LECTURE 11

Machine Learning

Understanding the usefulness of models and the simple linear regression model

Data Science, Spring 2024 @ Knowledge Stream

Sana Jabbar

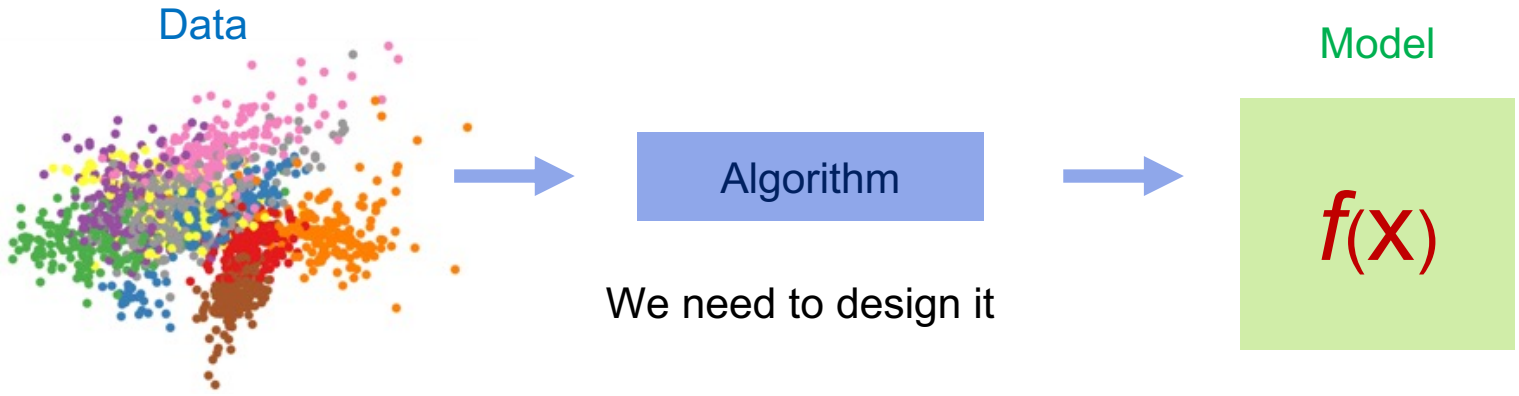
Goals for this Lecture

Lecture 11

- What is a Model?
- **The Modeling Process**
  - Choose a Model
  - Choose a Loss Function
  - **Fit the Model**
  - Evaluate the Model

Machine learning framework

Given examples (training data), develop a machine learning system to discover patterns



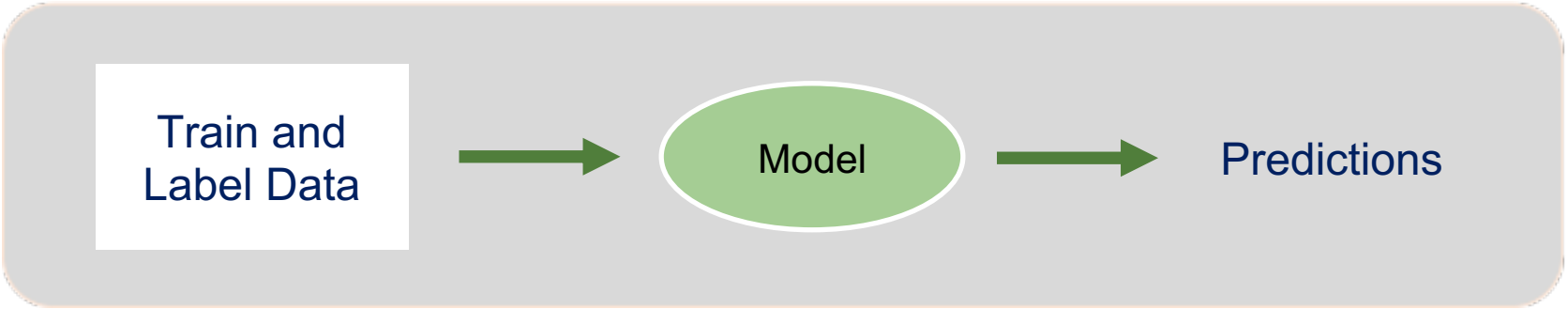
Algorithms vs Model

Linear regression algorithm produces a model, that is, a vector of values of the coefficients of the model.

Neural network along with backpropagation + gradient descent: produces a model comprised of a trained (weights assigned) neural network.

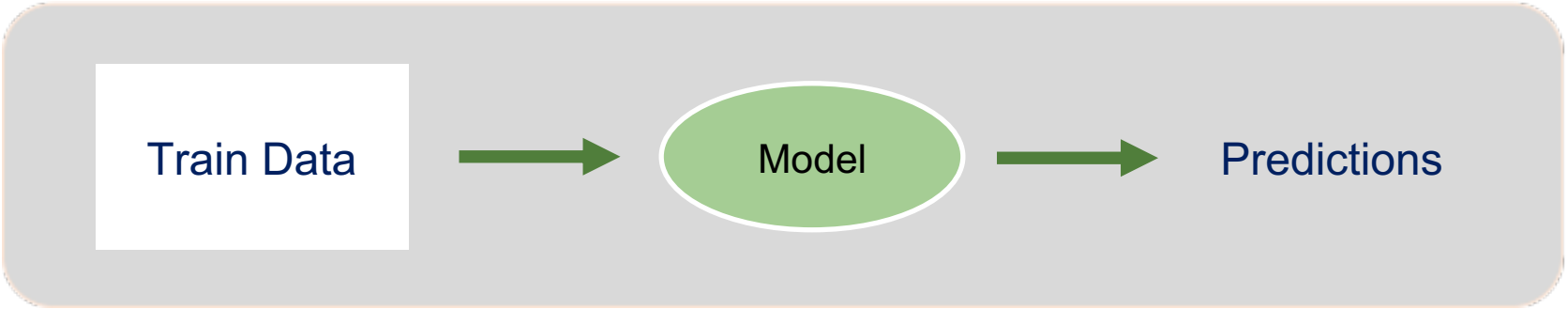
Supervised Learning

The learning algorithm would receive a set of inputs along with the corresponding correct to train a model



Unsupervised Learning

The learning algorithm would receive only a set of inputs to train a model

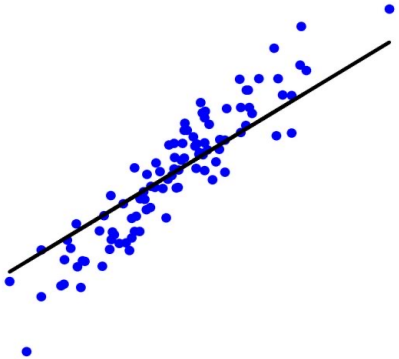


Regression

Quantitative Prediction on a continuous scale

Examples: Prediction of

1. Age of a person from his/her photo
2. Price of 10 Marla, 5-bedroom house in 2050
3. USD/PKR exchange rate after one week
4. Efficacy of Pfizer Covid vaccine
5. Average temperature/Rainfall during monsoon
6. Cumulative score in ML course
7. Probability of a decrease in the electricity prices in Pakistan



What do all these problems have in common?

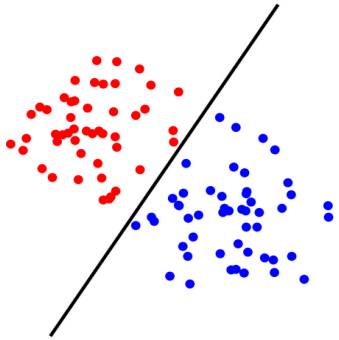
Continuous outputs

Classification:

Predicting a categorical output is called classification

Examples: Prediction of

1. Gender of a person using his/her photo or hand-writing style
2. Spam filtering
3. Temperature/Rainfall normal or abnormal during monsoon
4. Letter grade in a course
5. Decrease expected in electricity prices in Pakistan next year
6. More than 10000 Steps taken today



What do all these problems have in common?
Discrete outputs: Categorical Yes/No (Binary Classification)
Multi-class classification: multiple classes

Supervised Learning Setup

- In these regression or classification problems, we have
- Inputs – referred to as Features
- Output – referred to as Label
- Training data – (input, output) for which the output is known and is used for training a model by the ML algorithm
- A Loss, an objective, or a cost function – determines how well a trained model approximates the training data
- Test data – (input, output) for which the output is known and is used for the evaluation of the performance of the trained model

Supervised Learning Setup

Predict Stock Index Price

- Features (Input)
- Labels (Output)
- Training data
- Validation data

Interest_Rate	Unemployment_Rate	Stock_Index_Price
2.75	5.3	1464
2.5	5.3	1394
2.5	5.3	1357
2.5	5.3	1293
2.5	5.4	1256
2.5	5.6	1254
2.5	5.5	1234
2.25	5.5	1195
2.25	5.5	1159
2.25	5.6	1167
2	5.7	1130
2	5.9	1075
2	6	1047
1.75	5.9	965
1.75	5.8	943
1.75	6.1	958
1.75	6.2	971
1.75	6.1	949
1.75	6.1	884
1.75	6.1	866
1.75	6.2	?
1.75	6.2	?
1.75	6.1	?

Using the adopted notation, we can formalize the supervised machine learning setup. We represent the entire training data as

$$D = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\} \subseteq \mathcal{X}^d \times \mathcal{Y}$$

Here $\mathcal{X}^d$ - d dimensional feature space and $\mathcal{Y}$ is the label space.

Regression: $\mathcal{Y} = \mathbf{R}$ (prediction on continuous scale)

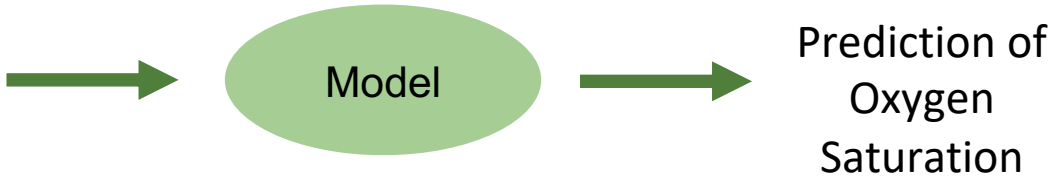
Classification: $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, 1\}$ or $\mathcal{Y} = \{1, 2\}$ (Binary classification)

$\mathcal{Y} = \{1, 2, \ldots, M\}$ (M-class classification)

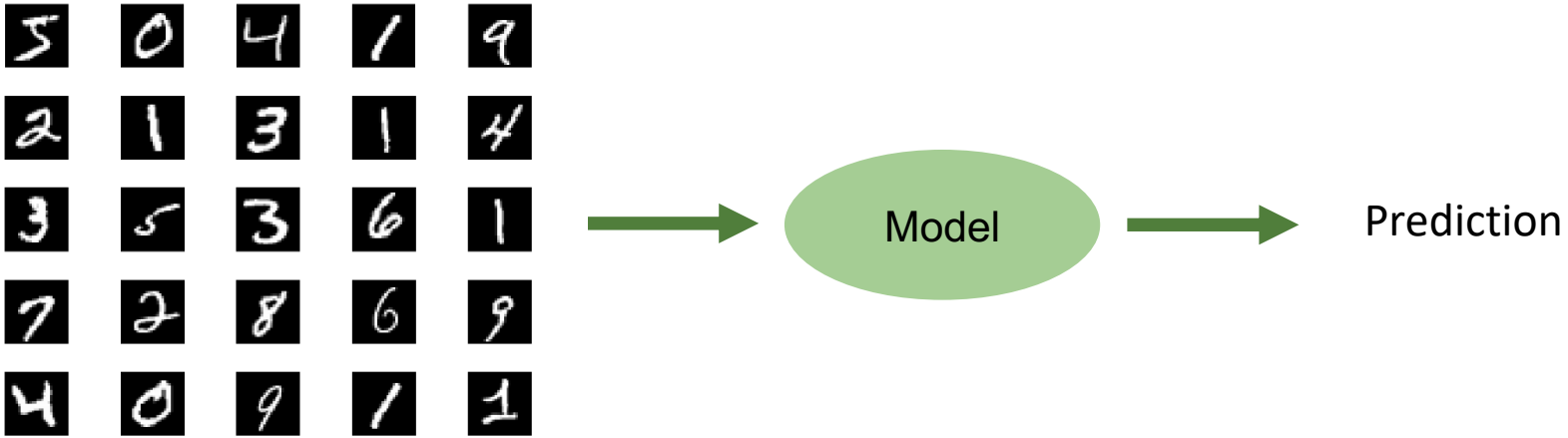
Example:

Data of 200 Patients:

- Age of the patient
- Cholesterol levels
- Glucose levels
- BMI
- Height
- Heart Rate
- Calories intake
- No. of steps taken



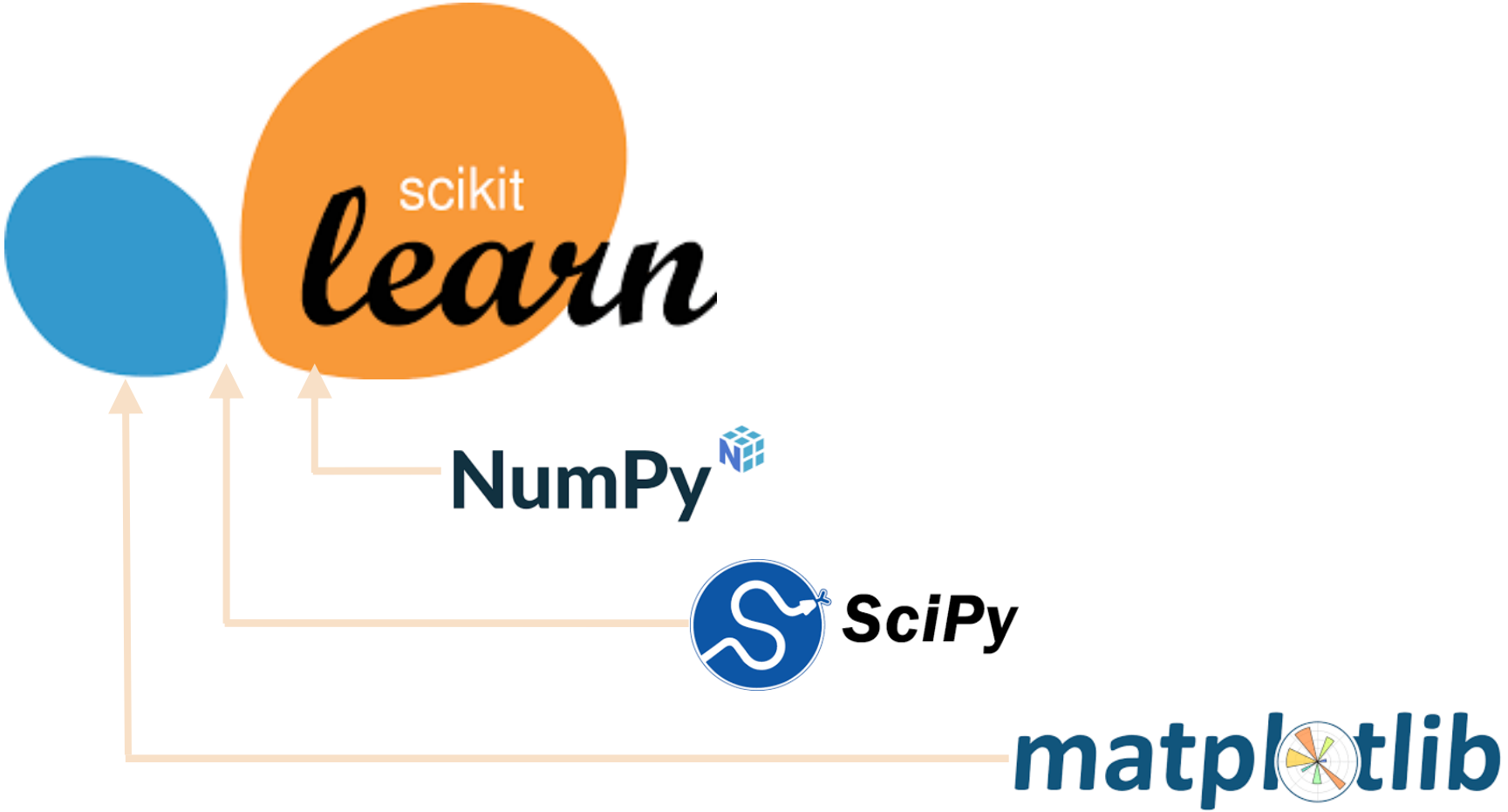
Example:



MNIST Data:

- Each sample 28x28 pixel image
- 60,000 training data
- 10,000 testing data

Scikit-learn



Regression:

```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.linear_model import ElasticNet
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
```

Regression

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split


X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Linear regression
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)
```

Today task is to repeat the previously provided notebooks with LinearRegression, Ridge, Lasso, ElasticNet

Fit the Model

Lecture 11

- What is a Model?
- **The Modeling Process**
  - Choose a Model
  - Choose a Loss Function
  - **Fit the Model**
  - Evaluate the Model

Minimizing MSE for the SLR Model

Recall: we wanted to pick the **regression line** $\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 x$

To minimize the (sample) **Mean Squared Error**: $MSE(\theta_0, \theta_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (\theta_0 + \theta_1 x_i))^2$

To find the best values, we set derivatives equal to zero to **obtain the optimality conditions:**

$$\frac{\partial}{\partial \theta_0} MSE = 0$$

$$\frac{\partial}{\partial \theta_1} MSE = 0$$

Partial Derivative of MSE with Respect to θ_0, θ_1

$$\frac{\partial}{\partial \theta_0} MSE = \frac{\partial}{\partial \theta_0} \frac{1}{n} \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i)^2$$

Derivative of sum is sum of derivatives

$$= \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial \theta_0} (y_i - \theta_0 - \theta_1 x_i)^2$$

Chain rule

$$= \frac{1}{n} \sum_{i=1}^{n} 2(y_i - \theta_0 - \theta_1 x_i)(-1)$$

Simplify constants

$$= \boxed{-\frac{2}{n} \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i)}$$

$$\frac{\partial}{\partial \theta_1} MSE = \frac{\partial}{\partial \theta_1} \frac{1}{n} \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i)^2$$

Derivative of sum is sum of derivatives

$$= \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial \theta_1} (y_i - \theta_0 - \theta_1 x_i)^2$$

Chain rule

$$= \frac{1}{n} \sum_{i=1}^{n} 2(y_i - \theta_0 - \theta_1 x_i)(-x_i)$$

Simplify constants

$$= \boxed{-\frac{2}{n} \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i) x_i}$$

Estimating Equations

To find the best values, we set derivatives equal to zero to **obtain the optimality conditions:**

$$0 = \frac{\partial}{\partial \theta_0} MSE = -\frac{2}{n}\sum_{i=1}^{n}\left(y_i - \hat{\theta}_0 - \hat{\theta}_1 x_i\right) \iff \frac{1}{n}\sum_i y_i - \hat{y}_i = 0$$

"Equivalent"

$$0 = \frac{\partial}{\partial \theta_1} MSE = -\frac{2}{n}\sum_{i=1}^{n}\left(y_i - \hat{\theta}_0 - \hat{\theta}_1 x_i\right)x_i \iff \frac{1}{n}\sum_i (y_i - \hat{y}_i)x_i = 0$$

Estimating equations

To find the best θ_0, θ_1, we need to solve the **estimating equations** on the right.

From Estimating Equations to Estimators

Goal: Choose θ_0, θ_1 to solve two estimating equations:

$$\frac{1}{n}\sum_i y_i - \hat{y}_i = 0 \quad \boxed{1} \qquad \text{and} \qquad \frac{1}{n}\sum_i (y_i - \hat{y}_i)x_i = 0 \quad \boxed{2}$$

