University of
South Australia

Linear and Logistic Regression
Week 4

Dr Srećko Joksimović

Linear Regression

What is a Regression Analysis?

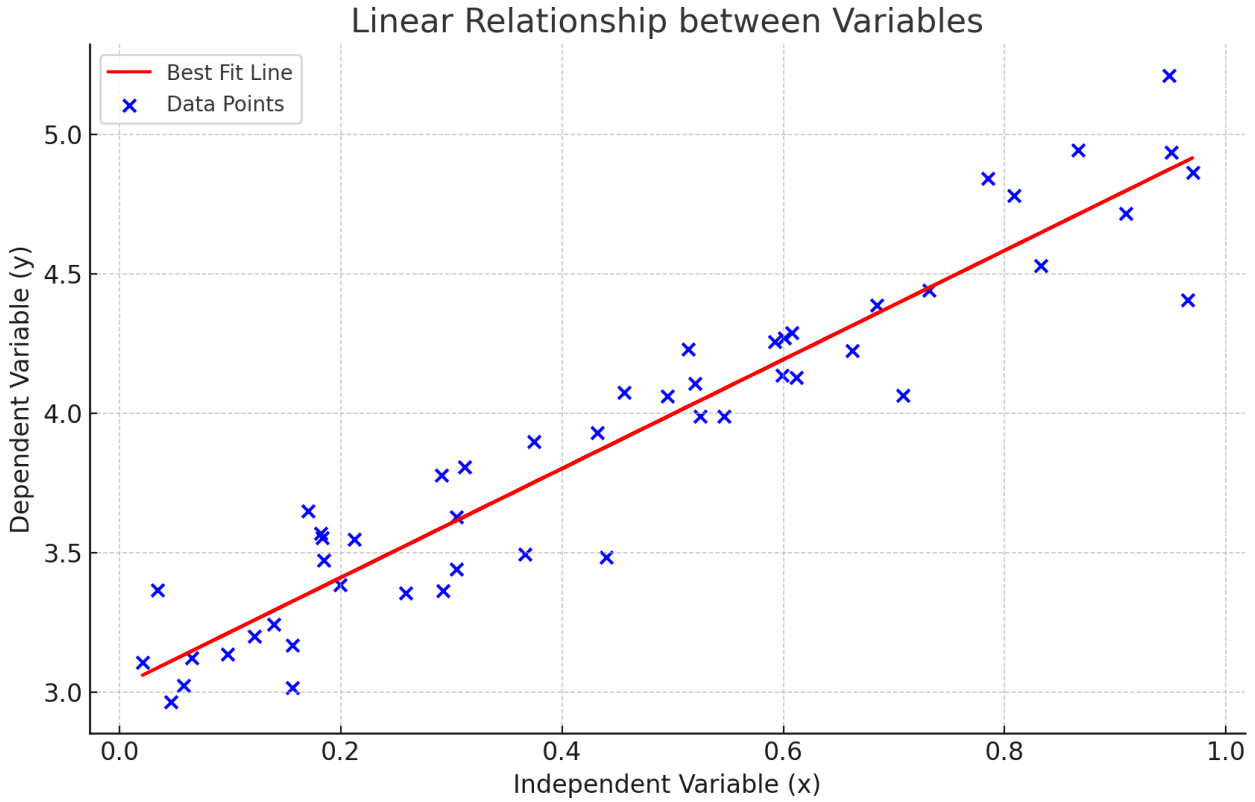
It is a statistical method used to model the relationships between variables

In Statistics

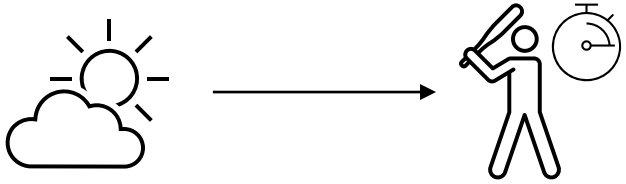
- Used to understand relationships, predict outcomes, and infer causal relationships.

In Machine Learning:

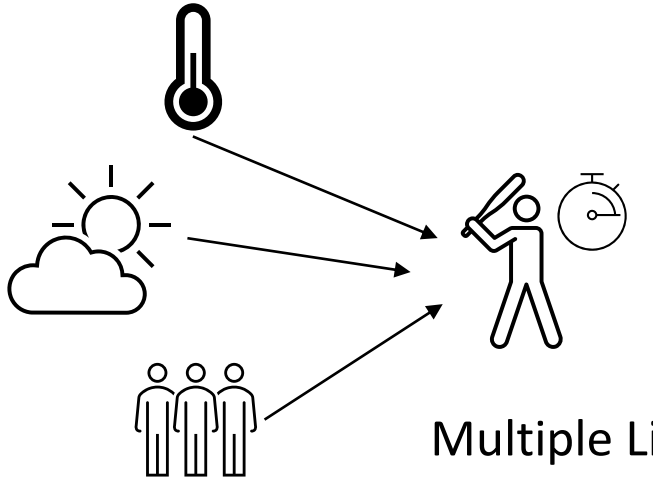
- Foundation for predictive modeling.



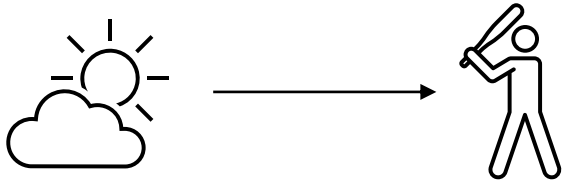
Types of Regression Analysis?



