classification-toy-example-2

March 5, 2024

Ratchanon Tarawan 65070503464

1 Lab 3: Introducing Classification

Objectives: - To gain hands-on experience classifying small dataset - To implement concepts related to Decision Tree classifier (i.e. Entropy, Information Gain), along with the Decision Tree algorithm

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

# Read the data
df = pd.read_csv('toy_data.csv')
df
```

[2]:

	age	income	student	credit rating	buys computer
0	<=30	high	no	fair	no
1	<=30	high	no	excellent	no
2	31-40	high	no	fair	yes
3	>40	medium	no	fair	yes
4	>40	low	yes	fair	yes
5	>40	low	yes	excellent	no
6	31-40	low	yes	excellent	yes
7	<=30	medium	no	fair	no
8	<=30	low	yes	fair	yes
9	>40	medium	yes	fair	yes
10	<=30	medium	yes	excellent	yes
11	31-40	medium	no	excellent	yes
12	31-40	high	yes	fair	yes
13	>40	medium	no	excellent	no

[3]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   age            14 non-null     object
```

```
 1   income          14 non-null     object
 2   student         14 non-null     object
 3   credit rating   14 non-null     object
 4   buys computer   14 non-null     object
dtypes: object(5)
memory usage: 688.0+ bytes
None
```

[4]:
```
target = df['buys computer'].value_counts()
income = df['income'].value_counts()
tar_yes = target.yes
tar_no = target.no
tar_all = tar_yes + tar_no
```

[5]:
```
Target = -(tar_yes/tar_all)*np.log2(tar_yes/tar_all) - (tar_no/tar_all)*np.
 ↪log2(tar_no/tar_all)
print(Target)
```

```
0.9402859586706311
```

[7]:
```
income = df['income'].value_counts()
in_high = income.high
in_medium = income.medium
in_low = income.low

high_count = df[df['income'] == 'high']['buys computer'].value_counts()
medium_count = df[df['income'] == 'medium']['buys computer'].value_counts()
low_count = df[df['income'] == 'low']['buys computer'].value_counts()
#print(high_count)
```

[8]:
```
entro_income_high = -(high_count.yes/in_high)*np.log2(high_count.yes/in_high) -␣
 ↪(high_count.no/in_high)*np.log2(high_count.no/in_high)
entro_income_medium = -(medium_count.yes/in_medium)*np.log2(medium_count.yes/
 ↪in_medium) - (medium_count.no/in_medium)*np.log2(medium_count.no/in_medium)
entro_income_low = -(low_count.yes/in_low)*np.log2(low_count.yes/in_low) -␣
 ↪(low_count.no/in_low)*np.log2(low_count.no/in_low)
```

[9]:
```
entro_income = (in_high/tar_all)*entro_income_high + (in_medium/
 ↪tar_all)*entro_income_medium + (in_low/tar_all)*entro_income_low
print(entro_income)
```

```
0.9110633930116763
```

[11]:
```
gain = Target - entro_income
```

[12]:
```
print('Gain (Income)' , gain)
```

```
Gain (Income) 0.02922256565895487
```