A picture containing text, metalware, gear

Description automatically generatedLogo

Description automatically generated with low confidence

**RECOMMENDATION SYSTEM BASED ON AGE AND GENDER**

**A PROJECT REPORT**

***Submitted by***

**MIRUTHUVIKASINI.S (18BCS021)**

**PRAGATHI.R (18BCS038)**

**RAGHAVI.S(18BCS057)**

***In partial fulfilment for the award of the degree***

*of*

**BACHELOR OF ENGINEERING**

in

**COMPUTER SCIENCE AND ENGINEERING**

**KUMARAGURU COLLEGE OF TECHNOLOGY**

**COIMBATORE-641 049**

(An Autonomous Institution Affiliated to Anna University, Chennai)

**January 2022**

Logo

Description automatically generated with low confidenceA picture containing logo

Description automatically generated**KUMARAGURU COLLEGE OF TECHNOLOGY**

**COIMBATORE 641 049**

(An Autonomous Institution Affiliated to Anna University, Chennai)

**BONAFIDE CERTIFICATE**

Certified that this project report **“RECOMMENDATION SYSTEM BASED ON AGE AND GENDER”** is the bonafide work of **S.MIRUTHUVIKASINI (18BCS021) ,R.PRAGATHI (18BCS038) AND S.RAGHAVI (18BCS057)”** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **Dr. Devaki. P, Ph.D.,** | **-----------------** |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Department of Computer Science and  Engineering, | Department of Computer Science and Engineering, |
| Kumaraguru College of Technology | Kumaraguru College of Technology |
| Coimbatore – 641 049. | Coimbatore – 641 049. |

The candidates with University register number **18BCS021, 18BCS038, 18BCS057** were examined in the Project Viva-Voce examination held on …………………

Internal Examiner External Examiner

**DECLARATION**

We affirm that the project work titled **“RECOMMENDATION SYSTEM BASED ON AGE AND GENDER”** being submitted in partial fulfillment for the award of B.E Computer Science and Engineering is the original work carried out by us. It has not formed the part of any other project work submitted for the award of any degree or diploma, either in this or any other University.

**S. MIRUTHUVIKASINI (18BCS021)**

**R. PRAGATHI (18BCS038)**

**S. RAGHAVI (18BCS057)**

I certify that the declaration made above by the candidates is true.

----------------------

Assistant Professor,

Department of Computer Science and Engineering,

Kumaraguru College of Technology,

Coimbatore – 641 049.

**ACKNOWLEDGEMENT**

We express our profound gratitude to the management of Kumaraguru College of Technology for providing as with the required infrastructure that enabled us to successfully complete the project.

We extend our gratefulness to our Principal, **Dr. D. Saravanan,** for providing us the necessary facilities to pursue the project.

We would like to acknowledge **Dr. P. Devaki,** Professor and Head, Department of Computer Science and Engineering, for her support and encouragement throughout this project.

We thank our project coordinator **Dr. L. Latha,** Professor, Department of Computer Science and Engineering and guide **--------------,** AssistantProfessor, Department of Computer Science and Engineering, for their constant and continuous effort, guidance and valuable time.

Our sincere and hearty thanks to staff members of Department of Computer Science and Engineering of Kumaraguru College of Technology for their well wishes, timely help and support rendered to us during our project. We are greatly indebted to our family, relatives and friends, without whom life would have not been shaped to this level.

**S.MIRUTHUVIKASINI**

**R.PRAGATHI**

**S.RAGHAVI**

### **TABLE OF CONTENTS**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | | | **PAGE NO.** | | | |
|  | **ABSTRACT** | | | 6 | | | |
| 1 | **INTRODUCTION** | | | 7 | | | |
|  | 1.1 | CONCEPTUAL STUDY OF THE PROJECT | | 7 | | | |
|  | 1.2 | OBJECTIVES OF THE PROJECT | | 7 | | | |
|  | 1.3 | SCOPE OF THE PROJECT | | 8 | | | |
| 2 | **LITERATURE REVIEW** | | | 8 | | | |
|  | 2.1 | LITERATURE REVIEW OF JOURNALS | | 8 | | | |
| 3 | **PROBLEM DEFINITION** | | | 11 | | | |
| 4 | **PROPOSED SYSTEM** | | | 11 | | | |
|  | 4.1 | METHODOLOGY | | 11 | | | |
|  | 4.2 | FLOW DIAGRAM | 12 | | |
|  | 4.3 | IMPLEMENTATION | | | 15 | |
| 5 | **SYSTEM REQUIREMENTS** | | | 31 | | | |
|  | 5.1 | REQUIREMENTS | | 31 | | | |
| 6 | **CONCLUSION** | | | 31 | | | |
| 7 | **APPENDIX** | | | 31 | | | |

### **ABSTRACT**

This paper proposes an age and gender prediction method from facial images using convolutional neural network (CNN). The CNN architecture involves two levels which are feature extraction and classification. Our CNN is pre-processed and trained on an UTKFace dataset. Out of the many applications of age and gender prediction, we apply it to improve marketing strategy by recommending products in any online platform based on the target audience. Our recommendation system is built using User-based Collaborative filtering where SVD Algorithm is applied.

**1.INTRODUCTION**

**1.1 CONCEPTUAL STUDY OF THE PROJECT:**

Facial analysis has gained much recognition in recent times. Age and gender predictions of unfiltered faces classify the real-world facial images into predefined age and gender. Using deep-CNN method provides an efficient performance compared to the traditional conventional methods[4]. We can use Haar cascade[8] approach which is an object detection algorithm used to identify facial images from real time videos also. The predicted age and gender is used in many applications such as access control, law, human-computer interaction, marketing, visual surveillance and so on.

Once the image is given as input, the face of the customer is detected and analysed for age and gender prediction. All the Data necessary for the recommendation system should be previously fed by us. We have used Collaborative filtering technique. Collaborative filtering is a technique that can filter out items that a user might like on the basis of reactions by similar users. It works by searching a large group of people and finding a smaller set of users with tastes similar to a particular user.

This system of approach will improve the marketing standards and can help both the customers to fulfil their needs and also the seller for greatly improving their sales percentage.

* 1. **OBJECTIVE OF THE PROJECT**

We propose an age and gender prediction method from facial images using convolutional neural network (CNN). Our CNN is pre-processed and trained on an UTK-Face dataset. Out of the many applications of age and gender prediction, we apply it to improve marketing strategy by recommending the products based on the target audience.

**1.3 SCOPE OF THE PROJECT**

Marketing strategy is greatly influenced by recommendation system. We are entering a data collection era, everything from our preferences to our search histories are stored and used to recommend us related items.

Example: YouTube recommends us what videos to watch next.

Amazon recommends related items to add to our shopping cart.

Our Work creates a demographic way of product recommendation in online shopping platforms. The face of the customer is detected by uploading the image. By detecting age and gender using our model we will suggest the products .

**2. LITERATURE REVIEW:**

**Strategies used:**

The Physiological parameters such as fingerprint, face, palm etc are the replica of DNA. So, while segmenting the facial feature there are lots of factors to be considered which includes pose, noise, texture, lighting conditions and distance between the object and the camera, Occlusion, Illumination etc. By considering all these features and building a responsive face recognition model has its own great impact on the real-world automation.

**Age Classification:**

After image recognition there are early classification of age was done by calculation ratio between different features of face like nose, eyes, mouth, chin etc. After localizing calculating their sizes and distances, ratio between them is calculated in order to predict age by using conventional methods. Gaussian Mixture Model (GMM) [5] works on facial patches distribution. Gabor image descriptor method is used in used along with Fuzzy-LDA classifier to detect image of face belonging to more than one age group. As a whole Biological Inspired features (BIF) and other manifold-learning features are used for age prediction[7]. Gabor image descriptor and local binary pattern (LBP) [3] were used with hierarchical age classifier method consists of Support Vector Machines (SVM) [5] are used to classify input image to a specific age class by using support vector regression so that the result obtained would be precise.

There are various methods proposed on image recognition all those were acceptable for well aligned and front facing images exactly, thus that proposition also don’t meet the needs. So, these methods give only experimental result on limited data sets. Thus, such methods are inappropriate to use in real time streaming images. Most of the methods described above uses FERET benchmark to develop a most precise system giving accurate results[1]. FERET images were taken to extremely measured complaint and the result obtained from them are highly saturated. It is actually difficult to find out actual advantages of these techniques as FERET benchmark contains filtered images which makes the result more precise[3].

**Gender Classification**

Same as Age classification, Gender classification also has its own significant implementation in image processing world. It may seem easy when we try to classify gender on common but it is not actually true. When it comes to Gender classification, we group into two classes (Male, Female) in general. Previous works were on the neural network trained on a small set of near-frontal face images such as UIUC-IFP-Y, FG-NET and MORPH. Some works used SVM Classifier directly on image intensities[8]. But the model which uses Webers Local texture Descriptor for gender recognition, demonstrated its perfectness comparing with the above models.EEG-Based Age and Gender Prediction Using Deep BLSTM-LSTM Network Model were also made by a couple of researchers [6][9].

**CNN on prediction:**

The first application of Convolutional Neural Network (CNN) is LeNet-5 network by using optical character recognition. In the previous work using deep CNN, model is trained to an extent that accuracy of Age and Gender become 79% using HAAR cascading. Its accuracy could be increased more using more efficient algorithms and more precise architecture of CNN so that it could have been used more in different platforms.

[1]Finally, a Deep convolution method was introduced by Levi and Hassner model where the model is trained with more of unclear and unfiltered images which gave a better result for uneven images. The model is so simple and solves the problem of overfitting and works fine for even unconstrained images. When we talk about the major contribution of Deep CNN it was working successfully even on the challenging ImageNet benchmark. Additionally, it was also used in pose estimation, face parsing, facial keypoint detection, speech recognition. In general Overfitting (Low bias and high variance) is the common problem across most of the machine learning model it is mainly due to minimal size of the data as it learns detailed noise in the training data.

Levi and Hassner Deep CNN was fed with rescaled images of 256 × 256 and a crop of 227 × 227. The network comprises of 3 Convolution Layers with 2 Fully connected layer.

Experimental results proved the small misalignments has created a drastic change over the result. So, to overcome the mentioned variation as well as the problem of overfitting the network training was given in 2 methods one is Centre Crop (Cropped to 227 × 227 around the face centre) and another is Over-sampling (Five 227 × 227 pixel crop regions, Four from the corners of the 256 × 256 face image, and an additional crop region from the centre of the face, along with horizontal reflections). Training the network by translating alignments of the same face made this model to stand out of all the existing systems even with simpler network architecture.

**Recommendation System:**

Taxi hunting recommendation system uses Demographic based filtering. The dynamic system consists of two phases-offline processing and fast online processing,based on geographical data(GPS History).1st phase- data preprocessing(File partition, Hotspot scanning) and offline graph trajectory model constructor using eigen values(Preference trajectory scanning). 2nd phase- Querying: Query input-time and location,Query output- Recommended taxi’s details.[11]

Hotel recommendation system uses Hybrid system approach.  In this paper, collaborative and content based filtering are used hand-in-hand. Firstly, it converts raw data into user-item rating matrix. On applying clustering technique(k means) the data will be split into cluster groups. Finally, association rule mining is used for rules generation and strong rules are generated. With the help of strong rules best items will be recommended. Advantages of this paper are Improved accuracy and Reduced sparsity.[12]

This paper is based on study based on Content-Based Recommendation Systems. Algorithms used: Decision Tress and Bayesian Classifiers. Some examples of content based recommender systems are 1. LIBRA : A Content-Based Book Recommending System, using learning for text categorization.2. CBMRS : A Content-Based Music Recommendation System.3. PRES : A Content-Based Home Improvement Recommender System.4. Cobra : A Content-Based Filtering and Aggregation of Blogs and RSS Feeds.[13]

Product Recommendation System for Supermarket  uses Collaborative filtering technique. Here Recommendation is done on two stages. In the first stage, recommendation is done in prior to the purchasing of product. In the second stage, it recommends the purchasing of associated products.

Product Recommendation System for Supermarket .User- Based method approach:User to user matrix is built by iterating through all user pairs and computing similarity matrix for each pair. Item-based method approach: Product-to product matrix is built by iterating through all item pairs and computing similar matrix for each pair. Compared to the user-based collaborative filtering item-based collaborative filtering is better. Reason :It is difficult to find the similar customers in a supermarket.[15]

Amazon.com Recommendations uses Item-to-Item Collaborative Filtering. Working: Item-to-item collaborative filtering matches each of the user’s purchased and rated items to similar items, then combines those similar items into a recommendation list. To determine the most-similar match for a given item, the algorithm builds a similar-items table by finding items that customers tend to purchase together. Similarity is calculated through cosine measure. Conclusion made from the Survey: The key to item-to-item collaborative filtering’s scalability and performance is that it creates the expensive similar-items table. Scales independently of the catalogue size or the total number of customers; it is dependent only on how many titles the user has purchased or rated. The algorithm is fast even for extremely large data sets. Because the algorithm recommends highly correlated similar items, recommendation quality is excellent.[15]

**Achievements of existing system:**

A deep convolutional method which was proposed by Levi and Hassner model was trained with more of unclear and unfiltered images. They constitute this model as a multi class classification problem. The existing model was originally pre-trained on age and gender called IMDB-WIKI dataset[2][8]. The images from the mentioned dataset were obtained directly from the website with some degrees of variability and the images were fine-tuned on MORPH II which was the other largest facial aging dataset which include age and gender annotations too[2]. And they also used OIU-Adience (original dataset) which consists of unfiltered faces of age and gender classification to finely tune the Levi and Hassner model. They used robust image pre-processing algorithm, which handles the maximum variability observed in the unfiltered real-world faces (facial images). While investigating the accuracy of classification on OIU-dataset both the age and gender classification accuracy achieved the state-of-the-art performance, which outperformed the existing models.

**Disadvantages of existing system:**

The biggest issue with the age prediction model which was trained by Levi and Hassner is it’s heavily biased toward the age group 25-32. This means that Levi ad Hassner age prediction model may predict the 25-32 age group when the actual age belongs to a different age group[1].

**3.PROBLEM DEFINITION:**

To build a gender and age detector that can approximately guess the gender and age of the person (face) in a picture, and recommend the products based on demographic parameters.

**4. PROPOSED SYSTEM:**

The main motto of our system is to recommend products based on demographic parameters such as Age and Gender. This system plays a crucial role in the marketing strategies due to its unique recommendation system. Our application will detect age and gender of users who wants to shop in the online platform, based on online facial analysis and by using recommendation algorithms it will suggest the products for the targeted audience. We will be training a convolutional neural network that will predict the age group and gender from an image containing the face of a person. Output of the above would-be age and gender. By feeding the above as input to the recommendation system which uses Collaborative based filtering technique we reach our goal finally by getting our recommended products.

**MODULES OF OUR PROPOSED SYSTEM:**

**4.1 METHODOLOGY:**

We have lots of Algorithms trained especially for facial recognition. But CNN has its own way of dealing with image processing. A convolution is essentially sliding a filter over the input. Each convolutional layer contains a series of filters known as convolutional kernels.

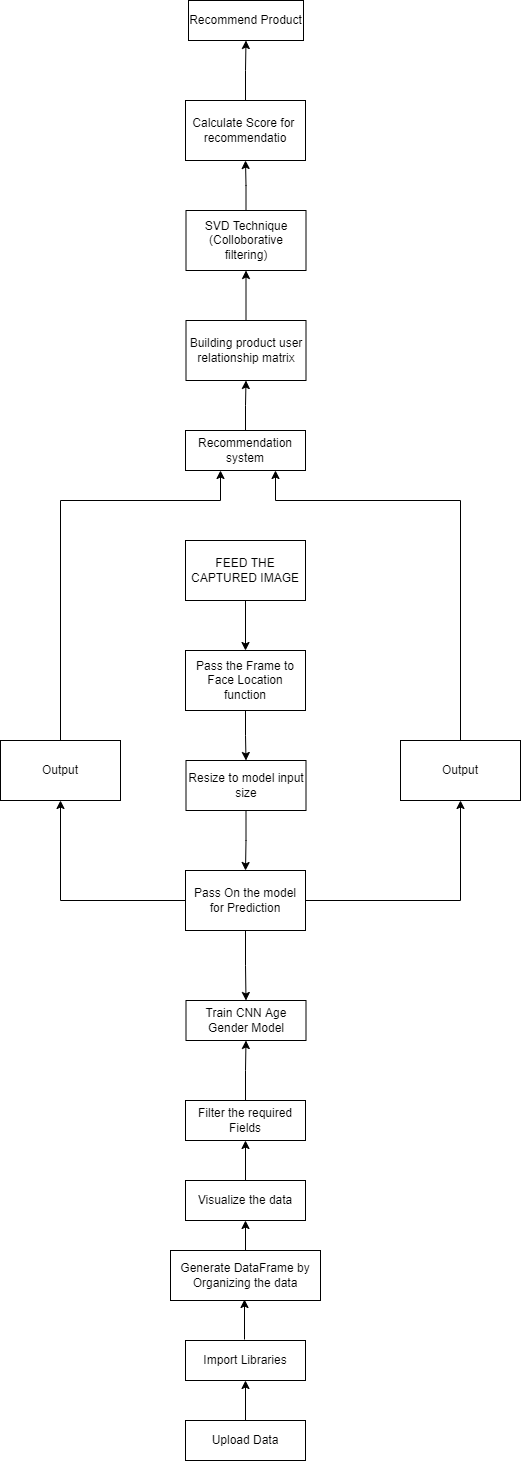
The filter is a matrix of integers that are used on a subset of the input pixel values, the same size as the kernel. Each pixel is multiplied by the corresponding value in the kernel, then the result is summed up for a single value for simplicity representing a grid cell, like a pixel, in the output channel/feature map. Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Moving on, we are going to into fully connected layer which we flatten the final output and feed it to a regular Neural Network for classification purposes.

Flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and back propagation applied to every iteration of training. The model is able to distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique. Finally Output is classified into different classes of age and gender.

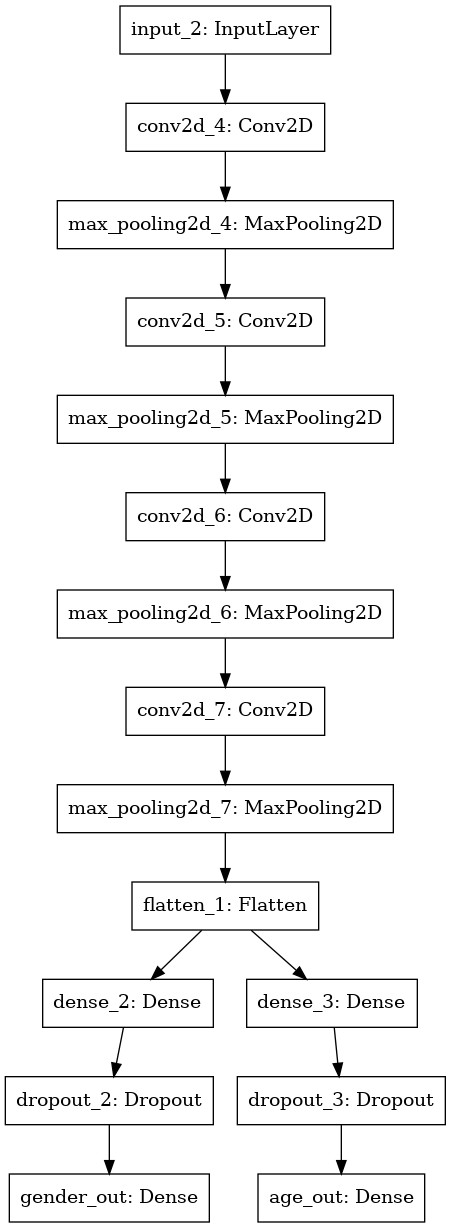
The output (Age and Gender) is passed as input to the Recommendation System. Here we used Collaborative based Filtering technique. Collaborative filtering is a technique that can filter out items that a user might like on the basis of reactions by similar users. It works by searching a large group of people and finding a smaller set of users with tastes similar to a particular user. User-Based Collaborative Filtering is a technique used to predict the items that a user might like on the basis of ratings given to that item by the other users who have similar taste with that of the target user. Many websites use collaborative filtering for building their recommendation system.

