## Program Code: J620-002-4:2020

## Program Name: FRONT-END SOFTWARE DEVELOPMENT

## Title : Exe21 - Decision Tree and Random Forest Exercise

**Name: Chuay Xiang Ze**

**IC Number: 021224070255**

**Date : 20/07/2023**

**Introduction : Learning how to use both Decision Tree and Random Forest**

**Conclusion : Managed to complete tasks using both classifiers.**

# Machine Learning and NLP Exercises

# Introduction

We will be using the same review data set from Kaggle for this exercise. The product we'll focus on this time is a cappuccino cup. The goal of this week is to not only preprocess the data, but to classify reviews as positive or negative based on the review text.

The following code will help you load in the data.

In [32]:

```
import nltk
import pandas as pd
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
from nltk.corpus import stopwords
import seaborn as sns
%matplotlib inline
```

In [33]:

```
data = pd.read_csv('./coffee.csv')
data.head()
```

Out[33]:

| | **user_id** | **stars** | **reviews** |
|---|---|---|---|
| **0** | A2XP9IN4JOMROD | 1 | I wanted to love this. I was even prepared for... |
| **1** | A2TS09JCXNV1VD | 5 | Grove Square Cappuccino Cups were excellent. T... |
| **2** | AJ3L5J7GN09SV | 2 | I bought the Grove Square hazelnut cappuccino ... |
| **3** | A3CZD34ZTUJME7 | 1 | I love my Keurig, and I love most of the Keuri... |
| **4** | AWKN396SHAQGP | 1 | It's a powdered drink. No filter in k-cup.<br ... |

In [34]:

```
data.columns
```

Out[34]:

```
Index(['user_id', 'stars', 'reviews'], dtype='object')
```

# Question 1

- Determine how many reviews there are in total.

Use the preprocessing code below to clean the reviews data before moving on to modeling.

In [35]:

```
# Text preprocessing steps - remove numbers, captial letters and punctuation
import re
import string

alphanumeric = lambda x: re.sub(r"""\w*\d\w*""", ' ', x)
punc_lower = lambda x: re.sub('[%s]' % re.escape(string.punctuation), ' ', x.lower())

data['reviews'] = data.reviews.map(alphanumeric).map(punc_lower)
data
```

Out[35]:

| | user_id | stars | reviews |
|---|---|---|---|
| **0** | A2XP9IN4JOMROD | 1 | i wanted to love this i was even prepared for... |
| **1** | A2TS09JCXNV1VD | 5 | grove square cappuccino cups were excellent t... |
| **2** | AJ3L5J7GN09SV | 2 | i bought the grove square hazelnut cappuccino ... |
| **3** | A3CZD34ZTUJME7 | 1 | i love my keurig and i love most of the keuri... |
| **4** | AWKN396SHAQGP | 1 | it s a powdered drink no filter in k cup br ... |
| **...** | ... | ... | ... |
| **537** | A398T38COTS30K | 5 | this is my favorite k cup flavor i like my c... |
| **538** | A1B410YK9O18XZ | 5 | if you are looking for the taste of french van... |
| **539** | A1W85A81467TCW | 5 | i have purchased and used boxes of the hazel... |
| **540** | A103FOM06QPAX8 | 5 | yummy great tasting and very convenient onl... |
| **541** | A1V5V04WIYLT8Q | 4 | for an enjoyable change from a coffee routine ... |

542 rows × 3 columns

In [36]:

```
len(data[data['stars'] == 5])
```

Out[36]:

```
308
```

# Question 2: Classsification *(20% testing, 80% training)*

Processes for classification

### Step 1: Prepare the data (identify the feature and label)

In [37]:

```
X = data['reviews']
y = data['stars']
```

### Step 2: Vectorize the feature

In [57]:

```
vectorizer = CountVectorizer()
X_vectorized = vectorizer.fit_transform(X)
```

### Step 3: Split the data into training and testing sets

In [58]:

```
X_train, X_test, y_train, y_test = train_test_split(X_vectorized, y, test_size=0.2, rand
```

### Step 4: Idenfity the model/ classifier to be used. Feed the train data into the model

### - Decision Tree

In [59]:

```
clf = DecisionTreeClassifier()

clf.fit(X_train,y_train)

y_pred_dt = clf.predict(X_test)
```

### - Random Forest

In [63]:

```
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
```

# Question 3

Generate the accuracy scores for Decision Tree and Random Forest.

In [64]:

```
accuracy_dt = accuracy_score(y_test, y_pred_dt)
print("Decision Tree Accuracy:", accuracy_dt)

# Calculate accuracy for Random Forest
accuracy_rf = accuracy_score(y_test, y_pred_rf)
print("Random Forest Accuracy:", accuracy_rf)
```

```
Decision Tree Accuracy: 0.5137614678899083
Random Forest Accuracy: 0.5779816513761468
```

# Question 4

Predict the rate of this review,

**"I dislike this coffee, terrible taste and very greasy."**

by using Decision Tree, Random Forest

In [65]:

```
new_review = "I dislike this coffee, terrible taste and very greasy."
new_review_vectorized = vectorizer.transform([new_review])
rate_dt = clf.predict(new_review_vectorized)[0]
rate_rf = rf.predict(new_review_vectorized)[0]

print("Decision Tree predicted rate:", rate_dt)
print("Random Forest predicted rate:", rate_rf)
```

```
Decision Tree predicted rate: 5
Random Forest predicted rate: 5
```

In [ ]: