University of Sussex

Department of Informatics and Engineering

Machine Learning

Luis Morales Layja

May, 2024

Contents

1 Features and Labels

To obtain a complete dataset containing all the features and the label, the following steps were taken:

1. **Analyze Each File**: Each of the thirteen files was analyzed individually to extract the relevant features and the label.
2. **Create Individual Datasets**: For each file, a dataset was created that included all the features from that file. If the data was not already represented annually, it was converted to an annual format.
3. **Preserve Key Columns**: In each dataset, the "Year" and "Area" columns were preserved.
4. **Create Label Dataset**: Similarly, a dataset containing the label was created.
5. **Unified Merge**: Once all the individual datasets were prepared, they were merged into a single unified dataset using the "Year" and "Area" columns as keys.

Furthermore, the function `extraction_features` is used for extracting all the features and the label. It takes a dataset (DataFrame) and extracts specific characteristics (features) based on a given element. If the data is represented monthly, it can be grouped annually, summing the values for each feature. Then, it combines these subsets into a final dataset where each column represents a feature, using a specified join method (inner or outer). The function can also filter data to exclude the specific year 2023, as the information for this year is incomplete for all countries. The functionality of this function is represented by the pseudocode in Algorithm 1.

1.1 The label

The label is defined in the file `Dataset/Foodtradeindicators-FAOSTAT_data_en_2-22-2024.csv`. This file includes two key metrics: "Export Value" and "Import Value", where "Export Value" is the value aimed to be predicted (label). Additionally, there are twelve items listed under the "Item" column. For "Export Value", each of these items represents a different crop product, meaning there is an export value associated with each product. To determine the total export value for a country in a given year, we sum the export values of all these products. If there are any missing (NaN) values in the export data, we will use the inner method to remove those rows to prevent significant bias in the model that might occur if we tried to fill in (impute) those missing values.

The label "Export Value" was obtained using the `extraction_features` function. This function processes the dataset to extract and aggregate the export values for each crop product into separate columns. Figure 1 is a representation of how the dataset looks after applying the function.

Algorithm 1 Extraction Features Function

Require: dataframe, column of features, months=False, how='inner'

First, obtain the unique values of the specified column of features from the dataset and store them in a variable called `features`.

Then, initialize an empty list named `minidatasets` to hold the intermediate datasets (dataframes).

if the parameter `months` is True **then**

  for each unique feature in `features` **do**

    Filter the dataset to include only the rows where the element equals the current feature.

    Group the filtered dataset by "Year" and "Area", and sum the "Value" for each group.

    Exclude the rows where the "Year" is 2023.

    Rename the "Value" column to the current feature's name.

    Group this modified dataset by "Area" and "Year", and sum the values for the current feature.

    Append this resulting dataset to the `minidatasets` list.

  end for

else

  for each unique feature in `features` **do**

    Filter the dataset to include only the rows where the element equals the current feature.

    Select the "Value", "Year", and "Area" columns, and rename "Value" to the current feature's name.

    Exclude the rows where the "Year" is 2023.

    Group this modified dataset by "Area" and "Year", and sum the values for the current feature.

    Append this resulting dataset to the `minidatasets` list.

  end for

end if

Initialize the final dataset `dataset_generated` with the first dataset in `minidatasets`.

for each remaining dataset in `minidatasets` **do**

  Merge `dataset_generated` with the current dataset of minidatasets, using the specified method (`how`).

end for

Remove any duplicate rows from `dataset_generated`.

Reset the index of `dataset_generated`.

return The final dataset `dataset_generated`.

	Area	Year	Cereals and Preparations	Fats and Oils (excluding Butter)	Meat and Meat Preparations	Sugar and Honey	Fruit and Vegetables	Dairy Products and Eggs	Alcoholic Beverages	Non-alcoholic Beverages
0	Afghanistan	2009	15.00	NaN	NaN	0.00	238846.00	157.00	NaN	NaN
1	Afghanistan	2010	54.00	NaN	NaN	0.00	169878.00	36.00	NaN	NaN
2	Afghanistan	2011	0.00	NaN	NaN	0.00	148362.00	217.00	NaN	NaN
3	Afghanistan	2012	0.00	NaN	NaN	0.00	129926.00	54.00	NaN	NaN
4	Afghanistan	2013	0.00	NaN	NaN	0.00	189275.00	54.00	NaN	NaN
5	Afghanistan	2014	1074.45	995.26	10.34	184.65	284254.63	22.52	39.04	45.25
6	Afghanistan	2015	173.08	25.14	21.78	122.59	280132.89	65.25	66.62	1.70
7	Afghanistan	2016	346.57	0.07	34.86	138.29	318657.69	34.68	8.25	62.58
8	Afghanistan	2017	1628.89	0.32	554.96	100.16	494369.74	59.59	NaN	371.92
9	Afghanistan	2018	591.29	104.00	151.41	463.66	508080.13	70.62	30.94	95.76

Figure 1: Dataset after applying `extraction_features` function. Each column represents a different crop product's export value.

The crop products (items) given are:

- Cereals and Preparations
- Fats and Oils (excluding Butter)
- Meat and Meat Preparations
- Sugar and Honey
- Fruit and Vegetables
- Dairy Products and Eggs
- Alcoholic Beverages
- Non-alcoholic Beverages
- Other food
- Non-food
- Non-edible Fats and Oils
- Tobacco

To obtain the total export value for each country per year, we summed the export values of all crop products, as depicted in Figure 2. This can be mathematically represented by the following equation:

$$\text{Total Export Value} = \sum_{i=1}^{n} \text{Export Value}_{\text{Item}_i} \tag{1}$$

where n is the number of crop products (items) and $\text{Export Value}_{\text{Item}_i}$ represents the export value of the i-th crop product.

	Year	Area	Export Value
0	2009	Afghanistan	320583.00
1	2010	Afghanistan	277114.00
2	2011	Afghanistan	249481.00
3	2012	Afghanistan	211619.00
4	2013	Afghanistan	290137.00
5	2014	Afghanistan	446654.48
6	2015	Afghanistan	403170.24
7	2016	Afghanistan	441584.20
8	2017	Afghanistan	723030.25
9	2018	Afghanistan	644107.50

Figure 2: Subset of final dataset with the total "Export Value" per country per year after summing all crop products.

1.2 The features

Features Extraction

The features extracted from each file and some particular cases that involve performing essential conversions to yearly data, and taking necessary intermediate steps to ensure consistent representation across all files in datasets are shown in this subsection. These 13 individual datasets were then merged into a single dataframe to facilitate the process of selecting which features to drop and which to retain.

File 1: Dataset/Consumer prices indicators - FAOSTAT_data_en_2-22-2024.csv.

This file contains two items: "Consumer Prices, Food Indices (2015 = 100)" and "Food Price Inflation", with monthly values for each. To align with the annual format of other files, the "Value" feature is summed for each month within a year. This results in yearly values instead of monthly values. This conversion is automatically handled by the `extraction_features` function, as explained earlier in this section.

No.	Feature Name
1	Consumer Prices, Food Indices (2015 = 100)
2	Food Price Inflation

Table 1: List of Featurese extracted from file 1.

File 2: Dataset/Crops production indicators - FAOSTAT_data_en_2-22-2024.csv

No.	Feature Name
3	Cereals, primary
4	Citrus Fruit, Total
5	Fibre Crops, Fibre Equivalent
6	Fruit Primary
7	Oilcrops, Cake Equivalent
8	Oilcrops, Oil Equivalent
9	Pulses, Total
10	Roots and Tubers, Total
11	Sugar Crops Primary
12	Treenuts, Total
13	Vegetables Primary

Table 2: List of Features extracted from file 2.

File 3: Dataset/Emissions - FAOSTAT_data_en_2-27-2024.csv

In this file, as well as in files seven and nine, there are multiple entries in the "Element" and "Item" columns. To use the `extraction_features` function properly, this dataset was divided into two separate datasets, each containing one element. Features were then created for each element and item, and these datasets were merged to include all the features from this file in one dataset.

No.	Feature Name
14	Crops total (Emissions N2O)
15	Crops total (Emissions CH4)
16	Cropland Emissions (N2O)
17	Cropland Emissions (CO2)
18	Grassland Emissions (N2O)
19	Grassland Emissions (CO2)

Table 3: List of Features extracted from file 3.

File 4: Dataset/Employment - FAOSTAT_data_en_2-27-2024.csv

No.	Feature Name
20	Mean weekly hours actually worked per employed person in agriculture, forestry and fishing
21	Employment in agriculture, forestry and fishing - ILO modelled estimates

Table 4: List of Features extracted from file 4.

