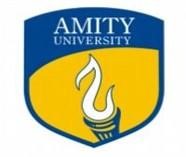
**A Project Plan on**

**Recommendation systems**

**Submitted to**

**Amity University Uttar Pradesh**



**In partial fulfillment of the requirements for the award of the degree**

**Of**

**Bachelor of Technology in**

**Computer Science and Engineering** **By**

**ARYAN RAWAT**

**A2305219448**

**Under the guidance of**

**Faculty Guide: Dr. Sunil Kumar**

**Industry Guide: Mr. Marghoob Suleman**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY** **AMITY UNIVERSITY UTTAR PRADESH**

**DECLARATION**

I, Aryan Rawat student of B. Tech (CSE) hereby declare that the project titled “Recommendation Systems” which is submitted by me to Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, in partial fulfillment of requirement for the award of the degree of Bachelor of Technology in Computer Science , has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

**Date: -** 25.7.2022

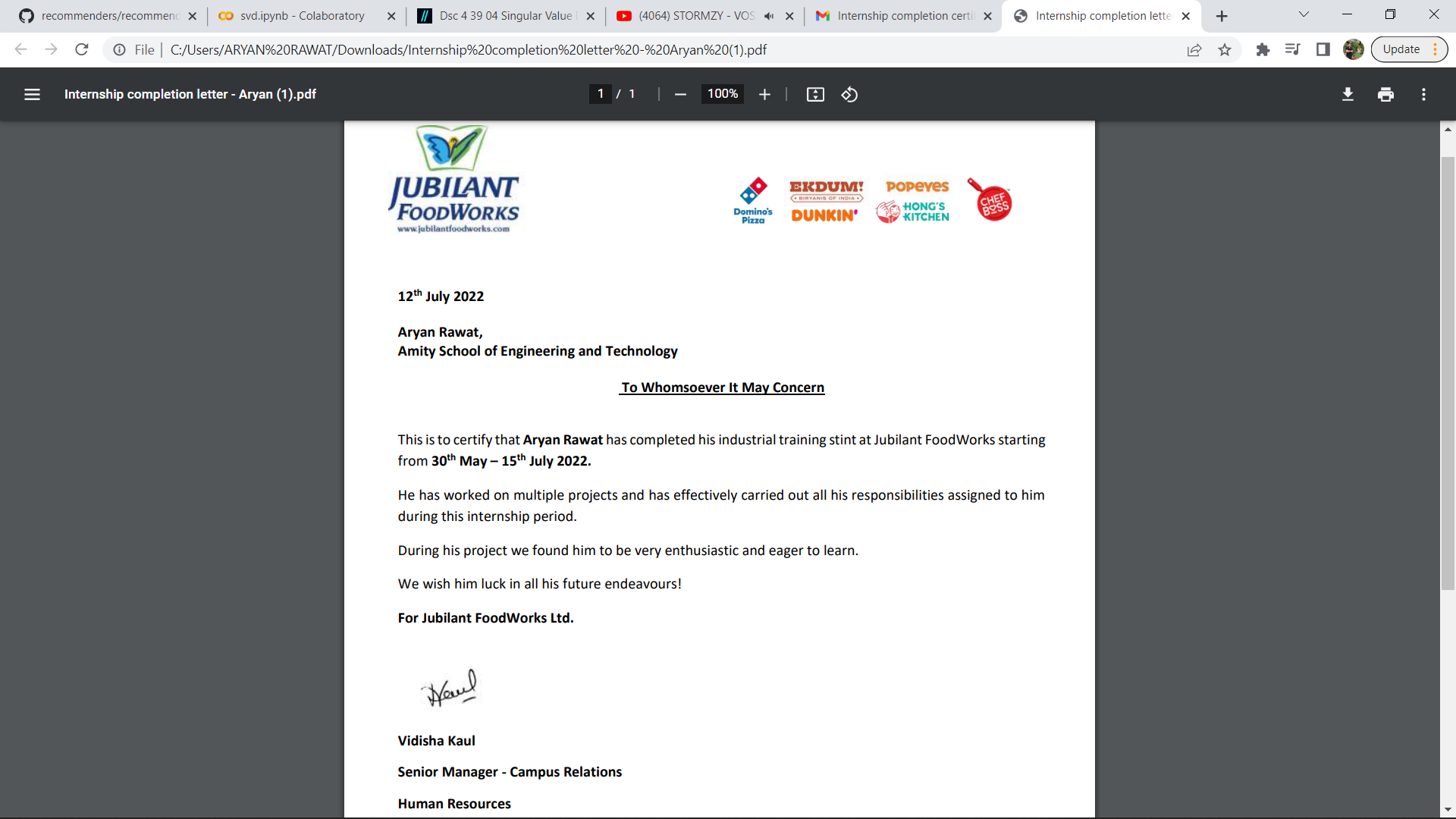
**Name: -** Aryan Rawat

**Roll No: -** A2305219448

Name and signature of Student(s)

Aryan Rawat

**CERTIFICATE**



**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. I would like to thank Prof (Dr.) Sunil Kumar Name, Head of Department-CSE, and Amity University for allowing me to undertake this project. I would like to thank my faculty guide Mr. Sunil who is the biggest driving force behind my successful completion of the project. She has always been there to solve any query of mine and also guided me in the right direction regarding the project. Without her help and inspiration, I would not have been able to complete the project. Also, I would like to thank my batch mates who guided me, helped me, and gave ideas and motivation at each step.

**Aryan Rawat** **A2305219448**

7CSE-7X

Contents

[1. INTRODUCTION 7](#_Toc109649669)

[2. IDEA OF IMPLEMENTATION 7](#_Toc109649670)

[3. RECOMMENDATION SYSTEMS 7](#_Toc109649671)

[4. Approaches of recommendation systems 9](#_Toc109649672)

[4.1. The content based approach 9](#_Toc109649673)

[5. ALGORITHMS USED FOR RECOMMENDERS 10](#_Toc109649674)

[5.1. Singular Value Decomposition SVD 10](#_Toc109649675)

[6. IMPLEMENTATION 11](#_Toc109649676)

[7. CONCLUSION 19](#_Toc109649677)

[8. REFERENCES 19](#_Toc109649678)

**ABSTRACT**

Recommendation system (RSs) are software that gives recommendation from a large set of data and tend to give answer for various problems on diverse platforms. Use of recommendation system is far more common than we can predict. For instance Food proposal frameworks (RSs) are programming frameworks that make customized suggestions from an enormous scope of various choices and in this manner give a promising answer for data over-burden and undesirable food choices. An overview of the types and approaches for food RSs is given and a recommender for pizza suggestion is made. Approach for pizza recommender can be of many types here user based data is used to calculate similarity between each pizza and suggest according to it. It can be concluded that in this age of data it is easy to capture large amounts of data and deduce patterns from it which can provide guesses about your likes and dislikes.

# INTRODUCTION

In the present society most people utilize the web to look for data, to speak with their informal community or to purchase labor and products. Close to this, We can also see how much the web services are expanding and is at present around 4.5 billion, of which 2.1 billion individuals purchase labor and products on the web (data by- internet world stats 2019). In other words, internet business is developing as it gives advantages to organizations and purchasers. Some benefits for online purchasers are that they can approach it at any spot and whenever; can buy similar labor and products from a few organizations; and can save time, cash, and exertion. It can likewise prompt data over-burden, meaning a wealth of data that makes it hard to isolate helpful from pointless data.[1]

This pointless data is a big issue for both consumers and sellers as both can’t find desired goals, information overload can be a very major problem as the consumer can decide to make sub optimum solution and face dissatisfaction. To solve these problems online websites and organizations made recommendation system. Recommendation system benefit purchasers as they diminish shoppers' mental exertion and search costs by making it quicker and simpler to analyze options and gather data. This can further develop the purchaser's choice quality and trust in their choice Next to this, RSs benefit organizations, subsequently, it isn't peculiar that an ever increasing number of areas use RSs, for example, schooling, medical care, internet business, and food. With the consistent improvement of electronic advances, RSs have become more exact in foreseeing clients' inclinations. Thusly, these improvements brought about RSs that can suggest choices and integral things, and RSs that can give proposals in any event, when online purchasers are curious about the products.[1][2]

As of late, RSs are likewise being created for the internet based food area as additional customers purchase food or look for food related content on the web. With 'food' additionally food-related things are implied, like dinner arranging, recipes, fixings, bistros, eateries, café menus, and shopping for food. With this comes the problem of selecting pizza from the long menu when you have not tried anything and can spend little time in selection process.

# IDEA OF IMPLEMENTATION

Implement a recommender for pizza based on ratings given by users to other pizzas, user behaviour includes ratings on items. In this manner, the model finds an association between the users and the items. The model is then used to predict the item or a rating for the item in which the user may be interested.

# RECOMMENDATION SYSTEMS

Research on RSs started during the 1990s and acquired consideration after the development of internet business. Different terms allude to RSs, for example, intuitive choice guide framework, "recommender frameworks, proposal specialists, shopping specialists, shopping bots, and correlation shopping specialists.[1]

**3.1. In the primary stage the principal point is to grasp the shopper.**

1. Consumer information collection- The recommendation system directly or indirectly collects user’s information. A direct method to collect user’s information is using set of questions for collecting information. An indirect or implicit method could be collection of data based on clicks, searches and cookies. Be that as it may, it is ideal to utilize the two strategies to expand the precision considerably more .A significant viewpoint to think about is the information on the shopper. In the event that the buyer has little information about the item, administration, or merchant the RS should evoke customer's necessities and ought to show less proposals. Notwithstanding, on the off chance that the buyer is more acquainted with the item, administration, or seller the RS ought to evoke the shopper's inclinations of item credits, like value, brand, and notoriety.
2. Building customer profile: a purchaser profile can be fabricated in view of the chose buyer data. Online organizations don't utilize all data since this isn't ideal and practical.
   1. **In the second stage the principal point is to recognize and conveying proposals.**
3. Matchmaking draws near: various matchmaking approaches can be utilized to recognize the items or administrations that matches the shopper's profile.
4. RS show: the subsequent stage is the RS show or connection point plan of the suggestions. In the event that the plan isn't sufficient, shoppers probably won't grasp the proposals or overlook them. Things that should be thought about are for instance, the quantity of proposals displayed on one page (set size), how much the suggestion matches the buyer's profile (arranging sign), proposal guidance offices, and the decision about whether to utilize an enlivened persona (symbol). Stage 1 and stage 2 decide the exactness of the customized proposals.
   1. **In the third stage** the effect of the RS on the purchasers, organizations, and market is estimated. The proposals are changed in view of the criticism from the client.

impact of

recommendation

system

Personalized

Recommendation

Fig. 1. Stages of recommendation

## Approaches of recommendation systems

## The content based approach

This technique is used to predicts the clients' inclinations by taking consideration of their evaluations of item or administration ascribes and buy history. The items or administrations that are like the high-appraised things and buy history are suggested.[8]

Content based strategies :-

They are based with respect to featuralization of things (as go against to clients) and a profile of a client's utility. It is the most appropriate to solve issue with known information on things and how the client generally cooperate with the recommender framework, yet absence of the client's very own data. Content-based recommenders is basically a client explicit learning issue to measure the client's utility (different preferences, rating, and so forth) in light of thing highlights.[8]

1. The collaborative filtering technique is based upon the fact that likeminded people exist and often consumers tend to buy products in same pattern. A client needs to rate certainly or expressly a few things and in view of this; the RS suggests things that similar individuals too.
   1. **Collaborative filtering**

It is the most appropriate to issue with known information on clients (age, orientation, occupation, and so on), however absence of information for things or hard to do highlight extraction for things of interest. In contrast to content-based approach, cooperative recommender frameworks attempt to anticipate a client's utility for a thing in light of other clients' past utility with the thing.

Prescribing the new things to clients in view of the interest and inclination of other comparative clients is essentially collaborative filtering. For eg:- When we shop on Amazon it suggests new items saying "Client who brought this likewise brought" as displayed underneath.[9][10]

There are 2 types of collaborative filtering:-

1. **User-Based Collaborative Filtering**

Rating of the thing is finished utilizing the rating of adjoining clients. In straightforward words, It depends on the idea of clients' closeness

1. **Item-Based Collaborative Filtering**

The rating of the thing is anticipated utilizing the client's own rating on adjoining things. In straightforward words, it depends on the idea of thing closeness.[7]

* 1. **The knowledge based approach**

It utilizes the data from the client and things, to match the things to the client's rules. This approach expressly evokes data from the client to construct a client profile. From one viewpoint the time has come consuming as the need might arise to connect with the RS. Then again, it gives a clarification to the client of why the thing has been suggested. One illustration of an information based approach is limitation based, that attempts to fulfill all requirements (e.g., client inclinations) given to them. Another model is case-based, that keeps a memory of their effective past suggestions and can change them or base their new proposals on them.[7]

* 1. **The setting based approach**

It use shopper qualities and thing credits, yet additionally context oriented data of the client, for example, why somebody is purchasing the product and for whom they are getting it.

* 1. **The hybrid approach**

It consolidates the methods of the other RS ways to deal with produce proposals, by doing this, they can utilize the benefits of every RS approach and can keep away from their detriments.

Carry out content based and cooperative techniques independently and consolidate their expectations. This is basically a model group approach. Integrate content-based qualities into a cooperative strategy. One method for doing this is to use client profile to gauge closeness between two clients, and utilize this likeness as weight during the accumulation step of cooperative methodology. "For example, cooperative sifting strategies need to confront the new-thing issue. Though happy based approaches can handle this issue in light of the fact that the expectation for new things is generally founded on accessible portrayals of these things"[8]

# ALGORITHMS USED FOR RECOMMENDERS

# Singular Value Decomposition SVD

Singular Value Decomposition (SVD), a strategy from linear algebra based math is getting well known in the field of data science, machine learning and AI. This prevalence is a direct result of its application in creating recommender frameworks. There are a ton of online client driven applications like video players, music players, web based business applications, and so on, where clients are prescribed with additional things to draw in with.

Finding and suggesting numerous reasonable things that would be enjoyed and chosen by clients is dependably a test. There are numerous strategies utilized for this undertaking and SVD is one of those procedures. The Singular Value Decomposition (SVD), a strategy from linear algebra math that has been for the most part utilized as a dimensionality reduction procedure in AI. SVD is a matrix factorisation procedure, which diminishes the quantity of highlights of a dataset by decreasing the space aspect from N-aspect to K-aspect (where K<N). With regards to the recommender framework, the SVD is utilized as a cooperative separating procedure. It utilizes a lattice structure where each line addresses a client, and every section addresses a thing. The components of this lattice are the appraisals that are given to things by clients.

The factorisation of this matrix is finished by singular value decomposition. It tracks down variables of networks from the factorisation of an high level (client-item rating) lattice. The singular value decomposition is a technique for disintegrating a matrix into three different matrix as given beneath: Where A will be a m x n utility matrix, U is a m x r orthogonal left singular grid, which addresses the connection among clients and inactive elements, S is a r x r diagonal matrix, which depicts the strength of each inert component and V is a r x n inclining right solitary grid, which demonstrates the closeness among items and idle elements. The inactive elements here are the attributes of the things, for instance, the class of the music. The SVD diminishes the element of the utility network A by separating its latent features. It maps every client and everything into a r-dimensional latent space. This planning works with an unmistakable portrayal of connections among clients and things.[6]

* 1. **Recommenders using deep learning (Wide and deep)**

Over the most recent 10 years, neural networks have taken a gigantic jump in development. Today they are applied in a great many applications and are bit by bit supplanting conventional ML strategies. For example Youtube’s recommendation algorithm for video recommendation Without a doubt, it's an exceptionally provoking errand to make suggestions for such a help as a result of the large scope, dynamic corpus, and different inconspicuous outside factors.

As per the review "Profound Neural Networks for YouTube Recommendations", the YouTube proposal framework calculation comprises of two neural network: one for applicant generation and one for positioning.

one such popular algorithm is wide and deep in Microsoft recommender repository. A linear model with a wide arrangement of crossed-column (co-ocuurence) features can remember the element collaborations, while deep neural networks (DNN) can sum up the component designs through low-dimensional dense embeddings learned for thesparse features. Wide-and-profound deep learning jointly trains wide linear model and deep neural networks to join the advantages of retention and speculation for recommender frameworks.

* 1. Gradient Boosting Machine (LightGBM)

LightGBM is a gradient boosting mechanism that take help of tree based learning algorithms. It is designed efficient and have these advantages:

* Quicker preparing speed and higher productivity.
* Lower memory use.
* Better precision.
* Backing of equal, conveyed, and GPU learning.
* Fit for dealing with enormous scope information.

1. **PROBLEMS FACED BY RECOMMENDERS**
   1. **Impact of external factors**

Each stage is affected by outer elements, such as individual qualities, social impacts, situational and financial variables, and the on the web. Individual qualities are, for instance the inspiration and contribution, information, values, character type, schooling level, way of life, and socioeconomics of the individual. Close to this, shoppers are socially affected by their own sub-culture , social class, reference bunch, family, companions, media, and the (on the web) exhortation of specialists and different buyers while making a choice . For example, individuals can peruse the assessment and experience of others in web-based surveys, sites, or virtual entertainment posts. This can influence the standing of the dealer and item. The justification behind this, is that these sources are viewed as more solid and dependable by purchasers, since the maker or vender of the item have zero control over them. This social impact not just assists shoppers with deciding for a specific item or administration, however it can likewise inspire unnoticed requirements or needs.

* 1. **Cold start problem**

Depending on client information has its disadvantages, one of which is the issue of 'cold start'. This is the point at which another client enters the framework or new things are added to the inventory, and thusly, it will be challenging for the calculation to anticipate the taste or inclinations of the new client, or the rating of the new things, prompting less exact proposals.

Nonetheless, a profound learning model can streamline the relationships between the client and the item by dissecting the setting of item and client subtleties like item depictions, pictures, and client ways of behaving. It can then make significant suggestions for every individual item or client under various situations. This outcomes in an exceptional arrangement of suggestions that thinks about this large number of factors. Since these profound learning models don't vigorously depend on client conduct information, they are the answer for the virus start issue.[4][5]

* 1. **Privacy concerns**

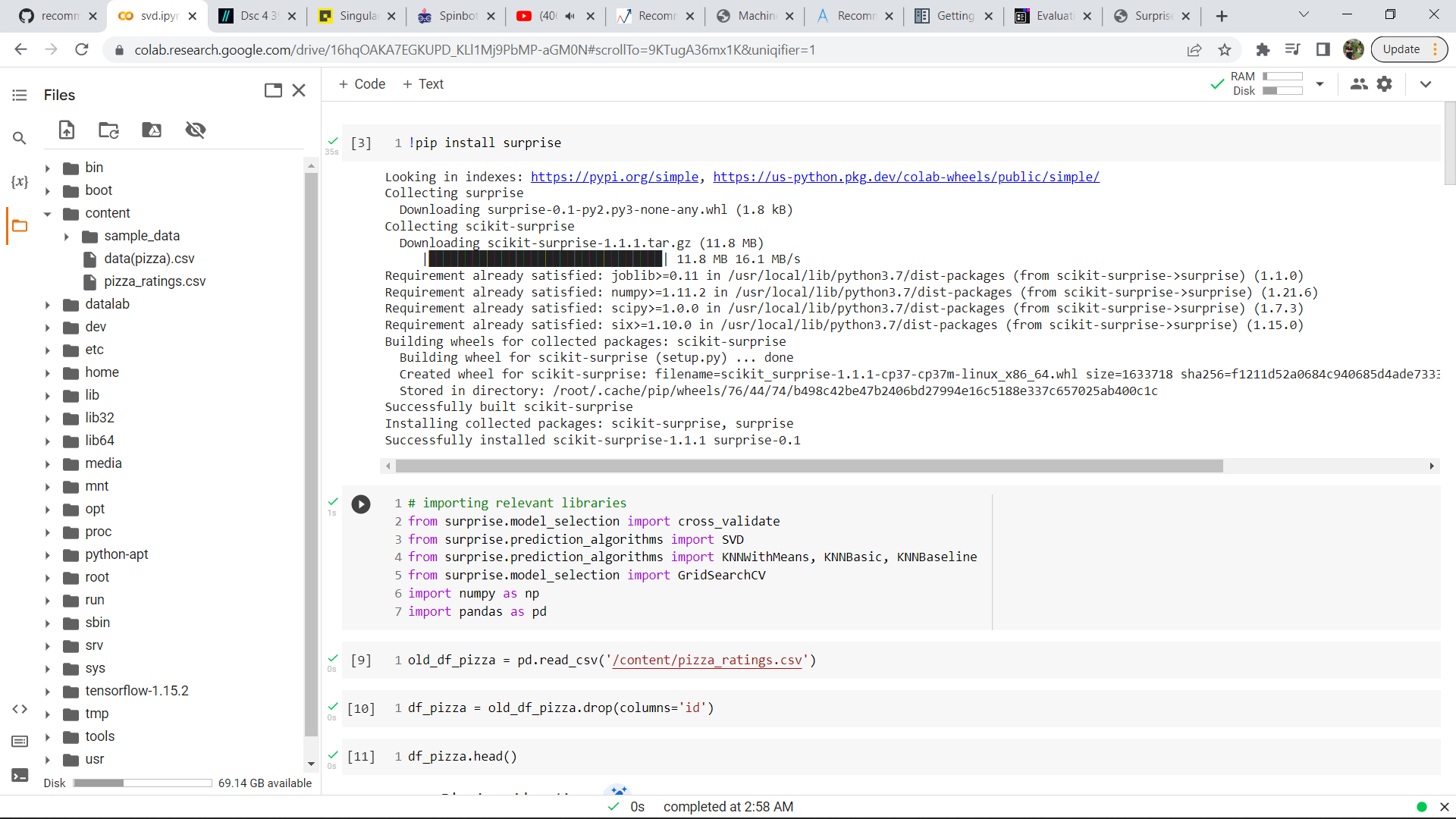
The more the calculation is familiar with the client, the more precise its proposals will be. Be that as it may, numerous clients are reluctant to surrender individual data, particularly given a few high-profile instances of client information spills as of late. Be that as it may, without this client information, the suggestion motor can't work successfully. Subsequently, building trust between the business and clients is vital.

Numerous organizations are flourishing thanks to suggestion motors. While they in all actuality do bring colossal open doors, it is fundamental to know about the many difficulties inborn to the innovation to use it without limit. We wouldn't suggest anything less.

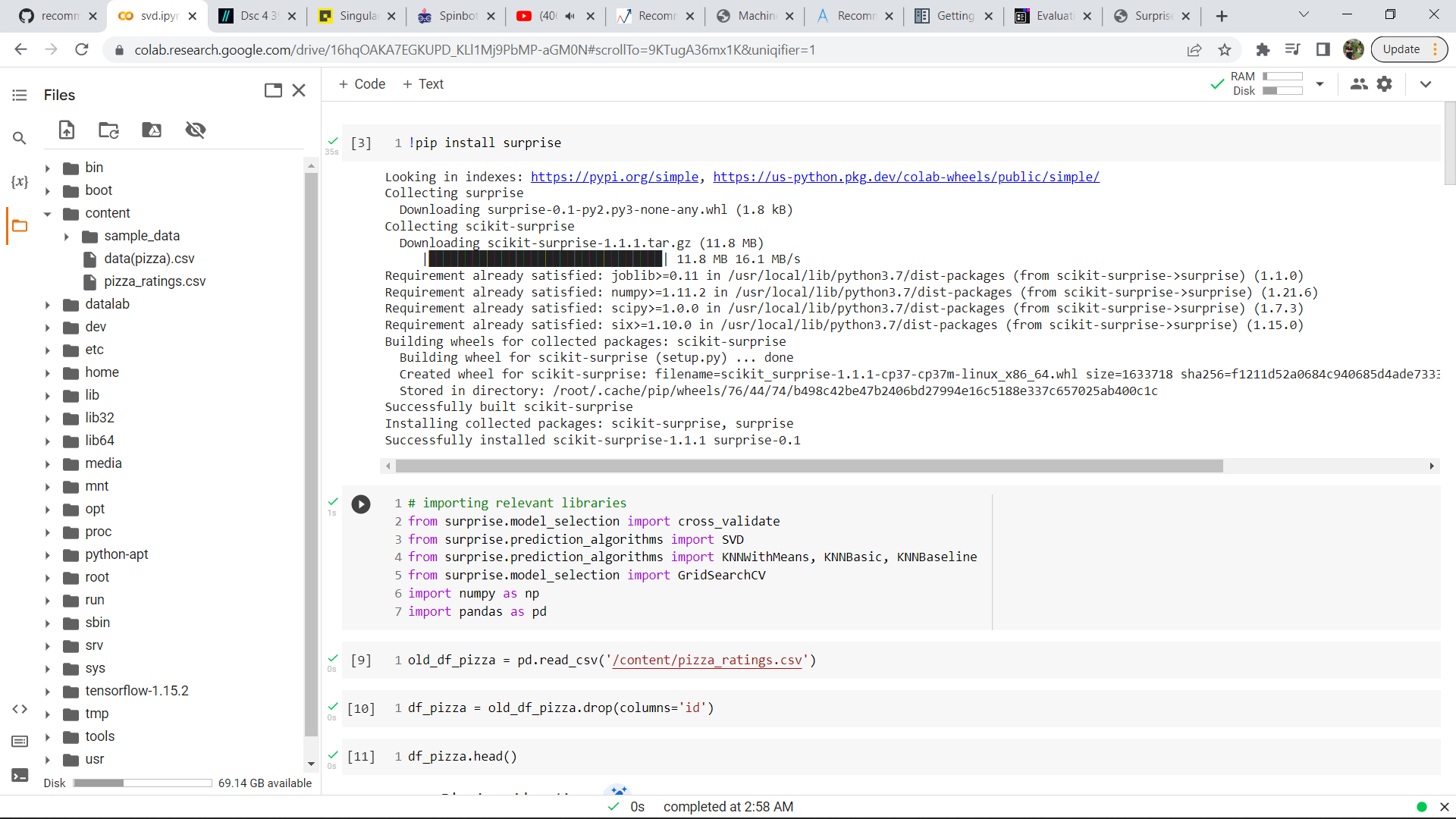
# IMPLEMENTATION

* 1. **DATA PREPROCESSING**
* Install surprise, surprise is a python scikit for building and analyzing recommender systems.

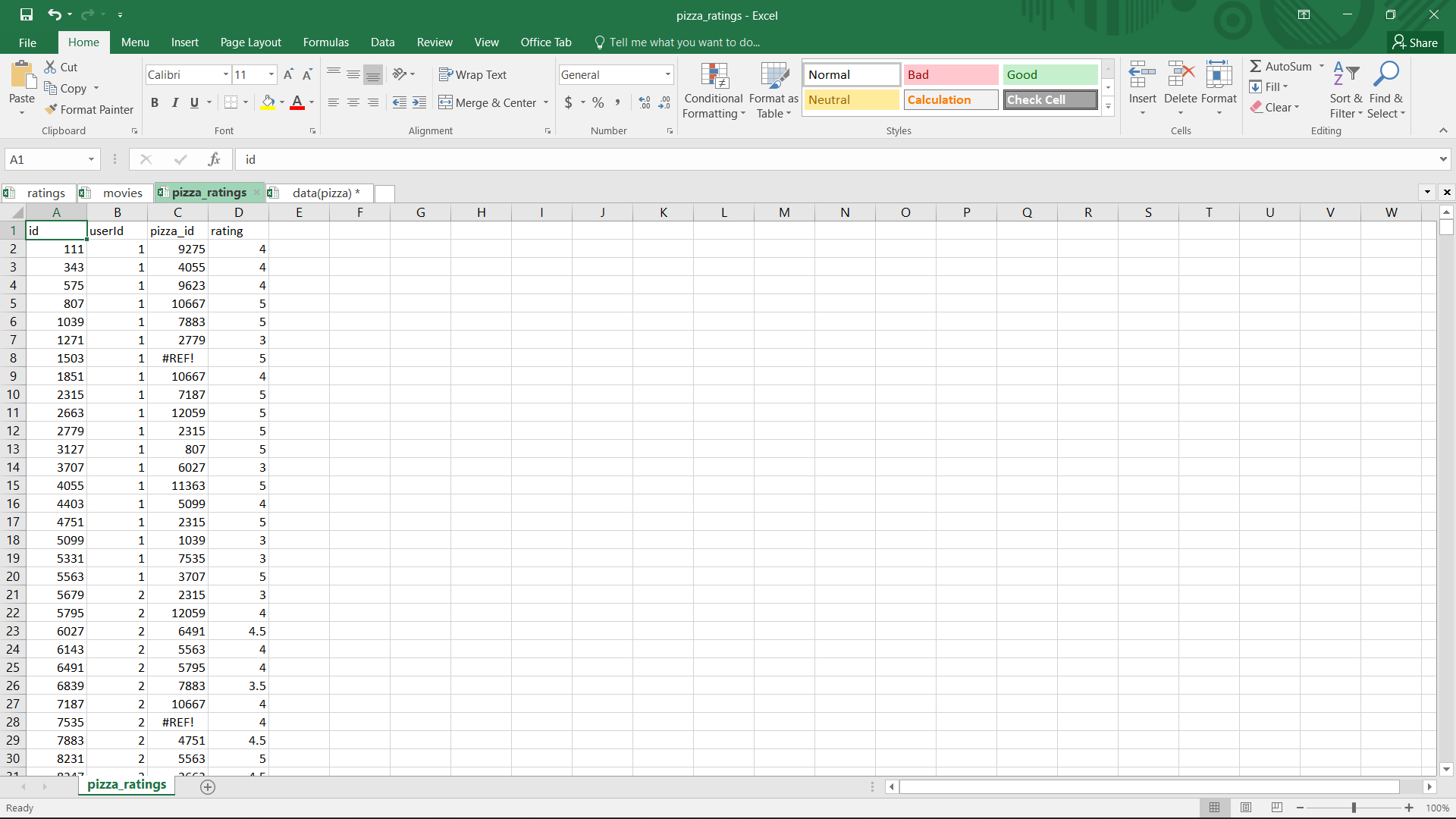
it has pre built algorithms and RMSE calculation methods as well as data sets

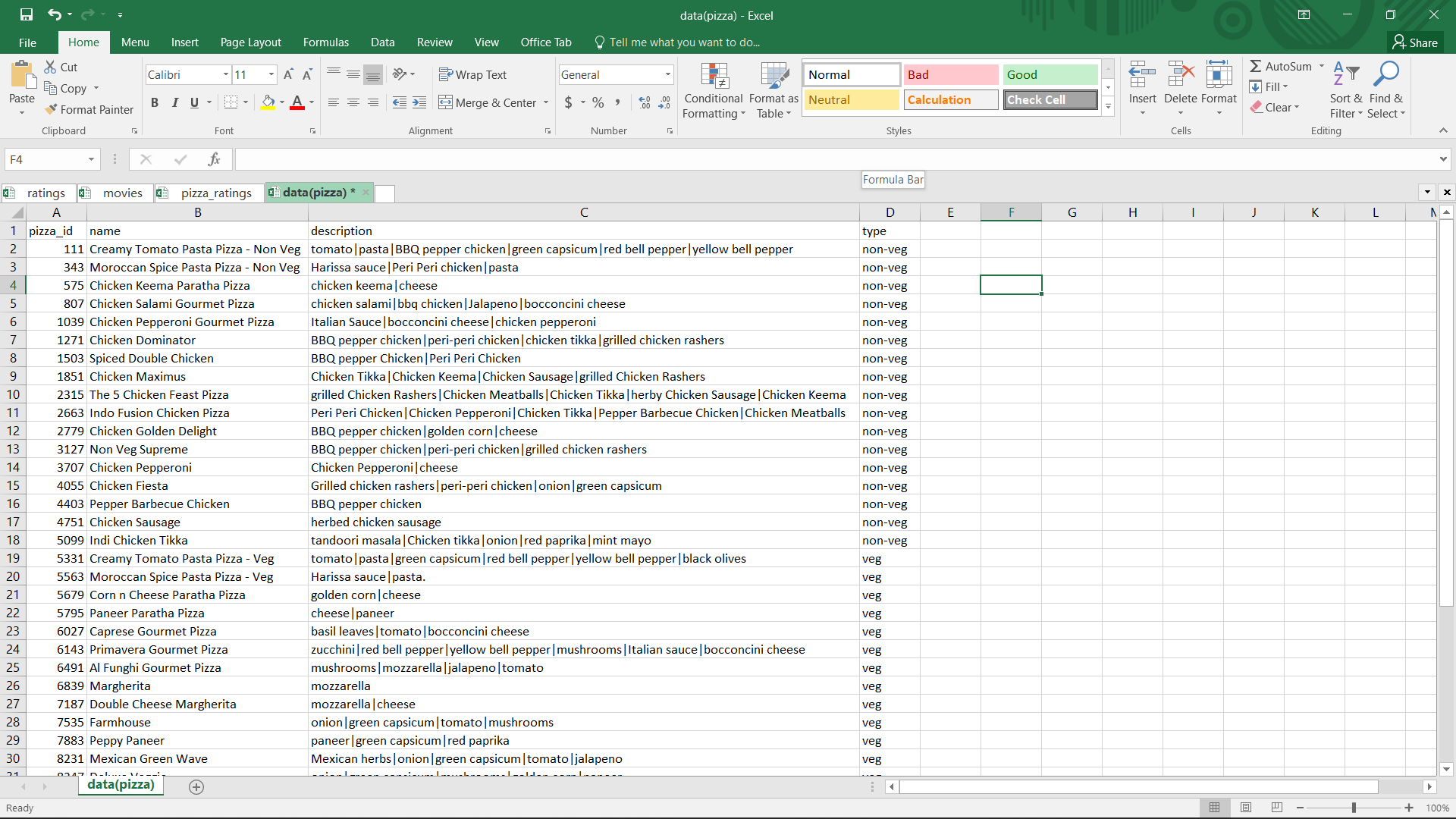


* import and install all packages and libraries needed for recommender



* I have made 2 csv files first has user Id and Pizza (denoted by their id) which are rated, second one has pizza id, pizza’s name corresponding to their ID and description of each pizza which have selective ingredients from a set of 22 ingredients.



