Shourya Jindal
2020336
ML Assignment 2

Section A

(Q1)

(a) In random forests (RF), correlation refers to the degree to which individual decision trees, in the RF are similar to each other.

High Correlation = Trees make similar error and decision on same data pts

Low " = Make diff. decisions on same data pts.

Diversity refers to differences among the trees in the RF.
↳ diff. feature selections, thresholds, rules.
High diversity ⇒ Reduces overfitting, reduces variance.

So, tradeoff b/w these 2 is that

- if we increase correlation too much, it is similar to having single decision tree. Leads to Overfitting ⇒ ~~High Bias~~ $\boxed{\text{Low Bias, High Variance}}$ ⇐ does not perform well on unseen data
- ~~But~~ And if increase diversity too much, it leads to lack of consensus in the predictions. Reduces overall accuracy. It reduces overfitting, thus reducing variance.

(b) In 'naive Bayes', the "curse of dimensionality" becomes an issue, when 'no. of features' i.e the dimensionality of data becomes very large. It leads to very high computational ~~complex~~ complexity, overfitting. In higher dimensions data becomes sparse making it difficult to calculate probabilities in 'Naive Bias' also leading to increased computational complexity. Also, leads to overfitting, thus poor performance on unseen data.

Strategies to mitigate this:

- Feature Selection: Use only relevant features and discard redundant ".
- Dimensionality Redn: Use methods like PCA, TSNE to reduce dimensionality
- Regularization: Use L1, L2 methods to prevent overfitting
- Ensemble Methods: Use ensemble methods like boosting, random forests ⇒ More effective in higher dimensions.

(c) Yes, if some value of attributes which was not present in the training set is encountered, it will affect the inference results.

Problems:

- when unseen value of attri. is encountered, it will assign it prob. 0.
- This can lead to wrong predictions
- Thus leading to high variance i.e low accuracy on unseen data.
- Accuracy reduces

Soln: • Laplace smoothing: Add a small value (generally 1, to count of each attribute value pair. ⟹ No zero prob.

- Use prior prob: If we have access to prior prob. for attri values, these can be given to the model to help estimate unseen values prob.

eg: Email classification < spam / not spam

Training:

Email	Lottery	free	Spam / Not
1	yes	yes	yes
2	yes	No	yes
3	No	yes	NO
4	No	No	No

Test sample: emails: "Lottery free hello". $= X^{new}$

In this, $P(\text{"hello"} \mid \text{"spam"}) = 0$

$\Rightarrow P(\text{Spam} \mid X^{new}) = 0$ although it contain both "lottery" and "free". but since "hello" is unseen, it is not declared as spam.

but if Laplace Smoothing will be used it will prob > 0.

So, ~~Prob.~~:
possible outcomes and prob: Using formulas from last part (c)

* Cardio:

$P[\text{Cardio}] = (0.42)(0.8).(0.5) + (0.18)(0.4)(0.5)$

$= 0.168 + 0.36 = 0.528$

* weights:

$P[\text{weights}] =$ same as cardio

$= 0.528$

* Plan: $P[\text{Plan}] = (0.42).(0.2) + (0.18)(0.6)$

$= 0.084 + 0.108$

$= 0.192$

So, To calculate exactly which sport he will play in info. is not available.

Most likely he will do:

So, He will visit gym and would either do Cardio OR weights with equal probability.

$= \boxed{0.528}$.

Now, we have to find

$$P[\text{Rain} \mid \text{Predicts Rainy}] = ??$$

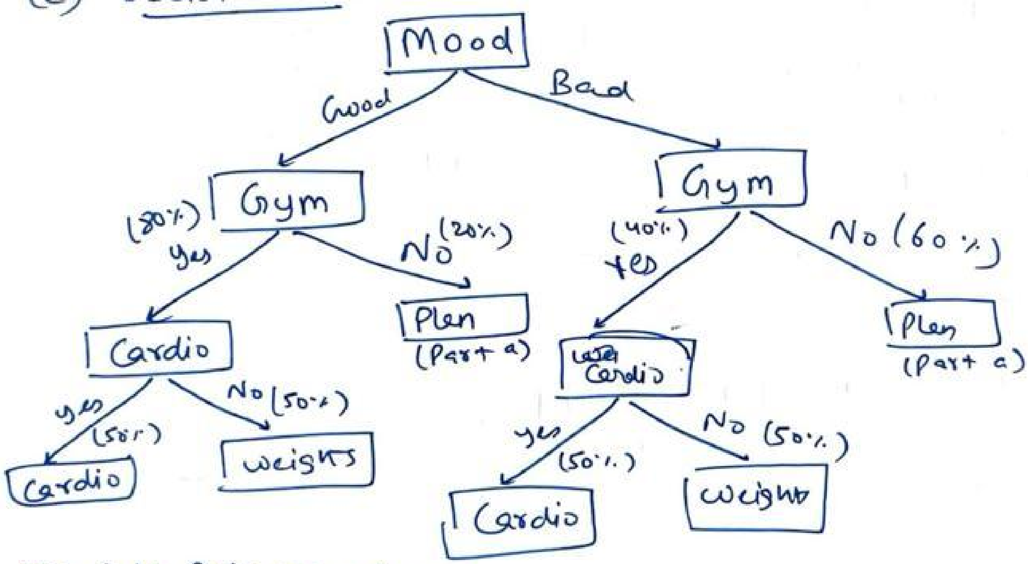
$$= \frac{P[\text{Predicts Rainy} \mid \text{Rain}] \times P[\text{Rain}]}{P[\text{Predicts Rainy}]}$$

(using given values and eqn (2))

$$= \frac{0.8 \times 2/7}{0.3} = \frac{1.6}{2.1} = \boxed{0.76}$$

↳ Ans

(C) Decision Tree:



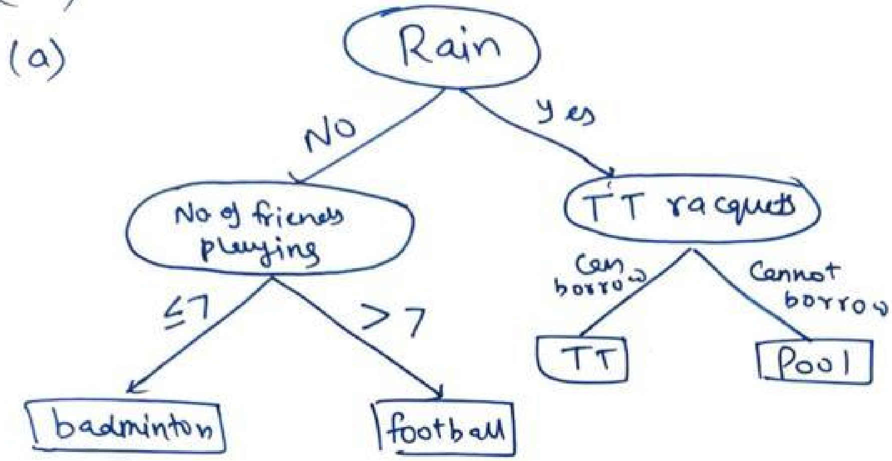
Possible Outcomes and prob.:

* $\boxed{\text{Cardio}}$: $P(\text{Good Mood}) \cdot \underbrace{P(\text{Gym} \mid \text{Good Mood})}_{=0.8} \cdot \underbrace{P(\text{Cardio} \mid \text{Gym})}_{=0.5}$
  $+ P(\text{Bad Mood}) \cdot \underbrace{P(\text{Gym} \mid \text{Bad Mood})}_{=0.4} \cdot \underbrace{P(\text{Cardio} \mid \text{Gym})}_{=0.5}$

* $P\,\boxed{\text{Weights}}$: $P(\text{Good Mood}) \cdot \underbrace{P(\text{Gym} \mid \text{Good Mood})}_{=0.8} \cdot \underbrace{P(\text{weights} \mid \text{Gym})}_{=0.5}$
  $+ P(\text{Bad mood}) \cdot \underbrace{P(\text{Gym} \mid \text{Bad Mood})}_{=0.4} \cdot P(\text{weights} \mid \text{Gym})$

(Q2)

(a)



Possible Outcomes & their Probabilities:

* Badminton: P(Badminton) = P(No Rain) * P(No. of friends playing ≤ 7 | No Rain)
* Football: P(football) = P(No Rain) * P(No. of playing friends > 7 | No Rain)
* TT: P(TT) = P(Rain) * P(Can borrow racquets | Rain)
* Pool: P(Pool) = P(Rain) * P(cannot borrow racquets | Rain)

(b) P(Predicts Rainy) = 0.3 P(Predicts Clear) = 0.7

P(Predicts Rainy | Rain) = 0.8 P(Predicts Clear | Clear) = 0.9

$\boxed{P(\text{Predicts Rain} \mid \text{Clear}) = 0.1}$ — ①

Now, first we will calculate P[Rain]
for this we will use total probability theorem:

P[Predicts Rainy] = P[Rain] * P[Predicts Rainy | Rain] +
P[Clear] * P[Predicts Rainy | Clear]

(Using given & and eqn ①)

$\Rightarrow$ 0.3 = 0.8 P[Rain] + 0.1 P[Clear]

$\Rightarrow$ 0.3 = 0.8 P[Rain] + 0.1 (1 − P[Rain])

$\Rightarrow \boxed{P(\text{Rain}) = \frac{2}{7}}$ — ②

Now, we have to find

$$P[\text{Rain} \mid \text{Predictics Rainy}] = ??$$

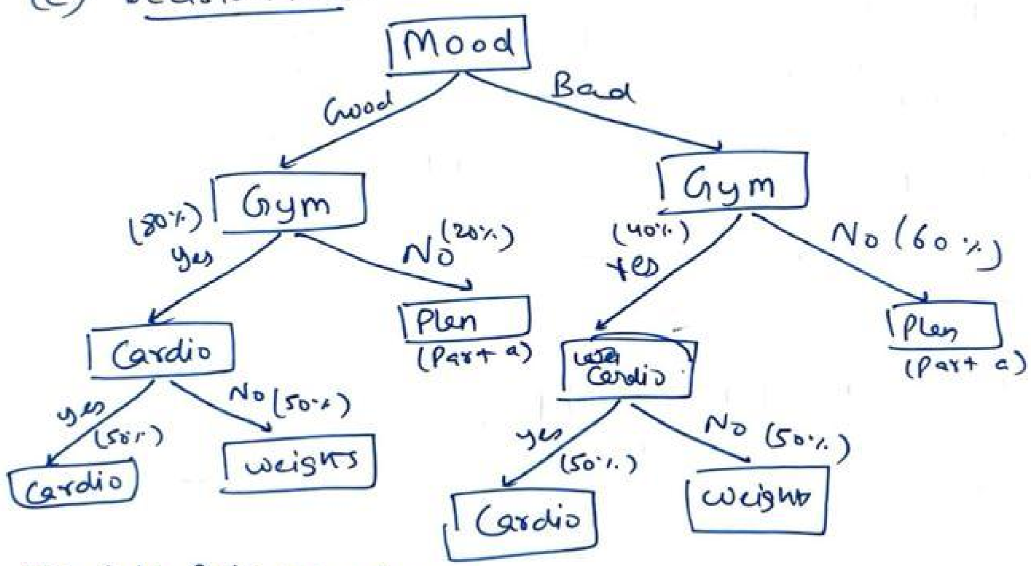
$$= \frac{P[\text{Predicts Rainy} \mid \text{Rain}] \times P[\text{Rain}]}{P[\text{Predicts Rainy}]}$$

