L4. Python

UCLA Masters of Applied Economics

Fall 2018

Melody Y. Huang

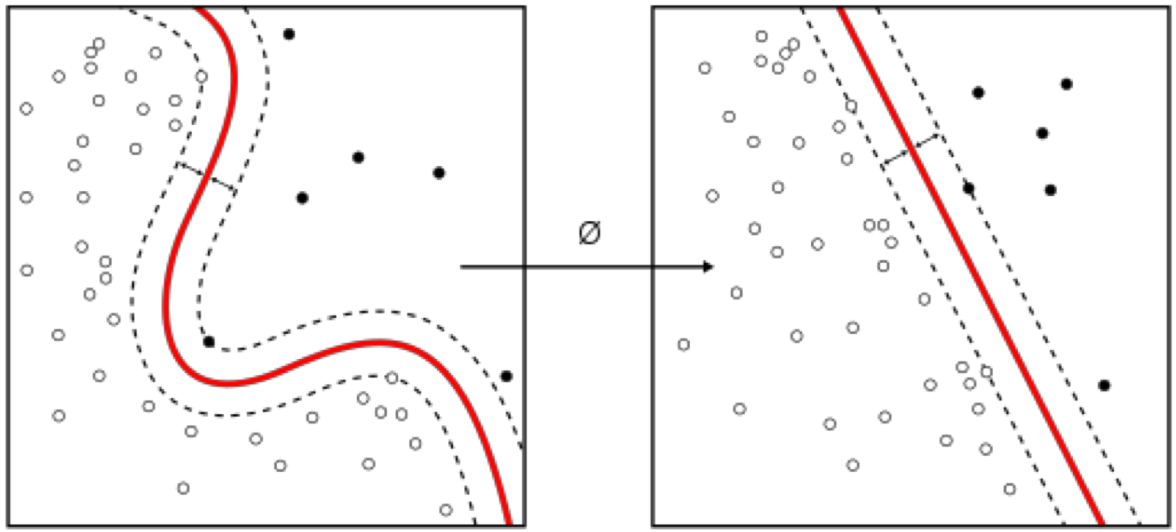
From last time…

- We learned about NumPy and Pandas[1]
  - Basis of data analysis objects
  - NumPy: Matrix representation of data
  - Pandas: data.frames in Python

[1] Which stands for Panel Data, and not the cute animals

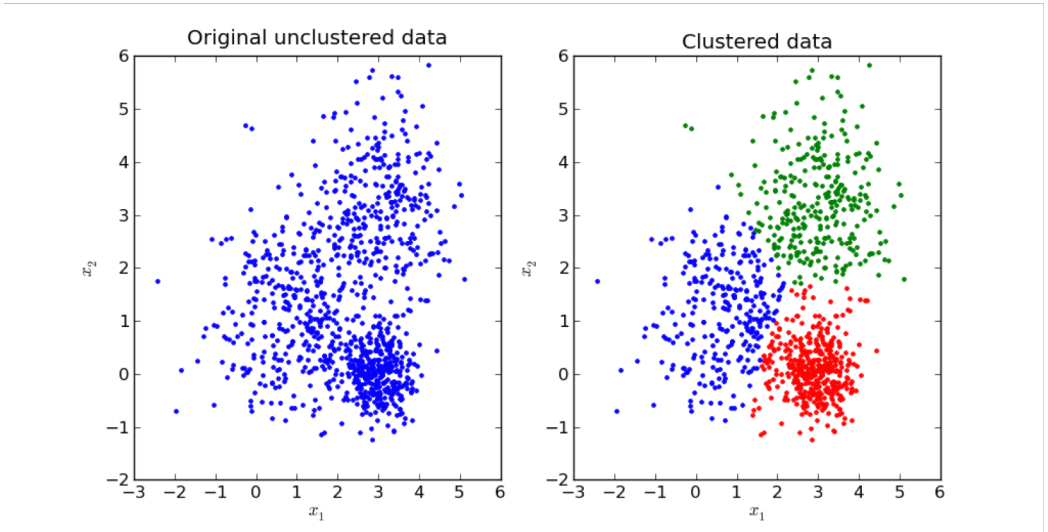
Overview of Machine Learning

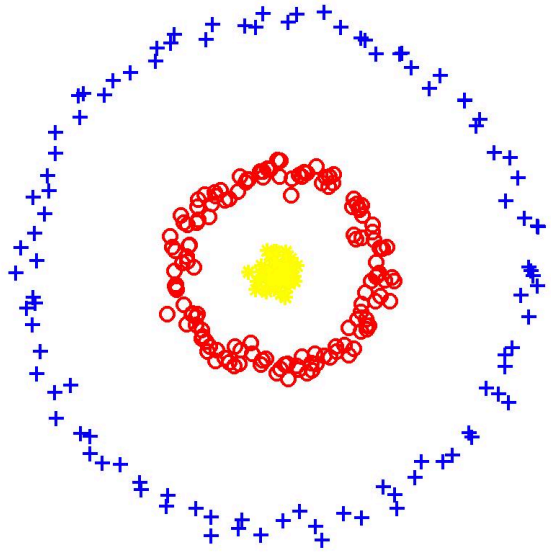
- Supervised Learning
  - We provide the computer with sample inputs and outputs, and the algorithm learns patterns that map the inputs to the outputs
    - Example: regression, trees, SVM, etc.



Overview of Machine Learning (cont.)

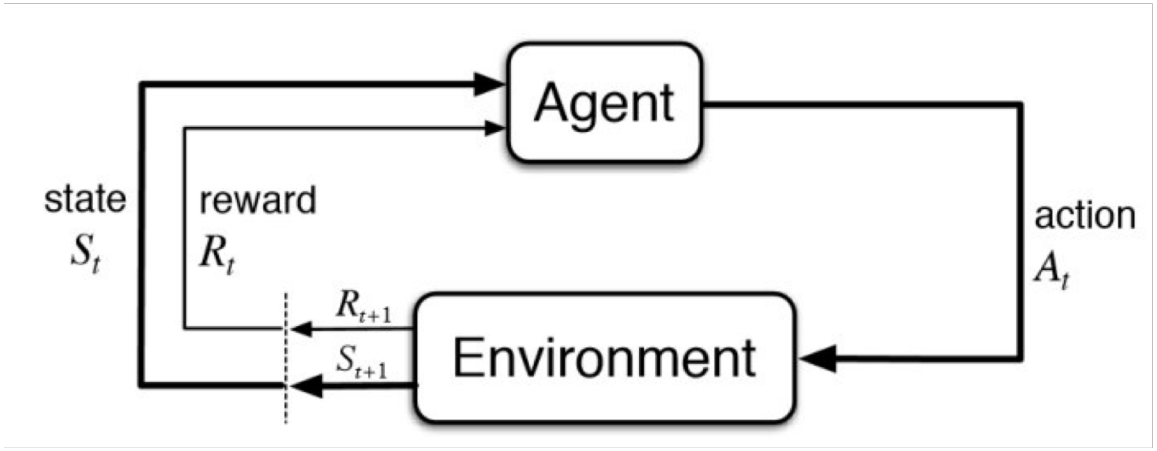
- Unsupervised Learning
  - We provide the computer with a bunch of data and the algorithm tries to detect some underlying structure – often can be used for feature engineering
    - **Example:** k-means clustering





Overview of Machine Learning (cont.)

- Other types:
  - Semi-supervised Learning
  - Active Learning
  - Reinforcement Learning



Supervised Learning

- Two main categories:
  - Classification
    - Y values are binary
    - Example: image recognition, marketing/clicks,
  - Regression
    - Y values take on values
    - Example: predicting stock market prices

Supervised Learning

- Recall from econometrics:

$$Y = \beta X + \varepsilon$$

- We generate predictions for Y using X:

$$\hat{Y} = \hat{\beta} X$$

Supervised Learning (cont.)

- Therefore, to find the optimal values of β, we minimize the mean square error:

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

$$\implies \min_{\beta} \frac{1}{n} \sum_{i=1}^{n} (Y_i - \beta X_i)^2$$

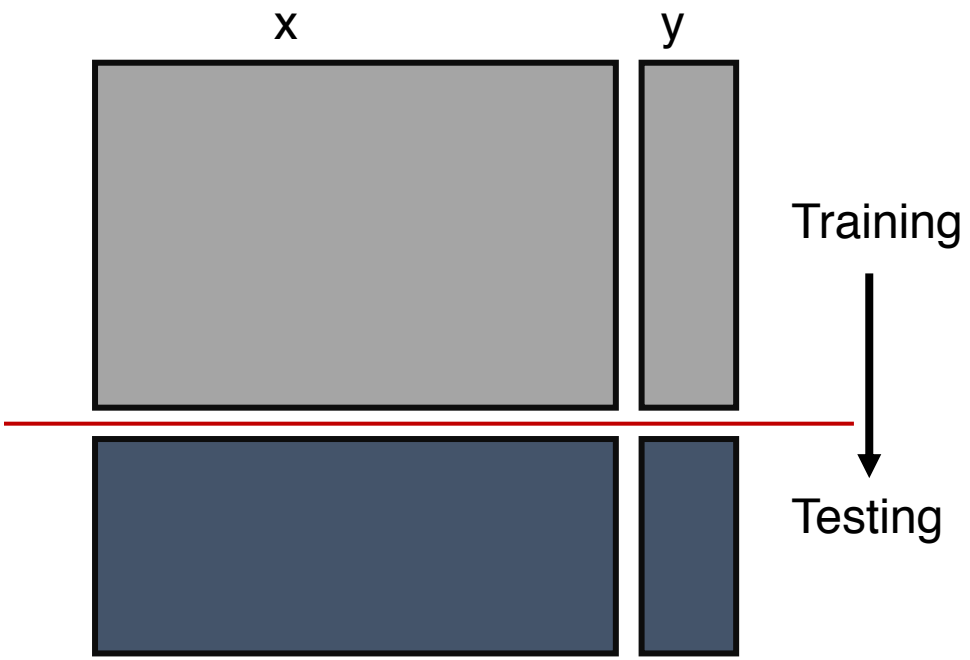
Supervised Learning (cont.)

- This is the same framework for which we want to think about supervised learning

- Different methods within supervised learning alter some component of this process by:
    1. Unique transformations to make non-linear relationships linear
    2. Different means by which we can minimize error or maximize our predictive capabilities

Supervised Learning (cont.)

- How do we assess how well our model is performing?
  1. Training Set (~2/3)
  2. Validation (Testing) Set (~1/3)



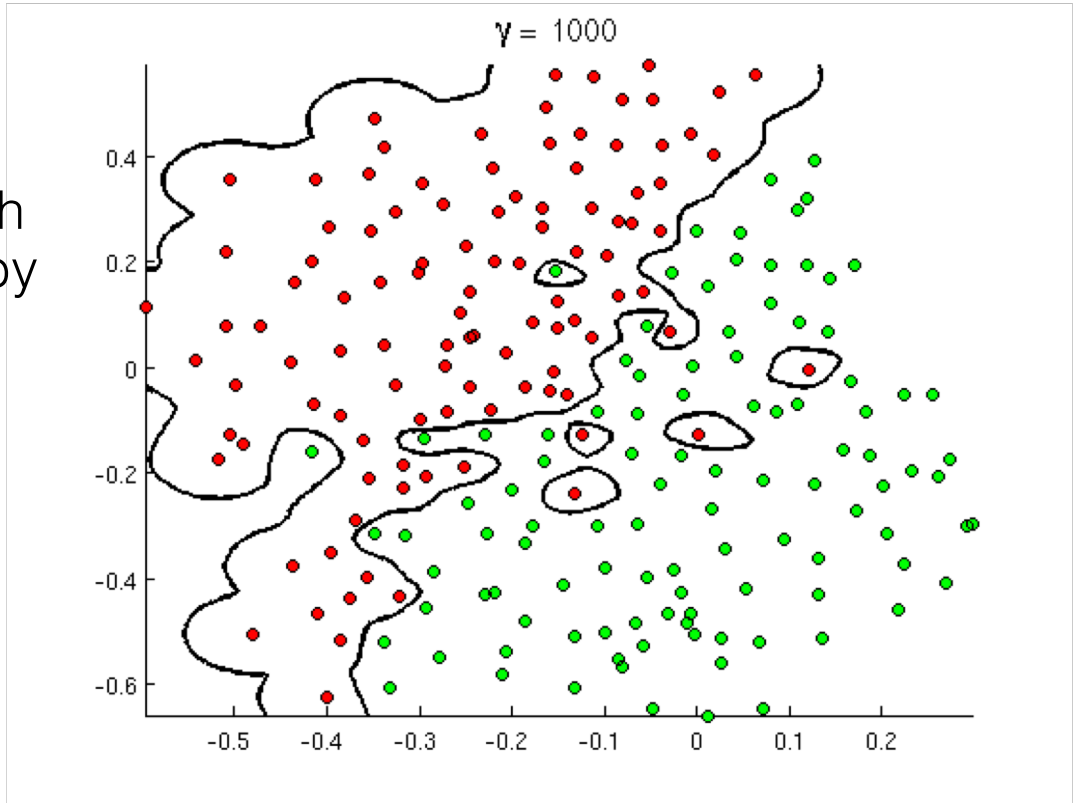
We fit our model first on the training data, and then measure the error accumulated by using that model on the testing data.

