Regression and Predictive Modeling

Empirical Models

An **empirical model** refers to a mathematical or computational model that is based on observed data rather than on derived theoretical principles or physical laws. These models are created by analyzing patterns, relationships, and trends in real-world data to make predictions or explain phenomena.

Key Features of Empirical Models:

Data-Driven: They rely on experimental or observational data as their primary source of information.
Approximation: They approximate the relationships between variables without necessarily explaining the underlying cause-and-effect mechanisms.
Simplicity: Often, empirical models are simpler than theoretical models and focus on achieving practical, predictive accuracy.
Limited Generalizability: They may not always be applicable outside the conditions or range of the data used to develop them.
Statistical Techniques: Empirical models frequently use techniques like regression analysis, machine learning, or curve fitting.

Examples:

Linear Regression: Predicting house prices based on features like size, location, and number of rooms.

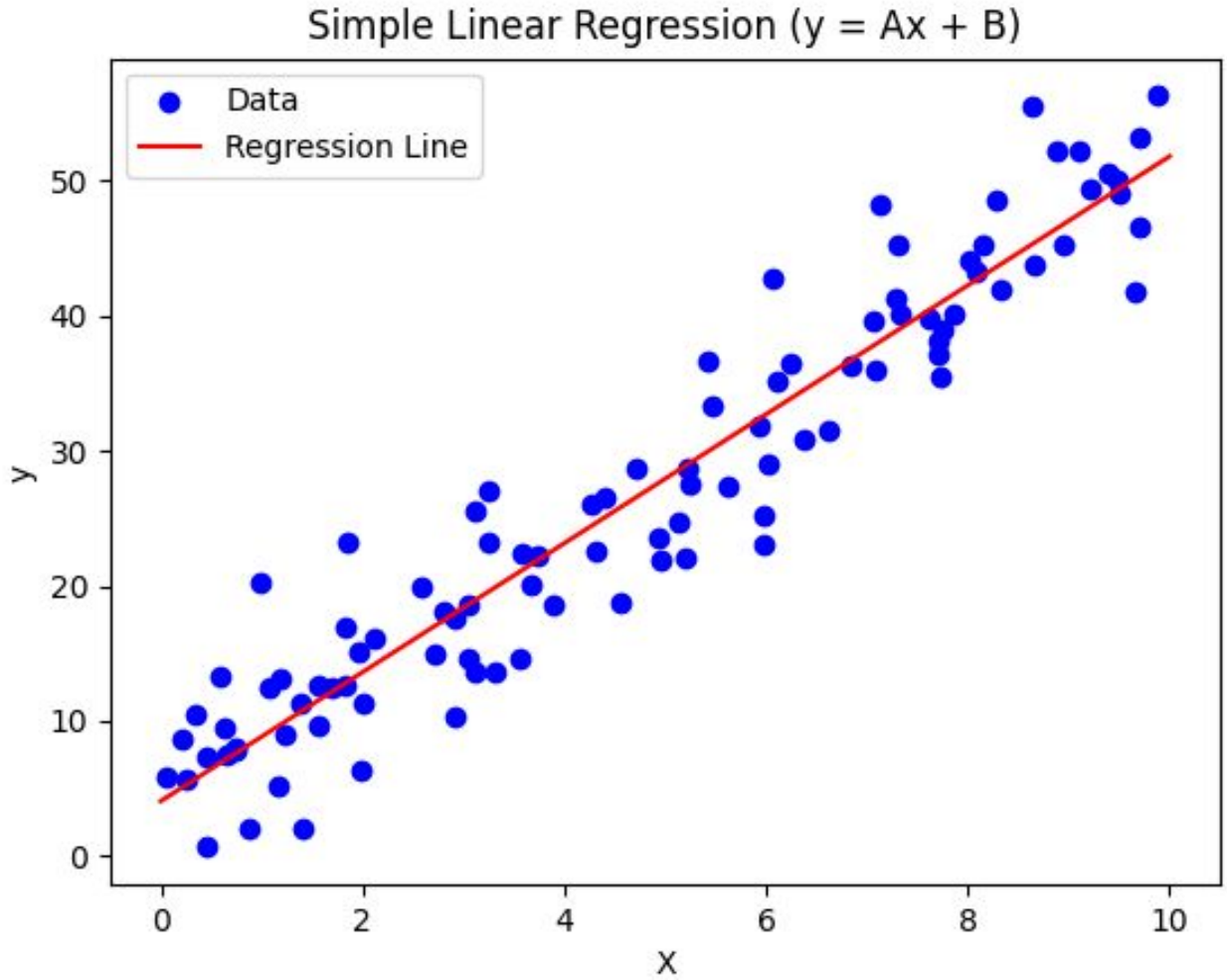
Weather Forecasting Models: Using historical weather data to predict future conditions.

Pharmacokinetics Models: Estimating how a drug is absorbed, distributed, metabolized, and excreted using experimental data.

Simple Linear Regression

Simple Linear Regression is a statistical method used to model the relationship between a dependent variable (response) and a single independent variable (predictor). The goal is to find a linear equation that best predicts the dependent variable based on the independent variable.

Y_i = observed value
$\hat{Y}_i$ = predicted value
$\hat{Y}_i = a + bX_i$


Simple Linear Regression (y = Ax + B)
Data
Regression Line
y
X
0
10
20
30
40
50
0
2
4
6
8
10

Our Goal is minimize Sum of Squared Error

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

$$SSE = \sum_{i=1}^{n} (y_i - a - bx_i)^2$$

$$\frac{\partial SSE}{\partial a} = 0 \qquad \frac{\partial SSE}{\partial b} = 0$$

$$\sum_{i=1}^{n} 2(y_i - a - bx_i)(-1) = 0$$

$$\sum_{i=1}^{n} 2(y_i - a - bx_i)(-x_i) = 0$$

$$\sum y_i = na + b \sum x_i$$

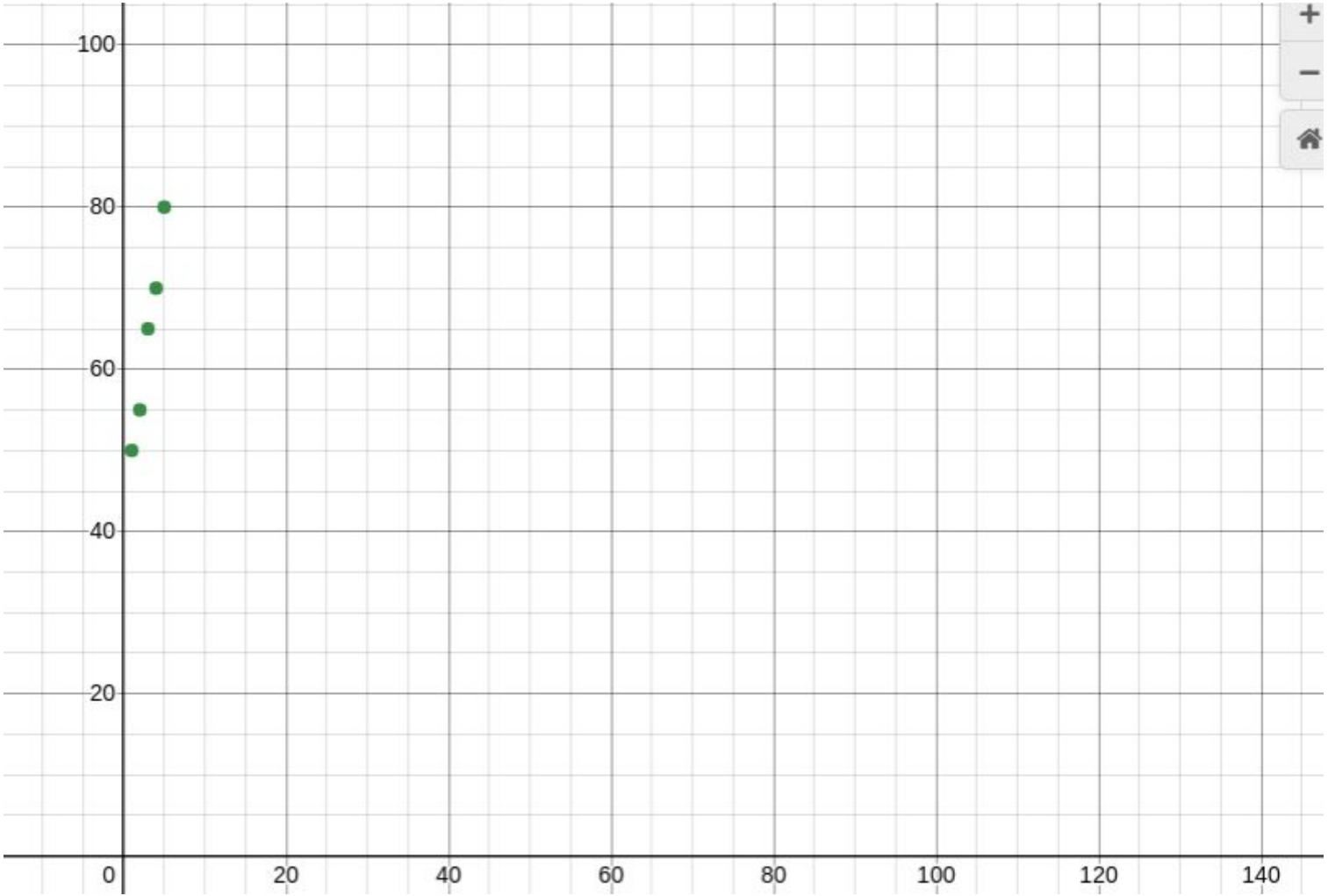
$$\sum x_i y_i = a \sum x_i + b \sum x_i^2$$

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix}$$

Solve to find a and b

Example

X (Hours Studied)	Y(Exam Score)
1	50
2	55
3	65
4	70
8	80


100
80
60
40
20
0
20
40
60
80
100
120
140

Solution

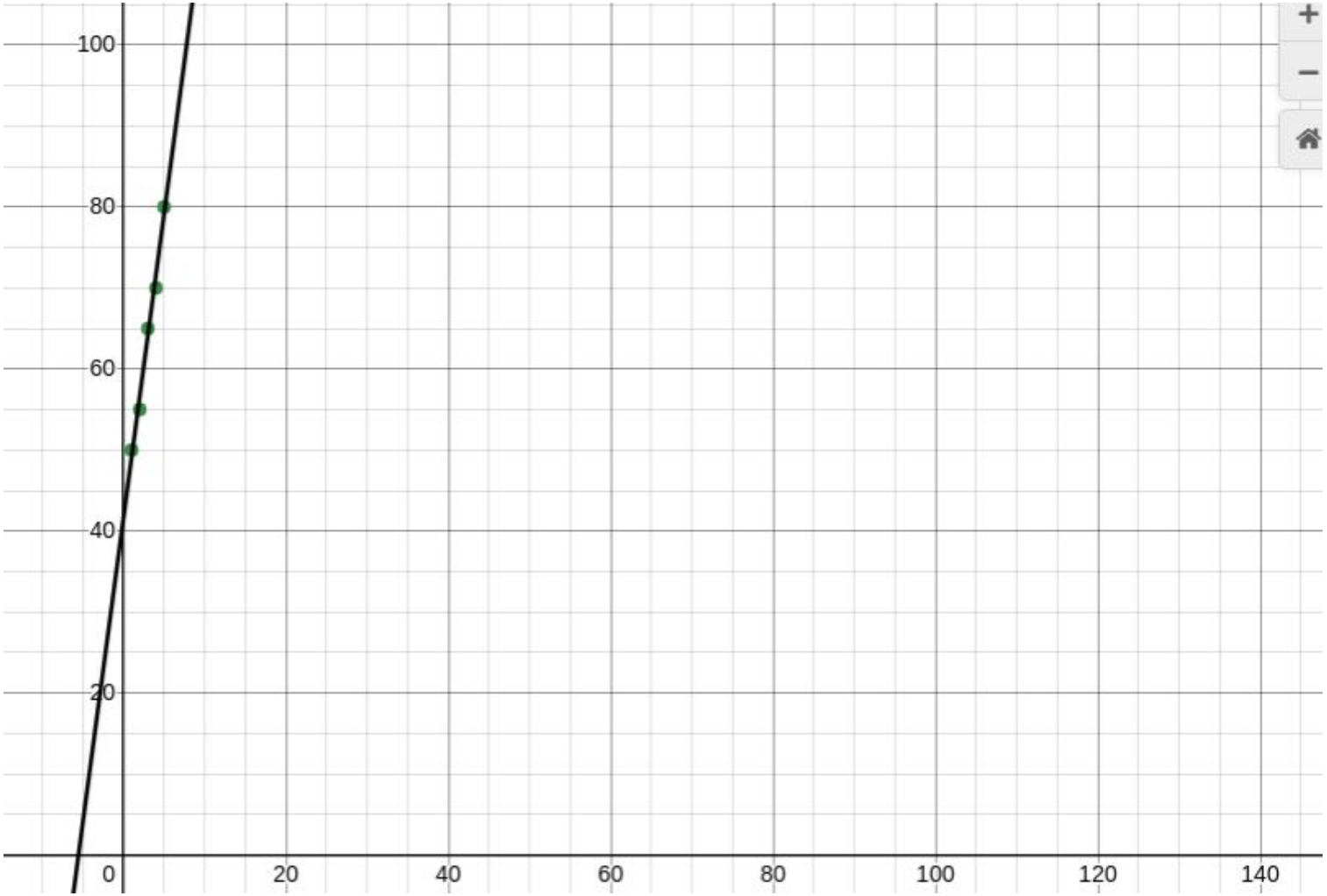
x	**y**	$\mathbf{x^2_i}$	$\mathbf{x_i y_i}$
1	50	1	50
2	55	4	110
3	65	9	195
4	70	16	280
5	80	25	400
$\mathbf{\sum x = 15}$	$\mathbf{\sum y = 320}$	$\mathbf{\sum x^2 = 55}$	$\mathbf{\sum xy = 1035}$

