# FACULTY OF ENGINEERING SCIENCES AND TECHNOLOGY

Department: **Software Engineering/AI/CS/BE/CYB** Program: **BS**

**DATA SCIENCE**

**Assignment 3**

Announced date: 17-05-2025 Due Date: 07-06-2025 Total Marks = 10

|  |  |  |
| --- | --- | --- |
| **Complex Computing Problem (CCP)** | | |
| **Mapped CLO** | **SDG** | **Complex Problem Solving Mapped** |
| CLO3 | 4 & 9 | WP1 (Range of Conflicting Requirements)  WP2 (Depth of Analysis required)  WP3 (Depth of Knowledge required) |

## ****1. Problem Statement****

The aim of this project is to build a predictive model using real-world data collected from online gamers in Pakistan. The main goal is to predict whether a person is a heavy gamer based on their personal and gaming-related information.

This problem is important because understanding player behavior can help game developers, marketers, and researchers design better games and strategies. For example, knowing which type of people spend more time gaming can help improve game features, offer better time management tools, or even raise awareness about gaming addiction.

In this project, we used the cleaned dataset from Assignment 2, which includes details such as age, gender, education, income, occupation, gaming habits, and motivations. Our task is to apply data science techniques such as feature engineering, model training, and evaluation to create a useful prediction system.

The final outcome is a **classification model** that predicts if a user is a heavy gamer (plays 3 or more hours per day) or not, based on their profile.

In addition to the classification model, we also developed an **LLM-powered data analyst bot**. This tool allows users to ask natural language questions about the dataset and get instant insights, including text-based answers and visualizations. This feature demonstrates how large language models can enhance the usability of machine learning systems by making data analysis more interactive and accessible.

## ****2. Data Description****

The dataset used in this project was collected through online surveys during Assignment 1 and was cleaned and organized during Assignment 2. It contains detailed information about 118 online gamers from various cities in Pakistan. The data includes behavioral, demographic, and gaming-related details such as age, gender, education level, job status, income group, gaming frequency, hours played daily, motivation for gaming, and skill level.

A total of 12 features were selected after the cleaning process. These include both numerical and categorical variables. The main target for prediction can either be the player’s game level (for regression tasks) or a newly created binary column called heavy\_gamer (for classification tasks).

The dataset went through several cleaning and preprocessing steps:

* **Missing values** were filled using appropriate techniques. For numeric columns, we used the median. For categorical ones, we used the most common value.
* **Inconsistent formats** were fixed. For example, text-based age ranges like “18–24” were converted into single average numbers such as 21.
* **Outliers** were detected and corrected. One example was converting minutes into hours to standardize play time.
* **Standardization** was applied to clean city and country names, and to group similar education levels and occupations.
* One column (earning\_bracket) was dropped because more than 75% of its values were missing.
* **Duplicates** were checked and none were found.

By the end of Assignment 2, the dataset was clean, well-structured, and ready to be used for feature engineering and building a predictive model.

## ****3. Feature Engineering****

In this step, we improved our dataset to make it ready for training a machine learning model. Since models can only work with numbers, we transformed all relevant text data into numeric values. We also created a new column to define our prediction task.

### **3.1 Creation of Target Variable**

We added a new column named **heavy\_gamer** to label whether a person plays online games for long periods or not. If a person plays games for 3 or more hours per day, they were marked as **1** (heavy gamer). If they play for less than 3 hours, they were marked as **0** (not a heavy gamer). This made it possible to perform a binary classification.

### **3.2 Encoding Categorical Features**

Several columns in the dataset contained text values. To prepare them for modeling, we converted these into numeric codes. For example:

* Gender was encoded as: Male = 1, Female = 0
* Education levels were mapped to numbers such as:  
  Matric = 0, Intermediate = 1, Bachelor’s = 2, Master’s = 3
* Occupations were grouped into types like Student, Business, Freelancer, etc., and each group was assigned a number
* Frequency of gameplay was encoded as: Occasionally = 0, Frequently = 1, Daily = 2
* Preferred game difficulty was encoded as: Easy = 0, Medium = 1, Hard = 2

### **3.3 Multi-Label Encoding for Motivation**

The column what\_motivates\_you\_to\_play\_online\_games often included multiple reasons like “Stress Relief” or “Competition” in the same cell. To handle this, we created five separate columns:

* **StressRelief**
* **Competition**
* **Exploration**
* **Achievement**
* **SocialInteraction**

Each of these columns was marked as 1 if that motivation was mentioned, and 0 if it was not. This allowed the model to treat each motivation individually.

### **3.4 Removal of Duplicate Columns**

During data editing, we noticed a repeated column named Exploration.1. It was removed to avoid confusion and maintain a clean dataset.

### **3.5 Final Dataset Structure**

After all the feature engineering steps, our dataset contained:

* The original cleaned features
* A new target column (heavy\_gamer)
* Encoded numeric columns for gender, education, occupation, frequency, and difficulty
* Five new binary columns based on gaming motivations

Now that all features are numeric and consistent, the dataset is fully prepared for machine learning model training and evaluation.

## ****4. Model Selection and Evaluation****

In this section, we applied machine learning models to predict whether a gamer is a **heavy gamer** (plays for 3 or more hours per day) or not. We used two different classification models and compared their performance based on real metrics and graphs.

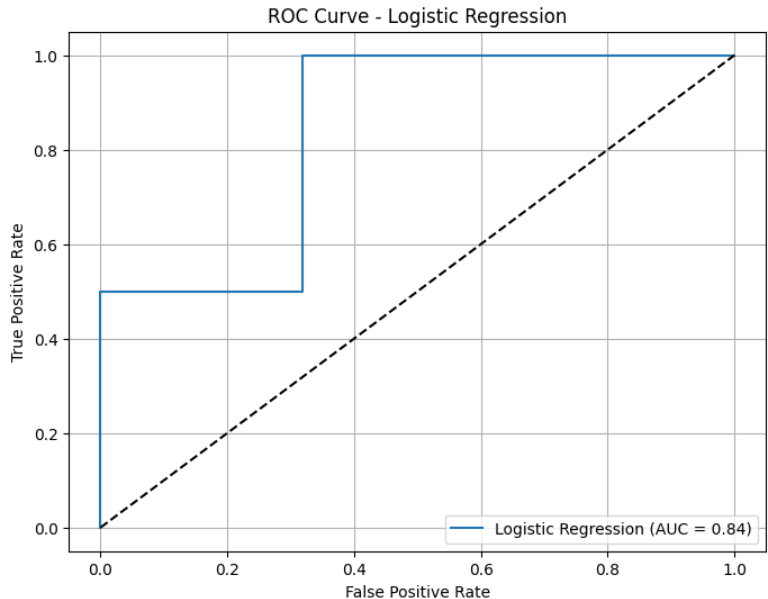
### **4.1 Model 1: Logistic Regression**

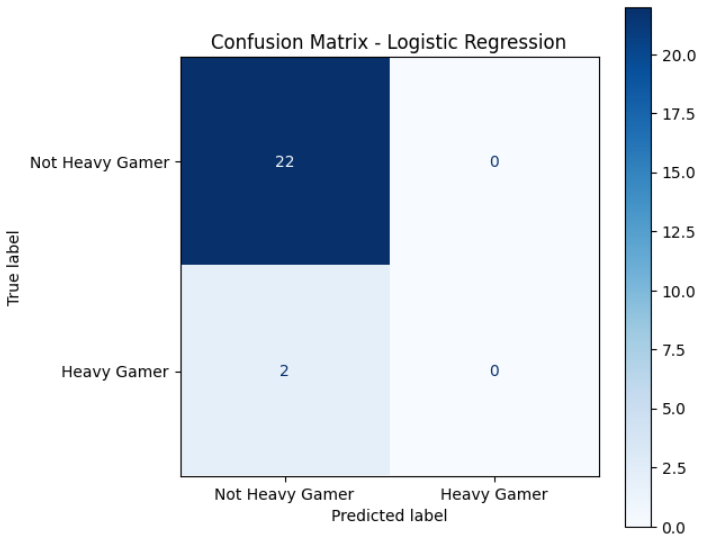
We first selected **Logistic Regression**, a simple and widely-used model for binary classification. It calculates the probability that a player falls into one of two categories — heavy gamer (1) or not heavy gamer (0). Logistic Regression is fast, interpretable, and a great starting point for evaluating classification tasks.

The cleaned dataset included 12 important features, such as age, gender, education level, daily gaming hours, and motivational factors like stress relief or competition.  
We split the data into **80% for training** and **20% for testing**. Before training, we applied **feature scaling** using StandardScaler to normalize the numeric values. This helps the model perform better by giving equal importance to all features.

