Statistics and Data Science 265

# Introductory Machine Learning

January 14

Yale

Version of 2026/01/14 at 12:53

# Outline

- Administrative items
- Elements of Python
- Demos

# Office Hours

| | | |
|---|---|---|
| John Lafferty | Monday | 10:30am |
| Jianliang He | Monday | 2:30pm |
| William Huang | Monday | 4:00pm |
| Ruiqi Li | Tuesday | 6:00pm |
| Awni Altabaa | Wednesday | 3:00pm |
| Yihan Bao | Wednesday | 6:00pm |
| Dingyao Zhang | Thursday | 8:00pm |

# Recordings

- Lectures will be recorded
- Available on Canvas under “Media”
- Please attend class and only use recordings for review

# Upcoming

- Assignment 1 will be posted next week
- Topics: Classification and regression

# Bookmark this page!

```
https://ydata123.org/sp26/introml/calendar.html
```

# Recap

- Last time: Course overview, logistics
- Any questions?

# Plan for Today

- Python elements
- Pandas and linear regression example
- Basics of classification, regression, overfitting

# Python primer: Concepts

- Python types: lists, tuples, strings, dictionaries
- Basics of iteration
- Comprehensions
- Arithmetic
- Printing
- NumPy and multi-dimensional arrays
- Array math
- DataFrames and pandas
- Matplotlib and basic plotting

# Python elements

+ Code + Text

## Python and Jupyter essentials for iML

This notebook was adapted from multiple resources including the Data8 curriculum, Yale EENG201, and Stanford CS231. It is intended to give you a quick "jumpstart" and introduction to the tools that we will use throughout the course, based on Python, Jupyter notebooks, and essential useful packages like `numpy` and `pandas`.

It's important to recognize that practice is crucial here---you need to write code and implement things, making mistakes along the way, to gain proficiency in this material.

Subtopics marked with the scream icon are a little more advanced, and can be skipped on a first reading.

## Get Started

### Different ways to run Python

1. Create a file using editor, then: `$ python myscript.py`
2. Run interpreter interactively `$ python`
3. Use a Python environment, e.g. Anaconda or Google Colab

We recommend Anaconda:

- easy to install
- easy to add additional packages
- allows creation of custom environments

But Google Colab is also a good option. We plan to create a video on how to use Google Colab.

# Resources

- Anaconda Python: https://www.continuum.io
- Jupyter notebooks: jupyter-notebook.readthedocs.io
- PyCharm debugger: www.jetbrains.com
- *Introducing Python*, Bill Lubanovic, O'Reilly
- *Python in a Nutshell*, Alex Martelli et al., O'Reilly
- *Python Cookbook*, David Beazley, Brian K. Jones, O'Reilly
- Google's Python class:
  https://www.youtube.com/watch?v=tKTZoB2Vjukxo
- https://docs.python.org/3.5/tutorial
- *Lots* of other materials available on the web

# Pandas example

+ Code + Text

## The New York Times Covid-19 Database

The New York Times Covid-19 Database is a county-level database of confirmed cases and deaths, compiled from state and local governments and health departments across the United States. The initial release of the database was on Thursday, March 26, 2020, and it is updated daily.

These data have fueled many articles and graphics by The Times; these are updated regularly at https://www.nytimes.com/interactive/2020/us/coronavirus-us-cases.html. The Times has created many visualizations that are effective communications of important information about the pandemic.

The data are publically available via GitHub: https://github.com/nytimes/covid-19-data. In this illustration we will only use the data aggregated at the state level.

```
[ ] import pandas as pd
    import numpy as np

    %matplotlib inline
    import matplotlib.pyplot as plt
    plt.style.use('fivethirtyeight')
```

```
[ ] covid_table = pd.read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-states.csv")
    covid_table = covid_table.drop('fips', axis=1)
    covid_table.tail(20)
```

| | date | state | cases | deaths |
|---|---|---|---|---|
| **30464** | 2021-09-07 | North Dakota | 119995 | 1596 |
| **30465** | 2021-09-07 | Northern Mariana Islands | 248 | 2 |
| **30466** | 2021-09-07 | Ohio | 1262018 | 21020 |
| **30467** | 2021-09-07 | Oklahoma | 570923 | 8001 |
| **30468** | 2021-09-07 | Oregon | [illegible] | [illegible] |

# Some Terminology

- supervised vs. unsupervised
- classification vs. regression
- prediction vs. inference

# Supervised Learning vs. Unsupervised Learning

Supervised learning:

- Given a set of $(x, y)$, learn to predict $y$ using $x$.
- e.g.
  - Predicting whether a loan will default based on customer characteristics

# Supervised Learning vs. Unsupervised Learning

Supervised learning:

- Given a set of $(x, y)$, learn to predict $y$ using $x$.
- e.g.
  - Predicting whether a loan will default based on customer characteristics

Unsupervised learning:

- Given a set of $x$, learn underlying structure or relationships of $x$.
- e.g.
  - Identifying market segments with similar spending patterns.

