Mohammed V University
Higher School of Computer Science
and System Analysis - ENSIAS

Data Driven Decision Making Project

Software Engineering Class (GL)

Forecasting Demand for Moroccan Local Products in National and International Markets: A Data-Driven Approach

Authors :

BELZEDI Ghizlane

DAOUDI Halima

Supervisor :

Prof. TABII Youness

Academic Year : 2023-2024

Acknowledgement

We would like to express our sincere gratitude to Mr. Younes Tabii for his invaluable guidance, support, and encouragement throughout this project. His expertise, dedication, and mentorship have been instrumental in shaping the direction and success of this endeavor. We are truly grateful for the opportunity to learn from him and for his unwavering commitment to our academic and professional development. Thank you, Mr. Tabii, for your tireless efforts and unwavering support.

Abstract

This document provides a comprehensive overview of a project focusing on forecasting the demand for Moroccan local products in both national and international markets. The objective is to anticipate market demand and formulate effective strategies for product development and commercialization. These strategies will be data-driven, based on insights collected from official sources. A crucial aspect of the project involves data cleaning to identify relevant products and characteristics for strategy development. The selection of algorithms for demand forecasting will be justified, and data visualization will play a key role in presenting the results.

Résumé

Ce document offre un aperçu complet de notre projet axé sur la prévision de la demande des produits du terroir marocain sur les marchés national et international. L'objectif est d'anticiper la demande du marché et de formuler des stratégies efficaces pour le développement et la commercialisation des produits. Ces stratégies seront basées sur les données, issues des sources officielles. Un aspect crucial du projet implique le nettoyage des données pour identifier les produits pertinents et les caractéristiques pour le développement des stratégies. Le choix des algorithmes de prévision de la demande sera justifié, et la visualisation des données jouera un rôle clé dans la présentation des résultats.

Table de matières

List of Figures

General introduction

In today's rapidly evolving marketplace, businesses encounter the ongoing challenge of understanding and meeting consumer demand effectively. This challenge is particularly pertinent for industries involved in the production and distribution of local products, where keeping abreast of market trends and demand fluctuations is essential for success. The project titled "Forecasting Demand for Moroccan Local Products in National and International Markets" seeks to address this challenge by employing data-driven decision-making techniques to anticipate demand patterns for Moroccan local products.

This project aims to provide valuable insights into the demand dynamics of Moroccan local products, both domestically and internationally. By analysing historical export data and leveraging predictive modelling techniques, we aim to forecast future demand trends and identify strategic pathways for product development and commercialisation. These strategies will be derived from the data collected from official sources and refined through meticulous data cleaning processes.

Through collaborative efforts and comprehensive analysis, this project endeavours to equip stakeholders in the Moroccan local products industry with the insights and strategies necessary to navigate the complexities of the market landscape and foster sustainable growth.

Chapter 1

Project Concept

1.1 Introduction

This chapter presents the objectives of our data-driven exploration for forecasting demand in the Moroccan local products industry. Our goal is to gain deeper insights into the market dynamics and factors influencing demand, and to develop effective predictive models. We will outline the methodology and approach employed to meet these objectives.

1.2 Objectives

The main objective of this report is to apply the techniques we learned in the Data Driven Decision-Making Class on some real-life data:

- Strategy formulation: Data-driven insights.
- Data cleaning: Relevant product identification.
- Algorithm selection: Demand forecasting.
- Effective visualization: Insight presentation.
- Stakeholder insights: Actionable recommendations.

The project aims to demonstrate the application of data-driven decision-making principles to address real-world challenges in market forecasting and product strategy development. Through comprehensive data analysis and visualization, we seek to provide valuable insights that inform strategic decision-making processes.

1.3 Approach and methodology

For this project, we implemented a structured approach to analyse the dataset and extract insights regarding the demand forecast for Moroccan local products. The methodology involved the following steps:

- **Data Collection and Exploration:** We collected the dataset and conducted an initial exploration to understand its characteristics, including the number of records, features, missing values, and statistical summaries of each feature.

- **Data Visualization:** Various visualization techniques such as histograms to gain insights into the distribution of product demand and to identify any trends or anomalies.

- **Data Cleaning:** We performed data cleaning tasks such as handling missing values, removing duplicates, and standardizing data formats.
- **Feature Engineering:** Additional features were created or derived from the original dataset to enhance the predictive power of the models.
- **Model Selection and Evaluation:** Several machine learning algorithms were trained and evaluated on the dataset to forecast demand accurately. Model performance was assessed using appropriate evaluation metrics and cross-validation techniques.
- **Interpretation and Presentation:** The insights obtained from the analysis were interpreted and presented in a clear and concise manner, highlighting key findings and actionable recommendations for stakeholders in the Moroccan local products' industry.

By adhering to this structured methodology, we were able to gain valuable insights into demand forecasting for Moroccan local products and provide actionable recommendations for stakeholders. This approach serves as a comprehensive framework for similar projects in the future.

1.4 Conclusion

In summary, this chapter has outlined our objectives, approach, and methodology for forecasting demand in the Moroccan local products industry. Our goal is to leverage data-driven techniques to accurately forecast demand, formulate effective strategies, and provide actionable insights to stakeholders.

Chapter 2

Dataset

2.1 Introduction

In this chapter, we will take care of exploring our dataset, describe it, visualize it to achieve a better understanding of it, clean it from outliers, missing values, and duplicates, and finally drop the useless columns

2.2 Data Collection

The data utilized in this report for the analysis of Moroccan terroir products was sourced meticulously from **TRADE MAP** Trade statistics for international business development. TRADE MAP provides an extensive range of trade data on a monthly, quarterly, and yearly basis, covering import and export values, volumes, growth rates, market shares, and additional metrics. Leveraging various visual aids such as tables, graphs, and maps,TRADE MAP offers essential insights into export performance, international demand, alternative markets, and competitive landscapes.

2.3 Data description

The dataset comprises three key files: "export_value_growth","exported_value" and "share_of_country_cluster_exports" offering insights into Morocco's export trends from 2003 to 2022.

- "export_value_growth" shows percentage growth.
- "share_of_country_cluster_exports" illustrates Morocco's cluster export share.
- "exported_value" presents actual export values.

After loading our dataset from the aforementioned files into a pandas dataframe, we can inspect the initial five rows of our dataset using the head() function:

```
[5] # Display the first few rows of the dataset to visualize the export values by product
exported_value.head()
```

	Code	Product label	Exported value in 2003	Exported value in 2004	Exported value in 2005	Exported value in 2006	Exported value in 2007	Exported value in 2008	Exported value in 2009	Exported value in 2010	...
0	TOTAL	All products	8738341	9870179	11197413	12731303	15321255	20094617	14027369	17764791	...
1	31	Fertilisers	371608	431325	444344	533674	876747	1412452	697642	1561945	...
2	87	Vehicles other than railway or tramway rolling...	46680	55256	76111	117890	161777	201129	223629	245346	...
3	85	Electrical machinery and equipment and parts t...	1176179	1257912	1599657	1930282	2359746	2504401	1998575	2632491	...
4	62	Articles of apparel and clothing accessories, ...	1952757	2116194	2040800	2377490	2532616	2483422	2236326	2142622	...

