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

```
# Run this cell if you use Colab
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd

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

	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

Next steps: View recommended plots

```
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
```

```
#1. Calculate Income Attribute

#65070503425 Pamika Lertsrisatit
#65070503428 Pitchayapat Wareevanich
#65070503442 Intouch Krajangprateep
#65070503465 Varod Tatiyatidsana

import numpy as np
import pandas as pd
```

```
#the formula for finding entropy
def entropy(p):
    return -p * np.log2(p) - (1-p) * np.log2(1-p)

#find information gain
def information_gain(parent, splits):
    parent_entropy = entropy(parent['buys computer'].value_counts(normalize=True).values[0])
    weighted_child_entropy = 0

    for split in splits:
        split_entropy = entropy(split['buys computer'].value_counts(normalize=True).values[0])
        weight = len(split) / len(parent)
        weighted_child_entropy = weighted_child_entropy + (weight*split_entropy)

        result=parent_entropy - weighted_child_entropy
    return result

parentn = df
child = [df[df['income']==value] for value in df['income'].unique()]

#print the result
gainsplit = information_gain(parentn, child)
print("Information Gain (Income):", gainsplit)
```

```
     Information Gain (Income): 0.02922256565895487
```