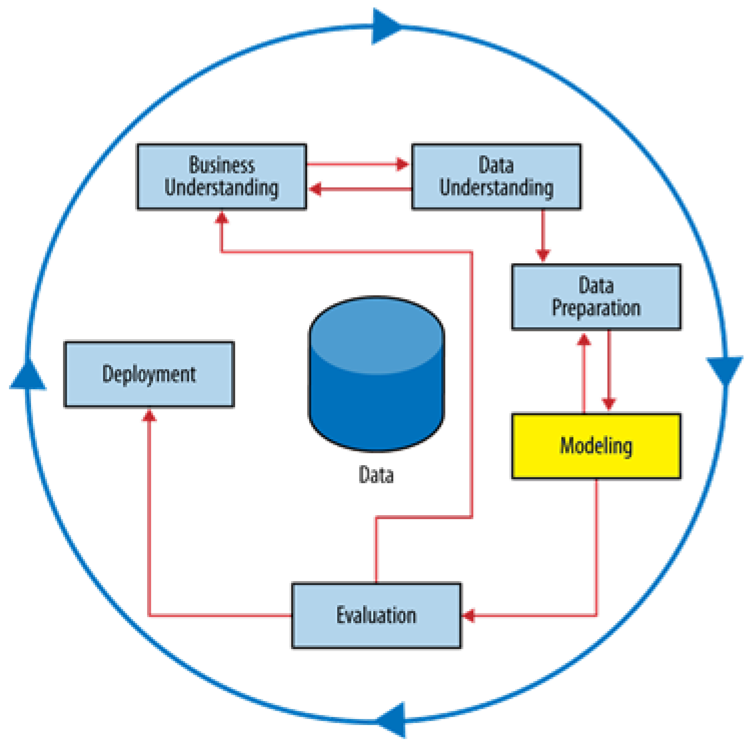
Linear Regression

MG2 & WM12

Learning Goals

- Predictive Modeling: a Regression Example
  - Machine learning with a mathematical formula
    - Supervised learning
  - Predict with a regression model
    - Predict a numeric target
  - Evaluation metrics
- Linear Models
  - Parameter learning and interpretation
  - Accommodate non-linear relationship
    - Categorical features
    - Polynomial features



Machine Learning

- **Machine Learning** refers to the process which computers learns patterns, trends, relationship (i.e., model) from data without being programmed explicitly.
  - The model could be a logical statement such as a rule (e.g., trees) or a mathematical statement (e.g., linear models).
  - Supervised learning is the most prevalent approach to train a predictive model: classification vs. regression.

- Linear Regression, Logistic Regression, Support Vector Machines are **mathematical functions** with a set of numeric parameters (i.e., coefficients).
  - The process of model training is to learn the best value for the parameters (i.e., parameter learning) from the data .

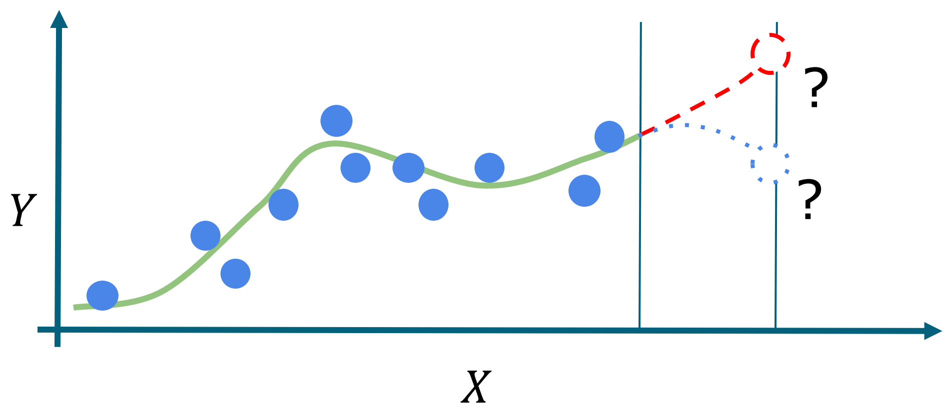
Statistical Models in Machine Learning

- Two purposes of modeling:
  - **Inference**: estimate the relationship between X & Y in the population, based on their observed relationship in a sample.
    - Regression coefficients (and p value) shows the relationship.
    - Model quality is measured by R^2, F statistic, p value.
  - **Prediction**: predict unknown Y given the known X values, with the model trained on a historical sample.
    - Regression coefficients indicates the importance of X in predicting Y.
    - Model quality is measured by R^2, Mean Squared Error (MSE), etc.