Figure 2.1: The first few rows of the dataset to visualize the export values by product.

```
# Display the initial rows of the dataset to visualize the growth in export values by product
export_value_growth.head()
```

	Code	Product label	exportations growth in value between 2003-2004, %	exportations growth in value between 2004-2005, %	exportations growth in value between 2005-2006, %	exportations growth in value between 2006-2007, %	exportations growth in value between 2007-2008, %	exportations growth in value between 2008-2009, %	exportations growth in value between 2009-2010, %	exportations growth in value between 2010-2011, %
0	TOTAL	All products	13.0	13.0	14.0	20.0	31.0	-30.0	27.0	22.0
1	31	Fertilisers	16.0	3.0	20.0	64.0	61.0	-51.0	124.0	48.0
2	87	Vehicles other than railway or tramway rolling...	18.0	38.0	55.0	37.0	24.0	11.0	10.0	74.0
3	85	Electrical machinery and equipment and parts t...	7.0	27.0	21.0	22.0	6.0	-20.0	32.0	21.0
4	62	Articles of apparel and clothing accessories,	8.0	-4.0	16.0	7.0	-2.0	-10.0	-4.0	6.0

Figure 2.2: The 5 first rows of the dataset to visualize the growth in export values by product.

```
share_of_country_cluster_exports.head()
```

	Code	Product label	Share in value in country's cluster exports, % in 2003	Share in value in country's cluster exports, % in 2004	Share in value in country's cluster exports, % in 2005	Share in value in country's cluster exports, % in 2006	Share in value in country's cluster exports, % in 2007	Share in value in country's cluster exports, % in 2008	Share in value in country's cluster exports, % in 2009	Share in value in country's cluster exports, % in 2010	...	Sha val coun cl exp % in
0	TOTAL	All products	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	...	
1	31	Fertilisers	4.25	4.37	3.97	4.19	5.72	7.03	4.97	8.79	...	
2	87	Vehicles other than railway or tramway rolling...	0.53	0.56	0.68	0.93	1.06	1.00	1.59	1.38	...	
3	85	Electrical machinery and equipment and parts t...	13.46	12.74	14.29	15.16	15.40	12.46	14.25	14.82	...	
4	62	Articles of apparel and clothing accessories,	22.35	21.44	18.23	18.67	16.53	12.36	15.94	12.06	...	

Figure 2.3: The 5 first rows of the dataset to visualize the share in export values by product.

We are first curious to know the dimensions of our dataset. We find this out using the shape attribute of the dataframe:

```
#Dimensions of our dataset
exported_value.shape
```

```
(98, 21)
```

Figure 2.4: Shape of the "exported_value" dataset

```
#Dimensions of our dataset
export_value_growth.shape
```

```
(98, 20)
```

Figure 2.5: Shape of the "export_value_growth" dataset

```
[50] #Dimensions of our dataset
     share_of_country_cluster_exports.shape
```

```
(98, 21)
```

Figure 2.6: Shape of the "share_of_country_cluster_exports"dataset

Using the info() method, we can expose more typical details of each column as follow:

```
#Check the data info
exported_value.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 98 entries, 0 to 97
Data columns (total 22 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Code                    98 non-null     object
 1   Product label           98 non-null     object
 2   Exported value in 2003  98 non-null     int64
 3   Exported value in 2004  98 non-null     int64
 4   Exported value in 2005  98 non-null     int64
 5   Exported value in 2006  98 non-null     int64
 6   Exported value in 2007  98 non-null     int64
 7   Exported value in 2008  98 non-null     int64
 8   Exported value in 2009  98 non-null     int64
 9   Exported value in 2010  98 non-null     int64
 10  Exported value in 2011  98 non-null     int64
 11  Exported value in 2012  98 non-null     int64
 12  Exported value in 2013  98 non-null     int64
 13  Exported value in 2014  98 non-null     int64
 14  Exported value in 2015  98 non-null     int64
 15  Exported value in 2016  98 non-null     int64
 16  Exported value in 2017  98 non-null     int64
 17  Exported value in 2018  98 non-null     int64
 18  Exported value in 2019  98 non-null     int64
```

Figure 2.7: info of the "exported_value" dataset

The columns include 'Code' and 'Product label' for identification purposes, alongside columns labeled 'Exported value in [year]' spanning from 2003 to 2022, indicating the exported value of the respective product in each year. Notably, the 'Code' column contains object data type, whereas the remaining columns primarily consist of integer data type, representing the exported values.

```
[9] export_value_growth.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 98 entries, 0 to 97
Data columns (total 22 columns):
 #   Column                                         Non-Null Count  Dtype
---  ------                                         --------------  -----
 0   Code                                           98 non-null     object
 1   Product label                                  98 non-null     object
 2   exportations growth in value between 2003-2004, %  96 non-null     float64
 3   exportations growth in value between 2004-2005, %  97 non-null     float64
 4   exportations growth in value between 2005-2006, %  96 non-null     float64
 5   exportations growth in value between 2006-2007, %  97 non-null     float64
 6   exportations growth in value between 2007-2008, %  97 non-null     float64
 7   exportations growth in value between 2008-2009, %  97 non-null     float64
 8   exportations growth in value between 2009-2010, %  97 non-null     float64
 9   exportations growth in value between 2010-2011, %  97 non-null     float64
 10  exportations growth in value between 2011-2012, %  97 non-null     float64
 11  exportations growth in value between 2012-2013, %  96 non-null     float64
 12  exportations growth in value between 2013-2014, %  96 non-null     float64
 13  exportations growth in value between 2014-2015, %  97 non-null     float64
 14  exportations growth in value between 2015-2016, %  97 non-null     float64
 15  exportations growth in value between 2016-2017, %  97 non-null     float64
 16  exportations growth in value between 2017-2018, %  97 non-null     float64
 17  exportations growth in value between 2018-2019, %  97 non-null     float64
 18  exportations growth in value between 2019-2020, %  97 non-null     float64
 19  exportations growth in value between 2020-2021, %  96 non-null     float64
 20  exportations growth in value between 2021-2022, %  95 non-null     float64
```

Figure 2.8: info of the "export_value_growth" dataset

The columns include 'Code' and 'Product label' for identification purposes. The remaining columns represent various growth rates in export values between consecutive years from 2003 to 2022, expressed as percentages. Additionally, the column 'Exported value in 2022, US Dollar thousand' presents the exported value in the year 2022, measured in US Dollars (thousands). The majority of columns contain float64 data type, denoting the numerical nature of the growth rates, except the 'Exported value in 2022, US Dollar thousand' column, which contains int64 data type.

```
share_of_country_cluster_exports.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 98 entries, 0 to 97
Data columns (total 22 columns):
 #   Column                                                 Non-Null Count  Dtype
---  ------                                                 --------------  -----
 0   Code                                                   98 non-null     object
 1   Product label                                          98 non-null     object
 2   Share in value in country's cluster exports, % in 2003 98 non-null     float64
 3   Share in value in country's cluster exports, % in 2004 98 non-null     float64
 4   Share in value in country's cluster exports, % in 2005 98 non-null     float64
 5   Share in value in country's cluster exports, % in 2006 98 non-null     float64
 6   Share in value in country's cluster exports, % in 2007 98 non-null     float64
 7   Share in value in country's cluster exports, % in 2008 98 non-null     float64
 8   Share in value in country's cluster exports, % in 2009 98 non-null     float64
 9   Share in value in country's cluster exports, % in 2010 98 non-null     float64
 10  Share in value in country's cluster exports, % in 2011 98 non-null     float64
 11  Share in value in country's cluster exports, % in 2012 98 non-null     float64
 12  Share in value in country's cluster exports, % in 2013 98 non-null     float64
 13  Share in value in country's cluster exports, % in 2014 98 non-null     float64
 14  Share in value in country's cluster exports, % in 2015 98 non-null     float64
 15  Share in value in country's cluster exports, % in 2016 98 non-null     float64
 16  Share in value in country's cluster exports, % in 2017 98 non-null     float64
 17  Share in value in country's cluster exports, % in 2018 98 non-null     float64
 18  Share in value in country's cluster exports, % in 2019 98 non-null     float64
 19  Share in value in country's cluster exports, % in 2020 98 non-null     float64
 20  Share in value in country's cluster exports, % in 2021 98 non-null     float64
```

Figure 2.9: info of the "share_of_country_cluster_exports"dataset

The columns include 'Code' and 'Product label' for identification purposes. The remaining columns present the share of value contributed by Morocco to its cluster exports, expressed as percentages, for each year from 2003 to 2022. All columns, except for 'Code' and 'Product label', contain float64 data type, reflecting the numerical nature of the share percentages.

Now, we're curious to know more about the statistical distribution of our data. For this aim, we use the describe() method that gives us the following results:

```
exported_value.describe()
```

	value_2003	value_2004	value_2005	value_2006	value_2007	value_2008	value_2009	value_2010
count	9.800000e+01	9.800000e+01	9.800000e+01	9.800000e+01	9.800000e+01	9.800000e+01	9.800000e+01	9.800000e+01
mean	1.783335e+05	2.014323e+05	2.285186e+05	2.598225e+05	3.126787e+05	4.100942e+05	2.862728e+05	3.625467e+05
std	9.112747e+05	1.025980e+06	1.159090e+06	1.318700e+06	1.583579e+06	2.081394e+06	1.445491e+06	1.828460e+06
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.920250e+03	1.464500e+03	3.471000e+03	5.006750e+03	5.985000e+03	6.472500e+03	4.936250e+03	6.190750e+03
50%	1.313200e+04	1.690100e+04	1.882900e+04	2.399000e+04	2.443850e+04	2.651550e+04	2.330050e+04	3.067950e+04
75%	4.119425e+04	4.821850e+04	5.780675e+04	6.796100e+04	8.801100e+04	9.365950e+04	8.859300e+04	1.162568e+05
max	8.738341e+06	9.870179e+06	1.119741e+07	1.273130e+07	1.532126e+07	2.009462e+07	1.402737e+07	1.776479e+07

Figure 2.10: statistical description of columns of the "exported_value" dataset

```
export_value_growth.describe()
```

	growth_2003	growth_2004	growth_2005	growth_2006	growth_2007	growth_2008	growth_2009
count	96.000000	97.000000	96.000000	97.000000	97.000000	97.000000	97.000000
mean	51.875000	641.061856	358.781250	169.494845	226.216495	18.216495	85.556701
std	200.681903	4170.929897	2732.128741	1180.416453	1354.814634	171.609551	415.011997
min	-97.000000	-100.000000	-89.000000	-99.000000	-91.000000	-95.000000	-99.000000
25%	-7.250000	-4.000000	-0.250000	-3.000000	-15.000000	-24.000000	-3.000000
50%	15.000000	15.000000	16.000000	19.000000	9.000000	-6.000000	13.000000
75%	34.500000	40.000000	48.250000	53.000000	36.000000	14.000000	50.000000
max	1625.000000	32733.000000	26749.000000	11467.000000	12339.000000	1590.000000	3363.000000

Figure 2.11: statistical description of columns of the "export_value_growth" dataset

```
share_of_country_cluster_exports.describe()
```

	share_2003	share_2004	share_2005	share_2006	share_2007	share_2008	share_2009
count	98.000000	98.000000	98.000000	98.000000	98.000000	98.000000	98.000000
mean	2.040714	2.040408	2.040714	2.040816	2.040816	2.040918	2.040714
std	10.428636	10.394769	10.351585	10.357796	10.335841	10.357963	10.304745
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.020000	0.012500	0.030000	0.042500	0.040000	0.032500	0.032500
50%	0.150000	0.170000	0.165000	0.190000	0.160000	0.130000	0.170000
75%	0.467500	0.490000	0.515000	0.535000	0.577500	0.465000	0.635000
max	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000

Figure 2.12: statistical description of columns of "share_of_country_cluster_exports"dataset

2.4 Data visualisation

In this section, we will present our data using various types of plots to get to know it better. First, we visualize the export data for the top 20 products for each dataset.

- The visual representation exhibits the initial rows of the dataset, showcasing the export values associated with various products.

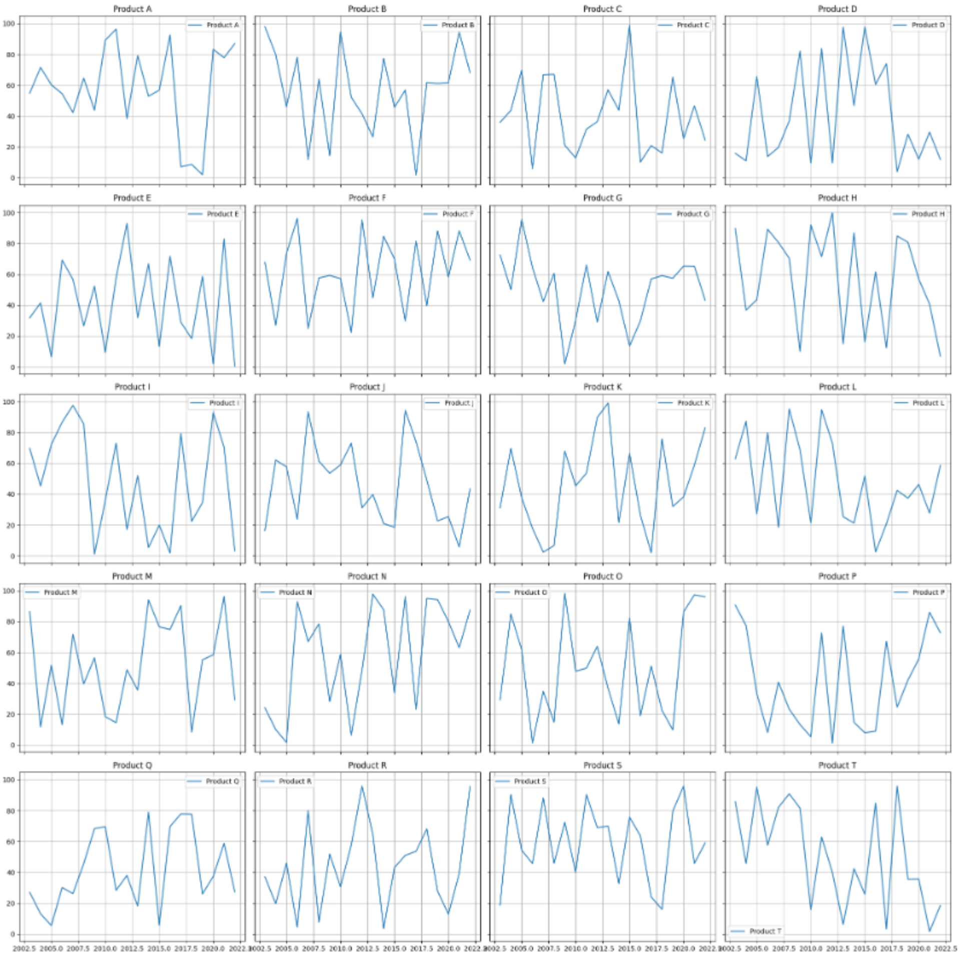


Figure 2.13: Displays initial rows to visualize export value growth by product.

- Displaying export data for the initial 20 products, utilizing subplots to illustrate the growth in export values over time. Products are arranged in multiple rows and columns for a clearer visualization.

