Robotics & Internet-of-Things

جامعة الأمير سلطان PRINCE SULTAN UNIVERSITY

CS316: INTRODUCTION TO AI AND DATA SCIENCE

CHAPTER 7
STATISTICAL LEARNING

LECTURE
SUPERVISED LEARNING
VS.
UNSUPERVISED LEARNING

Prof. Anis Koubaa

Nov 2024

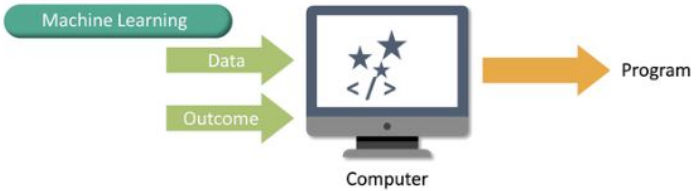
www.riotu-lab.org

Learning the Optimal Prediction Function

Problem Statement

- **Challenge of Optimal Prediction:**
  - Optimal function g^* depends on the unknown joint distribution of X, Y.
  - Practical approach uses a finite sample $T = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, termed as the training set.
- **Training Set Notation:**
  - Random training set: T.
  - Deterministic outcome: τ or τ_n (to emphasize size).

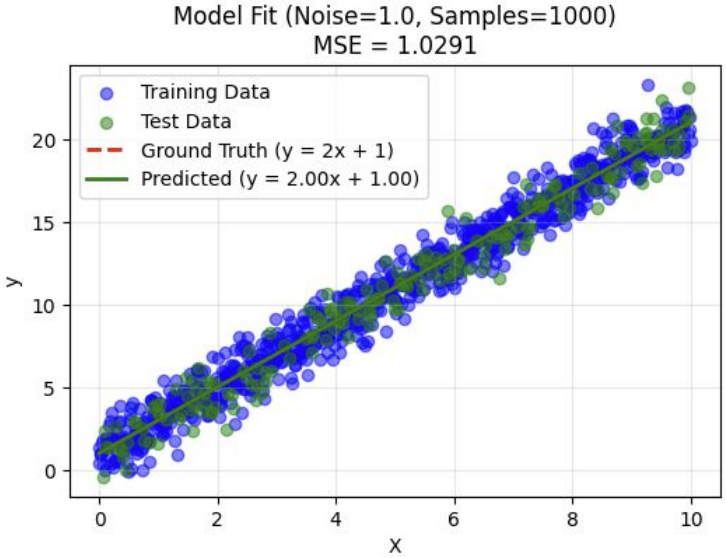
$$\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$$



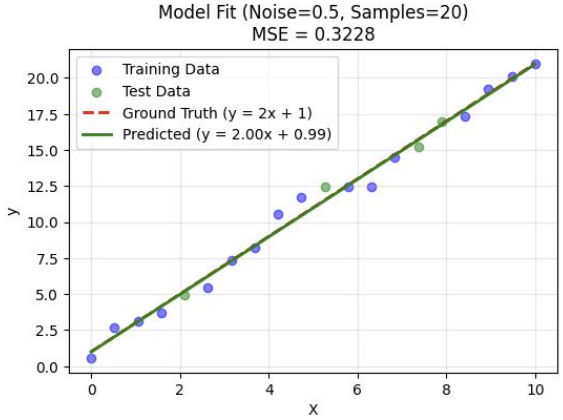
Learning the Optimal Prediction Function

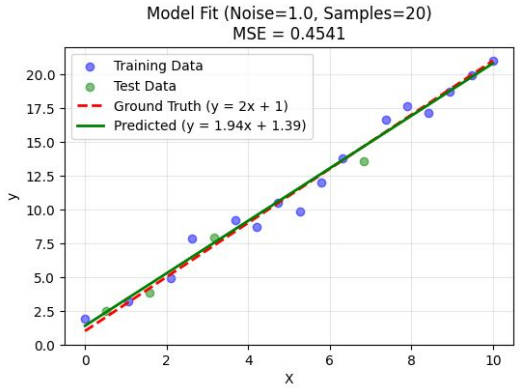
```
1  X, y, y_true = generate_dataset(n_samples=1000, noise_level=1.0)
2  model, train_loss, test_loss, (X_train, X_test, y_train, y_test) = train_and_evaluate(X, y, 0.2)
3  print("train loss: ", train_loss)
4  print("test loss: ", test_loss)
5  plot_model_fit(X, y, y_true, model, (X_train, X_test, y_train, y_test), noise_level=1.0, n_samples=1000)
```

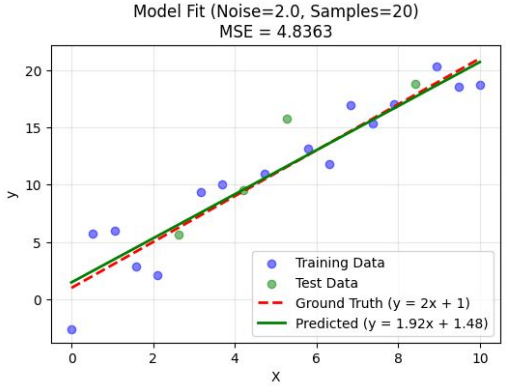
```
train loss:  1.00863532693082
test loss:  1.0290666563060962
```

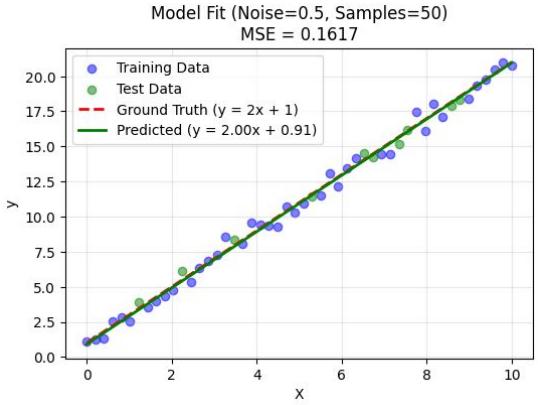


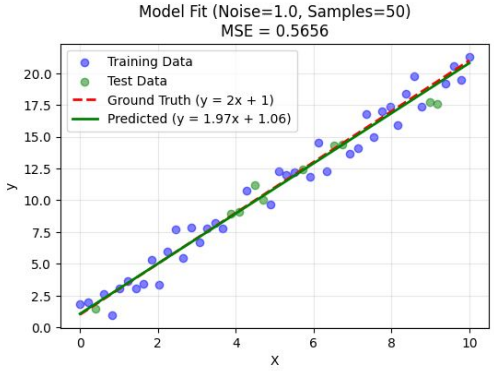
Learning the Optimal Prediction Function

