

Appendix 1



Comparative Analysis of Regression Models for Global Crude Oil Trade Prediction with Graph Analytics

A MINI PROJECT REPORT

Submitted by

BLESSY LOUIS (2348416)

in partial fulfilment for the award of the degree

of

MASTER OF SCIENCE

In

DATA SCIENCE

Christ (Deemed to be) University

AUGUST 2024

Appendix 2

CERTIFICATE

This is to certify that this project report “**Comparative Analysis of Regression Models for Global Crude Oil Trade Prediction with Graph Analytics**” is the bonafide work of “**BLESSY LOUIS**” who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

<<Signature of the Supervisor>>
SIGNATURE

Dr.Priya Stella Mary

SUPERVISOR

MSc Data Science

Department Of Computer Science

Submitted for Project Viva-voce examination held on _____

ACKNOWLEDGEMENT

I thank those people who have contributed in one way or another for the completion of this project.

I am greatly thankful to **Dr.Priya Stella** Mary who had been my supervisor. His expert guidance, much inspiration, and encouragement have really proved propelling in this project. His fruitful and constructive criticism has helped in improving the quality of this work.

Also, my thanks go to all my peers for the collaborative efforts and the intellectual interactions between us. The efforts and contributions, through brainstorming, reviewing my work, or giving technical help, did a big part in the addition of depth and breadth to this project.

Special thanks to **Christ University** for providing resources and datasets provided through these resources, which made this project possible for analysis and implementation. Their support has enabled me to perform comprehensive data analysis and leverage advanced machine learning models.

Appreciation of the instrumental help that the [Specific Software, Libraries, or Platforms] has been able to offer in executing the multiple tasks required for the project: functionalities offered by these tools and technologies were key to the realization of the objectives of the project.

I also appreciate the invaluable support and feedback from the [Project Review Committee/Industry Experts] that has gone an extra mile in ensuring the approaches taken for the project were sharpened in matching reality and relevance thereof.

Finally, a lot of love must be given to my family and friends for their strong support and encouragement. They are my motivators—understanding and patient—throughout this journey.

Thank you, everyone, for the great contribution and support. The project would not have been possible without your involvement and encouragement.

ABSTRACT

Graph analytics is therefore one of the powerful new tools within data science, which provides insights into very complex, doubtful relationships and interactions in data networks. This capability to model and analyze data as graphs allows one to have a nuanced understanding of the underlying structures from different phenomena and dynamics ensuing from the causality of those structures. It will be, therefore, a focus of the paper on how graph analytics can be critical in the global crude oil market, where central to any strategy or decision is the understanding of the trading pattern, relationships, and anomalies.

Here, the main objective of this research project will be twofold in seeking to address the challenge that arises from predicting future trade value and identifying anomalous patterns in trade benefits through the application of machine learning methods with graph-based techniques. This work will make use of graph representation on modeling trade relationships both at country and continent levels through GNN and the help of classic machine learning, like Linear Regression, Decision Trees, and Random Forest. It thus improves the prediction accuracy by making use of the graph embeddings as well as temporal features for a robust insight into trade dynamics.

The project contributes in terms of theory and practice. The project provides an idea about the most effective ways to predict trade values and detect anomalies by comparing different machine-learning models applied in graph analytics. The results that may be anticipated to arise from this would have very valuable insights into the patterns of world trade, which could find applications in economic forecasting, trade policy formulation, and strategic resource management. Coupling graph analytics with machine learning ushers in a new frontier in data science, which only promises additional opportunities to understand and exploit complex data structures further.

Appendix 3

CONTENTS

Certificate	ii
Acknowledgement	iii
Abstract	iv

SL No.	CHAPTER	PAGE
1	Introduction	7
2	Literature Review	9
3	Methodology	14
4	Findings and Interpretation	40
5	Conclusion and future work	43
6	Reference	47

LIST OF FIGURES AND TABLES

Figure 1: Graph Neural Networks Architecture

Figure 2: GCN operations

Figure 3: Applications of GNN

Figure 4: GNN for web scale recommendation systems

Figure 5: Kaggle view of the dataset

Figure 6: Python code of Detection of missing values

Figure 7: Python Implementation of Detection of outliers

Figure 8: Python Implementation of handling of outliers

Figure 9: Python implementation, Data normalization and Standardization, feature engineering

Figure 10: Graph created for the global crude oil trade between Continents

Figure 11: Excel preview of the dataset

Figure 12: Python implementation to understand the structure of the dataset

Figure 13: Graphical presentation of the features in the dataset

Figure 14: Flow of Operations perform

Figure 15: Python code for creation of network

Figure 16: Python implementation of Dijkstra Algorithm

Figure 17: Python implementation of Louvain method

Figure 18: Python implementation of regression models

Figure 19: Graph created

Figure 20: Highlighted graph after finding shortest path

Figure 21: Graph highlighting the communities identified

Figure 22: Results of Linear Regression

Figure 23: Results of Decision Tree

Figure 24: Results of Random Forest

Figure 25: Representation of key findings in the project

Figure 26: Gantt Chart of Project time line

Table 1: Comparative Analysis of the Models applied for trade value predictions.

1. INTRODUCTION

Graph Analytics and Overview of the Domain

Being the global transactional commodity, therefore, global crude oil trade is dynamic and complicated, but surely plays a very essential role in the world economy. This right here shows that crude oil is an important commodity, and trading in it is affected by various factors like geopolitical events, economic policies, and fluctuations in supply and demand of technological advancements. Such a complex configuration of a web of trade relationships in today's world makes the understanding of policymakers, economists, and energy companies obligatory. The multitudes of connected entities and transactions associated with such trade networks make their details extremely complex and often beyond traditional analysis techniques that rely on linearity. In so doing, the power of graph analytics delivers a potent framework for modeling and analysis of relationships and interactions in this sphere.

Analytics of graphs makes for an opportunity to represent the global crude oil trade network as a graph, where entities represent nodes. In this way, patronage flows can be analyzed, thus identifying and detecting the pivotal centers of trade within that network. Graph algorithms, especially shortest path and community detection, can be very useful in giving insights into not only the efficiency of the trade route but also into the clustering of trading partners. Graph analytic capabilities, through the means of integration with machine learning techniques, are brought to a level where they predict future trade patterns, optimize trade strategies, and increase decision information. ML is also a method to leverage the structural properties in the graph, along with the historical data linked to trades, to forecast the values of trade and find the emerging trends. This union of graph analytics and ML is gaining acceptance in the global trade domain and is bringing with it immense importance to deliver better accuracy and help provide insights that are actionable and important to handle complexities in an economy in global space.

1.1 Graph Analytics: Introduction and Significance in Data Science

Graph analytics is all about understanding the relationships and interactions that exist in data through the studying of networks and network-based components. Hence, it has become one of the basic building blocks of data science, with a graphical view where modeling of data and exploratory analysis can be done easily. This project capitalizes on

graph analytics to understand and predict global crude oil trade, which is a major element in the world economy. Through this relationship in a graph—where the continents are the nodes and trade actions are the edges—understand and penetrate the hidden patterns and relationships that support better decisions.

1.2 Motivation and Problem Statement

This project is motivated by the need to understand the increasing complexity of global trade networks and the need for adequate analytical tools to understand them. In most cases, the value of foreign trade has been needed to further understand by the use of an appropriate and workable statistical tool since simple approaches cannot capture the complex relationships. The problem at hand is the need to accurately predict trade values and understand the underlying structure of the global crude oil trade network. This project addresses the problem by the development of a graph-based approach to model and forecast trade dynamics. The problem added to this, though, is transforming a large dataset spanning just over two decades into an appropriate form for graph-based analysis.

1.3 Project Objectives

The main objectives of the project are:

- Preprocess the global crude oil trade data, making it ready to use for graph analytics.
- Build a trade network graph where the continents act as nodes and trade actions as weighted edges.
- Use graph algorithms for analyzing the structure and dynamics of the developed trade network, focusing on the shortest path analysis and community detection techniques.
- Development of machine learning models for predicting future trade values on the graph using graph features and historical data followed by an evaluation of the same.
- The visualization of the graph and model output that brings about insights on the global trade network and the performance of machine learning models.

2. LITERATURE REVIEW

2.1 Overview of Existing Graph Representation Techniques and ML Algorithms for Graph Analytics

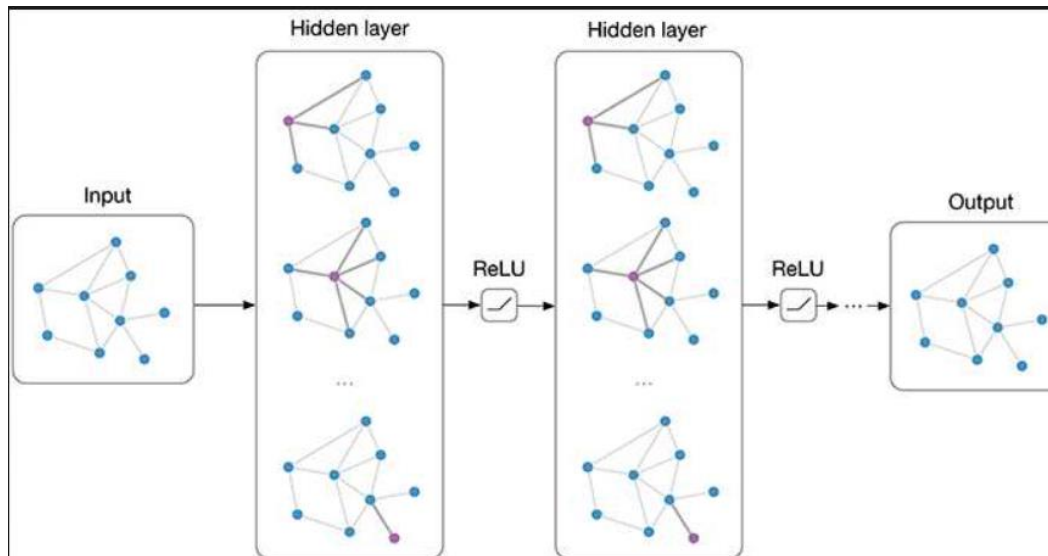


Figure 1: Graph Neural Networks Architecture

Representation of a graph is one of the techniques that works at the very base of the analysis and understanding of very complex data structures, where the relationships and interactions with the entities become more important than the entities themselves. Traditional techniques for such graph representations include adjacency matrices and lists[1]. More sophisticated techniques, such as graph embeddings, have been invented with the increasing complexity and scale of data and the importance of capturing structure and relations among the entities in a low-dimensional space.

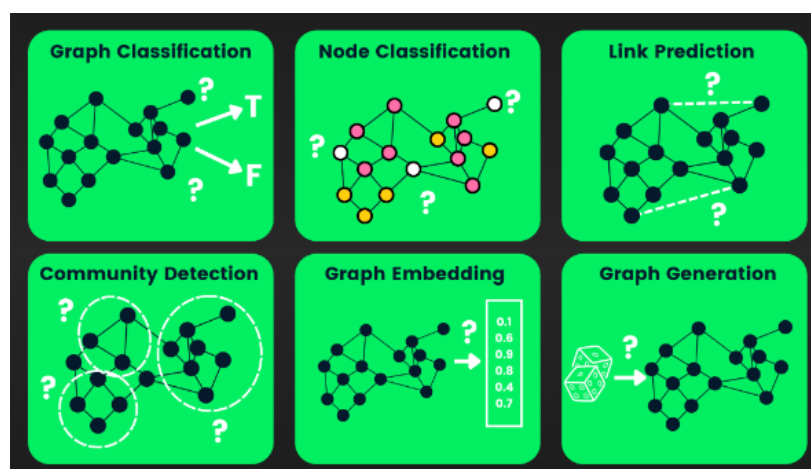


Figure 2: GCN operations

Graph embeddings, such as DeepWalk and Node2Vec, have grown in popularity due to their ability to represent nodes well in a continuous vector space and capture local and global properties of graphs. Among these methods, there have been numerous graphs, such as dimensionality reduction, when further applied in machine learning algorithms for node classification, link prediction, and clustering. These graph representation techniques, as briefly surveyed hereinabove, have facilitated powerful encodings in structural information, realized by ways of random walks and neighborhood aggregation, that enable more accurate graph analytics and at scale.

All the above representation techniques have been great in the world of graph analytics, but the Graph Neural Networks developed now take it to a whole new level. GNNs are an extension of traditional neural networks to graph-structured data and operate on nodes within a graph for direct graph processing. In such a manner, GNNs allow the aggregation of local neighbors' information by nodes through a message-passing mechanism in order to capture relations at higher abstract levels, which further need to be conserved for the solution of tasks like node classification, link prediction, and graph classification. By this virtue, therefore, GNNs are quite suited in any application where complex dependencies, as well as interactions, need to be modeled; these would include social network analysis, molecular biology, and recommendation systems.

