Linear Regression

Linear regression is the most basic and commonly used predictive analysis. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

There are several linear regression analyses available to the researcher.

Simple linear regression

- One dependent variable (interval or ratio)
- One independent variable (interval or ratio or dichotomous)

$$\hat{y} = \theta_0 + \theta_1 x$$

Multiple linear regression

- One dependent variable (interval or ratio)
- Two or more independent variables (interval or ratio or dichotomous)

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Logistic regression

- One dependent variable (binary)
- Two or more independent variable(s) (interval or ratio or dichotomous)

Ordinal regression

- One dependent variable (ordinal)
- One or more independent variable(s) (nominal or dichotomous)

Multinomial regression

- One dependent variable (nominal)
- One or more independent variable(s) (interval or ratio or dichotomous)

Discriminant analysis

- One dependent variable (nominal)
- One or more independent variable(s) (interval or ratio)

Linear regression in one variable, also known as simple linear regression, is a statistical method used to model the linear relationship between a dependent variable and a single independent variable.

For example we want to predict a student's exam score based on how many hours they studied. We observe that as students study more hours, their scores go up. In the example of predicting exam scores based on hours studied. Here

- **Independent variable (input):** Hours studied because it's the factor we control or observe.
- **Dependent variable (output):** Exam score because it depends on how many hours were studied.

We use the independent variable to predict the dependent variable.

Linear Equation: The relationship is represented by a straight line equation:

$$Y = a + bX$$

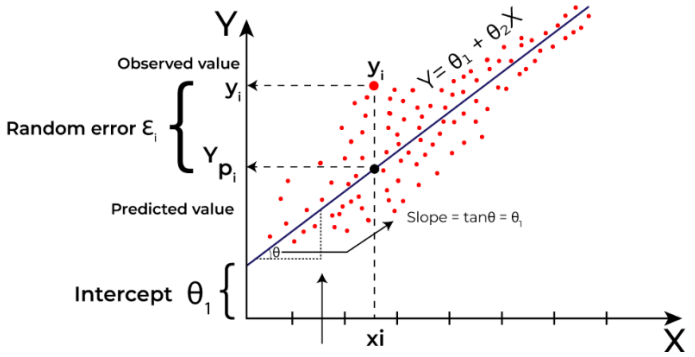
Y is the predicted value of the dependent variable.
X is the value of the independent variable.
a is the y-intercept, representing the value of Y when X is zero.
b is the slope of the line, indicating how much Y changes for each unit increase in X.

Dependent Variable (Y): The variable whose value is being predicted or explained.

Independent Variable (X): The single variable used to predict or explain the dependent variable.

x and *y* are two variables on the regression line.
b = Slope of the line.
a = *y*-intercept of the line.
x = Values of the first data set.
y = Values of the second data set.

$$b\,(slope) = \frac{n\sum xy - (\sum x)(\sum y)}{n\sum x^2 - (\sum x)^2} \qquad a\,(intercept) = \frac{\sum y \sum x^2 - \sum x \sum xy}{(\sum x^2) - (\sum x)^2}$$



x	2	4	6	8
y	3	7	5	10

Solution:

Construct the following table:

x	y	x^2	xy
2	3	4	6
4	7	16	28
6	5	36	30
8	10	64	80
$\sum x$ = 20	$\sum y$ = 25	$\sum x^2$ = 120	$\sum xy$ = 144

$b = \frac{n\sum xy - (\sum x)(\sum y)}{n\sum x^2 - (\sum x)^2}$

$b = \frac{4\times144 - 20\times25}{4\times120 - 400}$

b = 0.95

$a = \frac{\sum y \sum x^2 - \sum x \sum xy}{n(\sum x^2) - (\sum x)^2}$

$a = \frac{25\times120 - 20\times144}{4(120) - 400}$

a = 1.5

Linear regression is given by:

y = a + bx

y = 1.5 + 0.95 x

Why Linear Regression is Important?

Here's why linear regression is important:

- **Simplicity and Interpretability:** It's easy to understand and interpret, making it a starting point for learning about machine learning.
- **Predictive Ability**: Helps predict future outcomes based on past data, making it useful in various fields like finance, healthcare and marketing.
- **Basis for Other Models:** Many advanced algorithms, like logistic regression or neural networks, build on the concepts of linear regression.
- **Efficiency:** It's computationally efficient and works well for problems with a linear relationship.
- **Widely Used:** It's one of the most widely used techniques in both statistics and machine learning for regression tasks.
- **Analysis:** It provides insights into relationships between variables (e.g., how much one variable influences another).

