Chapter 3: Numeric data (9 hrs)

1. Multivariate Linear Regression
   a) Matrix formulation and OLS estimation
   b) Measures of fit: R-squared and Adjusted R-squared
   c) Multi-collinearity and variance-inflation factors
2. Non-parametric Regression: Nadaraya-Watson Kernel regression
   a) Derivation of the estimator
   b) Rules of setting the appropriate bandwidth size
3. Principal Component Analysis
   a) Mathematical formulation and relevant derivations
   b) Interpreting the principal components

1. Numeric data

Numeric data consists of values that represent **quantities**. These values can be used for performing mathematical operations such as addition, subtraction, averaging, correlation, regression, etc.

Examples: height (170 cm), weight (55 kg), marks (82), income (Rs. 50,000), temperature (24.5°C).

Types of Numeric Data

There are two major types:

i) Discrete Numeric Data

- Takes **countable** values.
- No fractions allowed.
- Examples:
  - Number of students (25)
  - Number of cars in parking (48)

ii) Continuous Numeric Data

- Takes **measurable** values and can have decimals.
- Infinite possible values within a range.
- Examples:
  - Height (162.5 cm)
  - Time (3.74 seconds)
  - Temperature (28.3°C)

Multivariate Data

Multivariate data is data involving multiple variables (two or more) recorded for each observation or individual.

Examples:

- A student's record: *(height, weight, marks, attendance)*
- A company's sales data: *(sales, profit, number of customers)*
- A country's development indicators: *(GDP, literacy rate, life expectancy, inflation)*

Each observation has multiple measurements**.**

Types of Multivariate Data

1. **Numeric multivariate data:** e.g., height, weight, age, income
2. **Categorical multivariate data:** e.g., gender, color, nationality
3. **Mixed-type multivariate data**: e.g., income (numeric) + education (categorical)

Examples from Real Life

Example 1: Health dataset

Person	Height	Weight	BP	Cholesterol
A	165	60	120	190
B	172	72	135	210

Each person has multiple features → multivariate.

Example 2: Education dataset

Student	Maths	Science	English
S1	80	75	70
S2	85	78	75

Structure of Multivariate Data

Usually presented as a **data matrix**:

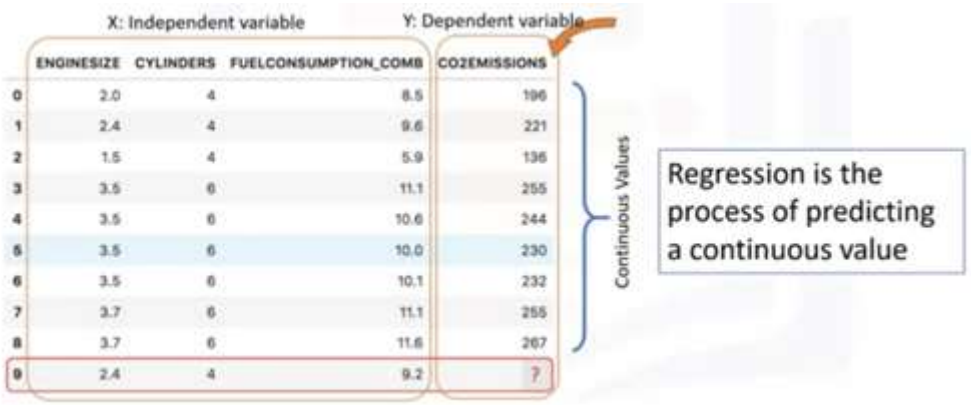
$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}$$

Where:

- **n** = number of observations
- **p** = number of variables
- Each row = one observation
- Each column = one variable

2. Introduction to Linear Regression

Linear regression is a powerful tool used to predict a continuous dependent variable based on one or more independent variables. It establishes a linear relationship between the dependent variable (target) and one or more independent variables (predictors). In our case, we are interested in predicting CO2 emissions from a car based on its engine size, among other potential factors.



The general equation of linear regression is:

$$Y = \beta_0 + \beta_1 \cdot X_1$$

Where:

- Y is the dependent variable (CO_2 emissions),
- X_1 is the independent variable (engine size),
- β_0 is the intercept of the line (where the line crosses the Y-axis),
- β_1 is the slope of the line, indicating how much Y (CO_2 emissions) changes for each unit change in X_1 (engine size).

3. Multivariate Linear Regression

It is the extension of simple linear regression.

Multivariate linear regression is a statistical technique that models the relationship between multiple independent variables and multiple dependent variables.

Multivariate linear regression is designed for situations where you need to predict more than one outcome at the same time.

It helps answer: *How does Y change when several predictors change at the same time?*

Examples:

- Predicting salary based on **experience, education, age**
- Predicting house price using **size, number of rooms, location score**
- Predicting marks using **study hours, attendance, IQ score**

Why Multivariate Regression? (Uses)

1. **Prediction**

   E.g., predicting house price using size, location, age.
2. **Modeling relationships**

   Understand how hours studied, IQ, and sleep affect exam score.
3. **Feature importance**

   Identify which variables have strongest influence.
4. **Control for multiple factors**

   Example: UNICEF MICS data — predicting child nutrition controlling for income, mother's education, sanitation.
5. **Forecasting**

   Useful for time-series + cross-sectional data.

