



INSTITUTE OF INFORMATICS AND COMMUNICATION

M.Sc. Informatics

(2022-24)

(Semester - IV)

Assignment

PAPER ITEC02: Data Analytics and Visualisation

SUBMITTED BY: -

NAME – Bharat Aggarwal

ROLL NO. – 2285756008/1507

SUBMITTED TO: -

Nitisha Aggarwal

1) Describe the importance of data exploration in the data science process.

In the realm of data science and machine learning, the significance of data exploration and analysis cannot be overstated. These preliminary steps set the foundation for successful project outcomes by providing crucial insights into the data, identifying patterns, uncovering relationships, and mitigating potential pitfalls.

Data exploration and analysis form the bedrock of any data science or machine learning endeavor. These initial steps are like a treasure hunt, where you unearth hidden gems of knowledge from your dataset.

These are some useful steps to highlights the importance of data exploration in the data science process:

1. Understanding the Data

The first step in data exploration is to gain a comprehensive understanding of the dataset. This involves examining the data's structure, format, and size, as well as identifying the variables and their types (numerical, categorical, etc.). Exploring the data distribution, summary statistics, and identifying missing values or outliers helps in forming initial hypotheses and creating a solid foundation for further analysis.

2. Data Cleaning and Preprocessing

Data exploration often reveals inconsistencies, errors, or missing values within the dataset. Cleaning and preprocessing the data involve handling missing values, removing duplicates, standardizing formats, and resolving inconsistencies. This step ensures the data is reliable, consistent, and suitable for subsequent analysis.

3. Exploratory Data Analysis (EDA)

EDA is a crucial step that involves visualizing and summarizing data to gain deeper insights. Through techniques such as histograms, scatter plots, box plots, and correlation matrices, EDA helps identify trends, patterns, and relationships within the data. It allows data scientists to make informed decisions about feature selection, identify potential biases, and refine the research questions or hypotheses.

4. Feature Engineering

Feature engineering involves transforming raw data into informative features that can enhance the performance of machine learning models. This step may include handling categorical variables, scaling numerical features, creating new features through mathematical operations, or applying domain-specific knowledge. Effective feature engineering can significantly improve model accuracy and generalization.

5. Statistical Analysis

Statistical analysis techniques, such as hypothesis testing and significance testing, help validate assumptions, determine statistical relationships, and identify factors that influence the target variable. These tests provide evidence for decision-making, model selection, and assessing the significance of the data's findings.

6. Model Selection and Validation

Based on the insights gained from data exploration and analysis, appropriate machine learning models can be selected. The chosen models should align with the project's objectives and the characteristics of the dataset. Model performance should be validated using suitable evaluation metrics, such as accuracy, precision, recall, or F1-score. This step helps ensure that the selected model performs well on unseen data and avoids overfitting or underfitting.

2) Explain the role of data visualization techniques in exploring and understanding data.

Data visualization is a graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

By bringing data to life with insightful plots and charts, data visualization techniques are vital in decision-making processes. Whether it's data analysts breaking down their findings to non-technical stakeholders, data scientists performing A/B tests for marketing purposes, or machine learning engineers

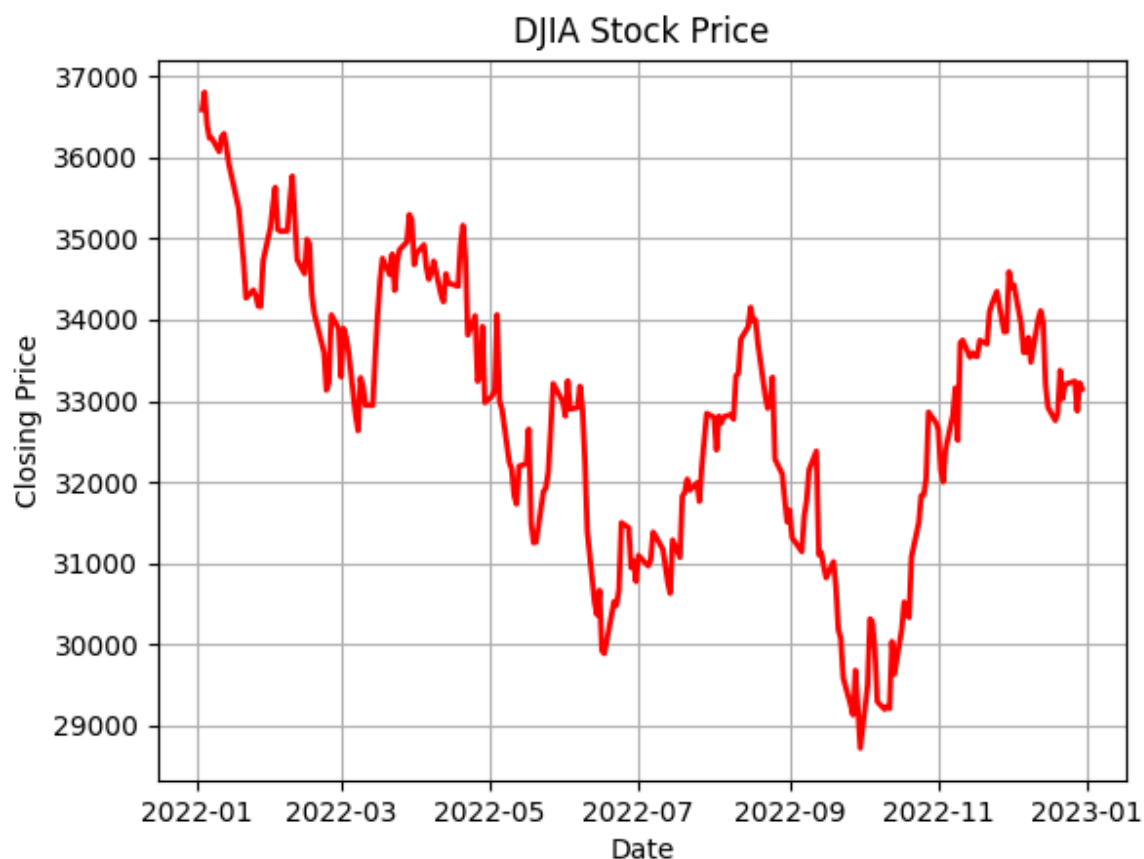
explaining potential bias in complex large language models like ChatGPT, data visualization is the key to moving from data insights to decision-making.

Key Data Visualization Techniques

Let's now examine the most popular data visualization techniques!

- **Line plots**

One of the most used visualizations, line plots are excellent at tracking the evolution of a variable over time. They are normally created by putting a time variable on the x-axis and the variable you want to analyze on the y-axis. For example, the line plot below shows the evolution of the DJIA Stock Price during 2022.



- **Bar plots**

A bar chart ranks data according to the value of multiple categories. It consists of rectangles whose lengths are proportional to the value of each category. Bar charts are prevalent because they are easy to read. Businesses commonly use bar charts to make comparisons, like comparing the market share of different brands or the revenue of different regions.

There are multiple types of bar charts, each suited for a different purpose, including vertical bar plots, horizontal bar plots, and clustered bar plots.

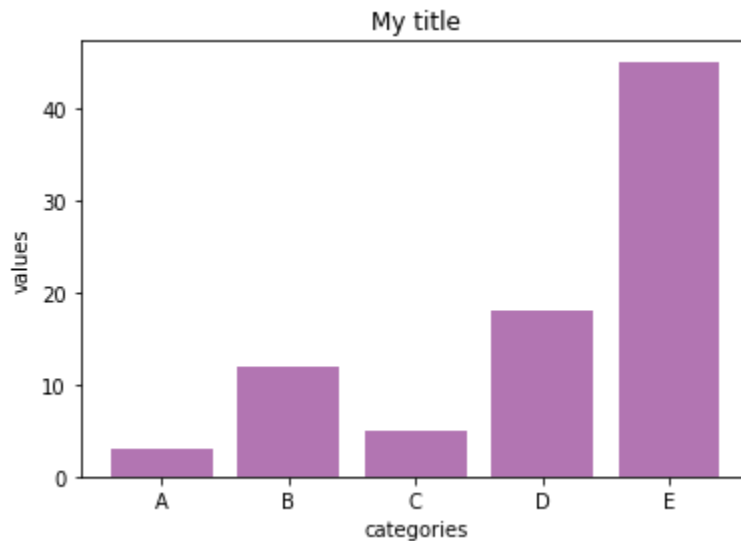


Fig. Vertical Bar Plot

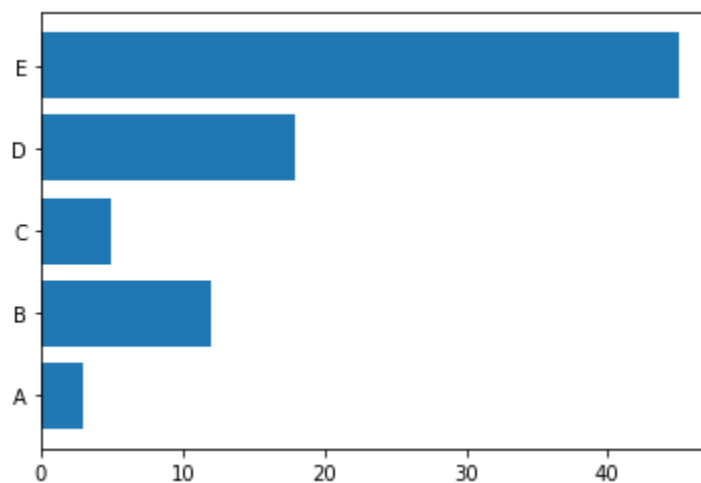


Fig. Horizontal Bar Plot

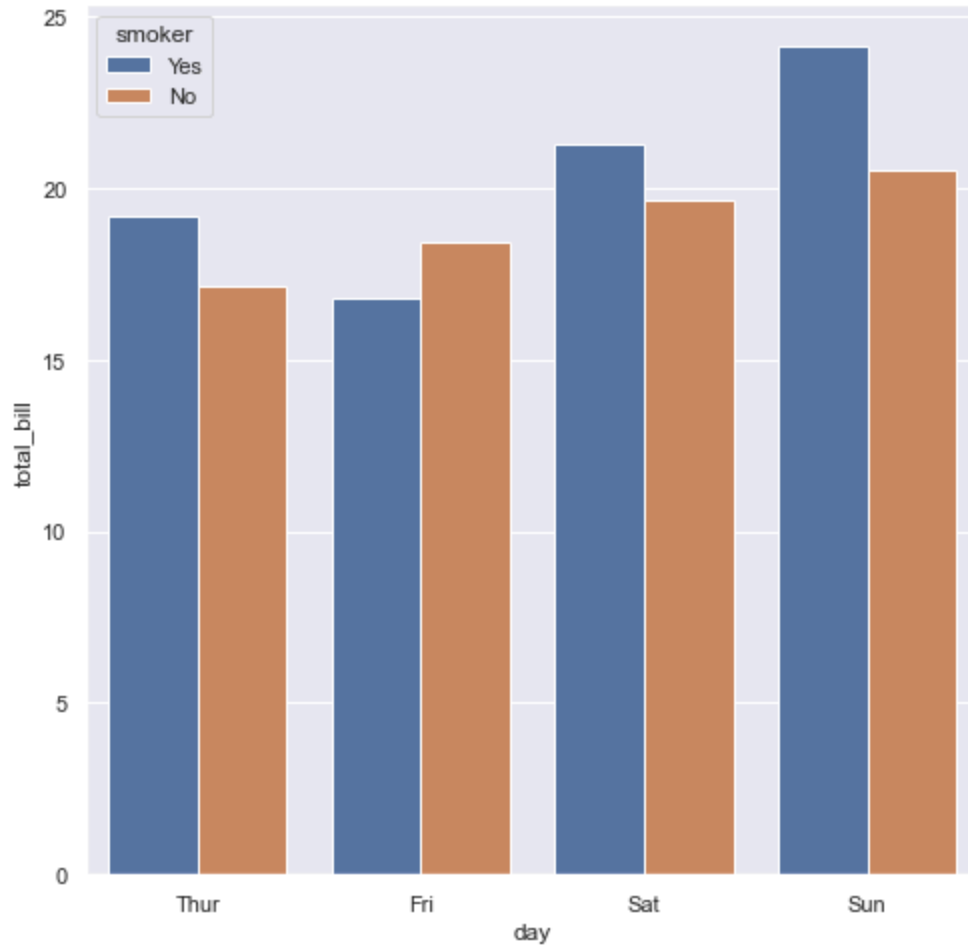


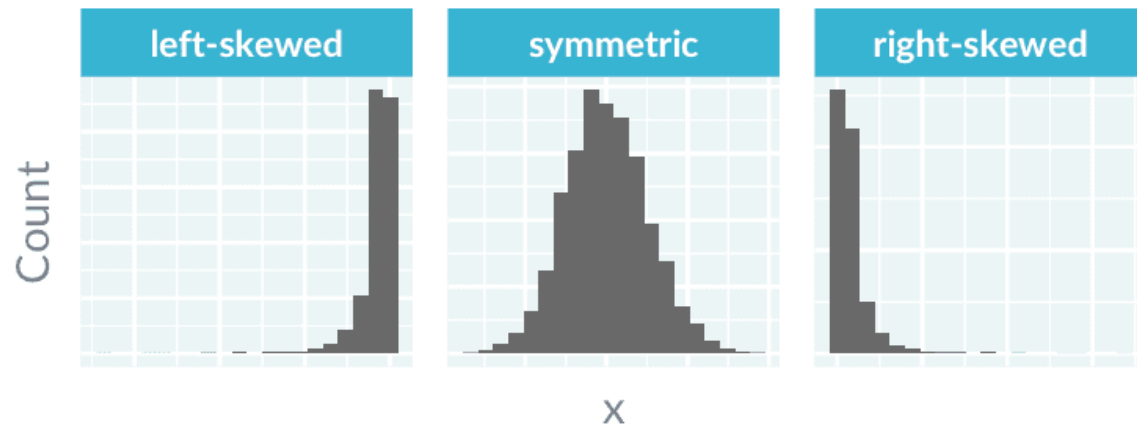
Fig. Clustered Bar Plot

- **Histograms**

Histograms are one of the most popular visualizations to analyze the distribution of data. They show the numerical variable's distribution with bars.

To build a histogram, the numerical data is first divided into several ranges or bins, and the frequency of occurrence of each range is counted. The horizontal axis shows the range, while the vertical axis represents the frequency or percentage of occurrences of a range.

Histograms immediately showcase how a variable's distribution is skewed or where it peaks.



- **Box and whisker plots**

Another great plot to summarize the distribution of a variable is boxplots. Boxplots provide an intuitive and compelling way to spot the following elements:

- **Median.** The middle value of a dataset where 50% of the data is less than the median and 50% of the data is higher than the median.
- **The upper quartile.** The 75th percentile of a dataset where 75% of the data is less than the upper quartile, and 25% of the data is higher than the upper quartile.
- **The lower quartile.** The 25th percentile of a dataset where 25% of the data is less than the lower quartile and 75% is higher than the lower quartile.
- **The interquartile range.** The upper quartile minus the lower quartile
- **The upper adjacent value.** Or colloquially, the “maximum.” It represents the upper quartile plus 1.5 times the interquartile range.
- **The lower adjacent value.** Or colloquially, the “minimum.” It represents the lower quartile minus 1.5 times the interquartile range.
- **Outliers.** Any values above the “maximum” or below the “minimum.”

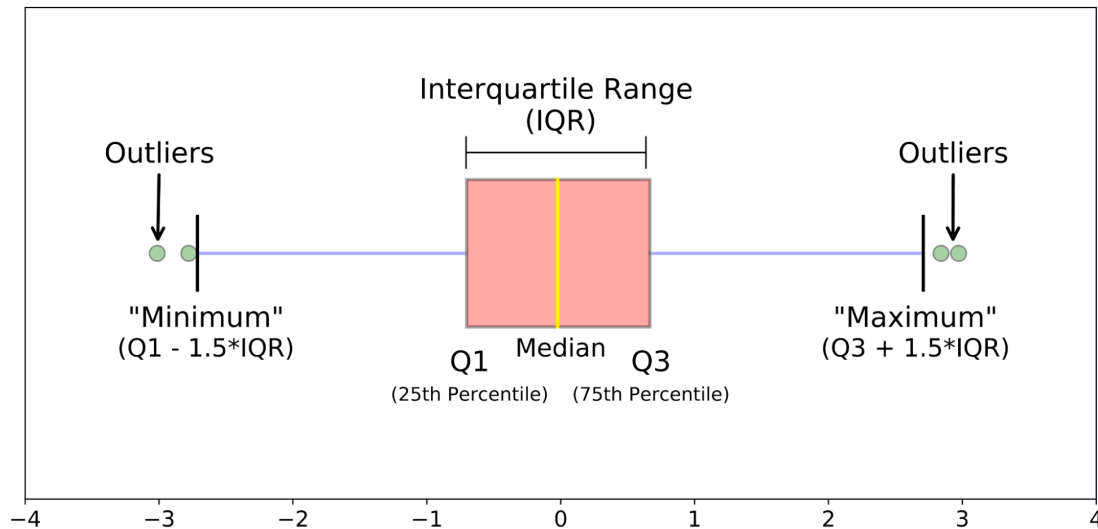
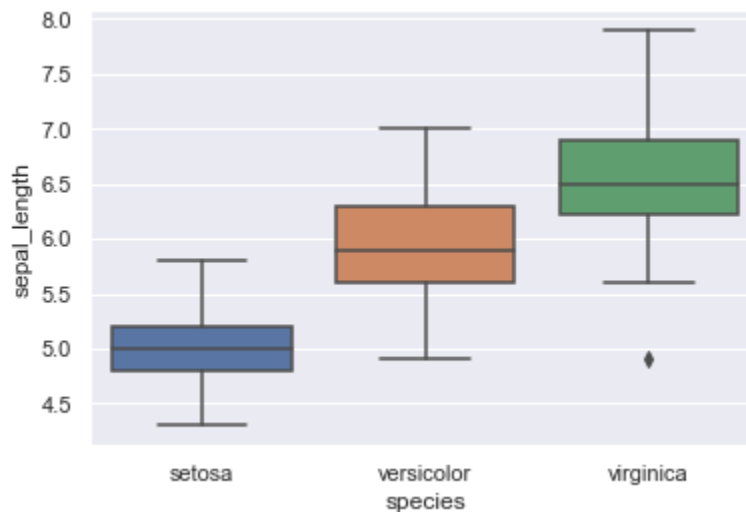


