Slide Name - 12Outlier

Book Chapter – 12

What Are Outliers?

In simple terms, an outlier is an extremely high or extremely low data point relative to the nearest data point and the rest of the neighboring co-existing values in a data graph or dataset. Outliers are data components that cannot be combined in a given class or cluster. These are the data objects which have several behaviour from the usual behaviour of different data objects. Ex: Unusual credit card purchase.

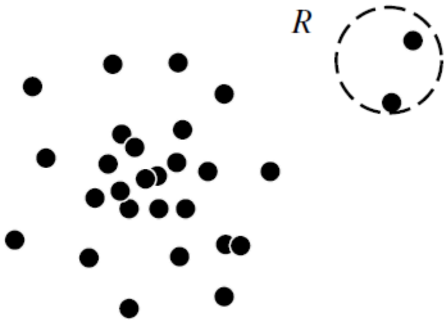
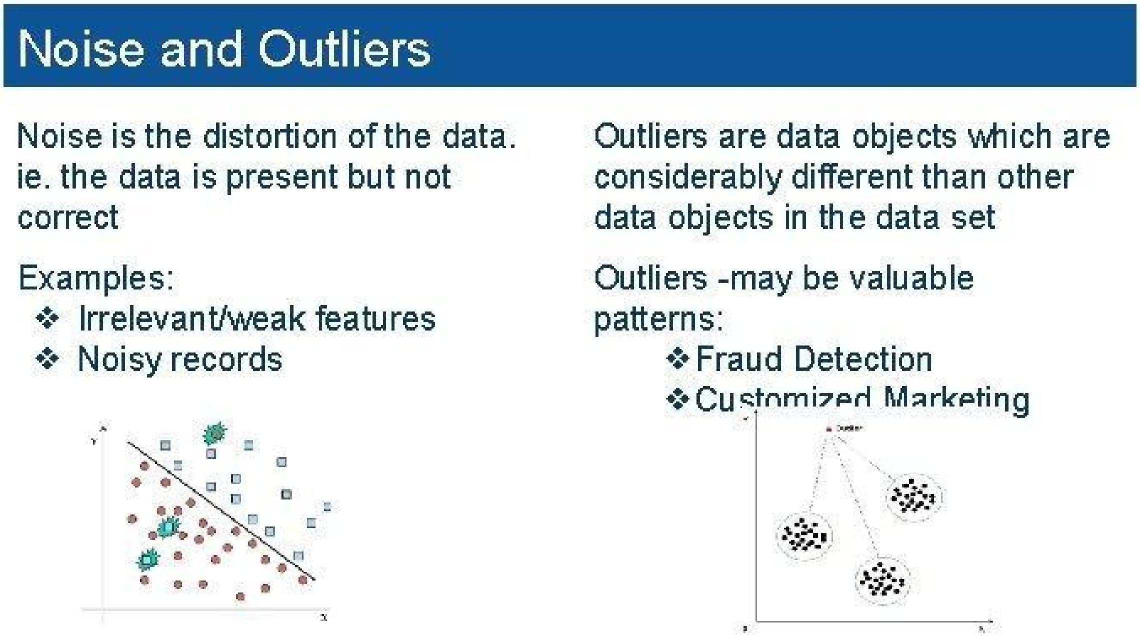


Figure1: The objects in region *R* are outliers.

The objects in region *R* are significantly different. It is unlikely that they follow the same distribution as the other objects in the data set. Thus, the objects in *R* are outliers in the data set.

Everything is not an outlier

Outliers are different from noisy data. Noise is a random error or variance in a measured variable. In general, noise is not interesting in data analysis, including outlier detection. For example, in credit card fraud detection, a customer's purchase behavior can be modeled as a random variable. A customer may generate some "noise transactions" that may seem like "random errors" or "variance," such as by buying a bigger lunch one day, or having one more cup of coffee than usual. Such transactions should not be treated as outliers; otherwise, the credit card company would incur heavy costs from verifying that many transactions. The company may also lose customers by bothering them with multiple false alarms.



Noise and Outliers

Noise is the distortion of the data. ie. the data is present but not correct

Examples:

❖ Irrelevant/weak features

❖ Noisy records

Outliers are data objects which are considerably different than other data objects in the data set

Outliers -may be valuable patterns:

❖ Fraud Detection

❖ Customized Marketing

Types of Outliers

Ref. https://www.geeksforgeeks.org/types-of-outliers-in-data-mining/

Outliers can be classified into three categories, namely global outliers, collective outliers and contextual (or conditional) outliers.

1) Global Outliers

They are also known as Point Outliers. These are the simplest form of outliers. If, in a given dataset, a data point strongly deviates from all the rest of the data points, it is known as a global outlier. Mostly, all of the outlier detection methods are aimed at finding global outliers.

For example, In Intrusion Detection System, if a large number of packages are broadcast in a very short span of time, then this may be considered as a global outlier and we can say that that particular system has been potentially hacked.