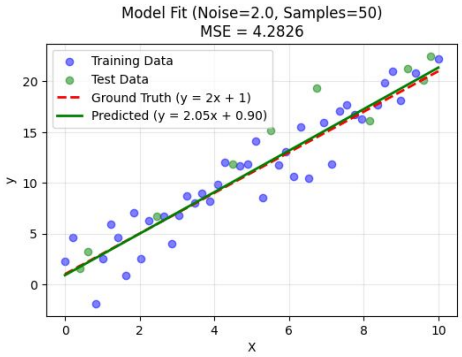










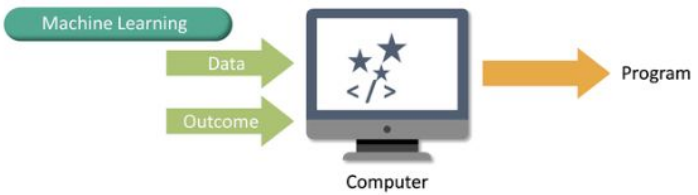


Dataset vs. Sample

number	student_id	gender	major01	major02	project	final	term	year
1	120060257	male	0.25	0	0	13	fall	2015
2	211110888	female	15.22	10.75	14.67	27	fall	2015
3	211210095	male	13.25	15.43	16.25	25	fall	2015
4	212110756	female	5.91	1.25	5	7	fall	2015
5	212110889	male		13	6.67	27	fall	2015
6	212110977	male	10.31	12.45		33	fall	2015
7	213110290	female	9.13	14.52	16.33	30	fall	2015
8	213110419	female	16	12.63	16.67	23	fall	2015
9	213110460	male		6.38	3		fall	2015
10	213110964	female	11.5	7.5	11.67	15	fall	2015
11	213111059	male	9.91	9.38	5.33	30	fall	2015
12	213111074	male	10.56	12	11.33	20	fall	2015
13	213111128	female	19	17.25	18	36	fall	2015
14	213210082	male	20	18.88	18.67	37	fall	2015
15	213210083	male	13.75	8	18	33	spring	2018
16	213210084	male	8.19	7.5	15.75	31	spring	2018
17	213210085	male	11		12	14	spring	2018
18	213210086	female	8.94	1	12	22	spring	2018
19	213210087	male	11.19	4.5	13	24	spring	2018
20	213210088	female	11	8.75	15	26	spring	2018
21	213210089	male	9.19		15	28	spring	2018
22	213210090	male	12.69	8	16	34	spring	2018
23	213210091	female	13.63	12	16	38	spring	2018
24	213210092	male	11.88	10.25	15.75	36	spring	2018
25	213210093	female		13	18	37	spring	2018
26	213210094	male	18.88	18.25	20	35	spring	2018
27	213210095	female	15.81	7.5	12	28	spring	2018
28	213210096	male	18.13	17	20	39	spring	2018
29	213210097	male	19.88	17.5	17.5	38	spring	2018
30	213210098	male	8.5	9.75		26	spring	2018
31	213210099	female	16.63	13	16	35	spring	2018
32	213210100	male	17.63	17.5	19	38	spring	2018
33	213210101	female	9.56		19	31	spring	2018
34	213210102	male	17.19	12.5	18.25	38	spring	2018
35	213210103	male	14.63	19	16	36	spring	2018
36	213210104	male	11.75	8.75	15	29	spring	2018
37	213210105	male	15.31		16	34	spring	2018

Learning the Optimal Prediction Function

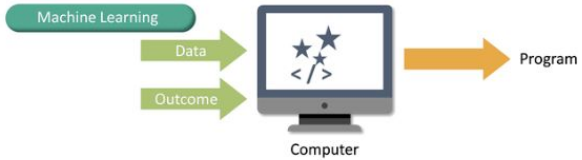
TEACHER-LEARNER PARADIGM



- **Educational Metaphor:**
  - Conceptualize g_T as a learner, trained via T to mimic $g^* : x \mapsto y$.
  - Training facilitated by a "teacher" providing n examples, guiding predictions on unseen data.
- **Supervised Learning Context:**
  - Engages in learning the function linking x (features) to y (response) under supervision.
  - Emphasizes on prediction, leveraging explanatory variables.

SUPERVISED LEARNING: CONCEPT AND APPLICATIONS

1. **Supervised Learning Defined:**
   - Learning the relationship between features (x) and response (y) with examples provided by a "teacher".
   - Focus on predicting y from x, using a set of explanatory variables.
2. **Learning Process:**
   - Objective: Learn unknown optimal function g^* from training data.
   - Approach: Use training set $T = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ to construct approximation g_T.
3. **Training Set Dynamics:**
   - Random set: T (training).
   - Deterministic outcome: τ (observed sample).
4. **Teacher-Learner Metaphor:**
   - g_T: Learner, trained to infer functional relationship $g^* : x \mapsto y$.
   - Teacher provides n examples, enabling g_T to predict on new inputs.

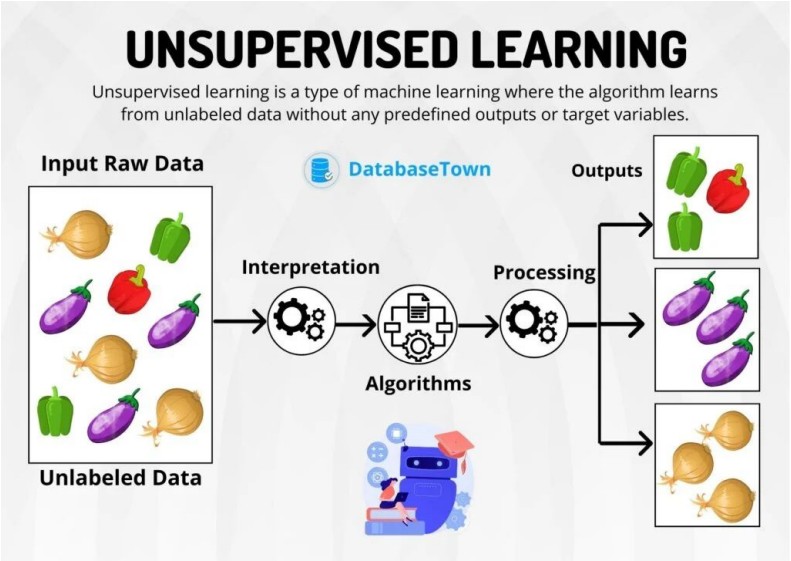


Significance of Conditional PDF:

- Understanding $f(y|x)$ allows for the determination of $g^*(x)$, facilitating prediction and analysis.

UNSUPERVISED LEARNING: EXPLORING DATA STRUCTURE

1. **Unsupervised Learning Overview:**
   - Focuses on learning the structure of an unknown distribution $f(x)$ without distinguishing between response and explanatory variables.
2. **Objective and Approach:**
   - Goal: Discover the intrinsic structure of data by approximating $f(x)$ with $g(x)$.
   - Risk Measurement: $l(g) = \mathbb{E}\,\text{Loss}(f(X), g(X))$.

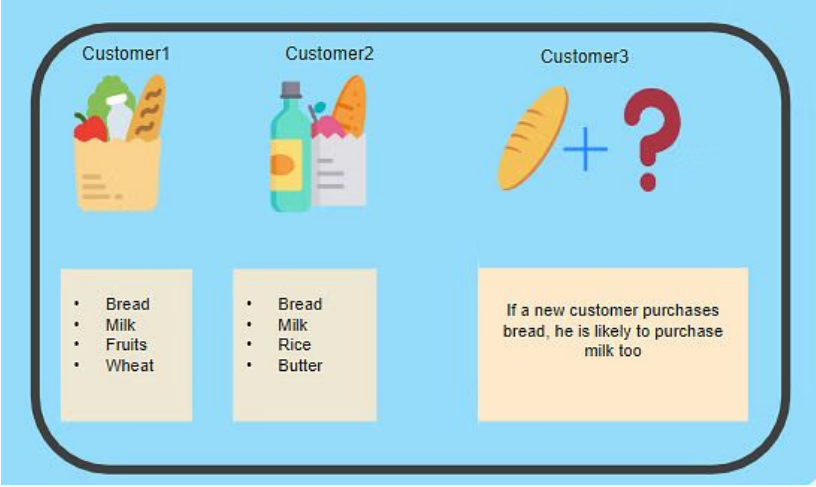


https://databasetown.com/unsupervised-learning-types-applications/

UNSUPERVISED LEARNING: EXPLORING DATA STRUCTURE

Example: Customer Purchase Patterns

- Analyze purchasing behaviors in a grocery shop with 100 items.
- Feature vector $x \in \{0,1\}^{100}$ represents items bought.
- Task: Identify patterns from binary vectors of purchases.



https://www.simplilearn.com/

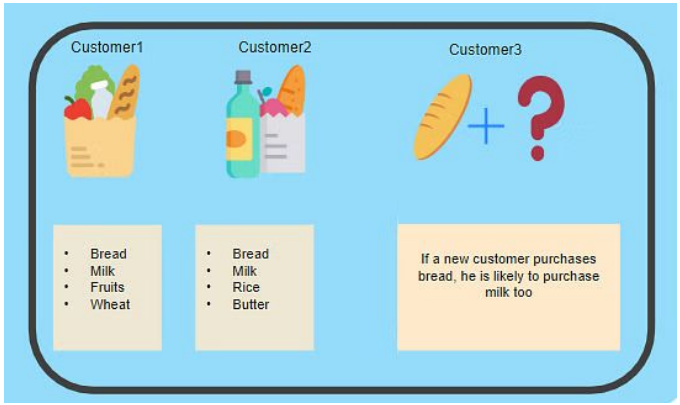
Data Representation:

- Feature vector $x \in \{0,1\}^{100}$: Indicates the presence (1) or absence (0) of 100 different grocery items in a customer's purchase.

