

CCT College Dublin

ARC (Academic Research Collection)

ICT

2023

Menu Recommendation system using Machine Learning

Kelly Crystine Ferreira Jesus
CCT College Dublin

Leo Jaime Kayser Macieski
CCT College Dublin

Follow this and additional works at: <https://arc.cct.ie/ict>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Ferreira Jesus, Kelly Crystine and Kayser Macieski, Leo Jaime, "Menu Recommendation system using Machine Learning" (2023). *ICT*. 38.

<https://arc.cct.ie/ict/38>

This Capstone Project is brought to you for free and open access by ARC (Academic Research Collection). It has been accepted for inclusion in ICT by an authorized administrator of ARC (Academic Research Collection). For more information, please contact debora@cct.ie.

Menu Assistance

Kelly Crystine Ferreira Jesus - 2019375
Leo Jaime Kayser Macieski - 2019221

A Report Submitted in Partial Fulfilment
of the requirements for the
Degree of
BSc in Computing in IT (4th year)



May 2023

Supervisor: Dr. Muhammad Iqbal

Abstract

Developing a recommendation menu system for restaurants based on the restaurant data and/or city food purchase data to help and change the way restaurants build their menu.

Using Data Analysis and Machine Learning to build a project that aims to solve the problem of restaurants and chefs when it comes to preparing menus, the latter with ingredients and dishes that encourage their customers to order more, come back and recommend the restaurant. Helping chefs to create dishes for their restaurants with more accuracy and higher probability to be ordered by their customers.

The project will cover tools to build the predictions, the project plan, collect datasets, manipulate data and evaluate the aspects of the situation. The main business goal of our project is to predict what ingredients customers would like to eat and, from that, give restaurants ingredient suggestions to create their next menu.

By providing an efficient ingredients decision maker, it will simplify the way menus are elaborated and improve overall customers satisfaction.

Another motivating factor in choosing this project was its potential to help society tackle the huge problem of food waste and the inequalities this entails.

Acknowledgements

Our appreciation for the lecturers' support, Ken Healy, David McQuaid and Aldana Louzan. The project supervisor Dr. Muhammad Iqbal, for their patience and feedback, which their classes helped in building our project structure.

We also could not have undertaken the completion without our classmates, who generously provided exchange of knowledge during the classes.

Table of Contents

Abstract	3
Acknowledgements	4
Introduction	6
Briefly about the project Methodology	7
Objectives to achieve the project goals	8
Roles and Responsibilities	9
1. Business Understanding - 1st Phase	11
1.1. Project plan	11
1.2. Project resources	12
1.3. Requirements, assumptions and constraints	12
1.4. Costs and benefits	13
1.5. Data mining/Machine Learning goals	13
1.6. Summary	14
2. Data Understanding - 2nd Phase	15
2.1. Collecting and Describing data	15
2.2. Exploratory Data Analysis (EDA)	18
2.3. Verifying data Quality	20
2.4. Summary	21
3. Data Preparation - 3rd Phase	22
3.1. Selecting Data	22
3.2. Data Cleaning	22
3.3. Data Construction	24
3.4. Data Integration	27
3.5. Summary	31
4. Modelling - 4th Phase	32
4.1. Models/Machine Learning (ML)	32
4.2. Splitting, Training and Testing models	34
4.3. Model Evaluation	38
4.4. Summary	40
5. Evaluation - 5th Phase	41
5.1. Evaluate results and Review process	41
5.2. Summary	44
6. Deployment - 6th Phase	45
6.1. Strategy and Steps	45
6.2. Monitoring and Maintenance	46
6.3. Final Thoughts	48
6.4. Summary	49
Conclusion	50
Appendix	51
Citations and references	52

Introduction

Developing a recommendation menu system for restaurants based on the restaurant data and/or city food purchase data to help and change the way restaurants build their menu.

Our project aims to solve the problem faced by restaurants and chefs when it comes to preparing menus, the latter with ingredients and dishes that encourage their customers to order more, come back and recommend the restaurant.

This project consists of analysing a database of foods consumed in recent times, focused on the interests of consumers, which will make predictions of ingredients to be consumed in the future.

The goal is to build a dashboard with a simple and informative view.

Many restaurants have an extensive menu, with many options, which can lead to wasted stock for fresh products or reliance on freezing with its required storage space and costs.

To avoid waste and bring fresher, more varied food to customers, we thought of creating a different style of creating menus. The dishes will be created from a combination of the most consumed ingredients, which will give more sales power.

The project will use multiple databases, Python programming, algorithms for machine learning to perform the prediction.

Making use of resources to build a business that will benefit small and large restaurants.

The vision for our project would have recommendations for the next dishes, sides, and pricing. It also would recommend ingredients that are in season and easily available locally. Also the best match for each type of meat: spices and sides. It could guide the chef to create the dish, leaving the cooking to the chef. People always want to try something new, but what foods/ingredients are they likely to eat?

Menu Assistance will use the restaurant data to predict the next ideal menu.

However, for the first time menu, we suggest using local grocery data or country/region grocery data available online.

Analysing Datasets to manage predictions of some aspects that we consider important to help restaurants to develop more accurate menus for local business. This will reduce cost and increase revenue (as the menus will be more attractive to the customers of the area and/or type of restaurant).

Another issue that we will be able to address is Food Waste. According to the Environmental Protection Agency EPA (2023) 23% of food waste in Ireland comes from restaurants and the Food Service. By helping to reduce the waste we would be simultaneously informing future plans regarding food production in the region or even country using our project. The project idea consists of promoting a better quality and more profitable menu to restaurants. Helping local restaurants, food production in the country and to control food waste.

Briefly about the project Methodology

We will use the CRISP-DM methodology model to build our project. CRISP DM is the abbreviation for Cross Industry Standard Process for Data Mining, a cross-industry standard process for data mining. It is a methodology that is capable of transforming data from the dataset into knowledge and management information.

The Crisp-DM methodology will guide us in our task of trying to:

- Get as much information as possible about our databases.
- Gain data insight for a more detailed analysis of the data available for mining.
- Prepare data for manipulations.
- Carry out the modelling for future decisions on the use of the model.
- Complete the models built according to the data mining success criteria after definition.
- Finally, a planning and monitoring of the tasks with a review of the project. (IBM, 2021)

Using CRISP-DM we will be able to break down the data mining project into six phases, as following:

Business Understanding - 1st Phase

Data Understanding - 2nd Phase

Data Preparation - 3rd Phase

Modelling - 4th Phase

Evaluation - 5th Phase

Deployment - 6th Phase

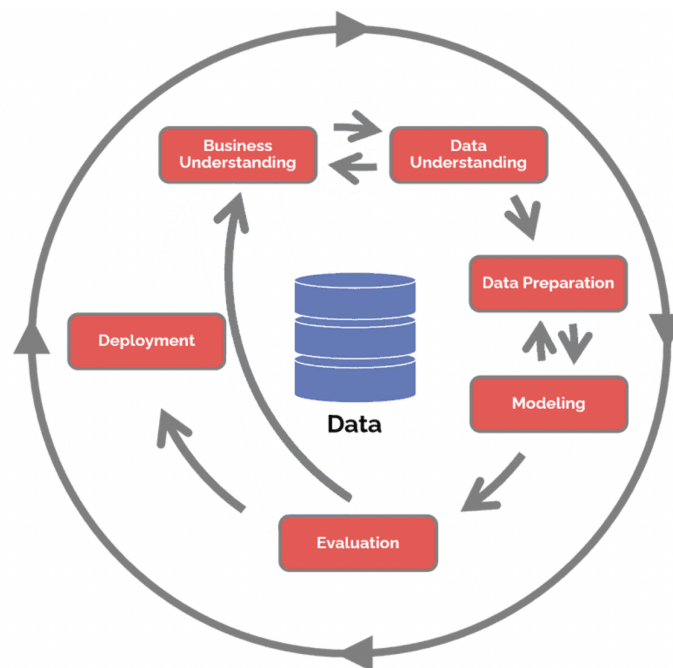


Figure 1. CRISP-DM life cycle

Objectives to achieve the project goals

- Q.1. How will Menu Assistance work as a business?
- Q.2. What are the data findings and their implications?
- Q.3. Are there any contributions for society?
- Q.4. Which model satisfies the project prediction?
- Q.5. How to incentivise local food retailers to contribute with their data?

Roles and Responsibilities

Kelly Jesus - I was responsible for writing the report, where I used the CRISP-DM methodology in following the six phases of the methodology. I gained knowledge of Business Understanding, with a deeper understanding after the SBIT/PSI integrated assignment (CA1) and the feedback received from the lecturers. Discuss dataset selection. I prepared the project plan and built a Strategic Plan excel spreadsheet to help us with group decision making while following the steps to build the project.

I manipulated the dataset in code using Python, doing data analysis, but mainly from the Data Preparation phase onwards. After this I had a better understanding for the writing of each phase.

During the preparation phase, I identified that to merge our three datasets into one would improve the implementation of the models for Machine Learning and I carried this out. I analysed the models used, researched how they worked and applied them to our project. I created some of the visualisations for analysis and model results.

I also created the Poster Presentation based on the model that the lecturer gave us in class and developed it with my partner.

I created a Google Word document for the project report based on the methodology, using the six phases and created the Poster presentation project content in Google Slides so that it could be managed, edited and reviewed by both members.

Leo Macieski - In this project I was responsible for the main idea, how the business would benefit from using the system, advantages and disadvantages. I also collected the datasets, where the initial idea would be to get a dataset directly from my old work but because they don't save this type of information, I took datasets available on the internet that would fit the project the most. After choosing the datasets, my partner and I worked together on the project's decisions on which data to keep, which to delete, which would be the best approach for carrying out the project. All business understanding, decision-making and models we worked together. I was also responsible for the Python code part. Visualisations, data cleaning, data preparation and models. Apart from the code part, I was also responsible for the presentation poster and slides and helped with the formatting of the report.

1. Business Understanding - 1st Phase

In this first phase of the CRISP-DM process will determine the Business Understanding, the objectives of our project and evaluation of the aspects of the situation to produce the project plan. From this exploring which tools, datasets and machine learning (ML) will be used.

Our business proposal is to help chefs to create dishes for their restaurants with more accuracy and with a higher probability of being ordered by their customers.

The main business goal of our project is to predict what ingredients customers would like to eat and from that, give restaurants ingredient suggestions to create their next menu.

Therefore, our target clients would be restaurants. The business goal is to upgrade their menu, improve taste satisfaction to attract new customers and to keep existing ones.

1.1. Project plan

We will process the selection of the datasets we chose, to be able to identify patterns and relationships in the data that could help us in our business project.

We intend to use multiple datasets with data on different types of ingredients that are being consumed, such as fish, meats, vegetables, spices, herbs and more, in order to predict the preference of customers in restaurants.

After choosing the datasets for the project, we need to observe, clean and prepare our data to exclude any unnecessary information and/or that we do not want to use. Once our data is prepared we are ready to do any analysis to capture relevant information.

To develop the project we will use the Python programming tool in Jupyter Notebook platform.

For Machine Learning we are keen on using one of the following models: KNN and Decision Tree Classifier. During the development of the project we will get more knowledge in regards to which form of Machine Learning model can be more beneficial for our project goal.

To achieve our data mining/machine learning goals we intend to gather the chosen datasets, do exploration to be able to identify the best option to clean and prepare them for visualisation and further analysis. After these steps, we will be able to make the ingredient predictions.

The steps above will be covered in more detail in later phases of the project such as Data Understanding, Data Preparation, Modelling and so on.

The datasets selected to build our project are from Kaggle.com.

Note: For new restaurants' first menu, a more accurate prediction of which ingredients are being consumed the most in the area can be made by gathering data from local supermarkets. For existing restaurants, the prediction can be made by using the restaurant's own data.

If the plan is to open a restaurant for a specific nationality, the best approach is to collect data from the country where the cuisine originates to find out which ingredients are popular in the country. Example: In an Italian restaurant here in Dublin, it is common to find the same dishes, such as carbonara, pizza margherita, calzone, etc; but when travelling to Italy, there are many other dishes which are also famous and popular there, but difficult to find here. If feasible, providing them in Ireland could open up possibilities to gain advantage on competitors.

1.2. Project resources

As mentioned previously in Project Plan, the tool that will be used to write the code is Python programming, using the Jupyter Notebook platform. Python is an open source and a general-purpose programming language, it uses less memory compared to other tools such as R programming.

Python open source licence technology is developed under an OSI-approved, free to use and also can be applied for commercial use.

We chose Python because it is versatile and can be used for a wide range of tasks, including web development, data manipulation and machine learning, which we will need for our project.

We consider it a great programming tool to build graphics and dashboards. Use can be made of its library options to get more freedom to explore the visualisations possibilities. For example, Seaborn and Matplot are libraries that are of interest to be used in our project.

Another factor that guided us to use Python is that Python works on different platforms such as Windows, Mac, Linux, Raspberry Pi, etc. (*W3Schools*)

Both of us are using Mac M1 and we faced some problems in previous semesters when using R programming.

1.3. Requirements, assumptions and constraints

Requirements

The Menu Assistance must:

R1 gather datasets of food/ingredients

R2 explore the data

R3 clean and prepare the data

R4 do visualisation and plotting to analyse the data

R5 analyse which is the best Machine Learning algorithm to make the prediction

R6 implement Machine Learning algorithm for the predictions

R7 produce a dashboard to display the ingredients predicted

Assumption

Use the restaurant's own data, if not available, to get data online including information from the restaurant location country. A suggestion is to get local grocery data for more relevant predictions. In this case, a letter of consent will be required (incentives could be offered) from the businesses that are providing their information. An advantage of this project is that we are manipulating data which does not compromise users' information.