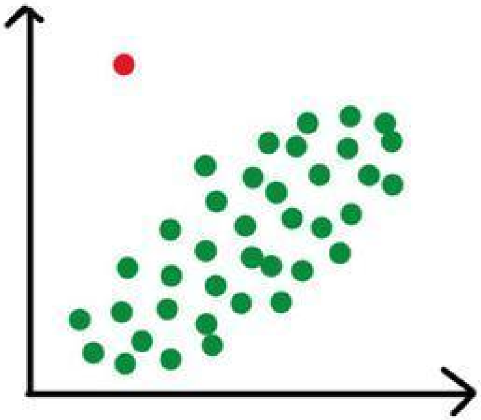


Figure2: The red data point is a global outlier.

2) Collective Outliers

As the name suggests, if in a given dataset, some of the data points, as a whole, deviate significantly from the rest of the dataset, they may be termed as collective outliers. Here, the individual data objects may not be outliers, but when seen as a whole, they may behave as outliers. To detect these types of outliers, we might need background information about the relationship between those data objects showing the behavior of outliers.

For example: In an Intrusion Detection System, a DOS (denial-of-service) package from one computer to another may be considered as normal behavior. However, if this happens with several computers at the same time, then this may be considered as abnormal behavior and as a whole they can be termed as collective outliers.

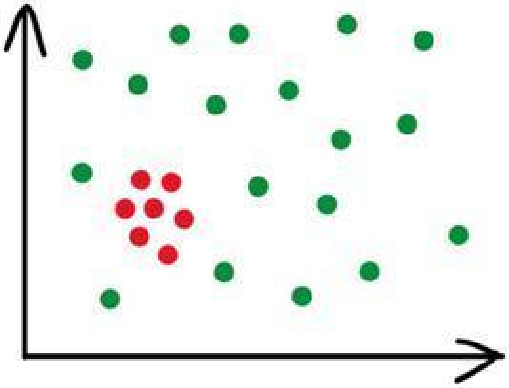


Figure3: The red data points as a whole are collective outliers.

3) Contextual Outliers

They are also known as Conditional Outliers. Here, if in a given dataset, a data object deviates significantly from the other data points based on a specific context or condition only. A data point may be an outlier due to a certain condition and may show normal behavior under another condition. Therefore, a context has to be specified as part of the problem statement in order to identify contextual outliers. Contextual outlier analysis provides flexibility for users where one can examine outliers in different contexts, which can be highly desirable in many applications. The attributes of the data point are decided on the basis of both contextual and behavioral attributes.

For example: A temperature reading of 40°C may behave as an outlier in the context of a “winter season” but will behave like a normal data point in the context of a “summer season”.

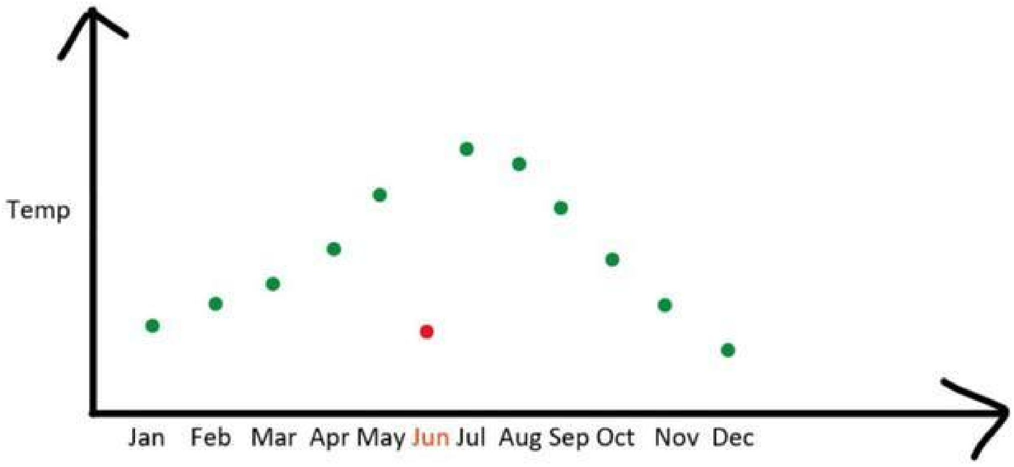


Figure4: A low temperature value in June is a contextual outlier because the same value in December is not an outlier.

Outlier Detection Methods

Ref. https://www.tutorialspoint.com/what-are-the-methods-of-outlier-detection

There are two ways to categorize outlier detection methods. One of them is based on whether user labeled examples of outliers can be obtained: Supervised, semi supervised vs. unsupervised methods.

1) **Supervised Methods**

Supervised methods model data normality and abnormality. Domain professional's tests and label a sample of the basic data. Outlier detection can be modeled as a classification issue. The service is to understand a classifier that can identify outliers.

