# CA2 Individual Report – LIM WEI YANG JEROME (A0124917A)

The team and I are proud of the recommender system that we have built. Nonetheless, despite the relatively smooth process throughout the project, the entire journey was not without any hurdles. This report will discuss some of these difficulties faced, specifically in the data preparation and solution building stages that the team had to face, and the solutions that were implemented to overcome these difficulties.

## Data Preparation Difficulties

### Dataset Finalization

Ironically, the seemingly simplest task of finding a dataset to start off the project took the team a significant amount of time. The criteria of the project stated that in addition to the building of the recommender systems using both content-based and collaborative filtering methods, the element of text analytics and Natural Language Processing also had to be incorporated. Essentially, it was difficult to look for a dataset that piqued the entire group’s interest, and simultaneously had all the different data that were required for the project. Fortunately, all team members were clear of the criteria, and took time to look out for potential datasets that satisfy all the criteria. The team held weekly meetings to discuss the datasets that we have shortlisted and finally selected the Yelp dataset that all members were happy with.

### Machine Capacity Issue

During the actual data preparation process, the team faced machine capacity issues. It was just not feasible to use the full dataset due to the sheer size of the raw dataset. As such, criteria for filtering had to be discussed. Firstly, the number of businesses selected had to be balanced - where the total number of reviews for these restaurants should be sufficient enough for a reliable recommender system to be built, yet not overwhelmingly large such that the teams’ machines are not able to process it at all. In the end, the team settled on a subset of restaurants based on the state of which the restaurants are located in, and then applied a random selection of 2000 restaurants over the first filter (by state).

### Serendipity

In addition, seemingly trivial decisions such as choosing the restaurants that were most reviewed, or to simply choose them randomly proved to be important to account for serendipity. When the team learned about the serendipity problem during the lessons, there was motivation to at least account for it in some way in the project. The first method was the decision to choose the universe of restaurant choices of which the user of the recommender system would eventually be recommended to randomly. This ensures that not only the most popular restaurants (most reviewed restaurants) are constantly recommended to a user. Also, to further introduce possible diversity during the actual model recommendation, the team has added a parameter that allows a fixed number of restaurants (that might be outside of the cuisine that the user of the recommender system has entered) into the choice universe. This also creates diversity as to what may be recommended to the user. The team’s main report discusses this in more detail.

### Natural Language Processing

Lastly, the team underestimated the time and capacity that was needed to conduct natural language processing on the text data. Initially, when NLP was done using the CoreNLP package to extract the verbs and nouns in the reviews-column, there was an assumption that the processing could be done over a short period of time. However, after several nights of trying, the team found that each of the members machine had crashed before the processing could be completed. Upon further investigation, the team found that it was because as NLP was done continually, RAM was also constantly being used up. And once the maximum RAM capacity was reached in each of the machines, they crash without completing the entire NLP process. To resolve this, the team did the NLP section in parts and added a line to automatically save rows of texts which NLP had been done already done after every 20 rows were completed. This way, even if the machine crashes, the entire procedure need not be repeated from scratch. With this procedure, the team was able to extract the relevant words quickly.

## Solution Building Difficulties

### Cold-Start Problem

The team started on the project even before the first lessons of recommender systems started. As such, the relaxed criteria of not having to address the cold start problem was not known at first. As such, a lot of time was spent to try and resolve this. When the team learned during the second recommender system lesson that the project need not address the cold start problem, there was already a simple workaround that was thought about. As such, the team decided on continue with the work that was already done, instead of simply building a system that only recommends to existing users, which was deemed to be trivial and easy. For a new user, since there is no information that the model may use, the team would map the new user of the recommender system to existing user(s) who are defined as experts in the cuisine that has chosen. With these expert(s) as representatives of the new user, the models would then be able to use the ratings by these expert(s) to generate a list of recommendations, which could then be recommended to the new user.

## Content Based Model

The team was not comfortable with the approach of recommending restaurants using only cosine distance. This was because cosine distance of one restaurant that the user has been to only takes into the account the similarity of this restaurant with that of the ones in the choice universe based on their features. The ratings that a user has given to these restaurants are not taken to account at all. As such, simply using cosine distance could result in a system that recommends a restaurant with the features that the user actually dislikes. To resolve this issue, the team did independent research to see how we may incorporate the use of the ratings that the user has already given to the past restaurants that he has visited. The team decided on an approach using the implicit normalized score of the features of the restaurant to determine if a restaurant should be recommended. The methodology is discussed in more detail in the main report.

## Integrating NLP

Another challenge was to integrate NLP into the building of the models. This was a challenge because the main textual information from the team’s dataset was the reviews of the respective restaurant by each customer. The categories that each restaurant belonged to, which the team used as features, were already relatively structured. To resolve this, the team had to think about the contents that go into review texts. The team posited that in addition to the sentiment of the user, review texts may also include more specific elements of that restaurant that is not captured by the broad categories. For instance, the review text may talk about how good the ‘fried rice’ is in a particular Chinese-cuisine restaurant. But ‘fried rice’ would not be included in the categories that represent this restaurant. As such, the team decided that when the user interacts with the recommender system interface, there would be a text box, where he may input any keywords that would describe the specific elements of the restaurant that he would be interested in. Using this keywords vector, cosine distance metric could be used to filter down and enhance the choice universe of restaurants that the system would use to recommend from. This novel approach not only satisfies the criterion listed in the assignment instructions, but also enhances the performance of the recommender system because it is making use of more specific information regarding what the user wishes to eat.

## Collaborative Filtering Model

When building the CF model, the team initially could not decide on whether it was better to use the model based on user-user similarity, item-item similarity, or the model-based approach such as ALS taught in the final lesson. While the user-user and item-item models were easier to understand, the team soon realized first hand that scalability was a huge issue. This problem was especially important in the team’s case because of the sheer number of unique users (approximately 150,000) and items (2,000) in the dataset. With these considerations, the team decided that building a collaborative filtering model based on ALS would be the optimal choice. Moreover, the availability of the easy to use library ‘recosystem’ also made the evaluation process smooth. The team was also encouraged when it was taught that ALS was the go-to model used in the industry currently; and that its accuracy is generally better than the easier models that the team initially wanted to use.

## Hybrid Model

Since the team has built 2 models, it was able to combine them into a hybrid model. However, the team also recognizes that one model may be stronger than the other. There was a conflict as to whether to simply use the model which gives the lower error based on evaluation, or to still make use of both models since we know that a combination of results would be more likely to recommend the ‘true’ restaurant that the user would like since the models are making use of different kinds of data. After much discussion, the team decided on an error-weighted approach which marries both issues above perfectly. By weighting the models based on their respective evaluation, this method will ensure that the model with a lower error is given a stronger weight on what final list of restaurants are recommended to the user.

## Conclusion and Lessons learned

In conclusion, the recommender project that the team has completed was a combination of applying concepts and knowledge that was totally new, as well as injecting creativity as to how to integrate different elements such as Natural Language Processing into the recommender models that the team had in mind. While there may still be improvements to be made, I am extremely proud of the final product that the team has created. The team tackled any challenges during each step of the planning and model-building process using rational and logical thinking. Also, the team was able to conceptualize how the user would interact with the final product at a very early stage. Essentially, having this resolved meant that little time was needed to plan on how the recommender system would be built and significantly more time could be given to the refining of the model at each stage. I learned that team work and communication is very important. This is because not all members might have the same understanding regarding what was discussed. I hope to carry these knowledge and lessons to future classes, as well as my workplace so that I would be able to value add to any projects that may come.