Chapter #5

“Regression & Predictive Modeling”

5 Hours | 6 Marks

Compiled By
Er. Sushil Dyopala

Topics

1. Empirical models, simple linear regression, MLE and least square estimator
2. Multiple linear regression, matrix approach to multiple linear regression, polynomial regression models, categorical regressors, indicator variables, selection of variables and model building
3. Logistic regression

1.1 Empirical Models

- Empirical models are models derived from observed data rather than theoretical principles.
- They are built based on statistical relationships and patterns found in real-world data.
- Unlike mechanistic models, which are based on underlying laws (e.g., physics-based models), empirical models rely on data-driven patterns.
- Empirical modeling involves the development of models that explain, predict, or simulate a particular aspect of the world, rather than purely theoretical or abstract principles.

1.1 Empirical Models

- Example: House Price Prediction Using Linear Regression
- Non-Empirical Example: Newton's Law of Motion for Object Acceleration [F = m.a]
- Features of Empirical Models
  - Data-Driven: Depend entirely on collected data.
  - Approximate Nature: They do not necessarily explain the underlying cause but provide useful approximations.
  - Predictive Ability: Used mainly for forecasting and decision support.
  - Adaptability: Can be updated with new data to improve accuracy.

1.1 Types of Empirical Models

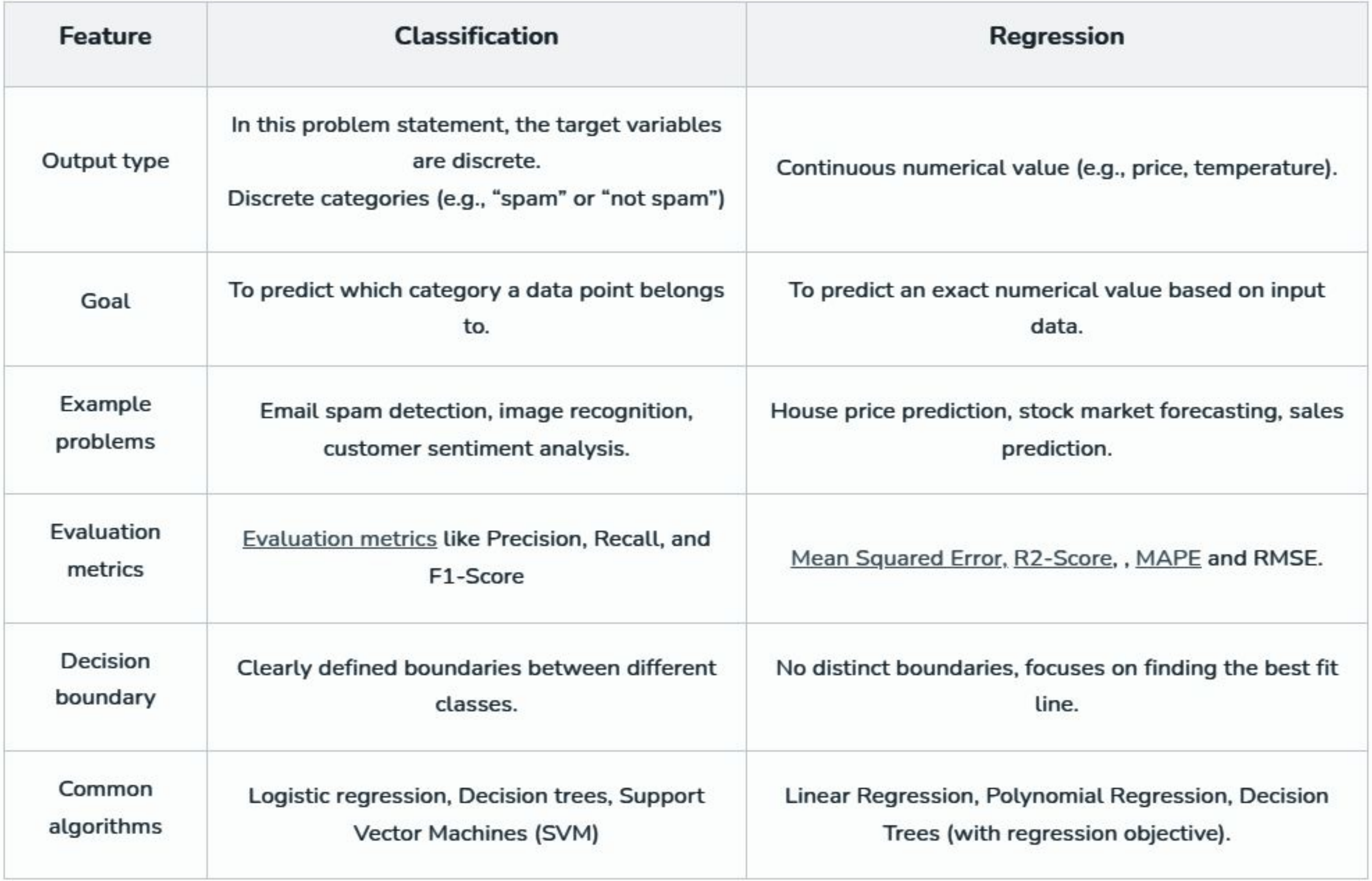
- Regression Models
  - Linear Regression
  - Polynomial Regression
  - Logistic Regression
- Time Series Models
  - Moving Average (MA)
  - Autoregressive (AR) Models: Uses previous values of the same variable to predict future values.
  - ARIMA (AutoRegressive Integrated Moving Average): Combines AR and MA for time series forecasting.
- Machine Learning-Based Empirical Models
  - Decision Trees & Random Forests
  - Support Vector Machines (SVM)
  - Neural Networks

1.2 Regression

- Regression is a supervised learning technique used for predicting continuous numerical values based on input features.
- It models the relationship between a dependent variable (target) and one or more independent variables (predictors).
- Examples:
  - Predicting house prices based on size, location, and amenities.
  - Estimating temperature based on historical weather data.
  - Forecasting sales for a retail business.
- Types:
  - Simple Linear Regression
  - Multiple Linear Regression
  - Polynomial Regression

1.2 Classification [Logistic Regression]

- Classification is a supervised learning technique used for predicting categorical values (classes or labels).
- Instead of predicting continuous values like regression, classification assigns data points to predefined categories.
- Examples:
  - Email spam detection (Spam or Not Spam).
  - Disease diagnosis (Diabetic or Non-Diabetic).
  - Sentiment analysis (Positive, Neutral, Negative).
- Types:
  - Binary Classification
  - Multi-Class Classification
  - Multi-Label Classification



