Thank you roshini, I will now explain the concept of machine learning which is the AI implemented in the system. Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being distinctly explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

The main model used is collaborative filtering. With research of the previous batch did. collaborative filtering has the most accuracy level compared to the other models. Collaborative filtering is a technique that can filter out items that a user might like based on the reactions by similar users. It works by searching a large group of people and finding a smaller set of users with tastes like similar to a particular user.

We had 4 collaborative models, they are singular value decomposition, user matrix and cosine similarity, User matrix and Euclidean distance and user matrix and Manhattan distance.

Firstly, what is singular value decomposition also known as SVD?? SVD uses a matrix structure where each row represents a user, and each column represents an item. The elements of this matrix are the ratings that are given to items by users.

Secondly, User matrix and cosine similarity, it is a metric used to determine how similar two entities are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space.

Thirdly, user matric and Euclidean distance, Euclidean Distance represents the shortest distance between two points. The “closeness” is defined by the difference (“distance”) along the scale of each variable, which is converted to a similarity measure. This distance is defined as the Euclidean distance.

Lastly, User Matrix & Manhattan Distance is the sum of absolute differences between points across all the dimensions. It is usually preferred over the more common Euclidean distance when there is high dimensionality in the data.

The statement of work consists of two parts, what is needed and what are our objectives. Firstly, we need a batch of data for machine learning and secondly find out the suitable model and algorithm to predict with best accuracy. We had two objectives for this project, firstly, train a model with data and algorithm and integrate the trained model into the system. Secondly, implement suitable model to recommend items based on the user case.

Now, I will past my time to roshini for the travel recommender system.

To start off, I will have an overview of what I did for the past sprints. In sprint 1, I did research on project background, convert past python project into executable file and improve on the layout of previous batch codes to ensure the alignment of the system fits to the booth laptop. Furthermore, conduct survey for my fashion recommender use case.

In sprint2, I collect more data of specific age and gender that has lesser responses. Encode data collected. Evaluate the models with the dataset I had by adding evaluation metrics for each model and integrate the best model into the system.

In sprint3, I change the previous batch interface codes so that after a specific gender is detected by the facial recognition, it will lead to respective catalogue pages I had created, as male and female has different catalogue pages and finished up my first recommender system’s interface. Furthermore, start thinking about new use case to work on.

On the first week of the final sprint, I implement a second layer of content-based filtering for fashion recommender based of the gender. Conduct survey for my new use case and encode data collected.

On the second week, I did model evaluation by adding evaluation metrics for each model. Start creating graphical user interface for my new use case and integrate the best model evaluated into the system. Furthermore, Improve on the facial recognition code in the system.

For the following few days of the last week, I am going to convert the recommender systems I made into executable files and deploy it to booth. As well as working on the documentation of my project.

Here are the enhancements I did during sprint 4. Firstly, I had improved on the model I did for my fashion recommender by adding a second layer of content-based filtering based on gender, so that the system will filter out my data based on the gender first then moving on to collaborative filtering. As for this case the system needs to know the gender of the user first before looking at the preferences of the user.

Secondly, I had created 2 more new user case scenarios. They are NYP CCA recommender and Singapore location recommender.

Thirdly, with the 2 new use case I had, I need to get new dataset by conducting survey.

Lastly, new facial recognition codes.

After implementing the second layer of content-based filtering of the fashion recommender. I did a new model evaluation with the new dataset I had for fashion recommender. Here is a graphical visualization of my model evaluation. For the first figure, it measures the root-mean-square-deviation also known as RMSE of the 4 collaborative filtering models we had mentions previously. It is to measure the error of a model in predicting quantitative data.

The second graph of mean squared error also known as MSE is to measure the amount of error in statistical models. It assesses the average squared difference between the observed and predicted values. Which means when a model has no error, the MSE is equals to zero. As model error increases, its value increases.

The third graph is mean absolute percentage error also known as MAPE. It is the average of the absolute percentage errors of forecasts. Error is defined as actual or observed value minus the forecast value. Which means percentage errors are summed without regard to sign to compute it.

These are the results of the model evaluation. As we can tell User- style matrix and cosine similarity has the least in terms of RMSE, MSE and MAPE compared to the other models. This means that this model has the least error among the other models, hence more accurate.

Thus , with the implementation of the second layer Content-Based Filtering and considering the size of my new style dataset and use case scenario while evaluating how the various model performs , I chose to go with the Collaborative ﬁltering that uses Cosine Similarity Matrix as a similarity measure

Here is the video demo on the enhancements done in the fashion recommender system

The first user is a female age 15 to 24

First by clicking on the start button for the facial recognition to load, it will take 5 seconds for the facial recognition to automatically detect.

After age and gender is detected, a randomized CCA catalogue page will be shown.

For my case I’m looking at biker style

Click on it and it will redirect me to the shopping link of the clothe

I can also click on any images of the recommended style to bring me to the redirect shopping link

The heatmap of my eye gaze will be shown here

The second user is a male age 15 to 24

First by clicking on the start button for the facial recognition to load.

After age and gender is detected, a randomized style catalogue page will be shown.

For his case, he’s looking at streetwear style

Click on it and it will redirect him to the shopping link of streetwear style

He can also click on any images of the recommended styles to bring him to the redirect shopping link of the style

This is the heatmap of his eye gaze just now

Furthermore, in the new facial recognition file I had improved on. I changed the codes so that it records the result of gender and group after 5 seconds. The result is based on the frequency of said gender and age group so that after 5 seconds of facial recognition scan, the most frequently detected gender and age group will be recorded. The recorded result will be written into a text file, in which will be used for the recommendation.

