https://ttpoll.com/p/569627

COSC 325: Introduction to Machine Learning

Dr. Hector Santos-Villalobos

Dr. Santos

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Lecture 02: Machine Learning Motivation

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

If you are not registered in this section, please yield your seat to registered students.

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Are you in the right place?

- Course
  - Course: COSC 325, Introduction to Machine Learning
  - Pre-Requisites: ECE 313 or ECE 317 or MATH 323; and MATH 251 or MATH 257 with a grade of C or better.
  - Recommended Background: Python Programming Language, Numpy, SciKit Learn
- Logistics
  - Location: MKB 524
  - Lectures: Tuesday/Thursday: 11:20 am to 12:35 pm
  - Canvas Link: https://utk.instructure.com/courses/206990

Discord Server

UTK Fall-24 COSC 325

Join @

https://discord.gg/DXpnvT9R

Class Announcements

Homework:

- First homework available in Canvas

Exams:

Lectures:

Join Discord Server:
UTK Fall-24 COSC 325
https://discord.gg/DXpnvT9R

Quizzes:

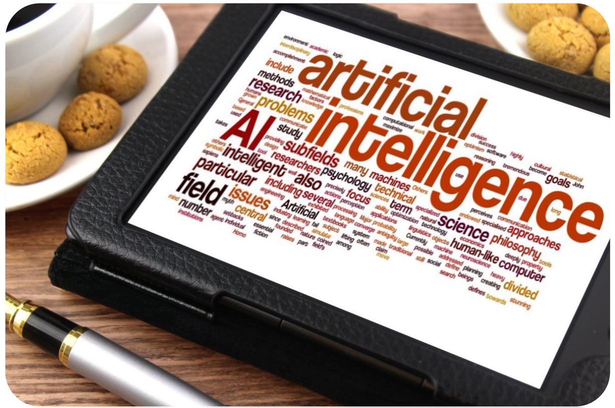
- The first quiz is due on Sunday.

Course Project:

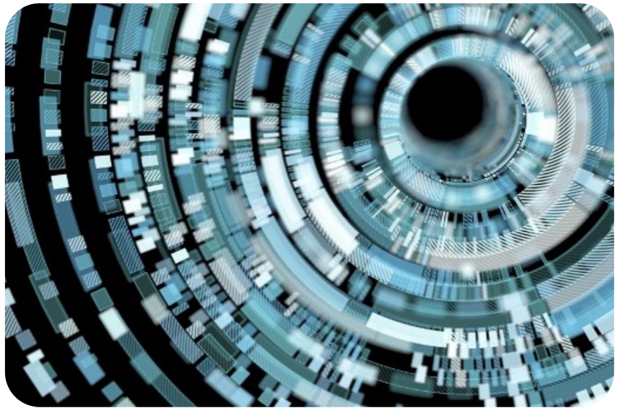
- Instruction files will be available early next week.
- Form teams by 08/29. Otherwise, we will randomly assign you a team.
- Send your team details via Discord #team-creation channel.
- Students working in teams of two may be paired with a third student.

Today’s Topics

What is Machine Learning?



Notation, Applications, and Tools



Last Lecture

- Course overview
- Work hard and have fun
- We are here to learn

https://ttpoll.com/p/569627

What is Machine Learning?

“Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed.”
— Arthur L. Samuel, AI pioneer, 1959

https://ttpoll.com/p/569627

“…machine learning is a subcategory within the field of computer science which allows you to implement artificial intelligence. So, it’s kind of a mechanism to get you to artificial intelligence.”

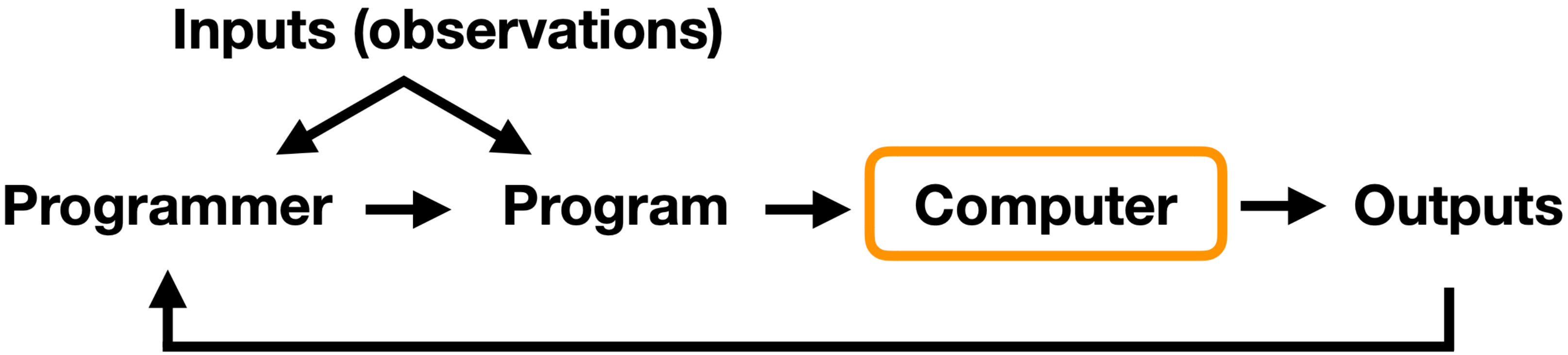
—Rana el Kaliouby, CEO at Affectiva

Machine Learning vs Computer Programming

- Traditional Programming:
  - Algorithms are sequences of instructions that are carried out to transform an input into an output
  - Fundamentally, they are lists of instructions
- Machine Learning:
  - The list of instructions is ***Learned*** from data
  - Useful when the sequence of instructions is difficult to define
  - Examples
    - Facial recognition
    - Autonomous driving

https://ttpoll.com/p/569627

Traditional Programming Paradigm

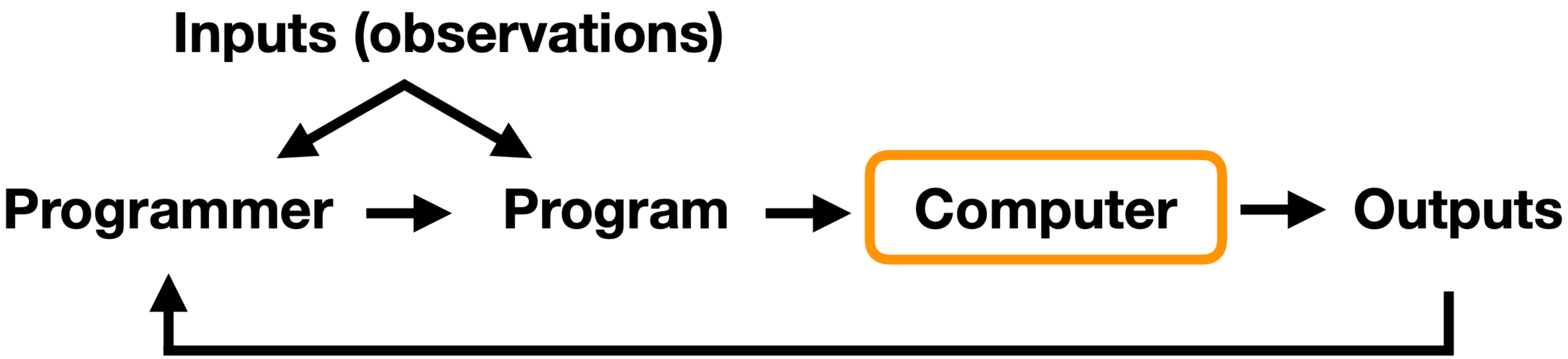


Inputs (observations)

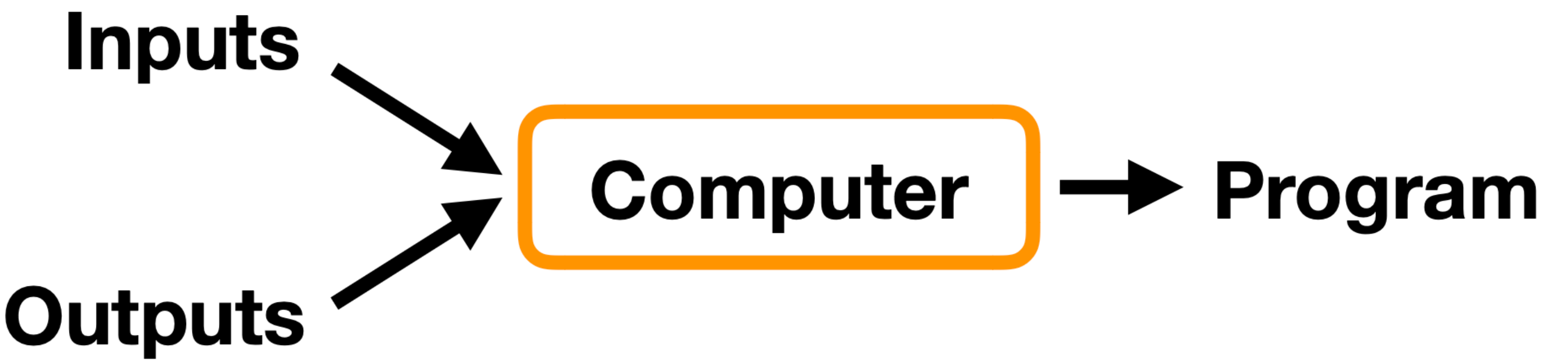
Programmer → Program → Computer → Outputs

Slide credit: Dr. Raschka

https://ttpoll.com/p/569627



“Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed.”
— Arthur L. Samuel, AI pioneer, 1959

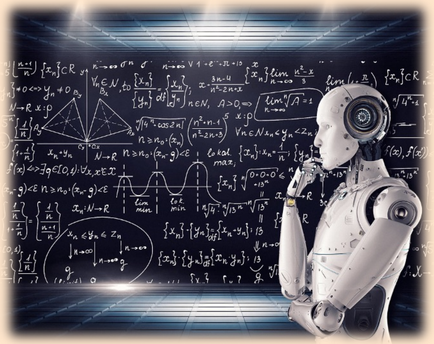


Slide credit: Dr. Raschka

AI vs Machine Learning

Artificial General Intelligence

Computers “mimic” how humans learn.

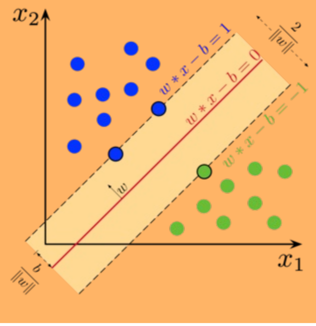


Artificial Intelligence

Computers mimic human behavior.

Machine Learning

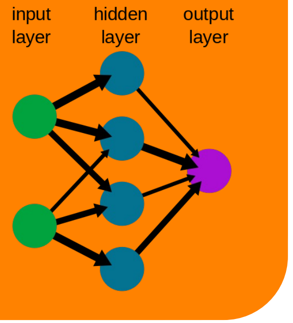
Ability to learn without explicit hand-made rules.



Deep Learning

Automated extraction of patterns/features from raw data using multi-layer neural networks.

A simple neural network



Teaching computers how to **learn a task** directly from **data**.

Figure inspired on MIT 6.S191 course slide.

Memory vs Learning

- What is an example of something that you memorize?
- What is an example of something that you learn?

https://ttpoll.com/p/569627

Memory vs Learning

- What is an example of something that you memorize?
- What is an example of something that you learn?

