S1: Overview of Machine Learning, Recap on Descriptive Statistics, Data Science in Python

Phanuphat Srisukhawasu (Oad)

Machine Learning Department
NUS Fintech Society

Week 3 (26 Aug 2024 - 30 Aug 2024)

Session Outline

But what is machine learning?

Definition in modern data science

- **General definition**:
  The field of study that give computers the ability to learn without being explicitly programmed.

  — Arthur Samuel, 1959

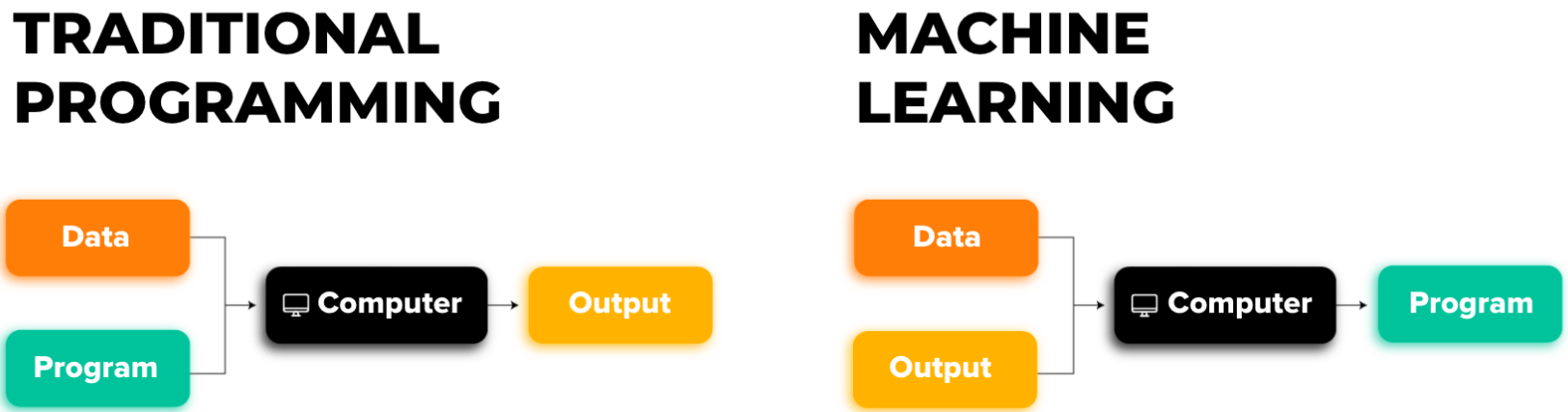
- **Engineering-oriented definition**:
  A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with E.

  — Tom Mitchell, 1997

Are we overcomplicating things?

Let's take a look at this diagram

Traditional Programming vs. Machine Learning



- **Traditional programming** is used when we know the specific *rules and logic* needed to write a program.
- In **machine learning**, we have the *inputs and outputs*, but the computer figures out the rules on its own.

We know the rules for most problems

Write a program that classifies numbers into odd and even

- Recall from math class, a number n is said to be even if n is divisible by 2 without a remainder.
- A number n is said to be odd if n is an integer and n is not an even number.

```
def is_even(n: int):
    if n % 2 == 0:
        return True
    return False
```

Note: *The else statement is unnecessary since the code executes only if n is not divisible by 2.*

There are some problems that have no exact rules

Write a program that detects spam emails (junk folder)

- Spam emails often contain specific phrases or keywords such as **"you're so close to FREE snacks!"**.
- In reality, there are millions of possible configurations, making it impractical for humans to manually account for all.
- This illustrates one of the broad applications of Machine Learning (ML). With **sufficient data**, it is possible to develop a program that effectively detects spam emails.
- During the training program, you will work with various ML models that handle different types of data.
- **Key Point:** ML improves with data. Higher quality data leads to better performance of the program.

Supervised Learning

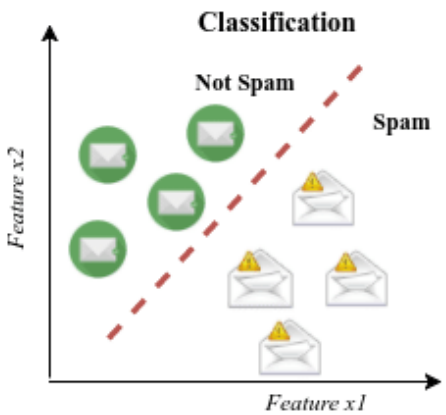
Models that learn on labeled data

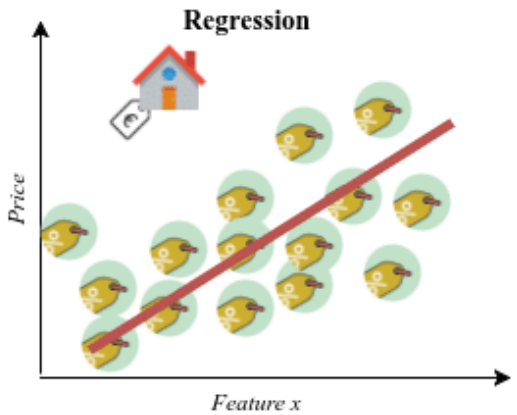
- Supervised learning algorithms are given labeled data to learn the *relationship between the inputs and outputs*.
- **Regression** is one of the common supervised problems where the models learn to map the inputs to the numeric outputs.
  - The simplest regression model is Linear Regression ($y = a_0 + a_1 x,$) which is used for modeling linear data.
  - E.g. house price prediction, stock price prediction, etc.
- Another common task is **Classification**. This is when the model learn to map the inputs to the categorical outputs.
  - Tasks likes spam/not spam email classification are called **Binary Classification** (two possible outputs.)
  - Tasks that involve more than two categories like handwritten character classification are called **Multiclass Classification**.

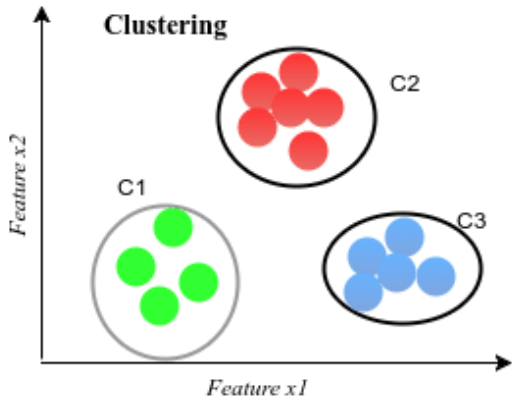
Unsupervsied Learning

Models that learn on unlabeled data

- Unsupervsied learning algorithms are given unlabeled data to learn the *underlying patterns between the inputs.* A common task is the models that group similar data together, we refer to the task as **Clustering**.



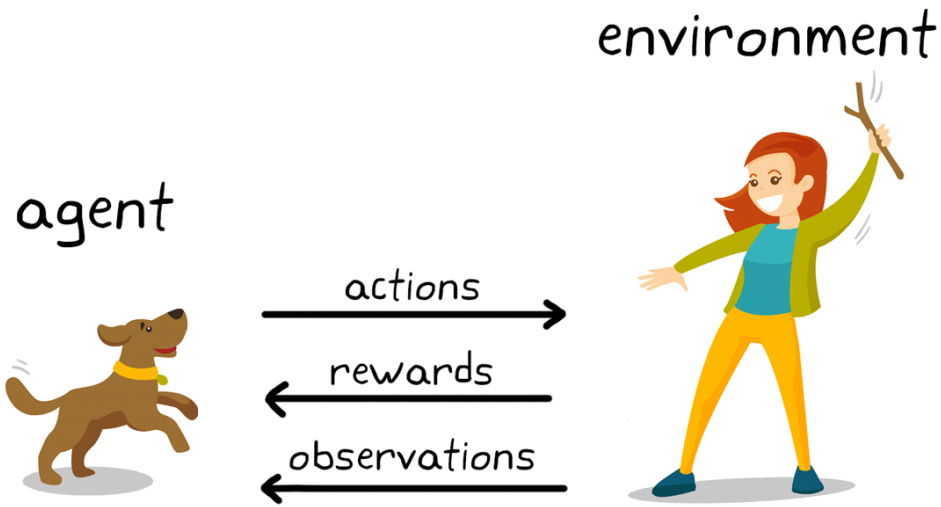




Reinforcement Learning

Models that learn to play games!

- In reinforcement learning, an agent observe the environment to perform an action A at a state S, achieving a reward R.
- The agent will learn to perform an action in each state such that it maximimizes the rewards.



