Principal Component Analysis (PCA)

Dimensionality Reduction Technique

1 Core Concept

Definition

PCA (Principal Component Analysis): A dimensionality reduction technique that transforms high-dimensional data into lower dimensions by finding new axes (principal components) that capture **maximum variance** in the data.

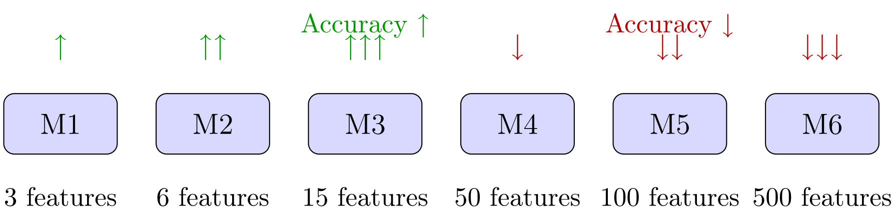
- Converts correlated features into uncorrelated principal components
- Each PC captures decreasing amounts of variance: $\text{Var}(PC_1) > \text{Var}(PC_2) > \text{Var}(PC_3) > \dots$
- Used for feature extraction (not selection)

2 Curse of Dimensionality

Definition

Curse of Dimensionality: As the number of features (dimensions) increases beyond an optimal point, model performance **degrades** instead of improving.

2.1 Why Does This Happen?



Important

Key Problems with Too Many Features:

1. **Overfitting:** Model learns noise along with signal
2. **Performance Degradation:** More dimensions = more computation = slower training
3. **Confusion:** Model tries to learn irrelevant features

2.2 Real-World Analogy

Location $\longrightarrow$ 450K-500K

3 BHK $\longrightarrow$ 500K-600K

Near Beach $\longrightarrow$ Price $\uparrow\uparrow$ Domain Expert

Near Celebrity $\longrightarrow$ Price $\uparrow\uparrow\uparrow$

Grocery Shops? $\dashrightarrow$ Confused!

Number of Schools? $\dashrightarrow$ Very Confused!

Note

Just like a domain expert gets confused with too many irrelevant features, ML models also get confused when fed unnecessary dimensions.

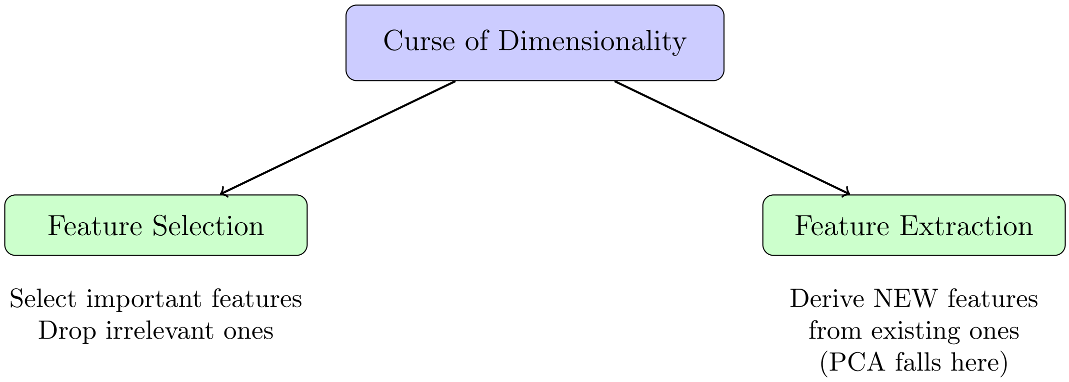
3 Why Use Dimensionality Reduction?

Remember

Three Key Reasons (Interview Question!)

1. **Prevent Curse of Dimensionality** — Avoid model overfitting and confusion
2. **Improve Model Performance** — Fewer dimensions = faster computation
3. **Visualize Data** — Humans can only see in 2D/3D, reduce to visualize high-dimensional data

4 Two Ways to Remove Curse of Dimensionality



Feature Selection	**Feature Extraction**
Select subset of original features	Create NEW derived features
Uses correlation/covariance	Uses transformation (PCA, etc.)
Original features retained	Original features transformed
Example: Drop "Fountain Size"	Example: Combine "Room Size" + "Num Rooms" → "House Size"

5 Feature Selection

Definition

Feature Selection: Process of selecting the most important features that have strong relationship with the output variable.

5.1 Covariance

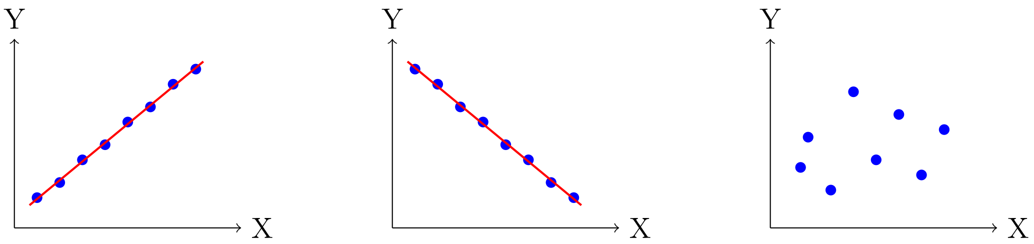
Important

Covariance Formula:

$$\boxed{\mathrm{Cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{n-1}}$$

Interpretation:

- $\mathrm{Cov}(X,Y) > 0$: Positive relationship (X $\uparrow \Rightarrow$ Y $\uparrow$)
- $\mathrm{Cov}(X,Y) < 0$: Negative relationship (X $\uparrow \Rightarrow$ Y $\downarrow$)
- $\mathrm{Cov}(X,Y) \approx 0$: No relationship



Cov > 0 Positive

Cov < 0 Negative

Cov ≈ 0 No Relation

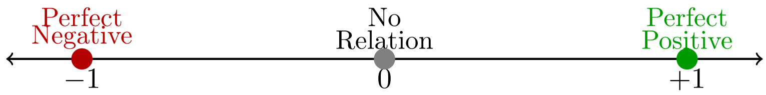
5.2 Pearson Correlation

Important

Pearson Correlation Coefficient:

$$\boxed{r = \frac{\text{Cov}(X,Y)}{\sigma_X \cdot \sigma_Y}}$$

Range: $-1 \leq r \leq +1$



Tip

Feature Selection Decision:

- If $|r| \approx 1$: Feature is **important** — KEEP
- If $|r| \approx 0$: Feature is **not useful** — DROP

5.3 Feature Selection Example

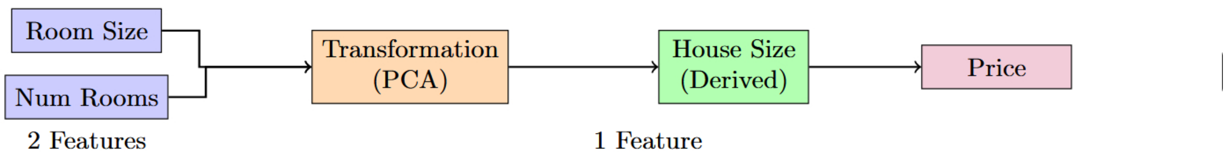
Strong correlation
KEEP

No correlation
DROP

6 Feature Extraction

Definition

Feature Extraction: Process of deriving NEW features from existing features through transformation, capturing the essence of original features in fewer dimensions.

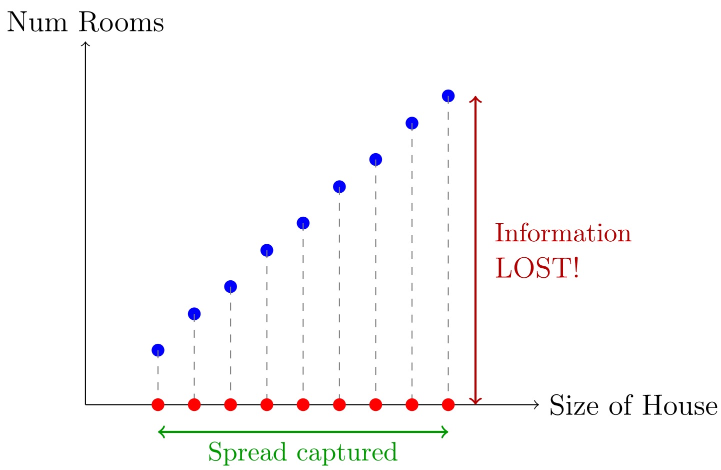


Warning

When to use Feature Extraction instead of Selection?
When ALL features are important and correlated with output, but you still need to reduce dimensions.
Example: Both "Room Size" and "Num Rooms" predict "Price" well — can't drop either!

7 Geometric Intuition Behind PCA

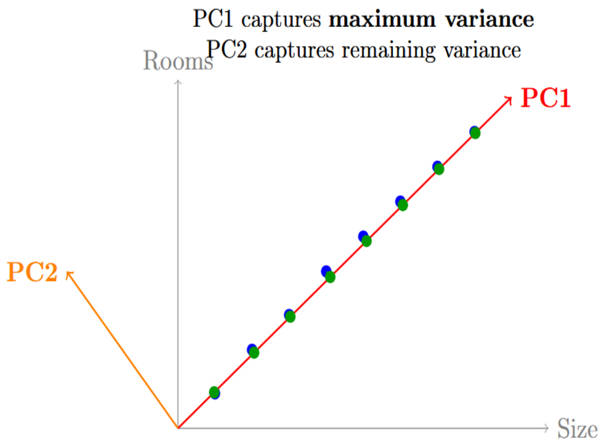
7.1 The Problem with Direct Projection



Warning

Problem: Direct projection to X-axis loses all Y-axis (Num Rooms) information!
The variance in Y-direction is completely ignored.

7.2 PCA Solution: Find Best Principal Component



Important

PCA Key Idea:

1. Apply **transformation** (Eigen decomposition) to find new axes
2. New axes = **Principal Components**
3. PC1 captures **maximum variance**
4. PC2 is perpendicular to PC1, captures next maximum variance
5. Project all points onto PC1 for dimensionality reduction

7.3 Variance and Spread Relationship

Remember

$$\text{Spread} \uparrow \implies \text{Variance} \uparrow \implies \text{Information} \uparrow$$

Goal: Find the line where projected points have MAXIMUM spread (variance)

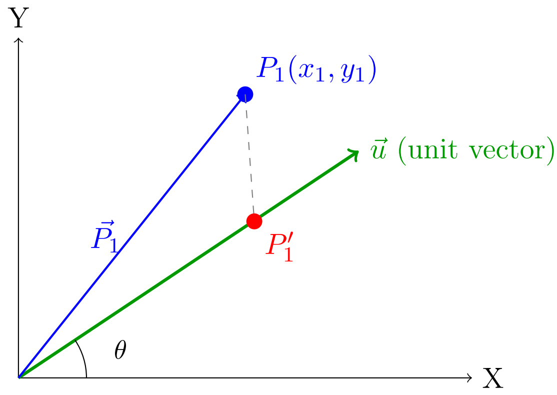
7.4 Principal Components Ordering

$$\text{Var}(PC_1) > \text{Var}(PC_2) > \text{Var}(PC_3) > \dots$$

PC1	$\longrightarrow$	PC2	$\longrightarrow$	PC3
Maximum Variance		2nd Maximum		3rd Maximum

8 Mathematical Intuition

8.1 Projection onto Unit Vector



Important

Projection Formula:

$$\boxed{\text{Projection of } \vec{P_1} \text{ on } \vec{u} = \frac{\vec{P_1} \cdot \vec{u}}{|\vec{u}|}}$$

Since $|\vec{u}| = 1$ (unit vector):

$$\boxed{P_1' = \vec{P_1} \cdot \vec{u}}$$

Result is a **scalar value** (distance from origin)

8.2 Cost Function: Maximize Variance

After projecting all points onto unit vector $\vec{u}$:

$$\text{Projected points: } X_0', X_1', X_2', \ldots, X_n'$$

Important

Variance of Projected Points:

$$\boxed{\text{Variance} = \frac{\sum_{i=1}^{n}(X_i' - \bar{X}')^2}{n}}$$

Goal: Find unit vector $\vec{u}$ that **MAXIMIZES** this variance

$$\boxed{\max_{\vec{u}} \text{Var}(X')}$$

9 Eigenvectors and Eigenvalues

Definition

Eigen Equation:

$$\boxed{A \cdot \vec{v} = \lambda \cdot \vec{v}}$$

Where:

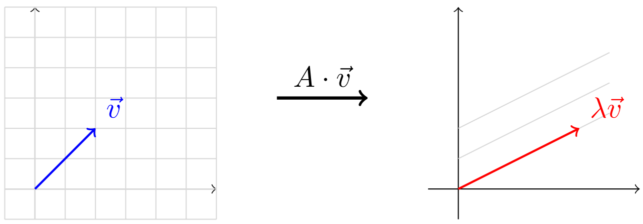
- A = Covariance matrix
- $\vec{v}$ = Eigenvector (direction of principal component)
- λ = Eigenvalue (magnitude/importance of that direction)

Remember

Key Insight:

- Eigenvector with **LARGEST eigenvalue** = PC1 (captures max variance)
- Eigenvector with 2nd largest eigenvalue = PC2
- And so on...

9.1 Linear Transformation Visualization



Before After

Note

When matrix A is applied to eigenvector $\vec{v}$, the vector only **scales** (by λ), doesn't change direction. This is what makes eigenvectors special!

10 Complete PCA Algorithm Steps

Step 1: Standardize the data (zero-centered)

$\downarrow$

Step 2: Compute Covariance Matrix

$\downarrow$

Step 3: Find Eigenvectors and Eigenvalues

$\downarrow$

Step 4: Sort by Eigenvalue (descending)

$\downarrow$

Step 5: Select top k Principal Components

$\downarrow$

Step 6: Project data onto selected PCs

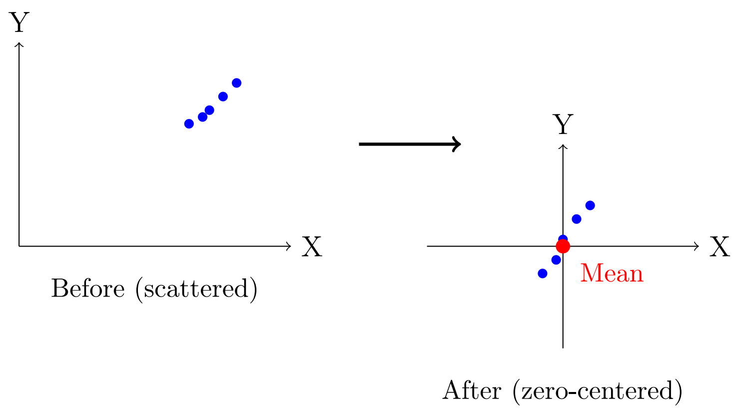
10.1 Step 1: Standardize Data

Important

Standardization Formula:

$$X_{std} = \frac{X - \mu}{\sigma}$$

This makes data **zero-centered** (mean = 0)



10.2 Step 2: Covariance Matrix

Important

For 2 Features (X, Y):

$$\text{Cov Matrix} = \begin{bmatrix} \text{Var}(X) & \text{Cov}(X, Y) \\ \text{Cov}(Y, X) & \text{Var}(Y) \end{bmatrix}$$

