HACKATHON 2020

PROBLEM STATEMENT 2

BUILD A CONVERSATIONAL SOLUTION THAT ENABLES CUSTOMERS TO DISCOVER AND ORDER PRODUCTS.

BY TEAM AAA

ANURAG R
AKSHAY KUMAR R
ABHISHEK A MESTA

ABOUT THE PROJECT

- This application provides a way to search and order products from a particular store by speaking to the microphone.

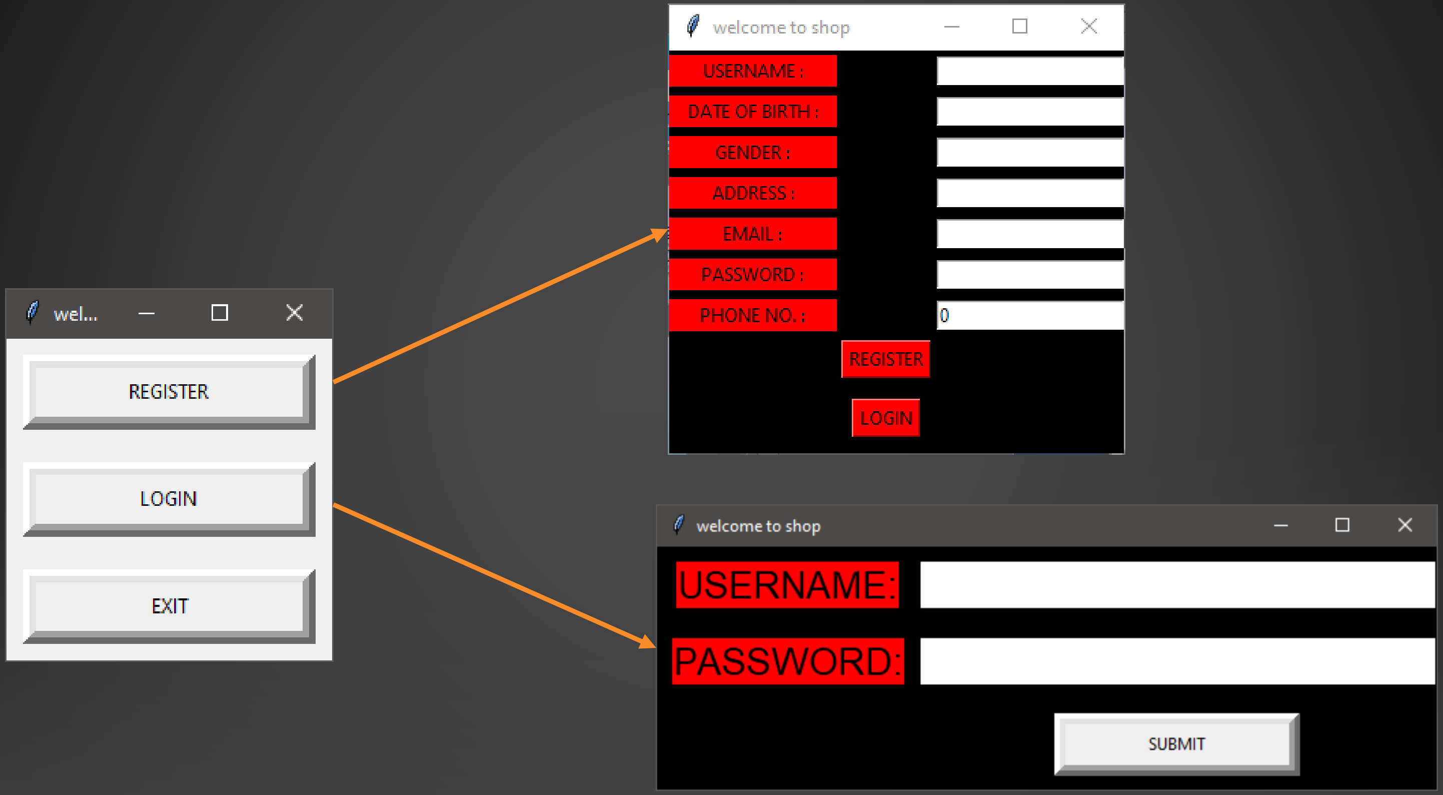
- It also uses an Item recommender system to recommend products to users.

