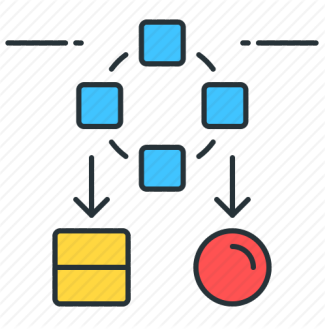
Data Modeling

Tushar B. Kute,
http://tusharkute.com



What is data modeling?

- Data modeling (data modelling) is the process of creating a data model for the data to be stored in a database.
- This data model is a conceptual representation of Data objects, the associations between different data objects, and the rules.
- Data modeling helps in the visual representation of data and enforces business rules, regulatory compliances, and government policies on the data.
- Data Models ensure consistency in naming conventions, default values, semantics, security while ensuring quality of the data.

Data Model

- The Data Model is defined as an abstract model that organizes data description, data semantics, and consistency constraints of data.
- The data model emphasizes on what data is needed and how it should be organized instead of what operations will be performed on data.
- Data Model is like an architect's building plan, which helps to build conceptual models and set a relationship between data items.

Data Models: Types

- The two types of Data Modeling Techniques are
  - Entity Relationship (E-R) Model
  - UML (Unified Modelling Language)

Why data modeling?

- Ensures that all data objects required by the database are accurately represented. Omission of data will lead to creation of faulty reports and produce incorrect results.
- A data model helps design the database at the conceptual, physical and logical levels.
- Data Model structure helps to define the relational tables, primary and foreign keys and stored procedures.

Why data modeling?

- It provides a clear picture of the base data and can be used by database developers to create a physical database.
- It is also helpful to identify missing and redundant data.
- Though the initial creation of data model is labor and time consuming, in the long run, it makes your IT infrastructure upgrade and maintenance cheaper and faster.

Data model types

- Types of Data Models: There are mainly three different types of data models: conceptual data models, logical data models, and physical data models, and each one has a specific purpose.
- The data models are used to represent the data and how it is stored in the database and to set the relationship between data items.

Data model types

- Conceptual Data Model: This Data Model defines WHAT the system contains.
- This model is typically created by Business stakeholders and Data Architects.
- The purpose is to organize, scope and define business concepts and rules.

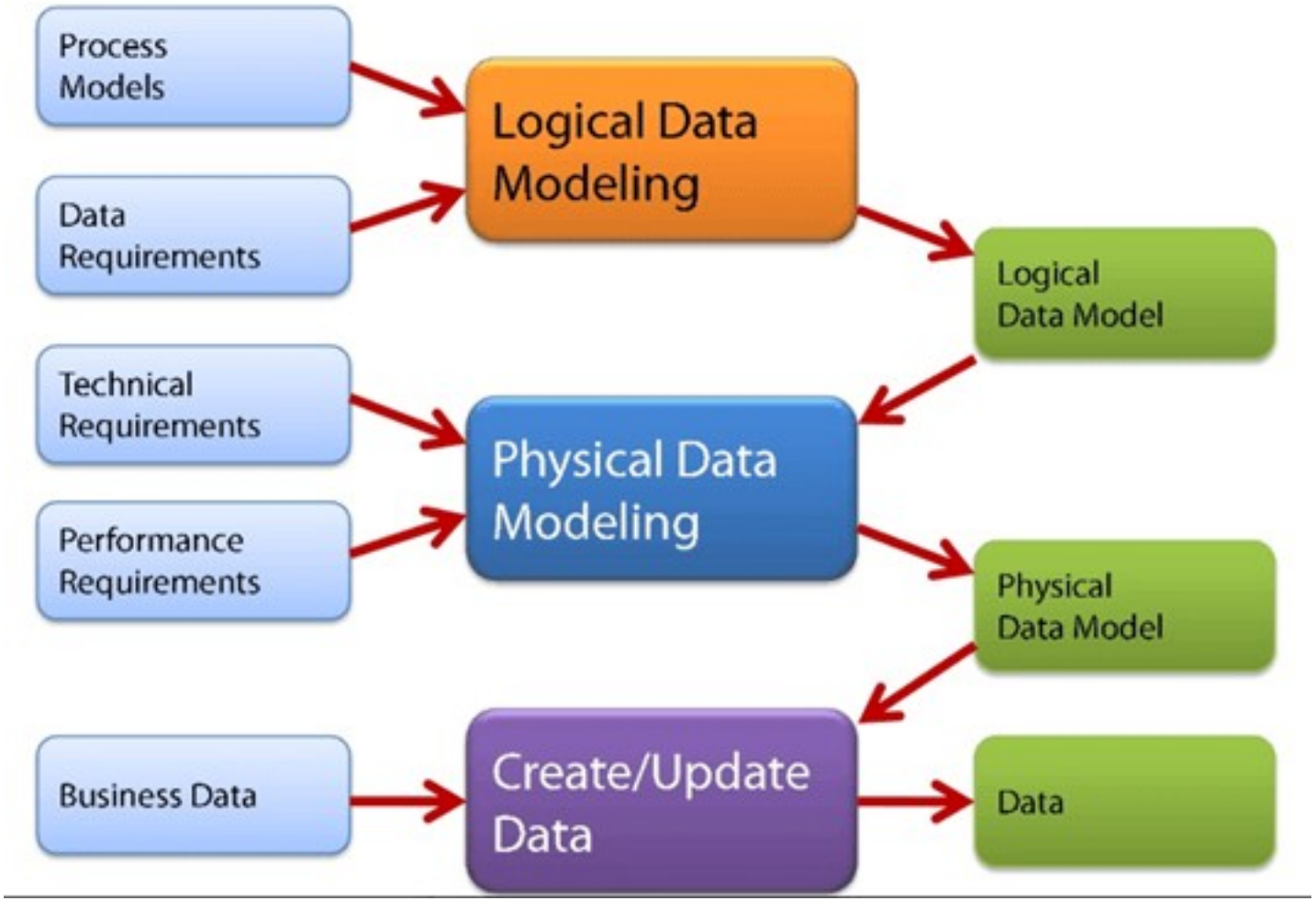
Data model types

- Logical Data Model: Defines HOW the system should be implemented regardless of the DBMS.
- This model is typically created by Data Architects and Business Analysts.
- The purpose is to developed technical map of rules and data structures.

Data model types

- Physical Data Model: This Data Model describes HOW the system will be implemented using a specific DBMS system.
- This model is typically created by DBA and developers.
- The purpose is actual implementation of the database.

Data model types

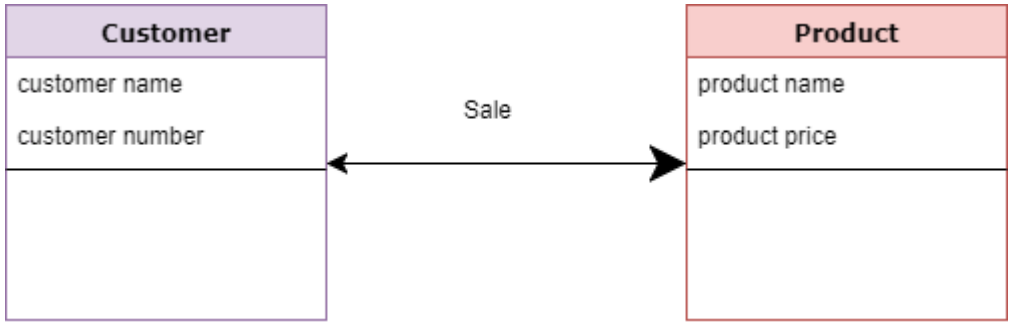


Conceptual data model

- A Conceptual Data Model is an organized view of database concepts and their relationships. The purpose of creating a conceptual data model is to establish entities, their attributes, and relationships.
- In this data modeling level, there is hardly any detail available on the actual database structure. Business stakeholders and data architects typically create a conceptual data model.
- The 3 basic tenants of Conceptual Data Model are
  - Entity: A real-world thing
  - Attribute: Characteristics or properties of an entity
  - Relationship: Dependency or association between two entities

Conceptual data model

- Data model example:
  - Customer and Product are two entities. Customer number and name are attributes of the Customer entity
  - Product name and price are attributes of product entity
  - Sale is the relationship between the customer and product

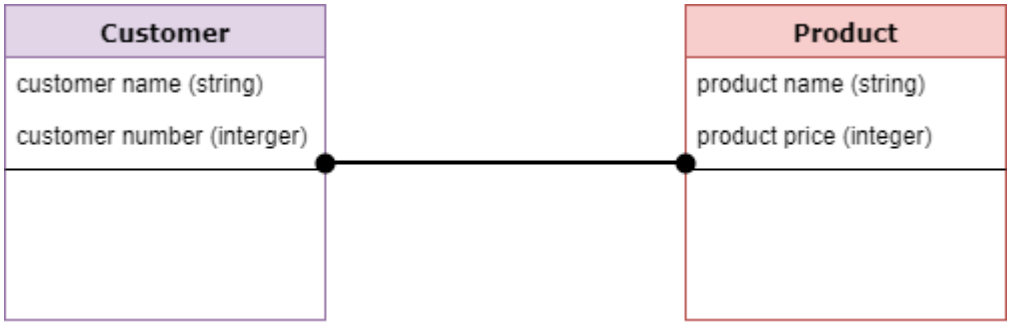


Characteristics of conceptual data model

- Offers Organisation-wide coverage of the business concepts.
- This type of Data Models are designed and developed for a business audience.
- The conceptual model is developed independently of hardware specifications like data storage capacity, location or software specifications like DBMS vendor and technology. The focus is to represent data as a user will see it in the "real world."
- Conceptual data models known as Domain models create a common vocabulary for all stakeholders by establishing basic concepts and scope.

Logical Data Model

- The Logical Data Model is used to define the structure of data elements and to set relationships between them.
- The logical data model adds further information to the conceptual data model elements.
- The advantage of using a Logical data model is to provide a foundation to form the base for the Physical model. However, the modeling structure remains generic.

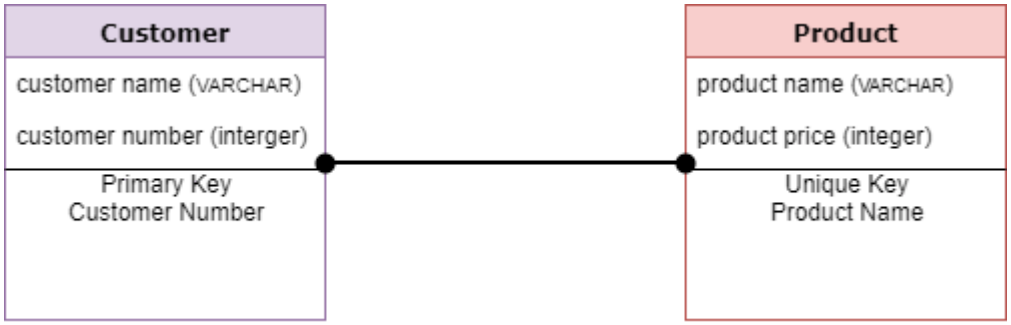


Characteristics of a Logical data model

- Describes data needs for a single project but could integrate with other logical data models based on the scope of the project.
- Designed and developed independently from the DBMS.
- Data attributes will have datatypes with exact precisions and length.
- Normalization processes to the model is applied typically till 3NF.

Physical Data Model

- A Physical Data Model describes a database-specific implementation of the data model. It offers database abstraction and helps generate the schema.
- This is because of the richness of meta-data offered by a Physical Data Model.
- The physical data model also helps in visualizing database structure by replicating database column keys, constraints, indexes, triggers, and other RDBMS features.



Characteristics

- The physical data model describes data need for a single project or application though it maybe integrated with other physical data models based on project scope.
- Data Model contains relationships between tables that which addresses cardinality and nullability of the relationships.
- Developed for a specific version of a DBMS, location, data storage or technology to be used in the project.
- Columns should have exact datatypes, lengths assigned and default values.
- Primary and Foreign keys, views, indexes, access profiles, and authorizations, etc. are defined.