For 3 Features (X, Y, Z):

$$\text{Cov Matrix} = \begin{bmatrix} \text{Var}(X) & \text{Cov}(X, Y) & \text{Cov}(X, Z) \\ \text{Cov}(Y, X) & \text{Var}(Y) & \text{Cov}(Y, Z) \\ \text{Cov}(Z, X) & \text{Cov}(Z, Y) & \text{Var}(Z) \end{bmatrix}$$

Note

$\text{Cov}(X, X) = \text{Var}(X)$ (diagonal elements)
$\text{Cov}(X, Y) = \text{Cov}(Y, X)$ (symmetric matrix)

10.3 Step 3: Find Eigenvectors & Eigenvalues

Apply eigen decomposition to covariance matrix A:

$$A \cdot \vec{v} = \lambda \cdot \vec{v}$$

For 2D data: Get λ_1, λ_2 and corresponding $\vec{v}_1, \vec{v}_2$

For 3D data: Get $\lambda_1, \lambda_2, \lambda_3$ and corresponding vectors

10.4 Step 4 & 5: Sort and Select

Eigenvalue	Eigenvector	Principal Component
λ_1 (largest)	$\vec{v}_1$	PC1
λ_2	$\vec{v}_2$	PC2
λ_3 (smallest)	$\vec{v}_3$	PC3

Tip

Dimension Reduction:

- 3D $\rightarrow$ 1D: Use only PC1
- 3D $\rightarrow$ 2D: Use PC1 + PC2
- 2D $\rightarrow$ 1D: Use only PC1

10.5 Step 6: Project Data

Project all data points onto selected principal components to get reduced-dimension data.

11 Dimension Reduction Examples

Original Dim	**Target Dim**	**Use These PCs**
2D	1D	PC1 only
3D	1D	PC1 only
3D	2D	PC1 + PC2
100D	3D	PC1 + PC2 + PC3

12 Key Formulas Summary

Remember

1. **Covariance:**

$$\mathrm{Cov}(X, Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{n-1}$$

2. **Correlation:**

$$r = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \cdot \sigma_Y}, \quad -1 \leq r \leq 1$$

3. **Variance:**

$$\mathrm{Var}(X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n}$$

4. **Projection:**

$$\mathrm{Proj}_{\vec{u}}(\vec{P}) = \vec{P} \cdot \vec{u} \quad \text{(for unit vector } \vec{u}\text{)}$$

5. **Eigen Equation:**

$$A\vec{v} = \lambda\vec{v}$$

13 Common Mistakes

Warning

1. Forgetting to Standardize:

- PCA is sensitive to scale
- Always standardize before applying PCA

2. Confusing Feature Selection vs Extraction:

- Selection: Choose existing features
- Extraction (PCA): Create NEW features

3. Wrong PC Selection:

- PC1 has HIGHEST eigenvalue (not lowest)
- More eigenvalue = More variance captured

4. Assuming PCA Always Helps:

- PCA assumes linear relationships
- May lose important information in non-linear data

14 Quick Revision Points

1. **Curse of Dimensionality:** Too many features → overfitting, slow performance
2. **Two solutions:** Feature Selection (drop) vs Feature Extraction (transform)
3. **PCA = Feature Extraction** — creates NEW features (Principal Components)
4. **Covariance** measures relationship between features (> 0: positive, < 0: negative, ≈ 0: none)
5. **Correlation** is normalized covariance (range: -1 to $+1$)
6. **PCA Goal:** Find direction (PC) that captures MAXIMUM variance
7. **Eigenvector** = direction of PC, **Eigenvalue** = magnitude (importance)
8. **Largest eigenvalue** → PC1 (most important)
9. **PCA Steps:** Standardize → Covariance Matrix → Eigen decomposition → Sort → Select top k → Project
10. **Variance** $\propto$ **Spread** $\propto$ **Information**

15 Visual Summary: PCA Flow