Miner ID – ajain28
Rank and accuracy – 17 and .85

Introduction –

Movie Reviews are something that is important for movie lovers, and we often see a lot of movie review websites give this opportunity for the user to view and give the review and make some decision out of it whether to watch a movie or not based on its reviews. For this Assignment, we are given a bunch of movie reviews and their label/sentiment as +1 for a positive review and -1 for a negative review. We are given test data to make our own knn classification to label the unlabeled movie reviews with utmost accuracy.

Problem Statement –

In this study, we want to classify the huge amount of movie reviews the effect of the feature selection method on movie review sentiment analysis accuracy. We will also want to know which the number of features to be used for sentiment analysis to get the best performance.

The Process-

For this process, I have divided my work into mainly two parts

a) The preprocessing
b) Applying own knn method.

Apart from preprocessing and my own implemented knn method, I have also used a feature selection process to use the best feature and reduce the

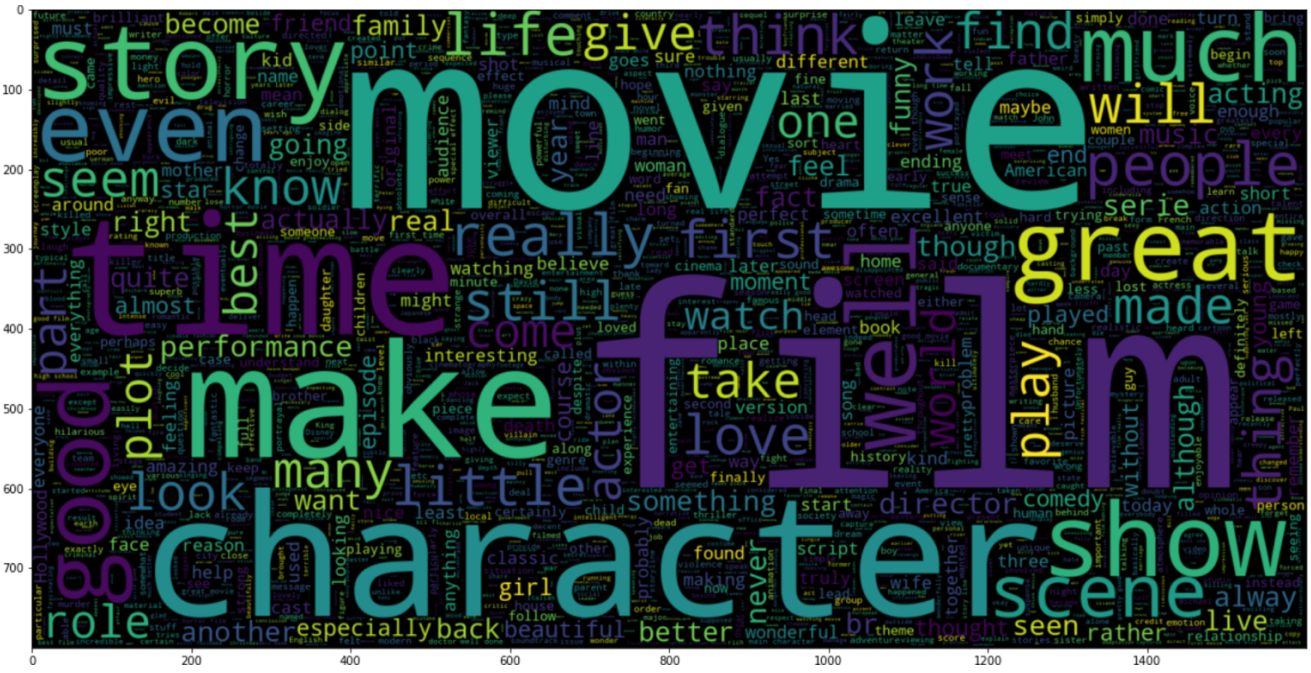
curse of dimensionality and overfitting the model, so I will detail my feature selection process in pre-processing section.

a) The Pre-Processing –

We have conducted a number of steps to analyse the sentiments, of course, the computer cannot understand English but it can understand numbers, so we have created a sparse matrix and vectorized it to feed to our model.

