Lesson 6:
Intro to Machine Learning & Linear Regression

Course Agenda

- **Machine Learning**
  - What is ML
  - Neural Networks
  - Deep Learning
  - Supervised Learning Process
  - Unsupervised Learning Process
- **Linear Regression**
  - History
  - Example
  - Evaluating Performance
  - Example in Python

Introduction to Machine Learning

- I advise you to read **Introduction to Statistical Learning** by **Gareth James** as a book for the mathematical theory.
- Read Chapters 1, 2 & 3 to gain a background understanding of Machine Learning and Linear Regression.

It's freely available online:

https://github.com/qx0731/Sharing_ISL_python

What is Machine Learning

- Machine learning is a method of data analysis that automates analytical model building.
- Using algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look.

What is it used for?

- Fraud detection.
- Web search results.
- Real-time ads on web pages
- Credit scoring and next-best offers.
- Prediction of equipment failures.
- New pricing models.
- Network intrusion detection.
- Recommendation Engines
- Customer Segmentation
- Text Sentiment Analysis
- Predicting Customer Churn
- Pattern and image recognition
- Email spam filtering
- Financial Modeling

What are Neural Networks?

- **Neural Networks** are a way of modeling biological neuron systems mathematically.
- These networks can then be used to solve tasks that many other types of algorithms cannot (e.g. image classification)
- **Deep Learning** simply refers to neural networks with more than one hidden layer.

Machine Learning

Machine Learning

- Automated analytical models.

Neural Networks

- A type of machine learning architecture modeled after biological neurons.

Deep Learning

- A neural network with more than one hidden layer.

Machine Learning

- There are two different types of machine learning:
    - **Supervised** Learning
    - **Unsupervised** Learning

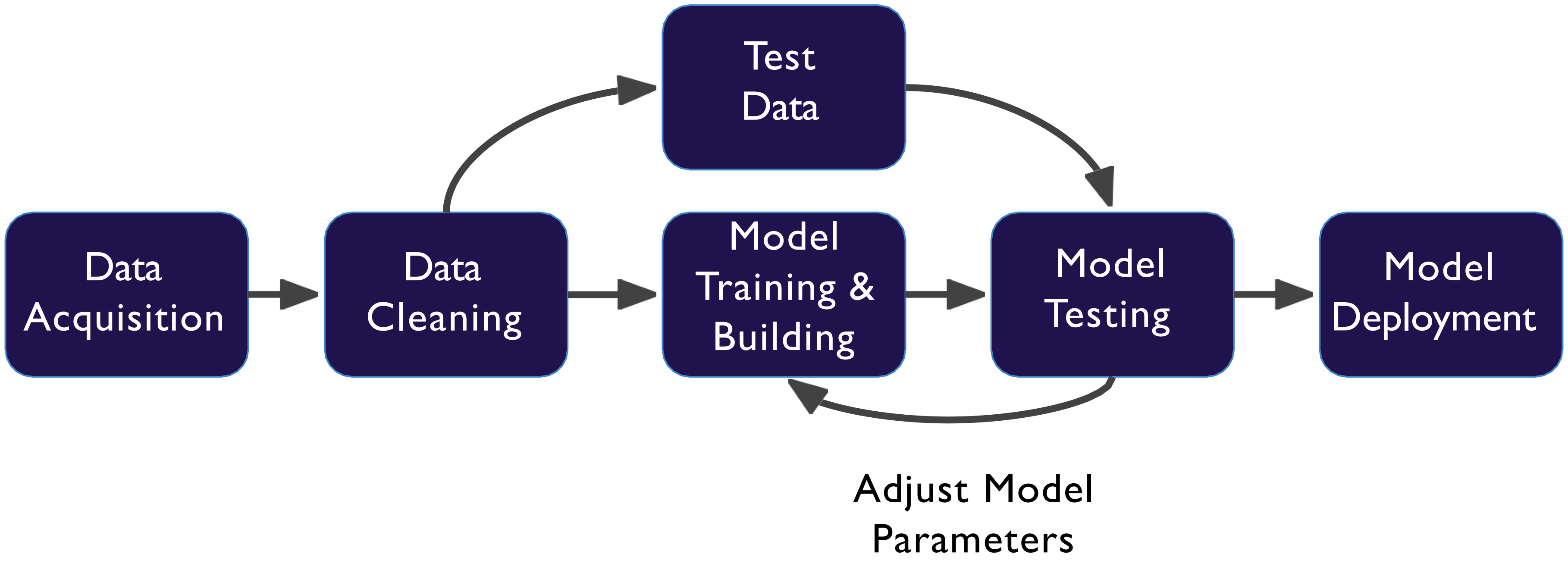
Supervised Learning

- **Supervised learning** algorithms are trained using **labeled** examples, such as an input where the desired output is known.
- For example, a segment of text could have a category label, such as:
  - **Spam** vs. **Legitimate** Email
  - **Positive** vs. **Negative** Movie Review

Supervised Learning

- The network receives a set of inputs along with the corresponding correct outputs, and the algorithm learns by comparing its actual output with correct outputs to find errors.
- It then modifies the model accordingly.
- Supervised learning is commonly used in applications where historical data predicts likely future events.

Machine Learning Process



Issue with the Splitting!

- What we just showed is a simplified approach to supervised learning, it contains an issue!
- Is it fair to use our single split of the data to evaluate our models performance?
- After all, we were given the chance to update the model parameters again and again.

Issue with the Splitting!

To fix this issue, data is often split into 3 sets

- **Training Data**

Used to train model parameters

- **Validation Data**

Used to determine what model hyperparameters to adjust

- **Test Data**

Used to get some final performance metric

Issue with the Splitting!

- This means after we see the results on the final test set, we don’t get to go back and adjust any model parameters!
- This final measure is what we label the true performance of the model to be.

Issue with the Splitting!

- In this lesson, we will simplify our data by using a simple **train/test** split.
- We will simply train and then evaluate on a test set (leaving the option to students to go back and adjust parameters)..

Unsupervised Learning

- **Unsupervised learning** is used against data that has no historical labels.
- The system is not told the "right answer." The algorithm must figure out what is being shown.
- The goal is to explore the data and find some structure within.

Unsupervised Learning

- It’s important to note, these are situations where we don’t have the correct answer for historical data!
- Which means evaluation is much harder and more nuanced!

Unsupervised Learning

Evaluating Performance

- We just learned that after our machine learning process is complete, we will use performance metrics to evaluate how our model did.

Evaluating Performance

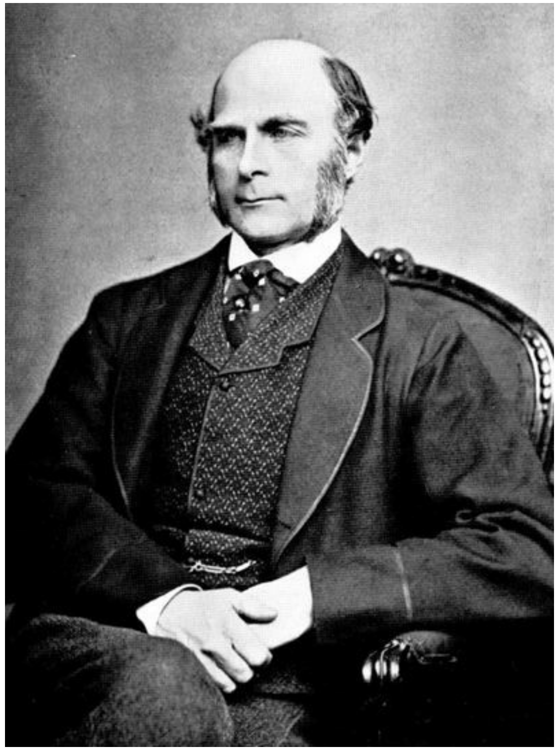
- **Regression** is a task when a model attempts to predict continuous values (unlike categorical values, which is classification)
- For regression problems we need metrics designed for **continuous** values.
- There are also certain evaluation metrics like accuracy or recall that are used for **classification** problems.

Evaluating Performance

- For example, attempting to predict the price of a house given its features is **a regression task**.
- Attempting to predict the country a house is in given its features would be **a classification task**.

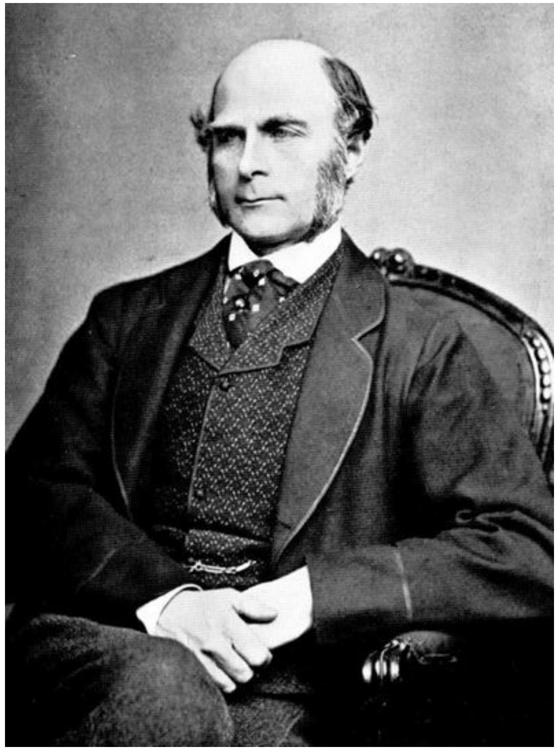
History

- This all started in the 1800s with a guy named Francis Galton. Galton was studying the relationship between parents and their children.
- In particular, he investigated the relationship between the heights of fathers and their sons.



History

- What he discovered was that a man's son tended to be roughly as tall as his father.
- However, Galton's breakthrough was that the son's height tended to be closer to the overall average height of all people.

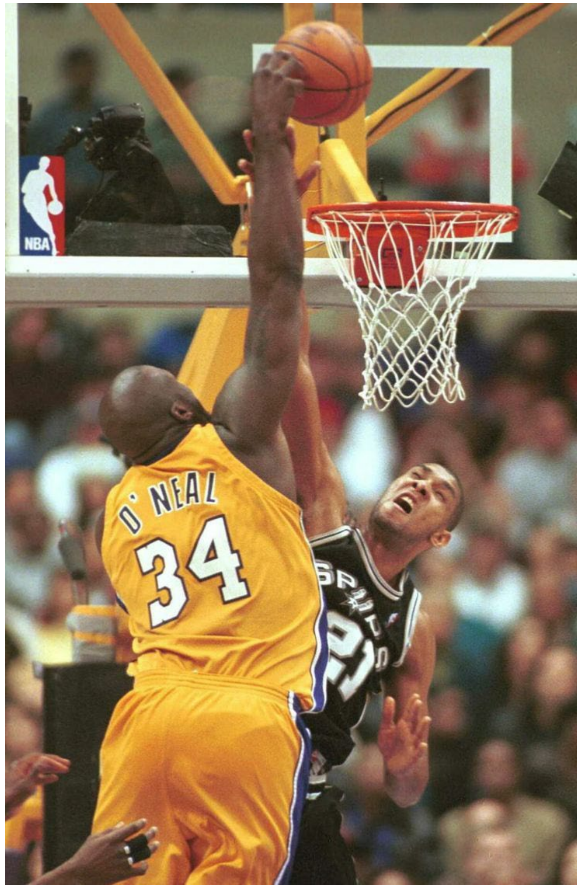


Example

Let's take **Shaquille O'Neal** as an example. Shaq is really tall: 2.2 meters.

If Shaq has a son, chances are he'll be pretty tall too.

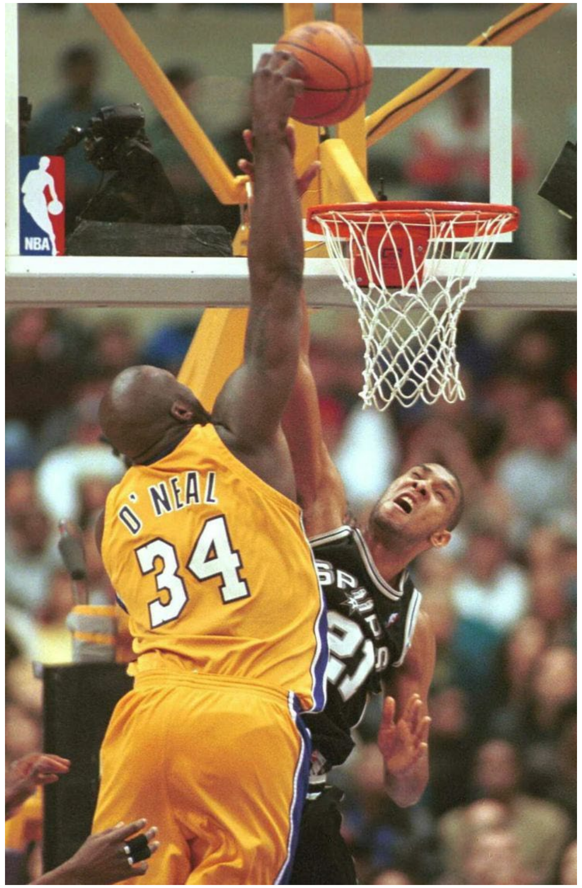
However, Shaq is such an anomaly that there is also a very good chance that his son will be not be as tall as Shaq.



Example

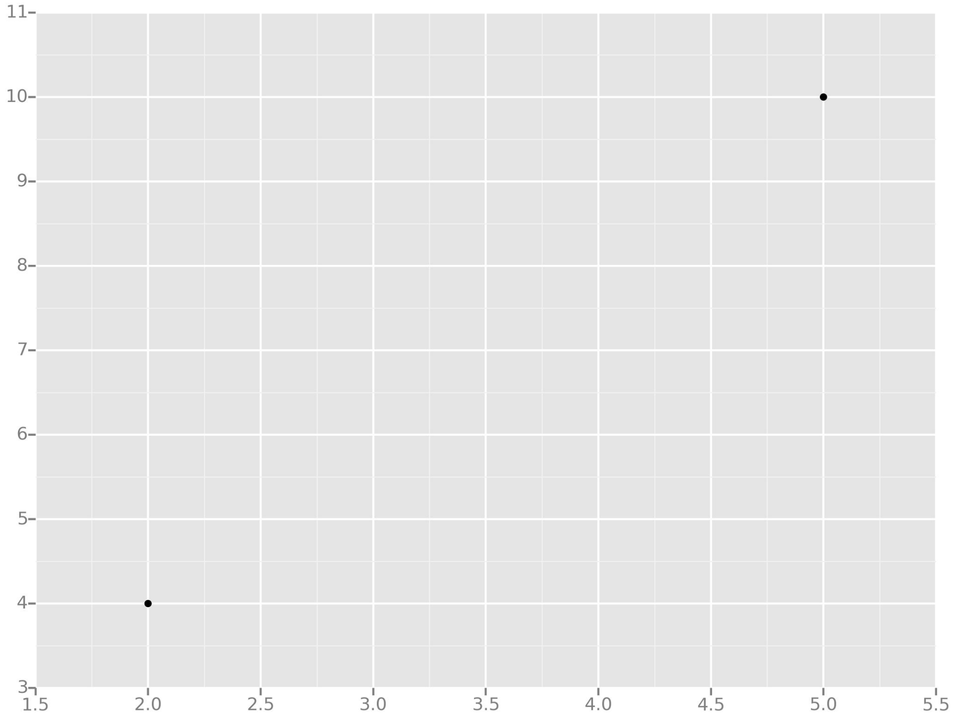
Turns out this is the case: Shaq's son is pretty tall 2.0 meters, but not nearly as tall as his dad.

Galton called this phenomenon regression, as in "A father's son's height tends to regress (or drift towards) the mean (average) height."



Example

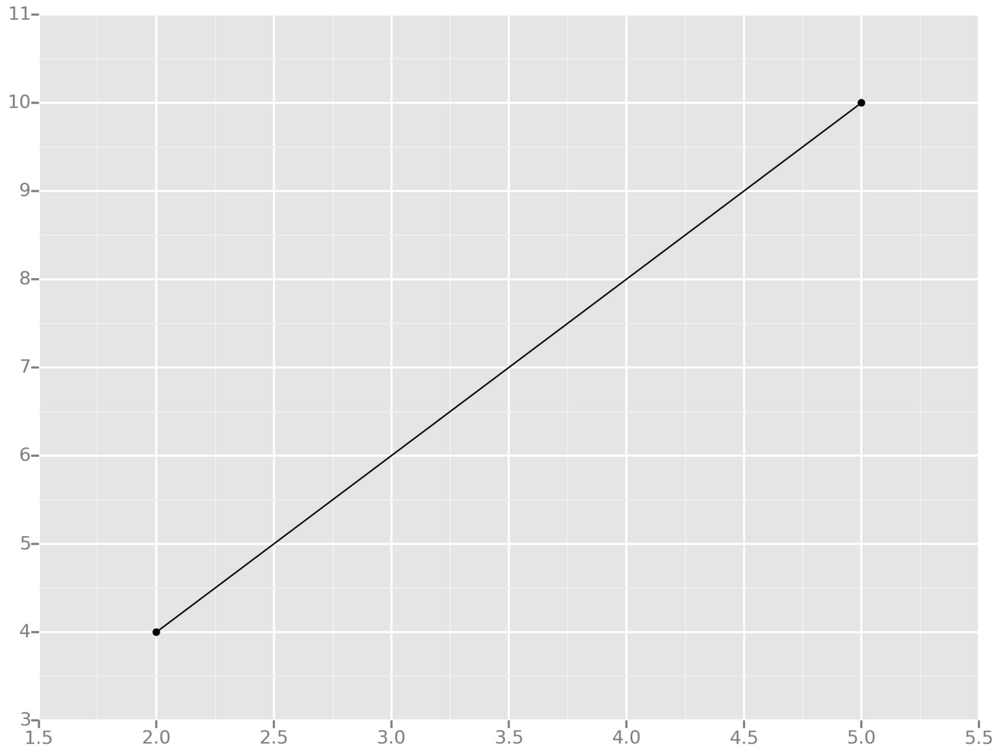
Let's take the simplest possible example: calculating a regression with only 2 data points.



Example

All we're trying to do when we calculate our regression line is draw a line that's as close to every dot as possible.

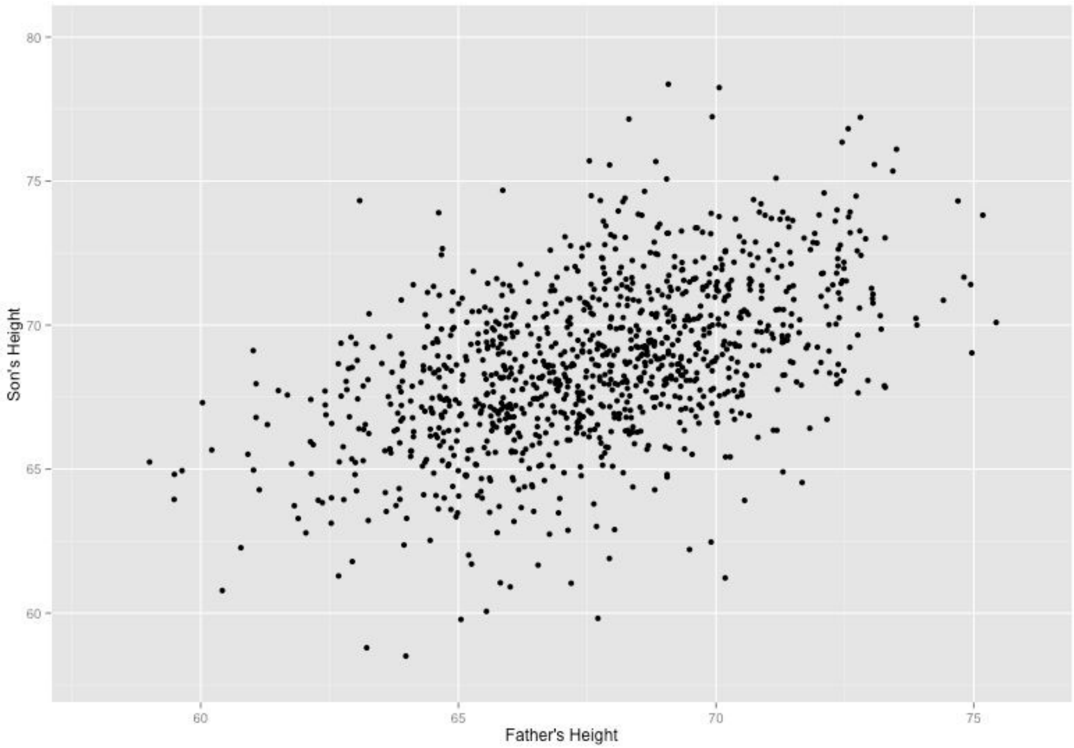
For classic linear regression, or "Least Squares Method", you only measure the closeness in the "up and down" direction



Example

Now wouldn't it be great if we could apply this same concept to a graph with more than just two data points?

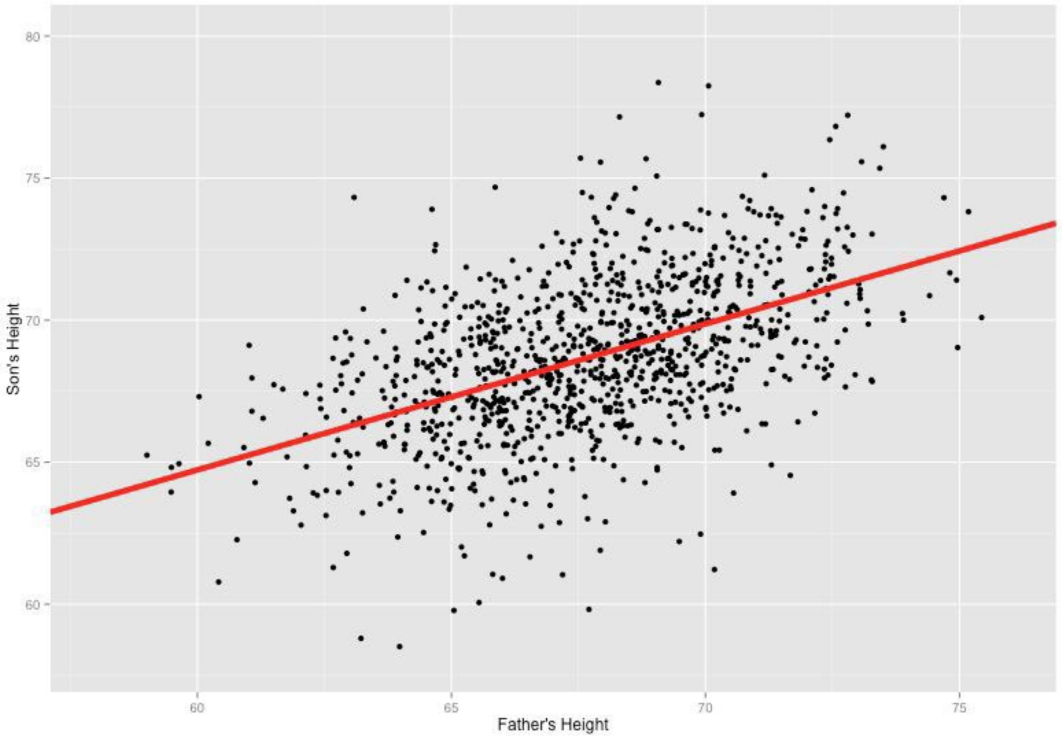
By doing this, we could take multiple men and their son's heights and do things like tell a man how tall we expect his son to be...before he even has a son!



Example

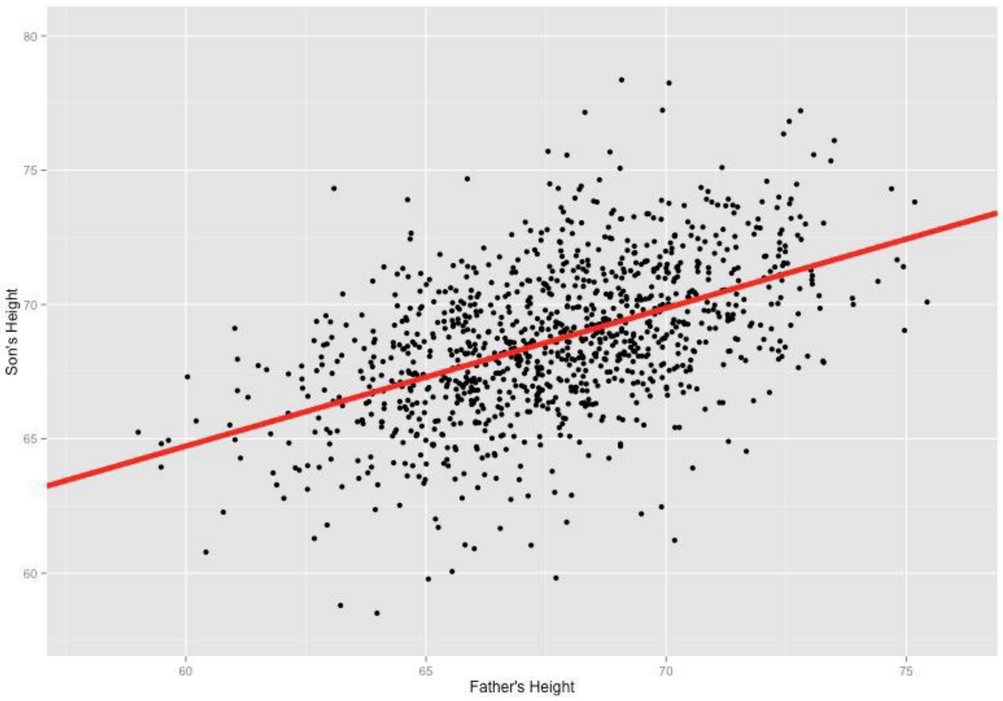
Our goal with linear regression is to minimize the vertical distance between all the data points and our line.

So, in determining the best line, we are attempting to minimize the distance between all the points and their distance to our line.



Example

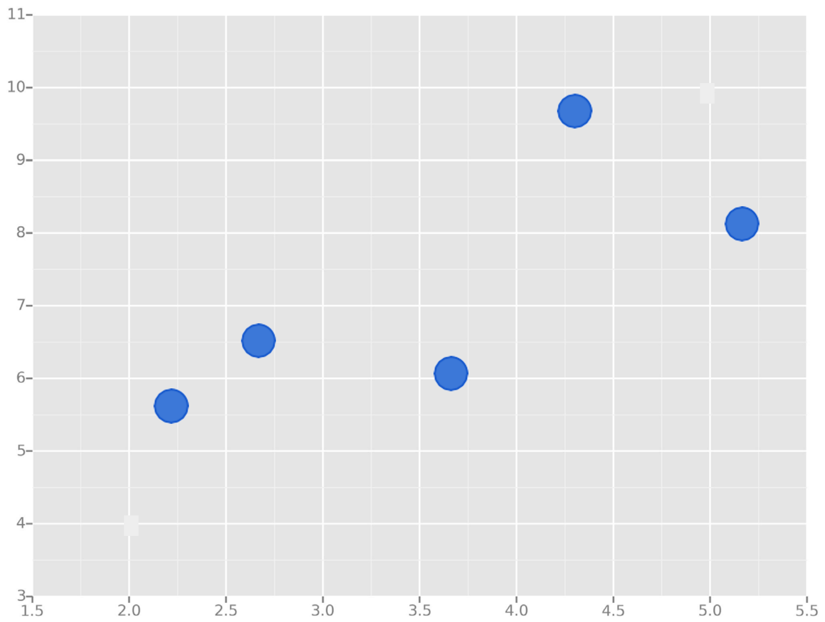
There are lots of different ways to minimize this, (sum of squared errors, sum of absolute errors, etc), but all these methods have a general goal of minimizing this distance.



Example

For example, one of the most popular methods is the least squares method.

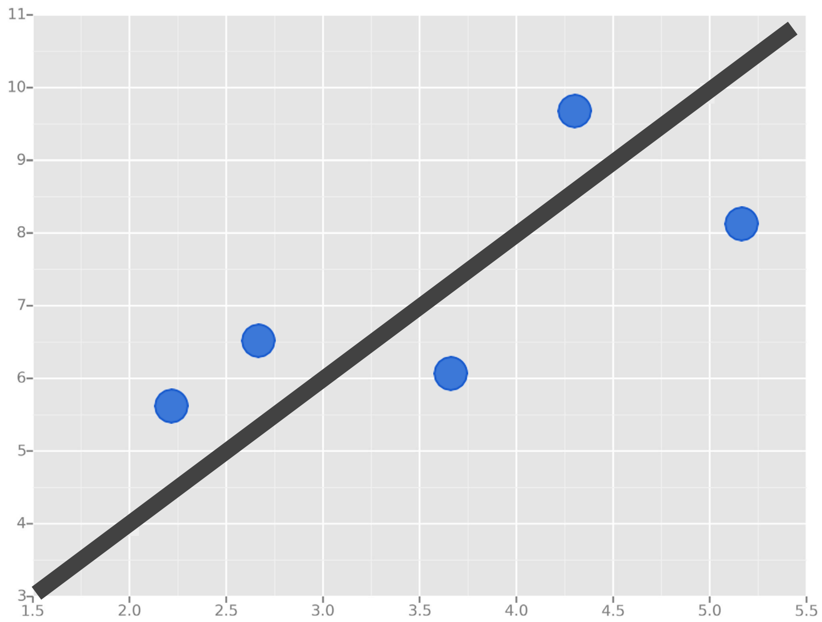
Here we have blue data points along an x and y axis.



Example

Now we want to fit a linear regression line.

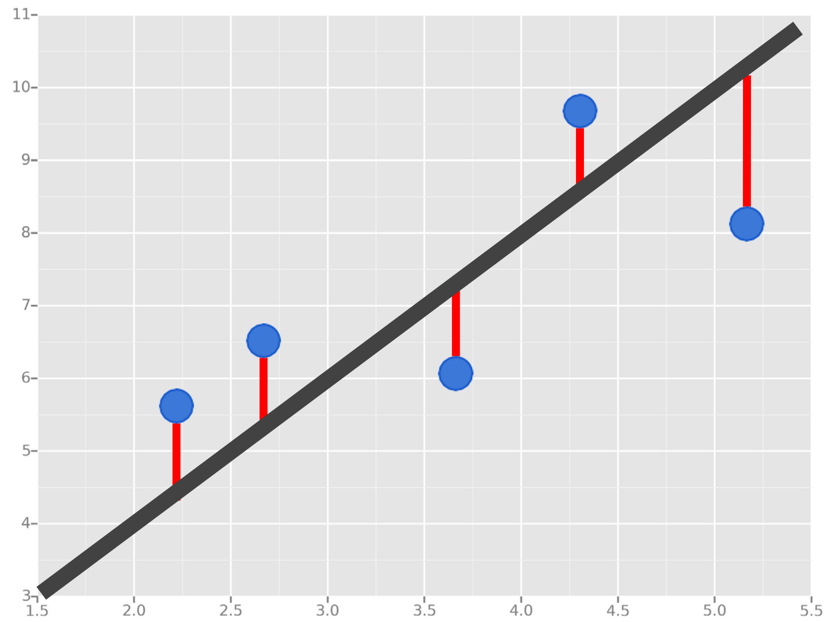
The question is, how do we decide which line is the best fitting one?



Example

We’ll use the Least Squares Method, which is fitted by minimizing the sum of squares of the residuals.

The residuals for an observation is the difference between the observation (the y-value) and the fitted line.



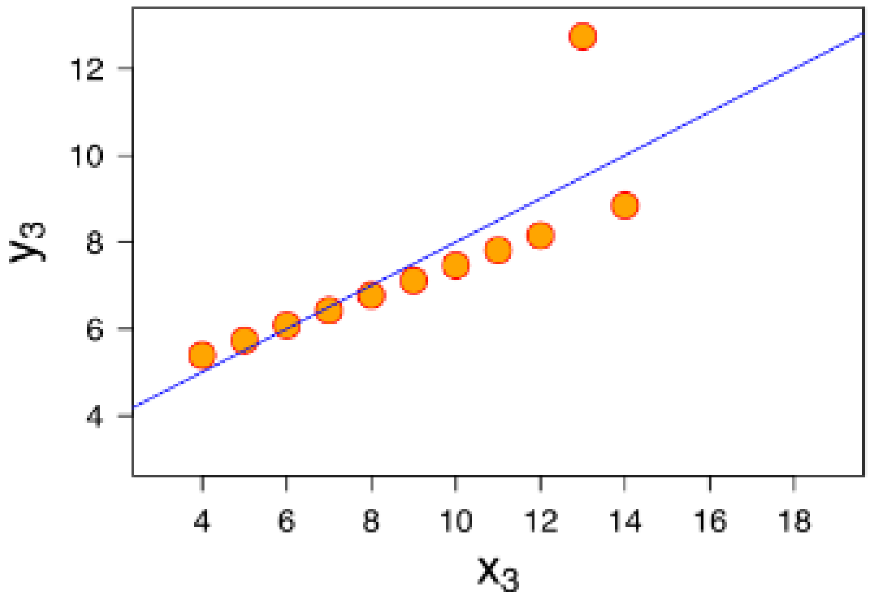
Evaluating Regression: MAE

- Mean Absolute Error (MAE)
  - This is the mean of the absolute value of errors.
  - Easy to understand

$$\overset{\text{number of samples}}{\frac{1}{\boxed{n}}} \underbrace{\sum_{i=1}^{\boxed{n}} \left(\overset{\text{real value}}{\boxed{Y_i}} - \overset{\text{predicted value}}{\boxed{\hat{Y}_i}}\right)^2}_{\text{sum of the errors of all samples}}$$

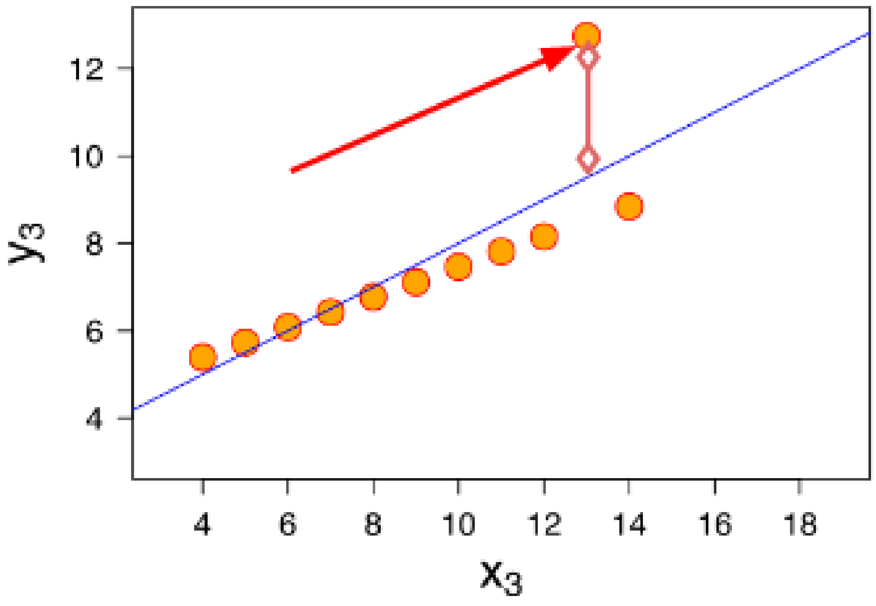
Evaluating Regression: MAE

MAE, however, won't
punish large errors.



Evaluating Regression: MAE

MAE, however, won't punish large errors.



Evaluating Regression: MSE

- Mean Squared Error (MSE)
- This is the mean of the squared errors.
- Larger errors are noted more than with MAE, making MSE more popular.

$$\mathrm{MSE} = \overset{\text{Mean}}{\boxed{\frac{1}{n}\sum_{i=1}^{n}}}\left(\overset{\text{Error}}{\boxed{Y_i - \hat{Y}_i}}\right)^{\overset{\text{Squared}}{\boxed{2}}}$$

$$\underbrace{\overset{\text{real value}}{\boxed{Y_i}} - \overset{\text{predicted value}}{\boxed{\hat{Y}_i}}}_{\text{error}}$$

Evaluating Regression: RMSE

- Root Mean Square Error (RMSE)
- This is the root of the mean of the squared errors.
- Most popular (has same units as y)

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Evaluating Regression: RMSE

Most common question from students:

- “Is this value of RMSE good?”
- Context is everything!
- A RMSE of $10 is fantastic for predicting the price of a house, but horrible for predicting the price of a candy bar!

Evaluating Regression: RMSE

- Compare your error metric to the average value of the label in your data set to try to get an intuition of its overall performance.
- Domain knowledge also plays an important role here!
- Context of importance is also necessary to consider.
- We may create a model to predict how much medication to give, in which case small fluctuations in RMSE may actually be very significant.

Course Agenda

- **Machine Learning**
  - What is ML
  - Neural Networks
  - Deep Learning
  - Supervised Learning Process
  - Unsupervised Learning Process
- **Linear Regression**
  - History
  - Example
  - Evaluating Performance
  - Example in Python