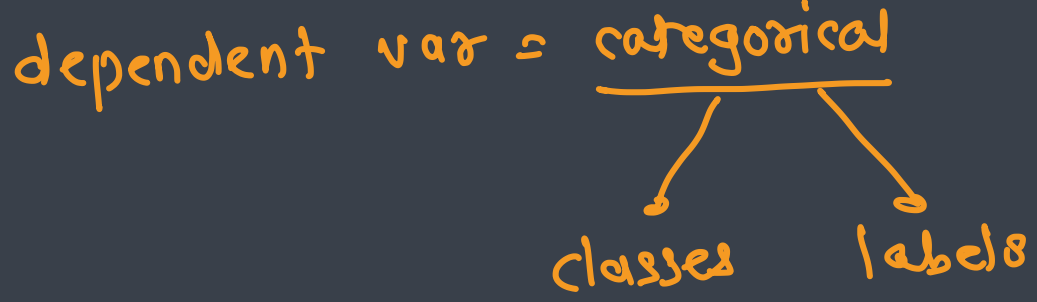
Classification



Introduction

label = class = categorical

- Classification is a supervised machine learning method where the model tries to predict the correct label of a given input data
- In classification, the model is fully trained using the training data, and then it is evaluated on test data before being used to perform prediction on new unseen data
- For instance, an algorithm can learn to predict whether a given email is spam or ham (no spam)

Learners → learns

training data → program → model

- Eager learners → model based
  - These are machine learning algorithms that first build a model from the training dataset before making any prediction on future datasets
  - They spend more time during the training process because of their eagerness to have a better generalization during the training from learning the weights, but they require less time to make predictions.
  - Most machine learning algorithms are eager learners, and below are some examples:
    - Logistic Regression
    - Support Vector Machine
    - Decision Trees
    - Artificial Neural Networks

weight = estimator = coefficient

$y = mx + c$

slope = coe

- Lazy learners or instance-based learners
  - They do not create any model immediately from the training data, and this is where the lazy aspect comes from
  - They just memorize the training data, and each time there is a need to make a prediction, they search for the nearest neighbor from the whole training data, which makes them very slow during prediction
  - Some examples:
    - K-Nearest Neighbor
    - Case-based reasoning

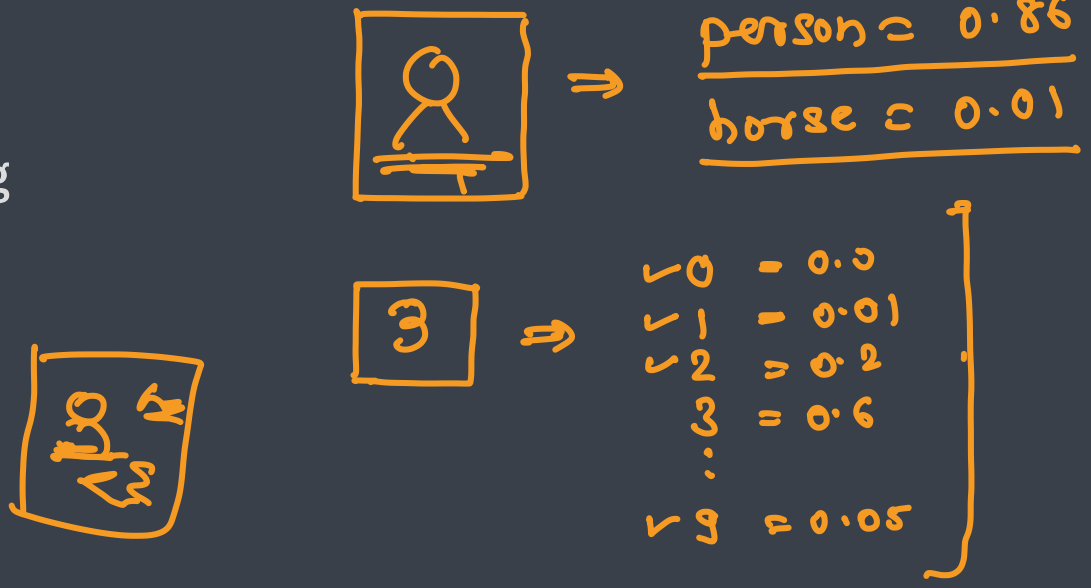
Types of Classification

- **Binary Classification**
  - The goal is to classify the input data into two mutually exclusive categories
  - The training data in such a situation is labeled in a binary format: true and false; positive and negative; 0 and 1; spam and not spam, etc. depending on the problem being tackled
  - For instance, we might want to detect whether a given image is a truck or a boat.
  - Logistic Regression and Support Vector Machines algorithms are natively designed for binary classifications
  - However, other algorithms such as K-Nearest Neighbors and Decision Trees can also be used for binary classification

- **Multiclass classification**
  - The multi-class classification, on the other hand, has at least two mutually exclusive class labels, where the goal is to predict to which class a given input example belongs to
  - For instance, classifying customer into life styles: lower class, middle class and upper class
  - Most of the binary classification algorithms can be also used for multi-class classification

Types of Classification

- **Multi-Label Classification**
  - In multi-label classification tasks, we try to predict 0 or more classes for each input example
  - In this case, there is no mutual exclusion because the input example can have more than one label
  - Such a scenario can be observed in different domains, such as auto-tagging in Natural Language Processing, where a given text can contain multiple topics
  - Similarly to computer vision, an image can contain multiple objects
  - It is not possible to use multi-class or binary classification models to perform multi-label classification
  - However, most algorithms used for those standard classification tasks have their specialized versions for multi-label classification
  - We can cite:
    - Multi-label Decision Trees
    - Multi-label Gradient Boosting
    - Multi-label Random Forests



