Detecting Fake Amazon Reviews With Machine Learning

Springboard Data Science Career Track
Capstone 2 - Final Report
by Chantel Clark

Suspect reviews

The Problem

A firms' online reputation is paramount to gaining the trust of potential customers. One way that customers determine whether or not to buy a product or service is through reading online reviews. However, it can be difficult to know which of these reviews are authentic. Products with a 5-star review may not actually have a true 5-star rating because of incentivized or paid reviews. This inflation makes it very difficult for firm's to have their products show up in user searches when another product has over a thousand fake, glowing reviews. Another type of fake review are those that are malicious and aim to destroy the online reputation of competitors. Those can be especially damaging to businesses, because according to a survey by Reviewtrackers, people are less inclined to buy a product if it has less than a 4 out of 5 star rating, and have been deterred from buying products due to a bad review.

Since 2015, Amazon has been working to crack down on the users who write paid reviews as well as companies that solicit them (Dwoskin and Timberg, 2018). However, in 2019 ReviewMeta found that even products which receive Amazon choice badges contain many suspicious reviews. Fake reviews pose a problem for both businesses and potential customers. How can companies compete with those that use paid reviews, and how can customers get a true estimate of the quality of product or service?

Here we develop a machine learning model that leverages Natural Language Processing (NLP) and the behavior of reviewers to automatically classify fraudulent reviews. By sifting through the reviews more efficiently, a machine learning model can help platforms like Amazon maintain high standards of quality in their marketplace.

Data

In this project we utilized data from an open Amazon dataset with over 130 million customer reviews collected between 1995 and 2015. It is available as TSV files on an Amazon Web Services S3 bucket. Reviews are grouped by product categories, and the apparel dataset with over 5.8 million reviews was used for this project. Each line in the dataset represents one product review. The columns in the dataset used for this project include:

customer ID - random identifier of author
review ID - unique ID of review
star rating - ordinal data ranging between 1 and 5
helpful votes - number of times a review was voted as 'helpful'
vine - a categorical value for whether or not a review was written by an Amazon vine member; vine is an invite-only program, which the most trusted reviewers are invited to
verified purchase - whether or not product was purchased on Amazon
review body - text of the review
review date - date review written

Analysis

To analyze the reviews and reviewer behavior, an exploratory data analysis was done to answer the following questions:

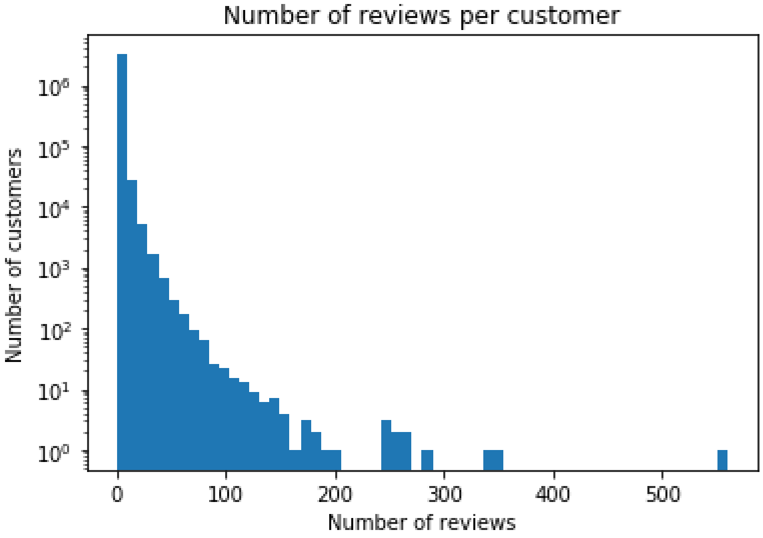
1. *What percentage of reviews are unverified?*
2. *What is the mean and median number of reviews per customer?*
3. *What do the reviews of a highly active (>500 reviews) reviewer look like?*
4. *What do the reviews from customers who post only once look like?*
5. *Do customers who write different amounts of reviews give the same distribution of star ratings?*

1) What percentage of reviews are unverified?

About 10.1% of the reviews were unverified purchases, meaning that the product was not bought through Amazon. A great majority of reviews were written by people who have actually purchased the product on Amazon. There could potentially be a correlation between unverified reviews and fake reviews, because anyone with an Amazon account could write an unverified review without actually purchasing the product.

2) What is the mean and median number of reviews per customer?

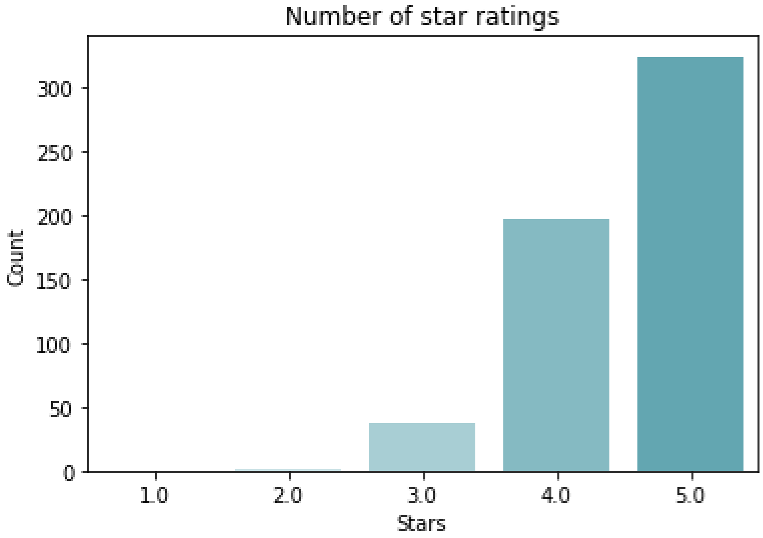
The mean number of apparel reviews that a customer writes is 1.83, while the median number of reviews per customer is 1. Because the average is larger than the median, it is evident that some customers are posting a very large number of reviews which is skewing the dataset to the right. The median number of reviews per customer is low, and could be a result of fake accounts where only one review is posted, or perhaps many Amazon reviewers do not review apparel products regularly.



68.4% of all customers in this dataset have written only one review. When those reviewers were removed, less than 1% (0.89%) of all customers wrote 12 or more reviews. Identifying the reviewers with the greatest number of reviews could be one way to find a suspicious reviewer.

3) *What do the reviews of a highly active (>500 reviews) reviewer look like?*

None of the 559 purchases were verified. The average star rating was 4.5, with just 300 5-star ratings, over 200 4-star ratings, and one 2-star rating.



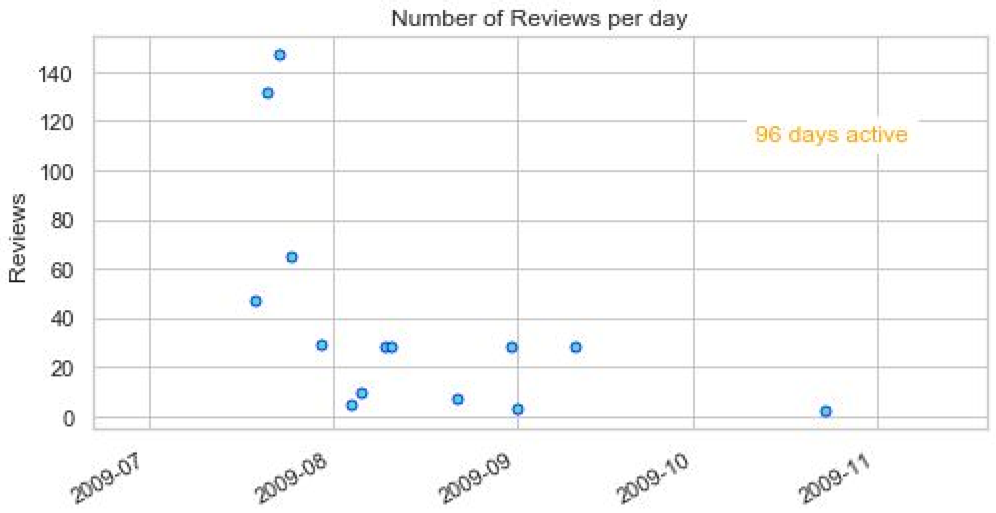
The low 2-star rating from the customer did not match the positive sentiment of the text. In this case, the text was an advertisement for the online store:

> *"Bollywood Style Designer Indian Kurti - Unique Indian Kurtis for Women -Summer Dress lucknowi-chikan-- It's the latest in Indian fashion! If you are looking to buy the designer kurtis (tunic) to match with your jeans, trousers or salwars, simply browse this store............."*

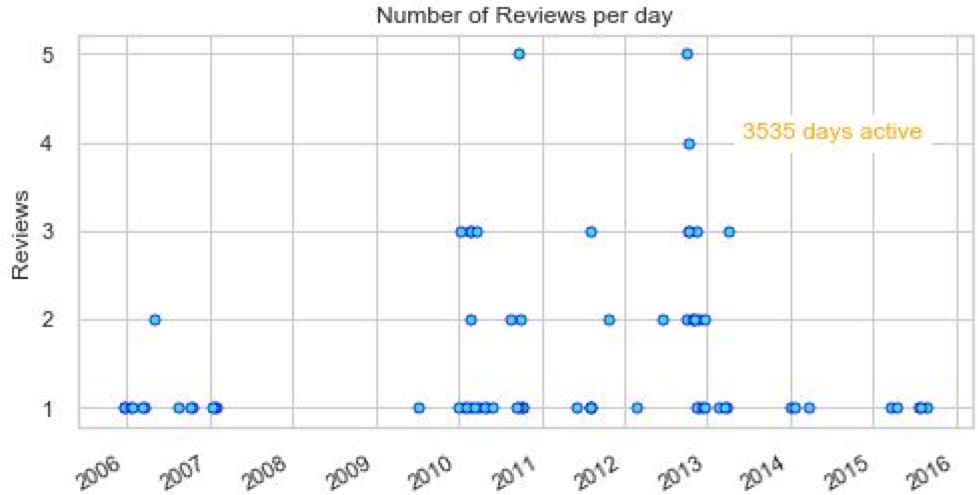
Many reviews from the most active customer reviewer were exactly the same. Below are four separate reviews:

> *“Stunning Cotton Kurti with gorgeous colored print. This ethnic kurti is master piece of Indian Ethnic Art.”*
>
> *“This classic cotton boho tier skirt are perfect for all long and short kurtis. A fun & fashionable skirt by Mogul Interior! This printed skirt offers a bohemian style and tie design Skirt is stylish, easy to wear, and is one of this season's hottest looks! Cotton.”*
>
> *“This classic cotton boho tier skirt are perfect for all long and short kurtis. A fun & fashionable skirt by Mogul Interior! This crinkle skirt offers a bohemian style and tie design Skirt is stylish, easy to wear, and is one of this season's hottest looks! Cotton.”*
>
> *“This classic cotton boho tier skirt are perfect for all long and short kurtis. A fun & fashionable skirt by Mogul Interior! This printed skirt offers a bohemian style and tie design Skirt is stylish, easy to wear, and is one of this season's hottest looks! Cotton.”*

The plot below shows the frequency and number of reviews posted over time by this highly active reviewer. This customer’s first review was written in July 2009, and last review in October 2009. The number of reviews written in a day starts very high and tapers off quickly and drastically within a month for other reviewers within this suspicious group. One way to flag suspicious activity is to identify a maximum limit of reviews written per day for each reviewer.

