



Nutan Maharashtra Vidya Prasarak Mandal's
**NUTAN MAHARASHTRA INSTITUTE OF
 ENGINEERING AND TECHNOLOGY**



Department of Computer Engineering



LABORATORY MANUAL

SUBJECT: Laboratory Practice-III

[SUBJECT CODE: 410246]

CLASS: B.E. COMP

YEAR: 2023-24

PREPARED BY:

(Prof. Sopan Kshirsagar)

APPROVED BY:

H.O.D. [COMP]

Vision and Mission of the Institute

1. **Vision of the Institute**-To be a notable institution for providing quality technical education, ensuring ethical, moral, and holistic development of students.
2. **Mission of the Institute**- To nurture engineering graduates with highest technical competence, professionalism and problem solving skills to serve the needs of industry and society.

Vision and Mission of the Department

Department Vision:

Imbibing Quality Technical Education and Overall Development by Endowing Students with Societal and Ethical skills in Computer Engineers

Department Mission:

- ① To impart engineering knowledge and skills by adopting effective teaching-learning processes.
- ② To develop professional, entrepreneurial & research competencies encompassing continuous intellectual growth
- ③ To produce educated students to exhibit societal and ethical responsibilities in the working environment.

Program Outcomes

1. Engineering knowledge:

Graduates can apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to Civil Engineering related problems.

2. Problem analysis:

An ability to identify, formulate, review research literature, and analyse Civil engineering problems reaching substantiated conclusions using principles of mathematics and engineering sciences.

3. Design/development of solutions:

An ability to plan, analyse, design, and implement engineering problems and design system components or processes to meet the specified needs.

4. Conduct investigations of complex problems:

An ability to use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage:

An ability to apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

6. The engineer and society:

An ability to apply contextual knowledge to assess societal, legal issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability:

An ability to understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics:

An ability to apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and teamwork:

An ability to function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings to accomplish a common goal.

10. Communication:

An ability to communicate effectively on engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation and make effective presentations.

11. Project management and finance:

Ability to demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning:

An ability to engage in independent and life-long learning in the broadest context of technological change.

Program Educational Objectives (PEOs)

PEO1: To produce globally competent graduates having Excellent fundamentals, domain knowledge, updated with modern technology to provide effective solutions for computer engineering problems.

PEO2: To prepare the graduates to work as a committed professional with professional ethics and values, sense of responsibilities, understanding of legal, safety, health, societal, cultural and environmental issues.

PEO3: To prepare committed and motivated graduates with research attitude, lifelong learning, investigative approach, and multidisciplinary thinking.

PEO4: To produce the graduates with well-built managerial and communication skills to work effectively as individual as well as in teams.

Program Specific Outcomes (PSOs)

PSO 1 : Employ knowledge to write programs and design algorithms to integrate them with the hardware/software products in the domains of embedded systems, data Science, networking and web technology.

PSO 2 : Apply standard practices and strategies in project development using open-ended programming environments to create innovative career paths to be an entrepreneur, and a zest for higher studies.

Course Outcomes (CO)

Course Outcome	Statement
At the end of the course, a student will be able to	
CO1	Apply preprocessing techniques on datasets.
CO2	Implement and evaluate linear regression and random forest regression models.
CO3	Apply and evaluate classification and clustering techniques.
CO4	Analyze performance of an algorithm.
CO5	Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound.
CO6	Interpret the basic concepts in Blockchain technology and its applications.

List of Experiments with Mapping

Sr. No.	Group	Name of the Experiment	CO	PO
1.	A	Write a program to calculate Fibonacci numbers and find its step count.	CO4,C05	P01,P02,P03,P05,P08,P09,P011,P012
2.		Implement job sequencing with deadlines using a greedy method.	CO4,C05	P01,P02,P05,P08,P011,P01
3.		Write a program to solve a fractional Knapsack problem using a greedy method.	CO4,C05	P01,P02,P05,P08,P011,P01
4.		Write a program to solve a 0-1 Knapsack problem using dynamic programming or branch and bound strategy.	CO4,C05	P01,P02,P05,P08,P011,P01
5.		Write a program to generate binomial coefficients using dynamic programming.	CO4,C05	P01,P02,P05,P08,P011,P01
6.		Design 8-Queens matrix having first Queen placed. Use backtracking to place remaining Queens to generate the final 8-queen's matrix.	CO4,C05	P01,P02,P05,P08,P011,P01
7.		<p>Write a program to implement matrix multiplication. Also implement multithreaded matrix multiplication with either one thread per row or one thread per cell. Analyze and compare their performance.</p> <p>OR</p> <p>Implement merge sort and multithreaded merge sort. Compare time required by both the algorithms. Also analyze the performance of each algorithm for the best case and the worst case.</p> <p>OR</p> <p>Implement the Naive string matching algorithm and Rabin-Karp algorithm for string matching. Observe difference in working of both the algorithms for the same input.</p>	CO4,C05	P01,P02,P05,P08,P011,P01
8.		<p>Predict the price of the Uber ride from a given pickup point to the agreed drop-off location.</p> <p>Perform following tasks:</p> <ol style="list-style-type: none"> 1. Pre-process the dataset. 2. Identify outliers. 3. Check the correlation. 4. Implement linear regression and random forest regression models. 5. Evaluate the models and compare their respective scores like R2, RMSE, etc. 	CO1,CO2,CO3	P01,P02, P03,P04, P05,P06, P08,P09, P011,P01 2

		Dataset link: https://www.kaggle.com/datasets/yasserh/uber-fares-dataset		
9		<p>Classify the email using the binary classification method. Email Spam detection has two states: a) Normal State – Not Spam, b) Abnormal State – Spam. Use K-Nearest Neighbors and Support Vector Machine for classification. Analyze their performance.</p> <p>Dataset link: The emails.csv dataset on the Kaggle https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv</p>	C01,C02,C03	P01,P02,P03,P04,P05,P06,P08,P09,P010,P011,P012
10	B	<p>Given a bank customer, build a neural network-based classifier that can determine whether they will leave or not in the next 6 months.</p> <p>Dataset Description: The case study is from an open-source dataset from Kaggle.</p> <p>The dataset contains 10,000 sample points with 14 distinct features such as CustomerId, CreditScore, Geography, Gender, Age, Tenure, Balance, etc.</p> <p>Link to the Kaggle project: https://www.kaggle.com/barelydedicated/bank-customer-churn-modeling</p> <p>Perform following steps:</p> <ol style="list-style-type: none"> 1. Read the dataset. 2. Distinguish the feature and target set and divide the data set into training and test sets. 3. Normalize the train and test data. 4. Initialize and build the model. Identify the points of improvement and implement the same. 5. Print the accuracy score and confusion matrix (5 points). 	C01,C02,C03	P01,P02,P03,P04,P05,P06,P08,P09,P010,P011,P012
11		<p>Implement Gradient Descent Algorithm to find the local minima of a function.</p> <p>For example, find the local minima of the function $y=(x+3)^2$ starting from the point $x=2$.</p>	C01,C02,C03	P01,P02,P03,P04,P05,P06,P08,P09,P010,P011,P012
12		<p>Implement K-Nearest Neighbors algorithm on diabetes.csv dataset. Compute confusion matrix, accuracy, error rate, precision and recall on the given dataset.</p> <p>Dataset link :</p> <p>https://www.kaggle.com/datasets/abdallamahgoub/diabetes</p>	C01,C02,C03	P01,P02,P03,P04,P05,P06,P08,P09,P010,P011,P012
13		<p>Implement K-Means clustering/ hierarchical clustering on sales_data_sample.csv dataset.</p> <p>Determine the number of clusters using the elbow method.</p> <p>Dataset link :</p> <p>https://www.kaggle.com/datasets/kyanyoga/sample-sales-data</p>	C01,C02,C03	P01,P02,P03,P04,P05,P06,P08,P09,P010,P011,P012

14		<p>Mini Project</p> <p>Use the following dataset to analyze ups and downs in the market and predict future stock price returns based on Indian Market data from 2000 to 2020.</p> <p>Dataset Link: https://www.kaggle.com/datasets/sagara9595/stock-data</p> <p>OR</p> <p>Build a machine learning model that predicts the type of people who survived the Titanic shipwreck using passenger data (i.e. name, age, gender, socio-economic class, etc.).</p> <p>Dataset Link: https://www.kaggle.com/competitions/titanic/</p>	C01,C02,C03	P01,P02,P04,P05,P08,P09,P012
15		Installation of Metamask and study spending Ether per transaction.	C06	P01,P02,P03,P04,P05,P08,P09,,P012
16		Create your own wallet using Metamask for crypto transactions.	C06	P01,P02,P03,P04,P05,P08,P09,,P012
17	C	<p>Write a smart contract on a test network, for Bank account of a customer for following operations:</p> <ul style="list-style-type: none"> ❑ Deposit money ❑ Withdraw Money ❑ Show balance 		P01,P02,P03,P04,P05,P08,P09,,P012
18		<p>Write a program in solidity to create Student data.</p> <p>Use the following constructs:</p> <ul style="list-style-type: none"> ❑ Structures ❑ Arrays ❑ Fallback <p>Deploy this as smart contract on Ethereum and Observe the transaction fee and Gas values.</p>	C06	P01,P02,P03,P04,P05,P08,P09,,P012
19		Write a survey report on types of Blockchains and its real time use cases.	C06	P01,P02,P03,P04,P05,P08,P09,,P012
20		Mini Project: Create a dApp (de-centralized app) for e-voting system.	C06	P01,P02,P03,P04,P05,P08,P09,,P012

Rubrics for Evaluation

Sr. No	Evaluation Criteria	Marks for each Criteria	Rubrics
1	Timely submission	5 or 10	Punctuality reflects the work ethics. Students should reflect that work ethics by completing the lab assignments and reports in a timely manner.
2	Journal Presentation	5 or 10	Students are expected to prepare the journal. The journal presentation of the course should be complete, clear, and understandable.
3	Performance	5 or 10	After performance, the students should have good knowledge of the experiment.
4	Understanding	5 or 10	The student should be able to explain methodology used for designing and developing the program/solution. Student should clearly understand the purpose of the assignment and its outcome.
5	Oral	5 or 10	The student should be able to answer the questions related to the lab assignments.

Group A: Design and Analysis of Algorithms

EXPERIMENT NO. 01

Title: Write a program non-recursive and recursive program to calculate Fibonacci numbers and analyze their time and space complexity.

Objective: Students should be able to perform non-recursive and recursive programs to calculate Fibonacci numbers and analyze time and space complexity of non-recursive and recursive program.

