Sentiment Analysis using Machine Learning

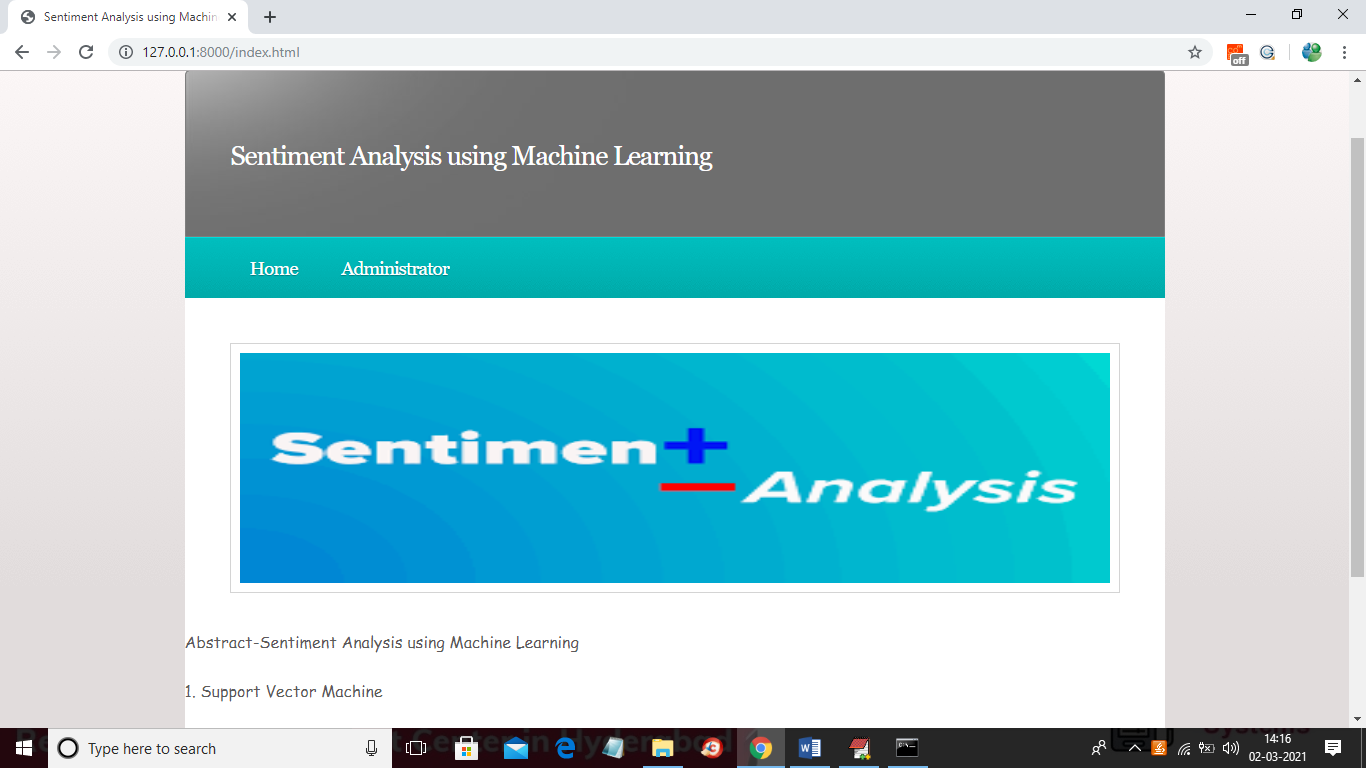
In this project we are comparing various machine learning algorithms performance such as SVM, KMEANS, KNN, Naïve Bayes and CNN in terms of accuracy and execution time. To train and test machine learning performance we are using ‘Trip Advisor Hotel Reviews Dataset’ and by using this dataset we are training ML to predict sentiments from reviews. To implement this project we have designed following modules

1. Administrator: Using this module admin can login to application by using username as ‘admin’ and password as ‘admin’ and then can run all algorithms.
2. Upload: using this module admin can upload reviews dataset and then application will read all reviews and then remove all stop words, special symbols and numeric values and make clean text. Clean text will be converted to TF-IDF (term frequency Inverse Document Frequency) vector and this vector contains count of each word from entire dataset. Vector will be splitted to train and test data where 80% dataset will used for training and 20% dataset will used for testing
3. Run KMEANS & SVM: Using this module application will used 80% data to train both algorithms and then apply 20% data on trained model to calculate accuracy of both algorithms and then calculate execution time also.
4. Run KNN & KMEANS: using this module run both algorithms to calculate its accuracy and execution time.
5. Run Naïve Bayes & CNN: using this module run both algorithms to calculate its accuracy and execution time.
6. Test Sentiment: using this module you can enter any sentence and then application predict sentence sentiment as neutral, positive or negative.

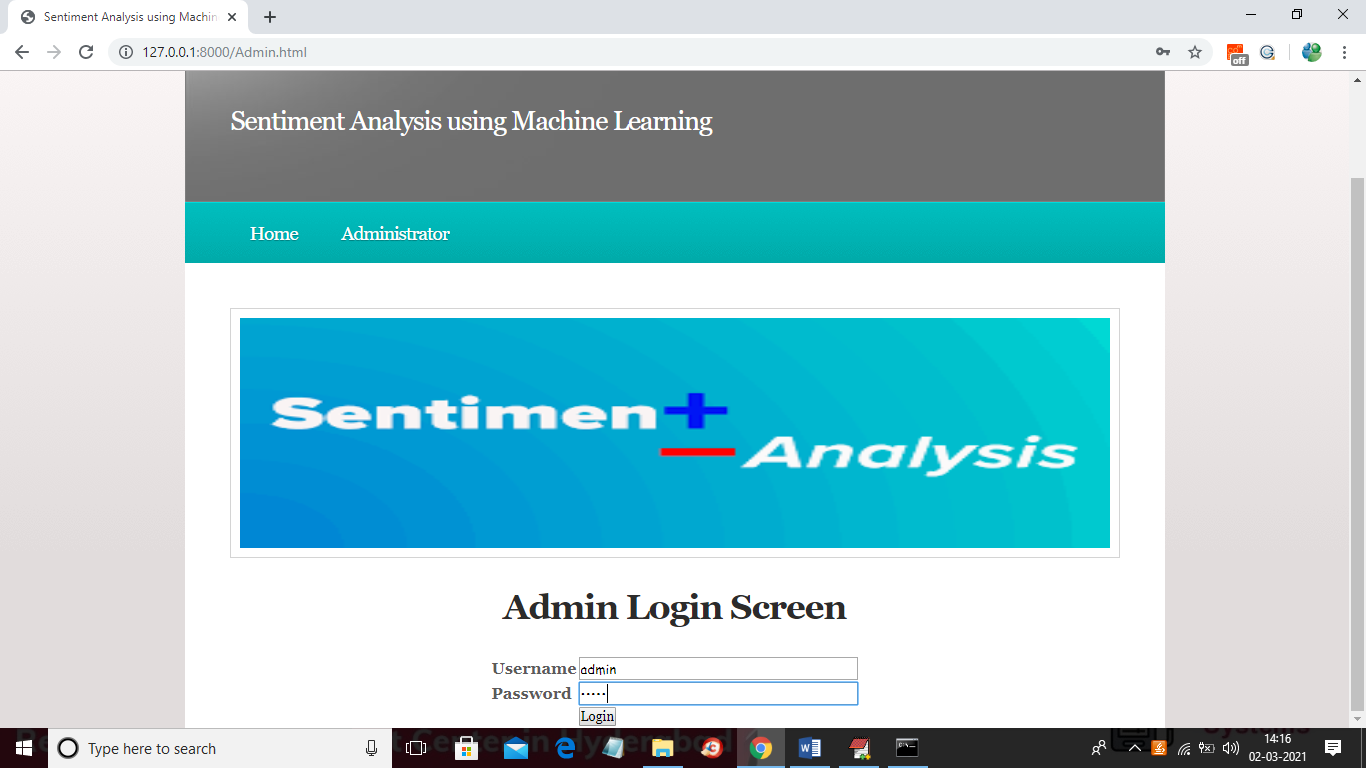
Note: You are asking to implement reinforcement learning but this algorithm used for gaming project and unsupervised algorithm is nothing but KMEANS only.