Fig. The anatomy of a box plot

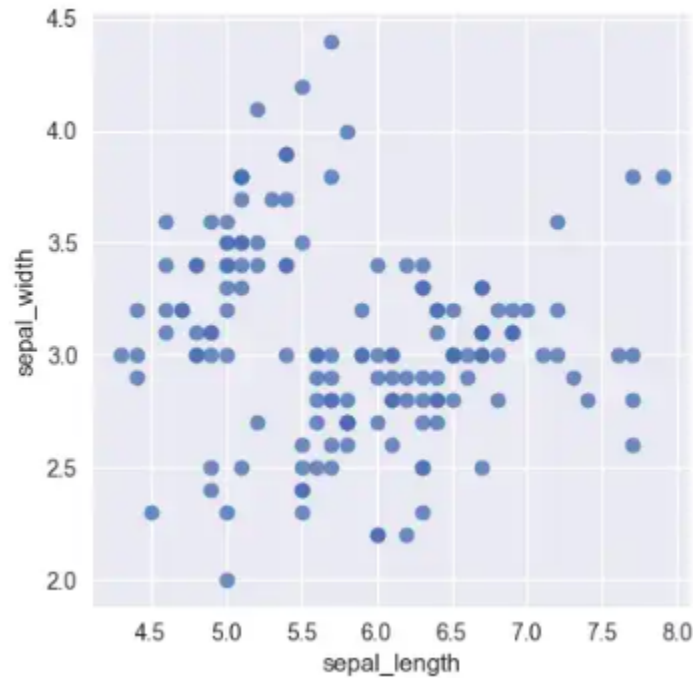
For example, the following seaborn-based boxplot shows the distribution of sepal length in three varieties of iris plants, drawing on the popular iris dataset.



- **Scatter plots**

Scatter plots are used to visualize the relationship between two continuous variables. Each point on the plot represents a single data point, and the position of the point on the x and y-axis represents the values of the two variables. It is often used in data exploration to understand the data and quickly surface potential correlations.

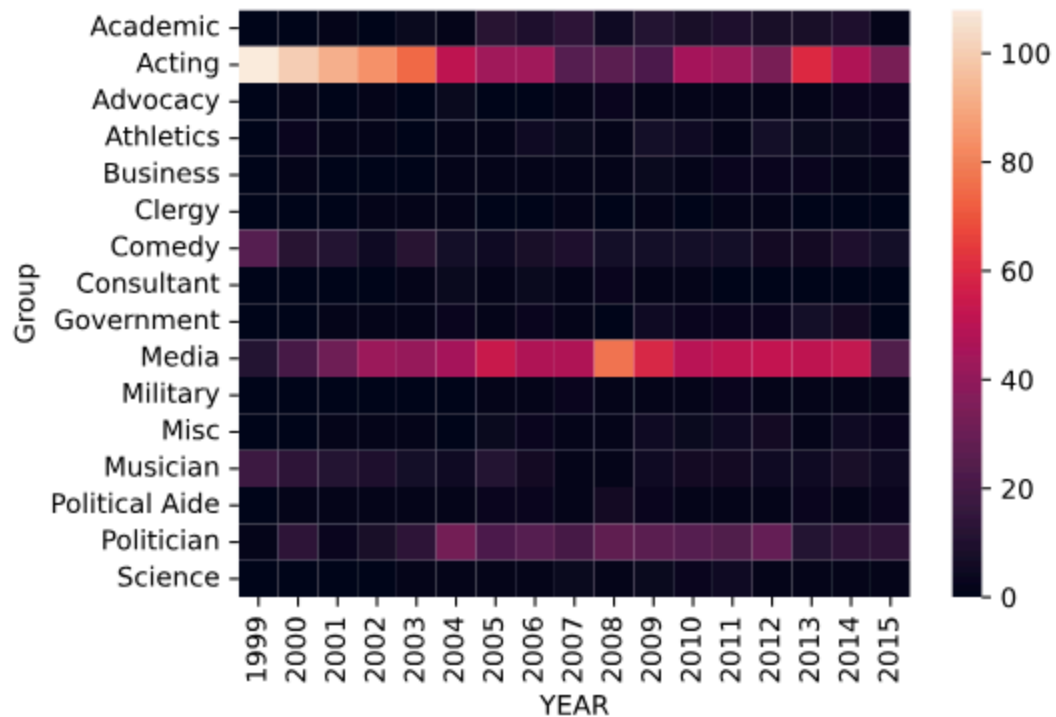
The following example takes again the iris dataset to plot the relationship between sepal width and sepal length.



- **Heat maps**

A heatmap is a common and beautiful matrix plot that can be used to graphically summarize the relationship between two variables. The degree of correlation between two variables is represented by a color code.

For example, this heat extracted from data of the occupation of the guests of the Daily Show during the 1999-2015 period. As expected, guests from the acting and media industries are the most frequent attendants.



3) Discuss common data preprocessing techniques such as missing value imputation, outlier detection, and feature scaling.

Data preprocessing is the transformation of raw data into a format that is more suitable and meaningful for analysis and model training. Data preprocessing plays a vital role in enhancing the quality and efficiency of ML models by addressing issues such as missing values, noise, inconsistencies, and outliers in the data.

Handling Missing Values

Missing values are a common issue in real-world data sets and can adversely affect the performance of ML models. To identify and deal with missing values:

- Use descriptive statistics or visualizations to identify columns/features with missing values. Common indicators of missing values include NaN (Not a Number) or NULL values.
- Determine the impact of missing values on your analysis or model. Consider the percentage of missing values in each column and their importance to the overall data set.

- If the percentage of missing values is small and those rows or columns are not critical, you can choose to remove them using methods like `dropna()` in pandas or similar functions in other tools.
- For numerical features, you can impute missing values using techniques like mean, median, or mode imputation (`fillna()` method in pandas). For categorical features, you can impute the most frequent category.

Handling Outliers

Outliers are data points that significantly differ from other observations in the data set and can skew statistical analysis or machine learning models.

To detect and handle outliers:

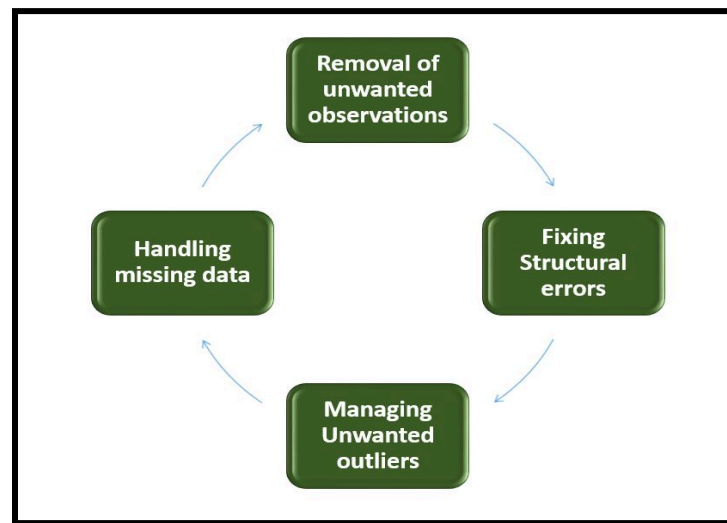
- Use box plots, histograms, or scatter plots to visualize the distribution of numerical features and identify potential outliers visually.
- Calculate summary statistics like mean, standard deviation, quartiles, and interquartile range (IQR). Outliers are often defined as data points that fall below $Q1 - 1.5 * IQR$ or above $Q3 + 1.5 * IQR$.
- In some cases, removing outliers can be appropriate, especially if they're due to data entry errors or anomalies. Use filtering techniques based on statistical thresholds to remove outliers.
- Apply transformations like log transformation, square root transformation, or Box-Cox transformation to make the data more normally distributed and reduce the impact of outliers.
- Consider using robust machine learning models that are less sensitive to outliers, such as support vector machines (SVM), Random Forests, or ensemble methods.

