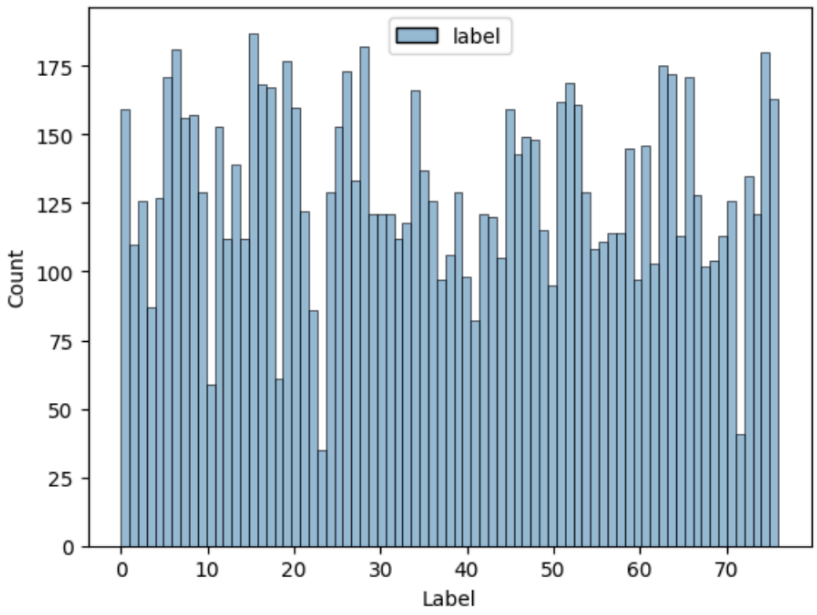
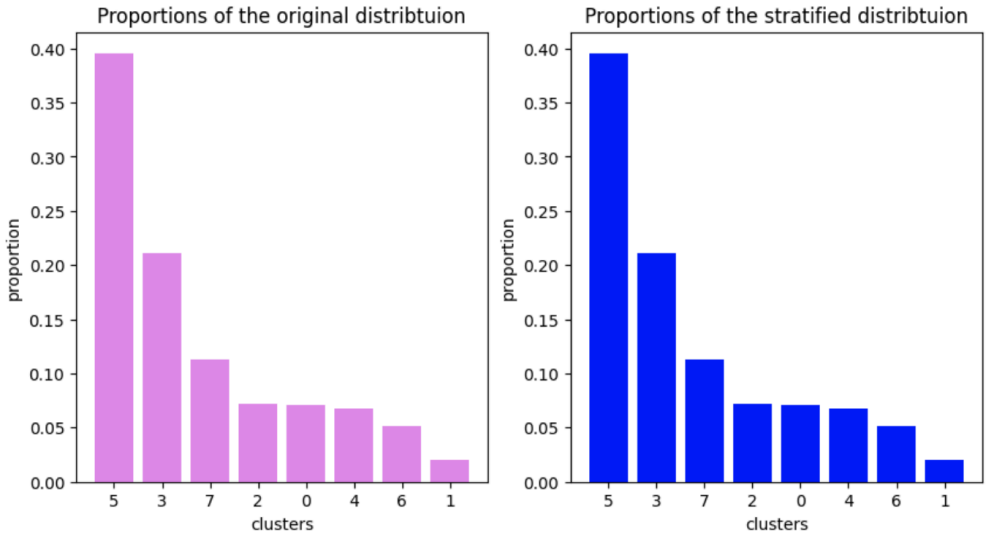
1. Initially, I noticed the datasets having a class imbalance as can been seen in the following histogram that plots the frequencies for each of the 77 classes:



2. After cleaning the datasets, lemmatizing them, I used a countvectorizer to convert the texts into sparse, numerical matrices.

3. I clustered the 77 classes into 8 distinct clusters using KMeans clustering. I used the elbow method to determine the optimal number of clusters.

4. I stratified the training dataset to ensure the clusters are sampled proportionally into the training and validation sets:



5. I built 3 models (a model with two layers, a model with 3 layers, and a model with 6 layers) and evaluated the training accuracy and loss, validation accuracy and loss, and testing accuracy and loss on 77 classes, 8 clusters using KMeans, and 4 clusters using Hierarchical clustering.

6. I visualised the confusion matrices for all three models that provided me more insights about the model's performance on specific clusters evaluating the instances of true positives, false positives, true negatives, and false negatives.

7. I reduced the dimensions of the datasets using PCA and UMAP. I noticed that the model's ability to generalize well and the time for training improved because of dimensionality reduction so I could decrease the learning rate to further enhance training.

8. I used hierarchical clustering to cluster the datasets into 4 distinct clusters.

9. I experimented with multiple different models: support vector machines, Naive Bayes, and Random forest, and noticed that the model accuracy differs across the models. I have also visualized the confusion matrices for all the models.

10. I performed LIME analysis (for interpretability) from which I could observe how different features were contributing to the classification results.

11. Alternatively, we could replace the vector representations from countvectorizer with Bert CLS embeddings to capture the underlying nuances in the dataset in more detail.