Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Exe21 - Decision Tree and Random Forest Exercise

Name: Phua Yan Han

IC Number: 050824070059

Date : 18/7/2023

Introduction :

Conclusion :

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 [5]:
```
import nltk
import pandas as pd
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [6]:
```
data = pd.read_csv('../Data files/coffee.csv')
data.head()
```

Out[6]:

	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 ...

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 [7]:
```
# 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.lo

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

Out[7]:

	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 [8]:
```
len(data)
```

Out[8]:
```
542
```

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

Processes for classification

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

In [9]:
```
X = data.reviews
y = data.stars
```

Step 2: Vectorize the feature

In [10]:
```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

# vectorizer = CountVectorizer()
# Count is for the count of words

vectorizer = TfidfVectorizer()
# Tfid for state of the art for Natural Language Processor (NLP)

X = vectorizer.fit_transform(X)

print(X.shape)
```

```
(542, 2320)
```

Step 3: Split the data into training and testing sets

In [11]:
```
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn import metrics, tree
import numpy as np

np.random.seed(42)

X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size
```

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

- Decision Tree

In [12]:
```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

from matplotlib import rcParams
import warnings

warnings.filterwarnings("ignore")

rcParams["figure.figsize"] = 10, 6

clf = DecisionTreeClassifier()
clf = clf.fit(X_train,y_train)

clf.predict(X_test)
```

Out[12]:
```
array([5, 1, 5, 5, 5, 3, 5, 1, 5, 5, 4, 2, 5, 5, 5, 5, 1, 5, 5, 5, 4, 5,
       5, 5, 1, 5, 5, 3, 5, 5, 1, 5, 3, 5, 1, 1, 4, 2, 1, 5, 5, 4, 1, 1,
       5, 5, 4, 5, 5, 1, 5, 5, 5, 1, 5, 5, 5, 5, 5, 5, 5, 5, 1, 1, 1, 5,
       5, 5, 5, 2, 1, 5, 1, 1, 5, 5, 5, 1, 5, 5, 1, 5, 4, 5, 5, 5, 5, 4,
       5, 5, 1, 5, 5, 1, 5, 5, 5, 5, 1, 2, 5, 5, 5, 5, 5, 5, 1, 1, 5],
      dtype=int64)
```

- Random Forest

In [13]:
```
from sklearn.ensemble import RandomForestClassifier

clf_forest = RandomForestClassifier(n_estimators=100)
clf_forest.fit(X_train, y_train)

clf_forest.predict(X_test)
```

Out[13]:
```
array([5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 1, 5, 1, 5, 5, 5, 5, 5, 1, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 1, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 1, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 1, 5],
      dtype=int64)
```

Question 3

Generate the accuracy scores for Decision Tree and Random Forest.

In [19]:
```
from sklearn.metrics import accuracy_score

y_pred_tree = clf.predict(X_test)
y_pred_forest = clf_forest.predict(X_test)

as_tree = accuracy_score(y_test, y_pred_tree)
as_forest = accuracy_score(y_test, y_pred_forest)

print("Decision Tree accuracy score:", as_tree)
print("Random Forest accuracy score:", as_forest)
```

```
Decision Tree accuracy score: 0.5596330275229358
Random Forest accuracy score: 0.5963302752293578
```

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]:
```
review_text = "I dislike this coffee, terrible taste and very greasy."

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

review_text = re.sub(r"""\w*\d\w*""", ' ', review_text)
review_text = re.sub('[%s]' % re.escape(string.punctuation), ' ', review_text.l
review_text = vectorizer.transform([review_text])

print("Decision Tree prediction:", clf.predict(review_text)[0])
print("Random Forest prediction:", clf_forest.predict(review_text)[0])
```

```
Decision Tree prediction: 5
Random Forest prediction: 5
```