**Toxic Comment Classification**

**Report**

1. **Dataset Description:**

The Toxic Comment Classification dataset provides a large number of Wikipedia comments which have been labeled by human raters for toxic behavior. The types of toxicity are:

* Toxic: general toxicity
* Severe\_toxic: highly offensive or threatening.
* Obscene: contains obscene language.
* Threat: contains threats of violence or harm.
* Insult: contains insulting language.
* Identity\_hate: contains hate speech directed at identity groups.

1. **Statistical analysis of dataset:**

Distribution of Comment Labels:

A group of blue bars

Description automatically generated

There is a significant class imbalance for all the labels. The majority of comments are not labeled as toxic, severe toxic, obscene, threat, insult, or identity hate.

Correlation Matrix:

A screenshot of a color chart

Description automatically generated

* Positive Correlations: Positive values (closer to 1) suggest that when one label occurs, there is a higher chance of another label also occurring.
* Negative Correlations: Negative values (closer to 0) indicate that when one label occurs, another label is less likely to occur.
* Based on matrix, **most of the toxic comments** are likely to be ‘toxic’, ‘obscene’ or ‘insulting’.

1. **Algorithm Choice**

The algorithm used for this project were the Deep Learning Model using a Bidirectional LSTM (Long Short-Term Memory) network.

* Why **LSTM**? LSTM networks are a type of recurrent neural network (RNN) capable of learning order dependence in sequence prediction problems, well-suited for text classification tasks.
* Why **Bidirectional**? A Bidirectional LSTM reads the input sequence in both forward and backward directions, capturing context from both past and future tokens.

Features and Parameters:

* Text Vectorization (`max\_tokens=200000`, ` output\_sequence\_length=1800`)
* Stopwords removal
* Model architecture:
* Model type: Sequential
* Embedding Layer (16 dimensions)
* Bidirectional LSTM Layer (16 ‘neurons’)
* Dense Layers (64 and 32 units)
* Output Layer (units)

All the parameters were determined by testing on the first epoch of the model to achieve faster learning for the model. Speed went up from 700ms to 1 sec.

References:  
Tutorials used:

<https://www.youtube.com/watch?v=ZUqB-luawZg&t=3436s>

<https://www.geeksforgeeks.org/removing-stop-words-nltk-python/>

Dataset:

<https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data>

1. **Results**

**Loss:**

During the training, loss went from first epoch with loss 0.0461 to fifth epoch with loss 0.0330, showing the improvement over time.

**A graph with blue and orange lines

Description automatically generated**

**Evaluation results:**

* Precision: 0.877 (or 87.7%)
* Recall: 0.688 (or 68.8%)
* Accuracy: 0.165 (or 16.5%)

**GUI**:

Made with Gradio, perfectly working.

1. **Summary**

* Precision (Good): The model is good at correctly predicting toxicity when it does so.
* Recall (Moderate): The model is reasonably good at finding most toxic comments but can improve.
* Accuracy (Low): Given the nature of the problem (multi-label classification with imbalanced classes), low accuracy isn't necessarily result of poor model performance.
* Focusing on precision and recall is more appropriate for this task than accuracy. The model can be considered reasonably good, but improving recall could make it even better at identifying all toxic comments.
* To improve the current model, there should be:
* Increase the LSTM Units (from 16 to 64 or 128)
* More Epochs with Early Stopping
* Hyperparameter Tuning