UNIT – 5 : DATA PRE-PROCESSING AND FEATURE SELECTION

Data Pre-Processing:

Real-world datasets are generally messy, raw, incomplete, inconsistent, and unusable. It can contain manual entry errors, missing values, inconsistent schema, etc.

Data Preprocessing is the process of converting raw data into a format that is understandable and usable. It is a crucial step in any Data Science project to carry out an efficient and accurate analysis. It ensures that data quality is consistent before applying any Machine Learning or Data Mining techniques.

Why is Data Preprocessing Important?

Data Preprocessing is an important step in the Data Preparation stage of a Data Science development lifecycle that will ensure reliable, robust, and consistent results. The main objective of this step is to ensure and check the quality of data before applying any Machine Learning or Data Mining methods. Let's review some of its benefits -

- **Accuracy** - Data Preprocessing will ensure that input data is accurate and reliable by ensuring there are no manual entry errors, no duplicates, etc.
- **Completeness** - It ensures that missing values are handled, and data is complete for further analysis.
- **Consistent** - Data Preprocessing ensures that input data is consistent, i.e., the same data kept in different places should match.
- **Timeliness** - Whether data is updated regularly and on a timely basis or not.
- **Trustable** - Whether data is coming from trustworthy sources or not.
- **Interpretability** - Raw data is generally unusable, and Data Preprocessing converts raw data into an interpretable format.

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.

Applications of Data Preprocessing:

Data Preprocessing is important in the early stages of a Machine Learning and AI application development lifecycle. A few of the most common usage or application include -

- **Improved Accuracy of ML Models** - Various techniques used to preprocess data, such as Data Cleaning, Transformation ensure that data is complete, accurate, and understandable, resulting in efficient and accurate ML models.
- **Reduced Costs** - Data Reduction techniques can help companies save storage and compute costs by reducing the volume of the data
- **Visualization** - Preprocessed data is easily consumable and understandable that can be further used to build dashboards to gain valuable insights.

Data Cleaning:

Data cleaning, also known as **data cleansing** or **data scrubbing**, is a crucial step in the data science pipeline that involves identifying and correcting or removing errors, inconsistencies, and inaccuracies in the data to improve its quality and usability. Data cleaning is essential because raw data is often noisy, incomplete, and inconsistent, which can negatively impact the accuracy and reliability of the insights derived from it.

Why Is Data Cleaning So Important?

The important thing about the data cleaning process is that data accuracy and reliability will be at the center of the process of the information used for analysis. Let me explain that with a cooking example, you cannot feed the wrong ingredients to the recipe – the dish will be a mess. In data, we have to credit the "garbage in, garbage out" rule. Here's why cleaning data is so important:

- **Better Decisions:** Dirty data generates lying output, or disinformation. With accurate data and clean data, your analysis connected to reality, and guides you for good options.
- **Saved Time and Money:** Incorrect data can make the right decision making very difficult, cause wasted efforts that might be directed towards unsuitable and wrong leads and solutions. Clean data saves the time and expense of redesigning processes that crashed due to a dirty data issue.
- **Improved Efficiency:** When data stays clean, the functioning of the whole system becomes easier. Dirty data leads to friction and inefficiency. The duplication of efforts to obtain reliable information will only add to the losses.

Data Cleaning Techniques:

Remove Duplicates: It is likely that you will have duplicate entries if you scrape your data or get it from a variety of sources. These duplication may result from human error on the part of the individual entering the data or completing a form.

Detect and Remove Outliers: Outliers are data points that fall significantly outside the expected range for a particular variable. They can be caused by errors in data collection or measurement, or they may represent genuine but unusual cases. Leaving outliers in your data set can skew your analysis and lead to misleading results.

There are a number of statistical methods for detecting outliers, and the best approach will depend on the specific nature of your data. Once outliers have been identified, you can decide whether to remove them from your data set or to investigate them further.

Remove Irrelevant Data: Any analysis you wish to perform will be slowed down and confused by irrelevant data. Thus, before you start cleaning your data, you must determine what is and is not significant. For example, you do not need to provide your customers' email addresses if you are studying the range of ages of your consumers.

Standardize Capitalization: You must ensure that the text in your data is consistent. Different incorrect categories may be formed if your capitalization is inconsistent.

Since capitalization can alter meaning, it could also be problematic if you had to translate something before processing. For example, a bill or to bill is something else entirely, yet Bill is a person's name.

Convert Data Types: When cleaning your data, numbers are the most frequent data type that needs to be converted. Numbers are frequently imputed as text, but they must appear as digits in order to be processed.

They are categorized as strings and cannot be used by your analytical algorithms to solve mathematical equations if they are shown as text.

Clear Formatting: Your input cannot be processed by machine learning models if it is highly structured. There probably are a variety of document formats if you are gathering data from several sources. Your data may become erroneous and unclear as a result.

To start from scratch, you should erase any formatting that has been applied to your papers. Usually, this is not a tough task to do; for instance, there is a straightforward standardization feature in both Google Sheets and Excel.

Fix Errors: It should go without saying that you must take great care to eliminate any inaccuracies from your data. Typographical errors are just as prone to error and might cause you to overlook important insights from your data. Something as easy as a fast spell check can help prevent some of them.

Errors in spelling or excessive punctuation in data, such as an email address, may prevent you from reaching out to customers. Additionally, you can end yourself sending unsolicited emails to recipients who never requested them.

Language Translation: You will want everything in the same language if you want consistent data. The majority of Natural Language Processing (NLP) models that underpin data analysis tools are monolingual, which means they cannot process more than one language. Thus, everything will have to be translated into a single language.

Handle Missing Values: Eliminating the absent value entirely could lead to the loss of valuable information from your data. You intended to extract this information in the first place for a reason, after all.

Thus, it could be preferable to fill in the blanks by looking up the appropriate information for that field. You might use the word missing in its place if you're not sure what it is. You can enter a zero in the blank box if it is numerical.

Characteristics of Data Cleaning:

To ensure the correctness, integrity, and security of corporate data, data cleaning is a requirement. These may be of varying quality depending on the properties or attributes of the data. The key components of data cleansing in data mining are as follows:

- **Accuracy:** The business's database must contain only extremely accurate data. Comparing them to other sources is one technique to confirm their veracity. The stored data will also have issues if the source cannot be located or contains errors.
- **Coherence:** To ensure that the information on a person or body is the same throughout all types of storage, the data must be consistent with one another.
- **Validity:** There must be rules or limitations in place for the stored data. The information must also be confirmed to support its veracity.
- **Uniformity:** A database's data must all share the same units or values. Since it doesn't complicate the process, it is a crucial component while doing the Data Cleansing process.
- **Data Verification:** Every step of the process, including its appropriateness and effectiveness, must be checked. The study, design, and validation stages all play a role in the verification process. The disadvantages are frequently obvious after applying the data to a specific number of changes.
- **Clean Data Backflow:** After addressing quality issues, the previously clean data must be replaced with data that is not present in the source so that legacy applications can profit from it and avoid the need for a subsequent data-cleaning program.

Tools for Data Cleaning:

Data Cleansing Tools can be very helpful if you are not confident of cleaning the data yourself or have no time to clean up all your data sets. You might need to invest in those tools, but it is worth the expenditure.

There are many data cleaning tools in the market. Here are some top-ranked data cleaning tools, such as:

1. OpenRefine
2. Trifacta Wrangler
3. Drake
4. Data Ladder
5. Data Cleaner
6. Cloudingo
7. Reifier
8. IBM Infosphere Quality Stage
9. TIBCO Clarity
10. Winpure

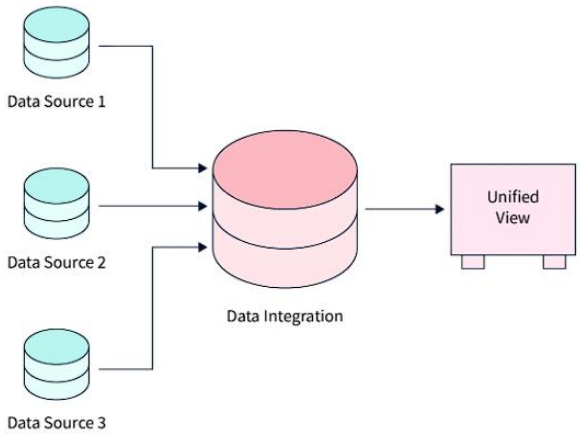
Data Integration:

Data integration is the process of combining data from multiple sources and consolidating it into a unified view. It is a critical aspect of data mining, which involves discovering patterns and insights from large datasets. But what is data integration, exactly? Simply put, it is the process of transforming and merging data from disparate sources so that it can be analyzed together.

The goal of data integration in data mining is to provide a complete and accurate representation of the data for further analysis. It involves extracting data from various sources, transforming it into a common format, and loading it into a target system. Data integration can be challenging, especially when dealing with large volumes of data, complex data structures, and different data formats.

Data integration strategy is typically described using a triple (**G, S, M**) approach, where **G** denotes the global schema, **S** denotes the schema of the heterogeneous data sources, and **M** represents the mapping between the queries of the source and global schema.

To understand the (**G, S, M**) approach, Let's consider a data integration scenario that aims to combine employee data from two different HR databases, database A and database B. The global schema (G) would define the unified view of employee data, including attributes like EmployeeID, Name, Department, and Salary. In the schema of heterogeneous sources, database A (S1) might have attributes like EmpID, FullName, Dept, and Pay, while database B's schema (S2) might have attributes like ID, EmployeeName, DepartmentName, and Compensation. The mappings (M) would then define how the attributes in S1 and S2 map to the attributes in G, allowing for the integration of employee data from both systems into the global schema.



Why is Data Integration Important?

Data integration in data mining is important for several reasons, as mentioned below -

- **Provides a Unified View of Data** - Data integration in data mining enables combining data from different sources into a unified view. This allows for better decision-making by providing a complete and accurate data representation.
- **Increases Data Accuracy** - Integrating data from multiple sources helps to identify and eliminate inconsistencies, redundancies, and errors in the data. This improves data accuracy and reliability, making it easier to draw accurate conclusions.
- **Improves Efficiency** - Data integration in data mining automates combining data from multiple sources, reducing the time and effort required to access and analyze the data. This improves efficiency and reduces the costs associated with data management.
- **Facilitates Data Analysis** - Integrating data from multiple sources provides a broader perspective of the data. This enables more sophisticated and accurate data analysis, leading to more informed and effective decision-making.
- **Enables Business Intelligence** - Data integration is essential for creating a reliable and accurate data warehouse supporting business intelligence initiatives.

Approaches for Data Integration:

There are mainly two kinds of approaches to data integration in data mining, as mentioned below -

Tight Coupling:

- This approach involves the creation of a centralized database that integrates data from different sources. The data is loaded into the centralized database using extract, transform, and load (ETL) processes.

- In this approach, the integration is tightly coupled, meaning that the data is physically stored in the central database, and any updates or changes made to the data sources are immediately reflected in the central database.
- Tight coupling is suitable for situations where real-time access to the data is required, and data consistency is critical. However, this approach can be costly and complex, especially when dealing with large volumes of data.

Loose Coupling:

- This approach involves the integration of data from different sources without physically storing it in a centralized database.
- In this approach, data is accessed from the source systems as needed and combined in real-time to provide a unified view. This approach uses middleware, such as application programming interfaces (APIs) and web services, to connect the source systems and access the data.
- Loose coupling is suitable for situations where real-time access to the data is not critical, and the data sources are highly distributed. This approach is more cost-effective and flexible than tight coupling but can be more complex to set up and maintain.

Issues in Data Integration:

- **Data Quality** - The quality of the data being integrated can be a significant issue in data integration. Data from different sources may have varying levels of accuracy, completeness, and consistency, which can lead to data quality issues in the integrated data.
- **Data Semantics** - Data semantics refers to the meaning and interpretation of data. Integrating data from different sources can be challenging because the same data element may have different meanings across sources. This can result in data integration issues and impact the integrated data's accuracy.
- **Data Heterogeneity** - Data heterogeneity refers to the differences in data formats, structures, and storage mechanisms across different data sources. Data integration can be challenging when dealing with heterogeneous data sources, as it requires data transformation and mapping to make the data compatible with the target data model.
- **Complexity** - Data integration can be complex, especially when dealing with large volumes of data or multiple data sources. As the complexity of data integration increases, it becomes more challenging to maintain data quality, ensure data consistency, and manage data security and privacy.

- **Data Privacy and Security** - Data integration can increase the risk of data privacy and security breaches. Integrating data from multiple sources can expose sensitive information and increase the risk of unauthorized access or disclosure.
- **Scalability** - Scalability refers to the ability of the data integration solution to handle increasing volumes of data and accommodate changes in data sources. Data integration solutions must be scalable to meet the organization's evolving needs and ensure that the integrated data remains accurate and consistent.

Data Integration Tools:

There are mainly three types of tools for data integration in data mining, as mentioned below -

- **On-Premise Data Integration Tools** - On-premise data integration tools are installed and run on the organization's infrastructure. These tools offer complete control over the data integration process and are typically used by larger organizations requiring high customization and security levels. Some popular on-premise data integration tools include IBM InfoSphere DataStage, Talend, and Microsoft SQL Server Integration Services (SSIS).
- **Open-Source Data Integration Tools** - Open-source data integration tools are free and often community-driven solutions that allow users to modify the source code and add new features. These tools are typically less expensive than proprietary tools and can be customized to fit the organization's specific needs. Some popular open-source data integration tools include Apache NiFi, Apache Kafka, and Pentaho.
- **Cloud-Based Data Integration Tools** - Cloud-based data integration tools are hosted in the cloud and accessed through a web browser. These tools offer scalability, flexibility, and easy access to data from different sources. They are ideal for organizations requiring quick implementation and not wanting to invest in on-premise hardware or software. Some popular cloud-based data integration tools include Amazon Web Services (AWS) Glue, Microsoft Azure Data Factory, and Google Cloud Data Fusion. However, privacy and security are major concerns when using cloud-based data integration tools. Storing and transferring sensitive data to the cloud can expose it to potential risks, such as unauthorized access, data breaches, or data leakage. Adequate security measures, including encryption, access controls, and secure authentication, must be implemented to protect data during storage and transmission.

Data Reduction:

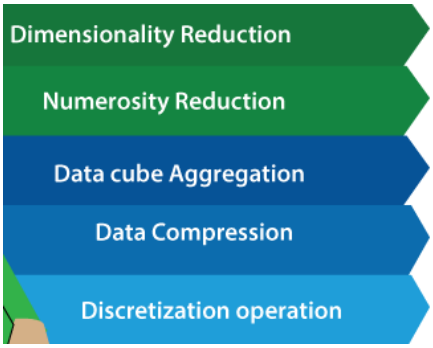
Data reduction techniques ensure the integrity of data while reducing the data. Data reduction is a process that reduces the volume of original data and represents it in a much smaller volume. Data reduction techniques are used to obtain a reduced representation of the dataset that is much smaller in volume by maintaining the integrity of the original data. By reducing the data, the efficiency of the data mining process is improved, which produces the same analytical results.

Data reduction does not affect the result obtained from data mining. That means the result obtained from data mining before and after data reduction is the same or almost the same.

Data reduction aims to define it more compactly. When the data size is smaller, it is simpler to apply sophisticated and computationally high-priced algorithms. The reduction of the data may be in terms of the number of rows (records) or terms of the number of columns (dimensions).

Techniques of Data Reduction:

Here are the following techniques or methods of data reduction in data mining, such as:



1. Dimensionality Reduction:

Whenever we encounter weakly important data, we use the attribute required for our analysis. Dimensionality reduction eliminates the attributes from the data set under consideration, thereby reducing the volume of original data. It reduces data size as it eliminates outdated or redundant features. Here are three methods of dimensionality reduction.

i. **Wavelet Transform:** In the wavelet transform, suppose a data vector A is transformed into a numerically different data vector A' such that both A and A' vectors are of the same length. Then how it is useful in reducing data because the data obtained from the wavelet transform can be truncated. The compressed data is obtained by retaining the smallest fragment of the strongest wavelet coefficients. Wavelet transform can be applied to data cubes, sparse data, or skewed data.

ii. **Principal Component Analysis:** Suppose we have a data set to be analyzed that has tuples with n attributes. The principal component analysis identifies k independent tuples with n attributes that can represent the data set. In this way, the original data can be cast on a much smaller space, and dimensionality reduction can be achieved. Principal component analysis can be applied to sparse and skewed data.

iii. **Attribute Subset Selection:** The large data set has many attributes, some of which are irrelevant to data mining or some are redundant. The core attribute subset selection reduces the data volume and dimensionality. The attribute subset selection reduces the volume of data by eliminating redundant and irrelevant attributes.
The attribute subset selection ensures that we get a good subset of original attributes even after eliminating the unwanted attributes. The resulting probability of data distribution is as close as possible to the original data distribution using all the attributes.

2. sNumerosity Reduction:

The numerosity reduction reduces the original data volume and represents it in a much smaller form. This technique includes two types parametric and non-parametric numerosity reduction.

i. **Parametric:** Parametric numerosity reduction incorporates storing only data parameters instead of the original data. One method of parametric numerosity reduction is the regression and log-linear method.

- **Regression and Log-Linear:** Linear regression models a relationship between the two attributes by modeling a linear equation to the data set. Suppose we need to model a linear function between two attributes.
  $y = wx + b$

  Here, y is the response attribute, and x is the predictor attribute. If we discuss in terms of data mining, attribute x and attribute y are the numeric database attributes, whereas w and b are regression coefficients.

  Multiple linear regressions let the response variable y model linear function between two or more predictor variables.

  Log-linear model discovers the relation between two or more discrete attributes in the database. Suppose we have a set of tuples presented in n-dimensional space. Then the log-linear model is used to study the probability of each tuple in a multidimensional space.

  Regression and log-linear methods can be used for sparse data and skewed data.

ii. **Non-Parametric:** A non-parametric numerosity reduction technique does not assume any model. The non-Parametric technique results in a more uniform reduction, irrespective of data size, but it may not achieve a high volume of data reduction like the parametric. There are at least four types of Non-Parametric data reduction techniques, Histogram, Clustering, Sampling, Data Cube Aggregation, and Data Compression.

- **Histogram:** A histogram is a graph that represents frequency distribution which describes how often a value appears in the data. Histogram uses the binning method to represent an attribute's data distribution. It uses a disjoint subset which we call bin or buckets.

  A histogram can represent a dense, sparse, uniform, or skewed data. Instead of only one attribute, the histogram can be implemented for multiple attributes. It can effectively represent up to five attributes.

- **Clustering:** Clustering techniques groups similar objects from the data so that the objects in a cluster are similar to each other, but they are dissimilar to objects in another cluster.

  How much similar are the objects inside a cluster can be calculated using a distance function. More is the similarity between the objects in a cluster closer they appear in the cluster.

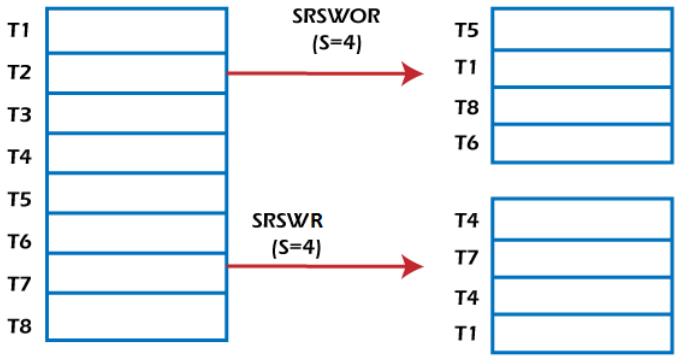
  The quality of the cluster depends on the diameter of the cluster, i.e., the max distance between any two objects in the cluster.

