# Analytics Day 2b

## Advanced Analytics

# Advanced Analytics Methods

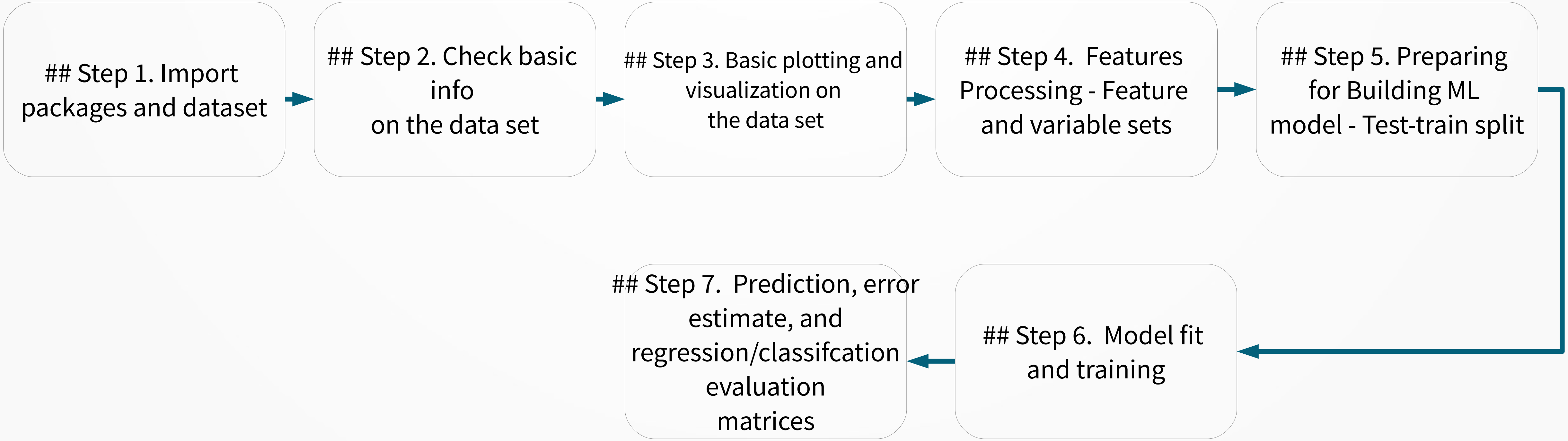
- Regression and classification are both related to prediction, where **regression predicts a value from a continuous set, whereas classification predicts the 'belonging' to the class.**
  - For example, the **price of a house** depending on the *'size'* (in some unit) and say *'location'* of the house, can be some 'numerical value' (which can be continuous): this relates to regression.
- Similarly, **the prediction of price can be in words, viz., 'very costly', 'costly', 'affordable', 'cheap', and 'very cheap**': this relates to **classification**.
- Each class may correspond to some range of values.

# Advanced Analytics Methods

- Classifcation:
- Classification algorithms are used when the desired output is a discrete label. In other words, they're helpful when the answer to your question about your business falls under a finite set of possible outcomes. Many use cases, such as **determining whether an email is spam or not, have only two possible outcomes. This is called binary classification.**
- Multi-label classification captures everything else, and is useful for customer segmentation, audio and image categorization, and text analysis for mining customer sentiment. If these are the questions you're hoping to answer with machine learning in your business, **consider algorithms like naive Bayes, decision trees, logistic regression, kernel approximation, and K-nearest neighbors**.

# Advanced Analytics Methods

- Classifcation: The flow of building a classification ML model



## Step 1. Import packages and dataset

## Step 2. Check basic info on the data set

## Step 3. Basic plotting and visualization on the data set

## Step 4. Features Processing - Feature and variable sets

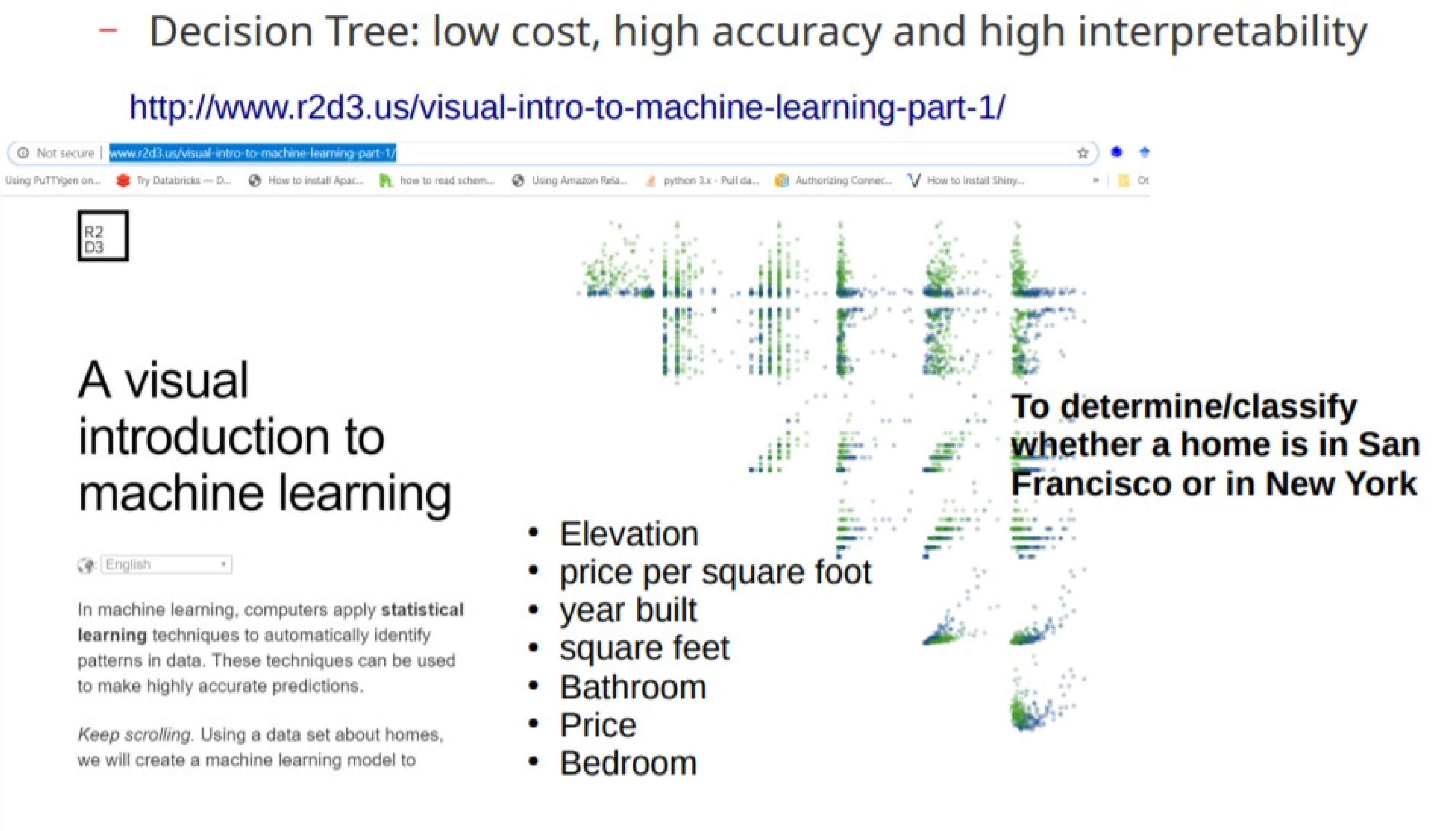
## Step 5. Preparing for Building ML model - Test-train split

## Step 6. Model fit and training

## Step 7. Prediction, error estimate, and regression/classifcation evaluation matrices

# Classification

- **Decision Tree**
- http://www.r2d3.us/visual-intro-to-machine-learning-part-1/
- http://www.r2d3.us/visual-intro-to-machine-learning-part-2/



– Decision Tree: low cost, high accuracy and high interpretability

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

A visual introduction to machine learning

In machine learning, computers apply **statistical learning** techniques to automatically identify patterns in data. These techniques can be used to make highly accurate predictions.

