NLP HW2 - DXP170020
Writeup Question 1.1

Pros and Cons of the three ways of handling the square brackets are as follows :

1. deleted those words completely
    a. Pros : Efficient for performing the further steps after tag_edits since we dont have to deal with the EDIT_ words again for further processing
    b. Cons : If we have deleted data, this will in turn lead to more processing for tag edits function. We would have to remove data from the snippet and hence we would be removing information from our data which in fact can be used for further information in future
2. could have removed the square brackets but left the words untagged
    a. Pros : This would have been very efficient and would save a lot of computations
    b. Cons: We would using the words that were not included by author in our training data and hence technically working on wrong data set . Since the edit words are words that are added by someone else for proper understanding and not by author.
3. The way described in the assignment
    a. Pros: We have all the information intact in our training data. We have marked the edited words as EDIT_ and hence we can further use this information if needed for future purposes.
    b. Cons: We have to do a lot of computations for the EDIT_ words which is of course decreasing the efficiency of the code

Writeup Question 1.2

I used a set for the negation ending tokens and defined the variable as follows : negation_end_tags_words = set('but','however','nonetheless',".","!","?"). The choice of my data type is because sets are significantly faster when it comes to determining if an object is present in the set as compared to list. We use set to lookup the word in order to identify when to stop the negation tagging.

Writeup Question 3.1

Precision : Precision is defined as the ratio of correctly predicted positive observation to the total predicted positive observations. If we have high precision, it means we have low false positive rate which is good . Precision is a measure of how accurate the model is out of those predicted positive, how many of them were actually positive.

Recall : Recall measures sensitivity of the model. It is the ratio of the correctly predicted positive observation to all the observations in actual class. Recall expresses the ability to identify all the relevant instances in dataset. Recall is the number of relevant documents that are retrieved. Recall is used to measure a classifiers completeness.

F1-Measure : It is the weighted average of precision and recall. Hence, it takes both false positives and false negatives into account. In case we have uneven class distribution, F1-measure helps us to understand the performance of model. If the cost of false positive and false negatives is different we should look both at Precision and recall.

We need all three of them because Precision is a good measure to determine when the cost of false positives is high. This is useful in applications like email spam detection etc. Recall is the model metric used to understand the best model when there is a high cost associated with false negatives. This is useful in cases like fraud bank transaction detection. F1-measure is used to seek balance between precision and recall and when there is an uneven class distribution. Thus precision and recall, unlike accuracy, emphasize true positives: finding the things that we are supposed to be looking for.

Writeup Question 3.2
precision for the Gaussian NB classifier is 0.627906976744186
recall for the Gaussian NB Classifier is 0.8157099697885196
F-measure for the Gaussian NB Classifier is 0.709592641261498

Writeup Question 3.3
Precision of Logistic Regression Model is 0.7763157894736842
Recall of the Logistic Regression Model is 0.7129909365558912
F-Measure of the Logistic Regression Model is 0.7433070866141732
Logistic Regression Model performs better since the precision for logistic regression is better than Gaussian Naive Bayes. Also the F-Measure is better for Logistic Regression than Gaussian Naive Bayes.
The reason that the performance of Gaussian Naive Bayes is low is due to the following reason : Generative classifiers like Gaussian Naive Bayes build a model of each class. Given an observation, they return the class most likely to have generated the observation. It assumes strong independence condition among the features. Discriminative classifiers like logistic regression instead learn what features from the input are most useful to discriminate between the different possible classes. Hence, logistic regression is much more robust to correlated features

Writeup Question 3.4
The top 10 features with their weights are as follows :
('too', -3.3631474044942853)
('bad', -2.4232039394333733)
('dull', -2.108912286315283)
('still', 1.8244021157729693)
('boring', -1.8144991903269514)
('fails', -1.7763300270937166)
('best', 1.5792222290755833)

('entertaining', 1.490690148832693)
('enjoyable', 1.4766986444135688)
('brilliant', 1.4058972354677095)

The magnitude of the weights in coefficient vector gives us the importance of the features.
A negative coefficient means that the higher value of the corresponding feature pushes the classification more towards the prediction of negative class.
A positive coefficient means that the higher value of the corresponding feature pushes the classification more towards the prediction of positive class.
The weight vector has its usage in finding the feature importance. Large positive values of any element , e.g. V[j] implies that the jth feature has higher importance in the prediction of positive class. Similarly, large negative value of any element in the weight vector, e.g. V[i] denote higher importance in the prediction of negative class. So , the absolute value of weights denote the importance (the negative sign means in the prediction of negative class and the positive sign means in the prediction of positive class).
There, we sort by the absolute weights so that we could understand which are the top k features for prediction of any class (either it be negative or positive).

Writeup Question 4.1
Precision of Modified Logistic Regression Model 0.7194444444444444
Recall of Modified Logistic Regression Model 0.7824773413897281
F-measure of Modified Logistic Regression Model 0.7596382054992764

Top 10 features are as follows :
('too', -3.4148690469569427),
('bad', -2.284879732217559),
('dull', -1.9559980404496213),
('still', 1.8371105432228885),
('boring', -1.7970924703711424),
('fails', -1.698296341035266),
('enjoyable', 1.4589362072938474),
('entertaining', 1.456439324085824),
('best', 1.4472796180787686),
('way', 1.4006989137274196)]

Here, the top weights are negative. The recall was increased from the previous model. The F-measure has also increased slightly.
We are adding 3 more features to every feature vector of dimension 1X|V|. This helps the model to learn better.

Meta Questions
Writeup Question 5.1
5 Days

Writeup Question 5.2
No

Writeup Question 6.1
Precision of New Logistic Regression Model 0.7822660098522167
Recall of New Logistic Regression Model 0.8368580060422961
F-measure of New Logistic Regression Model 0.751696065128901
Top K words are
('too', -3.442883655967257)
('bad', -2.2610143351470597)
('dull', -1.9590835586355733)
('still', 1.8192562991322758)
('fails', -1.6816706511517034)
('boring', -1.6725892417108355)
('best', 1.4145373536462087)
('entertaining', 1.4024739167178406)
('way', 1.3954666771920858)
('and', 1.3382889577589823)]

The model's performance is better than the previous score_snippet version because now we are trying to provide score for the words that are missing from the dictionary. So, we have rich features as compared to the previous one.