AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh

Assignment title: Airlines Booking
Course Name : Programming in python
Semester: Spring
Section: A
Teacher name: DR. Abdus Salam

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where suchcollaborationhas been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand thatPlagiarism is the presentation of the work, idea or creation of another person as though it is yourown. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copymy/our work.

* *Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.: 12

No	Name	ID	Program
1	HORISH DAS PRIYO	21-44816-1	CSE
2	MST.FARZANA RAHMAN	20-43123-1	CSE

Faculty use only

FACULTYCOMMENTS	**Marks Obtained**	

	Total Marks	

Project Description:

The final term project aimed to implement various machine learning techniques on a dataset using Python. The project required careful selection of a unique dataset, data cleaning, exploratory data analysis through graphical representations, feature scaling, splitting data into training and testing sets, applying the Naïve Bayes Classifier, evaluating model performance through confusion matrix, train-test accuracy comparison, and demonstrating the use of 10-fold cross-validation.

Dataset Selection:

Our group carefully selected a unique dataset for this project, ensuring it did not match with any other dataset chosen by classmates. The dataset chosen was [Airlines Booking], sourced from [Airlines_Booking.csv (kaggle.com)].

Task 1: Reading and Loading Dataset:
We utilized the Pandas library to read and load the dataset file into our program. This involved importing the necessary libraries and using the read_csv() function to load the dataset into a Pandas Data Frame.

Task 2: Data Cleaning:
Appropriate data cleaning techniques were applied to the dataset. We replaced bad data using proper methods and ensured no records were deleted except for duplicate records. Pandas library functionalities were employed for this task.

Task 3: Analyzing Frequency Distributions:
We utilized the Matplotlib library to draw graphs analyzing the frequency distributions of the features. All plots were drawn in a single figure using the subplot() function for easy comparison.

Task 4: Exploring Relationships:
Graphs were drawn to illustrate any relationship between the target column and other columns in the dataset. Matplotlib library was used, and the subplot() function was employed to display all plots in one figure.

Task 5: Feature Scaling:
Feature scaling was performed on the dataset. Prior data conversion was applied if necessary. Techniques such as normalization or standardization were employed to scale the features appropriately.

Task 6: Data Splitting:
The dataset was split into training and testing sets using the train_test_split() function. A random state parameter value of 321 was set for consistency in results.

Task 7: Naïve Bayes Classifier:
We applied the Naïve Bayes Classifier to the dataset and built the prediction model. Training of the model was performed in this step.

Task 8: Confusion Matrix:
The confusion matrix for the model was calculated to evaluate its performance. Detailed interpretation of the confusion matrix was provided in the report.

Task 9: Model Accuracy:
Train and test accuracies of the model were calculated and compared to assess its performance on both training and testing datasets.

Task 10: 10-Fold Cross-Validation:
We demonstrated how 10-fold cross-validation can be utilized to build a Naïve Bayes classifier. The accuracy of this model was reported as part of the evaluation.

Code:

Import necessary libraries:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler

FILE_PATH = '/content/customer_booking.csv'
```

Task 1: Read/Load the dataset file in your program. Use Pandas library to complete this task.

```
df=pd.read_csv(FILE_PATH,encoding="latin-1")
print(df.info())
```

Task 2: Apply appropriate data cleaning techniques to the dataset. In this step, replace bad data using proper methods and do not delete any record except duplicate records. Use Pandas library to complete this task.

```
print(df.info())
df = df.drop_duplicates()

print("Missing values summary:")
print(df.isnull().sum())
```

Task 3: Draw graphs to analyze the frequency distributions of the features. Use Matplotlib library to complete this task. Draw all the plots in a single figure so that all plots can be seen in one diagram (use subplot() function).

```
fig, axes = plt.subplots(4, 4, figsize=(15, 12))

feature_index = 0
for i in range(4):
    for j in range(4):
        if feature_index >= len(df.columns):
            break

        df[df.columns[feature_index]].value_counts().plot(kind='bar', ax=axes[i, j])
        axes[i, j].set_title(df.columns[feature_index])
        axes[i, j].set_xlabel(df.columns[feature_index])
        axes[i, j].set_ylabel('Frequency')
        feature_index += 1

plt.tight_layout()

plt.show()
```

Task 4: Draw graphs to illustrate if there is any relationship between target column to any other columns of the dataset. Use Matplotlib library to complete this task. Also use sublot() function to show all plots in one figure.

```
def plot_relationships(data, target_column, num_rows=4, num_cols=4, figsize=(15, 12)):
    """
    Creates subplots to visualize relationships between the target column and other features.

    Args:
        data (pandas.DataFrame): The cleaned DataFrame containing the data.
        target_column (str): The name of the target column.
        num_rows (int, optional): The number of rows for the subplots grid. Defaults to 4.
        num_cols (int, optional): The number of columns for the subplots grid. Defaults to 4.
```

```
     figsize (tuple, optional): The size of the figure. Defaults to (15, 12).
  """

  features = [col for col in data.columns if col != target_column]

  fig, axes = plt.subplots(num_rows, num_cols, figsize=figsize)

  feature_index = 0
  for i in range(num_rows):
     for j in range(num_cols):
        if feature_index >= len(features):
           break


        if data[features[feature_index]].dtype == object:
           data[features[feature_index]].value_counts().plot(kind='bar', ax=axes[i, j])
           axes[i, j].set_title(f"{features[feature_index]} (Categorical)")
           axes[i, j].set_xlabel(features[feature_index])
           axes[i, j].set_ylabel('Count')
        else:
           data.plot.scatter(x=features[feature_index], y=target_column, ax=axes[i, j])
           axes[i, j].set_title(f"{features[feature_index]} vs Target")
           axes[i, j].set_xlabel(features[feature_index])
           axes[i, j].set_ylabel('Target Column')

        feature_index += 1

  plt.tight_layout()

  plt.show()

plot_relationships(df, 'wants_extra_baggage')
```

Task 5: Perform scaling to the features of the dataset. Remember that you will need to apply data conversion before performing scaling if it is needed.

```
df=pd.read_csv(FILE_PATH,encoding="latin-1")
df['purchase_lead'] = df['purchase_lead'].fillna(df['purchase_lead'].mean())
df['length_of_stay'] = df['length_of_stay'].fillna(df['length_of_stay'].median())
df['flight_hour'] = df['flight_hour'].clip(lower=0, upper=23)
df['length_of_stay'] = df['length_of_stay'].clip(lower=0)
data = df.drop_duplicates()
numerical_features = [col for col in data.columns if data[col].dtype != 'object']
min_max_scaler = MinMaxScaler()
print("Before Scaling...")
print('data["flight_hour"].min(): ',data["flight_hour"].min())
print('data["flight_hour"].max(): ',data["flight_hour"].max())
data[numerical_features] = min_max_scaler.fit_transform(data[numerical_features])
print("After Scaling...")
print('data["flight_hour"].min()',data["flight_hour"].min())
print('data["flight_hour"].max()',data["flight_hour"].max())
```

Task 6: Split your data into two parts: Training dataset and Testing dataset. You must use the function train_test_split() to complete this task and use value 321 as the value of the `random_state` parameter of this function.

```
from sklearn.model_selection import train_test_split
```

```
target = 'wants_extra_baggage'

X = data.drop(target, axis=1)
y = data[target]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=321)

print("Training set size:", len(X_train))
print("Testing set size:", len(X_test))
```

Task 7: Apply Naïve Bayes Classifier to the dataset. Build (train) your prediction model in this step.

```
X_train_encoded = pd.get_dummies(X_train)
X_test_encoded = pd.get_dummies(X_test)

missing_cols = set(X_train_encoded.columns) - set(X_test_encoded.columns)
for c in missing_cols:
   X_test_encoded[c] = 0
X_test_encoded = X_test_encoded[X_train_encoded.columns]

gnb.fit(X_train_encoded, y_train)
```

Taks 8: Calculate the confusion matrix for your model. Interpret it in detail in the report.

```
from sklearn.metrics import confusion_matrix, classification_report

y_pred = gnb.predict(X_test_encoded)

cm = confusion_matrix(y_test, y_pred)

print("Confusion Matrix:")
print(cm)

report = classification_report(y_test, y_pred)
print("\nClassification Report:")
print(report)
```

Task 9: Calculate the train and test accuracy of your model and compare them.

```
from sklearn.metrics import accuracy_score

y_train_pred = gnb.predict(X_train_encoded)

train_accuracy = accuracy_score(y_train, y_train_pred)

test_accuracy = accuracy_score(y_test, y_pred)

print(f"Training Accuracy: {train_accuracy * 100:.2f}%")
print(f"Testing Accuracy: {test_accuracy * 100:.2f}%")
```

Task 10: Show how 10-fold cross validation can be used to build a naïve bayes classifier and report the accuracy of this model.

```
from sklearn.model_selection import cross_val_score

gnb = GaussianNB()

scores = cross_val_score(gnb, X_train_encoded, y_train, cv=10)
```

```
for i, score in enumerate(scores, 1):
    print(f"Accuracy for fold {i}: {score * 100:.2f}%")

print(f"\nMean Accuracy: {scores.mean() * 100:.2f}%")
```

Conclusion:

In conclusion, the project successfully implemented various machine learning techniques on the selected dataset. Through thorough data analysis, cleaning, and model building, valuable insights were gained into the dataset's characteristics and predictive capabilities. The report provides a comprehensive overview of the project's methodology and findings.