Cycle 1: Python Basics & Mathematical Concepts

1. **What is a prime number? How can you determine if a number is prime?**
   - A prime number is a number greater than 1 that has no divisors other than 1 and itself. To check if a number is prime, 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.**
   - The Fibonacci series is generated by adding the two previous numbers to get the next. Starting with 0 and 1, the series looks like: 0, 1, 1, 2, 3, 5, 8, etc.
3. **What are the types of triangles based on side lengths?**
   - Equilateral (all sides equal), Isosceles (two sides equal), Scalene (all sides unequal).
4. **Define coprime numbers.**
   - Two numbers are coprime if their greatest common divisor (GCD) is 1.
5. **How do you find the roots of a quadratic equation?**
   - Use the quadratic formula: x=−b±b2−4ac2ax = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}x=2a−b±b2−4ac, where ax2+bx+c=0ax^2 + bx + c = 0ax2+bx+c=0.
6. **What is a perfect number?**
   - A perfect number is a number equal to the sum of its proper divisors (excluding the number itself). Example: 6 (divisors: 1, 2, 3; 1 + 2 + 3 = 6).
7. **Describe what an Armstrong number is.**
   - An Armstrong number (or narcissistic number) is a number that is equal to the sum of its digits raised to the power of the number of digits. Example: 153 (1^3 + 5^3 + 3^3 = 153).
8. **What are the differences between lists, tuples, dictionaries, and sets in Python?**
   - Lists: Ordered, mutable, allow duplicates.
   - Tuples: Ordered, immutable, allow duplicates.
   - Dictionaries: Unordered, mutable, key-value pairs.
   - Sets: Unordered, mutable, no duplicates.
9. **Explain how to add two lists element-wise.**
   - Use a loop or list comprehension, e.g., result = [a + b for a, b in zip(list1, list2)].
10. **What is bubble sort?**

- A sorting algorithm where adjacent elements are compared and swapped if they are in the wrong order. Repeated until the list is sorted.

11. **How can you count the vowels in a string using a dictionary?**
    - Loop through the string, check if each character is a vowel, and update the count in a dictionary.
12. **What is the purpose of the sum_of_digits function in the provided code?**
    - It calculates the sum of all digits in a number.
13. **How do you find absent digits in a mobile number?**
    - Compare the digits in the mobile number with the set of all digits (0-9) and find the missing ones.
14. **Explain the logic used to check if a number is perfect.**
    - Calculate the sum of the divisors of the number. If the sum equals the number, it is perfect.
15. **What does cmath library do in Python?**
    - Provides mathematical functions for complex numbers.
16. **How do you ensure that input values are valid in your programs?**
    - Use exception handling (try-except), input validation loops, and input checks.
17. **What is the significance of the swapped flag in the bubble sort algorithm?**
    - The swapped flag is used to optimize the bubble sort. If no swaps are made in a pass, the list is already sorted.
18. **How can the efficiency of checking for prime numbers be improved?**
    - Instead of checking divisibility up to nnn, check up to n\sqrt{n}n.
19. **How do you handle invalid input for a mobile number?**
    - Use validation checks to ensure the input is numeric and contains exactly 10 digits.
20. **Why is it important to round the roots of a quadratic equation to two decimal places?**
    - To make the roots more readable and practical for real-world applications.

Cycle 2: NumPy

1. **What is NumPy, and why is it important in Python programming?**
   - NumPy is a library for numerical computing in Python. It provides support for large multi-dimensional arrays and matrices, and mathematical functions to operate on them.
2. **Explain the difference between a list and a NumPy array.**
   - A list is a general-purpose container for items, whereas a NumPy array is optimized for numerical operations, providing better performance and efficiency.
3. **What are the main advantages of using NumPy arrays over Python lists?**
   - Faster operations, memory-efficient, support for multi-dimensional data, and a wide range of mathematical functions.
4. **What is the shape of a NumPy array, and how is it determined?**
   - The shape is a tuple that represents the dimensions of the array (e.g., rows, columns).
5. **What are the different ways to create a NumPy array?**
   - Using np.array(), np.zeros(), np.ones(), np.arange(), and np.linspace().
6. **Explain the concept of broadcasting in NumPy.**
   - Broadcasting allows NumPy to perform arithmetic operations on arrays of different shapes by automatically adjusting their dimensions.
7. **What is the purpose of the reshape() function in NumPy?**
   - It changes the shape of an array without changing its data.
8. **Describe the difference between deep copy and shallow copy in NumPy.**
   - A deep copy creates a new independent copy, while a shallow copy creates a reference to the original array.
9. **What are universal functions (ufuncs) in NumPy?**
   - Functions that operate element-wise on arrays, e.g., np.sqrt(), np.exp().
10. **How can you perform matrix multiplication in NumPy?**
    - Use the np.dot() function or the @ operator.
11. **What is the difference between np.sum() and np.mean()?**
    - np.sum() returns the sum of the elements, and np.mean() returns the average.
12. **Explain the purpose of the axis parameter in NumPy functions.**

- The axis parameter specifies the axis along which the operation is performed (e.g., rows or columns).

