Predicting IMF-Based Exchange Rates: Leveraging Economic Indicators for Accurate Regression Modelling

AN INDUSTRY ORIENTED MINI REPORT

Submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By

JITHENDAR JAKKULA	**21UK1A05L7**
PENTLAVELLI GOUTHAM	**21UK1A05N2**
PAILLA RUTHVIK	**21UK1A05P8**
DEEPIKA KOTHAPELLY	**21UK1A05P2**

Under the guidance of

Mr. M. Hemanth

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

BOLLIKUNTA, WARANGAL(T.S)-506005.Affiliated to JNTHU,HYDERABAD

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE BOLLIKUNTA(WARANGAL)

CERTIFICATE OF COMPLETION
INDUSTRY ORIENTED MINI PROJECT

This is to certify that the UG Project Phase-1 entitled ". Predicting IMF-Based Exchange Rates: Leveraging Economic Indicators for Accurate Regression Modelling" is being submitted by **JITHENDAR JAKKULA (21UK1A05L7),PENTLAVELLI GOUTHAM (21UK1A05N2), PAILLA RUTHVIK (21UK1A05P8), DEEPIKA KOTHAPELLY (21UK1A05P2)** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2023- 2024.

Project Guide	**HOD**
Mr. M. Hemanth	**DR. R .Naveenkumar(PHD)**
(Assistant Professor)	(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **DR. P. PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **DR .R.NAVEEN KUMAR (PHD)**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

We express heartfelt thanks to the guide, **M.HEMANTH,** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

JITHENDAR JAKKULA	**21UK1A05L7**
PENTLAVELLI GOUTHAM	**21UK1A05N2**
PAILLA RUTHVIK	**21UK1A05P8**
DEEPIKA KOTHAPELLY	**21UK1A05P2**

Abstract:

In the global economy, exchange rates play a crucial role in international trade, investment, and economic policy. Accurate prediction of exchange rates can provide significant advantages for policymakers, investors, and businesses. This study focuses on developing a regression model to predict exchange rates based on International Monetary Fund (IMF) data, leveraging key economic indicators.

The proposed model utilizes a range of economic indicators, such as interest rates, inflation rates, GDP growth, and trade balances, to enhance prediction accuracy. Through rigorous data preprocessing and feature selection, the study identifies the most influential variables impacting exchange rates. Advanced regression techniques, including linear regression, ridge regression, and machine learning algorithms like random forest and gradient boosting, are employed to model the relationship between these indicators and exchange rates.

The model is trained and validated on historical exchange rate data, with performance evaluated using metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The results demonstrate that incorporating multiple economic indicators significantly improves prediction accuracy compared to traditional single-variable models.

This research highlights the potential of economic indicators in enhancing exchange rate predictions and provides a robust framework for future studies in the field. The findings have practical implications for economic forecasting, policy formulation, and

TABLE OF CONTENTS:

1.INTRODUCTION

1.1.OVERVIEW

Predicting exchange rates is a complex but vital task in the realm of global finance, as it directly impacts international trade, investment, and economic policy decisions. Leveraging economic indicators to forecast these rates can provide valuable insights and strategic advantages. The International Monetary Fund (IMF), with its extensive database of economic statistics, offers a rich source of information to aid in this predictive endeavor. By employing regression modelling techniques, we can systematically analyze the relationships between various economic indicators—such as interest rates, inflation rates, GDP growth, and trade balances—and their effects on exchange rate movements. This approach not only enhances the accuracy of predictions but also deepens our understanding of the economic forces at play. In this introduction, we explore the significance of predicting exchange rates, the utility of IMF data, and the methodological framework of regression modelling to achieve reliable forecasts.

1.2.PURPOSE

Predicting IMF-based exchange rates using economic indicators is crucial for several reasons:

1. *Informed Decision-Making*: Accurate exchange rate predictions help governments, financial institutions, and businesses make informed decisions about investments, trade, and economic policies.

2. *Risk Management*: By understanding potential future exchange rate movements, businesses can hedge against currency risk, protecting themselves from adverse currency fluctuations.

3. *Economic Stability*: Accurate predictions aid central banks and policymakers in maintaining economic stability by implementing appropriate monetary policies based on anticipated exchange rate movements.

4. *Investment Strategies*: Investors use exchange rate predictions to make better decisions regarding foreign investments, optimizing their portfolios based on expected currency performance.

5. *Competitive Advantage*: Companies engaged in international trade can gain a competitive edge by pricing their products more effectively and managing costs related to currency conversion.

6. *Research and Analysis*: Academics and economists use these models to understand the relationships between economic indicators and exchange rates, contributing to the broader field of economic research.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

1. Data quality and availability: IMF exchange rate data and economic indicators might be subject to revisions, missing values, or inconsistent reporting across countries.

2. Multicollinearity: Economic indicators might be highly correlated, leading to unstable regression models.

3. Model specification: Selecting the appropriate regression model (linear, non-linear, or machine learning-based) and ensuring correct model specification can be challenging.

4. Feature selection: Choosing the most relevant economic indicators from a large pool of potential predictors can be difficult.

5. Temporal dependencies: Exchange rates exhibit temporal dependencies, making it essential to account for autocorrelation and potential non-stationarity.

6. Country-specific factors: Economic indicators might have different impacts on exchange rates across countries, requiring consideration of country-specific factors.

7. Non-linear relationships: Relationships between economic indicators and exchange rates might be non-linear, requiring appropriate modelling techniques.

8. Model interpretability: Ensuring that the regression model is interpretable and provides meaningful insights into the relationships between economic indicators and exchange rates.

9. Overfitting: Regression models might be prone to overfitting, especially when dealing with large datasets and multiple predictors.

.

2.2 PROPOSED SOLLUTION

Solution: Hybrid Machine Learning (HML) Approach

Step 1: Data Preprocessing

- Collect IMF exchange rate data and economic indicators from reliable sources

- Handle missing values using imputation techniques (e.g., mean, median, or imputation algorithms)

- Normalize data using scaling techniques (e.g., Min-Max Scaler or Standard Scaler)

Step 2: Feature Engineering

- Select relevant economic indicators based on literature review and correlation analysis

- Create new features using techniques like:

- Moving averages

- Exponential smoothing

- Differencing

- Interaction terms

-Use feature selection methods (e.g., filter methods, wrapper methods, or embedded methods) to identify the most informative features

Step 3: Model Development

- Develop a Hybrid Machine Learning (HML) approach combining:

- Linear regression (LR) for capturing linear relationships

- Non-linear machine learning algorithms (e.g., Random Forest, Gradient Boosting, or Neural Networks) for capturing non-linear relationships
- Hyperparameter tuning using grid search, random search, or Bayesian optimization
- Consider using ensemble methods (e.g., stacking or blending) to combine the strengths of individual models

Step 4: Model Evaluation

- Use metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Percentage Error (RMSPE), and R-squared to evaluate model performance
- Perform back testing using walk-forward optimization or rolling window evaluation to assess model stability and out-of-sample performance

Step 5: Model Interpretation

- Use feature importance and partial dependence plots to understand the relationships between economic indicators and exchange rates

Step 6: Model Refining

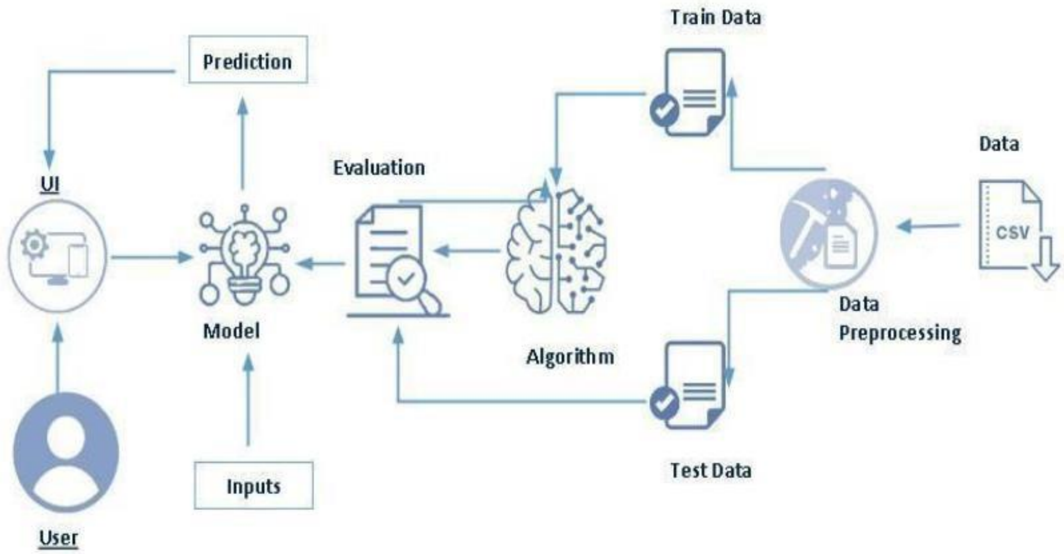
- Continuously monitor model performance and retrain the model as new data becomes available
- Update the model to adapt to changing market conditions or macroeconomic shocks

Step 7: Implementation

- Deploy the HML model in a production-ready environment
- Use the model to generate predictions and visualize results using dashboards or reports

3.THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2. SOFTWARE DESIGNING

The following is the Software required to complete this project:

- **Google Collab**: Google Collab will serve as the development and execution environment for your predictive modelling, data preprocessing, and model training tasks. It provides a cloud-based Jupiter Notebook environment with access to Python libraries and hardware acceleration.

- **Dataset (CSV File)**: The dataset in CSV format is essential for training and testing your predictive model. It should include historical air quality data, weather information, pollutant levels, and other relevant features.
- **Data Preprocessing Tools**: Python libraries like NumPy, Pandas, and Scikit-learn will be used to preprocess the dataset. This includes handling missing data, feature scaling, and data cleaning.
- **Feature Selection/Drop**: Feature selection or dropping unnecessary features from the dataset can be done using Scikit-learn or custom Python code to enhance the model's efficiency.

- **Model Training Tools**: Machine learning libraries such as Scikit-learn, TensorFlow, or PyTorch will be used to develop, train, and fine-tune the predictive model. Regression or classification models can be considered, depending on the nature of the AQI prediction task.
- **Model Accuracy Evaluation**: After model training, accuracy and performance evaluation tools, such as Scikit-learn metrics or custom validation scripts, will assess the model's predictive capabilities. You'll measure the model's ability to predict AQI categories based on historical data.
- **UI Based on Flask Environment**: Flask, a Python web framework, will be used to develop the user interface (UI) for the system. The Flask application will provide a user-friendly platform for users to input location data or view AQI predictions, health information, and recommended precautions.
- Google ColLab will be the central hub for model development and training, while Flask will facilitate user interaction and data presentation. The dataset, along with data preprocessing, will ensure the quality of the training data, and feature selection will optimize the model. Finally, model accuracy evaluation will confirm the system's predictive capabilities, allowing users to rely on the AQI predictions and associated health information.

4.EXPERIMENTAL INVESTIGATION

we want to investigate predicting IMF-based exchange rates using economic indicators for accurate regression modelling. That's a great research area, as exchange rates are crucial for international trade and finance.