1520 Middle Drive, Knoxville,
Tennessee 37796

https://ttpoll.com/p/569627

Generalization

- A key component of "learning" is the ability to generalize
  - Take information that has been learned previously and apply it to new but related scenarios
- For a technique to be considered a machine learning approach, it ***must*** be able to generalize
- Thus, we must evaluate its ability to generalize

Slide credit: Dr. Schuman

https://ttpoll.com/p/569627

Memory vs Learning

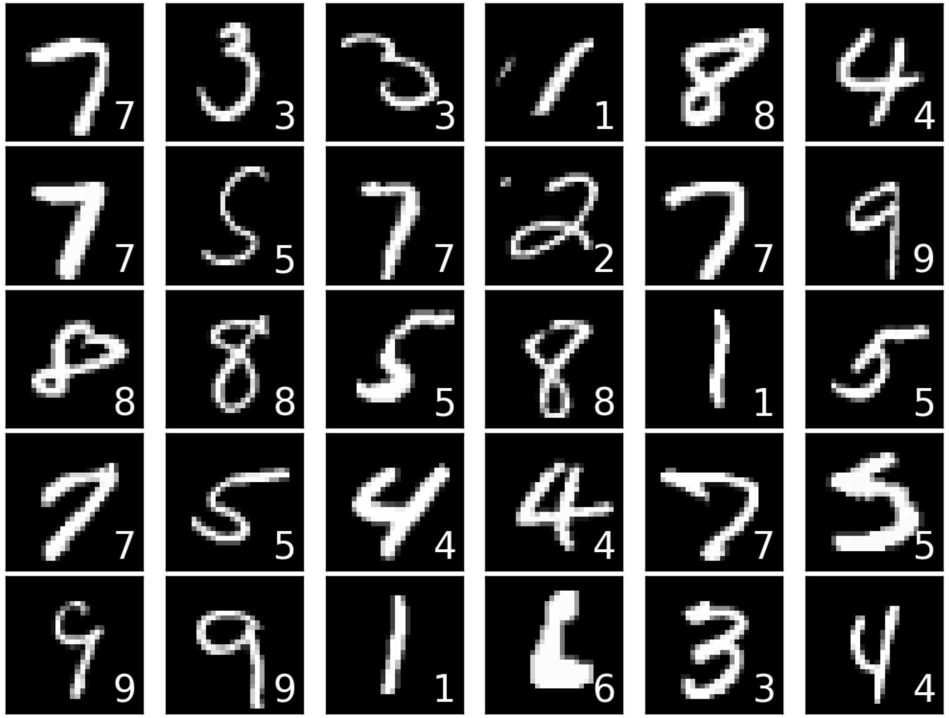
- What is an example of something that you memorize?
- What is an example of something that you learn?

1520 Middle Drive, Knoxville,
Tennessee 37796

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." **— Tom Mitchell, Professor at Carnegie Mellon University**

MNIST Images

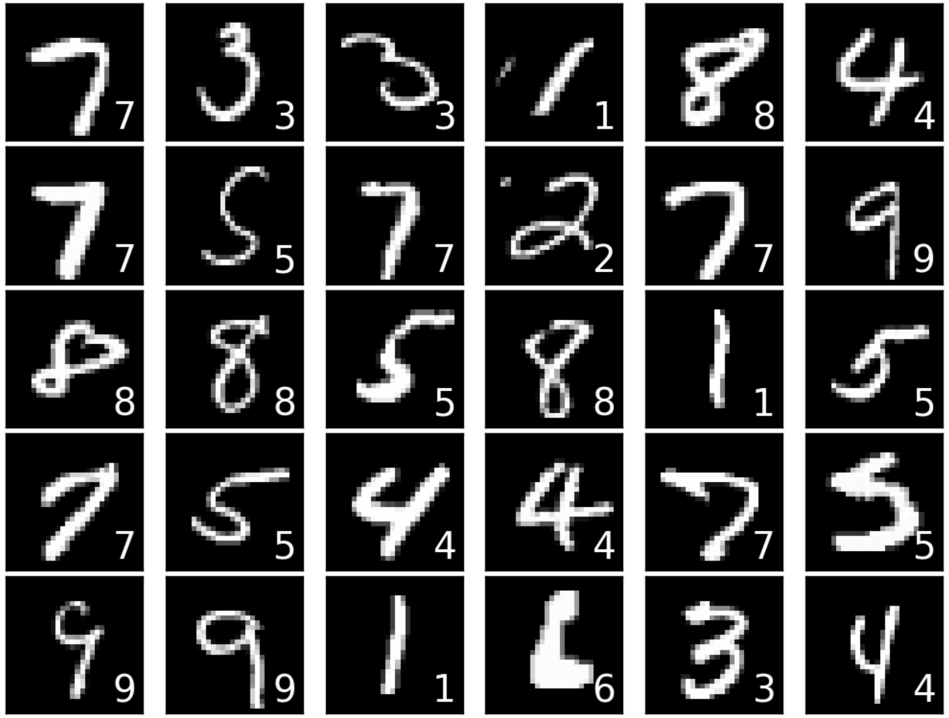
- Task T:__________
- Performance measure P: _________
- Training experience E: __________



https://ttpoll.com/p/569627

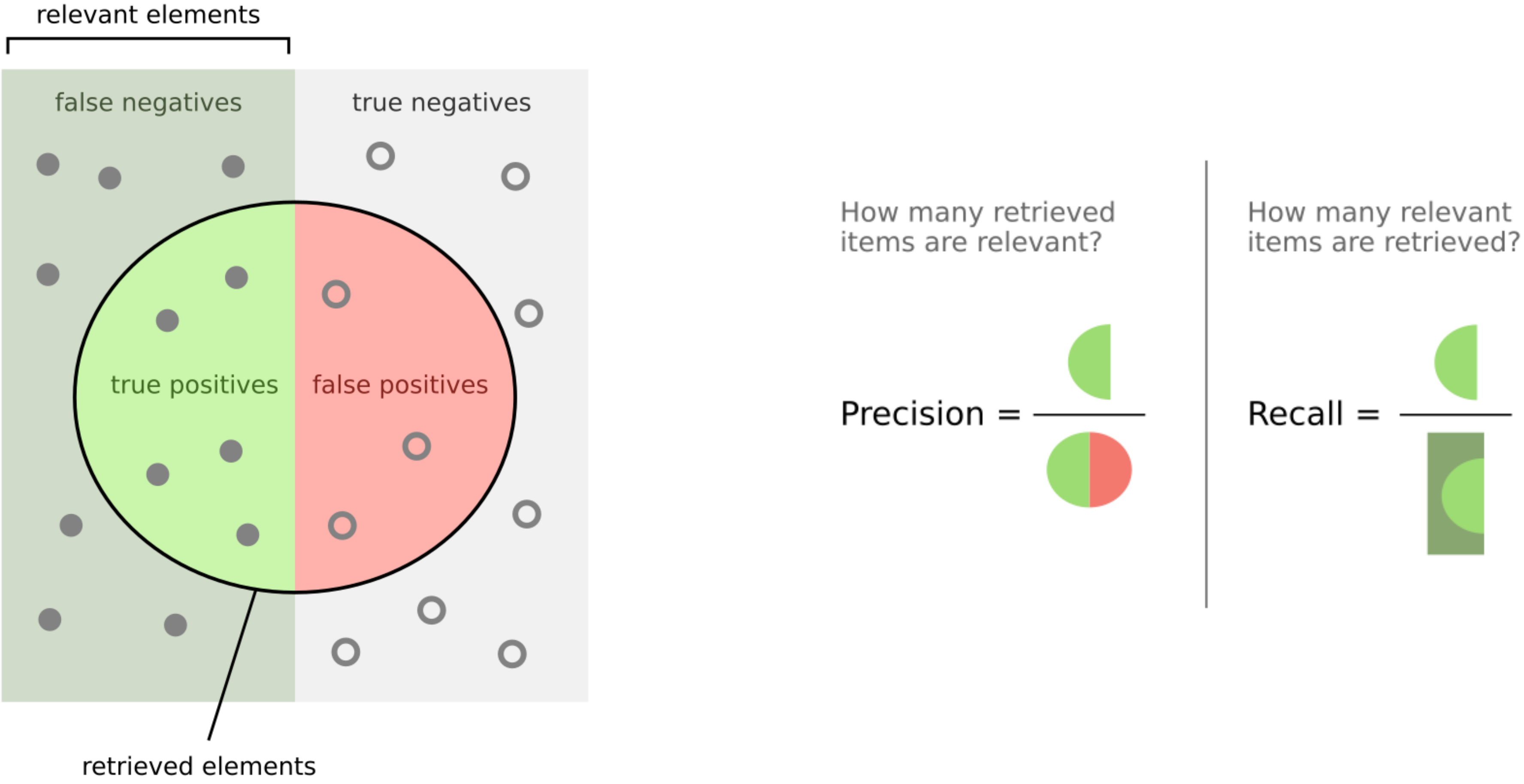
MNIST Images

- Task T:__________
- Performance measure P: _________
- Training experience E: __________



Features

Performance Metrics



How many retrieved items are relevant?

Precision =

How many relevant items are retrieved?

