

Introduction to Regression and Regularization

Erik Istre



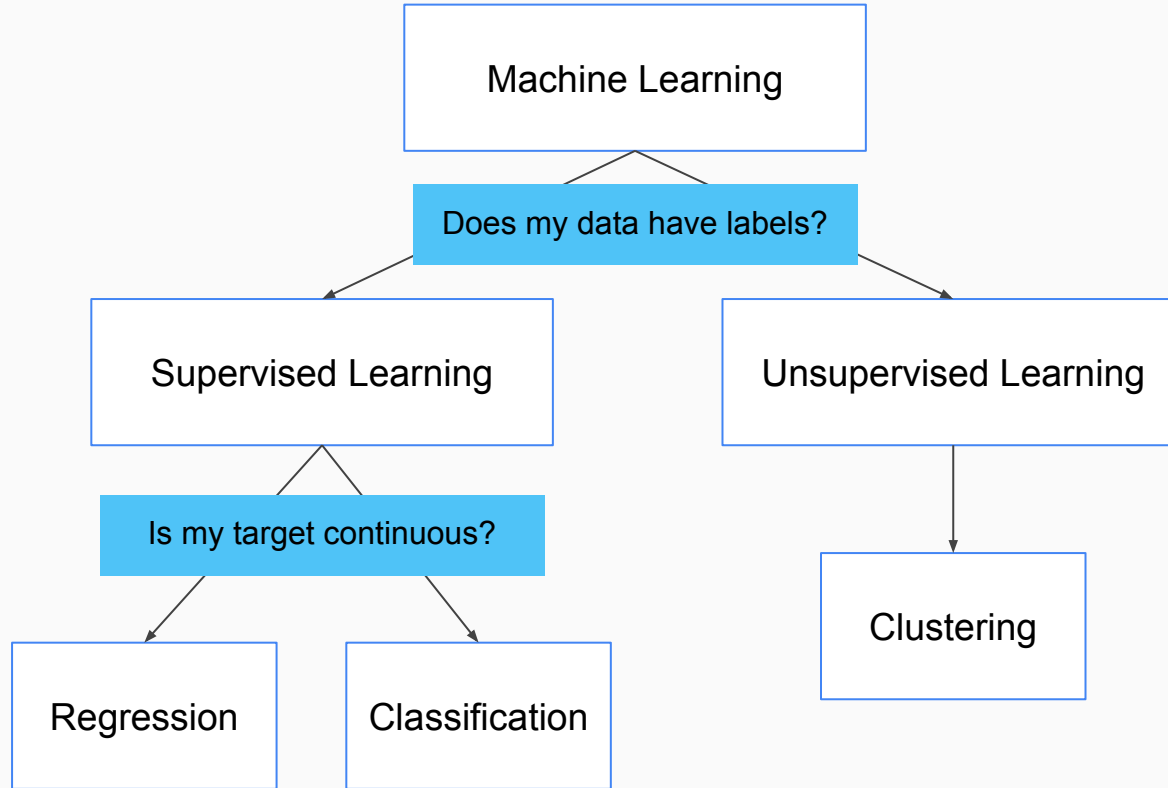
Overview

1. How does linear regression fit into ML?
2. What is linear regression?
3. Univariate Linear Regression
4. Multivariate Linear Regression
5. How to Do It In Python
6. Categorical vs Continuous Data
7. Optimizing Linear Regression
 - a. Determining Accuracy
 - b. Ways to Do “Better”
8. Ridge and Lasso Regression

