

Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.

Some common steps in data preprocessing include:

Data preprocessing is an important step in the data mining process that involves cleaning and transforming raw data to make it suitable for analysis. Some common steps in data preprocessing include:

Data Cleaning: This involves identifying and correcting errors or inconsistencies in the data, such as missing values, outliers, and duplicates. Various techniques can be used for data cleaning, such as imputation, removal, and transformation.

Data Integration: This involves combining data from multiple sources to create a unified dataset. Data integration can be challenging as it requires handling data with different formats, structures, and semantics. Techniques such as record linkage and data fusion can be used for data integration.

Data Transformation: This involves converting the data into a suitable format for analysis. Common techniques used in data transformation include normalization, standardization, and discretization. Normalization is used to scale the data to a common range, while standardization is used to transform the data to have zero mean and unit variance. Discretization is used to convert continuous data into discrete categories.

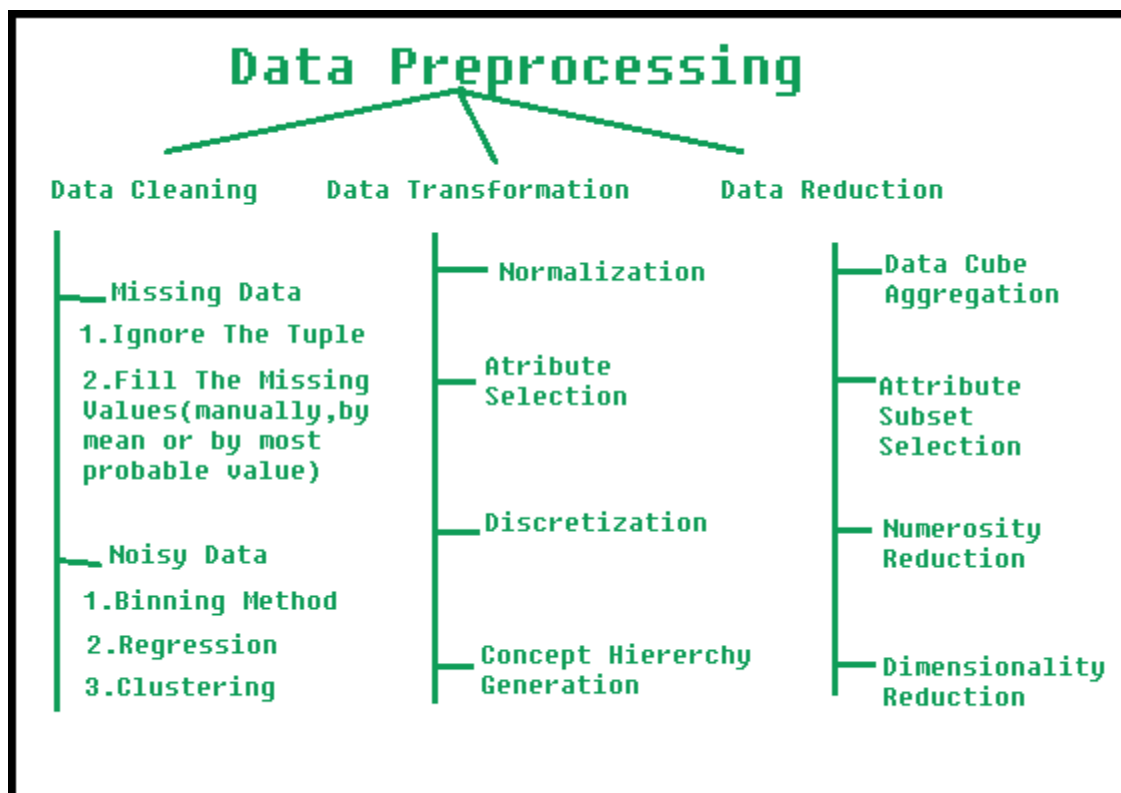
Data Reduction: This involves reducing the size of the dataset while preserving the important information. Data reduction can be achieved through techniques such as feature selection and feature extraction. Feature selection involves selecting a subset of relevant features from the dataset, while feature extraction involves transforming the data into a lower-dimensional space while preserving the important information.

Data Discretization: This involves dividing continuous data into discrete categories or intervals. Discretization is often used in data mining and machine learning algorithms that require categorical data. Discretization can be achieved through techniques such as equal width binning, equal frequency binning, and clustering.

Data Normalization: This involves scaling the data to a common range, such as between 0 and 1 or -1 and 1. Normalization is often used to handle data with different units and scales. Common normalization techniques include min-max normalization, z-score normalization, and decimal scaling.

Data preprocessing plays a crucial role in ensuring the quality of data and the accuracy of the analysis results. The specific steps involved in data preprocessing may vary depending on the nature of the data and the analysis goals.

By performing these steps, the data mining process becomes more efficient and the results become more accurate.



1. Data Cleaning:

The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data,

noisy data etc.

- **(a). Missing Data:**

This situation arises when some data is missing in the data. It can be handled in various ways.

Some of them are:

1. **Ignore the tuples:**

This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

2. **Fill the Missing values:**

There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

- **(b). Noisy Data:**

Noisy data is a meaningless data that can't be interpreted by machines. It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways :

1. **Binning Method:**

This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.

2. **Regression:**

Here data can be made smooth by fitting it to a regression function. The regression used may be linear (having one independent variable) or multiple (having multiple independent variables).

3. **Clustering:**

This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

2. Data Transformation:

This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

1. Normalization:

It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)

2. Attribute Selection:

In this strategy, new attributes are constructed from the given set of attributes to help the mining process.

3. Discretization:

This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.

4. Concept Hierarchy Generation:

Here attributes are converted from lower level to higher level in hierarchy. For Example-The attribute “city” can be converted to “country”.

3. Data Reduction:

Data reduction is a crucial step in the data mining process that involves reducing the size of the dataset while preserving the important information. This is done to improve the efficiency of data analysis and to avoid overfitting of the model. Some common steps involved in data reduction are:

1. **Feature Selection:** This involves selecting a subset of relevant features from the dataset. Feature selection is often performed to remove irrelevant or redundant features from the dataset. It can be done using various techniques such as correlation analysis, mutual information, and principal component analysis (PCA).

2. **Feature Extraction:** This involves transforming the data into a lower-dimensional space while preserving the important information. Feature extraction is often used when the original features are high-dimensional and complex. It can be done using techniques such as PCA, linear

discriminant analysis (LDA), and non-negative matrix factorization (NMF).

3. **Sampling:** This involves selecting a subset of data points from the dataset. Sampling is often used to reduce the size of the dataset while preserving the important information. It can be done using techniques such as random sampling, stratified sampling, and systematic sampling.
4. **Clustering:** This involves grouping similar data points together into clusters. Clustering is often used to reduce the size of the dataset by replacing similar data points with a representative centroid. It can be done using techniques such as k-means, hierarchical clustering, and density-based clustering.
5. **Compression:** This involves compressing the dataset while preserving the important information. Compression is often used to reduce the size of the dataset for storage and transmission purposes. It can be done using techniques such as wavelet compression, JPEG compression, and gzip compression.

What is Frequent-pattern Analysis?

Frequent pattern: A pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a dataset.

Motivation:

Finding inherent regularities in data: What products are often purchased together? Beer and diapers?!

What are the subsequent purchases after buying a PC?

FPGrowth: a frequent-pattern-growth approach.

"Who bought this has often also bought . . ." What kinds of DNA are sensitive to this new drug?

Can we automatically classify Web documents?

Applications: Basket-data analysis, cross-marketing, catalog design,

sale-campaign analysis, Web-log (click-stream) analysis, and DNA-sequence analysis

Basic Concepts:

Frequent Itemsets: These are subsets of items that frequently occur together in a dataset.