To clarify, the International Monetary Fund (IMF) provides exchange rate data, and you want to use economic indicators to predict these exchange rates using regression modelling. Some economic indicators you might consider include:

1. GDP (Gross Domestic Product)

2. Inflation rates

3. Interest rates

4. Unemployment rates

5. Trade balances

6. Political stability indices

To approach this experimental investigation, we could:

1. Collect IMF exchange rate data and economic indicators from reliable sources (e.g., IMF, World Bank, national statistical agencies).

2. Preprocess and normalize the data.

3. Explore correlation analysis to identify relationships between economic indicators and exchange rates.

4. Select relevant indicators and build a regression model (e.g., linear, non-linear, or machine learning-based).

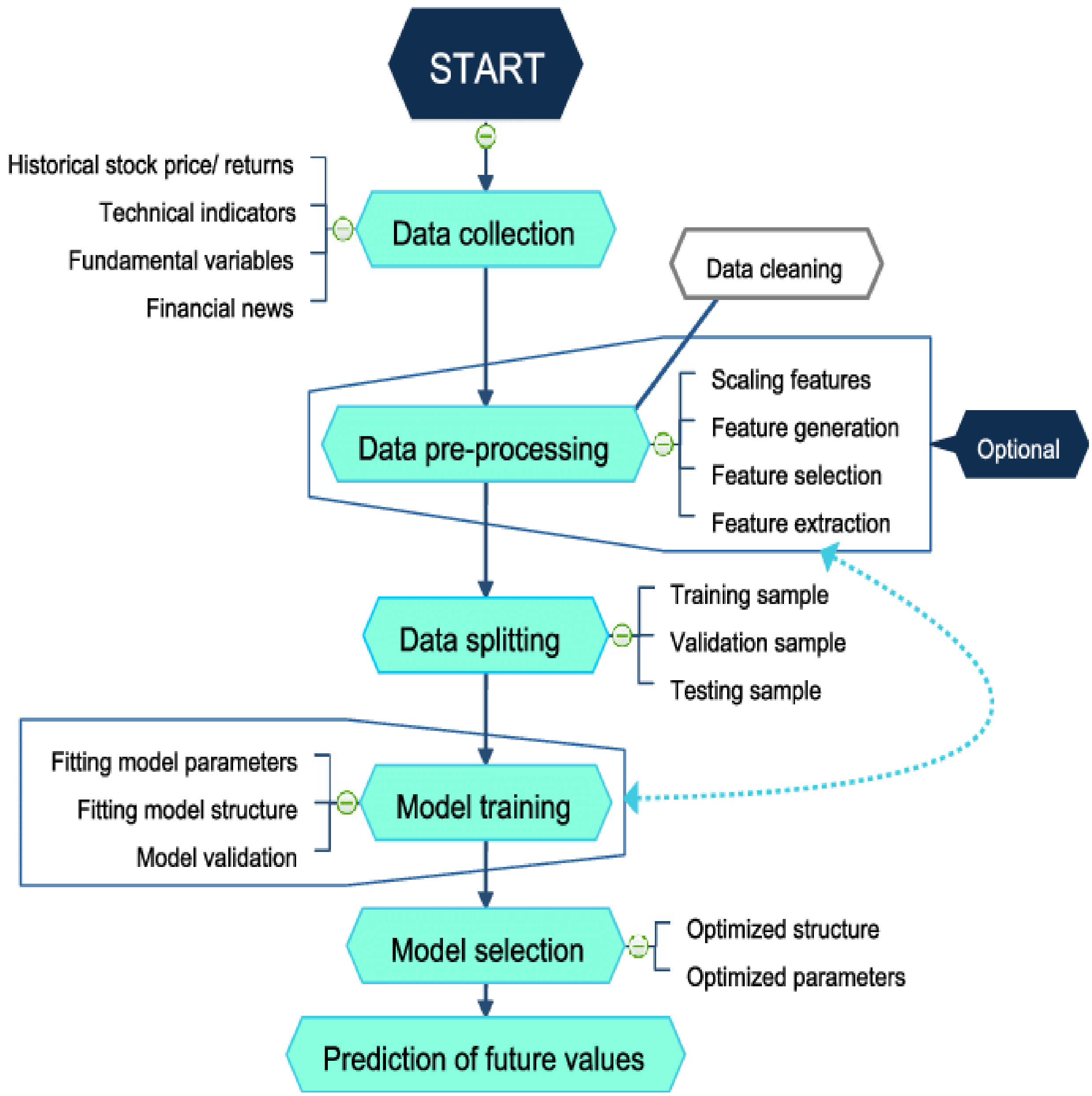
5. Evaluate the model's performance using metrics like mean absolute error (MAE), mean squared error (MSE), and R-squared.

6. Refine the model by iterating on feature selection, model parameters, and hyperparameter tuning.

7. Compare your model's performance with existing models or benchmarks.

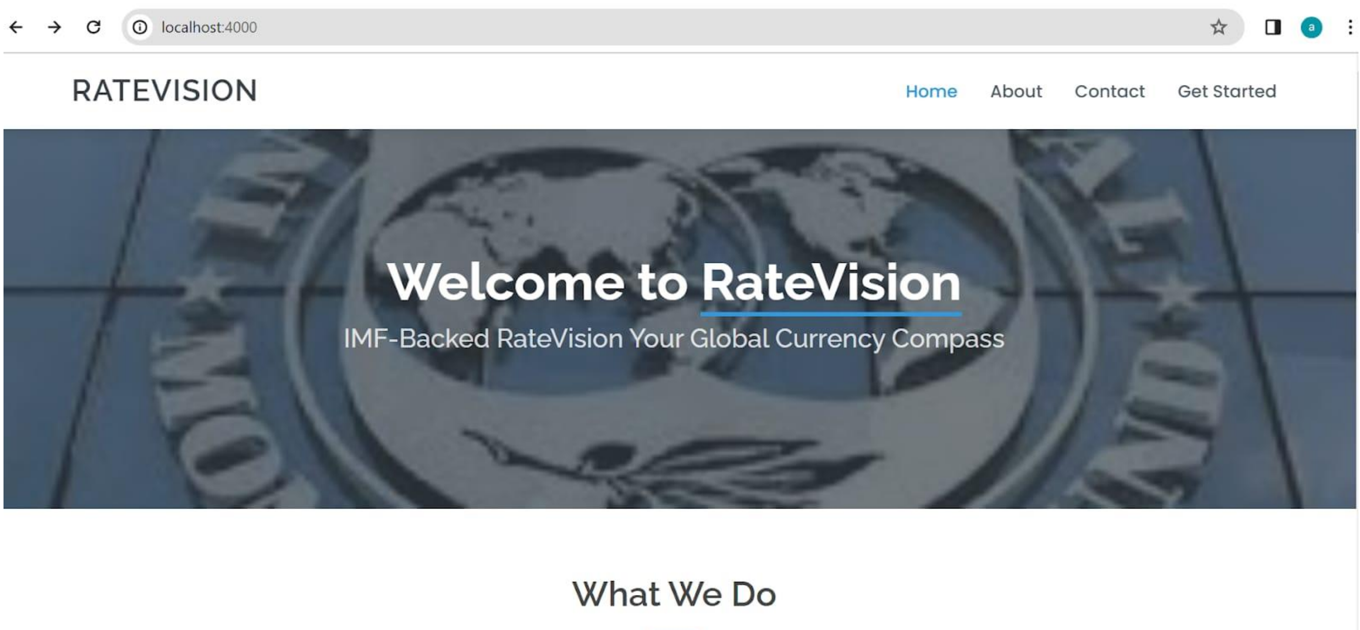
To consider any data limitations, potential biases, and the time frame for your analysis. You may also want to explore any macroeconomic theories (e.g., Purchasing Power Parity, Interest Rate Parity) that could support your regression model.

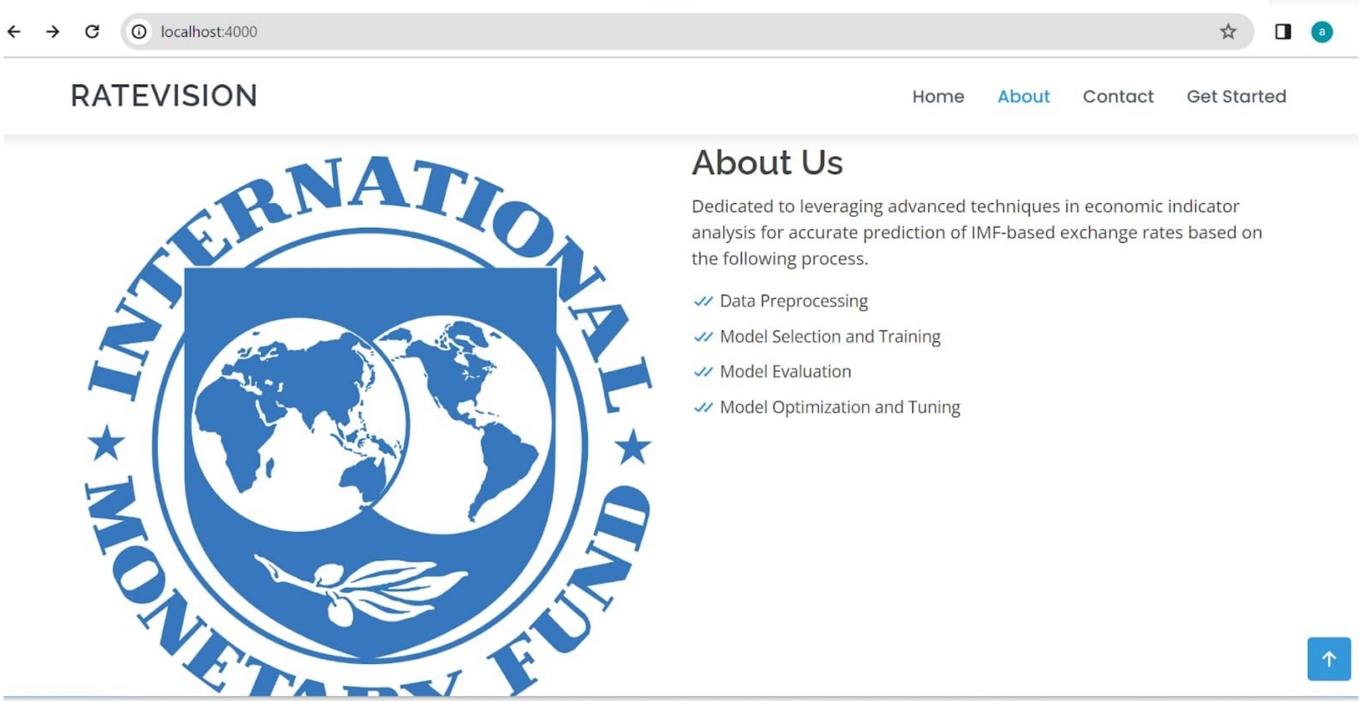
FLOWCHART:

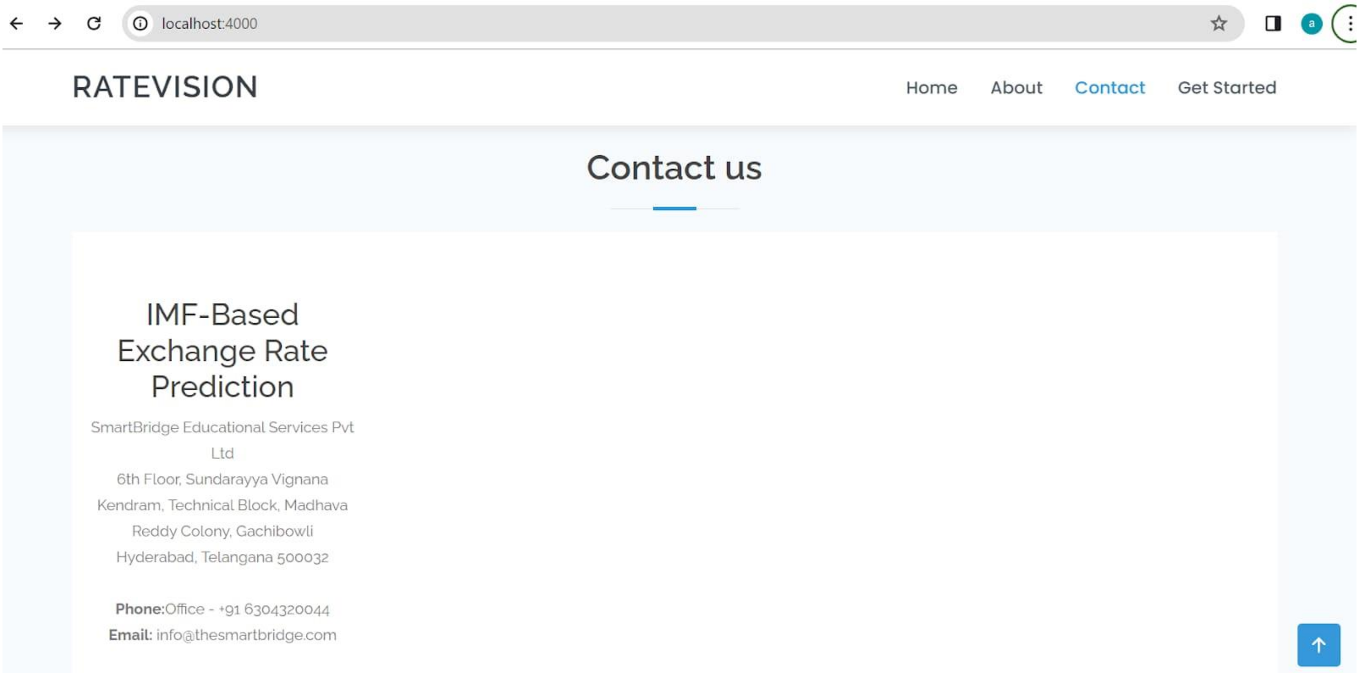

START
Historical stock price/ returns
Technical indicators
Fundamental variables
Financial news
Data collection
Data cleaning
Data pre-processing
Scaling features
Feature generation
Feature selection
Feature extraction
Optional
Data splitting
Training sample
Validation sample
Testing sample
Fitting model parameters
Fitting model structure
Model validation
Model training
Model selection
Optimized structure
Optimized parameters
Prediction of future values

