**Question 1:**

a) Download the Employees\_department\_dataset.csv dataset provided in CA

section of Moodle to a location on your PC (Link:

https://moodle2024.ncirl.ie/mod/resource/view.php?id=61673)

b) Create a Jupyter notebook from the Anaconda program at the same location of

your dataset.

c) Write python code/RapidMiner studio to import the dataset.

**import pandas as pd**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

**#loading Employee CSV**

**data = pd.read\_csv('C:/Users/HP/Downloads/Data Analytics/CA1/Employee (1).csv')**

**Question 2:**

Perform the following analysis on the dataset and provide the step-by-step process with

explanation (include screen shoots if applicable):

a) Display summary

**# display summary of the dataset**

**data.describe()**

A screenshot of a computer

AI-generated content may be incorrect.

2.b) Check for column names

**#Check for the column names**

**data.columns**

A screenshot of a computer

AI-generated content may be incorrect.

2.c) Display the top 10 rows

**#Display the top 10 rows by changing parameter**

**data.head(10)**

A screenshot of a computer

AI-generated content may be incorrect.

2.d) Display basic information about the dataset

**#Display basic information**

**data.info**

A screenshot of a computer

AI-generated content may be incorrect.

**2.e)**

Using any appropriate visualization (Plot) of your choice, analyze the Salary, Age,

and Experience Years. Provide some insights into your visualization.

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**# Visualization of Salary Distribution**

**plt.figure(figsize=(8, 5))**

**sns.histplot(data['Salary'], bins=20, kde=True, color='blue')**

**plt.xlabel('Salary')**

**plt.ylabel('Count')**

**plt.title('Distribution of Salary')**

**plt.show()**

**Insights:**

🡪The employee count is more in 5000 range and keep on till 15000 salaries but later people count lowered over 20000

🡪The presence of a long tail suggests some employees have significantly higher salaries.

A screenshot of a computer

AI-generated content may be incorrect.

**# Visualization of Age Distribution**

**plt.figure(figsize=(8, 5))**

**sns.histplot(data['Age'], bins=20, kde=True, color='green')**

**plt.xlabel('Age')**

**plt.ylabel('Count')**

**plt.title('Distribution of Age')**

**plt.show()**

**Insights:**

🡪The age distribution shows the most common age group in the company, which is over 27.5 age.

🡪It might indicate more younger or older employees in the dataset.

A screenshot of a computer

AI-generated content may be incorrect.

**# Visualization of Experience\_Years Distribution**

**plt.figure(figsize=(8, 5))**

**sns.histplot(data['ExperienceInCurrentDomain'], bins=20, kde=True, color='red')**

**plt.xlabel('Experience Years')**

**plt.ylabel('Count')**

**plt.title('Distribution of Experience Years')**

**plt.show()**

**Insights:**

The distribution of experience over 2 to 5 shows more count of employees

In left, we have most employees are experienced.

In right, we have employees with less experience.

A screenshot of a computer

AI-generated content may be incorrect.

**2f) What are the importance of visualisation in data analytics**  
  
**Importance of Visualization in Data Analytics**

Data visualization does a major role in analysing large datasets by transforming complex information into intuitive visual representations.

**Identifies Trends and Patterns** – Helps to identify trends and patterns in datasets in a visual way.

**Enhances Decision-Making** – we can do the strategies and data driven decisions with visualization

**Detects Outliers and Anomalies** – when the data have anomalies and blank values its easy to find and replace with visualization techniques

**Improves Communication** – As the business clients and stake holders are not familiar with technical data, it will be very easy to demonstrate with visuals

**Saves Time** – Instead of spending a lot of time around spread sheets like excels, we can easily use visualization for more data processing and will have more features of data representation.

**Question 3:   
a. Explain the following terms as applied in data analytics  
 I. Structured data type II. Unstructured data type III. Semi-structured data type IV. Relational Database**

**Answer:**

**Structured Data Type:**

Structured Data type is a well-organized and mostly indexed data for the easy data retrieval, usually the data is stored in rows and columns within a table like we see table format in excel, structured data is made data search easy, the queries also easy to frame and used widely in data analysis tools with relational databases. SQL databases are well organized with rows and columns where each filed is entity and each row is an entry to handle different domains data like customer data, transactions, addresses etc…

**Ex:** the employee data set having employee id, address, experience and salary etc...,

**II. Unstructured Data Type:**

Unlike structured data, unstructured data don’t have fixed structure of rows and columns, generallyunstructured data is type of data which don’t have a predefined structure and format, complex to analyse in general tools. Unstructured data type included with audio files, images, videos and text etc. To extract the data we use in Mango db, Dynamo db, Hadoop.

**Ex**: Emails, social media posts, customer reviews.

**III. Semi-structured Data Type:**

Semi Structured data has slight organization but not as much as relational databases, this type of data is something between structured and unstructured data. It will have tags or markers for entities and for certain data attributes, but the structure is more flexible. It is transparent than unstructured data, but some data processing will be required.

**Ex:** XML, JSON, or CSV files

**IV. Relational Database:**

Relational databases use SQL- Structured Query Language to modify and retrieve the data, where data stored in table format and have rows and columns. Tables in the data base holds different kind data and all are interlinked through primary and foreign key to maintain relation between the tables. This type of database is relied for transactional systems and applications where data need with consistency and integrity .

**Ex**: A MySQL, PostgreSQL, or Microsoft SQL Server database.

**3b. What are key step in data preprocessing?**

Data preprocessing ensures the data is clean, consistent, and ready for modeling. The main steps in data preprocessing are:

**Data Cleaning**:

Whenever the raw data is loaded directly into a database it will have data discrepancies, like duplicate records, missing values and correcting errors such as typos or incorrect entries. As we need clean data, it is essential for accurate analysis and model performance, so data cleaning involves identifying and fixing inconsistencies, errors and inaccuracies in the dataset. With Python, we use these functions for data cleaning dropna() to remove rows with missing data or  Duplicates can be removed using drop\_duplicates() or fillna() to fill the missing value data with mean, median, or mode.

**Data Transformation:**

Once the data is cleaned, we need a format suitable for analysis, so we use data transformation for encoding categorical variables (one-hot encoding or label encoding), transforming skewed data (log transformations), and converting data types if necessary. This converts data into a suitable format for analysis, for data, we use techniques like Label Encoding and One-Hot Encoding.

**Feature Scaling:**

Feature Scaling often deals with numerical data which has similar scales, which can create different biases when we are dealing with certain algorithms. Scaling standardizes or normalizes these features to a common range (like 0 to 1 or mean of 0 with a standard deviation of 1). Techniques like Min-Max scaling or Z-score normalization and Standardization (StandardScaler)

**Feature Engineering:**

Feature Engineering includes creating new features or helps to modify existing features to catch best patterns. For example, deriving age from a birthdate or creating interaction terms between features can improve model performance. extracting date components (like year, month is eeffective feature engineering can significantly improve model performance.

**Data Splitting:**

Data Split the data into training and testing sets, to make sure that the test dataset remains unseen during model training to retrieve the data effectively. The training set is used to train the model, in other hand the test set evaluates its generalization ability. Generally splits include 80/20 or 70/30 ratios, we use train\_test\_split() function from sklearn.model\_selection for this one.

**3.c)**

Perform the following operations on the dataset and provide the step by step

process with screen shoots if applicable:

**I. Encode all the Gender columns with Label Encoding**

**# Import LabelEncoder**

**from sklearn.preprocessing import LabelEncoder**

**# Initialize Label Encoder**

**label\_encoder = LabelEncoder()**

**# Encode Gender column**

**data['Gender'] = label\_encoder.fit\_transform(data['Gender'])**

**data.head()**

**A screenshot of a computer

AI-generated content may be incorrect.**

**3.c)**

**II. Add a new column called “Salary\_after\_tax” by reducing the Salary by 30%**

**tax.**

**Hint: (Multiply Salary Column by 0.7)**

**# Apply tax reduction (70% of Salary)**

**data['Salary\_after\_tax'] = data['Salary'] \* 0.7**

**# Display first few rows after adding the new column**

**data.head()**

**A screenshot of a computer

AI-generated content may be incorrect.**

**3.c)**

IV. Filter the rows with Salary > 50000 and Experience\_Years > 10 years.

**We dont have salary more than 21000 so no data showing up**

**# Filter employees with Salary > 50000 and Experience > 10 years**

**data\_salary\_experience\_filtered = data[(data['Salary'] > 50000) & (data['ExperienceInCurrentDomain'] > 10)]**

**# Display result**

**data\_salary\_experience\_filtered.head()**

A screenshot of a computer

AI-generated content may be incorrect.