UNIVERSITY MANAGEMENT SYSTEM (INTEGRATED WITH ML METHODS)

Project Overview

This project is a software application developed in Python, featuring a graphical user interface built with **Tkinter** and a relational database managed using **SQLite**. It is designed to simulate a university management system, allowing users to handle information about students, courses, professors, and exams.
The application stands out for its ease of use and the integration of a **machine learning model (linear regression)** to predict students' grade averages based on their age.

Project Features

1. Student Management

The application enables university administrators to manage student information through the following features:

- **Student Registration:**
  Users can add a new student by entering their name, surname, and age. After the data is entered:
  - The information is saved to the SQLite database.
  - A linear regression model predicts the grade average for students of the same age.
  - The predicted grade average is displayed in a popup message.
- **View Students:**
  Registered students can be viewed in an interactive tabular interface that displays all details for each student.

---

2. Course Management

The system allows the management of university courses through the following functionalities:

- **Add New Courses:**
  Users can register new courses by specifying the course name and the assigned professor.
- **View Courses:**
  Users can access a list of all registered courses along with details about the professors responsible for each course.

---

3. Exam Management

The exam management features include:

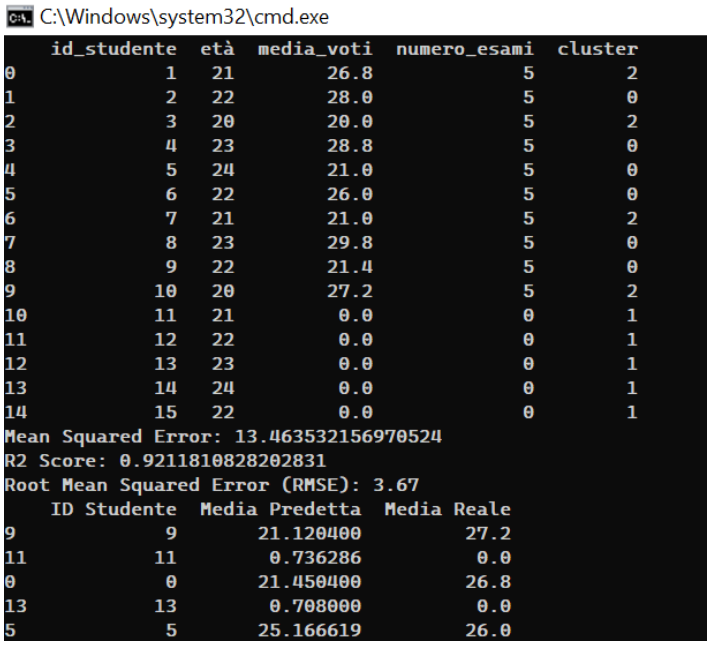
- **Add Exams:**
  Users can add details of exams taken by students, specifying the course, grade, and the student associated.
- **View Exams:**
  A tabular view displays the exams taken by students. Each record includes:
  - Student name and surname.
  - The course for which the exam was taken.
  - The grade obtained.

---

4. Grade Average Prediction

One of the standout features of this project is the integration of a linear regression model:

- **Model Training:**
  The model is trained on data from the database, correlating students' ages with their grade averages.
- **Grade Prediction:**
  Once trained, the model can estimate the grade average for a new student based on their age.



---

5. Interactive Visualization

The application offers various methods for interactive data visualization:

- **Treeview Tables:**
  Graphical tables are used to display detailed information about students, courses, and exams. These tables include scrollbars for smooth navigation.
- **Popup Dialogs:**
  Data entry is facilitated through intuitive dialog windows that guide the user step-by-step.

---

6. Main Menu

An interactive menu in the top bar provides easy navigation across the application's main functionalities:

- **View Options:**
  Access to view information about students, professors, courses, exams, and student transcripts.
- **Insert Options:**
  Tools for inserting new data related to students, courses, and exams.

Technical Architecture

SQLite Database

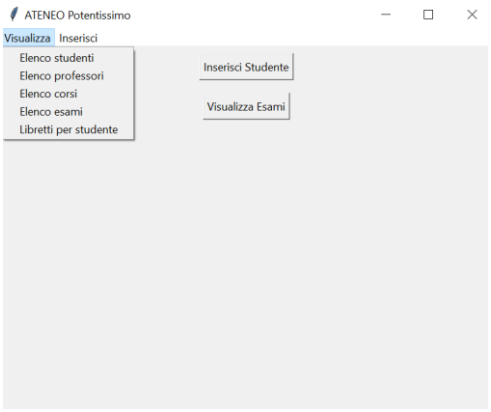
The project uses an SQLite database named ateneo.db, structured with the following main tables:

- **Students:**
  Contains information about students (ID, name, surname, age).
- **Courses:**
  Contains details about courses (ID, name, associated professor).
- **Exams:**
  Links students to specific courses and records the grades obtained.
- **Professors:**
  Stores information about professors (ID, name, surname).

Tkinter GUI

The user interface is designed using Tkinter and includes:

- **Main Window:**
  Access to all major functionalities through buttons and dropdown menus.



- **Secondary Windows:**
  Used for tabular data visualization and the entry of new information.

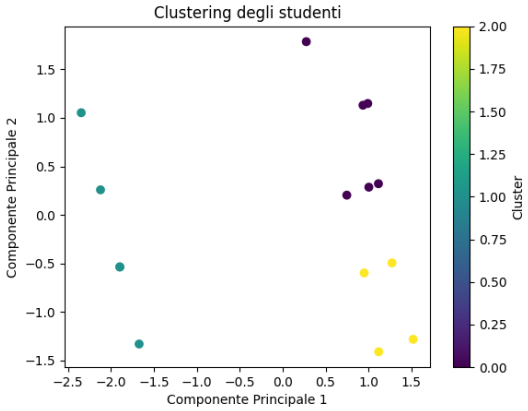
Linear Regression

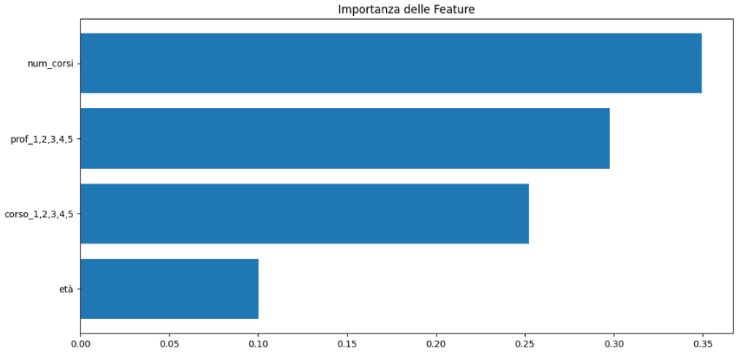
The scikit-learn library is used to implement the linear regression model:

- Training data is retrieved from the database.
- The model is trained to correlate students' ages with their grade averages.
- Predictions are made for new students based on their input age.

User Features

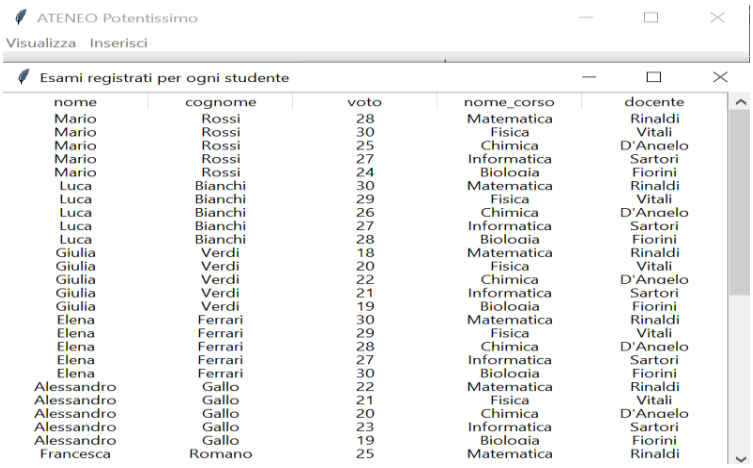
1. **Intuitive Interface:**
   The application is user-friendly, featuring buttons, tables, and dialog windows.
2. **Immediate Feedback:**
   Each operation (data insertion or visualization) is accompanied by informational or error popups, ensuring a clear workflow for the user.
3. **Data-Based Prediction:**
   The machine learning integration adds value by providing useful estimates based on real system data.





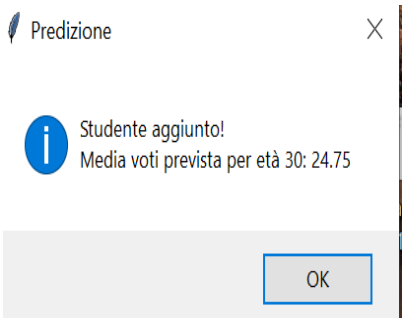
Usage Examples

- **Registering a New Student:**
  The user specifies the student's name, surname, and age. The system saves the data and displays the predicted grade average for students of the same age.
- **Viewing Available Courses:**
  The user selects the appropriate menu option to retrieve a complete list of courses and their respective professors.



nome	cognome	voto	nome_corso	docente
Mario	Rossi	28	Matematica	Rinaldi
Mario	Rossi	30	Fisica	Vitali
Mario	Rossi	25	Chimica	D'Angelo
Mario	Rossi	27	Informatica	Sartori
Mario	Rossi	24	Biologia	Fiorini
Luca	Bianchi	30	Matematica	Rinaldi
Luca	Bianchi	29	Fisica	Vitali
Luca	Bianchi	26	Chimica	D'Angelo
Luca	Bianchi	27	Informatica	Sartori
Luca	Bianchi	28	Biologia	Fiorini
Giulia	Verdi	18	Matematica	Rinaldi
Giulia	Verdi	20	Fisica	Vitali
Giulia	Verdi	22	Chimica	D'Angelo
Giulia	Verdi	21	Informatica	Sartori
Giulia	Verdi	19	Biologia	Fiorini
Elena	Ferrari	30	Matematica	Rinaldi
Elena	Ferrari	29	Fisica	Vitali
Elena	Ferrari	28	Chimica	D'Angelo
Elena	Ferrari	27	Informatica	Sartori
Elena	Ferrari	30	Biologia	Fiorini
Alessandro	Gallo	22	Matematica	Rinaldi
Alessandro	Gallo	21	Fisica	Vitali
Alessandro	Gallo	20	Chimica	D'Angelo
Alessandro	Gallo	23	Informatica	Sartori
Alessandro	Gallo	19	Biologia	Fiorini
Francesca	Romano	25	Matematica	Rinaldi

- **Grade Prediction:**
  The system automatically computes a prediction based on database data and displays it to the user.



Conclusions

This project serves as an effective tool for university management, combining established technologies like SQLite and Tkinter with modern machine learning techniques to offer advanced features and an intuitive user experience.

The application is a demonstration of skills in:

- Designing relational databases.
- Developing graphical user interfaces.
- Integrating machine learning models.
- Managing multi-functional applications.