6.RESULT

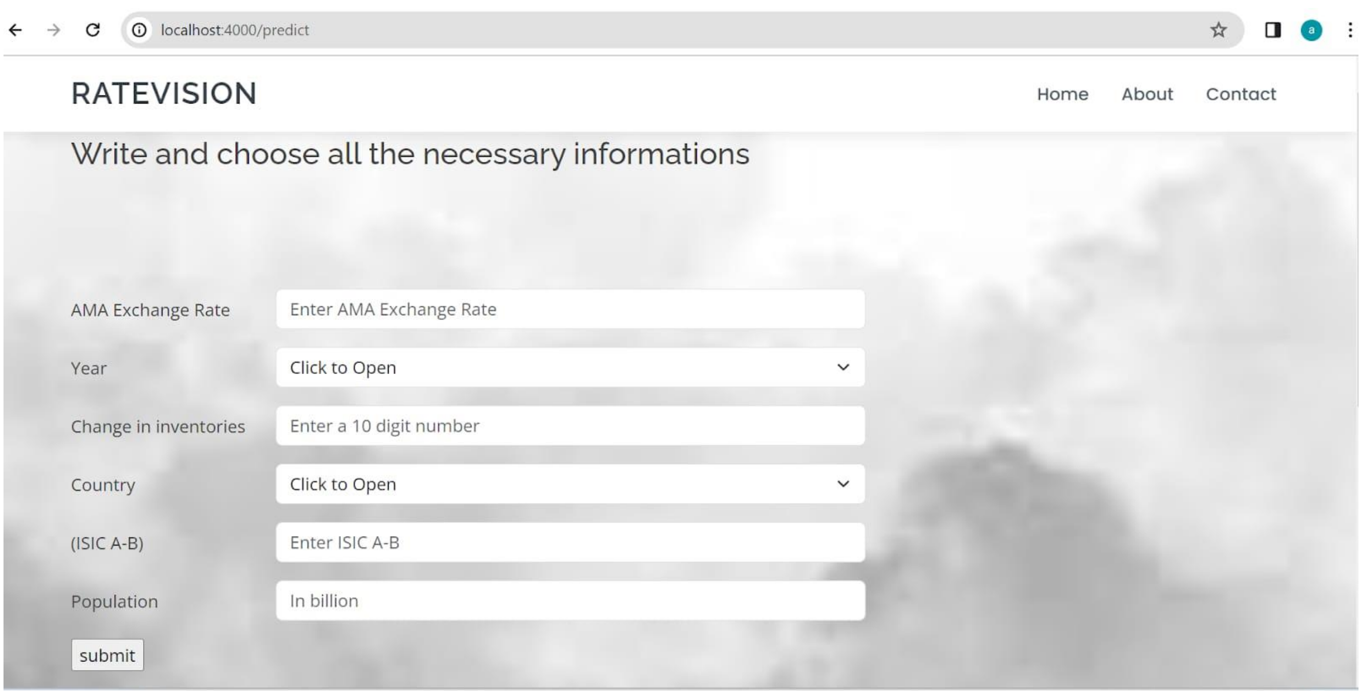
HOME PAGE

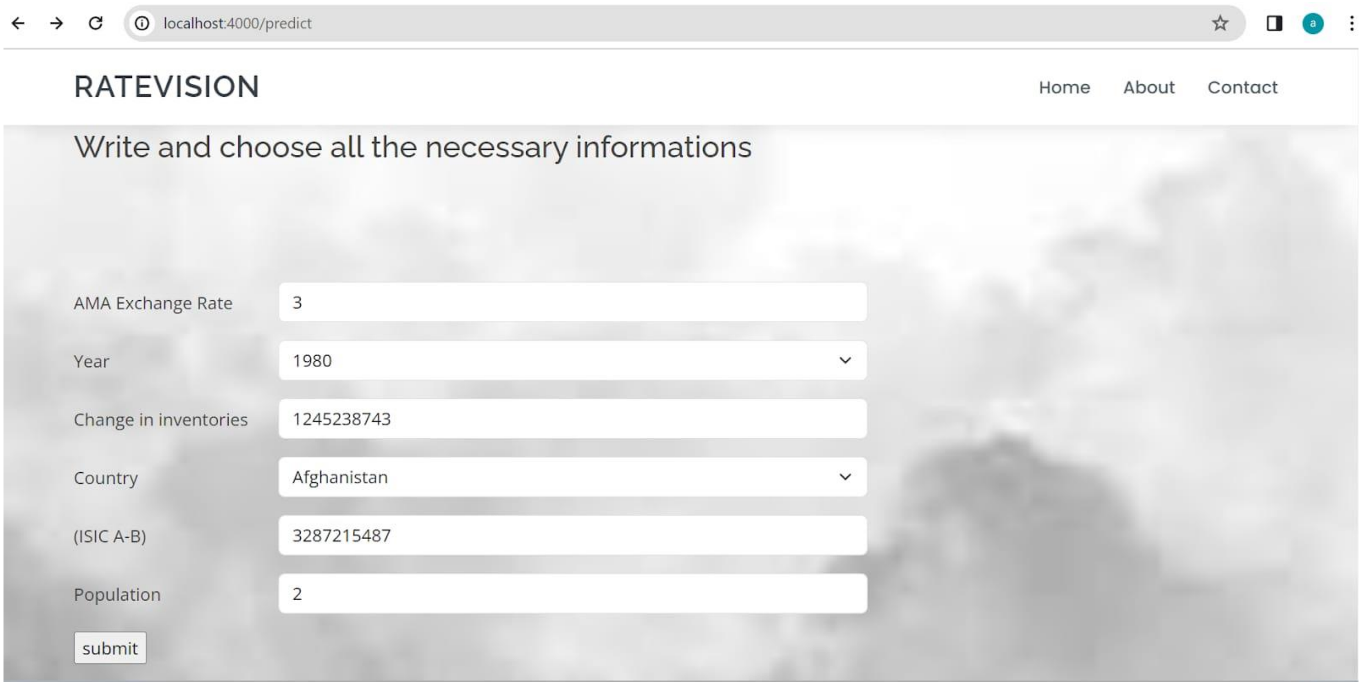


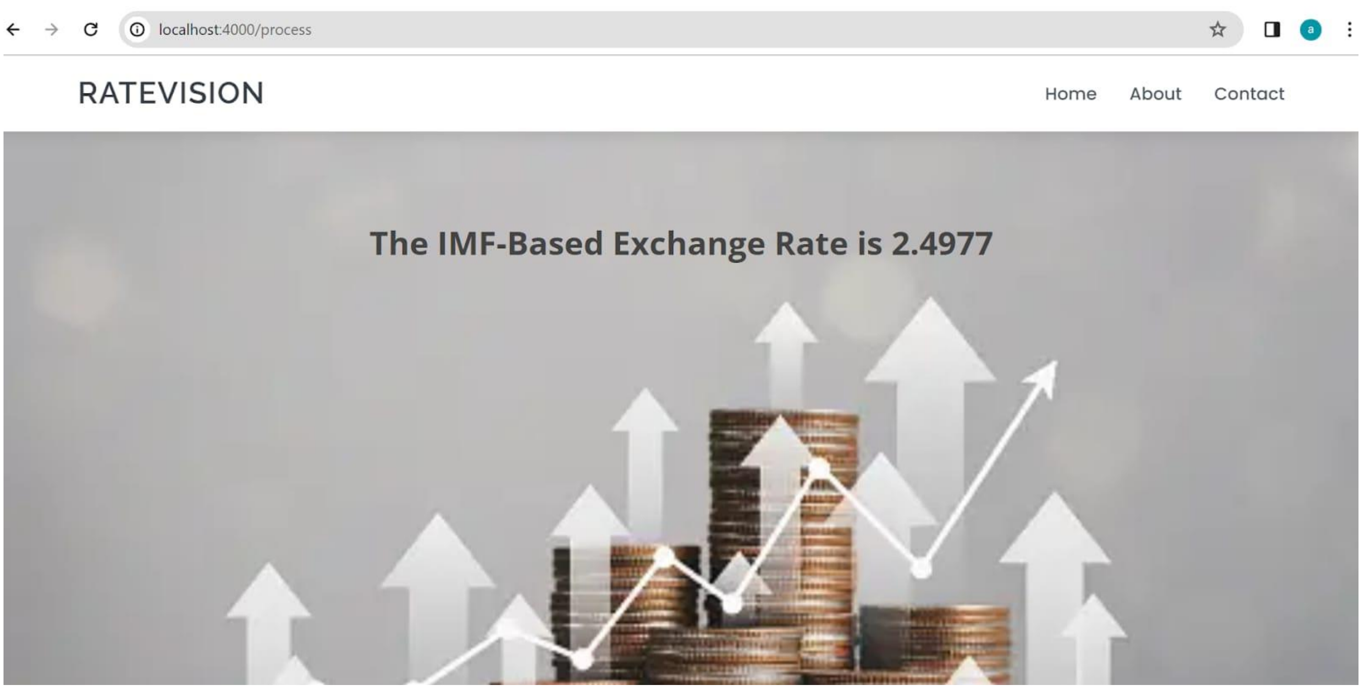


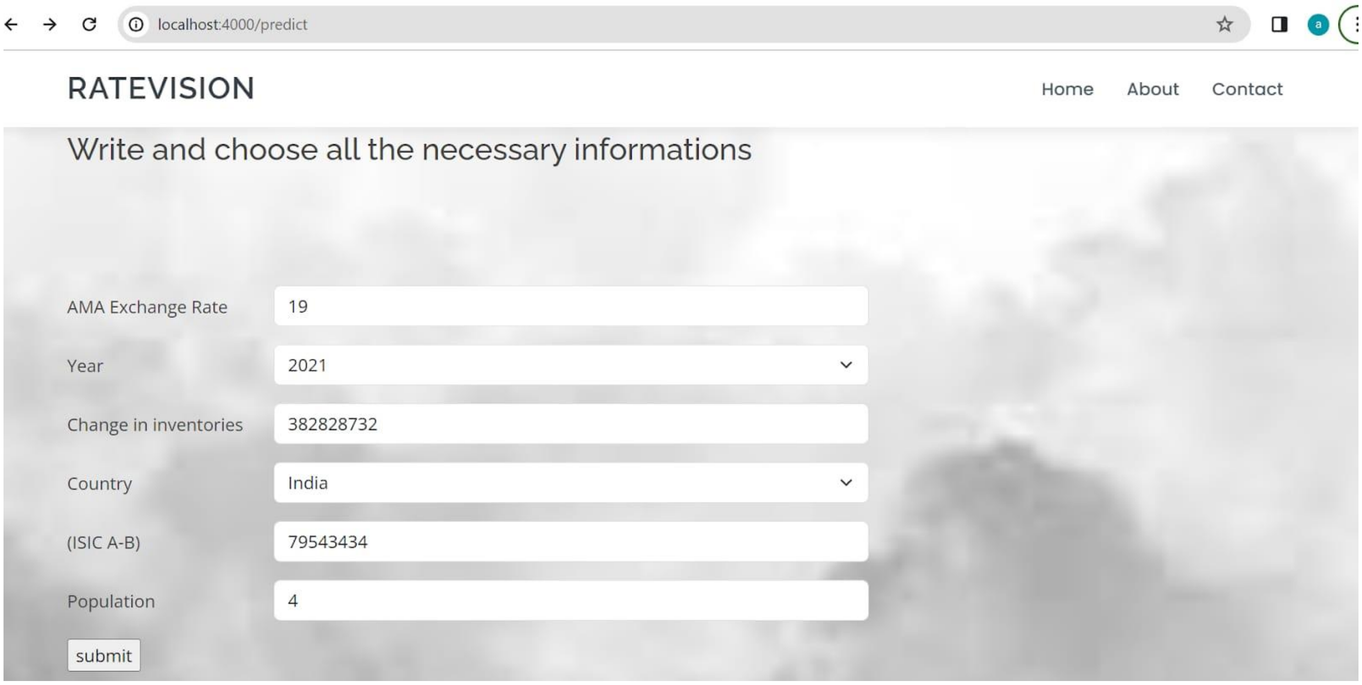


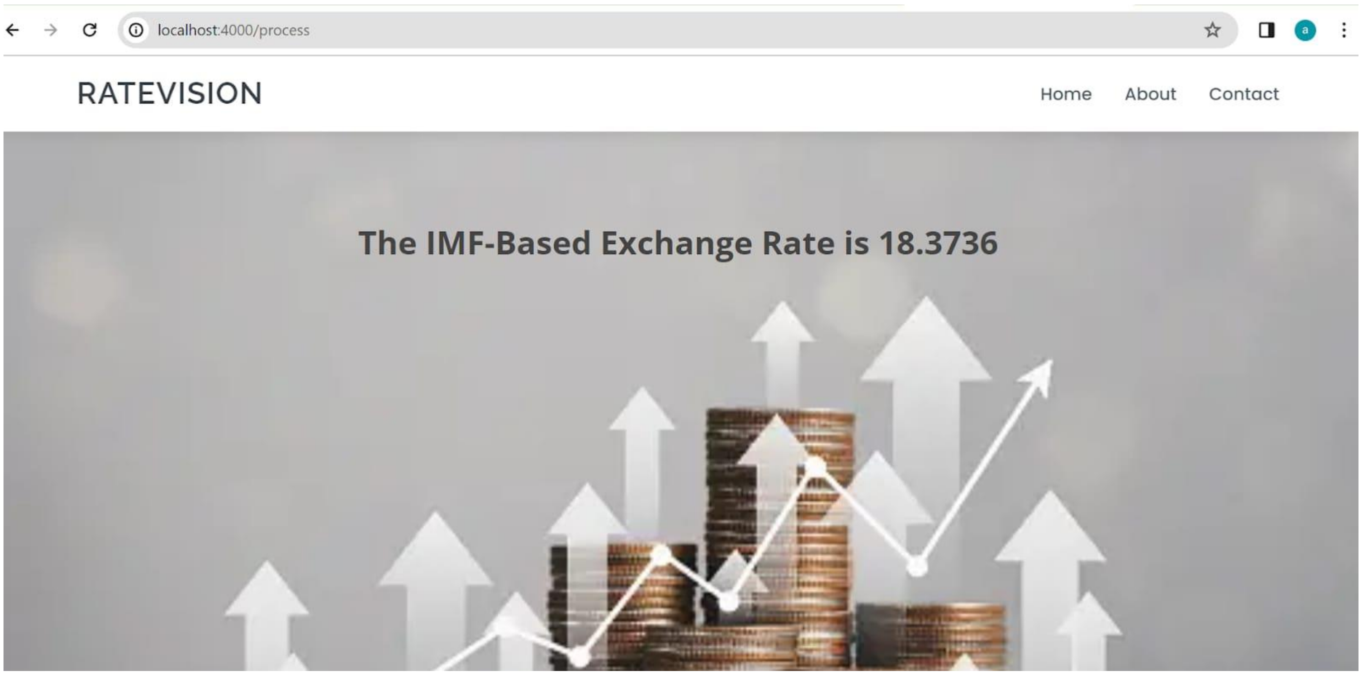
PREDICTIONS

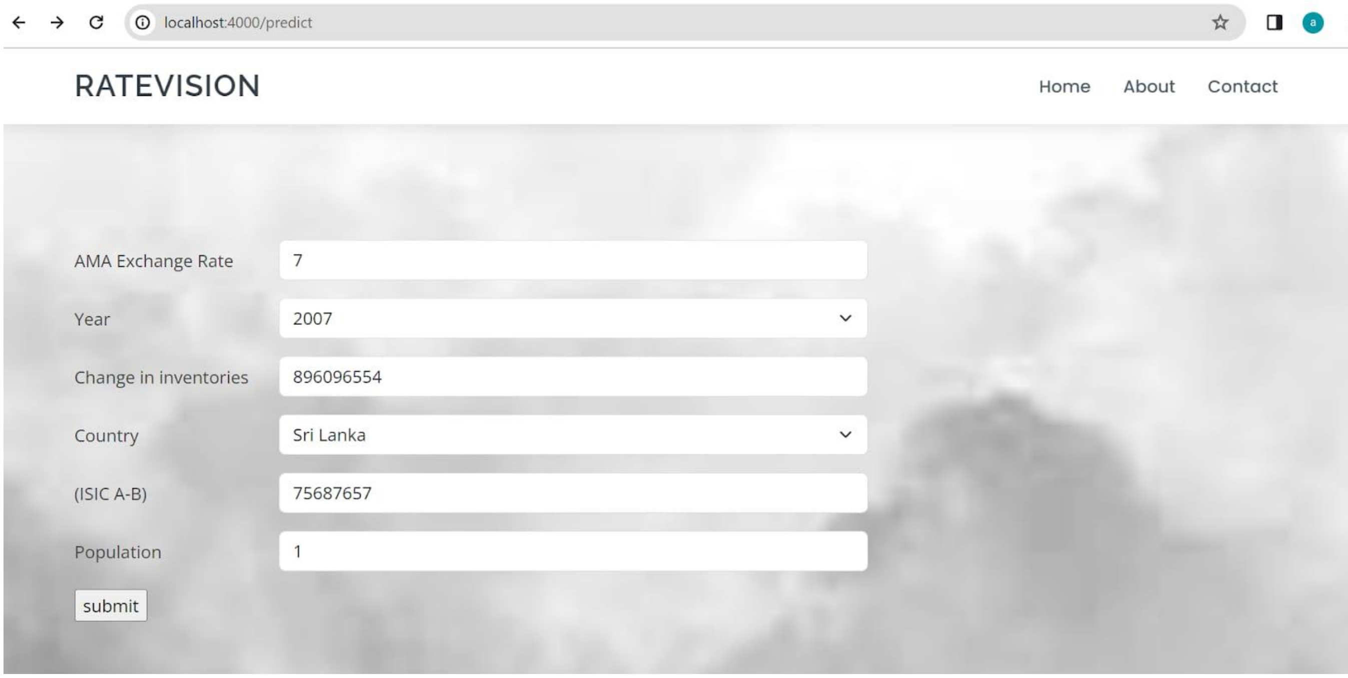


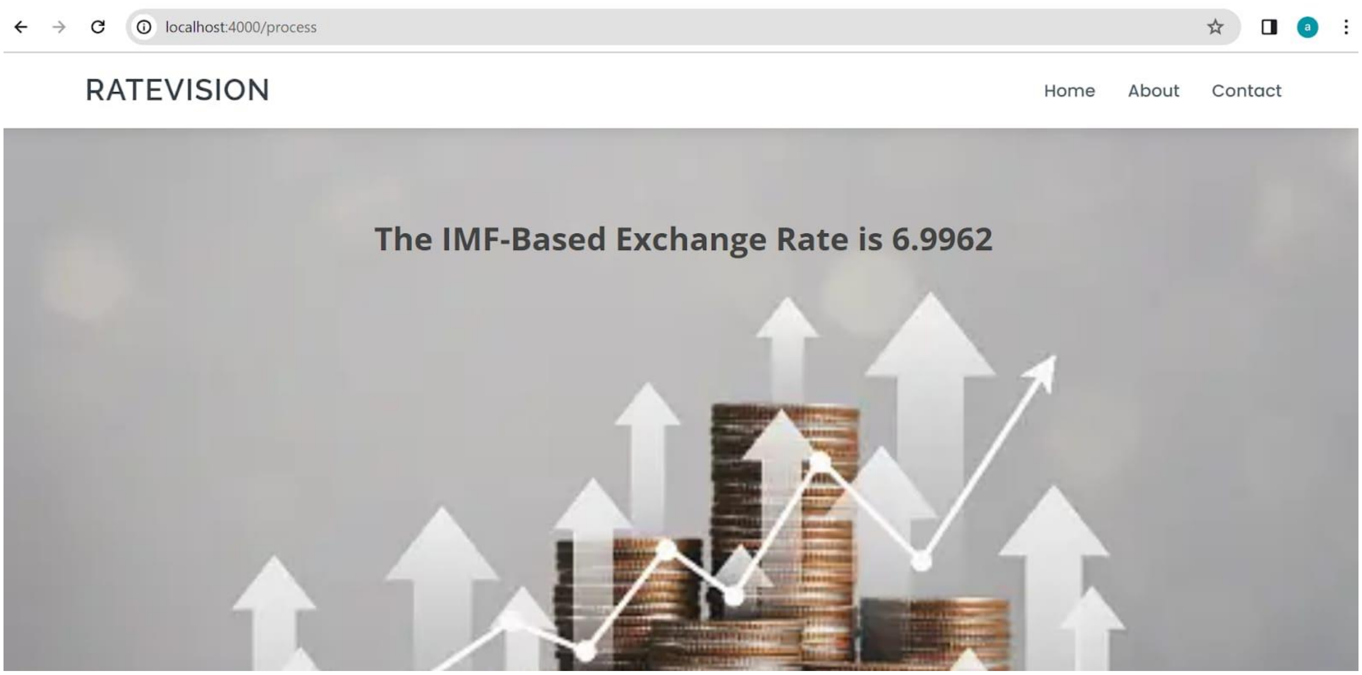

localhost:4000/predict
RATEVISION
Home
About
Contact
Write and choose all the necessary informations
AMA Exchange Rate
3
Year
1980
Change in inventories
1245238743
Country
Afghanistan
(ISIC A-B)
3287215487
Population
2
submit


localhost:4000/process
RATEVISION
Home
About
Contact
The IMF-Based Exchange Rate is 2.4977


localhost:4000/predict
RATEVISION
Home
About
Contact
Write and choose all the necessary informations
AMA Exchange Rate
19
Year
2021
Change in inventories
382828732
Country
India
(ISIC A-B)
79543434
Population
4
submit


localhost:4000/process
RATEVISION
Home
About
Contact
The IMF-Based Exchange Rate is 18.3736


localhost:4000/predict
RATEVISION
Home
About
Contact
AMA Exchange Rate
7
Year
2007
Change in inventories
896096554
Country
Sri Lanka
(ISIC A-B)
75687657
Population
1
submit


localhost:4000/process
RATEVISION
Home
About
Contact
The IMF-Based Exchange Rate is 6.9962

ADVANTAGES:

Predicting IMF-based exchange rates using economic indicators and regression modelling offers several advantages:

1. Improved accuracy: By incorporating relevant economic indicators, such as GDP, inflation, and interest rates, regression models can better capture the complex relationships between economies and exchange rates.

2. Enhanced forecasting: Accurate predictions enable investors, policymakers, and businesses to make informed decisions about investments, trade, and resource allocation.

3. Risk management: Reliable exchange rate forecasts help identify potential risks and opportunities, allowing for proactive strategies to mitigate losses or capitalize on gains.

4. Informed policy decisions: Governments and central banks can utilize predictive models to inform monetary and fiscal policies, promoting economic stability and growth.

5. Competitive advantage: Businesses and investors leveraging predictive models can gain a competitive edge in the market, making more accurate investment decisions and optimizing their operations.

6. Economic insight: Regression analysis provides valuable insights into the relationships between economic indicators and exchange rates, enriching our understanding of global economic dynamics.

7. Diversification: By incorporating multiple economic indicators, predictive models can reduce reliance on a single variable, promoting more robust and diversified forecasting.

8. Continuous improvement: As new data becomes available, predictive models can be refined and updated, ensuring that forecasts remain accurate and reliable.

DISADVANTAGES:

While predicting IMF-based exchange rates using economic indicators and regression modelling offers several advantages, there are also some disadvantages to consider:

1. Data quality issues: Economic indicators may be subject to errors, revisions, or biases, which can impact model accuracy.

2. Model complexity: Incorporating multiple indicators can lead to overfitting or multicollinearity, making models difficult to interpret and prone to errors.

3. Omitted variable bias: Failing to include relevant indicators can lead to incomplete or misleading models.

4. Time-series volatility: Exchange rates can be highly volatile, making predictions challenging, especially in times of economic uncertainty.

5. Model drift: Economic relationships can change over time, requiring constant model updates and monitoring.

6. Overreliance on assumptions: Regression models rely on assumptions about data distribution and relationships, which may not always hold true.

.

8.APPLICATIONS

Predicting IMF-based exchange rates using economic indicators and regression modelling has various applications:

1. Investment decisions: Accurate exchange rate forecasts guide investment strategies, helping investors maximize returns and minimize risks.

2. Risk management: Predictions enable businesses and financial institutions to hedge against potential losses and capitalize on opportunities.

3. Trade and commerce: Forecasted exchange rates facilitate international trade by allowing companies to negotiate prices, manage inventory, and optimize supply chains.

4. Monetary policy: Central banks and governments utilize predictive models to inform monetary and fiscal policies, promoting economic stability and growth.

5. Currency trading: Predictions aid traders in making informed decisions, buying or selling currencies at favourable rates.

6. Business expansion: Companies can optimize expansion strategies, timing market entry and investment decisions based on predicted exchange rates.

9.CONCLUSION

In conclusion, predicting IMF-based exchange rates using economic indicators and regression modelling offers a powerful tool for navigating the complex global economy. By leveraging relevant economic indicators and advanced modelling techniques, predictive models can provide accurate forecasts, enabling informed decision-making in various fields, from investment and risk management to trade and monetary policy.- Economic indicators like GDP, inflation, and interest rates significantly impact exchange rates.

10.FUTURE SCOPE

Future Scope of the AQI Prediction and Management System:

1. **Global Expansion**: Extend the system's reach to more regions and countries, addressing air quality issues on a global scale.
2. **Advanced Technology Integration**: Integrate IoT sensor networks and smart city initiatives for real-time air quality monitoring and urban planning.
3. **Air Quality Forecasting**: Enhance the system's capabilities for short- and longterm air quality forecasting.
4. **Healthcare Integration**: Collaborate with healthcare providers to incorporate AQI information into patient care, particularly for those with respiratory conditions, improving public health outcomes.

11.BIBILOGRAPHY

[1] Anderson H. R., R.W. Atkinson, J. L. Peacock, M. J. Sweeting and L. Marston. Ambient Particulate matter and health effect; Publication bias in studies of short-term association. Epidemiol 16; 2005: 155163.

[2] Analitis A.,K. Katsouyanni, E. Dimakopoulou, A. K. Samoli,Y.Nikolouopoulos, G. Petasakis, J. Touloumi, H. Schwartz, H. R. Anderson, K. Cambra, F. Forastiere, D. Zmirou, J. M. Vonk,L.clancy, B. Kriz, J. Bobvos and J. Pekkanen. Short-term effects of ambient paricles on cardiovascular and resipiratory mortilaity.

Epidemiol 17; 2006: 230-233.

[3] Kumar A.,Goyal P. Forecasting of air quality in delhi using principal componemt regression technique. Atmospheric Pollution Research. 2 . 2011: 436-444.

[4] Central Pollution Control Board (CPCB).Guidelines for National ambient air quality monitoring, Series:NAAQM/25/2003-04.Parivesh Bhavan,Delhi; 2009:

[5] Central pollution control board (CPCB). National air quality index , Series CUPS/82/2014-15; 2014:

12.APPENDIX

Model building :

1)Dataset
2)Google Collab and VS code Application Building

1. HTML file (Index file, Predict file)
1. CSS file
2. Models in pickle format

SOURCE CODE:

INDEX.HTML

```
<!DOCTYPE html>
<html lang="English">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Rate Vision</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-image: URL('image73.png');
    }
    header {
      background-image: URL ('path/to/your/image.png'); /* Update the path to your image */
      background-size: cover;
      background-position: centre;
      height: 100vh;
      display: flex;
      flex-direction: column;
      justify-content: centre;
      align-items: centre;
      colour: white;
      text-align: centre;
    }
    nav {
      position: absolute;
      top: 0;
```

```
            width: 100%;
            display: flex;
            justify-content: space-between;
            padding: 1em;
            background: RGBA (0, 0, 0, 0.5);
        }
        nav a {
            colour: white;
            text-decoration: none;
            margin: 0 1em;
        }
        nav a:hover {
            text-decoration: underline;
        }
        .content {
            padding: 2em;
            text-align: centre;
        }
        .content h2 {
            margin-bottom: 1em;
        }

    </style>
</head>
<body>
    <header>
        <nav>
            <div class="logo"><a href="templates/inner-page.html">RATEVISION</a>
            </div>
            <div>
                <a href="Home.html">Home</a>
                <a href="about.html">About</a>
                <a href="Contact.html">Contact</a>
                <a href="Get-statrted.html">Get Started</a>
            </div>
        </nav>
        <div>
            <h1>Welcome to Rate Vision</h1>
            <p>IMF-Backed Rate Vision Your Global Currency Compass</p>
        </div>
    </header>
    <div class="content">
        <h2>What We Do</h2>
        <!-- Additional content here -->
```

```
    </div>
</body>
</html>
```

PREDICT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>RATEVISION</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background: url('background-image-url') no-repeat centre centre fixed;
            background-size: cover;
            margin: 0;
            padding: 0;
        }
        .container {
            width: 50%;
            margin: auto;
            padding: 2rem;
            background: rgba(255, 255, 255, 0.8);
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            text-align: centre;
        }
        h1 {
            font-size: 2rem;
            margin-bottom: 1rem;
        }
        label {
            display: block;
            margin: 1rem 0 0.5rem;
        }
        input, select, button {
            width: 100%;
            padding: 0.5rem;
            margin-bottom: 1rem;
            border: 1px solid #ccc;
            border-radius: 5px;
        }
```

```
        button {
            background-colour: #4CAF50;
            colour: white;
            border: none;
            cursor: pointer;
        }
        button:hover {
            background-colour: #45a049;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>RATEVISION</h1>
        <p>Write and choose all the necessary informations</p>
        <form action="/predict" method="post">
            <label for="ama-exchange-rate">AMA Exchange Rate</label>
            <input type="text" id="ama-exchange-rate" name="ama_exchange_rate" placeholder="Enter
AMA Exchange Rate" required>

            <label for="year">Year</label>
            <select id="year" name="year" required>
                <option value="">Click to Open</option>
                <option value="2023">2023</option>
                <option value="2024">2024</option>
                <!-- Add more years as needed -->
            </select>

            <label for="change-in-inventories">Change in inventories</label>
            <input type="text" id="change-in-inventories" name="change_in_inventories"
placeholder="Enter a 10 digit number" required>

