MULTI-LINEAR & POLYNOMIAL REGRESSION

LECTURE 1
SECTION 2
JUNE 1ST

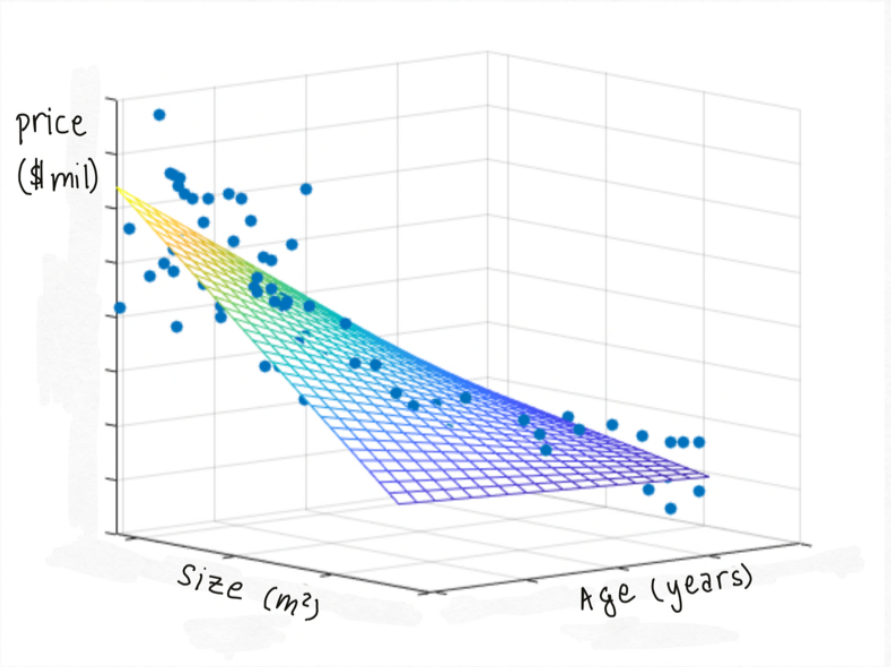
INSTITUTE FOR APPLIED
COMPUTATIONAL SCIENCE
AT HARVARD UNIVERSITY

UNIVERSITY of
RWANDA

MULTILINEAR REGRESSION

MULTILINEAR REGRESSION:

What if the target y depends on multiple predictors?



A multi-linear model for y given $\mathbf{x} = [1, x_2, x_3, \ldots, x_D]$ is:

$$y = w_D x_D + w_{D-1} x_{D-1} + \cdots + w_1 x_1 + w_0$$

often we write this in vector notation:

$$y = \mathbf{w}^T \mathbf{x}$$

where $\mathbf{w} = [w_0, w_1, \ldots, w_D]$.

MODEL TRAINING: MULTI-LINEAR REGRESSION

Given a training dataset $\{(x^{(1)}, y^{(1)}), \dots (x^{(N)}, y^{(N)})\}$ where

$$x_n = [x_1^{(n)}, x_2^{(n)}, \dots, x_D^{(n)}]$$

We find the optimal multi-linear model: $y = w^T x$

Find $w^* = \underset{w}{\operatorname{argmin}}\, \mathcal{L}(w) = \underset{w}{\operatorname{argmin}}\, \frac{1}{N} \sum_{n=1}^{N} (y^{(n)} - w^T x^{(n)})^2$

1. Compute the gradient: $\nabla_w \mathcal{L} = [\frac{\partial \mathcal{L}}{\partial w_0}, \frac{\partial \mathcal{L}}{\partial w_1}, \dots, \frac{\partial \mathcal{L}}{\partial w_D}]$

$$\nabla_w \mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} 2 (y^{(n)} - w^T x^{(n)}) x^{(n)}$$
$$= (y - w^T x) x$$

where $y = [y_1, \dots, y_n]$ is the vector of target values.

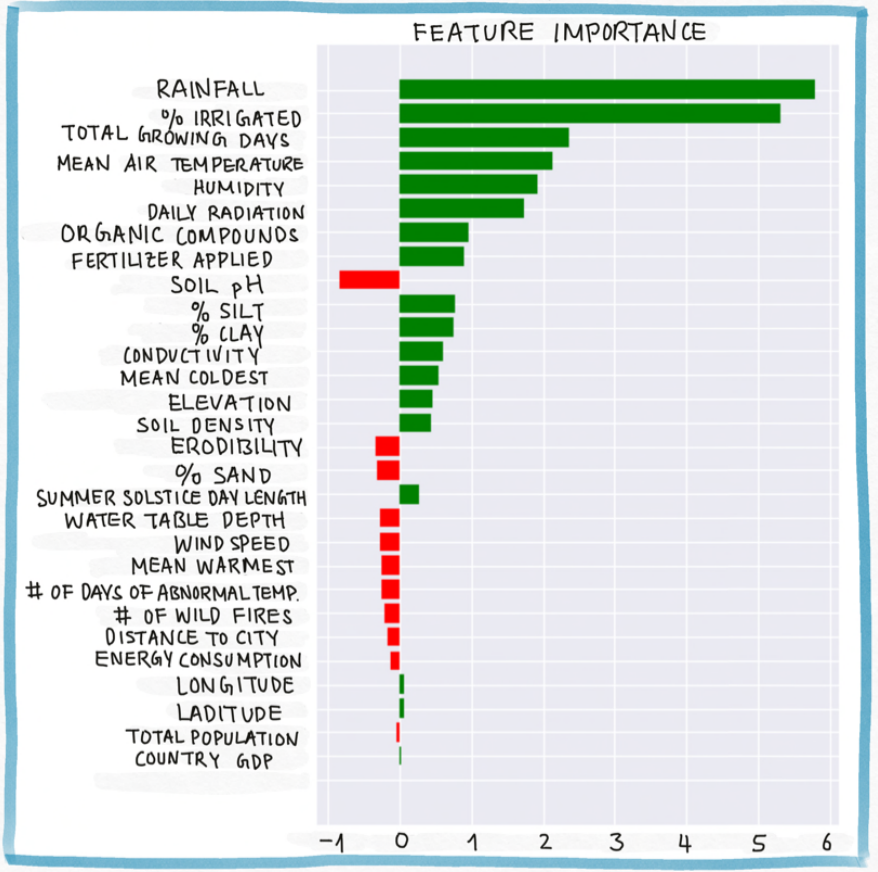
where $X = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_D^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_D^{(2)} \\ & & \vdots & & \\ 1 & x_1^{(N)} & x_2^{(N)} & \cdots & x_D^{(N)} \end{bmatrix}$ is the matrix of predictor values

2. Solve for stationary points: $\nabla_w \mathcal{L} = 0$

$$\boxed{w^* = (X^T X)^{-1} X^T y}$$

INTERPRETING MULTI-LINEAR REGRESSION:

For linear models, it's important to interpret the model parameters.



When there are a large number of predictors: $X_1, \ldots, X_D$, there will be a large number of model parameters: $w_1, \ldots, w_D$.

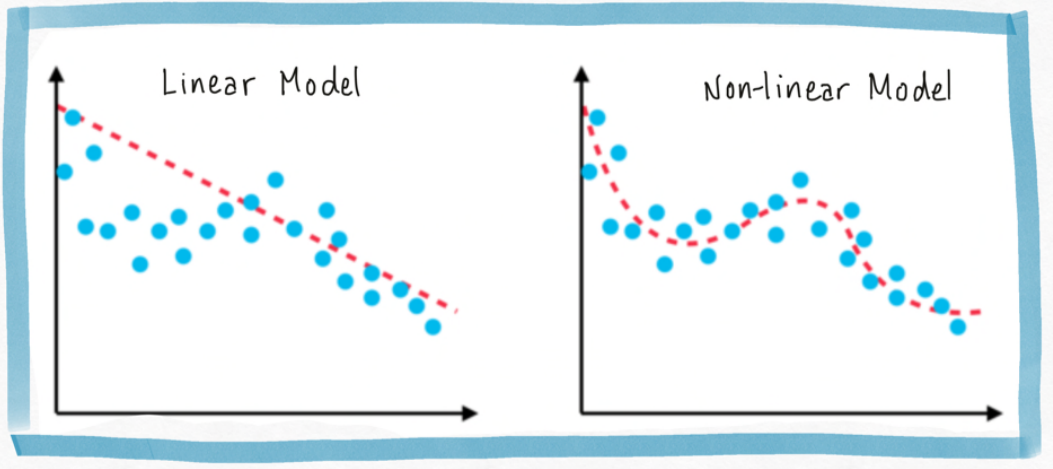
Looking at the values of $\mathbf{w}$ is impractical, so we visualize these values a feature importance graph.

The feature importance graph shows which predictor has the most impact on the model's prediction.

POLYNOMIAL REGRESSION

FITTING NONLINEAR DATA:

Multi-linear models can fit large datasets with many predictors. But the relationship between predictor and target isn't always linear.



We want a model:

$y = f_{\mathbf{w}}(x)$

where f is a non-linear function, and $\mathbf{w}$ is a vector of the parameters of f.

POLYNOMIAL REGRESSION:

The simplest type of non-linear function is polynomial:

$$y = w_K x^K + w_{K-1} x^{K-1} + \cdots + w_2 x^2 + w_1 x^1 + w_0$$

But this looks a lot like multi-linear regression where the predictors are powers of x!

Example: Selling price v.s. size

X (m^2)	y ($mil)
50	0.5
60	1.6
70	5.8
80	8.3

Training Data

X	X^2	X^3	y
50	2500	12500	0.5
60	3600	21600	1.6
70	4900	34300	5.8
80	6400	51200	8.3

Transformed Training Data

We transform the training data by adding powers of the single predictor x. Then fit a multi-linear model on the transformed data:

$$y = w_3 x^3 + w_2 x^2 + w_1 x + w_0$$

MODEL TRAINING: POLYNOMIAL REGRESSION

Given a dataset $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(N)}, y^{(N)})\}$, we find the optimal polynomial model: $y = w_K x^K + w_{K-1} x^{K-1} + \cdots + w_1 x + w_0$.

1. We transform the data by adding new predictors: $\tilde{x} = [1, \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_K]$ where $\tilde{x}_K = x^K$.
2. Find the parameters $w^* = [w_0^*, w_1^*, \ldots, w_K^*]$ by multi-linear regression

$$w^* = \arg\min_w \mathcal{L}(w) = \arg\min_w \frac{1}{N} \sum_{n=1}^{N} (y^{(n)} - w^T \tilde{x}^{(n)})^2$$

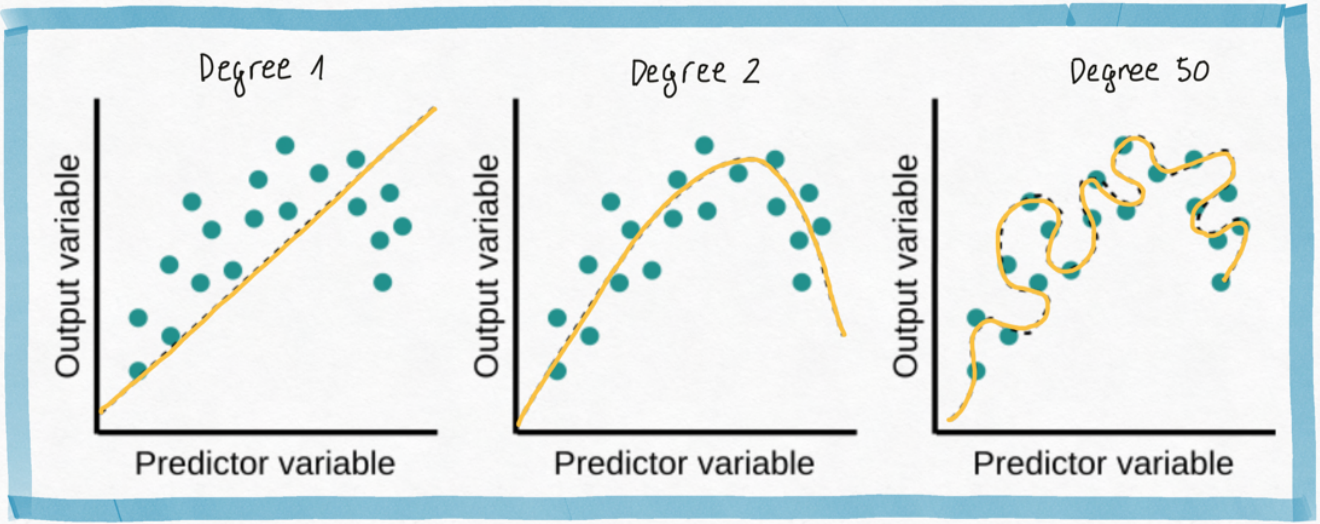
$$= (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T y$$

where $y = [y^{(1)}, \ldots, y^{(N)}]$ is the vector of response values,

where $\tilde{X} = \begin{bmatrix} 1 & \tilde{x}_1^{(1)} & \tilde{x}_2^{(1)} & \cdots & \tilde{x}_K^{(1)} \\ 1 & \tilde{x}_1^{(2)} & \tilde{x}_2^{(2)} & \cdots & \tilde{x}_K^{(2)} \\ & & \vdots & & \\ 1 & \tilde{x}_1^{(N)} & \tilde{x}_2^{(N)} & \cdots & \tilde{x}_2^{(N)} \end{bmatrix}$ is the matrix of predictor values in the transformed dataset.

MODEL SELECTION: CHOOSING THE DEGREE

Fitting a polynomial model requires choosing a degree.