Advantages of Data model

- The main goal of a designing data model is to make certain that data objects offered by the functional team are represented accurately.
- The data model should be detailed enough to be used for building the physical database.
- The information in the data model can be used for defining the relationship between tables, primary and foreign keys, and stored procedures.
- Data Model helps business to communicate the within and across organizations.
- Data model helps to documents data mappings in ETL process
- Help to recognize correct sources of data to populate the model

Disadvantages of Data model

- To develop Data model one should know physical data stored characteristics.
- This is a navigational system produces complex application development, management. Thus, it requires a knowledge of the biographical truth.
- Even smaller change made in structure require modification in the entire application.
- There is no set data manipulation language in DBMS.

Multidimensional Data Model

- Multidimensional Data Model can be defined as a method for arranging the data in the database, with better structuring and organization of the contents in the database.
- Unlike a system with one dimension such as a list, the Multidimensional Data Model can have two or three dimensions of items from the database system.
- It is typically used in the organizations for drawing out Analytical results and generation of reports, which can be used as the main source for imperative decision-making processes.
- This model is typically applied to systems that operate with OLAP techniques (Online Analytical Processing).

Multidimensional Data Model

- Multidimensional data model stores data in the form of data cube.Mostly, data warehousing supports two or three-dimensional cubes.
- A data cube allows data to be viewed in multiple dimensions.
- A dimensions are entities with respect to which an organization wants to keep records.
- For example in store sales record, dimensions allow the store to keep track of things like monthly sales of items and the branches and locations.

Multidimensional Data Model

- A multidimensional databases helps to provide data-related answers to complex business queries quickly and accurately.
- Data warehouses and Online Analytical Processing (OLAP) tools are based on a multidimensional data model.
- OLAP in data warehousing enables users to view data from different angles and dimensions.

Multidimensional Data Model

- Schemas for Multidimensional Data Model are:-
  - Star Schema
  - Snowflakes Schema
  - Fact Constellations Schema

Dimensional Modeling

- Dimensional Modeling (DM) is a data structure technique optimized for data storage in a Data warehouse.
- The purpose of dimensional modeling is to optimize the database for faster retrieval of data.
- The concept of Dimensional Modelling was developed by Ralph Kimball and consists of “fact” and “dimension” tables.

Dimensional Modeling

- A dimensional model in data warehouse is designed to read, summarize, analyze numeric information like values, balances, counts, weights, etc. in a data warehouse.
- In contrast, relation models are optimized for addition, updating and deletion of data in a real-time Online Transaction System.
- These dimensional and relational models have their unique way of data storage that has specific advantages.

Dimensional Modeling

- For instance, in the relational mode, normalization and ER models reduce redundancy in data.
- On the contrary, dimensional model in data warehouse arranges data in such a way that it is easier to retrieve information and generate reports.
- Hence, Dimensional models are used in data warehouse systems and not a good fit for relational systems.

Elements of multidimensional modeling

- Fact
- Dimension
- Attributes
- Fact table
- Dimension Table

Elements of multidimensional modeling

- Fact
  - Facts are the measurements/metrics or facts from your business process. For a Sales business process, a measurement would be quarterly sales number
- Dimension
  - Dimension provides the context surrounding a business process event. In simple terms, they give who, what, where of a fact. In the Sales business process, for the fact quarterly sales number, dimensions would be
    - Who – Customer Names
    - Where – Location
    - What – Product Name
- In other words, a dimension is a window to view information in the facts.

Elements of multidimensional modeling

- Attributes
  - The Attributes are the various characteristics of the dimension in dimensional data modeling.
- In the Location dimension, the attributes can be
  - State
  - Country
  - Zipcode etc.
- Attributes are used to search, filter, or classify facts. Dimension Tables contain Attributes

Elements of multidimensional modeling

- Fact Table
  - A fact table is a primary table in dimension modelling.
- A Fact Table contains
  - Measurements/facts
  - Foreign key to dimension table

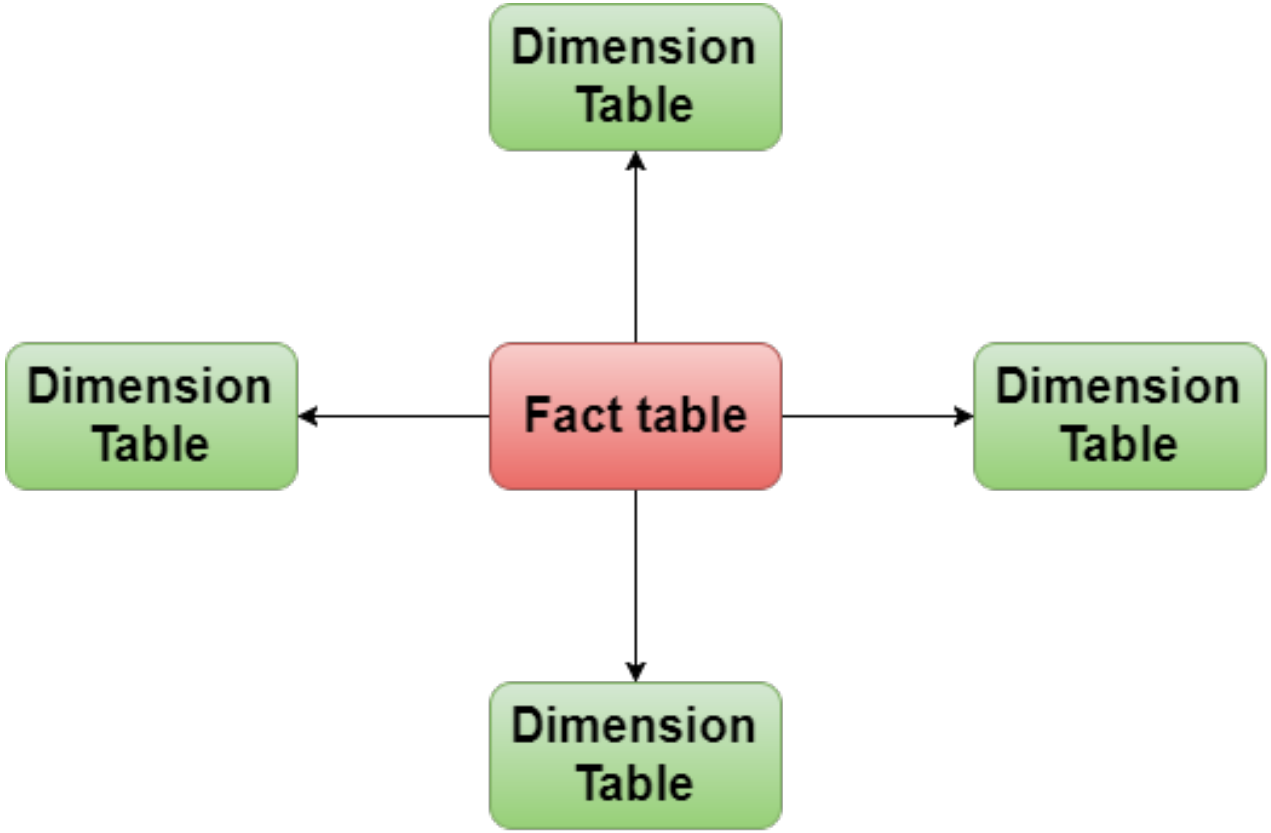
Elements of multidimensional modeling

- A dimension table contains dimensions of a fact.
- They are joined to fact table via a foreign key.
- Dimension tables are de-normalized tables.
- The Dimension Attributes are the various columns in a dimension table
- Dimensions offers descriptive characteristics of the facts with the help of their attributes
- No set limit set for given for number of dimensions
- The dimension can also contain one or more hierarchical relationships

Star Schema

- The simplest data warehouse schema is star schema because its structure resembles a star.
- Star schema consists of data in the form of facts and dimensions.
- The fact table present in the center of star and points of the star are the dimension tables.
- In star schema fact table contain a large amount of data, with no redundancy.
- Each dimension table is joined with the fact table using a primary or foreign key.

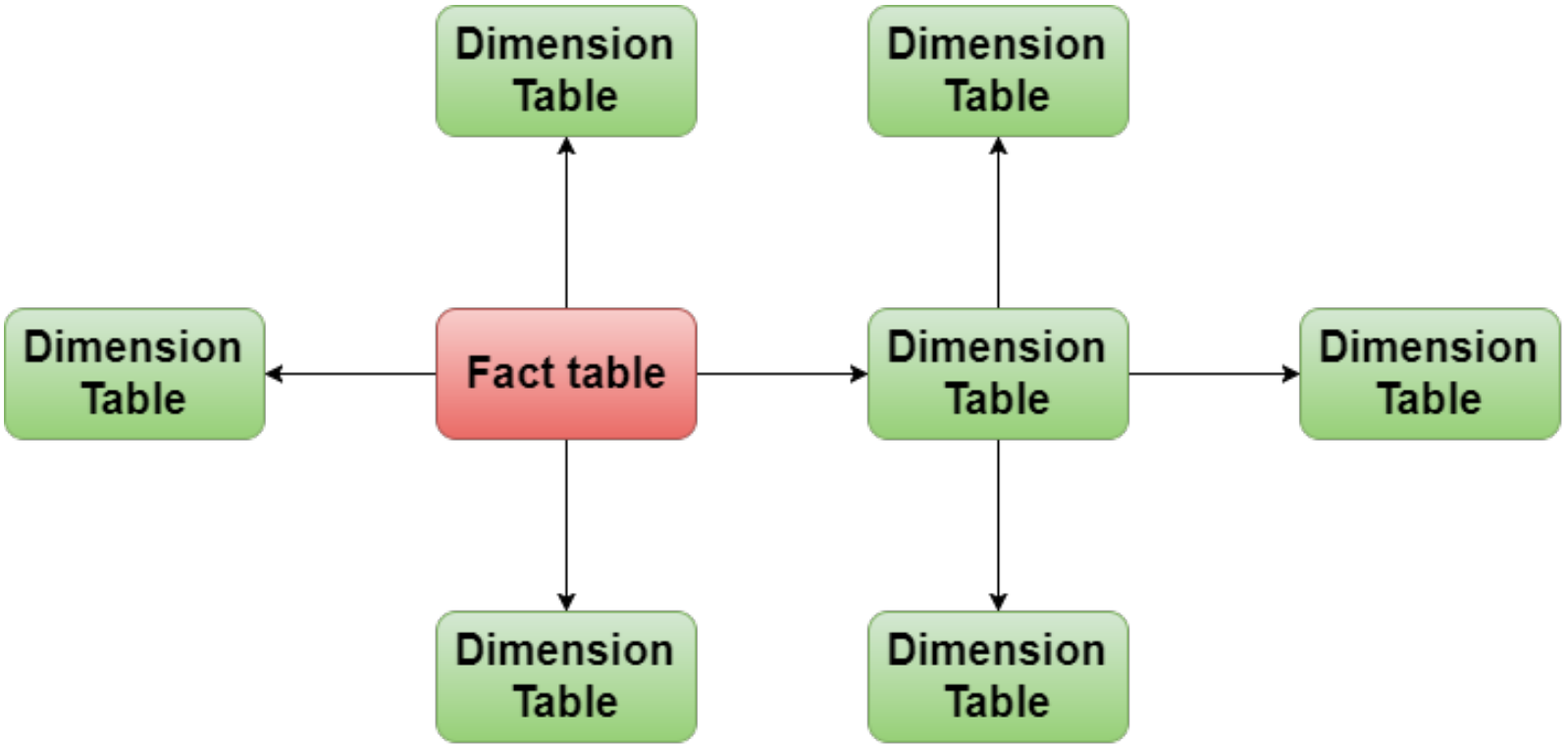
Star Schema


Dimension Table
Dimension Table
Fact table
Dimension Table
Dimension Table

Snowflake Schema

- The snowflake schema is a more complex than star schema because dimension tables of the snowflake are normalized.
- The snowflake schema is represented by centralized fact table which is connected to multiple dimension table and this dimension table can be normalized into additional dimension tables.
- The major difference between the snowflake and star schema models is that the dimension tables of the snowflake model are normalized to reduce redundancies.

