**CS3352- FOUNDATIONS OF DATA SCIENCE**

**UNIT I: INTRODUCTION**

**13-MARK QUESTIONS AND ANSWERS**

1. **Explain the benefits and uses of Data Science in various industries.**  
   Data Science helps businesses improve operations, forecast trends, personalize customer experiences, and automate decision-making. In healthcare, it is used for disease prediction, patient monitoring, and drug discovery. In finance, it supports fraud detection, risk analysis, and algorithmic trading. In retail, it is used for customer behavior analysis, inventory management, and recommendation systems. Overall, data science drives innovation and efficiency.
2. **Describe in detail the facets of data and why understanding them is important.**  
   The five key facets of data are:
   * **Volume:** Refers to the massive amount of data generated every second.
   * **Variety:** Refers to the different types of data (structured, unstructured, semi-structured).
   * **Velocity:** Refers to the speed at which data is generated and processed.
   * **Veracity:** Refers to the accuracy and trustworthiness of data.
   * **Value:** Refers to the usefulness of the data for decision-making. Understanding these aspects helps in designing efficient data storage, retrieval, and processing systems.
3. **Explain the Data Science Process with all its stages.**  
   The data science process includes:
   * **Defining Research Goals:** Understanding the problem and defining objectives.
   * **Retrieving Data:** Collecting relevant data from various sources.
   * **Data Preparation:** Cleaning, transforming, and structuring data.
   * **Exploratory Data Analysis (EDA):** Using statistics and visualizations to discover patterns.
   * **Building the Model:** Applying machine learning or statistical methods.
   * **Presenting Findings:** Communicating results using dashboards, reports, or presentations.
   * **Building Applications:** Creating data-driven applications or decision-making systems.
4. **Discuss the concepts of Data Mining and its role in Data Science.**  
   Data mining involves discovering hidden patterns and relationships in large datasets using algorithms like classification, clustering, association, and regression. It helps in predicting future trends and identifying anomalies. Examples include market basket analysis, customer segmentation, and credit scoring. It supports better decision-making and strategic planning in various domains.
5. **Differentiate between Data Warehousing and Data Mining with examples.**
   * **Data Warehousing:** Focuses on storing large volumes of historical data in a centralized repository for analysis and reporting (e.g., a retail company storing daily sales data).
   * **Data Mining:** Involves analyzing data to extract useful patterns (e.g., identifying customer buying habits).
   * **Comparison:** Data warehousing supports efficient storage and access, while data mining extracts insights. Both are essential parts of business intelligence and data science workflows.

**UNIT II: DESCRIBING DATA**

**13-MARK QUESTIONS AND DETAILED ANSWERS**

1. **Explain the types of data and variables used in statistical analysis.**

**Answer:** Data can be categorized as:

* + **Qualitative (Categorical) Data**: Describes categories or labels (e.g., gender, colors).
  + **Quantitative (Numerical) Data**: Represents measurable quantities (e.g., age, height).

**Variables:**

* + **Nominal Variable**: No natural order (e.g., religion, blood type).
  + **Ordinal Variable**: Has a meaningful order (e.g., customer satisfaction levels).
  + **Interval Variable**: Ordered with equal intervals but no true zero (e.g., temperature in Celsius).
  + **Ratio Variable**: Ordered, equal intervals, and true zero (e.g., weight, height).

Understanding these types is important for choosing the right analysis methods.

1. **Describe how data can be presented using tables and graphs.**

**Answer:** Data can be represented using:

* + **Tables**: Frequency tables, relative frequency tables, and cumulative frequency tables.
  + **Graphs**:
    - **Bar Graphs**: Show frequencies of categorical data.
    - **Histograms**: Show frequency distributions of quantitative data.
    - **Pie Charts**: Show proportions as slices of a pie.
    - **Line Graphs**: Show trends over time.
    - **Box Plots**: Show distribution, median, and outliers. Visualization helps in easy understanding of trends, comparisons, and distributions.

1. **Explain the different types of averages used to describe data.**

**Answer:** The three main types of averages are:

* + **Mean**: Sum of all values divided by the number of values. Sensitive to outliers.
  + **Median**: Middle value when data is arranged in order. Not affected by outliers.
  + **Mode**: Most frequent value in the dataset. Useful for categorical data.

Other types:

* + **Weighted Mean**: Mean where some values are given more importance.
  + **Geometric Mean**: Used in financial and growth data. These measures help in summarizing large datasets with a single representative value.

1. **Discuss variability and how it is measured.**

**Answer:** Variability shows how much the data values deviate from the average. Measures include:

* + **Range**: Difference between the highest and lowest value.
  + **Variance**: Average of squared differences from the mean.
  + **Standard Deviation (SD)**: Square root of variance. Indicates the spread of data.
  + **Interquartile Range (IQR)**: Difference between Q3 and Q1. Shows spread of the middle 50% of data. Variability is crucial in understanding data consistency and reliability.

