Iteration 4 - BDAS
Big Data Analytics Solutions

Student Name	Mengzhe Zhao
Student ID	219258024
Course Name	Data Mining and Big Data
Course Code	INFOSYS 722
Assignment Title	Iteration 4
Date	13th October 2023
Web Link to GitHub repository	https://github.com/MengzheZhao/aws-instance-import/tree/INFOSYS722---Iteration-4/Infosys722%20-%20Iteration%204%20(Mengzhe%20Zhao%20219258024)

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 CO_2 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 Spark library written in Python, PySpark, is form the website https://spark.apache.org/docs/latest/api/python/index.html. 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 (CO_2) 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	23rd September	All analysts	Data problems
Business understanding – Assess the situation	23rd September	All analysts	Data problems
Business understanding – Determine data mining objectives	23rd September	All analysts	Data problems
Business understanding – Produce a project plan	23rd September	All analysts	Data problems
Data understanding – Collect initial data	24th September	All analysts	Data problems, technology problems
Data understanding – Describe the data	24th September	All analysts	Data problems, technology problems
Data understanding – Explore the data	24th September	All analysts	Data problems, technology problems
Data understanding – Verify the data quality	24th September	All analysts	Data problems, technology problems
Data preparation – Select the data	25th September	Data mining consultant, database analyst	Data problems, technology problems
Data preparation – Clean the data	26th September	Data mining consultant, database analyst	Data problems, technology problems

Data preparation – Construct the data	27th September	Data mining consultant, database analyst	Data problems, technology problems
Data preparation – Integrate various data sources	28th September	Data mining consultant, database analyst	Data problems, technology problems
Data preparation – Format the data as required	29th September	Data mining consultant, database analyst	Data problems, technology problems
Data transformation – Reduce the data	30th September	Data mining consultant, database analyst	Data problems, technology problems
Data transformation – Project the data	1st October	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	2nd October	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	2nd October	Data mining consultant, database analyst	Technology problems, inability to find an adequate model
Data-mining algorithms selection – Conduct exploratory analysis and discuss	3rd October	Data mining consultant, database analyst	Technology problems, inability to find an adequate model
Data-mining algorithms selection – Select data-mining algorithms based on discussion	4th October	Data mining consultant, database analyst	Technology problems, inability to find an adequate model
Data-mining algorithms selection – Build/Select appropriate models and choose	5th October	Data mining consultant, database analyst	Technology problems, inability to find an adequate model

relevant parameters			
Data mining – Create and justify test designs	6th October	Data mining consultant, database analyst	Technology problems
Data mining – Conduct data mining: classify, regress, cluster, etc. (models must execute)	7th October	Data mining consultant, database analyst	Technology problems
Data mining – Search for patterns	8th October	Data mining consultant, database analyst	Technology problems
Interpretation – Study and discuss the mined patterns	9th October	All analysts	Inability to implement results
Interpretation – Visualize the data, results, models, and patterns	10th October	All analysts	Inability to implement results
Interpretation – Interpret the results, models, and patterns	11th October	All analysts	Inability to implement results
Interpretation – Assess and evaluate results, models, and patterns	12th October	All analysts	Inability to implement results
Interpretation – Iterate prior steps (1-7) as required	12th October	All analysts	Inability to implement results

2. Data understanding

2.1 Collect initial data

The compilation of the agricultural carbon dioxide (CO_2) 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 CO_2 emissions. CO_2 is recorded in kilotons (kt); 1 kt represents 1,000,000 kg of CO_2. 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 (CO_2)	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 disposal	Emissions from waste disposal in the agrifood system.
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.show()
```

0.4s Python

Area	Year	Savanna fires	Forest fires	Crop Residues	Rice Cultivation	Drained organic soils (CO2)	Pesticides Manufacturing	Food Transport	Forest
Afghanistan	1990	14.7237	0.0557	205.6077	686.0	0.0	11.80748296	63.1152	-2388
Afghanistan	1991	14.7237	0.0557	209.4971	678.16	0.0	11.71207309	61.2125	-2388
Afghanistan	1992	14.7237	0.0557	196.5341	686.0	0.0	11.71207309	53.317	-2388
Afghanistan	1993	14.7237	0.0557	230.8175	686.0	0.0	11.71207309	54.3617	-2388
Afghanistan	1994	14.7237	0.0557	242.0494	705.6	0.0	11.71207309	53.9874	-2388
Afghanistan	1995	14.7237	0.0557	243.8152	666.4	0.0	11.71207309	54.6445	-2388
Afghanistan	1996	38.9302	0.2014	249.0364	686.0	0.0	11.71207309	53.1637	-2388
Afghanistan	1997	30.9378	0.1193	276.294	705.6	0.0	11.71207309	52.039	-2388
Afghanistan	1998	64.1411	0.3263	287.4346	705.6	0.0	11.71207309	52.705	-2388
Afghanistan	1999	46.1683	0.0895	247.498	548.8	0.0	11.71207309	35.763	-2388
Afghanistan	2000	22.781	0.7111	168.807	509.6	0.0	11.71207309	38.556	-2388
Afghanistan	2001	0.2219	0.0	170.9884	474.32	0.0	11.71207309	39.1935	121.
Afghanistan	2002	9.0562	0.0	266.1975	529.2	0.0	11.71207309	37.5246	121.
Afghanistan	2003	55.8052	0.0	324.2195	568.4	0.0	11.71207309	60.7014	121.
Afghanistan	2004	11.9759	0.0	266.9995	764.4	0.0	11.71207309	48.7587	121.
Afghanistan	2005	5.3259	0.0	383.7498	627.2	0.0	11.98304718	73.1813	121.
Afghanistan	2006	4.4081	0.0	333.6093	627.2	0.0	12.93138916	103.2846	121.
Afghanistan	2007	2.8238	0.0	403.3749	666.4	0.0	13.42948626	114.7556	121.
Afghanistan	2008	27.7623	0.0	287.9099	744.8	0.0	29.91974111	230.5945	121.
Afghanistan	2009	2.6183	0.0	451.8647	784.0	0.0	75.0162572	385.5834	121.

only showing top 20 rows

Figure 2. Partial content of the population dataset

```
population_df.show()
```

Area	Year	Rural population	Urban population	Total Population - Male	Total Population - Female	total_emission	Average Temperature
Afghanistan	1990	9655167	2593947	5348387.0	5346409.0	2198.963539	0.536166667
Afghanistan	1991	10230490	2763167	5372959.0	5372208.0	2323.876629	0.020666667
Afghanistan	1992	10995568	2985663	6028494.0	6028939.0	2356.304229	-0.259583333
Afghanistan	1993	11858090	3237009	7003641.0	7000119.0	2368.470529	0.101916667
Afghanistan	1994	12690115	3482604	7733458.0	7722096.0	2500.768729	0.37225
Afghanistan	1995	13401971	3697570	8219467.0	8199445.0	2624.612529	0.285583333
Afghanistan	1996	13952791	3870093	8569175.0	8537421.0	2838.921329	0.036583333
Afghanistan	1997	14373573	4008032	8916862.0	8871958.0	3204.180115	0.415166667
Afghanistan	1998	14733655	4130344	9275541.0	9217591.0	3560.716661	0.890833333
Afghanistan	1999	15137497	4266179	9667811.0	9595036.0	3694.806533	1.0585
Afghanistan	2000	15657474	4436282	9815442.0	9727541.0	3113.528415	0.975666667
Afghanistan	2001	16318324	4648139	9895467.0	9793166.0	5038.533968	1.408916667
Afghanistan	2002	17086910	4893013	10562202.0	10438055.0	6035.816468	1.084166667
Afghanistan	2003	17909063	5155788	11397483.0	11247647.0	6449.089231	0.679333333
Afghanistan	2004	18692107	5426872	11862726.0	11690825.0	6734.998231	1.398833333
Afghanistan	2005	19378962	5691836	12302104.0	12109086.0	7001.297527	0.457333333
Afghanistan	2006	19961972	5931478	12828447.0	12614497.0	7076.181947	1.477333333
Afghanistan	2007	20464923	6151869	13067961.0	12835340.0	7281.053381	0.7865
Afghanistan	2008	20929119	6364912	13339006.0	13088192.0	8069.08633	0.835833333
Afghanistan	2009	21415593	6588738	13827977.0	13557331.0	8735.042447	0.897416667

only showing top 20 rows

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

```python
from pyspark.sql.types import DoubleType

for col in emission_df.columns:
    if col != 'Area':
        emission_df = emission_df.withColumn(col, emission_df[col].cast(DoubleType()))

emission_df.printSchema()
```

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

Figure 4. Attributes information of population dataset

```python
for col in population_df.columns:
    if col != 'Area':
        population_df = population_df.withColumn(col, population_df[col].cast(DoubleType()))

population_df.printSchema()
```

```
root
 |-- Area: string (nullable = true)
 |-- Year: double (nullable = true)
 |-- Rural population: double (nullable = true)
 |-- Urban population: double (nullable = true)
 |-- Total Population - Male: double (nullable = true)
 |-- Total Population - Female: double (nullable = true)
 |-- total_emission: double (nullable = true)
 |-- Average Temperature: double (nullable = true)
```

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

```python
emission_desc = emission_df.describe()
population_desc = population_df.describe()
```

Python

```python
emission_desc.show()
```

Python

```
+-------+-----------+-----------------+-----------------+-----------------+-----------------+-----------------+---------------------------+-------
|summary|       Area|             Year|    Savanna fires|     Forest fires|    Crop Residues| Rice Cultivation|Drained organic soils (CO2)|Pestici
+-------+-----------+-----------------+-----------------+-----------------+-----------------+-----------------+---------------------------+-------
|  count|       6965|             6965|             6934|             6872|             5576|             6965|                       6965|
|   mean|       NULL|2005.124910265613 8|1188.3908927603163|919.3021671420266| 998.7063092001471| 4259.666673432447|         3503.2286360373337|      3
| stddev|       NULL| 8.894665098397656| 5246.287782929853|3720.078752470731|3700.3453298519553|17613.825186797385|         15861.445677697498|      1
|    min|Afghanistan|           1990.0|              0.0|              0.0|           2.0E-4|              0.0|                        0.0|
|    max|   Zimbabwe|           2020.0|      114616.4011|       52227.6306|       33490.0741|      164915.2556|                241025.0696|
+-------+-----------+-----------------+-----------------+-----------------+-----------------+-----------------+---------------------------+-------
```

Figure 6. Statistic description of population dataset

```
population_desc.show()
```

summary	Area	Year	Rural population	Urban population	Total Population - Male	Total Population - Female	total_emission	Average Temperature
count	6965	6965	6965	6965	6965	6965	6965	6965
mean	NULL	2005.1249102656138	1.7857735393251974E7	1.693229697430007E7	1.761962962555205E7	1.732446929419813E7	64091.24414763604	0.8729890989691275
stddev	NULL	8.894665098397656	8.901521375631623E7	6.574361960972756E7	7.603993100724079E7	7.251711353615724E7	228312.95795610413	0.555929524008364
min	Afghanistan	1990.0	0.0	0.0	250.0	270.0	-391884.0563	-1.415833333
max	Zimbabwe	2020.0	9.00099113E8	9.0207776E8	7.43586579E8	7.13341908E8	3115113.748	3.558083333

Figure 7 shows the statistic description of ‘total_emission’ and ‘Average Temperature’ based on different areas.

Figure 7. Different areas statistic description

```python
result_df = emission_df.groupBy("Area").agg(*agg_exprs)

result_df.show(truncate=False)
```

Area	count	mean_total_emission	stddev_total_emission	min_total_emission	max_total_emission	mean_Average_Temperature	stdde
Chad	31	39162.270102903225	12093.626608465762	20886.97741	58155.73998	0.7723064516129031	0.394
Ethiopia PDR	3	62894.62087666666	2628.966362075241	61115.46849	65914.32071	0.3387037036666667	0.288
Micronesia (Federated States of)	30	5365.706396166668	1044.8477768898463	4457.762411	9489.934875	0.23188333326666666	0.306
Anguilla	31	12337.998410967743	773.2632286246469	12016.68806	15072.22882	0.6911290323870969	0.249
Paraguay	31	64001.78305741935	10080.5229689563	51767.34211	82247.57993	0.5834865591612903	0.476
Yemen	31	12664.442887516128	3087.4208279560016	7161.483097	16560.50374	0.8327671381290322	0.454
Senegal	31	15999.267189354845	1873.632426577216	12835.30972	19714.701	1.173569892483871	0.288
Cabo Verde	31	1792.0951166129034	247.23370493264534	1408.637099	2297.279573	1.221443469451613	0.396
Sweden	31	14978.75349564516	7815.004511977838	3079.142418	24486.25247	1.367548387064516	0.759
Tokelau	31	7255.679310838713	9.33013573236485	7251.256946	7282.33275	0.7910204610322579	0.235
Kiribati	31	4290.79752116129	1842.3919808448404	3212.758126	8869.0372	0.5172469615161291	0.387
Republic of Korea	31	78375.29503225806	17058.20312998938	31601.994	101253.6152	0.8374166666129034	0.459
Guyana	31	16717.12735	3987.0881424838713	11088.80888	23461.84553	0.9063902736774194	0.362
Eritrea	28	3862.5254971428576	1262.54876972538	2934.913891	7298.096613	0.8224477465714285	0.567
Philippines	31	85842.32444193547	14022.618018658459	63474.3205	111745.2361	0.8384274193548387	0.364
Djibouti	31	2012.7796722903224	312.72885176981845	1452.757241	2669.57649	0.9769390844838708	0.437
Tonga	31	2472.461860806452	16.08329245496489	2455.669645	2497.799404	0.6142016129032258	0.385
Malaysia	31	120219.9695983871	73974.26587456488	-52787.27562	180088.4015	0.8910376343870965	0.419
Singapore	31	15529.343654064513	5508.314222363532	6955.224834	24891.4606	0.8152035169032258	0.572
Fiji	31	2750.0464805161296	866.8540826063681	1532.790965	4163.571237	0.609327957	0.373

only showing top 20 rows

Figure 8 shows the global average temperature change and the global total emissions change by year. Figure 9 shows the distribution of the global average temperature change by year.

Figure 8. Global average temperature and total emissions change by year

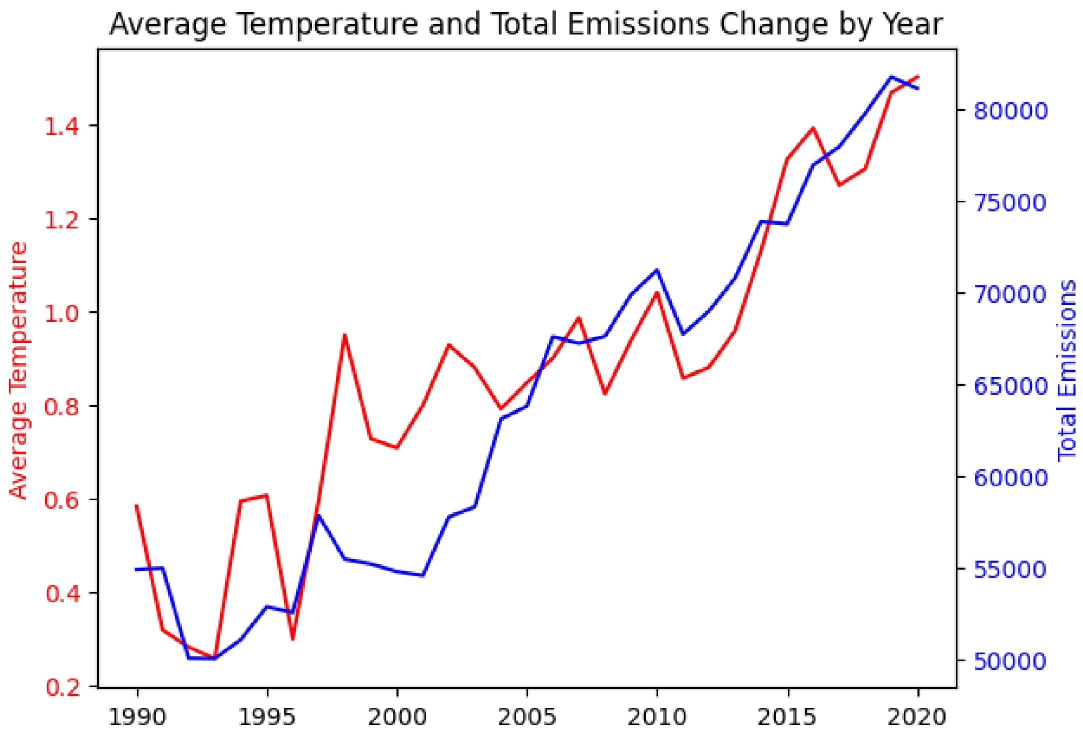


Figure 9. Global average temperature change distribution by year

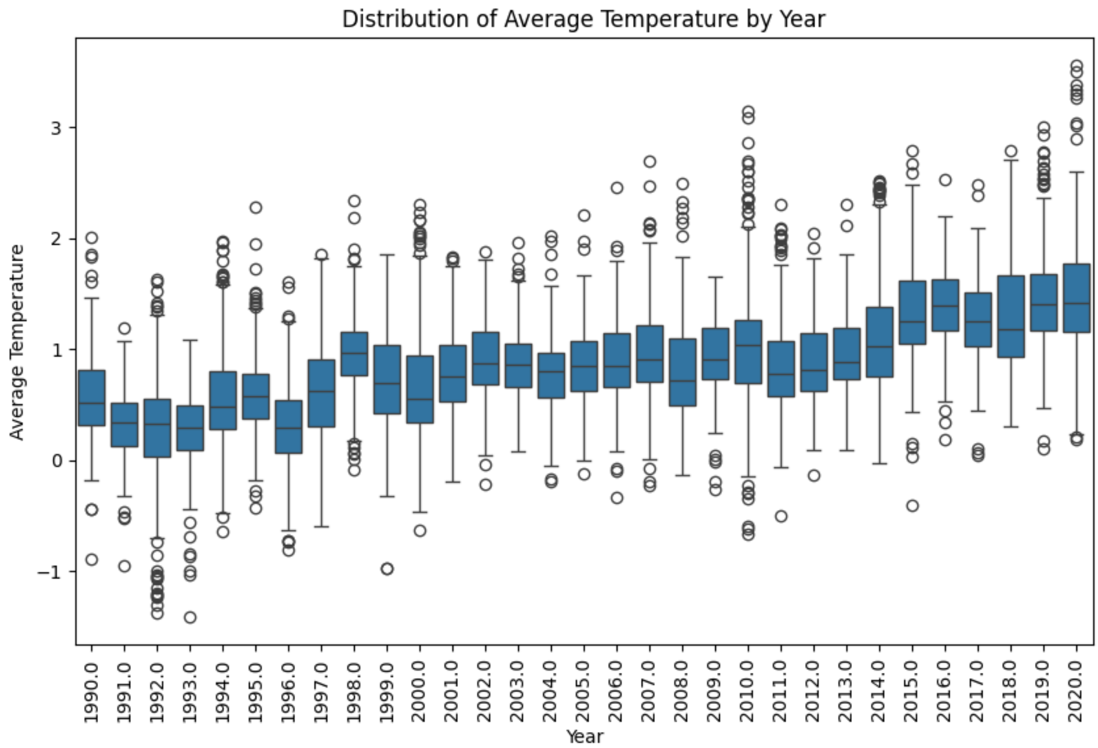


Figure 10 shows the average temperature change from 1990 to 2020 by area. Figure 11 shows the average agrifood CO2 emissions from 1990 to 2020 by top 50 areas.

Figure 10. Average temperature by area

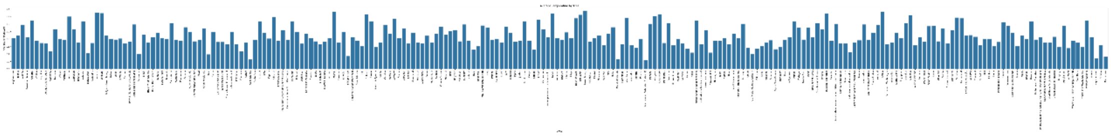
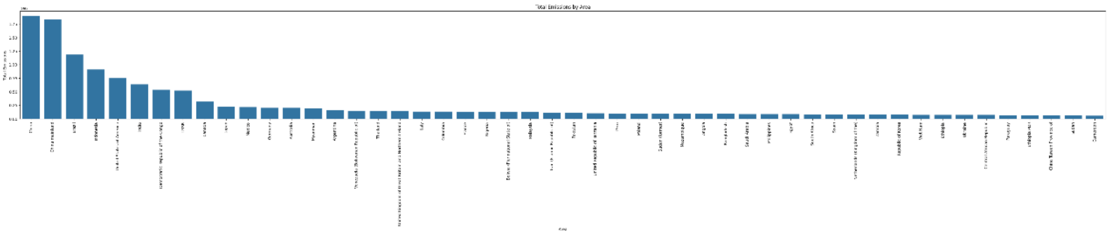


Figure 11. Average agrifood CO2 emissions by top 50 areas



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 12 shows each feature's missing values.

Figure 12. Missing values

```
missing_counts = emission_df.select([F.sum(F.when(F.col(c).isNull(), 1).otherwise(0)).alias(c) for c in emission_df.columns])

missing_counts.show()
                                                                                                                    Python

+----+----+-------------+------------+-------------+----------------+--------------------------+----------------------+--------------+----------+--
|Area|Year|Savanna fires|Forest fires|Crop Residues|Rice Cultivation|Drained organic soils (CO2)|Pesticides Manufacturing|Food Transport|Forestland|Ne
+----+----+-------------+------------+-------------+----------------+--------------------------+----------------------+--------------+----------+--
|   0|   0|           31|          93|         1389|               0|                         0|                     0|             0|       493|
+----+----+-------------+------------+-------------+----------------+--------------------------+----------------------+--------------+----------+--
```

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 13 and Figure 14 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 13. Detecting outliers

```python
from pyspark.sql.functions import col, count, when, approx_percentile

def compute_bounds(df, col_name):
    bounds = df.approxQuantile(col_name, [0.25, 0.75], 0.01)
    Q1 = bounds[0]
    Q3 = bounds[1]
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return lower_bound, upper_bound

# Define numeric columns (assuming you have the column types or you can filter out non-numeric ones programmatically)
numeric_cols = [col_name for col_name, col_type in emission_df.dtypes if col_type != 'string']

outlier_flags = {}
for column in numeric_cols:
    lower, upper = compute_bounds(emission_df, column)
    outlier_flags[column] = (emission_df[column] < lower) | (emission_df[column] > upper)

# Construct the final DataFrame indicating outliers
outliers_df = emission_df.select(*[outlier_flags[col].alias(col) for col in numeric_cols])

# Show the outlier DataFrame
outliers_df.show()
```

Figure 14. Results of detecting outliers

```
+-----+-------------+------------+-------------+----------------+---------------------------+------------------------+--------------+----------+------
| Year|Savanna fires|Forest fires|Crop Residues|Rice Cultivation|Drained organic soils (CO2)|Pesticides Manufacturing|Food Transport|Forestland|Net Fo
+-----+-------------+------------+-------------+----------------+---------------------------+------------------------+--------------+----------+------
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
|false|        false|       false|        false|           false|                      false|                   false|         false|     false|
+-----+-------------+------------+-------------+----------------+---------------------------+------------------------+--------------+----------+------
only showing top 20 rows
```

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 (CO_2) 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 12 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, Imputation estimator (*Imputer — PySpark 3.5.0 Documentation*, n.d.), is used to impute the missing values in this project (Figure 15). The imputation estimator is a method to complete missing values in a dataset. It involves using the columns' mean, median, or mode where the missing values are present. The input columns must be of numeric type. The current implementation of the Imputer needs to provide support for categorical features. Additionally, it may generate inaccurate values for categorical features (*Imputer — PySpark 3.5.0 Documentation*, n.d.). It is important to note that the mean, median, and mode values are calculated after removing any missing values. Null values in the input columns are considered missing values and are subsequently imputed. The computation of the median in PySpark utilizes the method `approxQuantile()` from the `pyspark.sql.DataFrame` module. This method is employed with a specified relative error of 0.001 (*Imputer — PySpark 3.5.0 Documentation*, n.d.).

Figure 15. Missing values impute method

```python
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']

from pyspark.ml.feature import Imputer
from pyspark.sql.functions import lit
```

```python
for col_name in cols_to_impute:
    median_value = emission_df.approxQuantile(col_name, [0.5], 0.1)[0]
    emission_df = emission_df.na.fill(median_value, [col_name])
```

Figure 16 shows the results after cleaning the missing values

Figure 16. Missing values cleaned

```
missing_counts = emission_df.select([F.sum(F.when(F.col(c).isNull(), 1).otherwise(0)).alias(c) for c in emission_df.columns])

missing_counts.show()
```

Python

```
+----+----+-------------+------------+-------------+----------------+--------------------------+------------------------+--------------+----------+--
|Area|Year|Savanna fires|Forest fires|Crop Residues|Rice Cultivation|Drained organic soils (CO2)|Pesticides Manufacturing|Food Transport|Forestland|Ne
+----+----+-------------+------------+-------------+----------------+--------------------------+------------------------+--------------+----------+--
|   0|   0|            0|           0|            0|               0|                         0|                       0|             0|         0|
+----+----+-------------+------------+-------------+----------------+--------------------------+------------------------+--------------+----------+--
```

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 17 & Figure 18).

Figure 17. New attribute – ‘Updated_total_emission’

```
from functools import reduce

cols_to_sum = ['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 = reduce(lambda a, b: a + b, (F.col(c) for c in cols_to_sum))

emission_df = emission_df.withColumn('Updated_total_emission', total_emission)
```

```
emission_df.show()
```

Figure 18. New attribute – ‘Updated_total_emission’

Manure Management	Fires in organic soils	Fires in humid tropical forests	On-farm energy use	total_emission	Average Temperature	Updated_total_emission
319.1763	0.0	0.0	47.5417	2198.963539	0.536166667	2246.5052390300007
342.3079	0.0	0.0	47.5417	2323.876629	0.020666667	2371.41832916
349.1224	0.0	0.0	47.5417	2356.304229	-0.259583333	2403.8459291600007
352.2947	0.0	0.0	47.5417	2368.470529	0.101916667	2416.0122291600005
367.6784	0.0	0.0	47.5417	2500.768729	0.37225	2548.3104291600007
397.5498	0.0	0.0	47.5417	2624.612529	0.285583333	2672.1542291600003
465.205	0.0	0.0	47.5417	2838.921329	0.036583333	2886.46302916
511.5927	0.0	0.0	47.5417	3204.180115	0.415166667	3251.72181486
541.6598	0.0	0.0	47.5417	3560.716661	0.890833333	3608.2583611600003
611.0611	0.0	0.0	47.5417	3694.806533	1.0585	3742.3482329600006
517.4928	0.0	0.0	47.5417	3113.528415	0.975666667	3161.0701148600006
426.2058	0.0	0.0	47.5417	5038.533968	1.408916667	5086.075667559999
592.5613	0.0	0.0	47.5417	6035.816468	1.084166667	6083.358167559999
603.1024	0.0	0.0	47.5417	6449.089231	0.679333333	6496.630930859999
576.0374	0.0	0.0	47.5417	6734.998231	1.398833333	6782.53993086
604.7668	0.0	0.0	47.5417	7001.297527	0.457333333	7048.839227349999
626.2428	0.0	0.0	47.5417	7076.181947	1.477333333	7123.72364693
647.4684	0.0	0.0	47.5417	7281.053381	0.7865	7328.595080729999
715.9345	0.0	0.0	47.5417	8069.08633	0.835833333	8116.628030479999
725.4414	0.0	0.0	47.5417	8735.042447	0.897416667	8782.58414657

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.join(population_df, ['Area', 'Year', 'total_emission', 'Average Temperature'])
agrifood_emission_df.show()
```

Figure 20. Merged table

n humid tropical forests	On-farm energy use	Updated_total_emission	Rural population	Urban population	Total Population - Male	Total Population - Female
0.0	47.5417	2246.5052390300007	9655167.0	2593947.0	5348387.0	5346409.0
0.0	47.5417	2371.41832916	1.023049E7	2763167.0	5372959.0	5372208.0
0.0	47.5417	2403.8459291600007	1.0995568E7	2985663.0	6028494.0	6028939.0
0.0	47.5417	2416.0122291600005	1.185809E7	3237009.0	7003641.0	7000119.0
0.0	47.5417	2548.3104291600007	1.2690115E7	3482604.0	7733458.0	7722096.0
0.0	47.5417	2672.1542291600003	1.3401971E7	3697570.0	8219467.0	8199445.0
0.0	47.5417	2886.46302916	1.3952791E7	3870093.0	8569175.0	8537421.0
0.0	47.5417	3251.72181486	1.4373573E7	4008032.0	8916862.0	8871958.0
0.0	47.5417	3608.2583611600003	1.4733655E7	4130344.0	9275541.0	9217591.0
0.0	47.5417	3742.3482329600006	1.5137497E7	4266179.0	9667811.0	9595036.0
0.0	47.5417	3161.0701148600006	1.5657474E7	4436282.0	9815442.0	9727541.0
0.0	47.5417	5086.075667559999	1.6318324E7	4648139.0	9895467.0	9793166.0
0.0	47.5417	6083.358167559999	1.708691E7	4893013.0	1.0562202E7	1.0438055E7
0.0	47.5417	6496.630930859999	1.7909063E7	5155788.0	1.1397483E7	1.1247647E7
0.0	47.5417	6782.53993086	1.8692107E7	5426872.0	1.1862726E7	1.1690825E7
0.0	47.5417	7048.839227349999	1.9378962E7	5691836.0	1.2302104E7	1.2109086E7
0.0	47.5417	7123.72364693	1.9961972E7	5931478.0	1.2828447E7	1.2614497E7
0.0	47.5417	7328.595080729999	2.0464923E7	6151869.0	1.3067961E7	1.283534E7
0.0	47.5417	8116.628030479999	2.0929119E7	6364912.0	1.3339006E7	1.3088192E7
0.0	47.5417	8782.58414657	2.1415593E7	6588738.0	1.3827977E7	1.3557331E7

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 ‘Total Population - Female’ and ‘Total Population - Male. The second is ‘Emissions_per_capita’, whose value is ‘Updated_total_emission’ divided by ‘Total_population.

Figure 21. New features – ‘Total_population’ and ‘Emissions_per_capita’

```python
agrifood_emission_df = agrifood_emission_df.withColumn('Total_population',
                                                        agrifood_emission_df['Total Population - Female'] + agrifood_emission_df['Total Population

agrifood_emission_df = agrifood_emission_df.withColumn('Emission_per_capita',
                                                        agrifood_emission_df['Updated_total_emission'] / agrifood_emission_df['Total_population'])

agrifood_emission_df.show()
```

Python

y use	Updated_total_emission	Rural population	Urban population	Total Population - Male	Total Population - Female	Total_population	Emission_per_capita
.5417	2246.5052390300007	9655167.0	2593947.0	5348387.0	5346409.0	1.0694796E7	2.100559224346121...
.5417	2371.41832916	1.023049E7	2763167.0	5372959.0	5372208.0	1.0745167E7	2.206962748145282...
.5417	2403.8459291600007	1.0995568E7	2985663.0	6028494.0	6028939.0	1.2057433E7	1.993663103216083E-4
.5417	2416.0122291600005	1.185809E7	3237009.0	7003641.0	7000119.0	1.400376E7	1.725259665375585...
.5417	2548.3104291600007	1.2690115E7	3482604.0	7733458.0	7722096.0	1.5455554E7	1.648799149587262E-4
.5417	2672.1542291600003	1.3401971E7	3697570.0	8219467.0	8199445.0	1.6418912E7	1.627485566132518...
.5417	2886.46302916	1.3952791E7	3870093.0	8569175.0	8537421.0	1.7106596E7	1.687339216498712E-4
.5417	3251.72181486	1.4373573E7	4008032.0	8916862.0	8871958.0	1.778882E7	1.827958130365027E-4
.5417	3608.2583611600003	1.4733655E7	4130344.0	9275541.0	9217591.0	1.8493132E7	1.951134270365885E-4
.5417	3742.3482329600006	1.5137497E7	4266179.0	9667811.0	9595036.0	1.9262847E7	1.942780437886466...
.5417	3161.0701148600006	1.5657474E7	4436282.0	9815442.0	9727541.0	1.9542983E7	1.617496220950507...
.5417	5086.075667559999	1.6318324E7	4648139.0	9895467.0	9793166.0	1.9688633E7	2.583254849414887...
.5417	6083.358167559999	1.708691E7	4893013.0	1.0562202E7	1.0438055E7	2.1000257E7	2.896801771311655...
.5417	6496.630930859999	1.7909063E7	5155788.0	1.1397483E7	1.1247647E7	2.264513E7	2.86888656892674E-4
.5417	6782.53993086	1.8692107E7	5426872.0	1.1862726E7	1.1690825E7	2.3553551E7	2.879625212716333E-4
.5417	7048.839227349999	1.9378962E7	5691836.0	1.2302104E7	1.2109086E7	2.441119E7	2.887544289053503...
.5417	7123.72364693	1.9961972E7	5931478.0	1.2828447E7	1.2614497E7	2.5442944E7	2.799881824575803...
.5417	7328.595080729999	2.0464923E7	6151869.0	1.3067961E7	1.283534E7	2.5903301E7	2.829212802155987E-4
.5417	8116.628030479999	2.0929119E7	6364912.0	1.3339006E7	1.3088192E7	2.6427198E7	3.071316160903626...
.5417	8782.58414657	2.1415593E7	6588738.0	1.3827977E7	1.3557331E7	2.7385308E7	3.20704231136272E-4

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

Figure 22. Checking missing values

```python
from pyspark.sql.functions import isnan, when, count, col

agrifood_emission_df.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in agrifood_emission_df.columns]).show()
```

Python

Area	Year	total_emission	Average Temperature	Savanna fires	Forest fires	Crop Residues	Rice Cultivation	Drained organic soils (CO2)	Pesticides Manufac
0	0	0	0	0	0	0	0	0	

Figure 23. Checking outliers

```python
numeric_cols = [col_name for col_name, col_type in agrifood_emission_df.dtypes if col_type != 'string']

outlier_flags = {}
for column in numeric_cols:
    lower, upper = compute_bounds(agrifood_emission_df, column)
    outlier_flags[column] = (agrifood_emission_df[column] < lower) | (agrifood_emission_df[column] > upper)

outliers_df = agrifood_emission_df.select(*[outlier_flags[col].alias(col) for col in numeric_cols])

outliers_df.show()
```

Python

```
+-----+--------------+-------------------+-------------+------------+-------------+----------------+---------------------------+--------------------
| Year|total_emission|Average Temperature|Savanna fires|Forest fires|Crop Residues|Rice Cultivation|Drained organic soils (CO2)|Pesticides Manufacturi
+-----+--------------+-------------------+-------------+------------+-------------+----------------+---------------------------+--------------------
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
|false|         false|              false|        false|       false|        false|           false|                      false|                  fal
```

4. Data transformation

4.1 Reduce the data

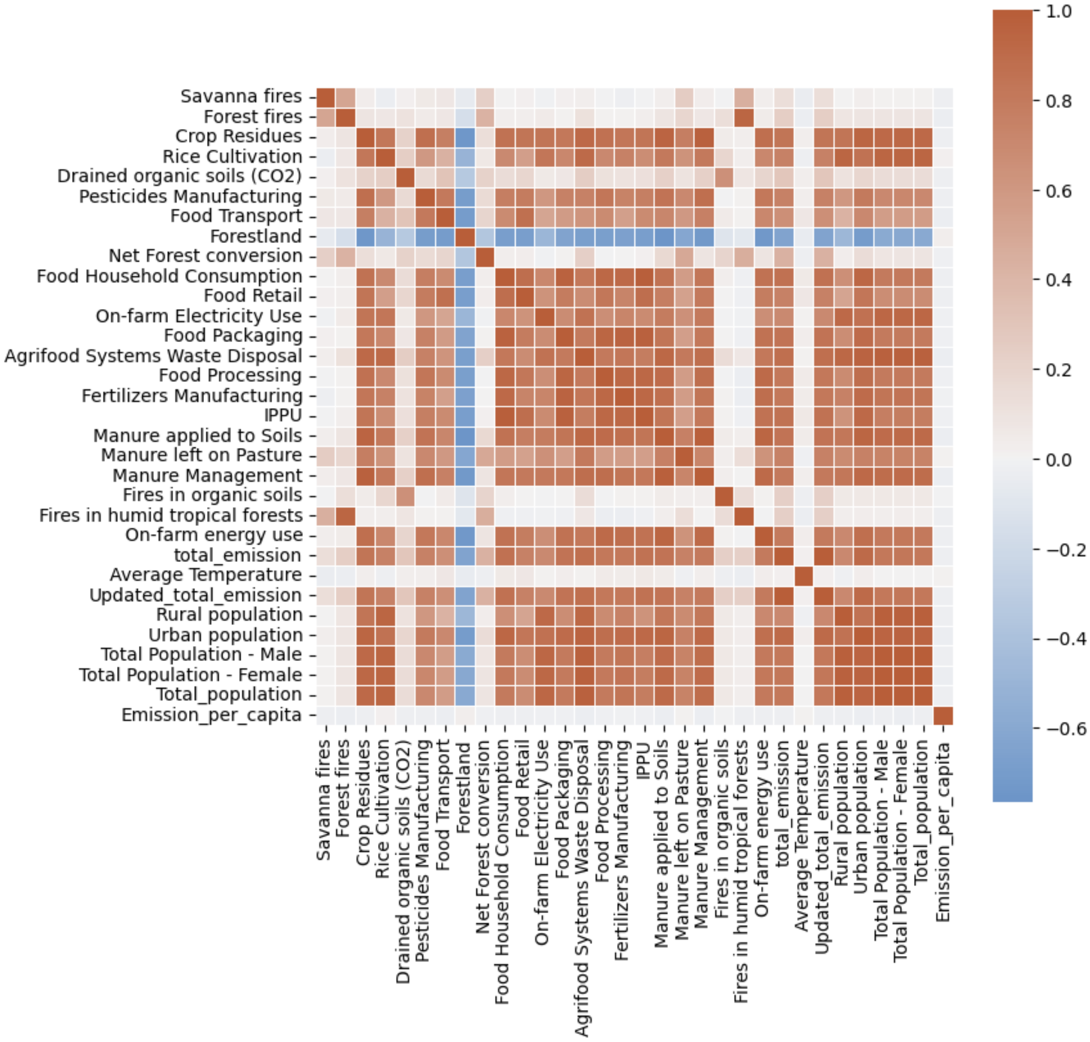
Figure 24 and Figure 25 show the correlation importance of all attributes before data transformation.

Figure 24. Codes for correlation importance of all attributes

```python
import matplotlib.pyplot as plt

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']
```

```python
pandas_df = agrifood_emission_df.select(cols).toPandas()

corr = pandas_df.corr()

plt.figure(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)

plt.show()
```

Figure 25. Correlation importance of all attributes before data transformation



Using the correlation importance of all other attributes and ‘Average Temperature’, feature selection is completed. Figure 26 shows the codes and the selection result of this method.

Figure 26. Feature selection by correlation importance

```
corr_avg_temp = corr['Average Temperature'].drop('Average Temperature')  # Drop the self-correlation
sorted_corr = corr_avg_temp.abs().sort_values(ascending=False)
print(sorted_corr)
```

```
Food Transport                     0.075724
Food Retail                        0.073404
IPPU                               0.062357
Food Household Consumption         0.055577
Food Processing                    0.053083
Forestland                         0.052053
Savanna fires                      0.046772
Manure applied to Soils            0.042311
Fertilizers Manufacturing          0.041462
Food Packaging                     0.040767
Forest fires                       0.039374
On-farm energy use                 0.039013
Fires in humid tropical forests    0.036910
Urban population                   0.036263
Manure Management                  0.032742
Drained organic soils (CO2)        0.029030
Pesticides Manufacturing           0.027960
Net Forest conversion              0.027359
Crop Residues                      0.025701
Fires in organic soils             0.023731
Rice Cultivation                   0.022532
Rural population                   0.019764
total_emission                     0.019043
Updated_total_emission             0.019041
Manure left on Pasture             0.015928
Emission_per_capita                0.012499
On-farm Electricity Use            0.009081
Agrifood Systems Waste Disposal    0.008995
Total Population - Female          0.005456
Total_population                   0.004518
Total Population - Male            0.003623
Name: Average Temperature, dtype: float64
```

‘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 'Food Transport', 'Food Retail', 'IPPU', 'Food Household Consumption', 'Food Processing', 'Forestland', 'Savanna fires', 'Manure applied to Soils', 'Fertilizers Manufacturing', 'Food Packaging', 'Forest fires', 'On-farm energy use', 'Fires in humid tropical forests', 'Urban population', 'Manure Management', 'Drained organic soils (CO2)', 'Pesticides Manufacturing', 'Net Forest conversion', 'Crop Residues', 'Fires in organic soils', 'Rice Cultivation', 'Rural population', 'Updated_total_emission', 'Manure left on Pasture', and 'Emission_per_capita'.

4.2 Project the data

In this project, StandardScaler is used to project the data. Standardising features involves removing the mean and scaling to unit variance. It is possible to utilise column summary statistics on the samples in the training set.The computation of the "unit std" involves the utilisation of the corrected sample standard deviation. This standard deviation is calculated as the square root of the unbiased sample variance (*StandardScaler — PySpark 3.5.0 Documentation*, n.d.).

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

Figure 27. Codes for StandardScaler implementation

```python
from pyspark.ml.feature import VectorAssembler

feature_columns = agrifood_emission_df2.columns
feature_columns.remove('Area')
feature_columns.remove('Year')
feature_columns.remove('Average Temperature')

assembler = VectorAssembler(inputCols=feature_columns, outputCol="features")
feature_vector = assembler.transform(agrifood_emission_df2)
```

```python
from pyspark.ml.feature import StandardScaler

scaler = StandardScaler(inputCol="features", outputCol="scaledFeatures", withStd=True, withMean=False)
scalerModel = scaler.fit(feature_vector)
agrifood_dm_df = scalerModel.transform(feature_vector)
```

Figure 28. Result after data transformation

```python
agrifood_dm_df.show()
```

```
-+--------------+-----------+--------+--------------------------+---------------+----------+-------------+-----------------------+---------------------
e|Food Transport|Food Retail|    IPPU|Food Household Consumption|Food Processing|Forestland|Savanna fires|Manure applied to Soils|Fertilizers Manufactu
-+--------------+-----------+--------+--------------------------+---------------+----------+-------------+-----------------------+---------------------
7|       63.1152|   109.6446|209.9778|                   79.0851|      252.21419| -2388.803|      14.7237|               260.1431|                   11
7|       61.2125|   116.6789|217.0388|                   80.4885|      252.21419| -2388.803|      14.7237|               268.6292|                   12.
3|        53.317|   126.1721|222.1156|                   80.7692|      252.21419| -2388.803|      14.7237|               264.7898|                   13.
7|       54.3617|    81.4607|201.2057|                   85.0678|      252.21419| -2388.803|      14.7237|               261.7221|                   14.
5|       53.9874|    90.4008|182.2905|                   88.8058|      252.21419| -2388.803|      14.7237|               267.6219|                   15.
3|       54.6445|     98.868|174.3647|                   90.1626|      252.21419| -2388.803|      14.7237|               275.2359|                   15.
3|       53.1637|    21.6458| 165.423|                   93.7905|      252.21419| -2388.803|      38.9302|               310.1306|                   16.
7|        52.039|    28.2132|164.4681|                   93.9696|    269.4764757| -2388.803|      30.9378|               338.9329|                   18.
3|        52.705|     30.887|163.5052|                   95.2597|     318.548422| -2388.803|      64.1411|               362.5683|                   19.
5|        35.763|    39.4317|163.5503|                   98.9876|    306.7799938| -2388.803|      46.1683|               400.5558|                   20.
7|        38.556|    73.7401|164.6033|                  103.4192|    269.4764757| -2388.803|       22.781|               343.3925|                   21.
7|       39.1935|   102.0122|151.6748|                  105.3407|    209.5877284|  121.9016|       0.2219|               297.2256|                   22.
7|       37.5246|    128.807|157.7543|                  108.7956|    209.5877284|  121.9016|       9.0562|               313.8126|                   44.
3|       60.7014|   157.5576|158.8417|                   110.807|    213.9037917|  121.9016|      55.8052|               320.9031|                   31.
3|       48.7587|   190.2416|160.9265|                   105.601|    213.9037917|  121.9016|      11.9759|               324.1949|                   36.
3|       73.1813|   230.9986|161.0008|                  115.7245|    226.8500141|  121.9016|       5.3259|               339.4386|                   29.
3|      103.2846|    241.914| 168.067|                  107.8508|    213.9037917|  121.9016|       4.4081|               325.8497|                   33.
5|      114.7556|   246.2698|170.1253|                  113.6165|    209.5877284|  121.9016|       2.8238|               315.0754|                   31.
3|      230.5945|   254.0779|177.1783|                  130.3898|    293.7114233|  121.9016|      27.7623|               367.3777|                   28.
7|      385.5834|   261.1027|179.2366|                  188.6719|    293.7114233|  121.9016|       2.6183|               383.5555|                   26.
-+--------------+-----------+--------+--------------------------+---------------+----------+-------------+-----------------------+---------------------
```

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 (CO_2) 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 (CO_2) 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.

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.).

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 is built. 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 shows the above model and its parameters.

Figure 29. Random forest regression model and parameters

```
# Initialize and train the model
rf = RandomForestRegressor(featuresCol="features", labelCol="Average Temperature")
model = rf.fit(train_data)

# Extract feature importances
importances = model.featureImportances

# Print feature importances
for feature, importance in zip(feature_names, importances):
    print(f"{feature}: {importance}")
```

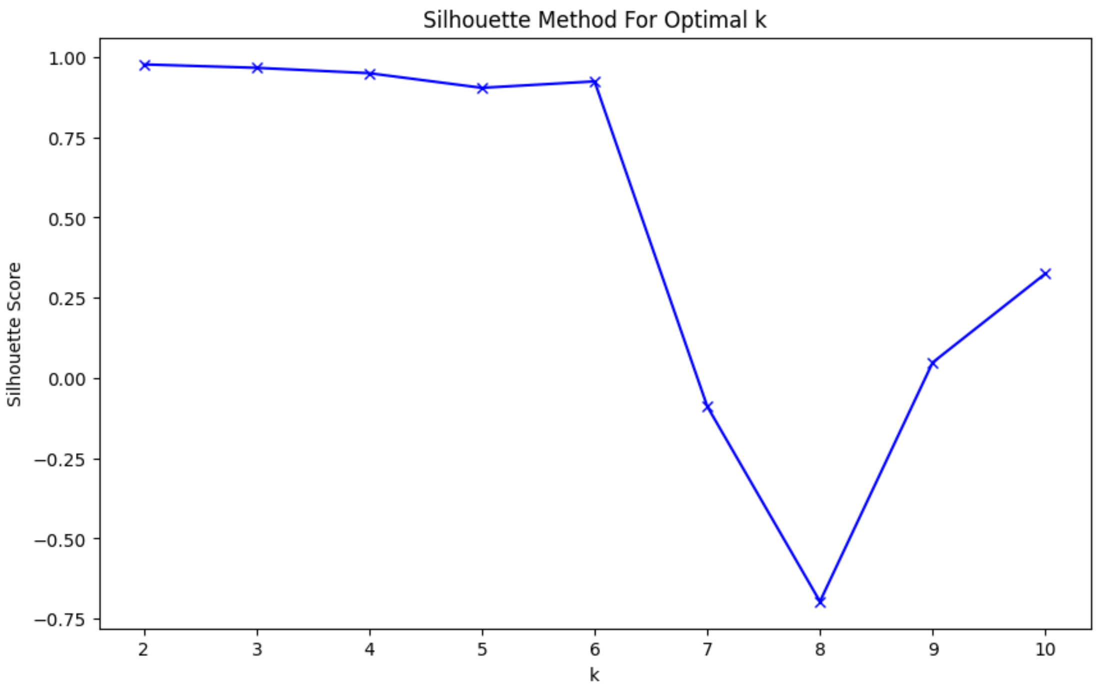
6.3.2 Clustering

For the second goal, K-means clustering model is built. The parameter n_cluster is determined as 4 (Figure 30 & Figure 31).

Figure 30. Codes for Silhouette Method

```
k_range = range(2, 11)
silhouette_list = []

for k in k_range:
    kmeans = KMeans().setK(k).setSeed(1).setFeaturesCol("scaledFeatures")
    model = kmeans.fit(agrifood_dm_df)
    predictions = model.transform(agrifood_dm_df)

    evaluator = ClusteringEvaluator()
    silhouette = evaluator.evaluate(predictions)
    silhouette_list.append(silhouette)

plt.figure(figsize=(10,6))
plt.plot(k_range, silhouette_list, 'bx-')
plt.xlabel('k')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Method For Optimal k')
plt.show()
```

Figure 31. Silhouette Method for K-Means 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 32). 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 32. Training set and testing set

```
train_data, test_data = feature_vector.randomSplit([0.8, 0.2], seed=42)

print(f"Training Dataset Count: {train_data.count()}")
print(f"Test Dataset Count: {test_data.count()}")
```

```
Training Dataset Count: 5637
Test Dataset Count: 1328
```

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

The random forest regression model runs successfully. Figure 33 shows the codes and the results of this regression models.

Figure 33. The random forest regression model

```python
# Initialize and train the model
rf = RandomForestRegressor(featuresCol="features", labelCol="Average Temperature")
model = rf.fit(train_data)

# Extract feature importances
importances = model.featureImportances

# Print feature importances
for feature, importance in zip(feature_names, importances):
    print(f"{feature}: {importance}")
```

```
Food Transport: 0.10388567771410835
Food Retail: 0.13845840664615747
IPPU: 0.09756849492802079
Food Household Consumption: 0.011650557433638426
Food Processing: 0.020468620791487244
Forestland: 0.021417711481979358
Savanna fires: 0.011982699664777264
Manure applied to Soils: 0.0486979794219924
Fertilizers Manufacturing: 0.034425302938738696
Food Packaging: 0.02245301650225138
Forest fires: 0.07432614909344701
On-farm energy use: 0.02553827976484697
Fires in humid tropical forests: 0.045750639389667405
Urban population: 0.028462724512517846
Manure Management: 0.04322087333305886
Drained organic soils (CO2): 0.020803068315075876
Pesticides Manufacturing: 0.01641888747863425
Net Forest conversion: 0.022087130281621847
Crop Residues: 0.017191962990456187
Fires in organic soils: 0.009382175480377595
Rice Cultivation: 0.04602793345740469
Rural population: 0.03014692524355831
Updated_total_emission: 0.012721547730729898
Manure left on Pasture: 0.06545908866494009
Emission_per_capita: 0.031454146740511786
```

7.2.2 Clustering

The K-Means clustering model runs successfully. Figure 34 and Figure 35 show the codes and the results of this clustering model.

Figure 34. K-Means clustering model – Clusters centers

```python
# Apply KMeans clustering with k=4
kmeans = KMeans().setK(4).setSeed(1).setFeaturesCol("scaledFeatures")
model = kmeans.fit(agrifood_dm_df)
predictions = model.transform(agrifood_dm_df)
```

```python
centers = model.clusterCenters()
print("Cluster Centers: ")
for index, center in enumerate(centers):
    print(f"Cluster {index}: {center}")
```

```
Cluster Centers: 
Cluster 0: [ 0.24013637  0.13493625  0.06729975  0.0760308   0.08901841 -0.10251315
  0.21900354  0.13633069  0.15525106  0.04512808  0.22215241  0.10389359
  0.18451477  0.13075592  0.13309576  0.20180314  0.11112109  0.10949915
  0.10601614  0.04684556  0.13143806  0.08278887  0.1597646   0.25626289
  0.12066012]
Cluster 1: [ 5.72561399e+00  9.17729055e+00  1.27021114e+01  1.23534396e+01
  1.16627595e+01 -6.81610099e+00  6.27320098e-02  9.43150958e+00
  1.22303265e+01  1.31304145e+01  1.79158060e-01  9.73302694e+00
  3.87323032e-04  1.09998558e+01  8.62350581e+00  2.43051336e-01
  8.15191267e+00  0.00000000e+00  8.87403679e+00  0.00000000e+00
  8.36225251e+00  7.47530854e+00  1.08717379e+01  6.09797212e+00
  8.40334280e-04]
Cluster 2: [ 2.65304874e+00  2.49702968e+00  2.33962302e+00  2.73084827e+00
  3.34058148e+00 -2.95480365e+00  1.11059629e-01  6.19056996e+00
  4.20988278e+00  2.59931903e+00  8.00997533e-01  5.03544182e+00
  1.97927782e-01  5.65884889e+00  6.43819637e+00  5.40501400e-01
  3.65555946e+00  5.30497190e-02  6.36480123e+00  0.00000000e+00
  7.65880468e+00  8.93150920e+00  3.71608691e+00  5.74713468e+00
  3.46153370e-04]
Cluster 3: [ 6.47322433e+00  4.57252233e+00  2.32879609e+00  2.49450454e+00
  2.39853583e+00 -5.51385915e+00  1.16524423e+00  3.73413211e+00
  1.23387831e+00  1.46954419e+00  2.21933801e+00  3.01914679e+00
  2.21111419e+00  2.88639058e+00  3.99568881e+00  1.93146940e+00
  5.75731803e+00  6.54345288e+00  4.39849432e+00  8.28468062e-01
  5.98757081e-01  5.44309701e-01  4.27296227e+00  6.08687559e+00
  2.15363871e-03]
```

Figure 35. K-Means clustering model – Clusters counts

```python
cluster_assignments = predictions.groupBy("prediction").count().orderBy("prediction")
cluster_assignments.show()
```

```
+----------+-----+
|prediction|count|
+----------+-----+
|         0| 6806|
|         1|   34|
|         2|   61|
|         3|   64|
+----------+-----+
```

7.3 Search for patterns

7.3.1 Regression

The random forest regression model pattern interprets that the most critical agri-food factor relative to the average temperature rise is food retail.

Figure 36. Random forest regression model pattern

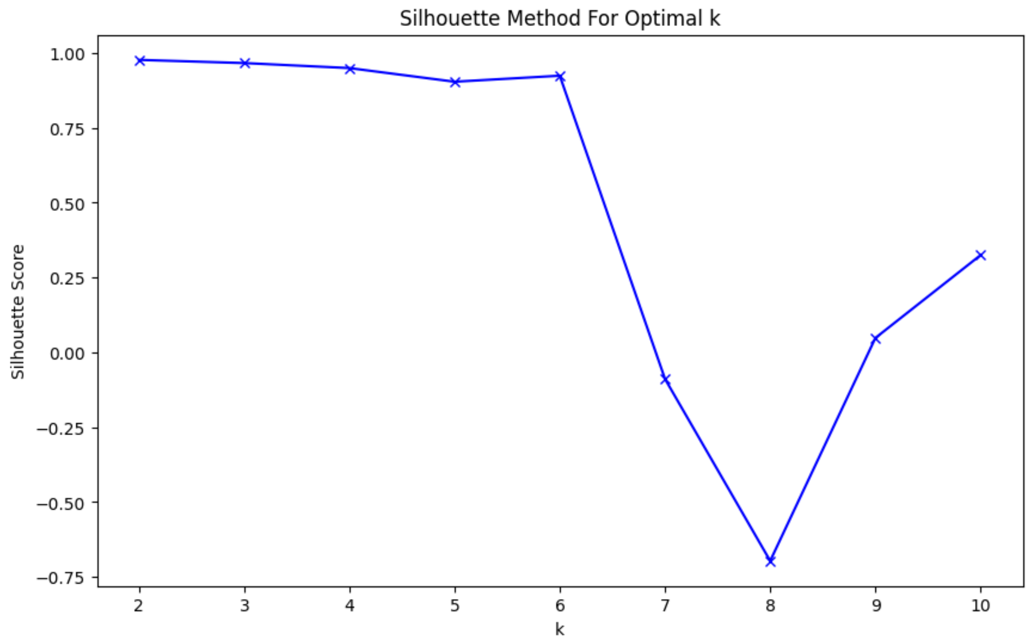
```
importances = model.featureImportances.toArray()

paired = list(zip(feature_names, importances))

sorted_features = sorted(paired, key=lambda x: x[1], reverse=True)

for feature, importance in sorted_features:
    print(f"{feature}: {importance}")
```

```
Food Retail: 0.13845840664615747
Food Transport: 0.10388567771410835
IPPU: 0.09756849492802079
Forest fires: 0.07432614909344701
Manure left on Pasture: 0.06545908866494009
Manure applied to Soils: 0.0486979794219924
Rice Cultivation: 0.04602793345740469
Fires in humid tropical forests: 0.045750639389667405
Manure Management: 0.04322087333305886
Fertilizers Manufacturing: 0.034425302938738696
Emission_per_capita: 0.031454146740511786
Rural population: 0.03014692524355831
Urban population: 0.028462724512517846
On-farm energy use: 0.02553827976484697
Food Packaging: 0.02245301650225138
Net Forest conversion: 0.022087130281621847
Forestland: 0.021417711481979358
Drained organic soils (CO2): 0.020803068315075876
Food Processing: 0.020468620791487244
Crop Residues: 0.017191962990456187
Pesticides Manufacturing: 0.01641888747863425
Updated_total_emission: 0.012721547730729898
Savanna fires: 0.011982699664777264
Food Household Consumption: 0.011650557433638426
Fires in organic soils: 0.009382175480377595
```

7.3.2 Clustering

Figure 37 shows Silhouette Score for K-Means Clustering. There are 4 clusters, and the results will be presented in the section 8.