SCREEN SHOTS

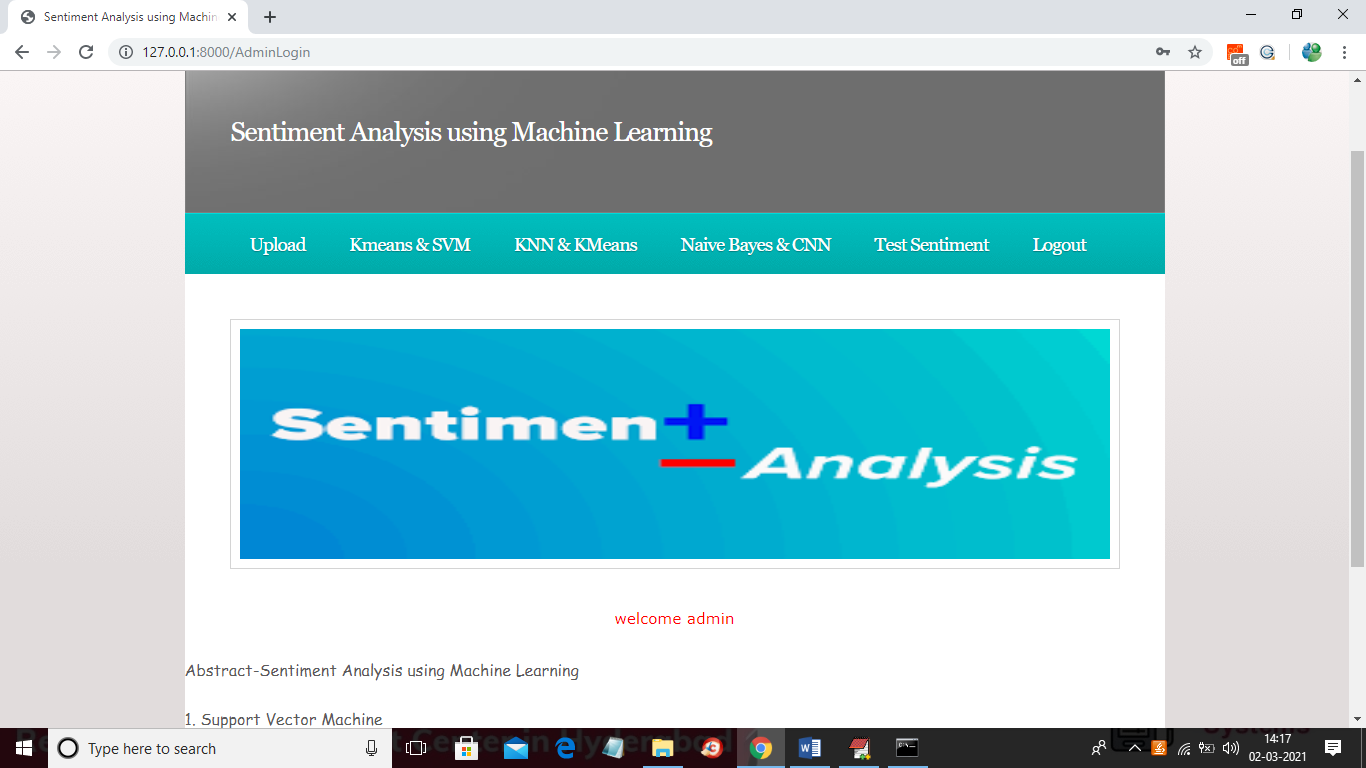
To run project install django and then run server to get below screen



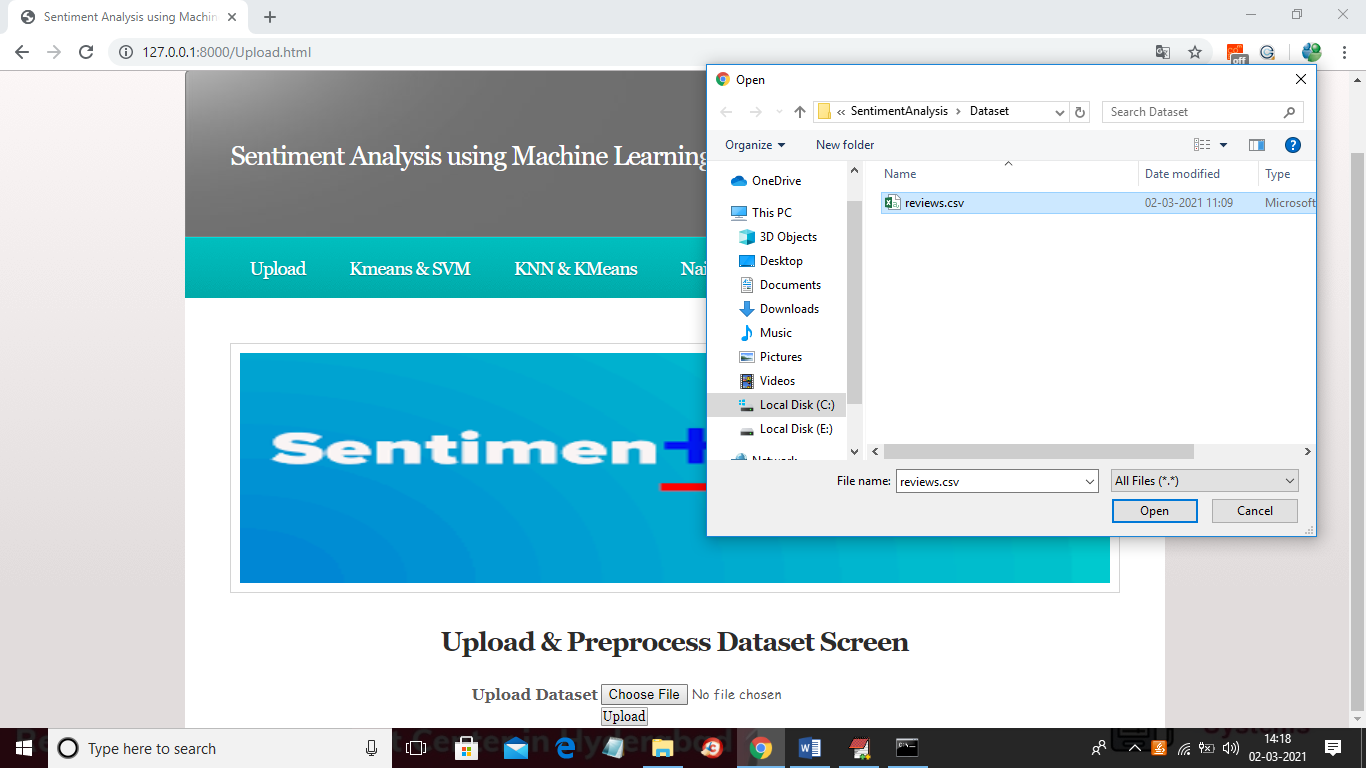
In above screen click on ‘Administrator’ link to get below screen