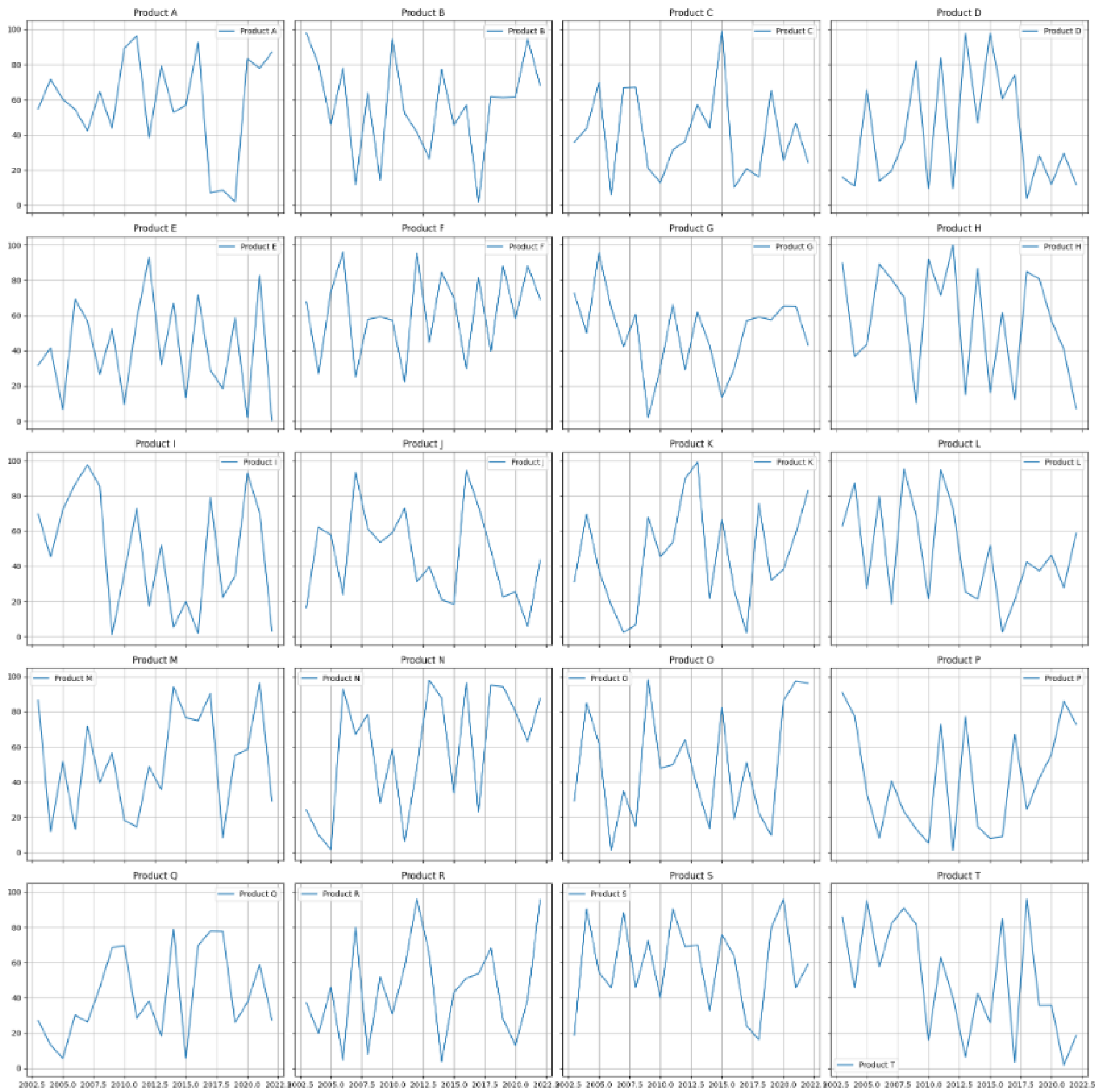


Figure 2.14: Visualize export growth for 20 products.

- The visualization constructs subplots to depict the growth in export value over time for each of the initial 20 products. By organizing the products into multiple rows and columns, the visualization ensures a structured and comprehensible presentation. This layout facilitates clear interpretation of export value trends across different product categories, aiding in the analysis of export dynamics.

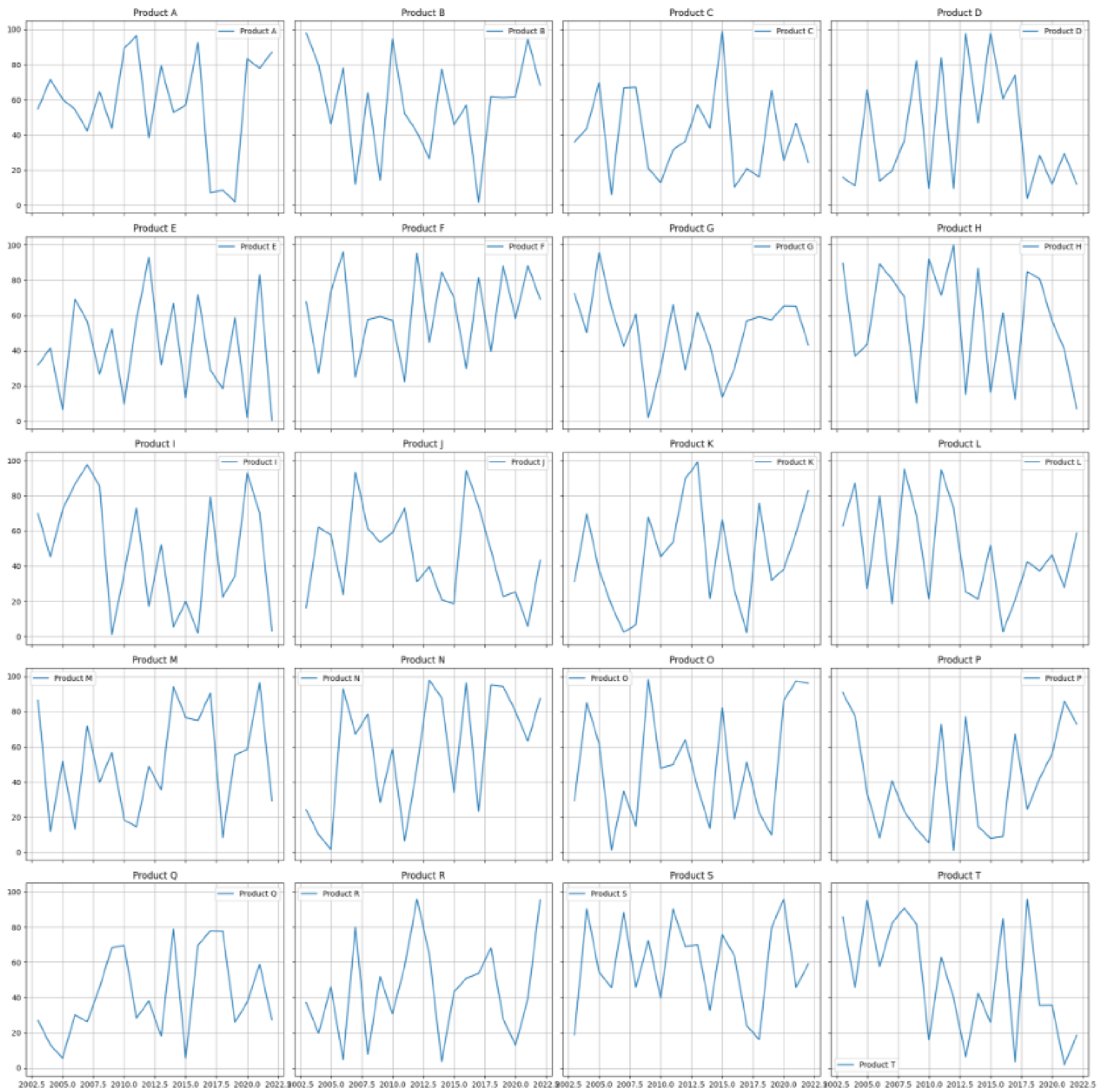


Figure 2.15: Visualize export growth for 20 products.

