Session 5
Introducing Cline

Fall 2025 ECE 157/272A
Seoyeon Kim, Heekyung Lee

Introducing Cline

Introducing Cline

- **Cline is an open source AI coding agent that can understand entire codebases (your working folder), plan and execute tasks**

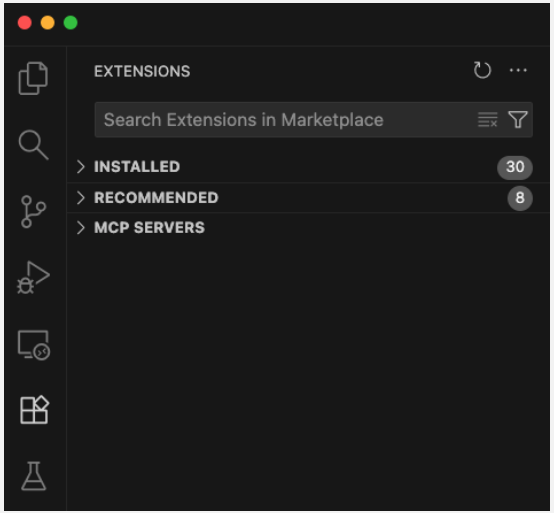
- **Cline website: https://cline.bot**

- **Cline docs: https://docs.cline.bot/introduction/welcome**

Install Cline

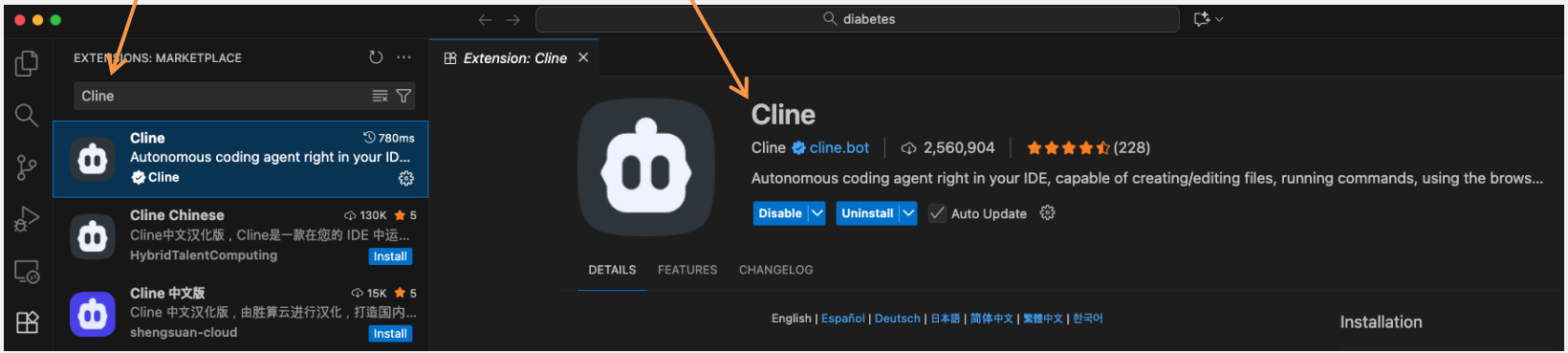
- **We will use Cline on VS Code**

1. **Create your account:** https://docs.cline.bot/getting-started/installing-cline
2. **Open VS Code and open the Extensions view**



