

Introduction to *Urban* Data Science

Machine Learning for Everyone

(EPA1316)

Lecture 9

Trivik Verma

THIS IS YOUR MACHINE LEARNING SYSTEM?

YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.

WHAT IF THE ANSWERS ARE WRONG?

JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.



Groups (28/09-16/10)

Form groups of **4 students** each. Each student needs to be part of a group. If you are going to drop the course, don't sign up for the final project as that may delay the progress of other students.

- If you have decided on a group of 4 and your partners haven't joined your group yet, do it right away
- If you haven't found a group, join a group that has 2 or 3 people
- If you are in a group of 1 or 2, leave and join other groups that are still open
- Who wants a group of 3 or 5? Email me. Good reasoning handy.

Last Time

- Exploratory Spatial Data Analysis (ESDA)
- Spatial Autocorrelation Measures
 - Global
 - Local

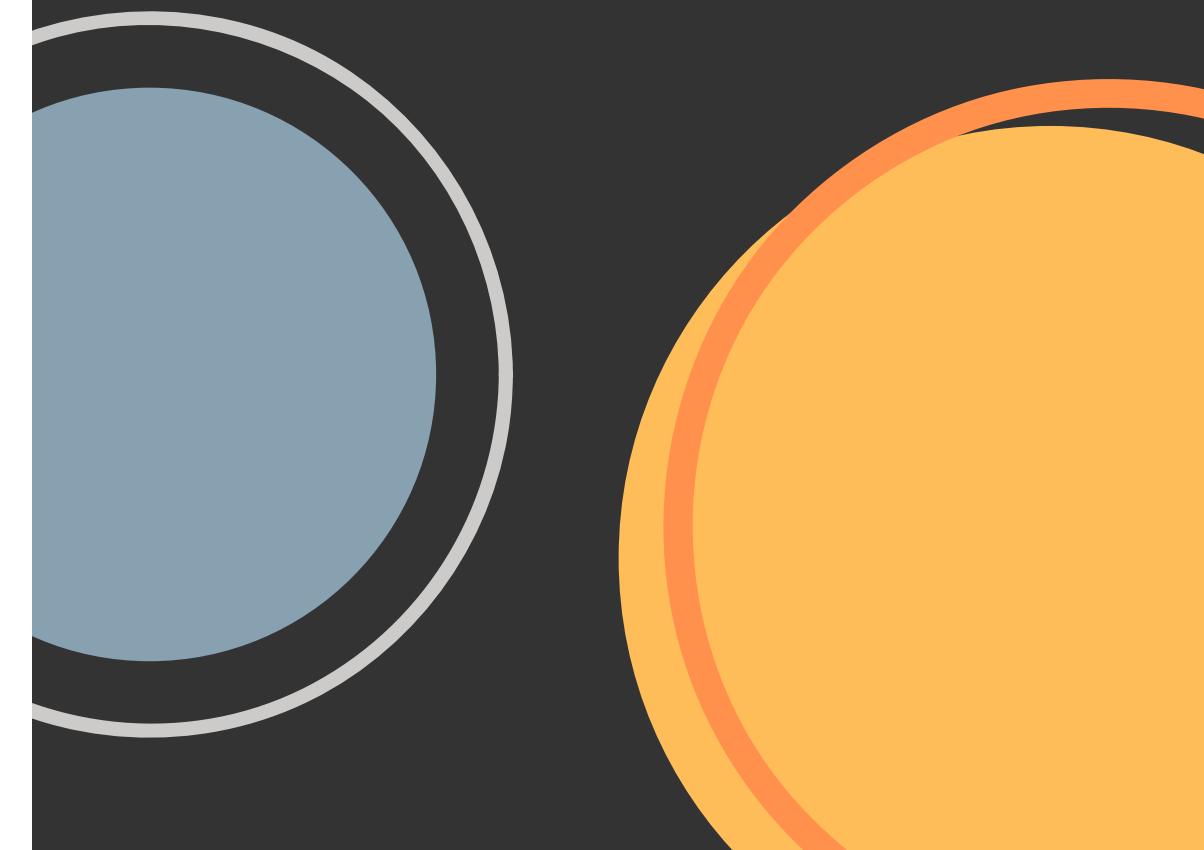
Today

- Machine Learning
- Predicting a Variable
- Error evaluation
- Model comparison
- Fitness of models

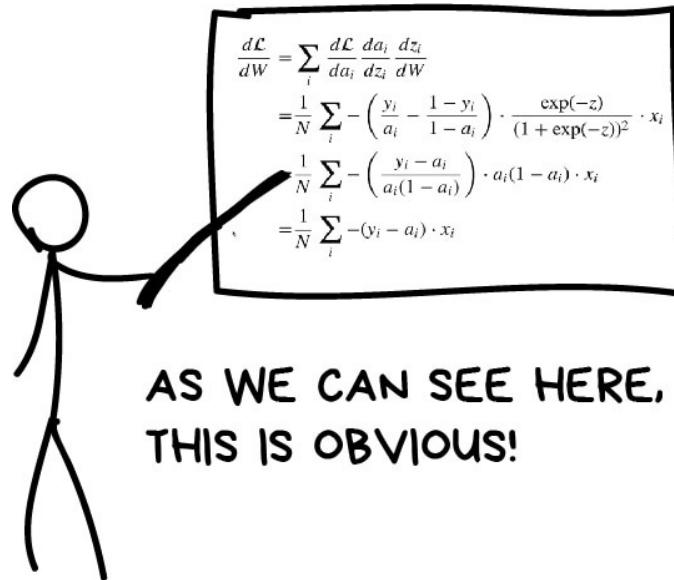
A beginner in machine learning will find in this book just enough details to get a comfortable level of understanding of the field and start asking the right questions.

Andriy Burkov's

THE HUNDRED-PAGE MACHINE LEARNING BOOK



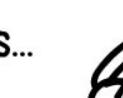
Let's start with the Truth
Machines don't learn

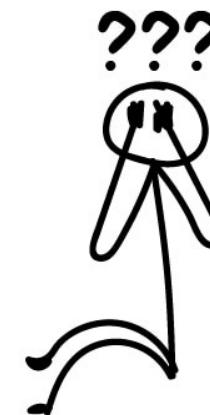


AS WE CAN SEE HERE
THIS IS OBVIOUS!

$$\begin{aligned}\frac{d\mathcal{L}}{dW} &= \sum_i \frac{d\mathcal{L}}{da_i} \frac{da_i}{dz_i} \frac{dz_i}{dW} \\&= \frac{1}{N} \sum_i -\left(\frac{y_i}{a_i} - \frac{1-y_i}{1-a_i} \right) \cdot \frac{\exp(-z)}{(1+\exp(-z))^2} \cdot x_i \\&\quad \cancel{\frac{1}{N} \sum_i -\left(\frac{y_i - a_i}{a_i(1-a_i)} \right) \cdot a_i(1-a_i) \cdot x_i} \\&= \frac{1}{N} \sum_i -(y_i - a_i) \cdot x_i\end{aligned}$$

PROGRAMMERS ARE PROGRAMMING!
DATASCIENCE!
PROFESSION OF FUTURE!
IN THE NEXT FIVE YEARS...
EXPONENTIAL GROWTH!!!
SMART MACHINES!
A-A-A-A-A-A-A-A-A-AAA!!!!!!





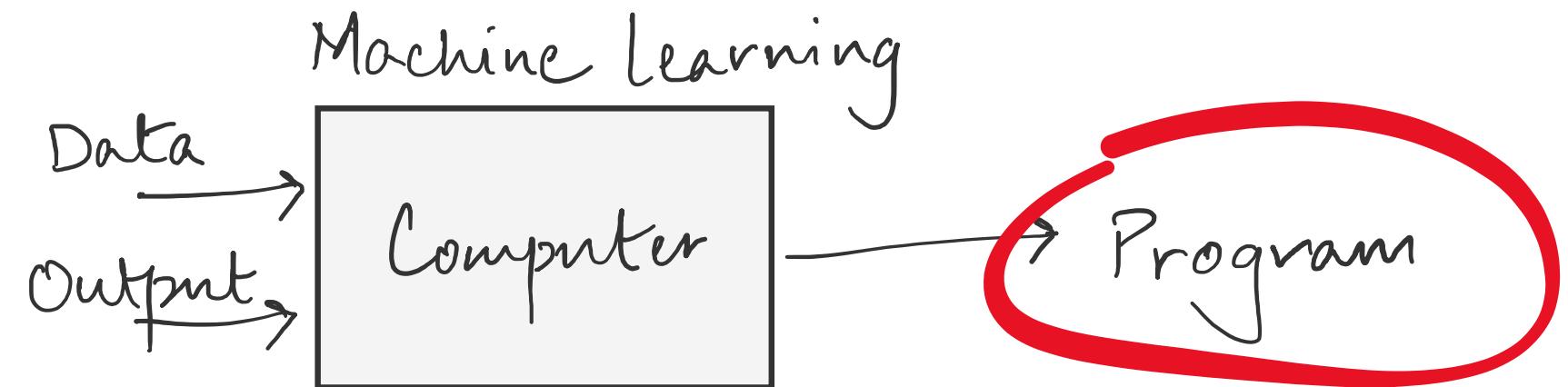
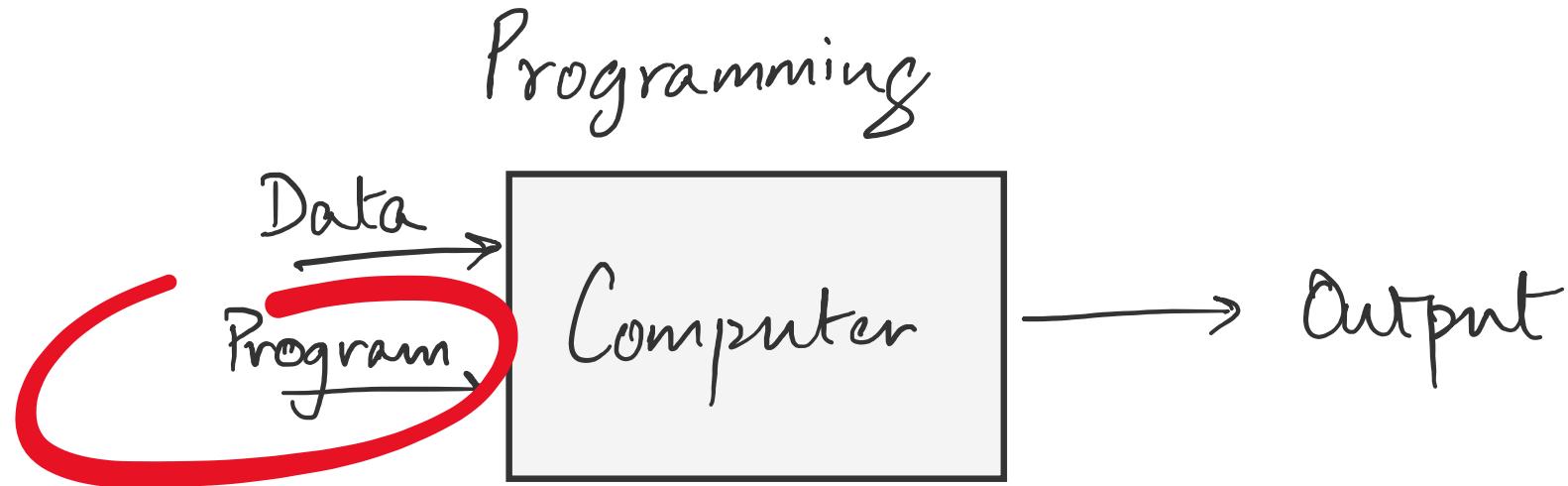
TWO TYPES OF ARTICLES ABOUT MACHINE LEARNING

In this course: Only real-world problems, practical solutions, simple language, and no high-level theorems

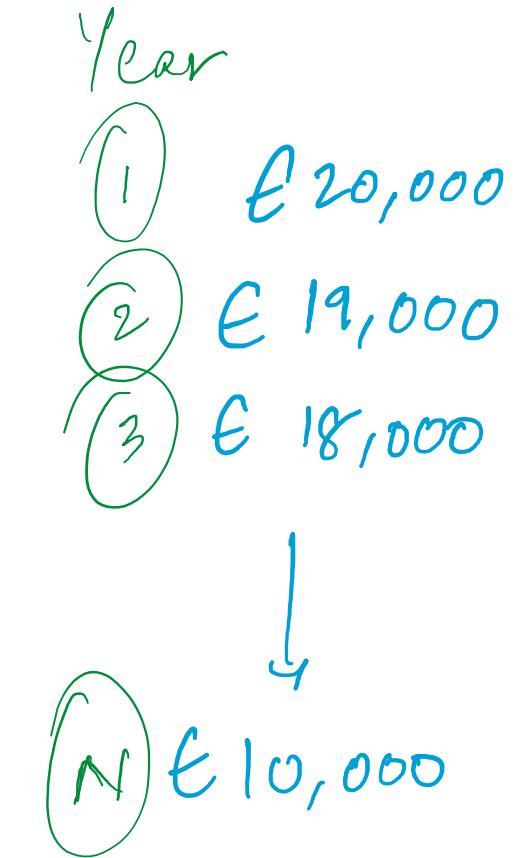
Arthur Samuel (1959)



“Field of study that gives computers the ability to learn without being explicitly programmed.”



Why do we want machines to learn?



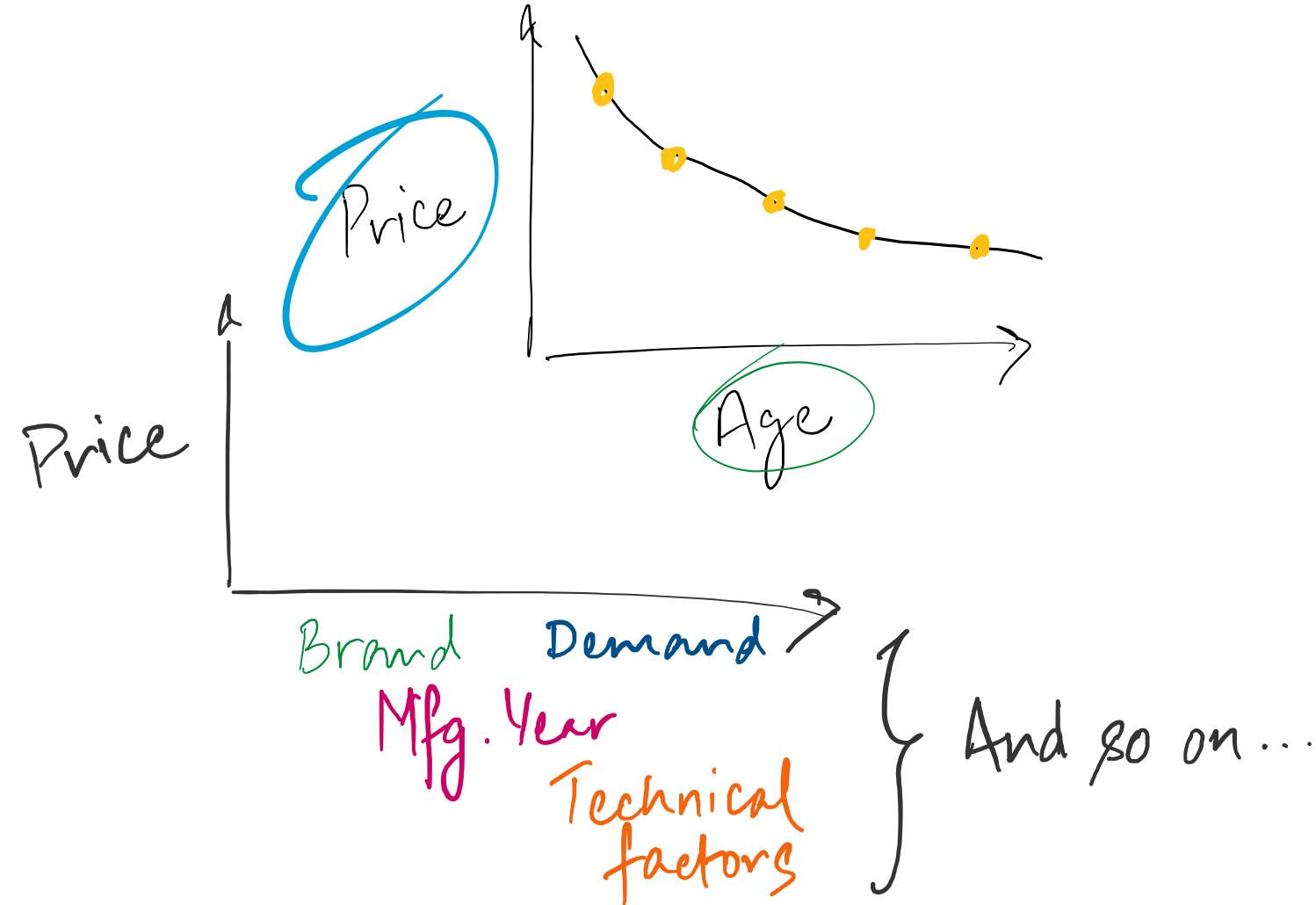
Price & Age

Why do we want machines to learn?