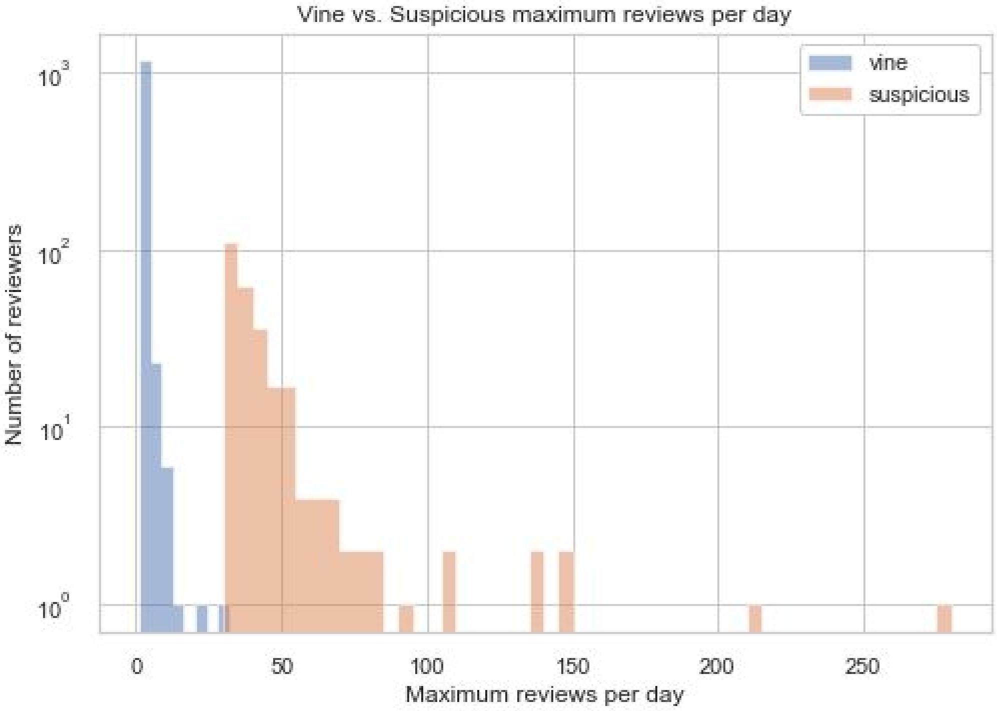


A large disparity can be seen between the distribution of the number of reviews posted in a day when compared to Amazon vine reviewers. The following image is a scatter plot of the number of reviews posted within a day by the vine reviewer with the most reviews.



The maximum number of reviews written within a day by this vine reviewer is 5. Because of the tremendous difference in review activity, the maximum number of reviews written within a day and being part of the Amazon vine program could be good indicators of whether or not a reviewer is suspicious.

The maximum number of reviews for the great majority of vine reviewers is much less than 25 reviews, therefore it might be reasonable to say that writing 30 or more reviews within a day is suspicious. Below is a histogram displaying the maximum number of reviews written within a day for vine and suspicious reviewers that posted 30 or more reviews within a day.



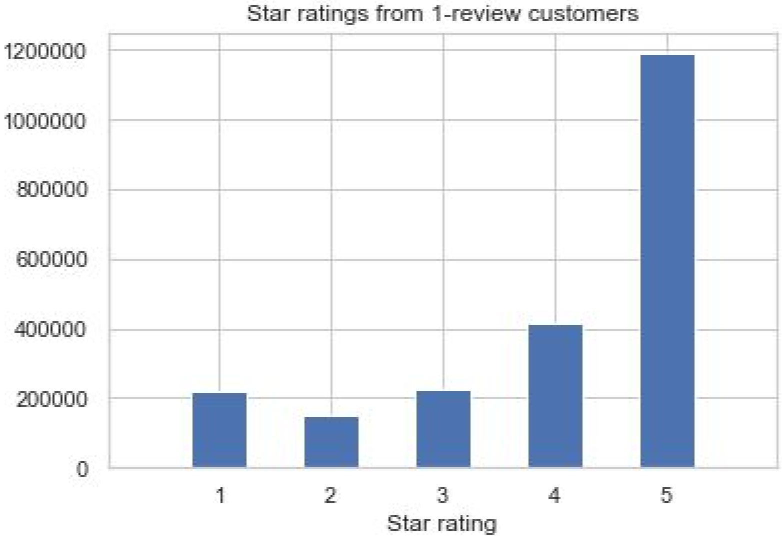
The maximum activity of vine reviewers is drastically less than the 'suspicious' reviewers. The most reviews written within a day by a vine reviewer was 32 in comparison with 280 reviews for a suspicious reviewer.

4) *What do the reviews from customers who post only once look like?*

There were 2,201,632 customers who left only one review (37.4% of all reviews) in the apparel dataset. A random sample of 50 reviews were chosen to inspect the reviews from customers that only posted one review. Most of the reviews in this subset were surprisingly thoughtful and unique, and did not seem to be computer generated. Even the reviews that were unverified and not voted as 'helpful' were surprisingly well-written:

> *"Fits true to size BUT don't be alarmed when you put them on the hips are a little tight. Trust me that they will loosen to a comfortable fit."*

The distribution of star ratings for the one-review customer were mostly 5-star ratings (54.1%) followed by 4-star ratings (18.9%). While most of these reviews were 5-star ratings, the amount of 1 through 4-star ratings was reasonable, within 1 order of magnitude.



5) *Do customers who write different amounts of reviews give the same distribution of star ratings?*

To compare, customers were segmented into group A if they wrote 1 review, group B for 2-5 reviews, group C for 6-9 reviews, group D for 10-12 reviews, and group E for 13 or more reviews.

Group	% of reviews	mean stars	median stars
A (1 review)	37.4	4.002	5
B (2-5 reviews)	40.1	4.130	5
C (6-9 reviews)	10.5	4.198	5
D (10-12 reviews)	3.6	4.228	5
E (13+ reviews)	8.4	4.280	5

Because the data is ordinal, and observations (reviews) can come from the same reviewer, a Kruskal Wallis test was used to determine if the star ratings have the same distribution across groups C, D and E.

