Introduction

Rishil Shrinivas Parab

Background

A recommender system is an information filtering system that is designed to predict the preference a user would give to an item. The assumption is made that students would prefer subjects which are most similar to those previously they have been studied and have previous knowledge about it or recommended by someone. In the context of course selection, recommendations are usually made by looking at previous studies, grades and by getting recommended by the course coordinators of the particular department. In our recommender system design, we are going to evaluate the students liking towards the course, have they enjoyed studying the course or not, what are their grades like in every subject and would they recommend this course to others or not.

Design Model

Recommender systems are generally based on one of three models which use different filtering methods. Collaborative filtering models use the collaborative power of the ratings provided by multiple users to make recommendations. Content-based recommender systems use the descriptive attributes of items and their ratings. Knowledge-based recommender systems are based on the user's requirements, not on ratings. The knowledge bases contain the rules and similarity functions which determine the correlation between the user specified requirements and the courses that can be selected. Both collaborative and content-based filtering systems require a significant amount of data about courses done in the past and rating experiences.

Evaluation

Recommender systems can be evaluated under six broad concepts: utility, novelty, diversity, unexpectedness, coverage and serendipity (Silveira et al., 2019). Utility is related to the relevance, usefulness, recommendation value and satisfaction of the user. Novelty is a measure of how different the recommendation item is when compared to what the user has consumed. Diversity is considered a user-independent concept as it exclusively assesses the variety of items in a recommendation list. Diversity metrics only require the recommendation list itself and information about the items, therefore it does not depend on user information. Unexpectedness is defined as the divergence from the expected recommendations. Coverage evaluates the whole recommender system, not a recommendation list. Serendipity describes surprising recommendations which are pleasant and satisfying to a user. We choose to evaluate this Recommender System on serendipity and diversity.

Evaluating the serendipity of a Recommender System can be performed by asking the users to mark the recommendations that they find unexpected. If the user then followed these recommendations this would make them unexpected and successful and therefore serendipitous. It is important to check the effect of serendipity over time, because users might at first be intrigued by the unexpected recommendations and try them out. If after following the suggestion they discover that the recommendations are inappropriate, they may stop following them in the future, or stop following surprising recommendations.

As diversity is fundamental to the design of this recommender system, it will also be a component of the evaluation of this recommender system. Diversity is usually defined as the inverse similarity of recommendation lists between all the users. The diversity of a recommendation is therefore measured between suggestions made to different users. Here, however, we measure diversity between the target attribute and the recommendations made. To approximate the diversity of recommendations from the target keyword we use the attributes of the first recommendation as t_i, as we assume these are close to the requirements of the user. We average the distance of the first recommendation's attributes from the subsequent recommendations' attributes to obtain this diversity measure:

$$\overline{D}(T,X) = \frac{\sum_{i=1}^{n} D(t_i, x_i)}{n}$$

Broad ethical considerations

Daniel Rance

There are some significant ethical considerations at play in our field of work, with many different viewpoints and various requests for more ethical data management. Most notably these are the requests for gender equity, racial equity, and Maori data sovereignty. These issues are institutional, reinforced through centuries of patriarchal, colonial systems. Even though these are not issues specifically relating to (or stemming from) data science, the negative effects of these systematic biases are still felt within the field of data science and have the capacity to influence and potentially taint our statistical learnings. As responsible system developers, it is up to us that, not only is our system well intended from the outset, but it also has structural qualities which prevent it from reinforcing biases when used by future users with poor intentions.

While we, as the initial developers, may have good intentions regarding the output and use of the system we are developing, we cannot exclude sources of potential bias on our own part, and it is necessary that we fully understand the systematic biases at play and how they influence our own perspective and personal biases. It is also of great importance that we understand different cultural perspectives, preferably through direct consultation with different cultural groups. As Tangata Tiriti, we have an obligation to uphold the values of the Treaty of Waitangi, upholding and embracing Maori tikanga on an equal footing to western thoughts and values. In a system where a significant power balance exists between Maori and Pakeha, it is important that we take an *active* stance in addressing these issues and not be complicit in the reinforcement of neo-colonial values and practices. Māori Data Sovereignty is a nuanced subject, with differing values being held regarding the tapu nature of data and information. It is critical therefore we try to consult with all relevant tangata whenua and iwi whose information is stored and accessed by our recommender system. Further to this, we must ensure that we do not breach the grounds of our agreements with these parties by acting in bad faith or non-transparently. The terms of our data agreement may include points such as data storage, right to forget and restrictions on transmitting data across borders, out of Aotearoa.