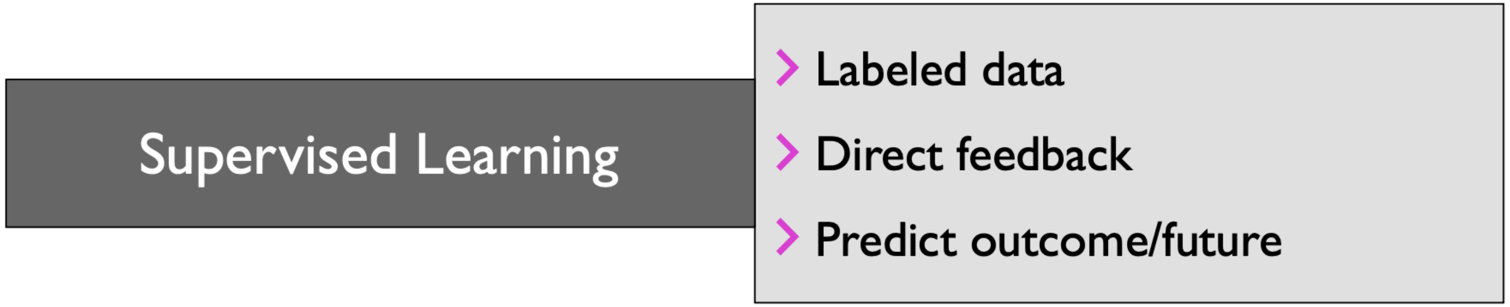
Recall =

https://ttpoll.com/p/569627

Categories of Machine Learning

https://ttpoll.com/p/569627

Categories of Machine Learning



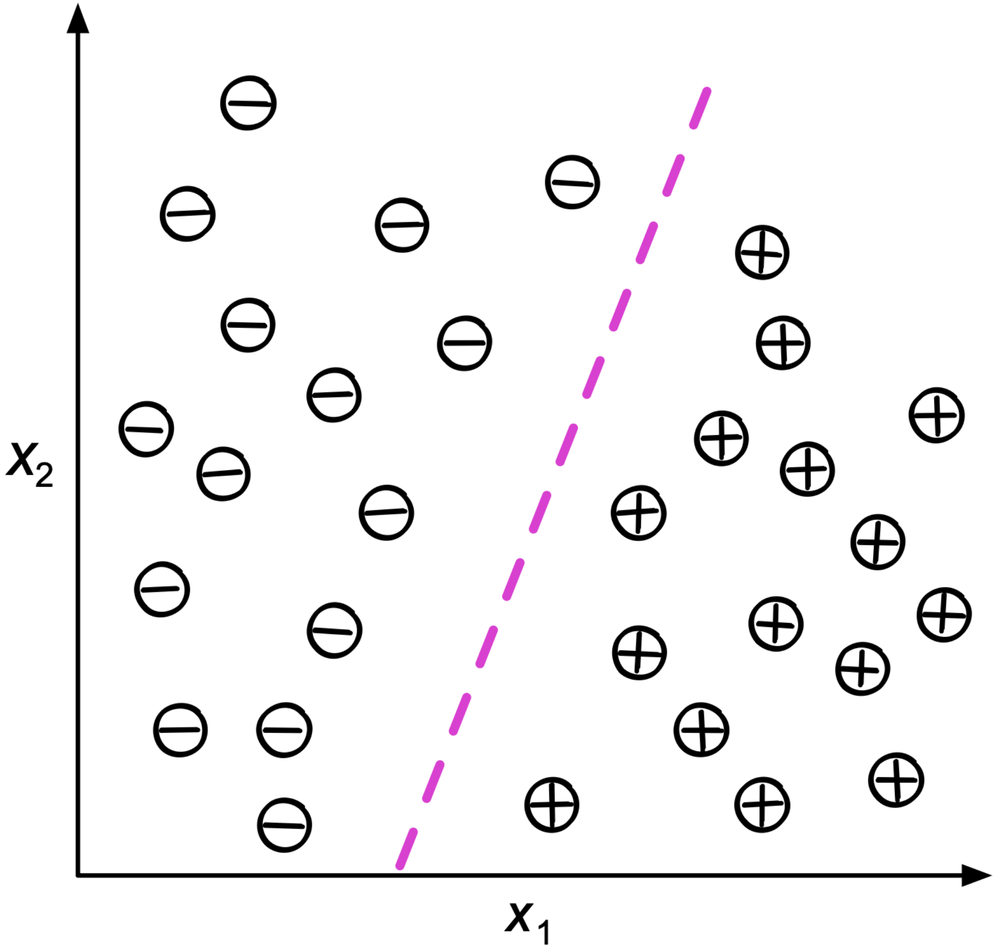
Supervised Learning

- Labeled data
- Direct feedback
- Predict outcome/future

Slide credit: Dr. Raschka

https://ttpoll.com/p/569627

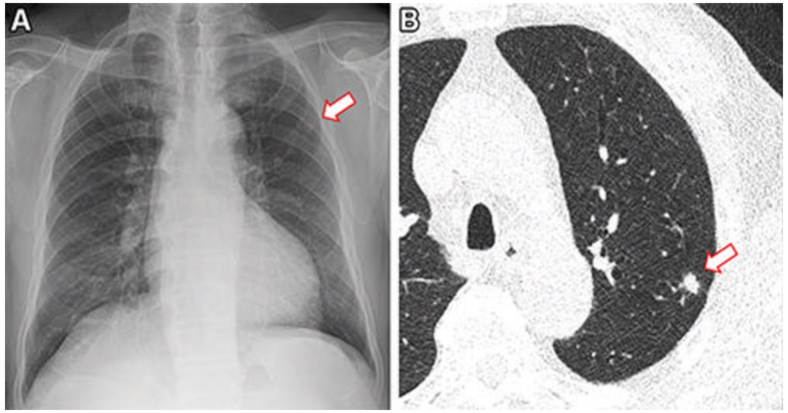
Supervised Learning: Classification



Slide credit: Dr. Raschka

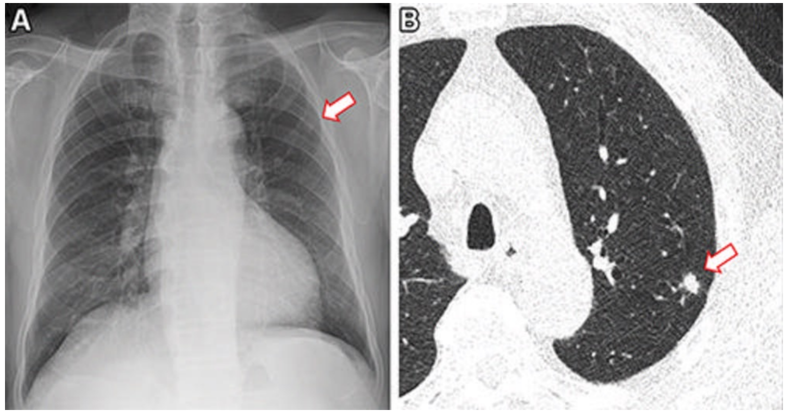
Binary Classification

Determine whether a tumor is benign or malign



- Task:
- Features/Experience:
- Performance:

Binary Classification



Determine whether a tumor is benign or malign

- Task: Yes/No Predictions
- Features/Experience: Radius (mean of distances from the center to points on the perimeter), texture (standard deviation of gray-scale values), perimeter, area, smoothness (local variation in radius lengths), compactness (perimeter^2 / area - 1.0), concavity (severity of concave portions of the contour), concave points (number of concave portions of the contour), symmetry, fractal dimension.
- Performance: Accuracy, Binary Cross Entropy, Precision, Recall

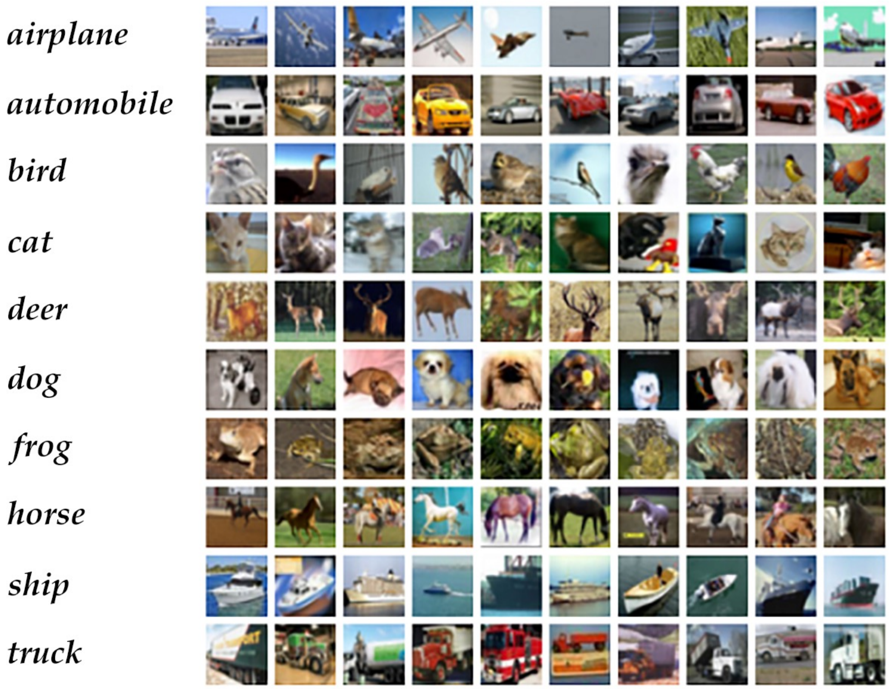
https://ttpoll.com/p/569627

Multi-Class Classification

Assign object label name to image

- Task:
- Features/Experience:
- Performance:

CIFAR10



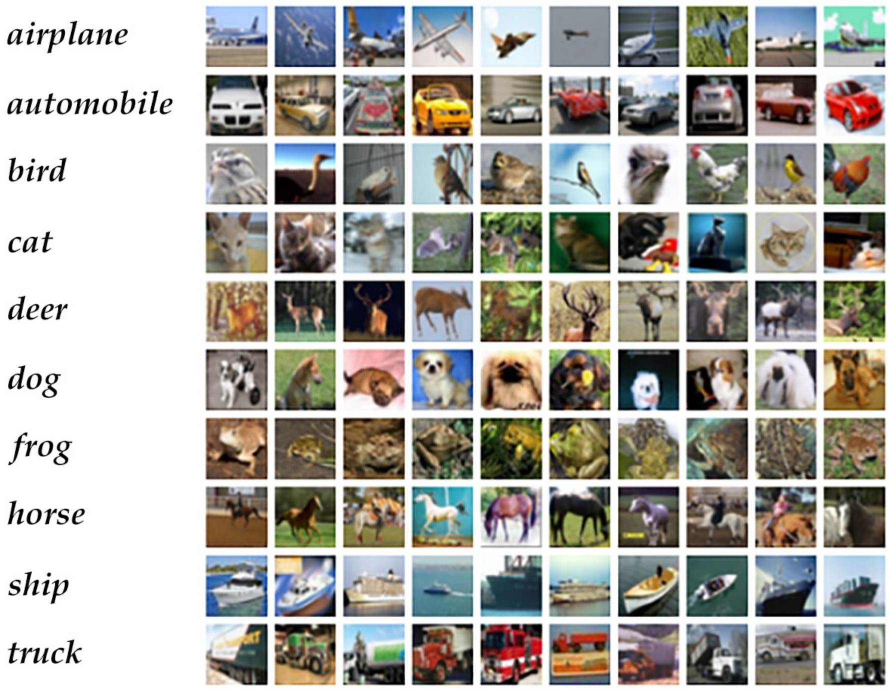
Multi-Class Classification

Assign object label name to image

- Task: Select a label within a set of possible predictions (labels)
- Features/Experience: ***pixel values****, color histogram, edge angle histogram, correlation filter response
- Performance: Precision, Recall, Accuracy, F1 Score

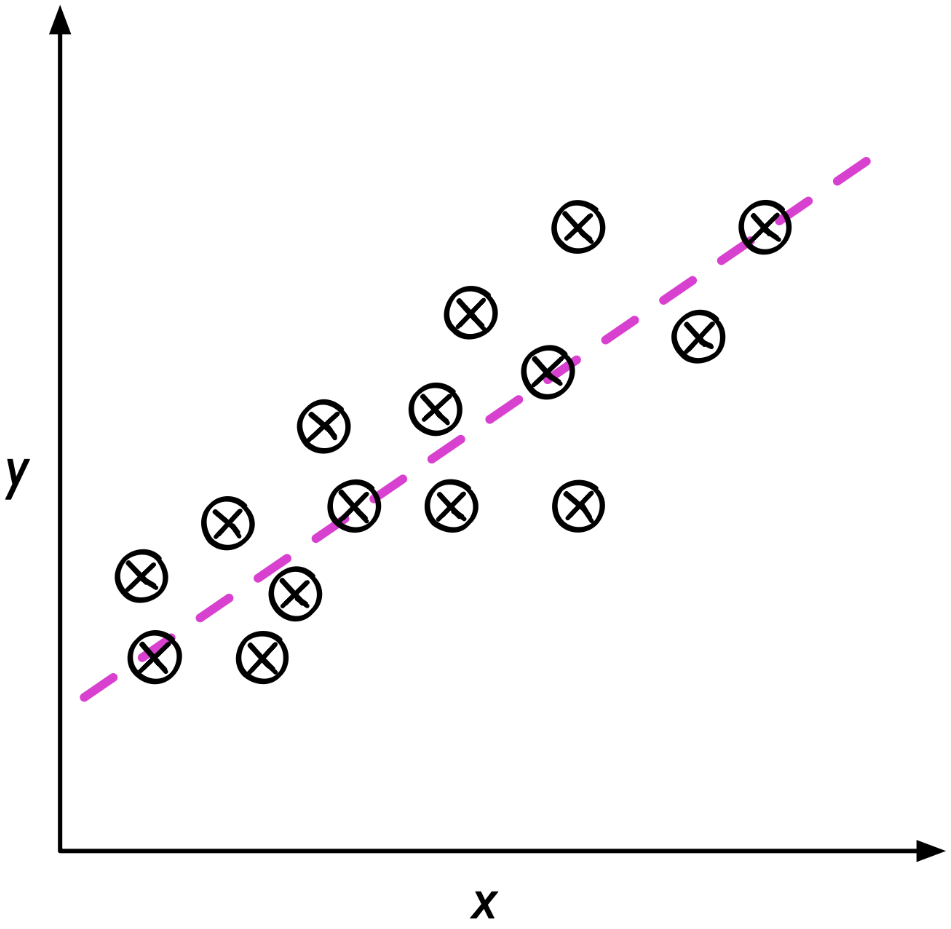
***NNs**

CIFAR10



https://ttpoll.com/p/569627

Supervised Learning: Regression



Examples:

- Market value
- Time to failure
- Age
- Weight
- Size
- Location

Slide credit: Dr. Raschka

Regression Example

Quantitative measure of diabetes disease progression one year after baseline.

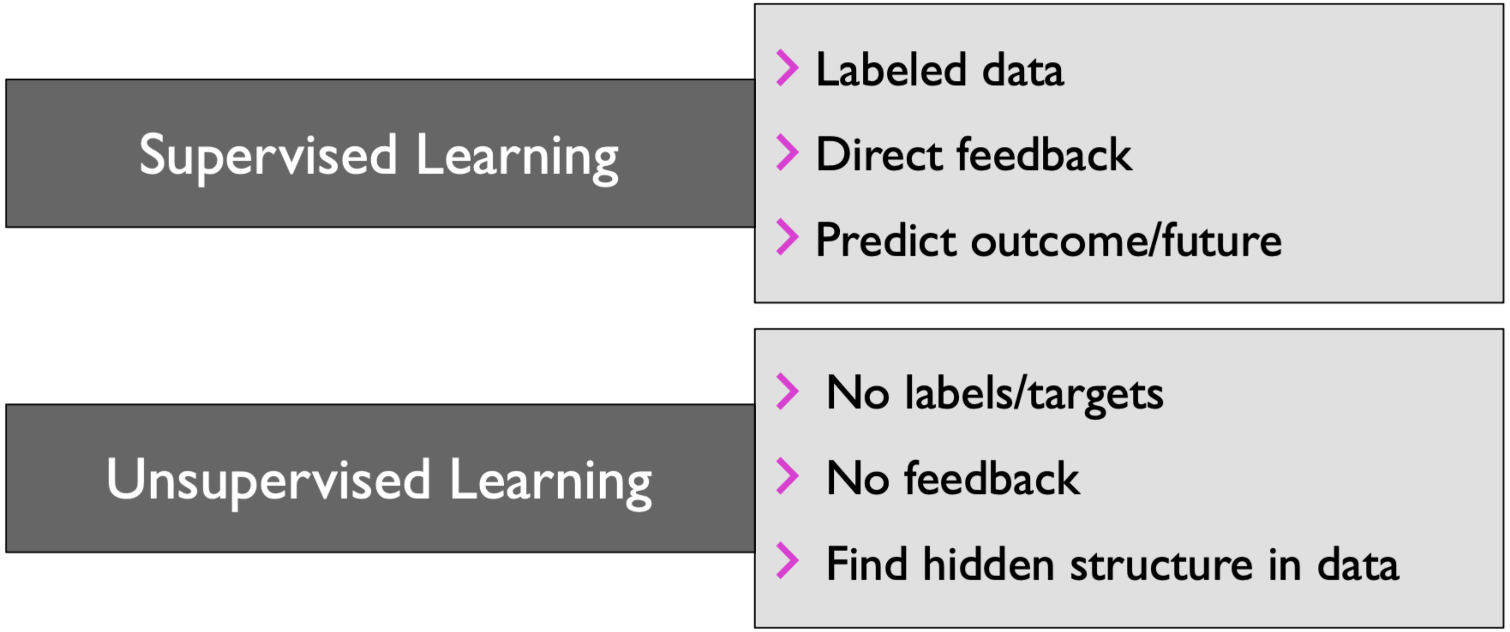
- Task:
- Features/Experience:
- Performance:

Regression Example

Quantitative measure of diabetes disease progression one year after baseline.

