Data Science with Python

Supervised Learning

Agenda

01 What is Supervised Learning?

02 Supervised vs Unsupervised

03 What is Classification?

04 Types of Classification

05 What is Regression?

06 Types of Regression

07 Linear Regression

08 Multiple Linear Regression

09 Logistic Regression

10 Decision Tree & Random Forest

11 Confusion Matrix

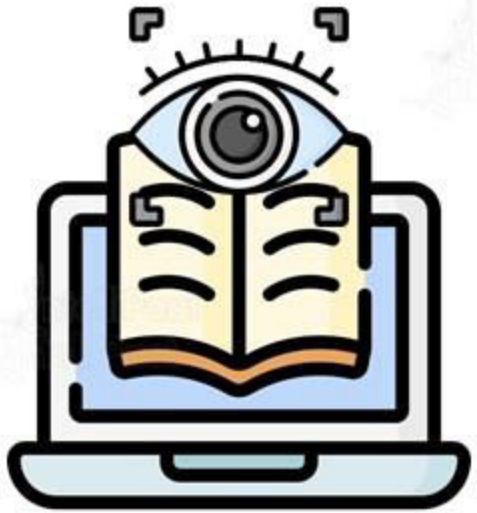
12 Naïve Bayes Classifier

What is Supervised Learning?

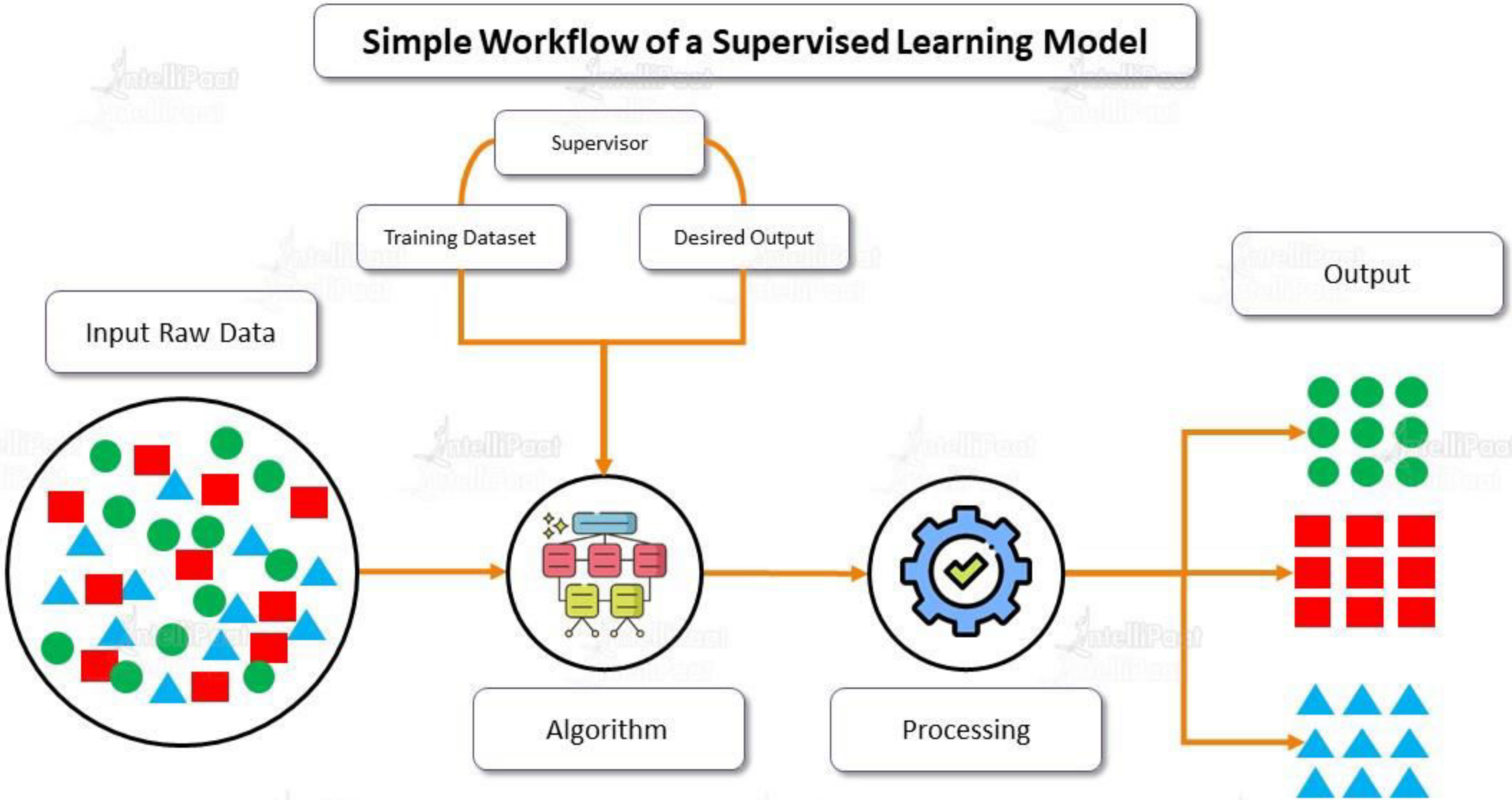
What is Supervised Learning?

In supervised learning, the machine learns from the labeled data, i.e., we already know the result of the input data

It is called '**supervised**' learning because the algorithm learns from a dataset. The goal is to make the algorithm's output as accurate as possible when a new input is fed to it



What is Supervised Learning?



Supervised vs Unsupervised

Supervised vs Unsupervised

Supervised Learning	Unsupervised Learning
1. Input and output variables will be given	1. Only the input variable is given
2. Labeled data is used	2. The data is not labeled
3. Training takes place offline, and the training data should be available	3. The training takes place in real time

What is Classification?

What is Classification?

Classification is the process of **grouping things according to the similar features** they share

Types of Classification

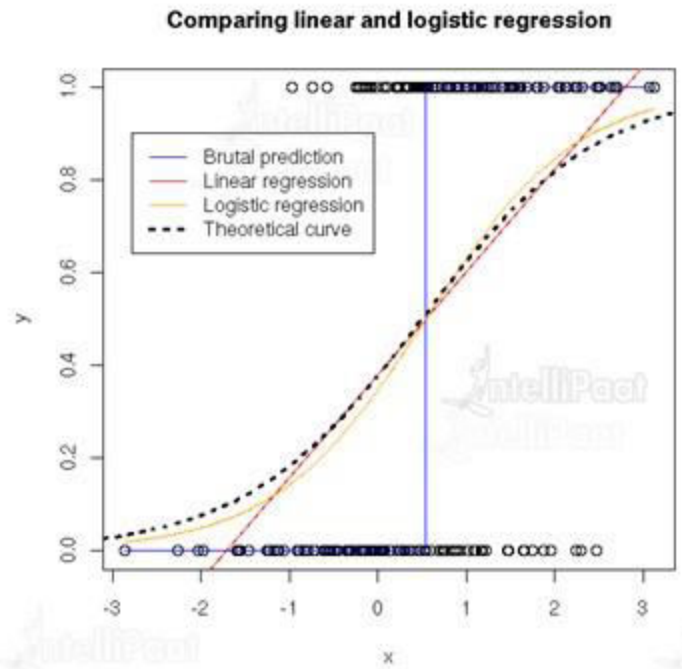
Types of Classification

Types of Classification

- Logistic Regression
- Decision Tree
- Random Forest
- K-Nearest Neighbor
- Naïve Bayes

Logistic regression is used when the dependent variable (target) is categorical

For example, predicting whether an email is spam (1) or not (0)

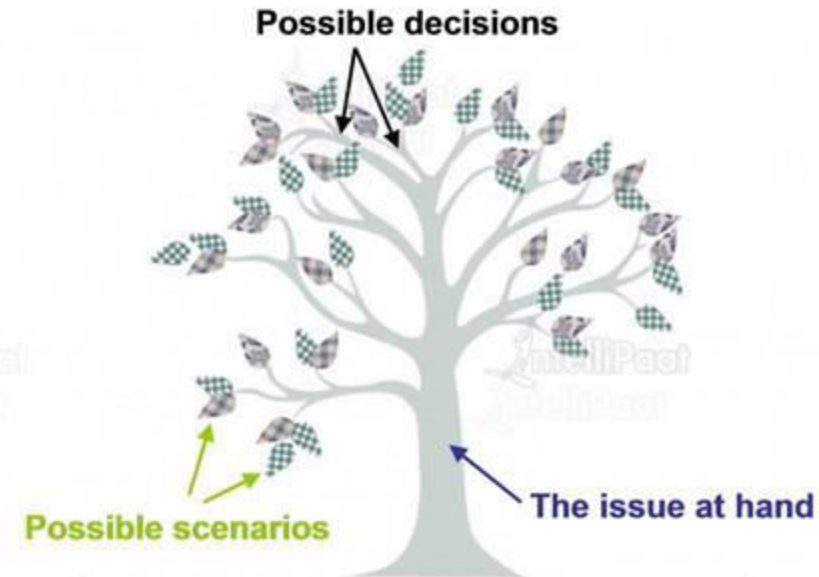


Types of Classification

- Logistic Regression
- Decision Tree
- Random Forest
- K-Nearest Neighbor
- Naïve Bayes

Graphical representation of all possible solutions to a problem

- Decisions are made based on some conditions
- The decisions made can be easily explained

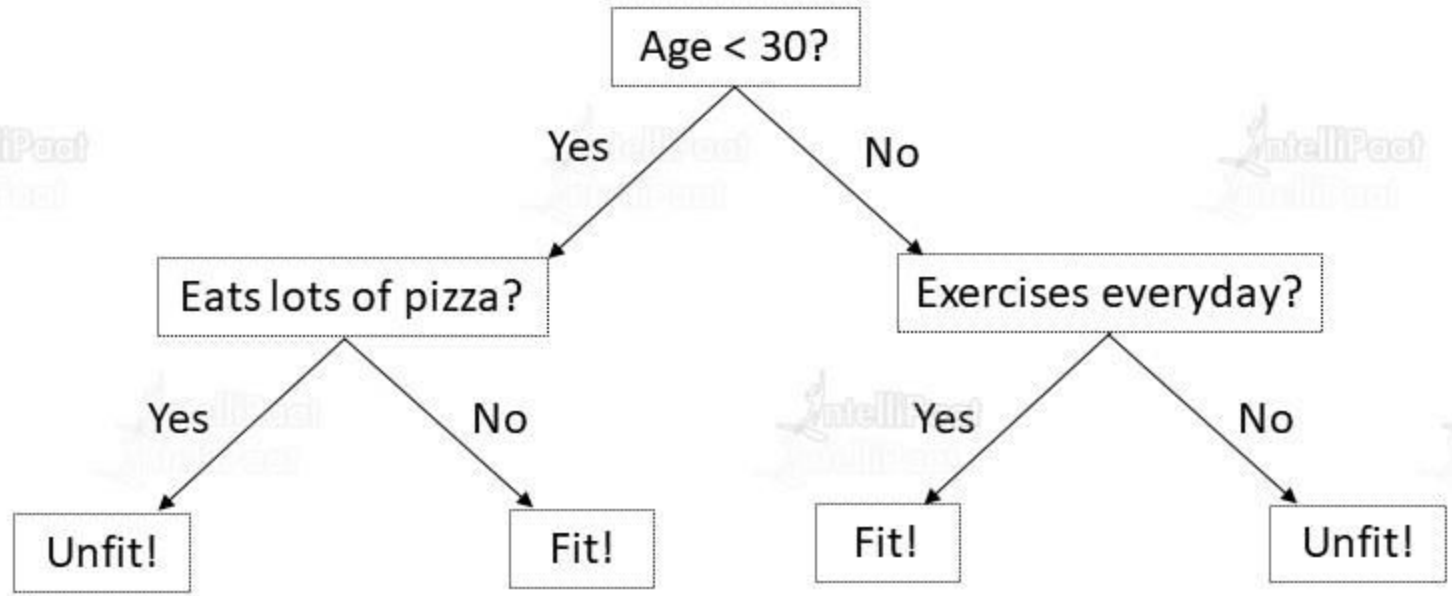


Types of Classification

- Logistic Regression
- Decision Tree
- Random Forest
- K-Nearest Neighbor
- Naïve Bayes

An Example of a Decision Tree

We are creating a decision tree to check if a person is fit

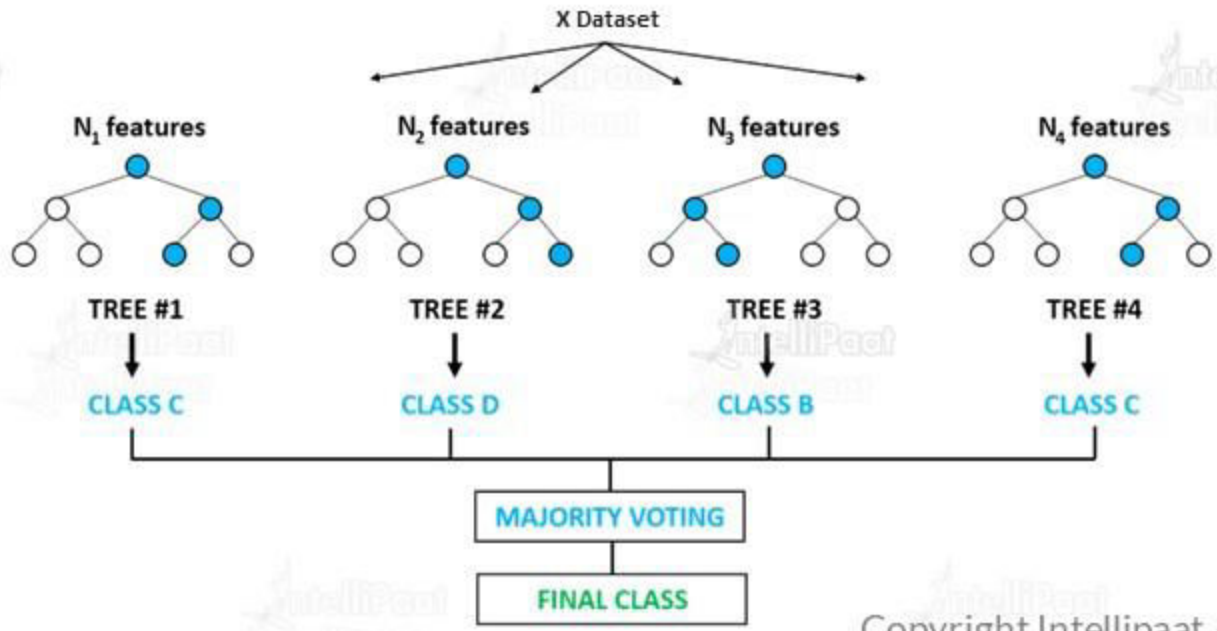


Types of Classification

- Logistic Regression
- Decision Tree
- **Random Forest**
- K-Nearest Neighbor
- Naïve Bayes

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction

- It corrects decision trees' habit of overfitting their training set
- It is trained with the 'bagging' method

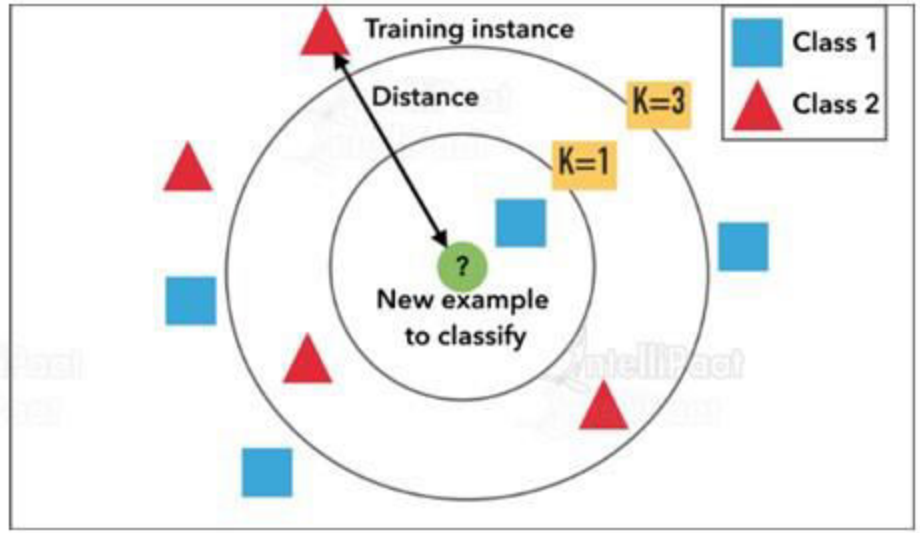


Types of Classification

- Logistic Regression
- Decision Tree
- Random Forest
- **K-Nearest Neighbor**
- Naïve Bayes

KNN algorithms use data and classify new data points based on some similarity measures

The 'K' in KNN algorithm is the nearest neighbors we wish to take vote from

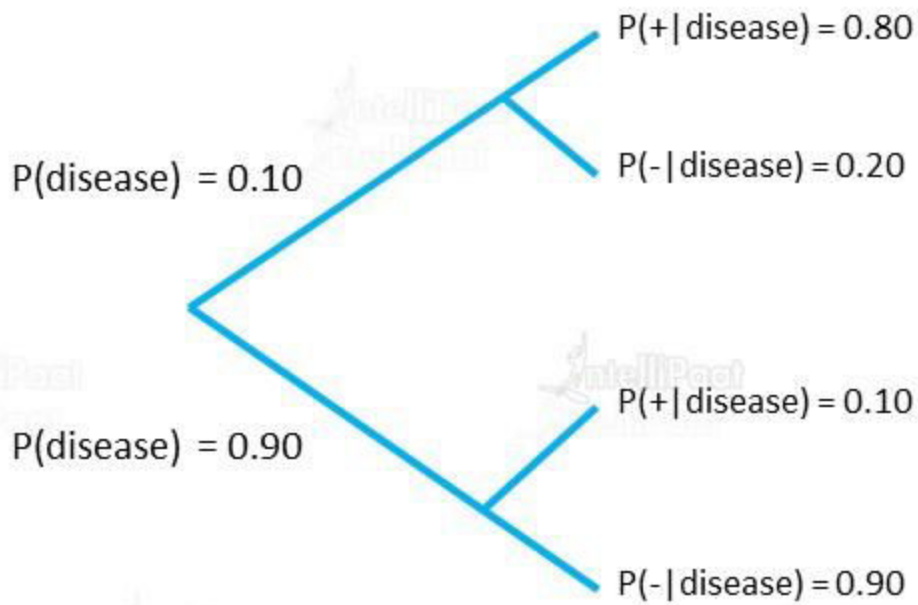


Types of Classification

- Logistic Regression
- Decision Tree
- Random Forest
- K-Nearest Neighbor
- Naïve Bayes

It is a classification based on the Bayes' theorem

Assumption: Presence of a particular feature in a class is not related to the presence of any other feature

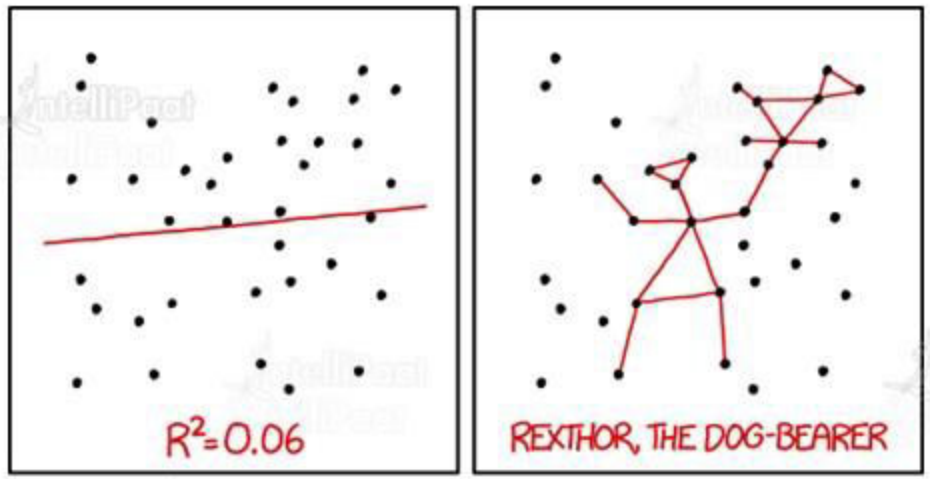


What is Regression?

What is Regression?

A technique of finding the relationship between two or more variables

Change in a dependent variable is associated with a change in one or more independent variables

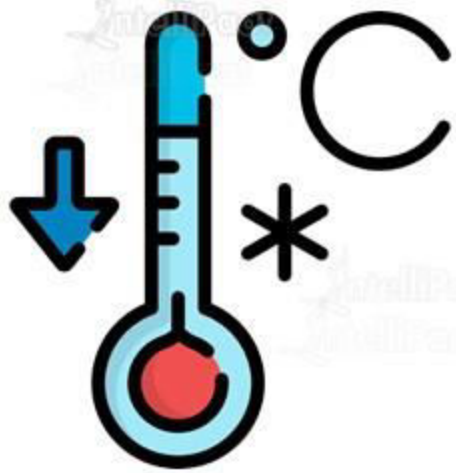


I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER TO GUESS THE DIRECTION OF THE CORRELATION FROM THE SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.

What is Regression?

Regression is a technique that displays the relationship between the variables 'x' and ‘y,’ i.e., the value of 'x' changes with respect to the changing values of variable ‘y’

For instance, let us consider the temperature as ‘y’ and people wearing jackets as ‘x.’ When temperature starts falling, people tend to wear jackets more often than when it is hot



What is Regression?

If we sense that there is a relation between two things, regression would help confirm it!