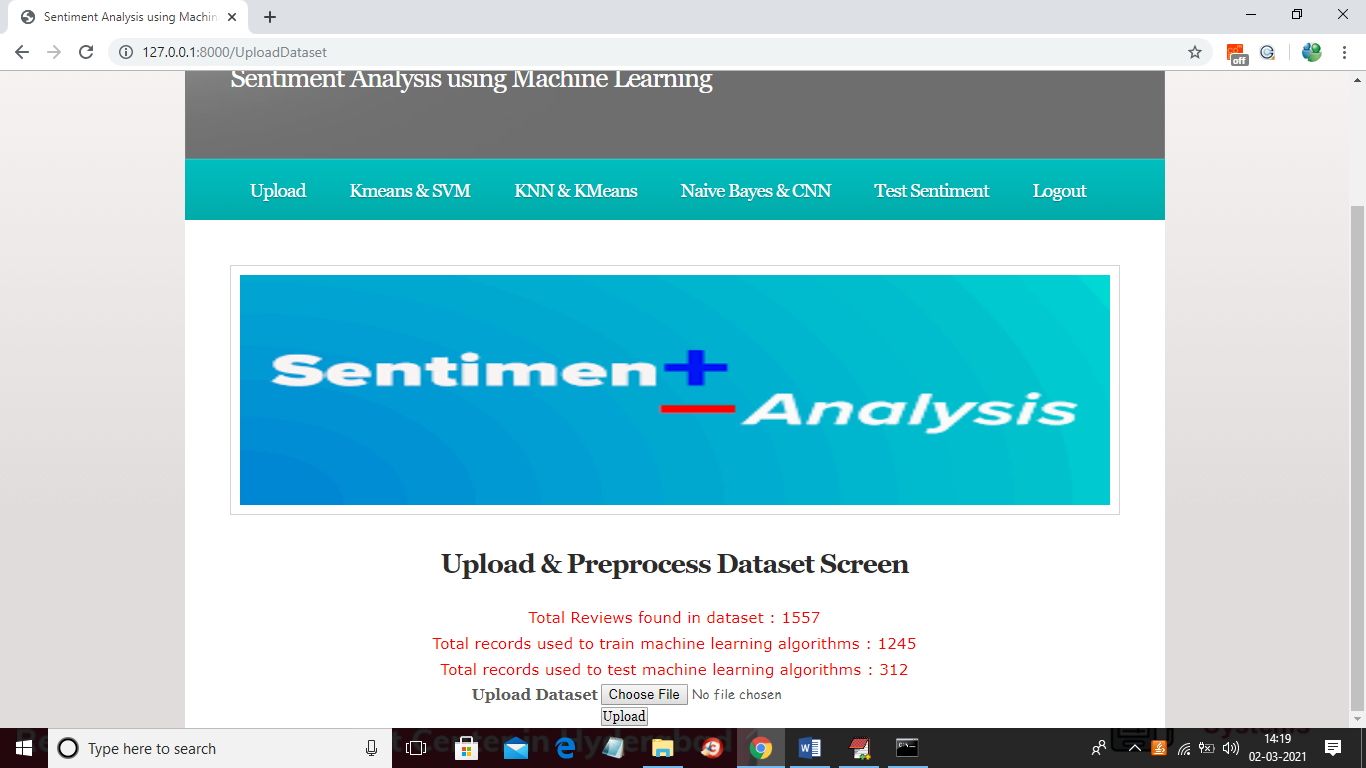
In above screen admin is login and then click ‘Login’ button to get below screen