13. **What are masked arrays in NumPy?**
    - Arrays that allow you to mask certain elements (e.g., ignore them in calculations).
14. **Describe how to index and slice a NumPy array.**
    - Indexing allows you to access individual elements, and slicing allows you to extract a sub-array.
15. **What is the purpose of the np.unique() function?**
    - It returns the sorted unique elements of an array.
16. **Explain the concept of data types in NumPy arrays.**
    - NumPy arrays support various data types like int, float, and complex, which are consistent within an array.
17. **What is an array view in NumPy?**
    - A view is a new array object that looks at the same data, not a copy.
18. **How does NumPy handle multidimensional arrays?**
    - NumPy provides tools to handle multi-dimensional arrays with indexing, slicing, and reshaping.
19. **What is the purpose of the np.random module in NumPy?**
    - It generates random numbers and random arrays for various distributions.
20. **What are some common applications of NumPy in data science?**
    - Data manipulation, matrix operations, statistical analysis, and machine learning algorithms.

Cycle 3: Data Visualization (Matplotlib & Seaborn)

1. **What is Matplotlib, and what are its primary uses?**
    - Matplotlib is a plotting library in Python used to create static, interactive, and animated visualizations like line plots, bar plots, and histograms.
2. **Describe the purpose of the plt.subplot() function in Matplotlib.**
    - plt.subplot() divides the figure into a grid and allows you to plot multiple subplots in the same figure.

3. **How can you customize the style of a line plot in Matplotlib?**
    - You can customize the line style, color, marker, and width using parameters like linestyle, color, marker, and linewidth.
4. **Explain the use of markers in Matplotlib plots.**
    - Markers are used to highlight data points on a plot. Common marker types include dots, squares, and triangles, set using the marker parameter.
5. **What is the significance of the grid() function in plots?**
    - The grid() function adds grid lines to the plot, helping to read values more clearly.
6. **How do you display a title on a plot and align it?**
    - Use plt.title("Title") to add a title. You can align it with loc (left, center, or right) or pad (distance from the plot).
7. **What are histograms used for in data visualization?**
    - Histograms show the distribution of a dataset by grouping data points into bins or intervals.
8. **Describe the role of the bar() function in Matplotlib.**
    - The bar() function is used to create bar charts, which are ideal for comparing categorical data.
9. **How can you create a scatter plot in Matplotlib, and what is its purpose?**
    - A scatter plot is created using plt.scatter(), and it is used to visualize relationships between two variables.
10. **What is the purpose of using plt.tight_layout()?**
    - plt.tight_layout() adjusts the spacing between subplots to prevent overlap of titles, labels, or axes.
11. **Explain the difference between the displot() and histplot() functions in Seaborn.**
    - displot() is a figure-level function for univariate distributions, whereas histplot() is an axes-level function for plotting histograms.
12. **What types of plots can you create using the pairplot() function in Seaborn?**
    - pairplot() creates a grid of scatter plots for all pairwise relationships in a dataset.
13. **How do you display a legend in a Matplotlib plot?**
    - Use plt.legend() to add a legend to a plot, which helps identify different data series.

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

    - Axis labels describe the data represented and help viewers understand the context of the visualization.

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

    - The size parameter controls the size of the markers in a scatter plot, allowing for a third dimension of data.

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

    - describe() provides a summary of statistics for numerical columns in a DataFrame, such as mean, median, and standard deviation.

17. **How does the read_csv() function work in Pandas?**

    - read_csv() is used to load data from a CSV file into a Pandas DataFrame.

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

    - Common visualizations for categorical data include bar charts, pie charts, and count plots.

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

    - Improve visual appeal by customizing colors, styles, adding grid lines, and adjusting axis labels, titles, and legends.

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

    - Data visualization helps to easily communicate insights, trends, and patterns in the data, making analysis clearer and more accessible.

Cycle 4: Machine Learning

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

    - KNN (K-Nearest Neighbors) is a classification algorithm that assigns a class based on the majority class of its K nearest neighbors in the feature space.

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

    - The value of K is chosen based on cross-validation or trial-and-error. A small K can lead to overfitting, and a large K can smooth out the data.

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

    - Splitting the data ensures that the model is trained on one set of data and tested on a separate set to assess its performance.

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

- The classification report provides metrics such as precision, recall, f1-score, and accuracy for each class in a classification problem.