The sample can be used for training and testing. In various applications, the professionals can label only the normal objects, and several objects not connecting the model of normal objects are documented as outliers. There are different methods model the outliers and consider objects not connecting the model of outliers as normal.

2) **Unsupervised Methods**

In various application methods, objects labeled as "normal" or "outlier" are not applicable. Therefore, an unsupervised learning approach has to be used. Unsupervised outlier detection methods create an implicit assumption such as the normal objects are considerably "clustered."

An unsupervised outlier detection method predict that normal objects follow a pattern far more generally than outliers. Normal objects do not have to decline into one team sharing large similarity. Instead, they can form several groups, where each group has multiple features.

This assumption cannot be true sometime. The normal objects do not send some strong patterns. Rather than, they are uniformly distributed. The collective outliers, share large similarity in a small area.

Unsupervised methods cannot identify such outliers efficiently. In some applications, normal objects are separately distributed, and several objects do not follow strong patterns. For example, in some intrusion detection and computer virus detection issues, normal activities are distinct and some do not decline into high-quality clusters.

Some clustering methods can be adapted to facilitate as unsupervised outlier detection methods. The main idea is to discover clusters first, and therefore the data objects not belonging to some cluster are identified as outliers. However, such methods deteriorate from two issues. First, a data object not belonging to some cluster can be noise rather than an outlier. Second, it is expensive to discover clusters first and then discover outliers.

3) **Semi-Supervised Methods**

In several applications, although obtaining some labeled instance is possible, the number of such labeled instances is small. It can encounter cases where only a small group of the normal and outlier objects are labeled, but some data are unlabeled. Semi-supervised outlier detection methods were produced to tackle such methods.

Semi-supervised outlier detection methods can be concerned as applications of Semi-supervised learning approaches. For example, when some labeled normal objects are accessible, it can use them with unlabeled objects that are nearby, to train a model for normal objects. The model of normal objects is used to identify outliers—those objects not suitable the model of normal objects are defined as outliers.

Cross-Validation

To evaluate the performance of any machine learning model we need to test it on some unseen data. Based on the models performance on unseen data we can say weather our model is Under-fitting/Over-fitting/Well generalized. Cross validation (CV) is one of the technique used to test the effectiveness of a machine learning models, it is also a re-sampling procedure used to evaluate a model if we have a limited data.

To perform cross validation, we need to keep aside a sample/portion of the data on which is not used to train the model, later use this sample for testing/validating. Below are the few common techniques used for CV:

Train-Test Split Approach

In this approach we randomly split the complete data into training and test sets. Then Perform the model training on the training set and use the test set for validation purpose, ideally split the data into 70:30 or 80:20. With this approach there is a possibility of high bias if we have limited data, because we would miss some information about the data which we have not used for training. If our data is huge and our test sample and train sample has the same distribution, then this approach is acceptable.

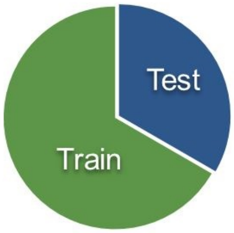


Figure: Train-Test Split approach

K-fold cross-validation

In k-fold cross-validation, the original dataset is equally partitioned into k subparts or folds. Out of the k-folds or groups, for each iteration, one group is selected as validation data, and the remaining (k-1) groups are selected as training data. This approach is a very popular CV approach because it is easy to understand, and the output is less biased than other methods.
The process is repeated for k times until each group is treated as validation and remaining as training data.

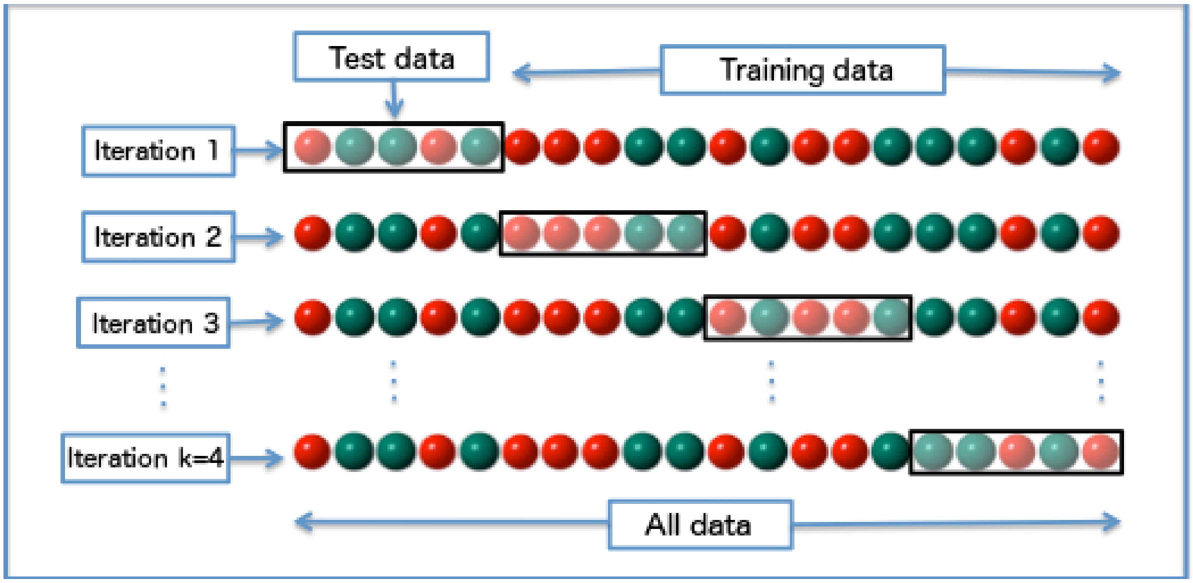


Figure5: K-fold cross-validation

Holdout Method

The holdout technique is an exhaustive cross-validation method, that randomly splits the dataset into train and test data depending on data analysis.

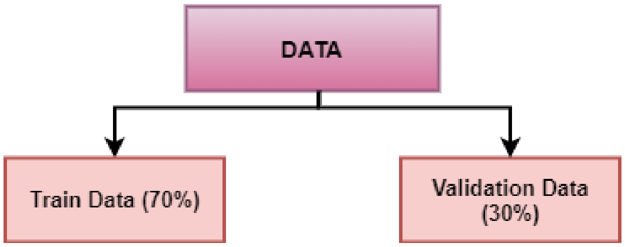


Figure: 70:30 split of Data into training and validation data respectively

In the case of holdout cross-validation, the dataset is randomly split into training and validation data. Generally, the split of training data is more than test data. The training data is used to induce the model and validation data is evaluates the performance of the model. The more data is used to train the model, the better the model is. For the holdout cross-validation method, a good amount of data is isolated from training.

Reference

1. https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f
2. https://towardsdatascience.com/understanding-8-types-of-cross-validation-80c935a4976d#:~:text=The%20holdout%20technique%20is%20an,data%20depending%20on%20data%20analysis.&text=In%20the%20case%20of%20holdout,is%20more%20than%20test%20data.

Slide - 10ClusBasic

Book Chapter – 10

What is Cluster Analysis?

Cluster Analysis is the process to find similar groups of objects in order to form clusters. It is an unsupervised machine learning-based algorithm that acts on unlabeled data.

Cluster:

The given data is divided into different groups by combining similar objects into a group. This group is nothing but a cluster. A cluster is nothing but a collection of similar data which is grouped together.

For example, consider a dataset of vehicles given in which it contains information about different vehicles like cars, buses, bicycles, etc. As it is unsupervised learning there are no class labels like Cars, Bikes, etc. for all the vehicles, all the data is combined and is not in a structured manner.

Now our task is to convert the unlabeled data to labeled data and it can be done using clusters.

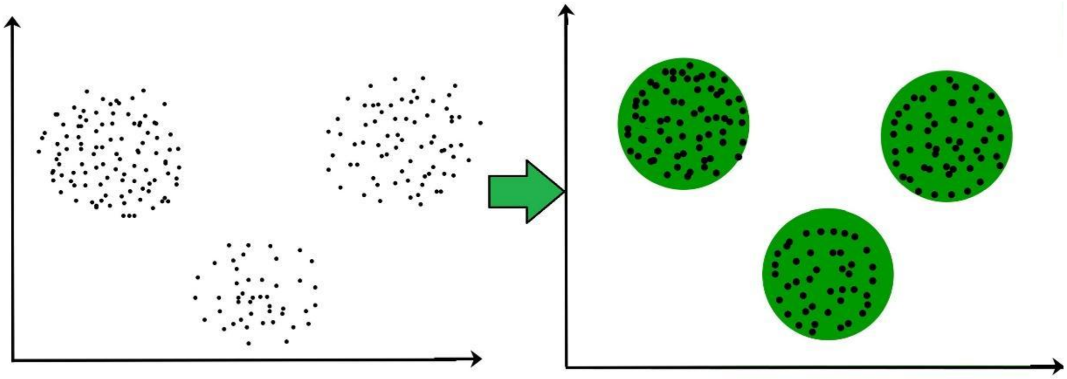


Figure: Clustering

The main idea of cluster analysis is that it would arrange all the data points by forming clusters like cars cluster which contains all the cars, bikes clusters which contains all the bikes, etc. Simply it is the partitioning of similar objects which are applied to unlabeled data.

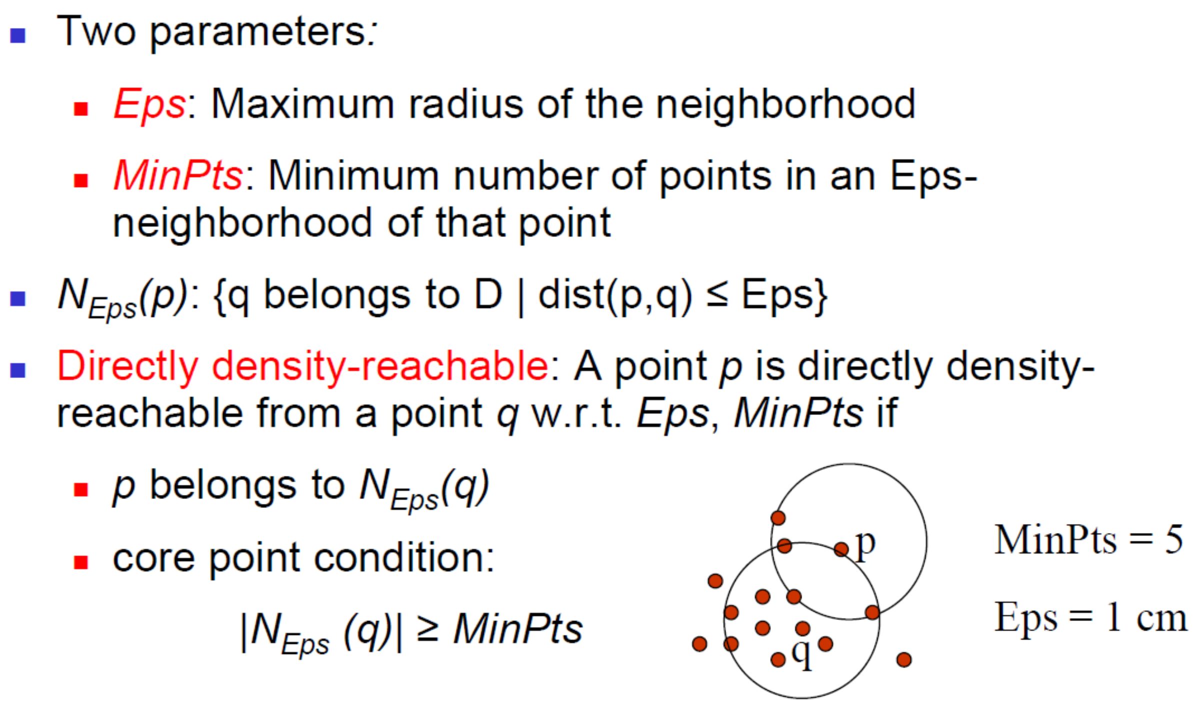
Applications of Cluster Analysis:

1. It is widely used in image processing, data analysis, and pattern recognition.
2. It helps marketers to find the distinct groups in their customer base and they can characterize their customer groups by using purchasing patterns.
3. It can be used in the field of biology, by deriving animal and plant taxonomies and identifying genes with the same capabilities.
4. It also helps in information discovery by classifying documents on the web.

DBSCAN: Density Based Clustering

1) What is Density-based clustering?

Density-Based Clustering refers to one of the most popular unsupervised learning methodologies used in model building and machine learning algorithms. The data points in the region separated by two clusters of low point density are considered as noise. The surroundings with a radius ε of a given object are known as the **ε** neighborhood of the object. If the **ε** neighborhood of the object comprises at least a minimum number, MinPts of objects, then it is called a core object.



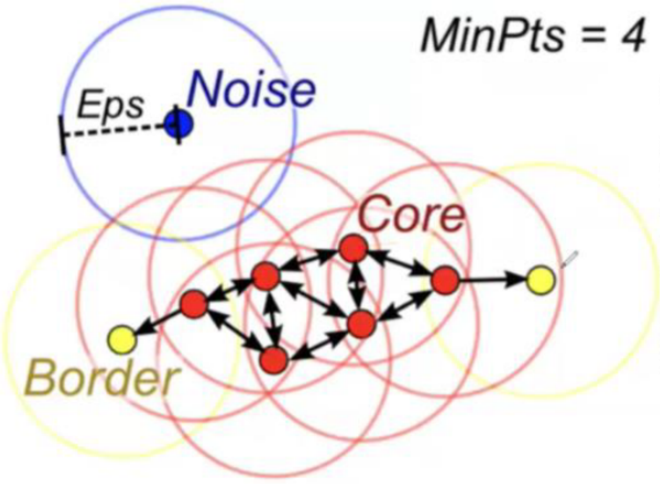
- Two parameters*:*
  - *Eps*: Maximum radius of the neighborhood
  - *MinPts*: Minimum number of points in an Eps-neighborhood of that point
- $N_{Eps}(p)$: {q belongs to D | dist(p,q) ≤ Eps}
- Directly density-reachable: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if
  - *p* belongs to $N_{Eps}(q)$
  - core point condition:

    $|N_{Eps}(q)| \geq MinPts$

MinPts = 5

Eps = 1 cm

Density-Based Spatial Clustering of Applications with Noise(DBSCAN)



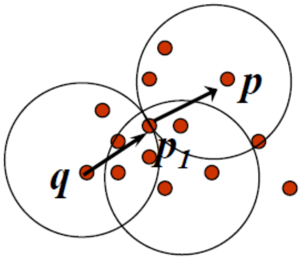
Red: Core Points

Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria

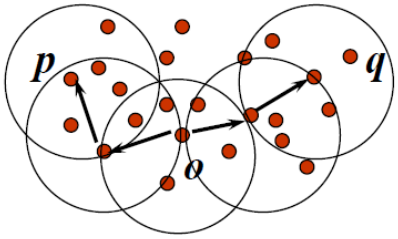
Blue: Noise point. Not assigned to a cluster

2) Explain Density-reachable and Density-connected.

