# CodeGenie – Model Benchmarking & Analysis Notebook

Milestone 2 Project | Infosys Springboard AI Internship

## Overview

CodeGenie is an interactive Jupyter/Colab-based system designed to benchmark and analyze multiple AI code generation models on programming prompts. It provides intuitive visual UIs using ipywidgets and powerful metric-based evaluations for code quality, maintainability, and generation performance. This project demonstrates the ability to run, compare, and visualize performance across various language models under consistent test conditions.

## Key Features

• Benchmark all models at once or selectively inspect specific ones.

• Calculates advanced code metrics such as Cyclomatic complexity, Maintainability index, and Generation time per model.

• Clean tabular results and comparative plots.

• Exportable logs for session analysis.

• Works seamlessly in Google Colab and Jupyter Notebook.

## Technologies Used

|  |  |
| --- | --- |
| Component | Description |
| Python 3.12+ | Core language |
| Transformers / Tokenizers | For AI code generation |
| Pandas & NumPy | Data handling & metric storage |
| Matplotlib & Seaborn | Visualization of model performance |
| Ipywidgets | Interactive UI for prompt-based benchmarking |
| JSON | Metadata cleaning and notebook structure |

## Notebook Structure

### Section 1–4: Model Loading and Utility Setup

Initializes and loads multiple pre-trained AI models for testing. Defines helper functions like generate\_code(model, tokenizer, prompt) and calculate\_advanced\_metrics(code).

### Section 5: UI #1 – Benchmark All Models

Allows users to run a single prompt across all models simultaneously and displays generated code per model, time taken, and code metrics.

### Section 6: UI #2 – Inspect Selected Models

Introduces checkbox-based control for selecting specific models to run the benchmark.

### Section 7: Final Analysis and Visualization Report

Summarizes all benchmark results with comprehensive data tables and comparative bar plots.

### Section 8: Cleaning Widget Metadata

Ensures clean metadata for GitHub rendering by removing 'metadata.widgets.state'.

## How to Use

1️⃣ Run in Google Colab

Upload the notebook to Google Colab, execute each cell sequentially, use UIs to run benchmarks, and generate the report in Section 7.

2️⃣ Clean and Upload to GitHub

Run the metadata-cleaning code before uploading to ensure proper rendering on GitHub. Upload the cleaned notebook named 'cleaned\_CodeGenie\_Milestone2.ipynb'.

## Example Output Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Prompt | Complexity | Maintainability | Gen Time (s) |
| GPT-2 | Fibonacci Series | 5.0 | 78.2 | 1.34 |
| CodeGen | Factorial Program | 3.0 | 85.7 | 0.98 |

## Future Enhancements

• Integrate LLM evaluation metrics like BLEU or CodeBLEU.

• Add syntax error detection with AST parsing.

• Allow CSV export of all results automatically.

• Add multi-threaded prompt execution for faster comparisons.

## Author

Piyushmani Tiwari  
B.Tech (Information Technology) — Government Engineering College, Bilaspur  
Email: piyushmanitiwari500@gmail.com  
GitHub: https://github.com/piyushmanitv  
LinkedIn: https://linkedin.com/in/piyushmani-tiwari-2349101a9

## License

This project is released under the MIT License, allowing open collaboration and modification for educational use.