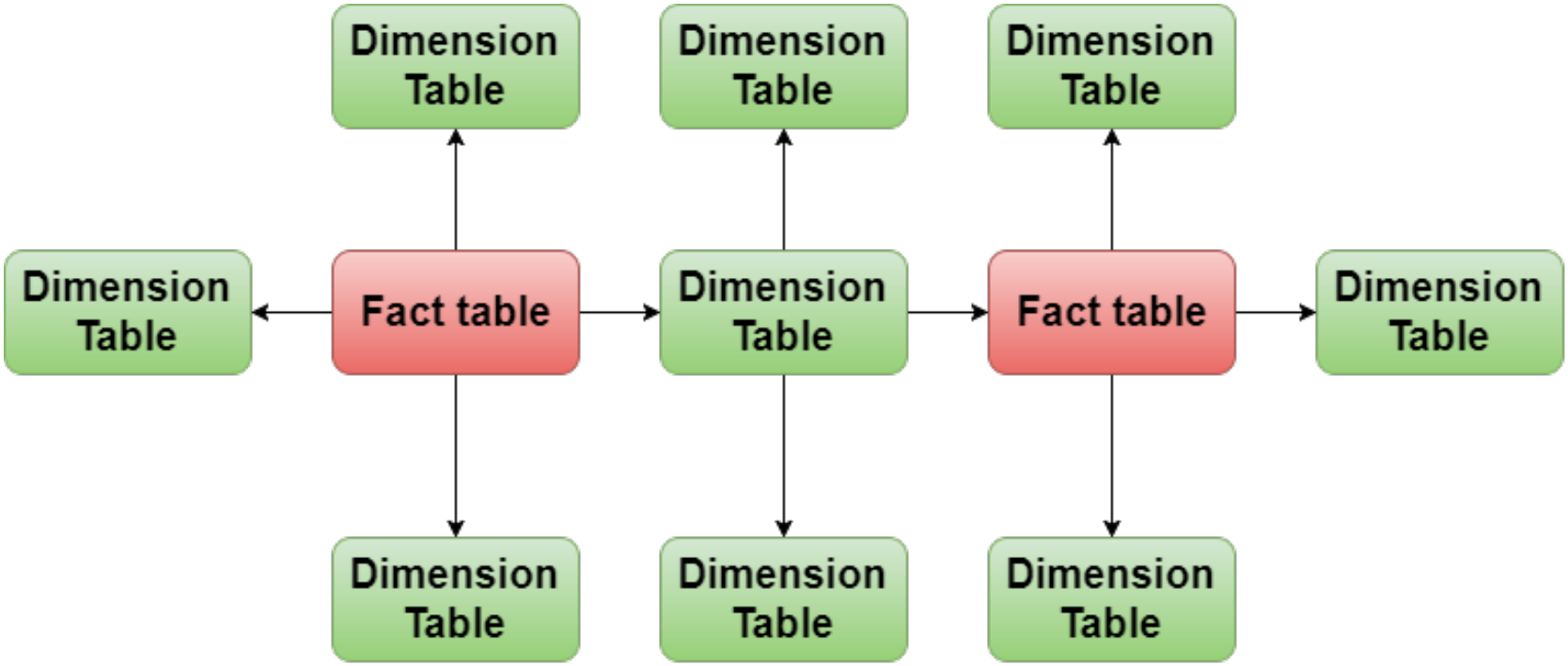
Snowflake Schema



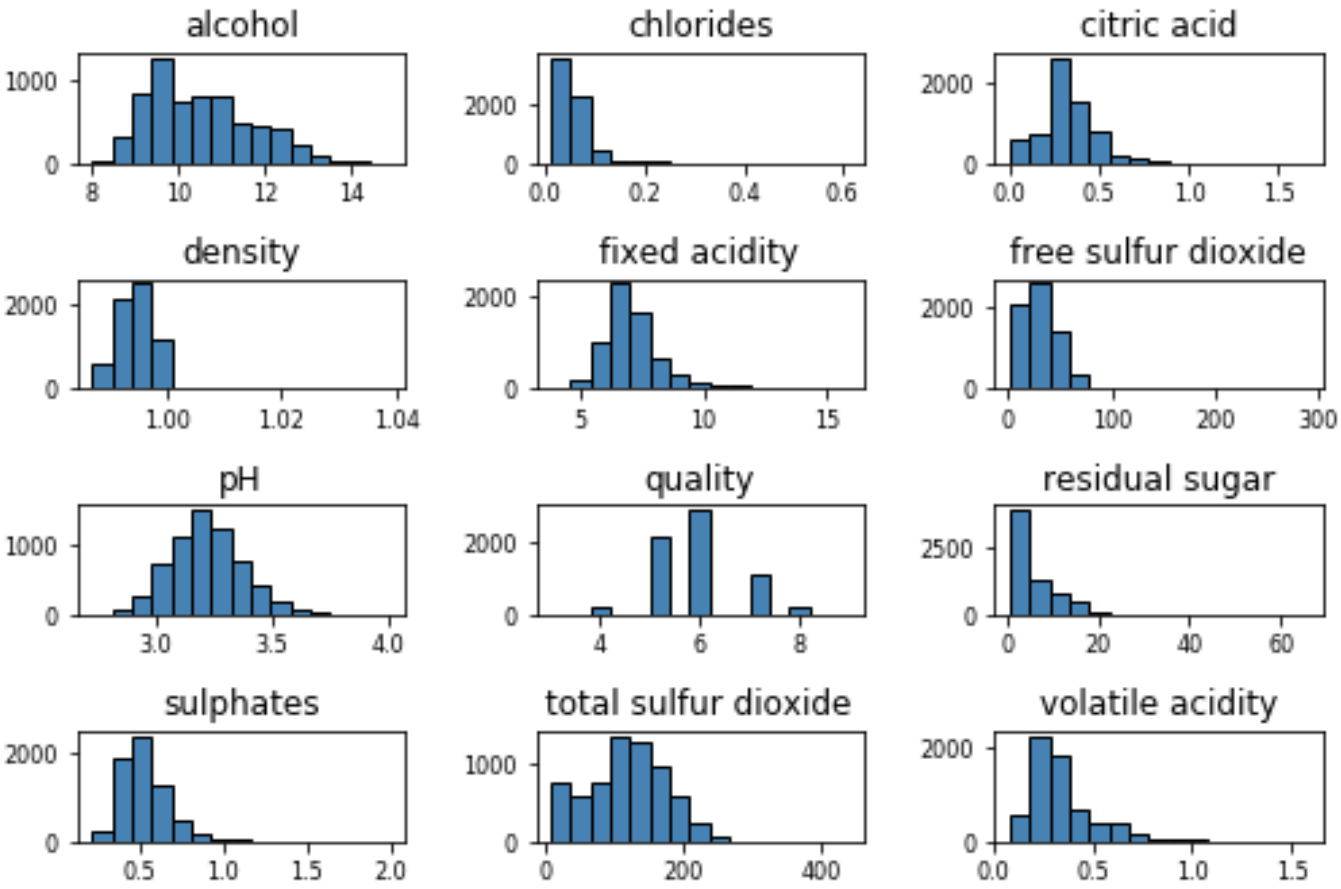
Fact Constellation Schema

- A fact constellation can have multiple fact tables that share many dimension tables.
- This type of schema can be viewed as a collection of stars, Snowflake and hence is called a galaxy schema or a fact constellation.
- The main disadvantage of fact constellation schemas is its more complicated design.

