LAB ASSIGNMENT – 8

DECISION MAKING CLASSIFIERS MEASURES

NAME : PRATHAPANI SATWIKA

REG.NO. : 20BCD7160

Develop python functions for the following Decision Tree measures, Information Gain, Gain Ratio, and Gini Index, and attribute types, Categorical and Numerical.

Input: A data frame consists of Attribute and its Class Label

Output: Splitting Criteria, Data Partitions after splitting, and corresponding calculated measure values.

Utilize these functions to find out best splitting criteria for the following datasets: tennis.csv and iris.csv

CODE & OUTPUT :

```
[2] import pandas as pd
    import numpy as np
```

```
list_colors = ['blue']*3+['orange']*2+['green']*2
colors = pd.Series(list_colors)
print(colors)
```

```
0      blue
1      blue
2      blue
3    orange
4    orange
5     green
6     green
dtype: object
```

```
probs = colors.value_counts(normalize=True)
probs
```

```
blue      0.428571
orange    0.285714
green     0.285714
dtype: float64
```

```
[5] probs_by_hand = [3/7, 2/7, 2/7]
    print(probs_by_hand)
```

```
[0.42857142857142855, 0.2857142857142857, 0.2857142857142857]
```

[6]
```
entropy = -1*np.sum(np.log2(probs) * probs)
entropy
```

```
1.5566567074628228
```

[7]
```
gini_index = 1-np.sum(np.square(probs))
gini_index
```

```
0.653061224489796
```

[8]
```
from collections import Counter
import math
def entropy(labels):
  entropy=0
  label_counts = Counter(labels)
  print("Label counts: ",label_counts)
  print("====================")
  for label in label_counts:
    print("Label: ",label)
    prob_of_label = label_counts[label]/len(labels)
    print("Probability of ",label," is ",prob_of_label)
    entropy -= prob_of_label * math.log2(prob_of_label)
    print("Entropy of ",label, " is ",entropy)
    print("===================")
    return entropy
```

[12]
```
def information_gain(starting_labels, split_labels):
  info_gain = entropy(starting_labels)
  for branched_subset in split_labels:
    info_gain -= len(branched_subset) * entropy(branched_subset) / len(starting_labels)
    print("Information Gain of",split_labels,":",info_gain)
```

```
    print("==================================================")
    return info_gain
```

[13]
```python
def split_info_calculators():
    diff_labels=df_iris['Species'].value_counts()
    diff_labels = diff_labels/len(df_iris['Species'])
    split_info = -1 * np.sum(np.log2(diff_labels)*diff_labels)
    print("Split Information: ",split_info)
    return split_info
```

[14]
```python
def gini_impurity(y):
     if isinstance(y, pd.Series):
       p = y.value_counts()/y.shape[0]
       gini = 1-np.sum(p**2)
       return(gini)
```

```python
def gini_index_value(starting_labels, split_labels):
    gini_index = gini_impurity(starting_labels)
    for branched_subset in split_labels:
      gini_index += (len(branched_subset) / len(starting_labels))
      gini_impurity(branched_subset)
      print("Gini Index",gini_index)
      print("=========================")
      return 1-gini_index
```

[16]
```python
def split(dataset,column):
    split_data = []
    cols_vals = df_iris[column].unique()
```

```python
    for col_val in col_vals:
        split_data.append(dataset[dataset[column] == col_val])
    return(split_data)
```

```python
df_iris=pd.read_csv("Iris.csv")
df_iris
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

```python
print('We have {} features in our data'.format(len(df_iris.columns)-1))
```

```
We have 5 features in our data
```

```python
features = list(df_iris.columns)
features.remove('Species')
for feature in features:
  print("Feature: ",feature)
  probs = df_iris[feature].value_counts(normalize=True)
  print("Information Gain: ", (-1 * np.sum(np.log2(probs) * probs)))
  print("Gini Index: ",1 - np.sum(np.square(probs)))
  print("Gain Ratio: ", (-1 * np.sum(np.log2(probs)*probs))/(-1*np.sum(np.log2(probs))))
  print("=====================================")
```

```
Feature:  Id
Information Gain:  7.228818690495879
Gini Index:  0.9933333333333333
Gain Ratio:  0.006666666666666664
=====================================
Feature:  SepalLengthCm
Information Gain:  4.822018088381166
Gini Index:  0.96
Gain Ratio:  0.024982440721704365
=====================================
Feature:  SepalWidthCm
Information Gain:  4.0117097612189285
Gini Index:  0.9214222222222223
Gain Ratio:  0.03400441274909124
=====================================
Feature:  PetalLengthCm
Information Gain:  5.033829378702224
Gini Index:  0.9611555555555555
Gain Ratio:  0.020120292313574192
=====================================
Feature:  PetalWidthCm
Information Gain:  4.065662933799395
Gini Index:  0.9230222222222222
Gain Ratio:  0.03792750522263747
=====================================
```

```python
from collections import Counter
import math
def find_best_split(dataset):
  best_gain =0
  best_gain_ratio = 0
  best_gini = 0
  best_feature_gain = 0
  best_feature_gainratio = 0
  best_feature_gini = 0
  features = list(dataset.columns)
  features.remove('Species')
  for feature in features:
    split_data = split(dataset, feature)
    split_labels = [dataframe['Species'] for dataframe in split_data]
        gain = information_gain(dataset['Species'], split_labels)
        gain_ratio = gain / split_info_calculators()
        gini = gini_index_value(dataset['Species'], split_labels)
    if gain_ratio > best_gain_ratio:
      best_gain_ratio ,best_feature_gainratio = gain_ratio, feature
    if gain > best_gain:
      best_gain, best_feature_gain = gain, feature
    if gini > best_gini:
      best_gini, best_feature_gini = gini,feature
  print("Best Splitting Attribute from gain: ", best_feature_gain)
  print("Best Splitting Attribute from gain ratio: ",best_feature_gainratio)
  print("Best Splitting Attribute from gini index: ",best_feature_gini)
  print("Best Information gain: ",best_gain)
  print("Best Gain Ratio: ",best_gain_ratio)
  print("Best Gini Index: ",best_gini)
  return best_feature_gain,best_gain
```

Entropy of *,label, * is 0.0

================

Label counts: Counter({'Iris-virginica': 1})

===================

Label: Iris-virginica

Probability of Iris-virginica is 1.0

Entropy of *,label, * is 0.0

================

Label counts: Counter({'Iris-virginica': 1})

===================

Label: Iris-virginica

Probability of Iris-virginica is 1.0

Entropy of *,label, * is 0.0

================

Label counts: Counter({'Iris-virginica': 1})

===================

Label: Iris-virginica

Probability of Iris-virginica is 1.0

Entropy of *,label, * is 0.0

================

Label counts: Counter({'Iris-virginica': 1})

===================

Label: Iris-virginica

Probability of Iris-virginica is 1.0

Entropy of *,label, * is 0.0

================

Label counts: Counter({'Iris-virginica': 1})

===================

Label: Iris-virginica

Probability of Iris-virginica is 1.0

Entropy of *,label, * is 0.0

================

Label counts: Counter({'Iris-virginica': 1})

===================

Label: Iris-virginica

Probability of Iris-virginica is 1.0

Label counts: Counter({'Iris virginica': 1})

===================

Label: Iris-virginica
Probability of Iris
Entropy of *,label, * is 0.0
================
Label counts: Counter({'Iris-virginica': 1})
===================
Label: Iris-virginica

Probability of Iris-virginica is 1.0 Entropy of *,label, * is 0.0 ================
Label counts: Counter({'Iris
=================== Label: Iris
Probability of Iris
Entropy of *,label, * is 0.0

================

Label counts: Counter({'Iris-virginica': 1})

===================

Label: Iris-virginica

Label counts: Counter({'Iris-

===================

Label: Iris-virginica

Probability of Iris-

Entropy of *,label, * is 0.0

================

Label counts: Counter({'Iris-

===================

Label: Iris-virginica

Probability of Iris-virginica is 1.0

Entropy of *,label, * is 0.0

================

Label counts: Counter({'Iris-virginica': 1})

===================

Label: Iris-virginica

Probability of Iris-virginica is 1.0

Entropy of *,label, * is 0.0

================

Label counts: Counter({'Iris-virginica': 1})

===================

Label: Iris-virginica

Probability of Iris-virginica is 1.0

Entropy of *,label, * is 0.0

================

Label counts: Counter({'Iris-virginica': 1})

===================

Label: Iris-virginica

Probability of Iris-virginica is 1.0

Entropy of *,label, * is 0.0

================

Label counts: Counter({'Iris-virginica': 1})

===================

Label: Iris-virginica

Probability of Iris-virginica is 1.0