# Classification vs. Regression

The `Income` dataset:

| Education | Seniority | Income |
|---|---|---|
| 21.58621 | 113.1034 | 99.91717 |
| 18.27586 | 119.3103 | 92.57913 |
| 12.06897 | 100.6897 | 34.67873 |
| 17.03448 | 187.5862 | 78.70281 |
| 19.93103 | 20.0000 | 68.00992 |
| 18.27586 | 26.2069 | 71.50449 |

Information for 30 *simulated individuals*.

# Classification vs. Regression

The `Income` dataset:

| Education | Seniority | Income |
|---|---|---|
| 21.58621 | 113.1034 | 99.91717 |
| 18.27586 | 119.3103 | 92.57913 |
| 12.06897 | 100.6897 | 34.67873 |
| 17.03448 | 187.5862 | 78.70281 |
| 19.93103 | 20.0000 | 68.00992 |
| 18.27586 | 26.2069 | 71.50449 |

Information for 30 *simulated individuals*.

Regression: Model income based on other characteristics.

# Classification vs. Regression

The `Income` dataset:

| Education | Seniority | Income |
|---|---|---|
| 21.58621 | 113.1034 | 99.91717 |
| 18.27586 | 119.3103 | 92.57913 |
| 12.06897 | 100.6897 | 34.67873 |
| 17.03448 | 187.5862 | 78.70281 |
| 19.93103 | 20.0000 | 68.00992 |
| 18.27586 | 26.2069 | 71.50449 |

Information for 30 *simulated individuals*.

Regression: Model income based on other characteristics.

Classification: Model whether someone will earn above the median income based on other characteristics.

# Inference vs. Prediction

The `Income` dataset:

| Education | Seniority | Income |
|---|---|---|
| 21.58621 | 113.1034 | 99.91717 |
| 18.27586 | 119.3103 | 92.57913 |
| 12.06897 | 100.6897 | 34.67873 |
| 17.03448 | 187.5862 | 78.70281 |
| 19.93103 | 20.0000 | 68.00992 |
| 18.27586 | 26.2069 | 71.50449 |

Information for 30 *simulated individuals*.

Prediction: accurately predict $Y$ for new observations

# Inference vs. Prediction

The `Income` dataset:

| Education | Seniority | Income |
|---|---|---|
| 21.58621 | 113.1034 | 99.91717 |
| 18.27586 | 119.3103 | 92.57913 |
| 12.06897 | 100.6897 | 34.67873 |
| 17.03448 | 187.5862 | 78.70281 |
| 19.93103 | 20.0000 | 68.00992 |
| 18.27586 | 26.2069 | 71.50449 |

Information for 30 *simulated individuals*.

Prediction: accurately predict $Y$ for new observations

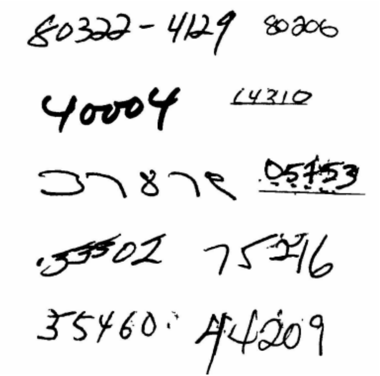
Inference: explain the underlying relationship between $Y$ and $X$

# Example: Handwritten Digit Recognition

- Data: images of handwritten digits (grayscale pixel values)
- Classify images as digits 0 to 9.

# Example: Handwritten Digit Recognition

- Data: images of handwritten digits (grayscale pixel values)
- Classify images as digits 0 to 9.



# Summary

- Two cultures: model based and prediction based
- Python, pandas, and linear regression example with Covid-19 data

Next week: Linear regression and classification

Class on Wed and Fri next week (MLK)