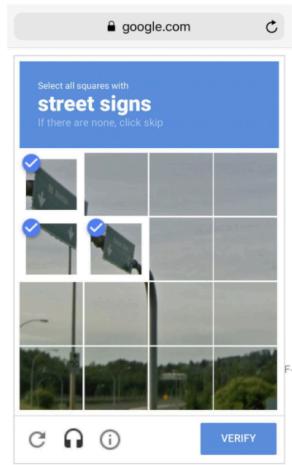
Year	Price
0	€ 20,000
1	€ 19,000
2	€ 18,000
N	€ 10,000

Price \propto Age



The goal of ML algorithms is to predict results based on incoming data. That's it!

What do we need to teach machines?



DATA



Crime
Poverty
Weather
Stocks
Socio-economic condition

Features

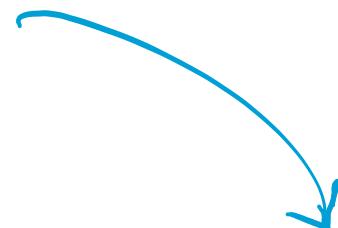
Income
Amenities
Gender
Stock price
Travel info
Family info

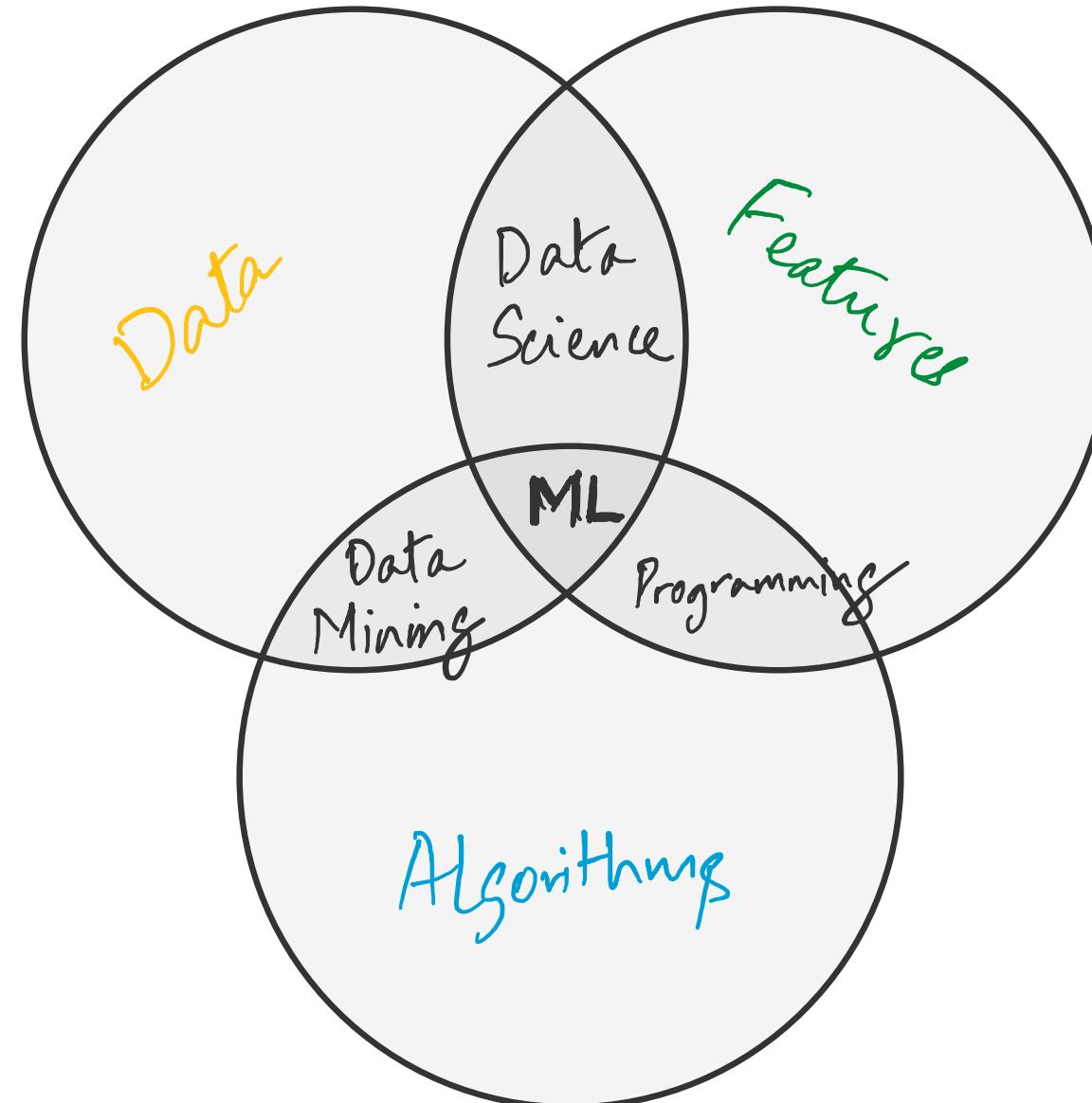


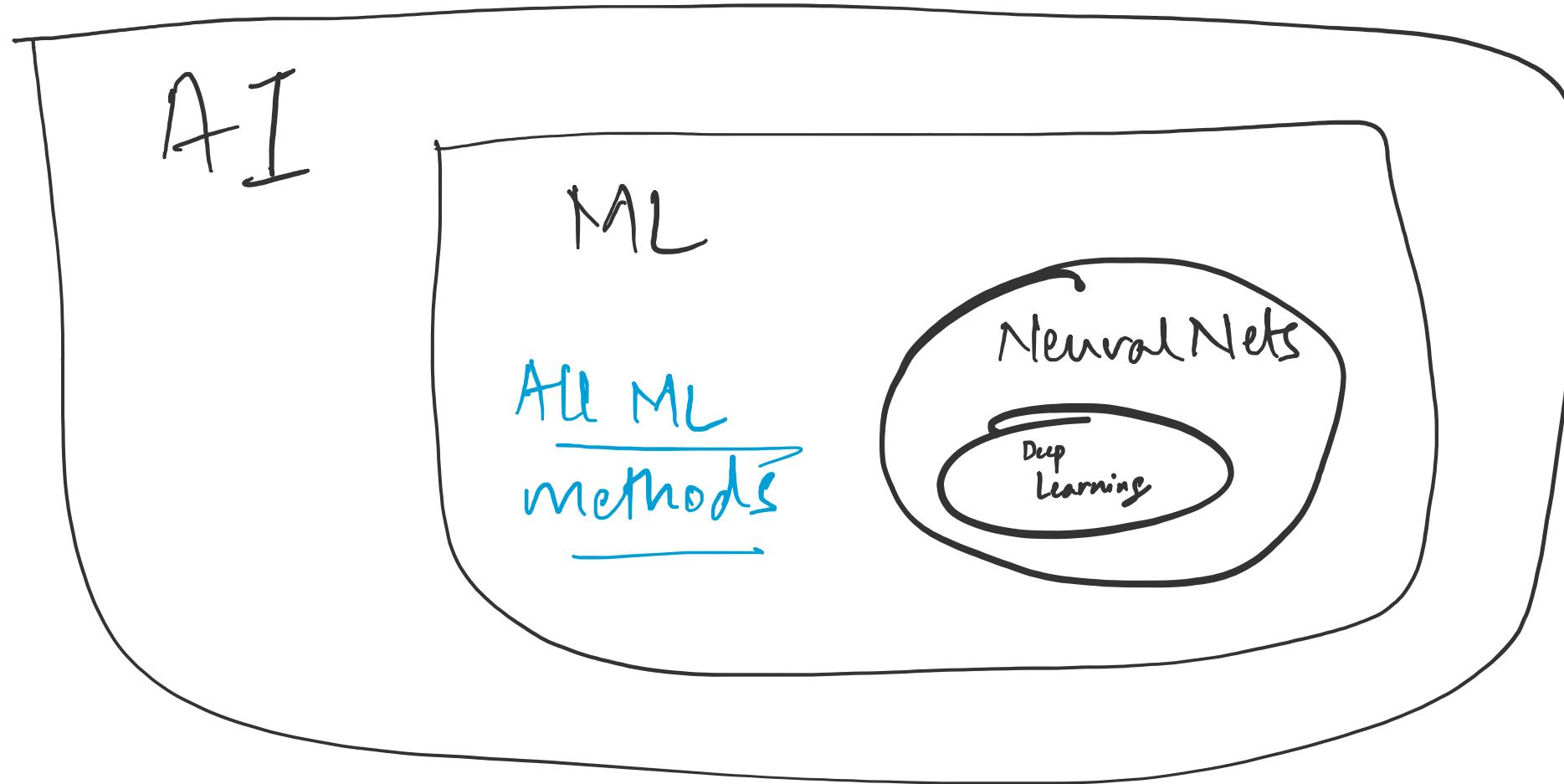
Algorithms

- precision
- performance
- size

[garbage in - garbage out]







AI : Field like

physics
chemistry
Math

ML : Part of AI / Not the only one

NN : Type of ML / One of the good guys.

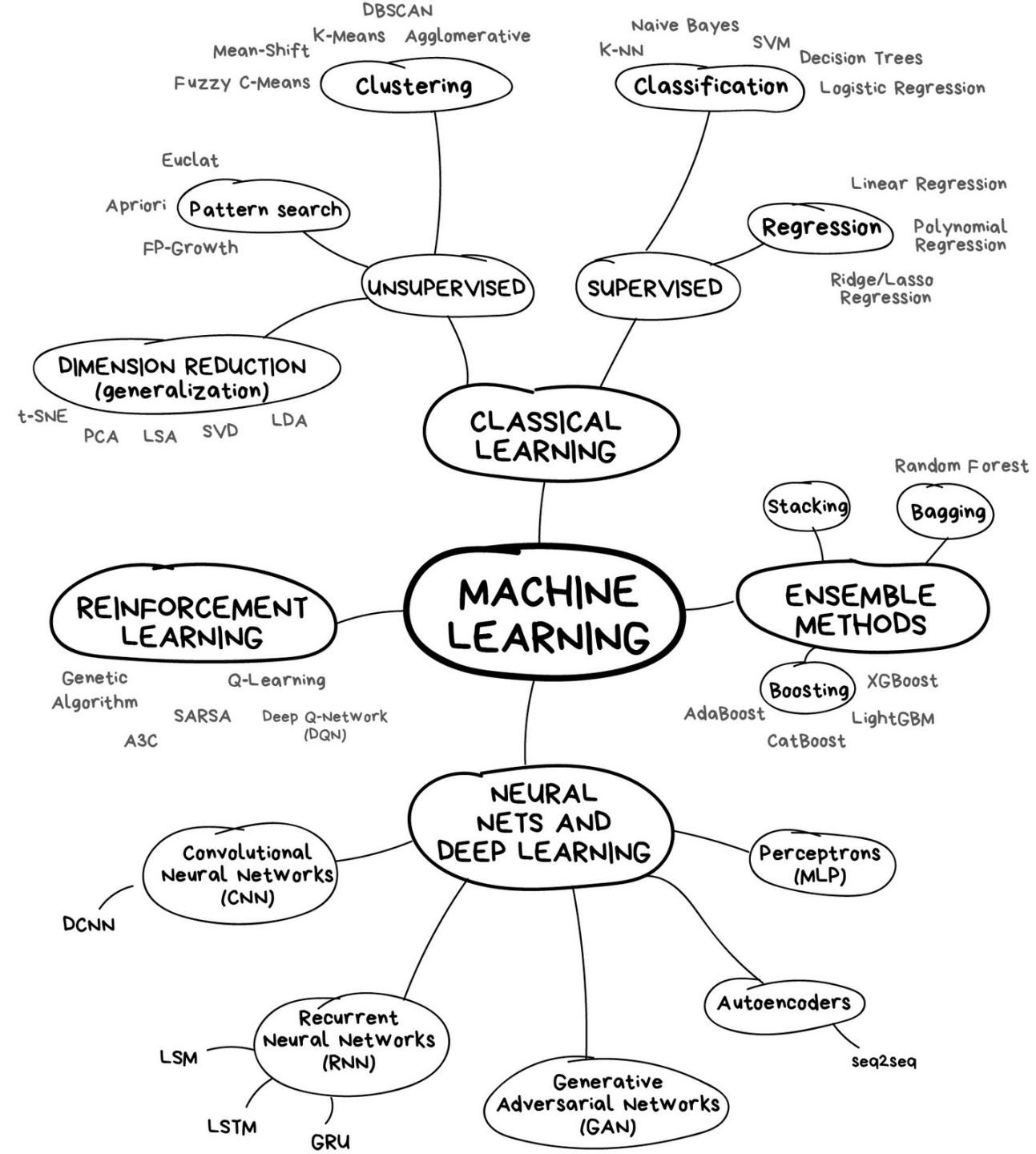
Deep Learning : Basically a new architecture for NN.

