**CONTENT**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **TOPIC NAME** | **Pg No.** |
| 1 | ABSTRACT | 3 |
| 2 | INTRODUCTION | 4 |
| 3 | BACKGROUND STUDY | 5 |
| 4 | AXIOMS | 6 |
| 5 | BLOCK DIAGRAM | 9 |
| 6 | ALGORITHM | 11 |
| 7 | FORMULATION | 13 |
| 8 | BIRDNEST | 16 |
| 9 | RETWEET US, WE RETWEET YOU | 18 |
| 10 | MORE ABOUT REV2 | 19 |
| 11 | RESULT | 20 |
| 12 | SNAPSHOTS | 22 |
| 13 | SUMMARY | 23 |
| 14 | FUTURE POSSIBILITIES | 23 |
| 15 | REFERENCES | 24 |

**ABSTRACT**

Rating platforms enable the large-scale collection of user opinions about items (e.g., products or other users). However, untrustworthy users give fraudulent ratings for excessive monetary gains. Rating improves customer trust and loyalty toward both products and services.

**Savvy shoppers almost never purchase a product without knowing how it’s going to work for them**. They read the good, the not-so-good, and the downright ugly to make the all-important decision: should I pull out my wallet and take the plunge?

The immediate benefit of reviews is that they can make your future customers feel that much more confident. **The more accurate reviews you have, the more convinced a shopper will be that they are making the right decision.** Reviews can help increase a store’s online presence, too.

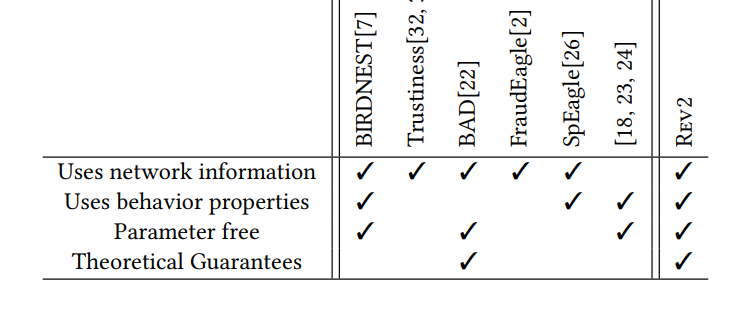
*But what if those reviews are not even right?* Correct, Next time people will think twice before shopping from your website.

**More accurate reviews = more exposure for your store, long-term.**

**INTRODUCTION –**

In this paper, we present REV2, a system to identify such fraudulent users. We propose three interdependent intrinsic quality metrics--- **1)** fairness of a user, **2)** reliability of a rating and **3)** goodness of a product. Fairness and reliability quantify the trustworthiness of a user and rating, respectively, and goodness quantifies the quality behavior duct. Intuitively, a user is fair if it provides reliable scores that are close to the goodness of products. We propose six axioms to establish the interdependency between the scores, and then, formulate a mutually recursive definition that satisfies these axioms. We extend the formulation to address the cold start problem and incorporate behavior properties. We develop the REV2 algorithm to calculate these intrinsic quality scores for all users, ratings, and products. By conducting extensive experiments on five rating datasets, we show that REV2 outperforms some existing algorithms (Some of them are BIRDNEST, FRAUDEAGLE, SPEAGLE, and BAD) in detecting fair and unfair users.

We ran this algorithm on our dataset and found the result 80-83% accurate.



**BACKGROUND STUDY –**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Paper id** | **Objective** | **Feature** | **Dataset** | **Technique** | **Performance** | **Result** |
| **1.** **BIRDNEST: Bayesian Inference for Ratings-Fraud Detection** | **To find the suspicious users on a platform** | **It takes the timestamp and the rating pattern of the user ratings.** | **FLIPKART**  **SWM** | **It uses Bayesian model (BIRDNEST ALGORITHM)** | **Precision of 1.0 for the first 50 users on Flipkart.** | **Gives fraudulent scores to the user and ranks them on the basis of this score.** |
| **2.** **CoReRank: Ranking to Detect Users Involved in Blackmarket-Based Collusive Retweeting Activities** | **To detect collusive users and suspicious tweets simultaneously with theoretical guarantees** | **It uses credibility of the user and the merit of the tweet by calculating the seed score and topical diversity** | **manually curated and annotated dataset** | **Uses the CORE-RANK algorithm** | **CoReRank achieves 0.85 average precision and 0.60 average recall** | **Ranks the user based on their collusive activities and tweets based on their merit score.** |
| **3.**  **Rev2:**  **Using Graph**  **Analysis** | **To find the collusive users who give a fake rating from the real ones.** | **It takes the use of behavioural aspects as well to give the most accurate result possible.** | **Amazon\_network.csv**  **Amazon\_gt.csv** | **It uses the algorithm named Rev2.** | **It manages to achieve a total accuracy of about 80 AUC scores.** | **Collusive users are sorted out and it gives a result that how many users are collusive.** |

**REV2 Properties** -

We consider directed bipartite rating networks of user-to-product, where each rating is from a user u to a product p. We propose that users and ratings have (unknown) intrinsic scores that quantify how trustworthy they are, and products have (unknown) intrinsic scores that quantify their quality or how a layman is likely to evaluate them. Naturally, these scores are interdependent and unknown apriori. In this section, we describe the axioms that establish the inter-dependency between these scores, and propose an algorithm that satisfies the axioms and calculates these scores.

**ORIGINAL INFORMATION NEEDED TO UNDERSTAND FORMULATIONS**

A bipartite rating graph G = (U, R, P) is a directed, weighted graph, where user u ∈ U gives a rating (u, p) ∈ R to product p ∈ P. Let the rating score be represented as score (u, p). Let U, R and P represent the set of all users, ratings and products, respectively, in a given bipartite network. We assume that all rating scores are scaled to be between -1 and +1, i.e. score (u, p) ∈ [−1, 1] ∀ (u, p) ∈ R. Let, Out (u) be the set of ratings given by user u and In(p) be the set of ratings received by product p. So, |Out(u)| and |In(p)| represents their respective counts.

**Intrinsic Properties: Fairness, Goodness, and Reliability**

We model the bipartite network such that users, ratings and products have (unknown) intrinsic quality scores. User and rating intrinsic scores indicate how trustworthy they are, and product intrinsic scores indicates how likable it is. Their meanings and purpose are explained below:

* Products vary in quality, measured by a metric called goodness. A **goodness score** is a single number indicating the most likely rating a fair user would give it. Intuitively, a good product would get several high positive ratings from fair users, and a bad product would receive high negative ratings from fair users. The need for incorporating the rating user’s fairness arises because simple measures like a product’s average or median scores can easily be manipulated by giving ratings using multiple fake accounts. The goodness score G(p) of a product p ranges from −1 (a very low-quality product) to +1 (a very high-quality product).
* Users vary in terms of their **fairness** which indicates how trustworthy it is. Fair users rate products without bias, i.e., they give high scores to high-quality products and low scores to bad products. On the other hand, users who frequently deviate from the above behavior are ‘unfair’, e.g., fraudulent users that give high ratings to low-quality products and low ratings to good products. Likewise, a ‘strict’ (‘liberal’, resp.) user who consistently gives negative (positive, resp.) ratings to high (low, resp.) goodness products is intuitively less fair than a user who gives negative (positive, resp.) ratings to low (high, resp.) goodness products. Fairness score F (u) of a user u lies in the [0, 1] interval ∀u ∈ U. 0 denotes a 100% untrustworthy user, while 1 denotes a 100% trustworthy user.
* Finally, ratings vary in terms of **reliability**, which reflects how trustworthy it is. The reliability score R (u, p) of a rating (u, p) ranges from 0 (untrustworthy) to 1 (trustworthy).

**Is a rating’s reliability identical to its user’s fairness?**

We need separate intrinsic scores for users and ratings because a user may give ratings with different reliabilities due to their biases and perceptions.

**AXIOMS THAT DEFINE INTERDEPENDENCIES BETWEEN THE THREE**

* **Axiom 1** (Better products get higher ratings). If two products have identically reliable ego networks and for one product, all the rating scores are higher, then quality of that product is more.
* **Axiom 2** (Better products get more reliable positive ratings). If two products have identical rating ego networks and for the first product, all positive ratings are more reliable and all negative ratings are less reliable than for the second product, then the first product has higher quality.
* **Axiom 3** (Reliable ratings are closer to goodness scores). For two ratings by equally fair users, the rating with score closer to the product’s goodness has higher reliability.
* **Axiom 4** (Reliable ratings are given by fairer users). For two equal ratings to equal goodness products, the one given by fairer user has higher reliability.

This axiom incorporates the user’s reputation in measuring reliability. This way same ratings received by a product may have different reliability scores.

* **Axiom 5** (Fairer users give more reliable ratings). For two users with equal number of ratings, if one has higher reliability for all its ratings than the other, then it has higher fairness.

**ALGORITHM (BLOCK DIAGRAM)**

****

**ALGORITHM THEORY EXPLANATION –**

We collected existing datasets on Kaggle namely Amazon\_network.csv and Amazon\_gt.csv for working on our project. We opted to work on these two datasets as they were of the same type and hence merging those into one had no issues involved. We also took use of the bird-nest algo dataset as it was also readily available and improves the accuracy of the result further.

We performed around 500 iterations to find out the value of fairness (of User) (F), goodness (of Product) (G), and reliability (of Rating) (R), and finally after all the iterations, we consider that consecutive R, G, F whose difference is minimum. We used these as parameters to get the final Fairness of the user.

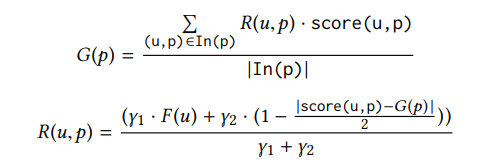
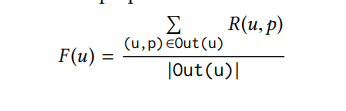
This data is further used to calculate the Average Precision Scores which shows how many real users are there compared to false users in the dataset.

Then we send the result obtained by REV2 to calculate the AUC score with the help of RANDOM-FOREST algo.

**FORMULATION** –

The Formula is derived in various Phases

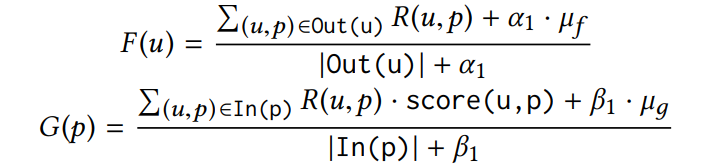
* **In First Phase**, we already derived all the relations in the Axioms above and do not we need to derive Formulas related to them which are –



* Fairness can be said as the sum of all ratings by a particular user divided by the number of ratings given by the user.
* Goodness is defined as the sum of all ratings given by users who rated the product \* scores divided by the total number of ratings given to a particular product.
* Reliability of ratings can be defined as the Fairness of users who are given a rating + (1 – (score given – goodness / 2)). Y1 and Y2 are the constants to apply for smoothing.
* *WHAT IS THE COLD START PROBLEM?*

In rating networks, most users give and most products receive only a few ratings. For such users and products, there is little information to measure their true quality. For example, a user with few but highly accurate ratings may be an honest user or a fraudster who is camouflaging himself. Conversely, a user with few and highly inaccurate ratings may be a benign novice or a throwaway fraud account. Due to a lack of sufficient information, little can be said about their true fairness. The same happens with products that receive a few ratings. This uncertainty due to insufficient information on less active users and products is the cold start problem.

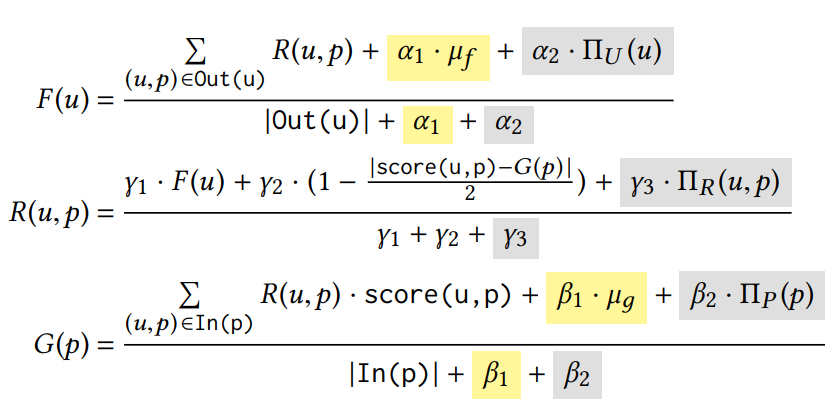
* **In the Second Phase** of Formulation, We have the problem of cold start, and hence to improve it we will take help from Laplace Smoothing.



Smoothing parameters α1 and β1 are non-negative integer pseudo counts, and µf and µд are prior beliefs for fairness and goodness scores of new nodes, respectively. The prior is the default score each user and product have if it didn’t give or get any ratings.

This works because if a user gives only a few ratings, then its fairness score is close to the default score µf. µf and µд are set as the mean scores of all users’ fairness and all products’ goodness scores, respectively.

* **In the Third Phase**, the behavior and characteristics of users, ratings, and products are very informative of their qualities as well.



Where PI(U) is the Behaviour Value which is given with the help of bird-nest,

In our Calculations, we took the help of an already available bird-nest data file instead of calculating its value on our own.

**BIRDNEST** **(Bayesian Inference for Rating Data (BIRD) Normalized Expected Surprise Total (NEST))**

Review fraud is a pervasive problem in online commerce, in which fraudulent sellers write or purchase fake reviews to manipulate perception of their products and services. Fake reviews are often detected based on several signs, including 1) they occur in short bursts of time 2) fraudulent user accounts have skewed rating distributions. However, these may both be true in any given dataset. Hence, in this paper, is proposed an approach for detecting fraudulent reviews which combine these 2 approaches in a principled manner, allowing successful detection even when one of these signs is not present. To combine these 2 approaches, they formulate a Bayesian Inference for Rating Data (BIRD) model, a flexible Bayesian model of user rating behavior. Based on our model we formulate a likelihood-based suspiciousness metric, Normalized Expected Surprise Total (NEST).  A linear-time algorithm for performing Bayesian inference using the BIRDNEST model and computing the metric. Experiments on real data show that BIRDNEST successfully spots review fraud in large, real-world graphs: the 50 most suspicious users of the Flipkart platform flagged by the BIRDNEST algorithm were investigated and all were identified as fraudulent by domain experts at Flipkart.

Intuitively, deciding how suspicious each user involves a two-step process: first, involves estimation of our beliefs for what that user’s true rating distribution is. Second, is an estimation of how suspicious we believe they are, given our beliefs.

**Bayesian Model**

The Bayesian approach applies this intuition in a principled manner. It first sets a prior, estimated from data, representing our ‘default’ beliefs about users’ rating behavior. It then estimates our beliefs (in the form of a posterior distribution) about their rating distribution. Finally, it is computed how suspicious we believe a user to be, averaging over their posterior distribution.

**NEST: Proposed Metric for Detecting Suspicious Users**

Suspiciousness is computed with respect to the rating and temporally, then normalized and combined to ensure that each has equal influence. This is a practically motivated decision that ensures that even in settings where one of the variables has a much finer resolution than the other (i.e. it is bucketed into more buckets), neither variable will dominate the other in determining suspiciousness.

DEFINITION 1. (EXPECTED SURPRISE) The expected surprise for user i measures how surprising the user i rating distribution is averaged over its posterior distribution.

DEFINITION 2. (NEST) NEST measures how jointly suspicious a user is based on his or her ratings and temporally.

Evaluation of Flipkart data Flipkart is an online e-commerce platform on which merchants sell products to customers, on which customers review products from 1 to 5 stars. BIRDNEST was applied to detect the 250 most suspicious users and provided them to Flipkart; these accounts were investigated and hand-labeled by Flipkart, finding that 211 users of the top 250 flagged by BIRDNEST were involved in fraud.

DATASET  USED

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | # of users | # of products | # of ratings |
| Flipkart | 1.1M | 550K | 3.3M |

**Retweet Us, We Will Retweet You: Spotting Collusive Retweeters Involved in Black-market Services**

Twitter has increasingly become a popular platform to share news and user opinion. A tweet is considered to be important if it receives a high number of affirmative reactions from other Twitter users via Retweets. Retweet count is thus considered a surrogate measure for positive crowd-sourced reactions – high a number of retweets of a tweet not only helps the tweet being broadcasted but also aids in making its topic trending. This in turn bolsters the social reputation of the author of the tweet. Since the social reputation/impact of users/tweets influences many decisions (such as promoting brands, advertisement, etc.), several black market syndicates have actively been engaged in producing fake retweets in a collusive manner. Users who want to boost the impact of their tweets approach the black market services and gain retweets for their own tweets by retweeting other customers’ tweets. Thus, they become customers of black market syndicates and engage in fake activities. Interestingly, these customers are neither bots, nor even fake users – they are usually normal human beings; they express a mix of organic and inorganic retweeting activities, and there is no synchronicity across their behaviors.

Challenges in detecting collusive retweeters:

(i) They are not bots, but human beings; therefore, bot detection algorithms cannot flag them.

(ii) Their Twitter accounts are not fake; therefore, fake account detection algorithms cannot detect them.

(iii) They express a mix of organic and inorganic behavior in their retweeting patterns – they organically retweet some genuine tweets; at the same time, they inorganically retweet tweets submitted to black market services. The extent of inorganic activities may differ across retweeters.

 (iv) They do not show any synchronicity across their retweeting patterns, thus making it difficult for the existing synchronous fake retweeter detection methods to detect them.

**MORE INTO REV2 -**

The number of iterations needed to reach convergence is at most 2+ ⌈ log (ϵ /2) log (3/4) ⌉. In other words, treating ϵ (acceptable error) as constant, the number of iterations needed to reach convergence is bounded by a constant. Thus, the lower the ϵ value, the higher the number of iterations till convergence.

ϵ -> It is the value that we need to end our iterations, If the difference of values reaches less than the particular value

of ϵ (which varies according to the need) -> Then we break our iterations, and the current value of goodness, Fairness and reliability will be the final values.

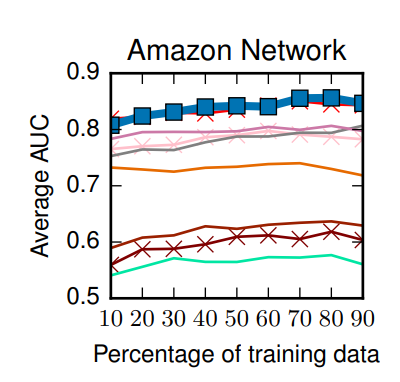
This algorithm is compared to other algorithms like SpEagle, FraudEagle, etc. and Average Precision Score is calculated for all. Result of Rev2 were on higher end when compared to its similar algorithm.

Average AUC on test sets, averaged over 50 random iterations of training data. Rev2 is robust to the amount of training data. Its performance is relatively stable (AUC ≥ 0.80 in almost all cases) as the amount of training data varies.

10-fold cross validation is used int REV2 algorithm while training and testing.

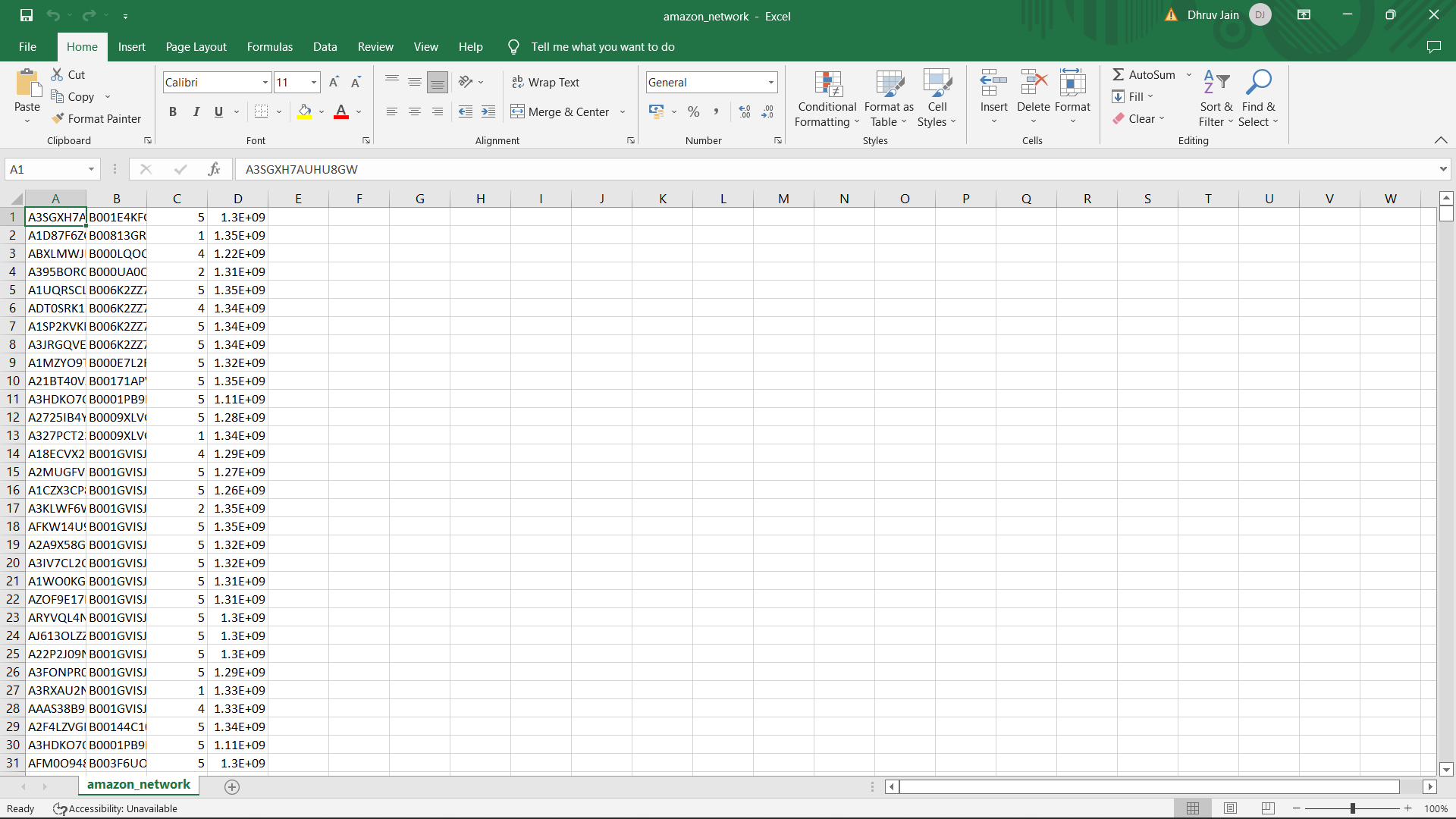




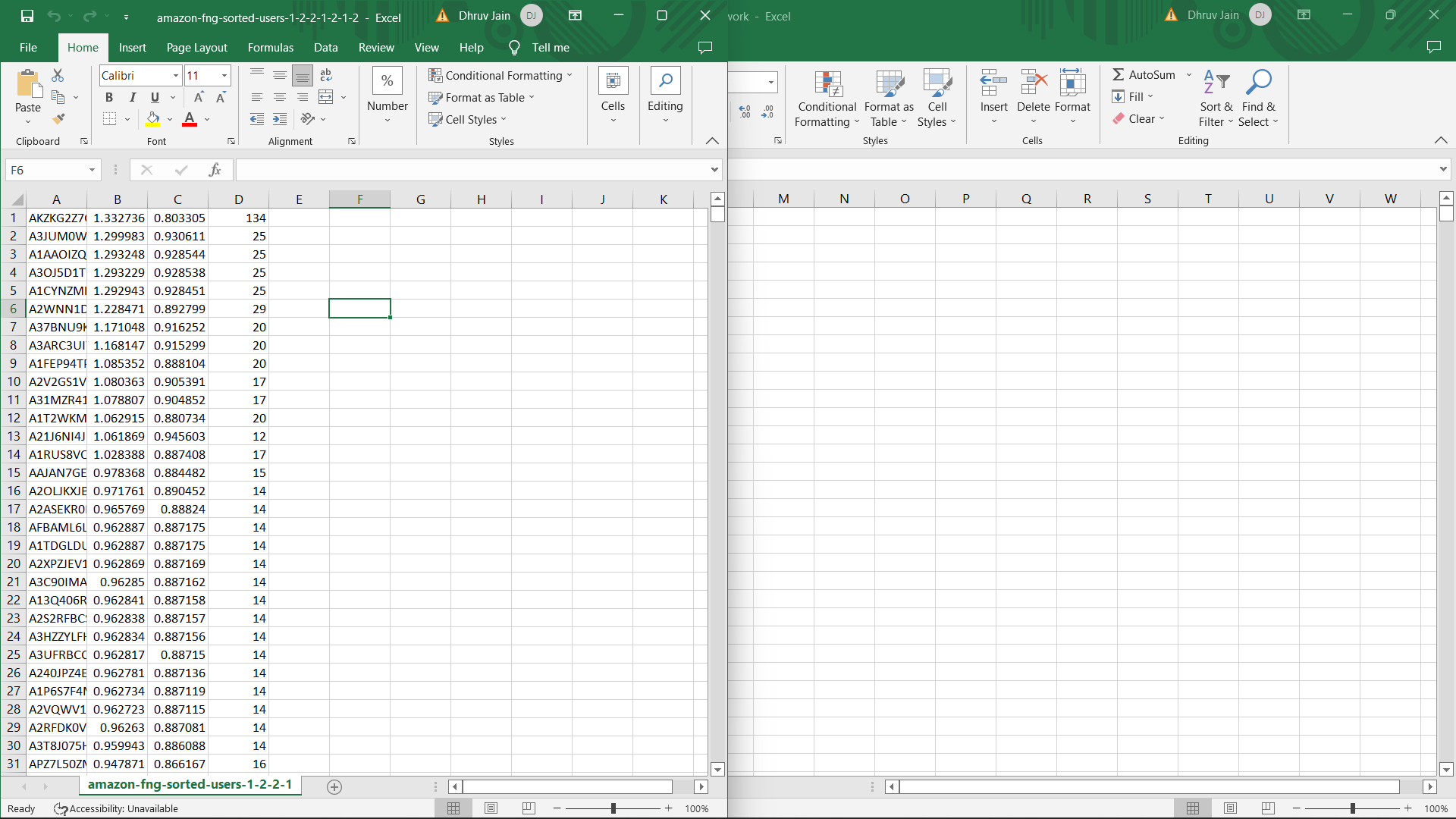


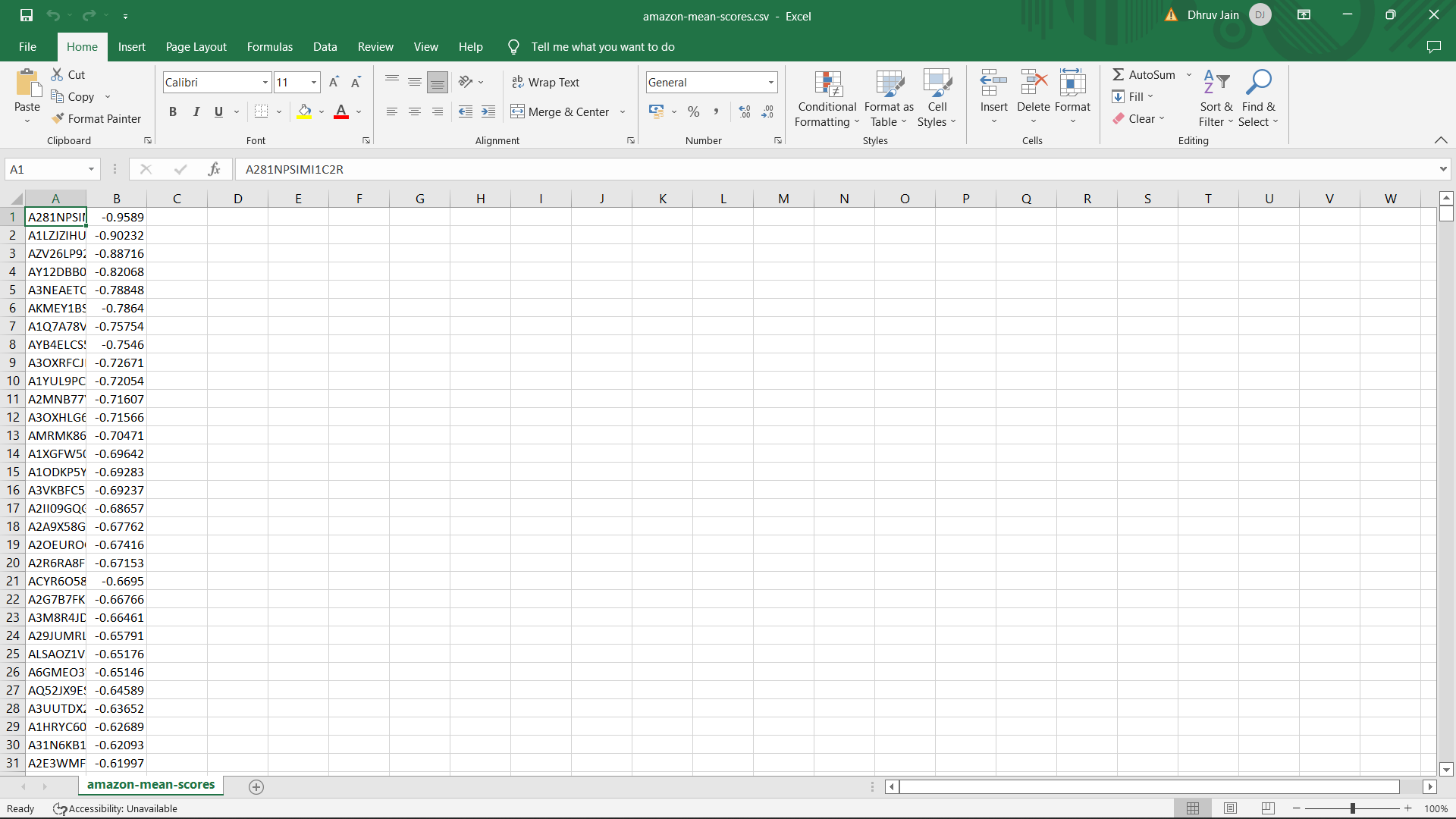
**SNAPSHOTS-**

INPUT FILE FOR REV2CODE.PY

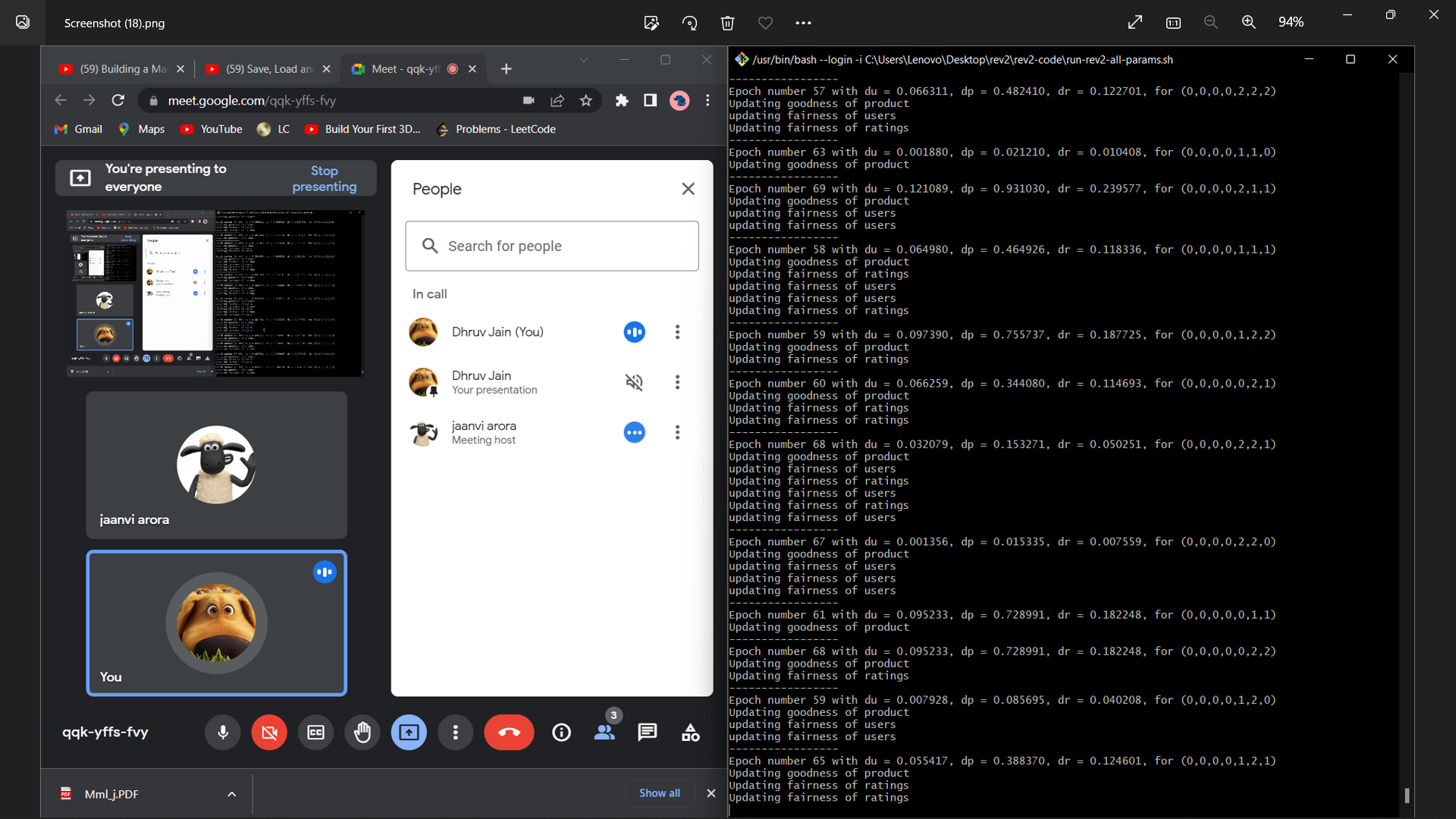