Theory:

Introduction to Fibonacci numbers

- The Fibonacci series, named after Italian mathematician Leonardo Pisano Bogollo, later known as Fibonacci, is a series (sum) formed by Fibonacci numbers denoted as F_n . The numbers in Fibonacci sequence are given as: 0, 1, 1, 2, 3, 5, 8, 13, 21, 38, ...
- In a Fibonacci series, every term is the sum of the preceding two terms, starting from 0 and 1 as first and second terms. In some old references, the term '0' might be omitted.

What is the Fibonacci Series?

- The **Fibonacci series** is obtained by taking the sum of the previous two numbers in the series, given that the first and second terms are 0 and 1, respectively.
- In mathematical terms, the sequence F_n of Fibonacci numbers is defined by the recurrence relation

$$F_n = F_{n-1} + F_{n-2}$$

with seed values $F_0 = 0$ and $F_1 = 1$.

The following are different methods to get the nth Fibonacci number.

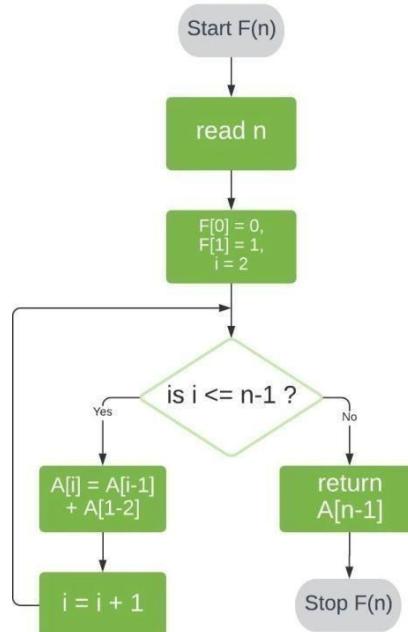
Method 1: Iterative Algorithm (Use Non-recursion)

A simple method that is a direct recursive implementation of mathematical recurrence relation is given above. First, we'll store 0 and 1 in $F[0]$ and $F[1]$, respectively.

Next, we'll iterate through array positions 2 to $n-1$. At each position i , we store the sum of the two preceding array values in $F[i]$.

Finally, we return the value of $F[n-1]$, giving us the number at position n in the sequence.

Here's a visual representation of this process:



Pseudocode to display the Fibonacci sequence up to n-th term

```

1  Algorithm Fibonacci( $n$ )
2  // Compute the  $n$ th Fibonacci number.
3  {
4    if ( $n \leq 1$ ) then
5      write ( $n$ );
6    else
7    {
8       $f_{nm2} := 0$ ;  $f_{nm1} := 1$ ;
9      for  $i := 2$  to  $n$  do
10     {
11        $f_n := f_{nm1} + f_{nm2}$ ;
12        $f_{nm2} := f_{nm1}$ ;  $f_{nm1} := f_n$ ;
13     }
14     write ( $f_n$ );
15   }
16 }
```

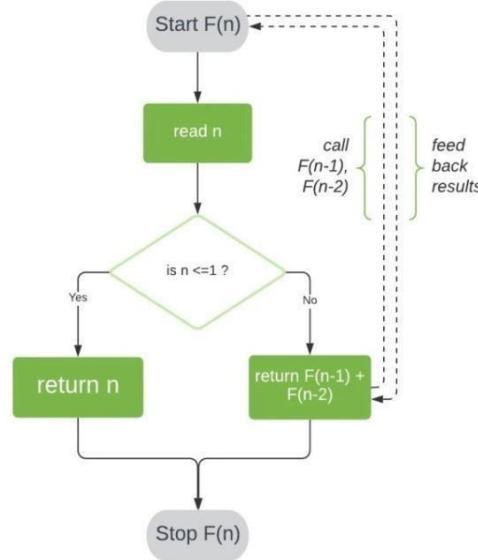
Time and Space Complexity of Space Optimized Method

- The time complexity of the Fibonacci series is $T(N)$ i.e, linear. We have to find the sum of two terms and it is repeated n times depending on the value of n .
- The space complexity of the Fibonacci series using dynamic programming is $O(1)$.

Method 2: Recursive Algorithm:

In recursion, the function calls itself until the base condition is met. To evaluate $F(n)$ for $n > 1$, we can reduce our problem into two smaller problems of the same kind: $F(n-1)$ and $F(n-2)$.

Here's a visual representation of this algorithm:

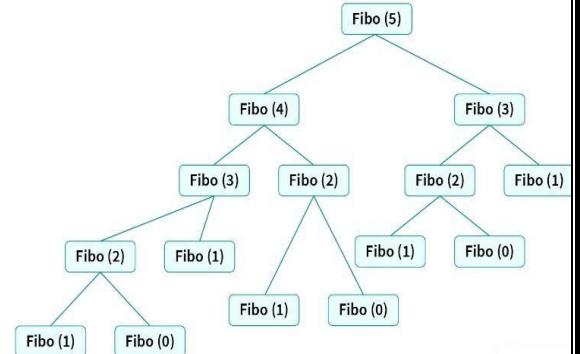


Here, the function `rec_Fibonacci()` makes a call to itself. This can be easily understood by the below illustration:

Algorithm rec_Fibonacci(n)

```

if(n <= 1)
    return n;
else
    return rFibonacci(n - 1) + rFibonacci(n - 2);
  
```



Analysis

$$\begin{aligned}
 T(n) &= T(n-1) + T(n-2) + c \\
 &= 2T(n-1) + c \\
 &= 2*(2T(n-2) + c) + c \\
 &= 4T(n-2) + 3c \\
 &= 8T(n-3) + 7c \\
 &= 2^k * T(n - k) + (2^k - 1)*c
 \end{aligned}
 \quad //\text{from the approximation } T(n-1) \sim T(n-2)$$

Let's find the value of k for which: $n - k =$

$$0k = n$$

$$T(n) = 2^n * T(0) + (2^n - 1)*c = 2^n * (1 + c) - c = 2^n$$

Time and Space Complexity

- The time complexity of the above code is $T(2^N)$, i.e., exponential.
- The Space complexity of the above code is $O(N)$ for a recursive series.

Using dynamic programming

In dynamic programming, we will store all the previously calculated values of the Fibonacci numbers in an array. We know that the array is zero-indexed. Therefore the Fibonacci numbers are stored in the $n-1$ array index. For example, the 2nd Fibonacci number is stored in the 1st index of the array.

Time Complexity and Space Complexity of Dynamic Programming

Space Optimized Method

In the previous method, we created an array to store the Fibonacci numbers but, as we need only the last two numbers to find the next one, it is **space-consuming** to store all the previously calculated numbers. Therefore we will use the space-optimized method, where just the previous two numbers are stored and using which the next number is found.

Time Complexity and Space Complexity of Space Optimized Method

- The time complexity of the Fibonacci series is **T(N)** i.e., **linear**. We have to find the sum of two terms, and it is repeated n times depending on the value of n.
- The space complexity of the Fibonacci series using dynamic programming is **O(1)**.

Method	Time complexity	Space complexity
Using recursion	$T(n) = T(n-1) + T(n-2)$	$O(n)$
Using DP	$O(n)$	$O(1)$
Space optimization of DP	$O(n)$	$O(1)$
Using the power of matrix method	$O(n)$	$O(1)$
Optimized matrix method	$O(\log n)$	$O(\log n)$
Recursive method in $O(\log n)$ time	$O(\log n)$	$O(n)$
Using direct formula	$O(\log n)$	$O(1)$
DP using memoization	$O(n)$	$O(1)$

Applications of Fibonacci Series

The Fibonacci series finds application in different fields in our day-to-day lives. The different patterns found in a varied number of fields from nature, to music, and to the human body follow the Fibonacci series.

Some of the applications of the series are given as,

- It is used in the grouping of numbers and used to study different other special mathematical sequences.
- It finds application in Coding (computer algorithms, distributed systems, etc). For example, Fibonacci series are important in the computational run-time analysis of Euclid's algorithm, used for determining the GCF of two integers.
- It is applied in numerous fields of science like quantum mechanics, cryptography, etc.
- In finance market trading, Fibonacci retracement levels are widely used in technical analysis.
- **Conclusion:**
In this way Concept of Fibonacci series is explored using recursive and non recursive method and also learn time and space complexity.

EXPERIMENT NO. 02

Title: Write a program to implement Huffman Encoding using a greedy strategy.

Objective: Students should be able to understand and solve Huffman Encoding and analyze time and space complexity using a greedy strategy.

Theory:

What is a Greedy Method?

- A greedy algorithm is an approach for solving a problem by selecting the best option available at the moment. It doesn't worry whether the current best result will bring the overall optimal result.
- The algorithm never reverses the earlier decision even if the choice is wrong. It works in a top-down approach.
- This algorithm may not produce the best result for all the problems. It's because it always goes for the local best choice to produce the global best result.

Advantages of Greedy Approach

- The algorithm is easier to describe.
- This algorithm can perform better than other algorithms (but, not in all cases).

Disadvantages of Greedy Approach

- As mentioned earlier, the greedy algorithm doesn't always produce the optimal solution. This is the major disadvantage of the algorithm
- For example, suppose we want to find the longest path in the graph below from root to leaf

Greedy Algorithm

1. To begin with, the solution set (containing answers) is empty.
2. At each step, an item is added to the solution set until a solution is reached.
3. If the solution set is feasible, the current item is kept.
4. Else, the item is rejected and never considered again.

Huffman Encoding

Huffman coding is a lossless data compression algorithm. The idea is to assign variable-length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding characters.

The most frequent character gets the smallest code and the least frequent character gets the largest code. The variable-length codes assigned to input characters are Prefix Codes means the codes (bit sequences) are assigned in such a way that the code assigned to one character is not prefix of code assigned to any other character. This is how Huffman Coding makes sure that there is no ambiguity when decoding the generated bit stream.

There are mainly two major parts in Huffman Coding

- 1) Build a Huffman Tree from input characters.
- 2) Traverse the Huffman Tree and assign codes to characters.

Steps to build Huffman Tree

Input is array of unique characters along with their frequency of occurrences and output is Huffman Tree.

1. Create a leaf node for each unique character and build a min heap of all leaf nodes (Min Heap is used as a priority queue. The value of frequency field is used to compare two nodes in min heap).

Initially, the least frequent character is at root)

2. Extract two nodes with the minimum frequency from the min heap.
3. Create a new internal node with frequency equal to the sum of the two nodes frequencies. Make the first extracted node as its left child and the other extracted node as its right child. Add this node to the min heap.
4. Repeat steps #2 and #3 until the heap contains only one node. The remaining node is the root node and the tree is complete.

Example:

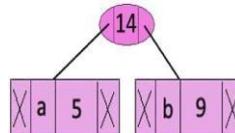
Let us understand the algorithm with an example:

character Frequency

a	5
b	9
c	12
d	13
e	16
f	45

Step 1: Build a min heap that contains 6 nodes where each node represents root of a tree with single node.

Step 2 : Extract two minimum frequency nodes from min heap. Add a new internal node with frequency $5 + 9 = 14$.