*Keep scrolling.* Using a data set about homes, we will create a machine learning model to

- Elevation
- price per square foot
- year built
- square feet
- Bathroom
- Price
- Bedroom

**To determine/classify whether a home is in San Francisco or in New York**

# Classification

- **K Nearest Neighbor**
- In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the **k closest training examples** in the feature space. The output depends on whether k-NN is used for classification or regression:
- In ***k-NN classification***, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the **object being assigned to the class most common among its k nearest neighbors** (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.
- In ***k-NN regression***, the output is the property value for the object. This value is the average of the values of k nearest neighbors.
- k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification.
- https://yihui.name/animation/example/knn-ani/

# Advanced Analytics Methods

- **Regression:**
- **Regression is useful for predicting outputs that are continuous**. That means the answer to your question is represented by a quantity that can be flexibly determined based on the inputs of the model rather than being confined to a set of possible labels. **Regression problems with time-ordered inputs are called time-series forecasting problems, like ARIMA forecasting**, which allows data scientists to explain seasonal patterns in sales, evaluate the impact of new marketing campaigns, and more.
- Linear regression is by far the most popular example of a regression algorithm. Though it's often underrated because of its relative simplicity, it's a versatile method that can be **used to predict housing prices, likelihood of customers to churn, or the revenue a customer will generate**. For use cases like these, regression trees and support vector regression are good algorithms to consider if you're looking for something more sophisticated than linear regression.

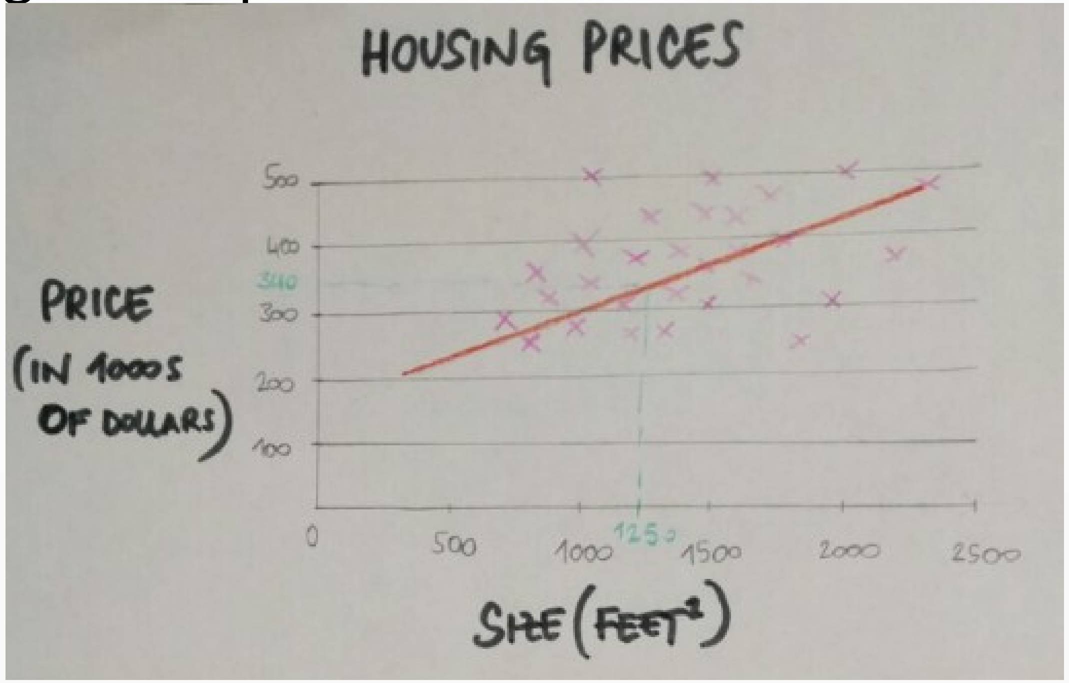
# Regression

- **Linear Regression**
  - Linear regression with one variable
  - Univariate linear regression is used when you want to predict a single output value from a single input value. We're doing supervised learning here, so that means we already have an idea what the input/output cause and effect should be.
  - For example, let's take the housing prices with different house sizes (and different prices). A friend of mine wants to sell his house that is 1250 feet^2 and he wants to know the approximate value of it.

http://www.battaly.com/stat/geogebra/linearregression/

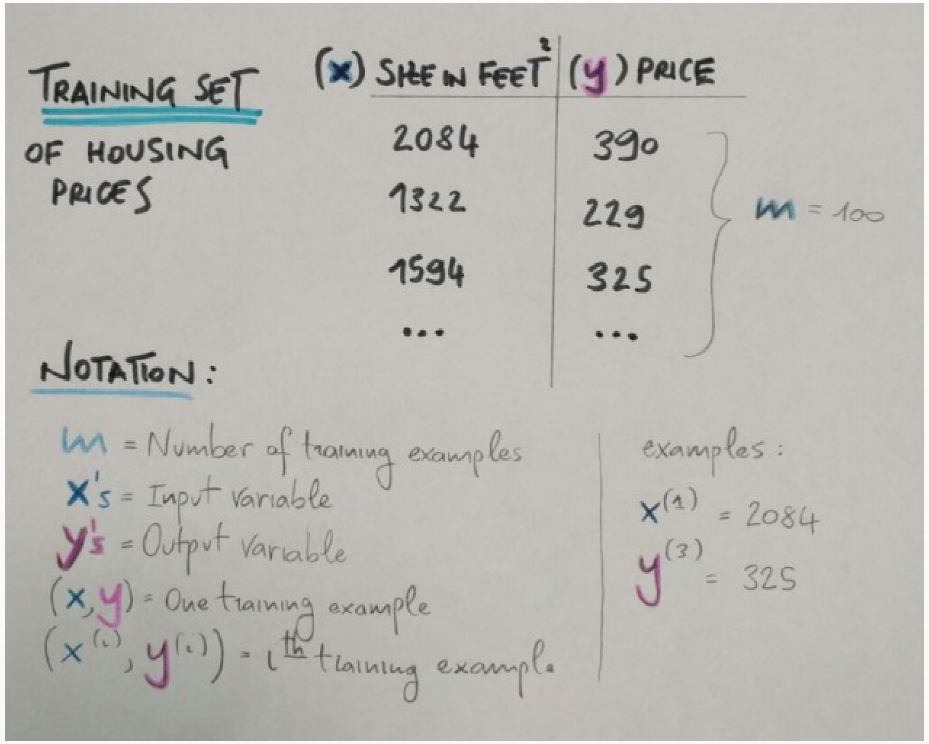
# Regression

- Linear Regression
  - Linear regression with one variable
  - The idea is that you plot a line that best fits the data and based on that I can tell my friend that the house is worth 340k. This is an example of supervised learning and regression problem.



# Regression

- Linear Regression
  - Linear regression with one variable
  - More formally, in supervised learning we have a data set called training set with the house prices and sizes. From this dataset I want to learn how to predict the house prices.



TRAINING SET OF HOUSING PRICES

| (x) SIZE IN FEET$^2$ | (y) PRICE |
|---|---|
| 2084 | 390 |
| 1322 | 229 |
| 1594 | 325 |
| ... | ... |

$m = 100$

NOTATION:

$m$ = Number of training examples<br>
$x$'s = Input variable<br>
$y$'s = Output variable<br>
$(x, y)$ = One training example<br>
$(x^{(i)}, y^{(i)})$ = $i^{th}$ training example

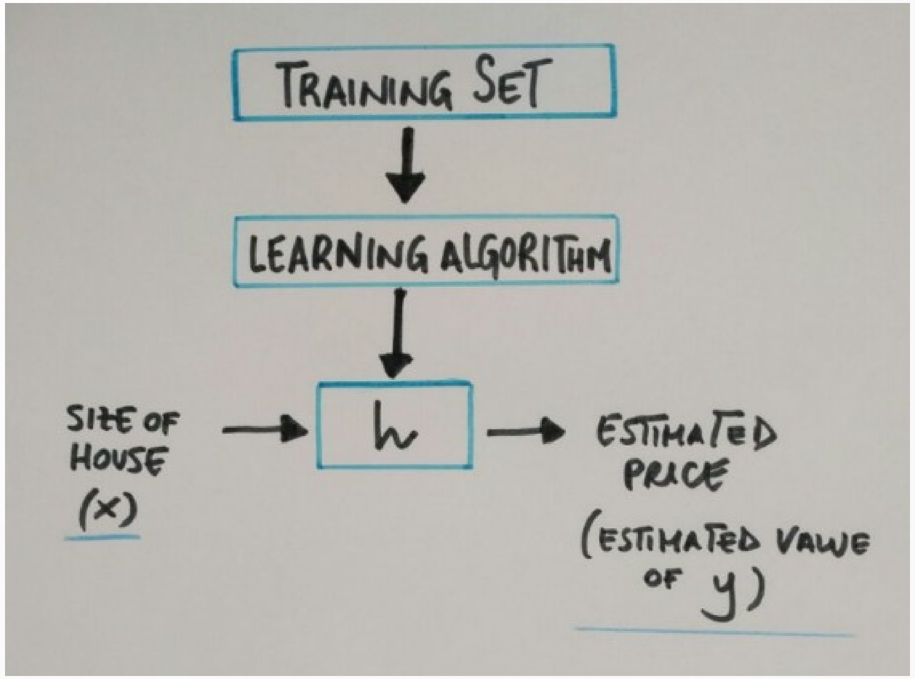
examples:

$x^{(1)} = 2084$

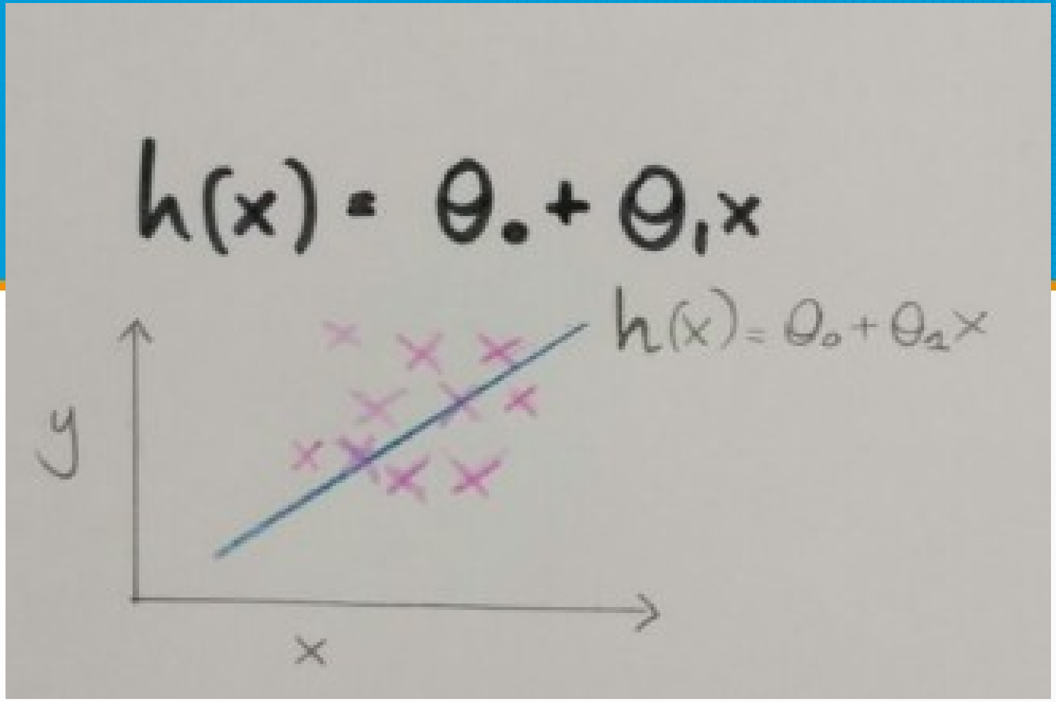
$y^{(3)} = 325$

# Regression

- Linear Regression
  - m = the number of training examples (number of rows)
  - x = input
  - y = output
  - (x,y) = one training example (one single row)



# Regression


$h(x) = \theta_0 + \theta_1 x$
$h(x) = \theta_0 + \theta_1 x$
y
x

- Linear Regression
  - general supervised learning needs:
  - a training set (house prices by sizes) which we fit to our learning algorithm. As an output we get h (hypothesis) which is a function that takes the size of houses as an input, and tries to output the estimate value of y.
  - h is a function that maps from x to y.
  - So how do we represent h?
  - The function is predicting the value of y, given the value of x.
  - This model is called linear regression, better known as univariate linear regression because we have just one variable.