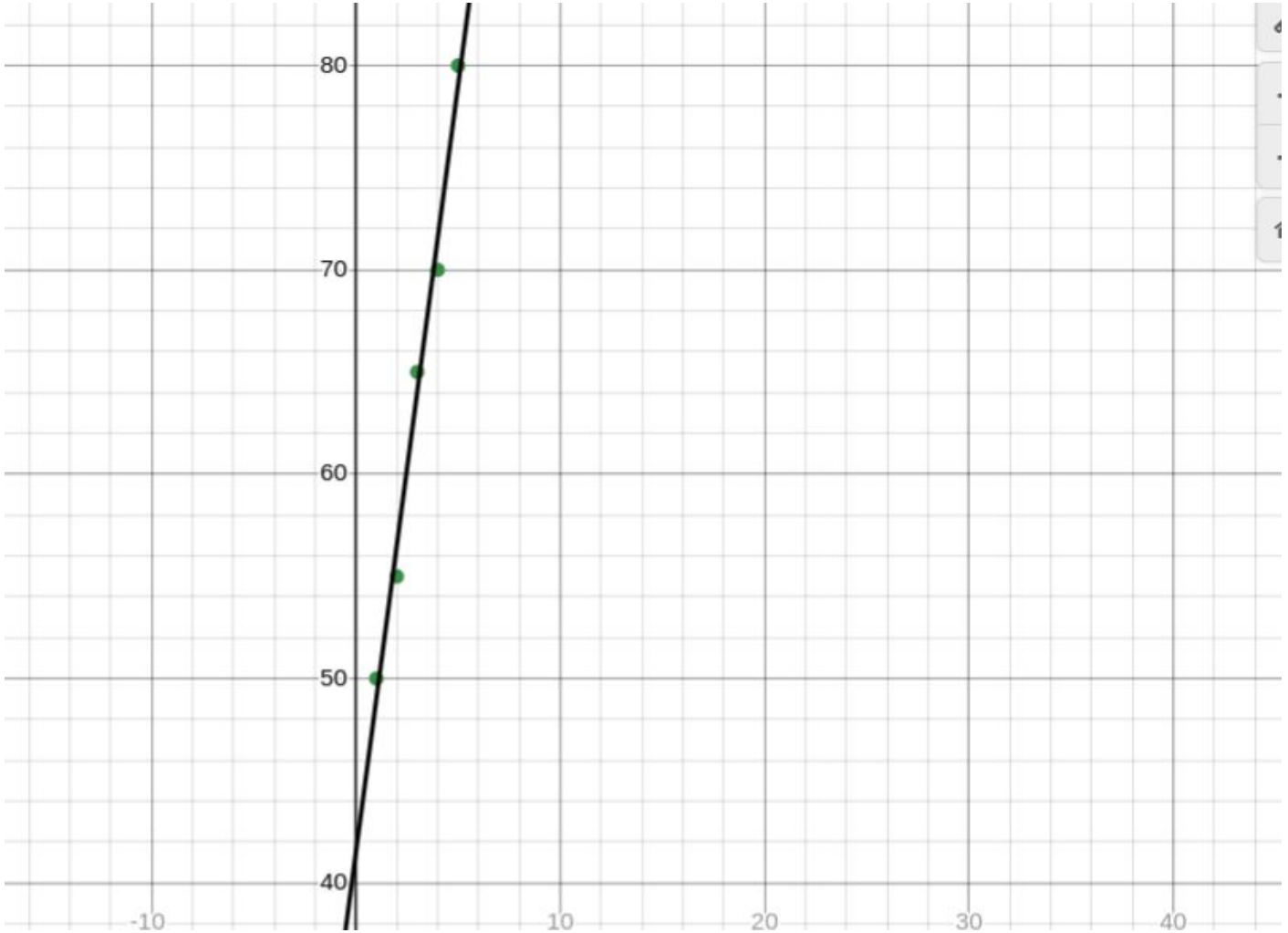
$$\begin{bmatrix} 5 & 15 \\ 15 & 55 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 320 \\ 1035 \end{bmatrix}$$