Model Representation

The multivariate linear regression model with *p* predictors is:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

Where:

- Y = dependent variable
- $X_1, X_2, \ldots, X_p$ = predictors
- β_0 = intercept
- $\beta_1, \ldots, \beta_p$ = regression coefficients
- ε = error term

a) Matrix formulation and OLS estimation

To simplify computations, the model is written in matrix form:

$$Y=X\beta+\varepsilon$$

Where:

- *Y* is an *n*×*p* matrix of dependent variables (*n* observations, *p* response variables)
- *X* is an *n*×*(k+1)* matrix of independent variables (including intercept)
- *B* is a *(k+1)*×*p* matrix of regression coefficients
- *E* is an *n*×*p* matrix of error terms

Y (n × 1 vector):

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

X (n × p matrix):

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}$$

β (p × 1 vector):

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$$

ε (n × 1 error vector)

$$\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Example:

Given dataset, represent this into matrix form to model: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$

Obs	X1	X2	Y
1	2	1	5
2	4	2	9
3	3	5	12

Solution:

Step 1: Form the X matrix

Always include:

- A column of **1s** for the intercept
- Columns for each predictor

$$X = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 4 & 2 \\ 1 & 3 & 5 \end{bmatrix}$$

Step 2 — Form the Y vector

$$Y = \begin{bmatrix} 5 \\ 9 \\ 12 \end{bmatrix}$$

Step 3 — Parameter vector

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

Final Matrix Regression Model

$$\begin{bmatrix} 5 \\ 9 \\ 12 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 4 & 2 \\ 1 & 3 & 5 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}$$

Ordinary Least Squares (OLS) estimation:

- **OLS** is the most commonly used method for estimating the parameters of a linear regression model.s
- It finds the **best-fitting regression line** by **minimizing the sum of squared errors** between the observed values and the predicted values.
- In multivariate regression with several predictors:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon$$

- OLS uses **matrix algebra** to estimate all coefficients at once.

OLS estimator is:

$$\hat{\beta} = (X'X)^{-1}X'Y$$

Where:

- X: matrix of predictors
- Y: vector of outputs
- $(X'X)^{-1}$: inverse of $X'X$
- $\hat{\beta}$: estimated regression coefficients

Example:

Obs	X1	X2	Y
1	2	1	5
2	4	2	9
3	3	5	12

1. Form the **design matrix** X (include intercept column of 1s) and response vector Y

$$X = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 4 & 2 \\ 1 & 3 & 5 \end{bmatrix} \qquad Y = \begin{bmatrix} 5 \\ 9 \\ 12 \end{bmatrix}$$

2. **Compute X′X and X′Y**

$$X'X = \begin{bmatrix} 3 & 9 & 8 \\ 9 & 29 & 25 \\ 8 & 25 & 30 \end{bmatrix} \qquad X'Y = \begin{bmatrix} 26 \\ 82 \\ 83 \end{bmatrix}$$

3. OLS formula

$$\hat{\beta} = (X'X)^{-1}X'Y.$$

So we need: $(X'X)^{-1}$.

$$(X'X)^{-1} = \begin{bmatrix} 5 & -\frac{10}{7} & -\frac{1}{7} \\ -\frac{10}{7} & \frac{26}{49} & -\frac{3}{49} \\ -\frac{1}{7} & -\frac{3}{49} & \frac{6}{49} \end{bmatrix}$$

Now,

$$\hat{\beta} = (X'X)^{-1}X'Y = \begin{bmatrix} 1 \\ \frac{9}{7} \\ \frac{10}{7} \end{bmatrix} \approx \begin{bmatrix} 1.0000 \\ 1.2857 \\ 1.4286 \end{bmatrix}$$

Final estimated regression equation

$$\hat{Y} = 1 + 1.2857\,X_1 + 1.4286\,X_2$$

Assignment:

A real estate analyst wants to predict house prices (in $1000s) based on:

- **X_1 = Size of the house in hundreds of sq. ft.**
- **X_2 = Number of bedrooms**

The dataset collected is:

House	Size (X_1)	Bedrooms (X_2)	Price (Y)
1	2	1	150
2	3	2	200
3	4	3	250
4	5	3	300
5	6	4	350

Task: Find the estimated regression equation, using matrix formulation and OLS estimation.

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

b) Measures of fit: R-squared and Adjusted R-squared

In regression analysis, after estimating the model, we want to know:

How well does the model explain the variation in the dependent variable (y)?

Two common measures are:

- R-squared (Coefficient of Determination)
- Adjusted R-squared

i) R-squared (Coefficient of Determination)

R-squared (R^2) or the coefficient of determination, is the statistical measure of the variance of the regression (best fit) line from the actual data points. In a generalized linear regression model the model accuracy ranging from 75 - 95% is considered as a good prediction model, whereas 100% accuracy means overfitting.

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

Where:

- *Numerator is **sum of squares of the residuals** (the difference between the observed and predicted values).*
- *and Denominator is the **total sum of squares** (the variance of the observed data).*

$$R^2 = 1 - \frac{\text{SSR}}{\text{SST}} = \frac{\text{SSE}}{\text{SST}}$$

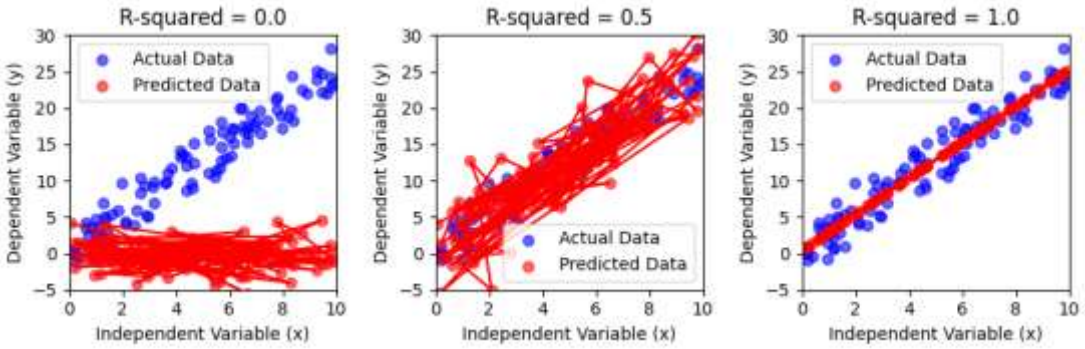
Where:

$$\text{SSR} = \sum(y_i - \hat{y}_i)^2 \quad \text{(Sum of Squared Residuals)}$$
$$\text{SST} = \sum(y_i - \bar{y})^2 \quad \text{(Total Sum of Squares)}$$
$$\text{SSE} = \sum(\hat{y}_i - \bar{y})^2 \quad \text{(Explained Sum of Squares)}$$

R-squared values range from **0 to 1**:

R-squared quantifies how well a model fits the data. Higher values indicate a better fit, while lower values suggest the model is less effective.

