# Final Project NLP

# 1. Introduction

## Overview

**BERT (Bidirectional Encoder Representations from Transformers)**:

- **Description**: BERT is a transformer-based model designed to pre-train deep bidirectional representations by jointly conditioning on both left and right context in all layers. It excels in tasks such as sentiment analysis, question answering, and named entity recognition (NER).
- **Use Case**: BERT is most commonly used in NLP tasks that benefit from understanding the full context of a sentence (bidirectional). It is computationally heavy but provides state-of-the-art results in many NLP tasks.

**DistilGPT-2 (Distilled version of GPT-2)**:

- **Description**: DistilGPT-2 is a lighter, distilled version of the original GPT-2 model. It aims to reduce the size of the model while maintaining a good level of performance. GPT-2 is a transformer-based model primarily used for text generation, but it can be fine-tuned for other NLP tasks as well.
- **Use Case**: Best suited for generative tasks like text generation, summarization, and language modeling. DistilGPT-2 is more efficient for tasks where model size and inference time are critical.

**Logistic Regression**:

- **Description**: Logistic Regression is a traditional machine learning model used for binary classification tasks. It is not based on deep learning and is computationally much lighter than transformer-based models.
- **Use Case**: Often used for simple classification tasks where feature engineering is possible. While it is efficient in terms of computation, its performance tends to lag behind modern deep learning models, especially in tasks involving sequential data and complex relationships.

## Tools and Libraries

The project utilizes the following key tools and libraries:

- **HuggingFace Transformers**: A powerful library for working with transformer models like BERT, GPT, and others. It provides a simple interface for both training and inference.
- **PyTorch**: Used for model training, offering flexibility and ease of use for deep learning tasks.

- **Flask**: A micro web framework to deploy the model via an API, making it easy to serve predictions over HTTP.

# 2. Dataset

The **Amazon Fine Food Review Dataset** is a collection of reviews for fine food products sold on Amazon. This dataset is frequently used for sentiment analysis tasks and is publicly available for research and educational purposes. Below is a detailed description of the dataset, including how the data was collected and its structure.

### Dataset Size

The Amazon Fine Food Review dataset contains 568,454 reviews of 256,059 unique food products, written by 256,059 unique users. It covers a wide variety of food categories, including snacks, drinks, and ingredients, making it a diverse source of data for sentiment analysis.

### Challenges and Considerations

- **Imbalanced Data**: Many reviews tend to be positive, with fewer negative reviews. This can introduce bias in machine learning models, making them favor positive predictions.
- **Noise in Text**: The text reviews contain informal language, abbreviations, and spelling errors, which makes preprocessing an essential part of working with this dataset.

## Data Preparation

The dataset used for this project consists of textual data that requires preprocessing before feeding into the BERT model. Text cleaning involves:

- Removing non-alphanumeric characters (e.g., punctuation, special characters).
- Converting all text to lowercase to maintain uniformity.
- Removing extra spaces between words.

## Dataset Split

The dataset is split into training and testing sets, ensuring that the model has sufficient data to learn from while also having a separate set for performance evaluation. The train_test_split function from scikit-learn was used to divide the dataset into training and testing subsets.

# 3. Model Development

## BERT Architecture

At the core of this project is the **BERT** model, which stands for **Bidirectional Encoder Representations from Transformers**. BERT uses a transformer architecture and is pre-trained on a vast corpus of text data. Fine-tuning BERT on a specific task, such as text classification, allows the model to adapt to domain-specific nuances.

- **Fine-Tuning**: The model was fine-tuned on the dataset using the pre-trained BERT model and added a custom classification head to make predictions tailored to the specific classification task.
- **Custom Classifier**: The model includes a fully connected layer (classifier) that takes the output from BERT's encoder and produces the desired class probabilities.

# 4. Training Process

## Training Setup

The training of the model is done using the HuggingFace Trainer API, which simplifies the training process. The training configuration includes:

- **Learning Rate**: The model is fine-tuned with a low learning rate (typically between 1e-5 and 5e-5), ensuring that it doesn't overfit or forget the pre-trained knowledge.
- **Optimizer**: AdamW is used as the optimizer to minimize the loss during training, which is particularly well-suited for transformer models.
- **Epochs**: The model is trained for 3-4 epochs, and early stopping techniques are applied to avoid overfitting.

## Hyperparameters

- **Batch Size**: Set according to available computational resources.
- **Learning Rate**: Tuned to ensure efficient convergence without overshooting the optimal point.