- This project is implemented in Python 3.7 with the help of various libraries

FEATURES

- Login and Register page for users.
- Search for products, view all products, view all your order using the speak function.
- Recommendation of products using collaborative filtering Item Recommender.
- Entering quantity using the speak function.
- View all products in the store
- View your previous orders

WORK FLOW



wel...

REGISTER

LOGIN

EXIT

welcome to shop

USERNAME :	
DATE OF BIRTH :	
GENDER :	
ADDRESS :	
EMAIL :	
PASSWORD :	
PHONE NO. :	0

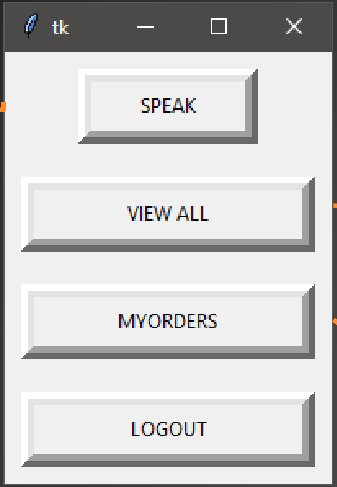
REGISTER

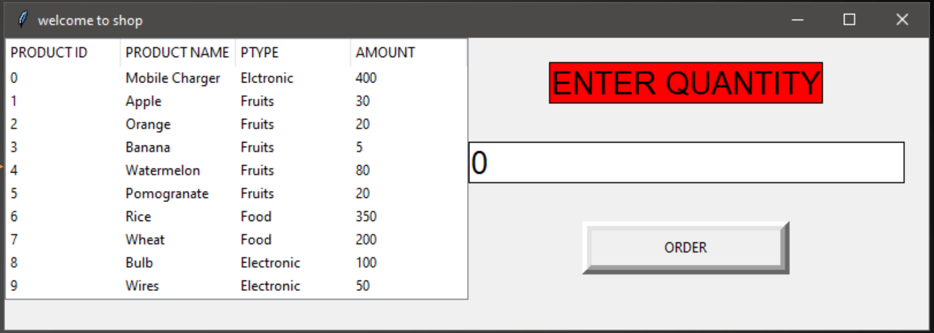
LOGIN

welcome to shop

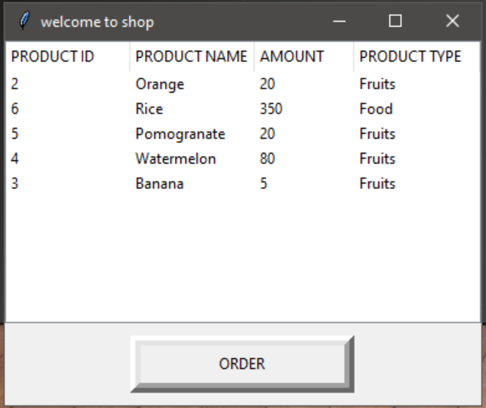
USERNAME:	
PASSWORD:	