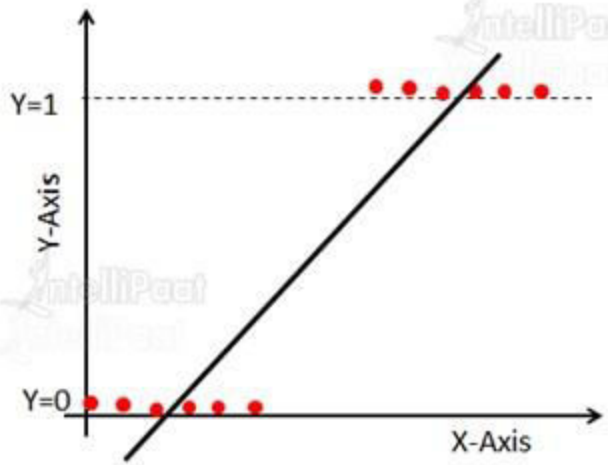
- Temperature vs the number of cones sold at the street shop
- Inches of rain vs new cars sold
- Daily snowfall vs the number of skier visits

Types of Regression

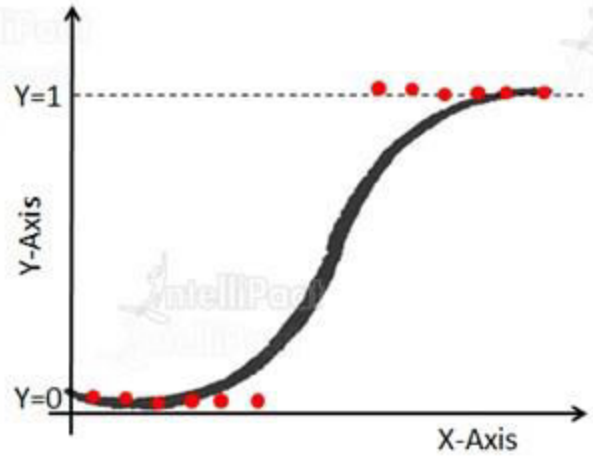
Types of Regression

There are a lot of different regression types, but the most commonly used regressions in Data Science are:

Linear Regression



Logistic Regression



Types of Regression

Linear Regression

1. Continuous variables
2. Solves regression problems
3. Straight line output

Logistic Regression

1. Categorical variables
2. Solves classification problems
3. S-curve output

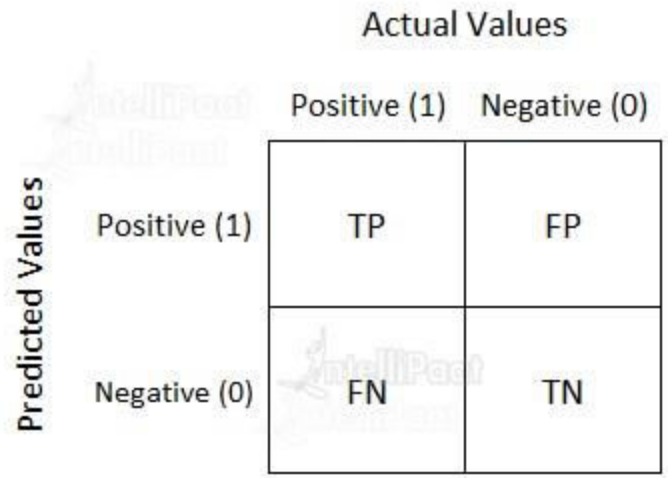
Confusion Matrix

Confusion Matrix

A confusion matrix is used to measure performance for a Machine Learning classification problem, where the output can be of two or more classes

- It is a table with different combinations of predicted and actual values based on the number of classes
- It is extremely useful for measuring Recall and Precision

Confusion Matrix



Confusion Matrix

Let us understand some terminologies

- **True Positive (TP)**: A record that predicted to have a class of positive and is actually positive (i.e., correctly classified as positive)
- **False Positive (FP)**: A record that predicted to have a class of positive and is actually negative (i.e., incorrectly classified as positive)
- **True Negative (TN)**: A record that predicted to have a class of negative and is actually negative (i.e., correctly classified as negative)
- **False Negative (FN)**: A record that predicted to have a class of negative and is actually positive (i.e., incorrectly classified as negative)

Note: Positive and negative are taken as examples. In the real world, there can be multiple classes

Confusion Matrix

Let us understand some terminology

- **Recall**: What proportion of positive identifications was actually correct?
- **Precision**: What proportion of positive identifications was actually correct?
- **Accuracy**: Out of all the classes, how much we predicted correctly

$$\textit{Recall} = \frac{TP}{TP + FN}$$

$$\textit{Precision} = \frac{TP}{TP + FP}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Confusion Matrix

How to calculate a confusion matrix?

Correct Prediction: 7/10

Accuracy: 70%

Expected	Predicted
man	woman
man	man
woman	woman
man	man
woman	man
woman	woman
woman	woman
man	man
man	woman
woman	woman

Confusion Matrix

How to calculate a confusion matrix?

	men	women
men	**3**	**1**
women	**2**	**4**

- Total actual men: (3 + 2)
- Total actual women: (1 + 4)
- Total correct values: (3 + 4)

Conclusion: More errors while predicting men as women than predicting women as men

Expected	Predicted
man	woman
man	man
woman	woman
man	man
woman	man
woman	woman
woman	woman
man	man
man	woman
woman	woman

Confusion Matrix

How to calculate a confusion matrix?

	men	women
men	**3**	**1**
women	**2**	**4**

- Recall (Men): 3 / (3 + 2) = 3 / 5 = 60%
- Precision (Men): 3 / (3 + 1) = 3 / 4 = 75%
- Recall (Women): 4 / (4 + 1) = 4 / 5 = 80%
- Precision (Women): 4 / (4 + 2) = 4 / 6 = 66.67%
- Accuracy: 7 / 10 = 70%

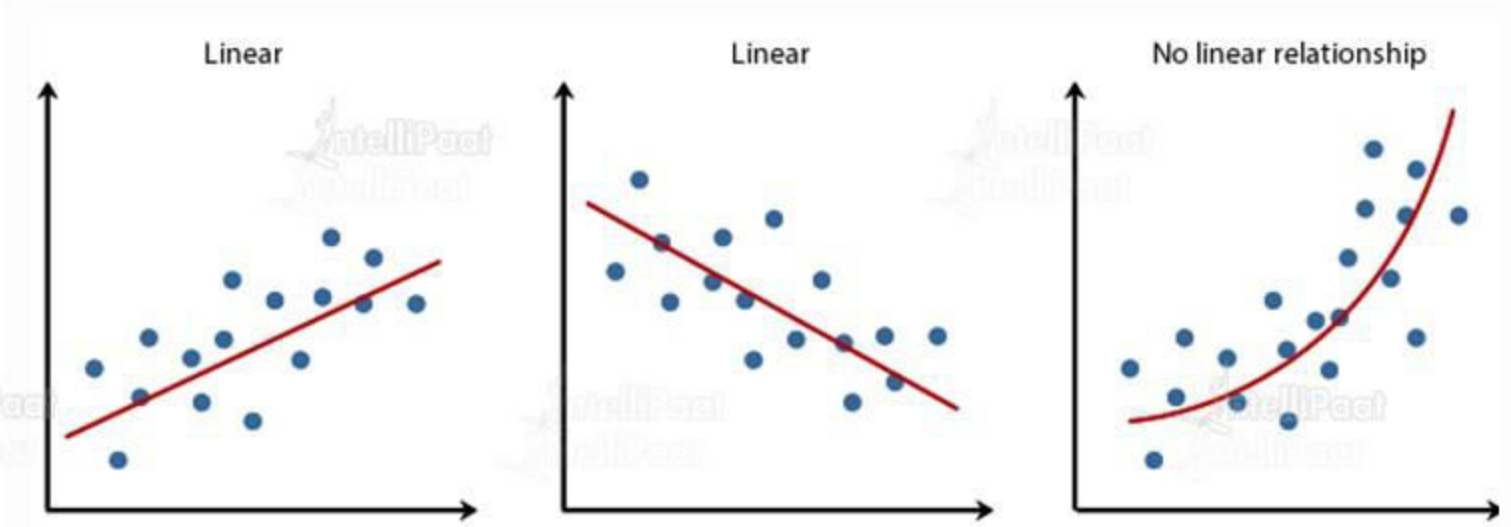
Expected	Predicted
man	woman
man	man
woman	woman
man	man
woman	man
woman	woman
woman	woman
man	man
man	woman
woman	woman

Linear Regression

Linear Regression

Simple linear regression is useful for finding the relationship between two continuous variables

One is the predictor or independent variable, and the other is the response or dependent variable



Linear Regression

There are four assumptions associated with a linear regression model:

- Linearity
- Homoscedasticity
- Independence
- Normality

The relationship between X and the mean of Y is linear

Linear Regression

There are four assumptions associated with a linear regression model:

- Linearity
- Homoscedasticity
- Independence
- Normality

The variance of residual (Error) is the same for any value of X

Linear Regression

There are four assumptions associated with a linear regression model:

- Linearity
- Homoscedasticity
- Independence
- Normality

Observations are independent of each other, i.e., no strong correlation between any independent variables

Linear Regression

There are four assumptions associated with a linear regression model:

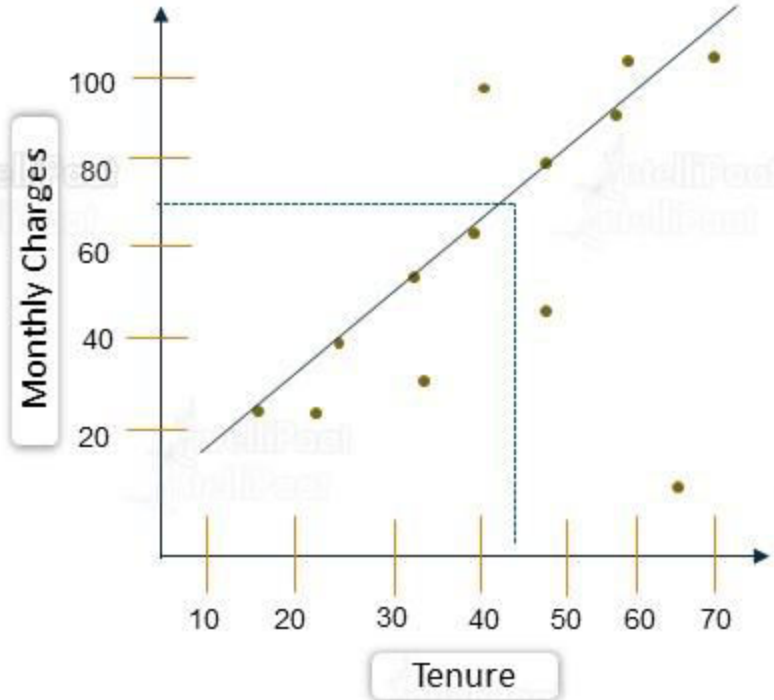
- Linearity
- Homoscedasticity
- Independence
- Normality

For any fixed value of X, Y is normally distributed

Linear Regression

Let us discuss linear regression with an example. We want to know how do the ***monthly charges*** of a customer vary with respect to the ***tenure***

Estimating the value of monthly charges with the tenure of the customer



Linear Regression

In general, the data doesn't fall exactly on a line, so the regression equation should include an implicit **error term**

The **fitted values (predicted values)** are typically denoted by $\hat{Y}$ (Y-hat)

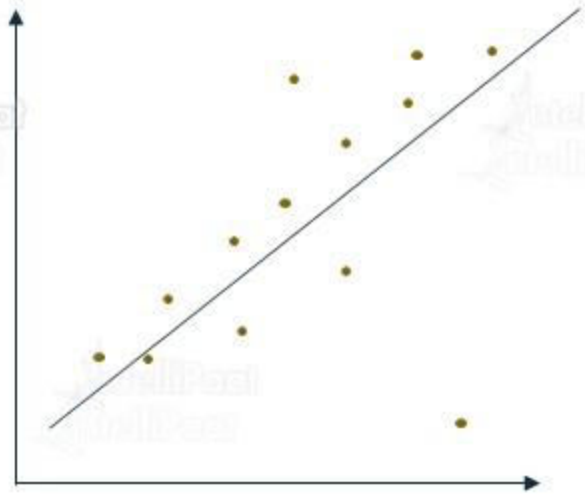
$$Y_i = b_0 + b_1 X_i + e_i$$

$$\hat{Y}_i = \hat{b}_0 + \hat{b}_1 X_i$$

Linear Regression

If **b1 > 0**, then **x(predictor)** and **y(target)** have a positive relationship, i.e., an **increase in *x* will increase *y***

b1 > 0 ⟹ Positive Relationship

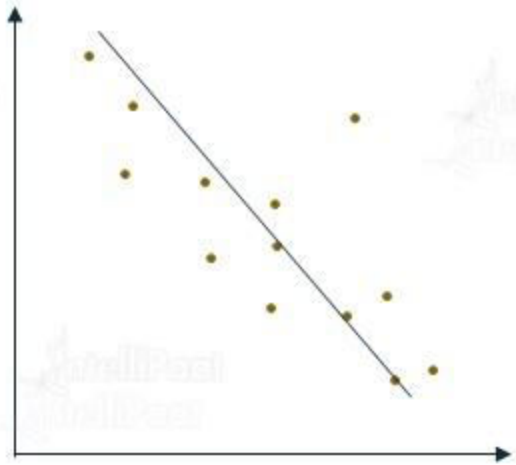


$$y = b_0 + b_1 x$$

Linear Regression

If **b1 < 0**, then **x(predictor)** and **y(target)** have a negative relationship, i.e., an **increase in *x* will decrease *y***

b1 < 0 ⟹ Negative Relationship

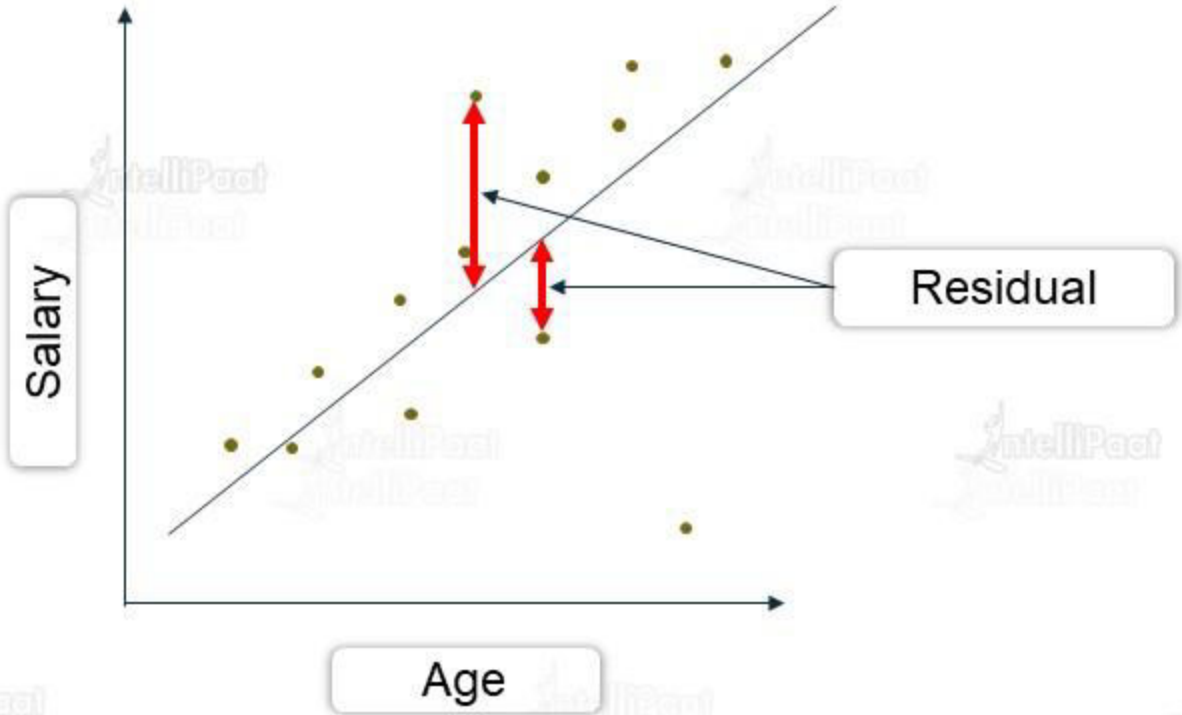


$$y = b_0 + b_1 x$$

Linear Regression

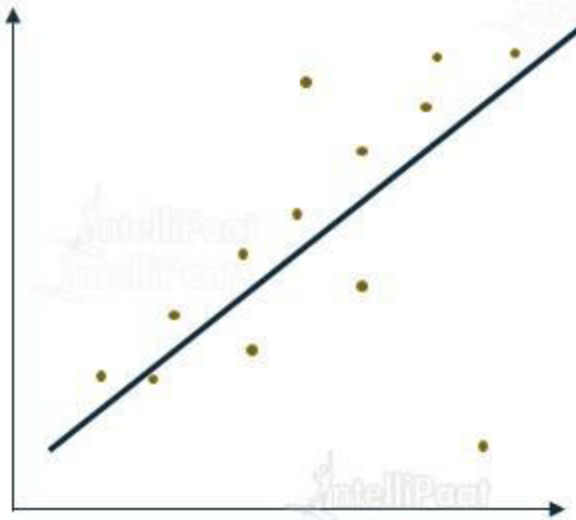
What are residuals?

The **difference** between the **observed value of the dependent variable (y)** and the **predicted value (ŷ)** is called the **residual**. Each data point has one residual

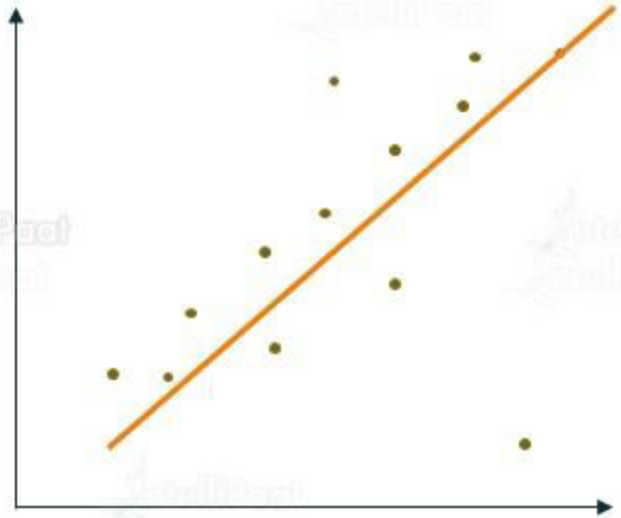


Linear Regression

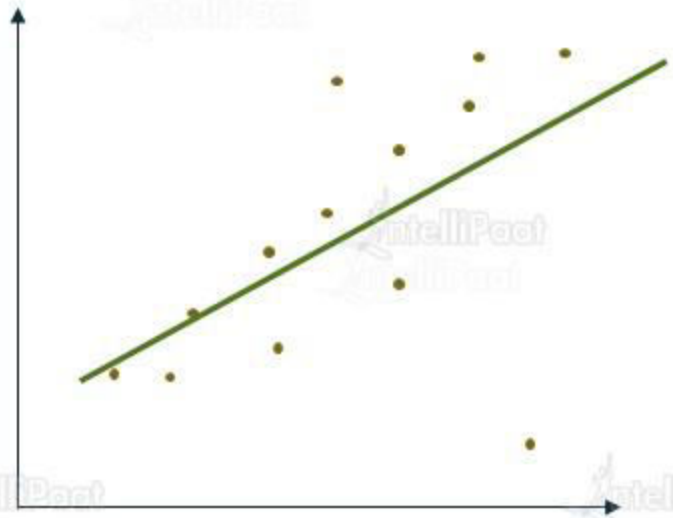
There could be multiple fit lines passing through the points, so how shall we choose the **line of best fit**?



Line Fit 1



Line Fit 2

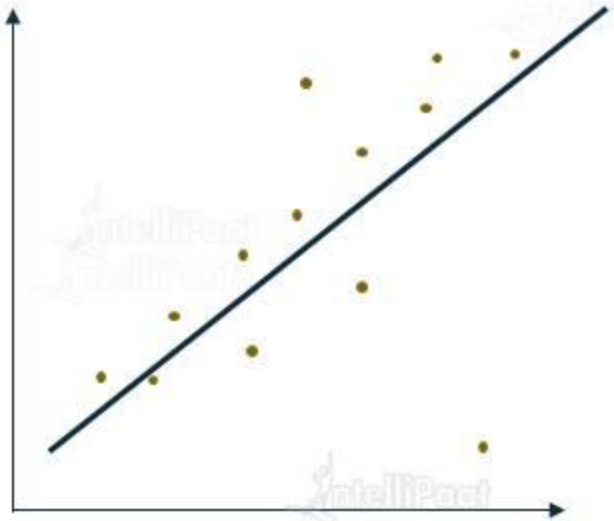


Line Fit 3

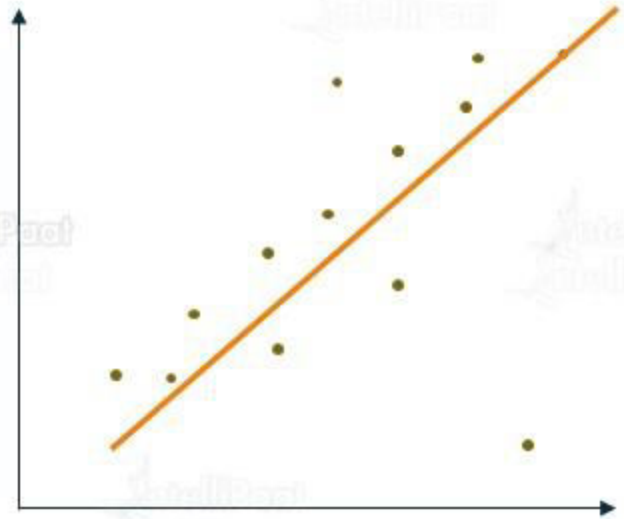
Linear Regression

The line with the lowest value of the residual sum of squares would be the best fit line

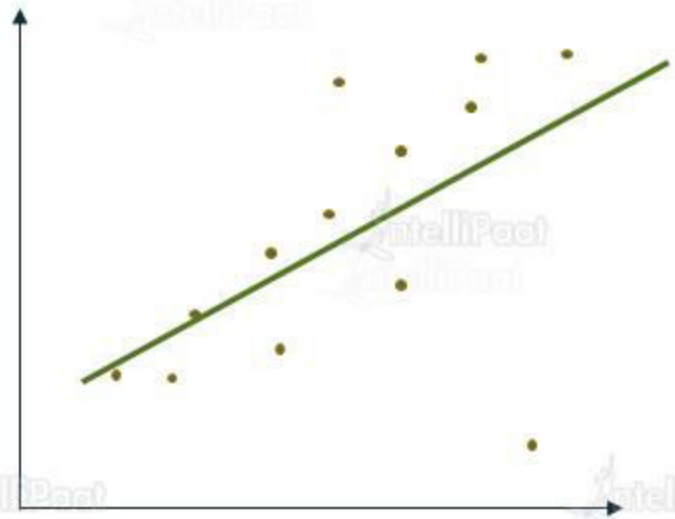
$$\text{RSS} = \sum_{k=1}^{n} (Actual - Predicted)^2$$



