

MODULE 3**CHAPTER 4****SIMILARITY-BASED LEARNING****Similarity or Instance-based Learning**

Similarity-based classifiers use similarity measures to locate the nearest neighbors and classify a test instance which works in contrast with other learning mechanisms such as decision trees or neural networks. Similarity-based learning is also called as Instance-based learning/Just-in time learning since it does not build an abstract model of the training instances and performs lazy learning when classifying a new instance. This learning mechanism simply stores all data and uses it only when it needs to classify an unseen instance. The advantage of using this learning is that processing occurs only when a request to classify a new instance is given. This methodology is particularly useful when the whole dataset is not available in the beginning but collected in an incremental manner.

Difference between Instance-and Model-based Learning**Table 4.1:** Differences between Instance-based Learning and Model-based Learning

Instance-based Learning	Model-based Learning
Lazy Learners	Eager Learners
Processing of training instances is done only during testing phase	Processing of training instances is done during training phase

Instance-based Learning	Model-based Learning
No model is built with the training instances before it receives a test instance	Generalizes a model with the training instances before it receives a test instance
Predicts the class of the test instance directly from the training data	Predicts the class of the test instance from the model built
Slow in testing phase	Fast in testing phase
Learns by making many local approximations	Learns by creating global approximation

Some examples of Instance-based Learning **algorithms** are:

KNN

Variants of KNN

Locally weighted regression

Learning vector quantization

Self-organizing maps

RBF networks

Nearest-Neighbor Learning

- A powerful classification algorithm used in pattern recognition.
- K nearest neighbors stores all available cases and classifies new cases based on a similarity measure (e.g distance function)
- One of the top data mining algorithms used today.
- A non-parametric lazy learning algorithm (An Instance based Learning method).
- Used for both classification and regression problems.
-

Algorithm 4.2: k-NN

Algorithm 4.1: k-NN

Inputs: Training dataset T , distance metric d , Test instance t , the number of nearest neighbors k

Output: Predicted class or category

Prediction: For test instance t ,

1. For each instance i in T , compute the distance between the test instance t and every other instance i in the training dataset using a distance metric (Euclidean distance).
[Continuous attributes - Euclidean distance between two points in the plane with coordinates (x_1, y_1) and (x_2, y_2) is given as $\text{dist}((x_1, y_1), (x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$]
[Categorical attributes (Binary) - Hamming Distance: If the value of the two instances is same, the distance d will be equal to 0 otherwise $d = 1$.]
2. Sort the distances in an ascending order and select the first k nearest training data instances to the test instance.
3. Predict the class of the test instance by majority voting (if target attribute is discrete valued) or mean (if target attribute is continuous valued) of the k selected nearest instances.

4.3 Weighted k-Nearest-Neighbor Algorithm

The weighted KNN is an extension of k-NN. It chooses the neighbors by using the weighted distance. In weighted kNN, the nearest k points are given a weight using a function called as the kernel function. The intuition behind weighted kNN, is to give more weight to the points which are nearby and less weight to the points which are farther away.

Algorithm 4.2: Weighted k-NN

Inputs: Training dataset ' T ', Distance metric ' d ', Weighting function $w(i)$, Test instance ' t ', the number of nearest neighbors ' k '

Output: Predicted class or category

Prediction: For test instance t ,

1. For each instance ' i ' in Training dataset T , compute the distance between the test instance t and every other instance ' i ' using a distance metric (Euclidean distance).
[Continuous attributes - Euclidean distance between two points in the plane with coordinates (x_1, y_1) and (x_2, y_2) is given as $\text{dist}((x_1, y_1), (x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$]
[Categorical attributes (Binary) - Hamming Distance: If the values of two instances are the same, the distance d will be equal to 0. Otherwise $d = 1$.]
2. Sort the distances in the ascending order and select the first ' k ' nearest training data instances to the test instance.
3. Predict the class of the test instance by weighted voting technique (Weighting function $w(i)$) for the k selected nearest instances:
 - Compute the inverse of each distance of the ' k ' selected nearest instances.
 - Find the sum of the inverses.
 - Compute the weight by dividing each inverse distance by the sum. (Each weight is a vote for its associated class).
 - Add the weights of the same class.
 - Predict the class by choosing the class with the maximum vote.

4.4 Nearest Centroid Classifier

The Nearest Centroids algorithm assumes that the centroids in the input feature space are different for each target label. The training data is split into groups by class label, then the centroid for each group of data is calculated. Each centroid is simply the mean value of each of the input variables, so it is also called as Mean Difference classifier. If there are two classes, then two centroids or points are calculated; three classes give three centroids, and so on.

Algorithm 4.3: Nearest Centroid Classifier

Inputs: Training dataset T , Distance metric d , Test instance t

Output: Predicted class or category

1. Compute the mean/centroid of each class.
2. Compute the distance between the test instance and mean/centroid of each class (Euclidean Distance).
3. Predict the class by choosing the class with the smaller distance.

4.5 Locally Weighted Regression (LWR)

Locally Weighted Regression (LWR) is a non-parametric supervised learning algorithm that performs local regression by combining regression model with nearest neighbor's model. LWR is also referred to as a memory-based method as it requires training data while prediction but uses only the training data instances locally around the point of interest. Using nearest neighbors algorithm, we find the instances that are closest to a test instance and fit linear function to each of those ' k ' nearest instances in the local regression model. The key idea is that we need to approximate the linear functions of all ' k ' neighbors that minimize the error such that the prediction line is no more linear but rather it is a curve.

Ordinary linear regression finds out a linear relationship between the input x and the output y .
 Given training dataset T ,
 Hypothesis function $h_\beta(x)$, the predicted target output is a linear function where β_0 is the intercept and β_1 is the coefficient of x .
 It is given in Eq. (4.1) as,

$$h_\beta(x) = \beta_0 + \beta_1 x \quad (4.1)$$

The cost function is such that it minimizes the error difference between the predicted value $h_\beta(x)$ and true value ' y ' and it is given as in Eq. (4.2).

$$J(\beta) = \frac{1}{2} \sum_{i=1}^m (h_\beta(x_i) - y_i)^2 \quad (4.2)$$

where ' m ' is the number of instances in the training dataset.

Now the cost function is modified for locally weighted linear regression including the weights only for the nearest neighbor points. Hence, the cost function is given as in Eq. (4.3).

$$J(\beta) = \frac{1}{2} \sum_{i=1}^m w_i (h_\beta(x_i) - y_i)^2 \quad (4.3)$$

where w_i is the weight associated with each x_i .

The weight function used is a Gaussian kernel that gives a higher value for instances that are close to the test instance, and for instances far away, it tends to zero but never equals to zero. w_i is computed in Eq. (4.4) as,

$$w_i = e^{-\frac{(x_i - x)^2}{2r^2}} \quad (4.4)$$

Where, r is called the bandwidth parameter and controls the rate at which w_i reduces to zero with distance from x_i .

MODULE 3

CHAPTER 5

REGRESSION ANALYSIS

1.1 Introduction to Regression

Regression analysis is a fundamental concept that consists of a set of machine learning methods that predict a continuous outcome variable (y) based on the value of one or multiple predictor variables (x).

OR

Regression analysis is a statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables.

Regression is a supervised learning technique which helps in finding the correlation between variables.

It is mainly used for prediction, forecasting, time series modelling, and determining the causal-effect relationship between variables.

Regression shows a line or curve that passes through all the data points on target-predictor graph in such a way that the vertical distance between the data points and the regression line is minimum." The distance between data points and line tells whether a model has captured a strong relationship or not.

- Function of regression analysis is given by:

$$Y=f(x)$$

Here, y is called dependent variable and x is called independent variable.

