NLP: Fake Review Detection

Name: Nischay Patel

Roll Number: 220721

Problem Statement

The dataset contains the reviews posted by several amazon users along with its metadata and the item metadata.
Identify fake reviews or anomalies in reviews by looking at the ratings and the review text: For example, if the review text has positive words/sentiment, it can be assumed to receive high rating, take it as an anomaly incase of the contrary. Similar logic applies for negative words/sentiments.

- You can use any text based analysis (ex. Negative Word frequency, LDA) we discussed in the class for this purpose.

- Use a regression model to evaluate whether the presence of fake reviews relates to the price of the item

Methodology

My approach follows a structured five-step analysis:

1. **Dataset Selection & Preprocessing**

   Selecting an appropriate subset of the dataset provided.

   Cleaning and standardizing text data.

2. **Preliminary Data Analysis**

   Exploratory Data Analysis (EDA) to understand patterns in review distribution, sentiment, and metadata.

   Identifying statistical anomalies in ratings.

3. **Duplicate & Pattern-Based Detection**

   Identifying duplicate or near-duplicate reviews.

   Detecting unnatural patterns in repeated phrases.

4. **Fake Review Detection**

   LDA Topic Modeling and **POS Tagging** to analyse syntactic structures to detect unnatural or overly generic phrasing.

5. **Regression Analysis**

   Use a regression model to evaluate whether the presence of fake reviews relates to the price of the item.

1. Dataset Selection & Preprocessing

We selected the Appliances category from the Amazon Reviews ‘23 dataset, which contains 1.8 million user reviews.
To enhance our analysis, we merged the reviews dataset with its metadata, resulting in a comprehensive dataset with 25 columns and 1.8 million rows.
For duplicate detection, we identified and removed rows where the following attributes were identical:

Timestamp
Review Title (title_x)
Review Text (text)
Title (title)
User ID (user_id)
ASIN (Amazon Standard Identification Number)
Parent ASIN

Only the first occurrence of each duplicate entry was retained to preserve data integrity. After deduplication, further preprocessing was conducted as and when required.

2. Preliminary Data Analysis

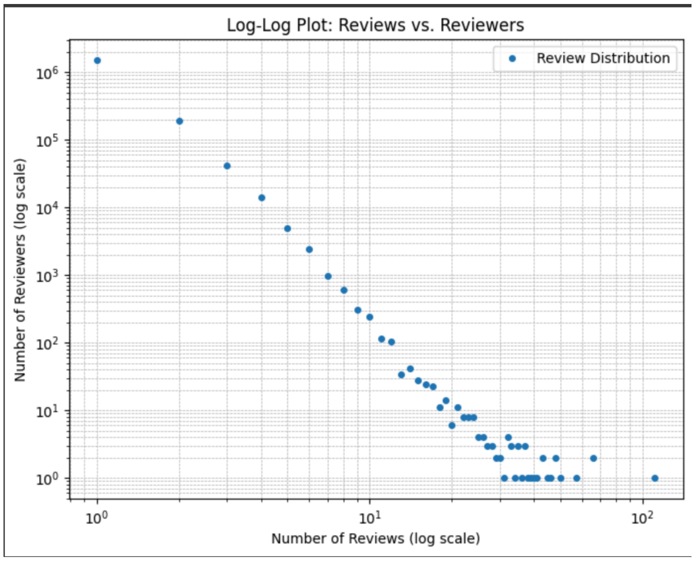
A comprehensive analysis of the reviews, reviewers and products was done to understand the nature of the data and any consistent patterns.
Specifically, we show the following plots:

1. Number of reviews vs. number of reviewers
2. Number of reviews vs. number of products

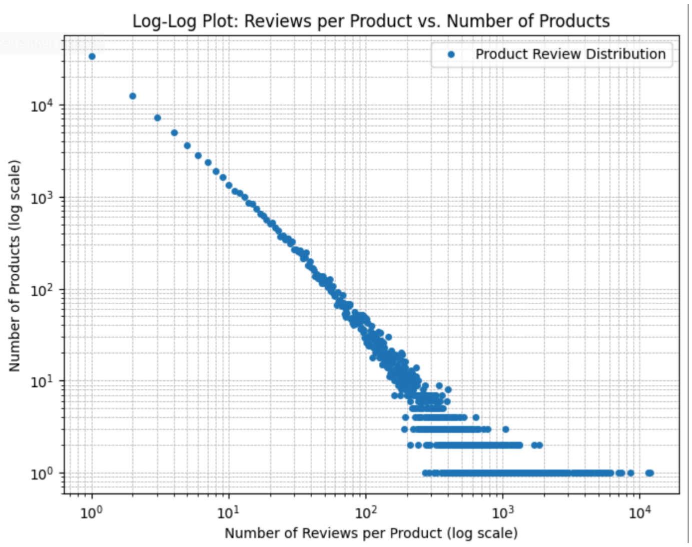
These relationships all follow the power law distribution. A power law relationship between two quantities x and y can be written as

$$y = ax^k$$

where a and k are constants. If we take the log on both sides, we obtain a straight line on a log-log plot.



The above figure shows the **log-log plot of “number of reviews vs. number of reviewers”**. We can see that a large number of reviewers write only a few reviews, and a few reviewers write a large number of reviews. There are 2 reviewers with more than 100,000 reviews, and 68% of reviewers wrote only 1 review. Only 8% of reviewers wrote at least 5 reviews.



The above figure shows the **log-log plot of “number of reviews vs. number of products”**. Again, we can see that a large number of products get very few reviews and a small number of products get a large number of reviews. For example, 50% of products have only 1 review. Only 19% of the products have at least 5 reviews.

3. Duplicate & Pattern Based Detection

Interestingly, in my analysis, I found a large number of duplicate and near-duplicate reviews. Our manual inspection of such reviews shows that they definitely contain some spam reviews. We are also sure that they contain untruthful reviews because of the following types of duplicates (the duplicates include near-duplicates):

1. Duplicates from different userids on the same product.
2. Duplicates from the same userid on different products.
3. Duplicates from different userids on different products.

To ensure meaningful duplicate detection, we focused on a niche subset of similar products, as analyzing duplicate reviews across a wide range of items would be ineffective.
We initially selected the "Industry and Scientific" category under Appliances, but it still encompassed a diverse set of products. To refine our analysis further, we narrowed it down to "Refrigerators," which contained 23,781 reviews.
This targeted approach allowed for more accurate identification of duplicate reviews while maintaining relevance within a specific product category.

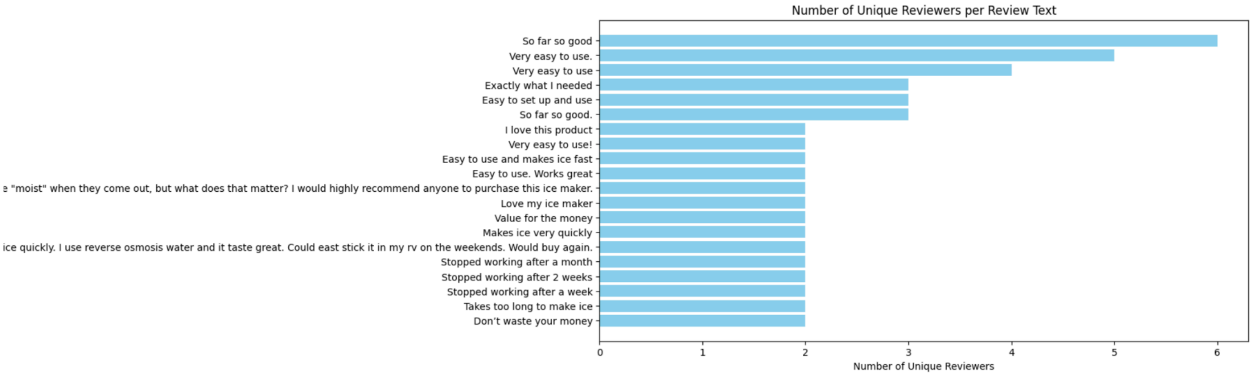
To detect duplicate reviews, we employed the MinHash LSH(Locality-Sensitive Hashing) method, which efficiently identifies duplicate and near-duplicate text entries.

The MinHash algorithm is a probabilistic technique used to estimate the Jaccard similarity between sets without explicitly comparing all elements. It works by generating multiple hash functions to create "signatures" for each text, allowing for quick similarity estimation.

Locality-Sensitive Hashing (LSH) extends MinHash by grouping similar items into "buckets" using a hashing mechanism. This reduces the number of pairwise comparisons needed, making it highly efficient for detecting near-duplicate text.

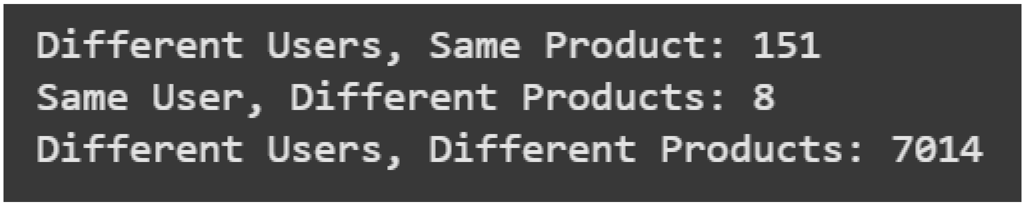
By computing similarity scores between reviews, we classified reviews as duplicates if their Jaccard similarity score exceeded 0.9. This approach enabled us to detect highly similar reviews while significantly reducing computational overhead compared to pairwise comparisons.

To visualize the extent of duplicate reviews, we created a bar plot showcasing the top 40 most frequently occurring reviews that contained at least 4 words. This filtering step ensured that we only considered meaningful text and avoided coincidental duplicates arising from short, common phrases.



The bar plot effectively highlighted the presence of multiple long duplicate reviews, suggesting a pattern of repetition that is unlikely to be organic. The high frequency of these reviews strongly indicates that they may be spam, artificially generated content, or untruthful reviews intended to manipulate product ratings.

To further confirm the hypothesis, the duplicate reviews were sorted into the three above mentioned categories:
1. Duplicates from different userids on the same product.
2. Duplicates from the same userid on different products.
3. Duplicates from different userids on different products.



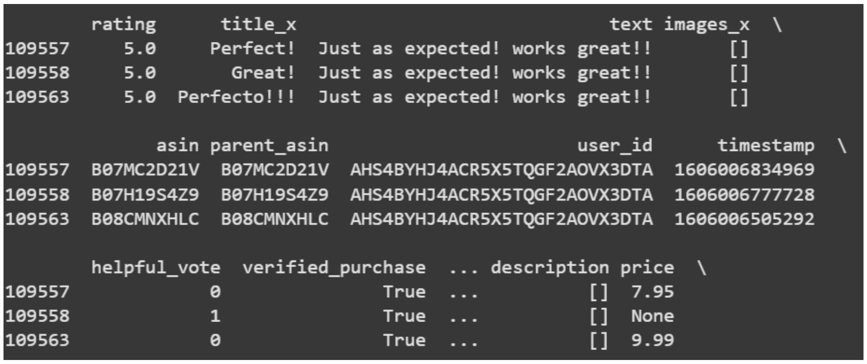
```
Different Users, Same Product: 151
Same User, Different Products: 8
Different Users, Different Products: 7014
```

The figure above presents the results after sorting the detected duplicates. Two key insights stand out—151 duplicate reviews were posted by different users for the same product and 8 duplicate reviews were posted by the same user for different products , which strongly suggests potential bot activity. This pattern indicates that automated accounts may have been used to artificially inflate or manipulate product ratings, reinforcing the need for rigorous fake review detection.

Duplicate Reviews CSV: File

Given below are some examples of duplicate reviews, which are highly suspicious and are most definitely bots.

Same User, Different Products



```
        rating      title_x                             text images_x  \
109557     5.0     Perfect!  Just as expected! works great!!       []
109558     5.0       Great!  Just as expected! works great!!       []
109563     5.0  Perfecto!!!  Just as expected! works great!!       []

              asin parent_asin                       user_id      timestamp  \
109557  B07MC2D21V  B07MC2D21V  AHS4BYHJ4ACR5X5TQGF2AOVX3DTA  1606006834969
109558  B07H19S4Z9  B07H19S4Z9  AHS4BYHJ4ACR5X5TQGF2AOVX3DTA  1606006777728
109563  B08CMNXHLC  B08CMNXHLC  AHS4BYHJ4ACR5X5TQGF2AOVX3DTA  1606006505292

        helpful_vote  verified_purchase  ... description price  \
109557             0               True  ...          []  7.95
109558             1               True  ...          []  None
109563             0               True  ...          []  9.99
```

