Simon Business School, University of Rochester

Spring B 2025

Social Media & Text Analytics - Project Report

Enhancing Craigslist Gigs with NLP

Date: April 30, 2025

Team 12:

Ashwa Ursani

Dipita Biswas

Krisha Vira

Parul Gupta

Tanjina Moon

Muhammad Zohaib Bin Jamil

Table of Contents

1. Background

1.1 About Craigslist

Craigslist is one of the most widely recognized platforms for peer-to-peer classified advertisements. Since its launch in 1995, it has grown to serve users across more than 70 countries, offering sections for housing, jobs, services, community events, items for sale, and more. Its simple, text-heavy design has remained largely unchanged over the years. While this minimalist approach ensures ease of posting, it often comes at the cost of user experience, especially in categories that rely on quick scanning, clear formatting, or emotion-driven decision-making.

1.2 Focus Area: The Gigs Section

For this project, we focused specifically on the **Gigs** subsection of Craigslist. This section features short-term, freelance, or one-off job opportunities such as event staffing, manual labor, delivery, creative projects, or participating in paid research. These gigs appeal to a wide audience—from students and freelancers to individuals looking for flexible side income.

While the Gigs section is popular and high traffic, it suffers from major usability and structure issues. Every post is written in free-form text, and there are no standardized fields or categories for gig type, duration, or tone. Users often struggle to interpret vague job titles or lengthy descriptions. As a result, it becomes difficult to quickly determine which posts are relevant or trustworthy.

1.3 Identified Problem

After reviewing dozens of listings, it became clear that Craigslist's Gigs section lacks three critical elements:

- **Structure:** There is no categorization system to group similar types of gigs (e.g., delivery vs. research tasks).
- **Tone awareness:** There's no way to detect whether a post sounds friendly, suspicious, urgent, or aggressive.
- **Filtering tools:** Users cannot filter posts by tone, type, or relevance—only by keyword, which often leads to poor search results.

This lack of structure not only frustrates users but also creates challenges for Craigslist's platform moderators. Without any automated system in place, moderators must rely on manual inspection to identify problematic posts, which is time-consuming and impractical at scale.

1.4 Why It Matters

Modern platforms are moving toward AI-enhanced user experiences, offering smart recommendations, tone detection, content filtering, and improved personalization. Craigslist, in contrast, is still operating on a basic free-text system that doesn't interpret or categorize user-generated content.

This is especially problematic in the Gigs section, where posts are time-sensitive and emotion-driven. Users need to be able to quickly find posts that match their interests and comfort levels. For instance, a post asking for "last-minute help at a concert" with an upbeat tone offers a very different experience than a post that is vague and emotionally intense.

By not organizing or labeling these posts in any meaningful way, Craigslist is missing an opportunity to improve trust, safety, and usability for its users.

1.5 Project Scope

Our goal is to apply natural language processing (NLP) techniques to improve the organization and interpretation of gig listings. Specifically, we aim to:

- Group posts into thematic clusters based on text similarity.
- Identify the sentiment (positive, neutral, or negative) of each listing.
- Provide structured metadata that can be used for filtering or flagging.

This solution has the potential to enhance both the user browsing experience and the platform's internal moderation capabilities, without changing the core simplicity that Craigslist is known for.

2. Business Analysis

2.1 Project Objective

The goal of this project is to bring structure, clarity, and intelligence to Craigslist's Gigs section using modern data science tools. Specifically, we aim to enhance the platform's ability to understand two things about every gig listing:

- What kind of gig is this? (i.e., its topic or category)
- How does this gig "feel"? (i.e., the emotional tone or sentiment behind it)

To answer these questions, we applied clustering techniques to group similar posts together, and sentiment analysis to detect whether the tone of a listing is positive, neutral, or negative. By combining both approaches, we can enrich each listing with structured labels that make it easier to search, browse, and moderate. This approach enables smarter gig discovery for users and better content oversight for platform managers, without altering the open-posting format Craigslist is known for.

2.2 Use Case for Users

From a user's perspective, the Gigs section is often overwhelming. Listings have no tags or filters, and many lack meaningful titles. For instance, a post titled "Help Needed Tonight" tells the user nothing about what kind of help is required or where it's located. Another post might be ten lines long, buried in vague language, and still not clarify the type of task.This inconsistency forces users to open multiple listings, skim through unstructured text, and guess at relevance or safety. Our system helps solve that problem by enabling:

- **Smarter browsing**: Users can view gigs grouped by type (e.g., Delivery, Events, Research)
- **Tone-based filtering**: A user who wants friendly and upbeat listings can avoid posts with urgent or aggressive tones.
- **Faster decision-making**: With structured metadata attached, users can spend less time scanning and more time applying.

Imagine a future Craigslist interface where users can click filters like:

- "Positive Tone + Manual Labor"
- "Neutral + Paid Research"
- "All gigs under Events category"

Even if these filters aren't built into the interface right away, our backend labels can serve as a foundation for such features in the future.

2.3 Use Case for Platform Moderators

Moderating Craigslist content is largely a manual task. With no machine-readable tags or emotional context, moderators must either rely on community flagging or scan posts themselves.

Our project introduces a scalable way to automate this review process. By identifying posts with extremely negative sentiment or emotionally charged language, our system could flag potential issues before users even see them. This includes:

- Posts with scam-like urgency ("GET PAID NOW! No questions asked!")
- Emotionally manipulative language
- Posts with repeated or spammy patterns

These flagged posts could be sent to a review queue, helping moderators prioritize and act faster.

2.4 Strategic Business Impact

By enabling text analysis and post categorization, our solution has the potential to unlock new value for Craigslist in several ways:

- **Improved user experience**
  Users find relevant, trustworthy gigs faster, which increases engagement and return visits.
- **Enhanced trust and safety**
  Emotionally extreme or suspicious posts can be proactively monitored, helping to prevent scams or unsafe situations.
- **Operational efficiency**
  Moderators can focus attention where it's most needed, rather than reviewing every post manually.
- **Competitive positioning**
  While Craigslist has maintained its minimalist legacy, competitors like TaskRabbit and Upwork offer smart filters, user ratings, and categorization. Our system helps Craigslist stay relevant without compromising its identity.
- **Future extensibility**
  The approach we used for Gigs can be extended to other sections like Jobs, Housing, or Services. Once the core engine is trained, scaling to other categories is highly feasible.

3. Data Analysis

3.1 Dataset Overview

We worked with a dataset of **939 gig postings** from the Boston Craigslist “Gigs” section, which was obtained from Kaggle. Each entry included fields like:

- post_title_text: the title of the listing
- description: the body text of the listing
- location: if specified by the poster
- pay_rate: optional field, often inconsistently filled
- post_datetime: timestamp of the posting

This dataset provided us with a rich sample of user-generated text in the wild—informal, inconsistent, and often emotionally expressive. It was the ideal candidate for applying natural language processing (NLP) techniques to extract insights.

3.2 Text Preprocessing

To prepare the data for modeling, we applied a series of standard text-cleaning steps:

- **Lowercasing:** To remove case sensitivity
- **Stopword Removal:** Removed common but non-informative words like "and", "the", etc.
- **Lemmatization:** Reduced words to their base form (e.g., “running” → “run”)

These steps helped standardize the language used across listings and made the posts machine-readable for downstream analysis.

3.3. Feature Engineering with TF-IDF

We used TF-IDF (Term Frequency – Inverse Document Frequency) to convert text into numerical feature vectors. This helped identify which terms were important within a post relative to the entire dataset.

TF-IDF Settings:

- **n-grams:** 1 to 2 (unigrams and bigrams)
- **min_df:** 3 (terms must appear in at least 3 documents to be included)

TF-IDF created a sparse matrix where each post was represented by thousands of weighted word features. This matrix became the input for both clustering and classification models.

3.4 Descriptive Model: Clustering

To identify hidden themes within the listings, we applied **KMeans clustering**, an unsupervised learning method. After testing different values of k, we selected **k = 4**, which produced coherent and meaningful clusters.

Resulting Gig Clusters:

1. Delivery & Logistics (e.g., food delivery, rideshare)
2. Events & Promo Work (e.g., event staff, brand ambassadors)
3. Research & Testing (e.g., paid surveys, clinical studies)
4. Manual Labor (e.g., yard work, moving help)

These clusters were validated through:

- Top keyword analysis from each cluster
- Manual inspection of representative samples

This thematic breakdown adds structure to the Gigs section, allowing for future development of category filters or smart search features.

3. 5. Sentiment Labeling with VADER

To analyze the emotional tone of each post, we applied **VADER (Valence Aware Dictionary and Sentiment Reasoner)**, a lexicon-based sentiment analysis tool well-suited for short, social-style text.

Each post received a compound score (ranging from -1 to 1) which we converted into discrete sentiment classes:

- Positive: Compound score > 0.05
- Neutral: Score between -0.05 and 0.05
- Negative: Score < -0.05

This approach gave us a labeled sentiment_label column to use in our predictive modeling.

3.6 Predictive Models

With TF-IDF as input features and sentiment labels as the target, we trained five different classification models:

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

We split the data into **70% training** and **30% testing**, with **stratification** to maintain the class distribution across splits.

3.7 Model Input Summary

- X (Input): TF-IDF matrix from gig descriptions
- y (Target): Sentiment label (Positive, Neutral, Negative) from VADER
- Train/Test Split: 70/30
- Cross-Validation: Manual tuning + stratified holdout validation

Each model was evaluated using standard classification metrics: accuracy, precision, recall, and F1 score. Full validation results are detailed in the next section.