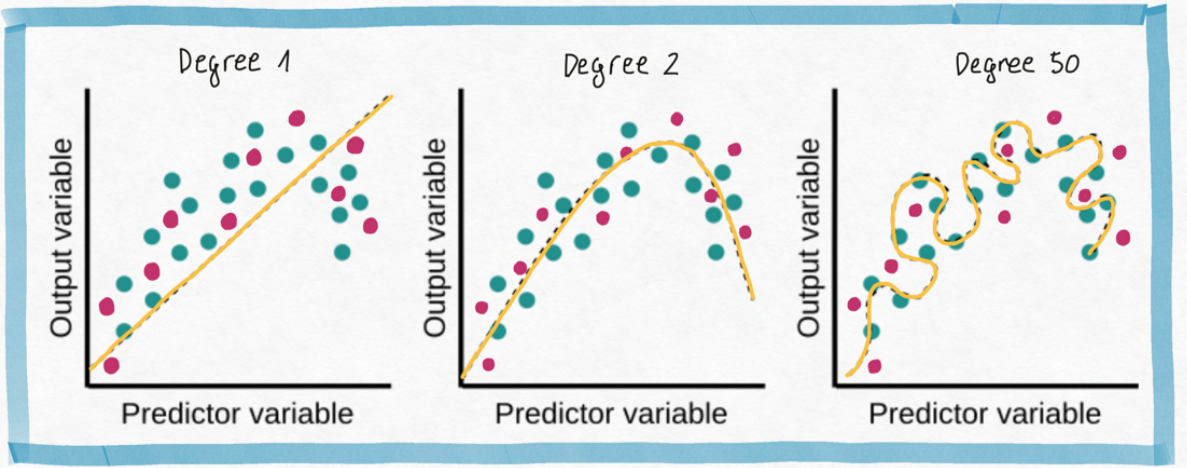
Underfitting: when the degree is too low, the model cannot fit the trend.

we want a model that fits the trend and ignores the noise?

Overfitting: when the degree is too high, the model fits all the noisy data points

OVERFITTING OR UNDERFITTING?

If the dataset contains more than one predictor then we cannot easily visualize the polynomial $y = f(X)$.



Train MSE: 2.1
Test MSE: 1.9
When we underfit, the train and test errors are similar and both large.

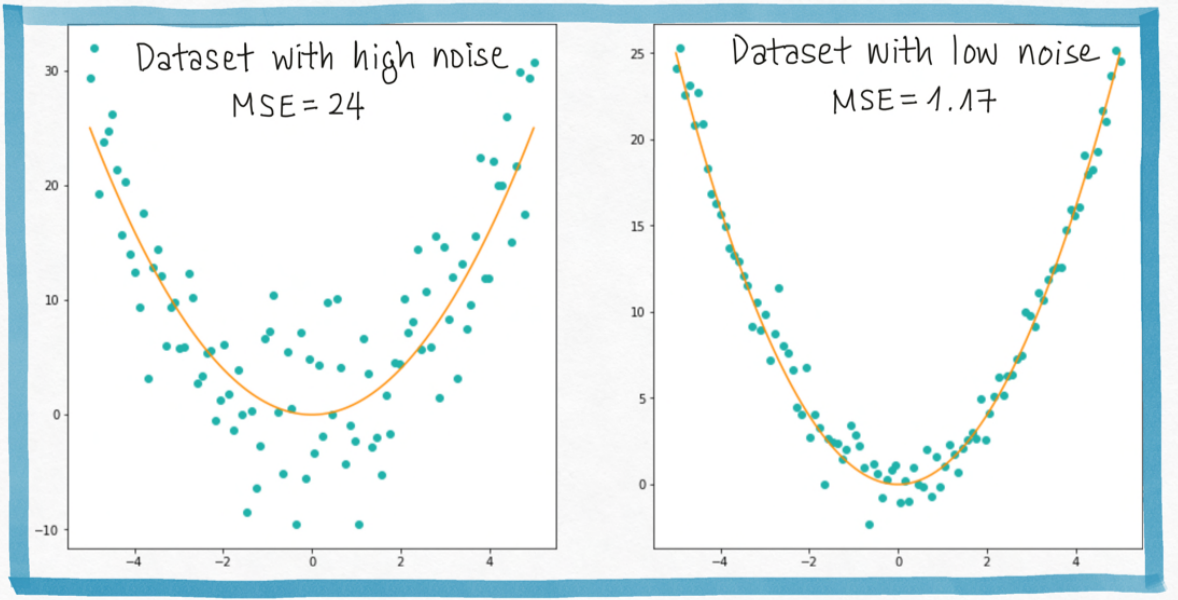
Train MSE: 0.5
Test MSE: 0.65
When we fit the trend correctly both train and test error "should" be small.

Train MSE: 0
Test MSE: 0.9
When we overfit, the train error is very small, and test error is <u>much</u> larger

UNDERFITTING OR HIGH NOISE?

We said that an ideal model would have similar train and test MSE, and both would be small; an underfitting model would have similar train and test MSE but both would be large.

Is this always true?



MSE is not a reliable way to detect underfitting, we need another method that accounts for noise in data.