Iteration 3 - OSAS
Open-Source Analytics Solution

Student Name	Mengzhe Zhao
Student ID	219258024
Course Name	Data Mining and Big Data
Course Code	INFOSYS 722
Assignment Title	Iteration 3
Date	22nd September 2023

Contents

1. Business understanding

1.1 Identify the objectives of the business

Agrifood systems encompass various stages of the agricultural value chain, including the production of both food and non-food agricultural products. These stages involve food storage, aggregation, post-harvest handling, transportation, processing, distribution, marketing, disposal, and consumption. Food systems within agrifood systems encompass a wide range of food products derived from various sources, including crop and livestock production, forestry, fisheries, aquaculture, and synthetic biology, with the primary purpose of being consumed by humans (‘Agrifood Systems’, 2023).

Agrifood system has three elements:

- Primary production, which encompasses both agricultural and non-agricultural food sources and non-food agricultural products that function as inputs for other industries.
- Food distribution, which connects production with consumption through supply chains and domestic transport networks. Food supply chains encompass a comprehensive range of participants and processes engaged in the post-harvest management, storage, consolidation, transportation, transformation, dissemination, and commercialization of food products.
- Household consumption, as a consequence of operational agrifood systems, which is susceptible to different levels of demand shocks, such as a decrease in income, contingent upon the prevalence of vulnerable segments within the population. As the proportion increases, safeguarding food security and nutrition from shocks becomes increasingly challenging.

Agrifood systems substantially impact anthropogenic greenhouse gas (GHG) emissions, accounting for approximately one-third of the overall emission (LavagnedOrtigue, n.d.). The emissions in question are derived from many sources, encompassing on-farm activities that pertain to the cultivation of crops and the rearing of livestock. Moreover, alterations in land use, such as deforestation and the drainage of peatlands to facilitate agricultural expansion, are significant contributors to greenhouse gas (GHG) emissions. In addition, emissions are also produced throughout the pre-and post-production phases, which include activities such as food manufacturing, retail operations, household consumption, and food disposal procedures (LavagnedOrtigue, n.d.).

This study is with the following objectives:

- Deeply understand the environmental impact, focusing on climate change and

global warming, from the agri-food industry.

- Provide evidence of policy setting to reduce the CO2 emissions from the agri-food sector.

1.2 Assess the situation

1.2.1 Resource inventory

The programming language, Python, used for this project is from the website www.python.org. The open-source package and environment management system, Anaconda, is from the website www.anaconda.com. The datasets used for this project are from www.kaggle.com/datasets. All references are from the websites: www.nzagrc.org.nz, www.fao.org, www.iaea.org, and www.beehive.govt.nz.

1.2.2 Requirements, assumptions, and constraints

Agricultural departments or organisations responsible for policymaking may benefit from establishing a dedicated data science team to undertake data mining and analysis tasks. Alternatively, they could consider engaging the services of a data mining consulting company to provide the necessary technical expertise.

From a database security standpoint, when data mining tasks are outsourced to a consulting firm, it becomes necessary for the consulting company to gain access to the backend database system. Ensuring the database system's security is paramount for agricultural organisations.

Economic factors significantly influence the outcome of the data mining project. The consideration of consulting fees and the comparative costs of competing products may play a significant role in determining whether to establish an internal team or seek the services of a consulting firm. Budgetary limitations may influence the decision-making process.

Assumptions regarding the quality of data play a pivotal role. The availability, accuracy, and integration of emissions, temperature, and agricultural data influence the reliability of the analysis. The resolution of data gaps and inconsistencies is of utmost importance. A specific assumption is that all agri-food factors are independent of the average temperature rise for implementing a linear regression model.

Gaining insight into the perspective of the project sponsor or management team is crucial. Are they interested in a comprehensive understanding of the data mining model, or are they primarily focused on obtaining practical and implementable outcomes?

Adapting communication strategies to align with individuals' areas of expertise is crucial for facilitating optimal decision-making processes. Achieving a successful project is contingent

upon the careful consideration and management of various factors, including the harmonisation of economic constraints, the dependability of data, and the fulfilment of stakeholder expectations.

In data access, it is imperative to acquire passwords for essential data sources to facilitate uninterrupted analysis. It is imperative to adhere to data security protocols. In the context of legal limitations, it is imperative to ascertain data usage rights and adhere to regulatory frameworks to mitigate potential legal complications and safeguard against privacy breaches. Concerning financial limitations, it is imperative to develop a comprehensive project budget that encompasses all expenditures, such as consulting fees, tool expenses, and any unforeseen costs that may arise. By considering these factors, data access protection, adherence to legal requirements, and preservation of budgetary integrity are ensured, facilitating a seamless and compliant project implementation.

1.2.3 Risks and contingencies

Regarding risk management, exercising control over consulting fees within the project budget is imperative. In addition to this, it is crucial to consider the cost of time, as policy formulation is frequently intertwined with strategic planning and the annual report. Data risks, such as inadequate data quality or coverage, can compromise the accuracy of analysis. Implementing rigorous data validation and preparation protocols is imperative to address this concern effectively. The management of potential risks associated with the outcomes, such as the possibility of less influential preliminary findings, can be effectively addressed by implementing transparent communication strategies. Effectively managing stakeholder expectations can be achieved by contextually presenting findings and emphasising the potential for further insights as the analysis progresses. It is imperative to ensure meticulous and comprehensive scheduling of the project.

1.3 Determine data mining objectives

With the help of a particular data mining team or a consulting company, the business objectives can be transferred to data mining objectives. The data mining goals of this project to be completed are the following:

- Examine the correlation between carbon dioxide (CO2) emissions within the agri-food sector and the subsequent temperature rise.
- Analyse the influence of various countries based on aggregated data on emissions and temperature change.

- Identify the countries with the highest average temperature increase and the highest total agrifood CO2 emissions in 2020 and analyse their contributions to the overall environmental impact.

1.4 Produce a project plan

Table 1. Project plan

Phase	Time	Resources	Risks
Business understanding – Identify the objectives of the business	21st August	All analysts	Data problems
Business understanding – Assess the situation	22nd August	All analysts	Data problems
Business understanding – Determine data mining objectives	23rd August	All analysts	Data problems
Business understanding – Produce a project plan	24th August	All analysts	Data problems
Data understanding – Collect initial data	25th August	All analysts	Data problems, technology problems
Data understanding – Describe the data	26th August	All analysts	Data problems, technology problems
Data understanding – Explore the data	27th August	All analysts	Data problems, technology problems
Data understanding – Verify the data quality	28th August	All analysts	Data problems, technology problems
Data preparation – Select the data	29th August	Data mining consultant, database analyst	Data problems, technology problems
Data preparation – Clean the data	30th August	Data mining consultant, database analyst	Data problems, technology problems
Data preparation –	31st August	Data mining	Data problems,

Construct the data		consultant, database analyst	technology problems
Data preparation – Integrate various data sources	1st September	Data mining consultant, database analyst	Data problems, technology problems
Data preparation – Format the data as required	2nd September	Data mining consultant, database analyst	Data problems, technology problems
Data transformation – Reduce the data	3rd September	Data mining consultant, database analyst	Data problems, technology problems
Data transformation – Project the data	4th September	Data mining consultant, database analyst	Data problems, technology problems
Data-mining methods selection – Match and discuss the objectives of data-mining to data mining methods	5th September	Data mining consultant, database analyst	Technology problems, inability to find an adequate model
Data-mining methods selection – Select the appropriate data-mining method based on discussion	6th September	Data mining consultant, database analyst	Technology problems, inability to find an adequate model
Data-mining algorithms selection – Conduct exploratory analysis and discuss	7th September	Data mining consultant, database analyst	Technology problems, inability to find an adequate model
Data-mining algorithms selection – Select data-mining algorithms based on discussion	8th September	Data mining consultant, database analyst	Technology problems, inability to find an adequate model
Data-mining algorithms selection – Build/Select appropriate models and choose relevant parameters	9th September	Data mining consultant, database analyst	Technology problems, inability to find an adequate model

Data mining – Create and justify test designs	10th September	Data mining consultant, database analyst	Technology problems
Data mining – Conduct data mining: classify, regress, cluster, etc. (models must execute)	11th September	Data mining consultant, database analyst	Technology problems
Data mining – Search for patterns	12th September	Data mining consultant, database analyst	Technology problems
Interpretation – Study and discuss the mined patterns	13th September	All analysts	Inability to implement results
Interpretation – Visualize the data, results, models, and patterns	14th September	All analysts	Inability to implement results
Interpretation – Interpret the results, models, and patterns	15th September	All analysts	Inability to implement results
Interpretation – Assess and evaluate results, models, and patterns	16th September	All analysts	Inability to implement results
Interpretation – Iterate prior steps (1-7) as required	17th to 21st September	All analysts	Inability to implement results

2. Data understanding

2.1 Collect initial data

The compilation of the agricultural carbon dioxide (CO2) emission dataset involved the integration and refinement of around twelve distinct datasets sourced from the Food and Agriculture Organisation (FAO) as well as data obtained from the Intergovernmental Panel on Climate Change (IPCC). The dataset is from the website https://www.kaggle.com/datasets/alessandrolobello/agri-food-co2-emission-dataset-forecasting-ml.

2.2 Describe the data

All features show the corresponding CO2 emissions. CO2 is recorded in kilotons (kt); 1 kt represents 1,000,000 kg of CO2. The "Average Temperature C°" feature serves as the machine learning model's target variable and signifies the mean annual temperature rise. For instance, when the value is 0.12, the temperature experienced at a specific location has risen by 0.12 degrees Celsius.

Forestland is the sole characteristic that demonstrates negative, as it functions as a carbon sink. Forests play a crucial role in photosynthesis, wherein they actively absorb carbon dioxide from the atmosphere and subsequently store it, thereby effectively mitigating its presence. Sustainable forest management, in conjunction with afforestation and reforestation endeavours, enhances negative emissions by augmenting the capacity for carbon sequestration.

All the dataset features are the following:

Table 2. Dataset features

Features	Explanation
Savanna fires	Emissions from fires in savanna ecosystems
Forest fires	Emissions from fires in forested areas.
Crop residues	Emissions from burning or decomposing leftover plant material after crop harvesting.
Rice cultivation	Emissions from methane released during rice cultivation.
Drained organic soils (CO2)	Emissions from carbon dioxide released when draining organic soils.
Pesticides manufacturing	Emissions from the production of pesticides.
Food transport	Emissions from transporting food products.
Forestland	Land covered by forests.
Net forest conversion	Change in forest area due to deforestation and afforestation.
Food household consumption	Emissions from food consumption at the household level.
Food retail	Emissions from the operation of retail establishments selling food.
On-farm electricity use	Electricity consumption on farms.
Food packaging	Emissions from the production and disposal of food packaging materials.
Agrifood system waste	Emissions from waste disposal in the agrifood system.

disposal	
Food processing	Emissions from processing food products.
Fertilizers manufacturing	Emissions from the production of fertilizers.
IPPU	Emissions from industrial processes and product use.
Manure applied to soils	Emissions from applying animal manure to agricultural soils.
Manure left on pasture	Emissions from animal manure on pasture or grazing land.
Measure management	Emissions from managing and treating animal manure.
Fires in organic soils	Emissions from fires in organic soils.
Fires in humid tropical forests	Emissions from fires in humid tropical forests.
On-farm energy use	Energy consumption on farms.
Rural population	Number of people living in rural areas.
Urban population	Number of people living in urban areas.
Total population – Male	The total number of male individuals in the population.
Total population – Female	The total number of female individuals in the population.
Total emission	Total greenhouse gas emissions from various sources.
Average temperature ℃	The average increase of temperature (by year) in degrees Celsius,

2.3 Explore the data

Figure 1 shows the partial content of the emission dataset, and Figure 2 shows the partial content of the population dataset.

Figure 1. Partial content of the emission dataset

```
emission_df = pd.read_csv('..\mydataset\Agrifood_co2_emission.csv')
population_df = pd.read_csv('..\mydataset\Agrifood_co2_population.csv')
emission_df
```

	Area	Year	Savanna fires	Forest fires	Crop Residues	Rice Cultivation	Drained organic soils (CO2)	Pesticides Manufacturing	Food Transport	Forestland	...	Fertilizers Manufacturing	IPPU	Manure applied to Soils	Manure left on Pasture	Manure Management	Fire orga s
0	Afghanistan	1990	14.7237	0.0557	205.6077	686.0000	0.0	11.807483	63.1152	-2388.8030	...	11.997000	209.9778	260.1431	1590.5319	319.1763	
1	Afghanistan	1991	14.7237	0.0557	209.4971	678.1600	0.0	11.712073	61.2125	-2388.8030	...	12.853900	217.0388	268.6292	1657.2364	342.3079	
2	Afghanistan	1992	14.7237	0.0557	196.5341	686.0000	0.0	11.712073	53.3170	-2388.8030	...	13.492900	222.1156	264.7898	1653.5068	349.1224	
3	Afghanistan	1993	14.7237	0.0557	230.8175	686.0000	0.0	11.712073	54.3617	-2388.8030	...	14.055900	201.2057	261.7221	1642.9623	352.2947	
4	Afghanistan	1994	14.7237	0.0557	242.0494	705.6000	0.0	11.712073	53.9874	-2388.8030	...	15.126900	182.2905	267.6219	1689.3593	367.6784	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
6960	Zimbabwe	2016	1190.0089	232.5068	70.9451	7.4088	0.0	75.000000	251.1465	76500.2982	...	2585.080847	858.9820	96.1332	2721.1459	282.5994	
6961	Zimbabwe	2017	1431.1407	131.1324	108.6262	7.9458	0.0	67.000000	255.7975	76500.2982	...	1227.240253	889.4250	81.2314	2744.8763	255.5900	
6962	Zimbabwe	2018	1557.5830	221.6222	109.9835	8.1399	0.0	66.000000	327.0897	76500.2982	...	1127.687805	966.2650	81.0712	2790.0949	257.2735	
6963	Zimbabwe	2019	1591.6049	171.0262	45.4574	7.8322	0.0	73.000000	290.1893	76500.2982	...	2485.528399	945.9420	85.7211	2828.7215	267.5224	
6964	Zimbabwe	2020	481.9027	48.4197	108.3022	7.9733	0.0	73.000000	238.7639	76500.2982	...	1227.240253	940.4200	85.3143	2829.7457	266.7316	

6965 rows × 27 columns

Figure 2. Partial content of the population dataset

population_df

	Area	Year	Rural population	Urban population	Total Population - Male	Total Population - Female	total_emission	Average Temperature
0	Afghanistan	1990	9655167	2593947	5348387.0	5346409.0	2198.963539	0.536167
1	Afghanistan	1991	10230490	2763167	5372959.0	5372208.0	2323.876629	0.020667
2	Afghanistan	1992	10995568	2985663	6028494.0	6028939.0	2356.304229	-0.259583
3	Afghanistan	1993	11858090	3237009	7003641.0	7000119.0	2368.470529	0.101917
4	Afghanistan	1994	12690115	3482604	7733458.0	7722096.0	2500.768729	0.372250
...	...	...	...	...	...	...	...	...
6960	Zimbabwe	2016	10934468	5215894	6796658.0	7656047.0	98491.026350	1.120250
6961	Zimbabwe	2017	11201138	5328766	6940631.0	7810471.0	97159.311550	0.046500
6962	Zimbabwe	2018	11465748	5447513	7086002.0	7966181.0	97668.308200	0.516333
6963	Zimbabwe	2019	11725970	5571525	7231989.0	8122618.0	98988.062800	0.985667
6964	Zimbabwe	2020	11980005	5700460	7385220.0	8284447.0	96505.221850	0.189000

6965 rows × 8 columns

Figure 3 shows the attributes information of the emission dataset, and Figure 4 shows the attributes information of the population dataset.

Figure 3. Attributes information of emission dataset

```
emission_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6965 entries, 0 to 6964
Data columns (total 27 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Area                            6965 non-null   object
 1   Year                            6965 non-null   int64
 2   Savanna fires                   6934 non-null   float64
 3   Forest fires                    6872 non-null   float64
 4   Crop Residues                   5576 non-null   float64
 5   Rice Cultivation                6965 non-null   float64
 6   Drained organic soils (CO2)     6965 non-null   float64
 7   Pesticides Manufacturing        6965 non-null   float64
 8   Food Transport                  6965 non-null   float64
 9   Forestland                      6472 non-null   float64
 10  Net Forest conversion           6472 non-null   float64
 11  Food Household Consumption      6492 non-null   float64
 12  Food Retail                     6965 non-null   float64
 13  On-farm Electricity Use         6965 non-null   float64
 14  Food Packaging                  6965 non-null   float64
 15  Agrifood Systems Waste Disposal 6965 non-null   float64
 16  Food Processing                 6965 non-null   float64
 17  Fertilizers Manufacturing       6965 non-null   float64
 18  IPPU                            6222 non-null   float64
 19  Manure applied to Soils         6037 non-null   float64
 20  Manure left on Pasture          6965 non-null   float64
 21  Manure Management               6037 non-null   float64
 22  Fires in organic soils          6965 non-null   float64
 23  Fires in humid tropical forests 6810 non-null   float64
 24  On-farm energy use              6009 non-null   float64
 25  total_emission                  6965 non-null   float64
 26  Average Temperature             6965 non-null   float64
dtypes: float64(25), int64(1), object(1)
memory usage: 1.4+ MB
```

Figure 4. Attributes information of population dataset

```
population_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6965 entries, 0 to 6964
Data columns (total 8 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Area                       6965 non-null   object
 1   Year                       6965 non-null   int64
 2   Rural population           6965 non-null   int64
 3   Urban population           6965 non-null   int64
 4   Total Population - Male    6965 non-null   float64
 5   Total Population - Female  6965 non-null   float64
 6   total_emission             6965 non-null   float64
 7   Average Temperature        6965 non-null   float64
dtypes: float64(4), int64(3), object(1)
memory usage: 435.4+ KB
```

Figure 5 shows the statistic description of the emission dataset, and Figure 6 shows the statistic description of the population dataset.

Figure 5. Statistic description of emission dataset

```
emission_desc = emission_df.describe()
population_desc = population_df.describe()
round(emission_desc.transpose(), 2)
```

	count	mean	std	min	25%	50%	75%	max
Year	6965.0	2005.12	8.89	1990.00	1997.00	2005.00	2013.00	2020.00
Savanna fires	6934.0	1188.39	5246.29	0.00	0.00	1.65	111.08	114616.40
Forest fires	6872.0	919.30	3720.08	0.00	0.00	0.52	64.95	52227.63
Crop Residues	5576.0	998.71	3700.35	0.00	11.01	103.70	377.64	33490.07
Rice Cultivation	6965.0	4259.67	17613.83	0.00	181.26	534.82	1536.64	164915.26
Drained organic soils (CO2)	6965.0	3503.23	15861.45	0.00	0.00	0.00	690.41	241025.07
Pesticides Manufacturing	6965.0	333.42	1429.16	0.00	6.00	13.00	116.33	16459.00
Food Transport	6965.0	1939.58	5616.75	0.00	27.96	204.96	1207.00	67945.76
Forestland	6472.0	-17828.29	81832.21	-797183.08	-2848.35	-62.92	0.00	171121.08
Net Forest conversion	6472.0	17605.64	101157.53	0.00	0.00	44.44	4701.75	1605106.10
Food Household Consumption	6492.0	4847.58	25789.14	0.00	11.40	155.47	1377.15	466288.20
Food Retail	6965.0	2043.21	8494.25	0.00	26.82	172.04	1076.00	133784.07
On-farm Electricity Use	6965.0	1626.68	9343.18	0.00	8.04	29.12	499.94	165676.30
Food Packaging	6965.0	1658.63	11481.34	0.00	67.63	74.02	281.79	175741.31
Agrifood Systems Waste Disposal	6965.0	6018.44	22156.74	0.34	86.68	901.28	3006.44	213289.70
Food Processing	6965.0	3872.72	19838.22	0.00	209.59	344.76	1236.91	274253.51
Fertilizers Manufacturing	6965.0	3035.72	11693.03	0.00	360.36	1115.05	2024.87	170826.42
IPPU	6222.0	19991.50	111420.85	0.00	39.03	803.71	6155.17	1861640.66
Manure applied to Soils	6037.0	923.23	3226.99	0.05	16.30	120.44	460.12	34677.36
Manure left on Pasture	6965.0	3518.03	9103.56	0.00	139.67	972.57	2430.79	92630.76
Manure Management	6037.0	2263.34	7980.54	0.43	37.63	269.86	1126.82	70592.65
Fires in organic soils	6965.0	1210.32	22669.85	0.00	0.00	0.00	0.00	991717.54

Figure 6. Statistic description of population dataset

```
round(population_desc.transpose(), 2)
```

	count	mean	std	min	25%	50%	75%	max
Year	6965.0	2005.12	8.89	1990.00	1997.00	2005.00	2013.00	2.020000e+03
Rural population	6965.0	17857735.39	89015213.76	0.00	97311.00	1595322.00	8177340.00	9.000991e+08
Urban population	6965.0	16932296.97	65743619.61	0.00	217386.00	2357581.00	8277123.00	9.020778e+08
Total Population - Male	6965.0	17619629.63	76039931.01	250.00	201326.00	2469660.00	9075924.00	7.435866e+08
Total Population - Female	6965.0	17324469.29	72517113.54	270.00	207890.00	2444135.00	9112588.00	7.133419e+08
total_emission	6965.0	64091.24	228312.96	-391884.06	5221.24	12147.65	35139.73	3.115114e+06
Average Temperature	6965.0	0.87	0.56	-1.42	0.51	0.83	1.21	3.560000e+00

Figure 7 shows the simple correlation between 'Year', 'total_emission', and 'Average Temperature'.

Figure 7. Simple correlation

```
emission_df[['Year', 'total_emission', 'Average Temperature']].corr()
```

	Year	total_emission	Average Temperature
Year	1.000000	0.041861	0.545932
total_emission	0.041861	1.000000	0.019043
Average Temperature	0.545932	0.019043	1.000000

Figure 8 shows the statistic description of 'total_emission' and 'Average Temperature' based on different areas.

Figure 8. Different areas statistic description

```
round(emission_df[['Area','total_emission', 'Average Temperature']].groupby('Area').describe().transpose(), 2)
```

	Area	Afghanistan	Albania	Algeria	American Samoa	Andorra	Angola	Anguilla	Antigua and Barbuda	Argentina	Armenia	...	Uzbekistan	Vanuatu	Venezuela (Bolivarian Republic of)	Viet Nam	Wallis and Futuna Islands
total_emission	**count**	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	29.00	...	29.00	31.00	31.00	31.00	31.00
	mean	7354.05	3696.33	40441.28	5498.12	5287.55	90258.38	12338.00	2604.91	154059.04	3360.25	...	38070.26	3438.68	146919.19	71865.07	11063.49
	std	4165.36	1061.32	15282.04	190.51	211.98	15442.69	773.26	34.44	29728.77	840.29	...	6621.52	1193.84	33054.34	64620.87	586.62
	min	2198.96	2074.20	22326.60	5278.03	5127.93	65955.64	12016.69	2554.09	112979.76	2038.44	...	28852.52	2977.60	87274.84	-18137.55	10270.83
	25%	3158.85	2832.52	27693.71	5290.25	5142.80	70949.93	12024.13	2572.38	123634.77	2505.96	...	31735.61	2995.09	129540.23	-4382.06	10271.71
	50%	7001.30	3516.55	35066.07	5547.79	5148.03	96788.24	12030.93	2612.91	150735.55	3709.43	...	38857.38	3044.92	156667.90	96242.32	11042.94
	75%	11487.14	4469.05	51286.86	5706.89	5578.16	102782.14	12178.30	2628.35	179482.94	4031.99	...	45327.92	3058.23	178560.23	112009.60	11721.98
	max	14032.42	5680.14	69603.09	5718.85	5606.13	111654.00	15072.23	2666.27	197788.77	4483.33	...	48990.62	7761.22	186915.75	171383.63	11740.68
Average Temperature	**count**	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	31.00	29.00	...	29.00	31.00	31.00	31.00	31.00
	mean	0.82	0.89	1.16	0.85	1.28	0.76	0.69	0.69	0.48	1.05	...	1.14	0.55	0.74	0.70	0.58
	std	0.54	0.60	0.50	0.43	0.59	0.42	0.25	0.24	0.35	0.87	...	0.68	0.36	0.28	0.44	0.41
	min	-0.26	-0.46	-0.21	-0.03	0.23	0.11	0.25	0.29	-0.21	-1.30	...	-0.56	-0.34	0.29	0.01	-0.29
	25%	0.39	0.47	1.00	0.58	0.91	0.40	0.52	0.51	0.23	0.74	...	0.68	0.36	0.53	0.43	0.28
	50%	0.89	1.04	1.20	0.85	1.25	0.68	0.67	0.66	0.52	1.00	...	1.32	0.57	0.73	0.64	0.69
	75%	1.27	1.26	1.39	1.22	1.81	0.98	0.86	0.87	0.74	1.44	...	1.55	0.75	0.92	0.96	0.83
	max	1.84	1.99	2.23	1.57	2.39	1.77	1.14	1.14	1.12	2.86	...	2.20	1.21	1.31	1.77	1.41

16 rows × 236 columns

Figure 9 shows the global average temperature change and the global total emissions change by year. Figure 10 shows the distribution of the global average temperature change by year.

Figure 9. Global average temperature and total emissions change by year

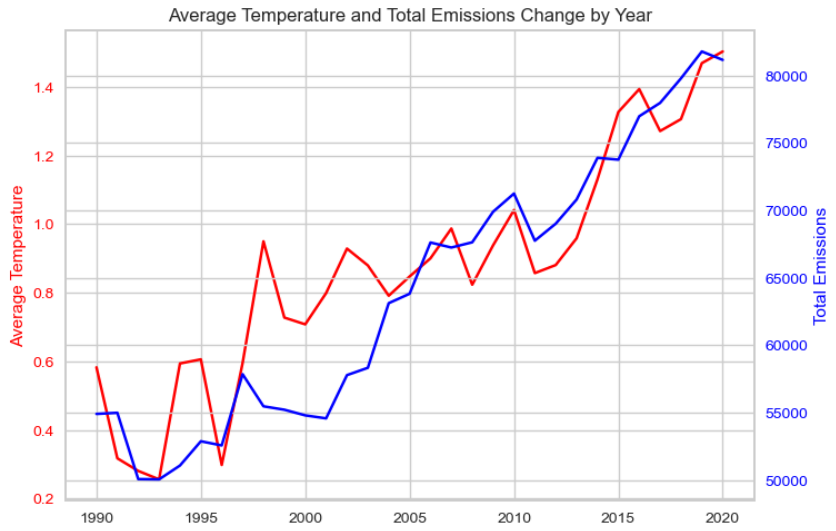


Figure 10. Global average temperature change distribution by year

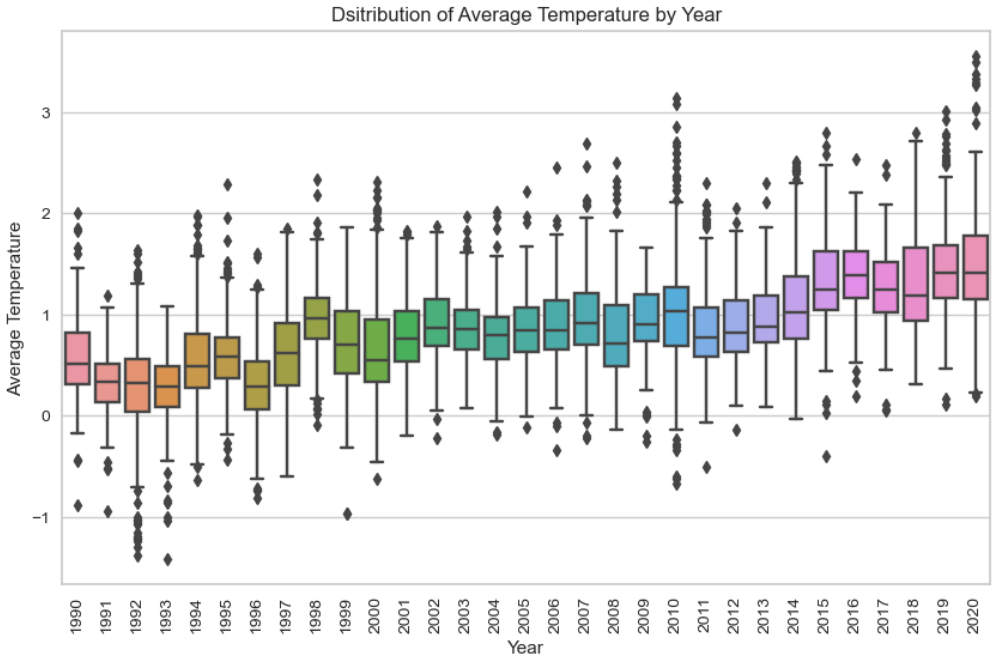


Figure 11 shows the average temperature change from 1990 to 2020 by area. Figure 12 shows the average agrifood CO2 emissions from 1990 to 2020 by top 50 areas.

Figure 11. Average temperature by area

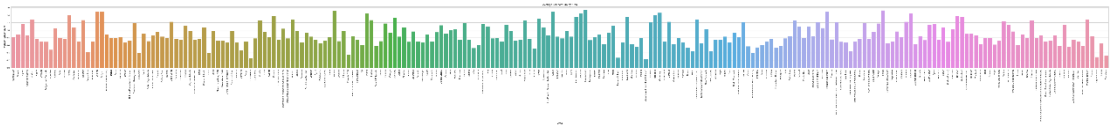
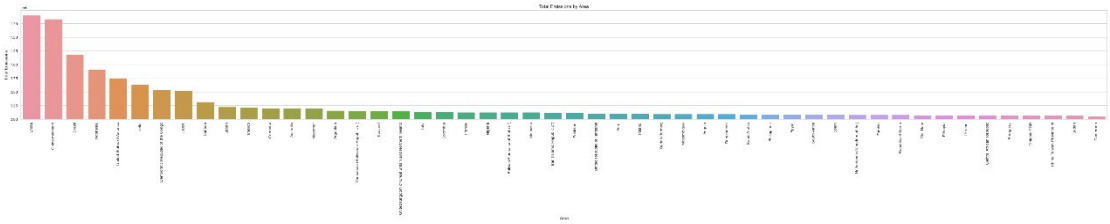


Figure 12. Average agrifood CO2 emissions by top 50 areas



2.4 Verify the data quality

Data need to be cleaned and prepared for machine learning models. Missing values, outliers, and feature engineering should be handled with advanced regression techniques. Data quality assessment is frequently conducted throughout description and exploration stages. Figure 13 shows each feature's missing values.

Figure 13. Missing values

```
emission_df.isna().sum()
```

```
Area                                0
Year                                0
Savanna fires                      31
Forest fires                       93
Crop Residues                    1389
Rice Cultivation                    0
Drained organic soils (CO2)         0
Pesticides Manufacturing            0
Food Transport                      0
Forestland                        493
Net Forest conversion             493
Food Household Consumption        473
Food Retail                         0
On-farm Electricity Use             0
Food Packaging                      0
Agrifood Systems Waste Disposal     0
Food Processing                     0
Fertilizers Manufacturing           0
IPPU                              743
Manure applied to Soils           928
Manure left on Pasture              0
Manure Management                 928
Fires in organic soils              0
Fires in humid tropical forests   155
On-farm energy use                956
total_emission                      0
Average Temperature                 0
dtype: int64
```

For detecting outliers in a dataset, there are various methods to detect outliers. A simple method called the interquartile range (IQR) method is used in this project. In this method, values outside a certain range are considered outliers. The following steps are the implementation of this method. Figure 14 and Figure 15 shows the process and the result of this method.

- A function detect_outliers is defined, which takes a column as input.
- It calculates the first quartile (Q1), third quartile (Q3), and interquartile range (IQR).
- It then defines lower and upper bounds beyond which values are considered outliers.
- The function returns a boolean Series indicating whether each value is an outlier or not.

Figure 14. Detecting outliers

```
# Define a function to detect outliers using IQR
def detect_outliers(column):
    Q1 = column.quantile(0.25)
    Q3 = column.quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return (column < lower_bound) | (column > upper_bound)

# Apply the function to each column
outliers1 = emission_df.select_dtypes(exclude='object').apply(detect_outliers)
# Now, 'outliers' is a DataFrame with True/False indicating outliers
outliers1
```

Figure 15. Results of detecting outliers

	Year	Savanna fires	Forest fires	Crop Residues	Rice Cultivation	Drained organic soils (CO2)	Pesticides Manufacturing	Food Transport	Forestland	Net Forest conversion	...	Fertilizers Manufacturing	IPPU	Manure applied to Soils	Manure left on Pasture	Manure Management	Fires in organic soils	Fires i humi tropica forest
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fals
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
6960	False	True	True	False	False	False	False	False	True	False	...	False	False	False	False	False	False	Fals
6961	False	True	False	False	False	False	False	False	True	False	...	False	False	False	False	False	False	Fals
6962	False	True	True	False	False	False	False	False	True	False	...	False	False	False	False	False	False	Fals
6963	False	True	True	False	False	False	False	False	True	False	...	False	False	False	False	False	False	Fals
6964	False	True	False	False	False	False	False	False	True	False	...	False	False	False	False	False	False	Fals

6965 rows × 26 columns

3. Data preparation

3.1 Select the data

After a profound understanding, according to the data collected during the initial phase of the CRISP-DM methodology, the data relevant to the data mining goals is selected. This part should contain selecting items and selecting attributes. In this project, the crucial data mining

objectives are to analyse the influence of various countries based on aggregated data on emissions and temperature change, identify the countries with the highest average temperature increase by year, and analyse their contributions to the overall environmental impact. Thus, all countries are considered, which means all items should be considered, so all items are selected. For selecting attributes, one of the data mining goals is to examine the correlation between carbon dioxide (CO2) emissions within the agri-food sector and the subsequent temperature rise, the attribute ‘total_emission’ is the summation of all types of carbon dioxide emissions from the agri-food system. Therefore, all features relevant to carbon dioxide emissions from the agrifood system are selected and considered. Only ‘Rural population’, ‘Urban population’, ‘Total Population – Male’, and ‘Total population – Female’ are excluded.

The next section will clean all data qualities including outliers and missing values. As Figure 13 shows, for the missing values, the attributes ‘Savanna fires’, ‘Forest fires’, ‘Crop Residues’, ‘Forestland’, ‘Net Forest conversion’, ‘Food Household Consumption’, ‘IPPU’, ‘Manure applied to Soils’, ‘Manure Management’, ‘Fires in humid tropical forests’, and ‘On-farm energy use’ are cleaned.

3.2 Clean the data

3.2.1 Missing values cleaning

The method, Multiple Imputation by Chained Equations (*Imputing Missing Values with Variants of IterativeImputer*, n.d.), is used to impute the missing values in this project (Figure 16). Multiple Imputation by Chained Equations is a dependable and informative technique for addressing absent data in datasets. The procedure 'fills in' (imputes) absent data in a dataset through an iterative series of predictive models. Each iteration imputes each specified variable in the dataset using the other variables. These iterations should continue until convergence has been achieved (*Introduction*, n.d.). This is a multivariate imputer that estimates each feature from all the others. This is a strategy for imputing missing values by modeling each feature with missing values as a function of other features in a round-robin fashion.

Figure 16. Missing values impute method

```
import warnings
warnings.filterwarnings("ignore")

# Multiple Imputation by Chained Equations
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
```

```
cols_to_impute = ['Savanna fires', 'Forest fires', 'Crop Residues',
                  'Rice Cultivation', 'Drained organic soils (CO2)',
                  'Pesticides Manufacturing', 'Food Transport', 'Forestland',
                  'Net Forest conversion', 'Food Household Consumption',
                  'Food Retail', 'On-farm Electricity Use', 'Food Packaging',
                  'Agrifood Systems Waste Disposal', 'Food Processing',
                  'Fertilizers Manufacturing', 'IPPU',
                  'Manure applied to Soils', 'Manure left on Pasture',
                  'Manure Management', 'Fires in organic soils',
                  'Fires in humid tropical forests', 'On-farm energy use']

mice_imputer = IterativeImputer()
emission_df[cols_to_impute] = mice_imputer.fit_transform(emission_df[cols_to_impute])
emission_df.isna().sum()
```

Figure 17 shows the results after cleaning the missing values

Figure 17. Missing values cleaned

```
Area                                0
Year                                0
Savanna fires                       0
Forest fires                        0
Crop Residues                       0
Rice Cultivation                    0
Drained organic soils (CO2)         0
Pesticides Manufacturing            0
Food Transport                      0
Forestland                          0
Net Forest conversion               0
Food Household Consumption          0
Food Retail                         0
On-farm Electricity Use             0
Food Packaging                      0
Agrifood Systems Waste Disposal     0
Food Processing                     0
Fertilizers Manufacturing           0
IPPU                                0
Manure applied to Soils             0
Manure left on Pasture              0
Manure Management                   0
Fires in organic soils              0
Fires in humid tropical forests     0
On-farm energy use                  0
total_emission                      0
Average Temperature                 0
dtype: int64
```

3.2.2 Outliers processing

For outliers, a common method of dealing with this is to coerce outliers. Thus, this project uses this method to process the outliers, before making further decisions.

3.3 Construct the data

All missing values and outliers are processed; in the previous table, the feature 'total_emission' is not the summation of all types of carbon dioxide emissions from the agri-food system. A new attribute called 'Updated_total_emission' is constructed, which calculates the summation of the cleaned data of all types of carbon dioxide emissions from the agrifood system. The values of 'Updated_total_emission' is different from the values of 'total_emission' (Figure 18).

