The Answer of These questions were related to task 1, therefore Answer all the question in the document separately:

Q 1: Which Python libraries did you find most useful in loading and exploring the dataset?

Ans: When loading and exploring datasets in Python, several libraries are commonly used.

They are: -

1. Pandas: Pandas is a powerful library for data manipulation and analysis. It provides data structures like Data Frames that allow easy loading, indexing, and filtering of datasets. Pandas also offers functions for handling missing values, grouping, sorting, and summarizing data.
2. NumPy: NumPy is a fundamental library for numerical computations in Python. It provides powerful data structures and functions for handling arrays and matrices. NumPy is often used in conjunction with Pandas for efficient numerical operations on datasets.
3. Scikit-learn: Scikit-learn is a comprehensive machine learning library in Python. It provides a wide range of tools for data pre-processing, feature selection, model training, and evaluation. Scikit-learn includes functions for splitting datasets into training and testing sets, as well as various evaluation metrics for classification and regression tasks.

So, these libraries, namely Pandas, NumPy, and Scikit-learn, form a powerful toolkit for loading, exploring, and analysing datasets in Python.

Q 2: What pre-processing steps did you find necessary to apply to the heart dataset?

Ans: To pre-process the Heart Disease dataset, the following pre-processing steps are typically necessary:

1. Handling Missing Values:
   * Identify missing values: Check if the dataset contains any missing values entries.
   * Fill or drop missing values: Depending on the amount and nature of missing values, you can either fill them with appropriate values (e.g., mean, median, or mode) or drop the rows or columns with missing values.
2. Encoding Categorical Variables:
   * Identify categorical variables: Determine which columns in the dataset represent categorical variables.
   * Label encoding: Convert categorical variables into numerical labels using techniques like label encoding or one-hot encoding. Label encoding assigns a unique numerical label to each category.
3. Scaling Numerical Variables:
   * Identify numerical variables: Identify the columns that represent numerical variables requiring scaling.
   * Standardization: Scale the numerical variables using techniques like standardization (also known as z-score normalization). Standardization transforms the variables to have zero mean and unit variance.

Q 3: What metrics were used to evaluate the Classification problem and why?

Ans: The metrics can be used to evaluate the performance of a classification problem. The choice of metrics depends on the specific requirements of the problem and the desired evaluation aspects. Here are some commonly used metrics for evaluating classification problems:

1. Accuracy: Accuracy is the most basic evaluation metric and measures the overall correctness of the predictions. It calculates the ratio of correctly predicted samples to the total number of samples. However, accuracy alone may not provide a complete picture of model performance, especially when dealing with imbalanced datasets.
2. Precision: Precision is the ratio of true positive predictions to the total number of positive predictions. It measures the accuracy of positive predictions and is particularly useful when the cost of false positives is high. High precision indicates a low rate of false positives.

Q 4: How did you detect overfitting in the model and what strategies did you use to mitigate?

Ans: To detect overfitting in the model and mitigate its impact, the following strategies can be employed:

1. Train/Test Performance Comparison:
   * Compare the performance of the model on the training set and the separate test set. If the model performs significantly better on the training set than the test set, it indicates overfitting.
2. Cross-Validation:
   * Compute the average performance across all folds to obtain a more reliable estimate of the model's generalization ability.
   * If the model consistently performs well across different folds, it indicates better generalization and reduced overfitting.
3. Early Stopping:
   * Implement early stopping during model training, where the training process stops if the performance on the validation set starts to degrade.
   * Early stopping prevents the model from continuing to train and overfit on the training data, resulting in better generalization.
4. Data Augmentation:
   * If the dataset is limited, data augmentation techniques can be employed to artificially increase the size and diversity of the dataset.
   * Data augmentation introduces variations to existing data, such as rotations, translations, or adding noise, which helps the model generalize better.

Q 5: How did you choose the number of clusters for the K-means algorithm? explain why?

Ans: Choosing the number of clusters for the K-means algorithm requires careful consideration. Here are a few approaches commonly used to determine the appropriate number of clusters:

1. Elbow Method:
   * The Elbow Method involves plotting the within-cluster sum of squares (WCSS) against the number of clusters.
   * WCSS represents the sum of squared distances between each data point and its nearest cluster centroid.
   * As the number of clusters increases, WCSS tends to decrease because more clusters allow for better data point assignment.
   * However, beyond a certain point, the rate of improvement diminishes significantly, resulting in an elbow-like bend in the plot.
   * The number of clusters corresponding to the elbow point is often considered a reasonable choice.

2. Visual Inspection:

* + Plotting the data points and cluster assignments can provide visual insights into the inherent structure and natural grouping.
  + By visually inspecting the scatterplot or other relevant visualizations, you can get an intuitive sense of the number of distinct clusters present.