5. **What are some common datasets suitable for KNN classification?**
    - Examples include the Iris dataset, the Titanic dataset, and the MNIST dataset for image classification.
6. **What is the Naive Bayes algorithm, and in what scenarios is it used?**
    - Naive Bayes is a probabilistic classifier based on Bayes' theorem, assuming features are independent. It is often used for text classification, spam detection, and sentiment analysis.
7. **How does Gaussian Naive Bayes differ from Bernoulli Naive Bayes?**
    - Gaussian Naive Bayes assumes features follow a normal distribution, while Bernoulli Naive Bayes assumes binary features (0 or 1).
8. **What metrics are commonly used to evaluate classification models?**
    - Common metrics include accuracy, precision, recall, f1-score, confusion matrix, and ROC-AUC.
9. **How do you interpret the confusion matrix?**
    - The confusion matrix shows the counts of true positives, false positives, true negatives, and false negatives, allowing you to assess model performance.
10. **What is the Decision Tree algorithm, and how does it make decisions?**
    - Decision Trees classify data by splitting it at each node based on feature values, creating branches leading to decision outcomes.
11. **Explain how misclassification is determined in classification tasks.**
    - Misclassification occurs when the predicted label does not match the true label.
12. **What is the importance of preprocessing data in machine learning?**
    - Preprocessing ensures that the data is in a suitable format, normalized, and cleaned, which improves model performance.
13. **Describe the linear regression algorithm and its assumptions.**
    - Linear regression models the relationship between a dependent variable and one or more independent variables, assuming linearity, homoscedasticity, independence, and normality.
14. **What is the difference between simple and multiple linear regression?**
    - Simple linear regression uses one independent variable, while multiple linear regression uses two or more independent variables.

15. **How can you assess the performance of a regression model?**
    - Using metrics like Mean Squared Error (MSE), R-squared, and Adjusted R-squared.
16. **What role does a neural network play in predicting house prices?**
    - A neural network can model complex non-linear relationships between features (e.g., square footage, location) and house prices.
17. **Explain the significance of the activation function in a neural network.**
    - Activation functions introduce non-linearity, allowing the network to model complex relationships in the data.
18. **How does the train-validation-test split work in neural networks?**
    - Data is split into three sets: training (for training the model), validation (for tuning hyperparameters), and test (for final evaluation).
19. **What is the purpose of using Min-Max scaling in data preprocessing?**
    - Min-Max scaling normalizes the data, bringing it into a specific range, typically [0, 1], which helps improve model performance.
20. **What are some common libraries used for implementing machine learning algorithms in Python?**
    - Common libraries include Scikit-learn, TensorFlow, Keras, and PyTorch.

Cycle 5: Web Crawling, NLP, and Clustering

1. **What is a web crawler, and how does it work?**
    - A web crawler automatically browses the internet, extracts data from web pages, and stores it for analysis.
2. **How can you extract specific HTML elements, like <p> tags, using BeautifulSoup?**
    - Use BeautifulSoup's find() or find_all() methods to extract tags, e.g., soup.find_all('p').
3. **Why is it important to check for empty links when extracting hyperlinks?**
    - Empty links can lead to errors or result in missing or incomplete data when scraping websites.
4. **Explain the concept of n-grams in Natural Language Processing.**
    - N-grams are sequences of N consecutive words or characters in a text. They are useful in NLP for capturing context.

5. **How would you implement n-grams without using a library?**
   - Split the text into words, then use a loop to create sequences of N words.
6. **What are the advantages of using libraries like NLTK for NLP tasks?**
   - NLTK provides built-in functions for tokenization, stemming, stop word removal, and many other NLP tasks, saving time and effort.
7. **What is the purpose of stop word removal in text processing?**
   - Stop word removal eliminates common but irrelevant words (e.g., "the," "is") to focus on meaningful words in text analysis.
8. **Describe the process of tokenization in NLP.**
   - Tokenization is the process of splitting text into smaller units, such as words or sentences, for further analysis.
9. **How does K-means clustering work?**
   - K-means is an iterative algorithm that divides data into K clusters by minimizing the variance within each cluster.
10. **What factors should you consider when choosing the number of clusters (K) in K-means?**
    - Factors include the dataset's size, the data's natural grouping, and the "elbow method" to find an optimal K.
11. **How can you visualize clusters in a scatter plot?**
    - Use colors or markers to distinguish different clusters in a scatter plot.
12. **Why is it important to visualize the cluster centers in K-means?**
    - Cluster centers represent the average of all points in a cluster and provide insight into the structure of the data.
13. **What does it mean to display cluster labels for each point?**
    - Cluster labels assign each data point to its respective cluster, making it easier to analyze groupings.
14. **How can the elbow method help in selecting K for K-means?**
    - The elbow method involves plotting the sum of squared distances (within-cluster variance) for different values of K. The "elbow" point indicates the optimal K.
15. **What are the potential challenges when implementing K-means clustering?**
    - Challenges include selecting the right K, dealing with outliers, and ensuring clusters are well-separated.

16. **How can different values of K affect the results of clustering?**
    - A low K can result in underfitting, while a high K may lead to overfitting and less meaningful clusters.
17. **What are some common metrics to evaluate clustering performance?**
    - Metrics include Silhouette score, Davies-Bouldin index, and Inertia.
18. **What are the practical applications of K-means clustering?**
    - Applications include customer segmentation, image compression, and anomaly detection.
19. **Why is Python a popular choice for implementing web crawlers and NLP tasks?**
    - Python has libraries like BeautifulSoup, Scrapy, NLTK, and SpaCy, which simplify web crawling and NLP tasks.
20. **How can data preprocessing impact the results of clustering algorithms like K-means?**
    - Proper preprocessing (e.g., scaling, handling missing values) can improve the quality of clusters and the algorithm's performance.