Requirement: -- Observe the behavior of social networking user response and analyze the prediction of on which points refers positive and negative behavior on personalized observation basis and find the predication and accuracy using datasets.

Tasks:

1. Load the csv data train.csv and show the results.
2. Train the data.
3. Load the csv data test.csv and show the results.
4. Train the results.
5. Combine the both results and show head and tails results.
6. Find the pattern i.e. @user and puts it in a list for further task.
7. Remove @user from the sentences in the dataset
8. Remove Twitter Handles (@user) and show the results.
9. Remove Short words by combine the tidy tweets.
10. Tokenize all the cleaned tweets in the dataset.
11. Stem the suffixes (Stemming is a rule-based process of stripping the suffixes (“ing”, “ly”, “es”, “s” etc) from a word. For example, For example – “play”, “player”, “played”, “plays” and “playing” are the different variations of the word – “play”)
12. Stitch the tokens back together.
13. Store all the words from the dataset which are non-racist/sexiest.
14. Show all word positive.
15. Show all word negative.
16. Find the nested list of all the hash tags from the positive.
17. Find the nested list of all the hash tags from the positive.
18. Create a nested list if all the hash tags from the negative and review from the negative.
19. Create a nested list if all the hash tags from the negative and review from the negative.
20. Plot the graph of positive datasets in the tweets counting the frequency of words having positive sentiment.
21. Plot the bar plot for the 10 most frequent words used for positive hash tags.
22. Count the frequency of the words having negative sentiment.
23. Create the data frame for the most frequently used words for negative hash tags.
24. Plot the bar plot for the 10 most frequent word used for negative hash tags.
25. Plot the pie plot for the 10 most frequent word used for negative hash tags.
26. Clean the tweets by bag of word features (bag of Words is a method to extract features from text documents. These features can be used for training machine learning algorithms. It creates a vocabulary of all the unique words occurring in all the documents in the training set.)
27. Apply TF-IDF Features (Tf-idf stands for term frequency-inverse document frequency, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.)
28. Apply machine learning models by using the features from Bag-of-Words for training set.
29. Apply machine learning models by using the features from TF-IDF for training sets.
30. Split the data into training and validation set into Bag-Of-words and TF-IDF features.
31. Apply Random Forest Algorithm for Bag-of-Words feature scaling.
32. Train the bag-of word by Random Forest Algorithm.
33. Evaluate the Random Forest Algorithm.
34. Classify the Random Forest Algorithm to evaluate the bag-of-words.
35. The first part of the list is predicting probabilities for label:0 and the second part of the list is predicting probabilities for label:1
36. If prediction is greater than or equal to 0.3 than 1 else 0 Where 0 is for positive sentiment tweets and 1 for negative sentiment tweets.
37. Apply Random Forest Algorithm for TF/IDF feature scaling.
38. Train the TF/IDF by Random Forest Algorithm.
39. Evaluate the Random Forest Algorithm.
40. Evaluate TF/IDF for Random Forest Algorithm Classification.
41. Apply Logistic Regression Algorithm.
42. By using Bag-of-words features fit the logistic regression model.
43. The first part of the list is predicting probabilities for label:0 and the second part of the list is predicting probabilities for label:1
44. Calculate the f1 score.
45. If prediction is greater than or equal to 0.3 than 1 else 0 Where 0 is for positive sentiment tweets and 1 for negative sentiment tweets.
46. By using TF/IDF features fit the logistic regression model.
47. The first part of the list is predicting probabilities for label:0 and the second part of the list is predicting probabilities for label:1
48. Calculate the f1 score.
49. If prediction is greater than or equal to 0.3 than 1 else 0 Where 0 is for positive sentiment tweets and 1 for negative sentiment tweets.
50. Apply Decision tree algorithm for Bag-of-word-features.
51. Find score of the data.
52. Convert the results into integer type.
53. Calculate the f1 score.
54. If prediction is greater than or equal to 0.3 than 1 else 0 Where 0 is for positive sentiment tweets and 1 for negative sentiment tweets.
55. Apply Decision tree algorithm for TF/IDF features.
56. Find probability.
57. Convert the results into integer type.
58. Calculate the f1 score.
59. If prediction is greater than or equal to 0.3 than 1 else 0 Where 0 is for positive sentiment tweets and 1 for negative sentiment tweets.
60. Compare all the models and represent in the point plot with Model versus Score.