Clean Data:

In the beginning, I imported useful packages and read data from the 3 txt.file respectively After that, I used concat in pandas to merge the three table.

Then, I created an empty list to store all the words in the sentence. I used for loop to access each sentence and added all the words in current sentence to the words list.

Assumptions:

(1) Assume punctuations (!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~), number and capitalization d o not have a significant effect on the result of sentiment

(2) Assume single letter does not have a significant effect on the result of sentiment, includin g those become single letter after removing the puncuations. (i.e. the “s” after removing t he "’" from “It’s”; the “p” and “s” after removing "." from “P.S”; the “a” and “m” after re moving "." from “a.m.”; etc.)

(3) Assume pronounce “I” does not have a significant effect on the result of sentiment

(4) Assume the “part of short term of word” are meaningful. (i.e. the “ll” after moving "’" fro m “I’ll”; the “Ph” after moving "’" from “Ph.D”; etc.)

Method:

(1) Tokenize

Based on the three assumptions above, I filtered out punctuations, number, and stemmed t he capitalization. I kept the noun, verb, adj, adv, prep and the “part of short term of word” I used CountVectorizer package to do it. Through vectorizer.fit(words) I filtered punctuat ions, number, and stemmed the capitalization. Then through vectorizer.transform(sentenc e_list), I passed each sentence and get the whole matrix which stands for the times each word in the vocabulary shows in each sentence.

(2) Prune Data

At first, I tried to select the words by words whose appearances are above medium. After finding the medium is 1. I choose to prune the data by the average value.

(3) Models

I wrote the algorithm for Multinomial Naïve Bayes and also imported the package for checking the values.

Also, I imported package from sklearn to use Decision Tree, KNN, SVM models.

(4) Metrics

I use Accuracy and Efficiency cost as metrics.

For fair comparison for the efficiency cost, I calculated the efficiency cost of the Naïve B ayes from the package I imported rather than the implementation I wrote by myself. Othe rwise, the efficiency cost of Naïve Bayes would be much higher than any other models an d it does not make sense to do the comparison and draw conclusion.

Conclusion:

Based on the comparison in efficiency cost and accuracy between each model for original data and prune data.

We can conclude that

(1) Though feature selection can help us decrease efficiency cost under most case, we los e a little accuracy at the same time.

(2) Naive Bayes has the highest accuracy.

(3) Naive Bayes has the lowest efficiency cost.

Overall, even though feature selection decreases the efficiency cost a little bit (there's actually not a bit difference), we lose more accuracy after comparison. Thus, the feature selection is not very helpful. Also, we can see that the efficiency cost when we use the Na ive Bayes Model is very little, so it is recommended to use the original data.