1. Data processing

I added my own variable text cleaning method to clean things such as emails, if they were harming performance.

An additional feature: removing stop Words for isiXhosa such as:'na', 'kwaye', 'kodwa'

- I implemented this to test if it would improve performance and to check how it affects the size of the vocabulary for the different feature extraction methods also because the two languages are so different so what could be a valid stop Word for the English language could be a very important word for isiXhosa.

2. Multinomial logistic regression implementation

For this class the only additional feature I added based on the assignment specifications was the “**get_weights()**” function to return the weights for later analysis.

3. Training

For the training I implemented it as required i.e. using gradient descent to minimize the cross-entropy loss

As part of my training, I had to search for the optimal stopping conditions for each language based on validation accuracy, and my findings were as follows:

(The different conditions I tried were early stopping with a patience of 5 or fixed number of epochs using 50,80 or 100 epochs)

- English: the model benefited mostly from early stopping, on average I would get the highest validation accuracy from the early stopping criterion
- IsiXhosa model: this model mostly benefited from training for many epochs which could be because it has smaller datasets for some classes
- Initially when I had just trained it for 50 or 100 epochs it gained the highest accuracy after training for 50 epochs but adding 80 leads to producing varying results but always having the highest validation accuracy either for 50 or for 80 epochs
- For these reasons I chose early stopping with a patience of 5 for the English model and chose a fixed number of 50 epochs for the isiXhosa model.

4. Hyperparameter tuning & 5. Feature extraction I arranged my experiments to be as follows:

I tuned the hyperparameters so that find the best model for each language and feature extraction method : for example in the end for English I had 3 best models , one for when using BAG OF WORDS ,another for when using BINARY and the last for when using TFIDF , I did the same for isiXhosa.

What Follows are each of the hyperparameters and how they each affect models:

Vocabulary size: For this Hyperparameter I had two settings

- Maximum features: 5000 (keeping the min_df =1)
- And minimum frequency cutoff (min_df) of 5 while having no maximum feature

Using the minimum frequency cutoff of 5 while keeping the maximum features unbounded (having a vocabulary of 6058 words) was more effective for isiXhosa as this meant that I had more words for training, thus longer training time, this can be seen in figure A with the IsiXhosa model achieving it highest accuracy while in this setting ,as established earlier this language benefits from longer training times. The same can be said for English that benefitted from both settings

Batch Size: I experimented with the batch sizes 25, 32 ,40 and 64

I learnt that the two lower batch sizes 25 and 32 were more stable and consistently yielded a higher accuracy, this can be seen in the images in **Figure D** of the Appendix, where there is not a single model out of all six optimal models that benefits from larger batch sizes

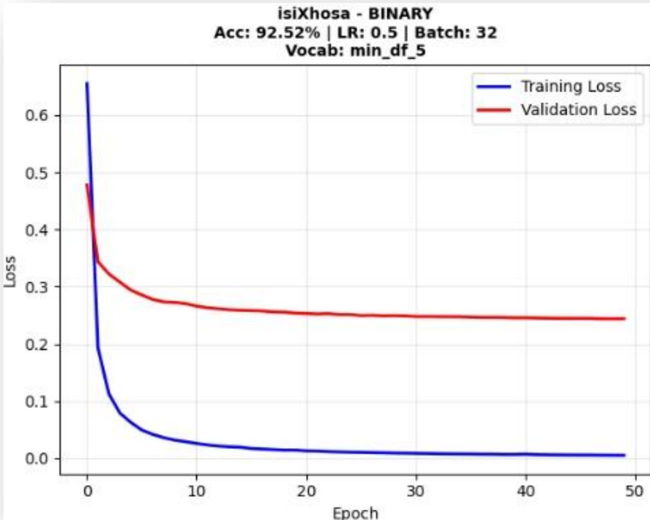
**Note: This led me to eliminating the batch sizes 40 and 64 from my notebook to lessen to runtimes for the tutors