File 5: Dataset/Exchange rate - FAOSTAT_data_en_2-22-2024.csv

No.	Feature Name
22	Local currency units per USD

Table 5: List of Features extracted from file 5.

File 6: Dataset/Fertilizers use - FAOSTAT_data_en_2-27-2024.csv

No.	Feature Name
23	NPK fertilizers
24	Urea
25	Ammonium nitrate (AN)
26	Ammonium sulphate
27	Calcium ammonium nitrate (CAN) and other mixtures with calcium carbonate
28	Diammonium phosphate (DAP)
29	Monoammonium phosphate (MAP)
30	Other NP compounds
31	PK compounds
32	Potassium chloride (muriate of potash) (MOP)
33	Potassium nitrate
34	Potassium sulphate (sulphate of potash) (SOP)
35	Sodium nitrate
36	Superphosphates above 35%
37	Superphosphates, other
38	Ammonia, anhydrous
39	Phosphate rock
40	Urea and ammonium nitrate solutions (UAN)
41	Fertilizers n.e.c.
42	Other nitrogenous fertilizers, n.e.c.
43	Other phosphatic fertilizers, n.e.c.
44	Other potassic fertilizers, n.e.c.
45	Other NK compounds

Table 6: List of Features extracted from file 6.

File 7: Dataset/Food balances indicators - FAOSTAT_data_en_2-22-2024.csv

No.	Feature Name	No.	Feature Name
46	Cereals - Excluding Beer exports	47	Starchy Roots exports
48	Sugar Crops exports	49	Sugar & Sweeteners exports
50	Pulses exports	51	Treenuts exports
52	Oilcrops exports	53	Vegetable Oils exports
54	Vegetables exports	55	Fruits - Excluding Wine exports
56	Stimulants exports	57	Spices exports
58	Alcoholic Beverages exports	59	Meat exports
60	Eggs exports	61	Milk - Excluding Butter exports
62	Fish, Seafood exports	63	Cereals - Excluding Beer imports
64	Starchy Roots imports	65	Sugar Crops imports
66	Sugar & Sweeteners imports	67	Pulses imports
68	Treenuts imports	69	Oilcrops imports
70	Vegetable Oils imports	71	Vegetables imports
72	Fruits - Excluding Wine imports	73	Stimulants imports
74	Spices imports	75	Alcoholic Beverages imports
76	Meat imports	77	Eggs imports
78	Milk - Excluding Butter imports	79	Fish, Seafood imports
80	Cereals - Excluding Beer losses	81	Starchy Roots losses
82	Sugar Crops losses	83	Sugar & Sweeteners losses
84	Pulses losses	85	Treenuts losses
86	Oilcrops losses	87	Vegetables losses
88	Fruits - Excluding Wine losses	89	Spices losses
90	Eggs losses	91	Milk - Excluding Butter losses
92	Meat losses	93	Stimulants losses
94	Alcoholic Beverages losses	95	Vegetable Oils losses
96	Cereals - Excluding Beer other	97	Vegetable Oils other
98	Fish, Seafood other	99	Starchy Roots other
100	Sugar & Sweeteners other	101	Oilcrops other
102	Spices other	103	Alcoholic Beverages other
104	Milk - Excluding Butter other	105	Meat other
106	Stimulants other	107	Pulses other
108	Treenuts other	109	Vegetables other
110	Fruits - Excluding Wine other	111	Eggs other
112	Sugar Crops other	113	Cereals - Excluding Beer food
114	Starchy Roots food	115	Sugar & Sweeteners food
116	Pulses food	117	Treenuts food
118	Oilcrops food	119	Vegetable Oils food

Table 7: List of Features extracted from file 7 (Part 1).

No.	Feature Name	No.	Feature Name
120	Vegetables food	121	Fruits - Excluding Wine food
122	Stimulants food	123	Spices food
124	Alcoholic Beverages food	125	Meat food
126	Eggs food	127	Milk - Excluding Butter food
128	Fish, Seafood food	129	Sugar Crops food

Table 8: List of Features extracted from file 7 (Part 2).

File 8: Dataset/Food security indicators - FAOSTAT_data_en_2-22-2024.csv

There is an inconsistency in the representation of the "Year" column, where it is shown in periods of three years (e.g., 2000-2002, 2001-2003, 2002-2004). To address this, the central year of each period is used as the new value for the "Year" column. Additionally, extra rows are added before and after the first and last periods to include the start and end years.

For this purpose, an algorithm with the pseudocode shown in the Algorithm 3 was implemented. Then, to convert the periods to single years, a function with the pseudocode depicted by Algorithm 2 was also implemented. This function replaces the three-year periods with the central year of each period.

Algorithm 2 Convert Year Periods to Central Year

```
Define function convert_year(year_range)
if year_range contains '-' then
    start, end ← split year_range by '-'
    central_year ← (start + end) // 2
    return central_year
else
    return year_range as integer
end if
```

Algorithm 2 is then applied to each value of the column "Year".

Algorithm 3 Add Rows Before and After Time Periods

```
Make a copy of the original dataset
food_security_indicators_data_copy ← copy of food_security_indicators_data
Initialize an offset counter
offset ← 0
for row in food_security_indicators_data do
    index ← current index + offset
    year ← value of "Year" column in the current row
    if year is "2000-2002" then
        Create a new row with Year ← "2000"
        Insert the new row before the current row in food_security_indicators_data_copy
        Increment offset by 1
    else if year is "2018-2020" AND the next row's "Year" is "2000-2002" then
        Create a new row with Year ← "2020"
        Insert the new row after the current row in food_security_indicators_data_copy
        Increment offset by 1
    else if year is "2019-2021" AND the next row's "Year" is "2000-2002" then
        Create a new row with Year ← "2021"
        Insert the new row after the current row in food_security_indicators_data_copy
        Increment offset by 1
    else if year is "2020-2022" then
        Create a new row with Year ← "2022"
        Insert the new row after the current row in food_security_indicators_data_copy
        Increment offset by 1
    end if
end for
```

The Algorithm 3 adds rows before and after the first and last periods, ensuring that the "Value" column contains the same values as the first or last year.

No.	Feature Name	No.	Feature Name
130	Average dietary energy supply adequacy (percent) (3-year average)	135	Per capita food production variability (constant 2014-2016 thousand int$ per capita)
131	Average protein supply (g/cap/day) (3-year average)	136	Per capita food supply variability (kcal/cap/day)
132	Cereal import dependency ratio (percent) (3-year average)	137	Prevalence of anemia among women of reproductive age (15-49 years)
133	Percent of arable land equipped for irrigation (percent) (3-year average)	138	Prevalence of low birthweight (percent)
134	Value of food imports in total merchandise exports (percent) (3-year average)	139	Political stability and absence of violence/terrorism (index)

Table 9: List of Features extracted from file 8.

File 9: Dataset/Food trade indicators - FAOSTAT_data_en_2-22-2024.csv

In this file, the label is extracted as explained in Section 1.1. The other data (features) extracted from this file is analyzed similarly to the other files.

No.	Feature Name	No.	Feature Name
140	Cereals and Preparations	146	Non-alcoholic Beverages
141	Fats and Oils (excluding Butter)	147	Other food
142	Meat and Meat Preparations	148	Non-food
143	Sugar and Honey	149	Non-edible Fats and Oils
144	Fruit and Vegetables	150	Tobacco
145	Dairy Products and Eggs		

Table 10: List of Features extracted from file 9.

File 10: Dataset/Foreign direct investment - FAOSTAT_data_en_2-27-2024.csv

No.	Feature Name	No.	Feature Name
151	Total FDI inflows	154	FDI inflows to Food, Beverages and Tobacco
152	Total FDI outflows	155	FDI outflows to Agriculture, Forestry and Fishing
153	FDI inflows to Agriculture, Forestry and Fishing	156	FDI outflows to Food, Beverages and Tobacco

Table 11: List of Features extracted from file 10.

File 11: Dataset/Land temperature change - FAOSTAT_data_en_2-27-2024.csv

Temperature change and its standard deviation are represented quarterly. However, the data also includes the annual average of these values when the "Months" column is "Meteorological year." Therefore, the provided annual averages will are used.

This can be easily verified by calculating the mean of the first four quarters of the year 2000 for Afghanistan and comparing it with the meteorological year's data, both of which equal 0.993.

No.	Feature Name
157	Temperature change
158	Standard Deviation

Table 12: List of Features extracted from file 11.

File 12: Dataset/Land use - FAOSTAT_data_en_2-22-2024.csv

No.	Feature Name	No.	Feature Name
159	Country area	170	Permanent crops
160	Land area	171	Permanent meadows and pastures
161	Agriculture	172	Perm. meadows & pastures - Nat. growing
162	Agricultural land	173	Land area equipped for irrigation
163	Cropland	174	Land area actually irrigated
164	Arable land	175	Agriculture area actually irrigated
165	Temporary crops	176	Farm buildings and Farmyards
166	Temporary meadows and pastures	177	Cropland area actually irrigated
167	Temporary fallow	178	Perm. meadows & pastures - Cultivated
168	Permanent crops	179	Perm. meadows & pastures area actually irrig.
169	Permanent meadows and pastures	180	Forestry area actually irrigated

Table 13: List of Features extracted from file 12.

File 13: Dataset/Pesticides use - FAOSTAT_data_en_2-27-2024.csv

It's important to note that "Pesticides (total)" appears to be the sum of the other items and is the only one with significant data. Therefore, It is decided to focus solely on "Pesticides (total)" and its elements (column "Element" values).

No.	Feature Name
181	Pesticides Agricultural Use
182	Pesticides Use per area of cropland
183	Pesticides Use per value of agricultural production

Table 14: List of Features extracted from file 13.

Merge of datasets

All the datasets extracted in this section have been merged using the "outer" method with the pandas library (Python). A subset of the resultant dataset is shown in Figure 3. It has 183 features, and 185 columns in total (due to "Year" and "Area" columns).

	Area	Year	Consumer Prices, Food Indices (2015 = 100)	Food price inflation	Cereals, primary	Citrus Fruit, Total	Fibre Crops, Fibre Equivalent	Fruit Primary	Oilcrops, Cake Equivalent	Oilcrops, Oil Equivalent
0	Afghanistan	2000	319.558176	NaN	8063.0	71245.0	3990.0	76730.0	3833.0	2231.0
1	Afghanistan	2001	358.722573	153.368307	10067.0	71417.0	3990.0	80268.0	3829.0	2217.0
2	Afghanistan	2002	424.138702	219.054193	16698.0	71477.0	3990.0	80174.0	3818.0	2202.0
3	Afghanistan	2003	482.437360	169.226933	14580.0	73423.0	3850.0	82792.0	3844.0	2532.0
4	Afghanistan	2004	550.086733	168.866060	13348.0	78025.0	3843.0	79157.0	3951.0	2716.0
5	Afghanistan	2005	619.263141	151.274875	17904.0	78230.0	3850.0	81541.0	3968.0	2708.0
6	Afghanistan	2006	658.032554	75.664094	15517.0	68560.0	3505.0	85655.0	3633.0	2409.0
7	Afghanistan	2007	739.200903	147.190990	19153.0	72575.0	3505.0	86204.0	3495.0	2225.0
8	Afghanistan	2008	1042.781720	493.637470	14554.0	67500.0	3500.0	84776.0	3371.0	2215.0
9	Afghanistan	2009	903.543503	-145.710833	20407.0	50000.0	4547.0	88478.0	3778.0	2451.0

Figure 3: Subset of the dataframe of the thirteen files.

Features Selection

It is important to note that this process was conducted after the one described in Section 3.1 (Outliers). Removing unusual values first is crucial to ensure a realistic analysis and accurately determine which features are worth retaining.

In Figure 4, a visual representation of the NaN values in the dataset is shown. White areas indicate the presence of NaN values, while black areas represent the presence of data. The vertical axis represents the rows, and the horizontal axis represents the features (not all of them).

Visualisizing NaN values in the entire dataset (heatmap).

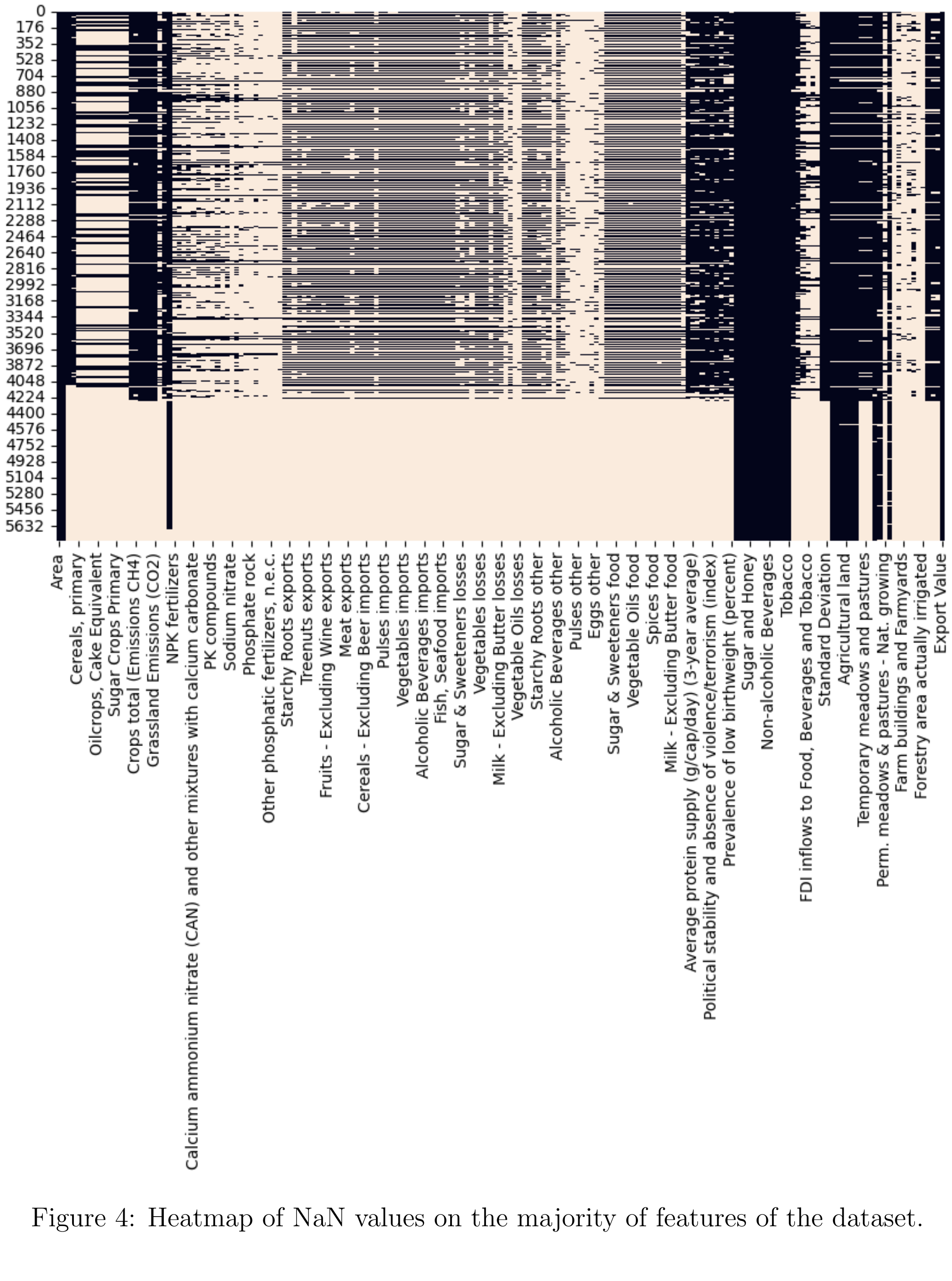

Figure 4: Heatmap of NaN values on the majority of features of the dataset.

Visualizising the impact of each feature on the label (Pearson's Correlation).