Now we are going to merge datasets related to exports by product values, growth in value, and share in value percentage. Initially, columns irrelevant to the merging process, such as 'code' from the 'exported_value' and 'export_value_growth' datasets, are dropped. Subsequently, the datasets are merged based on the 'product' column to create a comprehensive dataset, 'merged_exports,' encompassing all relevant information. This amalgamation facilitates a holistic analysis of export dynamics, allowing for insights into the export values, growth trends, and share percentages across different products. The 'merged_exports.head()' command is employed to display the initial rows of the merged dataset, providing a glimpse into the combined data for further analysis and interpretation.

```python
# Merge exports by product values, growth in value and share in value percentage
exported_value = exported_value.drop(columns=['code'])
export_value_growth = export_value_growth.drop(columns=['code', 'value_2022'])
share_of_country_cluster_exports = share_of_country_cluster_exports.drop(columns=['code'])
merged_exports = pd.merge(exported_value, share_of_country_cluster_exports, on='product')
merged_exports = pd.merge(merged_exports, export_value_growth, on='product')
merged_exports.head()
```

	product	value_2003	value_2004	value_2005	value_2006	value_2007	value_2008	value_2009	value_2010	value_2011
0	All products	8738341	9870179	11197413	12731303	15321255	20094617	14027369	17764791	21649934
1	Fertilisers	371608	431325	444344	533674	876747	1412452	697642	1561945	2317995
2	Vehicles other than railway or tramway rolling...	46680	55256	76111	117890	161777	201129	223629	245346	426212
3	Electrical machinery and equipment and parts t...	1176179	1257912	1599657	1930282	2359746	2504401	1998575	2632491	3172824
4	Articles of apparel and clothing	1952757	2116194	2040800	2377490	2532616	2483422	2236326	2142622	2280072

Figure 2.16: Merging datasets.

The following bar plots visualize the export values by product for the years 2003, 2013, and 2022.

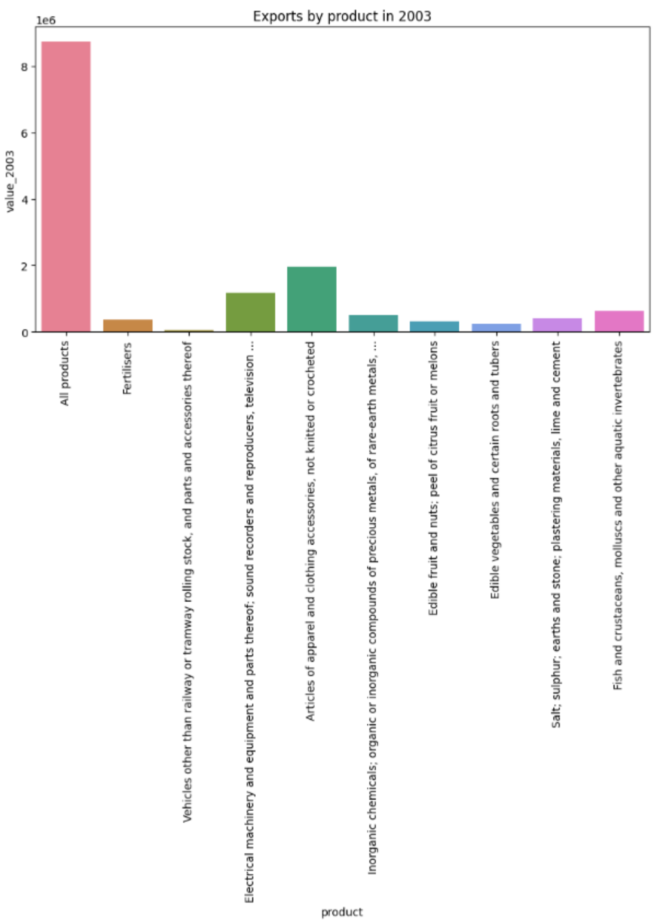


Figure 2.17: export values by product for 2003.

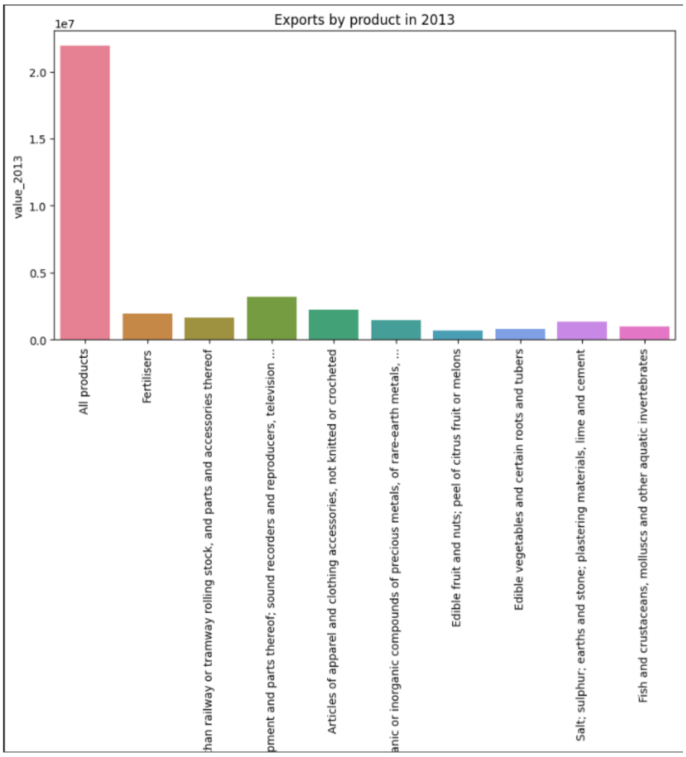


Figure 2.18: export values by product for 2013.

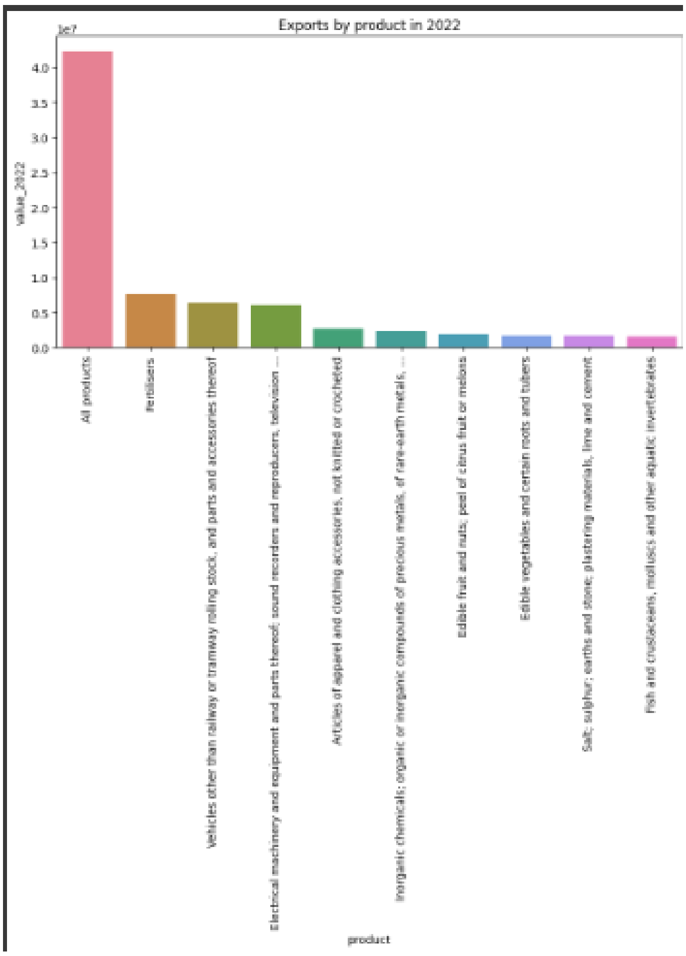


Figure 2.19: export values by product for 2022.

We saved the merged dataset, merged_exports, to a CSV file named "transitional_data.csv". By using the to_csv method with the parameter index=False, the DataFrame is exported without including the index column.

