

Customer Feedback Management System

Project Overview

The **Customer Feedback Management System** is a Python-based application designed to collect, store, and analyze customer feedback. This system allows users to gather essential information such as customer names, ratings, and detailed feedback. The feedback is stored in a CSV file for further data processing, enabling basic analysis such as categorizing feedback as positive or negative based on keywords.

This project is designed to demonstrate the use of basic Python programming concepts such as lists, dictionaries, control flow (loops and conditionals), functions, and file handling (CSV). Additionally, it uses the **Pandas** library to manage data efficiently, making it easy to save and read feedback from a CSV file.

Features

1. Collect Feedback

- Users can input the customer's name, rating (1-5), and detailed feedback.
- The system collects this information for multiple customers and stores it for further analysis.

2. Store Feedback in CSV

- After collecting feedback, users can save the data to a CSV file for long-term storage and further analysis.

- The CSV format ensures the feedback is easy to retrieve and manipulate using tools like Pandas or Excel.

3. Analyze Feedback

- The system performs a basic analysis of the feedback by looking for positive keywords (e.g., "good").
- It categorizes the feedback as **Positive** or **Negative**, depending on whether the keyword is found.

4. Display Feedback

- Users can display the feedback stored in the CSV file.
- This allows for quick reviews of the data collected, including names, ratings, and feedback comments.

5. Menu-Driven Interface

- The system provides an interactive menu that allows users to navigate through various functions, including collecting feedback, saving data, analyzing feedback, and displaying feedback.

How It Works

Data Structures

1. **customer_names (list):**
 - a. Stores the names of customers providing feedback.
 - b. Example:
2. **customer_ratings (list):**
 - a. Stores the ratings provided by customers on a scale of 1 to 5.
 - b. Example:
3. **customer_feedbacks (list):**

- a. Stores the detailed feedback provided by customers.
- b. Example:

Functions

1. *collect_feedback()*

- **Purpose:** Collects feedback from multiple customers.
- **Functionality:**
 - Prompts the user to input the customer's name, rating, and feedback.
 - Appends the input data to the appropriate lists (customer_names, customer_ratings, and customer_feedbacks).

2. *save_feedback_to_csv()*

- **Purpose:** Saves the collected feedback data to a CSV file.
- **Functionality:**
 - Uses the **Pandas** library to store the feedback in a CSV file named customer_feedback.csv.
 - Each row in the CSV file represents feedback from a single customer.

3. *analyze_feedback()*

- **Purpose:** Analyzes the feedback data to categorize it as **Positive** or **Negative**.
- **Functionality:**
 - Searches for the keyword "good" in each feedback entry to determine if the feedback is positive.
 - Stores the result in a list and displays whether each feedback is positive or negative.

4. *display_feedback()*

- **Purpose:** Displays the feedback stored in the CSV file.
- **Functionality:**

- Reads the data from the customer_feedback.csv file and prints the first few rows of feedback.
- Provides an overview of the feedback data, including customer names, ratings, and feedback.

5. feedback_menu()

- **Purpose:** Provides a menu-driven interface for the user to navigate the system.
- **Functionality:**
 - Displays a menu with options to collect feedback, save it to a CSV file, analyze feedback, or display feedback.
 - Allows users to select actions by inputting numbers (e.g., 1 for collecting feedback).

Code Flow

1. **User Interaction:**
 - a. The program starts by displaying a menu with several options: collect feedback, save feedback, analyze feedback, display feedback, or exit the system.
 - b. The user selects an option by inputting a number (1-5).
2. **Collecting Feedback:**
 - a. If the user selects the option to collect feedback, they are prompted to enter customer names, ratings, and feedback.
 - b. This data is temporarily stored in the corresponding lists (customer_names, customer_ratings, customer_feedbacks).
3. **Saving Feedback:**
 - a. After collecting feedback, the user can select the option to save the feedback to a CSV file.
 - b. The system will then create or overwrite a CSV file containing all feedback.
4. **Analyzing Feedback:**

- a. The system can analyze the feedback by checking for specific keywords like "good" to determine whether the feedback is positive.
 - b. The system prints whether each feedback entry is categorized as positive or negative.
5. **Displaying Feedback:**
- a. Users can view all saved feedback by selecting the display option, which reads the data from the CSV file and prints the first few entries for review.
6. **Exiting the Program:**
- a. The user can select the option to exit, terminating the program.

Sample Interaction

Collecting and Saving Feedback

```
Enter name for customer 1: Eva
Enter rating for customer 1 (1-5): 1
Enter feedback for customer 1: Poor
Enter name for customer 2: Minnie
Enter rating for customer 2 (1-5): 5
Enter feedback for customer 2: Good
Enter name for customer 3: Tommy
Enter rating for customer 3 (1-5): 4
Enter feedback for customer 3: Good
Enter name for customer 4: Jerry
Enter rating for customer 4 (1-5): 1
Enter feedback for customer 4: Bad
Enter name for customer 5: Donald
Enter rating for customer 5 (1-5): 3
Enter feedback for customer 5: Fair
Feedback has been saved to customer_feedback.csv
```

Analyzing Feedback

```
Customer Name  Rating  Feedback
0           Eva         1      Poor
1        Minnie         5      Good
2         Tommy         4      Good
3         Jerry         1      Bad
4        Donald         3      Fair
class pandas.core.frame.DataFrame:
```

Technologies Used

- **Python 3.x:** Core programming language used for collecting, analyzing, and saving feedback.
- **Pandas:** Python library used to handle CSV file operations, making it easy to store and retrieve feedback data.

Improvements for Future Versions

- **Advanced Sentiment Analysis:**
 - Implement more sophisticated Natural Language Processing (NLP) techniques to provide a more nuanced analysis of feedback sentiment.
- **Feedback Visualization:**
 - Add data visualization tools such as graphs and charts to illustrate trends in feedback, including the number of positive versus negative feedback entries.
- **Web or Mobile Interface:**
 - Develop a web or mobile interface using Flask or Django to make the system more accessible and user-friendly.
- **Persistent Storage:**
 - Store the feedback in a database (e.g., SQLite or MySQL) to allow for long-term storage and more efficient querying.

Conclusion

The **Customer Feedback Management System** is a useful tool for businesses to collect, store, and analyze customer feedback in an organized and efficient manner. The system demonstrates the power of Python for data collection and analysis, providing a solid foundation for expanding into more complex feedback analysis and reporting systems.

This project serves as an excellent starting point for learning how to build data-driven systems, and it can be extended with additional features and improvements to meet more sophisticated business needs.