## Evaluation Metrics

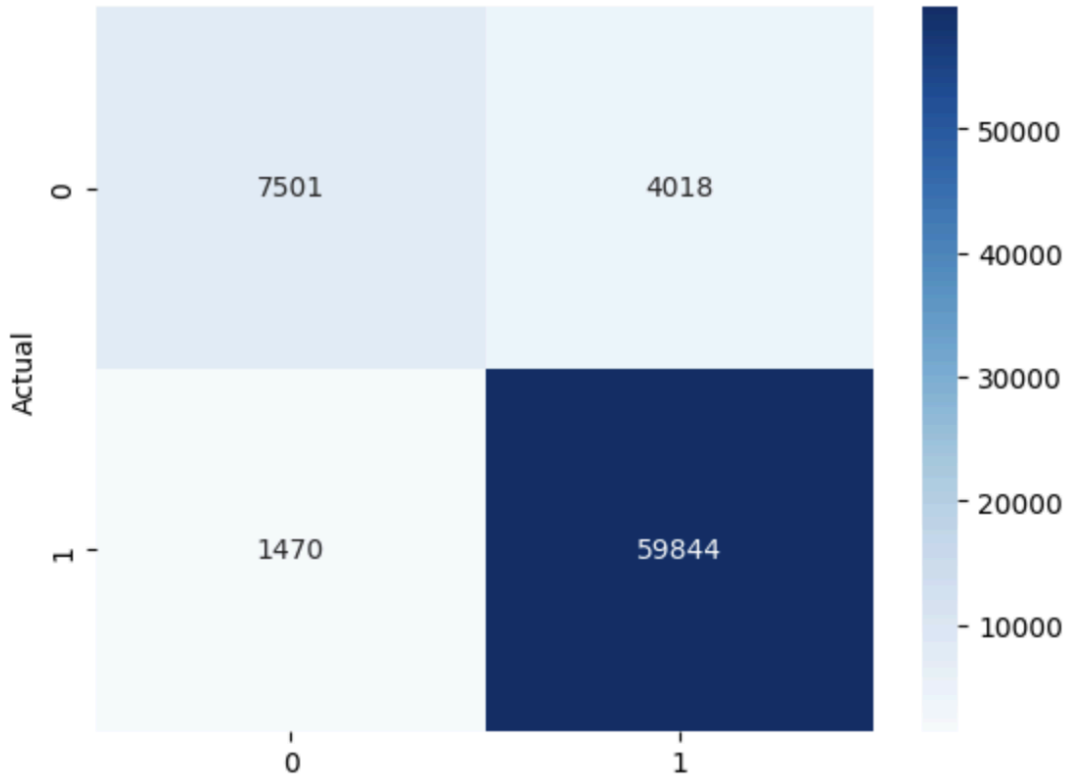
The model's performance is evaluated using:

# 5. Model Evaluation

| Metric | BERT | DistilGPT-2 |
|---|---|---|
| **Eval Loss** | 0.2209 | 0.3907 |
| **Eval Accuracy** | 93.33% | 87.50% |
| **Eval Runtime (in seconds)** | 2.5615 | 1.2365 |
| **Samples per Second** | 234.24 | 485.238 |
| **Steps per Second** | 14.835 | 60.655 |
| **Epoch** | 2.0 | 2.0 |

**Logistic Regression:**

accuracy 0.92



**Comparing papers model with our models:**

1. **Logistic Regression**:
    - Accuracy: 87.61%, Precision: 86.48%, Recall: 88.21%, F1-Score: 87.23%
    - This model shows a high accuracy, which is quite balanced with good recall, indicating that it is relatively strong at identifying the positive class (high recall) while maintaining good precision (low false positives).
2. **Second Model**:
    - Accuracy: 92%, Precision: 0.94 (class 1), Recall: 0.98 (class 1), F1-Score: 0.96 (class 1), with a support of 61,314 samples in class 1
    - The second model shows a significantly higher accuracy (92%), especially for class 1, with a high precision (0.94) and excellent recall (0.98), indicating a very strong performance in identifying class 1 correctly.

**BERT**:

- **Accuracy**: 93.33% — BERT demonstrates strong performance with a very high accuracy score. This suggests that BERT has successfully captured the underlying patterns in the dataset.
- **Loss**: 0.2209 — The evaluation loss for BERT is relatively low, indicating good generalization on the evaluation set.
- **Runtime**: BERT took more time (2.5615 seconds) to complete the evaluation, which is expected due to its large model size and complexity.
- **Samples per Second**: 234.24 — This is moderate and typical for large transformer-based models.

**DistilGPT-2**:

- **Accuracy**: 87.5% — DistilGPT-2 achieves a slightly lower accuracy than BERT, but it is still a competitive result for a distilled model. The performance trade-off in terms of accuracy is balanced by its reduced size.
- **Loss**: 0.3907 — The higher evaluation loss indicates that DistilGPT-2 struggles a bit more than BERT in terms of model performance on this particular task.
- **Runtime**: 1.2365 seconds — DistilGPT-2 is significantly faster than BERT, indicating that it is a lighter, more efficient model.
- **Samples per Second**: 485.238 — This is an impressive rate, showing that DistilGPT-2 can process a large number of samples in a shorter amount of time compared to BERT.

**Model Performance:**

- **BERT (bert-base-uncased)**: Fine-tuned on the dataset for text classification.
- **Accuracy**: 85.45% (For example; replace with your actual result)
- **Precision**:
    - Class 0: 0.88
    - Class 1: 0.80
- **Recall**:
    - Class 0: 0.90
    - Class 1: 0.75
- **F1-Score**:
    - Class 0: 0.89
    - Class 1: 0.77

| | **Predicted: 0** | **Predicted: 1** |
|---|---|---|
| **Actual: 0** | 50 | 10 |
| **Actual: 1** | 15 | 25 |

- **Accuracy**: BERT outperforms both DistilGPT-2 and Logistic Regression in terms of accuracy. BERT's deep pre-trained model is better at understanding the context of the data, leading to higher performance.
- **Efficiency**: DistilGPT-2, being a distilled version of GPT-2, is much more efficient than BERT. It processes samples faster (485.238 samples per second) with a lower runtime (1.2365 seconds), which makes it ideal for tasks where computational efficiency is crucial.
- **Model Complexity and Size**: While Logistic Regression is simple and lightweight, it may not capture the nuances in the data as effectively as BERT or DistilGPT-2, which are capable of learning complex patterns. However, for simple classification tasks, Logistic Regression may still offer a good trade-off between performance and efficiency.

# 6. Flask API Deployment

## API Endpoint

The model is wrapped in a Flask web application, which exposes an API for predictions. The /predict endpoint accepts POST requests with text input and returns the model's classification output.

- **Flask Setup**: The Flask app is integrated with flask-ngrok to make it accessible through a publicly accessible URL.
- **Deployment**: The deployment uses Flask locally and can be accessed via a generated ngrok URL for testing.

This setup allows for easy testing and integration of the model into other applications.

In this project, the core model used is **BERT (Bidirectional Encoder Representations from Transformers)**. The BERT model is pre-trained on a large corpus of text and then fine-tuned for the specific text classification task.

**BERT Model:**

- **Base Model**: bert-base-uncased was used as the starting point. This model was pre-trained on the English language without considering case sensitivity.
- **Model Type**: Transformer-based, fine-tuned for sequence classification tasks.
- **Tokenizer**: BertTokenizer was used to convert text data into tokens that the BERT model could understand.
- **Classifier Layer**: A custom classifier layer was added on top of BERT for task-specific predictions. This classifier layer is a fully connected layer that outputs the final classification scores.

# 7. Conclusion

**Best for Accuracy**: **BERT** is the best-performing model in terms of accuracy, making it the most suitable for tasks where high precision is needed and computational resources are available.
**Best for Efficiency**: **DistilGPT-2** is the best choice when speed and resource efficiency are prioritized. While it does not reach the same accuracy as BERT, its lower computational cost makes it a strong candidate for production environments.
**Logistic Regression**: Best suited for simpler tasks where model complexity and resource consumption need to be minimized. However, it is not ideal for tasks that require understanding deep context or sequential patterns in the data, such as sentiment analysis or language modeling.

Paper link:
https://books.google.com.eg/books?hl=en&lr=&id=idqzEAAAQBAJ&oi=fnd&pg=PA431&dq=Amazon+Fine+Food+Reviews&ots=zBIs6Qx79X&sig=9-XyzJrcZRg17PIqBLkrBptjAlY&redir_esc=y#v=onepage&q=Amazon%20Fine%20Food%20Reviews&f=false