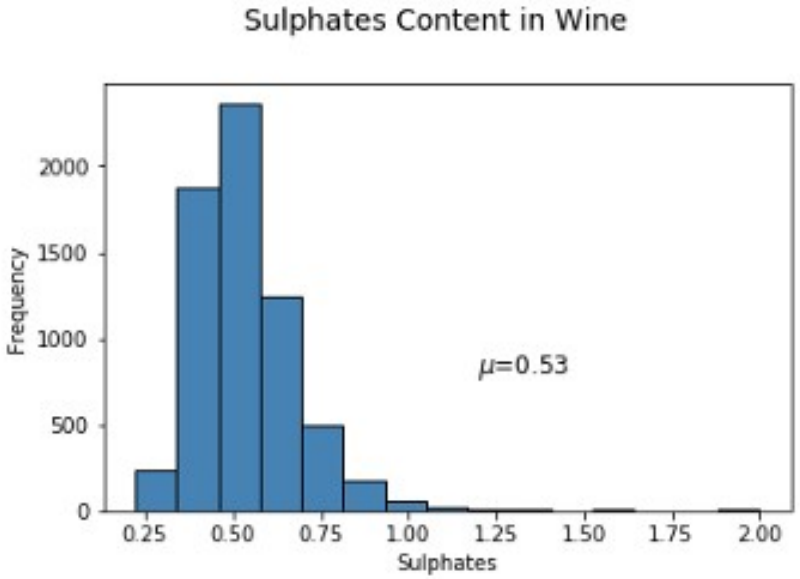
Fact Constellation Schema

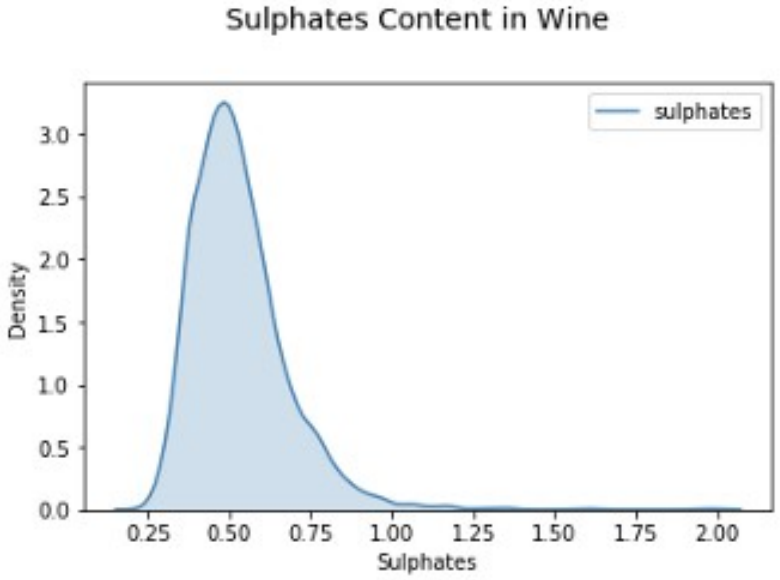


Visualize Multidimensional Data

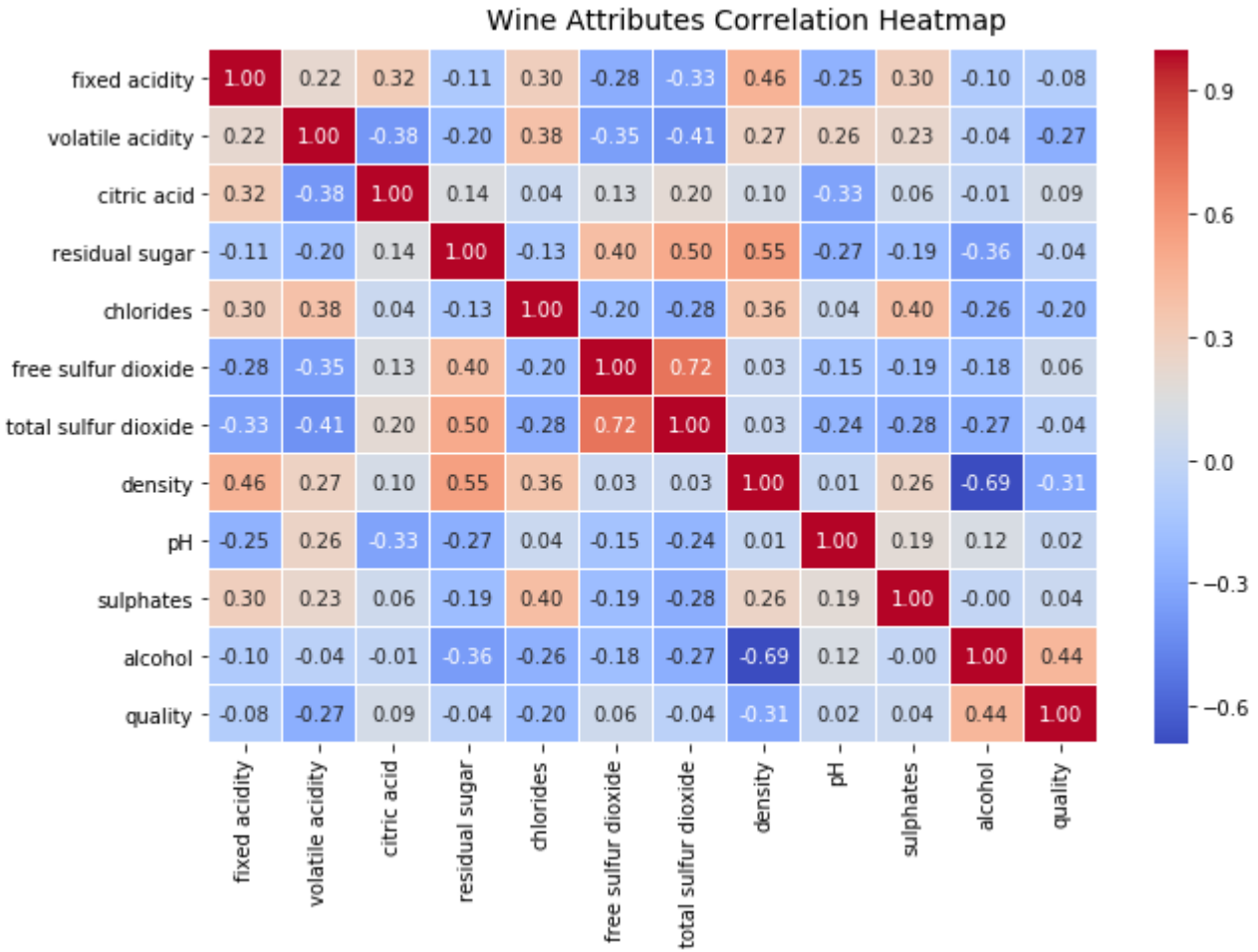


Visualize Multidimensional Data

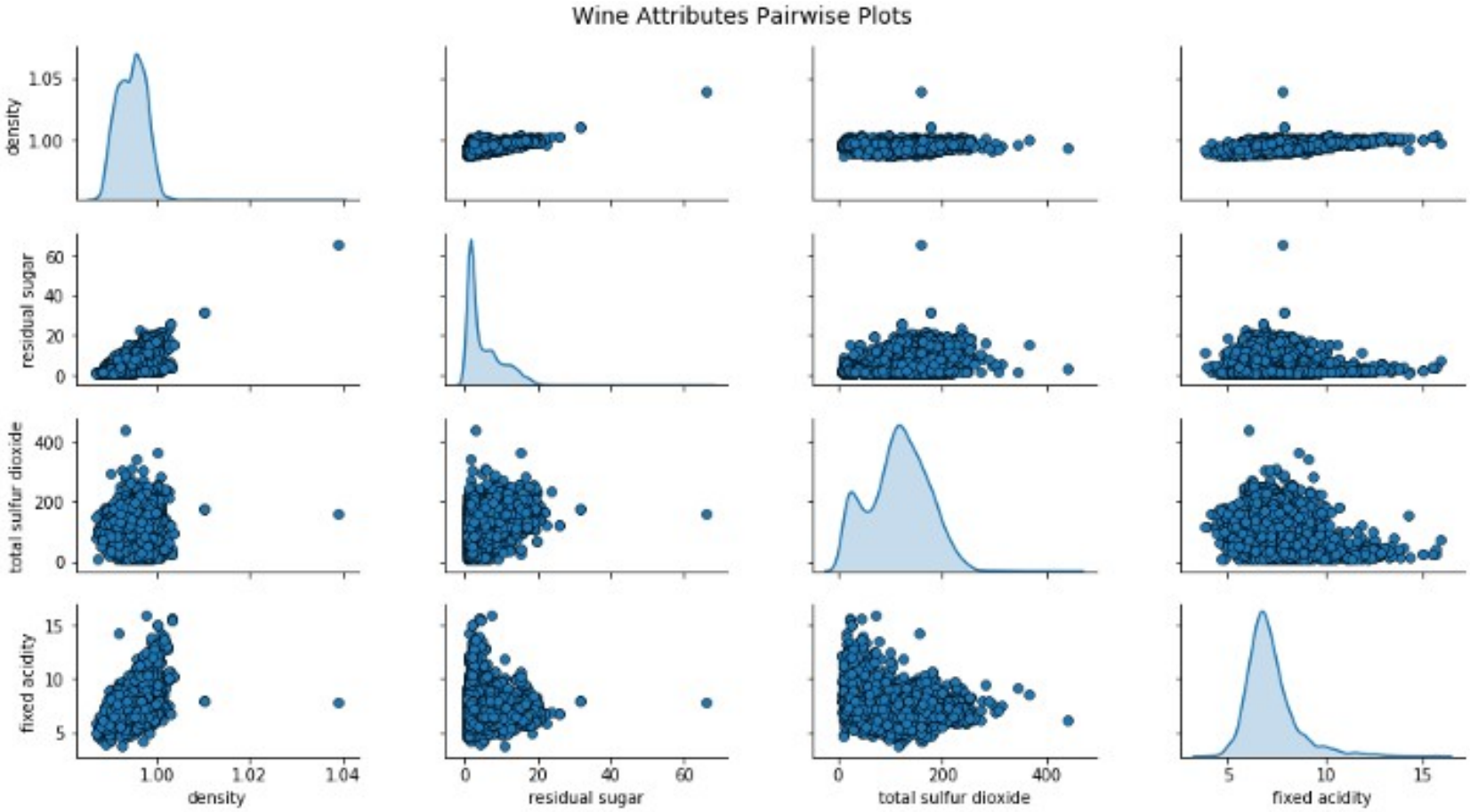




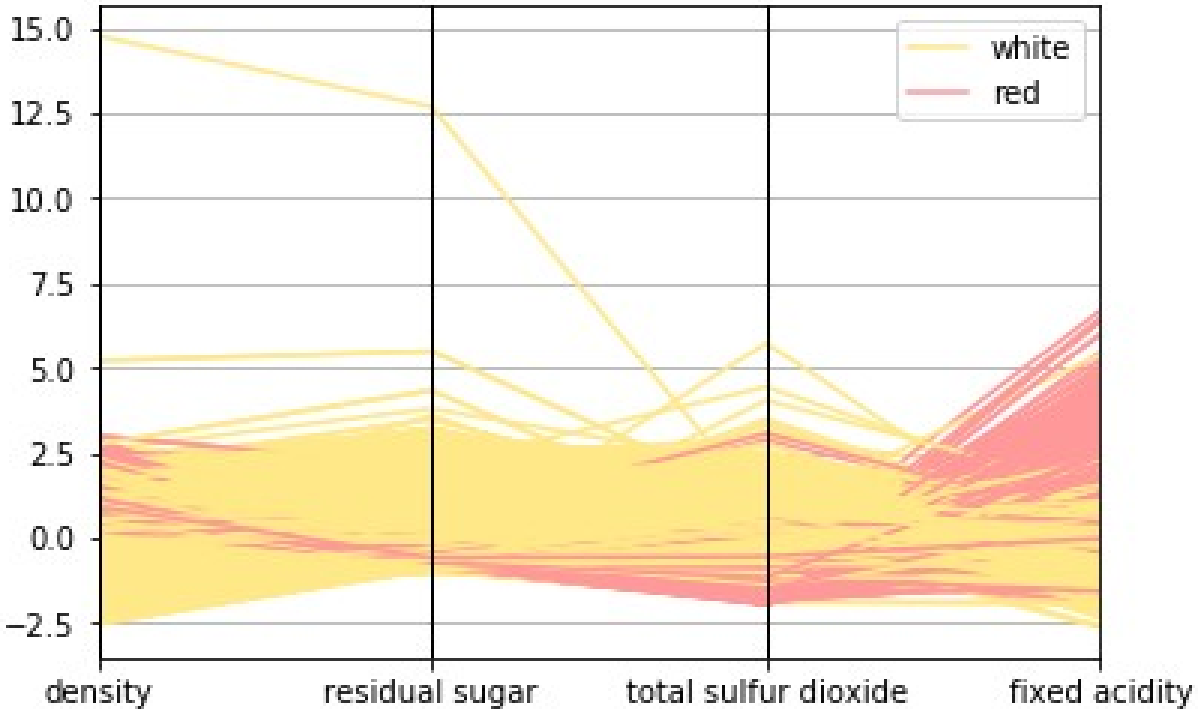
Visualize Multidimensional Data



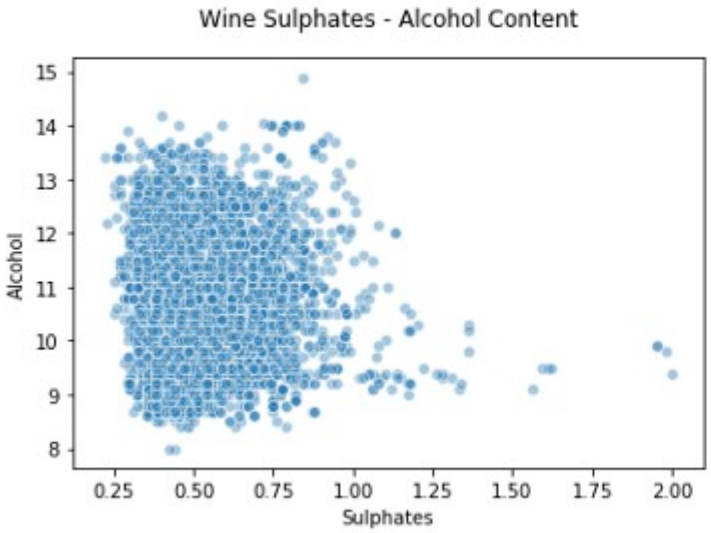
Visualize Multidimensional Data

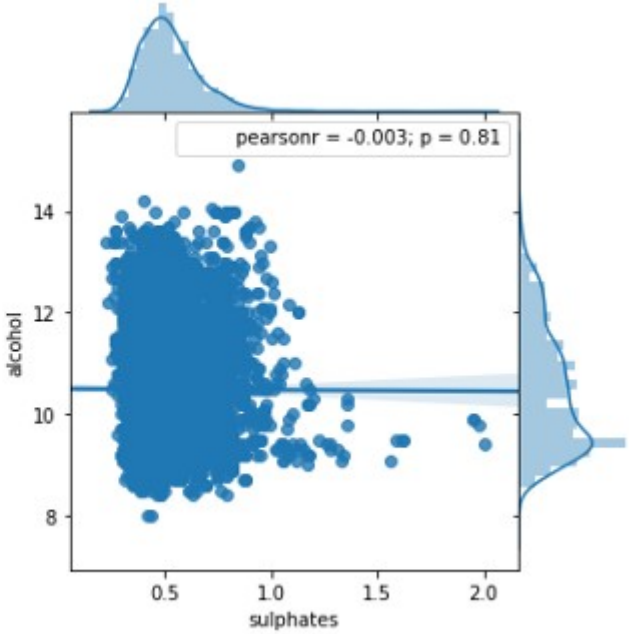


Visualize Multidimensional Data

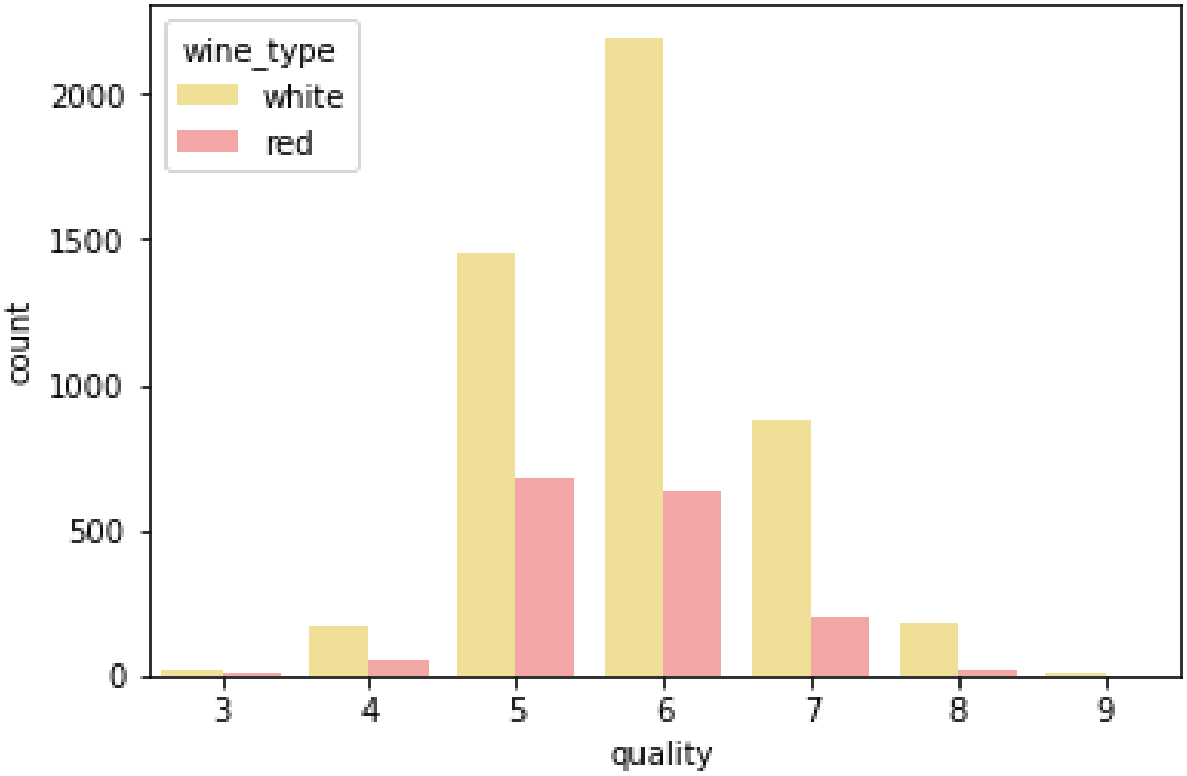


Visualize Multidimensional Data

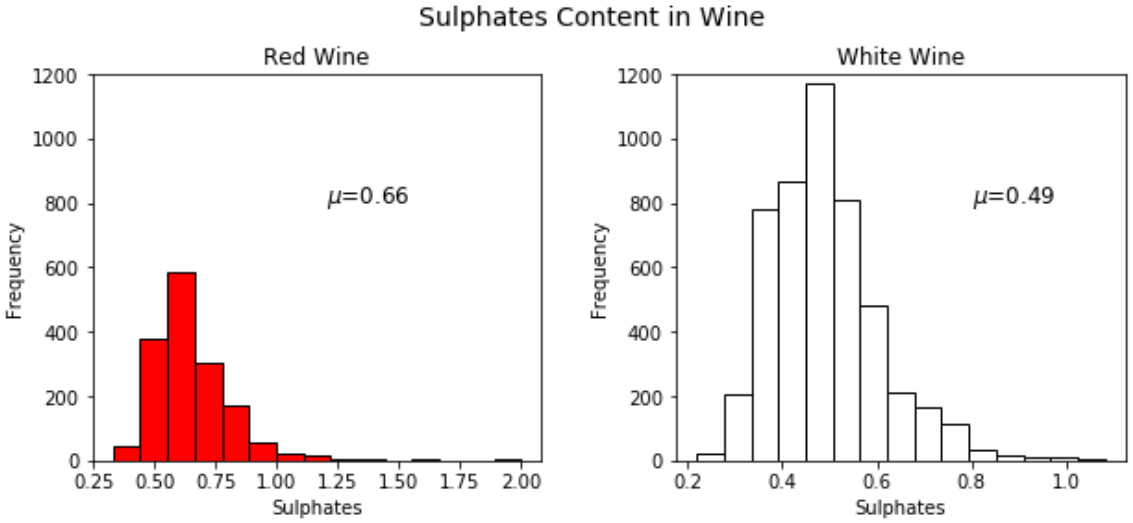


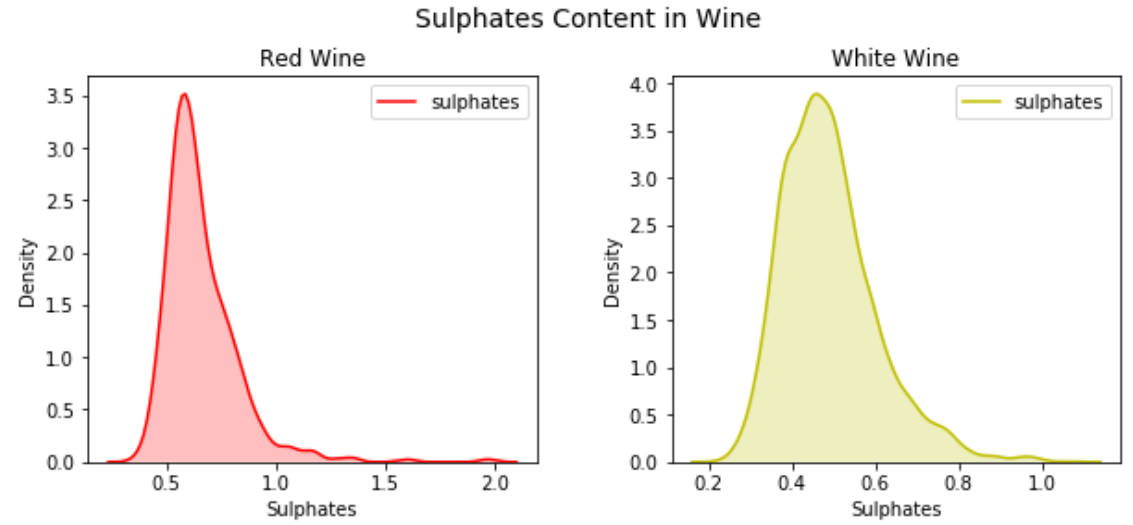


Visualize Multidimensional Data

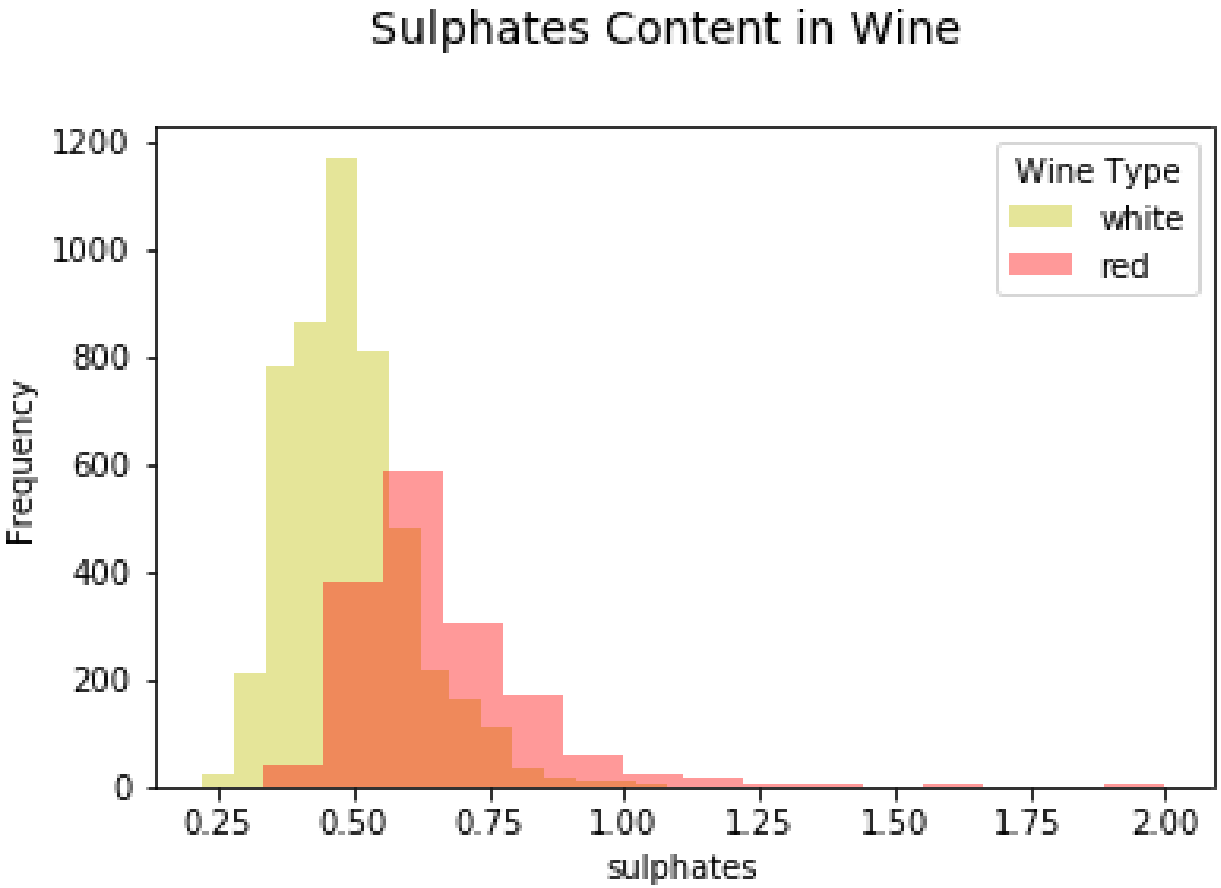


Visualize Multidimensional Data

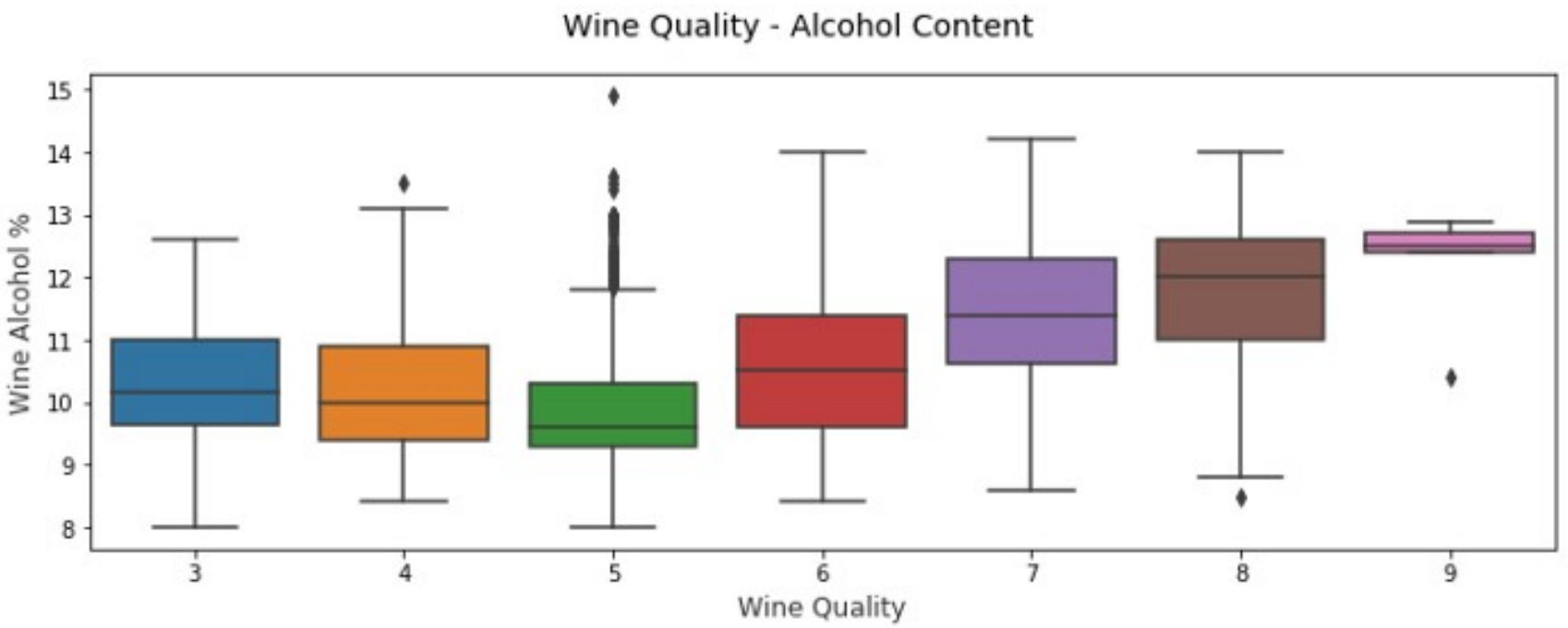




Visualize Multidimensional Data

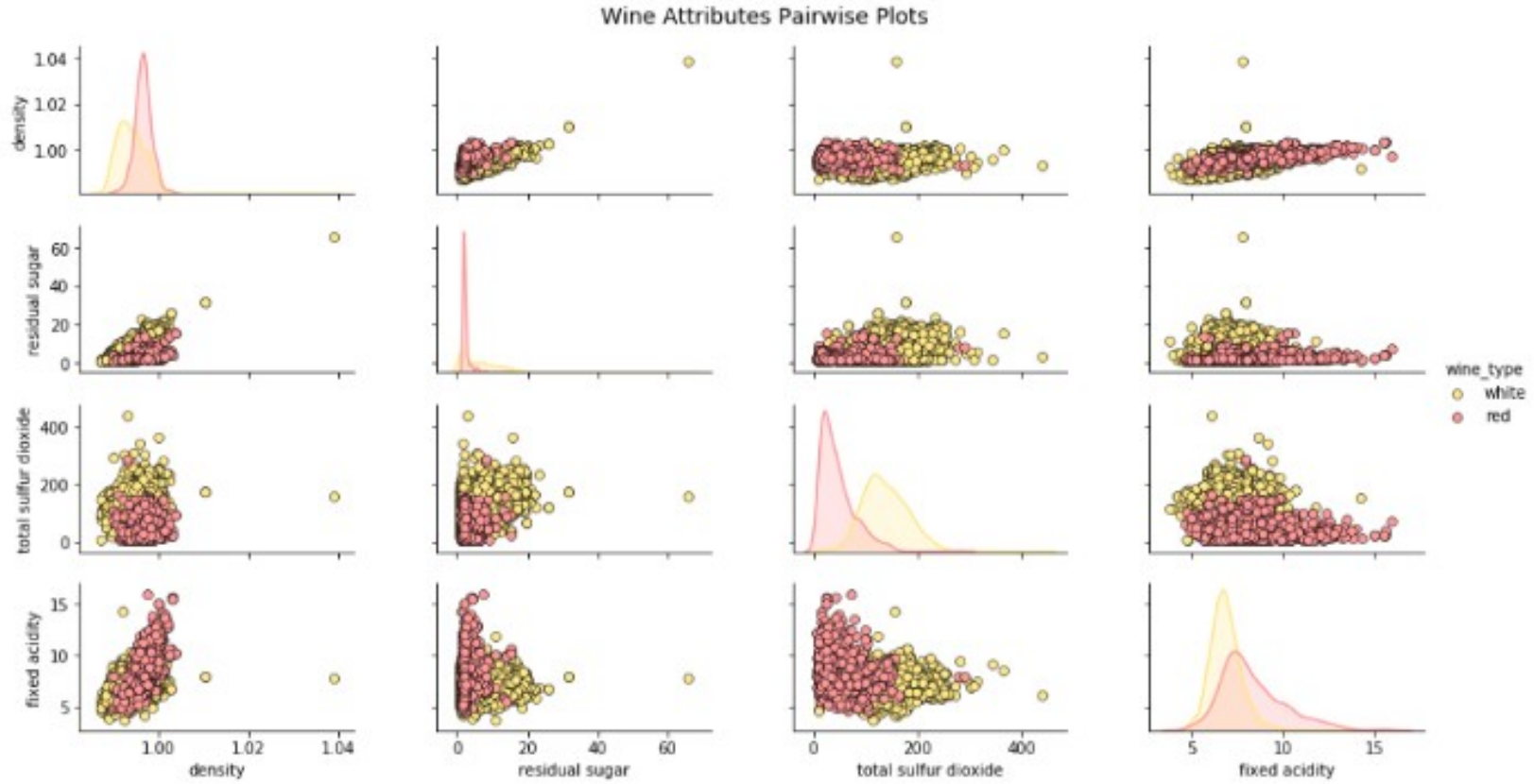


Visualize Multidimensional Data

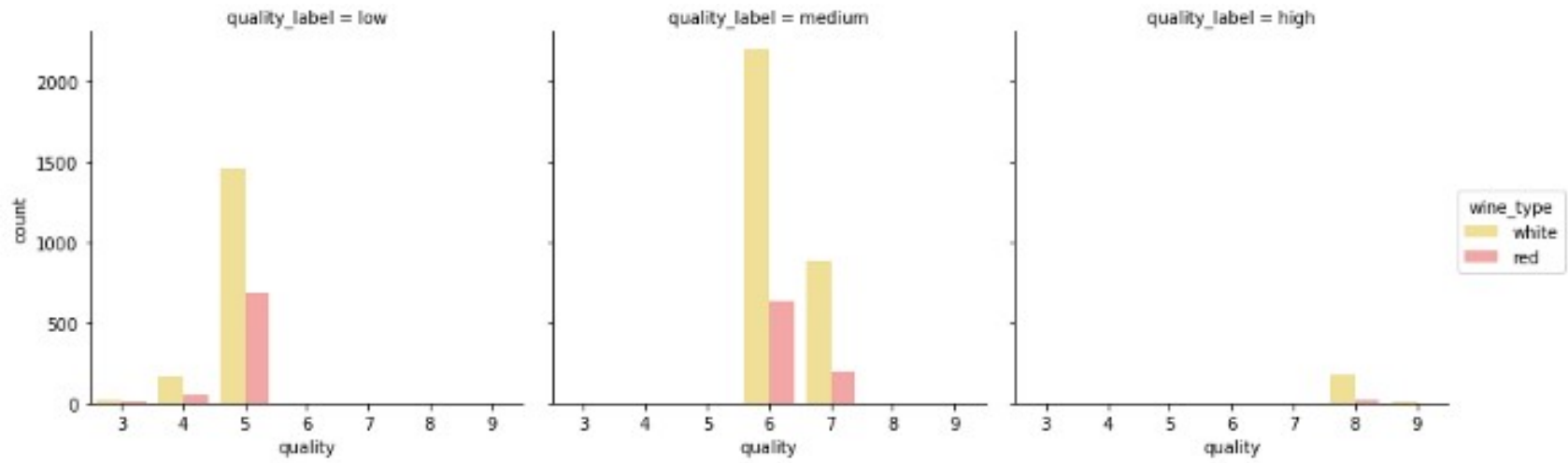


Visualize Multidimensional Data

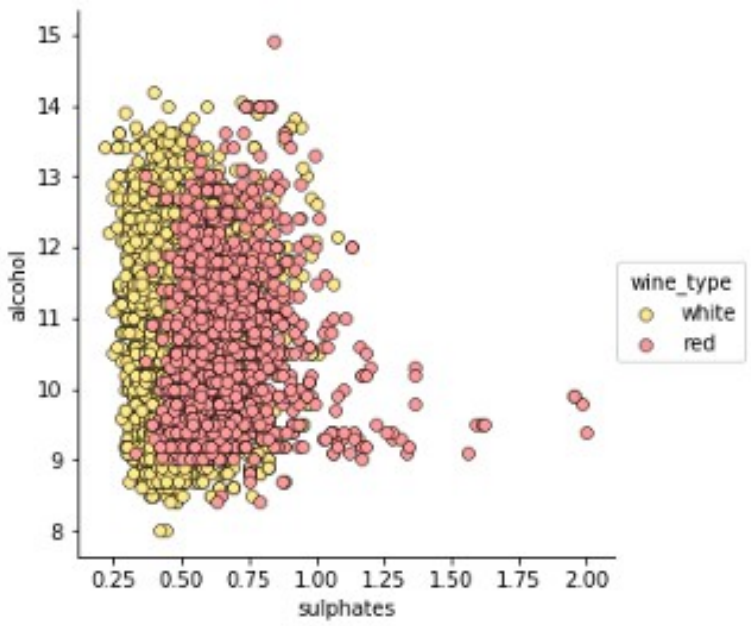
Visualize Multidimensional Data

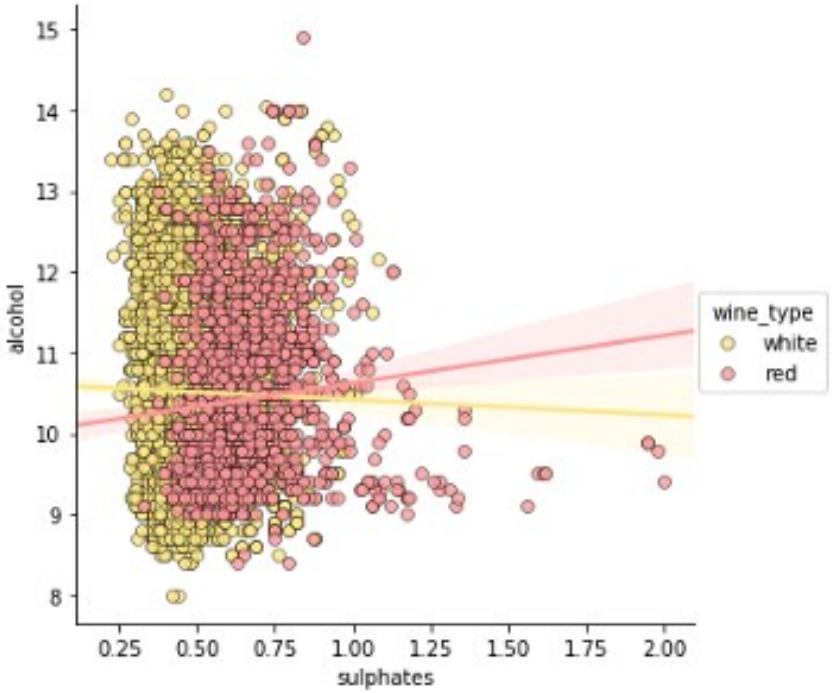


Visualize Multidimensional Data

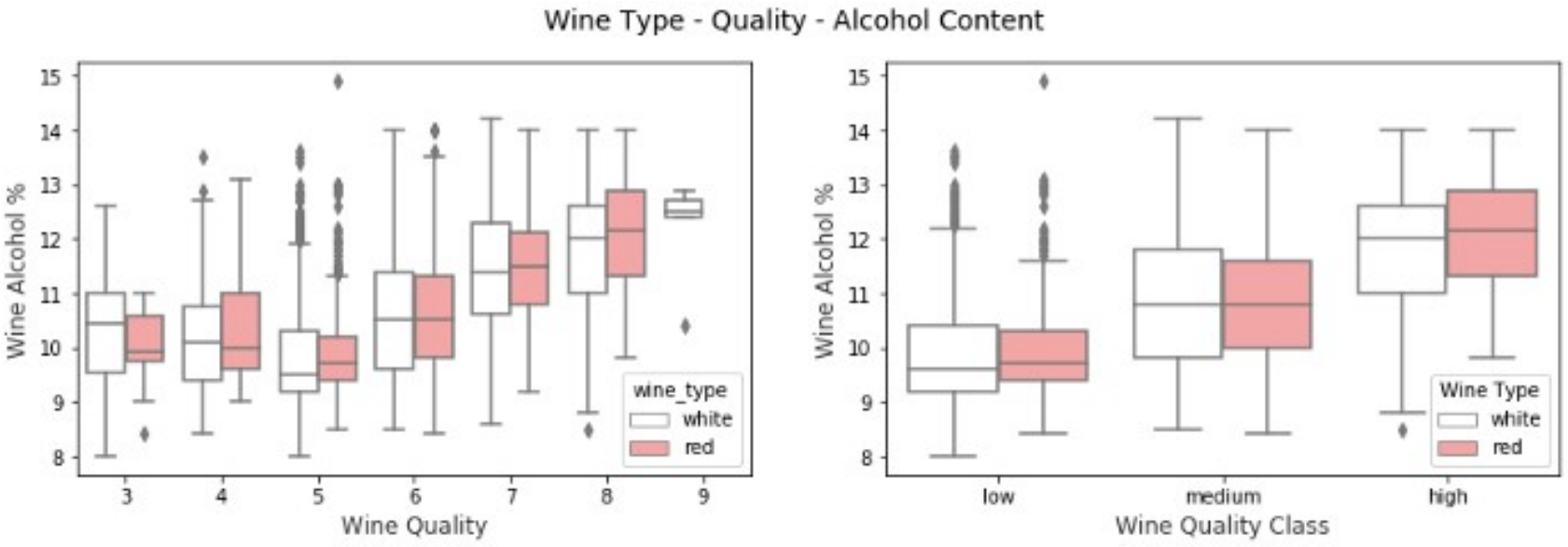


Visualize Multidimensional Data

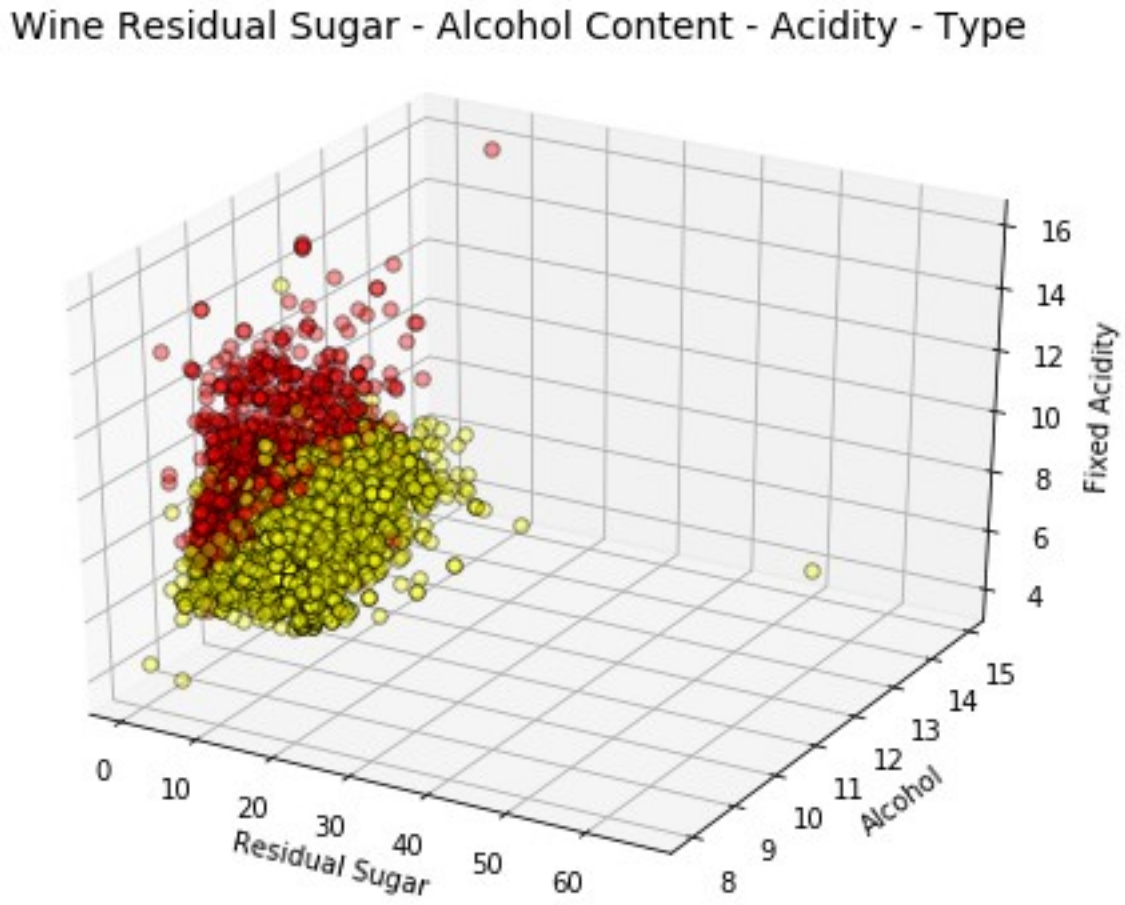




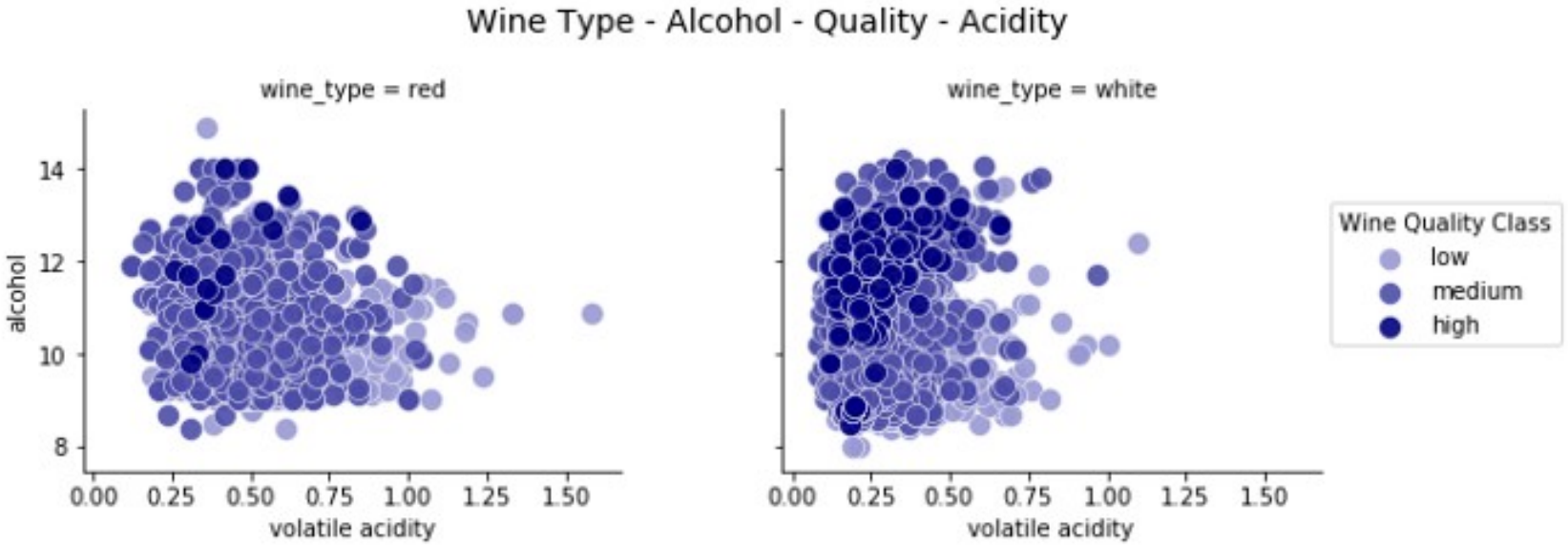
Visualize Multidimensional Data



Visualize Multidimensional Data



Visualize Multidimensional Data



Principal Component Analysis

- Large datasets are increasingly common and are often difficult to interpret.