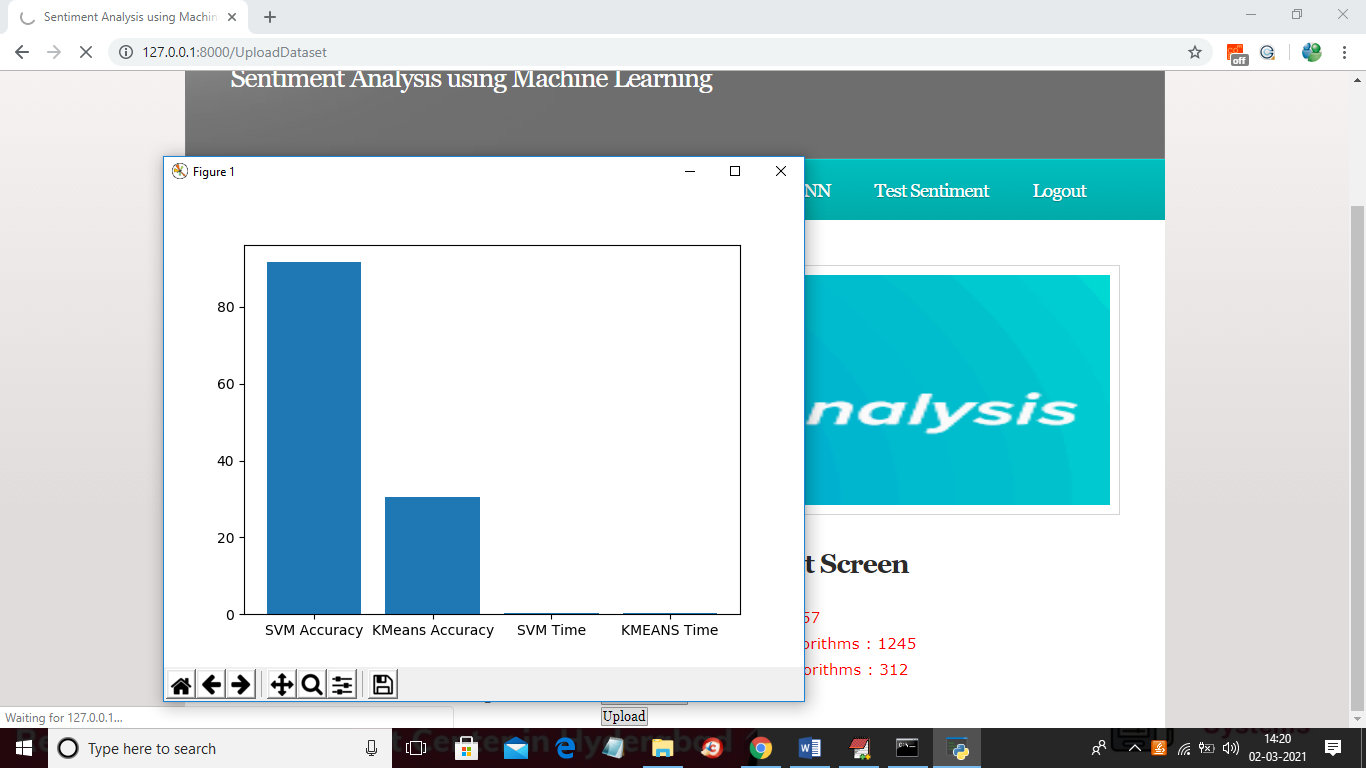
In above screen click on ‘Upload’ link to upload dataset



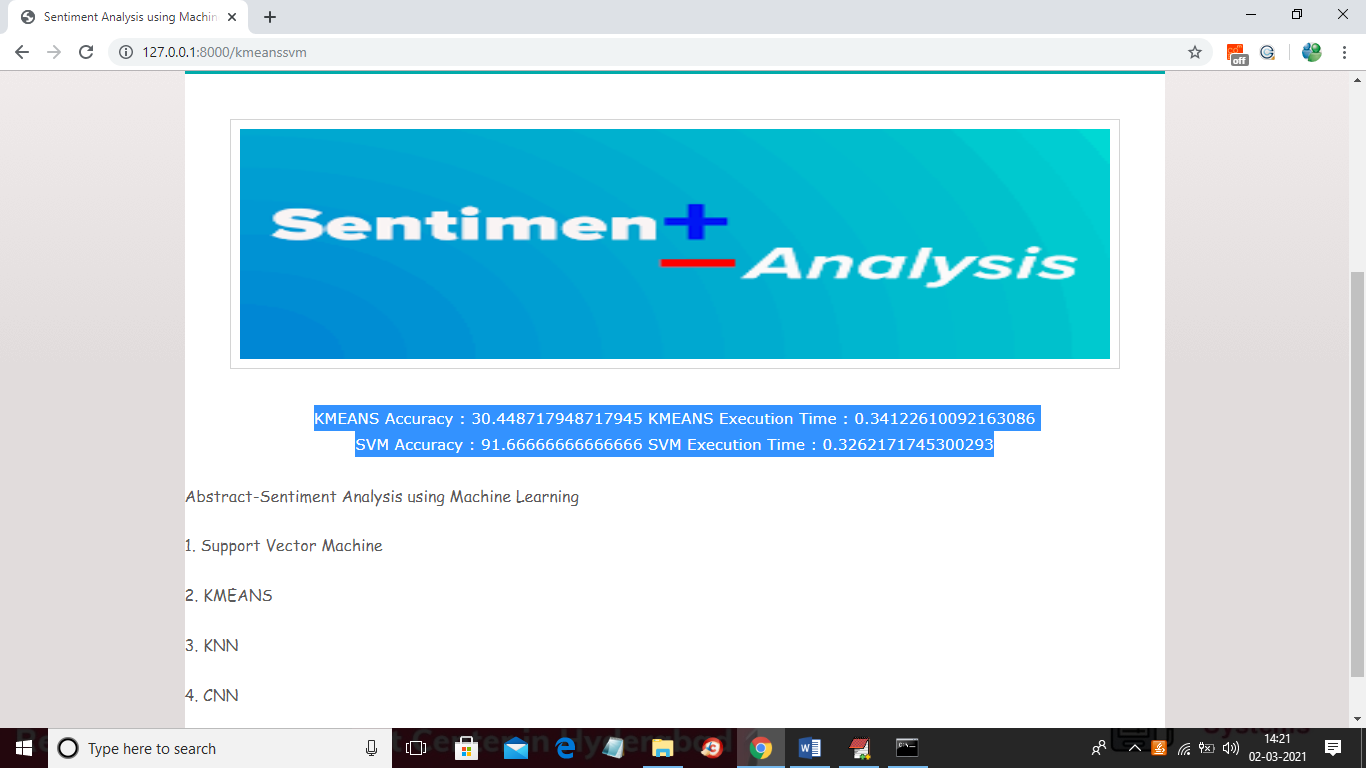
In above screen selecting and uploading ‘reviews.csv’ dataset and then click on ‘Open’ button and then click on ‘Upload’ button to load dataset and to get below screen



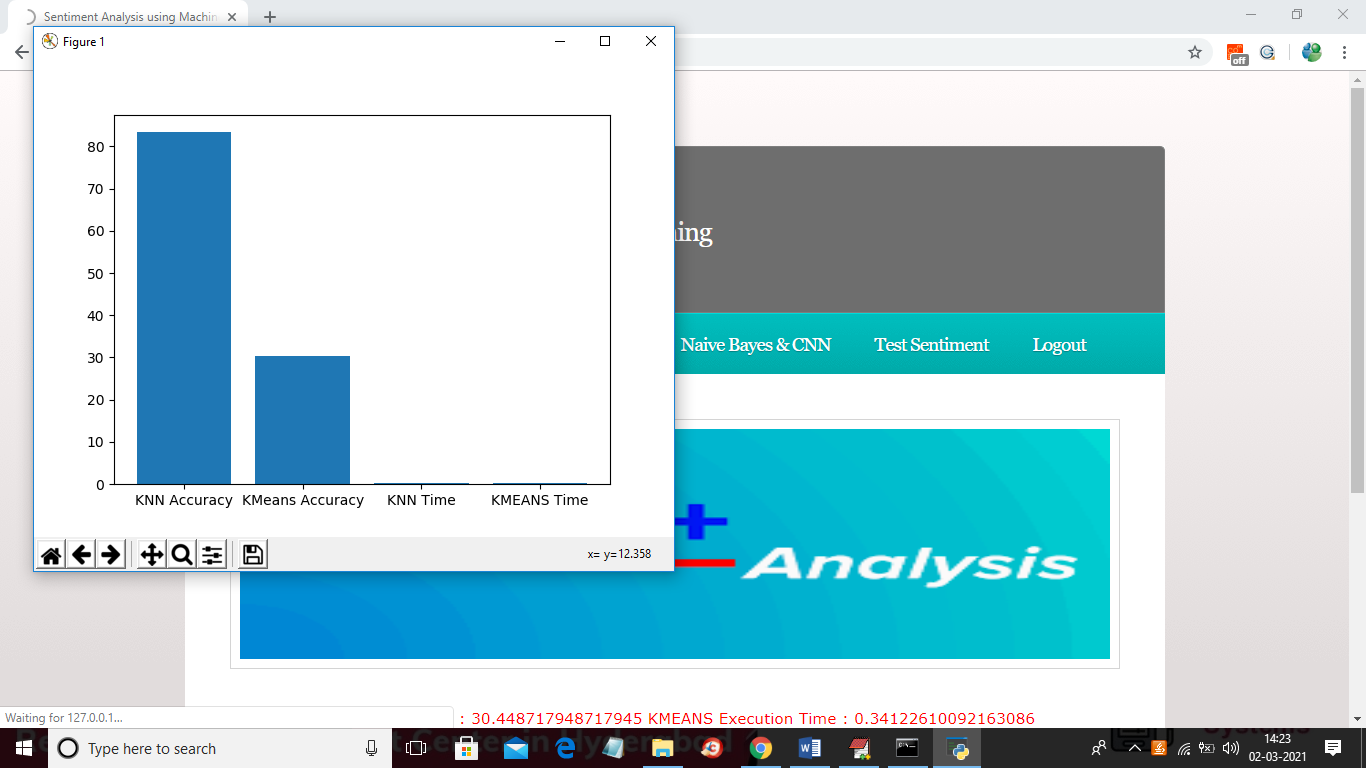
In above screen dataset loaded and dataset contains total 1557 reviews and application using 1245 reviews for training and 312 reviews for testing and now dataset is ready and now click on ‘Kmeans & SVM’ link to run both algorithms and to get below graph



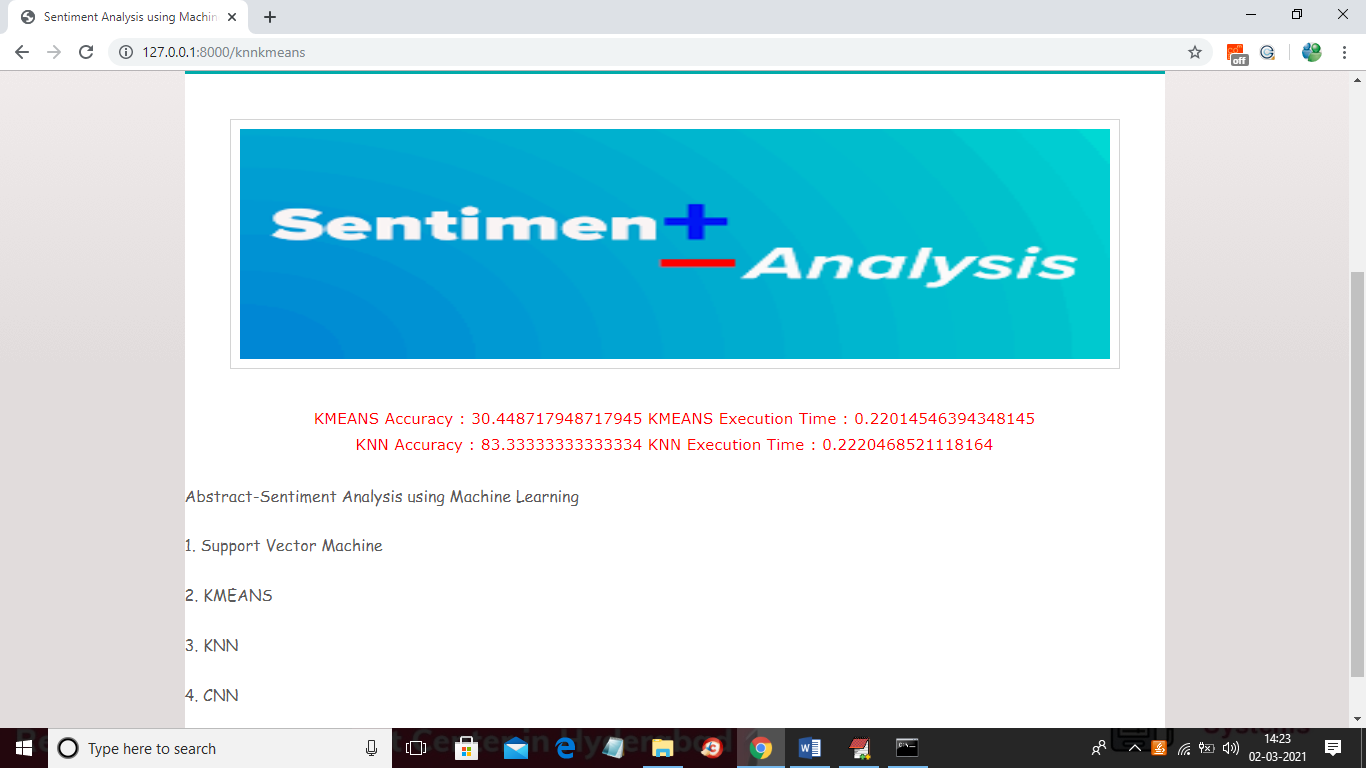
In above graph x-axis represents algorithm name and y-axis represents accuracy and time of both algorithms and from above graph we can see SVM got high accuracy and now close above graph to get below screen



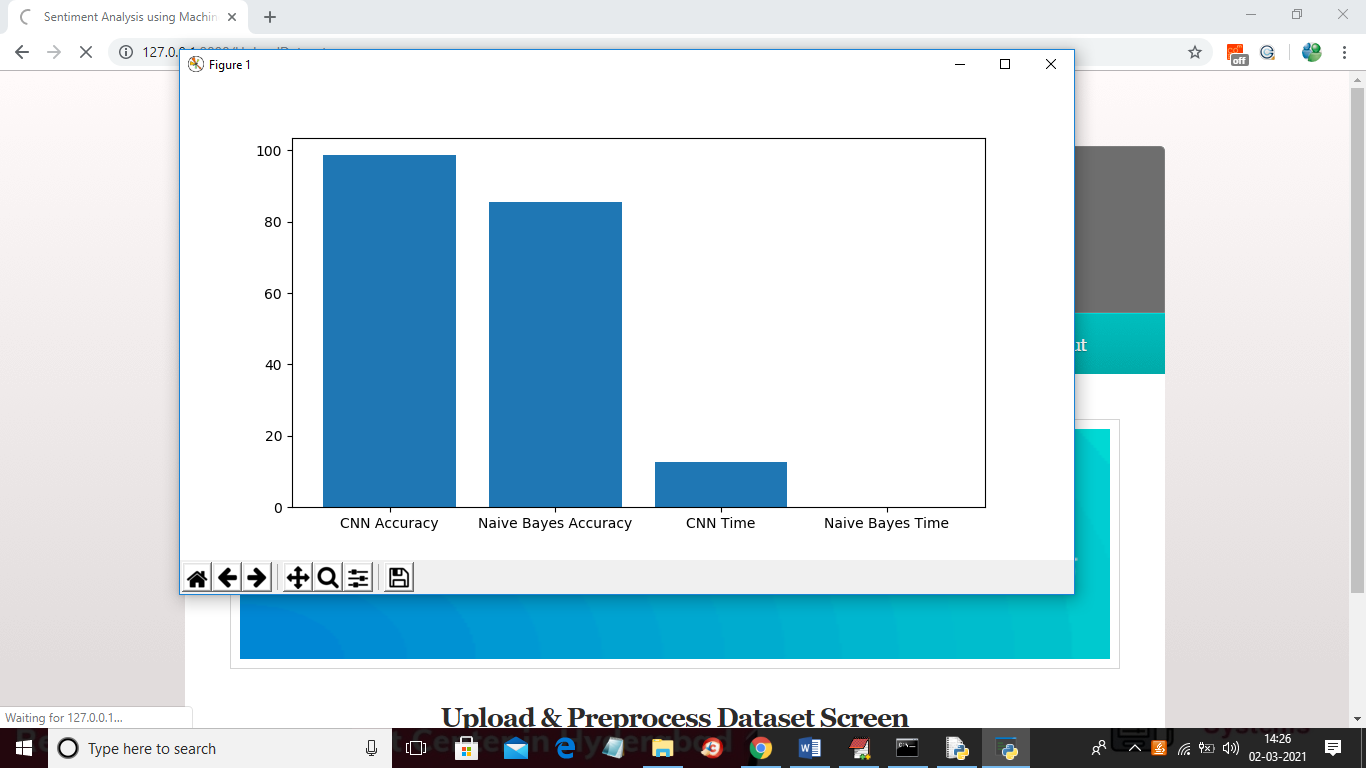
In above screen we can see KMEANS accuracy is 30% and execution time is 0.34 milli seconds and SVM accuracy is 91% with time 0.32 and now click on ‘KNN & KMEANS’ link to get its accuracy



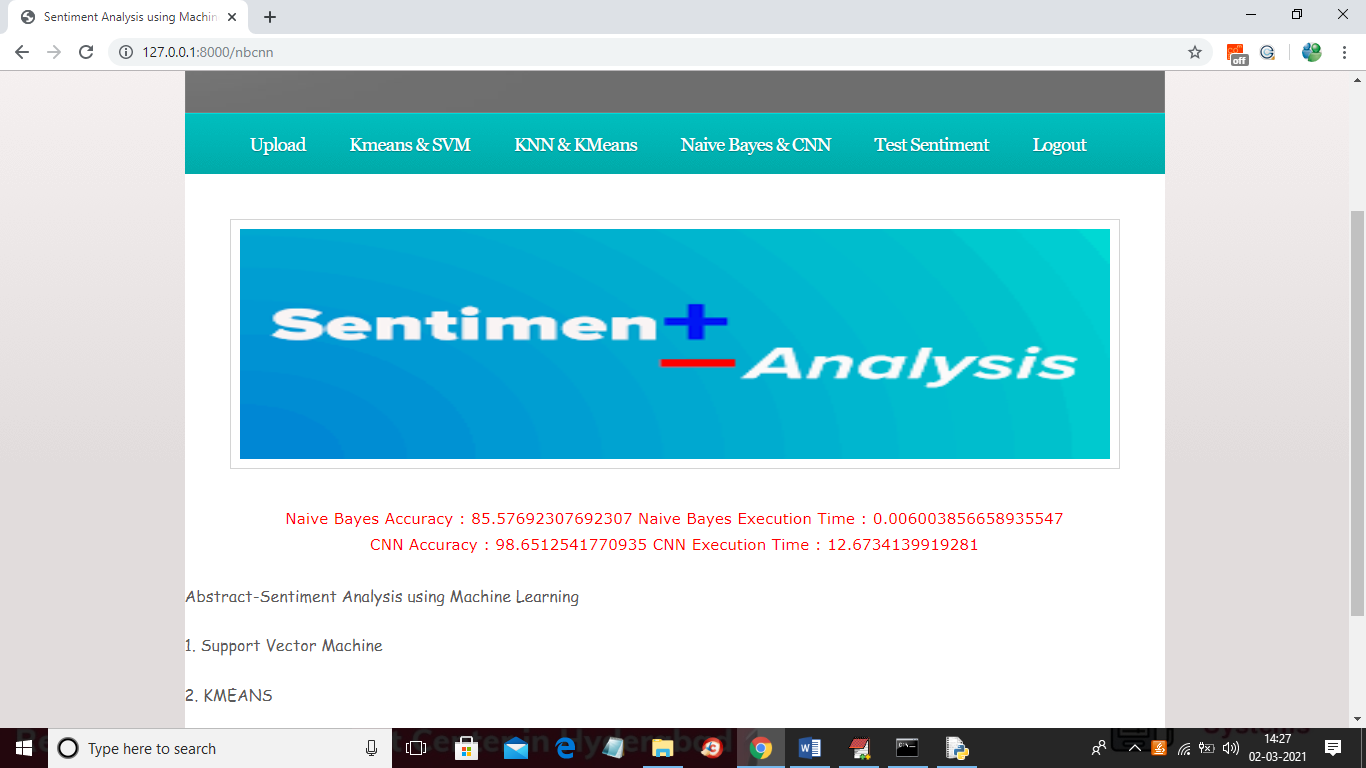
In above graph KMEANS accuracy is same and KNN got 80% accuracy and now close above graph to get below screen