Support: The frequency of occurrence of an itemset in the dataset. It's typically measured as the proportion of transactions in which the itemset appears.

Association Rules: These are implications of the form $X \rightarrow Y$, where X and Y are itemsets, indicating that if X occurs, Y is likely to occur as well.

Support and Confidence: Support measures how frequently the rule is applicable, while confidence measures the reliability of the implication.

Association refers to the identification of relationships or patterns in data where certain events or items tend to occur together. In the context of data mining, particularly in frequent pattern mining, association refers to finding associations or correlations between different items or attributes in a dataset. The most common technique used for association mining is the discovery of association rules.

Association Rule Mining:

Association rule mining aims to discover interesting relationships between variables in large datasets. These relationships are often represented as association rules of the form "if X then Y ," where X and Y are itemsets.

Basic Concepts:

1. **Association Rules:** Implications of the form $X \rightarrow Y$, indicating that if X occurs, Y is likely to occur as well.
2. **Support:** The frequency of occurrence of an itemset in the dataset. It measures how frequently a rule is applicable.

3. **Confidence:** Measures the reliability of the implication, indicating the conditional probability of the consequent given the antecedent.
4. **Lift:** Measures the likelihood of the occurrence of the consequent given the antecedent, normalized by the individual probabilities of the antecedent and consequent.

Applications:

- **Market Basket Analysis:** Understanding customer purchasing behavior.
- **Recommendation Systems:** Recommending items based on past user behavior.
- **Bioinformatics:** Discovering associations between genes and diseases.

In the context of mining frequent patterns, associations, and correlations, correlation typically refers to the relationship between different items or attributes in a dataset. While correlation analysis is more commonly associated with numerical data, in the context of frequent pattern mining, it can still be explored.

Correlation Measures:

1. **Pearson Correlation Coefficient:** It measures the linear correlation between two numerical variables. However, it might not capture non-linear relationships or relationships between non-numeric attributes.
2. **Spearman's Rank Correlation Coefficient:** It measures the strength and direction of association between the rankings of two variables. It's more robust to outliers and can capture monotonic relationships.
3. **Pointwise Mutual Information (PMI):** It measures the strength of association between two items in terms of their co-occurrence compared to their individual occurrence probabilities. It's commonly used in text mining and collaborative filtering.

Correlation Analysis in Association Rule Mining:

While traditional correlation measures may not directly apply to association rule mining, some techniques can infer relationships between items based on their co-occurrence patterns.

Extended Approaches:

1. **Correlation-Based Pruning:** In the context of frequent pattern mining algorithms like Apriori or FP-Growth, correlation measures can be used to prune uninteresting or redundant itemsets or association rules.
2. **Sequential Pattern Mining:** In sequences of events or transactions, correlation measures can help identify temporal relationships between items.

There are several different algorithms used for frequent pattern mining, including:

1. **Apriori algorithm:** This is one of the most commonly used algorithms for frequent pattern mining. It uses a “bottom-up” approach to identify frequent itemsets and then generates association rules from those itemsets.
2. **ECLAT algorithm:** This algorithm uses a “depth-first search” approach to identify frequent itemsets. It is particularly efficient for datasets with a large number of items.
3. **FP-growth algorithm:** This algorithm uses a “compression” technique to find frequent patterns efficiently. It is particularly efficient for datasets with a large number of transactions.

Market Basket Analysis

Market Basket Analysis is one of the key techniques used by large retailers to uncover associations between items.

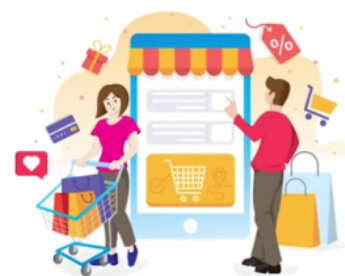


Bread and Jam

Laptop and Bag



associations between different items and products that can be sold together which



Association Rule Mining

$A \Rightarrow B$

$$Support = \frac{freq(A, B)}{N}$$

$$Confidence = \frac{freq(A, B)}{freq(A)}$$

$$Lift = \frac{Support}{Supp(A) \times Supp(B)}$$

93

Association Rule Mining



Rule	Support	Confidence	Lift
$A \Rightarrow D$	2/5	2/3	10/9
$C \Rightarrow A$	2/5	2/4	5/6
$A \Rightarrow C$	2/5	2/3	5/6
$B, C \Rightarrow A$	1/5	1/3	5/9

Apriori Algorithm

Apriori algorithm uses frequent item sets to generate association rules. It is based on the concept that a subset of a frequent itemset must also be a frequent itemset.



Frequent Itemset is an itemset whose support value is greater than a threshold value.

SU
CR
e

Apriori Algorithm Explained | Association Rule Mining | Finding Fre... Apriori Algorithm - 1st Iteration Watch later

C1

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5



Itemset	Support
{1}	3
{2}	3
{3}	4
{4}	1
{5}	4

Apriori Algorithm - 1st Iteration

C1

Itemset	Support
{1}	3
{2}	3
{3}	4
{4}	1
{5}	4

F1

Itemset	Support
{1}	3
{2}	3
{3}	4
{5}	4



Item sets with support value less than min. support value (i.e. 2) are eliminated

Apriori Algorithm - 2nd Iteration

Only Items present in F1

C2

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5



Itemset	Support
{1,2}	1
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3



F2

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

Item sets with support value less than min. support value (i.e. 2) are eliminated

Apriori Algorithm – Pruning

C3 ?

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5



Itemset	Support
{1,2,3}	
{1,2,5}	
{1,3,5}	
{2,3,5}	

Apriori Algorithm – Pruning

C3

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5



Itemset	In F2?
{1,2,3}, {1,2}, {1,3}, {2,3}	NO
{1,2,5}, {1,2}, {1,5}, {2,5}	NO
{1,3,5}, {1,5}, {1,3}, {3,5}	YES
{2,3,5}, {2,3}, {2,5}, {3,5}	YES

Apriori Algorithm – Pruning

F3

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5



Itemset	Support
{1,3,5}	2
{2,3,5}	2

Apriori Algorithm – 4th Iteration

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5



Itemset	Support
{1,3,5}	2
{2,3,5}	2

Itemset	Support
{1,2,3,5}	1

Apriori Algorithm – Subset Creation

Itemset	Support
{1,3,5}	2
{2,3,5}	2

For $I = \{1,3,5\}$, subsets are $\{1,3\}$, $\{1,5\}$, $\{3,5\}$, $\{1\}$, $\{3\}$, $\{5\}$

For $I = \{2,3,5\}$, subsets are $\{2,3\}$, $\{2,5\}$, $\{3,5\}$, $\{2\}$, $\{3\}$, $\{5\}$

- For every subsets S of I , output the rule:

$S \rightarrow (I-S)$ (S recommends $I-S$)

if $\text{support}(I)/\text{support}(S) \geq \text{min_conf value}$

SU
CR

Apriori Algorithm – Applying Rules

Applying Rules to Item set F3

1. $\{1,3,5\}$

- ✓ Rule 1: $\{1,3\} \rightarrow (\{1,3,5\} - \{1,3\})$ means $1 \ \& \ 3 \rightarrow 5$
Confidence = $\text{support}(1,3,5)/\text{support}(1,3) = 2/3 = 66.66\% > 60\%$
Rule 1 is selected
- ✓ Rule 2: $\{1,5\} \rightarrow (\{1,3,5\} - \{1,5\})$ means $1 \ \& \ 5 \rightarrow 3$
Confidence = $\text{support}(1,3,5)/\text{support}(1,5) = 2/2 = 100\% > 60\%$
Rule 2 is selected
- ✓ Rule 3: $\{3,5\} \rightarrow (\{1,3,5\} - \{3,5\})$ means $3 \ \& \ 5 \rightarrow 1$
Confidence = $\text{support}(1,3,5)/\text{support}(3,5) = 2/3 = 66.66\% > 60\%$
Rule 3 is selected

