Gudi Varaprasad

19BCE7048

CSE3008 – Introduction to Machine Learning

Lab Slot – L1 + L2 + L11 + L12 + L43 + L44

LAB 3: ID3 ALGORITHM (Decision Trees)

Code: (PlayTennis Dataset)

```
import pandas as pd
df_tennis = pd.read_csv('/content/sample_data/PlayTennis.csv')
def entropy(probs):
 import math
 return sum( [-prob*math.log(prob, 2) for prob in probs] )
def entropy_of_list(a_list):

 from collections import Counter
 cnt = Counter(x for x in a_list)
 num_instances = len(a_list)*1.0
 print("\n Number of Instances of the Current Sub Class is {0}:".format(nu
m_instances ))
 probs = [x / num_instances for x in cnt.values()]
 print("\n Classes:",min(cnt),max(cnt))
 print(" \n Probabilities of Class {0} is {1}:".format(min(cnt),min(probs)
))
 print(" \n Probabilities of Class {0} is {1}:".format(max(cnt),max(probs)
))
 return entropy(probs)
print("\n INPUT DATA SET FOR ENTROPY CALCULATION:\n",
df_tennis['PlayTennis'])
total_entropy = entropy_of_list(df_tennis['PlayTennis'])
print("\n Total Entropy of PlayTennis Data Set:",total_entropy)


def information_gain(df,split_attr_name,target_attr_name,trace=0):
 print("Information Gain Calculation of ",split_attr_name)
 df_split = df.groupby(split_attr_name)

 nobs = len(df.index) * 1.0
 df_agg_ent = df_split.agg({target_attr_name : [entropy_of_list,lambda x:
len(x)/nobs] })[target_attr_name]
```

```
 df_agg_ent.columns = ['Entropy', 'PropObservations']

 new_entropy = sum( df_agg_ent['Entropy'] *
df_agg_ent['PropObservations'] )
 old_entropy = entropy_of_list(df[target_attr_name])
 return old_entropy - new_entropy
print('Info-
gain for Outlook is :'+str( information_gain(df_tennis,'Outlook', 'PlayTen
nis')),"\n")
print('\n Info-gain for Humidity is: ' + str(
information_gain(df_tennis, 'Humidity', 'PlayTennis')),"\n")
print('\n Info-
gain for Wind is:' + str( information_gain(df_tennis,'Wind', 'PlayTennis')
),"\n")
print('\n Info-
gain for Temperature is:' + str(information_gain(df_tennis, 'Temperature',
'PlayTennis')),"\n")

def id3(df, target_attr_name,  attr_names, default_class=None):
  from collections import Counter
  cnt = Counter(x for x in df[target_attr_name])
  if len(cnt) == 1:
    return next(iter(cnt))
  elif df.empty or (not  attr_names):
    return default_class
  else:
    default_class = max(cnt.keys())
    gainz = [information_gain(df, attr, target_attr_name) for attr in  att
r_names]
    index_of_max = gainz.index(max(gainz))
    best_attr =  attr_names[index_of_max]
    tree = {best_attr:{}}
    remaining_attr_names = [i for i in  attr_names if i != best_attr]

    for attr_val, data_subset in df.groupby(best_attr):
      subtree = id3(data_subset,target_attr_name,remaining_attr_names,defa
ult_class)
      tree[best_attr][attr_val] = subtree
      return tree
attr_names = list(df_tennis.columns)
print("List of  attrs:",  attr_names)
attr_names.remove('PlayTennis')
print("Predicting  attrs:",  attr_names)
from pprint import pprint
tree = id3(df_tennis,'PlayTennis', attr_names)
print("\n\nThe Resultant Decision Tree is :\n")
pprint(tree)
attr = next(iter(tree))
print("Best attr :\n", attr)
print("Tree Keys:\n",tree[ attr].keys())
```

```
def classify(instance, tree, default=None):
 attr = next(iter(tree))
 print("Key:",tree.keys())
 print(" attr:", attr)
 if instance[ attr] in tree[ attr].keys():
  result =tree[attr][instance[attr]]
  print("Instance attr:",instance[attr],"TreeKeys :",tree[attr].keys())
  if isinstance(result, dict):
   return classify(instance,result)
  else:
   return result
 else:
  return default
df_tennis['predicted']= df_tennis.apply(classify,axis=1,args=(tree,'No') )

print(df_tennis['predicted'])
print('\n Accuracy is:\n' + str(
sum(df_tennis['PlayTennis']==df_tennis['predicted'] ) /
(1.0*len(df_tennis.index)) ))
df_tennis[['PlayTennis', 'predicted']]
```

