FALL 2018

TEXAS
The University of Texas at Austin

APPLIED ENGINEERING DATA ANALYSIS, OPTIMIZATION AND VISUALIZATION

Basic machine learning: Regression

JOSHUA RHODES, PHD
Research Fellow/Adjunct Professor, The University of Texas at Austin

You have your 1-page term project proposal assigned on Canvas

- Be concise, but explain what you want to do
- We will make sure it is not too little or too much
- Should take about 30-40 hours effort

By now we have cleaned, merged, aggregated, and stored data, now what?

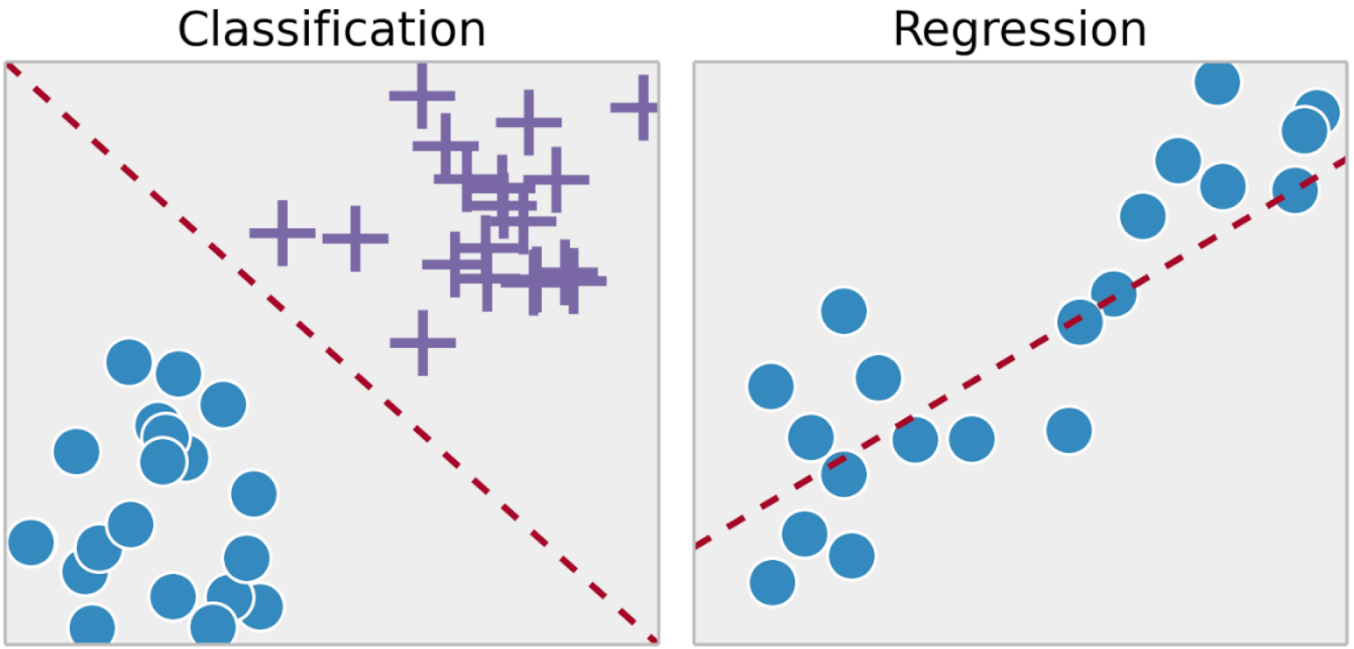
- Now comes the part that you are likely most after – getting that data to tell a story
- What story you want to tell and the data you have drive what methods you use

Machine learning is a collection of methods that allow computers to learn with data, very broadly:

- Supervised learning
  - Given both inputs and outputs, a computer learns rules to map inputs to outputs
  - Supervised learning is done using a **ground truth**, or in other words, we have prior knowledge of what the output values for our samples should be
- Unsupervised learning
  - Unsupervised learning, on the other hand, does not have labeled outputs, so its goal is to infer the natural structure present within a set of data points.