- Task: Predict a real value
- Features/Experience: Age in years, sex, BMI (body mass index), average blood pressure, total serum cholesterol, low-density lipoproteins, high-density lipoproteins, total cholesterol / HDL, possibly log of serum triglycerides level, and blood sugar level
- Performance: MSE between predicted and true values

Categories of Machine Learning



Supervised Learning

- Labeled data
- Direct feedback
- Predict outcome/future

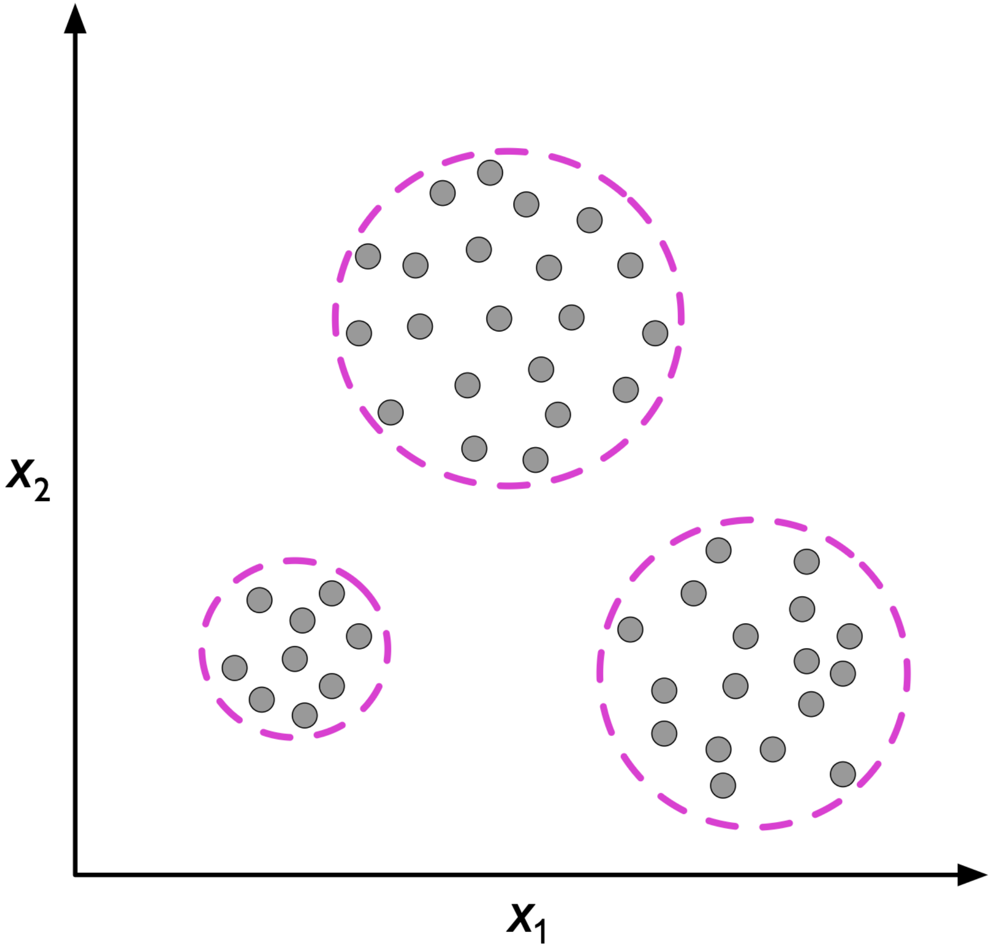
Unsupervised Learning

- No labels/targets
- No feedback
- Find hidden structure in data

Slide credit: Dr. Raschka

https://ttpoll.com/p/569627

Unsupervised Learning: Clustering



Slide credit: Dr. Raschka

Ranking Example

Return the top 10 most similar movies to a query movie

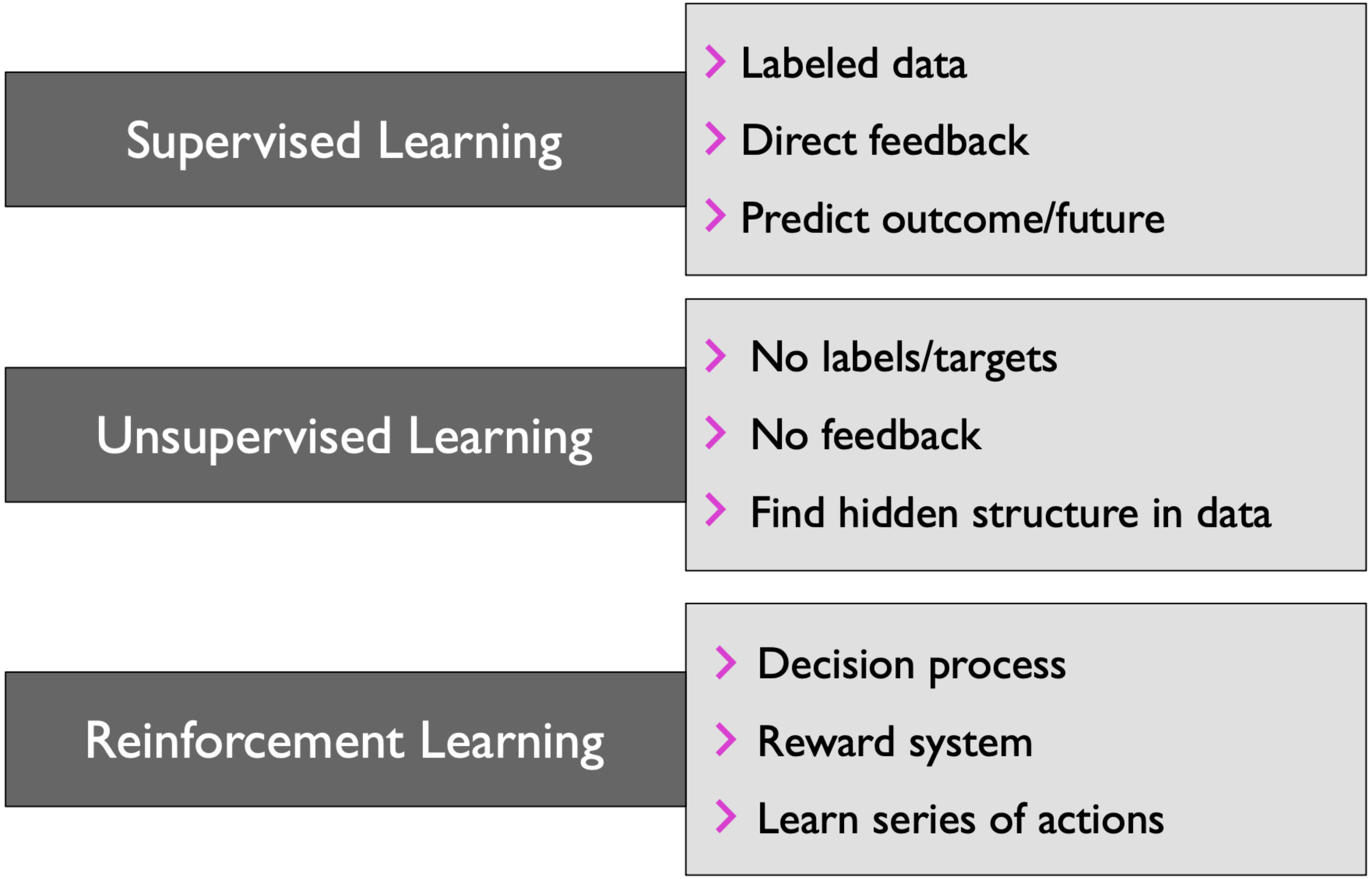
- Task: Compute relevance list
- Features/Experience:
- Performance:

Ranking Example

Return the top 10 most similar movies to a query movie

- Task: Compute relevance list
- Features/Experience: title, genre, year, synopsis, language, director, starring actors, studio, content advisory, maturity rating, [customer watch history].
- Performance: Recall@k from past user behavior

Categories of Machine Learning



Supervised Learning

- Labeled data
- Direct feedback
- Predict outcome/future

Unsupervised Learning

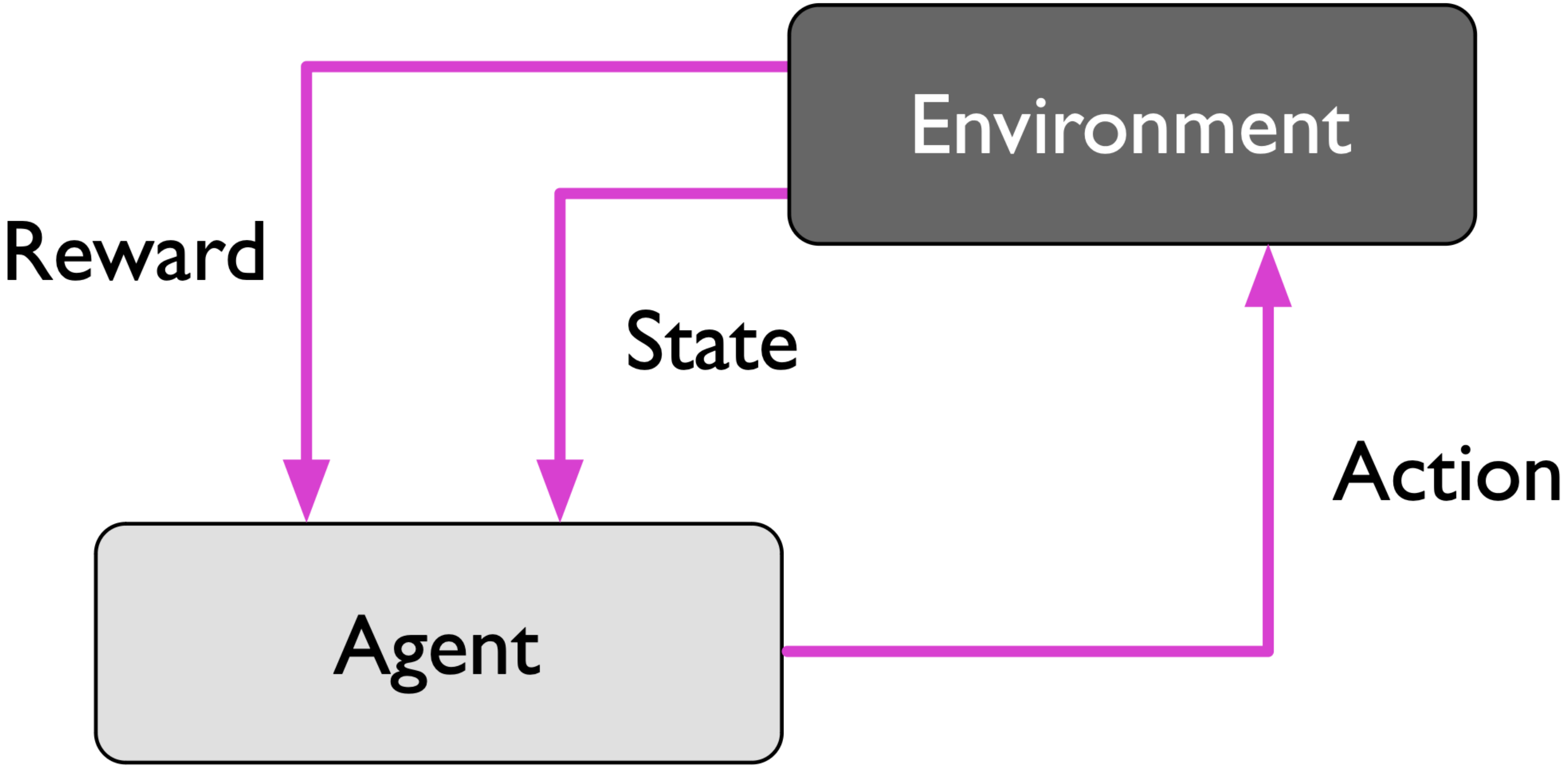
- No labels/targets
- No feedback
- Find hidden structure in data