H_0 : $mean\ rank_C = mean\ rank_D = mean\ rank_E$

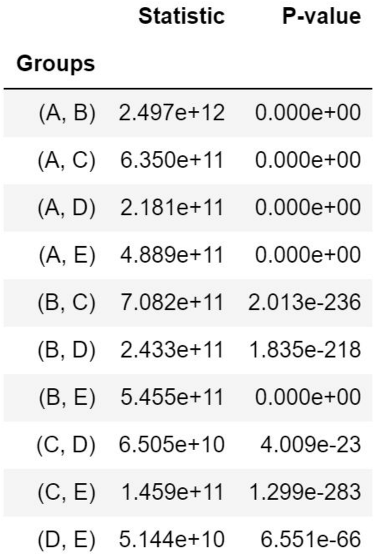
H_1 : Two or more of the groups do not have the same distribution

The Kruskal Wallis test returned a test statistic of 1298.9, and a p-value of 8.706e-283, and a p-value very close to zero, therefore the null hypothesis was rejected and we conclude that the star distributions from groups C, D, and E are significantly different.

To further analyze the differences between each pair of groups, a Mann-Whitney U test was used with the following hypotheses.

$H_0 : \ median_A = median_B$
$H_1 : \ median_A \neq median_B$

Results of the Mann-Whitney U tests are in the table below. P-values are all either equal to zero or very near to zero, therefore we reject the null hypothesis and conclude that each group has a median and distribution that is different from the other groups, and there are no two groups that are alike. Considering the star rating will most likely be important in building a machine learning model because customers with different types of behaviors tend to give different distributions of star ratings.



Groups	Statistic	P-value
(A, B)	2.497e+12	0.000e+00
(A, C)	6.350e+11	0.000e+00
(A, D)	2.181e+11	0.000e+00
(A, E)	4.889e+11	0.000e+00
(B, C)	7.082e+11	2.013e-236
(B, D)	2.433e+11	1.835e-218
(B, E)	5.455e+11	0.000e+00
(C, D)	6.505e+10	4.009e-23
(C, E)	1.459e+11	1.299e-283
(D, E)	5.144e+10	6.551e-66

Building a Machine Learning Model

Preparing the data

785 reviews without text in the review body were removed for Natural Language Processing (NLP). 784 reviews were from non-suspicious reviewers, and only one review from a suspicious reviewer.

Labeling ‘suspicious’ reviews

There existed no available data on which reviews were authentic or truly fake, so the information gained about suspicious behavior in the exploratory data analysis was used to create labels. The labeling of reviews enabled supervised methods of machine learning to find patterns of suspicious reviews in the text and customer behaviors.

The maximum number of reviews written within a day for the majority of trusted vine reviewers was much less than 30 reviews. Therefore, a conservative threshold of 30 or more reviews within a day was used to classify a review as suspicious or not suspicious. If a review was written by a customer who posted 30 or more reviews within a day, the review was given a label of 1, while reviews from non-suspicious customers were given a label of 0. The labels were highly imbalanced, with 15,360 reviews from suspicious customers, and 5,865,729 reviews from non-suspicious customers. Based on the threshold, only 0.26% of reviews were labelled as ‘suspicious’.

The criteria for being labeled as suspicious is meant to be solely a starting point and underestimates the true prevalence of fake reviews, as it will miss other types of suspicious behavior such as the creation of new accounts to post a single review.

Features

The features of this model included the text from the review body, star rating, number of helpful votes, vine and verified purchase status. Feature engineering was incorporated to create a cosine similarity score between 0 and 1 for each reviewer. A cosine score of 0 is given to the reviews of customers who write unique, or non-similar reviews, whereas a score of 1 is given to the reviews of customers who have the exact same text in two reviews. This is how it works, suppose there is a customer wrote two reviews:

(Review 1) “I like this shirt.”
(Review 2) “Nice shirt.”

These reviews must be transformed into numbers and vectors, so TF-IDF (term frequency - inverse document frequency) vectors are created. First, the words in each review are separated, this is called ‘tokenization’. Then, each unique word that was used in all of the user’s reviews is placed into a vocabulary. In this case, the vocabulary would be:

[‘i’, ‘like’, ‘nice’, ‘shirt’, ‘this’].

Next, the term frequency (*tf)*, ratio of the number of times a word appears within a review to the total number of words in the review, is calculated.

The first review becomes:

<1/4, 1/4, 0, 1/4, 1/4>

and the second review becomes:

<0, 0, 1/2, 1/2, 0>

Following, the inverse document frequency (*idf*) is calculated with the default scikit-learn formula:

$$idf_i = ln \frac{N+1}{df_i+1} + 1$$

where *N* is the total number of reviews and df_i is the number of reviews that a word is in. The final calculation is the product of $tf_i \times idf_i$.

Word	Review 1 *tf*	Review 2 *tf*	*idf*	Review 1 *tf x idf*	Review 2 *tf x idf*
'i'	1/4	0	$ln \frac{2+1}{1+1} + 1 = 1.405$	0.351	0
'like'	1/4	0	$ln \frac{2+1}{1+1} + 1 = 1.405$	0.351	0
'nice'	0	1/2	$ln \frac{2+1}{1+1} + 1 = 1.405$	0	0.703
'shirt'	1/4	1/2	$ln \frac{2+1}{2+1} + 1 = 1$	0.25	0.5
'this'	1/4	0	$ln \frac{2+1}{1+1} + 1 = 1.405$	0.351	0

The last two columns are the transformed tf-idf vectors for the reviews, and those two vectors can then be used to find the cosine similarity.

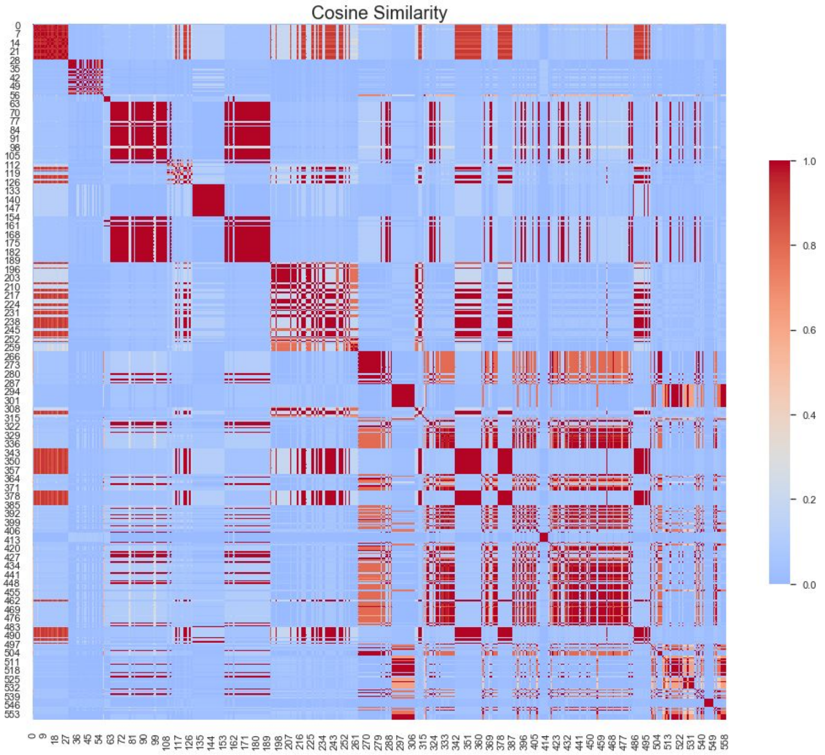
Cosine similarity $= \frac{a \cdot b}{|a||b|}$

$$= \frac{(0.351)(0) + (0.351)(0) + (0)(0.703) + (0.25)(0.5) + (0.351)(0)}{\sqrt{0.351^2+0.351^2+0^2+0.25^2+0.351^2} \cdot \sqrt{0^2+0^2+0.703^2+0.5^2+0^2}}$$

$$= 0.220$$

Imagine that the second review did not contain the word 'shirt' and that they no longer have any words in common. The numerator of the cosine similarity calculation would be equal to 0, and therefore the cosine similarity would be 0, which indicated that the reviews are not similar. If two reviews are identical, they would receive a cosine similarity score of 1.

Word for word repetition of reviews from the same customer can be seen in a cosine similarity matrix. The cosine similarity matrix below shows the similarity score of reviews of the most active customer who posted a total of 559 reviews in the dataset.



In this matrix, each row represents a review, and the color of each cell is the similarity to other reviews that this customer has written. The score ranges between 0 and 1, where a score of 0 represents very different reviews, and a score of 1 represents an identical review. It is expected that the diagonal will be red (have cosine similarity of 1) because reviews along the diagonal are the similarity of a review with itself. Large blocks of red cells off of the diagonal show that there are many reviews that are identical.

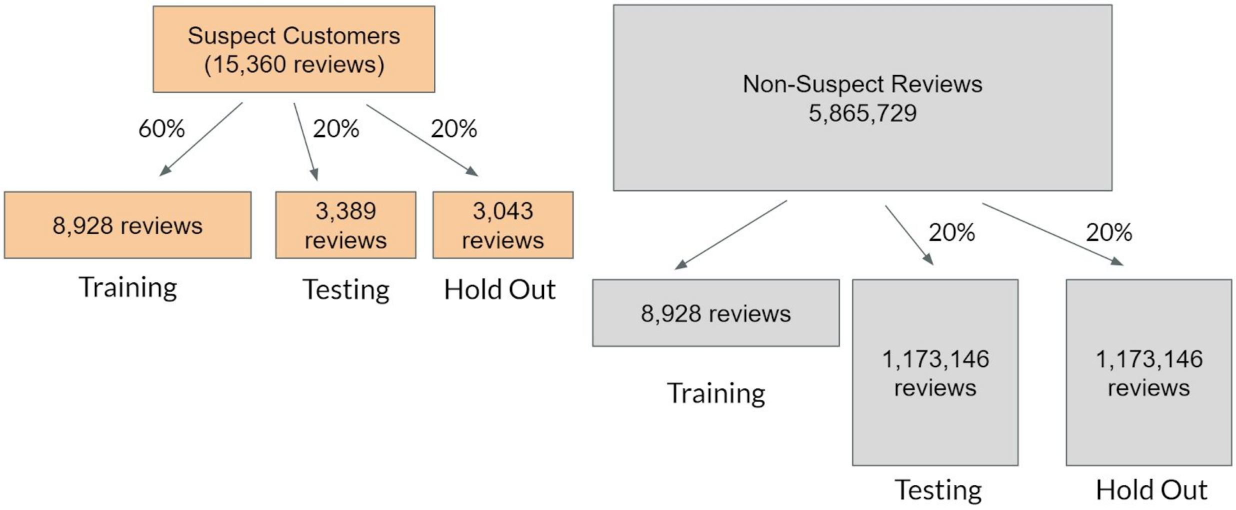
The cosine similarity metric used in the machine learning model was the average of the highest two cosine similarity scores from the particular user's reviews. From the customer's matrix above, all of the reviews written by this customer would receive a score of 1 because there are at least two scores equal to 1 off of the diagonal. In the case that a reviewer wrote only one review in total, the review received a 0.

Splitting data into training and test sets

Each row represents a review in the dataset, so data is not independent because reviews have the potential to be written by the same person, or for the same product. Having reviews from suspicious reviewers in both training and test sets would create a biased model, therefore the data was first split by reviewers/customer ID.

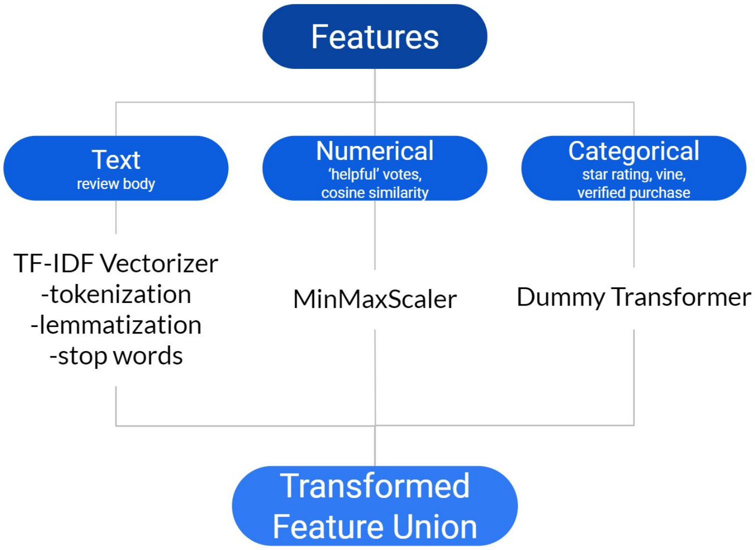
Suspect and non-suspect reviews were separated. There were 268 unique customers labeled as 'suspect' that wrote a total of 15,360 reviews. The reviews from 60% of the suspect customers were used for training (8,928 reviews), 20% of suspects were used for model testing

(3,389 reviews), and 20% of suspects were used for a hold out set (3,043 reviews). An equal number of suspect and non-suspect reviews (8,928 reviews) were used for training the model, while 20% of the non-suspect reviews (1,173,146 reviews) were used for the testing and hold out setset. This yielded a 0.29% rate of fake reviews in the testing set and 0.26% rate of fake reviews in the hold out set, very close to the total rate of 0.26% for the entire dataset.



Preprocessing pipeline

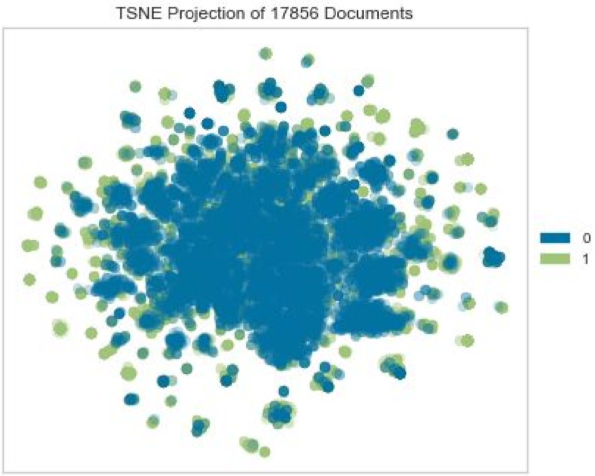
After data was split into training, testing and hold out sets, the datasets were fed into a FeatureUnion pipeline to transform and combine the different types of features. Text from the review body was vectorized with TfidfVectorizer, numerical features were scaled with MinMaxScaler, and categorical features were transformed into dummy variables. Vectorization of the review body included tokenizing the review, lemmatization, and removal of stop words and words that appeared in less than 5 reviews.



Natural Language Processing

Like in the feature engineering section, customer reviews were converted into TF-IDF (term frequency - inverse document frequency) vectors. However, instead of building a vocabulary from a particular user's reviews, the vocabulary was built from all reviews within the split dataset. Tokenization was used to break up each review by unique words, and stop words that have little meaning such as 'a', 'the', 'she' or 'he' were removed with the 'english' dictionary provided by TfidfVectorizer. To account for the varying forms of a word with the same general meaning, lemmatization was implemented to transform words into their root word (for example, 'bad' and 'badly' would both be converted to the root word 'bad').

The TF-IDF score is directly proportional to the word frequency in a specific review, and inversely related to the word frequency across all documents. As a result, words that are very common in other reviews receive a lower score than the words that are used infrequently. The length of each vector is the total number of unique words from the vocabulary, so as you can imagine, each vector contains many zeros because there are thousands of words in a corpus that are not in any given review.



The image to the left is a TSNE (t-distributed stochastic neighbor embedding) projection of the TF-IDF vectors of non-suspicious (blue, label '0') and suspicious (green, label '1') reviews in the training set.

While there is heavy overlap of the non-suspicious and suspicious reviews, there are quite a few suspicious reviews on the outer rim of the projection. This indicates that TF-IDF vectors can inform us about the difference in word patterns between suspicious and non-suspicious customer reviews.

Model selection

The binary classifiers that were trained and tested include Multinomial Naive-Bayes (α=1.0), Random Forests (200 estimators, max depth=10), and Linear SVC (L2 penalty, C=1.0) because of their ability to manage large amounts of observations and features. Furthermore, these models were found to be the top choices in previous research studies in the detection of fake reviews.

Model testing metrics

Multinomial Naive Bayes, Random Forest and Linear SVC models were tested using various combinations of features including text features only, numerical and categorical features only, and all features combined. For firms to detect fake customer reviews from millions of reviews, it would probably be best to balance precision and recall to limit the time spent in verifying that the reviews which are classified as suspicious are truly suspicious.

Recall is the proportion of suspect labels which the model was able to successfully classify. This is clearly going to be an important metric because the goal of the app is to identify fake reviews. While precision is important to consider because we want to know that a good proportion of reviews that the model is saying is suspicious is actually suspicious (the model is not classifying every review as suspicious). Precision is important but not as important as recall since there could potentially be suspicious reviews that were not labeled as suspicious. In other words, false positives might actually contain reviews that were not classified as suspicious.

$$Precision = \frac{True\ positives}{True\ positives + False\ positives}$$

$$Recall = \frac{True\ positives}{True\ positives + False\ negatives}$$

To weight recall as more important than precision, the F2 measure can be used. It is obtained from the F_β measure formula and a beta value of 2.

$$F_\beta\ measure = (1 + \beta^2) \times \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

$$F2\ measure = 5 \times \frac{precision \cdot recall}{(4 \cdot precision) + recall}$$

The Random Forest, Naive Bayes and Linear SVC classifiers contain different probability distributions and default thresholds for the suspect class. The default threshold for Naive Bayes and Random Forest classifiers is 0.5, such that any probability greater than or equal to 0.5 for being ‘suspicious’ would be predicted as ‘suspicious’ and anything under 0.5 would be predicted as ‘non-suspicious.’ The default threshold for a Linear SVC uses distances from the separating hyperplane. If it is on the positive side (greater than zero), review is classified as ‘suspicious’ and conversely if it is on the negative side (less than zero), review is classified as ‘non-suspicious.’