2.5 Data Cleaning and Preprocessing

The describe() function provides a statistical summary of the DataFrame transitional_df. It presents descriptive statistics such as count, mean, standard deviation, minimum, maximum, and quartile values for each column, including values from 2003 to 2022.

```
transitional_df.describe()
```

	value_2003	value_2004	value_2005	value_2006	value_2007
count	9.800000e+01	9.800000e+01	9.800000e+01	9.800000e+01	9.800000e+01
mean	1.783335e+05	2.014323e+05	2.285186e+05	2.598225e+05	3.126787e+05
std	9.112747e+05	1.025980e+06	1.159090e+06	1.318700e+06	1.583579e+06
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.920250e+03	1.464500e+03	3.471000e+03	5.006750e+03	5.985000e+03
50%	1.313200e+04	1.690100e+04	1.882900e+04	2.399000e+04	2.443850e+04
75%	4.119425e+04	4.821850e+04	5.780675e+04	6.796100e+04	8.801100e+04
max	8.738341e+06	9.870179e+06	1.119741e+07	1.273130e+07	1.532126e+07

8 rows × 59 columns

Figure 2.20: Statistical summary for each column, including values from 2003 to 2022 of the DataFrame.

To ensure the dataset's integrity and reliability, identifying and addressing missing values is crucial for accurate analysis and modeling.

```
#Check the count of missing values in the time series data
time_series.isna().sum()
```

```
product    0
year       0
value      0
share      0
dtype: int64
```

Figure 2.21: Count of missing values in the time series data.

2.6 Proportion of Exports by Product

Generating a donut chart to depict the proportion of exports for each product. It calculates the total exports for each product, then plots the chart with labels and percentages. Additionally, it adds a circle in the center to create a hole in the donut chart for aesthetic purposes. Font sizes are adjusted for readability, and a legend is included for clarity. The aspect ratio is set to ensure the pie is drawn as a circle. Finally, the plot is displayed

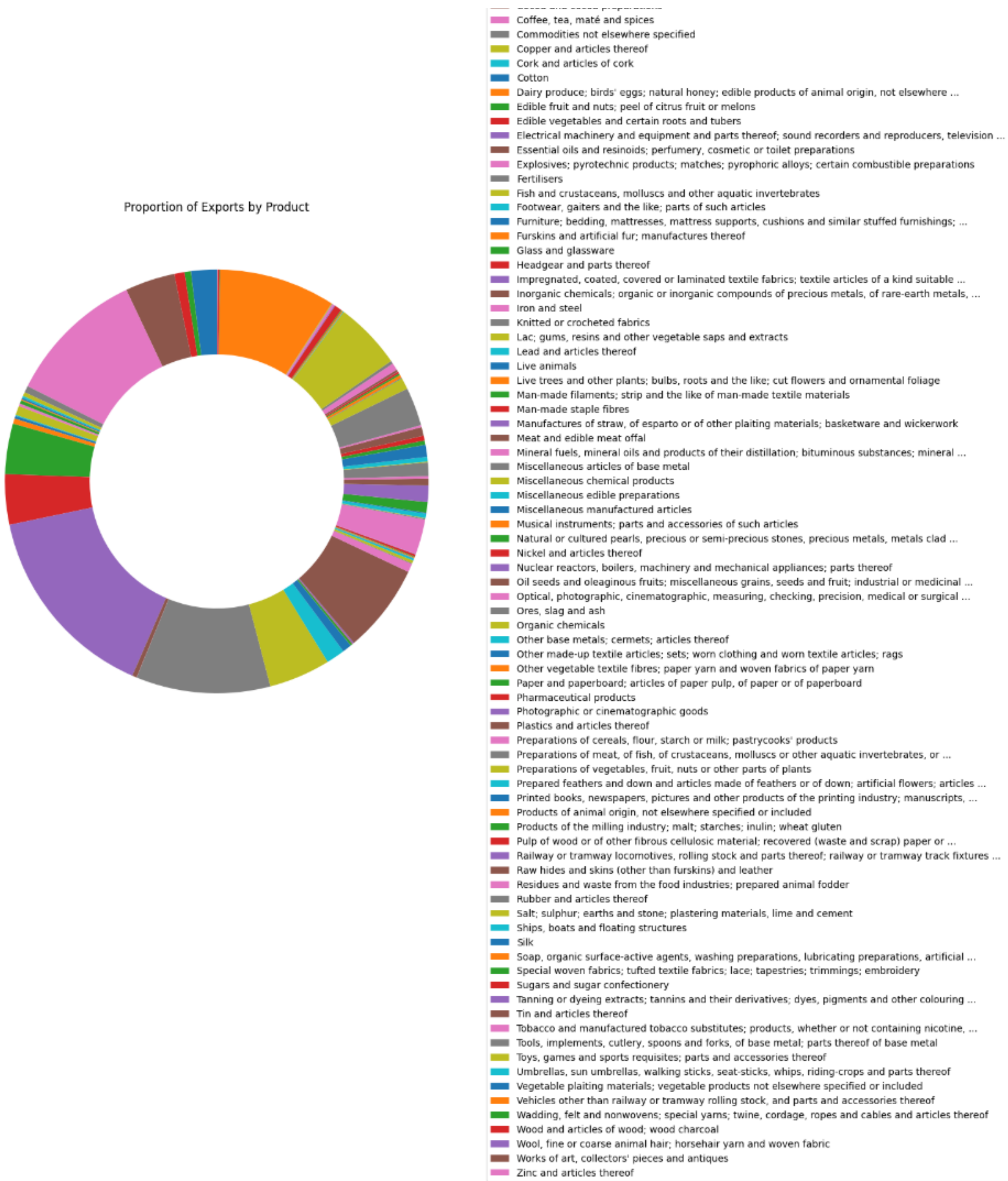


Figure 2.22: Donut chart to depict the proportion of exports.

Ploting the export time series for each product over the years, ensuring the x-axis displays integer years and includes all years from 2003 to 2022. Axis labels are set up for clarity, and the plot is displayed.

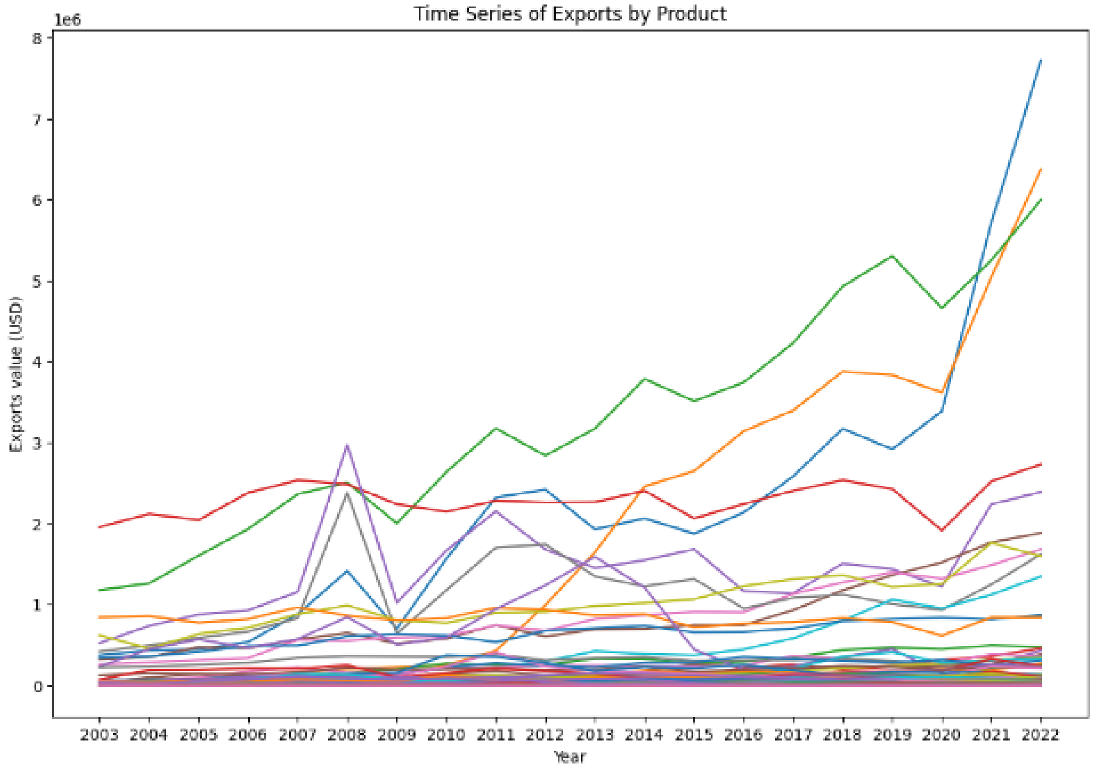
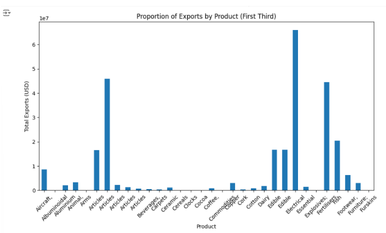
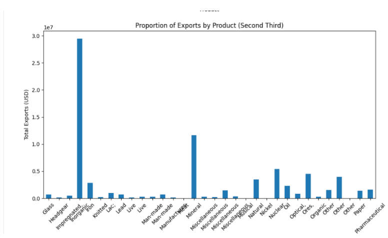