Research on the legal and ethical aspects shows that, apparently, there is no law in Ireland that prohibits the sharing of restaurant data, bearing in mind that the only information that is essential for the development of this project is the names of the dishes and the ingredients in them. We don't need names, age, gender, card numbers or any other personal information from users. (GDPR)

Constrains

1. The analysis must be made using food datasets
2. Use of the restaurant's own data, if not available, its preferable data comes from local sources/food datasets.

1.4. Costs and benefits

The cost to produce the analysis for this project is minimal since the datasets used are available from free dataset websites.

However, if restaurants want a more specific and accurate prediction based on their own business, an investment has to be made over time to build their own database collection. We recommend this is done.

For a start, a simple ingredients column added on their sales report will be helpful for extraction for analysis.

Another possibility is to contact their local grocery shop for example, who will also see a benefit from it.

There are potential benefits for the business. It aims for a more profitable menu to improve customer sanctification, to increase revenue, to reduce costs and on top of all that, to reduce waste.

1.5. Data mining/Machine Learning goals

The data mining/Machine Learning goal is to analyse and predict which ingredients customers are likely to order/buy, based on food purchases over the past few years.

To produce a dashboard that will display the best selling ingredients over time. With that, to help restaurants to make decisions, such as in which dishes to insert or remove from the menu.

- Build predictive model with 80% accuracy
- Build predictive model with 80% recall
- Build predictive model with 80% precision

1.6. Summary

In this 1st Phase of the CRISP-DM Phase, Business Understanding we determined the business objectives of our project. Evaluating the aspects of the situation to produce the project plan, and from this which tools, datasets and machine learning will be used. Our Project plan and the process of the selection of the datasets.

Approaching the dataset selection, data mining and machine learning goals we intend to gather the chosen datasets and project resources such as the tools to develop the project. Requirements, assumptions, constraints, costs and benefits of the project are also addressed.

2. Data Understanding - 2nd Phase

In the second phase of the CRISP-DM process we are going to access the data using the resources listed in the first stage in Project resources. The Data Understanding phase is important for understanding the data and it will help us to make decisions during data preparation.

We will cover collecting and describing the data, exploring the data using EDA to perform the investigations on our dataset and verifying the data quality.

2.1. Collecting and Describing data

We have already decided which datasets we will use for the project. The collection was not what we expected for the project, as we could not find any dataset from restaurants with ingredients description. We are using multiple datasets to conclude the project proposal.

To load, manage and do some basic visualisation we will need the following libraries:

```
#libraries to be used in the project
import pandas as pd
import seaborn as sns
import numpy as np
import plotly.express as px
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib import pyplot

sns.set()
```

We load the data into the Python/Jupyter Network using Pandas (*pandas as pd*).

The method `.read_csv()` allows reading and using the external dataset into the DataFrame. Our datasets are all in `.csv` files, they are the simplest type of raw data, their rows are placed on new lines and commas separate the columns in each line.

Loading the datasets, sample command:

```
#reading the dataset
dataFruitVeg = pd.read_csv("datasets/daily_consumption_fruit_vegetables_eu.csv")
```

Figure 2. Loading dataset

Using `.shape` to check how many rows and columns the datasets have.

`.shape` sample command:

```
#checking the total of rows and columns in the dataset
dataFish.shape
```

```
(11028, 4)
```

Figure 3. Using `.shape`

Use of .info() method to check the datasets datatypes

With the method info() we can see the number of the columns/variables, total number of rows (the size of the dataset), data types (if the variable is quantitative or qualitative), the memory usage also.

.info() method, sample command:

```
#checking the data types
dataFruitVeg.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43488 entries, 0 to 43487
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   unit        43488 non-null  object
1   n_portion   43488 non-null  object
2   sex         43488 non-null  object
3   age         43488 non-null  object
4   country     43488 non-null  object
```

Figure 4. Using .info() method

Using the method .describe()

With that we are able to check some information about the quantitative variables such as the minimum and the maximum values, the mean and standard deviation.

.describe() method sample command:

```
dataFruitVeg.describe()
```

	time	value
count	43488.000000	35610.000000
mean	2016.500000	33.190211
std	2.500029	20.122211
min	2014.000000	0.000000
25%	2014.000000	14.000000
50%	2016.500000	33.600000
75%	2019.000000	48.900000
max	2019.000000	94.100000

Figure 5. Using .describe() method

We can see in the above that the numerical variables are presented in the summary statistics (mean, median, min, max, etc.) when using the .describe() method.

Using .value_counts() to check the amount of entries and counts of unique values.

With that we will be able to get more details about the Country column in the dataset.

.value_counts() method sample command:

```
dataFruitVeg['time'].value_counts()

2019    21744
2014    21744
Name: time, dtype: int64
```

Figure 6. Using .value_counts() method

As we can see above there are only two years in the Fruit/Veg dataset 2014 and 2019, for now we can see that the values are the same for both years.

Below, we are displaying our three datasets loaded. Added with their attributes, number of rows and columns of the acquired datasets.

Fruit and vegetables dataset

```
dataFruitVeg = pd.read_csv("daily_consumption_fruit_vegetables_eu.csv")
dataFruitVeg.head()
```

	unit	n_portion	sex	age	country	time	value
0	PC	0	F	TOTAL	AT	2019	29.1
1	PC	0	F	TOTAL	BE	2019	17.1
2	PC	0	F	TOTAL	BG	2019	54.2
3	PC	0	F	TOTAL	CY	2019	32.7

Figure 7. Fruit & Vegetables dataframe

Dataset size: 43488 rows and 7 column

Format of the data: dtypes: float64(1), int64(1), object(5)

The attributes/column that we consider import for further use can be seen below:

country: the country/region → country

time: the year of the collected value → year

value: total quantity per year → value

Fish and Seafood dataset

```
dataFish = pd.read_csv("fish-and-seafood-consumption-per-capita.csv")
dataFish.head()
```

	Entity	Code	Year	Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)
0	Afghanistan	AFG	1961	0.03
1	Afghanistan	AFG	1962	0.03
2	Afghanistan	AFG	1963	0.03
3	Afghanistan	AFG	1964	0.03
4	Afghanistan	AFG	1965	0.03

Figure 8. Fish and Seafood dataframe

Dataset size: 11028 rows and 4 column

Format of the data: dtypes: float64(1), int64(1), object(2)

The attributes/column that we consider import for further use can be seen below:

Entity and **Code**: the country/region → country

Year: the year of the collected value → year

Fish, supply quantity...: total quantity per year → value

Meat consumption by different types of meat dataset

```
dataMeatCon = pd.read_csv("meat_consumption_worldwide.csv")
dataMeatCon.head()
```

	LOCATION	SUBJECT	MEASURE	TIME	Value
0	AUS	BEEF	KG_CAP	1991	27.721815
1	AUS	BEEF	KG_CAP	1992	26.199591
2	AUS	BEEF	KG_CAP	1993	26.169094
3	AUS	BEEF	KG_CAP	1994	25.456134
4	AUS	BEEF	KG_CAP	1995	25.340226

Figure 9. Meat dataframe

Dataset size: 13760 rows and 5 column

Format of the data: dtypes: float64(1), int64(1), object(3)

The attributes/column that we consider import for further use can be seen below:

LOCATION: the country/region → country

TIME : the year of the collected value → year

VALUE: total quantity per year → value

SUBJECT: the type of meat (beef, sheep, poultry and pig) → ingredient

We have certain difficulty in finding a single dataset from a restaurant or grocery shop with multiple types of ingredients. Thus, we decided to use multiple datasets to analyse the food consumption of the past years, *Fruit/Veg*, *Fish/Seafood* and *Meat* (different types of meat).

There are 3 common attributes on the datasets: country/region, year and quantity/value. From the data analytics point, we will be able to see if any changes of food “preference” occurred during the years.

2.2. Exploratory Data Analysis (EDA)

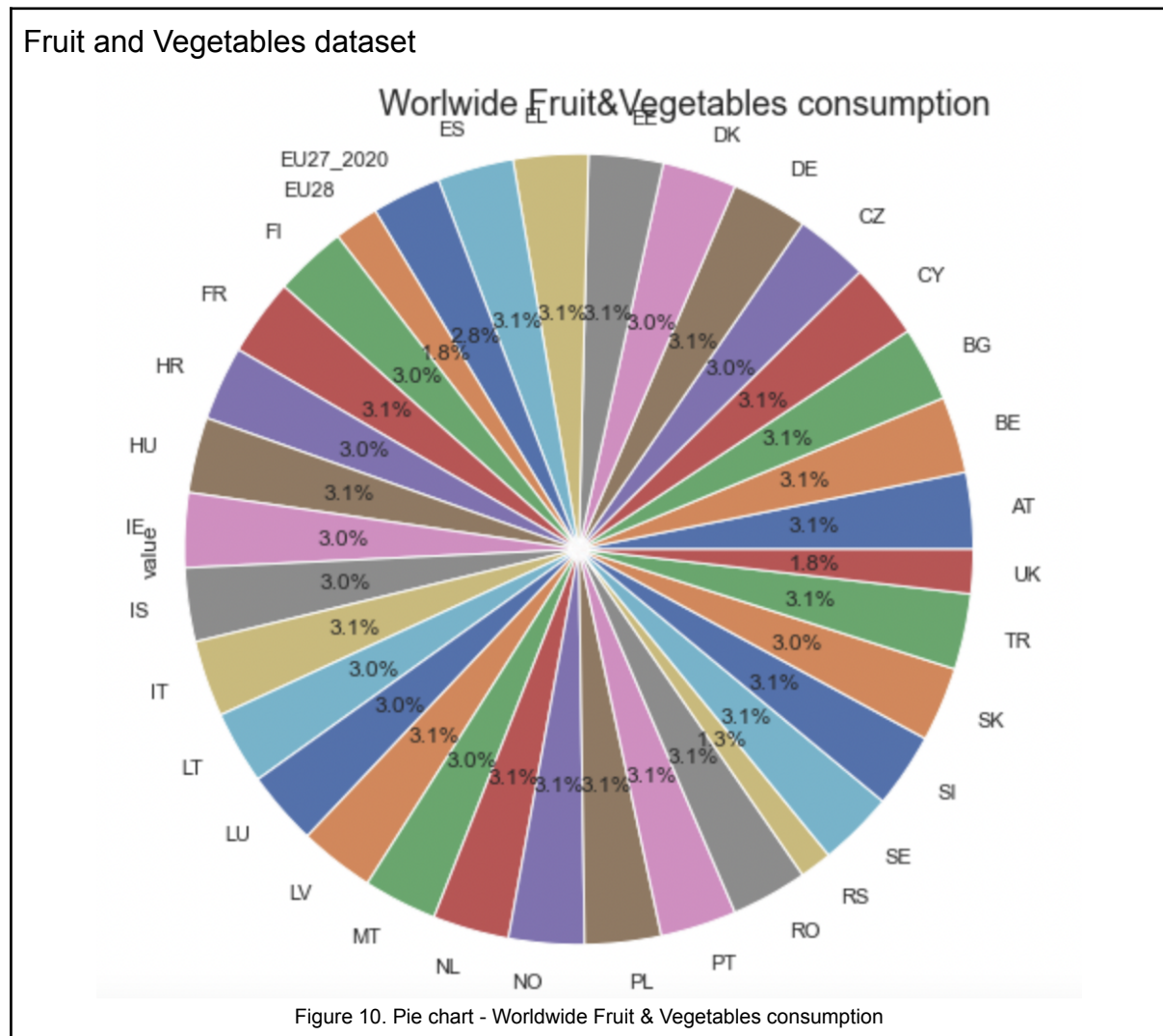
Before explaining exploratory data analysis (EDA) we must emphasise how it is an important aspect of any data analysis and for that reason its use in our project is essential. We received accuracy in assessing the quality of the data based on specific queries connected to the dataset. This gives us the possibility to do the most accurate analysis of our dataset and allows us to assess the quality of the data by asking questions about the dataset we have chosen, whether or not the data provided meets the required expectations.