Feature Scaling:

- **Normalization:** Normalization scales the values of numerical features to a similar range, typically between 0 and 1. Common normalization techniques include Min-Max scaling and z-score normalization (standardization).
- **Standardization:** Standardization scales the features to have a mean of 0 and a standard deviation of 1. It is particularly useful when the features have different units or scales.
- **Robust Scaling:** Robust scaling scales features based on their median and interquartile range, making it less sensitive to outliers compared to Min-Max scaling or standardization.

4) Explain the process of data cleaning and quality assurance to ensure data integrity.

Data cleaning, also known as data cleansing or data preprocessing, 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.



- **Removal of Unwanted Observations:** Identify and eliminate irrelevant or redundant observations from the dataset. The step involves scrutinizing data entries for duplicate records, irrelevant information, or data points that do not contribute meaningfully to the analysis. Removing unwanted observations streamlines the dataset, reducing noise and improving the overall quality.
- **Fixing Structure errors:** Address structural issues in the dataset, such as inconsistencies in data formats, naming conventions, or variable types. Standardize formats, correct naming discrepancies, and ensure uniformity in data representation. Fixing structure errors enhances data consistency and facilitates accurate analysis and interpretation.
- **Managing Unwanted outliers:** Identify and manage outliers, which are data points significantly deviating from the norm. Depending on the context,

decide whether to remove outliers or transform them to minimize their impact on analysis. Managing outliers is crucial for obtaining more accurate and reliable insights from the data.

- **Handling Missing Data:** Devise strategies to handle missing data effectively. This may involve imputing missing values based on statistical methods, removing records with missing values, or employing advanced imputation techniques. Handling missing data ensures a more complete dataset, preventing biases and maintaining the integrity of analyses.
- **Data Validation and Verification:**
 - Perform data validation checks to ensure that the data meets predefined quality criteria and business rules.
 - Validate data integrity constraints, such as referential integrity, uniqueness constraints, and domain constraints.
 - Verify data consistency, accuracy, and completeness through manual inspection, automated checks, or comparison with external sources.
- **Documentation and Metadata Management:**
 - Document the entire data cleaning and quality assurance process, including steps taken, decisions made, and any modifications to the original data.
 - Maintain metadata describing the dataset's characteristics, provenance, lineage, and quality attributes to facilitate data governance and auditability.

5) Discuss strategies for handling categorical variables and encoding them for machine learning models.

Categorical data cannot typically be directly handled by machine learning algorithms, as most algorithms are primarily designed to operate with numerical data only. Therefore, before categorical features can be used as inputs to machine learning algorithms, they must be encoded as numerical values.

- **Ordinal Data:** The categories have an inherent order
- **Nominal Data:** The categories do not have an inherent order

In Ordinal data, while encoding, one should retain the information regarding the order in which the category is provided. Like in the above example the highest degree a person possesses, gives vital information about his qualification. The degree is an important feature to decide whether a person is suitable for a post or not.

While encoding Nominal data, we have to consider the presence or absence of a feature. In such a case, no notion of order is present. For example, the city a person lives in. For the data, it is important to retain where a person lives. Here, We do not have any order or sequence. It is equal if a person lives in Delhi or Bangalore.

Label Encoding or Ordinal Encoding

We use this categorical data encoding technique when the categorical feature is ordinal. In this case, retaining the order is important. Hence encoding should reflect the sequence.

In Label encoding, each label is converted into an integer value. We will create a variable that contains the categories representing the education qualification of a person.

Degree	
0	1
1	4
2	2
3	3
4	3
5	4
6	5
7	1
8	1


One Hot Encoding

We use this categorical data encoding technique when the features are nominal(do not have any order). In one hot encoding, for each level of a categorical feature, we create a new variable. Each category is mapped with a binary variable containing

either 0 or 1. Here, 0 represents the absence, and 1 represents the presence of that category.

These newly created binary features are known as Dummy variables. The number of dummy variables depends on the levels present in the categorical variable. This might sound complicated. Let us take an example to understand this better.

Suppose we have a dataset with a category animal, having different animals like Dog, Cat, Sheep, Cow, Lion. Now we have to one-hot encode this data.

Index	Animal	One-Hot code	Index	Dog	Cat	Sheep	Lion	Horse
0	Dog		0	1	0	0	0	0
1	Cat		1	0	1	0	0	0
2	Sheep		2	0	0	1	0	0
3	Horse		3	0	0	0	0	1
4	Lion		4	0	0	0	1	0

Dummy Encoding

The Dummy coding scheme is similar to one-hot encoding. This categorical data encoding method transforms the categorical variable into a set of binary variables (also known as dummy variables). In the case of one-hot encoding, for N categories in a variable, it uses N binary variables. The dummy encoding is a small improvement over one-hot-encoding. Dummy encoding uses N-1 features to represent N labels/categories.

While one-hot uses 3 variables to represent the data whereas dummy encoding uses 2 variables to code 3 categories.

Column	Code	Column	Code
A	100	A	10
B	010	B	01
C	001	C	00

One- Hot Coding Dummy Code

Effect Encoding

This encoding technique is also known as Deviation Encoding or Sum Encoding. Effect encoding is almost similar to dummy encoding, with a little difference. In dummy coding, we use 0 and 1 to represent the data but in effect encoding, we use three values i.e. 1,0, and -1.

The row containing only 0s in dummy encoding is encoded as -1 in effect encoding. In the dummy encoding example, the city Bangalore at index 4 was encoded as 0000. Whereas in effect encoding it is represented by -1-1-1-1.

