Submission 1

Name	Dhanya Prabhu
SRN	PES2UG23CS169
Section	C

Output Table:

Task	Model	Classification (Success/ Failure)	Observation (What actually happened?)	Why did this happen? (Architectural Reason)
Generation	BERT	Failure	Generated incoherent and repetitive text	BERT is an encoder-only model and is not designed for autoregressive text generation
	RoBERTa	Failure	Output was meaningless or generation failed	RoBERTa is also an encoder-only model without a decoder
	BART	Partial Success	Generated a short but somewhat meaningful continuation	BART has an encoder-decoder architecture designed for sequence generation
Fill-Mask	BERT	Success	Correctly predicted words like "generate" or "create"	BERT is trained using Masked Language Modeling
	RoBERTa	Success	Accurate and relevant mask predictions	RoBERTa is optimized for Masked Language Modeling
	BART	Partial Success	Mask prediction worked but was less accurate	BART is trained mainly for denoising seq2seq tasks
QA	BERT	Partial Success	Returned incomplete or low-confidence answers	Base BERT is not fine-tuned for question answering
	RoBERTa	Partial Success	Slightly better span extraction but inconsistent	Strong encoder but not QA fine-tuned
	BART	Partial Success	Answered but sometimes incorrect or vague	Base BART is not specialized for extractive QA

Conclusion:

This experiment highlights the importance of model architecture. Encoder-only models like BERT and RoBERTa perform well on understanding tasks such as masked word prediction but fail at text generation. Encoder-decoder models like BART are better suited for generation tasks.

What I Understood:

Through this assignment, I understood that the architecture of a language model strongly influences the tasks it can perform well. Encoder-only models such as BERT and RoBERTa are designed primarily for understanding tasks like masked language modeling and contextual representation learning. In contrast, encoder-decoder models such as BART are better suited for sequence-to-sequence tasks like text generation.

I also learned that forcing a model to perform a task it was not designed for leads to poor or meaningless results, which helps highlight the importance of choosing the right model architecture for a given problem.

What I Built:

I created a Jupyter Notebook to benchmark three transformer-based models: BERT, RoBERTa, and BART. Using the HuggingFace `pipeline` API, I tested all three models on three different tasks:

1. Text Generation

2. Masked Language Modeling (Fill-Mask)

3. Question Answering

For each task, I observed the output produced by each model and classified the result as a success or failure based on the quality of the output. I then documented my observations and explained the results using architectural differences between encoder-only and encoder-decoder models.

This experiment helped me practically understand why certain models perform well on specific NLP tasks and reinforced the theoretical concepts covered in Unit 1.