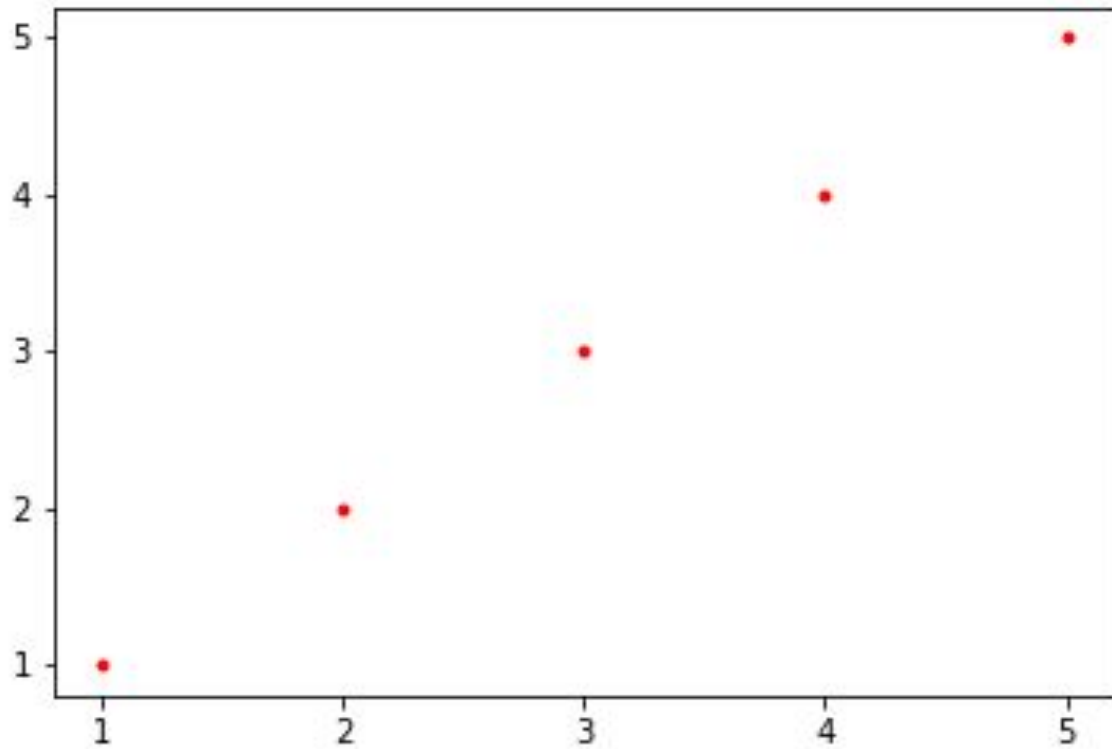


<https://brightonleadership.com/2016/07/05/why-does-the-path-matter/>

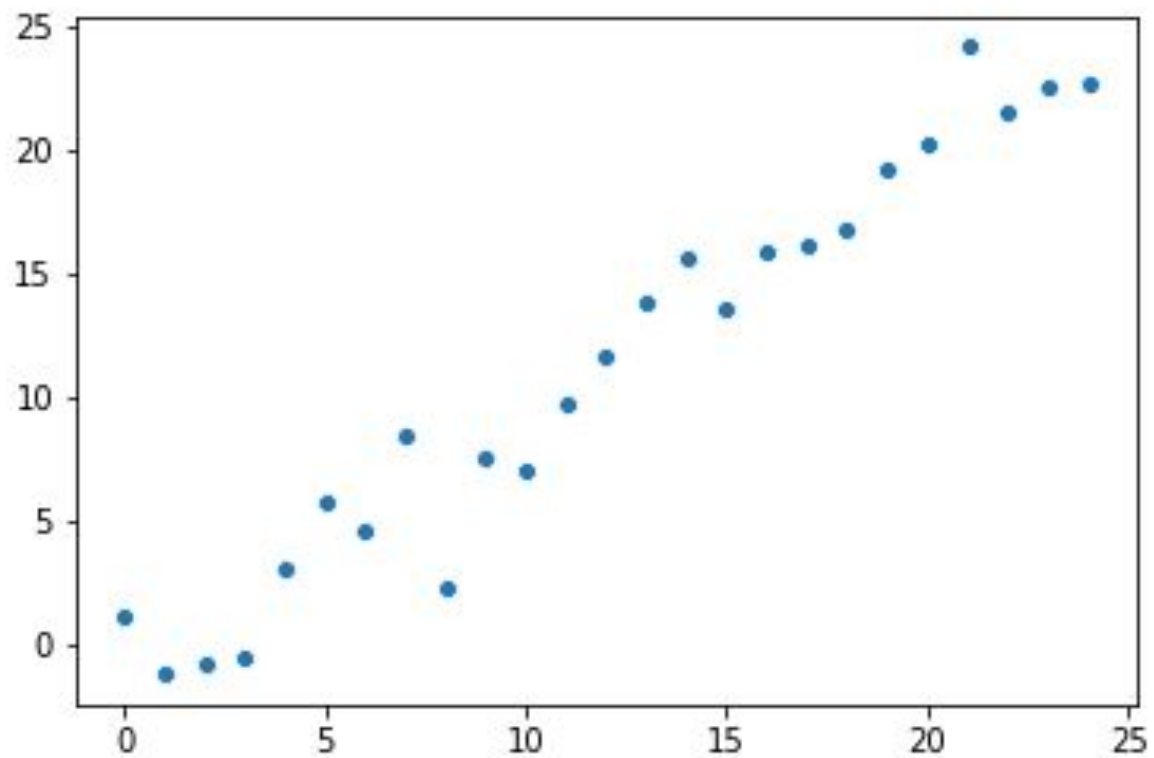
Machine Learning



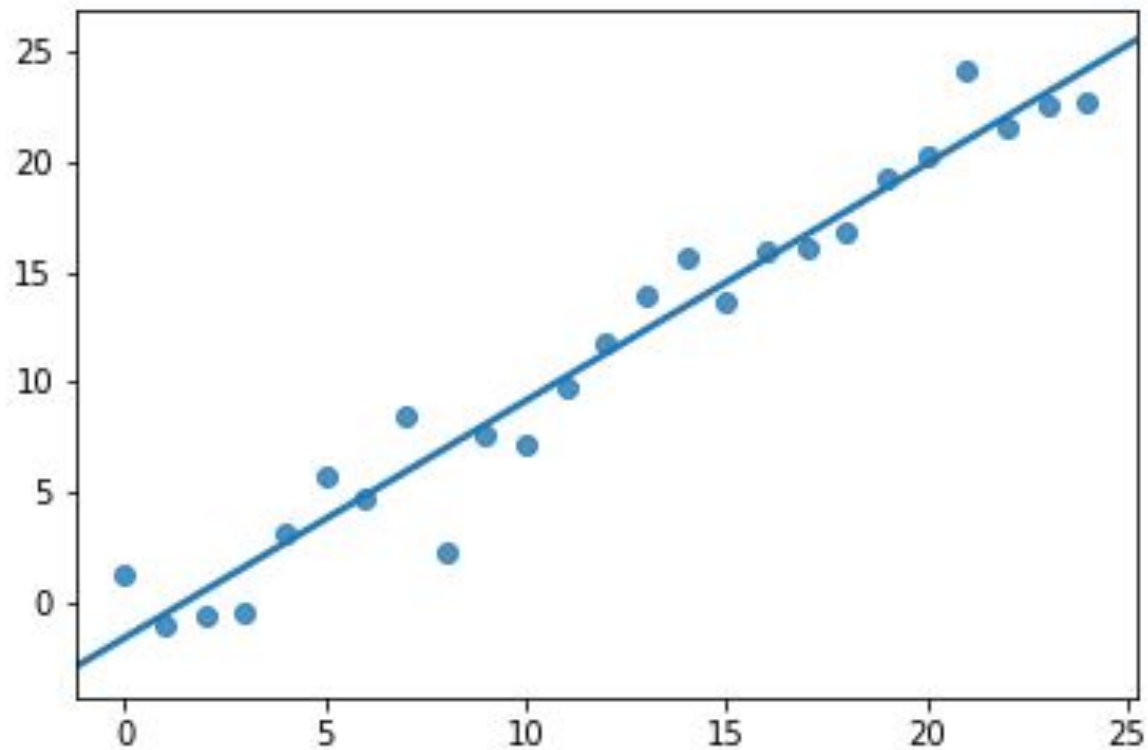
What is Linear Regression?



