Milestone 1 – Code Explainer Project

Objective:

The goal of Milestone 1 is to design a pipeline that parses code snippets, extracts structural features, tokenizes code, and forwards the processed data to pretrained NLP models (MiniLM, DistilRoBERTa, MPNet). The outputs and embeddings are compared to analyze representational differences.

Methodology:

1. Prepared 10+ code snippets in Python covering functions, classes, and imports.
2. Parsed code using *Abstract Syntax Tree (AST)* to extract functions, classes, imports, and code patterns.
3. Tokenized code with Hugging Face tokenizer for each pretrained model.
4. Forwarded code tokens to *MiniLM*, *DistilRoBERTa*, and *MPNet* to generate embeddings.
5. Compared embeddings using cosine similarity and visualized differences via dimensionality reduction (t-SNE/PCA).

Results:

- MiniLM produced compact embeddings with strong contextual capture.
- DistilRoBERTa showed higher sensitivity to variable naming.
- MPNet balanced semantic and syntactic understanding effectively.
- Visualizations highlighted clustering of similar code patterns.

Observations:

- AST parsing ensures structural clarity in code before model input.
- Pretrained NLP models differ in their treatment of syntax vs. semantics.
- Embedding comparisons provide insights for future fine-tuning on code-specific datasets.
- MPNet embeddings demonstrated the most stable clustering across snippets.

Conclusion:

The Code Explainer pipeline successfully integrates AST parsing, tokenization, and embedding generation using multiple pretrained NLP models. Comparative analysis shows that each model offers unique strengths, with MPNet proving most consistent for code representation tasks. This milestone lays the foundation for building robust AI-based code explanation and generation systems.