Database Definition and Integration

Create mySQL Database

To define a MySQL database that will store the training data, you need to perform the following steps:

Install MySQL: If you haven't already, you should install MySQL on your system. You can download and install MySQL from the official website.

Create a Database: You'll create a new database to store the training data. You can do this using the MySQL command-line client or a GUI tool like phpMyAdmin. Here's an example of how to create a database using the command-line client:

CREATE DATABASE crime_prediction;

Create a Table: You'll need a table to store the training data. Define the table structure to match the data you provided in your documentation. You can do this using SQL. Here's an example of how to create a table for your training data:

```sql
CREATE TABLE training_data (
    id INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(255),
    Age INT,
    DrugTest ENUM('Y', 'N'),
    Obedient ENUM('Y', 'N'),
    EmotionScore INT,
    ConfidenceScore INT,
    ConsistencyScore INT,
    InnocenceScore INT,
    Gender ENUM('M', 'F'),
    Criminal ENUM('Yes', 'No')
);
```

Insert Data: After defining the table, you can insert your training data into the database. You can do this with a series of INSERT statements or by importing data from a CSV file.

Here's an example of how to insert a single row of data:

```sql
INSERT INTO training_data (Name, Age, DrugTest, Obedient,
EmotionScore, ConfidenceScore, ConsistencyScore,
InnocenceScore, Gender, Criminal)
```

```
VALUES ('Michael', 20, 'Y', 'N', 90, 85, 50, 67, 'M', 'No');
```

You would repeat this for all the rows of data in your documentation.

Database Integration

Connect to the Database: In your Python code, you'll need to connect to the MySQL database to interact with it. You can use a library like mysql-connector-python or any other MySQL connector for Python.

Here's an example of how to establish a connection:

```python
import mysql.connector

# Establish a connection to the database
db_connection = mysql.connector.connect(
    host='your_host',
    user='your_user',
    password='your_password',
```

```
    database='crime_prediction'
)
```

Perform Database Operations: You can use the connection to perform operations like inserting new data, querying existing data, or updating records. You'll need to create appropriate functions or methods for these operations.

Close the Connection: After you're done with database operations, remember to close the database connection:

```
db_connection.close()
```

Be sure to replace 'your_host', 'your_user', and 'your_password' with your actual MySQL server information.