Logistic Regression

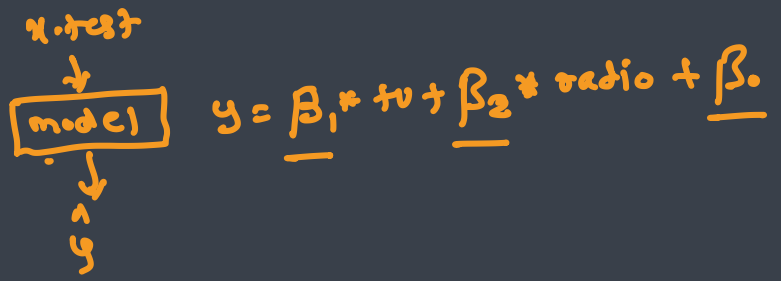
Linear Relationship

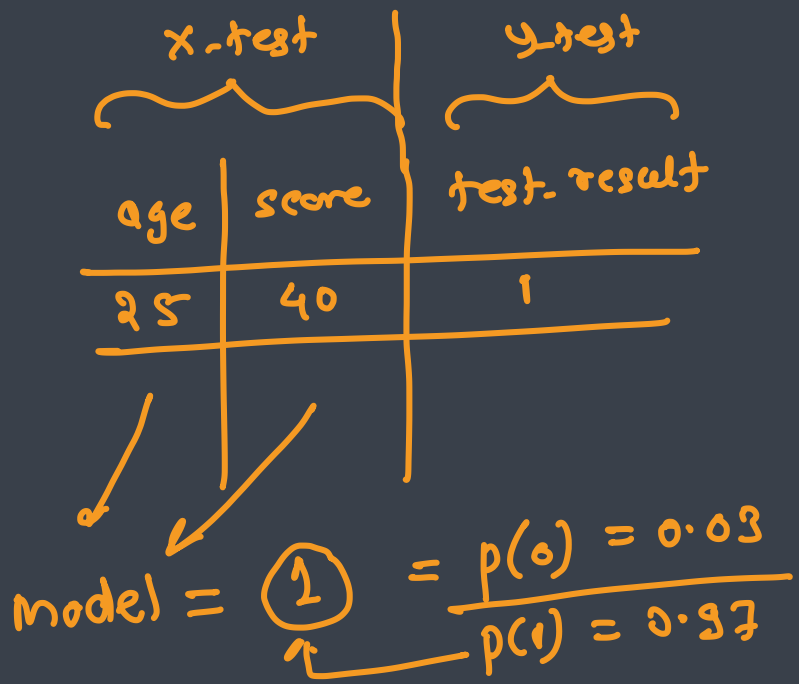
Overview

- Logistic Regression was used in the biological sciences in early twentieth century
- It was then used in many social science applications
- Logistic Regression is used when the dependent variable(target) is categorical
- The dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.)
- Unlike linear regression, logistic regression can directly predict probabilities (values that are restricted to the (0,1) interval)
- Furthermore, those probabilities are well-calibrated when compared to the probabilities predicted by some other classifiers
- E.g.
  - To predict whether an email is spam (1) or (0)
  - Whether the tumor is malignant (1) or not (0)

Assumptions

- Binary logistic regression requires the dependent variable to be binary
- For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome
- Only the meaningful variables should be included → feature Engineering
- The independent variables should be independent of each other. That is, the model should have little or no multi-collinearity
- The independent variables are linearly related to the log odds
- Logistic regression requires quite large sample sizes

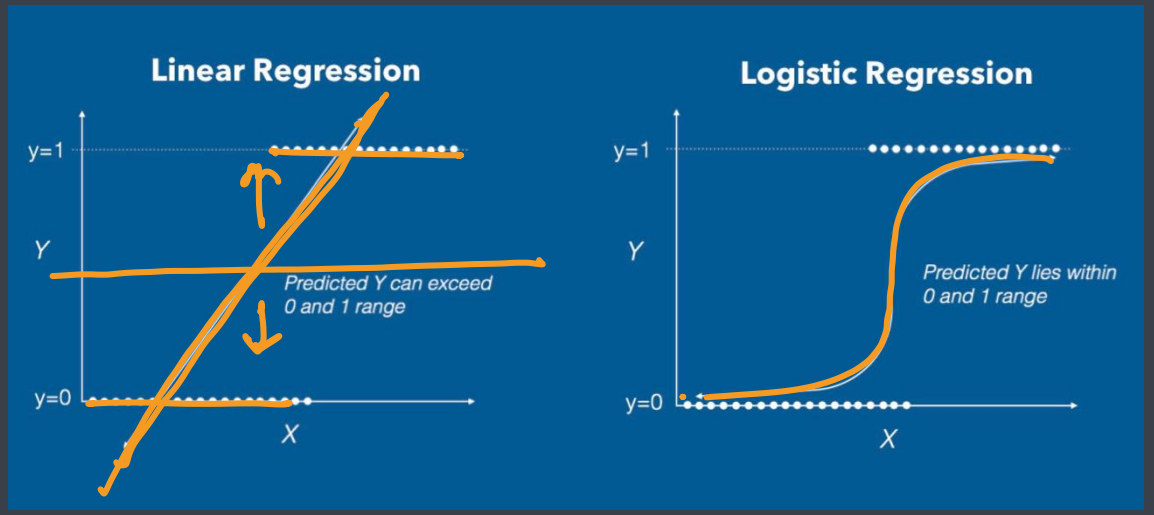




Linear vs Logistic

- Target variable is an interval variable
- Predicted values are the mean of the target variable at the given values of the input variable

- Target variable is a discrete (binary or ordinal) variable
- Predicted values are the probability of a particular level(s) of the target variable at the given values of the input variables



How does it work ?

- Logistics regression analysis starts with calculating the "Odds" of the dependent variable
- It is the ratio of the probability that an individual is a member of a particular group or category, p(y) divided by the probability that an individual is not a member of the group or category [1– p(y)]
- It is represented as:

$$Odds = \frac{p(y)}{[1 - p(y)]}$$

- In order to establish a linear relationship between the odds and the independent variables in the logistic regression model, the odds need to be transformed to logit (log-odds) by taking the natural logarithm (ln) of odds
- The logarithmic transformation creates a continuous dependent variable of the categorical dependent variable

Types

- **Binary Logistic Regression**
  - Most useful when you want to model the event probability for a categorical response variable with two outcomes
  - For example, its often used in credit analysis in determining the risk whether the next customer is likely to default — or not default — on a loan

multi-class

- **Multinomial Logistic Regression**
  - Used to classify subjects into groups based on a categorical range of variables to predict behaviour
  - For example, a survey can be conducted to aid advertising strategy where participants are asked to select one of several competing products as their favorite