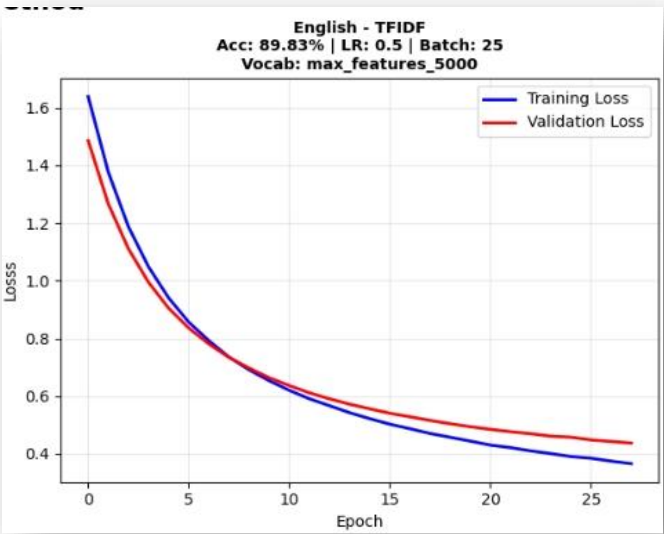
Learning rates: My initial values for experimenting were 0.1, 0.5 and 1.0

smaller learning rates typically <0.5 proved to take too long while any value between 0.5 and 1 lead to faster training times and converging earlier.

Throughout all my experiments the isiXhosa models were more effective when using the binary feature extraction method, meaning that for this language presence of words mattered more, the images in **Figure D** corroborate this statement with the 93.20% validation accuracy (highest in all my experiments) being in the BINARY method.

These are the best models for each Language and feature extraction method:





Looking at the English model in the above image and **Figure E** in the Appendix we can see that the TFIDF feature extraction is more potent in the English language than it is in isiXhosa , other than a very high validation accuracy, the gap between the validation loss and training loss in the plots is smaller for English meaning the model generalizes well, I then evaluated this English model on the test set and got the results in **Figure B** of the appendix.

6. isiXhosa training decisions

L1 and l2 regularization:

Both L1 and L2 performed poorly but decreasing the regularization coefficient improved performance but even so the highest validation accuracy attained was equal to the already existing best accuracy of 92.52%, when the value of alpha(α) was :0. 00001 or smaller

Table of varying coefficients and their results:

Alpha (α)	**L1(Validation accuracy)**	**L2(Validation accuracy)**
0.001	89.12%	91.84%
0.0001	89.12%	92.52%
0.00001	92.52%	92.52%
0.000001	92.52%	92.52%

Any higher than 0.001 had lower validation accuracy and was thus prone to overfitting and any lower than 0.000001 had no higher than 92.52% so we saw no improvement using this method

Handling class imbalance: Please refer to **figure A** in the Appendix(page6)

I applied the following sampling rates: 0.25, 0.5, 1.0, 2.0, 4.0 for both up-sampling and down-sampling and chose the best method based on validation accuracy: The best configuration was up-sampling with a rate of 0.5x providing an increase of 1.36% in validation accuracy from the original best model.

With this sampling rate applied in can be seen from **figure A** that the model did have a higher validation accuracy but had the same test set accuracy as the original model we got from tuning, this meant that even though by a small margin sampling does lead to overfitting on the validation set.

Up-sampling did improve the minority class performance, i.e. +7% increase for the Business class for recall and f1

7. Evaluation: I evaluated 3 models on the test set: the best for English and the best for isiXhosa with up-sampling applied and without up-sampling applied the results are in **figures A** and **B** of the appendix

IsiXhosa model saw very little improvement going from the original to the model with up-sampling thus I will mostly use the up-sampled model to report:

There is a significant gap between the micro and macro f1-scores with the micro f1-score being 0.8956 (89%) and the micro f1-score being 0.7816 (78%). This reveals the class imbalances as can be seen in the per class metric smaller classes such Business having a recall of 0.47 and an f1of 0.64 while majority classes such sports, entertainment perform excellently having f1-scores ranging between 0,90 to 0.98 and recall scores ranging from 0.94 to 0.97

As stated even without up-sampling isiXhosa performs the same, with the only difference being numerical values of the differences between the macro and micro averages.