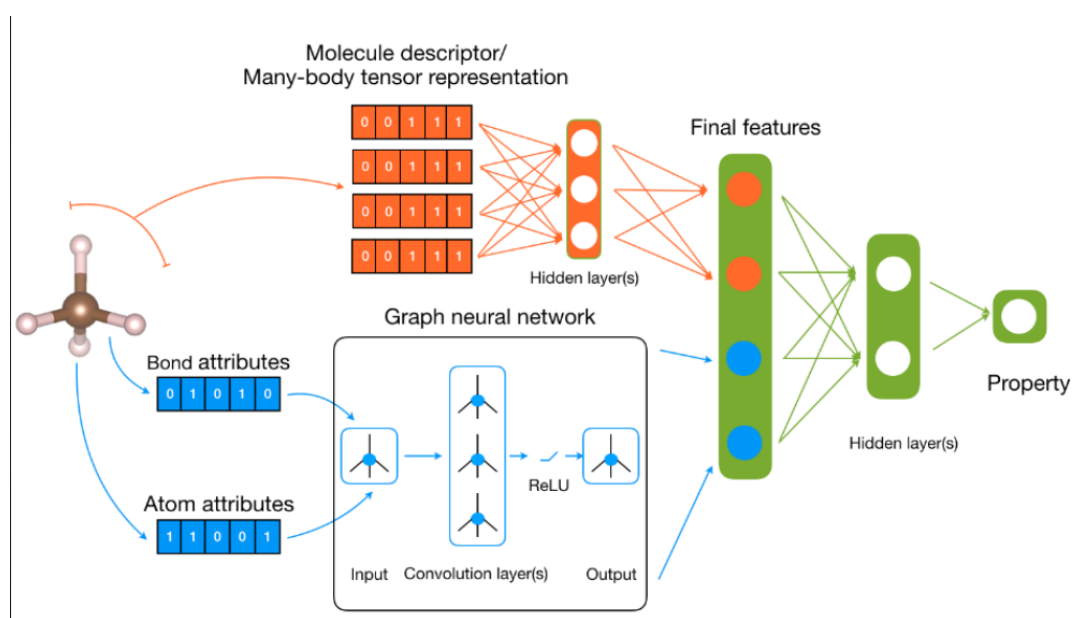


Figure 3: Applications of GNN

2.2 Review of Key Studies on GNNs, Graph Embeddings and Their Applications

Much recent progress has been made over the last decade in the development and application of graph neural networks and graph embeddings. Here, we point out certain important studies that have advanced our understanding of these techniques, particularly in multiple domains.

One of the seminal works in this field is the introduction of Graph Convolutional Networks, which generalizes convolution to graph data, enabling the effective extraction of local features from nodes and their neighbors. GCNs have thus gained wide acceptance for tasks such as node classification, where this has shown better results when the relationships are complex and nonlinear[2].

Another popular line of investigation is the use of attention mechanisms within Graph Attention Networks in order to assign different importance weights when aggregating the neighbors. This variant is shown to be particularly useful for tasks like knowledge graph reasoning, where most of the time, some relations are more significant than others. Experimental results showed that GAT achieved superior performance over classical methods in various applications and tasks, such as link prediction and recommendation systems, in which the quality of predictions requires fine-grained modeling of the graph structure.

Such application-specific architectures are the race lines in the application of GNNs and graph embeddings, for example, in the analysis of social networks for community detection and interaction prediction, or in molecular modeling, these methods highly boosted the prediction of molecular properties through molecular graph representations. The recommendation domain is where the versatility of such techniques has been well demonstrated. These techniques have been applied to recommendation systems to capture user-item interactions, thereby enabling more accurate and personalized recommendations[3].

2.3 Identification of Gaps in Existing Research

However, with such advanced techniques in graph representation and GNN, there are still several gaps on the research front today. Foremost among these issues is the scalability of the models. Even with neighbor sampling, a recently proposed technique, the difficulty of dealing with large-scale graphs remains one of the key issues that need to be resolved before practical application. This technique is usually computationally expensive in such a way that its application is usually limited to smaller graphs due to the design of the architecture.

Another area that will be developed further is the graph construction technique. The current methodologies often fail to depict such relationships and interactions, more so dealing with heterogeneous nature of data. It requires a stronger technique to deal with diverse data, to put that data into one universal graph structure that does increase the richness level of the analysis.

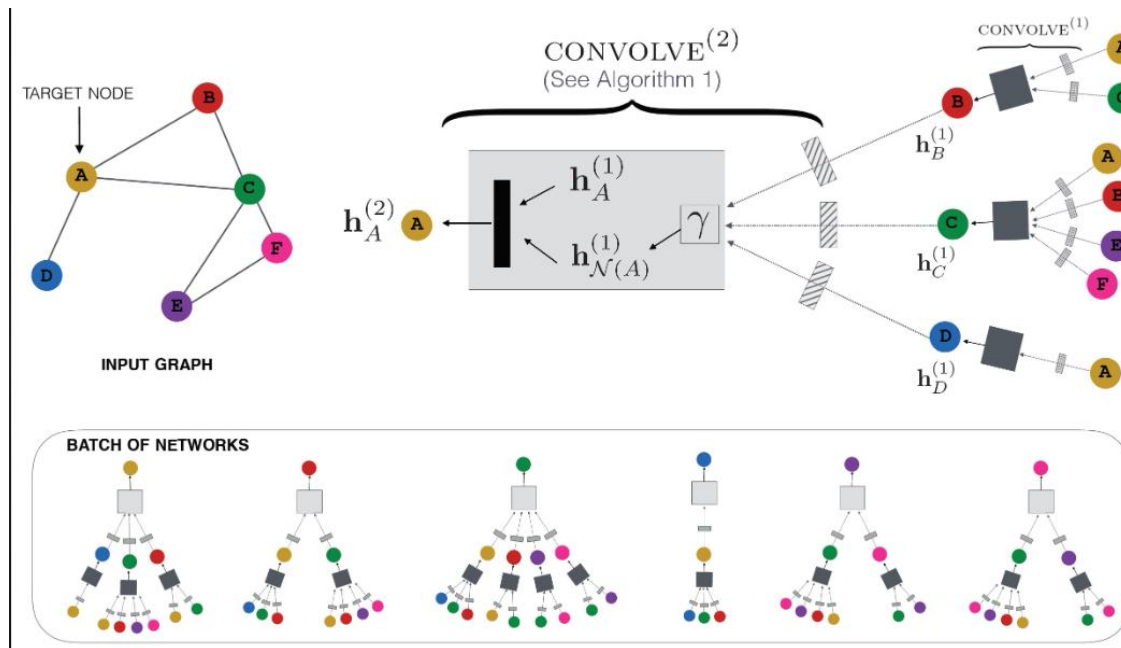


Figure 4: GNN for web scale recommendation systems

Moreover, the majority of the literature has focused on accuracy as the unique or main success metric, while other key goals, among which there are explainability, fairness, and user satisfaction, are commonly ignored. For example, in recommendation systems, such a focus on accuracy can introduce problems like filter bubbles, where users only get the kind of content they like, leading to their preferences being self-reinforcing. These aspects beyond accuracy are very important for the development of models that are more holistic and user-centric[4].

Lastly, GNNs are very strong in resisting attacks. Their robustness against adversarial in this domain is now becoming a budding concern. As more and more GNNs are applied to critical applications, their security against adversarial attacks is of high interest. The research in the current scenario, which is really very much in its infant stage, certainly calls for more detailed studies that can bring out some adversarial defense mechanism to be put into action to secure GNNs for application in real-world contexts.

Finally, although GNNs are very powerful and graph embeddings are a huge leap forward, it also may be about time to consider mechanisms of filling in these gaps for their potency to be manifested and them working across an array of complex and real-world environments[5].

3. METHODOLOGY

3.1 Data Collection and Preparation

3.1.1 Data Source

The dataset is titled "Global Crude Petroleum Trade 1995-2021.csv", sourced from a Kaggle platform where the community of developers, data scientists, and machine learning experts has immensely increased over the last few years. This platform allows loads of datasets to be accessed, and the one implied within the current text is for global crude petroleum trade transactions, between 1995 and 2021, and designed suitably for this kind of analysis for international trade patterns in the crude petroleum industry.

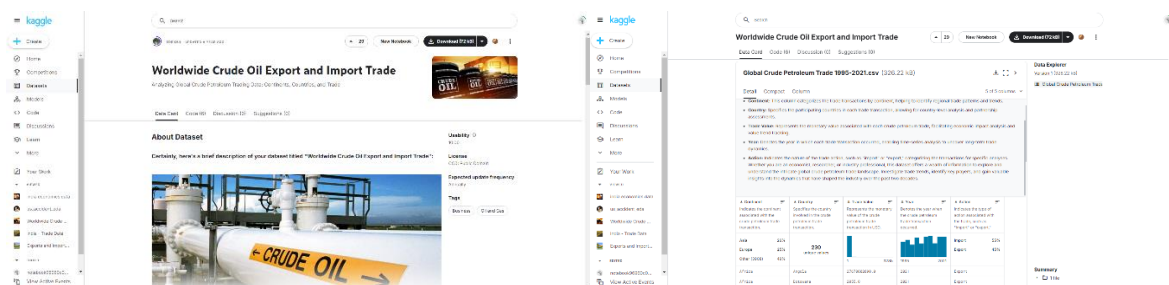


Figure 5: Kaggle view of the dataset

Source: <https://www.kaggle.com/datasets/toriqulstu/global-crude-petroleum-trade-1995-2021>

3.1.2 Data Cleaning and Preprocessing

Before analysis, the dataset must have been cleaned and preprocessed so that some sound analysis could be performed without bias, noise, and errors. The components of the processes implementing the steps for cleaning the data and preprocessing are:

1. Dealing with Missing Values:

- Missing values in the dataset were detected and treated. Depending upon the nature and importance of the missing data, the methods to be used include either imputation or deletion.

```
In [4]: df.describe()
```

	Trade Value	Year
count	7.925000e+03	7925.000000
mean	5.365169e+09	2008.589148
std	1.940226e+10	7.569435
min	1.000000e+00	1995.000000
25%	1.416080e+05	2002.000000
50%	8.549274e+07	2009.000000
75%	1.828030e+09	2015.000000
max	3.283380e+11	2021.000000

```
In [5]: df.isnull().sum()
```

	0
Continent	0
Country	0
Trade Value	0
Year	0
Action	0

Figure 6: Python code of Detection of missing values

2. Outlier Detection and Treatment:

- Statistical methods such as the Interquartile Range (IQR) and Z-score analysis were applied in outlier detection. Further, the outliers were checked if they actually signal some abnormality in the data, or these are the artifacts of feeding faulty data.



Figure 7: Python Implementation of Detection of outliers

```
In [8]: import numpy as np

# Detect outliers using the IQR method
Q1 = df['Trade Value'].quantile(0.25)
Q3 = df['Trade Value'].quantile(0.75)
IQR = Q3 - Q1

# Define the outlier range
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Filter out outliers
df = df[(df['Trade Value'] >= lower_bound) & (df['Trade Value'] <= upper_bound)]
```

Figure 8: Python Implementation of handling of outliers

3. Data Normalization and Standardization:

- Feature Engineering – Steps like data normalization or standardization, especially for a feature like Trade Value, were needed to make sure multiple different features mattered equally and the method should, therefore, yield better performance for the machine learning models.

4. Feature Engineering:

- Some of the available features have been used in deriving the new features that would cover a more complex relationship in the data. For instance, a more in-depth economic analysis is possible – feature: value of Export minus value of Import gives trade balance


```

In [63]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler

# Load the dataset
df = pd.read_csv('/content/Global Crude Petroleum Trade 1995-2021.csv')

# Print the first few rows and columns info to check data types
print(df.head())
print(df.info())

# Preprocess the dataset
# Convert 'Year' to a numeric format
df['Year'] = pd.to_datetime(df['Year'], format='%Y').dt.year

# Convert categorical variables to numeric using one-hot encoding
df = pd.get_dummies(df, columns=['Continent', 'Country', 'Action'], drop_first=True)

# Ensure 'Trade Value' is numeric
df['Trade Value'] = pd.to_numeric(df['Trade Value'], errors='coerce')

# Handle missing values if any
df = df.dropna()

# Define features and target
features = df.drop(['Trade Value'], axis=1)
target = df['Trade Value']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

# Standardize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

```

Figure 9: Python implementation , Data normalization and Standardization, feature engineering

3.1.3 Data Transformation

The dataset has undergone several transformations to make it friendly for graph analytics and machine learning tasks:

Categorical Encoding:

- The categorical columns 'Continent' and 'Country' were processed to be made useful within the machine learning workflows by encoding them using techniques like one-hot encoding or label encoding. As a result, it was made compatible with machine learning workflows.

Temporal Analysis Preparation:

- Time-based features are going to be developed using the 'Year' column for studying the temporal dependencies of global trade.

Graph Construction:

- The dataset was translated into the form of a graph structure, which contains the continents as nodes and their relationship in terms of trades (imports and exports) as

the edges. This translation is the necessary scaffold around which work can be done on the graph with algorithms like Dijkstra and Louvain.