Applications:

Simple linear regression is used for:

- **Prediction:**
  Estimating the value of the dependent variable for a given value of the independent variable.
- **Understanding Relationships:**
  Quantifying the strength and direction of the linear relationship between two variables.

Example:

Predicting a person's weight (dependent variable) based on their height (independent variable). The regression line would show how weight changes with height, and the equation would allow for predicting weight for a specific height.

Use Case of Multiple Linear Regression

Multiple linear regression allows us to analyze relationship between multiple independent variables and a single dependent variable. Here are some use cases:

- **Real Estate Pricing:** In real estate MLR is used to predict property prices based on multiple factors such as location, size, number of bedrooms, etc. This helps buyers and sellers understand market trends and set competitive prices.
- **Financial Forecasting:** Financial analysts use MLR to predict stock prices or economic indicators based on multiple influencing factors such as interest rates, inflation rates and market trends. This enables better investment strategies and risk management24.
- **Agricultural Yield Prediction:** Farmers can use MLR to estimate crop yields based on several variables like rainfall, temperature, soil quality and fertilizer usage. This information helps in planning agricultural practices for optimal productivity
- **E-commerce Sales Analysis:** An e-commerce company can utilize MLR to assess how various factors such as product price, marketing promotions and seasonal trends impact sales.

Now that we have understood about linear regression, its assumption and its type now we will learn how to make a linear regression model.

Cost function for Linear Regression

As we have discussed earlier about best fit line in linear regression, its not easy to get it easily in real life cases so we need to calculate errors that affects it. These errors need to be calculated to mitigate them. The difference between the predicted value Y^ and the true value Y and it is called cost function or the loss function.

In Linear Regression, the Mean Squared Error (MSE) cost function is employed, which calculates the average of the squared errors between the predicted values y^i and the actual values yi. The purpose is to determine the optimal values for the intercept θ1 and the coefficient of the input feature θ2ϑ2 providing the best-fit line for the given data points. The linear equation expressing this relationship is y^i=θ1+θ2xi

MSE function can be calculated as:

Cost function(J)=1n∑ni(yi^−yi)2

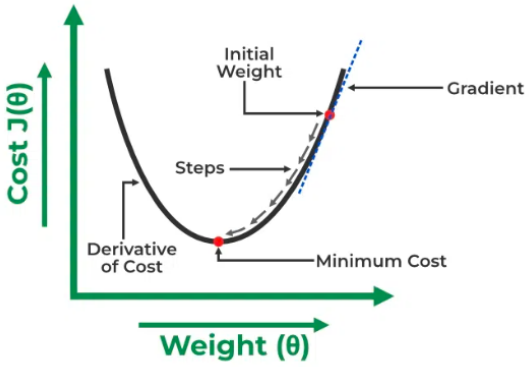
Utilizing the MSE function, the iterative process of gradient descent is applied to update the values of \θ1&θ2. This ensures that the MSE value converges to the global minima, signifying the most accurate fit of the linear regression line to the dataset.

This process involves continuously adjusting the parameters \(\theta_1\) and \(\theta_2\) based on the gradients calculated from the MSE. The final result is a linear regression line that minimizes the overall squared differences between the predicted and actual values, providing an optimal representation of the underlying relationship in the data.

Now we have calculated loss function we need to optimize model to mitigate this error and it is done through gradient descent.

Gradient Descent for Linear Regression

Gradient descent is an optimization technique used to train a linear regression model by minimizing the prediction error. It works by starting with random model parameters and repeatedly adjusting them to reduce the difference between predicted and actual values.



Gradient Descent

How it works:

- Start with random values for slope and intercept.
- Calculate the error between predicted and actual values.
- Find how much each parameter contributes to the error (gradient).
- Update the parameters in the direction that reduces the error.
- Repeat until the error is as small as possible.

This helps the model find the best-fit line for the data.

For more details you can refer to: Gradient Descent in Linear Regression

Evaluation Metrics for Linear Regression

A variety of evaluation measures can be used to determine the strength of any linear regression model. These assessment metrics often give an indication of how well the model is producing the observed outputs.

The most common measurements are:

1. Mean Square Error (MSE)

Mean Squared Error (MSE) is an evaluation metric that calculates the average of the squared differences between the actual and predicted values for all the data points. The difference is squared to ensure that negative and positive differences don't cancel each other out.

$MSE=1n\sum i=1n(yi-yi^)2$

Here,

- $n$is the number of data points.
- yi is the actual or observed value for theithithdata point.
- yi^yi is the predicted value for the ithithdata point.

MSE is a way to quantify the accuracy of a model's predictions. MSE is sensitive to outliers as large errors contribute significantly to the overall score.

2. Mean Absolute Error (MAE)

Mean Absolute Error is an evaluation metric used to calculate the accuracy of a regression model. MAE measures the average absolute difference between the predicted values and actual values.

Mathematically MAE is expressed as:

$MAE=1n\sum i=1n/Yi-Yi^/$

Here,

- n is the number of observations
- Yi represents the actual values.
- Yi^Yi represents the predicted values

Lower MAE value indicates better model performance. It is not sensitive to the outliers as we consider absolute differences.

3. Root Mean Squared Error (RMSE)

The square root of the residuals' variance is the Root Mean Squared Error. It describes how well the observed data points match the expected values or the model's absolute fit to the data.

In mathematical notation, it can be expressed as:

$$RMSE=nRSS=n\sum i=2n(yiactual-yipredicted)2$$

Rather than dividing the entire number of data points in the model by the number of degrees of freedom, one must divide the sum of the squared residuals to obtain an unbiased estimate. Then, this figure is referred to as the Residual Standard Error (RSE).

In mathematical notation, it can be expressed as:

$$RMSE=RSSn=\sum i=2n(yiactual-yipredicted)2(n-2)$$

RSME is not as good of a metric as R-squared. Root Mean Squared Error can fluctuate when the units of the variables vary since its value is dependent on the variables' units (it is not a normalized measure).

4. Coefficient of Determination (R-squared)

R-Squared is a statistic that indicates how much variation the developed model can explain or capture. It is always in the range of 0 to 1. In general, the better the model matches the data, the greater the R-squared number.

In mathematical notation, it can be expressed as:

$$R2=1-(RSSTSS)$$

- **Residual sum of Squares**(RSS): The sum of squares of the residual for each data point in the plot or data is known as the residual sum of squares or RSS. It is a measurement of the difference between the output that was observed and what was anticipated.

  $$RSS=\sum i=1n(yi-b0-b1xi)2$$

- **Total Sum of Squares (TSS):** The sum of the data points' errors from the answer variable's mean is known as the total sum of squares or TSS.

  $$TSS=\sum i=1n(y-yi^{-})2$$

R squared metric is a measure of the proportion of variance in the dependent variable that is explained the independent variables in the model.

5. Adjusted R-Squared Error

Adjusted R2 measures the proportion of variance in the dependent variable that is explained by independent variables in a regression model. Adjusted R-square accounts the number of predictors in the model and penalizes the model for including irrelevant predictors that don't contribute significantly to explain the variance in the dependent variables.

Mathematically, adjusted R2 is expressed as:

$$AdjustedR2=1-((1-R2).(n-1)n-k-1)$$

Here,

- n is the number of observations
- k is the number of predictors in the model
- R2 is co-efficient of determination

Adjusted R-square helps to prevent overfitting. It penalizes the model with additional predictors that do not contribute significantly to explain the variance in the dependent variable.

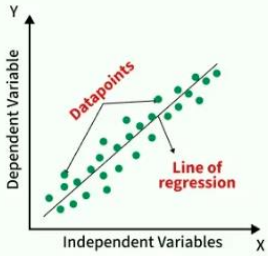
While evaluation metrics help us measure the performance of a model, regularization helps in improving that performance by addressing overfitting and enhancing generalization.

Introduction to Linear Regression

- Simple model for predicting continuous values.
- It finds the relationship between input features and output.
- It is used in forecasting and predicting salaries based on experience.

How Does Linear Regression Work?

- Finds the best-fitting line by minimizing prediction errors (least squares method)
- It calculates coefficients for variables that minimize the error in predictions.