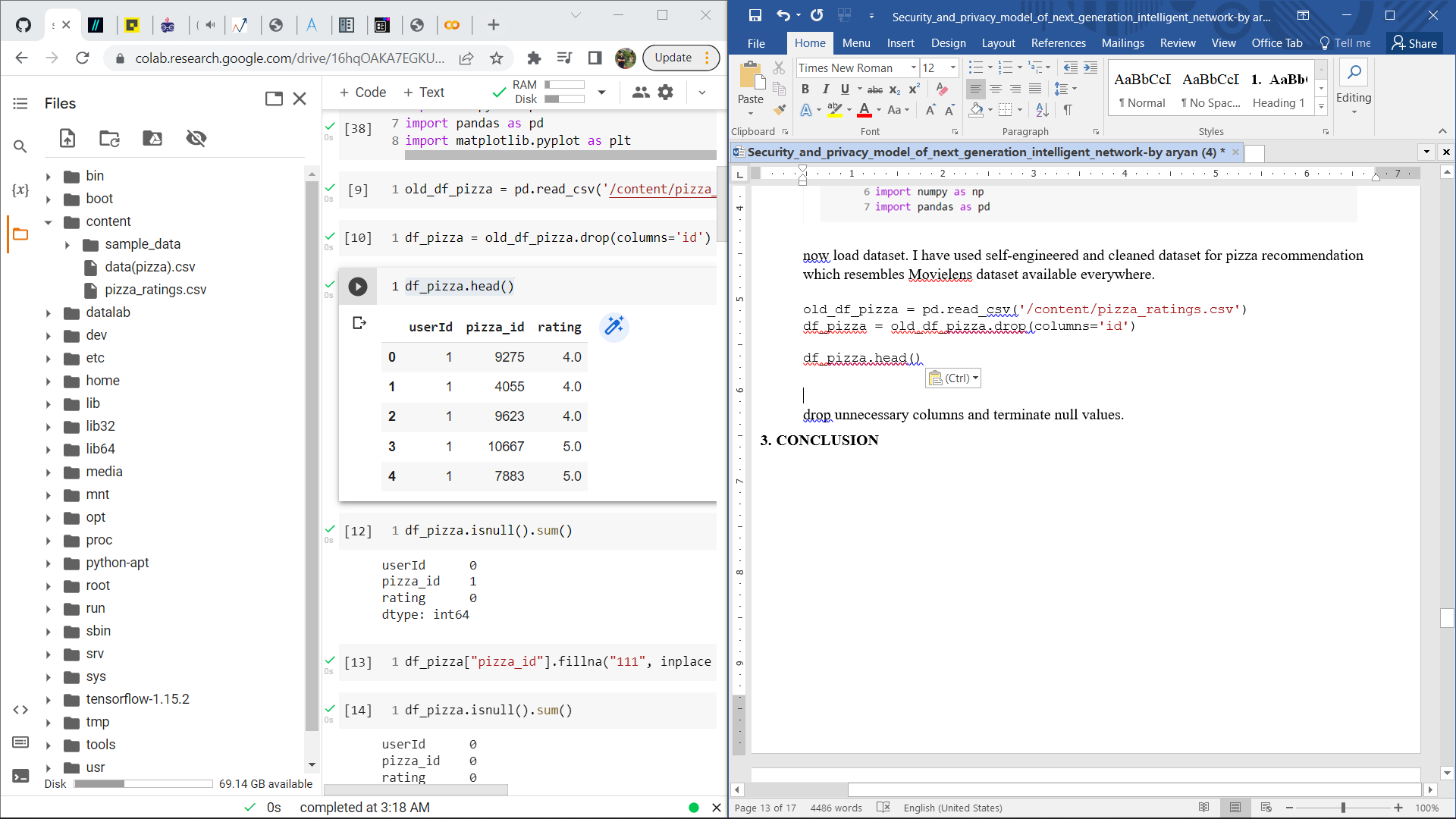


* now load dataset. I have used self-engineered and cleaned dataset for pizza recommendation which resembles Movielens dataset available everywhere.
* drop unnecessary columns and terminate null values.

old\_df\_pizza = pd.read\_csv('/content/pizza\_ratings.csv')

df\_pizza = old\_df\_pizza.drop(columns='id')

df\_pizza.head()

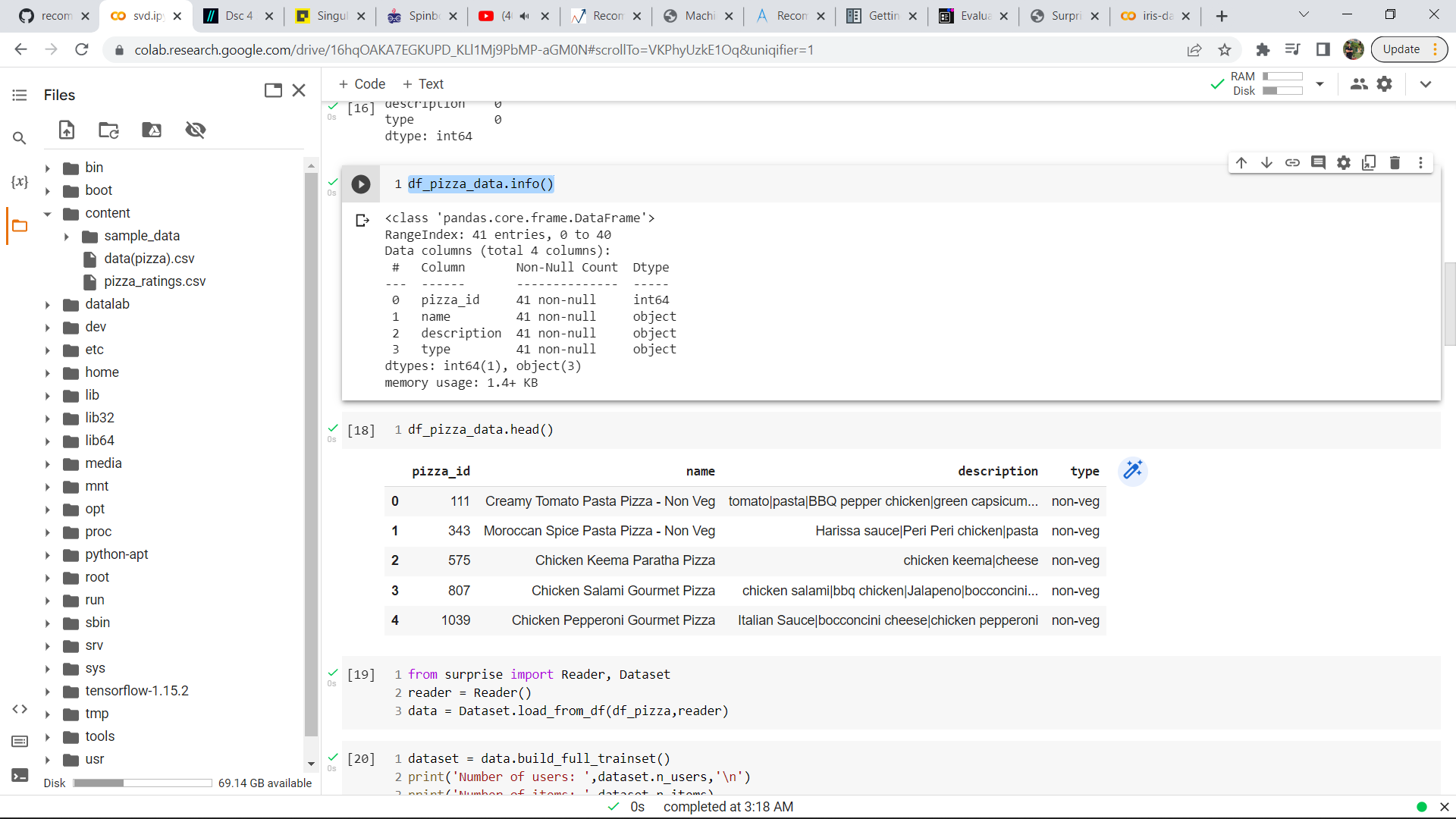


df\_pizza.isnull().sum()

df\_pizza\_data = pd.read\_csv('/content/data(pizza).csv')

df\_pizza\_data.isnull().sum()

df\_pizza\_data.info()



* All data preparations are completed now to fit data in surprise import reader and dataset classes.

from surprise import Reader, Dataset

reader = Reader()

data = Dataset.load\_from\_df(df\_pizza,reader)

dataset = data.build\_full\_trainset()

print('Number of users: ',dataset.n\_users,'\n')

print('Number of items: ',dataset.n\_items)

**DETERMINE MOST COMPATIBLE MODEL**

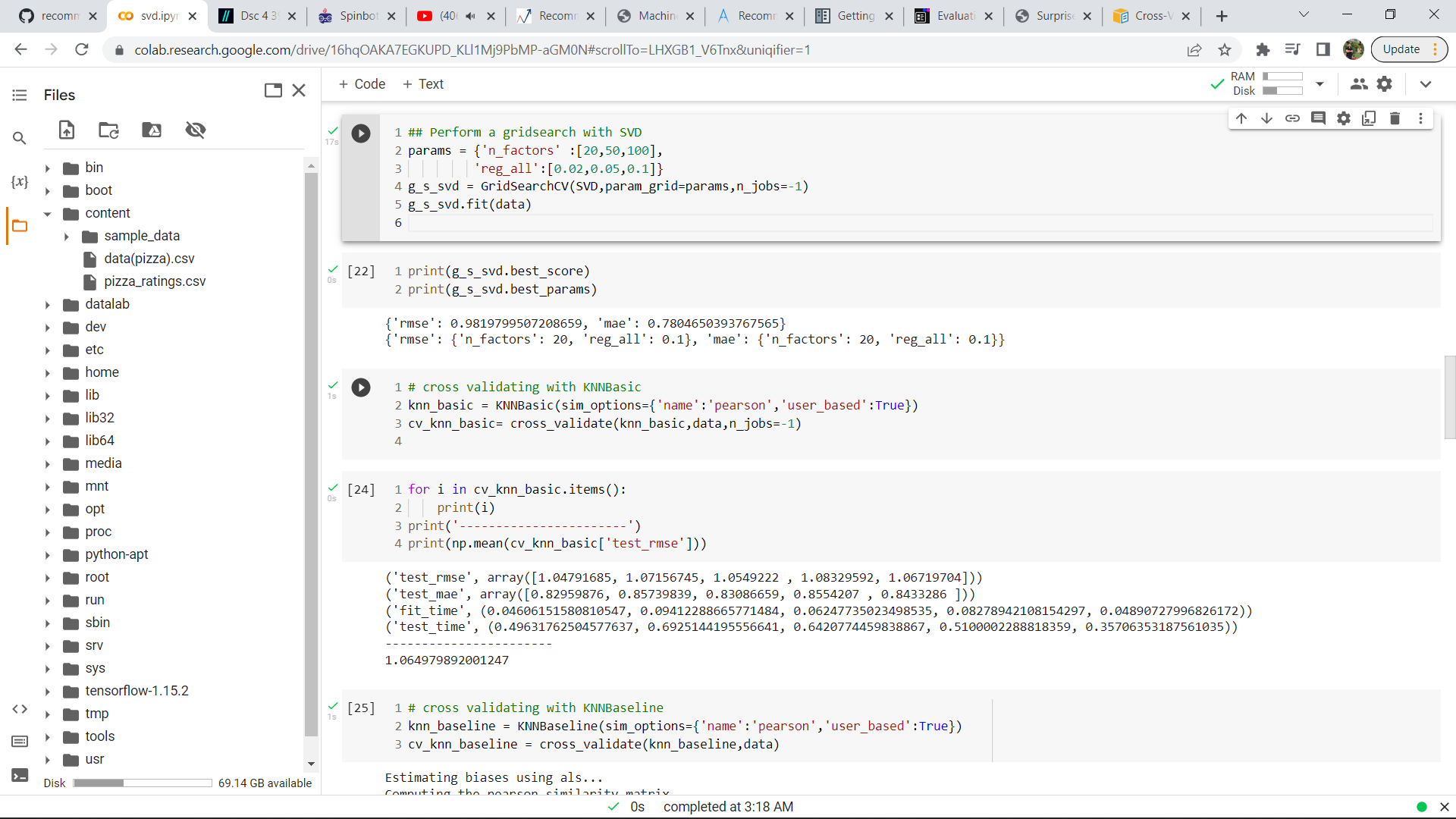
Now, compare the all the models and see which ones perform best. For consistency sake, use RMSE to evaluate models.

here RMSE is calculated for SVD, KNNWithMeans, KNNBasic, KNNBaseline and cross validation is performed.

**CROSS VALIDATION**

Cross-validation is a procedure for assessing ML models via preparing a few ML models on subsets of the accessible information and assessing them on the integral subset of the information. Utilize cross-approval to identify overfitting, ie, failing to generalize.

you can utilize the k-overlap cross-approval strategy to perform cross-approval. In k-overlap cross-approval, you split the information into k subsets of information (otherwise called folds). You train a ML model on everything except one (k-1) of the subsets, and afterward assess the model on the subset that was not utilized for preparing. This interaction is rehashed k times, with an alternate subset held for assessment (and rejected from preparing) each time.[3]



# cross validating with KNNBasic

knn\_basic = KNNBasic(sim\_options={'name':'pearson','user\_based':True})

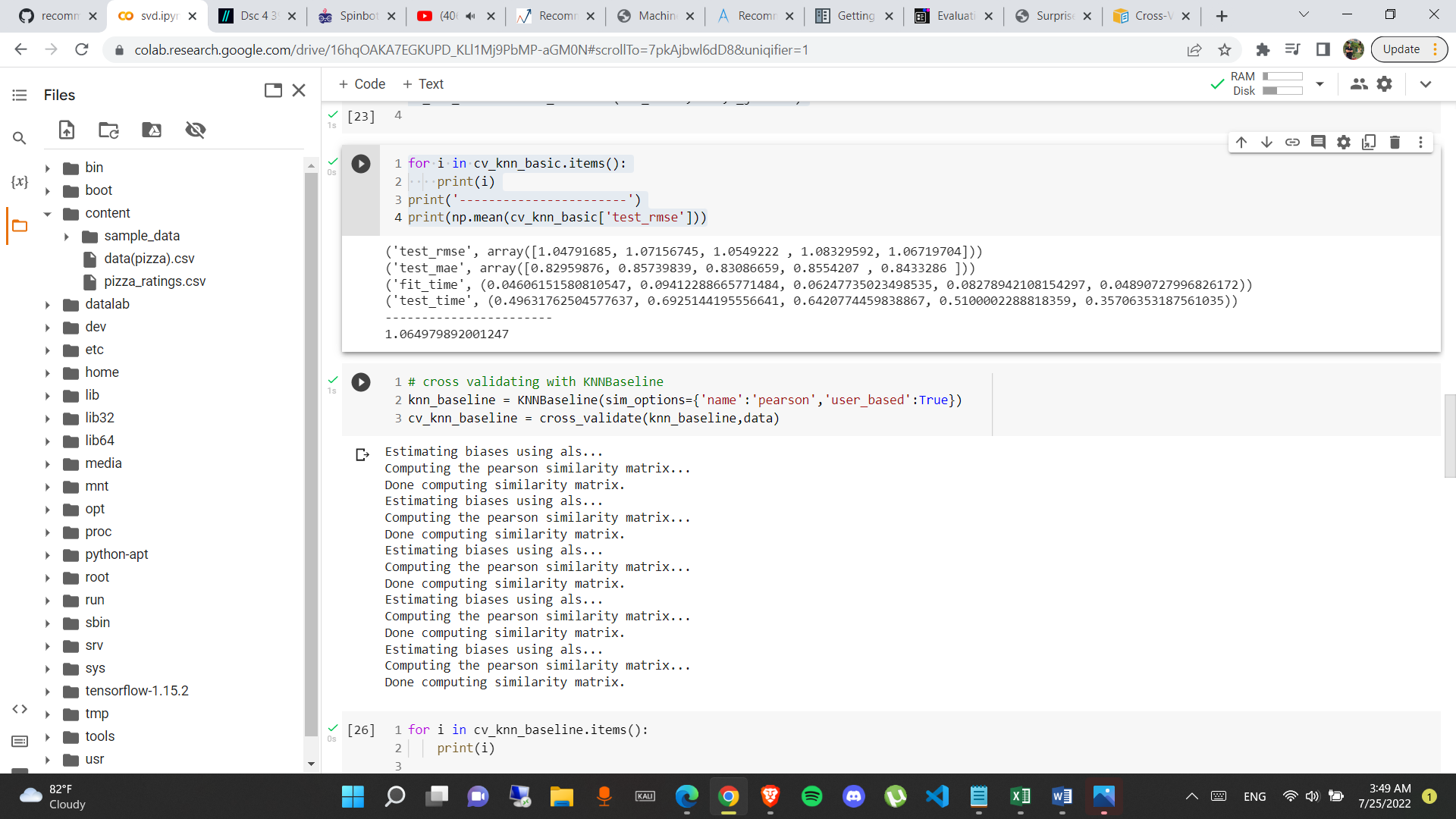
cv\_knn\_basic= cross\_validate(knn\_basic,data,n\_jobs=-1)

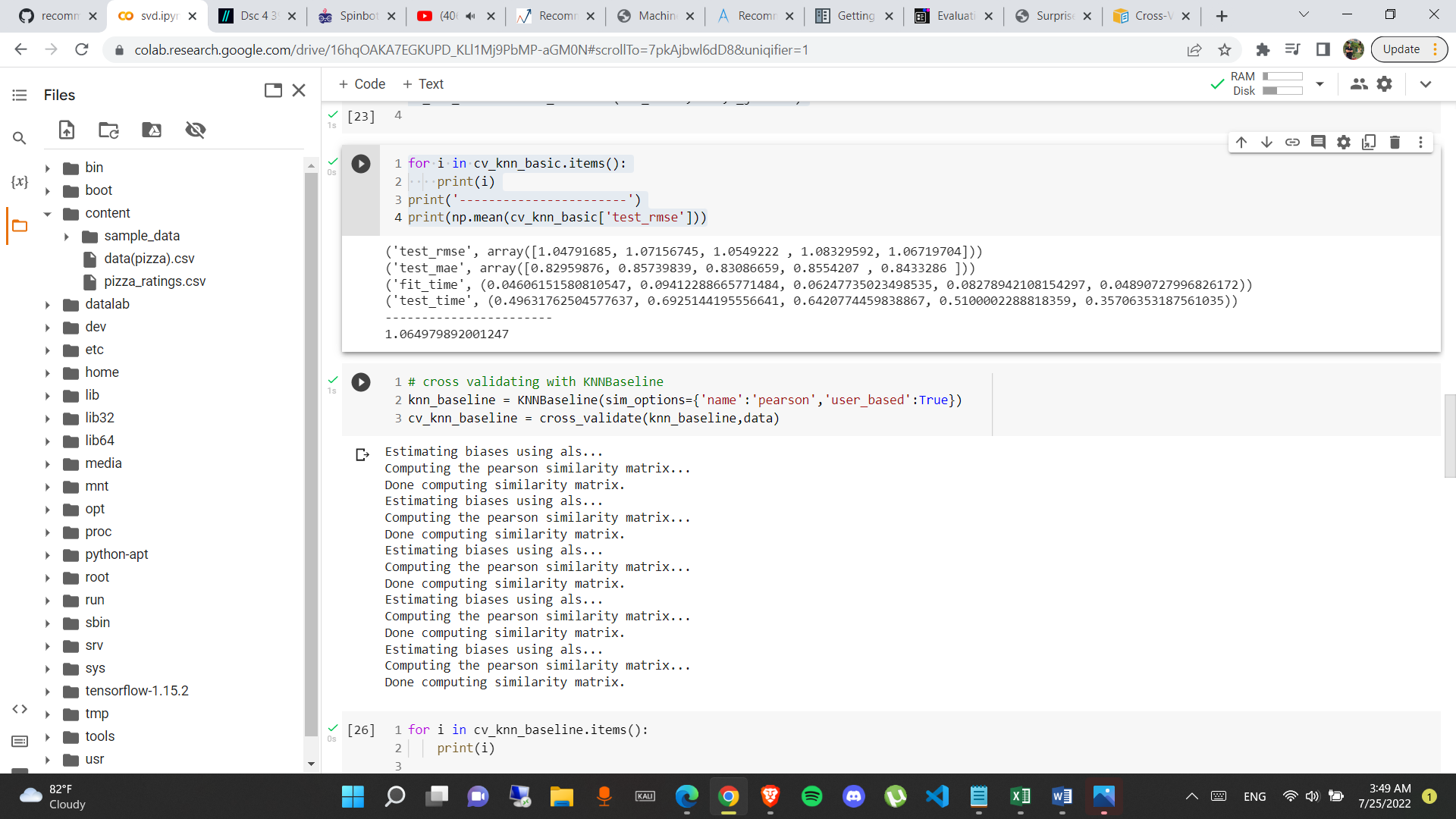
for i in cv\_knn\_basic.items():