a = 41.5

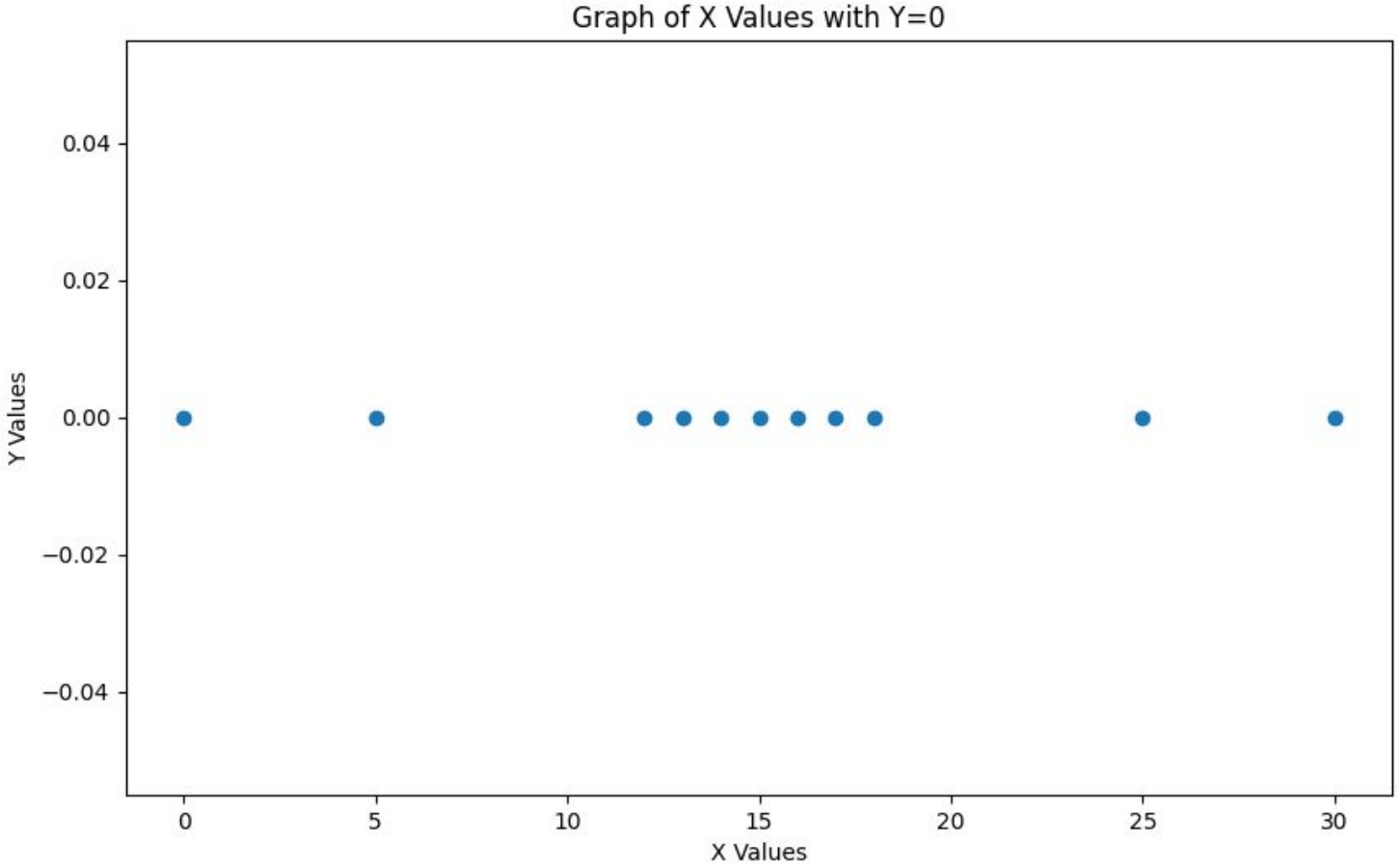
b = 7.5

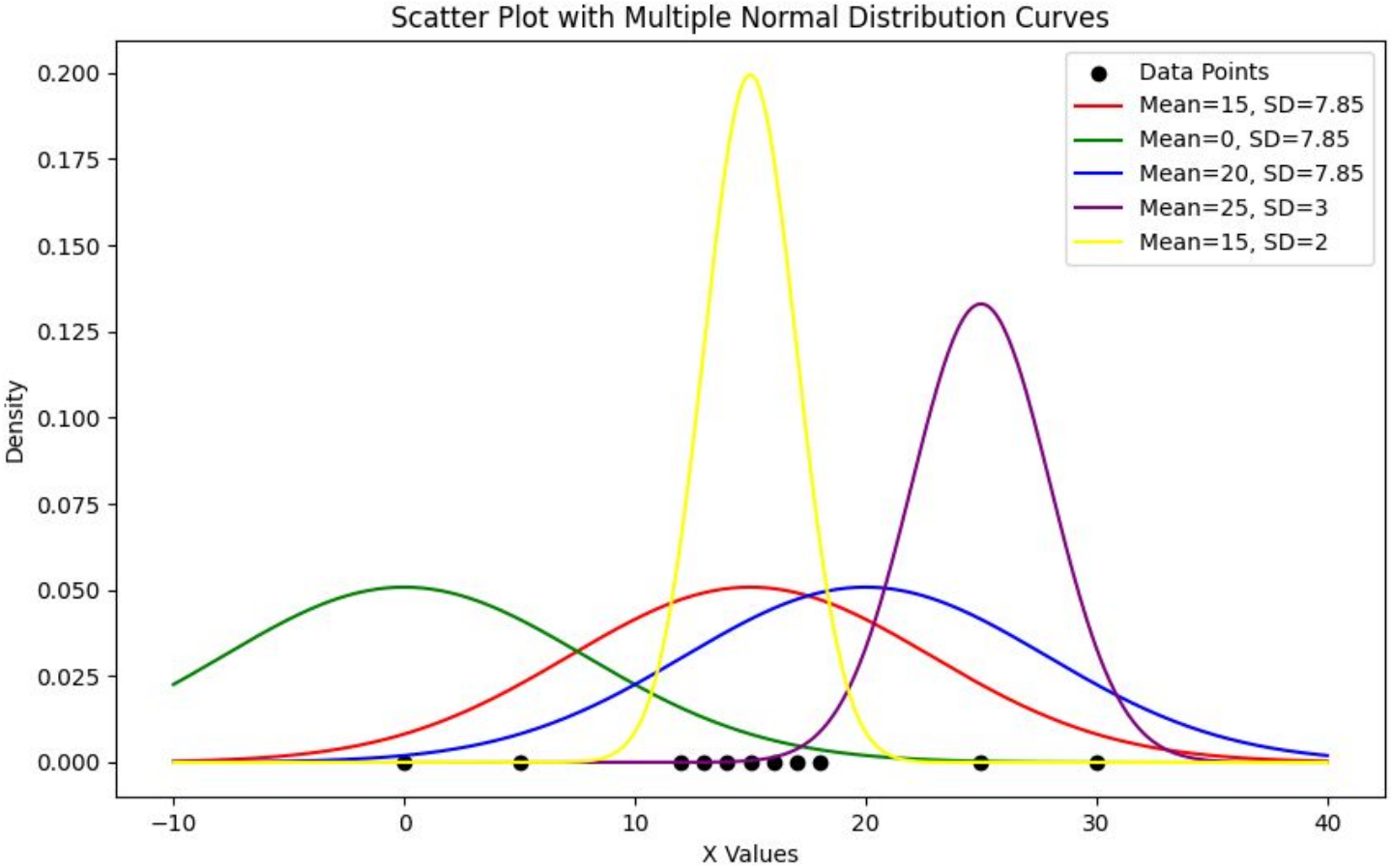
y = 41.5 + 7.5x

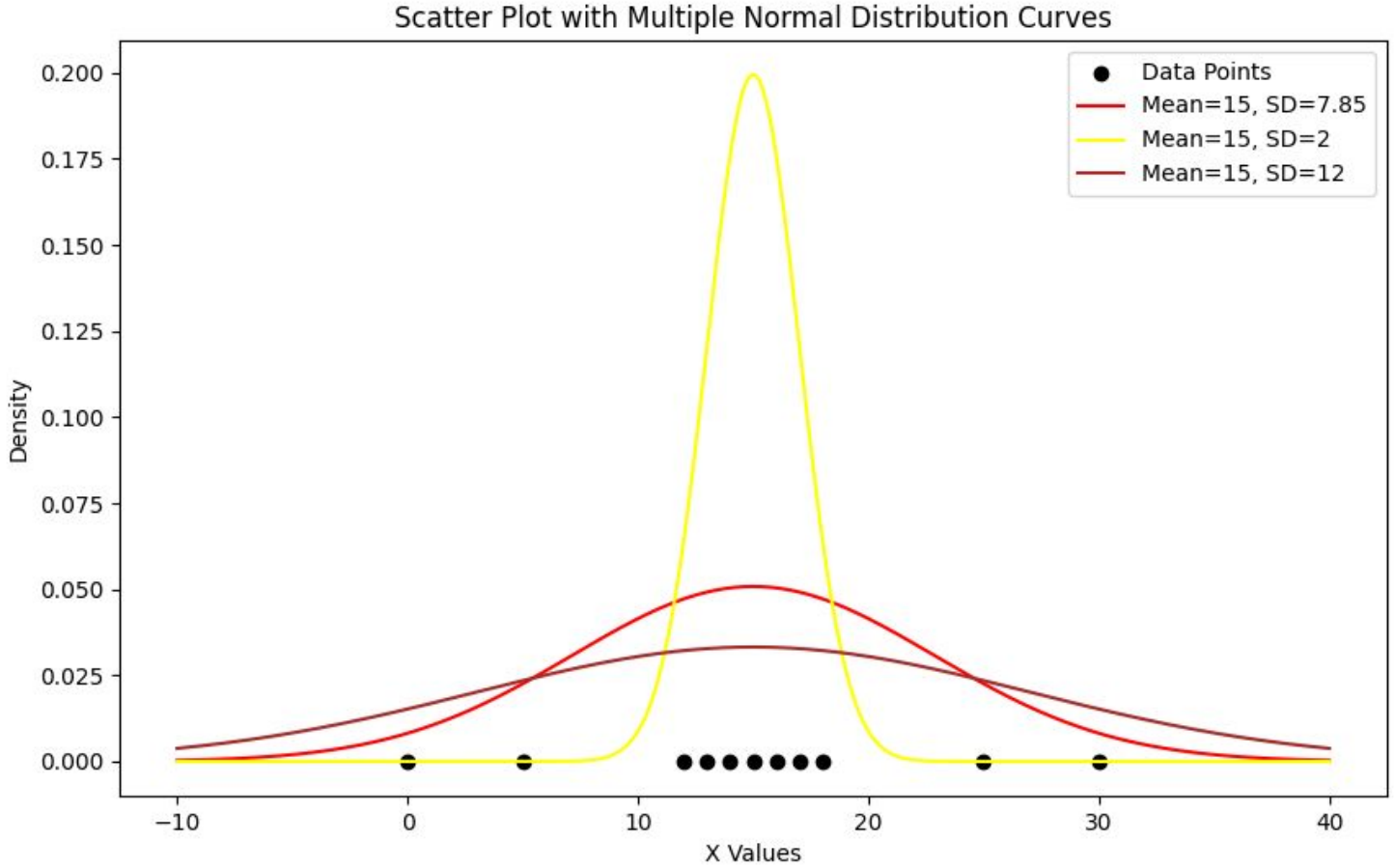

100
80
60
40
20
0
20
40
60
80
100
120
140


80
70
60
50
40
-10
10
20
30
40

Maximum Likelihood Estimation


Graph of X Values with Y=0
0.04
0.02
0.00
−0.02
−0.04
Y Values
0
5
10
15
20
25
30
X Values


Scatter Plot with Multiple Normal Distribution Curves
Data Points
Mean=15, SD=7.85
Mean=0, SD=7.85
Mean=20, SD=7.85
Mean=25, SD=3
Mean=15, SD=2
Density
X Values
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−10
0
10
20
30
40


Scatter Plot with Multiple Normal Distribution Curves
Data Points
Mean=15, SD=7.85
Mean=15, SD=2
Mean=15, SD=12
Density
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−10
0
10
20
30
40
X Values

Maximum Likelihood Estimation (MLE) for Normal Distribution

The Maximum Likelihood Estimation (MLE) method provides estimates of the parameters of a probability distribution by maximizing the likelihood function. For a normal distribution, the two parameters are the mean (μ) and the variance (σ2).

Normal Distribution PDF

The probability density function (PDF) of the normal distribution is:

$$f(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Given a dataset X={x1,x2,…,xn}the likelihood function for the parameters μ and σ2 is:

$$L(\mu,\sigma^2) = \prod_{i=1}^{n} f(x_i;\mu,\sigma^2)$$

The log-likelihood function is:

$$\ell(\mu,\sigma^2) = \log L(\mu,\sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

MLE for μ (Mean)

To find the MLE for μ, take the derivative of ℓ(μ,σ2) with respect to μ and set it to zero:

$$\frac{\partial \ell}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^{n} (x_i - \mu) = 0$$

Simplifying:

$$\mu_{\text{MLE}} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Thus, the MLE for the mean is the sample mean.

MLE for σ2 (Variance)

To find the MLE for σ2, take the derivative of ℓ(μ,σ2) with respect to σ2 and set it to zero:

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \sum_{i=1}^{n} (x_i - \mu)^2 = 0$$

Simplifying:

$$\sigma^2_{\text{MLE}} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2$$

Thus, the MLE for the variance is the sample variance .

Multiple Linear Regression

Multiple Linear Regression is a statistical technique used to model the relationship between one dependent variable (response variable) and two or more independent variables (predictor variables). It extends simple linear regression, which involves only one independent variable, to handle multiple predictors.

Model Equation

The general equation for multiple linear regression is:

$y = \square 0 + \square 1x1 + \square 2x2 + \square 3x3 + \ldots + \square pxp + \epsilon$

Where:

- y: Dependent variable (target or response).
- x1,x2,…,xp: Independent variables (predictors or features).
- β0: Intercept (value of yyy when all predictors are 0).
- β1,β2,…,βp: Coefficients for the predictors (βj represents the effect of xj on y).
- ϵ: Error term (accounts for randomness and unobserved factors).

Assumptions

Linearity: The relationship between the dependent variable and predictors is linear.

Independence: Observations are independent of each other.

Homoscedasticity: The variance of residuals is constant across all levels of the predictors.

Normality: The residuals are normally distributed.

No Multicollinearity: Predictors are not highly correlated with each other.

Multiple Linear Regression for Two Independent Variables

Yi = observed value

Ŷi = predicted value

Ŷi = □0 + □1x1i + □2x2i

We need to minimize

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

$$SSE = \sum_{i}^{n} (y_i - \beta_0 - \beta_1 x_{1i} - \beta_2 x_{2i})^2$$

$$\frac{\partial SSE}{\partial \beta_0} = 0 \qquad \frac{\partial SSE}{\partial \beta_1} = 0 \qquad \frac{\partial SSE}{\partial \beta_2} = 0$$

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{21} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i} x_{2i} \\ \sum x_{2i} & \sum x_{1i} x_{2i} & \sum x_{2i}^2 \end{bmatrix} x \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_{1i} y_i \\ \sum x_{2i} y_i \end{bmatrix}$$

Example

x1= height in inches

x2 = age in years

y = weight in pounds

Estimate weight of a boy who is 9 years old and 54 inches tall

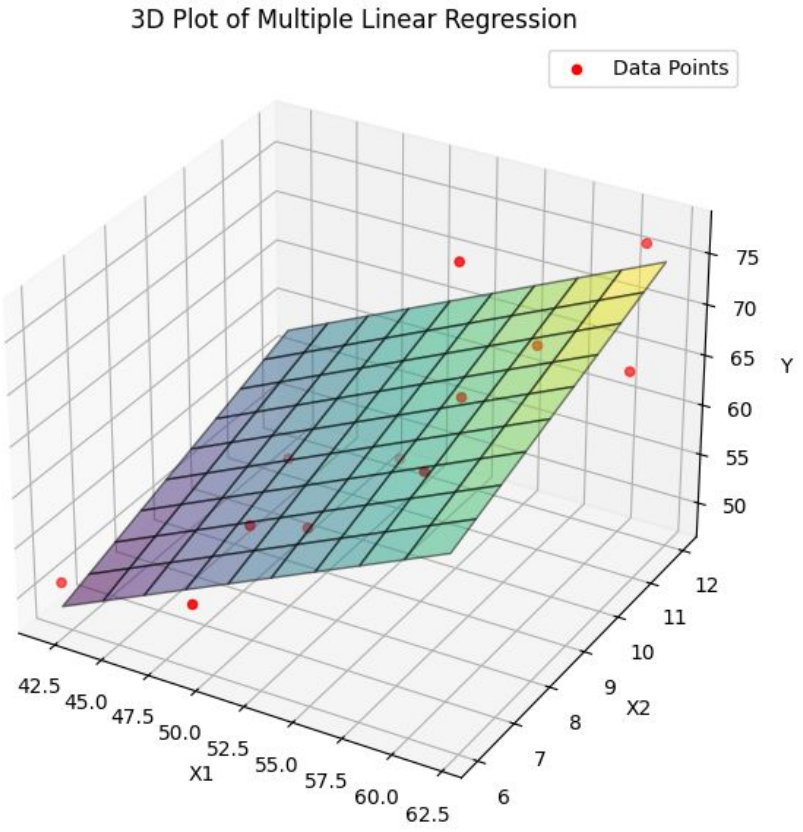
x1	57	59	49	62	51	50	55	48	52	42	61	57
x2	8	10	6	11	8	7	10	9	10	6	12	9
y	64	71	53	67	55	58	77	57	56	51	76	68

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{21} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i}x_{2i} \\ \sum x_{2i} & \sum x_{1i}x_{2i} & \sum x_{2i}^2 \end{bmatrix} x \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_{1i}y_i \\ \sum x_{2i}y_i \end{bmatrix}$$

$$\begin{bmatrix} 12 & 643 & 106 \\ 643 & 34843 & 5779 \\ 106 & 5779 & 97 \end{bmatrix} X \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} 753 \\ 408030 \\ 6796 \end{bmatrix}$$

□0 = 3.65, □1 = 0.8546, □2 = 1.5063

Estimated weight = 63 pound


3D Plot of Multiple Linear Regression
Data Points
75
70
65
60
55
50
Y
42.5
45.0
47.5
50.0
52.5
55.0
57.5
60.0
62.5
X1
6
7
8
9
10
11
12
X2

Multiple Linear Regression with Three Independent Variables

Yi = observed value

Ŷi = predicted value

Ŷi = □0 + □1x1 + □2x2 +□3x3

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{2i} & \sum x_{3i} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i}x_{2i} & \sum x_{1i}x_{3i} \\ \sum x_{2i} & \sum x_{1i}x_{2i} & \sum x_{2i}^2 & \sum x_{2i}x_{3i} \\ \sum x_{3i} & \sum x_{1i}x_{3i} & \sum x_{2i}x_{3i} & \sum x_{3i}^2 \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_{1i}y_i \\ \sum x_{2i}y_i \\ \sum x_{3i}y_i \end{bmatrix}$$

Multiple Linear Regression with n Independent Variables

Yi = observed value

Ŷi = predicted value

Ŷi = □0 + □1x1 + □2x2 +□3x3+ … +□nxn

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{2i} & \cdots & \sum x_{ni} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i}x_{2i} & \cdots & \sum x_{1i}x_{ni} \\ \sum x_{2i} & \sum x_{1i}x_{2i} & \sum x_{2i}^2 & \cdots & \sum x_{2i}x_{ni} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum x_{ni} & \sum x_{1i}x_{ni} & \sum x_{2i}x_{ni} & \cdots & \sum x_{ni}^2 \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_{1i}y_i \\ \sum x_{2i}y_i \\ \vdots \\ \sum x_{ni}y_i \end{bmatrix}$$