Install Cline

3. Search Cline and install

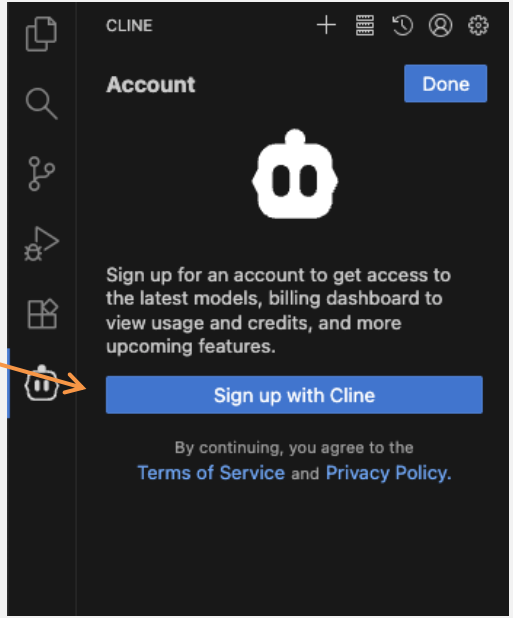


Install Cline

4. Click on Cline icon in the Activity Bar

5. Click the Sign-Up button

6. Log in with your account

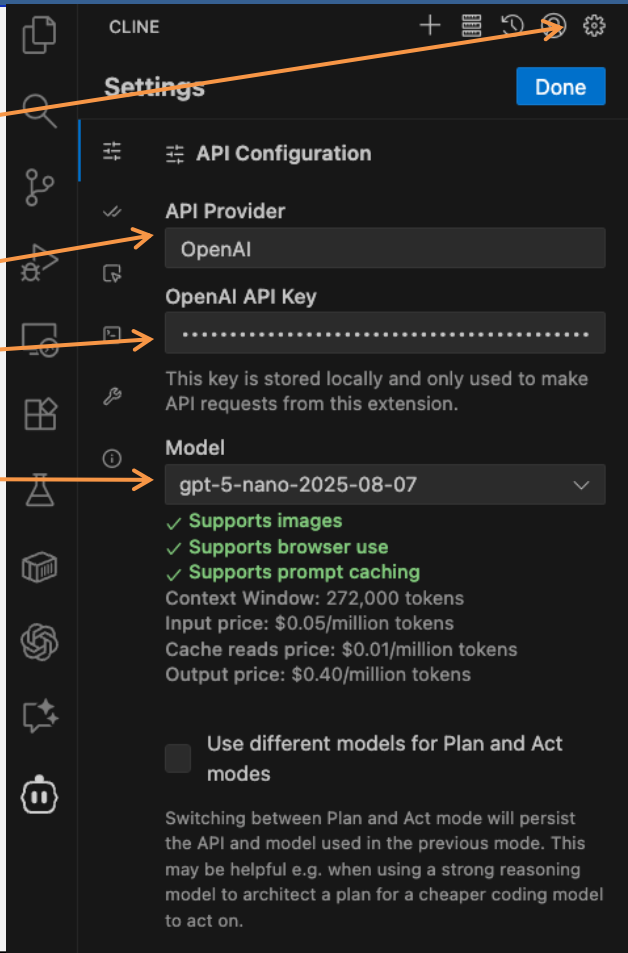


Install Cline

7. Open Cline Settings
8. Select “OpenAI” from API Provider
9. Enter your API key
10. Choose a model
    - We recommend gpt-5-nano
    - You can use other API Providers too,

e.g., Anthropic’s claude-sonnet, if you want

- Check out https://docs.cline.bot/getting-started/selecting-your-model



Introducing Cline

- **What is Cline?**

- **In easy terms, it is a ChatGPT that can explore the folder, read all the files in the folder, write, edit or debug codes, and even execute the command based on your prompt.**

Introducing Cline

- Let’s try Cline with a tutorial!
- https://docs.cline.bot/getting-started/your-first-project

Cline Features

- Plan & Act
- @ Mentions
- Cline Rules
- … and more

Plan & Act

Plan mode

- Reads your entire folder to understand the context
- Focuses on understanding prompts and creating a plan
- Helps identify potential issues before coding

Act mode

- Remembers the plan from the plan mode
- Makes changes to the code
- Executes the code

- https://docs.cline.bot/features/plan-and-act

@ Mentions

- **File/Folder Mentions**
  - Instead of describing the file and copy-and-pasting the chunk in the file/folder for the context, you can simply mention their names
- **Terminal Mentions**
  - You can directly bring your terminal output into the conversation
  - E.g. “ I’m getting this error: @terminal, what is causing this? ”
- **Problem Mentions**
  - You can give Cline access to all the errors and warnings in your workspace
  - E.g. “ Why am I having these TypeScript errors: @problems ”

- https://docs.cline.bot/features/at-mentions/overview

Cline Rules

- **System-level guidance**
- **A persistent way to include context and preferences**
- **We don’t necessarily need this for homework 5**

- **https://docs.cline.bot/features/cline-rules**

Example Cline Rule Structure

```
# Project Guidelines

## Documentation Requirements

- Update relevant documentation in /docs when modifying features
- Keep README.md in sync with new capabilities
- Maintain changelog entries in CHANGELOG.md

## Architecture Decision Records

Create ADRs in /docs/adr for:

- Major dependency changes
- Architectural pattern changes
- New integration patterns
- Database schema changes
  Follow template in /docs/adr/template.md

## Code Style & Patterns

- Generate API clients using OpenAPI Generator
- Use TypeScript axios template
- Place generated code in /src/generated
- Prefer composition over inheritance
- Use repository pattern for data access
- Follow error handling pattern in /src/utils/errors.ts

## Testing Standards

- Unit tests required for business logic
- Integration tests for API endpoints
- E2E tests for critical user flows
```

Homework 4

Overview

- **Goal**
  - to get a feel for using Cline
  - to explore how to use it efficiently and effectively

- **Using Cline effectively makes it more powerful than using LLMs**
- **We will perform some simple tasks using the Diabetes dataset, which is a binary classification dataset**

Diabetes Health Indicators Dataset

- **A CSV file with 70692 rows and 22 columns**

- **Binary classification dataset for predicting whether an individual is at risk of having diabetes or not based on features regarding their lifestyle and health measurements**