During the EDA process we are able to address data mining questions using querying, visualisation and to use some reporting techniques. These include:

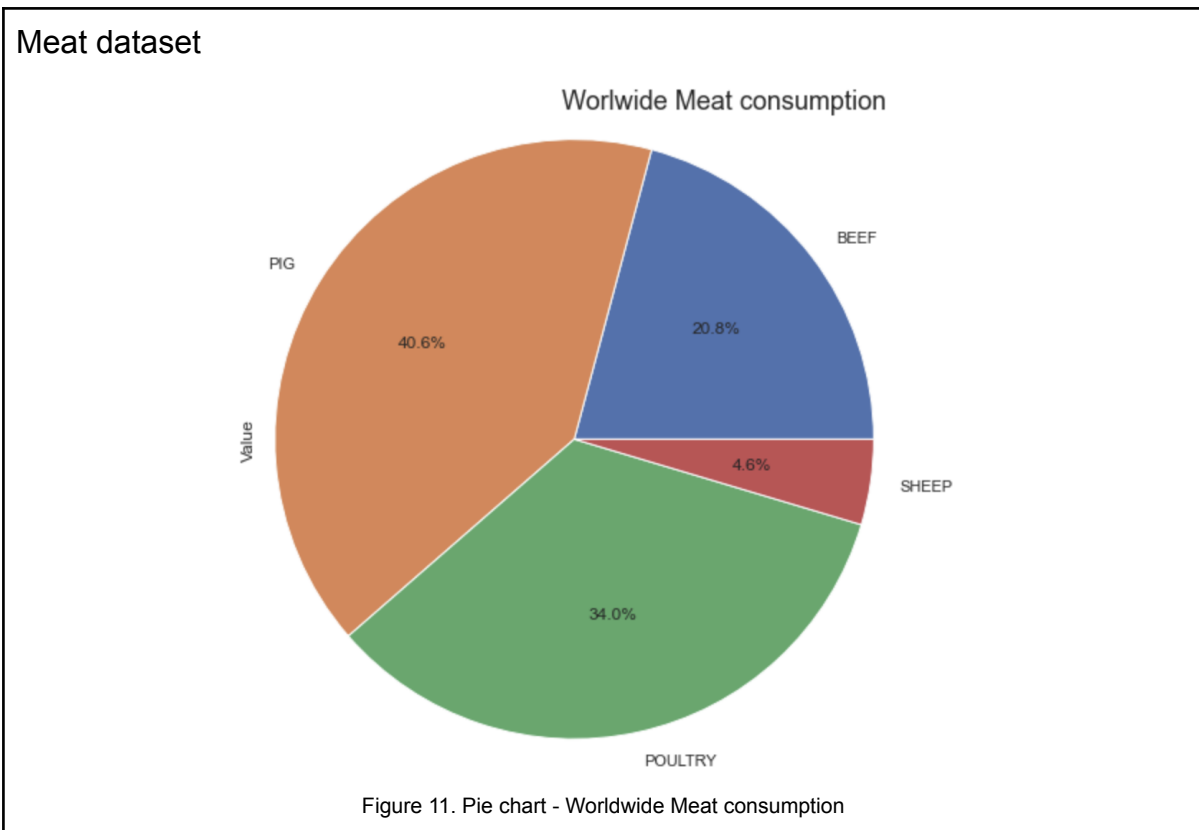
- distribution of the datasets attributes
- identifying the target attribute of the prediction
- the relationships between the numbers of attributes
- exploring if aggregations will be a good choice

- check if there are sub-populations attributes
- do some simple statistical analysis before cleaning and preparing the datasets.

These analyses will help us directly to address the data mining goals, get to know our datasets better and, from there, be more confident for any transformation and other necessary data preparation steps for the project's further analysis.



It is possible to observe above that the dataset Fruit and Vegetable has multiple countries, including Ireland as IE. Also, most of the countries have the same percentage of Fruit/Vegetable consumption. As we could see previously this dataset has only two years, will Ireland have the same values for both years? We will explore that further on the project.



The above visualisation describes the results of the Worldwide Meat consumption. We can observe that PIG/Pork has the highest value, followed by POULTRY. Will Ireland have the same results?

As we are studying the hypothesis of using only Ireland values to perform ML, it will be interesting to see it in comparison to worldwide values.

2.3. Verifying data Quality

We believe that the datasets will be good for our project goal. We can see that they have relevant attributes/columns. Thus, the quality of the data satisfies the project needs. From this point we could address some questions such as:

Do the datasets cover the requirements? Yes, they cover the requirements even though they are not the exact datasets we were looking for.

Do they have any errors? The Meat dataset does have an error in the TIME column regarding the years that goes up to 2026, as they were based on prediction.

Are there missing values in the datasets?

We will explore a bit more in the next phase during Cleaning.

We can see that some of the columns are categorical and we know for example that we need numerical values to perform the ML models.

We will cover in the next phase some solutions to improve the datasets quality. For example, data that needs to be cleaned and/or transformed during Data Preparation. After that, we will be able to do analysis of relevant information.

2.4. Summary

In this 2nd Phase of the CRISP-DM Phase, Data Understanding accessed the data, covered collection and description of the data and explored the data using EDA to perform the investigations on our dataset. Use was made of multiple methods to explore our datasets, loading and displaying them.

The more we know our data, the better we can perform the next phases. To aid this some graphics were plotted to visualise our datasets. Also, we verified data quality and asked some questions to be solved in phase three, Data Preparation.

3. Data Preparation - 3rd Phase

In the third phase of the CRISP-DM we will perform the Data Preparation process. Considered by IBM (2023), Eremenko (2020) and most of the researched resources generally, to be one of the most time-consuming and important phases of data mining. It is estimated that usually this phase takes 50-70% of a project's time and effort. Thus, more time and effort spent on this phase, we will be rewarded with smoother running during the next phases.

“When we prepare our data we are establishing a common language between human and machine.” (Eremenko, 2020)

3.1. Selecting Data

At this stage we have decided the datasets that will be used in our project. Also, the data that we are keen on using for analysis. The data selected does include relevance and quality to the data mining goals. What could be technical constraints are limits on data volume and the variety of attributes/columns that we were looking for in a more detailed dataset. However, we were able to manage this using multiple datasets to achieve the project goal.

3.2. Data Cleaning

As usually raw data comes ‘dirty’, the purpose of Data Cleaning is to eliminate or fix errors such as incorrect, corrupt, missing or invalid data.

The selected techniques to clean our datasets are the following:

```
dataFruitVeg.isnull().values.any()
```

```
dataFruitVeg.isnull().values.any()
```

```
True
```

```
duplicate_rows_dataFruitVeg = dataFruitVeg[dataFruitVeg.duplicated()]  
print("The number of duplicate rows are: ", duplicate_rows_dataFruitVeg.
```

```
The number of duplicate rows are: (5888, 7)
```

Figure 12. Null and duplicate values check

The `isnull()` method returns a DataFrame object where all the values are replaced with a Boolean value True for NULL values, and otherwise False. (W3Schools). As we can see above our `dataFruitVeg` dataset does have null values.

Below in figure 13, we are eliminating the N/A and duplicate values found in the dataset.

```

dataFruitVeg1 = dataFruitVeg

#using .dropna() to remove any NA
dataFruitVeg1 = dataFruitVeg1.replace('?', np.nan)
dataFruitVeg1 = dataFruitVeg1.dropna()
dataFruitVeg1 = dataFruitVeg1.apply(pd.to_numeric, errors='ignore')
#.drop_duplicates()
dataFruitVeg1 = dataFruitVeg1.drop_duplicates()

duplicate_rows_dataFruitVeg = dataFruitVeg1[dataFruitVeg1.duplicated()]
print("The number of duplicate rows are: ", duplicate_rows_dataFruitVeg)

The number of duplicate rows are: (0, 7)

dataFruitVeg1.isnull().values.any()

False

```

Figure 13. Cleaning N/A and duplicates from data

The Fruit&Veg dataset had 43488 rows and 7 columns before the cleaning and after the cleaning the values reduced to 35188 rows and 7 columns.

We applied the above techniques for the three datasets.

The Fish&Seafood dataset had 11028 rows and 4 columns before the cleaning and after the cleaning the values reduced to 9294 rows and 4 columns as the dataset had N/A values.

The Meat dataset had no N/A or duplicate data. But we had to fix the year's error in column 'TIME'.

See command below:

```

#getting only figures before 2022, as we spot an error in the meat dataset going up to 2026
dataMeatCon0 = dataMeatCon[dataMeatCon["TIME"].astype("int64")<2022]

```

The dataset had 13760 rows and 5 columns before the cleaning and after the cleaning. After keeping only values up to 2022, the values were reduced to 11880 rows and 5 columns.

3.3. Data Construction

This is the task that, for a time, we were wondering if it would be necessary or not. If we would prefer to continue with the three datasets separately or merging them.

After cleaning our data we will prepare our data in a way that will better benefit by making the process more practical and data easy to be manipulated.

We will not select/filter by years as we would like to predict and maintain traditional ingredients, opening the possibility to bring them to new dishes as trends often make a comeback.

The following is a sample of what the datasets should look like after this preparation:

country	ingredient	year	value
IE	VEGETABLE	2022	10
IE	BEEF	2019	3
EU28	FISH&SEAFOOD	2017	5

The process for the above display was the following:

We started by renaming the column names in the datasets, where the information was the same, thus having the same name instead of different labels. For example, country columns in all datasets have the same attribute name for 'country', using the method `.rename(columns={oldname: 'newname'})`.

After that, dropping the columns we considered not useful for the project goal with the method `.drop(columns=['columnname'])`.

The next step was using the method `.insert(1,'newcolumnname','newdataname')` to add the column 'ingredient' for ingredients in two of the datasets that did not contain this column/attribute, such as the *VEGETABLE* for Fruit&Vegetables and *FISH&SEAFOOD* for Fish&Seafood datasets.

Fruit&Vegetables dataset

Renamed: 'time' = 'year' and 'Value' = 'value'

Dropped: 'unit', 'n_portion', 'sex' and 'age'

Added: 'ingredient' (*VEGETABLE*)

Fish&Seafood dataset

Renamed: 'Code' = 'country', 'Year' = 'year' and 'Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)' = 'value'

Dropped: 'Entity'

Added: 'ingredient' (*FISH&SEAFOOD*)

Meat dataset

Renamed: 'TIME' = 'year', 'Value' = 'value', 'LOCATION' = 'country' and 'SUBJECT' = 'ingredient'

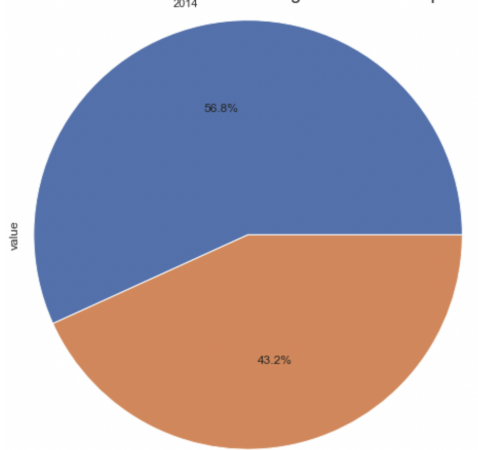
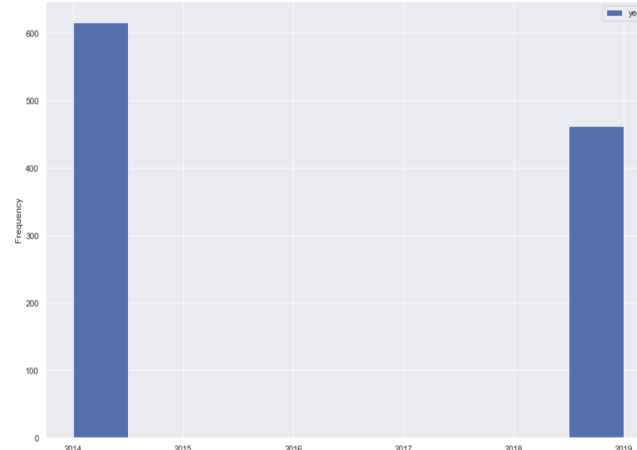
Dropped: 'MEASURE'

Sample of the steps below, using Fish&Seafood dataset:

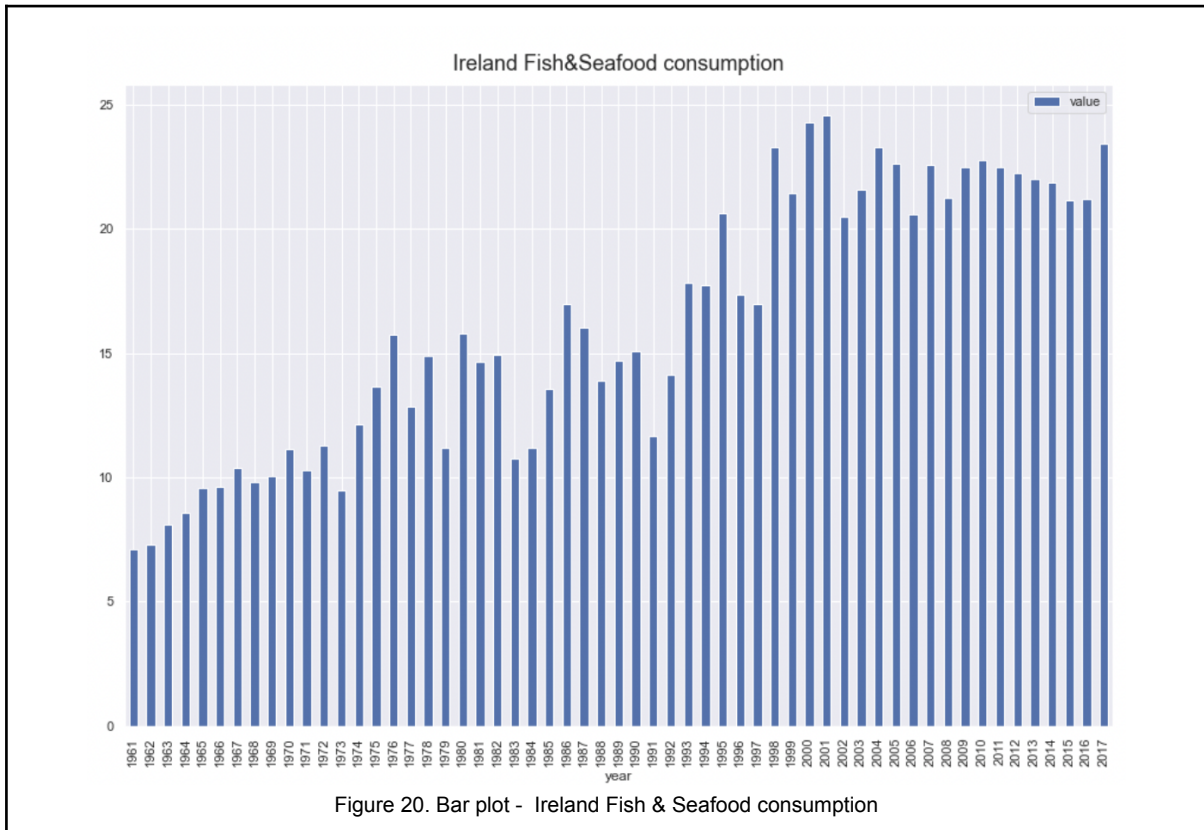
<p>Fish&Seafood dataset before preparation</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>Entity</th> <th>Code</th> <th>Year</th> <th>Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)</th> </tr> </thead> <tbody> <tr><td>0</td><td>Afghanistan</td><td>AFG</td><td>1961</td><td>0.03</td></tr> <tr><td>1</td><td>Afghanistan</td><td>AFG</td><td>1962</td><td>0.03</td></tr> <tr><td>2</td><td>Afghanistan</td><td>AFG</td><td>1963</td><td>0.03</td></tr> <tr><td>3</td><td>Afghanistan</td><td>AFG</td><td>1964</td><td>0.03</td></tr> <tr><td>4</td><td>Afghanistan</td><td>AFG</td><td>1965</td><td>0.03</td></tr> </tbody> </table> <p style="text-align: center;">Figure 14. Dataframe before preparation</p>		Entity	Code	Year	Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)	0	Afghanistan	AFG	1961	0.03	1	Afghanistan	AFG	1962	0.03	2	Afghanistan	AFG	1963	0.03	3	Afghanistan	AFG	1964	0.03	4	Afghanistan	AFG	1965	0.03	<p>Renaming the columns using the method <code>.rename(columns={'oldname': newname})</code></p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>Entity</th> <th>country</th> <th>year</th> <th>value</th> </tr> </thead> <tbody> <tr><td>0</td><td>Afghanistan</td><td>AFG</td><td>1961</td><td>0.03</td></tr> <tr><td>1</td><td>Afghanistan</td><td>AFG</td><td>1962</td><td>0.03</td></tr> <tr><td>2</td><td>Afghanistan</td><td>AFG</td><td>1963</td><td>0.03</td></tr> <tr><td>3</td><td>Afghanistan</td><td>AFG</td><td>1964</td><td>0.03</td></tr> <tr><td>4</td><td>Afghanistan</td><td>AFG</td><td>1965</td><td>0.03</td></tr> </tbody> </table> <p style="text-align: center;">Figure 15. Dataframe columns renamed</p>		Entity	country	year	value	0	Afghanistan	AFG	1961	0.03	1	Afghanistan	AFG	1962	0.03	2	Afghanistan	AFG	1963	0.03	3	Afghanistan	AFG	1964	0.03	4	Afghanistan	AFG	1965	0.03
	Entity	Code	Year	Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)																																																									
0	Afghanistan	AFG	1961	0.03																																																									
1	Afghanistan	AFG	1962	0.03																																																									
2	Afghanistan	AFG	1963	0.03																																																									
3	Afghanistan	AFG	1964	0.03																																																									
4	Afghanistan	AFG	1965	0.03																																																									
	Entity	country	year	value																																																									
0	Afghanistan	AFG	1961	0.03																																																									
1	Afghanistan	AFG	1962	0.03																																																									
2	Afghanistan	AFG	1963	0.03																																																									
3	Afghanistan	AFG	1964	0.03																																																									
4	Afghanistan	AFG	1965	0.03																																																									
<p>#removing a column using <code>.drop(columns=['columnname'])</code></p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>country</th> <th>year</th> <th>value</th> </tr> </thead> <tbody> <tr><td>0</td><td>AFG</td><td>1961</td><td>0.03</td></tr> <tr><td>1</td><td>AFG</td><td>1962</td><td>0.03</td></tr> <tr><td>2</td><td>AFG</td><td>1963</td><td>0.03</td></tr> <tr><td>3</td><td>AFG</td><td>1964</td><td>0.03</td></tr> <tr><td>4</td><td>AFG</td><td>1965</td><td>0.03</td></tr> </tbody> </table> <p style="text-align: center;">Figure 16. Dataframe after drop column 'Entity'</p>		country	year	value	0	AFG	1961	0.03	1	AFG	1962	0.03	2	AFG	1963	0.03	3	AFG	1964	0.03	4	AFG	1965	0.03	<p>#inserting a column with the method <code>.insert(1,'columnname','rowvalue')</code></p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>country</th> <th>ingredient</th> <th>year</th> <th>value</th> </tr> </thead> <tbody> <tr><td>11023</td><td>ZWE</td><td>FISH&SEAFOOD</td><td>2013</td><td>2.82</td></tr> <tr><td>11024</td><td>ZWE</td><td>FISH&SEAFOOD</td><td>2014</td><td>3.39</td></tr> <tr><td>11025</td><td>ZWE</td><td>FISH&SEAFOOD</td><td>2015</td><td>3.82</td></tr> <tr><td>11026</td><td>ZWE</td><td>FISH&SEAFOOD</td><td>2016</td><td>3.79</td></tr> <tr><td>11027</td><td>ZWE</td><td>FISH&SEAFOOD</td><td>2017</td><td>3.73</td></tr> </tbody> </table> <p style="text-align: center;">Figure 17. Dataframe after add column 'ingredient'</p>		country	ingredient	year	value	11023	ZWE	FISH&SEAFOOD	2013	2.82	11024	ZWE	FISH&SEAFOOD	2014	3.39	11025	ZWE	FISH&SEAFOOD	2015	3.82	11026	ZWE	FISH&SEAFOOD	2016	3.79	11027	ZWE	FISH&SEAFOOD	2017	3.73						
	country	year	value																																																										
0	AFG	1961	0.03																																																										
1	AFG	1962	0.03																																																										
2	AFG	1963	0.03																																																										
3	AFG	1964	0.03																																																										
4	AFG	1965	0.03																																																										
	country	ingredient	year	value																																																									
11023	ZWE	FISH&SEAFOOD	2013	2.82																																																									
11024	ZWE	FISH&SEAFOOD	2014	3.39																																																									
11025	ZWE	FISH&SEAFOOD	2015	3.82																																																									
11026	ZWE	FISH&SEAFOOD	2016	3.79																																																									
11027	ZWE	FISH&SEAFOOD	2017	3.73																																																									

For the below visualisation we filtered the country to Ireland (IE) just to see what is the consumption in the country using Pie Chart and Histogram. There are only two years in this dataset, 2014 and 2019, but this is one of our biggest dataset with 35188 rows.

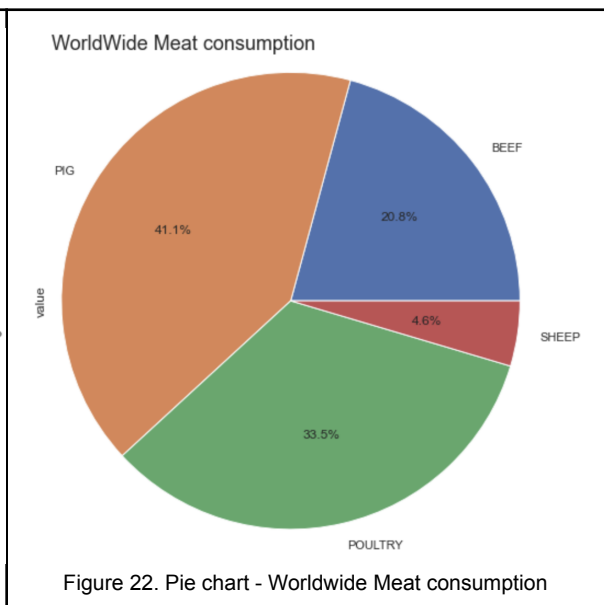
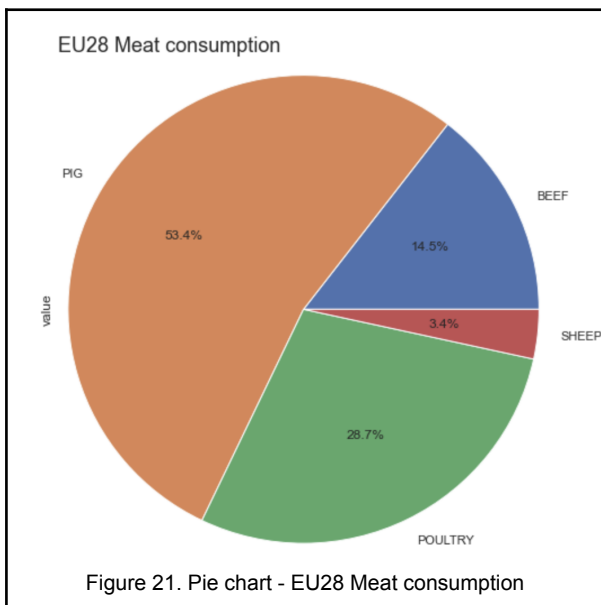
We can see a decrease of 13.6% less in the overall consumption of fruits and vegetables in the country in 2019 compared to 2014.

<p style="text-align: center;">Ireland Fruit&Vegetables consumption 2014</p>  <p style="text-align: center;">Figure 18. Pie chart - Ireland Fruits & Vegetables consumption</p>	<p style="text-align: center;">Ireland Fruit&Vegetables consumption</p>  <p style="text-align: center;">Figure 19. Bar plot - Ireland Fruits & Vegetables consumption</p>
---	--

Below in figure 20, we can observe that fish and seafood consumption increased in Ireland over the years. Having the highest pick in 2000 and 2001.



In the two figures below 21 and 22, we can see that EU28 consumes more PIG/pork meat compared to worldwide consumption and less BEEF than the worldwide percentages.



3.4. Data Integration

Our project would need a dataset with multiple ingredients. The ideal dataset would be from a restaurant's data history. As we did not find this specific dataset and decided to use food consumption data we ended with multiple datasets.

After cleaning and preparing the columns we want to integrate our datasets, combining them in a unique dataset with all the ingredients, creating new records and values.

Example: the Fruit&Vegetable dataset now has the same type of information and labels as the others (e.g., country, year, ingredient, value). Each of these columns contains records for each type of food. These datasets can be merged together into a new dataset, combining them from the source dataset using `.concat()` method to do operations in which new values are computed by summarising information from multiple records of multiple datasets.

Sample of dataset integration:

```
#categorising each dataset and merging them together
pieces = {"a":dataFruitVeg3,"b":dataFish4,"c":dataMeatCon3}
TOTALdataIngredients = pd.concat(pieces)
```

```
#categorising each dataset and merging them together
pieces = {"a":dataFruitVeg3,"b":dataFish4,"c":dataMeatCon3}
TOTALdataIngredients = pd.concat(pieces)
```

```
TOTALdataIngredients.head()
```

	country	ingredient	year	value
a	0	AT VEGETABLE	2019	29.1
	1	BE VEGETABLE	2019	17.1
	2	BG VEGETABLE	2019	54.2
	3	CY VEGETABLE	2019	32.7
	4	CZ VEGETABLE	2019	49.6

```
TOTALdataIngredients.shape
```

```
(58242, 4)
```

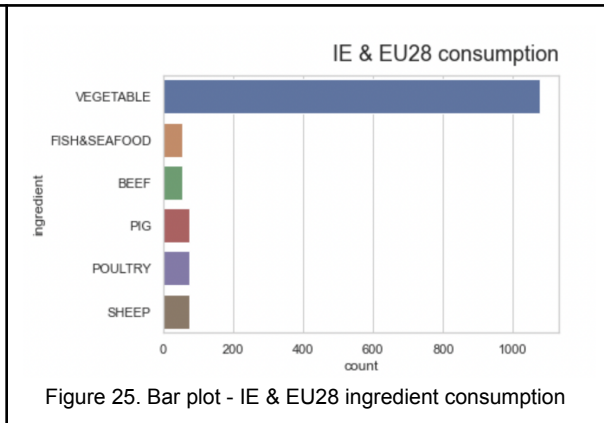
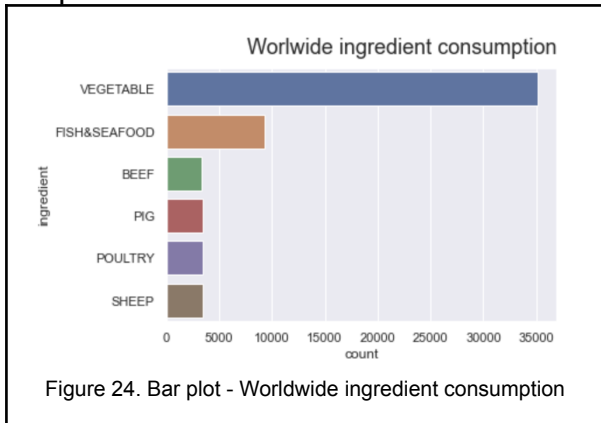
```
#checking the entries in each category
TOTALdataIngredients.groupby('ingredient').size()
```

```
ingredient
BEEF          3399
FISH&SEAFOOD  9294
PIG           3447
POULTRY       3457
SHEEP         3457
VEGETABLE     35188
dtype: int64
```