RSS = 120



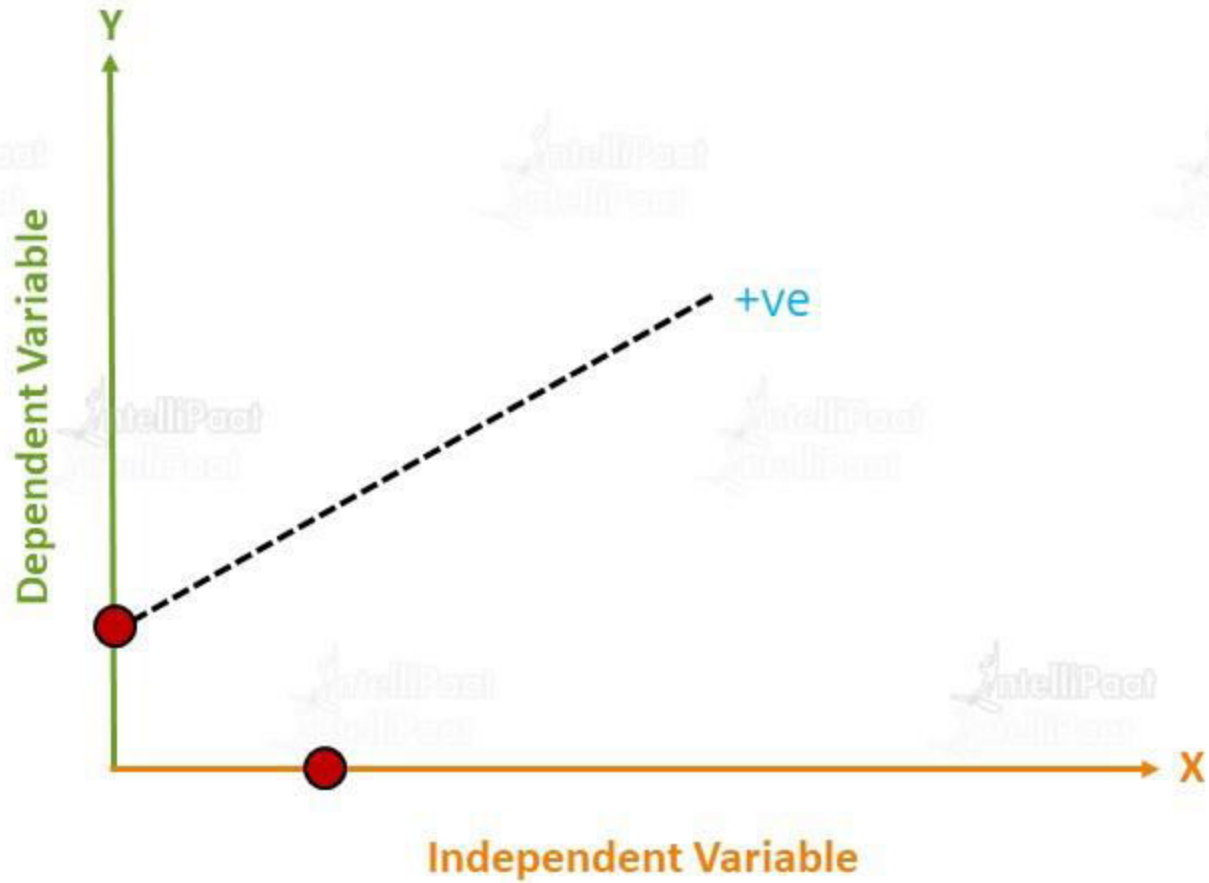
RSS = 80



RSS = 132

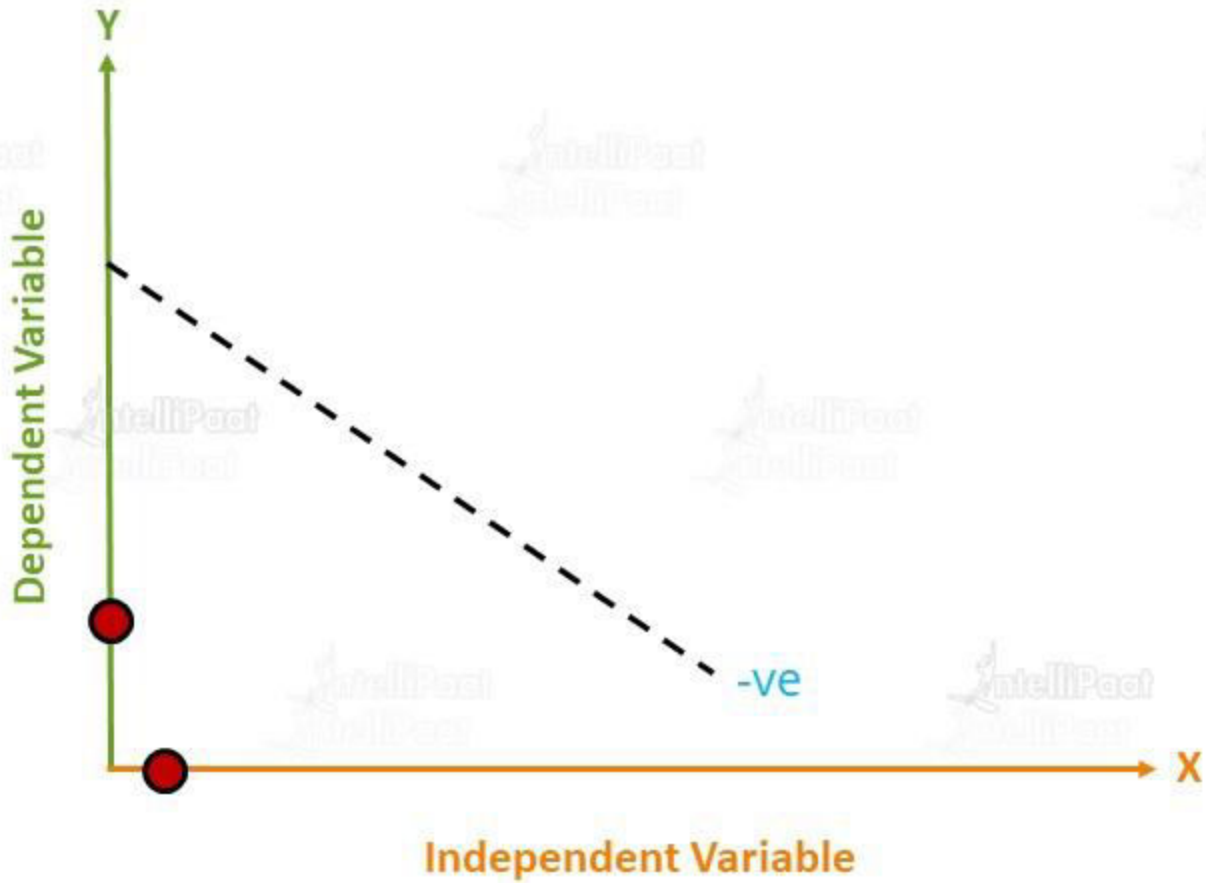
Linear Regression

Understanding Linear Regression Through an Example



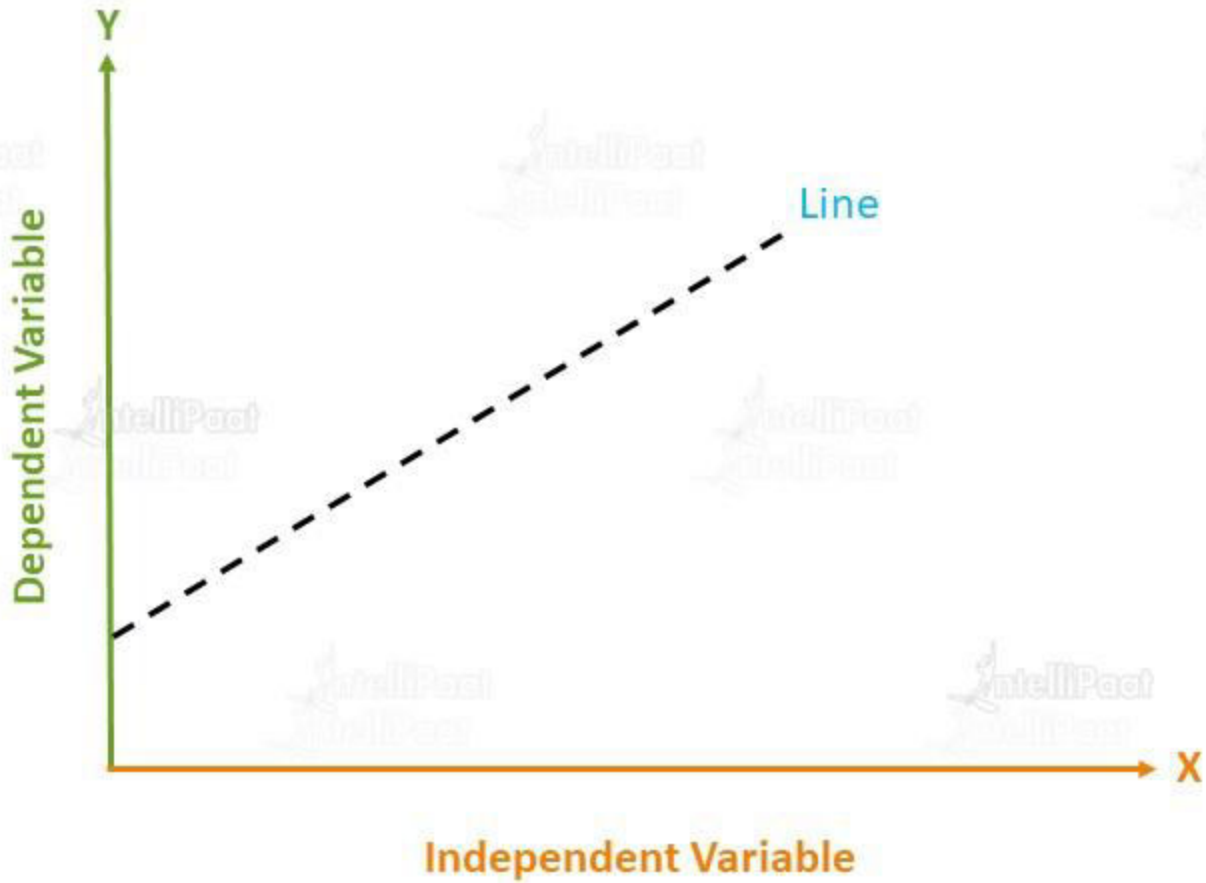
Linear Regression

Understanding Linear Regression Through an Example



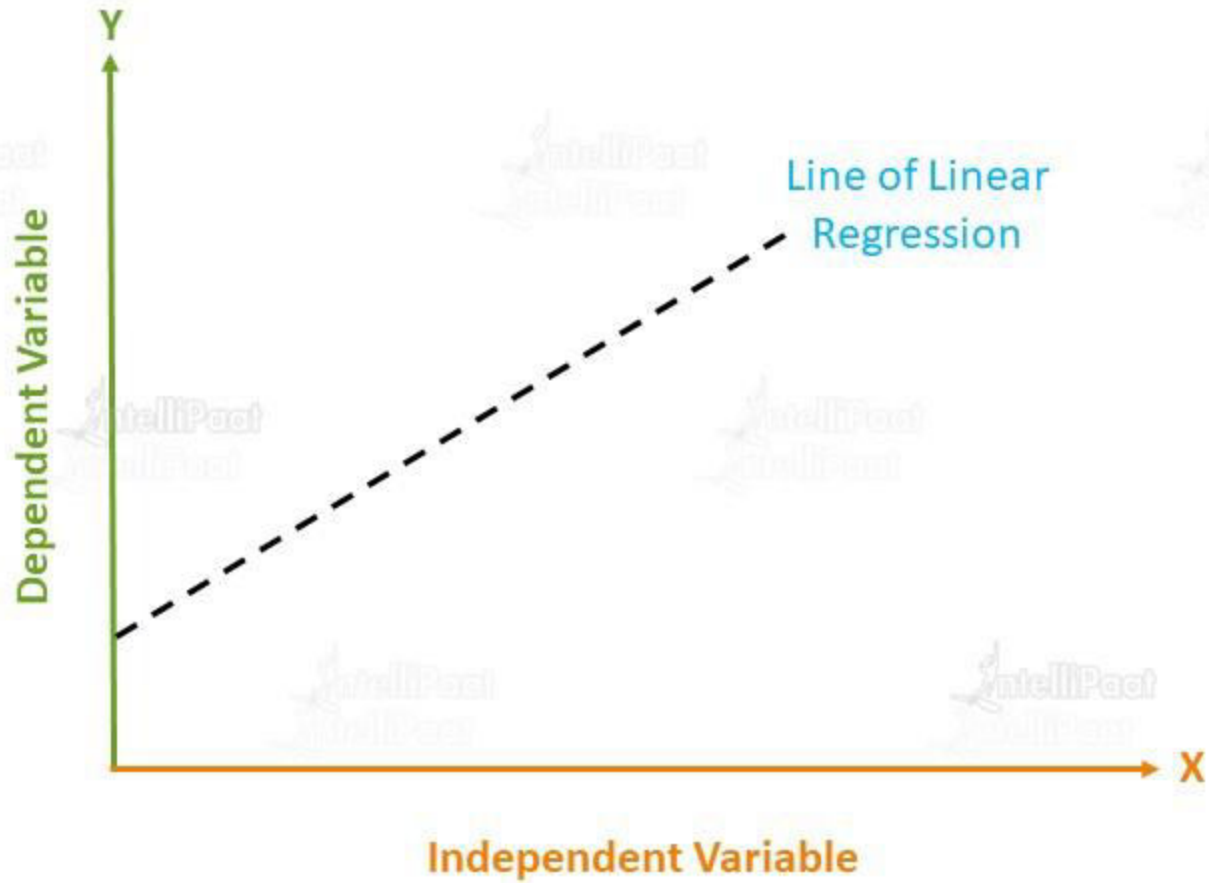
Linear Regression

Understanding Linear Regression Through an Example



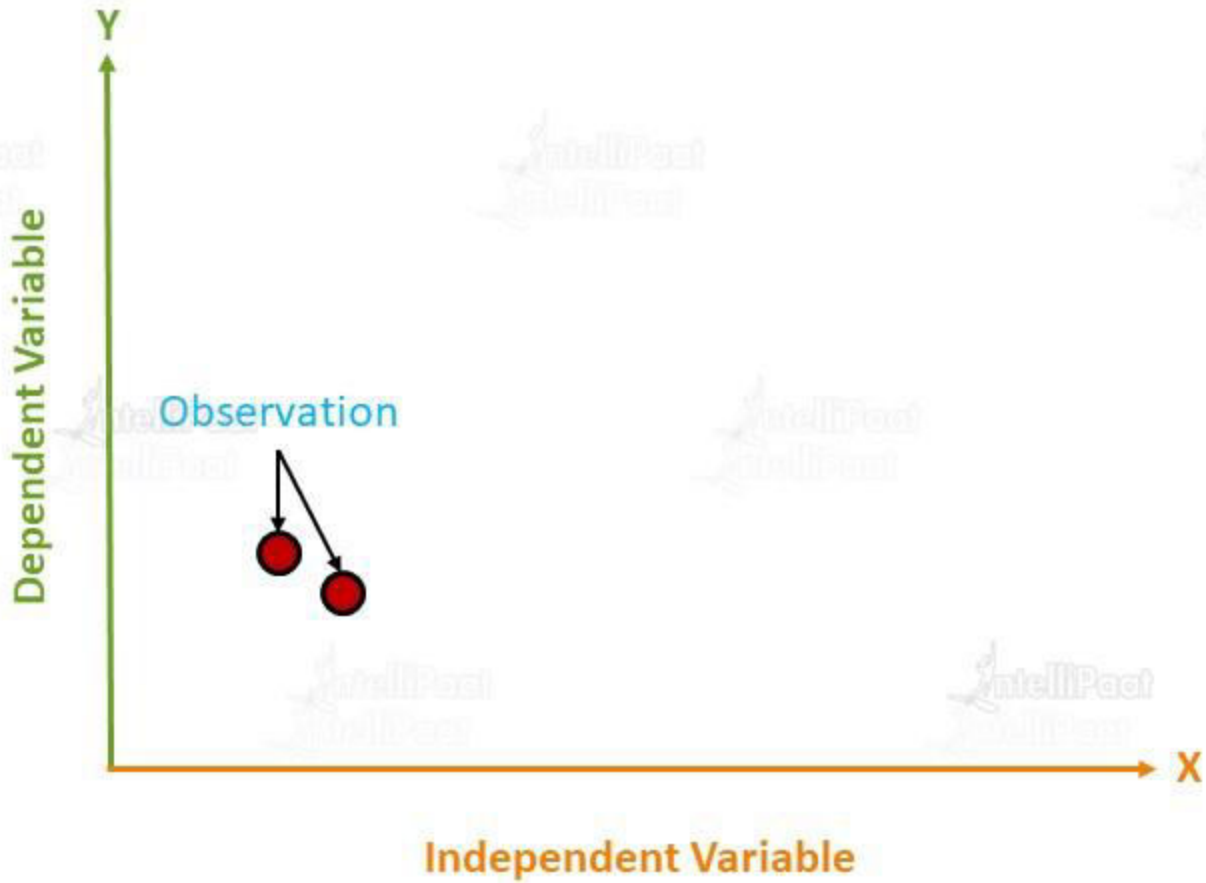
Linear Regression

Understanding Linear Regression Through an Example



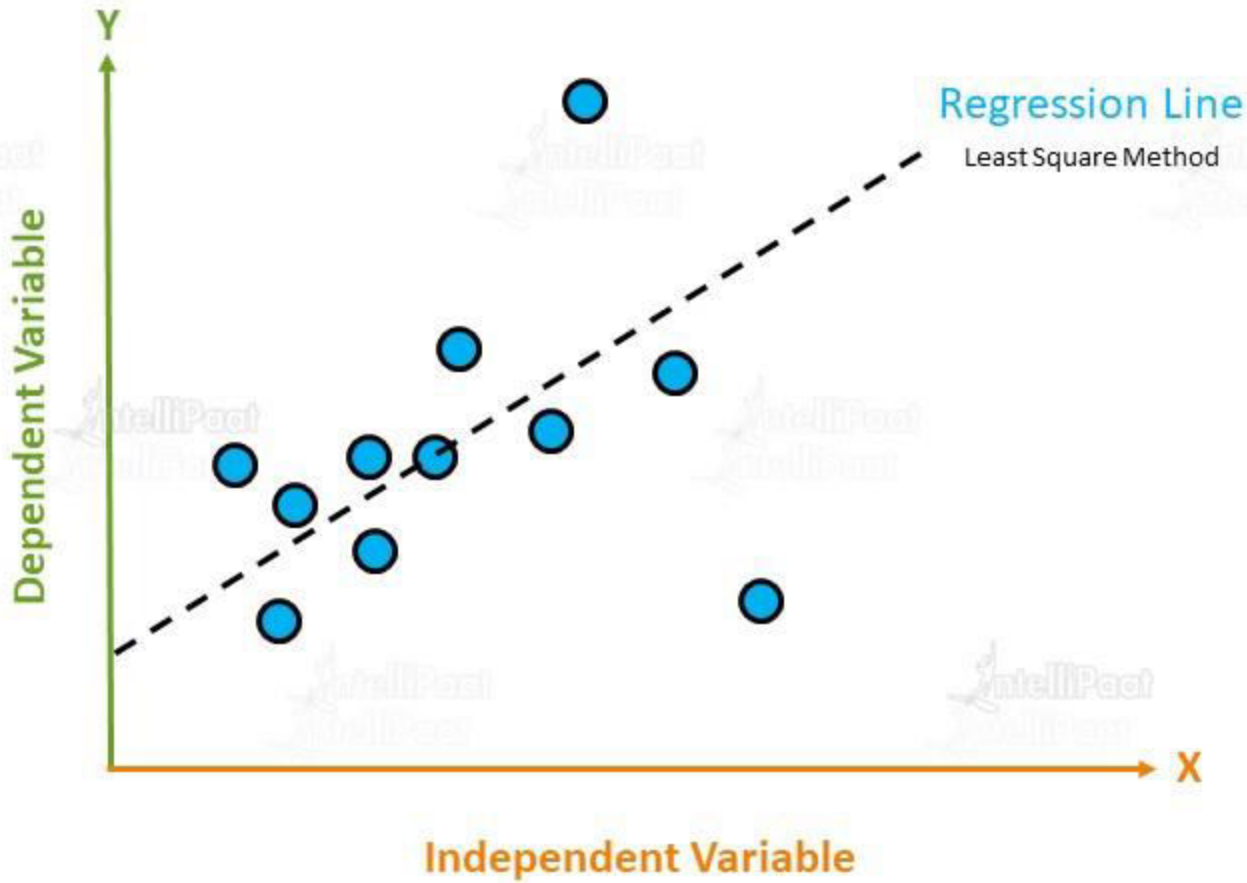
Linear Regression

Understanding Linear Regression Through an Example



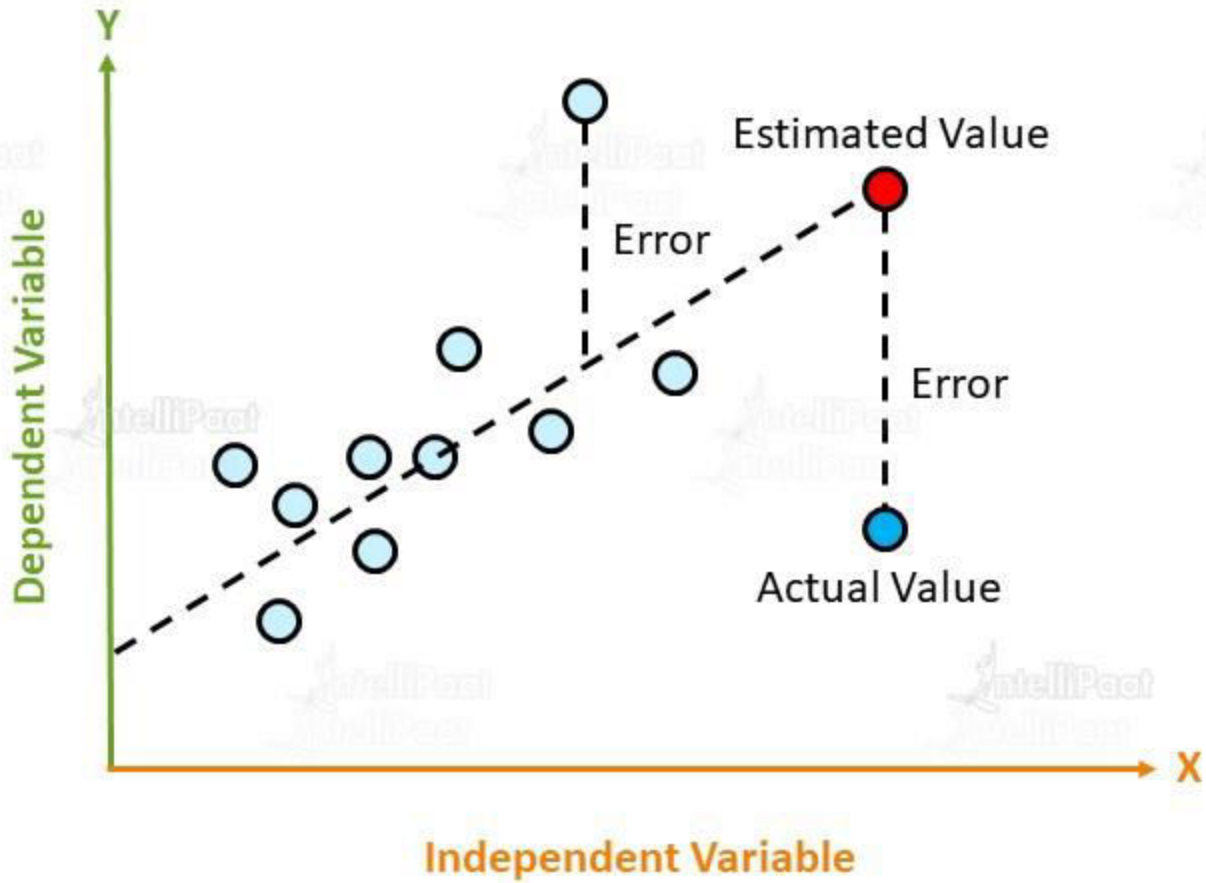
Linear Regression

Understanding Linear Regression Through an Example



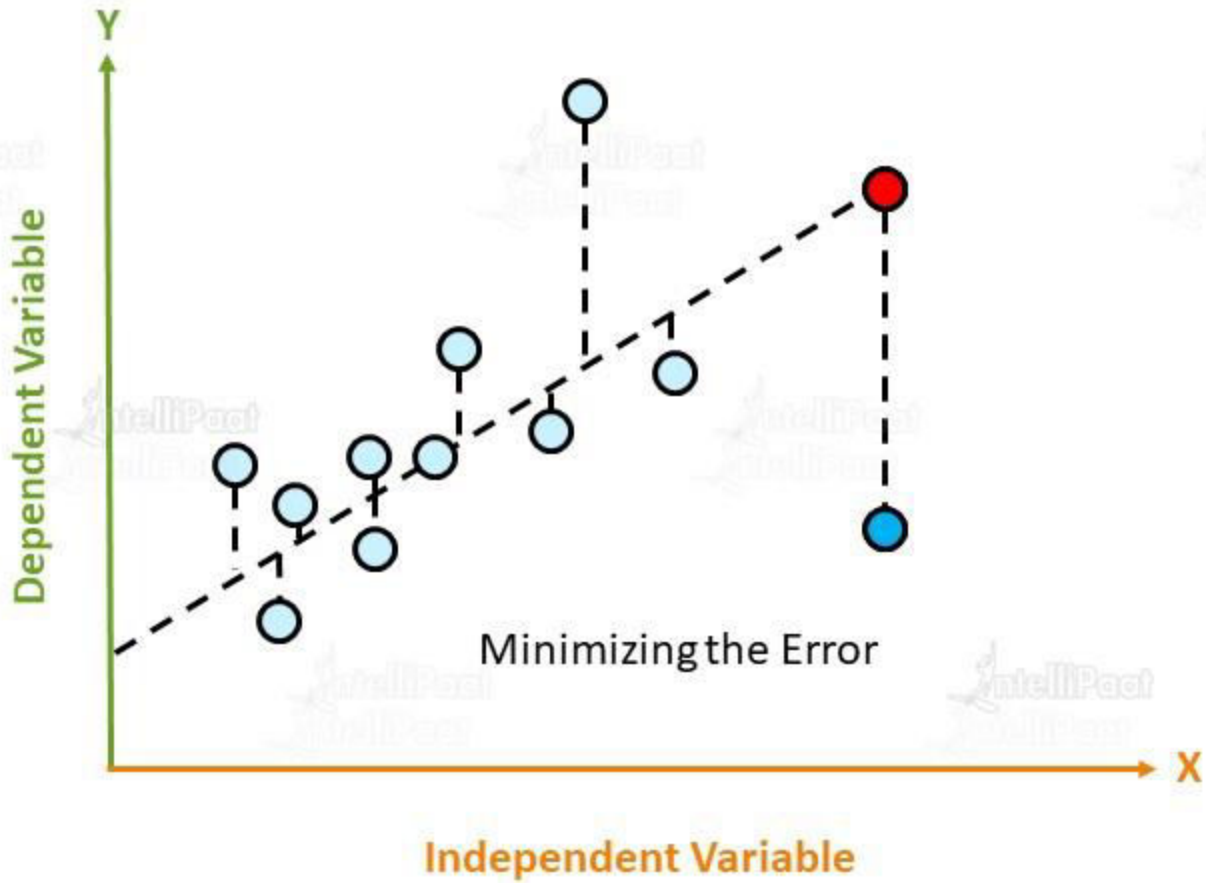
Linear Regression

Understanding Linear Regression Through an Example



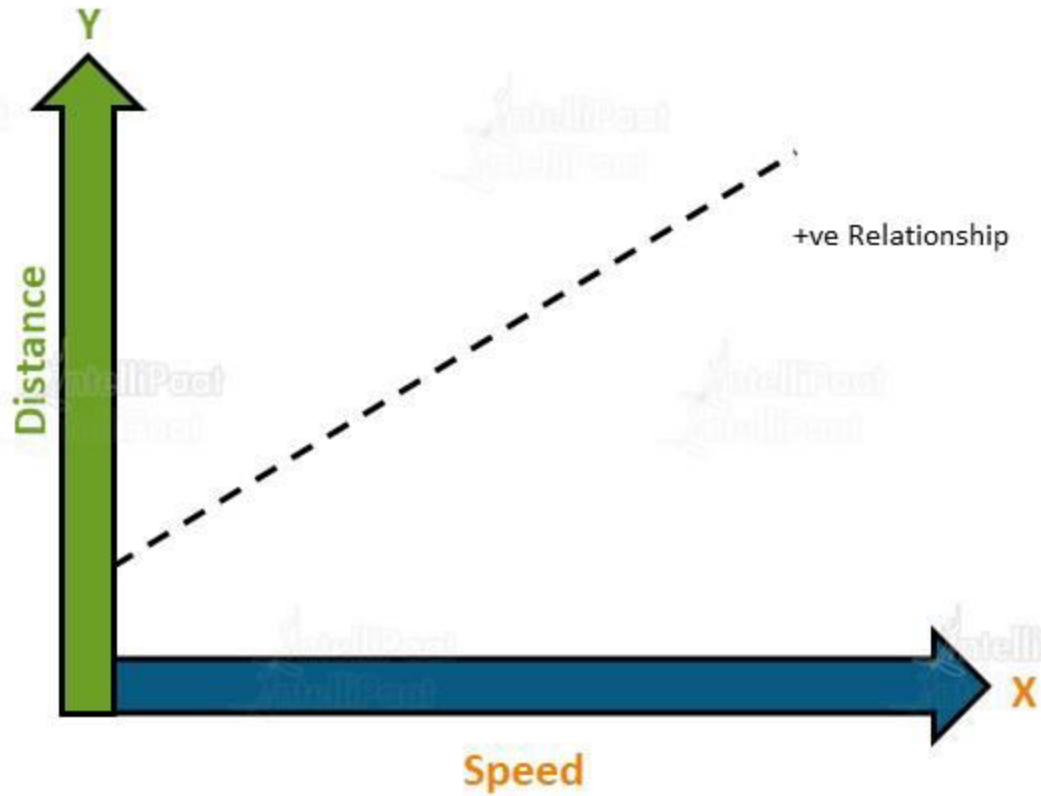
Linear Regression

Understanding Linear Regression Through an Example



