# Machine Learning

## Regression framework

Sebastian Engelke

Master in Business Analytics

UNIVERSITÉ
DE GENÈVE

# Regression: the stochastic framework

The stochastic/population model

- We have a quantitative response $Y$ and $p$ predictors $X_1, \ldots, X_p$.
- We assume there is a relationship between $Y$ and $X = (X_1, \ldots, X_p)$

$$Y = f(X) + \epsilon,$$

where $\epsilon$ is a random error term with $\mathrm{E}(\epsilon) = 0$ and $\mathrm{Var}(\epsilon) = \sigma^2$.

- The function $f$ is fixed but unknown and represents the systematic information that $X$ provides about $Y$.

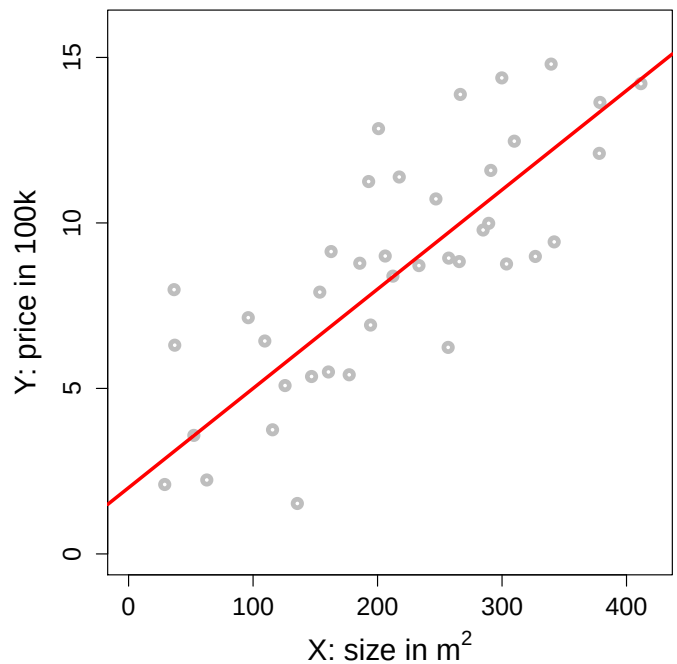
Example: [$Y =$ house price $(100k)$; $X =$ size $(m^2)$]

$$Y = 2 + 0.03X + \epsilon$$

$$\text{house price} = 2 + 0.03 \times \text{size} + \text{error}$$

- This true relationship $f(X) = 2 + 0.03X$ is called the population regression line.
- Note: $(X, Y)$ is a $(p+1)$-dimensional random vector. In this simulated example:

$$X = \mathcal{N}(0, \sigma_X^2), \text{ and } Y = 2 + 0.03X + \mathcal{N}(0, \sigma_\epsilon^2).$$



# Regression: the statistical framework

The `training data`

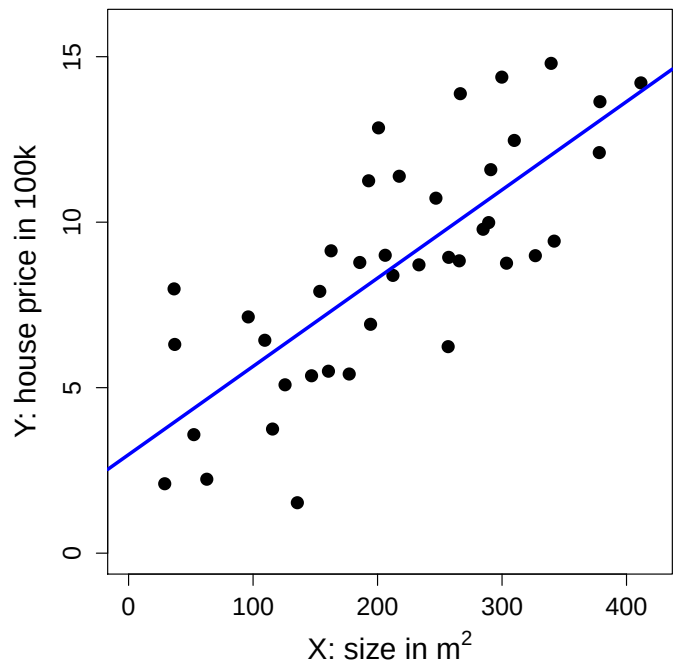
- Assume we are given $n$ measurements $(x_1, y_1), \ldots, (x_n, y_n)$ of $(X, Y)$, where $x_i = (x_{i1}, \ldots, x_{ip})^\top$, and
  - the inputs $x_i \in \mathbb{R}^p$ are called `predictors`;
  - the outputs $y_i$ are called `responses`.
- We will use this data to `learn` the unknown relationship $f$ between $Y$ and $X$, i.e., to `train` (or `estimate`) a predictive model $\widehat{f}$.

Example: [$Y =$ `house price` $(100k)$; $X =$ `size` $(m^2)$]

$$\texttt{house price} \approx \widehat{f}(\texttt{size}) = 2.977 + 0.027 \times \texttt{size}$$

- The training data are realizations of the random vector $(X, Y)$, e.g.,

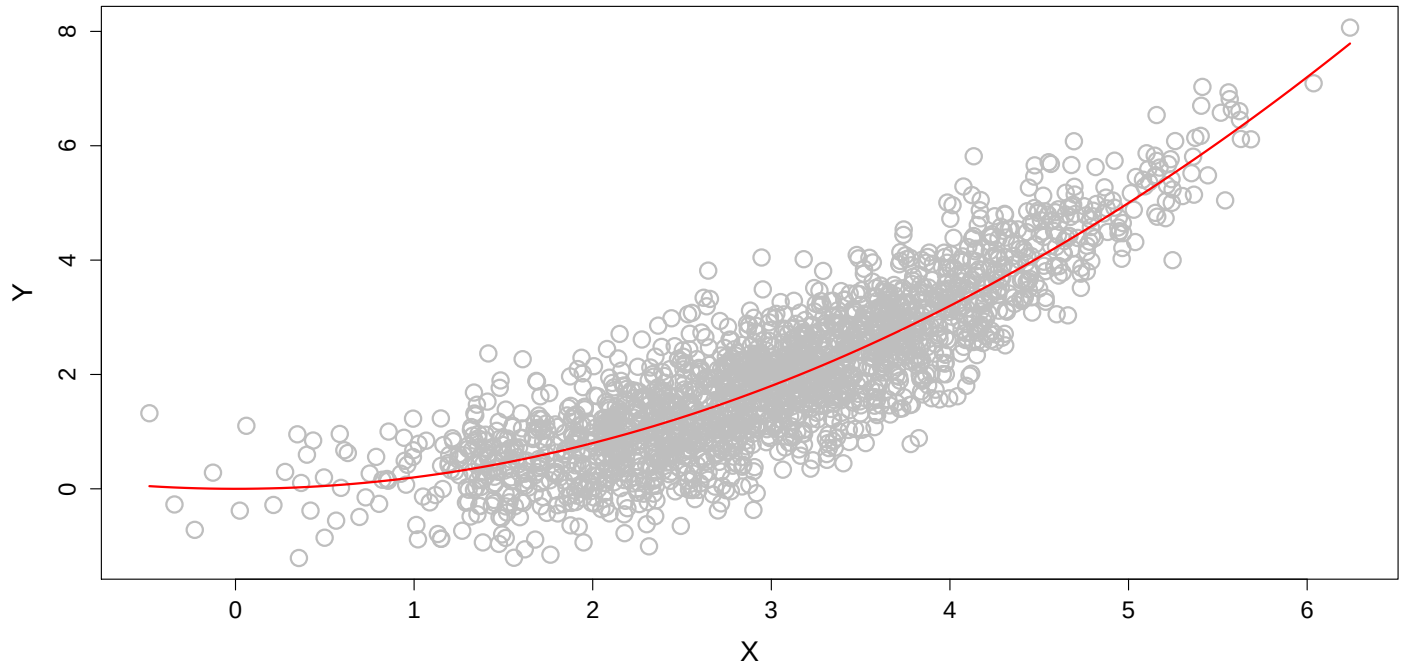
| | $X$ | $Y$ |
|---|---|---|
| $(x_1, y_1)$ | 327 | 9.0 |
| $(x_2, y_2)$ | 154 | 7.9 |
| $(x_3, y_3)$ | 233 | 8.7 |
| $(x_4, y_4)$ | 206 | 9.0 |
| $(x_5, y_5)$ | 96 | 7.1 |
| $(x_6, y_6)$ | 257 | 6.2 |
| ... | ... | ... |



# Why do we estimate $f$?

1. `Prediction` (Machine learning)
   - A set of inputs $X$ are readily available, but $Y$ cannot be easily obtained.
   - We can predict $Y$ by
   $$\widehat{Y} = \widehat{f}(X),$$
   where $\widehat{f}$ is our `predictive model` for $f$.
   - The function $\widehat{f}$ is often treated as `black box`: we are not concerned with the exact form of $\widehat{f}$ provided it yields good prediction for $Y$.
2. `Inference` (Classical multivariate statistics)
   - We are often interested in understanding `how` $Y$ is affected as $X_1, \ldots, X_p$ change.
   - Cannot treat $\widehat{f}$ as black box, we need to know the exact form.
   - We want to answer questions such as:
     - Which predictor is associated with the response?
     - What is the relationship of the response and each predictor (positive, negative,...)?
     - Is the relationship linear or more complicated?
     - etc.

# What is a good $\widehat{f}$?




- For a fixed value of $X$, say $X = 3$ is there an `optimal prediction` $\widehat{f}(X)$?
- There might be many values at $X = 3$, and a good choice, minimizing squared error, is

$$\widehat{f}(3) = \mathrm{E}(Y \mid X = 3),$$

where $\mathrm{E}(Y \mid X = 3)$ is the `expected value` (average) of $Y$ given $X = 3$.

# The loss function

- We have to specify what we mean with a `good prediction`. This will depend on the application.
- Introduce a `loss function` $L$ that measures the discrepancy between a response $y$ and its prediction $\widehat{y}$.
- In the regression setup, we typically use `squared error loss`, i.e.,

$$L(y, \widehat{y}) = (y - \widehat{y})^2.$$

- For our predictive model $\widehat{f}$, the `expected squared prediction error` is

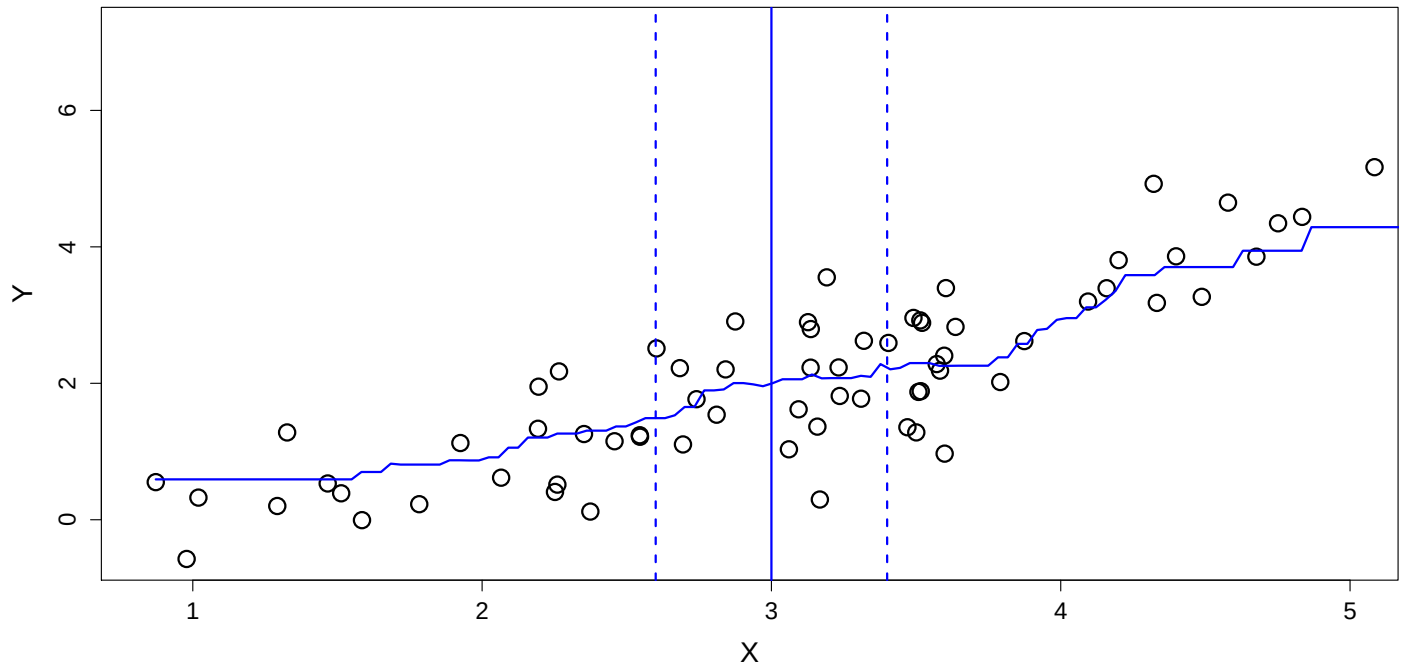
$$\mathrm{Err}_{\widehat{f}} = \mathrm{E}[\{Y - \widehat{f}(X)\}^2],$$

  where the expectation is taken over $(X, Y)$.
- Ideally, we would like to find the optimal $f^*$ that `minimizes` $\mathrm{Err}_{f^*}$.
- In fact, $f^*$ is the so-called `regression function`

$$f^*(x) = \mathrm{E}(Y|X = x),$$

  which satisfies $f^* = f$ if $Y = f(X) + \epsilon$.

# How do we estimate $\widehat{f}$?




- Typically we have `few or no` data points with $X = 3$.
- So we cannot compute $\mathrm{E}(Y \mid X = 3)$!
- The `k-Nearest-Neighbors` method aims at approximating this expectation by the average

$$\widehat{f}(x_0) = \mathrm{Ave}(Y \mid X \in N_k(x_0)),$$

  where $N_k(x_0)$ are the $k$ closest points $x_i$ to $x_0 = 3$.
- The number $k$ is a `tuning parameter`.

# How do we estimate $\widehat{f}$?

- Nearest neighbor averaging works good for small dimension $p$ and a large training data size $n$.
- When the number of predictors $X_1, \ldots, X_p$ is larger, then kNN does not work well! The reason is the `curse of dimensionality`.
- We will study `structured methods` that
  - need `less data` to fit;
  - may be nicely `interpretable`;
  - but at the price of `lower flexibility`.
- The difficulty is to find a good `compromise` between flexibility and parsimony.

# The empirical loss: model evaluation

How can we estimate the $\mathrm{Err}_{\widehat{f}} = \mathrm{E}[\{Y - \widehat{f}(X)\}^2]$?

The test data

- For model evaluation we require a new, independent test data set $\mathcal{T}_m = \{(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_m, \tilde{y}_m)\}$.
- For each input $\tilde{x}_i$ we predict the response with our model: $\widehat{\tilde{y}}_i = \widehat{f}(\tilde{x}_i)$.
- We compare this prediction with the observed output $\tilde{y}_i$.
- The empirical prediction error of our model $\widehat{f}$ is the empirical version of $\mathrm{Err}_{\widehat{f}}$, namely

$$\mathrm{err}_{\widehat{f}} = \frac{1}{m} \sum_{i=1}^{m} \{\tilde{y}_i - \widehat{f}(\tilde{x}_i)\}^2,$$

  also called mean squared error.
- In practice: separate all data into training and test and do cross-validation (later).