Mean, Median, Mode

- When describing some data, we have two main things we want to look at: central tendency and variabilitiy.
- **Central Tendency** is the statistical measurement to identify the *single value that represents an entire data.*
  - **Mean** $\overline{X} = \sum X/N$ (N is the number of data points.)
  - **Median** is the instance which is located at the *middle of the data when is sorted in ascending order.*
  - **Mode** is the instance that appears the most frequent in the data (usually used with *categorical data.*)
- Note: *Median is usually a better representative than Mean when the dataset is skewed or not symmetrical.*

Standard Deviation and Variance

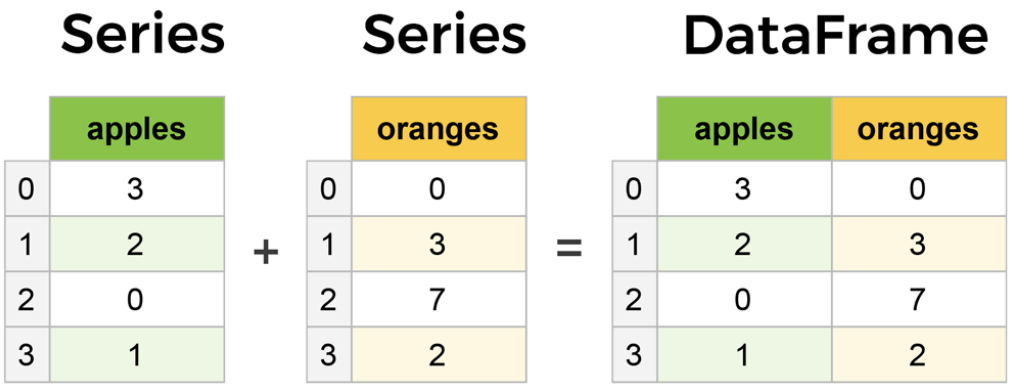
- **Variabilitiy** is the statistical measurement to tell *how far apart data points lie from each other and the center.*
  - **Standard Deviation** ($S.D.$) measures how far apart each data point is from the Mean in average.

$$S.D. = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{X})^2}{N}}$$

  - **Variance** is the squared of S.D.
- There are also some other common metrics to measure the variability such as **Range** (the differences between the maximum and minimum values).

Data structures in Pandas

- **Pandas** is a Python that library used for working effectively with data sets.
- There are 2 main types of structures in Pandas, namely **Series** and **Dataframe**.



- Note: The numbers on the left of the first column are called **indices**. Like most programming languages, they start from zero all the way to $N - 1$ for a sequence with length of N.

Loading Data with Pandas

- If you use Google Colab, Pandas is already installed. You can import it to your notebook using the following code.

```
import pandas as pd

# simply install with `!pip install pandas` if you haven't
```

- If you store data as CSV file, use `pd.read_csv()`.
- If you store data as Excel file, use `pd.read_excel()`.
- The loaded dataset is stored as a dataframe. We usually assigned it to `df`.
- For example, `df = pd.read_csv("data/iris.csv")`. The string inside the bracket is the relative path to the file.

Inspecting Data with Pandas (1)

- `df.head()` will give you the first 5 rows of the dataframe.

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa

- `df.info()` will give you some useful information about the dataframe like the data types of each column.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal.length  150 non-null    float64
 1   sepal.width   150 non-null    float64
 2   petal.length  150 non-null    float64
 3   petal.width   150 non-null    float64
 4   variety       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Inspecting Data with Pandas (2)

- Note: `df.info()` also shows the numbers of non-null instances in each column. For now, you can refer to the null instances as the missing values (empty cells).
- Finally, you can get the some descriptive statistics values using `df.describe()`.

	sepal.length	sepal.width	petal.length	petal.width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Filter and Slice Data

- You can use `df.iloc[rows, cols]` to select portion of dataframe by indices. Also, you can use `df.loc[rows, cols]` to select by labels.
- Examples:
  - `df.iloc[1]` will select a row with index `1`.
  - `df.loc[:, "sepal.length"]` will select all rows from the column `sepal.length`.
  - `df.loc[:, ["sepal.length, sepal.width"]]` will select all rows from the column `sepal.length` and `sepal.width`.
  - `df.iloc[3:8, :]` will select all rows with the indices of `3, 4, 5, 6, 7` from all columns.
  - `df.loc[df["petal.width"] >= 1.3]` will select all rows and columns from the data which have the values in the column `petal.width` greater than or equal to 1.3.

Sort Data

- You can sort a dataframe based on values on one column in both ascending and descending order.

```
df_sorted = df.sort_values(by="petal.width", ascending=True)
```
✓ 0.0s Python

```
df_sorted.head()
```
✓ 0.0s Python

	sepal.length	sepal.width	petal.length	petal.width	variety
32	5.2	4.1	1.5	0.1	Setosa
13	4.3	3.0	1.1	0.1	Setosa
37	4.9	3.6	1.4	0.1	Setosa
9	4.9	3.1	1.5	0.1	Setosa
12	4.8	3.0	1.4	0.1	Setosa

- Note: *If you specify* `ascending=False`, *it will be sorted in the descending order. It is also a good practice to store the sorted data in a new dataframe to avoid overwriting*.

Group Data to Perform Some Calculation

- (Extra) You may notice that the column `variety` is a categorical data. To check all possible categories, simply use `df["variety"].unique()`. This will output `array(['Setosa', 'Versicolor', 'Virginica'], dtype=object)`.
- For example, one may want to find the mean length of sepal for the flower in each group:

```
df.groupby(by="variety")['sepal.length'].mean()
```

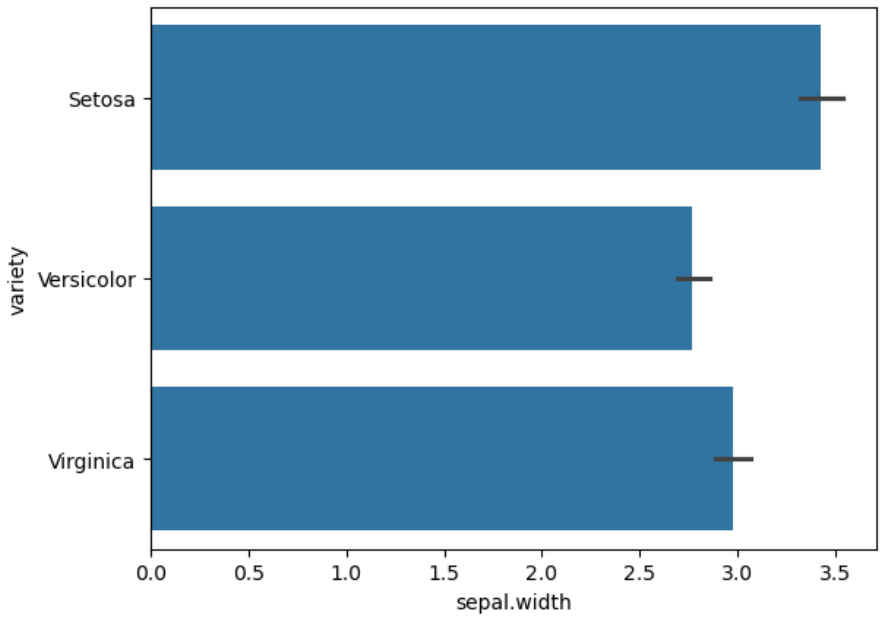
✓ 0.0s Python

```
variety
Setosa        5.006
Versicolor    5.936
Virginica     6.588
Name: sepal.length, dtype: float64
```

Basic Seaborn Plots

Bar plots

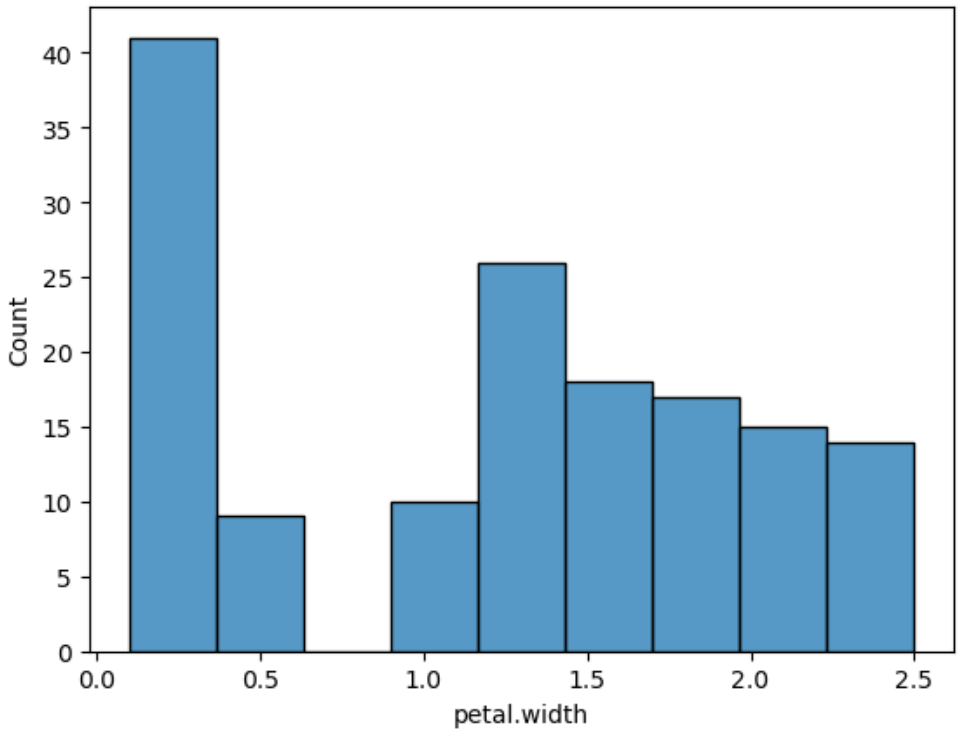
- Seaborn (`import seaborn as sns`) allows you to create complex plots with ease (though we will only cover the basics.)
- You can create bar plot like this: `sns.barplot(data=df, x="sepal.width", y="variety")`



Basic Seaborn Plots

Histograms

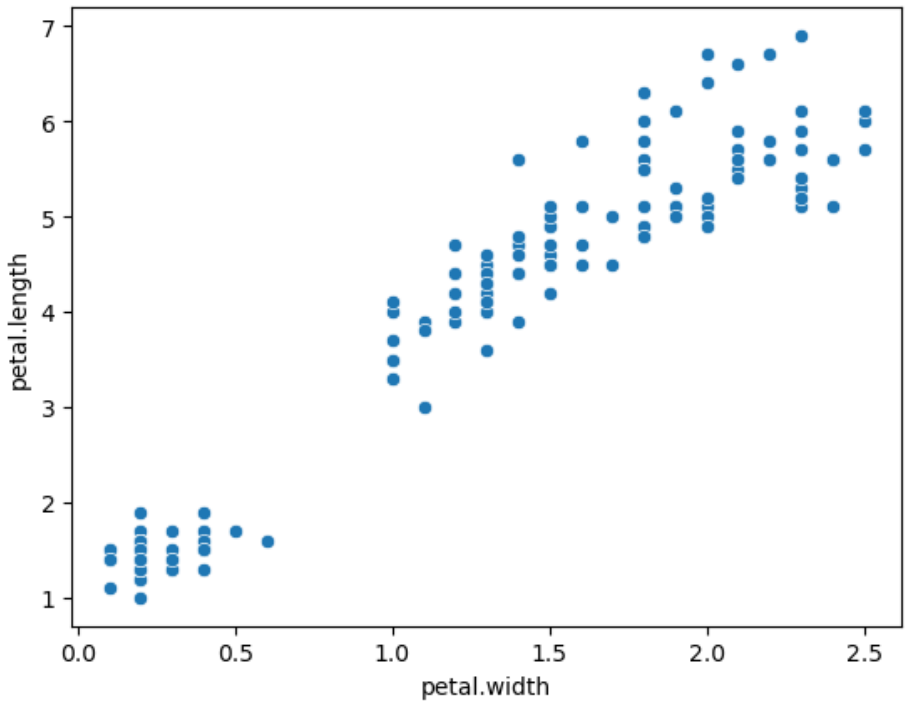
- You can create a histogram like this:
  `sns.histplot(data=df, x="petal.width")`



Basic Seaborn Plots

Scatter Plots

- You can create a scatter plot like this:
  `sns.scatterplot(data=df, x="petal.width", y="petal.length")`



Key Takeaways

- **Machine Learning**: Learns patterns from data, unlike traditional programming where rules are predefined. Key types: *supervised, unsupervised, and reinforcement learning.*
- **Statistics** are the backbone of data analysis, providing tools to describe, summarize, and infer from data.
- **Pandas** is essential for data manipulation, enabling tasks like *data cleaning, transformation, and analysis*.
- **Seaborn** helps create basic visualizations to explore data *patterns and relationships.*
- Just a reminder, we are assigning your first lab today. The due date is on 11:59 PM of the next week's training date.

Recommended Resources

- **Corey Schafer's Pandas Tutorials Playlist**: https://www.youtube.com/playlist?list=PL-osiE80TeTsWmV9i9c58mdDCSskIFdDS
- Take a look at the gallery (examples) for **Matplotlib** (another library for data visualization). This will give you an idea of different plots you could create: https://matplotlib.org/stable/gallery/index.html
- Great textbook for future reference throughout the training program: Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow
- **Weights & Biases' Math for Machine Learning Playlist**: (Here) Note: W&B is actually an AI development tool that allows you to *track experiments with different configurations.*