- Traditional statistics are mostly inferential, while machine learning focuses more on prediction.
  - Lots of concepts in traditional statistics are applicable to machine learning: e.g., model fitting, parameter learning.

Common Task: Regression

- **Regression**: features ($\boldsymbol{X}$) $\rightarrow$ **numerical target** ($\boldsymbol{Y}$)
  - Y is represented as a mathematical function of X.



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

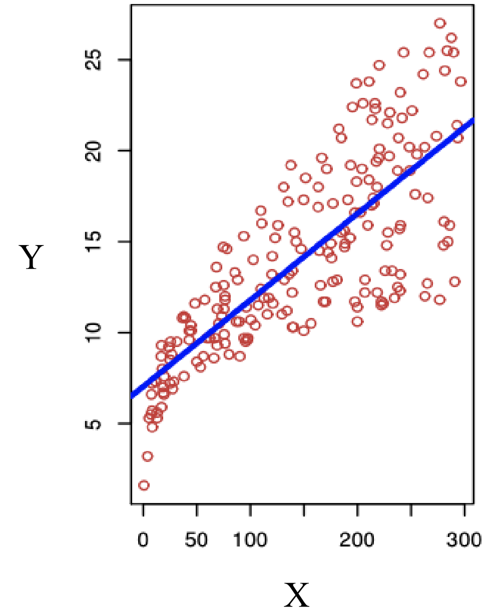
Years of education (X) $\longrightarrow$ Income (Y)

Simple Linear Regression

- **Simple Linear Regression**: a linear relationship is assumed between X and Y.

- ***Y* as a linear function of X:**

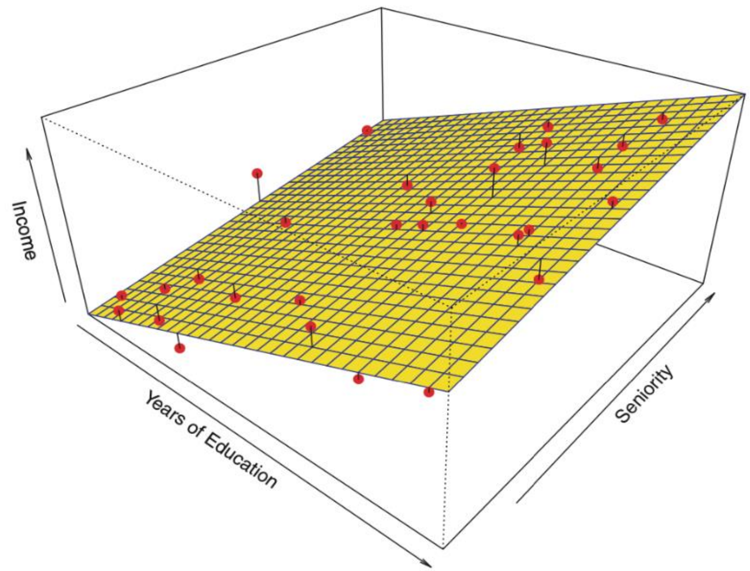
$$Y = f(X) + \varepsilon$$



- Different names for X and Y:
  - X: independent variable, predictor, or feature.
  - Y: dependent variable, target variable, or response.

Multiple Linear Regression

- **Multiple Linear Regression**: a linear relationship is assumed between X(s) and Y.



A linear model fit

Multiple Predictors:

- X_1: Years of education
- X_2: Seniority
- Y: Income

- Y as a linear function of X: $$Y = f(X) + \varepsilon$$

Simple Linear Regression

- **Task:**
  - Build a model to predict a market's ***sales revenue*** according to the ***advertising expenditure on TV***?
- **Data:**
  - Sales revenues in 200 different markets;
  - TV ads expenditure in each of those markets.