Advantages

- Widely used technique due to its simplicity, efficiency, easy interpretation, and usage of limited computational resources.
- Allows easy regularization of outputs to prevent overfitting, yielding probabilities as prediction results.
- Logistic Regression allows easy model updating using stochastic gradient descent.
- Logistic Regression models does not get effected to predict output probabilities on removal of variables uncorrelated to the output or multi-collinear variables

Disadvantages

- Logistic regression’s greatest disadvantage is fails to solve non-linear problems and it underperforms when there are multiple or non-linear decision boundaries. It fails to capture more complex relationships.
- Another important requirement for Logistic Regression to function properly is Feature Engineering as it helps to identify independent variables. Without proper identification of independent variables Logistic Regression fails to perform correctly.
- Logistic Regression can only predict a categorical outcome with discrete probability outcome

Applications of Logistic Regression

- Logistic regression has been widely used in medical research, in the field of predictive food microbiology, to describe bacterial growth/no growth interface, in the 0–1 format.
- Logistic regression may be used to predict the risk of developing a given disease based on observed characteristics of the patient.
- Logistic regression is widely used in credit risk modelling in identifying potential loan defaulters.
- Logistic regression is used in different predictive models which finds its usage in response modelling problems like Normal, Poisson, binomial responses and other distributions, even including hypothesis testing.

Naïve Bayes

Overview

- Naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features
- They are among the simplest Bayesian network models
- Naïve Bayes has been studied extensively since the 1960s
- Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem

Bayes Theorem

- Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred
- Bayes' theorem is stated mathematically as the following equation:

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

  - where A and B are events and P(B) ? 0.
  - Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as **evidence**.
  - P(A) is the **priori** of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
  - P(A | B) is a posteriori probability of B, i.e. probability of event after evidence is seen.

Bayes Theorem

- Now, with regards to our dataset, we can apply Bayes' theorem in following way:

$$P(y|X) = \frac{P(y) * P(X|y)}{P(X)}$$

How does it work ?

- Below is a training data set of weather and corresponding target variable ‘Play’ (suggesting possibilities of playing). We need to classify whether players will play or not based on weather condition.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

How does it work ?

- Step 1: Convert the data set into a frequency table

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

How does it work ?

- Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

- Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

How does it work ?

- **Problem:** Players will play if weather is sunny. Is this statement is correct?
- We can solve it using above discussed method of posterior probability.

$$P(Yes \mid Sunny) = P(Sunny \mid Yes) * P(Yes) / P(Sunny)$$

- Here we have

P (Sunny | Yes) = 3/9 = 0.33
P(Sunny) = 5/14 = 0.36
P(Yes)= 9/14 = 0.64

- Which means, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

Types of Naïve Bayes

- Gaussian Naïve Bayes classifier
- Multinomial Naive Bayes
- Bernoulli Naive Bayes

K-Nearest Neighbors

Overview

- The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems
- However, it is more widely used in classification problems in the industry
- It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection
- The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

How does it work ?

- A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function
- If K = 1, then the case is simply assigned to the class of its nearest neighbor.
- Note: all three distance measures are only valid for continuous variables.

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

How does it work?

- In the instance of categorical variables the Hamming distance must be used

Hamming Distance

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

X	Y	Distance
Male	Male	0
Male	Female	1

How does it work ?

- Choosing the optimal value for K is best done by first inspecting the data
- In general, a large K value is more precise as it reduces the overall noise but there is no guarantee
- Cross-validation is another way to retrospectively determine a good K value by using an independent dataset to validate the K value
- Historically, the optimal K for most datasets has been between 3-10. That produces much better results than 1NN.

Advantages

- No assumptions about data — useful, for example, for nonlinear data
- Simple algorithm — to explain and understand/interpret
- High accuracy (relatively) — it is pretty high but not competitive in comparison to better supervised learning models
- Versatile — useful for classification or regression

Disadvantages

- Computationally expensive — because the algorithm stores all of the training data
- High memory requirement
- Stores all (or almost all) of the training data
- Prediction stage might be slow (with big N)
- Sensitive to irrelevant features and the scale of the data

Applications of KNN

- Recommender system
- Relevant document classification
- OCR

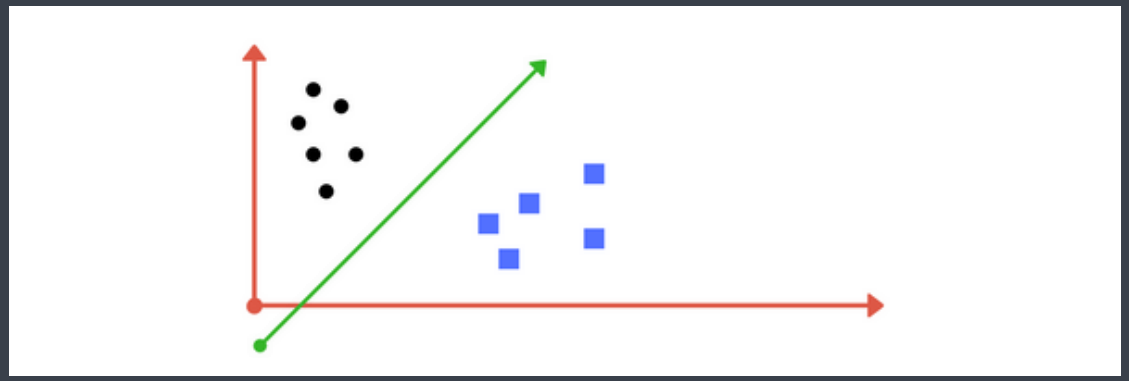
Support Vector Machine

Overview

- It is a supervised machine learning algorithm that can be used for both classification and regression
- However, it is mostly used in classification problems
- The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points