OUTPUT BY REV2CODE.PY



OUPUT BY EC\_2.PY

SHELL SCRIPT FOR AUTOMATING ALL SEVEN VALUES



**SUMMARY**-

This whole Project was a try to reduce the number of fake ratings which are all around the internet today. The amount of these rating is just reaching insane numbers, especially after the covid.

Fraudsters are trying at their level best to create these false ratings and hence companies also need to be aware of this more and hence we tried to help in this whole scenario by creating (modifying) this system of Rev2 just so that this

The whole process come in limelight and as the final goal reduce its effect.

Rev2 judge products based on parameters such as Goodness, fairness, and reliability and finally conclude whether the user giving rating is collusive or not and we are able to get an accuracy of about 80 AUC.

Reviews

**FUTURE POSSIBILITY** –

This Rev2 algorithm has some more scope for future upgradation like –

* We can Implement a Timestamp in a future approach, Logic is that if the user is collusive then he will give reviews in a very small time gap and we can take advantage of this to find which users are collusive.
* We can introduce the feature of behaviour aspects as well as a background check before assigning the initial value of fairness to the user. It tends to give better results theoretically as we have more information about the user available.
* We also want to introduce a system of Reviews also with ratings. During our research, we found the dataset of Twitter which is based on ratings and we intend to combine both these datasets so that we can have one another helpful parameter to define the reviews better.
* Another interesting factor can be to introduce and find the very best values of constants alpha, beta, and gamma so that the accuracy can be best. This can be done by writing a shell command which can check values at various different parameters which can not be humanly possible.

**References** –

* <https://github.com/Shlok2/Minor_1> (Research Papers)
* https://www.researchgate.net/publication/284219133\_BIRDNEST\_Bayesian\_Inference\_for\_Ratings-Fraud\_Detection
* <https://ieeexplore.ieee.org/document/8508801>

(Retweet us, and we will Retweet You)

* https://www.kaggle.com/code/vvineeth/twitter-spam-detection/notebook

(Twitter Scam Dataset)