Figure 2.23: Plot the export time series for each product.

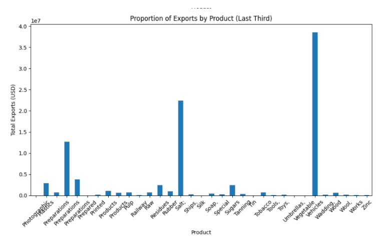
After that we divided products into three groups and plots bar charts for each group, illustrating the proportion of exports by product. Axis labels and titles are set for clarity, and x-axis labels are rotated for better readability. Finally, the plot is displayed.



(a)



(b)



(c)

Figure 2.24: Divide products into three groups and visualize their export proportions through bar charts.

Chapter 3

Models

3.1 Introduction

In this chapter, we will explore two advanced models used for forecasting demand for Moroccan local products. These models, an LSTM-based Recurrent Neural Network (RNN) and an ARIMA model, were selected for their ability to capture temporal dependencies and trends in time series data. The detailed methodology and results from both models will be discussed, providing a comprehensive understanding of their performance and utility in predicting demand.

3.2 Processed Data

After completing the data cleaning and preprocessing steps, we saved the processed dataset as processed_data.csv.

```
processed_df.describe()
```

	year	value	share
count	1940.000000	1.940000e+03	1940.000000
mean	2012.500000	2.241180e+05	1.030902
std	5.767768	6.286048e+05	2.642180
min	2003.000000	0.000000e+00	0.000000
25%	2007.750000	5.224000e+03	0.030000
50%	2012.500000	2.590350e+04	0.130000
75%	2017.250000	1.120512e+05	0.530000
max	2022.000000	7.715003e+06	22.350000

Figure 3.1: The first few rows of the processed dataset.

```
processed_df.head()
```

	product	year	value	share
0	Fertilisers	2003	371608	4.25
1	Fertilisers	2004	431325	4.37
2	Fertilisers	2005	444344	3.97
3	Fertilisers	2006	533674	4.19
4	Fertilisers	2007	876747	5.72

Figure 3.2: statistical description of the processed dataset.

3.3 The First Model: LSTM-based RNN

First, we load the preprocessed data, normalize it, and create temporal sequences for model training. We use an RNN (specifically LSTM) due to its ability to capture temporal dependencies in sequential data. The data is split into training and testing sets, and a simple RNN model with one LSTM layer and a dense output layer is created. Early stopping is implemented to prevent overfitting, and the model is trained and evaluated on the test set.
Data preprocessing and normalization are followed by the creation of temporal sequences for training an RNN model, specifically LSTM, which is then evaluated using early stopping to prevent overfitting.

3.3.1 Train and Test The Model

After loading the data, we normalize it, create temporal sequences for model training, split the data into training and testing sets, define and train an RNN model, evaluate its performance, and make predictions.

```
Epoch 1/100
44/44 [==============================] - 6s 28ms/step - loss: 0.0495 - val_loss: 0.0333
Epoch 2/100
44/44 [==============================] - 0s 8ms/step - loss: 0.0302 - val_loss: 0.0273
Epoch 3/100
44/44 [==============================] - 0s 10ms/step - loss: 0.0223 - val_loss: 0.0194
Epoch 4/100
44/44 [==============================] - 0s 8ms/step - loss: 0.0173 - val_loss: 0.0165
Epoch 5/100
44/44 [==============================] - 0s 8ms/step - loss: 0.0158 - val_loss: 0.0156
Epoch 6/100
44/44 [==============================] - 0s 8ms/step - loss: 0.0149 - val_loss: 0.0149
Epoch 7/100
44/44 [==============================] - 0s 8ms/step - loss: 0.0147 - val_loss: 0.0140
Epoch 8/100
44/44 [==============================] - 0s 8ms/step - loss: 0.0139 - val_loss: 0.0135
Epoch 9/100
44/44 [==============================] - 0s 9ms/step - loss: 0.0124 - val_loss: 0.0129
Epoch 10/100
44/44 [==============================] - 0s 8ms/step - loss: 0.0113 - val_loss: 0.0109
Epoch 11/100
44/44 [==============================] - 0s 8ms/step - loss: 0.0108 - val_loss: 0.0098
Epoch 12/100
44/44 [==============================] - 0s 8ms/step - loss: 0.0097 - val_loss: 0.0088
Epoch 13/100
...
Prédiction: [2.0110756e+03 6.5537594e+04 3.9619386e-02]
Valeur réelle: [2.011e+03 5.854e+04 2.700e-01]
---------------------------------------
Mean Squared Error: 12336641834.161583
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Figure 3.3: Training and testing the model.

3.3.2 Comparison of real values and predictions.

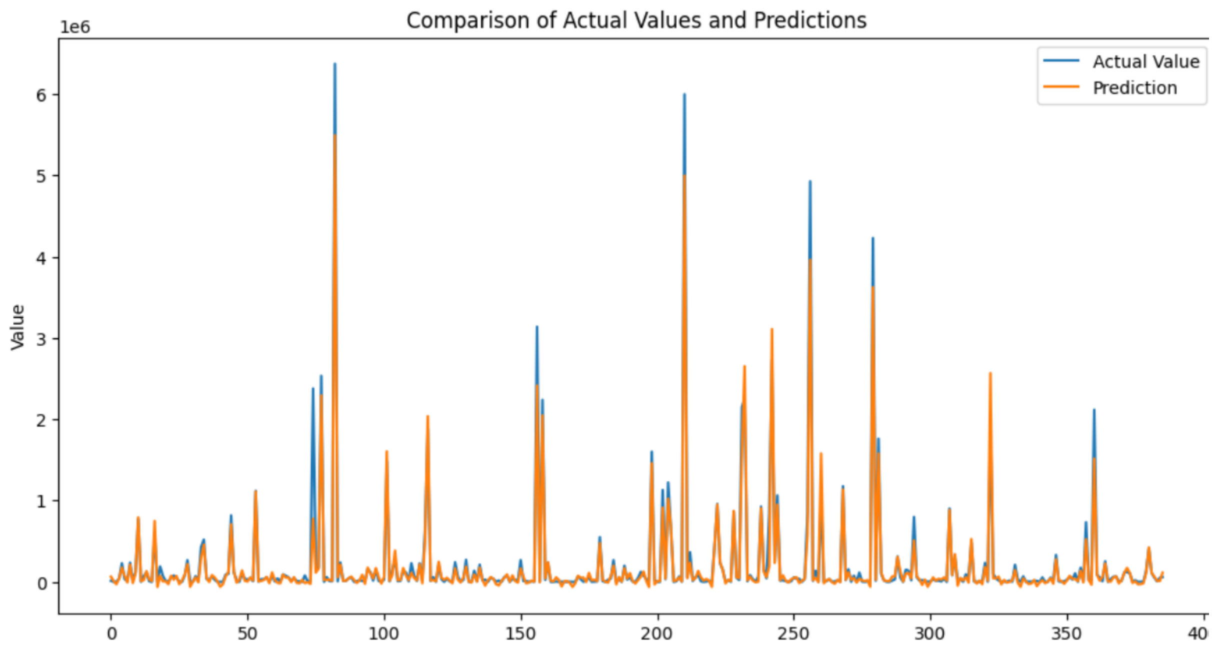


Figure 3.4: Comparison of real values and predictions.

3.4 The Second Model: ARIMA

The ARIMA (AutoRegressive Integrated Moving Average) model is a time series forecasting method that combines autoregressive (AR), differencing (I), and moving average (MA) components. It is suitable for analyzing and predicting time series data with temporal dependencies and trends. ARIMA models are capable of capturing seasonality and non-stationarity in the data, making them valuable tools for forecasting.
In the second model, we load the data, normalize it, create temporal sequences for model training, and split the data into training and testing sets. The ARIMA model is then defined, trained, and used to make predictions. We compare the real values and predictions to evaluate the model's performance.

3.4.1 Model Training

List of products to forecast demand, storing the forecast, actual values, and error for each product.

```
Forecasting  Aircraft, spacecraft, and parts thereof
Forecasting  Albuminoidal substances; modified starches; glues; enzymes
Forecasting  Aluminium and articles thereof
Forecasting  Animal, vegetable or microbial fats and oils and their cleavage products; prepared edible fats; ...
Forecasting  Arms and ammunition; parts and accessories thereof
Forecasting  Articles of apparel and clothing accessories, knitted or crocheted
Forecasting  Articles of apparel and clothing accessories, not knitted or crocheted
Forecasting  Articles of iron or steel
Forecasting  Articles of leather; saddlery and harness; travel goods, handbags and similar containers; articles ...
Forecasting  Articles of stone, plaster, cement, asbestos, mica or similar materials
Forecasting  Beverages, spirits and vinegar
Forecasting  Carpets and other textile floor coverings
Forecasting  Ceramic products
Forecasting  Cereals
Forecasting  Clocks and watches and parts thereof
Forecasting  Cocoa and cocoa preparations
Forecasting  Coffee, tea, maté and spices
Forecasting  Commodities not elsewhere specified
Forecasting  Copper and articles thereof
Forecasting  Cork and articles of cork
Forecasting  Cotton
Forecasting  Dairy produce; birds' eggs; natural honey; edible products of animal origin, not elsewhere ...
Forecasting  Edible fruit and nuts; peel of citrus fruit or melons
Forecasting  Edible vegetables and certain roots and tubers
Forecasting  Electrical machinery and equipment and parts thereof; sound recorders and reproducers, television ...
...
Forecasting  Wood and articles of wood; wood charcoal
Forecasting  Wool, fine or coarse animal hair; horsehair yarn and woven fabric
Forecasting  Works of art, collectors' pieces and antiques
Forecasting  Zinc and articles thereof
```

Figure 3.5: Training the model.

3.4.2 Model Prediction

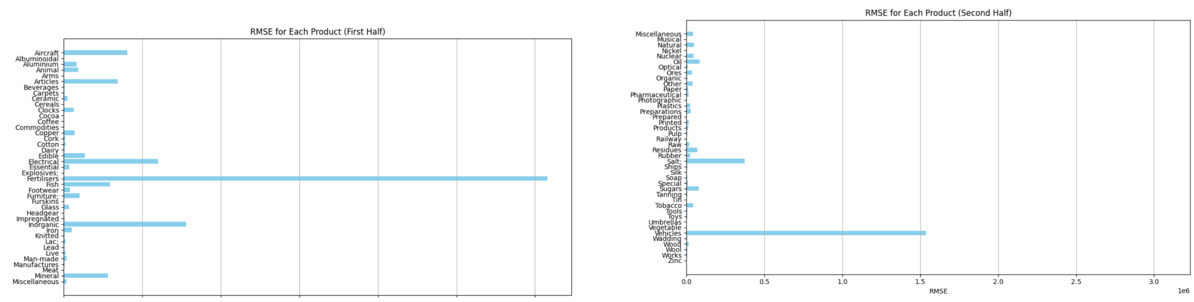
The model iterates a list of products to forecast demand, storing the forecast, actual values, and error for each product.

```
Product:  Aircraft, spacecraft, and parts thereof
RMSE:  403424.0398328307
----------------------------------------
Product:  Albuminoidal substances; modified starches; glues; enzymes
RMSE:  732.3753893244749
----------------------------------------
Product:  Aluminium and articles thereof
RMSE:  82937.18182234194
----------------------------------------
Product:  Animal, vegetable or microbial fats and oils and their cleavage products; prepared edible fats; ...
RMSE:  90344.75632482184
----------------------------------------
Product:  Arms and ammunition; parts and accessories thereof
RMSE:  295.55934772062557
----------------------------------------
Product:  Articles of apparel and clothing accessories, knitted or crocheted
RMSE:  110960.1803074277
----------------------------------------
Product:  Articles of apparel and clothing accessories, not knitted or crocheted
RMSE:  342616.39125152037
----------------------------------------
Product:  Articles of iron or steel
RMSE:  59715.20676845658
----------------------------------------
Product:  Articles of leather; saddlery and harness; travel goods, handbags and similar containers; articles ...
...
Product:  Zinc and articles thereof
RMSE:  2123.197528564087
----------------------------------------
Overall RMSE:  99609.54688945597
```

Figure 3.6: Model Prediction.

3.4.3 Comparison of RMSE

Divides products into two groups and creates horizontal bar charts to visualize the RMSE (Root Mean Squared Error) for each product in each group, facilitating comparison of forecasting accuracy.



(a) (b)

Figure 3.7: Comparison of RMSE.

3.5 Comparative Analysis

The code iterates through a list of products to forecast demand, storing the forecast, actual values, and error for each product. It then divides the products into two groups and creates horizontal bar charts to visualize the RMSE (Root Mean Squared Error) for each product in each group, facilitating the comparison of forecasting accuracy. This comparison helps in identifying the strengths and weaknesses of each model, providing valuable insights into their application in real-world scenarios.

Generale Conclusion

In our endeavor to forecast demand for Moroccan local products in both national and international markets using a data-driven approach, we have embarked on a journey of exploration and analysis that has yielded valuable insights and promising outcomes. Through meticulous data collection, cleaning, and preprocessing, we have set the stage for our predictive modeling efforts. Leveraging advanced techniques such as ARIMA and LSTM-based RNNs, we have developed models capable of capturing the intricate patterns and dynamics inherent in the demand for local products.

Our findings indicate that these models hold promise for accurately forecasting demand trends, enabling stakeholders in the Moroccan local products industry to make informed decisions regarding production, distribution, and market positioning. By uncovering market trends, identifying influential factors, and providing actionable insights, our project aims to empower industry stakeholders with the tools and knowledge needed to thrive in today's competitive landscape.

Bibliography

[1] https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python/

[2] https://www.kaggle.com/datasets