1. **Explain the characteristics and importance of normal distribution and standard (z) scores.**

**Answer:**

* + **Normal Distribution**: A symmetric, bell-shaped curve with:
    - Mean = Median = Mode
    - 68% of data within ±1 SD, 95% within ±2 SD, 99.7% within ±3 SD
  + **Z-score**: Tells how many standard deviations a value is from the mean.

Formula: z=(x−μ)σz = \frac{(x - \mu)}{\sigma} Where:

* + - xx = value
    - μ\mu = mean
    - σ\sigma = standard deviation

Importance:

* + Helps in comparing scores from different distributions.
  + Useful in standardizing data for further analysis.
  + Identifies outliers and probabilities under the curve.

**UNIT III: DESCRIBING RELATIONSHIPS**

**13-MARK QUESTIONS AND ANSWERS**

1. **Explain correlation and the correlation coefficient in detail.**
   * **Correlation** is a statistical measure that describes the strength and direction of the relationship between two variables. It can be positive (both variables increase together), negative (one increases as the other decreases), or zero (no relationship).
   * The **correlation coefficient (r)** ranges from -1 to +1:
     + r = +1: perfect positive correlation
     + r = -1: perfect negative correlation
     + r = 0: no correlation
   * It is calculated as:

r=n(∑xy)−(∑x)(∑y)[n∑x2−(∑x)2][n∑y2−(∑y)2]r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}}

1. **Describe regression analysis and the least squares regression line.**
   * **Regression** is used to predict the value of a dependent variable based on one or more independent variables.
   * The **least squares regression line** minimizes the sum of the squares of the residuals (errors). The formula is:

y=a+bxy = a + bx where:

* + - b=n∑xy−(∑x)(∑y)n∑x2−(∑x)2b = \frac{n\sum xy - (\sum x)(\sum y)}{n\sum x^2 - (\sum x)^2}
    - a=∑y−b∑xna = \frac{\sum y - b\sum x}{n}
  + The line represents the best prediction for the dependent variable based on the independent variable.

1. **Explain standard error of estimate and interpretation of r².**
   * **Standard Error of Estimate (SEE)** is the standard deviation of the observed values from the regression line. It shows the accuracy of predictions made by the regression line.

SEE=∑(y−y^)2nSEE = \sqrt{\frac{\sum (y - \hat{y})^2}{n}}

* + **r² (coefficient of determination)** tells how well the data fit the regression model. It represents the proportion of the variance in the dependent variable that is predictable from the independent variable.
    - r² = 0.9 means 90% of the variation is explained by the model.

1. **Write in detail about multiple regression and give examples.**
   * **Multiple regression** uses two or more independent variables to predict a dependent variable. The formula is:

y=a+b1x1+b2x2+...+bnxny = a + b\_1x\_1 + b\_2x\_2 + ... + b\_nx\_n

* + Each bib\_i indicates the effect of xix\_i on y, holding other variables constant.
  + **Example**: Predicting house prices using size, number of rooms, and location.
  + This is widely used in economics, marketing, and social sciences.

1. **Explain regression toward the mean with example.**
   * **Regression toward the mean** is a phenomenon where extreme values tend to move closer to the average on subsequent measurements.
   * **Example**: A student scoring exceptionally high on a test may score closer to the average on the next one due to chance.
   * This is not due to a causal effect but rather natural variability and randomness in measurement.

Here are the detailed 13-mark questions and answers for **UNIT IV: Python Libraries for Data Wrangling**, covering concepts related to **NumPy** and **Pandas**.

**UNIT IV: PYTHON LIBRARIES FOR DATA WRANGLING**

**13-MARK QUESTIONS AND ANSWERS**

**1. Explain the basics of NumPy arrays and how they are used for data manipulation.**

**Answer:**

**NumPy** (Numerical Python) is a powerful library for numerical computing in Python. It is used to work with large arrays and matrices of numeric data.

**Basics of NumPy arrays:**

* **Array**: A NumPy array is a multi-dimensional container for homogeneous data types. Arrays in NumPy are faster and more efficient than Python lists because they are implemented in C.
* **Creation**: You can create arrays using functions like np.array(), np.zeros(), np.ones(), and np.arange().