So, what does it improved on? It will automatically detect the age and gender by itself after 5 seconds whereby the previous facial recognition needs to press Q to proceed. The age detection of the facial recognition has also become more accurate by changing the proto text and Caffe model file of the age to a new one. Additionally, the new facial recognition also works more smoothly, as for the previous facial recognition, it might close out of a sudden and causing the system to crash.

Moving on, the first new use case I had. A NYP CCA Recommender.

Based on user’s gender, age and 1 NYP CCA that the user gaze at, recommend top 5 CCA that the group of similar users has chosen

This is an overview of the system design for the CCA recommender. With facial recognition detects the users age and gender. Eye tracker detects the eye gaze of user preferences. And integrate the trained machine learning model with the CCA dataset. It will recommend NYP CCA based on similar users.

For this dataset, I had collected a total of 500 responses. With their age, gender and top 6 most preferred CCA in NYP. There are total of 84 CCA in NYP and they fall into 8 types. Academic clubs, arts and culture, community service environment, sports and adventure, leadership and character development and lastly, societies.

Here is a graphical visualization of my model evaluation. It is shown clearly that the Manhattan distance model that’s in green is the lowest among all. And by looking at the results of the model evaluation. We can tell User-CCA matrix and Manhattan distance have the least in terms of RMSE, MSE and MAPE compared to the other models. This means that this model has the least error among the other models, hence more accurate.

Thus , considering the size of my CCA dataset and use case scenario while evaluating how the various model performs , I chose to go with the Collaborative ﬁltering that uses Matrix & Manhattan Distance as a similarity measure

Here is the video demo on the NYP CCA recommender system.

The first user is a female age 15 to 24

First by clicking on the start button for the facial recognition to load.

After age and gender is detected, a randomized CCA catalogue page will be shown.

For my case I’m looking at SIT club

Click on it and it will redirect me to the sign up link of the CCA

I can also click on any images of the recommended CCA to bring me to the redirect link of the CCA.

The heatmap of my eye gaze will be shown here

The second user is a male age 15 to 24

First by clicking on the start button for the facial recognition to load.

After age and gender is detected, a randomized CCA catalogue page will be shown.

For his case, he’s looking at judo

This is the heatmap of his eye gaze just now

He can click on any images of the recommended CCA to bring him to the redirect sign up link of the CCA.

Moving on the second new use case I had. A Singapore location recommender.

Based on user’s gender, age and 1 location that the user gaze at, recommend top 5 Singapore location that the group of similar users has chosen.

This is an overview of the system design for the Singapore location recommender. With facial recognition detects the users age and gender. Eye tracker detects the eye gaze of user preferences. And integrate the trained machine learning model with the SG location dataset. It will recommend SG location based on similar users.

For this dataset, I had collected a total of 500 responses. With their age, gender, and top 6 most preferred location in Singapore. I had found a total of 260 locations to visit in Singapore and with more research done, among the 260 locations, I implement the use the top 55 locations with the highest rating that is worth going.

Here is a graphical visualization of my model evaluation. It is shown clearly that the Manhattan distance model that’s in green is the lowest among all. And by looking at the results of the model evaluation. We can tell User-Singapore matrix and Manhattan distance have the least in terms of RMSE, MSE and MAPE compared to the other models. This means that this model has the least error among the other models, hence more accurate.

Thus , considering the size of my Singapore location dataset and use case scenario while evaluating how the various model performs , I chose to go with the Collaborative ﬁltering that uses Matrix & Manhattan Distance as a similarity measure

This is a video demo on the Singapore location recommender system.

The first user is a female age 15 to 24

First by clicking on the start button for the facial recognition to load, it will take 5 seconds for the facial recognition to automatically detect.

After age and gender is detected, a randomized cca catalogue page will be shown.

For my case I’m looking at Singapore cable car

Click on it and it will redirect me to the information of the location

I can also click on any images of the recommended location to bring me to the redirect link of specific location

The heatmap of my eye gaze will be shown here

The second user is a male age 15 to 24

First by clicking on the start button for the facial recognition to load.

After age and gender is detected, a randomized Singapore location catalogue page will be shown.

For his case, he’s looking at universal studio Singapore

Click on it and it will redirect him to more information of the location

He can click on any images of the recommended CCA to bring him to the redirect sign up link of the CCA.

Location of the place will be shown at the recommendation as well

This is the heatmap of his eye gaze just now

The major problem I encountered during the sprint is to get sufficient survey data on time, to solve this, I publish my survey on social media such as LinkedIn, reddit and twitter. Another problem is to ensure that each catalogue page has a wide range of items to choose from. As only 8 items are allowed to display in one catalogue page, it is very important to have a variety of choices for the user to choose from. To solve this problem, I split the items in different categories and to ensure that in every catalogue page there is item from different categories.

My plans for the rest of sprint 4 is to deploy the final version of my recommender system to executable files and with new facial recognition updated and continue to finish up the documentation in azure devops.

In conclusion, after second layer of content-based filtering by gender is added to the fashion recommender, the recommended content is more accurate. Collaborative filtering with cosine similarity works better on style dataset while collaborative filtering with Manhattan distance similarity performs better on NYP CCA dataset and Singapore location dataset. Lastly, with new facial recognition updated, user’s age can be detected more accurately, and it is able to run more smoothly.

This is the end of our presentation thank you did you have any question?