We are reading the training and test data, and for visual purposes, training data is split such that it has all its review on the x-axis and all the sentiments/lable data on the y-axis. We have started by tokenizing each document/review/x-axis, converting it to lower case and putting it in a set(mainly to remove duplicate words.) followed with stemming those words to its root by removing affixes such as prefix, infix and suffix. All the punctuation and stopwords in English are removed (using nltk library) from all the reviews leaving them with minimal meaningful words and that is best for sentimental analysis. Since we know that most of the sentiments are related to some heavy words, hence we have omitted words that are less than 3 letters as they make no sense. Thus, reducing words leads to the reduced number of features when we present them in a sparse matrix.

Let me show some words which are most used in an eye-catching plot –



Next in pre-processing, we have spread our documents/reviews into the sparse matrix to vectorize them among different features. We have used n-gram to analyze precise sentiments where (n=1,3), which increased our features to 1.5M. Now a huge number of attributes gives rise to curse of dimensionality which overfits our model and reduces the accuracy.

And at this point, we decide to limit our features by using the method of feature selection. We have implemented chi square feature selection to select topmost used feature in our model and it increased our precision.

b) Implementing k nearest neighbour.

Implementing knn starts with the best similarity function, here we have used cosine similarity for finding the best similar and closed elements on the graph. With that given we calculate the cosine similarity on all of the test data and going forward, we iterate over each similarity document and sort them in order of most similar data points at first.

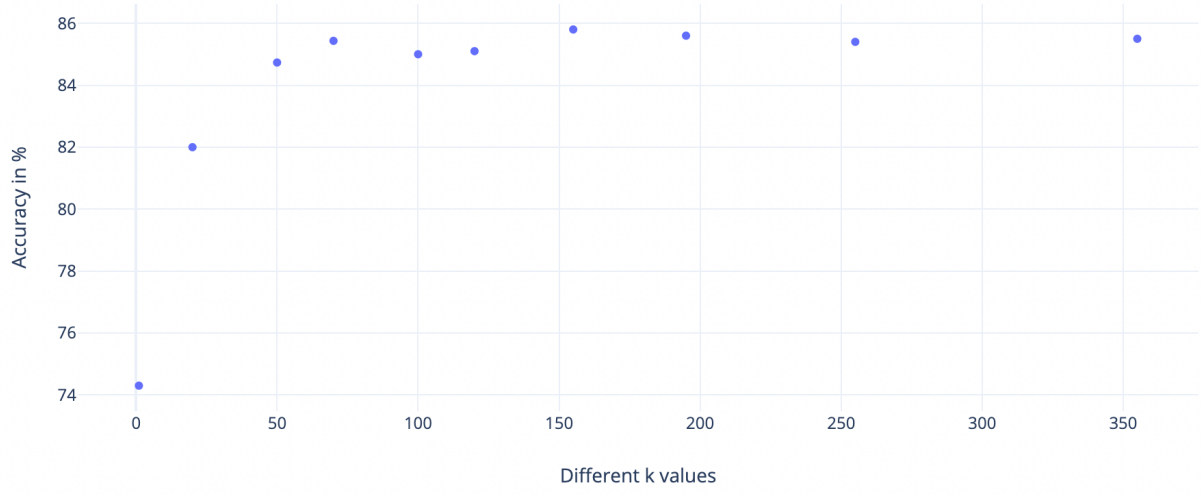
Now based on the best k value we have identified (k value would be odd mostly). Now that we have k most similar elements in our function, we will extract those top similar data point's labels/sentiments from train data (remember that y-axis) and add them up. If the addition is positive that means our test document is more similar to negative data or else, it is positive sentiment.

Suppose k = 5 and 5 nearest elements have labels as - +1, +1, -1, -1, -1, their sum would be a negative value, thus the test data is more of a negative sentiment type and thus we label that data as -1.

And that's how we are classifying our test data into positive and negative reviews.

How the best parameter/ k value of knn is selected.?

We can run some sample k values and calculate the best k value. We have looped some k values to show the best accuracy and narrowed down one k value. Here is a little plotting for various k values.



Why chi2 feature selection?

It calculates chi-square statistics between every feature variable and the target label to determine with the feature is more closely related to the label and which features are not, the ones which are not related are discarded, it is one of the simple and easy to use feature selection which nicely increases the accuracy.

Why cosine similarity?

It checks the similarity between two docs using the angle between two docs when plotted on a 2D graph. So, If we are using let's say Euclidean distance, we would only be focused on physical distance, so if the two points are far away, Euclidean distance would say that it will be not too similar (but they still could be a lot similar). The similarity coefficient index is calculated using the cosine angle between documents and if the angle is low the similarity would be higher, unlike Euclidean where they check only for the distance between points. Since this case, we get better precision and accuracy as the similarity is more accurate.