SUBMIT

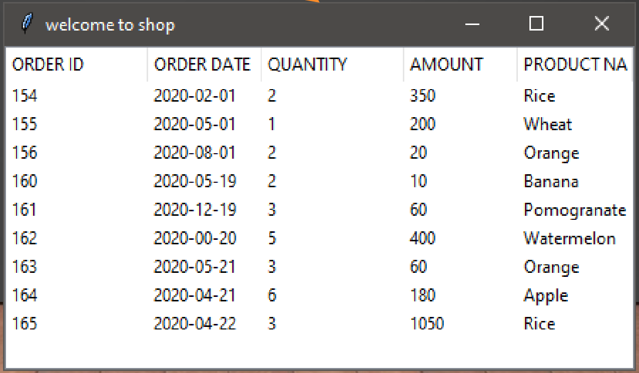




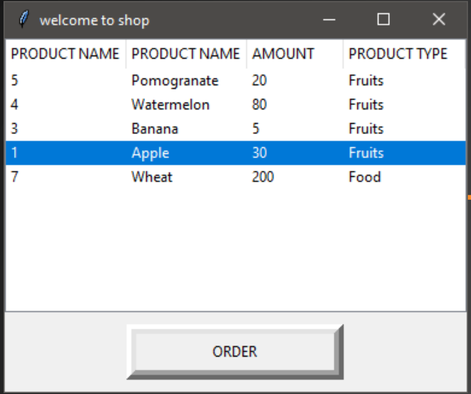
PRODUCT ID	PRODUCT NAME	PTYPE	AMOUNT
0	Mobile Charger	Elctronic	400
1	Apple	Fruits	30
2	Orange	Fruits	20
3	Banana	Fruits	5
4	Watermelon	Fruits	80
5	Pomogranate	Fruits	20
6	Rice	Food	350
7	Wheat	Food	200
8	Bulb	Electronic	100
9	Wires	Electronic	50



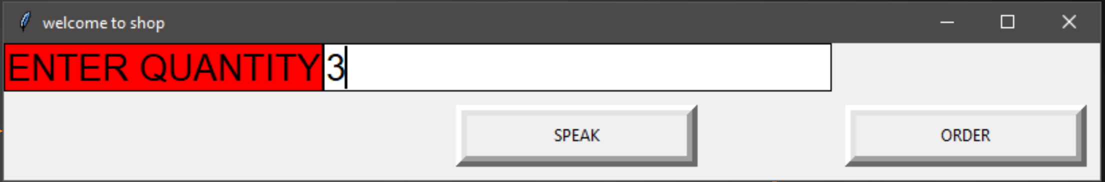
PRODUCT ID	PRODUCT NAME	AMOUNT	PRODUCT TYPE
2	Orange	20	Fruits
6	Rice	350	Food
5	Pomogranate	20	Fruits
4	Watermelon	80	Fruits
3	Banana	5	Fruits



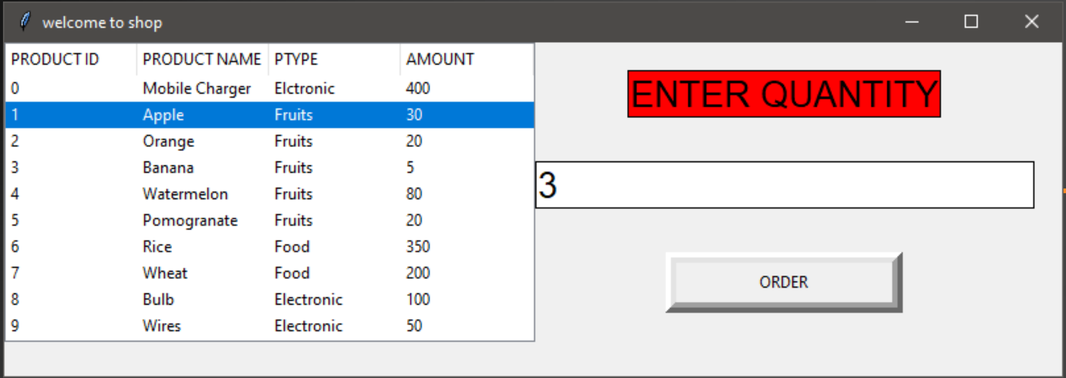
ORDER ID	ORDER DATE	QUANTITY	AMOUNT	PRODUCT NA
154	2020-02-01	2	350	Rice
155	2020-05-01	1	200	Wheat
156	2020-08-01	2	20	Orange
160	2020-05-19	2	10	Banana
161	2020-12-19	3	60	Pomogranate
162	2020-00-20	5	400	Watermelon
163	2020-05-21	3	60	Orange
164	2020-04-21	6	180	Apple
165	2020-04-22	3	1050	Rice