Machines CAN

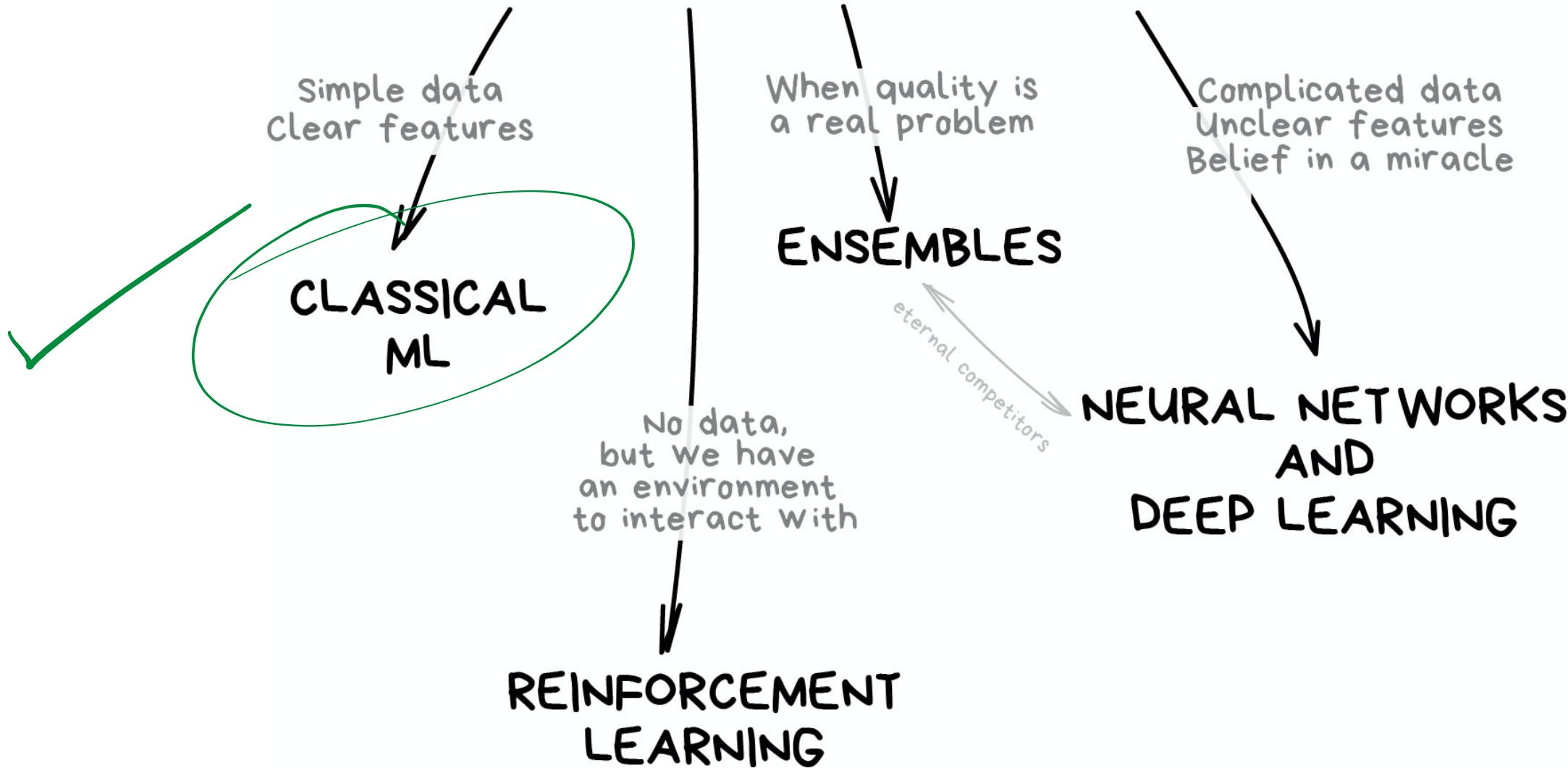
- Forecast
- Memorise
- Reproduce
- Choose best object

Machines CANNOT

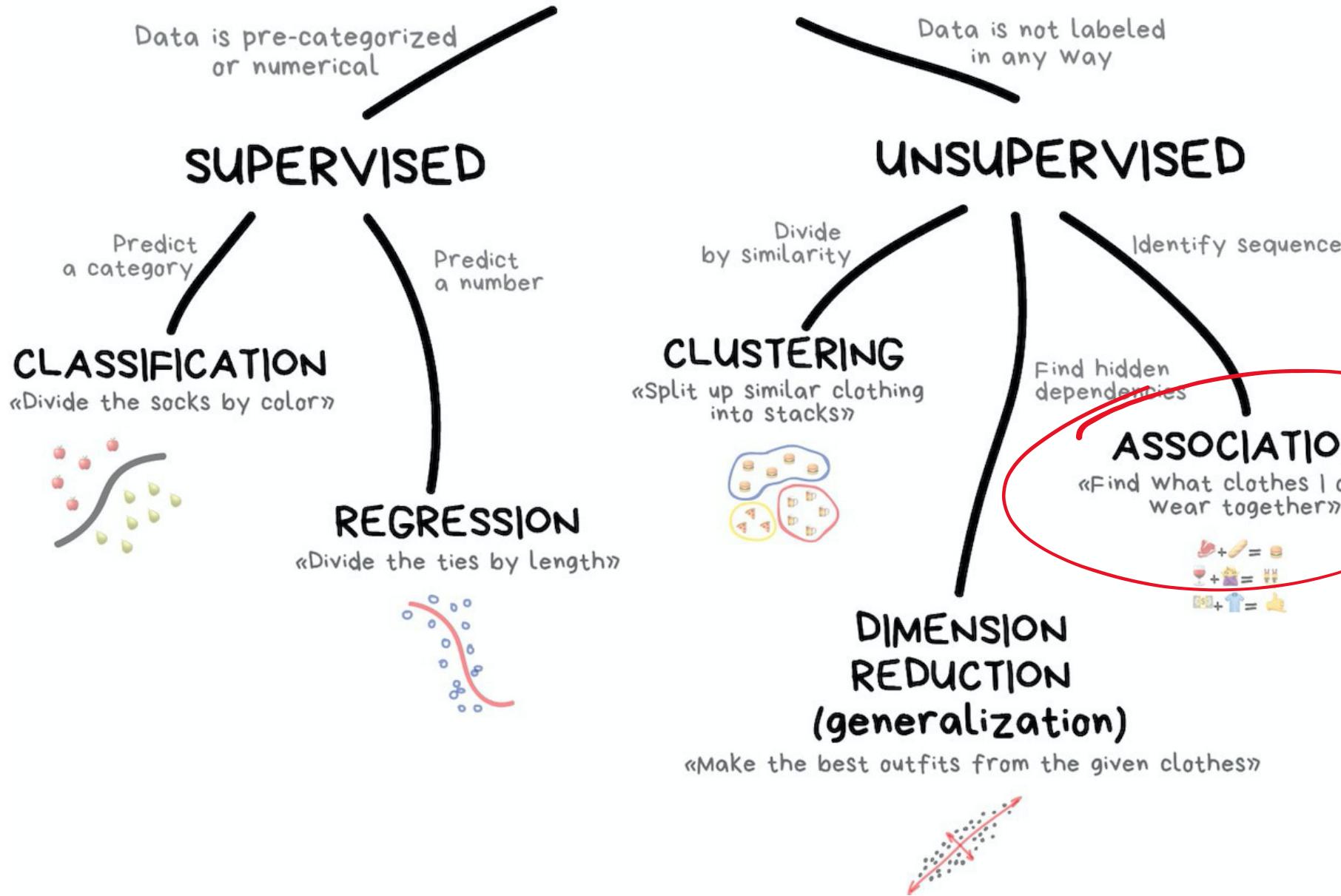
- Create / Innovate
- Get Smart
- Go beyond the task
- Take over the world

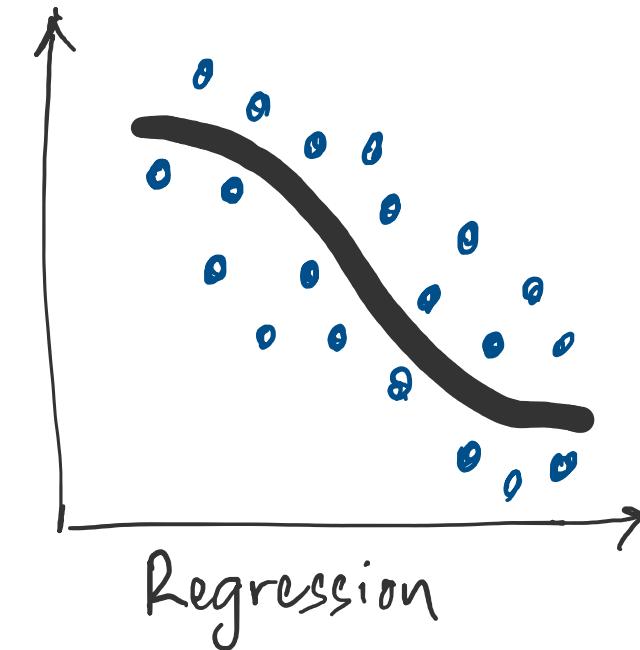
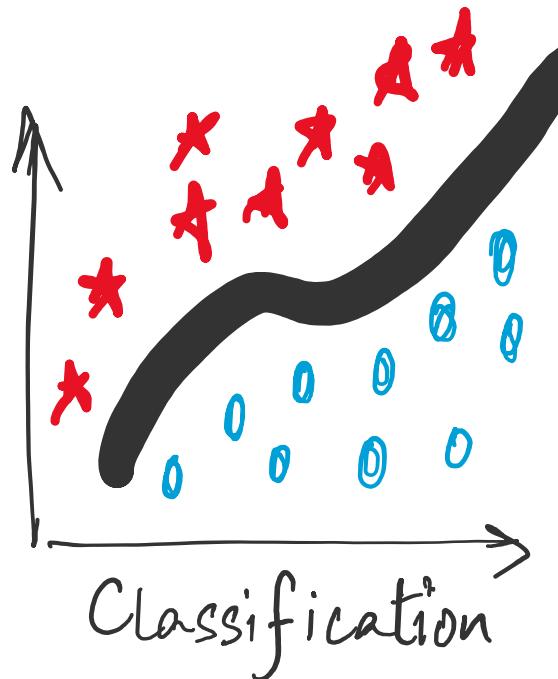


THE MAIN TYPES OF MACHINE LEARNING



CLASSICAL MACHINE LEARNING





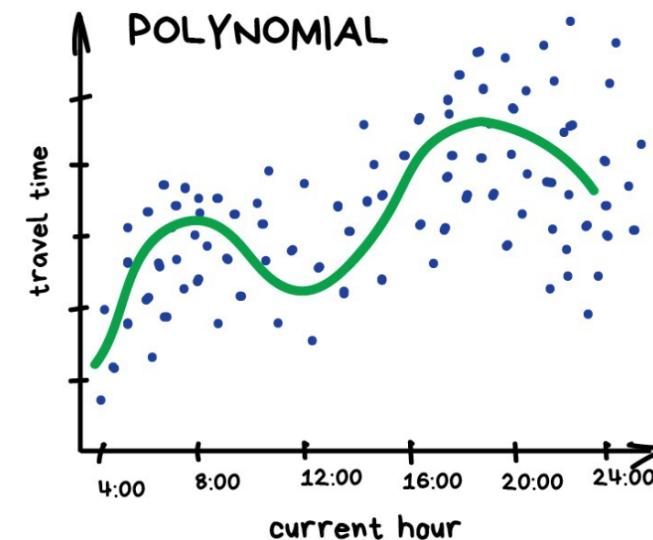
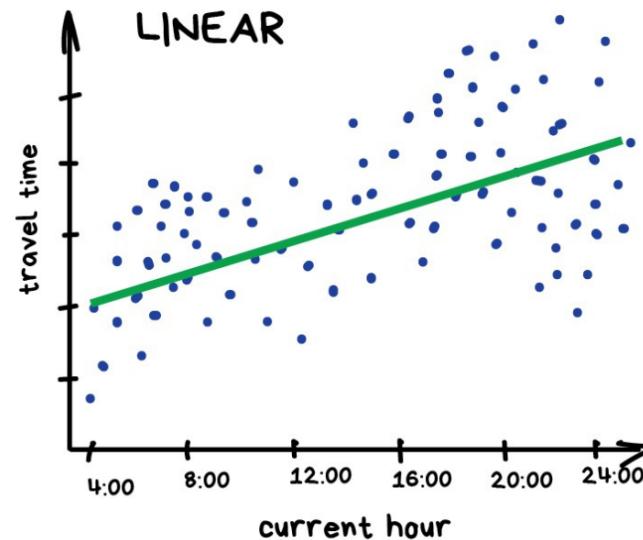
Today used for:

- Spam filtering
- Language detection
- A search of similar documents
- Sentiment analysis
- Recognition of handwritten characters and numbers
- Fraud detection
- Users based on interests (as algorithmic feeds do)

Today this is used for:

- Stock price forecasts
- Demand and sales volume analysis
- Medical diagnosis
- Any number-time correlations
- Car price by its mileage
- Traffic by time of the day

PREDICT TRAFFIC JAMS



REGRESSION

When the line is straight — it's a linear regression, when it's curved — polynomial. These are two major types of regression. The other ones are more exotic. Logistic regression is a [different](#) sheep in the flock. Don't let it trick you, as it's a classification method, not regression.

Predicting a variable

Let's imagine a scenario where we'd like to predict one variable using another (or a set of other) variables.

Examples:

- Predicting the number of views a YouTube video will get next week based on video length, the date it was posted, the previous number of views, etc.
- Predicting which movies a Netflix user will rate highly based on their previous movie ratings, demographic data, etc.

Data

The **Advertising data set** consists of the sales of a particular product in 200 different markets, and advertising budgets for the product in each of those markets for three different media: TV, radio, and newspaper. Everything is given in units of \$1000.

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9

Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) **without** permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani " (permission asked)

Response vs. Predictor Variables

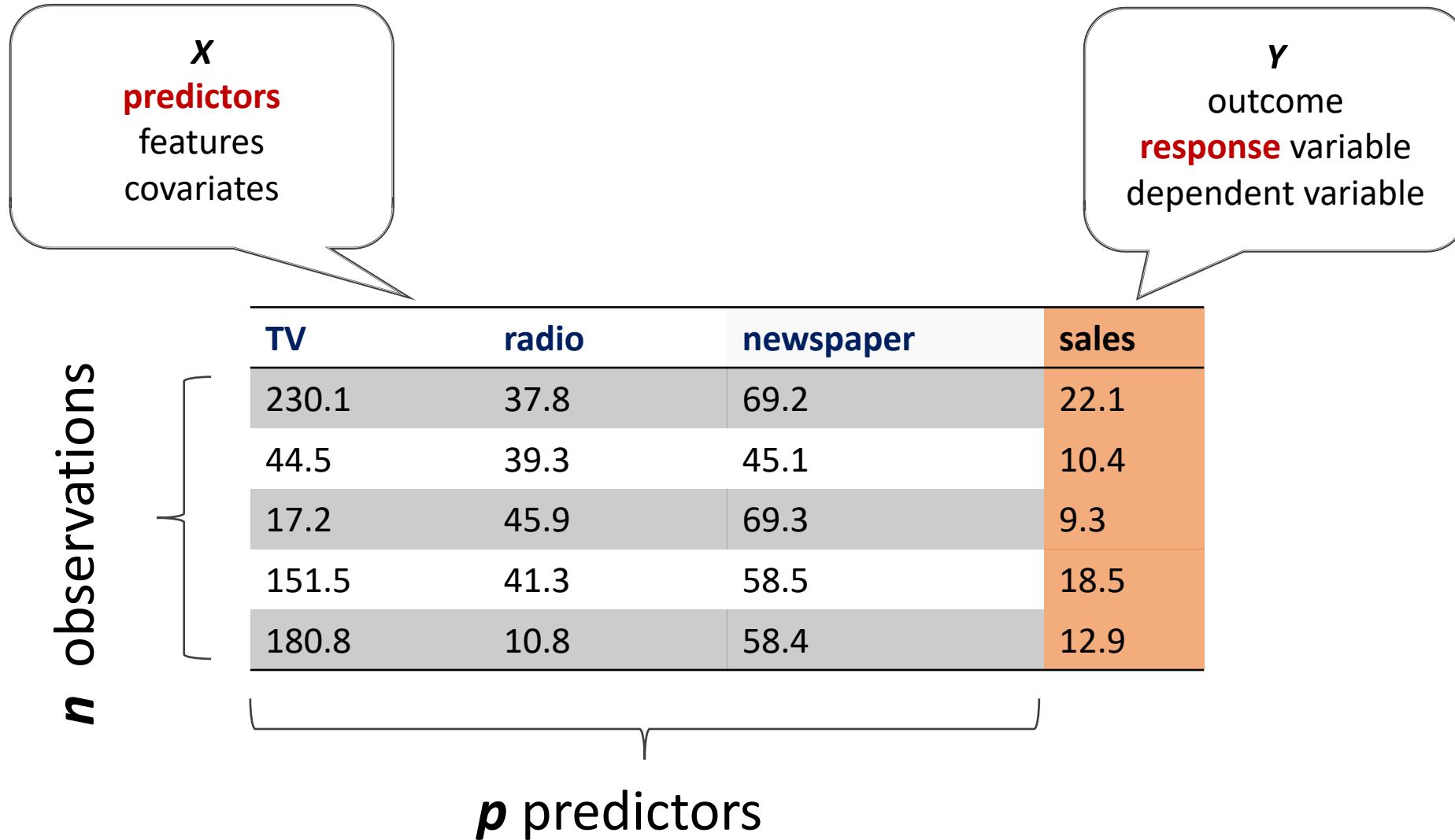
There is an **asymmetry** in many of these problems:

- The variable we would like to predict may be more difficult to measure, is more important than the other(s), or maybe directly or indirectly influenced by the other variable(s).