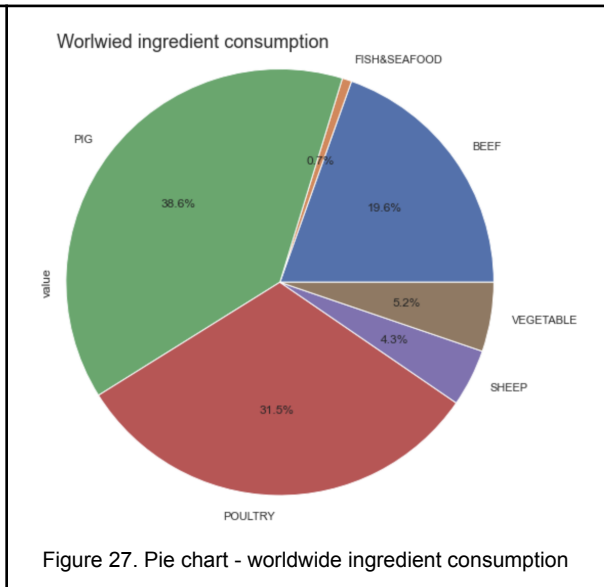
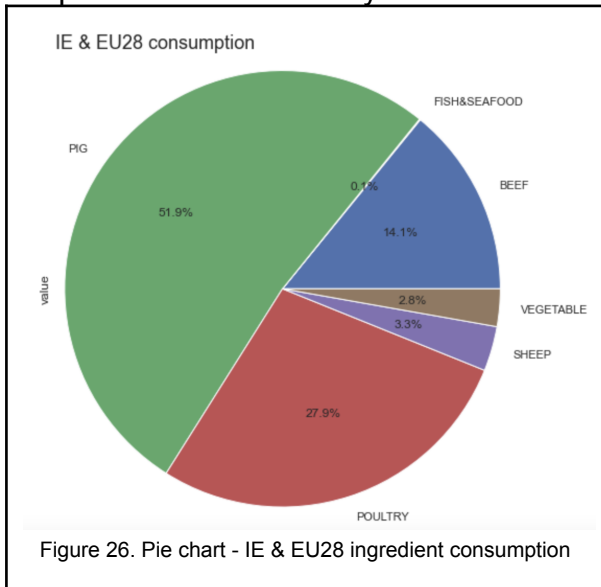
Figure 23. Dataframe worldwide values after integration (three dataset together)

Here we are plotting after the dataset integration. We can see that Ireland does not consume as much Fish&Seafood when we compare with the Worldwide results.

Graphic based on 'value' attribute:



Graphic based on 'country' attribute:



Next step will to be to prepare the data for Machine Learning (ML)

To perform the models in the next phase we need our column entries in numerical values and as we could observe, 'ingredient' and 'country' are categorical. The numerical preparation is done at this stage.

Mapping the column 'ingredient'

```
#mapping the ingredients with IDs
ingredient_id = {"VEGETABLE":1, "BEEF":2, "SHEEP":3, "POULTRY":4, "PIG":5, "FISH&SEAFOOD":6}
TOTALdataIngredients_num["ingredient"]=TOTALdataIngredients_num["ingredient"].map(ingredient_id)
TOTALdataIngredients_num.head()
```

Mapping the column 'country'

```
#creating an unique ID for each country
def generate_id(itr, fn):
    return dict(zip(itr, map(fn, itr)))

country_id = generate_id(country_list, lambda x: country_list.index(x) + 1)
```

Below in figure 28, the integrated dataframe after mapping 'ingredient' and 'country' columns to numerical values.

	country	ingredient	year	value	
a	0	1	1	2019	29.1
	1	2	1	2019	17.1
	2	3	1	2019	54.2
	3	4	1	2019	32.7
	4	5	1	2019	49.6

Figure 28. Dataframe after change to numeral values

Plotting a graphic after mapping 'ingredient' we can see the plot of the ingredients in numerical values. Graphic basen 'country' column.

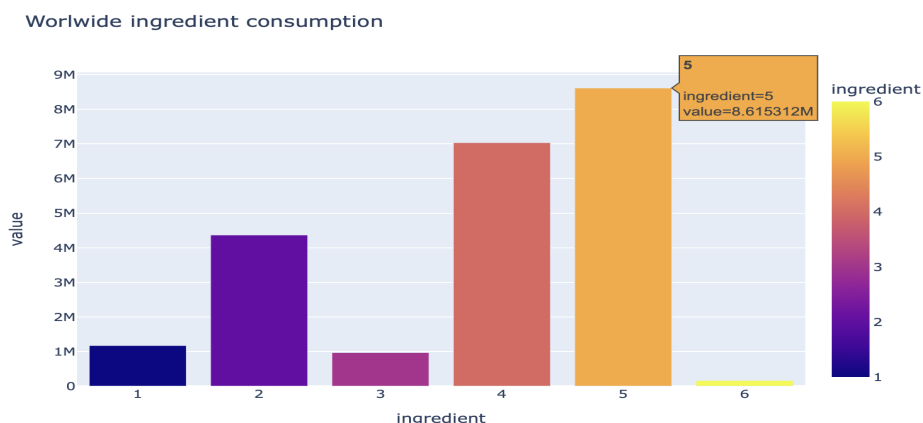
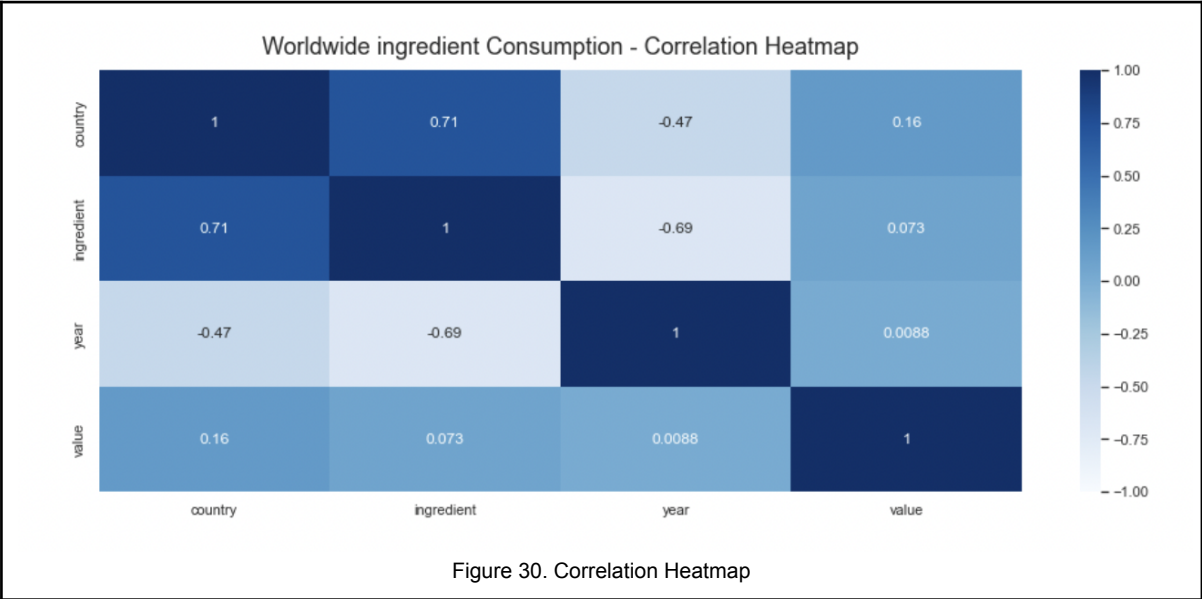


Figure 29. Interactive bar - worldwide ingredient consumption

1-Vegetable; 2-Beef; 3-Sheep(lamb); 4-Poultry; 5-Pig(pork); 6-Fish & SeaFood

Using the method corr()

To see the correlation between the variables and measure the strength correlation table between them.



Taking in consideration that when the correlation coefficient value is 1, it is a great correlation and a strong correlation if the value is between +/- 0.7 and 0.9. From that, we can see that the columns 'ingredient' and 'country' have a positive and strong correlation. Also, we can also see that most of the variables have a weak correlation between them.

3.5. Summary

In this 3rd Phase of the CRISP-DM Phase, Data Preparation we prepared our data to be ready for analysis and Machine Learning. The time we dedicated to this phase, we will allow for more efficient operating, thus benefiting the next phases in the project.

We started cleaning our datasets, eliminating and fixing errors. After that we constructed our data, deleting some columns, adding others and integrating our datasets to have only one dataset. The last step we did in this phase was to transform our categorical values into numerical values to be able to manipulate in the next phase, Modelling.

4. Modelling - 4th Phase

In the fourth phase of the CRISP-DM we will perform the Modelling. We selected a couple of models to test their quality and validity. From these results we will select a model to get the best two to do the ingredient prediction. We will briefly discuss their processes to understand the objectives of the two chosen model algorithms. Multiple interactions will be covered in this phase.

Note: We decided to use the Worldwide dataframe result instead of Ireland/EU dataframe, based on the fact that we want more possible values to run our models. As it has been said in previous phases of the project, a restaurant or city consumption/purchased dataset would be the better fit for real implementation.

4.1. Models/Machine Learning (ML)

Libraries to use for Machine Learning process:

```
from sklearn import metrics
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.linear_model import LogisticRegression
#from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
#from sklearn.svm import SVC

#To ignore/control warning messages
import warnings
warnings.filterwarnings('ignore')
```

Here we will perform our Machine Learning model for predicting the ingredients.

However, before we build the model we will generate a test to compare the model's quality and validity. To consider if a model is the right one for the prediction, we have to be aware of its accuracy, error can also be a factor to observe. So, here we are using accuracy and error rates as quality measures for data mining.

After analysing the data we get a good idea about our data, but we don't know yet which algorithms would be good for our project problem. We believe that KNN and

Decision Tree will be the best models for our project but we will do the above test to confirm this.

To help with this we will compare the results of five different models.

We will test 5 different algorithms:

1. Logistic Regression (LR)
2. Linear Discriminant Analysis (LDA)
3. K-Nearest Neighbors (KNN)
4. Classification and Regression Trees (CART)
5. Gaussian Naive Bayes (NB)

Linear: LR and LDA

Nonlinear: KNN, CART and NB

The results are:

LR: 0.799910 (0.002590)

LDA: 0.762580 (0.003759)

KNN: 0.885927 (0.003514)

CART: 0.944832 (0.003661)

NB: 0.801329 (0.001601)

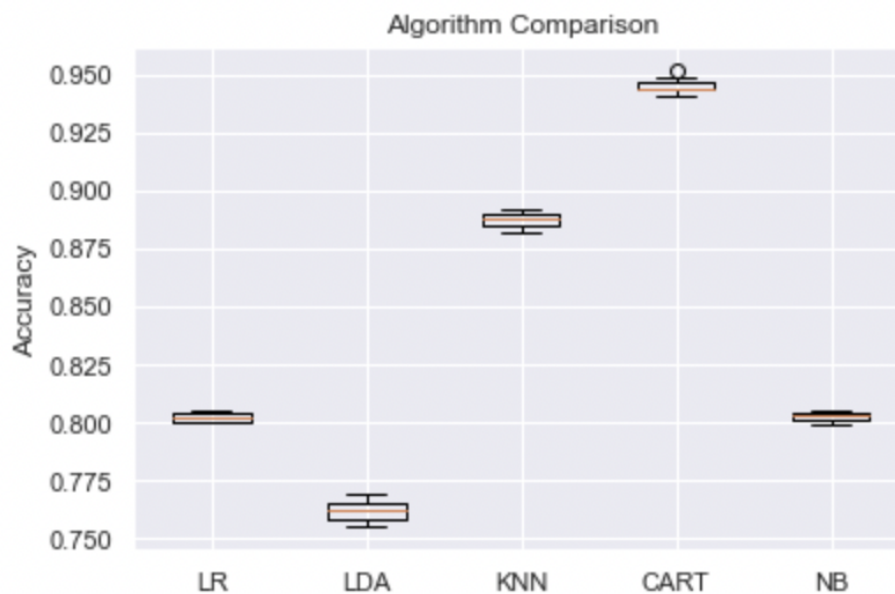


Figure 31. Box plot - Model Algorithm comparison

We are using two criterias accuracy and error to choose our model.

We can see above that Decision Tree/CART is probably the best model to be used for our project and KNN also might be a good option, both with high accuracy and less than 0.00% error.

We will explore a supervised data mining task using non linear Classification model algorithms. These models had better results and we decided to use K-Nearest Neighbors (KNN) and Decision Tree model algorithm. They will be used in our project to analyse our dataset.

K-Nearest Neighbors (KNN) model: K-Nearest Neighbors or KNN in its abbreviation. This is among the simplest of machine learning algorithms and can be used in different variations of institutions. KNN is a learning algorithm that can be slow depending on the size of the data. It is non-parametric, this means that no assumptions are made about the underlying data. In this way the selection is made based on the proximity of other data points, with the neighbours, regardless of which feature the numeric values are being represented. Being a “lazy” learning algorithm implies that it requires little or no training phase. Considered lazy/slow because it will calculate all the distances between the predicted target and training data point on the predicted phase (Marubon, 2017). Therefore, we can immediately classify new data points as they present themselves. But it can take a long time to run if dealing with big data.