PRODUCT NAME	PRODUCT NAME	AMOUNT	PRODUCT TYPE
5	Pomogranate	20	Fruits
4	Watermelon	80	Fruits
3	Banana	5	Fruits
1	Apple	30	Fruits
7	Wheat	200	Food



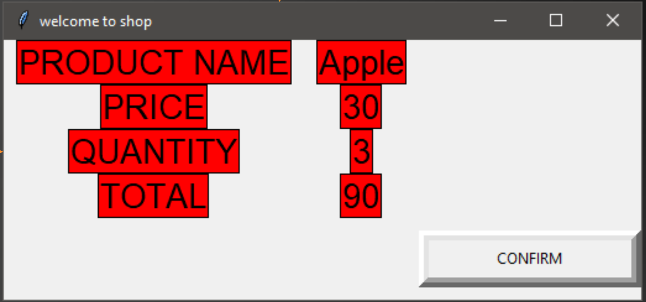
ENTER QUANTITY 3



PRODUCT ID	PRODUCT NAME	PTYPE	AMOUNT
0	Mobile Charger	Elctronic	400
1	Apple	Fruits	30
2	Orange	Fruits	20
3	Banana	Fruits	5
4	Watermelon	Fruits	80
5	Pomogranate	Fruits	20
6	Rice	Food	350
7	Wheat	Food	200
8	Bulb	Electronic	100
9	Wires	Electronic	50

ENTER QUANTITY

3



PRODUCT NAME	Apple
PRICE	30
QUANTITY	3
TOTAL	90

SPEECH RECOGNITION

- This is achieved basically through google speech recognizer. So we first import the library speech_recognition. A requisite is an internet connection.

- It uses the microphone as audio listener and it can be activated through the method or function called Recognizer().

- Using the listen () function the audio is received from the microphone and this is stored in a variable of type string.

- This function is to search for products, view all products, view all orders and to enter quantity.

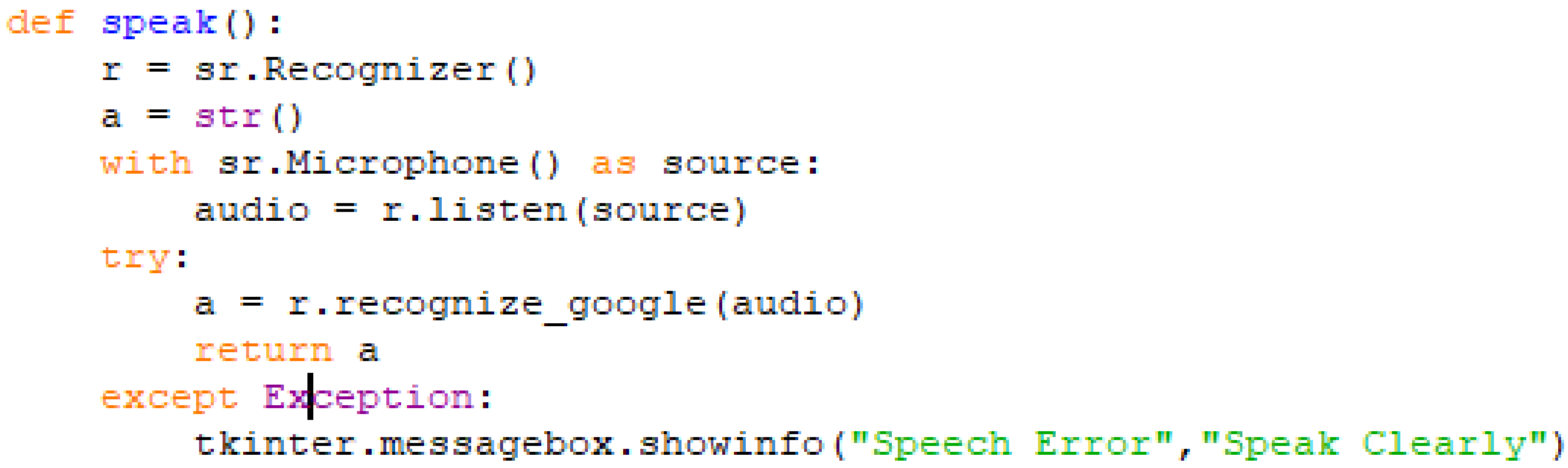
SPEECH RECOGNITION FOR SEARCHING PRODUCTS AND VIEWING ORDERS

