Text as Data

Q1 - Dataset

a) My dataset is about amazon fine food reviews. I selected this dataset because it's a large dataset which can be used for sentimental analysis, text classification, etc. This coursework main objective is about text classification, so I use this dataset. And automatic classification/labeling of this dataset can be used in practice in a lot of ways like understanding review of the particular product and discussing about the product. This will lead to improved product quality from the business people.

b) The labels of this dataset are ratings for each different food product sold and the input text to be used is the reviews given by the customer. The labels to be predicted are numerical review values ranging from 1 to 5, where 1, 2 are negative and 3 is neutral and 4, 5 are positive. No need to do preprocessing to create the labels because the number of labels was >=3 and <=10 where reviews values ranging from 1 to 5.

c) No, the dataset is not already split into a training, validation and test set. So I initiated the 60/20/20% split to create the training, validation and test set. By doing this, we could see that a positive review is the one mostly given by the customer and even so we could see that label distribution is balanced among other review ranges.

Q2 - Clustering

a) When k=5 clusters, the documents assigned to each cluster and top 5 tokens with the highest magnitude in the corresponding centroid is

```
Clusters 0:
Number of docs: 7
Sample docs:
 - ProductId
 - ProfileName
 - HelpfulnessNumerator
 - HelpfulnessDenominator
 - Score
Top terms: ['time', 'summary', 'score', 'profilename', 'productid']
Clusters 1:
Number of docs: 1
Sample docs:
 - Id
Top terms: ['id', 'userid', 'time', 'text', 'summary']
```

```
Clusters 2:
Number of docs: 1
Sample docs:
 - Text
Top terms: ['text', 'userid', 'time', 'summary', 'score']
Clusters 3:
Number of docs: 1
Sample docs:
 - reviews
Top terms: ['reviews', 'userid', 'time', 'text', 'summary']
Clusters 4:
Number of docs: 1
Sample docs:
 - UserId
Top terms: ['userid', 'time', 'text', 'summary', 'score']
```

b) The clusters do not make sense. And also there are no certain topics that appear in some but not others. Actually this is very difficult to conclude without doing more analysis. But based on the given sample documents, each cluster is too short and top terms in each cluster don't correspond with certain topics. So these clusters would be meaningless.

c)
```
True_Label  Negative  Neutral  Positive
Cluster
0           13091     7093     91863
1            6642     3125     43623
2            6304     4459     38576
3           11798     7588     53238
4           44202    20375    216477
```

d) The trends I notice from the confusion matrix is, most reviews in each cluster have positive reviews and mixed reviews, which will be imbalanced distribution labels in the dataset. And there are no clusters able to pick up on a single label, because of having mixed reviews. When a cluster includes multiple labels, some product reviews might be grouped together in the same cluster.

Q3 - Comparing Classifiers

a) The implementation shows that all classifiers are better than the most frequent baseline. And also logistic regression classifiers are better than the SVC classifier in terms of precision and F1-score. This concludes that logistic classifiers are better at finding positive reviews and SVC classifiers are better at finding negative reviews.
    1. The logistic regression classifiers overfit the training data but SVC classifiers with one-hot vectorization underfit the training data.
    2. The class distribution balanced with having more positive reviews. And cleaning the data, preprocessing the data, feature selection will help improve the classifiers.
    3. Dummy classifiers have simple baselines from the data. The most_frequent predicts majority class and stratified class predicts frequency in the dataset. Logistic regression with one-hot vectorization is linear and efficient in training time. Logistic regression with TF-IDF vectorization is the same as one-hot vectorisation and finds the important words in the documents. SVC classifiers with one-hot vectorization are very powerful classifiers but can be slow with large datasets

b) I will choose a multinomial naive bayes classifier with TF-IDF vectorization. Because multinomial naive bayes classifier is good for text classification. By using TF-IDF vectorization, I can manage the important words in the documents and across the corpus. I will evaluate the performance on the validation set and compare it to the five baselines.

Q4 - Parameter tuning

I tried tuning the regularization c values, sublinear_tf, max_featues, and min_df hyperparameters. The nine C values from 000.1 to 100,100 , so the higher the values, then better performance. The sublinear_tf is a boolean value to the term frequency counts. And the max_features is the maximum number of features in vectorizers with five values from 0 to 50,000. And the min_df is the minimum number of documents which will be included in the vocabulary.

Q6 - Conclusion and future work

a) The best approach would be logistic regression model with TF-IDF vectorization and same tuned with parameter tuning.
b) After manually examining the predictions on the test set, we can analyze the confusion matrix and for the pattern, where we can see that the model is working well with higher predictions in the negative class. And there are also false positives and false negatives, so the classifier may not be perfect. In such cases,

error analysis can identify the misclassifications in the model. Like, in some cases, the model might tell a negative statement as a positive statement because of misunderstanding words. And another error might be having wrong word or rare words used in the training set.

c) The final performance of this classifier is perfect. But with the false negatives and false positives can have an impact on the deployment of this machine learning pipeline, where if the classifier is used to identify fake cases, then false positives will result in a good person to be a fraud and fake negative will result in fraud but being good. So using the confusion matrix, I can see that false negatives and false positives are balanced, then both errors can have an impact on the deployment.
d) There is a possibility that deployment of this system can have negative societal effects. If the classifier is used to find a good candidate for the job, then it might result in getting unqualified people.
e) To improve the classification effectiveness of the system, we can use these steps:
   Collect more training data, trying with more advanced models like pre-trained language models, and performing more extensive hyperparameters.
f) I spent over 35 hours on this coursework.

Q5) - Research paper report

I chose *Latent Dirichlet Allocation* paper

<u>Background and context</u>

The paper addresses the issue of topic modeling in text corpora, which is aiming to find the hidden semantic structures within large collections of documents. Topic modeling is a basic challenge in Natural Language Processing(NLP) and Information Retrieval. So before this paper published, methods like probabilistic latent semantic indexing(pLSI) and term frequency-inverse document frequency (TF-IDF) were used in the topic modeling, but these methods had limitations in terms of scalability, generalization, and the ability to address the unseen documents.

<u>Contributions</u>

This paper introduces Latent Dirichlet Allocation(LDA), which is a generative probabilistic model for topic modeling. So the main contributions are:

- initiated the proposal for LDA as a fully generative model to the complete collections of discrete data, this will lead to better generalization and the ability to address the unseen documents.

- And explained the use cases of LDA in terms of easy to use, scalability with the comparison to previous methods like pLSI
- And also giving an algorithm to perform variational inference on the LDA model, which leads to improved estimation of model parameters.

Critique

The result given in the paper supports the conclusion that LDA will be an effective and scalable method than previous methods for topic modeling. The author also displays that LDA can perfectly find the hidden topic structures in various documents collections.

Positives:

1. LDA is actually a generative model, which leads to better generalization and ability to address unseen documents.
2. The paper display that LDA can perfectly find the hidden topic structures even when having large documents.
3. LDA gives a more effective and easy topic representation compared to previous methods.

Negatives:

1. The variational inference algorithm given in the paper can be expensive, usually when having large documents
2. LDA takes a fixed number of topics, which might not be a good idea for different corporations.

Writing Quality

The writing quality of the paper is pretty good and clear and concise with explaining all the important aspects of the paper and good explanation on the contributions and critique.