Now min heap contains 5 nodes where 4 nodes are roots of trees with single element each, and one heap node is root of tree with 3 elements,

Character

Frequency c 12

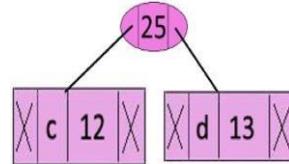
d 13

Internal Node 14

e 16

f 45

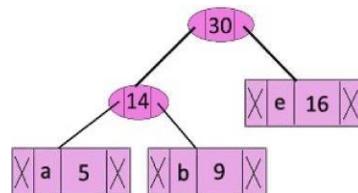
Step 3: Extract two minimum frequency nodes from heap. Add a new internal node with frequency $12 + 13 = 25$



Now min heap contains 4 nodes where 2 nodes are roots of trees with single element each, and twoheap nodes are root of tree with more than one nodes.

Character	Frequency
Internal Node	14
e	16
Internal Node	25
f	45

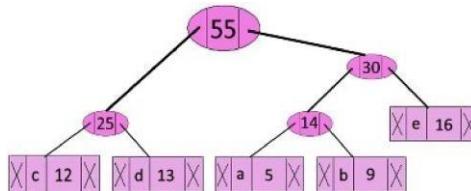
Step 4: Extract two minimum frequency nodes. Add a new internal node with frequency $14 + 16 = 30$



Now min heap contains 3 nodes.

Character	Frequency
Internal Node	25
Internal Node	30
f	45

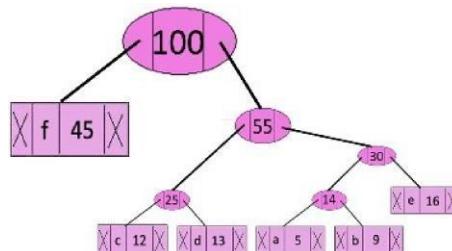
Step 5: Extract two minimum frequency nodes. Add a new internal node with frequency $25 + 30 = 55$



Now min heap contains 2 nodes.

Character	Frequency
f	45
Internal Node	55

Step 6: Extract two minimum frequency nodes. Add a new internal node with frequency $45 + 55 = 100$



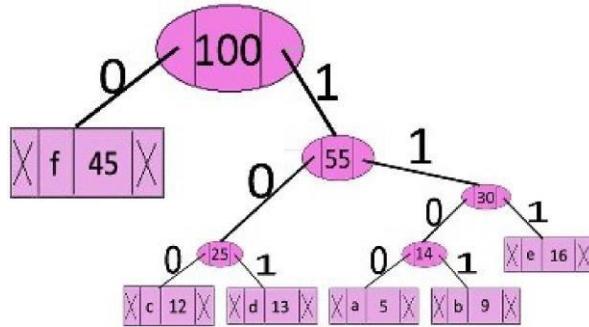
Now min heap contains only one node.

Character	Frequency
Internal Node	100

Since the heap contains only one node, the algorithm stops here.

Steps to print codes from Huffman Tree:

Traverse the tree formed starting from the root. Maintain an auxiliary array. While moving to the left child, write 0 to the array. While moving to the right child, write 1 to the array. Print the array when a leaf node is encountered.



The codes are as follows:

Character code-word

f	0
c	100
d	101
a	1100
b	1101
e	111

Algorithm for Huffman code

Input:-Number of message with frequency

count.Output: - Huffman merge tree.

1. *Begin*
2. *Let Q be the priority queue,*
3. *Q= {initialize priority queue with frequencies of all symbol or message}*
4. *Repeat n-1 times*
5. *Create a new node Z*
6. *X=extract_min(Q)*
7. *Y=extract_min(Q)*
8. *Frequency(Z) =Frequency(X) +Frequency(y);*
9. *Insert (Z, Q)*
10. *End repeat*
11. *Return (extract_min(Q))*
12. *End.*

Time Complexity-

$O(n \log n)$ where n is the number of unique characters. If there are n nodes, `extractMin()` is called $2^n - 1$ times. `extractMin()` takes $O(\log n)$ time as it calls `minHeapify()`. So, overall complexity is $O(n \log n)$.

Thus, Overall time complexity of Huffman Coding becomes $O(n \log n)$. If the input array is sorted, there exists a linear time algorithm.

Conclusion: In this way concept of Huffman Encoding is explored using greedy method.

EXPERIMENT NO. 03

Title: Write a program to solve a fractional Knapsack problem using a greedy method.

Objective: To analyze time and space complexity of fractional Knapsack problem using a greedy method.

Theory:

Knapsack Problem

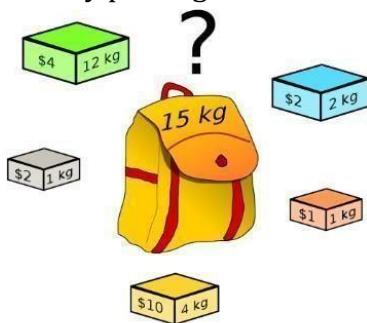
You are given the following-

- A knapsack (kind of shoulder bag) with limited weight capacity.
- Few items each having some weight and value.

The problem states-

Which items should be placed into the knapsack such that-

- The value or profit obtained by putting the items into the knapsack is maximum.
- And the weight limit of the knapsack does not exceed.



Knapsack Problem

Knapsack Problem Variants

Knapsack problem has the following two variants-

1. Fractional Knapsack Problem
2. 0/1 Knapsack Problem

Fractional Knapsack

Problem- In Fractional Knapsack Problem,

- As the name suggests, items are divisible here.
- We can even put the fraction of any item into the knapsack if taking the complete item is not possible.
- It is solved using the Greedy Method.

Fractional Knapsack Problem Using Greedy Method-

Fractional knapsack problem is solved using greedy method in the following steps-

Step-01: For each item, compute its value / weight ratio.

Step-02: Arrange all the items in decreasing order of their value / weight ratio.

Step-03: Start putting the items into the knapsack beginning from the item with the highest ratio. Put as many items as you can into the knapsack.

Example:

Find the optimal solution for the fractional knapsack problem making use of greedy approach. Consider-

$$n = 5$$

$$w = 60 \text{ kg}$$

$$(w_1, w_2, w_3, w_4, w_5) = (5, 10, 15, 22, 25)$$

$$(b_1, b_2, b_3, b_4, b_5) = (30, 40, 45, 77, 90)$$

Solution-

Step-01:

Compute the value / weight ratio for each item-

Items	Weight	Value	Ratio
1	5	30	6
2	10	40	4
3	15	45	3
4	22	77	3.5
5	25	90	3.6

Step-02:

Sort all the items in decreasing order of their value / weight ratio-

$$I1 \ I2 \ I5 \ I4 \ I3$$

$$(6) \ (4) \ (3.6) \ (3.5) \ (3)$$

Step-03:

Start filling the knapsack by putting the items into it one by one.

Knapsack Weight	Items in Knapsack	Cost
60	Ø	0
55	I1	30
45	I1, I2	70
20	I1, I2, I5	160

Now,

- Knapsack weight left to be filled is 20 kg but item-4 has a weight of 22 kg.
- Since in fractional knapsack problem, even the fraction of any item can be taken.
- So, knapsack will contain the following items-

< I1 , I2 , I5 , (20/22) I4 >

Total cost of the knapsack

$$\begin{aligned}
 &= 160 + (20/27) \times 77 \\
 &= 160 + 70 \\
 &= 230 \text{ units}
 \end{aligned}$$

Algorithm- Fractional knapsack

- Greedy-fractional-knapsack (w, v, W)
 1. for $i = 1$ to n
 2. do $x[i] = 0$
 3. weight = 0
 4. while weight $< W$
 5. do $i = \text{best remaining item}$
 6. if weight + $w[i] \leq W$
 7. then $x[i] = 1$
 8. weight = weight + $w[i]$
 9. else
 10. $x[i] = (w - \text{weight}) / w[i]$
 11. weight = W
 12. return x

Time Complexity-

- The main time taking step is the sorting of all items in decreasing order of their value / weightratio.
- If the items are already arranged in the required order, then while loop takes $O(n)$ time.
- The average time complexity of Quick Sort is $O(n\log n)$.
- Therefore, total time taken including the sort is $O(n\log n)$.

Viva Questions:

1. What is Greedy Approach?
2. Explain concept of fractional knapsack
3. Difference between Fractional and 0/1 Knapsack.

Conclusion: In this way concept of Fractional Knapsack is explained using greedy method.

Assignment No: 4

Title: Write a program to solve a 0-1 Knapsack problem using dynamic programming strategy.

Objective: To analyze time and space complexity of 0-1 Knapsack problem using dynamic programming.

Theory:

What is Dynamic Programming?

- Dynamic Programming is also used in optimization problems. Like divide-and-conquer method, Dynamic Programming solves problems by combining the solutions of subproblems.
- Dynamic Programming algorithm solves each sub-problem just once and then saves its answer in a table, thereby avoiding the work of re-computing the answer every time.
- Two main properties of a problem suggest that the given problem can be solved using Dynamic Programming. These properties are **overlapping sub-problems and optimal substructure**.
- Dynamic Programming also combines solutions to sub-problems. It is mainly used where the solution of one sub-problem is needed repeatedly. The computed solutions are stored in a table, so that these don't have to be re-computed. Hence, this technique is needed where overlapping sub- problem exists.
- For example, Binary Search does not have overlapping sub-problem. Whereas recursive program of Fibonacci numbers have many overlapping sub-problems.

Steps of Dynamic Programming Approach

Dynamic Programming algorithm is designed using the following four steps –

- Characterize the structure of an optimal solution.
- Recursively define the value of an optimal solution.
- Compute the value of an optimal solution, typically in a bottom-up fashion.
- Construct an optimal solution from the computed information.

Applications of Dynamic Programming Approach

- Matrix Chain Multiplication

- Longest Common Subsequence
- Travelling Salesman Problem

0/1 Knapsack Problem-

In 0/1 Knapsack Problem,

- As the name suggests, items are indivisible here.
- We can not take a fraction of any item.
- We have to either take an item completely or leave it completely.
- It is solved using a dynamic programming approach.

0/1 Knapsack Problem Using Greedy Method-

Consider-

- Knapsack weight capacity = w
- Number of items each having some weight and value = n

0/1 knapsack problem is solved using dynamic programming in the following steps-

Step-01:

- Draw a table say 'T' with $(n+1)$ number of rows and $(w+1)$ number of columns.
- Fill all the boxes of 0th row and 0th column with zeroes as shown-

	0	1	2	3	w
0	0	0	0	0	0
1	0					
2	0					
.....						
n	0					

T-Table

Step-02:

Start filling the table row wise top to bottom

from left to right. Use the following formula-

$$T(i, j) = \max \{ T(i-1, j), \text{value}_i + T(i-1, j - \text{weight}_i) \}$$

Here, $T(i, j)$ = maximum value of the selected items if we can take items 1 to i and have weight restrictions of j.

