Data Processing

We performed feature engineering and feature selection to prepare data for machine learning modeling. Feature Engineering: To prepare the data for modeling, we performed several feature engineering steps:

1. I used the NLTK library to preprocess the text data (comments)
    a. Contraction like taking care of the {"ain't": "are not","'s":" is","aren't": "are not"}
    b. Removed the stop words
    c. Stem words
    d. Lemmatize words
    e. Remove empty spaces
    f. Remove numbers
2. I made the assumption that those comment made has no relationship between the comments themselves and time when they were posted so I didn't include that in my data but this could be changed to include multivariate time series data for classification.
3. In the next iteration we can add a classification fitness to the latent space such that the latent space can be regularized for better classification. For example we can attach a mlp to the latent space to get the scores or upvote ratio.

Methodology

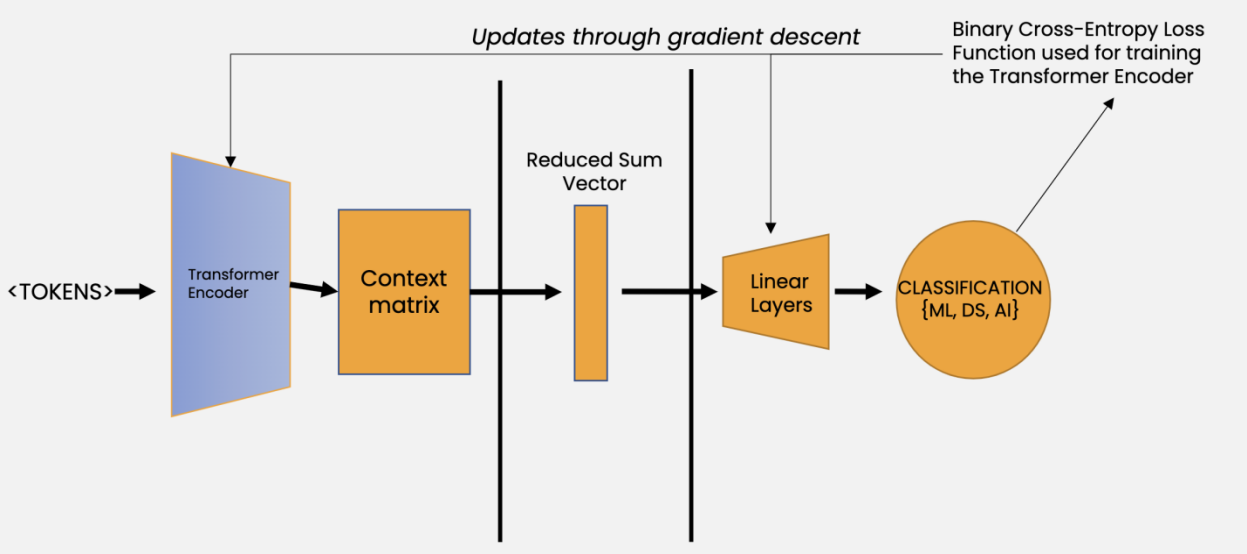
I made extensive modeling research in this project and developed a novel Transformer-Based deep learning model for classification, which incorporated one innovative contributions in its design.

Current version includes a transformer encoder only model with a classification head. The input to the encoder are token which are converted from the collection of words from all the text and assigned an individual unique number. I provided the model below

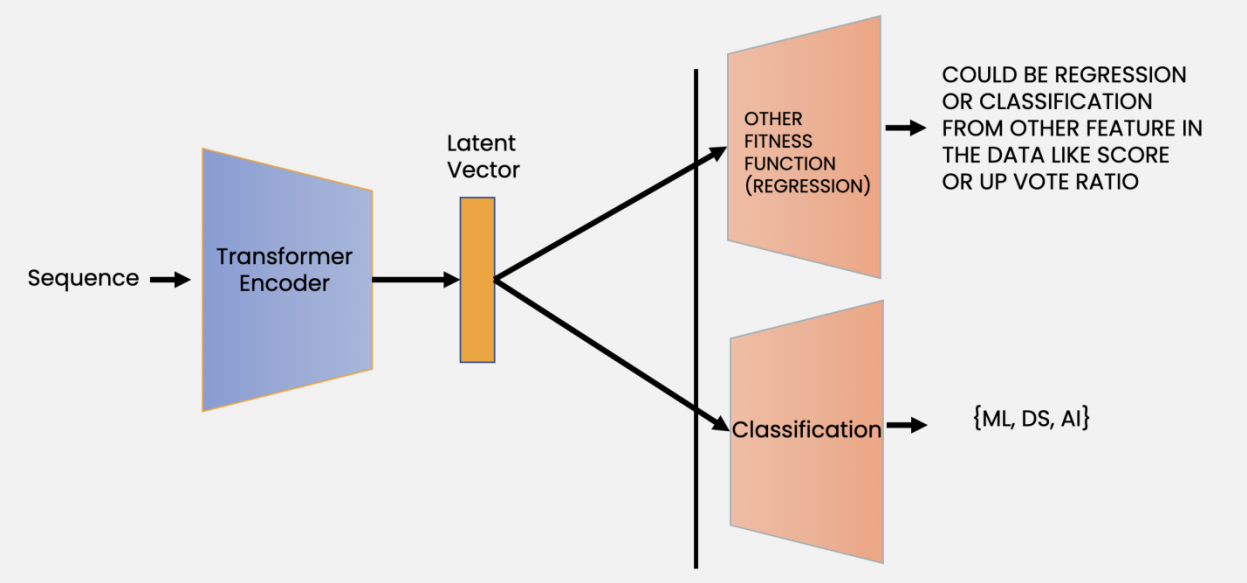
1. The transformer encoder has dimension of 2 transformer blocks
2. Encoder has 8 attention heads
3. Embed dimensions are 80
4. Latent vector dimension are 80
5. For classification I am using a mlp with 2 layers with 512 unit and 128 units

Already existing: Bert model approach was to use mean to reduce the context vector to get the latent vector and it has been trained on masked inputs.

Contribution: Instead of using masked input to train to predict the missing token I just didn't mask and I used sum to reduce the context vector and attached it to a classification head for this classification. For loss function I used cross entropy loss. The figure below shows the model architecture

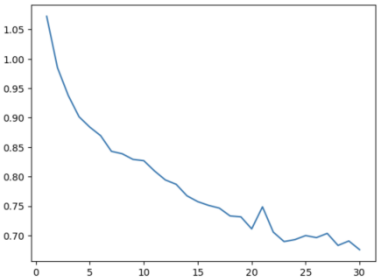


Future Iteration: With the classification head also add fitness function to get a better representation of the latent space. Combined loss (sum of weighted average) is used to adjust all the networks.



Conclusion:

Below includes the training result of different training loss during different hyperparameter experiments. The y-axis is the training loss and the x-axis is the number of epochs



For the encoder model I referred to the paper “Attention is all you need” and I built the model using there architecture expect I didn’t use as many transformer block or heads mentioned in the paper.
https://arxiv.org/pdf/1706.03762.pdf

All my files are in the github link I provided below
https://github.com/nammi-bharani/Subreddit-Text-Classification