Feature	Classification	Regression
Output type	In this problem statement, the target variables are discrete. Discrete categories (e.g., "spam" or "not spam")	Continuous numerical value (e.g., price, temperature).
Goal	To predict which category a data point belongs to.	To predict an exact numerical value based on input data.
Example problems	Email spam detection, image recognition, customer sentiment analysis.	House price prediction, stock market forecasting, sales prediction.
Evaluation metrics	Evaluation metrics like Precision, Recall, and F1-Score	Mean Squared Error, R2-Score, , MAPE and RMSE.
Decision boundary	Clearly defined boundaries between different classes.	No distinct boundaries, focuses on finding the best fit line.
Common algorithms	Logistic regression, Decision trees, Support Vector Machines (SVM)	Linear Regression, Polynomial Regression, Decision Trees (with regression objective).

https://www.geeksforgeeks.org/ml-classification-vs-regression/

1.3 Simple Linear Regression

- Simple Linear Regression is a statistical method used to model the relationship between one independent variable (predictor, X) and one dependent variable (target, Y) by fitting a straight line to the data.
- It is also called Univariate Linear Regression.
- Simple Linear Regression equation:

$$y = \theta_0 + \theta_1 x$$

where,

- y is dependent variable (target/output)
- x is independent variable (input/predictor)
- θ_0 is the intercept
- θ_1 is the slope of regression line

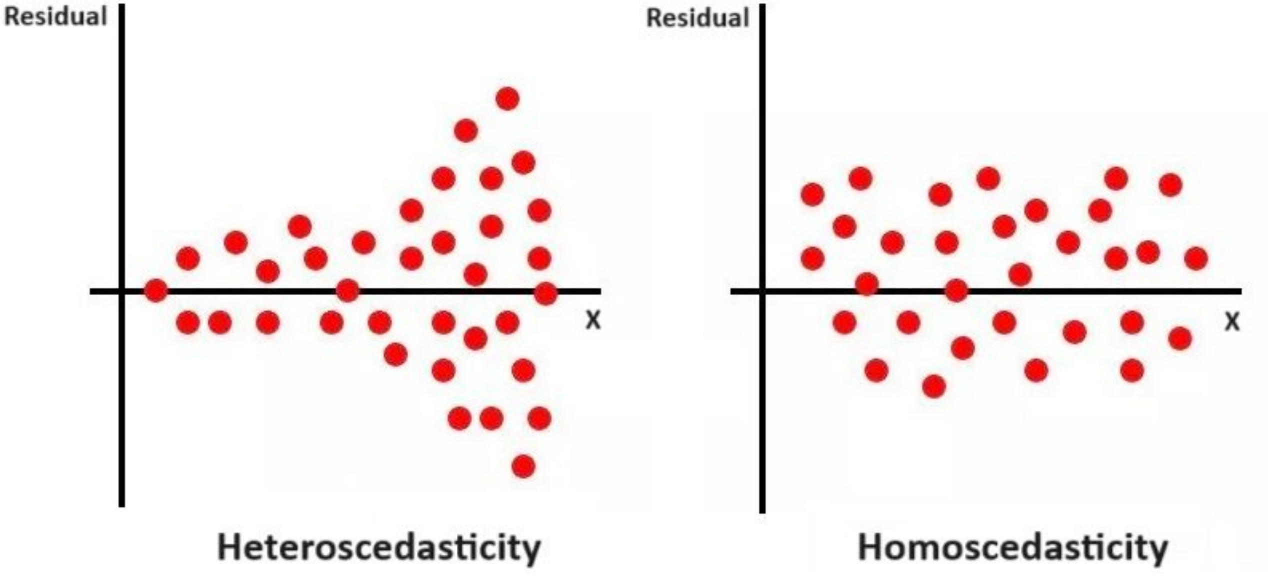
1.3 Simple Linear Regression: Derivation & Numerical

- Will be done in class using Least Square Estimates

1.3 Simple Linear Regression: Assumptions

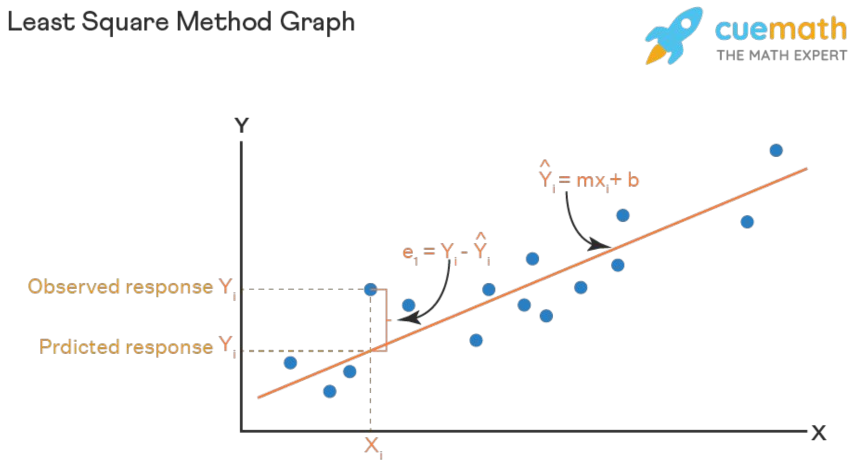
- Linearity: The relationship between X and Y should be linear.
- Independence: The observations should be independent from each other that is the errors from one observation should not influence other.
- Homoscedasticity: Across all levels of the independent variable(s), the variance of the errors is constant. This indicates that the amount of the independent variable(s) has no impact on the variance of the errors. If the variance of the residuals is not constant, then linear regression will not be an accurate model.
- Normality of Residuals: The residuals should be normally distributed.

1.3 Simple Linear Regression: Assumptions



1.4 Least Square Estimates

- Least Squares Estimation (LSE) is a method used to estimate parameters in regression by minimizing the sum of squared errors (residuals) between observed and predicted values.
- Derivation we have performed to find the best fit parameters for simple linear regression is based on least square estimates.



1.4 Likelihood

- The likelihood function is a fundamental concept in statistics and machine learning **used to estimate parameters of a probability distribution based on observed data.**
- It measures how likely a set of parameters is, given the observed data.

1.4 Likelihood Vs. Probability

- When Probability has to be calculated of any situation using dataset, then the dataset features will be constant i.e. mean & standard deviation of the dataset will be constant, they will not be altered.
- Let's say the probability of height > 170 cm has to be calculated for a random record in the dataset, then that will be calculated using the information shown below:

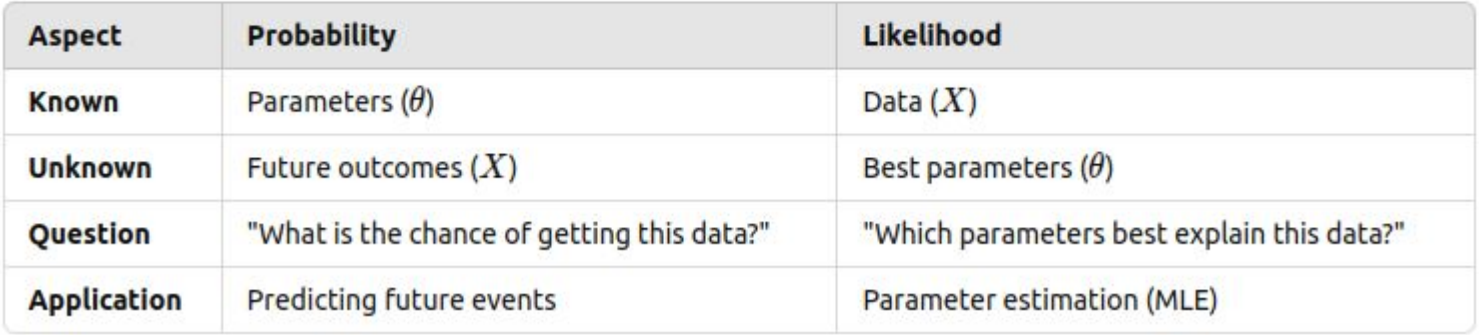
$$P(height > 170cm|\mu = 170, \sigma = 3.5)$$

- Likelihood calculation involves calculating the best distribution of data, given a particular feature value or situation.
- Consider the exactly same dataset example as provided above for probability, if their likelihood of height > 170 cm has to be calculated then it will be done using the information shown below:

$$Likelihood(\mu = 170, \sigma = 3.5|height > 170cm)$$

1.4 Likelihood Vs. Probability

- Probability is used to finding the chance of occurrence of a particular situation, whereas Likelihood is used to generally maximizing the chances of a particular situation to occur.



Aspect	Probability	Likelihood
Known	Parameters (θ)	Data (X)
Unknown	Future outcomes (X)	Best parameters (θ)
Question	"What is the chance of getting this data?"	"Which parameters best explain this data?"
Application	Predicting future events	Parameter estimation (MLE)

1.4 Maximum Likelihood Estimation (MLE)

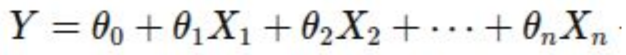
- Maximum Likelihood Estimation (MLE) is a statistical method used to estimate the parameters of a probabilistic model by maximizing the likelihood function.
- The goal is to find the parameter values that make the observed data most probable under the assumed statistical model.

1.4 MLE for Fitting Linear Regression

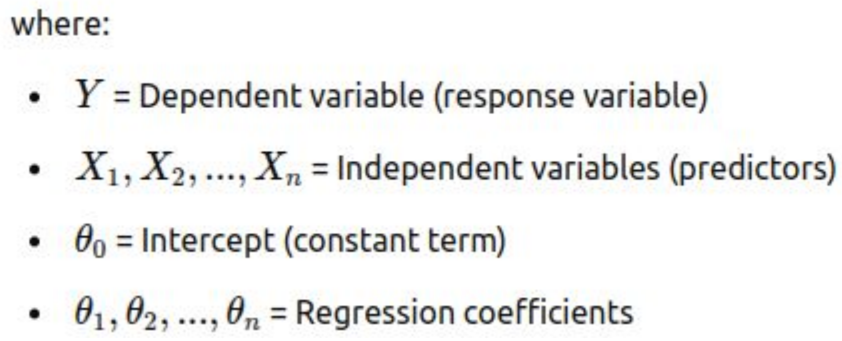
- Will be done in class.

2.1 Multiple Linear Regression (MLR)

- In Multiple Linear Regression (MLR), we extend simple linear regression by considering multiple independent variables (predictors) to explain the variation in a dependent variable.
- MLR is a statistical method that models the relationship between a dependent variable Y and multiple independent variables X1,X2,…,Xn using the equation:



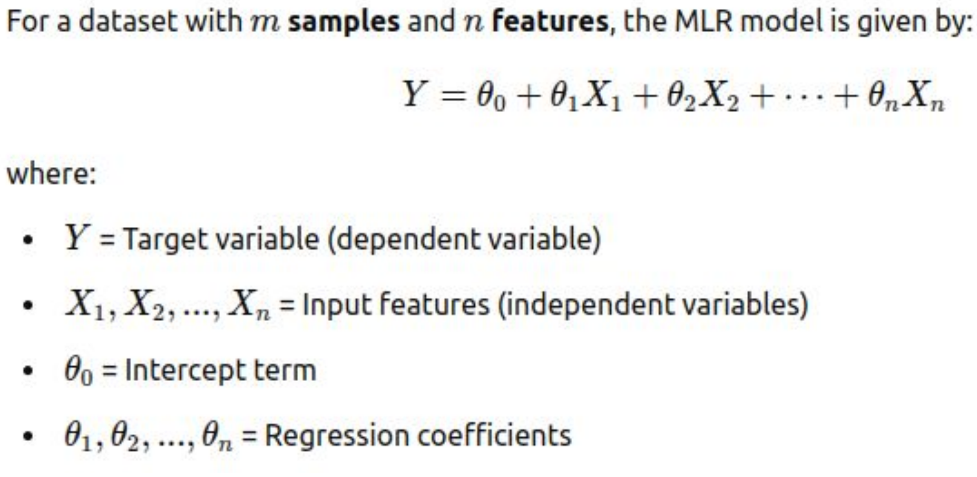
$$Y = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \cdots + \theta_n X_n$$



where:

- Y = Dependent variable (response variable)
- $X_1, X_2, ..., X_n$ = Independent variables (predictors)
- θ_0 = Intercept (constant term)
- $\theta_1, \theta_2, ..., \theta_n$ = Regression coefficients

2.1 Matrix Notation for MLR

- 

  For a dataset with m **samples** and n **features**, the MLR model is given by:

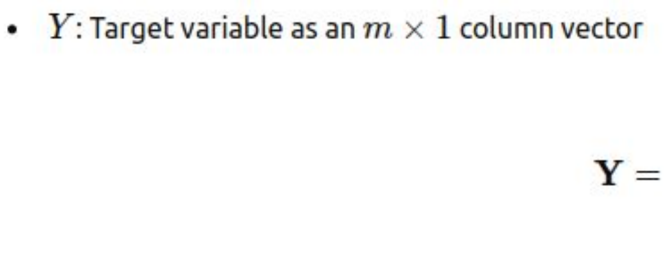
  $$Y = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \cdots + \theta_n X_n$$

  where:

  - Y = Target variable (dependent variable)
  - $X_1, X_2, \ldots, X_n$ = Input features (independent variables)
  - θ_0 = Intercept term
  - $\theta_1, \theta_2, \ldots, \theta_n$ = Regression coefficients

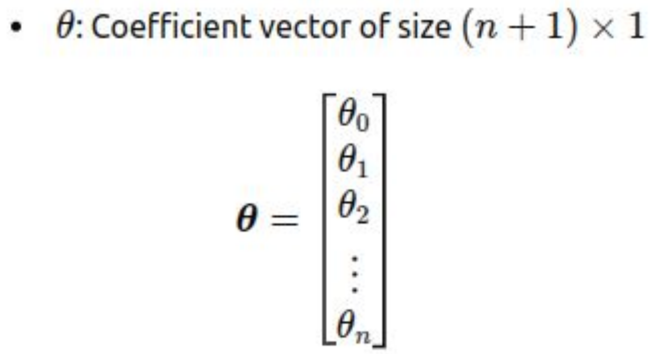
- Instead of writing separate equations for each sample, we represent all samples in a matrix form for computational efficiency.

2.1 Matrix Notation for MLR:Defining Matrices



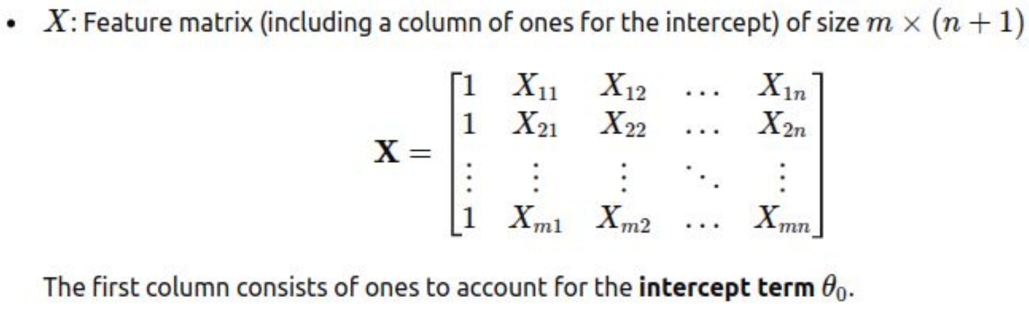
- Y: Target variable as an $m \times 1$ column vector

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_m \end{bmatrix}$$



- θ: Coefficient vector of size $(n+1) \times 1$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$



- X: Feature matrix (including a column of ones for the intercept) of size $m \times (n+1)$

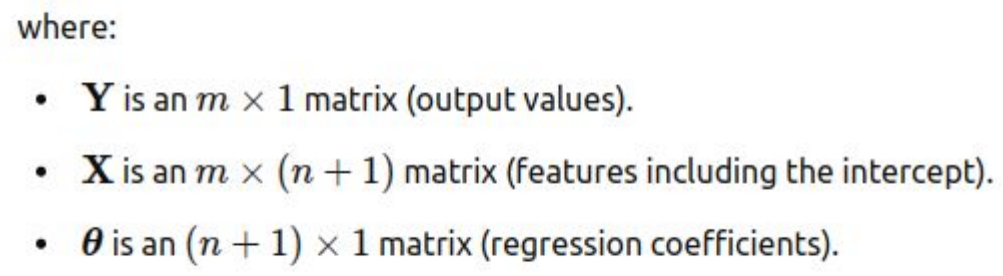
$$\mathbf{X} = \begin{bmatrix} 1 & X_{11} & X_{12} & \dots & X_{1n} \\ 1 & X_{21} & X_{22} & \dots & X_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{m1} & X_{m2} & \dots & X_{mn} \end{bmatrix}$$

The first column consists of ones to account for the **intercept term** θ_0.

2.1 Matrix Representation of MLR Equation

- MLR equation in matrix form

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\theta}$$



where:

- $\mathbf{Y}$ is an $m \times 1$ matrix (output values).
- $\mathbf{X}$ is an $m \times (n+1)$ matrix (features including the intercept).
- $\boldsymbol{\theta}$ is an $(n+1) \times 1$ matrix (regression coefficients).

- Derivation of Regression Coefficients for MLR will be done in class.

2.1 Advantages of the Matrix Approach: Vectorization

- The matrix approach allows us to perform operations on the entire dataset at once, rather than iterating over individual samples.
- The matrix approach eliminates explicit loops by leveraging built-in matrix operations, reducing execution time.
- The gradient of the cost function is computed efficiently in matrix form.
- When dealing with big data, matrix operations can be parallelized using optimized numerical computing libraries.

2.1 Assumptions of MLR

- Linearity:
  - The relationship between predictors and the target variable should be linear.
- Independence of Errors:
  - The residuals should not be correlated.
- Homoscedasticity (Constant Variance of Errors):
  - The variance of residuals should remain constant.
- No Multicollinearity:
  - Predictors should not be highly correlated with each other.
- Normality of Residuals:
  - Residuals should follow a normal distribution.

2.2 Polynomial Regression

- Linear regression models assume a linear relationship between the independent variables (features) and the dependent variable (target).
- However, many real-world problems exhibit non-linear relationships, requiring an extension of linear regression.
- One way to handle this is by transforming features to capture non-linearity while still using the linear regression framework i.e. using polynomial regression.

2.2 Polynomial Regression

- A polynomial model extends linear regression by including higher-order terms of the features.
- The key idea is to transform the input feature x into polynomial features x^2,x^3,... and then apply linear regression.
- A polynomial regression model of degree d can be written as:

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + ... + \theta_d x^d$$

- This model remains a linear regression model in terms of its parameters θ, but the features are non-linearly transformed.

2.2 Matrix Notation for Polynomial Regression

- If we have m training samples and a single feature x, we transform x into a polynomial feature matrix:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & \dots & x_1^d \\ 1 & x_2 & x_2^2 & x_2^3 & \dots & x_2^d \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & x_m^3 & \dots & x_m^d \end{bmatrix}$$

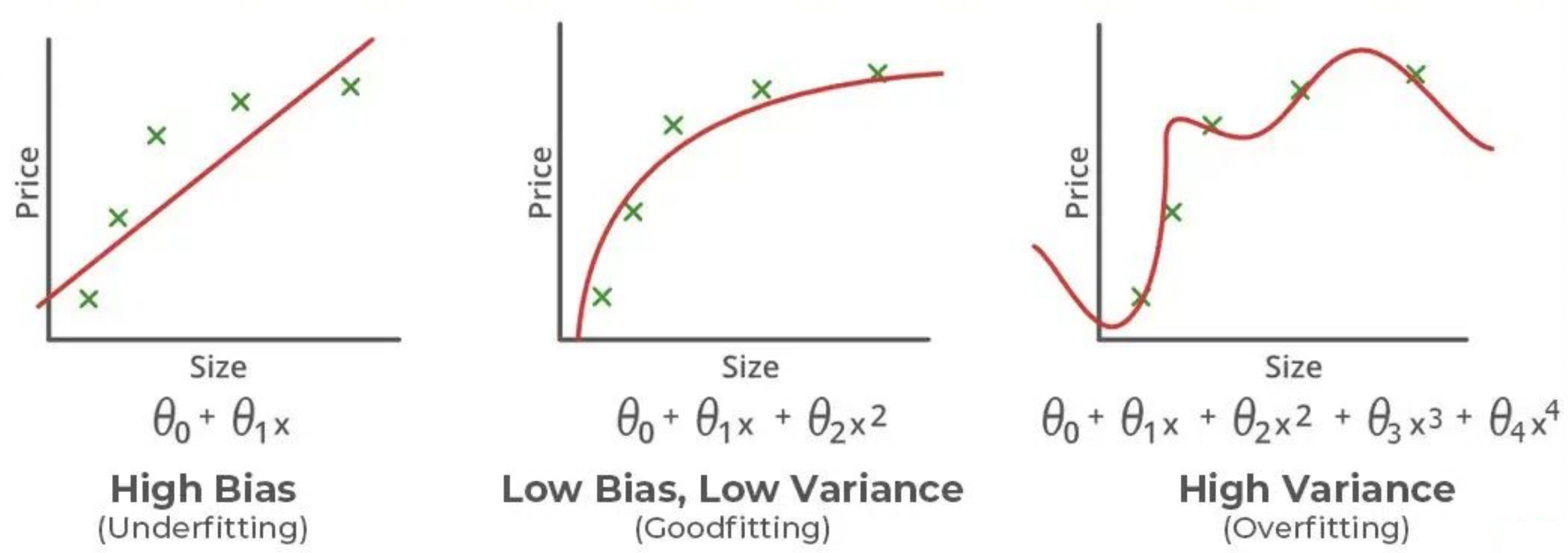
- The predicted values are computed using: **Y = Xθ**
- The parameters of polynomial regression i.e. thetas are computed using normal equations just like in MLR as:

$$\mathbf{\Theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

2.2 Bias-Variance Tradeoff in Polynomial Regression

- What is Underfitting?
  - Underfitting is the condition when the model performs poorly in both training data and validation data.
  - It occurs when the complexity of the model is very low [i.e. degree of polynomial is very low]. As a result the model fails to capture the pattern in the data.
  - It is characterized by high bias and low variance.
- What is Overfitting?
  - Overfitting is the condition when the model performs well in training data but performs poor in validation data.
  - It occurs when model complexity is very high [i.e. degree of polynomial is very high]. As a result the model memorizes the training data rather than generalizing it.
  - It is characterized by low bias and high variance.
- Good Fitting
  - Good balance
  - Low bias and low variance

2.2 Avoiding Overfitting in Polynomial Regression



$\theta_0 + \theta_1 x$

High Bias
(Underfitting)

$\theta_0 + \theta_1 x + \theta_2 x^2$

Low Bias, Low Variance
(Goodfitting)

$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High Variance
(Overfitting)

2.2 Avoiding Overfitting in Polynomial Regression

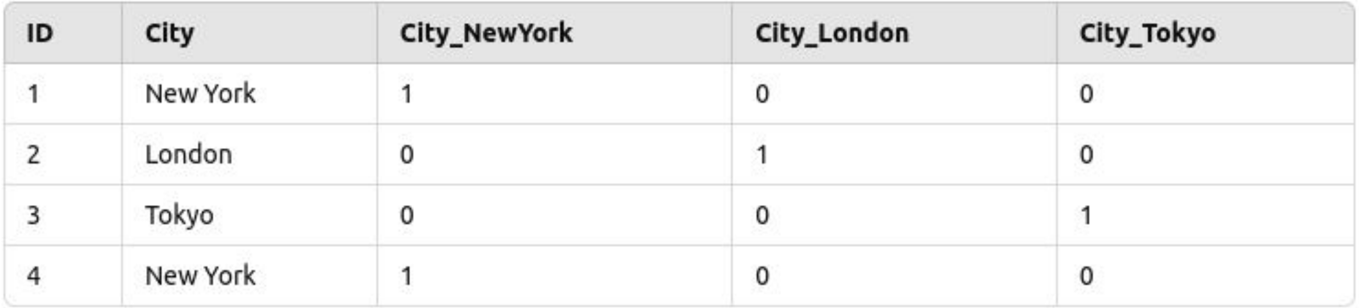
- A low-degree polynomial may underfit, while a high-degree polynomial may overfit.
- As we increase the polynomial degree, the model becomes more flexible and model will be able to capture complex patterns, but it can also lead to overfitting.
- Strategies to Prevent Overfitting
  - Choosing the Right Degree
    - Use cross-validation to find the best polynomial degree.
  - Use Regularization
    - Regularization is the techniques to prevent overfitting that eliminates or penalizes the unnecessary features.
    - Ridge regression (L2 regularization) adds a penalty on large coefficients, reducing overfitting.
    - Lasso regression (L1 regularization) eliminates unnecessary features.

2.3 Categorical Regressors

- In regression models, categorical variables are those that represent discrete groups or categories (e.g., color, gender, city, occupation).
- Unlike numerical variables, categorical variables cannot be directly used in mathematical operations.
- Categorical variables must be converted into numerical form using encoding techniques such as dummy variables (indicator variables), one hot encoding, label encoding.
- One hot encoding and label encoding techniques are covered in previous chapters.

2.3 Categorical Regressors:Use of Indicator (Dummy) Variables

- Indicator/Dummy variables are binary (0 or 1) variables used to represent categorical values. The concept of dummy or indicator variables are used in one hot encodings and label encodings.
- They allow categorical features to be included in regression models.
- Example: For the "City" variable with three unique values (New York, London, Tokyo), we create dummy variables:



ID	City	City_NewYork	City_London	City_Tokyo
1	New York	1	0	0
2	London	0	1	0
3	Tokyo	0	0	1
4	New York	1	0	0

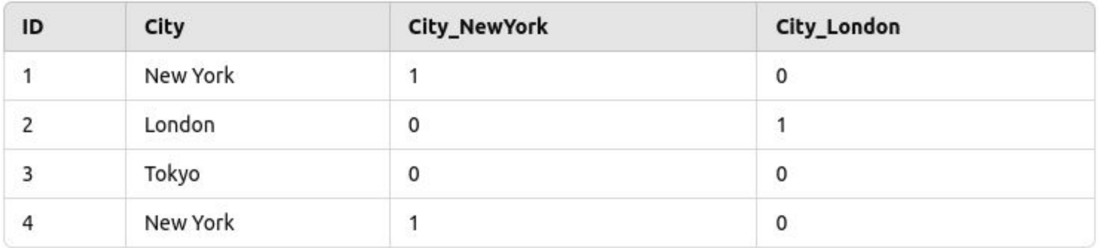
- Instead of using "City," we use the three new binary columns.

2.3 Dummy Variable Trap

- The dummy variable trap occurs when all dummy variables are used in the regression model, leading to multicollinearity (high correlation among independent variables).
- This makes matrix inversion unstable in calculations.
- To avoid this, we drop one dummy variable (called the reference category).

2.3 Dummy Variable Trap:Solution

- To avoid dummy variable trap, we drop one dummy variable (called the reference category).
- In the previous example of encoding city, let’s take “City_Tokyo” as reference category.
- After removing the reference category, the table will look like this:



ID	City	City_NewYork	City_London
1	New York	1	0
2	London	0	1
3	Tokyo	0	0
4	New York	1	0

- Interpretation:
  - If the city is New York, then City_NewYork = 1 and City_London = 0.
  - If the city is London, then City_NewYork = 0 and City_London = 1.
  - If the city is Tokyo (reference category), then both City_NewYork = 0 and City_London = 0.

2.4 Selection of Variables/Features

- Feature selection is a technique used to select the most relevant predictors (independent variables) in a regression model.
- It helps improve model accuracy, interpretability, and computational efficiency by removing redundant or irrelevant features.
- Feature Selection Techniques:
  - Filter Methods
  - Wrapper Methods
  - Embedded Methods
- All of these techniques are covered in chapter 4 but we will revise Wrapper Methods.

2.4 Selection of Variables/Features: Wrapper Methods

1. Forward Selection
   - Start with an empty model and add features one by one based on their significance.
   - Steps:
     1. Start with no predictors in the model.
     2. Add the most significant feature (lowest p-value in regression).
     3. Keep adding features one at a time until adding a new feature does not significantly improve the model.
   - Best when you have a large number of features, and you want to find the most important ones.

2. Backward Elimination
   - Start with all features and remove them one by one based on their significance.
   - Steps:
     1. Start with all predictors in the model.
     2. Remove the least significant feature (highest p-value).
     3. Keep removing features one by one until all remaining features are statistically significant ($p\text{-value} < 0.05$).
   - Best when you want to start with all variables and systematically remove the least useful ones.

2.4 Selection of Variables/Features: Wrapper Methods

3. Stepwise Selection

- Combination of forward selection and backward elimination
- Adds features like forward selection but also removes unimportant features like backward elimination.
- Steps:
    a. Start with no predictors.
    b. Add the most significant feature.
    c. After adding a feature, check if any previously added feature has become insignificant (p-value > 0.05). If so, remove it.
    d. Repeat steps b and c until no more features can be added or removed.
- Best when you want a balanced approach that both adds and removes features dynamically.

2.5 Regression Model Performance Evaluation

- Evaluating the performance of a regression model is crucial to understanding its accuracy, generalizability, and reliability.
- Metrics used to evaluate regression model performance are:
    - Residual Analysis
    - Mean Squared Error (MSE)
    - Root Mean Squared Error (RMSE)
    - R-squared (R^2)
    - Adjusted R-squared
    - Akaike Information Criterion (AIC)
    - Bayesian Information Criterion (BIC)

2.5 Evaluation Metrics for Regressors

- Residual Analysis
  - Residuals represent the difference between actual and predicted values:
  - Residuals should be normally distributed with zero mean.

$$Residual = y_{\text{actual}} - y_{\text{predicted}}$$

- Mean Squared Error (MSE)
  - MSE measures the average squared difference between actual and predicted values.
  - Penalizes larger errors more than smaller ones.
  - Lower MSE means a better model.

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

Notations:

- y means actual value or ground truth
- y_hat means predicted value
- m = number of data samples

2.5 Evaluation Metrics for Regressors

- Root Mean Squared Error (RMSE)
  - RMSE is the square root of MSE, providing an error measure in the same unit as the dependent variable.
  - More interpretable than MSE.
  - Lower RMSE means a better model.

$$RMSE = \sqrt{MSE}$$

- Mean Absolute Error (MAE)
  - MAE measures the average absolute difference between actual and predicted values.
  - Less sensitive to outliers than MSE.
  - Lower MAE means better performance.

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

2.5 Evaluation Metrics for Regressors

- R-squared (R²) or Coefficient of Determination
  - R² measures the proportion of variance explained by the model.
  - Ranges from 0 to 1.
  - If $R^2 = 0.85$, the model explains 85% of variance in y.
  - Higher values indicate a better fit, while lower values suggest the model is less effective.

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

- Adjusted R-squared
  - R-squared always increases when more predictors are added, Adjusted R-squared increases only if the new predictors genuinely improve the model.
  - If adding a predictor does not improve the model, Adjusted R² decreases.
  - It prevents overfitting by balancing the model's performance with its complexity.

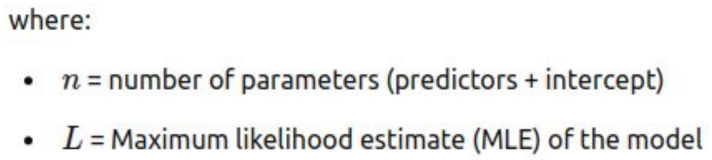
$$R^2_{\text{adj}} = 1 - \left(\frac{(1 - R^2)(m - 1)}{m - n - 1}\right)$$

where,
- m = number of data samples
- n = number of predictors/features

2.5 Evaluation Metrics for Regressors

- Akaike Information Criterion (AIC)
  - AIC helps in model selection, balancing accuracy and complexity.
  - Lower AIC means a better model.
  - AIC calculates model fit using log-likelihood while adding a penalty for the number of parameters, helping to avoid overfitting.

$$AIC = 2(n+1) - 2\log(L)$$



where:

- n = number of parameters (predictors + intercept)
- L = Maximum likelihood estimate (MLE) of the model

For linear regression:

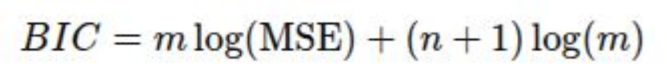
$$AIC = m\log(\text{MSE}) + 2(n+1)$$

where,

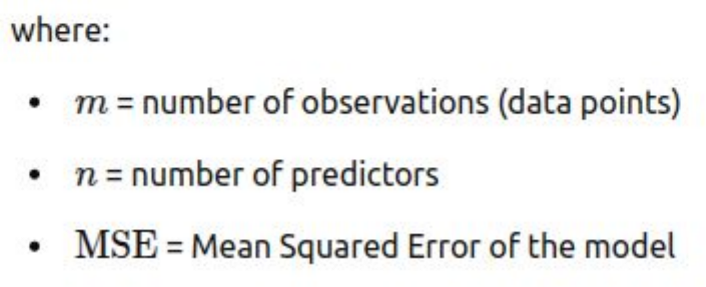
- m = number of data samples
- n = number of predictors/features
- MSE = Mean Squared Error

2.5 Evaluation Metrics for Regressors

- Bayesian Information Criterion (BIC)
  - BIC is similar to AIC but applies a stronger penalty for additional features.
  - Lower BIC indicates a better model.
  - Strongly penalizes more predictors, making it useful when dealing with many variables.



$$BIC = m \log(\mathrm{MSE}) + (n+1)\log(m)$$

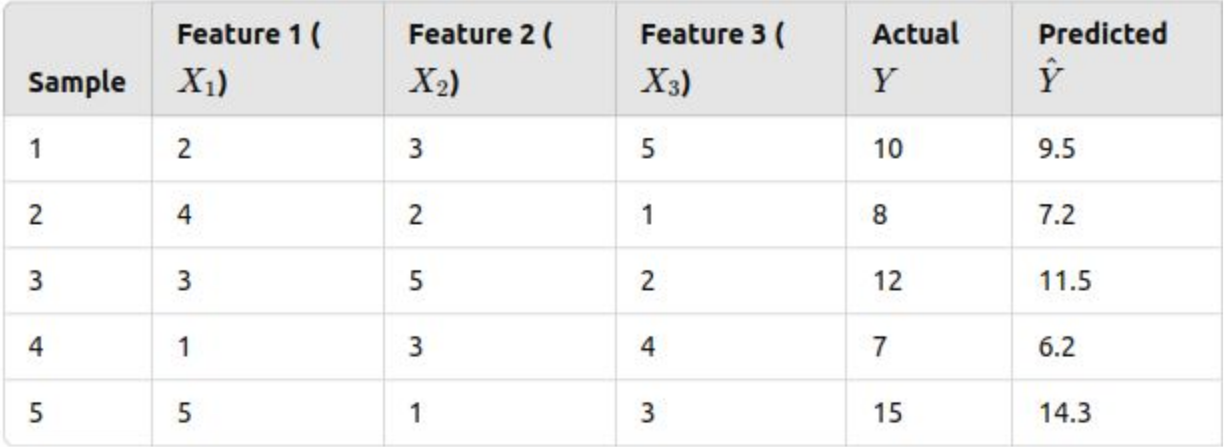


where:

- m = number of observations (data points)
- n = number of predictors
- MSE = Mean Squared Error of the model

2.5 Numerical on Regression Model Evaluation

- A linear regressor model trained with three features yields following predictions. The ground truth is also give below. Evaluate the performance of model using Residual Analysis, MSE, RMSE, MAE, R², Adjusted R², AIC, and BIC.



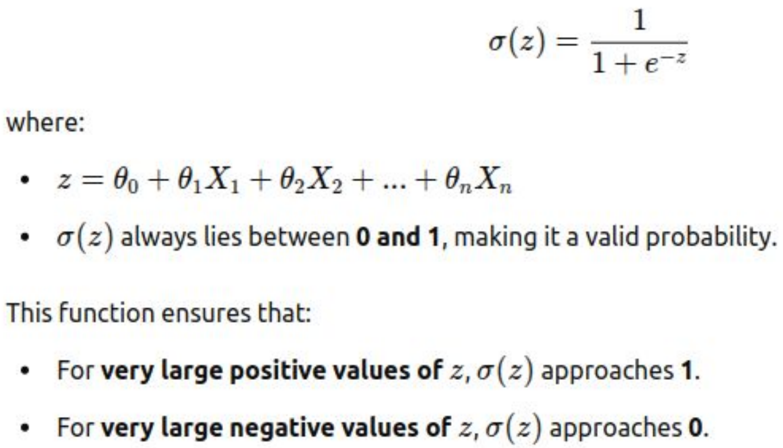
Sample	Feature 1 (X_1)	Feature 2 (X_2)	Feature 3 (X_3)	Actual Y	Predicted $\hat{Y}$
1	2	3	5	10	9.5
2	4	2	1	8	7.2
3	3	5	2	12	11.5
4	1	3	4	7	6.2
5	5	1	3	15	14.3

3.1 Logistic Regression

- Linear regression is gives us the prediction as continuous values.
- Classification Tasks:
    - Classification tasks refers to those tasks in which the output/prediction is one of the predefined classes or categories.
    - Example: Diabetic Vs Non-Diabetic Classification, spam vs. not spam.
    - For classification task, the outputs must be the probabilities i.e. values between 0 and 1.
- If we use linear regression for classification, the predicted values can be less than 0 or greater than 1, which is not meaningful for probability-based classification.
- We use logistic regression for the classification tasks.
- Logistic regression is a statistical method used for binary classification, meaning it predicts whether an instance belongs to one of two classes (e.g., spam vs. not spam, pass vs. fail, disease vs. no disease).
- Unlike linear regression, which predicts continuous values, logistic regression models the probability of an outcome using the **sigmoid function**.

3.1 Logistic Regression

- Logistic Regression is formulated by modifying the linear regression.
- Logistic regression uses sigmoid function that transforms the continuous values from linear regression to the range 0 to 1.
- Sigmoid Function:



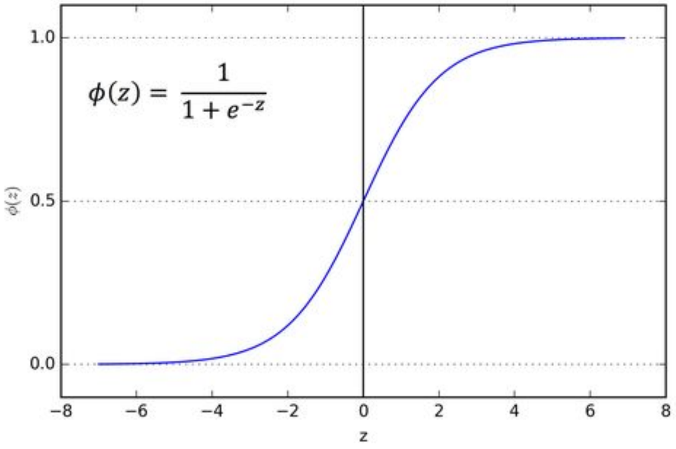
$$\sigma(z) = \frac{1}{1+e^{-z}}$$

where:

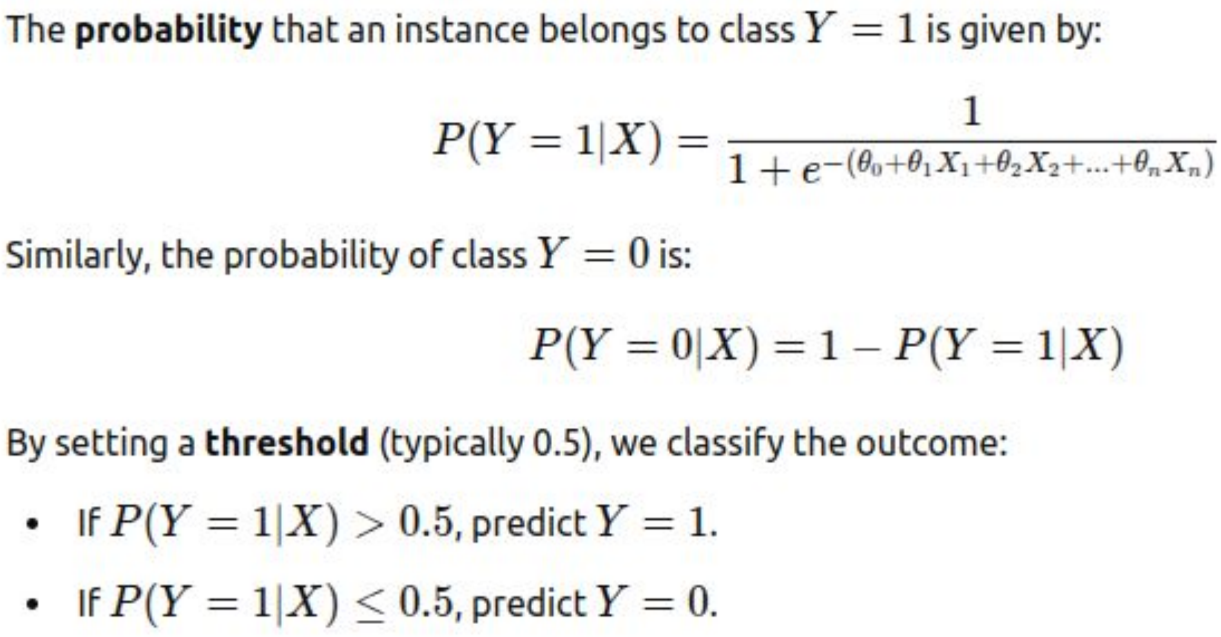
- $z = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_n X_n$
- $\sigma(z)$ always lies between **0 and 1**, making it a valid probability.

