**CCT College Dublin**

**Assessment Cover Page**

|  |  |
| --- | --- |
| **Module Title:** | Data Visualization Techniques, Machine Learning for Business |
| **Assessment Title:** | CA2 |
| **Lecturer Name:** | David McQuaid, Muhammad Iqbal |
| **Student Full Name:** | Arthur Claudino Gomes de Assis |
| **Student Number:** | 2023146 |
| **Assessment Due Date:** | 01/12/2023 |
| **Date of Submission:** | 05/12/2023 |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

# TOTAL WORDS ON REPORT: 2154

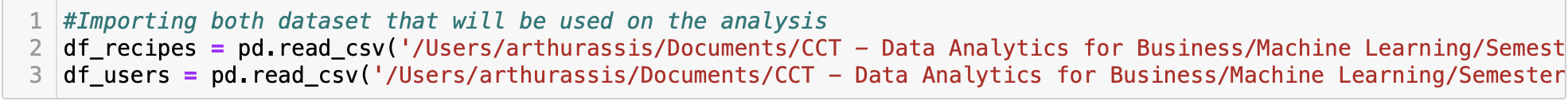
# INTRODUCTION AND BUSINESS UNDERSTANDING

Retailing is the selling of goods and services to consumer end users, according to (Montevirgen, 2023). Nowadays, the acquisition of data from users permits companies to achieve a certain level of knowledge about people navigating through internet that has made possible to them to create a customized experience to each person, turning them in potential customers. This experience comes often through advertising, customized according to each person personal data collected from diverse sources, including navigating history, previous online purchase activity, and even voice data collected from our mobile devices.

To use this data in an efficient way, retailers use Machine Learning models to understand peoples preferences, and recommend products according to behaviour and to their personal data. These Machine Learning models are called Recommendation Systems.

The dataset that will be explored in this project can be found in the link <https://www.kaggle.com/datasets/shuyangli94/food-com-recipes-and-user-interactions/data?select=interactions_validation.csv>, and it refers to data collected from a website that provides culinary recipes to users, in a format of a blog. The dataset was chosen because it contains similar features to retailers. Although it does not aim to sell its recipes, blogs are one of the sources targeted by companies to expose advertising. According to (Mimi Polner, 2023), blogs can be used to bring in sales or an income, and bloggers bring in an average annual income of $37,073. Some of the ways explored by the article that brings income to bloggers are Advertising Networks, Digital Products, Affiliate Links or Codes and Premium Content. We can therefore, consider blogs as a direct advertising retail business, in which the blogger aim to sell space for advertising in its website, and the quality of the content and experience will bring in more viewers, potentially increasing profits.

When we first verified our dataset, we noticed it contains 231,637 recipes! That may generate confusion on users that might head towards a more classified source of information. This is when recommendation systems come to action.



Words counting: 321.

# Data Cleaning

A screenshot of a computer program

Description automatically generated

We can verify that our dataset contains 231,637 observations and 12 features. It seems that the data is nearly complete, except for 1 recipe that does not have name, and around 5,000 that does not have any description. We can drop those observations once they represent something around 2% of our dataset.

We also verify that our dataset has a date feature, that is read as an object, and it will be converted into datetime object to help in future analysis.

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

It is possible to verify that in this case we have over a million observations and 5 features. Again the data is quite clean, and only less than 200 fields on reviews are null, what represents a negligible parcel of the whole, and there it will also be dropped. We will also convert the date column in this case to a datetime object.

We will now verify the number of times each recipe was reviewed.

A graph of a recipe

Description automatically generated

We could verify that among our whole dataset, that contains over 200,000 recipes, only around 1/5 contain more than 5 reviews. It will be considered relevant in this dataset only the 40,000 most reviewed recipes. The main reason is that while performing the methods required on the scope of the project for recommendation systems, the computer used could not perform the operations with more than 40,000 observations of recipes.



We understand that the number of reviews refers to engagement in one specific page of our website. Therefore, using the information that we have on our dataset, if we are able to discover what increases the number of reviews, we can optimize our website to increase its views, and therefore sales.

Words counting : 276

# Building a content based recommendation system

A content based recommendation is a system based exclusively in products characteristics. It bases its recommendations in the similarity existent between two products.

In our case, we will build a recommendation engine based in two features of our dataset: tags, and ingredients.

The first step is parsing our string to get a list of values. We can using the function literal\_eval that reads the line as though it was a code, and extracts the meaning of it.

We then parse our data once more removing inconvenient information that may lead our model to bad results. (Leitch, 2020) discussed the issue of the ingredients for a recommendation systems of recipes utilizing a different dataset. He discusses that some words on our ingredients do not contribute to our recommendations, because they are very common in almost every recipe, such as oil, weights and measures. He proposes a parser for his own project that will be applied in our dataset. The parser was found on his Github.

A screenshot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

We can verify through the examples above, that the first tags are usually very similar among themselves, what means that getting the first tags it is not the best idea, therefore, we will select 5 tags after the sixth one.

We can create now the metadata soup that will be feed in our vectorizer.

Our Vectorizer has found 2857 vocabularies used on the dataset according to the features we have used.

The next step is building a reverse mapping to get the name of our recommendations.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

As we can verify, we can find using our content recommender system recipes that are alike the one searched. Data for the searching engine could work as a link in a website. If one click on the recipe for dark chocolate cake, it will receive in some fields on the screen recommendations to related recipes.

Words counting: 305.

# Creating a Collaborative Filtering

Collaborative Filtering, different of content recommendation systems, do not use products data to find similarity among products. Instead, it uses data from different users, and based on what users like or dislike, the recommendation is made. There are two mechanisms on Collaborative Filtering that can be used to create recommendations, the user-based and the item-based.

Both of them are based in the similarity of users previous actions, rather than the content of the product. For instance, in our business case, we may discover that users that have rated chicken based recipes, have also rated nuts based recipes. Once a new user access the system and rate by the first time a chicken recipe, our recommendation system will recommend nuts based recipe, rather for the similarity between users preferences than for the similarity between the recipes itself.

We will in this project build an user-based recommendation system.

In the first step, we want to filter back in our interactions dataset, only reviews that were written to recipes in our list of relevant recipes.

Verifying our new dataset, we can see it still contains over 700,000 observations and nearly 200,000 users! Due to computational costs, we will need to filter only the 7,500 top of our website.

A screenshot of a computer code

Description automatically generated

Now that we reduced our dataset only to the top 7,500 users, we will apply the methods to seek for recommendations for a certain user, based on its similarity to other users.

The methods is based in a few steps:

1. Finding the average user in recipes they have reviewed.
2. Calculating an adjusted average for each user and recipe based in its average and its rating to the recipe.
3. Creating a matrix of users x recipes, and its adjusted average on the values.
4. Filling all recipes unreviewed by each user with an average of the adjusted average for the user.
5. Finding the similarity using this matrix through cosine similarity.
6. Sorting similarity for each user, and mapping index into recipes names.

After performing this process, we get the following results to a random user 353579.

A screenshot of a computer

Description automatically generated

Let’s see and compare some results we get when using the content based recommendation system for two of this recipes, and compare with another recipe provided by the recommendation.

A screenshot of a recipe

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

It is possible to verify in our example, that when we get recommendations based on content for potato kielbasa skillet, we get recipes that are similar to the first, taking the example of fried potatoes and smoked sausage, we can see they apparently have similar ingredients and they are tagged with similar tags. On the other hand, we do not find anything similar between the two first dishes recommended for the user 353579 as expected.

Words counting: 448.

# Market Basket Analysis

It will be used in our case, the list of ingredients for recipes as our basket. It is interesting to analyse this scenario because companies that sell professional products for chefs or restaurants may benefit from the analysis verifying which kind of products are necessary together in order to produce certain food.

A screenshot of a cell phone

Description automatically generated

We can verify that our most consumed ingredients are basic in any recipes, to season, and they are not products with a high value of market.

Using the algorithms that perform the Apriori principle in our dataset, we get the following results.

A screenshot of a computer

Description automatically generated

Still using the Apriori principle, we get the following association rules in our dataset.

A screenshot of a computer code

Description automatically generatedA table of numbers with text

Description automatically generated

Using the algorithm performing FP Growth, we reached the following results.

A screenshot of a computer

Description automatically generated

We get the following association rules using the algorithm FP Growth.

A screenshot of a computer

Description automatically generatedA table of numbers with text

Description automatically generated

Both algorithms identify well items that are usually used together in recipes, however, it seems that APriori gets more range in regards to ingredients. When we verify the association rules, APriori identify a higher variety of items, whereas FP Growth identify many time the same products in different combinations. The decision of which one of them is better would be possible only analysing the necessities of the business.

As we are dealing with recipes, most of the rules that we may find here are likely obvious, because items cannot be combined randomly when cooking, however, we can still use this information in formatting marketing strategies, and understanding preferences of people who access this website.

Words Counting: 244.

# Building a Dashboard to visualize the data

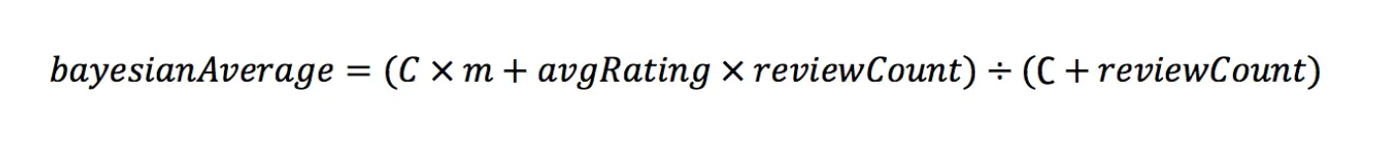
When it comes to the business we are exploring on our data, this is, using spaces in a culinary page to generate sales, and therefore income, some metrics are important to consider, the number of views on a certain page and the engagement of a user on the page. For instance, a recipes blog could sell space for advertising to grocery stores, that could explore this space to offer products related to the recipe that is being seen by the user. In this example, all the 4 factors listed above would increase the value generated for the grocery store when using the space.