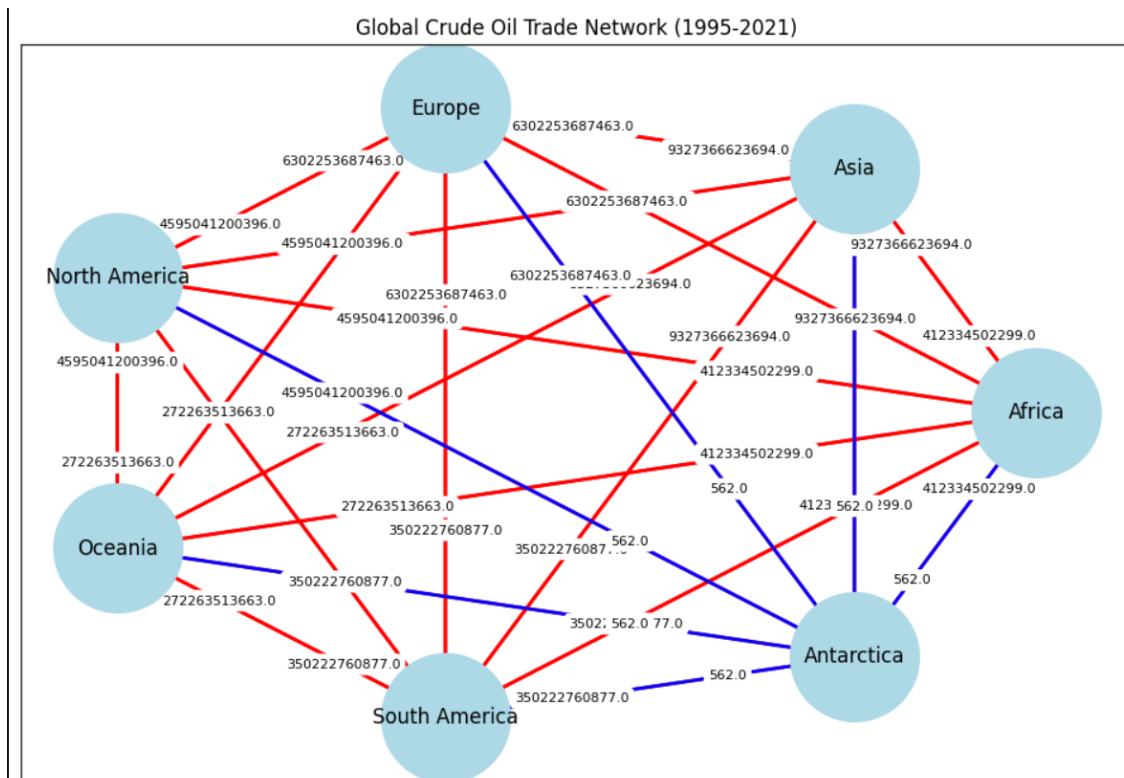


Figure 10: Graph created for the global crude oil trade between Continents

Here,

- Blue coloured edge represents export relationship
- Red Coloured edge represents import relationship

3.2 Data and Data Description

The following key features can be found in the dataset "Global Crude Petroleum Trade 1995-2021.csv," which depicts details of the global crude petroleum trade:

	A	B	C	D	E	F	G
1	Continent	Country	Trade Value	Year	Action		
2	Africa	Angola	27670002090	2021	Export		
3	Africa	Botswana	2055	2021	Export		
4	Africa	Cote d'Ivoire	444728243	2021	Export		
5	Africa	Cameroon	1865464729	2021	Export		
6	Africa	Democratic Republic of the Congo	581508635	2021	Export		
7	Africa	Republic of the Congo	1713917312	2021	Export		
8	Africa	Algeria	10695842162	2021	Export		
9	Africa	Egypt	3686004409	2021	Export		
10	Africa	Gabon	3606494908	2021	Export		
11	Africa	Ghana	3571526554	2021	Export		
12	Africa	Equatorial Guinea	2781491387	2021	Export		
13	Africa	Kenya	14624	2021	Export		
14	Africa	Liberia	150	2021	Export		
15	Africa	Libya	27004913406	2021	Export		
16	Africa	Morocco	115463681	2021	Export		
17	Africa	Mozambique	24226174	2021	Export		
18	Africa	Mauritania	274	2021	Export		
19	Africa	Mauritius	1	2021	Export		
20	Africa	Malawi	642	2021	Export		
21	Africa	Namibia	244	2021	Export		
22	Africa	Niger	86	2021	Export		
23	Africa	Nigeria	41841472432	2021	Export		
24	Africa	Rwanda	4452	2021	Export		
25	Africa	Sudan	395110945	2021	Export		
26	Africa	Senegal	19692639	2021	Export		
27	Africa	Sierra Leone	512	2021	Export		
28	Africa	South Sudan	455469279	2021	Export		
29	Africa	Eswatini	793	2021	Export		
30	Africa	Chad	1788077889	2021	Export		
31	Africa	Togo	39767544	2021	Export		
32	Africa	Tunisia	753554923	2021	Export		
33	Africa	Tanzania	297357	2021	Export		
34	Africa	Uganda	1122	2021	Export		

Figure 11: Excel preview of the dataset

3.2.1 Continent

Classifies each transaction in this particular trading feature by continent, giving a macro-level overview of regional trade patterns. It thus follows that analysis done via this particular feature will in a great way lead to an identification of which continents are the biggest players in crude petroleum and their regional dynamics that could actually impact how global markets behavior is.

```

import numpy as np

df=pd.read_csv('/content/global Crude Petroleum Trade 1995-2021.csv')
df.head()

Continent    Country    Trade Value    Year    Action
0    Africa    Angola    2.767000e+10    2021    Export
1    Africa    Botswana    2.050000e+03    2021    Export
2    Africa    Cote d'Ivoire    4.447282e+08    2021    Export
3    Africa    Cameroon    1.865465e+09    2021    Export
4    Africa    Democratic Republic of the Congo    5.815086e+08    2021    Export

Next steps: Generate code with df | View recommended plots | New interactive sheet

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7925 entries, 0 to 7924
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype  
---  --
 0   Continent    7925 non-null    object  
 1   Country      7925 non-null    object  
 2   Trade Value  7925 non-null    float64  
 3   Year         7925 non-null    int64  
 4   Action       7925 non-null    object  
dtypes: float64(1), int64(1), object(3)
memory usage: 389.7+ KB

```

Figure 12: Python implementation to understand the structure of the dataset

3.2.2 Country

The 'Country' column specifies the countries engaged in trading the transaction. This property offers the functionality where it is possible to analyze at the country level and, through this, check out the trade partnership and the individual country's role in the global crude petroleum market. It is the most essential column for identifying key exporting and importing countries.

3.2.3 Trade Value

The 'Trade Value' records the price in monetary terms for each of the transactions within a trade. This is one of the most important features within the economic analysis of the dataset to allow following the trend of volumes in trade over time and, at the same time, assessing quantitatively the financial deviations between countries and regions.

3.3.4 Year

The 'Year' feature represents the year of realization of any trade transaction. This feature can enable a time-series analysis, which will show long-term trends and changes in global crude petroleum trade over the analyzed period. This action also allows the study of the effects of significant global events on trade.

3.2.5. Action

The 'Action' column indicates whether the trade transaction was import or export. Binary classification on this field helps to focus the detailed analysis on trade flows and differentiates the countries that are net exporters from the ones that are net importers of crude petroleum.

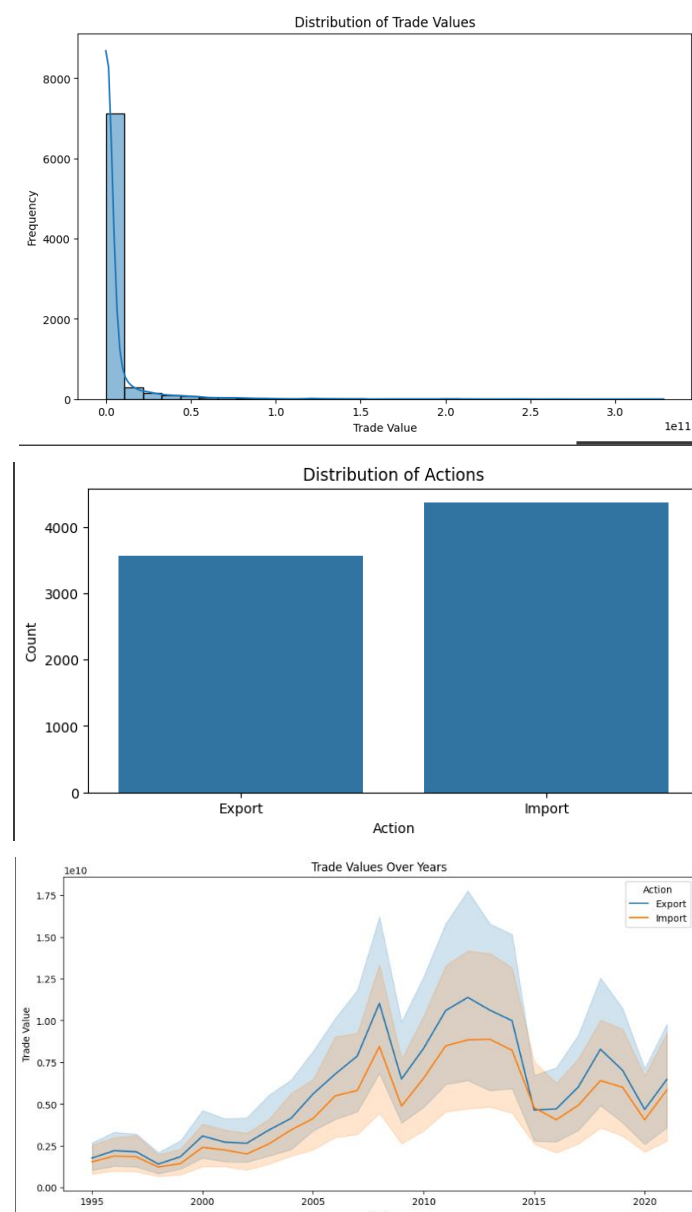


Figure 13: Graphical presentation of the features in the dataset

Interpretation of the Graphs:

The three graphs together reveal key insights into trade activities over time. The distribution of trade values is highly skewed, with the majority of transactions occurring at lower values and only a few at very high values. When comparing actions, imports slightly outnumber exports, though exports generally involve higher trade values. Over the years, both export and import values show a rising trend from the mid-1990s to around 2015, followed by some fluctuations, indicating significant growth in trade activities during that period, with exports consistently maintaining higher values than imports despite their lesser frequency.