Simple Linear Regression



Multiple Linear Regression



Logistic Regression

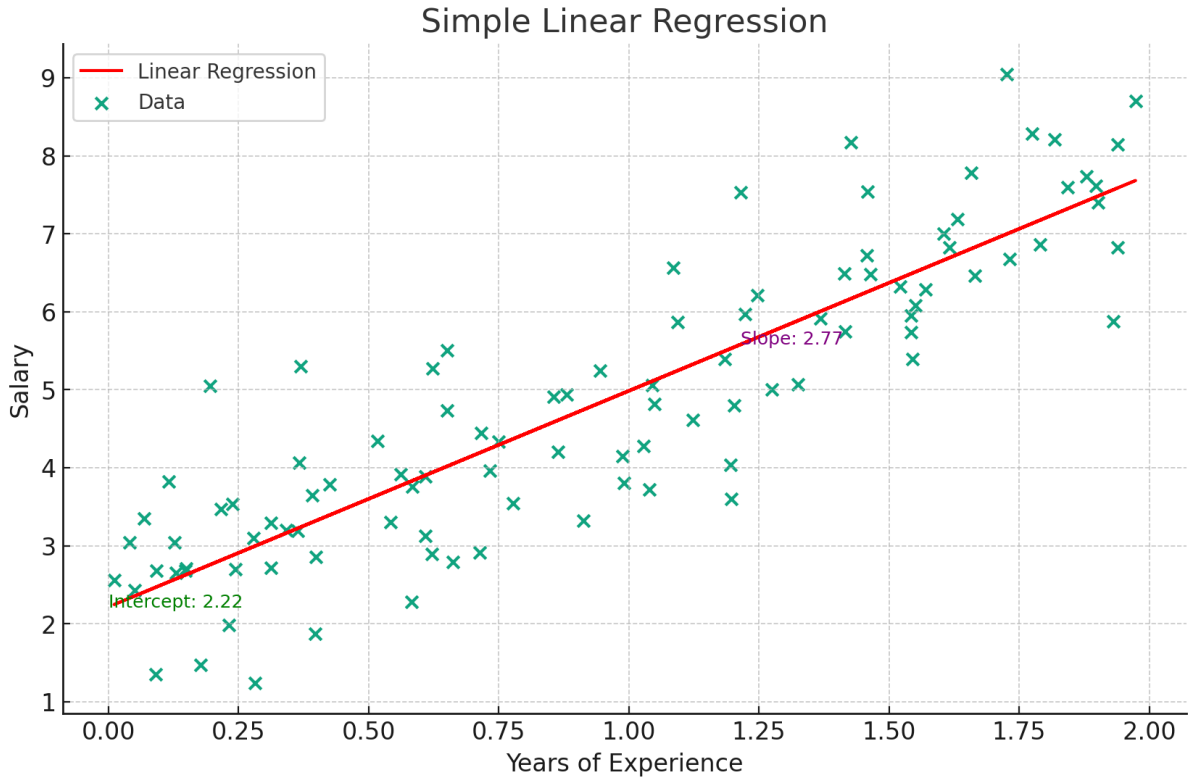
Simple Linear Regression

Independent Variable (X): This is the variable that you think will influence the dependent variable. It is also known as the **predictor** or **explanatory** variable.

Dependent Variable (Y): This is what you want to predict or understand. It is also known as the **response** or **outcome** variable.

$$y = \beta_0 + \beta_1 \cdot x + \epsilon$$

y Dependent variable
x Independent variable
β_0 Intercept (value of y when $x = 0$)
β_1 Slope (change in y for a one-unit change in x)
ϵ Error term (captures all other factors affecting y that are not included in x)



SLR – Use Cases

Economics: income vs spending

Healthcare: weight and blood pressure

Marketing: advertising expenditure and sales

Agriculture: fertilizer use and crop yield

Sports Analytics: player's practice time and performance metrics

Environmental Science: temperature and energy consumption

Multiple Linear Regression

- Multiple linear regression is an extension of simple linear regression that allows for the prediction of a dependent variable based on the values of **two or more independent variables**.

- This method provides a way to model a response variable based on multiple predictors, making it more versatile in handling real-world problems where multiple factors influence the outcome.

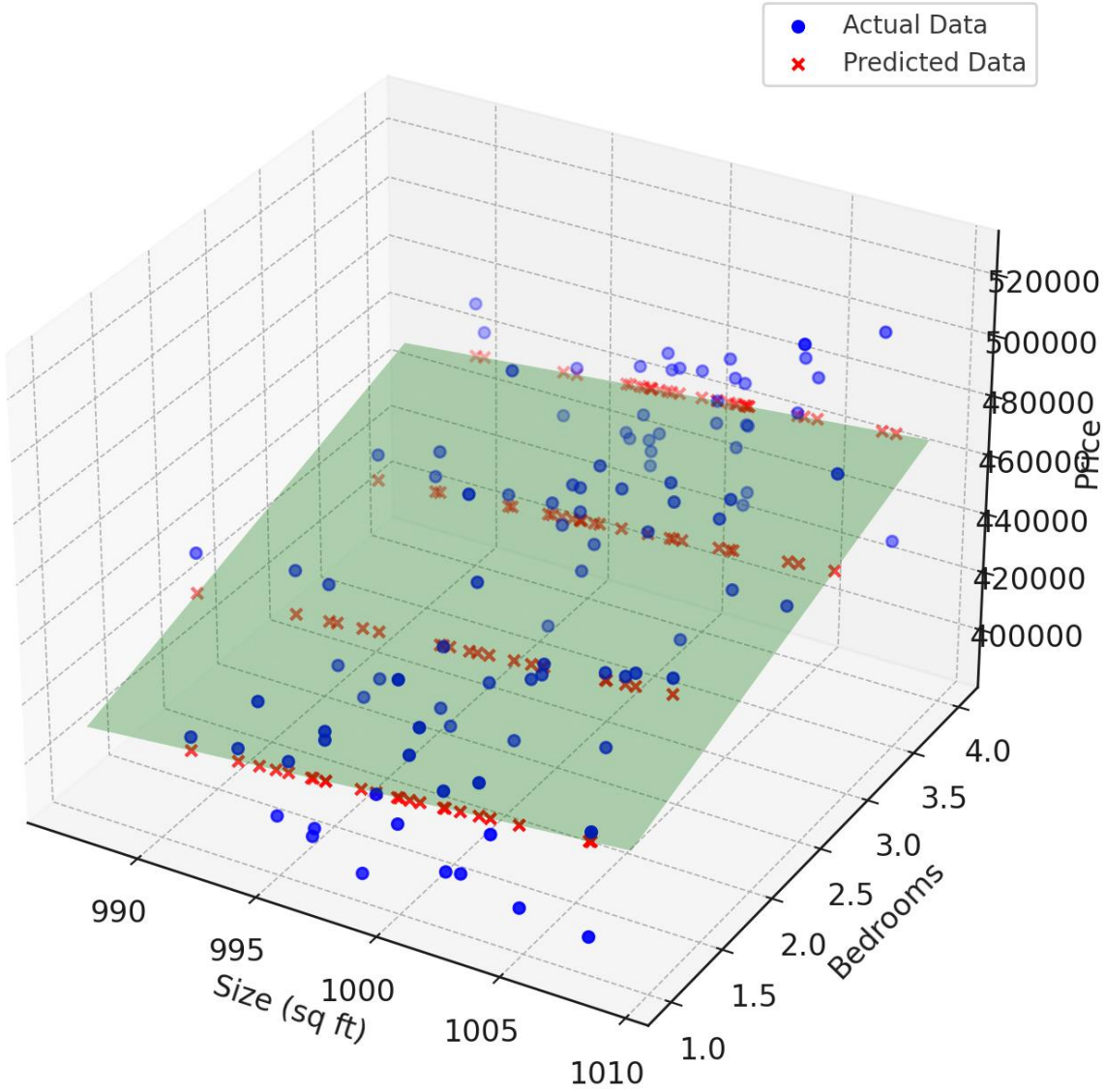
$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \ldots + \beta_n \cdot x_n + \epsilon$$

y - Dependent variable
$x_1, x_2, \ldots xn$ - Independent variables
β_0 - Intercept (value of y when $x = 0$)
$\beta_1, \beta_2, \ldots \beta_n$ - Coefficients for the corresponding independent variables
ϵ - Error term (captures all other factors affecting y that are not included in the model)

Multiple Linear Regression

- Let’s consider a scenario where we are trying to predict a house price (dependent variable) based on two independent variables: the **size of the house** in square feet and the **number of bedrooms**.
- We'll create some synthetic data for this example and then visualize the multiple linear regression model using a 3D plot.

$$Price = -304690.98 + 709.60\ x\ (size\ in\ sq\ ft) + 14076.88\ x\ (number\ of\ bedrooms) + \epsilon$$



MLR – How do we build a model?