	City	intercept	City_0	City_1	City_2	City_3
0	Delhi	1	1.0	0.0	0.0	0.0
1	Mumbai	1	0.0	1.0	0.0	0.0
2	Hyderabad	1	0.0	0.0	1.0	0.0
3	Chennai	1	0.0	0.0	0.0	1.0
4	Bangalore	1	-1.0	-1.0	-1.0	-1.0
5	Delhi	1	1.0	0.0	0.0	0.0
6	Hyderabad	1	0.0	0.0	1.0	0.0

Binary Encoding

Binary encoding is a combination of Hash encoding and one-hot encoding. In this encoding scheme, the categorical feature is first converted into numerical using an

ordinal encoder. Then the numbers are transformed into binary numbers. After that, the binary value is split into different columns.

Binary encoding works really well when there are a high number of categories. For example the cities in a country where a company supplies its products.

City		City_0	City_1	City_2	City_3
0	Delhi	0	0	0	1
1	Mumbai	0	0	1	0
2	Hyderabad	0	0	1	1
3	Chennai	0	1	0	0
4	Bangalore	0	1	0	1
5	Delhi	0	0	0	1
6	Hyderabad	0	0	1	1
7	Mumbai	0	0	1	0
8	Agra	0	1	1	0

Base N Encoding

In the numeral system, the Base or the radix is the number of digits or a combination of digits and letters used to represent the numbers. Another widely used system is binary i.e. the base is 2. It uses 0 and 1 i.e 2 digits to express all the numbers.

For Binary encoding, the Base is 2 which means it converts the numerical values of a category into its respective Binary form. If you want to change the Base of encoding scheme you may use Base N encoder. In the case when categories are more and binary encoding is not able to handle the dimensionality then we can use a larger base such as 4 or 8.

City		City_0	City_1	City_2
0	Delhi	0	0	1
1	Mumbai	0	0	2
2	Hyderabad	0	0	3
3	Chennai	0	0	4
4	Bangalore	0	1	0
5	Delhi	0	0	1
6	Hyderabad	0	0	3
7	Mumbai	0	0	2
8	Agra	0	1	1

6) Discuss common supervised learning algorithms such as linear regression, logistic regression, decision trees, and support vector machines.

Supervised Learning Algorithms

This algorithm consists of a target/outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using this set of variables, we generate a function that maps input data to desired outputs. The training process continues until the model achieves the desired level of accuracy on the training data. Examples of Supervised Learning: Regression, Decision tree, Random forest, KNN, Logistic Regression, etc.

1. Linear Regression

It is used to estimate real values (cost of houses, number of calls, total sales, etc.) based on a continuous variable(s). Here, we establish the relationship between independent and dependent variables by fitting the best line.

This best-fit line is known as the regression line and is represented by a linear equation

$$Y = a * X + b.$$

In this equation:

- Y – Dependent Variable
- a – Slope
- X – Independent variable
- b – Intercept

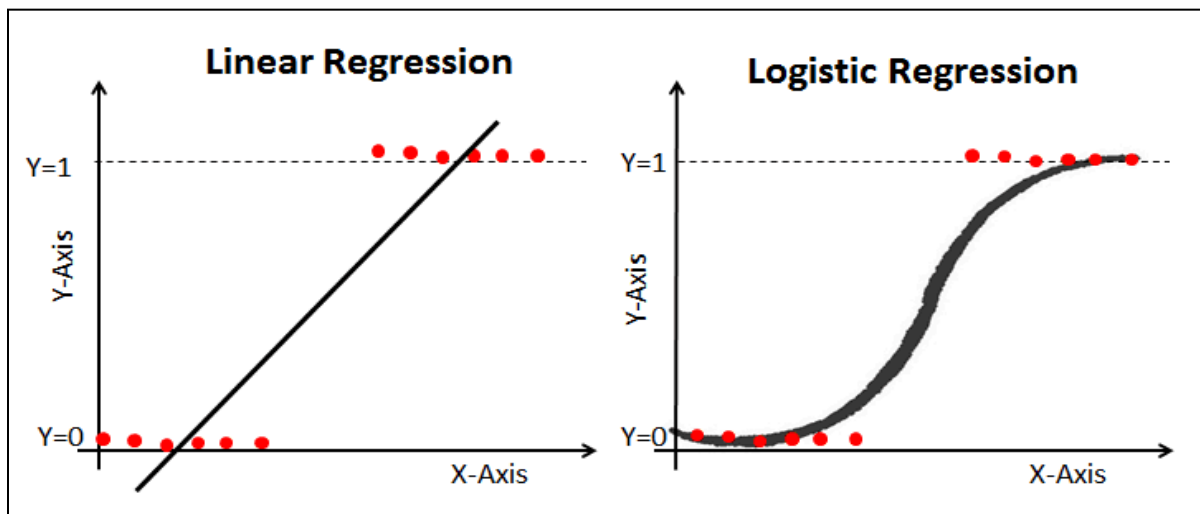
These coefficients a and b are derived based on minimizing the sum of the squared difference of distance between data points and the regression line.

Linear Regression is mainly of two types: Simple Linear Regression and Multiple Linear Regression. Simple Linear Regression is characterized by one independent variable. And, Multiple Linear Regression(as the name suggests) is characterized by multiple (more than 1) independent variables. While finding the best-fit line,

you can fit a polynomial or curvilinear regression. And these are known as polynomial or curvilinear regression.