Figure 18. New attribute – 'Updated_total_emission'

```
emission_df['Updated_total_emission'] = emission_df.loc[:, 'Savanna fires':'On-farm energy use'].sum(axis=1)
emission_df
```

Rice ation	Drained organic soils (CO2)	Pesticides Manufacturing	Food Transport	Forestland	...	IPPU	Manure applied to Soils	Manure left on Pasture	Manure Management	Fires in organic soils	Fires in humid tropical forests	On-farm energy use	total_emission	Average Temperature	Updated_total_emission
.0000	0.0	11.807483	63.1152	-2388.8030	...	209.9778	260.1431	1590.5319	319.1763	0.0	0.0	313.133908	2198.963539	0.536167	2512.097448
.1600	0.0	11.712073	61.2125	-2388.8030	...	217.0388	268.6292	1657.2364	342.3079	0.0	0.0	332.517903	2323.876629	0.020667	2656.394532
.0000	0.0	11.712073	53.3170	-2388.8030	...	222.1156	264.7898	1653.5068	349.1224	0.0	0.0	292.215672	2356.304229	-0.259583	2648.519901
.0000	0.0	11.712073	54.3617	-2388.8030	...	201.2057	261.7221	1642.9623	352.2947	0.0	0.0	288.385814	2368.470529	0.101917	2656.856343
.6000	0.0	11.712073	53.9874	-2388.8030	...	182.2905	267.6219	1689.3593	367.6784	0.0	0.0	309.629350	2500.768729	0.372250	2810.398080
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
.4088	0.0	75.000000	251.1465	76500.2982	...	858.9820	96.1332	2721.1459	282.5994	0.0	0.0	417.315000	98491.026350	1.120250	98491.026347
.9458	0.0	67.000000	255.7975	76500.2982	...	889.4250	81.2314	2744.8763	255.5900	0.0	0.0	398.164400	97159.311550	0.046500	97159.311553
.1399	0.0	66.000000	327.0897	76500.2982	...	966.2650	81.0712	2790.0949	257.2735	0.0	0.0	465.773500	97668.308200	0.516333	97668.308205
.8322	0.0	73.000000	290.1893	76500.2982	...	945.9420	85.7211	2828.7215	267.5224	0.0	0.0	444.233500	98988.062800	0.985667	98988.062799
.9733	0.0	73.000000	238.7639	76500.2982	...	940.4200	85.3143	2829.7457	266.7316	0.0	0.0	444.233500	96505.221850	0.189000	96505.221853

3.4 Integrate various data resources

A new dataset, 'population dataset', is merged into the previous dataset. The new dataset has the following features: 'Area', 'Year', 'Rural population', 'Urban population', 'Total Population – Male', 'Total Population – Female', 'total_emission', and 'Average Temperature'. The features 'Area', 'Year', 'total_emission', and 'Average Temperature' are duplicated, the same as the corresponding features of the previous dataset. Figure 19 shows the codes for merging two tables, and Figure 20 shows the merged table.

Figure 19. Codes for merging

```
agrifood_emission_df = emission_df.merge(population_df, on=['Area', 'Year', 'total_emission', 'Average Temperature'])
agrifood_emission_df
```

Figure 20. Merged table

	Area	Year	Savanna fires	Forest fires	Crop Residues	Rice Cultivation	Drained organic soils (CO2)	Pesticides Manufacturing	Food Transport	Forestland	...	Fires in organic soils	Fires in humid tropical forests	On-farm energy use	total_emission	Average Temperature	Updat
0	Afghanistan	1990	14.7237	0.0557	205.6077	686.0000	0.0	11.807483	63.1152	-2388.8030	...	0.0	0.0	313.133908	2198.963539	0.536167	
1	Afghanistan	1991	14.7237	0.0557	209.4971	678.1600	0.0	11.712073	61.2125	-2388.8030	...	0.0	0.0	332.517903	2323.876629	0.020667	
2	Afghanistan	1992	14.7237	0.0557	196.5341	686.0000	0.0	11.712073	53.3170	-2388.8030	...	0.0	0.0	292.215672	2356.304229	-0.259583	
3	Afghanistan	1993	14.7237	0.0557	230.8175	686.0000	0.0	11.712073	54.3617	-2388.8030	...	0.0	0.0	288.385814	2368.470529	0.101917	
4	Afghanistan	1994	14.7237	0.0557	242.0494	705.6000	0.0	11.712073	53.9874	-2388.8030	...	0.0	0.0	309.629350	2500.768729	0.372250	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
6960	Zimbabwe	2016	1190.0089	232.5068	70.9451	7.4088	0.0	75.000000	251.1465	76500.2982	...	0.0	0.0	417.315000	98491.026350	1.120250	
6961	Zimbabwe	2017	1431.1407	131.1324	108.6262	7.9458	0.0	67.000000	255.7975	76500.2982	...	0.0	0.0	398.164400	97159.311550	0.046500	
6962	Zimbabwe	2018	1557.5830	221.6222	109.9835	8.1399	0.0	66.000000	327.0897	76500.2982	...	0.0	0.0	465.773500	97668.308200	0.516333	
6963	Zimbabwe	2019	1591.6049	171.0262	45.4574	7.8322	0.0	73.000000	290.1893	76500.2982	...	0.0	0.0	444.233500	98988.062800	0.985667	
6964	Zimbabwe	2020	481.9027	48.4197	108.3022	7.9733	0.0	73.000000	238.7639	76500.2982	...	0.0	0.0	444.233500	96505.221850	0.189000	

6965 rows × 32 columns

3.5 Format the data as required

Considering the third data mining objective, two new features are constructed to identify the countries with the highest average temperature increase and the highest total agrifood CO2 emissions in 2020 and analyse their contributions to the overall environmental impact (Figure 21). The first is 'Total_population', which is the summation of 'Rural population' and 'Urban population'. The second is 'Emissions_per_capita', whose value is 'Updated_total_emission' divided by 'Total_population.

Figure 21. New features – 'Total_population' and 'Updated_total_population

```
agrifood_emission_df['Total_population'] = agrifood_emission_df['Total Population - Female'] + agrifood_emission_df['Total Population - Male']
agrifood_emission_df['Emission_per_capita'] = agrifood_emission_df['Updated_total_emission'] / agrifood_emission_df['Total_population']
agrifood_emission_df
```

sticides cturing	Food Transport	Forestland	...	On-farm energy use	total_emission	Average Temperature	Updated_total_emission	Rural population	Urban population	Total Population - Male	Total Population - Female	Total_population	Emission_per_capita
807483	63.1152	-2388.8030	...	313.133908	2198.963539	0.536167	2512.097448	9655167	2593947	5348387.0	5346409.0	10694796.0	0.000235
712073	61.2125	-2388.8030	...	332.517903	2323.876629	0.020667	2656.394532	10230490	2763167	5372959.0	5372208.0	10745167.0	0.000247
712073	53.3170	-2388.8030	...	292.215672	2356.304229	-0.259583	2648.519901	10995568	2985663	6028494.0	6028939.0	12057433.0	0.000220
712073	54.3617	-2388.8030	...	288.385814	2368.470529	0.101917	2656.856343	11858090	3237009	7003641.0	7000119.0	14003760.0	0.000190
712073	53.9874	-2388.8030	...	309.629350	2500.768729	0.372250	2810.398080	12690115	3482604	7733458.0	7722096.0	15455554.0	0.000182
...	...	...	...	...	...	...	...	...	...	...	...	...	...
000000	251.1465	76500.2982	...	417.315000	98491.026350	1.120250	98491.026347	10934468	5215894	6796658.0	7656047.0	14452705.0	0.006815
000000	255.7975	76500.2982	...	398.164400	97159.311550	0.046500	97159.311553	11201138	5328766	6940631.0	7810471.0	14751102.0	0.006587
000000	327.0897	76500.2982	...	465.773500	97668.308200	0.516333	97668.308205	11465748	5447513	7086002.0	7966181.0	15052183.0	0.006489
000000	290.1893	76500.2982	...	444.233500	98988.062800	0.985667	98988.062799	11725970	5571525	7231989.0	8122618.0	15354607.0	0.006447
000000	238.7639	76500.2982	...	444.233500	96505.221850	0.189000	96505.221853	11980005	5700460	7385220.0	8284447.0	15669667.0	0.006159

Other than that, the missing values and the outliers of the merged table are checked (Figure 22 & Figure 23).

Figure 22. Checking missing values

```
agrifood_emission_df.isna().sum()
```

```
Area                                0
Year                                0
Savanna fires                       0
Forest fires                        0
Crop Residues                       0
Rice Cultivation                    0
Drained organic soils (CO2)         0
Pesticides Manufacturing            0
Food Transport                      0
Forestland                          0
Net Forest conversion               0
Food Household Consumption          0
Food Retail                         0
On-farm Electricity Use             0
Food Packaging                      0
Agrifood Systems Waste Disposal     0
Food Processing                     0
Fertilizers Manufacturing           0
IPPU                                0
Manure applied to Soils             0
Manure left on Pasture              0
Manure Management                   0
Fires in organic soils              0
Fires in humid tropical forests     0
On-farm energy use                  0
total_emission                      0
Average Temperature                 0
Updated_total_emission              0
Rural population                    0
Urban population                    0
Total Population - Male             0
Total Population - Female           0
Total_population                    0
Emission_per_capita                 0
dtype: int64
```

Figure 23. Checking outliers

```
outliers2 = agrifood_emission_df.select_dtypes(exclude='object').apply(detect_outliers)
outliers2
```

	Year	Savanna fires	Forest fires	Crop Residues	Rice Cultivation	Drained organic soils (CO2)	Pesticides Manufacturing	Food Transport	Forestland	Net Forest conversion	...	On-farm energy use	total_emission	Average Temperature	Updated_total_emission	Ru populati
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
6960	False	True	True	False	False	False	False	False	True	False	...	False	True	False	False	Fal
6961	False	True	False	False	False	False	False	False	True	False	...	False	True	False	False	Fal
6962	False	True	True	False	False	False	False	False	True	False	...	False	True	False	False	Fal
6963	False	True	False	False	False	False	False	False	True	False	...	False	True	False	False	Fal
6964	False	True	False	False	False	False	False	False	True	False	...	False	True	False	False	Fal

6965 rows × 33 columns

4. Data transformation

4.1 Reduce the data

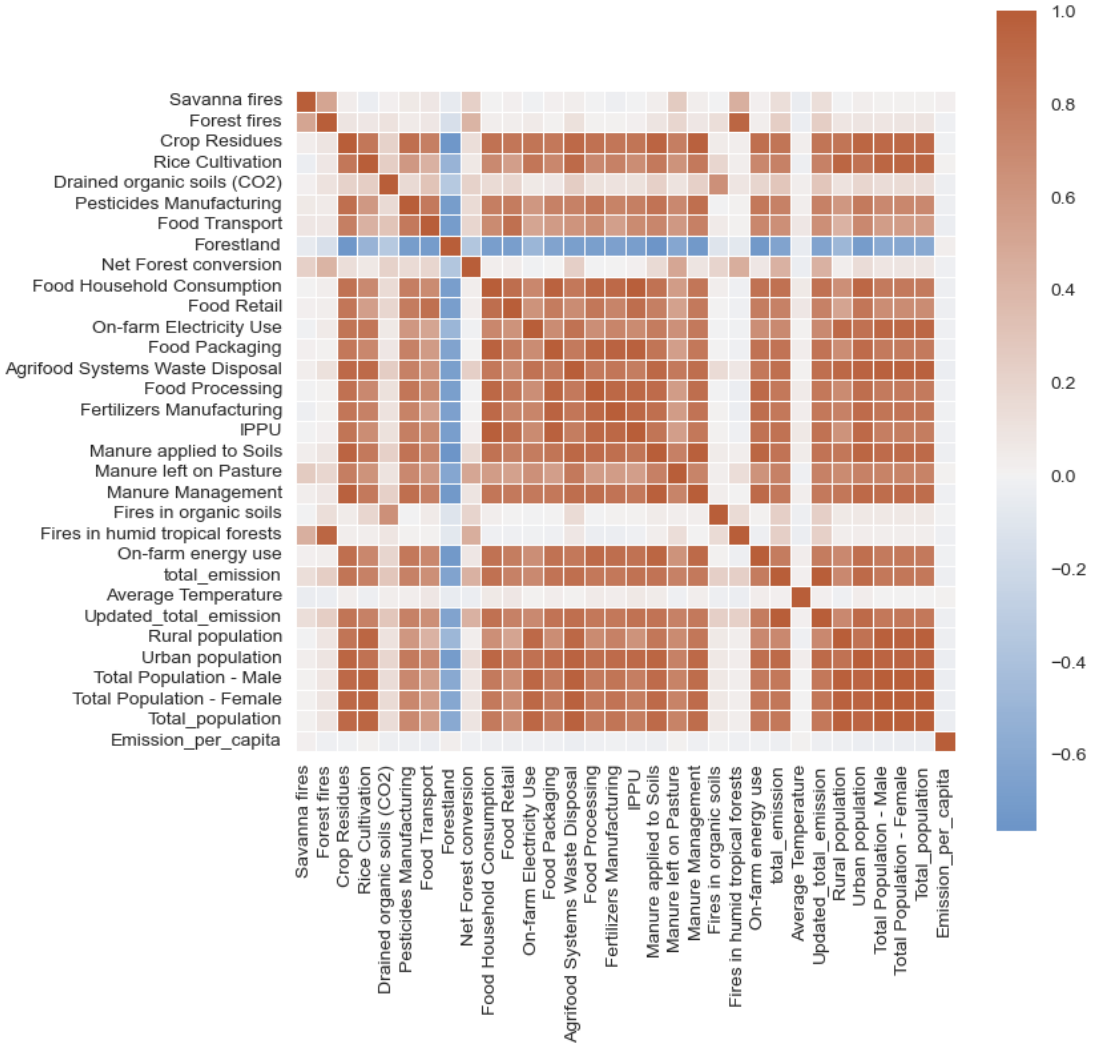
Figure 24 and Figure 25 show the correlation importance of all attributes before data transformation.

Figure 24. Codes for correlation importance of all attributes

```
cols = ['Savanna fires', 'Forest fires', 'Crop Residues',
       'Rice Cultivation', 'Drained organic soils (CO2)',
       'Pesticides Manufacturing', 'Food Transport', 'Forestland',
       'Net Forest conversion', 'Food Household Consumption', 'Food Retail',
       'On-farm Electricity Use', 'Food Packaging',
       'Agrifood Systems Waste Disposal', 'Food Processing',
       'Fertilizers Manufacturing', 'IPPU', 'Manure applied to Soils',
       'Manure left on Pasture', 'Manure Management', 'Fires in organic soils',
       'Fires in humid tropical forests', 'On-farm energy use',
       'total_emission', 'Average Temperature', 'Updated_total_emission',
       'Rural population', 'Urban population', 'Total Population - Male',
       'Total Population - Female', 'Total_population', 'Emission_per_capita']

# Correlation Heatmap
corr = agrifood_emission_df[cols].corr()
f, ax = plt.subplots(figsize=(8, 8))
cmap = sns.diverging_palette(250, 25, as_cmap=True)
sns.heatmap(corr, cmap=cmap, vmax=None, center=0,square=True, annot=False, linewidths=.5)
```

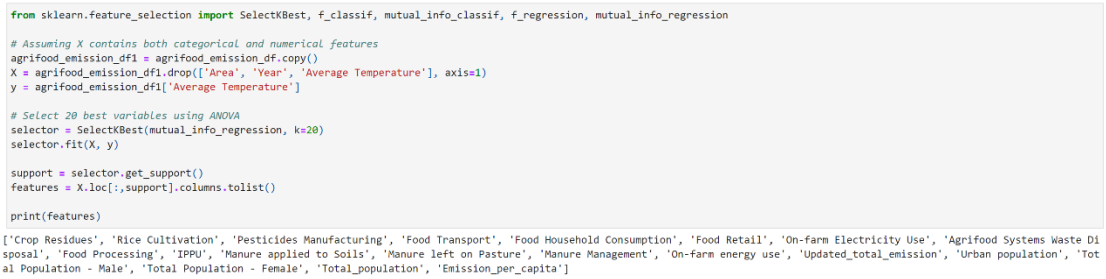
Figure 25. Correlation importance of all attributes before data transformation



For feature selection, Mutual Information filter method is utilised in this project. Mutual information measures the statistical dependence or relationship between two random variables (*Sklearn.Feature_selection.Mutual_info_regression*, n.d.). In feature selection, it quantifies the information gained about one variable (the target) by observing another variable (a feature).

When applied to feature selection, Mutual Information evaluates how much information a feature provides about the target variable. Features with high mutual information scores are considered more informative and are more likely to be selected. It is beneficial for both categorical and continuous variables. It can capture non-linear relationships between features and the target variable (*Sklearn.Feature_selection.Mutual_info_regression*, n.d.). Figure 26 shows the codes and the selection result of this method.

Figure 26. Feature selection by Mutual Information filter method



```
from sklearn.feature_selection import SelectKBest, f_classif, mutual_info_classif, f_regression, mutual_info_regression

# Assuming X contains both categorical and numerical features
agrifood_emission_df1 = agrifood_emission_df.copy()
X = agrifood_emission_df1.drop(['Area', 'Year', 'Average Temperature'], axis=1)
y = agrifood_emission_df1['Average Temperature']

# Select 20 best variables using ANOVA
selector = SelectKBest(mutual_info_regression, k=20)
selector.fit(X, y)

support = selector.get_support()
features = X.loc[:,support].columns.tolist()

print(features)
```

```
['Crop Residues', 'Rice Cultivation', 'Pesticides Manufacturing', 'Food Transport', 'Food Household Consumption', 'Food Retail', 'On-farm Electricity Use', 'Agrifood Systems Waste Disposal', 'Food Processing', 'IPPU', 'Manure applied to Soils', 'Manure left on Pasture', 'Manure Management', 'On-farm energy use', 'Updated_total_emission', 'Urban population', 'Total Population - Male', 'Total Population - Female', 'Total_population', 'Emission_per_capita']
```

‘Average Temperature’ is the target feature, and ‘Area’ and ‘Year’ are the necessary features based on three data mining objectives, so they are dropped before the feature selection. The features selected include 'Crop Residues', 'Rice Cultivation', 'Pesticides Manufacturing', 'Food Transport', 'Food Household Consumption', 'Food Retail', 'On-farm Electricity Use', 'Agrifood Systems Waste Disposal', 'Food Processing', 'IPPU', 'Manure applied to Soils', 'Manure left on Pasture', 'Manure Management', 'On-farm energy use', 'Updated_total_emission', 'Urban population', 'Total Population - Male', 'Total Population - Female', 'Total_population', and 'Emission_per_capita'.

4.2 Project the data

In this project, RobustScaler is used to project the data. The RobustScaler is a feature scaling technique commonly employed in machine learning, which leverages statistical measures that exhibit resilience to the presence of outliers (*Sklearn.Preprocessing.RobustScaler*, n.d.). This approach excludes the median value and the subsequent data adjustment based on the quantile range. The default quantile range used is the Interquartile Range (IQR), which represents the interval between the 25th and 75th quantiles, also known as the first and third quartiles (*Sklearn.Preprocessing.RobustScaler*, n.d.). The process involves independently centering and scaling each feature by calculating the corresponding statistics based on the samples in the training set. The transform method

subsequently retains the median and interquartile range for future data manipulation. Standardizing a dataset is a prevalent necessity for numerous machine learning estimators. The standard procedure involves the removal of the mean and subsequent scaling to achieve unit variance. Nevertheless, it is essential to note that outliers have the potential to exert a detrimental impact on the sample mean and variance. In the context of this project, it is frequently observed that the utilization of the median and the interquartile range yields more favourable outcomes.