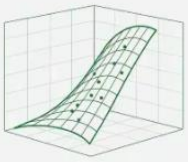


Types of Linear Regression

Simple Linear Regression

Predicts the dependent variable using a single independent variable.

Multiple Linear Regression



Uses two or more independent variables to predict the dependent variable.

Real-World Use Cases of Linear Regression

01 Stock Market Prediction

02 Real Estate Price Prediction

03 Medical Risk Prediction

04 Sales Forecasting

Challenges in Linear Regression

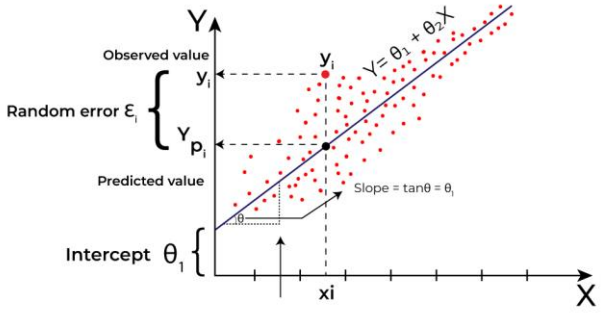
- Overfitting
- Multicollinearity
- Assumptions of Linearity
- Sensitive to Outliers

Best Fit Line in Linear Regression

In linear regression, the best-fit line is the straight line that most accurately represents the relationship between the independent variable (input) and the dependent variable (output). It is the line that minimizes the difference between the actual data points and the predicted values from the model.

1. Goal of the Best-Fit Line

The goal of linear regression is to find a straight line that minimizes the error (the difference) between the observed data points and the predicted values. This line helps us predict the dependent variable for new, unseen data.



Linear Regression

Here Y is called a dependent or target variable and X is called an independent variable also known as the predictor of Y. There are many types of functions or modules that can be used for regression. A linear function is the simplest type of function. Here, X may be a single feature or multiple features representing the problem.

2. Equation of the Best-Fit Line

For simple linear regression (with one independent variable), the best-fit line is represented by the equation

$y=mx+b$

Where:

- **y** is the predicted value (dependent variable)
- **x** is the input (independent variable)
- **m** is the slope of the line (how much y changes when x changes)
- **b** is the intercept (the value of y when x = 0)

The best-fit line will be the one that optimizes the values of m (slope) and b (intercept) so that the predicted y values are as close as possible to the actual data points.

3. Minimizing the Error: The Least Squares Method

To find the best-fit line, we use a method called **Least Squares**. The idea behind this method is to minimize the sum of squared differences between the actual values (data points) and the predicted values from the line. These differences are called residuals.

The formula for residuals is:

$Residual=y_i-\hat{y}$

Where:

- y_i is the actual observed value
- $\hat{y}_i$ is the predicted value from the line for that $x_i x_i$

The least squares method minimizes the sum of the squared residuals:

Sum of squared errors (SSE)=$\Sigma(y_i - \hat{y}_i)^2$

This method ensures that the line best represents the data where the sum of the squared differences between the predicted values and actual values is as small as possible.

4. Interpretation of the Best-Fit Line

- **Slope (m):** The slope of the best-fit line indicates how much the dependent variable (y) changes with each unit change in the independent variable (x). For example if the slope is 5, it means that for every 1-unit increase in x, the value of y increases by 5 units.
- **Intercept (b):** The intercept represents the predicted value of y when x = 0. It's the point where the line crosses the y-axis.

In linear regression some hypothesis are made to ensure reliability of the model's results.

Limitations

- ***Assumes Linearity:*** *The method assumes the relationship between the variables is linear. If the relationship is non-linear, linear regression might not work well.*
- ***Sensitivity to Outliers:*** *Outliers can significantly affect the slope and intercept, skewing the best-fit line.*

Hypothesis function in Linear Regression

In linear regression, the hypothesis function is the equation used to make predictions about the dependent variable based on the independent variables. It represents the relationship between the input features and the target output.

For a simple case with one independent variable, the hypothesis function is:

$h(x)=\beta_0+\beta_1 x$

Where:

- h(x)(ory^) is the predicted value of the dependent variable (y).
- x is the independent variable.
- β_0 is the intercept, representing the value of y when x is 0.
- β_1 is the slope, indicating how much y changes for each unit change in x.