- An **R-squared of 0** indicates that the model explains none of the variability of the dependent variable.
- An **R-squared of 1** indicates that the model explains all the variability of the dependent variable.



- **R-squared = 0.0**: The predicted points are randomly scattered, indicating the model has no predictive power. The independent variable (x) does not explain any variation in the dependent variable (y).
- **R-squared = 0.5**: The predicted points show a moderate relationship with the actual data. The model explains some variation in y, but a lot remains unexplained.
- **R-squared = 1.0**: The predicted points perfectly align with the actual data, indicating a perfect fit. The independent variable explains all the variation in y.

ii) Adjusted R-squared

- Adjusted R-squared is a performance metrics which can be termed as a more refined version of R-squared which priorities the **input features that correlates with the target variable.**
- It takes into account the number of predictors in the model and whether they are significant.
- While **R-squared** always increases when more predictors are added, **Adjusted R-squared** increases **only if the new predictors genuinely improve the model**.
- It prevents overfitting by balancing the model's performance with its complexity. Formula for Adjusted R-squared is:

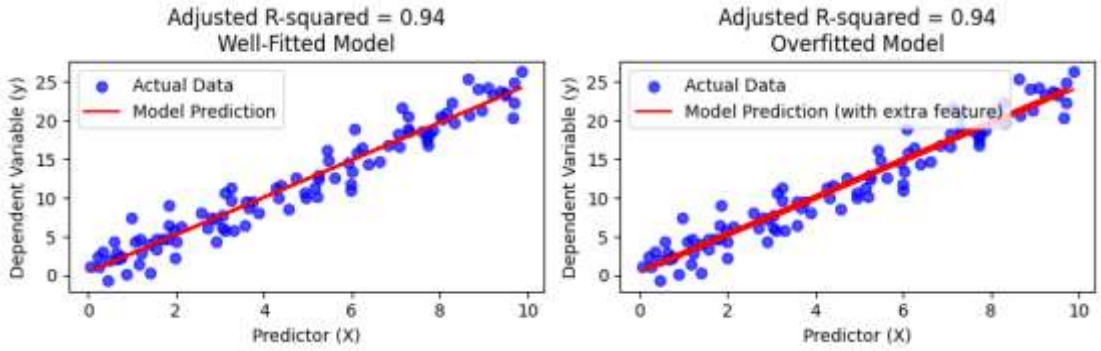
$$\overline{R}^2 = 1 - \frac{(1-R^2)(n-1)}{n-k-1}$$

Where:

- *n = number of data points (observations)*
- *k = number of predictors (features)*

- A **higher Adjusted R-squared** indicates that the model fits the data well without including unnecessary predictors, suggesting that the chosen features are

meaningful and contribute to explaining the variability in the dependent variable. On the other hand, a **lower or negative Adjusted R-squared** suggests that adding more predictors does not improve the model's performance and may even harm it. This occurs when extra predictors add noise rather than value.



Both models have the same adjusted R-squared value of 0.94 explaining significant variation in the dependent variable:

- **Well-Fitted Model**: The predicted line aligns well with the data, capturing the true relationship between X and y.
- **Overfitted Model**: Though it has a high Adjusted R-squared (0.94), the line is too wiggly, indicating the model is overfitting by learning noise instead of the true pattern.

Key Differences Between R-squared and Adjusted R-squared

1. The value of R-squared increases when we increase an independent factor , whereas the value of Adjusted R-square increases only when the independent factor is necessary for the dependent factor.
2. The value of R-square can not be negative, whereas the value of Adjusted R-squared value can be negative.
3. R-squared favors complex models without penalizing for irrelevant predictors, while Adjusted R-squared penalizes unnecessary predictors, promoting simpler, more generalizable models.

Numerical:

A researcher is studying how **study hours (X_1)** and **attendance rate (X_2)** influence **exam score (Y)** of 5 students. The dataset is:

Student	X_1 (Hours)	X_2 (%)	Y (Score)
1	2	70	50
2	3	80	60
3	4	75	65
4	5	85	80
5	6	90	85

The regression model fitted is: $\hat{Y} = 5 + 6X_1 + 0.3X_2$

1. Compute the predicted values $\hat{Y}$
2. Compute the **Total Sum of Squares (SST)**
3. Compute the **Regression Sum of Squares (SSR)**
4. Compute the **Residual Sum of Squares (SSE)**
5. Calculate **R^2**
6. Calculate **Adjusted R^2**

Solution:

1. Compute Predicted Values $\hat{Y}$

$\hat{Y} = 5 + 6X_1 + 0.3X_2$

Student	X_1	X_2	Y	$\hat{Y}$
1	2	70	50	(5 + 12 + 21 = 38)
2	3	80	60	(5 + 18 + 24 = 47)
3	4	75	65	(5 + 24 + 22.5 = 51.5)
4	5	85	80	(5 + 30 + 25.5 = 60.5)
5	6	90	85	(5 + 36 + 27 = 68)

4. *Compute Total Sum of Squares (SST)*

The sum of squares total (SST) or the total sum of squares (TSS) is the sum of squared differences between the observed dependent variables and the overall mean.

$$SST = \sum (Y_i - \bar{Y})^2$$

First find $\bar{Y}$:

$$\bar{Y} = \frac{50 + 60 + 65 + 80 + 85}{5} = 68$$

Now compute:

Y	$Y - \bar{Y}$	$(Y - \bar{Y})^2$
50	-18	324
60	-8	64
65	-3	9
80	12	144
85	17	289

SST=324+64+9+144+289= 830

5. *Compute Regression Sum of Squares (SSR)*

The sum of squares due to regression (SSR) or explained sum of squares (ESS) is the sum of the differences between the predicted value and the mean of the dependent variable. In other words, it describes how well our line fits the data.