- The is the product name is searched from the database and displayed on the GUI if present.
- If multiple words are present it tries to search for all the words in the string.
- If the product is not present in the database or if the user speaks about irrelevant data, the project displays a warning.
- The list of products are displayed along with similar items are displayed to the user.
- The user can also view all their orders by speaking ‘view my orders’.
- It also allows users to show all products by speaking ‘view all products’

SPEECH RECOGNITION FOR QUANTITY

- For Quantity, the working of the code is similar to that of searching for a product, but uses the concept of regular expressions or regex.
- Here the user first gives the input for the quantity of the product through speech. A variable holds the speech as a string.
- It can entered by speaking ‘quantity’ along with a number.
- If matched, the quantity is finalized and the GUI for confirmation page is displayed.

CODE SNIPPET FOR SPEECH RECOGNITION



```python
def speak():
    r = sr.Recognizer()
    a = str()
    with sr.Microphone() as source:
        audio = r.listen(source)
    try:
        a = r.recognize_google(audio)
        return a
    except Exception:
        tkinter.messagebox.showinfo("Speech Error","Speak Clearly")
```

MACHINE LEARNING CONCEPTS

- An Item Recommender System based on collaborative filtering is used to recommend products to the user.

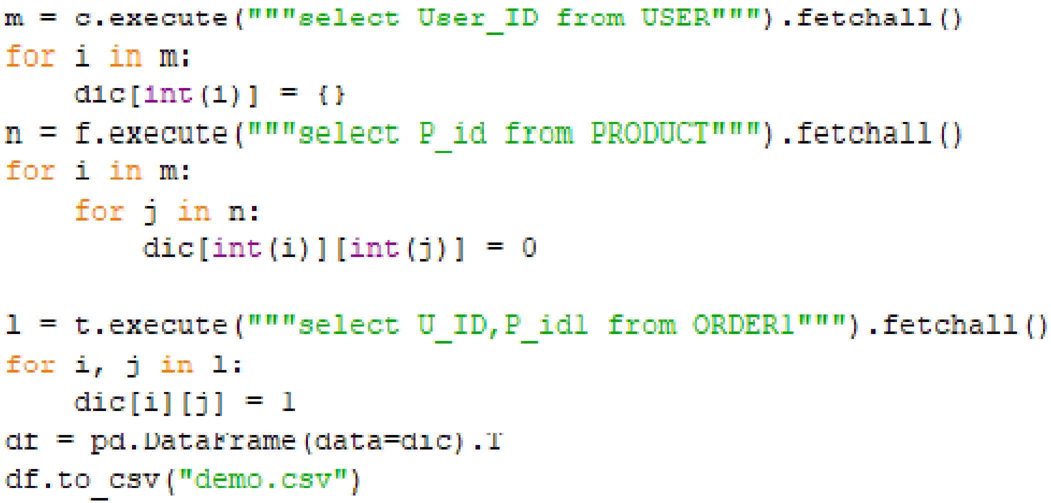
- Pandas library used to achieve the same. It has many functions to do operations on the dataset.