SU
CR

Apriori Algorithm – Applying Rules

Applying Rules to Item set F3

1. {1,3,5}

- ✓ Rule 4: $\{1\} \rightarrow (\{1,3,5\} - \{1\})$ means $1 \rightarrow 3 \ \& \ 5$
Confidence = $\text{support}(1,3,5)/\text{support}(1) = 2/3 = 66.66\% > 60\%$
Rule 4 is selected
- ✓ Rule 5: $\{3\} \rightarrow (\{1,3,5\} - \{3\})$ means $3 \rightarrow 1 \ \& \ 5$
Confidence = $\text{support}(1,3,5)/\text{support}(3) = 2/4 = 50\% < 60\%$
Rule 5 is rejected
- ✓ Rule 6: $\{5\} \rightarrow (\{1,3,5\} - \{5\})$ means $5 \rightarrow 1 \ \& \ 3$
Confidence = $\text{support}(1,3,5)/\text{support}(5) = 2/4 = 50\% < 60\%$
Rule 6 is rejected

- **Apriori Algorithm:**

The Apriori algorithm is one of the most well-known and widely used algorithms for repeating arrangement prospecting. It uses a breadth-first search strategy to discover repeating groupings efficiently. The algorithm works in multiple iterations. It starts by finding repeating individual objects by scanning the database once and counting the occurrence of each object. It then generates candidate groupings of size 2 by combining the repeating groupings of size 1. The support of these candidate groupings is calculated by scanning the database again. The process continues iteratively, generating candidate groupings of size k and calculating their support until no more repeating groupings can be found.

Here are the main elements of the Apriori algorithm:

1. Frequent Itemsets:

- **Frequent Itemset:** A set of items (or itemsets) whose support (frequency of occurrence) in the dataset is greater than or equal to a specified threshold (min_support).

2. Apriori Principle:

- **Apriori Property:** If an itemset is frequent, then all of its subsets must also be frequent. This property is used to reduce the search space by pruning infrequent itemsets.

3. Candidate Generation:

- **Candidate Itemset:** A potentially frequent itemset that needs to be verified.
- **Join Operation:** Involves combining frequent itemsets of size k to generate candidate itemsets of size $k+1$.
- **Prune Operation:** Eliminates candidate itemsets that contain subsets that are infrequent.

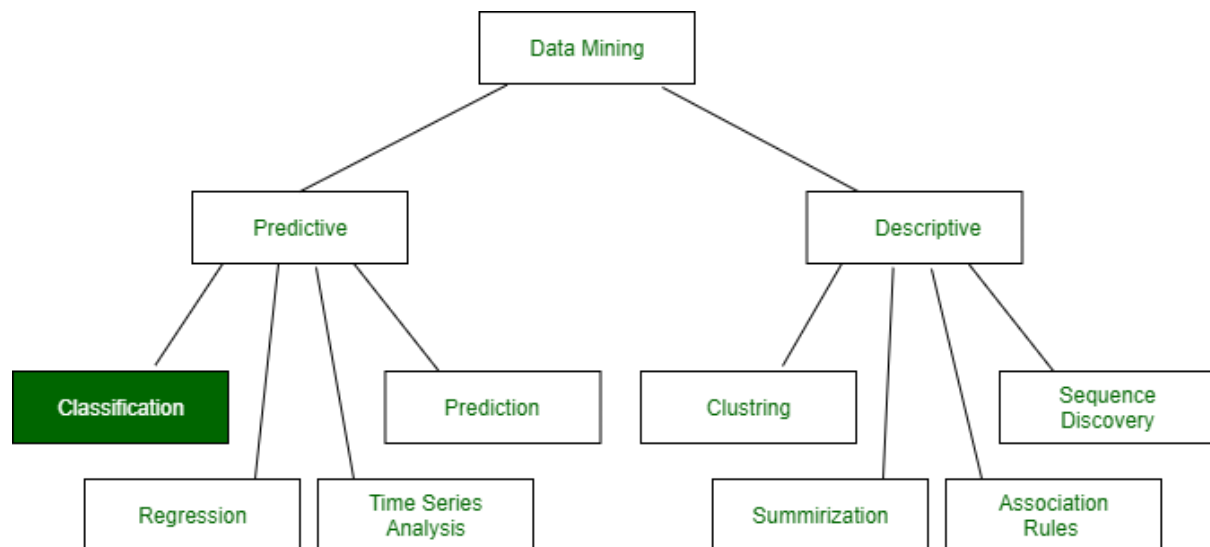
4. Support Counting:

- **Support Count:** The number of transactions containing a particular itemset.
- **Support Threshold:** A minimum threshold set by the user to determine which itemsets are considered frequent.

5. Iterative Process:

- **Iterative Exploration:** The algorithm iteratively explores the search space by incrementally increasing the size of itemsets until no new frequent itemsets can be found.

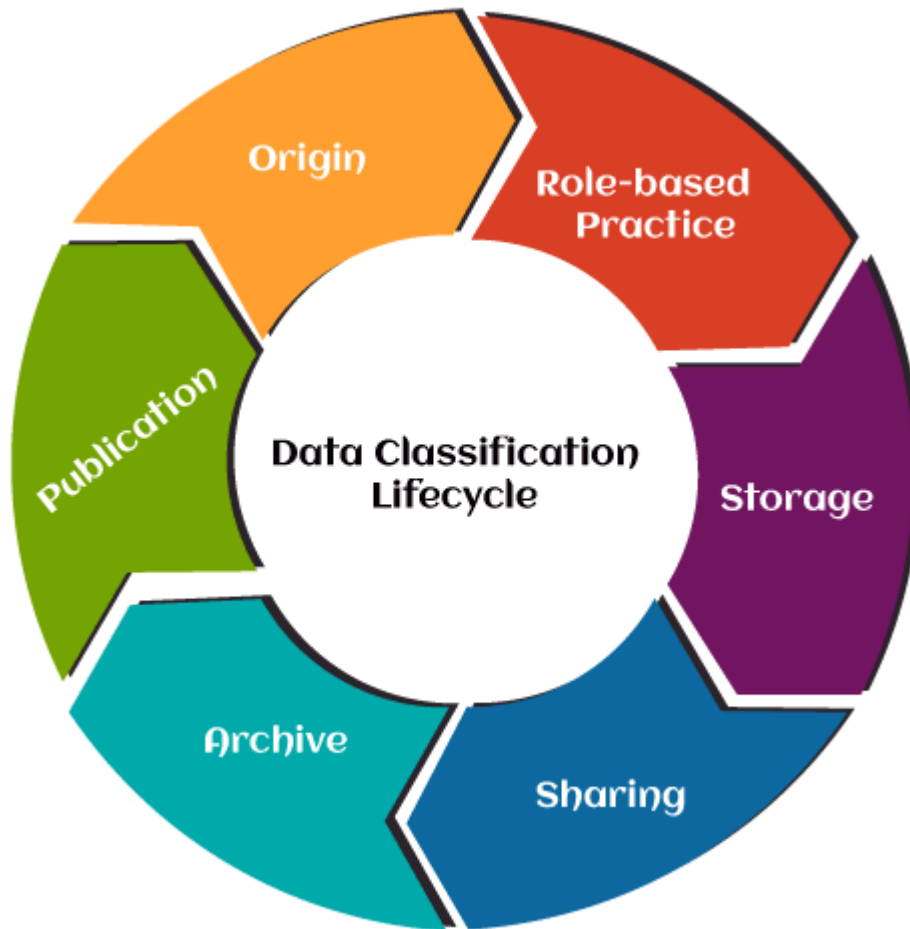
UNIT 3



Data Mining: Data mining in general terms means mining or digging deep into data that is in different forms to gain patterns, and to gain knowledge on that pattern. In the process of data mining, large data sets are first sorted, then patterns are identified and relationships are established to perform data analysis and solve problems.