3.2 Graph Representation and Modelling

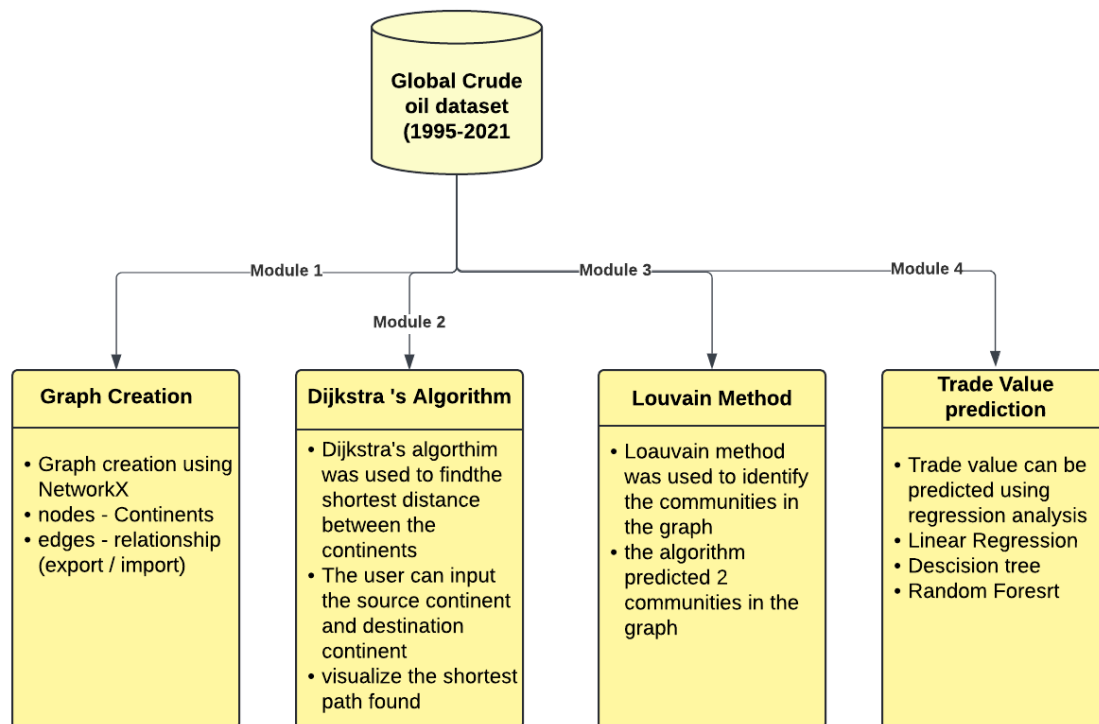


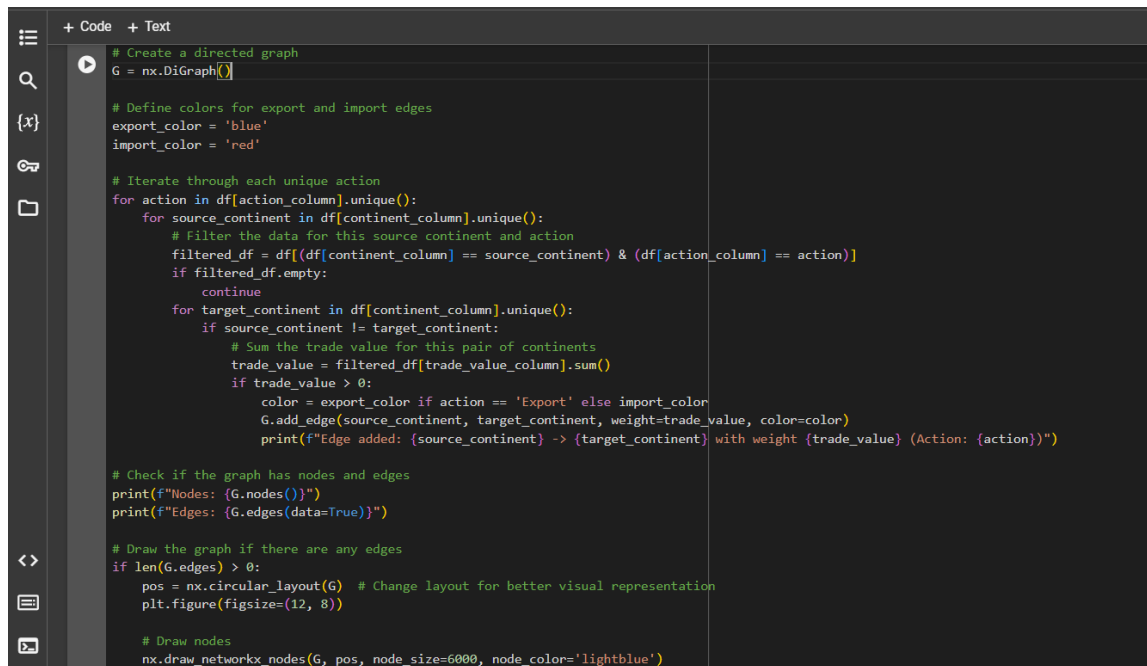
Figure 14: Flow of Operations perform

3.2.1 Graph Creation

One of the main steps in graph analytics is the creation of a graph, where a complex system is represented as set related nodes and edges. The nodes represent an entity, for example, people or countries, while the edges represent relationships or interaction between these entities. Graphs can be directed or undirected, the nature of the relationship that is represented by the edges. Graph, within the framework of trade networks, provides a visual and analytical tool for exploring how the entities overall are interlinked and sheds light on the patterns or clusters and the key players in the network. With the data structured as a graph, we are now able to use miscellaneous algorithms on it to extract the insights otherwise hard to perceive from raw data alone.

In this work, we have visualized the dataset "Global Crude Petroleum Trade 1995-2021" to observe the changing relational pattern of the trade flows between different continents. We have constructed node-and-edge graphs for the continents, with the emphasis put on the respective relations in between. In every case, the weight was the trade value, which helped in establishing how much trade was flowing between continents. For better clarity, the lines were

colored red for imports and blue for exports. This way of coloring enabled the obviousness in the visual analytics of the trade dynamics—whether there were active importers or exporters in a given region. To complete the dataset, it was made into a graph. Advanced graph algorithms like Dijkstra's and Louvain were used, thereby justifying an improved analysis and understanding of the global petroleum trade network.



```

+ Code + Text

# Create a directed graph
G = nx.DiGraph()

# Define colors for export and import edges
export_color = 'blue'
import_color = 'red'

# Iterate through each unique action
for action in df[action_column].unique():
    for source_continent in df[continent_column].unique():
        # Filter the data for this source continent and action
        filtered_df = df[(df[continent_column] == source_continent) & (df[action_column] == action)]
        if filtered_df.empty:
            continue
        for target_continent in df[continent_column].unique():
            if source_continent != target_continent:
                # Sum the trade value for this pair of continents
                trade_value = filtered_df[trade_value_column].sum()
                if trade_value > 0:
                    color = export_color if action == 'Export' else import_color
                    G.add_edge(source_continent, target_continent, weight=trade_value, color=color)
                    print(f"Edge added: {source_continent} -> {target_continent} with weight {trade_value} (Action: {action})")

# Check if the graph has nodes and edges
print(f"Nodes: {G.nodes()}")
print(f"Edges: {G.edges(data=True)}")

# Draw the graph if there are any edges
if len(G.edges) > 0:
    pos = nx.circular_layout(G) # Change layout for better visual representation
    plt.figure(figsize=(12, 8))

    # Draw nodes
    nx.draw_networkx_nodes(G, pos, node_size=6000, node_color='lightblue')
  
```

Figure 15: Python code for creation of network

3.2.2 Dijkstra's Algorithm

This is an algorithm that is quite commonly done to determine the length of the shortest paths between nodes in a weighted graph. This algorithm takes it tree node by node, determining possible smaller distance developments. It can visit any node with the shortest path known to it and this goes on until all nodes are visited or the shortest path to the destination node is found. Dijkstra's algorithm is an important algorithm in many network routing, GIS, and logistical applications that find the best route. The power in the optimization of transportation and communication networks, and similar other systems in which efficiency matters, comes from the ability of the already cost- or distance-minimizing algorithm to do so at this time.

For this very project, Dijkstra's algorithm was used to determine the shortest trade routes between continents from the viewpoint of the trade value of crude petroleum. Here the trade value was the weight of each edge and represented the cost side or benefit of trading between

any two continents. Application of Dijkstra's algorithm has enabled us to detect economically efficient trade routes of the global trade network, which could logically correspond to paths of least resistance or to the maximum profitability of the global trade network. This made knowing which trade routes were used most often pretty easy and which continents made the best partners when these analyses were done, these findings could be used further to make a strategy that optimizes the trade routes making global trade efficient.

```
def get_user_input():
    source_node = input("Enter the source continent: ")
    target_node = input("Enter the destination continent: ")
    return source_node, target_node

plt.figure(figsize=(12, 8))

# Draw nodes
nx.draw_networkx_nodes(G, pos, node_size=6000, node_color='lightblue')

# Draw edges
edges = G.edges()
edge_colors = [G[u][v].get('color', 'gray') for u, v in edges]
nx.draw_networkx_edges(G, pos, edgelist=edges, width=2, edge_color=edge_colors, arrows=True, arrowstyle='->')

# Highlight the shortest path edges
if path_edges:
    nx.draw_networkx_edges(G, pos, edgelist=path_edges, edge_color='green', width=4, alpha=0.6, arrows=True, arrowstyle='->')

# Draw labels
nx.draw_networkx_labels(G, pos, font_size=12, font_family="sans-serif")

# Draw edge labels (trade values)
edge_labels = nx.get_edge_attributes(G, 'weight')
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, label_pos=0.3, font_size=8, rotate=False)

plt.title(f'Global Crude Oil Trade Network with Highlighted Shortest Path ((Layout) Layout)')
plt.show()

# Main script
def main():
    # Get user input
    source_node, target_node = get_user_input()

    # Check if nodes exist
    if source_node in G.nodes and target_node in G.nodes:
        try:
            # Compute shortest path using Dijkstra's algorithm
            shortest_path = nx.shortest_path(G, source=source_node, target=target_node, weight='weight')
            print(f"Shortest path from {source_node} to {target_node}: {shortest_path}")

            # Convert the shortest path to edges for highlighting
            path_edges = list(zip(shortest_path, shortest_path[1:]))

            # Plot the graph with the shortest path highlighted
            plot_graph_with_highlight(G, path_edges=path_edges)
        except nx.NetworkXNoPath:
            print(f"No path found between {source_node} and {target_node}.")
        else:
            print("Invalid source or destination node.")

    if __name__ == "__main__":
        main()
```

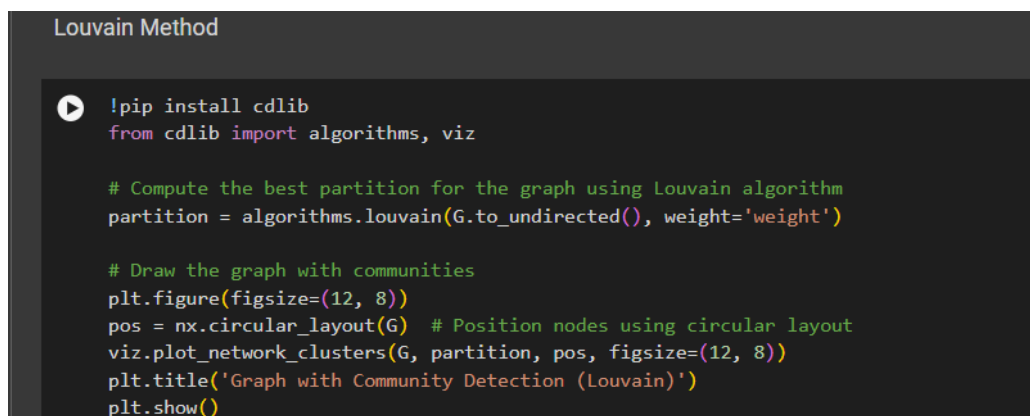
Figure 16: Python implementation of Dijkstra Algorithm

3.2.3 Louvain Algorithm

The Louvain algorithm is one efficient method to detect communities in large networks. It maximizes the function called modularity, which maximizes this for counting the number of edges in communities as opposed to out of them. The existing algorithm proceeds in two main phases. In the first one, each node belongs to a community of one, and then nodes are reallocated to increasing modularity. The second phase involves the aggregation of nodes that belong to the same community into one node, which forms a more condensed network. The

last-mentioned process is iterative by nature, and this cycle repeats until modularity no longer effectively increases. The Louvain algorithm finds especially widespread use in revealing latent structure of networks in social network analysis, biology, marketing, and in other fields where the very set of related clusters is not visible.

In the rough oil global trade network graph, the Louvain algorithm was applied. It detected hemisphere communities for which intra-community trade intensity is higher than inter-community trade. This can be interpreted as a certain clustering of regions, which have formed tight trading groups probably due to geographical proximity, shared economic interests, or historical trading relationships. By discovering communities, we were able to glean insight regarding the potential underlying structure of the world petroleum trading network. Final results obtained by the Louvain algorithm gave us a clearer picture of regional trade blocs and alliances, configuring the most important players and potential alliances for cooperation for which to offer competition. This would be especially important for policymakers and economists in understanding global dynamics in trade and the effect they may have on regional economies.

The image shows a Jupyter Notebook interface with a dark background. The title of the notebook is 'Louvain Method'. Below the title, there is a code cell containing Python code. The code starts with installing 'cdlib' using '!pip install cdlib' and importing 'algorithms' and 'viz' from 'cdlib'. It then uses 'algorithms.louvain' to compute the best partition for a graph 'G'. Finally, it uses 'viz' to plot the network clusters with a circular layout and shows the plot.

```
Louvain Method

!pip install cdlib
from cdlib import algorithms, viz

# Compute the best partition for the graph using Louvain algorithm
partition = algorithms.louvain(G.to_undirected(), weight='weight')

# Draw the graph with communities
plt.figure(figsize=(12, 8))
pos = nx.circular_layout(G) # Position nodes using circular layout
viz.plot_network_clusters(G, partition, pos, figsize=(12, 8))
plt.title('Graph with Community Detection (Louvain)')
plt.show()
```

Figure 17: Python implementation of Louvain method

3.2.4 Regression Analysis

Regression analysis is one of the most powerful statistical methods in investigating the relationship that exists between the dependent and an independent variable. The main objective of the regression is for modeling the relationship in making predictions or understanding some factors' effect. There are three regressions used for this project: Linear Regression, Decision Tree Regression, and Random Forest Regression. All three have their own advantages and have been recommended for various kinds of data and relationships.

Linear Regression:

It is the most fundamental and applied regression analysis technique. Linear regression expects a linear relationship between a dependent variable and one or more independent variables. It tries to make a line—which will turn into a hyperplane in more than one dimension—such that the sum of the squares of the differences between the observed and the fitted values is smallest.

Linear regression was used to model the relationship in such a project between the trade value of crude petroleum (dependent variable) and independent variables like Continent, Country, Year, and Action (Import or Export). Since linear regression is simple and interpretable, it is a good starting model for our analysis. Through linear modeling, we were able to put a quantitative measure of the effect of each of the predictors on the trade value, hence gaining insight into the way different factors shape global petroleum trade. Even though it is quite simple, some linearity restrictions have been imposed by linear regression, so they are not all held in complex trade networks. Hence, we explore more advanced models like the Decision Tree and Random Forest to capture potential non-linear relationships.