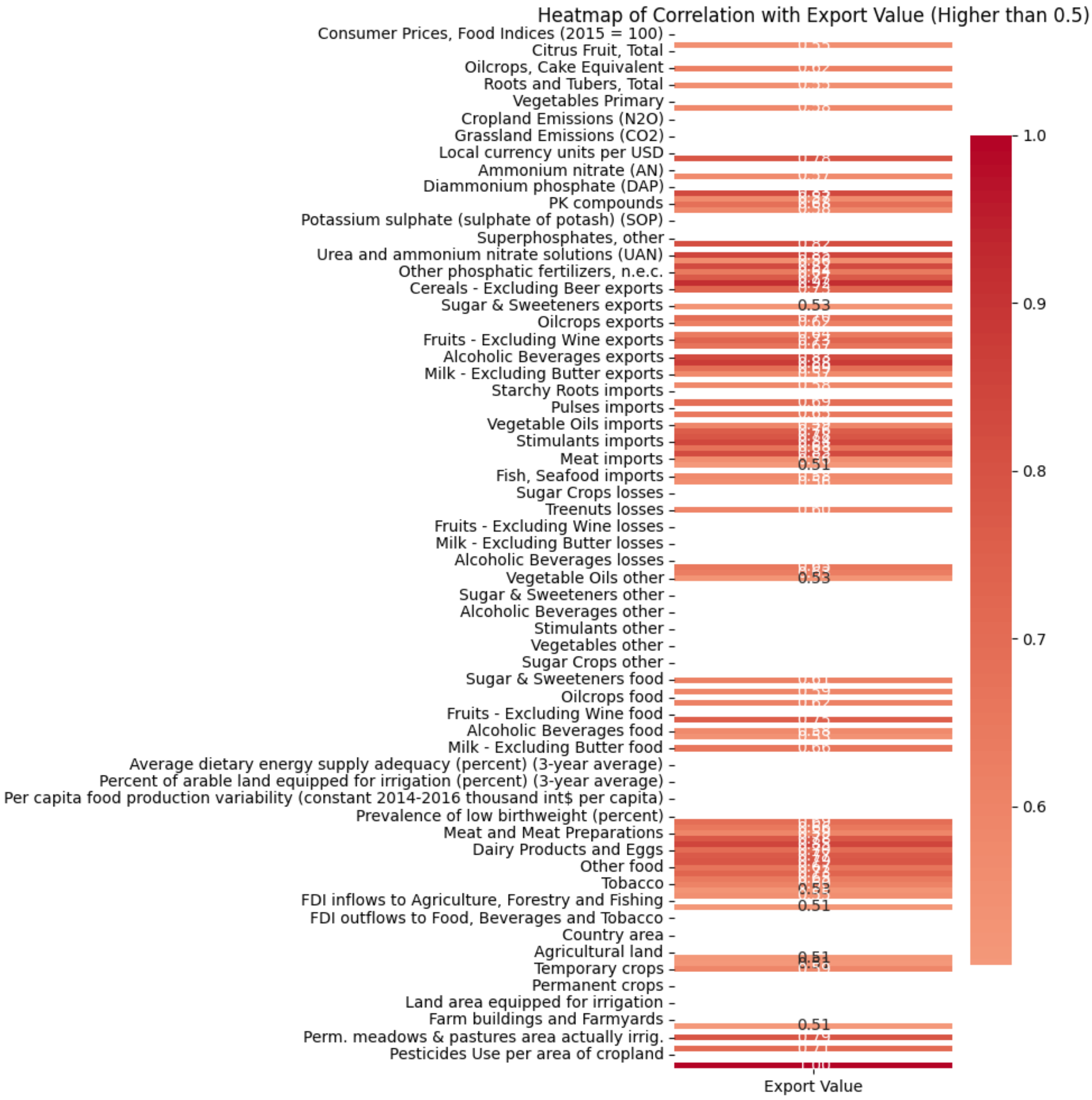

Figure 5: Heatmap of the linear correlation of each feature with the label, filtered to show only correlations greater than 0.5 in red.

Dropping features

Before deciding which features to drop or preserve, it is crucial to examine the NaN values distribution. As shown in Figure 6, some features have most of their NaN values concentrated at the end of the data, while others have high NaN percentages due to a general lack of information.

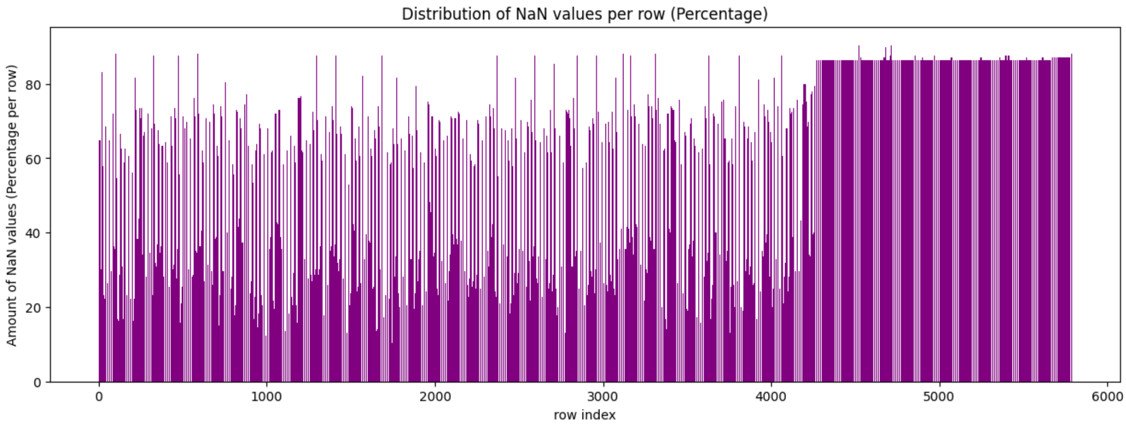

Figure 6: NaN values distribution throughout the rows.

To accurately reflect the percentage of NaN values, percentages are calculated excluding rows with the majority of data missing. Since most features have around 80% NaN values after row 4250, only rows before this point are considered. This approach provides a more reliable percentage of NaN values per feature across the entire dataframe.

Analysis of features with more than 90%.

No features in this range.

Analysis of columns with more than 80% and less than 90% of NaN values.

No features in this range.

Analysis of columns with more than 70% and less than 80% of NaN values.

Due to the excessive lack of information for these features throughout the entire dataframe, they have been dropped.

Feature	Percentage NaN values	Linear correlation
Sodium nitrate	70.47%	0.14
Other NK compounds	70.20%	0.92
Vegetable Oils losses	70.42%	0.65
Treenuts other	70.09%	-0.11
Land area actually irrigated	70.73%	-0.09
Farm buildings and Farmyards	70.40%	0.38
Perm. meadows & pastures area actually irrig.	70.70%	0.79
Forestry area actually irrigated	72.72%	-0.25

Table 15: Percentage of NaN values and linear correlation of features with the label.

Analysis of columns with more than 60% and less than 70% of NaN values.

Every feature have been dropped due to the lack of information.

Feature	Percentage NaN values	Linear correlation
PK compounds	60.25%	0.68
Superphosphates, other	63.44%	0.24
Ammonia, anhydrous	68.73%	0.82
Phosphate rock	69.35%	0.11
Urea and ammonium nitrate solutions (UAN)	63.79%	0.85
Fertilizers n.e.c.	69.13%	0.56
Other nitrogenous fertilizers, n.e.c.	68.68%	0.82
Other phosphatic fertilizers, n.e.c.	69.11%	0.64
Other potassic fertilizers, n.e.c.	68.99%	0.77
Meat losses	62.92%	0.11
Alcoholic Beverages losses	69.14%	0.39
Spices other	69.57%	0.29
Meat other	65.79%	NaN
Stimulants other	68.95%	0.10
Pulses other	69.87%	0.04
Vegetables other	69.42%	0.03
Fruits - Excluding Wine other	65.69%	-0.04
Sugar Crops other	66.60%	0.47
FDI outflows to Agriculture, Forestry and Fishing	61.13%	0.01
FDI outflows to Food, Beverages and Tobacco	61.42%	0.03
Cropland area actually irrigated	63.37%	0.51

Table 16: Percentage of NaN values and linear correlation of features with the label.

Analysis of columns with more than 50% and less than 60% of NaN values.

All these features have been dropped. The highest linear correlation is 0.83, but it has 58.1% NaN values, which could introduce significant bias.

Feature	Percentage NaN values	Linear correlation
Ammonium nitrate (AN)	54.47%	0.40
Ammonium sulphate	51.38%	0.57
Calcium ammonium nitrate (CAN) and other mixtures with calcium carbonate	58.69%	0.41
Diammonium phosphate (DAP)	53.30%	0.30
Monoammonium phosphate (MAP)	58.17%	0.83
Other NP compounds	56.25%	0.57
Potassium chloride (muriate of potash) (MOP)	51.64%	0.58
Potassium nitrate	59.21%	0.35
Potassium sulphate (sulphate of potash) (SOP)	55.48%	0.28
Superphosphates above 35%	57.00%	0.32
Sugar Crops exports	53.92%	0.20
Spices losses	54.80%	0.36
Stimulants losses	53.70%	0.15
Milk - Excluding Butter other	53.04%	0.46
Eggs other	59.24%	0.23
Sugar Crops food	56.74%	0.28
FDI inflows to Food, Beverages and Tobacco	54.42%	0.51
Perm. meadows & pastures - Nat. growing	57.00%	0.24
Agriculture area actually irrigated	55.81%	0.24
Perm. meadows & pastures - Cultivated	58.64%	0.35

Table 17: Percentage of NaN values and linear correlation of features with the label.

Analysis of columns with more than 40% and less than 50% of NaN values.

All these features will be dropped. Even the highest correlation does not justify the potential bias introduced by imputing data with such high percentages of NaN values.