Classification is a task in data mining that involves assigning a class label to each instance in a dataset based on its features. The goal of classification is to build a model that accurately predicts the class labels of new instances based on their features.

There are two main types of classification: binary classification and multi-class classification. Binary classification involves classifying instances into two classes, such as “spam” or “not spam”, while multi-class classification involves classifying instances into more than two classes.



1. **Origin:** It produces sensitive data in various formats, with emails, Excel, Word, Google documents, social media, and websites.
2. **Role-based practice:** Role-based security restrictions apply to all delicate data by tagging based on in-house protection policies and agreement rules.
3. **Storage:** Here, we have the obtained data, including access controls and encryption.
4. **Sharing:** Data is continually distributed among agents, consumers, and co-workers from various devices and platforms.
5. **Archive:** Here, data is eventually archived within an industry's storage systems.

The process of building a classification model typically involves the following steps:

Data Collection:

The first step in building a classification model is data collection. In this step, the data relevant to the problem at hand is collected. The data should be representative of the problem and should contain all the necessary attributes and labels needed for classification. The data can be collected from various sources, such as surveys, questionnaires, websites, and databases.

Data Preprocessing:

The second step in building a classification model is data preprocessing. The collected data needs to be preprocessed to ensure its quality. This involves handling missing values, dealing with outliers, and transforming the data into a format suitable for analysis. Data preprocessing also involves converting the data into numerical form, as most classification algorithms require numerical input.

Feature Selection:

The third step in building a classification model is feature selection. Feature selection involves identifying the most relevant attributes in the dataset for classification. This can be done using various techniques, such as correlation analysis, information gain, and principal component analysis.

Correlation Analysis: Correlation analysis involves identifying the correlation between the features in the dataset. Features that are highly correlated with each other can be removed as they do not provide additional information for classification.

Information Gain: Information gain is a measure of the amount of information that a feature provides for classification. Features with high information gain are selected for classification.

Principal Component Analysis:

Principal Component Analysis (PCA) is a technique used to reduce the dimensionality of the dataset. PCA identifies the most important features in the dataset and removes the redundant ones.

Model Selection:

The fourth step in building a classification model is model selection. Model selection involves selecting the appropriate classification algorithm for the problem at hand. There are several algorithms available, such as decision trees, support vector machines, and neural networks.

Decision Trees: Decision trees are a simple yet powerful classification algorithm. They divide the dataset into smaller subsets based on the values of the features and construct a tree-like model that can be used for classification.

Support Vector Machines: Support Vector Machines (SVMs) are a popular classification algorithm used for both linear and nonlinear classification problems. SVMs are based on the concept of maximum margin, which involves finding the hyperplane that maximizes the distance between the two classes.

Neural Networks:

Neural Networks are a powerful classification algorithm that can learn complex patterns in the data. They are inspired by the structure of the human brain and consist of multiple layers of interconnected nodes.

Model Training:

The fifth step in building a classification model is model training. Model training involves using the selected classification algorithm to learn the patterns in the data. The data is divided into a training set and a validation set. The model is trained using the training set, and its performance is evaluated on the validation set.

Model Evaluation:

The sixth step in building a classification model is model evaluation. Model evaluation involves assessing the performance of the trained model on a test set. This is done to ensure that the model generalizes well

Classification: It is a data analysis task, i.e. the process of finding a model that describes and distinguishes data classes and concepts. Classification is the problem of identifying to which of a set of categories (subpopulations), a new observation belongs to, on the basis of a training set of data containing observations and whose categories membership is known.

Example: Before starting any project, we need to check its feasibility. In this case, a classifier is required to predict class labels such as 'Safe' and 'Risky' for adopting the Project and to further approve it. It is a two-step process such as:

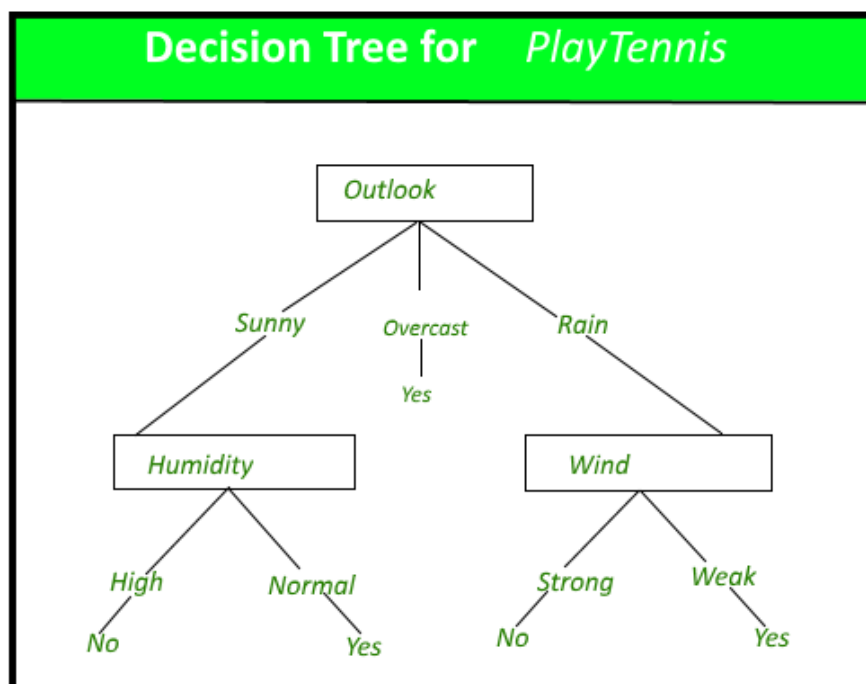
1. **Learning Step (Training Phase):** Construction of Classification Model
Different Algorithms are used to build a classifier by making the model learn using the training set available. The model has to be trained for the prediction of accurate results.
2. **Classification Step:** Model used to predict class labels and testing the constructed model on test data and hence estimate the accuracy of the classification rules.

Decision Tree Induction in Data Mining

- Decision tree induction is a common technique in data mining that is used to generate a predictive model from a dataset. This technique involves constructing a tree-like structure, where each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a prediction. The goal of decision tree induction is to build a model that can accurately predict the outcome of a given event, based on the values of the attributes in the dataset.
- To build a decision tree, the algorithm first selects the attribute that best splits the data into distinct classes. This is typically done using a measure of impurity, such as entropy or the Gini index, which measures the degree of disorder in the data. The algorithm

then repeats this process for each branch of the tree, splitting the data into smaller and smaller subsets until all of the data is classified.

- Decision tree induction is a popular technique in data mining because it is easy to understand and interpret, and it can handle both numerical and categorical data. Additionally, decision trees can handle large amounts of data, and they can be updated with new data as it becomes available. However, decision trees can be prone to overfitting, where the model becomes too complex and does not generalize well to new data. As a result, data scientists often use techniques such as pruning to simplify the tree and improve its performance.