Figure 37. Silhouette Score for K-Means 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', 'Food Transport', 'IPPU', 'Forest fires', 'Manure left on Pasture', 'Manure applied to Soils', 'Rice Cultivation', 'Fires in humid tropical forests', 'Manure Management', and 'Fertilizers Manufacturing' (Figure 38).

Figure 38. Important features based on random forest regression model

```
Food Retail: 0.13845840664615747
Food Transport: 0.10388567771410835
IPPU: 0.09756849492802079
Forest fires: 0.07432614909344701
Manure left on Pasture: 0.06545908866494009
Manure applied to Soils: 0.0486979794219924
Rice Cultivation: 0.04602793345740469
Fires in humid tropical forests: 0.045750639389667405
Manure Management: 0.04322087333305886
Fertilizers Manufacturing: 0.034425302938738696
Emission_per_capita: 0.031454146740511786
Rural population: 0.03014692524355831
Urban population: 0.028462724512517846
On-farm energy use: 0.02553827976484697
Food Packaging: 0.02245301650225138
Net Forest conversion: 0.022087130281621847
Forestland: 0.021417711481979358
Drained organic soils (CO2): 0.020803068315075876
Food Processing: 0.020468620791487244
Crop Residues: 0.017191962990456187
Pesticides Manufacturing: 0.01641888747863425
Updated_total_emission: 0.012721547730729898
Savanna fires: 0.011982699664777264
Food Household Consumption: 0.011650557433638426
Fires in organic soils: 0.009382175480377595
```

8.1.2 Clustering

The high CO_2 emission countries have higher average temperature than the low CO_2 emission countries (Figure 39). The difference in emission per capita between the high CO_2 emission countries and the low CO_2 emission countries is not quite significant (Figure 40). The high food retail CO_2 emission countries have higher average temperature than the low food retail CO_2 emission countries (Figure 41).

Figure 39. Average temperature and total CO2 emissions of K-Means Clustering

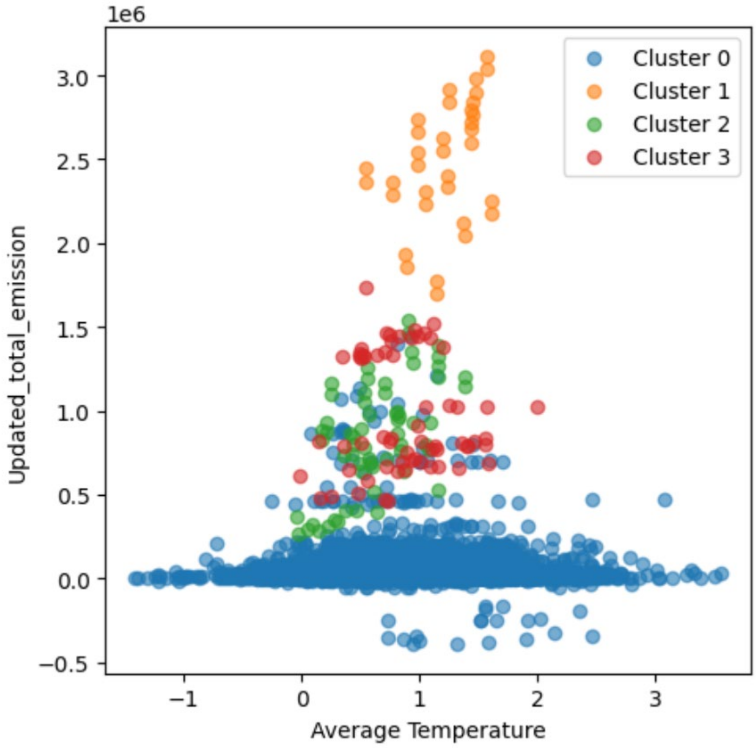


Figure 40. Average temperature and emission per capita of K-Means Clustering

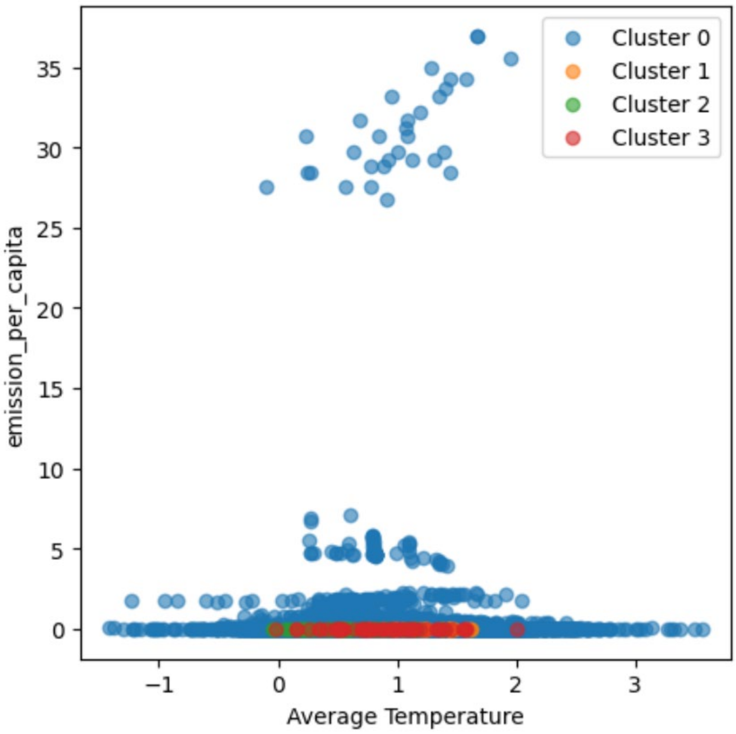
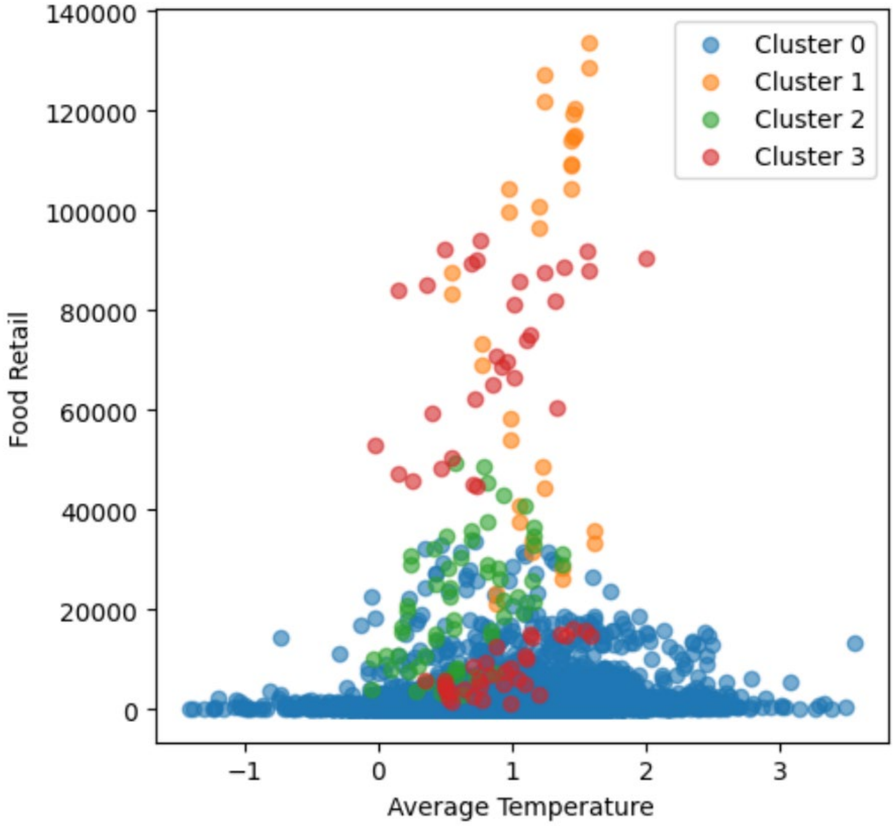


Figure 41. Average temperature and Food Retail emission of K-Means Clustering

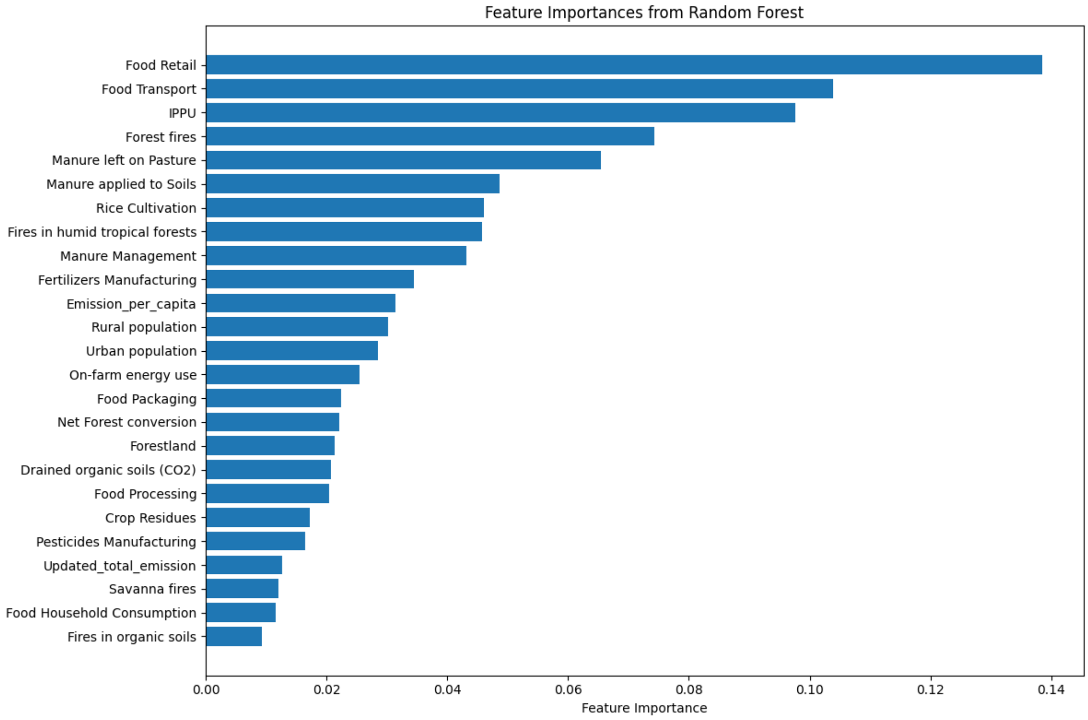


8.2 Visualize the data, results, models, and patterns

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 42. Feature Importance based on random forest 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 43. K-Means Clustering result

