Python: scikit-learn

June 28, 2024

Download the materials from GitHub
https://github.com/nuitrcs/scikit-learn-workshop

We start at 9:35am

Coffee available!

Get help from Northwestern IT Research Computing and Data Services!

Get 1-1 help from us! – bit.ly/rcdsconsult

Bring Your Own Data (BYOD) working groups – bit.ly/byod_groups

Collaborative Project Support – bit.ly/collaborative_project_support

Take a minute to introduce yourself to your neighbors!

Northwestern IT
RESEARCH COMPUTING AND DATA SERVICES

Instructors:

Ritika Giri

efrén cruz cortés

John Lee

Sticky Notes

I'm done/
doing OK

I need help/
I'm stuck

Think before coding!

- You'll have the inclination to start coding immediately... but wait!
- Before coding, think about the problem and...
  - Define the task that you're trying to accomplish
    - Identify the input and the output
  - Break the task into subtasks
  - Order the subtasks logically

Help!

- Every programmer needs help. As you code, these will be your best friends:
  - Documentation
  - Stackoverflow.com
  - Github.com
  - Google
  - LLMs (Microsoft Bing/Copilot, ChatGPT, Google Gemini)
  - Your colleagues
  - Our free consult service: bit.ly/rcdsconsult

Let’s start !

Machine learning:

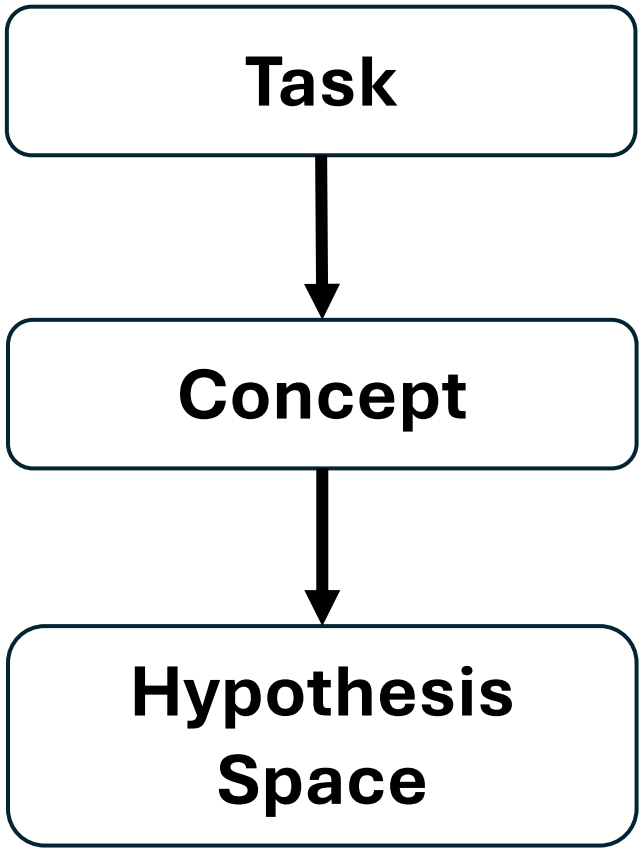
Learning from data..

without explicit programming

What do we want to learn with ML?

We will take the **task** approach:

- There is a task we want the machine to **do**.
- To perform this task, the machine must learn an underlying **concept**.
- We'll try to learn this concept by searching over a **space of hypotheses**.



Main Machine Learning Approaches

- Classification
- Regression
- Cluster Analysis

Classification

- **Goal**: Predict category given input data
  - Target is categorical variable

- **Examples**
- Classify tumor as benign or malignant
- Determine if credit card transaction is legitimate or fraudulent
- Identify customer as residential, commercial, public
- Predict if weather will be sunny, cloudy, windy, or rainy

Regression

- **Goal**: Predict numeric value given input data
  - Target is numeric variable

- **Examples**
- Predict the price of a stock
- Estimate demand for a product based on time of year
- Determine risk of loan application
- Predict amount of rain

Cluster Analysis

- **Goal**: Organize similar items into groups
  - There is no “target” variable

- **Examples**
- Group customer base into segments for effective targeted marketing
- Identify areas of similar topography (desert, grass, etc.)
- Categorize different types of tissues from medical images
- Discover crime hot spots