Applications of Regression Analysis

- Sales of a goods or services
- Value of bonds in portfolio management
- Premium on insurance companies
- Yield of crop in agriculture
- Prices of real estate

1.2 INTRODUCTION TO LINEARITY, CORRELATION AND CAUSATION

A correlation is the statistical summary of the relationship between two sets of variables. It is a core part of data exploratory analysis, and is a critical aspect of numerous advanced machine learning techniques.

Correlation between two variables can be found using a **scatter** plot

There are different types of correlation:

Positive Correlation: Two variables are said to be positively correlated when their values move in the same direction. For example, in the image below, as the value for X increases, so does the value for Y at a constant rate.

Negative Correlation: Finally, variables X and Y will be negatively correlated when their values change in opposite directions, so here as the value for X increases, the value for Y decreases at a constant rate.

Neutral Correlation: No relationship in the change of variables X and Y. In this case, the values are completely random and do not show any sign of correlation, as shown in the following image:

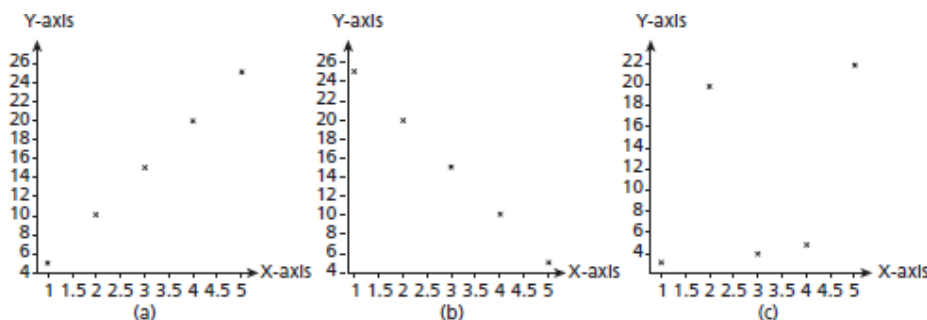


Figure 5.1: Examples of (a) Positive Correlation (b) Negative Correlation (c) Random Points with No Correlation

Causation

Causation is about relationship between two variables as x causes y. This is called x implies y. Regression is different from causation. Causation indicates that one event is the result of the occurrence of the other event; i.e. there is a causal relationship between the two events.

Linear and Non-Linear Relationships

The relationship between input features (variables) and the output (target) variable is fundamental. These concepts have significant implications for the choice of algorithms, model complexity, and predictive performance.

Linear relationship creates a straight line when plotted on a graph, a Non-Linear relationship does not create a straight line but instead creates a curve.

Example:

Linear-the relationship between the hours spent studying and the grades obtained in a class.

Non-Linear-

Linearity:

Linear Relationship: A linear relationship between variables means that a change in one variable is associated with a proportional change in another variable. Mathematically, it can be represented as $y = a * x + b$, where y is the output, x is the input, and a and b are constants.

Linear Models: Goal is to find the best-fitting line (plane in higher dimensions) to the data points. Linear models are interpretable and work well when the relationship between variables is close to being linear.

Limitations: Linear models may perform poorly when the relationship between variables is non-linear. In such cases, they may underfit the data, meaning they are too simple to capture the underlying patterns.

Non-Linearity:

Non-Linear Relationship: A non-linear relationship implies that the change in one variable is not proportional to the change in another variable. Non-linear relationships can take various forms, such as quadratic, exponential, logarithmic, or arbitrary shapes.

Non-Linear Models: Machine learning models like decision trees, random forests, support vector machines with non-linear kernels, and neural networks can capture non-linear relationships. These models are more flexible and can fit complex data patterns.

Benefits: Non-linear models can perform well when the underlying relationships in the data are complex or when interactions between variables are non-linear. They have the capacity to capture intricate patterns.

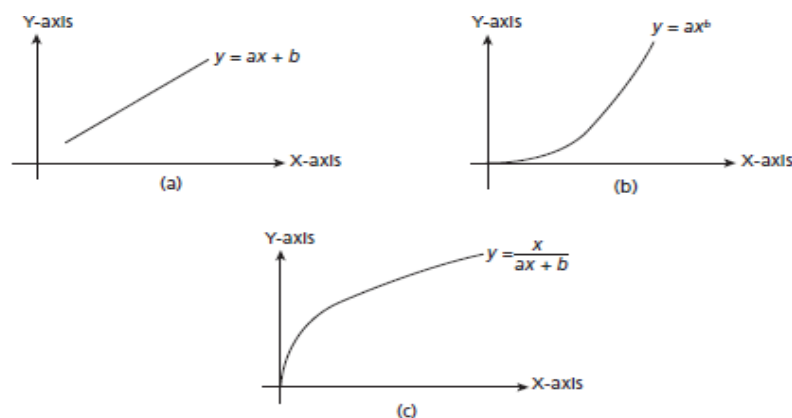


Figure 5.2: (a) Example of Linear Relationship of the Form $y = ax + b$ (b) Example of a Non-linear Relationship of the Form $y = ax^b$ (c) Examples of a Non-linear Relationship $y = \frac{x}{ax + b}$

Types of Regression

Types of Regression Methods

The classification of regression methods is shown in Figure 5.3.

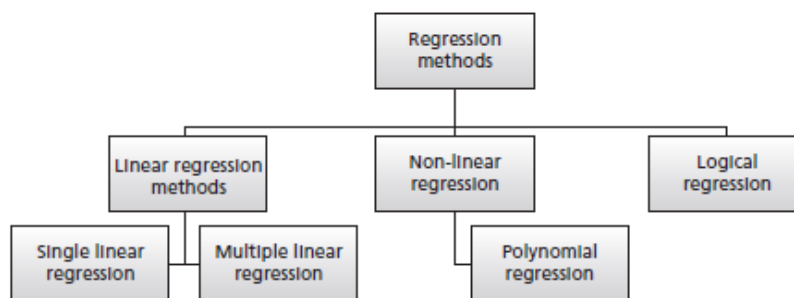


Figure 5.3: Types of Regression Methods

Linear Regression:

Single Independent Variable: Linear regression, also known as simple linear regression, is used when there is a single independent variable (predictor) and one dependent variable (target).

Equation: The linear regression equation takes the form: $Y = \beta_0 + \beta_1 X + \epsilon$, where Y is the dependent variable, X is the independent variable, β_0 is the intercept, β_1 is the slope (coefficient), and ϵ is the error term.

Purpose: Linear regression is used to establish a linear relationship between two variables and make predictions based on this relationship. It's suitable for simple scenarios where there's only one predictor.

Multiple Regression:

Multiple Independent Variables: Multiple regression, as the name suggests, is used when there are two or more independent variables (predictors) and one dependent variable (target).

Equation: The multiple regression equation extends the concept to multiple predictors: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \epsilon$, where Y is the dependent variable, $X_1, X_2, \dots, X_n$ are the independent variables, β_0 is the intercept, $\beta_1, \beta_2, \dots, \beta_n$ are the coefficients, and ϵ is the error term.

Purpose: Multiple regression allows you to model the relationship between the dependent variable and multiple predictors simultaneously. It's used when there are multiple factors that may influence the target variable, and you want to understand their combined effect and make predictions based on all these factors.

Polynomial Regression:

Use: Polynomial regression is an extension of multiple regression used when the relationship between the independent and dependent variables is non-linear.

Equation: The polynomial regression equation allows for higher-order terms, such as quadratic or cubic terms: $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \dots + \beta_n X^n + \epsilon$. This allows the model to fit a curve rather than a straight line.

Logistic Regression:

Use: Logistic regression is used when the dependent variable is binary (0 or 1). It models the probability of the dependent variable belonging to a particular class.

Equation: Logistic regression uses the logistic function (sigmoid function) to model probabilities: $P(Y=1) = 1 / (1 + e^{(-z)})$, where z is a linear combination of the independent variables: $z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$. It transforms this probability into a binary outcome.

