CIS434-SB

SOCIAL MEDIA AND TEXT ANALYTICS

SPRING 2025

SMART CATEGORIZATION OF CRAIGLIST

Group 9:

Shan Ali Shah Sayed
Jiahai Chen
Muskan Hisaria

Date: May 2nd, 2025

Table of Contents

1. Background

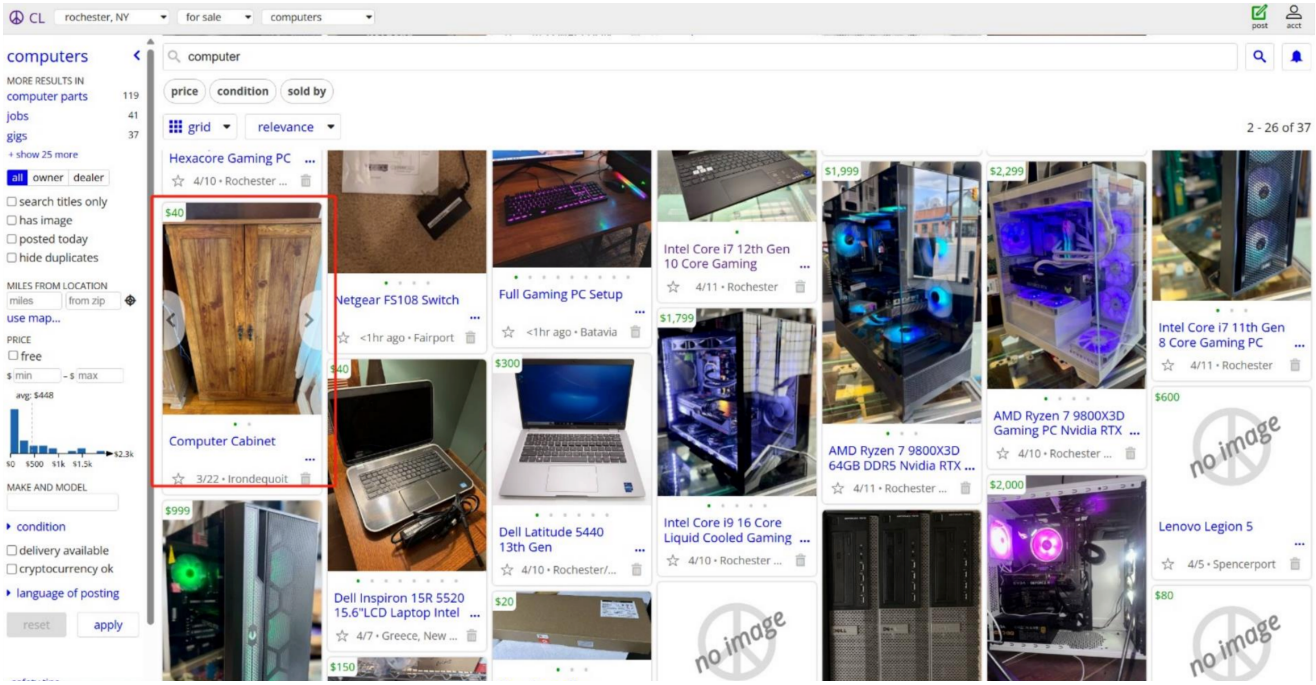
Client Context

Craigslist is a dominant online classifieds platform operating across hundreds of local subdomains in the United States and internationally. It allows users to post, browse, and respond to a wide variety of advertisements covering services, job postings, housing, gigs, and items for sale. Despite its reach and popularity, Craigslist's minimalist user interface and lack of structured data management introduce several usability issues. Categories are broad, there is limited use of metadata, and ads often include inconsistent, unmoderated text, which affects both search precision and user experience.

Craigslist's current design encourages flexibility, allowing users to freely post and categorize their own listings. However, this comes at the expense of accuracy. One area especially prone to error is the "For Sale" section, and more specifically, the closely related categories "Computers" and "Computer Parts".