Supervised vs Unsupervised Learning

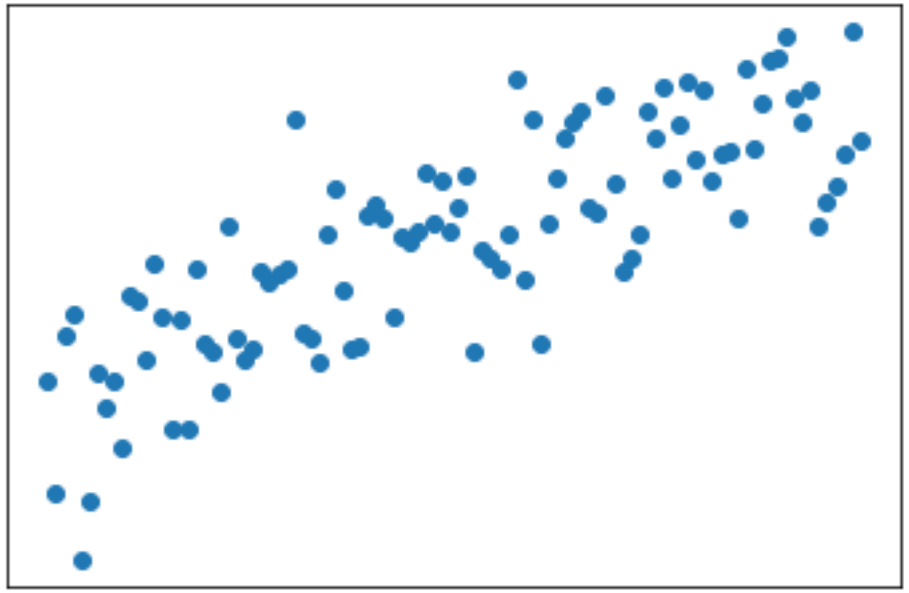
- Supervised Approaches
  - Target (what you're trying to predict) is provided
  - "Labeled" data
  - Classification and regression approached are supervised

- Unsupervised Approaches
  - Target is unknown or unavailable
  - "Unlabeled" data
  - Cluster analysis is unsupervised

Task Example 1: learning a linear relationship

Imagine we observe the variation of one variable with respect to another. We presume the relationship is linear:

Response, target, dependent variable, etc…

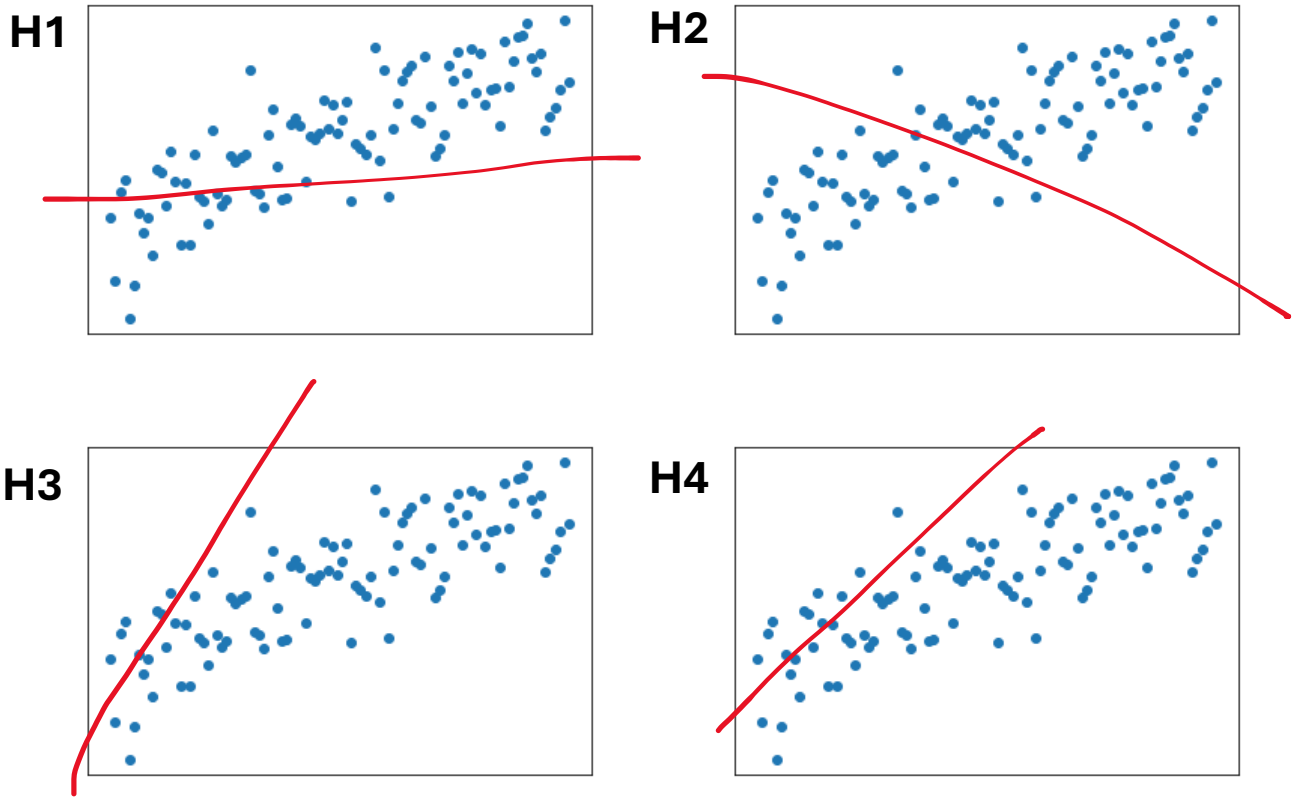


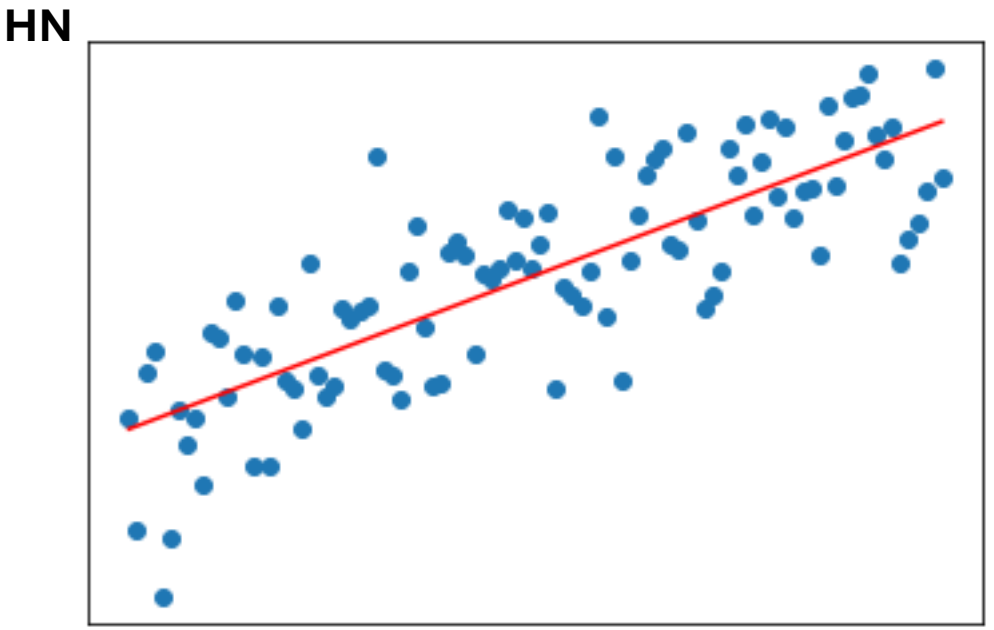
Predictive Observation

Learning a linear relationship

In this task, ML will aim to find the best-fit linear relationship i.e. best slope and intercept

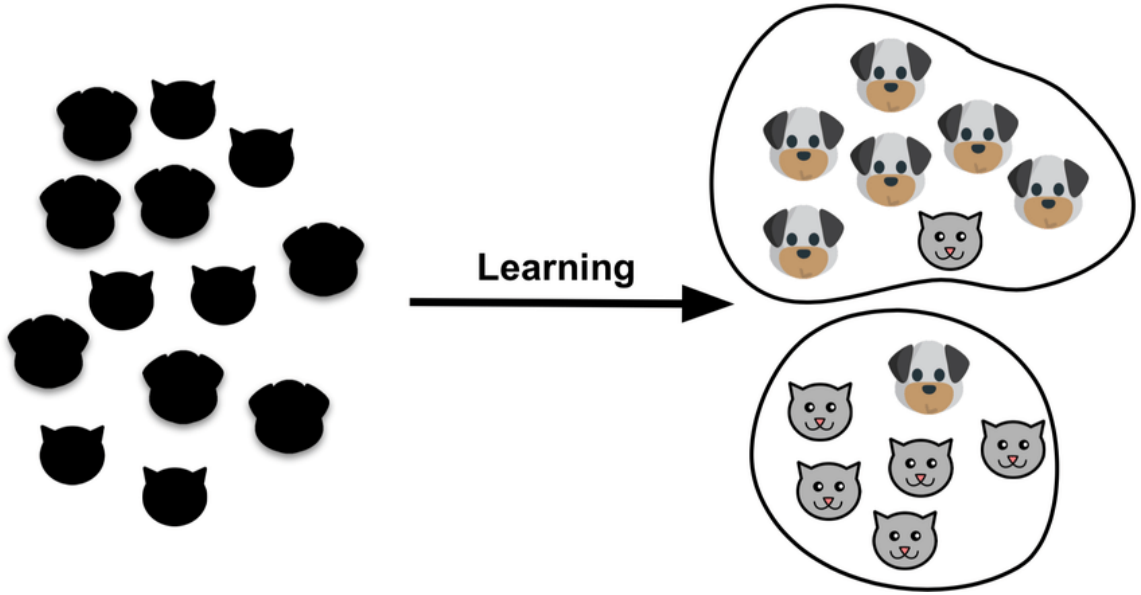
Our hypothesis space is the set of all straight lines:

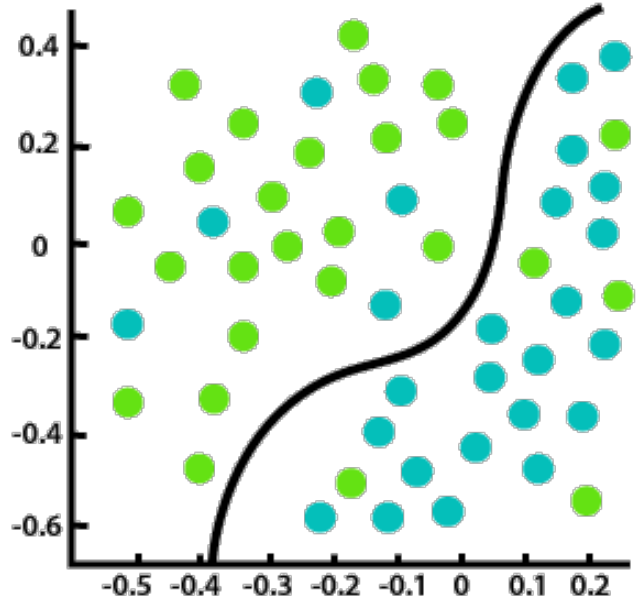




Task Example 2: classifying into categories

What makes a “dog” vs a “cat” ?





When Do We Use Machine Learning?

- **ML is used when:**
  - Human expertise does not exist (navigating on Mars)
  - Humans can’t explain their expertise (speech recognition)
  - Models must be customized (personalized medicine)
  - Models are based on huge amounts of data (genomics)

- **When you should not use ML:**
  - There is no need to “learn” to calculate payroll.
  - When it is unethical!

More examples of ML tasks

- **Recognizing patterns:**
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- **Generating patterns:**
  - Generating images or motion sequences
- **Recognizing anomalies:**
  - Unusual credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant
- **Prediction:**
  - Future stock prices or currency exchange rates

What is scikit-learn?

scikit-learn

- scikit-learn is a python library that helps you automate many machine learning tasks.
- It contains many modules, each of which has many models and useful utilities.
- It has modules for preprocessing data, for regression tasks, for classification, for obtaining metrics, etc.
- The “syntax”, meaning the way you write your code using scikit-learn, is heavily based on python’s **object oriented programming** capabilities.

The object oriented syntax

- You may recall from the python fundamentals bootcamp, that a *class* is a way in which we identify different types of entities in python.

- For example, we have strings, lists, numpy arrays, pandas dataframes, etc. Each of these is a *class*, a type of object.

- Each instance of a class, for example a specific pandas dataframe, has its own attributes (variables the object contains), as well as methods (things the object can do)

Example

- `df = pandas.DataFrame(some_data)`
  - Here, df is an **instance** of a pandas DataFrame class.

- `df.columns`
  - columns is an **attribute** (or property) of df, some piece of data it contains.

- `df.drop(columns = [“some column”])`
  - drop() is a **method** of df. This method can be performed by all pandas dataframes.

The triforce of objects' syntax

The three most important things to remember about objects:

Create an instance of some class:

```
x = SomeClass()
```

Obtain an attribute of an instance:

```
x.some_attribute
```

Perform a method:

```
x.some_method()
```

scikit-learn’s common steps

You will see that repeatedly we do the following in scikit-learn

Import some model from some module:

```
from sklearn.module import Model
```

Create instance of the model:

```
f = Model()
```

Fit that instance to some data:

```
f.fit(data)
```

Predict new responses based on previously unobserved data:

```
predicted_outcome = f.predict(some_other_data)
```

Compute the accuracy of your predictions:

```
compute_accuracy(predicted_outcome, true_outcome)
```

* module, Model, and compute_accuracy are not actual names in scikit-learn

Example

We will see this in mode detail in our jupyter notebook. But here's a preview:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X)
model.predict(new_X)
```

Quick Exercise

Order the following statements in the correct order:

```
linear_model.fit(x, y)

model_score = Accuracy (y_observed, y_predicted)

linear_model = LinearRegression()

y_predicted = linear_model.predict(x_new)

from sklearn.regression import LinearRegression
```

Solution

Order the following statements in the correct order:

```
from sklearn.regression import LinearRegression

linear_model = LinearRegression()

linear_model.fit(x, y)

y_predicted = linear_model.predict(x_new)

model_score = Accuracy (y_observed, y_predicted)
```