Lecture 2 - AI Programming

Management Science

Dr. Tobias Vlćek

Programming with AI

Using AI to generate code

- Coding by hand is not the only way to generate code
- Most likely, a lot of you have already used ChatGPT

. . .

How do

Large Language

Models work?

Photo by Taylor Vick on Unsplash

Large Language Models (LLMs)

- Think of them like advanced pattern recognition systems
- They have “read” massive amounts of text
- Books, websites, articles, code, and more
- Text is broken into tokens, parts of words or punctuation
- Based on patterns, they can generate new text

Training LLMs

- Imagine learning a language by reading millions of books
- Learns patterns in how words and ideas connect via tokens
- Interconnected nodes with weights representing patterns
- During training, these weights are adjusted
- Once trained, applying them takes much less ressources

. . .

> Tip
>
> Using a trained model is called inference.

Pattern Recognition

- Not like a search engine!
- When asked, it looks for relevant patterns it learned
- Like having a huge library in its “memory” to draw from
- It can find patterns between concepts and your question
- Knows only limited text at once (context window)

. . .

> ⚠ Warning
>
> Managing context windows is crucial!

Probability based responses

- After each written token, it predicts “what should come next?”
- Like a advanced version of the word prediction on your phone
- Chooses the most likely next token based on training
- But can’t actually “think” or “understand” like humans

Limitations

- No true understanding of cause and effect[1]
- Sometimes makes mistakes or “hallucinates”
- Mostly only knows what it was trained on[2] and can reflect biases
- No emotional understanding (but can simulate responses!)[3]

Impact on Jobs

- Question: What do you think about their impact on jobs?
- Question: What are the implications for us?
- Question: Can we use them to our advantage?

. . .

> ⚠ Warning
>
> If you use free models, be aware that your prompts are going to be used by the providers and are not private. But for learning and experimenting, this should be no issue.

A Great Overview by 3Blue1Brown

- Greg Sanderson provides an excellent explanation of LLMs

---

[1] https://www.anthropic.com/research/tracing-thoughts-language-model
[2] This can partially be improved by using context from the internet.
[3] User can get attached to talking to models

- Great starting point to understand LLMs
- Check out his YouTube channel, 3Blue1Brown for more

https://www.youtube.com/embed/LPZh9BOjkQs

AI Coding Partner

What is GitHub Copilot?

GitHub Copilot is an AI pair programmer that helps you write code faster and with less effort.

. . .

Think of it as:

- An autocomplete for entire lines or blocks of code
- A coding assistant that understands context
- A learning tool that shows you coding patterns

. . .

> **Note**
>
> Copilot uses AI trained on billions of lines of public code to suggest completions.

. . .

> **Tip**
>
> There are alternative like Zed or Cursor, but you can use Copilot for free as student.

Used autocomplete before?

. . .

When you type on your phone, it suggests the next word.

. . .

GitHub Copilot does the same for code:

- You write a comment describing what you want
- Copilot suggests the code to do it
- You accept, modify, or reject the suggestion

. . .

It's autocomplete but much better then what you are used to

Why Use Copilot?

Benefits you while learning and working with Python:

- Faster coding: Less time typing boilerplate code
- Learn patterns: See how experienced programmers code
- Understand syntax: Get correct syntax without memorizing
- Stay in flow: Focus on logic, not syntax errors

. . .

> Tip
>
> Especially helpful when you know WHAT you want to do but forget HOW to do it.

When should you use Copilot?

. . .

Good uses of Copilot

- Understanding Python syntax you forgot
- Writing repetitive or boilerplate code
- Getting unstuck on simple problems
- Exploring different approaches

What shouldn't you do with Copilot?

. . .

Not so good uses of Copilot

- Replacing learning fundamentals
- Accepting code you don't understand
- Skipping practice exercises
- Copy-pasting without reading

. . .

Always understand what Copilot suggests before accepting!

. . .

But of course I know you will not do that ;)

Live Demo

Demo: Writing a Function

Scenario: Calculate shipping costs based on weight and distance.

. . .

Step 1: Write a text describing what you want

```
"""
Calculate shipping cost based on weight (kg) and distance (km)
Base rate: €5, plus €0.10 per kg, plus €0.05 per km
"""
```

...

Step 2: Copilot suggests the function

...

Watch how Copilot solves this in the IDE!

Demo: The Result (hopefully)

```python
def calculate_shipping_cost(weight, distance):
    """Calculate shipping cost based on weight and distance."""
    base_rate = 5.0
    weight_rate = 0.10
    distance_rate = 0.05
    return base_rate + (weight * weight_rate) + (distance * distance_rate)
```

...

What Copilot did:

- Created function with proper parameters
- Implemented the calculation logic
- Used clear variable names

...

> Tip
>
> Notice how the text guided Copilot to generate exactly what we needed!

Accept without reading?

...

Copilot might suggest code that:

- Works but uses concepts you haven't learned yet
- Contains subtle bugs or edge cases
- Doesn't match your specific requirements
- Uses inefficient approaches
- Introduce dangerous code in your project

...

> ⚠ Warning
>
> Dangerous code can lead to security vulnerabilities, data loss, or other issues. In the context of this lecture it should be no issue, but in companies it can be one!

My take: Just be careful, ok?

Your code, your responsibility:

1. Read the suggestion carefully
2. Understand what it does
3. Test it with examples
4. Modify if needed

. . .

> ⚠ Warning
>
> Don't accept code blindly, especially later if things are more complicated.

Getting Started with Copilot

Get Free Access

GitHub Student Developer Pack gives you free Copilot access!

. . .

1. Go to education.github.com/pack
2. Sign up with your university email
3. Verify your student status
4. Wait for approval (usually 1-2 days)
5. Login into your account in VS Code

. . .

> ℹ Note
>
> You'll need a GitHub account. Create one at github.com if you don't have one.

Verifying Copilot is Working

1. Create a new Python file (`.py`) and type something
2. Wait 1-2 seconds

. . .

If working, you'll see:

- Gray “ghost text” suggesting code
- Press `Tab` to accept
- Press `Esc` to reject

. . .

> Tip
>
> Try to get copilot running on your own until next session.

Best Practices

When to Use Copilot

Use Copilot for:

. . .

1. Syntax Help

```
# Convert string to datetime
# Copilot remembers: pd.to_datetime()
```

. . .

2. Boilerplate Code

```
# Create a function to read CSV file and return DataFrame
# Copilot writes the import and function structure
```

. . .

3. Getting Unstuck

```
# I know I need to filter this list, but forget the syntax
# Copilot suggests: [x for x in items if x > 10]
```

When Not to Use Copilot

Don’t use Copilot for

. . .

1. Learning Fundamentals

- You should understand loops, functions, and data types

. . .

2. Skipping Problem-Solving

- Figure out the logic first, then use Copilot for syntax

. . .

3. Accepting Without Understanding

- Ask: “Can I explain what this code does?”

Summary

Key Takeaways

1. GitHub Copilot is a tool: You still need to learn fundamentals
2. Understand before accepting: Read every suggestion carefully
3. Use it strategically: Syntax help yes, thinking replacement no
4. Get free access: GitHub Student Developer Pack
5. Practice:The more you use it, the more helpful it becomes

. . .

> **Note**
>
> Before next lecture, make sure you applied for GitHub Student Developer Pack to get access to copilot!

. . .

> **Tip**
>
> If you run into issues, we will solve them next week together!

Bibliography