- Density-reachable:
  - A point *p* is density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i



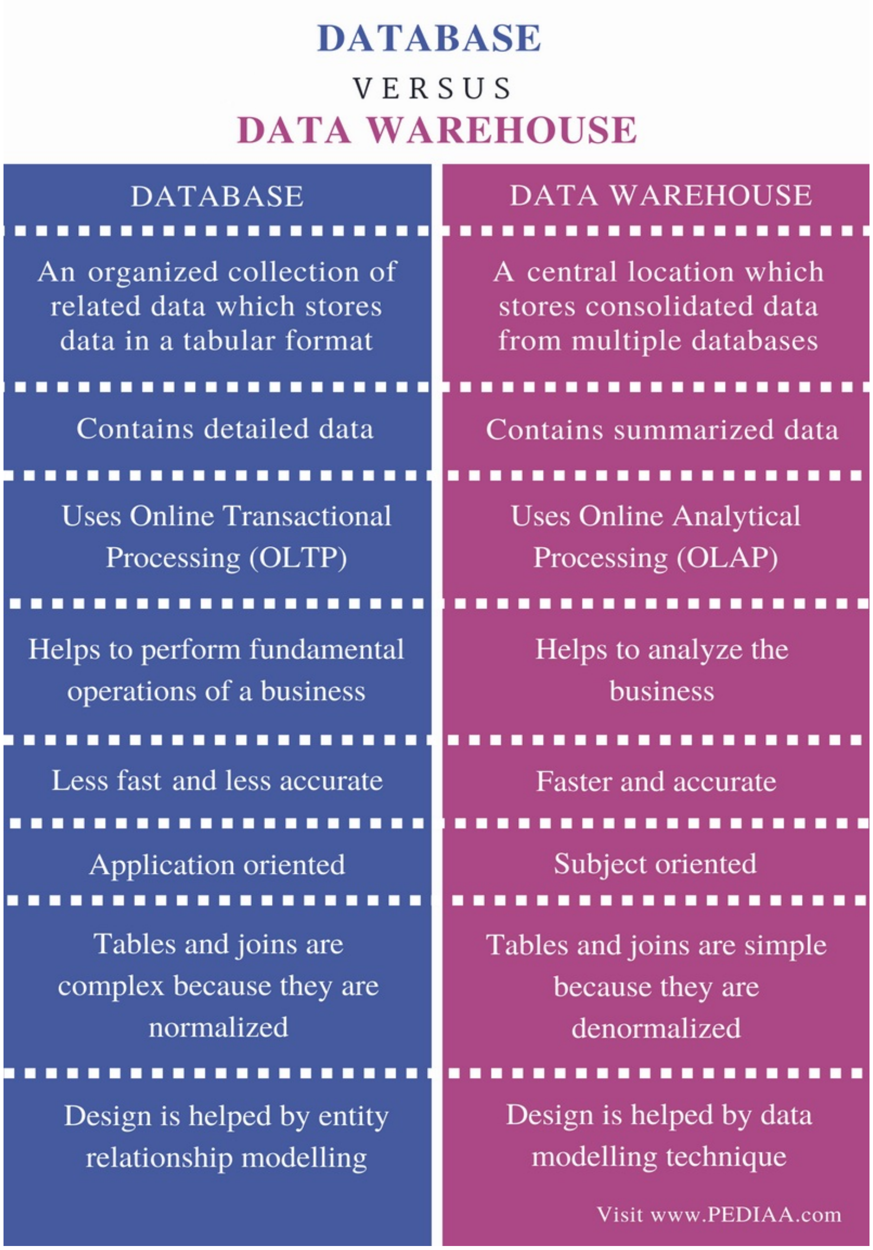
- Density-connected
  - A point *p* is density-connected to a point *q* w.r.t. *Eps*, *MinPts* if there is a point *O* such that both, *p* and *q* are density-reachable from *O* w.r.t. *Eps* and *MinPts*



Slide - 04 DW-OLAP

Book Chapter – 4

1) Database VS Datawarehouse



DATABASE VERSUS DATA WAREHOUSE

DATABASE	DATA WAREHOUSE
An organized collection of related data which stores data in a tabular format	A central location which stores consolidated data from multiple databases
Contains detailed data	Contains summarized data
Uses Online Transactional Processing (OLTP)	Uses Online Analytical Processing (OLAP)
Helps to perform fundamental operations of a business	Helps to analyze the business
Less fast and less accurate	Faster and accurate
Application oriented	Subject oriented
Tables and joins are complex because they are normalized	Tables and joins are simple because they are denormalized
Design is helped by entity relationship modelling	Design is helped by data modelling technique

Visit www.PEDIAA.com

Why do we need a data warehouse when we already have a database?