  The cluster representation replaces the original data. This technique is more effective if the present data can be classified into a distinct clustered.

- **Sampling:** One of the methods used for data reduction is sampling, as it can reduce the large data set into a much smaller data sample. Below we will discuss the different methods in which we can sample a large data set D containing N tuples:

a. **Simple random sample without replacement (SRSWOR) of size s:** In this s, some tuples are drawn from N tuples such that in the data set D (s<N). The probability of drawing any tuple from the data set D is 1/N. This means all tuples have an equal probability of getting sampled.

b. **Simple random sample with replacement (SRSWR) of size s:** It is similar to the SRSWOR, but the tuple is drawn from data set D, is recorded, and then replaced into the data set D so that it can be drawn again.



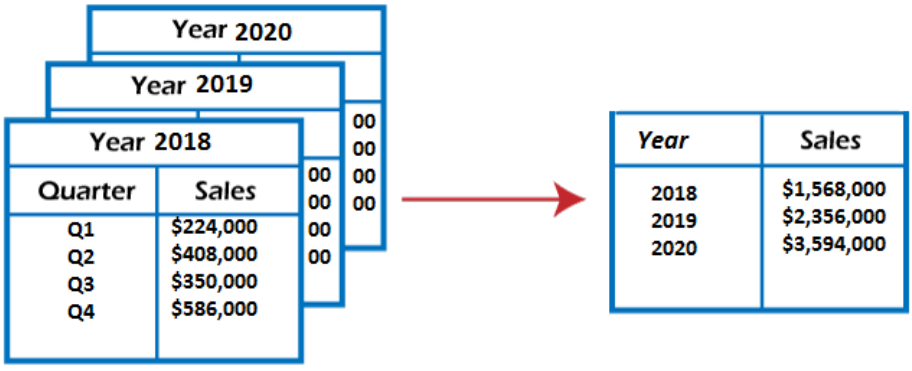
c. **Cluster sample:** The tuples in data set D are clustered into M mutually disjoint subsets. The data reduction can be applied by implementing SRSWOR on these clusters. A simple random sample of size s could be generated from these clusters where s<M.

d. **Stratified sample:** The large data set D is partitioned into mutually disjoint sets called 'strata'. A simple random sample is taken from each stratum to get stratified data. This method is effective for skewed data.

3. Data Cube Aggregation:

This technique is used to aggregate data in a simpler form. Data Cube Aggregation is a multidimensional aggregation that uses aggregation at various levels of a data cube to represent the original data set, thus achieving data reduction.

For example, suppose you have the data of All Electronics sales per quarter for the year 2018 to the year 2022. If you want to get the annual sale per year, you just have to aggregate the sales per quarter for each year. In this way, aggregation provides you with the required data, which is much smaller in size, and thereby we achieve data reduction even without losing any data.

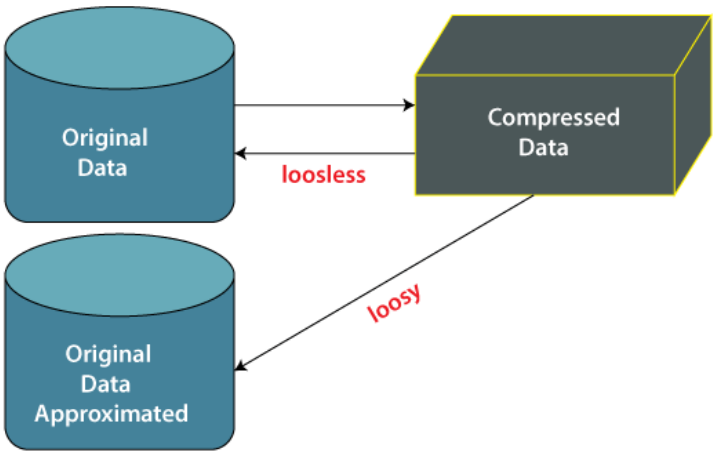


Aggregated Data

The data cube aggregation is a multidimensional aggregation that eases multidimensional analysis. The data cube present precomputed and summarized data which eases the data mining into fast access.

4. Data Compression:

Data compression employs modification, encoding, or converting the structure of data in a way that consumes less space. Data compression involves building a compact representation of information by removing redundancy and representing data in binary form. Data that can be restored successfully from its compressed form is called Lossless compression. In contrast, the opposite where it is not possible to restore the original form from the compressed form is Lossy compression. Dimensionality and numerosity reduction method are also used for data compression.



This technique reduces the size of the files using different encoding mechanisms, such as Huffman Encoding and run-length Encoding. We can divide it into two types based on their compression techniques.

i. **Lossless Compression:** Encoding techniques (Run Length Encoding) allow a simple and minimal data size reduction. Lossless data compression uses algorithms to restore the precise original data from the compressed data.

ii. **Lossy Compression:** In lossy-data compression, the decompressed data may differ from the original data but are useful enough to retrieve information from them. For example, the JPEG image format is a lossy compression, but we can find the meaning equivalent to the original image. Methods such as the Discrete Wavelet transform technique PCA (principal component analysis) are examples of this compression.

5. Discretization Operation

The data discretization technique is used to divide the attributes of the continuous nature into data with intervals. We replace many constant values of the attributes with labels of small intervals. This means that mining results are shown in a concise and easily understandable way.

i. **Top-down discretization:** If you first consider one or a couple of points (so-called breakpoints or split points) to divide the whole set of attributes and repeat this method up to the end, then the process is known as top-down discretization, also known as splitting.

ii. **Bottom-up discretization:** If you first consider all the constant values as split-points, some are discarded through a combination of the neighborhood values in the interval. That process is called bottom-up discretization.

Benefits of Data Reduction:

The main benefit of data reduction is simple: the more data you can fit into a terabyte of disk space, the less capacity you will need to purchase. Here are some benefits of data reduction, such as:

- Data reduction can save energy.
- Data reduction can reduce your physical storage costs.
- And data reduction can decrease your data center track.

Data reduction greatly increases the efficiency of a storage system and directly impacts your total spending on capacity.

Data Transformation:

Data Transformation is a technique used to transform raw data into a more appropriate format that enables efficient data mining and model building.

When to Transform Data?

- **Data Transformation** is an essential technique that must be used before Data Mining so that it can help in extracting meaningful patterns and insights.
- It is also essential to perform before training and developing an ML model. While training an ML model, both datasets used in the training and testing phase of the model need to be transformed in the same way.

Benefits and Challenges of Data Transformation:

A few of the benefits companies get by using Data Transformation include as following:

- **Maximize Value of Data:** Data Transformation standardizes data from various data sources to increase its usability and accessibility. This will ensure that maximum data is used in **Data Mining** and **model building**, resulting in extracting maximum value from data.