Ordinary Least Squares (OLS)

- The standard method for fitting a linear regression model.
- Minimizes the sum of the squared differences between the observed and predicted values.
- Implemented in most statistical software.

Stepwise Regression

- Involves automatic selection of independent variables by adding or removing predictors based on statistical criteria.
- **Forward Selection**: Starts with no variables and adds them one by one.
- **Backward Elimination**: Starts with all variables and removes them one by one.
- **Bidirectional**: Combination of forward and backward methods.

MLR – How do we build a model?

Ridge Regression (L2 Regularization)

- The Adds a penalty term to the OLS loss function, which helps prevent overfitting.
- Can be useful when there is multicollinearity among the independent variables.

Lasso Regression (L1 Regularization)

- Similar to Ridge, but uses a different penalty that can force some coefficients to be exactly zero.
- Useful for variable selection and reducing complexity.

Generalized Linear Models (GLMs)

- Extends linear regression to allow for non-normal distributions of the dependent variable and non-linear relationships through link functions.
- Includes logistic regression, Poisson regression, etc.
- More flexible in handling different types of response variables and relationships.

MLR - Interpretation

Intercept (β_0)

- Represents the predicted value of the dependent variable when all independent variables are zero.
- Often has no practical interpretation, especially when zero values for all independent variables do not make sense in the given context.

Coefficients ($\beta_1, \beta_2, \ldots \beta_n$)

- Represent the change in the dependent variable for a one-unit change in the corresponding independent variable, assuming all other variables are held constant.
- Positive coefficients indicate a positive relationship, while negative coefficients indicate a negative relationship.
- The magnitude of the coefficient tells you the size of the effect.

MLR - Interpretation

Statistical Significance

- ***P-values*** for the coefficients test the null hypothesis that the coefficient is equal to zero (no effect).
- A small p-value (e.g., < 0.05) suggests that there is evidence to reject the null hypothesis, implying that the independent variable is a significant predictor of the dependent variable

Confidence Intervals

- Provide a range of plausible values for the coefficients.
- If the confidence interval does not include zero, it's another indication of statistical significance.

MLR - Interpretation

Multiple Coefficient of Determination (R^2)

- Measures the proportion of variance in the dependent variable that is explained by the independent variables.
- Ranges from 0 to 1, with higher values indicating a better fit
- Higher R^2 values indicate a better fit but should be considered alongside other diagnostics, especially when comparing models with different numbers of predictors.

$$R^2 = \frac{Explained\ Variance}{Total\ Variance} = 1 - \frac{Sum\ of\ Squared\ Residuals}{Total\ Sum\ of\ Squares}$$

Adjusted R^2

- Adjusts R^2 for the number of predictors, preventing it from automatically increasing with more variables.
- Useful for comparing models with different numbers of independent variables.

MLR - Interpretation

Residual Analysis

- Plotting residuals (differences between observed and predicted values) can help detect non-linearity, unequal error variances, and outliers.
- Residuals should be randomly scattered around zero with no clear patterns.

Check Assumptions

- Assess the assumptions of linearity, independence, homoscedasticity, and normality.
- Violations of these assumptions may require transformations, robust methods, or different modeling approaches.

Domain Knowledge

- Consider the practical importance of the effects, not just statistical significance.
- Collaborate with domain experts to ensure the interpretation aligns with theoretical expectations and real-world applicability.

MLR - Assumptions

- **Linearity**
  - The relationship between the response variable and the explanatory variable is linear
- **Independence (of errors)**
  - The errors of the responses variable are uncorrelated with each other
- **Normality (of errors)**
  - The errors of the response variable are normally distributed
- **Constant variance (of errors)**
  - Also called **homoscedasticity**
  - The response variables have the same variance in their error regardless of the values of the explanatory variables
- **Multicollinearity**
  - The explanatory variables are not correlated

MLR – Pros and Cons

Advantages

- Model more complex relationships by considering multiple factors.
- More applicable to real-world scenarios where multiple variables influence the outcome.

Limitations

- Requires careful selection of relevant independent variables to avoid multicollinearity (high correlation between predictors – next slides).
- More complex to interpret.
- Can lead to overfitting if too many variables are included without proper validation and selection techniques.

Logistic regression

Logistic regression - overview

- It's a type of regression analysis used when the dependent variable is **categorical**.
- While linear regression models a continuous outcome, logistic regression models **the probability of a binary outcome** (two classes, such as 0 or 1, Yes or No, True or False).

Logistic Regression	Linear Regression
Dependent variable	
Binary/Categorical (e.g., pass/fail, diseased/healthy)	Continuous (e.g., house price, weight)
Model function	
Logistic function (S-curve) that ensures predicted values are between 0 and 1	Linear combination of predictors (straight line)
Interpretation of Coefficients	
Change in the log odds of the dependent variable for a one-unit change in the predictor	Change in the dependent variable for a one-unit change in the predictor
Assumptions	
Independence, linearity of log odds, absence of multicollinearity	Linearity, independence, absence of multicollinearity, homoscedasticity, normality of residuals

Model function

Binary outcome: Spam/Not spam? Fraudulent (Yes/No)?

