

Project Proposal: Commodity Trading Ranker System

Team Members:

Hanlin Wang - hanlinw4@illinois.edu
Ruide Zhu - ruidezhu2000@gmail.com
Haotian Wang - hw55@illinois.edu
Huiyang Chi - huiyang_chi@outlook.com
Mengchen Chou - mcqiu3@gmail.com

Task Introduction:

For the free topic project, our team proposed a commodity trading system that allows users to find the corresponding product accurately and efficiently through the keywords or short descriptions that the user input. The system will simulate real online trading market conditions by extracting data from the posts in several selected commodities trading Facebook groups. The tasks include but are not limited to gathering data, cleaning data, and modeling data by building a web application to present our data, etc.

The importance of the project:

Environment: Not only does buying used reduce the number of natural resources being used, but it also reduces the amount of energy used and pollution that's being emitted. Things like pesticides, burning fuel in the trucks that haul the items, toxic chemicals, and carbon emissions.

Save money: One of the most obvious and well-known benefits of buying secondhand is cost savings. You can often find secondhand goods up to 50% cheaper than you could if you were buying new. When you consider that Americans spend over a trillion dollars annually on nonessential goods, those savings can add up.

To search and find the good we want, we need to filter the output from keywords inputted by users. And we should apply our filtering method to determine the items that are displayed to users and ensure they are correlated.

What techniques is the project based on:

We mainly will use python for the infrastructure under the hood. For the related package we will use our flask, flask, redirect, url_for, render_template, request, session, abort, make_response, metapy, pymongo, pprint, pymongo, Django, and random. We'll render our web page using vanilla HTML and CSS.

What is the expected outcome:

So after finishing the project, we'll get a web page containing classified information on products sold on Facebook. By diving into each sub-page, we can get the details of that specific kind of product and related descriptions. Taking our project as an example of a folder, each deeper layer of the folder shows a more specific category of products. What is the planned approach:

Our approach is to break the problem down into gathering data, cleaning data, modeling data as well as using a web application to present our data.

In the process of data gathering, we continue to break the problem down into gathering data for two purposes, which are data preprocessing for classifying tasks and data for ranking tasks. For data for classification, we decided to give Meal DB's API to give it a try, and use GraphAPI to extract data for ranking and store it in MongoDB. As for data cleaning, we exclude data by specific criteria like the message that is too short to make more sense to our project. Then, for modeling, We use Metapy as the implementation tool. Finally, regarding the web application, we decide to use Flask to build our search engine.

How to evaluate the project:

To evaluate the effectiveness of the project, we can compare the result with the expected outcome evaluated by a human. More specifically, users can test the classifier function of products by going to the "classifier" page then the user can input a product name to test the classifier. For example, "LEGO" should be classified as a toy, and "Bottle" should be identified as household goods. If the output matched with expected result, then the project can be classified as efficient.