Polynomial Regression

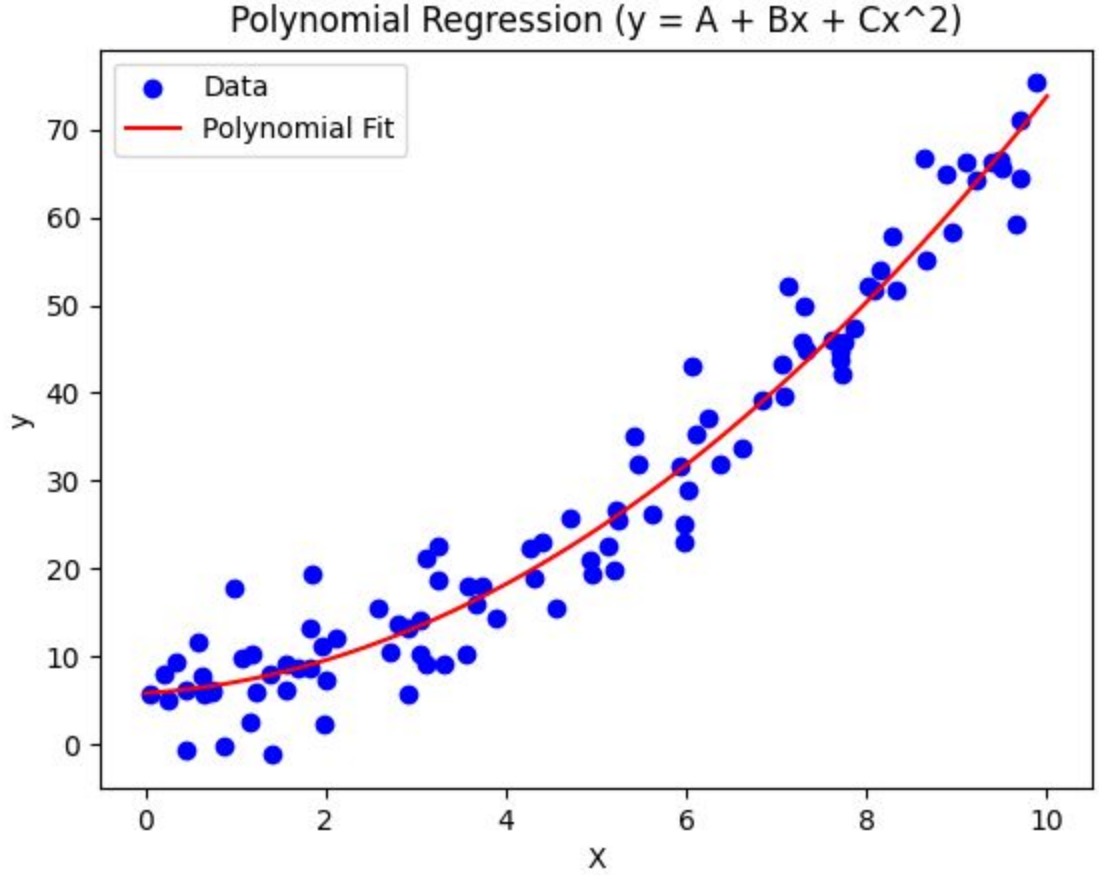
$Y_i = \square 0 + \square 1x + \square 2x^2 + e$

$\hat{Y}_i = \square 0 + \square 1x + \square 2x^2$

Y_i = observed value

$\hat{Y}_i$ = predicted value

e = Error

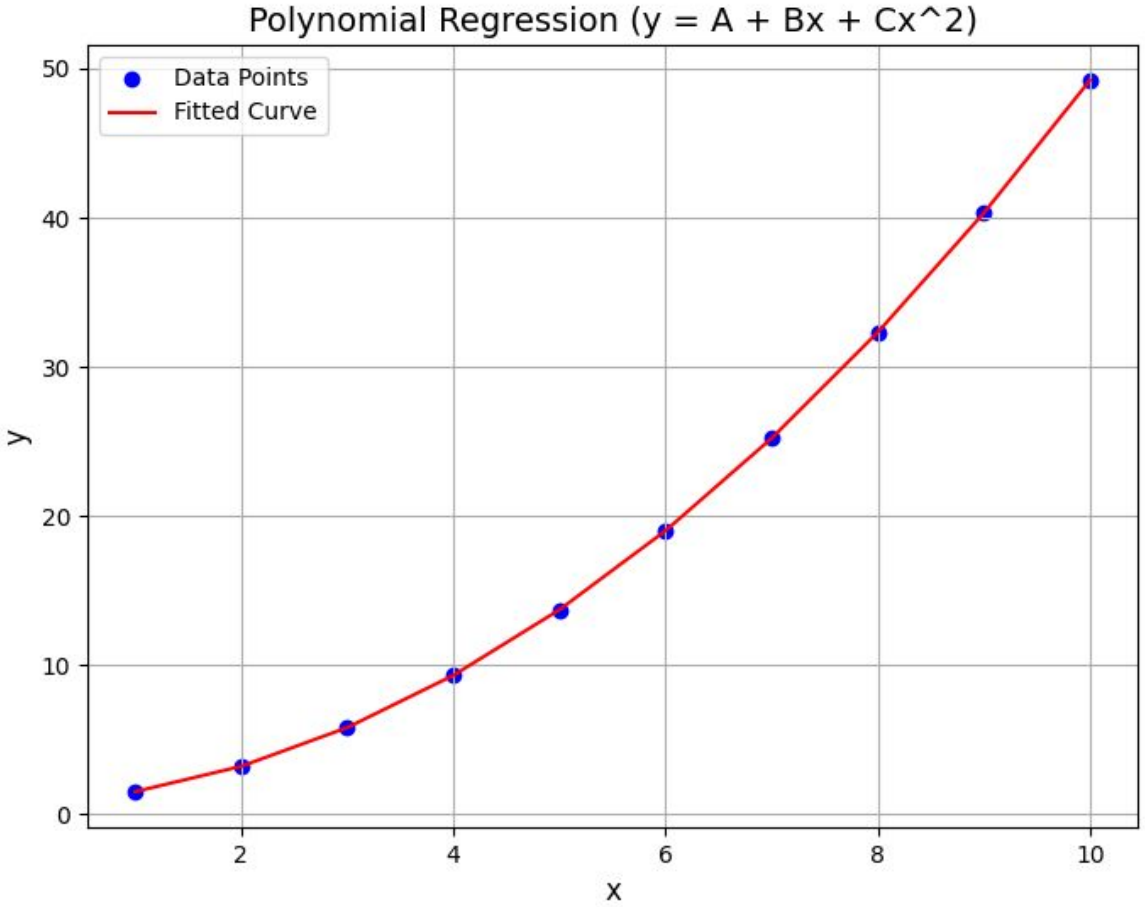

Polynomial Regression (y = A + Bx + Cx^2)
Data
Polynomial Fit
y
X
0
10
20
30
40
50
60
70
0
2
4
6
8
10

Example

Fit $y = A + Bx + Cx^2$

x	1	2	3	4	5	6	7	8	9	10
y	1.5	3.2	5.8	9.3	13.7	19	25.2	32.3	40.3	49.2

A = 0.7, B=0.35,C=0.45


Polynomial Regression (y = A + Bx + Cx^2)
Data Points
Fitted Curve
y
x
0
10
20
30
40
50
2
4
6
8
10

Categorical Regressor

Categorical Regressors

These are categorical variables used as predictors (independent variables) in the regression models.

These variables represent discrete categories rather than continuous variables.

Examples:

- Gender: (male,female)
- Color: (Red, Blue, Green)
- Region: (North, South, East, West)

Since regression models typically requires numerical inputs, categorical variables must be converted into numerical values.

Common Technique (Categorical to Numerical)

1. One Hot Encoding

   Creates binary columns for each category in the variable.

   Example:

   Original Variable: color = [red,blue,green]

   After Encoding:

   Red : [1,0,0]

   Blue: [0,1,0]

   Green: [0,0,1]

Dataset: Customer Preferences for Fruit

Customer ID	Favorite Fruit
1	Apple
2	Banana
3	Orange
4	Apple
5	Grape

After Applying One-Hot Encoding:

Customer ID	Apple	Banana	Orange	Grape
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	1	0	0	0
5	0	0	0	1

Use Case:

- Suitable for **nominal data** where there is no inherent order or ranking (e.g., colors, genders, product categories).
- Ideal when using machine learning models that assume no ordinal relationships between features (e.g., decision trees, neural networks).

Common Technique (Categorical to Numerical)

2. Ordinal Encoding
   Assigns unique integer to each category.
   Example:
   Use for `Education Levels` (`['High School', 'Bachelor', 'Master', 'PhD']`) where the levels are ordered but not equidistant.
   After Encoding:
   High School : 0
   Bachelor: 1
   Master: 2
   PhD: 3

Dataset: Customer Feedback on Service Quality

Customer ID	Service Quality
1	Poor
2	Fair
3	Good
4	Very Good
5	Excellent

After Applying Ordinal Encoding:

Customer ID	Service Quality (Encoded)
1	1
2	2
3	3
4	4
5	5

Why Ordinal Encoding?

- **Ordinal Data**: "Service Quality" has a meaningful order that should be captured in the encoding.
- Helps algorithms that can leverage ordinal relationships (e.g., linear regression or some tree-based models).

Common Technique (Categorical to Numerical)

3. Label Encoding

   Assigns unique integer to each category.

   Example:

   Original Variable: color = [red,blue,green]

   After Encoding:

   Red : 0

   Blue: 1

   Green: 2

Dataset: Pet Ownership

Owner ID	Pet Type
1	Dog
2	Cat
3	Rabbit
4	Dog
5	Rabbit

After Applying Label Encoding:

Owner ID	Pet Type (Encoded)
1	0
2	1
3	2
4	0
5	2

Why Label Encoding?

- **Nominal Data**: "Pet Type" has no inherent order or ranking.
- Suitable when working with models that handle categorical values as integers naturally, such as decision trees or random forests, without implying ordinal relationships.

Common Technique (Categorical to Numerical)

4. Target Encoding
   Replaces each variable with statistic (eg. mean of target variable for that category)
   Example:
   Categories and their corresponding average sales.
   Red: 100
   Blue: 150
   Green: 120

Common Technique (Categorical to Numerical)

5. Frequency Encoding

   Encodes each category based on its frequency in the dataset.

   Example

   Red: 50

   Blue: 20

   Green: 30

Feature Selection Techniques

Forward Selection

Forward Selection is a **sequential feature selection technique** used in feature selection to iteratively add features to a model based on their predictive power. It begins with an empty model and adds features one at a time until a stopping criterion is met, such as a desired number of features, improvement threshold, or model performance stagnation.

Let's consider a simple example to illustrate forward selection. Assume we have a dataset with 5 features (X1,X2,X3,X4,X5) and a target variable y. We'll use Forward Selection to determine the most relevant features.

We use Linear Regression as the model and evaluate performance using R^2(coefficient of determination).

Step by Step

Step 1: Start with an Empty Model

Initial Model: No Features

1. Calculate R^2 for each feature individually:
    - $R^2(X1)=0.6$
    - $R^2(X2)=0.55$
    - $R^2(X3)=0.45$
    - $R^2(X4)=0.50$
    - $R^2(X5)=0.40$

Choose X1, as it gives the highest R2=0.6

Step 2: Add X1 to the Model

Model: X1

Now calculate R^2 for models combining X1 with each remaining feature:

- R2(X1,X2)=0.72
- R2(X1,X3)=0.68
- R2(X1,X4)=0.70
- R2(X1,X5)=0.65

Choose X2, as it gives the highest R^2=0.72.

Step 3: Add X2 to the Model

Model: X1,X2

Now calculate R^2 for models combining X1,X2 with each remaining feature:

- R^2(X1,X2,X3)=0.75
- R^2(X1,X2,X4)=0.78
- R^2(X1,X2,X5)=0.74

Choose X4, as it gives the highest R^2=0.78.

Step 4: Add X4 to the Model

Model: X1,X2,X4

Stop if R^2 improvement is below the threshold (e.g., 0.02) or desired number of features is reached.

Final Model

Selected Features: X1,X2,X4
Final R2=0.78

This process shows how Forward Selection iteratively builds a model by adding the most predictive features at each step.

Backward Elimination

Backward Elimination is a **sequential feature selection technique** where all features are initially included in the model, and features are removed one at a time based on their lack of contribution to the model's performance. The process continues until only the most significant features remain, satisfying a specific stopping criterion.

Steps in Backward Elimination:

1. **Start with all features**: Build an initial model that includes all independent variables.
2. **Fit the model**: Use a statistical method (e.g., linear regression) to fit the model.
3. **Identify the least significant variable**: Based on a significance test (like the p-value in regression analysis), identify the variable with the highest p-value (i.e., least significant variable).
4. **Eliminate the least significant variable**: Remove this variable if its p-value is greater than a predefined significance level (e.g., 0.05).