This function ensures that:

- For **very large positive values of** z, $\sigma(z)$ approaches **1**.
- For **very large negative values of** z, $\sigma(z)$ approaches **0**.



3.1 Logistic Regression Model



The **probability** that an instance belongs to class $Y = 1$ is given by:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_n X_n)}}$$

Similarly, the probability of class $Y = 0$ is:

$$P(Y = 0|X) = 1 - P(Y = 1|X)$$

By setting a **threshold** (typically 0.5), we classify the outcome:

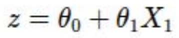
- If $P(Y = 1|X) > 0.5$, predict $Y = 1$.
- If $P(Y = 1|X) \leq 0.5$, predict $Y = 0$.

3.1 Logistic Regression Model: Numerical Concepts

- Let's consider a logistic regression model with one feature X_1, and assume the parameters (θ) are: $\theta_0=-2$, $\theta_1=0.8$

  What is the probability that Y=1 for a given input X1=3?

- We know,



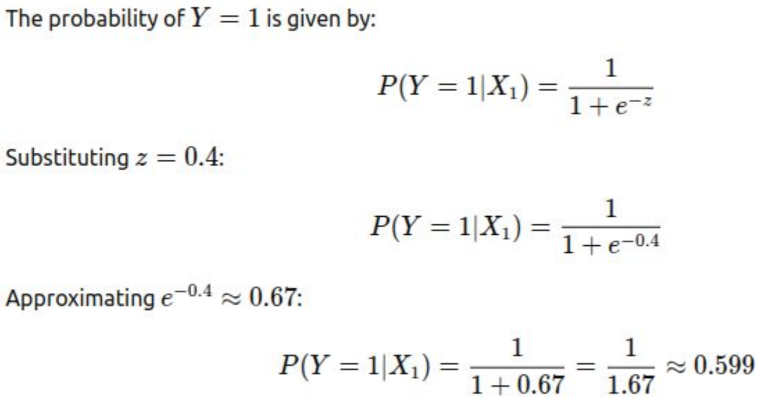
$$z = \theta_0 + \theta_1 X_1$$

Substituting values:

$$z = (-2) + (0.8 \times 3)$$

$$z = -2 + 2.4 = 0.4$$

Apply the Sigmoid Function,



The probability of $Y = 1$ is given by:

$$P(Y = 1|X_1) = \frac{1}{1 + e^{-z}}$$

Substituting $z = 0.4$:

$$P(Y = 1|X_1) = \frac{1}{1 + e^{-0.4}}$$

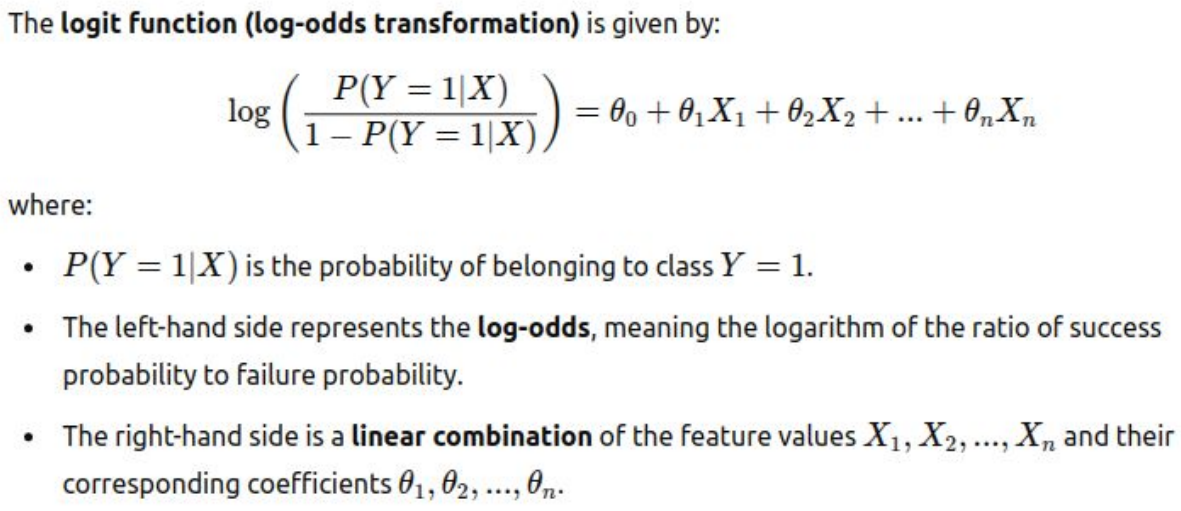
Approximating $e^{-0.4} \approx 0.67$:

$$P(Y = 1|X_1) = \frac{1}{1 + 0.67} = \frac{1}{1.67} \approx 0.599$$

Make a Prediction
Using a threshold of 0.5:
Since 0.599 > 0.5, we classify Y=1 (positive class).

3.2 Log-Odds in Logistic Regression

- Logistic regression does not directly predict class labels but estimates log-odds (logit function), which are then converted into probabilities using the sigmoid function.
- In the previous numerical “z” is the log-odd which is then converted to probability by applying sigmoid function i.e. Probability is obtained by applying the sigmoid function to log-odds.



The **logit function (log-odds transformation)** is given by:

$$\log\left(\frac{P(Y=1|X)}{1-P(Y=1|X)}\right) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_n X_n$$

where:

- $P(Y=1|X)$ is the probability of belonging to class $Y=1$.
- The left-hand side represents the **log-odds**, meaning the logarithm of the ratio of success probability to failure probability.
- The right-hand side is a **linear combination** of the feature values $X_1, X_2, ..., X_n$ and their corresponding coefficients $\theta_1, \theta_2, ..., \theta_n$.

This equation shows that **log-odds change linearly** with respect to the input features.

Relationship between log-odds and regression coefficients.

3.3 Logistic Regression Use Cases

- Healthcare: Disease Prediction
  - Cancer or Non-Cancer Prediction
  - Diabetic or Non-Diabetic Prediction
- Finance
  - Low risk or High risk classification
  - Fraud or not fraud transaction classification
- Marketing
  - Customer Churn Prediction
  - Positive or negative feedback classification