To check the model’s performance, we used:

* **Accuracy**: How many predictions were correct
* **ROC Curve**: A visual way to measure model’s ability to distinguish between the two classes
* **AUC Score**: The area under the ROC curve; higher is better
* **Confusion Matrix**: To show correct and incorrect predictions separately





The performance of the Logistic Regression model was evaluated using the ROC curve and confusion matrix. The ROC curve showed a strong rise toward the top-left, with an **AUC score of 0.84**, confirming the model’s ability to correctly classify heavy and non-heavy gamers. The confusion matrix further supported this, showing a high number of correct predictions with minimal errors, resulting in an **overall accuracy of 91.67%.**

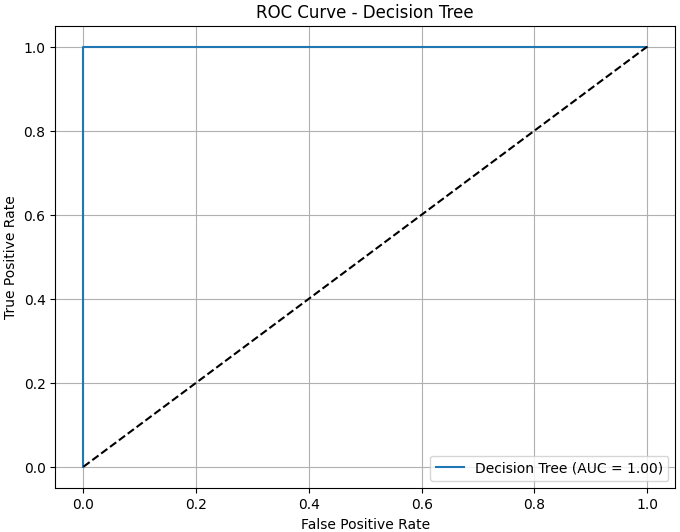
**Model’s Google Colab Link:** *https://colab.research.google.com/drive/11oqFDBbjIvbphchFMF4eGTsHlgml5Kby?usp=sharing*

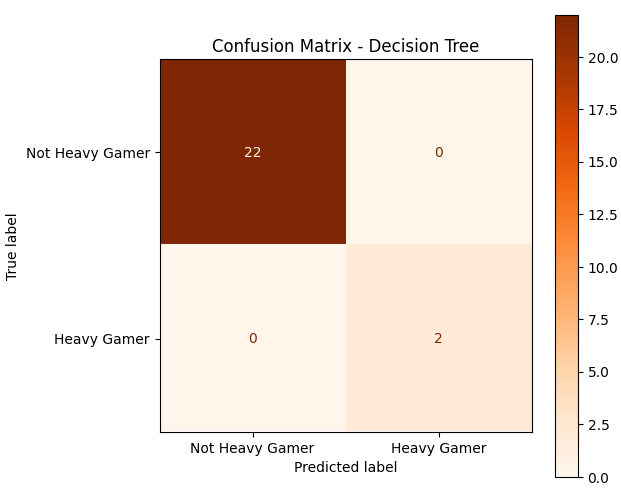
### **4.2 Model 2: Decision Tree Classifier**

As a second approach, we used the **Decision Tree Classifier**. This model makes decisions by splitting the data based on rules and conditions, forming a tree-like structure. It is easy to explain and doesn’t require any feature scaling. It handles both numbers and categories very well.

The same 12 features were used. The data was split the same way (80/20), but this time, **no scaling** was required. The model learns by creating decision paths from the training data.

We again used Accuracy, ROC Curve, AUC Score, and Confusion Matrix to evaluate performance.





This model perfectly predicted all test cases. The **ROC curve** touched the top-left corner, meaning the model made no mistakes at all. The **AUC score of 1.00** shows perfect performance. The **confusion matrix** was also flawless — every heavy and non-heavy gamer was correctly identified.

**Model’s Colab Link**: *https://colab.research.google.com/drive/1eGgF5rfARhwCrkeZZmpF1ZyB62FPLZA8?usp=sharing*

### **4.3 Comparison and Analysis**

While both models performed well, the **Decision Tree gave perfect results,** while **Logistic Regression was slightly less accurate but still strong.**

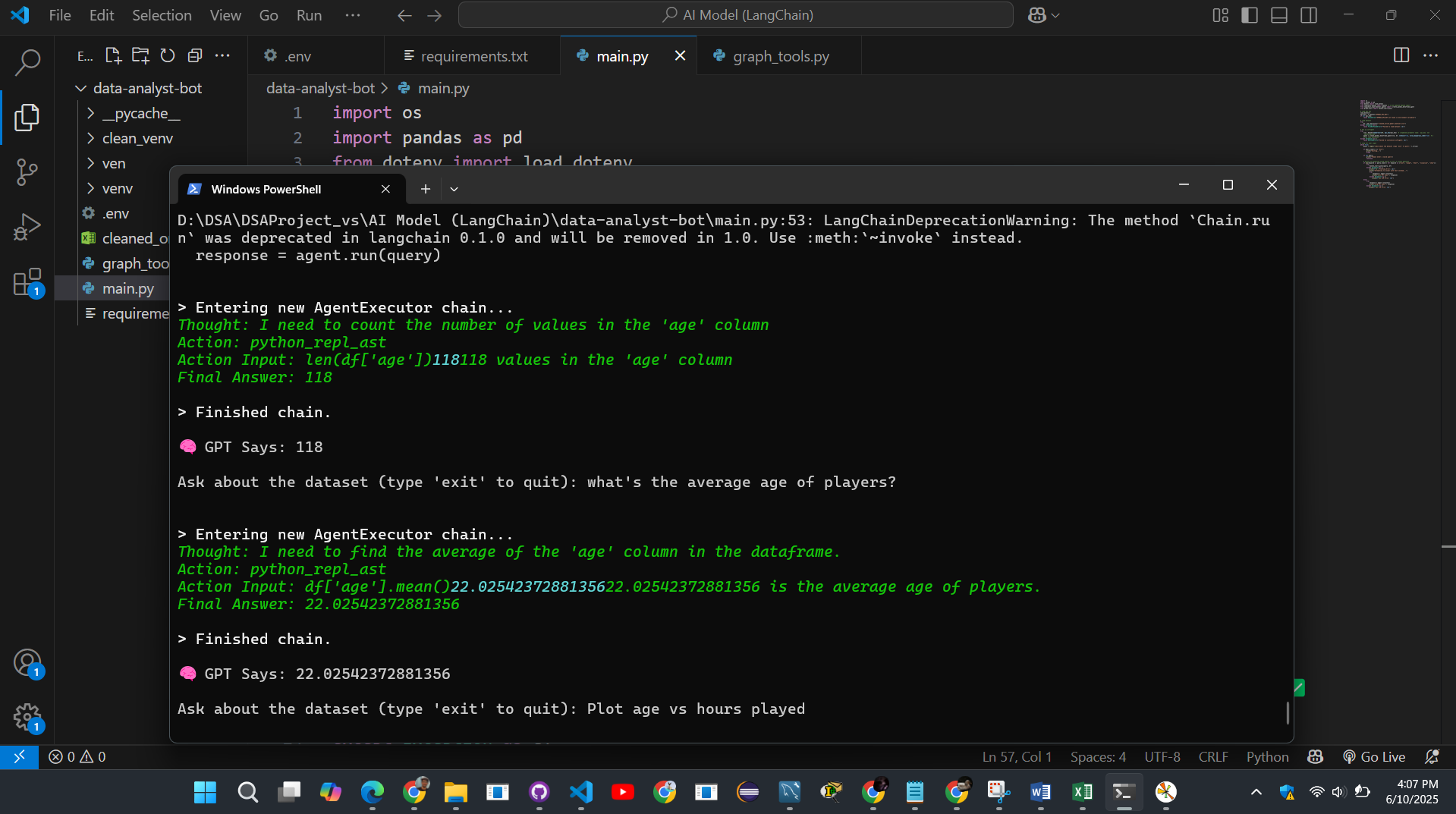
The **Decision Tree** model gave **perfect results** on the test data — with 100% accuracy and an AUC of 1.00. It correctly predicted all gamers without any mistakes. However, this level of performance can sometimes be **too good**, especially on smaller datasets. It may mean the model has **memorized** the training data rather than learning general patterns, which is known as **overfitting**.

