Course Project

Advanced Machine Learning

GDP Forecasting Using ML Methods on Time Series Data

Master's in Software Engineering

Students: Anjeza Xhelilaj, Ema Elezi, Laura Dule

08-02-2024

Përmbajtja

Authorship Declaration

We, the members of the group , hereby declare with full responsibility that this assignment represents our original work. The task has been completed using only the references and resources specifically related to this project. The work has been equally contributed to by all

members of the group, with each member actively participating in every phase of its development. By submitting this assignment, we confirm our understanding of the importance of originality in academic research, as well as our commitment to the ethical considerations associated with it.

Introduction

Problem Description

Gross Domestic Product (GDP) is a key indicator of a country's economic health. Accurate GDP forecasting helps policymakers, businesses, and investors make informed decisions about their economic strategies. Traditional statistical methods, while effective, often fail to capture the complex patterns within time series data. With advancements in **Machine Learning** and **time series forecasting methods**, it is now possible to develop more accurate and sophisticated models for GDP prediction.

Importance of the Problem

An accurate GDP forecasting model can have a significant impact in several areas:

- **Economic Policies** – Governments can use these forecasts to design better fiscal and monetary policies.
- **Business Planning** – Companies can adjust their strategies based on expected economic growth or slowdowns.
- **Investments and Financial Markets** – Investors can make more informed decisions by leveraging advanced forecasting models.

Project Objectives

1. **Perform data preprocessing** and **exploratory data analysis (EDA)** to understand GDP trends and patterns.
2. **Develop** various time series **forecasting models**, including **ARIMA, VAR, and Random Forest,** using metrics such as RMSE, MAPE, and R^2 to evaluate the accuracy of the predictions.
3. **Compare model performance** to identify the most accurate approach for GDP prediction.

About the dataset

The process of extracting economic data for Albania from the EuroSTAT database was carried out using its official API through Python (for more details, see "Demonstration Code for Data Extraction" in the appendix) **[1]**.

The dataset contains 67 quarterly observations (2008Q1 – 2024Q3) and 10 economic variables measured in Million Euros at current prices (CP_MEUR), as follows:

- **GDP (B1GQ):** Represents the total value of goods and services produced within Albania's territory during a given quarter. Numeric, continuous variable.
- **Value Added (B1G):** Measures the difference between total output and intermediate consumption, indicating the real added value in the economy. Numeric, continuous variable.
- **Consumption (P3):** Includes all final consumption expenditures in the economy, combining both private and public consumption. Numeric, continuous variable.
- **Government Individual Consumption (P31_S13):** Covers government expenditures on services that directly benefit individuals (such as education and healthcare). Numeric, continuous variable.
- **Government Collective Consumption (P32_S13):** Covers government expenditures on services that benefit society collectively (such as defense and public order).
- **Household Consumption (P31_S14):** Includes all household expenditures on final goods and services.
- **Gross Capital Formation (P51G):** Represents total investments in fixed assets within the economy, including infrastructure and equipment investments.
- **Exports (P6):** Includes the total value of goods and services exported abroad.
- **Imports (P7):** Includes the total value of goods and services imported from abroad.
- **Taxes Less Subsidies (D21X31):** Represents the difference between collected product taxes and granted subsidies, reflecting net fiscal intervention.

Detaje të Dataset-it:

- **Source:** EuroSTAT Database (namq_10_gdp)
- **Period:** 2008-2024
- **Frequency:** Quarterly
- **Measurement Unit:** Current Prices in Million Euros (CP_MEUR)
- **Seasonal Adjustment:** Not seasonally adjusted (NSA)
- **Country:** Albania
- **Size:** 67 rows × 11 columns
- **Dependent Variable:** GDP

The tools and technologies used for this project include the programming languages **R** and **Python**, **Orange** as well as **Chat GPT**, which facilitated data visualization and result interpretation.

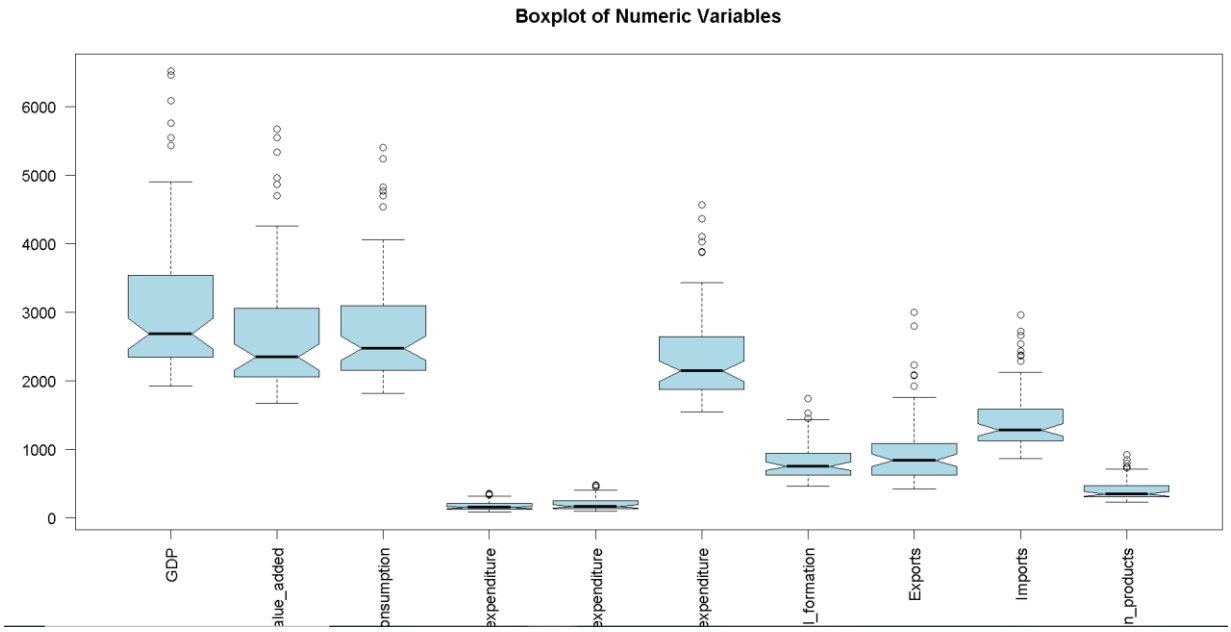


Figure 1: Boxplot of the key factors in the GDP dataset. This chart helps identify the distribution and outliers **[2]**.

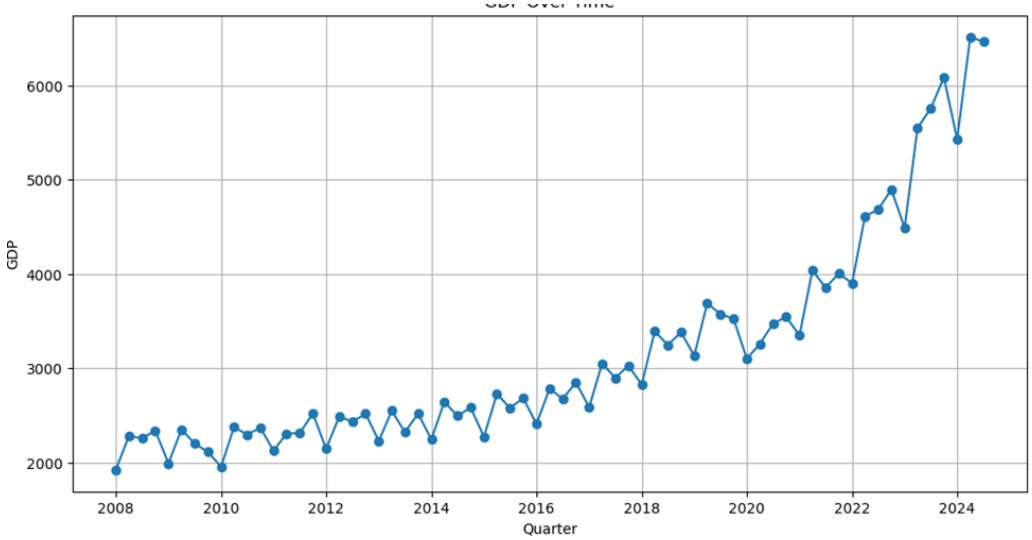


Figure 2: This chart illustrates the evolution of GDP over time, expressed in quarterly periods **[3]**.