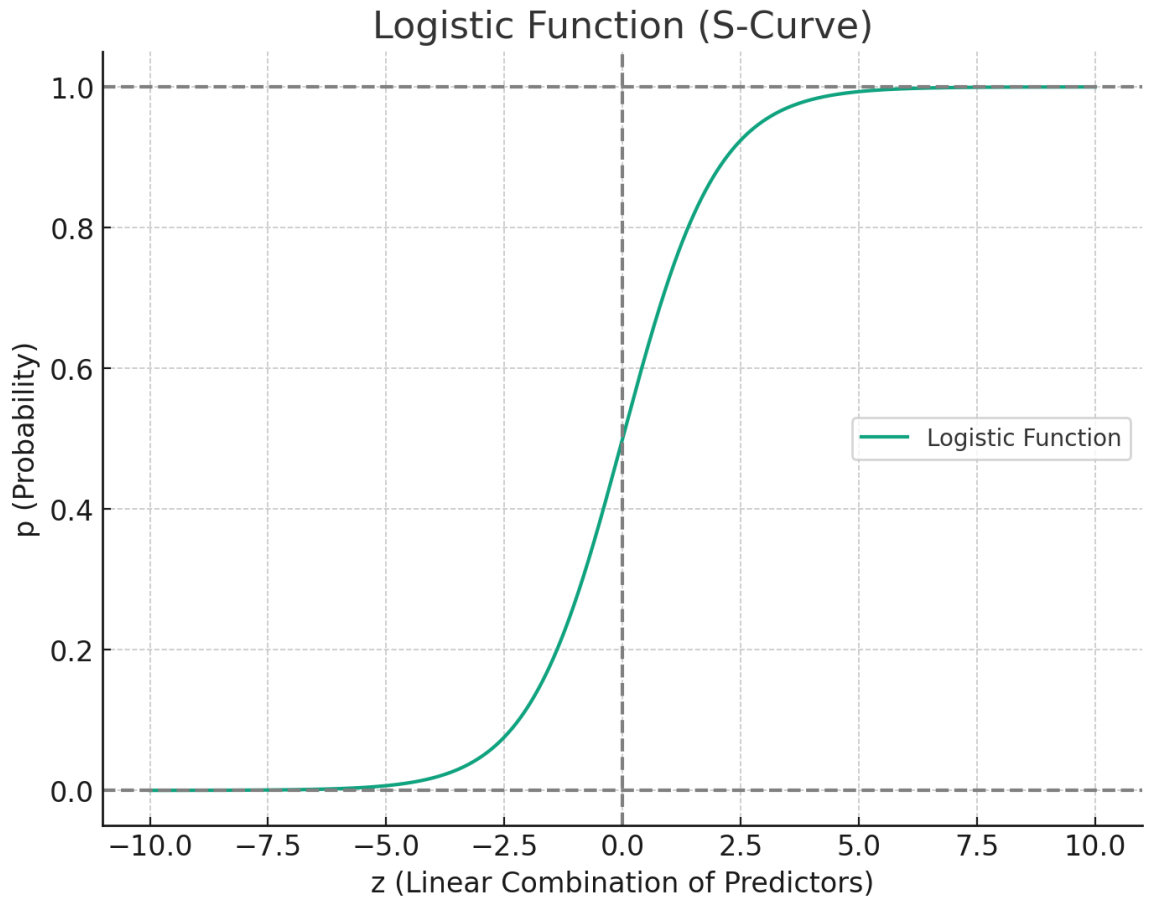
- What is the probability of the outcome?

$$z = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \ldots + \beta_n \cdot x_n + \epsilon$$

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

The linear combination *z* represents the **log odds** of the outcome. The logistic function transforms these log odds into **probabilities**.

- x-axis - the linear combination of predictors [-∞,∞]
- y-axis the probability *p* (the output of the logistic function [0, 1]
- The shape of the curve ensures that the probabilities are bounded between 0 and 1, no matter the value of *z*.
- The point where the curve crosses the x-axis corresponds to a probability of 0.5.



Odds & Log Odds

Odds

Odds represent the ratio of the probability of an event occurring to the probability of it not occurring

The odds of an event with probability p are given by $\frac{p}{1-p}$

Example: If the probability of an event success is 0.75, the odds are $\frac{0.75}{0.25} = 3$, meaning that success is three times more likely than failure

Log Odds

Log odds (or logit) is the natural logarithm of the odds.

The log odds for probability p are $\ln(\frac{p}{1-p})$

In logistic regression, the log odds are modelled as a linear combination of predictors:

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

That's why we use logistic function to transform the log odds back to a probability:

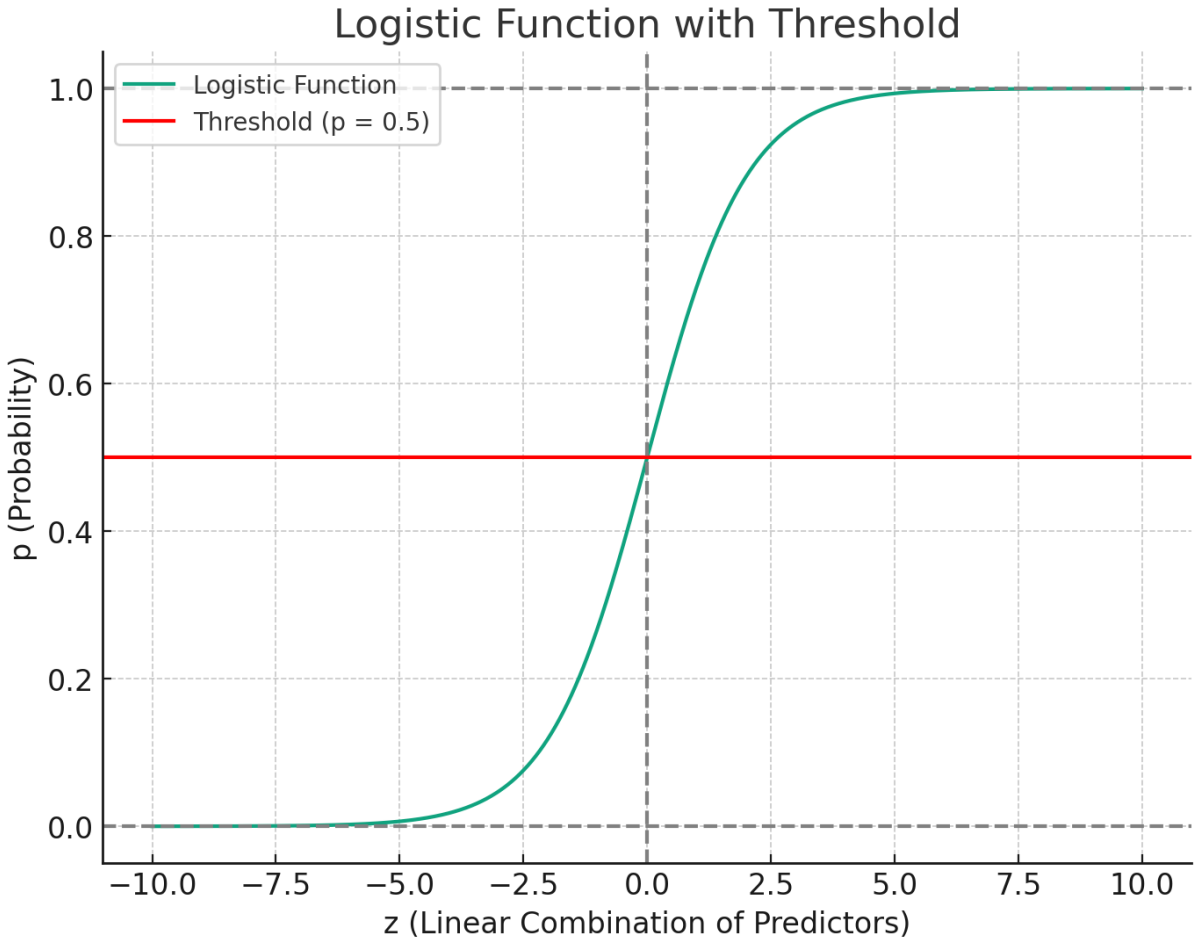
$$p = \frac{1}{1 + e^{-z}}$$

Interpretation

If p=0.7, it means there's a **70% chance of the outcome being 1**, according to the model

Often, a threshold (commonly 0.5) is used to classify the predicted probabilities into the binary categories. If $p \geq 0.5$, the predicted class is 1; otherwise, it's 0.

By **adjusting the threshold**, you can control the trade-off between sensitivity (true positive rate) and specificity (true negative rate), which is often visualized using a Receiver Operating Characteristic (ROC) curve.



Interpretation - Coefficients

- In logistic regression, the coefficients represent the change **in the log odds of the dependent variable being 1 for a one-unit change in the predictor**, assuming all other variables are held constant.
  - Let's say we have a coefficient β_i for a predictor x_i. The interpretation is that a one-unit increase in x_i leads to a β_i change in the log odds of the dependent variable being 1.

- Exponentiating the coefficient gives us **the odds ratio**, which is a more intuitive way to interpret the effect of the predictor.
  - **Odds Ratio**: The odds ratio for a predictor with coefficient β_i is e^{β_i} .
  - **Interpretation**: The odds ratio tells us the multiplicative change in the odds for a one-unit increase in the predictor.

Positive Coefficient ($\beta_i > 0$):

- A one-unit increase in x_i **increases** the log odds by β_i.
- The odds are multiplied by e^{β_i} (greater than 1) for a one-unit increase in x_i.
- Example: If $\beta_i = 0.5$, a one-unit increase in x_i multiplies the odds by $e^{0.5} \approx 1.65$

Negative Coefficient ($\beta_i < 0$):

- A one-unit increase in x_i **decreases** the log odds by β_i.
- The odds are multiplied by e^{β_i} (less than 1) for a one-unit increase in x_i.
- Example: If $\beta_i = -0.5$, a one-unit increase in x_i multiplies the odds by $e^{0.5} \approx 0.61$

Logistic regression - Example

```
# Sample Data
car_data <- data.frame(
  PurchaseLuxuryCar = c(0, 1, 0, 1, 1),
  AnnualIncome = c(50000, 100000, 40000, 80000, 120000), # in dollars
  Age = c(25, 35, 22, 40, 45),
  HomeOwner = as.factor(c(0, 1, 0, 1, 1)) # 0 = No, 1 = Yes
)

# Fitting a Logistic Regression Model
model <- glm(PurchaseLuxuryCar ~ AnnualIncome + Age + HomeOwner, data = car_data, family = binomial)

# Summary of the Model
summary(model)
```

Logistic regression - Example

```
Call:
glm(formula = PurchaseLuxuryCar ~ AnnualIncome + Age + HomeOwner, family = binomial,
    data = car_data)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.4567  -0.4567   0.5433   0.5433   1.4567

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    -3.2051     1.2573  -2.549   0.0108
AnnualIncome    0.0001     0.0001   1.978   0.0481
Age             0.0723     0.0321   2.254   0.0243
HomeOwner1      1.5433     0.7521   2.053   0.0401

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6.7301  on 4  degrees of freedom
Residual deviance: 3.1823  on 1  degrees of freedom
AIC: 11.182
```

Logistic regression - Example

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

$$z = Intercept + \beta_1 \, x \, AnnualIncome + \beta_2 \, x \, Age + \beta_3 \, x \, HomeOwner$$

$$Intercept = -3.2051$$
$$\beta_1(AnnualIncome) = 0.0001$$
$$\beta_2 \, (Age) = 0.0723$$
$$\beta_3 \, (HomeOwner1) = 1.5433$$

New Data Point
Annual Income: $60,000
Age: 30
Homeowner: Yes (1)

$$z = -3.2051 + (0.0001 \, x \, 60{,}000) + (0.0723 \, x \, 30) + (1.5433 \, x \, 1)$$

$$z = -3.2051 + 6 + 2.169 + 1.5433 = 6.5072$$

$$p = \frac{1}{1 + e^{-6.5072}} = \frac{1}{1 + 0.0014926533} = 0.9985$$

$$p = 99.85\%$$

Learning Logistic regression

How do we train a model?

1. Define the Model (previous slides)
2. Define the Cost Function
    1. Optimisation Algorithm
    2. Regularisation (Optional)
    3. Convergence and Solution
3. Model Evaluation
4. Happy times – making predictions (previous example)

Cost function

- Cost function for logistic regression is derived from the likelihood function

$$If\ y = 1: -\log(p)$$

$$If\ y = 0: -\log(1 - p)$$

$$Cost = -y * log(p) - (1 - y) * \log(1 - p)$$

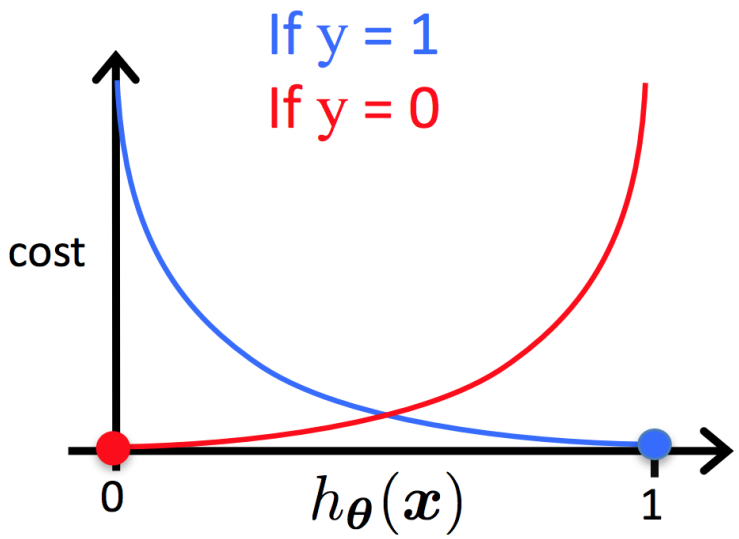
Cost function

- Cost function for logistic regression is derived from the likelihood function

$$If\ y = 1: -\log(p)$$

$$If\ y = 0: -\log(1 - p)$$

$$Cost = -y * log(p) - (1 - y) * \log(1 - p)$$



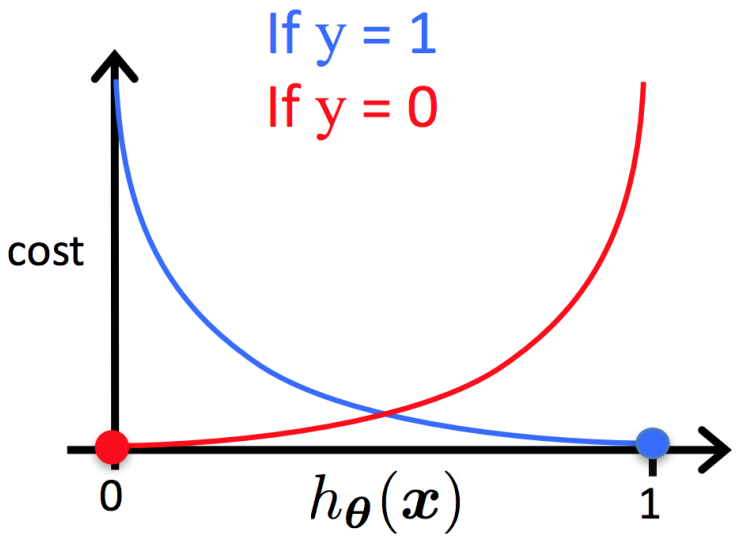
Cost function

- Cost function for logistic regression is derived from the likelihood function

$$If\ y = 1: -\log(p)$$

$$If\ y = 0: -\log(1 - p)$$

$$Cost = -y * log(p) - (1 - y) * \log(1 - p)$$



Predicted Probability: The predicted probability *p* comes from the logistic function.

•**Actual Outcome**: *y* is the actual binary outcome (0 or 1).

•**Penalising Incorrect Predictions**: The cost function is designed to heavily penalize predictions that are wrong:

- If *y*=1 but *p* is close to 0, the cost will be large.
- If *y*=0 but *p* is close to 1, the cost will also be large.

Overall Cost function

- The overall cost function (or log loss) for the entire dataset is the average cost over all observations

$$J(\beta) = \frac{1}{n}\sum_{i=1}^{n}[-y_i \log(p_i) - (1 - y_i)\log(1 - p_i)]$$

Logistic regression - learning

1. **Initialise parameters**
   - The coefficients β are initialized with some starting values, often zeros or small random numbers.

Logistic regression - learning

1. **Initialise parameters**
   - The coefficients β are initialized with some starting values, often zeros or small random numbers.

2. **Compute the Predicted Probabilities**
   - For each observation i, compute the predicted probability p_i using the logistic function applied to the linear combination of predictors:

$$p_i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_n x_{in})}}$$

Logistic regression - learning

2. Compute the Predicted Probabilities

- For each observation i, compute the predicted probability p_i using the logistic function applied to the linear combination of predictors:

$$p_i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_n x_{in})}}$$

3. Compute the Cost Function (Negative Log-Likelihood)

- Calculate the value of the cost function $J(\beta)$ using the formula below with the current coefficients and predicted probabilities:

$$J(\beta) = \frac{1}{n} \sum_{i=1}^{n} [-y_i \log(p_i) - (1 - y_i) \log(1 - p_i)]$$

Logistic regression - learning

3. Compute the Cost Function (Negative Log-Likelihood)

- Calculate the value of the cost function $J(\beta)$ using the formula below with the current coefficients and predicted probabilities:

$$J(\beta) = \frac{1}{n}\sum_{i=1}^{n}[-y_i \log(p_i) - (1 - y_i)\log(1 - p_i)]$$

4. Compute the Gradients

- The gradients are **the partial derivatives** of the cost function with respect to each coefficient. They indicate the direction and rate of change of the cost function as the coefficients are adjusted

4. Compute the Gradients

$$J(\beta) = \frac{1}{n}\sum_{i=1}^{n}[-y_i \log(p_i) - (1 - y_i)\log(1 - p_i)]$$

$$p_i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_n x_{in})}}$$

Partial derivatives for a specific coefficient β_i

$$\frac{\partial J(\beta)}{\partial \beta_j} = \sum_{i=1}^{n}(p_i - y_i)x_{ij}$$

p_i is the predicted probability for observation i.
y_i is the actual outcome for observation i.
x_{ij} is the value of predictor j for observation i.

4. Compute the Gradients

$$J(\beta) = \frac{1}{n}\sum_{i=1}^{n}[-y_i \log(p_i) - (1 - y_i)\log(1 - p_i)]$$

$$p_i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_n x_{in})}}$$

Partial derivatives for a specific coefficient β_i

$$\frac{\partial J(\beta)}{\partial \beta_j} = \sum_{i=1}^{n}(p_i - y_i)x_{ij}$$

p_i is the predicted probability for observation i.
y_i is the actual outcome for observation i.
x_{ij} is the value of predictor j for observation i.

Example:
Let's consider a simple example with a single predictor and compute the gradient for β_1:

- Observations: $n = 3$
- Predicted probabilities: $p = [0.7, 0.6, 0.8]$
- Actual outcomes: $y = [1, 0, 1]$
- Predictor values: $x = [2, 3, 1]$

4. Compute the Gradients

$$J(\beta) = \frac{1}{n}\sum_{i=1}^{n}[-y_i \log(p_i) - (1 - y_i)\log(1 - p_i)]$$

$$p_i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_n x_{in})}}$$

Partial derivatives for a specific coefficient β_i

$$\frac{\partial J(\beta)}{\partial \beta_j} = \sum_{i=1}^{n}(p_i - y_i)x_{ij}$$

p_i is the predicted probability for observation i.
y_i is the actual outcome for observation i.
x_{ij} is the value of predictor j for observation i.

Example:
Let's consider a simple example with a single predictor and compute the gradient for β_1:

- Observations: $n = 3$
- Predicted probabilities: $p = [0.7, 0.6, 0.8]$
- Actual outcomes: $y = [1, 0, 1]$
- Predictor values: $x = [2, 3, 1]$

$$\frac{\partial J(\beta)}{\partial \beta_1} = (0.7 - 1) * 2 + (0.6 - 0) * 3 + (0.8 - 1) * 1$$

$$\frac{\partial J(\beta)}{\partial \beta_1} = 1$$

Logistic regression - learning

4. Compute the Gradients

- The gradients are **the partial derivatives** of the cost function with respect to each coefficient. They indicate the direction and rate of change of the cost function as the coefficients are adjusted

5. Update the Coefficients

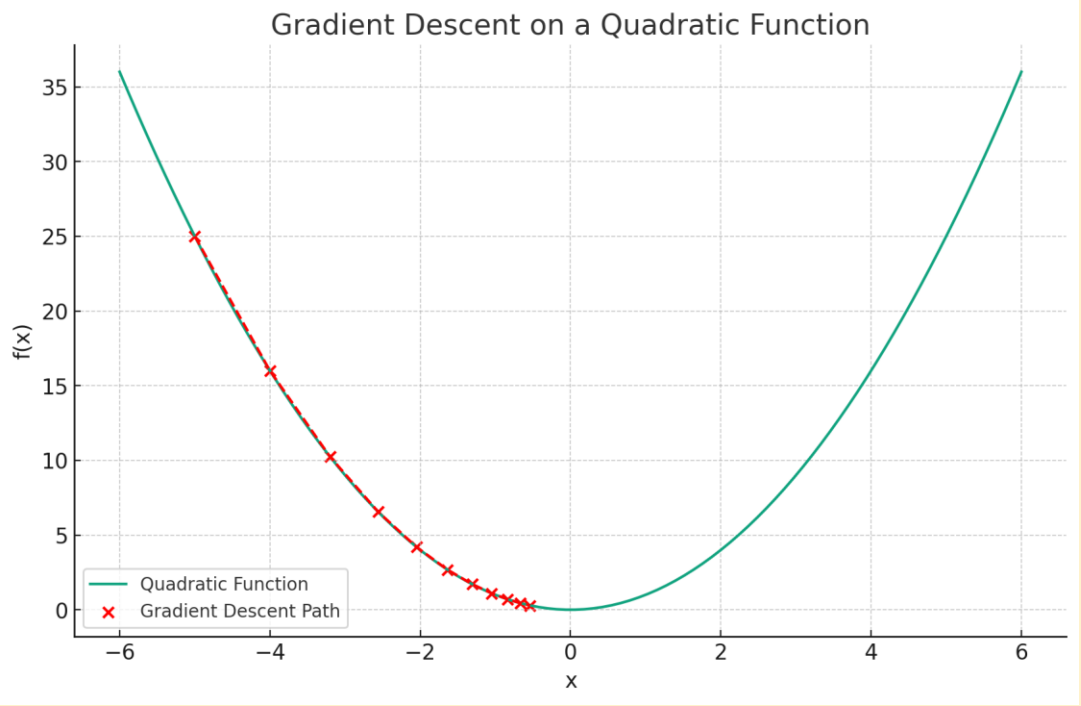
- Use an optimization algorithm like Gradient Descent to update the coefficients in the direction that reduces the cost function. The update is often done with a learning rate that controls the step size

$$\beta_j := \beta_j - \alpha \frac{\partial J(\beta)}{\partial \beta_j}$$

Gradient Descent

- Imagine you're on a mountain, and your goal is to reach the lowest point in the valley. You can't see the entire landscape, so you decide to take steps in the direction where the slope is steepest downward. By repeatedly taking steps in the steepest downward direction, you hope to reach the valley.

- Gradient Descent is an optimization algorithm that works in a similar way, but in the context of mathematical functions.

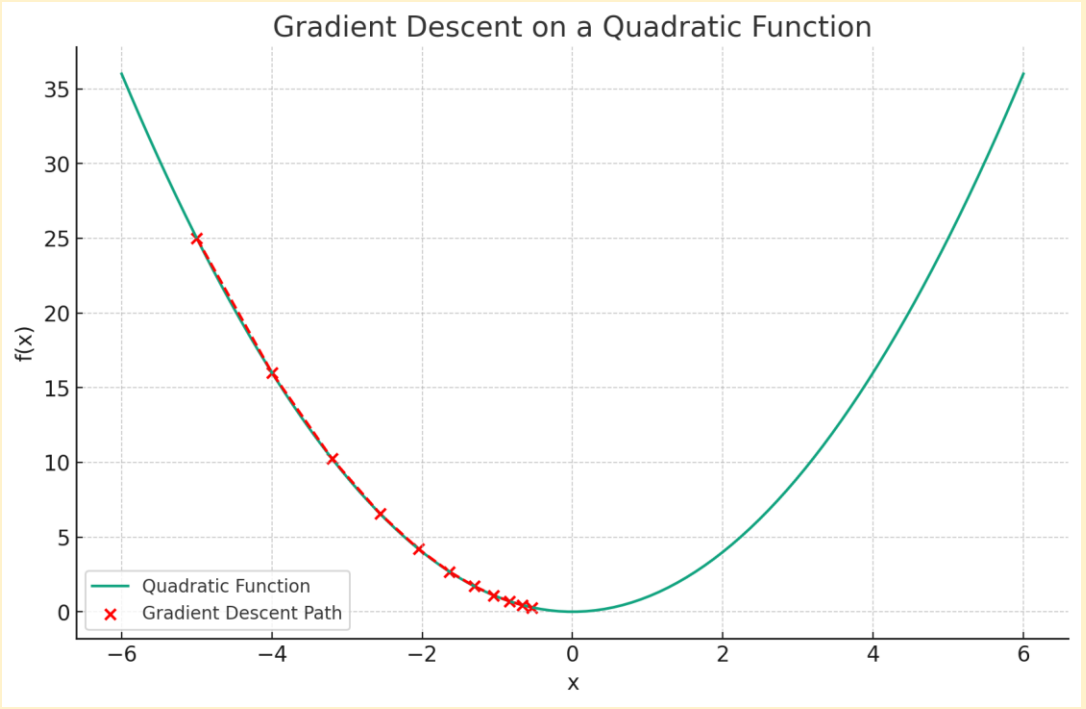


1. Start with Initial Guesses: Just like you start at a specific point on the mountain, Gradient Descent starts with initial guesses for the coefficients of the model.
2. Find the Slope (Gradient): Determine the slope or "steepness" of the function at the current point. In mathematical terms, this is the gradient, which is calculated using **the partial derivatives** of **the cost function with respect to each coefficient**.
3. Take a Step Downward: Move in the direction where the **slope** is steepest downward. This step is controlled by a parameter called the **learning rate**, which determines how big each step is.
4. Repeat Steps 2-3: Continue finding the slope and taking steps downward until you reach a point where the slope is close to zero, meaning you've found a minimum.
5. Reach the Minimum: The point you've reached represents the minimum of the function, which corresponds to the best-fitting model in the context of logistic regression.