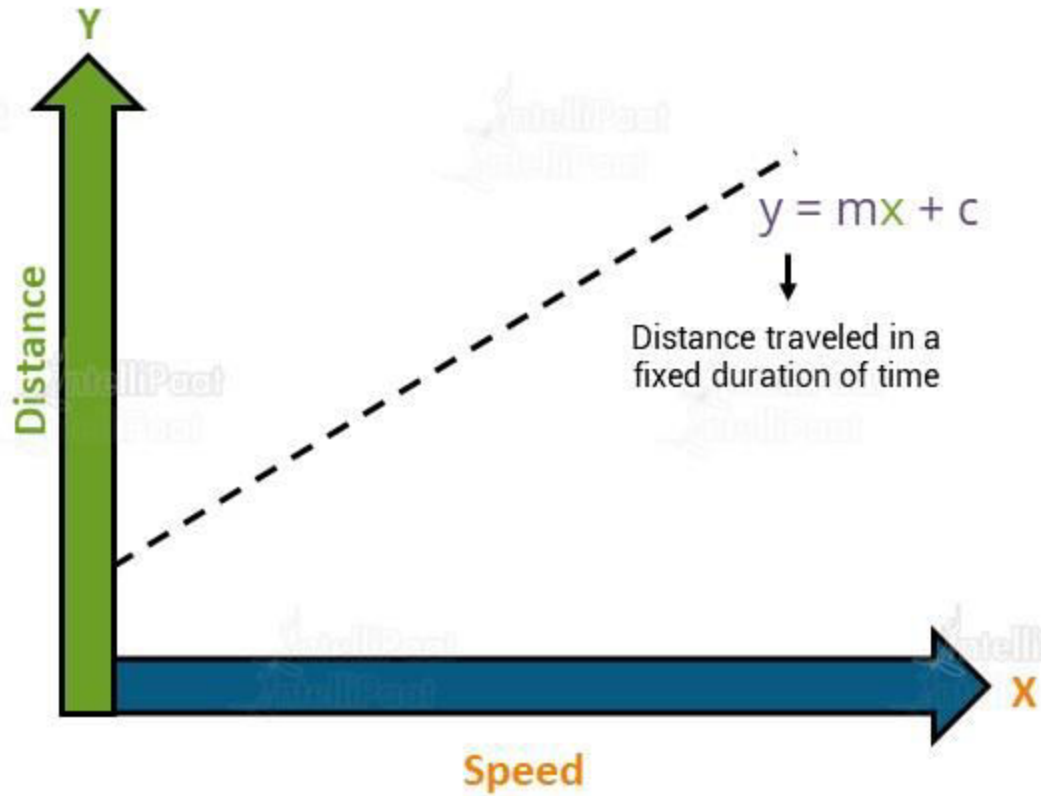
Linear Regression

Understanding Linear Regression Through an Example



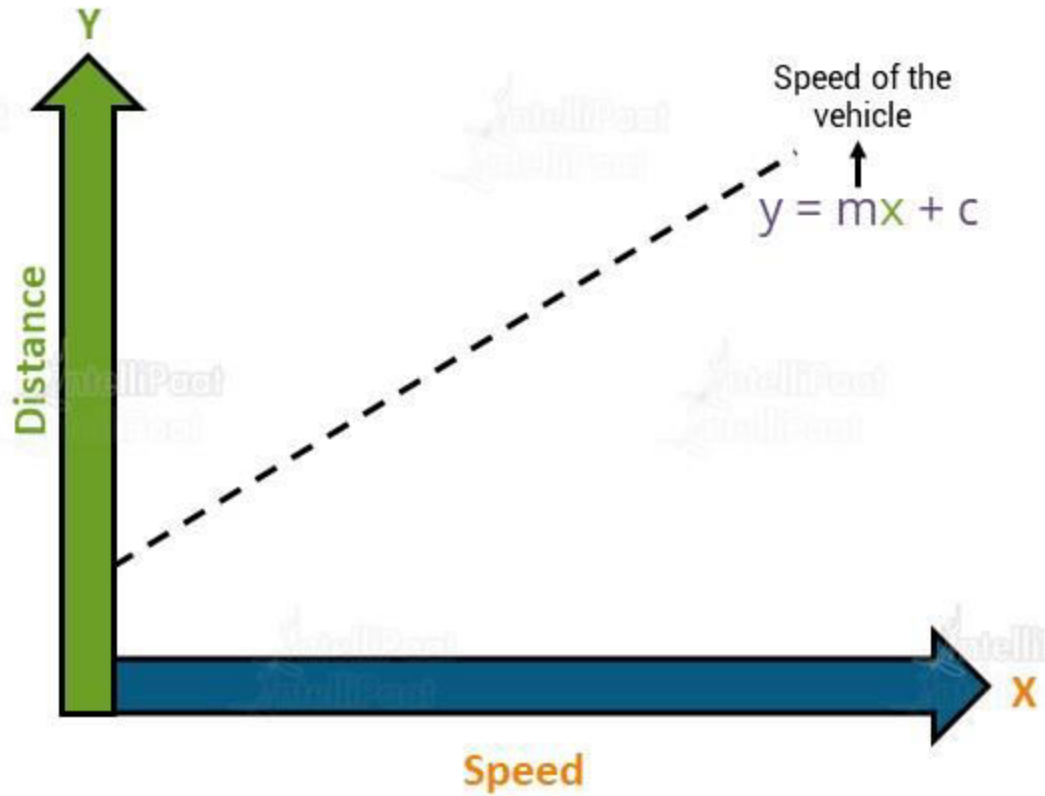
Linear Regression

Understanding Linear Regression Through an Example



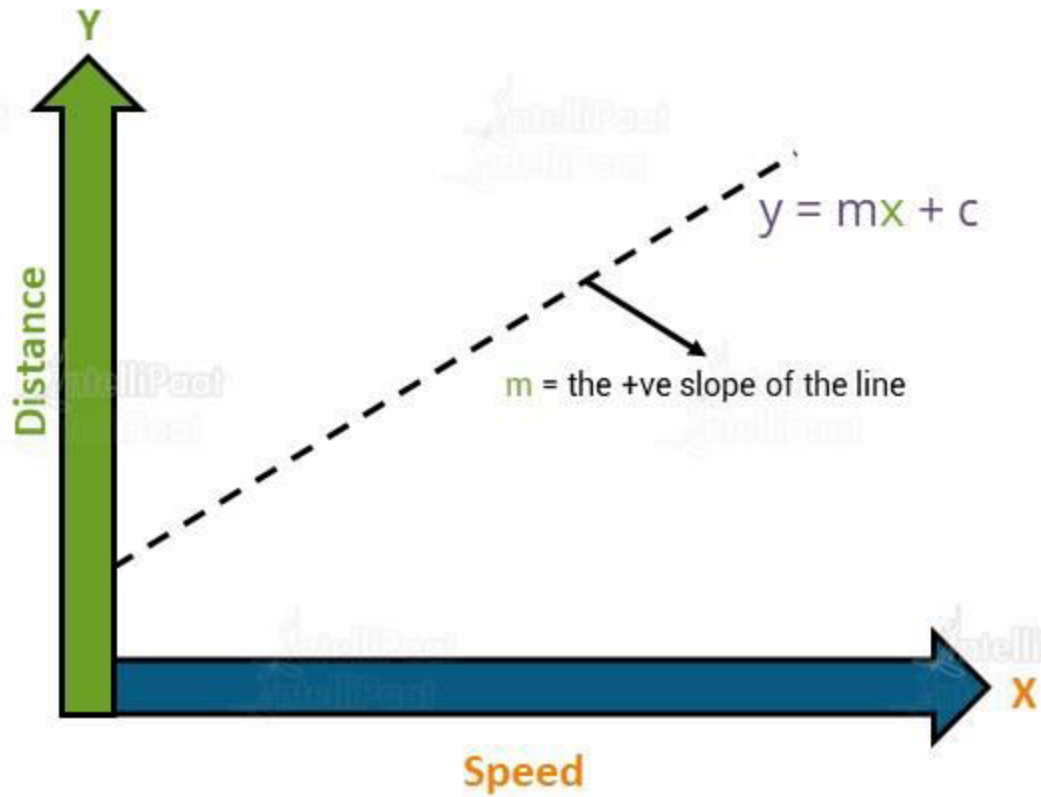
Linear Regression

Understanding Linear Regression Through an Example



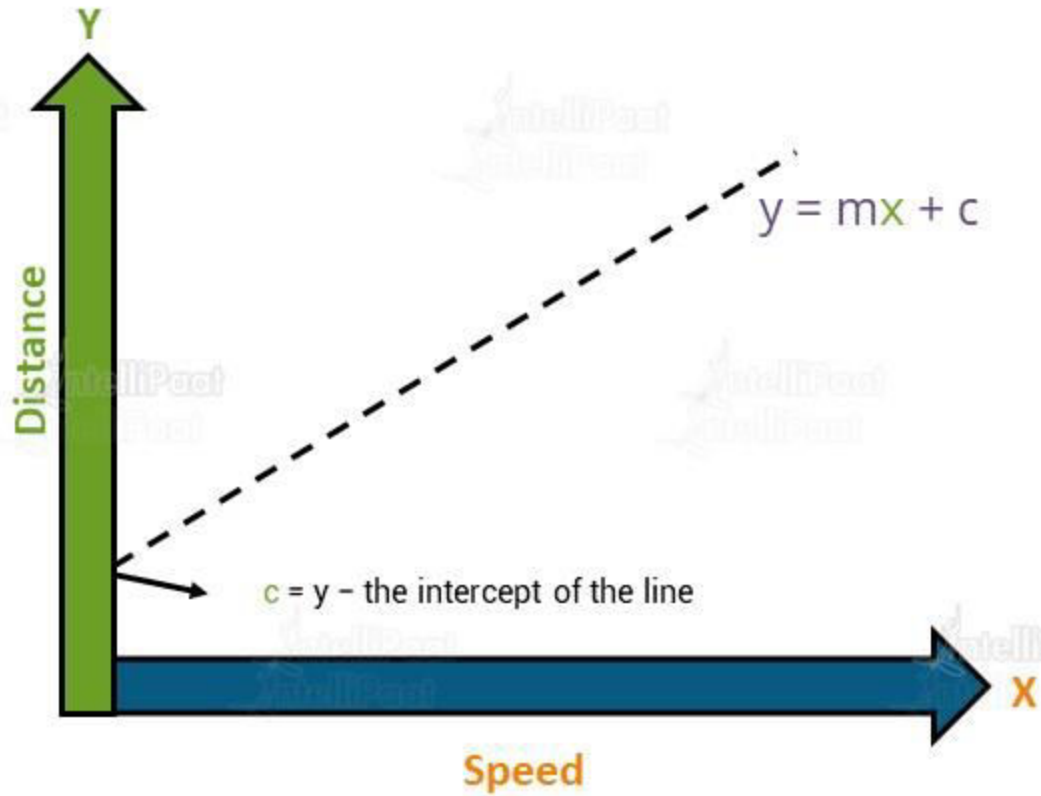
Linear Regression

Understanding Linear Regression Through an Example



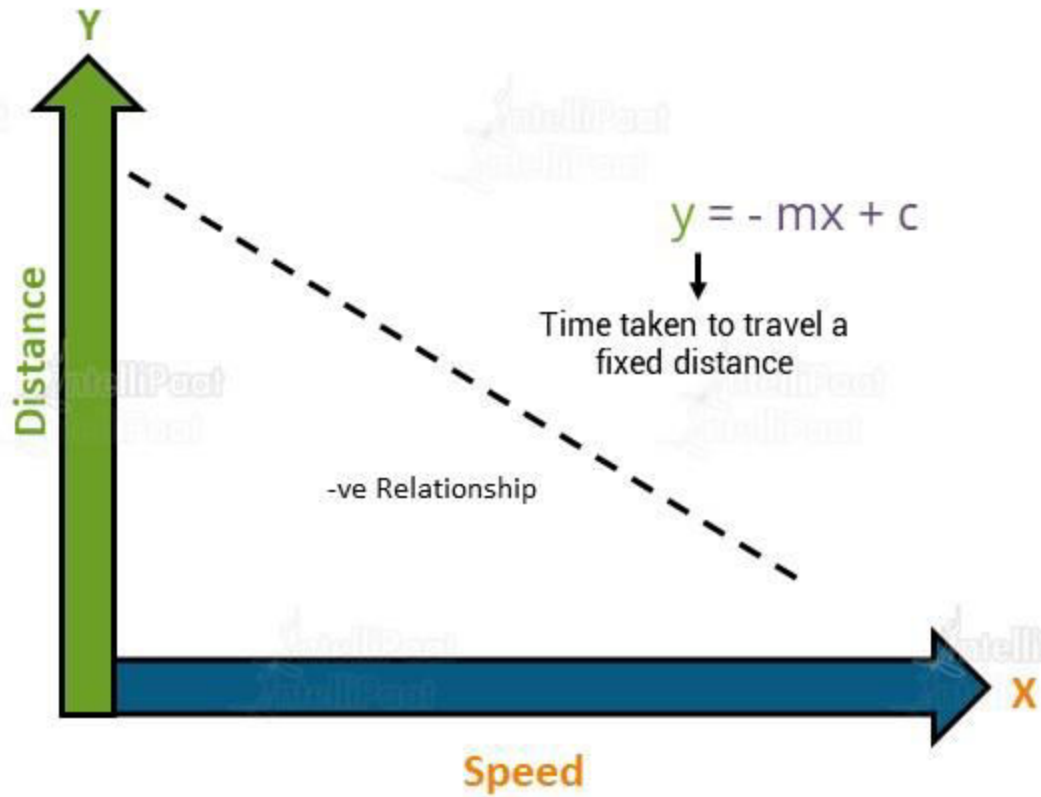
Linear Regression

Understanding Linear Regression Through an Example



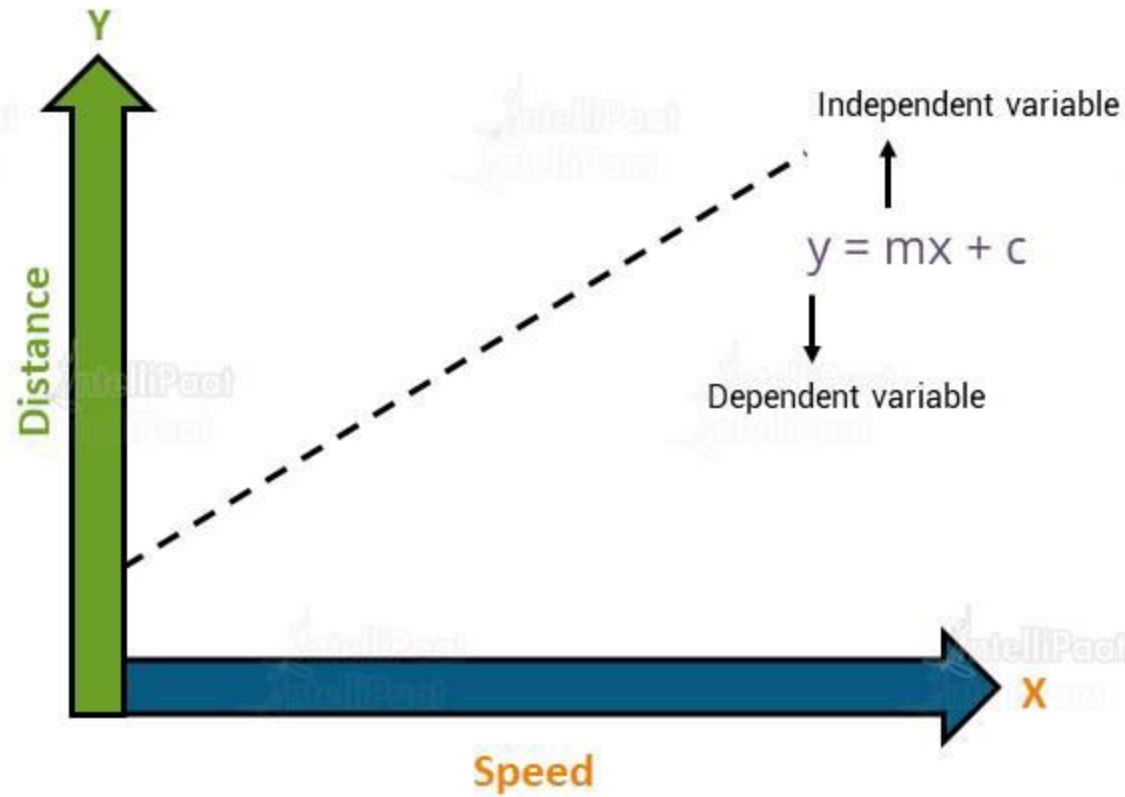
Linear Regression

Understanding Linear Regression Through an Example



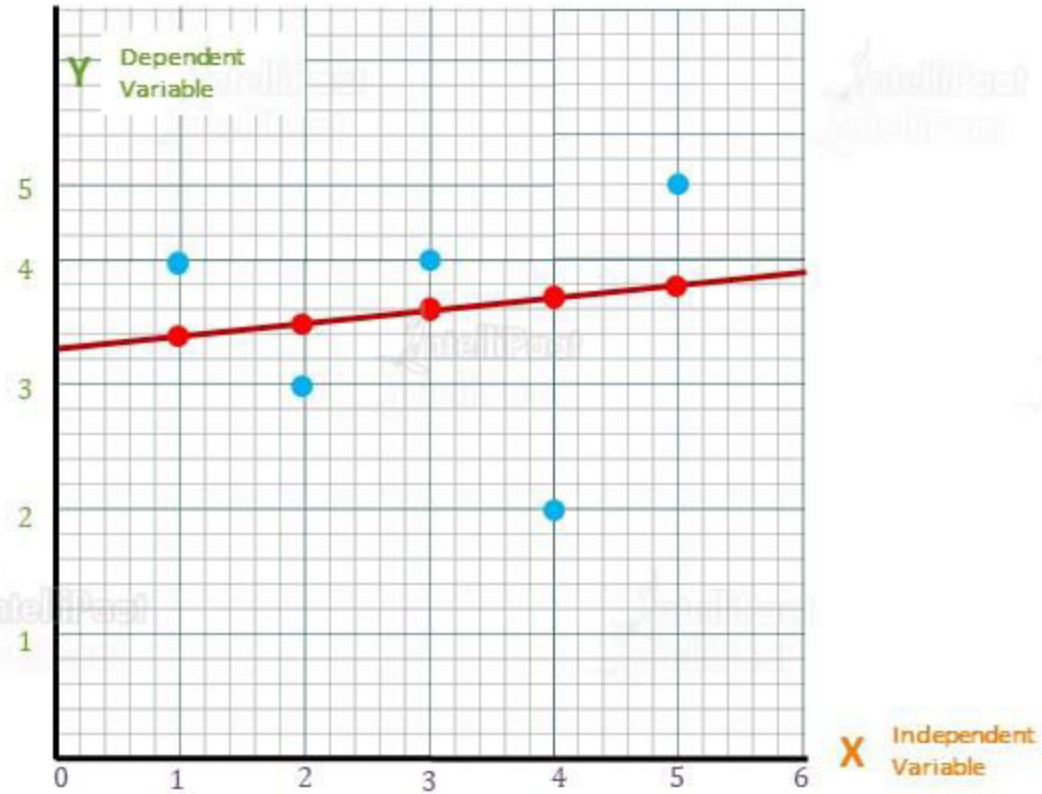
Linear Regression

Understanding Linear Regression Through an Example



Linear Regression

Mean Square Error



$m = 0.1$

$c = 3.3$

$y = 0.1x + 3.3$

For the given $\mathbf{m = 0.1}$ and $\mathbf{c = 3.3}$, let's predict the values for y, when $x = \{1,2,3,4,5\}$

$y = 0.1 \times 1 + 3.3 = 3.2$

$y = 0.1 \times 2 + 3.3 = 3.1$

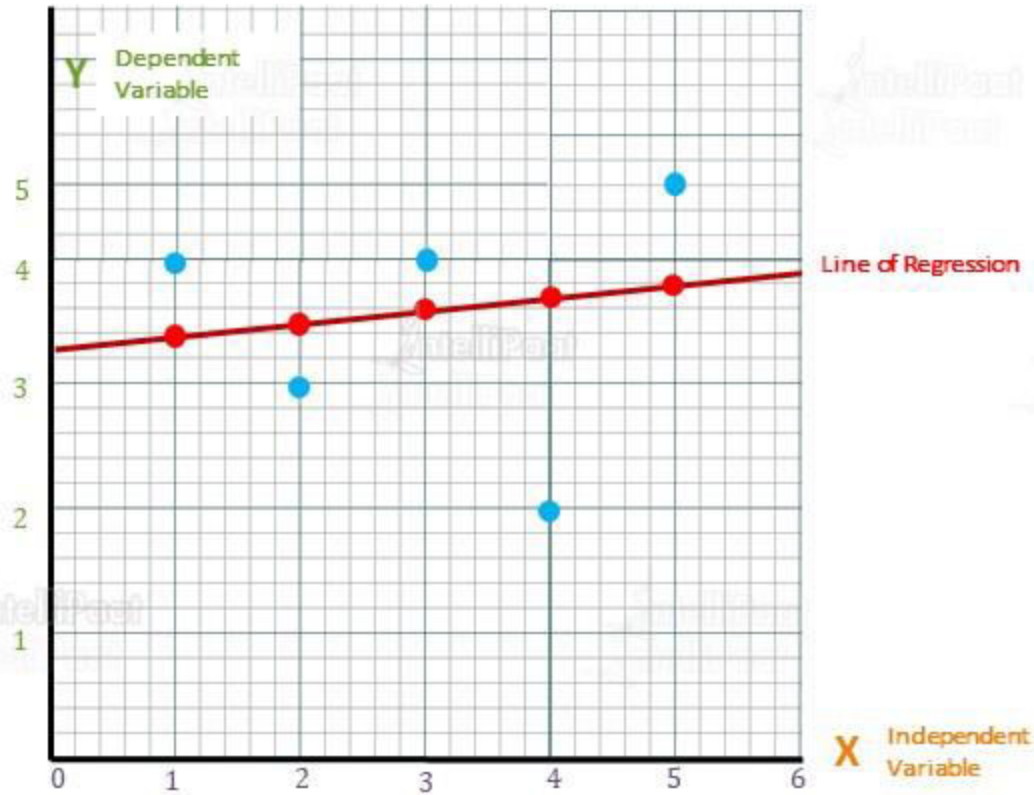
$y = 0.1 \times 3 + 3.3 = 3.0$

$y = 0.1 \times 4 + 3.3 = 2.9$

$y = 0.1 \times 5 + 3.3 = 2.8$

Linear Regression

Mean Square Error



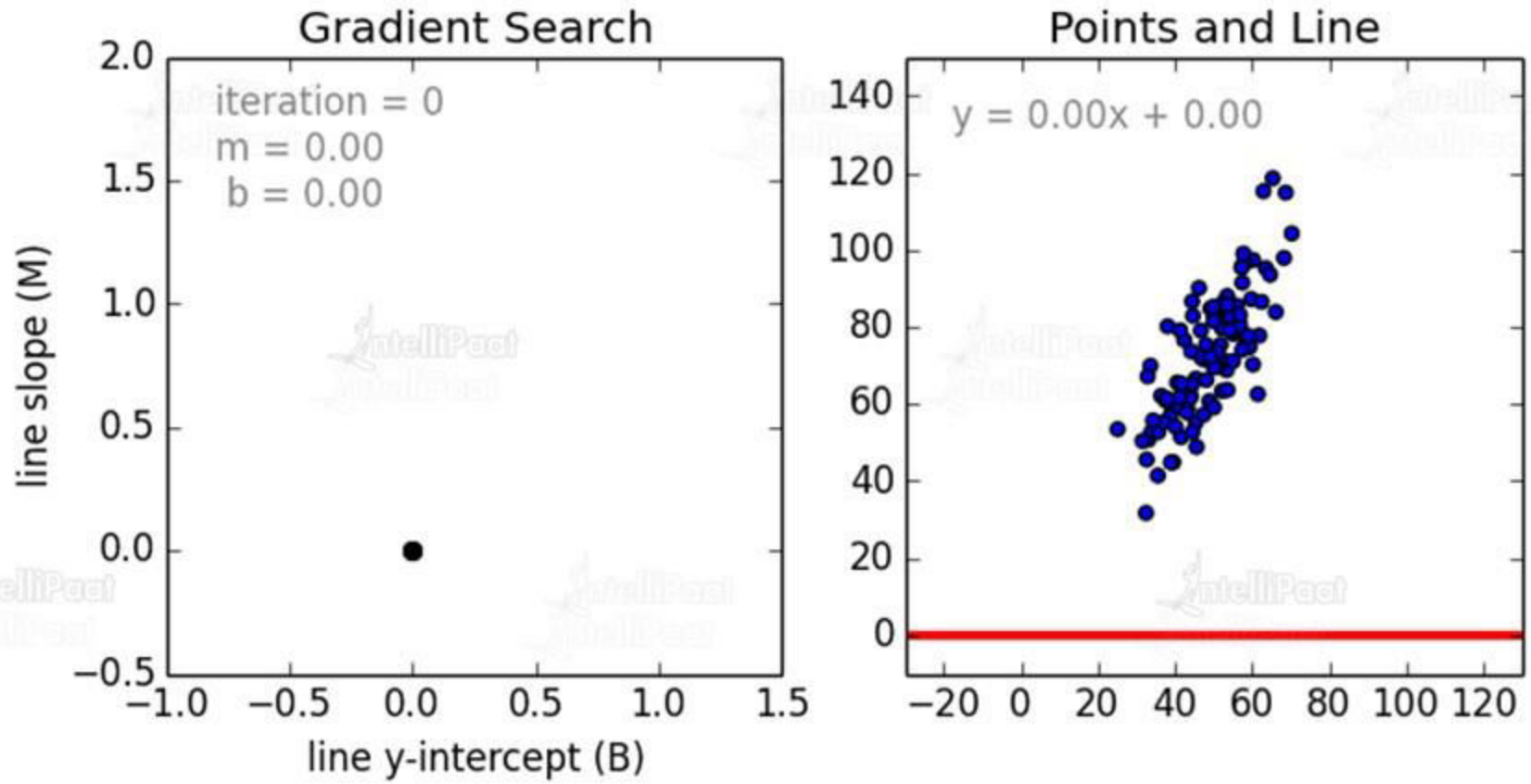
$$m = 0.1$$
$$c = 3.3$$
$$y = 0.1x + 3.3$$

For the given $\mathbf{m = 0.1}$ and $\mathbf{c = 3.3}$, let's predict the values for y, when $x = \{1,2,3,4,5\}$

$$y = 0.1 \times 1 + 3.3 = 3.4$$
$$y = 0.1 \times 2 + 3.3 = 3.5$$
$$y = 0.1 \times 3 + 3.3 = 3.6$$
$$y = 0.1 \times 4 + 3.3 = 3.7$$
$$y = 0.1 \times 5 + 3.3 = 3.8$$

Linear Regression

Finding the best fit line



Linear Regression

Goodness of Fit – R^2

R-squared is interpreted as the proportion of the variance in the dependent variable that is predictable from the independent variable

An **R-squared of 0** means that the dependent variable cannot be predicted from the independent variable

An **R-squared of 1** means that the dependent variable can be predicted without any error from the independent variable

An **R-squared between 0 and 1** indicates the extent to which the dependent variable is predictable. An R-squared of 0.10 means that 10% of variance in Y is predictable from X; an R-squared of 0.20 means that 20% is predictable, and so on

Linear Regression

Adjusted R-squared

1. The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model
2. The adjusted R-squared increases only if the new term improves the model more than it would be expected by chance
3. The adjusted R-squared will always be less than or equal to the R-squared

Hands-on: Linear Regression

Multiple Linear Regression

Multiple Linear Regression

Multiple linear regression uses several explanatory variables to predict the outcome of a response variable. We call it 'multiple' because in this case, unlike in simple linear regression, we have many independent variables trying to predict a dependent variable

Formula for Multiple Linear Regression: $Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \varepsilon$

Y_i = Dependent Variable

x_i = Explanatory Variables

β_0 = y-intercept (a constant term)

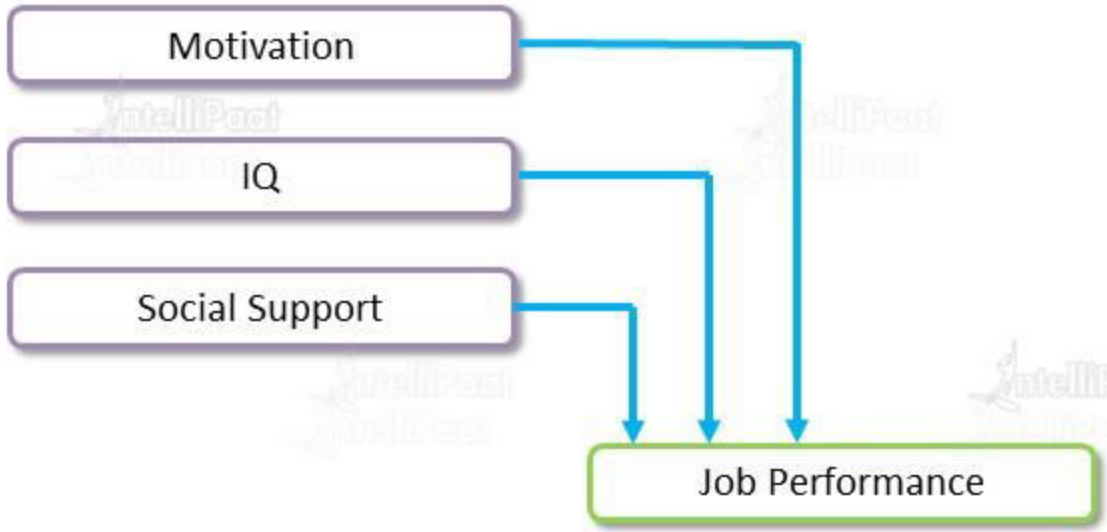
β_p = Slope coefficients for each explanatory variable

ε = Model's error term (also known as the residual)

Multiple Linear Regression

Understanding Multiple Linear Regression Through an Example

Suppose, you own a company and you want to know how your employees' job performance relates to their IQ, their motivation, and the amount of social support they receive



Multiple Linear Regression

Understanding Multiple Linear Regression Through an Example

Regression analysis provides numeric estimates of the strengths of such relations. To use regression analysis, here, we need data on the 4 variables (1 criterion and 3 predictors) in the model. Therefore, we will have the employees take some tests that measure these

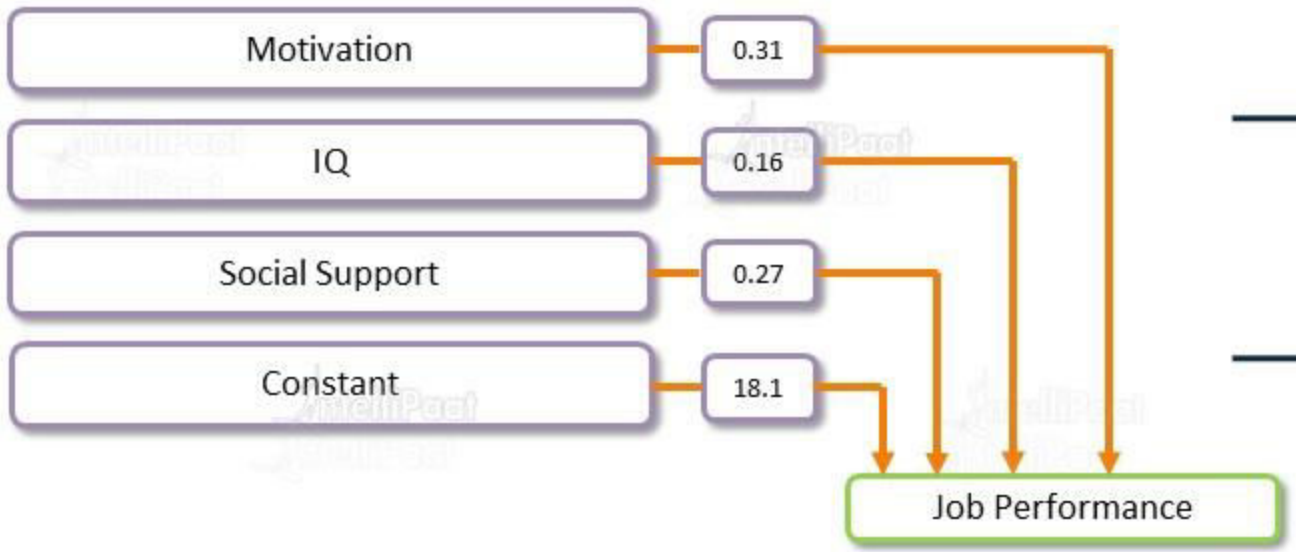
Name	IQ	Mot	Soc
Sam	110	74	73
BOb	97	92	67
Anne	125	56	80

Here we see that, Intelligence Quotient (IQ), Motivation (Mot), and Social Support (Soc) have been measured in a numerical form

Multiple Linear Regression

Understanding Multiple Linear Regression Through an Example

Now, let's apply linear regression on this data and find the β coefficients

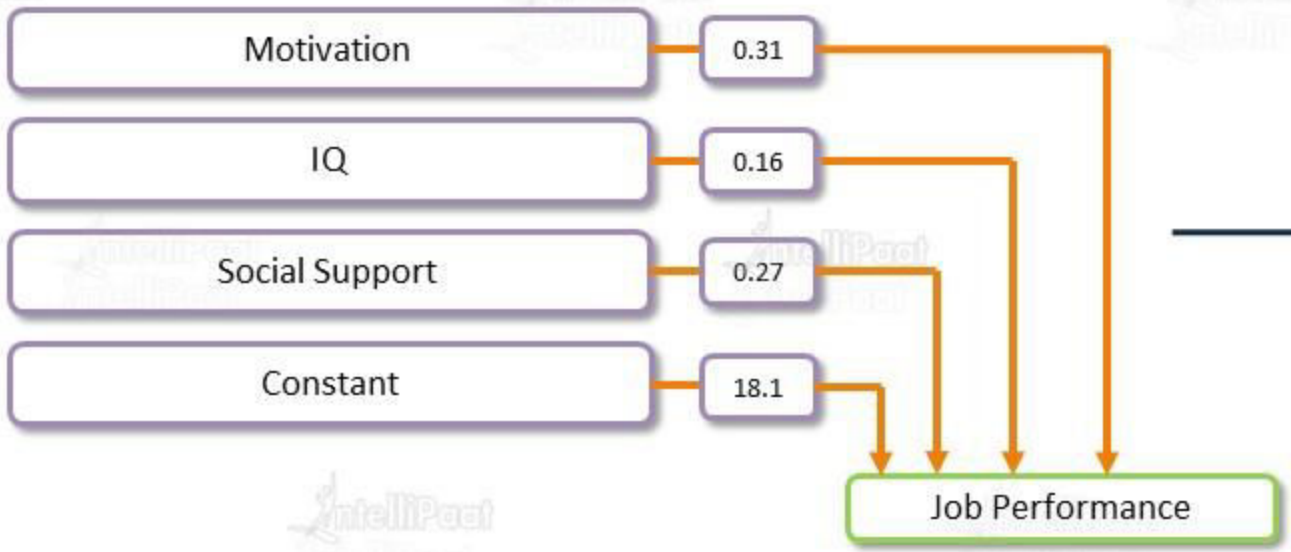


Job Performance = (0.31*Motivation) + (0.16*IQ) + (0.27*Social Support) + 18.1

This formula shows how job performance is estimated: we add up each of the predictor scores after multiplying them with some number. These numbers are known as **β coefficients** or unstandardized regression coefficients

Multiple Linear Regression

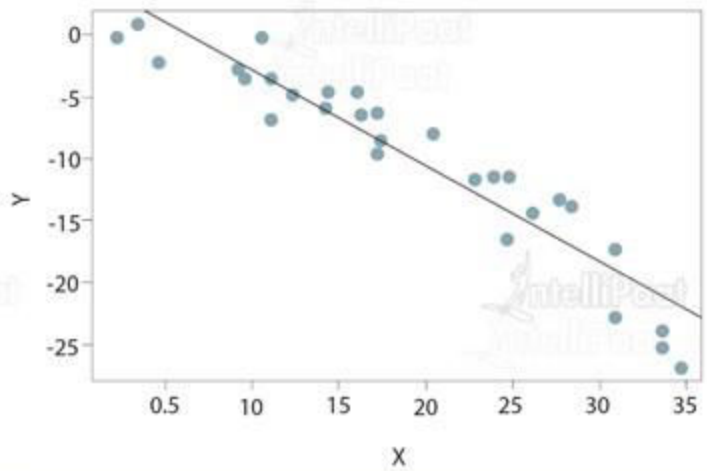
Understanding Multiple Linear Regression Through an Example



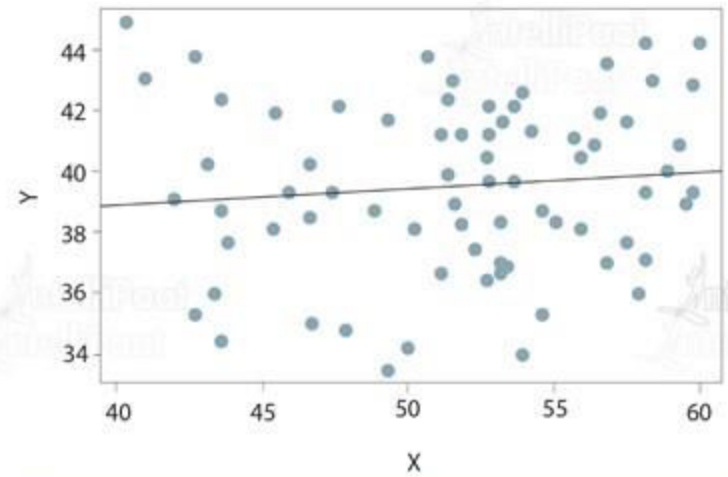
In this model, **18.1** is a baseline score that is unrelated to any other variable. It is a constant over respondents, i.e., it is the same 18.1 points for each respondent

Assumptions in Linear Regression

1. There should be a linear and additive relationship between the dependent and independent variables



Satisfies the assumption

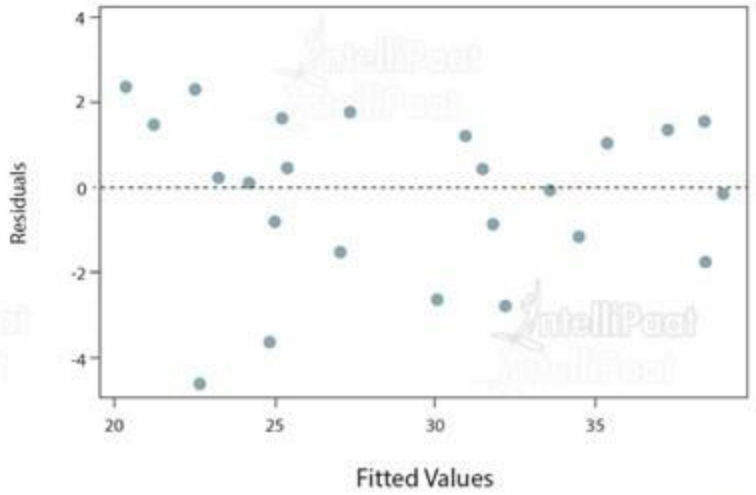


Does not satisfy the assumption

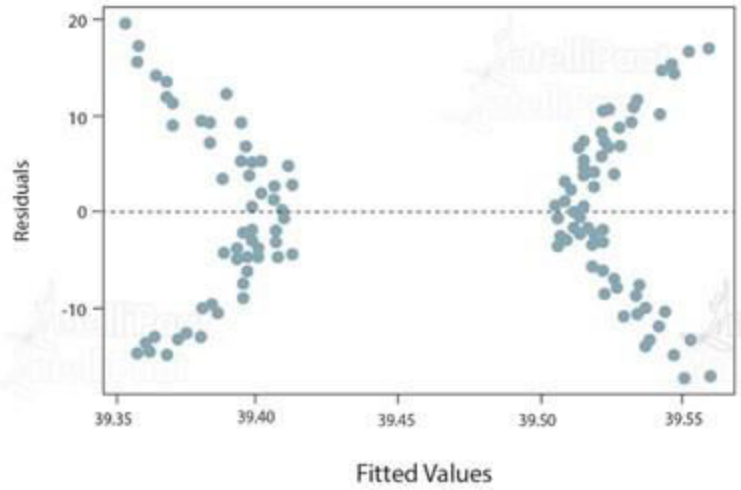
Assumptions in Linear Regression

2

Residuals must have constant variance



If there is no pattern, data is random and, hence, satisfies the condition

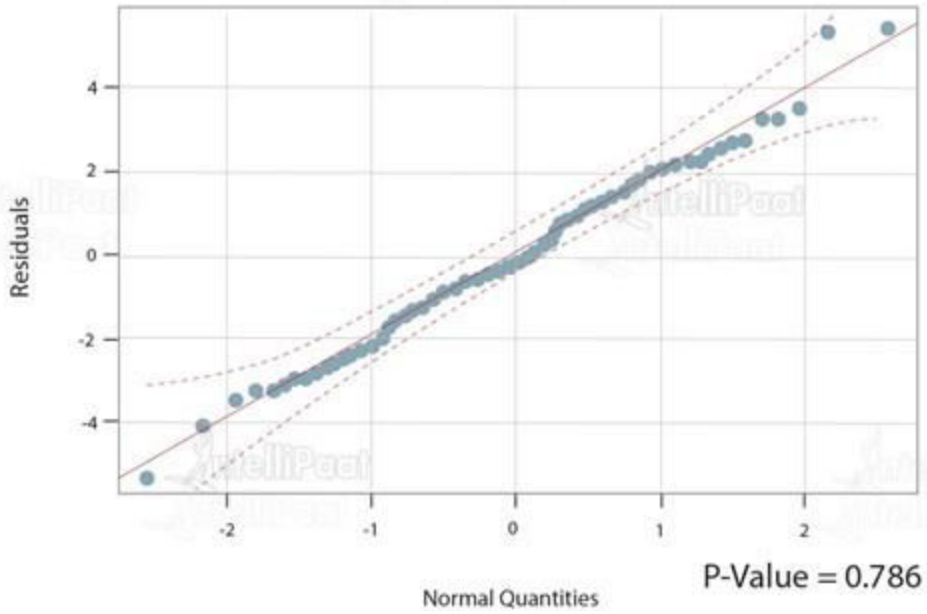


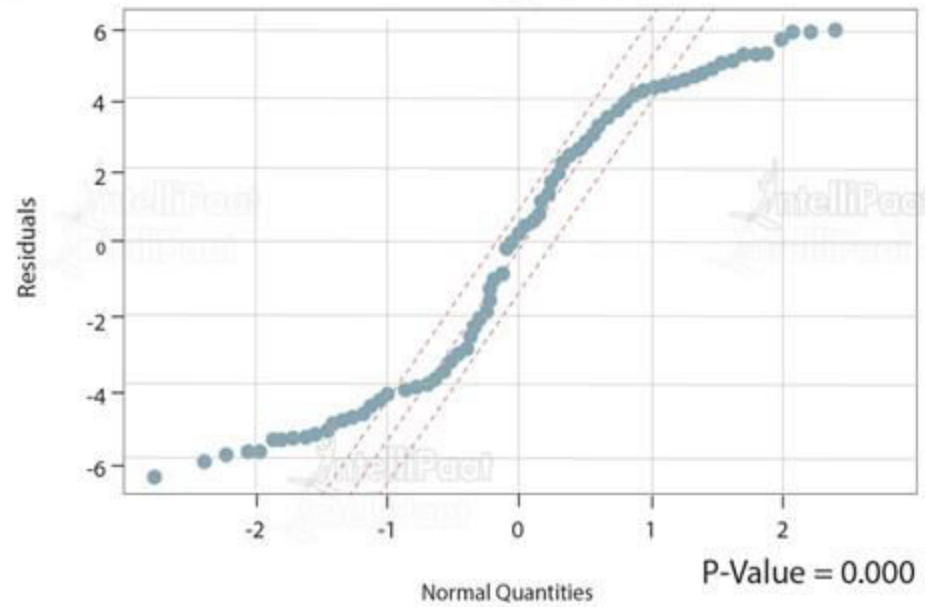
If there is a pattern, the data is not random and, hence, does not satisfy the condition

Assumptions in Linear Regression

3

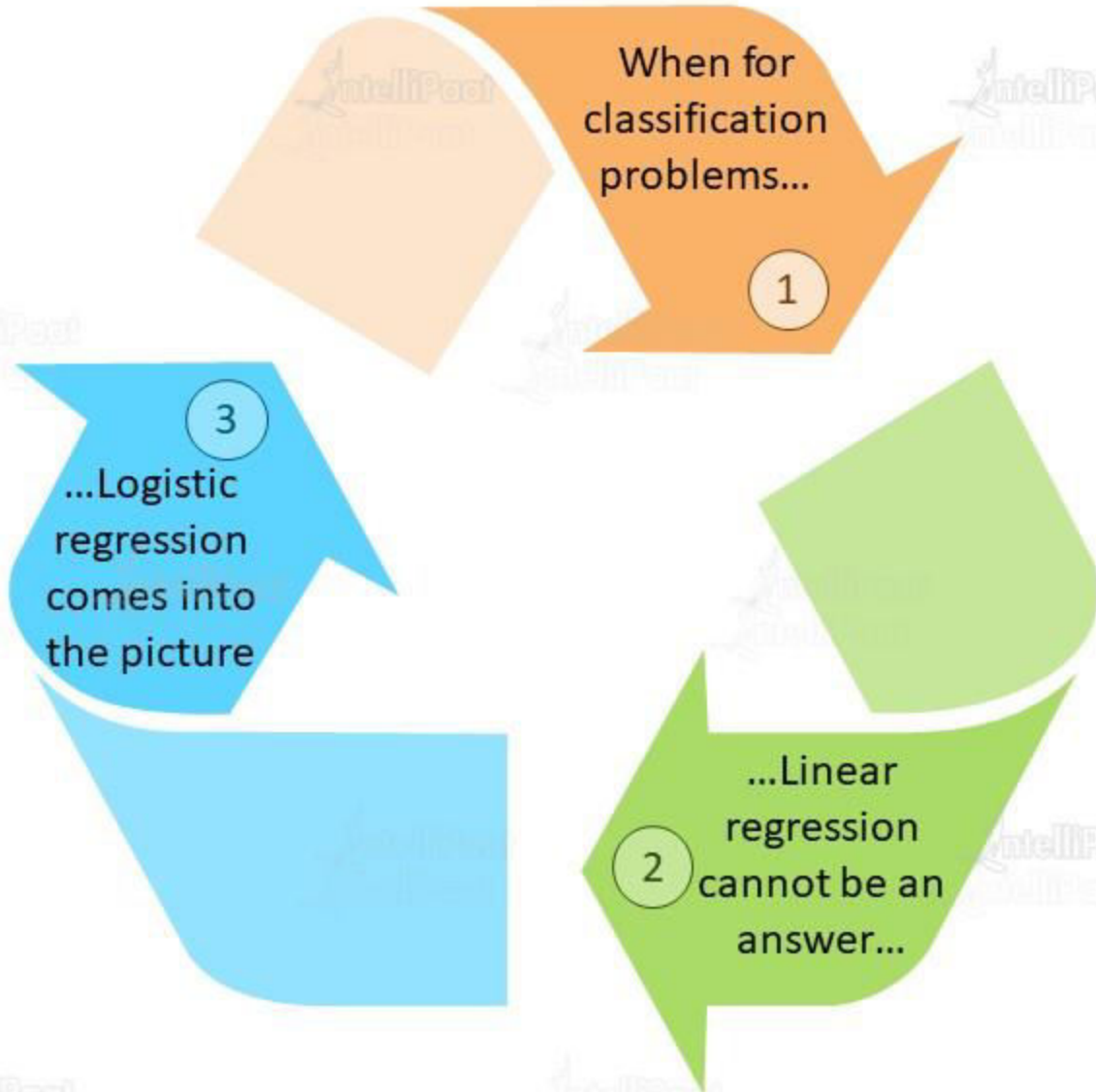
The residuals must be normally distributed



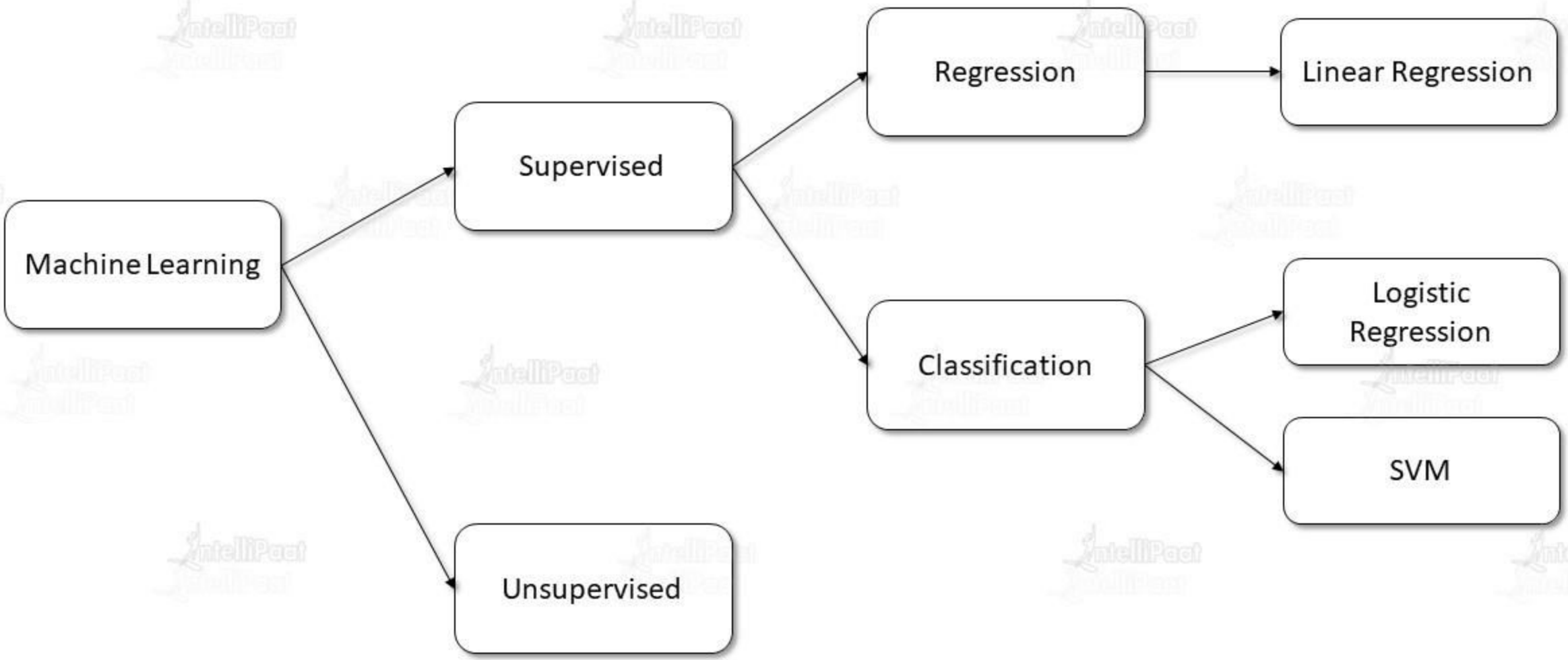


Logistic Regression

Logistic Regression



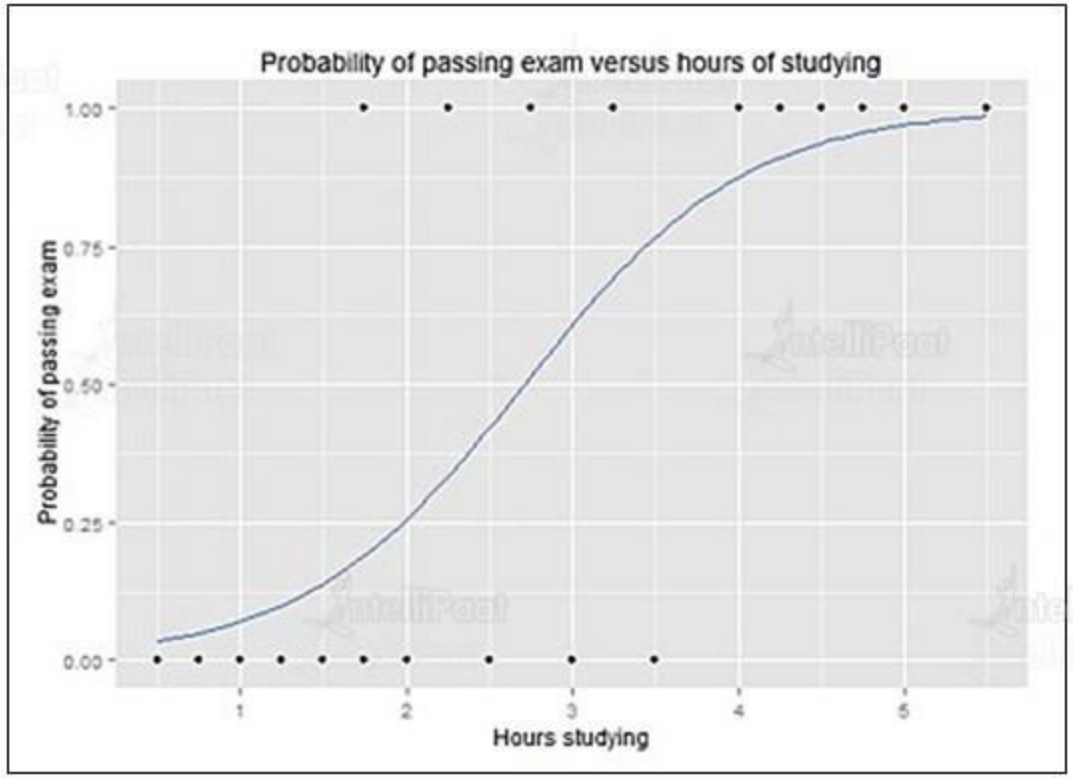
Logistic Regression



Logistic Regression

What is Logistic Regression?

1. A statistical classification model
2. Deals with categorical dependent variables
3. Could be binary or dichotomous
4. Could be multinomial
5. Takes both continuous and discrete input data



Logistic Regression

Why Logistic Regression?

1. A tool for applied statistics and discrete data analysis
2. Gives the outcome in terms of probability
3. Helps in classifying the given data

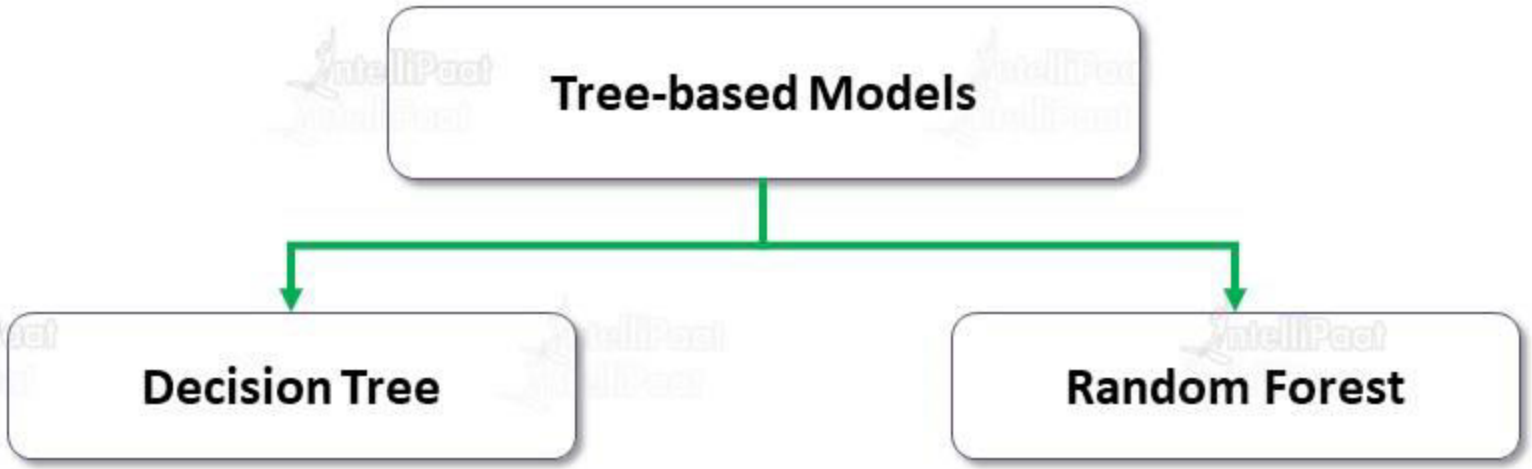
Hands-on: Logistic Regression

Decision Tree and Random Forest

Decision Tree and Random Forest

Tree-based models are said to be the most used supervised learning algorithms. They can be used for both classification and regression problems

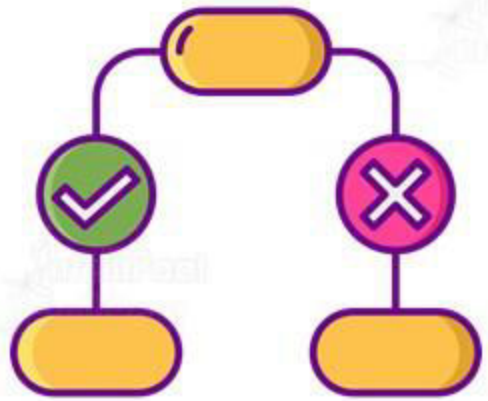
We can create highly accurate and stable predictive models



Decision Tree and Random Forest

Decision Tree

A graphical representation of all possible solutions to a decision, a decision tree is mostly used on classification problems. Decisions are mainly based on some conditions. Decisions made can easily be explained

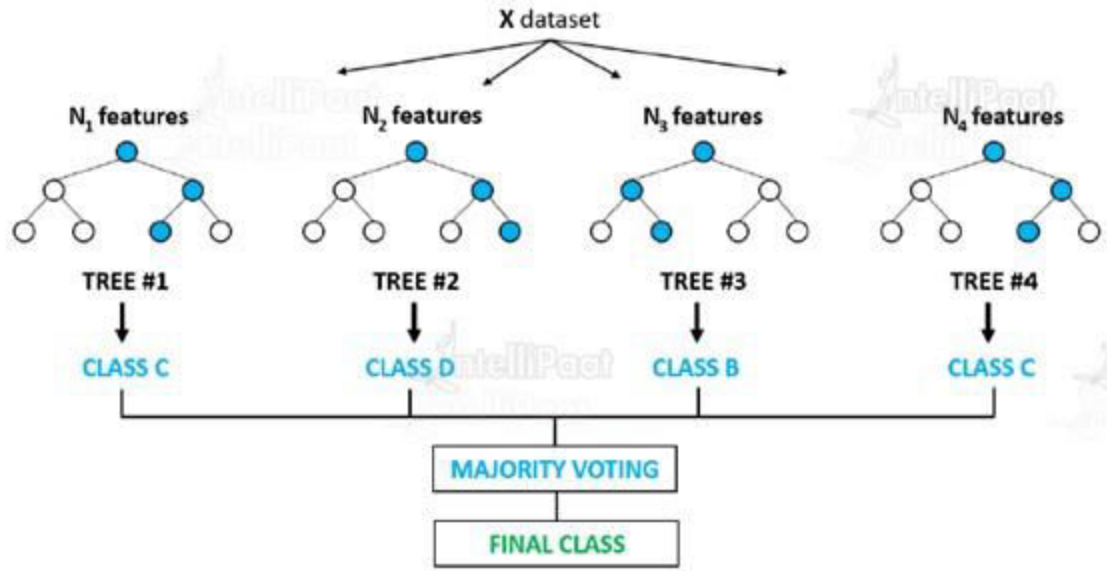


Decision Tree and Random Forest

Random Forest

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction

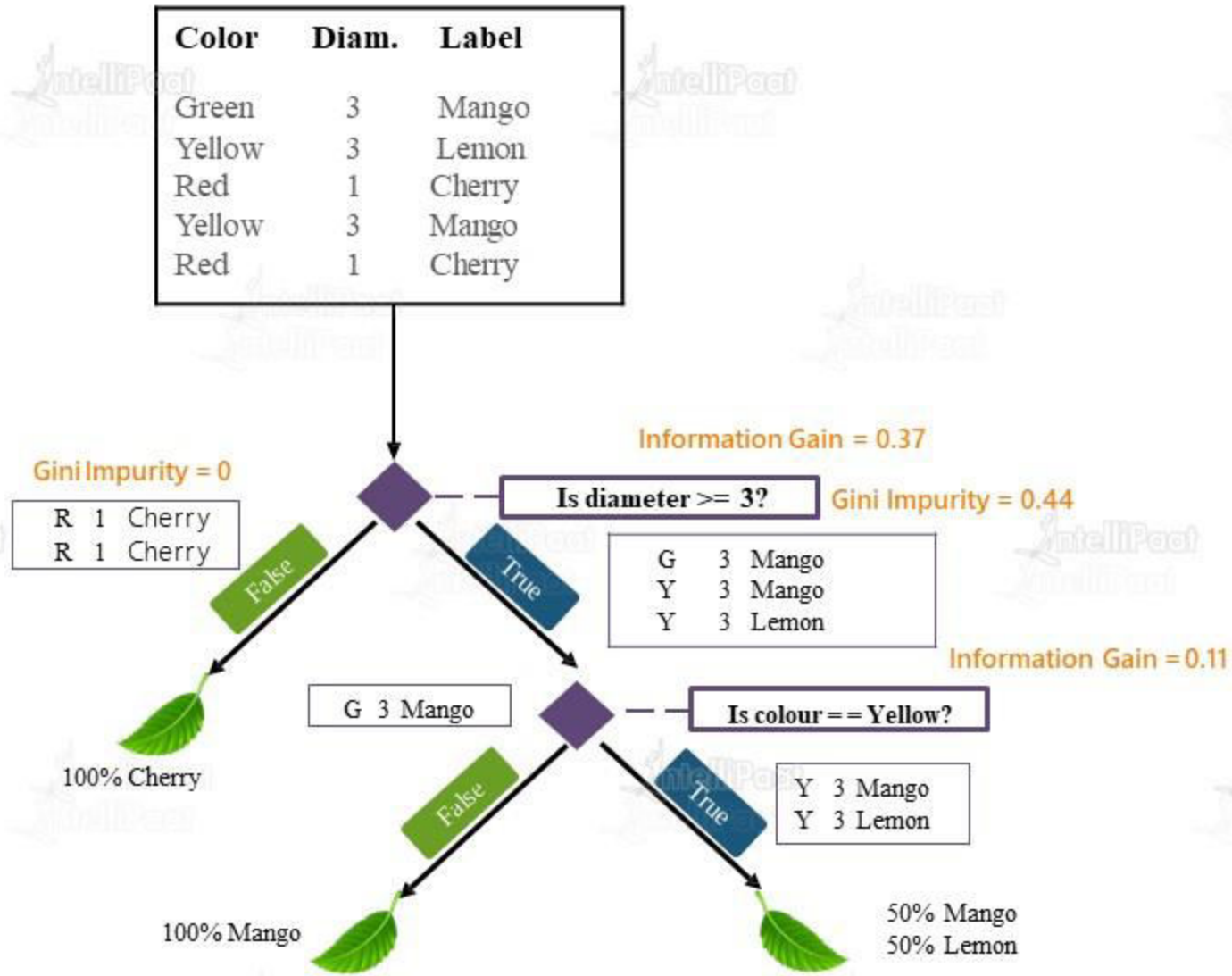
- It is a type of ensemble learning method, where a group of weak models combine to form a powerful model
- Trained with the 'bagging' method