    print(i)

print('-----------------------')

print(np.mean(cv\_knn\_basic['test\_rmse']))





* Based on outputs SVD is the best performing model. SVD for collaborative filtering for user –user relation is one the most widely used and easy to understand for recommenders.

Now to overcome a limitation of recommenders called “cold start” which is basically unavailability of data at start of recommendation system and make the system more dynamic specially tailored for every user to give them recommendations based on their current choices also.

A function pizza\_rater is used to take ratings from new users based on their favorite topping on pizza and if the user have not tried the pizza suggested they can directly choose to skip.

def pizza\_rater(pizza\_df,num, genre=None):

    userID = 1000

    rating\_list = []

    while num > 0:

        if genre:

            pizz = pizza\_df[pizza\_df['description'].str.contains(genre)].sample(1)

        else:

            pizz = pizza\_df.sample(1)

        print(pizz)

        rating = input('How do you rate this pizza on a scale of 1-5, press n if you have not seen :\n')

        if rating == 'n':

            continue

        else:

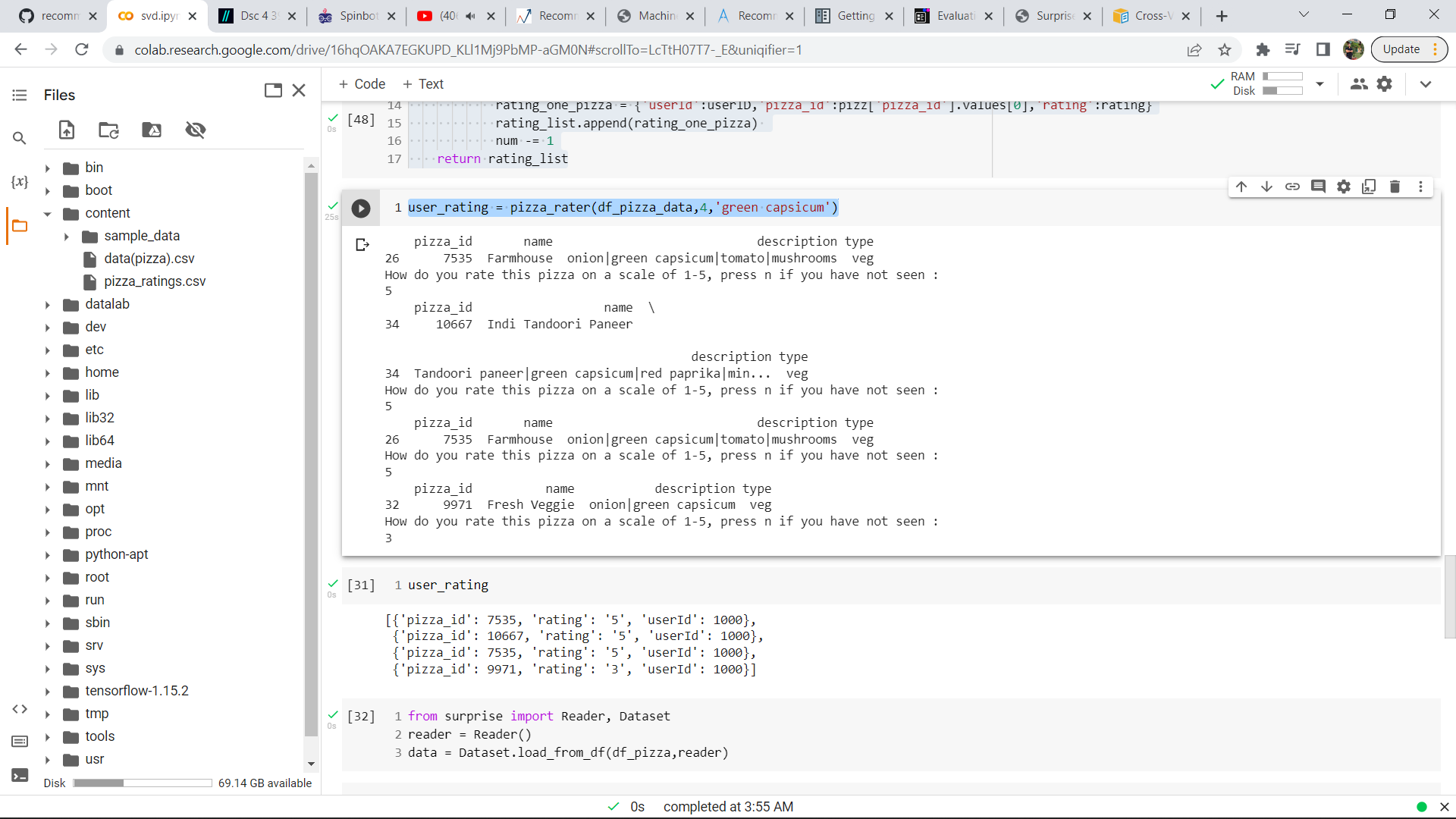
            rating\_one\_pizza = {'userId':userID,'pizza\_id':pizz['pizza\_id'].values[0],'rating':rating}

            rating\_list.append(rating\_one\_pizza)