Steps in Backward Elimination:

5. **Refit the model**: Fit the model again using the remaining variables.
6. **Repeat**: Repeat steps 3–5 until all remaining variables in the model have p-values below the significance threshold.
7. **Finalize the model**: The remaining variables constitute the final model.

Example

Dataset:

Observation	X_1	X_2	X_3	Y
1	2	3	5	10
2	4	6	8	20
3	6	9	11	30
4	8	12	14	40
5	10	15	17	50

Step 1: Fit the full model

We fit a multiple linear regression model:

$Y=\beta0+\beta1X1+\beta2X2+\beta3X3$

Assume the output from the regression gives us the following coefficients and p-values:

Variable	Coefficient (β)	p-value
Intercept	0.0	-
X_1	1.5	0.03
X_2	0.5	0.15
X_3	0.2	0.08

The significance level (α) is set at **0.05**.

Step 2: Identify the least significant variable

- X2 has the highest p-value (0.15>0.05).
- X2 is the least significant variable.

Step 3: Eliminate X2

We remove X2 from the model and refit the regression using X1 and X3X_3X3.

$$Y = \beta_0 + \beta_1 X_1 + \beta_3 X_3$$

The new regression output is as follows:

Variable	Coefficient (β)	p-value
Intercept	0.0	-
X_1	1.7	0.02
X_3	0.1	0.06

Step 4: Repeat the process

- X3 now has the highest p-value (0.06>0.05).
- X3 is the least significant variable.

Step 5: Eliminate X3

We remove X3 from the model and refit the regression using only X1.

$Y=\beta0+\beta1X1$
The new regression output is:

Variable	Coefficient (β)	p-value
Intercept	0.0	-
X_1	2.0	0.01

Step 6: Final model

Since X1 has a p-value (0.01<0.05), we stop here.

The final model is:

$Y=2.0X1$

Summary

- **Initial model**: $Y=\beta0+\beta1X1+\beta2X2+\beta3X3$
- **Final model**: $Y=2.0X1$

  This demonstrates backward elimination by manually removing variables based on p-values.

Stepwise Selection

Stepwise selection is a feature selection technique that iteratively builds or reduces a regression model by adding or removing predictors based on statistical criteria, such as p-values or metrics like AIC (Akaike Information Criterion). It combines aspects of **forward selection** and **backward elimination**, aiming to find the best subset of predictors.

Model Performance

Residual Analysis

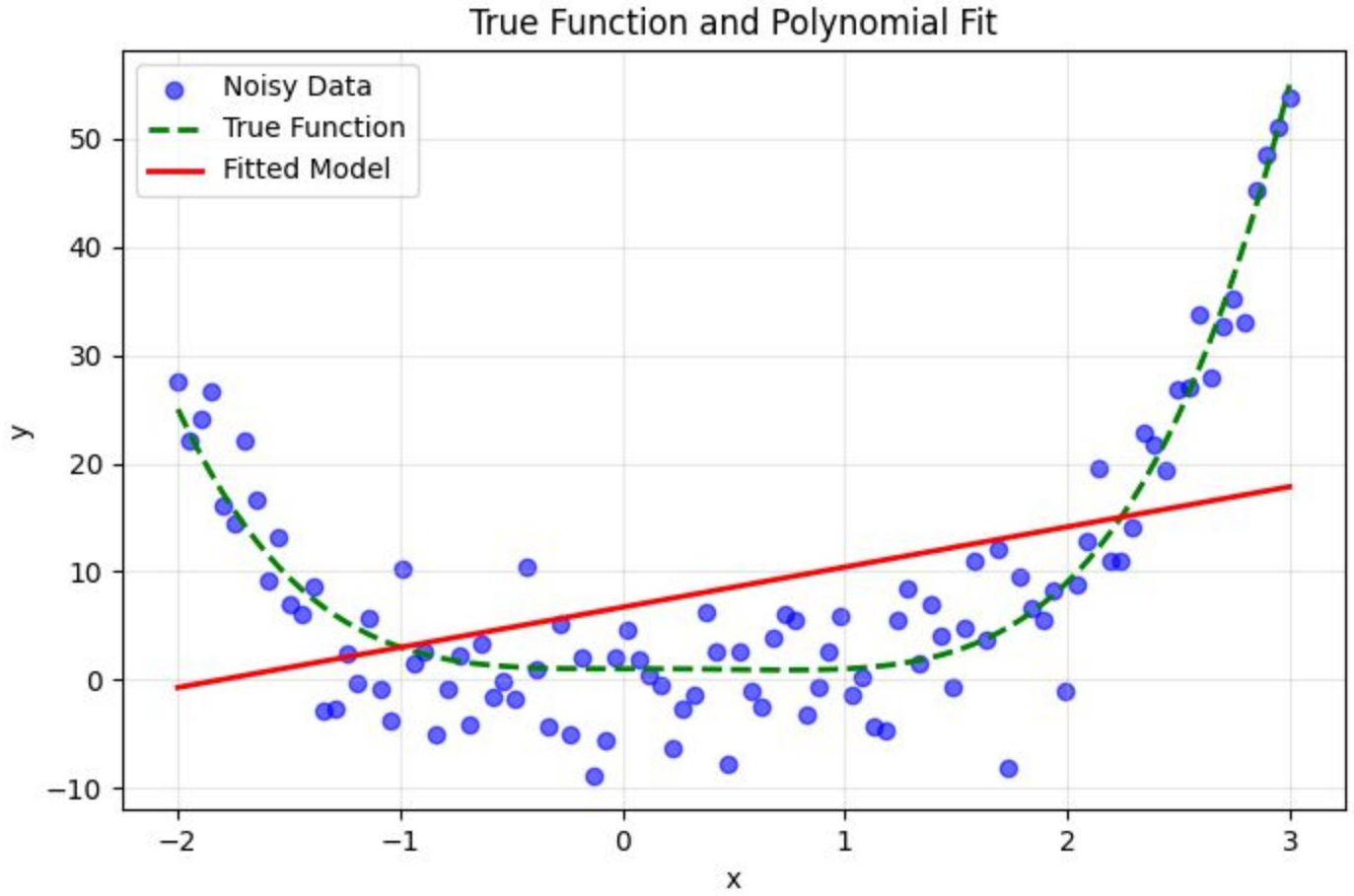
- **Residuals** are the differences between the actual (yi) and predicted (ŷi) values:

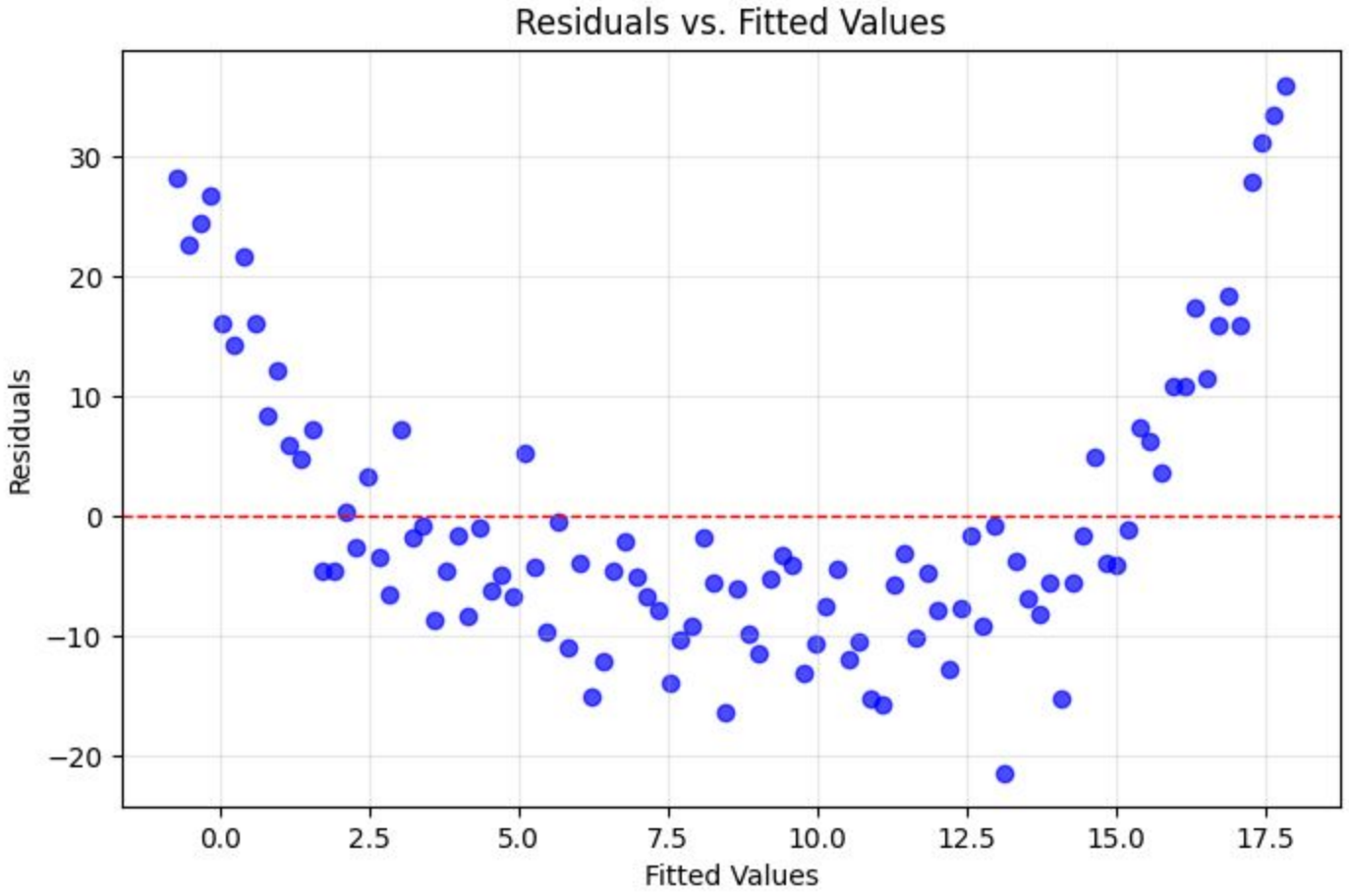
  Residual $= y_i - \hat{y}_i$

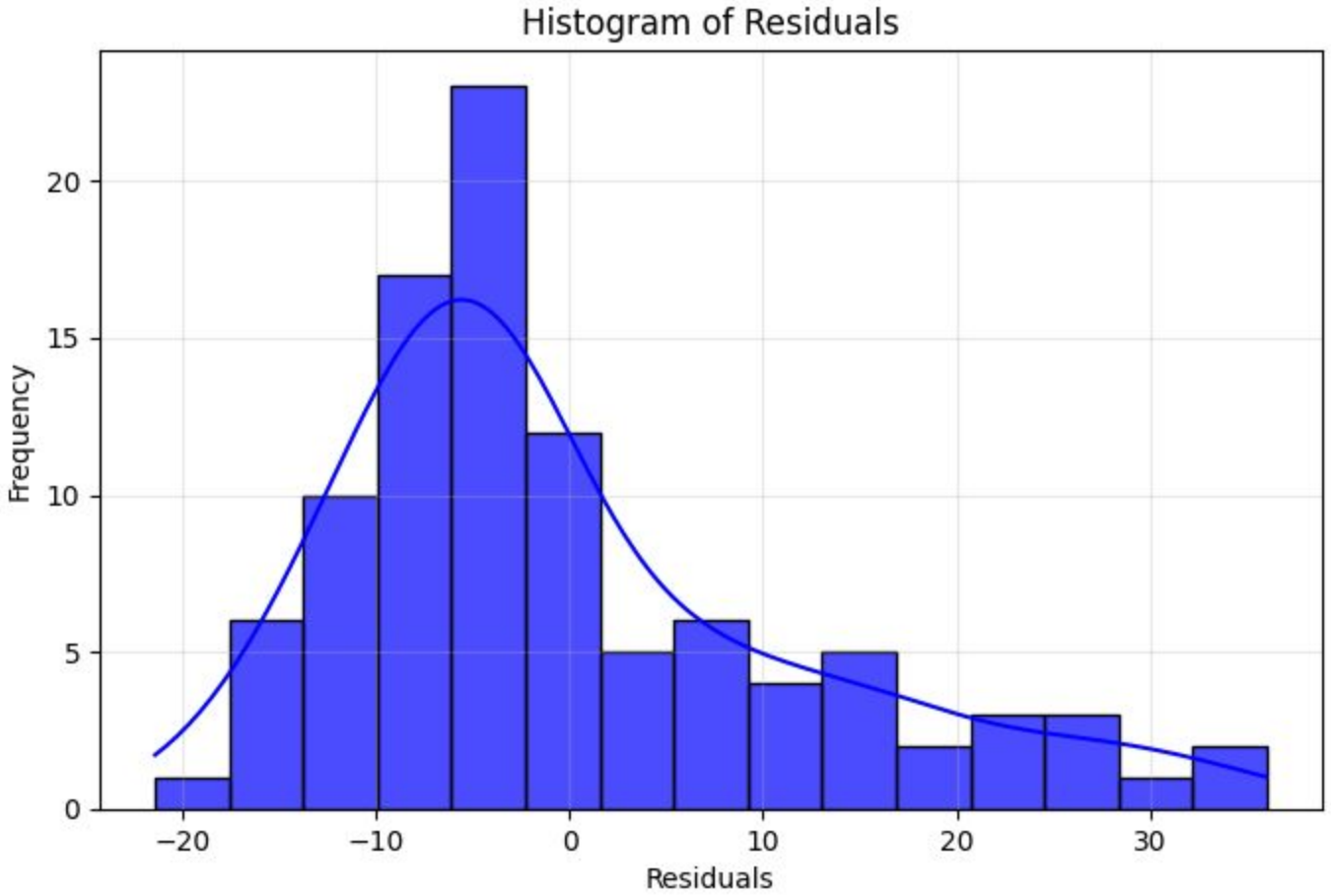
- Residual analysis involves examining the residuals for patterns to check the model's assumptions (e.g., linearity, homoscedasticity).
- **Key Plots**:
  - **Residuals vs. Fitted Values**: Should show no pattern; randomness indicates a good fit.
  - **Histogram or Q-Q Plot**: Checks if residuals are normally distributed.