*https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

Supervised Learning



Supervised learning includes some techniques you might be familiar with

- Linear regression
  - Outputs a continuous value
- Logistic regression
  - Outputs a binary classification
- Neural networks
  - Similar to regression, but with less structured data inputs

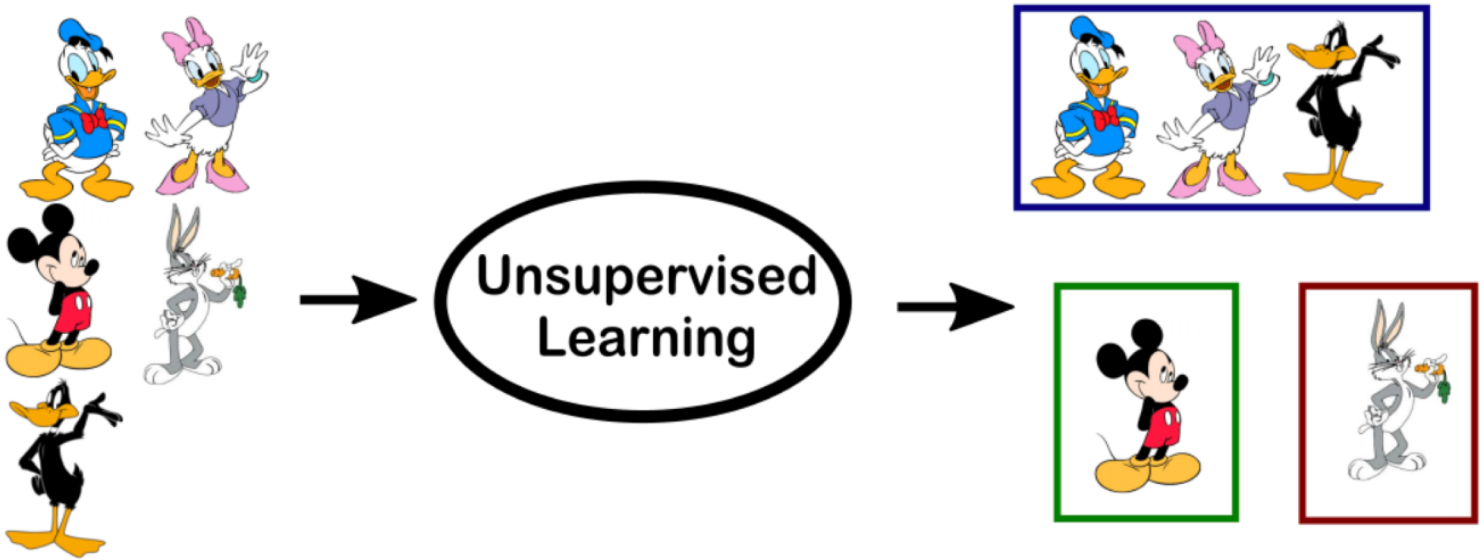
With these methods, we are trying to map inputs to outputs

- Output labels (yes/no, male/female, dead/alive)
- Continuous output = forms of regression
- “correct” output is determined entirely from the training data
  - Issues when too small or poor quality data

Two main considerations with supervised learning

- Model complexity
  - Complexity of the function you are trying to fit
  - Easy to overfit and just learn your training data
  - https://xkcd.com/2048/
- Bias/variance
  - Bias: constant error
  - Variance: amount by which the error may vary between different training sets

Unsupervised learning



Unsupervised learning: there is no spoon…

- Clustering
  - K-means clustering
    - Grouping objects by *like* characteristics
  - Hierarchical clustering
    - Top-down or bottom-up orderly grouping
- Density estimation
  - Estimating underlying density functions