- Principal component analysis (PCA) is a technique for reducing the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss.
- It does so by creating new uncorrelated variables that successively maximize variance.
- Finding such new variables, the principal components, reduces to solving an eigenvalue/eigenvector problem, and the new variables are defined by the dataset at hand, not a priori, hence making PCA an adaptive data analysis technique.

Dimensionality Reduction

- Dimensionality reduction or dimension reduction is the process of reducing the number of random variables under consideration by obtaining a set of principal variables.
- It can be divided into feature selection and feature extraction.
  - Feature selection approaches try to find a subset of the original variables (also called features or attributes).
  - Feature projection or Feature extraction transforms the data in the high-dimensional space to a space of fewer dimensions.

Large Dimensions

- Large number of features in the dataset is one of the factors that affect both the training time as well as accuracy of machine learning models. You have different options to deal with huge number of features in a dataset.
  - Try to train the models on original number of features, which take days or weeks if the number of features is too high.
  - Reduce the number of variables by merging correlated variables.
  - Extract the most important features from the dataset that are responsible for maximum variance in the output. Different statistical techniques are used for this purpose e.g. linear discriminant analysis, factor analysis, and principal component analysis.

Principal Component Analysis

- Principal component analysis, or PCA, is a statistical technique to convert high dimensional data to low dimensional data by selecting the most important features that capture maximum information about the dataset.
- The features are selected on the basis of variance that they cause in the output.
- The feature that causes highest variance is the first principal component. The feature that is responsible for second highest variance is considered the second principal component, and so on.
- It is important to mention that principal components do not have any correlation with each other.

Advantages of PCA

- The training time of the algorithms reduces significantly with less number of features.
- It is not always possible to analyze data in high dimensions. For instance if there are 100 features in a dataset. Total number of scatter plots required to visualize the data would be 100(100-1)2 = 4950. Practically it is not possible to analyze data this way.

Normalization of features

- It is imperative to mention that a feature set must be normalized before applying PCA. For instance if a feature set has data expressed in units of Kilograms, Light years, or Millions, the variance scale is huge in the training set. If PCA is applied on such a feature set, the resultant loadings for features with high variance will also be large. Hence, principal components will be biased towards features with high variance, leading to false results.
- Finally, the last point to remember before we start coding is that PCA is a statistical technique and can only be applied to numeric data. Therefore, categorical features are required to be converted into numerical features before PCA can be applied.

Steps in PCA

- Standardization
- Covariance Matrix Computation
- Computer Eigen vector and eigen values
- Feature vector
- Recast the Data Along the Principal Components Axes

Standardization

- The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.
- More specifically, the reason why it is critical to perform standardization prior to PCA, is that the latter is quite sensitive regarding the variances of the initial variables.
- That is, if there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges

Standardization

- Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable.

$$z = \frac{value - mean}{standard\ deviation}$$

- Once the standardization is done, all the variables will be transformed to the same scale.

Covariance Matrix Computation

- The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them.
- Because sometimes, variables are highly correlated in such a way that they contain redundant information.
- So, in order to identify these correlations, we compute the covariance matrix.

Covariance Matrix Computation

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

Covariance Matrix Computation

- What do the covariances that we have as entries of the matrix tell us about the correlations between the variables?
- It's actually the sign of the covariance that matters :
  - if positive then : the two variables increase or decrease together (correlated)
  - if negative then : One increases when the other decreases (Inversely correlated)
- Now, that we know that the covariance matrix is not more than a table that summaries the correlations between all the possible pairs of variables

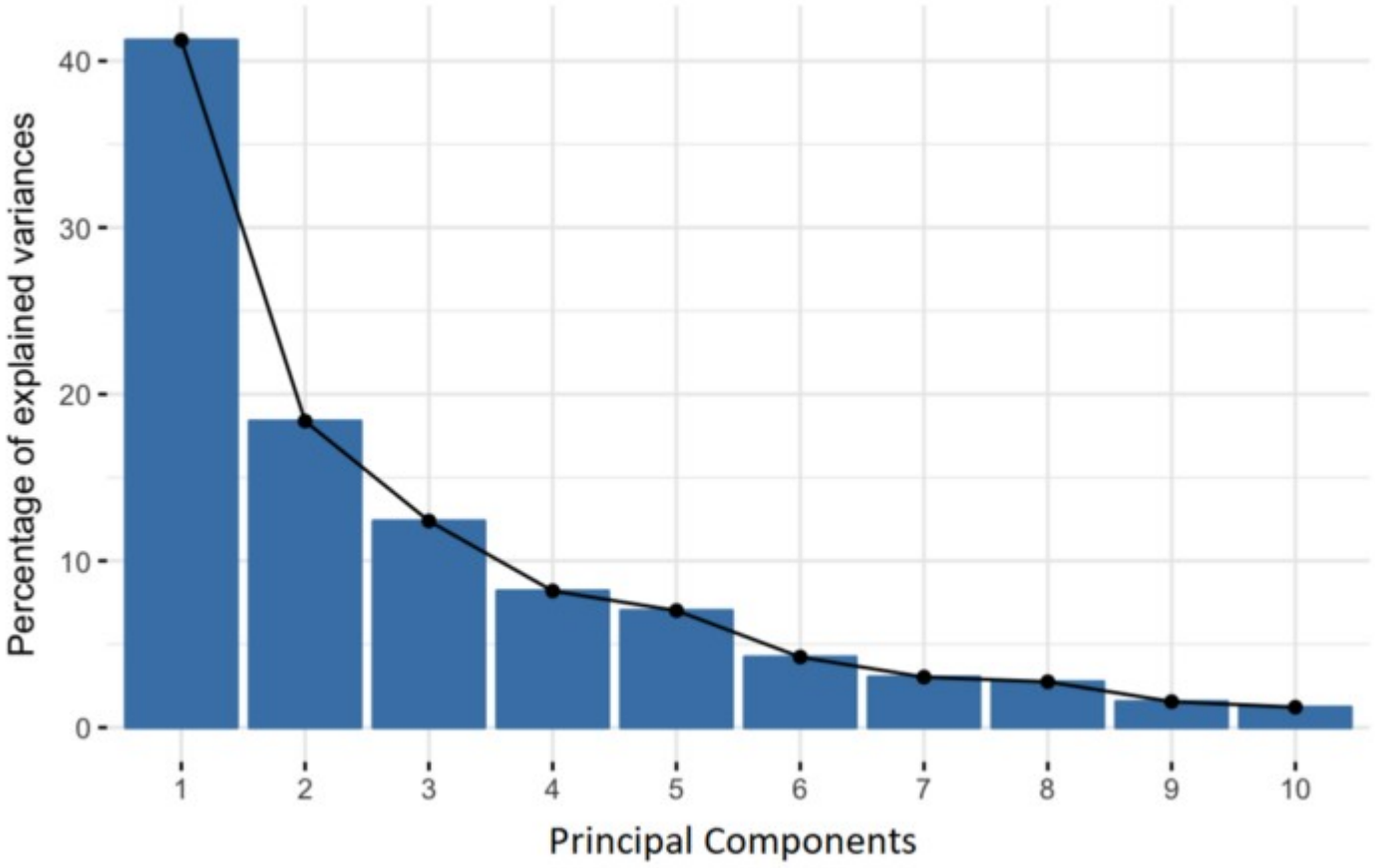
Eigenvector and eigenvalues

- Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the principal components of the data.
- Before getting to the explanation of these concepts, let’s first understand what do we mean by principal components.

Eigenvector and eigenvalues

- Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables.
- These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components.
- So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on, until having something like shown in the scree plot below.

Eigenvector and eigenvalues



Principal Components

- Organizing information in principal components this way, will allow you to reduce dimensionality without losing much information, and this by discarding the components with low information and considering the remaining components as your new variables.
- An important thing to realize here is that, the principal components are less interpretable and don’t have any real meaning since they are constructed as linear combinations of the initial variables.
- Geometrically speaking, principal components represent the directions of the data that explain a maximal amount of variance, that is to say, the lines that capture most information of the data.

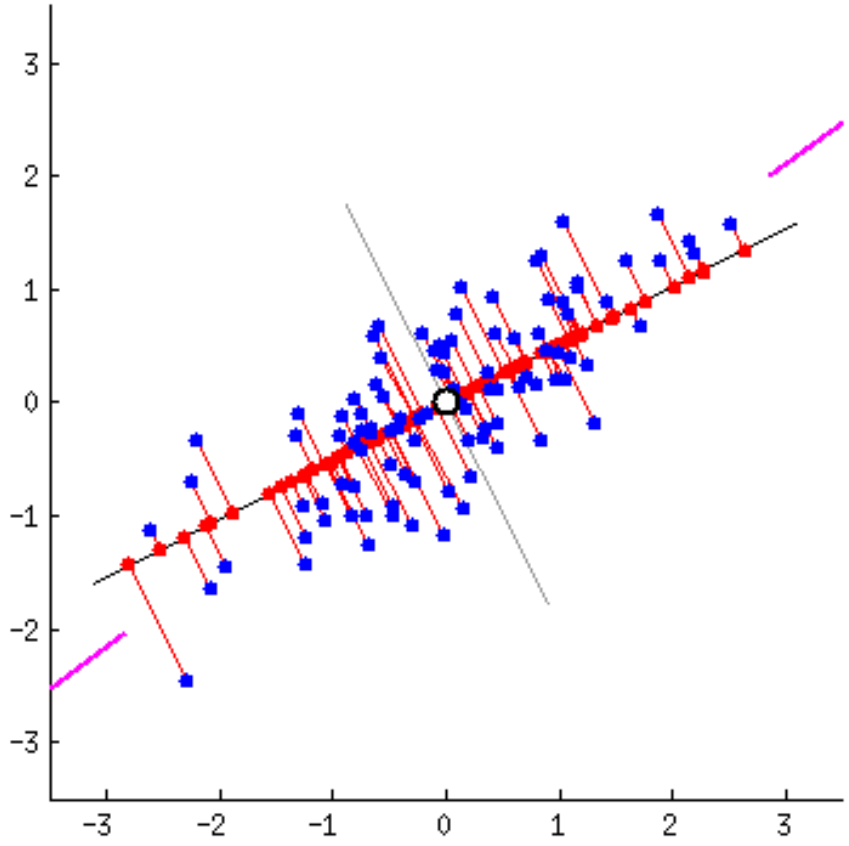
Principal Components

- As there are as many principal components as there are variables in the data, principal components are constructed in such a manner that the first principal component accounts for the largest possible variance in the data set.

Principal Components

- For example, let’s assume that the scatter plot of our data set is as shown below, can we guess the first principal component ?
- Yes, it’s approximately the line that matches the purple marks because it goes through the origin and it’s the line in which the projection of the points (red dots) is the most spread out.
- Or mathematically speaking, it’s the line that maximizes the variance (the average of the squared distances from the projected points (red dots) to the origin).

Principal Components



Example

- Let's suppose that our data set is 2-dimensional with 2 variables x,y and that the eigenvectors and eigenvalues of the covariance matrix are as follows:

$$v1 = \begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix} \qquad \lambda_1 = 1.284028$$

- $$v2 = \begin{bmatrix} -0.7351785 \\ 0.6778736 \end{bmatrix} \qquad \lambda_2 = 0.04908323$$

If we rank the eigenvalues in descending order, we get λ1>λ2, which means that the eigenvector that corresponds to the first principal component (PC1) is v1 and the one that corresponds to the second component (PC2) isv2.

Feature Vector

- As we saw in the previous step, computing the eigenvectors and ordering them by their eigenvalues in descending order, allow us to find the principal components in order of significance.
- In this step, what we do is, to choose whether to keep all these components or discard those of lesser significance (of low eigenvalues), and form with the remaining ones a matrix of vectors that we call Feature vector.
- So, the feature vector is simply a matrix that has as columns the eigenvectors of the components that we decide to keep.
- This makes it the first step towards dimensionality reduction, because if we choose to keep only p eigenvectors (components) out of n, the final data set will have only p dimensions.

Example:

- Continuing with the example from the previous step, we can either form a feature vector with both of the eigenvectors v1 and v2:

$$\begin{bmatrix} 0.6778736 & -0.7351785 \\ 0.7351785 & 0.6778736 \end{bmatrix}$$

- Or discard the eigenvector v2, which is the one of lesser significance, and form a feature vector with v1 only:

$$\begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix}$$

- Discarding the eigenvector v2 will reduce dimensionality by 1, and will consequently cause a loss of information in the final data set.

Last step

- The aim is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components (hence the name Principal Components Analysis).
- This can be done by multiplying the transpose of the original data set by the transpose of the feature vector.

$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T$$

Clustering high dimensional data

- Clustering high-dimensional data is the cluster analysis of data with anywhere from a few dozen to many thousands of dimensions.
- Such high-dimensional spaces of data are often encountered in areas such as medicine, where DNA microarray technology can produce many measurements at once, and the clustering of text documents, where, if a word-frequency vector is used, the number of dimensions equals the size of the vocabulary.

Problems

- Multiple dimensions are hard to think in, impossible to visualize, and, due to the exponential growth of the number of possible values with each dimension, complete enumeration of all subspaces becomes intractable with increasing dimensionality. This problem is known as the curse of dimensionality.
- The concept of distance becomes less precise as the number of dimensions grows, since the distance between any two points in a given dataset converges. The discrimination of the nearest and farthest point in particular becomes meaningless:

$$\lim_{d \to \infty} \frac{dist_{\max} - dist_{\min}}{dist_{\min}} = 0$$

Problems

- A cluster is intended to group objects that are related, based on observations of their attribute's values. However, given a large number of attributes some of the attributes will usually not be meaningful for a given cluster.
- Given a large number of attributes, it is likely that some attributes are correlated. Hence, clusters might exist in arbitrarily oriented affine subspaces.

Solutions

- Subspace Clustering
- Projected Clustering
- Projection Based Clustering
- Correlation Clustering

Subspace Clustering

- Subspace clustering is an extension of traditional clustering that seeks to find clusters in different subspaces within a dataset.
- Often in high dimensional data, many dimensions are irrelevant and can mask existing clusters in noisy data.
- Feature selection removes irrelevant and redundant dimensions by analyzing the entire dataset.
- Subspace clustering algorithms localize the search for relevant dimensions allowing them to find clusters that exist in multiple, possibly overlapping subspaces.

Projected Clustering

- Projected clustering is the first, top-down partitioning projected clustering algorithm based on the notion of k-medoid clustering which was presented by Aggarwal (1999).
- It determines medoids for each cluster repetitively on a sample of data using a greedy hill climbing technique and then upgrades the results repetitively.
- Cluster quality in projected clustering is a function of average distance between data points and the closest medoid.
- Also, the subspace dimensionality is an input framework which generates clusters of alike sizes.

Projection Based Clustering

- Projection-based clustering is based on a nonlinear projection of high-dimensional data into a two-dimensional space.
- Typical projection-methods like t-distributed stochastic neighbor embedding (t-SNE), or neighbor retrieval visualizer (NerV) are used project data explicitly into two dimensions disregarding the subspaces of higher dimension than two and preserving only relevant neighborhoods in high-dimensional data.
- In the next step, the Delaunay graph between the projected points is calculated, and each vertex between two projected points is weighted with the high-dimensional distance between the corresponding high-dimensional data points.

Correlation Clustering

- Clustering is the problem of partitioning data points into groups based on their similarity.
- Correlation clustering provides a method for clustering a set of objects into the optimum number of clusters without specifying that number in advance

Thank you

This presentation is created using LibreOffice Impress 5.1.6.2, can be used freely as per GNU General Public License

@mitu_skillologies

/mITuSkillologies

@mitu_group

/company/mitu-skillologies

MITUSkillologies

Web Resources
https://mitu.co.in
http://tusharkute.com

contact@mitu.co.in

tushar@tusharkute.com