Project Title:

Toxic Comment Detection Using DistilBERT

1. Student Details

- **Name:** Sadman Sakib
- **Student ID:** 48031275
- **Email:** sadman.sakib2@students.mq.edu.au

2. Introduction: What This Project Is About

The internet is a powerful place for communication, but sometimes people use it to say bad or hurtful things. These are called **toxic comments**. Toxic comments can include hate speech, threats, insults, and other harmful messages that can cause emotional damage to others. These comments can be seen on social media, blogs, news websites, and forums.

Most websites try to remove such content to keep their communities safe and respectful. However, it is hard to check millions of comments every day manually. That's why machine learning can help, it can find these bad comments automatically.

In this project, I built a machine learning model that **automatically detects toxic comments**. I used a popular **pre-trained language model called DistilBERT** and fine-tuned it using a labeled dataset of online comments. The goal was to make a system that can decide if a comment is toxic or not based on its text.

This kind of system can be useful for:

- Social media moderation
- Blog or forum filters
- Comment flagging tools
- Real-time chat safety features

3. Technologies and Tools Used

Instead of training a model from the beginning (which takes a lot of time and resources), I used a **pre-trained model**. A pre-trained model has already learned how English works. It was trained on huge amounts of text like books and Wikipedia. It understands grammar, word meanings, and sentence structures.

I chose **DistilBERT** because:

- It is faster and smaller than BERT
- It works well on small datasets
- It fits well in Google Colab (free GPU)

I used the HuggingFace transformers library, which made it easy to load the model, tokenize the text, and train. I also used:

- **Google Colab:** for training and testing
- **Python:** main language
- **Matplotlib:** for drawing graphs
- **Scikit-learn:** for evaluation metrics like accuracy and confusion matrix

4. The Dataset and Data Preparation

I used the **Jigsaw Toxic Comment Classification dataset** from Kaggle. It includes comments written by real users. Each comment is labeled as:

- **1 = toxic**
- **0 = non-toxic**

4.1 Dataset Details:

- Total comments: 160,000+
- For my project, I used a **smaller subset of 50,000 samples** due to Colab time limits.
- Each row had a comment and a label.

4.2 Data Cleaning:

I removed rows where the comment text was missing or empty. I also checked for unusual lengths.

4.3 Tokenization:

I used the **AutoTokenizer** from HuggingFace. This tool splits each comment into smaller parts (called tokens), which are numbers that the model understands. I also checked the average token length to make sure it was below 512, which is the maximum for DistilBERT.

Example:

Comment: "You are so stupid!"
Tokens: ['you', 'are', 'so', 'stupid', '!']

5. Fine-Tuning the Model

The main part of the project was to **fine-tune** the pre-trained DistilBERT model. Fine-tuning means updating the model with new data to make it good at a specific job, in my case, classifying comments as toxic or not.

5.1 Training Process:

- I split the data: **80% for training**, **20% for testing**.
- Used Trainer API from HuggingFace to run the training.
- I used 2 epochs (loops through the data), which was enough for the small dataset.
- I used Cross Entropy Loss for binary classification.

The labels were adjusted to match model expectations:

- 0 for clean
- 1 for toxic

Training took around 1 hour in Google Colab with GPU.

6. Model Evaluation and Results

After training, I tested the model using the 20% test set. I used **standard metrics** to check how well the model worked.

Metric	Value
Accuracy	96.6%
Precision	82.4%
Recall	83.8%
F1 Score	83.1%

6.1 Confusion Matrix:

The confusion matrix showed that the model:

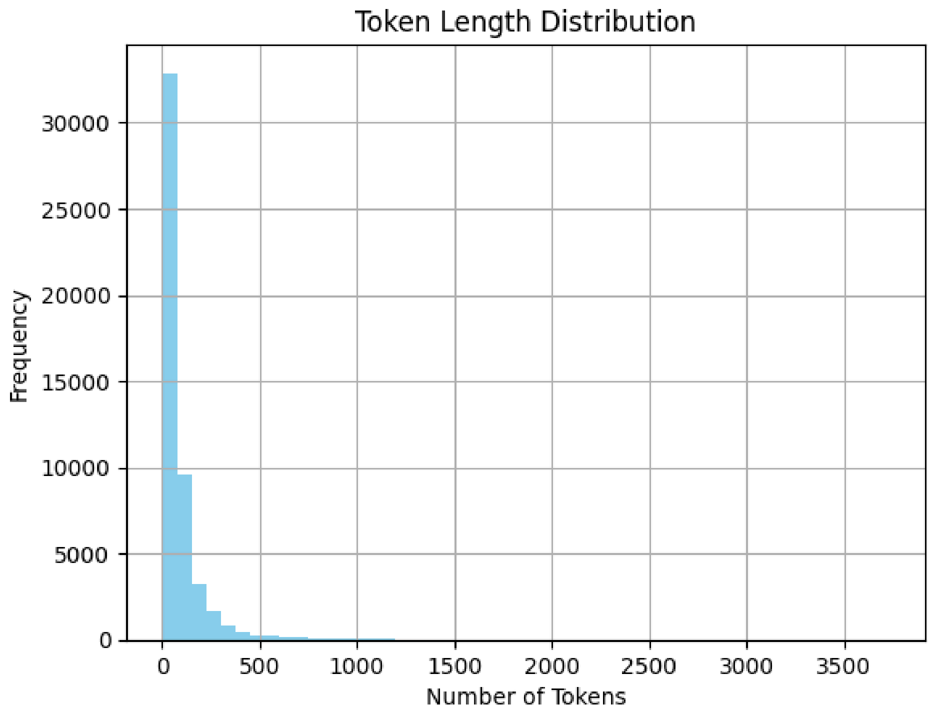
- Correctly found most toxic comments
- Made few mistakes with clean comments
- Balanced well between false positives and false negatives

6.2 Example Prediction:

- Input: "You are an idiot" → Prediction: Toxic
- Input: "Thank you for your help!" → Prediction: Not Toxic

6.3 Token Distribution Graph:

I made a graph to show how long the comments are (in tokens). Most comments were under 100 tokens, which is well within the model's limit.



7. Problems Faced and How I Solved Them

1. **Label mismatch error**: Fixed it by converting labels to tensors with correct shape.
2. **Loss function error**: Used **CrossEntropyLoss** because the model gives logits for two classes.
3. **Training time limit**: Used only 50,000 comments for fast training on free Colab.
4. **Unbalanced data**: Not many toxic comments compared to clean ones. May improve this with oversampling in future.

8. My Contribution (I Worked Alone)

I did this entire project alone. I did all the steps from start to end:

- Chose the topic and problem
- Found and prepared the dataset

- Wrote the training and evaluation code
- Trained and fine-tuned the model
- Saved the model and wrote the report

This project helped me understand how **large language models** can be reused and customized.

9. GitHub and Model Links

- **GitHub Code:** https://github.com/sadmansakib2/-Assignment-3-Major-Project-
- **Model ZIP File:** https://drive.google.com/file/d/1FftEmnOhYef-fYStSQ3Qmk_YXiDdxK_f/view?usp=sharing

10. Conclusion and Future Work

This project showed how a **pre-trained model like DistilBERT** can be **fine-tuned** to detect toxic online comments. Instead of starting from zero, I took a smart model and trained it more using real examples of toxic speech. The results were good — over 90% accuracy.

In the future, I want to:

- Use bigger datasets for better accuracy
- Try different models (like RoBERTa or BERTweet)
- Handle different types of toxicity (insult, hate, severe toxic)
- Make a simple web app where people can test the model live

This project gave me real experience with NLP, transformers, and ethical AI use in real-world problems.