In the case of our dataset, what we have in regards to these factors are n\_reviews, rating and the reviews itself, that could be parsed to deliver a more accurate sentiment analysis than just rating. Watching this factors, we can analyse how some variables influence on the, such as minutes, n\_steps and n\_ingredients.

We will build a dashboard that deals with this variables, and that can deliver the information of how to generate value to the blog through recommendation systems explored before.

We will use Bayesian average as suggested by (Bengfort, 2017) that discussed its use and apply according to (Odabaşı, 2019) that shows a practical example how to calculate it using R.

The Bayesian average is calculated through the formula below, in which m is used for the prior value of average ratings and C is used for confidence in the prior values. Following the suggestion of (Odabaşı, 2019), I will use average of rating by recipe for m, and average number of reviews by recipe for C.



Given the initial ideas, I decided to create an interactive graph in which I can select among the variables stated, and visualize the distribution of the data, number of reviews vs variables, and Bayesian average, averaged by the variable vs the variable. As we could see in some graphs above, some features contain data varying widely, so it was decided to add the possibility to change between Linear Scale or Logarithmic Scale on the graph, making possible to observe better a wide range of values. It was also included on the graph the possibility of adjusting the length of the x-axis, so that it is possible to select parts of the graph that might be triggering in an analysis, and visualize it closer.

In regards to the target for the graph being people over 65, font sizes were increased, especially on titles, the design was decided to be kept simple, white background and dark blue colors, presenting high contrast, and not too much information on the graphs, just what is essential, in such a way that information is not overwhelming.

A screenshot of a computer

Description automatically generatedA graph on a white background

Description automatically generated

It is possible to verify for example that the majority of our reviews are given in recipes that take under 60 minutes to finish, approximately 8 steps, and no much more than 10 ingredients to get ready, therefore, spaces for advertising in recipes with these features are more likely to engage customers.

An observation in regards to the algorithm of the dashboards itself is that the library used to produce the graphs has an issue in regards to logarithm scales on histogram, and therefore, when changing to logarithm scale, no error will arise, however no graph will appear. This issue is being discussed in these links <https://github.com/plotly/plotly.py/issues/2899>, <https://github.com/plotly/plotly.js/issues/6200>.

Words counting: 560.

# References

Bengfort, B., 2017. *District Insights.* [Online]   
Available at: https://medium.com/district-data-labs/computing-a-bayesian-estimate-of-star-rating-means-651496a890ab  
[Accessed 01 12 2023].

Cloud, S., 2023. *SaturnCloud.* [Online]   
Available at: https://saturncloud.io/blog/how-to-select-rows-from-a-dataframe-based-on-list-values-in-a-column-in-pandas/#:~:text=To%20select%20rows%20from%20a%20DataFrame%20based%20on%20a%20list,to%20select%20the%20desired%20rows.  
[Accessed 29 11 2023].

Coppola, D., 2023. *Statista.* [Online]   
Available at: https://www.statista.com/statistics/678153/amazon-ad-cost/#:~:text=Since%202014%2C%20Amazon%27s%20annual%20advertising,dollars%20in%20the%20previous%20year  
[Accessed 30 11 2023].

Leitch, J., 2020. *Towards Data Science.* [Online]   
Available at: https://towardsdatascience.com/building-a-recipe-recommendation-api-using-scikit-learn-nltk-docker-flask-and-heroku-bfc6c4bdd2d4  
[Accessed 28 11 2023].

Li, S., 2019. *Kaggle.* [Online]   
Available at: https://www.kaggle.com/dsv/783630  
[Accessed 22 11 2023].

Mimi Polner, C. B., 2023. *Forbes Advisor.* [Online]   
Available at: https://www.forbes.com/advisor/business/start-a-blog/  
[Accessed 11 29 2023].

Montevirgen, K., 2023. *Brittanica Money.* [Online]   
Available at: https://www.britannica.com/money/retailing  
[Accessed 30 11 2023].

Naveen(NKK), 2023. *Spark by {Examples}.* [Online]   
Available at: https://sparkbyexamples.com/pandas/conver-pandas-column-to-list/#  
[Accessed 29 11 2023].

Odabaşı, E., 2019. *Medium.* [Online]   
Available at: https://medium.com/@ertuodaba/web-scraping-with-python-and-calculating-bayesian-averages-with-sql-for-better-product-ranking-218dcbb75e4b  
[Accessed 01 12 2023].