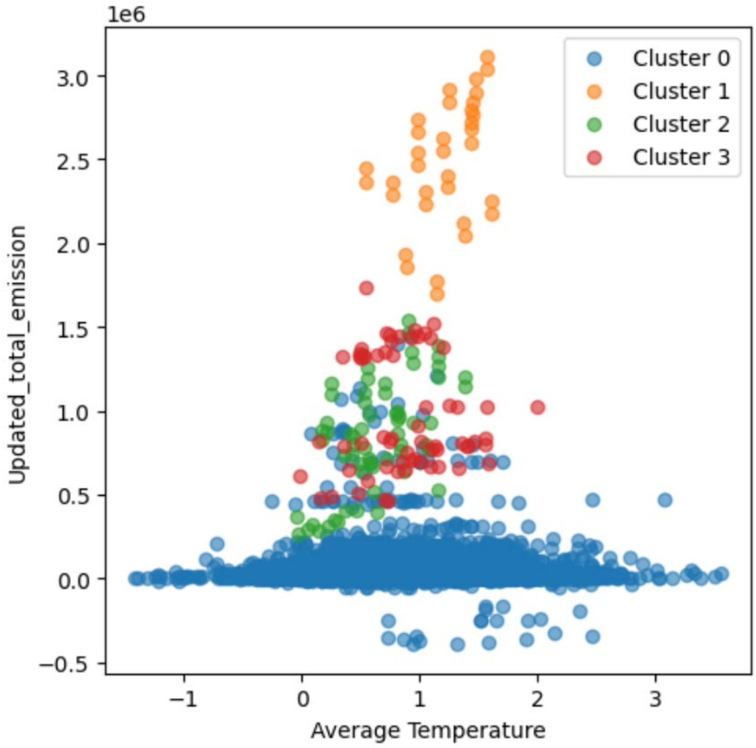


Figure 44. K-Means Clustering result

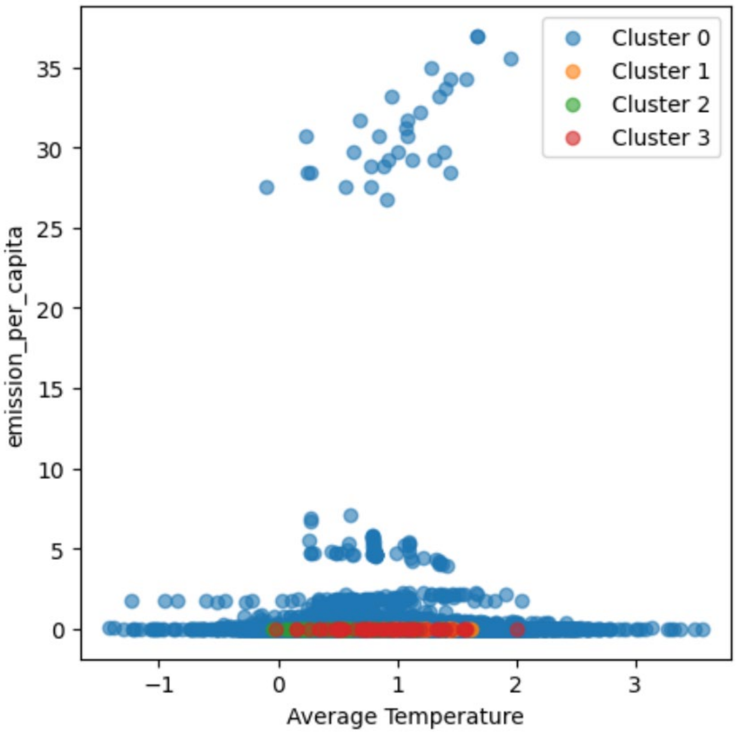
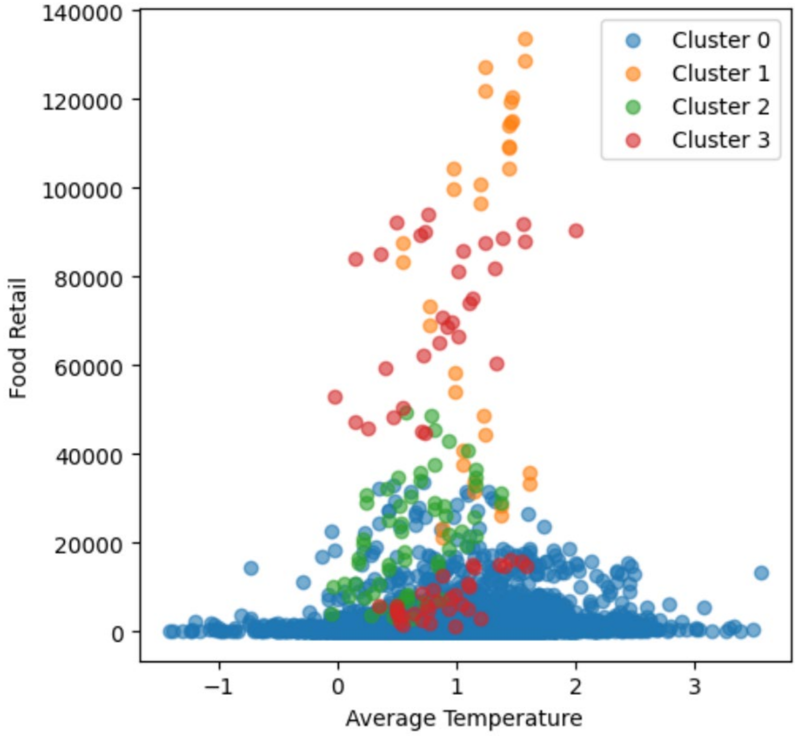


Figure 45. K-Means Clustering result



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 CO_2 emissions in 2020 and analyse their contributions to the overall environmental impact.