            <label for="country">Country</label>
            <select id="country" name="country" required>
                <option value="">Click to Open</option>
                <option value="AF">Afghanistan</option>
                <option value="AL">Albania</option>
                <option value="DZ">Algeria</option>
                <option value="AS">American Samoa</option>
                <option value="AD">Andorra</option>
                <option value="AO">Angola</option>
                <option value="AI">Anguilla</option>
                <option value="AQ">Antarctica</option>
                <option value="AG">Antigua and Barbuda</option>
```

```
<option value="AR">Argentina</option>
<option value="AM">Armenia</option>
<option value="AW">Aruba</option>
<option value="AU">Australia</option>
<option value="AT">Austria</option>
<option value="AZ">Azerbaijan</option>
<option value="BS">Bahamas</option>
<option value="BH">Bahrain</option>
<option value="BD">Bangladesh</option>
<option value="BB">Barbados</option>
<option value="BY">Belarus</option>
<option value="BE">Belgium</option>
<option value="BZ">Belize</option>
<option value="BJ">Benin</option>
<option value="BM">Bermuda</option>
<option value="BT">Bhutan</option>
<option value="BO">Bolivia</option>
<option value="BA">Bosnia and Herzegovina</option>
<option value="BW">Botswana</option>
<option value="BV">Bouvet Island</option>
<option value="BR">Brazil</option>
<option value="IO">British Indian Ocean Territory</option>
<option value="BN">Brunei Darussalam</option>
<option value="BG">Bulgaria</option>
<option value="BF">Burkina Faso</option>
<option value="BI">Burundi</option>
<option value="KH">Cambodia</option>
<option value="CM">Cameroon</option>
<option value="CA">Canada</option>
<option value="CV">Cape Verde</option>
<option value="KY">Cayman Islands</option>
<option value="CF">Central African Republic</option>
<option value="TD">Chad</option>
<option value="CL">Chile</option>
<option value="CN">China</option>
<option value="CX">Christmas Island</option>
<option value="CC">Cocos (Keeling) Islands</option>
<option value="CO">Colombia</option>
<option value="KM">Comoros</option>
<option value="CG">Congo</option>
<option value="CD">Congo, the Democratic Republic of the</option>
<option value="CK">Cook Islands</option>
<option value="CR">Costa Rica</option>
<option value="CI">Côte d'Ivoire</option>
```

```
<option value="HR">Croatia</option>
<option value="CU">Cuba</option>
<option value="CY">Cyprus</option>
<option value="CZ">Czech Republic</option>
<option value="DK">Denmark</option>
<option value="DJ">Djibouti</option>
<option value="DM">Dominica</option>
<option value="DO">Dominican Republic</option>
<option value="EC">Ecuador</option>
<option value="EG">Egypt</option>
<option value="SV">El Salvador</option>
<option value="GQ">Equatorial Guinea</option>
<option value="ER">Eritrea</option>
<option value="EE">Estonia</option>
<option value="ET">Ethiopia</option>
<option value="FK">Falkland Islands (Malvinas)</option>
<option value="FO">Faroe Islands</option>
<option value="FJ">Fiji</option>
<option value="FI">Finland</option>
<option value="FR">France</option>
<option value="GF">French Guiana</option>
<option value="PF">French Polynesia</option>
<option value="TF">French Southern Territories</option>
<option value="GA">Gabon</option>
<option value="GM">Gambia</option>
<option value="GE">Georgia</option>
<option value="DE">Germany</option>
<option value="GH">Ghana</option>
<option value="GI">Gibraltar</option>
<option value="GR">Greece</option>
<option value="GL">Greenland</option>
<option value="GD">Grenada</option>
<option value="GP">Guadeloupe</option>
<option value="GU">Guam</option>
<option value="GT">Guatemala</option>
<option value="GN">Guinea</option>
<option value="GW">Guinea-Bissau</option>
<option value="GY">Guyana</option>
<option value="HT">Haiti</option>
<option value="HM">Heard Island and McDonald Islands</option>
<option value="VA">Holy See (Vatican City State)</option>
<option value="HN">Honduras</option>
<option value="HK">Hong Kong</option>
<option value="HU">Hungary</option>
```

```
            <option value="IS">Iceland</option>
            <option value="IN">India</option>
            <option value="ID">Indonesia</option>
            <option value="IR">Iran, Islamic Republic of</option>
            <option value="IQ">Iraq</option>
            <option value="IE">Ireland</option>
            <option value="IL">Israel</option>
            <option value="IT">Italy</option>
            <option value="JM">Jamaica</option>
            <option value="JP">Japan</option>
            <option value="JO">Jordan</option>
            <option value="KZ">Kazakhstan</option>
            <option value="KE">Kenya</option>
            <option value="KI">Kiribati</option>
            <option value="KP">Korea, Democratic People's Republic of</option>
            <option value="KR">Korea, Republic of</option>
            </select>




            <label for="isic">ISIC A-B</label>
            <input type="text" id="isic" name="isic_ab" placeholder="Enter ISIC A-B" required>

            <label for="population">Population</label>
            <input type="text" id="population" name="population" placeholder="In billion" required>

            <button type="submit">Submit</button>
        </form>
    </div>
</body>
</html>
```

APP.PY

```
 from flask import Flask, request, render _template
import pickle
import numpy as np
import pandas as pd


app = Flask(__name__)
```

```
model=pickle. load(open("model1.pkl", “rb”))
#model1 = pickle. load(open("robust. pkl", ‘rb’))


@app.route('/')
def home():
   return render_ template("index.html")

@app.route('/predict')
def innerpage():
   return render_ template("inner-page.html")

@app.route('/process',methods=["POST","GET"]) # Ensure only POST requests
are accepted
def submit():

      # Reading the inputs given by the user from the form
      AMA_ exchange_ rate = float(request. form["AMA_ exchange_ rate"])
      Year = float(request. form["year"])
      Change_ in_ inventories = float(request. form["Change _in_ inventories"])
      Country = int(request. form["Country"])
      ISIC_AB = float(request. form["(ISIC A-B)"])
      Population = float(request. form["Population"])

      # Transform the input features as required by your model
      X = [[AMA_ exchange_ rate, Year, Change_ in_ inventories, Country, ISIC_AB,
Population]]

      # Make predictions using the loaded model
      prediction = model. predict(X)
      result="The IMF-Based Exchange Rate is "+str(round(prediction[0],4))
      # Provide the prediction to the template
      return render_ template("portfolio-details.html", predict=result)
```

```
if __name__ == '__main__':
    app. run(debug=True, port=4000)
```

CODE SNIPPETS

MODEL BUILDING

Importing the libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.linear_model import Lasso
import xgboost as xgb
```

Loading the dataset

```
data=pd.read_csv("Global Economy Indicators.csv")
```

```
data
```

	CountryID	Country	Year	AMA exchange rate	IMF based exchange rate	Population	Currency	Per capita GNI	Agriculture, hunting, forestry, fishing (ISIC A-B)	Changes in inventories	...	Household consumption expenditure (including Non-profit institutions serving households)	Imports of goods and services
0	4	Afghanistan	1970	0.044998	0.044998	10752971	Afghani	164	8.699174e+08	NaN	...	1.551094e+09	1.952772e+08
1	4	Afghanistan	1971	0.044998	0.044998	11015857	Afghani	168	9.108281e+08	NaN	...	1.675426e+09	2.762965e+08
2	4	Afghanistan	1972	0.044998	0.044998	11286753	Afghani	149	8.279453e+08	NaN	...	1.498812e+09	2.903704e+08
3	4	Afghanistan	1973	0.044998	0.044998	11575305	Afghani	150	8.554869e+08	NaN	...	1.508024e+09	2.629629e+08

```
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
```

```python
data.drop(' CountryID ',axis=1, inplace=True)
```

```python
data=data.rename(columns={'AMA exchange rate':'AMA_exchange_rate',
                          "IMF based exchange rate":"IMF_exchange_rate",
                          'Agriculture, hunting, forestry, fishing (ISIC A-B)':'(ISIC A-B)',
                          'Construction (ISIC F)':'(ISIC F)',
                          'Changes in inventories':'Changes_in_inventories',
                           'Exports of goods and services':"Exports",
                           'Final consumption expenditure':'Final_expenditure',
                           'General government final consumption expenditure':"Govt_expenditure",
                          'Gross capital formation':'GCF',
                          'Gross fixed capital formation (including Acquisitions less disposals of valuables)':"GFCF",
                           'Household consumption expenditure (including Non-profit institutions serving households)':"HCE",
                           'Imports of goods and services':"Imports", 'Manufacturing (ISIC D)':" (ISIC D)",
                          'Mining, Manufacturing, Utilities (ISIC C-E)':"(ISIC C-E)",
      'Other Activities (ISIC J-P)':"(ISIC J-P)",
                          'Total Value Added': 'Total_Value_Added',
      'Transport, storage and communication (ISIC I)':" (ISIC I)",
      'Wholesale, retail trade, restaurants and hotels (ISIC G-H)':"(ISIC G-H)",
                          'Gross National Income(GNI) in USD':"GNI",
                          'Gross Domestic Product (GDP)':"GDP"}).copy()
```

In [14]:
```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10512 entries, 0 to 10511
Data columns (total 25 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Country                10512 non-null  object
 1   Year                   10512 non-null  int64
 2   AMA_exchange_rate      10512 non-null  float64
 3   IMF_exchange_rate      10512 non-null  float64
 4   Population             10512 non-null  int64
 5   Currency               10512 non-null  object
 6   Per capita GNI         10512 non-null  int64
 7   (ISIC A-B)             10391 non-null  float64
 8   Changes_in_inventories 8671 non-null   float64
 9   (ISIC F)               10512 non-null  float64
 10  Exports                10491 non-null  float64
 11  Final_expenditure      10512 non-null  float64
 12  Govt_expenditure       10460 non-null  float64
 13  GCF                    10460 non-null  float64
 14  GFCF                   10460 non-null  float64
 15  HCE                    10460 non-null  float64
 16  Imports                10470 non-null  float64
 17  (ISIC D)               10469 non-null  float64
 18  (ISIC C-E)             10512 non-null  float64
 19  (ISIC J-P)             10512 non-null  float64
```


```
 18  (ISIC C-E)             10512 non-null  float64
 19  (ISIC J-P)             10512 non-null  float64
```


```
 20  Total_Value_Added      10512 non-null  float64
 21  (ISIC I)               10463 non-null  float64
 22  (ISIC G-H)             10463 non-null  float64
 23  GNI                    10512 non-null  float64
 24  GDP                    10512 non-null  float64
dtypes: float64(20), int64(3), object(2)
memory usage: 2.0+ MB
```

```
5]: data.isnull().sum()
```

```
5]: Country                    0
    Year                       0
    AMA_exchange_rate          0
    IMF_exchange_rate          0
    Population                 0
    Currency                   0
    Per capita GNI             0
    (ISIC A-B)               121
    Changes_in_inventories  1841
    (ISIC F)                   0
    Exports                   21
    Final_expenditure          0
    Govt_expenditure          52
    GCF                       52
    GFCF                      52
    HCE                       52
    Imports                   42
    (ISIC D)                  43
    (ISIC C-E)                 0
    (ISIC J-P)                 0
    Total_Value_Added          0
    (ISIC I)                  49
    (ISIC G-H)                49
    GNI                        0
    GDP                        0
    dtype: int64
```

```
data['(ISIC A-B)']=data['(ISIC A-B)'].fillna(data['(ISIC A-B)'].mean())
data['Changes_in_inventories']=data['Changes_in_inventories'].fillna(data['Changes_in_inventories'].mean())
data['Exports']=data['Exports'].fillna(data['Exports'].mean())
data['Govt_expenditure']=data['Govt_expenditure'].fillna(data['Govt_expenditure'].mean())
data['GCF']=data['GCF'].fillna(data['GCF'].mean())
data['GFCF']=data['GFCF'].fillna(data['GFCF'].mean())
data['HCE']=data['HCE'].fillna(data['HCE'].mean())
data[ 'Imports']=data[ 'Imports'].fillna(data[  'Imports'].mean())
data['(ISIC D)']=data['(ISIC D)'].fillna(data[ '(ISIC D)'].mean())
data[ '(ISIC I)']=data['(ISIC I)'].fillna(data[  '(ISIC I)'].mean())
data[  '(ISIC G-H)']=data[   '(ISIC G-H)'].fillna(data[    '(ISIC G-H)'].mean())
```

```
data.isnull().sum()
```

```
Country                    0
Year                       0
AMA_exchange_rate          0
IMF_exchange_rate          0
Population                 0
Currency                   0
Per capita GNI             0
(ISIC A-B)                 0
Changes_in_inventories     0
(ISIC F)                   0
Exports                    0
Final_expenditure          0
Govt_expenditure           0
GCF                        0
GFCF                       0
HCE                        0
Imports                    0
(ISIC D)                   0
(ISIC C-E)                 0
(ISIC J-P)                 0
Total_Value_Added          0
(ISIC I)                   0
(ISIC G-H)                 0
GNI                        0
GDP                        0
dtype: int64
```