Thus, we'd like to define two categories of variables:

- variables whose values we want to predict
- variables whose values we use to make our prediction

Response vs. Predictor Variables



X
predictors
features
covariates

Y
outcome
response variable
dependent variable

	TV	radio	newspaper	sales
n observations	230.1	37.8	69.2	22.1
	44.5	39.3	45.1	10.4
	17.2	45.9	69.3	9.3
	151.5	41.3	58.5	18.5
	180.8	10.8	58.4	12.9

p predictors

Response vs. Predictor Variables

$$X = X_1, \dots, X_p$$

$$X_j = x_{1j}, \dots, x_{ij}, \dots, x_{nj}$$

Predictors, features, covariates

$$Y = y_1, \dots, y_n$$

outcome

response variable

dependent variable

n observations

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9

p predictors

Break



WATER



WALK



COFFEE OR TEA



MAKE FRIENDS

Statistical Model

True vs. Statistical Model

We will assume that the response variable, Y , relates to the predictors, X , through some unknown function expressed generally as:

$$Y = f(X) + \varepsilon$$

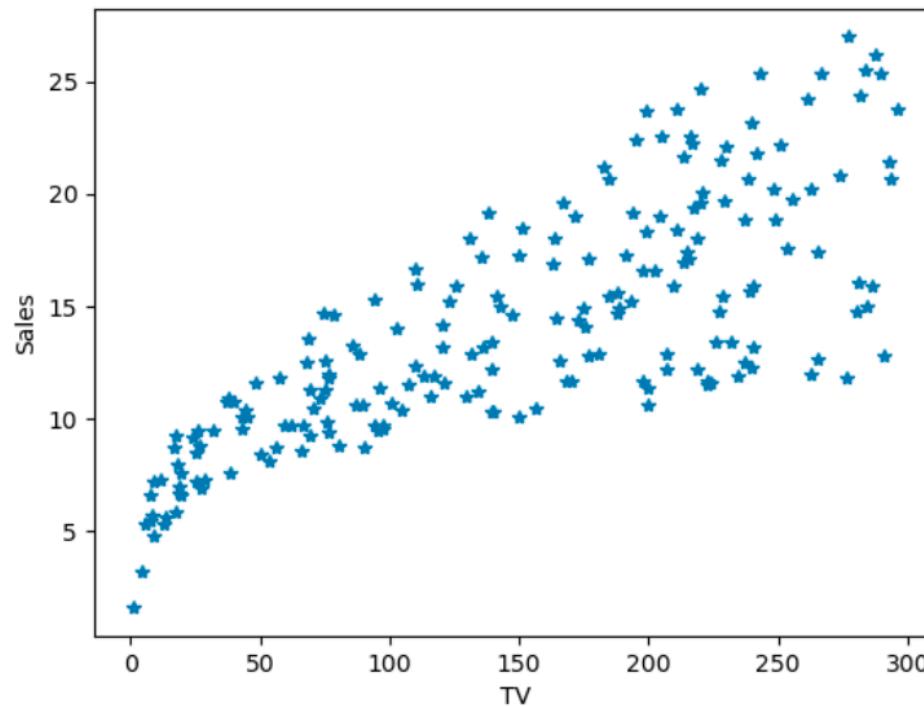
Here, f is the unknown function expressing an underlying rule for relating Y to X , ε is the random amount (unrelated to X) that Y differs from the rule $f(X)$.

A **statistical model** is any algorithm that estimates f . We denote the estimated function as \hat{f} .

Example: predicting sales

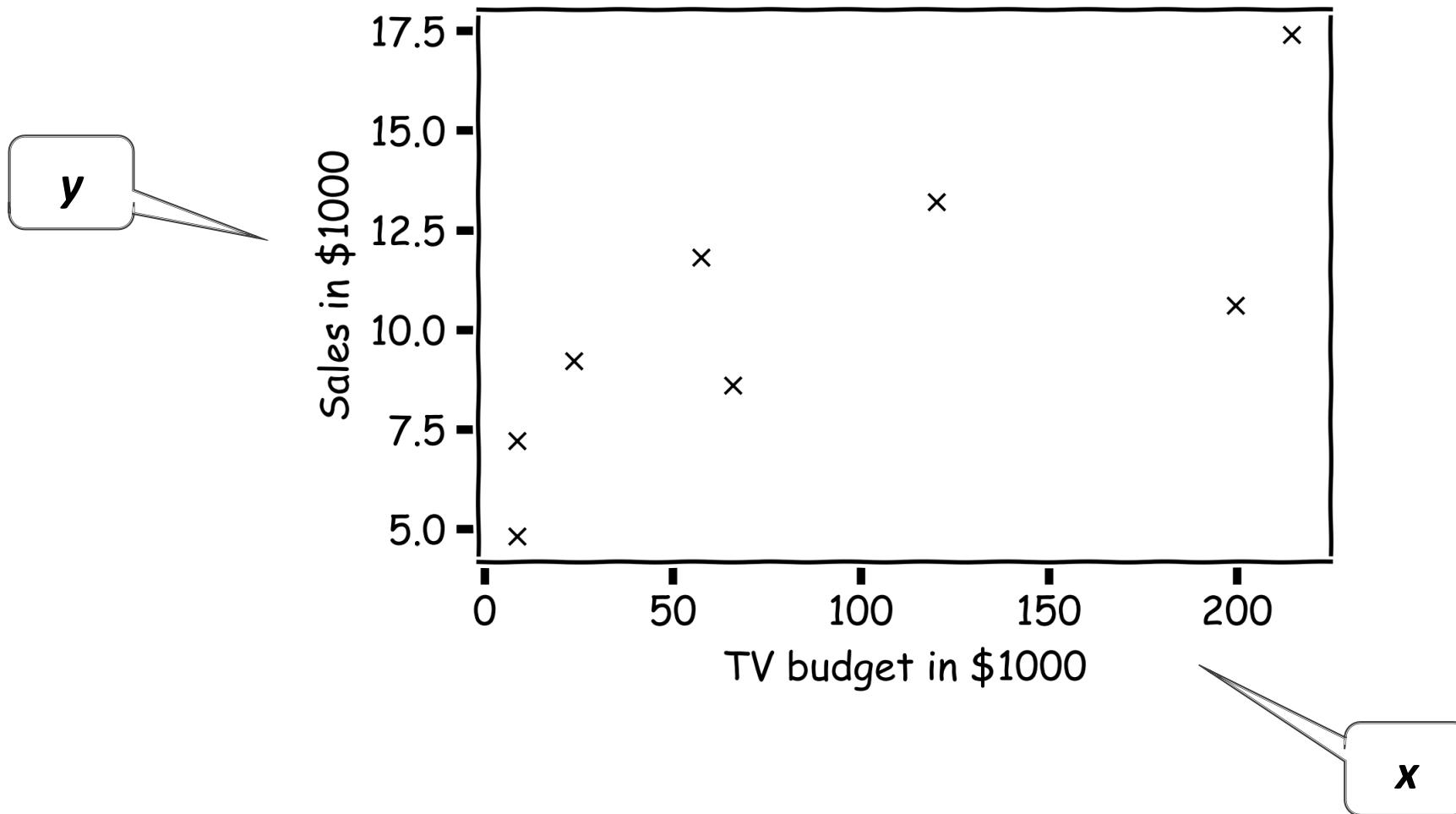
Motivation: Predict Sales

Build a model to **predict** sales based on TV budget



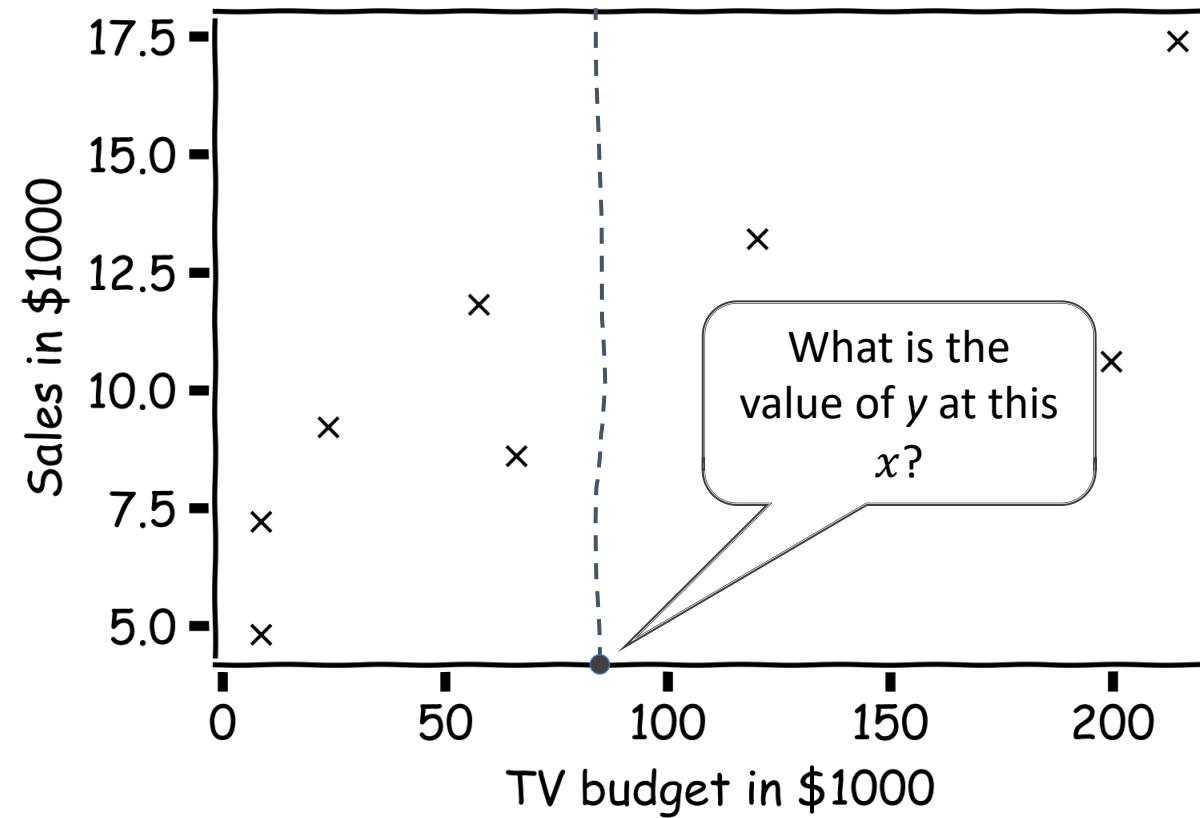
The response, y , is the sales
The predictor, x , is TV budget