What is Linear Regression?

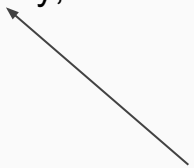


What is Linear Regression?



Let's Put It In Context

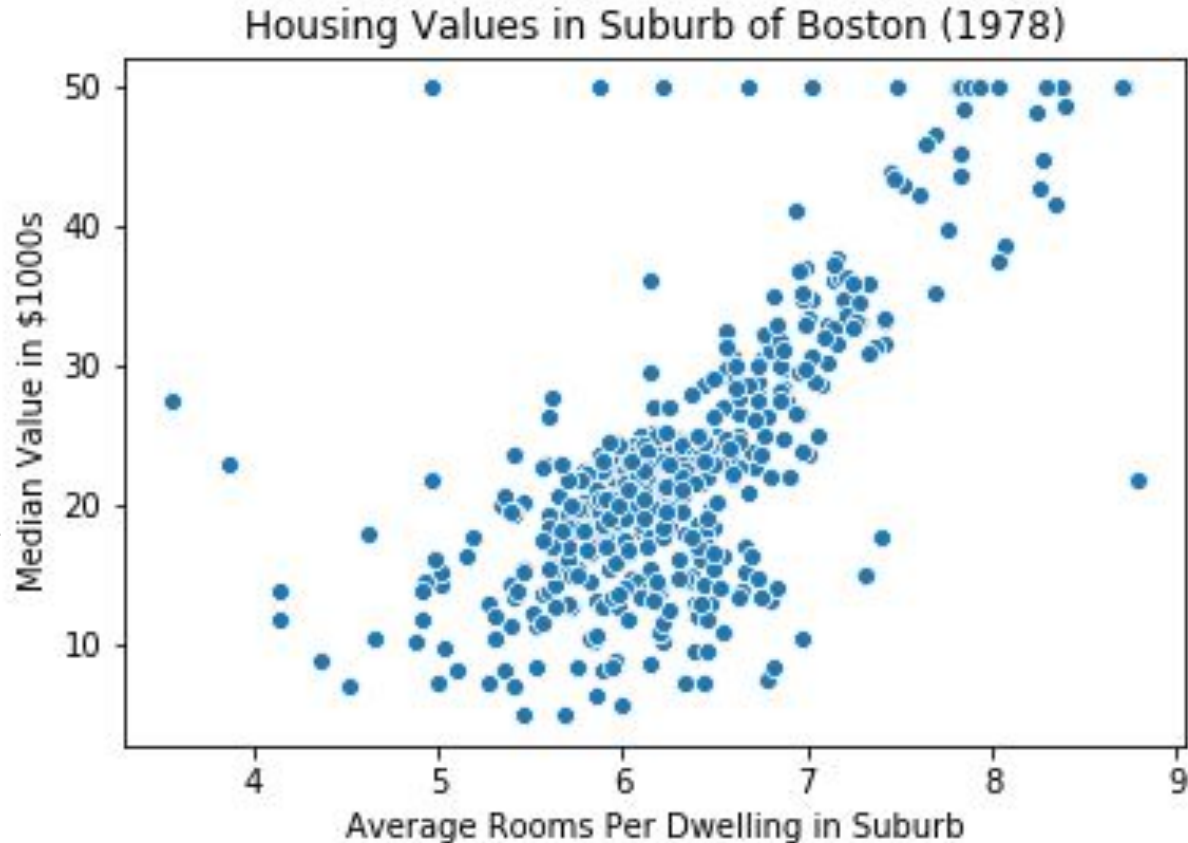
- Housing Values in Suburbs of Boston
 - Harrison, D. and Rubinfeld, D.L. (1978) Hedonic prices and the demand for clean air. J. Environ. Economics and Management 5, 81–102.
 - Belsley D.A., Kuh, E. and Welsch, R.E. (1980) Regression Diagnostics. Identifying Influential Data and Sources of Collinearity. New York: Wiley.
- Sklearn, a Python library, comes preloaded with this dataset for experimentation.



A “library” is a prepackaged collection of algorithms and data for some set purpose.

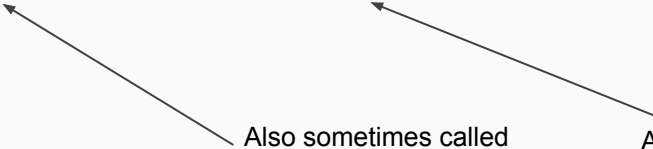
What is Linear Regression?

Median: the value in the middle of a sorted list of values, for example the median of [1, 2, 3, 4, 5] is 3



How to Find the Line

- Our goal is a line that “explains” the data.
 - How does the “feature” relate to the “target”.



Also sometimes called
the explanatory or
independent variable.
Usually the x.

Also called dependent variable,
response, or outcome variable.
Usually the y.

Let's Have Math Do It For Us

DEAR MATH

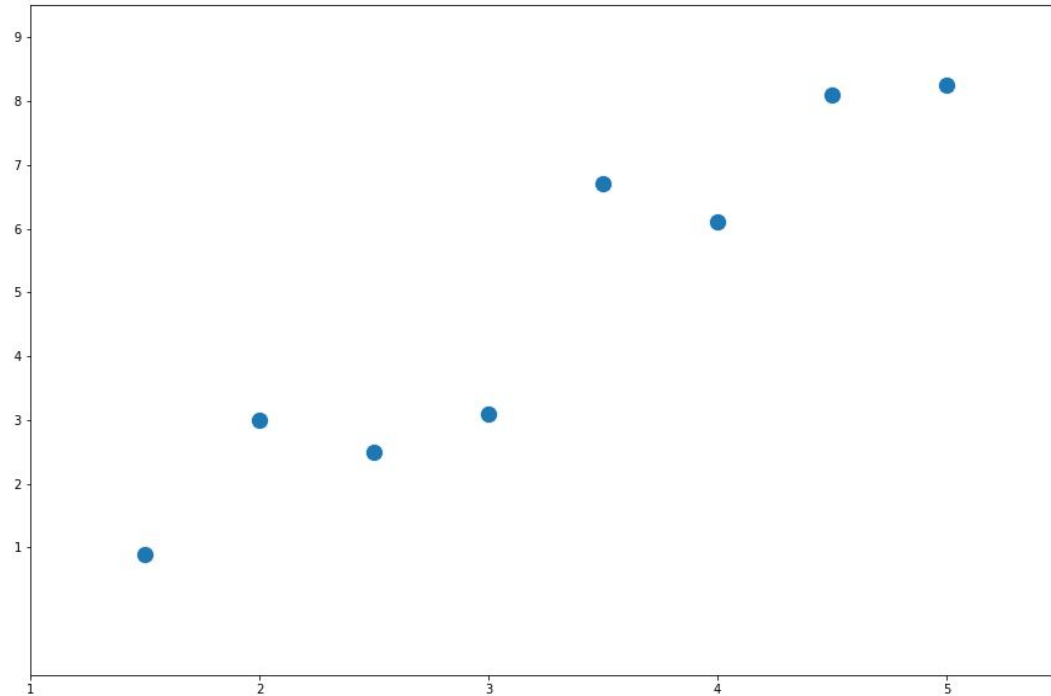
PLEASE GROW UP AND
SOLVE YOUR OWN
PROBLEMS,

I WON'T HELP YOU FIND
YOUR X
AND DON'T ASK Y.

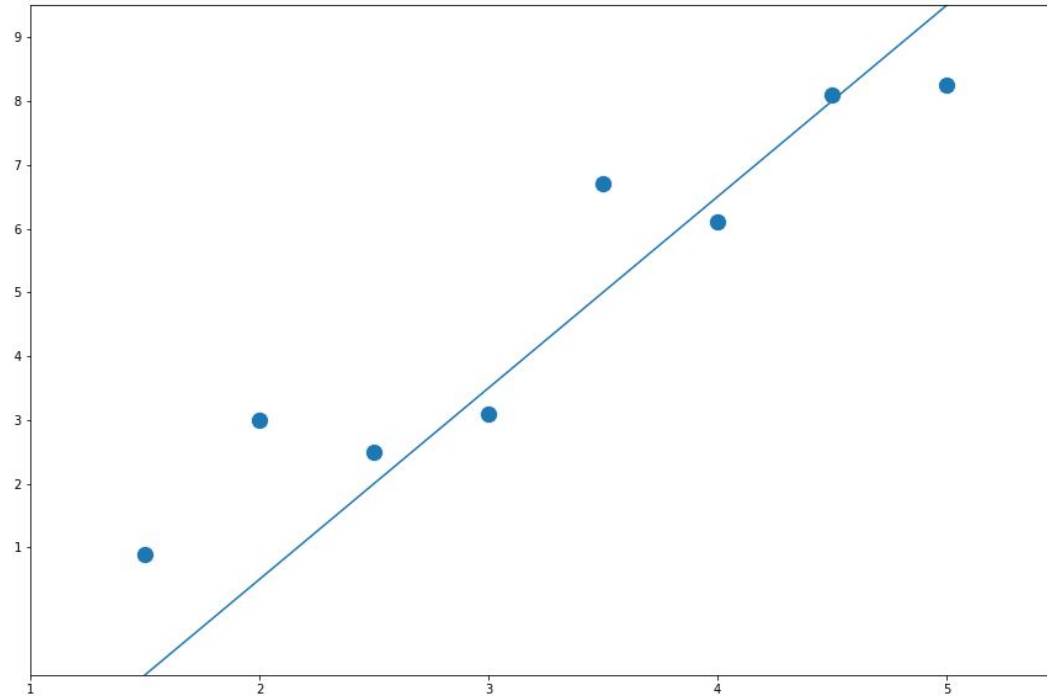
<https://boldomatic.com/p/tD02fQ/dear-math-please-grow-up-and-solve-your-own-problems-i-won-t-help-you-find-your>

- Let's define the problem precisely.
 - Then math and computers can do it for us!
- We need a measure of how “bad” our line is.