- **Effective Data Management:** Data Transformation helps remove inconsistencies in the data by applying various techniques so that it is easier to understand and retrieve data.
- **Better Model Building and Insights:** Typically, the distribution of features in a dataset is highly skewed. So, Data Transformation helps remove bias in the model by standardizing and normalizing features in the same range.
- **Improve Data Quality:** Data Transformation helps organizations improve data quality by handling missing values and other inconsistencies.

Data Transformation comes with its own challenges as well. Let's have a look at some of the challenges of the Data Transformation process.

- Data Transformation is an **expensive** and **resource-intensive** process. This cost depends upon many factors such as infrastructure, tools, company requirements, data size, etc.
- Data Transformation requires professionals with appropriate subject matter expertise as faulty Data Transformation can lead to inaccurate business insights.

Data Transformation Techniques:

A few of the most common Data Transformation techniques include as following:

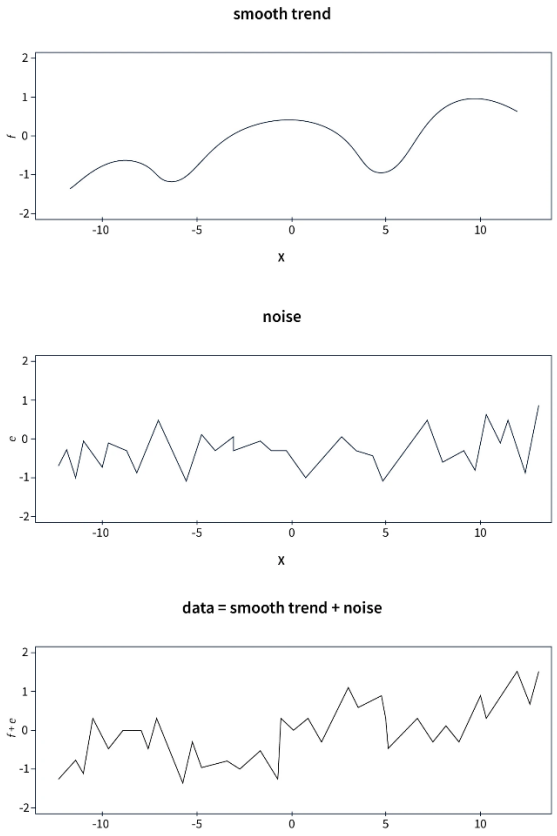
1. Data Smoothing:

Data Smoothing is used to remove noise in the data, and it helps inherent patterns to stand out. Therefore, Data Smoothing can help in predicting trends or future events.

For example, as shown in the below diagram, smoothing allows us to remove noise from the input data that helps identify implicit seasonality and growth trends.

Some of the ways to perform Data Smoothing are moving average, exponential average, random walk, regression, binning, etc.

- **Binning:** This method splits the sorted data into the number of bins and smoothens the data values in each bin considering the neighborhood values around it.
- **Regression:** This method identifies the relation among two dependent attributes so that if we have one attribute, it can be used to predict the other attribute.
- **Clustering:** This method groups similar data values and form a cluster. The values that lie outside a cluster are known as outliers.

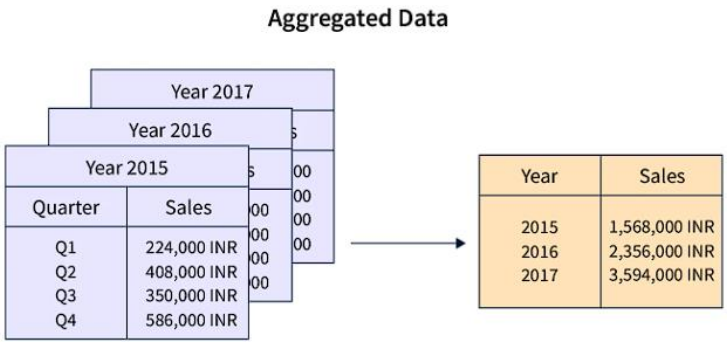


2. Attribute Construction:

In this method, new attributes or features are created out of the existing features. It simplifies the data and makes data mining more efficient. For example, if we have height and weight features in the data, we can create a new attribute, BMI, using these two features.

3. Data Aggregation:

Data Aggregation is the process of compiling large volumes of data and transforming it into an organized and summarized format that is more consumable and comprehensive. Data Aggregation can enable the capability to forecast future trends and aid in predictive analysis. For example, a company may look at monthly sales data of a product instead of raw sales data to understand its performance better and forecast future sales.



4. Data Normalization:

The range of values for each attribute in a dataset can vary greatly. Some features might contain large numbers, such as sales data, etc., while others might have comparatively smaller matters, such as age, etc. This could introduce a bias in the model building. Therefore, it is essential to normalize every feature in the dataset. **Data Normalization** is a technique that is used to convert a numeric variable into a specified range such as [-1,1], [0,1], etc. A few of the most common approaches to performing normalization include:

- **Min-Max Normalization:** This is a linear transformation and will convert the data into the [0,1] range. The formula for Min-Max Normalization is:

$$v_i' = \frac{v_i - \min_A}{\max_A - \min_A}(new_{\max_A} - new_{\min_A}) + new\_\min_A$$

- **Z-Score Normalization:** It utilizes the mean and standard deviation of the attribute to normalize it. It will ensure that the attribute has a 0 mean and 1 standard deviation. **Z-Score Normalization** is also called **Data Standardization** or **Data Scaling**. The below formula is used to perform Z-Score Normalization:

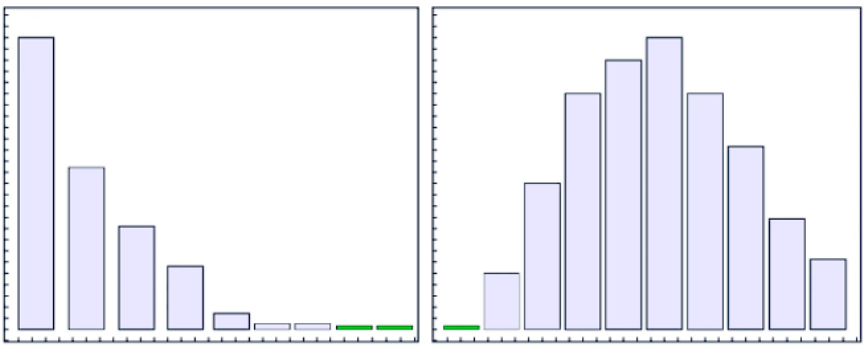
$$v_i' = \frac{v_i - \bar{A}}{\sigma_A}$$

5. Data Discretization:

It is a process of converting numerical or continuous variables into a set of intervals. This makes data easy to analyze and understand. For example, the age features can be converted into intervals such as (0-10, 11-20, ..) or (child, young, …).