Decision Tree model: Decision Tree model is a data classification and prediction method widely used due to its intuitive explainability characteristics. A Decision Tree starts with a single node and divides the dataset into several branches of outcomes which in turn lead to additional nodes. It evaluates the individual data records, which can be described by the attributes of the dataset. In this way, a Decision Tree provides facilities for dividing a dataset into segments. In addition to being easy to use in order to visualise the classification process, they can be explained by a series of nested if-then-else or rule-based statements. The criterion Gini for the impurity is the default function to measure the quality of a split. (Scikit-learn, 2023)

We are using Target: ‘ingredient’ and for Features: ‘country’, ‘year’, ‘value’.

4.2. Splitting, Training and Testing models

Running the modelling tool on the two models KNN and Decision Tree.

Splitting

Split the dataset, separating it into training and testing sets. Based on these two sets, we will use the training set to build the model, and separately in the test set to estimate its quality.

KNN model

For KNN we will leave the split Test and Train as default, the default will divide our data with 25% for testing and 75% training to evaluate the models.

Splitting

```
#splitting the train and test KNN
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
#X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 70)
```

Figure 32. KNN model Test and Train split

Training

```
knn = KNeighborsClassifier(n_neighbors=36, metric='euclidean')
knn.fit(X_train, y_train)
```

```
KNeighborsClassifier(metric='euclidean', n_neighbors=36)
```

Figure 33. KNN model Training

Euclidean distance

The KNN classification model calculates the distances between point to point in the training data set. Euclidean distance measurement technique is being used to classify the unknown instance represented by some feature vectors as a point in the feature space. (Guide and Badole, 2021)

By using Euclidean to measure the closest point/neighbour, our k-values "n_neighbors=36" will identify the 36 closest points.

Testing

```
#Predicting labels of unseen (test) data
#testing
knn.predict(X_test)

array([1, 3, 6, ..., 1, 1, 1])

y_pred = knn.predict(X_test)

print('KNN Accuracy is: {:.2f}'.format(metrics.accuracy_score(y_test, y_
KNN Accuracy is: 0.85
```

Figure 34. KNN model Testing and accuracy

Decision Tree model

We will divide our data with 30% for testing and 70% training to evaluate the models.

Splitting

```
#splitting the train and test DecisionTree  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 80)
```

Figure 35. Decision Tree model Test and Train split

Training

```
dtree = DecisionTreeClassifier(max_depth = 6, random_state = 0)  
dtree = dtree.fit(X_train, y_train)
```

Figure 36. Decision Tree model Training

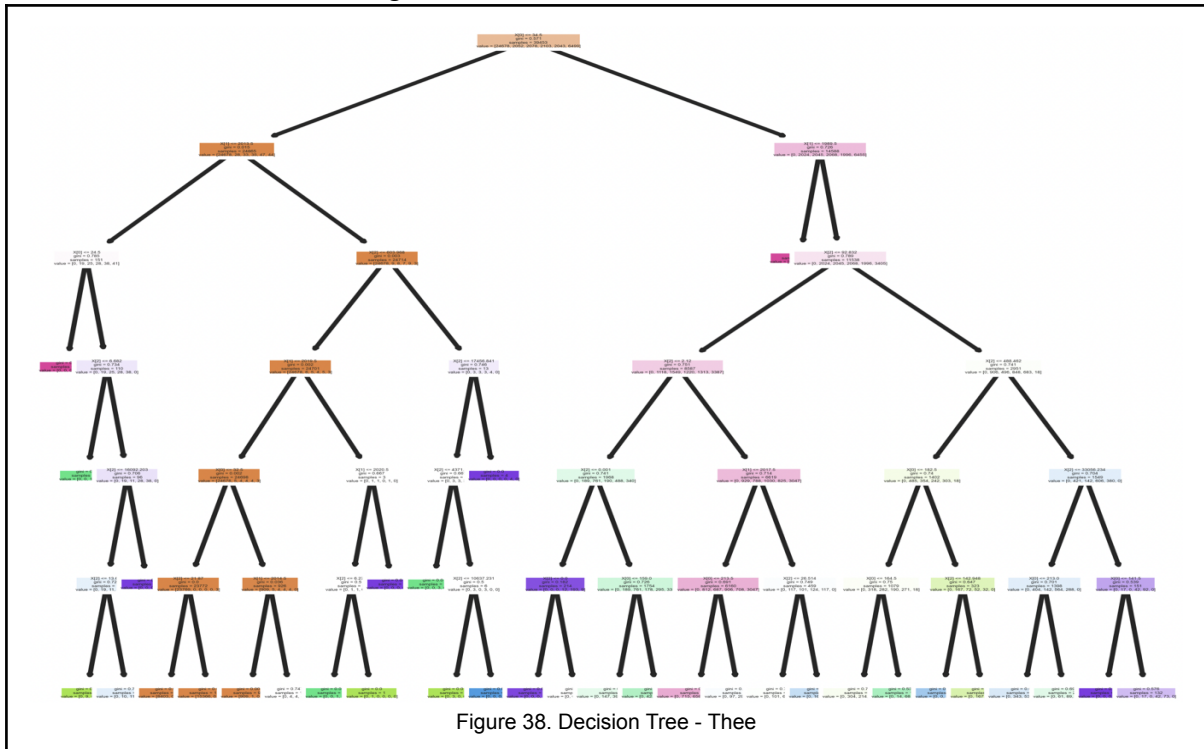
max_depth is set as 6, this determines the maximum depth of our tree.

Testing

```
#Predicting labels of unseen (test) data  
#testing  
dtree.predict(X_test)  
  
array([1, 1, 1, ..., 6, 1, 6])  
  
y_pred = dtree.predict(X_test)  
  
#print('Accuracy:',metrics.accuracy_score(y_test, y_pred))  
  
print('Decision Tree Accuracy is: {:.2f}'.format(metrics.accuracy_score(  
Decision Tree Accuracy is: 0.84
```

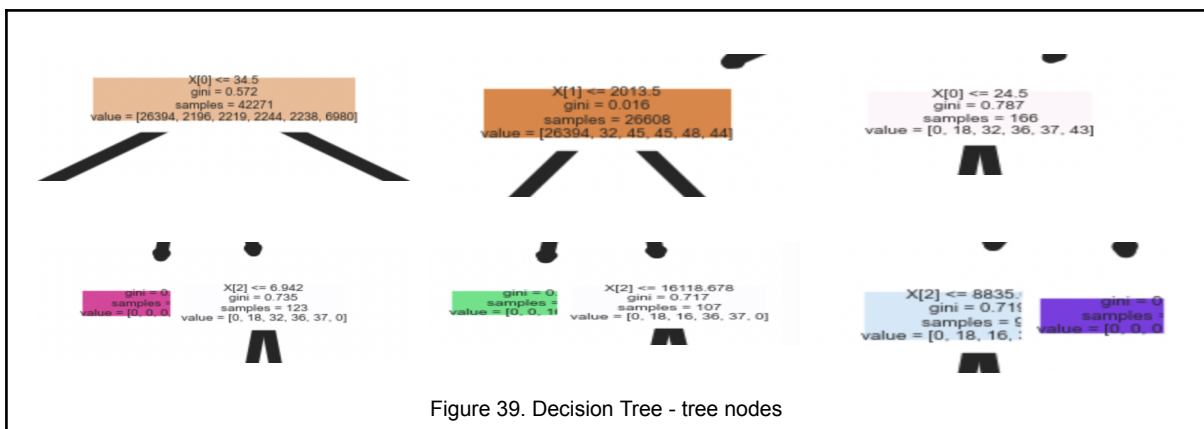
Figure 37. Decision Tree model Testing and accuracy

Our Decision Tree tree in figure 38.



Gini Index or Impurity

In Decision Tree, Gini is used as a measurement to build the algorithm that will determine how the features of our dataset should split nodes to form a tree. The probability of a random instance is measured and will be misclassified randomly. The Gini calculation is made by subtracting the sum of the squared probabilities of each class of the dataset from one, and the lower the Gini Index, the lower the likelihood of misclassification. It aims to decrease the impurities from the root nodes to the leaf nodes of a Decision Tree model. (Dash, 2022)



4.3. Model Evaluation

To evaluate the models data mining success criteria and our test design, in this step, we will evaluate the results of the models used throughout our project, KNN and Decision Tree so that we can define which algorithm model best fits our dataset and satisfies our project. Focusing in terms of accuracy and ranking the quality against one or the other.

Confusion Matrix → KNN model

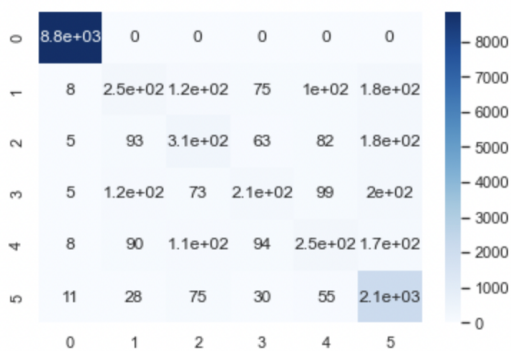
Model accuracy: 85%

Model error: 0.003514

```
#matrix for the result
print(confusion_matrix(y_test, y_pred))
```

```
[[8843  0  0  0  0  0]
 [  8 248 123  75 103 181]
 [  5  93 311  63  82 185]
 [  5 123  73 207  99 201]
 [  8  90 113  94 247 172]
 [ 11  28  75  30  55 2140]]
```

	precision	recall	f1-score	support
1	1.00	1.00	1.00	8843
2	0.43	0.34	0.38	738
3	0.45	0.42	0.43	739
4	0.44	0.29	0.35	708
5	0.42	0.34	0.38	724
6	0.74	0.91	0.82	2339
accuracy			0.85	14091
macro avg	0.58	0.55	0.56	14091
weighted avg	0.84	0.85	0.84	14091



VEGETABLE = 1 | BEEF = 2 | SHEEP (lamb) = 3 | POULTRY = 4 | PIG (pork) = 5 | FISH&SEAFOOD = 6

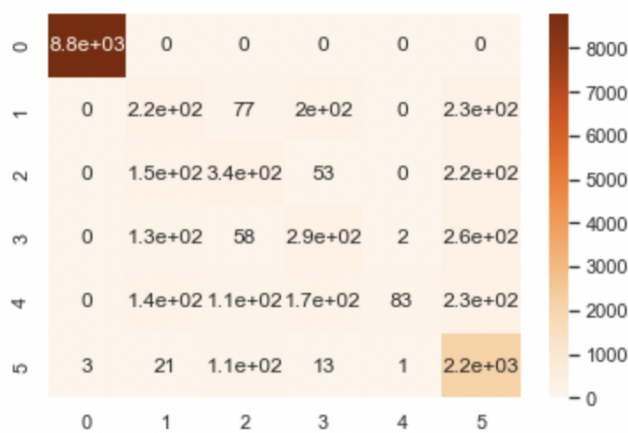
Figure 40. Confusion Matrix → KNN model (accuracy, recall and precision)

Confusion Matrix → Decision Tree model

Model accuracy: 84%

Model error: 0.003661

	precision	recall	f1-score	support
1	1.00	1.00	1.00	8794
2	0.33	0.30	0.31	733
3	0.49	0.44	0.46	768
4	0.40	0.39	0.39	743
5	0.97	0.11	0.20	739
6	0.69	0.94	0.80	2314
accuracy			0.84	14091
macro avg	0.65	0.53	0.53	14091
weighted avg	0.85	0.84	0.83	14091



VEGETABLE = 1 | BEEF = 2 | SHEEP (lamb) = 3 | POULTRY = 4 | PIG (pork) = 5 | FISH&SEAFOOD = 6

Figure 41. Confusion Matrix → Decision Tree model (accuracy, recall and precision)

As we can see from analysing the two models above, KNN model has the best results. Better accuracy with 85% and with low model error. Also because as our dataset is not too big it doesn't take long to run/compute. KNN model algorithm is a good choice for our project in making accurate predictions.

4.4. Summary

In this 4th Phase of the CRISP-DM Phase, Modelling we performed the model algorithms to make predictions in our project. We selected a couple of models to test their quality and validity and check their results. We chose the models we wanted to use and gave a brief explanation about them. Splitted our data in training and testing, confusion matrix and explored the outcomes. At the end doing the model evaluation, there was a lot to process in this phase.

5. Evaluation - 5th Phase

Evaluation is the fifth phase of CRISP-DM. When we reached this point, a large part of our project was completed. Both in the data mining phase and in the Modeling phase. We will evaluate the data mining results, the degree to which the model meets our business objectives and the project as whole.

5.1. Evaluate results and Review process

Regarding the results, in terms of the commercial success criteria of the Menu Assistance project, we can say that we are happy with the outcome of the models. With both, we can see that the Machine Learning goals were achieved, as the two chosen Models, KNN and Decision Tree have more than 80% accuracy and less than 0.00 model error.

KNN model accuracy					Decision Tree model accuracy				
	precision	recall	f1-score	support		precision	recall	f1-score	support
1	1.00	1.00	1.00	8848	1	1.00	1.00	1.00	10510
2	0.46	0.37	0.41	733	2	0.38	0.27	0.32	877
3	0.47	0.46	0.46	733	3	0.44	0.44	0.44	909
4	0.47	0.33	0.38	718	4	0.38	0.34	0.36	884
5	0.48	0.33	0.39	801	5	0.80	0.12	0.21	934
6	0.73	0.92	0.82	2258	6	0.67	0.94	0.79	2795
accuracy			0.85	14091	accuracy			0.84	16909
macro avg	0.60	0.57	0.58	14091	macro avg	0.61	0.52	0.52	16909
weighted avg	0.84	0.85	0.85	14091	weighted avg	0.84	0.84	0.82	16909

Figure 42. Confusion Matrix → KNN model (accuracy, recall and precision)

Figure 43. Confusion Matrix → Decision Tree model (accuracy, recall and precision)

Above we can also observe the accuracy, precision and recall results for each of the models.

Both models could be used for our project as the results are very good for both, high accuracy and low model error.

KNN is an intuitive model, easy to use and to understand. While Decision Tree, even though easy to use, is a complex model. However for big data it would be the best solution.

As a fit for our data set structure, KNN seems the preferred choice as explained earlier but the evaluation for both follows.

Below, we can see the models performance in a graphic, giving us the possibility to visualise which ingredients were consumed the most over the years, and also shows us the outliers.

KNN model Ingredients prediction

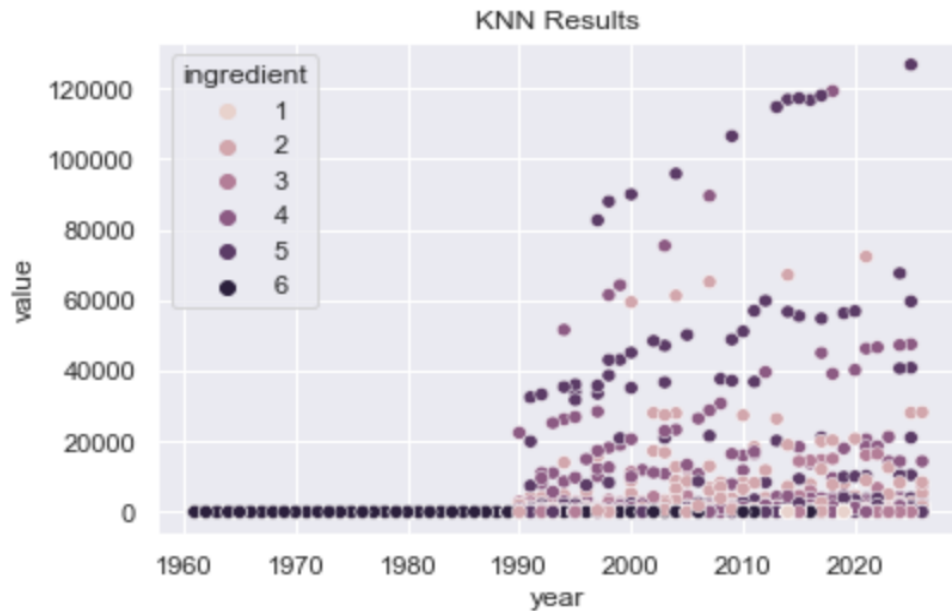


Figure 44. KNN model ingredient prediction

1-Vegetable; 2-Beef; 3-Sheep(lamb); 4-Poultry; 5-Pig(pork); 6-Fish & SeaFood

Decision Tree model Ingredients prediction

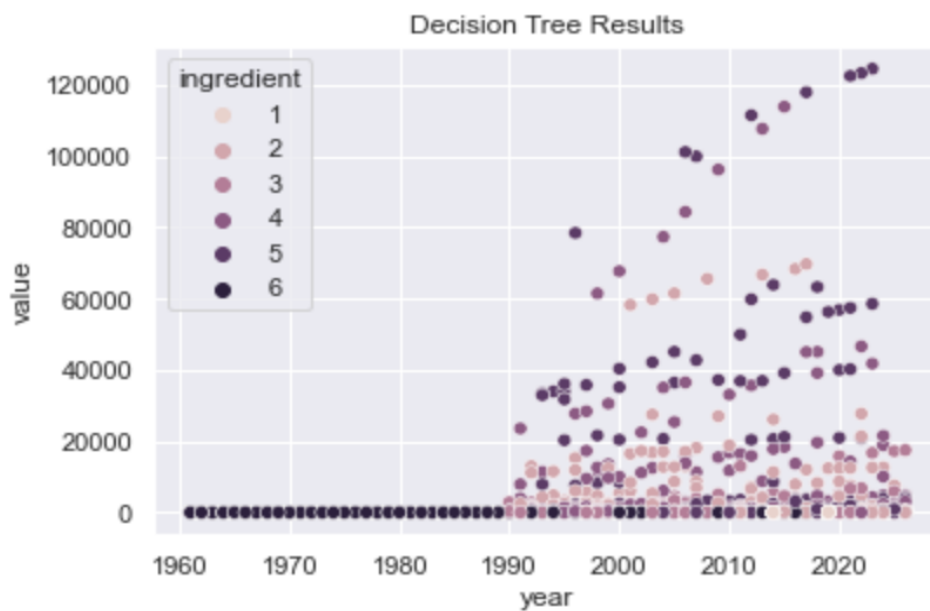


Figure 45. Decision Tree model ingredient prediction

1-Vegetable; 2-Beef; 3-Sheep(lamb); 4-Poultry; 5-Pig(pork); 6-Fish & SeaFood

Improving the Model

We can see that both models are suitable for our project. However, as we said previously KNN model might be the better fit. We will use this phase to improve the KNN model. We lowered the 'n_neighbors' to 6, so Euclidean will measure the closest neighbours, identifying the 6 closest points.

As we can see below:

Improving KNN model - our k-values for 6 closest points

```
ing the train and test KNN
, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
```

Using Euclidean to measure the closest point, "n_neighbors=6" wil identify our k-values by analyzing the 6 closest points.

```
#training
knn = KNeighborsClassifier(n_neighbors=6, metric='euclidean')
knn.fit(X_train, y_train)
```

```
KNeighborsClassifier(metric='euclidean', n_neighbors=6)
```

```
y_pred = knn.predict(X_test)
```

```
curacy is: {:.2f}'.format(metrics.accuracy_score(y_test, y_pred)))
```

```
KNN Accuracy is: 0.88
```

Figure 46. KNN model improved

The results obtained with the algorithm model we chose seem satisfactory, KNN model corresponds to the needs of this project dataset structure.

In the implementation of data mining, the challenge that we considered to have been the biggest since the beginning was to find and choose a dataset that met the design criteria. However, when we chose and explored the three datasets we learned how to work out the solutions.

In the Modeling phase, for example, we chose to use Worldwide values to obtain more values. Ireland only values were too limited and we know that the more values, the more accurate the forecast can be.

Note: Almost all code used to build our project in Python were extracted from our CCT classes.

5.2. Summary

In this 5th Phase of the CRISP-DM Phase Evaluation, we evaluated the results, comparing the models used and improved the chosen model. Also we reviewed the process and some decisions that we made during the process.

6. Deployment - 6th Phase

In the sixth phase of the CRISP-DM, Deployment we will summarise the strategy and steps to perform it taking in consideration the evaluation results. Here we will document the steps for future deployment.

6.1. Strategy and Steps

After exploring a business idea for the project, the first thing to do is write about the idea as a business prospect. This will make the idea clearer and what are the developments that can be made. Following a methodology is the best way to achieve a project goal.

We chose CRISP-DM methodology, this helped us as we could follow steps:

- Choose/explore a business idea for the project
- Understand the business idea
- Collect data that will satisfy the business project (we used three datasets)
- Choose the tools as resources to build the project (using Python in Jupyter Notebook)
- Analyse the data
- Prepare the data
- Testing model to process our data for prediction
- Choose the best model

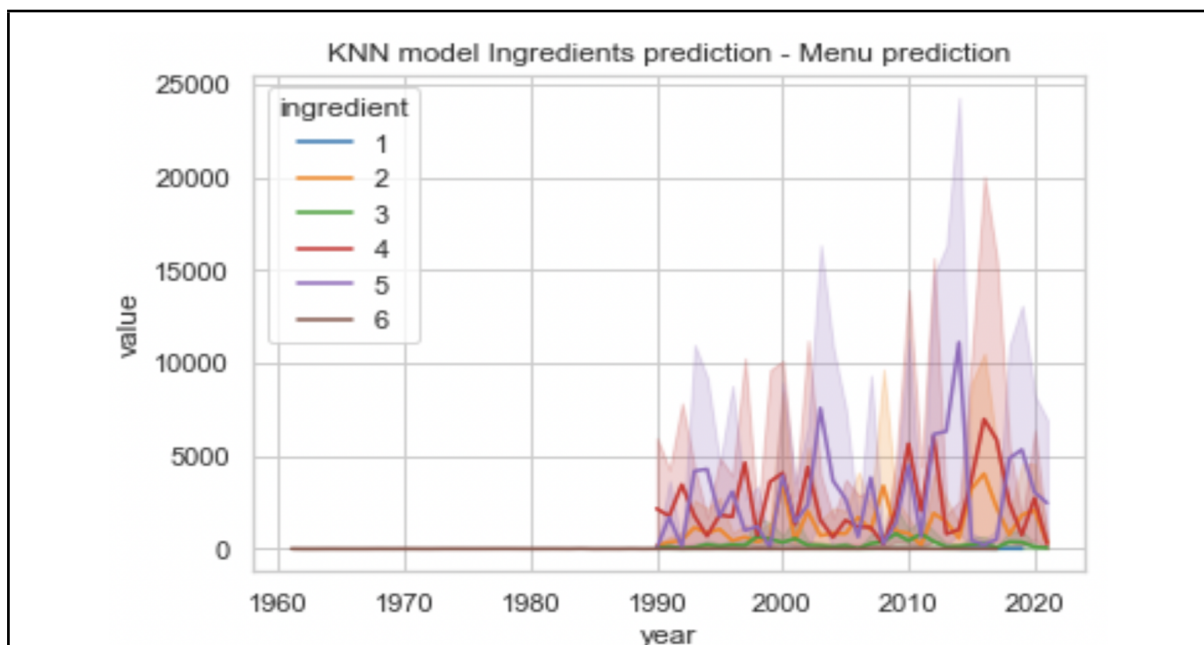
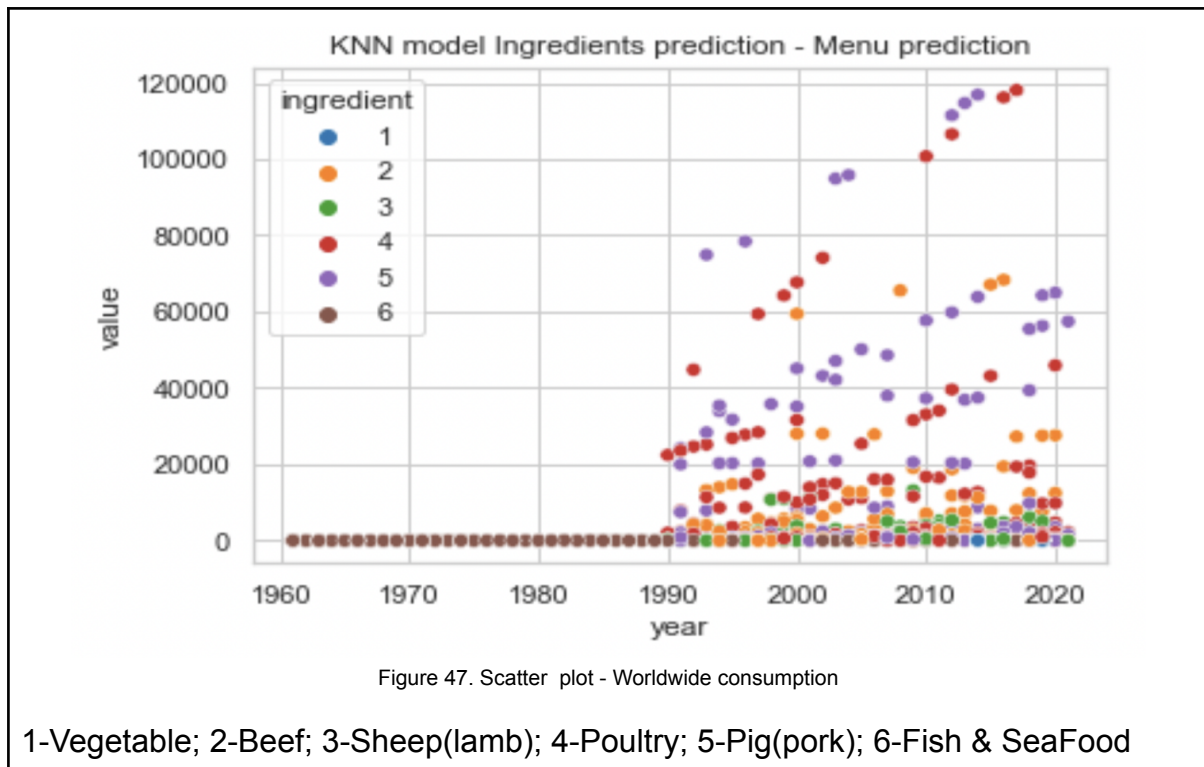


Figure 46. Line plot - Worldwide consumption

1-Vegetable; 2-Beef; 3-Sheep(lamb); 4-Poultry; 5-Pig(pork); 6-Fish & SeaFood



```
#Predicting labels of unseen (test) data
#testing
knn.predict(X_test)

array([1, 1, 6, ..., 1, 1, 6])
```

Figure 48. KNN model test prediction - Worldwide consumption

Fish & Seafood (6) are the ingredients that are more consumed during the years since 1960s and Fruit & Vegetables(1) are the ones with high volume in 2014 and 2019, that's why in our prediction is one of the ingredients selected.

Only in the 90s the consumption of other ingredients had a jump and the preferences and as we can see in figure 46 and 47, Pig/pork (4) and Poultry (5) might be a good choice.

According to Attinasi and Balatti (2021), the pace of globalisation has accelerated significantly since the early 1990s with profound implications for the structure of the global economy.

6.2. Monitoring and Maintenance

Monitoring and maintenance are important issues to address. We include the necessary steps (and how to take them) for the real implementation of the project idea and how to make it part of the day-to-day business and restaurant environment. As we were using general datasets, and not from a specific restaurant, the graphics ended up coming out in a more general way, so we ended up doing a more general analysis at the beginning. The ideal for this project would be to use a dataset from an

existing restaurant, but the options we had (places where we already work and the ones contacted) did not record the dishes with ingredients, a key factor for us.

Potential future developments could be where the price of imported products can be analysed to get better deals and change the menus according to the season in that part of the world.

The ideal dataset for carrying out this work would be if the restaurant stored, in table format, the ingredients and prices of all the dishes that are sold in the establishment (see table below). With the division of each ingredient, the algorithm would be able to make a more precise analysis with the real data of the establishment and the prediction would be more accurate.

As we don't have the possibility to test our project's model in real test applications, there will unfortunately be no real restaurant demo, but it would be interesting to test the project in a local restaurant and follow the process of adding a set of data to be filled and gather the data for them.

Follow an example below:

Main_ingredient	Side_1	Side_2	Side_3	Seasoning	Price
Ribeye Steak	Mash potato	Green beans	Dried tomatoes	Peppercorn sauce	32.99
Tofu	Lettuce	Carrot	Chickpea	Olive oil	16.00
Chicken breast	Rice	Lettuce	Onion	Curry sauce	19.50

	Main ingredient	Side 1	Side 2	Side 3	Seasoning
0	Ribeye Steak	Mash potato	Green beans	Dried tomatoes	Peppercorn sauce
1	Tofu	Lettuce	Carrot	Chickpea	Olive oil
2	Chicken breast	Rice	Lettuce	Onion	Curry sauce

Figure 49. Example of a restaurant dataset (created in our code section)

6.3. Final Thoughts

Evaluating our project, we are satisfied with the outcome and the knowledge obtained during its process, such as the data mining techniques and the models to carry out the predictions. They were challenging. As mentioned previously in other phases, selecting a suitable dataset was a challenge and time consuming step.

Other challenges included: code errors (for example trying to implement new features to our code) and Pandas did update and plot with Seaborn stopped running. This happened with both of us at different stages of the code process. Our version is 1.4.2 but the update went to 2.0.1 (see image below):



However, the main challenge faced during the project was in how to deal with decision making and learning how to work when the workflow changed direction. Also in the project process itself and in managing our time to develop it.

These factors made us take different directions to complete the project, and also gave us tools to deal with problems, mainly knowing how to deal with challenges and being able to develop solutions from these challenges.

6.4. Summary

In this 6th Phase of the CRISP-DM Phase, Deployment. Steps of the project process list are summarised, Doing plots to show the final outcome of the chosen model. Also, the challenges we faced during the project process and how we faced them.

Conclusion

When we started to prepare the project, we knew that we would have a big challenge, mainly due to time constraints and the upskilling needed. We knew it would require time and dedication from both group members.

The project was developed based on our business idea, which applies to the construction of an ingredient recommendation system for restaurant menus.

We did a database analysis of food consumed in recent times, to simulate the focus on consumer interests, and what is likely to be consumed in the future.

For Menu Assistance to predict the ingredients, it used multiple databases, Python programming in Jupyter Notebook as a tool, and CRISP/DM methodology as a guide to build our project report and code. This methodology distributes the work into six phases, which helps to maintain focus and a structured project.

We use EDA to explore and understand our data before doing anything else. We analysed the three Datasets, cleaned, prepared and integrated them to manage ingredient predictions to help restaurants to develop more accurate menus for local business, to reduce cost and increase revenue. To achieve this goal we used two model algorithms (Decision Tree and KNN model) for machine learning to perform the prediction. At the end we chose the one we considered better fit to our project.

We envisage contributing to society by helping local restaurants, controlling food waste and stimulating local, efficient food production. By helping to reduce food waste we would be simultaneously contributing to making future plans regarding food production in the region/country more efficient.

By making use of data resources and Machine Learning we aim to build a business plan that can benefit small and large restaurants by reducing food waste while bringing fresher ingredients and exciting dishes to customers.

Appendix

Group Communication and Activity

The project idea since day one was discussed and decisions made by both. We have met each other every Tuesdays and Fridays before college classes, most of the time using CCT premises to have our meetings.

Google Docs was used for the report documentation as we could work simultaneously if needed. GitHub was used to sync the Main Assistance project code, datasets and poster for pair collaboration during the development and implementation.

Strategic planning

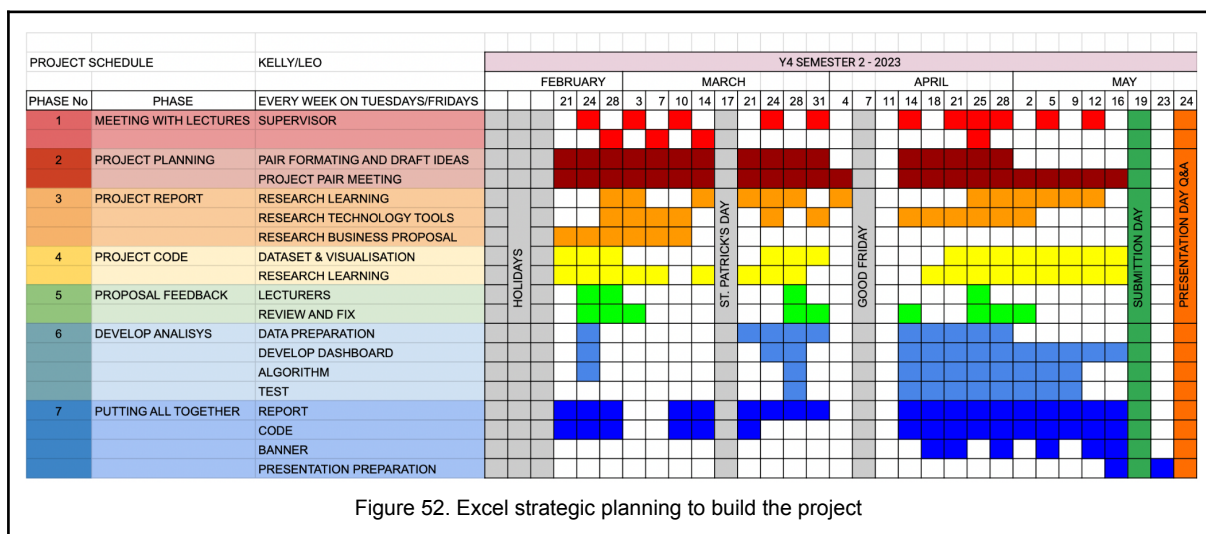


Figure 52. Excel strategic planning to build the project

Collaboration tools

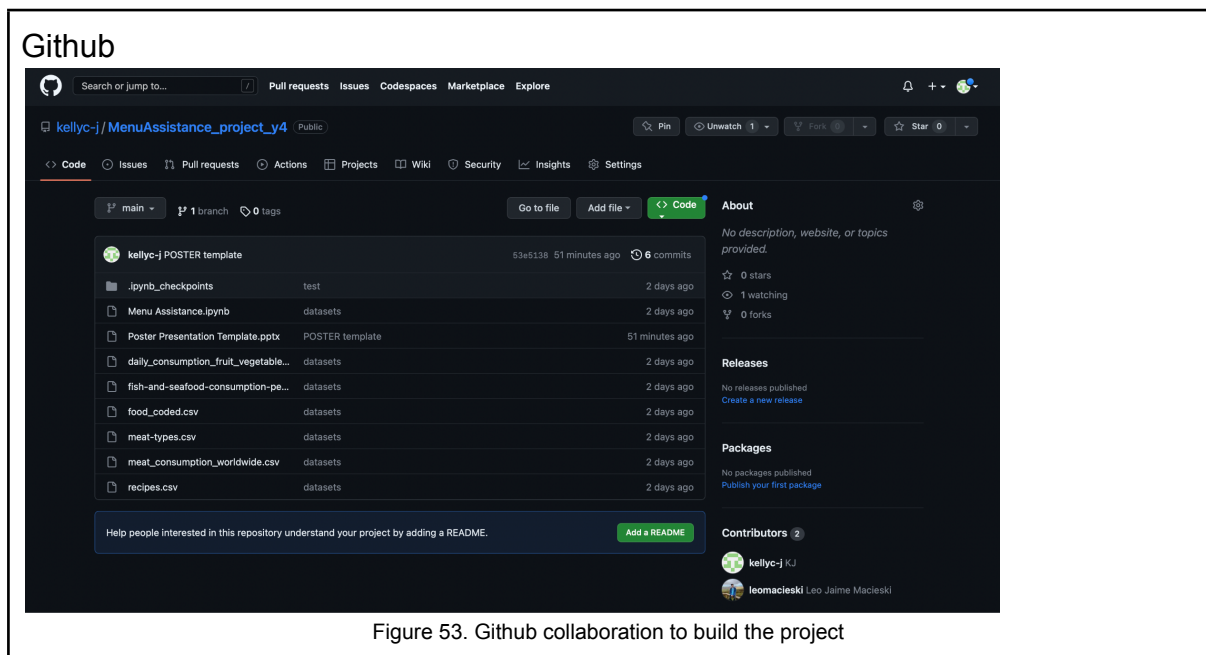


Figure 53. Github collaboration to build the project

Citations and references

PSII CCT (2023) Iqbal, Dr. Muhammad. "Problem Solving In Industry." CCT College Dublin, 2023 ed., Ireland.

IBM. (2023) "Introduction to CRISP-DM." IBM Documentation, 2021, <https://www.ibm.com/docs/en/spss-modeler/saas?topic=guide-introduction-crisp-dm>.

Adyatama, Arga (2020) "CRISPR-DM Case Study." RPubS, 11 October 2020, https://rpubs.com/Argaadya/crispr_dm.

Preda, G (2022) Kaggle. "Daily consumption of fruit & vegetables in Europe." Kaggle, <https://www.kaggle.com/datasets/gpreda/daily-consumption-of-fruit-vegetables-in-europe>

Geukjian, S (2022) Kaggle. "Fish and Overfishing." Kaggle, <https://www.kaggle.com/datasets/sergegeukjian/fish-and-overfishing>

Vagif, A (2020) Kaggle. "Worldwide Meat Consumption." Kaggle, <https://www.kaggle.com/datasets/vagifa/meatconsumption>

Marketman (2023) Control Your Restaurant Inventory https://try.marketman.com/cap/?utm_category=restaurant_mgmt&utm_source=capterra

Flipdish (2023) Sweeney, Claire. "Restaurant POS systems: Everything you need to know." Flipdish, 27 January 2023, <https://www.flipdish.com/ie/resources/blog/restaurant-pos-system>

EPA (2023) "Food Waste Statistics." Environmental Protection Agency, <https://www.epa.ie/our-services/monitoring--assessment/waste/national-waste-statistics/food/>.

Enterprise.gov.ie GDPR (2023) ("Data Protection and the General Data Protection Regulation (GDPR).") Department of Enterprise, Trade and Employment, <https://enterprise.gov.ie/en/data-protection/>.

W3Schools (2023) "Introduction to Python." W3Schools, https://www.w3schools.com/python/python_intro.asp.

Pandas (2023) Pandas 2.0, docs. "pandas documentation — pandas 2.0.1 documentation." Pandas, <https://pandas.pydata.org/docs/index.html>.

Eremenko, Kirill. (2020) Confident Data Skills: How to Work with Data and Futureproof Your Career. KoganPage, 2020.

Scikit-learn (2023) scikit-learn: machine learning in Python — scikit-learn 1.2.2 documentation, <https://scikit-learn.org/stable/>.

Scikit-learn (2023) “sklearn.tree.DecisionTreeClassifier — scikit-learn 1.2.2 documentation.” Scikit-learn, <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>.

Guide, Step, and Mayur Badole. “How KNN Uses Distance Measures?” Analytics Vidhya, 6 August 2021, <https://www.analyticsvidhya.com/blog/2021/08/how-knn-uses-distance-measures/>.

Galarnyk, Michael. (2020) “Visualizing Decision Trees with Python (Scikit-learn, Graphviz, Matplotlib).” Towards Data Science, 2 April 2020, <https://towardsdatascience.com/visualizing-decision-trees-with-python-scikit-learn-graphviz-matplotlib-1c50b4aa68dc>.

Müller, Andreas C. and Sarah Guido.(2016) Introduction to Machine Learning with Python. O'Reilly, 2016.

Dash, Shailey. (2022) “Decision Trees Explained — Entropy, Information Gain, Gini Index, CCP Pruning..” Towards Data Science, 2 November 2022, <https://towardsdatascience.com/decision-trees-explained-entropy-information-gini-index-ccp-pruning-4d78070db36c>.

Attinasi, Maria Grazia, and Mirco Balatti. (2021) “Globalisation and its implications for inflation in advanced economies.” European Central Bank, https://www.ecb.europa.eu/pub/economic-bulletin/articles/2021/html/ecb.ebart202104_01~ae13f7fe4c.en.html.

End of Final Project Report

May 2023

CCT College - Problem Solving In Industry

By

Kelly Crystine Ferreira Jesus - 2019375 | Leo Jaime Kayser Macieski - 2019221

*