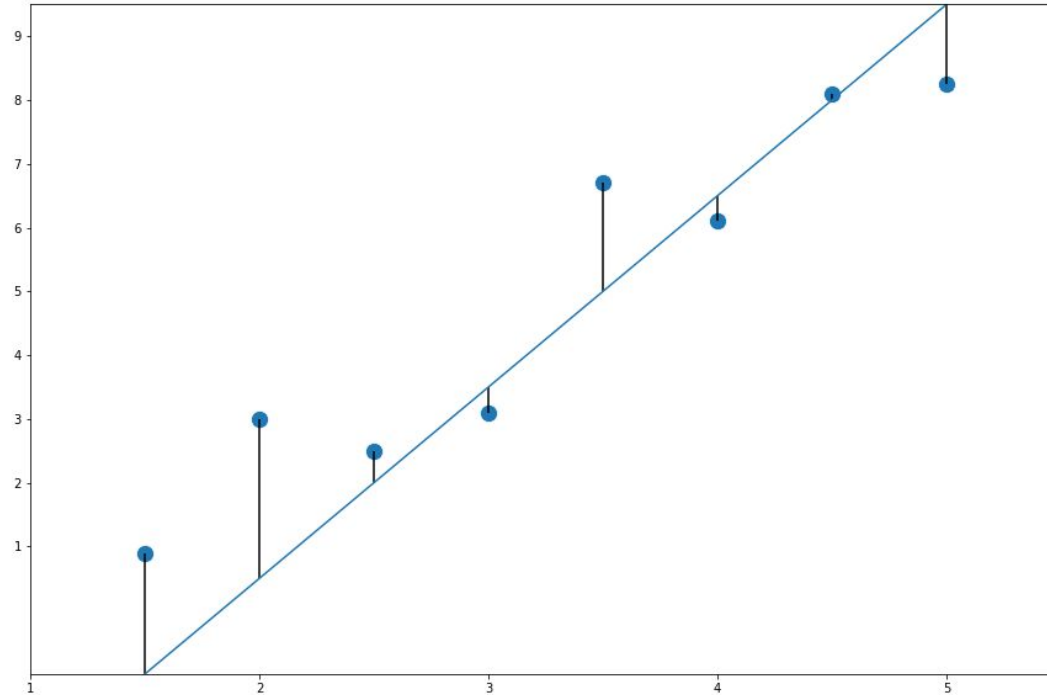
Let's Have Math Do It For Us



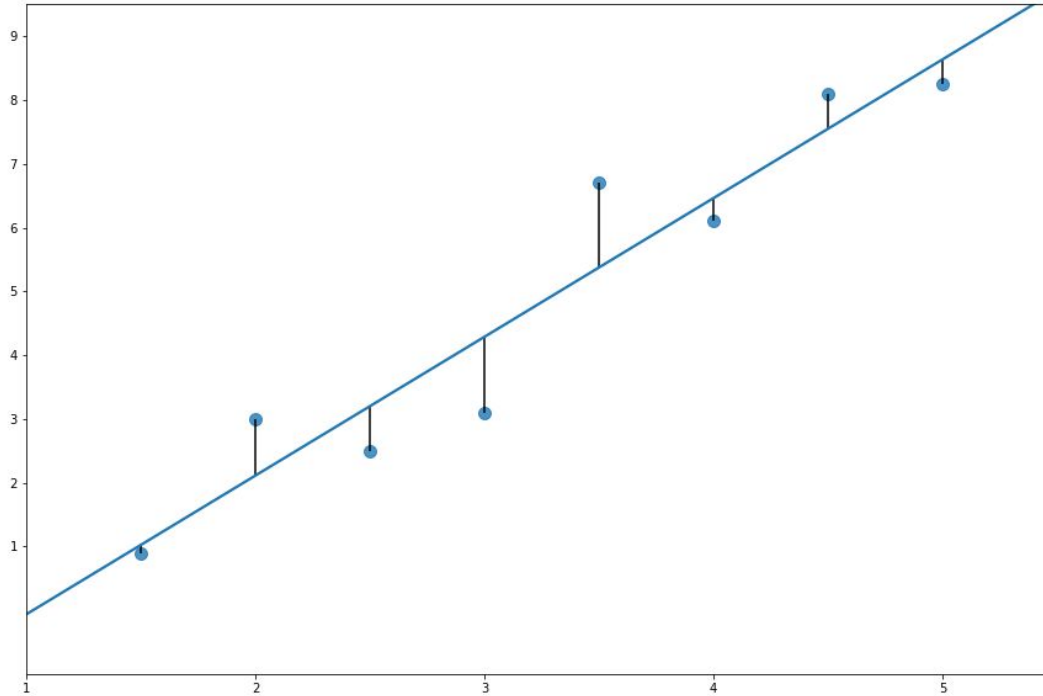
Let's Have Math Do It For Us



Let's Have Math Do It For Us



Let's Have Math Do It For Us



The MATH!



