I/L/TX

Reg. No:

Final Assessment Test – November 2024

VIT
Vellore Institute of Technology

Course: **BCSE206L - Foundations of Data Science**
Class NBR(s): **1535/1540/1544**
Time: **Three Hours**

Slot: **F1+TF1**
Max. Marks: **100**

- **KEEPING MOBILE PHONE/ANY ELECTRONIC GADGETS, EVEN IN 'OFF' POSITION IS TREATED AS EXAM MALPRACTICE**
- **DON'T WRITE ANYTHING ON THE QUESTION PAPER**

Answer ALL Questions
(10 X 10 = 100 Marks)

1. You're part of a team at VITcodeGuardian, a company managing a large, complex code repository that's accumulated technical debt, including issues like outdated dependencies, duplicate code, and inconsistent practices. Your task is to use data science to analyze the codebase, reduce technical debt, and improve code quality. How would you approach this, from extracting and processing the code data to identifying trends in complexity, duplication, and dependencies? Explain the tools and algorithms you would use to recommend refactoring areas, and propose a machine learning model to predict future bugs or performance issues based on historical data. How would your approach complement traditional code reviews, and how could this evolve into an automated system for ongoing code quality monitoring?

2. Consider the following relations:

Plant(**plant_id**, plant_name, height_cm, growth_stage)

Farm(**farm_id**, farm_name, region, size_acres)

Farm_Plant(**farm_id**, **plant_id**, planted_date)

a) Write a query to find pairs of plants where the second plant is taller than the first plant but they are in the same growth stage.

b) Write a query to find all farms where the length of the farm name is greater than 10 characters, and display the farm name along with the region, with both values concatenated into one column.

c) Write a query to find the names of plants and the farms where they were planted **before the year 2020**.

d) Write a query to rank the plants based on their height in descending order, assigning the same rank for plants with the same height.

e) Write a query to find the farms where the average height of the plants planted is greater than 150 cm.

3. a) Given the following SQL table schema for an e-commerce application: [6]

- **Product(product_id, product_name, category, price)**
- **Review(review_id, product_id, user_id, rating, comment)**

Convert the SQL schema into a suitable NoSQL data model, justifying your choice of NoSQL database type. Discuss how you would structure the data to accommodate product details and user reviews while ensuring efficient querying and retrieval.

After defining your NoSQL schema, populate the database with sample data for at least two products and their associated reviews.

Based on this NoSQL model, write a query to retrieve the product details for Product ID 1, along with only those reviews where the rating is greater than 4.

b) Given the following four scenarios, identify the most appropriate NoSQL database type for each case and justify your choice: [4]

i) A social media platform needs to store user profiles, including varying data such as usernames, profile pictures, and posts. Each user can have different attributes.

ii) A real-time analytics application requires storing and quickly retrieving large volumes of user activity data (clicks, views) as key-value pairs for quick lookup.

iii) An online retail platform wants to manage a vast inventory of products, including product descriptions, prices, and customer reviews, where the structure of product data can vary significantly.

iv) A transportation app needs to analyze relationships between cities, routes, and vehicles, focusing on how they are interconnected.

4. You are a senior data scientist at a company facing frequent deadlock issues in its distributed cloud systems due to high resource contention. The current Banker's Algorithm is no longer sufficient for managing complex and dynamic resource allocation patterns.

Your task is to develop an **AI-driven solution using Generative AI** to predict and resolve deadlocks in real-time, improving system performance.

Using the Data Analytics Lifecycle, outline how you would approach this problem. Consider which data to gather, and plan a model design leveraging techniques like reinforcement learning or GANs to detect and resolve deadlocks. Finally, explain how you would validate and present the effectiveness of the model to the technical team while ensuring seamless integration into the existing system.