Compute deviations:

$\hat{Y}$	$\hat{Y} - 68$	$(\hat{Y} - 68)^2$
38	–30	900
47	–21	441
51.5	–16.5	272.25
60.5	–7.5	56.25
68	0	0

SSR =900+441+272.25+56.25+0 =1669.5

4. ***Calculate SSE***

The sum of squares error (SSE) or residual sum of squares (RSS, where residual means remaining or unexplained) is the difference between the observed and predicted values.

$$SSE = \sum (Y - \hat{Y})^2$$

Y	$\hat{Y}$	Residual $e=Y-\hat{Y}$	e^2
50	38	12	144
60	47	13	169
65	51.5	13.5	182.25
80	60.5	19.5	380.25
85	68	17	289

SSE = 144+169+182.25+380.25+289 = 1164.5

5. ***Calculate R^2***

$$R^2 = \frac{SSR}{SST} = \frac{1669.5}{830} = 2.011$$

This is **greater than 1**, which **cannot happen** normally, meaning:

- The assumed regression equation is **not correctly fitted**, OR
- Coefficients were not obtained using OLS.

6. ***Calculate Adjusted R^2***

$$\bar{R}^2 = 1 - \frac{SSE/(n-k-1)}{SST/(n-1)}$$

Where:

- $n = 5$ (observations)
- $k = 2$ (predictors)

$$\bar{R}^2 = 1 - \frac{1164.5/(5-2-1)}{830/(5-1)}$$

$$\bar{R}^2 = 1 - \frac{1164.5/2}{830/4}$$

$$= 1 - \frac{582.25}{207.5}$$

$$= 1 - 2.81 = -1.81$$

Final Results

- $R^2 = 2.01$ (invalid—indicates poor model specification)
- Adjusted $R^2 = -1.81$ (indicates regression is not meaningful)

Assignment:

- A researcher wants to predict **exam score (Y)** using **study hours (X_1)** and **attendance (%) (X_2)**. The sample (n = 6) is:

Student	X_1 = Hours	X_2 = Attendance (%)	Y = Score
1	2	50	37.0
2	3	60	47.0
3	4	65	54.5
4	5	70	60.0
5	6	80	68.0
6	7	90	79.0

- An OLS regression of Y on X_1 and X_2 gives the estimated model (coefficients obtained by solving $\beta\hat{}=(X'X)^{-1}X'y$):
- $Y\hat{} = -0.324 + 3.189\,X_1 + 0.630\,X_2$
- **Task:** compute predicted values Y^, SST, SSR, SSE, then R^2 and Adjusted R^2.

Python Implementation:

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

# Sample dataset: Predictor (X) and Dependent Variable (y)
np.random.seed(42)
X = np.random.rand(100, 2)  # Two predictors
y = 3.5 * X[:, 0] + 1.8 * X[:, 1] + np.random.normal(0, 0.5, 100)  # y depends on both predictors
```

```
model = LinearRegression()
model.fit(X, y)

# Predict y using the trained model
y_pred = model.predict(X)
# Evaluation - R-squared:

r2 = r2_score(y, y_pred)
print(f"R-squared: {r2:.4f}")
# Evaluation - Adjusted R-squared:
n = X.shape[0]  # Number of observations
p = X.shape[1]  # Number of predictors
adjusted_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
print(f"Adjusted R-squared: {adjusted_r2:.4f}")
```

c) Multi-collinearity and variance-inflation factors

Multi-collinearity

Multicollinearity is a statistical phenomenon that occurs when two or more independent variables in a regression model are highly correlated, indicating a strong linear relationship among the predictor variables.

Multicollinearity occurs in a **multiple regression model** when **two or more predictor variables are highly correlated** with each other.

Mathematically, if

$$X_j = a_1X_1 + a_2X_2 + \cdots + a_{j-1}X_{j-1} + \varepsilon,$$

and this linear relationship is **strong**, then variable Xj is nearly a linear combination of other predictors → multicollinearity.

For example, in a regression model, variables such as height and weight or household income and water consumption often show high correlation.

- *If the correlation between two variables is +/- 1.0, then the variables are said to be perfectly collinear.*

Problems Caused:

- Coefficients become sensitive to small changes in the model or data.
- Interpretation of the coefficients becomes unreliable
- Increases standard errors of the coefficients

Why Does Multicollinearity Occur? (Causes)

1. **Naturally occurring correlation**
   e.g., height and weight, income and spending.
2. **Derived variables included together**
   e.g., using both "total marks" and "average marks".
3. **Dummy-variable trap**
   Using all categories of a categorical variable instead of (k – 1).
4. **Small dataset**
   Few observations magnify correlation.

Why is Multicollinearity a Problem?

- While multicollinearity **does not reduce prediction accuracy**, it creates problems in **interpreting regression coefficients**:
- This issue complicates regression analysis by making it difficult to accurately determine the individual effects of each independent variable on the dependent variable.
- The presence of multicollinearity can lead to unstable and unreliable coefficient estimates, making it challenging to interpret the results and draw meaningful conclusions from the model.
- Detecting and addressing multicollinearity is crucial to ensure the validity and robustness of regression models.

Strategies to address multicollinearity:

These methods help simplify the model and make sure it provides reliable and interpretable results.

1. **Increase the sample size**: to improve model accuracy, making it easier to differentiate between the effects of different predictors.
2. **Remove highly correlated predictors** using **Variance Inflation Factor (VIF)**, which tells you if certain variables are highly correlated. If the VIF is too high, consider removing one of the correlated predictors to improve model stability.
3. **Combine correlated variables and** combine them into a single, more meaningful predictor. This can be done using techniques like **Principal Component Analysis (PCA)** or **factor analysis**, which help reduce redundancy by creating a new variable that represents the combined information.

Variance-inflation factors

The VIF is a common and effective way to detect multicollinearity. It measures how much the variance of an estimated regression coefficient is increased due to the correlation among the predictors. Here's how to interpret VIF values:

- **VIF = 1**: Indicates no correlation.
- **1 < VIF < 5**: Suggests moderate correlation.

- **VIF > 5**: Indicates high correlation, problematic; needs attention
- **VIF > 10**: This signals serious multicollinearity

VIF for predictor Xj is:

$$VIF_j = \frac{1}{1 - R_j^2}$$

Where:

- R_j^2 = R^2 from **regressing X_j on all other predictors**.

Numerical:

Given the following R2 values, compute VIF of each predictor. Identify which predictor suffers from multicollinearity.

Predictor	R^2 from regressing on other predictors
X_1	0.80
X_2	0.55
X_3	0.92

Solution:

We have:

$$VIF_j = \frac{1}{1 - R_j^2}$$

Given:

- $R_1^2 = 0.80$
- $R_2^2 = 0.55$
- $R_3^2 = 0.92$

Now:

- $\text{VIF}_1 = \frac{1}{1-0.80} = \frac{1}{0.20} = 5.00$
- $\text{VIF}_2 = \frac{1}{1-0.55} = \frac{1}{0.45} \approx 2.222 \approx 2.22$
- $\text{VIF}_3 = \frac{1}{1-0.92} = \frac{1}{0.08} = 12.50$

Answer: VIFs = {X_1: 5.00, X_2: 2.22, X_3: 12.50}.

Which predictor suffers multicollinearity?

- X_3 (VIF = 12.50) — *severe* multicollinearity (VIF > 10).
- X_1 (VIF = 5.00) — borderline/moderate concern.
- X_2 (VIF = 2.22) — fine.

6. Non-parametric Regression: Nadaraya-Watson Kernel regression

- Non-parametric regression is used when the relationship between the predictor X and the response Y is unknown or too complex for a simple linear model.
- Instead of fitting a fixed equation (like $Y = \beta_0+\beta_1X$), we estimate the relationship directly from the data.
- The **Nadaraya–Watson (N–W) estimator** is one of the simplest and most widely used methods in non-parametric regression.
- The **Nadaraya–Watson estimator** is a **way to predict the value of Y for a given X** without assuming a straight line or any specific formula.