Statistical Model



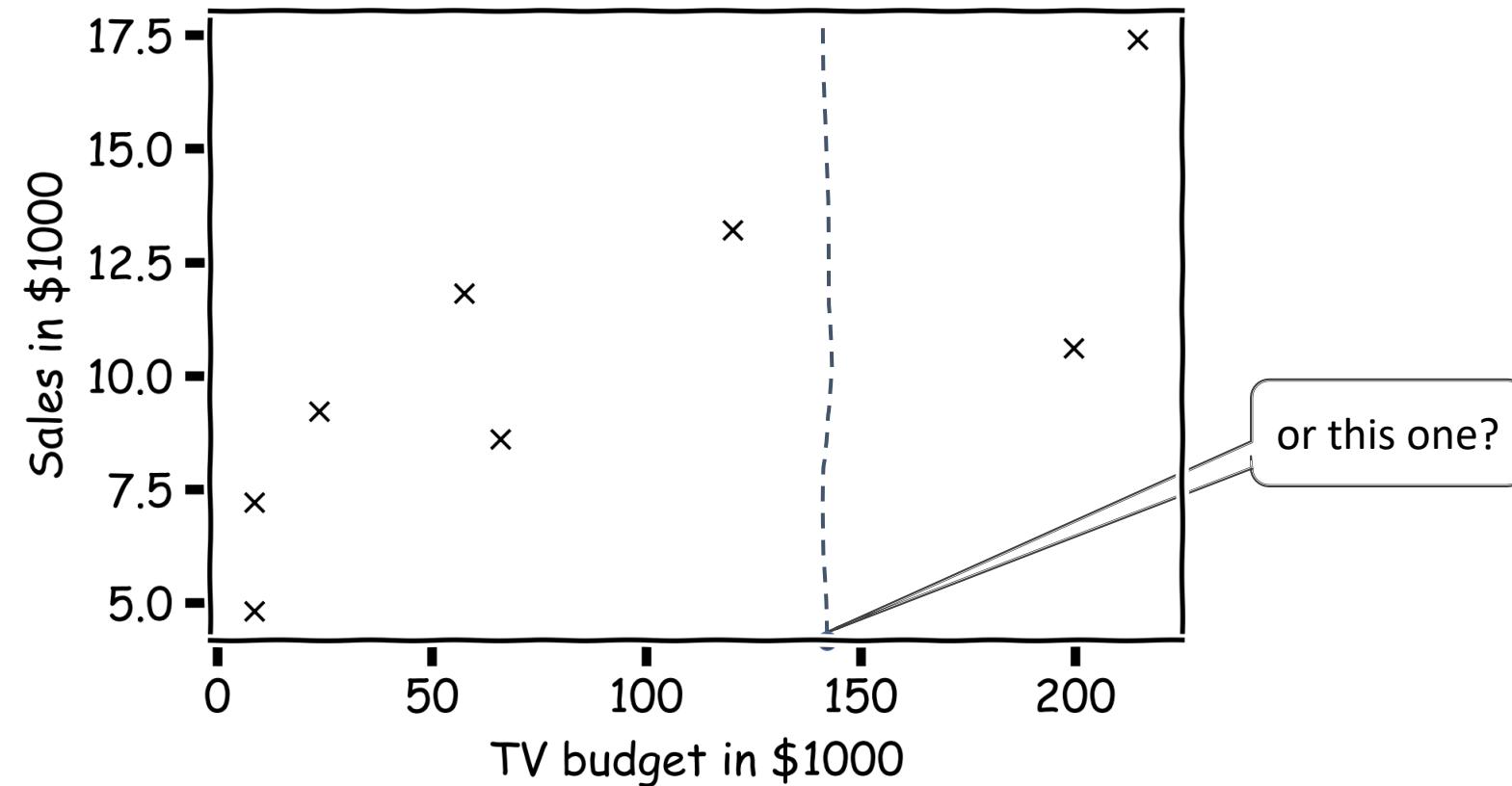
Statistical Model

How do we predict y for some x



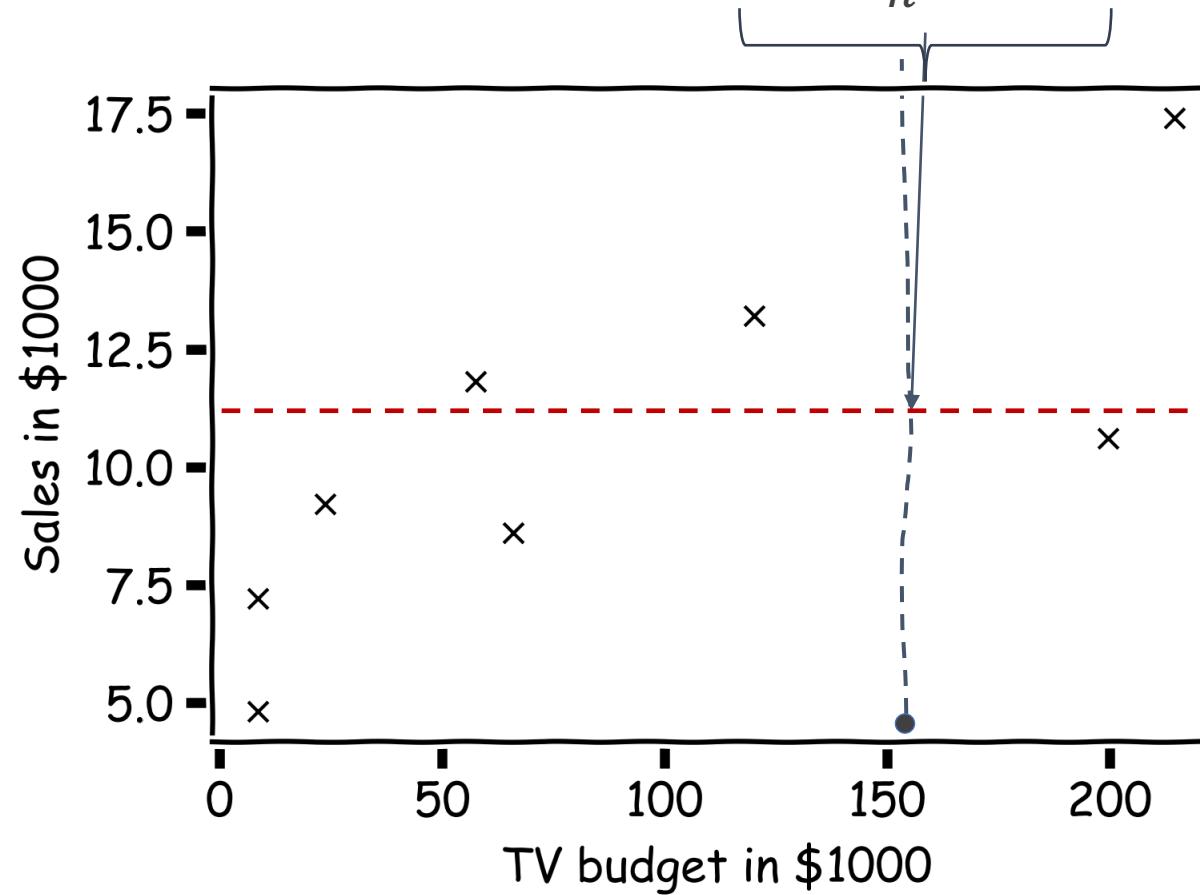
Statistical Model

How do we predict y for some x



Statistical Model

Simple idea is to take the mean of all y 's: $\frac{1}{n} \sum_1^n y_i$



Prediction vs. Estimation

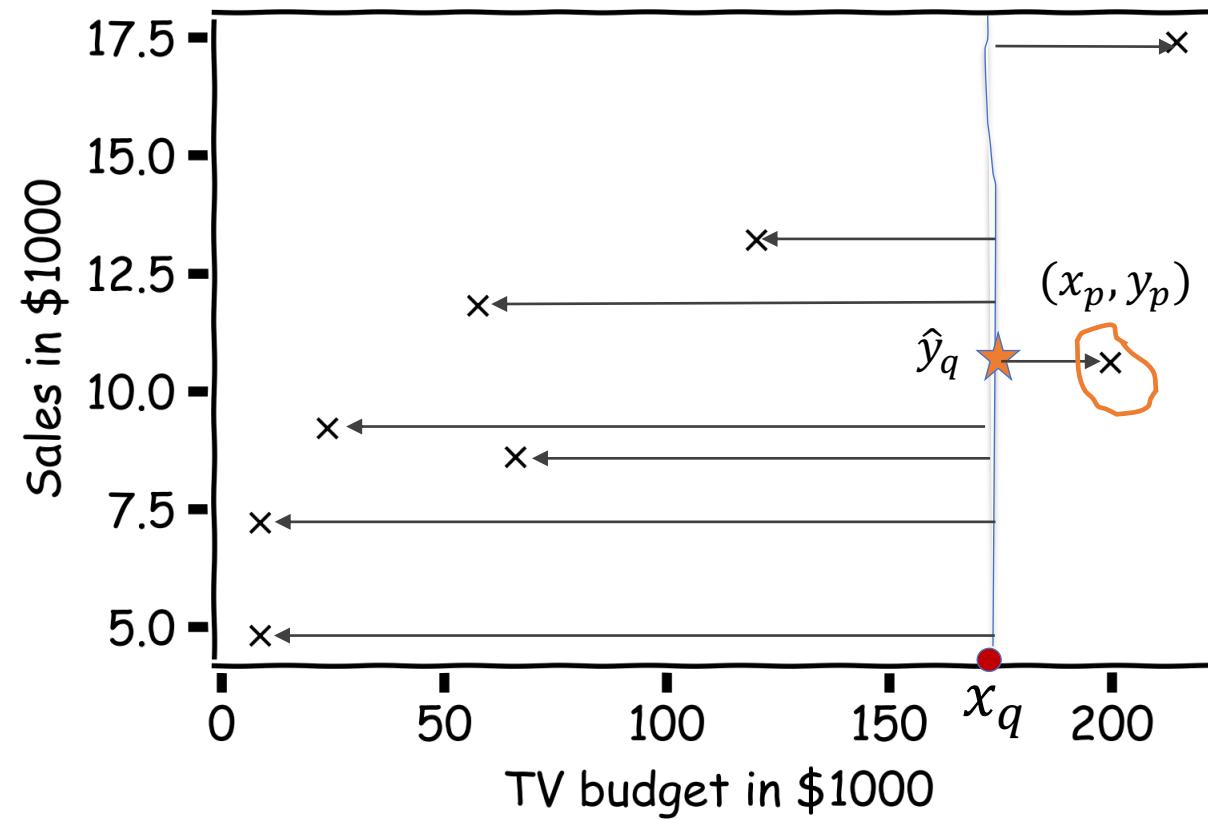
For some problems, what's important is obtaining \hat{f} , our estimate of f . These are called ***inference*** problems.

When we use a set of measurements, $(x_{i,1}, \dots, x_{i,p})$ to predict a value for the response variable, we denote the ***predicted*** value by:

$$\hat{y}_i = \hat{f}(x_{i,1}, \dots, x_{i,p}).$$

For some problems, we don't care about the specific form of \hat{f} , we just want to make our predictions \hat{y} 's as close to the observed values y 's as possible. These are called ***prediction problems***.

Simple Prediction Model



What is \hat{y}_q at some x_q ?

Find distances to all other points

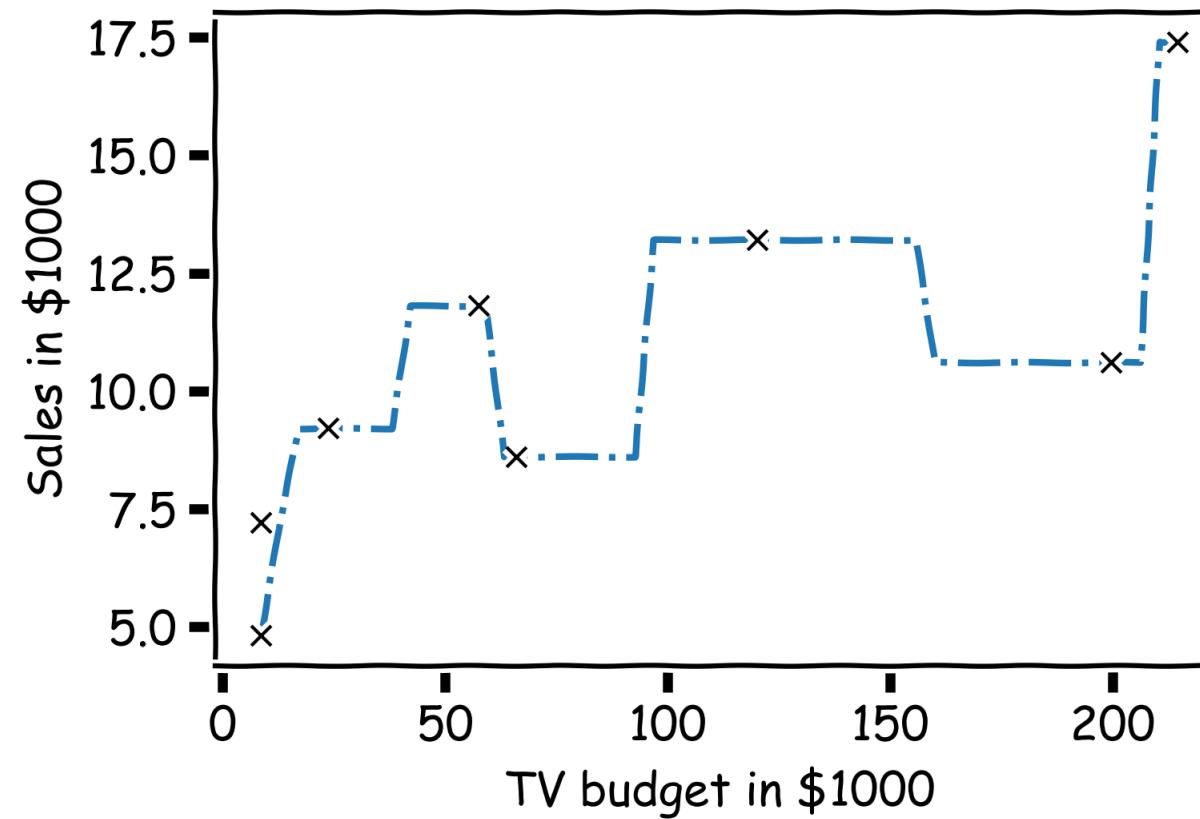
$$D(x_q, x_i)$$

Find the nearest neighbor, (x_p, y_p)