Feature	Percentage NaN values	Linear correlation
Cereals, primary	49.12%	0.55
Citrus Fruit, Total	49.12%	0.38
Fibre Crops, Fibre Equivalent	49.12%	0.29
Fruit Primary	49.12%	0.28
Oilcrops, Cake Equivalent	49.12%	0.62
Oilcrops, Oil Equivalent	49.12%	0.11
Pulses, Total	49.12%	0.26
Roots and Tubers, Total	49.12%	0.55
Sugar Crops Primary	49.12%	0.16
Treenuts, Total	49.12%	0.12
Vegetables Primary	49.12%	0.41
Mean weekly hours actually worked per employed person in agriculture, forestry and fishing	44.19%	0.11
NPK fertilizers	48.57%	0.78
Urea	45.78%	0.24
Pulses exports	41.53%	0.36
Treenuts exports	40.41%	0.70
Eggs exports	47.18%	0.69
Sugar Crops imports	49.33%	0.25
Sugar Crops losses	46.13%	0.38
Sugar & Sweeteners losses	48.29%	0.32
Treenuts losses	48.48%	0.60
Milk - Excluding Butter losses	42.64%	0.22
Cereals - Excluding Beer other	40.46%	0.63
Starchy Roots other	44.47%	0.30
Sugar & Sweeteners other	48.39%	0.42
Oilcrops other	45.18%	0.24
Alcoholic Beverages other	40.19%	0.36
FDI inflows to Agriculture, Forestry and Fishing	45.97%	0.21

Table 18: Percentage of NaN values and linear correlation of features with the label.

Analysis of columns with more than 30% and less than 40% of NaN values.

Features with a correlation higher than 0.69 will be retained. This is because the strong correlations (e.g., 0.83, 0.88, 0.78, among others) could justify careful imputation.

Feature	Percentage NaN values	Linear correlation
Cereals - Excluding Beer exports	37.80%	0.73
Starchy Roots exports	39.75%	0.36
Sugar & Sweeteners exports	38.23%	0.53
Oilcrops exports	38.13%	0.62
Vegetable Oils exports	38.04%	0.33
Vegetables exports	37.72%	0.64
Fruits - Excluding Wine exports	37.92%	0.73
Stimulants exports	38.04%	0.67
Spices exports	38.56%	0.36
Alcoholic Beverages exports	38.51%	0.83
Meat exports	39.51%	0.88
Milk - Excluding Butter exports	39.84%	0.57
Fish, Seafood exports	37.25%	0.47
Cereals - Excluding Beer imports	37.25%	0.58
Starchy Roots imports	37.25%	0.40
Sugar & Sweeteners imports	37.25%	0.69
Pulses imports	37.28%	0.32
Treenuts imports	37.42%	0.65
Oilcrops imports	37.37%	0.36
Vegetable Oils imports	37.25%	0.59
Vegetables imports	37.25%	0.76
Fruits - Excluding Wine imports	37.25%	0.78
Stimulants imports	37.25%	0.84
Spices imports	37.28%	0.68
Alcoholic Beverages imports	37.25%	0.83
Meat imports	37.25%	0.56
Eggs imports	37.56%	0.51
Milk - Excluding Butter imports	37.25%	0.47
Fish, Seafood imports	37.25%	0.58

Table 19: Percentage of NaN values and linear correlation of features with the label.

Feature	Percentage NaN values	Linear correlation
Cereals - Excluding Beer losses	37.96%	0.56
Starchy Roots losses	37.54%	0.27
Pulses losses	39.36%	0.21
Oilcrops losses	38.48%	0.23
Vegetables losses	37.25%	0.33
Fruits - Excluding Wine losses	37.49%	0.40
Eggs losses	37.92%	0.35
Vegetable Oils other	38.18%	0.53
Fish, Seafood other	37.25%	0.27
Cereals - Excluding Beer food	37.25%	0.37
Starchy Roots food	37.25%	0.37
Sugar & Sweeteners food	37.25%	0.61
Pulses food	37.25%	0.21
Treenuts food	37.32%	0.59
Oilcrops food	37.32%	0.41
Vegetable Oils food	37.25%	0.62
Vegetables food	37.25%	0.34
Fruits - Excluding Wine food	37.25%	0.48
Stimulants food	37.25%	0.75
Spices food	37.25%	0.20
Alcoholic Beverages food	37.46%	0.58
Meat food	37.25%	0.53
Eggs food	37.25%	0.40
Milk - Excluding Butter food	37.25%	0.66
Fish, Seafood food	37.25%	0.38

Table 20: Percentage of NaN values and linear correlation of features with the label.

Analysis of columns with more than 20% and less than 30% of NaN values.

No features in this range.

Analysis of columns with more than 10% and less than 20% of NaN values.

All the features will be preserved and imputed. The lack of information is manageable and can be addressed effectively.

Feature	Percentage NaN values	Linear correlation
Average protein supply (g/cap/day) (3-year average)	11.68%	0.35
Cereal import dependency ratio (percent) (3-year average)	14.53%	-0.23
Per capita food production variability (constant 2014-2016 thousand int$ per capita)	10.61%	0.14
Per capita food supply variability (kcal/cap/day)	10.18%	-0.14
Prevalence of anemia among women of reproductive age (15-49 years)	12.63%	-0.33
Prevalence of low birthweight (percent)	19.70%	-0.23
Total FDI outflows	15.10%	0.55

Table 21: Percentage of NaN values and linear correlation of features with the label.

Analysis of columns with more than 0% and less than 10% of NaN values.

In this case, all the features will be preserved as they have a substantial amount of available data. Even if the correlation does not indicate a high impact on the label, it will be managed accordingly.

Feature	Percentage NaN values	Linear correlation
Consumer Prices, Food Indices (2015 = 100)	2.83%	-0.01
Food price inflation	5.81%	-0.03
Crops total (Emissions N2O)	5.29%	0.58
Crops total (Emissions CH4)	6.43%	0.29
Cropland Emissions (N2O)	3.18%	0.35
Cropland Emissions (CO2)	3.18%	0.24
Grassland Emissions (N2O)	3.18%	0.34
Grassland Emissions (CO2)	3.18%	0.42
Employment in agriculture, forestry and fishing - ILO modelled estimates	6.34%	0.22
Local currency units per USD	2.59%	-0.00

Table 22: Percentage of NaN values and linear correlation of features with the label.

Feature	Percentage NaN values	Linear correlation
Average dietary energy supply adequacy (percent) (3-year average)	6.46%	0.37
Percent of arable land equipped for irrigation (percent) (3-year average)	9.21%	0.11
Value of food imports in total merchandise exports (percent) (3-year average)	7.17%	-0.11
Political stability and absence of violence/terrorism (index)	9.19%	0.14
Total FDI inflows	2.38%	0.53
Country area	3.18%	0.43
Land area	3.18%	0.43
Agriculture	3.52%	0.46
Agricultural land	3.52%	0.46
Cropland	3.52%	0.51
Arable land	3.52%	0.51
Temporary crops	6.62%	0.59
Temporary meadows and pastures	6.62%	0.28
Temporary fallow	6.62%	0.34
Permanent crops	4.28%	0.38
Permanent meadows and pastures	6.13%	0.39
Land area equipped for irrigation	8.38%	0.37
Pesticides Agricultural Use	5.08%	0.71
Pesticides Use per area of cropland	8.72%	0.08
Pesticides Use per value of agricultural production	9.83%	0.06

Table 23: Percentage of NaN values and linear correlation of features with the label.

Selected features for the model.

In total, fifty one features were chosen. They can be observed in Table 24.

Feature	Feature
Consumer Prices, Food Indices (2015 = 100)	Average dietary energy supply adequacy (percent) (3-year average)
Food price inflation	Average protein supply (g/cap/day) (3-year average)
Crops total (Emissions N2O)	Cereal import dependency ratio (percent) (3-year average)
Crops total (Emissions CH4)	Percent of arable land equipped for irrigation (percent) (3-year average)
Cropland Emissions (N2O)	Value of food imports in total merchandise exports (percent) (3-year average)
Cropland Emissions (CO2)	Political stability and absence of violence/terrorism (index)
Grassland Emissions (N2O)	Per capita food production variability (constant 2014-2016 thousand int$ per capita)
Grassland Emissions (CO2)	Per capita food supply variability (kcal/cap/day)
Employment in agriculture, forestry and fishing - ILO modelled estimates	Prevalence of anemia among women of reproductive age (15-49 years)
Local currency units per USD	Prevalence of low birthweight (percent)
Cereals and Preparations	Fats and Oils (excluding Butter)
Meat and Meat Preparations	Sugar and Honey
Fruit and Vegetables	Dairy Products and Eggs
Alcoholic Beverages	Non-alcoholic Beverages
Other food	Non-food
Non-edible Fats and Oils	Tobacco
Total FDI inflows	Total FDI outflows
Temperature change	Standard Deviation
Country area	Land area
Agriculture	Agricultural land
Cropland	Arable land
Temporary crops	Temporary meadows and pastures
Temporary fallow	Permanent crops
Permanent meadows and pastures	Land area equipped for irrigation
Pesticides Agricultural Use	Pesticides Use per area of cropland
Pesticides Use per value of agricultural production	

Table 24: List of Selected Features

Dropping rows

As seen in Figures 4 and 6, there is an exaggerated concentration of NaN values towards the end of the dataset. The rows from 4250 to the end contain approximately 80% NaN values, so they have been dropped.