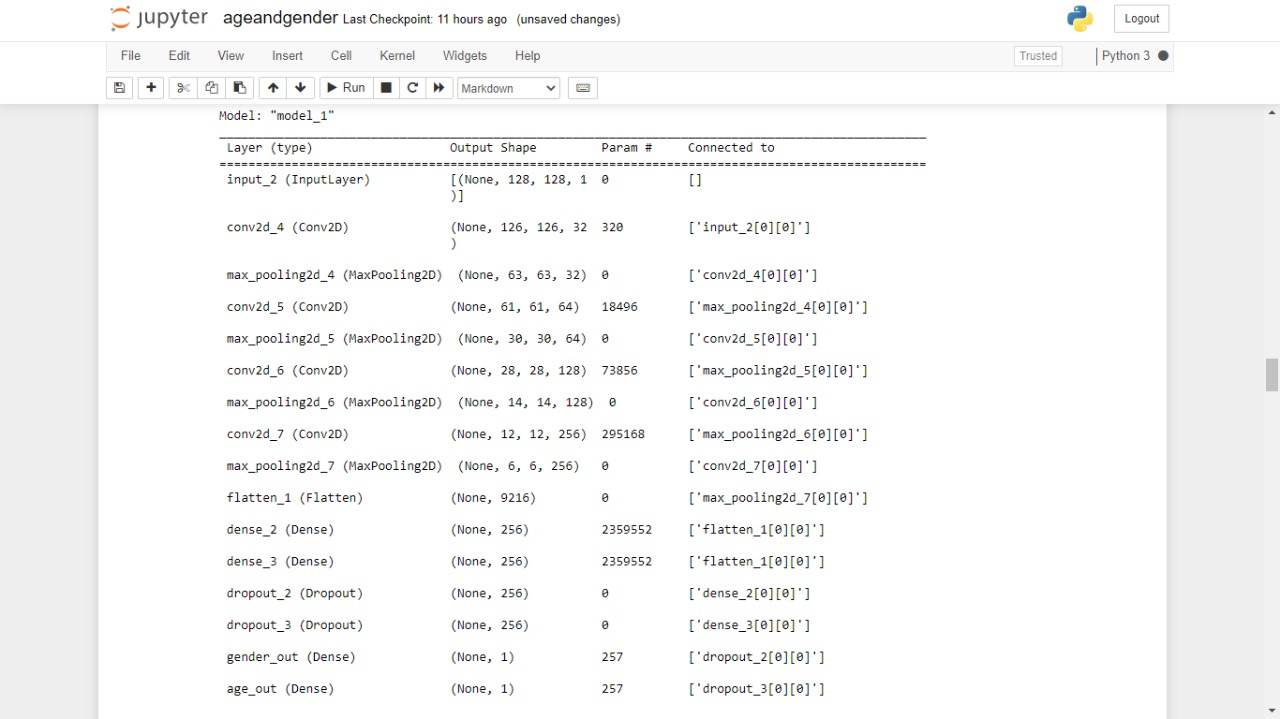
For used based collaborative filtering technique SVD algorithm is used. It is a method from linear algebra that has been generally used as a dimensionality reduction technique in machine learning. It is a matrix factorisation technique, which reduces the number of features of a dataset by reducing the space dimension. The factorisation of this matrix is done by the singular value decomposition. It finds factors of matrices from the factorisation of a high-level (user-item-rating) matrix.

**4.2 FLOW DIAGRAM:**

****

**Network Architecture:**

****

****

**4.3. IMPLEMENTATION:**

**DATA COLLECTION:**

For this python project, we had used the UTKFace dataset; the dataset is available in the public domain. The UTKFace dataset is a large-scale face dataset with long age span (range from 0 to 116 years old). The dataset consists of over 22,000 face images with annotations of age, gender, and ethnicity. The images cover large variation in pose, facial expression, illumination, occlusion, resolution, etc.

#### **Uploading the data**

Mount the drive and navigate to the folder that has the dataset in it.

#### **Import the necessary libraries for loading and viewing the data and Read the Data**:

**Numpy:** for working with the data, cleaning it, formatting it in the required way and deleting irrelevant data.

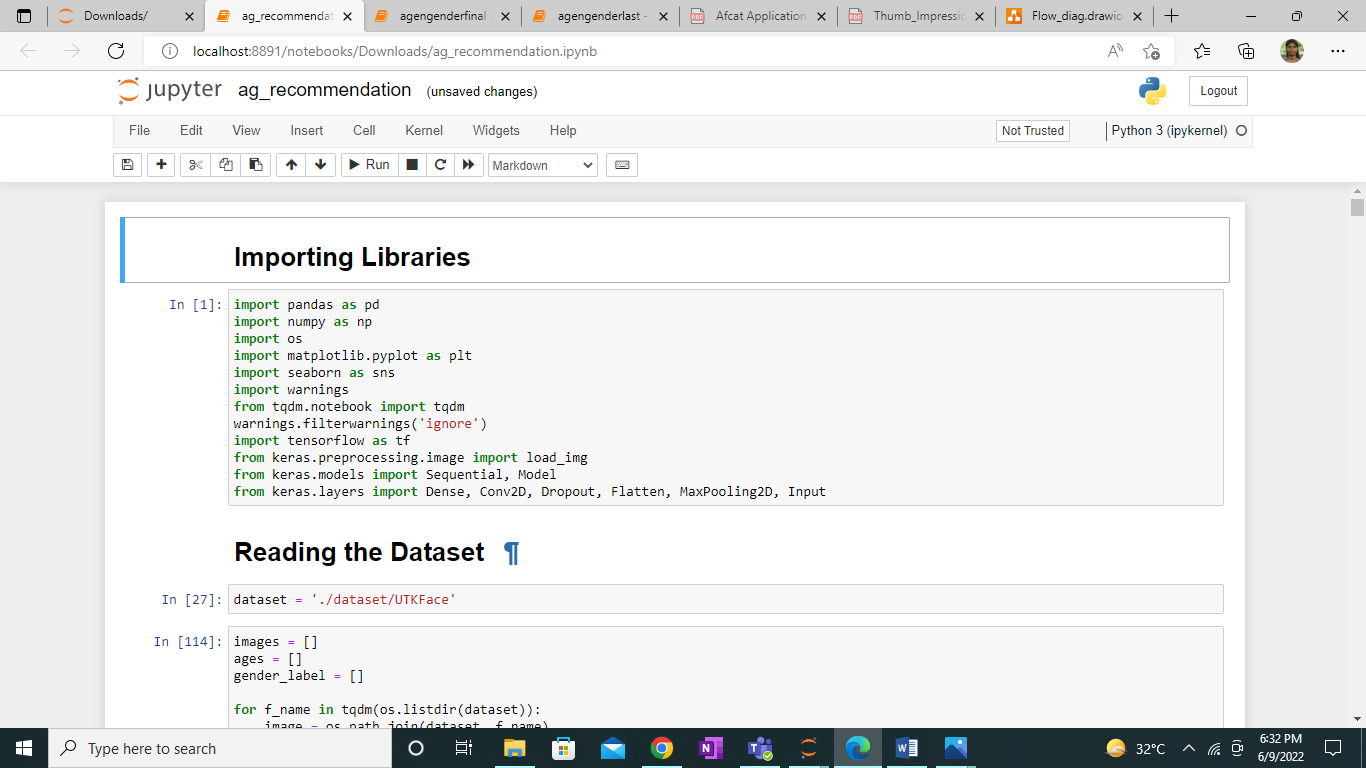
**Pandas:** for reading the dataset csv files.

**Matplotlib:** for plotting the graphs and showing images inside of the collab console along with seaborn.

**Seaborn** is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

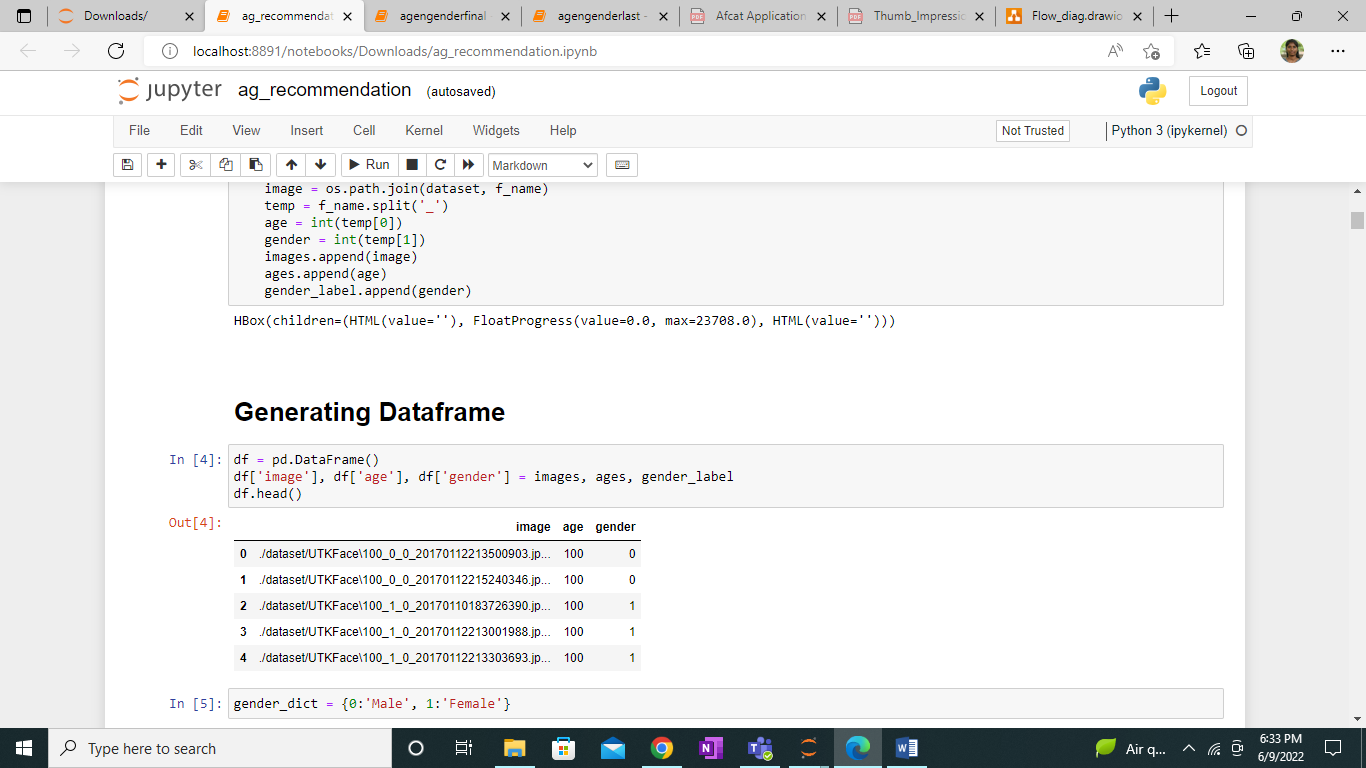
**tqdm.notebook:** TQDM is a progress bar library with good support for nested loops and Jupyter / Python notebooks.

**Keras:** Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation. Keras acts as an interface for the TensorFlow library.



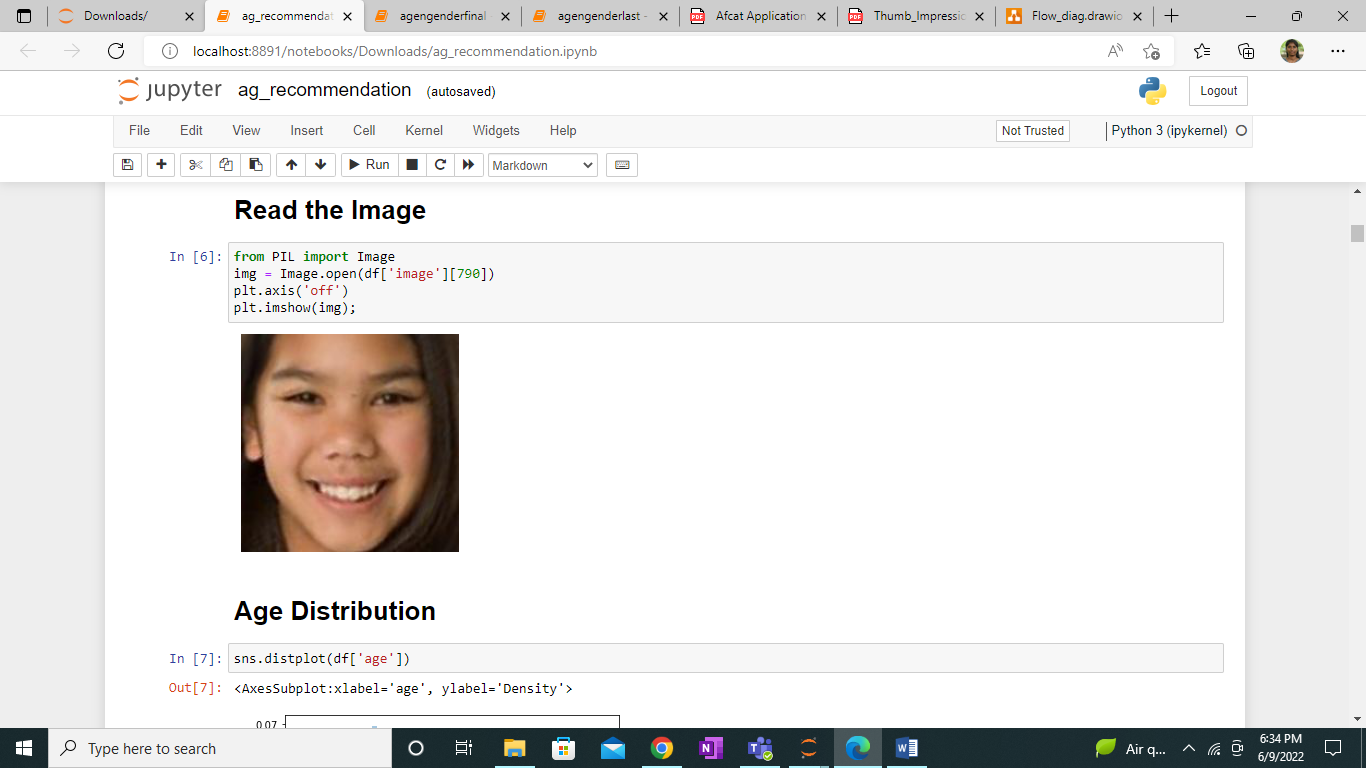
**Mapping Gender as 0 and 1:**

Map the gender to class labels 0 and 1 and print the first five records to check the integrity of the data.

****

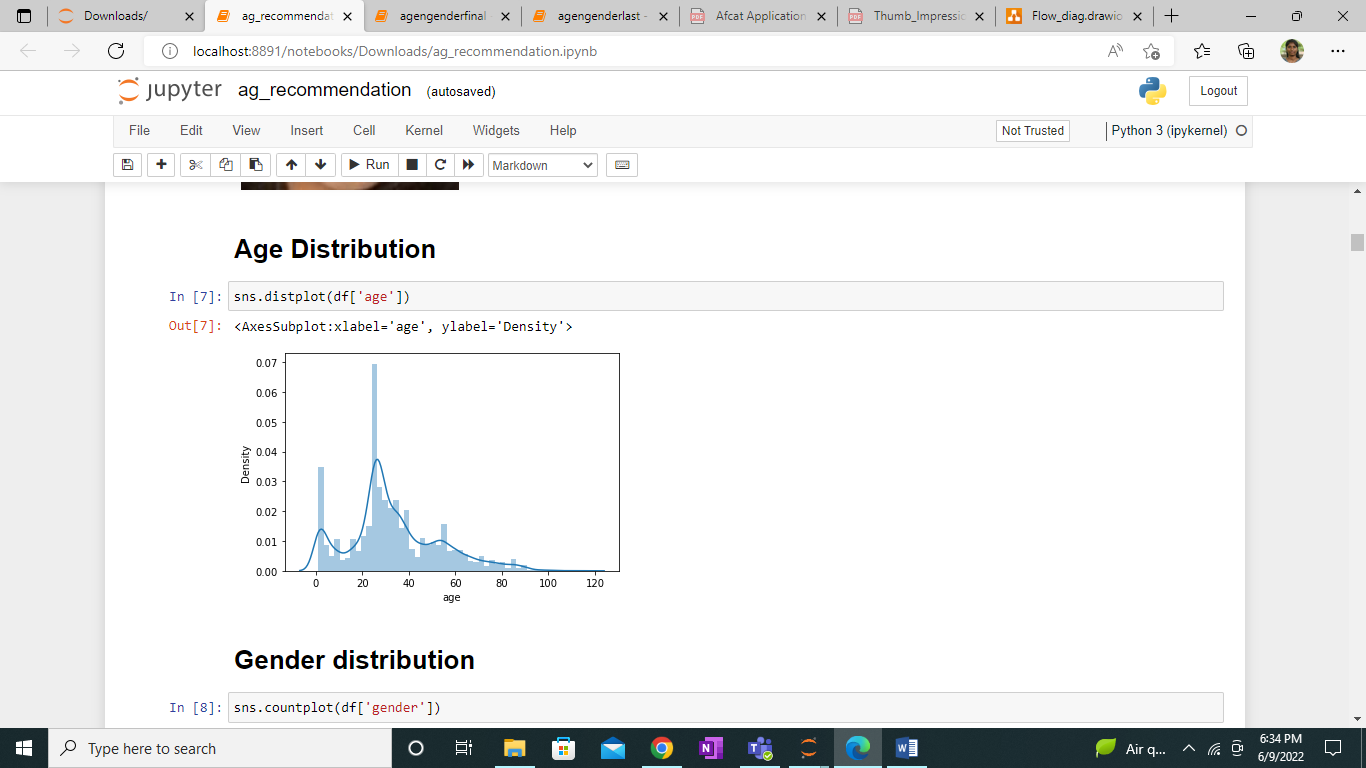
**Sample Check:**

Load the first sample data and check it. This will verify that the data structure and our project structure are correct.

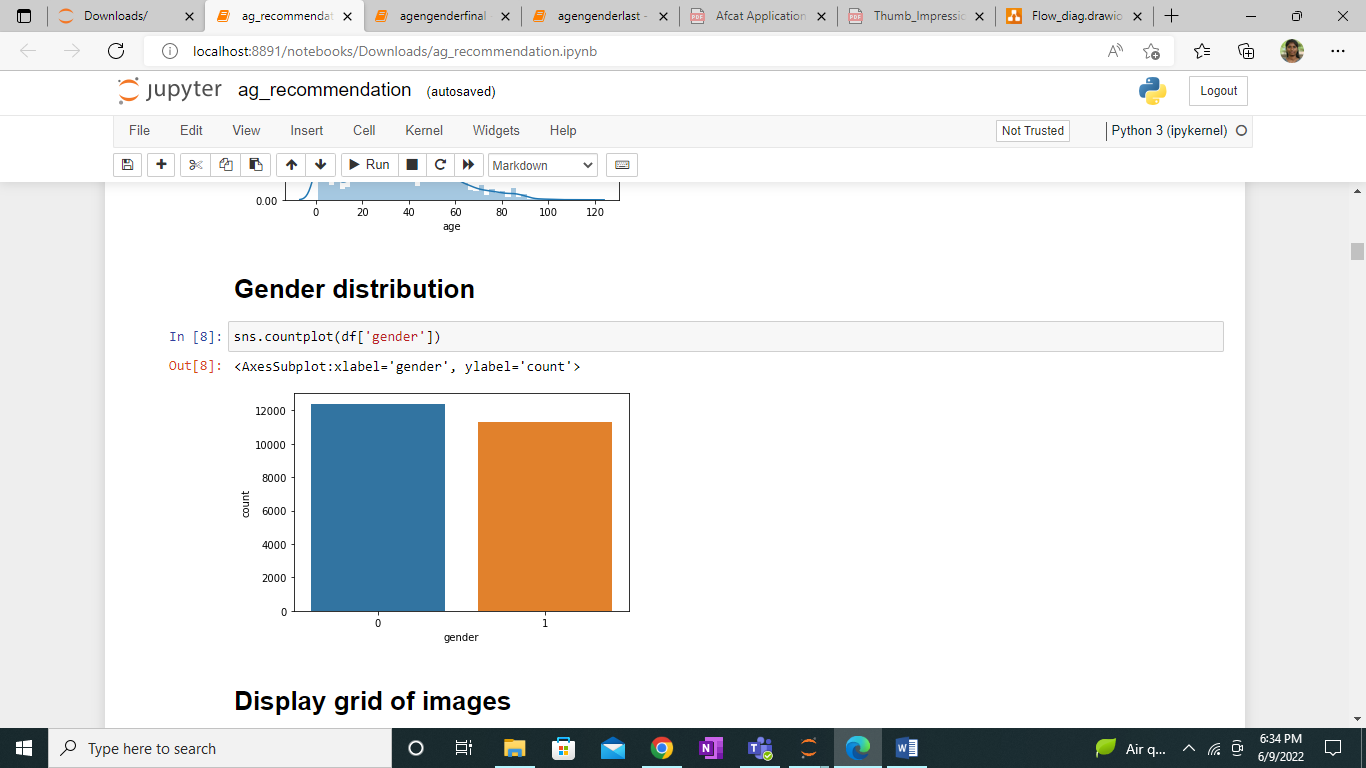
****

**Visualization**

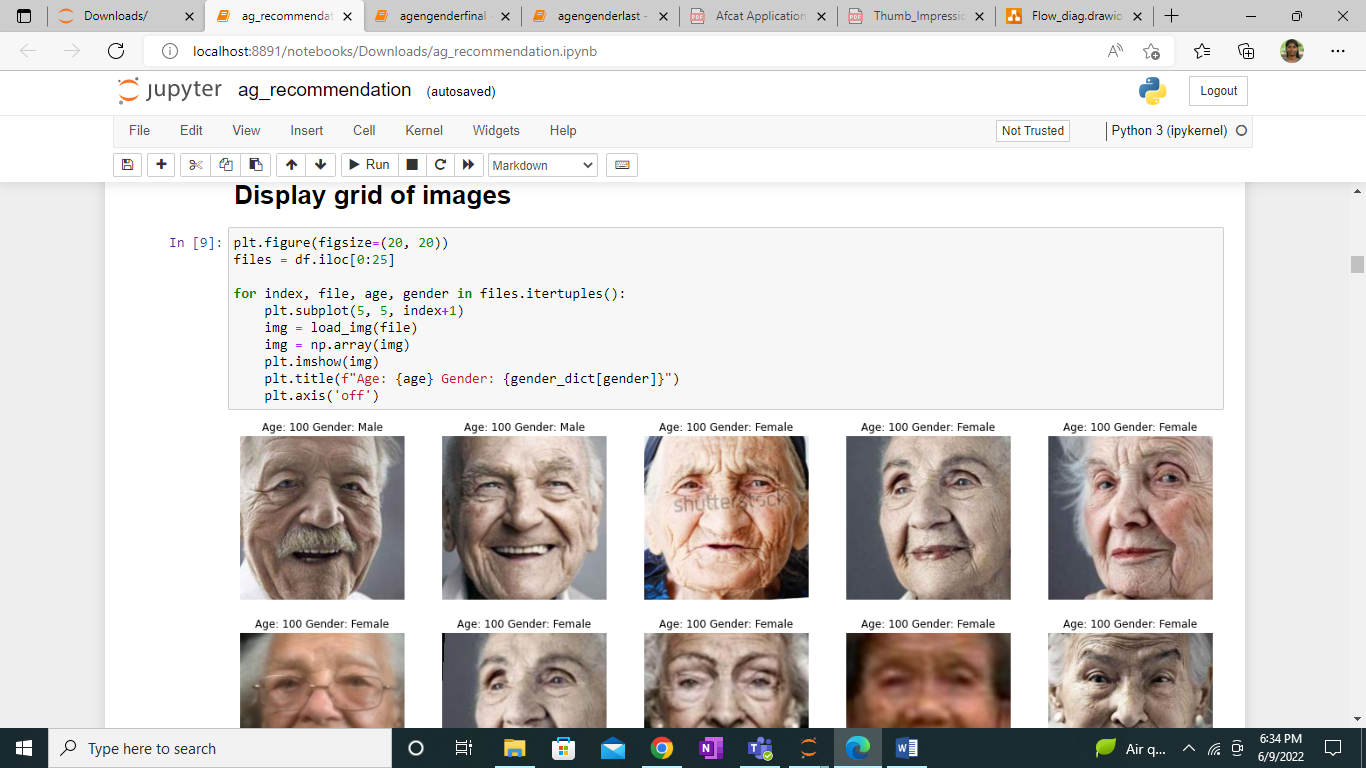
Plotting a bar graph for age values. This will visualize the variance in data as well as overview of what the age data holds.

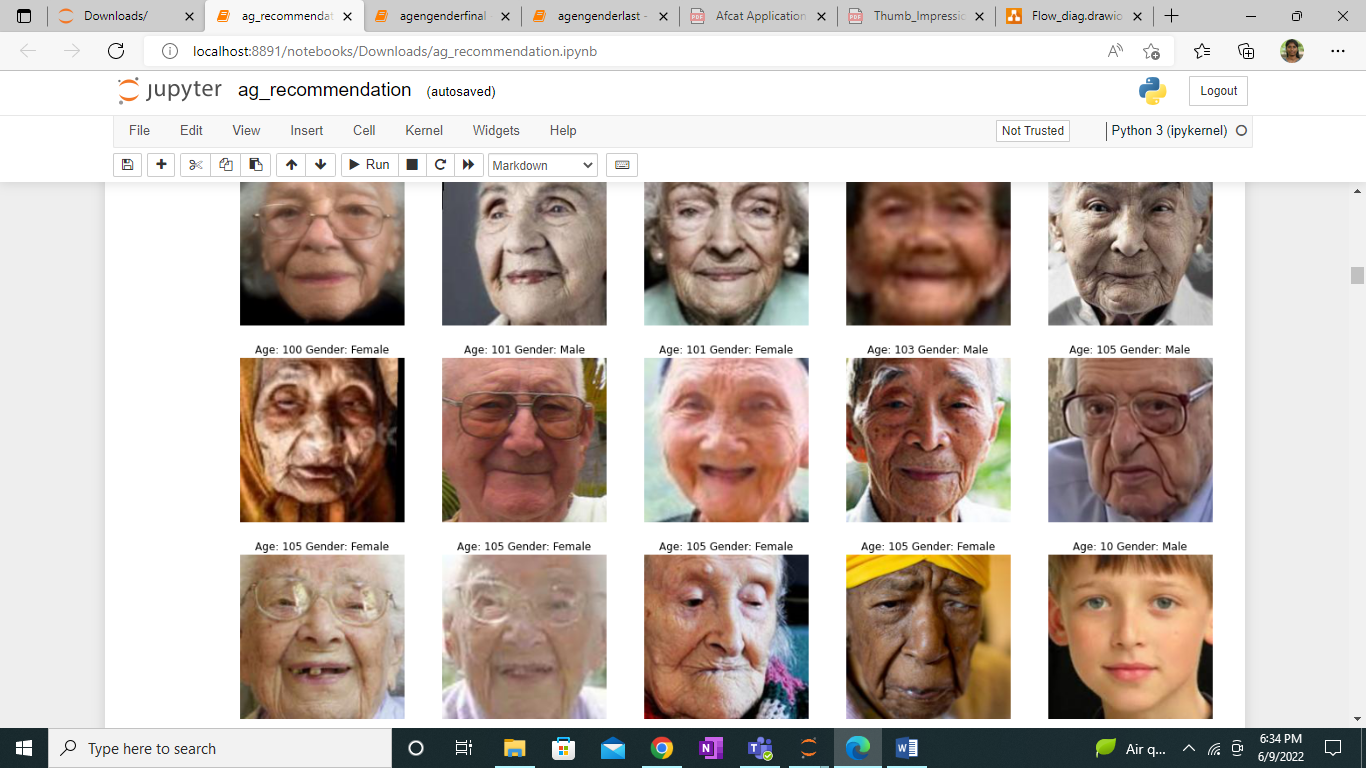


Plotting a bar graph for gender values. This will visualize the variance in data as well as overview of what the gender data holds.



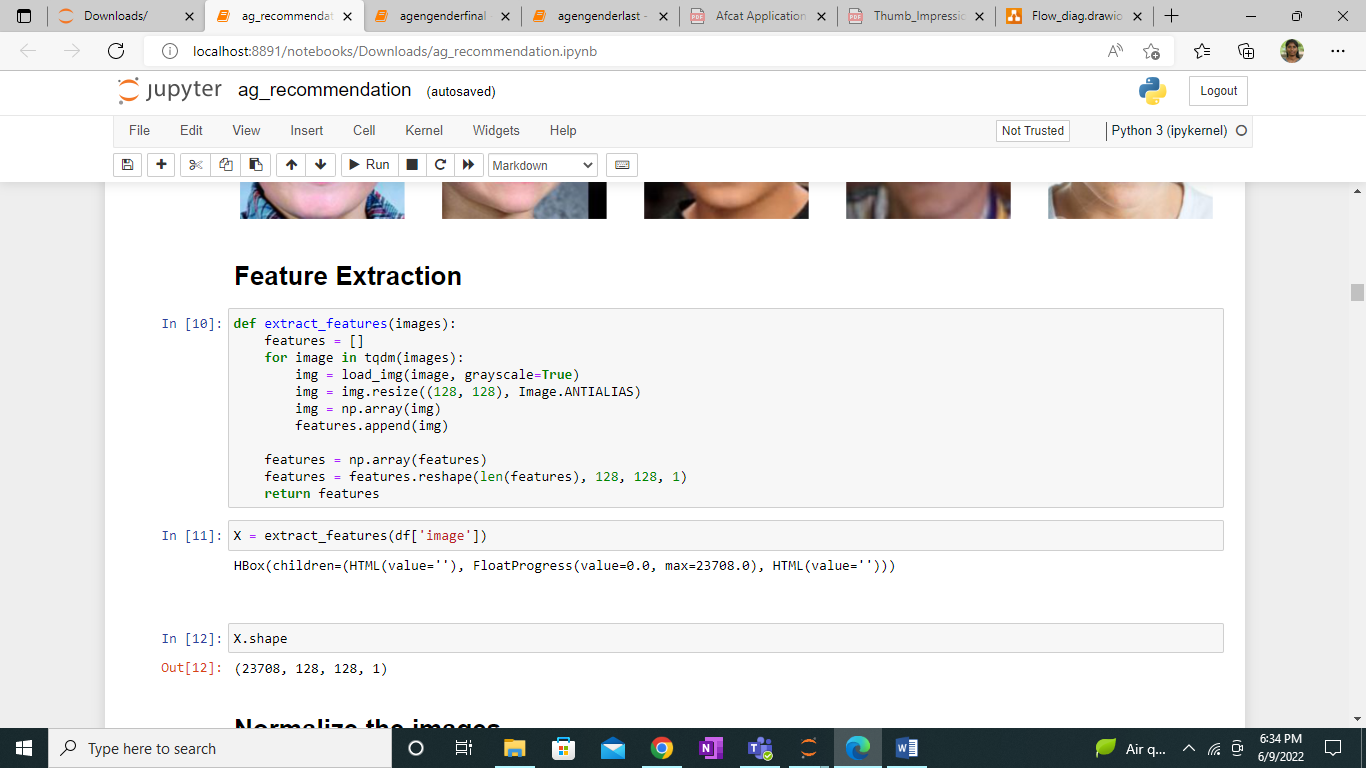
**Display Grid of Images**





**Feature Extraction:**

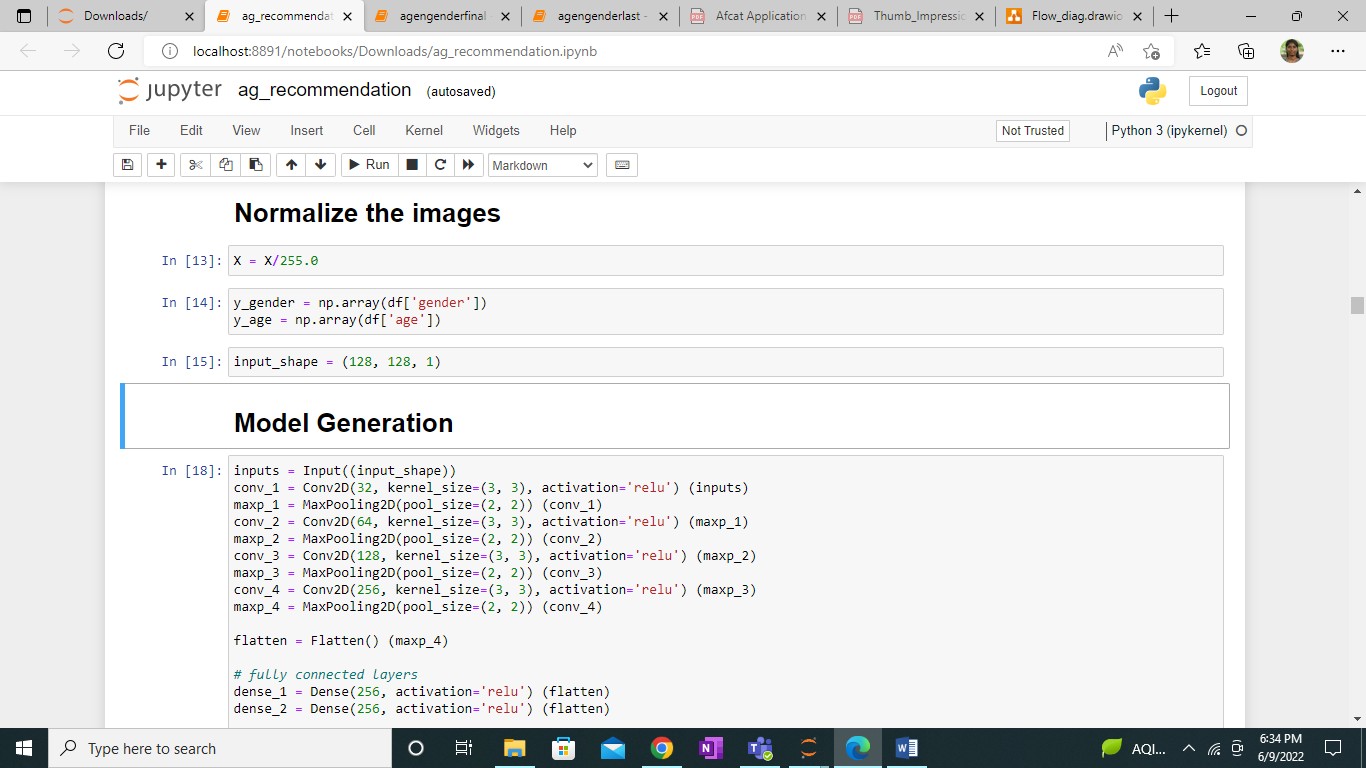
Then features are extracted from the images and shape of the dataset is found. It is the process which convert an input image into a range of pixel values (0 and 255) ie, the dark or unclear images become more clearer.

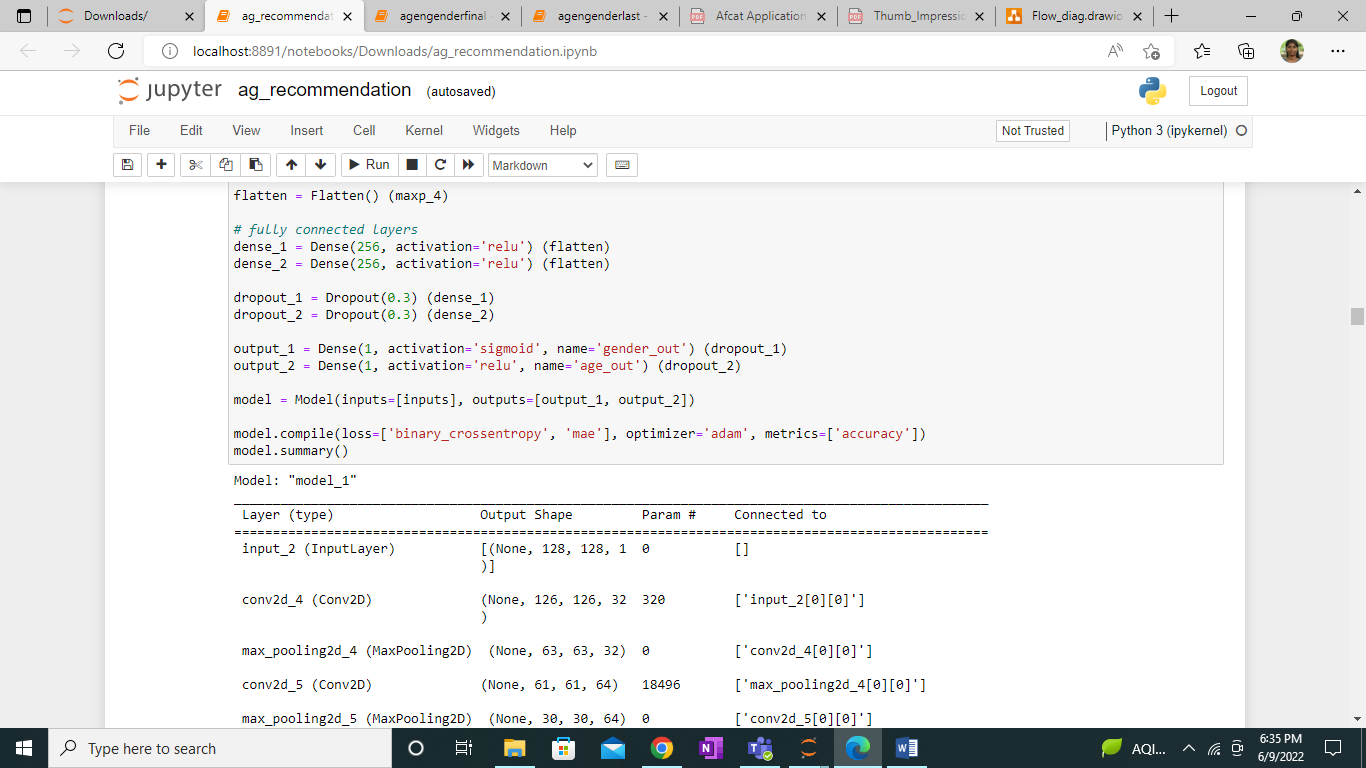


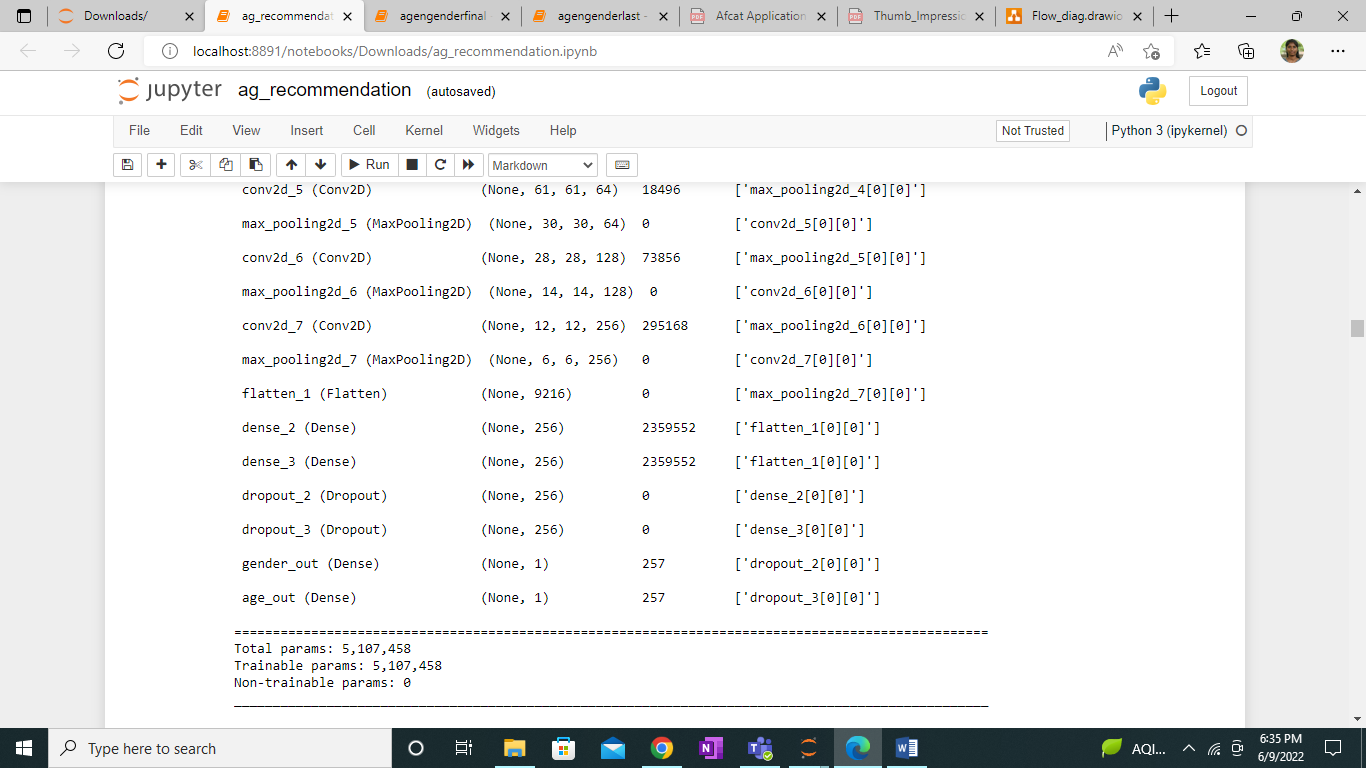
**MODEL GENERATION:**

We define the CNN model here, we are using a sequential model i.e. data will travel sequentially from one layer to another without any jumps and splitting.

We use convolutional neural network for age and gender model generation. Our CNN consists of 4 convolutional layers, 4 max pooling layers and 2 fully connected layers. For this model a facial image is passed as input with the shaped (128,128,1). So the input is passed through these convolutional, max pooling and fully connected layers. Finally we get the output which is age and gender.

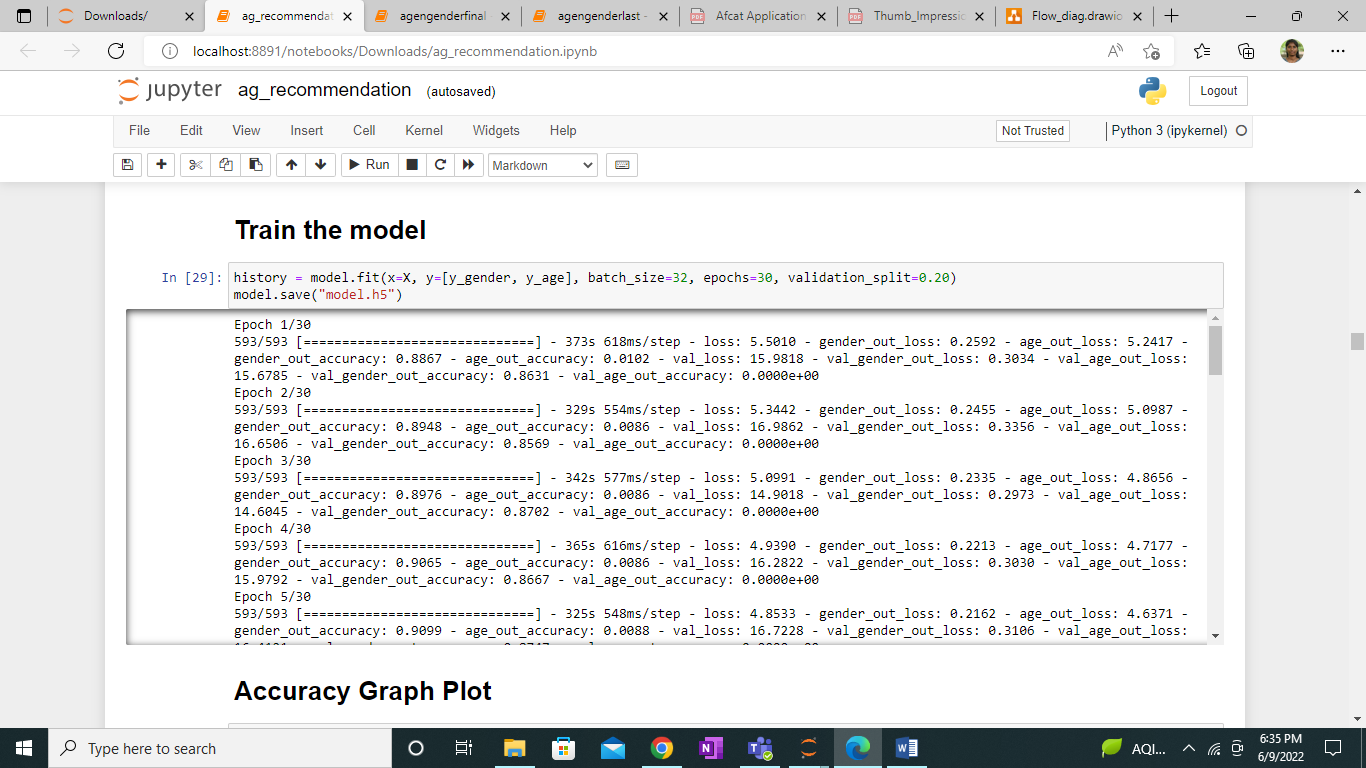






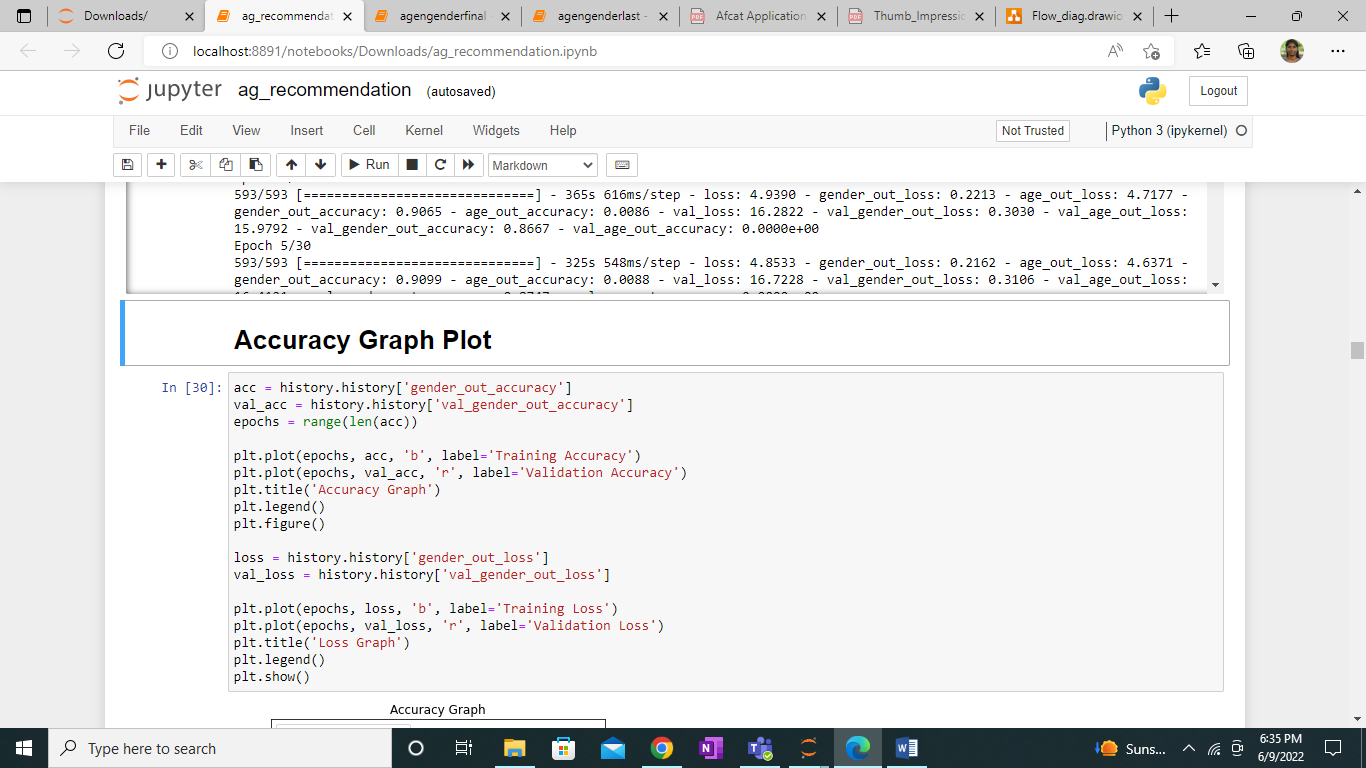
#### **Train the model**

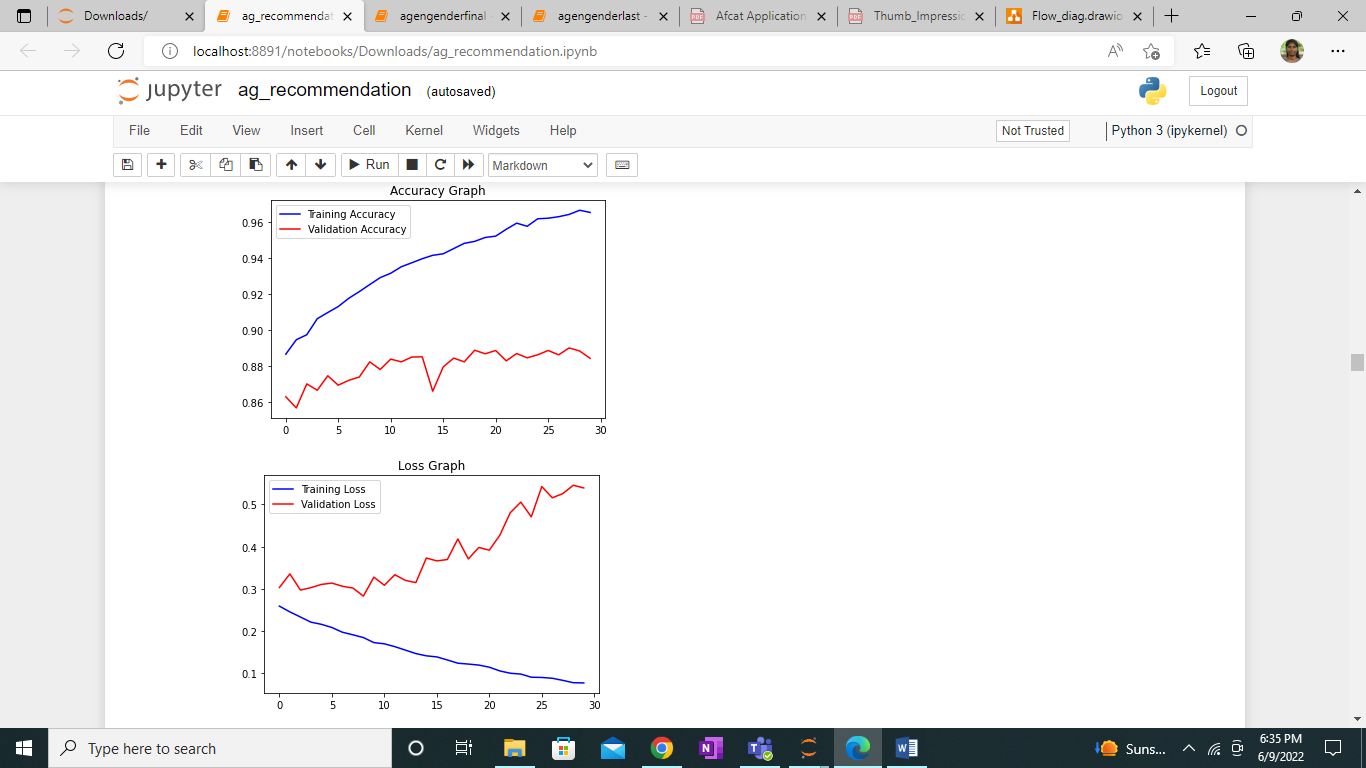
The generated model has been saved and implemented here to train with various items from the dataset.

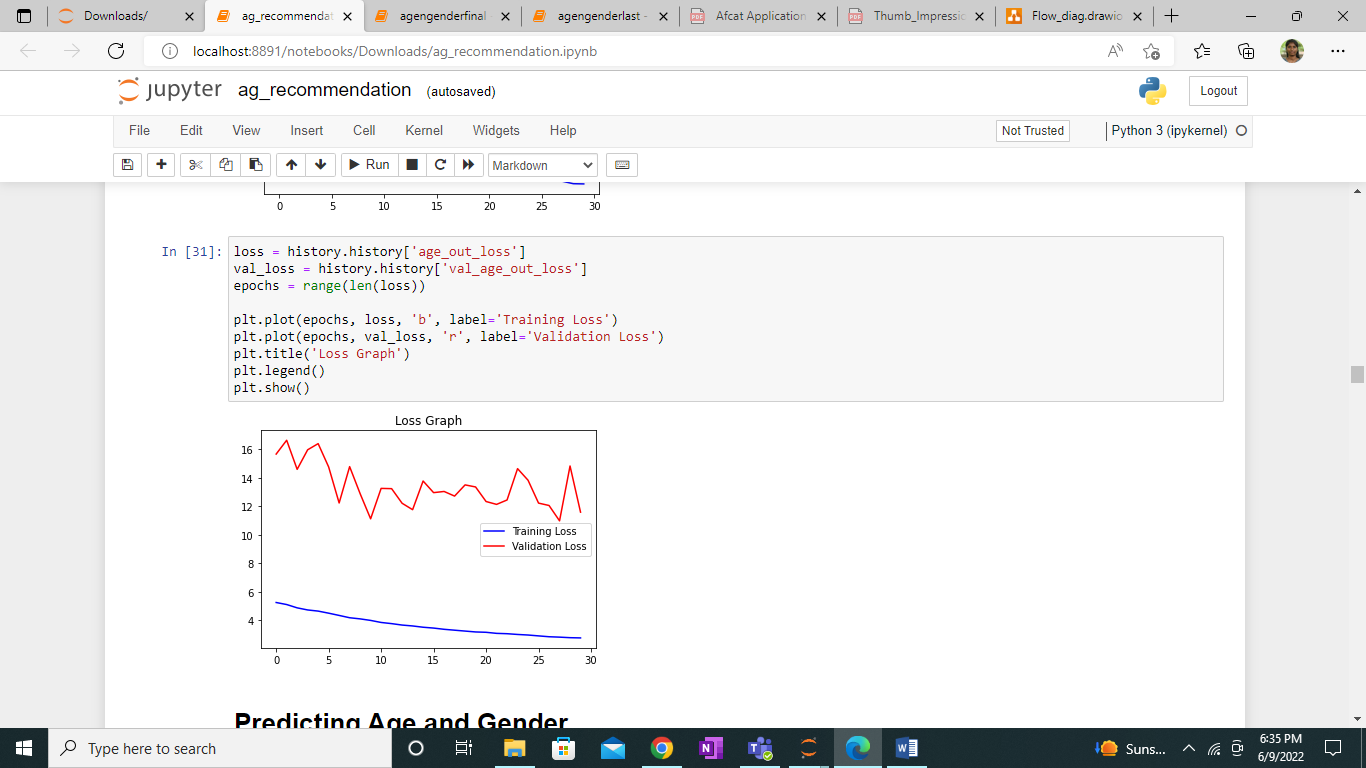


#### **Graph Plot:**

Accuracy graph and loss graph are plotted between training and validation.

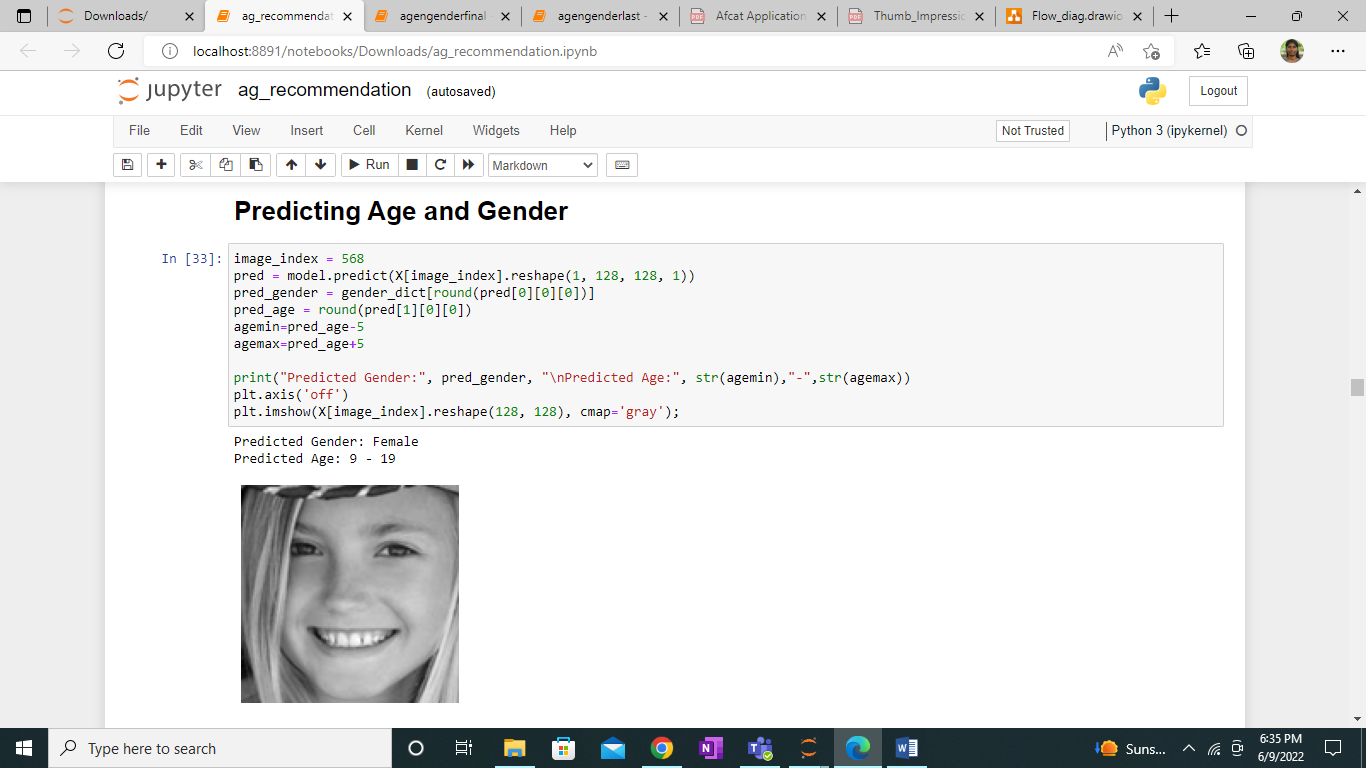


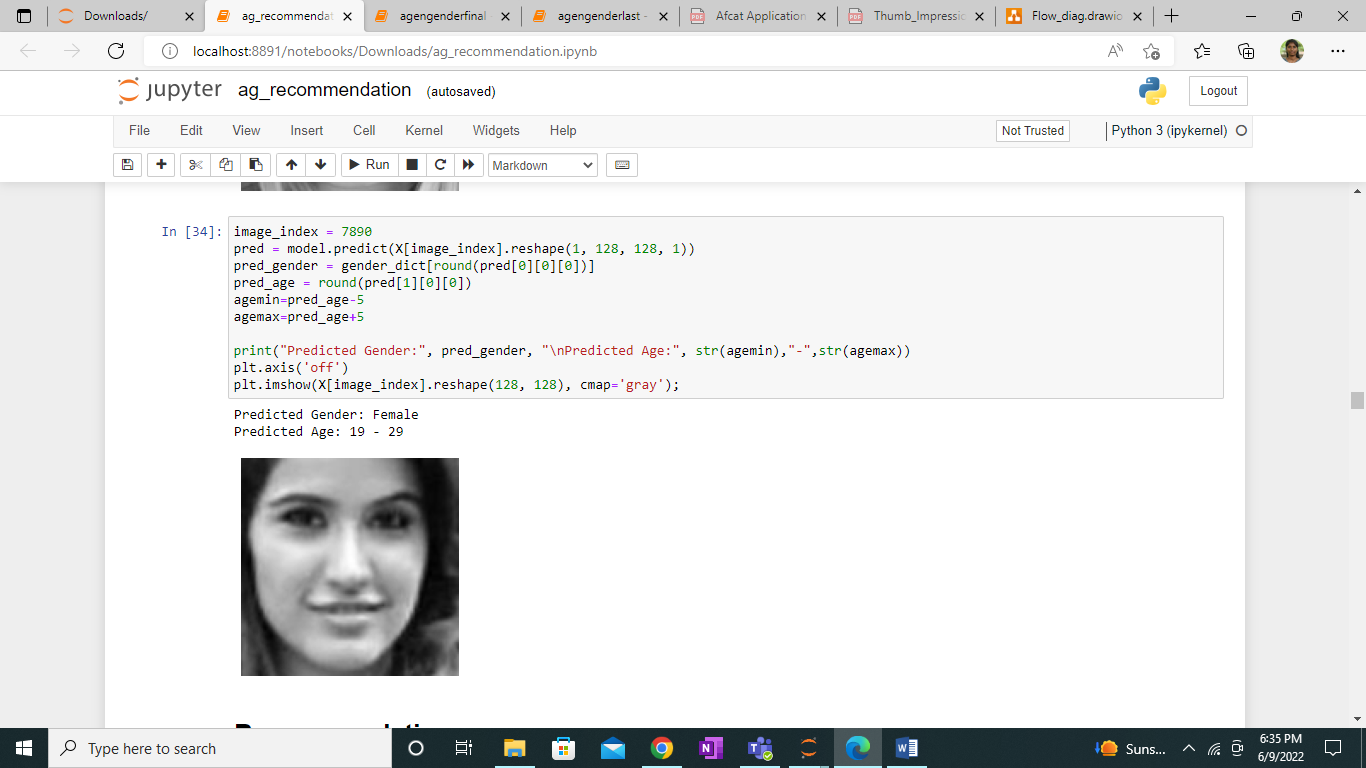




#### **Prediction of Age and Gender:**

Sample prediction of age and gender.

