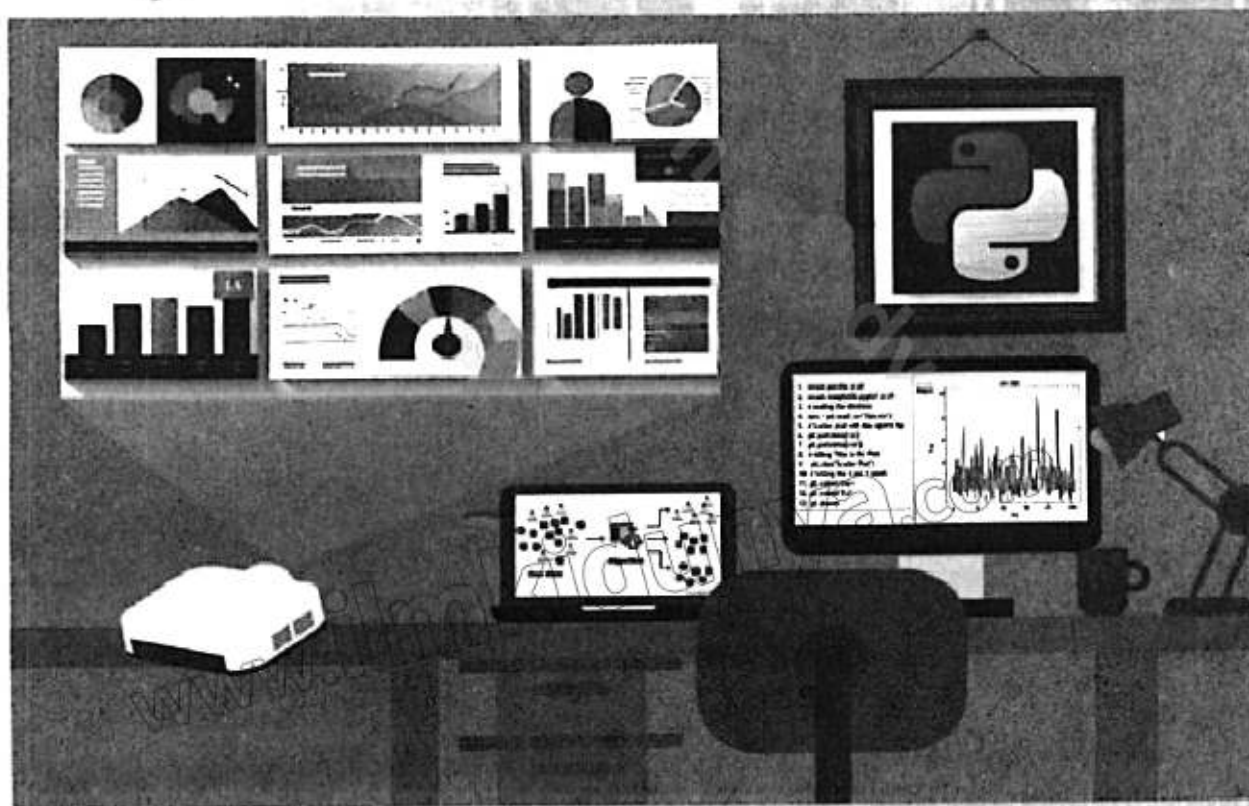




Learning Outcomes

At the end of this unit students will be able to:

- understand and evaluate applications of various programming paradigms.
- use more advanced programming constructs such as data structures (lists etc.), file handling (disk IO to write to storage), and databases in Python.
- implement complex algorithms that use lists etc. in Python
- determine more advanced techniques (unit tests, breakpoints, watches) for testing and debugging their code in Python



Give answer of the following:

- a) Algorithm
- b) Efficiency
 - Time Complexity
 - Space Complexity
- c) Clarity
- d) Correctness

Problem 4: Checking for Prime Numbers

Problem Statement: Determine if a given integer n is a prime number (i.e., it has no divisors other than 1 and itself).

Solution: Simple Divisibility Test

Give answer of the following:

- a) Algorithm
- b) Efficiency
 - Time Complexity
 - Space Complexity
- c) Clarity
- d) Correctness

Problem 5: Finding the Average of a List

Problem Statement: Calculate the average value of a list of numbers.

Solution: Summation and Division

Give answer of the following:

- a) Algorithm
- b) Efficiency
 - Time Complexity
 - Space Complexity
- c) Clarity
- d) Correctness

- 6. How can abstraction help a student create a simple timetable app for school?
- 7. A shopkeeper wants to use a program to quickly find a product in a long list. Which type of algorithm should they use and why?



Give long answers to the following Extended Response Questions (ERQs).

- 1. Compare and evaluate the efficiency of Bubble Sort and Merge Sort in sorting a list of 1,000 student names. Discuss in terms of time complexity and clarity. Which is more suitable and why?
- 2. Describe the difference between Stack and Queue. How does their operation affect algorithm design? Provide one real-life example for each data structure.
- 3. Explain the importance of tree traversal techniques. Compare in-order, pre-order, and post-order traversal with examples and their use cases.

14. What is the time complexity of Linear Search?

a. $O(1)$

b. $O(\log n)$

c. $O(n)$

d. $O(n^2)$

15. During an inter-school tech competition, students were asked to develop a solution for managing traffic signals based on traffic density. Which approach best shows their use of computational thinking?

a. They memorized how traffic lights work in real life.

b. They created a flowchart outlining how traffic signal timing changes with vehicle count.

c. They guessed a solution based on their intuition.

d. They programmed a random timer for each signal.



Give short answers to the following Short Response Questions (SRQs).

Problem 1: Finding the Sum of Digits

Problem Statement: Given an integer, calculate the sum of its digits.

Solution: Digit Extraction and Summation

Give answer of the following:

a) Algorithm

b) Efficiency

➤ Time Complexity

➤ Space Complexity

c) Clarity

d) Correctness

Problem 2: Finding the Factorial of a Number

Problem Statement: Compute the factorial of a non-negative integer n (i.e., $n!$).

Solution: Iterative Approach

Give answer of the following:

a) Algorithm

b) Efficiency

➤ Time Complexity

➤ Space Complexity

c) Clarity

d) Correctness

Problem 3: Finding the Largest Number in a List

Problem Statement: Given a list of integers, find the largest number in the list.

Solution: Linear Scan

Exercise



Select the best answer for the following Multiple-Choice Questions (MCQs).

- Which of the following best defines a data structure?
 - A method of communication
 - Away to store and organize data in memory
 - A system for internet access
 - A graphical interface design
- Which data structure allows dynamic memory allocation and easy insertion/deletion?
 - Array
 - Stack
 - Linked List
 - Queue
- In a singly linked list, each node contains:
 - Data and address of previous node
 - Only data
 - Data and address of next node
 - Index value and data
- The main advantage of a doubly linked list over a singly linked list is:
 - Random access
 - Uses less memory
 - Allows traversal in both directions
 - Faster arithmetic operations
- What type of linked list connects the last node back to the first?
 - Doubly linked list
 - Circular linked list
 - Singly linked list
 - Linear linked list
- Which operation removes the top element from a stack?
 - Enqueue
 - Dequeue
 - Pop
 - Top
- Which node in a tree does not have any children?
 - Root node
 - Leaf node
 - Sibling node
 - Parent node
- Which data structure operates on a Last-In, First-Out (LIFO) principle?
 - Array
 - Stack
 - Queue
 - Tree
- Which algorithm is used for sorting data?
 - Quick Sort
 - Binary Search
 - Depth-First Search
 - Euclidean Algorithm
- Which algorithm is an example of a Greedy Algorithm?
 - Bubble Sort
 - Huffman Coding
 - Depth-First Search
 - Binary Search
- What is the space complexity of the Binary Search algorithm?
 - $O(1)$
 - $O(n)$
 - $O(\log n)$
 - $O(n^2)$
- Which data structure allows random access to elements by index?
 - Stack
 - Queue
 - Array
 - Tree
- A school wants to automate the process of generating student report cards based on input marks. Which step of computational thinking would most likely involve identifying how to separate marks by subject, calculate averages, and assign grades?
 - Decomposition
 - Pattern Recognition
 - Abstraction
 - Algorithm Design

Summary

- **Decomposition:** Breaking down the complex problem into smaller parts, making it easier to manage and solve. For example, building a website involves separate tasks like designing, coding, backend setup, and testing.
- **Pattern Recognition:** Identifying commonalities or trends within a problem or across similar problems. For instance, in data analysis, recognizing user behavior patterns can help enhance product design.
- **Abstraction:** Focusing on the essential details of a problem while ignoring unnecessary ones, simplifying complex systems. For example, creating a user interface mockup focuses on key interactions without detailing all technical aspects.
- **Algorithm Design:** Creating a clear, step-by-step process to solve a problem efficiently, which is essential in programming. For example, sorting numbers involves using a defined procedure to arrange them from smallest to largest.
- **Correct:** Produces the desired result for all valid inputs.
- **Clear:** Easy to understand and follow.
- **Efficient:** Optimizes resources like memory and time.
- **Sorting Algorithms:** Organize data in a particular order (e.g., Quick Sort, Bubble Sort).
- **Searching Algorithms:** Find specific items in data (e.g., Linear Search, Binary Search).
- **Graph Algorithms:** Work with nodes and edges (e.g., Dijkstra's Algorithm, Depth-First Search).
- **Dynamic Programming:** Solves problems by breaking them into smaller subproblems and storing solutions (e.g., Fibonacci numbers).
- **Greedy Algorithms:** Make the best choice at each step for overall optimization (e.g., Huffman Coding).
- **Backtracking Algorithms:** Explore solutions and backtrack when necessary (e.g., N-Queens Problem, Sudoku Solver).
- **Binary Search:** Efficient for finding a target value in a sorted array, with a time complexity of $O(\log n)$.
- **Linear Search:** Examines each element sequentially in an unsorted array, with a time complexity of $O(n)$.
- **Lists and Arrays:** Store elements in sequence, allowing random access via indices, and are crucial for operations involving iteration and loops.
- **Stacks:** Follow the Last-In-First-Out (LIFO) principle, commonly used in algorithms like depth-first search.

Algorithm:

1. Initialize a counter to zero.
2. Iterate through each character in the string.
3. For each character, check if it is a vowel (i.e., one of 'a', 'e', 'i', 'o', 'u').
4. If it is a vowel, increment the counter.

After the iteration, the counter will contain the number of vowels.

Efficiency:

Time Complexity: $O(n)$, where n is the length of the string.

Space Complexity: $O(1)$ as only a counter variable is used.

Clarity:

The algorithm is simple and clear, involving basic iteration and conditional checks.

Correctness:

This method correctly counts the number of vowels by evaluating each character.

3. The search is completed when the target value matches with an element.
4. If the array ends and no match is found, conclude that the target is not in the array.

Time Complexity

The Linear Search Algorithm has the Time Complexity of $O(n)$, because the required number of operations increase linearly with the size of the input. When the algorithm starts its initial state is an array of n elements. We check each element in sequence e.g. First element, second element and so on up to the n th element. For worst case scenario, every single element in the array is examined before finding the target value (or concluding that it's not in the array).

Space Complexity

The Linear Search Algorithm has the Space Complexity of $O(1)$ - Constant Space. The Linear search utilizes fixed amount of extra space regardless of the size of the input. It only needs a few variables to store the current index and the target value, which does not scale with the size of the input. Thus, its space complexity is constant.

2.2.4 Case Studies

Problem 1: Find Greatest Common Divisor (GCD)

Problem Statement: Given two integers, find their greatest common divisor (GCD).

Solution: Euclidean Algorithm

Algorithm:

1. Take two integers a and b where $a \geq b$.
2. While b is not zero:
 - Set a to b
 - b to $a \% b$ (the remainder of a divided by b).
3. When b becomes zero, a contains the GCD.

Efficiency:

- Time Complexity: $O(\log \min(a, b))$
- Space Complexity: $O(1)$ as only a few variables are used.

Clarity:

- The Euclidean Algorithm is elegant and clear, involving a simple iterative process to find the GCD.

Correctness:

- The algorithm correctly finds the GCD based on the mathematical properties of division and remainders.

Problem 2: Count Numbers of Vowels in a String

Problem Statement: For an input string, count the number of vowels in that string.

Solution: Simple Iteration

Algorithm Steps:

1. Start by examining the number in the center of the list.
2. Check to see whether this middle number equals the one we are looking for.
3. If it matches, we have located the correct number.
4. If the number we are searching for is on the left part of the list. Repeat the process while ignoring the right side and focusing on the left.
5. If the number we are looking to get is in the list's right half. Ignore the left side and focus on the right portion, continuing the procedure again.
6. Carry on to filter across the list till we identify the number or figure out that it is not on the list.

Time Complexity

The Binary Search Algorithm has Time Complexity of $O(\log n)$. The binary search divides the search space in half with each step, this halving process means that the number of elements to search through is reduced exponentially.

When the algorithm starts its initial state is an array of n elements. At each step, we apply the divide and conquer process. This is done by comparing the target value to the middle element of the array or current search range. This comparison helps in eliminating half of the current search range (this half could be either half of the current search range).

- After the first step, we have left with $n/2$ elements.
- After the second step, we have left with $n/4$ elements.
- After k steps, we have left with $n/2^k$ elements.

The process continues until the number of remaining elements is 1 (or the target is found). The number of steps required is relative to the logarithm of n because we keep halving the problem size each time. Specifically, the number of steps is $\log_2(n)$.

Space Complexity

The Binary Search Algorithm has the space complexity of $O(\log n)$ - Logarithmic Space. To track the current state of recursion, the binary search depends upon the stack. Each recursive step adds a new frame to the stack. For worst case scenario, the number of the recursions is relative to the number of times the problem size can be halved, which is $\log_2(n)$. Thus, the space complexity is logarithmic due to the recursive stack usage.

Example 2: Linear Search Algorithm

For Linear search, each element in the array is examined one by one till the target value is found or we reach at the end of the array.

Algorithm Steps:

1. Start at the beginning of the array and examine each element one by one.
2. Compare target value to each element of the array .

often expressed using Big O notations (e.g., $O(n)$, $O(\log n)$, $O(n^2)$).

Time Complexity:

Time complexity provides a way to evaluate efficiency of a solution. This is done by estimating the time required to solve a problem. While solving a problem, the main outcome of the solution is algorithm. Therefore, this measures the execution time of an algorithm with respect to the size of input. The input size refers to problem size and it represents the size of task and amount of data.

Space Complexity:

Like Time complexity, it evaluates the efficiency of solution but focus on memory used by the solution. This measures the amount of memory an algorithm uses with respect to the size of input.

When evaluating the efficiency of an algorithm, we often focus on the worst-case scenario. For instance, we are searching for a specific element in an array by comparing it with each element at every index, the worst-case situation occurs when the desired element is at the very last index. Big O notation helps us understand the upper bound of an algorithm's running time or space usage relative to the size of the input, providing a way to gauge its efficiency in the most challenging conditions.

Common Big O Notations:

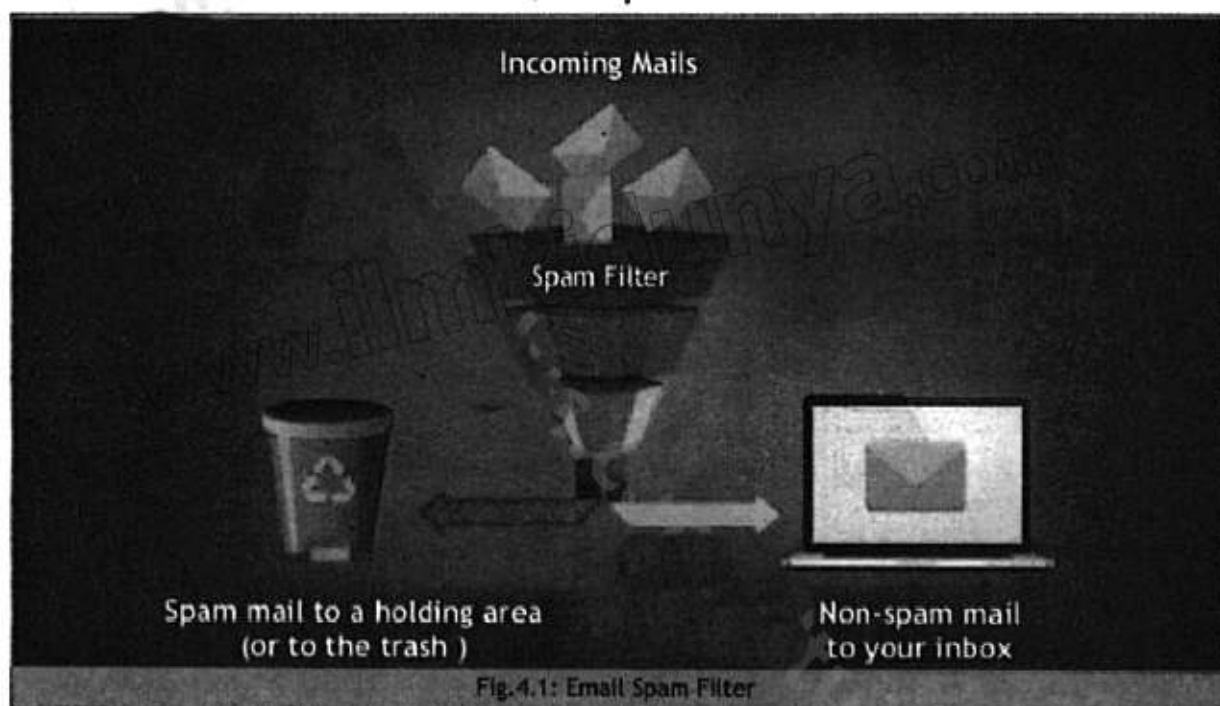
- **$O(1)$ - Constant Time:** The algorithm's running time is constant, regardless of input size. For example, finding a given member in a list involves the same length of time regardless the list has 10 or 10,000 entries.
- **$O(\log n)$ - Logarithmic Time:** Algorithm execution time gradually increases with increasing input size. This is comparable to looking for a word in a dictionary, in which each step reduces the search space by half.
- **$O(n)$ - Linear Time:** The algorithm's execution time scales up with input size i.e. if an input size doubles, so does the processing time. One example is a basic loop that goes through each entry in a list.
- **$O(n \log n)$ - Linearithmic Time:** This time-based complexity combines linear and logarithmic growth. It is found in effective sorting algorithms such as Merge Sort and Quick Sort, in which the algorithm separates the problem into smaller sections and processes each one separately.
- **$O(n^2)$ - Quadratic Time:** Execution time increases dramatically with input size. This is characteristic of algorithms that use nested loops and compare every member in a list to every other part. When the input size is doubled, the time required to finish the algorithm quadruples.

Example 1: Binary Search Algorithm

Binary search is a search algorithm that locates a target value in a sorted array. It operates by repeatedly partitioning the search interval.

Introduction

Machine Learning is a method in which computers learn from existing data to make new decisions. A computer learns from the predefined rules, which are provided at the time of training it. These algorithms are designed carefully with the help of raw data and then transforming it into useful models. These models are assessed based on their performance in various environments. During these assessments, the models learn patterns and improve their performance. After attaining a certain level of accuracy, these models are able to predict future values based on the past data patterns. After analyzing the results obtained by such machine learning models, we can understand cause and effect relationships in data. In this way, we can use data and computers to visualize the future results based on existing data



We can understand it with the help of an example of Email Spam Filter. A spam filter learns from past emails' labels whether they are spam or not. When we mark some email in our inbox as spam the machine learning model learns from the keywords used in that email. Next time if we receive an email that contains similar words, which were used in "previously marked spam" email, it will automatically send that email into spam folder.

4.1 Data Types

Data is a collection of raw facts and figures collected from different sources. In data science and machine learning data is categorized into various types. The following table provides a brief description and example of some data types:

Data Type		Example
Numerical	It is quantitative data and used to represent numerical values.	Marks, Salary, Age
Categorical	It is data which has discrete value. It is further divided into Nominal data and Ordinal Data.	Nominal: Male, Female, Color Categorical: Rank, High, Low, Small, Large.
Boolean	It has only two possible values 0 or 1.	True/False, Yes/No
Text Data	It contains a sequence of characters.	Description, Review, Feedback
Time Series	It is collected over a long period of time.	Stock price, weather data
Image	It is represented as a pixel value in the image.	X-Ray, Ultrasound, Photographs
Audio	It is in the form of sound waves.	Speech, Music
Structured	It is organized in the form of rows and columns (tables).	Spreadsheet, Database
Semi Structured	It is organized in some specific formats, but there is no fixed pattern/schema.	XML, JSON files
Unstructured	It is without any predefined structure.	Videos, Emails, Web Pages.

Introduction to Machine Learning

Before going into the core concepts of machine learning, it is important to understand the foundational topics that make these systems work. These include understanding how models are trained, evaluated, and improved using data, as well as learning the key metrics used to measure the performance of these models. The following are some basic but important concepts about Machine Learning. These are essential building blocks to understand Machine Learning models and related topics.

4.1.1 Difference Between Machine Learning and Rule-based Algorithms

Machine learning models learn from data to make decisions. They adapt and improve over time as they process more data. Whereas rule-based algorithms follow fixed instructions. They do not change unless we deliberately update them. Machine learning can handle complex problems and unexpected situations. Rule-based systems are simpler and only work well if the rules are correct. Machine learning models are flexible and can learn new things. Rule-based systems are consistent but need new rules for different problems. Machine learning is more dynamic and can handle a variety of data.

Both Machine Learning and Rule Based algorithms have their advantages and limitations, which is as follows:

Advantages of Machine Learning:

It can handle complex and unstructured data. It can automatically learn patterns from data without any explicit programming. It does not require expert knowledge of human intervention. Its accuracy depends on the size of the dataset.

Advantages of Rule-based Algorithms:

It does not require large datasets. It can work effectively with limited data because it depends upon predefined rules. Its performance is consistent with the same set of rules. It requires fewer resources and can be implemented easily.

Limitations of Machine Learning:

It does not perform well if data is too little. Their decision-making process is complex and cannot be easily interpreted. It requires significant computational resources. If the model is too complex it can lead to overfitting. If training data is biased it can provide wrong results.

Limitations of Rule-based Algorithms:

It has limited flexibility, as it is not adaptable to new data which is not anticipated in predefined rules. It can provide wrong results if data is unstructured or ambiguous. It cannot learn from data. If too many rules accumulate it can slow down the system.

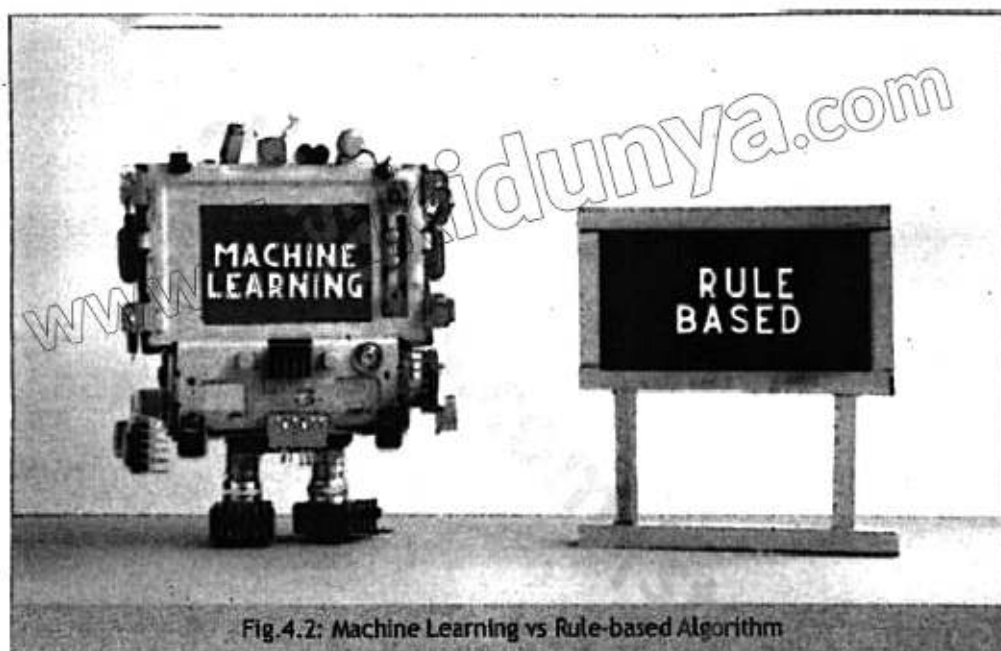


Fig. 4.2: Machine Learning vs Rule-based Algorithm

4.1.2 Model Building in Machine Learning

Model building in machine learning involves several key steps to develop a system that can learn from data and make predictions or decisions. At first, data is collected and preprocessed to ensure it is clean and suitable for analysis. This might involve normalizing values, handling missing data, and encoding categorical variables. After data preprocessing, a suitable machine learning algorithm is chosen based on the problem at hand, whether it's classification, regression, clustering, or another type. The algorithm is then trained on the dataset, adjusting its parameters to minimize error and improve accuracy.

During training, the model's performance is evaluated using a separate validation set of data to ensure its accuracy about new unseen data. Once the model is optimized, it is tested on a final test dataset to assess its effectiveness. If the results are satisfactory, the model can be deployed for real-world use; otherwise, it may require further improvements in code. Throughout this

process, every time the model reads data and calculates results, the algorithm keeps on learning. Repeated refinement and evaluation help in achieving a model that performs well and meets the specific needs of the application.

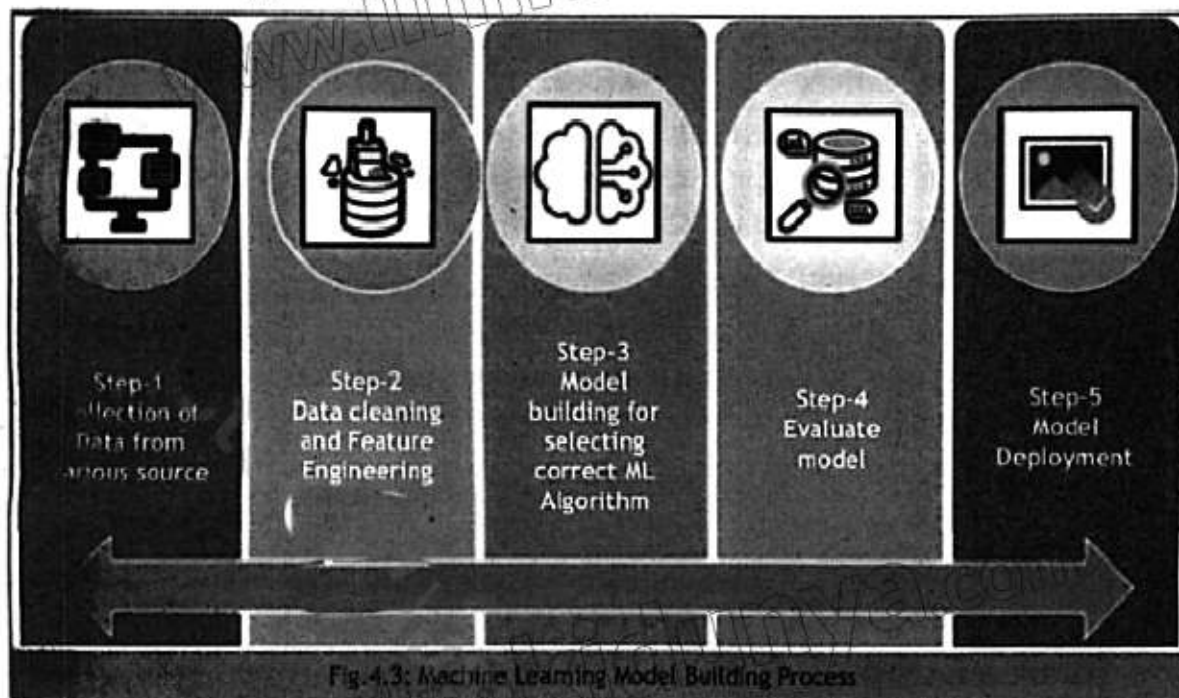


Fig.4.3: Machine Learning Model Building Process

Key Terms in Model Building using Machine Learning:

To understand the entire process of model building using Machine Learning, some key terms are necessary to understand. We will take a real-life problem, so that we can better understand each term. We want to develop a model which can identify or differentiate the images of dogs, cats and horses. We must have a reasonable collection of images of all the three animals. Now we will learn each key term with reference to this example as following:

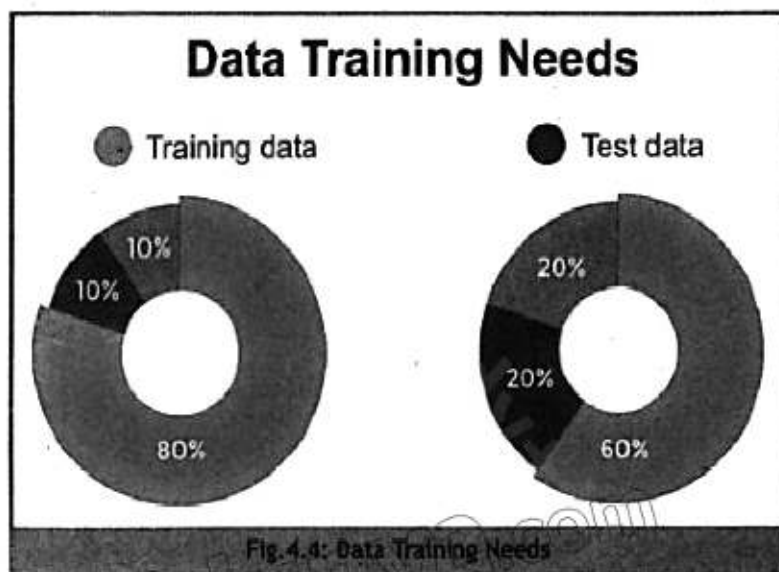


Fig.4.4: Data Training Needs

Feature Engineering: Feature engineering is the process of selecting and transforming the most relevant data features to improve a model's performance. For example, if you want to predict that a given image is of a dog, cat or horse, we have to make a list of common and distinct features of each animal i.e. size, color, appearance, shape of eyes, ears, hair, nose etc. Our model will keep data about all possible variation of features of these animals, so that a sitting horse cannot be confused with the image of a big size standing dog.

Test-Train Split: A test-train split divides data into two parts i.e training data (main portion) and test data (smaller portion). In our example we decided to make a test train split of seventy and thirty percent. This ratio should be chosen wisely. If we provide too much data at the time of training our model will overfit. If we provide too little data for training, then the model will underfit.

Model Training: Model training is the key feature in Machine Learning. It helps the model to make decisions which are more accurate and meaningful. Without proper model training, the machine is unable to decide about the new data. Supervised Learning, Unsupervised Learning and Reinforcement Learning are commonly used Machine Learning models.

In supervised learning, the model is trained using labeled data, where the correct output is provided for each input. A common example of a supervised learning algorithm is Linear Regression, which is used to predict continuous values such as prices or temperatures.

In unsupervised learning, the data is unlabeled, and the model tries to find patterns or groupings on its own. An example of this type is K-Means Clustering, which groups similar data points together, such as in customer segmentation.

Reinforcement learning involves a model that learns by interacting with an environment, receiving feedback in the form of rewards or penalties. A popular algorithm in this category is Q-Learning, often used in game playing or robotic navigation where the model learns to make decisions to maximize its rewards.