6. Log Transformation:

When input data does not conform to the normal distribution and has a skewed distribution, then Log transformation is used to transform/convert it into a normal distribution.



7. Data Generalization

It converts low-level data attributes to high-level data attributes using concept hierarchy. This conversion from a lower level to a higher conceptual level is useful to get a clearer picture of the data. Data generalization can be divided into two approaches:

- Data cube process (OLAP) approach.
- Attribute-oriented induction (AOI) approach.

For example, age data can be in the form of (20, 30) in a dataset. It is transformed into a higher conceptual level into a categorical value (young, old).

Data Transformation Process:

1. **Data Discovery**: During the first stage, analysts work to understand and identify data in its source format. To do this, they will use data profiling tools. This step helps analysts decide what they need to do to get data into its desired format.
2. **Data Mapping:** During this phase, analysts perform data mapping to determine how individual fields are modified, mapped, filtered, joined, and aggregated. Data mapping is essential to many data processes, and one misstep can lead to incorrect analysis and ripple through your entire organization.
3. **Data Extraction:** During this phase, analysts extract the data from its original source. These may include structured sources such as databases or streaming sources such as customer log files from web applications.
4. **Code Generation and Execution:** Once the data has been extracted, analysts need to create a code to complete the transformation. Often, analysts generate codes with the help of data transformation platforms or tools.
5. **Review:** After transforming the data, analysts need to check it to ensure everything has been formatted correctly.
6. **Sending:** The final step involves sending the data to its target destination. The target might be a data warehouse or a database that handles both structured and unstructured data.

Ways of Data Transformation:

- **Scripting:** Data transformation through scripting involves ***Python*** **or** ***SQL*** to write the code to extract and transform data. Python and SQL are scripting languages that allow you to automate certain tasks in a program. They also allow you to extract information from data sets. Scripting languages require less code than traditional programming languages. Therefore, it is less intensive.

- **On-Premises ETL Tools:** ETL tools take the required work to script the data transformation by automating the process. On-premises ETL tools are hosted on company servers. While these tools can help save you time, using them often **requires extensive expertise and significant infrastructure costs**.
- **Cloud-Based ETL Tools:** As the name suggests, cloud-based ETL tools are **hosted in the cloud**. These tools are often the easiest for non-technical users to utilize. They allow you to collect data from any cloud source and load it into your data warehouse. With cloud-based ETL tools, you can decide how often you want to pull data from your source, **and you can monitor your usage**.

Data Discretization:

Discretization in data mining refers to converting a range of continuous values into discrete categories. In simpler terms, it's like grouping ages into categories like 'child,' 'teen,' 'adult', and 'senior' instead of dealing with each age individually. This method is particularly useful in data mining because it can help uncover patterns and relationships in the data that are not immediately apparent when dealing with continuous values.

For instance, imagine a dataset of patient blood pressure readings. Discretization would involve categorizing these readings into 'low', 'normal', and 'high' groups. This categorization makes it easier to analyze the data, especially when working with machine learning algorithms, which often perform better with discrete input.

The primary goal of discretization is to reduce the complexity of continuous data, making it more digestible for analysis. It simplifies the data without losing its intrinsic value, balancing detail and usability.

Some Famous Techniques of Data Discretization:

Data discretization encompasses various techniques, each with its unique approach and application. Understanding these methods is crucial for students venturing into data mining. Let's delve into some of the most well-known discretization techniques:

Equal-Width Intervals: This technique divides the range of attribute values into intervals of equal size. The simplicity of this method makes it popular, especially for initial data analysis. For example, if you're dealing with an attribute like height, you might divide it into intervals of 10 cm each.

Equal-Frequency Intervals: Unlike equal-width intervals, this method divides the data so that each interval contains approximately the same number of data points. It's particularly useful when the data is unevenly distributed, as it ensures that each category has a representative sample.

Cluster Analysis: This more complex technique uses clustering algorithms to group data points based on similarity. The clusters formed represent the intervals. It's a powerful method for discovering natural groupings in the data.

Decision Tree Based: Here, decision trees are used for discretization. The tree splits the continuous attribute at various points, and the resulting tree structure helps determine the intervals. This method is beneficial when the discretization needs to align closely with the predictive modeling goals.

Entropy-Based Discretization: This technique uses information entropy to find the optimal data partitioning. It effectively finds boundaries that provide the most informational gain, making it highly suitable for preparing data for information-theoretic models.

Importance of Discretization:

A discretization is important because it is useful:

1. To generate concept hierarchies.
2. Transform numeric data.
3. To ease evaluation and management of data.
4. To minimize data loss.
5. To produce a better result.
6. Generate a more understandable structure viz. decision tree.

Data Discretization & Concept Hierarchy Generation:

This aspect of data mining is about transforming raw data into more abstract layers of knowledge. Concept hierarchy generation is a process that builds upon discretization to further abstract the data. It's like creating a tree where leaves represent the most specific information, and branches represent more general concepts.

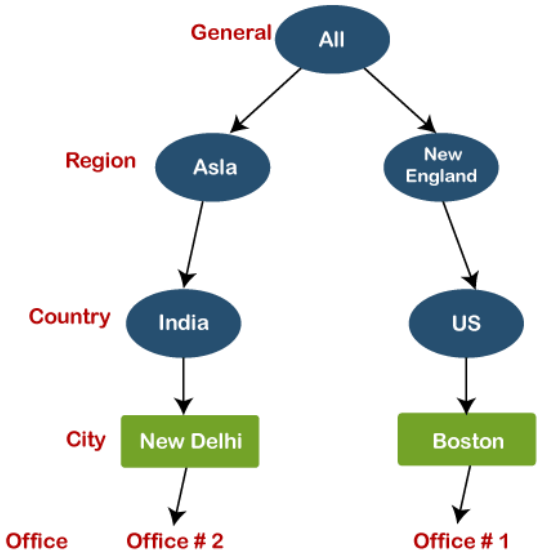
Let's understand this concept hierarchy for the dimension location with the help of an example.

A particular city can map with the belonging country. For example, New Delhi can be mapped to India, and India can be mapped to Asia.

Top-down mapping: Top-down mapping generally starts with the top with some general information and ends with the bottom to the specialized information.

Bottom-up mapping: Bottom-up mapping generally starts with the bottom with some specialized information and ends with the top to the generalized information.

The process involves:



Concept Hierarchy Generation

Data Discretization and Binarization:

Data discretization is a method of converting attributes values of continuous data into a finite set of intervals with minimum data loss. In contrast, data binarization is used to transform the continuous and discrete attributes into binary attributes.