Decision Tree Regression:

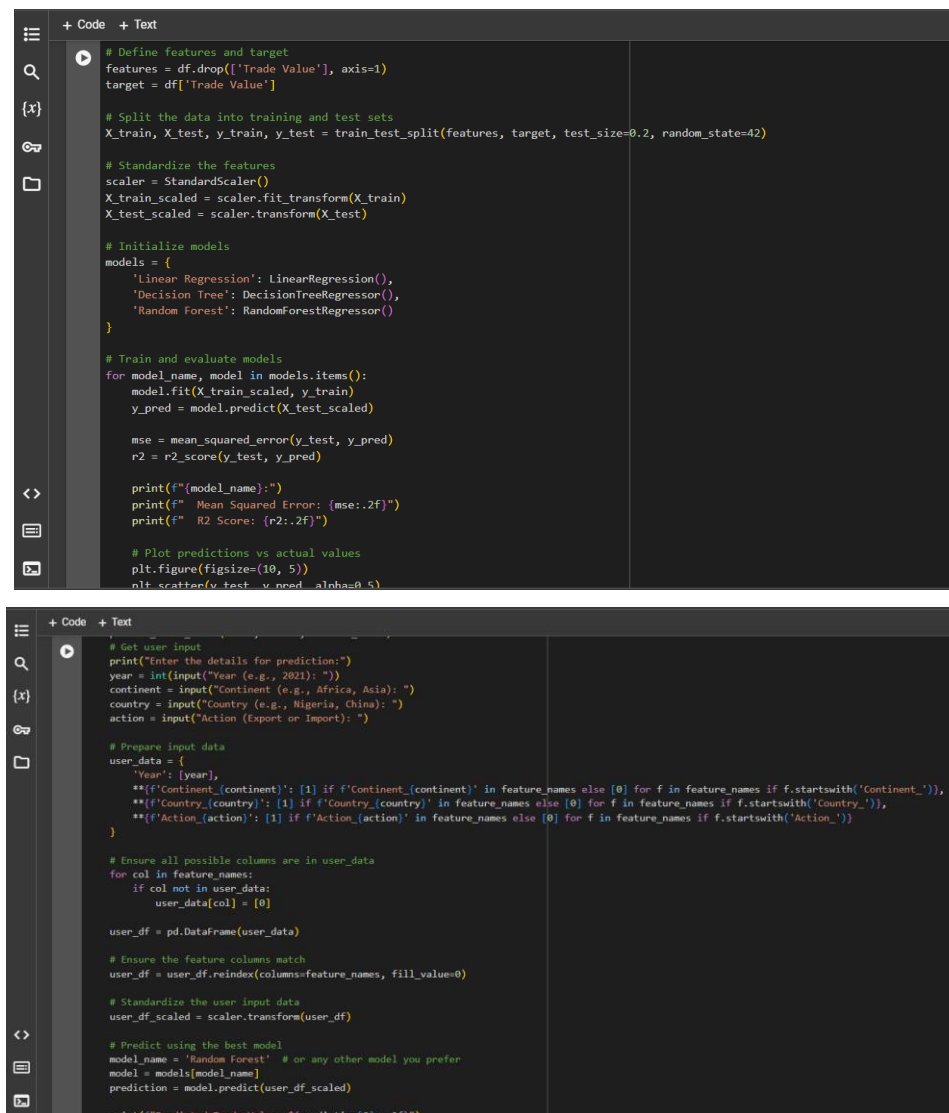
This is a non-parametric regression that establishes the relationship between the target variable and predictors by splitting the data at every step into subsets based on some conditions. In this process, at each split, the algorithm selects the feature and the threshold that maximally decreases variance, or another metric selected, in the target variable. At each point when the iteration is done, the algorithm stops based on predefined stopping criteria or if further splitting of the current model does not strongly improve the target prediction of the model. This results in a structure of tree-like nature wherein a leaf node passes the predicted value.

The predictive equation utilized in this work was Decision Tree Regression, in consideration of catching probable nonlinear relationships between the predictors and trade value. Decision Tree Regression is able to model variable interactions in a very complex manner; it is more flexible than Linear Regression in this way. For example, the result would come out like some countries or continents significantly vary in trade behaviors, whether based on the year or that type of trade action taken, either import or export. Visualizing the decision tree helped us see how the various factors interacted with one another in determining the trade value. However, Decision Trees can be very prone to overfitting, especially in large feature dimension datasets, for which another ensemble method, namely Random Forest, was also employed.

Random Forest Regression:

The Random Forest Regression is a kind of ensemble learning technique, which uses a lot of Decision Trees to achieve better prediction and limit overfitting for models. Then, with the algorithm, during the training of the ensemble of Decision Trees, each tree will be fitted on a bootstrap sample of the data: a random subset of data points chosen with replacement. Also, while performing node splitting in the making of a Random Forest, consideration is given only to a random subset of all features, ensuring further diversity among such trees. Final prediction by the random forest model is obtained by averaging these outputs of trees, the result indeed gives a model with a very low variance compared to a single Decision Tree model.

In this project, trade value for crude petroleum was modeled using a Random Forest Regression. Random Forest captured the high complex relationships within the data by aggregating the predictions of multiple Decision Trees. The risk of overfitting is thus mitigated by ways that sometimes become vulnerable to the use of a single Decision Tree. This makes the algorithm, almost ideal for this study given the ability to deal with large datasets and model non-linear interactions. The model itself was judged against the benchmark of Linear Regression and Decision Tree Regression. It became clear from the results of the Random Forest—with superior predictive accuracy and generalizability—that it gave the most reliable forecasts concerning trade value in the future. This yields further affirmation in the predictions and a deeper understanding of the driving forces within global petroleum trade.



```

+ Code + Text

# Define features and target
features = df.drop(['Trade Value'], axis=1)
target = df['Trade Value']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

# Standardize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Initialize models
models = {
    'Linear Regression': LinearRegression(),
    'Decision Tree': DecisionTreeRegressor(),
    'Random Forest': RandomForestRegressor()
}

# Train and evaluate models
for model_name, model in models.items():
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)

    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print(f"{model_name}:")
    print(f"  Mean Squared Error: {mse:.2f}")
    print(f"  R2 Score: {r2:.2f}")

# Plot predictions vs actual values
plt.figure(figsize=(10, 5))
plt.scatter(y_test, y_pred, alpha=0.5)

```

```

+ Code + Text

# Get user input
print("Enter the details for prediction:")
year = int(input("Year (e.g., 2021): "))
continent = input("Continent (e.g., Africa, Asia): ")
country = input("Country (e.g., Nigeria, China): ")
action = input("Action (Export or Import): ")

# Prepare input data
user_data = {
    'Year': [year],
    **{'f' + f'Continent_{continent}': [1] if f'Continent_{continent}' in feature_names else [0] for f in feature_names if f.startswith('Continent_')},
    **{'f' + f'Country_{country}': [1] if f'Country_{country}' in feature_names else [0] for f in feature_names if f.startswith('Country_')},
    **{'f' + f'Action_{action}': [1] if f'Action_{action}' in feature_names else [0] for f in feature_names if f.startswith('Action_')}
}

# Ensure all possible columns are in user_data
for col in feature_names:
    if col not in user_data:
        user_data[col] = [0]

user_df = pd.DataFrame(user_data)

# Ensure the feature columns match
user_df = user_df.reindex(columns=feature_names, fill_value=0)

# Standardize the user input data
user_df_scaled = scaler.transform(user_df)

# Predict using the best model
model_name = 'Random Forest' # or any other model you prefer
model = models[model_name]
prediction = model.predict(user_df_scaled)

print(f"Predicted Trade Value: ${prediction[0]:.2f}")

```

Figure 18: Python implementation of regression models

3.3 Results and Analysis

3.3.1 Presentation of Model Performance Metrics

Graph Creation:

Results:

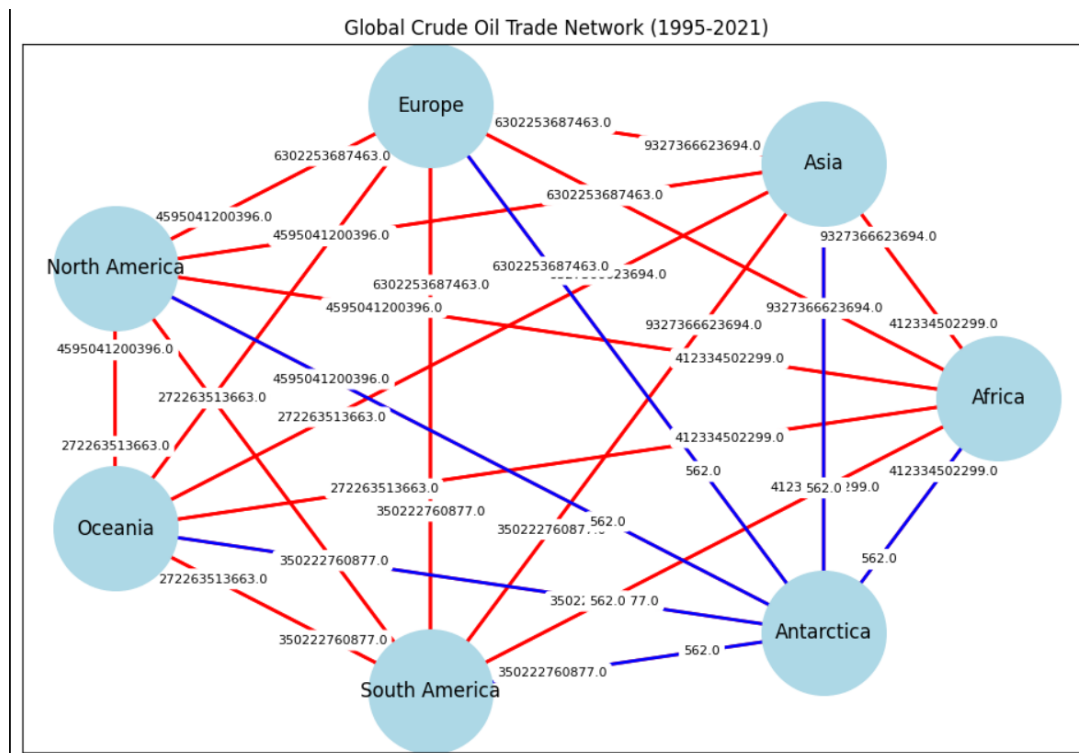


Figure 19: Graph created

Global crude petroleum trade was able to make one effective network graph between the years 1995 to 2021. The continents are represented with the nodes created. The trade relationships either import or export, are shown with the nodes in the edges. The graph is color-coded so that it is easy to differentiate between and represents import or export action. The graph perfectly visualized the trading patterns and allowed an intuitive understanding of how various regions interacted with one another over the years.

Analysis of the graph gave interesting insights about global traded in crude petroleum. Important trading hubs, continents that had a large number of connections both for import and export, could be easily recognized. The density of the connections indicated regions with high trade activity, while lighter regions implied that those regions were not involved in global trade activity as much. The structure of the graph allowed one to pinpoint the key players in the

petroleum marketplace and see, for example, the up-strengthening or weak-strengthening of some regions in global trading activity during that two-decade period.

Dijkstra Algorithm:

Results:

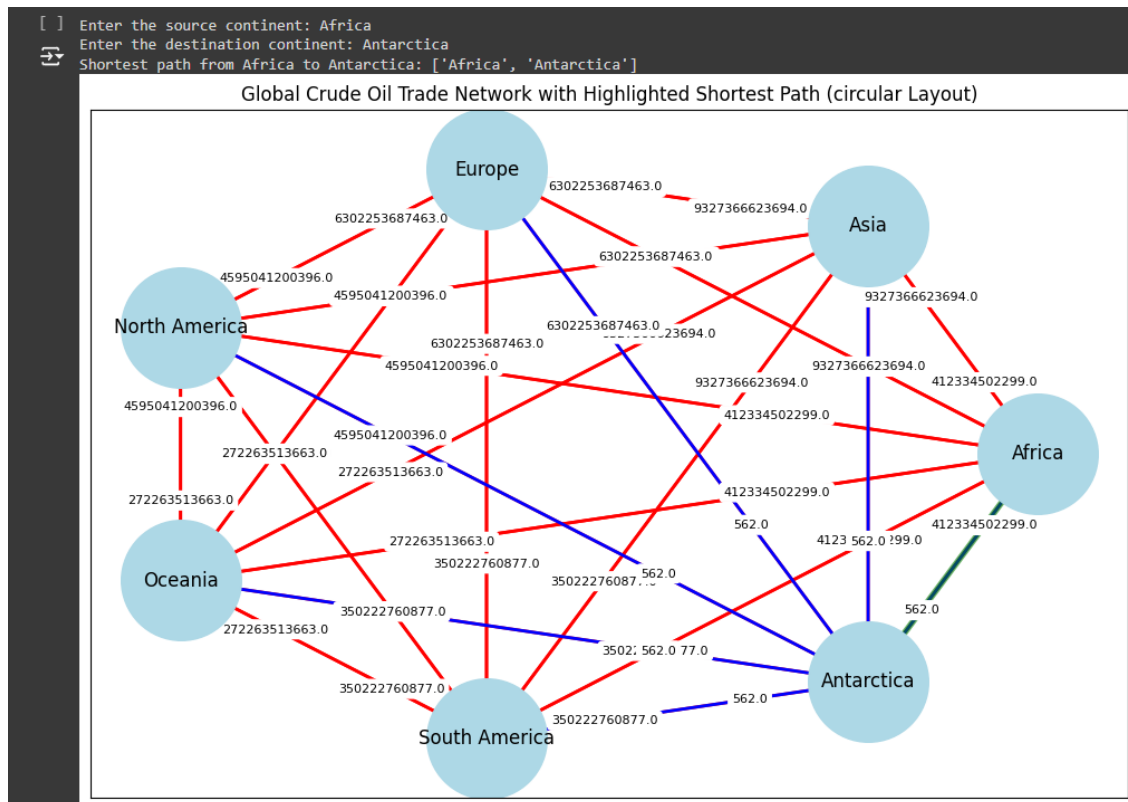


Figure 20: Highlighted graph after finding shortest path

The Dijkstra algorithm was used to calculate the shortest path to connect all the nodes: it gives the shortest path distances between all nodes, representing the optimal trade routes where the edge weight of each country pair in the respective graph was the value of trade between those two countries. It supplies the result of the most optimal trade routes by emphasizing the pathways for crude petroleum movement between continents.

From figure 20, we can understand that the program allows users to enter the source and destination of their choice and the program generates a graph highlighting the shortest path. In this case the entered source and destination are “Africa” and “Antarctica” respectively. The results from the Dijkstra algorithm revealed short paths connecting multiple other regions, which would be named as highly important trade hubs and, thereby, assumed a central position within the global supply chain. But the finding also highlighted the potential bottlenecks where longer paths indicated less efficient trade routes that could be optimized. Being aware of such

paths served to focus attention on strategic nodes within the system where trade policies or infrastructure investment would have maximum impact.

Louvain Algorithm:

Results:

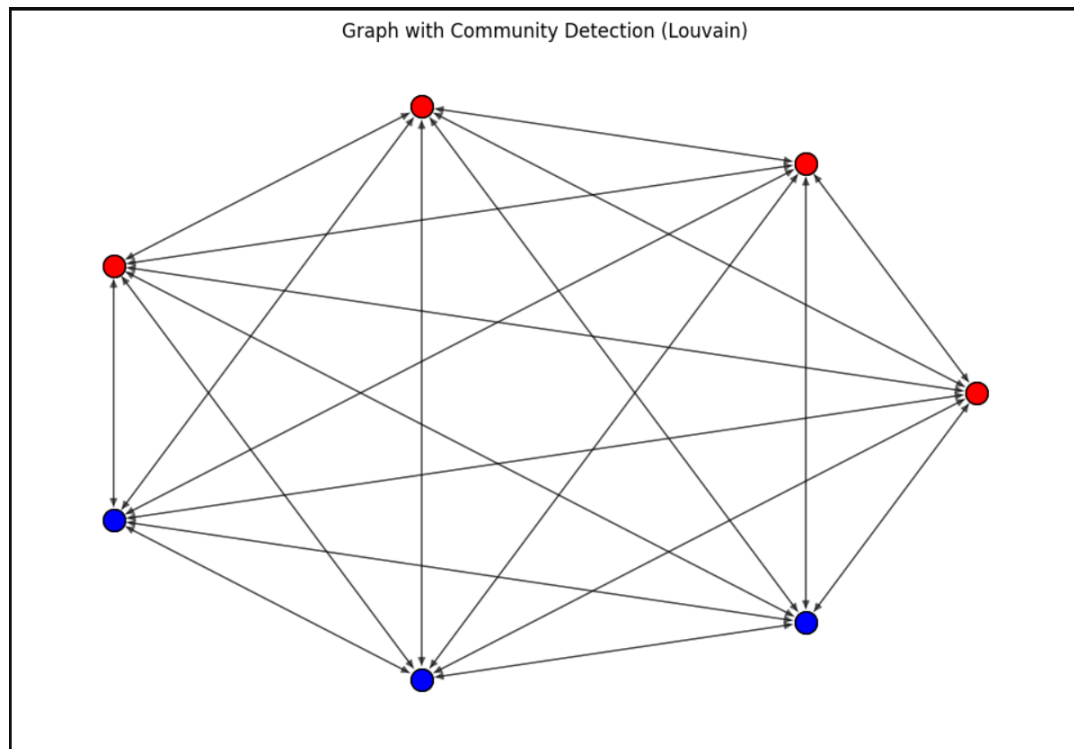


Figure 21: Graph highlighting the communities identified

The Louvain algorithm was applied to detect communities or clusters of closely connected nodes in the graph, meaning groups of continents with strong trade relationships. Obviously, it gave modules of the global trade network in which continents group up, having more intense trade activities with each other than with the rest of the world. Thus, communities clearly gave a picture of regional trade blocs and alliances. From figure 21 we can understand that Louvain method identified 2 communities in the graph, here the nodes represent the continents.

Analysis of the results of community detection with the Louvain algorithm: regional trade blocs, as in Europe, Asia, or the Americas, present themselves. In other words, there are communities among continents engaged in high trade with one another, which reveals the regionalization of the global petroleum market. In the identification of such communities, insight is derived on regional trade policies, economic alliances, and even hints regarding future trade accords. The results also indicated that the changes in trade dynamics across time seemed

to reflect some regions with strengthened internal trade networks and others with a more global integration.

Linear Regression:

Results

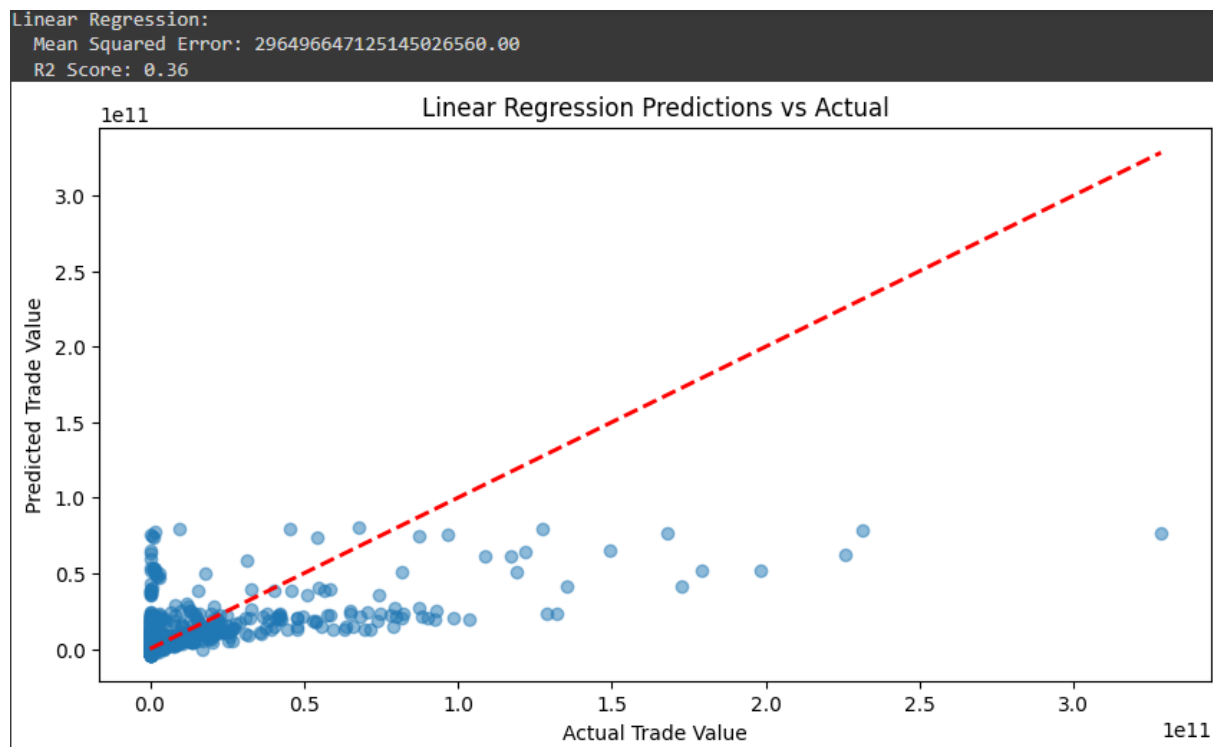


Figure 22: Results of Linear Regression

The Linear Regression model gave a baseline prediction for the trade value of crude petroleum using features such as continent, country, year, and action. Given the coefficients of the model, the model's strength and the direction of the relationship with the predictor on the trade value were determined. The correctness of the model was moderate, and the trade value had some significant predictiveness from the predictors. On training, the Linear Regression model returned an MSE of 296,496,647,125,145,026,560.00, and an R2 score of 0.36. The very high MSE reflects huge deviations between predicted and actual trade values, thus leaving doubts over the precision in the predictions from the model. An R2 score of 0.36 denotes that approximately 36% of the variance in values of trade is explained by the model; it clarifies that by catching underlying tendencies, it still leaves a large part of the variance unexplained. This is a clear hint that linear regression would not be most appropriate for this data set, and further model tuning or alternative approaches would be required.

The initial results from the Linear Regression analysis demonstrated insights into how the various factors affected global trade in crude petroleum. Some of the linear relationships could be captured with this model; however, it was clear that more sophisticated models were required in order to capture non-linear patterns existing in the trade data. The coefficients of the model underlined the comparative importance of different predictors, like the year of transaction and country, in explaining trade values. However, due to cutoff limitations in handling nonlinear relationships and variable interactions, necessitated collaborating with more advanced regression techniques.

Decision Tree Regression:

Results:

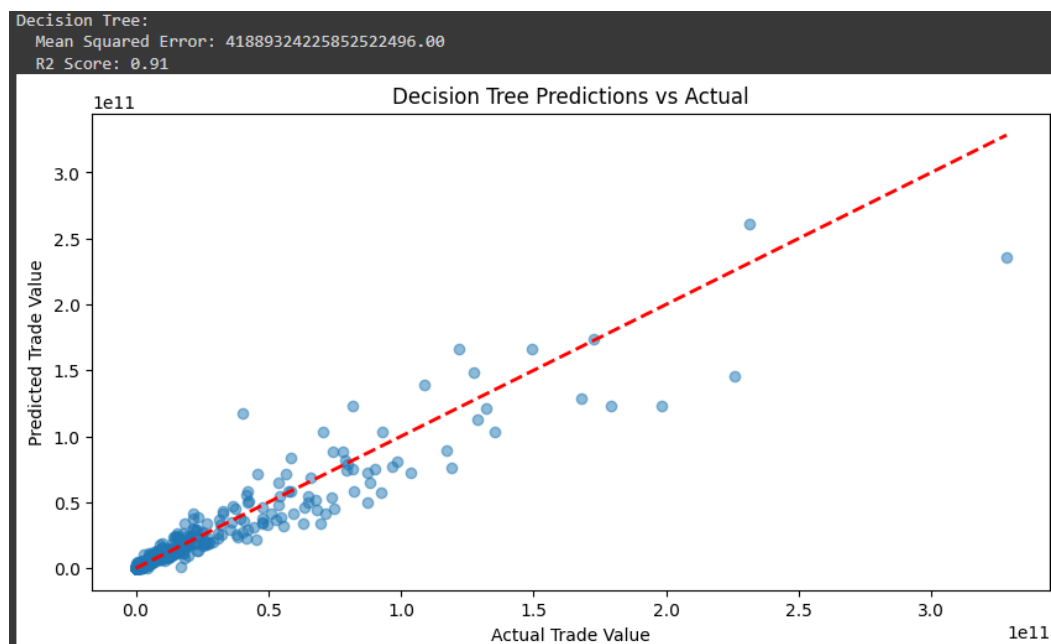


Figure 23: Results of Decision Tree

The Decision Tree Regression model provided more flexibility when predicting trade values because it included non-linear relationships and interactions between predictors. The output model of the decision tree, which divided the data based on the most influencing features, was based on a hierarchical structure of the model showing how the trade values would range under specific conditions. The values of Mean Squared Error obtained by the Decision Tree model are 41,889,324,225,852,522,496.00, while the R2 Score is 0.91. The relatively lower MSE indicates that the Decision Tree model does a better job compared to the Linear Regression model in its predictions. The high score of R2 itself, 0.91, indicates that the model explains

around 91% of the variances in trade values, hence a very good fit with an effective capture of the underlying patterns. This means the Decision Tree model has much better accuracy and explanatory power for this exact data set.

The Decision Tree Regression model showed complex cultural interactions among the features that a simple Linear Regression model could not represent. For example, using the model, it is possible to determine in which continent or country some special condition is checked for having a high or low value of trade. The tree structure made these conditions clearly visible and thus improved the interpretability of the results. However, the model was prone to overfitting, more so with the deep trees, which did, in fact, reduce the generalizability of the predictions. This made one have to use the Random Forest Regression model to control overfit and improve prediction.

Random Forest Regression:

Results:

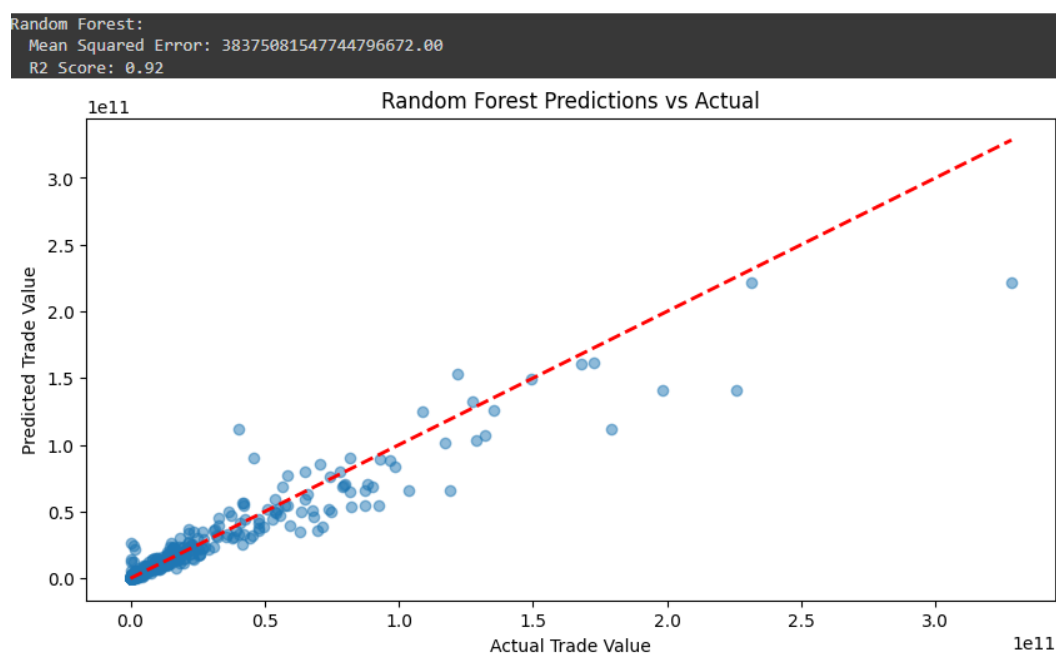


Figure 24: Results of Random Forest

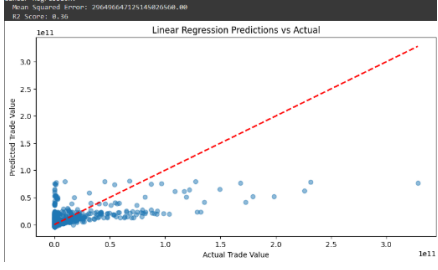
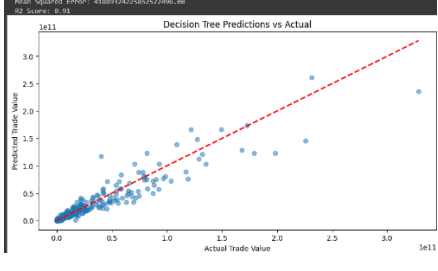
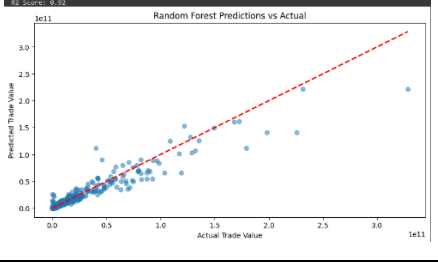
The main advantage of the Random Forest model is, therefore, to produce an improvement in predictive accuracy by averaging the results of many trees, every one of which should have been de-correlated by training each tree on a different subsample of the data. The model helped obtain strong predictions for the trade values with far less variance and better generalization compared to the predictions made using the single tree model. Similarly, the Random Forest

model resulted in a Mean Squared Error (MSE) of 38,375,081,547,744,796,672.00 and an R2 Score of 0.92. As can be deduced from the MSE value, the predictions from the Random Forest model are reasonably correct, with the error values distributed lower compared to the Linear Regression and Decision Tree models. The corresponding R2 Score of value 0.92 is indicative of the fact that this type of model describes about 92% of the variance in trade values, meaning high goodness of fit and moderate to peak complex patterns within the data. This indicates that the Random Forest model performs way better on predicting trade values and explains the most variations than Linear Regression and the Decision Tree model.

The Random Forest model applied in the performance of the Random Forest Regression model had better performance in the prediction of trade values for the global crude petroleum market. The ensemble method picked up on a subset of these complex, non-linear relationships in the data and decreased the dangers of any overfitting. It was also visible which features were of high importance regarding critical predictors verifying the importance of certain countries, years, and actions in determining trade values. The Random Forest model would be the best model for this project, where large datasets could be handled and concrete prediction can be given to provide substantial insights into future trade trends.

3.3.2. Comparative Analysis of Models

Table 1: Comparative Analysis of the Models applied for trade value predictions.

Model	R-score (%)	MSE	Graph
Linear Regression	36	296,496,647,125,145,026,560.00	
Decision Tree	91	41,889,324,225,852,522,496.00	
Random Forest	92	38,375,081,547,744,796,672.00	

A comparative analysis is conducted to understand the strengths and weaknesses of each model. The Random Forest model outperformed the Linear Regression and Decision Tree models, particularly in capturing non-linear relationships in the data. The analysis discusses the trade-offs between model complexity and predictive accuracy, providing insights into why the ensemble method was more effective.

The important implication of the comparison of the three regression models—Linear Regression, Decision Tree, and Random Forest—was making a difference between the performance of RS and MSE.

In regard to **Linear Regression**, it had an insignificant R-score of 36%, with a high value for MSE at roughly 296.5 trillion. This indicates that the model has the lowest capacity for explaining the variation in the trade values and the worst prediction. The high value of MSE shows a great level of prediction error; that is, the model is far from being optimal to establish the structures within the dataset, almost certainly due to its linearity assumptions that do not suit the dataset's non-linear relationships.

Decision Tree: far better than Linear Regression with a score of 91% and a much, much lower MSE at approximately 41.9 trillion. It was very high, and the upshot of this is that it showed up to 91% of the variance in the trade values using the Decision Tree, thereby much better fitting. The lower value of MSE foretells that the predictions by the Decision Tree model will be rather accurate; even the nonlinear pattern in data can be captured much better compared to Linear Regression. Despite the high R-score, it is slightly outperformed by the Random Forest model.

Out of three models, **Random Forest** gave the highest R-score of 92% and the lowest MSE of about 38.4 trillion. The high R-squared signifies 92% of the variance is explained by the Random Forest model, which means really high predictivity and way better fit as compared to Linear Regression and Decision Tree. It is indicative of the fact that Random Forest effectively deals with the complexities and the many non-linear relations, as shown by the best accuracy in prediction of the smallest MSE. It is an ensemble model involving combining predictions from many decision trees; hence, high performance and reliability.

Among applied models, the Random Forest model outperformed the Decision Tree and Linear Regression in predictive accuracy and model fit while capturing subtle details in the dataset.

3.3.3 Interpretation of Results

The results are interpreted in the context of the project's objectives. This section discusses how the findings align with the original problem statement, providing insights into the dynamics of global crude oil trade. The implications of the model predictions for real-world applications, such as economic forecasting and policy-making, are also explored.

The integration of the graph development at the back end level simultaneously with Dijkstra optimization algorithm and Louvain community detection and the regression analysis has, in effect, together, contributed to the fulfilment of the objective of the project work and offered a comprehensive solution to the problem statement. With substantial referencing of fished out

global petroleum trade network and with the implication of Dijkstra optimization algorithm, the aim is to ascertain the shortest path opted among various analytical zones scattered throughout the continents for the regional critical trade route identification and for the logistic optimization process. The community detection method sheds light on groups of related continents, pointing toward the configuration of regional trade and also zones on which more strategy can be focused. This graph-based analysis highlights the key patterns in the world trade network and facilitates predictive model development.

Here is the analysis using regression: Linear Regression, Decision Tree, and Random Forest to rework the trend in value of trade. With the best performance, the Random Forest model accomplished an R-score of 92% and the lowest MSE, which proved its efficiency in capturing the complexities of the trade value predictions. In all, the approach satisfied the purpose of the project: improvement in trade value forecasting through machine learning and graph analytics. This project thus merges graph-based insights with powerful predictive modeling to provide the detailed levels that are required to elucidate global crude petroleum trade, thus providing stakeholders in the oil industry with implementable strategies.

3.4 Visualization of Results

3.4.1 Visualization Tools and Techniques

The project utilized tools such as NetworkX and Matplotlib to visualize the trade network and model outputs. This subsection describes the tools and techniques used to create these visualizations, including graph layouts (e.g., circular, spring) and edge coloring based on trade actions. The visualizations played a crucial role in understanding the structure of the graph and the results of the machine learning models.

3.4.2 Model Interpretability and Decision-Making Process

Visualizations, such as global crude oil trade network and community detection maps, incorporated in this project go a long way in enhancing any process of decision making by converting raw data into easily understandable and actionable information. Specifically, a global trade network visualization will provide information on how crude oil flows among continents to enable stakeholders to identify major trade routes, regional trade imbalances, and key trading partners. As an illustration, countries with dense trade connections or high trade intensities can be pinpointed as strategic areas for business expansion or developing an optimal

supply chain. This way of visualizing trade relationships can help decision-makers identify key trends instantly, such as promising market entries or potential trade bottlenecks.

Identified cluster of continents, based on their trade relationships, after the community detection visualization using the Louvain algorithm is explained in the following subsections. It identifies regional players that are very closely integrated, and therefore might be useful for understanding regional trade dynamics in order to deduce focused trade policies or partnerships. For instance, if a cluster of continents is found to be trading intensively with one another, that would speak to a regional trade alliance or economic bloc that is conditioning any choice regarding trade agreements or strategic investments in infrastructure.

These are results obtained with predictive models of linear regression, decision tree, and random forest for future trends in the trade variable values. The linear regression model allows the establishment of a baseline for the trade value prediction models, while both the decision tree and random forest deliver more accurate and robust forecasts due to the attainment of higher R-squared scores—it enabled good performance with decreasing mean squared errors. These predictions are very useful in strategic planning, as they enable stakeholders to have a view of the trade trend value, meaning that the best outcomes can be anticipated, resources planned, and how any modification of business strategies can be adjusted appropriately. In this way, therefore, stakeholders are able to deal with risks, capitalize on emerging opportunities, and align their overall strategies with anticipated market changes through ignited predictive insight.

4. FINDINGS AND INTERPRETATIONS

4.1 Synthesis of Key Findings

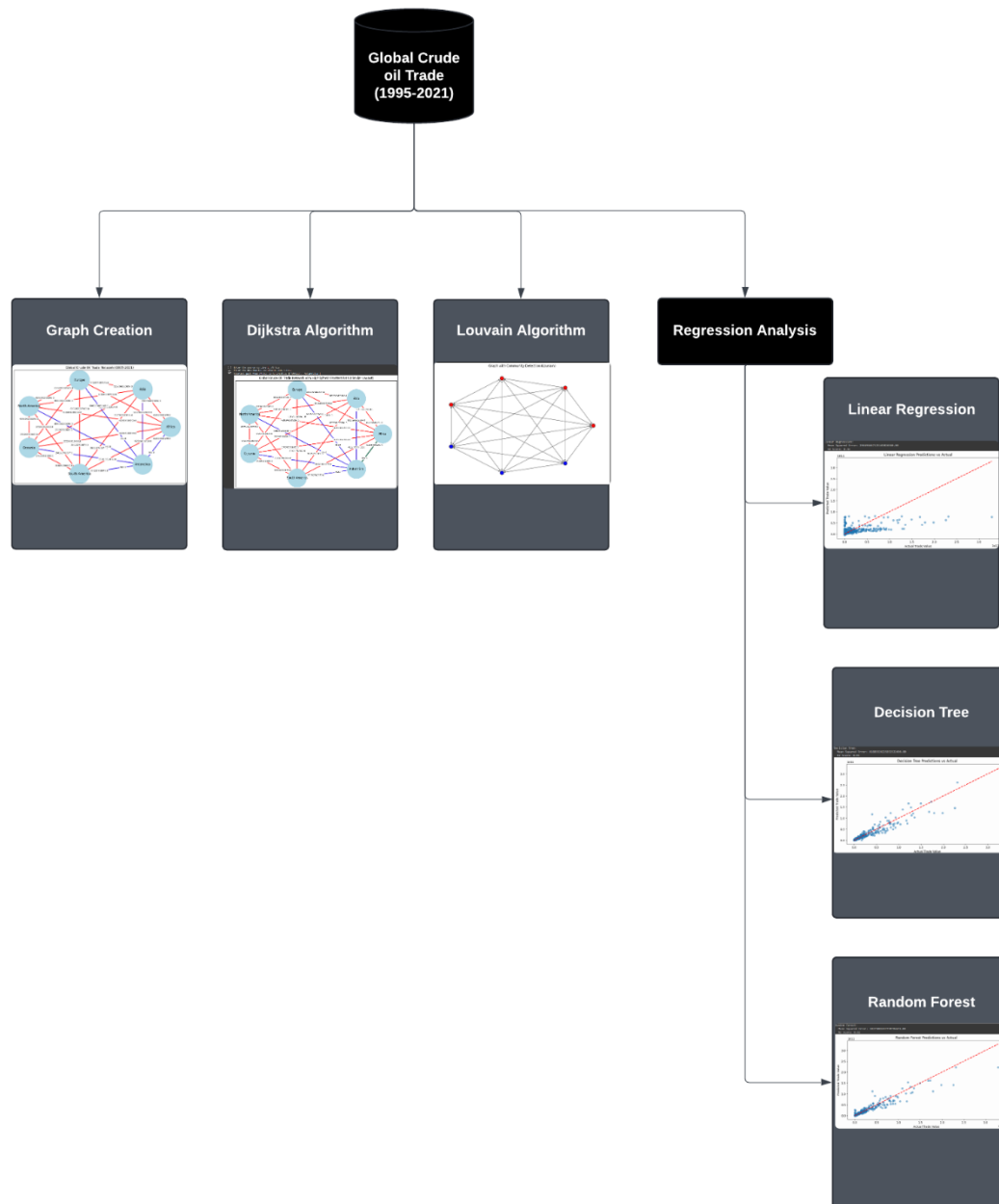


Figure 25: Representation of key findings in the project

The key findings of the project are synthesized in this section, highlighting the successful application of graph analytics and machine learning to predict global trade dynamics. The synthesis focuses on the insights gained from the models, particularly the Random Forest model, and how these insights contribute to a better understanding of global crude oil trade.

The Figure 25 illustrates a data analysis workflow applied to global crude oil trade data from 1995 to 2021. The initial step involves creating a graph that represents the network of crude oil trade between various countries or entities. This network is then analyzed using several algorithms. The Dijkstra algorithm is employed to find the shortest paths within the trade network, possibly to identify the most efficient trade routes. The Louvain algorithm is used for community detection within the graph, helping to identify clusters or groups of countries that have strong trade relationships with each other.

After the graph-based analysis, the workflow proceeds to regression analysis to model and predict trade dynamics. Three types of regression models are applied: Linear Regression, Decision Tree, and Random Forest. These models aim to identify and quantify the relationships between different trade variables, such as trade volume and prices, and predict future trends or patterns in global crude oil trade. The inclusion of multiple regression models suggests a comparative analysis to determine which model provides the most accurate predictions based on the data set.

4.2 Contribution of ML Models to Understanding Graph Structure and Dynamics

Machine learning models such as Random Forest, Decision Tree, Linear Regression are quite central in arriving at the structure and dynamics of a complex trade network like global crude oil trade. In this context, these models were utilized to conduct graph analysis that emerged from the trade data, hence elaborating on a number of aspects related to the structure, behavior, and patterns in the latent network. For example, the Random Forest model, which can be adapted for high-dimensional data and non-linear relationship fits, was very effective in pinning the most influential factors affecting the trade flows between different countries or continents. The feature importance would, with a little analysis, be able to portray how strongly different variables would affect trade relationships—be it geographical proximity, trade agreements, or historical volumes. This information would be very essential in showing the network's underlying structure, important trading hubs, and patterns that one might not draw from a simple graphical analysis.

Not only could these machine learning models help but also use historical data in estimating future trade values. The foresight future scenario of the trends is provided by models that are trained on the past trade volumes and several predictive features. This would go well in predicting changes such as shifts in trading partners in the atmosphere for trading of crude oil or even if new trading hubs develop. This is further useful in understanding potential

vulnerabilities or on the contrary, strengths in the network. This is to say, for example, the resistance of over-fitting and the ability to model complex interactions with many predictors, a Random Forest model was the best model to support a forecasting exercise where classic linear models would normally fail to perform. This has very broad consequences for policy formulation in respect to what policymakers, economists, and businesses base their decision on in respect to valid forecasting exercises regarding trade policies, investment, and strategic planning.

4.3 Broader Implications for Graph Analytics and Data Science

The implications of this project's findings are much more general in graph analytics and data science and extend to complex network modeling and analysis. First, integrating any machine learning models into the graph analysis shows a kind of frontier in the approach to carrying out studies on the networks. Naturally, graph analytics heavily depends on algorithms designed for functioning from within a graph structure, e.g., centrality measures or community detection. Applying machine learning models allows for a much more subtle analysis capable of including both the structure of the graph and exogenous variables or attributes associated with nodes and edges. This hybrid approach opens up insights into the behavior of a network and therefore into the identification of latent patterns and relationships that would be missed by a standard graph approach.

Finally, it might be worth attempting to apply such techniques that have proven successful in the context of global trade networks to other areas where, essentially, complex networks are abundant. For example, an area that may gain a lot through the integration of graph analytics and machine learning is social, biological, and transport networks. In such contexts, knowledge about the dynamics in a network—be it spreading of information in a social network, diseases in a biological network, or traffic in a transportation network—is greatly enhanced through the use of machine learning models that can learn from historical data and carry out predictions on future states of the network. The project is based on the potential of an interdisciplinary application where data science provides robust, data-driven information through machine learning to solidify the decision-making process in a variety of fields.

5. CONCLUSION AND FUTURE WORK

This project made serious contributions to graph analytics by proposing an end-to-end system for analyzing global crude petroleum trade data with various graph-based and machine learning methodologies. By constructing a directed graph using the trade data and running graph analytics methods, such as Dijkstra's method for the shortest path method and the Louvain method for community detection, insight was further extracted into the trade patterns and regional trade dynamics. Network graphs and community clusters consisting of graph visualizations offer a window into understanding relationships for stakeholders with respect to better trade, bringing into view strategic trade routes and interconnected clusters of continents. The technique enhances interpretation of even very complex trade data and is general enough to be applicable to graph analytics that are of relevance to global trade studies.

5.1 Reflection on the Effectiveness of the ML Models Used in the Project

Three different machine learning models have been used in the project: Linear Regression, Decision Tree, and Random Forest. Accuracy among models varied, with the Random Forest model performing the best and being the most solid. A good show was given by the Decision Tree model, maintaining the best balance between accuracy and interpretability. The Linear Regression model performed relatively poorly but was useful for baseline comparisons. The performance of the models in predicting trade values highlights the importance of choosing and tuning machine learning algorithms based on the needs of the dataset and problems to be solved. This comparative analysis of the models focused on the strengths of ensemble methods, such as Random Forest, in dealing with complex, non-linear relations within trade data.

5.2 Suggestions for Future Work

The improvement of such developed predictive and analytical frameworks could be made by including more sources of data, perhaps including geopolitical or real-time economic indicators in a bid to make the model predictions better and present a fuller view of what goes into making global trade. It would also be worthwhile to investigate state-of-the-art techniques in graph representation and machine learning models, such as the GNNs or deep learning-based methods, to potentially gain better insights and predictive accuracy. User feedback would also be incorporated on the front end to make visualizations and predictive models more useful and user-friendly for stakeholders. An additional way to deepen this line of research would be to

stretch the use of these methodologies to other activity domains, such as the financial market or the management of the supply chain, to see how versatile and appropriate the methodological exploration is.

APPENDIX

Appendix 1: Front Page

- This appendix contains the front page of the project report, including the title, author name, institution, and date.

Appendix 2: Certificate

- This appendix includes the certification of the completion of the project. It may include signatures from the supervising faculty or institution confirming the authenticity and completion of the project.

Appendix 3: Contents Page

- This appendix provides a detailed table of contents for the project report, listing all chapters, subsections, and appendices for easy navigation.

Appendix 4: Timeline or Gantt Chart

- This appendix includes a timeline or Gantt chart illustrating the project's progress over time. It shows key milestones, deadlines, and the sequence of tasks performed during the project.

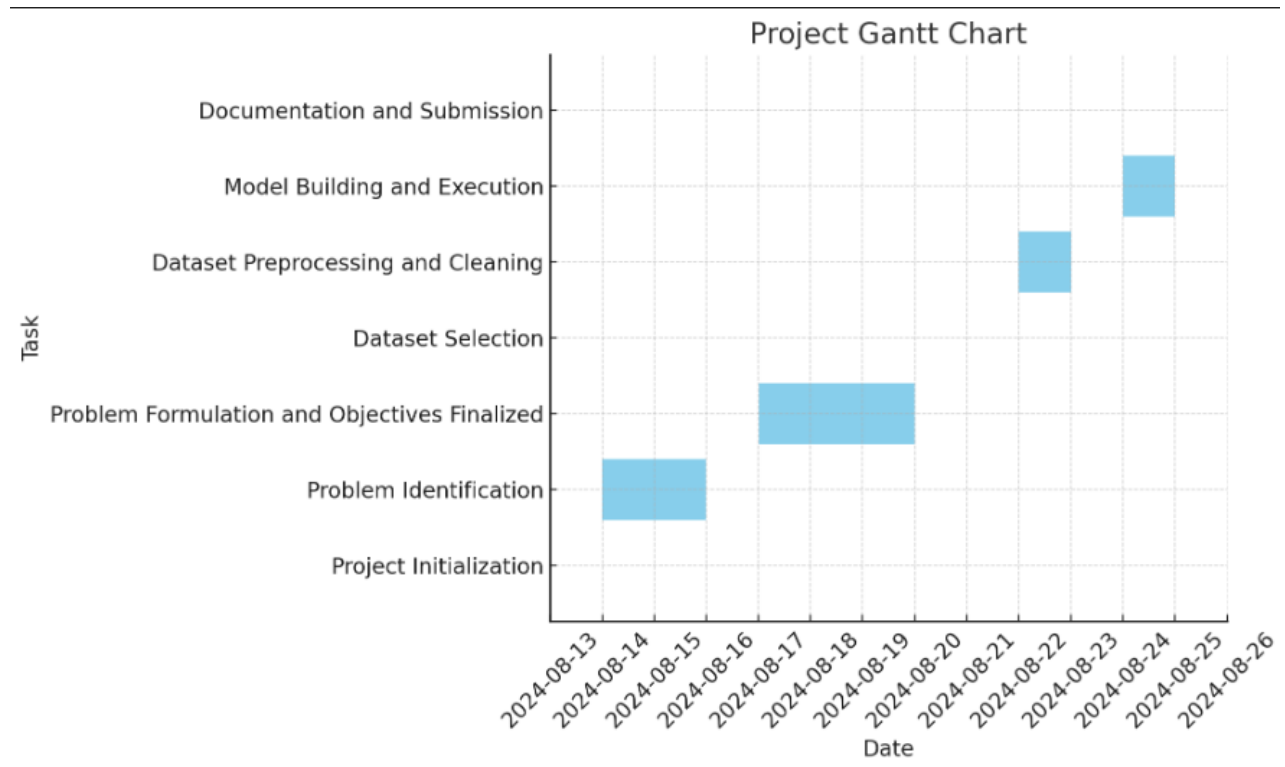


Figure 26: Gantt Chart of Project time line

Appendix 5: Plagiarism Report

This appendix includes the plagiarism report for the project, verifying the originality of the work and ensuring that proper citations and references are made.

The screenshot shows the QuillBot AI Detector interface. The text being analyzed is: "changing relational pattern of the trade flows between different continents. We have constructed node-and-edge graphs for the continents, with the emphasis put on the respective relations in between. In every case, the weight was the trade value, which helped in establishing how much trade was flowing between continents. For better clarity, the lines were colored red for imports and blue for exports. This way of coloring enabled the obviousness in the visual analytics of the trade dynamics—whether there were active importers or exporters in a given region. To complete the dataset, it was made into a graph. Advanced graph algorithms like Dijkstra's and Louvain were used, thereby justifying an improved analysis and understanding of the global petroleum trade network."

The analysis result shows 0% of text is likely AI-generated. The breakdown is as follows:

Category	Percentage
AI-generated	0%
AI-generated & AI-refined	0%
Human-written & AI-refined	0%
Human-written	100%

The interface also includes a sidebar with various tools like Paraphraser, Grammar Checker, Plagiarism Checker, Summarizer, Translator, Citation Generator, and QuillBot Flow. The bottom status bar indicates "Analysis complete" and "9,079/1,200 words".

The report suggests that this project report is 100 % human written

References

- [1] Gao, C., Zheng, Y., Li, N., Li, Y., Qin, Y., Piao, J., ... & Li, Y. (2023). A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems*, 1(1), 1-51.
- [2] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., ... & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI open*, 1, 57-81.
- [3] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1), 4-24.
- [4] Liang, F., Qian, C., Yu, W., Griffith, D., & Golmie, N. (2022). Survey of graph neural networks and applications. *Wireless Communications and Mobile Computing*, 2022(1), 9261537.
- [5] Wu, S., Sun, F., Zhang, W., Xie, X., & Cui, B. (2022). Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5), 1-37.