Section-A

1.
a. A. **Correlation**: In a Random Forest, the individual decision trees in the ensemble are very similar in their predictions. They tend to make the same decisions for the same data points. On one hand, high correlation can lead to increased accuracy, but it can also make the model being overfitted on the training data on the other data.

   B. **Diversity**: Diversity is when the individual decision trees are different. They make other predictions based on their unique perspectives. Diversity might have different views and helps Random Forest make accurate and robust decisions as it gets different input. Still, it can reduce the overall accuracy and reliability of the model.

   The trade-off is about finding the right balance in making the trees a little related and different. We want the trees to mostly agree (a bit of correlation) to be good at solving problems together. But we also want them to be a little different (diversity) to avoid big mistakes and find the best answers. It's like trees having a variety to make sure it is both accurate and reliable.

b. 'Curse of Dimensionality' can sometimes be a problem with Naive Bayes. It means we have too many features or attributes to be looking at one time.
   Some of the possible problems are as follows:
   a. **Complexity:** The cost of calculating the probability of each feature under each target will take a long time and memory, eventually making it a slower process.
   b. **Overfitting:** The model might remember all the values and have low bias but will give high variance.

   Some methods to mitigate the problem:
   a. **Dimensionality Reduction:** Machine learning methods like PCA, t-SNE, etc., will help reduce the number of features while keeping the vital information.
   b. **Regularization:** Use regularization methods that will help in preventing overfitting and high complexity of the model in high dimensional space.
   c. **Ensemble Methods:** Instead of using a single Naive Bayes model for predictions, you can combine multiple models that will collaboratively improve prediction accuracy and mitigate the challenges posed by high-dimensional data.

c. If the Naive Bayes Classifiers face a value not present in the training dataset, it will not be able to predict the probability of that class as it doesn't have any prior information about the same.
   It will affect the inference results as it will directly assign a probability of 0 for an unseen attribute, which can lead to wrong predictions.

Some methods to mitigate the problem:

a. **Laplace Smoothing:**
   - Laplace smoothing is like adding a boost to the counts when calculating probabilities. This prevents zero probabilities and helps when encountering something the model hasn't seen before.
   - For example, if your spam classifier has never seen a particular word, Laplace smoothing ensures it doesn't give a zero chance of that word being spam.
   - P(x|c) = (count(x, c) + 1) / (count(c) + |V|)
     - count(x,c): number of instances of attribute x in class c
     - V: number of unique attribute values
     - count(c): total number of instances of class c

b. **Frequent Color Mapping:**
   - For categorical attributes, handle unseen values by mapping them to the most common category, like "unknown."
   - For example, if your model knows "black," "blue," and "brown" as car colors and sees "purple," you can label it as "unknown" or choose the most common color, like "blue."

d. Yes, Information Gain in decision trees can lead to biased decisions and can favor attributes with more options, as attributes with more values might give detailed information.
To mitigate this bias, we can use a different criterion, i.e., 'Gain Ratio.'
The Gain Ratio attempts to lessen the bias of Information Gain on highly branched predictors by introducing a normalizing term called Intrinsic Information.
The Intrinsic Information (II) is the entropy of sub-dataset proportions, which measures the potential uncertainty introduced by the attribute's cardinality.

The formula of Intrinsic Information is:

$$II = -(\sum \frac{|D_j|}{|D|} * \log_2 \frac{|D_j|}{|D|})$$

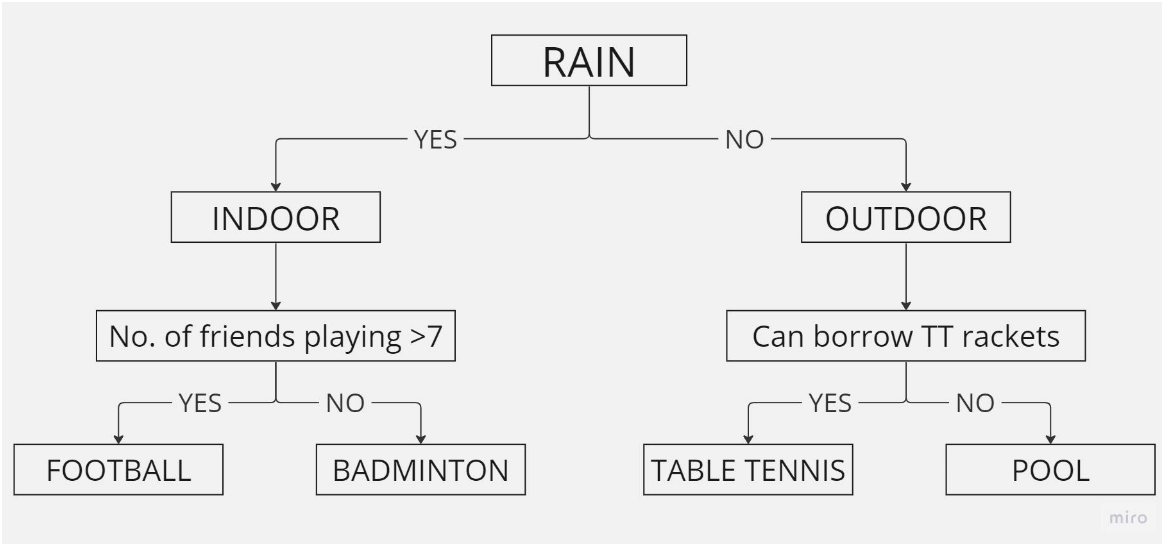
|Dj|/|D| = number of samples in j-th subset / number of total samples available

The Gain Ratio is:

$$GainRatio = \frac{\text{Information Gain}}{\text{Intrinsic Information}}$$

For example, we are building a decision tree to classify fruits and have 'Fruit ID' and 'Color' as attributes. Now, 'Fruit ID' will have many more unique values than 'Color.'
If we use Information Gain, it might prefer 'Fruit ID' because of its high cardinality, even though 'Color' might be more helpful. However, if we use Gain Ration, it considers both cases and as a result, 'Color' might be a more informative choice.

2.



a.

b.

b) P(App predicts Rain) = 0.3
P(App predicts Clear) = 0.7

P(App predicts Rain | Rainy) = 0.8
P(App predicts Clear | Clear) = 0.9

* P(App predicts Rain) = P(AR)
P(App predicts Clear) = P(AC)

$P(AR) = P(AR|R)\,P(R) + \cancel{P(AR|C)}\ P(AR|C)\,P(C)$

$0.3 = 0.8 * P(R) + (1-0.9) * (1-P(R))$

$0.3 = 0.1 + 0.7 * P(R)$

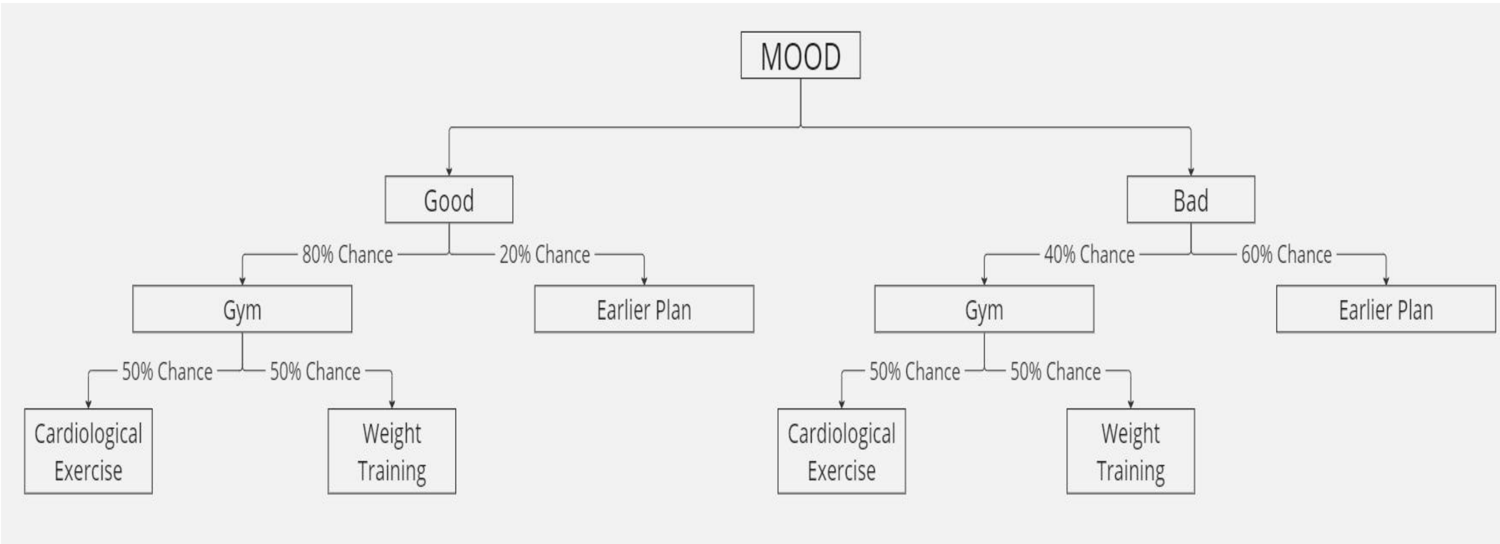
$P(R) \approx 0.2857$

$$P(R|AR) = P(AR|R) * \frac{P(R)}{P(AR)}$$

$$= 0.8 * \frac{0.2857}{0.3}$$

$$\approx 0.7619$$

∴ Probability of raining given App predicts 'Rainy' ≈ 0.7619 c.



<u>NOTE:</u> Earlier Plan refers to the decision tree made in Part A of Ques 2, which has Rain, No. of friends, etc. as attributes.

GM : Good mood
BM : Bad mood
GC : Cardiological exercise at Gym
GW : Weight Training at Gym
SP : Stick to Plan

$$P(Gym) = 0.8 * P(GM) + 0.4 * P(BM)$$
$$P(SP) = 0.2 * P(GM) + 0.6 * P(BM)$$

$$P(GC) = 0.5 * 0.8 * P(GM) + 0.5 * 0.4 * P(BM)$$

$$P(GW) = 0.5 * 0.8 * P(GM) + 0.5 * 0.4 * P(BM)$$

$P(GM) = 0.6$ (Good mood)
$P(BM) = 0.4$ (Bad mood)
F : amount of sleep last night

$P(F=7 \mid GM) = 0.7$
$P(F=7 \mid BM) = 0.45$

$$* \quad P(F=7) = P(F=7 \mid GM) * P(GM) + P(F=7 \mid BM) * P(BM)$$
$$= 0.7 * 0.6 + 0.45 * 0.4$$
$$= 0.6$$

$$* \quad P(GM \mid F=7) = P(F=7 \mid GM) * \frac{P(GM)}{P(F=7)}$$
$$= 0.7 * \frac{0.6}{0.6}$$
$$= 0.7$$

$$* \quad P(BM \mid F=7) = P(F=7 \mid BM) * \frac{P(BM)}{P(F=7)}$$
$$= 0.45 * \frac{0.4}{0.6}$$
$$= 0.3$$

d.

$$* \quad P(GC) = 0.5 * 0.8 * P(GM \mid F=7) + 0.5 * 0.4 * P(BM \mid F=7)$$
$$= 0.34$$

$$* \quad P(GW) = 0.5 * 0.8 * P(GM \mid F=7) + 0.5 * 0.4 * P(BM \mid F=7)$$
$$= 0.34$$

$$* \quad P(SP) = 0.2 * P(GM \mid F=7) + 0.6 * P(BM \mid F=7)$$
$$= 0.32$$

Rahul is most likely to do cardiological exercise and weight training.

Most Likely Event: ‘GYM’ / (‘Cardiological Activities’ and ‘Weight Training’)
Since P(SP) <= P(GC) and P(SP) <= P(GW), the probability of splitting each node of SP, i.e., football, table tennis, etc. will be <= 0.32 as there will be a multiplication of 0.32 and number between 0 and 1, so it will be <= 0.32.