<https://i.pinimg.com/236x/43/3d/c6/433dc65c867f4112b28e749c9efb6f4d--best-holiday-movies-kid-movies.jpg>

- The most important part is understanding the “why” but not the “y”.
- Please ask any questions as we go through this!

The Setup

- The equation for a line is $y = mx + b$. The m is the “slope” or “coefficient of x ” and the b is the intercept or bias.

Remember the “hat” marks our model’s values: the suspicious imposters.

The brackets denote a list.

Like

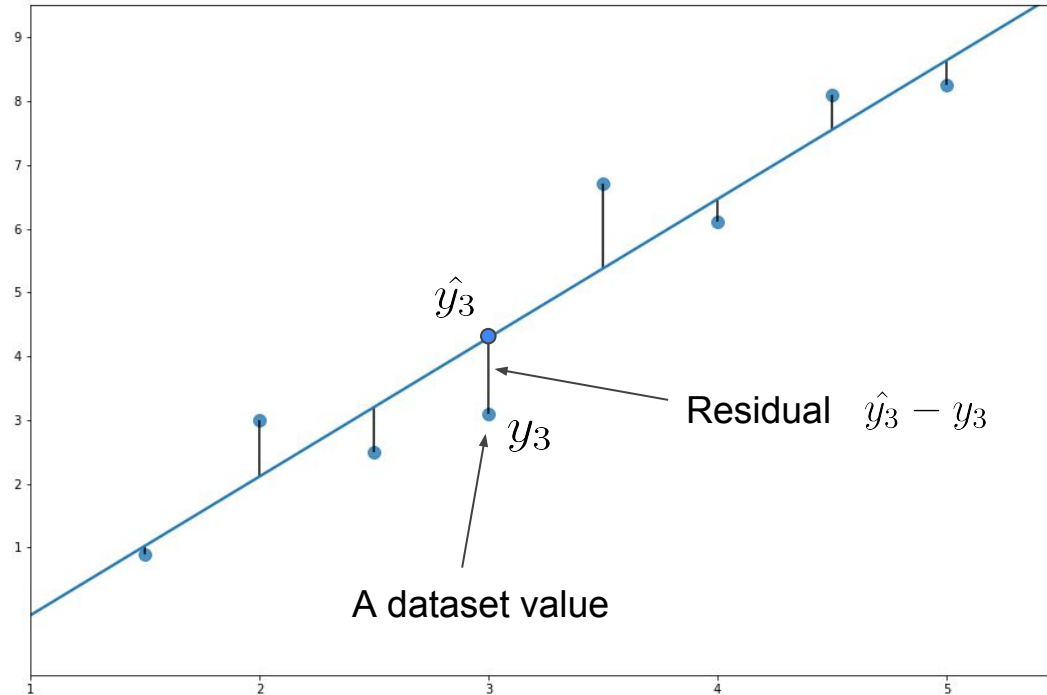
1. First thing
2. Second thing
3. Third thing

Except they are usually weird and start with 0.

Residuals

- To find the line of best fit, we minimize the “residual sum of squares”.
- A residual is the difference between the true value of a data point, and our model’s predicted value.

Residuals



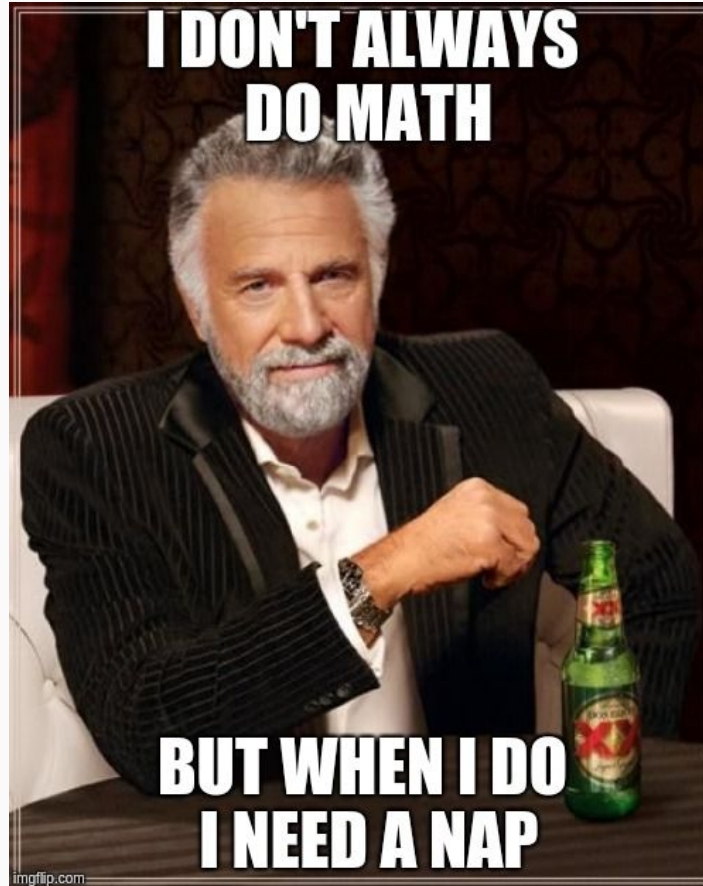
Residual Sum of Squares

- Residuals are usually defined as $y_i - \hat{y}_i$.