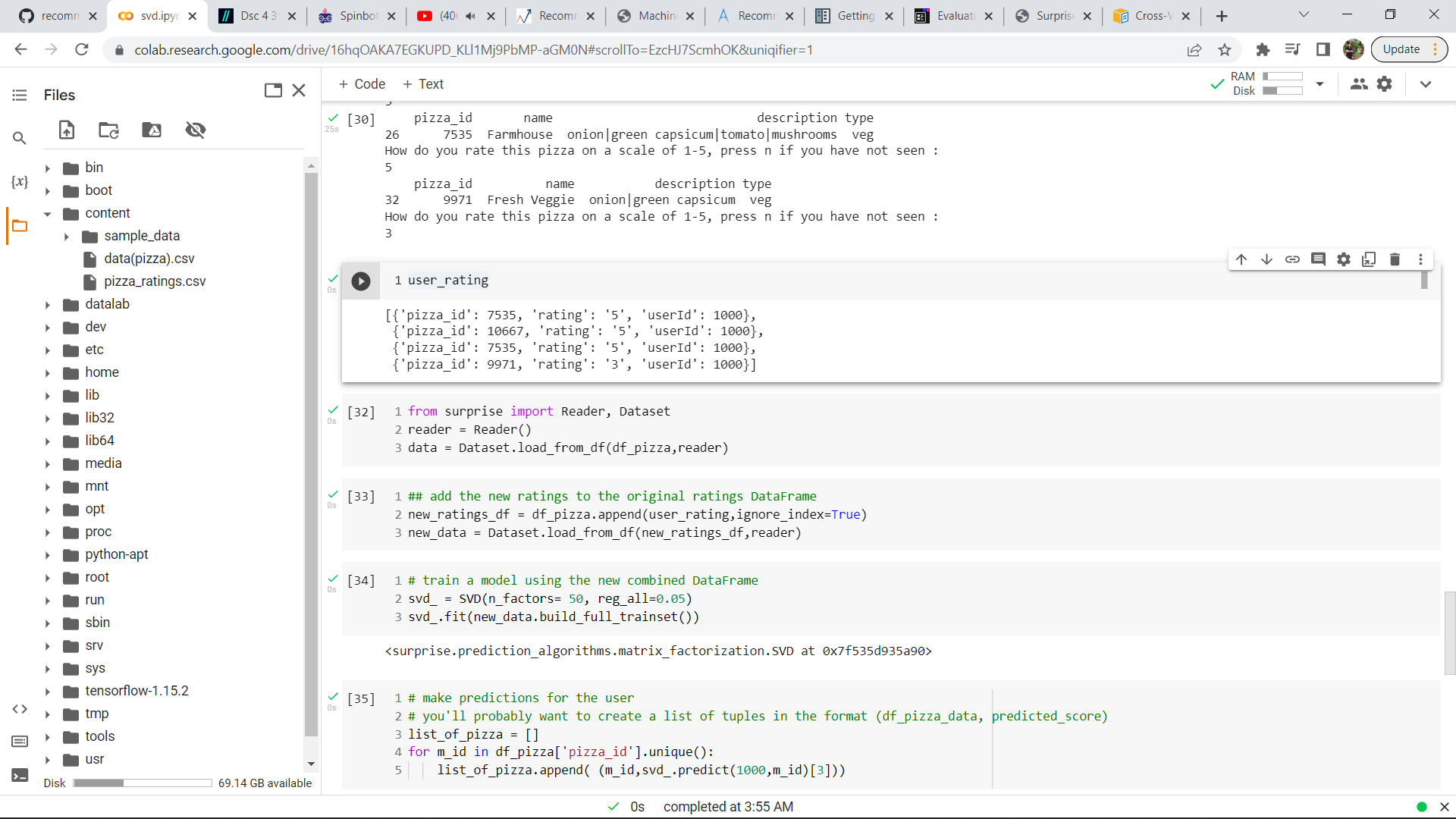
            num -= 1

    return rating\_list

user\_rating = pizza\_rater(df\_pizza\_data,4,'green capsicum')



user\_rating



from surprise import Reader, Dataset

reader = Reader()

data = Dataset.load\_from\_df(df\_pizza,reader)

#append new rating list in dataset

new\_ratings\_df = df\_pizza.append(user\_rating,ignore\_index=True)

new\_data = Dataset.load\_from\_df(new\_ratings\_df,reader)

# fit the new dataframe in SVD

svd\_ = SVD(n\_factors= 50, reg\_all=0.05)

svd\_.fit(new\_data.build\_full\_trainset())

# make recommendation for user

# you'll probably want to create a list of tuples in the format (df\_pizza\_data, predicted\_score)

list\_of\_pizza = []

for m\_id in df\_pizza['pizza\_id'].unique():

    list\_of\_pizza.append( (m\_id,svd\_.predict(1000,m\_id)[3]))

sort your recommendations in decreasing order

ranked\_pizza = sorted(list\_of\_pizza,key=lambda x:x[1],reverse=True)

def recommended\_pizza(user\_ratings,pizza\_title\_df,n):

        for idx, rec in enumerate(user\_ratings):

            title = pizza\_title\_df.loc[pizza\_title\_df['pizza\_id'] == int(rec[0])]['name']

            print('Recommendation # ',idx+1,': ',title,'\n')

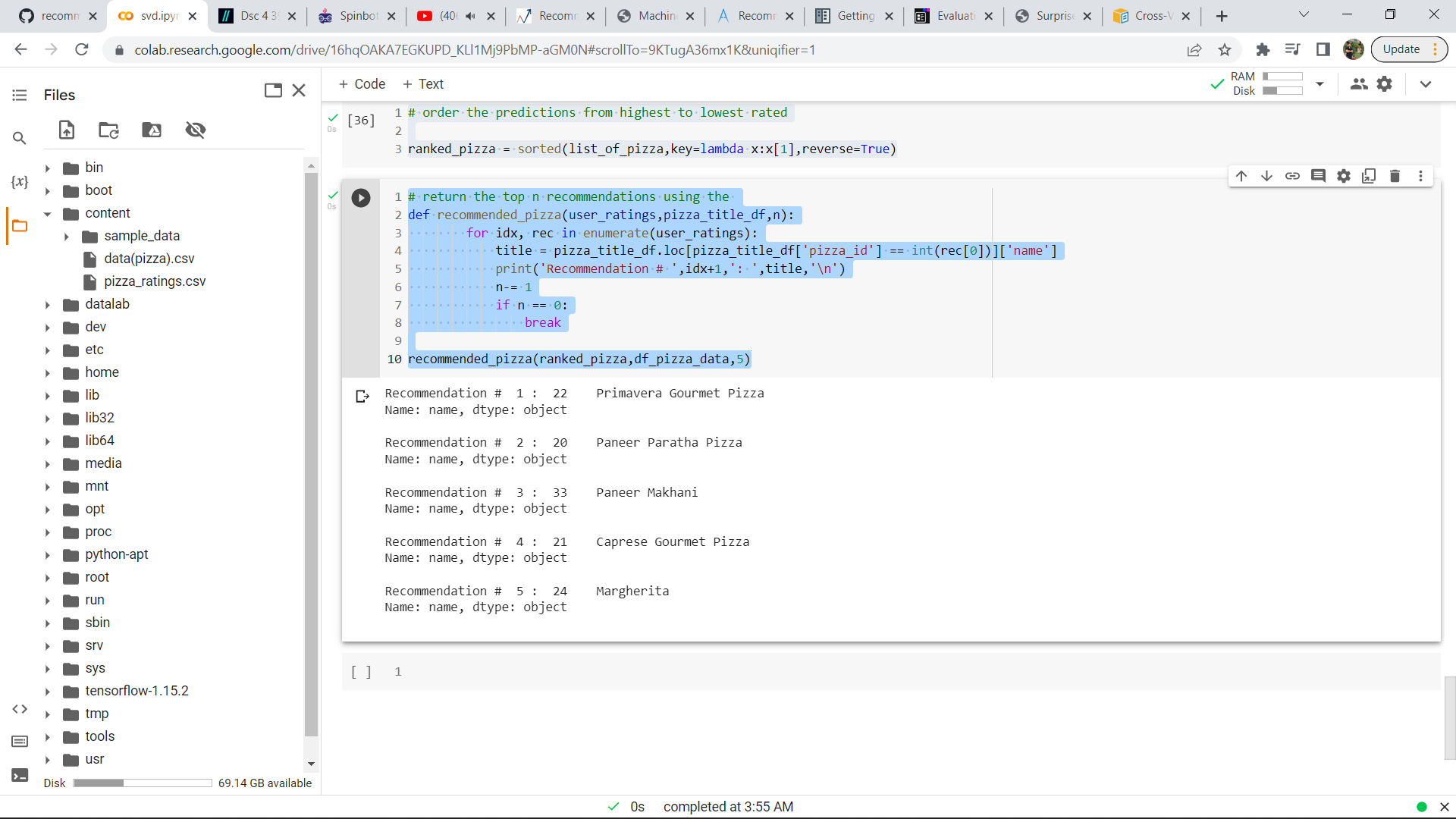
            n-= 1

            if n == 0:

                break

recommended\_pizza(ranked\_pizza,df\_pizza\_data,5)

now finally the output through the recommender are:



# CONCLUSION

A recommender is successfully made whose sole purpose is to recommend you pizza according to your best interest. This recommender works on the basis of collaborative filtering for user-user relation using singular value decomposition. The model is derived considering various limitations of recommenders. Cold start is solved by taking input from new users in form of user rating. Data is carefully cleaned and processed to minimize error.

# REFERENCES

1. Trattner, Christoph & Elsweiler, David. (2017). Food Recommender Systems: Important Contributions, Challenges and Future Research Directions.
2. Aberg, J. (2006, March 27-29)\*. Dealing with malnutrition: A meal planning system for elderly. In Proceedings AAAI Spring Symposium SS-06-01: Argumentation for consumers of healthcare, (pp. 1-7). Retrieved from https://www.researchgate.net/publication/221250510\_Dealing\_with\_Malnutrition\_A\_Meal\_ Planning\_System\_for\_Elderly
3. <https://docs.aws.amazon.com/machine-learning/latest/dg/cross-validation.html>
4. <https://medium.com/recombee-blog/machine-learning-for-recommender-systems-part-1-algorithms-evaluation-and-cold-start-6f696683d0ed>
5. <https://www.analyticsvidhya.com/blog/2021/07/recommendation-system-understanding-the-basic-concepts/>
6. <https://learn.co/lessons/dsc-4-39-04-singular-value-decomposition-numpy-scipylab>
7. G. Linden, B. Smith and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," in IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, Jan.-Feb. 2003, doi: 10.1109/MIC.2003.1167344.
8. M. Balabanovic and Y. Shoham "Content-Based Collaborative Recommendation" Comm. ACM pp. 66-72 Mar. 1997.
9. G.D. Linden J.A. Jacobi and E.A. Benson Collaborative Recommendations Using Item-to-Item Similarity Mappings 2001.
10. B.M. Sarwar "Item-Based Collaborative Filtering Recommendation Algorithms" 10th Int'l World Wide Web Conference pp. 285-295 2001.