AI-Driven Customer Feedback Analysis for Wex Photo Video

Aman Seth

August 29, 2024

1 Introduction

This project is designed to perform AI-driven customer feedback analysis specifically for Wex Photo Video. The project leverages advanced natural language processing techniques using BERT for sentiment analysis on customer reviews.

2 Prerequisites

Ensure you have the following prerequisites installed on your system before proceeding:

- Python 3.7 or newer version
- Visual Studio Code (Community Edition)

3 Setup Instructions

3.1 Step 1: Clone the Git Repository and Load the Project

1. Clone the Git repository by running the following command:

```
git clone https://github.com/amansethuea/WexPhotoVideo.git
```

2. Load the cloned folder in Visual Studio Code (Community Edition).
3. Open a new terminal in Visual Studio Code.
4. Create a new virtual environment by running the following commands:

```
python -m venv <name_of_env>  % Example: python -m venv venv
venv\Scripts\activate
```

Note: Creating a virtual environment is a one-time activity. You only need to activate the virtual environment each time you work on the project by running:

```
venv\Scripts\activate
```

3.2 Step 2: Install Required Python Modules

After activating the virtual environment, install the required Python modules by running the following commands:

```
pip install matplotlib==3.8.2
pip install matplotlib-inline==0.1.7
pip install scikit-learn==1.4.2
pip install pandas==2.2.2
pip install plotly==5.22.0
pip install contractions==0.1.73
pip install demoji==1.1.0
pip install spacy==3.7.4
pip install streamlit==1.36.0
pip install ollama==0.3.1
pip install kaleido==0.2.1
pip install transformers==4.42.3
pip install torch==2.3.1
pip install dash==2.17.1
```

Note: Installing these modules is a one-time activity and does not need to be repeated each time, as long as you activate the same virtual environment.

4 Running the BERT Dashboard

1. After setting up the environment, navigate to the `BERT` folder and run the following command to start the dashboard:

   ```
   python bert_dashboard.py
   ```

2. The output should be something like:

   ```
   Dash is running on http://127.0.0.1:8050/

   2024-08-29 13:52:27,513 - INFO - Dash is running on http
       ://127.0.0.1:8050/

    * Serving Flask app 'bert_dashboard'
    * Debug mode: on
   ```

3. Click on the link `http://127.0.0.1:8050/` to open the web app in your default web browser (e.g., Google Chrome).

5 Final Note

Important: Since GitHub's basic version does not allow files larger than 100 MB to be pushed, the DistilBERT best model file isn't part of this repository.