Problem Focus

Within the Rochester, NY Craigslist region, we identified rampant misclassification between the "Computers" category - meant for full computer systems - and the "Computer Parts" category - meant for individual components or peripherals. For example, it is common to see keyboards, monitors, and power supplies misposted under "Computers".



This misclassification leads to several operational and experiential challenges:

- **Decreased User Trust**: Shoppers browsing for complete systems encounter irrelevant posts.
- **Increased Cognitive Load**: Users must sift through large volumes of noise.
- **Manual Moderation Costs**: Craigslist staff must identify and recategorize listings manually.

We estimate that at least **~30%** of the posts in either category are misclassified. Addressing this issue with an intelligent solution can save time, enhance user satisfaction, and improve the quality of listings on the platform.

Client Need

Craigslist would benefit from a scalable, automated solution that leverages modern natural language processing (NLP) techniques to assist in **accurate, text-based categorization** of listings. Given that Craigslist receives millions of monthly posts, such a system must be interpretable, lightweight, and able to generalize to new listings without retraining on a massive corpus.

2. Business Analysis

Primary Objective

Our team aimed to design and evaluate an NLP-based classification system capable of identifying whether a given Craigslist post in the Rochester region belongs in the "Computers" or "Computer Parts" category. This classifier would use only the **free-text content** (title and description) of each listing as input.

Key Business Goals

- **Reduce Misclassification Rate**: Achieve over 85% classification accuracy.
- **Flag Suspicious Listings**: Identify listings that are likely miscategorized.
- **Generalizable Design**: Ensure that the framework is replicable across other regions and categories.

Practical Outcomes

If implemented by Craigslist, this tool could:

- **Improve Search Results**: Users will find what they're looking for faster.
- **Reduce Moderator Burden**: Human moderators can focus on edge cases.
- **Lay Foundation for Broader Automation**: The same logic can be extended to Jobs, Cars, Housing, Services, etc.

Constraints

Our project focused on the **text analytics and classification modeling** only. We did not explore real-time deployment or front-end integration with the Craigslist UI.

3. Data Analysis

Data Collection

We used **Selenium and BeautifulSoup** in Python to scrape postings from the "Computers" and "Computer Parts" categories on Craigslist's Rochester, NY domain. We followed scraping practices and ensured we did not overload the server.

- **Computers**: 400 listings
- **Computer Parts**: 342 listings
- **Total scraped**: 742 listings

For each listing, we extracted:

- **Title**
- **Description**
- **Listing URL**
- **Original category**

These were saved in two CSVs: computers.csv and computer_parts.csv.

Preprocessing

To prepare the dataset for analysis, we first merged the scraped data from both categories - Computers and Computer Parts - into a unified DataFrame. Each entry was assigned two labels: a label based on the original Craigslist category it was scraped from, and a human_label, which was manually verified to correct any misclassifications and ensure data quality.

Listings missing either a title or description were dropped to maintain textual completeness. We then concatenated the title and description fields into a single unified text field for modeling. This combined text was lowercased, stripped of leading/trailing whitespace, and cleaned of non-alphanumeric characters to reduce noise. We also removed duplicate listings by comparing the cleaned text field.

