**Observations**

**LM\_full:**

Trained this model on all of the data and tested it on the toxic subset of the test data, non toxic subset and the complete test data. Observed the following when trained on all the train data:

|  |  |  |
| --- | --- | --- |
| Sl no | Test Data Set | Average MLE score |
| 1. | The complete test Dataset | 3.29 |
| 2. | Toxic Dataset | 1.73 |
| 3. | Non – Toxic Dataset | 3.65 |

**Observations and Analysis:**

* Based on the above observations we can see that when the language model is trained on the complete train dataset its gives the highest Average MLE score for the Non Toxic Dataset
* This test dataset contains majority of non-toxic comments.
* The number of toxic comments labeled 1 are 6090 comments in the test dataset
* The number of non-toxic comments labeled 0 are 57888 in the test dataset
* The train dataset also contains a majority of nontoxic comments with non toxic comments summing up to 144277 in the train data set and toxic comments summing up to 15294 comments
* Therefore, the train dataset(LM\_full) gives the best average MLE score on the non toxic test dataset because the train data set has majority of non-toxic comments

**LM\_not:**

Trained this model on a subset of the train data with labels 0(non-toxic). Tested it on the toxic subset of the test data, non-toxic subset and the complete test data. Observed the following:

|  |  |  |
| --- | --- | --- |
| Sl no | Test Data Set | Average MLE score |
| 1. | The complete test Dataset | 3.21 |
| 2. | Toxic Dataset | 1.23 |
| 3. | Non – Toxic Dataset | 3.67 |

**Observations and Analysis:**

* We can observe that when the model is trained on the non toxic subset it gives the best average MLE score for the non toxic test dataset. This is an observation I had expected since both train and test datasets in this case contain non toxic comments
* The Toxic Dataset has the least Average MLE score
* The complete test Dataset has a slightly less average MLE score compared to the Non-Toxic test Dataset due to the presence of toxic comments in it.

**LM\_toxic:**

Trained this model on the toxic subset of the train data with labels 1 for the toxic field. Tested it on the toxic subset of the test data, non-toxic subset and the complete test data. Observed the following:

|  |  |  |
| --- | --- | --- |
| Sl no | Test Data Set | Average MLE score |
| 1. | The complete test Dataset | 2.50 |
| 2. | Toxic Dataset | 1.82 |
| 3. | Non – Toxic Dataset | 2.67 |

**Observations and Analysis:**

* Here the toxic train dataset performs better with the complete test dataset and the Non-Toxic test Dataset as compared to the Toxic test Dataset
* This was not an expected output
* One probable reason could be that toxic words might not be depended on the previous word. Since we use a bigram model here the toxic test Dataset probably has the lowest MLE score always. A unigram model might perform better and give a higher score since unigram models do not depend on the previous word. Further analysis is required.

**Further analysis to check if the unigram model performs better when toxic train data is tested on toxic test data.**

I did an additional analysis to check if the unigram model performs better with the toxic data. I trained a unigram model using only toxic data and tested it on all three datasets:

The table is listed below:

|  |  |  |
| --- | --- | --- |
| Sl no | Test Data Set | Average MLE score |
| 1. | The complete test Dataset | 0.25 |
| 2. | Toxic Dataset | 0.14 |
| 3. | Non – Toxic Dataset | 0.26 |

The toxic train dataset again gives a lesser Average MLE score for the Toxic test Dataset compared to the other two datasets. There could be two possible cases :

1. Toxic words do not depend on previous context of the word which is why we ge a lower MLE score when a bigram model is used on the toxic test dataset
2. Also individually in a comment of this dataset the toxic words are less frequent compared to other words which is why the unigram model also gives a lesser average MLE score for the toxic test dataset compared to the other two datasets.

**This analysis is not in the code I have submitted since it is not part of MLE bigram score calculation. It follows the same logic as the bigram model but is trained on unigram model**

**Summary and Analysis**

* While calculating the MLE score we smoothened very small values in order to avoid the compiler from throwing errors and taking time during execution. If lm.score was a very small value lesser than 1e-6 we automatically converted these values to 0
* We observed a deviation from the expected output while testing out bigram models using toxic data.
* We observed that when toxic data is used to train a bigram model it gives a smaller Average MLE score on toxic test data compared to the entire test dataset and non-toxic test dataset. We assumed that this could be the case since toxic words do not depend on previous context. While testing out the unigram model also we got a smaller average MLE score while testing toxic train data on toxic test data
* The other language models performed as expected and their observations and analysis are listed above