As a result of the varying distributions of probabilities, other metrics that were used to compare models were the Receiver Operating Characteristic Area Under the Curve (ROC AUC) and Precision Recall (PR) AUC because they are a summative measure of the models across all thresholds. The ROC curve plots the true positive rate versus the false positive rate across all thresholds, and will allow for comparison on how well the models will do in general with classifying suspicious reviews.

Model testing

The ROC AUC, PR AUC, recall and F2 measure shown below were obtained by using the default thresholds for classification in each model on the test set.

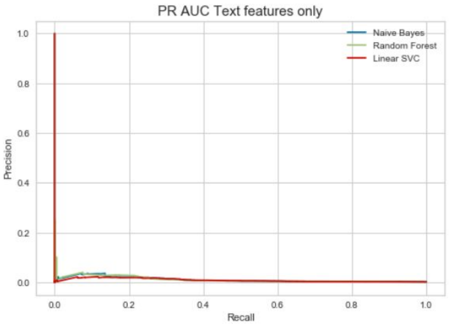
Features in Model	Number of Features	Naive Bayes	Random Forest	Linear SVC
Text only	2813	ROC AUC: 0.726 PR AUC: 0.012 Recall: 0.437 F2: 0.037	ROC AUC: 0.723 PR AUC: 0.012 Recall: 0.581 F2: 0.028	ROC AUC: 0.692 PR AUC: 0.010 Recall: 0.463 F2: 0.034
Numerical and categorical features	6	ROC AUC: 0.975 **PR AUC: 0.300** Recall: 1.00 F2: 0.043	ROC AUC: 0.980 PR AUC: 0.291 Recall: 0.998 F2: 0.087	**ROC AUC: 0.987** PR AUC: 0.176 Recall: 1.00 F2: 0.044
Text, numerical, and categorical features	2819	ROC AUC: 0.928 PR AUC: 0.048 Recall: 0.962 F2: 0.042	ROC AUC: 0.960 PR AUC: 0.085 Recall: 0.929 **F2: 0.142**	ROC AUC: 0.947 PR AUC: 0.049 Recall: 1.00 F2: 0.044

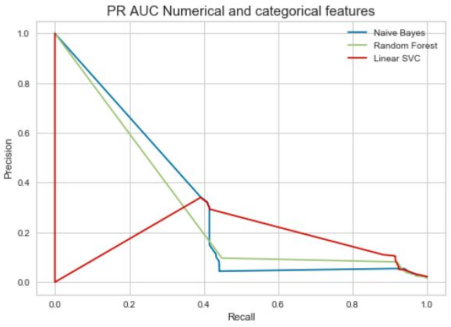
The Naive Bayes, Random Forest and Linear SVC model using numerical and categorical features had the highest ROC AUC and PR AUC scores. While the recall is high in the models, the F2 measures were quite low due to poor precision. The low F2 measure indicates that the model is actually classifying too many reviews as suspicious, also classifying non-suspicious labeled reviews as suspicious. However, as mentioned earlier, it is highly likely that fake reviews were not labeled as suspicious, because only one criteria of posting 30 or more reviews within a day was used to create the labels.

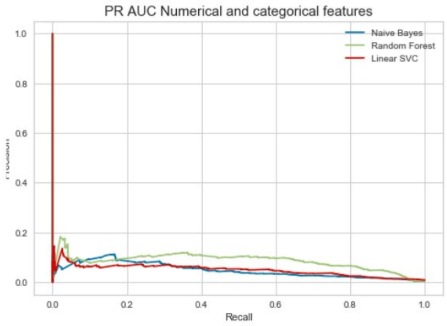
The next highest scoring model was the Random Forest with all features. It scored the highest F2 measure with default thresholds, and resulted in a good ROC AUC score of 0.960. The PR AUC score was much lower than the models which used only numerical and categorical features.

The lowest performing models used text as the only feature. These results show that it was difficult for the model to predict suspicious reviews solely on word patterns, and that the addition of customer behaviors substantially improves predictive power.

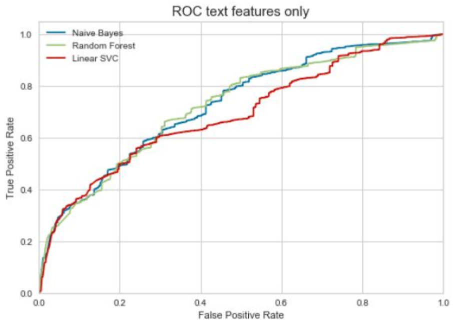
The overall higher precision and recall in models with numerical and categorical features can be observed below. Precision is very low in models with text features due to the large number of text features. When text features are included, the number of features increases by 2813.

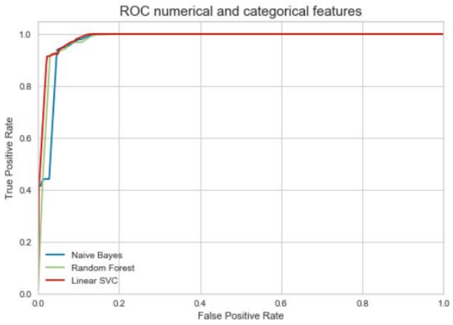


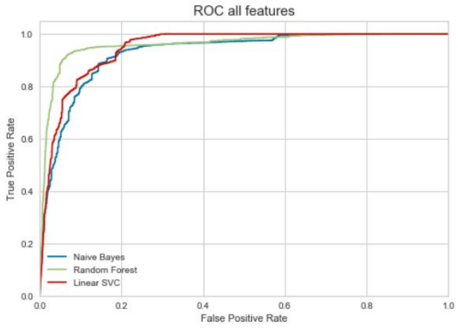




The ROC curves exhibit the significantly higher performance of models that include numerical and categorical features compared with models that use text features only.