For **multiple linear regression** (with more than one independent variable), the hypothesis function expands to:

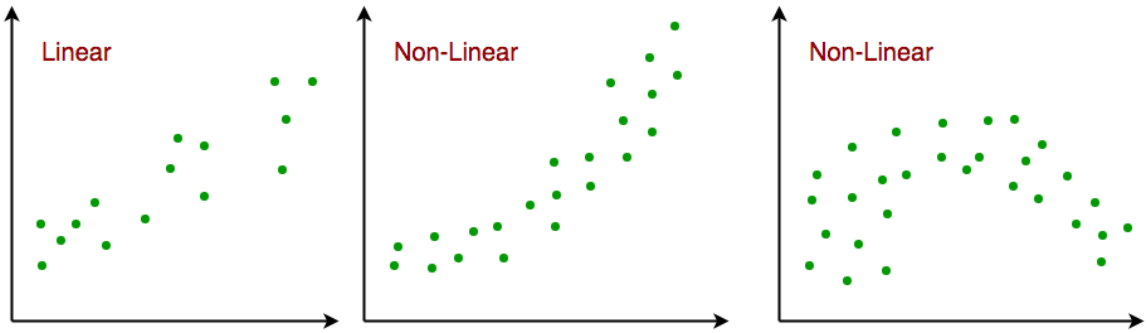
$h(x_1,x_2,...,x_k)=\beta_0+\beta_1 x_1+\beta_2 x_2+...+\beta_k x_k$

Where:

- $x_1,x_2,...,x_k$ are the independent variables.
- β_0 is the intercept.
- $\beta_1,\beta_2,...,\beta_k$ are the coefficients, representing the influence of each respective independent variable on the predicted output.

Assumptions of the Linear Regression

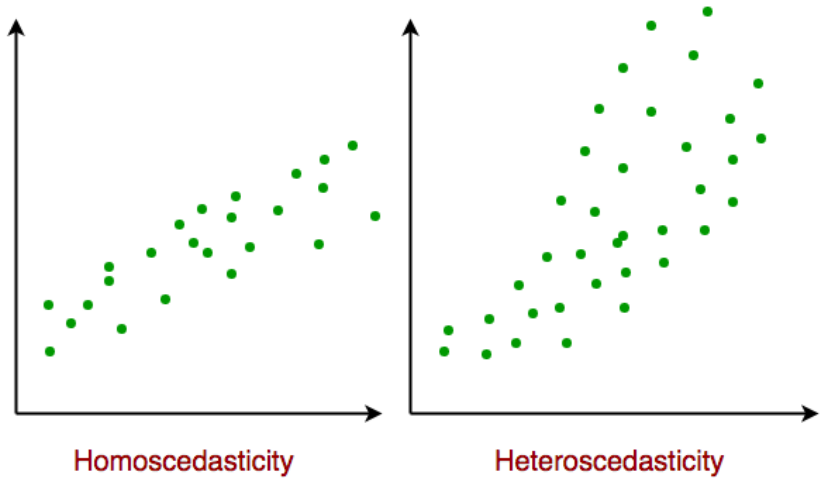
1. Linearity: The relationship between inputs (X) and the output (Y) is a straight line.



Linearity

2. Independence of Errors: The errors in predictions should not affect each other.

3. Constant Variance (Homoscedasticity): The errors should have equal spread across all values of the input. If the spread changes (like fans out or shrinks), it's called heteroscedasticity and it's a problem for the model.



Homoscedasticity

4. Normality of Errors: The errors should follow a normal (bell-shaped) distribution.

5. No Multicollinearity*(*for multiple regression): Input variables shouldn't be too closely related to each other.

6. No Autocorrelation: Errors shouldn't show repeating patterns, especially in time-based data.

7. Additivity: The total effect on Y is just the sum of effects from each X, no mixing or interaction between them.'

To understand Multicollinearityin detail refer to article: ***Multicollinearity****.*

Regularization Techniques for Linear Models

1. Lasso Regression (L1 Regularization)

Lasso Regression is a technique used for regularizing a linear regression model, it adds a penalty term to the linear regression objective function to prevent overfitting.

The objective function after applying lasso regression is:

$$J(\vartheta)=2m1\sum i=1m(yi-yi)2+\lambda\sum j=1n/\vartheta j/$$

- the first term is the least squares loss, representing the squared difference between predicted and actual values.
- the second term is the L1 regularization term, it penalizes the sum of absolute values of the regression coefficient θj.

2. Ridge Regression (L2 Regularization)

Ridge regression is a linear regression technique that adds a regularization term to the standard linear objective. Again, the goal is to prevent overfitting by penalizing large coefficient in linear regression equation. It useful when the dataset has multicollinearity where predictor variables are highly correlated.

The objective function after applying ridge regression is:

$$J(\vartheta)=2m1\sum i=1m(yi-yi)2+\lambda\sum j=1n\vartheta j2$$

- the first term is the least squares loss, representing the squared difference between predicted and actual values.
- the second term is the L1 regularization term, it penalizes the sum of square of values of the regression coefficient θj.

3. Elastic Net Regression

Elastic Net Regression is a hybrid regularization technique that combines the power of both L1 and L2 regularization in linear regression objective.

$$J(\vartheta)=12m\sum i=1m(yi\hat{}-yi)2+\alpha\lambda\sum j=1n/\vartheta j/+12(1-\alpha)\lambda\sum j=1n\vartheta j2$$

the first term is least square loss.

- the second term is L1 regularization and third is ridge regression.
- λ is the overall regularization strength.
- ααcontrols the mix between L1 and L2 regularization.

Decision tree

A decision tree is a supervised learning algorithm used for both classification and regression tasks. It has a hierarchical tree structure which consists of a root node, branches, internal nodes and leaf nodes. It works like a flowchart help to make decisions step by step where:

- Internal nodes represent attribute tests
- Branches represent attribute values
- Leaf nodes represent final decisions or predictions.

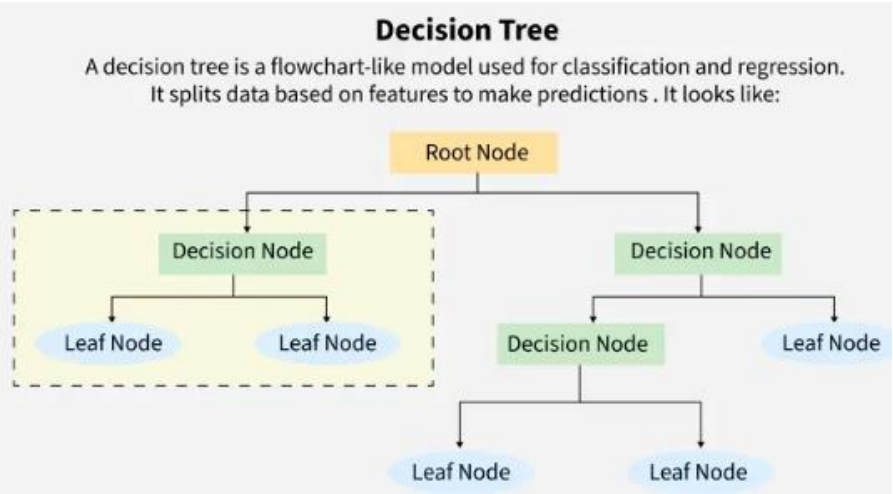
We have two popular attribute selection measures used:

Information Gain	Gini Index
Information Gain tells us how useful a question (or feature) is for splitting data into groups. It measures how much the uncertainty decreases after the split. A good question will create clearer groups and the feature with the highest Information Gain is chosen to make the decision.	Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified. It means an attribute with a lower Gini index should be preferred. Sklearn supports "Gini" criteria for Gini Index and by default it takes "gini" value.
$Gain(S,A) = Entropy(S) - \sum_v^A \frac{\lvert S_v \rvert}{\lvert S \rvert}.Entropy(S_v)$	$Gini = 1 - \sum_{i=1}^{n} p_i^2$
$Gain(S,A) = Entropy(S) - \sum_{v \epsilon Values(A)} \frac{\lvert S_v \rvert}{\lvert S \rvert}.Entropy(S_v)$	

Suppose *S* is a set of instances A is an attribute, Sv*Sv* is the subset of S*S*, v*v* represents an individual value that the attribute A can take and Values (A) is the set of all possible values of A then