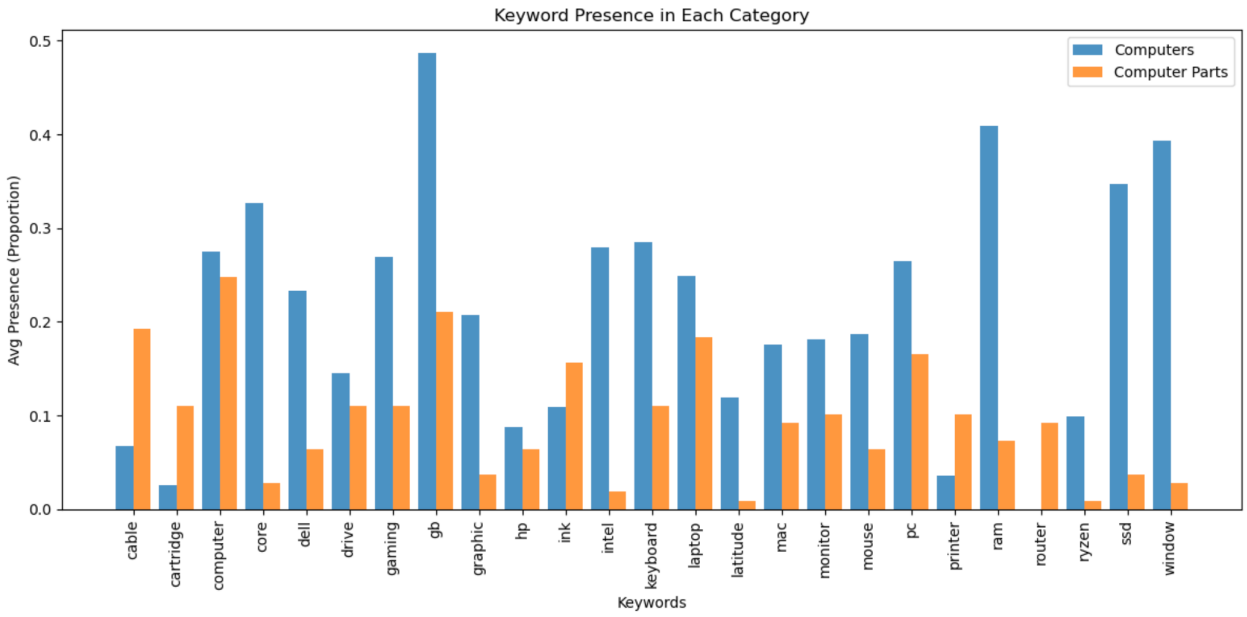
This well-structured dataset served as the foundation for feature extraction and model training.

Feature Engineering

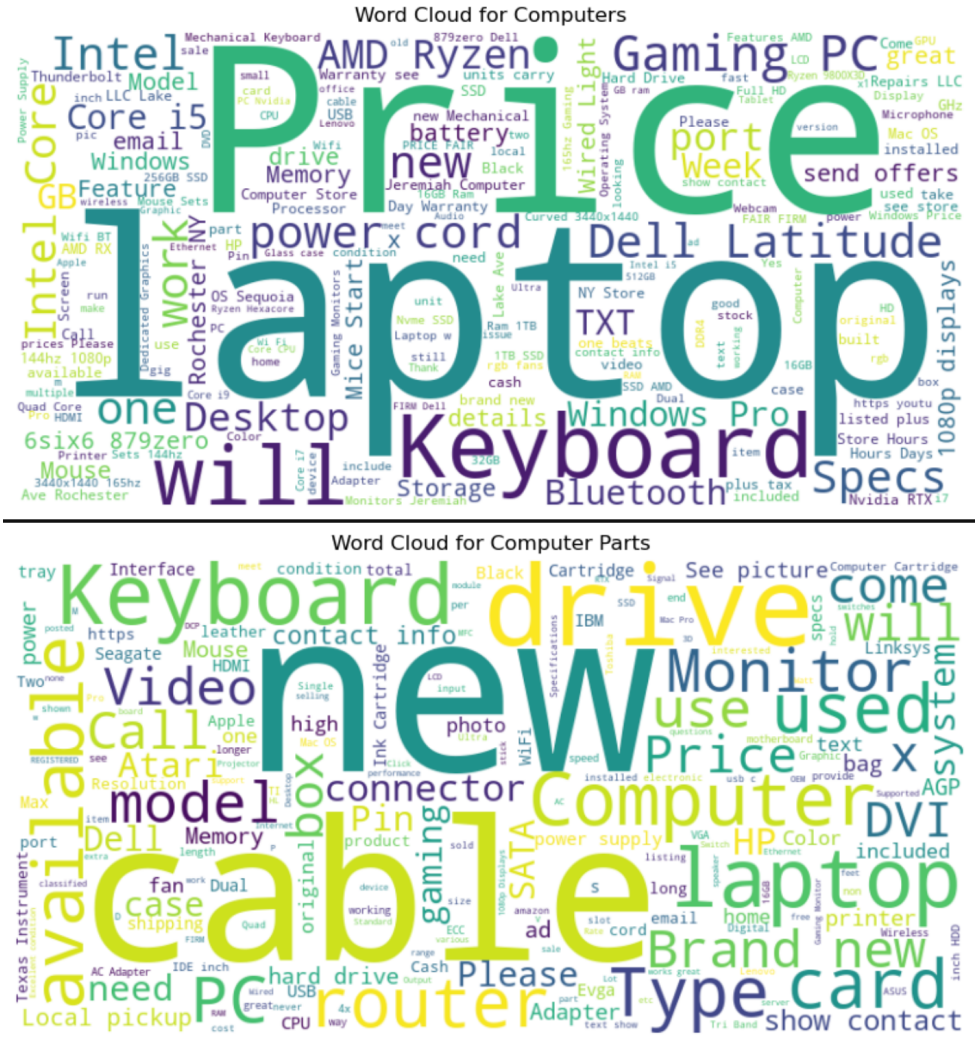
To analyze and quantify the importance of words across ad listings, we applied **TF-IDF (Term Frequency-Inverse Document Frequency) vectorization**. We used scikit-learn's TfidfVectorizer with a cap of 3,000 features and integrated built-in English stopword removal to eliminate common, uninformative words. The tokenizer was configured to extract both unigrams and bigrams, enabling the model to capture not only individual words but also meaningful two-word combinations (e.g., "hard drive", "graphics card").

We computed separate TF-IDF scores for each class (Computers and Computer Parts) to identify the most influential keywords per category. These keyword importance values helped us understand class-specific language patterns and informed both manual feature engineering and exploratory analysis. The most impactful terms were visualized using **bar charts and word clouds** as shown below, offering an intuitive view of the linguistic distinction between product listings.

Keyword Bar chart:



Word Cloud:



Building upon insights from the TF-IDF analysis, we curated a focused list of over 25 domain-relevant keywords that were strongly associated with each category. For the "Computers" class, keywords included terms such as laptop, dell, gaming, intel, core, window, pc, ram, ssd, and ryzen - terms typically used in full system listings. In contrast, the "Computer Parts" class featured terms like monitor, cable, printer, router, ink, and cartridge, which are commonly associated with accessories or components.

For each of these selected terms, we engineered a binary feature representing its presence or absence in a given listing's text. This resulted in a set of intuitive and interpretable features that captured the semantic signals most indicative of each class. This keyword-based approach not only simplified the model's input but also improved interpretability without compromising classification performance.

4. Validation

Model Selection

Evaluation Strategy:

To rigorously assess model performance, we applied **Stratified 5-Fold Cross-Validation** using the engineered keyword-based features. This approach maintains the same class distribution across folds, ensuring reliable generalization performance while accounting for the moderate size of our dataset (302 labeled ads). We evaluated four classification models:

- Logistic Regression
- Support Vector Machine (SVM)
- Multinomial Naive Bayes
- Random Forest

Each model was implemented using scikit-learn's standard classification tools and evaluated using four key metrics: **accuracy**, **precision**, **recall**, and **F1-score**.

All models used the **keyword presence matrix** as input features and human_label as the ground truth.

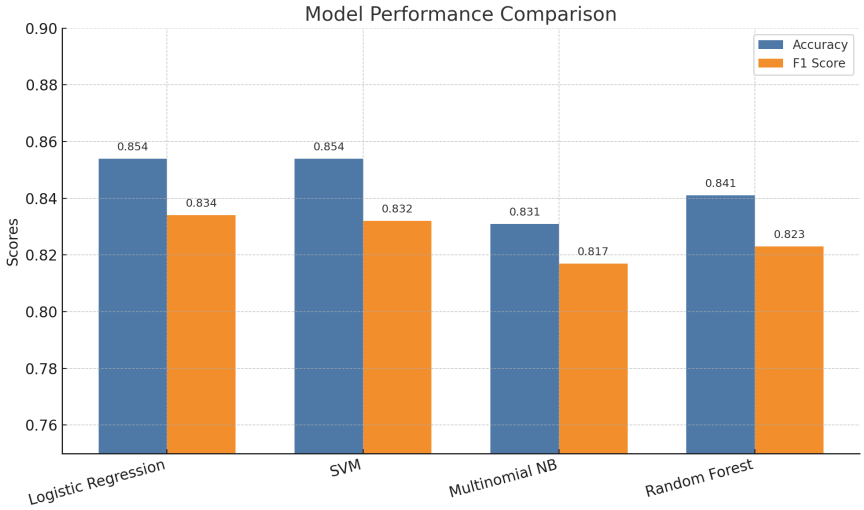
Table 1: Model Performance (5-Fold Cross-Validation)

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.854	0.885	0.793	0.834
SVM	0.854	0.886	0.793	0.832
Random Forest	0.841	0.848	0.800	0.823
Multinomial Naive Bayes	0.831	0.822	0.814	0.817

Model Comparison:

As visualized in the **first bar chart** below, Logistic Regression and SVM both achieved the highest accuracy (85.4%), followed closely by Random Forest (84.1%) and Naive Bayes (83.1%). However, Logistic Regression slightly outperformed all models in **F1-score** (83.4%), indicating a strong balance between precision and recall.

The **second chart** highlights each model's precision and recall trade-offs. Logistic Regression demonstrated the highest **precision (89%)**, suggesting strong confidence in its positive predictions, while its recall (79%) indicates solid retrieval of true positives. Multinomial Naive Bayes and Random Forest offered competitive recall but fell behind in precision.



Model Selection Rationale:

While SVM matched Logistic Regression in accuracy, its slightly lower F1-score (83.2%) and higher computational cost made Logistic Regression the more practical choice. Furthermore, Logistic Regression's **interpretability** - particularly valuable in this context where specific keyword influence is important - offered an added advantage for transparency and explanation.

Henceforth, we chose **Logistic Regression** for its balance of:

- Strong performance (87% in-sample accuracy)
- High interpretability (important for understanding misclassifications)
- Simplicity and speed of training

Final Model Performance:

The finalized Logistic Regression model, trained on the entire cleaned dataset, achieved:

- **Accuracy**: 87%
- **Precision (Computers class)**: 89%
- **Recall (Computer Parts class)**: 91%
- **F1-score (Macro Average)**: 87%

These results demonstrate the model’s strong generalization capability and robust classification performance, making it a reliable candidate for deployment in assisting Craigslist moderation and improving user experience.

5. Conclusion

Summary of Accomplishments

We built and validated an NLP pipeline that can classify Craigslist ads into "Computers" or "Computer Parts" with high accuracy using only the text data. We:

- Conducted ethical data scraping from Craigslist
- Manually verified labels to improve data quality
- Created keyword-based interpretable features
- Tested and evaluated multiple ML models
- Selected a performant and explainable classifier (Logistic Regression)

Visual Insights

- Word clouds showed clear linguistic differences (e.g., "dell", "gaming" vs. "monitor", "router")
- TF-IDF keyword bar plots revealed strong discriminative terms
- Bar plots of accuracy, precision, recall, and F1-score supported our model choice

Real-World Value

- **Moderation Support**: Reduces workload by flagging questionable listings
- **User Experience**: Improves trust and reduces search friction
- **Scalability**: Same pipeline can be trained for Cars vs. Car Parts, Furniture vs. Decor, etc.

Future Directions

- Deploy as an API plugin to auto-suggest categories during post creation
- Train on additional cities and categories
- Extend features to include image metadata or posting behavior
- Explore neural models (e.g., BERT) for deeper semantic understanding