SIMPLE LINEAR REGRESSION

FIT THE MODEL TO THE DATA

The model $y = w_0 + w_1 * x + \varepsilon$ $y \approx w_0 + w_1 * x$

$\text{sales} = w_0 + w_1 * TV + \varepsilon$

The unknown parameters w_0: intercept, w_1 : coefficient (slope)

Average y when $x = 0$

The average change in y, for 1 unit change in x

Predicted Y $\hat{y} = \hat{w}_0 + \hat{w}_1 * x$

$\text{predicted sales} = \hat{w}_0 + \hat{w}_1 * TV$

Question: how can we find the best parameter values, i.e., $\hat{w}_0$ and $\hat{w}_1$?

Parameter Learning

- For each instance (i) in the training data, the **Residual** (ε_i) is:

$$\varepsilon_i = y_i - \hat{y}_i \quad \text{where} \quad \hat{y}_i = \widehat{w}_0 + \widehat{w}_1 * x_i$$

- Then **Residual Sum of Squares (RSS)** for entire training data is:

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

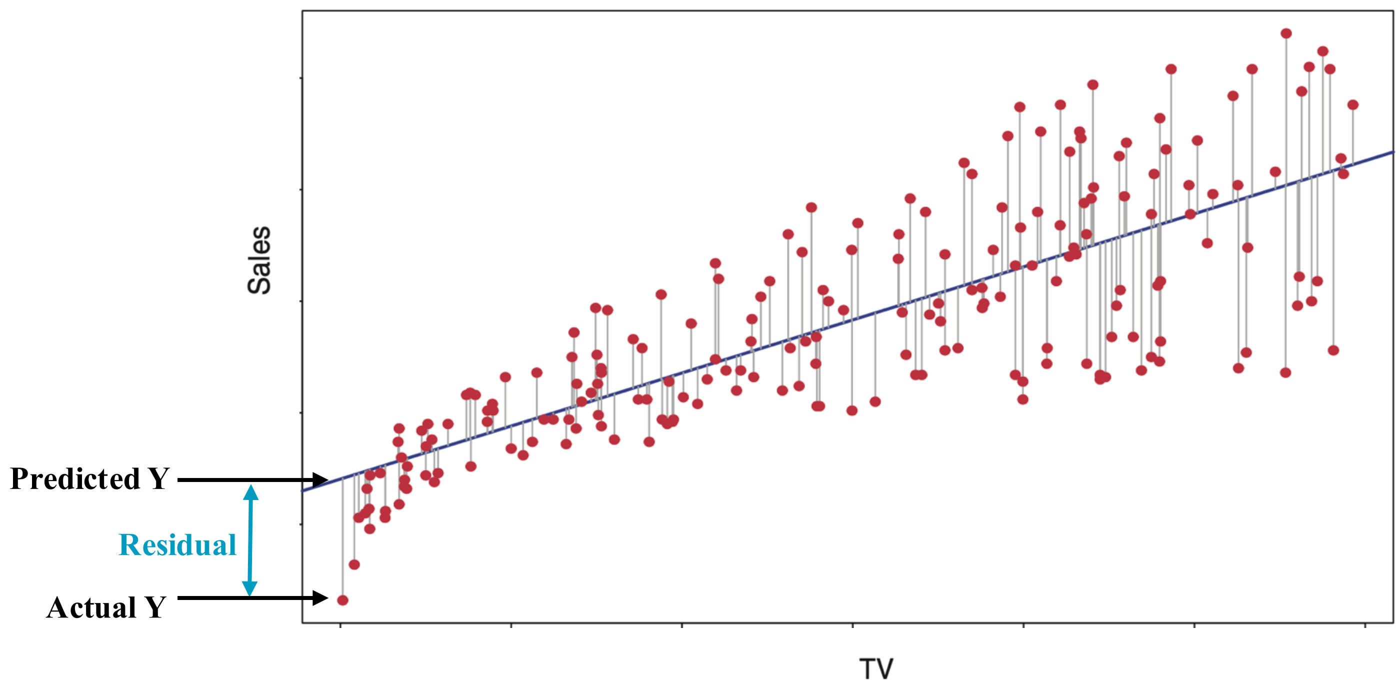
- In the modeling process, the best value for the parameters are learnt by optimizing the following **Objective Function**:

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

  - The algorithm is called "**Ordinary Least Square Estimator**".