$\boxed{1}$

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{\theta}_0 - \hat{\theta}_1 x_i\right) = 0$$

Separating terms

$$\iff \left(\frac{1}{n}\sum_{i=1}^{n} y_i\right) - \hat{\theta}_0 - \hat{\theta}_1\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right) = 0$$

$$\iff \bar{y} - \hat{\theta}_0 - \hat{\theta}_1 \bar{x} = 0$$

$$\iff \hat{\theta}_0 = \bar{y} - \hat{\theta}_1 \bar{x}$$

From Estimating Equations to Estimators

Goal: Choose θ_0, θ_1 to solve two estimating equations:

$$\frac{1}{n}\sum_i y_i - \hat{y}_i = 0 \quad \boxed{1} \qquad \text{and} \qquad \frac{1}{n}\sum_i (y_i - \hat{y}_i)x_i = 0 \quad \boxed{2}$$

Now, let's try: 2 - 1 * $\bar{x}$

$$\frac{1}{n}\sum_i (y_i - \hat{y}_i)x_i - \frac{1}{n}\sum_i (y_i - \hat{y}_i)\bar{x} = 0 \iff \frac{1}{n}\sum_i (y_i - \hat{y}_i)(x_i - \bar{x}) = 0$$

$$\left(\text{using } \hat{y}_i = \hat{\theta}_0 + \hat{\theta}_1 x_i\right) \Rightarrow \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{\theta}_0 - \hat{\theta}_1 x_i\right)(x_i - \bar{x}) = 0$$

$$\left(\text{using } \hat{\theta}_0 = \bar{y} - \hat{\theta}_1 \bar{x}\right) \Rightarrow \frac{1}{n}\sum_i \left(y_i - \bar{y} + \hat{\theta}_1 \bar{x} - \hat{\theta}_1 x_i\right)(x_i - \bar{x}) = 0$$

$$\Rightarrow \frac{1}{n}\sum_i \left((y_i - \bar{y}) - \hat{\theta}_1 (x_i - \bar{x})\right)(x_i - \bar{x}) = 0$$

$$\Rightarrow \frac{1}{n}\sum_i \left[(y_i - \bar{y})(x_i - \bar{x}) - \hat{\theta}_1 (x_i - \bar{x})^2\right] = 0$$

$$\Rightarrow \frac{1}{n}\sum_i (y_i - \bar{y})(x_i - \bar{x}) = \hat{\theta}_1 \frac{1}{n}\sum_i (x_i - \bar{x})^2$$

Plug in definitions of correlation and SD:

$$r\sigma_y\sigma_x = \hat{\theta}_1\sigma_x^2$$

Solve for $\hat{\theta}_1$:

$$\hat{\theta}_1 = r\frac{\sigma_y}{\sigma_x}$$

Reminder

$$\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$r = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma_x}\right)\left(\frac{y_i - \bar{y}}{\sigma_y}\right)$$

Estimating Equations

Estimating equations are the equations that the model fit has to solve. They help us:

- Derive the estimates.
- Understand what our model is paying attention to.

$$\frac{1}{n}\sum_i y_i - \hat{y}_i = 0$$

$$\frac{1}{n}\sum_i (y_i - \hat{y}_i)x_i = 0$$

For SLR:

- The residuals should **average to zero** (otherwise we should fix the intercept!)

The Modeling Process

1. Choose a model ✅

How should we represent the world?

$$\hat{y} = \theta_0 + \theta_1 x$$

SLR model

2. Choose a loss function ✅

How do we quantify prediction error?

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

Squared loss

3. Fit the model ✅

How do we choose the best parameters of our model given our data?

$$\hat{R}(\theta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - (\theta_0 + \theta_1 x))^2$$

MSE for SLR

4. Evaluate model performance

How do we evaluate whether this process gave rise to a good model?

$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 x \quad \begin{cases} \hat{\theta}_0 = \bar{y} - \hat{\theta}_1 \bar{x} \\ \hat{\theta}_1 = r \dfrac{\sigma_y}{\sigma_x} \end{cases}$$