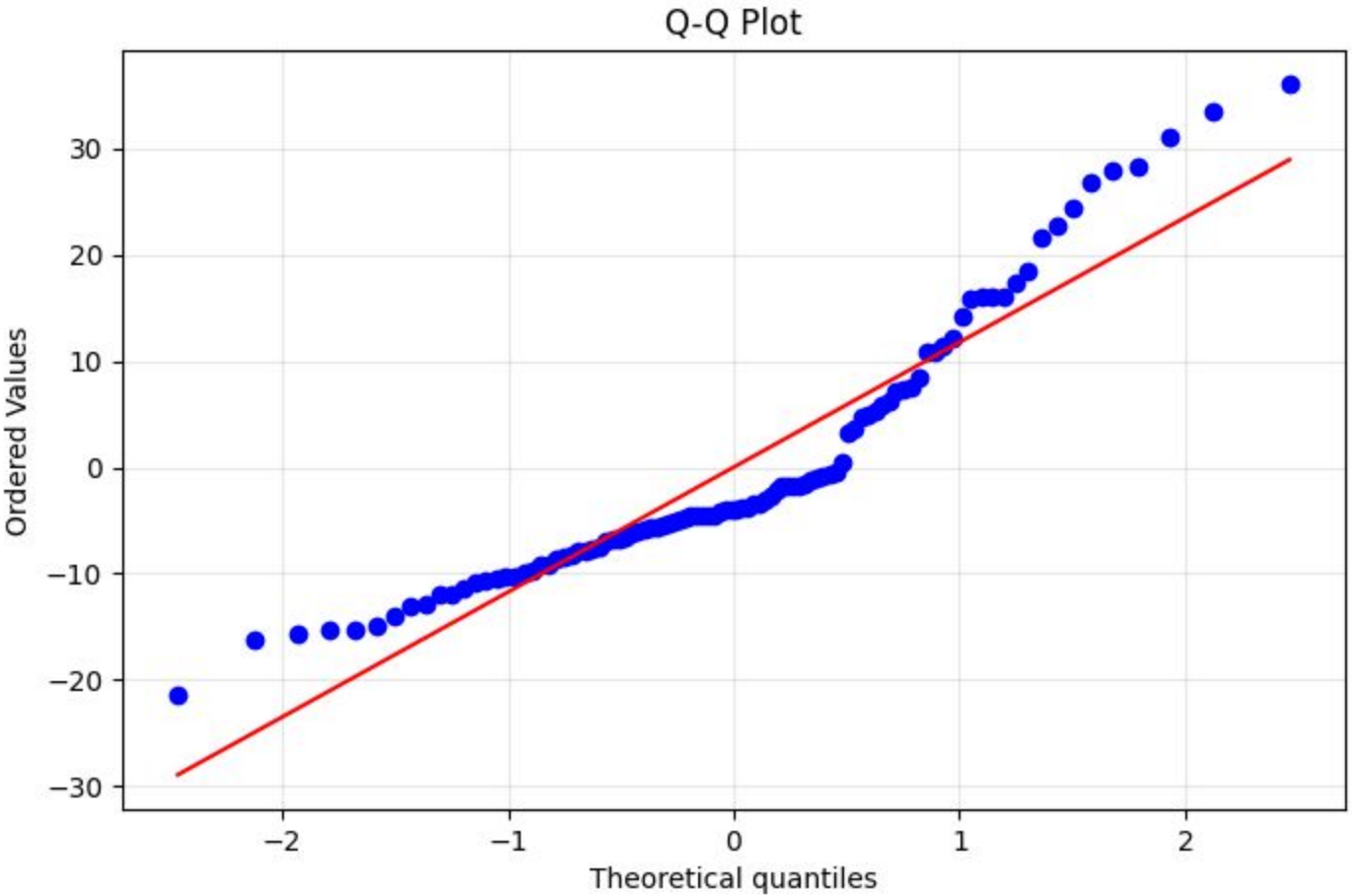
Plots

The Bad Fit

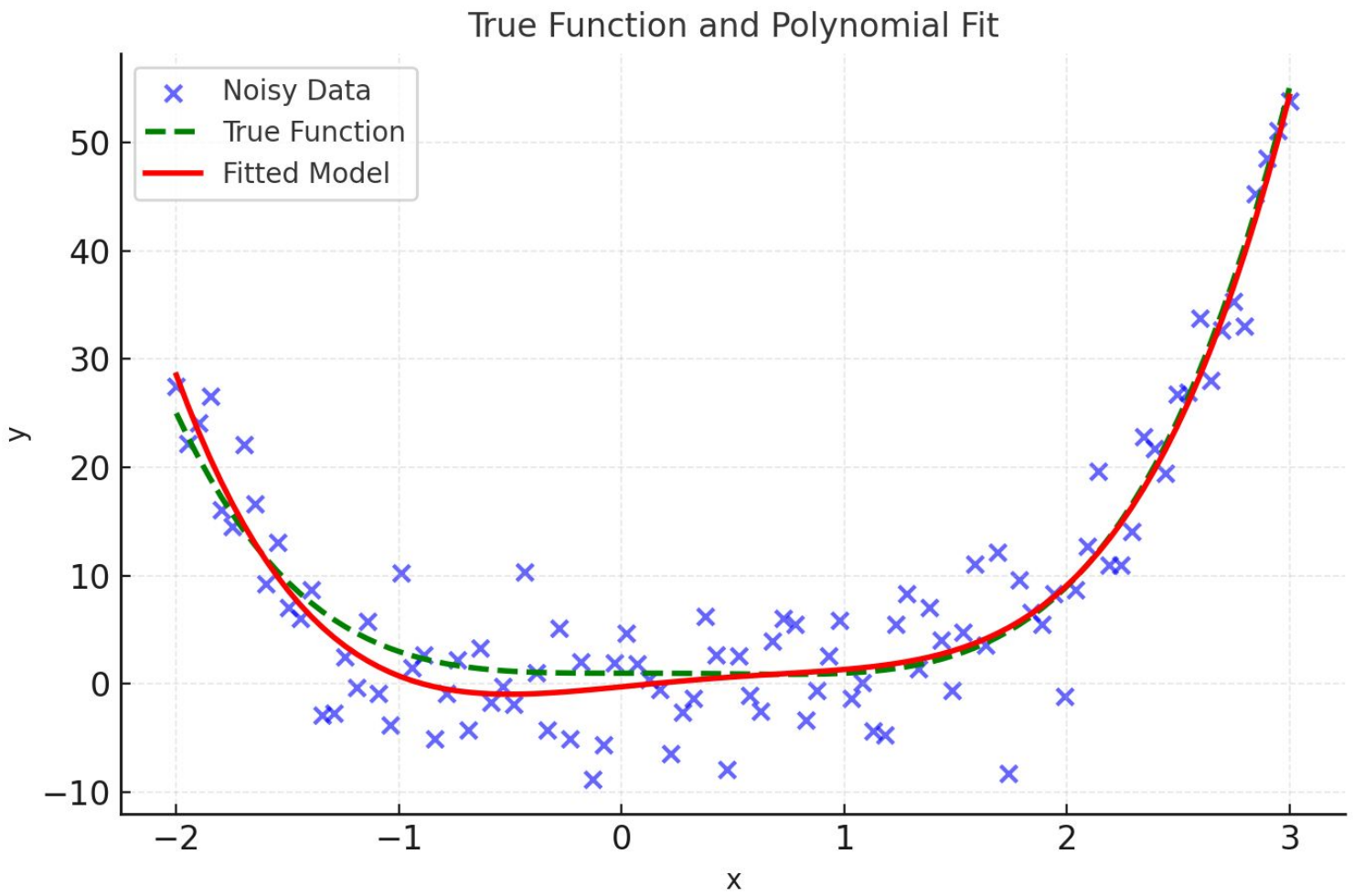

True Function and Polynomial Fit
Noisy Data
True Function
Fitted Model
50
40
30
20
10
0
−10
y
−2
−1
0
1
2
3
x

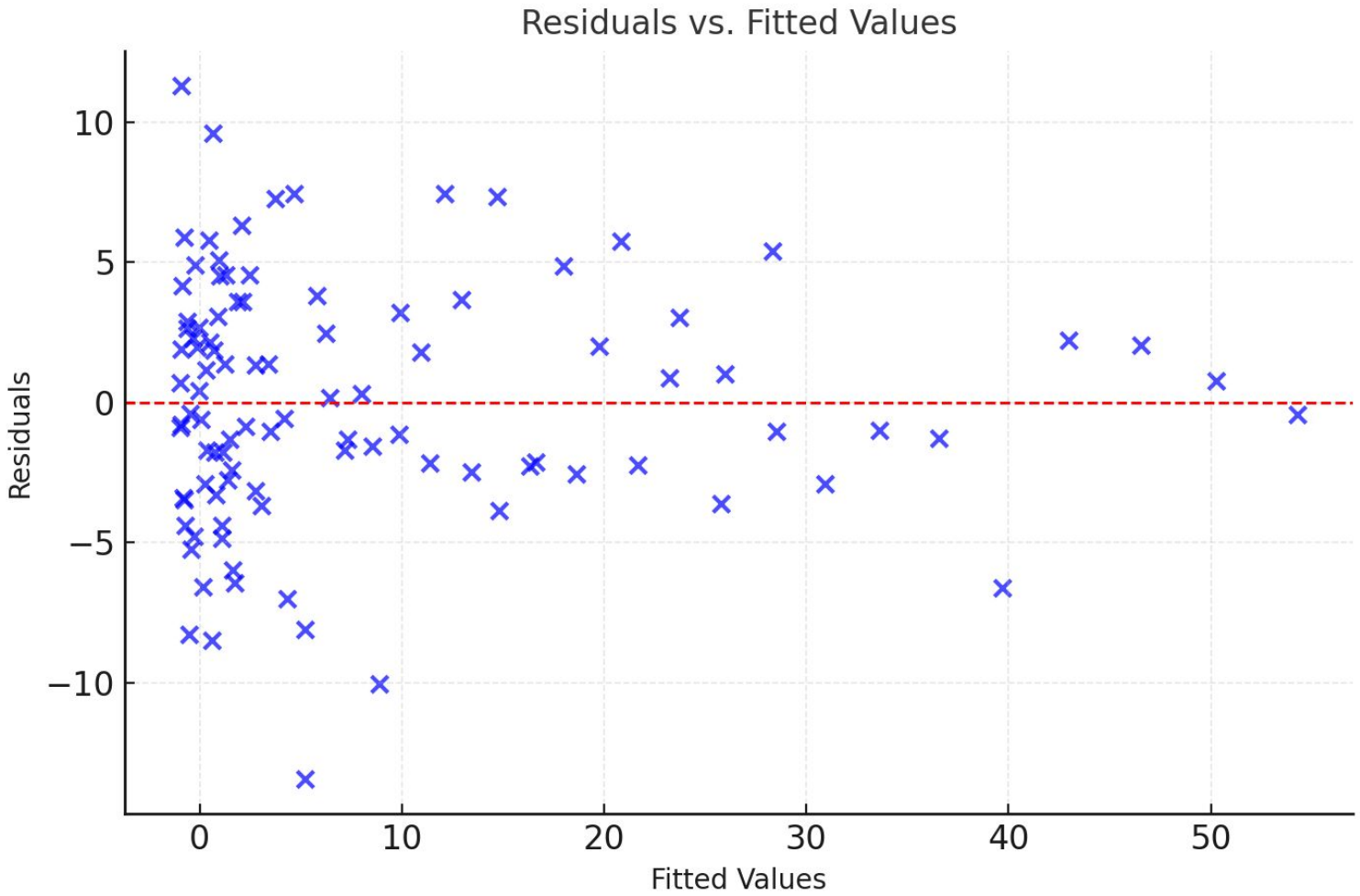

Residuals vs. Fitted Values
Residuals
Fitted Values
30
20
10
0
−10
−20
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

