### **DATA SCIENCE LAB**

### **VIVA QUESTIONS & ANSWERS**

### **CYCLE 1**

### 1. What is a prime number? How can you determine if a number is prime?

Answer: A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. To determine if a number is prime, you can check if it is divisible by any number from 2 to the square root of the number.

### 2. Explain how the Fibonacci series is generated.

Answer: The Fibonacci series is generated by starting with two numbers, typically 0 and 1. Each subsequent number in the series is the sum of the two preceding ones. This can be represented as F(n) = F(n-1) + F(n-2).

### 3. What are the types of triangles based on side lengths?

Answer: Triangles can be classified as:

* Equilateral: All sides are equal.
* Isosceles: Two sides are equal.
* Scalene: All sides are of different lengths.

### 4. Define coprime numbers.

Answer: Two numbers are said to be coprime if their greatest common divisor (GCD) is 1, meaning they have no common factors other than 1.

### **5.** How do you find the roots of a quadratic equation?

Answer: The roots of a quadratic equation ax2+bx+c=0 can be found using the quadratic formula: x=2a−b±b2−4ac

### 6. What is a perfect number?

Answer: A perfect number is a positive integer that is equal to the sum of its proper divisors (excluding itself). For example, 6 is a perfect number because its divisors (1, 2, 3) sum to 6.

### 7. Describe what an Armstrong number is.

Answer: An Armstrong number (or narcissistic number) for a number of n digits is an integer such that the sum of its digits raised to the power n equals the number itself. For example, 153 is an Armstrong number since 13+53+33=153.

### 8. What are the differences between lists, tuples, dictionaries, and sets in Python?

Answer:

* List: Ordered, mutable, allows duplicates.
* Tuple: Ordered, immutable, allows duplicates.
* Dictionary: Unordered, mutable, stores key-value pairs, keys must be unique.
* Set: Unordered, mutable, no duplicates allowed.

### 9. Explain how to add two lists element-wise.

Answer: You can add two lists element-wise by using a loop or a list comprehension that iterates through both lists, adding corresponding elements. They must be of the same length.

### 10. What is bubble sort?

Answer: Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. This process is repeated until no swaps are needed.

### 11. How can you count the vowels in a string using a dictionary?

Answer: You can initialize a dictionary with vowels as keys and their counts as values. As you iterate through the string, you increment the count in the dictionary for each vowel found.

### 12. What is the purpose of the sum\_of\_digits function in the provided code?

Answer: The sum\_of\_digits function calculates the sum of all the digits of a given number. It helps in reducing the number by the sum of its digits iteratively.

### 13. How do you find absent digits in a mobile number?

Answer: To find absent digits, you can create a set of all digits (0-9) and compare it to the digits present in the mobile number. The difference will give you the absent digits.

### 14. Explain the logic used to check if a number is perfect.

Answer: The logic involves iterating through all numbers less than the given number, summing those that divide the number evenly. If this sum equals the original number, it is perfect.

### 15. What does cmath library do in Python?

Answer: The cmath library provides mathematical functions for complex numbers, including functions for calculating the square root of negative numbers.

### 16. How do you ensure that input values are valid in your programs?

Answer: You can use try-except blocks to catch exceptions when converting input values, and use conditions to check if the input meets certain criteria (like length for mobile numbers).

### 17. What is the significance of the swapped flag in the bubble sort algorithm?

Answer: The swapped flag helps optimize the bubble sort algorithm. If no elements were swapped during a pass, the list is already sorted, and the algorithm can terminate early.

### 18. How can the efficiency of checking for prime numbers be improved?

Answer: Instead of checking all numbers up to n, you can check up to the square root of n. Additionally, skipping even numbers after checking for 2 can also improve efficiency.

### 19. How do you handle invalid input for a mobile number?

Answer: You can check the length of the input and ensure it contains only digits. If the input doesn't meet these criteria, you can inform the user about the invalid input.

### 20. Why is it important to round the roots of a quadratic equation to two decimal places?

Answer: Rounding the roots to two decimal places provides a cleaner output that is easier to read and interpret, especially for practical applications where exact precision may not be necessary.

### **CYCLE 2**

### 1. What is NumPy, and why is it important in Python programming?

Answer: NumPy is a library in Python for numerical computing. It provides support for arrays, matrices, and many mathematical functions, making it essential for data analysis, machine learning, and scientific computing.

### 2. Explain the difference between a list and a NumPy array.

Answer: Lists in Python are flexible and can contain mixed data types, while NumPy arrays are homogeneous and provide better performance and functionality for numerical operations.

### 3. What are the main advantages of using NumPy arrays over Python lists?

Answer: NumPy arrays are more memory efficient, faster for numerical operations, and provide a vast array of mathematical functions for performing complex calculations.

### 4. What is the shape of a NumPy array, and how is it determined?

Answer: The shape of a NumPy array is a tuple that represents the size of each dimension of the array. It is determined by the number of elements in each dimension.

### 5. What are the different ways to create a NumPy array?

Answer: NumPy arrays can be created using functions like np.array(), np.zeros(), np.ones(), np.arange(), and np.linspace(), among others.

### 6. Explain the concept of broadcasting in NumPy.

Answer: Broadcasting is a technique that allows NumPy to perform operations on arrays of different shapes by automatically expanding the smaller array to match the shape of the larger one.

### 7. What is the purpose of the reshape() function in NumPy?

Answer: The reshape() function is used to change the shape of an existing NumPy array without changing its data. It allows for easy manipulation of array dimensions.

### 8. Describe the difference between deep copy and shallow copy in NumPy.

Answer: A shallow copy creates a new array object that points to the same data as the original array, while a deep copy creates a new array object with its own separate copy of the data.

### 9. What are universal functions (ufuncs) in NumPy?

Answer: Universal functions are functions that operate element-wise on arrays, providing vectorized operations for mathematical functions like addition, subtraction, and trigonometric operations.

### 10. How can you perform matrix multiplication in NumPy?

Answer: Matrix multiplication in NumPy can be performed using the @ operator or the np.dot() function, which computes the dot product of two arrays.

### 11. What is the difference between np.sum() and np.mean()?

Answer: np.sum() calculates the total sum of array elements, while np.mean() computes the average of the array elements.

### 12. Explain the purpose of the axis parameter in NumPy functions.

Answer: The axis parameter specifies the dimension along which an operation is performed. For example, in a 2D array, axis=0 operates on columns, while axis=1 operates on rows.

### 13. What are masked arrays in NumPy?

Answer: Masked arrays are arrays that allow for the exclusion of certain elements based on a mask, enabling operations on only the valid data while ignoring masked values.

### 14. Describe how to index and slice a NumPy array.

Answer: Indexing in NumPy allows accessing individual elements using square brackets. Slicing is done using the colon : operator to specify a range of indices, allowing for extraction of subarrays.

### 15. What is the purpose of the np.unique() function?

Answer: The np.unique() function returns the unique elements of an array, removing duplicates and providing options to sort and return indices.

### 16. Explain the concept of data types in NumPy arrays.

Answer: Data types in NumPy arrays define the type of elements stored (e.g., integers, floats, strings). They ensure efficient memory usage and dictate the operations that can be performed on the array.

### 17. What is an array view in NumPy?

Answer: An array view is a new array object that looks at the same data as the original array. Changes made to a view affect the original array and vice versa.

### 18. How does NumPy handle multidimensional arrays?

Answer: NumPy provides specialized functions and methods to manipulate and perform operations on multidimensional arrays, treating each dimension as a separate axis for computations.

### 19. What is the purpose of the np.random module in NumPy?

Answer: The np.random module is used to generate random numbers and perform random sampling. It provides various functions for creating arrays of random numbers with different distributions.

### 20. What are some common applications of NumPy in data science?

Answer: Common applications of NumPy in data science include data manipulation, numerical analysis, statistical modeling, machine learning, and image processing.

**CYCLE 3**

### **1. What is Matplotlib, and what are its primary uses?**

Answer: Matplotlib is a plotting library for Python that enables the creation of static, animated, and interactive visualizations in a variety of formats. It is primarily used for data visualization in scientific computing, data analysis, and machine learning.

### 2. Describe the purpose of the plt.subplot() function in Matplotlib.

Answer: The plt.subplot() function is used to create multiple plots in a single figure, allowing the user to specify the number of rows and columns for the subplots.

### 3. How can you customize the style of a line plot in Matplotlib?

Answer: You can customize a line plot's style by changing properties such as the line color, line style (e.g., solid, dashed), marker type, marker color, and size using parameters in the plt.plot() function.

### 4. Explain the use of markers in Matplotlib plots.

Answer: Markers are symbols used to denote data points on a plot. They can vary in shape, color, and size, allowing for enhanced visualization and differentiation of data series.

### 5. What is the significance of the grid() function in plots?

Answer: The grid() function adds a grid to the plot, which can help improve readability by making it easier to align data points with the axes.

### 6. How do you display a title on a plot and align it?

Answer: You can use the plt.title() function to display a title on a plot. The alignment can be controlled using the loc parameter, which can take values like 'left', 'center', or 'right'.

### 7. What are histograms used for in data visualization?

Answer: Histograms are used to represent the distribution of a dataset by dividing the data into bins and counting the number of observations in each bin, allowing for visual assessment of data distribution.

### 8. Describe the role of the bar() function in Matplotlib.

Answer: The bar() function is used to create bar charts, which represent categorical data with rectangular bars, where the height of each bar corresponds to the frequency or value of the category.

### 9. How can you create a scatter plot in Matplotlib, and what is its purpose?

Answer: A scatter plot is created using the plt.scatter() function. It is used to show the relationship between two continuous variables, with each point representing an observation.

### 10. What is the purpose of using plt.tight\_layout()?

Answer: The plt.tight\_layout() function automatically adjusts subplot parameters to give specified padding, preventing overlap of plot elements and ensuring a cleaner layout.

### 11. Explain the difference between the displot() and histplot() functions in Seaborn.

Answer: displot() is used to create a distribution plot that can show both histograms and kernel density estimates (KDE), while histplot() specifically focuses on creating histograms.

### 12. What types of plots can you create using the pairplot() function in Seaborn?

Answer: The pairplot() function can create pairwise relationships between multiple variables, allowing for different types of plots such as scatter plots, histograms, and KDE plots based on specified parameters.

### 13. How do you display a legend in a Matplotlib plot?

Answer: You can display a legend using the plt.legend() function, which identifies different data series in a plot by their labels, helping to clarify the information presented.

### 14. Why is it important to provide appropriate labels for axes in a plot?

Answer: Providing appropriate labels for axes enhances the clarity and interpretability of a plot, helping viewers understand what data is being represented and making the visualization more informative.

### 15. What is the significance of the size parameter in scatter plots?

Answer: The size parameter in scatter plots determines the size of the markers, which can be used to represent additional information, such as the magnitude or importance of data points.

### 16. Describe the describe() function in Pandas and its utility.

Answer: The describe() function in Pandas provides a summary of descriptive statistics for DataFrame columns, such as count, mean, standard deviation, minimum, and quartiles, facilitating quick insights into the dataset.

### 17. How does the read\_csv() function work in Pandas?

Answer: The read\_csv() function reads a CSV (Comma-Separated Values) file into a Pandas DataFrame, allowing for easy data manipulation and analysis.

### 18. What are the typical visualizations you can perform with categorical data?

Answer: Typical visualizations for categorical data include bar charts, pie charts, and count plots, which display the frequency of each category.

### 19. How can you improve the visual appeal of a plot?

Answer: You can improve the visual appeal of a plot by customizing colors, styles, and markers, adding grid lines, adjusting the layout, using appropriate labels, and including annotations as needed.

### 20. Why is data visualization crucial in data analysis?

Answer: Data visualization is crucial because it helps to identify patterns, trends, and anomalies in data quickly, making complex data more understandable and facilitating better decision-making.

**CYCLE 4**

### **1. What is the KNN algorithm, and how does it work?**

Answer: The KNN (K-Nearest Neighbors) algorithm is a non-parametric classification method that classifies data points based on the majority class among the K nearest neighbors in the feature space. It uses distance metrics (like Euclidean distance) to find the closest neighbors.