Feature Generation:

Feature generation, also known as **feature engineering**, is the process of creating new features or variables from existing data to improve the performance of machine learning models. This process can involve creating new features based on domain knowledge, transforming existing features, or generating features from raw data. Effective feature generation can significantly enhance model accuracy and predictive power.

Need for Feature Generation:

- **Improve User Experience:** The primary reason we engineer features is to enhance the user experience of a product or service. By adding new features, we can make the product more intuitive, efficient, and user-friendly, which can increase user satisfaction and engagement.
- **Competitive Advantage:** Another reason we engineer features is to gain a competitive advantage in the marketplace. By offering unique and innovative features, we can differentiate our product from competitors and attract more customers.

- **Meet Customer Needs:** We engineer features to meet the evolving needs of customers. By analyzing user feedback, market trends, and customer behavior, we can identify areas where new features could enhance the product's value and meet customer needs.
- **Increase Revenue:** Features can also be engineered to generate more revenue. For example, a new feature that streamlines the checkout process can increase sales, or a feature that provides additional functionality could lead to more upsells or cross-sells.
- **Future-Proofing:** Engineering features can also be done to future-proof a product or service. By anticipating future trends and potential customer needs, we can develop features that ensure the product remains relevant and useful in the long term.

Processes Involved in Feature Generation:

Feature engineering in Machine learning consists of mainly 5 processes: Feature Creation, Feature Transformation, Feature Extraction, Feature Selection, and Feature Scaling. It is an iterative process that requires experimentation and testing to find the best combination of features for a given problem.

1. Feature Creation: Feature Creation is the process of generating new features based on domain knowledge or by observing patterns in the data. It is a form of feature engineering that can significantly improve the performance of a machine-learning model.

2. Feature Transformation: Feature Transformation is the process of transforming the features into a more suitable representation for the machine learning model. This is done to ensure that the model can effectively learn from the data.

3. Feature Extraction: Feature Extraction is the process of creating new features from existing ones to provide more relevant information to the machine learning model. This is done by transforming, combining, or aggregating existing features.

4. Feature Selection: Feature Selection is the process of selecting a subset of relevant features from the dataset to be used in a machine-learning model. It is an important step in the feature engineering process as it can have a significant impact on the model's performance.

5. Feature Scaling: Feature Scaling is the process of transforming the features so that they have a similar scale. This is important in machine learning because the scale of the features can affect the performance of the model.

Feature Selection:

Feature selection is a process that chooses a subset of features from the original features so that the feature space is optimally reduced according to a certain criterion.

Feature selection is a critical step in the feature construction process. In text categorization problems, some words simply do not appear very often. Perhaps the word "groovy" appears in exactly one training document, which is positive. Is it really worth keeping this word around as a feature? It's a dangerous endeavor because it's hard to tell with just one training example if it is really correlated with the positive class or is it just noise. You could hope that your learning algorithm is smart enough to figure it out. Or you could just remove it.

There are three general classes of feature selection algorithms: **Filter methods, wrapper methods and embedded methods**.

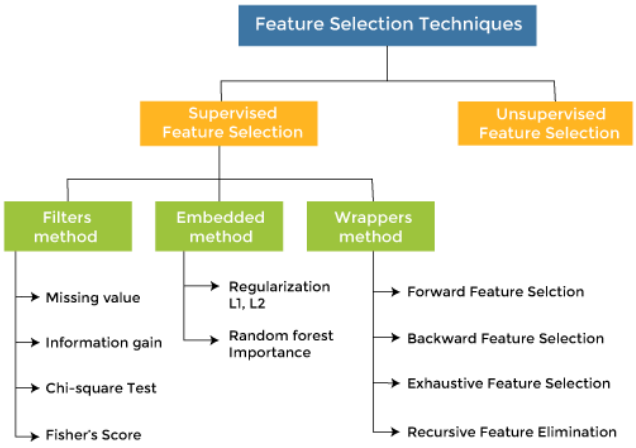
The role of feature selection in machine learning is,

1. To reduce the dimensionality of feature space.
2. To speed up a learning algorithm.
3. To improve the predictive accuracy of a classification algorithm.
4. To improve the comprehensibility of the learning results.

Feature Selection Techniques/ Algorithms:

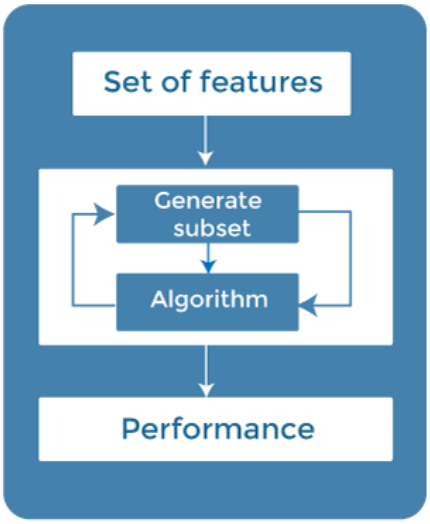
There are mainly two types of Feature Selection techniques, which are:

- **Supervised Feature Selection technique**: Supervised Feature selection techniques consider the target variable and can be used for the labelled dataset.
- **Unsupervised Feature Selection technique**: Unsupervised Feature selection techniques ignore the target variable and can be used for the unlabelled dataset.



1. Wrapper Methods:

In wrapper methodology, selection of features is done by considering it as a search problem, in which different combinations are made, evaluated, and compared with other combinations. It trains the algorithm by using the subset of features iteratively.



On the basis of the output of the model, features are added or subtracted, and with this feature set, the model has trained again.

Some techniques of wrapper methods are:

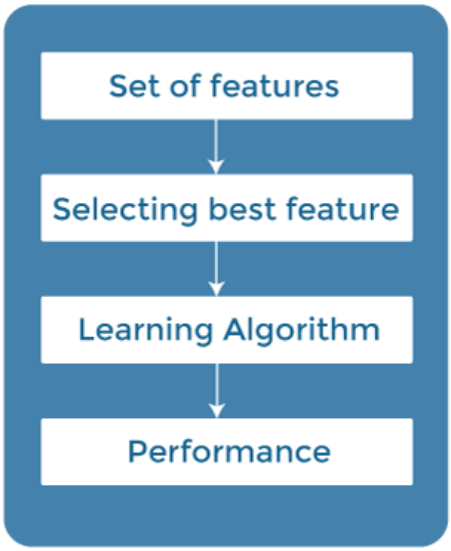
- **Forward selection** - Forward selection is an iterative process, which begins with an empty set of features. After each iteration, it keeps adding on a feature and evaluates the performance to check whether it is improving the performance or not. The process continues until the addition of a new variable/feature does not improve the performance of the model.
- **Backward elimination** - Backward elimination is also an iterative approach, but it is the opposite of forward selection. This technique begins the process by considering all the features and removes the least significant feature. This elimination process continues until removing the features does not improve the performance of the model.
- **Exhaustive Feature Selection** - Exhaustive feature selection is one of the best feature selection methods, which evaluates each feature set as brute-force. It means this method tries & make each possible combination of features and return the best performing feature set.
- **Recursive Feature Elimination** - Recursive feature elimination is a recursive greedy optimization approach, where features are selected by recursively taking a smaller and smaller subset of features. Now, an estimator is trained with each set of features, and the importance of each feature is determined using *coef_attribute* or through a *feature_importances_attribute.*

2. Filter Methods:

In Filter Method, features are selected on the basis of statistics measures. This method does not depend on the learning algorithm and chooses the features as a pre-processing step.