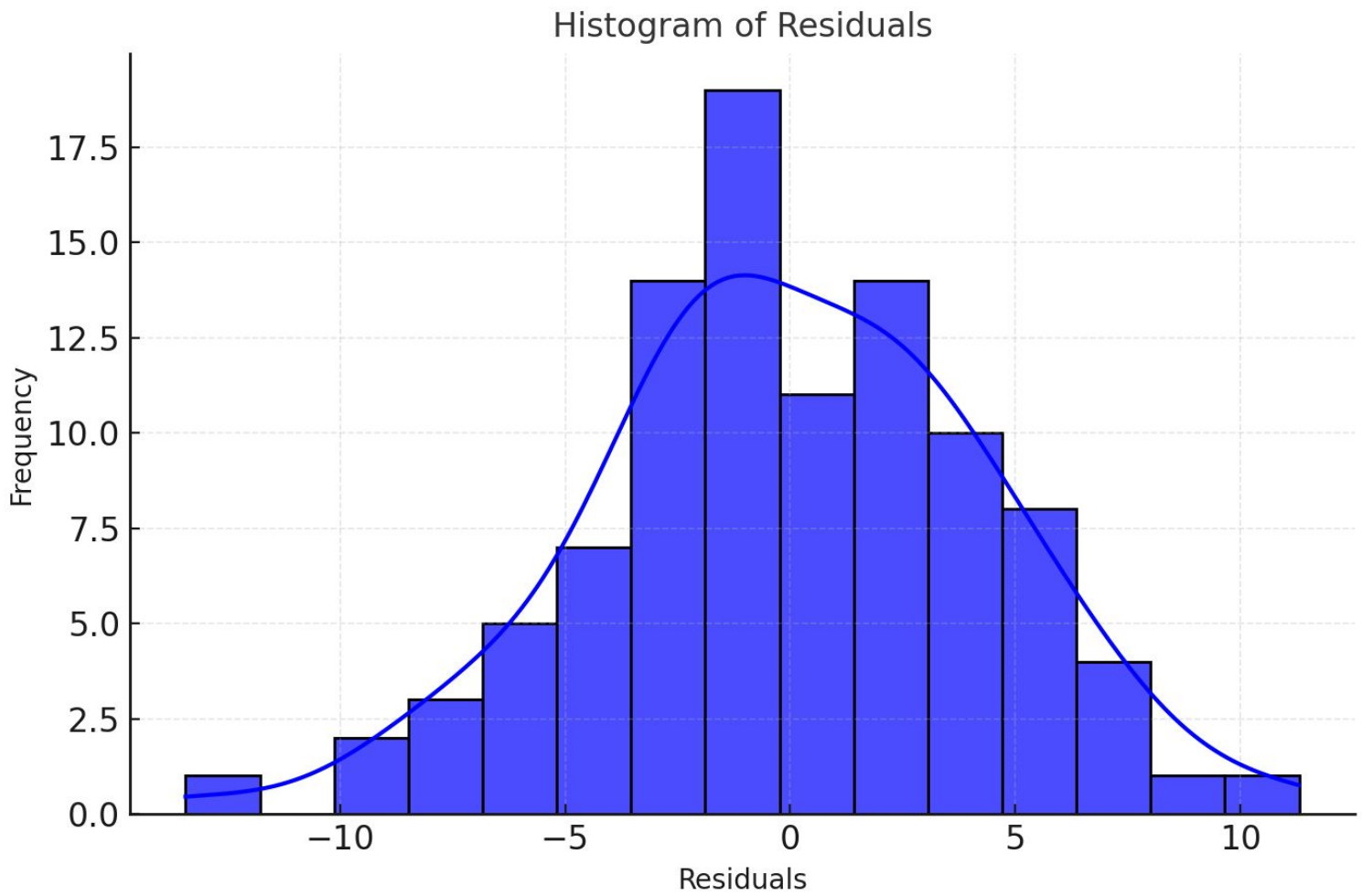

Histogram of Residuals
Frequency
Residuals
0
5
10
15
20
−20
−10
0
10
20
30

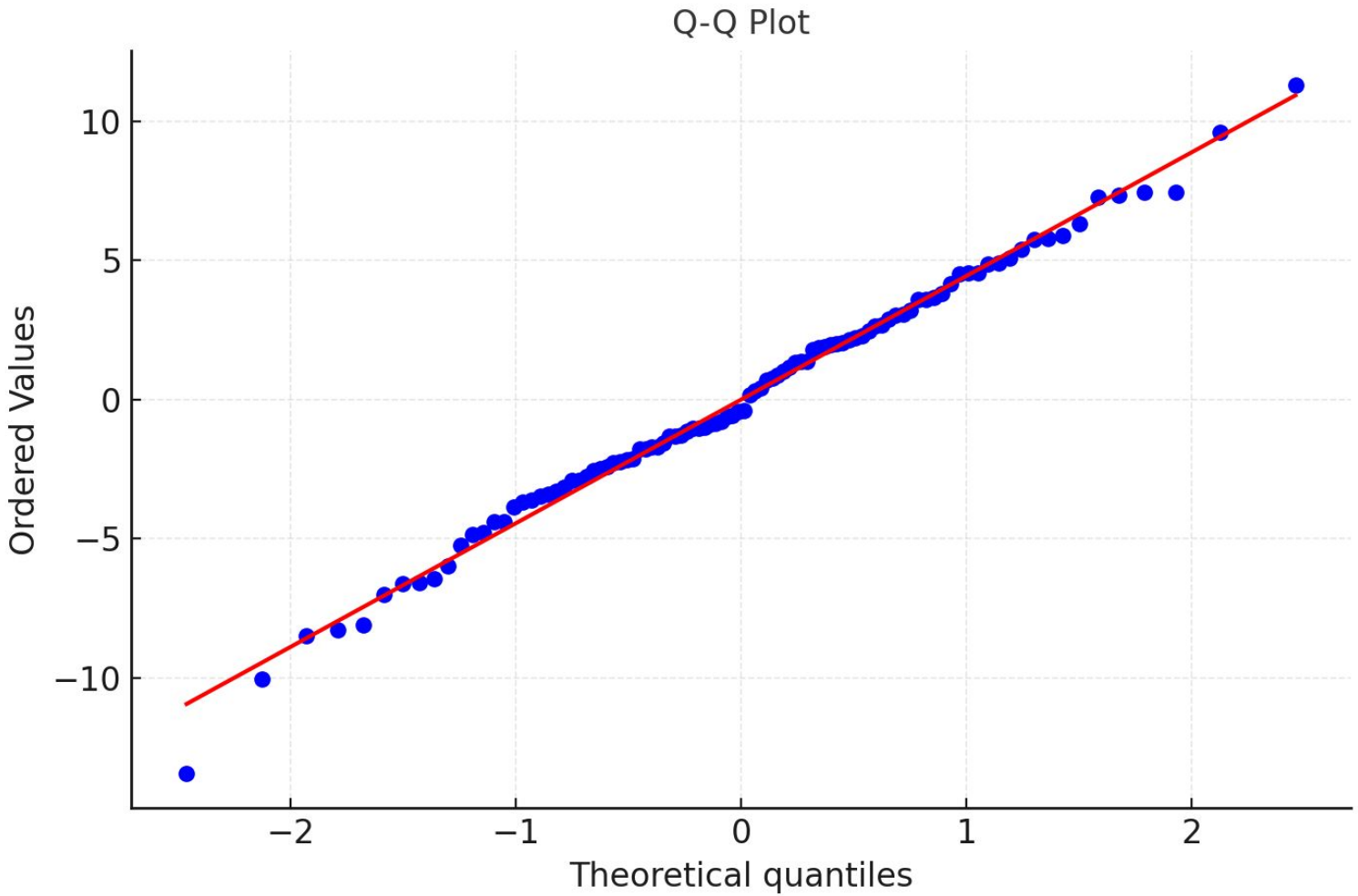

Q-Q Plot
Ordered Values
Theoretical quantiles
30
20
10
0
−10
−20
−30
−2
−1
0
1
2

The good fit


True Function and Polynomial Fit
Noisy Data
True Function
Fitted Model
50
40
30
20
10
0
−10
y
−2
−1
0
1
2
3
x


Residuals vs. Fitted Values
Residuals
Fitted Values
10
5
0
−5
−10
0
10
20
30
40
50


Histogram of Residuals
Frequency
Residuals
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
−10
−5
0
5
10


Q-Q Plot
Ordered Values
Theoretical quantiles
10
5
0
−5
−10
−2
−1
0
1
2

Mean Square Error (MSE)

- Measures the average squared difference between actual and predicted values:
- Sensitive to large errors due to squaring.

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

Root Mean Square Error

The square root of MSE, representing the error in the same units as the target variable:

RMSE= $\sqrt{(MSE)}$

Easier to understand and compare with the scale of the actual data.

Mean Absolute Error (MAE)

Measures the average absolute difference between actual and predicted values:

Less sensitive to outliers compared to MSE.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

R-Squared (R^2)

Explains the proportion of variance in the dependent variable that is predictable from the independent variables:

Range: $0 \leq R^2 \leq 1$; higher values indicate better fit.

$R^2=1$: Perfect fit, $R^2=0$: No predictive power.

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

Adjusted R^2

Adjusts R^2 for the number of predictors to penalize overfitting:

Better for comparing models with different numbers of features.

$$\text{Adjusted } R^2 = 1 - \left(1 - R^2\right) \frac{n-1}{n-p-1}$$

Where:

- n: Number of data points.
- p: Number of predictors.

When to use

- **Use Adjusted R^2** when you have multiple models with different numbers of predictors, and you want to compare them while accounting for the complexity of the model. It adjusts the R^2 value based on the number of predictors, making it more reliable than regular R^2 when comparing models with different numbers of predictors.
- It's useful when you want to assess how well the model fits the data and if adding additional predictors improves the model sufficiently (without overfitting).

Significance

- **Positive**: Adjusted R^2 will increase only when new predictors improve the model beyond what would be expected by chance.
- **Negative**: Adding irrelevant features or predictors will decrease the Adjusted R^2, discouraging overfitting.
- **Key Insight**: **Adjusted R^2** is good for **model comparison**, especially when you're adding more predictors and want to avoid overfitting.

Limitations

- It doesn’t account for the model’s complexity or penalize it directly, unlike AIC and BIC, which are more robust in preventing overfitting.

Akaike Information Criterion (AIC)

A measure of model quality that penalizes complexity

Lower AIC values indicate a better model fit while avoiding overfitting.

$$\mathrm{AIC} = 2k - 2\ln(L)$$

Where:

- k: Number of parameters.
- L: Maximum likelihood of the model.

When to use

Use AIC when you want to compare models, balancing between goodness-of-fit and model complexity. A lower AIC value suggests a better model.

It’s best used when comparing **non-nested models** (i.e., models that are not necessarily sub-models of each other).

It can be used in **forecasting models** and **time series** models as well.

Significance

Lower AIC values indicate better models. It seeks a good balance between fit and complexity by penalizing models with more parameters.

AIC is particularly useful for **model selection**, especially when you need to compare models that have different numbers of predictors.

Limitations

AIC tends to favor models that are too complex (with many parameters) since it only uses a small penalty for the number of parameters.

Bayesian Information Criterion (BIC)

Similar to AIC but with a stronger penalty for complexity:

$$\text{BIC} = k \ln(n) - 2 \ln(L)$$

Where:

- n: Number of data points.
- k: Number of parameters.

When to use

Use BIC when you want to avoid overfitting and favor more parsimonious models. BIC tends to **penalize complexity more heavily** than AIC, especially with large datasets.

It is best used for **nested models** and when **model selection** is done under the Bayesian framework.

Significance

Lower BIC values indicate better models. It is **more conservative** than AIC in selecting complex models, making it more likely to favor simpler models.

BIC is generally preferred when you have a large dataset and wish to avoid overfitting.

Limitations

BIC can **over-penalize model complexity** in smaller datasets and might prefer overly simple models, so it might discard some predictors that could still improve the model’s performance.

Logistic Regression

Logistic regression is a type of model of probabilistic statistical classification. It is used as a binary model to predict a binary response, the outcome of a categorical dependent variable (i.e., a class label), based on one or more variables.