5. A store tracks daily transactions in a dataset with the following columns:
Transaction_ID (unique identifier for each transaction)
Date (date of the transaction)
Region (e.g., North, South, East, West)
Product_Type (e.g., Electronics, Clothing, Home Goods)
Units_Sold
Revenue

a) Sketch a mock line chart to show Revenue over time for each Region. Ensure each region has a separate line with a distinct colour. Label the peak revenue point for any one region and describe briefly how you would accomplish this in Tableau. [4]

b) Describe how you would apply a filter in Tableau to view data only for the Product_Type "Electronics." Additionally, explain how you could drill down from monthly to daily data on the line chart to analyze trends in more detail. [3]

c) Assume that, after creating a dashboard with this chart, you observe that the South region has a consistent upward trend in Revenue, while the West region shows a decline over the last three months. Suggest two possible actions the store might take to address these trends, specifically one for each region. [3]

6. Design a Hidden Markov Model (HMM) tagger for the following sentences using VITERBI approximation.

The Doctor is in

Transition Probability:

<s> -> Start Probability

	Noun	Verb	Det	Prep	Adv	</s>
<s>	0.3	0.1	0.3	0.2	0.1	0
Noun	0.2	0.4	0.01	0.3	0.04	0.05
Verb	0.3	0.05	0.3	0.2	0.1	0.05
Det	0.9	0.01	0.01	0.01	0.07	0
Prep	0.4	0.05	0.4	0.1	0.05	0
Adv	0.1	0.5	0.1	0.1	0.1	0.1

Emission Probability:

Emission Probability $P(w_i|t_i)$:

	a	cat	doctor	in	is	the	very
Noun	0	0.5	0.4	0	0.1	0	0
Verb	0	0	0.1	0.9	0	0	0
Det	0.3	0	0	0	0	0.7	0
Prep	0	0	0	1.0	0	0	0
Adv	0	0	0.1	0	0	0	0.9

7. a) You are given a dataset as a python dictionary representing sales data for a company, structured as follows: [5]

```
sales_data = {
  'product': ['A', 'B', 'C', 'A', 'B', 'C', 'A', 'B', 'C'],
  'region': ['North', 'South', 'East', 'North', 'South', 'East', 'North', 'South', 'East'],
  'sales': [100, 150, 200, 120, 160, 210, 130, 170, 220],
  'date': ['2023-01-01', '2023-01-01', '2023-01-01',
      '2023-01-02', '2023-01-02', '2023-01-02',
      '2023-01-03', '2023-01-03', '2023-01-03'] }
```

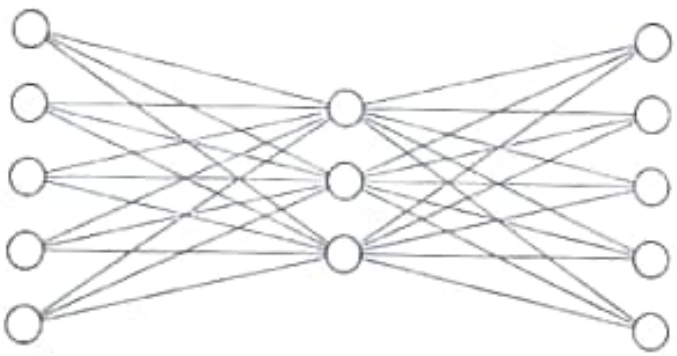
Using the Pandas library present in python, provide code to perform the following tasks:

i. Load the data into a **pandas DataFrame** and display the first 5 rows.
ii. Calculate and display the total sales for each product across all regions.
iii. Pivot the data to show total sales for each product across different regions and visualize the results using a bar plot.
iv. Calculate a 3-day **rolling average** for the sales of each product and display the results.
v. Apply **groupby** to analyze the sales trends for each region, showing the total sales per day.

b) Give applications for the following python modules: [5]

i) Sympy
ii) NLTK
iii) Seaborn
iv) Torch
v) pySpark

8. a) You are a machine learning engineer developing a neural network for object classification in self-driving cars. [7]



Input Layer $\in \mathbb{R}^4$ Hidden Layer $\in \mathbb{R}^4$ Output Layer $\in \mathbb{R}^4$

The network architecture includes an input layer with 5 neurons (object features), a hidden layer with 3 neurons using the ReLU activation function, and an output layer with 5 neurons using the Softmax activation function to output probabilities for five object categories.