Handling Categorical columns

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()

data['Country'] = label_encoder.fit_transform(data['Country'])
data['Currency'] = label_encoder.fit_transform(data['Currency'])
```

```
df.duplicated().sum()
```

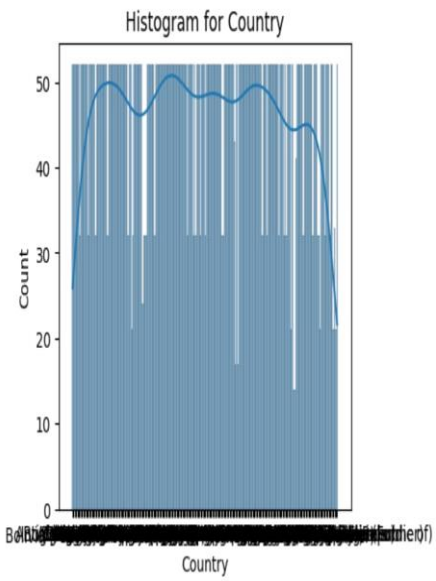
```
0
```

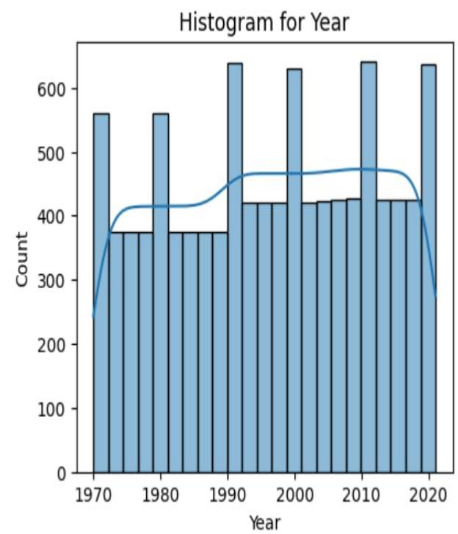
```
data.describe()
```

	Country	Year	AMA_exchange_rate	IMF_exchange_rate	Population	Currency	Per capita GNI	(ISIC A-B)	Changes_in_inventories
count	10512.000000	10512.000000	1.051200e+04	1.051200e+04	1.051200e+04	10512.000000	10512.000000	1.051200e+04	1.051200e+04
mean	107.378139	1996.262747	3.305837e+01	3.096314e+01	1.159665e+07	71.765221	8965.564593	2.821832e+09	7.808287e+08
std	62.698262	14.900361	4.979378e+01	4.670960e+01	1.421149e+07	44.747553	17070.205895	3.618141e+09	1.307993e+09
min	0.000000	1970.000000	4.300000e-14	4.300000e-14	4.359000e+03	0.000000	34.000000	2.813900e+04	-2.430032e+09
25%	54.000000	1984.000000	1.000000e+00	1.000000e+00	6.330615e+05	36.750000	730.000000	1.336557e+08	3.913107e+06
50%	108.000000	1997.000000	2.812895e+00	2.761315e+00	5.051556e+06	60.000000	2316.500000	9.569466e+08	1.890237e+08
75%	160.000000	2009.000000	5.134316e+01	4.806684e+01	1.678862e+07	110.000000	8965.750000	4.213059e+09	1.626543e+09
max	219.000000	2021.000000	1.268579e+02	1.186671e+02	4.102195e+07	152.000000	234317.000000	1.033216e+10	4.060488e+09

8 rows × 25 columns

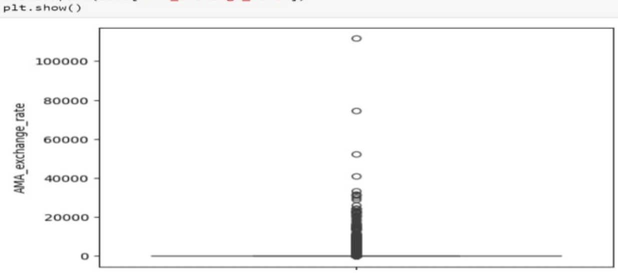
```
first_three_columns = data.iloc[:, :2]

for column in first_three_columns.columns:
    plt.figure(figsize=(15, 4))
    plt.subplot(1, 3, 1)
    sns.histplot(first_three_columns[column], kde=True)
    plt.title(f"Histogram for {column}")

plt.show()
```






```
for i in data.columns:
    if data[i].dtype == 'int64' or data[i].dtype == 'float64':
        sns.boxplot(y=data[i], orient='vertical')
        plt.title(f'Boxplot for {i}')
        plt.show()
```

```
sns.boxplot(data['AMA_exchange_rate'])
plt.show()
```

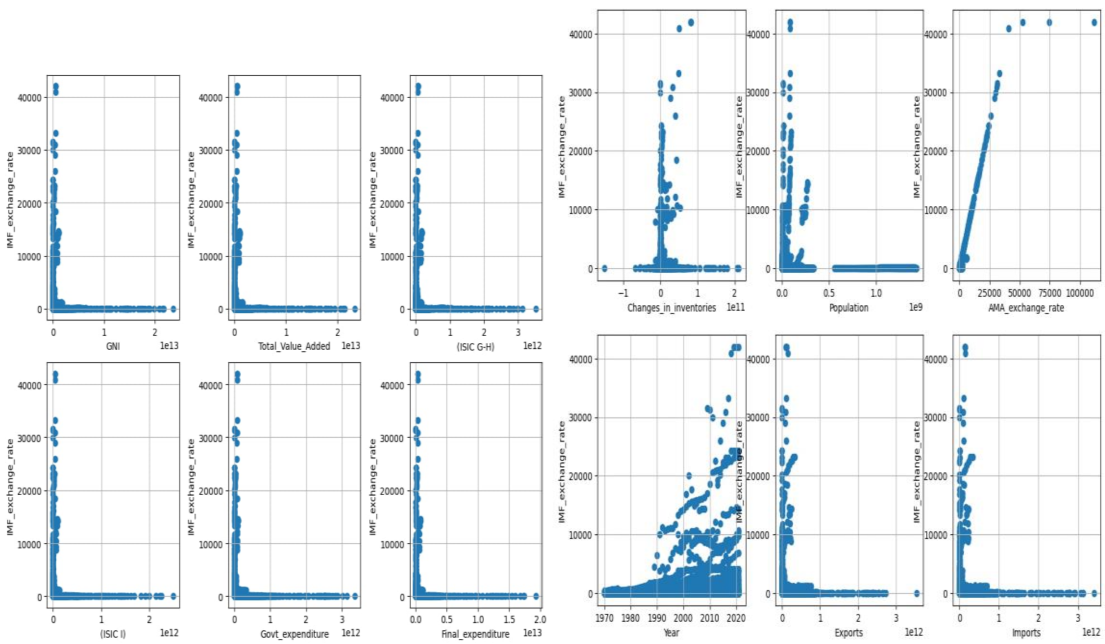


```python
quartile = data['AMA_exchange_rate'].quantile(q=[0.25,0.75])
quartile
```

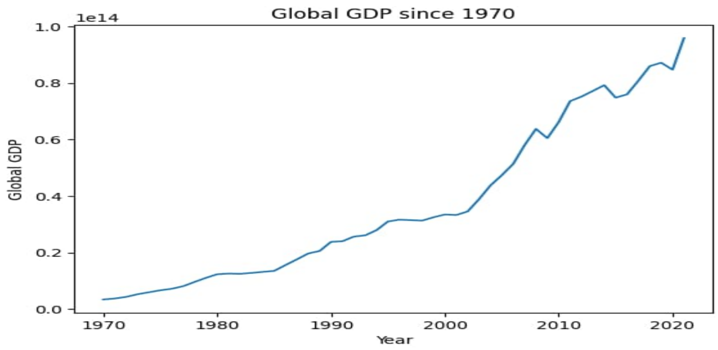
```
0.25     1.000000
0.75    51.343163
Name: AMA_exchange_rate, dtype: float64
```

```python
import numpy as np
quartile.iloc[1]
q1 = quartile.iloc[0]
q3 = quartile.iloc[1]

iqr = q3 - q1
iqr

maxwisker = q3+1.5*(iqr)
print(maxwisker)

minwisker = q1-1.5*(iqr)
print(minwisker)

data['AMA_exchange_rate'] = np.where(data['AMA_exchange_rate']>126.85790676875,126.85790676875,data['AMA_exchange_rate'])
```

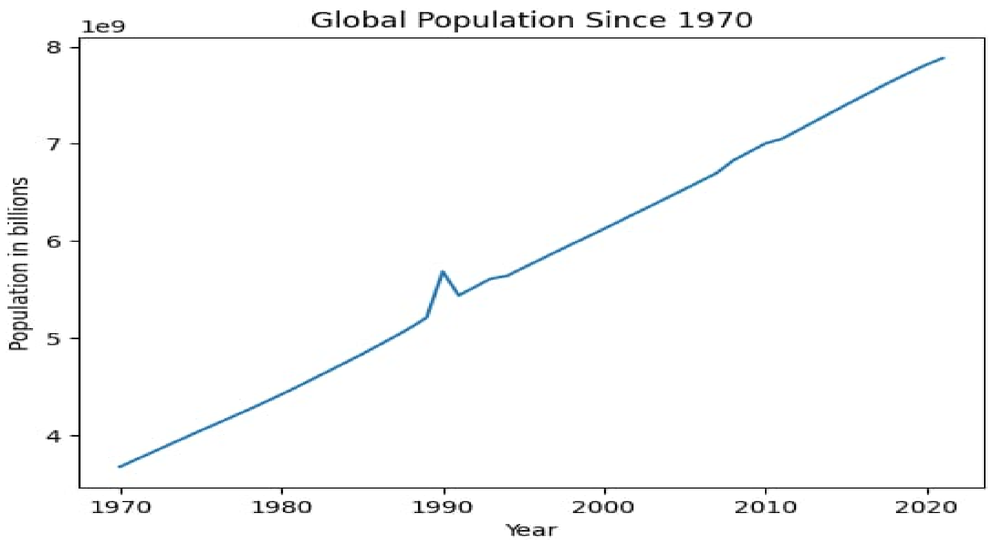
```
126.85790676875
-74.51474406125
```




