

1. Fitness Coach Agent

Scenario: A user inputs their age, weight, and fitness goal (e.g., lose weight, gain muscle). The agent provides a personalized workout and diet plan.

Objectives:

- Offer tailored fitness recommendations based on user profiles.
- Adapt plans over time as user progress is tracked.

Requirements:

- User input collection for age, weight, height, gender, and fitness goals.
- Machine learning model to generate personalized plans.
- Database to store user profiles and progress.
- User interface for data input and plan display.

Deliverables:

- Data collection forms for user profiles.
- Trained machine learning model for recommendations.
- Dashboard displaying workout and diet plans.
- Progress tracking and plan adjustment mechanism.

Bonus Features:

1. Export the trained model to a `.pkl` file.
2. Develop a Streamlit interface for user interaction.
3. Deploy the application on the Streamlit Community Cloud.

Datasets:

- [Gym Recommendation Dataset](#)
 - [Diet Recommendations Dataset](#)
-

2. Email Spam Detector

Scenario: An incoming email is automatically classified as "Spam" or "Not Spam" and directed to the appropriate folder.

Objectives:

- Automate email classification to filter out spam.
- Improve accuracy over time with model training.

Requirements:

- Preprocessing of email text data.
- Feature extraction from email content.
- Machine learning model for classification.
- Evaluation metrics for model performance.

Deliverables:

- Cleaned and labeled email dataset.
- Trained classification model.
- Performance evaluation report.
- Integration script for email systems.

Bonus Features:

1. Export the trained model to a `.pk1` file.

2. Create a Streamlit interface to input email text and display classification results.
3. Deploy the application on the Streamlit Community Cloud.

Datasets:

- [Spambase Dataset](#)
 - [Enron Spam Dataset](#)
-

3. Virtual Travel Agent

Scenario: A user specifies their preferred travel season, budget, and interests. The agent recommends suitable travel destinations.

Objectives:

- Provide personalized travel destination recommendations.
- Enhance user experience by aligning suggestions with preferences.

Requirements:

- User input collection for travel preferences.
- Database of destinations with attributes like climate, cost, and attractions.
- Recommendation algorithm matching user preferences to destinations.
- Feedback mechanism to refine recommendations.

Deliverables:

- User input forms for travel preferences.
- Destination recommendation engine.
- Detailed destination profiles.

- Feedback collection and analysis module.

Bonus Features:

1. Export the recommendation model to a `.pkl` file.
2. Develop a Streamlit interface for user interaction.
3. Deploy the application on the Streamlit Community Cloud.

Datasets:

- [Travel Recommendation Dataset](#)
 - [Tourism DatasetKaggle](#)
-

4. Shopping Recommendation Agent

Scenario: A user browses an online store, and the agent suggests products based on their shopping behavior and preferences.

Objectives:

- Enhance user shopping experience through personalized recommendations.
- Increase sales by suggesting relevant products.

Requirements:

- Tracking of user interactions and purchase history.
- Product catalog with detailed information.
- Recommendation algorithms (collaborative and content-based filtering).
- Personalization based on user profiles.

Deliverables:

- User behavior tracking module.
- Recommendation engine.
- Integration with user interface to display suggestions.
- Analytics dashboard for performance monitoring.

Bonus Features:

1. Export the recommendation model to a `.pkl` file.
2. Create a Streamlit interface for product recommendations.
3. Deploy the application on the Streamlit Community Cloud.

Datasets:

- [E-commerce Behavior Data](#)
- [Online Retail Dataset](#)