- This step leads to completely filling the table.
- Then, value of the last box represents the maximum possible value that can be put into the knapsack.

Step-03:

- To identify the items that must be put into the knapsack to obtain that maximum profit,
- Consider the last column of the table.
- Start scanning the entries from bottom to top.
- On encountering an entry whose value is not same as the value stored in the entry immediately above it, mark the row label of that entry.
- After all the entries are scanned, the marked labels represent the items that must be put into the knapsack

Problem-.

For the given set of items and knapsack capacity = 5 kg, find the optimal solution for the 0/1 knapsack problem making use of a dynamic programming approach.

Item	Weight	Value
1	2	3
2	3	4
3	4	5
4	5	6

$n = 4$

$w = 5 \text{ kg}$

$(w_1, w_2, w_3, w_4) = (2, 3, 4, 5)$

$(b_1, b_2, b_3, b_4) = (3, 4, 5, 6)$

Solution-

Given

- Knapsack capacity (w) = 5 kg
- Number of items (n) = 4

Step-01:

Draw a table say 'T' with $(n+1) = 4 + 1 = 5$ number of rows and $(w+1) = 5 + 1 = 6$ number of columns. Fill all the boxes of 0th row and 0th column with 0.

Identifying Items To Be Put Into Knapsack

Following Step-04,

- We mark the rows labelled "1" and "2".
- Thus, items that must be put into the knapsack to obtain the maximum value 7 are **Item-1 and Item-2**.

Time Complexity-

- Each entry of the table requires constant time $\theta(1)$ for its computation.
- It takes $\theta(nw)$ time to fill $(n+1)(w+1)$ table entries.
- It takes $\theta(n)$ time for tracing the solution since tracing process traces the n rows.

Conclusion: In this way we have explored Concept of 0/1 Knapsack using Dynamic approach.

Assignment No: 5

Title: Write a program for analysis of quick sort by using deterministic and randomized variant.

Objective: To analyze time and space complexity of quick sort by using deterministic and randomized variant.

Theory:

What is a Randomized Algorithm?

- An algorithm that uses random numbers to decide what to do next anywhere in its logic is called Randomized Algorithm..
- For example, in Randomized Quick Sort, we use random number to pick the next pivot (or we randomly shuffle the array).
- Typically, this randomness is used to reduce time complexity or space complexity in other standard algorithms.
- Randomized algorithm for a problem is usually simpler and **more efficient** than its deterministic counterpart.
- The output or the running time are functions of the input and random bits chosen.

Types of Randomized Algorithms

1. Las Vegas Algorithms

- These algorithms always produce correct or optimum result.
- Time complexity of these algorithms is based on a random value and time complexity is evaluated as expected value.
- For example, Randomized QuickSort always sorts an input array and expected worst case time complexity of QuickSort is $O(n \log n)$.
- A Las Vegas algorithm fails with some probability, but we can tell when it fails. In particular, we can run it again until it succeeds, which means that we can eventually succeed with probability 1.
- Alternatively, we can think of a Las Vegas algorithm as an algorithm that runs for an unpredictable amount of time but always succeeds

2. Monte Carlo Algorithms

- Produce correct or optimum result with some probability.
- These algorithms have deterministic running time and it is generally easier to find out worst case time complexity.
- For example Karger's Algorithm produces minimum cut with probability greater than or equal to $1/n^2$ (n is number of vertices) and has worst case time complexity as $O(E)$.
- A Monte Carlo algorithm fails with some probability, but we can't tell when it fails.
- The polynomial equality-testing algorithm is an example of a Monte Carlo algorithm

Applications of Randomized Algorithms

- Randomized algorithms have huge applications in Cryptography.
- Load Balancing.
- Number-Theoretic Applications: Primality Testing
- Data Structures: Hashing, Sorting, Searching, Order Statistics and ComputationalGeometry.
- Algebraic identities: Polynomial and matrix identity verification. Interactive proof systems.
- Mathematical programming: Faster algorithms for linear programming, Rounding linearprogram solutions to integer program solutions

Analysis of Randomized Quick sort

The running time of quicksort depends mostly on the number of comparisons performed in all calls to the Randomized-Partition routine. Let X denote the random variable counting the number of comparisons in all calls to Randomized-Partition.

Let z_i denote the i -th smallest element of $A[1..n]$.

Thus $A[1..n]$ sorted is $\langle z_1, z_2, \dots, z_n \rangle$.

Let $Z_{ij} = \{z_i, \dots, z_j\}$ denote the set of elements between z_i and z_j , including these elements.

$X_{ij} = I\{ z_i \text{ is compared to } z_j \}$.

Thus, X_{ij} is an indicator random variable for the event that the i -th smallest and the j -th smallest elements of A are compared in an execution of quicksort.

Number of Comparisons

Since each pair of elements is compared at most once by quicksort, the number X of comparisons is given by

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^n X_{ij}$$

Therefore, the expected number of comparisons is

$$E[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^n E[X_{ij}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \Pr[z_i \text{ is compared to } z_j]$$

Expected Number of Comparisons

Conclusion: In this way we have explored Concept of quick sort by using deterministic and randomized variant.

Assignment No: 6

Mini Project

Mini Project - Write a program to implement matrix multiplication. Also implement multithreadedmatrix multiplication with either one thread per row or one thread per cell. Analyze and compare their performance.

Mini Project - Implement merge sort and multithreaded merge sort. Compare time required by boththe algorithms. Also analyze the performance of each algorithm for the best case and the worst case.

Mini Project - Different exact and approximation algorithms for Travelling-Sales-Person Problem

Objective: To implement and analyze performance of algorithm.

Theory:

1. Introduction to problems
2. Approach used to solve problem (Introduction with example)
3. Algorithm/ Pseudo code of problems
4. Complexity Analyze for all cases
5. Implementation of project with output

Group B

Assignment no:1

Title of the Assignment: Predict the price of the Uber ride from a given pickup point to the agreed drop-off location. Perform following tasks:

1. Pre-process the dataset.
2. Implement linear regression and random forest regression models.
3. Evaluate the models and compare their respective scores like R2, RMSE, etc.

Dataset Description: The project is about on world's largest taxi company Uber inc. In this project, we're looking to predict the fare for their future transactional cases. Uber delivers service to lakhs of customers daily. Now it becomes really important to manage their data properly to come up with new business ideas to get best results. Eventually, it becomes really important to estimate the fare prices accurately.

Link for Dataset: <https://www.kaggle.com/datasets/yasserh/uber-fares-dataset>

Objective of the Assignment:

Students should be able to preprocess dataset and identify outliers, to check correlation and implement linear regression and random forest regression models. Evaluate them with respective scores like R2, RMSE etc.

Prerequisite:

1. Basic knowledge of Python
2. Concept of preprocessing data
3. Basic knowledge of Data Science and Big Data Analytics.

Contents of the Theory:

1. Data Preprocessing
2. Linear regression
3. Random forest regression models
4. Box Plot
5. Outliers
6. Haversine
7. Matplotlib
8. Mean Squared Error

Data Preprocessing:

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

Why do we need Data Preprocessing?

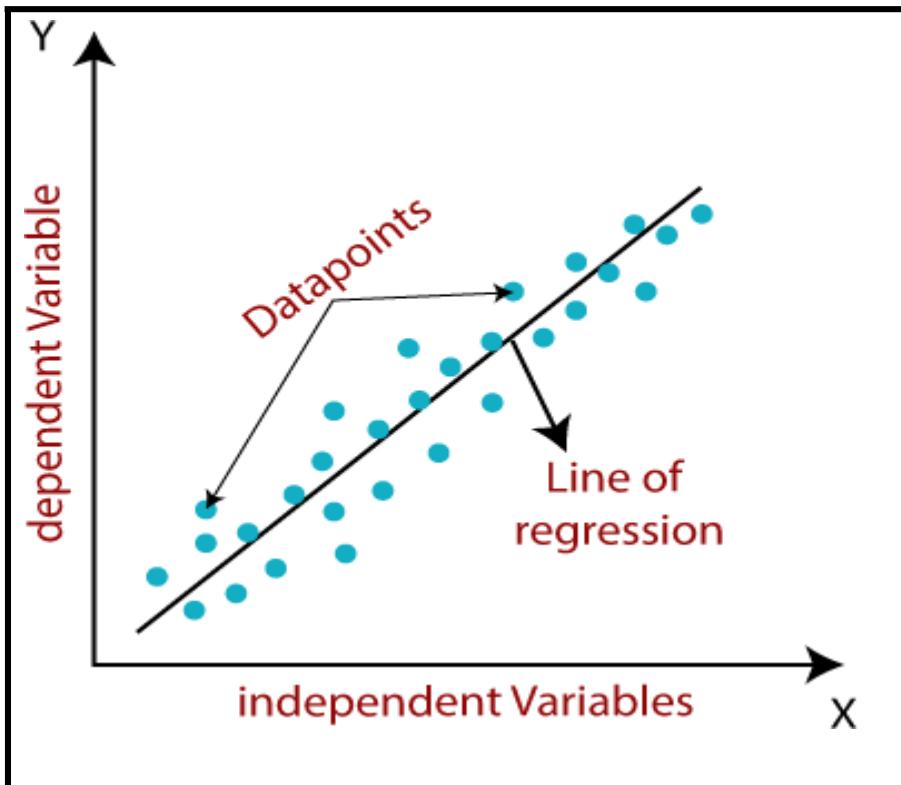
A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy.