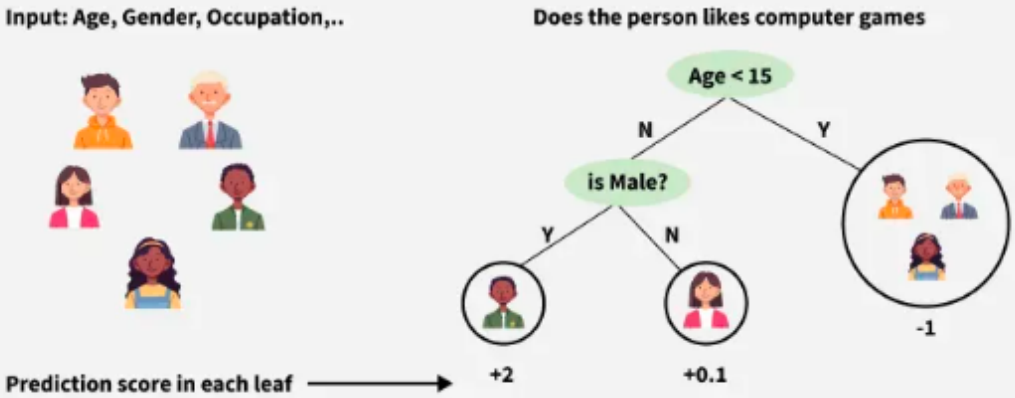
Entropy: is the measure of uncertainty of a random variable it characterizes the impurity of an arbitrary collection of examples. The higher the entropy more the information content.

Suppose S is a set of instances, A is an attribute, Sv is the subset of SS with A= v and Values (A) is the set of all possible values of A, then



Working of Decision Tree

The model checks conditions like age and gender to split users into groups. Each group (leaf node) gets a prediction score based on user preferences for computer games.



Splitting Criteria In Decision Tree

In decision trees, splitting criteria help decide which feature to split on at each node. The two most common criteria are:

Gini Index

$$I_G = 1 - \sum_{j=1}^{c} p_j^2$$

P_j : proportion of the samples that belongs to class c for a particular node

Entropy

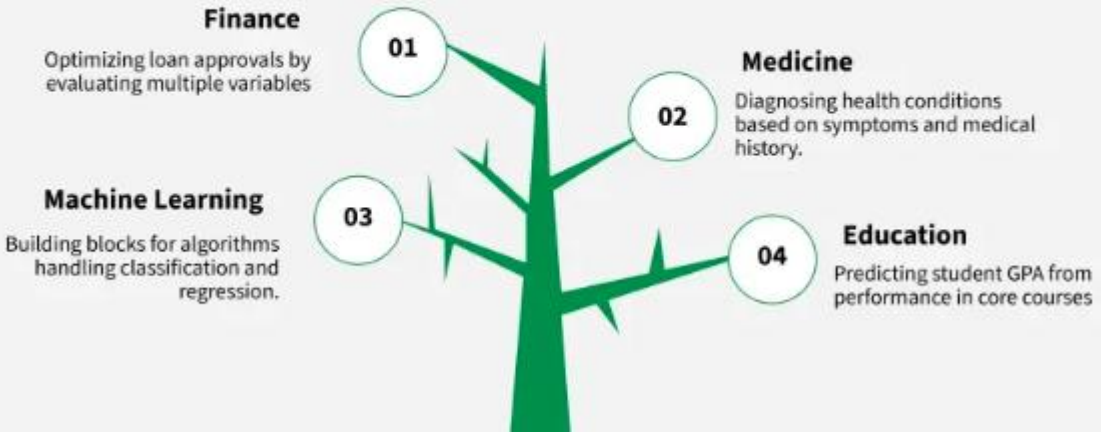
$$I_H = -\sum_{j=1}^{c} p_j log_2(p_j)$$

p_j: proportion of the samples that belongs to class c for a particular node.

*This is the the definition of entropy for all non-empty classes (p ne0) The entropy is 0 if all samples at a node belong to the same class.