Furtheremore, Table 25 shows the range of NaN values, the number of rows with this percentage of NaN values, and their proportion of the entire dataset (excluding rows from 4250 onwards).

Range of NaN values	Number of Rows	Percentage of Dataset
80%-90%	0	0%
70%-80%	0	0%
60%-70%	40	0.8791%
50%-60%	144	3.1648%
40%-50%	3	0.0659%
30%-40%	60	1.3186%

Table 25: Number of Rows and Percentage of Dataset

Based on the percentage of NaN values per row and their proportion of the entire dataset, rows with 40% to 80% NaN values will be dropped. This represents approximately 4% of the dataset, which is still not significant. Rows remaining: 4065.

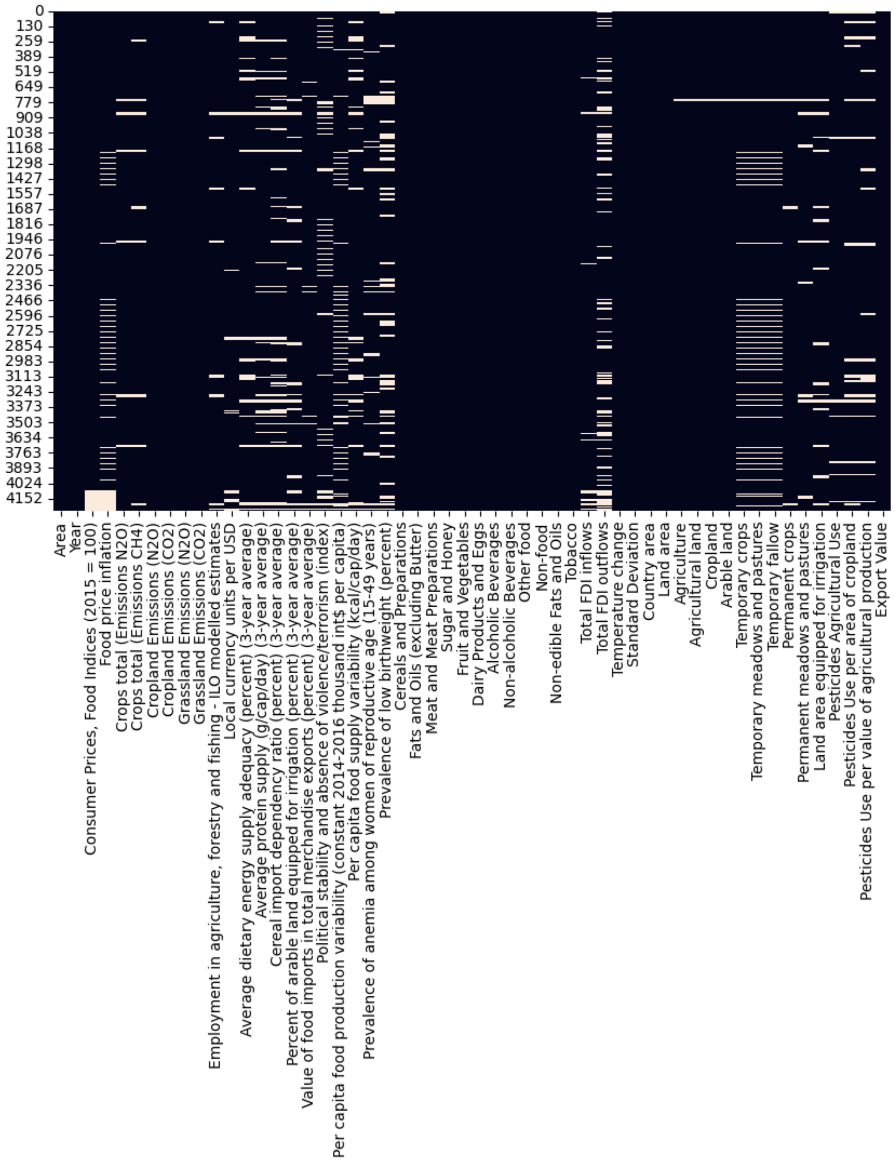

Figure 7: Heatmap of NaN values after dropping features and rows.

2 Preprocessing

2.1 Outliers

In this case, it is fundamental recognisize that boxplots will show many apparent outliers. However, this is due to the nature of the data, which is a collection of multiple features in different countries, where one country could keep a very high magnitude in one feature, while other country a very low magnitud in the same feature.

Due to this, a double analysis have been carried out. One approach is the one shown below:

1. **Divide the dataset**: Filter the original dataset to obtain sub-datasets for each country.

2. **Apply Interquartile Range Method (IQR) to each subset**: Delete outliers for each country separately.

3. **Recombine the sub-datasets**: After deleting the outliers for each country, recombine the sub-datasets.

The second approach is use the original features on the dataset together, using the Interquantile Range method.

Interquantile Range method.

Mathematically, the Interquartile Range (IQR) is the difference between the third quartile (Q3) and the first quartile (Q1) of a dataset. Quartiles divide a dataset into four equal parts, and are defined as follows:

- $Q1$ (first quartile): the value that separates the first 25% of the ordered data from the rest.

- $Q3$ (third quartile): the value that separates the first 75% of the ordered data from the rest.

The Interquartile Range is calculated as:

$$\text{IQR} = Q3 - Q1 \tag{2}$$

To identify outliers, the lower and upper limits are defined using the IQR. Values falling outside these limits are considered outliers.

Calculation of the limits

Lower limit

$$\text{Lower Limit} = Q1 - 1.5 \times \text{IQR} \tag{3}$$

Upper limit

$$\text{Upper Limit} = Q3 + 1.5 \times \text{IQR} \tag{4}$$

Any value in the dataset that is less than the lower limit or greater than the upper limit is considered an outlier.

Interquantile Range method to deal with outliers of each country.

Unfortunately, 49.01%, 56.66% and 71.66% of the data is missed when the coefficient parameter is 3, 2.5 and 1.5, respectively. Therefore, it was decided to try a general outlier identification on the original dataset, instead of dividing it by country.

Interquantile Range method to deal with outliers (Original dataset).

The interquartile Range method was used with different coefficients. Specifically with 1.5, 2 and 3, The number of rows eliminated were 74.82%, 68.81% and 59.29%. This led to the decision to manually drop the outliers by visualizing the box plots and histograms.

Manually dropping Outliers.

Significant data loss occurs when potential outliers are removed based on histograms and box plots (too many to put them on this report). Therefore, no outliers will be dropped to avoid losing potentially important information and due to time constraints for individual feature analysis.

2.2 Imputation

As shown in Figure 8, a Pearson's Correlation (Correlation Matrix) is depicted by a heatmap, revealing high linear correlation among features. Therefore, a regression model was chosen to predict the NaN values, leveraging the high correlation for accurate imputation (which is usually more precise than using basic methods as mean or median). For this, `IterativeImputer` from sklearn was used.

Figure 8: Heatmap of Pearson's Correlation (filtered higher than 0.5).

By default, the `IterativeImputer` uses the mean, median, or a similar strategy to initially impute the missing values. Then, it iterates over each feature with missing values, using it as the target to train a linear regression model (in this case) with the other features. It predicts and replaces the missing values of the target feature, ignoring its own imputation in that step. This process is repeated for all features until convergence [6] . A pseudocode is provided by Algorithm 4.

Algorithm 4 Iterative Imputer using Linear Regression

1: Make an initial imputation of missing values using a simple strategy (e.g., mean or median).
2: **while** not converged and iteration < max_iterations **do**
3:     **for** each feature with missing values **do**
4:         Select the feature as the target variable (y).
5:         Select the other features as independent variables (X).
6:         Train a linear regression model on (X, y) using only the rows where y is not missing.
7:         Predict the missing values of y using the trained model.
8:         Replace the missing values with the predicted values.
9:     **end for**
10: **end while**

	Consumer Prices, Food Indices (2015 = 100)	Food price inflation	Crops total (Emissions N2O)	Crops total (Emissions CH4)	Cropland Emissions (N2O)	Cropland Emissions (CO2)	Grassland Emissions (N2O)	Grassland Emissions (CO2)
0	319.558176	211.958848	0.7056	20.8471	0.0	0.0	0.0	0.0
1	358.722573	153.368307	0.7054	19.2605	0.0	0.0	0.0	0.0
2	424.138702	219.054193	1.0656	21.2553	0.0	0.0	0.0	0.0
3	482.437360	169.226933	1.3117	23.7017	0.0	0.0	0.0	0.0
4	550.086733	168.866060	1.0856	30.3089	0.0	0.0	0.0	0.0
5	619.263141	151.274875	1.5382	25.8752	0.0	0.0	0.0	0.0
6	658.032554	75.664094	1.3433	25.6535	0.0	0.0	0.0	0.0
7	739.200903	147.190990	1.6075	27.0921	0.0	0.0	0.0	0.0
8	1042.781720	493.637470	1.1634	29.5674	0.0	0.0	0.0	0.0
9	903.543503	-145.710833	1.7949	31.4614	0.0	0.0	0.0	0.0

Figure 9: Imputed Dataset.

2.3 Normalization/Scaling

In this step, the `StandardScaler()` object from `sklearn` was used and the `.fit_transform()` function was applied. This function performs two steps:

1. **Fitting the Scaler (fit)**

First, the scaler calculates the mean (μ) and standard deviation (σ) of each feature (column) in the dataset $X\_imputed\_lr$.

For a feature j:

$$\mu_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij} \tag{5}$$

$$\sigma_j = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_{ij} - \mu_j)^2} \tag{6}$$

where:
x_{ij} is the value of the j-th feature for the i-th sample.
n is the total number of samples.

2. **Transforming the Data (transform)**

Next, the scaler transforms each value of feature j in the dataset $X\_imputed\_lr$ using the mean and standard deviation calculated in the previous step. The transformation is performed with the following formula [5]:

$$x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{7}$$

where:
x'_{ij} is the scaled value of the j-th feature for the i-th sample.
x_{ij} is the original value of the j-th feature for the i-th sample.
μ_j is the mean of the j-th feature.
σ_j is the standard deviation of the j-th feature.

The result is a scaled dataset $X_{\text{scaled\_lr}}$ where each feature has a mean of 0 and a standard deviation of 1.

In this case, just the features are normalized.

2.4 Principal Component Analysis (PCA)

With the features normalized, dimensionality reduction was performed using PCA to explain 95% of the total variance in the dataset. PCA transforms the original data into a new set of variables called principal components. These components are ordered such that the first captures the most variance, the second captures the most remaining variance, and so on. The explained variance by a principal component is the proportion of the total information in the original data represented by that component. In this case, 95% of the original data's information is aimed to be captured by the principal components [4] .

The explained variance for the k-th principal component is defined as:

$$\text{Explained Variance} = \frac{\lambda_k}{\sum_{i=1}^{p} \lambda_i}$$

where λ_k is the eigenvalue associated with the k-th principal component and p is the total number of original features.

The cumulative explained variance up to the k-th principal component is:

$$\text{Cumulative Explained Variance} = \sum_{i=1}^{k} \frac{\lambda_i}{\sum_{i=1}^{p} \lambda_i}$$

As shown in the information source for this section, the PCA object from scikit-learn was used to implement this. The Figure below show the explained variance along each component:

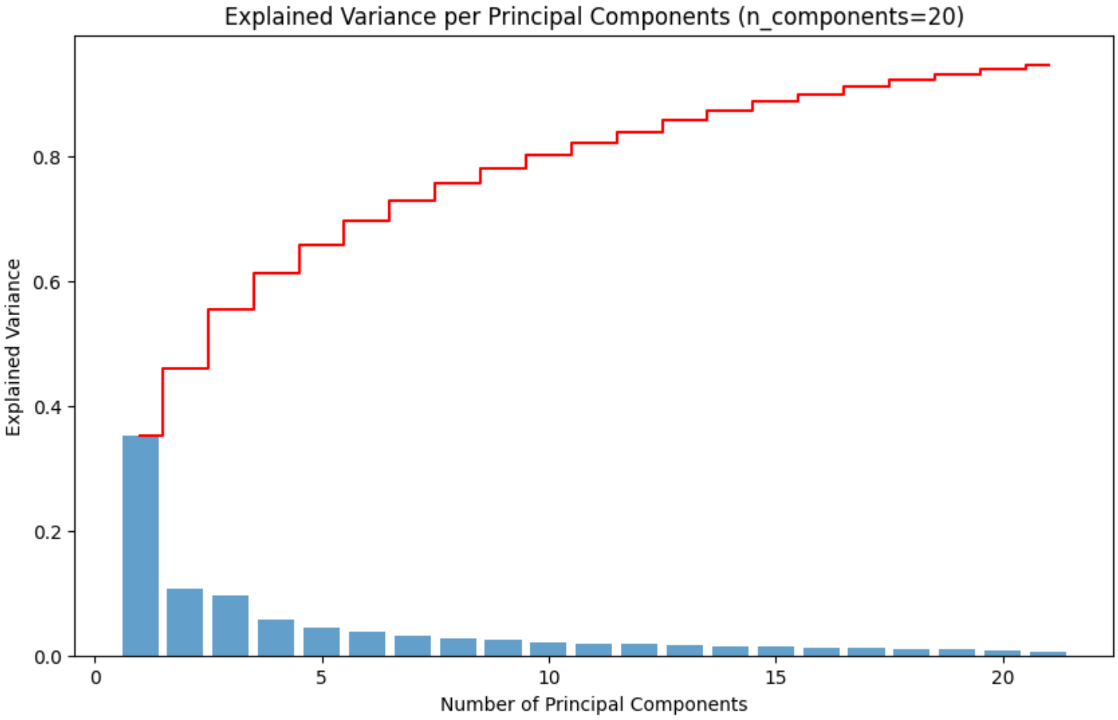

Figure 10: Explained variance throughout principal components.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
0	-1.535470	1.291266	-0.750724	0.137767	-2.140455	-0.532107	0.482078
1	-1.193337	1.322027	-0.956970	0.293983	-1.427179	-0.329719	0.153415
2	-1.250720	1.384482	-1.011676	0.103266	-1.859350	-0.541522	-0.037955
3	-1.292185	1.394197	-0.762238	0.123750	-1.816574	-0.489314	0.397510
4	-1.171728	1.324854	-0.709230	0.087486	-1.804264	-0.427156	-0.215365
5	-1.158968	1.411046	-0.674993	0.300119	-1.744745	-0.155409	0.644994
6	-1.108421	1.158409	-0.693646	0.088675	-1.756211	-0.447164	-0.497750
7	-1.121754	1.310965	-0.671412	0.328308	-1.831654	-0.053480	0.496116
8	-1.078301	1.377950	-0.376745	0.282526	-1.958777	-0.090329	0.444653
9	-1.119969	1.104745	-1.042413	0.285659	-1.820377	-0.096929	0.268498

Figure 11: Subset of the dataset with Dimensionality Reduction (PCA).

3 Multilayer Perceptron Model

3.1 Neurons and Hidden Layers

The specific structure of hidden layers and neurons was defined:

- **First hidden layer**: 300 neurons
- **Second hidden layer**: 300 neurons
- **Third hidden layer**: 300 neurons

This is the best model found with different combinations perfomed using GridSearchCV, as shown in Section 3.4.

3.2 Activation Function

The activation function used for the hidden layers was the Rectified Linear Unit (ReLU). This choice was optimized using `GridSearchCV`, which found that ReLU outperformed both the tanh and sigmoid functions in 100% of the cases with any combination of parameters for this dataset.

The ReLU function is defined mathematically as:

$$\mathrm{ReLU}(x) = \max(0, x) \tag{8}$$

In simple terms, ReLU works by outputting the input directly if it is positive; otherwise, it outputs zero. This helps to introduce non-linearity into the model while being computationally efficient.

ReLU was expected to be most effective in this case because is considered a common choice for regression tasks. It helps mitigate the vanishing gradient problem and accelerates convergence during training. According to Glorot et al. (2011), ReLU enables the model to learn faster and perform better than traditional activation functions like sigmoid and tanh in deep learning models [1] .

3.3 Loss Function

The loss function used to train the model was the Mean Squared Error (MSE). Although `GridSearchCV` uses `neg_mean_squared_error`, which is simply the MSE multiplied by -1, as `GridSearchCV` aims to maximize the scoring metric, this does not affect the interpretation of the MSE itself [2] .

The Mean Squared Error (MSE) was chosen for one main reason: It penalizes larger errors more severely than smaller ones. This property is useful in applications where larger errors are particularly undesirable, especially given the nature and magnitudes of the Export Values, which tend to be in the millions.

