Kingdom of Saudi Arabia
Ministry of Education
University of Jeddah
College of Computer Science and Engineering
Department of Computer Science and Artificial Intelligence

المملكة العربية السعودية
وزارة التعليم
جامعة جدّة
كلية علوم وهندسة الحاسب
قسم علوم الحاسب والذكاء الاصطناعي

Lab 4
CCAI 312 Pattern Recognition
Third Trimester 2023

Student Name: Ruba Khalid Alsulami

Student ID: 2110618

		Max Score	Student Score
PLO S2 / CLO 2 / SO 2	Task 1	2	
PLO C4 / CLO 3 / SO 7	Task 2	2	
Total			

Task 1: [PLO S2 / CLO 2 / SO 2] [2 marks]

1. First import required libraries, load "**Hotel Reservations**" dataset into a data frame and use `Booking_ID` as index_col.

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import GridSearchCV

hotel_reservations = pd.read_csv('/kaggle/input/lab4-t/Hotel Reservations.csv', index_col='Booking_ID')
```

+ Code + Markdown

2. Explore dataset and answer the following questions:

   What is the dataset's shape? How many samples are there?How many features are there?

```
[4]: # get the shape
     hotel_reservations.shape
[4]: (36275, 18)

[5]: # get the number of samples
     len(hotel_reservations)
[5]: 36275

[6]: # get the number of features
     len(hotel_reservations.columns)
[6]: 18
```

   How many categorical features are there in the dataset? What is the distribution of the target label "booking_status"?

```
[7]: # Get the number of categorical features
     categorical_features = hotel_reservations.select_dtypes(include=['object']).columns
     len(categorical_features)
[7]: 4

[8]: #the distribution of the target label
     hotel_reservations['booking_status'].value_counts()
[8]: Not_Canceled    24390
     Canceled        11885
     Name: booking_status, dtype: int64
```

3. Split dataframe to two variables X (all features except booking_status) and y (booking_status).

```
[7]: X = hotel_reservations.iloc[:, :-1]
     y = hotel_reservations.loc[:, 'booking_status']
```

4. Split the dataset to train/test split.

```
[11]: #split the dataset to train/test split
      from sklearn.model_selection import train_test_split

      X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 100)
```

5. Fit and evaluate DecisionTree Classifier using train/test split (use `random_state=100`). What is the accuracy of the model?

```
# fit and evaluate DecisionTree Classifier using train/test split
from sklearn.preprocessing import OneHotEncoder
encoder = OneHotEncoder(handle_unknown='ignore')
X_train_encoded = encoder.fit_transform(X_train.select_dtypes(include='object'))
X_test_encoded = encoder.transform(X_test.select_dtypes(include='object'))
dtc = DecisionTreeClassifier(random_state=100)
dtc.fit(X_train_encoded, y_train)
y_pred = dtc.predict(X_test_encoded)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy of DecisionTree Classifier=",accuracy)
```

```
Accuracy of DecisionTree Classifier= 0.6804962095106822
```

+ Code + Markdown

6. Create a new variable X_numric that contain only numeric features.

```
[12]: X_numeric=hotel_reservations.select_dtypes(['number'])

      X_numeric.shape
```

```
[12]: (36275, 14)
```

7. Repeat steps 5 & 6 using X_numeric instead of X.

```
dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)
y_pred= dtc.predict(X_numeric)

print("Accuracy with numeric features=",accuracy_score(y_test, predictions))
```

```
Accuracy with numeric features= 0.8534568309626199
```

+ Code + Markdown

How accurate is the model? Is the accuracy good or bad?
The accuracy of the model with all features is lower than the accuracy of the model with only numeric features

8. Use GridSearchCV to find the best values for hyperparameter (see step 13)

```
from sklearn.model_selection import GridSearchCV

params = {
    'criterion':  ['gini', 'entropy'],
    'max_depth':  [None, 2, 4, 6, 8, 10],
    'max_features': [None, 'sqrt', 'log2', 0.2, 0.4, 0.6, 0.8],
    'splitter': ['best', 'random']
}

dtc = GridSearchCV(
    estimator=DecisionTreeClassifier(),
    param_grid=params,
    cv=5,
    n_jobs=5,
    verbose=1,
)
dtc.fit(X_train_encoded, y_train)
print(dtc.best_params_)
```

```
Fitting 5 folds for each of 168 candidates, totalling 840 fits
{'criterion': 'entropy', 'max_depth': 6, 'max_features': 0.6, 'splitter': 'random'}
```

9. What is the best values for 'criterion' , 'max_depth' , 'max_features', and 'splitter'.

'max_depth': 6, 'max_features': 0.6, 'splitter': 'random'

10. Fit and evaluate new DecisionTree Classifier using best hyperparameter values.

```
dtc = DecisionTreeClassifier(max_depth=6, criterion='entropy', max_features=0.6, splitter='best')

dtc.fit(X_train_encoded, y_train)
y_pred = dtc.predict(X_test_encoded)
print(accuracy_score(y_test,y_pred))
```

```
0.6804962095106822
```

+ Code + Markdown

11. How accurate is the new model? Does the model improve?

The new model does not improve on the original model

Task? 2: [PLO C4 / CLO 3 / SO 7] [2 mark]

1. Use OneHotEncoder to convert categorical features to numbers (see step 11)

```
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer

column_transformer = make_column_transformer((OneHotEncoder(), ['type_of_meal_plan', 'room_type_reserved','market_segment_type']),remainc

# First, we must transform the entire dataset
X_transformed = column_transformer.fit_transform(X)
X_transformed = pd.DataFrame(data=X_transformed)

#second,we divide the transformed dataset into train/test dataset
X_train, X_test, y_train, y_test = train_test_split(X_transformed, y, random_state = 100)
```

+ Code + Markdown

2. Fit and evaluate DecisionTree Classifier using all features.

```
dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)

predictions = dtc.predict(X_test)
print(accuracy_score(y_test, predictions))
```

```
0.8663579225934502
```

+ Code + Markdown

How accurate is the model? Does the model improve by using categorical features? The model has improved a bit