Applications of Decision Trees

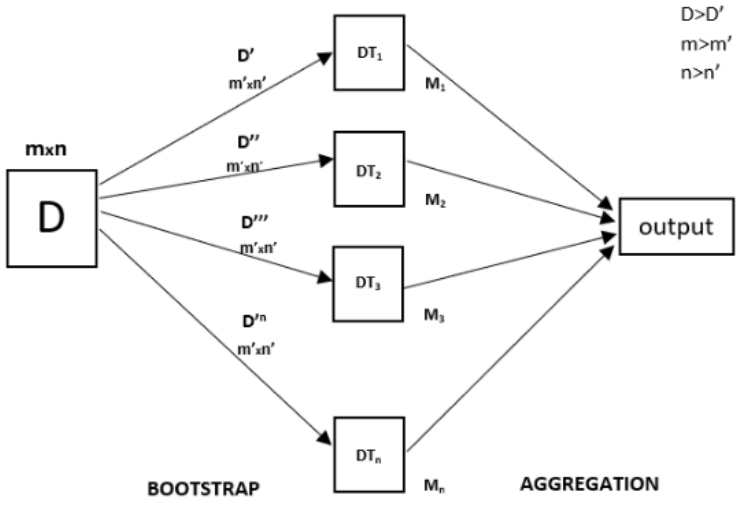
A decision tree is a flowchart-like model used for classification and regression. It splits data based on features to make predictions . It looks like:



https://www.geeksforgeeks.org/machine-learning/decision-tree-introduction-example/

Random forest

A random forest is an ensemble learning method that combines the predictions from multiple decision trees to produce a more accurate and stable prediction. It is a type of supervised learning algorithm that can be used for both classification and regression tasks.
In regression task we can use **Random Forest Regression** technique for predicting numerical values. It predicts continuous values by averaging the results of multiple decision trees.



Random Forest Regression Model Working

The process begins with Bootstrap sampling where random rows of data are selected with replacement to form different training datasets for each tree. After this we do **feature sampling** where only a random subset of features is used to build each tree ensuring diversity in the models.
After the trees are trained each tree make a prediction and the final prediction for regression tasks is the average of all the individual tree predictions and this process is called as **Aggregation.**

Applications of Random Forest Regression

The Random forest regression has a wide range of real-world problems including:

- **Predicting continuous numerical values:** Predicting house prices, stock prices or customer lifetime value.
- **Identifying risk factors:** Detecting risk factors for diseases, financial crises or other negative events.
- **Handling high-dimensional data:** Analyzing datasets with a large number of input features.
- **Capturing complex relationships:** Modeling complex relationships between input features and the target variable.

Advantages of Random Forest Regression

- **Handles Non-Linearity**: It can capture complex, non-linear relationships in the data that other models might miss.
- **Reduces Overfitting**: By combining multiple decision trees and averaging predictions it reduces the risk of overfitting compared to a single decision tree.
- **Robust to Outliers**: Random Forest is less sensitive to outliers as it aggregates the predictions from multiple trees.
- **Works Well with Large Datasets**: It can efficiently handle large datasets and high-dimensional data without a significant loss in performance.
- **Handles Missing Data**: Random Forest can handle missing values by using surrogate splits and maintaining high accuracy even with incomplete data.
- **No Need for Feature Scaling**: Unlike many other algorithms Random Forest does not require normalization or scaling of the data.

Disadvantages of Random Forest Regression

- **Complexity**: It can be computationally expensive and slow to train especially with a large number of trees and high-dimensional data. Due to this it may not be suitable for real-time predictions especially with a large number of trees.
- **Less Interpretability**: Since it uses many trees it can be harder to interpret compared to simpler models like linear regression or decision trees.
- **Memory Intensive**: Storing multiple decision trees for large datasets require significant memory resources.
- **Overfitting on Noisy Data**: While Random Forest reduces overfitting, it can still overfit if the data is highly noisy especially with a large number of trees.
- **Sensitive to Imbalanced Data**: It may perform poorly if the dataset is highly imbalanced like one class is significantly more frequent than another.

Implementing Random Forest Regression steps in Python

1. **Importing Libraries**
2. **Importing Dataset**
3. Data Preparation
4. **Random Forest Regressor Model**
5. **Making predictions and Evaluating**
6. **Visualizing**
7. Visualizing a Single Decision Tree from the Random Forest Model

https://www.geeksforgeeks.org/machine-learning/random-forest-regression-in-python/