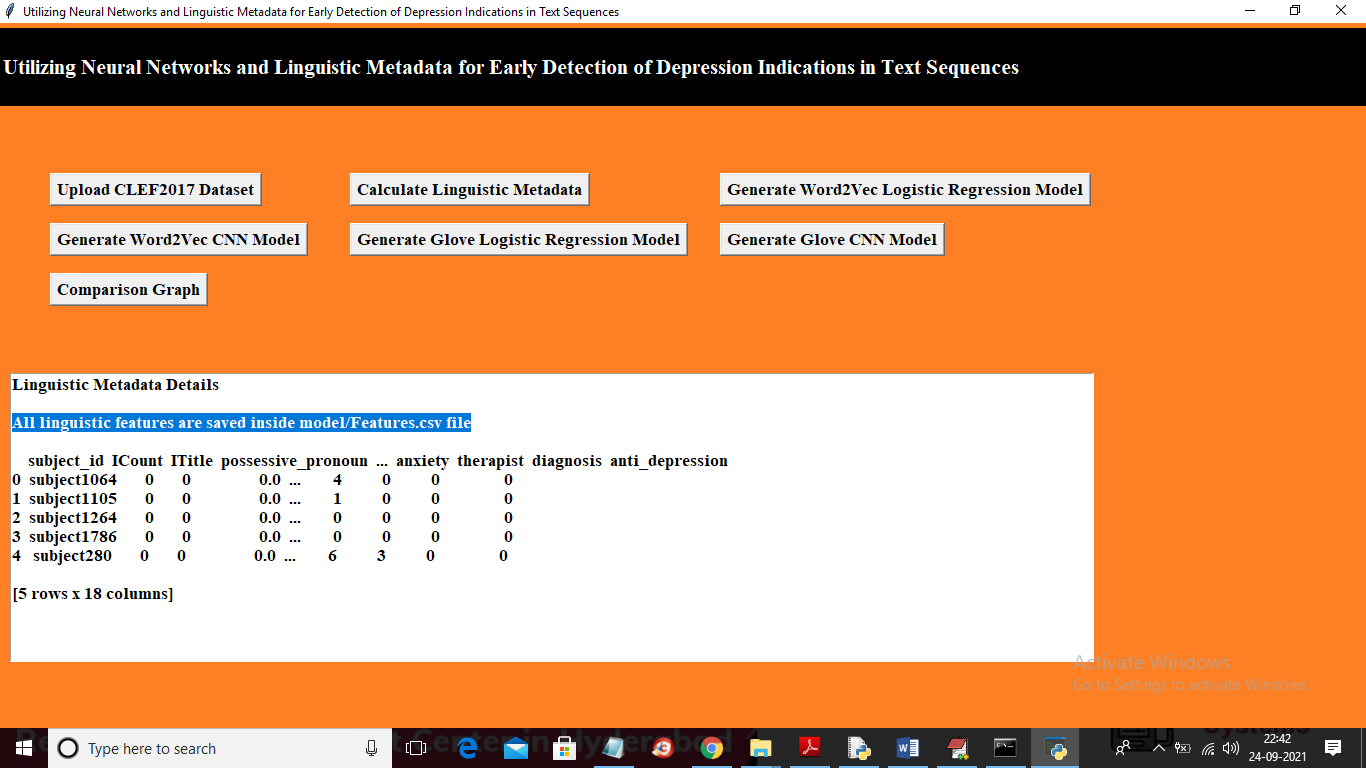
In above screen click on ‘Upload CLEF2017 Dataset’ button to upload dataset and to get below screen



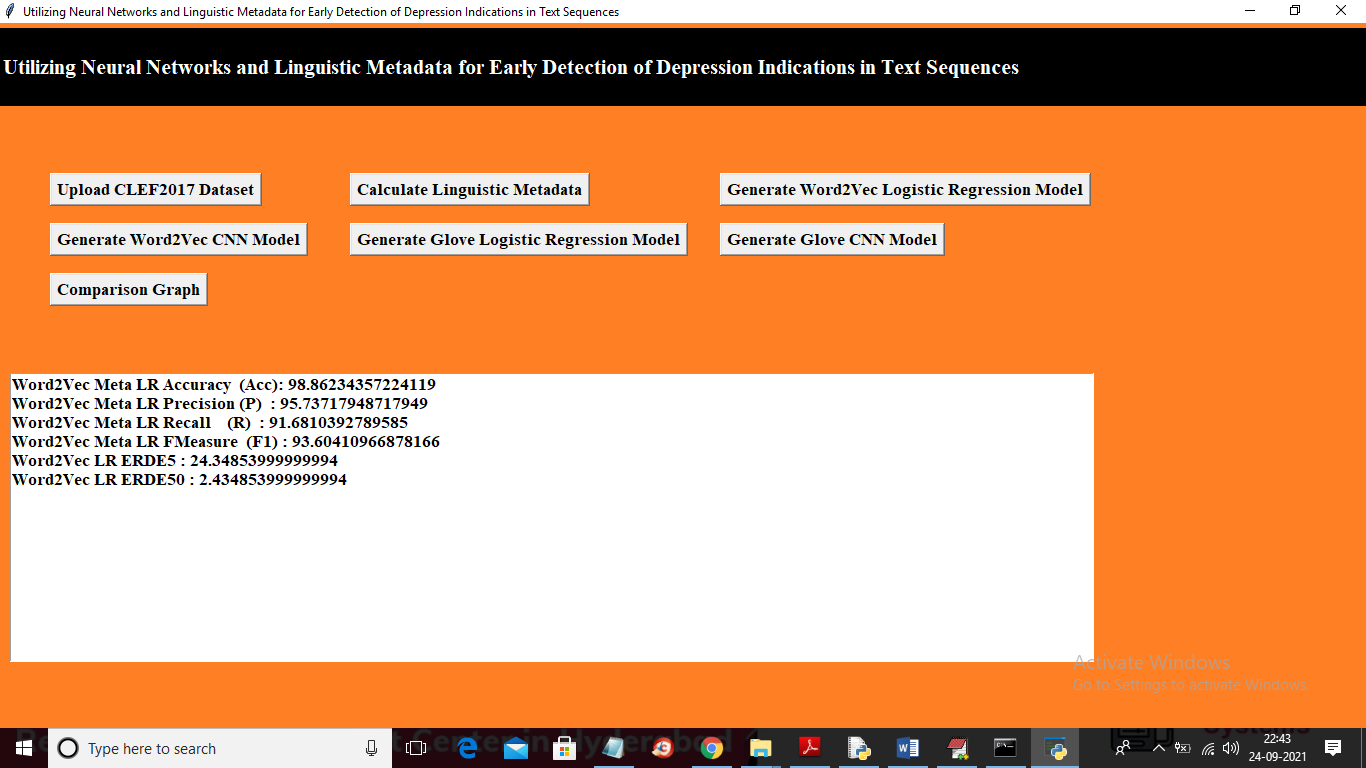
In above screen selecting and uploading ‘train.csv’ file and then click on ‘Open’ button to load dataset and to get below screen



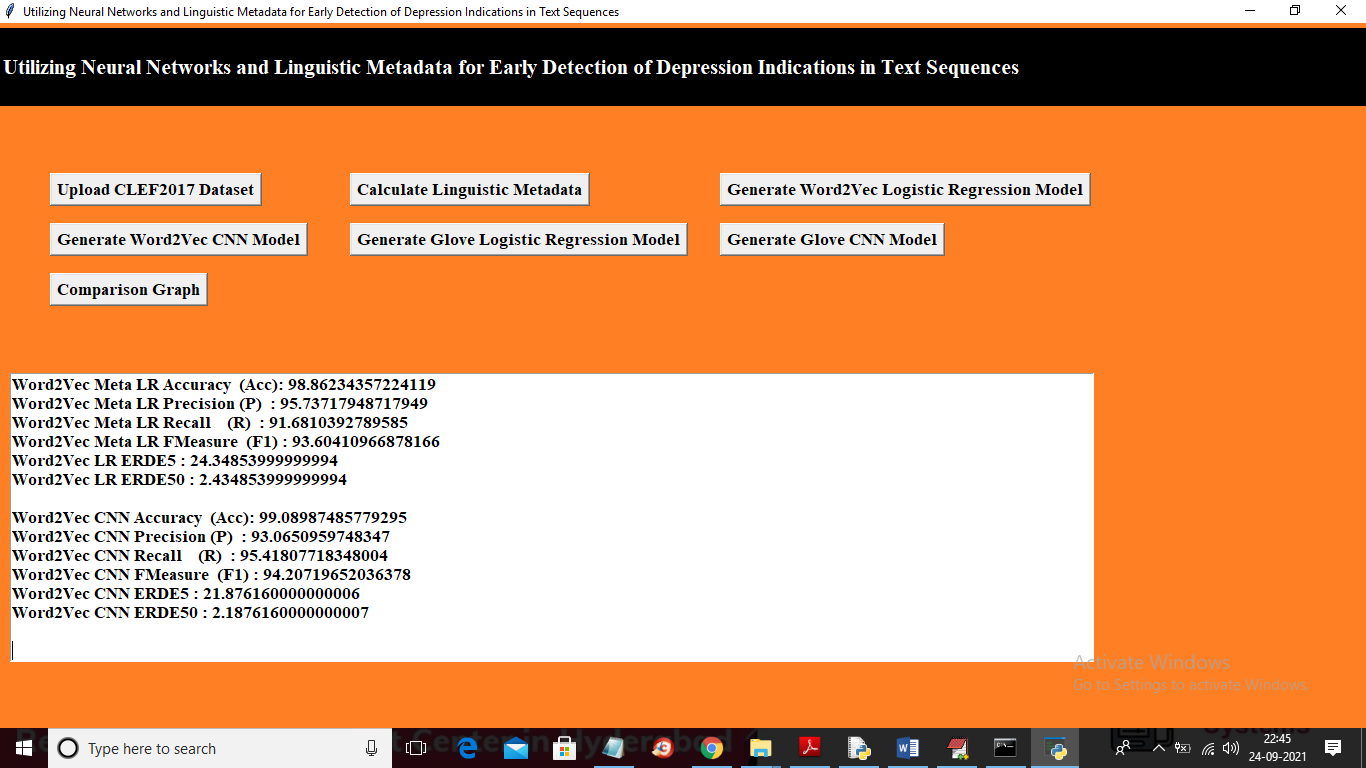
In above screen we can see dataset loaded and now click on ‘Calculate Linguistic Metadata’ button to calculate metadata features



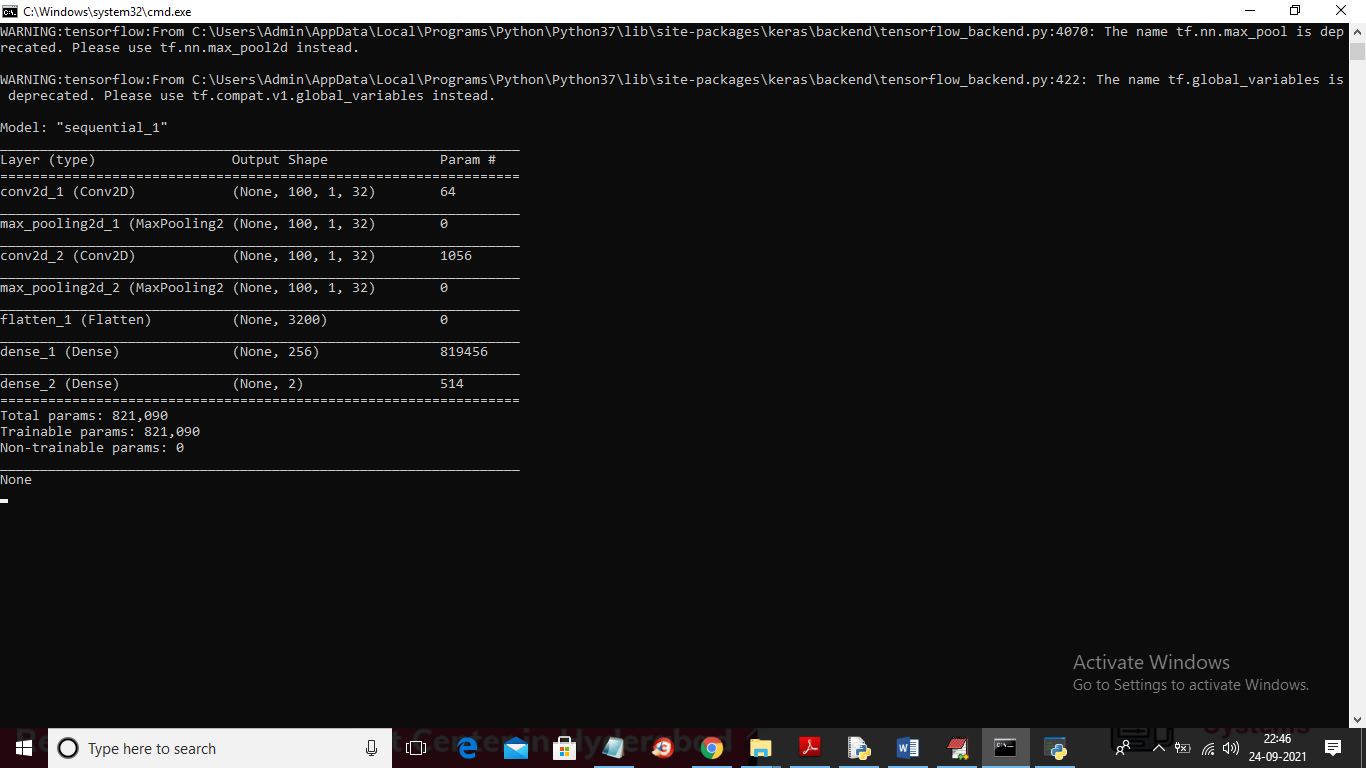
In above screen in blue colour text you can see all features calculated and saved inside ‘model/Features.csv’ file and in above screen I am displaying few features also. Now features are ready and now click on ‘Generate word2Vec Linguistic Regression Model’ button to train logistic regression with Word2Vec and will get below output



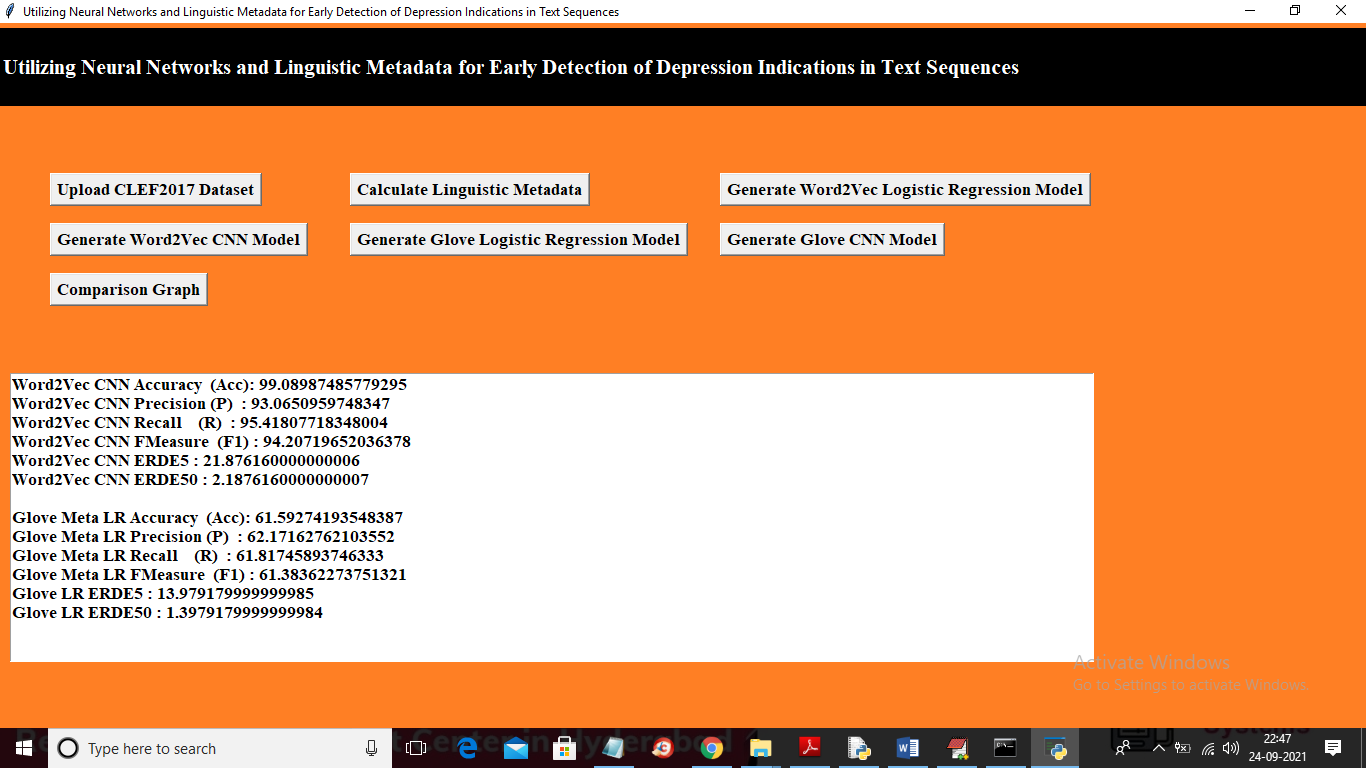
In above screen with Word2Vec logistic regression we got 98% accuracy and 95% precision and ERDE5 as 24% and Erde50 as 2%. The higher the ERDE the more depression is detected. Now click on ‘generate word2Vec CNN Model’ button to train CNN with Word2Vec and to get below output



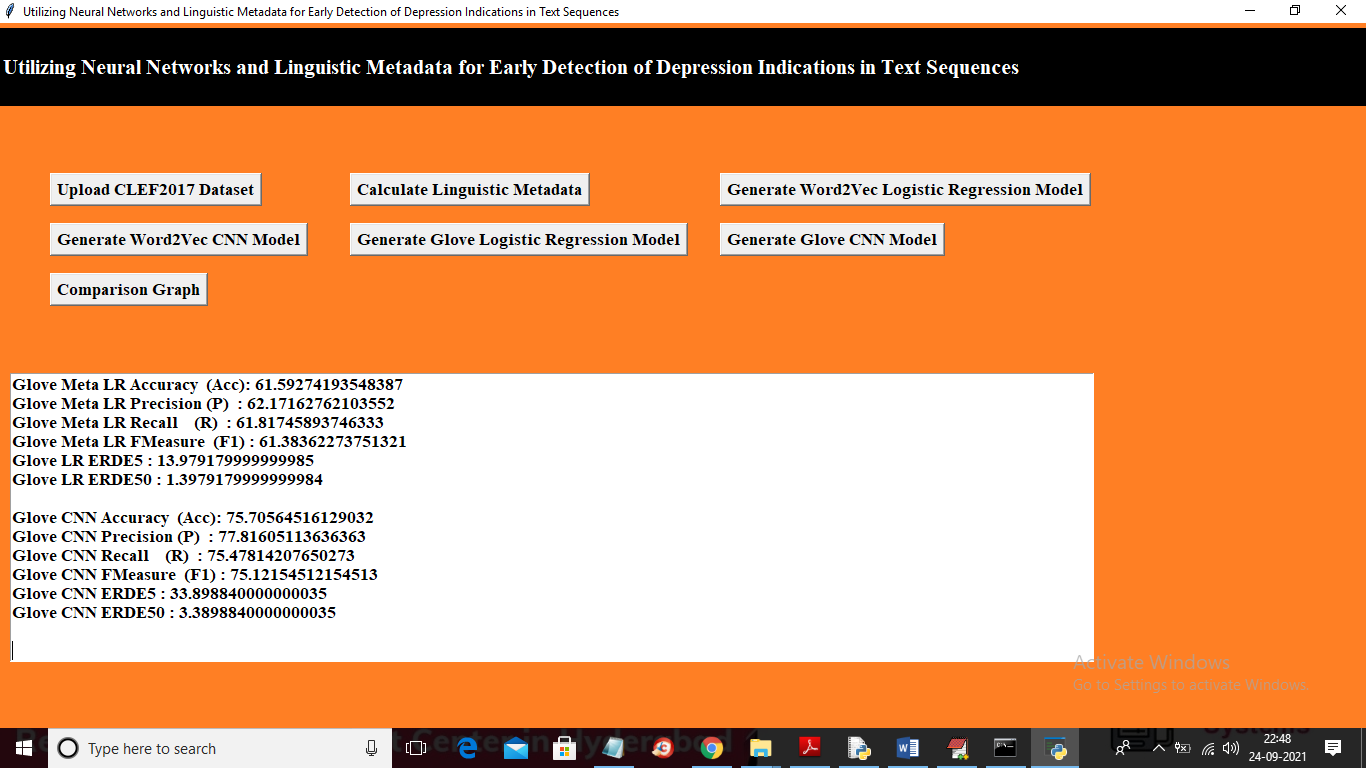
In above screen with word2Vec we got 99% accuracy and in below screen we can see CNN layer details with multiple filters



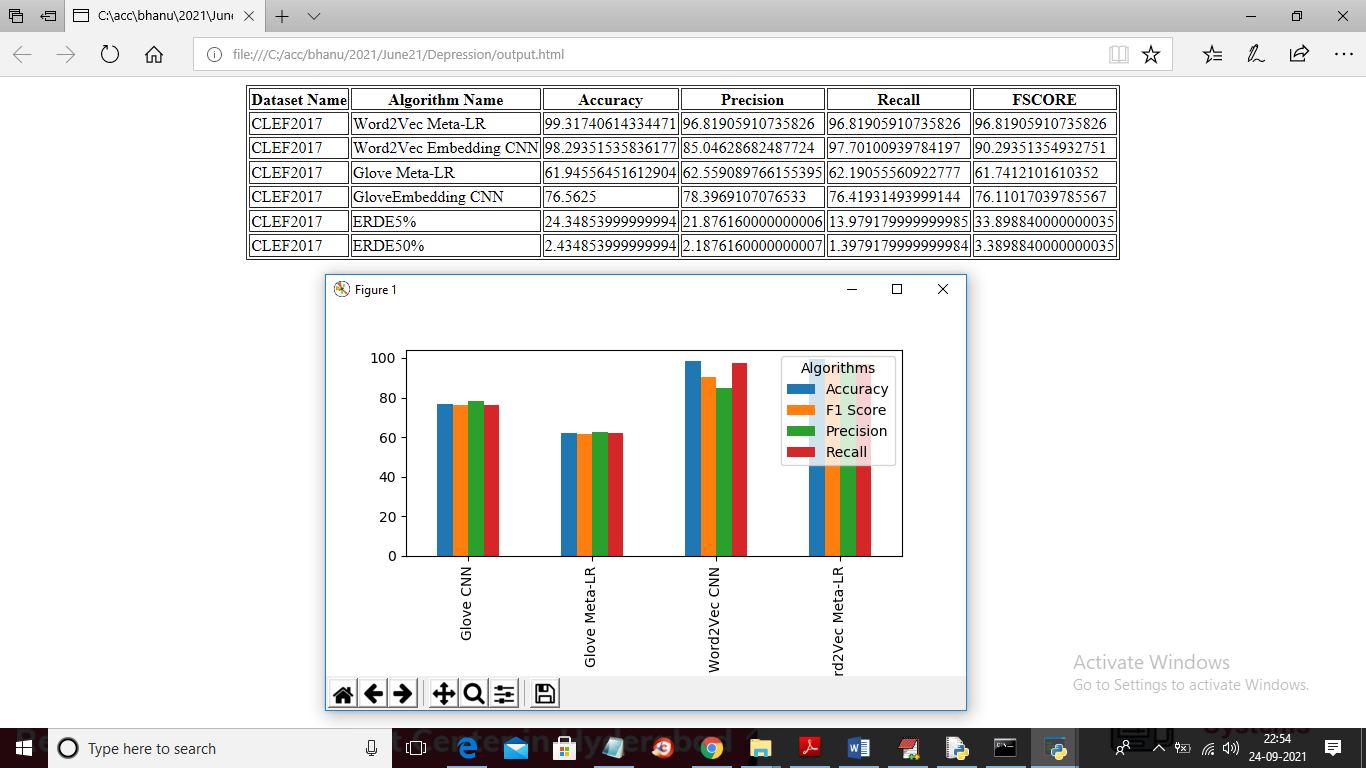
Now go back to main application and then click on ‘Generate Glove Logistic Regression Model’ button to build Logistic Regression with Glove word embedding and to get below output



In above screen with Glove logistic regression we got 61% accuracy and now click on ‘Generate Glove with CNN model’ button to generate CNN model on Glove word embedding and to get below output



In above screen with Glove CNN we got 75% accuracy and now click on ‘Comparison Graph’ button to get below graph



In above graph x-axis represents algorithm names and y-axis represents precision, recall, FSCORE and Accuracy values and same values you can see in tabular format.