UNSUPERVISED LEARNING: EXPLORING DATA STRUCTURE

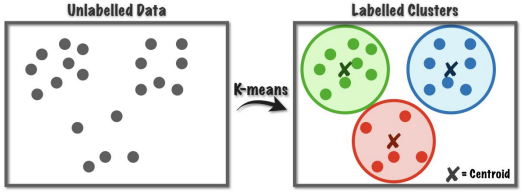
```python
binary_representation = [
    # Milk, Bread, Eggs
    [1, 1, 0],  # Transaction 1: Milk, Bread
    [0, 1, 1],  # Transaction 2: Bread, Eggs
    [1, 0, 1],  # Transaction 3: Milk, Eggs
    [1, 1, 1],  # Transaction 4: Milk, Bread, Eggs
    [0, 1, 0],  # Transaction 5: Bread
]

# Items for header
items_header = ['Milk', 'Bread', 'Eggs']

# Displaying as a simple text-based table
table_header = ' | '.join(items_header)
print(table_header)
print('-' * len(table_header))

for row in binary_representation:
    print(' | '.join(str(x) for x in row))
```

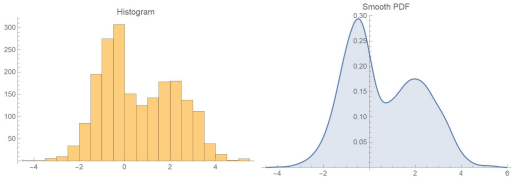
	Milk	Bread	Eggs	
0	1	1	0	Transaction 1
1	0	1	1	Transaction 2
2	1	0	1	Transaction 3
3	1	1	1	Transaction 4
4	0	1	0	Transaction 5



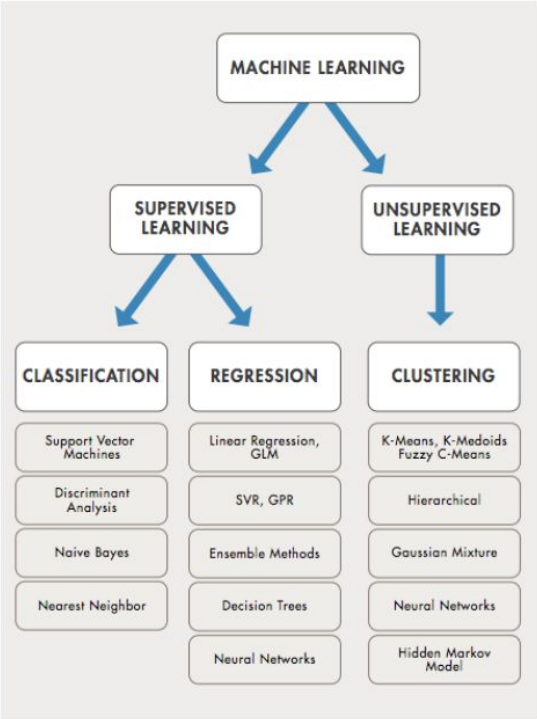
UNSUPERVISED LEARNING: EXPLORING DATA STRUCTURE

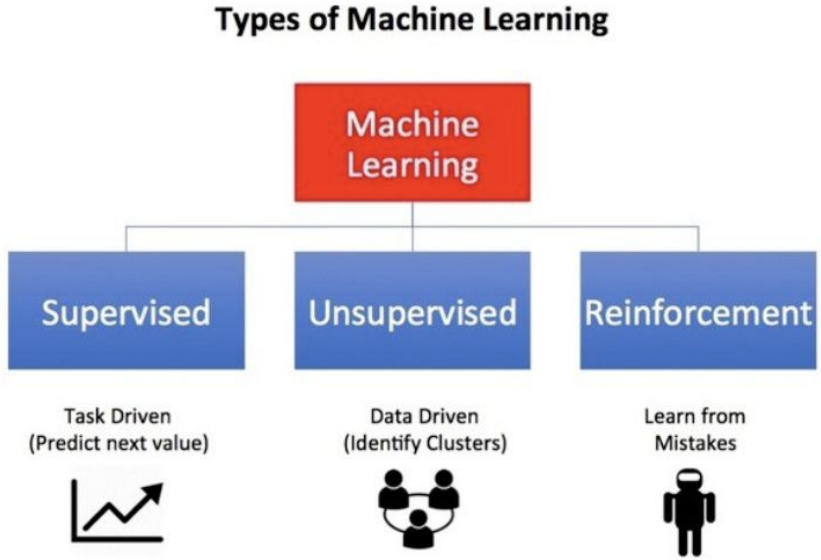
4. **Challenges in Unsupervised Learning:**
   - Lack of clear success metrics due to the absence of a "teacher" or labeled examples.
   - Difficulty in evaluating the performance of unsupervised learning algorithms.
5. **Key Methodologies:**
   - **Clustering:** Group similar data points.
   - **Principal Component Analysis (PCA):** Reduce dimensionality while preserving variance.
   - **Kernel Density Estimation:** Estimate the probability density function of a random variable.
6. **Significance:**
   - Enables discovery of hidden patterns and structures in data without predefined labels.
   - Essential for exploratory data analysis and understanding complex datasets.