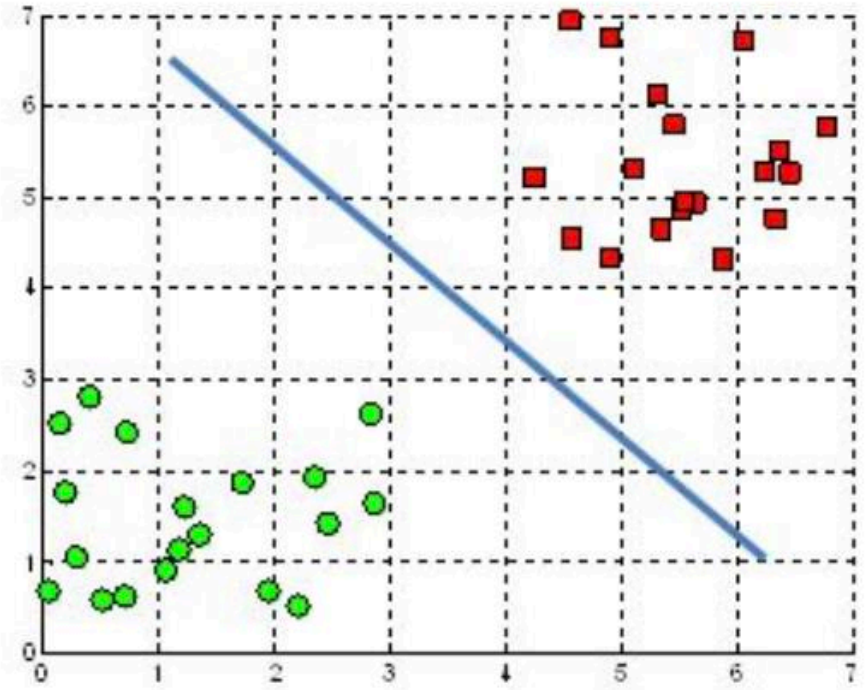
How does it work ?

- SVM separates the classes using hyperplane
- In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side
- To separate the two classes of data points, there are many possible hyperplanes that could be chosen
- Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes

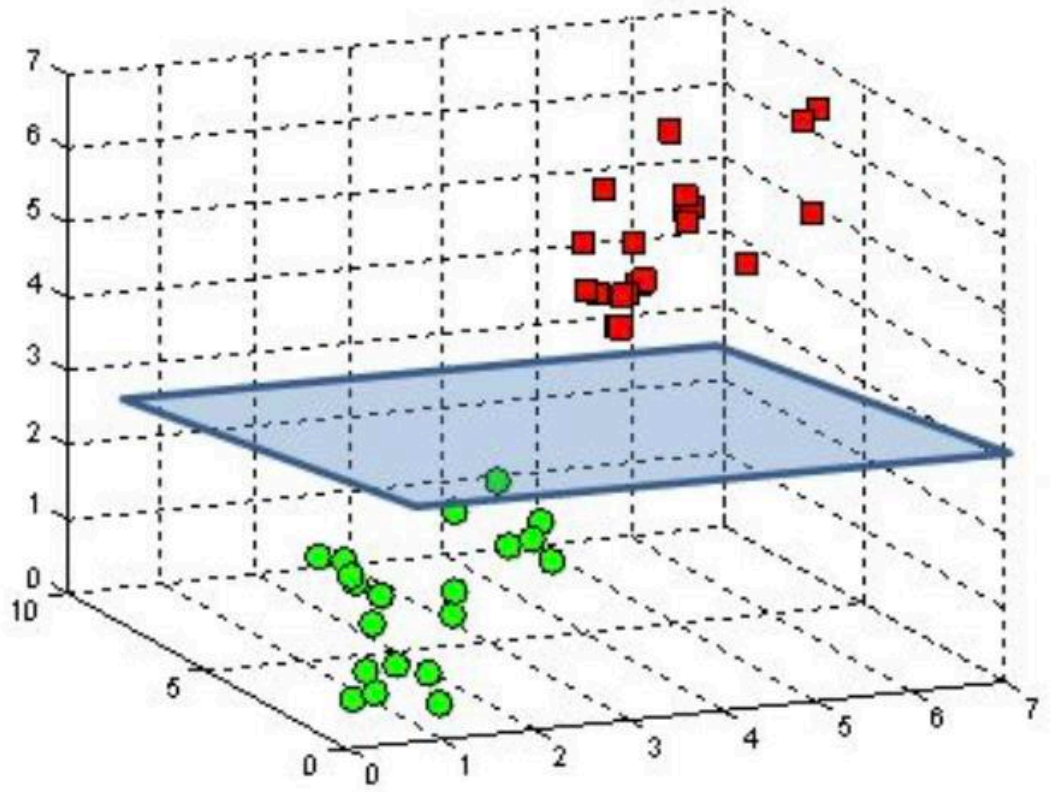


Hyperplane

A hyperplane in $\mathbb{R}^2$ is a line

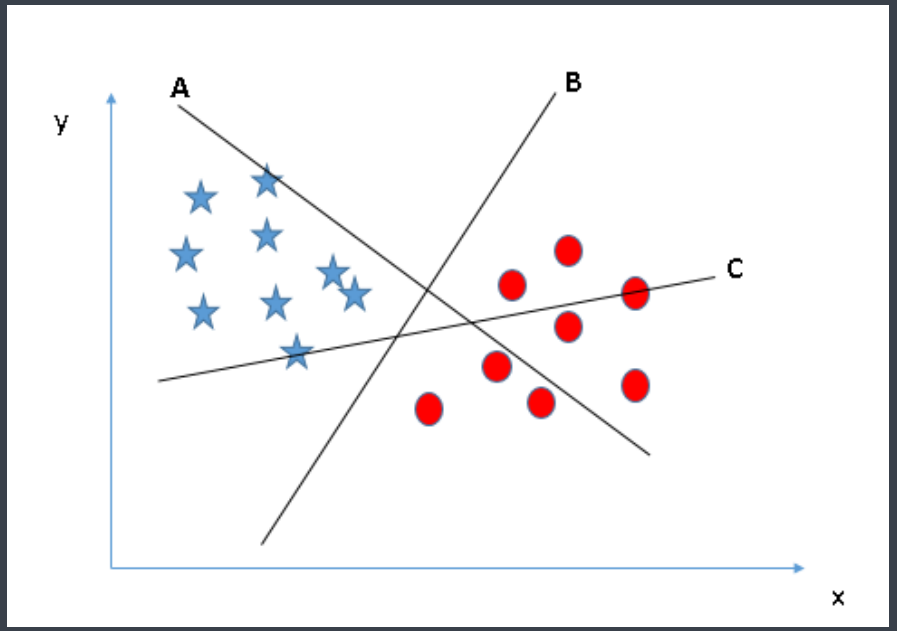


A hyperplane in $\mathbb{R}^3$ is a plane



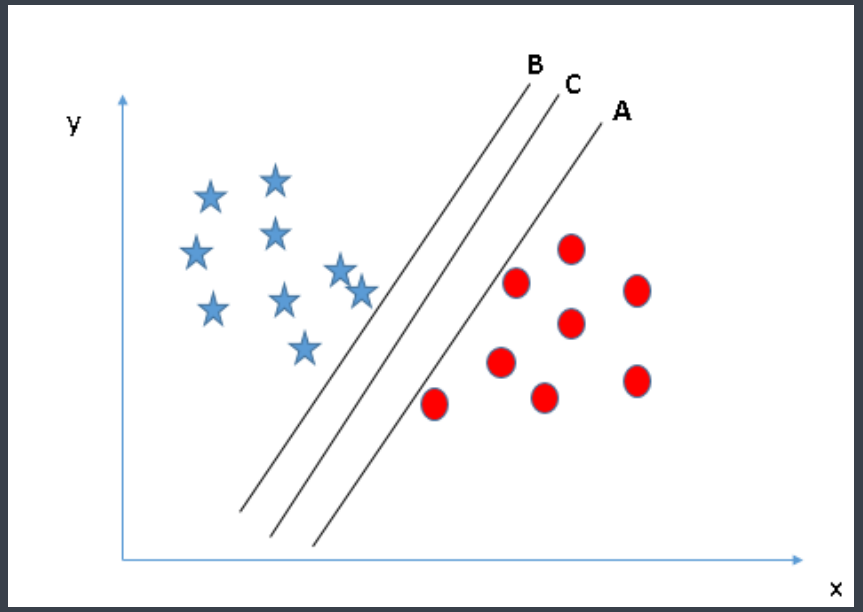
Scenario 1

- Here, we have three hyper-planes (A, B and C)
- Now, identify the right hyper-plane to classify star and circle
- You need to remember a thumb rule to identify the right hyper-plane
  - Select the hyper-plane which segregates the two classes better
- In this scenario, hyper-plane “B” has excellently performed this job



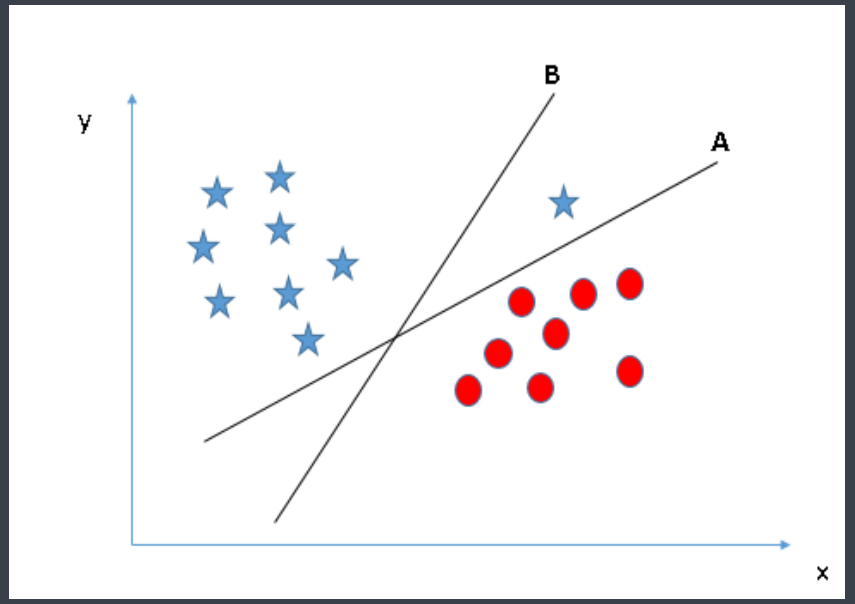
Scenario 2

- Here, we have three hyper-planes (A, B and C) and all are segregating the classes well
- Now, How can we identify the right hyper-plane?
- Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin.**



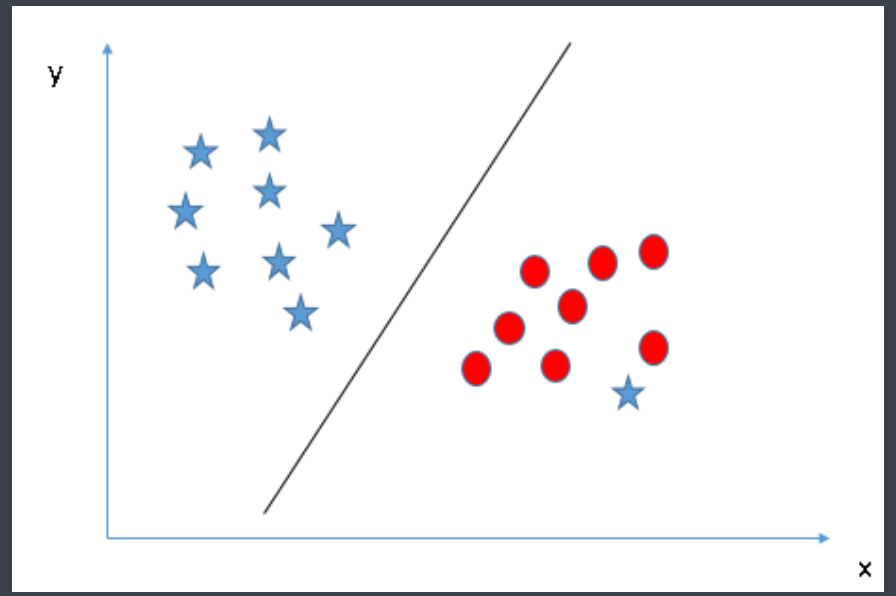
Scenario 3

- Use the rules as discussed in previous section to identify the right hyper-plane
- Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A.**
- But, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin
- Here, hyper-plane B has a classification error and A has classified all correctly
- Therefore, the right hyper-plane is **A.**