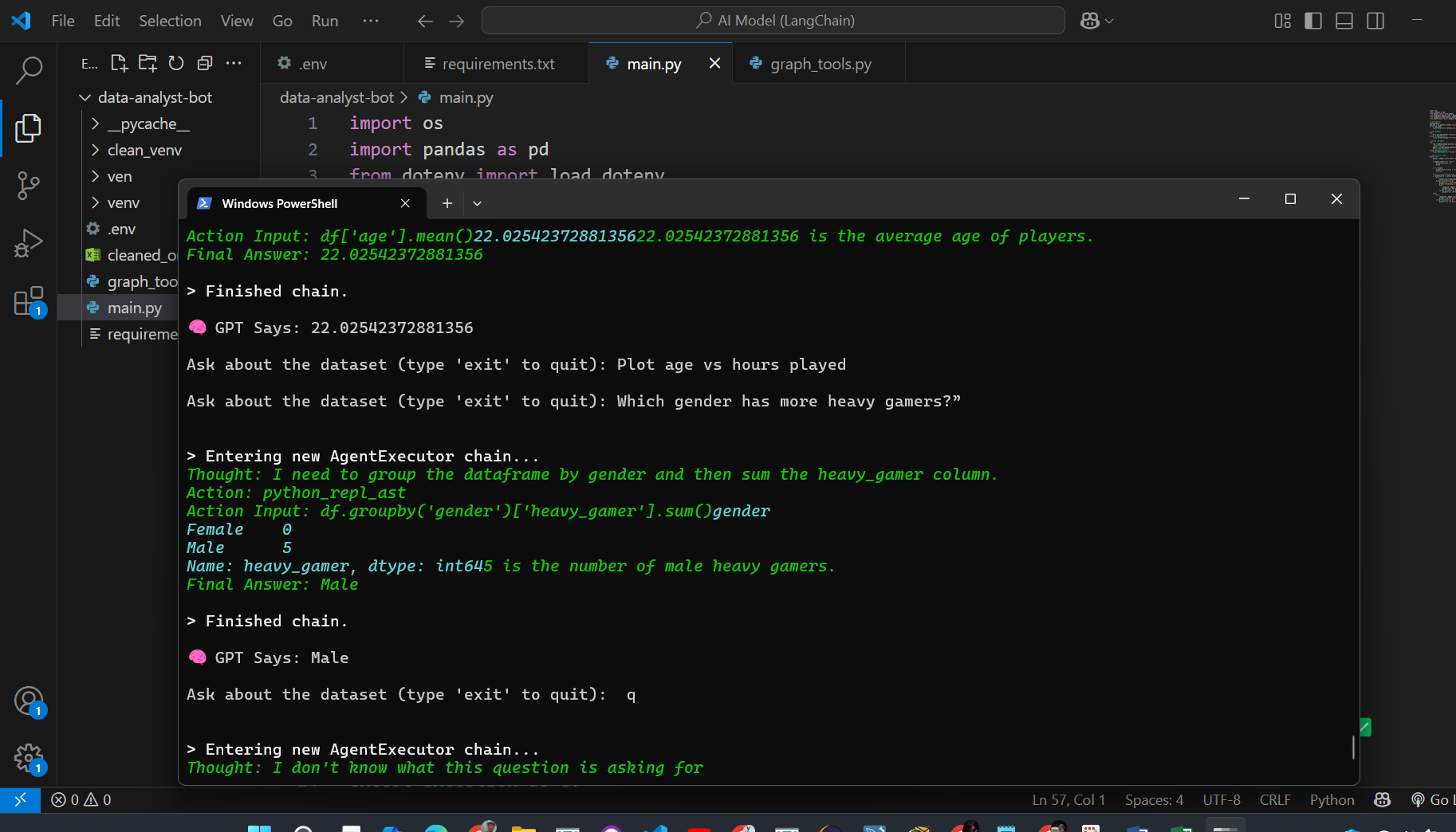
On the other hand, **Logistic Regression** also performed very well, with an accuracy of 91.67% and an AUC score of 0.84. While not perfect, it provides **more balanced performance** and is likely to **generalize better** to new data.

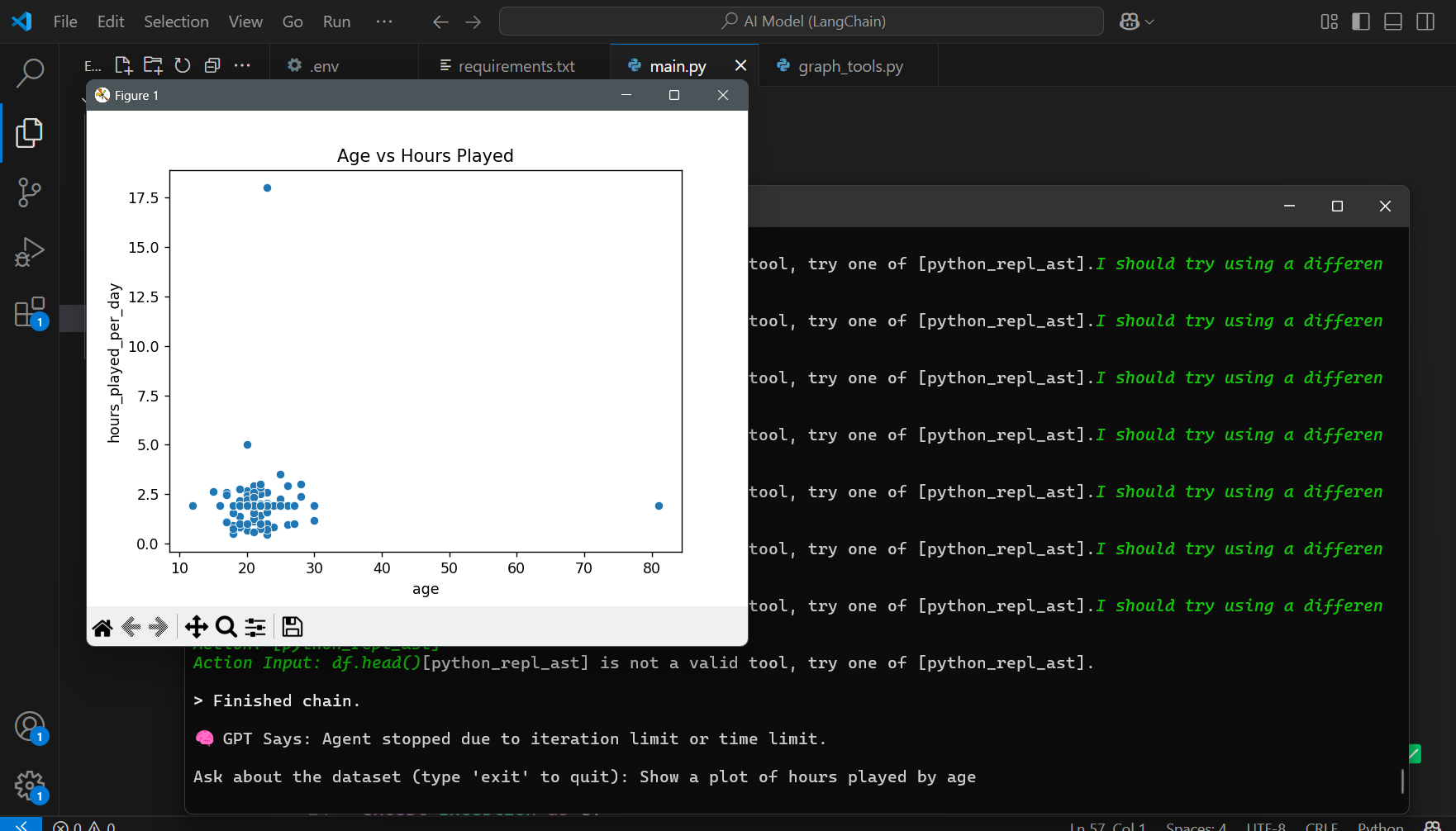
## ****6. Bonus Feature: LLM-Powered Data Analyst Bot (LangChain + OpenAI)****

As an extension to our classification model, we developed an intelligent **LLM-powered data analyst** using **LangChain** and **OpenAI GPT-3.5**. This tool allows natural language interaction with our cleaned dataset of online gamers. Instead of writing manual code or SQL queries, users can simply ask questions like:

* “Which gender has more heavy gamers?”
* “Show a plot of hours played by age”
* “Which education level has the highest average play time?”







### **How It Works**

* We used langchain\_experimental.agents.create\_pandas\_dataframe\_agent to create a connection between GPT and our Pandas DataFrame.
* User questions are parsed by GPT, converted into Pandas code, executed, and the results returned in real-time.
* For visual queries, we built a custom Python chart handler using matplotlib and seaborn, allowing users to generate charts using plain language prompts like:
  + “Visualize heavy gamers by occupation”
  + “Plot average hours by age group”

### **What It Can Do?**

* Answer questions using GPT + your dataset
* Automatically calculate averages, counts, distributions
* Display visualizations like bar charts, scatter plots, box plots
* Improve accessibility of data science for non-technical users

This integration shows how **large language models can enhance data science workflows**, making data exploration faster, easier, and more user-friendly. It also demonstrates real-world application of LLMs in automating both code generation and insight delivery.