Decision Tree

Decision Tree

Visualizing a Decision Tree



Decision Tree

Decision Tree Terminology

Pruning

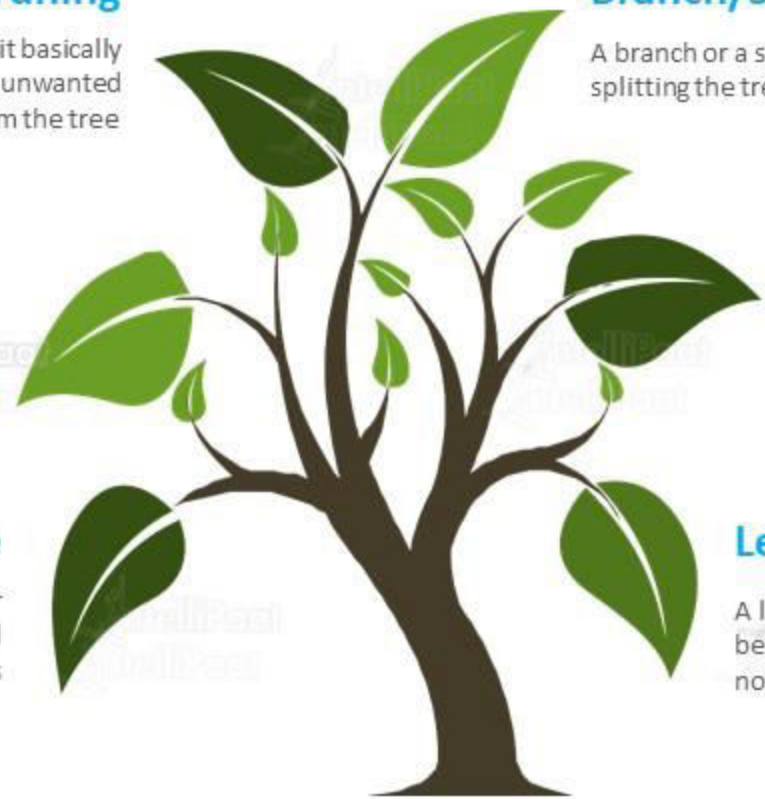
Opposite of splitting, it basically refers to removing the unwanted branches from the tree

Branch/Sub-tree

A branch or a sub-tree is formed by splitting the tree/node

Parent/Child Node

The root node is the parent node, and all other nodes branched from this node are known as child nodes



Splitting

It is the process of dividing the root node or a sub node into different parts on the basis of some condition

Root Node

It represents the entire population or the sample and further gets divided into two or more homogenous sets

Leaf Node

A leaf node is a node that cannot be further segregated into child nodes

Decision Tree

This is Our Dataset

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Decision Tree

How to decide whether we will play or not?

Which one among these should we pick first?

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Decision Tree

How do we choose the best attribute?

Or

How does a tree decide where to split?

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

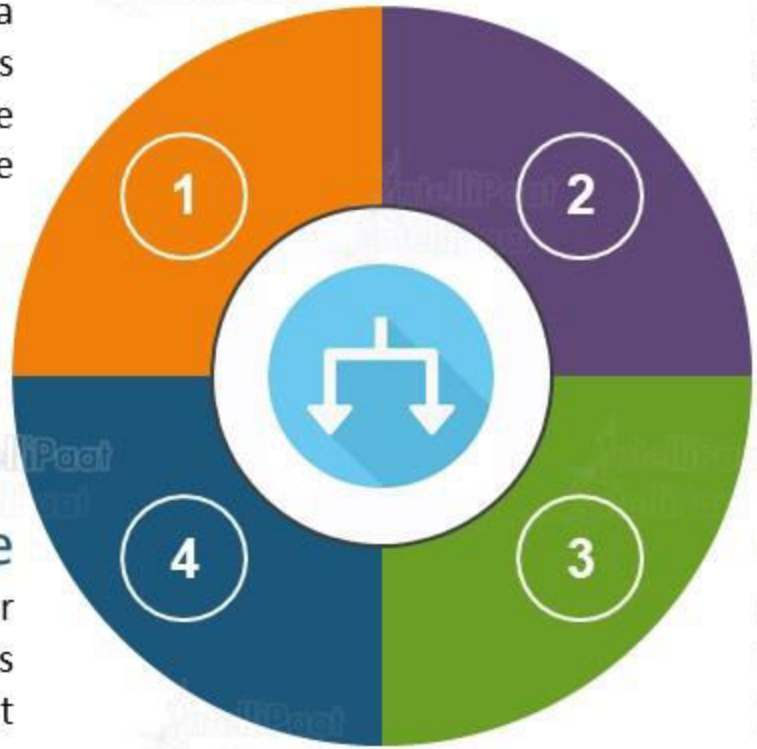
Decision Tree

Entropy

It defines the randomness in data and is a metric that measures impurity. It is the first step to solve a problem with a decision tree

Information Gain

It is the decrease in entropy after a dataset is split based on an attribute. Constructing a decision tree is for finding the attribute that returns the highest information gain



Reduction in Variance

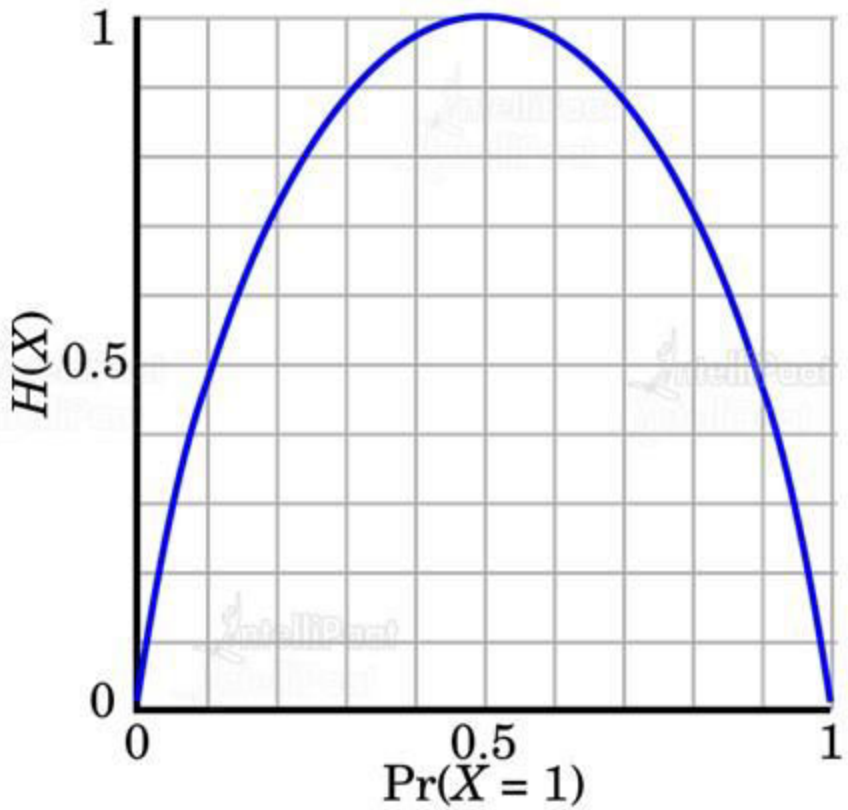
It is an algorithm used for continuous target variables (regression problems). The split with lower variance is selected as the criteria to split the population

Gini Index

Gini Index is the measure of impurity (or purity) used in building a decision tree in CART

Decision Tree

Calculating Entropy



Entropy(s) = - P(yes) $\log_2$ P(yes) – P(no) $\log_2$ P(no)

Where,

- **S** is the **total sample space**
- **P(yes)** is the **probability of yes**

If number of 'yes' = number of 'no', i.e., ***P(S) = 0.5***

⇒ ***Entropy(s) = 1***

If it contains all 'yes' or all 'no', i.e., ***P(S) = 1 or 0***

⇒ ***Entropy(s) = 0***

Decision Tree

Calculating Entropy: Example

	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no

14 instances: 9 Yes and 5 No

We have the formula,

$$E(S) = -P(\text{Yes}) \log_2 P(Yes) - P(No) \log_2 P(No)$$

$$E(S) = -(9/14)^* \log_2 9/14 - (5/14)^* \log_2 5/14$$

$$E(S) = 0.41 + 0.53 = 0.94$$

Decision Tree

Calculating Entropy: Example

	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no

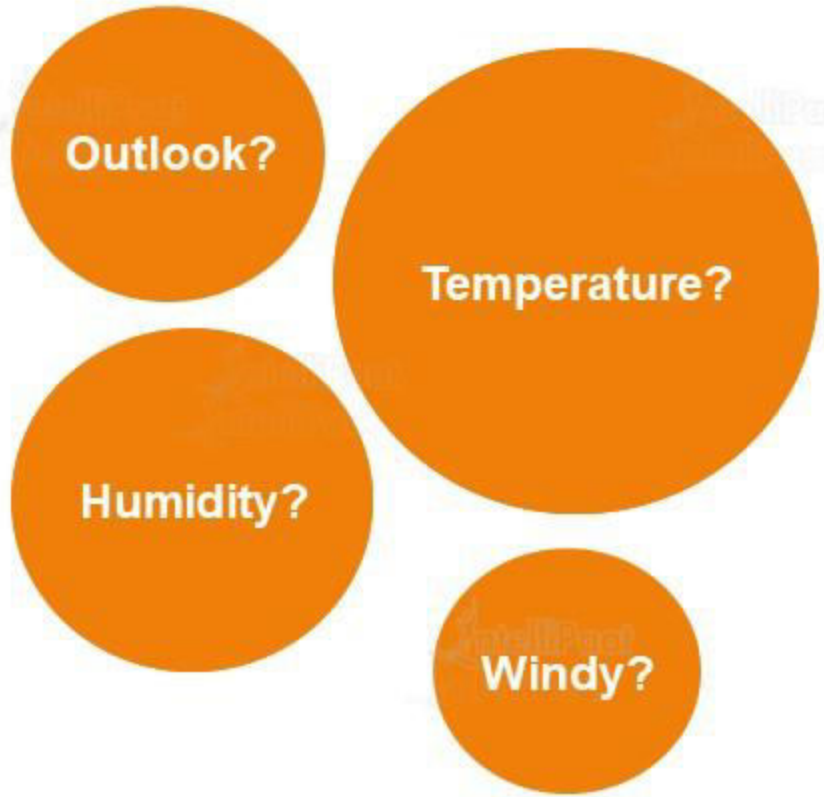
If **S** is **total collection**,

$$\text{Information Gain} = \text{Entropy}(S) - [(\text{Weighted Avg}) \text{ x Entropy}(\text{each feature})]$$

Decision Tree

Calculating Entropy: Example

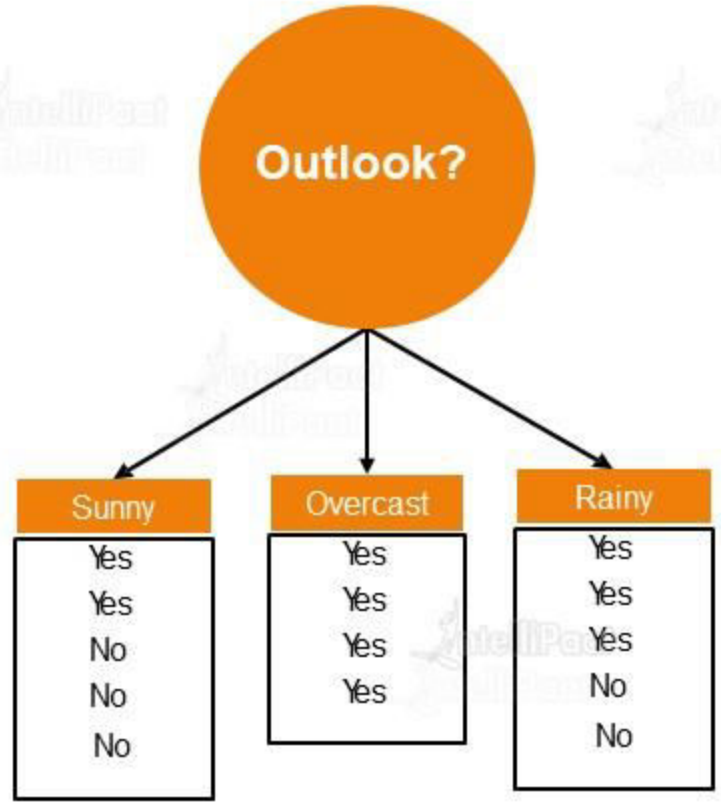
	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no



Decision Tree

Calculating Entropy: Example

	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no



Decision Tree

Calculating Entropy: Example

	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no

$E(Outlook = Sunny) = -2/5 \log_2 2/5 - 3/5 \log_2 3/5 = 0.971$

$E(Outlook = Overcast) = -1 \log_2 1 - 0 \log_2 0 = 0$

$E(Outlook = Rainy) = -3/5 \log_2 3/5 - 2/5 \log_2 2/5 = 0.971$

Information from Outlook,

$I(Outlook) = 5/14 \times 0.971 + 4/14 \times 0 + 5/14 \times 0.971 = 0.693$

Information gained from Outlook,

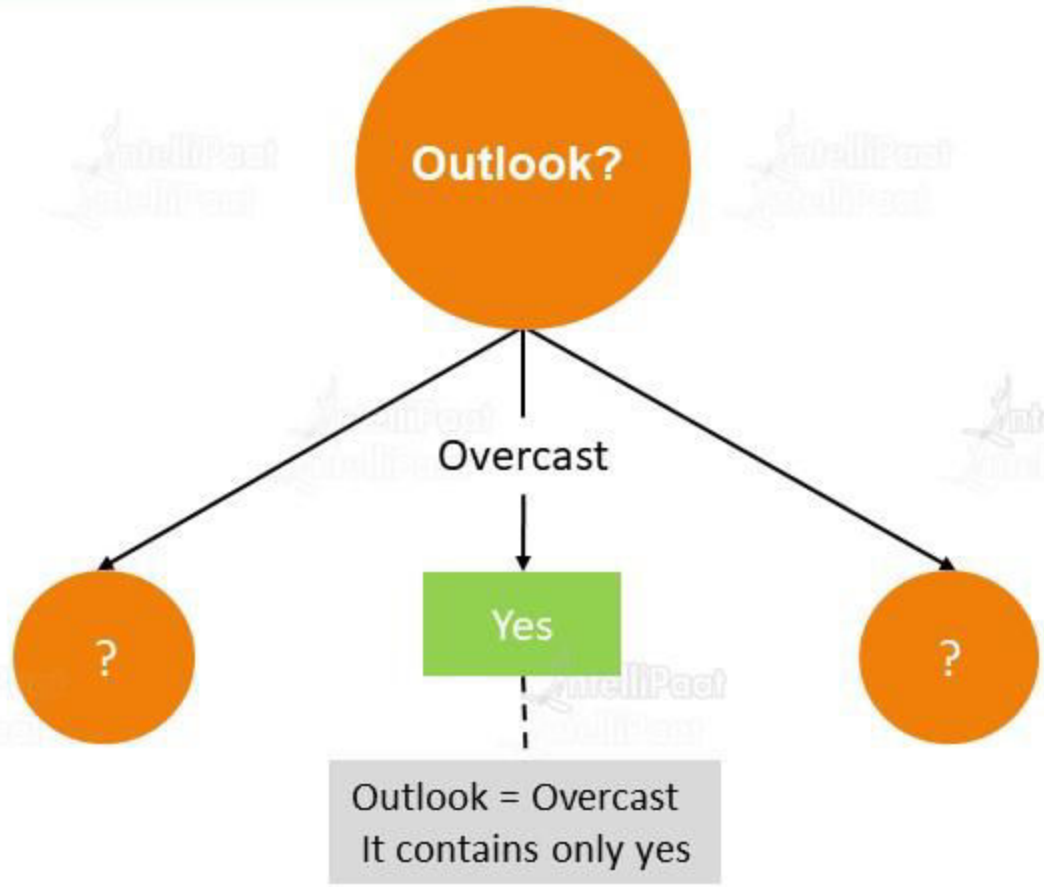
$Gain(Outlook) = E(S) - I(Outlook)$

$0.94 - 0.693 = 0.247$

Decision Tree

Calculating Entropy: Example

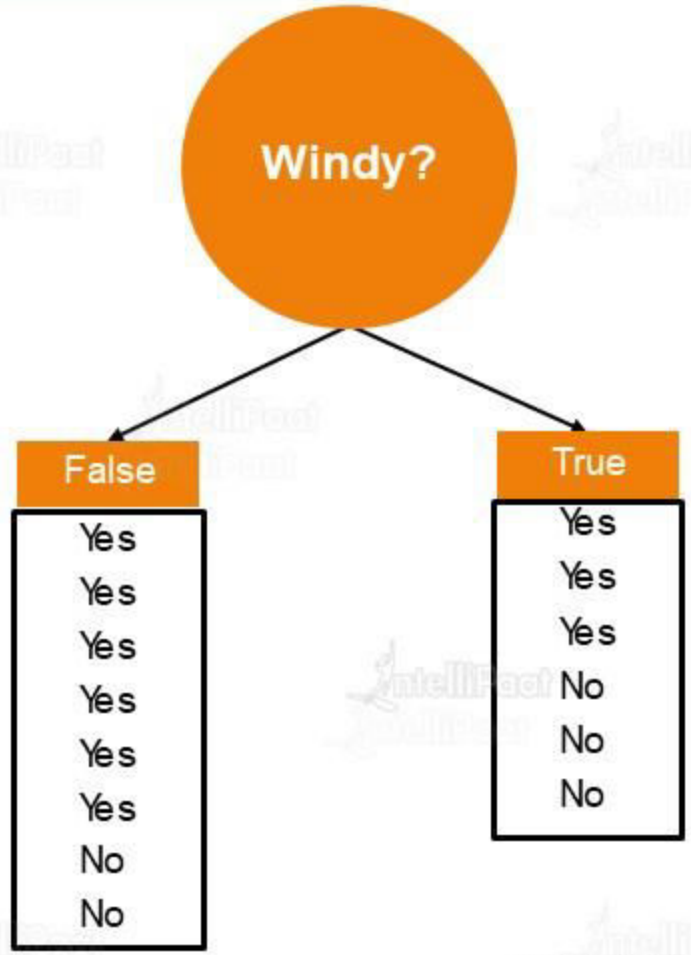
	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no