Key Concepts

1. Sigmoid Function (logistic Function)

   Logistic regression uses the sigmoid function to map predicted values to probabilities between 0 and 1.

$$h_\theta(z) = \frac{1}{1 + e^{-z}}$$

$Z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n$
If $h_\theta(z) > 0.5$, classify as 1
If $h_\theta(z) \leq 0.5$, classify as 0

Key Concepts

2. Cost Function

   Instead of using Mean Square Error, logistic regression uses a log-loss function.

$$J(\theta) = \frac{-1}{m} \sum_{i=1}^{m} [y_i log(h_\theta(x_i)) + (1 - y_i) log(1 - h_\theta(x_i))]$$

   This function penalizes wrong predictions move heavily as probabilities move away from the true labels.

Key Concepts

3. Parameters (w0,w1, … ,wn) are optimized using Gradient Descent to minimize the cost function.

Example: Predicting Email Spam

Email Length	Contains “Free”	Spam(1) or Not (0)
150	0	0
100	0	0
50	0	0
25	0	0
115	1	1
140	1	1
200	0	1
400	0	1

Steps to Solve

1. **Feature Selection**

   Use "Email Length" and "Contains 'Free'" as features.

2. **Model**

   Fit the logistic regression model:

   z=w0+w1(Email Length)+w2(Contains ’Free’)

$$P(\text{Spam}) = \frac{1}{1 + e^{-z}}$$

Steps to Solve

3. Prediction

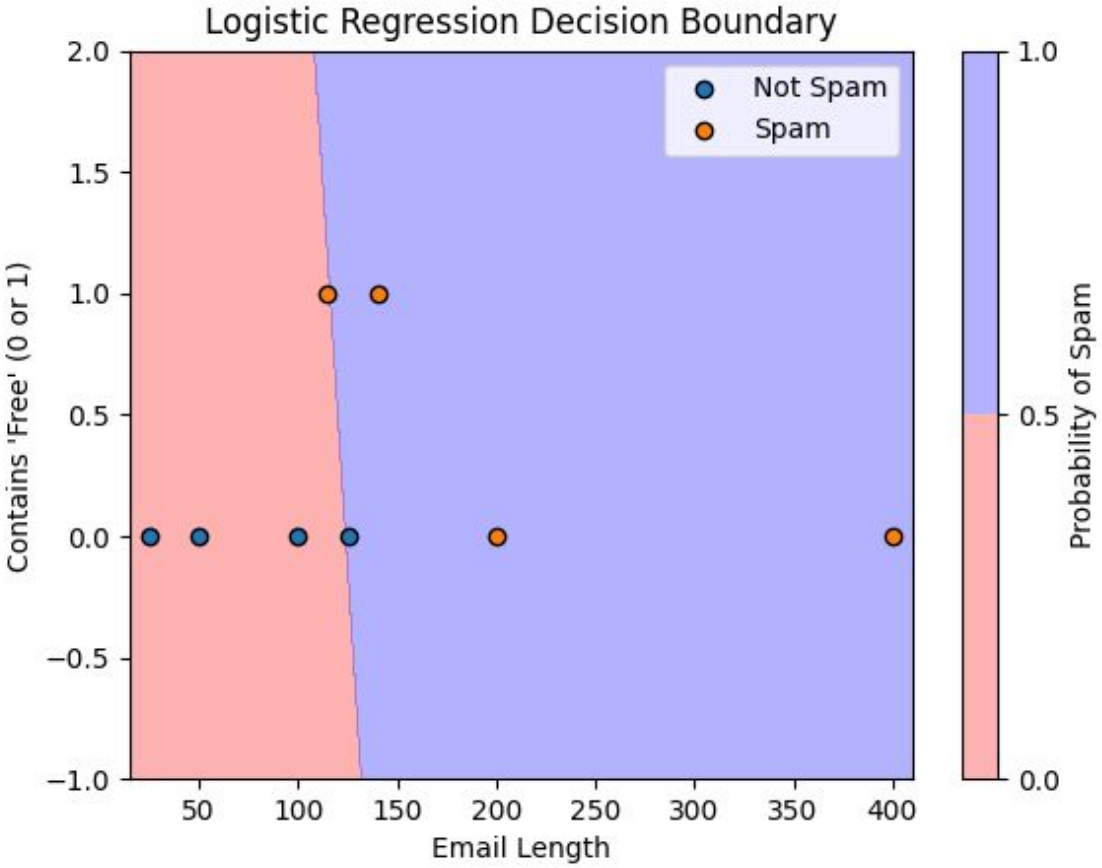
For a new email:

Length = 250, Contains "Free" = 1

$z=w0+w1(250)+w2(1)$

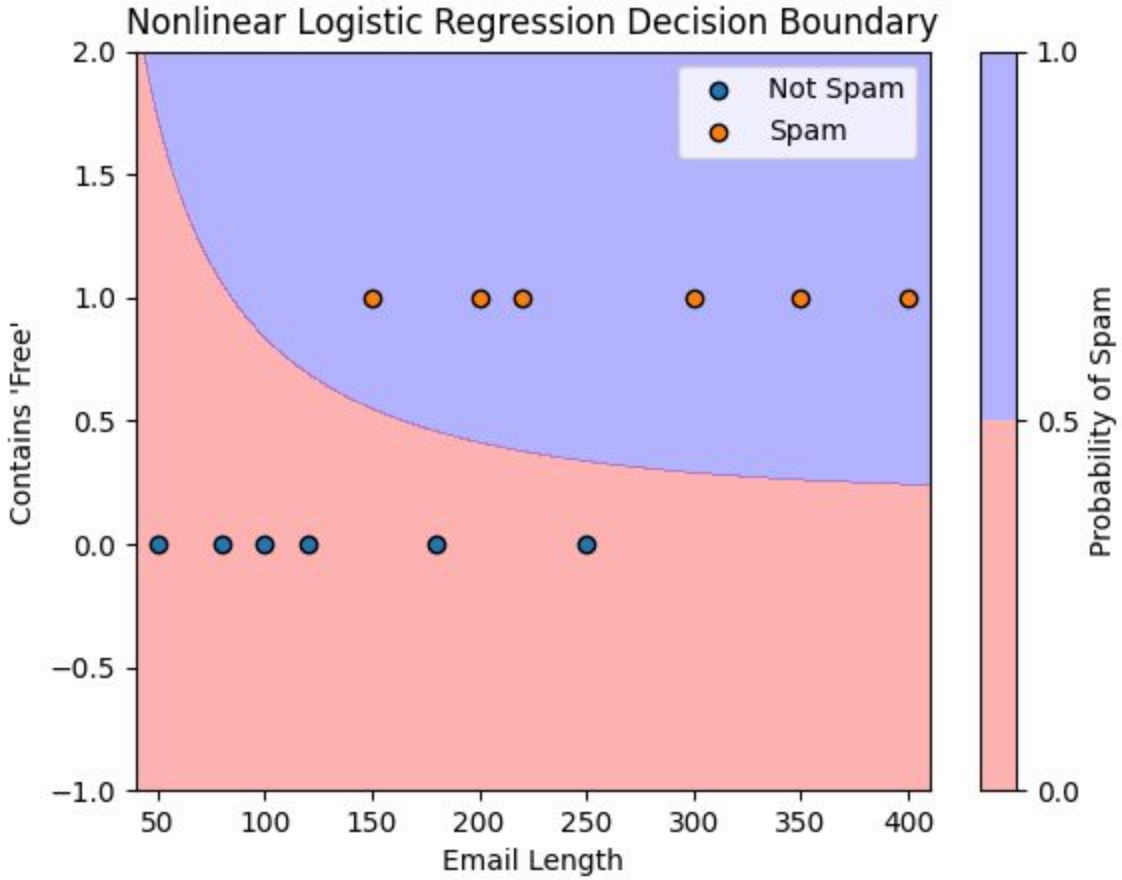
Compute P(Spam).

If P(Spam)>0.5P, classify as Spam (1).


Logistic Regression Decision Boundary
Not Spam
Spam
Contains 'Free' (0 or 1)
Email Length
Probability of Spam
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
50
100
150
200
250
300
350
400
1.0
0.5
0.0

Logistic Regression With Polynomial Boundary

Email Length (x1)	Contains 'Free' (x2)	Spam (y)
200	1	1
50	0	0
300	1	1
100	0	0
150	1	1
250	0	0
350	1	1
120	0	0
400	1	1
80	0	0
220	1	1
180	0	0


Nonlinear Logistic Regression Decision Boundary
Not Spam
Spam
Contains 'Free'
Email Length
Probability of Spam
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
50
100
150
200
250
300
350
400
1.0
0.5
0.0

Log Odds and Coefficient

Odd = wins/loss

In last 10 matches if Anil won 7 and lost 3. Odds of him winning is 7 to 3.

p(win) = number of matches won / total number of matches

p(loss) = number of matches lost / total number of matches

Odd = p(win)/p(loss)

Few examples

1. If MD wins 10 games and lose 2

   Odds of winning = 5
2. If MD wins 20 games and lose 2

   Odds of winning = 10
3. If MD lose 10 games and win 2

   odds of winning = 0.2
4. If MD lose 20 games and win 2

   odds of winning = 0.1

If no of wins > no of loss: value of odds can be very big

If no of loss > no of wins: value of odds just lies between 0-1.

To deal with this we use log of odd instead of odd.

Few examples

1. If MD wins 10 games and lose 2

   Log(Odds of winning) = 1.609
2. If MD wins 20 games and lose 2

   log(Odds of winning) = 2.30
3. If MD lose 10 games and win 2

   log(odds of winning) = -1.609
4. If MD lose 20 games and win 2

   log(odds of winning) = -2.3

Log odds in Logistic regression

$Z = \beta 0 + \beta 1x1 + \beta 2x2 + \beta 3x3 + \ldots + \beta nxn$

Z is the log odds

$$p(y = 1) = \frac{1}{1 + e^{-z}}$$

$$z = \frac{ln(p(y = 1))}{ln(1 - p(y = 1))}$$

Example

$Z = -1 + 0.5x1 + 0.3x2$

X1 = hours of studying

X2 = hours of attending classes

With coefficient of X1 = 0.5, for every additional hour of studying (x1) log of odds increase by 0.5 or odds increase by $e^{(0.5)} \approx 1.65$ (65% increase in odds).

With coefficient of X2 = 0.3, for every additional hour of studying (x2) log of odds increase by 0.3 or odds increase by $e^{(0.3)} \approx 1.35$ (35% increase in odds).

Example

Let x1 = 5 and x2 = 10

$Z = -1 + 0.5x5 + 0.3x10 = 4.5$

$p(y=1) = 1/(1+e^{(-4.5)}) \simeq 0.989$

Odds = 0.989/(1-0.989) = 89.9

Let x1 = 6 and x2 = 10

$Z = -1 + 0.5x6 + 0.3x10 = 5$

$p(y=1) = 1/(1+e^{(-5)}) \simeq 0.9933$

Odds = 0.9933/(1-0.9933) = 148.254 (ie 65% increase in odds)

Case Studies Using **Logistic Regression Models** in Various Domains

Predicting Diabetes

Model: Logistic Regression

Problem: Predict whether a patient is diabetic based on factors such as age, BMI, family history, glucose levels, etc.

Outcome: Healthcare providers can identify high-risk patients and intervene early to prevent or manage diabetes.

Data Example: A dataset might include variables like blood pressure, BMI, glucose levels, insulin, and age.

Cancer Diagnosis

Model: Logistic Regression

Problem: Classify a tumor as benign or malignant based on features like size, shape, and location of the tumor in medical imaging or biopsies.

Outcome: Healthcare professionals can make faster, more accurate diagnoses to begin treatment sooner.

Data Example: A dataset could include tumor size, shape, texture, and other imaging features.

Fraud Detection

Model: Logistic Regression

Problem: Detect fraudulent transactions by predicting whether a transaction is legitimate or fraudulent based on transaction features (amount, time, location, and user behavior).

Outcome: Helps financial institutions quickly detect and block fraudulent transactions, protecting users and institutions from financial loss.

Data Example: A dataset could include transaction amount, location, user ID, time, and merchant information.

Predicting Customer Churn

- **Model**: Logistic Regression
- **Problem**: Predict whether a customer will churn (leave the service) based on their usage patterns, demographic information, and customer service interactions.
- **Outcome**: Companies can proactively target at-risk customers with retention strategies such as personalized offers or loyalty programs.
- **Data Example**: A dataset might include customer tenure, usage frequency, customer satisfaction, and service complaints.

Predicting Purchase Likelihood

Model: Logistic Regression

Problem: Predict whether a customer is likely to make a purchase based on features like browsing history, product views, time spent on the website, and past buying behavior.

Outcome: Retailers and e-commerce companies can target high-probability customers with tailored offers and discounts to increase conversion rates.

Data Example: Data might include website clicks, product views, customer demographics, and past purchasing behavior.