Our Wonderful Loss Function!

- Let n be the number of real data points that we have.

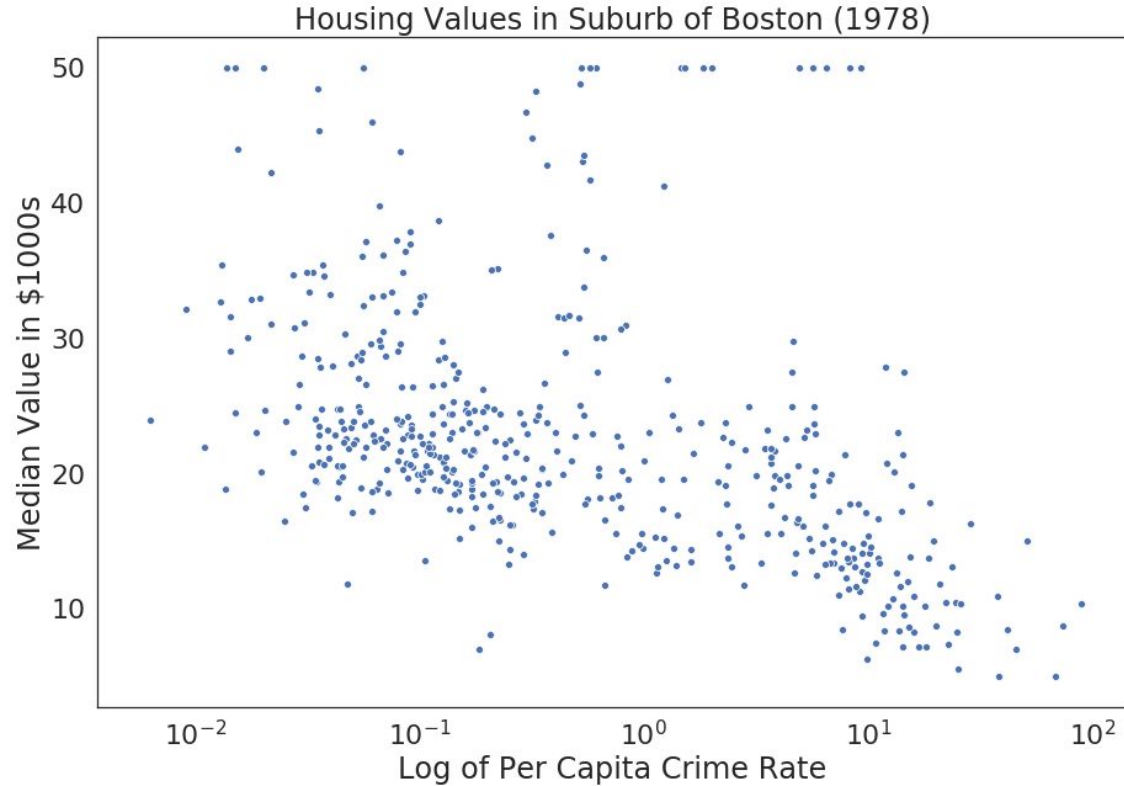
Whew! Take a breather and ask questions.