Overfitting: It happens when a machine learning model learns the training data too well, including its noise and irrelevant details. This makes the model perform very well on the training data but poorly on new, unseen data because it's too specific to the training set.

Underfitting: It occurs when a model is too simple and fails to capture the underlying patterns in the data. As a result, it performs poorly both on the training data and on new data, as it hasn't learned enough from the available information.

Model Assessment: A Model assessment evaluates a model's performance using parameters like accuracy, precision, and recall. Keeping in view our example, we provide 100 new images of cats, dogs and horses which were not part of our train and test data. If our model predicts the correct results for 80 images and for rest of 20 images it predicts wrong labels, like it gives horse label for an image of dog then the accuracy of our Machine learning model is 80%.

Accuracy: This is the percentage of total predictions that are correct. It tells how often the model is right overall.

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

Precision: This measures how many of the predictions the model made for a certain class (e.g., "positive" cases) are actually correct. It's about the quality of the positive predictions.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall: This tells how well the model identifies all the actual positive cases in the data. It shows the model's ability to find all the relevant instances.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

True Positive (TP): Model predicts "Cat," and the actual animal is a Cat.

False Positive (FP): Model predicts "Cat," but the actual animal is a Dog or Horse.

True Negative (TN): Model predicts "Not Cat," and the actual animal is a Dog or Horse.

False Negative (FN): Model predicts "Not Cat," but the actual animal is a Cat.

F1 Score: It is the harmonic mean of Precision and Recall. It balances the trade-off between precision and recall. It is very useful when the dataset is imbalanced, giving a single score that reflects both precision and recall performance.

$$F1_{\text{Score}} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Training Data: Training data is the main dataset used to teach a model to make predictions or decisions. In our example we will provide a reasonable number of images of dogs, cats and horses from our huge dataset. If we have a total of one hundred thousand images, then we can provide sixty to seventy thousand images to our Machine Learning Model to learn. Each picture will contain relevant labels like cat, dog or horse if we are using supervised learning technique. These concepts have been already discussed in Grade 11.

Test Data: Test data is a small portion of data that we decide to use for evaluation purposes of our Machine Learning model. For example, after providing training data when our model is trained, we can provide unknown images from our dataset and check whether the model provides the right answer or not. These images will not be given to our model at the stage of training, rather they will be from that forty or thirty percent of images which were not provided at the time of training.

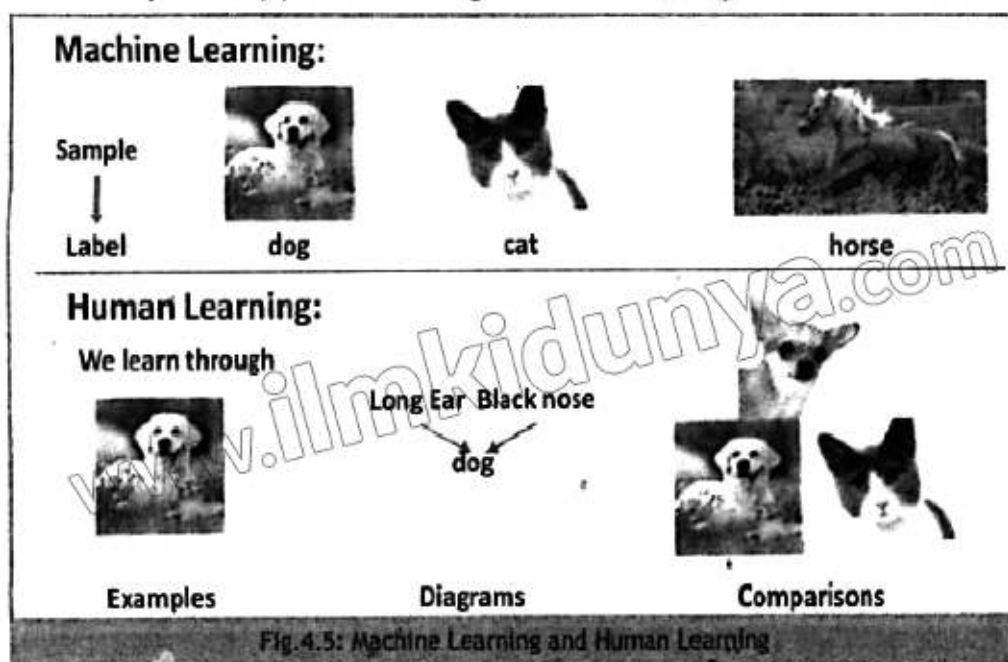


Fig. 4.5: Machine Learning and Human Learning

4.1.3 Learning from Data

In machine learning, models find patterns in data. The more data the model has, the better it can learn. These models improve their performance based on feedback from the data. Learning from data helps the model make accurate predictions. Different algorithms work better with specific types of data. Learning involves understanding trends and patterns in the data. As the model sees more examples, its performance improves.

Data quality affects how well the model learns. For example, if our dataset contains very clear images of cats, dogs and horses the learning of our model will be good. Continuous learning keeps the model accurate and improves its accuracy over time. Good data helps models learn and make better predictions.

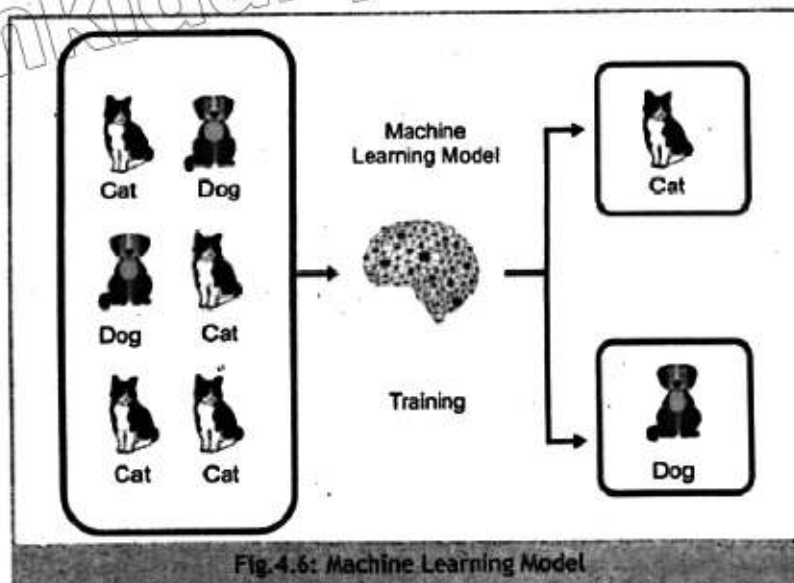


Fig.4.6: Machine Learning Model

4.1.4 Algorithms for Different Data Structures

Selecting appropriate algorithm for Machine Learning model is very important. There are certain guidelines which help us to decide which algorithm we should select. There are numerous algorithms available. It is important to understand which algorithm is more appropriate and provide desired results. The following are some key concepts to consider while choosing the right algorithm:

Algorithm selection according to data type:

1. **Structured data:** It is organized in tables with rows and columns. The structured data is easily searchable and quantifiable. The examples of structured data are databases, spreadsheets, and CSV files. Traditional algorithms like regression and classification are suitable for this type of data.
2. **Unstructured data:** It is unorganized data and has no specific structure. It includes text, images, audio, and video files. The unstructured data lacks a predefined format. The example of unstructured data are social media posts, emails, and sensor readings. Advanced algorithms like neural networks are suitable for this data.

Supervised learning uses labeled data to train models. Unsupervised learning finds patterns in unlabeled data. Regression algorithms predict continuous values like prices. Classification algorithms sort data into categories. Clustering algorithms group similar data points together. Neural networks are used for complex patterns and large data. For example, if the data is a collection of cat images, then there will be very complex patterns like colors, shapes, textures, position of the cat, angle of the picture etc. Neural networks can handle this complex pattern

easily. Similarly, if size of our dataset is very large, for example we take social media posts on the topic of COVID-19, it will have a huge number of posts. Neural networks can be trained on many posts in parallel by using powerful computers and GPUs. Therefore, these algorithms are suitable for such tasks. Choosing the right algorithm depends on the data type and problem. Matching algorithms to data structures is key for effective modeling.

4.2 Data Visualization

We are already familiar with the term data visualization in which we used programming paradigms and Python programming language to visualize Tips dataset. Data visualization is the 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. Additionally, it provides an excellent way for employees or business owners to present data to non-technical audiences without confusion.

There are many tools that can help us turn data into pictures and charts, making it easier to understand. For example, SQL is a language that helps us get data from databases. Python has libraries like Matplotlib and Seaborn that create charts and graphs. R is a special tool that's great at creating statistical graphics and visualizations.

Using these tools, we can create charts and graphs that show us patterns and trends in the data. Different tools are better suited for different tasks and skill levels. The right tool can help us present our findings in a clear and effective way. Some tools even let us interact with the data, making it more fun to explore. Choosing the right tool depends on what we want to do with the data. Good visualization tools help us tell a story with the data, making it easier for others to understand.

