Forward
School

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

## Program Name: FRONT-END SOFTWARE DEVELOPMENT

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

#### Name: Ooi Caaron

#### IC Number: 990701-07-5837

#### Date :21/7/23

#### Introduction : Decision Tree algorithm partitions the data into subsets by repeatedly asking questions about the features of the data points.

#### Conclusion : Still need to practice more and do revision

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 [1]:

```
import nltk
import pandas as pd
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:

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

Out[2]:

	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 3 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

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 [3]:

```
# 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.head()
```

Out[3]:

	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 [4]:

```
len(data)
```

Out[4]:

```
542
```

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

Processes for classification

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

In [5]:

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
X = data['reviews']
y = data['stars']
```

Step 2: Vectorize the feature

In [6]:

```
vectorizer = TfidfVectorizer(analyzer='word')
X = vectorizer.fit_transform(X)
print(X.shape)
```

```
(542, 2320)
```

Step 3: Split the data into training and testing sets

In [7]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42
print("Training set size:", X_train.shape)
print("Testing set size:", X_test.shape)
```

```
Training set size: (433, 2320)
Testing set size: (109, 2320)
```

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

- Decision Tree

In [8]:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
DT_Classifier = DecisionTreeClassifier()
DT_Classifier.fit(X_train, y_train)
DT_Classifiers = DT_Classifier.predict(X_test)
```

- Random Forest

In [9]:

```
from sklearn.ensemble import RandomForestClassifier
rf_classifier = RandomForestClassifier()
rf_classifier.fit(X_train, y_train)
rf_classifiers = rf_classifier.predict(X_test)
```

Question 3

Generate the accuracy scores for Decision Tree and Random Forest.

In [10]:

```
from sklearn.metrics import accuracy_score

accuracy_dt = accuracy_score(y_test, DT_Classifiers)
print('Decision Tree Accuracy:', accuracy_dt)

accuracy_rf = accuracy_score(y_test, rf_classifiers)
print('Decision Tree Accuracy:', accuracy_rf)
```

```
Decision Tree Accuracy: 0.48623853211009177
Decision Tree 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 [20]:

```
ts = "I dislike this coffee, terrible taste and very greasy."
# ts = re.sub(r"""\w*\d\w*""",'', ts)
# ts = re.sub('[%s]'%re.escape(string.punctuation),'',ts.lower())
# ts = [ts]
tsv = vectorizer.transform([ts])
rate_rf = rf_classifier.predict(tsv)[0]
rate_dt = DT_Classifier.predict(tsv)[0]
print(rate_rf)
print(rate_dt)
```

```
5
5
```

In [ ]:

```

```