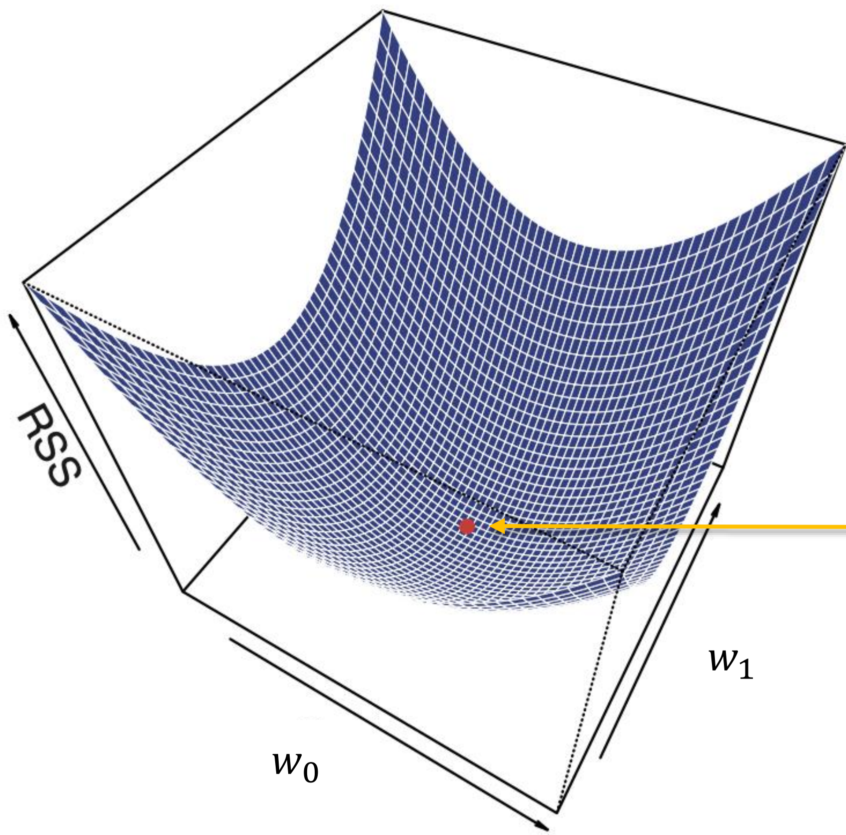
Parameter Learning

- Step 1: for each line (a model), calculated the squared **residuals** for all training instances and sum them up for **RSS**.
- Step 2: find a line (a model) which returns the **minimized RSS**.
  - The intercept, slope for this line are the best parameter values.



Parameter Learning

For Simple Linear Regression



The red dot in the bottom of the net corresponds to best parameter values, which minimized the RSS.

How could the computer find the best parameter values (the red dot)?

The gradient descent optimization algorithm

Model Evaluation

- **MSE:** **mean squared error** can be calculated on train and test data.
  - MSE is averaged RSS.

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

n = number of instances

y_i = observed target value

$\hat{y}_i$ = predicted target value

- $\boldsymbol{R^2}$: The proportion of variance in Y predicted/explained by X.
  - R^2 can also be calculated on both training and test data.

$$R^2 = \frac{\mathrm{TSS} - \mathrm{RSS}}{\mathrm{TSS}} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

where

$$\mathrm{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

  - R^2 is more interpretable as it is often in the range [0,1].
    - $R^2 = 0$ means the model fails to explain any variance in Y (TSS = RSS)
    - $R^2 = 1$ means the model explains all variance in Y (RSS = 0)

For an arbitrarily bad model (e.g., when it is even worse than a constant function that always predict $\bar{y}$), R^2 can be negative!

Multiple Linear Regression

- **Task:**
  - Build a model to predict ***<u>sales revenue</u>*** according to the ***<u>advertising expenditure on TV, radio and newspaper</u>***?
- **Data:**
  - Sales revenues in 200 different markets;
  - Advertising expenditure in each market for three channels: TV, radio and newspaper.

MULTIPLE LINEAR REGRESSION

FIT THE MODEL TO THE DATA

The model $y = w_0 + w_1 x_1 + \cdots + w_p x_p + \varepsilon$ $\quad y \approx w_0 + w_1 x_1 + \cdots + w_p x_p$

$$\text{sales} = w_0 + w_1 * \text{TV} + w_2 * \text{radio} + w_3 * newspaper + \varepsilon$$

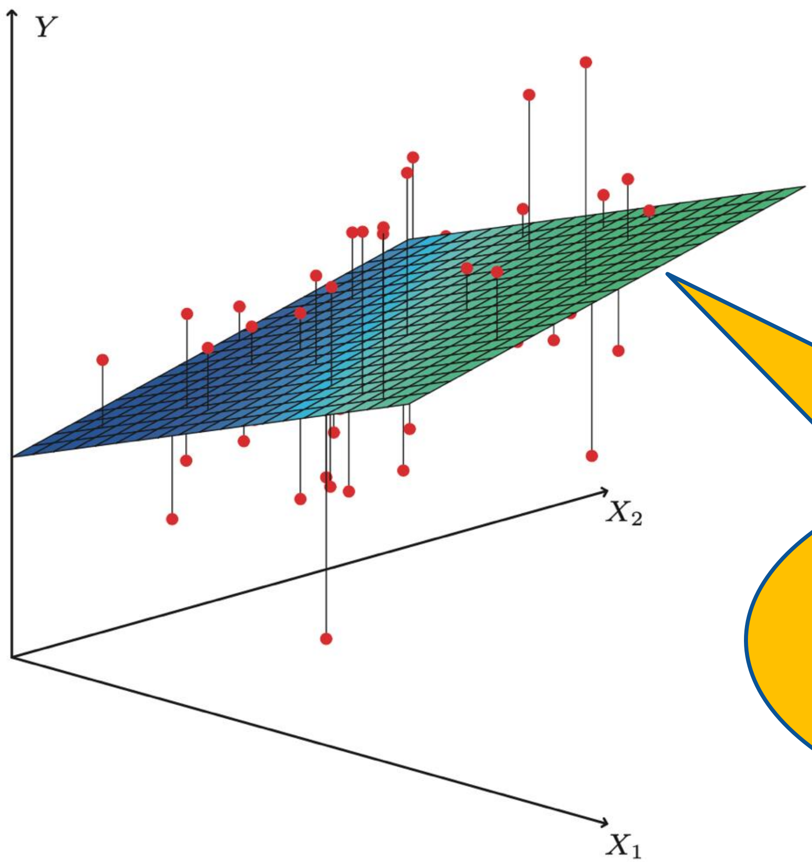
Predicted Y $\quad \hat{y} = \widehat{w}_0 + \widehat{w}_1 * x_1 + \ldots + \widehat{w}_p * x_p$

$$\text{predicted sales} = \widehat{w}_0 + \widehat{w}_1 * TV + \widehat{w}_2 * \text{radio} + \widehat{w}_3 * newspaper$$

- Parameter interpretation:
  - w_o : average y when all x = 0.
  - w_p : average change in y for 1 single unit change in x_p, holding all other features constant.

Find the Best Fit for Multiple Linear Regression

Least Squares Approach



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

The linear line in simple linear regression becomes a linear hyperplane in multiple linear regression.

Accommodating Non-linear Relationship

Categorical Features

- We'd like to predict house price (USD) with (1) distance to city center (km), (2) house age (year), (3) number of rooms, (4) top school within 2km (1 for yes, 0 for no)?
  - How to prepare data and interpret the coefficient for school (i.e., w_4)?

$$house\ price = w_0 + w_1\, dist + w_2 age + w_3 room + w_4 school + \varepsilon$$

Intercept	2.733e+04
dist	-206.6458
age	-760.5100
room	321.3594
school	548.7643

Compared to houses without top schools, houses with top schools in the neighborhood will be valued 548.76 USD higher on average, holding other features unchanged.

Categorical Features

- For categorical features with multiple levels, convert them into multiple dummy (indicator) variables.
  - pandas.get_dummies()

df

	dist	age	room	school	price
0	2.000000	12	3	T1	17748.526691
1	2.048048	15	3	T2	15734.586643
2	2.096096	21	4	T3	17801.694257
3	2.144144	0	3	T2	20155.145308
4	2.192192	3	4	T1	18883.183762

pd.get_dummies(df)

	dist	age	room	price	school_T1	school_T2	school_T3
0	2.000000	12	3	17748.526691	1	0	0
1	2.048048	15	3	15734.586643	0	1	0
2	2.096096	21	4	17801.694257	0	0	1
3	2.144144	0	3	20155.145308	0	1	0
4	2.192192	3	4	18883.183762	1	0	0

Polynomial Features: Interaction Terms

- Is there synergy/interaction effect between distance and age?
  - Create polynomial features (e.g., interaction terms).

$$house\ price = w_0 + w_1 dist + w_2 age + w_3 room + w_4 school + w_5 dist * age + \varepsilon$$

Intercept	2.261e+04
dist	-23.1048
age	-406.4423
room	333.3230
school	487.0140
dist age	-13.8050

The effect of age on price is affected by distance:

With 1 unit increase in distance, the effect of age on house price further drops by 13.81.

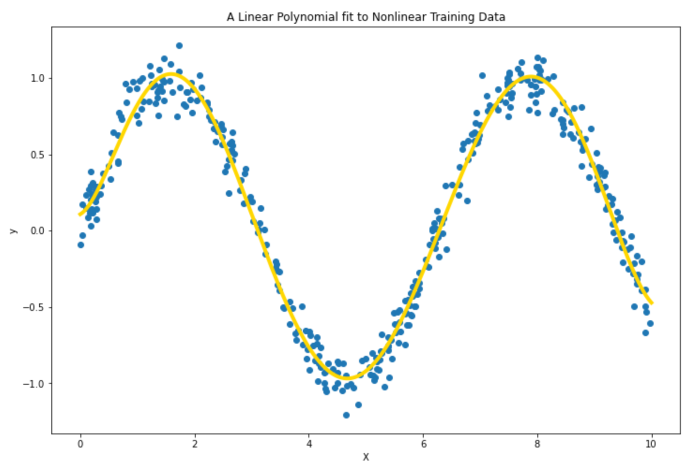
Other features unchanged, for 1-year increase in age:

- Price for houses in the city center (dist = 0) will drop by 406.44 USD .
- Price for houses 1 km away from city center (dist = 1) will drop by 420.25 USD.
- Price for houses 2 km away from city center (dist = 2) will drop by 434.05 USD.
- …

Polynomial Features: Exponential Terms

- What if the relationship between X and Y is not linear?
  - Create polynomial features (e.g., exponential terms): X, X^2, X^3….

$$y = w_0 + w_1 X + w_2 X^2 + w_3 X^3 + \cdots + w_7 X^7 + \varepsilon$$



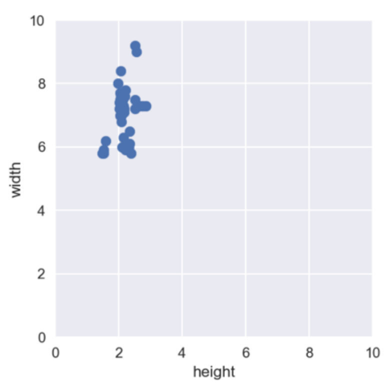
- Note this is still a linear model.
  - All parameters are constant value: i.e., the relationship between target and each new feature (X, X^2, …, X^7) is linear.
  - The model looks linear if it is visualized in an 8-dimensional space.

The Need for Feature Scaling

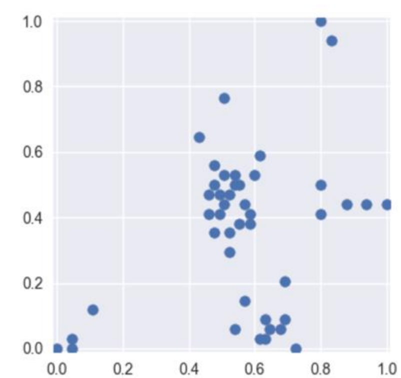
- For some models(e.g., regularized regression, k-NN, SVM, neural networks), it is important that **all features are on the same scale**.
  - Faster convergence in model training.
  - More uniform or “fair” influence for all features.
    - With two features on different scales, difficult to compare their coefficients (i.e., average change in y for 1 unit change in feature)?

- Lots of methods are available: e.g., **MinMax scaling**:

$$x_i' = (x_i - x_i^{MIN})/(x_i^{MAX} - x_i^{MIN})$$



Raw data



Scaled data
with MinMaxScaler