There are more ethical considerations to be made outside of those relating specifically to Māori, specifically these are the issues of privacy and data anonymization - issues that relate to all New Zealanders regardless of their ethnicity. It is of vital importance that we do our utmost to ensure data privacy and anonymity for all users of our system. If we can do our best to understand and combat these institutional issues, we can influence and reinforce other organizations to uphold good data standards and refrain from compounding harm on already marginalized groups. Keeping these broad ethical considerations in mind, we must implement some concrete, algorithmic steps to reduce some of the institutional bias present in our system.

k-NN Function

Liam Gibson, 98217422

Why did we choose k-NN?

Once we had decided what our recommender system would do, we had to cut-down the multiplicity of forms it could take. This required us to choose a concrete implementation.

To recommend courses which a given student is not likely to take, we came up with several ideas. Notably, we considered randomly choosing a course which the student was eligible for, however, there are problems with this approach. Sometimes people would be recommended courses which are not ‘out of their wheelhouse’; sometimes they would be recommended courses which perfectly align with their current interests. This defeats the point of our project, and so, we needed a different solution.

How did we implement k-NN?

We decided, based on a homophily assumption, to compute the similarity between different students, take the inverse to find ‘dissimilarity’, and then randomly select a course from the pool of people who were considered ‘dissimilar’. This combines elements of randomness with network analysis.

For similarity to be calculated, we needed a way to compare students. This was done using three metrics per student per course. Grade, reported easiness and reported enjoyment. Using only three metrics allowed us to treat each piece of a data as a point in three-dimensional space. This opened up opportunities for data visualization and lead naturally to the concept of ‘distance’ between different students. This ‘distance’ gave us a measure of how similar students were. Students close together were more similar, students far apart were less similar. Of course, we can reverse this idea to get a measure of dissimilarity. That is, students who are far apart are considered more dissimilar, students close together are considered less dissimilar.

This choice was not a triviality. To build our system, we were forced to choose how to compute the ‘distance’ between two people. We made the decision to use the Euclidean norm, however, other norms would have given equally valid results. For example, the infinity norm. Given a data set, the infinity norm returns the supremum (or maximum). For most points in our data, this would increase the distance between people, implying a more conservative estimate of similarity, hence a more liberal sense of dissimilarity. Following this line of thought, we see that if we had used the infinity norm, the pool of dissimilar students would have increased (compared to the Euclidean norm), which would have decreased the ‘out of wheelhouse’ nature of our function. This is because people considered ‘similar’ under the Euclidean norm are now considered ‘dissimilar’ under the infinity norm.

To make this point concrete, imagine an engineering student. She uses a lot of mathematical approximations to arrive at good estimates of physical phenomena. Now, imagine there is another student, a maths major. She loves exploring the world of mathematics, proving theorems, playing with impossible shapes, all that jazz. These two students have the subject ‘mathematics’ in common, but how they think about mathematics is worlds apart. Perhaps our Euclidean norm would label these people as ‘similar’. This makes sense, they both enjoy maths, therefore, they are similar students. However, under the infinity norm, perhaps they are considered dissimilar. They do enjoy different aspects of mathematics, after all. If the infinitity norm was used, then we could recommend our engineer a pure maths course (or our mathematician an applied math course).

The point of this example is not to say, “one is right, one is wrong”. We are simply providing an example of an ethical dilemma.

One solution we discussed was to give the user the choice of norm. There are barriers to this approach. Not everyone has the mathematical comprehension to understand the full implications of choosing one norm over the other, but under the gauze of less technical language, or perhaps by giving an example like the one above, there is no reason why a user would not be able to choose for themselves.

As with the choice of norm, our choice of metrics was not neutral. Choosing to compute distance using metrics begs the question “Why did we choose those metrics?” and further, “What are the ethical consequences of these metrics?”

What are the ethical consequences of k-NN?

There are two driving questions which encapsulate some of the ethics we thought about. These are accompanied by a third question which explores some alternative ways of creating our system. We will state them now, then give some of our thoughts after.

(1) What are our metrics measuring?

(2) Does our system actively fight institutional bias?

(3) What are some alternatives to course metrics?

What are our metrics measuring?

Question (1) is motivated by the fact that we cannot directly access what a student will find easy or new and exciting. Therefore, we must do it by proxy. But, what if our proxies are measuring something else? A good example of this is the ‘grade’ metric. We chose to include ‘grade’ in our calculation, because we made the assumption that a good grade implies an easy course. However, this is a large assumption. There are many reasons why a student may have received a high grade on a course. Perhaps they found the course challenging, applied themselves, and received a good mark. In this case, a good grade represents the opposite of what we assumed. Another consideration to make is, what is not captured in our data? What are the environmental factors driving a student's performance? Is an A+ from a student who works a full-time job on top of university equivalent to an A+ from a student who does not work at all? Answers to these questions put us in a difficult position when trying to defend that grade faithfully represents ‘easiness’. In discussing these problems, we considered adding another metric to our line-up: ‘workload’.

This could be used in conjunction with ‘grade’ to give a more comprehensive view of ‘easiness’. Prima facie, this comes with some downsides. Namely, with three metrics, we have the opportunity for all sorts of interesting visualizations in three-dimensional space. These would become harder if we added another metric. However, it’s not too hard to see ways in which we could combine ‘grade’ and ‘workload’ into a composite metric. For example, we could define ‘easiness’ like so,

$$\text{easiness} = \frac{\text{grade}}{\text{workload}} \tag{1}$$

Therefore, easiness is directly proportional to grade and inversely proportional to workload. This has its own set of ethical concerns. To name just a few: Does easiness vary linearly with grade and workload? What if easiness is some other function of the two? Are there more factors which we must take into account? Both ‘grade’ and ‘workload’ could have equity issues: it’s not hard to see socio-economic status influencing both. As interesting as these questions are, for the sake of brevity, we'll shelve them for now. Instead, we will move on to question (2).

Does our system actively fight institutional bias?

In short − yes and no.

Primarily, we thought about institutional bias in the context of gender equity. This is because reported easiness and enjoyment are both gendered concepts. A 2018 study by Cambridge researchers Mitchell and Martin found that students rated the same course more favourably when it was taught by a man, as opposed to a woman (Mitchell & Martin, 2018). And, while it has been contentious in the past (Centra & Gaubatz, 2016), more recent studies suggest that teacher evaluations are not gender-neutral either (Mitchell & Martin, 2018; MacNell, Driscoll & Hunt, 2015). If you assume that the teacher has an effect on the student's perceived enjoyment or easiness, then this presents more evidence to the same position that these are gendered concepts.

For these reasons, we chose to compare the relative ratings given by students, rather than the magnitude of the ratings themselves. This corrects for the most vulgar forms of sexism. However, it has problems. It assumes that female and male students rank female and male teachers in the same way. Studies (Basow & Silberg, 1987; Boring, 2016) have shown that this tends to be false; there is an interaction between student's gender, teacher's gender and teacher's behaviour.

Therefore, relative ranking still shows problems, but when working with the 'out of wheelhouse' system, this isn't as much of a problem. This is because each student-slash-course which we mark as 'dissimilar' is assigned a uniform probability of being chosen. This flattens the hierarchy which is usually present in ranked lists given by recommender systems.

Additionally, working with ‘out of wheelhouse’ helps alleviate ethical concerns regarding positive feedback loops.

To see why positive feedback loops could be harmful, consider the following example. Imagine that our recommender system produced a hierarchy of courses, the algorithm recommends lots of people the same course, which in-turn causes more people to have common experiences of the course, increasing its rank, causing the system to recommend it more, etcetera ad infinitum. Luckily, our ‘out of wheelhouse’ recommendation does not produce a hierarchy, it produces a uniform probability distribution. This is not true for the ‘easy mode’ recommendation.

Because the easy mode does not use ‘dissimilarity’ and instead produces a ranked list, we have to deal with the issues mentioned above. Namely, because courses taught by men, and areas dominated by men, are rated higher than female taught courses and female-dominated areas, our algorithm could reinforce these biases in positive feedback loops.

One solution could be to explicitly adjust weights in order to counteract the discrimination. This is great in theory, however, the value for these weights would vary in space, they would vary in time; there is no obvious way of measuring them, so we would have to use more proxies; this introduces more ethical complications, plus, the weights would vary with our research methodology: it gets very messy, very quickly. This brings us to question (3).

What are some alternatives to course metrics?

We have seen some of the ways in which course metrics are limited. This prompts us to propose an alternative. One notable alternative (which was discussed after the main project had been completed) is using *demographic* metrics instead of *course* metrics. This would classify people, not based on their course ratings, but on their demographics, i.e., age, gender, ethnicity, etc…

If implemented correctly, this would have the effect of mixing people with different socio-economic backgrounds. This gives 'out of wheelhouse' an entirely different meaning. Instead of being introduced to new disciplines, the user would be introduced to new people.