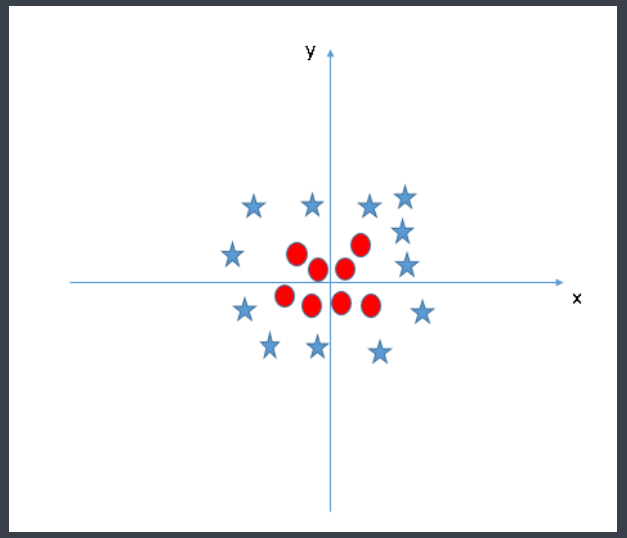
Scenario 4

- Unable to segregate the two classes using a straight line, as one of star lies in the territory of other(circle) class as an outlier
- SVM has a feature to ignore outliers and find the hyper-plane that has maximum margin
- Hence, we can say, SVM is robust to outliers.



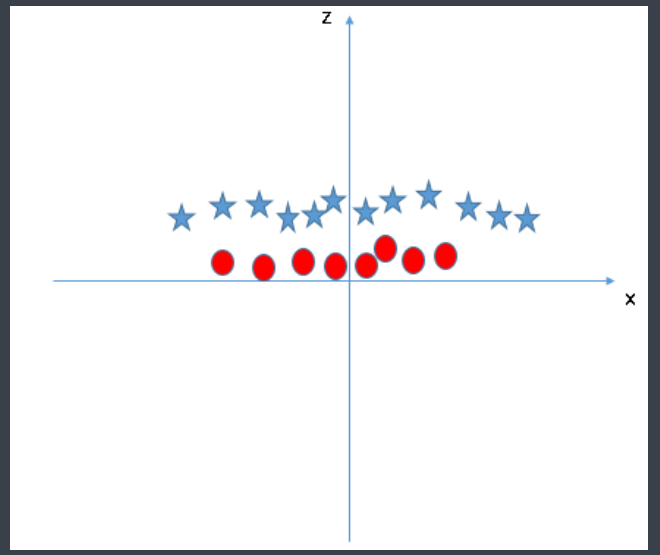
Scenario 5

- In the scenario below, we can't have linear hyper-plane between the two classes



- SVM solves this problem by introducing

$$z=x^2+y^2$$

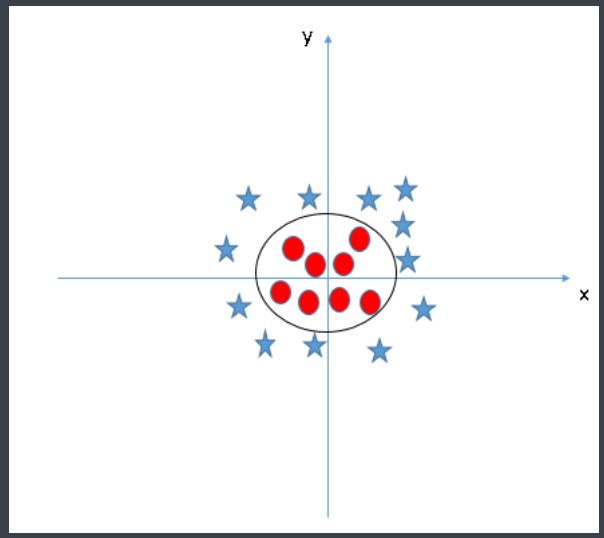


Scenario 5

- Points to consider are:
  - All values for z would be positive always because z is the squared sum of both x and y
  - In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z.
- In SVM, it is easy to have a linear hyper-plane between these two classes, but for such scenarios, SVM uses a trick called as **Kernel**.
- These are functions which takes low dimensional input space and transform it to a higher dimensional space, i.e. it converts non separable problem to separable problem
- It is mostly useful in non-linear separation problem
- Simply put, it does some extremely complex data transformations,

then find out the process to separate the data based on the

labels or outputs you've defined.



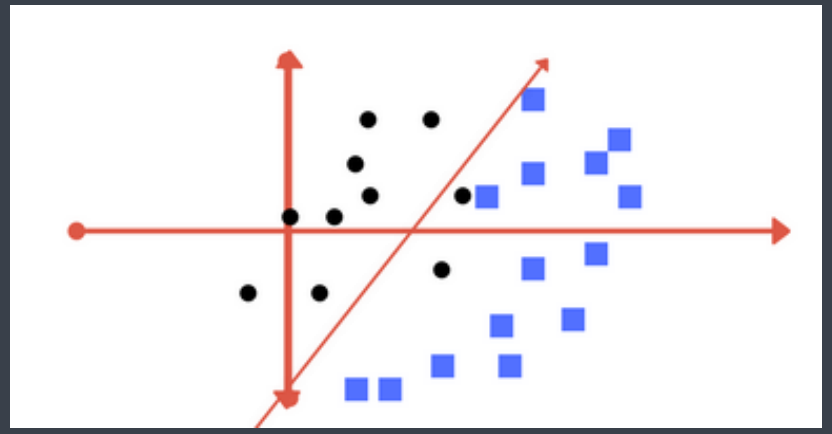
Tuning Parameters - Kernels

- The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra. This is where the kernel plays role.

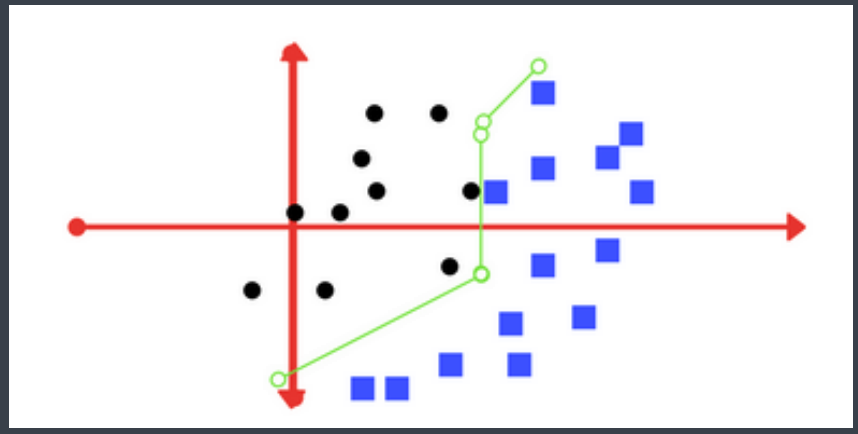
Tuning Parameters - Regularization

- The Regularization parameter tells the SVM optimization how much you want to avoid misclassifying each training example
- For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly
- Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points