```python
df = data.groupby(['Year']).sum()
df_gdp = df['GDP']
df_gdp.plot();
plt.ylabel('Global GDP')
plt.xlabel('Year')
plt.title('Global GDP since 1970')
```



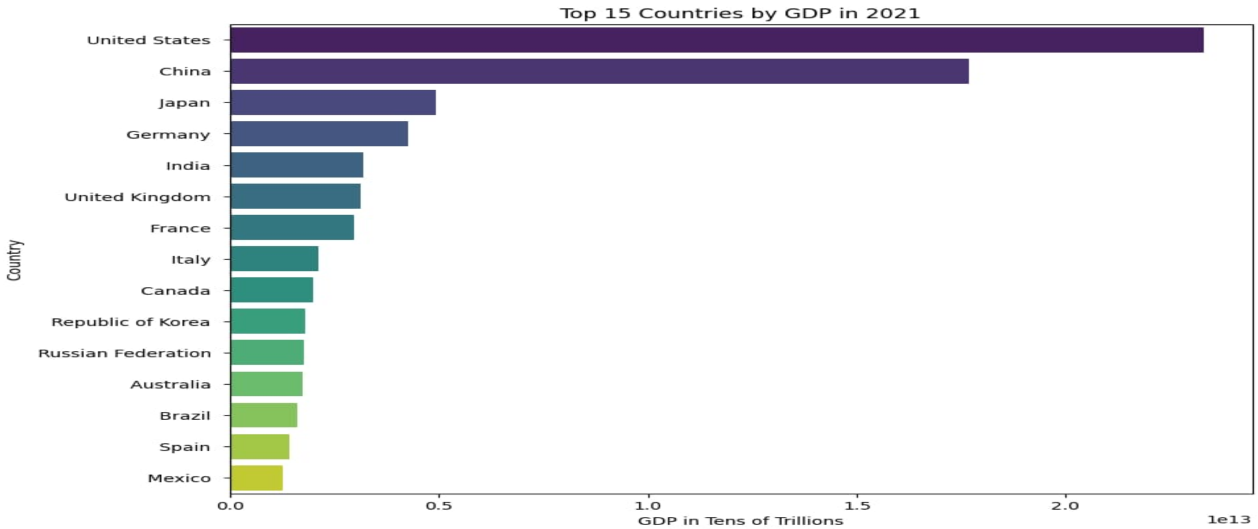

```
df1 = data[['Country','Year','Population']]
df1.groupby(['Year']).Population.sum().sort_index().plot()
plt.ylabel('Population in billions')
plt.xlabel('Year')
plt.title('Global Population Since 1970');
```




```
df_2021 = data[data['Year'] == 2021]
top_15_countries = df_2021.nlargest(15, 'GDP')
```

```
plt.figure(figsize=(10, 8))
sns.barplot(x='GDP', y='Country', data=top_15_countries, palette='viridis')
plt.title('Top 15 Countries by GDP in 2021')
plt.xlabel('GDP in Tens of Trillions')
plt.ylabel('Country')

plt.show()
```

MULTIVARIATE ANALYSIS



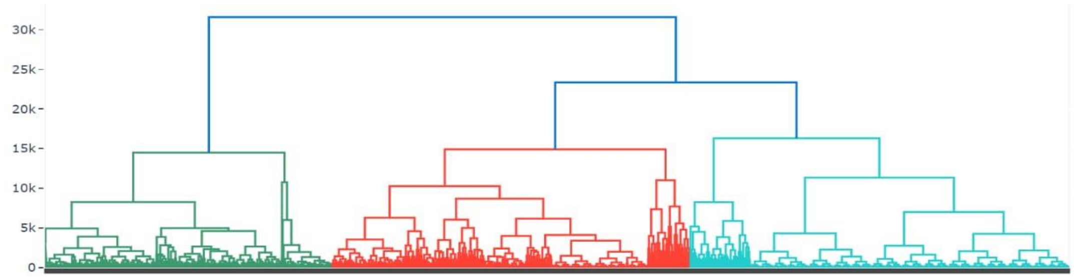

```
data.corr()['IMF_exchange_rate'].sort_values(ascending=False)
```

```
IMF_exchange_rate         1.000000
AMA_exchange_rate         0.948164
Year                      0.145939
Changes_in_inventories    0.131074
(ISIC A-B)                0.047207
Population                0.036559
(ISIC C-E)                0.006260
GCF                       0.003999
Imports                   0.003666
Exports                   0.003662
(ISIC F)                  0.002432
GFCF                     -0.000032
(ISIC D)                 -0.000565
Total_Value_Added        -0.003638
GDP                      -0.003951
(ISIC G-H)               -0.004356
(ISIC I)                 -0.004447
GNI                      -0.004487
HCE                      -0.005180
Final_expenditure        -0.006682
(ISIC J-P)               -0.012210
Govt_expenditure         -0.012832
Per capita GNI           -0.067830
Name: IMF_exchange_rate, dtype: float64
```

```
from scipy.cluster.hierarchy import linkage, fcluster
from sklearn.metrics import silhouette_score

Z1 = linkage(df, method='average', metric='cityblock')
clusters = fcluster(Z1, t=2, criterion='maxclust')
silhouette_avg = silhouette_score(df, clusters)
print(f'Silhouette Score for average linkage and cityblock distance metric: {silhouette_avg}')
```

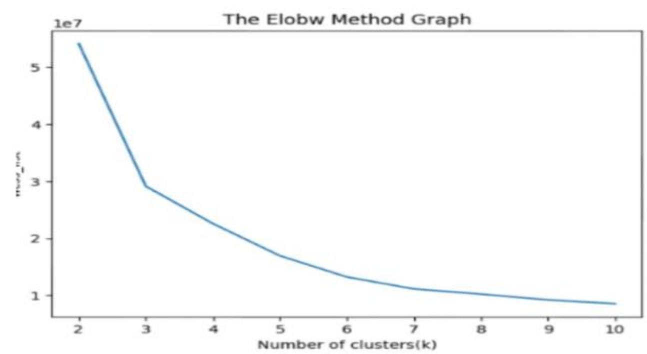
```
Silhouette Score for average linkage and cityblock distance metric: 0.9447199329887062
```

```
import plotly.figure_factory as ff
fig = ff.create_dendrogram(Z1)
fig.update_layout(autosize=True, hovermode='closest')
fig.update_xaxes(mirror=False, showgrid=True, showline=True, showticklabels=False)
fig.update_yaxes(mirror=False, showgrid=True, showline=True)
fig.show()
```



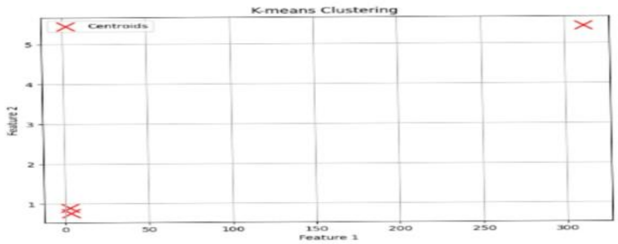
K MEANS CLUSTERING

```
from sklearn.cluster import KMeans
wcss_list= []

for i in range(2, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42)
    kmeans.fit(df)
    wcss_list.append(kmeans.inertia_)
plt.plot(range(2, 11), wcss_list)
plt.title('The Elobw Method Graph')
plt.xlabel('Number of clusters(k)')
plt.ylabel('wcss_list')
plt.show()
```




```
kmeans = KMeans(n_clusters=3)

kmeans.fit(df)
centroids = kmeans.cluster_centers_
labels = kmeans.labels_

plt.figure(figsize=(8, 6))
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=200, marker='x', label='Centroids')

plt.title('K-means Clustering')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()
plt.grid(True)
plt.show()
```

FEATURE SELECTION



Feature Selection

```
features=['AMA_exchange_rate','Year','Changes_in_inventories','Country','(ISIC A-B)','Population','Currency']
X=data[features]
```

```
X=df.drop(["IMF_exchange_rate"],axis=1)
y=data["IMF_exchange_rate"]
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
model1 = LinearRegression()
model1.fit(X_train, y_train)
```

```
LinearRegression()
```

```
y_pred = model1.predict(X_test)
```

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

print("Mean Squared Error:", mse)
print("R-squared Score:", r2)
```

```
Mean Squared Error: 1912.4072205848418
R-squared Score: 0.10502662158394682
```

```
y_pred=model2.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared Score:", r2)
```

```
Mean Squared Error: 1912.4002330743629
R-squared Score: 0.1050298916175425
```

RANDOM FOREST REGRESSOR

```
model3=SVR()
model3.fit(X_train,y_train)
```

```
SVR()
```

```
y_pred=model3.predict(X_test)
mse=mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test,y_pred)
print("Mean Squared Error:", mse)
print("R-squared Score:", r2)
```

```
Mean Squared Error: 2842.6338000438527
R-squared Score: -0.3303032681747007
```

```
model4=RandomForestRegressor(n_estimators=100, random_state=42)
model4.fit(X_train,y_train)
```

```
RandomForestRegressor(random_state=42)
```

```
y_pred=model4.predict(X_test)
mse=mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test,y_pred)
print("Mean Squared Error:", mse)
print("R-squared Score:", r2)
```

```
Mean Squared Error: 37.57894940284747
R-squared Score: 0.9824137040780953
```

```
model5=Lasso(alpha=1.0)
model5.fit(X_train,y_train)
```

```
Lasso()
```

```
y_pred=model5.predict(X_test)
mse=mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test,y_pred)
print("Mean Squared Error:", mse)
print("R-squared Score:", r2)
```

```
Mean Squared Error: 1908.9877913683163
R-squared Score: 0.10662685509343561
```

COMAPRISON OF MODELS

DecisionTree Classification_Report

```
print(classification_report(dt_pred , y_test))
```

```
              precision    recall  f1-score   support

         0.0       0.81      0.80      0.80       404
         1.0       0.80      0.80      0.80       396

    accuracy                           0.80       800
   macro avg       0.80      0.80      0.80       800
weighted avg       0.80      0.80      0.80       800
```

RandomForest Classification_Report

```
from sklearn.metrics import classification_report
print(classification_report(forest, y_test))
```

```
              precision    recall  f1-score   support

         0.0       0.91      0.91      0.91       399
         1.0       0.91      0.91      0.91       401

    accuracy                           0.91       800
   macro avg       0.91      0.91      0.91       800
weighted avg       0.91      0.91      0.91       800
```

XGBoost Classification_Report

```
print(classification_report(y_pred, y_test))
```

```
              precision    recall  f1-score   support

           0       0.91      0.92      0.91       395
           1       0.92      0.91      0.92       405

    accuracy                           0.92       800
   macro avg       0.92      0.92      0.91       800
weighted avg       0.92      0.92      0.92       800
```