Model Documentation & Selection Criteria

Scenario1 : Using user's age , gender and the most looked at food , recommend the top 5 food items based on this group of similar user

All Codes file can be found in the FYP repo under the filename "all models"

Only Collaborative filtering models using features from user-food matrix & cosine similarity/ manhattan distance can be found in the UpdatedFoodRecommender_new & Sprint 3 folder. The rest of them are in the all models folder > fooddataset_model.ipynb.

Model 1) Neural Network Collaborative Filtering Model

Codes file name:

- neuralCF_model.py (found in sprint 3 folder)
- Copy paste and run in jupyter notebook or google collab

Explanation:

- One hot encode categorical datas
- For every row of records , loop them, so that the i th food in the row is the indicated food and the next food(k t) chosen by the user will be the recommended food for that user .

Evaluation Metrics:

- We calculate the error by imposing an intersection matrix which separates the data set into 80 by 20 . And we are treating this 20% of the data as the test data while 80% as training data. By randomly taking 1 value (food) from each of the rows (top 6 food recommendations)in the 20% , we parse them into the model to see if the prediction actually matches the top 6 food recommendations that this user has already provided.

Model Results:

1. Accuracy : 12% (round up)
2. Classification Accuracy : 49%

Model 2) Collaborative Filtering Model using features from user-food matrix & cosine similarity

Codes file name:

- Age_gender_1food_recommender.py (found in sprint 3 & Updatedfoodrecommender_new folder)

Explanation:

- Firstly , we group each age group and gender together and then based on this find the model that belongs to this age group and use it. Here vectors in the user-food matrix have been directly used as feature vectors as shown in below example.

	user	course 1	course 2	course 3
0	1	0	0	1
1	2	1	1	1
2	3	0	1	0

- The cosine similarity is the basic similarity calculation for the Collaborative filtering model

Evaluation metrics explanation:

- For this , we count how many errors there are in the prediction. One food from each user has been selected as the test data and the index of the column that represents the selected food is saved in a list.

Model results (there's 10 values as 1 value for each group(age group+gender)) **:**

1. **RMSE = [** 0.7045295724414412, 0.7815355136296579, 0.7753156587418183, 0.7568961844822517, 0.5848456268594958, 0.7495821138526133, 0.7814356586960407, 0.7488251360542963, 0.7755105453750466, 0.515046767033722]
2. **MSE= [**0.49636191844451993, 0.6107977590643732, 0.6011143706902596, 0.5728918340837909, 0.3420444072566766, 0.5618733454077521, 0.610641688681715, 0.5607390843867354, 0.6014166059879023, 0.26527317223188907**]**
3. **MAPE= [**0.6972939194013893, 0.7797498439616243, 0.7732771775522841, 0.7547135933982964, 0.5587154456045916, 0.7463330774343778, 0.7788456880568617, 0.7448603385345375, 0.7728699401789614, 0.5002727545032003**]**

Recommend function

- Take in gender , age group , most favorite food , the sim_matrix model u want to use and 5(meaning u want to get the top 5 food)

Model 3) Collaborative Filtering Model using features from Svd & cosine similarity

Codes file name:

- Fooddataset_model.ipynb (found in all model folder)

Explanation:

- This model also started with grouping the age group and gender together and using the model that belongs to that group. This uses the Singular value decomposition approach to generate the feature vectors and the cosine similarity for the similarity measure

Evaluation Metrics:

- Same approach as the Collaborative filtering

Model results :

1. **RMSE =[**1.019136512886548, 1.0393102133099983, 1.0240803672988077, 1.0293073350350936, 1.0527186582035202, 1.0235748954701809, 1.039119713561543, 1.0234554087612824, 1.0358094449891186, 1.0867965566848719**]**
2. **MSE=[**1.019136512886548, 1.0393102133099983, 1.0240803672988077, 1.0293073350350936, 1.0527186582035202, 1.0235748954701809, 1.039119713561543, 1.0234554087612824, 1.0358094449891186, 1.0867965566848719**]**
3. **MAPE=[** 1.0167571216728855,1.0377659496968197,1.0208993716117258, 1.0214470469840993, 1.033671652899761, 1.022389495792945, 1.0399676972120906, 1.026581207270373,1.0406214380325525, 1.0488710652287636**]**