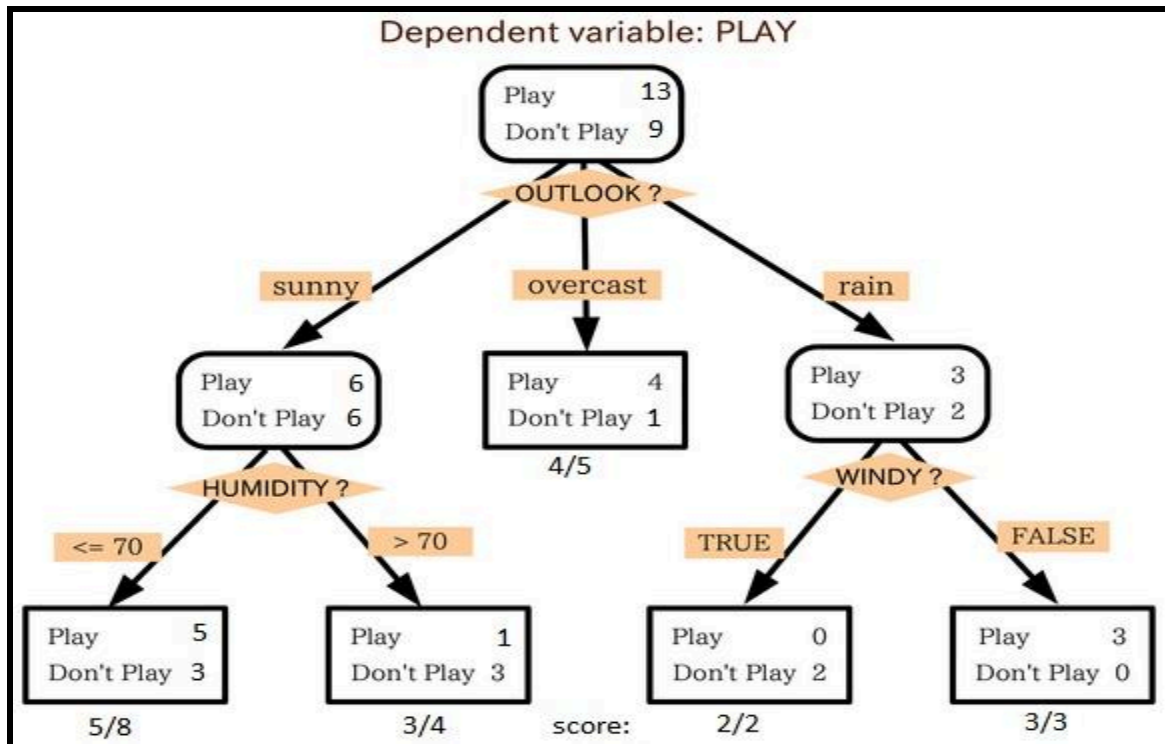
2. Logistic Regression

Logistic Regression is a special case of Linear Regression where target variable (y) is discrete / categorical such as 1 or 0, True or False, Yes or No, Default or No Default. A log of the odds is used as the dependent variable. Using a logit function, logistic regression makes predictions about the probability that a binary event will occur.



3. Decision Tree

Decision Tree algorithms are a type of probability tree-like structural model that continuously separates data in order to categorize or make predictions depending on the results of the previous set of questions. The model analyzes the data and provides responses to the questions in order to assist you in making more informed choices.



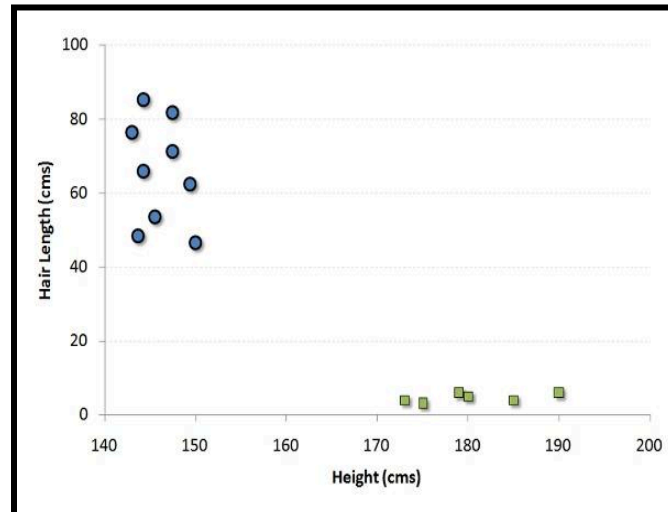
In the image above, you can see that the population is classified into four different groups based on multiple attributes to identify 'if they will play or not'. To split the population into different heterogeneous groups, it uses various techniques like Gini, Information Gain, Chi-square, and entropy.

4. Support Vector Machine (SVM) Algorithm

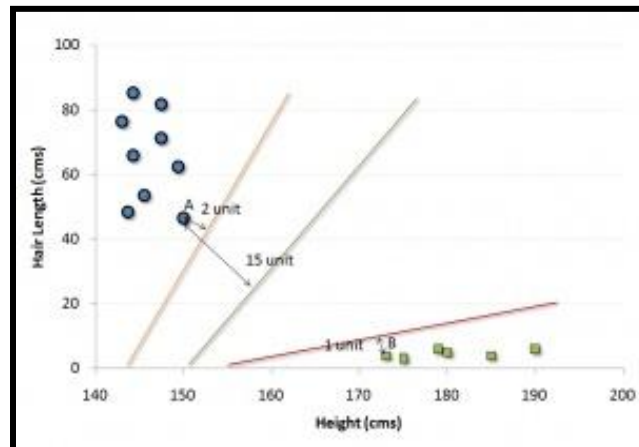
Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene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

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

For example, if we only had two features like the Height and Hair length of an individual, we'd first plot these two variables in two-dimensional space where each point has two coordinates (these coordinates are known as Support Vectors)



Now, we will find some lines that split the data between the two differently classified groups of data. This will be the line such that the distances from the closest point in each of the two groups will be the farthest away. If there are more variables, a hyperplane is used to separate the classes.



In the example shown above, the line which splits the data into two differently classified groups is the black line since the two closest points are the farthest apart from the line. This line is our classifier. Then, depending on where the testing data lands on either side of the line, that's what class we can classify the new data as.

7) Explain common feature engineering techniques such as feature scaling, transformation, and creation of new features.

Feature engineering is the process of selecting, manipulating and transforming raw data into features that can be used in supervised learning. It's also necessary to design and train new machine learning features so it can tackle new tasks. A "feature" is any measurable input that can be used in a predictive model. It could be the color of an object or the sound of someone's voice. Feature engineering is the act of converting raw observations into desired features using statistical or machine learning approaches.

Feature Engineering Processes

Feature engineering consists of various processes:

- **Feature creation:** Creating features involves creating new variables which will be most helpful for our model. This can be adding or removing some features. As we saw above, the cost per sq. ft column was a feature creation.
- **Transformations:** Feature transformation is simply a function that transforms features from one representation to another. The goal here is to plot and visualize data. If something isn't adding up with the new features, we can reduce the number of features used, speed up training or increase the accuracy of a certain model.
- **Feature extraction:** Feature extraction is the process of extracting features from a data set to identify useful information. Without distorting the original relationships or significant information, this compresses the amount of data into manageable quantities for algorithms to process.
- **Exploratory data analysis:** Exploratory Data analysis(EDA) a powerful and simple tool that can be used to improve your understanding of your data, by exploring its properties. The technique is often applied when the goal is to create new hypotheses or find patterns in the data. It's often used on large amounts of qualitative or quantitative data that haven't been analyzed before.