```
------------------------------------------------
Best Splitting Attribute from gain:  petal_length
Best Splitting Attribute from gain ratio:  petal_length
Best Splitting Attribute from gini index:  petal_length
Best Information gain:  1.4463165236457998
Best Gain Ratio:  0.9125241278501715
Best Gini Index:  0.2706666666666665
```

```
[    sepal_length  sepal_width  petal_length  petal_width      species
 0            5.1          3.5           1.4          0.2  Iris-setosa
 1            4.9          3.0           1.4          0.2  Iris-setosa
 4            5.0          3.6           1.4          0.2  Iris-setosa
 6            4.6          3.4           1.4          0.3  Iris-setosa
 8            4.4          2.9           1.4          0.2  Iris-setosa
 12           4.8          3.0           1.4          0.1  Iris-setosa
 17           5.1          3.5           1.4          0.3  Iris-setosa
 28           5.2          3.4           1.4          0.2  Iris-setosa
 33           5.5          4.2           1.4          0.2  Iris-setosa
 45           4.8          3.0           1.4          0.3  Iris-setosa
 47           4.6          3.2           1.4          0.2  Iris-setosa
 49           5.0          3.3           1.4          0.2  Iris-setosa,
     sepal_length  sepal_width  petal_length  petal_width      species
 2            4.7          3.2           1.3          0.2  Iris-setosa
 16           5.4          3.9           1.3          0.4  Iris-setosa
 36           5.5          3.5           1.3          0.2  Iris-setosa
 38           4.4          3.0           1.3          0.2  Iris-setosa
 40           5.0          3.5           1.3          0.3  Iris-setosa
 41           4.5          2.3           1.3          0.3  Iris-setosa
 42           4.4          3.2           1.3          0.2  Iris-setosa,
     sepal_length  sepal_width  petal_length  petal_width      species
 3            4.6          3.1           1.5          0.2  Iris-setosa
 7            5.0          3.4           1.5          0.2  Iris-setosa
 9            4.9          3.1           1.5          0.1  Iris-setosa
 10           5.4          3.7           1.5          0.2  Iris-setosa
 15           5.7          4.4           1.5          0.4  Iris-setosa
 19           5.1          3.8           1.5          0.3  Iris-setosa
 21           5.1          3.7           1.5          0.4  Iris-setosa
 27           5.2          3.5           1.5          0.2  Iris-setosa
 31           5.4          3.4           1.5          0.4  Iris-setosa
 32           5.2          4.1           1.5          0.1  Iris-setosa
 34           4.9          3.1           1.5          0.1  Iris-setosa
 37           4.9          3.1           1.5          0.1  Iris-setosa
 39           5.1          3.4           1.5          0.2  Iris-setosa
 48           5.3          3.7           1.5          0.2  Iris-setosa,
     sepal_length  sepal_width  petal_length  petal_width      species
 5            5.4          3.9           1.7          0.4  Iris-setosa
 18           5.7          3.8           1.7          0.3  Iris-setosa
 20           5.4          3.4           1.7          0.2  Iris-setosa
 23           5.1          3.3           1.7          0.5  Iris-setosa,
     sepal_length  sepal_width  petal_length  petal_width      species
 11           4.8          3.4           1.6          0.2  Iris-setosa
 25           5.0          3.0           1.6          0.2  Iris-setosa
 26           5.0          3.4           1.6          0.4  Iris-setosa
```

```python
df_tennis = pd.read_csv("tennis.csv")
df_tennis
```

	day	outlook	temp	humidity	wind	play
0	D1	Sunny	Hot	High	Weak	No
1	D2	Sunny	Hot	High	Strong	No
2	D3	Overcast	Hot	High	Weak	Yes
3	D4	Rain	Mild	High	Weak	Yes
4	D5	Rain	Cool	Normal	Weak	Yes
5	D6	Rain	Cool	Normal	Strong	No
6	D7	Overcast	Cool	Normal	Strong	Yes
7	D8	Sunny	Mild	High	Weak	No
8	D9	Sunny	Cool	Normal	Weak	Yes
9	D10	Rain	Mild	Normal	Weak	Yes
10	D11	Sunny	Mild	Normal	Strong	Yes
11	D12	Overcast	Mild	High	Strong	Yes
12	D13	Overcast	Hot	Normal	Weak	Yes
13	D14	Rain	Mild	High	Strong	No

