Breadth - Data

What Julian talked about.

How did he get the data?

How did he get the data?

- I am still trying to figure that out!!!!!
- He “scrapes” certain websites for their data.
- He basically searches a website to find where they store their data.

Data is not always pretty

- As you may guess, its not always the case that data comes in nice, easy to read formats.
- If your data is messy, there are ways to 'clean' the data in order to put it into a nice form to work with.
- We will be looking at data in the form of a .json file. (JavaScript Open Notation.)
- JSON files look very similar to python dicts which make them ideal and easy to interpret.

APIs

- Does anyone know what an API is?

APIs

- Does anyone know what an API is?

- It stands for Application Programming Interface.

- Basically, it is a way for a website to share data that is in a nice format.
- (Why would a website want people to access their data easily?)

APIs

- Try your luck with IMDB API, Twitter API, Yelp API, and more.

- There are a lot of resources out there if you look for them.

Julian's Data

- Julian has accumulated some very nice, easy to read data.
- You can find it here:
- https://github.com/ucsd-cse-spis-2016/ucsd-cse-spis-2016.github.io/blob/master/_lectures/week1/breadth.md
- Today, we will talk a little about what julian did.

Julian's Data

- He uses some "libraries"
- import numpy
- import urllib
- import scipy.optimize
- import random
- We will talk about how to install these another time.

Julian's Data

- In Julian's code he has this thing to parse the data

- def parseData(fname):
  - for l in urllib.urlopen(fname):
    - yield eval(l)

Julian's Data

- He uses a generator

- Why may this be better than downloading all the data at once?

Julian's Data

- Now that we have the data stored, what fun things can we do with it?

- We can try to use the data to make some predictions about things we don't know.

Beer Data

- Julian made a predictor.

- What is a predictor?

Beer Data

- Julian made a predictor.

- What is a predictor?

- It's a function f, that takes some input (user,features,item) and outputs a prediction of the behavior of that input (ratings given, whether the user will purchase some item.

Beer Data

- We can use some data to help “train” our predictor to be more accurate.
- The easiest straightforward predictor is to take the average rating of your training data and say that for each user, they will rate it with the average rating.
- f(user) = average
- This is a constant function.

Beer Data

- The way we will first do this is make a vector of all ones
- X = [[1],[1],[1],[1],[1]…,[1],[1],[1]]
- and a vector of the ratings of all of the data points
- y = [1.5, 3.0, 3.0, …, 4.0, 4.5]
- Then we want to solve for a value θ that gives me the least "error" when I multiply

$$X\theta = \text{y}$$

MSE (mean squared error)

The mean squared error is the average of all the “squared errors”, i.e. (prediction-actual)^2

Beer Data with simple predictor.

- Then we want to solve for a value θ that gives me the least "error" when I multiply

$$X\theta = \mathrm{y}$$

The best "fit" for θ is the average of all the values in y.

$$\theta = 3.88871$$

Linear regression

- We can have python (numpy) give us the best "fit" for θ in the equation $X\theta = y$ by calling
- numpy.linalg.lstsq(X, y)

- The result is a tuple of 4 values.
- The first value is theta.
- (the second value is the sum of the squared errors.)

Linear regression

- Recall when Julian attempted to make some predictor using the age of the user to help predict the rating given. He assumed there was some sort of "correlation" between age and rating.

- Maybe older people rated beer higher than younger people. Is this a reasonable assumption?

Linear regression

- Recall when Julian attempted to make some predictor using the age of the user to help predict the rating given. He assumed there was some sort of "correlation" between age and rating.

- Maybe older people rated beer higher than younger people. Is this a reasonable assumption?

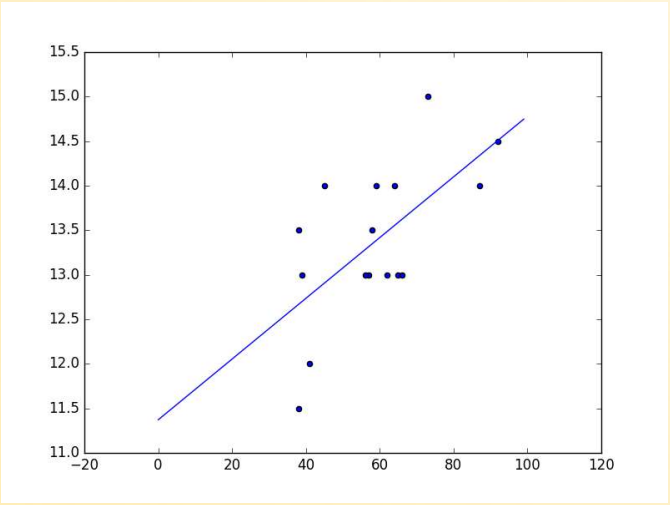
Linear regression

- To solve this problem, we will consider X to be a list of vectors [[1,age_1],[1,age_2],…,[1,age_n]]
- and y to remain the list of ratings
- Then we still want the best fit for θ in the equation $X\theta = y$ but now θ is a 2-dimensional vector.

$$\theta[0] + age * \theta[1] = y$$

Linear regression

- With these two values, we can define a line.
- This line is the best ‘fit’ for the data



Linear regression

- With these two values, we can define a line.
- This line is the best ‘fit’ for the data