Given the input feature vector $x = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 1 \\ 0 \end{bmatrix}$, and the weight matrices W1 and W2 below, write a GNU Octave script to simulate the forward pass of the network. Display the outputs at each layer, apply the ReLU activation after the hidden layer, and apply the Softmax function at the output layer.

$$W1 = \begin{bmatrix} 0.3 & -0.2 & 0.5 & 0.1 & 0.4 \\ -0.1 & 0.6 & 0.2 & -0.3 & 0.2 \\ 0.1 & 0.1 & -0.2 & 0.5 & -0.3 \end{bmatrix}$$

$$W2 = \begin{bmatrix} 0.1 & 0.5 & -0.7 \\ -0.4 & 0.2 & 0.1 \\ 0.2 & 0.4 & 0.6 \\ 0.3 & -0.2 & -0.8 \\ 0.7 & -0.1 & -0.1 \end{bmatrix}$$

b) What are the differences between MATLAB and GNU Octave? **[3]**

9.a) You are given the following Python function segment_string, which attempts to segment a concatenated string s into valid words from a dictionary D. However, this implementation is inefficient due to redundant computations and lacks optimization. Optimize this code to improve its efficiency. Provide the optimized version of the segment_string function using dynamic programming techniques.

```
def segment_string(s, D):
    def helper(sub_s):
        if not sub_s:
            return []
        for i in range(1, len(sub_s) + 1):
            word = sub_s[:i]
            if word in D:
                rest = helper(sub_s[i:])
                if rest is not None:
                    return [word] + rest
        return None
    return helper(s)
```

```
D = {
    'i': -0.5,
    'like': -0.2,
    'sam': -1.0,
    'sung': -0.8,
    'samsung': -0.3,
    'mobile': -0.4,
    'ice': -0.7,
    'cream': -0.6,
    'icecream': -0.4,
    'man': -0.9,
    'go': -0.6,
    'mango': -0.5
}
s = "ilikesamsungmobile"
segmented = segment_string(s, D)
print("Segmented String:", segmented)
```

OR

9.b) Implement the **Minimum Edit Distance** algorithm to compute the minimum number of edits required to transform one word into another using operations: insertion, deletion, and substitution (each with a cost of 1). For example, the minimum edit distance between the words **"flaw"** and **"lawn"** is **2(Remove f and insert n)**.

i) Show the **dynamic programming table** filling process step by step for transforming "sunday" to "saturday", clearly indicating how you compute the cost at each cell. **[5]**

ii) Write a **Python function** using dynamic programming to calculate the minimum edit distance between any two given words. **[5]**

10.a) You are provided with the following database schema for a retail store.

Employees(**employee_id**, first_name, last_name, department, salary, hire_date)

Sales(**sale_id**, employee_id, sale_amount, sale_data)

Develop SQL query for the following:

i. For each employee, calculate their total sales amount and rank them within their department based on this total sales amount. Provide employee_id, first_name, last_name, department, total_sales, and sales_rank.

ii. Display each employee's salary along with their previous hired employee's salary in the same department. Include employee_id, first_name, last_name, department, salary, and previous_employee_salary.

iii. Divide all employees into three groups based on their salaries across the entire company. List employee_id, first_name, last_name, salary, and salary_group.

iv. For each sale, display the sale amount and the average sale amount for that day. Include sale_id, sale_date, sale_amount, and daily_average_sale.

v. For each employee, show their salary and the difference between their salary and the average salary of all employees hired before them. Include employee_id, first_name, last_name, salary, and salary_difference.

OR

10.b) i) The python code below, implements the gradient decent algorithm for logistic regression. The parameters of the model are w and b. The learning rate of the model is 0.1. Start with w = 0 and b =0, and report the value of w and b after three iterations. **[5]**

```
data = [(1, 0), (2, 0), (3, 1), (4, 1)]
w = 0.0
b = 0.0
alpha = 0.1
iterations = 3
def sigmoid(z):
  return 1 / (1 + math.exp(-z))
for i in range(iterations):
dw = 0.0
db = 0.0
n = len(data)
for x_i, y_i in data:
  z = w * x_i + b
  y_pred = sigmoid(z)
  error = y_pred - y_i
  dw += error * x_i / n
  db += error / n
w -= alpha * dw
b -= alpha * db
print(f"Iteration {i+1}: w = {w}, b = {b}")
```

ii) Given the following points in 2D space, <(2,10) (2,5) (8,4) (5,8) (7,5) (6,4) (1,2) (4,9). Suppose we initially assign (2,10), (5,8) and (1,2) as centroids, apply k-means clustering to cluster the given data points where k = 3. **[5]**

⇔⇔⇔ I/L/TX ⇔⇔⇔