Python has a very useful function for this called:
`train_test_split()`

Supervised Learning (cont.)

Why do we do this?

- Overfitting!
- It is easy to get a very high in-sample accuracy rate by simply increasing the complexity of our model!
- However, a very complex model will not generalize very well and provide limited predictive capabilities



sklearn

- Python houses most of their machine learning models under one package known as sklearn

- Pro's:
  - You can import pretty much everything!

```
import [algorithm] from sklearn as algo
model = algo.fit(X_train, Y_train)
model.predict(X_test)
```

- Con's
  - Regression diagnostics are not always neatly packaged

Wine Example 🍷

Original Paper:

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis.

Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4): 547-553

Summary of Data:

- Red and White variants of Portuguese “Vinho Verde” wine
- Physicochemical variables describing the wine (i.e., pH)
- Wine experts evaluated each wine tested and assigned it a score between 0 to 10 (denoted as “quality”)

Example (cont.)

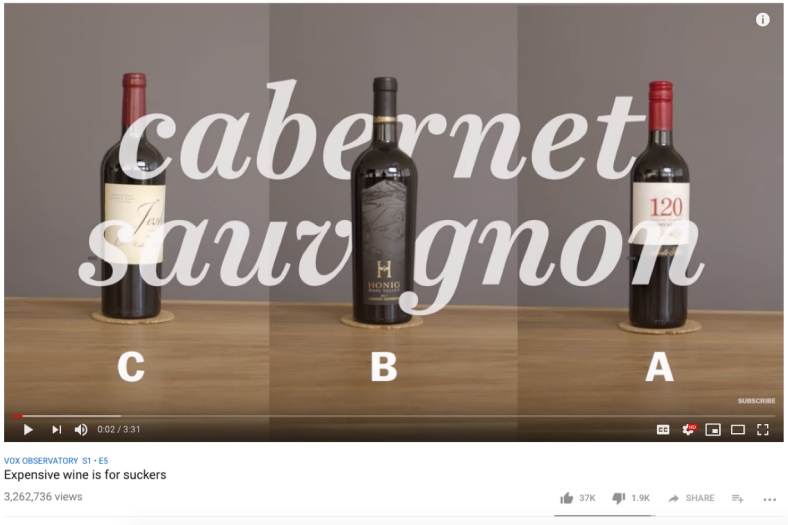
(Brief Digression)

<u>Disclaimer:</u>

Is it actually possible to evaluate wine “objectively”?

Can wine experts even discern the differences between expensive and cheap wines?

Questionable.



🍷 Example (cont.)

Digression aside, our goal is going to be:

1. Analyze the wine data set
2. Fit different classifiers to predict wine quality

Personal Suggestion:

For maximal empirical testing, try all the wines at home to see if your rankings converge with that of the classifiers.[2]

[2] I would suggest writing your classifier first before conducting your first-hand experiment.

🍷 Example (cont.)

- We will use three different classifiers:
  1. Logistic Regression
  2. Random Forest
  3. Support Vector Machine