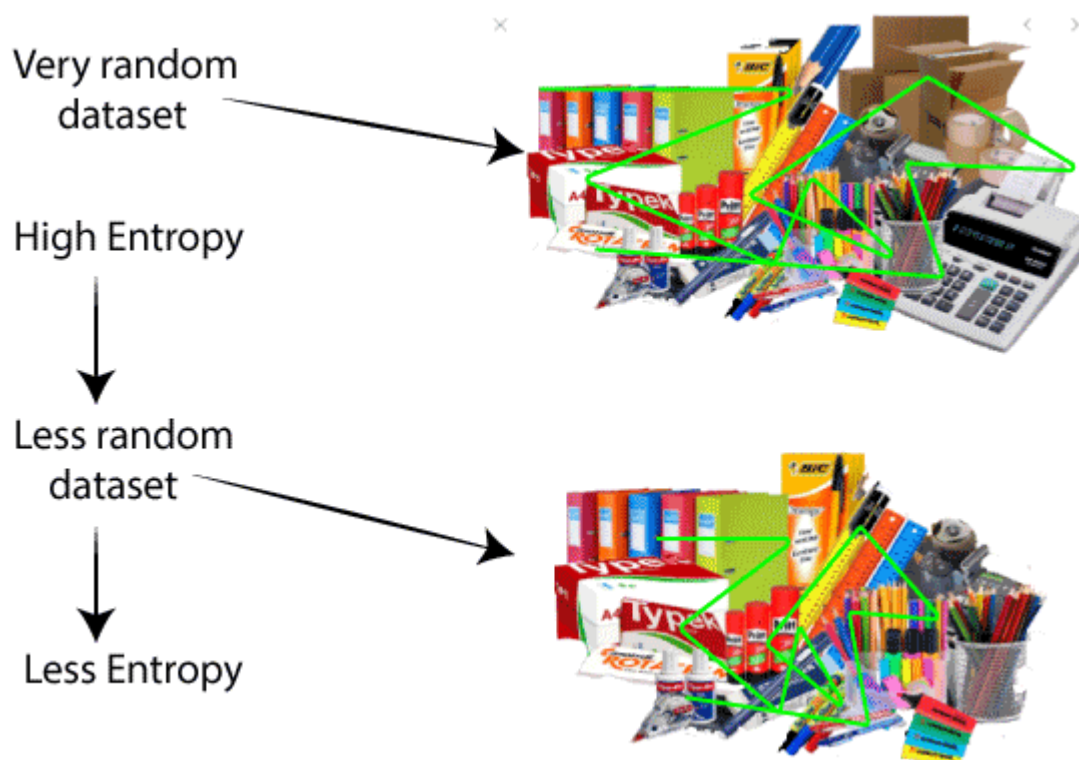
In other words, we can say that a decision tree is a hierarchical tree structure that can be used to split an extensive collection of records into smaller sets of the class by implementing a sequence of simple decision rules. A decision tree model comprises a set of rules for portioning a huge heterogeneous population into smaller, more homogeneous, or mutually exclusive classes. The attributes of the classes can be any variables from nominal, ordinal, binary, and quantitative values, in contrast, the classes must be a qualitative type, such as

categorical or ordinal or binary. In brief, the given data of attributes together with its class, a decision tree creates a set of rules that can be used to identify the class. One rule is implemented after another, resulting in a hierarchy of segments within a segment. The hierarchy is known as the **tree**, and each segment is called a **node**. With each progressive division, the members from the subsequent sets become more and more similar to each other. Hence, the algorithm used to build a decision tree is referred to as recursive partitioning. The algorithm is known as **CART** (Classification and Regression Trees)

Key factors:

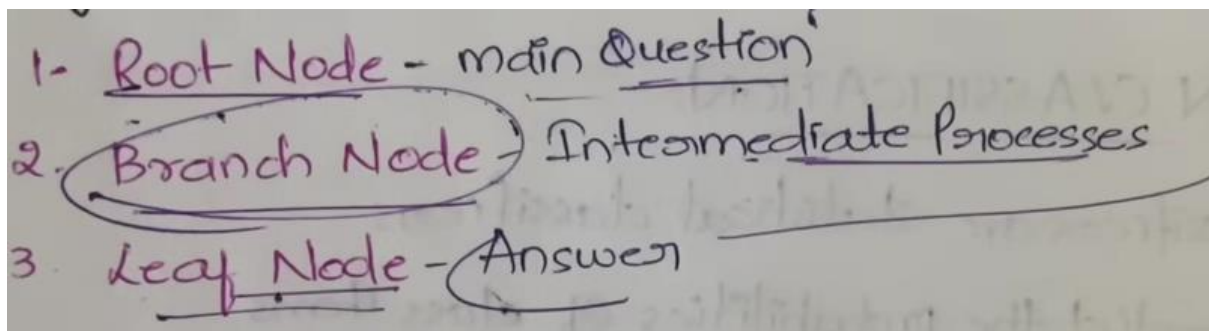
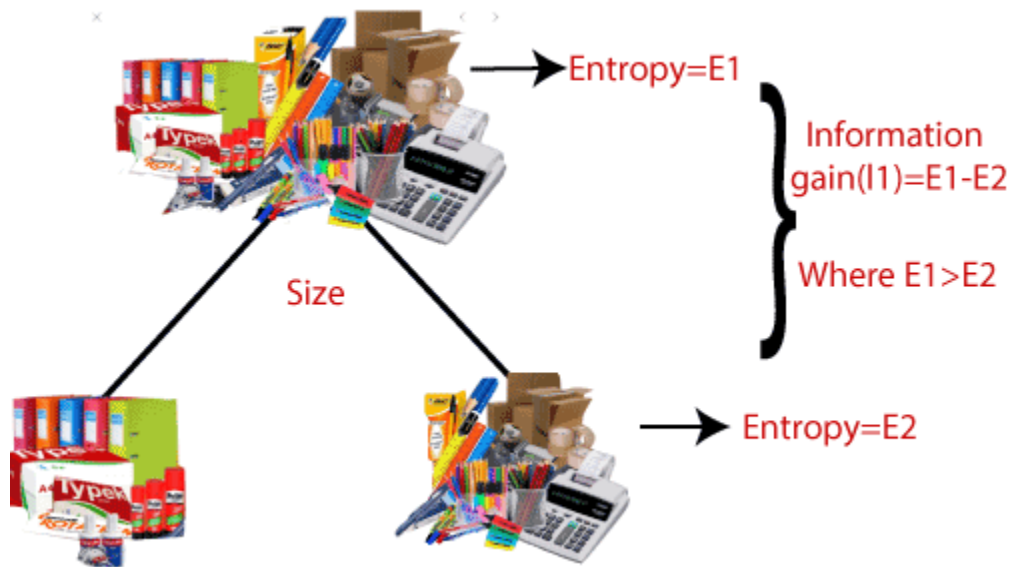
Entropy:

Entropy refers to a common way to **measure impurity**. In the decision tree, it measures the **randomness or impurity in data sets**.



Information Gain:

Information Gain refers to the decline in entropy after the dataset is split. It is also called **Entropy Reduction**. Building a decision tree is all about discovering attributes that return the highest data gain.



Imagine you have a big decision to make, like whether to go out or stay in on a weekend. You might make this decision based on various factors like weather, mood, plans with friends, etc. Now, think of a decision tree as a visual representation of how you make this decision.