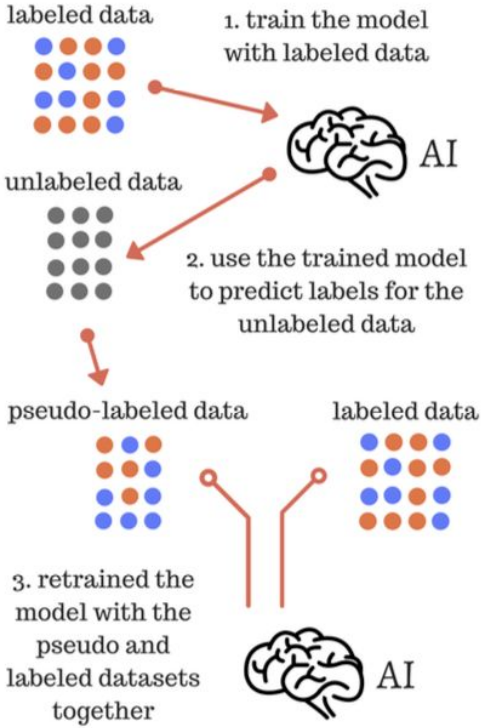


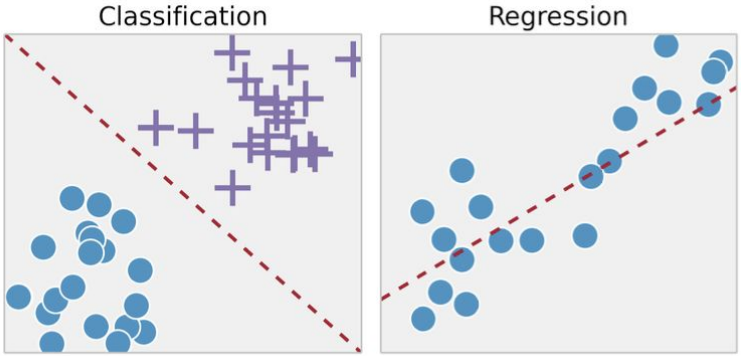
Supervised vs. Unsupervised Learning

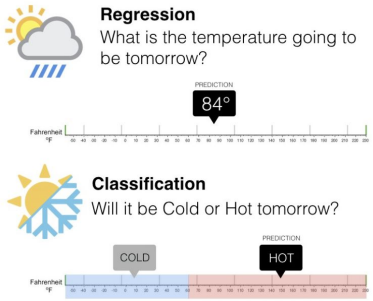




Supervised Learning





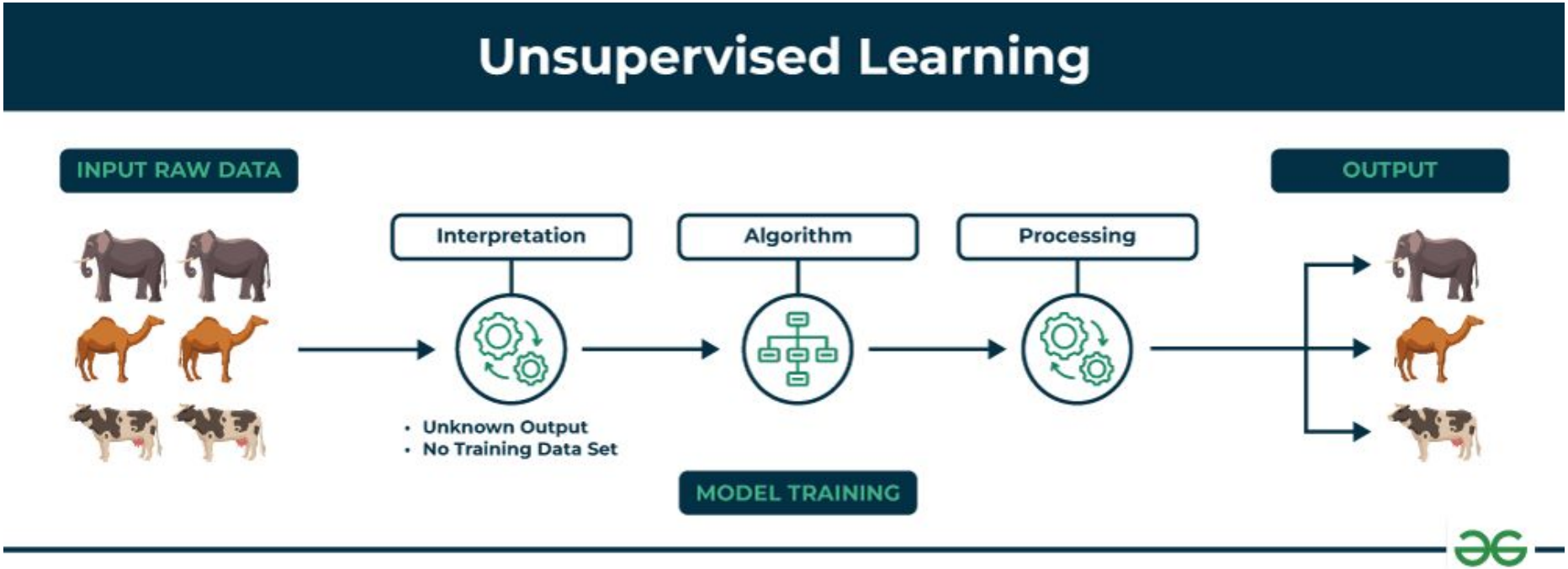


Regression and **classification** are categorized under the same umbrella of **supervised machine learning**.

The main difference the **output variable:**

- **regression** output is **numerical** (or continuous)
- **classification** output is **categorical** (or discrete)

Unsupervised Learning



CHAPTER 7
STATISTICAL LEARNING

LECTURE 2
FOUNDATIONS OF PREDICTIVE MODELING IN DATA SCIENCE

TRAIN vs. TEST LOSS

CS316: INTRODUCTION TO AI AND DATA SCIENCE

Prof. Anis Koubaa

Training Loss

Training Loss Definition:

$$l_T(g) = \frac{1}{n}\sum_{i=1}^{n} \text{Loss}(Y_i, g(X_i))$$

not possible to compute its real risk (Expected Loss) `R(g)`

$$R(g) = \mathbb{E}[L(Y, g(X))]$$

1. **Empirical Risk Approximation:**
   - **Training Loss ($l_T(g)$)**: Empirical risk calculated as the sample average loss:
     $l_T(g) = \frac{1}{n}\sum_{i=1}^{n} \text{Loss}(Y_i, g(X_i))$
   - Represents an unbiased estimate of the risk (expected loss) for g based on training data.

2. **Identifying the Optimal Prediction Function:**
   - **Selection Process**: Choose g_{GT} from a function set G that minimizes training loss:
     $g_{GT} = \arg\min_{g \in G} l_T(g)$
   - **Example of G**: The set of linear functions, where $g : x \mapsto \beta^\top x$ for a vector β.

CHAPTER 7
STATISTICAL LEARNING

LECTURE 2
FOUNDATIONS OF PREDICTIVE MODELING IN DATA SCIENCE

OVERFITTING

CS316: INTRODUCTION TO AI AND DATA SCIENCE

Prof. Anis Koubaa

Understanding Overfitting

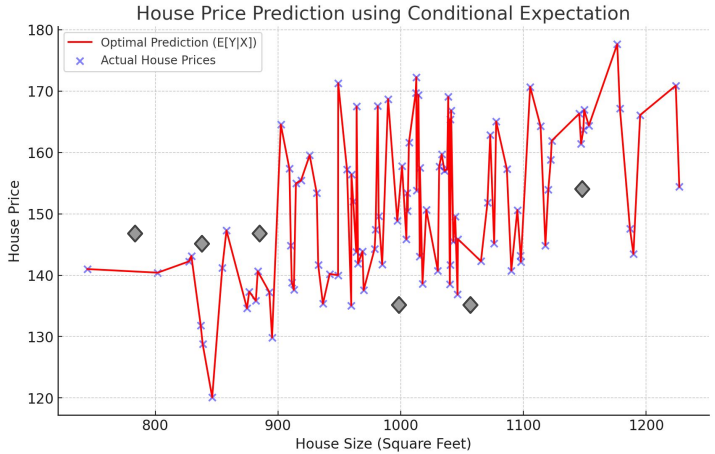
- **Overfitting Defined:**

  Overfitting occurs when a model learns the training data too closely, capturing noise and outliers, which results in poor performance on unseen data.

- **Training Loss Indicator:**

  $$\text{Training Loss} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

  Approaches zero in overfitting scenarios.



Overfitting Consequences

- **Generalization Failure:**

  Overfitting results in a substantial gap between training accuracy and the accuracy on novel, unseen data.

- **Quantifying Generalization Risk:**

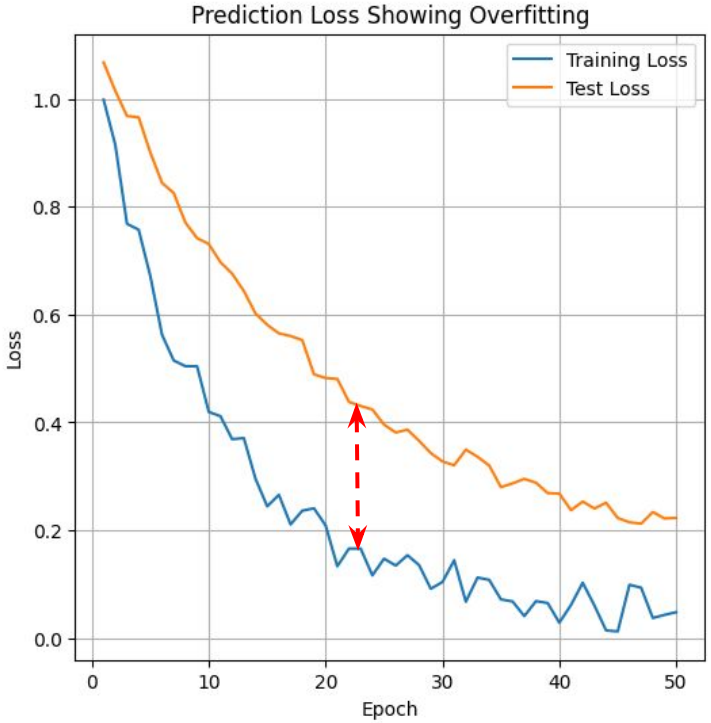
  $$\text{Generalization Risk} = \mathbb{E}[\text{Loss}(Y, g_{G_\mathcal{T}}(X))]$$

  $$l(g_{G_\mathcal{T}}) = \sum_{x,y} \text{Loss}(y, g_{G_\mathcal{T}}(x)) f(x, y)$$

  (In the continuous case, replace the sum with an integral.)