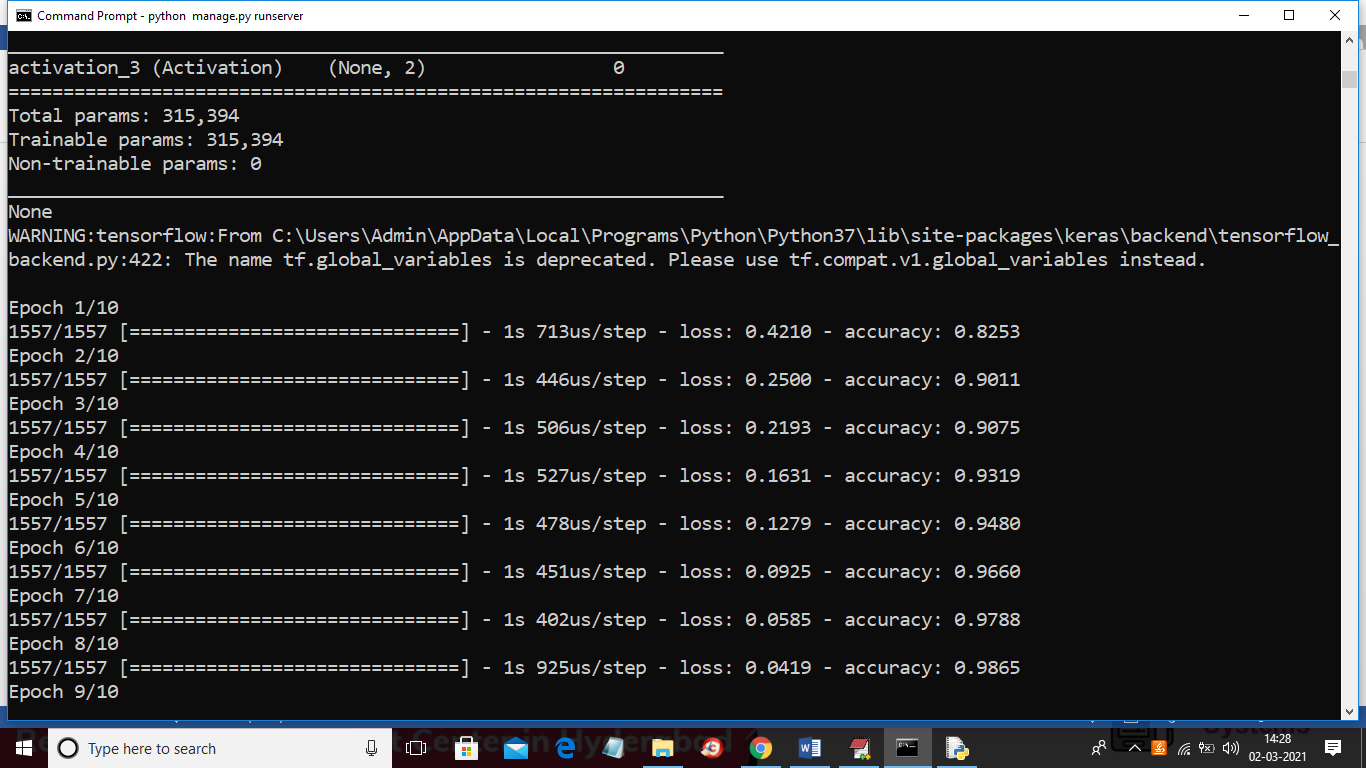
In above screen you can see accuracy and time of both algorithms and now click on ‘Naïve Bayes & CNN’ link to get there accuracy



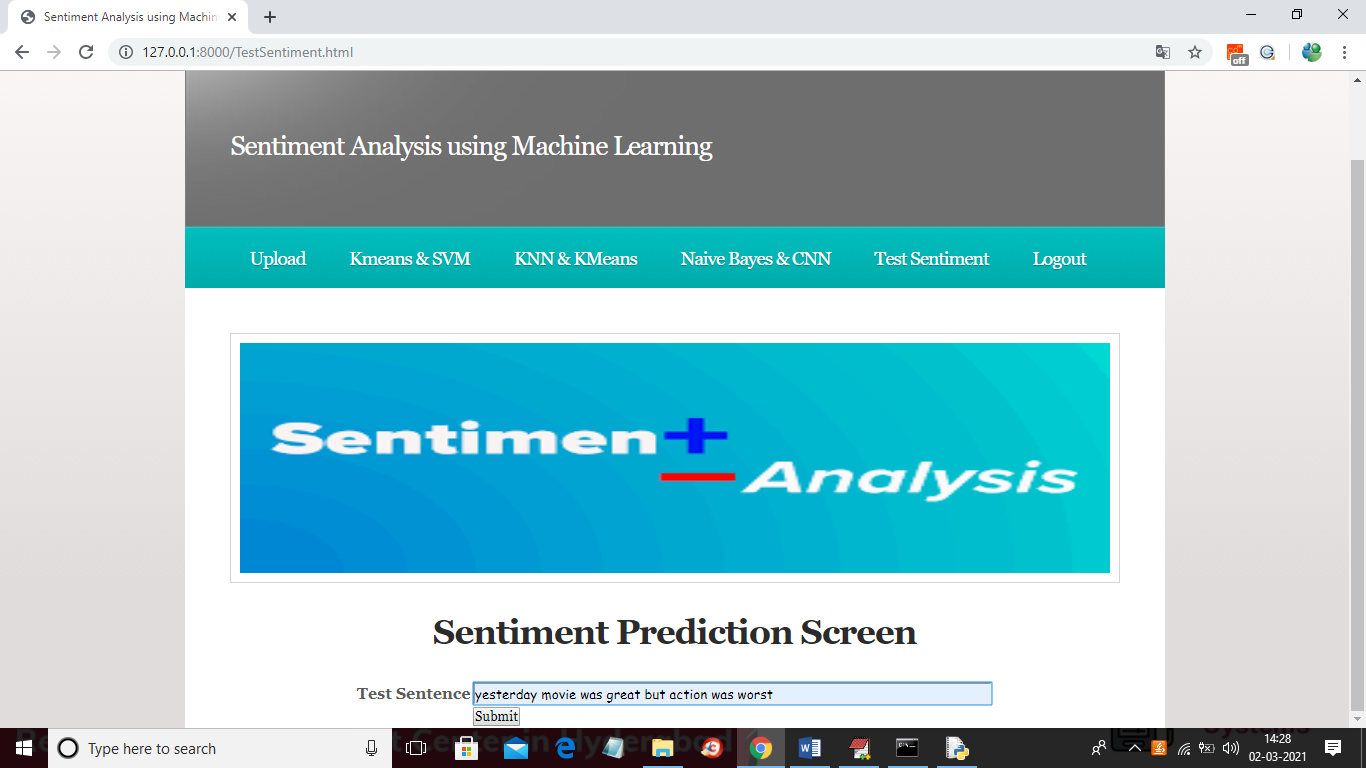
In above graph CNN accuracy is closed to 100% and it took little more execution time compare to other algorithms but its accuracy is high and now close above graph to get below screen



