Data collection

A screenshot of a computer program

Description automatically generated

The purpose of this code is to demonstrate how to use Selenium to scrape data from a Reddit thread, specifically extracting comments and sub-comments.

Selenium is a powerful tool used for automating web browsers. Originally designed for testing web applications, it has become widely used for web scraping due to its ability to interact with web pages just like a human user would. This makes it particularly effective for scraping data from dynamic, JavaScript-heavy websites.

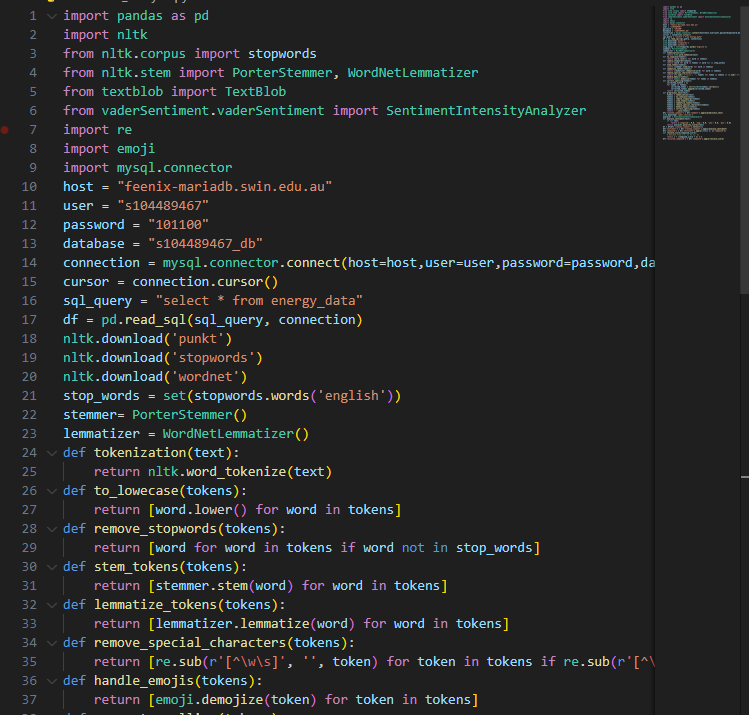
To do this step we need to have Python, Selenium library, compatible web driver and added to system’s path.

|  |  |
| --- | --- |
| Step | Code demonstration |
| Import necessary libraries | Imports selenium for web automation, time for delays, pandas for data manipulation, and expected\_conditions for waiting for elements to be visible. |
| Set up WebDriver | Initializes a Chrome WebDriver instance using webdriver.Chrome(). |
| Navigate to the target subreddit | Navigates to the specific Reddit post URL using browser.get(). |
| Find and extract post title and content | Finds elements for the post title and content using CSS selectors and XPaths.  Extracts the text from these elements using element.text. |
| Handle potential comment loading | Uses WebDriverWait to wait for a specific element to be clickable (the "Load more comments" button).  If found, clicks the button to load additional comments. |
| Extract comments and sub-comments | Finds all comment elements using find\_elements\_by\_xpath.  Iterates through each comment element:   * Finds sub-comment elements within the comment using find\_elements\_by\_tag\_name * Extracts the text from each sub-comment and appends it to a list. * Appends the combined sub-comments to the main comments list. |
| Create dataframe and save it to csv file | Creates a Pandas DataFrame from the extracted comments.  Saves the DataFrame to a CSV file named "reddit\_comments\_windpower.csv". |
| Close the browser | Closes the WebDriver instance to release resources. |

Output

The code will output a DataFrame containing the extracted comments and sub-comments from the Reddit platform.