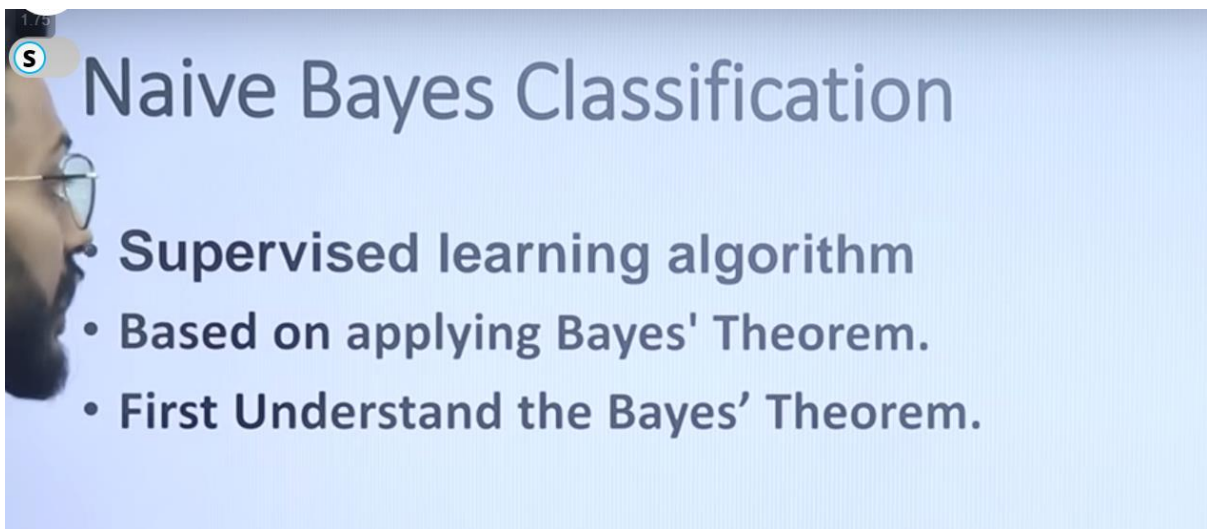
Here's how it works:

1. **Start with a Big Question:** At the top of the tree, you have a big question, like "Should I go out this weekend?"
2. **Split into Smaller Questions:** Based on different factors that influence your decision, you split this big question into smaller questions. For example, you might ask:
 - Is the weather nice?
 - Am I feeling energetic?
 - Do I have plans with friends?
3. **Follow the Branches:** Each of these smaller questions becomes a branch of the tree. Depending on your answers, you follow the branches until you reach a conclusion.
4. **Make a Decision:** Finally, at the end of each branch, you make a decision. For example:
 - If the weather is nice, and you're feeling energetic, you might decide to go out.
 - If the weather is nice, but you're tired and have no plans, you might decide to stay in.

Decision Tree Induction in Data Mining:

In data mining, decision tree induction works similarly. Instead of making decisions about going out, we're making decisions about data based on its features (like weather, mood, etc.).

- **Big Question:** Instead of "Should I go out?", we might have a big question like "Will a customer buy this product?"
- **Splitting Criteria:** Instead of factors like weather or mood, we have features of the data (like age, income, etc.). We ask questions like "Is the customer's income above \$50,000?"
- **Branches:** Each answer to a question leads us down a different branch of the tree.
- **Decision:** At the end of each branch, we make a decision, like "Yes, the customer will buy the product" or "No, the customer won't buy the product."



Naive Bayes Classification

Supervised learning algorithm

- Based on applying Bayes' Theorem.
- First Understand the Bayes' Theorem.

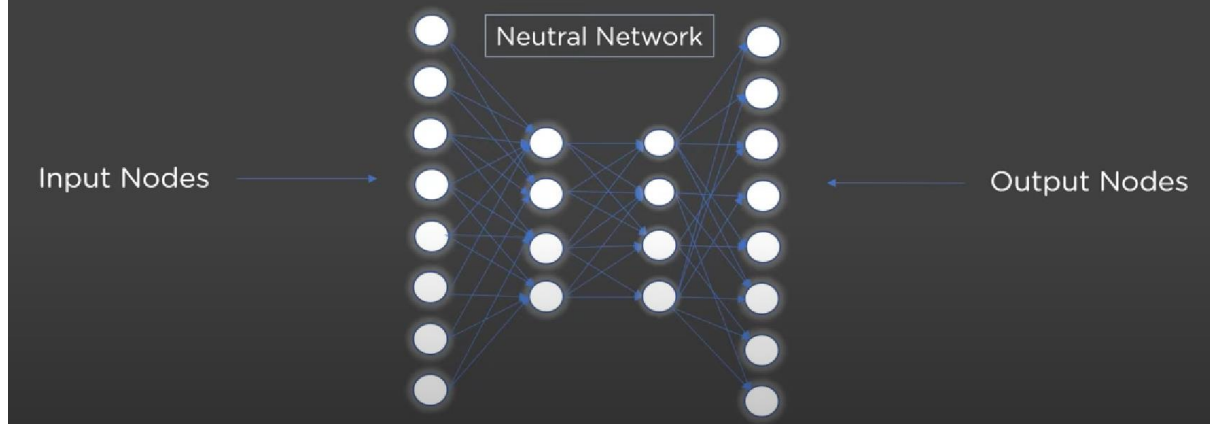
naïve mtlb hum maan kr chl rahe hai ki yaha sb variables independent hai

Baye's Classification Method:

Bayesian classification in data mining is a statistical approach to data classification that uses Bayes' theorem to make predictions about a class of a data point based on observed data. It is a popular data mining and machine learning technique for modelling the probability of certain outcomes and making predictions based on that probability.

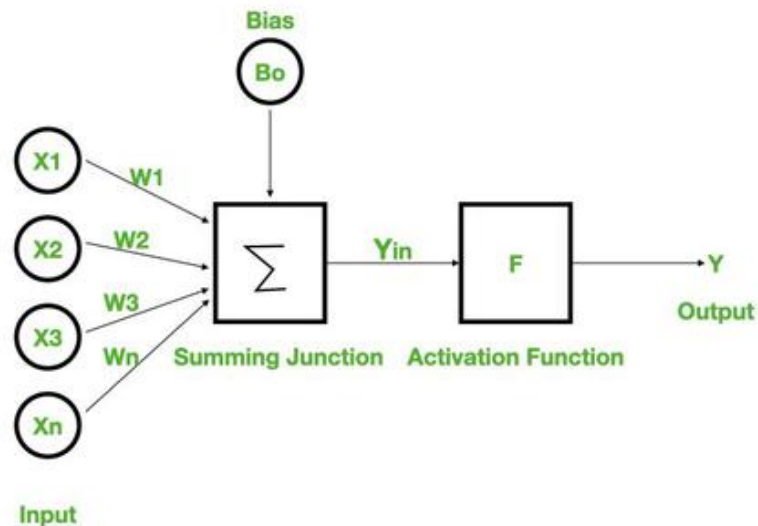
The basic idea behind Bayesian classification in data mining is to assign a class label to a new data instance based on the probability that it belongs to a particular class, given the observed data. Bayes' theorem provides a way to compute this probability by multiplying the prior probability of the class (based on previous knowledge or assumptions)

Backpropagation is an algorithm which is created to test errors which will travel back from input nodes to output nodes.



Backpropagation is an algorithm that backpropagates the errors from the output nodes to the input nodes. Therefore, it is simply referred to as the backward propagation of errors. It uses in the vast applications of neural networks in data mining like Character recognition, Signature verification, etc.

Neural networks are an information processing paradigm inspired by the human nervous system. Just like in the human nervous system, we have biological neurons in the same way in neural networks we have artificial neurons, artificial neurons are mathematical functions derived from biological neurons. The human brain is estimated to have about 10 billion neurons, each connected to an average of 10,000 other neurons. Each neuron receives a signal through a synapse, which controls the effect of the signal concerning on the neuron.



Classification by Backpropagation:

Classification by backpropagation is a type of supervised learning algorithm that is used to train a neural network to classify data into different classes. The backpropagation algorithm is based on the idea of adjusting the weights and biases of a network in order to minimize the error between the predicted output and the actual output.

