AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology

Assignment Title:	Python Final-Term Project Report			
Assignment No:			Date of Submission:	18 January 2025
Course Title:	Programming in Python			
Course Code:			Section:	A
Semester:	Fall	2019-20	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.: 12

No	Name	ID	Program	Signature
1	Khondaker Faisal Ibn Aziz	21-44398-1	BSc [CSE]	
2	Irfan Kabir Shihab	20-43853-2	BSc [CIS]	
3				
4				
5				
6				
7				
8				
9				
10				

Faculty use only		
FACULTYCOMMENTS	**Marks Obtained**	
	Total Marks	

Python Final Project Report

Dataset Description: The dataset used in this project explores player behavior in online gaming and provides insights into how demographic and behavioral factors influence gaming engagement. It includes both numerical and categorical features, ensuring a diverse range of information for analysis and prediction. The dataset contains over 200 entries with the following key features:

- PlayerID: A unique identifier for each player.
- Age: The age of the player (numerical).
- Gender: The gender of the player (categorical: Male/Female).
- Location: The geographical location of the player (categorical).
- GameGenre: The type of game the player plays most frequently (categorical: e.g., Action, Strategy).
- PlayTimeHours: The average number of hours the player spends gaming per day (numerical).
- InGamePurchases: The number of purchases made by the player within games (numerical).
- GameDifficulty: The difficulty level of the game (categorical: Easy, Medium, Hard).
- SessionsPerWeek: The number of gaming sessions the player engages in per week (numerical).
- AvgSessionDurationMinutes: The average duration of each gaming session in minutes (numerical).
- PlayerLevel: The current level of the player in the game (numerical).
- AchievementsUnlocked: The total number of achievements unlocked by the player (numerical).
- EngagementLevel: The level of engagement of the player, categorized as Low, Medium, or High (target variable).

This dataset is well-suited for machine learning tasks due to its mix of numerical and categorical data, making it ideal for analyzing and predicting gaming engagement patterns.

Task Descriptions

Task 1: Load the Dataset

Objective: Load the dataset into the Python environment for analysis and preparation.

Implementation: The dataset was read with a structured data manipulation library. It was loaded into a DataFrame, which enabled easy inspection and access to its structure, including column names and data types. This step guaranteed that the data was in a usable state for future tasks.

Task 2: Data Cleaning

Objective: To deal with any inconsistencies in the dataset, including missing values and duplicates.

Implementation: Removed duplicate rows with drop_duplicates(). Replaced missing numerical values with the column mean and categorical values with 'Unknown' using fillna(). These steps ensured that the dataset was comprehensive, reliable, and ready for analysis.

Task 3: Frequency Distribution Analysis

Objective: Analyze and display the distribution of features in the dataset.

Implementation: Created bar plots for selected features using matplotlib. And used subplot() to arrange all plots in one figure for better comparison. Frequency distributions were plotted for all features to help identify patterns and probable abnormalities.

Task 4: Feature Scaling

Objective: Normalize the feature values to ensure that they contribute equally to the model.

Implementation: Applied one-hot encoding to categorical columns using pd.get_dummies() and scaled all features using StandardScaler. This scaling technique was used to convert all feature values into a consistent range. This phase reduced the possibility of larger-range features dominating the learning process, resulting in more balanced and successful model training.

Task 5: Data Splitting

Objective: Split data into training and testing sets.

Implementation: Used train_test_split() with 80% data for training and 20% for testing, ensuring reproducibility with random_state=3241. This division ensured that the model's performance could be assessed on unseen data, validating its generalization capability.

Task 6: SVM Classifier

Objective: To build a classification model using the Support Vector Machine (SVM) algorithm.

Implementation: An SVM classifier was trained on the labeled training data. Initialized an SVM classifier (SVC) with a linear kernel and trained it using the training dataset.

Task 7: Confusion Matrix

Objective: To evaluate the model's classification performance using a confusion matrix.

Implementation: A confusion matrix was created to show the counts of correctly and wrongly classified cases. Calculated the confusion matrix using confusion_matrix() and visualized it with ConfusionMatrixDisplay to assess prediction accuracy for each class. This highlighted both the strengths and weaknesses of the SVM classifier, guiding potential improvements in future iterations of the model.

Task 8: Accuracy Comparison

Objective: Compare training and testing accuracy.

Implementation: The accuracies were computed for both training and testing datasets. With both accuracies at 82%, the model exhibited balanced performance, indicating no significant overfitting or underfitting and demonstrating robustness. Calculated accuracy for both datasets using accuracy_score() to identify overfitting or underfitting.

Conclusion

This project effectively applied machine learning techniques to a real-world dataset, covering all stages from data preprocessing to model evaluation. The SVM classifier achieved balanced accuracy, reflecting its ability to generalize well. This project provided valuable insights into the practical implementation of machine learning, and future work could include fine-tuning hyperparameters or testing additional algorithms to further enhance performance.