WEEK 11 REPORT

Group Name:

	Member 1	Member 2
Name	Keilor Fallas Prado	Ky Dang
Email	kfallasprado@gmail.com	Keith.dang1610@gmail.com
Country	Costa Rica	Vietnam
College/ Company		
Specialization	NLP	NLP

Problem description:

In the previous week, I decided to choose the **pre-trained DistilBERT model** for the hate speech detection task. DistilBERT is a lighter, more efficient version of BERT, which significantly reduces resource consumption while maintaining a high level of performance (few hiden layers than BERT's).

One key consideration in applying this model for tweet classification is the presence of **emojis**, which are commonly used in social media posts. We used to think of using the other **emojiBERT** for accuracte prediction but considering about resource consumption. Emojis carry important contextual and emotional information that could be critical for detecting hate speech. However, DistilBERT does not inherently understand emojis, as it has been trained on textual data without specific emoji knowledge.

To address this, I plan to utilize the **"emoji" library**, which I mentioned in week 9. This library can convert emojis into corresponding text, making it easier for DistilBERT to interpret them. For example, the emoji "😊" would be translated to the text "smiling face," allowing the model to process the sentiment and context conveyed by the emoji.

Moving forward, I will explore two potential strategies for tokenizing and feature extraction:

1. **Manually Tokenizing Tweets with DistilBERT**:
   The first approach is to manually tokenize the tweets before passing them to the model. This involves converting the raw tweet text into a format using DistilBERT. The tweet will first go through a preprocessing pipeline where emojis are translated to text using the "emoji" library. After this, the text will be tokenized using the tokenizer that corresponds to the DistilBERT model. This ensures that the text is processed in the same way as the model was trained.

2. **Using the DistilBERT Tokenizer with Preprocessing**:
   The second approach is to leverage the **DistilBERT tokenizer** directly and include

an additional preprocessing step that handles emojis separately. Here, the tweet text can be processed to replace emojis with their textual descriptions using the "emoji" library, and then the entire tweet will be tokenized into subword units. The DistilBERT tokenizer is capable of breaking down words and handling out-of-vocabulary tokens, which will help improve the representation of the tweet, especially when emojis are involved.

I will experiment with both approaches to see which one yields the best results for hate speech detection. The goal is to ensure that the model retains as much contextual information as possible, including the emotional tone conveyed by emojis, while leveraging the efficiency of the DistilBERT model. Once I have preprocessed the tweets and tokenized them properly, I will train the model and evaluate its performance using the **Test** dataset.

Additionally, I plan to assess the impact of emoji tokenization on model performance. Specifically, I will analyze whether converting emojis to text improves the accuracy of hate speech detection compared to a version of the model that does not handle emojis.

Project lifecycle

Weeks	**Due date**	**Plan**
Week 8	11/26/2024	Review data source and ensure it is representative of hate speech contexts.
Week 9	12/02/2024	Remove duplicates, nulls, and irrelevant data.
Week 10	12/09/2024	Evaluate and select models such as Logistic Regression, SVM, or Transformers (e.g., BERT).
Week 11	12/16/2024	Tokenization - Identify relevant linguistic and contextual features
Week 12	12/23/2024	Training and evaluation model
Week 13	12/30/2024	Document the challenge

Github Repo link:

- Individual GitHub links:
  - KyDang: https://github.com/KeithDang1610/NLP_HateSpeech-Detection
  - Keilor: