AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology

Report Cover Sheet

Assignment Title:	Report on Airline Passenger Satisfaction Prediction		
Assignment No:	1	Date of Submission:	18 January 2025
Course Title:	Introduction to Python		
Course Code:	00789	Section:	A
Semester:	Fall 2024-25	Course Teacher:	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 such collaborationhas 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 your own. 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 copy my/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.:	05

No	Name	ID	Program	Signature
1	Md. Labib	21-45925-3	BSc [CSE]	
2	Hojaifa Hossain	20-44218-3	BSc [CSE]	
3	Md. Shadman Tahsin Khan	21-45796-3	BSc [CSE]	
4			Choose an item.	
5			Choose an item.	
6			Choose an item.	
7			Choose an item.	
8			Choose an item.	
9			Choose an item.	
10			Choose an item.	

Faculty use only

FACULTYCOMMENTS	**Marks Obtained**	

Dataset Overview

The dataset contains an airline passenger satisfaction survey, aimed at analyzing factors that contribute to passenger satisfaction or dissatisfaction. It includes various features such as gender, customer type, age, type of travel, flight class, flight distance, and satisfaction levels of passengers. The target column, "Satisfaction," indicates whether a passenger was satisfied, neutral or dissatisfied with their airline experience. There are 25 columns and almost 26000 entries in total in the dataset. The goal is to predict passenger satisfaction based on these features.

Task 1: Load Dataset

The first task involves reading the dataset into the program. We used the pandas library to load the dataset. Here's the code used to load the CSV file into a Data Frame. This step successfully loads the dataset and previews the first few rows.

```
file_path = '/content/drive/My Drive/Colab Notebooks/datasets/test.csv'

df = pd.read_csv(file_path)

df.head()
```

Task 2: Data Cleaning

For data cleaning, we applied techniques to handle duplicates and missing values. we removed duplicate rows using drop_duplicates(). For missing values, we used the mean for numerical columns and the mode for categorical columns. This ensures the data is clean and ready for analysis.

```
df = df.drop_duplicates()

num_col = df.select_dtypes(include=['float64', 'int64']).columns
df[num_col] = df[num_col].fillna(df[num_col].mean())

cat_col = df.select_dtypes(include=['object']).columns
for column in cat_col:
    df[column] = df[column].fillna(df[column].mode()[0])

df.info()
```

Task 3: Frequency Distribution Analysis

For the analysis of feature distributions, we used the matplotlib library to draw histograms for both numerical and categorical features. The categorical features were visualized using bar plots. All plots were combined in a single figure using subplots() for better readability.

```
num_columns = df.select_dtypes(include=['float64', 'int64']).columns
cat_columns = df.select_dtypes(include=['object', 'category']).columns

all_columns = list(num_columns) + list(cat_columns)
num_plots = len(all_columns)
rows = (num_plots + 4) // 5

fig, axes = plt.subplots(nrows=rows, ncols=5, figsize=(15, 3 * rows))
axes = axes.flatten()

for i, column in enumerate(all_columns):
    ax = axes[i]
    if column in num_columns:
        ax.hist(df[column], bins=20, edgecolor='black')
        ax.set_title(column)
        ax.set_xlabel('Value')
        ax.set_ylabel('Frequency')
    else:
        value_counts = df[column].value_counts()
        ax.bar(value_counts.index.astype(str), value_counts.values, color='skyblue', edgecolor='black')
        ax.set_title(column)
        ax.set_xlabel('Category')
        ax.set_ylabel('Count')

plt.tight_layout()
plt.show()
```

Task 4: Feature Scaling

In this task, we applied feature scaling using the StandardScaler from scikit-learn. Scaling ensures that features with different ranges are brought to a comparable scale, which is essential for algorithms like SVM. We excluded the target column (satisfaction) from scaling. This transforms the features, ensuring they have a mean of 0 and a standard deviation of 1.

```
target_column = 'satisfaction'

features = df.drop(columns=[target_column])
target = df[target_column]

features = pd.get_dummies(features, drop_first=False)

scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

scaled_df = pd.DataFrame(scaled_features, columns=features.columns)
scaled_df[target_column] = target.reset_index(drop=True)

print(scaled_df.head())
```

Task 5: Data Splitting

To split the data into training and testing datasets, we used the train_test_split function from scikit-learn. The data was split into 80% training and 20% testing, with a fixed random state (random_state=3241) to ensure reproducibility. This step ensures that we have a separate test dataset to evaluate the model's performance.

```
X = scaled_df.drop(columns=[target_column])
y = scaled_df[target_column]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=3241)

print(f"Training set size: {X_train.shape[0]}")
print(f"Testing set size: {X_test.shape[0]}")
```

Task 6: Apply SVM Classifier

We applied the Support Vector Machine (SVM) classifier using SVC from scikit-learn. The model was trained using the training dataset and then used to make predictions on the test dataset. This trained the model to predict passenger satisfaction based on the features.

```
from sklearn.svm import SVC

svm_model = SVC()
svm_model.fit(X_train, y_train)

print("Model training completed.")
```

Task 7: Confusion Matrix

For evaluating the performance of the SVM model, we used a confusion matrix to compare the predicted values with the actual values. This matrix allows us to evaluate the number of true positives, true negatives, false positives, and false negatives in the prediction.

```
y_pred = svm_model.predict(X_test)
cm = confusion_matrix(y_test, y_pred)

disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap=plt.cm.Blues)
plt.title("Confusion Matrix")
plt.show()
```

Task 8: Model Accuracy

Finally, we calculated the train and test accuracy to assess the performance of the SVM model. This was done using the accuracy_score function from scikit-learn. This provides a comparison of how well the model performs on both the training and testing datasets.

```
train_accuracy = accuracy_score(y_train, svm_model.predict(X_train))
test_accuracy = accuracy_score(y_test, y_pred)

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