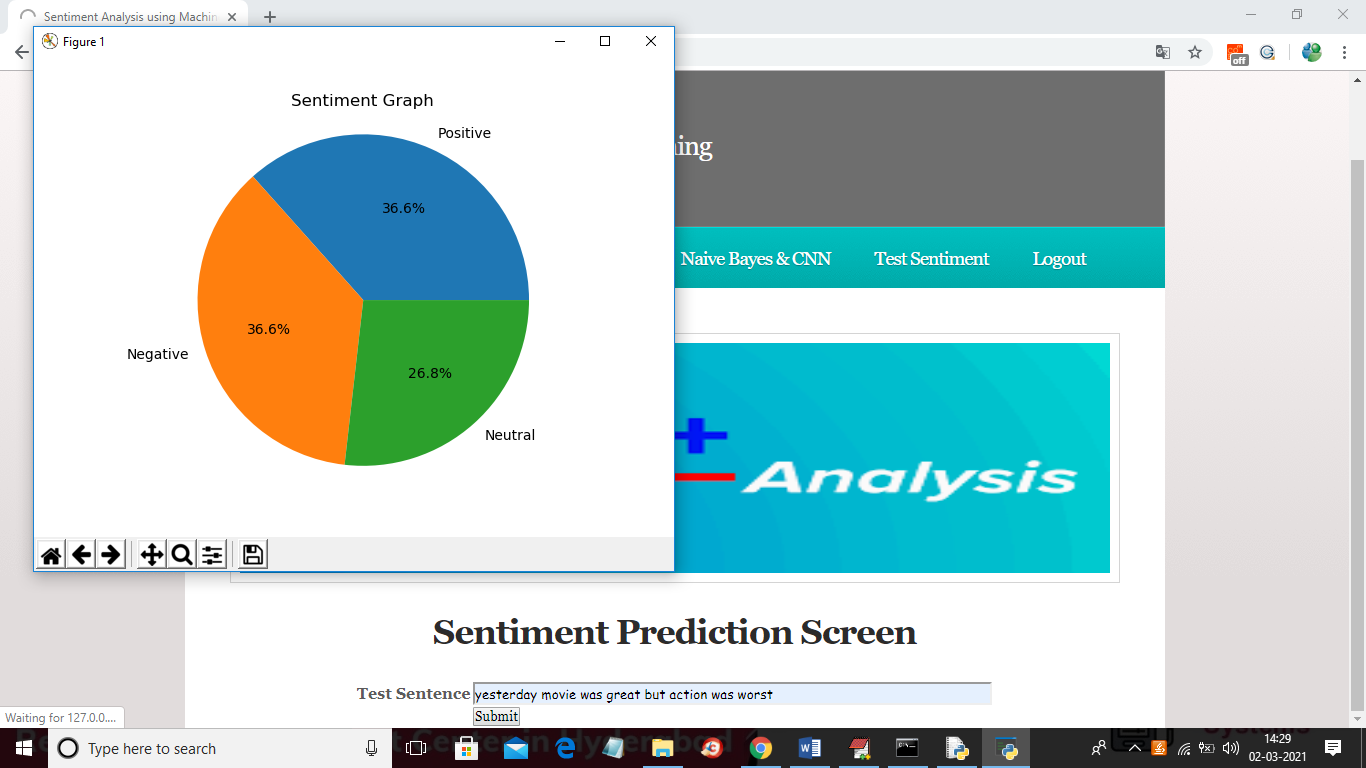
In above screen we can see Naïve Bayes and CNN accuracy with time and in below console we can see CNN layer details



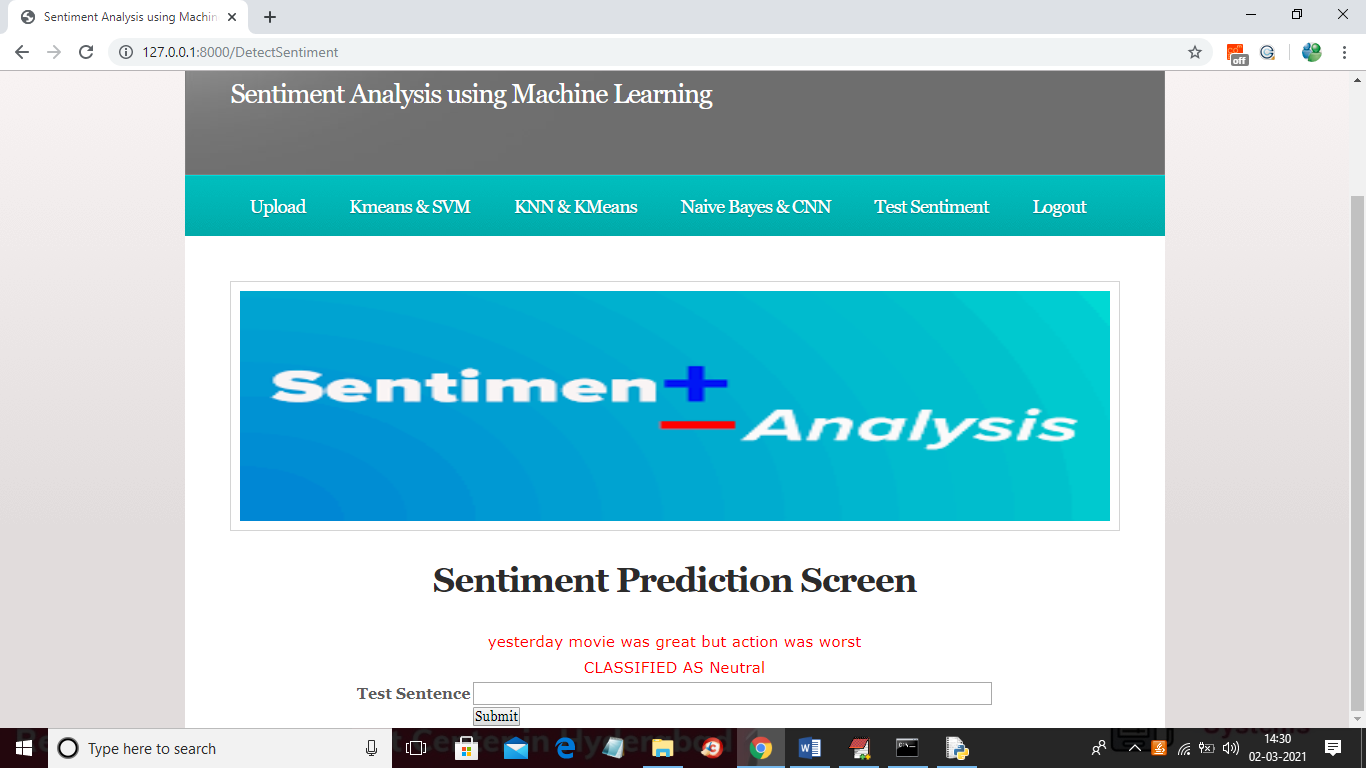
In above screen to train CNN we took 10 iterations and in each iteration CNN accuracy got better and better and now click on ‘Test Sentiment’ link to get below screen



In above screen you can enter some sentence and then click on ‘Submit’ button to get below result



In above graph application calculate percentage of sentiments from that sentence and then give below final detection result



In above screen that sentence classified as NEUTRAL. Similarly you can enter any sentence and get result