Business recall does improve from 0.4 to 0.47 after applying up-sampling with 0.5x rate

English model outperformed both isiXhosa and had more balance between classes such that micro averages ≈ macro averages.

For this model the most misclassified category is the Business category, and it is being misclassified for Technology most of the time leading to Business having higher precision (0.91), but lower recall (0.84) compared to Technology class: lower precision (0.86), but higher recall (0.89).

8. Weight analysis: Evidence in figure C in Appendix

Figure C clearly shows the models having learned words such "`ishishini`" meaning business and "`oosomashishini`" the meaning businessman/woman for the Business category and assigning large positive weights for these words which shows it associates the correct words to the class. But an example of the model learning incorrect associations is the word "nge" being assigned a moderately positive weight as this word translates to "with" majority of the times when used in isiXhosa so it is no different than a stopping word in English and should not be used as one of the distinguishing words while words such "afrika" are assigned negative weights of "`-0.2917`". This was my motivation for experimenting with removing isiXhosa stop words, but for isiXhosa this had very little impact as after not removing the stop words: the word "nge" was replaced by "sebe" with a weight of "`0.3340`" and this word is in similar standing as "nge".

APPENDIX

Figure A: two of my best IsiXhosa models

isiXhosa evaluation without 0.5x up-sampling applied	isiXhosa evaluation with 0.5x up-sampling applied

```
Final validation accuracy: 89.56%
Validation Accuracy (from tuning): 92.52%


Final Test Set Evaluation:
Test set acuracy: 89.56%


==================================
MODEL METRICS!!
==================================
Model: BINARY | LR: 0.5 | Batch: 32
Vocabulary: min_df_5


Micro-Averaged:
   Precision: 0.8956
   Recall:    0.8956
   F1-Score:  0.8956


Macro-Averaged :
   Precision: 0.8809
   Recall:    0.7375
   F1-Score:  0.7720


Per-class Metrics:
               precision    recall  f1-score


     business       1.00      0.40      0.57
entertainment       0.89      0.95      0.92
       health       0.73      0.40      0.52
     politics       0.79      0.97      0.87
       sports       1.00      0.97      0.98


     accuracy                           0.90
    macro avg       0.88      0.74      0.77
 weighted avg       0.90      0.90      0.89
```

```
Final validation accuracy: 93.20%
Test set acuracy: 89.56%


==================================
MODEL METRICS!!
==================================
Model: BINARY | LR: 0.5 | Batch: 32
Vocabulary: min_df_5


Micro-Averaged:
   Precision: 0.8956
   Recall:    0.8956
   F1-Score:  0.8956


Macro-Averaged :
   Precision: 0.8703
   Recall:    0.7489
   F1-Score:  0.7816


Per-class Metrics:
               precision    recall  f1-score


     business       1.00      0.47      0.64
entertainment       0.90      0.94      0.92
       health       0.67      0.40      0.50
     politics       0.79      0.97      0.87
       sports       1.00      0.97      0.98


     accuracy                           0.90
    macro avg       0.87      0.75      0.78
 weighted avg       0.90      0.90      0.89
```

Figure B:
Metrics for the best English model

```
Early stopping at epoch 39
Final validation accuracy: 91.56%
Validation Accuracy (from tuning): 89.83%


Final Test Set Evaluation:
Test set acuracy: 91.14%


==================================
MODEL METRICS!!
==================================
Model: TFIDF | LR: 0.5 | Batch: 25
Vocabulary: max_features_5000


Micro-Averaged:
   Precision: 0.9114
   Recall:    0.9114
   F1-Score:  0.9114


Macro-Averaged :
   Precision: 0.9097
   Recall:    0.9076
   F1-Score:  0.9081
```