**Example:**

import numpy as np

arr = np.array([1, 2, 3, 4])

print(arr)

Output:

[1 2 3 4]

**Array Operations**:

* NumPy allows you to perform element-wise operations like addition, subtraction, multiplication, and division efficiently.
* **Example**: Array addition:

arr1 = np.array([1, 2, 3])

arr2 = np.array([4, 5, 6])

result = arr1 + arr2

print(result)

Output:

[5 7 9]

**Benefits**:

* **Performance**: NumPy arrays are faster than Python lists for operations due to their compactness and optimized C backend.
* **Broadcasting**: Allows operations on arrays of different shapes, making code more readable and concise.

**2. Explain aggregations in NumPy and how they are used in data analysis.**

**Answer:**

**Aggregations** refer to the process of summarizing or combining data from multiple elements into a single value, and NumPy provides various functions for this purpose.

**Common aggregation functions in NumPy**:

1. **np.sum()**: Sums elements of an array.
2. **np.mean()**: Calculates the mean of the elements.
3. **np.min() and np.max()**: Return the minimum and maximum values.
4. **np.std()**: Computes the standard deviation.
5. **np.median()**: Finds the median value.
6. **np.percentile()**: Computes the nth percentile of data.

**Example:**

arr = np.array([1, 2, 3, 4, 5])

mean = np.mean(arr)

std\_dev = np.std(arr)

print(f"Mean: {mean}, Standard Deviation: {std\_dev}")

Output:

Mean: 3.0, Standard Deviation: 1.4142135623730951

**Why Aggregations are Useful**:

* Aggregation functions help summarize large datasets, making it easier to draw insights from data.
* They allow you to calculate central tendencies, variances, and more, which are important in statistical analysis and machine learning preprocessing.

**3. Describe how boolean indexing and masks are used in NumPy for data selection.**

**Answer:**

**Boolean Indexing** and **Masks** are powerful techniques in NumPy that allow you to select or manipulate elements in an array based on certain conditions.

**Boolean Indexing**:

* Boolean indexing involves creating a boolean array (True/False) and using it to index the original array.
* The boolean array has the same shape as the original array, and only elements corresponding to True values in the boolean array are selected.

**Example:**

arr = np.array([10, 20, 30, 40, 50])

mask = arr > 25 # Create a boolean mask

print(arr[mask]) # Select elements where the condition is True

Output:

[30 40 50]

**Why It's Useful**:

* Boolean indexing is efficient for filtering and selecting elements from a large dataset based on conditions, making it essential in data wrangling.

**Masks**:

* A **mask** is an array of boolean values used to filter data or apply conditions in operations.

**Example**: Masking elements:

arr = np.array([1, 2, 3, 4, 5])

mask = arr % 2 == 0 # Find even numbers

print(arr[mask]) # [2 4]

**4. Explain the concept of fancy indexing and provide an example.**

**Answer:**

**Fancy Indexing** in NumPy refers to using an array or list of indices to access multiple elements of an array at once. It’s a more flexible indexing technique that allows non-contiguous selection.

**Example**:

arr = np.array([10, 20, 30, 40, 50])

indices = [0, 2, 4] # Indices to select

print(arr[indices]) # Select elements at positions 0, 2, and 4

Output:

[10 30 50]

**Why Fancy Indexing is Useful**:

* **Efficient Data Selection**: Allows easy selection of non-contiguous data, which is useful when dealing with complex datasets or performing complex data manipulations.
* **Modification**: You can modify multiple elements of an array at once using fancy indexing.

**5. Discuss hierarchical indexing in Pandas and its use in multi-dimensional data manipulation.**

**Answer:**

**Hierarchical Indexing** in **Pandas** refers to the ability to have multiple index levels in a DataFrame or Series. It is useful for dealing with multi-dimensional data in a 2D structure, like representing data from multiple sources or categories.

**Creating Hierarchical Indexing**: You can create hierarchical indexing by passing multiple columns as indices when creating a DataFrame or Series.

**Example**:

import pandas as pd

data = {

'Region': ['North', 'South', 'East', 'West'],

'Year': [2020, 2021, 2022, 2023],

'Sales': [100, 150, 200, 250]

}

df = pd.DataFrame(data)

df.set\_index(['Region', 'Year'], inplace=True)

print(df)

Output:

Sales

Region Year

North 2020 100

South 2021 150

East 2022 200

West 2023 250

**Benefits of Hierarchical Indexing**:

* **Multi-dimensional Data**: Allows you to work with multiple dimensions of data efficiently, making it easier to represent grouped or nested data.
* **Advanced Data Operations**: You can use .loc and .xs() to perform advanced querying and manipulation based on index levels.

**Example**: Querying by multiple levels of indexing:

# Select sales data for the 'East' region in 2022

print(df.loc['East', 2022])

Output:

200

These answers focus on using **NumPy** for numerical data manipulation and **Pandas** for structured data wrangling, both of which are essential skills in data analysis and preprocessing tasks in Python.

Let me know if you need further explanations or more questions on any topic!