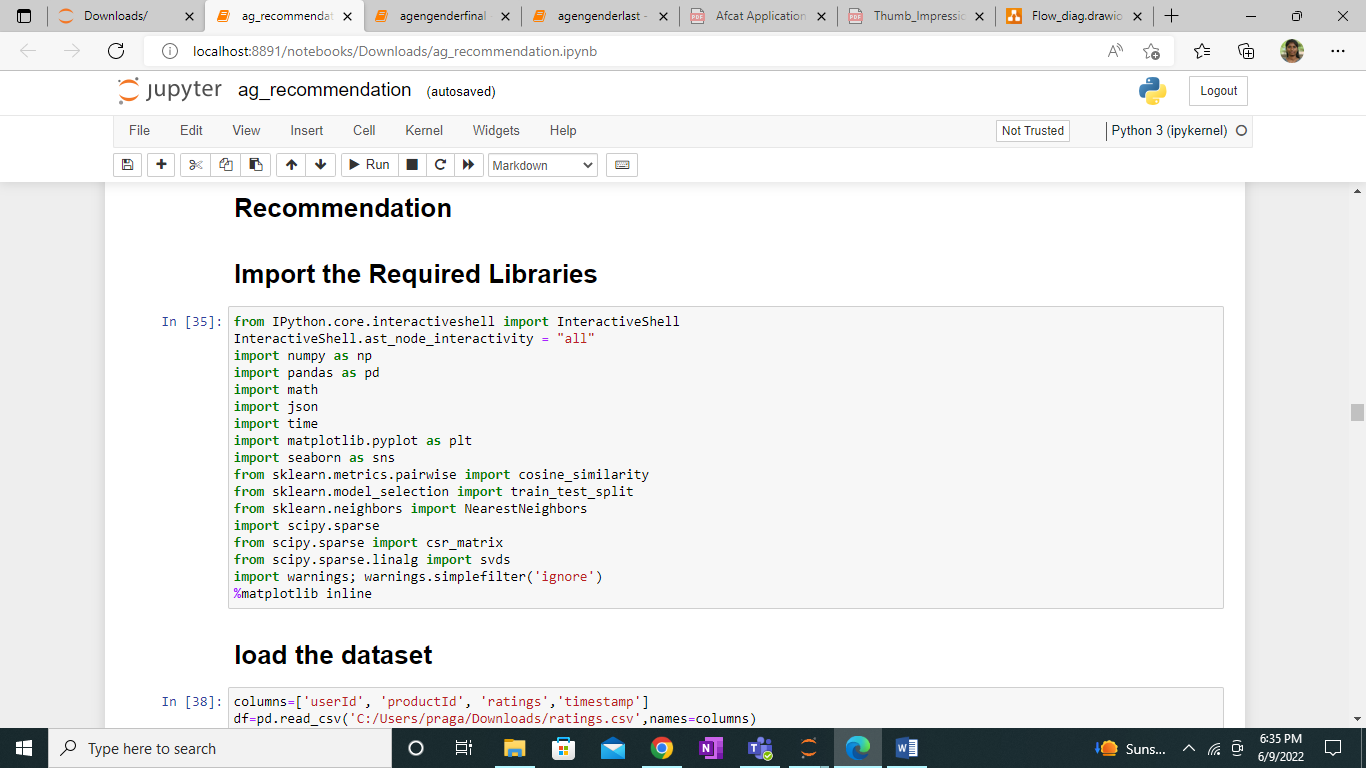




**RECOMMENDATION PART:**

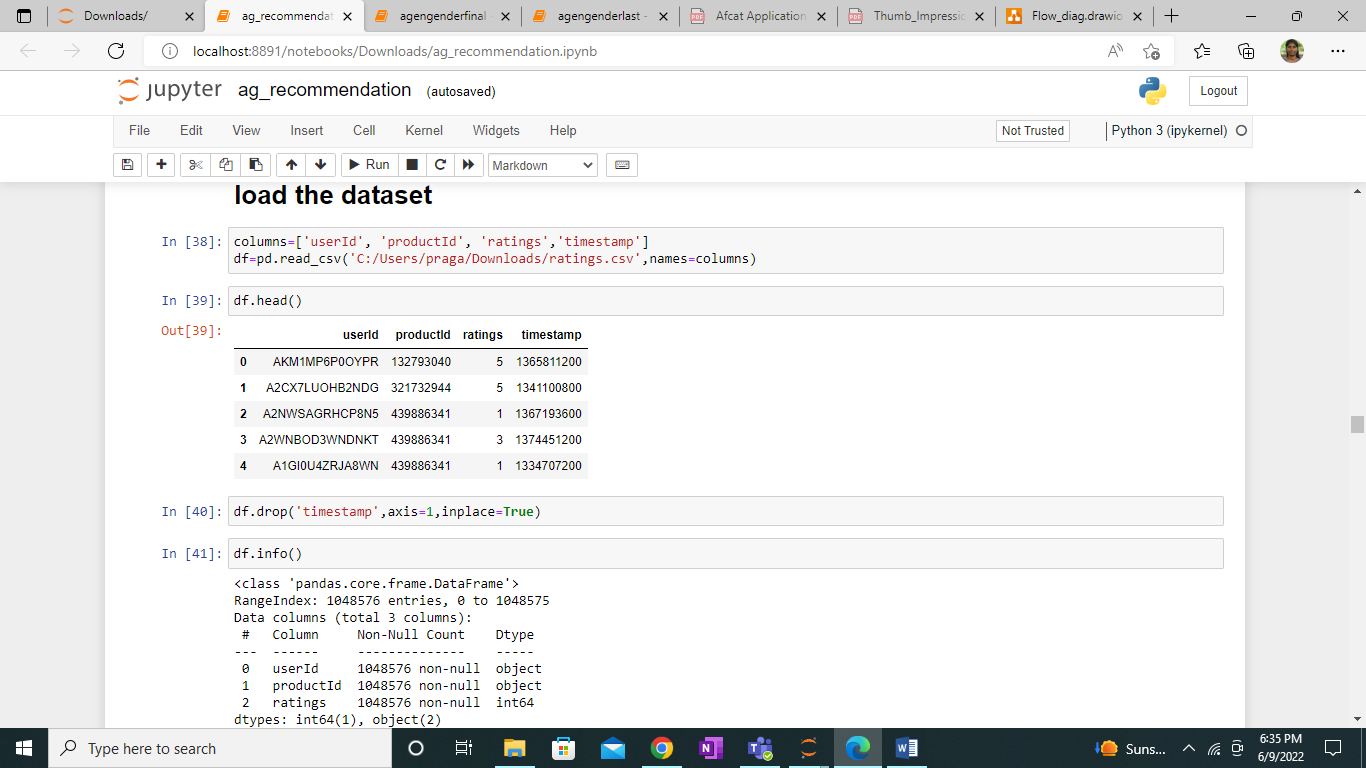
**Importing the libraries:**

Firstly the necessary libraries like numpy, pandas, matplot, seaborn, mat, json, time are imported.

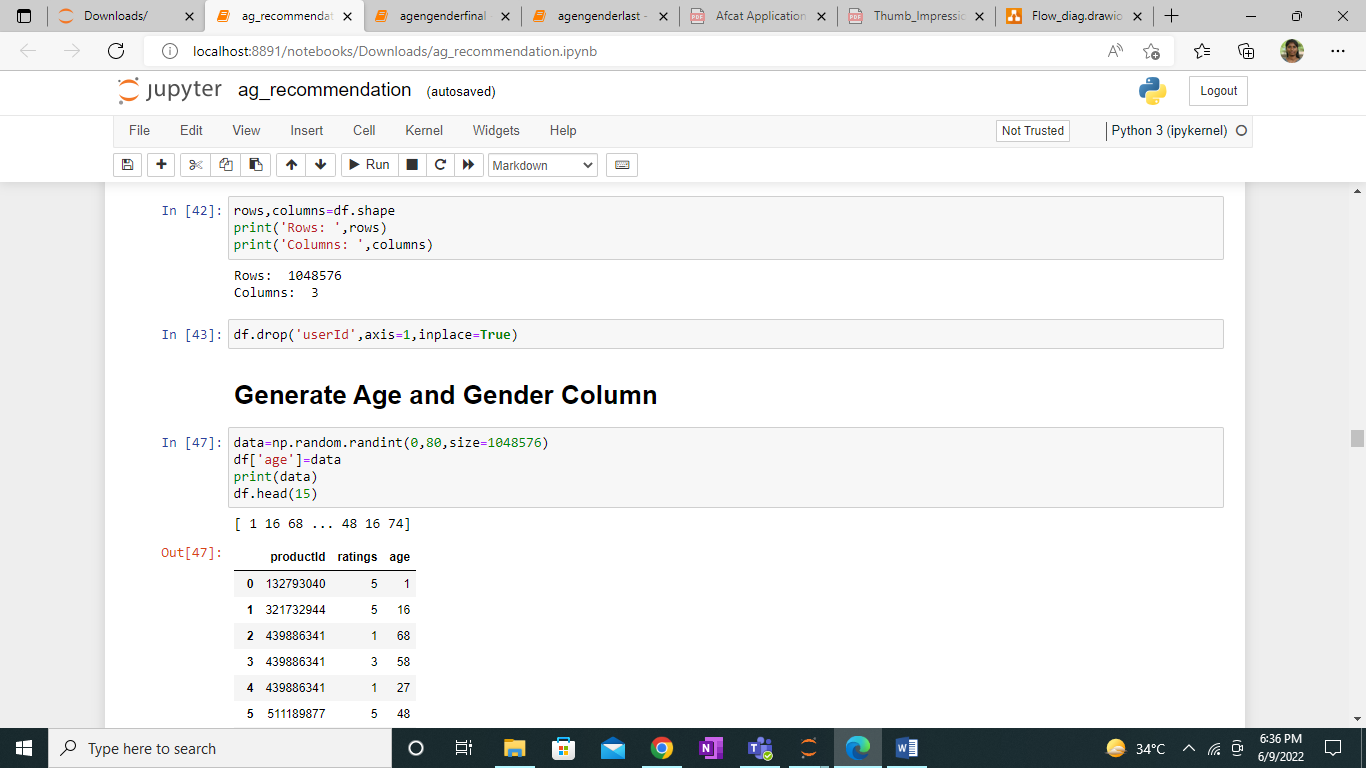


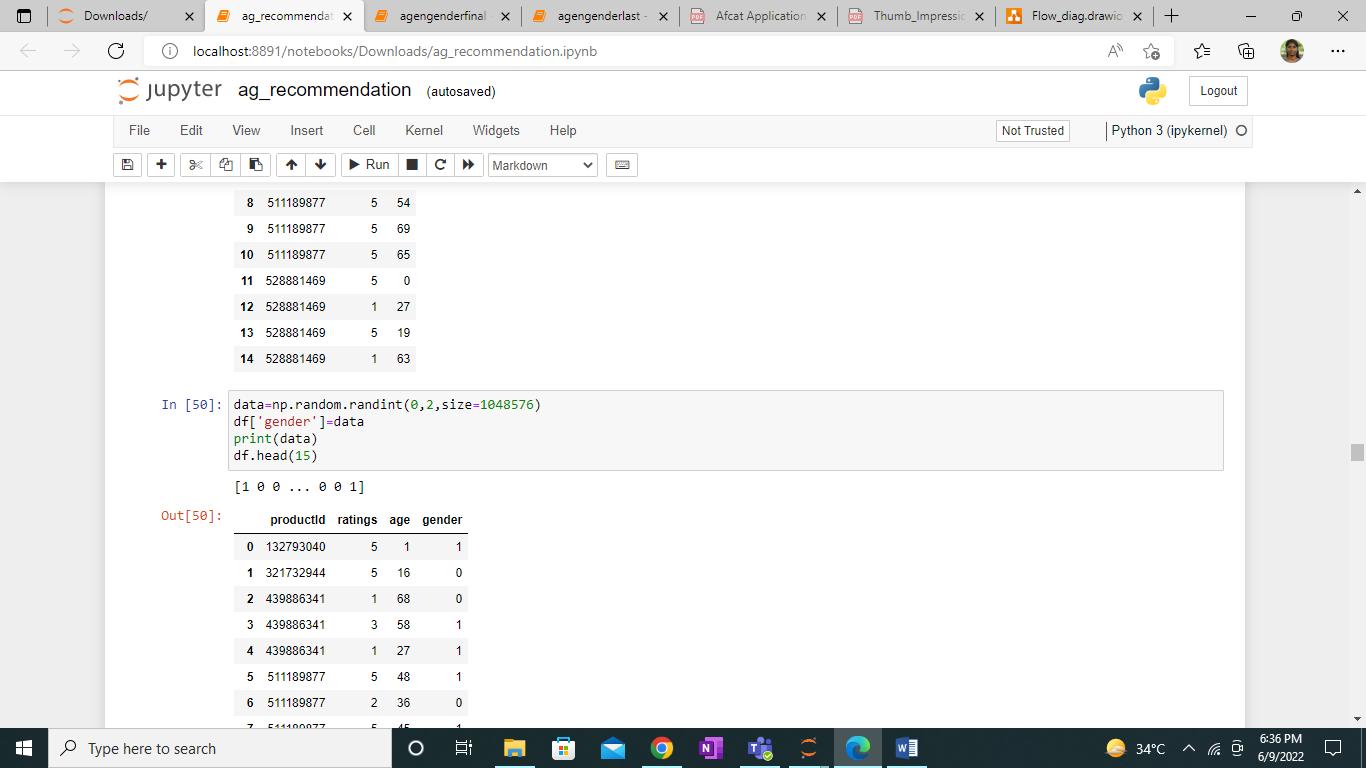
#### **Load the Dataset:**

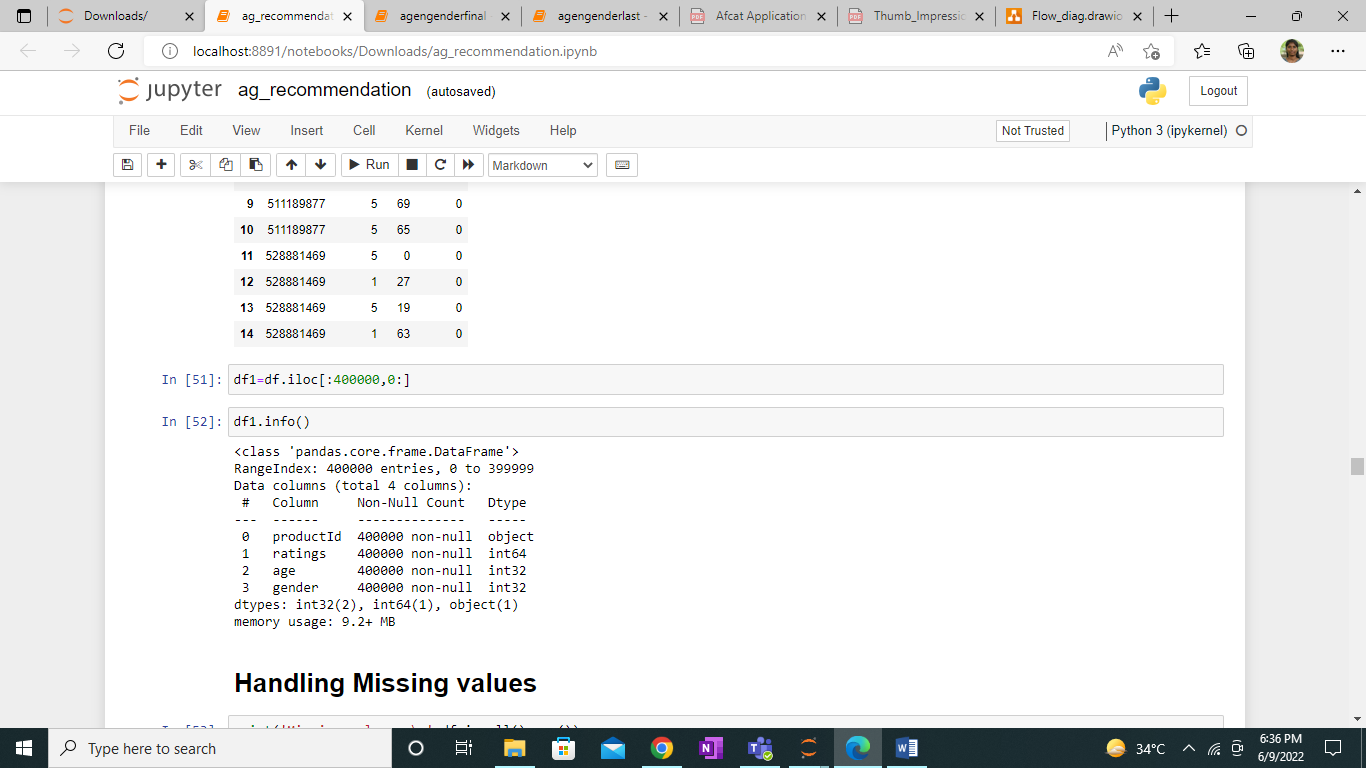
For recommendation system we have chosen amazon customer rating dataset. It consists of columns names user id, product id, ratings and timestamp. Here timestamp column is unnecessary, so we drop it. As we use the demographic details which is age and gender for recommending products we have to manually generate two columns- age and gender. The two columns- age and gender has been successfully generated using random function. This dataset consists of over 700000 rows out of which we have chosen 400000 rows for our project. The dataset consists of null missing values.



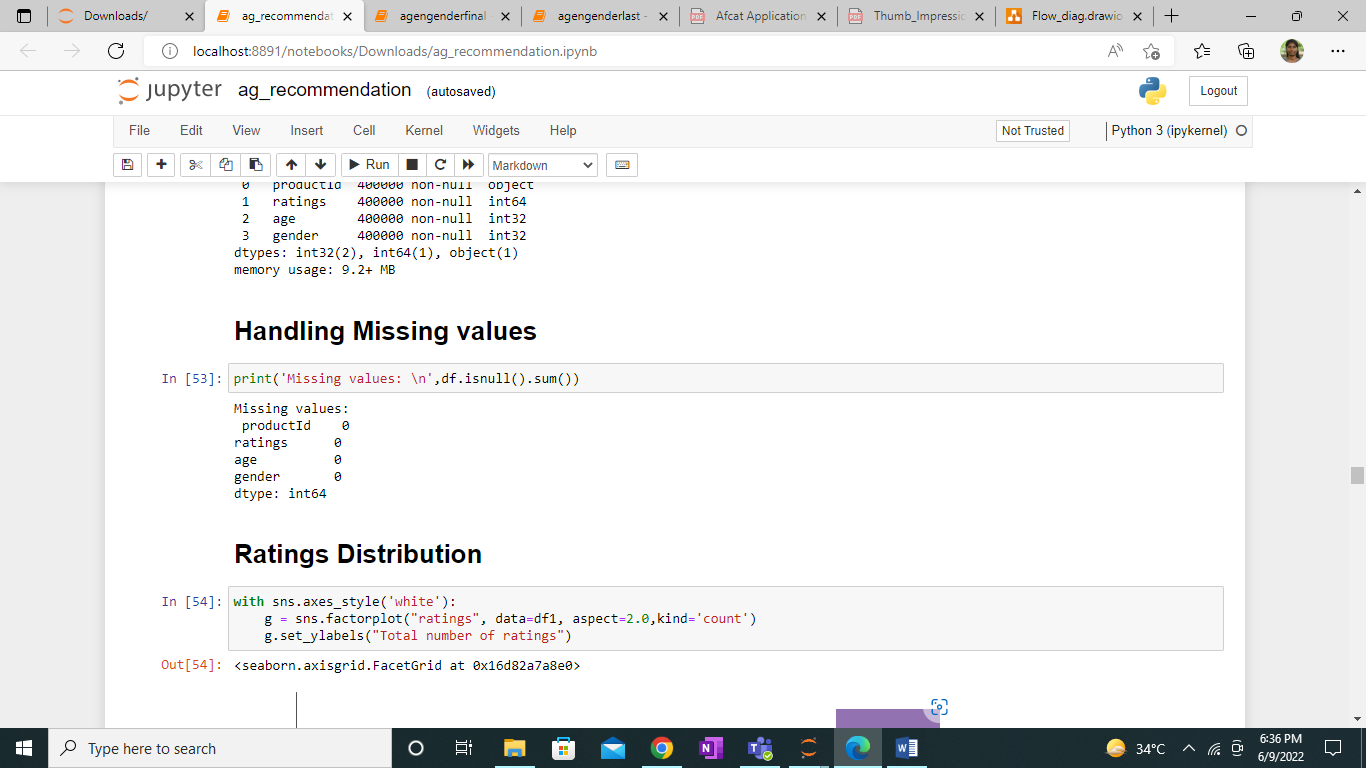
#### **Generate Age and Gender Column:**





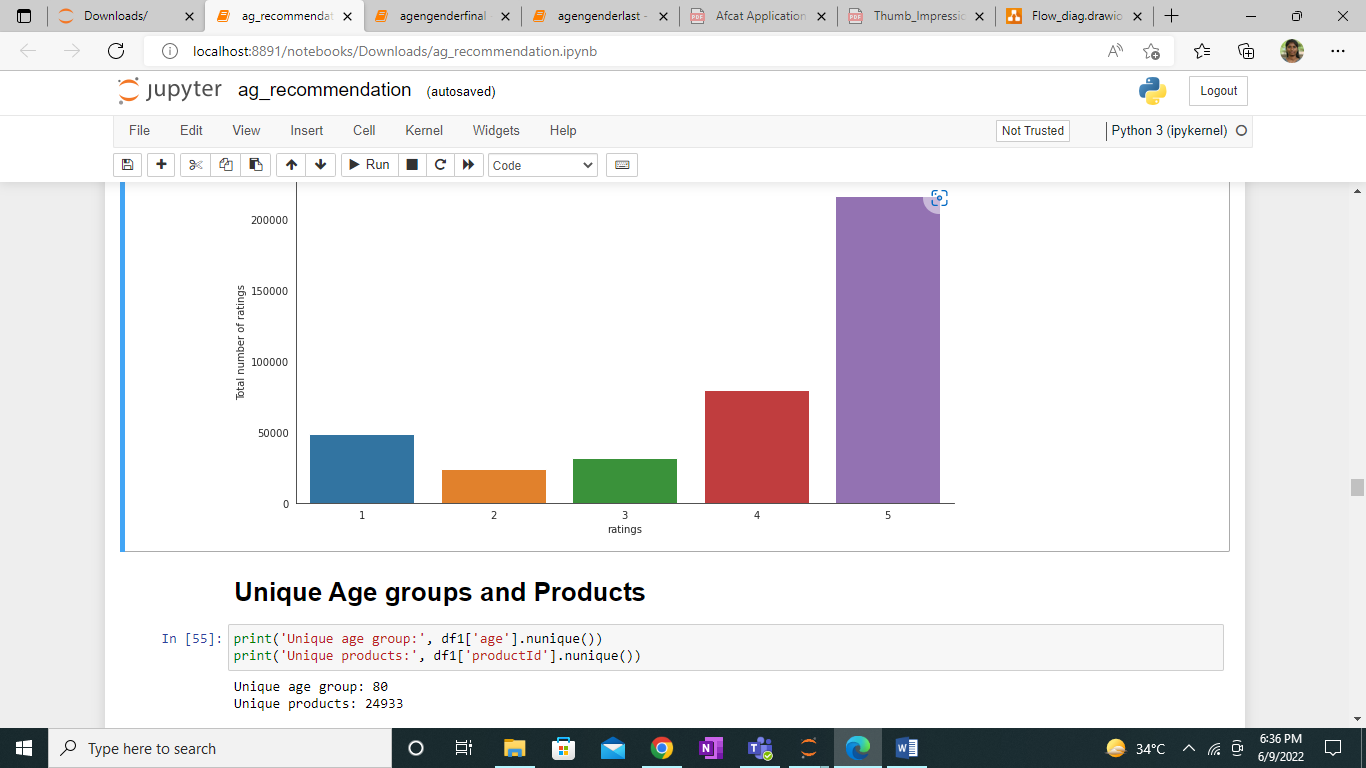


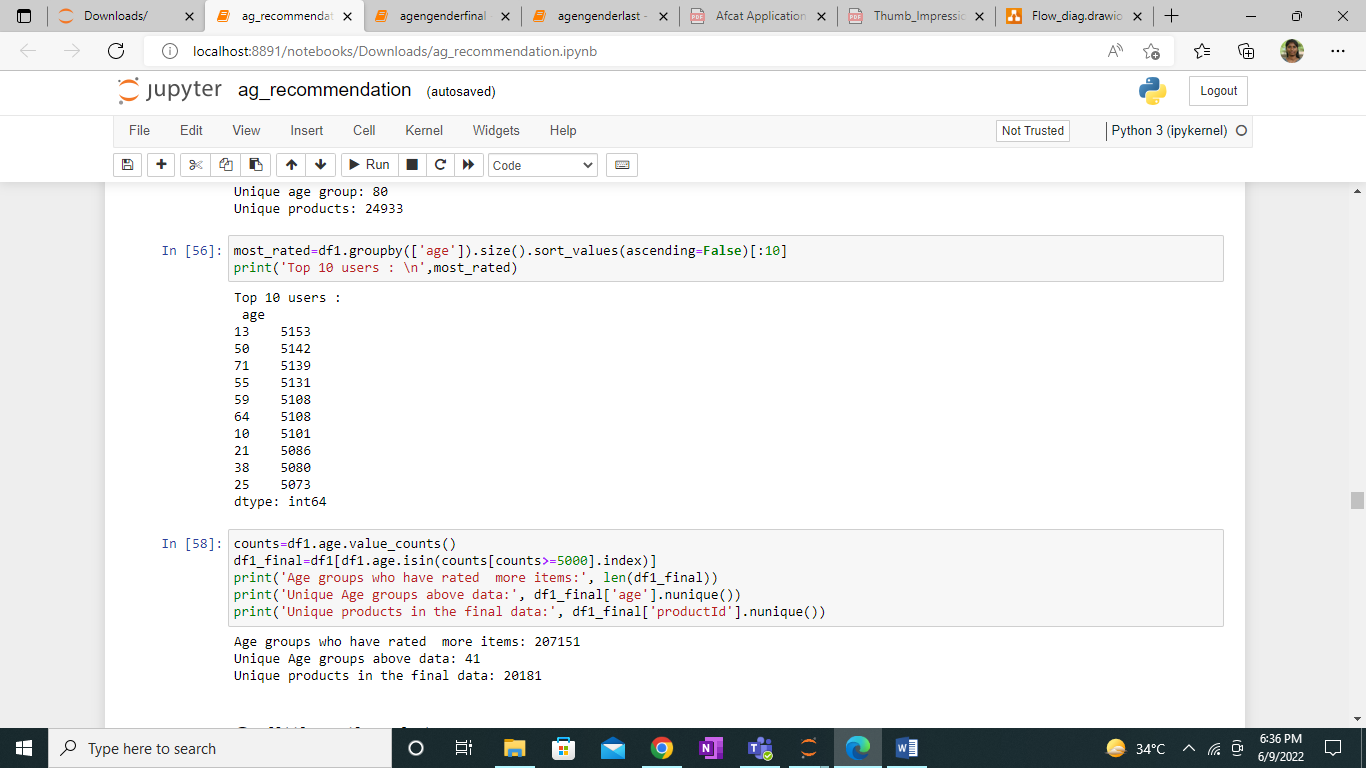
#### **Handling Missing Values:**



#### **Ratings Distribution:**

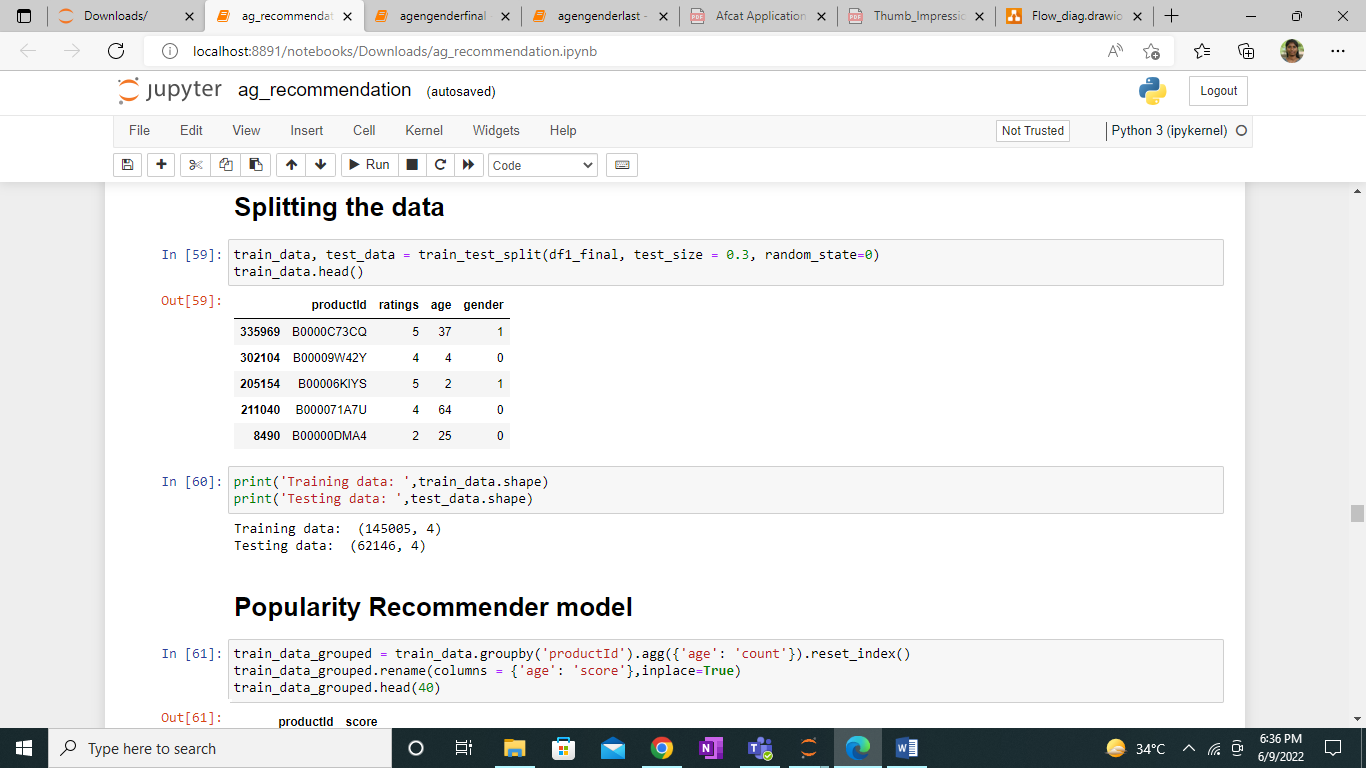
Distribution of the ratings from 1 to 5 is found and unique groups of age and unique number of products are calculated.



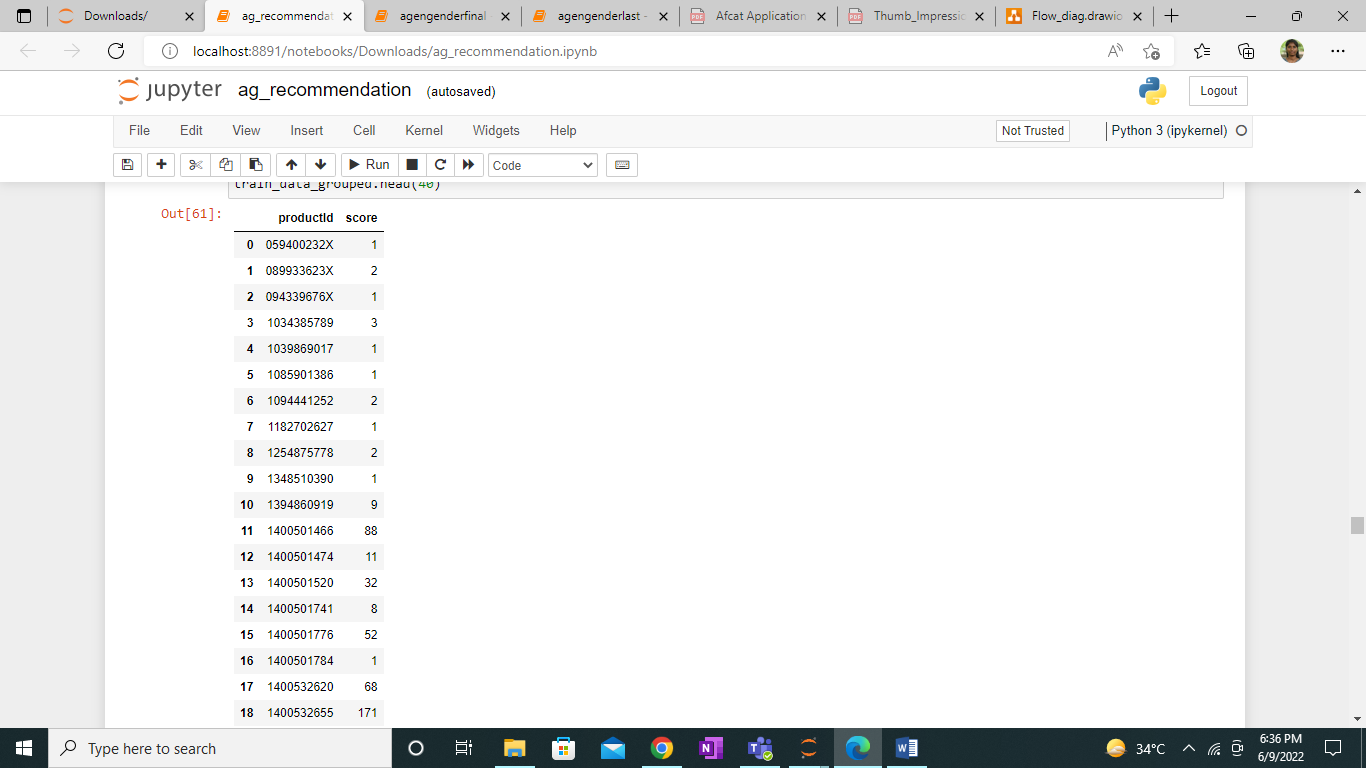


#### **Splitting Data:**

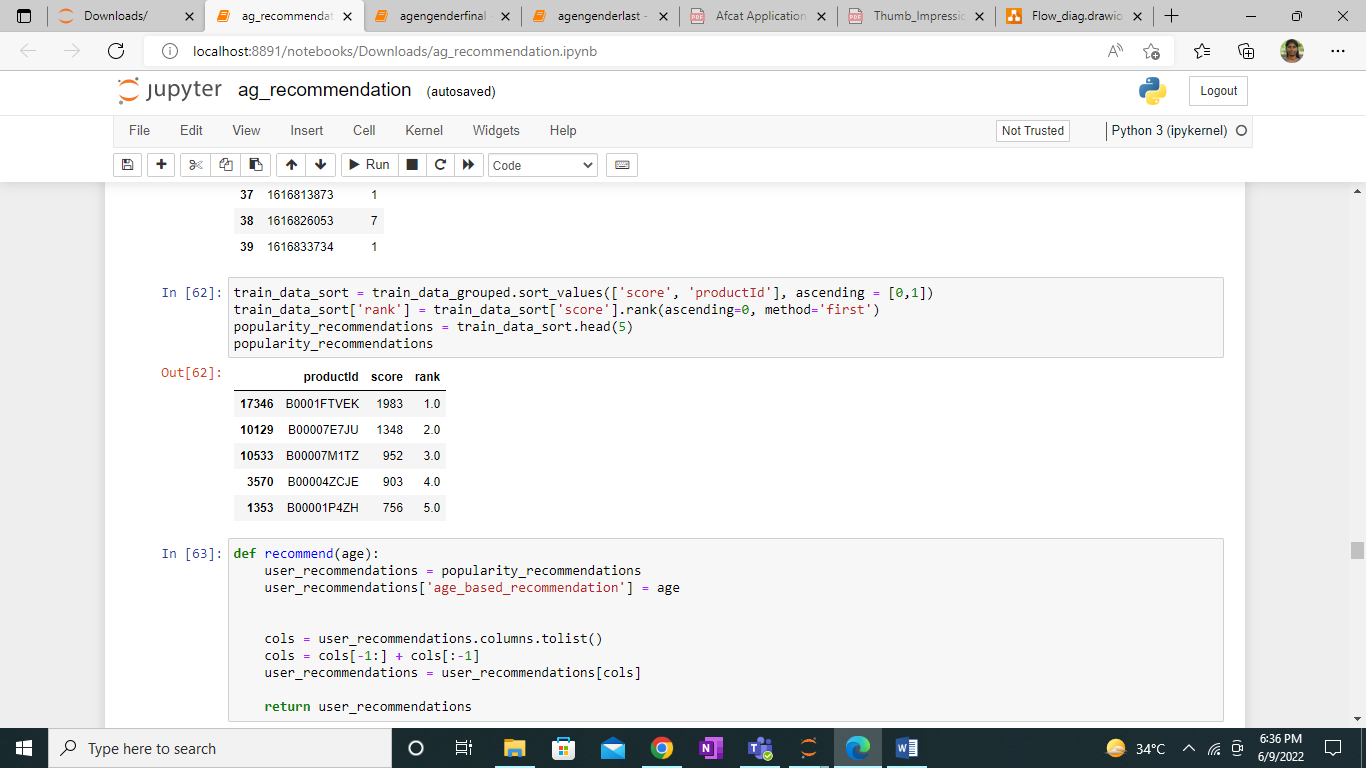
Splitting the data into train and test data.



Based on the popularity recommender model, most sought products are given a score.



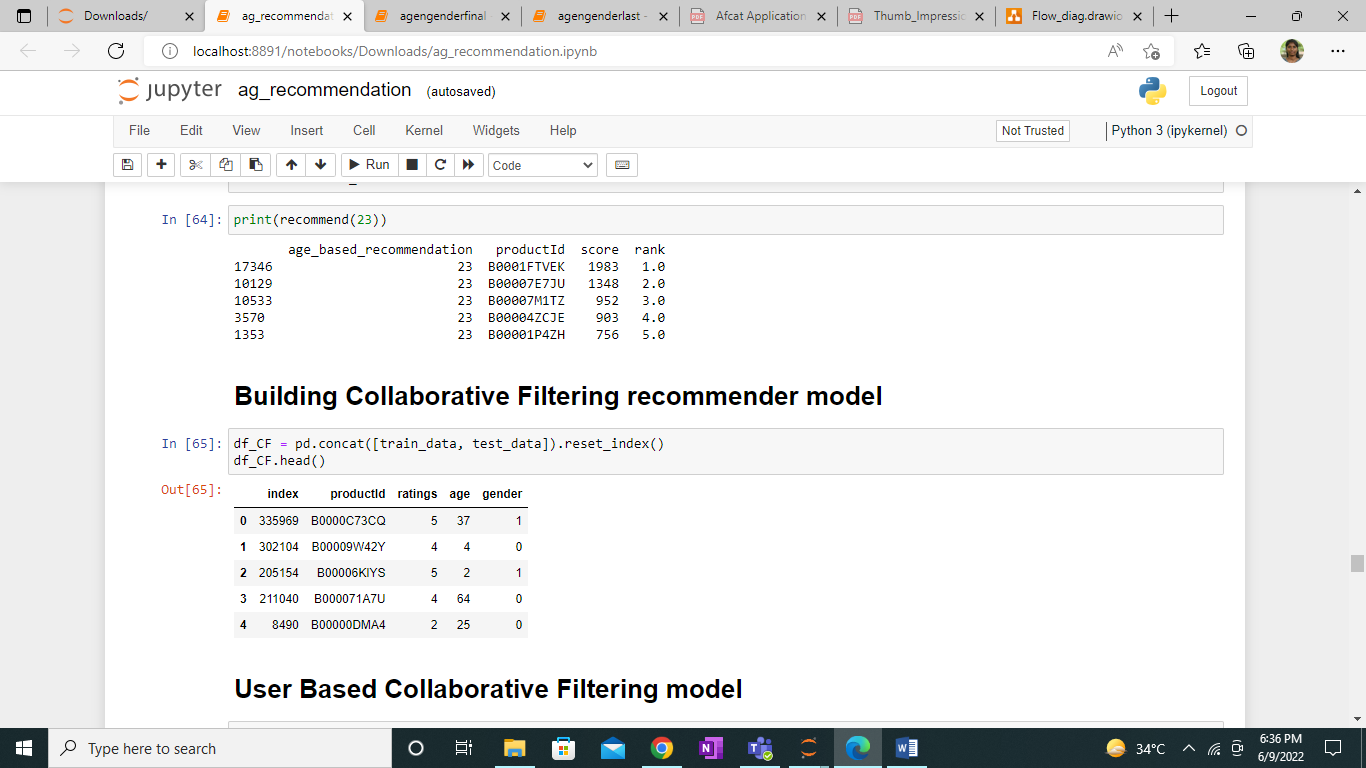
Training data is sorted based on score.



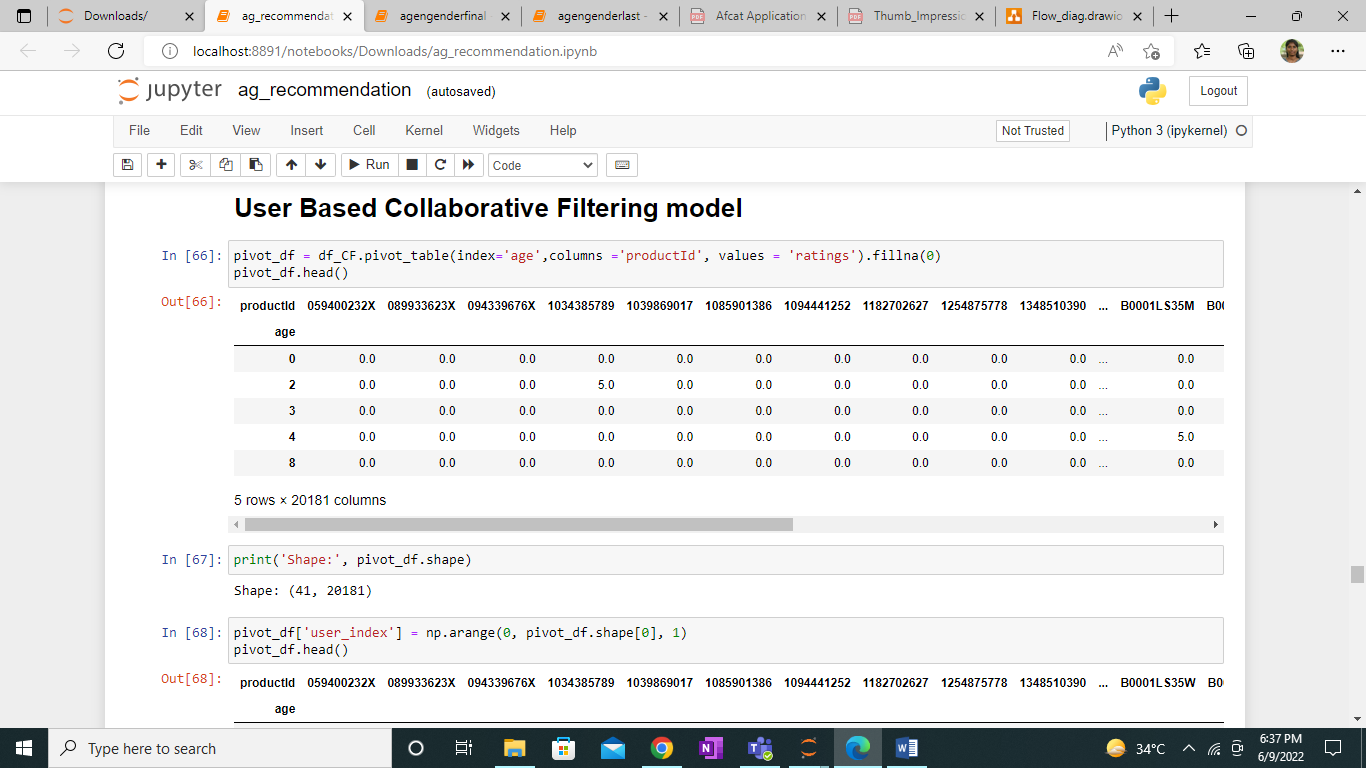
#### **Collaborative Filtering Recommender Model:**

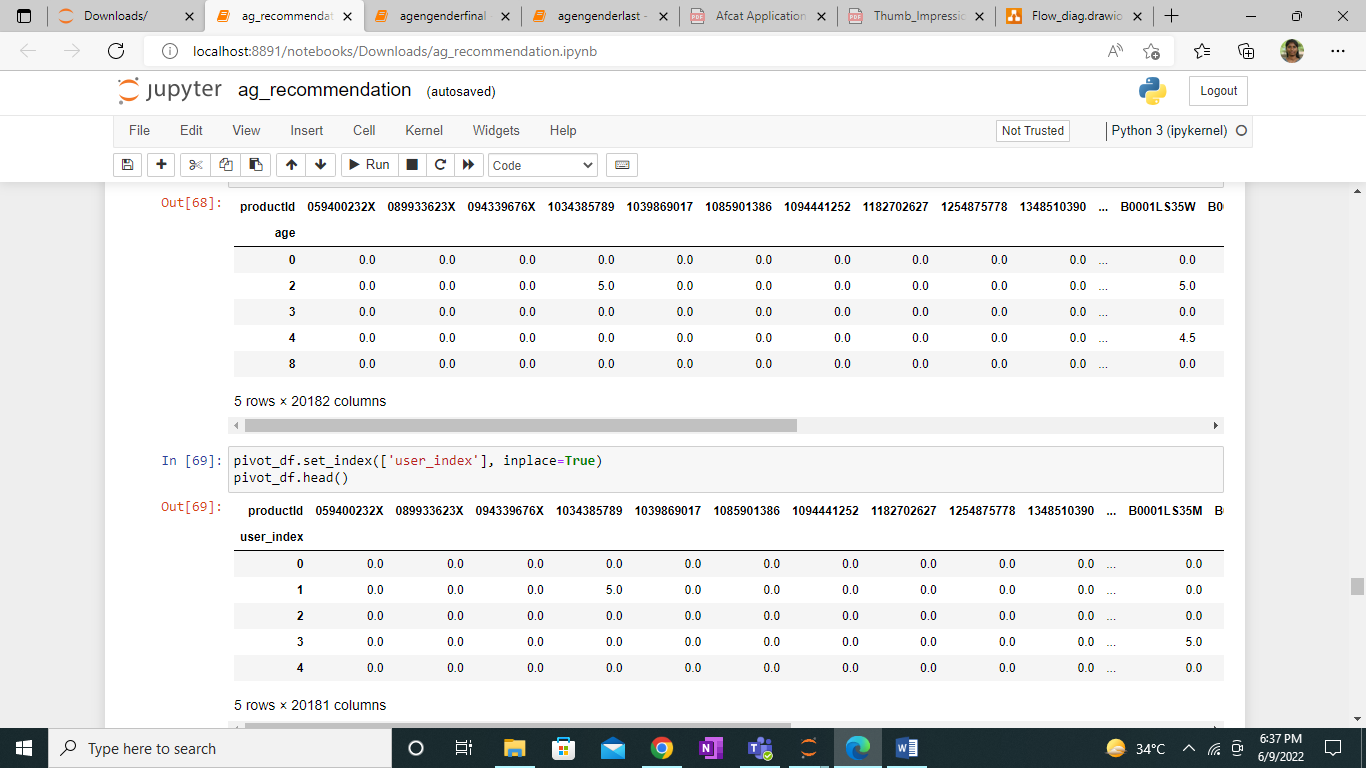
Collaborative filtering is a technique which filters matching products to the users based on their similar users or previous purchases. There are several techniques to implement collaborative filtering.

Here we use SVD (Singular Value Decomposition). The SVD uses the structure of matrix where each column represents the product and each row represents the respective age. SVD reduces the number of features in the dataset by reducing the dimension of space from N to K.



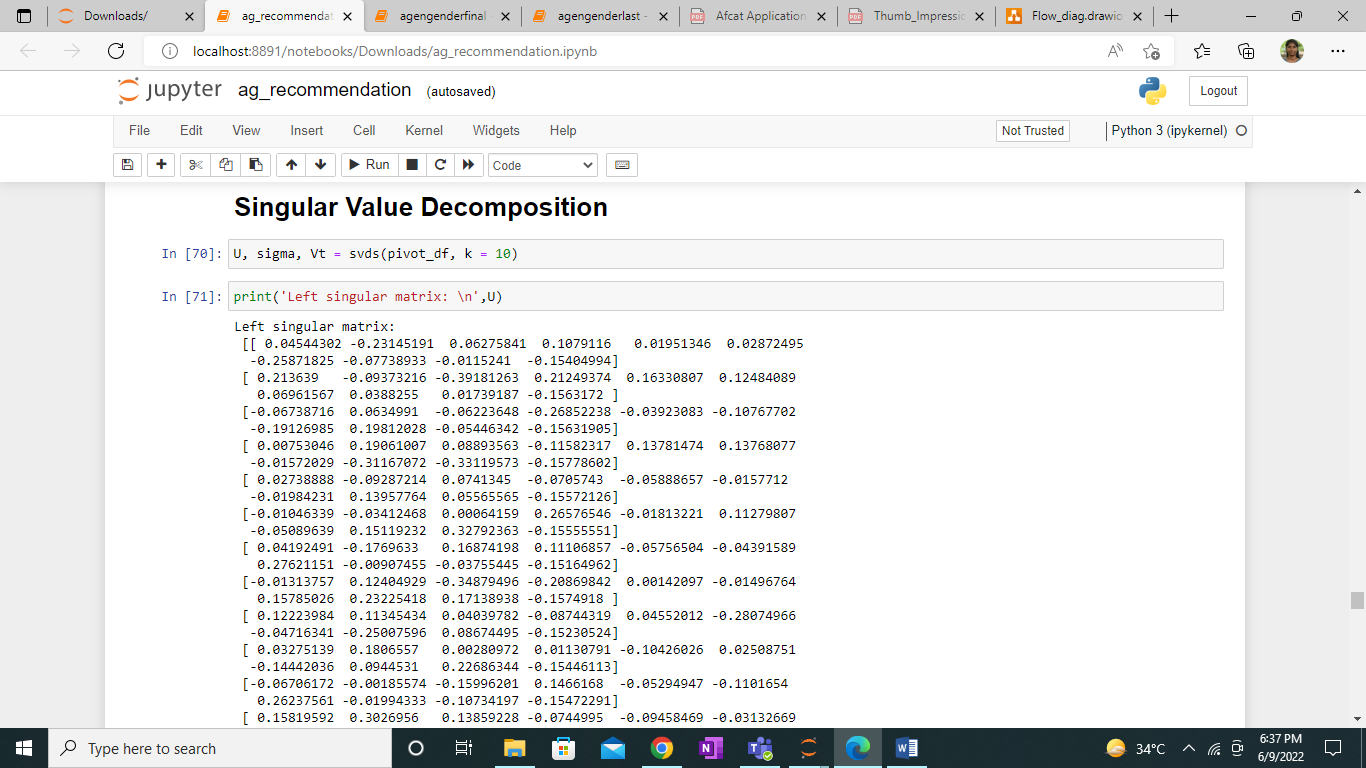
Finally, based on the built user based collaborative model products are recommended to the target audience.



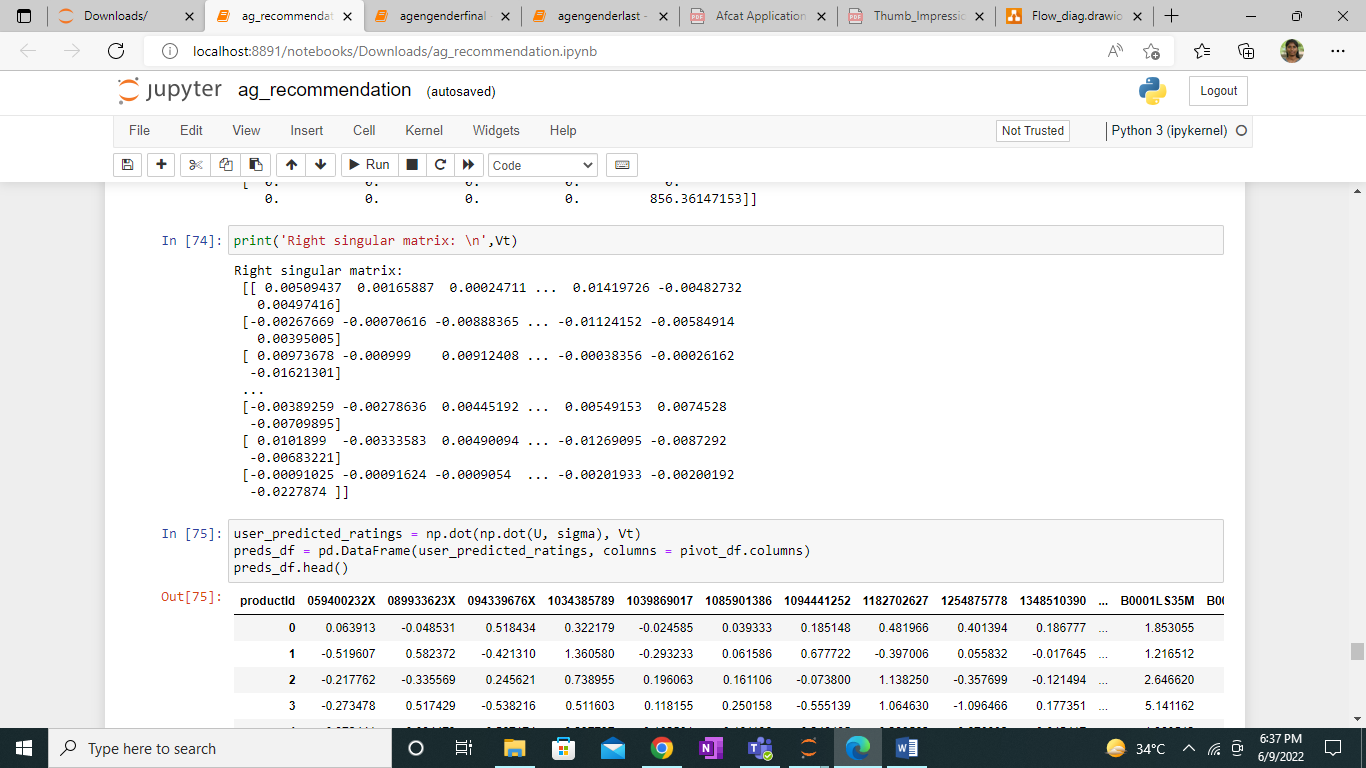


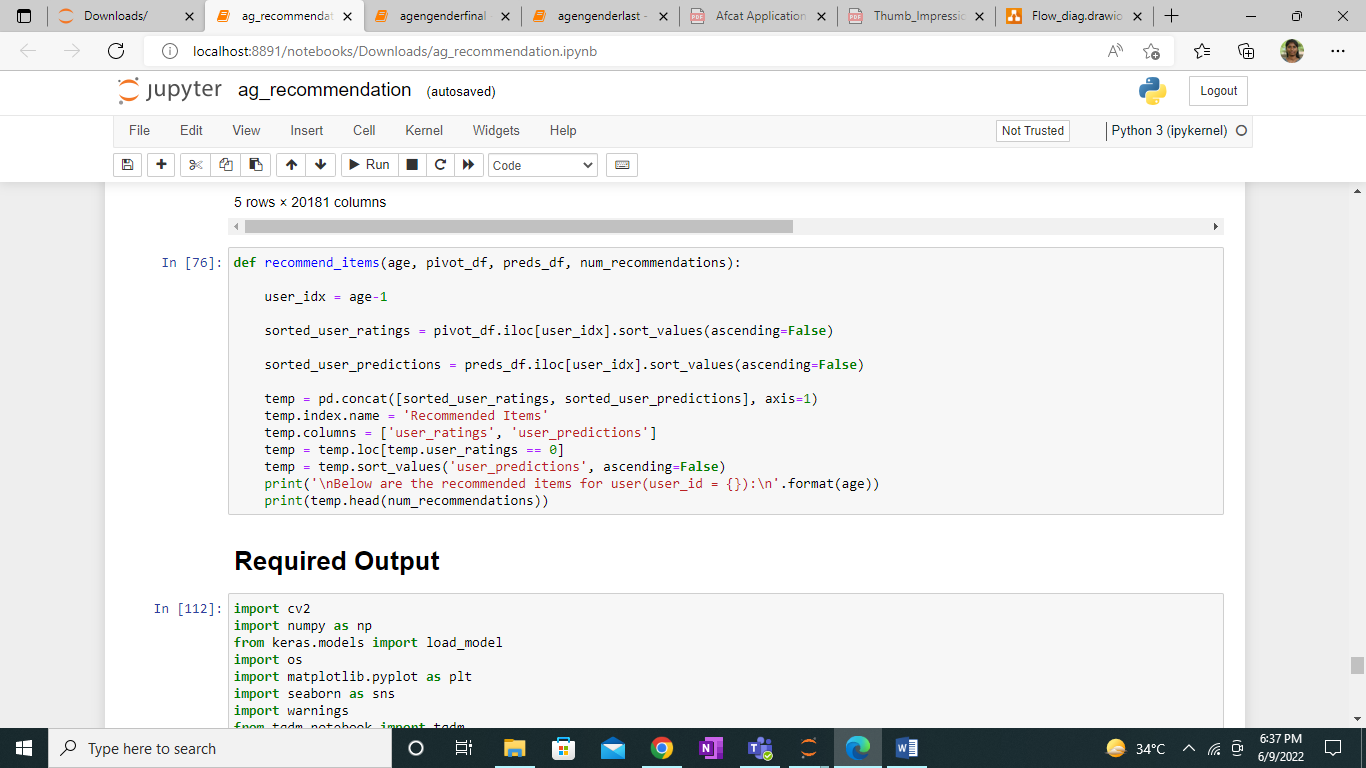
**Singular value decomposition:**

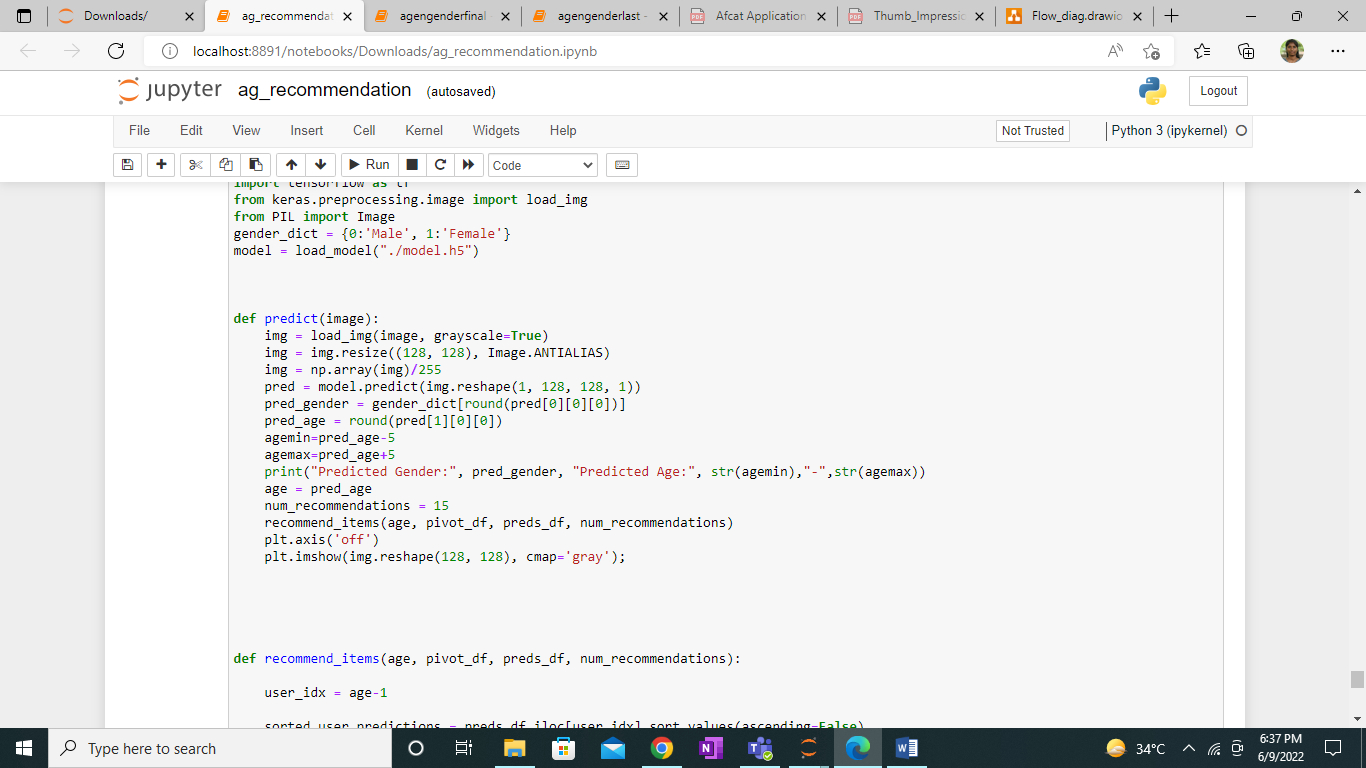
The singular value decomposition is a factorization of a real or complex matrix.  It has some cool applications in Machine Learning and Image Processing.

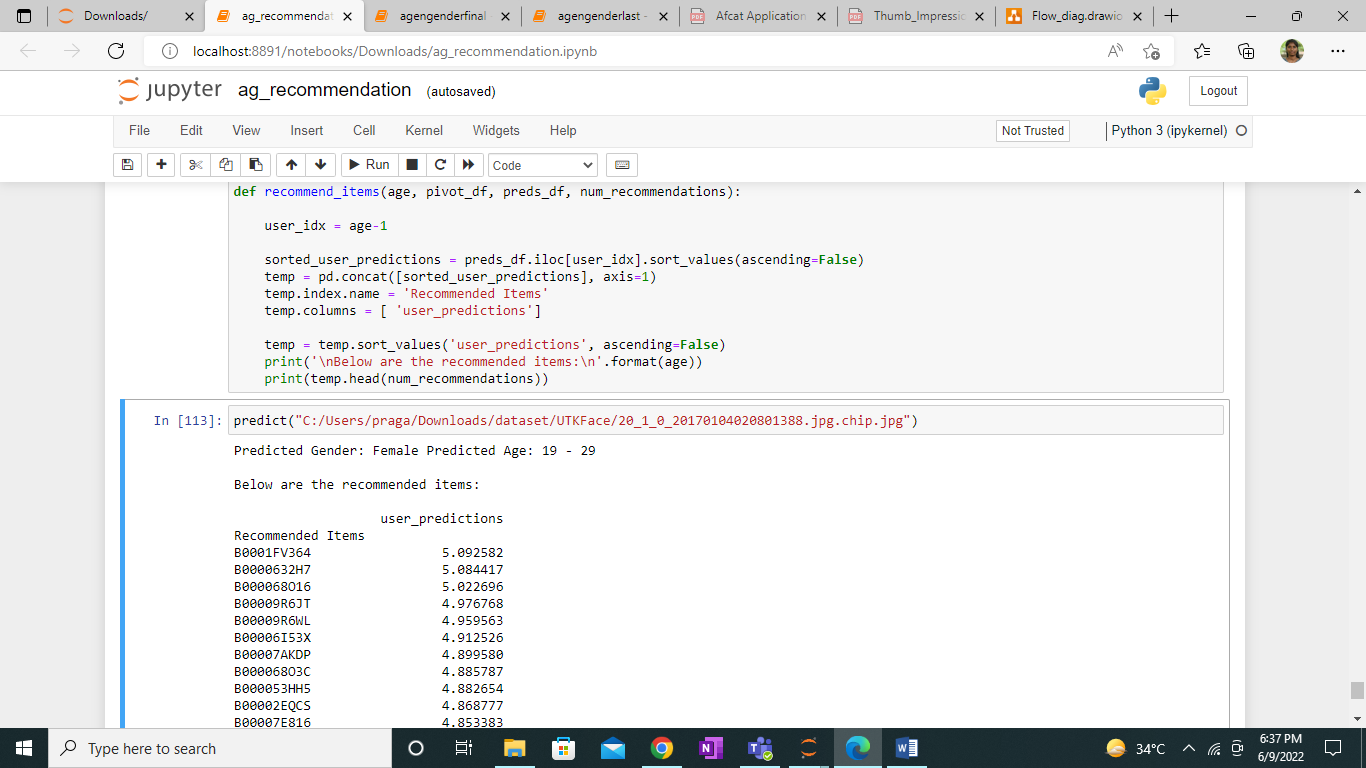


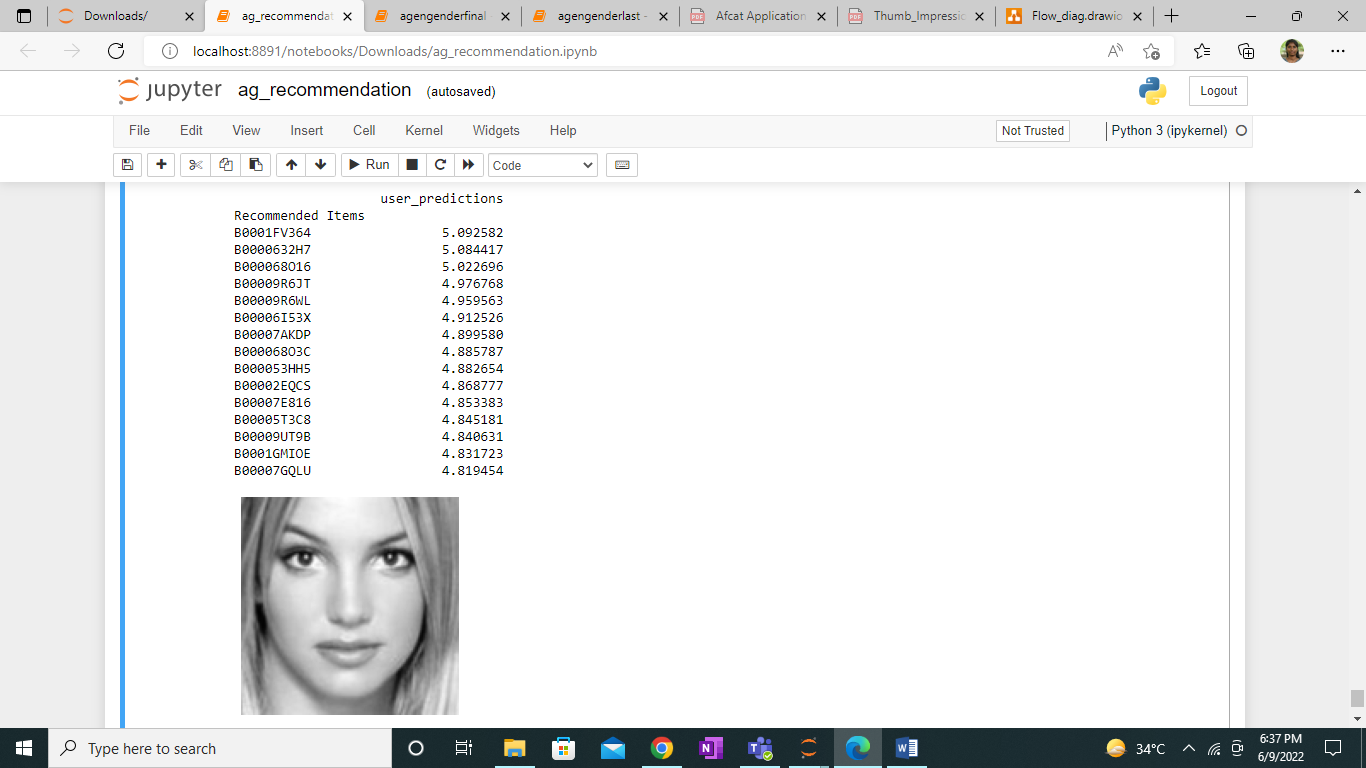












**5. SYSTEM REQUIREMENTS:**

Operating system : Windows 10

Framework : Tensorflow, Keras, NumPy, Pandas, Matplotlib, seaborn,

sklearn, opencv and pillow v8

Language : Python

Source : OIU-Adience Dataset

CPU : intel i5

RAM : 8 GB

**6.CONCLUSION**

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. Hence our project is designed to detect the demographic details such as age and gender and recommend the products to user based on result with help of deep learning techniques. Now the dataset is pre processed and trained .

**7.APPENDIX**

import pandas as pd

import numpy as np

import os

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

from tqdm.notebook import tqdm

warnings.filterwarnings('ignore')

import tensorflow as tf

from keras.preprocessing.image import load\_img

from keras.models import Sequential, Model

from keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D, Input

dataset = './dataset/UTKFace'

images = []

ages = []

gender\_label = []

for f\_name in tqdm(os.listdir(dataset)):

image = os.path.join(dataset, f\_name)

temp = f\_name.split('\_')

age = int(temp[0])

gender = int(temp[1])

images.append(image)

ages.append(age)

gender\_label.append(gender)

df = pd.DataFrame()

df['image'], df['age'], df['gender'] = images, ages, gender\_label

df.head()

gender\_dict = {0:'Male', 1:'Female'}

from PIL import Image

img = Image.open(df['image'][790])

plt.axis('off')

plt.imshow(img);

sns.distplot(df['age'])

sns.countplot(df['gender'])

plt.figure(figsize=(20, 20))

files = df.iloc[0:25]

for index, file, age, gender in files.itertuples():

plt.subplot(5, 5, index+1)

img = load\_img(file)

img = np.array(img)

plt.imshow(img)

plt.title(f"Age: {age} Gender: {gender\_dict[gender]}")

plt.axis('off')

def extract\_features(images):

features = []

for image in tqdm(images):

img = load\_img(image, grayscale=True)

img = img.resize((128, 128), Image.ANTIALIAS)

img = np.array(img)

features.append(img)

features = np.array(features)

features = features.reshape(len(features), 128, 128, 1)

return features

X = extract\_features(df['image'])

X.shape

X = X/255.0

y\_gender = np.array(df['gender'])

y\_age = np.array(df['age'])

input\_shape = (128, 128, 1)

inputs = Input((input\_shape))

conv\_1 = Conv2D(32, kernel\_size=(3, 3), activation='relu') (inputs)

maxp\_1 = MaxPooling2D(pool\_size=(2, 2)) (conv\_1)

conv\_2 = Conv2D(64, kernel\_size=(3, 3), activation='relu') (maxp\_1)

maxp\_2 = MaxPooling2D(pool\_size=(2, 2)) (conv\_2)

conv\_3 = Conv2D(128, kernel\_size=(3, 3), activation='relu') (maxp\_2)

maxp\_3 = MaxPooling2D(pool\_size=(2, 2)) (conv\_3)

conv\_4 = Conv2D(256, kernel\_size=(3, 3), activation='relu') (maxp\_3)

maxp\_4 = MaxPooling2D(pool\_size=(2, 2)) (conv\_4)

flatten = Flatten() (maxp\_4)

# fully connected layers

dense\_1 = Dense(256, activation='relu') (flatten)

dense\_2 = Dense(256, activation='relu') (flatten)

dropout\_1 = Dropout(0.3) (dense\_1)

dropout\_2 = Dropout(0.3) (dense\_2)

output\_1 = Dense(1, activation='sigmoid', name='gender\_out') (dropout\_1)

output\_2 = Dense(1, activation='relu', name='age\_out') (dropout\_2)

model = Model(inputs=[inputs], outputs=[output\_1, output\_2])

model.compile(loss=['binary\_crossentropy', 'mae'], optimizer='adam', metrics=['accuracy'])

model.summary()

history = model.fit(x=X, y=[y\_gender, y\_age], batch\_size=32, epochs=30, validation\_split=0.20)

model.save("model.h5")

acc = history.history['gender\_out\_accuracy']

val\_acc = history.history['val\_gender\_out\_accuracy']

epochs = range(len(acc))

plt.plot(epochs, acc, 'b', label='Training Accuracy')

plt.plot(epochs, val\_acc, 'r', label='Validation Accuracy')

plt.title('Accuracy Graph')

plt.legend()

plt.figure()

loss = history.history['gender\_out\_loss']

val\_loss = history.history['val\_gender\_out\_loss']

plt.plot(epochs, loss, 'b', label='Training Loss')

plt.plot(epochs, val\_loss, 'r', label='Validation Loss')

plt.title('Loss Graph')

plt.legend()

plt.show()

loss = history.history['age\_out\_loss']

val\_loss = history.history['val\_age\_out\_loss']

epochs = range(len(loss))

plt.plot(epochs, loss, 'b', label='Training Loss')

plt.plot(epochs, val\_loss, 'r', label='Validation Loss')

plt.title('Loss Graph')

plt.legend()

plt.show()

image\_index = 568

pred = model.predict(X[image\_index].reshape(1, 128, 128, 1))

pred\_gender = gender\_dict[round(pred[0][0][0])]

pred\_age = round(pred[1][0][0])

agemin=pred\_age-5

agemax=pred\_age+5

print("Predicted Gender:", pred\_gender, "\nPredicted Age:", str(agemin),"-",str(agemax))

plt.axis('off')

plt.imshow(X[image\_index].reshape(128, 128), cmap='gray');

image\_index = 7890

pred = model.predict(X[image\_index].reshape(1, 128, 128, 1))

pred\_gender = gender\_dict[round(pred[0][0][0])]

pred\_age = round(pred[1][0][0])

agemin=pred\_age-5

agemax=pred\_age+5

print("Predicted Gender:", pred\_gender, "\nPredicted Age:", str(agemin),"-",str(agemax))

plt.axis('off')

plt.imshow(X[image\_index].reshape(128, 128), cmap='gray');

from IPython.core.interactiveshell import InteractiveShell

InteractiveShell.ast\_node\_interactivity = "all"

import numpy as np

import pandas as pd

import math

import json

import time

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics.pairwise import cosine\_similarity

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import NearestNeighbors

import scipy.sparse

from scipy.sparse import csr\_matrix

from scipy.sparse.linalg import svds

import warnings; warnings.simplefilter('ignore')

%matplotlib inline

columns=['userId', 'productId', 'ratings','timestamp']

df=pd.read\_csv('C:/Users/praga/Downloads/ratings.csv',names=columns)

df.head()

df.drop('timestamp',axis=1,inplace=True)

df.info()

rows,columns=df.shape

print('Rows: ',rows)

print('Columns: ',columns)

df.drop('userId',axis=1,inplace=True)

data=np.random.randint(0,80,size=1048576)

df['age']=data

print(data)

df.head(15)

data=np.random.randint(0,2,size=1048576)

df['gender']=data

print(data)

df.head(15)

df1=df.iloc[:400000,0:]

df1.info()

print('Missing values: \n',df.isnull().sum())

with sns.axes\_style('white'):

g = sns.factorplot("ratings", data=df1, aspect=2.0,kind='count')

g.set\_ylabels("Total number of ratings"

print('Unique age group:', df1['age'].nunique())

print('Unique products:', df1['productId'].nunique())

most\_rated=df1.groupby(['age']).size().sort\_values(ascending=False)[:10]

print('Top 10 users : \n',most\_rated)

counts=df1.age.value\_counts()

df1\_final=df1[df1.age.isin(counts[counts>=5000].index)]

print('Age groups who have rated more items:', len(df1\_final))

print('Unique Age groups above data:', df1\_final['age'].nunique())

print('Unique products in the final data:', df1\_final['productId'].nunique())

train\_data, test\_data = train\_test\_split(df1\_final, test\_size = 0.3, random\_state=0)

train\_data.head()

print('Training data: ',train\_data.shape)

print('Testing data: ',test\_data.shape)

train\_data\_grouped = train\_data.groupby('productId').agg({'age': 'count'}).reset\_index()

train\_data\_grouped.rename(columns = {'age': 'score'},inplace=True)

train\_data\_grouped.head(40)

train\_data\_sort = train\_data\_grouped.sort\_values(['score', 'productId'], ascending = [0,1])

train\_data\_sort['rank'] = train\_data\_sort['score'].rank(ascending=0, method='first')

popularity\_recommendations = train\_data\_sort.head(5)

popularity\_recommendations

def recommend(age):

user\_recommendations = popularity\_recommendations

user\_recommendations['age\_based\_recommendation'] = age

cols = user\_recommendations.columns.tolist()

cols = cols[-1:] + cols[:-1]

user\_recommendations = user\_recommendations[cols]

return user\_recommendations

print(recommend(23))

df\_CF = pd.concat([train\_data, test\_data]).reset\_index()

df\_CF.head()

pivot\_df = df\_CF.pivot\_table(index='age',columns ='productId', values = 'ratings').fillna(0)

pivot\_df.head()

print('Shape:', pivot\_df.shape)

pivot\_df['user\_index'] = np.arange(0, pivot\_df.shape[0], 1)

pivot\_df.head()

pivot\_df.set\_index(['user\_index'], inplace=True)

pivot\_df.head()

U, sigma, Vt = svds(pivot\_df, k = 10)

print('Left singular matrix: \n',U)

print('Sigma: \n',sigma)

sigma = np.diag(sigma)

print('Diagonal matrix: \n',sigma)

print('Right singular matrix: \n',Vt)

user\_predicted\_ratings = np.dot(np.dot(U, sigma), Vt)

preds\_df = pd.DataFrame(user\_predicted\_ratings, columns = pivot\_df.columns)

preds\_df.head()

def recommend\_items(age, pivot\_df, preds\_df, num\_recommendations):

user\_idx = age-1

sorted\_user\_ratings = pivot\_df.iloc[user\_idx].sort\_values(ascending=False)

sorted\_user\_predictions = preds\_df.iloc[user\_idx].sort\_values(ascending=False)

temp = pd.concat([sorted\_user\_ratings, sorted\_user\_predictions], axis=1)

temp.index.name = 'Recommended Items'

temp.columns = ['user\_ratings', 'user\_predictions']

temp = temp.loc[temp.user\_ratings == 0]

temp = temp.sort\_values('user\_predictions', ascending=False)

print('\nBelow are the recommended items for user(user\_id = {}):\n'.format(age))

print(temp.head(num\_recommendations))

import cv2

import numpy as np

from keras.models import load\_model

import os

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

from tqdm.notebook import tqdm

warnings.filterwarnings('ignore')

import tensorflow as tf

from keras.preprocessing.image import load\_img

from PIL import Image

gender\_dict = {0:'Male', 1:'Female'}

model = load\_model("./model.h5")

def predict(image):

img = load\_img(image, grayscale=True)

img = img.resize((128, 128), Image.ANTIALIAS)

img = np.array(img)/255

pred = model.predict(img.reshape(1, 128, 128, 1))

pred\_gender = gender\_dict[round(pred[0][0][0])]

pred\_age = round(pred[1][0][0])

agemin=pred\_age-5

agemax=pred\_age+5

print("Predicted Gender:", pred\_gender, "Predicted Age:", str(agemin),"-",str(agemax))

age = pred\_age

num\_recommendations = 15

recommend\_items(age, pivot\_df, preds\_df, num\_recommendations)

plt.axis('off')

plt.imshow(img.reshape(128, 128), cmap='gray');

def recommend\_items(age, pivot\_df, preds\_df, num\_recommendations):

user\_idx = age-1

sorted\_user\_predictions = preds\_df.iloc[user\_idx].sort\_values(ascending=False)

temp = pd.concat([sorted\_user\_predictions], axis=1)

temp.index.name = 'Recommended Items'

temp.columns = [ 'user\_predictions']

temp = temp.sort\_values('user\_predictions', ascending=False)

print('\nBelow are the recommended items:\n'.format(age))

print(temp.head(num\_recommendations))

predict("C:/Users/praga/Downloads/dataset/UTKFace/20\_1\_0\_20170104020801388.jpg.chip.jpg")

**GITHUB Link:**

<https://github.com/PRAGATHIRAM/Recommendation-System-Based-on-Age-and-Gender>

**REFERENCES:**

1. G. Levi and T. Hassner, “Age and gender classification using convolutional neural networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015, pp. 34–42.
2. K. Ito, H. Kawai, T. Okano and T. Aoki, "Age and Gender Prediction from Face Images Using Convolutional Neural Network," 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2018, pp. 7-11, doi:
   1. 10.23919/APSIPA.2018.8659655.
3. T. Ahonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: Application to face recognition,” IEEE Trans. Pattern Anal. Mach. Intell., no. 12, pp. 2037– 2041, 2006.
4. Rafique, A. Hamid, S. Naseer, M. Asad, M. Awais and T. Yasir, "Age and Gender Prediction using Deep Convolutional Neural Networks," 2019 International Conference on Innovative Computing (ICIC), 2019, pp. 1-6, doi: 10.1109/ICIC48496.2019.8966704.

1. H. A. Sánchez-Hevia, R. Gil-Pita, M. Utrilla-Manso and M. Rosa-Zurera, "Convolutionalrecurrent Neural Network for Age and Gender Prediction from Speech," 2019 Signal Processing Symposium (SPSympo), 2019, pp. 242-245, doi: 10.1109/SPS.2019.8881961.
2. P. Kaushik, A. Gupta, P. P. Roy and D. P. Dogra, "EEG-Based Age and Gender Prediction Using Deep BLSTM-LSTM Network Model," in IEEE Sensors Journal, vol. 19, no. 7, pp. 2634-2641, 1 April1, 2019, doi: 10.1109/JSEN.2018.2885582.
3. Khaung Tin, Dr.Hlaing Htake. (2011). Gender and Age Estimation Based on Facial Images. International Journal : ACTA TECJNICA NAPOCENSIS Electronics and Telecommunications.
4. *Michal Uricar, Radu Timofte, Rasmus Rothe, Jiri Matas, Luc Van Gool*; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2016, pp. 25-33
5. P. Nguyen, D. Tran, X. Huang and W. Ma, "Age and gender classification using EEG paralinguistic features," *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)*, 2013, pp. 1295-1298, doi: 10.1109/NER.2013.6696178.
6. Marquardt, James, et al. “Age and Gender Identification in Social Media.” *Proceedings of CLEF 2014 Evaluation Labs*, vol. 1180, 2014, pp. 1129–1136.
7. X. Xu, J. Zhou, Y. Liu, Z. Xu and X. Zhao, "Taxi-RS: Taxi-Hunting
8. Recommendation System Based on Taxi GPS Data," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, pp. 1716-1727, Aug. 2015, doi: 10.1109/TITS.2014.2371815.
9. P. Satheesan, P. S. Haddela and J. Alosius, "Product Recommendation System for Supermarket," 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 2020, pp. 930-935, doi: 10.1109/ICMLA51294.2020.00151.
10. 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.