Fake review Detection in Yelp

Team Hatomugi:

Yu Miyauchi

Objective

❖ Determine if the review in Yelp is fake or real

Given data: Over 300000 rows with about 6 columns.

Important columns: Date, user_id, product_id, rating, and review

Target variable: Real review = 1 (90%), Fake review = -1 (10%)

Workflow

Domain knowledge (Put myself in fake reviewers' shoes.)
"Why do I want to write fake reviews and how?"

↓

Exploratory Data Analysis → Statistical test → Insight

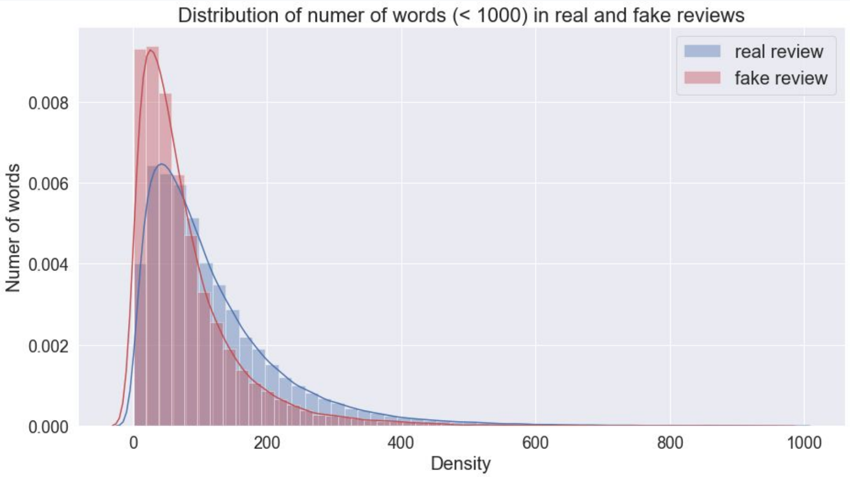
↓

Feature Engineering (including NLP)

↓

Build machine learning models

Distribution for number of words in review



Domain knowledge:
It may be painful to write reviews that are fake, so fake reviews should have fewer words.

↓ True!

1. **Fake review has fewer words.**

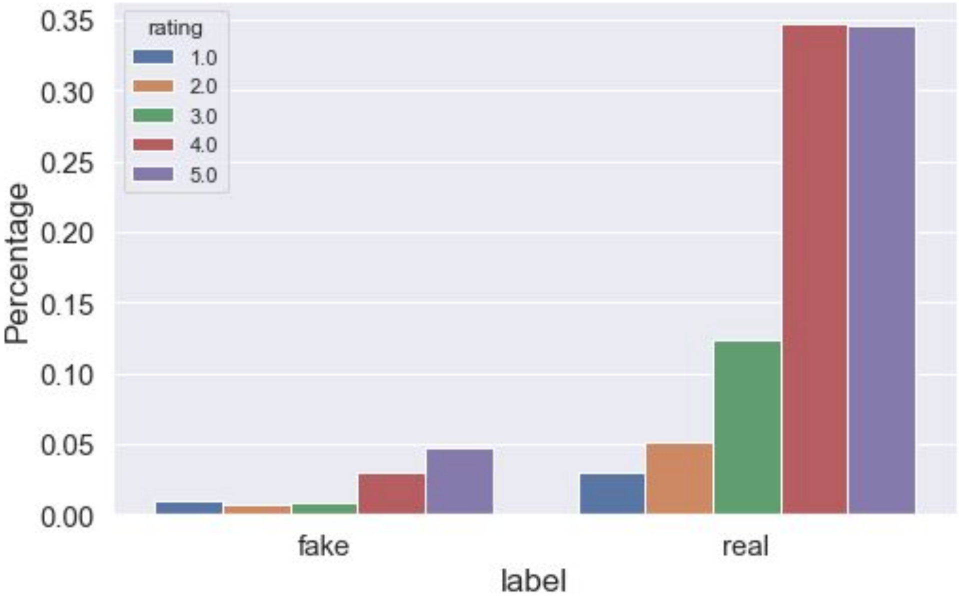
   Noise? (given sample size = thirty thousand)

   Kolmogorov-Smirnov test (KS-Test)
   Ho: Two distributions are from same pop. dist.
   Ha: Ho is not true

   p-value = 0.0
   → There is difference in two distributions.