Reinforcement Learning

- Decision process
- Reward system
- Learn series of actions

Slide credit: Dr. Raschka

Reinforcement Learning



Environment

Reward

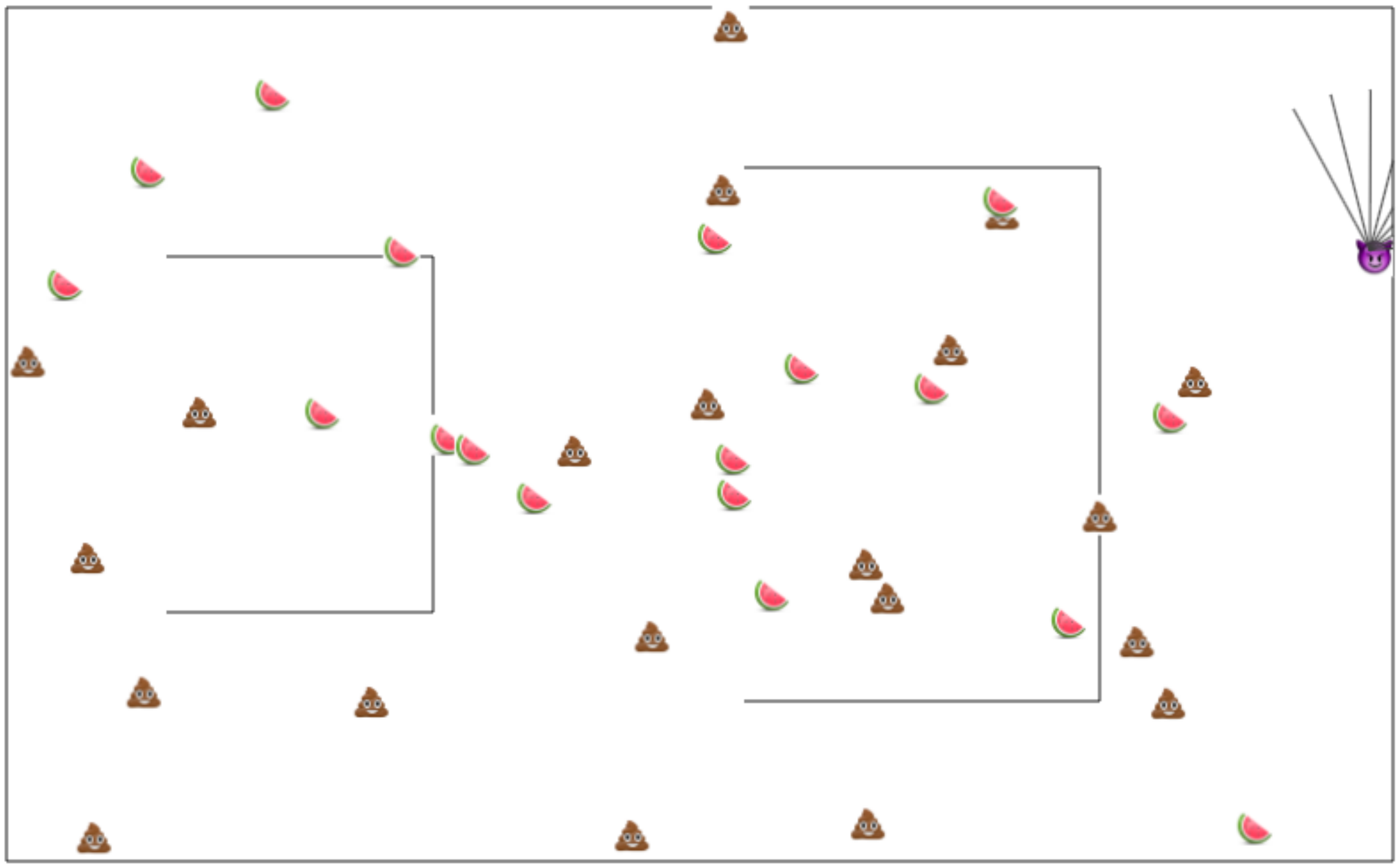
State

Action

Agent

Slide credit: Dr. Raschka

Example

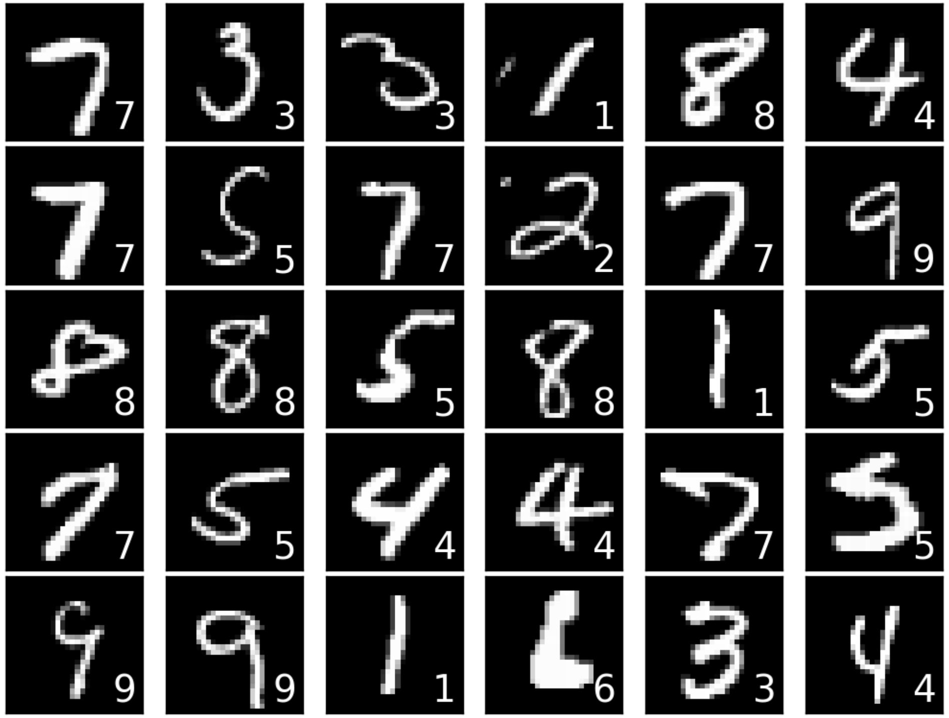


https://projects.rajivshah.com/rldemo/

https://ttpoll.com/p/569627

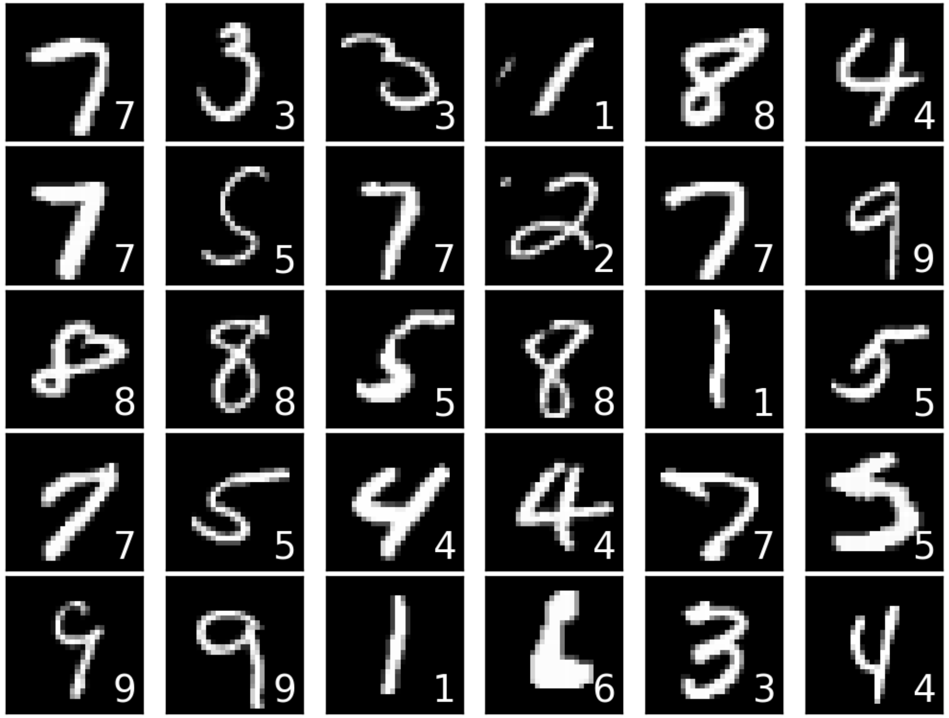
MNIST Images (Pop Quiz)

- Task T:__________
- Performance measure P: ________
- Training experience E: __________



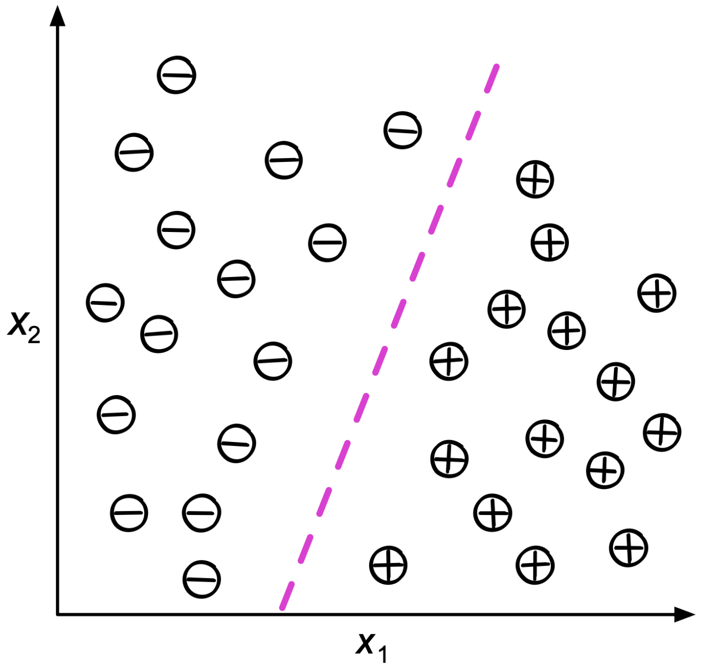
MNIST Images (Pop Quiz)

- Task T: Multi-class labeling
- Performance measure P: Accuracy, precision, etc.
- Training experience E: Edge histogram, correlation filter response, etc.