Data Pre-processing

The steps we have taken to clean and prepare the data include:

- **Removal of outliers.** Based on the above boxplot, it can be observed that some of the numerical data contain abnormal values (outliers), which have been removed using an R script described in the respective file in this document. A comparison of the boxplot before and after the cleaning shows that the outliers have been eliminated.

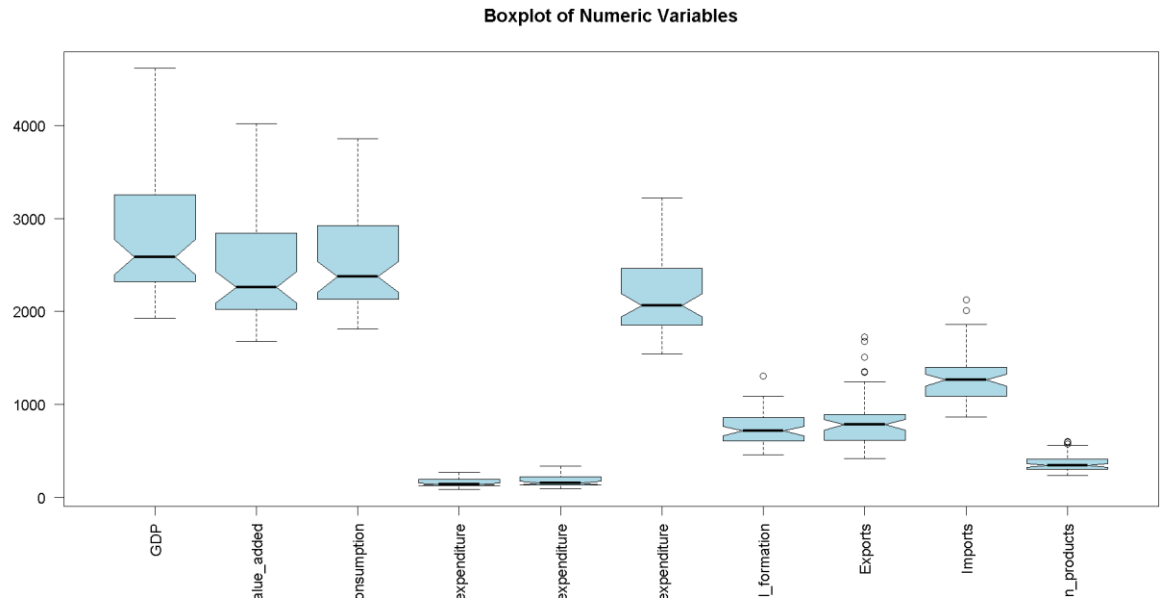


Figure 3: Boxplot after the removal of outliers **[4]**.

- Our target variable is GDP, which is a continuous numerical variable.
- For this dataset, no missing values were identified.

EDA - Exploratory Data Analysis

In this section, we will examine the exploratory analyses of this model and the trends of the time series over these years.

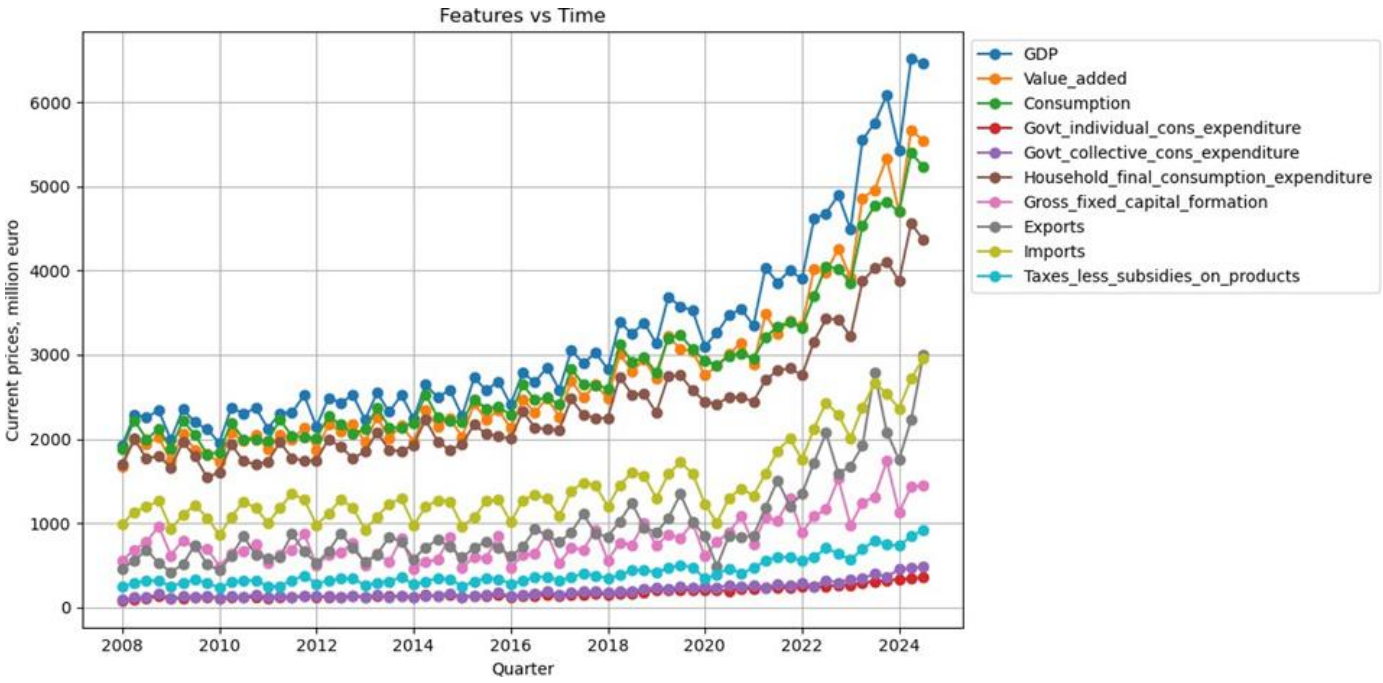


Figure 4: A graph containing various economic factors that contribute to GDP, such as Consumption, Exports, Imports, and Government Spending **[5]**.

Delving deeper into further analyses, we observe that most indicators show an upward trend, indicating economic development over time. The strong growth **after 2020** could be linked to the economic recovery post-pandemic or changes in economic policies.

Consumption (green) and **Household Spending** (brown) show a similar movement and are increasing. This indicates that household spending is an important factor in GDP growth.

Imports (yellow) and **exports** (pink) follow a cyclical pattern, reflecting changes in international trade. Imports are generally higher than exports, which could indicate trade deficits during certain periods.

Government individual spending (red) and **collective spending** (purple) are more stable compared to other variables. This suggests that government spending experiences less fluctuation than other economic factors.