Worth adding number of words as feature.

Distribution of rating for fake and real review



Domain knowledge:
Companies or individuals write very bad reviews for rival companies and very good reviews for theirs.

True ! ↓

1. Fake review is more likely to get review=1 or 5

Noise? (given sample size = thirty thousand)

↓

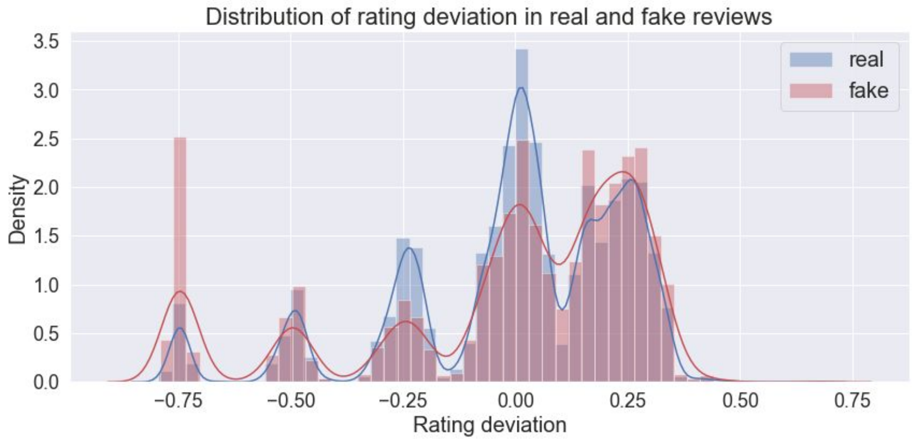
Chi-square independence test

p-value = 0.0

→ Some relationship between label and rating

Worth adding rating as feature.

Distribution of rating deviation



Domain knowledge:
∃ Many bad reviews
→ Desire to make fake reviews with good rating

∃ Many good reviews already
→ No need to make fake review with good rating

Rating deviation = deviation from mean rating of that product

↓True!

1. **Fake review has either very positive deviations or very negative deviations and true review has values close to 0.0 deviations .**

There is difference in two distributions.

Worth adding it as feature.

Volatility (Burstiness) for count of reviews

Volatility - liability to change rapidly

Domain knowledge:

It is suspicious if one restaurant gets two reviews on average in one day, but suddenly started to get twenty reviews in three days.

Volatility (Burstiness) for count of reviews

Volatility (window = 3 days)
Real review: 1.216
Fake review: 1.285

Volatility (window = 7 days)
Real review: 1.115
Fake review: 1.1711

Volatility (window = 14 days)
Real review: 1.069
Fake review: 1.116

Fake review has more volatility of count for every single windows

= Gets more number of reviews suddenly within some (window) days

↓

Noise? (given sample size = thirty thousand)

↓

Use mann whitney-u test.
(Normality wasn't satisfied for population distribution)

↓

P-value ≈ 0.0

↓

There is difference between mean values of volatility.
Worth adding it as feature.

Subjectivity and objectivity of review

1. How many first person pronoun there are (I, my, me, we, our, us ...)

   Mean ratio:
   Real review : 4.01% of sentence
   Fake review: 3.77% of sentence

   Mann whitney-u test results in p-value=0. (Sample size = thirty thousand)

   Since fake users don't have real experiences, they tend to use slightly less first person pronouns?

2. Check emotion
   → Sentimental score from review (negative, neutral, positive, and compound)
   I found out that there were some differences! Worth adding it as feature.

Other questions

1. Does fake account only post fake reviews or also real reviews?

   → **99.59% of users post either fake or real reviews but not both.**

   User id is very important. If one person writes fake review, almost all reviews from this person are also fake.

2. Number of reviews per real and fake accounts.
   Real account : **2.43 reviews** on average. (322167 samples)
   Fake account: **1.29 reviews** on average. (36885 samples)

   Fake account gets deleted? Fake account is only dedicated to write review for only one product?

   Count feature per user id may be important (how many reviews that account wrote).

Other questions

3. More desire to make fake reviews when there are less reviews for that product?

Because 1. A few bad reviews could affect their rating largely (unstability).
2. They just want to increase number of reviews to get more customers.

True!

Product that contains REAL reviews : 349.043 reviews on average.
Product that contains FAKE reviews: 40.356 reviews on average.

Product that contains fake reviews have less number of reviews.

Count feature per product_id should be good feature.

Model (LightGBM)

Two reasons

1. Data is large (Total data used is about 80000 rows)
   → Need fast model like LightGBM
   (Faster than XGBoost with about same accuracy)

2. Gradient Boosted Decision Trees (GBDT) models have better accuracy than any other ml models for most of times and are near equal to have same accuracy with deep learning models in tabular data if feature engineering is properly applied.

Model (LightGBM)

Target is adjusted. (Fake review=50%, Real review=50%).

To get more insights and research about features (Which features improve better to distinguish fake and real reviews?) but not to get practical machine that works in real life.

1. Use time series split to prevent target leakage (= model learns future values).

   (* When we used Stratified KFold, the test accuracy dropped largely comparing to validation accuracy which showed that there is some target leakage if we don't use time series split.)

2. Numerically encode product id and user id b/c it has high dimensionality.
   Since tree models use inequality signs, it is able to distinguish them correctly.
   (Accuracy tends to increase also in this way when the feature has high dimensionality.)

Result (Target is adjusted)

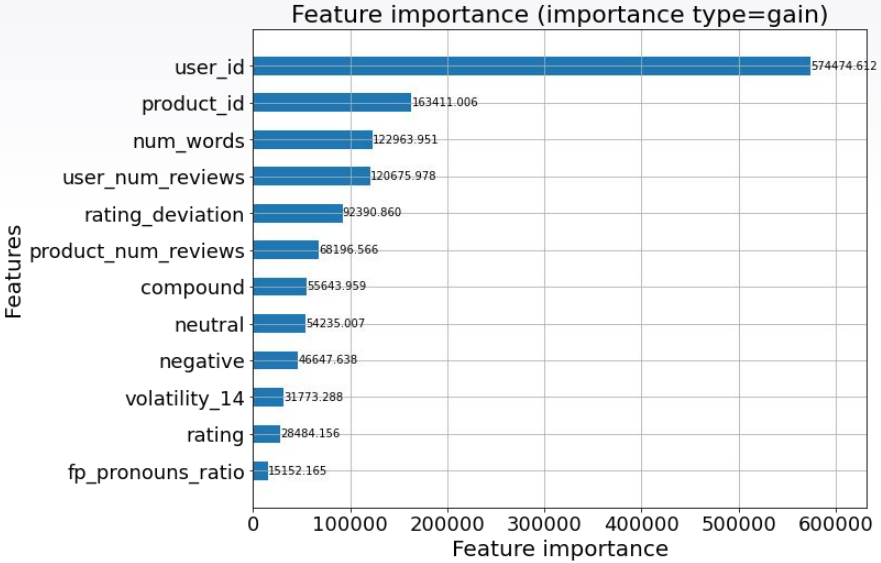
Randomly sampled.
Fake:50%, Real:50% Train: 2005~ 2013 (70% of data) Validation: 2014~2015 (30% of data)

	F1 Score	Binary Log loss	Accuracy
Three basic features (user_id, product_id, rating)	0.694	0.613	0.653
+ Sentiment score	0.706	0.595	0.669
+ Other created features	0.714	0.582	0.680

Since target is evenly distributed, we focus on binary log loss and accuracy here.
Model is able to predict target accurately for 68% of times.

Nice improvement by adding created features!

Feature Importance



Some points

1. User id is the most important.
   b/c 99% of user only posts either fake or real review but not both.

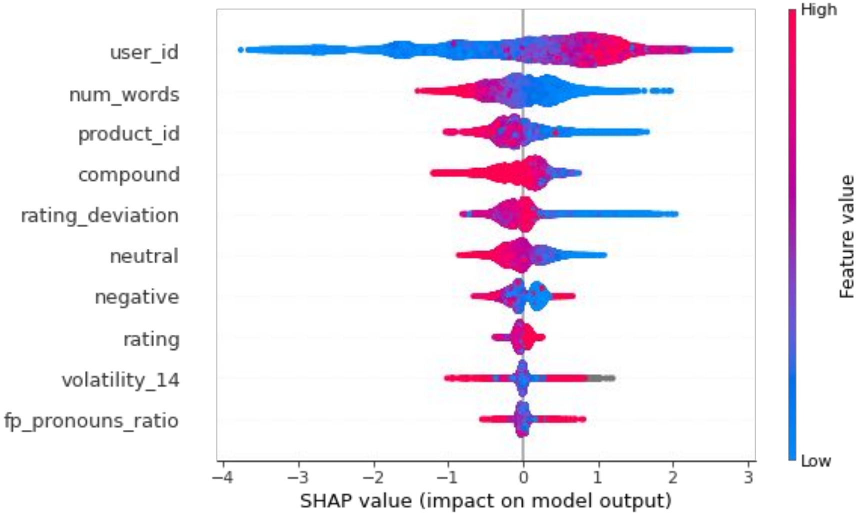
2. Number of reviews per product and per user have higher importance.

3. Sentiment (emotion) score has good amount of importance as well.

SHAP values

SHAP describes which feature affects target how much in the model.

SHAP values ∝ probability of being fake



Real review: label = 0 Fake review: label = 1

* Compound is the sum of all feelings

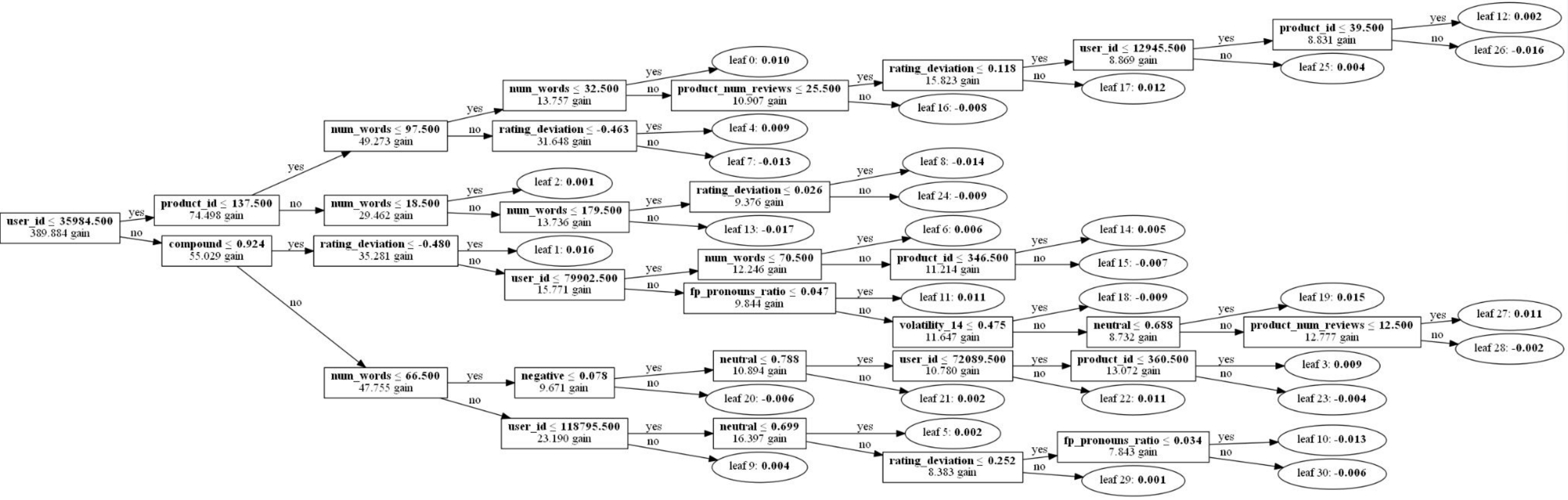
Interesting points

1. Probability of being fake review (=1) increases as the number of words gets lower.

2. When rating deviation is very low (< 0.0), the probability of being fake increases. (Ex: Let's say average rating of restaurant A is 4.0. If the rating is 1.0, the probability of being fake review is higher. (Rating deviation is negative value))

3. When neutral feeling is low, the probability of being fake increases while it decreases when neutral feeling is high. If the user review contains extreme feelings, it is more to be fake review.

Tree split of the first tree boosted by LightGBM



1. Num_words, rating deviation, and sentimental score plays significant role.

* Value for user_id is irrelevant. Tree model is just trying to narrows down fake users using inequality signs.

Thank you for listening!