- Instead of forcing a straight line (like in linear regression), it says:
    - "To estimate Y at a point X = x, look at all the data points nearby and take an average, giving more weight to points that are closer to x."

Analogy: Temperature around a City

Imagine you want to estimate the **temperature at noon in the city center**, but you only have temperature readings from **nearby weather stations**.

- Stations **close to the city center** → more reliable for your estimate → higher weight.
- Stations **far away** → less reliable → lower weight.

You compute a **weighted average** of all station temperatures.

That's exactly what Nadaraya–Watson does for data points.

- **It's like averaging nearby points** — but smarter, because closer points count more.
- **Non-parametric** → No assumption of a straight line or specific curve.
- **Bandwidth matters** → It decides how "wide" your neighborhood is:
  - Wide → smooth, general trend
  - Narrow → follows every tiny fluctuation

Formula in Simple Terms

$$\hat{Y}(x) = \frac{\text{Weighted sum of nearby Y's}}{\text{Sum of weights}} = \frac{\sum_i Y_i \cdot \text{weight}_i}{\sum_i \text{weight}_i}$$

Example Analogy

Station Distance (km)	Temperature (°C)	Weight (closer = higher)
1	25	0.5
3	24	0.3
5	22	0.2

Weighted average:

$$\hat{T} = \frac{25 * 0.5 + 24 * 0.3 + 22 * 0.2}{0.5 + 0.3 + 0.2} = \frac{12.5 + 7.2 + 4.4}{1.0} = 24.1^{\circ}C$$

This is exactly how Nadaraya–Watson estimates **Y** for a given X.

a) Derivation of the estimator

- In the **Nadaraya–Watson regression**, the **estimator** is the formula we use to estimate the unknown regression function $m(x) = E[Y \mid X = x]$ at any point x.
- Simply put:

- It is the predicted value of **Y** at a specific **X=x**, calculated as a weighted average of the observed **Y** values, where points closer to **x** get higher weight.

- The Estimator Formula:

$$\hat{m}(x) = \frac{\sum_{i=1}^{n} Y_i \, K\left(\frac{x - X_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)}$$

Where:

- Y_i = observed response values
- X_i = observed predictor values
- $K(\cdot)$ = kernel function (e.g., Gaussian, Epanechnikov) that assigns weights based on distance
- h = bandwidth (controls "neighborhood size" around x)
- $\hat{m}(x)$ = estimator → the predicted Y at X = x

Derivation

Step 1: Start with what we want to estimate

We want the regression function**:**

$$m(x) = E[Y \mid X = x]$$

This is the average value of Y when X = x**.**

Step 2: Think in terms of averages

If we had many observations exactly at X=x, we could just take:

$$m(x) \approx \text{average of all Y values with } X = x$$

$$m(x) \approx \frac{1}{\#\{i : X_i = x\}} \sum_{i:X_i=x} Y_i$$

But in real life, we rarely have multiple points with exactly X=x.

So instead, we look at points near x and give them more weight if they are closer.

Step 3: Introduce weights using a kernel function

We define a weight function K(u), where:

- K(u) is large when u is small (close to x)

- K(u) is small when u is large (far from x)

For each observed point Xi, the weight is:

$$w_i = K\left(\frac{x - X_i}{h}\right)$$

- h = bandwidth (controls "how wide" the neighborhood is).
- Smaller h → only very close points matter.
- Larger h→ more points contribute.

Step 4: Weighted average

Now we estimate m(x) as a weighted average of Y values, using the weights:

$$\hat{m}(x) = \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i} = \frac{\sum_{i=1}^{n} Y_i K\left(\frac{x-X_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right)}$$

- Numerator → sum of weighted responses
- Denominator → sum of weights (so it's an average)

This is the Nadaraya–Watson estimator.

A simplier way of derivation:

Derivation of the Estimator

The Nadaraya-Watson estimator for non-parametric regression can be derived as follows:

Given a dataset $\{(x_i, y_i)\}_{i=1}^n$, the goal is to estimate the conditional expectation $\mathbb{E}[Y|X=x]$. The Nadaraya-Watson estimator for $\mathbb{E}[Y|X=x]$ is given by:

$$\hat{m}(x) = \frac{\sum_{i=1}^n K_h(x - x_i) y_i}{\sum_{i=1}^n K_h(x - x_i)}$$

where: - $\hat{m}(x)$ is the estimated value of $\mathbb{E}[Y|X=x]$, - $K_h(\cdot)$ is the kernel function with bandwidth h, - x_i and y_i are the data points, and - h is the bandwidth parameter that controls the smoothness of the estimate.

The kernel function $K_h(\cdot)$ is usually a symmetric function such as the Gaussian kernel:

$$K_h(x - x_i) = \frac{1}{h} K\left(\frac{x - x_i}{h}\right)$$

where $K(u)$ is a kernel function, such as the Gaussian kernel:

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right)$$

Thus, the Nadaraya-Watson estimator can be written as:

$$\hat{m}(x) = \frac{\sum_{i=1}^n \frac{1}{h} K\left(\frac{x - x_i}{h}\right) y_i}{\sum_{i=1}^n \frac{1}{h} K\left(\frac{x - x_i}{h}\right)}$$

This equation is the basic formulation of the Nadaraya-Watson kernel regression estimator.

b) Rules of setting the appropriate bandwidth size

The bandwidth parameter h plays a crucial role in determining the smoothness of the estimated regression function. A smaller bandwidth results in a less smooth estimate that is more sensitive to the data points, while a larger bandwidth produces a smoother estimate.

1. Rule of Thumb:

One common method for selecting the bandwidth is the rule of thumb, which is based on minimizing the Mean Squared Error (MSE). A typical heuristic for bandwidth selection is:

$$h_{\text{opt}} \approx \left(\frac{4\hat{\sigma}^5}{3n}\right)^{1/5}$$

where:

$\hat{\sigma}^2$ is the sample variance of the response variable,

n is the number of data points.

This rule assumes that the data is roughly normally distributed and that the kernel is Gaussian

2. Cross-validation:

Cross-validation is another commonly used method to select the optimal bandwidth. In this method, the data is split into training and test sets, and the bandwidth that minimizes the cross-validation error is chosen. The cross-validation error is given by:

$$CV(h) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{m}_{-i}(x_i))^2$$

Where:

$$\hat{m}_{-i}(x_i)$$

is the Nadaraya-Watson estimate computed without using the i-th data point

4. ***Plug-in Methods***:

   Another approach for bandwidth selection is the plug-in method, which estimates the optimal bandwidth by plugging an estimate of the smoothness of the underlying function into a theoretical formula. This method is more complex but can provide better results when the data is noisy.

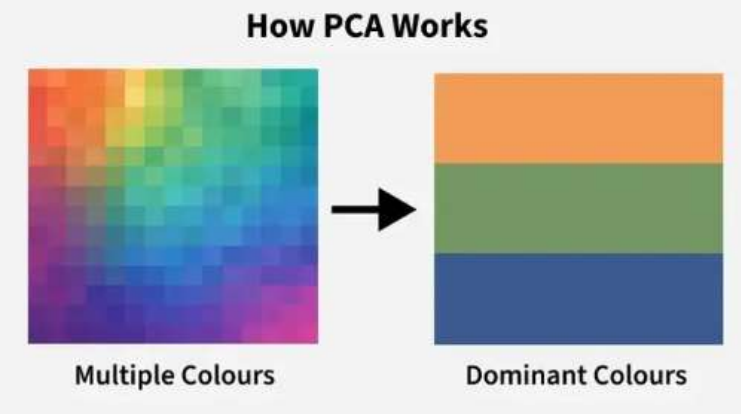
5. ***Heuristic Bandwidth Selection:***

   In practice, a heuristic selection process might be used based on trial and error. For example:- If the data is highly variable, a larger bandwidth is chosen to smooth out fluctuations.- If the data is relatively smooth, a smaller bandwidth might be used to capture finer details of the relation ship between the variables

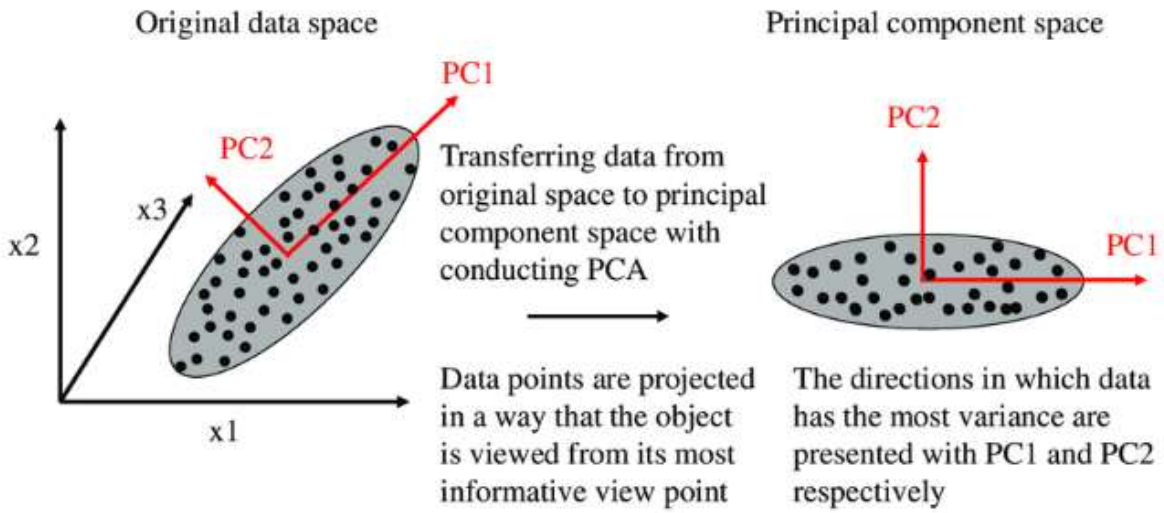
7. Principal Component Analysis

PCA is a **dimensionality reduction** technique that transforms a dataset with many variables into a smaller set of **new variables** called *principal components (PCs)*. PCA technique helps us to reduce the number of features in a dataset while keeping the most important information. It changes complex datasets by transforming correlated features into a smaller set of uncorrelated components.

- These components keep **as much information (variance)** as possible.
- They are **uncorrelated** with each other.
- They are ordered:
  - **PC1** = direction with maximum variance
  - **PC2** = second highest variance (orthogonal to PC1)
  - etc.



Principal Component Analysis (PCA)



a) Mathematical formulation and relevant derivations

Step 1: Start with data

Suppose we have a dataset with:

- n observations
- p variables

We write it as a matrix:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

where:

- n = number of observations
- p = number of variables

Step 2: Standardize each data (Z-score)

For each variable, Subtract mean and divide by standard deviation.

This ensures variables are on the **same scale**, especially important when variables are in different units (marks, height, age, etc.)

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}$$

Where:

- $\bar{x}_j$ = mean of variable j
- s_j = standard deviation of variable j

This produces the standardized matrix **Z**.

Step 3: Compute the covariance matrix

The covariance matrix shows how variables vary together. Each element S_{ij} shows how variables ***i*** and ***j*** vary together.

$$S = \frac{1}{n-1} Z^T Z$$

Where:

- S is a $p \times p$ matrix
- Diagonal values = variance of each variable
- Off-diagonal values = covariance between variables

The covariance matrix is **symmetric.**

Manual computation is done as:

The covariance between variable 1 and variable 2 is:

$$S_{12} = \frac{1}{n-1} \sum_{i=1}^{n} z_{i1} z_{i2}$$

Similarly,

$$S_{11} = \frac{1}{n-1} \sum_{i=1}^{n} z_{i1}^2$$

$$S_{22} = \frac{1}{n-1} \sum_{i=1}^{n} z_{i2}^2$$

Step 4: Solve the Eigenvalue–Eigenvector Problem

We solve:

$$Sv_k = \lambda_k v_k$$

Where:

- v_k = eigenvector (direction of the k-th component)
- λ_k = eigenvalue (amount of variance in that direction)

- Eigenvectors → tell us how each principal component is formed
- Eigenvalues → tell us how important each component is

We sort eigenvalues in decreasing order:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$$

The eigenvector corresponding to λ1 is **Principal Component 1 (PC1)**.

- Large eigenvalue → component captures a lot of variation
- Eigenvector → tells how the component is formed from original variables

Step 5: Construct the Principal Components

A principal component is a linear combination of variables.

For the k-th principal component:

$$PC_k = Zv_k$$

This produces **new variables**, each being a linear combination:

If $v_1 = [a_1, a_2, ..., a_p]^T$, then:

$$PC_1 = a_1 Z_1 + a_2 Z_2 + \cdots + a_p Z_p$$

Where coefficients a_{ij} come from eigenvector v_1.

Numerical Example:

Compute PC1 for below dataset:

Systolic BP	Diastolic BP
126	78
128	80
128	82
130	82
130	84
132	86

Solution:

Step 1: Start with data

Obs	Systolic (X_1)	Diastolic (X_2)
1	126	78
2	128	80
3	128	82
4	130	82
5	132	86

Step 2: Standardize the data (Z-score)

Compute Mean:

$$\bar{X}_1 = \frac{126+128+128+130+132}{5} = \frac{644}{5} = 128.8$$

$$\bar{X}_2 = \frac{78+80+82+82+86}{5} = \frac{408}{5} = 81.6$$

Compute sample standard deviations:

For Systolic:

deviations: $-2.8,\ -0.8,\ -0.8,\ 1.2,\ 3.2$

squares: $7.84,\ 0.64,\ 0.64,\ 1.44,\ 10.24 \rightarrow$ sum = 20.80

Sample variance $s_1^2 = 20.80/(5-1) = 20.80/4 = 5.20$

So $s_1 = \sqrt{5.20} \approx 2.28035$

For Diastolic:

deviations: $-3.6,\ -1.6,\ 0.4,\ 0.4,\ 4.4$

