steam\_sentimentanalysis/

│

├── app/ # Subfolder for app-related files

│ ├── static/ # Folder for static files (CSS, JS, images, etc.)

│ ├── templates/ # Folder for HTML templates

│ ├── utils/ # Utility scripts and files

│ │ ├── steam\_scrape.py # Script for scraping Steam reviews

│ │ └── tokenize\_csv.py # Script for tokenizing and lemmatizing review data

│ ├── \_\_init\_\_.py # Initialization file for the app package

│ └── routes.py # Python script defining routes for your app

│

├── data/ # Subfolder for data files

│ ├── steam\_reviews.csv # CSV file with original steam reviews

│ ├── train\_reviews.csv # CSV file for training data

│ └── test\_reviews.csv # CSV file for testing data

│

├── models/ # Subfolder for model scripts

│ ├── hier\_cluster.py # Python script for hierarchical clustering model

│ ├── kmeans\_cluster.py # Python script for K-Means clustering model

│ ├── LDA\_model.py # Python script for LDA topic modeling

│ └── train\_model.py # Script for training models

│

├── tests/ # Subfolder for test scripts

│

├── venv/ # Virtual environment folder

│

├── .gitignore # Specifies intentionally untracked files to ignore

├── README.md # Markdown file with project description

├── requirements.txt # List of dependencies for the project

├── run.py # Main script to run the application

└── setup.py # Setup script for the project

**Project Summary So Far:**

1. **Data Collection (steam\_scrape.py)**:
   * You've created a script to scrape Steam reviews. This script collects user reviews from Steam games, providing a dataset for your analysis.
2. **Data Preprocessing and Tokenization (tokenize\_csv.py)**:
   * This script processes the scraped data by tokenizing and applying lemmatization to the review text. The goal is to prepare the text data for further analysis by breaking it down into more manageable and meaningful components.
3. **Training and Test Datasets**:
   * The data collected and processed are split into training and testing datasets, which are stored in the **data** folder. This split is crucial for training your models and evaluating their performance.
4. **Model Selection and Scripts**:
   * You've chosen three types of unsupervised learning models for sentiment analysis: Hierarchical Clustering, K-Means Clustering, and LDA (Latent Dirichlet Allocation) for Topic Modeling. Each of these models will offer a different perspective on the sentiment and themes within the Steam reviews.
   * Corresponding scripts for each model are placed in the **models** folder.
5. **Web Application Setup**:
   * Your project includes an **app** folder with subfolders for **static**, **templates**, and **utils**, which suggests you're setting up a web application to display the results of your sentiment analysis.
   * The **routes.py** file indicates the configuration of different routes or endpoints for your web app.
6. **Project Infrastructure**:
   * Basic project setup files like **run.py**, **setup.py**, **requirements.txt**, and **.gitignore** are in place, indicating a well-structured and manageable project environment.

### Project Summary: Steam Sentiment Analysis

#### **Overview**

Your project involves developing an advanced sentiment analysis tool specifically tailored for Steam game reviews. The goal is to analyze these reviews to extract meaningful insights about user sentiments, preferences, and opinions regarding various games available on Steam. This comprehensive analysis aims to offer valuable information that could benefit both game developers and gamers.

#### **Objectives**

1. **Understand User Sentiment**: Determine how users feel about different games – which aspects they enjoy, what issues they face, and their overall satisfaction.
2. **Identify Common Themes**: Uncover recurring topics or themes in reviews, such as gameplay, graphics quality, storyline, customer service, or technical issues.
3. **Facilitate Informed Decisions**: Provide game developers with actionable insights to improve game design and marketing strategies. Also, assist potential buyers in making informed choices based on community feedback.

#### **Methodology**

1. **Data Collection**: Automated scraping of Steam reviews to build a rich dataset of user opinions.
2. **Data Preprocessing and Tokenization**: Processing the raw review text through tokenization and lemmatization to make it suitable for analysis. This step involves breaking down the text into basic elements and standardizing the language used.
3. **Model Development and Training**: Implementing unsupervised machine learning models to analyze the processed text. The project utilizes three different models:
   * Hierarchical Clustering: To explore detailed relationships and sentiment structures in the data.
   * K-Means Clustering: For general grouping and categorization of reviews.
   * Latent Dirichlet Allocation (LDA): To discover latent topics and themes within the reviews.
4. **Result Visualization and Analysis**: The findings from these models will be analyzed and potentially displayed through a web application, enabling easy access and interpretation of the results.
5. **Testing and Evaluation**: Using a separate test dataset to evaluate the effectiveness and accuracy of the models in real-world scenarios.

#### **Expected Outcomes**

* **Enhanced User Understanding**: Gain in-depth insights into user satisfaction and preferences.
* **Game Improvement**: Direct feedback to game developers for improving game features and addressing user concerns.
* **Market Research**: Valuable data for understanding market trends and user expectations in the gaming industry.

#### **Technical Infrastructure**

* The project is structured with separate folders for application components (**app**), data (**data**), models (**models**), and tests (**tests**).
* Key scripts for data scraping, preprocessing, and model training are organized within the relevant subfolders.
* A virtual environment (**venv**) is used to manage dependencies and ensure consistency across different development setups.

### Conclusion

This project represents a comprehensive effort to harness the power of sentiment analysis and machine learning for deriving actionable insights from Steam game reviews. Through its methodical approach, it aims to bridge the gap between user feedback and game development, ultimately enhancing the gaming experience and informing decision-making in the gaming industry.

Top of Form