Data preprocessing and data modelling



A screen shot of a computer program

Description automatically generated

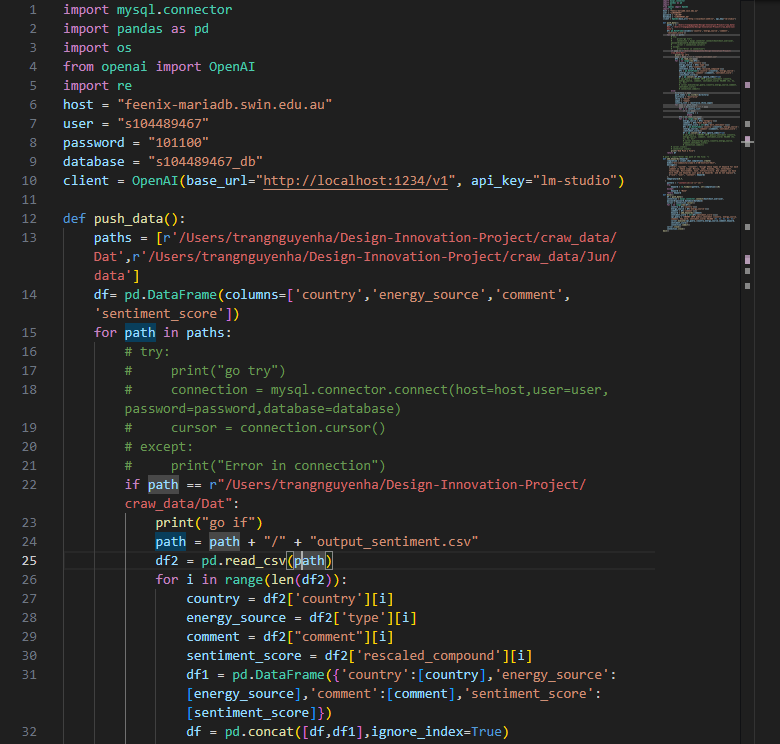
The purpose of this code is to demonstrate text preprocessing and sentiment analysis techniques using python libraries.

To do this step we need to have python, nltk, textblob, vaderSentiment emoji and mysql-connector-python

There are 10 steps

|  |  |
| --- | --- |
| Step | Code demonstration |
| Import necessary libraries | Imports pandas, mysql.connector, nltk, and SentimentIntensityAnalyzer for data manipulation, database connection, text processing, and sentiment analysis. |
| Connect to the MySQL database | Establishes a connection to your MySQL database using the provided credentials. |
| Retrieve data from the database | Executes a SQL query to select the comment column from your desired table and stores the results in a list. |
| Download NLTK resources | Downloads required NLTK resources for text processing and sentiment analysis. |
| Handle potential comment loading | Uses WebDriverWait to wait for a specific element to be clickable (the "Load more comments" button).  If found, clicks the button to load additional comments. |
| Extract comments and sub-comments | Finds all comment elements using find\_elements\_by\_xpath.  Iterates through each comment element:   * Finds sub-comment elements within the comment using find\_elements\_by\_tag\_name * Extracts the text from each sub-comment and appends it to a list. * Appends the combined sub-comments to the main comments list. |
| Create dataframe and save it to csv file | Creates a Pandas DataFrame from the extracted comments.  Saves the DataFrame to a CSV file named "reddit\_comments\_windpower.csv". |
| Close the browser | Closes the WebDriver instance to release resources. |
| Define text preprocessing functions | Tokenize splits the text into individual words  to\_lowercase converts all words to lowercase for consistent comparison.  Remove\_stopwords removes common words like "the," "and," "a" that don't add much meaning.  stem\_tokens reduces words to their root form (e.g., "running" becomes "run").  lemmatize\_tokens finds the base form of a word (e.g., "better" becomes "good").  remove\_special\_characters removes characters like punctuation and symbols.  handle\_emojis converts emojis to their textual representation (e.g., "👍" becomes "thumbs up").  remove\_numbers removes numeric values.  remove\_short\_words removes words that are too short to be meaningful. |
| Preprocess text and analyze sentiment by using VADER | Applies the preprocess\_text function to clean the comments and the sia.polarity\_scores() function to analyze sentiment using VADER. Stores the results in new columns sentiment and rescaled\_compound. |
| Print or analyze the results | Prints the DataFrame to view the original comments, cleaned comments, sentiment scores, and rescaled compound scores. |
| Close the database connection | Closes the database connection to release resources. |

Data Pushing to data warehouse



A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

The purpose of this code is to process comments by using an LLM model to generate keywords based on the comments, and store data to MySQL database.

To run this code we need to have Python environment, pandas, mysql-connector-python and openai. OpenAI API key obtain an API key from OpenAI to access their LLM models.

There are 5 steps:

|  |  |
| --- | --- |
| Step | Code demonstration |
| Import necessary libraries | Imports pandas, mysql.connector, openai, nltk, and SentimentIntensityAnalyzer for data manipulation, database connection, text processing, keyword generation, and sentiment analysis. |
| Set up database connection | Establishes a connection to your MySQL database using the provided credentials. |
| Load data from CSV | Reads the CSV file containing your data into a Pandas DataFrame. |
| Process data and generate keywords | Applies text preprocessing steps (e.g., tokenization, stemming, lemmatization, stop word removal) to clean the text.  Uses the OpenAI API to generate keywords based on the cleaned text.  Analyzes sentiment using VADER and stores the results in a new column. |
| Close database connections | Closes the database connection to release resources. |