Lasso Regression (L1 Regularization):

Use: Lasso regression is used for feature selection and regularization. It penalizes the absolute values of the coefficients, which encourages sparsity in the model.

Objective Function: Lasso regression adds an L1 penalty to the linear regression loss function: $\text{Lasso} = \text{RSS} + \lambda \sum |\beta_i|$, where RSS is the residual sum of squares, λ is the regularization strength, and $|\beta_i|$ represents the absolute values of the coefficients.

Ridge Regression (L2 Regularization):

Use: Ridge regression is used for regularization to prevent overfitting in multiple regression. It penalizes the square of the coefficients.

Objective Function: Ridge regression adds an L2 penalty to the linear regression loss function: $\text{Ridge} = \text{RSS} + \lambda \sum (\beta_i^2)$, where RSS is the residual sum of squares, λ is the regularization strength, and (β_i^2) represents the square of the coefficients.

Limitations of Regression

1. **Outliers** – Outliers are abnormal data. It can bias the outcome of the regression model, as outliers push the regression line towards it.
2. **Number of cases** – The ratio of independent and dependent variables should be at least 20 : 1. For every explanatory variable, there should be at least 20 samples. Atleast five samples are required in extreme cases.
3. **Missing data** – Missing data in training data can make the model unfit for the sampled data.
4. **Multicollinearity** – If exploratory variables are highly correlated (0.9 and above), the regression is vulnerable to bias. Singularity leads to perfect correlation of 1. The remedy is to remove exploratory variables that exhibit correlation more than 1. If there is a tie, then the tolerance ($1 - R^2$) is used to eliminate variables that have the greatest value.

1.3 INTRODUCTION TO LINEAR REGRESSION

Linear regression model can be created by fitting a line among the scattered data points. The line is of the form:

The diagram shows the linear regression equation $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$. Labels with arrows point to each term: 'Dependent Variable' points to Y_i , 'Population Y intercept' points to β_0 , 'Population Slope Coefficient' points to β_1 , 'Independent Variable' points to X_i , and 'Random Error term' points to ϵ_i . A blue bracket under $\beta_0 + \beta_1 X_i$ is labeled 'Linear component', and a blue bracket under ϵ_i is labeled 'Random Error component'.

The assumptions of linear regression are listed as follows:

1. The observations (y) are random and are mutually independent.
2. The difference between the predicted and true values is called an error. The error is also mutually independent with the same distributions such as normal distribution with zero mean and constant variables.
3. The distribution of the error term is independent of the joint distribution of explanatory variables.
4. The unknown parameters of the regression models are constants.

Ordinary Least Square Approach

The ordinary least squares (OLS) algorithm is a method for estimating the parameters of a linear regression model. **Aim:** To find the values of the linear regression model's parameters (i.e., the coefficients) that minimize the sum of the squared residuals.

In mathematical terms, this can be written as: **Minimize** $\sum (y_i - \hat{y}_i)^2$

where y_i is the actual value, $\hat{y}_i$ is the predicted value.

A linear regression model used for determining the value of the response variable, $\hat{y}$, can be represented as the following equation.

$$y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n + e$$

- where: y - is the dependent variable, b_0 is the intercept, e is the error term
- $b_1, b_2, \dots, b_n$ are the coefficients of the independent variables $x_1, x_2, \dots, x_n$

The coefficients $b_1, b_2, \dots, b_n$ can also be called the **coefficients of determination**. The goal of the OLS method can be used to estimate the unknown parameters (**$b_1, b_2, \dots, b_n$**) by minimizing the sum of squared residuals (RSS). The sum of squared residuals is also termed the sum of squared error (SSE).

This method is also known as the **least-squares method** for regression or linear regression.

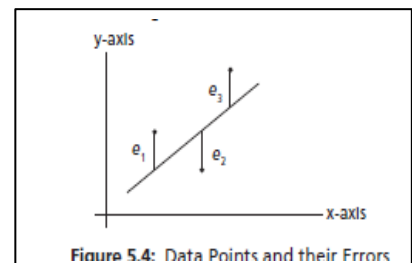
Mathematically the line of equations for points are:

$$y_1 = (a_0 + a_1x_1) + e_1$$

$$y_2 = (a_0 + a_1x_2) + e_2 \quad \text{and so on}$$

$$\dots y_n = (a_0 + a_1x_n) + e_n.$$

$$\text{In general } e_i = y_i - (a_0 + a_1x_i)$$



Here, the terms $(e_1, e_2, \dots, e_n)$ are error associated with the data points and denote the difference between the true value of the observation and the point on the line. This is also called as residuals. The residuals can be positive, negative or zero.

A regression line is the line of best fit for which the sum of the squares of residuals is minimum. The minimization can be done as minimization of individual errors by finding the parameters a_0 and a_1 such that:

$$E = \sum_{i=1}^n e_i = \sum_{i=1}^n (y_i - (a_0 + a_1 x_i)) \quad (5.5)$$

Or as the minimization of sum of absolute values of the individual errors:

$$E = \sum_{i=1}^n |e_i| = \sum_{i=1}^n |(y_i - (a_0 + a_1 x_i))| \quad (5.6)$$

Or as the minimization of the sum of the squares of the individual errors:

$$E = \sum_{i=1}^n (e_i)^2 = \sum_{i=1}^n (y_i - (a_0 + a_1 x_i))^2 \quad (5.7)$$

Sum of the squares of the individual errors, often preferred as individual errors (positive and negative errors), do not get cancelled out and are always positive, and sum of squares results in a large increase even for a small change in the error. Therefore, this is preferred for linear regression.

Therefore, linear regression is modelled as a minimization function as follows:

$$\begin{aligned} J(a_1, a_0) &= \sum_{i=1}^n [y_i - f(x_i)]^2 \\ &= \sum_{i=1}^n [y_i - (a_0 + a_1 x_i)]^2 \end{aligned} \quad (5.8)$$

Here, $J(a_0, a_1)$ is the criterion function of parameters a_0 and a_1 . This needs to be minimized. This is done by differentiating and substituting to zero. This yields the coefficient values of a_0 and a_1 . The values of estimates of a_0 and a_1 are given as follows:

$$a_1 = \frac{(\overline{xy}) - (\bar{x})(\bar{y})}{(\overline{x^2}) - (\bar{x})^2} \quad (5.9)$$

And the value of a_0 is given as follows:

$$a_0 = (\bar{y}) - a_1 \times \bar{x} \quad (5.10)$$

Let us consider a simple problem to illustrate the usage of the above concept.

Linear Regression Example

Example 5.1: Let us consider an example where the five weeks' sales data (in Thousands) is given as shown below in Table 5.1. Apply linear regression technique to predict the 7th and 9th month sales.

Table 5.1: Sample Data

x_i (Week)	y_i (Sales in Thousands)
1	1.2
2	1.8
3	2.6
4	3.2
5	3.8

Table 5.2: Computation Table

x_i	y_i	$(x_i)^2$	$x_i \times y_i$
1	1.2	1	1.2
2	1.8	4	3.6
3	2.6	9	7.8
4	3.2	16	12.8
5	3.8	25	19
Sum = 15	Sum = 12.6	Sum = 55	Sum = 44.4
Average of (x_i) $= \bar{x} = \frac{15}{5}$ $= 3$	Average of (y_i) $= \bar{y} = \frac{12.6}{5}$ $= 2.52$	Average of (x_i^2) $= \overline{x_i^2} = \frac{55}{5}$ $= 11$	Average of $(x_i \times y_i)$ $= \overline{xy} = \frac{44.4}{5}$ $= 8.88$

Let us compute the slope and intercept now using Eq. (5.9) as:

$$a_1 = \frac{8.88 - 3(2.52)}{11 - 3^2} = 0.66$$

$$a_0 = 2.52 - 0.66 \times 3 = 0.54$$

The fitted line is shown in Figure 5.5.

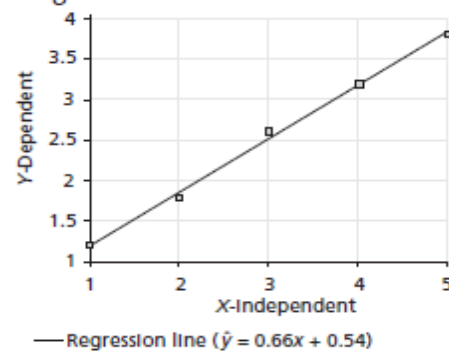


Figure 5.5: Linear Regression Model Constructed

Let us model the relationship as $y = a_0 + a_1 \times x$. Therefore, the fitted line for the above data is:
 $y = 0.54 + 0.66 \times x$.

The predicted 7th week sale would be (when $x = 7$), $y = 0.54 + 0.66 \times 7 = 5.16$ and the 12th month, $y = 0.54 + 0.66 \times 12 = 8.46$. All sales are in thousands.

Linear Regression in Matrix Form

Linear Regression in Matrix Form

Matrix notations can be used for representing the values of independent and dependent variables. This is illustrated through Example 5.2.

The Eq. (5.3) can be written in the form of matrix as follows:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & \vdots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix} \quad (5.11)$$

This can be written as:

$Y = Xa + e$, where X is an $n \times 2$ matrix, Y is an $n \times 1$ vector, a is a 2×1 column vector and e is an $n \times 1$ column vector.

Example 5.2: Find linear regression of the data of week and product sales (in Thousands) given in Table 5.3. Use linear regression in matrix form.

Table 5.3: Sample Data for Regression

x_i (Week)	y_i (Product Sales in Thousands)
1	1
2	3
3	4
4	8

Solution: Here, the dependent variable X is be given as:

$$x^T = [1 \ 2 \ 3 \ 4]$$

And the independent variable is given as follows:

$$y^T = [1 \ 3 \ 4 \ 8]$$

The data can be given in matrix form as follows:

$$X = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{pmatrix}. \text{ The first column can be used for setting bias.}$$

$$\text{and } Y = \begin{pmatrix} 1 \\ 3 \\ 4 \\ 8 \end{pmatrix}$$

The regression is given as:

$$a = ((X^T X)^{-1} X^T) Y$$

The computation order of this equation is shown step by step as:

1. Computation of $(X^T X) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{pmatrix} \times \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{pmatrix} = \begin{pmatrix} 4 & 10 \\ 10 & 30 \end{pmatrix}$
2. Computation of matrix inverse of $(X^T X)^{-1} = \begin{pmatrix} 4 & 10 \\ 10 & 30 \end{pmatrix}^{-1} = \begin{pmatrix} 1.5 & -0.5 \\ -0.5 & 0.2 \end{pmatrix}$
3. Computation of $((X^T X)^{-1} X^T) = \begin{pmatrix} 1.5 & -0.5 \\ -0.5 & 0.2 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0.5 & 0 & -0.5 \\ -0.3 & -0.1 & 0.1 & 0.3 \end{pmatrix}$
4. Finally, $((X^T X)^{-1} X^T) Y = \begin{pmatrix} 1 & 0.5 & 0 & -0.5 \\ -0.3 & -0.1 & 0.1 & 0.3 \end{pmatrix} \times \begin{pmatrix} 1 \\ 3 \\ 4 \\ 8 \end{pmatrix} = \begin{pmatrix} -1.5 \\ 2.2 \end{pmatrix} = \begin{pmatrix} \text{Intercept} \\ \text{slope} \end{pmatrix}$

Thus, the substitution of values in Eq. (5.11) using the previous steps yields the fitted line as $2.2x - 1.5$.

of determination r^2 is the ratio of the explained and unexplained variations.

Standard Error

Residuals or error is the difference between the actual (y) and predicted value ($\hat{y}$).

If the residuals have normal distribution, then the mean is zero and hence it is desirable. This is a measure of variability in finding the coefficients. It is preferable that the error be less than the coefficient estimate. The standard deviation of residuals is called residual standard error. If it is zero, then it means that the model fits the data correctly.

Mean Absolute Error (MAE)

MAE is the mean of residuals. It is the difference between estimated or predicted target value and actual target incomes. It can be mathematically defined as follows:

$$MAE = \frac{1}{n} \sum_{i=0}^{n-1} |y_i - \hat{y}_i| \quad (5.12)$$

Here, $\hat{y}$ is the estimated or predicted target output and y is the actual target output, and n is the number of samples used for regression analysis.

Mean Squared Error (MSE)

It is the sum of square of residuals. This value is always positive and closer to 0. This is given mathematically as:

$$\frac{1}{n} \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2 \quad (5.13)$$

CHAPTER 5
REGRESSION ANALYSIS

2 Consider the following dataset in Table 5.11 where the week and number of working hours per week spent by a research scholar in a library are tabulated. Based on the dataset, predict the number of hours that will be spent by the research scholar in the 7th and 9th week. Apply Linear regression model.

Table 5.11

x_i (week)	1	2	3	4	5
y_i (Hours Spent)	12	18	22	28	35

Solution

The computation table is shown below:

x_i	y_i	$x_i \times x_i$	$x_i \times y_i$
1	12	1	12
2	18	4	36
3	22	9	66
4	28	16	112
5	35	25	175
Sum = 15 avg(x_i)=15/5=3	Sum = 115 avg(y_i)=115/5=23	Avg ($x_i \times x_i$)=55/5=11	Avg($x_i \times y_i$)=401/5=80.2

The regression Equations are

$$a_1 = \frac{(\overline{xy}) - (\bar{x})(\bar{y})}{(\overline{x^2}) - (\bar{x})^2}$$

$$a_0 = (\bar{y}) - a_1 \times \bar{x}$$

$$a_1 = \frac{80.2 - 3(23)}{11 - 3^2} = \frac{80.2 - 69}{11 - 9} = \frac{11.2}{2} = 5.6$$

$$a_0 = 23 - 5.6 \times 3 = 23 - 16.8 = 6.2$$

Therefore, the regression equation is given as

$$y = 5.6 + 6.2 \times x$$

The prediction for the 7th week hours spent by the research scholar will be

$$y = 5.6 + 6.2 \times 7 = 49 \text{ hours}$$

The prediction for the 9th week hours spent by the research scholar will be

$$y = 5.6 + 6.2 \times 9 = 61.4 \approx 61 \text{ hours}$$

3 The height of boys and girls is given in the following Table 5.12.

Table 5.12: Sample Data

Height of Boys	65	70	75	78
Height of Girls	63	67	70	73

Fit a suitable line of best fit for the above data.

Solution

The computation table is shown below:

x_i	y_i	$x_i \times x_i$	$x_i \times y_i$
65	63	4225	4095
70	67	4900	4690
75	70	5625	5250
78	73	6084	5694
Sum = 288 Mean(x_i)=288/4=72	Sum = 273 Mean(y_i)=273/4=68.25	Avg ($x_i \times x_i$)=20834/4=5208.5	Avg($x_i \times y_i$)=19729/4=4932.25

The regression Equations are

$$a_1 = \frac{(\overline{xy}) - (\bar{x})(\bar{y})}{(\overline{x^2}) - (\bar{x})^2}$$

$$a_0 = (\bar{y}) - a_1 \times \bar{x}$$

$$a_1 = \frac{4932.25 - 72(68.25)}{5208.5 - 72^2} = \frac{18.25}{24.5} = 0.7449$$

$$a_0 = 68.25 - 0.7449 \times 72 = 68.25 - 53.6328 = 14.6172$$

Therefore, the regression line of best fit is given as

$$y = 0.7449 + 14.6172 \times x$$

4 Using multiple regression, fit a line for the following dataset shown in Table 5.13. Here, Z is the equity, X is the net sales and Y is the asset. Z is the dependent variable and X and Y are independent variables. All the data is in million dollars.

Table 5.13: Sample Data

Z	X	Y
4	12	8
6	18	12
7	22	16
8	28	36
11	35	42

Solution

The matrix X and Y is given as follows:

$$X = \begin{pmatrix} 1 & 12 & 8 \\ 1 & 18 & 12 \\ 1 & 22 & 16 \\ 1 & 28 & 36 \\ 1 & 35 & 42 \end{pmatrix}$$

$$Y = \begin{pmatrix} 4 \\ 6 \\ 7 \\ 8 \\ 11 \end{pmatrix}$$

The regression coefficients can be found as follows

$$\hat{a} = ((X^T X)^{-1} X^T) Y$$

Substituting the values one get,

$$\hat{a} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 12 & 18 & 22 & 28 & 35 \\ 8 & 12 & 16 & 36 & 42 \end{pmatrix} \times \begin{pmatrix} 1 & 12 & 8 \\ 1 & 18 & 12 \\ 1 & 22 & 16 \\ 1 & 28 & 36 \\ 1 & 35 & 42 \end{pmatrix}^{-1} \times \begin{pmatrix} 1 & 12 & 8 \\ 1 & 18 & 12 \\ 1 & 22 & 16 \\ 1 & 28 & 36 \\ 1 & 35 & 42 \end{pmatrix}^T \times \begin{pmatrix} 4 \\ 6 \\ 7 \\ 8 \\ 11 \end{pmatrix}$$

$$= \begin{pmatrix} 5 & 115 & 114 \\ 115 & 2961 & 3142 \\ 114 & 3142 & 3524 \end{pmatrix}^{-1} \times \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 12 & 18 & 22 & 28 & 35 \\ 8 & 12 & 16 & 36 & 42 \end{pmatrix} \times \begin{pmatrix} 4 \\ 6 \\ 7 \\ 8 \\ 11 \end{pmatrix}$$

$$= \begin{pmatrix} -0.4135 \\ 0.39625 \\ -0.0658 \end{pmatrix}$$

Therefore, the regression line is given as

$$y = 0.39625x_1 - 0.0658x_2 - 0.4135$$

5.5 MULTIPLE LINEAR REGRESSION

Multiple regression model involves multiple predictors or independent variables and one dependent variable. This is an extension of the linear regression problem. The basic assumptions of multiple linear regression are that the independent variables are not highly correlated and hence multicollinearity problem does not exist. Also, it is assumed that the residuals are normally distributed.

For example, the multiple regression of two variables x_1 and x_2 is given as follows:

$$\begin{aligned} y &= f(x_1, x_2) \\ &= a_0 + a_1x_1 + a_2x_2 \end{aligned} \quad (5.21)$$

In general, this is given for ' n ' independent variables as:

$$\begin{aligned} y &= f(x_1, x_2, x_3, \dots, x_n) \\ &= a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n + \varepsilon \end{aligned} \quad (5.22)$$

Here, $(x_1, x_2, \dots, x_n)$ are predictor variables, y is the dependent variable, $(a_0, a_1, \dots, a_n)$ are the coefficients of the regression equation and ε is the error term. This is illustrated through Example 5.5.



Example 5.5: Apply multiple regression for the values given in Table 5.7 where weekly sales along with sales for products x_1 and x_2 are provided. Use matrix approach for finding multiple regression.

Table 5.7: Sample Data

x_1 (Product One Sales)	x_2 (Product Two Sales)	y Output Weekly Sales (in Thousands)
1	4	1
2	5	6
3	8	8
4	2	12

Solution: Here, the matrices for Y and X are given as follows:

$$X = \begin{pmatrix} 1 & 1 & 4 \\ 1 & 2 & 5 \\ 1 & 3 & 8 \\ 1 & 4 & 2 \end{pmatrix} \text{ and } Y = \begin{pmatrix} 1 \\ 6 \\ 8 \\ 12 \end{pmatrix}.$$

The coefficient of the multiple regression equation is given as $a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}$.

The regression coefficient for multiple regression is calculated the same way as linear regression:

$$\hat{a} = ((X^T X)^{-1} X^T) Y \quad (5.23)$$

Using Eq. (5.23), and substituting the values (Similar to Problem 5.2), one gets $\hat{a}$ as:

$$\begin{aligned} &= \left(\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{pmatrix} \begin{pmatrix} 1 & 1 & 4 \\ 1 & 2 & 5 \\ 1 & 3 & 8 \\ 1 & 4 & 2 \end{pmatrix} \right)^{-1} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 6 \\ 8 \\ 12 \end{pmatrix} \\ &= \begin{pmatrix} -1.69 \\ 3.48 \\ -0.05 \end{pmatrix} \end{aligned}$$

Here, the coefficients are $a_0 = -1.69$, $a_1 = 3.48$ and $a_2 = -0.05$. Hence, the constructed model is:
 $y = -1.69 + 3.48x_1 - 0.05x_2$



Polynomial Regression

It can handle non-linear relationships among variables by using n^{th} degree of a polynomial. Instead of applying transforms, polynomial regression can be directly used to deal with different levels of curvilinearity.

Polynomial regression provides a non-linear curve such as quadratic and cubic. For example, the second-degree transformation (called quadratic transformation) is given as: $y = a_0 + a_1x + a_2x^2$ and the third-degree polynomial is called cubic transformation given as: $y = a_0 + a_1x + a_2x^2 + a_3x^3$. Generally, polynomials of maximum degree 4 are used, as higher order polynomials take some strange shapes and make the curve more flexible. It leads to a situation of overfitting and hence is avoided.

Let us consider a polynomial of 2nd degree. Given points $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, the objective is to fit a polynomial of degree 2. The polynomial of degree 2 is given as:

$$y = a_0 + a_1x + a_2x^2 \quad (5.26)$$

Such that the error $E = \sum_{i=1}^n [y_i - (a_0 + a_1x_i + a_2x_i^2)]^2$ is minimized. The coefficients a_0, a_1, a_2 of Eq. (5.26) can be obtained by taking partial derivatives with respect to each of the coefficients as $\frac{\partial E}{\partial a_0}, \frac{\partial E}{\partial a_1}, \frac{\partial E}{\partial a_2}$ and substituting it with zero. This results in 2 + 1 equations given as follows:

$$\begin{aligned} na_0 + \left(\sum_{i=1}^n x_i \right) a_1 + \left(\sum_{i=1}^n x_i^2 \right) a_2 &= \sum_{i=1}^n y_i \\ \left(\sum_{i=1}^n x_i \right) a_0 + \left(\sum_{i=1}^n x_i^2 \right) a_1 + \left(\sum_{i=1}^n x_i^3 \right) a_2 &= \sum_{i=1}^n x_i y_i \\ \left(\sum_{i=1}^n x_i^2 \right) a_0 + \left(\sum_{i=1}^n x_i^3 \right) a_1 + \left(\sum_{i=1}^n x_i^4 \right) a_2 &= \sum_{i=1}^n x_i^2 y_i \end{aligned} \quad (5.27)$$

The best line is the line that minimizes the error between line and data points. Arranging the coefficients of the above equation in the matrix form results in:

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum (x_i, y_i) \\ \sum (x_i^2, y_i) \end{bmatrix} \quad (5.28)$$

This is of the form $Xa = B$. One can solve this equation for a as:

$$a = X^{-1}B \quad (5.29)$$

Table 5.8: Sample Data

x	y
1	1
2	4
3	9
4	15

Solution: For applying polynomial regression, computation is done as shown in Table 5.9.

Here, the order is 2 and the sample i ranges from 1 to 4.