Section-B

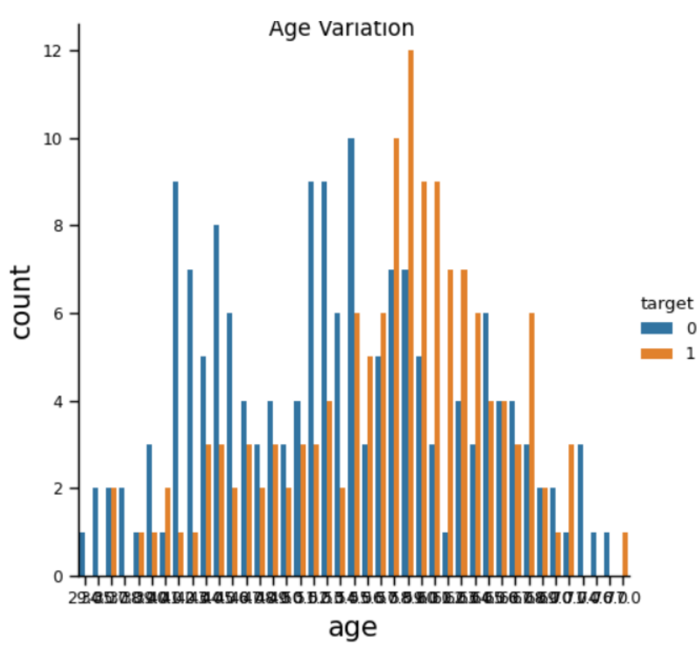
a. Initially, converting the target into a binary classification using:
   - data['target'] = data['target'].apply(lambda x: 1 if x != 0 and x != 1 else x)

b. EDA

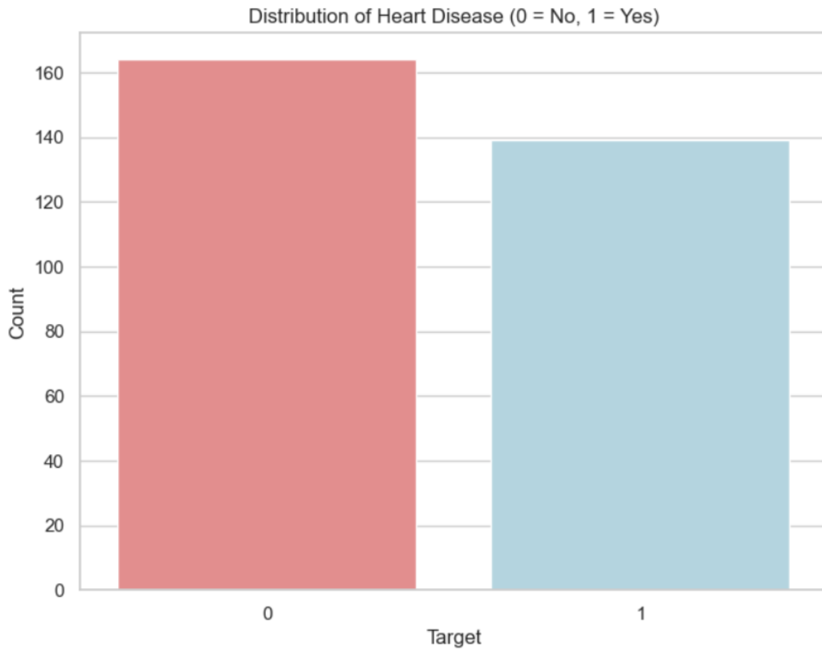
-

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	target
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.438944	0.679868	3.158416	131.689769	246.693069	0.148515	0.990099	149.607261	0.326733	1.039604	1.600660	0.458746
std	9.038662	0.467299	0.960126	17.599748	51.776918	0.356198	0.994971	22.875003	0.469794	1.161075	0.616226	0.499120
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	1.000000	0.000000
25%	48.000000	0.000000	3.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000
50%	56.000000	1.000000	3.000000	130.000000	241.000000	0.000000	1.000000	153.000000	0.000000	0.800000	2.000000	0.000000
75%	61.000000	1.000000	4.000000	140.000000	275.000000	0.000000	2.000000	166.000000	1.000000	1.600000	2.000000	1.000000
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	3.000000	1.000000

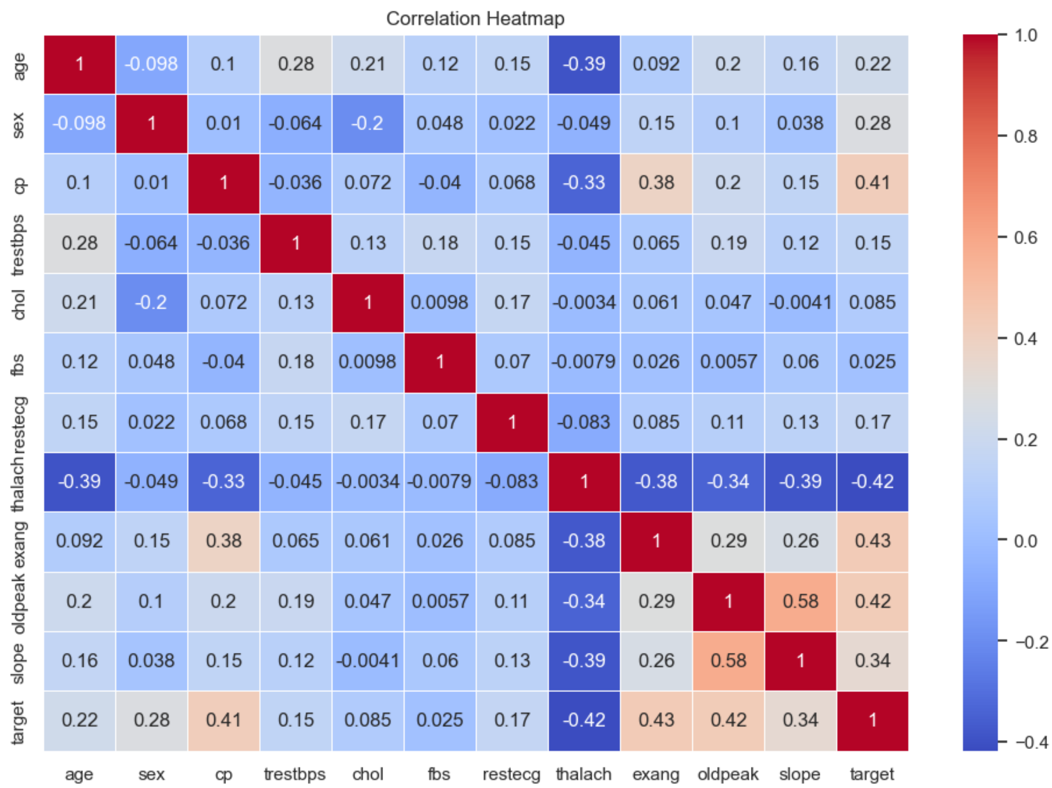
- Looking at the Count vs Age graph, we can say 'Heart Disease on an avg. increases with age'



- No class imbalance



- Highest Co-relation can be seen between ‘oldpeak’ and ‘slope’
- Co-relation can also be seen between ‘cp’ and ‘target’ & ‘exang’ and ‘target’



Dataset statistics

Number of variables	14
Number of observations	303
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	33.3 KiB
Average record size in memory	112.4 B

Variable types

Numeric	5
Categorical	9

-
- EDA has been done for each variable and there’s a separate graph being generated with information which can be seen in the code.

c. Pre-processing Steps:
    - Missing values were handles:
        - ‘ca’ column missing values were replaced by mean
        - ‘thal’ column missing values were replaced with mode
    - Standardization was done on numeric variable values

d. Decision trees with entropy and gini criteria were made along with the following accuracy scores:

```
Accuracy with 'entropy' criterion: 0.7868852459016393
Accuracy with 'gini' criterion: 0.7540983606557377
The best criteria is 'entropy'
```

e. Using the following list of parameters, grid search was performed and the results were as follows:

```
# Defining the hyperparameters and their possible values
parameters = {
    'min_samples_split': [2, 5, 9, 10, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 50, 100],
    'max_features': ['sqrt', None, 0.1, 'log2', 0.2, 'auto', 0.3]  # Added more values
}
```

```
Best Hyperparameters: {'max_features': None, 'min_samples_split': 19}
Test Accuracy with Best Hyperparameters: 0.8524590163934426
```

f. Grid search for conducted to get various best parameters for the Random Forest Classifier and the result were as follows:

```
param_grid_rf = {
    'n_estimators': [50, 60, 70, 75, 80],
    'max_depth': [None, 1, 2, 3, 4, 5, 6, 7 ,10],
    'min_samples_split': [1, 2, 3, 4, 5, 6]
}
```

```
Best Hyperparameters for Random Forest: {'max_depth': 3, 'min_samples_split': 5, 'n_estimators': 60}
Test Accuracy with Best Hyperparameters for Random Forest: 0.9016393442622951
```

g. Finally, a classification report was generated from which we can say that the model was performing well as precision recal f1 score were above 85%.

```
Classification Report for Random Forest:
              precision    recall  f1-score   support

           0       0.85      0.97      0.90        29
           1       0.96      0.84      0.90        32

    accuracy                           0.90        61
   macro avg       0.91      0.90      0.90        61
weighted avg       0.91      0.90      0.90        61
```