The MSE measures the average squared difference between the actual and predicted values, and is defined as:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{9}$$

where:

- y_i are the actual values.
- $\hat{y}_i$ are the predicted values.
- n is the total number of samples.

3.4 Optimization with GridSearchCV

The model was trained using `GridSearchCV`, which performs cross-validation to evaluate the model with a series of different hyperparameters and determine the best combination.

The following hyperparameter grid was used (with numeric random values):

```
param_grid = {
    'hidden_layer_sizes': [(600, 600, 600)],
    'activation': ['relu'],
    'max_iter': [300],
    'alpha': [0.001, 0.01, 0.1],
    'learning_rate_init': [0.001, 0.01, 0.1]
}
```

The `hidden_layer_sizes` parameter defines the number of neurons in each hidden layer. The `activation` parameter specifies the activation function for the hidden layers. The `max_iter` parameter sets the maximum number of iterations for the solver. The `alpha` parameter represents the L2 regularization term, and `learning_rate_init` sets the initial learning rate for weight updates.

3.5 Preventing Overfitting

To prevent overfitting, several strategies were employed:

Regularization

As seen in Figures 8 and 12, many features are highly correlated, suggesting that most features play a significant role in the prediction. Therefore, L2 regularization (Ridge) was applied because it keeps the features while preventing the coefficients from becoming zero and also handles multicollinearity effectively.

$$\text{L2 Penalty} = \lambda \sum_{j=1}^{p} \theta_j^2$$

where:

- λ is the regularization parameter (alpha) controlling the strength of the penalty.
- θ_j are the model coefficients.
- p is the number of features.

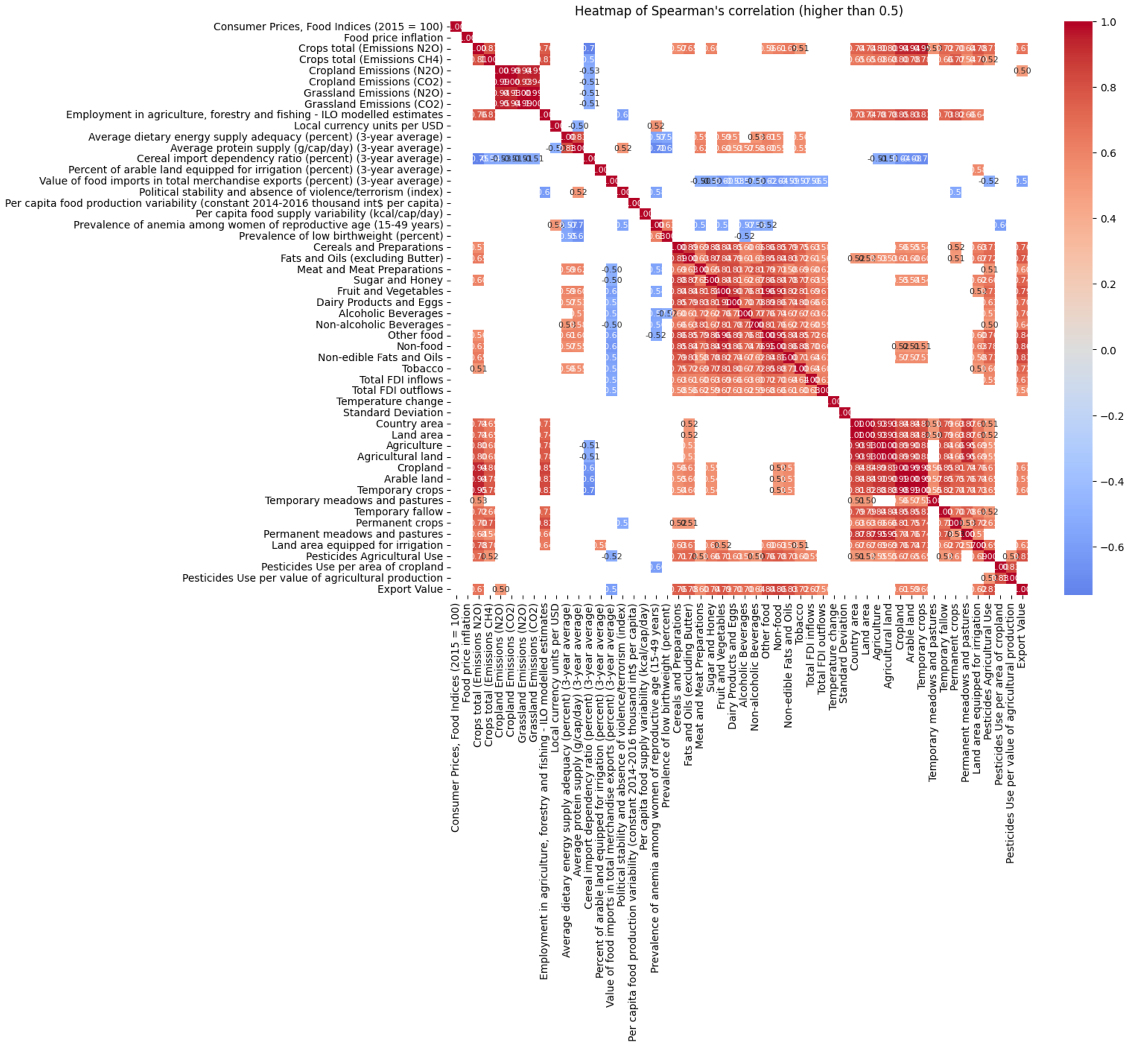

Figure 12: Heatmap of Spearman's Correlation (filtered higher than 0.5).

Cross-Validation

Cross-validation was used to ensure that the model generalizes well to unseen data. `GridSearchCV` was employed to perform cross-validation and select the best hyperparameters.

Limiting the Number of Iterations

The number of training iterations was limited to prevent the model from overfitting to the training data. This was done by setting the `max_iter` parameter in the MLPRegressor.

4 Performance

4.1 Total Number of Instances

- The total number of instances used: 4065
- Number of instances used in the training set: 4053
- Number of instances used in the test set: 4

4.2 Derivation of Training and Test Sets

The training and test sets were derived from the given data using the following process:

1. **Features test set**: The preprocessed dataset after PCA was temporarily merged with the "Year" and "Area" columns from the original dataset. Rows matching the selected countries (Mexico, Germany, Australia, and Argentina) and years (2019, 2020, 2021) were extracted to form the test set. From the extracted rows, only the data from the year 2021 was kept to form the final test set.
2. **Features training set**: The "Export Value" column was filtered from the dataset before applying PCA to include only the rows where the year was 2021 and the countries were the selected ones.
3. **Label test set**: The remaining rows, excluding those used for the test set, were used to form the training set. Then, it was scaled.
4. **Label training set**: It was filtered from the imputed dataset to exclude the rows used by the label test set. Then, it was scaled.

4.3 Evaluation

The performance of the model was evaluated using the following metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). The formulas for these metrics are as follows:

1. Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where y_i are the actual values, $\hat{y}_i$ are the predicted values, and n is the total number of samples.

2. Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where y_i are the actual values, $\hat{y}_i$ are the predicted values, and n is the total number of samples.

3. Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where y_i are the actual values, $\hat{y}_i$ are the predicted values, and n is the total number of samples.

The performance of the model

Hyperparameter	Values
Hidden Layer Sizes	3
Neurons Per Layer	300
Activation	relu
Max Iterations	300
Alpha	0.01
Learning Rate Init	0.001

Table 26: Hyperparameters

Metric	Value
Mean Squared Error (MSE)	0.6338
Root Mean Squared Error (RMSE)	0.7961
Mean Absolute Error (MAE)	0.6471

Table 27: Performance Metrics

	Actual (scaled)	Predicted (scaled)
0	2.062168	1.953829
1	2.378977	2.089917
2	5.437992	6.391813
3	2.137488	0.900487

Table 28: Comparison of Actual and Predicted Values (Scaled)

	Actual (original)	Predicted (original)	Error Percentage (%)
0	38,143,627.59	36,455,630.23	4.42%
1	43,079,750.56	38,575,977.64	10.44%
2	90,741,554.61	105,602,809.57	16.38%
3	39,317,163.59	20,043,745.81	49.03%

Table 29: Comparison of Actual and Predicted Values (Original) and Error (%)

Bibliography

[1] Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 315-323.

[2] Scikit-learn 0.24.2 documentation. GridSearchCV. Retrieved from `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`

[3] Scikit-learn 0.24.2 documentation. GridSearchCV. Retrieved from `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.htm`

[4] Scikit-learn 0.24.2 documentation. PCA. Retrieved from `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.htm`

[5] Scikit-learn 0.24.2 documentation. StandardScaler. Retrieved from `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html`

[6] Scikit-learn 0.24.2 documentation. IterativeImputer. Retrieved from `https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html`