Low regularization

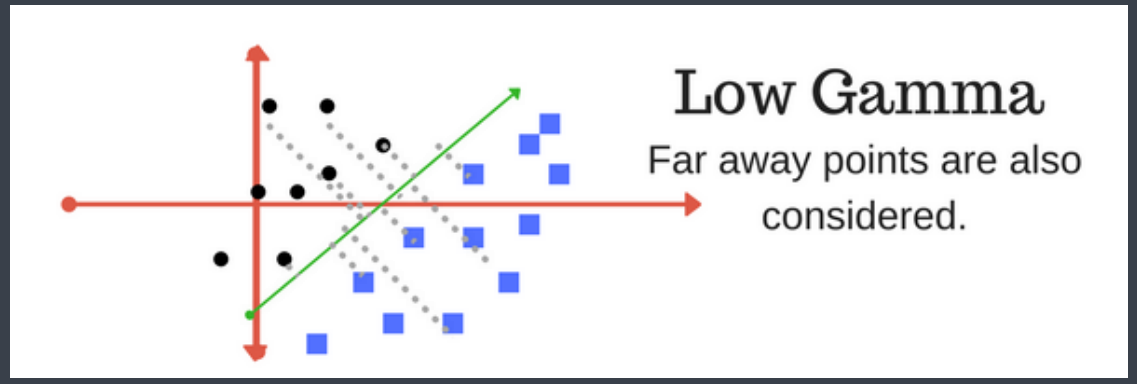


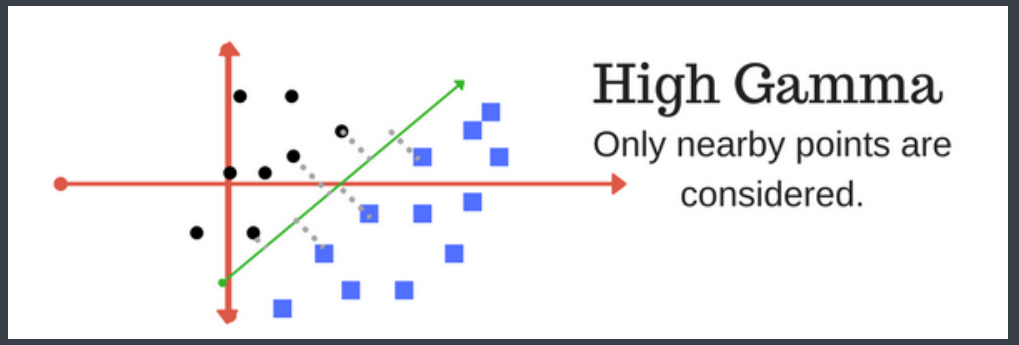
High regularization



Tuning Parameters - Gamma

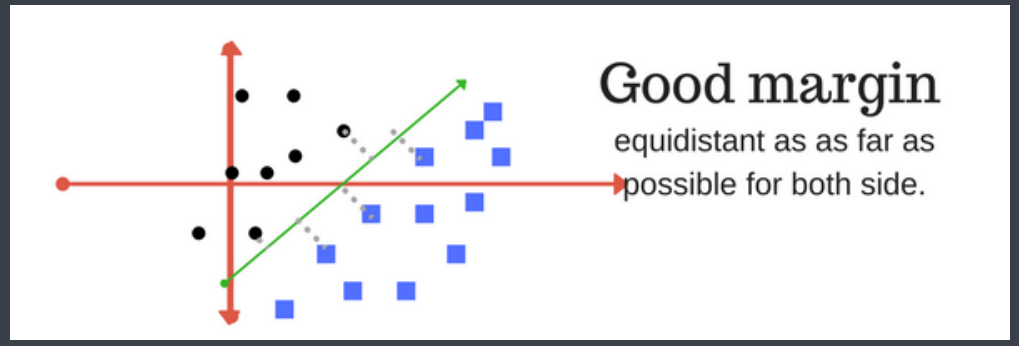
- The gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’
- In other words
  - With low gamma, points far away from plausible separation line are considered in calculation for the separation line
  - Where as high gamma means the points close to plausible line are considered in calculation.

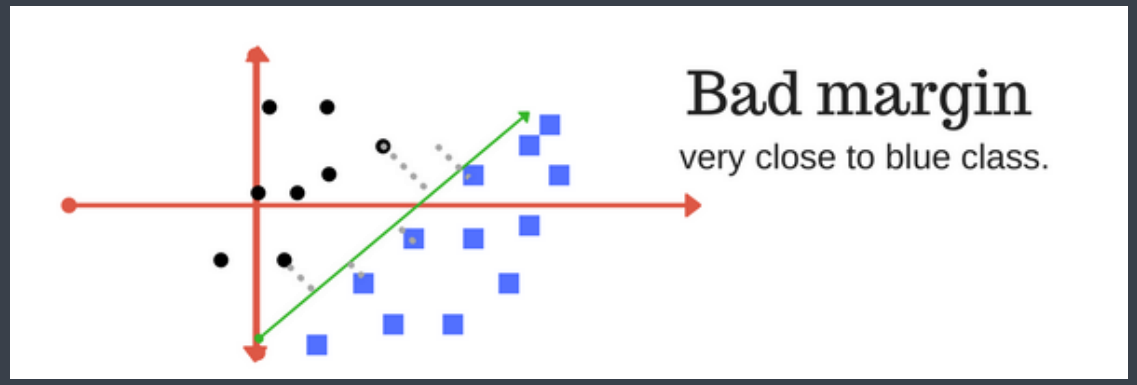




Tuning Parameters - Margin

- A margin is a separation of line to the closest class points
- A good margin is one where this separation is larger for both the classes





Decision Tree

Overview

- A **decision tree** is a decision support tool that uses a tree like model of decisions and their possible consequences
- A decision tree is a flowchart-like structure in which each internal node represents a “test” on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules
- It is one way to display an algorithm that only contains conditional control statements
- Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods
- Tree based methods empower predictive models with high accuracy, stability and ease of interpretation
- Unlike linear models, they map non-linear relationships quite well
- Decision Tree algorithms are referred to as **CART (Classification and Regression Trees)**

Terminologies

- **Root Node:** It represents entire population or sample and this further gets divided into two or more homogeneous sets.
- **Splitting:** It is a process of dividing a node into two or more sub-nodes.
- **Decision Node:** When a sub-node splits into further sub-nodes, then it is called decision node.
- **Leaf/ Terminal Node:** Nodes do not split is called Leaf or Terminal node.
- **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.
- **Branch / Sub-Tree:** A sub section of entire tree is called branch or sub-tree.
- **Parent and Child Node:** A node, which is divided into sub-nodes is called parent node of sub-nodes whereas sub-nodes are the child of parent node.