- **Impact of Overfitting on Generalization Risk:**

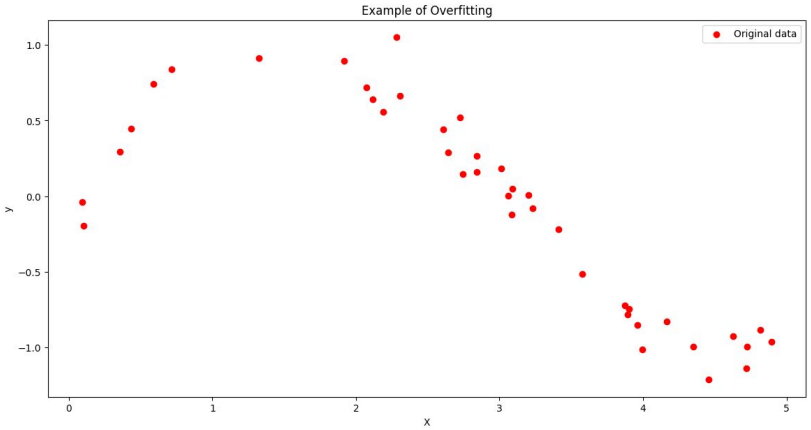
  Overfitting is marked by elevated generalization risk, signifying a model's poor generalization capability. This misalignment underscores the model's inability to accurately predict outcomes for unseen data, demonstrating the critical need for models that can generalize well beyond their training dataset.



Overfitting in Model Training

1. **Training Loss Minimization:**
   - Objective: Find g_{GT} within a function class G that minimizes training loss.
   - **Formula:**

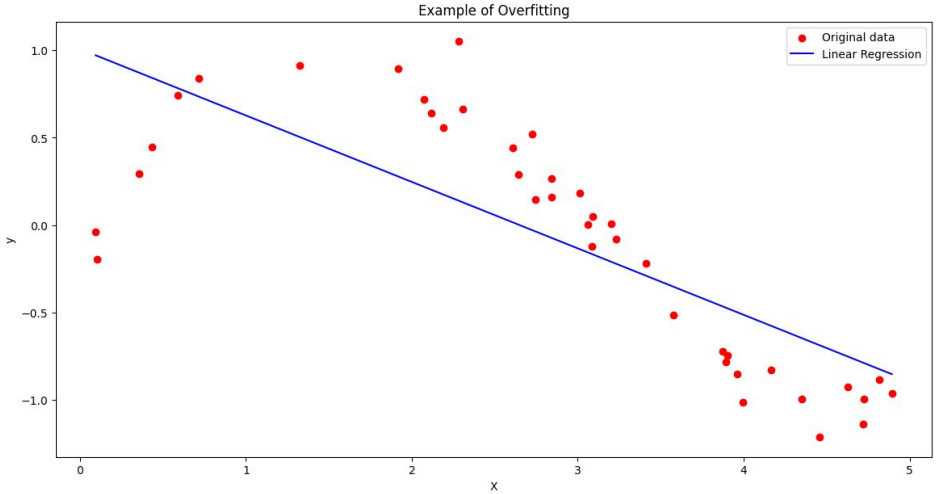
     $$g_{GT} = \arg\min_{g \in G} \frac{1}{n} \sum_{i=1}^{n} \text{Loss}(Y_i, g(X_i))$$
2. **Overfitting:**
   - Occurs when g_{GT} fits the training data too perfectly, leading to zero training loss.
   - **Implication:** Poor prediction on new, independent data due to lack of generalization.
3. **Example of Overfitting:**
   - A model where $g(X_i) = Y_i$ for all i in the training set, achieving zero squared-error training loss.



Overfitting in Model Training

1. **Training Loss Minimization:**
   - Objective: Find g_{GT} within a function class G that minimizes training loss.
   - **Formula:**

     $$g_{GT} = \arg\min_{g \in G} \frac{1}{n} \sum_{i=1}^{n} \text{Loss}(Y_i, g(X_i))$$
2. **Overfitting:**
   - Occurs when g_{GT} fits the training data too perfectly, leading to zero training loss.
   - **Implication:** Poor prediction on new, independent data due to lack of generalization.
3. **Example of Overfitting:**
   - A model where $g(X_i) = Y_i$ for all i in the training set, achieving zero squared-error training loss.

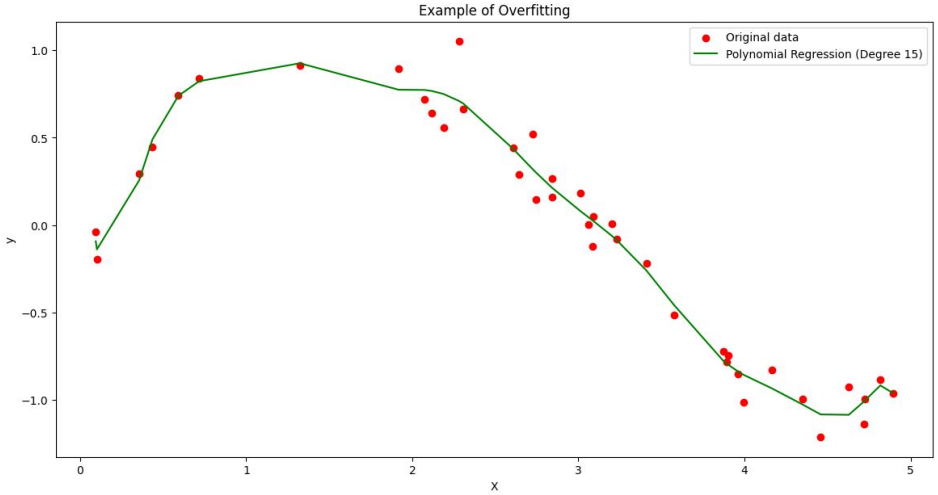


MSE = 0.1864

Overfitting in Model Training

1. **Training Loss Minimization:**
   - Objective: Find g_{GT} within a function class G that minimizes training loss.
   - **Formula:**

     $$g_{GT} = \arg\min_{g \in G} \frac{1}{n} \sum_{i=1}^{n} \text{Loss}(Y_i, g(X_i))$$
2. **Overfitting:**
   - Occurs when g_{GT} fits the training data too perfectly, leading to zero training loss.
   - **Implication:** Poor prediction on new, independent data due to lack of generalization.
3. **Example of Overfitting:**
   - A model where $g(X_i) = Y_i$ for all i in the training set, achieving zero squared-error training loss.

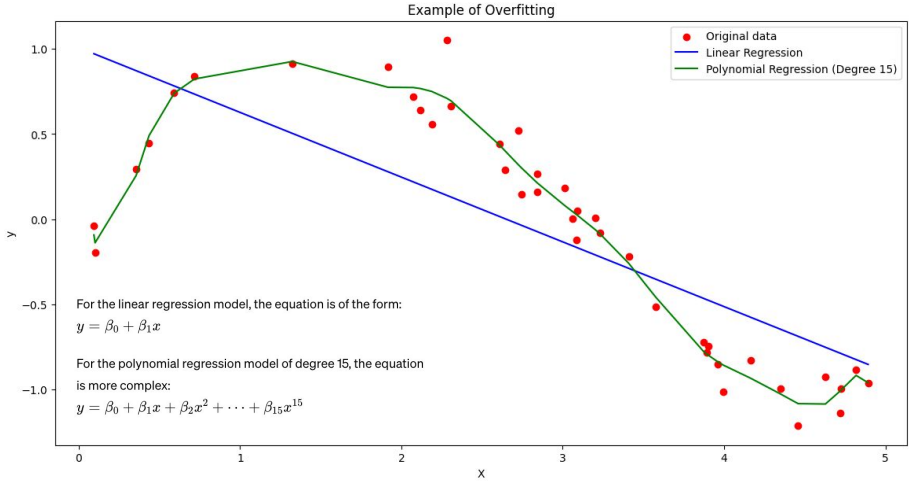


MSE = 0.010

Overfitting in Model Training

1. **Training Loss Minimization:**
   - Objective: Find g_{GT} within a function class G that minimizes training loss.
   - **Formula:**

     $$g_{GT} = \arg\min_{g \in G} \frac{1}{n} \sum_{i=1}^{n} \text{Loss}(Y_i, g(X_i))$$
2. **Overfitting:**
   - Occurs when g_{GT} fits the training data too perfectly, leading to zero training loss.
   - **Implication:** Poor prediction on new, independent data due to lack of generalization.
3. **Example of Overfitting:**
   - A model where $g(X_i) = Y_i$ for all i in the training set, achieving zero squared-error training loss.



Linear Regression Equation:

$$y = 1.01 - 0.38x$$

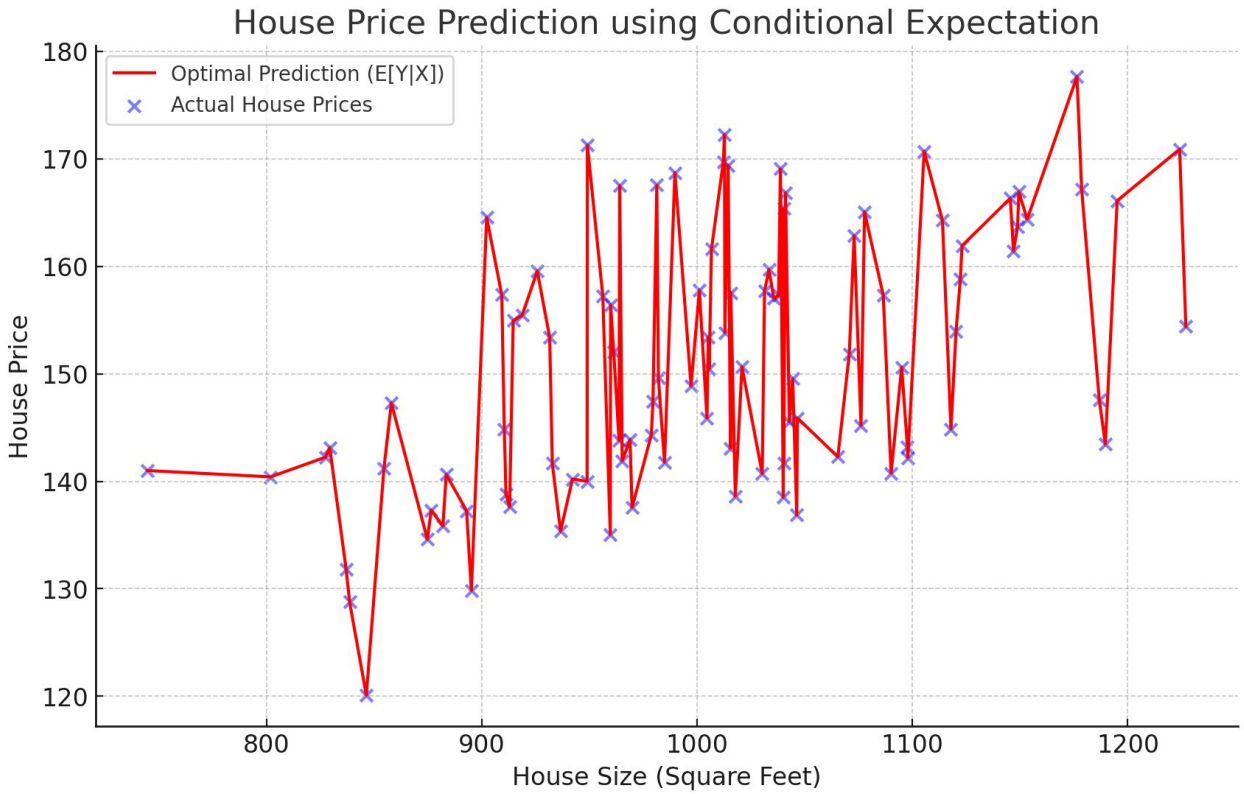
Polynomial Regression Equation (truncated):

$$y = 1.40e+00 - 2.75e+01x + 1.55e+02x^2 - 3.77e+02x^3 + 4.60e+02x^4 - 1.99e+02x^5 - \cdots$$

Mean Squared Error (MSE) Comparison:

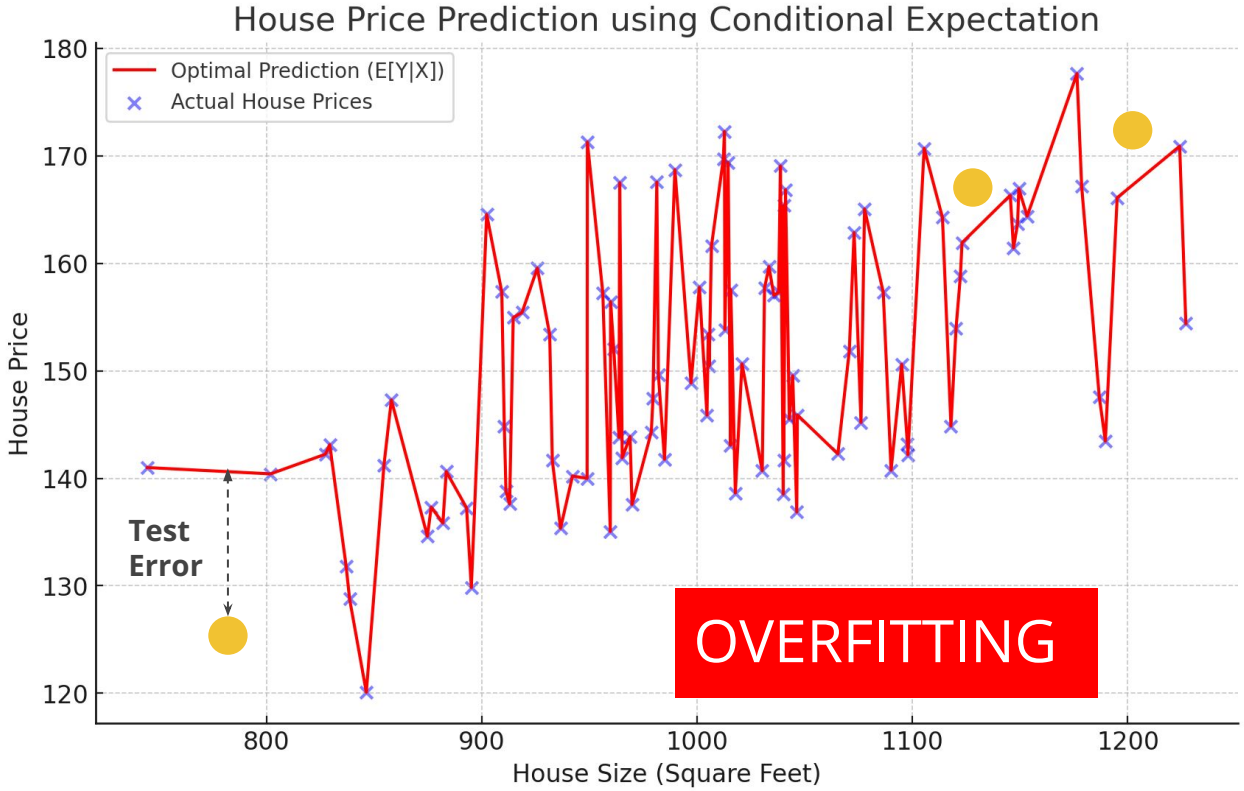
- **Linear Regression MSE:** 0.186
- **Polynomial Regression (Degree 15) MSE:** 0.011

What is the MSE Train Loss?



Train Loss = 0

What is the MSE Train/TEST Loss?