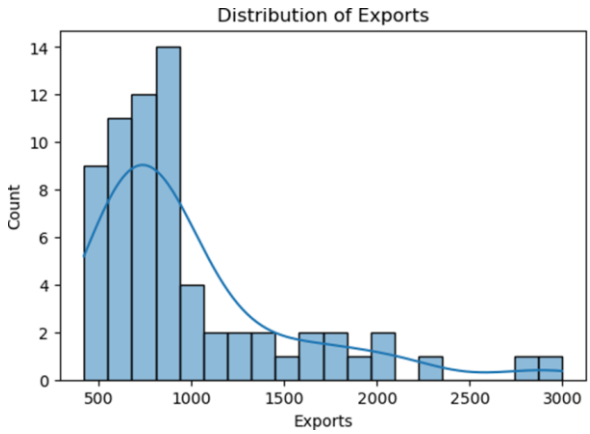


Figura 5: This graph shows the distribution of exports and is a histogram with a fitted density (KDE – Kernel Density Estimation) **[6]**.

X-axis (Exports): Represents the values of exports, distributed at different intervals. **Y-axis** (Count): Represents the number of cases for each export interval.The distribution is right-skewed, meaning that most of the values are in the lower interval (around 500-1000), while there are some higher values that occur less frequently. This feature suggests that most exports are at low or medium levels, with a few rare cases of very high exports. The density line (KDE) shows the smooth distribution of the data and confirms their asymmetric nature.

Now, we will examine the correlation between GDP and Household Consumption, visually represented in a graph, to see how household expenditures impact GDP.

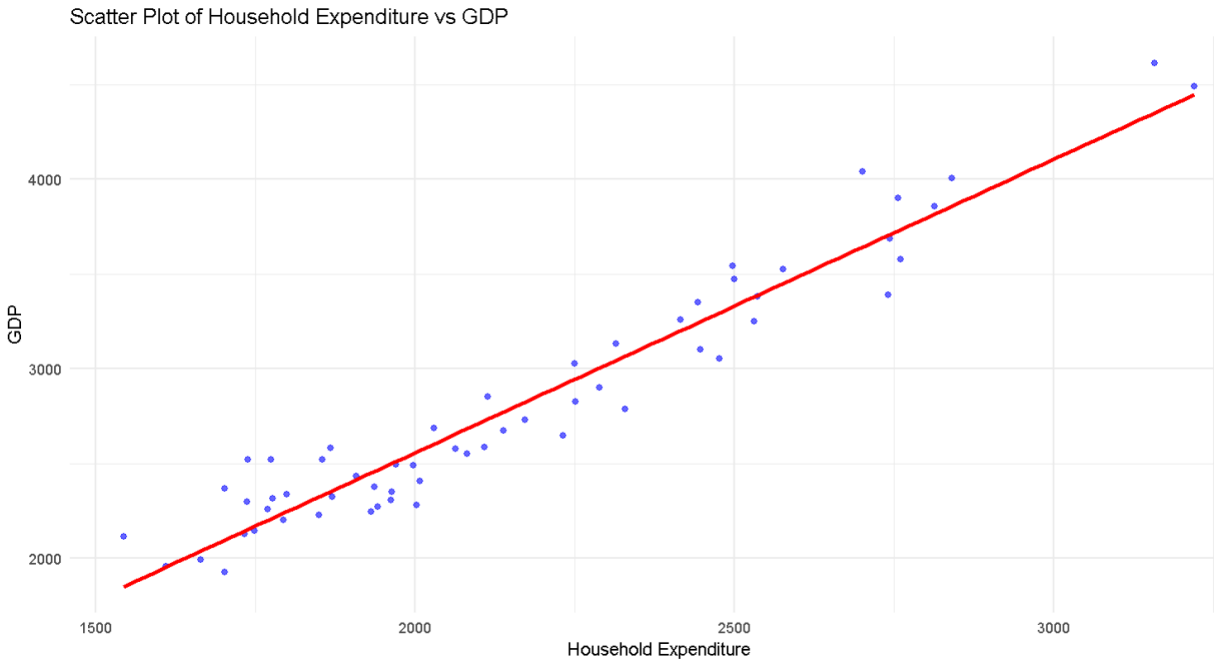


Figure 6a: The scatter points show an upward trend, indicating a **positive correlation** between Household Expenditure and GDP. This means that **as Household Expenditure increases, GDP** also tends to **increase**. The red regression line confirms this relationship by following the positive trend. The results suggest that **Household Expenditure** is a key driver of GDP in this dataset. This aligns with economic theory, where higher consumer spending **contributes** to economic growth. Governments and policymakers may focus on stimulating household expenditure (e.g., through tax cuts or subsidies) to **boost** GDP **[7]**.

```
> cor(Dataset_GDP_out$Household_final_consumption_expenditure, Dataset_GDP_out$GDP)
[1] 0.9647779
```

Figura 6b: The correlation coefficient was calculated, and it is **close to +1**, reinforcing a strong linear relationship between the variables.

Now let's see a more general view:

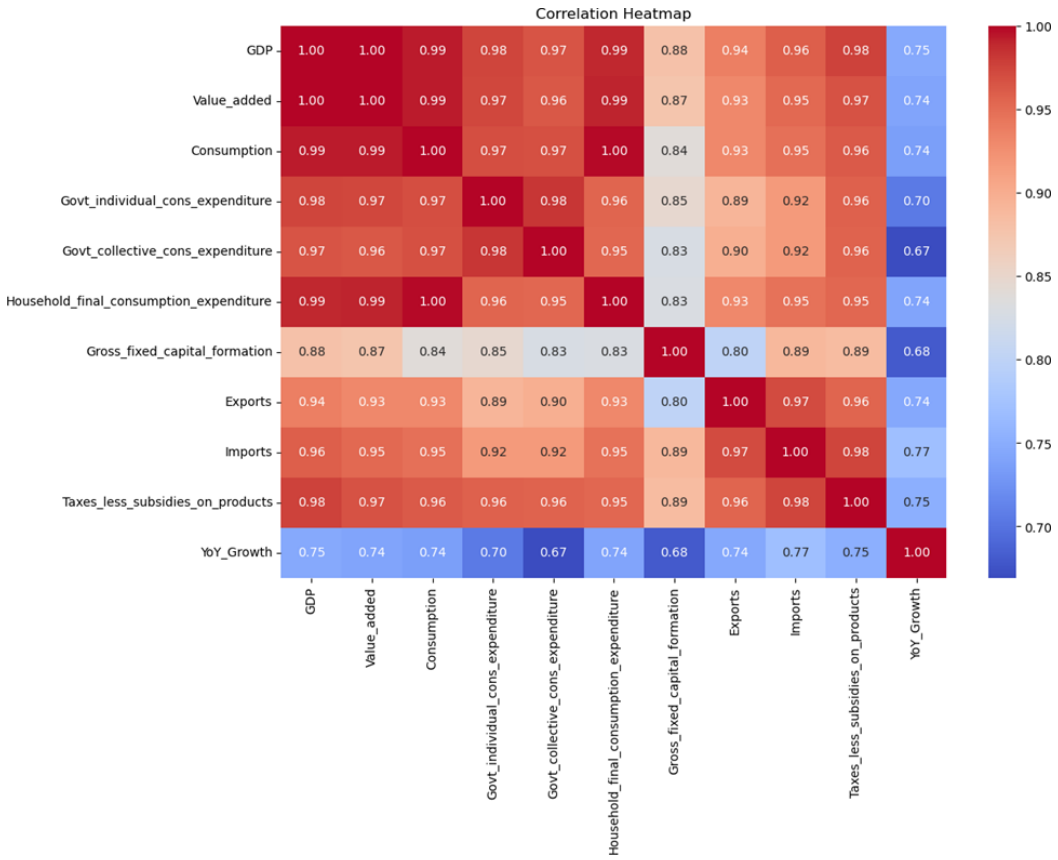


Figura 7: The heatmap highlights strong relationships between GDP and macroeconomic indicators. This means that as macroeconomic indicators increase, GDP also tends to increase. **Strong correlations** (0.95-1.00) identified between: *GDP and Value Added, GDP and Consumption, GDP and Household Final Consumption*. **Moderate correlations** (0.85-0.94) observed with: *Exports and Imports, Gross Fixed Capital Formation* **[8]**.