The figure above highlights a few key observations:

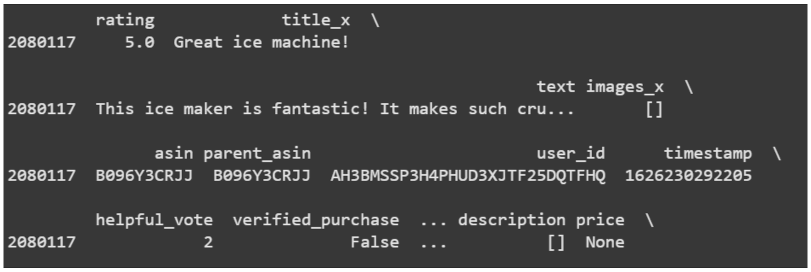
Different product titles, same user ID, and identical review text – This suggests the user has posted the same review across multiple products.

All reviews are verified purchases – Adding credibility despite their duplication.

One review even received a helpful vote – Indicating that duplicate reviews can influence customer perception.

These patterns reinforce the idea that fake or bot-generated reviews may significantly impact buyers' trust and purchasing decisions.

Different users, Different Products

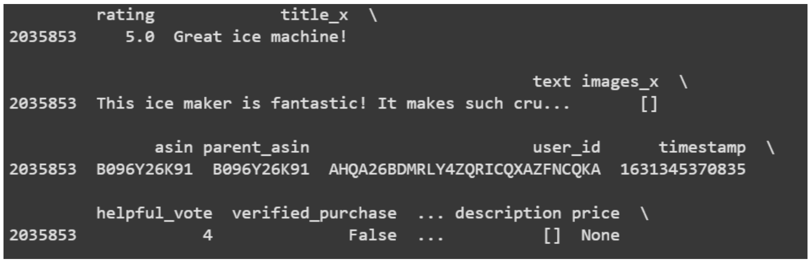


```
         rating             title_x  \
2080117     5.0  Great ice machine!   

                                                      text images_x  \
2080117  This ice maker is fantastic! It makes such cru...       []   

               asin parent_asin                       user_id      timestamp  \
2080117  B096Y3CRJJ  B096Y3CRJJ  AH3BMSSP3H4PHUD3XJTF25DQTFHQ  1626230292205   

         helpful_vote  verified_purchase  ... description price  \
2080117             2              False  ...          []  None   
```



```
         rating             title_x  \
2035853     5.0  Great ice machine!   

                                                      text images_x  \
2035853  This ice maker is fantastic! It makes such cru...       []   

               asin parent_asin                       user_id      timestamp  \
2035853  B096Y26K91  B096Y26K91  AHQA26BDMRLY4ZQRICQXAZFNCQKA  1631345370835   

         helpful_vote  verified_purchase  ... description price  \
2035853             4              False  ...          []  None   
```

The figure highlights a few key observations:

Same product title, Identical review text but Different user ids and Different products - This suggests that the same user has posted the exact same review under different accounts.

Not a verified purchase but 4 helpful votes - Even though it is not a verified purchase, customers found this review helpful possibly due to the length of the review.

4. Fake Review Detection And Labelling

To refine our dataset for fake review detection, we applied the following filters:

Excluded reviews that did not have a 5-star rating – Fake reviews are often overly positive, and it is rare for someone to fabricate a positive review while assigning a low rating.

Removed reviews with fewer than 150 characters – Short reviews typically lack substance, making them less useful for detecting patterns of deception.

After applying these filters, we were left with 3,624 reviews, which formed the basis for our subsequent analysis.

Since Support Vector Machines (SVM) require a labeled dataset, we created a training subset of 100 randomly selected reviews. However, to label these reviews accurately, we first needed to analyze the linguistic traits of fake vs. truthful reviews.

To distinguish between truthful and fake reviews, we categorized extracted topics into five distinct linguistic patterns:

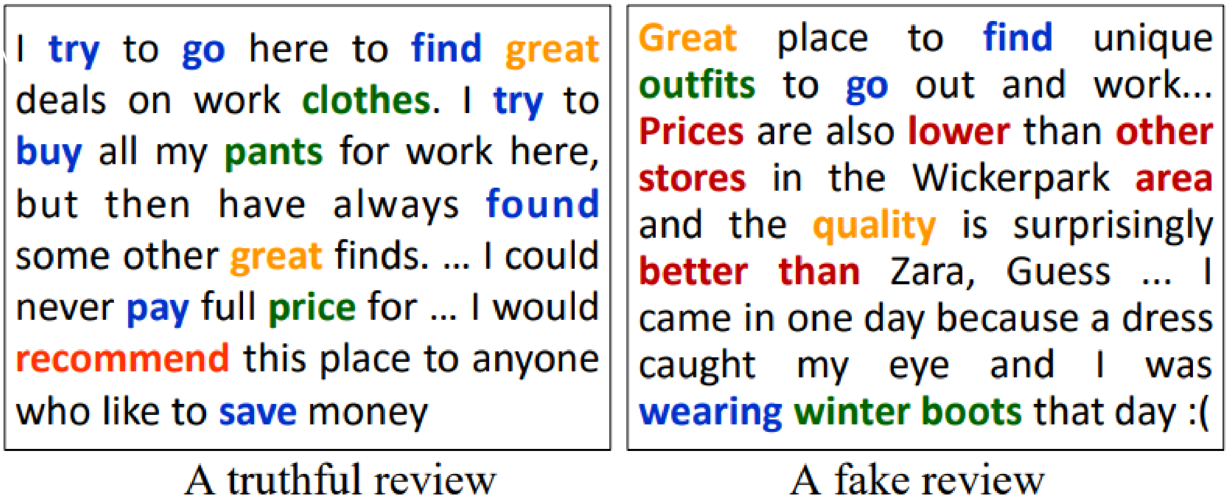
Concrete Experience (CE): Topics dominated by verbs, indicating firsthand experience.

Detailed Information (DI): Topics rich in specific nouns and adjectives, demonstrating factual descriptions.

General Comments (GC): Topics that contain abstract evaluations, often vague or overly generic.

Comparative Assessment (CA): Topics including comparative or superlative words, such as "better," "best," "worse."

Recommendation and Reference (RR): Topics containing words like “recommend” or “refer”, commonly seen in marketing-driven reviews.



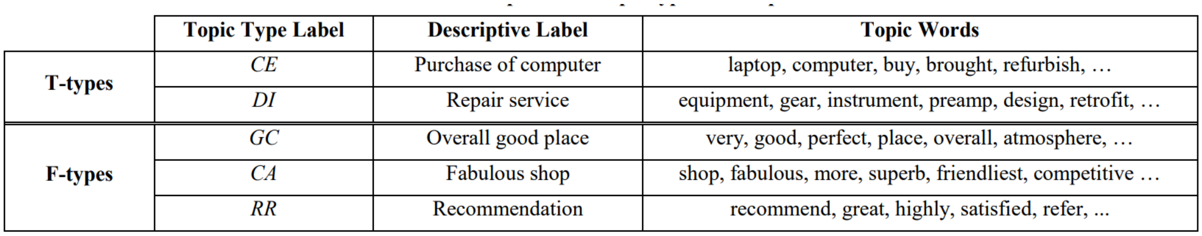
I try to go here to find great deals on work clothes. I try to buy all my pants for work here, but then have always found some other great finds. ... I could never pay full price for ... I would recommend this place to anyone who like to save money

A truthful review

Great place to find unique outfits to go out and work... Prices are also lower than other stores in the Wickerpark area and the quality is surprisingly better than Zara, Guess ... I came in one day because a dress caught my eye and I was wearing winter boots that day :(

A fake review

Figure 2: Examples of a truthful and a fake review. The Color of words denotes type of topic which the word is included in: CE (blue), DI (green), GC (yellow), CA (red) and RR (orange).



	Topic Type Label	Descriptive Label	Topic Words
T-types	*CE*	Purchase of computer	laptop, computer, buy, brought, refurbish, …
	DI	Repair service	equipment, gear, instrument, preamp, design, retrofit, …
F-types	*GC*	Overall good place	very, good, perfect, place, overall, atmosphere, …
	CA	Fabulous shop	shop, fabulous, more, superb, friendliest, competitive …
	RR	Recommendation	recommend, great, highly, satisfied, refer, ...

To label the training dataset systematically, we followed a structured approach:

1. Extracted 100 topics using **Latent Dirichlet Allocation (LDA) modeling.**
2. Applied **POS (Part-of-Speech) tagging** to classify each word (verb, noun, adjective, etc.).
3. Used a predefined abstract nouns set to identify general vs. specific nouns.
4. Categorized each word into one of the five linguistic types:

   CE +1 → If the word is a verb.

   DI +1 → If the word is a noun (excluding abstract nouns).

   GC +1 → If the word is an adjective or found in the abstract nouns set.

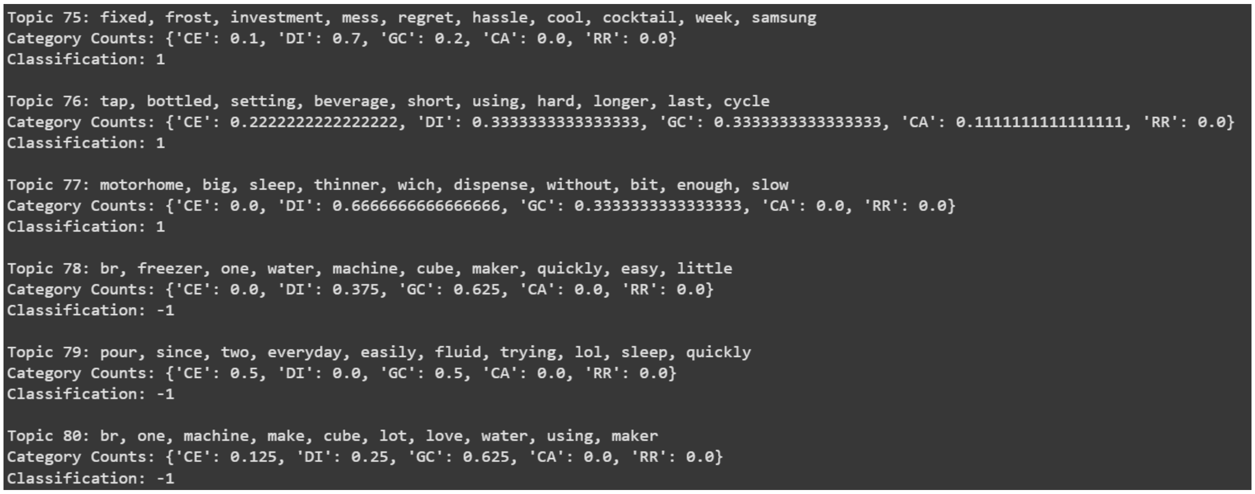
   CA +1 → If the word is a comparative or superlative.

   RR +1 → If the word belongs to a custom set of recommendation-related words.

5. Computed proportions for each category by normalizing the values.
6. Labelled a topic as ‘truthful’ if **CE + DI words > GC + CA + RR words**, indicating more descriptive and experience-based content.

Once the training dataset was labeled, we used it to train an SVM classifier.

The SVM classifier trained with an accuracy of 95%. (Topic_Classifications.csv)



```
Topic 75: fixed, frost, investment, mess, regret, hassle, cool, cocktail, week, samsung
Category Counts: {'CE': 0.1, 'DI': 0.7, 'GC': 0.2, 'CA': 0.0, 'RR': 0.0}
Classification: 1

Topic 76: tap, bottled, setting, beverage, short, using, hard, longer, last, cycle
Category Counts: {'CE': 0.2222222222222222, 'DI': 0.3333333333333333, 'GC': 0.3333333333333333, 'CA': 0.1111111111111111, 'RR': 0.0}
Classification: 1

Topic 77: motorhome, big, sleep, thinner, wich, dispense, without, bit, enough, slow
Category Counts: {'CE': 0.0, 'DI': 0.6666666666666666, 'GC': 0.3333333333333333, 'CA': 0.0, 'RR': 0.0}
Classification: 1

Topic 78: br, freezer, one, water, machine, cube, maker, quickly, easy, little
Category Counts: {'CE': 0.0, 'DI': 0.375, 'GC': 0.625, 'CA': 0.0, 'RR': 0.0}
Classification: -1

Topic 79: pour, since, two, everyday, easily, fluid, trying, lol, sleep, quickly
Category Counts: {'CE': 0.5, 'DI': 0.0, 'GC': 0.5, 'CA': 0.0, 'RR': 0.0}
Classification: -1

Topic 80: br, one, machine, make, cube, lot, love, water, using, maker
Category Counts: {'CE': 0.125, 'DI': 0.25, 'GC': 0.625, 'CA': 0.0, 'RR': 0.0}
Classification: -1
```

The figure above is a snippet of the labelling the training dataset.

For labeling the remaining reviews:

Each review was converted into a feature vector based on POS-tagging-derived proportions of the five linguistic categories.

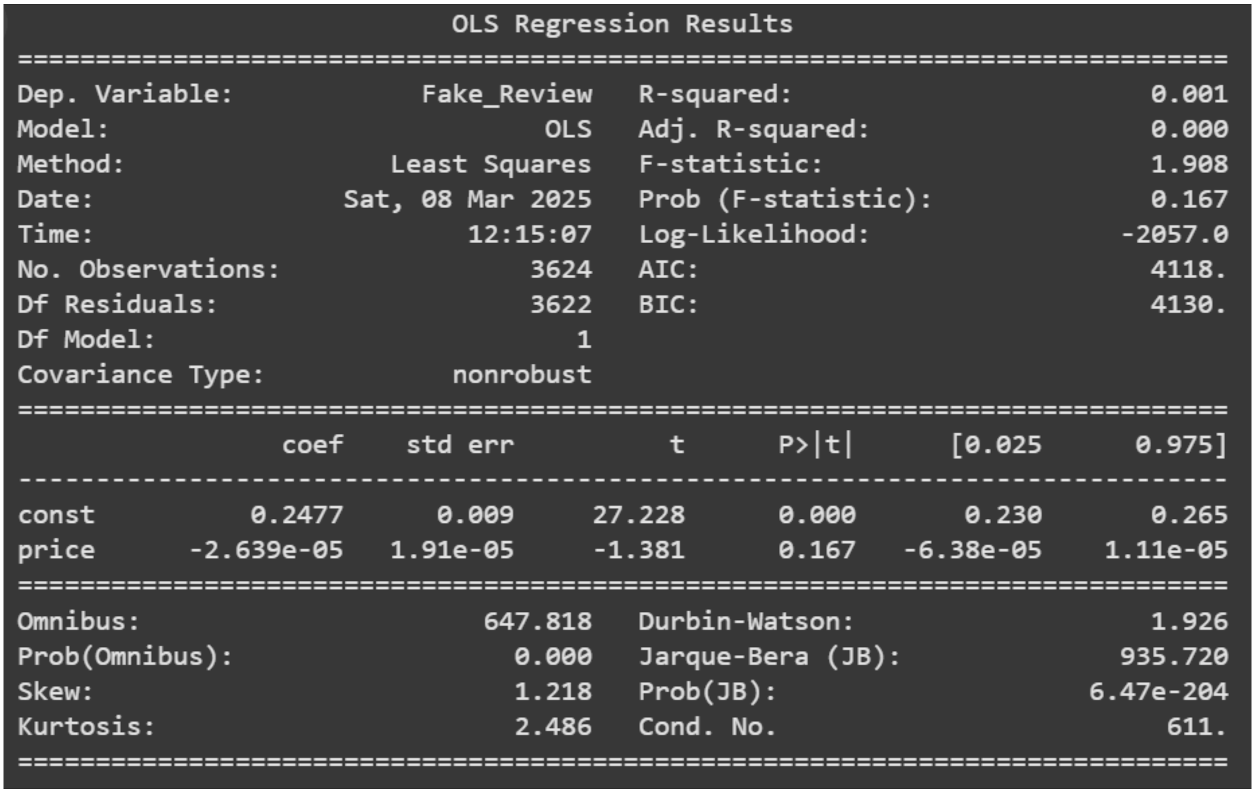
The trained SVM model was used to classify each review as either fake or truthful. (Final_Predictions)

This methodology ensured a data-driven and linguistically informed approach to identifying fake reviews while reducing bias in manual labeling.

5. Regression Analysis

Now using the previously labelled dataset, a regression analysis was performed to evaluate if the presence of fake reviews relates to the price of the product.

The following results were obtained:



```
                            OLS Regression Results
==============================================================================
Dep. Variable:            Fake_Review   R-squared:                       0.001
Model:                            OLS   Adj. R-squared:                  0.000
Method:                 Least Squares   F-statistic:                     1.908
Date:                Sat, 08 Mar 2025   Prob (F-statistic):              0.167
Time:                        12:15:07   Log-Likelihood:                -2057.0
No. Observations:                3624   AIC:                             4118.
Df Residuals:                    3622   BIC:                             4130.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.2477      0.009     27.228      0.000       0.230       0.265
price      -2.639e-05   1.91e-05     -1.381      0.167   -6.38e-05    1.11e-05
==============================================================================
Omnibus:                      647.818   Durbin-Watson:                   1.926
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              935.720
Skew:                           1.218   Prob(JB):                    6.47e-204
Kurtosis:                       2.486   Cond. No.                         611.
==============================================================================
```

R-squared: 0.001 → This means that only 0.1% of the variation in Fake_Review is explained by price, indicating an extremely weak relationship.

F-statistic: 1.908, with a p-value of 0.167, suggesting that price does not significantly predict Fake_Review.

Intercept (const) = 0.2477 → When price = 0, the expected probability of a fake review is 24.77%.

Price coefficient = -2.639e-05 → A 1-unit increase in price decreases the probability of a review being fake by 0.00002639 (which is negligible).

P-value for price = 0.167 → Since this value is greater than 0.05, the relationship is not statistically significant, meaning price has no strong impact on the likelihood of a review being fake.

Conclusions:

- The price of a product does not significantly impact the probability of a review being fake.
- The model is weak ($R^2 = 0.001$), meaning price explains almost none of the variation in fake reviews.
- Other factors (e.g., review text, user behavior, timestamps) are likely better predictors of fake reviews.

Tech Stack

Programming Language:

- **Python** – for data processing, analysis, and model building.

Libraries & Frameworks:

- **Pandas** – for data manipulation and preprocessing.
- **NumPy** – for numerical computations.
- **NLTK (Natural Language Toolkit)** – for POS tagging and text analysis.
- **Scikit-learn** – for training the SVM and Logistic Regression models.
- **Gensim** – for LDA (Latent Dirichlet Allocation) topic modeling.
- **Matplotlib/Seaborn** – for data visualization (bar plots, duplicate review analysis).
- **Statsmodels** – for performing OLS (Ordinary Least Squares) regression analysis.
- **MinHash LSH (from Datasketch)** – for detecting duplicate reviews based on similarity.

Machine Learning Models:

- **Support Vector Machine (SVM)** – for classifying reviews as fake or genuine.
- **LDA Topic Modeling** – for categorizing reviews into different topic types.

Data Processing Techniques:

- **POS Tagging (Part-of-Speech Tagging)** – for identifying key linguistic patterns in reviews.
- **TF-IDF (Term Frequency-Inverse Document Frequency)** – for feature extraction from text.
- **MinHash + Locality-Sensitive Hashing (LSH)** – for duplicate review detection.

Problems Encountered

1. Handling some subsets of the dataset on a laptop was impractical due to its massive size, so only specific subsets were selected for analysis.
2. Additionally, there was no definitive way to verify whether the labeled reviews were entirely accurate. The labeling process relied on automated methods, which inherently introduce some degree of uncertainty.
3. Furthermore, the NLTK library used for POS-tagging occasionally misclassified certain words, potentially impacting the accuracy of the labelling process. These limitations highlight the challenges in ensuring precise identification of fake reviews.

Future Scope

1. The SVM model used for fake review detection can be further enhanced by incorporating additional features such as helpful votes, verified purchase status, and images attached to the review. These features provide valuable context, helping to improve the model's accuracy by distinguishing between genuine and suspicious reviews more effectively.

2. Furthermore, the duplicate reviews identified through MinHash LSH can be leveraged to train a logistic regression model aimed at detecting non-identical but still fraudulent reviews. By introducing a combination of review-centric, product-centric, and reviewer-centric features, the model can capture patterns beyond simple text similarity.

Code File: CGS616_2.ipynb

References

[1] Vaidya, R. *Fake Review Detection* [Bachelor's thesis, Jaypee University of Information Technology]. Department of Computer Science & Engineering and Information Technology.

[2] K. D. Lee, K. Han, and S.-H. Myaeng, "Capturing word choice patterns with LDA for fake review detection in sentiment analysis," *School of Computing, Korea Advanced Institute of Science and Technology*, Daejeon, South Korea.

[3] N. Jindal and B. Liu, "Opinion spam and analysis," *Department of Computer Science, University of Illinois at Chicago*, Chicago, IL, USA.