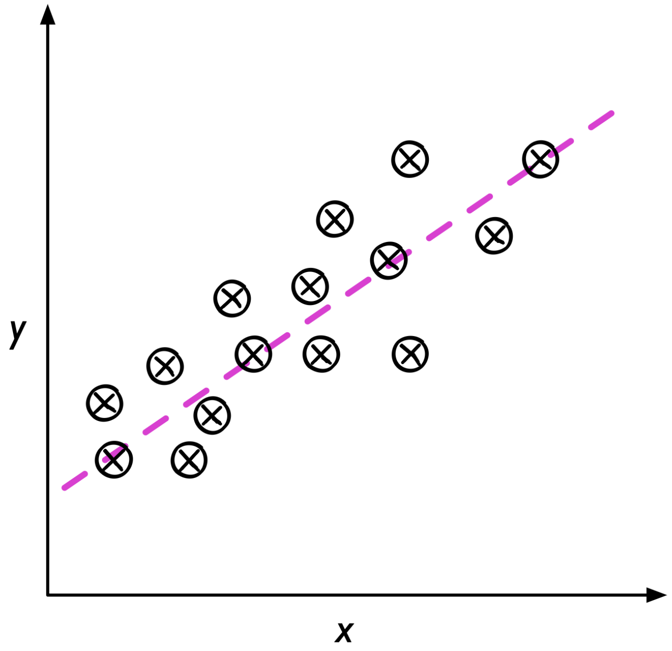


https://ttpoll.com/p/569627

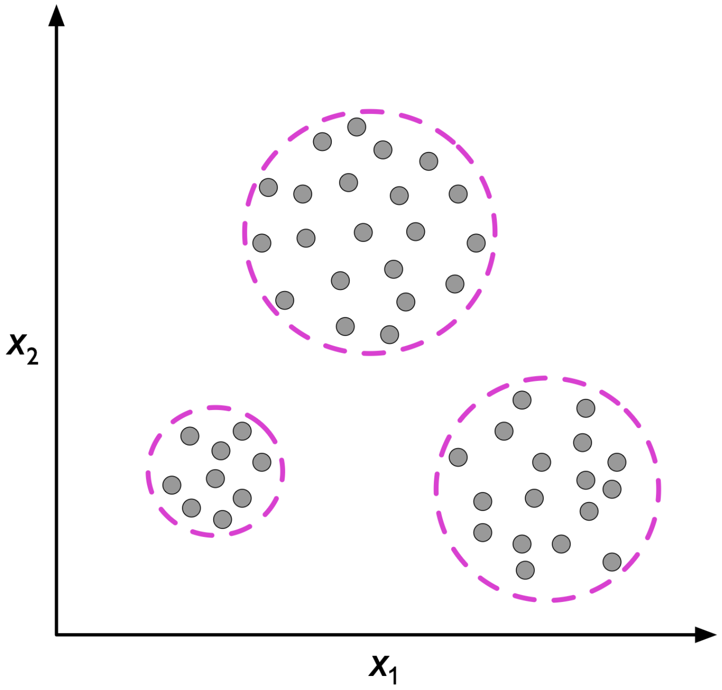
What do these plots have in common?



Classification



Regression



Clustering

Python

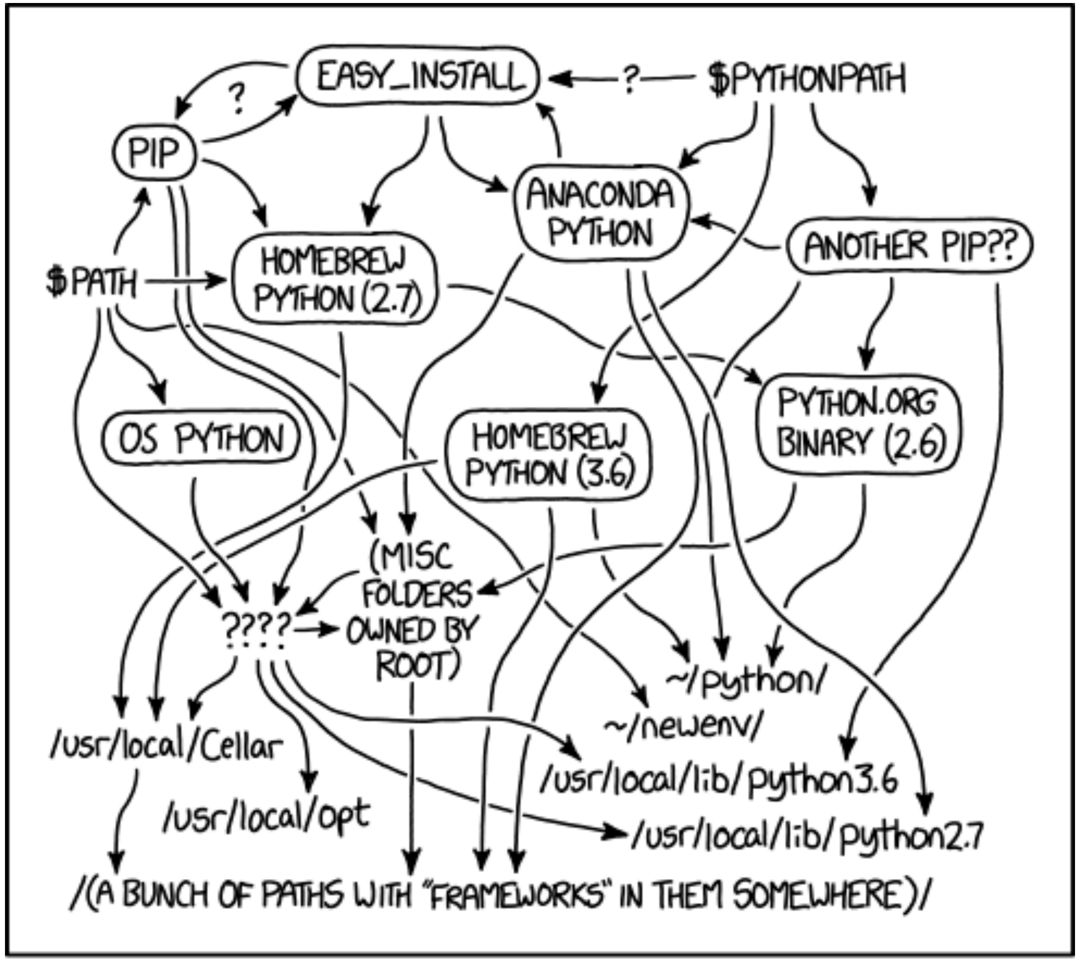
Why Python?

- **Extensive Libraries and Frameworks:** Rich ecosystem with tools like TensorFlow, PyTorch, and Scikit-Learn simplifies complex algorithm implementation.
- **Ease of Learning and Use:** Clear, readable syntax accelerates learning and allows rapid prototyping.
- **Active Community and Support:** Vast, active community offers extensive documentation, tutorials, and forums for support.
- **Versatility and Integration:** Seamlessly integrates with other languages and platforms, supporting various programming paradigms.
- **Data Handling Capabilities:** Efficient data structures and functions for handling large datasets through libraries like NumPy and Pandas.
- **Visualization Tools:** Powerful libraries like Matplotlib and Seaborn aid in data exploration and result presentation.
- **Industry Adoption and Support:** Widespread use in academia and industry ensures ongoing investment and support.

https://ttpoll.com/p/569627

Setting up Python

- Python 3
- Environments:
  - Anaconda
  - PyEnv
  - Virtual Environments
- Pip
- Jupyter Notebooks
- Git



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Source: https://xkcd.com/1987/

Anaconda

- **Comprehensive Package Management:** Anaconda includes over 7,500 open-source pre-installed packages and simplifies the installation and management of additional packages through Conda, ensuring compatibility and reducing dependency conflicts.
- **Environment Management:** Easily create, manage, and switch between multiple isolated environments, allowing for clean and reproducible project setups without interfering with system-wide settings.
- **Integrated Development Environment:** Comes with Jupyter Notebook and Spyder, providing powerful, interactive tools for developing and testing code, data analysis, and visualization.
- **Cross-Platform Support:** Compatible with Windows, macOS, and Linux, ensuring consistent development environments across different operating systems.
- **Enhanced Data Science Tools:** Bundles essential libraries for machine learning, data analysis, and scientific computing (e.g., NumPy, Pandas, SciPy, and Matplotlib), streamlining the setup process for data science projects.

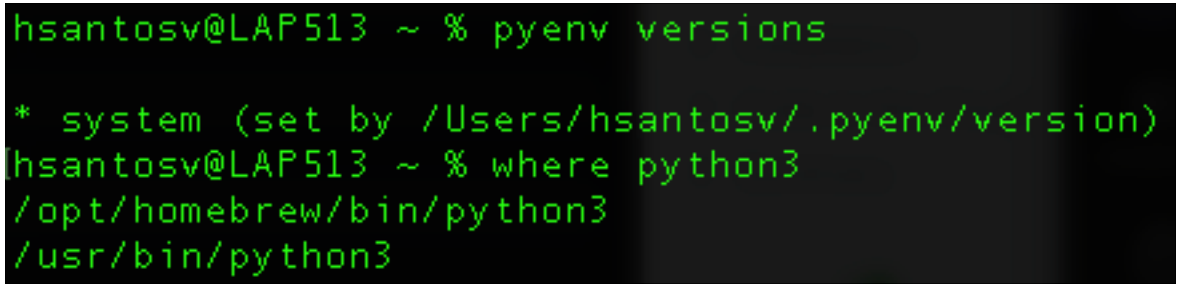
Important!

Don’t mess with your operating system’s Python installation.

Python Installation

- PyEnv in MacOS

```
# install pyenv requirements
brew install openssl readline sqlite3 xz zlib

# install pyenv (add to PATH)
brew install pyenv

# list available python versions
pyenv versions
```



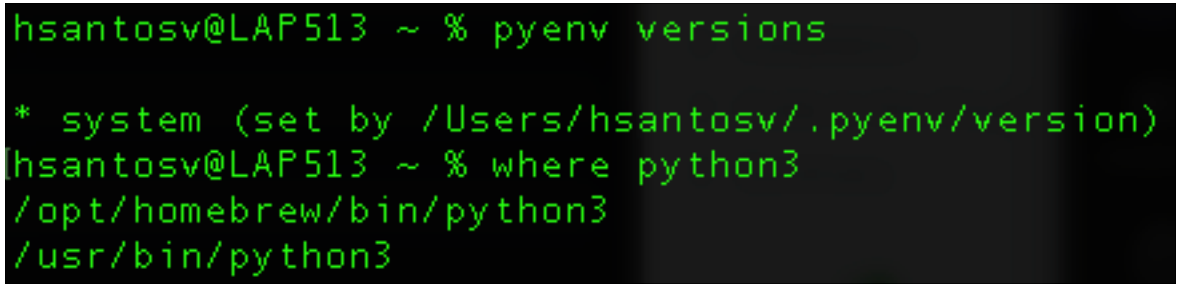
```
hsantosv@LAP513 ~ % pyenv versions

* system (set by /Users/hsantosv/.pyenv/version)
[hsantosv@LAP513 ~ % where python3
/opt/homebrew/bin/python3
/usr/bin/python3
```

Python Installation

- PyEnv in MacOS

```
# install pyenv requirements
brew install openssl readline sqlite3 xz zlib

# install pyenv (add to PATH)
brew install pyenv

# list available python versions
pyenv versions
```

Make sure you tell your OS where to find these tools.

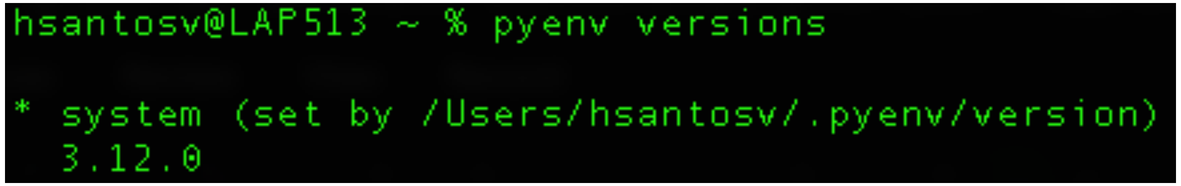


```
hsantosv@LAP513 ~ % pyenv versions

* system (set by /Users/hsantosv/.pyenv/version)
hsantosv@LAP513 ~ % where python3
/opt/homebrew/bin/python3
/usr/bin/python3
```

Python Installation

- PyEnv in MacOS

```
# install python version 3.12.0
pyenv install 3.12.0

# list available python versions
pyenv versions
```

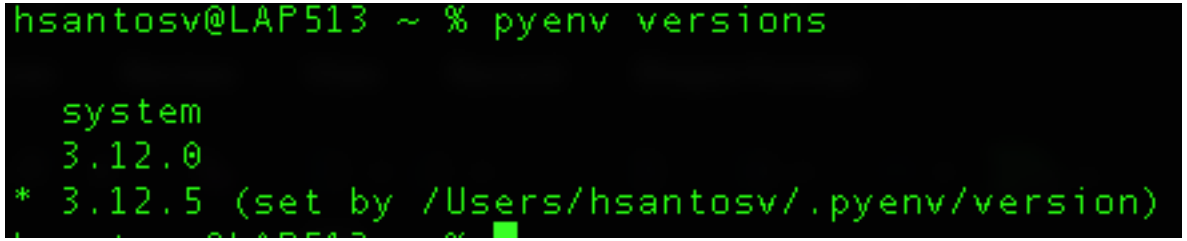


```
hsantosv@LAP513 ~ % pyenv versions

* system (set by /Users/hsantosv/.pyenv/version)
  3.12.0
```

Python Installation

- PyEnv in MacOS

```
# install separate python version 3.12.5
pyenv install 3.12.5

# set default version to newly installed 3.12.5
pyenv global 3.12.5

# list available python versions
pyenv versions
```



```
hsantosv@LAP513 ~ % pyenv versions

  system
  3.12.0
* 3.12.5 (set by /Users/hsantosv/.pyenv/version)
```

Python Environment

- Good resource: https://realpython.com/python-virtual-environments-a-primer/
- Manage the libraries, python version, etc. for your project
- You can use conda, venv, or ***pyenv-vritualenv***

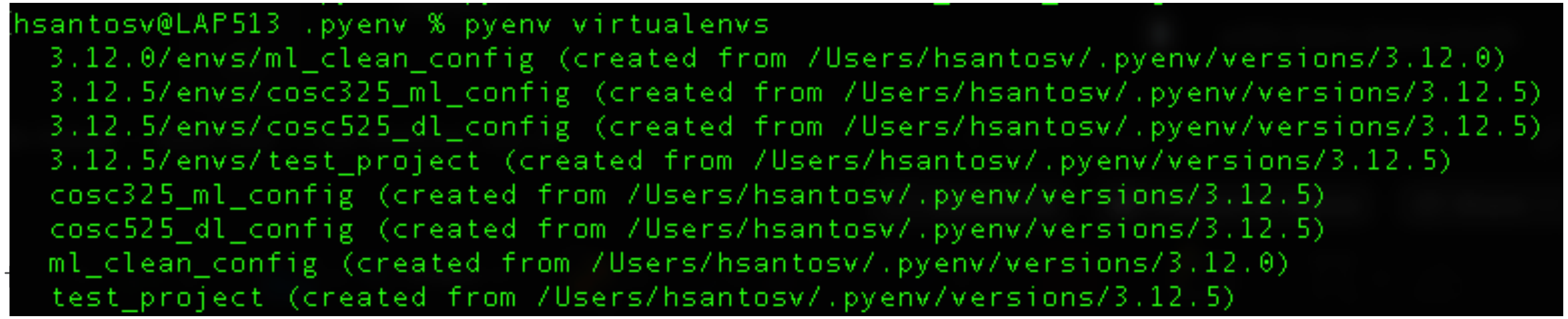
```
# install pyenv-virtualenv plugin
brew install pyenv-virtualenv
```

Python Environment

```
# create new virtualenv
# (e.g. pyenv virtualenv <python-version> <env-name>)
pyenv virtualenv 3.12.5 test_project

# activate the virtualenv
pyenv activate test_project

# deactivate the virtualenv
pyenv deactivate test_project

# list all available virtual environments
pyenv virtualenvs
```

Python Environment

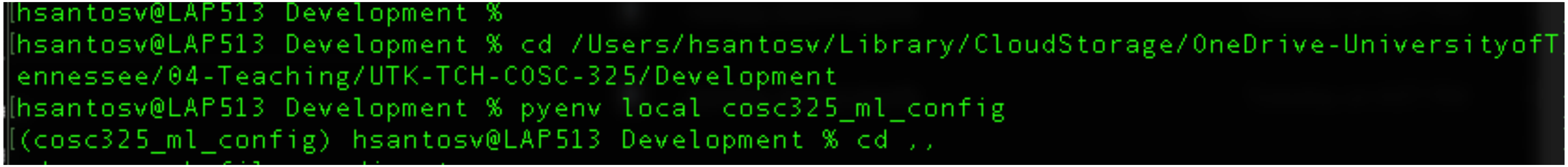
```
# create new virtualenv
# (e.g. pyenv virtualenv <python-version> <env-name>)
pyenv virtualenv 3.12.5 test_project

# activate the virtualenv
pyenv activate test_project

# deactivate the virtualenv
pyenv deactivate test_projec

# list all available virtual environments
pyenv virtualenvs
```



```
hsantosv@LAP513 .pyenv % pyenv virtualenvs
  3.12.0/envs/ml_clean_config (created from /Users/hsantosv/.pyenv/versions/3.12.0)
  3.12.5/envs/cosc325_ml_config (created from /Users/hsantosv/.pyenv/versions/3.12.5)
  3.12.5/envs/cosc525_dl_config (created from /Users/hsantosv/.pyenv/versions/3.12.5)
  3.12.5/envs/test_project (created from /Users/hsantosv/.pyenv/versions/3.12.5)
  cosc325_ml_config (created from /Users/hsantosv/.pyenv/versions/3.12.5)
  cosc525_dl_config (created from /Users/hsantosv/.pyenv/versions/3.12.5)
  ml_clean_config (created from /Users/hsantosv/.pyenv/versions/3.12.0)
  test_project (created from /Users/hsantosv/.pyenv/versions/3.12.5)
```

Local Python Environment

```
# Set a directorate environment to specific Python Environment
cd local_directory
pyenv local cosc325_ml_config
```



```
[hsantosv@LAP513 Development %
[hsantosv@LAP513 Development % cd /Users/hsantosv/Library/CloudStorage/OneDrive-UniversityofT
ennessee/04-Teaching/UTK-TCH-COSC-325/Development
[hsantosv@LAP513 Development % pyenv local cosc325_ml_config
[(cosc325_ml_config) hsantosv@LAP513 Development % cd ..
```

PIP COSC325 Libraries

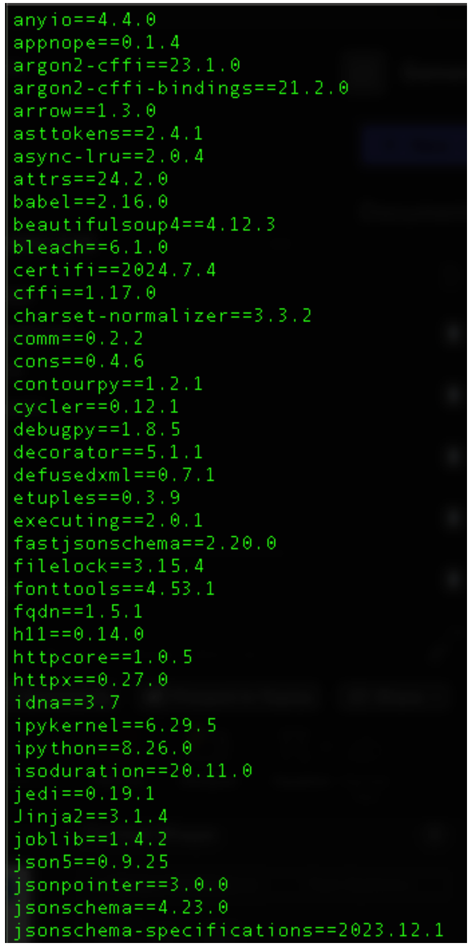
```
# install jupyterlab into your virtual environment
pip install jupyterlab

# install Numpy for, Matlab-like, matrix multiplication capabilities
pip install numpy
pip install scipy     #optional: expands numpy functions

# install Pandas for manipulation of tabulated data and seaborn for pandas-supported plots
pip install pandas
pip install seaborn

# install Scikit-Learn for most classical machine learning needs
pip install scikit-learn

# install matplotlib for general generation of graphs/plots
pip install matplotlib

# install PyTensor (formaly Theano) to optimize matrix computations (e.g., use of GPUs)
pip install pytensor
```

Saving/Exporting Your Environment

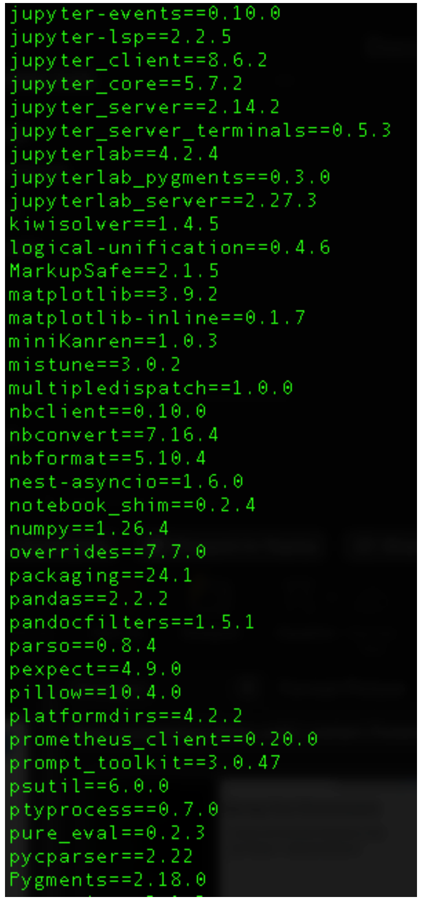
```
# save environment libraries to file
pip freeze > requirements.txt
```

```
# EXPORT
# Activate environment
pyenv activate <env_name>

# import environment libraries from file
(<env_name>)$ pip install -r path/to/requirements.txt
```

Note that it does not save the Python version.



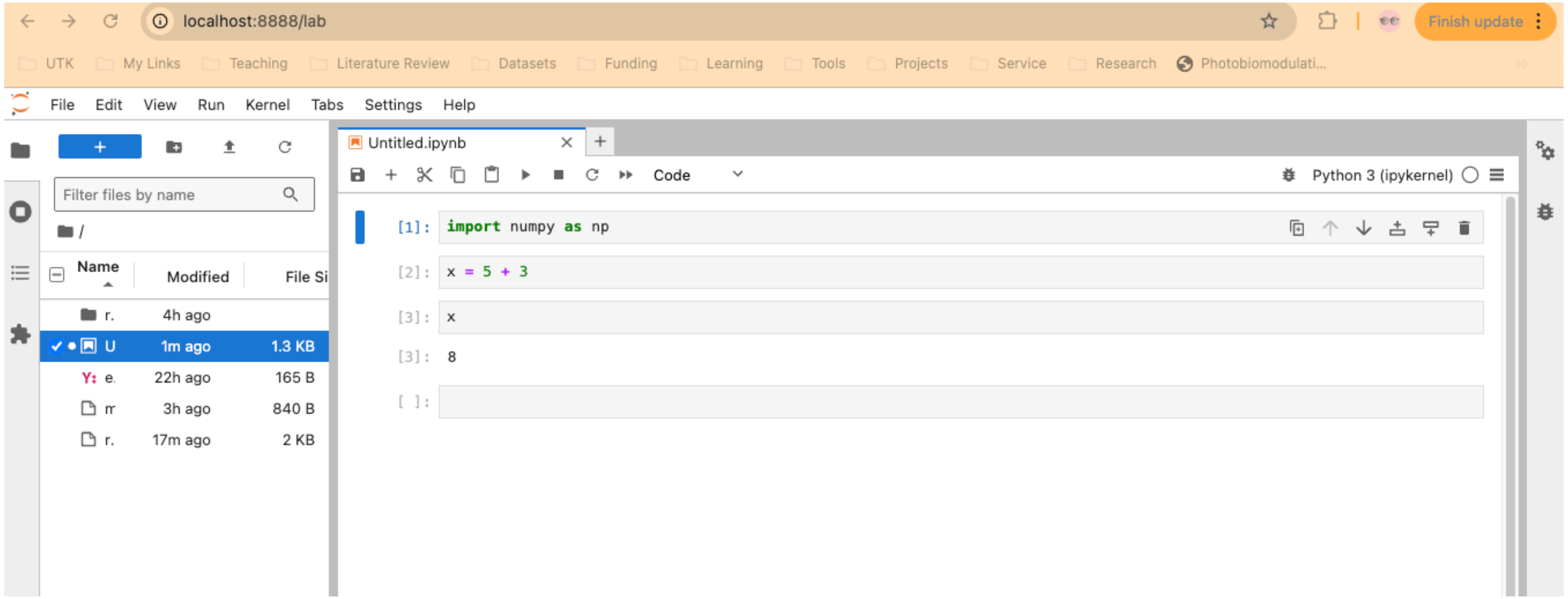
```
anyio==4.4.0
appnope==0.1.4
argon2-cffi==23.1.0
argon2-cffi-bindings==21.2.0
arrow==1.3.0
asttokens==2.4.1
async-lru==2.0.4
attrs==24.2.0
babel==2.16.0
beautifulsoup4==4.12.3
bleach==6.1.0
certifi==2024.7.4
cffi==1.17.0
charset-normalizer==3.3.2
comm==0.2.2
cons==0.4.6
contourpy==1.2.1
cycler==0.12.1
debugpy==1.8.5
decorator==5.1.1
defusedxml==0.7.1
etuples==0.3.9
executing==2.0.1
fastjsonschema==2.20.0
filelock==3.15.4
fonttools==4.53.1
fqdn==1.5.1
h11==0.14.0
httpcore==1.0.5
httpx==0.27.0
idna==3.7
ipykernel==6.29.5
ipython==8.26.0
isoduration==20.11.0
jedi==0.19.1
Jinja2==3.1.4
joblib==1.4.2
json5==0.9.25
jsonpointer==3.0.0
jsonschema==4.23.0
jsonschema-specifications==2023.12.1
```



```
jupyter-events==0.10.0
jupyter-lsp==2.2.5
jupyter_client==8.6.2
jupyter_core==5.7.2
jupyter_server==2.14.2
jupyter_server_terminals==0.5.3
jupyterlab==4.2.4
jupyterlab_pygments==0.3.0
jupyterlab_server==2.27.3
kiwisolver==1.4.5
logical-unification==0.4.6
MarkupSafe==2.1.5
matplotlib==3.9.2
matplotlib-inline==0.1.7
miniKanren==1.0.3
mistune==3.0.2
multipledispatch==1.0.0
nbclient==0.10.0
nbconvert==7.16.4
nbformat==5.10.4
nest-asyncio==1.6.0
notebook_shim==0.2.4
numpy==1.26.4
overrides==7.7.0
packaging==24.1
pandas==2.2.2
pandocfilters==1.5.1
parso==0.8.4
pexpect==4.9.0
pillow==10.4.0
platformdirs==4.2.2
prometheus_client==0.20.0
prompt_toolkit==3.0.47
psutil==6.0.0
ptyprocess==0.7.0
pure_eval==0.2.3
pycparser==2.22
Pygments==2.18.0
```

