Zekai Hu, Bofei Xu (EDUC 244), [Esther Jeongwon Choi](mailto:estherchoi4@berkeley.edu), [Kylie Ren](mailto:kxyr@berkeley.edu), [Serena Bai](mailto:serena_818@berkeley.edu)(DATA 144)

Instructor: Professor Zachary Pardos

Subject Name: Depop Scam Listing Identification

05 December 2023

Final Project Report

# **INTRODUCTION**

We seek to identify the scammers/scamming listings on E-commerce platforms, using machine learning methods. Our project group decided to train our model on Depop, the secondhand ecommerce platform, due to our members’ familiarity with the service and user interface. Many users on the platform get scammed due to buying fake listings, and although Depop promises a refund for users who have purchased items through their service and have been scammed, many transactions still occur over Paypal or Klarna or other outside financial transactional services. Our stakeholders would be both the users, who have a personal interest in the verification of the products they’re buying, and the Depop corporation itself, as the high volume of scammers affects not only the reputation of the platform but also their profits as the scam purchases aren’t making them any money.

**DATASET**

In order to detect scam users, our project group collected data from Depop users, both real and fake. We hand-labeled the users according to criteria tabulated in Table 1. The project members identified scam users and products on the website, the details of which were then scraped using Python and the Cloudscraper module, which bypassed any bot detection by Depop. We developed our own scripts for username scraping, and then adopted the product and user variable scraper from github by user [Gertje823](https://github.com/Gertje823/Vinted-Scraper/commits?author=Gertje823) (https://github.com/Gertje823/Vinted-Scraper/tree/main). The scraper also included a rate limiter and a sleep time in order to not overwhelm the Depop servers with API requests. The collected data had features about the users such as positive feedback count and the last logged-in time, as well as features describing the various projects on their page such as price, last updated, product descriptions, etc. The full list of both user and product features are tabulated in Table 2. For this project, we annotated 234 users, of which 162 were labeled as established users (Class 0), 12 were labeled as fake users (Class 1), 60 were labeled as new users (Class 2). All products of these users, including selling and sold, sums to a number of 171,910, of which 165,028 were from established users, 4,682 were from fake users, 2,200 were from new users.

Table 1. User Labeling Criteria

| **Established Users (Class 0)** | **Fake Users (Class 1)** | **New Users (Class 2)** |
| --- | --- | --- |
| many sold products | few or no sold products | few or no sold products |
| many good reviews | 0 good reviews | 0 to 2 good reviews |
| social media linked | no other form of ID | social media linked |
| bought products previously | new user | bought products previously |
| reasonable pricing | often considerably lower than market | Reasonable pricing, occasionally deviates from market due to their lack of experience |
| consistent photo background | inconsistent photo background (photos stolen from others) | consistent photo background |

After some initial data analysis, our group realized that a significant portion of the predictions was affected by new sellers (Class 2), which can be more clearly defined as users who have only recently started selling and only have a few products sold, if any. To be more specific, a new seller might not necessarily be a new user on the platform, but their lack of sold products were leading to a less accurate model because they were being falsely predicted as fake sellers. A few of the differences between each class is encapsulated in the Table 1 above but explained more in depth in the following sentences. Many of these new sellers were verifiable as real users, whether because they had social media in their profile’s bio or because they had purchased items from other sellers and received good quality reviews from them. Some other key features of new sellers that differentiated them from established sellers (Class 0) and fake sellers (Class 1) were if they had received less than two good reviews, sold only a few products or if they had migrated from another E-commerce platform such as Poshmark or eBay (but they were selling on Depop under the same username and photos). In these cases we want to not only make a distinction between fake users that might be bots or scammers and the new sellers trying to sell their products, but also between established users with more than a few good reviews and many sold products.

Table 2. Available Variables Considered in This Project

| User Variables | Product Variables |
| --- | --- |
| Username  User\_id  Bio  first\_name  followers  following  initials  items\_sold  last\_name  last\_seen  Avatar  reviews\_rating  reviews\_total  verified  website | ID  Sold  User\_id  Gender  Category  Size  State  Brand  Colors  Price  Image  Description  Title  Platform  Address  discountedPriceAmount  dateUpdated |

**EDA AND METHODS**

After an initial EDA, our project group decided to train the model for classification on a product level, for which the labels were “inherited” from their sellers. We decided to use features that we noticed in our initial runthrough of Depop for scam users that were key identifiers, such as suspiciously low prices, negative reviews, lack of sold products, and so on. By looking at a subset of raw data, we realized that there was an enormous amount of words in product descriptions when combined so we decided to featurize the description with unigrams, which will be discussed further in the individual contribution sections. We tabulated all the features considered by the tree-based models in Table 3.

Table 3. Features Considered by the Tree-based Models

| Variable Names |
| --- |
| 'followers', 'following', 'items\_sold', 'reviews\_rating',  'reviews\_total', 'size', 'Price', 1000 BOW features |

For more details on the tree-based models, please refer to Kai’s individual contribution section.

Following the tree-based models, we implemented a Neural Net with only the product description textual data as the input variable. For the motivations behind this, modeling details, and result interpretations, please refer to Bofei’s individual contribution section.

**INDIVIDUAL CONTRIBUTIONS**

# **Individual Contribution (Kai):**

**Data Scraping:**

Collaborating with Bofei, I contributed to the modification of the Depop Scraper script, enabling the extraction of user data from Depop and storing it in a SQLite database.

**Data Processing:**

Data Merging:

To consolidate the efforts of all team members, I oversaw the integration of scraped data, stored in SQLite format, from various contributors. Employing SQL, the necessary data was extracted and merged into a comprehensive dataframe (df\_all). This dataframe includes a Depop user table and a table containing user-listed products. Annotation data, classifying users as real or fake, was incorporated based on their usernames. If a user was flagged as fake, all products listed by that user were likewise labeled as fake, as most legitimate sellers wouldn’t list fake products among their real ones.

Data Processing and Feature Engineering:

Working with the merged dataframe, I ensured uniform data types, dropped irrelevant columns, and applied one-hot encoding for categorical features. For product descriptions, a Bag of Words approach was employed, wherein the top 1000 most common words were converted into binary features. To streamline the dataset, columns with high missing ratios were eliminated. The final dataframe was shaped as (95347, 3016).

**Exploratory Data Analysis (EDA):**

I conducted EDA on the processed data, presenting visualizations on the top ten brands with the highest number of fake products, fake products distribution across states, gender-wise distribution of fake products, and fake products across different categories. Unsupervised learning techniques, particularly K-means clustering with K=2 using 'reviews\_rating' and 'followers’, revealed that products with a 'reviews\_rating' greater than 3.5 stars out of 5 are more likely to be real.

**Tree-based Model:**

Initial Analysis:

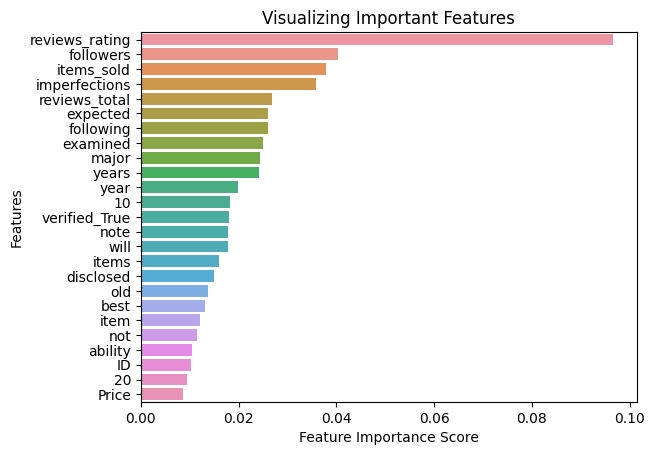
The processed data revealed a no-information rate of 95.2%, indicating that 95.2% of data points were labeled as real products. The objective of the subsequent tree-based classification was to surpass this no-information rate.

Model Training:

A 70-30 train-test split was implemented, and a Random Forest model was trained and fine-tuned using one-hot encoding for categorical features, numerical features, and Bag of Words features.

Model Evaluation:

The model achieved an impressive accuracy of 99.9% on the testing data, surpassing the established no-information rate. Notably, the feature importance plot highlighted 'reviews\_rating' and 'followers' as the top two influential features, aligning with the insights gained from the K-means clustering results.

Figure 1. Feature importance plot

K-Means Cluster-Type Ratio Table

|  | Type 0 | Type 1 | Type 2 |
| --- | --- | --- | --- |
| Cluster 0 | 0.98 | 0.02 | 0.01 |
| Cluster 1 | 1.00 | 0.00 | 0.00 |
| Cluster 2 | 0.02 | 0.79 | 0.18 |

This collaborative effort in data scraping, processing, EDA, and model fitting contributed significantly to the project's success, providing valuable insights into distinguishing real and fake products on Depop.

