Course Project Documentation
Team: Mango

Project Coordinator: Saket Vissapragada

Team Members:

Ameya Gharpure (ameyapg2@illinois.edu)

Sruthi Kilari (skilari2@illinois.edu)

Saket Vissapragada (saketv2@illinois.edu),

Shreya Sharma (ssharm90@illinois.edu)

Amrith Balachander (amrithb2@illinois.edu)

Link to Software Usage Tutorial:
https://docs.google.com/presentation/d/1FlEPT93hch7turiqAhhT4gqFaEBDUutnAeeuTLnThTE/edit?usp=sharing

Link to video: https://youtu.be/eUvOhQ5dF6U

1) An overview of the function of the code (i.e., what it does and what it can be used for).

The function of the code was to provide a standardized ranking system for google reviews (1-5 stars). First, the user launches the chrome extension. Then the user can input text into the text field. Finally, the user can click the "OK" button to receive the standardized review rating. In the background, the chrome extension uses sentiment analysis to give an unbiased and standardized ranking to all the text-based reviews. To create our sentiment analysis application, we made use of machine learning algorithms. First, we needed to preprocess our data (tokenization, converting text to lowercase, removing punctuation, removing numbers, removing stop words, and normalization). After preprocessing text data, we sent the textual data into a machine learning pipeline (bag-of-words, TFIDF, Naive Bayes) for the machine to be able to understand what the text means and train the model.

2) Documentation of how the software is implemented with sufficient detail so that others can have a basic understanding of your code for future extension or any further improvement.

1. Flask App
   a. JSON endpoint
      i. Create a JSON endpoint that can receive POST requests from the chrome extension
   b. Run Model
      i. Import the model and run the model on the given review
   c. Send value back to Extension
      i. Return the value back to the Chrome extension
2. Chrome Extension
   a. Popup
      i. In the *background* Javascript file, we implemented an onClick listener that launches a textbox prompt when the user clicks on the Chrome extension's icon.
   b. Post JSON to FlaskApp

      i.   Create a JSON object that has the data from the popup

      ii.   Send the POST request to the Flask server that contains the JSON object

  c. Display

      i.   After the model runs on the inputted data, the Chrome extensions alerts the user of the determined standardized review rating as can be seen in the *post* Javascript file.

3. Model
   a. Preprocessing
      i. Tokenization
      ii. Removing stopwords
      iii. Punctuation removal
      iv. Number removal
      v. Stop words removal
      vi. Normalization via lemmatization or stemming
   b. Machine Learning Pipeline
      i. Bag of Words
      ii. TF-IDF
      iii. Naive Bayes

3) Documentation of the usage of the software including either documentation of usages of APIs or detailed instructions on how to install and run a software, whichever is applicable.

We utilized Flask to run a backend application for our Chrome extension. The documentation to install Flask is found in https://flask.palletsprojects.com/en/2.0.x/installation/#.

To summarize…

1. Open up a terminal and clone the repository using this link: https://github.com/saketv2/CourseProject.git and navigate inside this directory
2. Place the pipeline.pkl file from https://drive.google.com/drive/folders/1hx66MiifLZFUbvnJv5_HuQMYRdQx6viI?usp=sharing inside the model directory
3. Activate a virtual environment
   a. python3 -m venv venv
   b. source venv/bin/activate
4. Install Flask
   a. pip install Flask
5. Install requirements from requirements.txt
   a. pip install -r requirements.txt
6. Run the Flask app
   a. $ export FLASK_APP=app
   b. $ flask run

In order to install our Chrome extension in the browser, a user must navigate to the *Manage extensions* page of Chrome and turn on *developer mode*. From there, you can select *Load unpacked* and select Restaurant Reviewer from the project's directory. This will install the Chrome extension on your browser. Now, when you click on the extension's icon in the browser menu, you will be able to see the popup window to enter a review.

4) Brief description of contribution of each team member in case of a multi-person team.

Amrith

- Project Planning and Task Assignment
    - Developed detailed diagrams about the workflow
    - Scheduled and integrated different stories from the scrum board
- Chrome extension
    - Created the JSON Post Request to send to flask app
- Apply algorithms covered in class to generate sentiment analysis scores for each review based on commonly used words
    - Train the model using previous data from Kaggle dataset
- Verified standardized star ratings
- User testing to determine accuracy of standardized scores
    - Presented project to 8 users and had them test with their own inputted reviews to determine the accuracy of the model

Ameya

- Project Planning and Task Assignment
    - Scheduled tasks and organized group meetings
- Chrome extension
    - Worked on alerting user of result from model
- Apply algorithms covered in class to generate sentiment analysis scores for each review based on commonly used words
    - Applied TF-IDF, Naive Bayes algorithms on cleaned data
- Rank these scores against each other to determine standardized star ratings and compute an average standardized score for restaurants.
- User testing to determine accuracy of standardized scores
    - Presented project to 8 users and had them test with their own inputted reviews to determine the accuracy of the model

Sruthi

- Project Planning and Task Assignment
    - Organized and updated tasks in scrum board
- Chrome extension
    - Implemented pop-up box to allow users to input text that is sent to the model
- Apply algorithms covered in class to generate sentiment analysis scores for each review based on commonly used words
    - Preprocessed training data obtained from Kaggle dataset
- Verified standardized star ratings
- User testing to determine accuracy of standardized scores
    - Presented project to 8 users and had them test with their own inputted reviews to determine the accuracy of the model

Saket

- Project Planning and Task Assignment
    - Created and managed scrum board to assign group tasks.

- Chrome Extension
    - Worked on testing, edge-case detection, and user testing for the usability of the chrome extension.
- Apply algorithms covered in class to generate sentiment analysis scores for each review based on commonly used words
    - Developed the ML pipeline for training the model.
- Determine standardized star ratings and compute an average standardized score for restaurants.
- User testing to determine accuracy of standardized scores
    - Presented project to 8 users and had them test with their own inputted reviews to determine the accuracy of the model

Shreya

- Project Planning and Task Assignment
    - Managed scrum board and organized and scheduled tasks
- Chrome Extension
    - Made json object with inputted text from review
- Apply algorithms covered in class to generate sentiment analysis scores for each review based on commonly used words
    - Preprocessed training data and outputted to model csv
- Determine standardized star ratings and compute an average standardized score for restaurants.
- User testing to determine accuracy of standardized scores
    - Presented project to 8 users and had them test with their own inputted reviews to determine the accuracy of the model

---

In our original project proposal, we had planned to create a webscraper to scrape review data from the web that would be used as training data from the model. We developed this webscraper (located in the project directory) but we noticed that the data would require large amounts of preprocessing. Therefore, we decided to obtain the dataset from Kaggle instead.