Linear Regression:

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price, etc.**

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (x) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:



Random Forest Regression Models:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning**, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

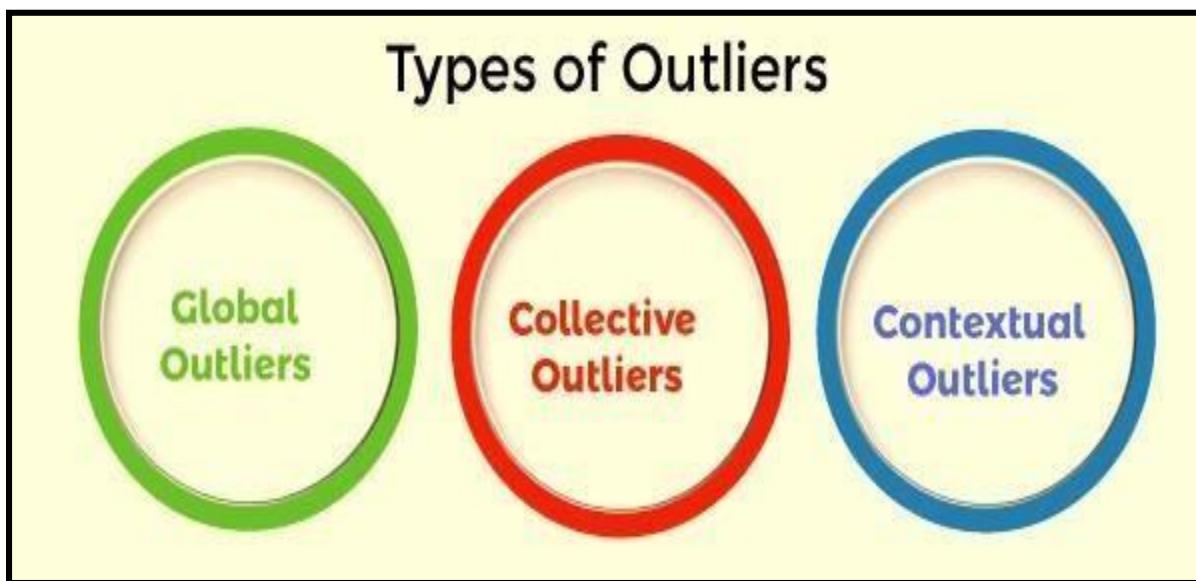
As the name suggests, "**Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.**" Instead of relying on one decision tree, the random forest takes the

prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Outlier:

The major thing about the outliers is what you do with them. If you are going to analyze any task to analyze data sets, you will always have some assumptions based on how this data is generated. If you find some data points that are likely to contain some form of error, then these are definitely outliers, and depending on the context, you want to overcome those errors. The data mining process involves the analysis and prediction of data that the data holds. In 1969, Grubbs introduced the first definition of outliers.

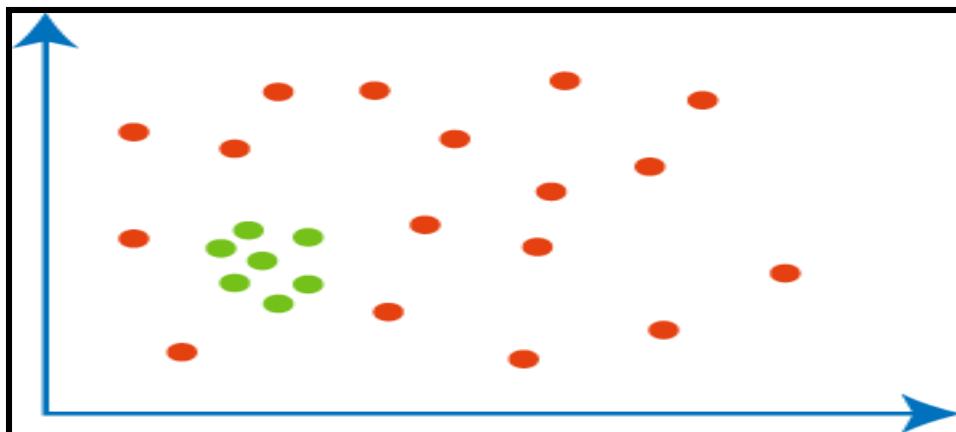


Global Outliers

Global outliers are also called point outliers. Global outliers are taken as the simplest form of outliers. When data points deviate from all the rest of the data points in a given data set, it is known as the global outlier. In most cases, all the outlier detection procedures are targeted to determine the global outliers. The green data point is the global outlier.

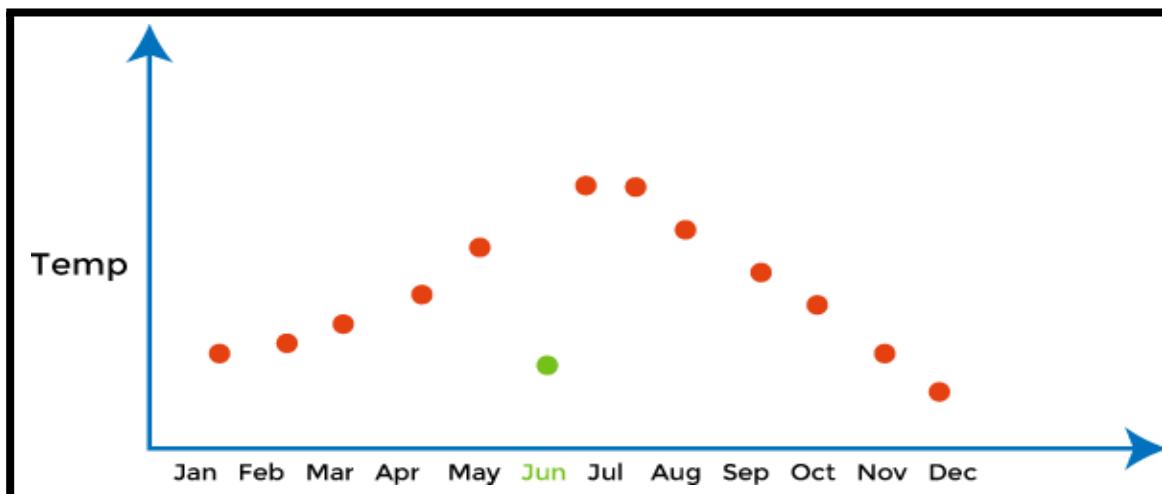
Collective Outliers

In a given set of data, when a group of data points deviates from the rest of the data set is called collective outliers. Here, the particular set of data objects may not be outliers, but when you consider the data objects as a whole, they may behave as outliers. To identify the types of different outliers, you need to go through background information about the relationship between the behavior of outliers shown by different data objects. For example, in an Intrusion Detection System, the DOS package from one system to another is taken as normal behavior. Therefore, if this happens with the various computer simultaneously, it is considered abnormal behavior, and as a whole, they are called collective outliers. The green data points as a whole represent the collective outlier.



Contextual Outliers

As the name suggests, "Contextual" means this outlier introduced within a context. For example, in the speech recognition technique, the single background noise. Contextual outliers are also known as Conditional outliers. These types of outliers happen if a data object deviates from the other data points because of any specific condition in a given data set. As we know, there are two types of attributes of objects of data: contextual attributes and behavioral attributes. Contextual outlier analysis enables the users to examine outliers in different contexts and conditions, which can be useful in various applications. For example, A temperature reading of 45 degrees Celsius may behave as an outlier in a rainy season. Still, it will behave like a normal data point in the context of a summer season. In the given diagram, a green dot representing the low-temperature value in June is a contextual outlier since the same value in December is not an outlier.



Haversine:

The Haversine formula calculates the shortest distance between two points on a sphere using their latitudes and longitudes measured along the surface. It is important for use in navigation.

Matplotlib:

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Mean Squared Error:

The **Mean Squared Error (MSE)** or **Mean Squared Deviation (MSD)** of an estimator

measures the average of error squares i.e. the average squared difference between the estimated values and true value. It is a risk function, corresponding to the expected value of the squared error loss. It is always non-negative and values close to zero are better.

Conclusion:

In this way we have explored Concept correlation and implement linear regression and random forest regression models.

Assignment no:2

Title of the Assignment: Classify the email using the binary classification method. EmailSpam detection has two states:

- a) Normal State – Not Spam,
- b) Abnormal State – Spam.

Use K-Nearest Neighbors and Support Vector Machine for classification. Analyze their performance.

Dataset Description: The csv file contains 5172 rows, each row for each email. There are 3002 columns. The first column indicates Email name. The name has been set with numbers and not recipients' name to protect privacy. The last column has the labels for prediction : 1 for spam, 0 for not spam. The remaining 3000 columns are the 3000 most common words in all the emails, after excluding the non-alphabetical characters/words. For each row, the count of each word(column) in that email(row) is stored in the respective cells. Thus, information regarding all 5172 emails are stored in a compact dataframe rather than as separate text files.

Link: <https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv>

Objective of the Assignment:

Students should be able to classify email using the binary Classification and implement email spam detection technique by using K-Nearest Neighbors and Support Vector Machine algorithm.

Prerequisite:

- 1. Basic knowledge of Python
- 2. Concept of K-Nearest Neighbors and Support Vector Machine for classification.

Contents of the Theory:

1. Data Preprocessing
2. Binary Classification
3. K-Nearest Neighbours
4. Support Vector Machine
5. Train, Test and Split Procedure

Data Preprocessing:

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

Why do we need Data Preprocessing?

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set
- Feature scaling

Assignment no:3

Title of the Assignment: Given a bank customer, build a neural network-based classifier that can determine whether they will leave or not in the next 6 months

Dataset Description: The case study is from an open-source dataset from Kaggle. The dataset contains 10,000 sample points with 14 distinct features such as CustomerId, CreditScore, Geography, Gender, Age, Tenure, Balance, etc.

Link for Dataset: <https://www.kaggle.com/barelydedicated/bank-customer-churn-modeling>

Perform the following steps:

1. Read the dataset.
2. Distinguish the feature and target set and divide the data set into training and test sets.
3. Normalize the train and test data.
4. Initialize and build the model. Identify the points of improvement and implement the same.
5. Print the accuracy score and confusion matrix (5 points).

Objective of the Assignment:

Students should be able to distinguish the feature and target set and divide the data set into training and test sets and normalize them and students should build the model on the basis of that.

Prerequisite:

1. Basic knowledge of Python
2. Concept of Confusion Matrix

Contents of the Theory:

1. Artificial Neural Network
2. Keras
3. tensorflow
4. Normalization
5. Confusion Matrix

- An **Artificial Neural Network** in the field of **Artificial intelligence** where it attempts to mimic the network of neurons makes up a human brain so that computers will have the option to understand things and make decisions in a human-like manner. The artificial neural network is designed by programming computers to behave simply like interconnected brain cells.

There are around 1000 billion neurons in the human brain. Each neuron has an association point somewhere in the range of 1,000 and 100,000. In the human brain, data is stored in such a manner as to be distributed, and we can extract more than one piece of this data when necessary from our memory parallelly. We can say that the human brain is made up of incredibly amazing parallel processors.

We can understand the artificial neural network with an example, consider an example of a digital logic gate that takes an input and gives an output. "OR" gate, which takes two inputs. If one or both the inputs are "On," then we get "On" in output. If both the inputs are "Off," then we get "Off" in output. Here the output depends upon input. Our brain does not perform the same task. The outputs to inputs relationship keep changing because of the neurons in our brain, which are "learning."

The architecture of an artificial neural network:

- To understand the concept of the architecture of an artificial neural network, we have to understand what a neural network consists of. In order to define a neural network that consists of a large number of artificial neurons, which are termed units arranged in a sequence of layers. Let's us look at various types of layers available in an artificial neural network.

Confusion Matrix:

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an **error matrix**. Some features of Confusion

matrix are given below:

- For the 2 prediction classes of classifiers, the matrix is of 2*2 table, for 3 classes, it is 3*3 table, and so on.
- The matrix is divided into two dimensions, that are **predicted values** and **actual values** along with the total number of predictions.
- Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.