4. Validation

4.1 Evaluation Strategy

We used a **70/30 stratified train-test split** to maintain balanced sentiment classes. Models were evaluated using standard metrics:

- Accuracy: Overall correctness
- Precision: How many predicted positives were actually correct
- Recall: How many actual positives were correctly predicted
- F1 Score: Harmonic mean of precision and recall

4.2 Model Performance Summary

Naïve Bayes: The Naive Bayes model achieved an accuracy of 0.71, with a precision of 0.62, recall of 0.71, and an F1 score of 0.66. Its performance was consistent across metrics, indicating a balanced ability to classify different sentiment classes. While it did not outperform other models in any specific area, its simplicity and efficiency make it a strong baseline classifier for text classification tasks.

Logistic Regression: Logistic Regression produced an accuracy of 0.70, precision of 0.62, recall of 0.70, and F1 score of 0.66. The model demonstrated solid overall performance, particularly in handling neutral and positive sentiments. However, it did not offer improvements over Naive Bayes and struggled to distinguish underrepresented classes, which limits its effectiveness in this application.

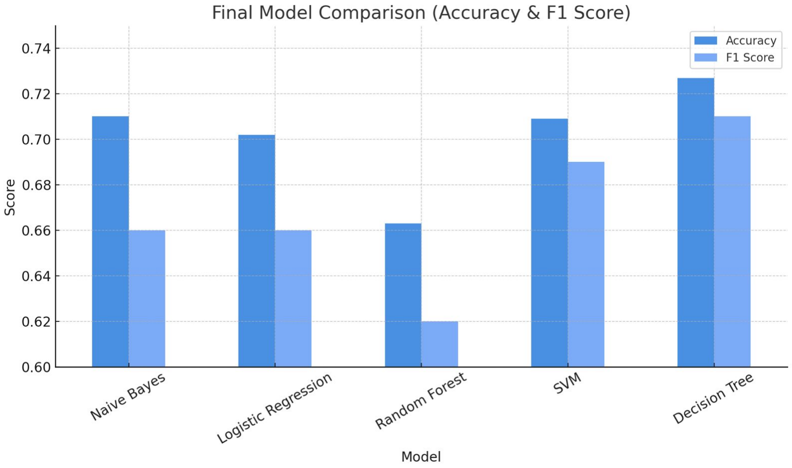
Random Forest: The Random Forest model obtained an accuracy of 0.66, with a precision of 0.58, recall of 0.66, and an F1 score of 0.62—the lowest overall among the models tested. Despite its strength as an ensemble method, its relatively low precision and F1 score suggest that it may have overfitted to the training data or was negatively affected by class imbalance.

Support Vector Machine (SVM): SVM demonstrated strong and consistent performance, with an accuracy of 0.71, precision of 0.69, recall of 0.71, and an F1 score of 0.69. These results indicate that the model was well-suited for the sentiment classification task, offering a good balance between precision and recall across sentiment classes. It was among the most effective models evaluated.

Decision Tree: The Decision Tree model achieved the highest performance overall, with an accuracy of 0.73, precision of 0.71, recall of 0.73, and an F1 score of 0.71. It demonstrated superior ability to classify sentiment accurately while maintaining model interpretability.

Based on its leading performance across all evaluation metrics, the Decision Tree was selected as the final model for deployment.

Model	Accuracy	Precision	Recall	F1 Score
Naive Bayes	0.71	0.62	0.71	0.66
Logistic Regression	0.70	0.62	0.70	0.66
Random Forest	0.66	0.58	0.66	0.62
SVM	0.71	0.69	0.71	0.69
Decision Tree	**0.73**	**0.71**	**0.73**	**0.71**



Best Model: Decision Tree**:** it achieved the highest scores across all four metrics, while also being transparent and easy to interpret.

4.3 Sentiment Label Distribution

Out of 939 posts: **62% Positive**, **28% Neutral**, **10% Negative**. This distribution influenced model behavior, and the stratified split helped mitigate any class imbalance.

5. Conclusion

5.1 Project Summary

The goal of this project was to improve the structure and usability of Craigslist's Gigs section using natural language processing and machine learning. We addressed the core issue of unstructured, inconsistent postings by developing a system that analyzes:

- **What the post is about** (via clustering)
- **How it feels** (via sentiment analysis)

We applied KMeans clustering to group posts into themes like Delivery, Events, Research, and Manual Labor, and used VADER sentiment analysis to label each post as Positive, Neutral, or Negative. These sentiment labels were then used to train five classification models. Among them, the Decision Tree model performed best, achieving 73% accuracy with balanced precision and recall.

5.2 Value to Craigslist

Our solution brings structure and intelligence to Craigslist without changing its core interface.

For Users:

- Smarter browsing with category and tone filters
- Faster decision-making with cleaner post summaries
- Improved trust through clearer, more relevant listings

For Moderators:

- Flagging negative/emotional posts for review
- Time-saving moderation powered by classification
- Better content quality without manual scanning

5.3 Impact & Scalability

This solution adds intelligence without changing Craigslist's simple interface. It can scale to other sections like Jobs or Services, improving usability, safety, and engagement across the platform.