

Image created with ChatGPT

## Using Machine Learning to Predict the Next Command in Revit \_ Part 7 - Outro



Let's BIM Together 728 followers





September 25, 2024

(Part 7 of a 7 series blog. Click here to go to Part 1)

I must admit, I was excited to wrap up this learning journey because I underestimated how long it would take to get here. As I mentioned earlier, even with the compromises made in the prediction logic (e.g., not using the most recent command for prediction), there are other areas that could be explored further too. This entire project could've been done using just *Visual Studio* with C# and *Power BI* - without *PyRevit*, *PowerShell*, or *Notepad++*. But since this wasn't a linear process, I used whatever tools were most convenient at the time.

One significant limitation was the small dataset. Machine learning models, like the *LightGBM* algorithm used in *ML.NET*, typically perform better with larger datasets. The small size of my data likely impacted the model's accuracy, and with more data, the results would likely differ. It is possible, with a few changed to *Script 1: 1\_SCRAPE-COMMANDS.ps1* from Part 2 placed in the Revit's journal path location, to create your own data set with your own personal data. On one of my machines, I found around 50,000 logged commands! That being said, for now, I'm setting this project aside to focus on a few others that I'm excited to dive into. However, I plan to revisit this with a different approach in the future.

The predictions in this project were made using ML.NET's LightGBM algorithm which is a gradient-boosting machine-learning technique. LightGBM is designed for structured, tabular data, where specific features (inputs) are defined and the algorithm learns from those to make predictions. This is considered machine learning.

Deep learning, by contrast, involves neural networks with multiple layers and is used for more complex tasks like image recognition or unstructured data processing. It's not limited to predefined features and can automatically learn representations from data. While I used LightGBM for Revit command predictions, I am curious how different this approach would be using a deep learning framework such as PyTorch.

Then there are Large Language Models (LLMs), such as Meta's LLaMA. These models are a subset of deep learning but are specifically optimized for natural language tasks like text generation or language understanding. Unlike LightGBM or even standard deep learning models. LLMs are designed to process vast amounts of text data, focusing on language-related problems rather than structured command prediction like in this project.

Lastly, I want to mention a tool that helped organize my thoughts while breaking this project into digestible parts: Canva flowcharts! Canva is intuitive, free-to-use, and ideal for organizing workflows visually. While there is a premium version, the free one offered everything I needed for flowchart creation, and I even use it as a lightweight alternative to Illustrator for simple visuals. You can access it here: www.canva.com



If you've made it this far in reading, first of all, thank you. Hopefully you found something useful in this series or have suggestions to share about how I could improve this. I have worked on this project sporadically on and off for quite some time. As I am finalizing this expedition, I already have a few more projects almost done that I am eager to present in a similar format, though they will be much shorter.

Go to the previous part

Part 1 - Introduction & Index

Part 2 - Data Preprocessing Using PowerShell & Python with PyRevit

Part 3 - Data Analysis Using Power BI

Part 4 - Model Trainer Application Using C# ML.NET PowerShell Notepad++

Download AG Feeling Lucky Plugin for Revit & Trainer/Analysis Console Application here:

Part 5 - Parameter Optimization Algorithms Using Python with Optuna

Part 6 - Revit Plugin Using C# ML.NET & Revit API

Part 7 - Outro (Canva flow chart)

https://letsbimtogether.com/blog.html