Need for Confusion Matrix in Machine learning

- It evaluates the performance of the classification models, when they make predictions on test data, and tells how good our classification model is.
- It not only tells the error made by the classifiers but also the type of errors such as it is either type-I or type-II error.
- With the help of the confusion matrix, we can calculate the different parameters for the model, such as accuracy, precision, etc.

Conclusion: In this way we build a neural network.

Assignment no:4

Title of the Assignment: Implement K-Nearest Neighbors algorithm on diabetes.csv dataset. Compute confusion matrix, accuracy, error rate, precision and recall on the given dataset.

Dataset Description: We will try to build a machine learning model to accurately predict whether or not the patients in the dataset have diabetes or not?

The datasets consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

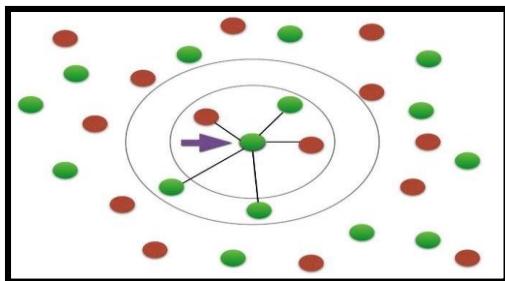
Link for Dataset: [Diabetes predication system with KNN algorithm | Kaggle](#)

Objective of the Assignment:

Students should be able to preprocess dataset and identify outliers, to check correlation and implement KNN algorithm and random forest classification models. Evaluate them with respective scores like confusion_matrix, accuracy_score, mean_squared_error, r2_score, roc_auc_score, roc_curve etc.

Prerequisite:

1. Basic knowledge of Python
2. Concept of Confusion Matrix
3. Concept of roc_auc curve.
4. Concept of Random Forest and KNN algorithms



k-Nearest-Neighbors (k-NN) is a supervised machine learning model. Supervised learning is when a model learns from data that is already labeled. A supervised learning model takes in a set of input objects and output values. The model then trains on that data to learn how to map the inputs to the desired output so it can learn to make predictions on unseen data.

k-NN models work by taking a data point and looking at the 'k' closest labeled data points. The data point is then assigned the label of the majority of the 'k' closest points.

For example, if $k = 5$, and 3 of points are 'green' and 2 are 'red', then the data point in question would be labeled 'green', since 'green' is the majority (as shown in the above graph).

Scikit-learn is a machine learning library for Python. In this tutorial, we will build a k-NN model using Scikit-learn to predict whether or not a patient has diabetes.

Reading in the training data

For our k-NN model, the first step is to read in the data we will use as input. For this example, we are using the diabetes dataset. To start, we will use Pandas to read in the data. I will not go into detail on Pandas, but it is a library you should become familiar with if you're looking to dive further into data science and machine learning.

Split the dataset into train and test data

Now we will split the dataset into training data and testing data. The training data is the data that the model will learn from. The testing data is the data we will use to see how well the model performs on unseen data.

Scikit-learn has a function we can use called 'train_test_split' that makes

it easy for us to split our dataset into training and testing data.

```
from sklearn.model_selection import train_test_split#split dataset into train and test data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1, stratify=y)
```

'train_test_split' takes in 5 parameters. The first two parameters are the input and target data we split up earlier. Next, we will set 'test_size' to 0.2. This means that 20% of all the data will be used for testing, which leaves 80% of the data as training data for the model to learn from. Setting

'random_state' to 1 ensures that we get the same split each time so we can reproduce our results.

Setting 'stratify' to y makes our training split represent the proportion of each value in the y variable. For example, in our dataset, if 25% of patients have diabetes and 75% don't have diabetes, setting 'stratify' to y will ensure that the random split has 25% of patients with diabetes and 75% of patients without diabetes.

Building and training the model

Next, we have to build the model. Here is the code:

```
from sklearn.neighbors import KNeighborsClassifier#
Create      KNN      classifier      knn      =
KNeighborsClassifier(n_neighbors = 3)# Fit the
```

First, we will create a new k-NN classifier and set 'n_neighbors' to 3. To recap, this means that if at least 2 out of the 3 nearest points to a new data point are patients without diabetes, then the new data point will be labeled as 'no diabetes', and vice versa. In other words, a new data point is labeled with by majority from the 3 nearest points.

We have set 'n_neighbors' to 3 as a starting point. We will go into more detail below on how to better select a value for 'n_neighbors' so that the model can improve its performance.

Conclusion:

In this way we build a neural network-based classifier that can determine whether they will leave or not in the next 6 months

Assignment no:5

Title of the Assignment: Implement K-Means clustering/hierarchical clustering on sales_data_sample.csv dataset.
Determine the number of clusters using the elbow method.

Dataset Description: The data includes the following features:

1. Customer ID
2. Customer Gender
3. Customer Age
4. Annual Income of the customer (in Thousand Dollars)
5. Spending score of the customer (based on customer behavior and spending nature)

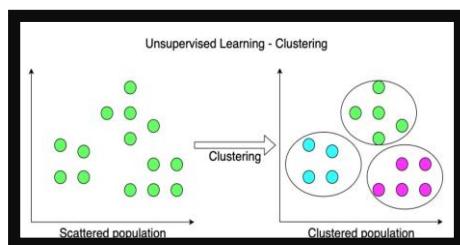
Objective of the Assignment:

Students should able to understand how to use unsupervised learning to segment different-different clusters or groups and used to them to train your model to predict future things.

Prerequisite:

1. Knowledge of Python
2. Unsupervised learning
3. Clustering
4. Elbow method

Clustering algorithms try to find natural clusters in data, the various aspects of how the algorithms to cluster data can be tuned and modified. Clustering is based on the principle that items within the same cluster must be similar to each other. The data is grouped in such a way that related elements are close to each other.



Diverse and different types of data are subdivided into smaller groups.

Uses of Clustering

Marketing:

In the field of marketing, clustering can be used to identify various customer groups with existing customer data. Based on that, customers can be provided with discounts, offers, promo codes etc.

BookStore and Library management:

Libraries and Bookstores can use Clustering to better manage the book database. With proper book ordering, better operations can be implemented.

Document Analysis:

Often, we need to group together various research texts and documents according to similarity. And in such cases, we don't have any labels. Manually labelling large amounts of data is also not possible. Using clustering, the algorithm can process the text and group it into different themes.

These are some of the interesting use cases of clustering.

K-Means Clustering

K-Means clustering is an unsupervised machine learning algorithm that divides the given data into the given number of clusters. Here, the "K" is the given number of predefined clusters, that need to be created.

It is a centroid based algorithm in which each cluster is associated with a centroid. The main idea is to reduce the distance between the data points and their respective cluster centroid.

The algorithm takes raw unlabelled data as an input and divides the dataset into clusters and the process is repeated until the best clusters are found.

K-Means is very easy and simple to implement. It is highly scalable, can be applied to both small and large datasets. There is, however, a problem with choosing the number of clusters or K. Also, with the increase in dimensions, stability decreases. But, overall K Means is a simple and robust algorithm that makes clustering very easy.

```
#Importing the  
necessary  
librariesimport  
numpy as np  
import pandas as pd  
import  
matplotlib.py  
plot as plt  
import  
seaborn as  
sns  
from mpl_toolkits.mplot3d import Axes3D  
%matplotlib inline
```

Conclusion: We can clearly see that 5 different clusters have been formed from the data. The red cluster is the customers with the least income and least spending score, similarly, the blue cluster is the customers with the most income and most spending score.

Group C

Assignment no:1

Title of the Assignment: Installation of MetaMask and study spending Ether per transaction

Objective of the Assignment: Students should be able to learn new technology such as metamask. Its application and implementations

Prerequisite:

1. Basic knowledge of cryptocurrency
2. Basic knowledge of distributed computing concept
3. Working of blockchain

Introduction to Blockchain

- Blockchain can be described as a data structure that holds transactional records and while ensuring security, transparency, and decentralization. You can also think of it as a chain or records stored in the forms of blocks which are controlled by no single authority.
- A blockchain is a distributed ledger that is completely open to any and everyone on the network. Once an information is stored on a blockchain, it is extremely difficult to change or alter it.
- Each transaction on a blockchain is secured with a digital signature that proves its authenticity. Due to the use of encryption and digital signatures, the data stored on the blockchain is tamper-proof and cannot be changed.
- Blockchain technology allows all the network participants to reach an agreement, commonly known as consensus. All the data stored on a blockchain is recorded digitally and has a common history which is available for all the network participants. This way, the chances of any fraudulent activity or duplication of transactions is eliminated without the need of a

third-party.

Blockchain Features

The following features make the revolutionary technology of blockchain stand out:

- *Decentralized*

Blockchains are decentralized in nature meaning that no single person or group holds the authority of the overall network. While everybody in the network has the copy of the distributed ledger with them, no one can modify it on his or her own. This unique feature of blockchain allows transparency and security while giving power to the users.

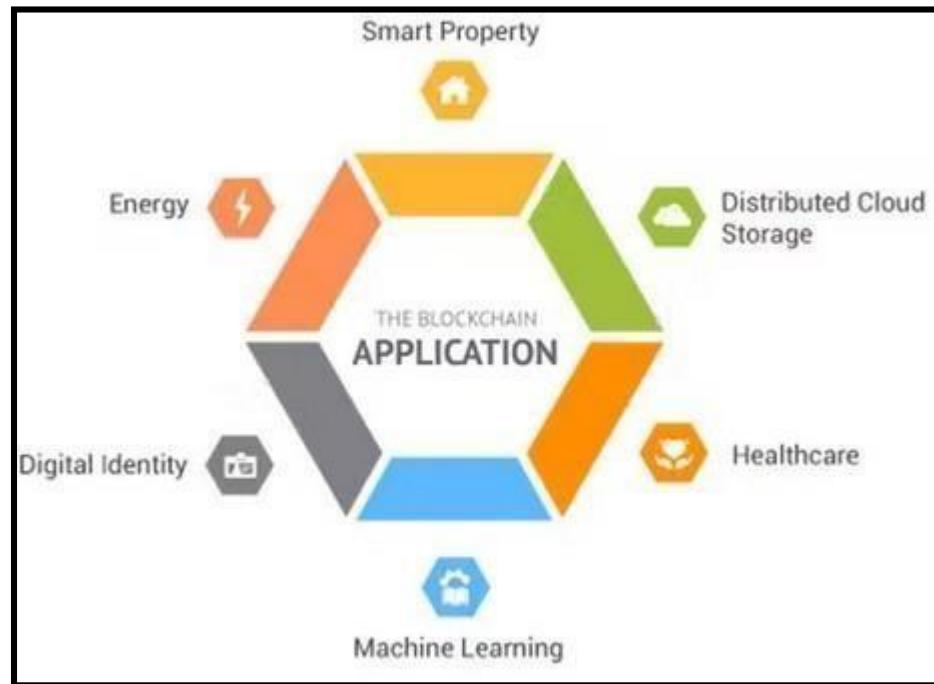
- *Immutable*

The immutability property of a blockchain refers to the fact that any data once written on the blockchain cannot be changed. To understand immutability, consider sending email as an example. Once you send an email to a bunch of people, you cannot take it back. In order to find a way around, you'll have to ask all the recipients to delete your email which is pretty tedious. This is how immutability works.