Figure 27 shows the codes for RobustScaler implementation, only selected features are scaled by RobustScaler. Figure 28 shows the result after data transformation.

Figure 27. Codes for RobustScaler implementation

```
from sklearn.preprocessing import RobustScaler

scaler = RobustScaler()
agrifood_emission_df2 = agrifood_emission_df.copy()
scaled_columns = agrifood_emission_df2[['Crop Residues', 'Rice Cultivation', 'Pesticides Manufacturing', 'Food Transport', 'Net Forest conversion', 'Food Household Consumption', 'Fo
# numeric_columns = agrifood_emission_df2.select_dtypes(include=[np.number])
scaler.fit(scaled_columns)
scaled_data = scaler.transform(scaled_columns)
scaled_df = pd.DataFrame(scaled_data, columns=scaled_columns.columns)

scaled_df
```

Figure 28. Result after data transformation

	Area	Year	Average Temperature	Crop Residues	Rice Cultivation	Pesticides Manufacturing	Food Transport	Net Forest conversion	Food Household Consumption	Food Retail	...	Agrifood Systems Waste Disposal	IPPU	Manure applied to Soils	Manure left on Pasture	Manure Management
0	Afghanistan	1990	0.536167	0.318966	0.111543	-0.010809	-0.120307	-0.005524	-0.088036	-0.059473	...	-0.071748	-0.108416	0.276634	0.269721	-0.058480
1	Afghanistan	1991	0.020667	0.331274	0.105758	-0.011674	-0.121921	-0.005524	-0.087042	-0.052769	...	-0.065229	-0.107237	0.299558	0.298836	-0.035514
2	Afghanistan	1992	-0.259583	0.290251	0.111543	-0.011674	-0.128618	-0.005524	-0.086844	-0.043720	...	-0.053977	-0.106389	0.289186	0.297208	-0.028748
3	Afghanistan	1993	0.101917	0.398746	0.111543	-0.011674	-0.127732	-0.005524	-0.083802	-0.086336	...	-0.037452	-0.109881	0.280899	0.292605	-0.025598
4	Afghanistan	1994	0.372250	0.434291	0.126004	-0.011674	-0.128049	-0.005524	-0.081156	-0.077815	...	-0.023755	-0.113040	0.296837	0.312856	-0.010325
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6960	Zimbabwe	2016	1.120250	-0.107195	-0.389123	0.561973	0.039171	1.948657	0.033819	0.258339	...	0.060266	-0.000022	-0.166417	0.763197	-0.094795
6961	Zimbabwe	2017	0.046500	0.012052	-0.388726	0.489461	0.043115	1.948657	-0.000253	0.260532	...	0.065782	0.005063	-0.206672	0.773555	-0.121611
6962	Zimbabwe	2018	0.516333	0.016348	-0.388583	0.480397	0.103581	1.948657	0.005430	0.305779	...	0.070991	0.017896	-0.207105	0.793291	-0.119940
6963	Zimbabwe	2019	0.985667	-0.187855	-0.388810	0.543845	0.072285	1.948657	0.017804	0.353180	...	0.075366	0.014502	-0.194544	0.810150	-0.109764
6964	Zimbabwe	2020	0.189000	0.011027	-0.388706	0.543845	0.028668	1.948657	0.007389	0.377439	...	0.079763	0.013580	-0.195642	0.810597	-0.110549

6965 rows × 21 columns

5. Data mining methods selection

5.1 Match and discuss the objectives of data mining to data mining methods

Three different data mining methods are discussed and matched to three data mining objectives, respectively.

The first data mining objective, to examine the correlation between carbon dioxide (CO2) emissions within the agri-food sector and the subsequent temperature rise, involves studying the correlation between two variables. Correlation analysis or regression analysis can be used to examine the relationship between carbon dioxide emissions and temperature rise. Correlation

analysis is a statistical technique employed to assess the magnitude and direction of the association between two or more variables (Dean, n.d.). The process entails the computation of a correlation coefficient, a quantitative measure that assesses the extent of the relationship between the variables. The correlation coefficient is a statistical measure that varies between -1 and 1. When the coefficient is close to -1, it suggests a robust negative relationship. Conversely, a coefficient near 1 indicates a strong positive relationship. On the other hand, a coefficient close to 0 signifies the absence of any relationship. Regression analysis is a statistical technique employed to establish a mathematical model that describes the association between a dependent variable and one or more independent variables (Dean, n.d.). The process entails using a line or curve to establish a relationship with the data, thereby enabling the generation of predictions regarding the dependent variable by considering the values of the independent variables. Regression analysis encompasses various techniques, such as linear regression, multiple linear regression, and nonlinear regression, employed to model relationships between variables. Therefore, these methods can determine the strength and direction of the relationship between these two variables.

The second data mining objective, to analyse the influence of various countries based on aggregated data on emissions and temperature change, involves assessing how different countries' emissions impact temperature change. Clustering or segmentation techniques can group countries based on their emissions and temperature change data. Clustering is a method used to locate different subgroups within a more enormous collection (Dean, n.d.). When analysts divide the data into subgroups, often referred to as clusters, their goal is to distribute the data so that the cases within a group are pretty like one another, while the cases in other clusters are incredibly distinct from one another. On the other hand, segmentation refers to categorising consumers or other things into different groups based on the commonalities they share. When it comes to grouping and segmentation, there are a wide variety of algorithmic and methodological options. Examples of popular approaches are clustering techniques such as k-means, hierarchical clustering, and decision trees (Dean, n.d.). Using these methods, the data can be automatically segmented based on criteria, such as the degree of similarity or distance between two points in the data. These methods can identify patterns and trends in the data and understand how different countries contribute to emissions and temperature change.

The third data mining objective, to identify the countries with the highest average temperature increase and the highest total agrifood CO2 emissions in 2020, and analyse their contributions to the overall environmental impact, involves finding countries with the highest average temperature increase and the highest total agrifood CO2 emissions in 2020, and

understanding how their emissions contribute to the overall environmental impact. Descriptive statistics or ranking methods can identify the countries with the highest average temperature increase and the highest total agrifood CO2 emissions in 2020 (Marr, n.d.). Once these countries are identified, regression or decision tree analysis can be used to understand their contribution to the overall environmental impact. The process of clustering is one method that can be used to determine the existence of subgroups within a more extensive set. In dividing the data into subgroups, often referred to as clusters, analysts intend to distribute the data so that the cases within a group are incredibly like one another. However, the cases in other clusters are incredibly dissimilar to one another. The process of classifying consumers or other things into subcategories according to the shared characteristics of those subcategories is known as segmentation. When it comes to clustering and segmentation, there are a wide variety of options in terms of algorithms and methods. Common approaches include clustering techniques such as k-means, hierarchical clustering, and decision trees (Marr, n.d.). Using these methods, the data can be automatically segmented depending on criteria, such as similarities between the segments or distances. A decision tree is a type of decision support tool that uses a tree-like model of decisions and the probable repercussions of those actions. These potential implications include the outcomes of random events, the costs of resources, and the utility of those resources. Displaying an algorithm that consists solely of conditional control statements can be done in this manner. Decision trees are a prominent tool in machine learning, in addition to their widespread application in operations research, specifically in decision analysis. These trees are used to determine which approach is most likely to achieve a given objective.

5.2 Select the appropriate data mining methods based on discussion

- Examine the correlation between carbon dioxide (CO2) emissions within the agri-food sector and the subsequent temperature rise: regression analysis is utilised to investigate the association between carbon dioxide (CO2) emissions and the increase in temperature. This methodology facilitates the assessment of the magnitude and orientation of the association between the variables mentioned above.
- Analyse the influence of various countries based on aggregated data on emissions and temperature change: clustering techniques are utilised to categorise countries according to their emissions and temperature change data. This approach enables the identification of patterns and trends within the dataset, facilitating a comprehensive comprehension of the various countries' contributions to

emissions and temperature fluctuations.

- Identify the countries with the highest average temperature increase and the highest total agrifood CO2 emissions in 2020, and analyse their contributions to the overall environmental impact: descriptive statistics is utilised to ascertain the nations exhibiting the most significant average temperature increase and the highest total agrifood CO2 emissions in 2020, and to gain insights into their contributions to the overall environmental impact.

6. Data mining algorithms selection

6.1 Conduct exploratory analysis and discuss

The first data mining objective is to examine the correlation between carbon dioxide (CO2) emissions within the agri-food sector and the subsequent temperature rise, for which regression analysis is used. The second data mining goal is to analyse the influence of various countries based on aggregated data on emissions and temperature change, for which clustering is utilised. The third data mining objective is to identify the countries with the highest average temperature increase and the highest total agrifood CO2 emissions in 2020, and analyse their contributions to the overall environmental impact, for which descriptive statistics is used. Descriptive statistics is not a typical data mining method, and the results of the third goal are presented directly in the eighth step. This step discusses the regression analysis for the first objective and the clustering for the second goal.

6.1.1 Regression

Based on the first objective, regression analysis is utilised. Regression in machine learning is a supervised learning methodology wherein the algorithm is trained using input features and corresponding output labelslinear. Estimating how one variable affects another assist in establishing a relationship among the variables. Regression analysis aims to make predictions about a continuous dependent variable (y) by utilising one or more independent variables (x) as predictors.

Linear regression is widely recognised as the most employed regression analysis method due to its simplicity and effectiveness in prediction and forecasting (*Sklearn.Linear_model.LinearRegression*, n.d.). The assumption is made that a linear relationship exists between the input variables (x) and the single output variable (y). To be more precise, the value of y can be determined by computing a linear combination of the input

variables (x). Linear data is considered appropriate when it exhibits a linear pattern.

Polynomial regression is a statistical technique that extends the concept of linear regression by modelling the relationship between the independent variable, denoted as x, and the dependent variable, denoted as y, as a polynomial function of degree n (*Sklearn.Preprocessing.PolynomialFeatures*, n.d.). This method is appropriate in cases where the data exhibits a curved shape.

Ridge regression is a statistical technique employed in cases where the dataset exhibits multicollinearity, which refers to a high degree of correlation among the independent variables (*Sklearn.Linear_model.Ridge*, n.d.). Ridge regression is a statistical technique that introduces a degree of bias to the regression estimates, reducing the standard errors.

Like ridge regression, Lasso Regression can effectively nullify the influence of specific extraneous variables on the projected output (*Sklearn.Linear_model.Lasso*, n.d.).

The Elastic Net Regression technique compromises Ridge Regression and Lasso Regression. The Elastic Net method incorporates a dual penalty term and a mixing parameter to balance the Ridge and Lasso regularisation techniques (*Sklearn.Linear_model.ElasticNet*, n.d.).

Support Vector Regression (SVR) is a machine-learning algorithm for regression tasks. It is based on the Support Vector Machine (SVM) algorithm, primarily used for classification tasks (*Sklearn.Svm.SVR*, n.d.) . SV This represents an expansion of the Support Vector Machine (SVM) algorithm within the context of regression analysis. High-dimensional data is appropriate in this context.

Decision tree regression is an algorithm that utilises a decision tree model as a predictive tool to establish relationships between observations of an item and the corresponding conclusions regarding the item's target value (*Decision Tree Regression*, n.d.). This method is appropriate in cases where the input variables are categorical.

Random Forest Regression is an ensemble learning technique that involves the construction of multiple decision trees during the training phase (*Sklearn.Ensemble.RandomForestRegressor*, n.d.). The final prediction is obtained by taking the average of the predictions made by each tree. This approach is appropriate when there are both categorical and numerical input variables.

Gradient Boosting regression is an ensemble machine learning algorithm capable of addressing regression and classification problems (*Sklearn.Ensemble.GradientBoostingRegressor*, n.d.). Gradient boosting (GB) constructs an additive model forward stage-wise, optimising loss functions that are differentiable in an

arbitrary manner.

6.1.2 Clustering

Based on the second goal, clustering is used. Clustering, as employed in machine learning, is an unsupervised learning technique. This method aims to identify significant patterns, elucidate fundamental mechanisms, ascertain generative characteristics, and discern inherent categorisations within a given set of instances. The process of clustering involves partitioning a population or set of data points into multiple groups to ensure that data points within the same group exhibit more significant similarity to one another compared to those in different groups (Dean, n.d.).

The K-means algorithm is a popular clustering technique used in machine learning and data analysis. Clustering refers to grouping similar data points based on their inherent (*Sklearn.Cluster.KMeans*, n.d.). The discussed algorithm is a centroid-based clustering algorithm widely utilised in various applications. The given algorithm is designed to divide a set of n observations into k distinct clusters, assigning each observation to the cluster whose mean is closest to it. Linear data is considered appropriate when it exhibits a linear pattern.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is an algorithm that belongs to the category of density-based clustering methods. The algorithm identifies regions characterised by a high concentration of data points called clusters (*Sklearn.Cluster.DBSCAN*, n.d.). One notable aspect of this phenomenon is the ability of clusters to exhibit various shapes.

Gaussian Mixture Models (GMMs) are a statistical modelling technique commonly used in machine learning and data analysis. The proposed methodology adopts a distribution-based clustering approach, wherein each data point is assigned to a cluster based on the likelihood of its membership in that cluster (*2.1. Gaussian Mixture Models*, n.d.). If one is still determining the data distribution, it is advisable to explore alternative algorithms.

Hierarchical clustering is a method that constructs a dendrogram, representing a hierarchical structure of clusters. Hierarchical clustering is particularly well-suited for analysing hierarchical data structures, such as taxonomies. Agglomerative clustering is a hierarchical clustering technique that employs a bottom-up methodology (*Sklearn.Cluster.AgglomerativeClustering*, n.d.). Divisive clustering is a variant of hierarchical clustering that employs a top-down methodology.

6.2 Select data mining algorithms based on discussion

The random forest is selected for the first data mining objective to examine the correlation between carbon dioxide (CO2) emissions within the agri-food sector and the subsequent temperature rise. The random forest algorithm is classified as an ensemble learning technique that combines multiple decision trees to enhance the precision and resilience of predictive models (Dean, n.d.). The algorithm in question is widely recognised in machine learning and can perform classification and regression tasks. The algorithm generates numerous decision trees during the training phase. It establishes the class that manifests itself most frequently among the trees (for classification) or the average prediction made by the various trees (for regression). Random forests are renowned for their adeptness in managing extensive datasets, feature spaces with numerous dimensions, and intricate interdependencies among features. Moreover, these tools are known for their user-friendly nature and straightforward interpretation, rendering them highly favoured across various domains. Additionally, this project also uses a linear algorithm for the first data mining objective. Linear regression is a supervised machine learning algorithm to determine the linear association between a dependent variable and one or more independent features. The dependent variable in this study is the average temperature rise, while the independent variables consist of the different factors associated with CO2 emissions within the agri-food sector. Linear regression is a statistical technique that interprets how alterations in the independent variables correspond to variations in the dependent variable. The linear regression algorithm can determine the optimal linear equation for predicting the value of the dependent variable, taking into account the independent variables. The coefficients associated with the independent variables in the equation provide insights into the extent to which each variable contributes to the predictive model of the dependent variable.

The K-means is selected for the second data mining goal, to analyse the influence of various countries based on aggregated data on emissions and temperature change. K-means clustering is a technique in vector quantisation that originated in the field of signal processing (Dean, n.d.). Its objective is to divide a set of n observations into k clusters. Every individual observation is assigned to the cluster with the closest mean, which acts as the prototype or representative of that cluster. This technique is an unsupervised learning method utilised to classify unlabeled data. This is achieved by grouping the data based on their shared features instead of predefined categories. K-means clustering is commonly employed in situations where there is no predetermined outcome variable being targeted for prediction. However, this technique is employed when there is a specific set of features that one wishes to utilise to identify groups of observations that exhibit similar characteristics. The k-means algorithm is

designed to be employed exclusively when all the features in a dataset are numeric. There exist strategies for accommodating categorical features within data adaptation processes; however, it is generally recommended that a substantial proportion of the features be numeric (Dean, n.d.). In addition, Agglomerative clustering is also used for the second data mining purpose. Agglomerative clustering, alternatively referred to as Hierarchical Agglomerative Clustering (HAC), is a form of hierarchical clustering that follows a "bottom-up" approach (Dean, n.d.). In the context of this clustering technique, it is initially established that each data point is considered an independent cluster. As the algorithm progresses through the hierarchy, clusters are merged in pairs. The algorithm commences by considering each object as an individual cluster. Subsequently, clusters are merged in pairs until all clusters have been amalgamated into a singular cluster encompassing all objects (Dean, n.d.). Unsupervised machine learning tasks involve the utilisation of unlabeled datasets, wherein the objective is to cluster similar instances together. This method is highly suitable for conducting exploratory data analysis and identifying inherent clusters within a given dataset. This method is recommended over K-means when the goal is to employ the trained algorithm to make inferences on novel, unseen observations. Additionally, this method proves advantageous in cases where there is a need to generate a dendrogram, a graphical representation of objects organised in a tree-like structure.

6.3 Build/Select appropriate models and choose relevant parameters

6.3.1 Regression

For the first data mining objective, the random forest algorithm model and the linear regression model are built. There are four linear regression models built. The first one is to impute all features excluding ‘Average Temperature’. The second one is to split the dataset into the training dataset and testing dataset. The third one is to select 10 features by using ANOVA (Analysis of Variance). The fourth one is established by using statsmodels.api (Perktold et al., 2023). Random forest regression model is established to examine the correlative importance between the CO2 emissions of the agrifood factor and the subsequent temperature increase. Figure 29, Figure 30, Figure 31, Figure 32, and Figure 33 show the above five models and their parameters.

Figure 29. The first linear regression model and parameters

```
Y = agrifood_dm_df1['Average Temperature']
X = agrifood_dm_df1.drop(['Average Temperature'], axis=1)

X_all = X.values
y = Y.values
r1 = LinearRegression().fit(X_all, y)
print(r1.intercept_, r1.coef_)
r1.score(X_all, y)
```

Figure 30. The second linear regression model and parameters

```
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2)
r2 = LinearRegression().fit(X_train, Y_train)
print(r2.intercept_, r2.coef_)
r2.score(X_train, Y_train)
r2.score(X_test, Y_test)
```

Figure 31. The third linear regression model and parameters

```
SelectKBest(f_regression, k=10).fit(X, Y).get_feature_names_out()
```

```
array(['Year', 'Crop Residues', 'Food Transport',
       'Food Household Consumption', 'Food Retail', 'IPPU',
       'Manure applied to Soils', 'Manure Management',
       'On-farm energy use', 'Urban population'], dtype=object)
```

```
X_1 = X[['Year', 'Crop Residues', 'Food Transport',
       'Food Household Consumption', 'Food Retail', 'IPPU',
       'Manure applied to Soils', 'Manure Management',
       'On-farm energy use', 'Urban population']]
r3 = LinearRegression().fit(X_1, Y)
print(r3.intercept_, r3.coef_)
r3.score(X_1, y)
```

Figure 32. The fourth linear regression model and parameters

```
X_sig = X[['Area', 'Year', 'Food Transport', 'Net Forest conversion',
           'Food Household Consumption', 'On-farm Electricity Use', 'Manure applied to Soils',
           'Emission_per_capita', 'Agrifood Systems Waste Disposal', 'IPPU']]
r6 = sm.OLS(Y, sm.add_constant(X_sig)).fit()
r6.summary()
```

Figure 33. Random forest regression model and parameters

```
agrifood_emission_df1 = agrifood_emission_df.copy()
X = agrifood_dm_df1.drop(['Area', 'Year', 'Average Temperature', 'Pred_r1','Pred_r2','Pred_r3','Pred_r6'], axis=1)
y = agrifood_dm_df1['Average Temperature']

selector = SelectFromModel(rf(n_estimators=100, random_state=0), max_features=18)
selector.fit(X, y)

support = selector.get_support()
features = X.loc[:,support].columns.tolist()

print(features)
```

6.3.2 Clustering

For the second goal, K-means clustering and Agglomerative clustering models are built. The parameter n_cluster is determined as 5 (Figure 34 & Figure 35).

Figure 34. Distortion Score Elbow for K-Means Clustering

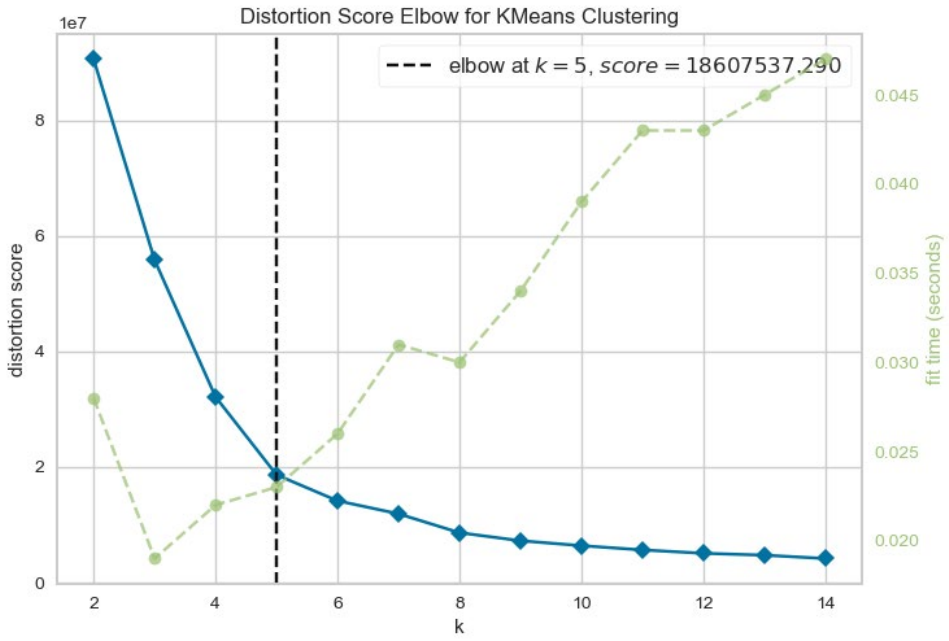
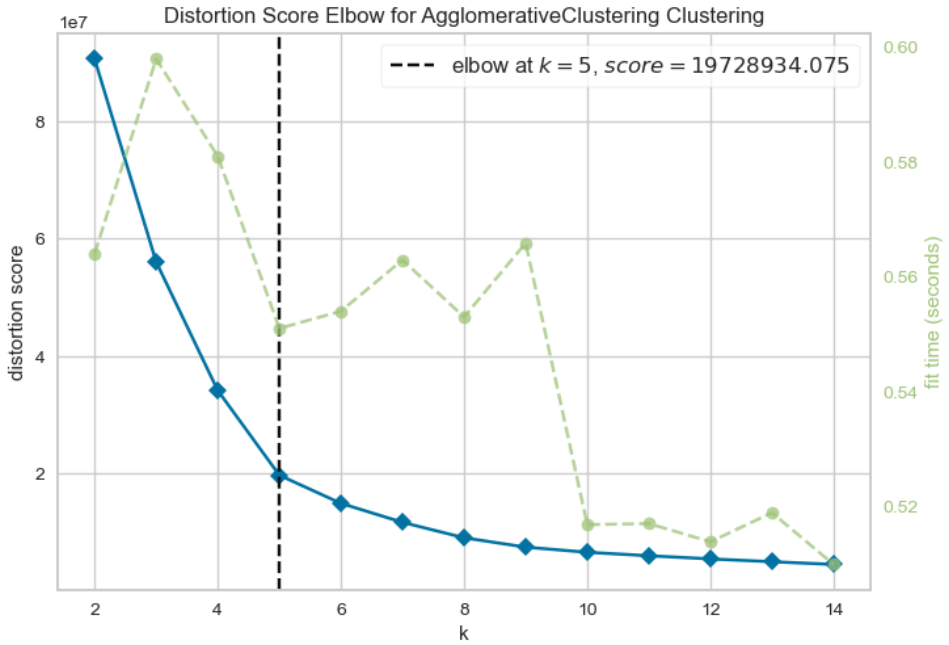


Figure 35. Distortion Score Elbow for Agglomerative Clustering



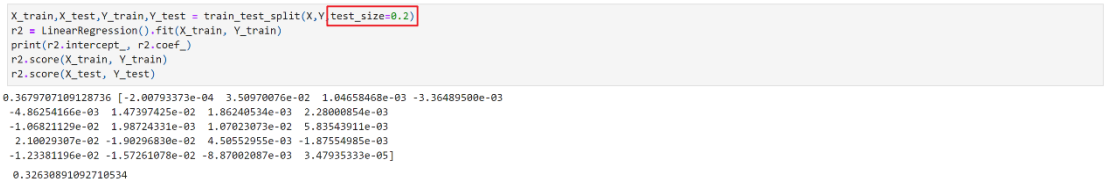
7. Data mining

7.1 Create and justify test designs

The regression is a supervised learning algorithm. Thus, training and testing sets are separated from the whole dataset. The training set is 80% of the whole dataset, and the testing

set is 20% (Figure 36). The utilisation of an 80/20 split for training and testing sets is widely acknowledged as a prevalent guideline within machine learning. The guideline above possesses broad applicability across various models and problem domains. The rationale behind this division is allocating a subset of the data to evaluate the model's performance while utilising the more significant data to train the model. The exact training-to-testing data ratio may vary depending on the analysis's requirements and the dataset's inherent attributes. The crucial aspect is to guarantee sufficient data in the training set to effectively train the model while simultaneously setting aside an adequate amount of data in the testing set to yield a dependable evaluation of the model's performance on unfamiliar data.

Figure 36. Training set and testing set



The clustering is an unsupervised learning algorithm. Unsupervised learning algorithms generally do not necessitate dividing data into separate training and testing sets. Unsupervised learning algorithms are advantageous due to their ability to train models without the need for labelled data. Instead, these algorithms rely on the calculation of relationships between data points to uncover the underlying structure of the data. Consequently, the entirety of the dataset is utilised to train an unsupervised learning model.

7.2 Conduct data mining – regression and clustering

7.2.1 Regression

Four linear regression models and one random forest regression model run successfully. Figure 37, Figure 38, Figure 39, Figure 40, Figure 41 , and Figure 42 show the codes and the results of these five regression models.

Figure 37. The first linear regression model

```
Y = agrifood_dm_df1['Average Temperature']
X = agrifood_dm_df1.drop(['Average Temperature'], axis=1)

X_all = X.values
y = Y.values
r1 = LinearRegression().fit(X_all, y)
print(r1.intercept_, r1.coef_)
r1.score(X_all, y)
```

```
0.3728338709365958 [-2.32031657e-04  3.48712040e-02  6.31184275e-04 -4.01469024e-03
 -4.76076315e-03  1.43087991e-02  1.83365665e-03  2.93440912e-03
 -1.01028384e-02  1.98179995e-03  1.13741316e-02  4.05738695e-03
  2.14144675e-02 -2.04033198e-02  5.68584372e-03 -1.44822104e-03
 -1.29411939e-02 -1.21596444e-02 -1.05432968e-02  2.84857910e-05]
```

```
0.32382836946269233
```

```
agrifood_dm_df1['Pred_r1'] = r1.predict(X_all)
agrifood_dm_df1
```

ood sport	Net Forest conversion	Food Household Consumption	Food Retail	...	IPPU	Manure applied to Soils	Manure left on Pasture	Manure Management	On-farm energy use	Updated_total_emission	Urban population	Total_population	Emission_per_capita	Pred_r1
0307	-0.005524	-0.088036	-0.059473	...	-0.108416	0.276634	0.269721	-0.058480	0.119616	-0.293009	0.029327	0.326378	-0.248271	0.369836
1921	-0.005524	-0.087042	-0.052769	...	-0.107237	0.299558	0.298836	-0.035514	0.135709	-0.289349	0.050322	0.329213	-0.247180	0.404394
8618	-0.005524	-0.086844	-0.043720	...	-0.106389	0.289186	0.297208	-0.028748	0.102248	-0.289549	0.077928	0.403063	-0.249620	0.437938
7732	-0.005524	-0.083802	-0.086336	...	-0.109881	0.280899	0.292605	-0.025598	0.099069	-0.289338	0.109114	0.512595	-0.252270	0.471905
8049	-0.005524	-0.081156	-0.077815	...	-0.113040	0.296837	0.312856	-0.010325	0.116706	-0.285444	0.139586	0.594297	-0.252968	0.505505
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9171	1.948657	0.033819	0.258339	...	-0.000022	-0.166417	0.763197	-0.094795	0.206113	2.141051	0.354641	0.537860	0.334230	1.170040
3115	1.948657	-0.000253	0.260532	...	0.005063	-0.206672	0.773555	-0.121611	0.190213	2.107279	0.368645	0.554653	0.314034	1.203736
3581	1.948657	0.005430	0.305779	...	0.017896	-0.207105	0.793291	-0.119940	0.246345	2.120187	0.383379	0.571596	0.305364	1.238349
2285	1.948657	0.017804	0.353180	...	0.014502	-0.194544	0.810150	-0.109764	0.228462	2.153656	0.398765	0.588616	0.301659	1.271158
8668	1.948657	0.007389	0.377439	...	0.013580	-0.195642	0.810597	-0.110549	0.228462	2.090691	0.414763	0.606346	0.276157	1.305696

Figure 38. The second linear regression model

```
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2)
r2 = LinearRegression().fit(X_train, Y_train)
print(r2.intercept_, r2.coef_)
r2.score(X_train, Y_train)
r2.score(X_test, Y_test)
```

```
0.3679707109128736 [-2.00793373e-04  3.50970076e-02  1.04658468e-03 -3.36489500e-03
 -4.86254166e-03  1.47397425e-02  1.86240534e-03  2.28000854e-03
 -1.06821129e-02  1.98724331e-03  1.07023073e-02  5.83543911e-03
  2.10029307e-02 -1.90296830e-02  4.50552955e-03 -1.87554985e-03
 -1.23381196e-02 -1.57261078e-02 -8.87002087e-03  3.47935333e-05]
```

```
0.3263089102710534
```

```
agrifood_dm_df1['Pred_r2']=r2.predict(X)
agrifood_dm_df1
```

Food sport	Net Forest conversion	Food Household Consumption	Food Retail	...	Manure applied to Soils	Manure left on Pasture	Manure Management	On-farm energy use	Updated_total_emission	Urban population	Total_population	Emission_per_capita	Pred_r1	Pred_r2
20307	-0.005524	-0.088036	-0.059473	...	0.276634	0.269721	-0.058480	0.119616	-0.293009	0.029327	0.326378	-0.248271	0.369836	0.365612
21921	-0.005524	-0.087042	-0.052769	...	0.299558	0.298836	-0.035514	0.135709	-0.289349	0.050322	0.329213	-0.247180	0.404394	0.400318
28618	-0.005524	-0.086844	-0.043720	...	0.289186	0.297208	-0.028748	0.102248	-0.289549	0.077928	0.403063	-0.249620	0.437938	0.434094
27732	-0.005524	-0.083802	-0.086336	...	0.280899	0.292605	-0.025598	0.099069	-0.289338	0.109114	0.512595	-0.252270	0.471905	0.468404
28049	-0.005524	-0.081156	-0.077815	...	0.296837	0.312856	-0.010325	0.116706	-0.285444	0.139586	0.594297	-0.252968	0.505505	0.502259
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
39171	1.948657	0.033819	0.258339	...	-0.166417	0.763197	-0.094795	0.206113	2.141051	0.354641	0.537860	0.334230	1.170040	1.179968
43115	1.948657	-0.000253	0.260532	...	-0.206672	0.773555	-0.121611	0.190213	2.107279	0.368645	0.554653	0.314034	1.203736	1.214000
03581	1.948657	0.005430	0.305779	...	-0.207105	0.793291	-0.119940	0.246345	2.120187	0.383379	0.571596	0.305364	1.238349	1.248843
72285	1.948657	0.017804	0.353180	...	-0.194544	0.810150	-0.109764	0.228462	2.153656	0.398765	0.588616	0.301659	1.271158	1.281736
28668	1.948657	0.007389	0.377439	...	-0.195642	0.810597	-0.110549	0.228462	2.090691	0.414763	0.606346	0.276157	1.305696	1.316489

Figure 39. The third linear regression model

```
X_1 = X[['Year', 'Crop Residues', 'Food Transport',
       'Food Household Consumption', 'Food Retail', 'IPPU',
       'Manure applied to Soils', 'Manure Management',
       'On-farm energy use', 'Urban population']]
r3 = LinearRegression().fit(X_1, Y)
print(r3.intercept_, r3.coef_)
r3.score(X_1, y)
```

```
0.34337921141769356 [ 0.03461675 -0.00820779  0.01196814  0.00207326 -0.0050108   0.00452587
  0.02906221 -0.00246806 -0.00358496 -0.02480448]
0.3140645099325349
```

```
agrifood_dm_df1['Pred_r3'] = r3.predict(X_1)
agrifood_dm_df1
```

Food nsport	Net Forest conversion	Food Household Consumption	Food Retail	...	Manure left on Pasture	Manure Management	On-farm energy use	Updated_total_emission	Urban population	Total_population	Emission_per_capita	Pred_r1	Pred_r2	Pred_r3
20307	-0.005524	-0.088036	-0.059473	...	0.269721	-0.058480	0.119616	-0.293009	0.029327	0.326378	-0.248271	0.369836	0.365612	0.345974
21921	-0.005524	-0.087042	-0.052769	...	0.298836	-0.035514	0.135709	-0.289349	0.050322	0.329213	-0.247180	0.404394	0.400318	0.380475
28618	-0.005524	-0.086844	-0.043720	...	0.297208	-0.028748	0.102248	-0.289549	0.077928	0.403063	-0.249620	0.437938	0.434094	0.414424
27732	-0.005524	-0.083802	-0.086336	...	0.292605	-0.025598	0.099069	-0.289338	0.109114	0.512595	-0.252270	0.471905	0.468404	0.447355
28049	-0.005524	-0.081156	-0.077815	...	0.312856	-0.010325	0.116706	-0.285444	0.139586	0.594297	-0.252968	0.505505	0.502259	0.481231
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
39171	1.948657	0.033819	0.258339	...	0.763197	-0.094795	0.206113	2.141051	0.354641	0.537860	0.334230	1.170040	1.179968	1.229401
43115	1.948657	-0.000253	0.260532	...	0.773555	-0.121611	0.190213	2.107279	0.368645	0.554653	0.314034	1.203736	1.214000	1.261633
03581	1.948657	0.005430	0.305779	...	0.793291	-0.119940	0.246345	2.120187	0.383379	0.571596	0.305364	1.238349	1.248843	1.296198
72285	1.948657	0.017804	0.353180	...	0.810150	-0.109764	0.228462	2.153656	0.398765	0.588616	0.301659	1.271158	1.281736	1.331912
28668	1.948657	0.007389	0.377439	...	0.810597	-0.110549	0.228462	2.090691	0.414763	0.606346	0.276157	1.305696	1.316489	1.363800

Figure 40. The fourth linear regression model – part 1

```
X_sig = X[['Area', 'Year', 'Food Transport', 'Net Forest conversion',
           'Food Household Consumption', 'On-farm Electricity Use', 'Manure applied to Soils',
           'Emission_per_capita', 'Agrifood Systems Waste Disposal', 'IPPU']]
r6 = sm.OLS(Y, sm.add_constant(X_sig)).fit()
r6.summary()
```

OLS Regression Results

Dep. Variable:	Average Temperature	**R-squared:**	0.309
Model:	OLS	**Adj. R-squared:**	0.308
Method:	Least Squares	**F-statistic:**	311.2
Date:	Thu, 21 Sep 2023	**Prob (F-statistic):**	0.00
Time:	15:05:09	**Log-Likelihood:**	-4505.3
No. Observations:	6965	**AIC:**	9033.
Df Residuals:	6954	**BIC:**	9108.
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.3728	0.015	25.226	0.000	0.344	0.402
Area	-0.0002	8.3e-05	-2.194	0.028	-0.000	-1.94e-05
Year	0.0343	0.001	54.542	0.000	0.033	0.036
Food Transport	0.0018	0.002	1.019	0.308	-0.002	0.005
Net Forest conversion	-0.0006	0.000	-1.512	0.131	-0.001	0.000
Food Household Consumption	-0.0007	0.002	-0.356	0.722	-0.005	0.003
On-farm Electricity Use	-0.0011	0.001	-1.804	0.071	-0.002	9.65e-05
Manure applied to Soils	0.0175	0.002	8.154	0.000	0.013	0.022
Emission_per_capita	1.148e-05	1.94e-05	0.593	0.553	-2.65e-05	4.94e-05
Agrifood Systems Waste Disposal	-0.0142	0.002	-5.752	0.000	-0.019	-0.009

Figure 41. The fourth linear regression model – part 2

```
agrifood_dm_df1['Pred_r6'] = r6.predict(sm.add_constant(X_sig))
agrifood_dm_df1
```

Food nsport	Net Forest conversion	Food Household Consumption	Food Retail	...	Manure Management	On-farm energy use	Updated_total_emission	Urban population	Total_population	Emission_per_capita	Pred_r1	Pred_r2	Pred_r3	Pred_r6
20307	-0.005524	-0.088036	-0.059473	...	-0.058480	0.119616	-0.293009	0.029327	0.326378	-0.248271	0.369836	0.365612	0.345974	0.378519
21921	-0.005524	-0.087042	-0.052769	...	-0.035514	0.135709	-0.289349	0.050322	0.329213	-0.247180	0.404394	0.400318	0.380475	0.413122
28618	-0.005524	-0.086844	-0.043720	...	-0.028748	0.102248	-0.289549	0.077928	0.403063	-0.249620	0.437938	0.434094	0.414424	0.447064
27732	-0.005524	-0.083802	-0.086336	...	-0.025598	0.099069	-0.289338	0.109114	0.512595	-0.252270	0.471905	0.468404	0.447355	0.480975
28049	-0.005524	-0.081156	-0.077815	...	-0.010325	0.116706	-0.285444	0.139586	0.594297	-0.252968	0.505505	0.502259	0.481231	0.515350
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
)39171	1.948657	0.033819	0.258339	...	-0.094795	0.206113	2.141051	0.354641	0.537860	0.334230	1.170040	1.179968	1.229401	1.215839
)43115	1.948657	-0.000253	0.260532	...	-0.121611	0.190213	2.107279	0.368645	0.554653	0.314034	1.203736	1.214000	1.261633	1.249658
03581	1.948657	0.005430	0.305779	...	-0.119940	0.246345	2.120187	0.383379	0.571596	0.305364	1.238349	1.248843	1.296198	1.283879
)72285	1.948657	0.017804	0.353180	...	-0.109764	0.228462	2.153656	0.398765	0.588616	0.301659	1.271158	1.281736	1.331912	1.318254
)28668	1.948657	0.007389	0.377439	...	-0.110549	0.228462	2.090691	0.414763	0.606346	0.276157	1.305696	1.316489	1.363800	1.352390

Figure 42. Random forest regression model

```
agrifood_emission_df1 = agrifood_emission_df.copy()
X = agrifood_dm_df1.drop(['Area', 'Year', 'Average Temperature', 'Pred_r1','Pred_r2','Pred_r3','Pred_r6'], axis=1)
y = agrifood_dm_df1['Average Temperature']

selector = SelectFromModel(rf(n_estimators=100, random_state=0), max_features=18)
selector.fit(X, y)

support = selector.get_support()
features = X.loc[:,support].columns.tolist()

print(features)
```

```
['Crop Residues', 'Rice Cultivation', 'Food Transport', 'Food Retail', 'IPPU', 'Manure left on Pasture', 'Manure Management', 'On-farm energy use', 'Emission_per_capita']
```

```
# Create a DataFrame to store feature names and their importances
feature_df = pd.DataFrame({'Feature': X.columns, 'Importance': rf(n_estimators=100, random_state=0).fit(X,y).feature_importances_})

# Sort the DataFrame by importance in descending order
feature_df = feature_df.sort_values(by='Importance', ascending=False)
```

```
feature_df
```

	Feature	Importance
6	Food Retail	0.117665
11	Manure left on Pasture	0.069527
1	Rice Cultivation	0.066542
13	On-farm energy use	0.062608
17	Emission_per_capita	0.057857
0	Crop Residues	0.057337
3	Food Transport	0.056783
12	Manure Management	0.056255
9	IPPU	0.055783
14	Updated_total_emission	0.054713
5	Food Household Consumption	0.052787

7.2.2 Clustering

The K-Means clustering, and Agglomerative clustering models run successfully. Figure 43 and Figure 44 show the codes and the results of these two clustering models.