- There are 2 parts of machine learning concepts that we have used in our project:
  - ➢ Creating the dataset from the database
  - ➢ Item Recommendation using the dataset

CREATING THE DATASET

- For creating the dataset from the database we have used a dictionary.
- From the orders table we create the dictionary. The dictionary will hold the key value as user_id and the values as product id.
- First, we will initialize all the values in the dictionary to 0.
- Later, If the user has ordered the product then the dictionary value with that key (user id) will be replaced by 1.
- Finally, the dictionary will be converted to 2 dimensional array with the help of pandas library and then it will be converted to csv file using df.to_csv("demo.csv")

CODE SNIPPET TO WRITE TO FILE



```
m = c.execute("""select User_ID from USER""").fetchall()
for i in m:
    dic[int(i)] = {}
n = f.execute("""select P_id from PRODUCT""").fetchall()
for i in m:
    for j in n:
        dic[int(i)][int(j)] = 0

l = t.execute("""select U_ID,P_id1 from ORDER1""").fetchall()
for i, j in l:
    dic[i][j] = 1
df = pd.DataFrame(data=dic).T
df.to_csv("demo.csv")
```

RECOMMENDATION USING THE DATASET

- The second part will read the csv file and the variable ratings will take the data from the csv file and for that ratings we will find correlation matrix using the Pearson correlation.

- Pearson correlation is a standardized measure of linear association between two sets of scores (standardized, meaning that the result is constrained between -1 and +1).

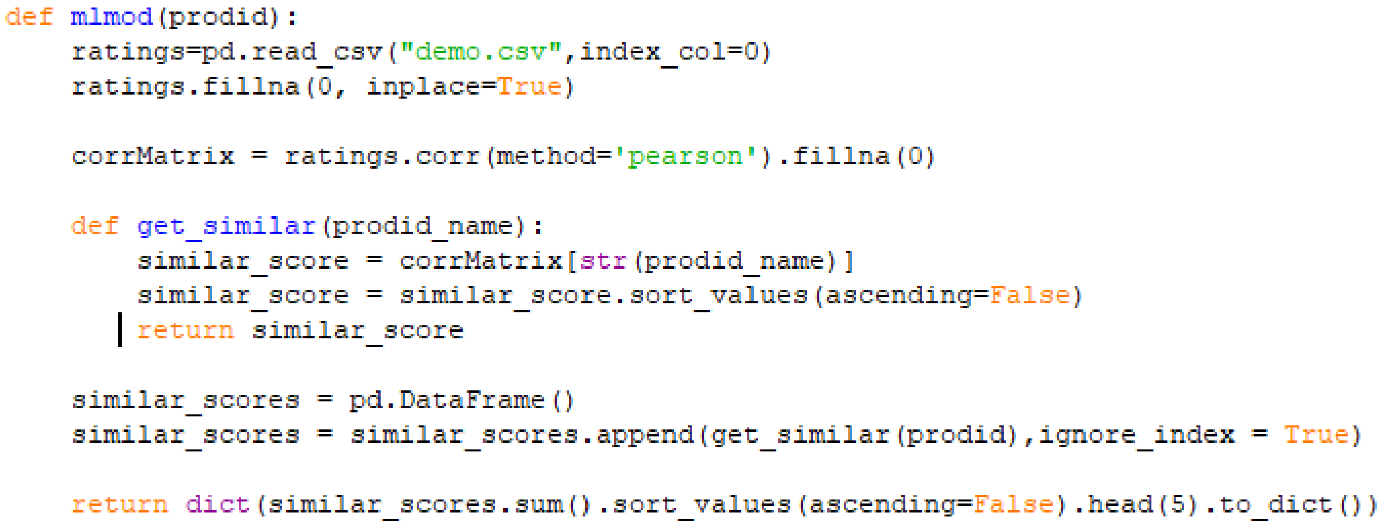
Formula: r(x,y) = cov(x,y) /σx σy

Where Cov(x,y) = Σ ((xi – x) * (yi – y)) / (N – 1)