Train Loss = 0

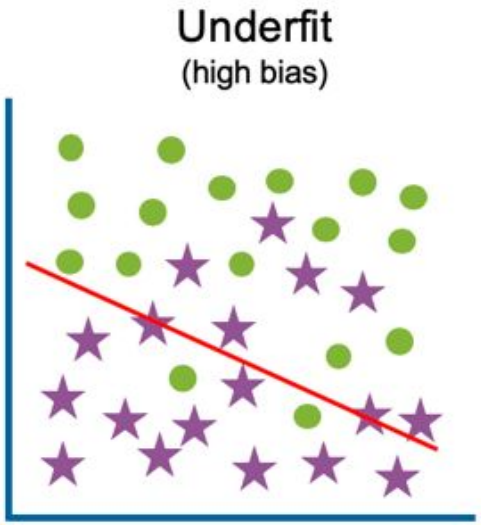
Unseen (Test) Data

Test Loss > 0

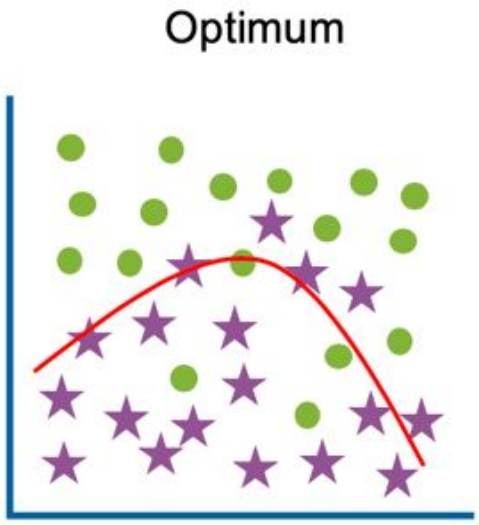
Training Goal

The ultimate goal is **not merely to minimize training loss** but to build models that **generalize well** to **unseen data**.

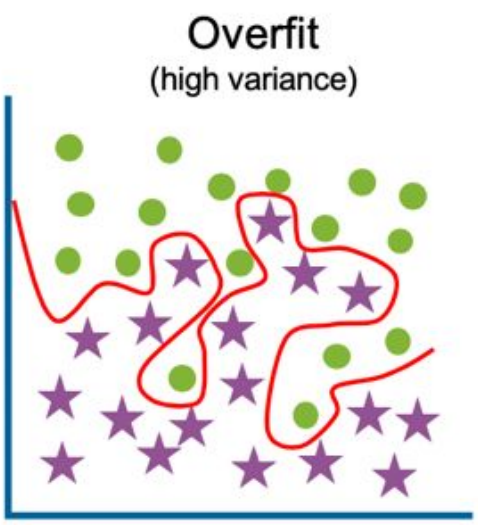
Overfitting | Optimal | Underfit



High training error
High test error



Low training error
Low test error



Low training error
High test error

CHAPTER 7
STATISTICAL LEARNING

LECTURE 2
FOUNDATIONS OF PREDICTIVE MODELING IN DATA SCIENCE

Generalization Risk

CS316: INTRODUCTION TO AI AND DATA SCIENCE

Prof. Anis Koubaa

Generalization Risk and Overfitting

- **Generalization Risk Formula:**

  $l_\tau(g_\tau^G) = \mathbb{E}[\text{Loss}(Y, g_\tau^G(X))]$

  - *Clarification:* g_τ^G denotes a function within the class G, specifically trained on dataset τ.
- **Understanding the Context:**
  - This formula encapsulates the expected accuracy of model g_τ^G on new, unseen data, where (X, Y) are sampled according to distribution $f(x, y)$.

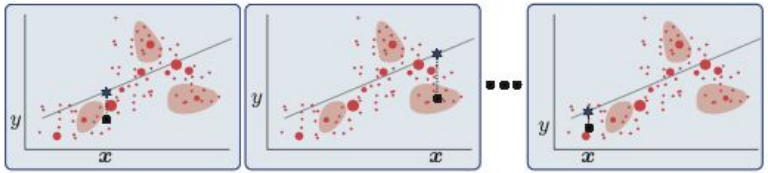


Figure 2.1: The generalization risk for a fixed training set is the weighted-average loss over all possible pairs (x, y).

FOR A FIXED TRAINING SET
In this scenario, generalization risk represents the **average expected loss** of a **prediction model (straight line)** on **new, unseen data points (black dots)**, taking into account the **underlying distribution of data (red dots).**

https://people.smp.uq.edu.au/DirkKroese/DSML/DSML.pdf

Expected Generalization Risk with Random Training Set T

- **Expected Generalization Risk for Random T:**

  $\mathbb{E}_T[l(g_T^G)] = \mathbb{E}[\text{Loss}(Y, g_T^G(X))]$

  - Averages generalization risk over all possible T.
  - g_T^G is the best predictor in G for training set T.
- **Expression in Discrete Case:**

  $\mathbb{E}_T[l(g_T^G)] = \sum_{x,y,x_1,y_1,\ldots,x_n,y_n} \text{Loss}(y, g_T^G(x)) f(x,y) \prod_{i=1}^n f(x_i, y_i)$

  - Considers all possible training sets and data points.
- **Key Insights:**
  - Generalization risk is a random variable due to T.
  - Expected generalization risk evaluates model's average performance across different training sets.
  - Aims for robustness in model prediction across varying training data instances.

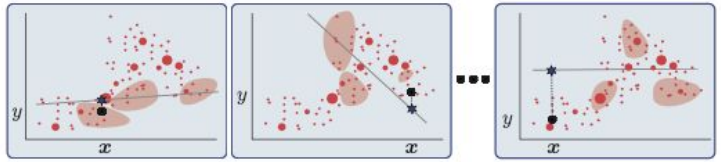
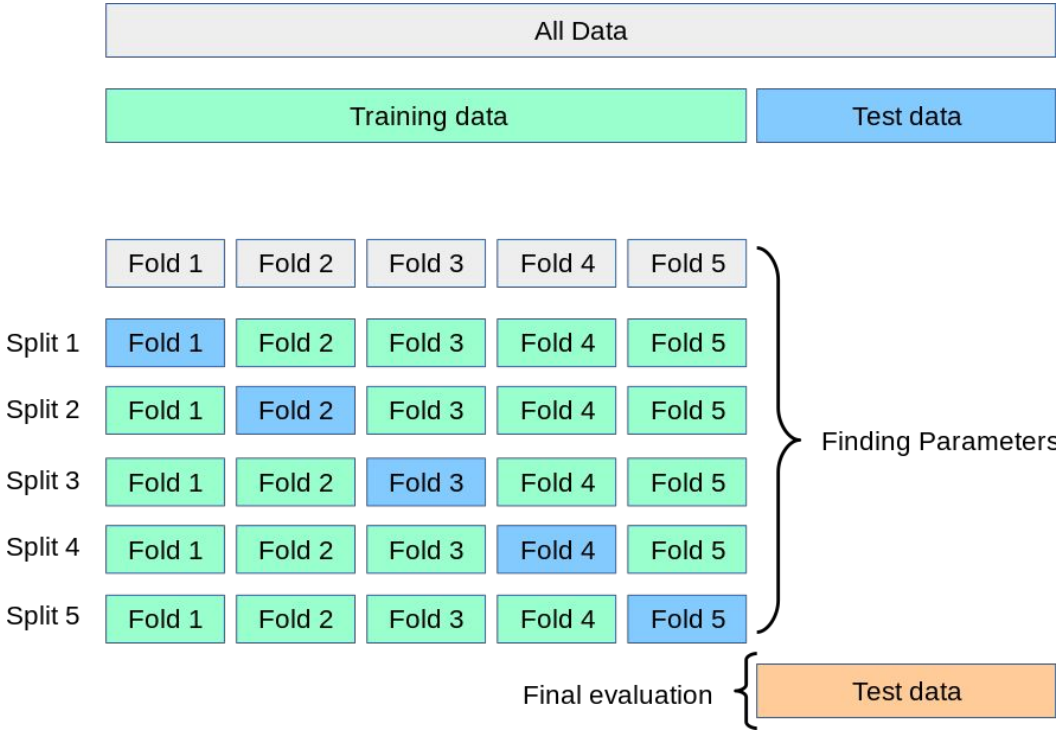


Figure 2.2: The expected generalization risk is the weighted-average loss over all possible pairs (x, y) and over all training sets.