Gradient Descent

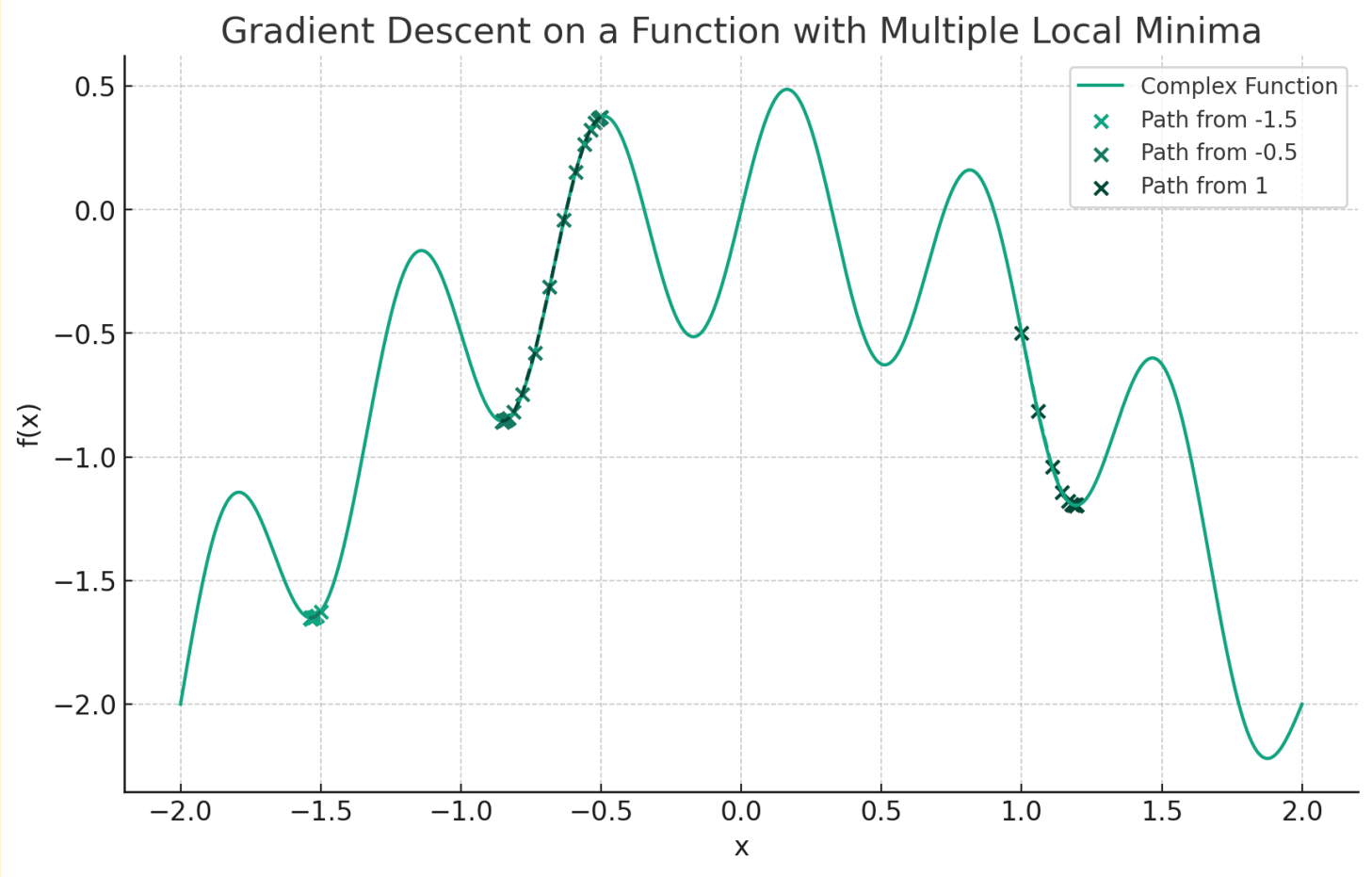
Application to Logistic Regression:
In logistic regression, the "mountain" is the cost function, representing how well the model fits the data. The "valley" is the minimum of the cost function, representing the best-fitting model.

- **Coefficients**: These are like your coordinates on the mountain.
- **Cost Function**: This represents the height of the mountain at your current coordinates (coefficients).
- **Gradient**: This is the slope of the mountain, telling you how to adjust the coefficients to reduce the cost function.



1. Start with Initial Guesses: Just like you start at a specific point on the mountain, Gradient Descent starts with initial guesses for the coefficients of the model.
2. Find the Slope (Gradient): Determine the slope or "steepness" of the function at the current point. In mathematical terms, this is the gradient, which is calculated using **the partial derivatives** of **the cost function with respect to each coefficient**.
3. Take a Step Downward: Move in the direction where the **slope** is steepest downward. This step is controlled by a parameter called the **learning rate**, which determines how big each step is.
4. Repeat Steps 2-3: Continue finding the slope and taking steps downward until you reach a point where the slope is close to zero, meaning you've found a minimum.
5. Reach the Minimum: The point you've reached represents the minimum of the function, which corresponds to the best-fitting model in the context of logistic regression.

Gradient Descent – so, what's the problem?



Logistic regression - learning

5. Update the Coefficients

- Use an optimization algorithm like Gradient Descent to update the coefficients in the direction that reduces the cost function. The update is often done with a learning rate that controls the step size

$$\beta_j := \beta_j - \alpha \frac{\partial J(\beta)}{\partial \beta_j}$$

6. Compute the Gradients

- Continue iterating through **steps 2-5**, recalculating **the predicted probabilities**, **cost function, gradients**, and **updating the coefficients** until the cost function **converges to a minimum**.

Assumptions

1. **Binary Outcome**:
   - The dependent variable is binary (two categories).
   - Example: Pass/Fail, Buy/Don't Buy.
2. **Linearity of Log Odds**:
   - The log odds of the outcome are modeled as a linear combination of predictor variables.
   - Non-linear relationships may require transformations or polynomial terms.
3. **Independence of Observations**:
   - Observations are independent of each other.
   - No autocorrelation (e.g., time-series data may violate this).
4. **No Perfect Multicollinearity**:
   - Predictor variables are not perfectly correlated with each other.
   - High correlation between predictors can lead to unstable estimates.
5. **No Outliers or High Leverage Points**:
   - Extreme values can overly influence the model.
   - Consider diagnostics to detect influential observations.
6. **Large Sample Size**:
   - Logistic regression requires a sufficient number of cases for each predictor variable.
   - Rule of thumb: At least 10-15 cases for each predictor in the smaller outcome category.
7. **No Complete Separation**:
   - A situation where the outcomes are perfectly predicted by a predictor should be avoided.
   - It leads to infinite estimates for coefficients.

Pros & Cons

- **Advantages**
  - Simplicity and interpretability
  - Efficient with linearly separable data
  - Probability estimation
  - Low computational cost

- **Disadvantages**
  - Limited complexity and feature interactions
  - Susceptible to overfitting
  - Assumption of linearity
  - Sensitivity to outliers

University of
South Australia

INFS 5100 Predictive Analytics

Q&A