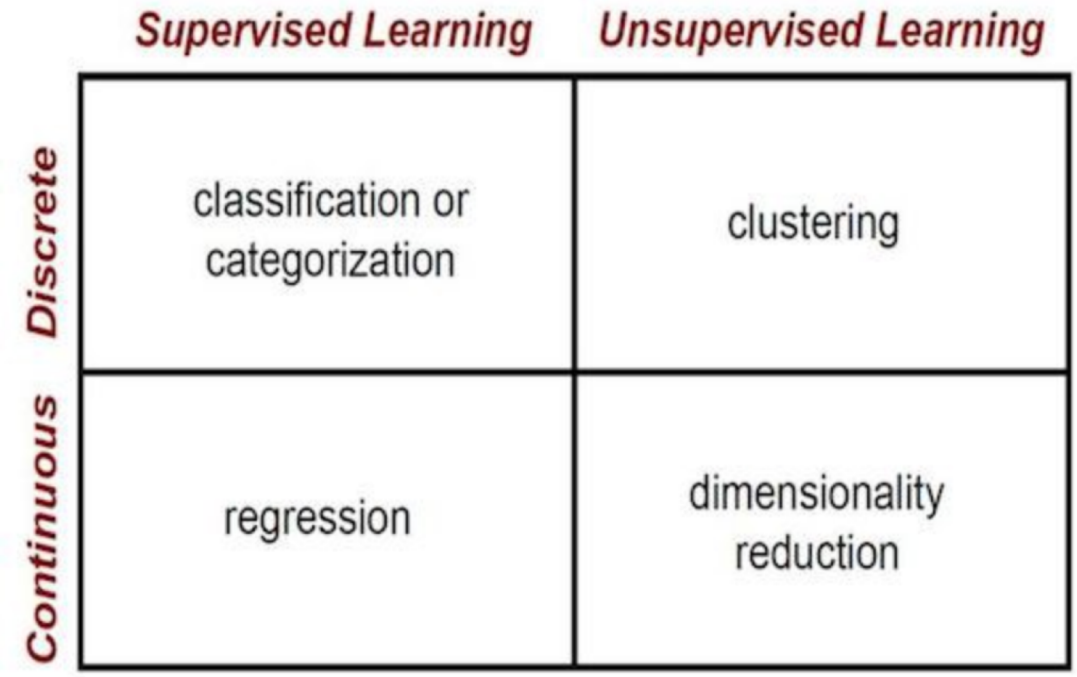
Unsupervised learning

- We wish to learn the inherent structure of our data without using explicitly-provided labels
- Since no labels are provided, there is no specific way to compare model performance in most unsupervised learning methods

Unsupervised learning

- Very useful in exploratory analysis because it can automatically identify structure in data
- Dimensionality reduction: methods used to represent data using less columns or features