A data warehouse is designed to separate big data analysis and query processes (more focused on data reading) from transactional processes (focused on writing). This approach, therefore, allows a company to multiply its analytical power without impacting its transactional systems and day-to-day management needs.

This is a question comparing scope/intent with a technology. On a technical level, a data warehouse *is* a database but It's just a big one. We need a data warehouse because most databases are not capable of handing the amount of data and the multi-dimensional queries that a data warehouse can for a large organization. Data warehouses typically denormalize their data, prioritizing read operations over write operations.

2) Different features & Benefits of data warehouse

A data warehouse is designed to separate big data analysis and query processes (more focused on data reading) from transactional processes (focused on writing).
data warehouses never put emphasis only on current operations. Instead, it focuses on demonstrating and analysis of data to make various decisions.

Features:

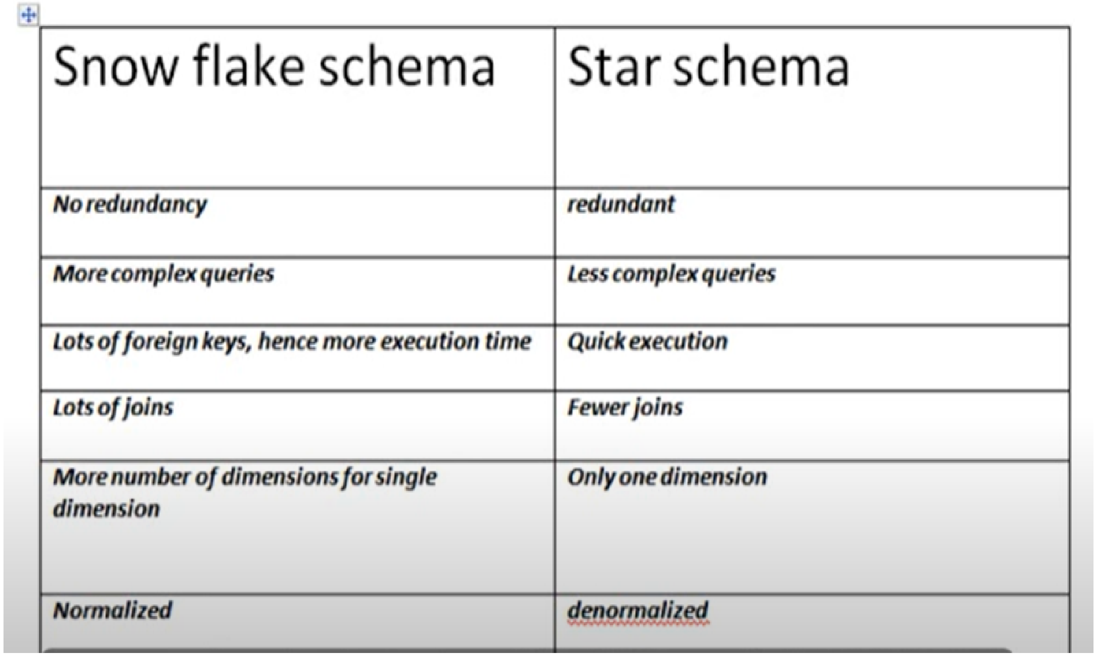
- Are often deployed as a central database for the enterprise.
- Provide ETL (extract, transform, load) data processing capability.
- Store metadata.
- Include access to reporting tools.

Benefits:

- It Saves Time.
- Improves Data Quality.
- Improves Business Intelligence.
- Leads to Data Consistency.

- Enhances Return on Investment (ROI)
- Stores Historical Data.
- Increases Data Security.

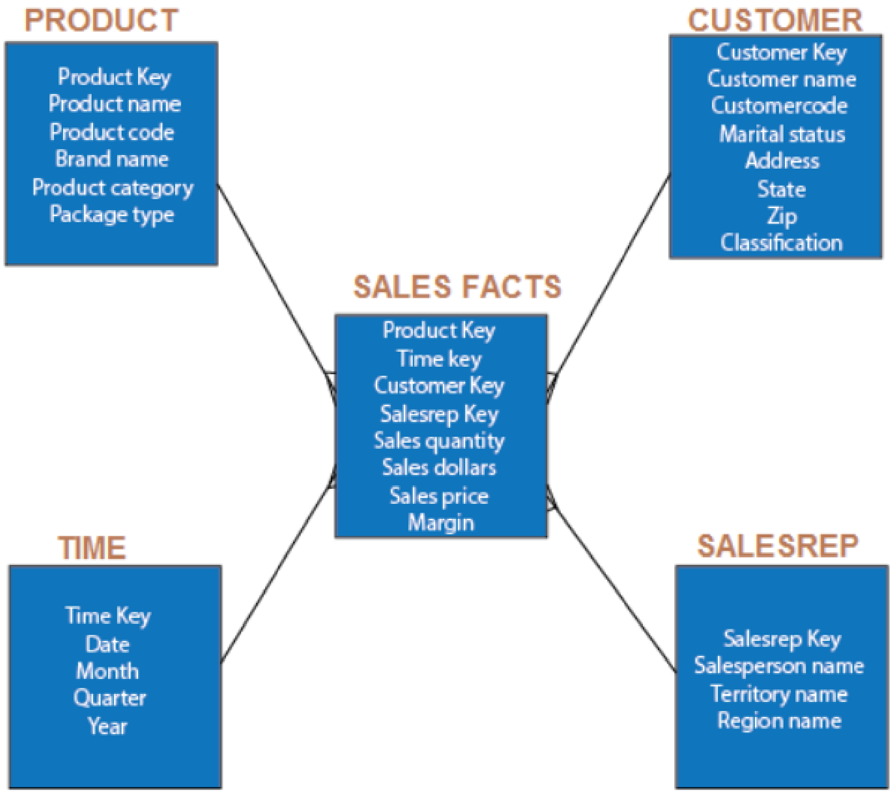
3) Difference between star schema & snowflake schema & example



Snow flake schema	Star schema
No redundancy	*redundant*
More complex queries	*Less complex queries*
Lots of foreign keys, hence more execution time	*Quick execution*
Lots of joins	*Fewer joins*
More number of dimensions for single dimension	*Only one dimension*
Normalized	*denormalized*

Star schema:

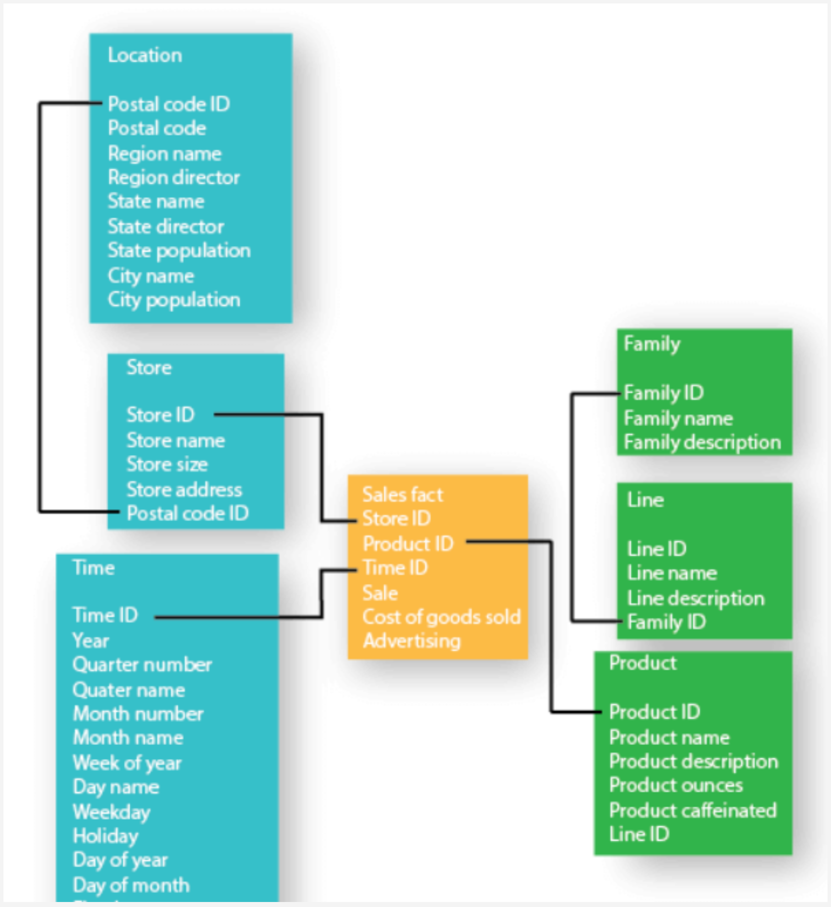
Sales price, sale quantity, distance, speed, weight, and weight measurements are a few examples of fact data in a star schema.



STAR Schema

The figure shows a simple STAR schema for sales in a manufacturing company.

snowflake schema:



The figure shows a snowflake schema with a Sales fact table, with Store, Location, Time, Product, Line, and Family dimension tables.

Data Cube

The data cube method is an interesting technique with many applications. Data cubes could be sparse in many cases because not every cell in each dimension may have corresponding data in the database.