Model Fitting

In this section, we focus on building and fine-tuning our forecasting model for GDP. Since our dataset contains multiple economic indicators, we first convert it into a **multivariate time series** to capture the relationships between GDP and other relevant variables. This transformation allows our model to leverage additional information beyond just past GDP values, improving the accuracy of our predictions. Since our dataset has multiple variables recorded at regular yearly intervals, we will use the `ts()` function in R language (*more info in the appendix*) **[9].**

Now, we will use the `auto.arima()` function, which is a built-in function in R that automates the selection of the best-fitting ARIMA model for a given time series. In our

case, we will apply `auto.arima()` to the GDP variable, as it is the key variable we aim to forecast and analyze in our model.

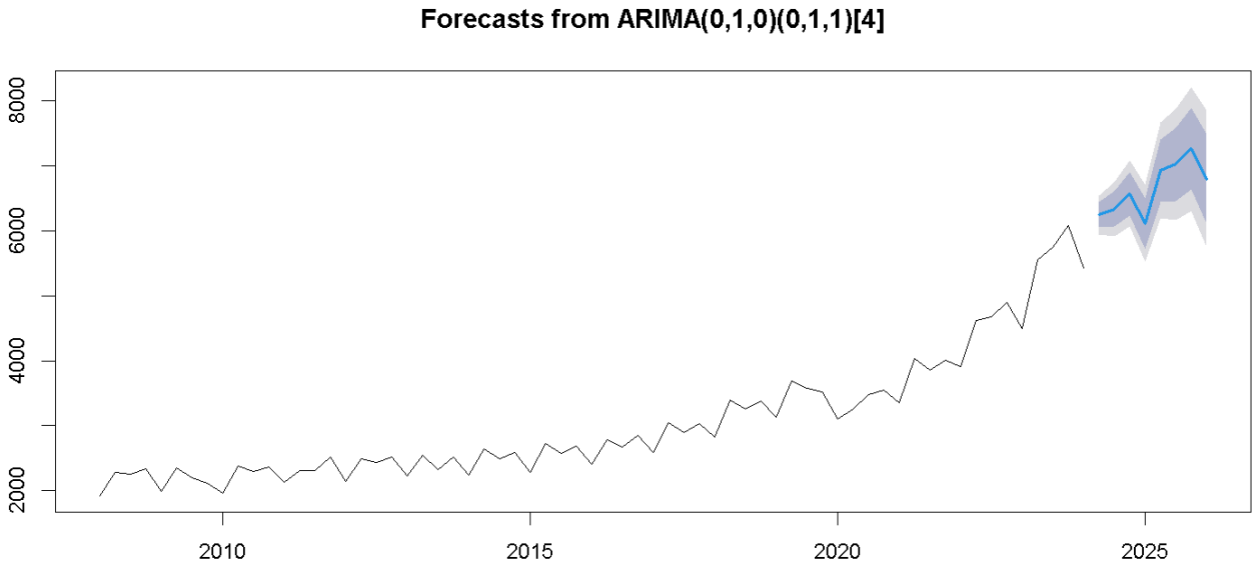
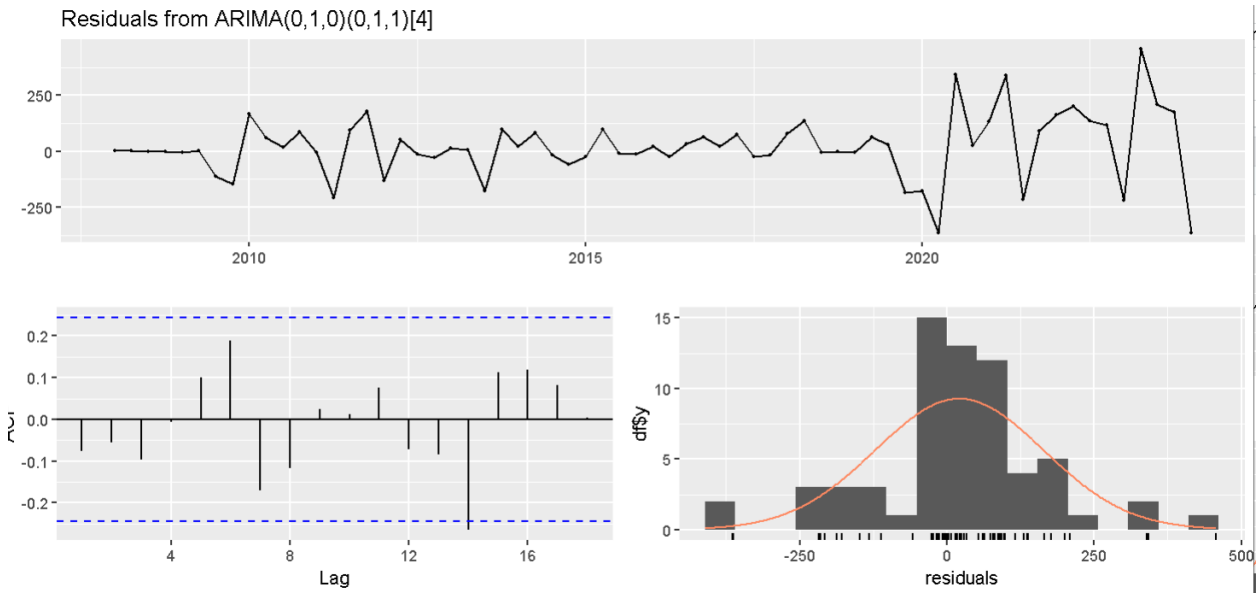


Figure 8: The model forecasts GDP for future quarters beyond 2024. The **blue line** represents the predicted values. The **shaded regions** indicate confidence intervals (uncertainty in the forecast). The darker shade shows the **most probable range**. The lighter shade represents **wider uncertainty**. **(0,1,0):** This means the model uses **one difference** to make the series stationary (to remove trends). **(0,1,1) [4]:** Seasonal component with **quarterly** (4-period) seasonality. The model predicts that GDP will **continue to rise** in the coming quarters, but with some fluctuations. The model captures both the **trend** and **seasonal patterns** present in past GDP data **[10].**

The Ljung-Box test checks for significant autocorrelations in a time series, determining if residuals are random (white noise) or if further modeling is needed. Let's apply it on our ARIMA model and see the output.




```
        Ljung-Box test

data:  Residuals from ARIMA(0,1,0)(0,1,1)[4]
Q* = 7.811, df = 7, p-value = 0.3496

Model df: 1.   Total lags used: 8
```

Figura 9: Here we have:

Residual Diagnostic Plots:

In the Top plot **(Residuals over time), r**esiduals appear to fluctuate randomly **around zero**, which is a good sign. In the **bottom left plot (ACF of residuals),** most of the autocorrelations are within the confidence bands, further confirming that there is **no significant** autocorrelation. In the bottom right plot **(Histogram of residuals),** the residuals appear roughly normal, which **supports** model adequacy.

Ljung-Box Test Output:

In the Ljung-Box Test Output we can see that the p-value is 0.3496, so it is greater than 0.05. Therefore, we **fail to reject the null hypothesis**, meaning there is **no significant autocorrelation** in the residuals, suggesting a well-fitted model **[11]**.

