1. Abstract
2. Problem Statement
3. Dataset
4. Data Exploration and Understanding
5. EDA and Data Processing
6. Merging and comparing data features
7. Choice of Machine Learning (Grid Search)
8. Analyze Variables
9. T-Test
10. Z-Score
11. ANOVA
12. Chi-Test
13. V
14. Training and Testing Data
15. Visualization and choice of made in approach
16. Machine Learning Modelling Outcomes/Results
17. Conclusion

**DATASETS**

1. **Source**

In this report I am going to analyses Ireland’s agricultural data against other countries, mainly in Europe­. Addition to this, the data used in report focused on Gross Domestic Product (GDP) and Daily Calory consumed versus population across countries in different continents. To begin with, we will first go through the process of analysing data which will include data exploration, describing the data which will be followed by plots as well as data visualization. The data we are going to use has been collected in comma separated values CSV format through the link <https://ourworldindata.org/> and our data is [Food Supply - Our World in Data](https://ourworldindata.org/food-supply) and (), this has been loaded into Python as Pandas data frame, however two sets of data has been collected from Our World in Food website and merged in order to get and accurate insight as well as picture of data.

1. **Features**

The most crucial and unavoidable step towards data analysis is loading the right and correct libraries which I did just before loading our data. Libraries are a collection of modules that contain functions and classes, and these can be used by various tasks as well as programs (Pydata.org, 2019). Pandas and NumPy are two of the most essential libraries to be used in Jupiter Notebook. As I was loading our data, we noticed that both of our data has same CSV format but different attributes as well as the number of rows. Our first data to be collected was daily per capita supply of calories vs GDP per capita and the second data was global food. The image attached below shows a true reflection of our data after being loaded and just before being explored.

Text

Description automatically generated

Fig 1: *Information for two datasets used.*

**DATA EPLORATION AND UNDERSTANDING**

**EDA and Data Processing**

This is another important thing I carried out before going deep in our analysis, data processing usually helps us in producing meaningful information that can be detected by an observer. So, under this section of our research in I will discuss and go through the steps I took in analyzing data, its characteristics and the whole process of converting the raw data into a form that is machine readable.

So upon importing both of my data (Daily per capita supply of calories vs gdp per capita.csv and Global food.csv) I renamed both of them for easily identification to (***cs*** for Calory Supply and ***gf*** for Global food), I then performed *“****cs.head( )”***and *“****gf.head( )”*** in order to check first few rows that my data has as seen on the figure below:

Table

Description automatically generated

Figure 2: *Checking first few rows on both* ***cs*** *and* ***gf*** *data.*

As seen from the Figure 2: above we can see that both imported data frames have **NaN** values that needs to be dropped as we do not need them and to make sure our data is clean. But before dropping all **NaN** rows by running a *“****cs = cs.dropna( )”*** for calory supply data frame I also checked to see all number of years the **Year** columns have in cs data frame. I again run *“****cs.head( )****”*to check first few rows of the **cs** data frame to have a clear view of what my new data looked like after dropping NaN values.

Then next step I took was to check all the descriptive statistics my two datasets has, and this includes count, mean, standard deviation, minimum, maximum etc and I did this by running “**cs.describe( )**”**.** Descriptive statistics is another essential step in exploring data as it computes and displays summary statistics for a python data frame, in this case I managed to compute this step on all the numerical values I have in all of my two loaded data frames. And I also checked the number of rows and columns I have in both of my data frames by running *“****data.shape( )”***  in both data frames. Next step I did in data preparation was to rename columns for easy Identification, and one way of renaming these columns in Pandas is by using the rename function ***data.rename(columns=****” ”****)*** to rename my cs data frame**,** as we know renaming method is quite beneficial especially when we need to change names of some selected columns where the information need to be specified. Upon renaming the columns I checked and verified all the columns I have in my ***gf*** dataset as well as checking the first few columns of my dataset once more to see what my dataset and how it looks like.

The next step I took in this chapter I had to drop all unnecessary or unused features/columns in both datasets as I decided not to use in my project as well as dropping all unwanted **NaN** values in my both datasets as they will make the data in both datasets inconsistent.

**Merging and Comparing Data Features**

The power of merging two or more datasets has always had advantages of bringing in information that we never had a chance to see in one place, this is another challenge data scientist has always had. Merging two of my datasets has helped me to bring together the features I wanted to analyze and compare. However, i have used the code below to join both of my datasets in order to be able to use the data and features I wanted. And the following figure shows the outcome after joining the data.

***data = pd.merge(gf, cs, on=["Country", "Year", "Population"], how="left" )***

Graphical user interface, table

Description automatically generated

Figure 3: *Merging two Datasets*

After joining the two collected data I also too advantage to explore the newly formed data by going again through the steps of analyzing and processing the data to have a new clear understanding of my newly formed data.

Table

Description automatically generated

Figure 4: *New dataset after merging two datasets*

**Comparing features and countries (Ireland vs other Countries)**

In this part of the document will focus more on comparing agriculture in Ireland versus other countries in Europe. First, I started by checking what’s in the Ireland row and analyze its features as seen from the image below.

Graphical user interface, application

Description automatically generated

Figure 5: *Ireland’s data*

Looking at Ireland’s data we can see what its features has following its columns, from the year we chose in first place and from the image we see that in 2015 Ireland had a population of **4 665 764**, **3717** Daily Calory and a GDP of **71508**, after printing all European countries by running the code “***data[data['Continent'] == 'Europe']”*** and by picking random six European countries (*Poland, Sweden, Cyprus, Belgium, Ireland and Luxembourg*) we can see that Luxembourg was the only country that has higher GDP than Ireland and we can also see that Luxembourg and Macao were the only countries with higher GDP than Ireland in the world, judging from the results of the results we can also conclude that the GDP for most countries were also been affected by population while Daily Calory has been the opposite case as we can see Ireland has higher Daily Calory than all of the European countries. And this has been checked by running the code “***data[data['Continent'] == 'Europe']***” that checks all features in our dataset.

Apart from GDP I also checked countries with higher daily calory in 2015 than Ireland and I noticed that only Belgium had a higher daily calory than Ireland. We can reflect this on the previous GDP result as well as Belgium has higher population than Ireland but with less GDP thank Ireland while Luxembourg which has higher GDP than Ireland has population less than Ireland. I also printed random data from Hungary to Japan from a list of countries of the world to see what the data looked like which gave me eight countries in turn and here we can also have another picture of how our data is been lined up.

Numerical data can somehow be tricky to read or analyze especially if it is in the object format, knowing that some of the data features I have in my new dataset needed to be converted to numerical for proper reading I changed their format by passing ***data['column’] = data['column'].astype(int)*** code, then again check its information to confirm the changes and as well to check again information in Ireland tuple and list countries with less GDP and Daily Calory than Ireland.

**Data Visualization**

In this section of our project, we will talk about data visualization which includes how to use a graphical or a linear decision-making process to encode our information visually. Visual displays usually help us to communicate a complex data relationship among variables and provide insights that is usually simplified and/or easily to grasp. The first presentation I made originated from top five countries with higher GDP Macao, Luxembourg, Ireland, Switzerland and United Arab Emirate, although these five countries were picked at random we can still notice that my focus is still on the European countries as we can still clearly see that Luxembourg remained the country with higher GDP than Ireland, after these two countries we can also see that Switzerland is the third highest in Europe although it comes after Ireland.

The second presentation is quite like the first one only that this listed all the features (GDP, Population and Daily Calory) including all countries, by hovering around our interactive chats we can see that different countries displays different information that our data has. And by hovering on the third point of our second graph we can see that in 2015 Ireland was the third highest when it comes to GDP worldwide as depicted by our dataset which we can also conclude that Ireland was one of the that had good consumption, government spending, investments as well as exports plus imports as opposed to the countries with lower GDP. Three figures below show how first, second and third visualization has been presented according to the data collected in the year 2015.

Chart, bar chart

Description automatically generated

Figure 6: *Top 5 countries with higher GDP*

Graphical user interface

Description automatically generated

Figure 7: *Daily Calory, GDP and Population of Countries*

A picture containing table

Description automatically generated

Figure 8: *Daily Calory against GDP across continents*

The next two visualizations in this section I focused on our area of study Europe, and these are Choropleth Maps with Pandas, similarly to the previous presentations I have also presented how Daily Calory on the first graph and GDP on the second graph across European continent. And data on these presentations can also be explored by hovering around the map area to get more information from them. The last graph is the world map/natural earth that has been constructed to view GDP across the globe

**Analyze Variables**