**Ml PROJECT Report**

Topic: LLM

Name: Pranshul Thakur

Registration No: 12219336

Roll No: 3, K22UR

**1. Problem Understanding & Definition**

**1.1 Clarity of Problem Statement (4 Marks)**

This project focuses on building a **large language model (LLM)** capable of **text generation, sequence modelling, and contextual understanding**. The goal is to train a transformer-based model that can **generate coherent text, learn from structured data, and adapt to various NLP tasks**. The dataset consists of **literary text** from data.txt, which provides structured prose and poetry, making it suitable for training an LLM.

**1.2 Justification for Solving the Problem (3 Marks)**

LLMs are revolutionizing **AI-driven applications**, including **chatbots, code generation, automated summarization, and AI-assisted writing**. Training an LLM on structured literary data enhances its **ability to understand natural language, generate human-like responses, and improve contextual learning**. **By preprocessing and tokenizing the text effectively, the model can be optimized for high-quality language generation.**

**1.3 Defined Objectives & Hypotheses (3 Marks)**

**Objectives:**

* Build and train a **transformer-based LLM** capable of generating meaningful text.
* Preprocess and tokenize the dataset to improve training efficiency.
* Implement **batch processing and optimized data handling** to scale the model.
* Evaluate the LLM's **performance in text generation and contextual accuracy**.

**Hypothesis:**

* Proper text preprocessing (e.g., stopword removal, lowercasing) improves model accuracy.
* Training on structured text allows better generalization in **text generation and reasoning tasks**.
* Larger context windows lead to **more coherent and context-aware responses**.

**2. Dataset Selection & Preprocessing**

**2.1 Dataset Relevance and Quality (3 Marks)**

**2.1.1 Dataset Selection**

The dataset (data.txt) consists of **literary text**, making it a good candidate for **training an LLM**. It is designed to help the model learn **contextual relationships, text structure, and coherence** in language generation.

**Dataset Overview:**

* **Source:** Provided text file (data.txt)
* **Format:** Plain text
* **Feature Type:** Unstructured text
* **Length:** Multiple paragraphs of structured literary text

**2.2 Handling Missing Values, Outliers, and Data Normalization (3 Marks)**

**2.2.1 Handling Missing Values**

The dataset was **checked for missing values** using strip() and empty line checks. The preprocessing step:

* **Removes blank lines** before encoding the text.
* **Ignores encoding errors** by using errors='ignore'.

**2.2.2 Handling Outliers**

Outliers (unwanted noise in the text) were handled by:

* **Removing stopwords** to improve model focus on essential words.
* **Normalizing text** (e.g., converting to lowercase) for consistency.

**2.2.3 Data Normalization & Standardization**

* **Lowercased all text** to maintain uniformity.
* **Stopword removal** applied using the nltk.stopwords library.
* **Tokenization applied** using AutoTokenizer from bert-base-uncased.

**2.3 Feature Selection & Engineering (4 Marks)**

**A. Feature Selection**

* The dataset was tokenized using **BERT tokenizer (AutoTokenizer)**.
* Stopwords were removed using **NLTK stopwords** to reduce unnecessary words.
* **Non-alphabetic characters were kept** to retain meaning.

**B. Feature Engineering**

* **Encoding function created** (encode = lambda s: tokenizer.encode(s, add\_special\_tokens=True)) to convert text into token IDs.
* **Custom preprocessing function added (preprocess\_text)** for stopword removal and text normalization.
* **Sequence chunking applied** to structure input data for transformer models.

**C. Code Snippets**