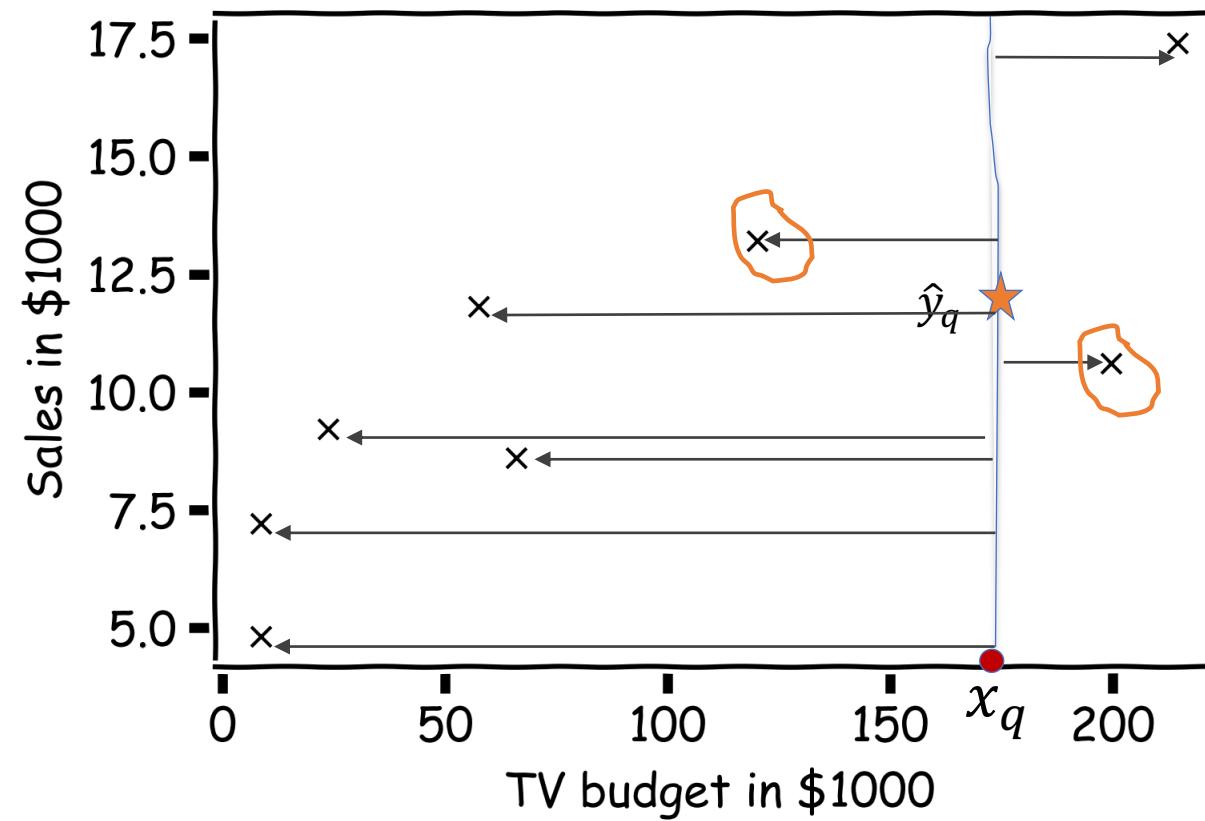
Predict $\hat{y}_q = y_p$

Simple Prediction Model

Do the same for “all” x ’s



Extend the Prediction Model



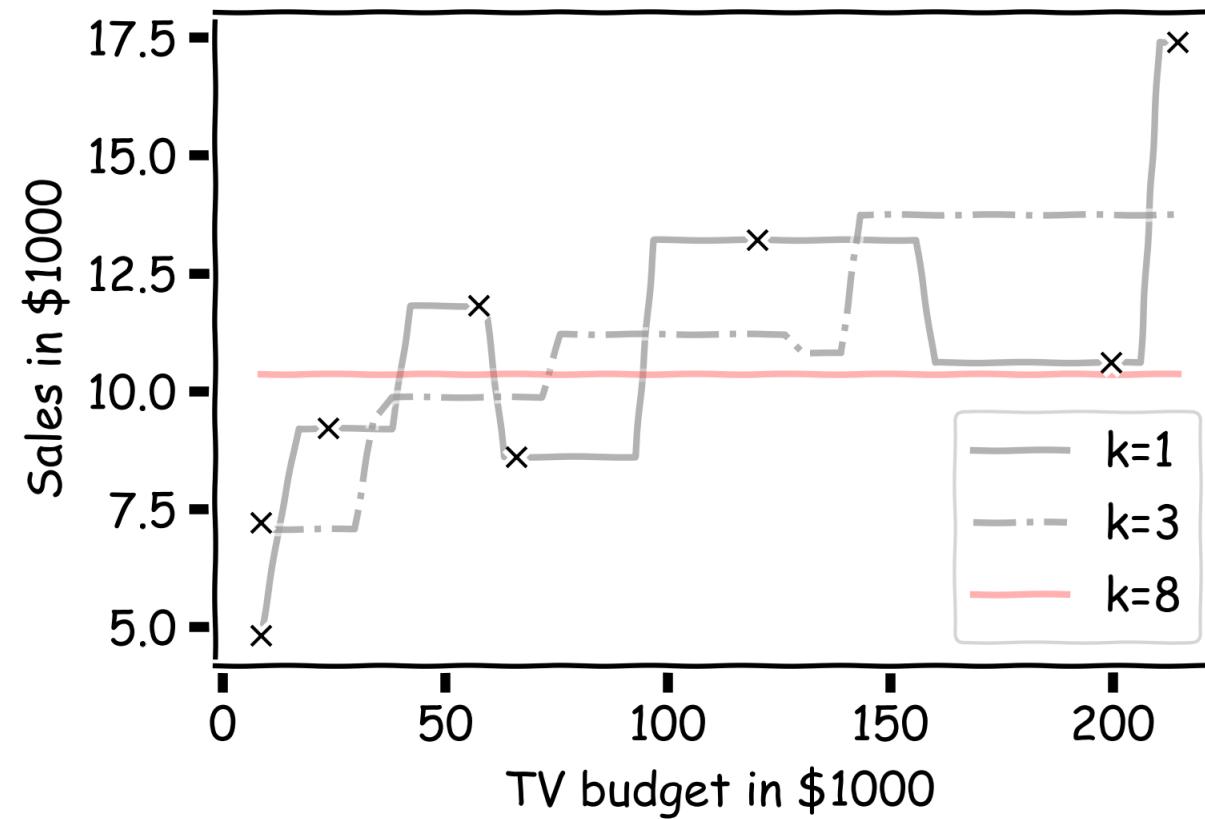
What is \hat{y}_q at some x_q ?

Find distances to all other points
 $D(x_q, x_i)$

Find the k-nearest neighbors, x_{q_1}, \dots, x_{q_k}

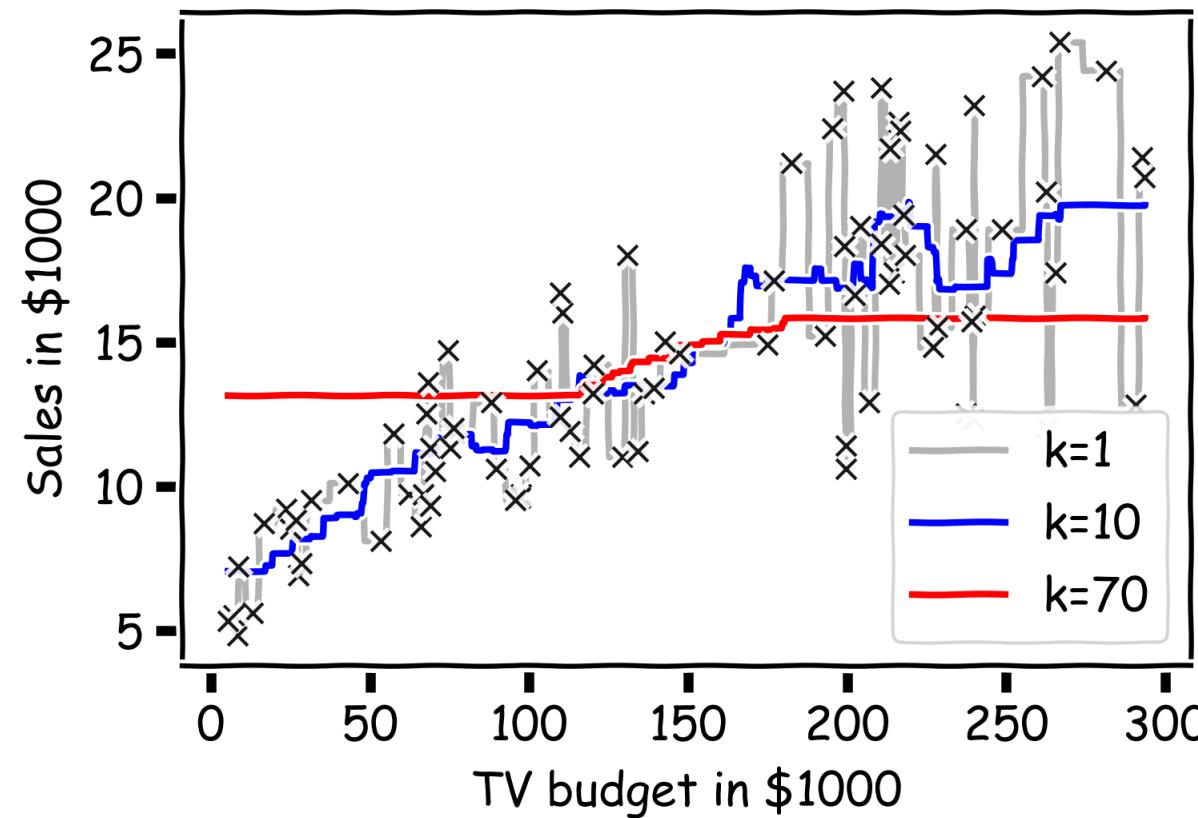
Predict $\hat{y}_q = \frac{1}{k} \sum_i^k y_{q_i}$

Simple Prediction Models



Simple Prediction Models

We can try different k-models on more data



k-Nearest Neighbors

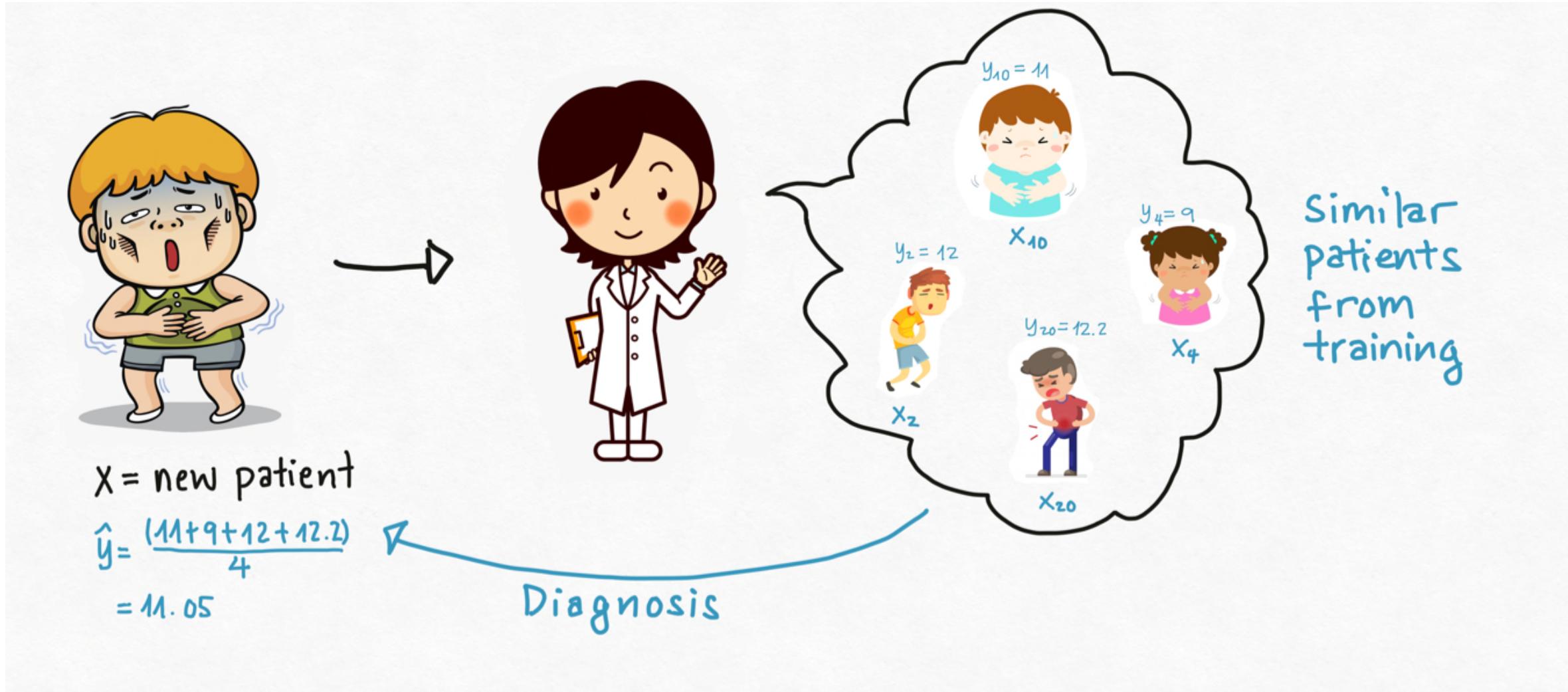
The ***k-Nearest Neighbor (kNN) model*** is an intuitive way to predict a quantitative response variable:

to predict a response for a set of observed predictor values, we use the responses of other observations most like it

kNN is a **nonparametric** learning algorithm. When we say a technique is nonparametric, it means that it does not make any assumptions on the underlying data distribution.

Note: this strategy can also be applied in classification to predict a categorical variable. But classification is not in the scope of this course.

k-Nearest Neighbors



k-Nearest Neighbors

The **very human way** of decision making by similar examples. kNN is a **nonparametric** learning algorithm.

The k-Nearest Neighbor Algorithm:

Given a dataset $D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})\}$. For every new X :

1. Find the k-number of observations in D most like X :

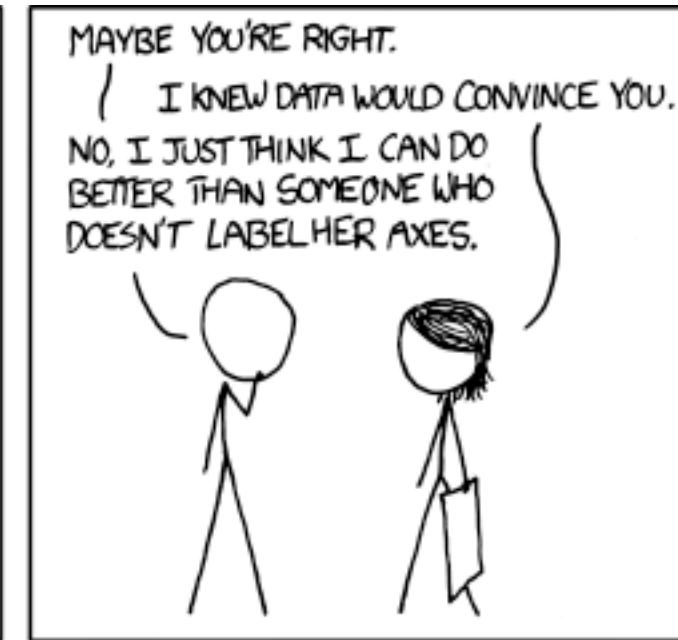
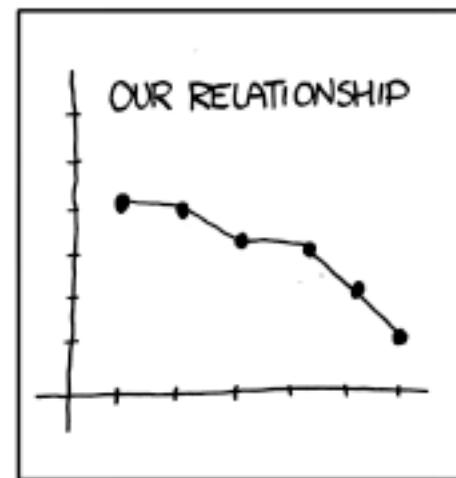
$$\{(x^{(n_1)}, y^{(n_1)}), \dots, (x^{(n_k)}, y^{(n_k)})\}$$

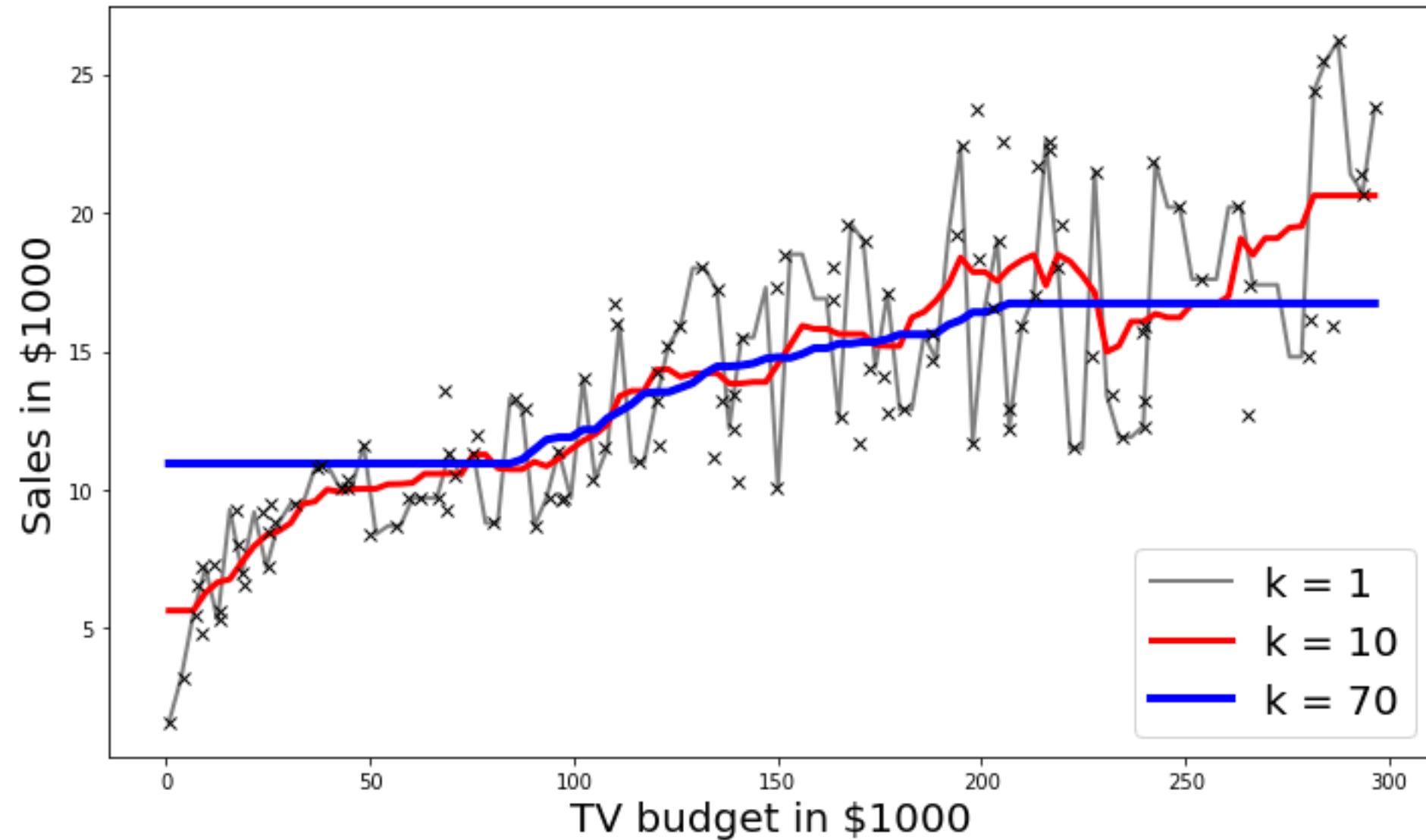
These are called the **k-nearest neighbors** of x

2. Average the output of the k-nearest neighbors of x

$$\hat{y} = \frac{1}{K} \sum_{k=1}^K y^{(n_k)}$$

Error Evaluation and Model Comparison

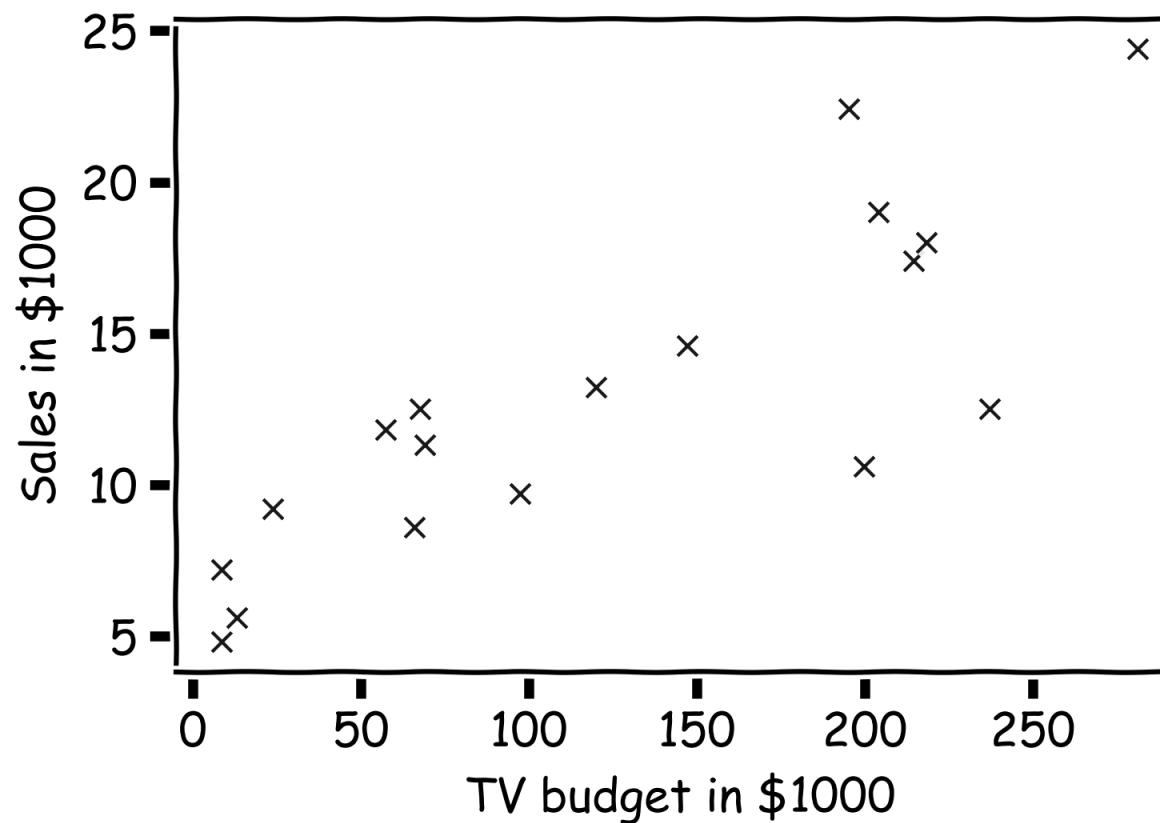




Error Evaluation

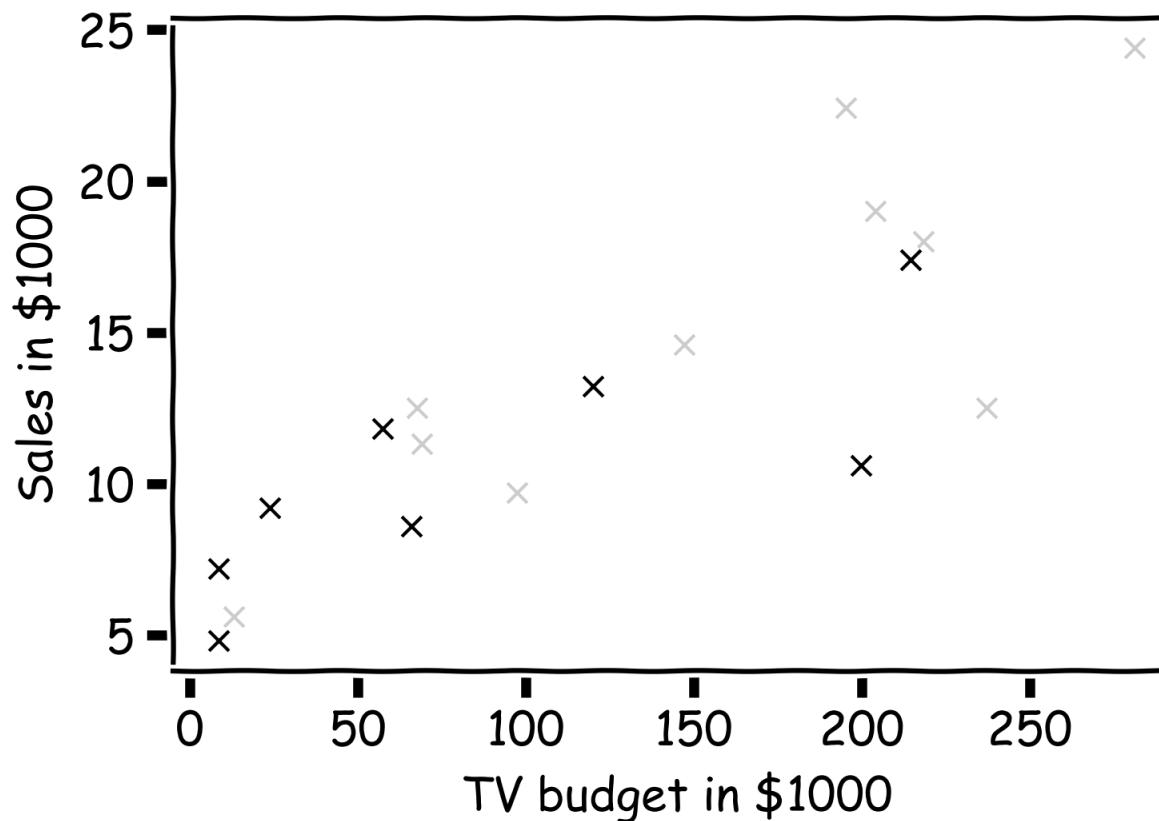
Error Evaluation

Start with some data.



Error Evaluation

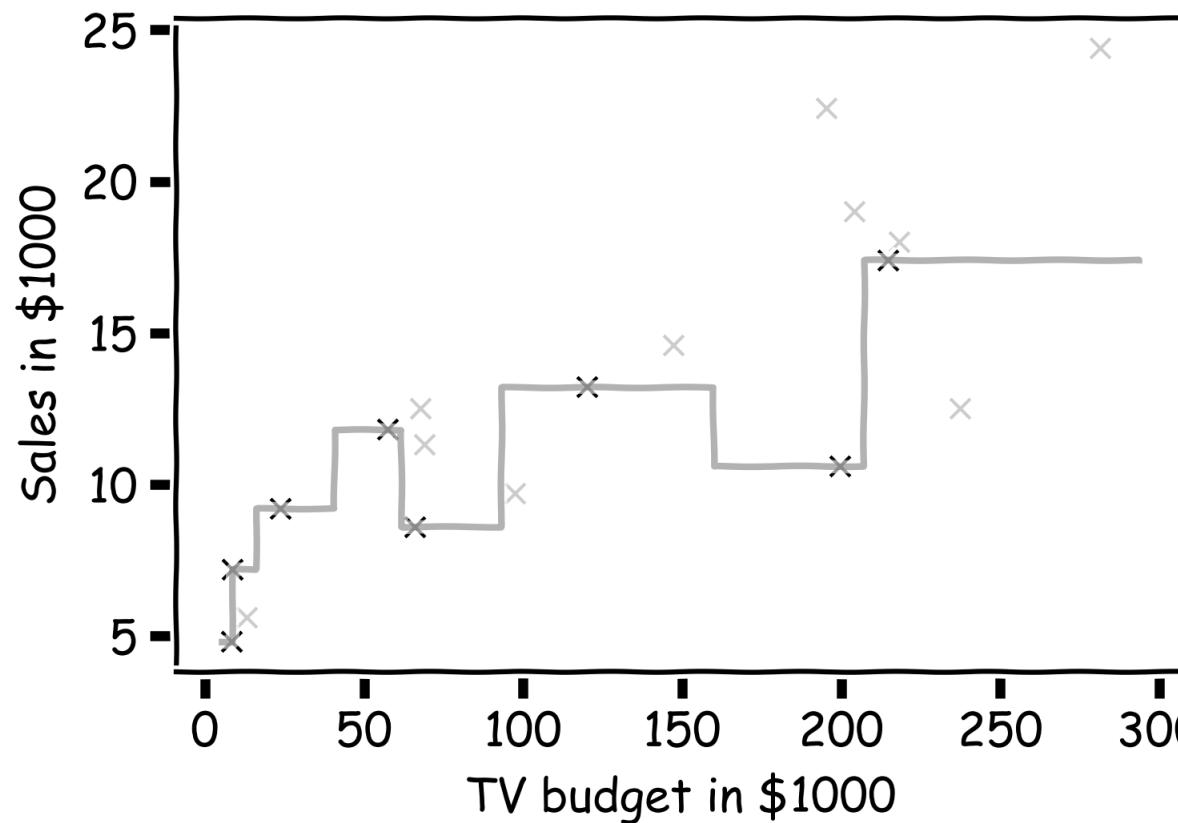
Hide some of the data from the model. This is called **train-test** split.



We use the **train** set to estimate \hat{y} , and the **test** set to **evaluate** the model.

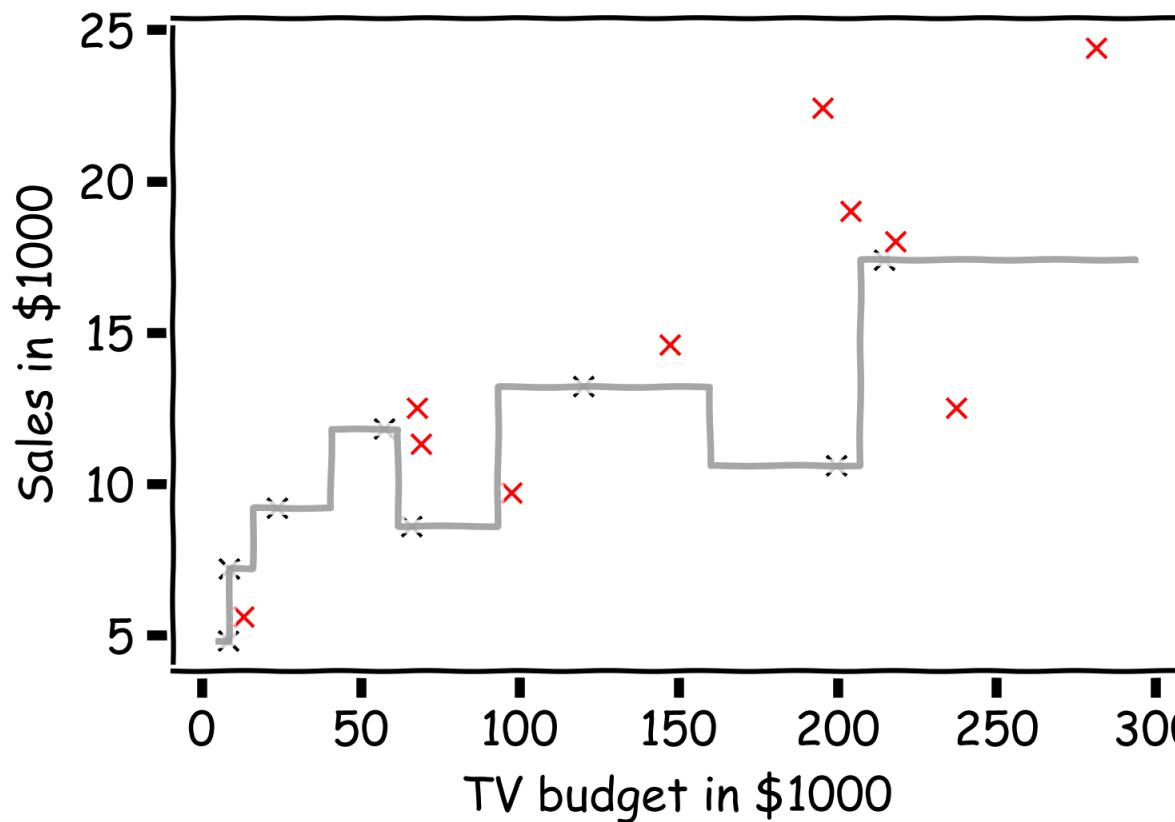
Error Evaluation

Estimate \hat{y} for $k=1$.



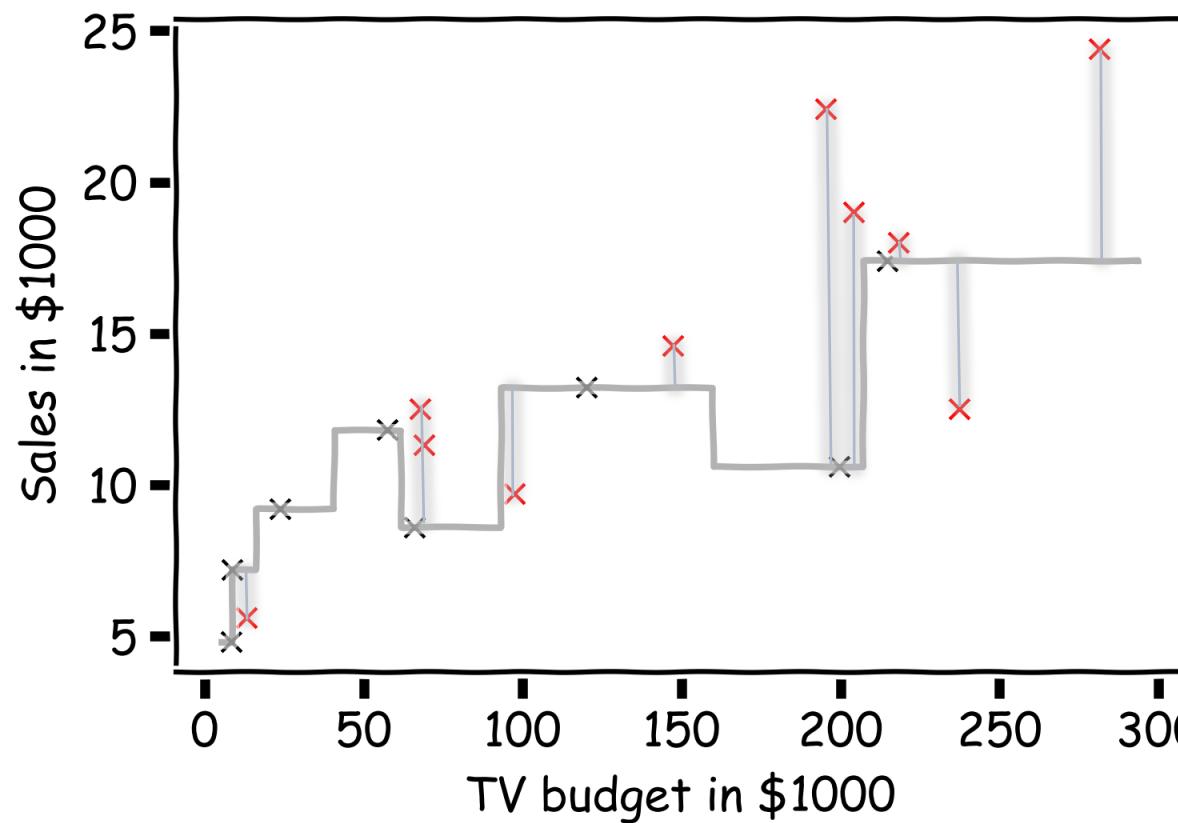
Error Evaluation

Now, we look at the data we have not used, the **test data** (red crosses).



Error Evaluation

Calculate the **residuals** ($y_i - \hat{y}_i$).



Error Evaluation

In order to quantify how well a model performs, we **aggregate** the errors, and we call that the ***loss* or *error* or *cost function***.

A common **loss function** for quantitative outcomes is the **Mean Squared Error (MSE)**:

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

Note: Loss and cost function refer to the same thing. Cost usually refers to the total loss where loss refers to a single training point.

Error Evaluation

Caution: The MSE is by no means the only valid (or the best) loss function!

1. Max Absolute Error
2. Mean Absolute Error
3. Mean Squared Error

We can choose other functions here (but out of scope of this course).

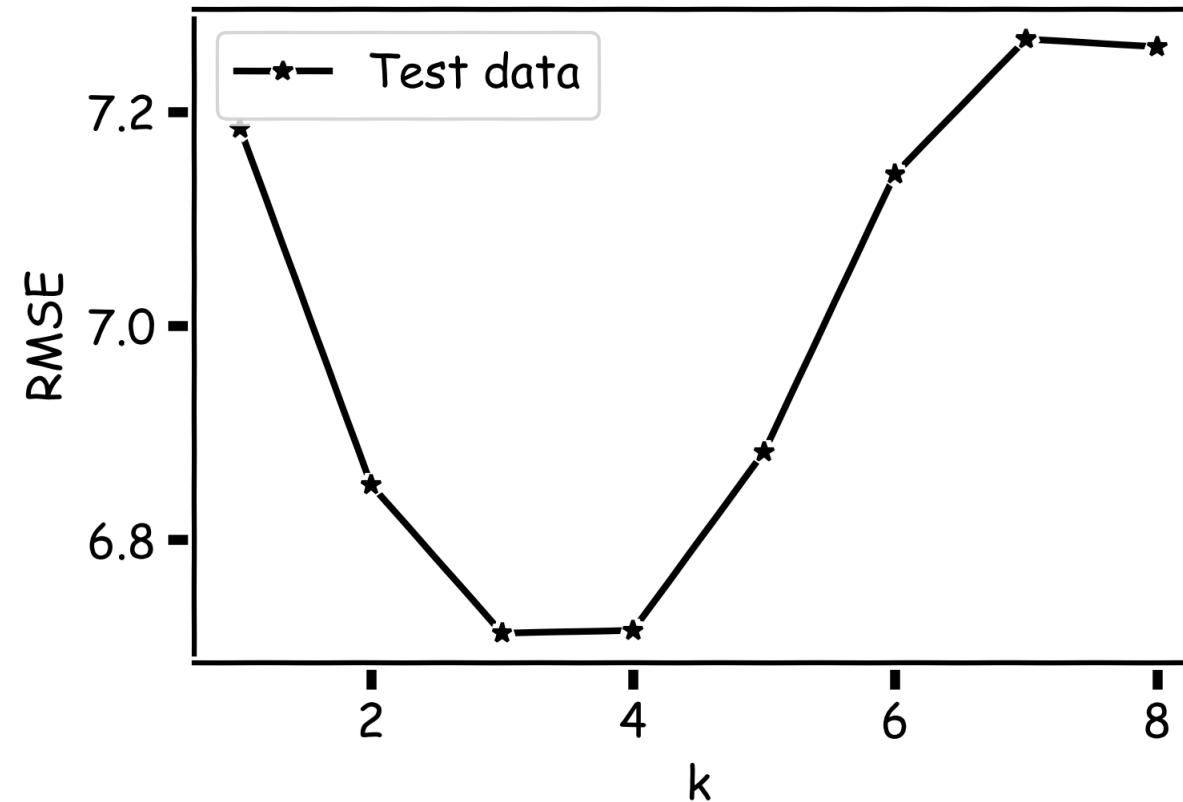
Note: The square Root of the Mean of the Squared Errors (RMSE) is also commonly used.

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

Model Comparison

Model Comparison

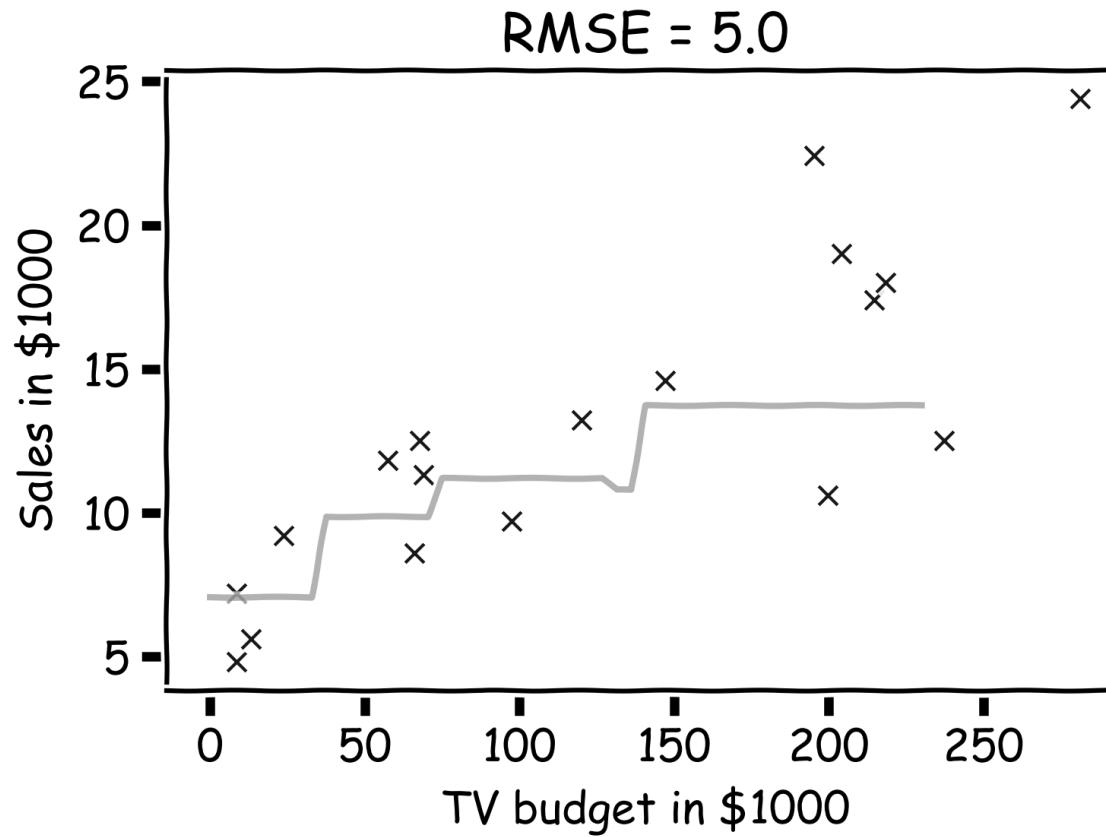
Do the same for all k 's and compare the RMSEs. $k=3$ seems to be the **best model**.



Model Fitness

Model fitness

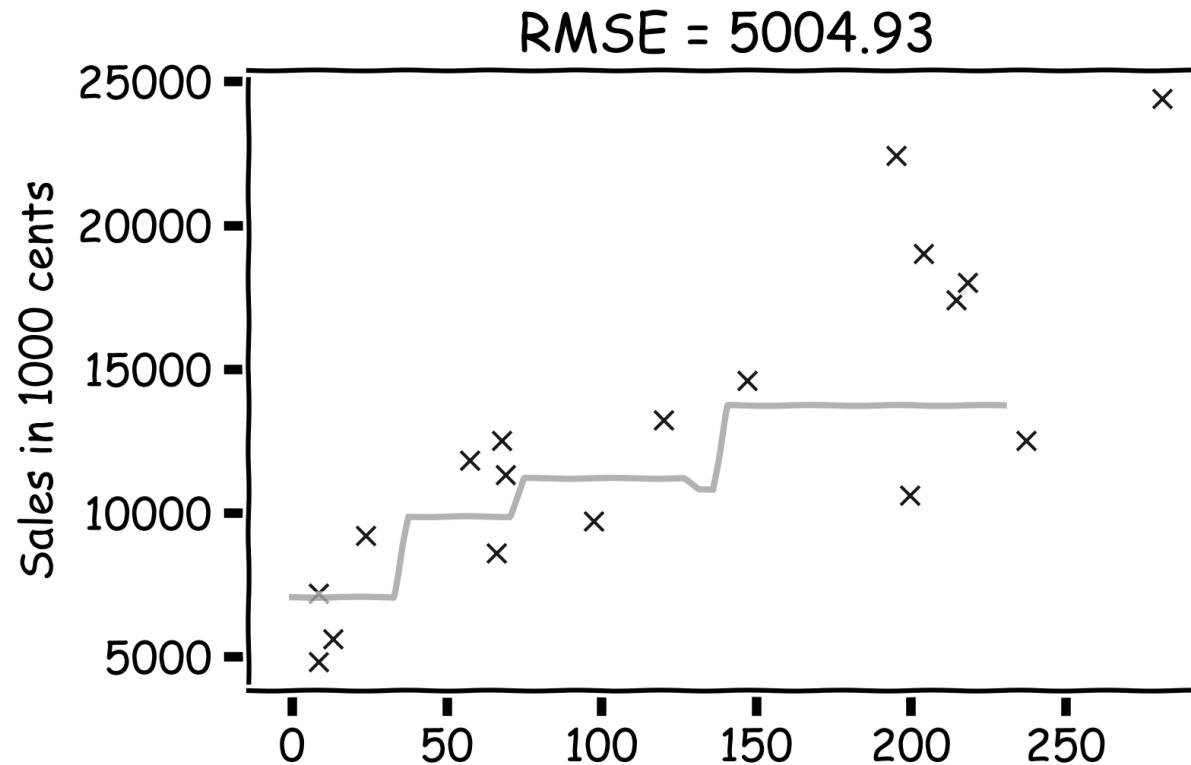
For a subset of the data, calculate the RMSE for $k=3$.



Is RMSE = 5.0 good enough?

Model fitness

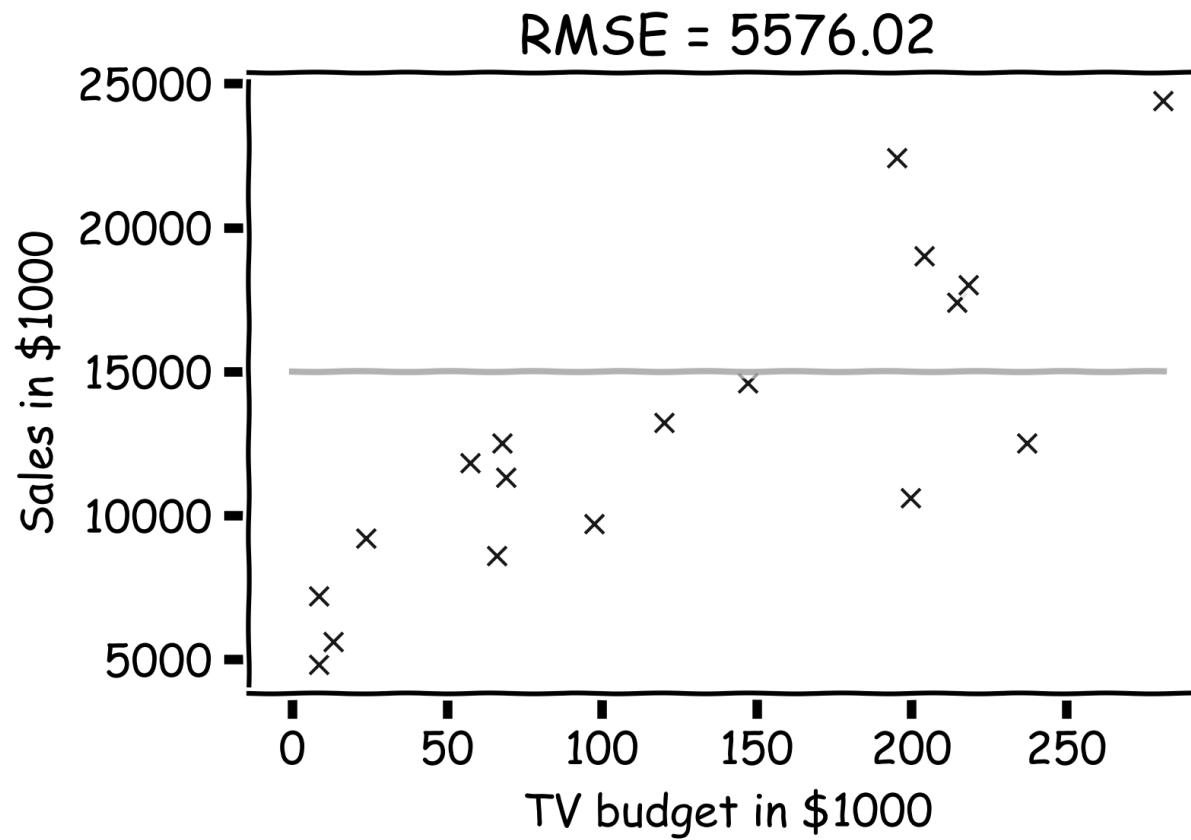
What if we measure the Sales in cents instead of dollars?



RMSE is now 5004.93.
Is that good?

Model fitness

It is better if we compare it to something.



We will use the simplest model:

$$\hat{y} = \bar{y} = \frac{1}{n} \sum_i y_i$$

as the **worst** possible model and

$$\hat{y}_i = y_i$$

as the **best** possible model.

R-squared

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{y} - y_i)^2}$$

- If our model is as good as the mean value, \bar{y} , then $R^2 = 0$
- If our model is perfect, then $R^2 = 1$
- R^2 can be negative if the model is worst than the average. This can happen when we evaluate the model in the test set.

Recapitulation

Machine learning is for everyone and it only predicts results
based on incoming data!

For next class..



Finish Lab 06 to practice programming



Submit Homework 06 for peer review on Peer



Submit Assignment 3 – due in **Week 7** on Sunday at **2330**



See “To do before class” for every lecture (~ 1 hour of self study)



Read paper for **Discussion** session before every Friday



Post questions on the **Discussion** forum on Brightspace (especially on **ML** for this week)