### 2. How do you choose the value of K in KNN?

Answer: The value of K can be chosen based on cross-validation or experimentation. A smaller K can lead to noise sensitivity, while a larger K may smooth over the classification boundaries. Typically, odd values are preferred to avoid ties.

### 3. What is the purpose of splitting data into training and test sets?

Answer: Splitting data into training and test sets helps evaluate the model's performance on unseen data. The training set is used to train the model, while the test set assesses how well the model generalizes.

### 4. Explain the output of the classification report in KNN.

Answer: The classification report includes metrics such as precision, recall, F1-score, and support for each class. It provides a summary of the model's performance, indicating how well it predicts each class.

### 5. What are some common datasets suitable for KNN classification?

Answer: Common datasets for KNN include the Iris dataset, the Wine dataset, and the Blood Transfusion dataset, as they contain labeled data for classification tasks.

### 6. What is the Naive Bayes algorithm, and in what scenarios is it used?

Answer: Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes independence between features and is often used for text classification, spam detection, and sentiment analysis.

### 7. How does Gaussian Naive Bayes differ from Bernoulli Naive Bayes?

Answer: Gaussian Naive Bayes assumes that the features follow a normal distribution, making it suitable for continuous data, while Bernoulli Naive Bayes is used for binary/boolean features.

### 8. What metrics are commonly used to evaluate classification models?

Answer: Common metrics include accuracy, precision, recall, F1-score, and confusion matrix, which provide insights into the model's performance and classification effectiveness.

### 9. How do you interpret the confusion matrix?

Answer: A confusion matrix displays the true positives, true negatives, false positives, and false negatives. It helps to understand the model's performance in terms of correctly and incorrectly classified instances.

### 10. What is the Decision Tree algorithm, and how does it make decisions?

Answer: The Decision Tree algorithm is a non-parametric supervised learning method used for classification and regression. It makes decisions based on a series of questions about the features, resulting in a tree-like structure.

### 11. Explain how misclassification is determined in classification tasks.

Answer: Misclassification is determined by comparing predicted labels to actual labels. Instances where the predicted label differs from the true label are counted as misclassified.

### 12. What is the importance of preprocessing data in machine learning?

Answer: Preprocessing is crucial as it improves data quality, ensures compatibility with models, and enhances model performance. Techniques include normalization, encoding categorical variables, and handling missing values.

### 13. Describe the linear regression algorithm and its assumptions.

Answer: Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. Assumptions include linearity, independence, homoscedasticity, and normal distribution of errors.

### 14. What is the difference between simple and multiple linear regression?

Answer: Simple linear regression involves one independent variable and one dependent variable, while multiple linear regression involves two or more independent variables predicting a dependent variable.

### 15. How can you assess the performance of a regression model?

Answer: Performance can be assessed using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared, which evaluate prediction accuracy.

### 16. What role does a neural network play in predicting house prices?

Answer: A neural network can model complex relationships between features and the target variable (house prices), learning patterns from data to predict whether prices are above or below a median value.

### 17. Explain the significance of the activation function in a neural network.

Answer: The activation function determines the output of a neuron, introducing non-linearity into the model. Common functions include ReLU, sigmoid, and tanh, which help the network learn complex patterns.

### 18. How does the train-validation-test split work in neural networks?

Answer: The data is split into three sets: the training set for model training, the validation set for tuning hyperparameters, and the test set for evaluating the final model's performance.

### 19. What is the purpose of using Min-Max scaling in data preprocessing?

Answer: Min-Max scaling normalizes the feature values to a specific range (usually [0, 1]), which helps improve the convergence speed of gradient descent algorithms and ensures features contribute equally to the model.

### 20. What are some common libraries used for implementing machine learning algorithms in Python?

Answer: Common libraries include Scikit-learn for classical algorithms, TensorFlow and Keras for deep learning, and Pandas for data manipulation, along with Matplotlib and Seaborn for data visualization.

**CYCLE 5**

### **1. What is a web crawler, and how does it work?**

Answer: A web crawler is an automated script that browses the web to collect and index data from web pages. It works by sending requests to URLs, retrieving the HTML content, and parsing it to extract information, such as text or links.