Decision Tree

Calculating Entropy: Example

	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no



Decision Tree

Calculating Entropy: Example

	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no

E(Windy = True) = 1

E(Windy = False) = 0.811

Information from Windy,

I(Windy) = 8/14 x 0.811+ 6/14 x 1 = 0.892

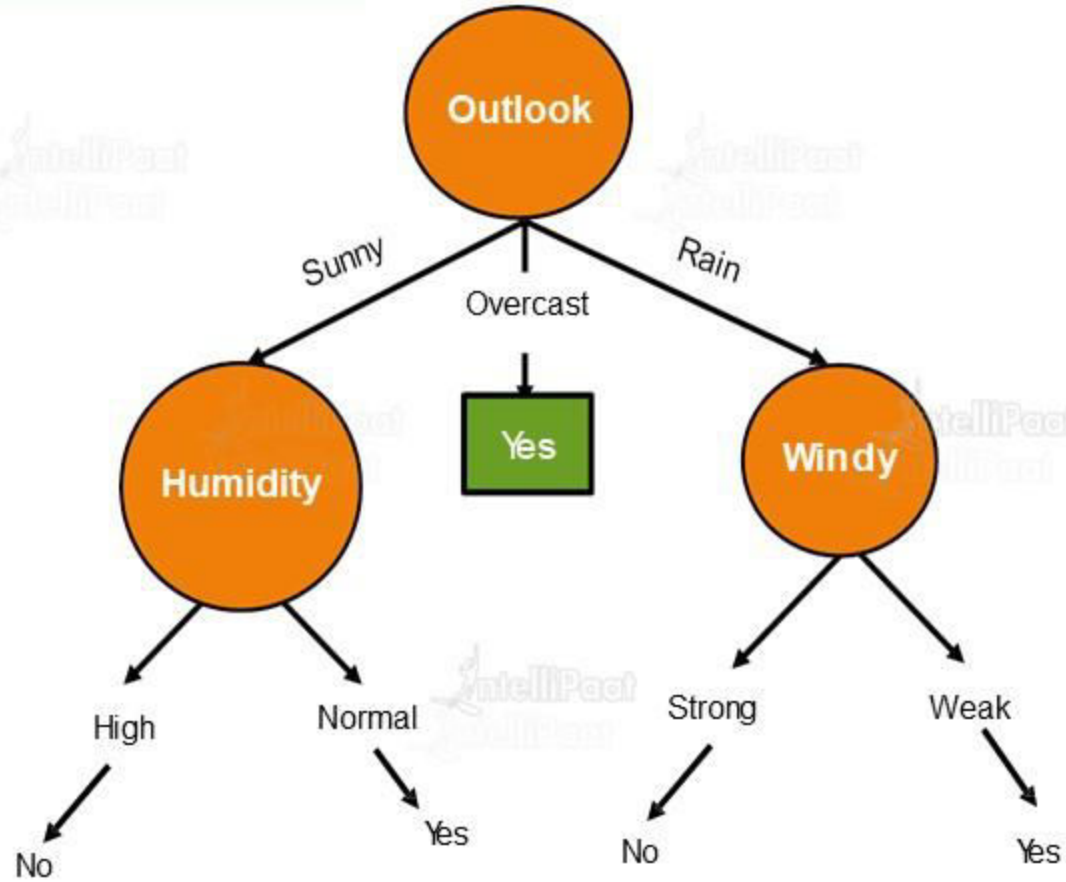
Information gained from Windy,

Gain(Windy) = E(S) – I(Windy) 0.94 – 0.892 = 0.048

Decision Tree

Calculating Entropy: Example

	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no



Hands-on: Decision Tree

Random Forest

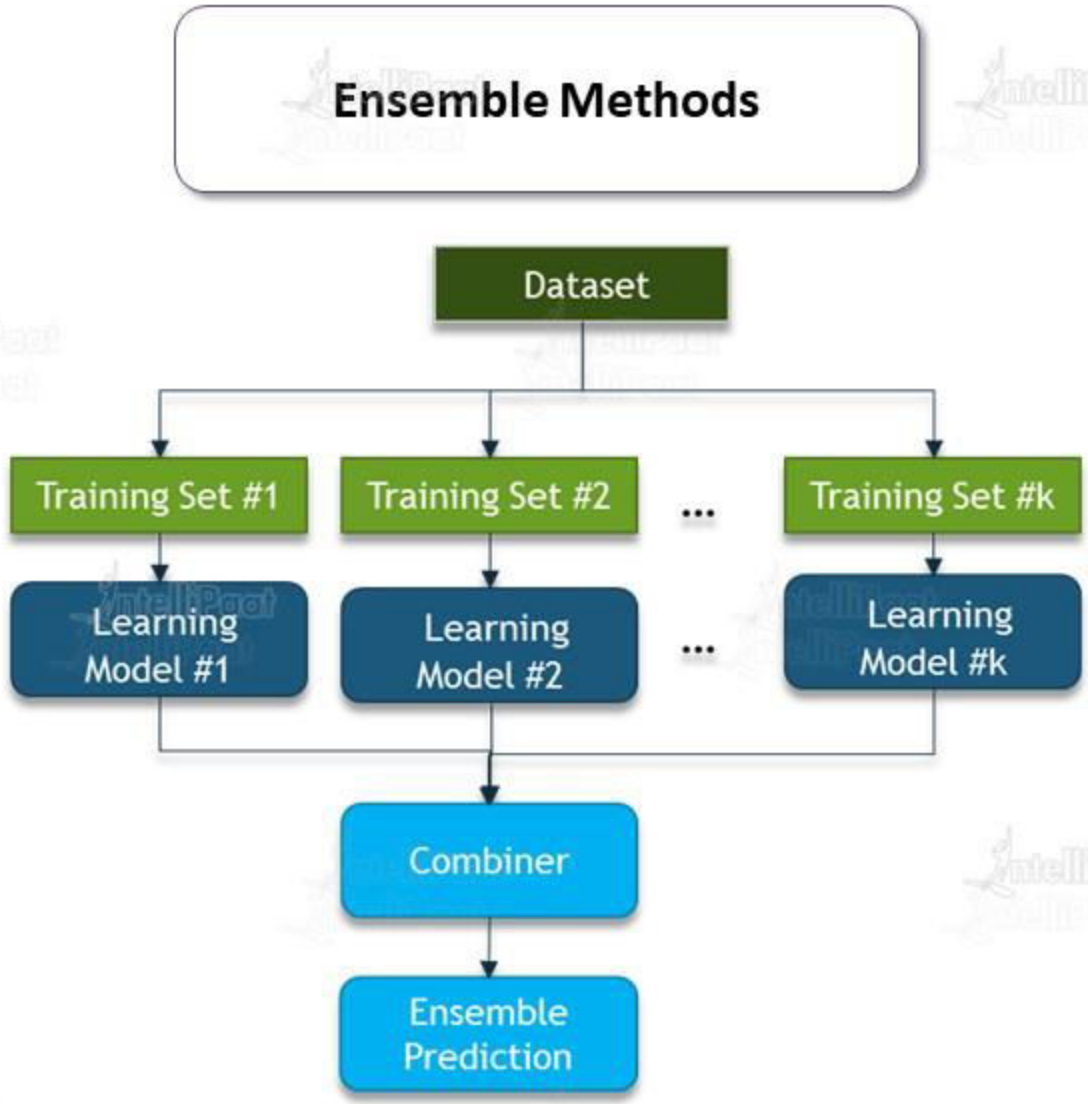
Random Forest

Did you know?

Decision trees have been around for a long time and also known to suffer from bias and variance. We will have a large bias with simple trees and a large variance with complex trees

Ensemble methods combine several decision trees to produce better predictive performance than utilizing a single decision tree

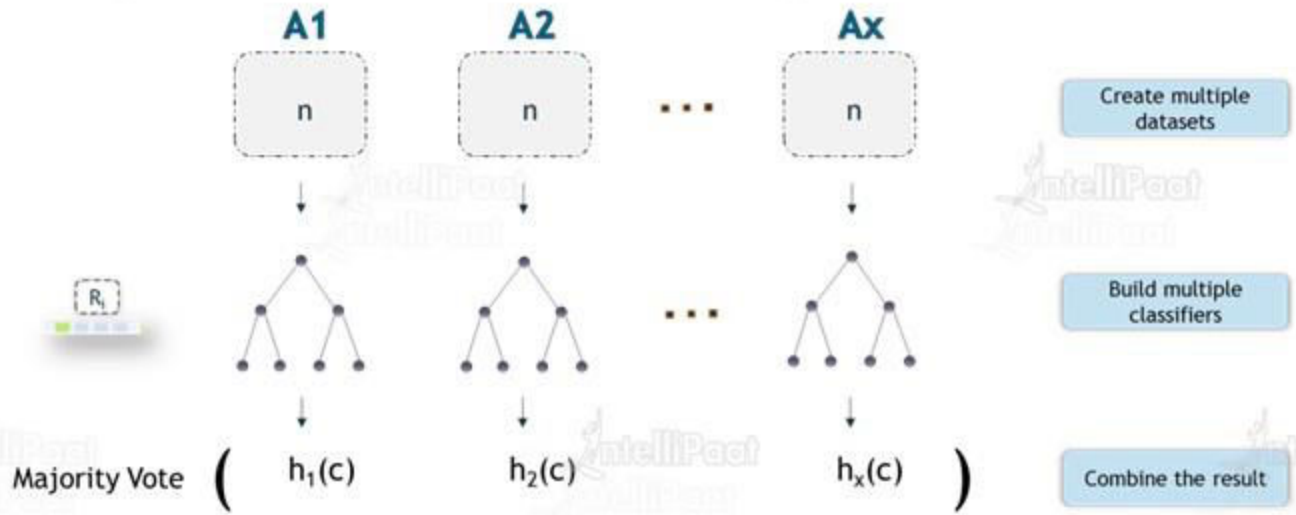
Random Forest



Random Forest

Bagging

- A technique to perform ensemble decision trees
- Used when our goal is to reduce the variance of a decision tree
- Creates several subsets of data from the training sample chosen randomly with replacement
- Each collection of the subset data is used to train its decision tree ending up with an ensemble of different models



Random Forest

Decision Tree

1. If we input a training dataset with features and labels into a decision tree, it will formulate some sets of rules, which will be used to make predictions
2. Deep decision trees might suffer from overfitting

Random Forest

1. The random forest algorithm randomly picks observations and features to build several decision trees and then averages the results
2. Mostly, a random forest prevents overfitting by creating random subsets of features and building smaller trees using these subsets

Random Forest

How does a random forest work?

01 The algorithm creates random subsets with random values from the complete dataset

02 From each subset, it creates a decision tree. Each tree is built from a sample

03 Thus, it creates multiple decision trees and then merges the results

Random Forest

How does a random forest work?

04 Sampling is done on the training dataset. Every time, a new sample is chosen to build the tree. This introduction of randomness increases the bias and reduces the variances of the model

05 This prevents the overfitting of the model, which is a serious concern in the case of decision trees

06 This also yields much better performing generalized models

Random Forest

Important Hyperparameters in Random Forest

Hyperparameters in random forest are either used to increase the predictive power of the model or to make the model faster

Increasing the Predictive Power

- ***n_estimators*** is the number of trees the algorithm builds before taking the maximum voting or the averages of predictions
- ***max_features*** is the maximum number of features the random forest considers to split a node
- ***min_sample_leaf*** determines the minimum number of leaves that are required to split an internal node

Random Forest

Important Hyperparameters in Random Forest

Hyperparameters in random forest are either used to increase the predictive power of the model or to make the model faster

Increasing the Models Speed

- ***n_jobs*** tells the engine how many processors it is allowed to use
- ***random_state*** makes the model's output replicable
- ***oob_score,*** also called ***oob sampling***, is a random forest cross validation method, in which about one-third of the data is not used to train the model and can be used to evaluate its performance

Random Forest

This is the Out Weather Dataset

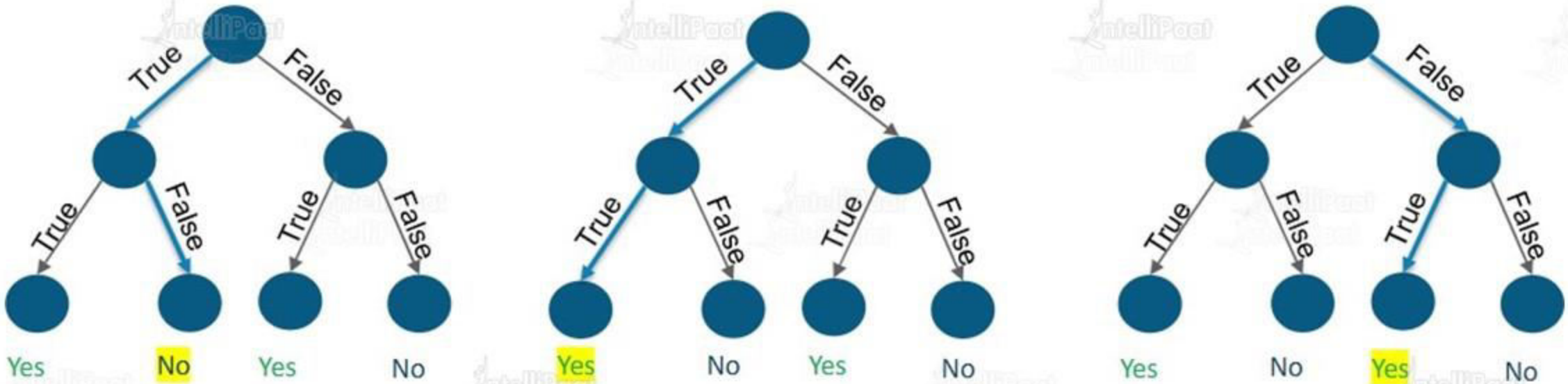
outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Random Forest

- The first step in random forest is that it will divide the data into smaller subsets
- Every subset does not need to be distinct, some may overlap
- For each subset, a decision tree is made

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Random Forest



The output of each tree will be predicted. If any two decision trees predicted that the game will happen while one predicted that it won't, then on the basis of the number of votes, the final output will be selected, in this case: ***the game will happen***

Hands-on: Random Forest

Naïve Bayes Classifier

Naïve Bayes Classifier

What is Naïve Bayes Classifier?

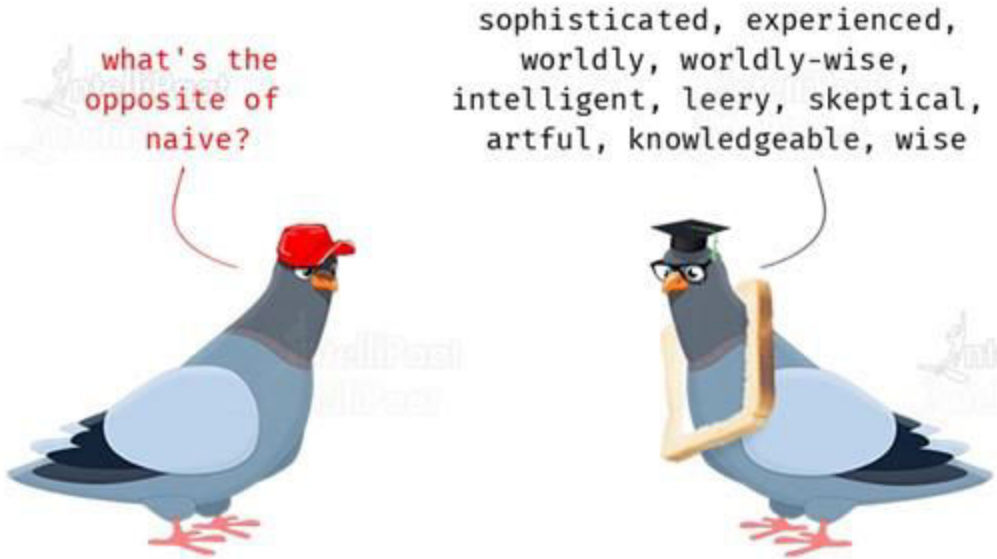
Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature

- Red
- Round
- 3 inches in diameter

Naïve Bayes Classifier

Why is Naïve Bayes so naive?

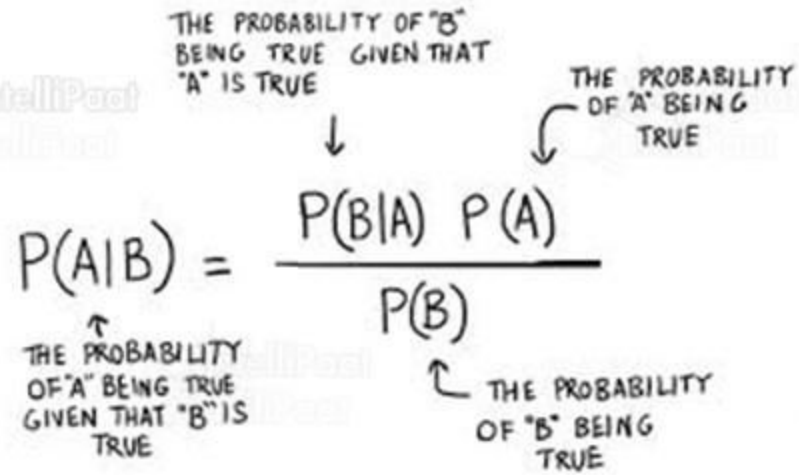
Because it makes assumptions that may or may not be correct



Naïve Bayes Classifier

Why Bayes?

Because its fundamentals are based on the Bayes theorem



Naïve Bayes Classifier

Understanding Conditional Probability

Calculating the probability of the second event (Event B) given that the first event (Event A) has already happened

For example:

- Drawing a second ace from a deck given that we got an ace in the first attempt
- Finding the probability of having a disease given that we were tested positive
- Finding the probability of liking *Game of Thrones* given that the person likes fiction

Naïve Bayes Classifier

Understanding Conditional Probability

Let's define two events:

- Event A is the probability of the event we're trying to calculate
- Event B is the condition that we know, or it is the event that has happened

Conditional Probability of P(A|B):

The probability of the occurrence of Event A given that the Event B has already happened

$$P\left(\frac{A}{B}\right) = \frac{P(A \text{ and } B)}{P(B)} = \frac{\text{Probability of the occurence of both A and B}}{\text{Probability of B}}$$

Naïve Bayes Classifier

Understanding Conditional Probability

Suppose, we have a jar containing **six** marbles: **three black** and **three red**

What is the probability of getting a black given that the first one was black?

Naïve Bayes Classifier

Understanding Conditional Probability

Suppose, we have a jar containing **six** marbles: **three black** and **three red**

What is the probability of getting a black given that the first one was black?

P (A) = Getting a black marble in the first turn
P (B) = Getting a black marble in the second turn
P (A) = 3/6
P (B) = 2/5
P (A | B) = ½*2/5 = 1/5

$$P\left(\frac{B}{A}\right) = \frac{P(\text{A and B})}{P(A)} = \frac{0.2}{0.5} = 0.4$$

Naïve Bayes Classifier

Understanding Conditional Probability

John's favorite breakfast is cereal and his favorite lunch is pizza

- The probability of John having cereal for breakfast is 0.6
- The probability having pizza for lunch is 0.5
- The probability of him having a cereal for breakfast given that he eats a pizza for lunch is 0.7

Naïve Bayes Classifier

Understanding Conditional Probability

Now, what if we want to know the probability of John having a pizza given that he had a bowl of cereal for breakfast?

Here, we have to calculate: $P\left(\frac{B}{A}\right)$

Now this is where the **Bayes theorem** comes into the picture

Naïve Bayes Classifier

Understanding Bayes Theorem

The Bayes theorem describes the probability of an event based on the prior knowledge of the conditions that might be related to the event

It shows the relation between a conditional probability and its reverse form

- If the conditional probability = $P\left(\frac{A}{B}\right)$
- We use the Bayes rule to find the reverse probability: $P\left(\frac{B}{A}\right)$

Naïve Bayes Classifier

Bayes Theorem: Use Case

Find out a patient's probability of having liver disease if he/she is an alcoholic

Naïve Bayes Classifier

Understanding the Bayes Theorem

Event A: Patient has liver disease
Event B: Whether the patient is an alcoholic

1. Past data tells us that 10% of patients entering our clinic have liver disease: P(A) = 0.10

2. 5% of the clinic's patients are alcoholics: P(B) = 0.05

3. We might also know that among those patients diagnosed with liver disease, 7% are alcoholics. This is our B|A, i.e., the probability that a patient is alcoholic given that he/she has liver disease is 7%

Naïve Bayes Classifier

Understanding the Bayes Theorem

According to the Bayes theorem:

1. **P(A|B) = (0.07 * 0.1)/0.05 = 0.14**

2. In other words, if a patient is an alcoholic, then his/her chances of having liver disease is 0.14 (14%). This is a large increase from the 10% suggested by the past data. However, it is still unlikely that any particular patient has liver disease

Naïve Bayes Classifier

	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no

Should I play today?

Total Sample: 14

Total Yes: 9

Total No: 5

P(Yes): 9/14

P(No): 5/14

Naïve Bayes Classifier

	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no

Frequency table of each attribute:

Outlook	Yes	No
Sunny	2	3
Overcast	4	0
Rainy	3	2
Total	9	5

Temperature	Yes	No
Hot	2	2
Mild	4	2
Cool	3	1
Total	9	5

Humidity	Yes	No
High	3	4
Normal	6	1
Total	9	5

Windy	Yes	No
False	6	2
True	3	3
Total	9	5

Naïve Bayes Classifier

	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no

Probability of each attribute: P(A)

Outlook	Yes	No	P(Attrib)
Sunny	2	3	5/14
Overcast	4	0	4/14
Rainy	3	2	5/14
Total	9	5	100%

Temperature	Yes	No	P(Attrib)
Hot	2	2	4/14
Mild	4	2	6/14
Cool	3	1	4/14
Total	9	5	100%

Humidity	Yes	No	P(Attrib)
High	3	4	7/14
Normal	6	1	7/14
Total	9	5	100%

Windy	Yes	No	P(Attrib)
False	6	2	8/14
True	3	3	6/14
Total	9	5	100%

Naïve Bayes Classifier

	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no

Probability of each attribute: P(B)

Play		Probability
Yes	9	9/14
No	5	5/14
Total	14	100%

Naïve Bayes Classifier

	outlook	temp.	humidity	windy	play
D1	sunny	hot	high	false	no
D2	sunny	hot	high	true	no
D3	overcast	hot	high	false	yes
D4	rainy	mild	high	false	yes
D5	rainy	cool	normal	false	yes
D6	rainy	cool	normal	true	no
D7	overcast	cool	normal	true	yes
D8	sunny	mild	high	false	no
D9	sunny	cool	normal	false	yes
D10	rainy	mild	normal	false	yes
D11	sunny	mild	normal	true	yes
D12	overcast	mild	high	true	yes
D13	overcast	hot	normal	false	yes
D14	rainy	mild	high	true	no

Likelihood of each attribute: P(A|B)

Outlook	Yes	No	P(Yes)	P(No)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%

Temperature	Yes	No	P(Yes)	P(No)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity	Yes	No	P(Yes)	P(No)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Windy	Yes	No	P(Yes)	P(No)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

Play		Probability
Yes	9	9/14
No	5	5/14
Total	14	100%

Naïve Bayes Classifier

Now, we have to calculate P(B | A)

$$P(A|B) = \frac{P(A \bigcap B)}{P(B)} = \frac{P(A) \cdot P(B|A)}{P(B)}$$

Outlook			
	Yes	No	P(Attrib)
Sunny	2	3	5/14
Overcast	4	0	4/14
Rainy	3	2	5/14
Total	9	5	100%

Temperature			
	Yes	No	P(Attrib)
Hot	2	2	4/14
Mild	4	2	6/14
Cool	3	1	4/14
Total	9	5	100%

Humidity			
	Yes	No	P(Attrib)
High	3	4	7/14
Normal	6	1	7/14
Total	9	5	100%

Windy			
	Yes	No	P(Attrib)
False	6	2	8/14
True	3	3	6/14
Total	9	5	100%

Play		Probability
Yes	9	9/14
No	5	5/14
Total	14	100%

Ideal condition:

P(Outlook = Sunny) = 5/14

P(Temperature = Cool) = 4/14

P(Humidity = Normal) = 7/14

P(Wind = False) = 8/14

Probability of ideal condition:

P(X) =(5/14)*(4/14)*(7/14)*(8/14)

P(X) = 0.029

Naïve Bayes Classifier

Outlook				
	Yes	No	P(Yes)	P(No)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%

Temperature				
	Yes	No	P(Yes)	P(No)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity				
	Yes	No	P(Yes)	P(No)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Windy				
	Yes	No	P(Yes)	P(No)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

Play		Probability
Yes	9	9/14
No	5	5/14
Total	14	100%

Probability of playing the game in the ideal condition:

$$P(Yes|X) = \frac{P(X|Yes) \times P(Yes)}{P(X)}$$

$$P(Yes|X) = \frac{0.033 \times (9/14)}{0.029}$$

$$P(Yes|X) = 0.73$$

Hands-on: Naïve Bayes Classifier

Hands-on: Email Classification Using Naïve Bayes

Hands-on: Email Classification

Why Naïve Bayes for email spam classification?

1. Naïve Bayes assumes all features to be conditionally independent
2. If we are trying to classify a dataset with independent variables, then Naïve Bayes is the best choice

Steps for Email Classification

1. Import the dataset
2. Analyze the dataset
3. Remove unnecessary columns
4. Create a CountVectorizer
5. Train the CountVectorizer on the data
6. Create a Naive Bayes Classifier
7. Train it on the vectorized data
8. Train the model
9. Check its accuracy

India: +91-7847955955

US: 1-800-216-8930 (TOLL FREE)

support@intellipaat.com

24/7 Chat with Our Course Advisor