The backpropagation algorithm works by taking a set of training examples and feeding them through the neural network. The output of the network is compared to the desired output, and the error is calculated using a cost function such as mean squared error.

The error is then propagated backwards through the network, with each neuron in the network adjusting its weights and biases based on its contribution to the error. This is done using a gradient descent algorithm, where the weights and biases are adjusted in the direction that reduces the error.

The backpropagation algorithm is an iterative process that continues until the error is minimized or until a predetermined number of iterations

is reached. The final set of weights and biases is then used to classify new data.

Types of Backpropagation

There are two types of backpropagation networks.

- **Static backpropagation:** Static backpropagation is a network designed to map static inputs for static outputs. These types of networks are capable of solving static classification problems such as OCR (Optical Character Recognition).
- **Recurrent backpropagation:** Recursive backpropagation is another network used for fixed-point learning. Activation in recurrent backpropagation is feed-forward until a fixed value is reached. Static backpropagation provides an instant mapping, while recurrent backpropagation does not provide an instant mapping.
- **Backpropagation Algorithm:**
 - **Step 1:** Inputs X , arrive through the preconnected path.
 - **Step 2:** The input is modeled using true weights W . Weights are usually chosen randomly.
 - **Step 3:** Calculate the output of each neuron from the input layer to the hidden layer to the output layer.
 - **Step 4:** Calculate the error in the outputs
 - **Backpropagation Error**= Actual Output – Desired Output
 - **Step 5:** From the output layer, go back to the hidden layer to adjust the weights to reduce the error.
 - **Step 6:** Repeat the process until the desired output is achieved.

Need for Backpropagation:

Backpropagation is “backpropagation of errors” and is very useful for training neural networks. It’s fast, easy to implement, and simple. Backpropagation does not require any parameters to be set, except the number of inputs. Backpropagation is a flexible method because no prior knowledge of the network is required.

Support Vector Machines:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

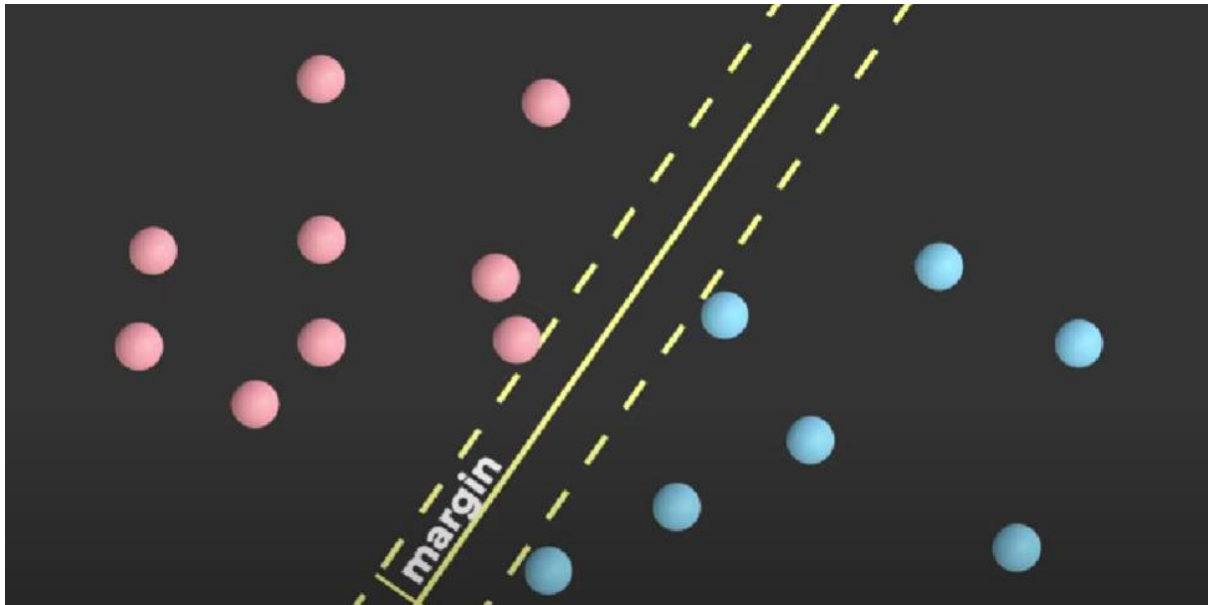
The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

SVMs are widely adopted across disciplines such as healthcare, natural language processing, signal processing applications, and speech & image recognition fields.

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.



Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. SVMs are adaptable and efficient in a variety of applications because they can manage high-dimensional data and nonlinear relationships.

SVM algorithms are very effective as we try to find the maximum separating hyperplane between the different classes available in the target feature.

Support Vector Machine

Support Vector Machine (SVM) is a [supervised machine learning](#) algorithm used for both classification and regression. Though we say regression problems as well it's best suited for classification. The main objective of the SVM algorithm is to find the optimal [hyperplane](#) in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

What is Lazy Learning?

Lazy learning algorithms work by memorizing the training data rather than constructing a general model.

Lazy learning is a type of machine learning that doesn't process training data until it needs to make a prediction. Instead of building models during training, lazy learning algorithms wait until they encounter a new query. This method stores and compares training examples when making predictions. It's also called instance-based or memory-based learning.

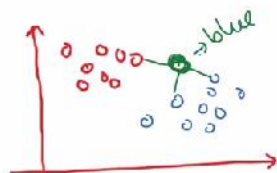
Lazy vs Eager



Eager Learners	Lazy Learners
<ul style="list-style-type: none">• Do lot of work on training data	<ul style="list-style-type: none">• Do less work on training data
<ul style="list-style-type: none">• Do less work when test tuples are presented	<ul style="list-style-type: none">• Do more work when test tuples are presented

Monday, July 25, 2022 4:43 PM

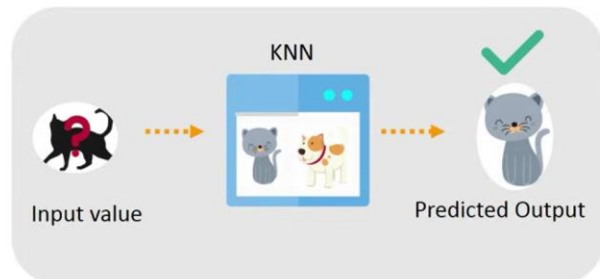
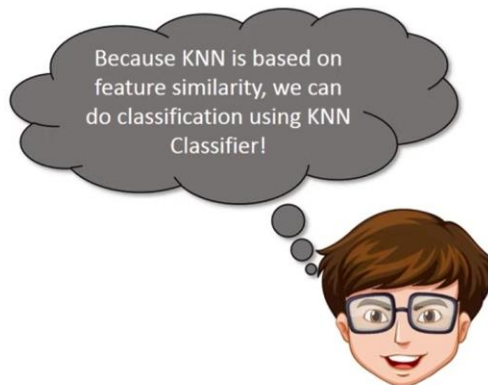
Why KNN is called lazy algorithm?



$K=3$
2 - Blue
1 - Red
/

It does not do anything in training phase.
It does not training at all when we supply the training data. It only stores data in training time.
All the computations happens during scoring that is when we apply the model on unseen data point!

Why KNN?



