**Task 1 Question :**

**1] Which Python libraries did you find most useful in loading and exploring the dataset?**The most useful Python libraries for loading and exploring the dataset were:

* **Pandas**: For data manipulation and analysis, including loading the dataset, handling missing values, and transforming data.
* **NumPy**: For numerical operations and handling arrays.
* **Matplotlib and Seaborn**: For data visualization, including creating plots and visualizing correlations and distributions in the dataset.
* **Scikit-learn (sklearn)**: For preprocessing the data, implementing machine learning models, and evaluating model performance.

**2] What preprocessing steps did you find necessary to apply to the heart dataset?**The necessary preprocessing steps applied to the heart dataset were:

* **Handling Missing Values**: Using SimpleImputer to fill in missing values with the mean of the respective columns.
* **Encoding Categorical Variables**: Using LabelEncoder to convert categorical variables into numerical format, which is required for most machine learning algorithms.
* **Scaling Numerical Variables**: Using StandardScaler to standardize the features to have a mean of 0 and a standard deviation of 1, which helps in improving the performance of machine learning models.

**3] What metrics were used to evaluate the Classification problem and why?**The metrics used to evaluate the classification problem were:

* **Accuracy**: Measures the ratio of correctly predicted instances to the total instances.
* **Precision**: Measures the ratio of correctly predicted positive observations to the total predicted positives.
* **Recall (Sensitivity)**: Measures the ratio of correctly predicted positive observations to all the actual positives.
* **F1 Score**: The harmonic mean of precision and recall, providing a balance between the two.
* **ROC AUC Score**: Measures the area under the receiver operating characteristic curve, indicating how well the model distinguishes between classes.

These metrics were chosen because they provide a comprehensive understanding of the model's performance from different perspectives, particularly in imbalanced datasets where accuracy alone might be misleading.

**4] How did you detect overfitting in the model and what strategies did you use to mitigate it?**Overfitting was detected by comparing the model's performance on the training data and the testing data. If the model performed significantly better on the training data than on the testing data, it indicated overfitting. To mitigate overfitting, the following strategies were used:

* **Cross-validation (10-fold)**: By dividing the dataset into 10 parts, training the model on 9 parts and testing it on the remaining part, and repeating this process 10 times with different testing parts each time. This ensures the model generalizes well on different subsets of the data.

**5] How did you choose the number of neighbors for the KNN algorithm? Explain why.**The number of neighbors (k) for the KNN algorithm was chosen based on cross-validation performance. The process involved:

* **Starting with an initial value of k** (e.g., k=5).
* **Evaluating the performance** of the KNN model using cross-validation for different values of k (e.g., k=1 to k=20).
* **Selecting the k** that provided the best cross-validation accuracy, ensuring that the model was neither overfitting nor underfitting.

Typically, an odd number is selected to avoid ties in voting among neighbors. The chosen k value balances bias and variance, optimizing the model's performance on unseen data.