



Fake Google Reviews Detection

Team Members: Nick Sietsema, Rebecca Toland, and Amber Tin



Project Description

- 95% of customers read online reviews before buying a product
- More than 30% of online reviews **deemed fake**
- 2021 Google local reviews dataset for Alaska used to detect and flag fake reviews
- Dataset URL:
 - https://datarepo.eng.ucsd.edu/mcauley_group/gdrive/googlelocal/



Question(s) sought to answer

- Is the customer's username significant?
- How many reviews per day would we expect the normal customer to post?
- Are there sudden spikes in review activity from multiple accounts within a short period of time?

Ultimately...

Can we uncover patterns and use data mining to detect fake reviews?



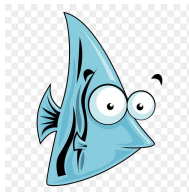


Data Preparation Work

- Reducing the size of our dataset from 89.9 MB to 68.5 MB
- Chunking the data allowing it to run on specific server
- Drop duplicates and null values
- Dropped irrelevant columns such as the pictures and business responses
- Changed the unix date to a more readable time -> Year-Month-Day
- Creating new columns with our different classifications
- Feature Engineering:
 - Review frequency by user
 - Mean word count in text reviews

Dataset Distribution

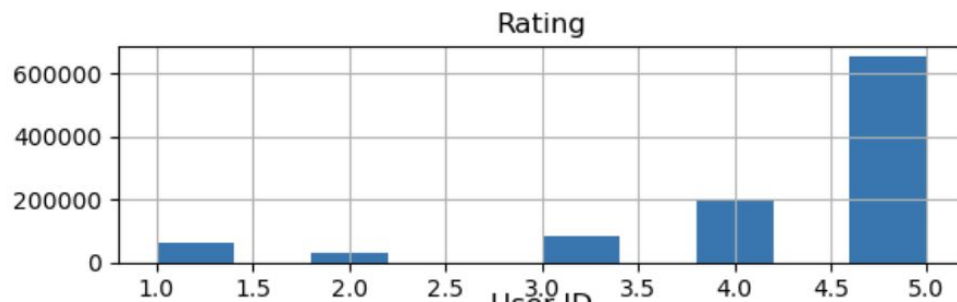
- 1.03 million reviews within dataset
 - 621,209 reviews after dropping NA
 - 411,557 reviews with no content
 - 571,540 unique reviews
 - Means there are duplicate entries for reviews
- 278,478 Unique User IDs
- 1,032,766 Timestamps
 - 1,032,412 unique timestamps
 - Duplicate timestamps - very fishy!



```
df.count()
```

Business ID	1032766
Name	1032766
Rating	1032766
Review	621209
Time	1032766
User ID	1032766
dtype:	int64

Dataset Distribution Cont.



5 star reviews: 651555

4 star reviews: 201298

3 star reviews: 85478

1 star reviews: 62473

2 star reviews: 31962



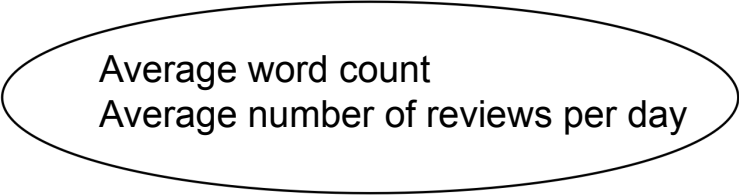
Tools Used

- Python 3
- Pandas for data cleaning and preprocessing
 - Allowed us to use a variety of mathematical operations on arrays
 - Open up compressed files
 - Chunking our data to readable sets and concatenating them back into one array
- NumPy
 - Used `np.where()` to create a new conditional array for features like fake names, mean word count, flag users with more than 1 name and user IDs with a mean rating >3.
- Matplotlib used for data visualization
 - Visualize our K means visualization
- Datetime
 - Changing unix time to ms
- Sklearn
 - Cluster package for K means Data Mining Technique
 - Preprocessing for Standard Scaler
- Github
 - Committing all changes made by each individual
- Slack
 - Main communication platform



Unsupervised Learning (K-means)

New Features - User Level:



Average word count
Average number of reviews per day

Fake name flag
More than one user name per ID



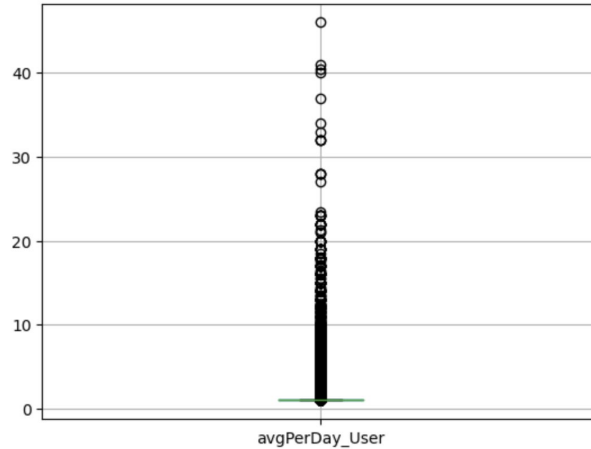
K-means Clustering



Classify: Fake or Real

Feature Engineering

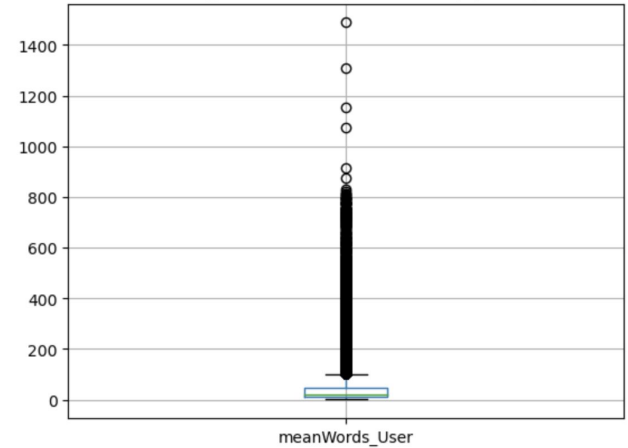
Average # of Reviews per Day by User



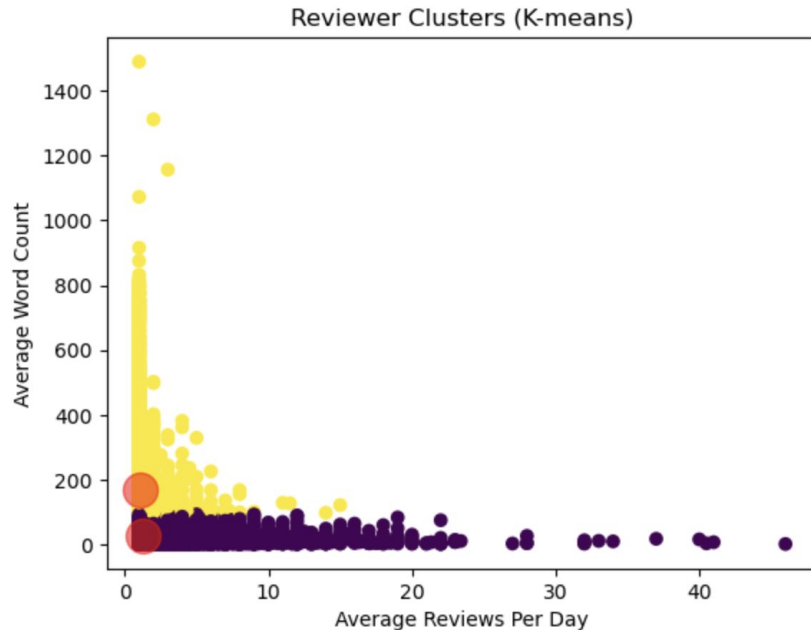
Averages by User

	Reviews per Day	Words per Review
Mean	1	38
50%	1	22
Maximum	46	1,489

Average # of Words per Review (by User)

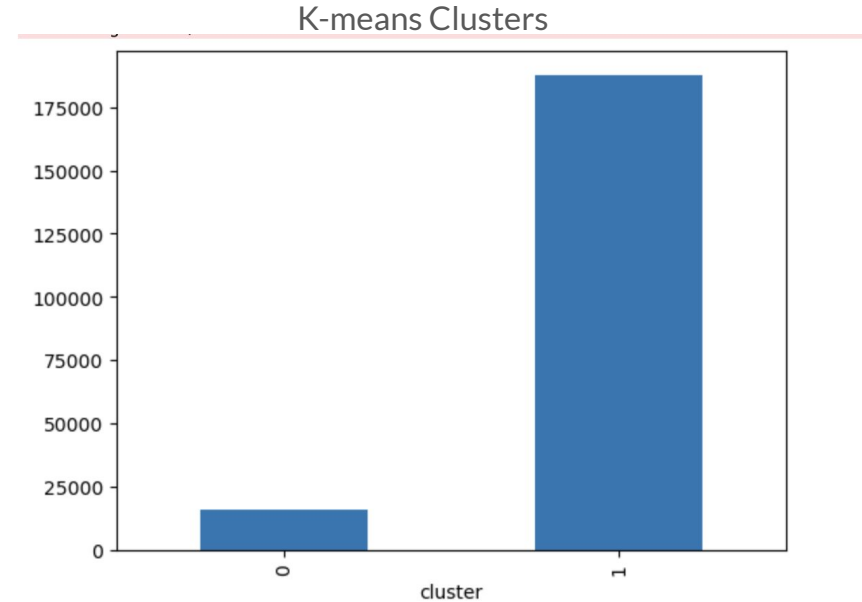


Knowledge gained



Purple: Fake Reviewers
Yellow: Legitimate Reviewers

Estimated Fake Google Reviewers: 10.7% (CX solutions report)
K-means Clustering Result: 8% Fake Reviewers



0 = Fake Reviewers (8%)
1 = Legitimate Reviewers (92%)



How that knowledge can be applied

Protect Consumers and Honest Businesses:

- Prevent wasteful spending
- Consumers need access to honest review platforms -> higher quality products & services
- Support honest businesses and prevent competitors from tarnishing honest ratings

Apply Knowledge Towards Further Research:

- Machine learning for analyzing text / user profiles
- Exploring other unsupervised learning models