Being optimistic, thinking big-picture, and following this style of recommendation through to its conclusion, the best possible outcome would be increased diversity in the workforce and in academia. However, with grand assertions like this come many assumptions. Among other things, we are assuming that an increased diversity at the course level would carry over into an increased diversity at the graduate and post-graduate levels. This is not necessarily true. There is a well-documented phenomenon in STEM subjects commonly referred to as 'the leaky pipeline'. As we work our way up the STEM hierarchy, we see less and less women. This suggests (among other things) differences in attrition rates between women and men (Xu, 2008). UC is no exception, with men about twice as likely to be ranked professor, compared to women (Brower & James, 2020).

Therefore, sadly, there is no silver bullet to be found in demographic data. However, it is still an interesting mode of operation for our recommender system.

Conclusion

In summary, we have explored how k-NN can be used to compute the similarity between different students. With some further investigation, we found the k-NN framework to be theory-laden, holding within it a whole world of ethical considerations.

In the next section we will consider how our system makes a recommendation and explore some of the ethical implications.

References

Basow, S. &. (1987). Student evaluations of college professors: Are female and male professors rated differently? *Journal of Educational Psychology*, 308-314.

Boring, A. (2017). Gender biases in student evaluations of teaching. *Journal of Public Economics*, 27-41.

Brower, A. J. (2020). Research performance and age explain less than half of the gender pay gap in New Zealand universities.

Gaubatz, J. C. (2016). Is There Gender Bias in Student Evaluations of Teaching? *The Journal of Higher Education*, 17-33.

Lillian MacNell, A. D. (2014). What's in a Name: Exposing Gender Bias in Student Ratings of Teaching. *Innovative Higher Education*.

Mitchell, K. &. (2018). Gender Bias in Student Evaluations. *Political Science & Politics*, 648-652.

Xu, Y. J. (2008). Gender Disparity in STEM Disciplines: A Study of Faculty Attrition and Turnover Intentions. *Research in Higher Education*, 607-624.

Code

Below is a link to a GitHub repository:

github.com/dot-jpgg/k-NN-For-Course-Recommender-System

This contains the code used to create the k-NN function. The majority of the code was written by myself (Liam) and several functions were written by Tessa.

DATA417 The Chosen One

Brett Ellis

Selecting the courses

The initial idea for choosing how to decide which courses to recommend was to use all previous students who have done a given course to calculate a *rank* value which would represents how *good* a course would be for a student. This would allow us to simply choose the top n courses to recommend. Each student would have an associated weight given to them based on some similarity measure to the student whom is being considered.

$Rank = f(metric_1 * x_1 + metric_1 * x_2 ... + metric_1 * x_i + metric_2 * x_1 ... + metric_2 * x_i ... + metric_j * x_1 ... metric_j * x_i)$
where x is the weight for the students compared to student being evaluated

We decided against using this model for multiple reasons. Trying to find a way to calculate *rank* was found to be to difficult as what is considered *good* for a student is different for each given student. The model also assumes homopholy, that a student is somewhat comparable with every other student and whatever courses that should be recommended can be measured on every other student, we decided that it would be not only be easier but better to take a subset of students that are considered *similar* and use these students to make a decision on what courses to recommend. By comparing different students that model is making an assumption of homopholy, that there are students that are inherently similar and that can be used to predict future behavior, however the model that we end up using also makes this assumption. It is almost impossible to make a recommender system without the assumption of homopholy, the only way to do this would be to recommend courses completely at random which is not very useful.

In the end we decided on using a kNN model. Below is the code that is used to recommend some courses with both the metric average function (MA) and the out of bag function (OOB), as seen below.

The Chosen One Data Set

The input that is used here is a csv that contains all students, their courses done, their associated metric values for each course (in this case it is grade, enjoyment and easyness) as well as the similarity value that had been calculated earlier. This section of the code simply reads that csv into a dataframe and removes the student of interest or the *Chosen one* as they would be perfectly similar with themselves so would end of getting recommendations from courses they have already done.

```r
library(tidyverse)
library(dplyr)
metric.df <- read.csv("metric_df_2.csv")
metric.df$Student_id <- as.character(metric.df$Student_id)
metric.df <- metric.df %>%
  filter(Student_id != c("1"))
```

kNN function

The kNN function has a k input which indicates how many of the most similar students you want to have in your output, it also uses a dataframe that is like the one from the first block (student_id, course, metrics, similarity). The output of this function is a dataframe like the one that is used in the input except it only has the top k similar students and all there associated values.

```r
KNN <- function(k, sim.df){
  sim.df1 <- sim.df %>%
    group_by(Student_id) %>%
    summarise(sim = mean(Similarity))
  sim.df1 <- sim.df1[order(-sim.df1$sim),]
  top_students <- c(sim.df1[1:k, 1])
  output <- data.frame()
  for (i in 1:k){
    student <- sim.df %>%
      filter(Student_id == top_students$Student_id[i])
    output <- rbind(student, output)
  }
  return(output)
}
```

MA function

The Ma function is the first function that actually returns courses that will be recommended for the *chosen one*. The functions inputs are k, the number of courses that you would like to be recommended. Metric, this is the metric that you are evaluating the courses on, this will be a number from 1 to n with n being the number of metrics. In this example there are three metrics, grade which is denoted by a 1, enjoyment which is denoted by a 2 and easyness which is denoted by a 3. The last input needed is a data frame that contains the students that you are comparing to the *chosen one* and there courses and metric results. The function then calculates the means for each metric and orders them from highest to lowest based on the metric that was chosen. The function then returns the top k courses to be recommended.

```r
top_courses <- function(k, metric, sim.df){
  sim.df <- sim.df %>%
    group_by(Course) %>%
    summarize(Grade = mean(Grade), Enjoyment = mean(Enjoyment), Easyness = mean(Easyness))
  sim.df <- sim.df[order(-sim.df[1+metric]),]
  courses <- sim.df$Course[1:k]
  return(courses)
}
```

OOB function

The OOB function is the other function that returns courses to be recommended, but unlike the MA function this function aims at returning courses that are taken randomly form a reduced subset. The functions inputs are k, the number of courses you would like to recommend. n, the amount of courses you would like to drop from the top and bottom. The last input is a dataframe that has all the similarity values for students as well as the courses they have taken. The function works by removing the n most and least similar students from the dataframe, this means that you must have more then 2n students in your dataframe. The function then removes duplicate courses and takes a k sized random sample of courses that are left, this means that you must have more then k unique courses in your subgroup. Although there are some restrictions based on your n and k values, the datasets that are being used will tend to be very large so unless extreme values are used for k and n this should not be a major drawback for the function.

The reason that we decided to drop some of the most and least similar students is because of what kind of courses we wanted to include in our output. We did not want to recommend courses form students that were very similar as these might have already been recommended to the student. We also did not want to include extremely dissimilar students as the courses from these students might have been to different from what they were doing.

```
OOB_courses <- function(k, n, sim.df){
  sim.df <- sim.df[order(-sim.df$Similarity),]
  sim.df <- head(sim.df, -n)
  sim.df <- sim.df[-(1:n),]
  sim.df <- sim.df %>%
    group_by(Course) %>%
    unique()
  courses_OOB <- sample(sim.df$Course, size = k)
  return(courses_OOB)
}
```

An example of the functions

Using the data provided, an example of these functions were done below using a dataframe with the intended structure. The functions are designed in such a way that you use the knn function to make a dataframe that can then be used for both the MA function and the OOB function as seen below.

```
head(metric.df)
```

```
##   Student_id  Course Grade Enjoyment Easyness Similarity
## 1          2  ENG302     3         3        3          0
## 2          2  LIT123     1         0        0          0
## 3          2  MUS210     3         5        8          0
## 4          2 MÄ\200101     1         0        0          0
## 5          2 MÄ\200102     1         1        1          0
## 6          2 MÄ\200103     1         2        3          0
```

```
knn.df <- KNN(10, metric.df)
top_courses <- top_courses(5, 1, knn.df)
random_courses <- OOB_courses(5, 10, knn.df)
top_courses
```

```
## [1] "CS221"  "MAT120" "MUS323" "PSY202" "CS111"
```

```
random_courses
```

```
## [1] "PSY202" "CS321"  "CS321"  "CS112"  "CS210"
```

```
print(":) It works!")
```

```
## [1] ":) It works!"
```

Limitations and Ethics

As this was a proof of concept and we only had limited time there were components of our functions that we would have looked to improve and optimize as there are clear limitations of these functions. One of the limitations of the functions is that any course can be recommended to the student regardless of year. This is big drawback of the design as this means that the *chosen one* could get courses recommended to them that they have either already done or are for a year that they do not have the prerequisites for. These recommendations are useless as they are courses that the student cannot take. With more time the code could have been adapted such that courses that the student is ineligible for would be removed from the dataset so that they could not be recommended.

Another limitation of the function is that there is a lot of information that is not used in the selection of courses and similarity. If we had more time to work on the algorithm one thing that we would want to include would be student demographics. Using other variables such as race and age in the algorithm would allow us to calculate a possibly more accurate group of students that are more similar and would therefore be expected to have similar metric results, however again this rides very heavily on the assumption of homopholy.

One thing that would have a lot of ethical implications is whether or not to display the means from the MA function in the output. The problem with displaying the means is that the user might take these values as gospel and assume that the *chosen one* would get these same metric values for the courses that are recommended to them while in actuality this is unlikely. We decided that it would be appropriate to display the means in the output, however the values would be displayed in the form of a confidence interval, a range in which we would expect the value to be between. This was another thing that would be implemented given more time to work on the algorithm. The confidence intervals would be difficult to construct but the most likely way to do this would be using either just the sub setted courses that are used in the MA function or it would be built on all of the metric outputs for that given course. If the latter idea was used this would mean that the confidence intervals would be based on the course itself and independent of the student. If confidence were implemented we would need to see the difference between these two methods to make a decision.

Another problem that is both a limitation and an ethical concern is knowing how well the algorithm is actually working, as in are the courses that are being recommended to the *chosen one* actually useful? We had some ideas on how this would be achieved but again as we were limited on time none of these ideas were tried. The most common way to test an algorithm is to see how accurate a prediction is but this is not so easily done for a recommender system as what is considered *accurate* is not trivial. The way that you would decide to test your algorithm somewhat comes down to how you want define accuracy. Our initial idea was to simply see if the course that were recommended to the student from the MA function were similar to the ones that were predicted from course adviser, this would defining accuracy as predicting expected courses. Assessing the algorithm like this is not very useful, defining the algorithm as accurate when it predicts the same courses causes it to be somewhat redundant as the main idea is that we can recommend courses that the course adviser would not have considered. Another clear draw back of this method is that you cannot really assess the accuracy of the OOB function as course advisers do not recommend random courses which means there is nothing to compare it to. Another method of testing the algorithm which I would recommend but is somewhat difficult to do is to follow up on students whom have been recommended courses. If the students that has been recommended courses are asked how they enjoyed the courses etc than this information could be feed back into the algorithm and assess whether or not the predictions were considered good or not. This method would be testing accuracy as a measure of how the student did in a recommended course. A clear advantage that this method has over the first is that you can test how well the OOB function performed as the way you would test accuracy from the MA function and the OOB function would be the same. Even just the metrics would be very useful as this would allow confidence intervals to be made that would be better as before they would have been built based on all sutdents who have done a given course while now they could be built based on students who have been recommended the course by the algorithm. Having this kind of method for testing accuracy allows the model to get better each year that it is implemented, this also comes with the drawback of not being able to test accuracy that well untill it has actually been implemented for a few years. There is also the risk of a feedback loop occouring, if a course happens to be recommended a little bit more than other courses then more infomation will be given for that course causing that model to become biased for that course, this an issue that would need to be considered for future development.

In the next section of the report we will discuss how the courses will be displayed.

DATA417 report section

Tessa Grant

This part of the report will discuss the user interface design as well as the implementation of our recommender system. As an initial note, the user interface has not been implemented in the code of our project, as it is not essential to creating a working a proof of concept. In the place of this a sketch of a possible user interface has been made and will be included in the references of this section. In particular this sketched interface will be the results printout that is available for both the course coordinator and the student.

First I will discuss the implementation of our recommender tool in the university community. The first step of implementation is to supply the tool to the course advisors, at this point previous data (student grades and survey responses) will be requested to act as an initial training data set. Additionally, the course advisor will be given a lesson on how the tool works, going over the decisions encoded in the algorithm at a level that suits that specific advisor to ensure they have an initially strong understanding of the decisions that the algorithm carries out. A key point that must be emphasised to the advisors is the fact that the recommendation given by the algorithm is not gospel, and should be analysed in a critical manner to ensure the best outcomes for the student as they are the ultimate beneficiaries of our project. As a final note regarding this, it is important that we make ourselves available in early stages of implementation so that any concerns or questions about the algorithm can be heard and discussed, with any issues coming from this discussion being addressed in a timely manner.

This results printout (the element of the user interface which has been represented as the sketch) is split into four major sections. Perhaps the least important section is the user info recap in the bottom left corner of the printout. This interface element reiterates the user’s information that is fed into to the algorithm and used in calculating the recommendation. The purpose of this is just to ensure that the correct information has been fed into the algorithm. The top left and centre left sections present the top ten recommendations for ‘easiness’ and ‘out of wheelhouse’. The number ten was an arbitrary choice and could be easily altered to present more or less results depending on what was appropriate. The final, and most essential, element of the results printout is the flow chart found along the right hand side of the printout. The purpose of this flowchart is to detail the process the algorithm follows on a very high, and basic, level. To quickly summarise contents of this flow chart, the chart details how the algorithm calculates which students are most like you, then explains the logic behind the ‘easyness’/‘out of wheelhouse’ decisions. The purpose of this very broad generalisation of the algorithm’s process is to inform the student on how the algorithm works so that they understand the process in the interest of transparency (transparency is an important ethical concept that will be discussed in a more in depth manner later in this section).

The second component of the results user interface (which hasn’t been graphically represented) is a spreadsheet of statistical measurements. These measurements include values such as the mean, confidence interval etc. This component of the user interface is restricted to only be available to the course advisor.

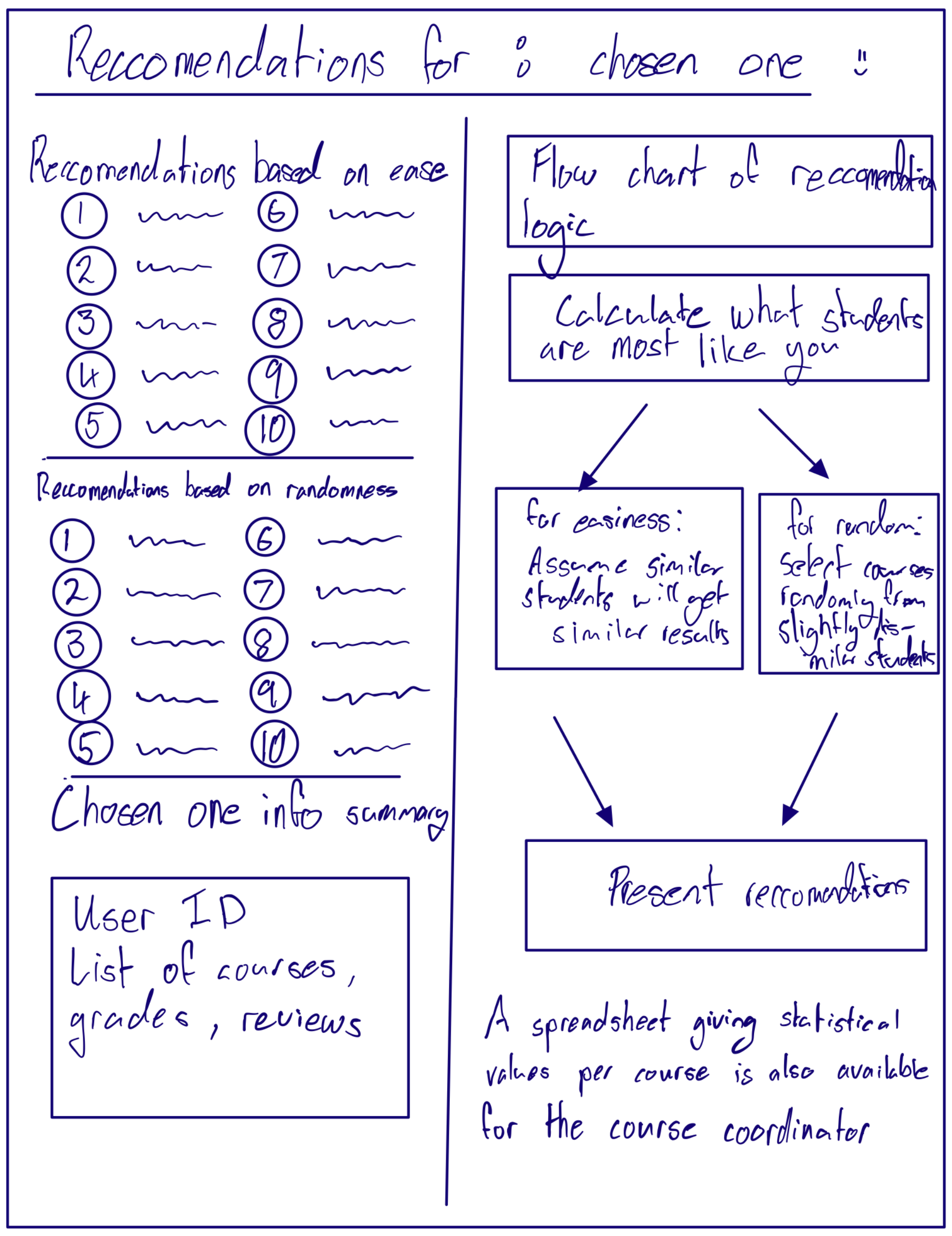
Making algorithms transparent is an important ethical request directed at all decision making algorithms. To briefly step outside of the scope of our recommender system, it seems appropriate to an example where a non-transparent recommender system has the potential to create ethical issues. Our lecturer (Giulio Dalla Riva) synthesised an example algorithm that had the function of recommending to a bank whether a loan applicant should be approved for their loan. Within this algorithm was encoded racial prejudice, with the bank being much less likely to approve of a loan when the applicant was of a specific ethnicity. The code behind this algorithm was not available to the applicant, and therefore was not transparent. For someone only capable of seeing one recommendation given by the system (such as an individual applying for a loan) there is no way to notice a prejudice like this. The lack of transparency with the system made this unethical prejudice un-detectable.