(using given values and eqn (2))

$$= \frac{0.8 \times 2/7}{0.3} = \frac{1.6}{2.1} = \boxed{0.76}$$

↳ Ans

(C) Decision Tree:



Possible Outcomes and prob.:

* $P(\boxed{\text{Cardio}})$: $P(\text{Good Mood}) \cdot \underbrace{P(\text{Gym} \mid \text{Good Mood})}_{=0.8} \cdot \underbrace{P(\text{Cardio} \mid \text{Gym})}_{=0.5}$
$+ P(\text{Bad Mood}) \cdot \underbrace{P(\text{Gym} \mid \text{Bad Mood})}_{=0.4} \cdot \underbrace{P(\text{Cardio} \mid \text{Gym})}_{=0.5}$

* $P(\boxed{\text{Weights}})$: $P(\text{Good Mood}) \cdot \underbrace{P(\text{Gym} \mid \text{Good Mood})}_{=0.8} \cdot \underbrace{P(\text{Weights} \mid \text{Gym})}_{=0.5}$
$+ P(\text{Bad Mood}) \cdot \underbrace{P(\text{Gym} \mid \text{Bad Mood})}_{=0.4} \cdot P(\text{Weights} \mid \text{Gym})$

(d) Yes, if some attri have more cardinality, there can be bias in splitting decision tree node using Info. Gain.

This is because cardinality means that no. of distinct values an attribute can take.

If an attribute has high cardinality
↳ they can be split into more subset (more branches in a tree)
↳ low impurity
↳ More information Gain
↳ Thus, biasedness.

* Other criterion that can be used:

- Gini Index: $Gini(S) = 1 - \sum p_i^2$
  it is less sensitive to cardinality
- Gain Ratio: It takes into cardinality and penalizes attributes with high ".

Example:

Suppose, we are building a decision tree to predict, ~~which type of~~ ice-cream sales.

~~Attributes consider~~

Consider 3 attributes:

1. Age Group: {<18, 18-35, 36-60, >60}
   ↳ low cardinality
2. Weather conditions: {Rainy, hot, cold}
   ↳ low cardinality
3. Flavour = {Chocolate, butterscotch, berry, ..}
   ↳ High cardinality

If we use IG as criterion, it can favour 'flavour' attribute (due to its high cardinality).

But, this may not be optimal approach, as sales of ice-cream are generally affected most by the factors like 'weather condition'.

* Rest of the outcomes are same
just use previous formula ~~and multiply them with~~ as do following:

$$\underbrace{P'(\text{Event})}_{\text{new prob}} = P(\text{Plan}) \cdot \underbrace{P(\text{Event})}_{\text{older prob}}$$

where $P(\text{Plan}) = P(\text{Good Mood}) \cdot \overbrace{P(\text{No Gym} \mid \text{Good Mood})}^{=0.2}$
$+ P(\text{Bad Mood}) \cdot \underbrace{P(\text{" "} \mid \text{Bad Mood})}_{=0.6}$

(d) $P(\underbrace{\text{Good Mood}}_{=(GM)}) = 0.6$

$P(\underbrace{\text{Bad Mood}}_{=(BM)}) = 0.4$

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

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

Assuming he slept 7 hrs one night before

so, $P[F=7] = 1$

~~We are assuming that he slept~~ ~~$P(GM) = 0.6$ & $P(BM) = 0.4$ are given of one day earlier.~~

~~So, On each last night~~

~~$P[F=7] = P[GM] \cdot P[F=7 \mid GM] + P[BM] \cdot P[F=7 \mid BM]$~~

~~$= (0.6)(0.7) + (0.4)(0.45)$~~

~~$\boxed{P[F=7] = 0.6}$~~

Now, for next day: $P[GM \mid F=7] = \dfrac{P[F=7 \mid GM] \cdot P[GM]}{P[F=7]}$

$= (0.7)(0.6) = 0.42$

$P[BM \mid F=7] = \dfrac{P[F=7 \mid BM] \cdot P[BM]}{P[F=7]}$

$= (0.45)(0.4) = 0.18$

Section B

PART A: Preprocessing and EDA

Read the data and store it in the pandas' data frame.
In total, 13 features and 1 target column.
Toral rows: 303.

Preprocessing:

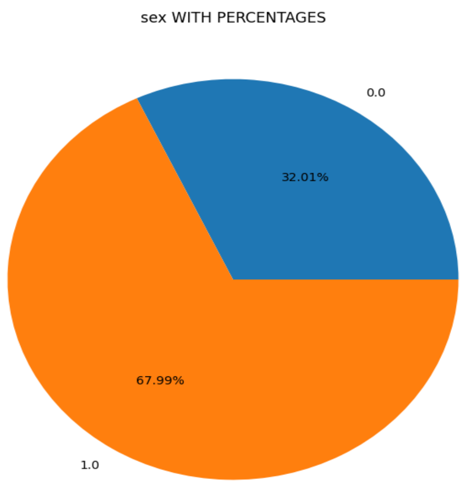
- Converted the classification to binary classification.
  Target values with value > 0 were converted to 1

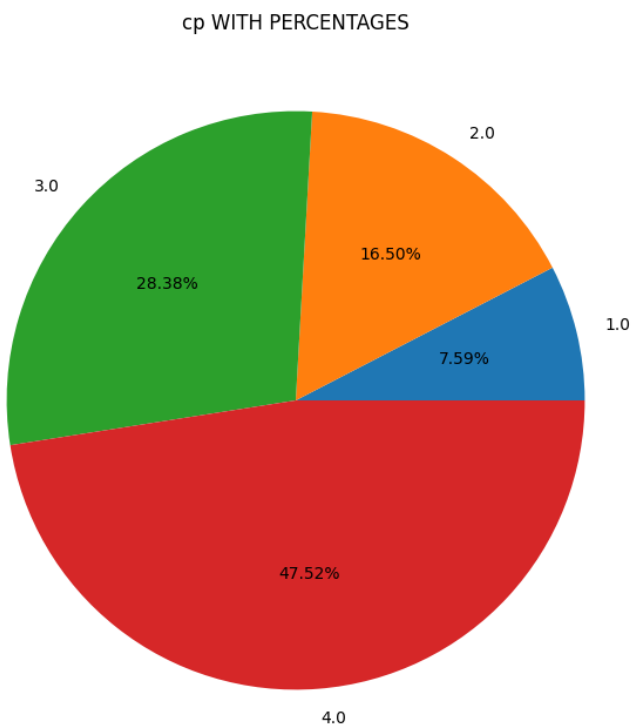
- In the data, missing values were marked as ‘?’. Replaced them with the mode value of their respective columns, as most values were categorical data.