Scaling

Feature scaling is one of the most pervasive and difficult problems in machine learning, yet it's one of the most important things to get right. In order to train a

predictive model, we need data with a known set of features that needs to be scaled up or down as appropriate. After a scaling operation, the continuous features become similar in terms of range. Although this step isn't required for many algorithms, it's still a good idea to do so. Distance-based algorithms like KNN and k-means, on the other hand, require scaled continuous features as model input. There are two common ways for scaling:

1. Normalization

All values are scaled in a specified range between 0 and 1 via normalization (or min-max normalization). This modification has no influence on the feature's distribution, however, it does exacerbate the effects of outliers due to lower standard deviations. As a result, it's advised that outliers be dealt with prior to normalization.

2. Standardization

Standardization, or z-score normalization, is the process of scaling values while accounting for standard deviation. If the standard deviation of features differs, the range of those features will differ. The effect of outliers in the characteristics is reduced as a result. To arrive at a distribution with a 0 mean and 1 variance, all the data points are subtracted by their mean and the result divided by the distribution's variance.

8) Discuss strategies for handling high-dimensional data and selecting informative features for machine learning models.

Handling high-dimensional data and selecting informative features are crucial steps in machine learning model development. High-dimensional data refers to datasets with a large number of features, which can pose challenges such as increased computational complexity, overfitting, and difficulty in interpretation.

1. Dimensionality Reduction Techniques:

- **Principal Component Analysis (PCA):** PCA identifies the principal components that capture the maximum variance in the data and reduces the dimensionality while retaining most of the information.

- **Linear Discriminant Analysis (LDA):** LDA finds linear combinations of features that best separate different classes in the data, making it useful for classification tasks.

2. Feature Selection Methods:

- **Filter Methods:** Filter methods evaluate the relevance of features based on statistical measures like correlation, mutual information, or ANOVA F-value, and select the most informative ones.
- **Wrapper Methods:** Wrapper methods assess feature subsets by training the model iteratively with different combinations of features and selecting the subset that yields the best performance.
- **Embedded Methods:** Embedded methods incorporate feature selection directly into the model training process, such as regularization techniques like Lasso (L1 regularization) and Ridge (L2 regularization) regression.
- **Feature Importance:** For tree-based models like Random Forests or Gradient Boosting Machines, feature importance scores can be calculated to identify the most influential features in predicting the target variable.

3. Model-based Feature Importance:

- Train a machine learning model and examine the importance of each feature based on how much it contributes to the model's performance.
- Random Forests, Gradient Boosting Machines, and Linear Models with regularization (e.g., Lasso, Ridge) are commonly used models for this purpose.

4. Cross-Validation:

- Use cross-validation techniques to assess the generalization performance of the model with different feature subsets.
- Cross-validation helps in selecting features that generalize well to unseen data and reduce overfitting.

5. Regularization:

- Regularization techniques like Lasso (L1 regularization) penalize the coefficients of less important features, effectively driving them towards zero and eliminating them from the model.

6. Ensemble Methods:

- Ensemble methods like Random Forests and Gradient Boosting Machines inherently handle feature selection by selecting subsets of features at each tree/node split, focusing on the most informative features.

9) Differentiate between supervised and unsupervised machine learning algorithms.

Supervised learning

When an algorithm is trained on a labelled dataset—that is, when the input data used for training is paired with corresponding output labels—it is referred to as supervised learning. Supervised learning aims to find a mapping or relationship between the input variables and the desired output, which enables the algorithm to produce precise predictions or classifications when faced with fresh, unobserved data.

An input-output pair training set is given to the algorithm during a supervised learning process. For every example in the training set, the algorithm iteratively modifies its parameters to minimize the discrepancy between its predicted output and the actual output (the ground truth). This procedure keeps going until the algorithm performs at an acceptable level.

Supervised learning can be divided into two main types:

1. **Regression:** In regression problems, the goal is to predict a continuous output or value. For example, predicting the price of a house based on its features, such as the number of bedrooms, square footage, and location.
2. **Classification:** In classification problems, the goal is to assign input data to one of several predefined categories or classes. Examples include spam

email detection, image classification (e.g., identifying whether an image contains a cat or a dog), and sentiment analysis.

Unsupervised Learning

Unsupervised learning is a type of machine learning where the algorithm is given input data without explicit instructions on what to do with it. In unsupervised learning, the algorithm tries to find patterns, structures, or relationships in the data without the guidance of labeled output.

The main goal of unsupervised learning is often to explore the inherent structure within a set of data points. This can involve identifying clusters of similar data points, detecting outliers, reducing the dimensionality of the data, or discovering patterns and associations.

There are several common types of unsupervised learning techniques:

1. **Clustering:** Clustering algorithms aim to group similar data points into clusters based on some similarity metric. K-means clustering and hierarchical clustering are examples of unsupervised clustering techniques.
2. **Dimensionality Reduction:** These techniques aim to reduce the number of features (or dimensions) in the data while preserving its essential information. Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) are examples of dimensionality reduction methods.
3. **Association:** Association rule learning is used to discover interesting relationships or associations between variables in large datasets. The Apriori algorithm is a well-known example used for association rule learning.

	Supervised Learning	Unsupervised Learning
Definition	It's a learning setup commonly used in machine learning where human supervision is involved as they label the data and provide target output to the algorithm to map the input to it	It's a learning setup commonly used in machine learning where human supervision is minimal, as the model finds the patterns in the data without any supervision
Input Data	Data has labels	Data doesn't have labels
Data Usage	The data has x features and Y variable, and the model finds $Y = f(x)$	Patterns are found in the x features of the data as no Y variable is present
When to use	Know the expected outcome and what is being looked for	Don't know the expected outcome and what is being looked for
Nature of Problems / Types	Regression and Classification	Clustering, Dimensionality Reduction, and Association
Goal	Predict outcomes for new data based on training data	Get hidden patterns and useful insights from large datasets
Output	Predicted labels	Clusters or association rules

10) Download the dataset from the link

-<https://archive.ics.uci.edu/dataset/20/census+income>

a. Perform exploratory data analysis (EDA) techniques on this data and also discuss insights of EDA such as histograms, scatter plots, and box plots.

The insights are written in the notebook itself.

b. Perform data preprocessing pipelines on the dataset provided in question - 10 using libraries like pandas and scikit-learn.

Link of the Colab Notebook :-

<https://colab.research.google.com/drive/1KUJ04vp2pMsu8mqK-a8r2PE3wTro4DRo#scrollTo=t2b1Auhe7kfh>