Applications of Decision Tree

- It is one of the more popular classification algorithms being used in Data Mining
- Determination of likely buyers of a product using demographic data to enable targeting of limited advertisement budget
- Prediction of likelihood of default for applicant borrowers using predictive models generated from historical data
- Help with prioritization of emergency room patient treatment using a predictive model based on factors such as age, blood pressure, gender, location etc.
- Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goaln, and other measurements
- Because of their simplicity, tree diagrams have been used in a broad range of industries and disciplines including civil planning, energy, financial, engineering, healthcare, pharmaceutical, education, law, and business

How does Decision Tree work?

- Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems
- It works for both categorical and continuous input and output variables
- In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables

Steps

- Place the best attribute of the dataset at the **root** of the tree.
- Split the training set into **subsets**. Subsets should be made in such a way that each subset contains data with the same value for an attribute.
- Repeat step 1 and step 2 on each subset until you find **leaf nodes** in all the branches of the tree.

Assumptions

- At the beginning, the whole training set is considered as the **root**
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are **distributed recursively** on the basis of attribute values
- Order to placing attributes as root or internal node of the tree is done by using some statistical approach

Decision Tree Types

- **Categorical Variable Decision Tree**
  - Decision Tree which has categorical target variable then it called as categorical variable decision tree
  - E.g.:- In an scenario of students data, where the target variable was “Student will play cricket or not” i.e. YES or NO.
- **Continuous Variable Decision Tree**
  - Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree

Advantages of Decision Tree

- **Easy to Understand**
  - Decision tree output is very easy to understand even for people from non-analytical background
  - It does not require any statistical knowledge to read and interpret them
  - Its graphical representation is very intuitive and users can easily relate their hypothesis
- **Useful in Data exploration**
  - Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables
  - With the help of decision trees, we can create new variables / features that has better power to predict target variable
  - It can also be used in data exploration stage
  - For e.g., we are working on a problem where we have information available in hundreds of variables, there decision tree will help to identify most significant variable.

Advantages of Decision Tree

- Decision trees implicitly perform variable screening or feature selection
- Decision trees require relatively little effort from users for data preparation
- **Less data cleaning required**
  - It requires less data cleaning compared to some other modelling techniques
  - It is not influenced by outliers and missing values to a fair degree.
- **Data type is not a constraint**
  - It can handle both numerical and categorical variables
- **Non-Parametric Method**
  - Decision tree is considered to be a non-parametric method
  - This means that decision trees have no assumptions about the space distribution and the classifier structure
- Non-linear relationships between parameters do not affect tree performance.
- The number of hyper-parameters to be tuned is almost null.

Disadvantages of Decision Tree

- **Over fitting**
  - Decision-tree learners can create over-complex trees that do not generalize the data well. This is called overfitting.
  - Over fitting is one of the most practical difficulty for decision tree models
  - This problem gets solved by setting constraints on model parameters and pruning
- **Not fit for continuous variables**
  - While working with continuous numerical variables, decision tree loses information, when it categorizes variables in different categories.
- Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This is called variance, which needs to be lowered by methods like bagging and boosting
- Decision tree learners create *biased* trees if some classes dominate. It is therefore recommended to balance the data set prior to fitting with the decision tree
- Calculations can become complex when there are many class label

Regression and Classification Tree

- Regression trees are used when dependent variable is continuous. Classification Trees are used when dependent variable is categorical
- In case of Regression Tree, the value obtained by terminal nodes in the training data is the mean response of observation falling in that region. Thus, if an unseen data observation falls in that region, we'll make its prediction with mean value.
- In case of Classification Tree, the value (class) obtained by terminal node in the training data is the mode of observations falling in that region. Thus, if an unseen data observation falls in that region, we'll make its prediction with mode value.
- Both the trees divide the predictor space (independent variables) into distinct and non-overlapping regions.
- Both the trees follow a top-down greedy approach known as recursive binary splitting.
- This splitting process is continued until a user defined stopping criteria is reached

Commonly used algorithms - Gini Index

- Gini index says, if we select two items from a population at random then they must be of same class and probability for this is 1 if population is pure
- It works with categorical target variable “Success” or “Failure”.
- It performs only Binary splits
- Higher the value of Gini higher the homogeneity.
- CART (Classification and Regression Tree) uses Gini method to create binary splits.

Commonly used algorithms - Chi-Square

- It is an algorithm to find out the statistical significance between the differences between sub-nodes and parent node
- We measure it by sum of squares of standardized differences between observed and expected frequencies of target variable
- It works with categorical target variable “Success” or “Failure”.
- It can perform two or more splits.
- Higher the value of Chi-Square higher the statistical significance of differences between sub-node and Parent node
- It generates tree called CHAID (Chi-square Automatic Interaction Detector)

Commonly used algorithms - Information Gain

- Less impure node requires less information to describe it. And, more impure node requires more information
- Information theory is a measure to define this degree of disorganization in a system known as Entropy
- If the sample is completely homogeneous, then the entropy is zero and if the sample is an equally divided (50% – 50%), it has entropy of one
- Entropy can be calculated using formula

$$\text{Entropy} = -p \log_2 p - q \log_2 q$$

- **Steps to calculate entropy for a split:**
  - Calculate entropy of parent node
  - Calculate entropy of each individual node of split and calculate weighted average of all sub-nodes available in split.
  - We can derive information gain from entropy as **1- Entropy.**