Figure 43. K-Means clustering model

```
cl_2= KMeans(n_clusters=5, init='k-means++').fit(agrifood_dm_df2)
agrifood_cl_1=pd.concat([agrifood_dm_df2,pd.Series(cl_2.labels_,name='cluster_1',\
                                    dtype='category')],axis=1)

agrifood_cl_1['cluster_1'].value_counts()
```

```
cluster_1
4    3387
0    3279
3     234
2      34
1      31
Name: count, dtype: int64
```

Figure 44. Agglomerative clustering model

```
cl_4=AgglomerativeClustering(n_clusters=5, linkage='ward').fit(agrifood_dm_df2)

agrifood_cl_2=pd.concat([agrifood_dm_df2,pd.Series(cl_4.labels_,name='cluster_2',\
                                                   dtype='category')],axis=1)

agrifood_cl_2['cluster_2'].value_counts()
```

```
cluster_2
0    3844
3    2781
4     275
2      34
1      31
Name: count, dtype: int64
```

7.3 Search for patterns

7.3.1 Regression

There are two regression models, including the random forest, and the linear regression, running successfully, so two patterns are based on different models, respectively (Figure 45, Figure 46). The linear regression model pattern presents that splitting the dataset into the training dataset and the testing dataset is the best model. The random forest regression model pattern interprets that the most critical agri-food factor relative to the average temperature rise is food retail.

Figure 45. Linear regression model pattern

```
pred_df = round(agrifood_dm_df1[['Average Temperature', 'Pred_r1','Pred_r2','Pred_r3','Pred_r6']].corr(), 2)
pred_df
```

	Average Temperature	Pred_r1	Pred_r2	Pred_r3	Pred_r6
Average Temperature	1.00	0.57	0.57	0.56	0.56
Pred_r1	0.57	1.00	1.00	0.98	0.98
Pred_r2	0.57	1.00	1.00	0.99	0.98
Pred_r3	0.56	0.98	0.99	1.00	0.99
Pred_r6	0.56	0.98	0.98	0.99	1.00

Figure 46. Random forest regression model pattern

	Feature	Importance
6	Food Retail	0.117665
11	Manure left on Pasture	0.069527
1	Rice Cultivation	0.066542
13	On-farm energy use	0.062608
17	Emission_per_capita	0.057857
0	Crop Residues	0.057337
3	Food Transport	0.056783
12	Manure Management	0.056255
9	IPPU	0.055783
14	Updated_total_emission	0.054713
5	Food Household Consumption	0.052787
7	On-farm Electricity Use	0.052015
10	Manure applied to Soils	0.045006
8	Agrifood Systems Waste Disposal	0.044563
15	Urban population	0.041505
16	Total_population	0.041251
2	Pesticides Manufacturing	0.040175
4	Net Forest conversion	0.027627

7.3.2 Clustering

Figure 47 shows Distortion Score and Silhouette Score for K-Means Clustering. Figure 48 show Distortion Score and Silhouette Score for Agglomerative Clustering. There are 5 clusters, and the results will be presented in the section 8.

Figure 47. Distortion Score and Silhouette Score for K-Means Clustering

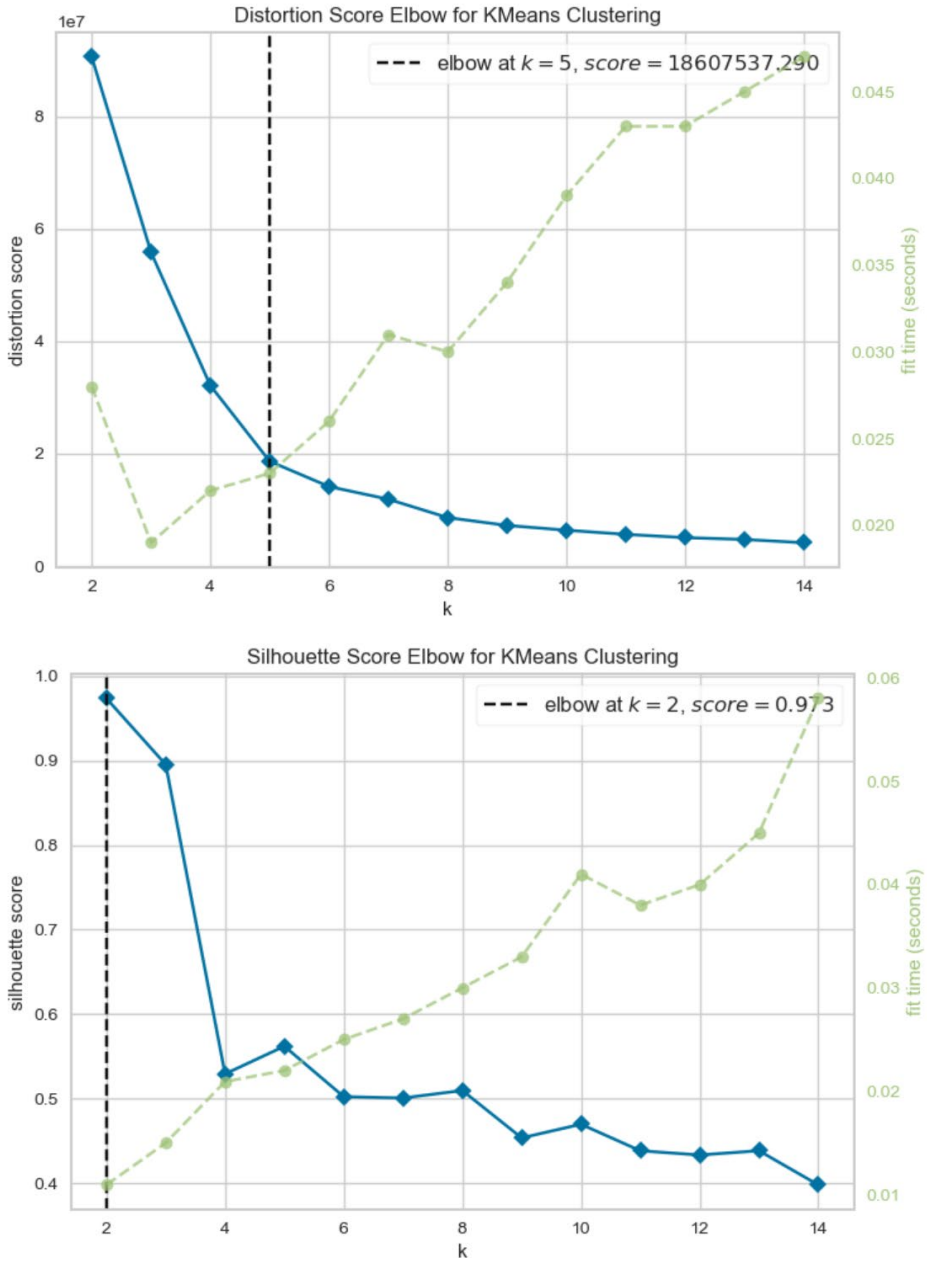
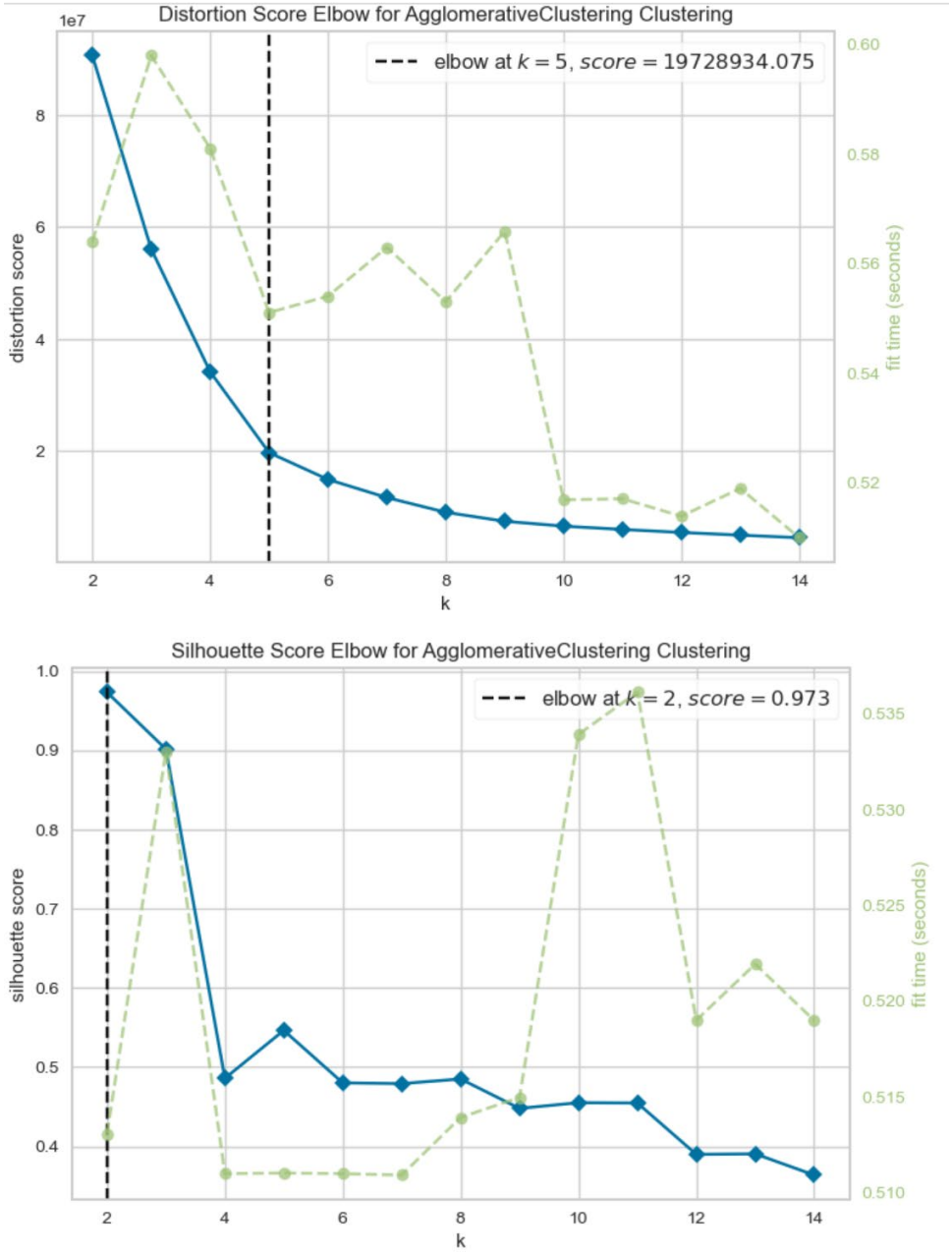


Figure 48. Distortion Score and Silhouette Score for Agglomerative Clustering



8. Interpretation

8.1 Study and discuss the mined patterns

8.1.1 Regression

Based on the random forest regression algorithms, the top 10 most crucial agri-food features, which affect the subsequent temperature rise, are ‘Food Retail’, ‘Manure left on Pasture’, ‘Rice Cultivation’, ‘On-farm energy use’, ‘Emission_per_capita’, ‘Crop Residues’, ‘Food Transport’, ‘Manure Management’, ‘IPPU’, and ‘Updated_total_emission’ (Figure 49).

Figure 49. Important features based on random forest regression model

	Feature	Importance
6	Food Retail	0.117665
11	Manure left on Pasture	0.069527
1	Rice Cultivation	0.066542
13	On-farm energy use	0.062608
17	Emission_per_capita	0.057857
0	Crop Residues	0.057337
3	Food Transport	0.056783
12	Manure Management	0.056255
9	IPPU	0.055783
14	Updated_total_emission	0.054713
5	Food Household Consumption	0.052787
7	On-farm Electricity Use	0.052015
10	Manure applied to Soils	0.045006
8	Agrifood Systems Waste Disposal	0.044563
15	Urban population	0.041505
16	Total_population	0.041251
2	Pesticides Manufacturing	0.040175
4	Net Forest conversion	0.027627

8.1.2 Clustering

The high CO2 emission countries have more population than the low CO2 emission countries (Figure 50 & Figure 51). The difference in average temperature rise between the high CO2 emission countries and the low CO2 emission countries is not quite significant (Figure 52 & Figure 53).