Figure C:
IsiXhosa learned weights analysis(snippet)

```
============================================
Weight Analysis and Extraction for isiXhosa
============================================
Model weights shape: torch.Size([5, 6058])
Number of classes: 5
Vocabulary size: 6058


Class: BUSINESS
----------------------------------------------
----
Top 10 words that strongly indicate
'business':
  ishishini: 0.6412
  oosomashishini: 0.5361
  amashishini: 0.4857
  amafama: 0.4765
  lezolimo: 0.3770
  asakhasayo: 0.3719
  ushishino: 0.3638
  shishini: 0.3605
```

```
Per-class Metrics:
               precision    recall  f1-score

     business       0.91      0.84      0.87
entertainment       0.91      0.89      0.90
       health       0.96      0.90      0.93
     politics       0.89      0.94      0.91
       sports       0.93      0.97      0.95
   technology       0.86      0.89      0.88

     accuracy                           0.91
    macro avg       0.91      0.91      0.91
 weighted avg       0.91      0.91      0.91
```

```
  kudizwe: 0.3382
  nge: 0.3338

Top 10 words that strongly indicate any other
class:
  lo: -0.3945
  ngalo: -0.3237
  afrika: -0.2917
  akukho: -0.2536
  uza: -0.2439
  emva: -0.2434
  ka: -0.2198
  bafuna: -0.2152
  lezempilo: -0.2065
  abantwana: -0.2062
```

Figure D: one of the results from my experiments

```
======================================
BEST MODELS FOR - ENG
======================================

 BOW:
  Validation Accuracy: 89.62%
  Learning Rate: 0.5
  Batch Size: 25
  Vocabulary Setting: min_df_5
  Vocabulary Size: 12754

 BINARY:
  Validation Accuracy: 88.56%
  Learning Rate: 0.5
  Batch Size: 25
  Vocabulary Setting: min_df_5
  Vocabulary Size: 12754

 TFIDF:
  Validation Accuracy: 90.68%
  Learning Rate: 1.0
  Batch Size: 25
  Vocabulary Setting: min_df_5
  Vocabulary Size: 12754
```