The filter method filters out the irrelevant feature and redundant columns from the model by using different metrics through ranking.

The advantage of using filter methods is that it needs low computational time and does not overfit the data.



Some common techniques of Filter methods are as follows:

Information Gain: Information gain determines the reduction in entropy while transforming the dataset. It can be used as a feature selection technique by calculating the information gain of each variable with respect to the target variable.

Chi-square Test: Chi-square test is a technique to determine the relationship between the categorical variables. The chi-square value is calculated between each feature and the target variable, and the desired number of features with the best chi-square value is selected.

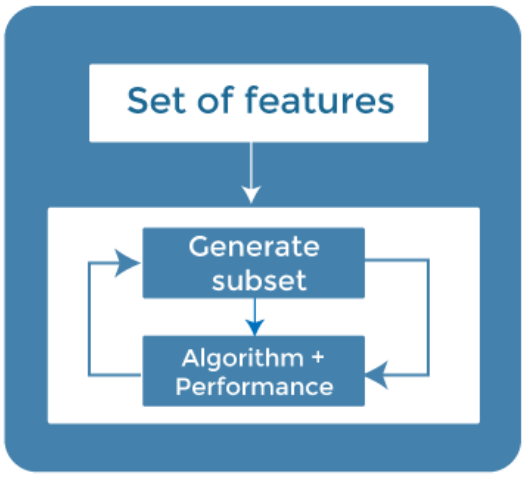
Fisher's Score: Fisher's score is one of the popular supervised technique of features selection. It returns the rank of the variable on the fisher's criteria in descending order. Then we can select the variables with a large fisher's score.

Missing Value Ratio: The value of the missing value ratio can be used for evaluating the feature set against the threshold value. The formula for obtaining the missing value ratio is the number of missing values in each column divided by the total number of observations. The variable is having more than the threshold value can be dropped.

$$\text{Missing Value Ratio}=\frac{Number\ of\ Missing\ values*100}{Total\ number\ of\ observations}$$

3. Embedded Methods:

Embedded methods combined the advantages of both filter and wrapper methods by considering the interaction of features along with low computational cost. These are fast processing methods similar to the filter method but more accurate than the filter method.



These methods are also iterative, which evaluates each iteration, and optimally finds the most important features that contribute the most to training in a particular iteration. Some techniques of embedded methods are:

- **Regularization**- Regularization adds a penalty term to different parameters of the machine learning model for avoiding overfitting in the model. This penalty term is added to the coefficients; hence it shrinks some coefficients to zero. Those features with zero coefficients can be removed from the dataset. The types of regularization techniques are L1 Regularization (Lasso Regularization) or Elastic Nets (L1 and L2 regularization).
- **Random Forest Importance** - Different tree-based methods of feature selection help us with feature importance to provide a way of selecting features. Here, feature importance specifies which feature has more importance in model building or has a great impact on the target variable.

  Random Forest is such a tree-based method, which is a type of bagging algorithm that aggregates a different number of decision trees. It automatically ranks the nodes by their performance or decrease in the impurity (Gini impurity) over all the trees. Nodes are arranged as per the impurity values, and thus it allows to pruning of trees below a specific node. The remaining nodes create a subset of the most important features.

Feature selection using Decision Tree:

Feature selection using decision trees involves identifying the most important features in a dataset based on their contribution to the decision tree's performance.

What are decision trees?

Decision trees are a popular machine learning algorithm used for both classification and regression tasks. They model decisions based on the features of the data and their outcomes.

How do decision trees play a role in feature selection?

- Decision trees select the 'best' feature for splitting at each node based on information gain.
- Information gain measures the reduction in entropy (disorder) in a set of data points.
- Features with higher information gain are considered more important for splitting, thus aiding in feature selection.
- By recursively selecting features for splitting, decision trees inherently prioritize the most relevant features for the model.

Feature Selection Using Random Forest:

Random Forest, an ensemble learning method, is widely used for feature selection due to its inherent ability to rank features based on their importance. This article explores the process of feature selection using Random Forest, its benefits, and practical implementation.

What is Random Forest?

Random Forest is a versatile machine learning algorithm that operates by constructing multiple decision trees during training and outputting the mode of the classes (classification) or mean prediction (regression) of the individual trees. It combines the concepts of bagging (bootstrap aggregating) and random feature selection, leading to improved accuracy and robustness.

Key Concepts of Random Forest:

- **Ensemble Learning:** Combines the predictions of several base estimators to improve generalizability and robustness.
- **Decision Trees:** Each tree is trained on a bootstrap sample from the training data.
- **Random Feature Selection:** At each split in the tree, a random subset of features is considered for splitting.

Why Use Random Forest for Feature Selection?

Random Forest is particularly suited for feature selection for several reasons:

- **Intrinsic Feature Ranking:** Random Forest provides a built-in method to evaluate the importance of features.
- **Handles High Dimensionality:** Effective even when the number of features is much larger than the number of samples.
- **Non-Linearity:** Can capture complex interactions between features without requiring explicit specification of interactions.

Here's how feature selection can be performed using Random Forests:

1. **Train a Random Forest Model**: First, you need to train a Random Forest model on your dataset. Random Forests are an ensemble learning method that constructs multiple decision trees during training.
2. **Calculate Feature Importance**: Similar to decision trees, Random Forests provide a measure of feature importance. However, instead of using a single tree, Random Forests calculate the importance of each feature by averaging the importance scores across all the individual trees in the ensemble. There are different methods to calculate feature importance in Random Forests, but a common approach is to use the mean decrease in impurity (e.g., Gini impurity or entropy) or the mean decrease in accuracy when the values of a feature are permuted randomly.
3. **Rank Features**: Once you have the importance scores for each feature, you can rank them in descending order of importance.
4. **Select Top Features**: Similar to the decision tree approach, you can choose to select a subset of the top-ranked features or use a threshold to filter out less important features based on your requirements and domain knowledge.
5. **Train Model with Selected Features**: After selecting the relevant features, you can train your machine learning model (e.g., Random Forest, or any other algorithm) using only the selected features.