Figure 50. CO2 emissions and population of K-Means Clustering

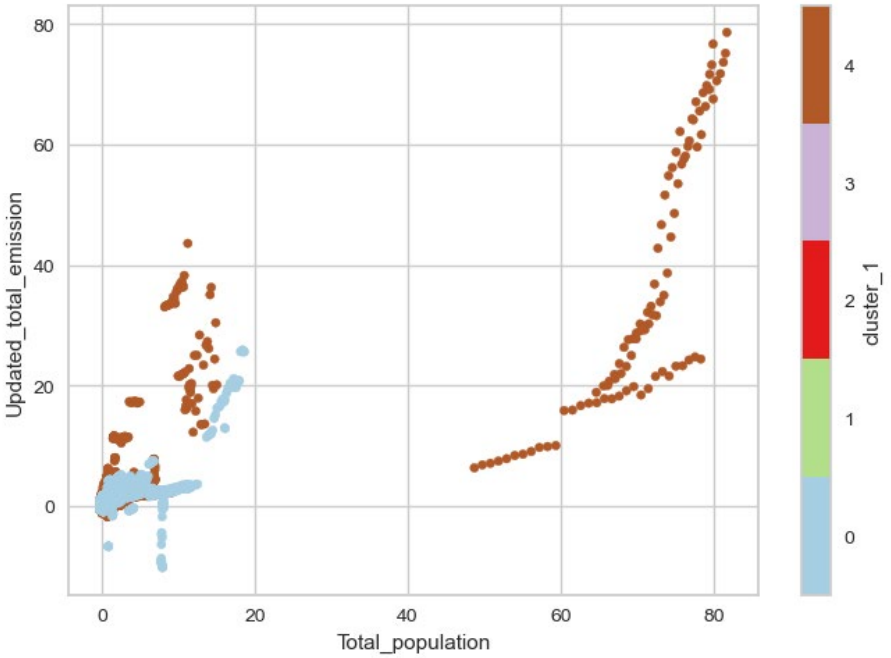


Figure 51. CO2 emissions and average temperature of K-Means Clustering

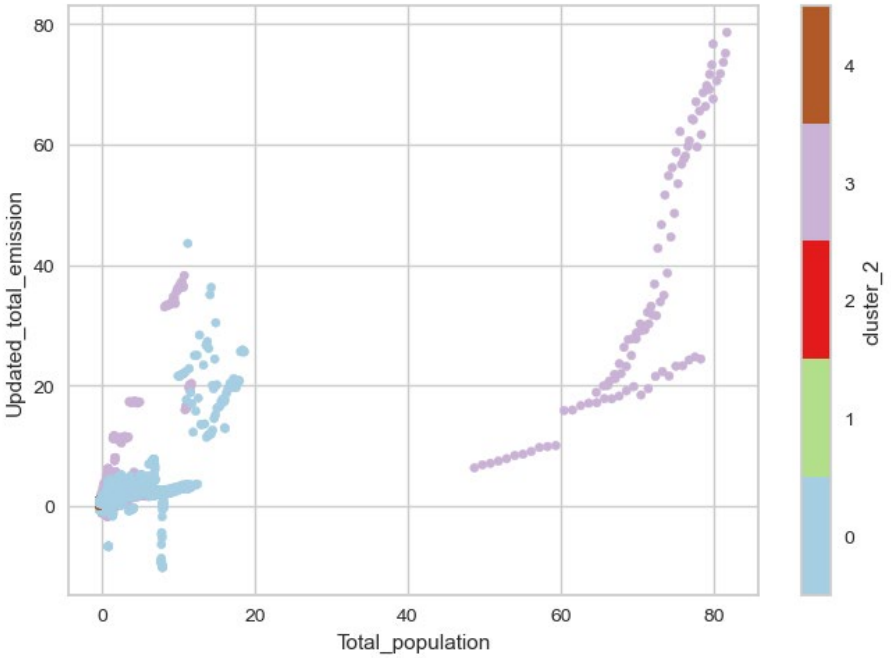


Figure 52. CO2 emissions and population of Agglomerative Clustering

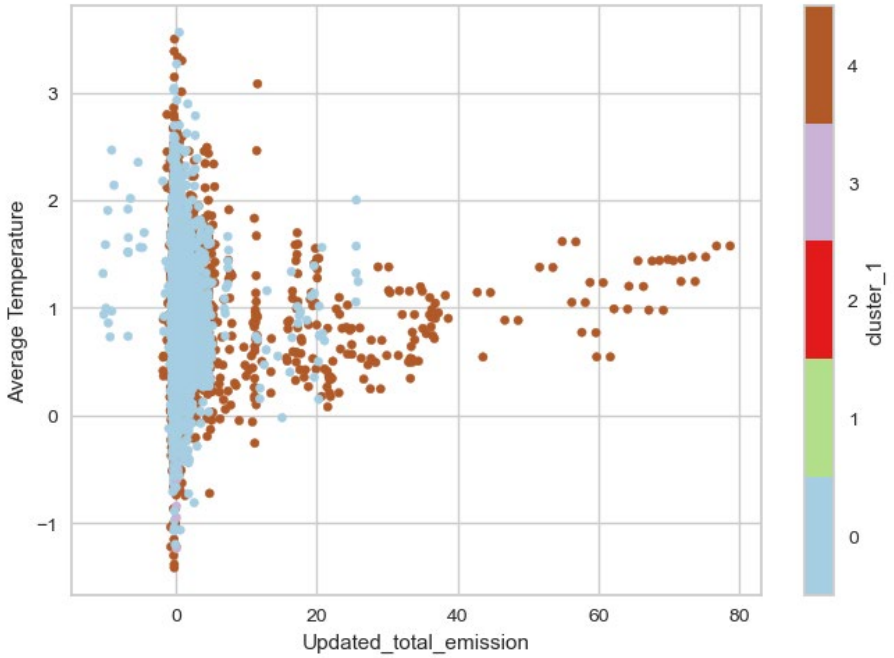
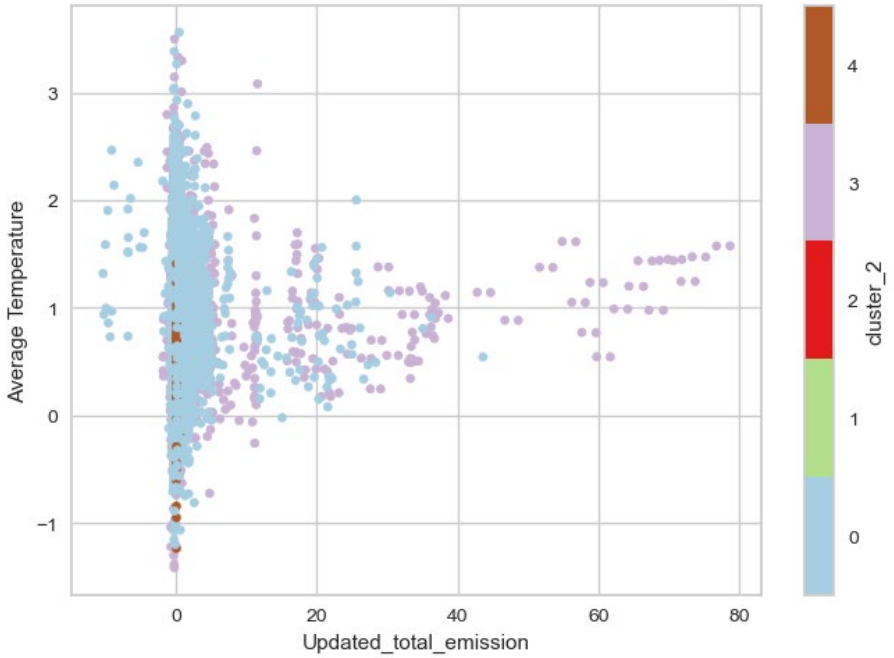


Figure 53. CO2 emissions and average temperature of Agglomerative Clustering



8.2 Visualize the data, results, models, and patterns

8.2.1 The first data mining objective

The first data mining objective is to examine the correlation between CO2 emissions within the agri-food sector and the subsequent temperature rise.

Figure 54. Correlative importance of different features

Figure 55. Predictor Importance based on random forest regression model

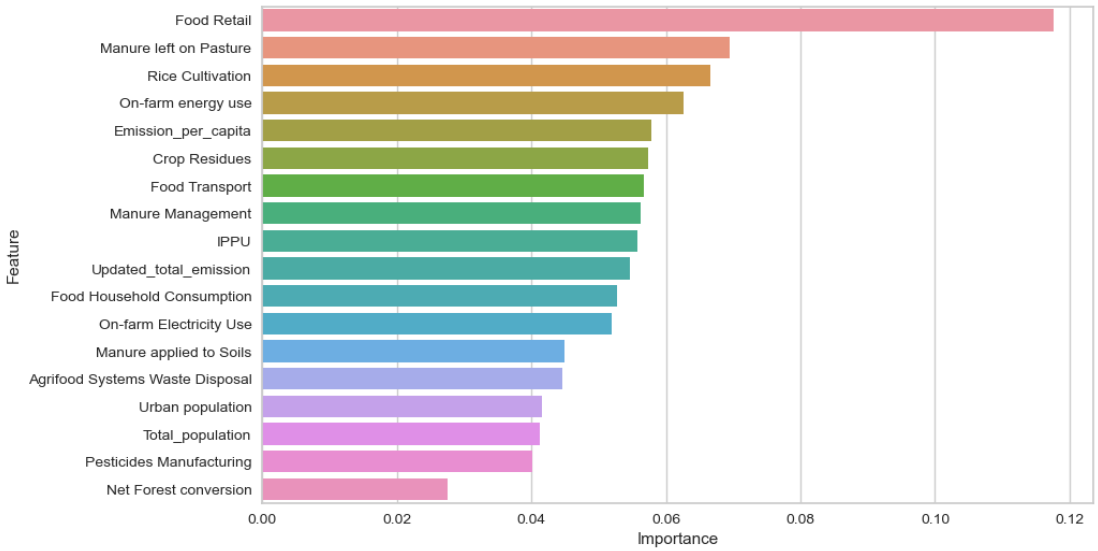
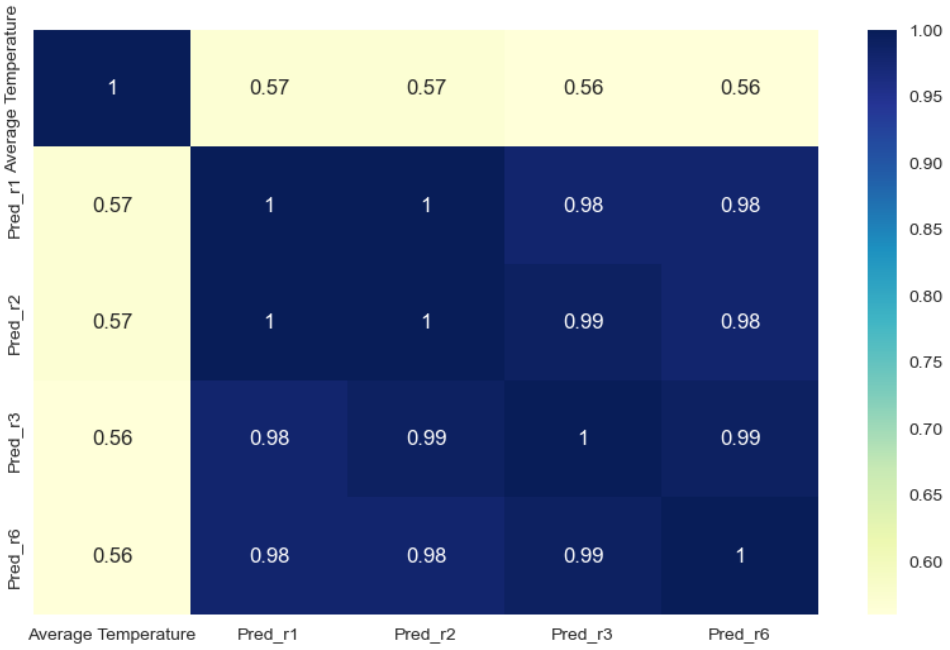


Figure 56. Accuracy of different linear regression model



8.2.2 The second data mining objective

The second data mining objective is to anlalyse the influence of various countries based on aggregated data on emissions and temperature change.

Figure 57. K-Means Clustering result

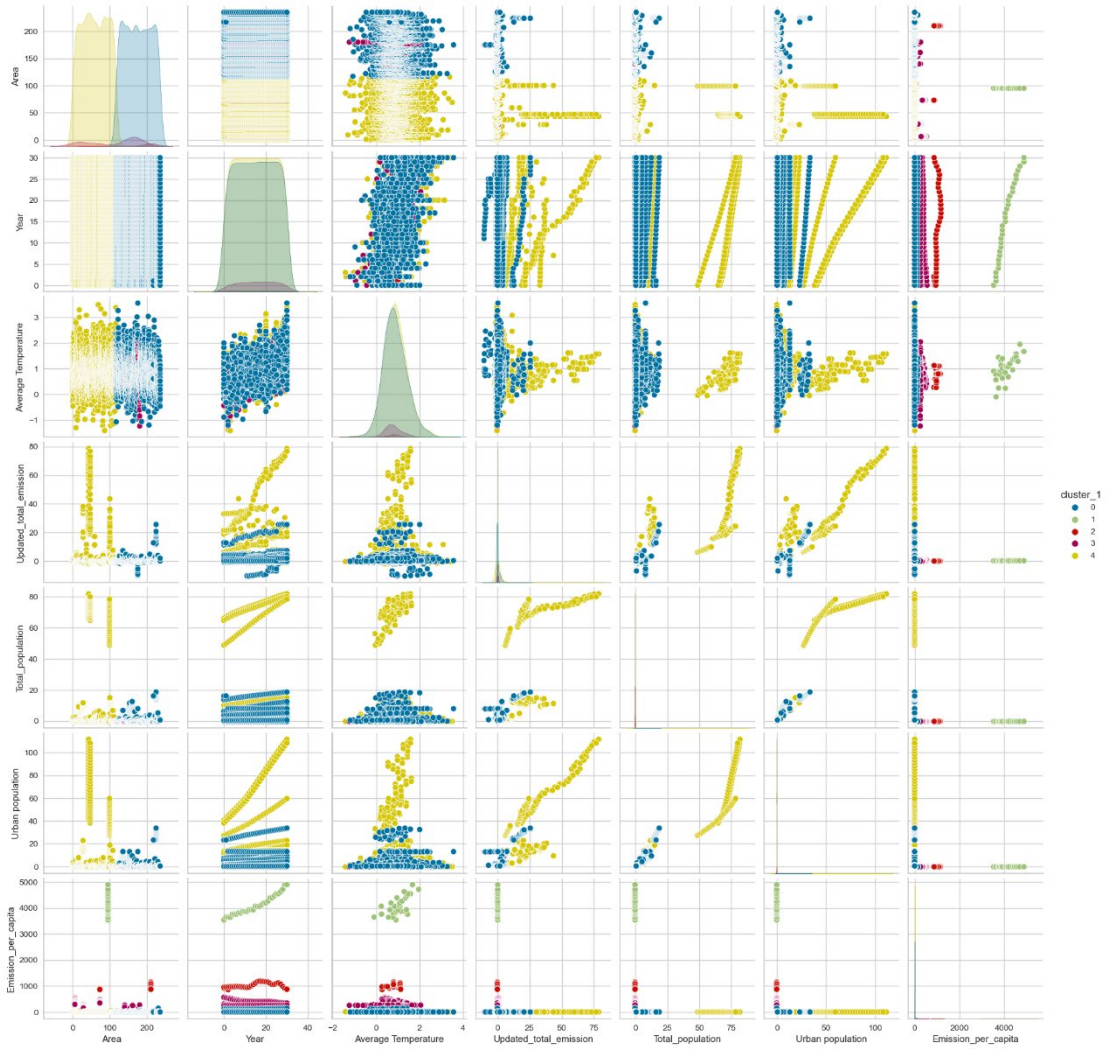
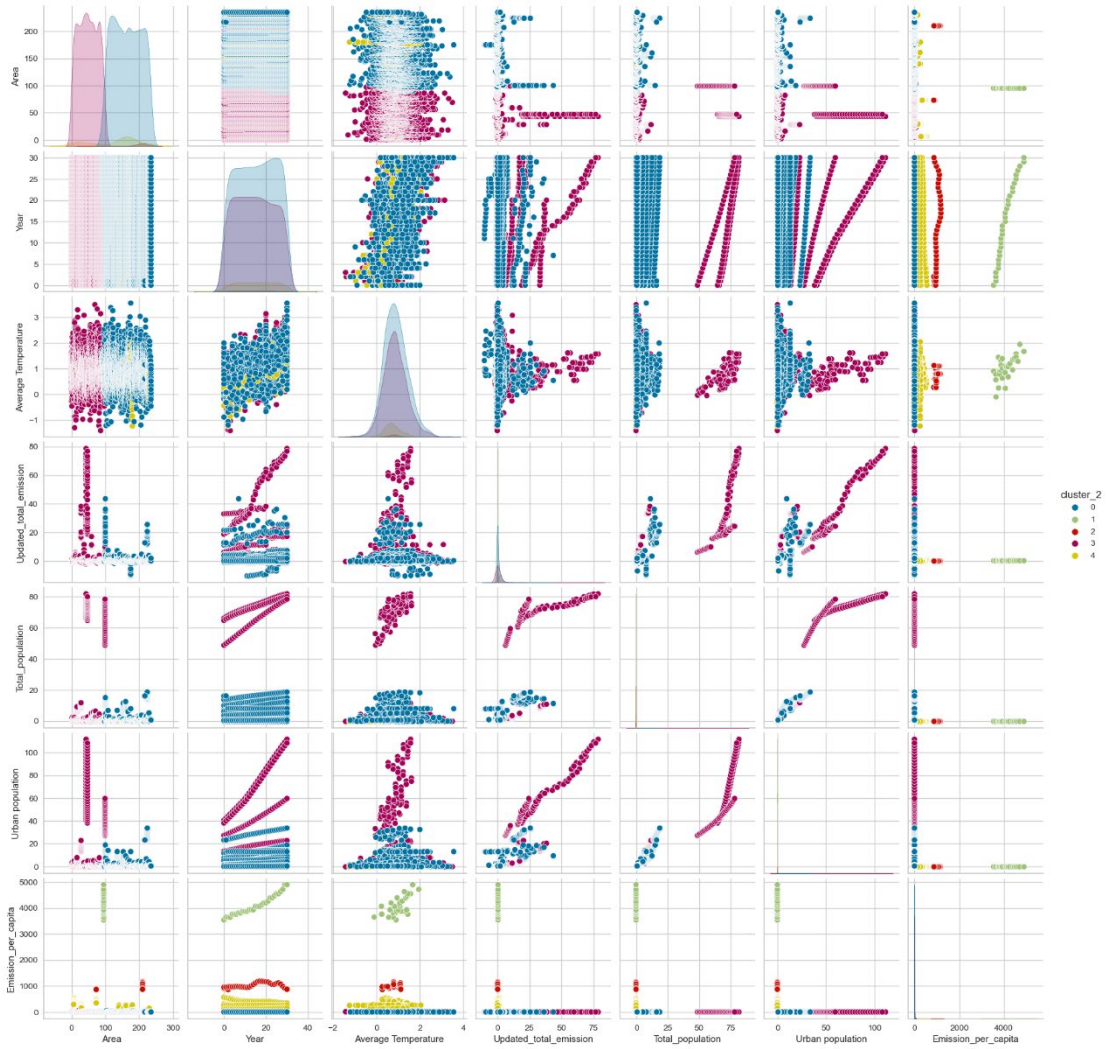


Figure 58. Agglomerative clustering result



8.2.3 The third data mining objective

The third data mining objective is to identify the countries with the highest average temperature increase and the highest total agrifood CO2 emissions in 2020 and analyse their contributions to the overall environmental impact.