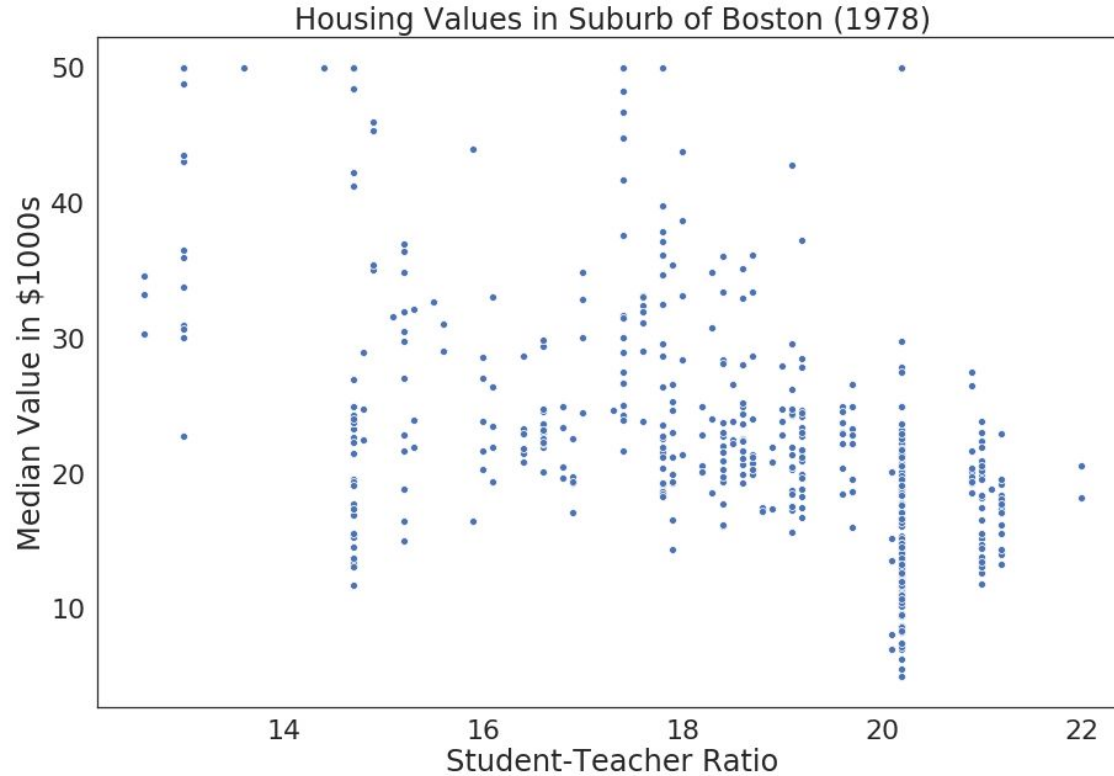
More Than One

- Rarely does one feature tell us everything about our target.
- For example, home values aren't just about the number of rooms.

Crime Rates



Student Teacher Ratio



Time for a little more math!



More features, more math

- To train a linear regression with two features, we need a “3D Line” otherwise known as a plane.
 - Imagine an infinite sheet of paper.
- To train a linear regression with more than 2 features, we need a “hyperplane” or more!

A Plane with Residuals

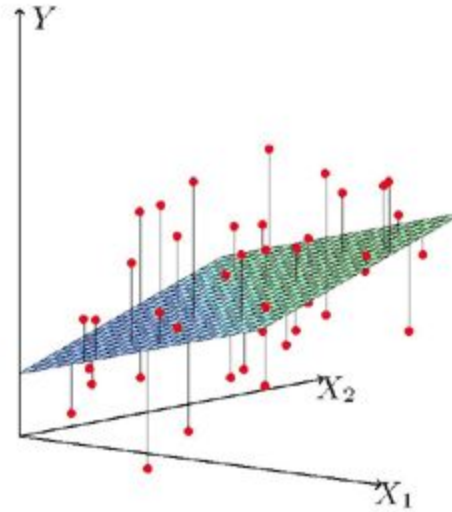


Figure 3.1: *Linear least squares fitting with $X \in \mathbb{R}^2$. We seek the linear function of X that minimizes the sum of squared residuals from Y .*

More features, more math

- Let p equal our number of features.
- The equation for our linear regression in higher dimensional space:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$$

How to Python!



- Install Anaconda
 - <https://www.anaconda.com/download/>
- Use a terminal or fancier, a jupyter notebook.
 - Go to your conda terminal (if Windows), or command line and type “jupyter notebook”.
 - Take a moment to feel like a cool hacker!
- Click new in the top right and create a new notebook.
 - Probably called Python [default].
- Give your new notebook a title by clicking Untitled at the top.

Load Your Libraries

- Type the magical incantations in the first cell.
 - `import sklearn`
 - `from sklearn.datasets import load_boston`
 - `from sklearn.linear_model import LinearRegression`
 - `import pandas as pd`
- This sets up your environment with some cool libraries.
- Press shift+enter to “evaluate” the cell.
 - If you ever need a new cell, click the plus button in the top left to add a new “cell”.

Load Your Data!

- We're going to load our dataset in this new cell.
 - `b = load_boston()`
 - `boston = pd.DataFrame(b['data'], columns=b['feature_names'])`
 - `boston['MEDV'] = b['target']`
- Add another cell and try evaluating
 - `boston.head()`

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2

Now We Can Make Our Model!

- Type the following for a univariate model.
 - `model_uni = LinearRegression()`
 - `model_uni.fit(boston[['RM']], boston['MEDV'])`
- You just made a regression model using the “average number of rooms” to predict “median value”.
- Try a multivariate one!
 - `model_multi = LinearRegression()`
 - `model_multi.fit(boston[['RM','CRIM','PTRATIO']], boston['MEDV'])`

Let's Have Some Fun

- We can get some predictions from our model.
 - What is the expected home value in a suburb where the average amount of rooms per home is 4?
 - `model_uni.predict([[4]])`
 - `array([1.73781515])` - This is a target value of 1.73781515.
 - What is the expected home value in a suburb where the average amount of rooms per home is 4, the crime rate per capita is .05 and the student-teacher ratio is 15?
 - `model_multi.predict([[4, .05, 15]])`
 - `array([10.09674222])` - This is a target value of 10.09674222.

How “good” is our linear regression?

- A linear regression’s accuracy can be measured by its R^2 score.
- This is a number between 0 and 1, higher is better.
- It describes what amount of the “change” in the data the model explains.
- Sklearn will tell us this value like this:
 - `model_uni.score(boston[['RM']], boston['MEDV'])`
 - 0.4835254559913343
 - `model_multi.score(boston[['RM', 'CRIM', 'PTRATIO']], boston['MEDV'])`
 - 0.5934141551136979

Pros/Cons

Easy to understand.

Interpretable.

Can do pretty well even though it makes strong assumptions.

Only captures linear relationships.

Vulnerable to outliers.

e.istre91@gmail.com

<https://github.com/eistre91>

<https://www.linkedin.com/in/erikistre/>

Image:

<https://www.pexels.com/photo/green-water-fountain-25769/>



Continuous vs Categorical Data

Continuous data:

An infinite (or near infinite) number of numerical values in some range.

Examples:

Speed, distance, time, money...

Categorical data:

A finite number of categories or groups. May be comparable to each other or not.

Examples:

Color, rank, gender, limited frequency options...