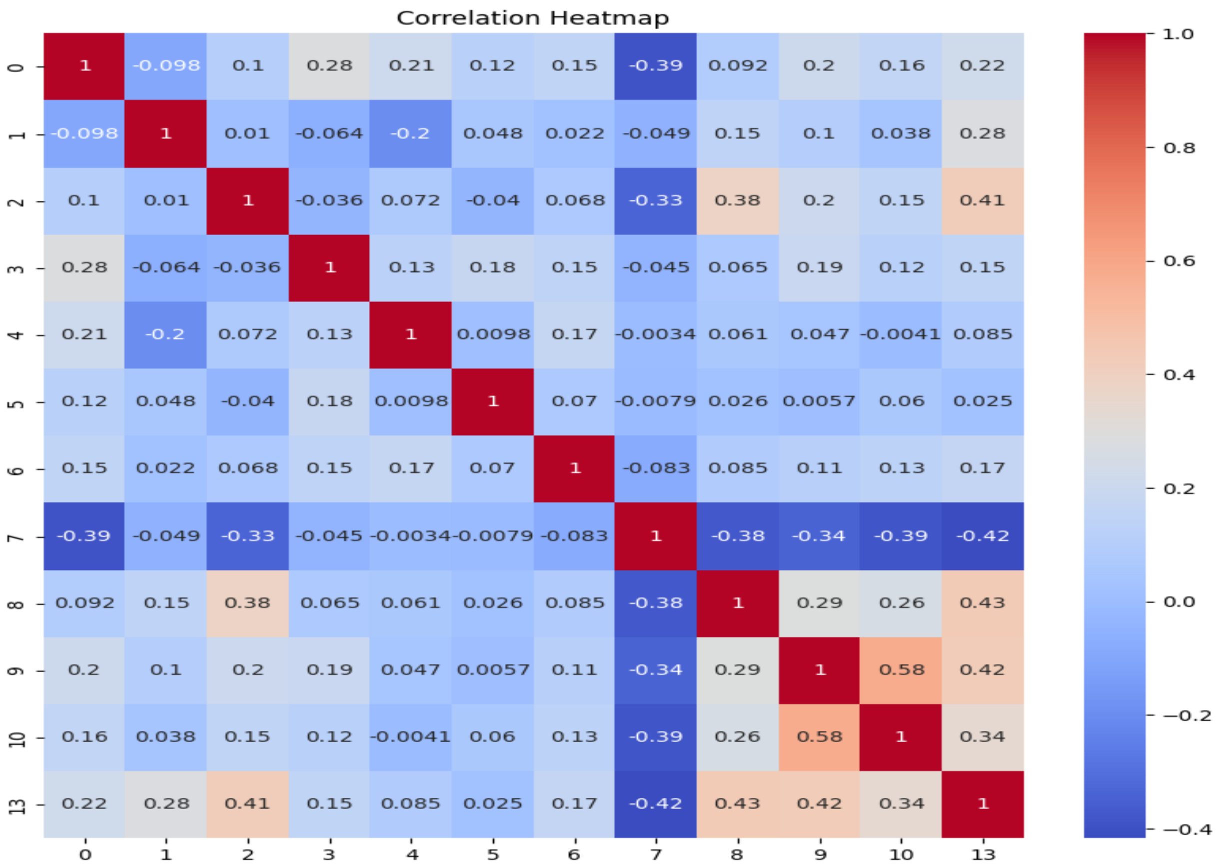
Columns: {0: 'age', 1: 'sex', 2: 'cp', 3: 'trestbps', 4: 'chol', 5: 'fbs', 6: 'restecg', 7: 'thalach', 8: 'exang', 9: 'oldpeak', 10: 'slope', 11: 'ca', 12: 'thal', 13: 'num'}

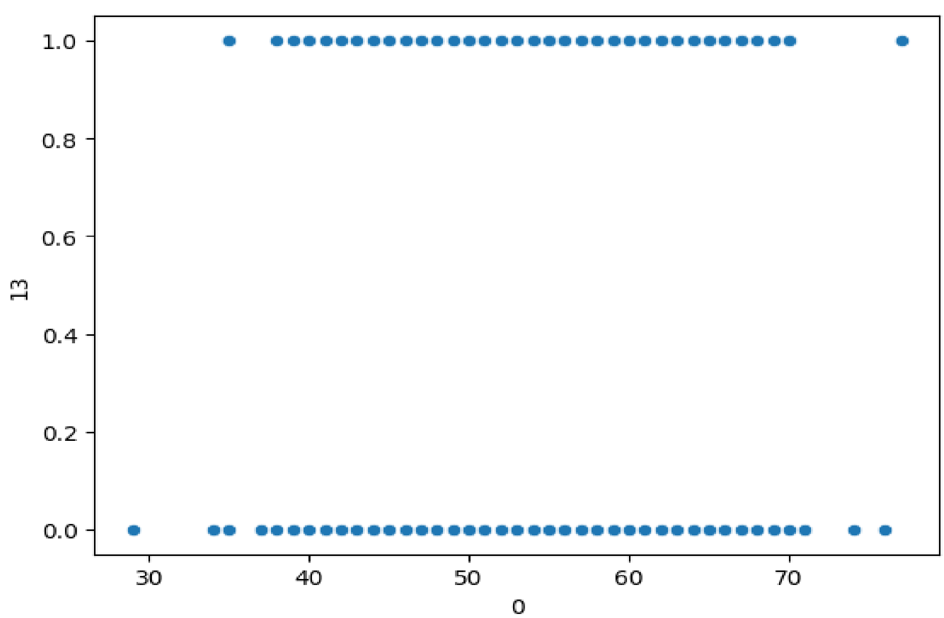
Where the target variable is column no. 13 or num.

EDA: I visualised the data by creating several graphs and plots. Scatter plot, heatmap, pairplots, piecharts. Some of these are:









X-axis: age. Y-axis: target.

Some insights are:

- From the correlation heatmap, we can see that most of the features are not correlated, except features no. 9 and 10. Also, target variable no. 13 somewhat correlates with features 2, 8, and 9.
- Made scatter plots and pair plots to find outliers. But there was not much of any outlier.
- On a surprising, from the scatter plot above, we can see that age does not much affect the outcome of heart disease.
- Data has an unequal distribution of sex. One gender data is almost double the other, but it is not specified.

Part B:

Split the data into training and testing in the ratio of 80:20.
Done using sklearn library:

```
x_train, x_test,y_train,y_test = train_test_split(X,Y,test_size = 0.2)
```

Part C:

Trained the decision tree using <u>entropy</u> as the splitting criterion.
Got the following accuracy:

```
#Entropy
Entropy = DecisionTreeClassifier(criterion='entropy')
Entropy.fit(x_train, y_train)
ypredEntropy = Entropy.predict(x_test)
accuracyEntropy = accuracy_score(y_test, ypredEntropy)
print(accuracyEntropy)

0.7049180327868853
```

When trained using ‘gini impurity’ as the splitting criterion, it got the following accuracy:

```
#Gini
Gini = DecisionTreeClassifier(criterion='gini')
Gini.fit(x_train, y_train)
ypredGini = Gini.predict(x_test)
accuracyGini = accuracy_score(y_test, ypredGini)
print(accuracyGini)
```

```
0.6557377049180327
```

So, based on the accuracy score, **'Entropy'** was the better criterion for node splitting.

Part D:

Performed hyperparameter search for the parameters min_samples_split and max_features using Grid search, with the help sklearn.

Values passed in the grid were:

```
'min_samples_split': [2, 3, 5, 7, 9, 10],
'max_features': ['sqrt', 'log2', None]
```

The criterion used for splitting was: 'Entropy' (from part c).
After running this and using the best parameters that came out were:

```
Hyperparameters Values:
min_samples_split:9
max_features:log2
Accuracy:0.7213114754098361
```

With an accuracy of 0.72.

Part E: Random Forests

Performed hyperparameter search for the parameters n_estimators, max_depth and min_samples_split using Grid search, with the help of sklearn.

Values passed in the grid were:

```
    'n_estimators': [50, 100, 150],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 3, 5, 7, 9, 10]
```

After running a grid search, the following were the results:

```
Hyperparameters Values:
n_estimators:50
max_depth:20
min_samples_split:9
Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.91      0.81        32
           1       0.86      0.62      0.72        29

    accuracy                           0.77        61
   macro avg       0.79      0.76      0.76        61
weighted avg       0.79      0.77      0.76        61


Accuracy:0.7213114754098361
```

This shows the combination of best hyperparameters, classification reports, and accuracy.