Figure 46. Average Temperature Change by Top 30 countries in 2020

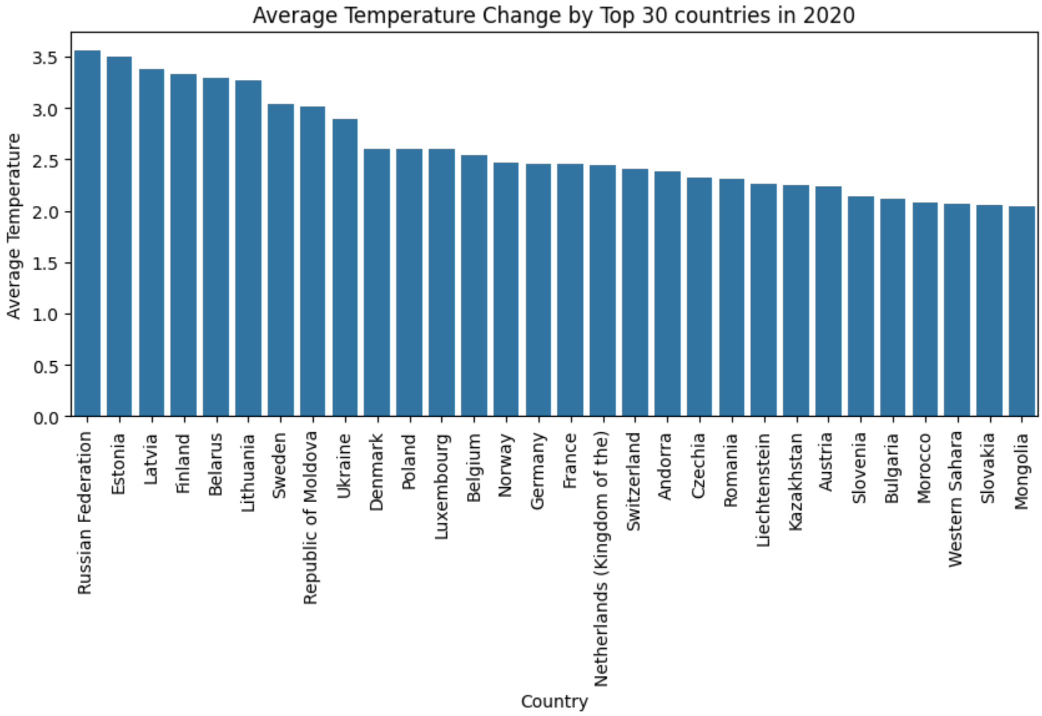


Figure 47. Total Agrifood CO2 emissions by Top 30 countries in 2020

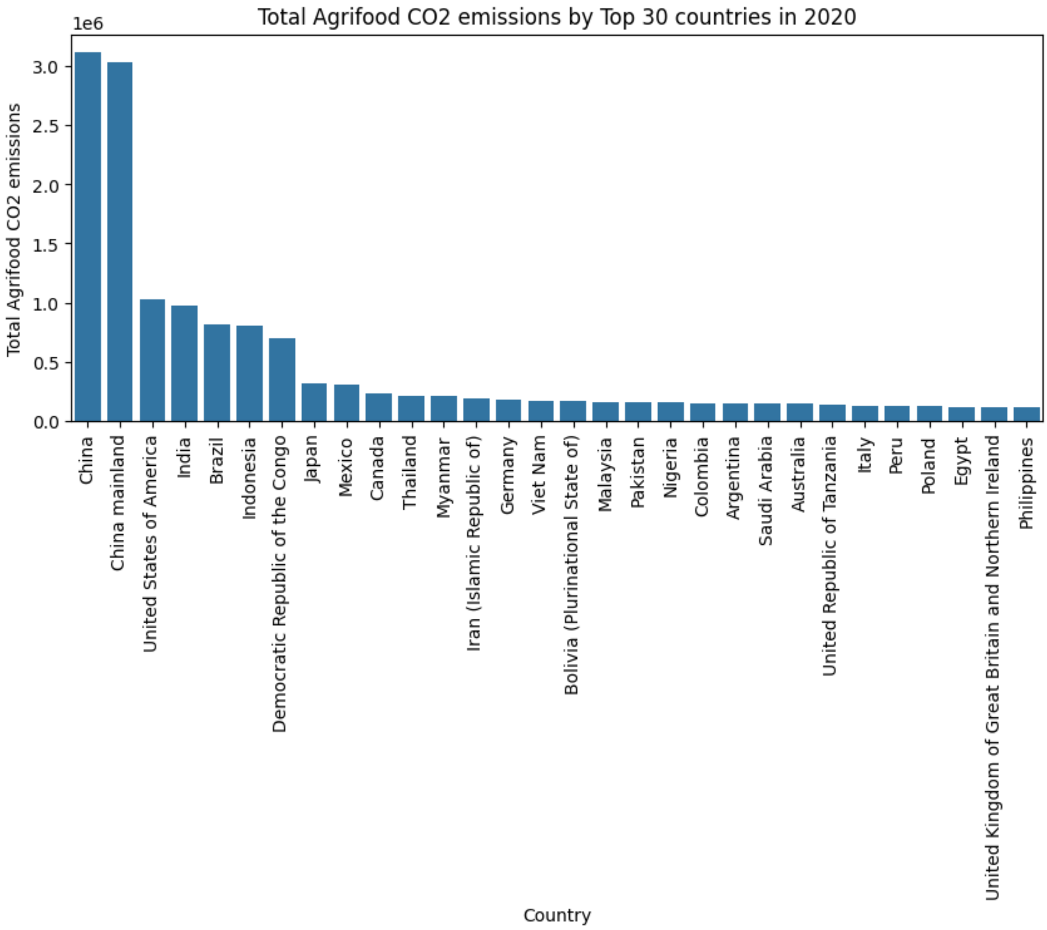
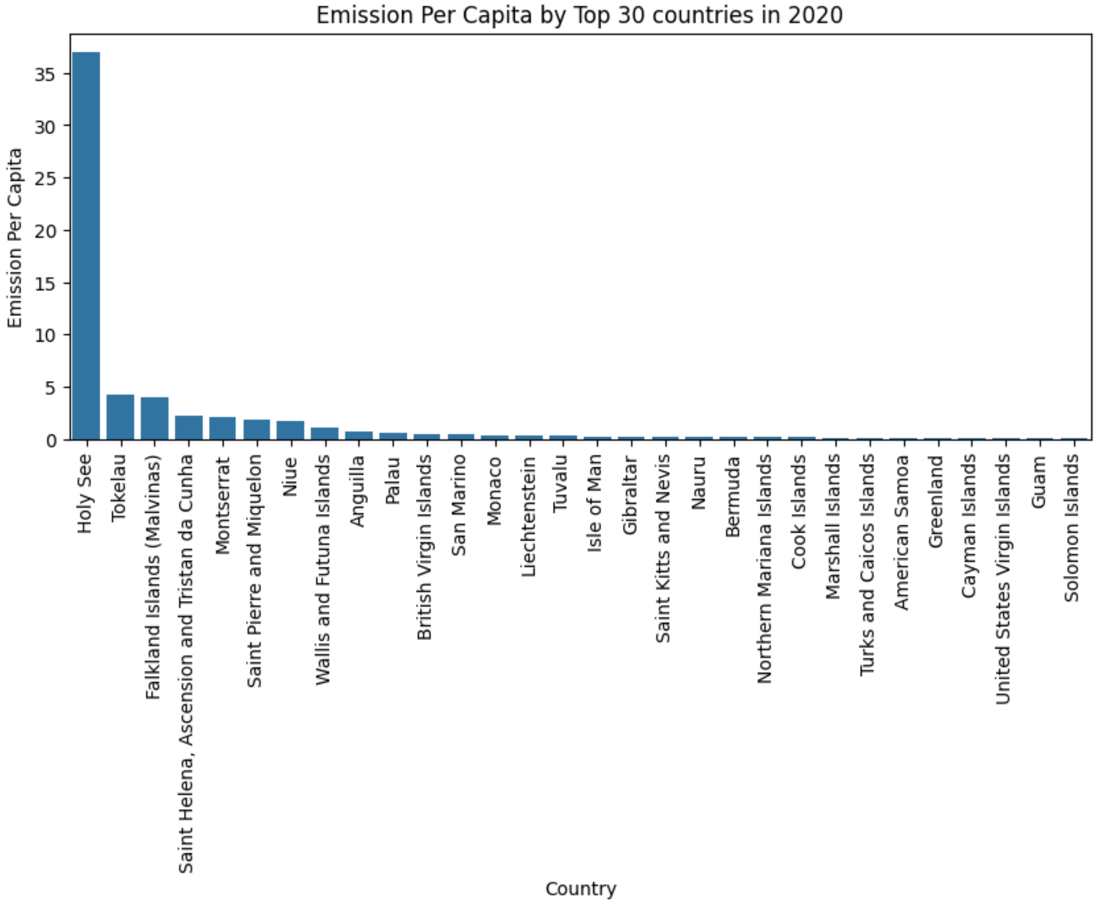


Figure 48. CO2 Agrifood per Capita Emission by Top 30 countries in 2020



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.

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. The high CO2 emission countries have higher average temperature than the low CO2 emission countries. The difference in emission per capita between the high CO2 emission countries and the low CO2 emission countries is not quite significant. The high food retail CO2 emission countries have higher average temperature than the low food retail CO2 emission countries.

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 34468.791 kilotons, and the CO2 emissions per capita are 2.367 tons (Figure 49). 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 3115113.749 kilotons, and the CO2 emissions per capita are 0.00214 tons (Figure 50)

Figure 49. Russian Federation with the highest average temperature increase

```
russian_federation_data = agrifood_2020.filter(agrifood_2020['Area'] == 'Russian Federation') \
                                    .select('Area', 'Average Temperature', 'Updated_total_emission', 'Emission_per_capita')

russian_federation_data.show()
```

```
+------------------+-------------------+----------------------+--------------------+
|              Area|Average Temperature|Updated_total_emission| Emission_per_capita|
+------------------+-------------------+----------------------+--------------------+
|Russian Federation|        3.558083333|    34468.790900000015|2.367080303794615...|
+------------------+-------------------+----------------------+--------------------+
```

Figure 50. China with the highest total agrifood CO2 emission

```
china_data = agrifood_2020.filter(agrifood_2020['Area'] == 'China') \
                                    .select('Area', 'Average Temperature', 'Updated_total_emission', 'Emission_per_capita')

china_data.show()
```

```
+-----+-------------------+----------------------+--------------------+
| Area|Average Temperature|Updated_total_emission| Emission_per_capita|
+-----+-------------------+----------------------+--------------------+
|China|              1.574|          3115113.7488|0.002138137716844574|
+-----+-------------------+----------------------+--------------------+
```

8.4 Assess and evaluate results, models, and patterns

8.4.1 Regression

For the random forest regression model, Root Mean Squared Error (RMSE) on the test dataset is 0.498 (Figure 51). A sample's RMSD is calculated as the square root of the mean of the squared differences between the observed and predicted values (‘Root-Mean-Square Deviation’, 2023). The deviations observed during calculations over the data sample used for estimation are commonly referred to as residuals. However, when these calculations are performed out-of-sample, the deviations are referred to as errors or prediction errors. The RMSD is a metric used to quantify the overall accuracy of predictions by combining the errors of multiple data points into a single measure. The RMSD is a measurement of accuracy. It facilitates the comparison of forecasting errors among various models for a specific dataset rather than across different datasets. It is significant to note that the RMSD is scale-dependent, meaning that it depends on the size of the data (‘Root-Mean-Square Deviation’, 2023).

Figure 51. Random regression models evaluation results

```
evaluator = RegressionEvaluator(labelCol="Average Temperature", predictionCol="prediction", metricName="rmse")

rmse = evaluator.evaluate(test_predictions)
print(f"Root Mean Squared Error (RMSE) on test data: {rmse}")
```

```
Root Mean Squared Error (RMSE) on test data: 0.498492998379684
```

8.4.2 Clustering

The clustering results based on 4 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. This is the reason why grid search by cross validation method can provide advantage.

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.

Figure 52, Figure 53, and Figure 54 show the process and the result of the iterative model.

Figure 52. Iterative model process

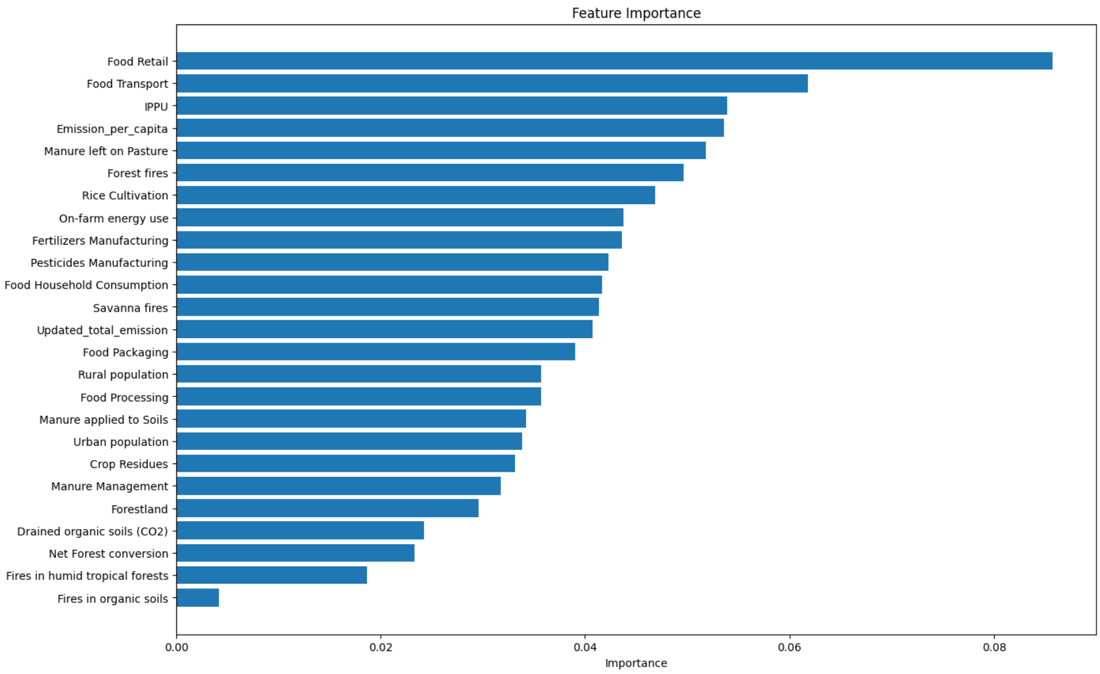
```
# Set up the parameter grid
paramGrid = (ParamGridBuilder()
             .addGrid(rf.numTrees, [10, 20, 30])  # Number of trees
             .addGrid(rf.maxDepth, [5, 10, 20])   # Maximum depth of each tree
             .build())

# Set up 5-fold cross validation
crossval = CrossValidator(estimator=rf,
                          estimatorParamMaps=paramGrid,
                          evaluator=RegressionEvaluator(labelCol="Average Temperature"),
                          numFolds=5)

cvModel = crossval.fit(feature_vector)
bestModel = cvModel.bestModel
```

Figure 53. Iterative model result

```python
best_importances = bestModel.featureImportances
paired = list(zip(feature_names, best_importances))
sorted_features = sorted(paired, key=lambda x: x[1], reverse=True)

for feature, importance in sorted_features:
    print(f"{feature}: {importance}")
```

```
Food Retail: 0.08573924600041849
Food Transport: 0.061773344479291316
IPPU: 0.05387587730418323
Emission_per_capita: 0.053609272533391476
Manure left on Pasture: 0.05180171826414463
Forest fires: 0.04963162841698903
Rice Cultivation: 0.04682953853135529
On-farm energy use: 0.04377633312460344
Fertilizers Manufacturing: 0.0436019124992591
Pesticides Manufacturing: 0.04227081112285545
Food Household Consumption: 0.04164271903181559
Savanna fires: 0.04133109872526139
Updated_total_emission: 0.040755094342541265
Food Packaging: 0.039045371911786544
Rural population: 0.03569253644943484
Food Processing: 0.035686565486168845
Manure applied to Soils: 0.0342668303004297
Urban population: 0.03381918596856676
Crop Residues: 0.033171198267981455
Manure Management: 0.03173731551007057
Forestland: 0.02961681185319776
Drained organic soils (CO2): 0.02426136637389842
Net Forest conversion: 0.023276478049852917
Fires in humid tropical forests: 0.01862924621610925
Fires in organic soils: 0.004158499236393073
```

Figure 54. Iterative model result diagram



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

Imputer—PySpark 3.5.0 documentation. (n.d.). Retrieved 12 October 2023, from https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.feature.Imputer.html

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

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

Root-mean-square deviation. (2023). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Root-mean-square_deviation&oldid=1179174918

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

StandardScaler—PySpark 3.5.0 documentation. (n.d.). Retrieved 12 October 2023, from https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.feature.StandardScaler.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."