### 2. How can you extract specific HTML elements, like <p> tags, using BeautifulSoup?

Answer: BeautifulSoup allows you to parse HTML and access elements using methods like find() or find\_all(). For example, soup.find\_all('p') retrieves all <p> tags from the parsed HTML.

### 3. Why is it important to check for empty links when extracting hyperlinks?

Answer: Checking for empty links prevents errors and ensures that only valid hyperlinks are processed. Empty links can lead to broken URLs and unintended behavior in applications.

### 4. Explain the concept of n-grams in Natural Language Processing.

Answer: N-grams are contiguous sequences of n items (words or characters) from a given text. They are used to analyze text patterns, generate language models, and improve various NLP tasks like text classification.

### 5. How would you implement n-grams without using a library?

Answer: You can implement n-grams by splitting the text into words and using a loop to combine consecutive words into tuples of size n. This allows you to generate the desired n-grams from the input text.

### 6. What are the advantages of using libraries like NLTK for NLP tasks?

Answer: Libraries like NLTK provide pre-built functions and tools for text processing tasks such as tokenization, stemming, and n-gram generation, making it easier and faster to implement complex NLP algorithms.

### 7. What is the purpose of stop word removal in text processing?

Answer: Stop word removal eliminates common words (e.g., "and," "the") that do not carry significant meaning, reducing noise and improving the focus on relevant terms in text analysis.

### 8. Describe the process of tokenization in NLP.

Answer: Tokenization is the process of breaking down text into smaller units, such as words or sentences. This helps in analyzing the structure and meaning of the text, facilitating further processing.

### 9. How does K-means clustering work?

Answer: K-means clustering partitions data into K clusters by assigning each data point to the nearest centroid and iteratively updating the centroids based on the mean of assigned points until convergence.

### 10. What factors should you consider when choosing the number of clusters (K) in K-means?

Answer: Considerations include the dataset's characteristics, the elbow method for identifying the optimal K, and silhouette scores to evaluate the quality of clustering.

### 11. How can you visualize clusters in a scatter plot?

Answer: Clusters can be visualized by plotting data points with different colors for each cluster. The cluster centroids can also be plotted as distinct markers to show their positions.

### 12. Why is it important to visualize the cluster centers in K-means?

Answer: Visualizing cluster centers helps understand the central tendencies of each cluster, providing insights into the data distribution and characteristics of different segments.

### 13. What does it mean to display cluster labels for each point?

Answer: Displaying cluster labels means showing which cluster each data point belongs to, allowing for an assessment of how well the clustering algorithm has grouped the data.

### 14. How can the elbow method help in selecting K for K-means?

Answer: The elbow method involves plotting the sum of squared distances from each point to its assigned cluster center for different values of K. The "elbow" point indicates the optimal K, where adding more clusters provides diminishing returns.

### 15. What are the potential challenges when implementing K-means clustering?

Answer: Challenges include choosing the right K, sensitivity to initialization (different initial centroids can yield different results), and the assumption that clusters are spherical and equally sized.

### 16. How can different values of K affect the results of clustering?

Answer: Different values of K can lead to varying cluster formations, with too few clusters oversimplifying the data and too many clusters resulting in overfitting and capturing noise.

### 17. What are some common metrics to evaluate clustering performance?

Answer: Common metrics include silhouette score, Davies-Bouldin index, and within-cluster sum of squares, which assess how well-separated the clusters are and how compact they are.

### 18. What are the practical applications of K-means clustering?

Answer: K-means clustering is used in customer segmentation, image compression, anomaly detection, and organizing large datasets into meaningful groupings for analysis.

### 19. Why is Python a popular choice for implementing web crawlers and NLP tasks?

Answer: Python is popular due to its simplicity, readability, and extensive libraries (like BeautifulSoup for web scraping and NLTK for NLP) that facilitate rapid development and prototyping.

### 20. How can data preprocessing impact the results of clustering algorithms like K-means?

Answer: Data preprocessing, such as scaling and normalization, can significantly impact clustering results by ensuring that features contribute equally to the distance calculations, leading to better-defined clusters.