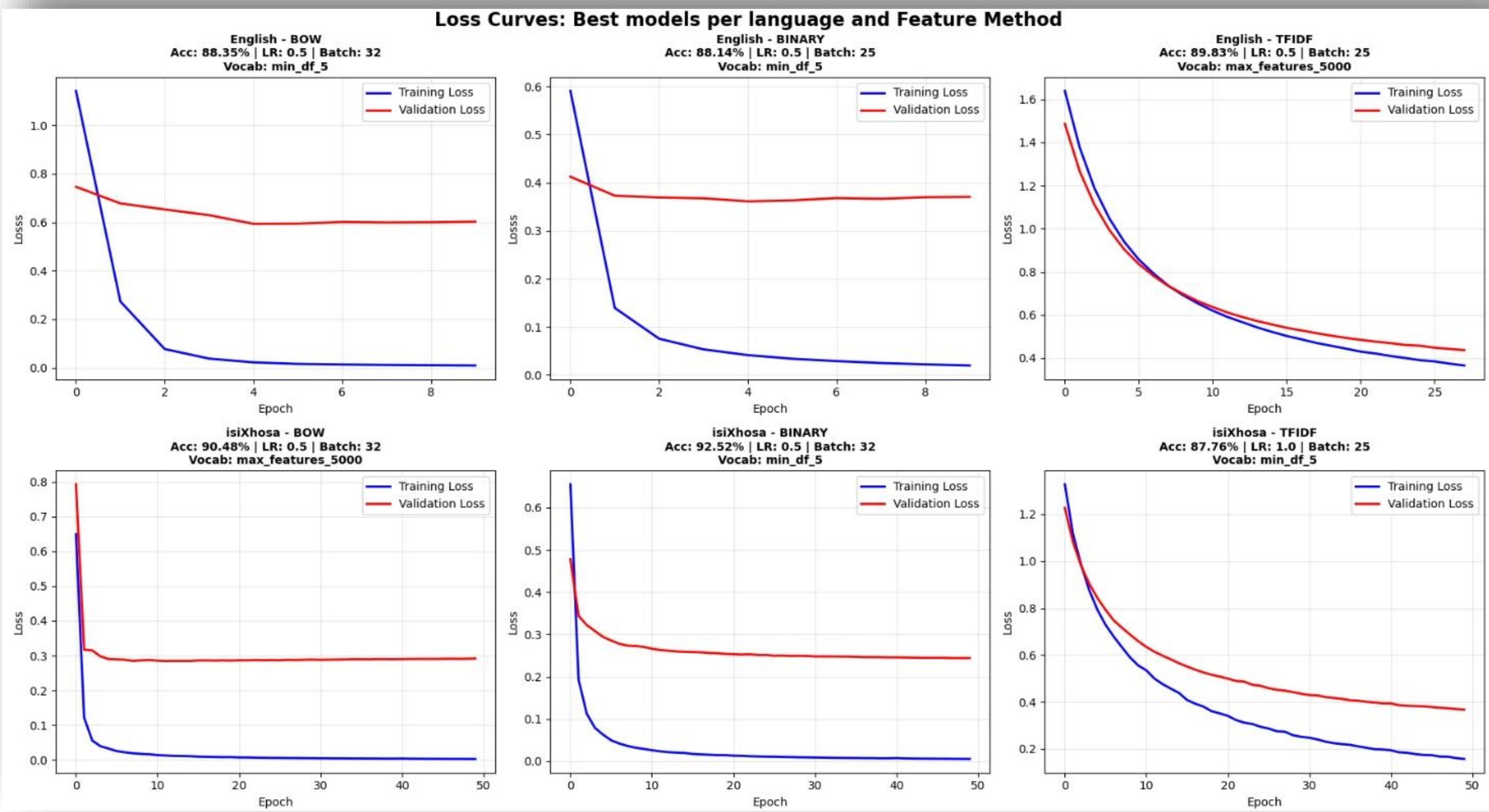
```
==========================================
BEST MODELS FOR - XHO
==========================================

 BOW:
  Validation Accuracy: 90.48%
  Learning Rate: 0.5
  Batch Size: 32
  Vocabulary Setting: max_features_5000
  Vocabulary Size: 5000

 BINARY:
  Validation Accuracy: 93.20%
  Learning Rate: 0.5
  Batch Size: 32
  Vocabulary Setting: min_df_5
  Vocabulary Size: 6058

 TFIDF:
  Validation Accuracy: 87.76%
  Learning Rate: 1.0
  Batch Size: 32
  Vocabulary Setting: min_df_5
  Vocabulary Size: 6058
```

Figure E: six best plots from hyper-pameter tuning 3 for English and 3 for isiXhosa(below)


Loss Curves: Best models per language and Feature Method
English - BOW
Acc: 88.35% | LR: 0.5 | Batch: 32
Vocab: min_df_5
English - BINARY
Acc: 88.14% | LR: 0.5 | Batch: 25
Vocab: min_df_5
English - TFIDF
Acc: 89.83% | LR: 0.5 | Batch: 25
Vocab: max_features_5000
isiXhosa - BOW
Acc: 90.48% | LR: 0.5 | Batch: 32
Vocab: max_features_5000
isiXhosa - BINARY
Acc: 92.52% | LR: 0.5 | Batch: 32
Vocab: min_df_5
isiXhosa - TFIDF
Acc: 87.76% | LR: 1.0 | Batch: 25
Vocab: min_df_5
Training Loss
Validation Loss
Losss
Loss
Epoch