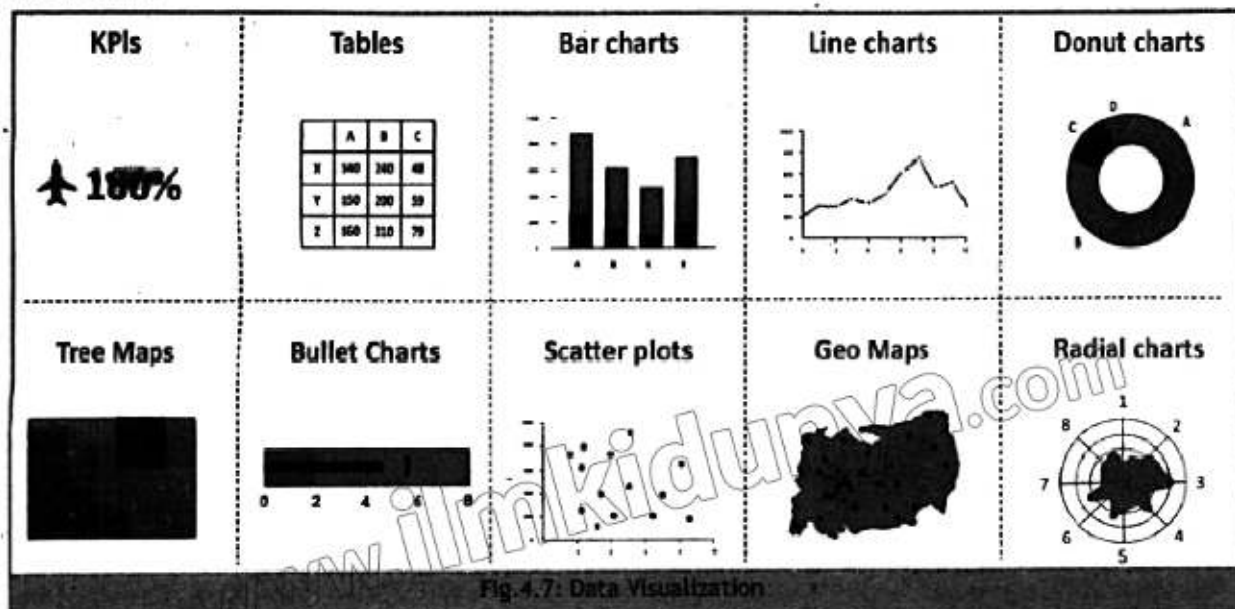


Fig.4.7: Data Visualization

4.2.1 Data visualization by using Python

The following code can be used to learn how python libraries can be used to visualize data:

```

DataVisualization.ipynb
File Edit View Insert Runtime Tools Help

Q Commands + Code + Text

# Import numpy library and give it a short name 'np'
# We use numpy to create and manage arrays (tables of numbers)
import numpy as np

# Import the 'pyplot' part of matplotlib and give it a short name 'plt'
# We use 'pyplot' to make graphs like scatter plots and lines
import matplotlib.pyplot as plt

# Import the LinearRegression class from sklearn's linear_model module
# We use it to create a machine learning model that can draw a straight line through points
from sklearn.linear_model import LinearRegression

# Step 1: Prepare the data (study hours and marks)
x = np.array([[1], [2], [3], [4], [5]]) # Input: hours studied
y = np.array([10, 20, 30, 40, 50]) # Output: marks obtained

# Step 2: Create and train the Linear Regression model
model = LinearRegression()
model.fit(X, y)

# Step 3: Predict the marks based on study hours
y_pred = model.predict(X)

# Step 4: Plot the actual points and the prediction line
plt.scatter(X, y, color='blue') # Plot real data points in blue
plt.plot(X, y_pred, color='red') # Plot the prediction line in red
plt.title('Study Hours vs Marks') # Title of the graph
plt.xlabel('Hours Studied') # X-axis label
plt.ylabel('Marks Obtained') # Y-axis label
plt.show() # Show the final graph

```

Fig.4.8: Python Code for Data Visualization

After executing the Python code by using google colab, the following will be the output:

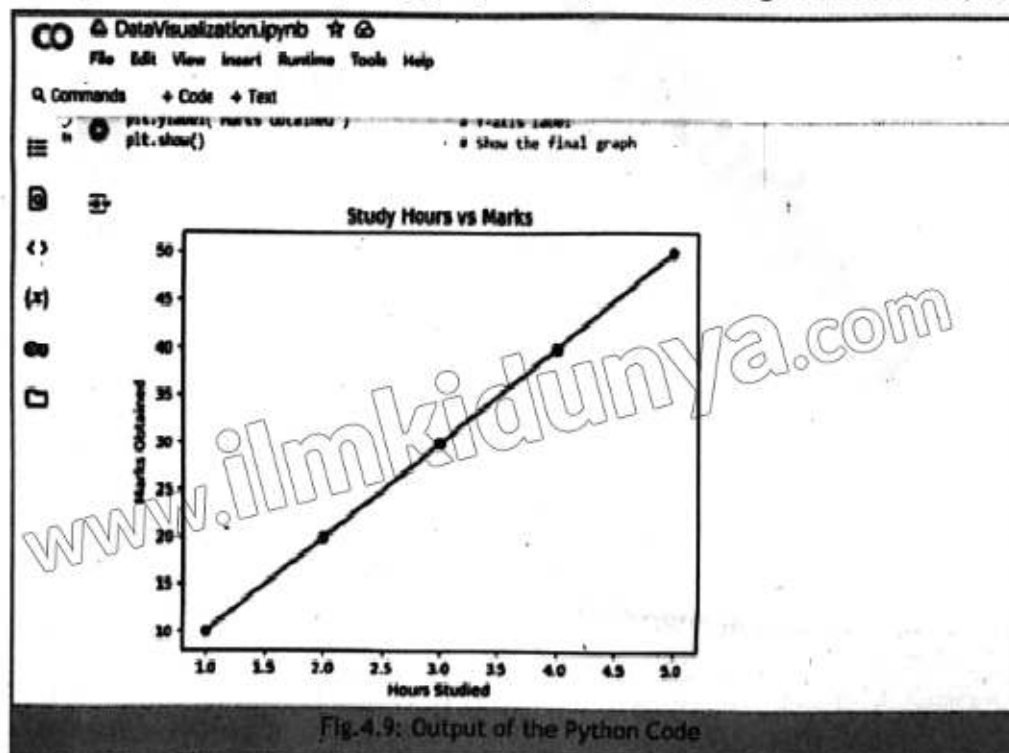


Fig.4.9: Output of the Python Code

4.2.2 Data Storytelling

Data storytelling uses visuals to communicate insights from data. It combines charts, graphs, and narratives to explain findings. A good data story engages the audience and makes data relatable. It helps in making data-driven decisions by highlighting key points. Clear storytelling simplifies complex data for better understanding. It shows the impact of the data on real-world issues. Effective data stories make the information memorable and actionable. They provide context and meaning to raw data. Data storytelling bridges the gap between data and decision-making. It turns data into a compelling narrative.

Data Storytelling Example/Case Study

Context: A school wants to improve student attendance.

Data:

- Average attendance rate: 85%
- Highest attendance rate: 9th grade (90%)
- Lowest attendance rate: 12th grade (78%)
- Average absences per student: 5 days

Story:

Meet Jameel, a 12th-grade student who represents our school's attendance challenge. Jameel misses an average of 5 days of school, which puts him at risk of falling behind. But our 9th-grade students are showing us that it is possible to do better, with an impressive 90% attendance rate. What if we could identify the factors driving their success and apply them to our 12th-grade students?

By doing so, we could help Jameel and his fellows to stay on track and do better. Let's work together to make attendance a priority and support our students' success!

Key elements:

1. Relatable character (Jameel)
2. Contextual data (attendance rates, absences)
3. Insight (9th-grade success)
4. Call to action (apply insights to 12th-grade students)

This simple story uses data to paint a picture, evoke empathy, and inspire action. It's a basic example, but it illustrates how data storytelling can make complex information more engaging and memorable.

4.2.3 Formulating Questions and Data Evaluation

Formulating questions guides data exploration and analysis. Identify relevant data sets to answer these questions. Evaluate how well the data addresses the questions posed. It compares new data with previous findings to see if they align or not. The questions are used to focus on specific data insights and trends. They are refined after evaluation to ensure the data quality. The good quality

of questions ensures accurate results. Effective question formulation drives meaningful analysis. Good questions lead to better data insights. Data evaluation helps in understanding and interpreting findings. We can better understand by the following example that how to formulate a question and evaluate data:

Topic: Analyzing the Impact of School Breakfast Programs on Student Attendance

Formulating Questions:

1. Does participating in a school breakfast program improve student attendance?
2. Which grade levels benefit most from breakfast programs in terms of attendance?
3. Do students from low-income families show greater improvement in attendance compared to their peers?

Identifying Relevant Data Sets:

1. Student attendance records
2. Breakfast program participation data
3. Demographic data (grade level, family income, etc.)

Evaluating Data Quality:

1. Check for missing or inconsistent data
2. Ensure attendance records are accurate and up-to-date.
3. Verify breakfast program participation data matches student records

This example demonstrates how formulating questions guides data exploration and analysis, leading to meaningful insights and informed decision-making.

4.2.4 Descriptive Statistics Techniques

There are some descriptive statistical techniques to summarize and describe data. The basic techniques include Mean, Median and Mode. Mean is the average value of the data set. Median is the middle value when data is ordered. Mode is the most frequent value in the data. Range shows the difference between the highest and lowest values. Standard



Fig.4.10: School Breakfast Program

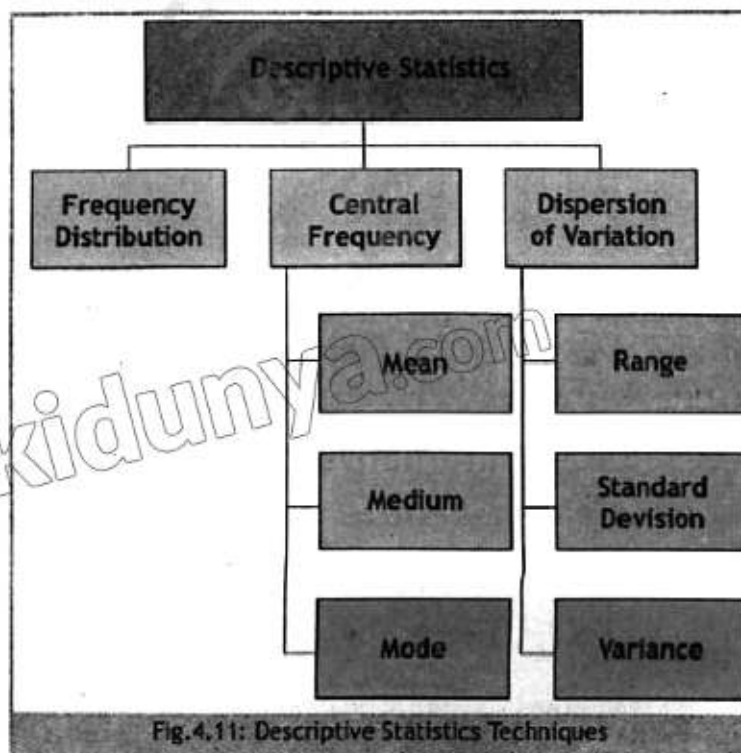


Fig.4.11: Descriptive Statistics Techniques

Deviation measures how spread out the values are. Variance is the average of squared differences from the mean. Percentiles indicate values below which a certain percentage of data falls. Histograms display the frequency distribution of data. These techniques help in understanding and summarizing data effectively. These statistical techniques are frequently used to get true insight and make informed decisions in the Machine Learning Model.

4.3 Hypothesis Formulation and Hypothesis Testing

Hypothesis formulation is the process of making an educated guess or assumption about a relationship between variables based on existing data or prior knowledge. This assumption, or hypothesis, is a starting point for investigating how certain factors might influence an outcome. It provides a framework for testing and discovering whether these relationships exist in the data or not.

For example, we formulate a hypothesis: "Increased study hours lead to better exam scores for students." This hypothesis proposes a cause-and-effect relationship between study hours (independent variable) and exam scores (dependent variable). A cause-and-effect relationship is the connection between two events or factors, where one event (the cause) directly influences or leads to another event (the effect).

Once a hypothesis is formulated, hypothesis testing is the statistical process of determining whether the data supports or refutes the hypothesis. It is a critical tool used to validate assumptions, model performance, and patterns in data. There are two key hypotheses in hypothesis testing, Null Hypothesis and Alternate Hypothesis.

In hypothesis testing, statistical tools like p-values and significance tests are used to determine whether to accept or reject the null hypothesis. It allows us to validate whether a certain is statistically significant or merely due to chance.

4.3.1 Null and Alternative Hypotheses

Null Hypothesis (H_0): This states that there is no effect or relationship between the variables. It assumes any observed differences are due to random chance.

Alternative Hypothesis (H_1): This suggests there is a significant effect or relationship between variables.

The Null Hypothesis assumes no effect or relationship exists. The Alternative Hypothesis suggests

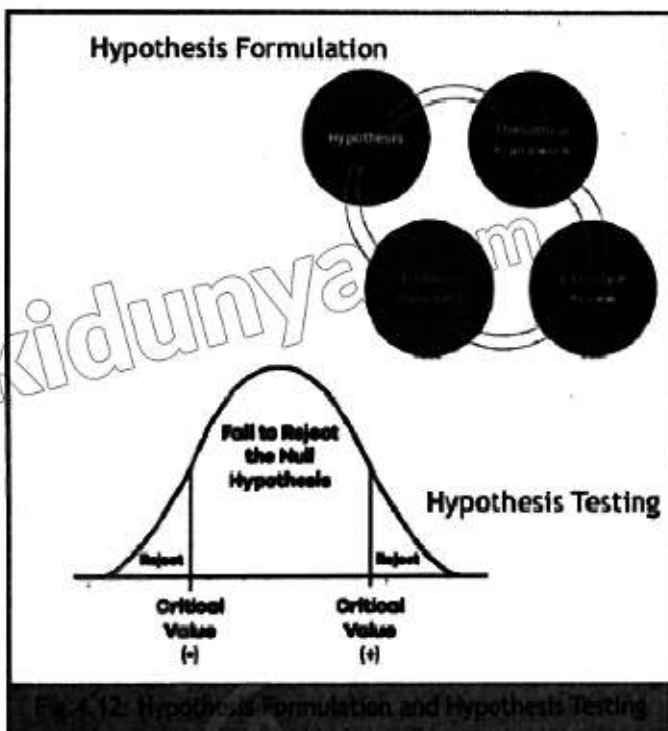


Fig. 4.12: Hypothesis Formulation and Hypothesis Testing

an effect or relationship is present. Testing aims to confirm or reject the null hypothesis. The null hypothesis is the default assumption that needs to be challenged. The alternative hypothesis is what researchers want to prove. Results help in deciding which hypothesis is supported. Both hypotheses guide the research and testing process. Clear definitions of these hypotheses are essential for accurate testing. They provide a framework for analyzing data and drawing conclusions. Effective hypothesis testing relies on understanding both types. The following example helps to understand the concept easily:

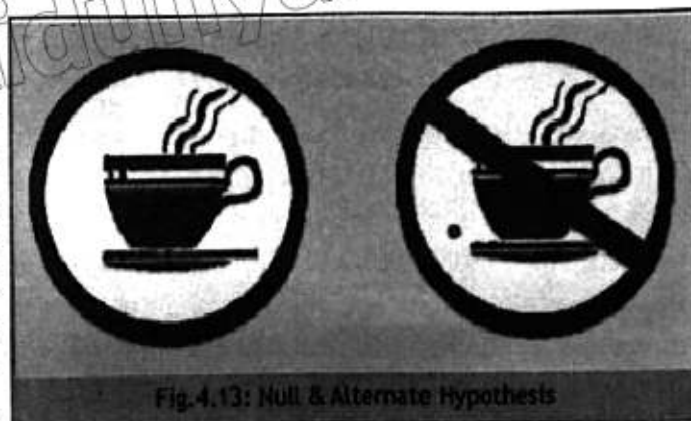


Fig.4.13: Null & Alternate Hypothesis

Example: "Does drinking coffee improve cognitive function in students?"

Null Hypothesis (H_0): Drinking coffee has no effect on cognitive function in students. (Assumes any observed differences are due to random chance.)

Alternative Hypothesis (H_1): Drinking coffee improves cognitive function in students. (Suggests a significant effect or relationship exists.)

Testing: Researchers collect data on students' cognitive function with and without coffee consumption.

Results: If the data shows a significant improvement in cognitive function with coffee consumption, the null hypothesis is rejected, and the alternative hypothesis is supported.

Conclusion: Drinking coffee has a positive effect on cognitive function in students.

4.3.2 P-values

P-values measure how likely results are due to chance. A low P-value means strong evidence against the null hypothesis. A high P-value suggests weak evidence and possibly random results. Understanding P-values helps in interpreting test results. It ensures that conclusions are based on reliable evidence. For example, if we flip a coin ten times and we get 7 times heads and 3 times tails. We might consider that coin is not fair, or it just happened by chance.

We can take another example to find the effectiveness of a new fertilizer. We want to see if it helps plants grow taller. We divide the plants into two groups: one group gets the new fertilizer, and the other doesn't. After a month, we measured the height of the plants and found that the fertilized plants grew taller than the others. Now the question arises, "Did the fertilizer really make a difference, or did the plants grow taller just by

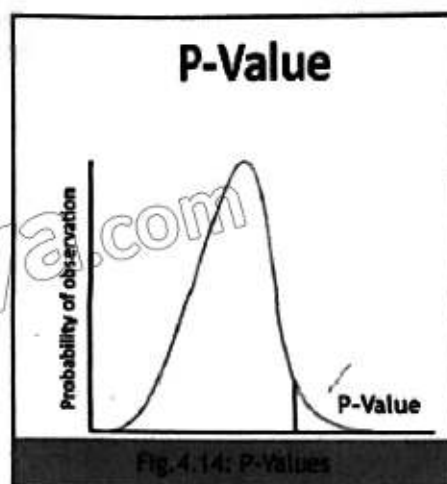


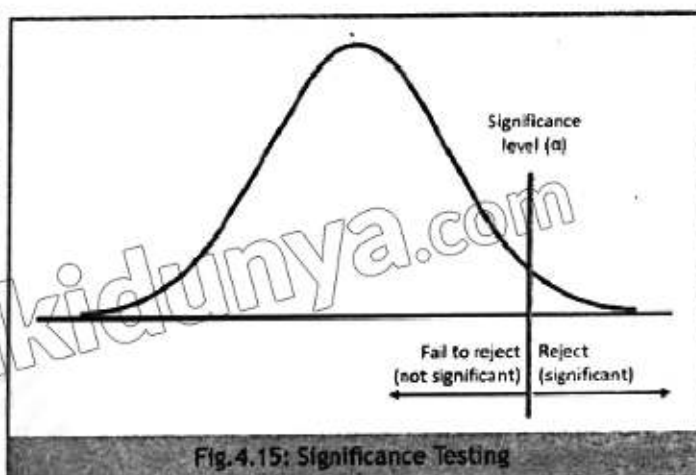
Fig.4.14: P-Values

chance?" The P-value helps answer this. It tells you the chance that the difference in plant height happened randomly, without the fertilizer having any effect.

If the P-value is small (like 0.02), it means it's very unlikely the taller growth happened by chance. This suggests that the fertilizer probably did help the plants grow taller. If the P-value is large (like 0.5), it means there's a good chance the difference in height was just random, and the fertilizer might not have had any real effect. In simple terms, the P-value helps us decide whether the fertilizer really worked or if the result happened by luck. There are certain formulas or tests used to calculate P-values, which are beyond the scope of this grade yet.

4.3.3 Significance Testing

It is a statistical method used to determine if a result is due to chance or it is statistically significant. Statistically significant means it is likely due to some real effect or factor. Significance testing helps determine if results are meaningful. It compares the test statistics with critical value. Setting a significant level helps to decide the threshold for importance. Significance testing helps in making data-driven decisions. It is crucial for validating research findings. For example, if we



formulate a research question "Can exercising for six weeks help lower blood pressure?" The null hypothesis is "Exercise doesn't affect blood pressure" while the alternative hypothesis will be "Exercise lowers blood pressure". For statistical experimentation we collect data from the people who exercised for six weeks and had a reduction in blood pressure of 8mmHg. The significant test will be to check if this result is real or just a coincidence. By applying statistical formula, the P-Value will be calculated which indicates that there is only 0.5% chance of getting this result by chance. So, we conclude that since P-Value is very small, that is 0.005 we reject the null hypothesis and accept the alternate hypothesis. It means that exercising for six weeks likely lowers blood pressure.

4.3.4 Hypothesis Testing

Hypothesis testing evaluates assumptions using statistical methods. The Test Statistic measures how much data deviates from the null hypothesis. Hypothesis testing involves certain statistical procedures and terms. The terms will be explained with the help of subsequent example. Confidence Interval shows the range where the true value likely falls. Type I Error occurs when a true null hypothesis is incorrectly rejected. Type II Error happens when a false null hypothesis is not rejected. Power measures the ability to detect an effect if one exists. Significance Level is the threshold for determining if results are significant. Sample Size affects the reliability of test results. Understanding these concepts helps in accurate hypothesis testing. They guide in

drawing valid conclusions from data. Now we will explore each term briefly with the help of a simple example:

Question: Does drinking coffee improve productivity?

Null Hypothesis (H_0): Coffee has no effect on productivity.

Alternative Hypothesis (H_1): Coffee improves productivity.

Test Statistic: Measure productivity in a sample of people who drink coffee and those who don't. Calculate the test statistic (e.g., t-statistic) to see how much the data deviates from H_0 .

Confidence Interval: Calculate a 95% confidence interval for the average productivity increase with coffee consumption. If the interval is (5, 15), we're 95% confident the true productivity increase falls between 5 and 15.

Type I Error: Rejecting H_0 when it's actually true (i.e., coffee has no effect). This occurs when we conclude coffee improves productivity when it doesn't.

Type II Error: Failing to reject H_0 when it's actually false (i.e., coffee does improve productivity). This occurs when we conclude coffee has no effect when it actually does.

Power: The ability to detect a real productivity increase if coffee actually improves it. A larger sample size increases power.

Significance Level: Set $\alpha = 0.05$ as the threshold for determining if results are significant. If p-value < 0.05 , we reject H_0 .

Sample Size: A larger sample size (e.g., 1000 people) provides more reliable results than a smaller sample size (e.g., 10 people).

By understanding these concepts, we can accurately test hypotheses and draw valid conclusions from data:

- Hypothesis testing evaluates assumptions using statistical methods.
- Test statistic measures deviation from H_0 .
- Confidence interval estimates true values.
- Type I/II Errors occur from incorrect assumptions.
- Power detects real effects.
- Significance level sets thresholds.
- Sample size impacts reliability.



These concepts help us make informed decisions based on data analysis.

4.3.5 Steps of Simple Hypothesis Testing

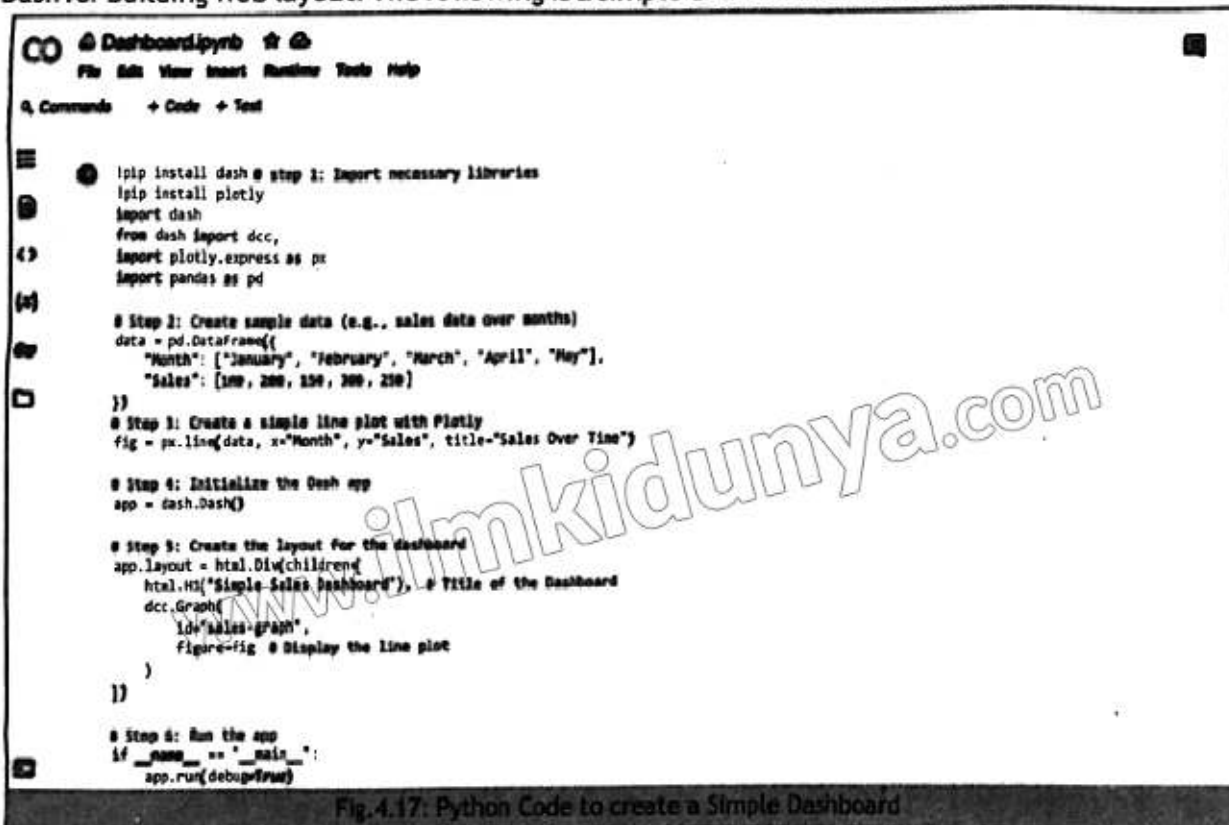
Simple hypothesis testing involves a certain sequence of steps. We can list them as follows:

- We start by formulating null and alternative hypotheses.
- Collect data and perform the analysis.
- Calculate the test statistic from the data.
- Compare the test statistic with the critical value to draw conclusions.
- Determine the P-value to assess significance.
- Decide whether to reject or not reject the null hypothesis.
- Use the results to make data-driven decisions.
- Interpret the findings in the context of the research question.
- Report the results clearly with appropriate visuals.

Therefore, hypothesis testing helps in understanding basic data relationships.

4.3.6 Creating Dashboards with Python

A dashboard is a visual tool that helps to understand data by displaying it in the form of charts, graphs, tables etc. We can create these dashboards by using Python programming language. In Python Dash is a framework developed by Plotly to create interactive dashboards. Dash allows us to build dashboards by combining Python's powerful libraries Plotly for interactive charts and Dash for building web layout. The following is a simple code to create a dashboard:

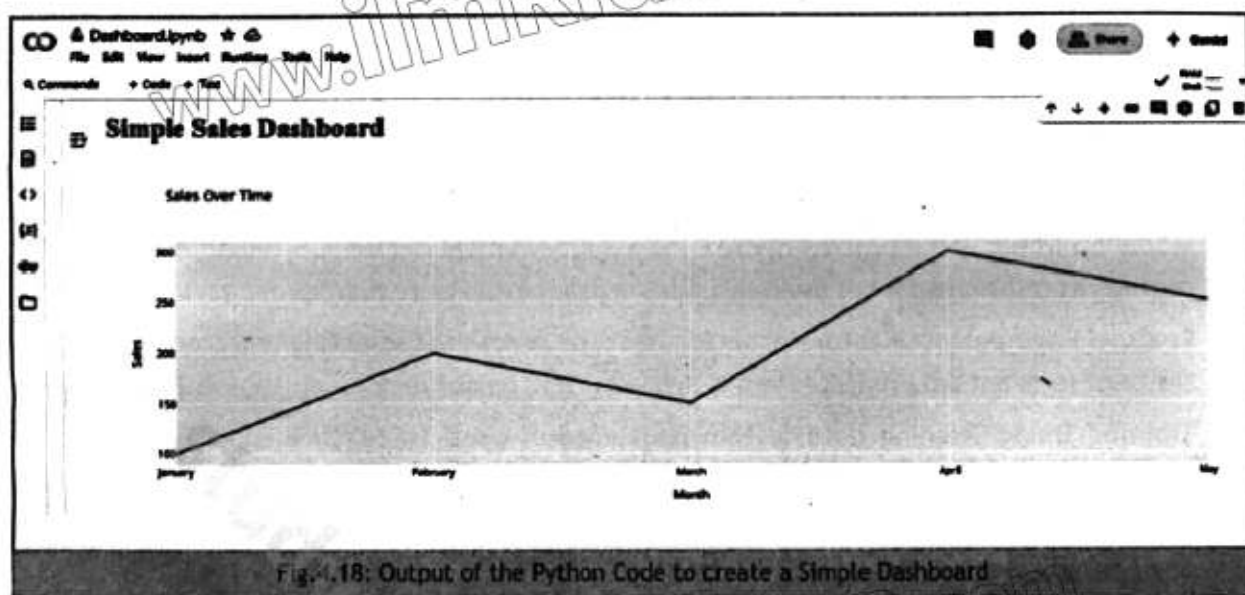


```
Dashboard.py
File Edit View Insert Runtime Tools Help

Q Commands + Code + Test

1 | !pip install dash # step 1: Import necessary libraries
2 | !pip install plotly
3 | import dash
4 | from dash import dcc,
5 | import plotly.express as px
6 | import pandas as pd
7 |
8 | # Step 2: Create sample data (e.g., sales data over months)
9 | data = pd.DataFrame({
10 |     "Month": ["January", "February", "March", "April", "May"],
11 |     "Sales": [100, 200, 150, 250, 210]
12 | })
13 |
14 | # Step 3: Create a simple line plot with Plotly
15 | fig = px.line(data, x="Month", y="Sales", title="Sales Over Time")
16 |
17 | # Step 4: Initialize the Dash app
18 | app = dash.Dash()
19 |
20 | # Step 5: Create the layout for the dashboard
21 | app.layout = html.Div(children=[
22 |     html.H1("Simple Sales Dashboard"), # Title of the Dashboard
23 |     dcc.Graph(
24 |         id="sales-graph",
25 |         figure=fig # Display the line plot
26 |     )
27 | ])
28 |
29 | # Step 6: Run the app
30 | if __name__ == '__main__':
31 |     app.run(debug=True)
```

Fig.4.17: Python Code to create a Simple Dashboard



www.ilmkidunya.com

Glossary

- **Machine learning models:** learn from data to make decisions.
- **Rule based algorithms:** follow fixed instructions. They do not change unless we deliberately update them.
- **Model building:** is the process of creating a mathematical or computational model that represents relationships between variables in data to make predictions or gain insights.
- **Feature Engineering:** Feature engineering is the process of selecting and transforming the most relevant data features to improve a model's performance.
- **Training Data:** Training data is the main dataset used to teach a model to make predictions or decisions.
- **Test Data:** Test data is a small portion of data that we decide to use for evaluation purposes of our Machine Learning model.
- **Test-Train Split:** A test-train split divides data into two parts i.e training data (main portion) and test data (smaller portion).
- **Overfitting:** It happens when a machine learning model learns the training data too well, including its noise and irrelevant details. This makes the model perform very well on the training data but poorly on new, unseen data because it's too specific to the training set.
- **Underfitting:** It occurs when a model is too simple and fails to capture the underlying patterns in the data. As a result, it performs poorly both on the training data and on new data, as it hasn't learned enough from the available information.
- **Model Assessment:** A Model assessment evaluates a model's performance using parameters like accuracy, precision, and recall.
- **Accuracy:** This is the percentage of total predictions that are correct. It tells how often the model is right overall.
- **Precision:** This measures how many of the predictions the model made for a certain class (e.g., "positive" cases) are actually correct. It's about the quality of the positive predictions.
- **Recall:** This tells how well the model identifies all the actual positive cases in the data. It shows the model's ability to find all the relevant instances.
- **Structured data:** is organized in tables with rows and columns.
- **Unstructured data:** includes text, images, and more.
- **Supervised learning:** uses labeled data to train models.
- **Unsupervised learning:** finds patterns in unlabeled data.
- **Regression algorithms:** predict continuous values like prices.

