

# Data Science Retreat Roadmap

**Presented by:** Salman Fariz Navas

[Connect on LinkedIn](#)

# Today's Agenda



## Welcome & Introductions

Getting to know each other and building our learning community



## Roadmap

Week-by-week breakdown of skills, projects, and milestones



## Software Setup

# Welcome to Data Science Retreat!



# About Me

Computational physicist transitioning into Machine Learning



**DSR participant: Batch 43**



**Key Interests:** Multimodal ML,  
Digital twins, Agentic workflows.



**Connect on LinkedIn**

# Let's Get to Know Each Other



Your Name



Professional Background

Previous experience, or field you're transitioning from.



Why Data Science?

What drew you to this field and this specific bootcamp program.



Program Expectations

What specific skills or knowledge do you hope to gain here



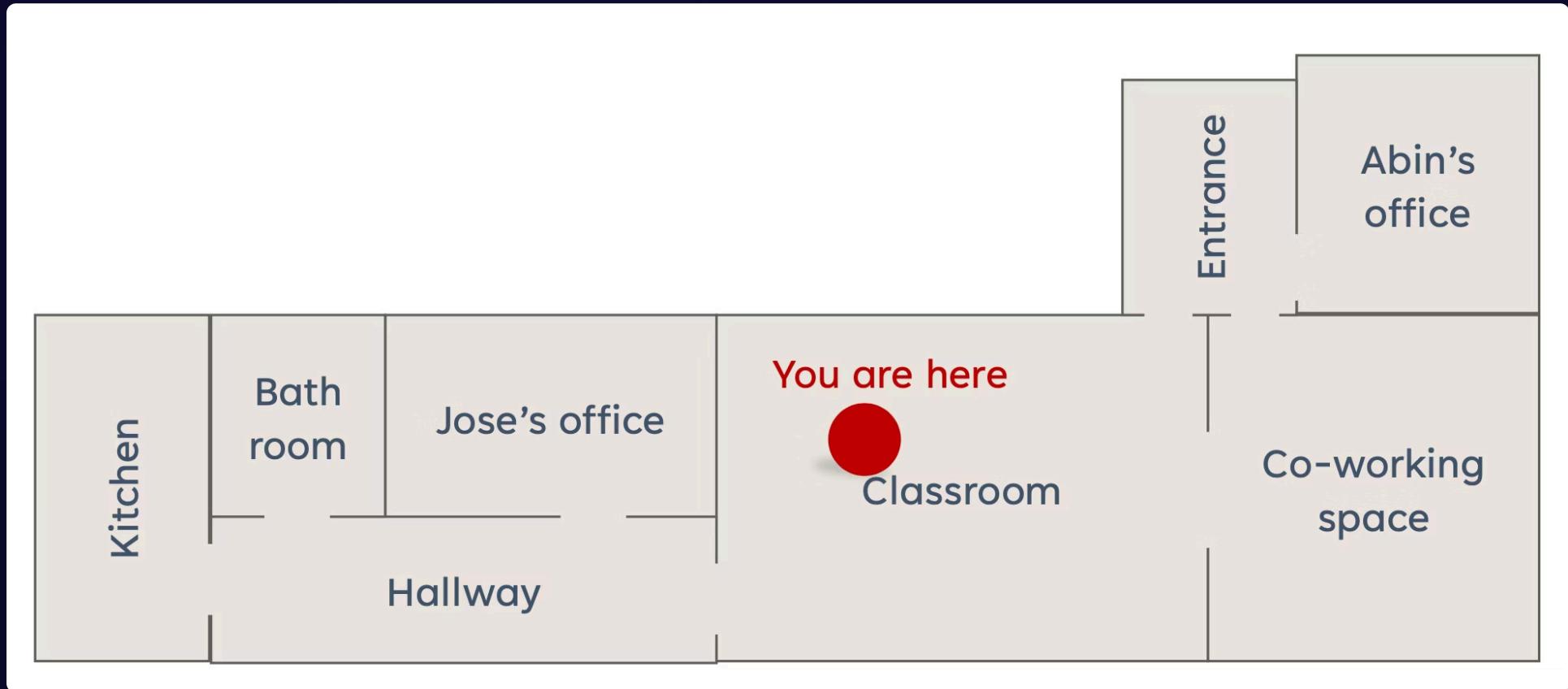
Long-term Goals

Take 2-3 minutes to introduce yourself.

# General Info



## The Venue



# The Teachers



Industry Data Science/ML Experts

Hands-on practical knowledge.

Updated tech stack.



Some are DSR graduates



They are people too :)

Working on their free time to teach at DSR in between  
their busy schedules.

# Organizational Info

# Organizational Info



## Communication



## General Class guidelines

Timing: 09:30 - 17:30  
Please be on time.  
Be prepared.



## Attendance

Participants registered for the in-person format are required to attend physically at the venue.



## Feedback

Abin & Arun are always available for any help.

# ROADMAP

# Programming Fundamentals



## Python, numpy, pandas

Introduction to python and the libraries dumpy and pandas for Data Science/ML.



## Bash/CLI

Introduction to the common line interface and bash commands and actions.



## Git

Introduction to git and related workflows.



## SQL

SQL for DS.

# Statistics Fundamentals

Deep-dive into the statistics and probability topics necessary for Data Science/Machine Learning.

# Data Science Fundamentals (Tabular Data)



## Data Scraping

Data acquisition, management.



## Data Cleaning



## Feature Engineering

Modifications and alterations to the data to aid in model development and performance.



## Exploratory Data Analysis

Visualizing and summarizing key information and statistics about your data



## Model Selection & Evaluation

Deciding on a model that is best suitable for your problem. Evaluating the performance of the model.

# Machine Learning Fundamentals (Tabular Data)



## Supervised Learning Algorithms

Linear regression, Logistic regression., Support Vector Machines.



## Unsupervised Learning Algorithms

K- Nearest Neighbors (KNN), K-Means Clustering, Principal Component Analysis (PCA)



## Decision Trees

Decision tree based classification and regression algorithms. Boosting, Bagging etc.

# Time Series Analysis



## Component Analysis

Trends, Noise, Seasonality.

Goal: *Understand what the data looks like, and what underlying structures exist.*



## Time Series Modeling

Autoregressive (AR) processes • Moving Average (MA) processes • ARMA / ARIMA / SARIMA • Exponential Smoothing (ETS, Holt-Winters)

Goal: *Fit statistical models to capture dependencies and make forecasts.*



Feature engineering for time series data

# Data Visualization



Interactive plots

D3, Plotly



Storytelling & Infographics

Principles of effective visualization

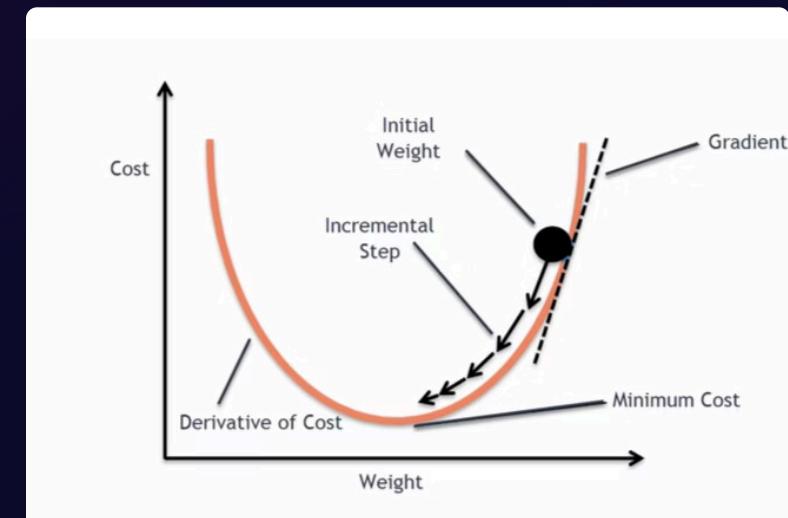
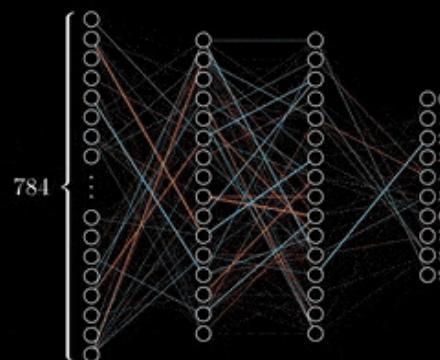
# Deep Learning Fundamentals



## Fundamentals

Neural Networks, Activation functions, Forward pass, Loss Functions, Back propagation, Gradient Descent, Model evaluation etc

Training in progress...

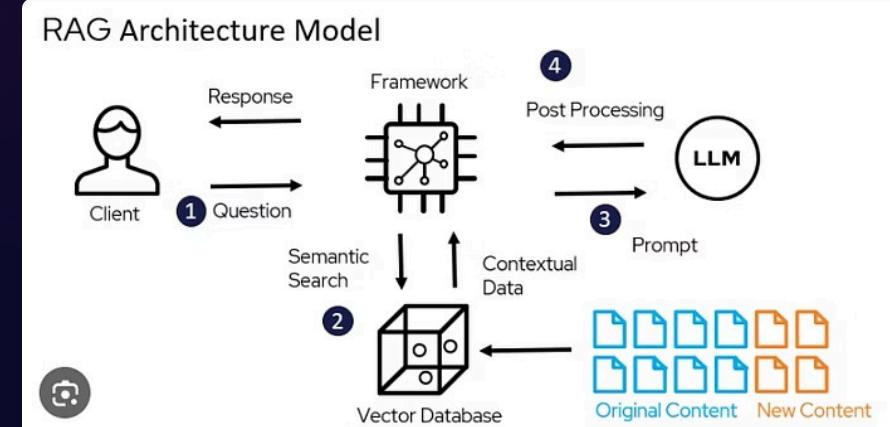
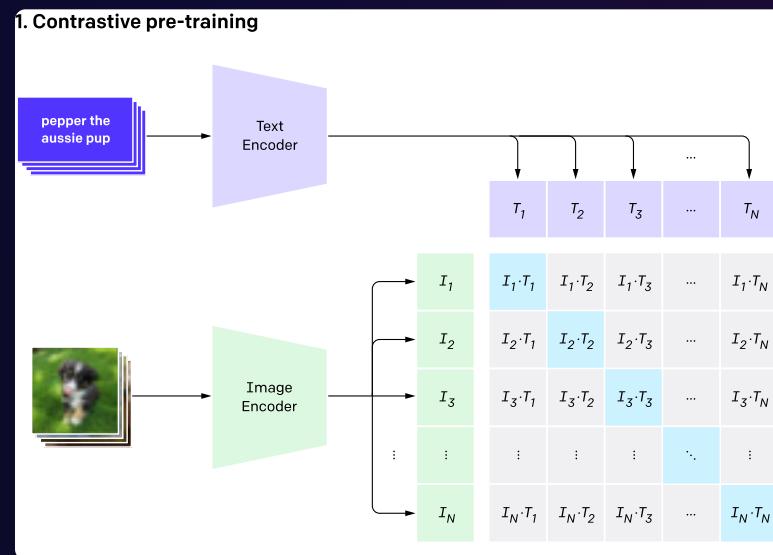


# Natural Language Processing

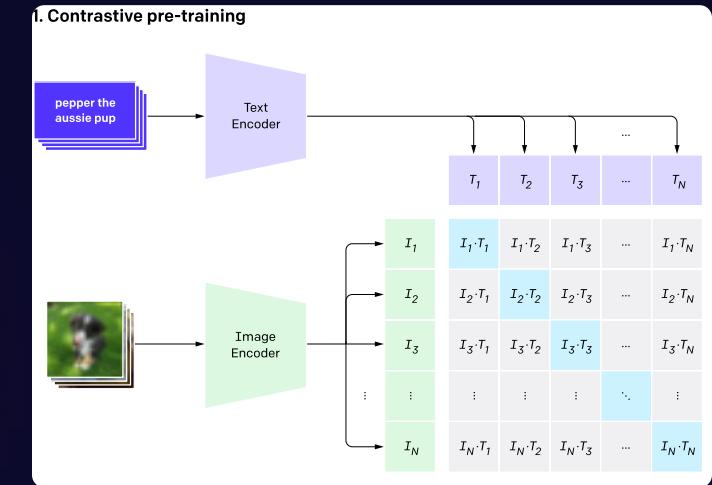
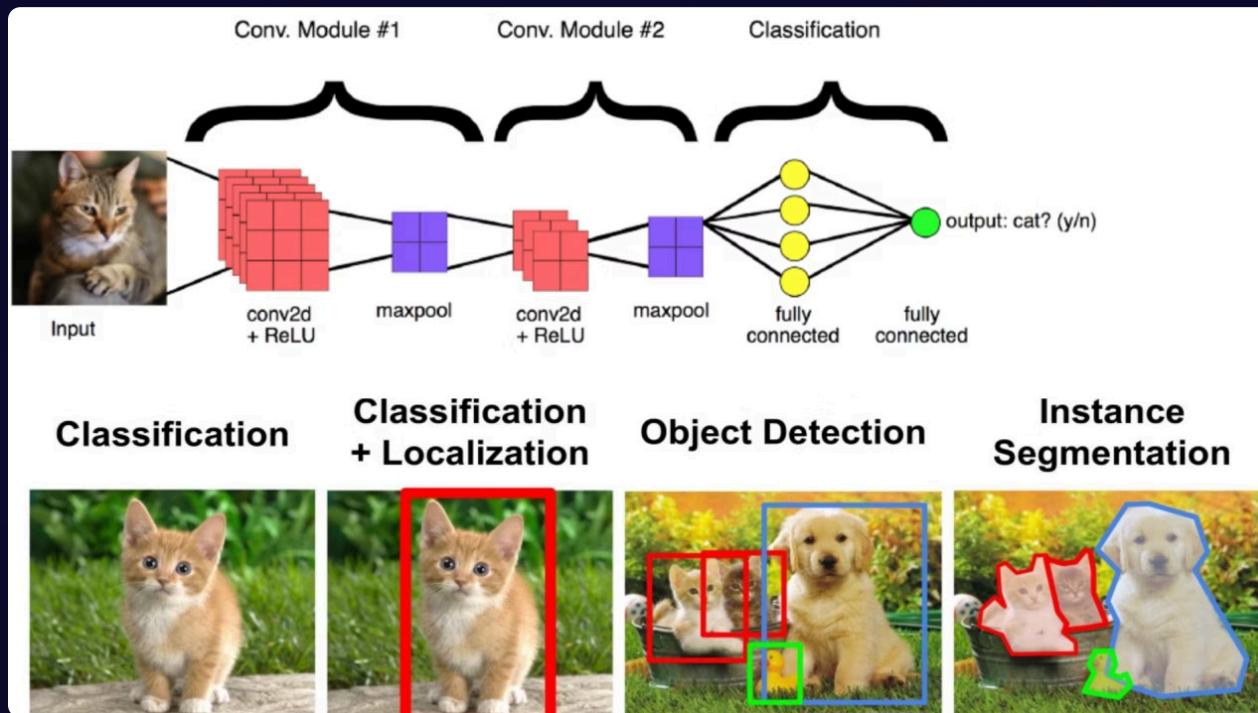
A|  
x

## Natural Language Processing

Text embedding algorithms, Text generation (next token predictions), Semantic similarity, Text generation models, Transformers, LLMs, Agentic Workflows, RAG. Fine-tuning, transfer learning, Quantization.



# Computer Vision



# Data Science Competition



Take on a Data competition



Work in teams



Collaboration/Coding best practices

KANBAN boards, Git,

# Advanced Deep Learning Algorithms



## Geometric Deep Learning

Graph Neural Networks (GNNs), Concept of Invariance and Equivariance for GNNs



## Reinforcement Learning

# Advanced Software Dev Skills (for ML)



Object Oriented  
Programming (Python)



Structuring ML projects



Docker



Databases



SQL



Testing

CI/CD pipelines, Types of tests.



UI & Prototype  
Development in Python  
  
Streamlit



Debugging



ML Ops

Developing and deploying an ML  
model. Setting up automated  
experiment tracking.

# Career Guidance



## Session with a Tech Recruiter

1:1 CV Review, insider knowledge about hiring processes.



## Business Communication

Techniques for effective communication in interviews and on the job.

# Portfolio Project



## Mentoring Sessions

Begins on November 12th. The 2 hour sessions with Francisco: Group mentoring sessions. You are free to book further 1:1 sessions with Francisco/Jose after this point.



## Decision Day

December 4th. You should have had sufficient 1:1 meetings and finalized your project by now. Send a 1 page 300 word abstract to Abin/Arun.



## Demo Day

Final presentation of your project: 10 minutes. January 10th.

# General Tips



## LLM usage

Do not overuse it, especially in class, ask your questions to the teachers/peers. Only implement code suggestions that you fully understand.



## Session pre-prep

Try to grasp the fundamental concept before each session, so that the tutor can go into the more advanced topics. Go through the Git/Colab notebooks. Install any required software, env setup etc.



## Stay Updated

Regularly check all the communication channels: Slack and Google Calendar.



## Project Ideas

Start thinking about potential project ideas: the more the better. Do not over-rely on information from LLMs.

# Review and Prep Days



## Review Day: October 10th

Review and ensure that you understand the fundamentals covered until this point.

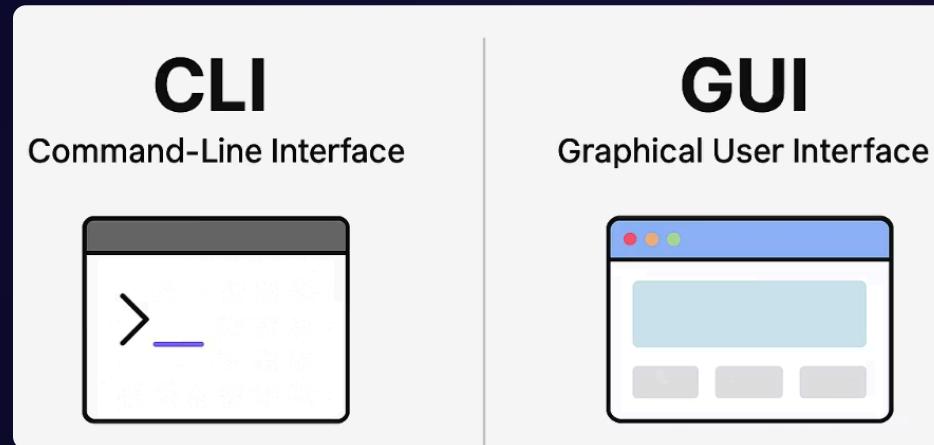


## Prep Day: October 15th

Go through and understand the contents on the repo for the next 2 important sessions.

# Software Setup

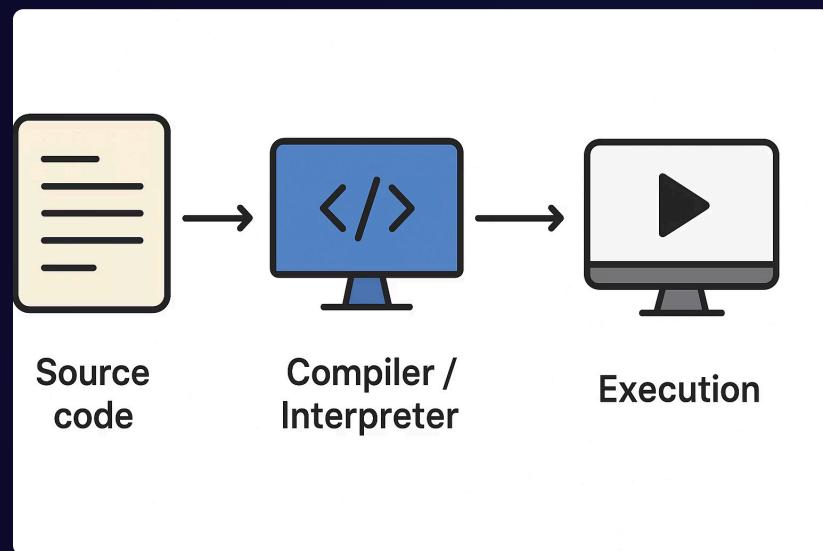
# The Command Line Interface (CLI)



**SHELL**: CLI program to the OS (manage files, make files etc.): Bash, Zsh, Powershell (Windows).  
Run the SHELL on app called Terminal.

The session on September 24th goes into the details of Bash.

# What is source code?



## Compiler/Interpreter

They **understand specific instructions** written in a programming language and convert them into something the machine can execute. e.g., Python (Interpreter), C++ (Compiler), etc.

A compiler **translates the source code into another form** (usually machine code or bytecode) for execution. An interpreter, in contrast, directly executes instructions line by line.

## Source Code

The recipe (written in text, as a text file) for the specific compiler/interpreter.

File extensions like `.py` or `.cpp` are mainly for human readability, editors, and tooling; essentially, they are all text files.

# Code Editors vs. IDEs

	Code editors	IDEs
 Feature set	Provides standard code editing functionality, with modern code editors including more advanced functionality.	Provides comprehensive functionality.
 Language support	Polyglot with support for additional languages through plugins or extensions.	Provides extensive support for specific programming languages and frameworks.
 Extensibility	Lightweight feature set that can be built on with community-driven extensions.	Large feature set already built in with APIs for developing community extensions.
 Resource usage	Consumes fewer system resources.	Consumes more system resources.
 Learning curve	Generally simpler, less complex, and easier to learn.	Generally a wider range of features, more complex, and harder to learn.
 Integration with external tools	Integrates with external tools through plugins or extensions, which can need some configuration.	Preconfigured integration with external tools and services commonly used in software development.
 Examples	VS Code, Cursor, Zed, Nova, Sublime Text**, Vim**, Notepad++** <small>**Text editors</small>	JetBrains IDEs (IntelliJ IDEA, WebStorm, PhpStorm, PyCharm, Rider), Visual Studio, Eclipse

# VSCode

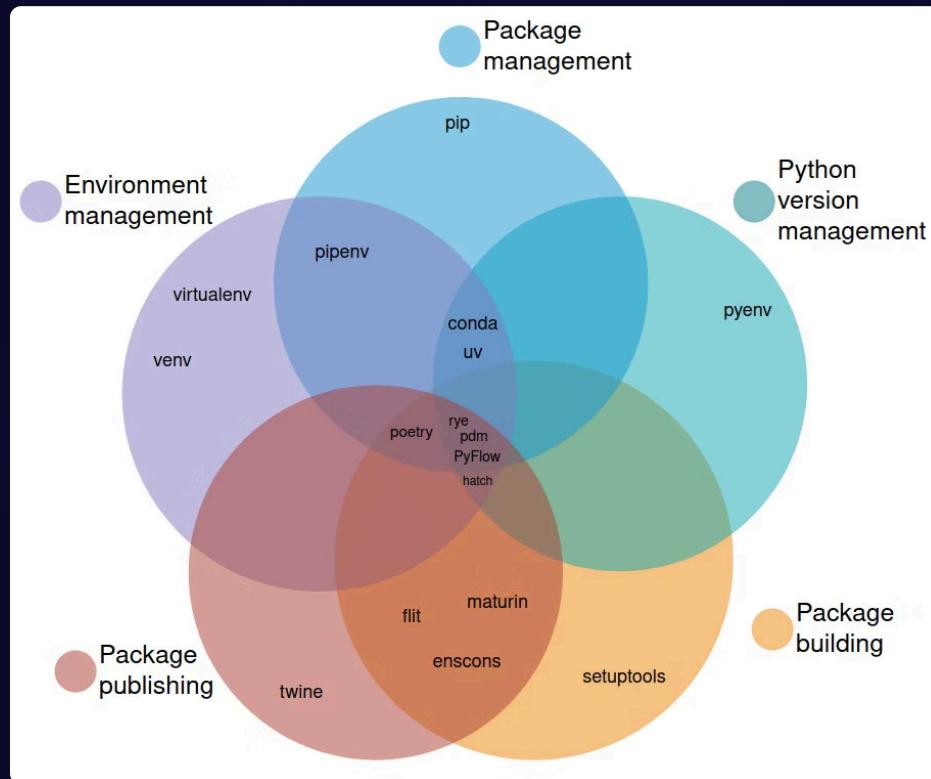


Light-weight code editor that can be configured to be an IDE for any programming language.

Extensions for AI coding agents.

[Download VSCode](#)

# Python Environment Management



Popular choices:

 pip + venv + pyenv

 conda (Anaconda, Miniconda)

 uv (fast, very flexible)

More detailed discussion during the session on 13th & 14TH October.

# Git



## Version Control

Record and track all changes to all files in a directory (repository).



## Branches

Create branches (or copies) for parallel development on different parts.



## Locally

Version controlling locally on your machine.



## GitHub hosting

Host your Git repositories on cloud using GitHub.



More on September 2

# My Project Structure

Each project in a separate directory.



Separate venv for each project.



VSCode

Version controlling locally on your machine.



uv

Host your Git repositories on cloud using GitHub.

Ask me anything!

Thank you and Good Luck!