KNN - K Nearest Neighbors, is one of the simplest **Supervised** Machine Learning algorithm mostly used for

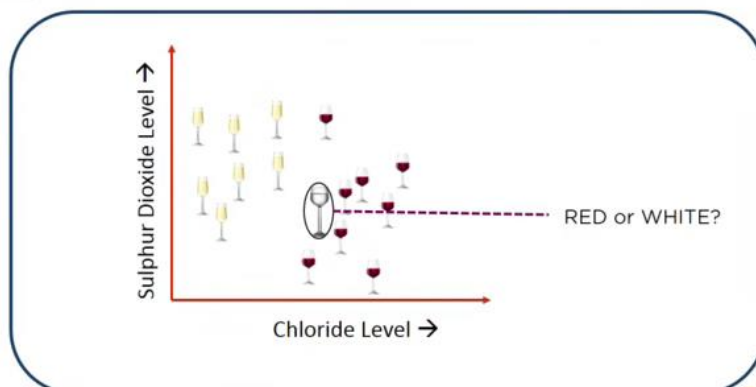
Classification



It classifies a data point based on how its neighbors are classified

What is KNN Algorithm?

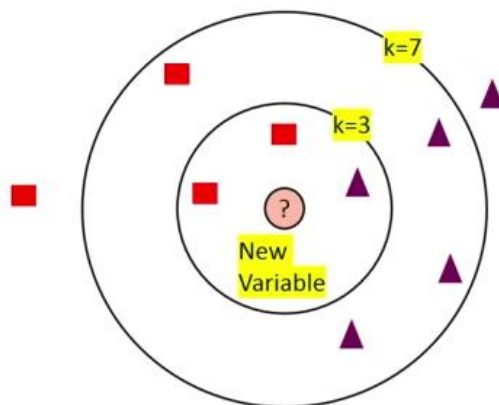
KNN stores all available cases and classifies new cases based on a similarity measure



k in KNN is a parameter that refers to the number of nearest neighbors to include in the majority voting process

How do we choose the factor 'k'?

KNN Algorithm is based on feature similarity: Choosing the right value of k is a process called parameter tuning, and is important for better accuracy



But at $k=7$, we classify '?' as



si

How do we choose the factor 'k'?


To choose a value of k :



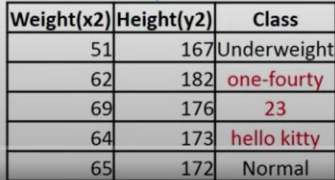
\sqrt{n} , where n is the total number of data points

Odd value of K is selected to avoid confusion between two classes of data

When do we use KNN Algorithm?

Watch later Share

 We can use KNN when

-  Data is labeled
Dog
-  Dataset is small
-  Data is noise free

Because KNN is a 'lazy learner' i.e. doesn't learn a discriminative function from the training set

Weight(x2)	Height(y2)	Class
51	167	Underweight
62	182	one-fourty
69	176	23
64	173	hello kitty
65	172	Normal

Noise

6:39 / 27:43 • When do we use KNN?

simplele YouTube

How does KNN Algorithm work?

Watch later Share



Consider a dataset having two variables: height (cm) & weight (kg) and each point is classified as Normal or Underweight

Weight(x2)	Height(y2)	Class
51	167	Underweight
62	182	Normal
69	176	Normal
64	173	Normal
65	172	Normal
56	174	Underweight
58	169	Normal
57	173	Normal
55	170	Normal

How does KNN Algorithm work?



On the basis of the given data we have to classify the below set as Normal or Underweight using KNN

57 kg	170 cm	?
-------	--------	---



Assuming, we don't know how to calculate BMI!

To find the nearest neighbors, we will calculate Euclidean distance

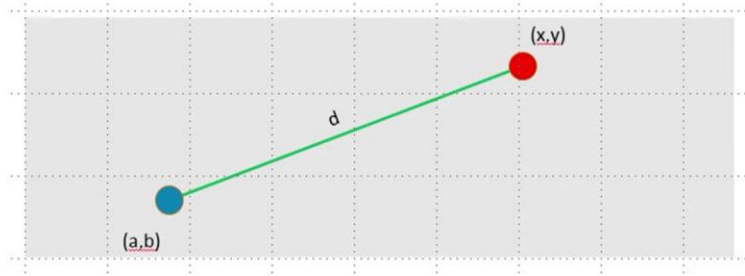


But, what is Euclidean distance?

How does KNN Algorithm work?

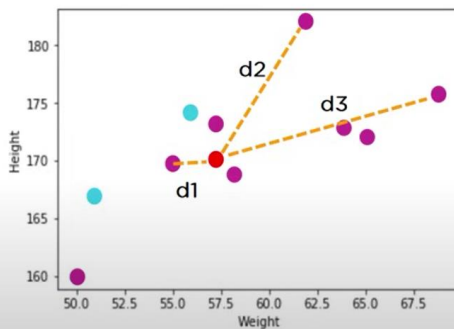
According to the Euclidean distance formula, the distance between two points in the plane with coordinates (x, y) and (a, b) is given by:

$$\text{dist}(d) = \sqrt{(x - a)^2 + (y - b)^2}$$



simplilearn

Let's calculate it to understand clearly:



● Unknown data point

$$\text{dist}(d1) = \sqrt{(170-167)^2 + (57-51)^2} \approx 6.7$$

$$\text{dist}(d2) = \sqrt{(170-182)^2 + (57-62)^2} \approx 13$$

$$\text{dist}(d3) = \sqrt{(170-176)^2 + (57-69)^2} \approx 13.4$$

Similarly, we will calculate Euclidean distance of unknown data point from all the points in the dataset

Hence, we have calculated the Euclidean distance of unknown data point from all the points as shown:

Where (x1, y1) = (57, 170) whose class we have to classify

Weight(x2)	Height(y2)	Class	Euclidean Distance
51	167	Underweight	6.7
62	182	Normal	13
69	176	Normal	13.4
64	173	Normal	7.6
65	172	Normal	8.2
56	174	Underweight	4.1
58	169	Normal	1.4
57	173	Normal	3
55	170	Normal	2

simplilearn

How does KNN Algorithm work?

Now, let's calculate the nearest neighbor at $k=3$

Weight(x2)	Height(y2)	Class	Euclidean Distance
51	167	Underweight	6.7
62	182	Normal	13
69	176	Normal	13.4
64	173	Normal	7.6
65	172	Normal	8.2
56	174	Underweight	4.1
58	169	Normal	1.4
57	173	Normal	3
55	170	Normal	2

$k = 3$

57 kg	170 cm	?
-------	--------	---



Class	Euclidean Distance
Underweight	6.7
Normal	13
Normal	13.4
Normal	7.6
Normal	8.2
Underweight	4.1
Normal	1.4
Normal	3
Normal	2

$k = 3$

So, majority neighbors are pointing towards 'Normal'

Hence, as per KNN algorithm the class of (57, 170) should be 'Normal'



Recap of KNN

- A positive integer k is specified, along with a new sample
- We select the k entries in our database which are closest to the new sample
- We find the most common classification of these entries
- This is the classification we give to the new sample

Lazy Learners k-Nearest-Neighbor Classifiers:

Lazy learning is a machine learning method that **doesn't process training data until it needs to make a prediction. It's also called instance-based or memory-based learning. In lazy learning, algorithms wait until they encounter a new query, and then store and compare training examples when making predictions.** The model memorizes the entire training dataset and uses it as the knowledge source for making predictions on new, unseen instances. The model looks for similar instances in the training data and applies their labels to the new instance, making predictions based on the most similar examples.

Lazy learning is useful when working with large datasets that have a few attributes, or when working with datasets that are constantly updated with new entries. Some examples of lazy learning include: Instance-based learning, Local regression, K-Nearest Neighbors (K-NN), and Lazy Bayesian Rules.

K-Nearest Neighbors (K-NN):

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

- K-NN algorithm assumes the **similarity between the new case/data and available cases** and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for **Regression as well as for Classification** but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase **just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.**