- **Classification algorithms:** sort data into categories. Clustering algorithms group similar data points together.
- **Neural networks:** are used for complex patterns and large data.
- **Predictive outcomes:** forecast future events based on existing data patterns.
- **Causality:** shows cause-and-effect relationships
- **Model interpretation** helps us understand how a model makes decisions. It explains why the model gives certain predictions.
- **In healthcare:** it predicts disease outbreaks and helps diagnose conditions.
- **In finance:** it detects fraudulent transactions to protect accounts.
- **Retailers** use it to personalize shopping experiences for customers. Transportation companies optimize delivery routes to save time.
- **In agriculture,** it monitors crop health for better yields.
- **Education systems** tailor learning materials to individual needs. Use cases show how machine learning can solve practical problems.
- **Entertainment:** platforms recommend movies or songs based on user preferences. They highlight the impact and benefits of applying machine learning.
- **Data storytelling:** uses visuals to communicate insights from data. It combines charts, graphs, and narratives to explain findings.
- **Reading and critiquing data stories** helps improve understanding. To get a clear understanding of the data story look for clarity and accuracy in published visuals.
- **Clarity of Purpose:** A good data story should have a clear goal or message. The first step is to understand the objective.
- **Data Accuracy and Source Credibility:** The accuracy of the data is fundamental. The sources of the data used in the story should be accurate and credible.
- **Relevance:** Relevance means that the data and narrative should be directly connected to the main topic or issue being discussed.
- **Insight:** refers to the deeper understanding that the data story provides beyond just presenting facts. A good data story should reveal patterns, trends, or conclusions that help the audience learn something new or make better decisions.
- **Persuasion:** Persuasion is about how well the data story influences or convinces the audience to take an action or consider a particular point of view
- **Formulating questions:** guides data exploration and analysis. Identify relevant data sets to answer these questions. Evaluate how well the data addresses the questions posed.
- **Descriptive statistical techniques:** include Mean, Median and Mode. Mean is the

average value of the data set. Median is the middle value when data is ordered. Mode is the most frequent value in the data.

- **Hypothesis formulation** is the process of making an educated guess or assumption about a relationship between variables based on existing data or prior knowledge.
- **Null Hypothesis (H_0):** This states that there is no effect or relationship between the variables. It assumes any observed differences are due to random chance.
- **Alternative Hypothesis (H_1):** This suggests there is a significant effect or relationship between variables.
- **P-values** measure how likely results are due to chance. A low P-value means strong evidence against the null hypothesis. A high P-value suggests weak evidence and possibly random results.
- **Significance Testing:** is a statistical method used to determine if a result is due to chance or it is statistically significant. Statistically significant means it is likely due to some real effect or factor.
- **Hypothesis testing:** evaluates assumptions using statistical methods. The Test Statistic measures how much data deviates from the null hypothesis. Hypothesis testing involves certain statistical procedures and terms.
- **Visuals:** help in presenting the results of hypothesis testing. We can use graphs and charts to show data distributions. These visuals can illustrate the test statistics and P-values.

Exercise



Select the best answer for the following Multiple-Choice Questions (MCQs).

1. The spam filter is used to do which of the following tasks:
 - a. It helps to delete all emails
 - b. It learns from past emails to identify spam emails
 - c. It increases the email storage capacity
 - d. It mark all the emails to spam
2. Overfitting in a Machine Learning Model leads to:
 - a. The model performs poorly on training data
 - b. The model performs well on training data but poorly on new data
 - c. The model performs equally well on all types of data
 - d. The model does not generate results at all
3. The feature engineering involves:
 - a. Collecting raw data
 - b. Transforming and selecting relevant data features
 - c. Testing the model's accuracy
 - d. Calculating the significance of the model
4. What is precision in a Machine Learning Model?
 - a. The percentage of total correct predictions
 - b. The measure of how many positive predictions are correct
 - c. The measure of how well the model identifies all actual positive cases
 - d. The percentage of training data used
5. How does continuous learning impact a Machine Learning Model?
 - a. It reduces the model's accuracy over time
 - b. It helps the model improve its performance by adapting to new data
 - c. It makes the model slower to process data
 - d. It simplifies the model's decision-making process
6. How does storytelling enhance the communication of data insights?
 - a. By focusing solely on statistical analysis
 - b. By using visuals and narrative to make data relatable and engaging
 - c. By ignoring complex data patterns.
 - d. By minimizing the use of charts and graphs
7. Which of the following statements describe a Null hypothesis?
 - a. It proposes that there is a significant effect or relationship between variables
 - b. It assumes that observed differences are due to random chance

- c. It is the hypothesis that researchers want to disprove
 - d. It is a statistical method used to measure variability
8. In hypothesis testing P-Value indicates:
- a. The probability that the null hypothesis is true
 - b. The probability that the results are due to some random chance
 - c. The significance level for the test statistics
 - d. The confidence interval for the hypothesis
9. Why is it important to use visuals effectively in hypotheses testing?
- a. To replace the need for statistical analysis
 - b. To present data in a more understandable manner and enhance communication
 - c. To make the data collection process easier
 - d. To avoid need of a hypothesis formulation
10. In evaluating the credibility of a data story, which factor is least important?
- a. Source of data
 - b. Clarity of the visual presentation
 - c. The time of day the data was collected
 - d. Relevance to the topic
11. A model shows 98% accuracy, but precision and recall are both below 60%. What is the most likely issue?
- a. The model is underfitting
 - b. The data is imbalanced
 - c. The model is overfitting
 - d. Labels are missing in training data
12. A retail analyst wants to segment customers to discover hidden shopping patterns without any labeled categories. Which approach is most appropriate?
- a. Logistic regression
 - b. Clustering
 - c. Decision trees
 - d. Linear regression



Give short answers to the following Short Response Questions (SRQs).

1. Explain what 'Feature Engineering' is and give an example related to image classification.
2. Describe the difference between 'Overfitting' and 'Underfitting' in Machine Learning.
3. What is the purpose of 'Test Data' in evaluating a Machine Learning model?
4. Define 'Model Assessment' and list two metrics used for it.
5. What is the main goal of 'Data Visualization' in data analysis?
6. What is 'Data Storytelling' and why is it important?
7. How does continuous learning benefit a Machine Learning model?
8. Explain the term 'Test-Train Split' and its importance in model training.
9. How does the significance level affect hypothesis testing?
10. Describe how you would identify and mitigate bias in a published data story.
11. What are the key steps in hypothesis testing and why are they important?

12. Explain the role of standard deviation in understanding data variability.
13. How can descriptive statistics be used to summarize a large data set effectively?
14. How does the concept of statistical significance support or challenge conclusions in experimental research? Explain with an example.
15. How can data visualization bridge the gap between technical analysis and non-technical audiences? Explain with a real-world scenario.



Give long answers to the following Extended Response Questions (ERQs).

1. Discuss the differences between Machine Learning models and Rule-based Algorithms, including their advantages and limitations.
2. Explain the process of model building in Machine Learning, including data collection, preprocessing, training, and evaluation.
3. Compare and contrast 'Predictive Models' and 'Causal Models,' discussing their uses and limitations.
4. Discuss how formulating good research questions influences the quality of data analysis. Provide an example of how a well-formulated question led to meaningful insights in a study.
5. Examine the process of hypothesis formulation and testing in a real-world context. How do null and alternative hypotheses guide research and decision-making? Illustrate with a relevant example.
6. Critique a published data story for its use of clarity and accuracy. Identify any misleading or biased elements and suggest improvements.
7. Explore the role of p-values in hypothesis testing. How do they help determine the significance of research findings, and what are their limitations?
8. Analyze the steps of simple hypothesis testing and their importance in validating research findings. How does each step contribute to drawing accurate conclusions?
9. Discuss the challenges in ensuring data accuracy and relevance in a data story. How can these challenges be addressed to improve the credibility of the data story?
10. Evaluate how feedback can be used to refine data storytelling techniques. Provide examples of how constructive criticism can lead to more effective data presentations.
11. Discuss the challenges in ensuring data accuracy and relevance in a data story. How can these challenges be addressed to improve the credibility of the data story?
12. Evaluate how feedback can be used to refine data storytelling techniques. Provide examples of how constructive criticism can lead to more effective data presentations.



Activity 1: Feature Engineering Exercise:

Objective: Identify and list features for a given dataset. i.e. images of the animals.

1. Divide the students into small groups depending on the available resources and strength of the class.
2. Helps the students to find some pet datasets from the internet. For example "Cats vs Dogs" dataset from <https://www.kaggle.com/c/dogs-vs-cats> or Oxford pet dataset from <https://www.robots.ox.ac.uk/~vgg/data/pets/>
3. Ask the students to select relevant features that can help classifying images of dogs, cats and horses.
4. Supervise the students to identify new data and make predictions about the unseen data based on the information learned from the dataset.

Outcome: This activity will help students understand how Machine Learning Models learn data and make predictions about new unseen data.



Activity 2: Data Story Critique.

Objective: Students will practice critiquing data stories by evaluating clarity accuracy insight and persuasion.

1. Divide the students into small groups depending on the strength of the class.
2. Assign or let them choose a data story of their own choice e.g. economic trends, social media usage, prices of smartphones, performance of various video games etc.
3. Each group evaluates their assigned story using the critique framework.
4. The group present their story to the class and provide feedback on how the data story can be improved

Outcome: This activity will help students develop critical thinking, collaboration, communication and data analysis.



Activity 3: Hypothesis Testing Role Play

Objective: To help students simulate hypothesis testing using real world examples.

1. Assign the students various roles like researchers, statisticians, subjects and create a scenario such as "Does increased study hours improve grades?"
2. Students playing the role of researchers will formulate null hypothesis and alternate hypothesis.
3. Students playing the role of statisticians will analyze data provided by the teacher.
4. Students playing the role of subjects will help interpret the results and make decisions based on data.

Outcome: This activity will help students to develop the skills of hypothesis formulation, statistical reasoning and decision making.



Activity 4: Implementation of Machine Learning by designing an interactive dashboard

Objective: To help students understand how to design and create interactive dashboards by simulating a real-world project.

1. Assign students different roles like: Data Collectors (gather and clean the sample data), Dashboard Designers (design how the dashboard will look), Graph Makers (create plots and graphs using Plotly), App Builders (set up the layout and run the Dash app)
2. Create a scenario like, The school principal wants a dashboard to track students' performance across subjects and attendance trends.
3. Students playing the role of Data Collectors will organize basic data, such as students' marks and attendance percentages.
4. Students playing the role of Dashboard Designers will sketch or plan which graphs and filters should be on the dashboard.
5. Students playing the role of Graph Makers will use Plotly to create charts like bar graphs, line charts, and pie charts.
6. Students playing the role of App Builders will use Dash to assemble everything into a working dashboard and run it.

Outcome: This activity will help students develop skills in Data organization, Data visualization, Basic app development, and Collaborative project work. They will also experience how real-world data dashboards are built by teams working together.