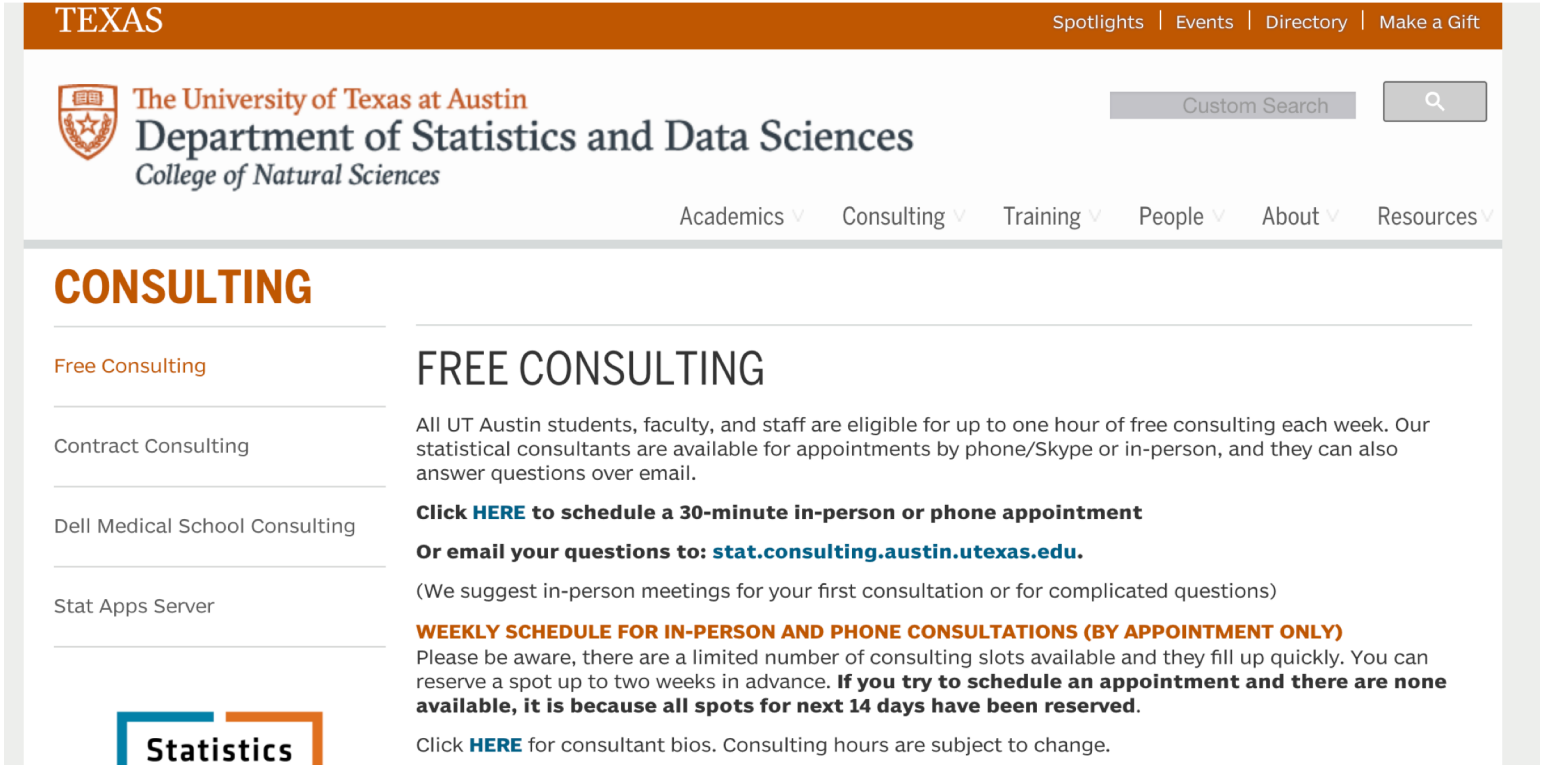
TI;dr


Supervised Learning
Unsupervised Learning
Discrete
Continuous
classification or categorization
clustering
regression
dimensionality reduction

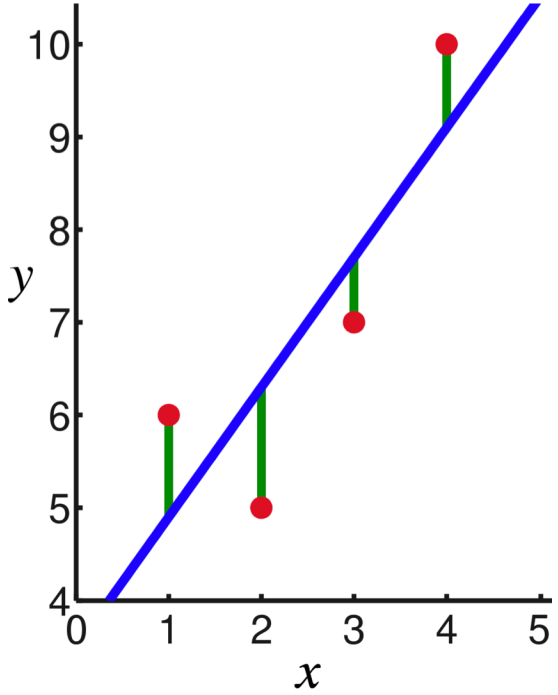
Some basic regression in R

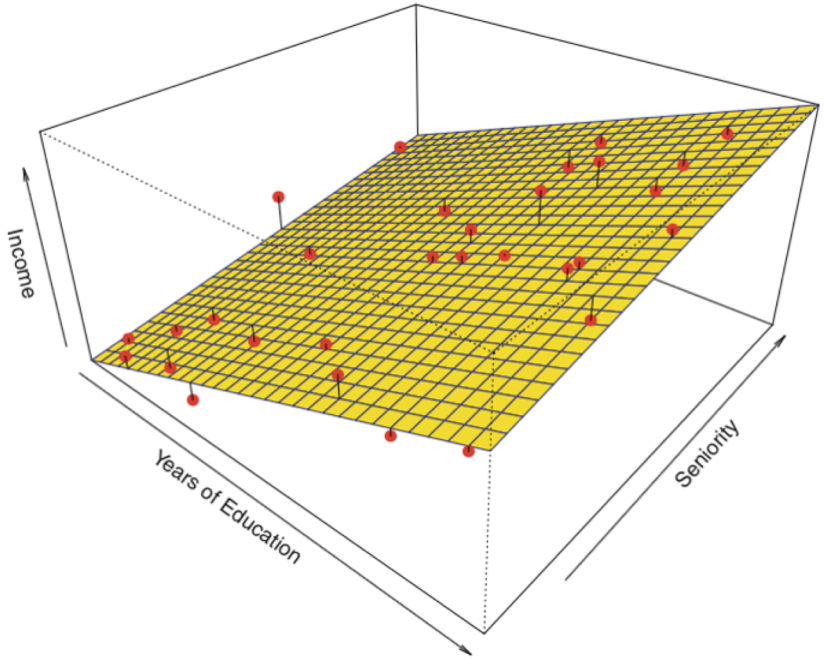
Warning: Regression is a complicated subject and taught in multiple semester-long classes. This is just likely to make you dangerous.

Stats are hard, help is free



Basically you are looking to define a function that minimizes the sum of square errors from the data to the function estimate





Given p dimensions, you *can* come up with a function that allows you to predict anything

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},$$

$$X = \begin{pmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix},$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

For OLS (most common):

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} = \left(\sum \mathbf{x}_i \mathbf{x}_i^\top \right)^{-1} \left(\sum \mathbf{x}_i y_i \right).$$

The inverse operator can get you into trouble later on down the road.

A matrix that is *linearly dependent*, cannot be inverted

- This means that one column is a linear combination of other columns
- This means that you can create a column of 0's
- That means the determinate of the matrix is zero -> singular matrix -> no inverse

Let’s dive in a bit