Figure 59. Average Temperature Change by Top 30 countries in 2020

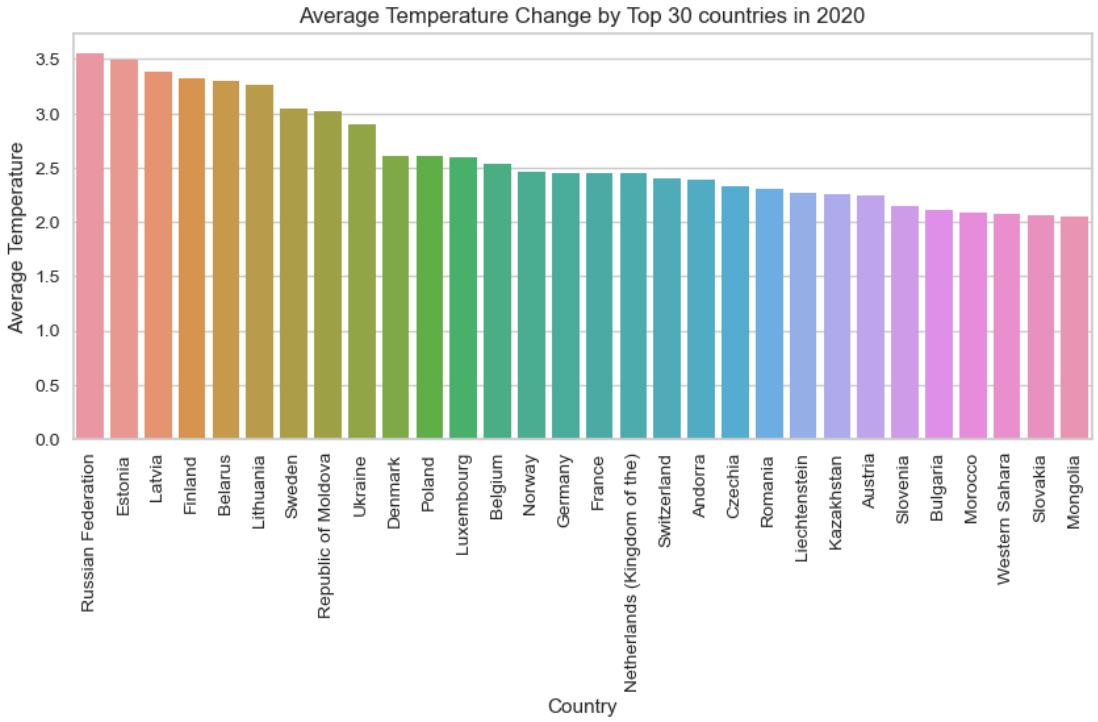


Figure 60. Total Agrifood CO2 emissions by Top 30 countries in 2020

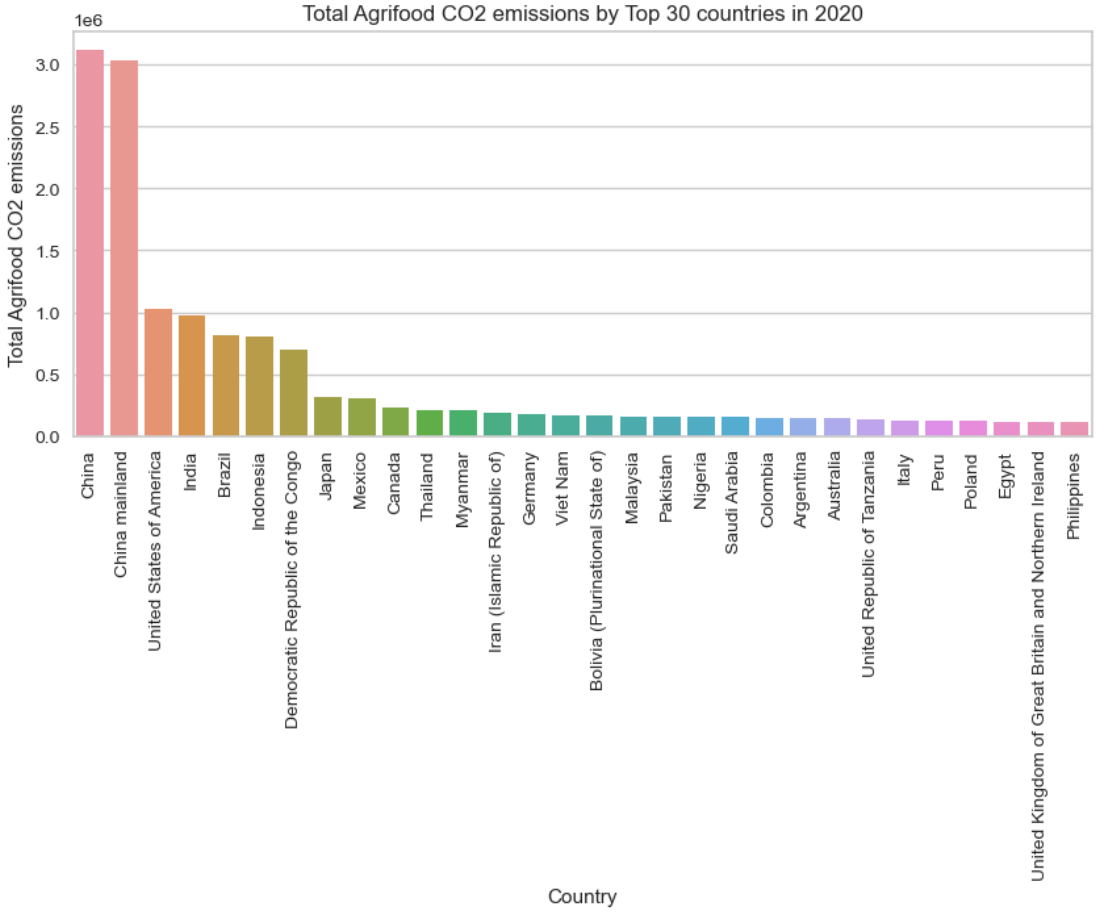
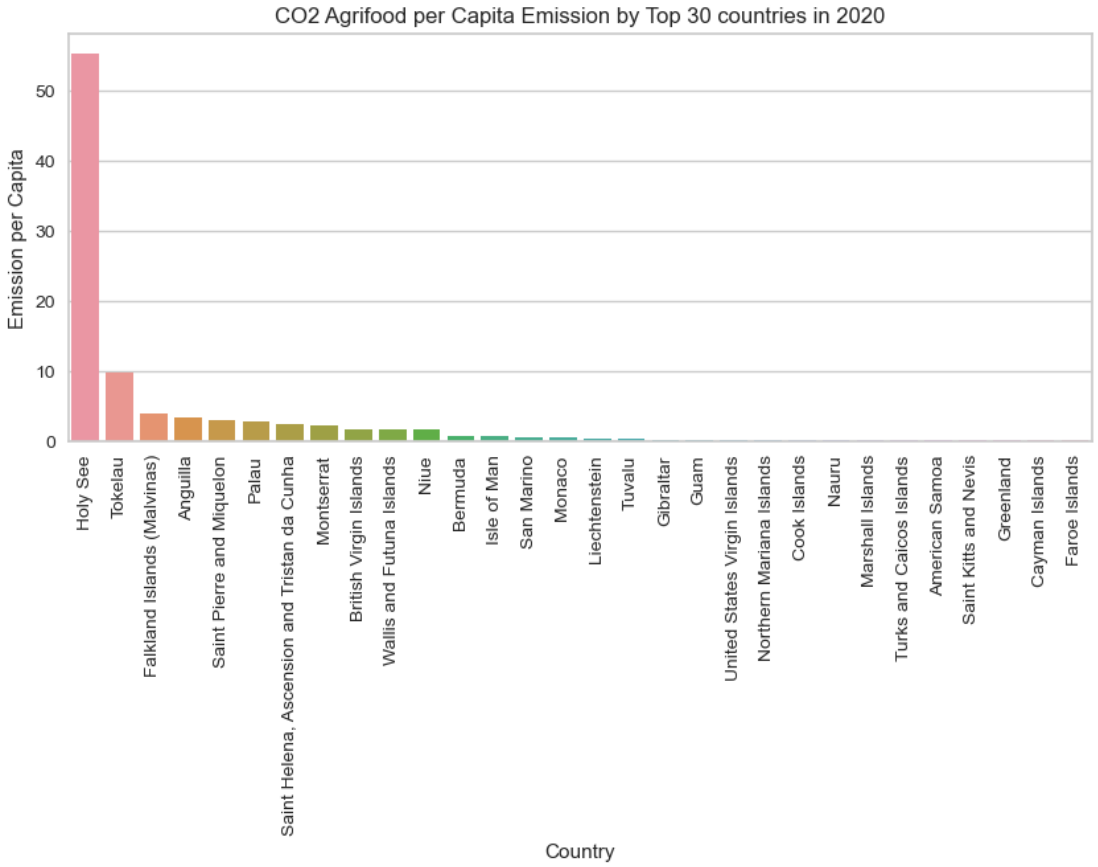


Figure 61. CO2 Agrifood per Capita Emission by Top 30 countries in 2020



8.3 Interpret the results, models, and patterns

8.3.1 The first data mining objective

The first data mining objective is to examine the correlation between CO2 emissions within the agri-food sector and the subsequent temperature rise. Through the random forest regression models, 'Food Retail' is the most important feature affecting the temperature increase. The linear regression analysis presents the linear relationship between the most crucial agri-food features and the temperature increase.

8.3.2 The second data mining objective

The second data mining objective is to anlalyse the influence of various countries based on aggregated data on emissions and temperature change. Countries with higher CO2 emissions tend to have larger populations than countries with lower CO2 emissions. The disparity in average temperature increases between countries with high CO2 emissions and low CO2 emissions is not particularly substantial.

8.3.3 The third data mining objective

The third data mining objective is to identify the countries with the highest average temperature increase and the highest total agrifood CO2 emissions in 2020 and analyse their contributions to the overall environmental impact. In 2020, the country with the highest average yearly temperature increase is Russian Federation. The average temperature increase in this country is 3.558℃, the total CO2 emissions are 34468.79 kilotons, and the CO2 emissions per capita are 0.237 tons (Figure 62). The country with the highest total agrifood CO2 emission is China. The average temperature increase in this country is 1.574℃, the total CO2 emissions are 3115114 kilotons, and the CO2 emissions per capita are 2.138 tons (Figure 63)

Figure 62. Russian Federation with the highest average temperature increase

```
agrifood_2020.loc[agrifood_2020['Area']=='Russian Federation', ['Area', 'Average Temperature', 'Updated_total_emission', 'Emission_per_capita']]
```

	Area	Average Temperature	Updated_total_emission	Emission_per_capita
5261	Russian Federation	3.558083	34468.7909	0.000237

Figure 63. China with the highest total agrifood CO2 emission

```
agrifood_2020.loc[agrifood_2020['Area']=='China', ['Area', 'Average Temperature', 'Updated_total_emission', 'Emission_per_capita']]
```

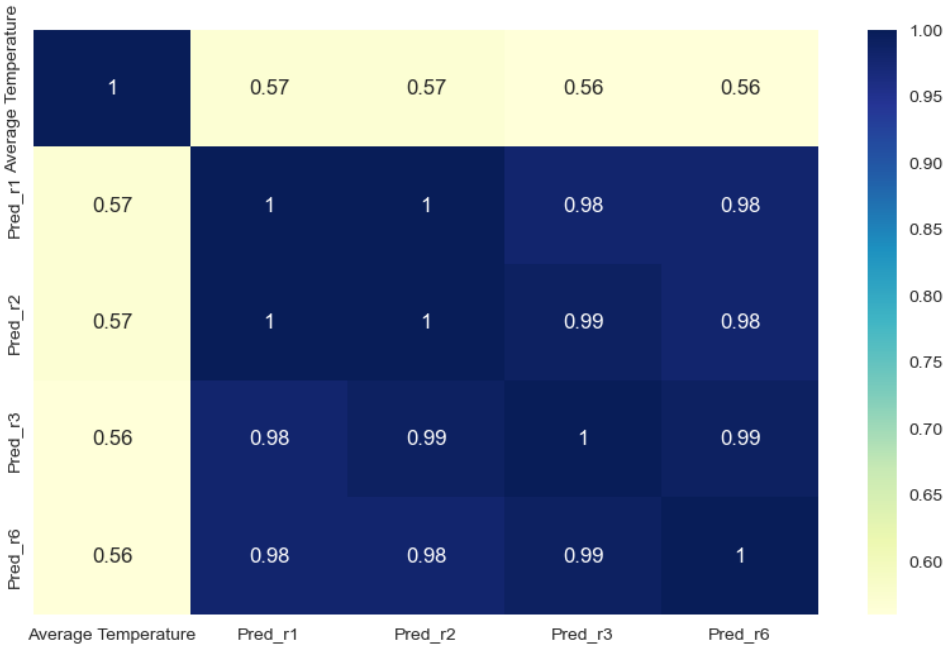
	Area	Average Temperature	Updated_total_emission	Emission_per_capita
1324	China	1.574	3.115114e+06	0.002138

8.4 Assess and evaluate results, models, and patterns

8.4.1 Regression

For the linear regression models, the best correlation is 0.57, which means the models are not quite accurate (Figure 64).

Figure 64. Linear regression models prediction results



8.4.2 Clustering

The clustering results based on 5 clusters are significant and satisfy the second data mining objective.

8.5 Iterate prior steps 1-7 as required

The primary objective of the iterative model is to enhance the predictive precision of the initial model. There are several reasons why each method can provide advantages.

Modifying the order of the polynomial: The complexity of the model can be influenced by the degree of the polynomial. Obtaining a higher degree has the potential to encompass more intricate connections; however, it also amplifies the likelihood of encountering overfitting. One can ascertain an optimal equilibrium between bias and variance by systematically varying the degrees of freedom.

Modifying the regularization parameter in Ridge Regression: The alpha parameter governs the degree of regularization intensity. Regularization is a widely employed method in machine learning to mitigate the issue of overfitting by incorporating a penalty component into the loss function. An increase in alpha leads to a higher level of regularization, thereby enhancing the model's capacity for generalization. Nevertheless, the model may exhibit underfitting if the alpha value is substantial.

The utilization of GridSearchCV for hyperparameter optimization is being discussed.

GridSearchCV is a highly effective tool that automatically identifies the most optimal hyperparameters for a given model. The algorithm operates by comprehensively exploring a predefined parameter grid and subsequently identifying the parameters that result in the highest performance.

Feature scaling is an essential consideration in the context of Ridge Regression, as this regression technique is known to be sensitive to the scale of input features. Feature scaling techniques, such as standardization, ensure all features are normalized to a comparable scale. This normalization process has been observed to enhance the system's overall performance.

Figure 65, Figure 66, and Figure 67 show the process and the result of the iterative model.

Figure 65. Iterative model process

```
from sklearn.linear_model import Ridge
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV

param_grid = {
    'polynomialfeatures__degree': [1, 2, 3],
    'ridge__alpha': [0.1, 1.0, 10.0],
}

grid = GridSearchCV(model, param_grid, cv=5)

grid.fit(X_all, y)

print('Best parameters: ', grid.best_params_)

model = grid.best_estimator_

print(model.named_steps['ridge'].intercept_, model.named_steps['ridge'].coef_)

print(model.score(X_all, y))

agrifood_dm_df1['Pred_r7'] = model.predict(X_all)
```

```
Best parameters:  {'polynomialfeatures__degree': 1, 'ridge__alpha': 10.0}
0.8729890989691312 [ 0.         -0.01548198  0.3095326   0.00625147 -0.0542614  -0.06142295
  0.06453452  0.03299416  0.04954672 -0.08247003  0.03380828  0.07693855
  0.06122727  0.15735698 -0.08139313  0.05377511 -0.01026293 -0.07393301
 -0.06706597 -0.09028087  0.00825618]
0.3238007957900825
```

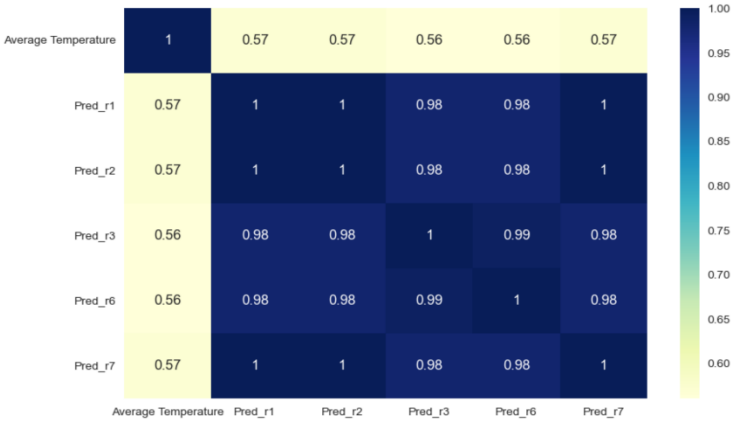
Figure 66. Iterative model result table

```
agrifood_dm_df1['Pred_r7'] = r1.predict(X_all)
agrifood_dm_df1
```

Food Transport	Net Forest conversion	Food Household Consumption	Food Retail	...	On-farm energy use	Updated_total_emission	Urban population	Total_population	Emission_per_capita	Pred_r1	Pred_r2	Pred_r3	Pred_r6	Pred_r7
-0.120307	-0.005524	-0.088036	-0.059473	...	0.119616	-0.293009	0.029327	0.326378	-0.248271	0.369836	0.379909	0.345974	0.378519	0.369836
-0.121921	-0.005524	-0.087042	-0.052769	...	0.135709	-0.289349	0.050322	0.329213	-0.247180	0.404394	0.414265	0.380475	0.413122	0.404394
-0.128618	-0.005524	-0.086844	-0.043720	...	0.102248	-0.289549	0.077928	0.403063	-0.249620	0.437938	0.447679	0.414424	0.447064	0.437938
-0.127732	-0.005524	-0.083802	-0.086336	...	0.099069	-0.289338	0.109114	0.512595	-0.252270	0.471905	0.481425	0.447355	0.480975	0.471905
-0.128049	-0.005524	-0.081156	-0.077815	...	0.116706	-0.285444	0.139586	0.594297	-0.252968	0.505505	0.514890	0.481231	0.515350	0.505505
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0.039171	1.948657	0.033819	0.258339	...	0.206113	2.141051	0.354641	0.537860	0.334230	1.170040	1.152487	1.229401	1.215839	1.170040
0.043115	1.948657	-0.000253	0.260532	...	0.190213	2.107279	0.368645	0.554653	0.314034	1.203736	1.185941	1.261633	1.249658	1.203736
0.103581	1.948657	0.005430	0.305779	...	0.246345	2.120187	0.383379	0.571596	0.305364	1.238349	1.220473	1.296198	1.283879	1.238349
0.072285	1.948657	0.017804	0.353180	...	0.228462	2.153656	0.398765	0.588616	0.301659	1.271158	1.253278	1.331912	1.318254	1.271158
0.028668	1.948657	0.007389	0.377439	...	0.228462	2.090691	0.414763	0.606346	0.276157	1.305696	1.287765	1.363800	1.352390	1.305696

Figure 67. Iterative model result diagram

```
pred_df = round(agrifood_dm_df1[['Average Temperature', 'Pred_r1','Pred_r2','Pred_r3','Pred_r6', 'Pred_r7']].corr(), 2)
pred_df
plt.figure(figsize=(10, 6))
sns.heatmap(pred_df, annot=True, cmap='YlGnBu')
plt.show()
```



Reference

2.1. Gaussian mixture models. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/modules/mixture.html

Agrifood systems. (2023). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Agrifood_systems&oldid=1153743860

Dean, J. (n.d.). *Big Data, Data Mining, and Machine Learning*.

Decision Tree Regression. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/auto_examples/tree/plot_tree_regression.html

F-Tables. (n.d.). Retrieved 19 August 2023, from http://faculty.washington.edu/heagerty/Books/Biostatistics/TABLES/F-Tables/

Imputing missing values with variants of IterativeImputer. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/auto_examples/impute/plot_iterative_imputer_variants_comparison.html

Introduction. (n.d.). [Computer software]. Retrieved 21 September 2023, from https://cran.r-project.org/web/packages/miceRanger/vignettes/miceAlgorithm.html

LavagnedOrtigue, O. (ESS). (n.d.). *Greenhouse gas emissions from agrifood systems*.

Marr, B. (n.d.). *Big Data in Practice*.

Perktold, J., Skipper Seabold, Sheppard, K., ChadFulton, Kerby Shedden, Jbrockmendel, J-Grana6, Quackenbush, P., Arel-Bundock, V., McKinney, W., Langmore, I., Baker, B., Gommers, R., Yogabonito, S-Scherrer, Zhurko, E., Brett, M., Giampieri, E., Yichuan Liu, … Halchenko, Y. (2023). *statsmodels/statsmodels: Release 0.14.0* (v0.14.0)

[Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.593847

Sklearn.cluster.AgglomerativeClustering. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

Sklearn.cluster.DBSCAN. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/modules/generated/sklearn.cluster.DBSCAN.html

Sklearn.cluster.KMeans. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/modules/generated/sklearn.cluster.KMeans.html

Sklearn.ensemble.GradientBoostingRegressor. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html

Sklearn.ensemble.RandomForestRegressor. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

Sklearn.feature_selection.mutual_info_regression. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/modules/generated/sklearn.feature_selection.mutual_info_regression.html

Sklearn.linear_model.ElasticNet. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/modules/generated/sklearn.linear_model.ElasticNet.html

Sklearn.linear_model.Lasso. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from

https://scikit-learn/stable/modules/generated/sklearn.linear_model.Lasso.html

Sklearn.linear_model.LinearRegression. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/modules/generated/sklearn.linear_model.LinearRegression.html

Sklearn.linear_model.Ridge. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/modules/generated/sklearn.linear_model.Ridge.html

Sklearn.preprocessing.PolynomialFeatures. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html

Sklearn.preprocessing.RobustScaler. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

Sklearn.svm.SVR. (n.d.). Scikit-Learn. Retrieved 21 September 2023, from https://scikit-learn/stable/modules/generated/sklearn.svm.SVR.html

Disclaimer

"I acknowledge that the submitted work is my own original work in accordance with the University of Auckland guidelines and policies on academic integrity and copyright. (See: https://www.auckland.ac.nz/en/students/forms-policies-and-guidelines/student-policies-and-guidelines/academic-integrity-copyright.html).

I also acknowledge that I have appropriate permission to use the data that I have utilised in this project. (For example, if the data belongs to an organisation and the data has not been published in the public domain then the data must be approved by the rights holder.) This includes permission to upload the data file to Canvas. The University of Auckland bears no responsibility for the student's misuse of data."