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

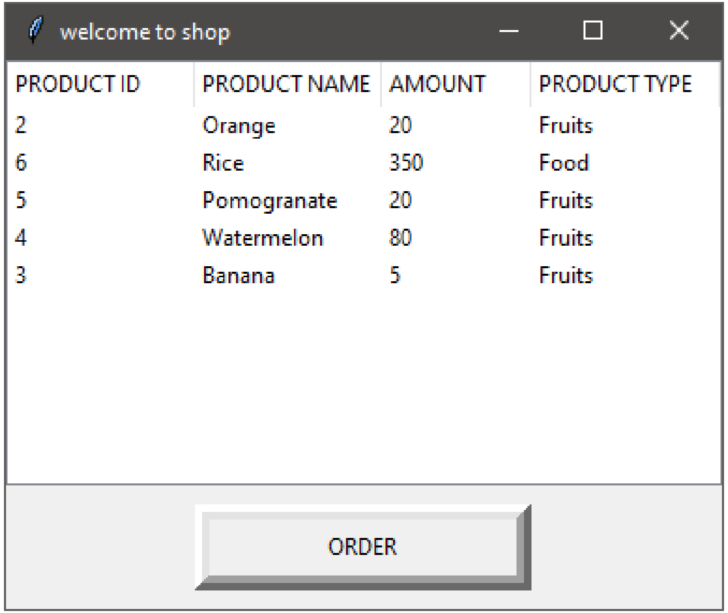
- We find the correlation data for all the products with the current product.
- The correlation data for the product is sorted in the descending order and converted into a dataFrame.
- Finally, that dataFrame values will be given to dictionary where it stores data into keys and values. And also the dictionary will contain only first 5 data.
- The 5 key values of the dictionary will be the recommended products, one will be the searched product and remaining 4 will be the recommended products.

CODE SNIPPET TO RECOMMEND PRODUCTS



```
def mlmod(prodid):
    ratings=pd.read_csv("demo.csv",index_col=0)
    ratings.fillna(0, inplace=True)

    corrMatrix = ratings.corr(method='pearson').fillna(0)

    def get_similar(prodid_name):
        similar_score = corrMatrix[str(prodid_name)]
        similar_score = similar_score.sort_values(ascending=False)
        return similar_score

    similar_scores = pd.DataFrame()
    similar_scores = similar_scores.append(get_similar(prodid),ignore_index = True)

    return dict(similar_scores.sum().sort_values(ascending=False).head(5).to_dict())
```

SEARCHED AND RECOMMENDED PRODUCTS



PRODUCT ID	PRODUCT NAME	AMOUNT	PRODUCT TYPE
2	Orange	20	Fruits
6	Rice	350	Food
5	Pomogranate	20	Fruits
4	Watermelon	80	Fruits
3	Banana	5	Fruits

CHALLENGES FACED

- Since the project is based on speech recognition, finding an appropriate speech recognizer was a major issue. This was however solved using the google speech recognition engine.

- Another challenge faced is that, the speech recognizer always needs an internet connection in order to listen continuously.

- The GUI used in the project is Tkinter which is an old tool, hence the customization options are limited.

- Another major challenge was to choose an appropriate Machine Learning model and the libraries to be used in order to meet the requirements of an Item recommender.

- Finally, in order for the machine learning to work efficiently, dataset is a very important aspect. Since there was no proper data available to use, we had to create a dataset manually.

FUTURE IMPROVEMENTS

- Speech recognition can be improved by adding more natural sounding conversation matching to the project.
- The quality of the recommendation can be improved with the increase in dataset size and quality.
- Since the project is a real time application, more features like admin access can be added so that a worker is allotted to a user to deliver the ordered products.
- The project can be made object oriented so that data gets distributed module wise.
- The GUI can be improved using tools like PyQT, kivy etc in order to appeal a more presentable look and feel.
- Refactoring of code to make it faster and easier to read.

THANK YOU