To circle back around to our particular recommender system, the purpose of the flow chart detailed previously is to provide some level of transparency for the student as to how the decision is being made, allowing them to take this information into account when considering how much credence they wish to place with our recommendation. Choosing a university paper to take as a student is an incredibly important decision. If a student chooses an incorrect course they have wasted a large amount of money and a large amount of time, as such it is important that our recommender system does not lead people astray. The statistics spreadsheet gives a deeper level of insight into the decision making process, allowing the course advisor to assess how confident the recommendation is. The fact that this spreadsheet is not available to the student was a deliberate decision. This ties back into the fact that our recommendation is not gospbel, it won't always be perfect. By making the spreadsheet only available to the course advisor we introduce a human intermediary between the recommendation and the student. The course advisor is able to mitigate issues that may occur when the confidence intervals are either particularly low or high.

The negative outcomes associated with a student over-emphasising the importance of a recommendation with a low confidence is obvious, the course may simply not be as good for them as suggested. The danger of an overly-confident recommendation is more subtle. Consider the scenario where a student has access to the spreadsheet and they see a recommendation with a high confidence interval. The student may place too much faith in this recommendation, potentially choosing to take the course without enough consideration. By having the spreadsheet available to the advisor they are able to inform the student when a recommendation is weak or not overstate the recommendation when the confidence is strong. Having the course advisor as the intermediary between the data available in the spreadsheet and the student is critical.

As a final point within this section I would like to touch upon future improvements that could be included in the user interface. The major improvement that comes to mind is incorporating a greater number of graphical representations, both on the result screen available to both the course advisor and student, and also accompanying the spreadsheet that is available to just the advisor. The key benefit of this is to make the information easier to understand for students/course advisors who do not necessarily have a background in statistics or data science. A second advantage of this is that it forces us as designers to look at the data in different ways, potentially revealing flaws in the system or further areas for future improvement.

Result Printout Sketch:


Reccomendations for : chosen one :)
Reccomendations based on ease
1 2 3 4 5 6 7 8 9 10
Reccomendations based on randomness
1 2 3 4 5 6 7 8 9 10
Chosen one info summary
User ID
List of courses, grades, reviews
Flow chart of reccomendation logic
Calculate what students are most like you
for easiness: Assume similar students will get similar results
for random: select courses randomly from slightly dis-milar students
Present reccomendations
A spreadsheet giving statistical values per course is also available for the course coordinator

What's Next?

Daniel Rance

1. **Hard Mode**
   In terms of areas of expansion for our recommender system, one of the first pathways we could follow is that of implementing a 'hard mode' recommendation for our system. This addition would give users the option to be recommended a course that will be challenging for them, alongside our already implemented recommendations for easy and 'out of bag' courses.

2. **Year-level specific recommendations**
   Another area of our recommender system which could be expanded upon further is 'per year recommendation'. Often a student looks for a course at a specific year level in order to satisfy some conditions of their graduation. Our recommender system currently does not have the functionality to recommend course based on their year level (100, 200 etc), so logically, this would be one of the first steps in further development. Theoretically this would be a relatively straight forward adjustment to our recommender system. If we simply split user datasets (course grade, enjoyment etc) and course datasets (course code, school etc) by their respective year levels, we would be able to make year level specific recommendations to potential users of our system.

3. **Further work to reduce systematic bias in our system**
   It should be clear that we will never be able to completely eliminate the effects of institutional bias on our statistical learnings. Despite this, we must remain vigilant and cautious of any potential harm being cause by our recommender system. We should strive to have open and transparent communication lines between ourselves and the community and use their input to guide the future direction of our project.