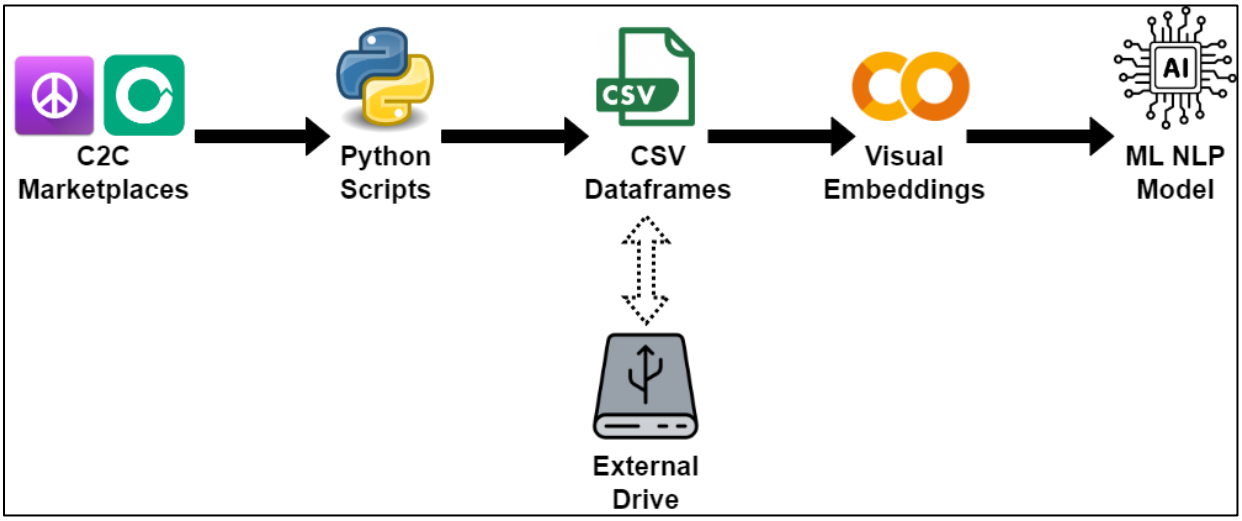
Training AI on Stolen Car Parts

Developer Documentation

Harmond Drenth, Kurt Wokoek, Warren Burrus, Courtney Hodge, Denny Lee, Tacoma Velez

Architecture Design



Risk Score Formula

- Evaluates the suspiciousness of car posts on a scale from 0.0 (low) to 1.0 (high)

$$Score = 0.25\left(\frac{|P_{Market} - P_{Post}|}{P_{Market}}\right) + 0.10\left(1 - \frac{1}{R+1}\right) + 0.15\left(1 - \frac{1}{K+1}\right) + 0.20\left(\frac{N+P}{2}\right)$$

  - **P_{Market}** = average market price of car part
  - **P_{Post}** = selling price of car part in post
  - **R** = number of **R**ed flags in photos (tags, opened packages, etc.)
  - **K** = number of suspicious **K**eywords, phrases and terms in description
  - **N** = binary 0 or 1 if **N**ame is standard first and last name or not
  - **P** = binary 0 or 1 if **P**hone number is standard number format or not

$$+ 0.10\left(1 - \frac{1}{D+1}\right) + 0.10\left(\frac{1}{\Delta T + 1}\right) + 0.10\left(\frac{100 - r}{100}\right)$$

  - **D** = number of **D**uplicate posts with same content
  - **ΔT** = difference in days between **T**imestamps of two posts

- **r** = mile **r**adius from high-crime city (0 <= r <= 100)

API Specifications

- raw_process.py
  - Process the JSON files and create the data frame/csv files
- CarParts_Text_Processing.ipynb
  - Get data from the csv files
  - Produce the visualization and perform data analysis in the csv files

Prerequisites

- You will need a Google Colab account to run the provided scripts
- You will need to upload the required files to the Google Colab runtime

Risk Score Testing in Colab

The risk score formula and weights can be tested on a given set of post data in Colab. First, manually enter the post data into the arrays storing metadata on the posts. Each post corresponds to a single index across all arrays.

```
# MANUALLY FILL WITH INPUT DATA FROM EACH POST!!!
marketPrices    = [350, 350, 350, 350, 350, 350, 350, 1000, 1000]
sellerPrices    = [65, 65, 65, 300, 125, 299, 350, 1000, 1000]
redFlagWords    = [0, 0, 0, 1, 4, 0, 1, 1, 1]
oddNamesPhones  = [1, 1, 1, 1, 1, 0, 0, 2, 2]
duplicates      = [2, 2, 2, 0, 0, 296, 0, 1, 1]
crimeCityRanges = [35, 35, 35, 100, 27, 0, 0, 100, 100]
monthDayYrDates = ["8/11/2023", "7/8/2023", "6/4/2023",
                   "7/23/2023", "7/13/2023", "6/28/2023",
                   "9/12/2023", "8/16/2023", "5/18/2023"]
timestamps = datesToTimestamps(monthDayYrDates)
```

- oddNamesPhones[] - 0 means neither phone nor name is suspicious; 1 means only one of phone or name is suspicious; 2 means both phone and name are suspicious
- duplicates[] - number of duplicate posts, excluding self

- crimeCityRanges[] - if more than 100 miles from crime city, enter 100 miles

After manual data entry, run the statistical analysis in Colab.

Chrome Extension

- Canceled for this research project
    - If the extension produced a false positive post score that discouraged customers from buying a car part, vendors (Craigslist, OfferUp) could press charges for loss of business.