- *Tamper-Proof*

With the property of immutability embedded in blockchains, it becomes easier to detect tampering of any data. Blockchains are considered tamper-proof as any change in even one single block can be detected and addressed smoothly. There are two key ways of detecting tampering namely, hashes and blocks.

Popular Applications of Blockchain Technology



Benefits of Blockchain Technology:

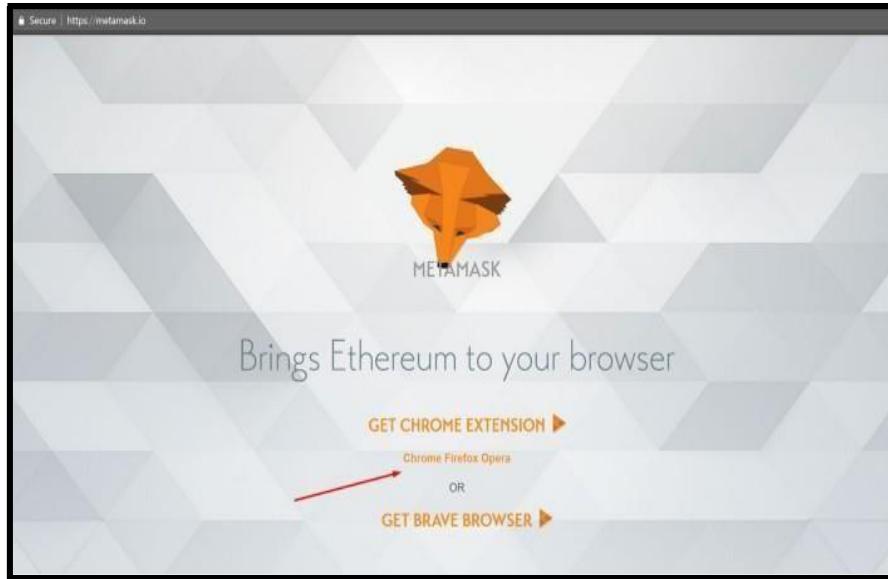
- **Time-saving:** No central Authority verification needed for settlements making the process faster and cheaper.
- **Cost-saving:** A Blockchain network reduces expenses in several ways. No need for third-party verification. Participants can share assets directly. Intermediaries are reduced. Transaction efforts are minimized as every participant has a copy of shared ledger.
- **Tighter security:** No one can tamper with Blockchain Data as it is shared among millions of participants. The system is safe against cybercrimes and Fraud.
- In finance market trading, Fibonacci retracement levels are widely used in technical analysis.

How to use MetaMask: A step by step guide

MetaMask is one of the most popular browser extensions that serves as a way of storing your Ethereum and other [ERC-20 Tokens](#). The extension is free and secure, allowing web applications to read and interact with Ethereum's blockchain.

Step 1. Install MetaMask on your browser.

To create a new wallet, you have to install the extension first. Depending on your browser, there are different marketplaces to find it. Most browsers have MetaMask on their stores, so it's not that hard to see it, but either way, here they are [Chrome](#), [Firefox](#), and [Opera](#).



- Click on **Install MetaMask** as a Google Chrome extension.
- Click **Add to Chrome**.
- Click **Add Extension**.

Step 2. Create an account.

- Click on the extension icon in the upper right corner to open MetaMask.
- To install the latest version and be up to date, [click Try it now](#).
- **Click Continue**.
- You will be prompted to create a new password. [Click Create](#).

Step 3. Depositing funds.

- Click on **View Account**.

What advantages does MetaMask have?

- **Popular** - It is commonly used, so users only need one plugin to access a wide range of dapps.
- **Simple** - Instead of managing private keys, users just need to remember a list of words, and transactions are signed on their behalf.
- **Saves space** - Users don't have to download the Ethereum blockchain, as MetaMask sends requests to nodes outside of the

user's computer.

- **Integrated** - Dapps are designed to work with MetaMask, so it becomes much easier to sendEther in and out.

Conclusion- In this way we have explored Concept Blockchain and metamat wallet for transaction of digital currency

Assignment no:2

Title of the Assignment: Create your own wallet using Metamask for crypto transactions

Objective of the Assignment: Students should be able to learn about cryptocurrencies and learn how transaction done by using different digital currency

Prerequisite:

1. Basic knowledge of cryptocurrency
 2. Basic knowledge of distributed computing concept
 3. Working of blockchain
-

Contents for Theory:

1. *Cryptocurrency*
2. **Transaction Wallets**
3. *Ether transaction*

Introduction to Cryptocurrency

- Cryptocurrency is a digital payment system that doesn't rely on banks to verify transactions. It's a peer-to-peer system that can enable anyone anywhere to send and receive payments. Instead of being physical money carried around and exchanged in the real world, cryptocurrency payments exist purely as digital entries to an online database describing specific transactions. When you transfer cryptocurrency funds, the transactions are recorded in a public ledger. Cryptocurrency is stored in digital wallets.
- Cryptocurrency received its name because it uses encryption to verify transactions. This means advanced coding is involved in storing and transmitting cryptocurrency data between wallets and to public ledgers. The aim of encryption is to provide security and safety.
- The first cryptocurrency was Bitcoin, which was founded in 2009 and remains the best known today. Much of the interest in

cryptocurrencies is to trade for profit, with speculators at times driving prices skyward.

How does cryptocurrency work?

- Cryptocurrencies run on a distributed public ledger called blockchain, a record of all transactions updated and held by currency holders.
- Units of cryptocurrency are created through a process called mining, which involves using computer power to solve complicated mathematical problems that generate coins. Users can also buy the currencies from brokers, then store and spend them using cryptographic wallets.
- If you own cryptocurrency, you don't own anything tangible. What you own is a key that allows you to move a record or a unit of measure from one person to another without a trusted third party.
- Although Bitcoin has been around since 2009, cryptocurrencies and applications of blockchain technology are still emerging in financial terms, and more uses are expected in the future. Transactions including bonds, stocks, and other financial assets could eventually be traded using the technology.

Cryptocurrency examples

There are thousands of cryptocurrencies. Some of the best known include:

- *Bitcoin:*

Founded in 2009, Bitcoin was the first cryptocurrency and is still the most commonly traded. The currency was developed by Satoshi Nakamoto – widely believed to be a pseudonym for an individual or group of people whose precise identity remains unknown.

- *Ethereum:*

Developed in 2015, Ethereum is a blockchain platform with its own cryptocurrency, called Ether (ETH) or Ethereum. It is the most popular cryptocurrency after Bitcoin.

- *Litecoin:*

This currency is most similar to bitcoin but has moved more quickly to develop new innovations, including faster payments and processes to allow more transactions.

- *Ripple:*
Ripple is a distributed ledger system that was founded in 2012. Ripple can be used to track different kinds of transactions, not just cryptocurrency. The company behind it has worked with various banks and financial institutions.
- Non-Bitcoin cryptocurrencies are collectively known as “altcoins” to distinguish them from the original.

How to store cryptocurrency

- Once you have purchased cryptocurrency, you need to store it safely to protect it from hacks or theft. Usually, cryptocurrency is stored in crypto wallets, which are physical devices or online software used to store the private keys to your cryptocurrencies securely. Some exchanges provide wallet services, making it easy for you to store directly through the platform. However, not all exchanges or brokers automatically provide wallet services for you.
- There are different wallet providers to choose from. The terms “hot wallet” and “cold wallet” are used:
- **Hot wallet storage:** “hot wallets” refer to crypto storage that uses online software to protect the private keys to your assets.
- **Cold wallet storage:** Unlike hot wallets, cold wallets (also known as hardware wallets) rely on offline electronic devices to securely store your private keys.

Conclusion- In this way we have explored Concept Cryptocurrency and learn how transactions are done using digital currency

Assignment no:3

Title of the Assignment: Write a smart contract on a test network, for Bank account of a customer for following operations:

- Deposit money
- Withdraw Money
- Show balance

Objective of the Assignment: Students should be able to learn new technology such as metamask. Its application and implementations

Prerequisite:

1. Basic knowledge of cryptocurrency
 2. Basic knowledge of distributed computing concept
 3. Working of blockchain.
-
-

Contents for Theory:

The contract will allow deposits from any account, and can be trusted to allow withdrawals only by accounts that have sufficient funds to cover the requested withdrawal.

This post assumes that you are comfortable with the ether-handling concepts introduced in our post, [Writing a Contract That Handles Ether](#).

That post demonstrated how to restrict ether withdrawals to an “owner’s” account. It did this by persistently storing the owner account’s address, and then comparing it to the msg.sender value for any withdrawal attempt. Here’s a slightly simplified version of that smart contract, which allows anybody to deposit money, but only allows the owner to make withdrawals:

pragma

solidity

^0.4.19;

contract

TipJar {

 address owner; // current owner of the contract

 function

 n

 TipJa

 r()

```

public
{
    owner
    r =
    msg.sender
    end
    r;
}

function withdraw()
public {
    require(owner ==
msg.sender);
    msg.sender.transfer(address(this).balance);
}

function deposit(uint256 amount) public
payable { require(msg.value ==
amount);
}

function getBalance() public view
returns (uint256) { return
address(this).balance;
}
}

```

I am going to generalize this contract to keep track of ether deposits based on the account address of the depositor, and then only allow that same account to make withdrawals of that ether. To do this, we need a way to keep track of account balances for each depositing account—a mapping from accounts to balances. Fortunately, Solidity provides a ready-made mapping data type that can map account addresses to integers,

The code above demonstrates the following:

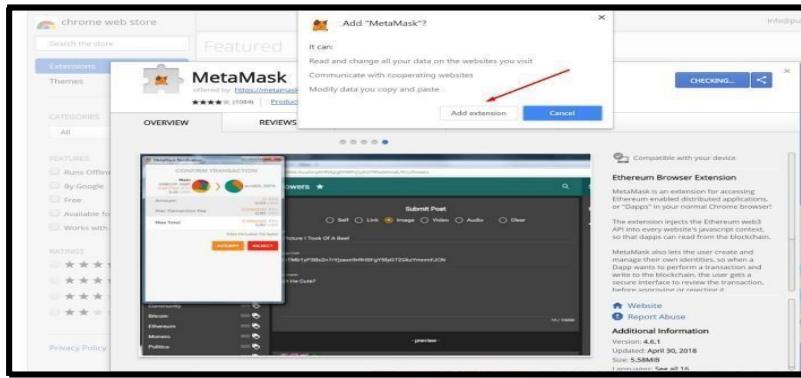
- The `require(amount <= balances[msg.sender])` checks to make sure the sender has sufficient funds to cover the requested withdrawal. If not, then the transaction aborts without making any state changes or ether transfers.
- The `balanceOf` mapping must be updated to reflect the lowered residual amount after the withdrawal.
- The funds must be sent to the sender requesting the withdrawal.

In the `withdraw()` function above, it is very important to adjust `balanceOf[msg.sender]` **before** transferring ether to avoid an exploitable vulnerability. The reason is specific to smart contracts and the fact that a transfer to a smart contract executes code in that smart contract. (The essentials of Ethereum transactions are discussed in [How Ethereum Transactions Work](#).)

Now, suppose that the code in `withdraw()` did not adjust `balanceOf[msg.sender]` before making the transfer *and* suppose that `msg.sender` was a malicious smart contract. Upon receiving the transfer—handled by `msg.sender`'s fallback function—that malicious contract could initiate

another withdrawal from the banking contract. When the banking contract handles this second withdrawal request, it would have already transferred ether for the original withdrawal, but it would not have an updated balance, so it would allow this second withdrawal!

To avoid this sort of reentrancy bug, follow the “Checks-Effects-Interactions pattern” as described in the Solidity documentation. The withdraw() function above is an example of implementing this pattern



Assignment no:4

Title of the Assignment: Write a survey report on types of Blockchains and its real time use cases.

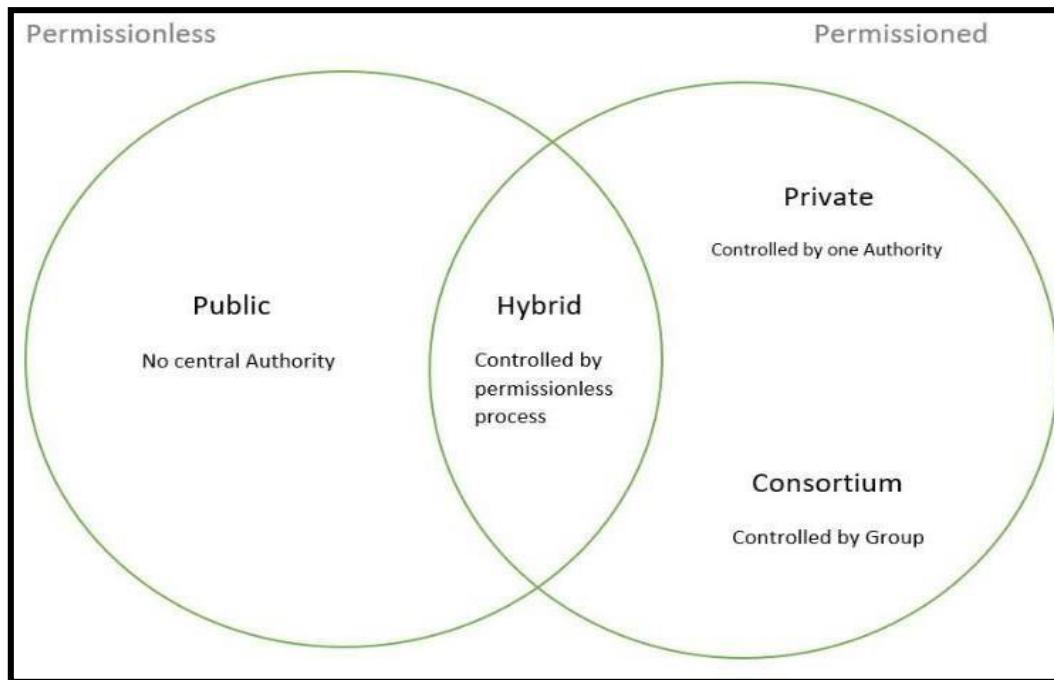
Objective of the Assignment: Students should be able to learn new technology such as metamask. Its application and implementations

Prerequisite:

1. Basic knowledge of cryptocurrency
 2. Basic knowledge of distributed computing concept
 3. Working of blockchain
-
-

Contents for Theory:

There are 4 types of blockchain: Public Blockchain.
Private Blockchain. Hybrid Blockchain. Consortium Blockchain



1. Public Blockchain

These blockchains are completely open to following the idea of decentralization. They don't have any restrictions, anyone having a computer and internet can participate in the network.

As the name is public this blockchain is open to the public, which means it is not owned by anyone. Anyone having internet and a computer with good hardware can participate in this public blockchain. All the computer in the network hold the copy of other nodes or block present in the network. In this public blockchain, we can also perform verification of transactions or records. Advantages:

Trustable: There are algorithms to detect no fraud. Participants need not worry about the other nodes in the network.

Secure: This blockchain is large in size as it is open to the public. In a large size, there is greater distribution of records.

Anonymous Nature: It is a secure platform to make your transaction properly at the same time, you are not required to reveal your name and identity in order to participate.

Decentralized: There is no single platform that maintains the network, instead every user has a copy of the ledger.

Disadvantages:

Processing: The rate of the transaction process is very slow, due to its large size. Verification of each node is a very time-consuming process.

Energy Consumption: Proof of work is high energy-consuming. It requires good computer hardware to participate in the network.

Acceptance: No central authority is there so governments are facing the issue to implement the technology faster.

Use Cases: Public Blockchain is secured with proof of work or proof of stake they can be used to displace traditional financial systems. The more advanced side of this blockchain is the smart contract that enabled this blockchain to support decentralization. Examples of public blockchain are Bitcoin, Ethereum.

2. Private Blockchain

These blockchains are not as decentralized as the public blockchain only selected nodes can participate in the process, making it more secure than the others.

These are not as open as a public blockchain. They are open to some authorized users only.

These blockchains are operated in a closed network.

In this few people are allowed to participate in a network within a company/organization. Advantages:

Speed: The rate of the transaction is high, due to its small size. Verification of each node is less time-consuming.

Scalability: We can modify the scalability. The size of the network can be decided manually. **Privacy:** It has increased the level of privacy for confidentiality reasons as the businesses required.

Balanced: It is more balanced as only some user has the access to the transaction which improves the performance of the network.

Disadvantages:

Security- The number of nodes in this type is limited so chances of manipulation are there. These blockchains are more vulnerable.

Centralized- Trust building is one of the main disadvantages due to its central nature. Organizations can use this for malpractices.

Count- Since there are few nodes if nodes go offline the entire system of blockchain can be endangered. **Use Cases:** With proper security and maintenance, this blockchain is a great asset to secure information without exposing it to the public eye. Therefore companies use them for internal auditing, voting, and asset management. An example of private blockchains is Hyperledger, Corda.

3. Hybrid Blockchain

It is the mixed content of the private and public blockchain, where some part is controlled by some organization and other makes are made visible as a public blockchain.

It is a combination of both public and private blockchain. **Permission-based** and **permissionless** systems are used. User access information via smart contracts

Even a primary entity owns a hybrid blockchain it cannot alter the transaction

Advantages:

Ecosystem: Most advantageous thing about this blockchain is its hybrid nature. It cannot be hacked as 51% of users don't have access to the network

Cost: Transactions are cheap as only a few nodes verify the transaction. All the nodes don't carry the verification hence less computational cost.

Architecture: It is highly customizable and still maintains integrity, security, and transparency. **Operations:** It can choose the participants in the blockchain and decide which transaction can be made public.

Disadvantages: **Efficiency:** Not everyone is in the position to implement a hybrid Blockchain. The organization also faces

Speed: A limited number of users make verification fast. The high speed makes this more usable for organizations.

Authority: Multiple organizations can take part and make it decentralized at every level. Decentralized authority, makes it more secure.

Privacy: The information of the checked blocks is unknown to the public view. but any member belonging to the blockchain can access it.

Flexible: There is much divergence in the flexibility of the blockchain. Since it is not a very large decision can be taken faster.

Disadvantages:

Approval: All the members approve the protocol making it less flexible. Since one or more organizations are involved there can be differences in the vision of interest.

Transparency: It can be hacked if the organization becomes corrupt. Organizations may hide information from the users.

Vulnerability: If few nodes are getting compromised there is a greater chance of vulnerability in this blockchain

Use Cases: It has high potential in businesses, banks, and other payment processors.

Food tracking of the organizations frequently collaborates with their sectors making it a federated solution ideal for their use. Examples of consortium Blockchain are Tendermint and Multichain.

Conclusion-In this way we have explored types of blockchain and its applications in real time

Assignment no:5

Title of the Assignment: Write a program to create a Business Network using Hyperledger.

Objective of the Assignment: Students should be able to learn hyperledger .Its application and implementations

Prerequisite:

1. Basic knowledge of cryptocurrency
 2. Basic knowledge of distributed computing concept
 3. Working of blockchain
-
-

Contents for Theory:

Hyperledger Composer is an extensive, open development toolset and framework to make developing blockchain applications easier. The primary goal is to accelerate time to value, and make it easier to integrate your blockchain applications with the existing business systems.

- You can use Composer to rapidly develop use cases and deploy a blockchain solution in days.
- Composer allows you to model your business network and integrate existing systems and data with your blockchain applications.
- Hyperledger Composer supports the existing [Hyperledger Fabric blockchain](#) infrastructure and runtime.
- Hyperledger Composer generates business network archive (bna) file which you can deploy on existing Hyperledger Fabric network

You can use Hyperledger Composer to model business network, containing your existing assets and the transactions related to them

**Key Concepts of
Hyperledger
Composer**

1. Blockchain State Storage: It stores all transaction that happens in your hyperledger composer application. It stores transaction in Hyperledger fabric network.
2. Participants: Participants are members of a business network. They may own assets and submit transactions. Participant must have an identifier and can have any other properties.
3. Identities and ID cards: Participants can be associated with an identity. ID cards are a combination of an identity, a connection profile, and metadata. ID cards simplify the process of connecting to a business network.
4. Transactions: Transactions are the mechanism by which participants interact with assets. Transaction processing logic you can define in JavaScript and you can also emit event for transaction.
5. Queries: Queries are used to return data about the blockchain world-state. Queries are defined within a business network, and can include variable parameters for simple customisation. By using queries, data can be easily extracted from your blockchain network. Queries are sent by using the Hyperledger Composer API.
6. Events: Events are defined in the model file. Once events have been defined, they can be emitted by transaction processor functions to indicate to external systems that something of importance has happened to the ledger.
7. Access Control: Hyperledger is enterprise blockchain and access control is core feature of any business blockchain. Using Access Control rules you can define who can do what in Business networks. The access control language is rich enough to capture sophisticated conditions.
8. Historian registry: The historian is a specialised registry which records successful

transactions, including the participants and identities that submitted them. The historian stores transactions as HistorianRecordassets, which are defined in the Hyperledger Composer system namespace.

Conclusion: In this way we have learnt about hyperledger and its use case in business world.