Output: (PlayTennis Dataset)

```
INPUT DATA SET FOR ENTROPY CALCULATION:
0      No
1      No
2     Yes
3     Yes
4     Yes
5      No
6     Yes
7      No
8     Yes
9     Yes
10    Yes
11    Yes
12    Yes
13     No
Name: PlayTennis, dtype: object

 Number of Instances of the Current Sub Class is 14.0:

 Classes: No Yes

 Probabilities of Class No is 0.35714285714285715:

 Probabilities of Class Yes is 0.6428571428571429:

 Total Entropy of PlayTennis Data Set: 0.9402859586706309
Information Gain Calculation of  Outlook

 Number of Instances of the Current Sub Class is 4.0:
```

Classes: Yes Yes

Probabilities of Class Yes is 1.0:

Probabilities of Class Yes is 1.0:

Number of Instances of the Current Sub Class is 5.0:

Classes: No Yes

Probabilities of Class No is 0.4:

Probabilities of Class Yes is 0.6:

Number of Instances of the Current Sub Class is 5.0:

Classes: No Yes

Probabilities of Class No is 0.4:

Probabilities of Class Yes is 0.6:

Number of Instances of the Current Sub Class is 14.0:

Classes: No Yes

Probabilities of Class No is 0.35714285714285715:

Probabilities of Class Yes is 0.6428571428571429:

Info-gain for Outlook is :0.2467498197744391

Information Gain Calculation of Humidity

Number of Instances of the Current Sub Class is 7.0:

Classes: No Yes

Probabilities of Class No is 0.42857142857142855:

Probabilities of Class Yes is 0.5714285714285714:

Number of Instances of the Current Sub Class is 7.0:

Classes: No Yes

Probabilities of Class No is 0.14285714285714285:

Probabilities of Class Yes is 0.8571428571428571:

Number of Instances of the Current Sub Class is 14.0:

Classes: No Yes

Probabilities of Class No is 0.35714285714285715:

Probabilities of Class Yes is 0.6428571428571429:

Info-gain for Humidity is: 0.15183550136234136

Information Gain Calculation of Wind

Number of Instances of the Current Sub Class is 6.0:

Classes: No Yes

Probabilities of Class No is 0.5:

Probabilities of Class Yes is 0.5:

Number of Instances of the Current Sub Class is 8.0:

Classes: No Yes

Probabilities of Class No is 0.25:

Probabilities of Class Yes is 0.75:

Number of Instances of the Current Sub Class is 14.0:

Classes: No Yes

Probabilities of Class No is 0.35714285714285715:

Probabilities of Class Yes is 0.6428571428571429:

Info-gain for Wind is:0.04812703040826927

Information Gain Calculation of Temperature

Number of Instances of the Current Sub Class is 4.0:

Classes: No Yes

Probabilities of Class No is 0.25:

Probabilities of Class Yes is 0.75:

Number of Instances of the Current Sub Class is 4.0:

Classes: No Yes

Probabilities of Class No is 0.5:

Probabilities of Class Yes is 0.5:

Number of Instances of the Current Sub Class is 6.0:

Classes: No Yes

Probabilities of Class No is 0.3333333333333333:

Probabilities of Class Yes is 0.6666666666666666:

Number of Instances of the Current Sub Class is 14.0:

Classes: No Yes

Probabilities of Class No is 0.35714285714285715:

Probabilities of Class Yes is 0.6428571428571429:

Info-gain for Temperature is:0.029222565658954647

List of attrs: ['Outlook', 'Temperature', 'Humidity', 'Wind', 'PlayTennis']
Predicting attrs: ['Outlook', 'Temperature', 'Humidity', 'Wind']
Information Gain Calculation of Outlook

Number of Instances of the Current Sub Class is 4.0:

Classes: Yes Yes

Probabilities of Class Yes is 1.0:

Probabilities of Class Yes is 1.0:

Number of Instances of the Current Sub Class is 5.0:

Classes: No Yes

Probabilities of Class No is 0.4:

Probabilities of Class Yes is 0.6:

Number of Instances of the Current Sub Class is 5.0:

Classes: No Yes

Probabilities of Class No is 0.4:

Probabilities of Class Yes is 0.6:

Number of Instances of the Current Sub Class is 14.0:

Classes: No Yes

Probabilities of Class No is 0.35714285714285715:

Probabilities of Class Yes is 0.6428571428571429:
Information Gain Calculation of Temperature

Number of Instances of the Current Sub Class is 4.0:

Classes: No Yes

Probabilities of Class No is 0.25:

Probabilities of Class Yes is 0.75:

Number of Instances of the Current Sub Class is 4.0:

Classes: No Yes

Probabilities of Class No is 0.5:

Probabilities of Class Yes is 0.5:

Number of Instances of the Current Sub Class is 6.0:

Classes: No Yes

Probabilities of Class No is 0.3333333333333333:

Probabilities of Class Yes is 0.6666666666666666:

Number of Instances of the Current Sub Class is 14.0:

Classes: No Yes

Probabilities of Class No is 0.35714285714285715:

Probabilities of Class Yes is 0.6428571428571429:

Information Gain Calculation of Humidity

Number of Instances of the Current Sub Class is 7.0:

Classes: No Yes

Probabilities of Class No is 0.42857142857142855:

Probabilities of Class Yes is 0.5714285714285714:

Number of Instances of the Current Sub Class is 7.0:

Classes: No Yes

Probabilities of Class No is 0.14285714285714285:

Probabilities of Class Yes is 0.8571428571428571:

Number of Instances of the Current Sub Class is 14.0:

Classes: No Yes

Probabilities of Class No is 0.35714285714285715:

Probabilities of Class Yes is 0.6428571428571429:

Information Gain Calculation of Wind

Number of Instances of the Current Sub Class is 6.0:

Classes: No Yes

Probabilities of Class No is 0.5:

Probabilities of Class Yes is 0.5:

Number of Instances of the Current Sub Class is 8.0:

```
 Classes: No Yes

 Probabilities of Class No is 0.25:

 Probabilities of Class Yes is 0.75:

 Number of Instances of the Current Sub Class is 14.0:

 Classes: No Yes

 Probabilities of Class No is 0.35714285714285715:

 Probabilities of Class Yes is 0.6428571428571429:


The Resultant Decision Tree is :

{'Outlook': {'Overcast': 'Yes'}}
Best attr :
 Outlook
Tree Keys:
 dict_keys(['Overcast'])
Key: dict_keys(['Outlook'])
 attr: Outlook
Key: dict_keys(['Outlook'])
 attr: Outlook
Key: dict_keys(['Outlook'])
 attr: Outlook
Instance attr: Overcast TreeKeys : dict_keys(['Overcast'])
Key: dict_keys(['Outlook'])
 attr: Outlook
Key: dict_keys(['Outlook'])
 attr: Outlook
Key: dict_keys(['Outlook'])
 attr: Outlook
Key: dict_keys(['Outlook'])
 attr: Outlook
Instance attr: Overcast TreeKeys : dict_keys(['Overcast'])
Key: dict_keys(['Outlook'])
 attr: Outlook
Key: dict_keys(['Outlook'])
 attr: Outlook
Key: dict_keys(['Outlook'])
 attr: Outlook
Key: dict_keys(['Outlook'])
 attr: Outlook
Key: dict_keys(['Outlook'])
 attr: Outlook
Instance attr: Overcast TreeKeys : dict_keys(['Overcast'])
Key: dict_keys(['Outlook'])
 attr: Outlook
Instance attr: Overcast TreeKeys : dict_keys(['Overcast'])
Key: dict_keys(['Outlook'])
 attr: Outlook
0      No
1      No
```

```
2      Yes
3       No
4       No
5       No
6      Yes
7       No
8       No
9       No
10      No
11     Yes
12     Yes
13      No
Name: predicted, dtype: object

 Accuracy is:
0.6428571428571429
```

	PlayTennis	predicted
0	No	No
1	No	No
2	Yes	Yes
3	Yes	No
4	Yes	No
5	No	No
6	Yes	Yes
7	No	No
8	Yes	No
9	Yes	No
10	Yes	No
11	Yes	Yes
12	Yes	Yes
13	No	No

Accuracy for Play Tennis dataset is: 64.285%

Now, given a test dataset, based on the decision tree obtained from above classification, we need to predict the output for following PlayTennisTest Dataset.

Output: (PlayTennisTest Dataset)

Outlook	Temperatu	Humidity	Wind	PlayTennis
Sunny	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Decision Tree: