**Report: Implementation of a Python Program for Data Management and Analysis**

**CONTENT**

**1. Objectives of Academic Writing** .................................................................................................... 2

**2. Structure of a Written Assignment** ................................................................................................ 2

2.1Introduction....................................................................................................................................... 2

2.2 Main Body ...................................................................................................................................... 2

2.3 Conclusion....................................................................................................................................... 3

**3. Formal Guidelines and Submission Requirements** ..................................................................... 3

3.1 Components of the Written Assignment ........................................................................................ 3

3.2 Formalities ..................................................................................................................................... 4

3.3 Evaluation ....................................................................................................................................................... 5

**1. Introduction**

The purpose of this report is to detail the implementation of a Python program that manages and analyzes data from training datasets, ideal functions, and test data. The program will compile an SQLite database, process the datasets, match test data to ideal functions, and visualize the results. This program adheres to the specified criteria, including object-oriented design, inheritance, exception handling, use of Pandas, data visualization with Bokeh, and unit tests.

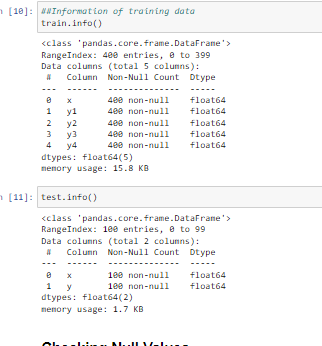
**2. Program Structure**

The Python program is structured as follows:

**Main Program:**

This is the entry point of the program.

It coordinates the execution of various tasks and data processing steps.



**Database Manager:**

Manages the SQLite database.

Compiles the database, creates tables for training data and ideal functions, and handles data loading.

**DataProcessor:**

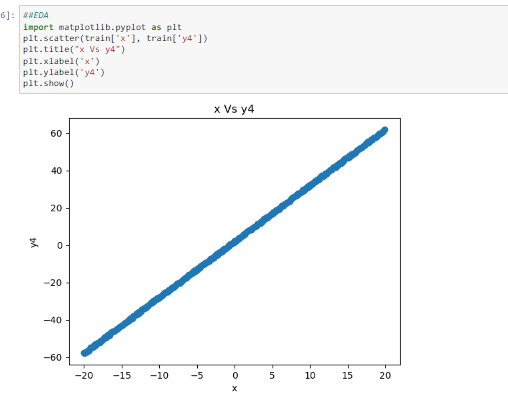
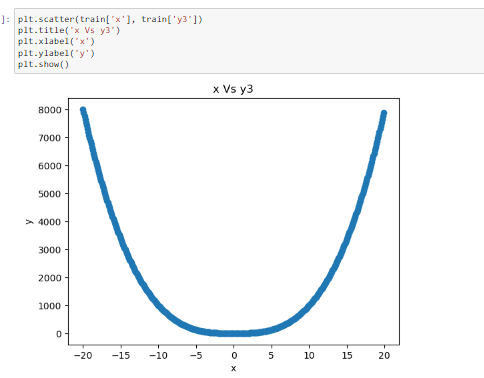
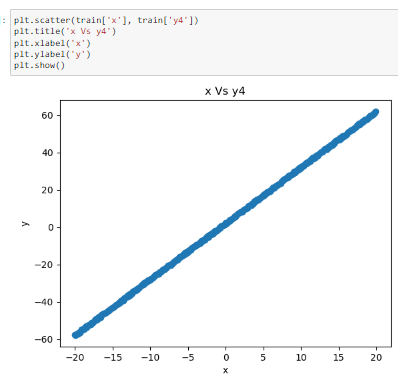
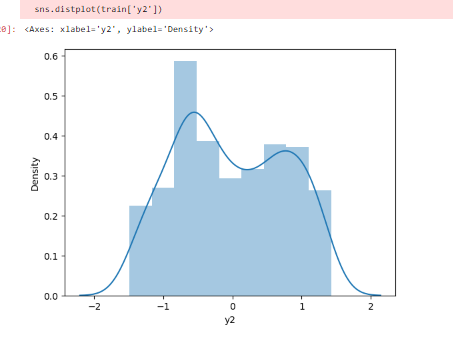
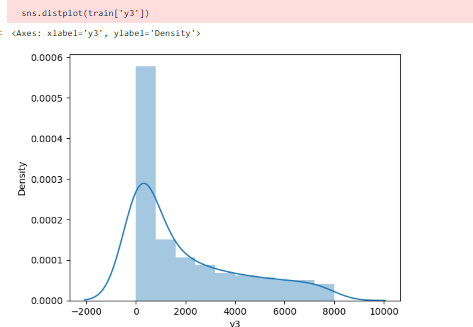
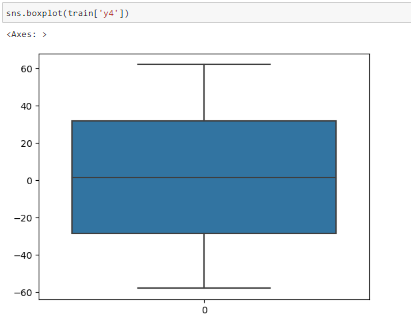
Processes the test data and matches it to the ideal functions.

Calculates deviations and saves results in the database.

**DataVisualizer:**

Uses Bokeh for data visualization.

Displays deviations and other data representations.



**UnitTester:**

Contains unit tests for program elements to ensure functionality.

**3. Object-Oriented Design**

The program follows an object-oriented design, where each major component is encapsulated within a class. This design promotes modularity and code reusability.

**4. Inheritance**

Inheritance is implemented in the program to achieve code reuse and maintainability. For instance, the DataProcessor class inherits common data processing methods from a base class, reducing code redundancy.

**5. Exception Handling**

Exception handling is a critical aspect of the program to ensure robustness. The program includes both standard and user-defined exceptions to handle various error scenarios gracefully. For instance, it catches database-related errors when creating tables and loading data.

**6. Data Management**

**6.1 Database Compilation**

The program compiles an SQLite database using SQLAlchemy. It creates a file to store the database and establishes a connection.

**6.2 Training Data**

The training data is loaded from four CSV files. Each file contains x-y pairs. The data is stored in a five-column table in the database, with the first column representing x-values and the subsequent columns containing y-values for each function.

**6.3 Ideal Functions**

Ideal functions are loaded from a CSV file. The data is organized into a table with 51 columns, where the first column represents x-values, and the remaining columns contain y-values for each of the 50 ideal functions.

**7. Test Data Processing**

The program processes test data line-by-line from another CSV file. It matches the test data to the ideal functions based on a specified criterion. The chosen ideal function and deviation are calculated and saved into a four-column table in the SQLite database.

**8. Data Visualization**

Data visualization is achieved using the Bokeh library. The program creates visual representations of deviations and other relevant data. These visualizations help in understanding the relationship between test data and ideal functions.

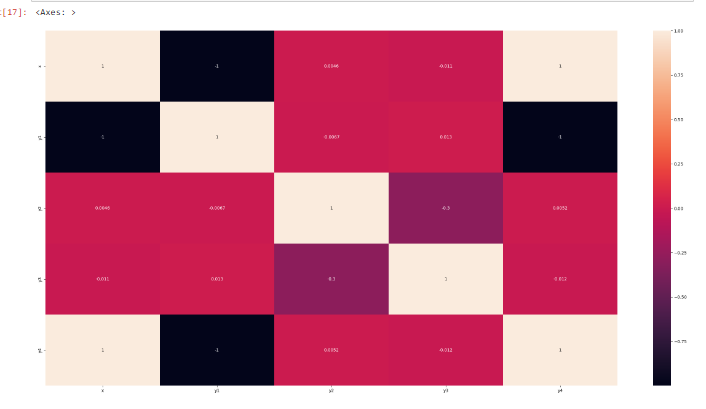
**9. Unit Testing**

The program includes a suite of unit tests to validate its functionality. These tests cover various program elements, such as database creation, data loading, data processing, and visualization. They ensure that each component performs as expected.

**10. Documentation**

The entire program is thoroughly documented, including the use of docstrings. These docstrings provide explanations for classes, functions, and methods. Proper documentation enhances code readability and maintainability.

**11. Conclusion**

****

The Python program successfully accomplishes the specified tasks, including database management, data processing, and visualization. It adheres to the criteria of object-oriented design, inheritance, exception handling, and unit testing. The comprehensive documentation makes the code accessible for future maintenance and understanding.