Table 5.9: Computation Table

x_i	y_i	$x_i y_i$	x_i^2	$x_i^2 y_i$	x_i^3	x_i^4
1	1	1	1	1	1	1
2	4	8	4	16	8	16
3	9	27	9	81	27	81
4	15	60	16	240	64	256
$\Sigma x_i = 10$	$\Sigma y_i = 29$	$\Sigma x_i y_i = 96$	$\Sigma x_i^2 = 30$	$\Sigma x_i^2 y_i = 338$	$\Sigma x_i^3 = 100$	$\Sigma x_i^4 = 354$

It can be noted that, $N = 4$, $\Sigma y_i = 29$, $\Sigma x_i y_i = 96$, $\Sigma x_i^2 y_i = 338$. When the order is 2, the matrix using Eq. (5.28) is given as follows:

$$\begin{bmatrix} 4 & 10 & 30 \\ 10 & 30 & 100 \\ 30 & 100 & 354 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 29 \\ 96 \\ 338 \end{bmatrix}$$

Therefore, using Eq. (5.29), one can get coefficients as:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 4 & 10 & 30 \\ 10 & 30 & 100 \\ 30 & 100 & 354 \end{bmatrix}^{-1} \times \begin{bmatrix} 29 \\ 96 \\ 338 \end{bmatrix} = \begin{bmatrix} -0.75 \\ 0.95 \\ 0.75 \end{bmatrix}.$$

This leads to the regression equation using Eq. (5.26) as:

$$y = -0.75 + 0.95x + 0.75x^2$$

5.7 LOGISTIC REGRESSION

Linear regression predicts the numerical response but is not suitable for predicting the categorical variables. When categorical variables are involved, it is called classification problem. Logistic regression is suitable for binary classification problem. Here, the output is often a categorical variable. For example, the following scenarios are instances of predicting categorical variables.

1. Is the mail spam or not spam? The answer is yes or no. Thus, categorical dependant variable is a binary response of yes or no.
2. If the student should be admitted or not is based on entrance examination marks. Here, categorical variable response is admitted or not.
3. The student being pass or fail is based on marks secured.

Thus, logistic regression is used as a binary classifier and works by predicting the probability of the categorical variable. In general, it takes one or more features x and predicts the response y . If the probability is predicted via linear regression, it is given as:

$$p(x) = a_0 + a_1x$$

Hence, logistic regression tries to model the probability of the particular response variable. In email classification problem, say normal email or spam, if the probability of the response variable is 0.7, then there is a 70% possibility of a normal mail.

Linear regression generated value is in the range $-\infty$ to $+\infty$, whereas the probability of the response variable ranges between 0 and 1. Hence, there must be a mapping function to map the value $-\infty$ to $+\infty$ to 0–1. The core of the mapping function in logistic regression method is sigmoidal function. A sigmoidal function is a 'S' shaped function that yields values between 0 and 1. This is known as logit function. This is mathematically represented as:

$$\text{logit}(x) = \frac{1}{1 + e^{-x}} \quad (5.30)$$

Here, x is the independent variable and e is the Euler number. The purpose of the logit function is to map any real number to zero or 1.

Logistic regression can be viewed as an extension of linear regression, but the only difference is that the output of linear regression can be an extremely high number. This needs to be mapped into the range 0–1, as probability can have values only in the range 0–1. This problem is solved using log odd or logit functions. What is the difference between odds and probability? Odds and probability (or likelihood) are two sides of a coin and represent uncertainty. The odds are defined as the ratio of the probability of an event and probability of an event that is not happening. This is given as:

$$\text{odd} = \frac{\text{probability of an event}}{\text{probability of an non-event}} = \frac{p}{1-p} \quad (5.31)$$

Log-odds can be taken for the odds, resulting in:

$$\log\left(\frac{p(x)}{1-p(x)}\right) = a_0 + a_1x \quad (5.32)$$

Here, $\log(.)$ is a logit function or log odds function. One can solve for $p(x)$ by taking the inverse of the above function as:

$$p(x) = \frac{\exp(a_0 + a_1x)}{1 + \exp(a_0 + a_1x)} \quad (5.33)$$

This is the same sigmoidal function. It always gives the value in the range 0–1. Dividing the numerator and denominator by the numerator, one gets:

$$p(x) = \frac{1}{1 + \exp(-a_0 - a_1x)} \quad (5.34)$$

Example 5.7: Let us assume a binomial logistic regression problem where the classes are pass and fail. The student dataset has entrance mark based on the historic data of those who are selected or not selected. Based on the logistic regression, the values of the learnt parameters are $a_0 = 1$ and $a_1 = 8$. Assuming marks of $x = 60$, compute the resultant class.

Solution: The values of regression coefficients are $a_0 = 1$ and $a_1 = 8$, and given that $x = 60$.

Based on the regression coefficients, z can be computed as:

$$\begin{aligned} z &= a_0 + a_1 x \\ &= 1 + 8 \times 60 = 481 \end{aligned}$$

One can fit this in a sigmoidal function using Eq. (5.30) to get the probability as:

$$y = \frac{1}{1 + \exp(-481)} = \frac{1}{2.271} = 0.44$$

If we assume the threshold value as 0.5, then it is observed that $0.44 < 0.5$, therefore, the candidate with marks 60 is not selected.

To determine the relationship between dependant and independent variables, parameters need to be obtained. In logistic regression, the parameters are obtained through maximum likelihood function (MLE) using the training data. The aim is to learn the values of parameters of the logistic model (a 's) by minimizing the error in the probability predicted by the model.

There can be many different sets of coefficients available. The optimal value of the parameters is obtained by using the MLE function, which is a set of coefficients for which the probability of getting the observed data is maximum.

If π is the success of the outcome and $1 - \pi$ is the failure of the outcome, then the likelihood function is given as:

$$L(a : y) = \prod_{i=1}^n \left(\frac{\pi_i}{1 - \pi_i} \right)^{y_i} (1 - \pi_i) \quad (5.37)$$

To determine the value of the parameters, the log of the likelihood function is taken. Techniques like Newton method can be used to maximize the log-likelihood of the function.

Logistic regression is suitable for binary classification. The idea can be extended for multiple classes called multinomial logistic regression. Let us assume that there are three classes 1, 2 and 3. Then, the multinomial logistic regression creates three classification problems – class 1 and Not class 1, class 2 and Not class 2, and finally class 3 and Not class 3. Three problems are simultaneously used to find the maximum probability relative to others to get the appropriate class.

Logistic regression is a simple and efficient method for binary classification. The model can be easily interpreted too. The disadvantages of logistic regression are that multinomial logistic regression cannot handle many attributes and can handle only linear features. Also, if all the attributes have multicollinearity problem, the logistic method does not work effectively.

CHAPTER 6

DECISION TREE LEARNING

6.1 Introduction

Decision tree learning model, one of the most popular supervised predictive learning models, classifies data instances with high accuracy and consistency. The model performs an *inductive inference* that reaches a general conclusion from observed examples. This model is variably used for solving complex classification applications.

Decision tree is a concept tree which summarizes the information contained in the training dataset in the form of a tree structure. Once the concept model is built, test data can be easily classified.

- Why called as decision tree ?
 - As starts from root node and finds number of solutions .
- The benefits of having a decision tree are as follows :
 - It does not require any domain knowledge.
 - It is easy to comprehend.
 - The learning and classification steps of a decision tree are simple and fast.
- Example : Toll free number

6.1.1 Structure of a Decision Tree A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

Applies to classification and regression model.