```python
features = list(df_tennis.columns)
features.remove('play')
for feature in features:
  print("Feature: ",feature)
  probs = df_tennis[feature].value_counts(normalize=True)
  print("Information Gain: ",(-1 * np.sum(np.log2(probs)*probs)))
  print("Gini Index: ",1 - np.sum(np.square(probs)))
  print("Gini Ratio: ",(-1*np.sum(np.log2(probs)*probs))/(-1*np.sum(np.log2(probs))))
  print("======================================")
```

```
Feature:  day
Information Gain:  3.8073549220576055
Gini Index:  0.9285714285714286
Gini Ratio:  0.07142857142857145
======================================
Feature:  outlook
Information Gain:  1.5774062828523454
Gini Index:  0.6632653061224489
Gini Ratio:  0.33012503695295503
======================================
Feature:  temp
Information Gain:  1.5566567074628228
Gini Index:  0.653061224489796
Gini Ratio:  0.32181595964613585
======================================
Feature:  humidity
Information Gain:  1.0
Gini Index:  0.5
Gini Ratio:  0.5
======================================
Feature:  wind
Information Gain:  0.9852281360342515
Gini Index:  0.4897959183673470
Gini Ratio:  0.48539447002640107
======================================
```

```
from collections import Counter
import math
def find_best_split(dataset):
  best_gain = 0
  best_gain_ratio = 0
  best_gini = 0
  best_feature_gain = 0
  best_feature_gainratio = 0
  best_feature_gini = 0
  features = list(dataset.columns)
  features.remove('play')
  for feature in features:
    split_data = split(dataset,feature)
    split_labels = [dataframe['play'] for dataframe in split_data]
    gain = information_gain(dataset['play'], split_labels)
    gain_ratio = gain / split_info_calculators()
    gini = gini_index_value(dataset['play'], split_labels)
    if gain_ratio > best_gain_ratio:
      best_gain_ratio ,best_feature_gainratio = gain_ratio, feature
    if gain > best_gain:
      best_gain, best_feature_gain = gain, feature
    if gini > best_gini:
      best_gini, best_feature_gini = gini, feature

  print("Best Splitting Attribute from gain: ", best_feature_gain)
  print("Best Splitting Attribute from gain ratio: ",best_feature_gainratio)
  print("Best Splitting Attribute from gini index: ",best_feature_gini)
  print("Best Information gain: ",best_gain)
  print("Best Gain Ratio: ",best_gain_ratio)
  print("Best Gini Index: ",best_gini)
  return best_feature_gain, best_gain
```

```
new_data = split(df_tennis, find_best_split(df_tennis)[0])
```

```
Label counts:  Counter({'yes': 9, 'no': 5})
==============================================
Label:  no
Probability of  no  is  0.35714285714285715
Entropy of  no  is  0.5305095811322292
==============================================
Label:  yes
Probability of  yes  is  0.6428571428571429
Entropy of  yes  is  0.9402859586706311
==============================================
```

```
==================================================
Best Splitting Attribute from gain:  outlook
Best Splitting Attribute from gain ratio:  outlook
Best Splitting Attribute from gini index:  outlook
Best Information gain:  0.24674981977443927
Best Gain Ratio:  0.2624199771347136
Best Gini Index:  0.19795918367346932
```

```
new_data
```

```
[   outlook  temp humidity  windy play
 0    sunny   hot     high  False   no
 1    sunny   hot     high   True   no
 7    sunny  mild     high  False   no
 8    sunny  cool   normal  False  yes
 10   sunny  mild   normal   True  yes,
     outlook  temp humidity  windy play
 2   overcast   hot     high  False  yes
 6   overcast  cool   normal   True  yes
 11  overcast  mild     high   True  yes
 12  overcast   hot   normal  False  yes,
   outlook  temp humidity  windy play
 3    rainy  mild     high  False  yes
 4    rainy  cool   normal  False  yes
 5    rainy  cool   normal   True   no
 9    rainy  mild   normal  False  yes
 13   rainy  mild     high   True   no]
```