FOR A FIXED TRAINING SET
In this scenario, the **Expected Generalization Risk** is the **average loss of a trained model (gGT)** over **all possible training sets (T)**, calculated using the loss function and the probability distributions of data and training instances.

https://people.smp.uq.edu.au/DirkKroese/DSML/DSML.pdf

Application: Cross Validation



https://scikit-learn.org/stable/modules/cross_validation.html

- τ - **Specific Training Dataset:**
  - Represents a specific instance of a training dataset.
  - Used to discuss outcomes related to this particular dataset.
  - Focuses on the concrete realization of data points.
- T - **Random Training Set:**
  - Denotes the concept of a random training set, highlighting variability in dataset selection.
  - Generalizes findings across different instances of training data.
  - Used in theoretical analyses for expected performance over all possible training sets.

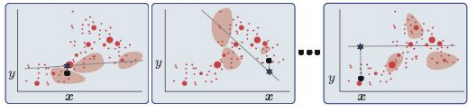


Figure 2.2: The expected generalization risk is the weighted-average loss over all possible pairs (x, y) and over all training sets.

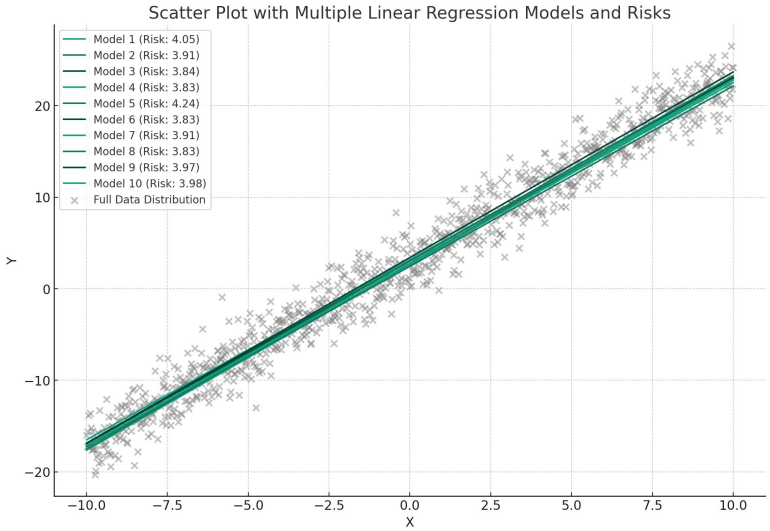
Example: Impact of Training Set Variability on Generalization Risk

- **Study Objective:**

  Examine the effect of training set selection on the generalization risk of linear regression models.

- **Methodology:**
  - Simulated 100 linear regression models using varied training sets.
  - True data model: $Y = 2X + 3$, with added noise.
  - Evaluated each model using Mean Squared Error (MSE) on a comprehensive dataset.
- **Results:**
  - **Average Generalization Risk:** Approximately 3.98.
  - **Risk Variability:** Standard deviation around 0.15.
- **Key Insight:**

  Variability in training sets significantly impacts generalization risk, highlighting the importance of evaluating models across a broad range of datasets to ensure robust performance assessment.

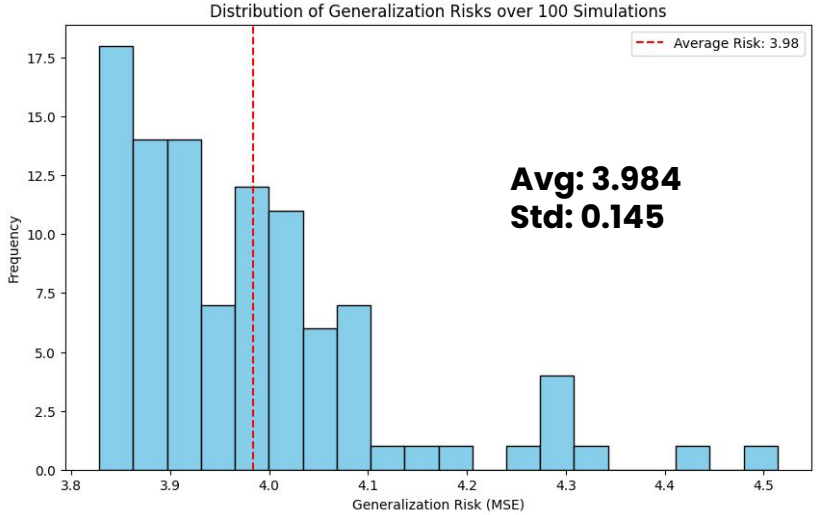


*The graph demonstrates the **variability in generalization risk** (measured by MSE) **across different training sets**, highlighting how **model performance can fluctuate** due to randomness in training data selection, with the average generalization risk providing a more reliable performance metric.*

https://scikit-learn.org/stable/modules/cross_validation.html

Example: Impact of Training Set Variability on Generalization Risk

- **Study Objective:**
  Examine the effect of training set selection on the generalization risk of linear regression models.
- **Methodology:**
  - Simulated 100 linear regression models using varied training sets.
  - True data model: $Y = 2X + 3$, with added noise.
  - Evaluated each model using Mean Squared Error (MSE) on a comprehensive dataset.
- **Results:**
  - **Average Generalization Risk:** Approximately 3.98.
  - **Risk Variability:** Standard deviation around 0.15.
- **Key Insight:**
  Variability in training sets significantly impacts generalization risk, highlighting the importance of evaluating models across a broad range of datasets to ensure robust performance assessment.



*The graph demonstrates the **variability in generalization risk** (measured by MSE) **across different training sets**, highlighting how **model performance can fluctuate** due to randomness in training data selection, with the average generalization risk providing a more reliable performance metric.*

https://scikit-learn.org/stable/modules/cross_validation.html

Example: Impact of Training Set Variability on Generalization Risk

```python
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import numpy as np

# Define the true relationship between X and Y
a, b = 2, 3  # Coefficients for the linear relationship
np.random.seed(42)  # For reproducibility

# Generate a large pool of data points that represents the entire distribution of X, Y pairs
X_full = np.linspace(-10, 10, 1000).reshape(-1, 1)
Y_full = a * X_full + b + np.random.normal(0, 2, X_full.shape)  # Adding some noise

# Function to simulate training and evaluate generalization risk
def simulate_generalization_risk(n_simulations=100, n_samples=50):
    risks = []  # To store the generalization risks

    for _ in range(n_simulations):
        # Randomly sample a training set
        indices = np.random.choice(range(len(X_full)), size=n_samples, replace=False)
        X_train = X_full[indices]
        Y_train = Y_full[indices]

        # Fit a linear regression model
        model = LinearRegression().fit(X_train, Y_train)

        # Predict on the full dataset to evaluate generalization risk
        Y_pred = model.predict(X_full)

        # Calculate MSE as the generalization risk for this simulation
        risk = mean_squared_error(Y_full, Y_pred)
        risks.append(risk)

    # Return the average generalization risk
    return np.mean(risks), np.std(risks)

# Simulate the generalization risk over 100 instances of the training set
avg_risk, std_risk = simulate_generalization_risk()

avg_risk, std_risk

```

```python
import matplotlib.pyplot as plt

# Run the simulation to get risks and their standard deviation
risks, _ = np.array([simulate_generalization_risk(1, 50) for _ in range(100)]).T

# Plotting the distribution of generalization risks
plt.figure(figsize=(10, 6))
plt.hist(risks, bins=20, color='skyblue', edgecolor='black')
plt.title('Distribution of Generalization Risks over 100 Simulations')
plt.xlabel('Generalization Risk (MSE)')
plt.ylabel('Frequency')
plt.axvline(x=avg_risk, color='r', linestyle='--', label=f'Average Risk: {avg_risk:.2f}')
plt.legend()
plt.show()
```