# **Individual Contribution (Bofei):**

**Data Scraping:**

Performing research on existing E-commerce data scraping methods and scripts, I decided to use the data collection scraper mentioned above. I also collaborated with Kai in developing a username scraping script with specific search keywords. For example, searching the keyword “carhartt jacket” and sorting by “Newly listed” products on Depop allowed the script to obtain all the unique usernames of the sellers of the first 200 listings, which were then used as inputs for the data scraper.

**Annotation:**

I am a part-time vintage reseller on eBay, Grailed, and Instagram. Although I do not sell on Depop, I source products on Depop, and therefore have a lot of experience dealing with both legitimate sellers and scammers. I contributed in annotation by suggesting the following behavior patterns:

1. The fake users tend to have only a few reviews, but it’s more common for them to have zero reviews.
2. The fake users’ listings tend to have all kinds of backgrounds in their listing photos, such as different flooring. While the real users tend to use consistent or a few reappearing photo backgrounds.
3. The fake users’ listings tend to list their items for a price significantly lower than their market value. However, this is trickier in that more domain knowledge is required to make this judgment. Additionally, some scammers will price the item accurately in order to gain more money.
4. Most or all product photos in the fake users’ listings are stock photos directly from the website or even sourced from other real sellers’ photos from eBay, Vinted, Grailed, etc., which we discovered with liberal usage of Google Image search. Because they tend to not actually have the products that they’re selling, the fake users have to source their photos from other places.

Note that this is not a comprehensive list of criteria we used for annotation, but only what I contributed. Please find the full list in the earlier sections.

**Feature Engineering:**

For the tree-based model our group used, Kai and I discussed what features to use in order to optimize the prediction of scam users. My particular contribution includes the following:

1. The omission of the “last\_seen” feature. Due to the way we annotated the data by individual users, the “last\_seen” feature, when merged with the product data, is almost categorical and equivalent to a user\_id feature.
2. For the usage of the description text data, I suggested that we implement a unigram (BOW) feature with one-hot encoding based on the vocabulary of the description. To limit the number of the features considered, we decided to only consider the 1000 most frequent words.

**Sentence Embedding Based NN (SentEmb NN) Model:**

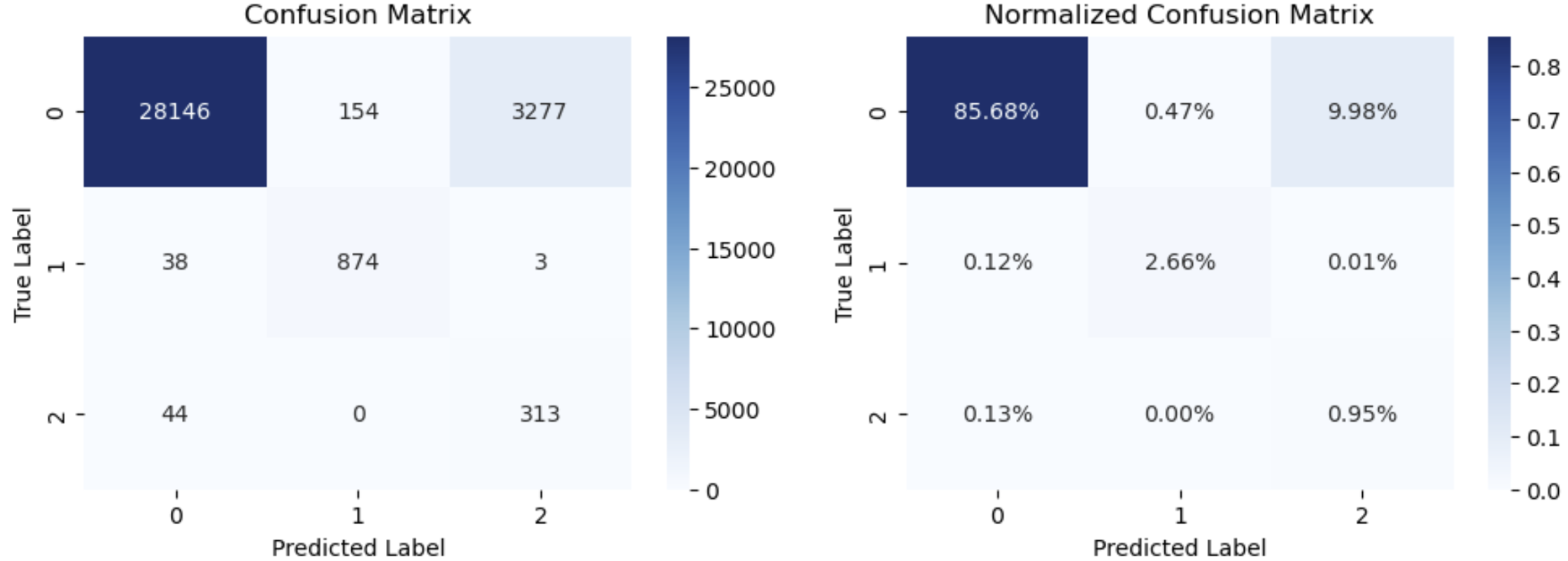
After viewing the feature split frequency, we realized that the tree-based model relied heavily on a seller’s average review and the review count (Please refer to the later sections for more details). Therefore, we realized an extension of this project would be to identify the scammers among all the relatively newer users who could have zero reviews when first starting out. For this, we implemented a NN model which only considers the listing descriptions as the input variable. We featurized the listing descriptions to perform sentence-level embedding, in which we implemented the pretrained all-distilroberta-v1 model, which is based on BERT (Bidirectional Encoder Representations from Transformers). The embedding size of all-distilroberta-v1 is 768.

After embedding the sentences, I implemented a NN with dimensions [768, 512, 512, 512, 256, 256, 3] , with one linear layer, followed by non-linear layers with ReLU as their activation function, to perform the classification task. The configuration was confirmed after experimenting over a few different parameters.

Additionally, in modeling, the following were taken into consideration.

1. The dataset were split into the training set and a validation set by a 0.8/0.2 ratio.
2. The Cross Entropy loss function was considered.
3. Adam was selected as the optimizer.
4. The class imbalance was addressed by implementing class weights in evaluating the loss.
5. An early stopping criterion of “No improvement of validation loss over 20 epochs” was selected.

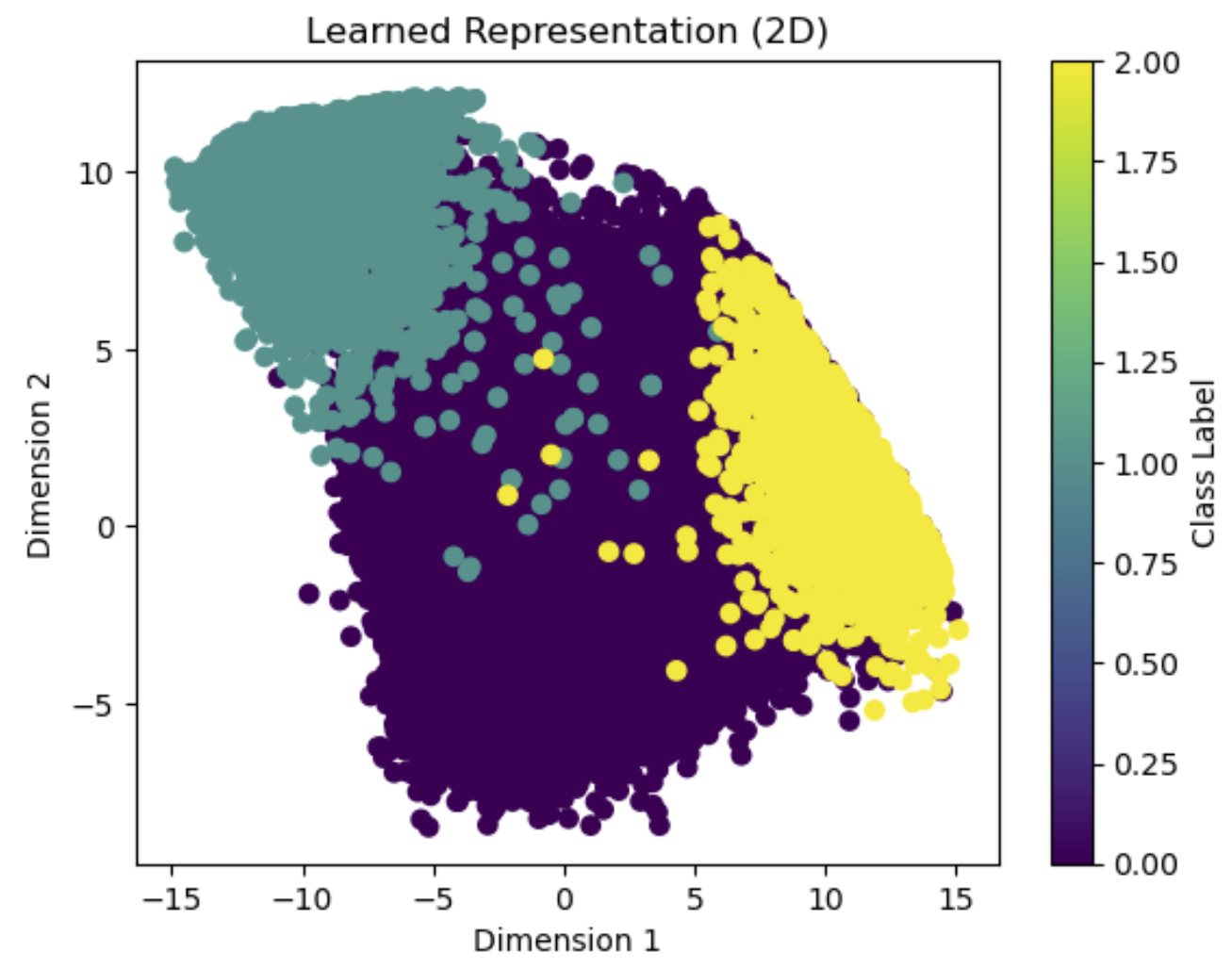
Figure 2. Confusion Matrix and the Normalized Confusion Matrix of the SentEmb NN



The early stopping criterion was triggered at epoch 96, and the model weights obtained at the end of epoch 76 was saved as the final model. The confusion matrix as well as the normalized confusion matrix are presented in Figure 2.

One drawback of NNs is the low interpretability of their self-extracted “features”, to interpret the model’s internal representations, I performed PCA on the hidden vector before the output layer, and plotted the data points with the X-axis as the first principal component, with the Y-axis as the second principal component. The plotting is shown in Figure 3.

Figure 3. Visualization on the First Two Principal Components of the Internal Representation of the SentEmb NN from the Hidden Layer Prior to the Output Layer



This result is very encouraging as the separation between Class 1 (by fake users), and Class 2 (by new users) listings are very clear. Together with the results shown in the confusion matrix, we determined that the SentEmb NN model was able to differentiate between these two target classes very well. While a considerable number of Class 0 users were misclassified as the other two classes by this model, this task could have been easily handled by a tree-based model and were not the focus of the SentEmb NN model. The overall validation accuracy was approximately 90%.

# **CONCLUSION AND LIMITATIONS**

Tree-Based models perform well in classifying different types of products, and the feature importance plot from Tree-Based model is intuitive and reasonable. The result of feature importance analysis matches with the Unsupervised Learning result (K-means). Review ratings play a key role in the Tree-based model, and the K-means clustering shows Cluster 2 contains a high ratio of Type 1 (0.79) products compared with Cluster 1 and 0. This implies the unsupervised learning method uses Cluster 2 to represent the Type 1 product correctly.

The SentEmb NN produced encouraging results in the overall accuracy, confusion matrix results, and the internal representation principal component clusters.

Real-World Implications:

This is particularly meaningful in differentiating the new sellers from the fake sellers, who could both have a 0 review, a few to none followers, and haven’t sold many items.

Some of the limitations include:

1. A larger number of annotators would have captured the variability of human judgment better, and could have potentially changed the categorical labeling into scoring from the level of agreement between annotators.
2. The labeling of products was inherent from the sellers, which does not capture the occasional foul plays by the established sellers. Albeit this is quite rare.
3. The scrapings could be expanded to different key word search results, to capture variability between departments.
4. For the SentEmb model, it would be interesting to train/test it on data exclusively from sellers with a 0 review, a few to none followers, and haven’t sold many items. This was not feasible due to the limitation on our scope.

Works Cited

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. *"RoBERTa: A Robustly Optimized BERT Approach."* arXiv preprint arXiv:1907.11692 (2019).

Kingma, D. P., & Ba, J. (2014), *"Adam: A Method for Stochastic Optimization.",* arXiv preprint arXiv:1412.6980.

Github User [Gertje823](https://github.com/Gertje823/Vinted-Scraper/commits?author=Gertje823), *Vinted-Depop Scraper, https://github.com/Gertje823/Vinted-Scraper/blob/main/README.md*