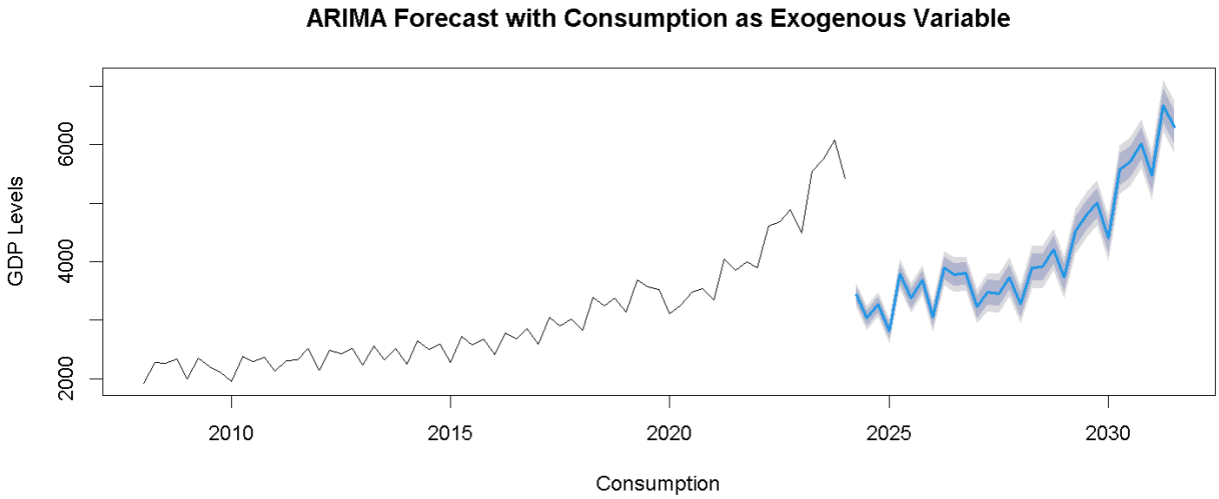


Figura 10: This graph shows the **forecasted GDP** levels using an **ARIMA** model with consumption as an exogenous variable. The exogenous variable (Consumption) helps improve the forecasting of GDP, assuming a relationship between them. Since **Consumption** is used as an explanatory variable, the GDP forecast depends on future consumption values. If **consumption increases**, GDP is predicted to grow accordingly. However, as with all forecasts, the **confidence interval widens over time**, indicating increasing uncertainty in long-term predictions **[12]**.

Visualization and Interpretation

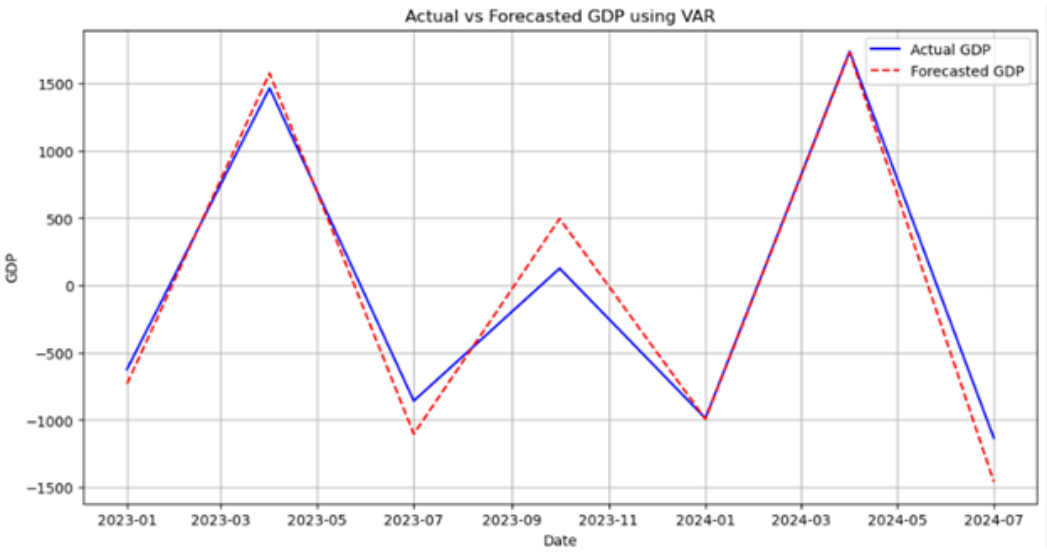


Figure 11: "Actual vs. Forecasted GDP using VAR" **[13].**

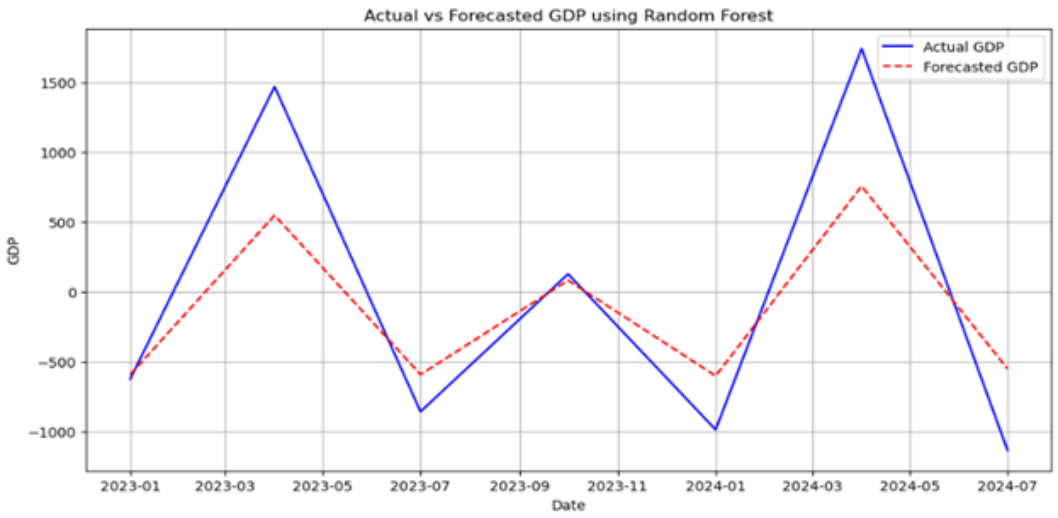


Figura 12: "Actual vs. Forecasted GDP using Random Forest" **[13].**

VAR Model:

- **Strengths:** Captures GDP trends well, closely follows actual values, and performs effectively in both growth and decline phases.
- **Weaknesses:** Slight lag in predictions, larger errors at extreme points (e.g., July 2023 drop, April 2024 rise), and occasional overshooting.

Random Forest Model:

- **Strengths:** Provides a stable forecast, effectively capturing general trends in smoother GDP transitions.
- **Weaknesses:** Underestimates peaks, overestimates troughs, appears less responsive to sharp variations, and deviates more from actual GDP than the VAR model.

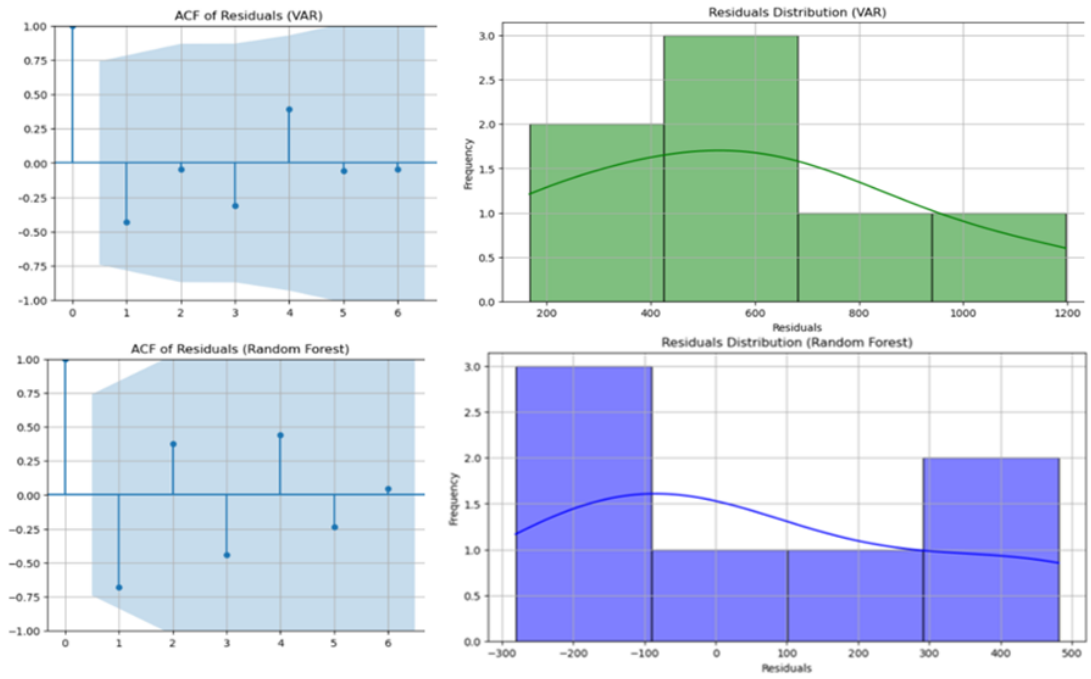


Figura 13: Histograms of residuals for both models indicate the distribution of prediction errors. Autocorrelation function (ACF) plots are used to check for residual dependencies, ensuring model assumptions are met **[14]**.

Model Evaluation

Now, let's evaluate the performance of the ARIMA, VAR, and Random Forest (RF) models, a key step in assessing their effectiveness.

```
Series: Dataset_GDP_ts[, "GDP"]
ARIMA(0,1,0)(0,1,1)[4]

Coefficients:
          sma1
      -0.5330
s.e.   0.1553

sigma^2 = 22848:  log likelihood = -386.4
AIC=776.8   AICc=777.01   BIC=780.99

Training set error measures:
                    ME     RMSE      MAE       MPE   MAPE      MASE        ACF1
Training set 20.65945 144.0112 98.99249 0.4373584 3.0092 0.3741931 -0.07656866
```

Figura 14: This output provides details on the **ARIMA(0,1,0)(0,1,1)[4]** model used for forecasting GDP **[15]**.

We have the **coefficients** where **`sma1 is the`** seasonal MA (Moving Average) coefficient, showing the effect of past seasonal errors and **`s.e. is the`** standard error, indicating the uncertainty in this coefficient.

Moreover, we have the **model fit statistics like:**

- **Log-likelihood = -386.4** . Used to compare models; higher is better.
- **AIC (Akaike Information Criterion) = 776.8**
- **AICc (Corrected AIC) = 777.01**
- **BIC (Bayesian Information Criterion) = 780.99**

Training Set Error Measures:

- **ME** (Mean Error) = 20.6595 → Average prediction error (bias)
- **RMSE** (Root Mean Squared Error) = 144.0112 → Standard deviation of prediction errors (lower is better)
- **MAE** (Mean Absolute Error) = 98.99249 → Average absolute difference between actual and predicted values.
- **MPE** (Mean Percentage Error) = 0.4373% → Percentage bias (close to 0 means low bias)
- **MAPE** (Mean Absolute Percentage Error) = 3.0092% → Percentage error in prediction (3% is relatively low, indicating accurate forecasting.)
- **MASE** (Mean Absolute Scaled Error) = 0.3741 → Scaled MAE; values <1 suggest a better model than naive forecasting
- **ACF1** (Autocorrelation of residuals at lag 1) = -0.0766 → Near zero, indicating little correlation in residuals (a good sign).

VAR Model Performance

The performance of the Vector Autoregressive (VAR) model is assessed using three key evaluation metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2).

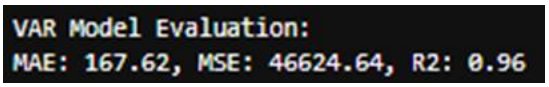


Figura 15: "VAR Model Evaluation" **[16].**

- The **MAE** of 167.62 indicates the average magnitude of the errors in the model's predictions.
- The **MSE** of 46,624.66 quantifies the average squared difference between the predicted and actual values.
- The **R-squared** value of 0.96 is a measure of how well the model explains the variance in the dependent variable.

Random Forest Model Performance

The performance of the Random Forest model is assessed using also the three key evaluation metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2).

```
Random Forest Model Evaluation:
MAE: 459.19, MSE: 339681.08, R2: 0.72
```

Figura 16: "Random Forest Model"**[17]**.

- The **MAE** of 459.19 indicates the average magnitude of the errors in the model's predictions.
- The **MSE** of 339,681 quantifies the average squared difference between the predicted and actual values.
- The **R-squared** value of 0.72 indicates that the model explains 72% of the variance in the data

Conclusions

We have now reached the final stage, where, based on our work and experimentation, we will draw conclusions on GDP forecasting, the key influencing factors, and provide recommendations for future improvements.

Below is a summary table of the three time series models we analyzed. Based on this table, we will rank the models from the best to the least effective.

Model Performance Overview

Metric	ARIMA	VAR	Random Forest
MAE (Lower is better)	**98.99** (Best)	**167.62** (Moderate)	**459.19** (Worst)
RMSE/MSE (Lower is better)	**144.01** (Best)	**46,624.66** (Moderate)	**339,681** (Worst)
MAPE (Lower is better)	**3.01%** (Best)	(Not provided)	(Not provided)
R^2 (Higher is better)	(Not applicable)	**0.96** (Best)	**0.72** (Worst)
Residual Autocorrelation	**-0.0766** (Near zero, good)	**Present** (Moderate)	**Weak** (Good)

Residual Distribution	**Stable, normal** (Good)	**Right-skewed, large spread** (Bad)	**High variance** (Bad)

🥇 ***1st Place: ARIMA (Best Model)***

✓ **Lowest MAE, RMSE, and MAPE** → most accurate predictions

✓ **Minimal autocorrelation** → residuals behave well

✓ **Residuals follow normal distribution** → stable model

🔹 **Best choice for short-term GDP forecasting.**

🥈 ***2nd Place: VAR (Moderate Performance)***

✓ **High R^2 (96%)** → good explanatory power

❌ **Higher MAE & RMSE than ARIMA** → less accurate predictions

❌ **Residuals show right-skewness & spread** → less stable model

🔹 **Better suited for capturing multivariate relationships but weaker at precise forecasting.**

🥉 ***3rd Place: Random Forest (Worst Fit)***

❌ **Worst MAE and RMSE** → largest prediction errors

❌ **Lower R^2 (72%)** → weaker explanatory power

✓ **Weak residual autocorrelation** (good)

❌ **High variance in residuals** → unstable predictions

🔹 **Less suitable for time series forecasting, better for classification or regression.**

Key Findings

1. **Economic Trends & Influencing Factors**:

a. GDP exhibits an overall upward trend, with fluctuations linked to external shocks, such as the **COVID-19 pandemic** and subsequent economic recovery.
b. **Household Consumption** strongly correlates with GDP, indicating that consumer spending is a key driver of economic growth.
c. **Imports & Exports** follow cyclical patterns, reflecting international trade fluctuations.
d. **Government Spending** remains relatively stable, contributing to GDP growth without sharp variations.

Referencat

- TATBurimi i setit të të dhënave: *GDP and main components (output, expenditure and income). (2025, January 20). EuroSTAT. Retrieved January 25, 2025, from https://ec.europa.eu/eurostat/databrowser/view/namq_10_gdp/default/table?lang=en&category=na10.namq_10.namq_10_ma*
- *https://www.edu4schools.gr/SP/SPMaterial/Download?filename=1350282.html&view=True&backUrl=%2FSP%2FSPLessons%2FLesMods%3FlesId%3D705377%26stuId%3D1746289*

Shtojca (Appendix)

[1]. Kodi demostrues i ekstraktimit të të dhënave:

```
#Library Installation
pip install eurostat
pip install openpyxl
pip install pandas

#Importing Required Libraries
import eurostat
import pandas as pd

#Retrieving data from Eurostat
code = 'namq_10_gdp' # Eurostat code for quarterly GDP data
data = eurostat.get_data(code)
df = pd.DataFrame(data)
print(df)

#Filtering economic data for Albania
my_filter_pars = {
```

```
    'startPeriod': 2008,
    'endPeriod': 2024,
    'unit': ['CP_MEUR'],
    's_adj': ['NSA'],
    'na_item': ['B1GQ', 'B1G', 'P3', 'P31_S13', 'P32_S13', 'P31_S14', 'P51G',
    'P6', 'P7', 'D21X31'],
    'geo': ['AL'] #Geographical region: Albania
}
data = eurostat.get_data_df(code, filter_pars=my_filter_pars)
out = (data.set_index(["freq", "unit", "s_adj", "na_item"])
           .T.droplevel([0, 1, 2], axis=1)
           .reset_index(names="na_item")
           .rename_axis(columns=None))
print(out)

# Exporting filtered data to an Excel file
out.to_excel('Dataset_AML.xlsx', sheet_name='Dataset_AML')
df = pd.read_excel('Dataset_AML.xlsx')
print(df)

# Enhancing Readability
df = df.rename(columns={
    "P3": "Consumption",
    "B1GQ": "GDP",
    "B1G": "Value_added",
    "P31_S13": "Govt_individual_cons_expenditure",
    "P32_S13": "Govt_collective_cons_expenditure",
    "P31_S14": "Household_final_consumption_expenditure",
    "P51G": "Gross_fixed_capital_formation",
    "P6": "Exports",
    "P7": "Imports",
    "D21X31": "Taxes_less_subsidies_on_products",
    "na_item": "Quarter"
})

# Converting Quarters to a standard date format
def convert_quarter_to_date(quarter_str):
    year, quarter = quarter_str.split('-')
    month = {'Q1': '1', 'Q2': '4', 'Q3': '7', 'Q4': '10'}[quarter]
    return f"{month}/1/{year}"
df['Quarter'] = df['Quarter'].apply(convert_quarter_to_date)
print(df[['Quarter']])
```

```
# Saving the final dataset
df.to_excel(r'C:\\Users\\Lenovo\\Dataset_GDP.xlsx', index=False)
```

[2]. Kodi në R krijon një grafik boxplot që tregon shpërndarjen e të dhënave për secilin variabël në dataset. Ai tregon medianën, kuartilet, si dhe mund të identifikojë vlerat ekstreme (outliers) për secilin nga variablat që përfshihen në dataset.

```
> file_path <- "C:\\Users\\User\\OneDrive\\Documents\\ML Master\\Dataset_GDP.csv"
> Dataset_GDP <- read.csv(file_path)
> numeric_data <- Dataset_GDP[sapply(Dataset_GDP, is.numeric)]
> boxplot(numeric_data, main = "Boxplot of Numeric Variables", col = "lightblue", las = 2, notch = TRUE, outline = TRUE)
```

[3]. This Python code generates a time series visualization of GDP over time. It first sets the figure size to 12x6 for better readability. Then, it plots the GDP values using a line with markers to highlight individual data points. The plot is labeled with a title, "GDP Over Time," and includes axis labels for quarters (x-axis) and GDP values (y-axis). A grid is added to enhance readability, and finally, the plot is displayed using plt.show().

```
# Step 1.1: Time Series Visualization
plt.figure(figsize=(12, 6))
plt.plot(data.index, data['GDP'], marker='o', linestyle='-')
plt.title('GDP Over Time')
plt.xlabel('Quarter')
plt.ylabel('GDP')
plt.grid()
plt.show()
```

[4]. Me poshte gjender kodi I perdorur ne R per te bere heqjen e vlerave anormale (outliers), duke përdorur një funksion që kontrollon nëse një vlerë është një **outlier** përmes *boxplot.stats()*. Nëse një vlerë është jashtë kufijve të përcaktuar nga **IQR (Interquartile Range)**, ajo merret si outlier. Pas kësaj, një matricë logjike krijohet për të identifikuar të gjitha vlerat e outliers në dataset, dhe më pas hiqen të gjitha rreshtat që përmbajnë outliers.

```
# Define a function to identify outliers in a single column
> is_outlier <- function(column) {
+     if (is.numeric(column)) {
+         outliers <- boxplot.stats(column)$out   # Identify outliers
+         return(column %in% outliers)
+     } else {
+         return(rep(FALSE, length(column)))   #Non-numeric columns have no outliers
+     }
+ }
# Create a logical matrix for outliers
> outlier_matrix <- sapply(Dataset_GDP, is_outlier)
>
```

```
# Remove rows where any column has an outlier
> Dataset_GDP_out <- Dataset_GDP[!rowSums(outlier_matrix), ]
```

[5]. This Python code visualizes all features in the dataset over time, excluding the 'Quarter' column. It iterates through each feature, plots its values with markers and lines, adds a title, labels, a legend, and a grid for clarity, then displays the plot with a tight layout.

```
# Step 1.1: Time Series Visualization for All Features
plt.figure(figsize=(12, 6))

# Loop through each feature (except 'Quarter') and plot it
for feature in data.columns:
    if feature != 'Quarter':
        plt.plot(data.index, data[feature], marker='o', linestyle='-', la

# Customize the plot
plt.title('Features vs Time')
plt.xlabel('Quarter')
plt.ylabel('Current prices, million euro')
plt.legend(loc='upper left', bbox_to_anchor=(1, 1))
plt.grid()

# Show the plot
plt.tight_layout()
plt.show()
```

[6]. This code performs a distribution analysis for each column in the dataset. For each column, it first generates a histogram with a Kernel Density Estimate (KDE) overlay to show the distribution of values. It also creates a boxplot for each column to visually display the spread, central tendency, and potential outliers. The plots are displayed one by one with appropriate titles for each feature, helping in understanding the data distribution and any potential anomalies.

```
# Distribution Analysis
for col in data.columns:
    plt.figure(figsize=(6, 4))
    sns.histplot(data[col], kde=True, bins=20)
    plt.title(f'Distribution of {col}')
    plt.show()

    plt.figure(figsize=(6, 4))
    sns.boxplot(y=data[col])
    plt.title(f'Boxplot of {col}')
    plt.show()
```

[7]. This R code uses the ggplot2 library to create a scatter plot to visualize the relationship between household final consumption expenditure (Household_final_consumption_expenditure) and GDP (GDP). It plots the data points as blue dots with some transparency (alpha = 0.6) to better visualize overlapping points. A red linear regression line is added (geom_smooth(method = "lm", color = "red", se = FALSE)) to show the trend between the two variables, without the confidence interval shading (se = FALSE). The plot is titled "Scatter Plot of Household Expenditure vs GDP" with labeled axes for clear interpretation. The minimal theme (theme_minimal()) is used for a cleaner presentation.

```
> library(ggplot2)
```

```
> ggplot(Dataset_GDP_out, aes(x = Household_final_consumption_expenditure, y
= GDP)) +
+     geom_point(color = "blue", alpha = 0.6) +  # Add scatter points
+     geom_smooth(method = "lm", color = "red", se = FALSE) +  # Add
regression line
+     ggtitle("Scatter Plot of Household Expenditure vs GDP") +
+     xlab("Household Expenditure") +
+     ylab("GDP") +
+     theme_minimal()
```

[8]. This code creates a heatmap to visualize the correlation matrix of the dataset. It shows the correlation values between numerical features, with annotations and color coding (warm for positive, cool for negative correlations).

```
# Step 1.2: Correlation Analysis
plt.figure(figsize=(12, 8))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()
```

[9]. This code creates a time series object (Dataset_GDP_ts) in R from the Dataset_GDP dataframe. It removes the first column (presumably a date or index column) using [, -1], and then sets the time series to start in 2008 and end in 2024 with an annual frequency (frequency = 4). The print() function then outputs the created time series object to the console.

```
> Dataset_GDP_ts <- ts(Dataset_GDP[, -1], start = 2008, end = 2024, frequency =
4)
> print(Dataset_GDP_ts)
```

[10]. This R code uses the `auto.arima` function to automatically fit an ARIMA model to the GDP time series (`Dataset_GDP_ts`). It then generates forecasts for future values based on

the `test_data` length and plots the forecasted GDP values along with confidence intervals, providing a visual representation of the model's predictions.

```
>  library(forecast)
> arima_model <- auto.arima(Dataset_GDP_ts)
> summary(arima_model)
>  arima_forecast <- forecast(arima_model, h = length(test_data))
>  plot(arima_forecast)
```

[11]. The `checkresiduals(arima_model)` function checks the residuals of the ARIMA model to see if they are random (white noise). If there are patterns or autocorrelations, the model may need adjustments. It provides diagnostic plots and statistical tests to evaluate model adequacy.

```
> checkresiduals(arima_model)
```

[12]. This code forecasts GDP using ARIMA, incorporating Consumption as an exogenous variable. It extracts the GDP and Consumption data, creates time series for both from 2008 to 2024, and builds an ARIMA model where GDP is predicted using past values of both GDP and Consumption. It then forecasts the next 30 quarters of GDP, using the forecasted Consumption data, and visualizes the results, showing how GDP is expected to change with future Consumption.

```
> GDP_data <- Dataset_GDP$GDP  # Dependent variable (Ozone levels)
> consumption_data <- Dataset_GDP$Consumption # Explanatory variable
(Temperature)
> GDP_data <- Dataset_GDP$GDP  # Dependent variable (GDP levels)
> consumption_data <- Dataset_GDP$Consumption # Explanatory variable
(Consumption)
> gdp_ts <- ts(Dataset_GDP$GDP, frequency = 4, start = 2008, end = 2024)  #
Time series for GDP
> consumption_ts <- ts(Dataset_GDP$Consumption, frequency = 4, start = 2008,
end = 2024)  # Time series for Consumption
> arima_model_compsumtion <- auto.arima(gdp_ts, xreg = consumption_ts)
> forecast_horizon <- 30> consumption_forecast <- tail(consumption_data,
forecast_horizon)
> arima_forecast_consumption<- forecast(arima_model_compsumtion, xreg =
consumption_forecast, h = forecast_horizon)
> plot(arima_forecast_consumption, main = "ARIMA Forecast with Consumption as
Exogenous Variable", ylab = "GDP Levels", xlab = "Consumption")
```

[13]. Visualization for VAR Predictions in Python. A DataFrame, comparison_var_df, is created to hold the dates, actual GDP values, and forecasted GDP values. A line plot is generated to visualize the actual GDP (in blue) and the forecasted GDP (in red, dashed line) over time. The plot includes labels for the x-axis (Date) and y-axis (GDP), a title, and a legend to distinguish between actual and forecasted values.

```
import matplotlib.pyplot as plt
```

```
# Create a DataFrame for comparison
comparison_var_df = pd.DataFrame({
   'Date': data['Quarter'].iloc[split_index:].values,
   'Actual GDP': actual_gdp,
   'Forecasted GDP': predicted_gdp_var
})
# Plot the actual vs forecasted GDP
plt.figure(figsize=(12, 6))
plt.plot(comparison_var_df['Date'], comparison_var_df['Actual GDP'], label='Actual GDP', color='blue')
plt.plot(comparison_var_df['Date'], comparison_var_df['Forecasted GDP'], label='Forecasted GDP',
color='red', linestyle='--')
plt.xlabel('Date')
plt.ylabel('GDP')
plt.title('Actual vs Forecasted GDP using VAR')
plt.legend()   plt.show()
```

The code below visualizes the comparison between actual and forecasted GDP values using a Random Forest model:

- Create DataFrame: Combines dates, actual GDP (y_test), and forecasted GDP (y_pred_rf).
- Initialize Plot: Sets figure size to 12x6 inches.
- Plot Data: Plots actual GDP in blue and forecasted GDP in red dashed line.
- Labels and Title: Adds axis labels and a title.
- Legend and Show: Adds a legend and displays the plot.

```
# Visualization for Random Forest Predictions
comparison_rf_df = pd.DataFrame({
    'Date': data['Quarter'].iloc[split_index:].values,
    'Actual GDP': y_test.values,
    'Forecasted GDP': y_pred_rf
})
plt.figure(figsize=(12, 6))
plt.plot(comparison_rf_df['Date'], comparison_rf_df['Actual GDP'], label='Actual
GDP', color='blue')
plt.plot(comparison_rf_df['Date'], comparison_rf_df['Forecasted GDP'],
label='Forecasted GDP', color='red', linestyle='--')
plt.xlabel('Date')
plt.ylabel('GDP')
plt.title('Actual vs Forecasted GDP using Random Forest')
plt.legend()
plt.show()
```

[14]. Residual Analysis & Autocorrelation. Residuals: Computed for both models and visualized via histograms to assess distribution and bias.Autocorrelation (ACF): Checked for patterns in residuals to evaluate model performance over time.

```
# Residual Analysis for Random Forest
residuals_rf = y_test.values - y_pred_rf
plt.figure(figsize=(10, 5))
sns.histplot(residuals_rf, kde=True, color='blue')
plt.title('Residuals Distribution (Random Forest)')
plt.xlabel('Residuals')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()

# Residual Analysis for VAR
residuals_var = actual_gdp - predicted_gdp_var
plt.figure(figsize=(10, 5))
sns.histplot(residuals_var, kde=True, color='green')
plt.title('Residuals Distribution (VAR)')
plt.xlabel('Residuals')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()

# Autocorrelation for Residuals (Random Forest)
if len(residuals_rf) > 0:
    plot_acf(residuals_rf, lags=min(20, len(residuals_rf) - 1))
    plt.title('ACF of Residuals (Random Forest)')
    plt.grid(True)
    plt.show()

# Autocorrelation for Residuals (VAR)
if len(residuals_var) > 0:
    plot_acf(residuals_var, lags=min(20, len(residuals_var) - 1))
    plt.title('ACF of Residuals (VAR)')
    plt.grid(True)
    plt.show()
```

[15] . The `summary(arima_model)` provides key details about the ARIMA model, including coefficients, AIC, BIC, and residual diagnostics, helping assess the model's fit and performance.

```
> summary(arima_model)
```

[16] This code evaluates the performance of a Vector Autoregression (VAR) model by comparing the actual GDP values from the test dataset with the predicted GDP values from the VAR model forecast. It calculates three metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2) score.

```
# Evaluate VAR Model
actual_gdp = test_data['GDP'].values
predicted_gdp_var = forecast_var_df.iloc[:, 0].values
mae_var = mean_absolute_error(actual_gdp, predicted_gdp_var)
mse_var = mean_squared_error(actual_gdp, predicted_gdp_var)
r2_var = r2_score(actual_gdp, predicted_gdp_var)
```

[17] This code evaluates the performance of a Random Forest model by comparing the actual values (y_test) with the predicted values (y_pred_rf). It calculates three metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2) score. Finally, it prints these evaluation metrics.

```
# Evaluate Random Forest Model
mae_rf = mean_absolute_error(y_test, y_pred_rf)
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)
print(f"Random Forest Model Evaluation:\nMAE: {mae_rf:.2f}, MSE: {mse_rf:.2f},
R2: {r2_rf:.2f}")
```