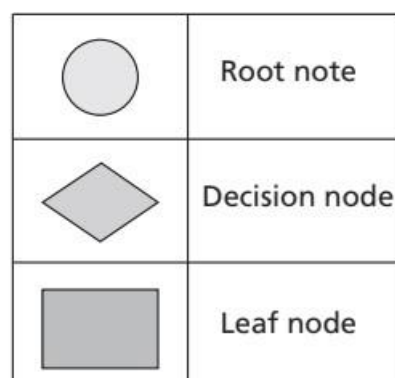


Figure 6.1: Nodes in a Decision Tree

Note: A decision tree is not always a binary tree. It is a tree which can have more than two branches.

The decision tree consists of 2 major procedures:

- 1) Building a tree and
- 2) Knowledge inference or classification.

Building the Tree

Goal Construct a decision tree with the given training dataset. The tree is constructed in a top-down fashion. It starts from the root node. At every level of tree construction, we need to find the best split attribute or best decision node among all attributes. This process is recursive and continued until we end up in the last level of the tree or finding a leaf node which cannot be split further. The tree construction is complete when all the test conditions lead to a leaf node. The leaf node contains the target class or output of classification.

Output Decision tree representing the complete hypothesis space.

Knowledge Inference or Classification

Goal Given a test instance, infer to the target class it belongs to.

Classification Inferring the target class for the test instance or object is based on inductive inference on the constructed decision tree. In order to classify an object, we need to start traversing the tree from the root. We traverse as we evaluate the test condition on every decision node with the test object attribute value and walk to the branch corresponding to the test's outcome. This process is repeated until we end up in a leaf node which contains the target class of the test object.

Output Target label of the test instance.

Advantages of Decision Trees

1. Easy to model and interpret
2. Simple to understand
3. The input and output attributes can be discrete or continuous predictor variables.
4. Can model a high degree of nonlinearity in the relationship between the target variables and the predictor variables
5. Quick to train

Disadvantages of Decision Trees

Some of the issues that generally arise with a decision tree learning are that:

1. It is difficult to determine how deeply a decision tree can be grown or when to stop growing it.
2. If training data has errors or missing attribute values, then the decision tree constructed may become unstable or biased.
3. If the training data has continuous valued attributes, handling it is computationally complex and has to be discretized.
4. A complex decision tree may also be over-fitting with the training data.
5. Decision tree learning is not well suited for classifying multiple output classes.
6. Learning an optimal decision tree is also known to be NP-complete.

6.1.2 Fundamentals of Entropy

- How to draw a decision tree ?

Entropy

Information gain

Entropy is the amount of uncertainty or randomness in the outcome of a random variable or an event. Moreover, entropy describes about the homogeneity of the data instances. The best feature is selected based on the entropy value. For example, when a coin is flipped, head or tail are the two outcomes, hence its entropy is lower when compared to rolling a dice which has got six outcomes.

Higher the entropy → Higher the uncertainty

Lower the entropy → Lower the uncertainty

Let P be the probability distribution of data instances from 1 to n as shown in Eq. (6.2).

$$\text{So, } P = P_1, \dots, P_n \quad (6.2)$$

Entropy of P is the information measure of this probability distribution given in Eq. (6.3),

$$\begin{aligned} \text{Entropy_Info}(P) &= \text{Entropy_Info}(P_1, \dots, P_n) \\ &= -(P_1 \log_2(P_1) + P_2 \log_2(P_2) + \dots + P_n \log_2(P_n)) \end{aligned} \quad (6.3)$$

where, P_1 is the probability of data instances classified as class 1 and P_2 is the probability of data instances classified as class 2 and so on.

$$P_1 = \frac{|\text{No of data instances belonging to class 1}|}{|\text{Total no of data instances in the training dataset}|}$$

Algorithm 6.1: General Algorithm for Decision Trees

Algorithm 6.1: General Algorithm for Decision Trees

1. Find the best attribute from the training dataset using an attribute selection measure and place it at the root of the tree.
2. Split the training dataset into subsets based on the outcomes of the test attribute and each subset in a branch contains the data instances or tuples with the same value for the selected test attribute.
3. Repeat step 1 and step 2 on each subset until we end up in leaf nodes in all the branches of the tree.
4. This splitting process is recursive until the stopping criterion is reached.

Stopping Criteria

The following are some of the common stopping conditions:

1. The data instances are homogenous which means all belong to the same class C_i and hence its entropy is 0.
2. A node with some defined minimum number of data instances becomes a leaf (Number of data instances in a node is between 0.25 and 1.00% of the full training dataset).
3. The maximum tree depth is reached, so further splitting is not done and the node becomes a leaf node.

6.2 DECISION TREE INDUCTION ALGORITHMS

There are many decision tree algorithms, such as ID3, C4.5, CART, CHAID, QUEST, GUIDE, CRUISE, and CTREE, that are used for classification in real-time environment. The most commonly used decision tree algorithms are ID3 (Iterative Dichotomizer 3), developed by J.R Quinlan in 1986, and C4.5 is an advancement of ID3 presented by the same author in 1993. CART, that stands for Classification and Regression Trees, is another algorithm which was developed by Breiman et al. in 1984.

The accuracy of the tree constructed depends upon the selection of the best split attribute. Different algorithms are used for building decision trees which use different measures to decide on the splitting criterion. Algorithms such as ID3, C4.5 and CART are popular algorithms used in the construction of decision trees. The algorithm ID3 uses 'Information Gain' as the splitting criterion whereas the algorithm C4.5 uses 'Gain Ratio' as the splitting criterion. The CART algorithm is popularly used for classifying both categorical and continuous-valued target variables. CART uses GINI Index to construct a decision tree.

6.2.1 ID3 Tree Construction(ID3 stands for Iterative Dichotomiser 3)

A decision tree is one of the most powerful tools of **supervised learning** algorithms used for both **classification** and **regression** tasks.

It builds a flowchart-like tree structure where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal

node) holds a class label. It is **constructed** by recursively splitting the training data into subsets based on the values of the attributes until a stopping criterion is met, such as the maximum depth of the tree or the minimum number of samples required to split a node.

Definitions

Let T be the training dataset.

Let A be the set of attributes $A = [A_1, A_2, A_3, \dots, A_n]$.

Let m be the number of classes in the training dataset.

Let P_i be the probability that a data instance or a tuple ' d ' belongs to class C_i .

It is calculated as,

$$P_i = \frac{\text{Total no of data instances that belongs to class } C_i \text{ in } T}{\text{Total no of tuples in the training set } T} \quad (6.6)$$

Mathematically, it is represented as shown in Eq. (6.7).

$$P_i = \frac{|d_{C_i}|}{|T|} \quad (6.7)$$

$$\text{Entropy_Info}(T) = - \sum_{i=1}^m P_i \log_2 P_i \quad (6.8)$$

$$\text{Entropy_Info}(T, A) = \sum_{i=1}^r \frac{|A_i|}{|T|} \times \text{Entropy_Info}(A_i) \quad (6.9)$$

$$\text{Information_Gain}(A) = \text{Entropy_Info}(T) - \text{Entropy_Info}(T, A) \quad (6.10)$$

Algorithm 6.2: Procedure to Construct a Decision Tree using ID3

1. Compute Entropy_Info Eq. (6.8) for the whole training dataset based on the target attribute.
2. Compute Entropy_Info Eq. (6.9) and Information_Gain Eq. (6.10) for each of the attribute in the training dataset.
3. Choose the attribute for which entropy is minimum and therefore the gain is maximum as the best split attribute.
4. The best split attribute is placed as the root node.
5. The root node is branched into subtrees with each subtree as an outcome of the test condition of the root node attribute. Accordingly, the training dataset is also split into subsets.
6. Recursively apply the same operation for the subset of the training set with the remaining attributes until a leaf node is derived or no more training instances are available in the subset.

6.2.2 C4.5 Construction

C4.5 is a widely used algorithm for constructing decision trees from a dataset.

Disadvantages of ID3 are: Attributes must be nominal values, dataset must not include missing data, and finally the algorithm tend to fall into overfitting.

To overcome this disadvantage Ross Quinlan, inventor of ID3, made some improvements for these bottlenecks and created a new algorithm named C4.5. Now, the algorithm can create a more generalized models including continuous data and could handle missing data. And also works with discrete data, supports post-pruning.

Given a Training dataset T ,

The Split_Info of an attribute A is computed as given in Eq. (6.11):

$$\text{Split_Info}(T, A) = -\sum_{i=1}^v \frac{|A_i|}{|T|} \times \log_2 \frac{|A_i|}{|T|}$$

where, the attribute A has got ' v ' distinct values $\{a_1, a_2, \dots, a_v\}$, and $|A_i|$ is the number of instances for distinct value ' i ' in attribute A .

The Gain_Ratio of an attribute A is computed as given in Eq. (6.12):

$$\text{Gain_Ratio}(A) = \frac{\text{Info_Gain}(A)}{\text{Split_Info}(T, A)}$$

Algorithm 6.3: Procedure to Construct a Decision Tree using C4.5

1. Compute Entropy_Info Eq. (6.8) for the whole training dataset based on the target attribute.
2. Compute Entropy_Info Eq. (6.9), Info_Gain Eq. (6.10), Split_Info Eq. (6.11) and Gain_Ratio Eq. (6.12) for each of the attribute in the training dataset.
3. Choose the attribute for which Gain_Ratio is maximum as the best split attribute.
4. The best split attribute is placed as the root node.
5. The root node is branched into subtrees with each subtree as an outcome of the test condition of the root node attribute. Accordingly, the training dataset is also split into subsets.
6. Recursively apply the same operation for the subset of the training set with the remaining attributes until a leaf node is derived or no more training instances are available in the subset.

Dealing with Continuous Attributes in C4.5

The C4.5 algorithm is further improved by considering attributes which are continuous, and a continuous attribute is discretized by finding a split point or threshold. When an attribute 'A' has numerical values which are continuous, a threshold or best split point 's' is found such that the set of values is categorized into two sets such as $A < s$ and $A \geq s$. The best split point is the attribute value which has maximum information gain for that attribute.

Now, let us consider the set of continuous values for the attribute CGPA in the sample dataset as shown in Table 6.12.

Table 6.12: Sample Dataset

S.No.	CGPA	Job Offer
1.	9.5	Yes
2.	8.2	Yes
3.	9.1	No
4.	6.8	No
5.	8.5	Yes
6.	9.5	Yes
7.	7.9	No
8.	9.1	Yes
9.	8.8	Yes
10.	8.8	Yes

First, sort the values in an ascending order.

6.8	7.9	8.2	8.5	8.8	8.8	9.1	9.1	9.5	9.5
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Remove the duplicates and consider only the unique values of the attribute.

6.8	7.9	8.2	8.5	8.8	8.8	9.1	9.5
-----	-----	-----	-----	-----	-----	-----	-----

Now, compute the Gain for the distinct values of this continuous attribute. Table 6.13 shows the computed values.

Table 6.13: Gain Values for CGPA

	6.8		7.9		8.2		8.5		8.8		9.1		9.5	
Range	$\leq$	$>$	$\leq$	$>$	$\leq$	$>$	$\leq$	$>$	$\leq$	$>$	$\leq$	$>$	$\leq$	$>$
Yes	0	7	0	7	1	6	2	5	4	3	5	2	7	0
No	1	2	2	1	2	1	2	1	2	1	3	0	3	0
Entropy	0	0.7637	0	0.5433	0.9177	0.5913	1	0.6497	0.9177	0.8108	0.9538	0	0.8808	0
Entropy_Info (S, T)	0.6873		0.4346		0.6892		0.7898		0.8749		0.7630		0.8808	
Gain	0.1935		0.4462		0.1916		0.091		0.0059		0.1178		0	

Table 6.14: Discretized Instances

S.No.	CGPA Continuous	CGPA Discretized	Job Offer
1.	9.5	>7.9	Yes
2.	8.2	>7.9	Yes
3.	9.1	>7.9	No
4.	6.8	≤7.9	No
5.	8.5	>7.9	Yes
6.	9.5	>7.9	Yes
7.	7.9	≤7.9	No
8.	9.1	>7.9	Yes
9.	8.8	>7.9	Yes
10.	8.8	>7.9	Yes

6.2.3 Classification and Regression Trees Construction

Classification and Regression Trees (CART) is a widely used algorithm for constructing decision trees that can be applied to both classification and regression tasks. CART is similar to C4.5 but has some differences in its construction and splitting criteria.

The classification method CART is required to construct a decision tree based on Gini's impurity index. It serves as an example of how the values of other variables can be used to predict the values of a target variable. It functions as a fundamental machine-learning method and provides a wide range of use cases

subset. For example, if an attribute A has three distinct values say $\{a_1, a_2, a_3\}$, the possible subsets are $\{\}$, $\{a_1\}$, $\{a_2\}$, $\{a_3\}$, $\{a_1, a_2\}$, $\{a_1, a_3\}$, $\{a_2, a_3\}$, and $\{a_1, a_2, a_3\}$. So, if an attribute has 3 distinct values, the number of possible subsets is 2^3 , which means 8. Excluding the empty set $\{\}$ and the full set $\{a_1, a_2, a_3\}$, we have 6 subsets. With 6 subsets, we can form three possible combinations such as:

$\{a_1\}$ with $\{a_2, a_3\}$

$\{a_2\}$ with $\{a_1, a_3\}$

$\{a_3\}$ with $\{a_1, a_2\}$

Hence, in this CART algorithm, we need to compute the best splitting attribute and the best split subset i in the chosen attribute.

Higher the GINI value, higher is the homogeneity of the data instances.

Gini_Index(T) is computed as given in Eq. (6.13).

$$\text{Gini_Index}(T) = 1 - \sum_{i=1}^w P_i^2 \quad (6.13)$$

where,

P_i be the probability that a data instance or a tuple ' d ' belongs to class C_i . It is computed as:

$P_i = \frac{\text{No. of data instances belonging to class } i}{\text{Total no of data instances in the training dataset } T}$

GINI Index assumes a binary split on each attribute, therefore, every attribute is considered as a binary attribute which splits the data instances into two subsets S_1 and S_2 .

Gini_Index(T, A) is computed as given in Eq. (6.14).

$$\text{Gini_Index}(T, A) = \frac{|S_1|}{|T|} \text{Gini}(S_1) + \frac{|S_2|}{|T|} \text{Gini}(S_2) \quad (6.14)$$

The splitting subset with minimum Gini_Index is chosen as the best splitting subset for an attribute. The best splitting attribute is chosen by the minimum Gini_Index which is otherwise maximum ΔGini because it reduces the impurity.

ΔGini is computed as given in Eq. (6.15):

$$\Delta\text{Gini}(A) = \text{Gini}(T) - \text{Gini}(T, A) \quad (6.15)$$

6.2.4 Regression Trees

Regression trees are a variant of decision trees where the target feature is a continuous valued variable. These trees can be constructed using an algorithm called reduction in variance which uses standard deviation to choose the best splitting attribute.

Algorithm 6.5: Procedure for Constructing Regression Trees

1. Compute standard deviation for each attribute with respect to target attribute.
2. Compute standard deviation for the number of data instances of each distinct value of an attribute.
3. Compute weighted standard deviation for each attribute.

4. Compute standard deviation reduction by subtracting weighted standard deviation for each attribute from standard deviation of each attribute.
5. Choose the attribute with a higher standard deviation reduction as the best split attribute.
6. The best split attribute is placed as the root node.
7. The root node is branched into subtrees with each subtree as an outcome of the test condition of the root node attribute. Accordingly, the training dataset is also split into different subsets.
8. Recursively apply the same operation for the subset of the training set with the remaining attributes until a leaf node is derived or no more training instances are available in the subset.