Diabetes Health Indicators Dataset

Column Name	Data Type	Description
Diabetes_binary (target value)	float	0 = no diabetes, 1 = prediabetes or diabetes
HighBP	float	0 = no high blood pressure, 1 = high blood pressure
HighChol	float	0 = no high cholesterol, 1 = high cholesterol
CholCheck	float	0 = no cholesterol check in 5 years, 1 = yes cholesterol check in 5 years
BMI	float	Body Mass Index
Smoker	float	Have you smoked at least 100 cigarettes in your entire life? 0 = no, 1 = yes
Stroke	float	Ever told you had a stroke. 0 = no, 1 = yes
HeartDiseaseorAttack	float	Coronary heart disease (CHD) or myocardial infarction (MI). 0 = no, 1 = yes
PhysActivity	float	Physical activity in past 30 days - not including job. 0 = no, 1 = yes
Fruits	float	Consume fruit one or more times per day. 0 = no, 1 = yes
Veggies	float	Consume vegetables one or more times per day. 0 = no, 1 = yes
HvyAlcoholConsump	float	Adult men drink >=14 drinks per week, or adult women drink >=7 drinks per week. 0 = no, 1 = yes
AnyHealthcare	float	Have any kind of health care coverage, including health insurance, prepaid plans such as HMO, etc. 0 = no, 1 = yes
NoDocbcCost	float	Was there a time in the past 12 months when you needed to see a doctor but could not because of cost? 0 = no, 1 = yes
GenHlth	float	Would you say that in general your health is: scale 1-5. 1 = excellent, 2 = very good, 3 = good, 4 = fair, 5 = poor
MentHlth	float	Days of poor mental health scale 1-30 days.
PhysHlth	float	Physical illness or injury days in past 30 days scale 1-30.
DiffWalk	float	Do you have serious difficulty walking or climbing stairs? 0 = no, 1 = yes
Sex	float	0 = female 1 = male
Age	float	13-level age category. 1 = 18-24, 2 = 25-29, 3 = 30-34, 4 = 35-39, 5 = 40-44, 6 = 45-49, 7 = 50-54, 8 = 55-59, 9 = 60-64, 10 = 65-69, 11 = 70-74, 12 = 75-76, 13 = 80 or older
Education	float	Education level scale 1-6. 1 = never attended school or only kindergarten, 2 = elementary (grades 1 - 8), 3 = some high school (grades 9 - 11), 4 = high school graduate (grade 12 or GED), 5 = some college or technical school (College 1 year to 3 years), 6 = college graduate (college 4 years or more)
Income	float	Income scale scale 1-8. 1 = less than $10,000, 2 = $10,000 to $15,000, 3 = $15,000 to $20,000, 4 = $20,000 to $25,000, 5 = $25,000 to $35,000, 6 = $35,000 to $50,000, 7 = $50,000 to $75,000, 8 = $75,000 or more

Task Overview

- Open the homework folder on VS Code
  - This will be your workspace
  - There should be “diabetes.csv” file in the folder
- Use Cline for the tasks
- Each task will create files (.py files and .png files)
  - Do not remove them unless you want to redo the task
  - You will compress and submit the folder including these files

Task 1

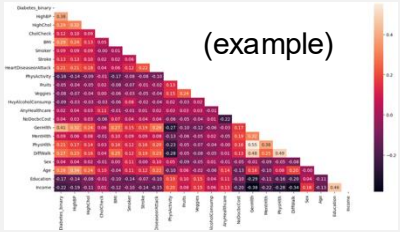
- Make ‘standardize.py’ that standardizes column *BMI* in ‘diabetes.csv’
- Make ‘stand_check.py’ that calculates and prints a mean and a standard deviation of column *BMI*
- Screenshot the printed result, and save it as ‘task1.png’

Task 2

- **Use Plan Mode of Cline to plan below task:**
  - Find Binary, Categorical, and Numeric types of columns
  - Plot binary columns' distributions as pie charts → save as 'binary.png'
  - Plot categorical columns' distributions as bar charts → save as 'category.png'
  - Plot numeric column's distribution as a histogram → save as 'numeric.png'
- **Screenshot the plan result, and save it as 'task2.png'**
- **Use Act Mode of Cline to run the plan**
  - Create 'distribution.py' that performs the plan

Task 3

- **Make ‘corr.py’ that plots the correlation matrix of all 22 columns as a heatmap**
  - **Save the plot as ‘heatmap.png’**



- **Mention (@) ‘corr.py’ to modify the code:**
  - **Calculate the correlation between each feature and the target variable *Diabetes_binary***
  - **Plot a bar chart showing the top 10 features with the highest absolute correlation values**
  - **Save the plot as ‘bar_chart.png’**
  - **Screenshot your prompt for modification and save as ‘task3.png’**

Submission (Due: 11/19 Wed 4PM)

- **Make sure every output file is in the folder**
  - diabetes.csv (BMI modified)
  - standardize.py
  - stand_check.py
  - task1.png
  - distribution.py
  - binary.png
  - category.png
  - numeric.png
  - task2.png
  - corr.py
  - heatmap.png
  - bar_chart.png
  - task3.png
- **Compress the folder**
- **Submit .zip file on Gradescope**

Questions?