Recommend function

- Take in gender , age group , most favorite food , the sim_matrix model u want to use and 5(meaning u want to get the top 5 food)

Model 4) Collaborative Filtering Model using features from user-food matrix & euclidean distance similarity

Codes file name:

- Fooddataset_model.ipynb (found in all model folder)

Explanation:

- Similarly , we group each age group and gender together and then based on this find the model that belongs to this age group and use it. Here vectors in the user-food matrix have been directly used as feature vectors as shown in below example. However , we are using euclidean distance similarity here instead of cosine similarity .
- To derive the euclidean distance similarity ,
  1. firstly we calculate the distance
     `euclidean_dist=euclidean_distances(matrix1,matrix2)`
  1. Then we scale it
  `euclidean_dist_scaled = scaler.fit_transform(euclidean_dist)`
  3. Lastly , we use 1 to minus to distance to get the similarity
     `euclidean_sim=1-euclidean_dist_scaled`

Evaluation Metrics:

- Same approach as the other Collaborative filtering

  Model results :

  1. **RMSE =[**0.850406221336782, 0.8641972670869635, 0.8666706376715128, 0.8509421052240019, 0.8257995839867314, 0.8346311181321494, 0.8774936323914835, 0.8380277461165684, 0.8570463767112962, 0.8407310919741262**]**
  2. **MSE= [**0.7231907412883039, 0.7468369164405766, 0.7511179942019466, 0.7241024664430563, 0.6819449529126586, 0.696609103354522, 0.7699950748876, 0.7022905032612156, 0.7345284918339611, 0.7068287690120068**]**
  3. **MAPE=[**0.8451980935408517, 0.8630029183313367, 0.8654079173733976, 0.8490464181672938, 0.8191635353427169, 0.8315591438135962, 0.8758365026278027, 0.8341801643191011, 0.8555417804991299, 0.8392283442782751**]**

Recommend function

- Take in gender , age group , most favorite food , the sim_matrix model u want to use and 5(meaning u want to get the top 5 food)

Model 5) Collaborative Filtering Model features from user-food matrix & manhattan distance similarity

Codes file name:

- Fooddataset_model.ipynb (found in all model folder)

Explanation:

- For this model it works the same way as the other collaborative filtering model that uses vectors in the user-food matrix as feature vectors. We group each age group and gender together and then based on this find the model that belongs to this age group and use it. Here vectors in the user-food matrix have been directly used as feature vectors as shown in below example. However , we are using manhattan distance similarity here instead of cosine similarity or euclidean distance similarity.
- To derive the euclidean distance similarity ,
  1. firstly we calculate the distance
     `manhattan_dist=manhattan_distances(matrix1,matrix2)`
  2. Then we scale it
     `manhattan_dist_scaled = scaler.fit_transform(manhattan_dist)`
  3. Lastly , we use 1 to minus to distance to see the similarity
     `manhattan_sim=1-manhattan_dist_scaled`

Evaluation Metrics:

- Same approach as the other Collaborative filtering

  Model results :

  1. **RMSE = [**0.7446861622663645, 0.757477203033629, 0.7601041700532033, 0.7379689032317274, 0.7124779165204906, 0.7146377472717956, 0.7805220364597792, 0.7240684595298313, 0.7458398724631733, 0.7186160666317453**]**
  2. **MSE= [**0.5545574802710063, 0.5737717131156496, 0.577758349332269, 0.5445981021370386, 0.5076247815293793, 0.5107071098257068, 0.609214649399321, 0.524275134085903, 0.5562771153558825, 0.5164090512212809**]**
  3. **MAPE=[**0.7315186609721154, 0.7536219971461843, 0.7559819598379619, 0.732216212361515, 0.6985177304555452, 0.7059990599639788, 0.7756629243156864, 0.7130586999316051, 0.7413955765509095, 0.7143633221220042**]**

Recommend function

- Take in gender , age group , most favorite food , the sim_matrix model u want to use and 5(meaning u want to get the top 5 food)

Links (Here are some links that we feel would be helpful for your understanding)

- https://medium.com/@gshriya195/top-5-distance-similarity-measures-implementation-in-machine-learning-1f68b9ecb0a3
- https://medium.com/analytics-vidhya/euclidean-and-manhattan-distance-metrics-in-machine-learning-a5942a8c9f2f
- https://intellifysolutions.com/blog/techniques-of-calculating-similarity-distance-measure/
- https://www.analyticsvidhya.com/blog/2020/02/4-types-of-distance-metrics-in-machine-learning/
- Collaborative Filtering for Movie Recommendations (keras.io)

Calculation Explanation

rating_(i,j) = similarity_matrix_i * data_matrix_j * weight

Similarity_matrix_i = i th row of user_similarity_matrix

data_matrix_j= j th column of data_matrix_useritem

weight=1/sum(data_matrix_useritem_j !=0)

Movie

User				
	A	?	3	2
	B	1	2	?
	C	5	?	4

We don't know how much A likes to watch movie A , can find using rating of B & C & their similarity to user A

Summary for all CF models

Model Name	Model Evaluation Metrics (RMSE)	Model Evaluation Metrics (MSE)	Model Evaluation Metrics (MAPE)
Collaborative Filtering Model using features from user-food matrix & cosine similarity	0.7045295724414412, 0.7815355136296579, 0.7753156587418183, 0.7568961844822517, 0.5848456268594958, 0.7495821138526133, 0.7814356586960407,	0.49636191844451993, 0.6107977590643732, 0.6011143706902596, 0.5728918340837909, 0.3420444072566766, 0.5618733454077521, 0.610641688681715,	[0.6972939194013893, 0.7797498439616243, 0.7732771775522841, 0.7547135933982964, 0.5587154456045916, 0.7463330774343778, 0.7788456880568617,

	0.7488251360542963, 0.7755105453750466, 0.515046767033722 avg: **0.719142599026741**	0.5607390843867354, 0.6014166059879023, 0.26527317223188907 Avg: **0.5252084065519684**	0.7448603385345375, 0.7728699401789614, 0.5002727545032003] Avg: **0.7122433038273517**
Collaborative Filtering Model using features from Svd & cosine similarity	[1.019136512886548, 1.0393102133099983, 1.0240803672988077, 1.0293073350350936, 1.0527186582035202, 1.0235748954701809, 1.039119713561543, 1.0234554087612824, 1.0358094449891186, 1.0867965566848719] Avg: 1.0327504873781448	[1.019136512886548, 1.0393102133099983, 1.0240803672988077, 1.0293073350350936, 1.0527186582035202, 1.0235748954701809, 1.039119713561543, 1.0234554087612824, 1.0358094449891186, 1.0867965566848719] Avg: 1.0327504873781448	[1.0167571216728855, 1.0377659496968197, 1.0208993716117258, 1.0214470469840993, 1.03367165289976, 1.022389495792945, 1.0399676972120906, 1.026581207270373, 1.0406214380325525, 1.0488710652287636] Avg: 1.0308972046402016
Collaborative Filtering Model using features from user-food matrix & euclidean distance similarity	[0.850406221336782, 0.8641972670869635, 0.8666706376715128, 0.8509421052240019, 0.8257995839867314, 0.8346311181321494, 0.8774936323914835, 0.8380277461165684, 0.8570463767112962, 0.8407310919741262] Avg: 0.8494765381379448	[0.7231907412883039, 0.7468369164405766, 0.7511179942019466, 0.7241024664430563, 0.6819449529126586, 0.696609103354522, 0.7699950748876, 0.7022905032612156, 0.7345284918339611, 0.7068287690120068] Avg: 0.7217127247106558	[0.8451980935408517 , 0.8630029183313367, 0.8654079173733976, 0.8490464181672938, 0.8191635353427169, 0.8315591438135962, 0.8758365026278027, 0.8341801643191011, 0.8555417804991299, 0.8392283442782751] Avg: 0.8466170158430254
Collaborative Filtering Model features from user-food matrix & manhattan distance similarity	[0.7446861622663645 , 0.757477203033629, 0.7601041700532033, 0.7379689032317274, 0.7124779165204906, 0.7146377472717956, 0.7805220364597792, 0.7240684595298313, 0.7458398724631733, 0.7186160666317453] Avg: 0.7382855098249826	[0.5545574802710063, 0.5737717131156496, 0.577758349332269, 0.5445981021370386, 0.5076247815293793, 0.5107071098257068, 0.609214649399321, 0.524275134085903, 0.5562771153558825, 0.5164090512212809] Avg: 0.5452659390334373	[0.7315186609721154 , 0.7536219971461843, 0.7559819598379619, 0.732216212361515, 0.6985177304555452, 0.7059990599639788, 0.7756629243156864, 0.7130586999316051, 0.7413955765509095, 0.7143633221220042] Avg: 0.7303543280327481

Model 6) Simple Similarity Search algorithm (Jovester)

Codes file name:

- Similarity Search Full Version(found in all model folder)
- Similarity Search With Only Food Item Input(found in all model folder)

Explanation:

- Inputs : Gender, Age and Food Choice
- Output : 5 Food Recommendations
- LabelEncoder is used to turn **Genders** and **Age Groups** to single digit identifiers;
    a. Male to Female will be represented with 0 - 1
    b. Age Groups in increment will be labeled from 0 - 4
- Steps of the Similarity Search:
    a. User input **Food Choice** will be taken in and used to sort out the data frame, only similar users who had chosen the same Food Choice as one of their 6 will be returned as a new dataframe.
    b. Using the new data frame, the user input **Gender** and **Age**, will be taken in and used to sort the new data frame accordingly;
        - Gender
            - Male
            - Female
        - Age will be converted into Age Group
            - 14 and below
            - 15 - 24
            - 25 - 54
            - 55 - 64
            - 65 and above
    c. After sorting again, the returned data frame's users' food features will be collated and converted into percentages that represent the amount of users who had chosen that food out of the total users — all 24 food features will undergo calculation.
    d. The percentage of the user input **Food Choice** will be removed and the remaining top 5 foods with the highest percentage will be recorded into a list.

Model 7) Decision Tree Model and Random Forest Model

Codes file name:

- Decision Tree Model(found in all model folder)
- Random Forest Model(found in all model folder)
- Random Forest Model v2(found in all model folder)

Explanation:

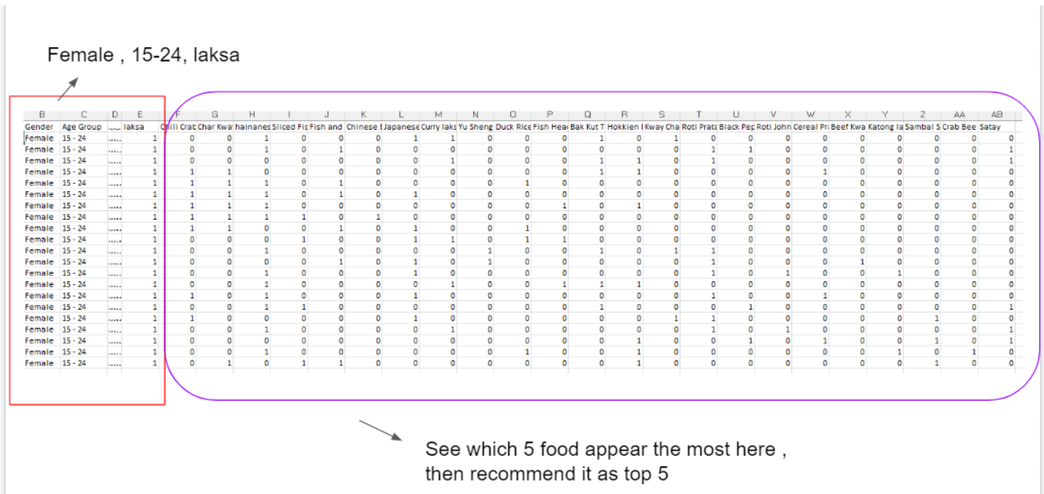
- Inputs : Gender, Age and Food Choice
- Output : 4 to 5 Food Recommendations
- LabelEncoder is used to turn **Genders**, **Age Groups** and **Food Preferences** to digit identifiers;
    a. Male to Female will be represented with 0 - 1
    b. Age Groups in increment will be labeled from 0 - 4
    c. Food Preference is referring to the 6 foods users had chosen but in text.
        - It will be LabelEncoded into values, eg. 221, 557 or 120
        - Note that there can be more than one user who possesses the same value, this is possible because of a possibility of multiple users who had chosen the same 6 food items.
- The 1s and 0s of the food items will be combined to create a long string of 0s and 1s;
    a. Eg. 100110000110000010000000;
    b. The long string above represents the 6 food items that one of the user in the data frame has picked.
    c. Eg. 100000000000000000000000;
    d. The long string above represents the 1 food item the current user has picked, representing laksa; laksa being the first food item in the data frame.
    e. The data will be split into test and train data to be used to train the model.
    f. The model will be fit and will predict by outputting a LabelEncoded Food Preferences who is predicted to have the most similar taste according to the data inputted.
    g. Using the output from the model, the data frame is looped through to get the user corresponding to the output and the long string of 1s and 0s of their / his / her chosen 6 food items will be returned.
    h. With that long string of 1s and 0s, you save the 6 food items into a list depending on the position of the 1s.
    i. The food recommendation would be the food items in that list excluding the current user's chosen food item.
- That list will be looped and it will **recommend** the user **5 other foods** that the user might like according to the data.
- Confusion Matrices are used to determine the accuracy of the model.
  Note : The way I had used these models is mediocrely done and I advise that it is not to be used as a frame of reference
  I put the Decision Tree Model and Random Forest Model together because what I had done was essentially the same process.

Scenario 1(Using user's age , gender and the most looked at food , recommend the top 5 food items based on this group of similar user)'s conclusion

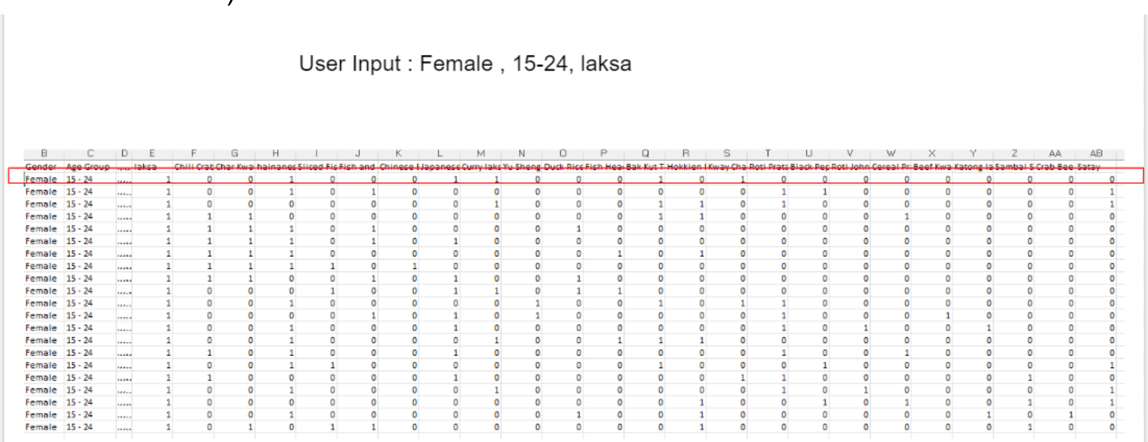
1. Based on all 7 models , we can firstly disregard neural network collaborative filtering models as the <u>accuracy is low</u> . We cannot compare neural networks with the other collaborative filtering model as the comparison metrics are different . Thus , we cannot use it

2. Decision Tree Model and Random Forest Model are models that return the most similar user so it does not give back the top 5 recommendations based on a group of similar users. Therefore we cannot use it.

   It should be like this :

   Instead of just returning the first row that is the most similar to the user (Does not match with our scenario.)

3. Simple Similarity Search algorithm is a rule based algorithm so its not acceptable and there are no evaluation metrics to compare the results of the model. Hence , we can't use this.

4. Collaborative Filtering Model using features from Svd & cosine similarity and Collaborative Filtering Model using features from user-food matrix & euclidean distance similarity and Collaborative Filtering Model features from user-food matrix & manhattan distance similarity , all has a higher RMSE , MSE and MAPE than Collaborative Filtering Model features from user-food matrix & cosine similarity model . Therefore , they are lacking behind this model so they are not chosen.

Withthat , for our scenario 1 our chosen model is : Collaborative Filtering Model features from user-food matrix & cosine similarity model

Scenario2 : Based on **user's gender** & **1 polytechnic course** that the user gaze at , recommend top 5 polytechnic course that the group of similar user has chosen

** Only All collaborative filtering models have been used to try out the new dataset. **

New grouping : Group the Collaborative Filtering model by gender

There will therefore be only 2 collaborative filtering models , one for each gender.

Chosen Model: Collaborative Filtering Model features from user-food matrix & manhattan distance similarity(can be found in the sprint 3 Folder as for the rest its in nypcourse_model.ipynb) -> (found in all model folder)

Summary for all CF models

Model Name	**Model Evaluation Metrics (RMSE)**	**Model Evaluation Metrics (MSE)**	**Model Evaluation Metrics (MAPE)**
Collaborative Filtering Model using features from user-food matrix & cosine similarity	[0.6613870246104936 , 0.6800198294109461] Avg: 0.6707034270107198	[0.4374327963231216, 0.4624269683920922] Avg: 0.4499298823576069	[0.6417904762711245, 0.6684495801044602] Avg: 0.6551200281877924
Collaborative Filtering Model using features from Svd & cosine similarity	[0.7742572186832984 , 0.8086730487640382] Avg: 0.7914651337236683	[0.599474240683197, 0.6539520997973245] Avg: 0.6267131702402607	[[0.752573085033358, 0.7950027113143744] Avg: 0.7737878981738662

Collaborative Filtering Model using features from user-food matrix & euclidean distance similarity	[0.7082700710819311, 0.7198093218587178] Avg: 0.7140396964703244	[0.5016464935904037, 0.5181254598347071] Avg: 0.5098859767125554	[0.6982979773848165, 0.7074653451116474] Avg: 0.8466170158430254
Collaborative Filtering Model features from user-food matrix & manhattan distance similarity	[0.5362329004799613, 0.5591561846142057] Avg: **0.5476945425470835**	[0.28754572355715213, 0.3126556387923157] Avg: **0.30010068117473393**	[0.5100161673421306, 0.5273835507257916] Avg: **0.5186998590339611**

Scenario 2 (Based on **user's gender** & **1 polytechnic course** that the user gaze at , recommend top 5 polytechnic course that the group of similar user has chosen)

1. Based on all 4 Collaborative filtering models , Collaborative Filtering Model using features from user-food matrix & cosine similarity and Collaborative Filtering Model using features from Svd & cosine similarity and Collaborative Filtering Model using features from user-food matrix & euclidean distance similarity has a higher RMSE , MSE and MAPE as compared to Collaborative Filtering Model features from user-food matrix & manhattan distance similarity. One reason for it is because the NYP course data is higher dimensional than the food dataset . And , Manhattan distance works better with high dimensional dataset . Therefore , the chosen model for this scenario will be : <u>Collaborative Filtering Model features from user-food matrix & manhattan distance similarity</u>