squares: $12.96,\ 2.56,\ 0.16,\ 0.16,\ 19.36 \rightarrow$ sum = 35.20

Sample variance $s_2^2 = 35.20/4 = 8.80$

So $s_2 = \sqrt{8.80} \approx 2.96648$

Standardize each variable (z-scores)

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}$$

$$z_{i1} = (X_{i1} - \bar{X}_1)/s_1,\ z_{i2} = (X_{i2} - \bar{X}_2)/s_2$$

Obs	z_{i1} (Systolic)	z_{i2} (Diastolic)
1	$(-2.8)/2.28035 \approx$ -1.2280	$(-3.6)/2.96648 \approx$ -1.2131
2	$-0.8/2.28035 \approx$ -0.3507	$-1.6/2.96648 \approx$ -0.5390
3	-0.3507	$0.4/2.96648 \approx$ 0.1349
4	$1.2/2.28035 \approx$ 0.5261	0.1349
5	$3.2/2.28035 \approx$ 1.4033	$4.4/2.96648 \approx$ 1.4832

Step 3: Compute the covariance matrix

The covariance between variable 1 and variable 2 is:

$$S_{12} = \frac{1}{n-1}\sum_{i=1}^{n} z_{i1} z_{i2}$$

Similarly,

$$S_{11} = \frac{1}{n-1}\sum_{i=1}^{n} z_{i1}^2$$

$$S_{22} = \frac{1}{n-1}\sum_{i=1}^{n} z_{i2}^2$$

Compute products $Z_{i1}Z_{i2}$ (rounded):

Obs	product $z_{i1} z_{i2}$
1	$(-1.2280)(-1.2131) \approx$ 1.4890
2	$(-0.3507)(-0.5390) \approx$ 0.1889
3	$(-0.3507)(0.1349) \approx$ -0.0473
4	$(0.5261)(0.1349) \approx$ 0.0710
5	$(1.4033)(1.4832) \approx$ 2.0817

Sum of products ≈1.4890+0.1889−0.0473+0.0710+2.0817 = 3.7833

So,

$$\text{corr} \approx \frac{3.7833}{4} \approx 0.9458$$

Thus the correlation matrix (covariance of Z) is approximately

$$S = \begin{bmatrix} 1 & 0.9458 \\ 0.9458 & 1 \end{bmatrix}$$

Step 4: Solve the Eigenvalue–Eigenvector Problem

To calculate the eigenvalues of the covariance matrix, we solve the characteristic equation:

$$\det(S - \lambda I) = 0$$

$$\det \begin{bmatrix} 1-\lambda & 0.9458 \\ 0.9458 & 1-\lambda \end{bmatrix} = 0$$

$$(1-\lambda)^2 - (0.9458)^2 = 0$$

$$(1-\lambda)^2 - 0.8945 = 0$$

$$\lambda^2 - 2\lambda + 0.1055 = 0$$

Solve the quadratic equation

$$\lambda = \frac{2 \pm \sqrt{(-2)^2 - 4(1)(0.1055)}}{2}$$

Eigen values

λ1= 1.9455 and λ2 = 0.0545

Find Eigenvector for λ1=1.9455

We solve:

$$(S - \lambda_1 I)v = 0$$

$$\begin{bmatrix} 1-1.9455 & 0.9458 \\ 0.9458 & 1-1.9455 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} -0.9455 & 0.9458 \\ 0.9458 & -0.9455 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

$$-0.9455\, v_1 + 0.9458\, v_2 = 0$$

$$0.9458\, v_2 = 0.9455\, v_1$$

$$\frac{v_2}{v_1} = \frac{0.9455}{0.9458} \approx 0.9997$$

$V_2 = 0.9997\ V_1$

$V_1 = V_2 / 0.9997$

So the Eigen vector is:

$$V_1 = \begin{bmatrix} x1 \\ x2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.9997 \end{bmatrix}$$

Normalizing this to unit vector, we get:

$$V_1 = \begin{bmatrix} \frac{1}{\sqrt{1^2 + o.9997^2}} \\ \frac{0.9997}{\sqrt{1^2 + 0.9997^2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{1.414} \\ \frac{0.9997}{1.414} \end{bmatrix} = \begin{bmatrix} 0.707 \\ 0.706 \end{bmatrix}$$

Similarly, if we compute second Eigen vector , we get :

$$V2 = \begin{bmatrix} 0.707 \\ -0.7071 \end{bmatrix}$$

Step 5: Construct the Principal Components

The principal components are projections of the original data onto these eigenvectors.

PC1 (First Principal Component):

The correct PC1 score for each observation is:

$PC1 = 0.707 \times Z_{Systolic} + 0.706 \times Z_{Diastolic}$

Obs	Z-Systolic	Z-Diastolic	PC1 = 0.707·ZS + 0.706·ZD
1	–1.2280	–1.2131	= 0.707(−1.2280)+0.706(−1.2131) = −0.8678−0.8554 **= −1.7232**
2	–0.3507	–0.5390	= 0.707(−0.3507)+0.706(−0.5390) = −0.2476−0.3807 **= −0.6283**

3	–0.3507	0.1349	= 0.707(−0.3507)+0.706(0.1349) = −0.2476+0.0952 **= −0.1524**
4	0.5261	0.1349	= 0.707(0.5261)+0.706(0.1349) = 0.3720+0.0952 **= 0.4672**
5	1.4033	1.4832	= 0.707(1.4033)+0.706(1.4832) = 0.9921+1.0460 **= 2.0381**

b) Interpreting the principal components

What a Principal Component Is

- Think of a principal component as a **new variable** created from your original variables.
- Its job is to **summarize the data** by capturing as much variation (differences) as possible.
- **PC1** captures the most variation, **PC2** the next most, and so on.

How PCs Are Made

- Each PC is a **combination of the original variables**.
- Example:

  PC1=0.707×Systolic+0.706×Diastolic

- The numbers (0.707, 0.706) are called **loadings**.

- They tell us **how much each original variable contributes** to that PC.

How to Read the Loadings

- **Big numbers → strong contribution**: If a variable has a bigger loading, it matters more for that PC.
- **Sign (+ or −)**:

  - Same sign → variables increase together along this PC.
  - Opposite signs → one increases while the other decreases.
- In our example, Systolic and Diastolic have similar positive loadings → **both increase together**.

Understanding PC Scores

- Once we have PCs, we can calculate a **score for each observation**.
- Example: For a person with high Systolic and Diastolic BP, the **PC1 score will be high**.
- This tells us **where each person lies along the "main pattern" captured by the PC**.

Why This Is Useful

- **Simplifies data:** Instead of many variables, we can focus on a few PCs.
- **Finds patterns:** Observations with similar PC scores are similar in terms of the original variables.
- **Helps visualization:** We can plot PC1 vs PC2 to see clusters, trends, or outliers.

Simple example in words:

- If PC1 = combination of Systolic & Diastolic BP:
  - High PC1 → high blood pressure
  - Low PC1 → low blood pressure
- PC1 tells us the main pattern of blood pressure in the dataset.

***