Jupyter Notebook

```
# Open jupyter server
jupyter lab
```



Name	Modified	File Si
r.	4h ago	
U	1m ago	1.3 KB
e.	22h ago	165 B
rr	3h ago	840 B
r.	17m ago	2 KB

```
[1]: import numpy as np
[2]: x = 5 + 3
[3]: x
[3]: 8
[ ]:
```

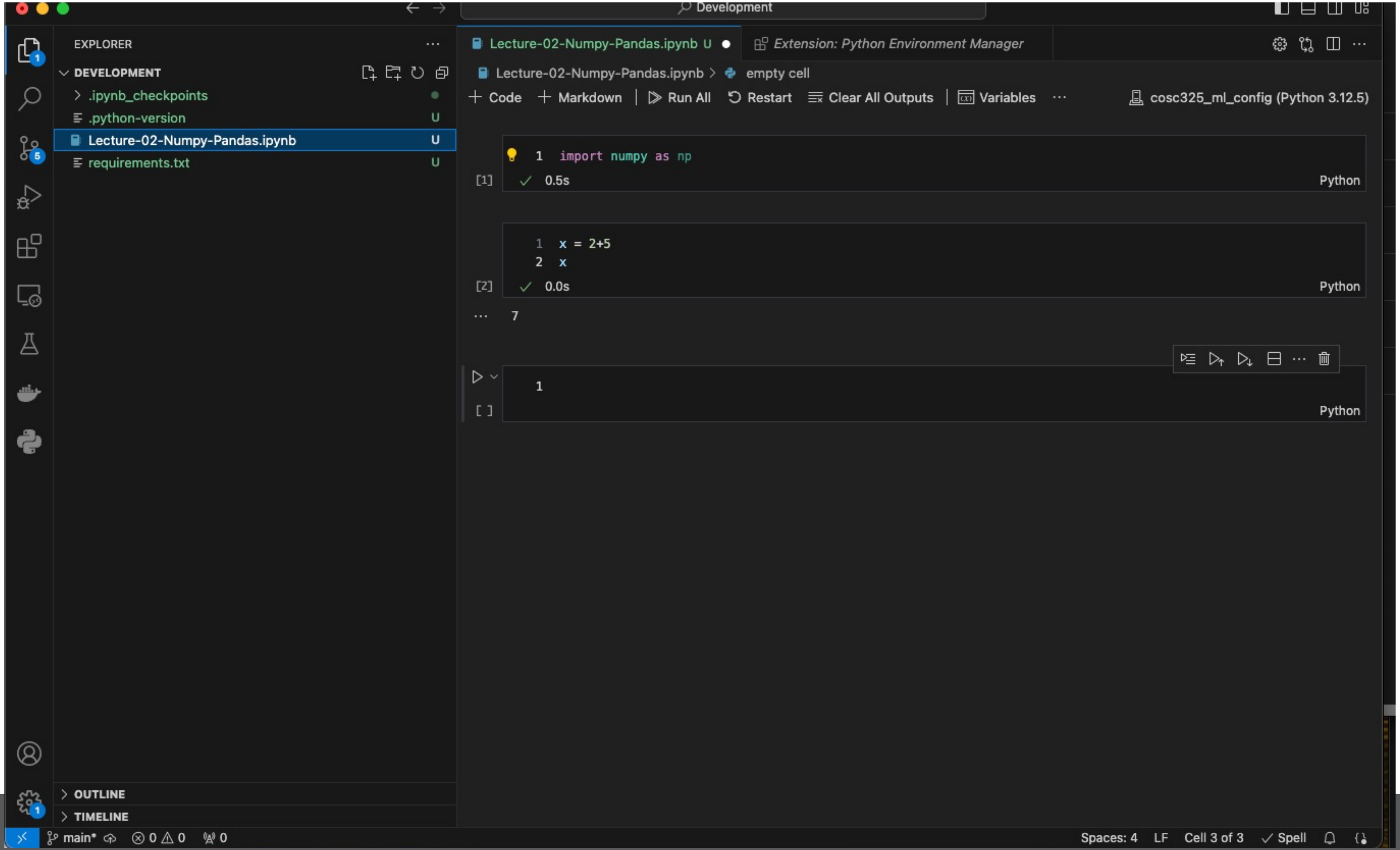
Jupyter Notebooks

- **Interactive Development:** Jupyter Notebooks allow real-time code execution and visualization, enabling immediate feedback and iterative development, which is ideal for data exploration and analysis.
- **Rich Media Support:** Integrates code, text, images, and visualizations in a single document, enhancing documentation and presentation of data analysis and research findings.
- **Collaborative Features:** Facilitates collaboration by enabling easy sharing and version control of notebooks, allowing multiple users to work on the same document seamlessly.
- **Wide Language Support:** Supports over 40 programming languages, including Python, R, and Julia, making it versatile for various data science and computational tasks.

```
pip install jupyterlab
```

https://ttpoll.com/p/569627

VSCode

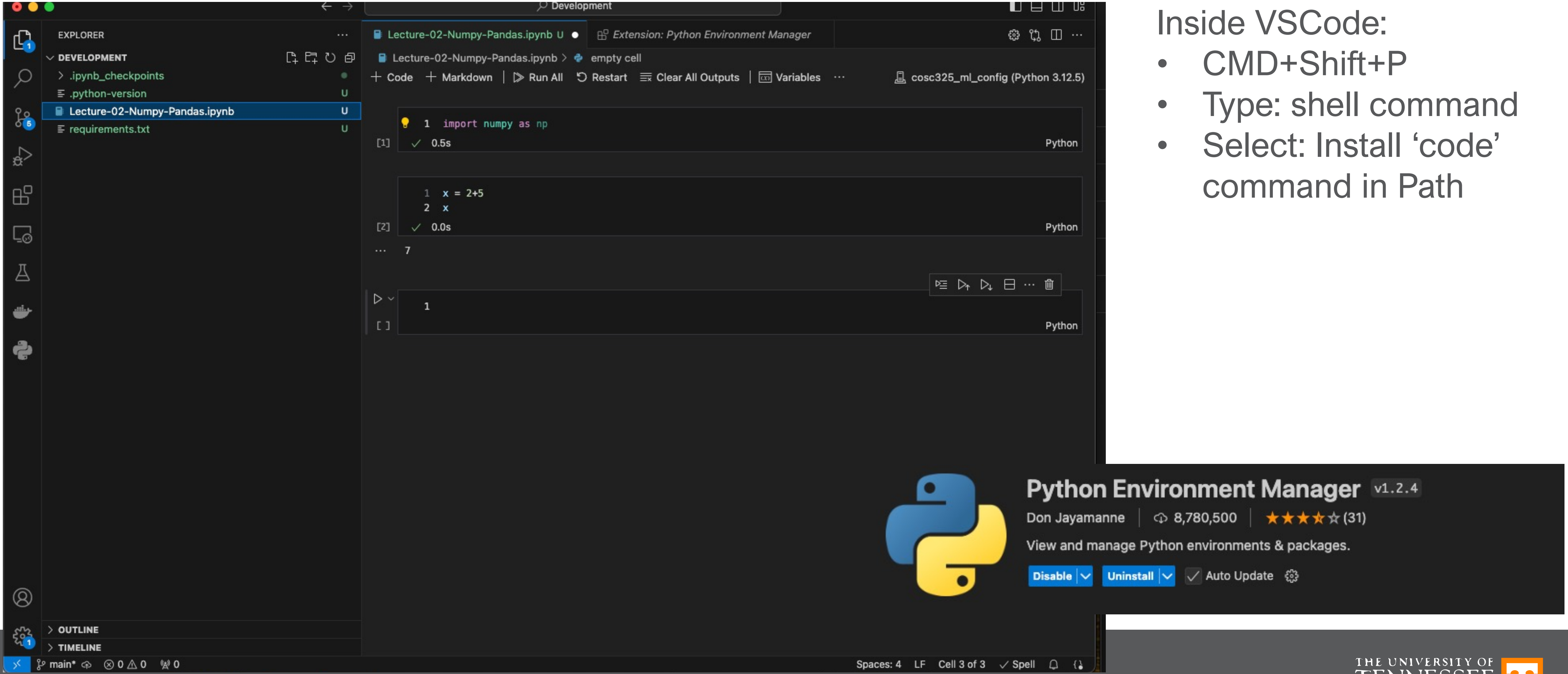


```
[1] import numpy as np
✓ 0.5s
```

```
[2] x = 2+5
    x
✓ 0.0s

7
```

```
[ ]
```

VSCode



Lecture-02-Numpy-Pandas.ipynb > empty cell

cosc325_ml_config (Python 3.12.5)

```
[1]  import numpy as np
     ✓ 0.5s
```

```
[2]  x = 2+5
     x
     ✓ 0.0s

     7
```

Python Environment Manager v1.2.4

Don Jayamanne | 8,780,500 | ★★★★☆ (31)

View and manage Python environments & packages.

Inside VSCode:

- CMD+Shift+P
- Type: shell command
- Select: Install ‘code’ command in Path

Other IDEs

- PyCharm [Local]
  - https://www.jetbrains.com/pycharm/
- Google Colab
  - https://colab.research.google.com/
- Anaconda Notebooks
  - https://anaconda.cloud/code-in-the-cloud
- Kaggle
  - https://www.kaggle.com/code
- JupyterLab
  - https://jupyter.org/try-jupyter/lab/

Python Basics

- General and lambda functions
  - https://www.geeksforgeeks.org/python-functions/
- Classes
  - https://www.geeksforgeeks.org/python-classes-and-objects/
- Loops and how to iterate over lists
  - https://www.geeksforgeeks.org/loops-in-python/
- Data types, such as Lists, Tuples, Dictionaries
  - https://www.geeksforgeeks.org/python-lists/
  - https://www.geeksforgeeks.org/python-tuples/
  - https://www.geeksforgeeks.org/python-dictionary/

Python Basics

- General and lambda functions
  - https://www.geeksforgeeks.org/python-functions/
- Classes
  - https://www.geeksforgeeks.org/python-classes-and-objects/
- Loops and how to iterate over lists
  - https://www.geeksforgeeks.org/loops-in-python/
- Data types, such as Lists, Tuples, Dictionaries
  - https://www.geeksforgeeks.org/python-lists/
  - https://www.geeksforgeeks.org/python-tuples/
  - https://www.geeksforgeeks.org/python-dictionary/

https://www.geeksforgeeks.org/python-difference-between-list-and-tuple/

Lecture Recap

- Machine learning
  - A subfield of artificial intelligence
  - Models need to generalize (i.e., learn)
    - Task, Experience, Performance
  - Different learning categories: supervised, self-supervised, and reinforcement.
- Programming
  - Python: flexible, efficient, collaborative, powerful
  - Always work from a dockerized or virtual environment
  - Practice

https://ttpoll.com/p/569627

Next Week

- Scientific computing with Python
  - Deep dive for Numpy, Pandas, and Scikit-Learn libraries
- Bring your laptops with Python environment ready.
  - Check tutorial in Canvas