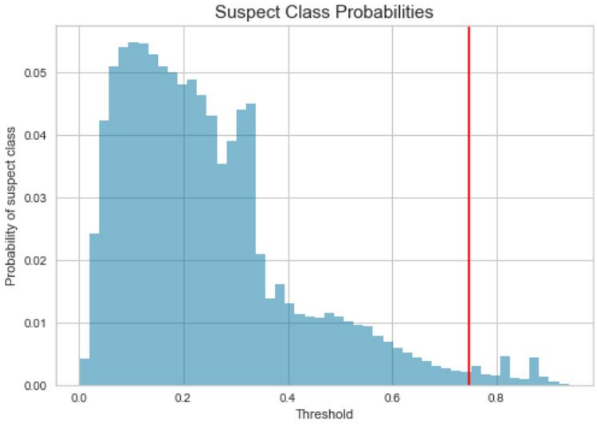


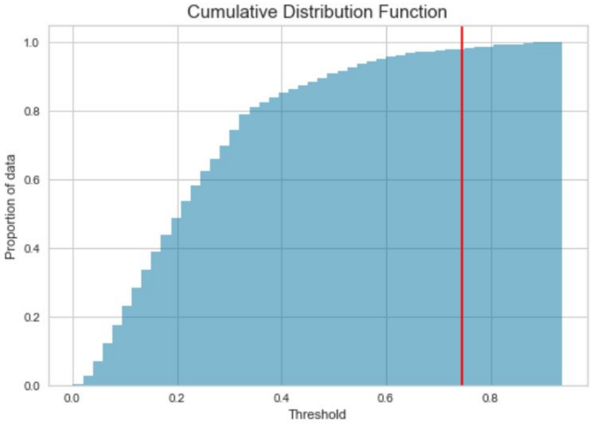


Model tuning

In the case of fake review identification for a firm, a model balanced in precision and recall is most likely desired, where we want to catch as many fake reviews as possible, but we also do not want to classify too many authentic reviews as suspicious (false positives). In order to balance the precision and recall of the model, the thresholds of each model were tuned on the test set for maximum F2 measure.

Hyperparameter and threshold tuning was completed on the test set with the four best models from the previous section: Naive Bayes, Random Forest and Linear SVC with numerical and categorical features, and Random Forest with all features. GridSearchCV was used on the training set for hyperparameter tuning of the number of trees (‘n_estimators;) and maximum tree depth for the Random Forest including all features. The optimal number of trees was 200 with a maximum depth of 30, and the classification threshold of 0.746 yielded the best F2 measure.





Using the tuned hyperparameters and threshold for maximum F2 measure, model predictions were made on data that the model has not seen in the hold out set.

Model	Number of Features	Test Set	Hold out set
Naive Bayes (numerical and categorical)	6	ROC AUC: 0.975 **PR AUC: 0.300** Recall: 1.00 F2: 0.043	ROC AUC: 0.964 PR AUC: 0.067 Recall: 0.156 F2: 0.116
Linear SVC (numerical and categorical)	6	**ROC AUC: 0.987** PR AUC: 0.176 Recall: 1.00 F2: 0.044	**ROC AUC: 0.982** PR AUC: 0.071 Recall: 0.073 F2: 0.069
Random Forest (numerical and categorical)	6	ROC AUC: 0.980 PR AUC: 0.291 Recall: 0.998 F2: 0.087	ROC AUC: 0.966 **PR AUC: 0.107** **Recall: 0.862** F2: 0.250
Random Forest (all features)	2819	ROC AUC: 0.960 PR AUC: 0.085 Recall: 0.929 **F2: 0.142**	ROC AUC: 0.974 PR AUC: 0.075 Recall: 0.658 **F2: 0.283**

The model returned an ROC AUC score of 0.974, recall of 0.658 and F2 measure of 0.283. The F2 measure doubled and AUC ROC of the model increased by 0.014, but recall decreased by

0.271. Although the model did not catch as many suspicious labels in the test set, it was more precise in its classification of ‘suspicious’ reviews. This model returned the highest F2 measure, indicating that it was the most balanced model in terms of recall and precision.

Furthermore, this model showed the least difference between AUC scores in the test and hold out sets, which means that it is the most generalizable. The three other models that used numerical and categorical features resulted in larger drops in PR AUC scores and decreases in ROC AUC scores due to overfitting.

Summary

The Random Forest classifier with a combination of text, numerical and categorical features had the best overall performance when considering ROC AUC, precision and recall. The F2 score was generally quite low due to poor precision, meaning that the classifier is quite optimistic in its classification of suspicious labels. However, a caveat to consider when judging the precision is that the labels are not pure, consequently reviews that were not labeled as suspicious might actually be fake.

Future studies should consider additional criteria for labeling reviews as suspicious or using repeated sampling from the suspect class. Having a greater number of targets can help to increase the precision of the model.

Recommendations to improve performance:

- Increase target labels by adding criteria or using repeated sampling.
- Refine the cosine similarity feature
  - Take emoticons into account by transforming them into text first.
  - Calculate the similarity of each review with the review previously written by the customer.
- Add features
  - User activity time. Trusted vine reviewers tended to post reviews over long periods of time (years), while suspicious reviewers usually posted over a duration of a few months.
  - Sentiment analysis - is the review positive or negative? Does the sentiment match the star rating?

The best results were achieved with the utilization of text from customer reviews in combination with customer behavior metrics in a Random Forest model, where 65.8% of fake reviews were found. Identifying fake reviews can be like finding a needle in a haystack due to the large volume of reviews, and relatively sparse fake reviews. However, machine learning makes it possible to filter and detect patterns of fraudulent reviews so that firms will be able to maintain a high quality platform that customers and businesses can trust.