**SUMMER TRAINING/INTERNSHIP**

**PROJECT REPORT**

(Term June-July 2025)

## (Population Growth Predictor)

Submitted by

(Sanika Chiddarwar , P Priya)

Registration No – 12316239 , 12314100

Course Code PETV79

Under the Guidance of

**(Mahipal Singh Papola)**

**Lovely School of CSE**

**Lovely Professional University, Phagwara**

APRIL 2025

**Lovely Professional University, Punjab**

**BONAFIDE CERTIFICATE**

Certified that this project report **“POPULATION GROWTH PREDICTOR USING MACHINE LEARNING TECHNIQUES”** is the Bonafide work of **"** **Sanika Chiddarwar and P Priya "** who carried out the project work under my supervision.

**SIGNATURE**

**<<Name of the Supervisor>>**

**Sanika Chiddarwar**

**P Priya**

**Signature**

**<<Signature of the Head of the Department>>  
SIGNATURE  
<<Name>>  
HEAD OF THE DEPARTMENT**

**ACKNOWLEDGEMENT**

I have taken efforts in this assignment. However, it would have not been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to them. I am highly indebted to ‘Lovely Professional University’ for their guidance and constant supervision as well as for providing necessary information regarding the assignment and also their support in completing the assignment. I would like to express my gratitude towards our parents and members of our families for their kind cooperation and encouragement which helped me in completing this assignment. My thanks and appreciation also go to the people who are directly or indirectly helped us out in developing the assignment.

**Table of Contents**

1. Acknowledgement
2. Project Title
3. Chapter 1: Introduction  
   1.1 Company Profile  
   1.2 Overview of Training Domain  
   1.3 Objective of the Project
4. Chapter 2: Training Overview  
   2.1 Tools & Technologies Used
5. Chapter 3: Project Details  
   3.1 Title of the Project  
   3.2 Problem Definition  
   3.3 Scope and Objectives  
   3.4 System Requirements  
   3.5 Architecture Diagram
6. Chapter 4: Implementation  
   4.1 Tools Used  
   4.2 Methodology  
   4.3 Modules / Screenshots  
   4.4 Code Snippets
7. Chapter 5: Results and Discussion  
   5.1 Output / Report  
   5.2 Challenges Faced  
   5.3 Learnings
8. Chapter 6: Conclusion  
   6.1 Summary

**INTRODUCTION**

**1.1 University Profile**

This project was carried out as a requirement for the Artificial Intelligence and Machine Learning program at Lovely Professional University (LPU), one of the top private universities in India. LPU is renowned for its dedication to hands-on learning, industry-focused programs, and academic excellence.   
The university places a strong emphasis on creativity and hands-on experience in a variety of fields, such as artificial intelligence, machine learning, data science, and emerging technologies. With its cutting-edge facilities and robust industry partnerships, LPU offers the perfect setting for students to apply their theoretical knowledge to practical situations.

**1.2 Overview of Training Domain**

This project focusses on machine learning (ML), a fundamental component of artificial intelligence (AI). In machine learning, models are trained on historical data to generate decisions or predictions without explicit programming.   
Regression-based prediction models, which are used to comprehend and forecast continuous outcomes, are the specific focus of this project. The project's goal is to forecast future population trends by using regression algorithms on population data. These methods are essential in fields such as economics, environmental studies, public planning, and healthcare.

**1.3 Objective of the Project**

The primary objectives of this project are as follows:

* To explore and understand historical population data using Exploratory Data Analysis (EDA).
* To apply machine learning regression algorithms—Linear Regression, Random Forest Regressor, and Support Vector Regressor (SVR)—for predicting population growth.
* To compare the performance of these models using evaluation metrics such as Mean Squared Error (MSE) and R² Score.
* To gain practical experience in the end-to-end machine learning pipeline including data preprocessing, model training, and performance evaluation.
* To demonstrate how AI/ML techniques can be effectively applied to solve real-world problems through predictive modeling.

**Training Overview**

**2.1 Tools & Technologies Used**

Several tools and technologies were used to analyse data, create machine learning models, and assess their effectiveness during this project and training in the AI/ML field at Lovely Professional University. Among the essential instruments and technologies are:   
• Python: The main programming language for machine learning and data analysis.   
• Jupyter Notebook: An interactive development environment for project code writing, testing, and documentation.   
Pandas: For preprocessing and data manipulation.   
• NumPy: For managing arrays and numerical calculations.   
• Matplotlib and Seaborn: For exploratory data analysis and data visualisation.   
• Scikit-learn: Used to implement machine learning algorithms such as Random Forest Regressor, Support Vector Regressor (SVR), Linear Regression, and Model Evaluation Metrics (MSE, R2 Score).   
• CSV File Handling: For reading and examining dataset files in the CSV format.

**Project Details**

**3.1 Title of the Project**

Predicting Population Growth Using Machine Learning Techniques

**3.2 Problem Definition**

Urban planning, healthcare, economic development, and resource distribution are all significantly impacted by population growth. Policymakers can make well-informed decisions about employment, education, infrastructure, and environmental sustainability when population trends are accurately predicted.   
This project's goal is to use supervised machine learning techniques to analyse historical population data and create predictive models that can predict future population growth. Finding and assessing suitable models that can successfully identify patterns in the data for precise forecasting is a challenge.

**3.3 Scope and Objectives**

**Scope:**

• The project can only use a historical population dataset in CSV format.   
• It discusses how to predict future values using machine learning regression techniques.   
• Concentrates on developing models, assessing them, and contrasting various algorithms.

**Objectives:**  
The purpose of data exploration and visualisation is to comprehend population trends.   
• To prepare the dataset for ML model training by preprocessing it.   
• To use and contrast different regression algorithms:   
1. Linear Regression   
2. Support Vector Regressor (SVR)

3. Random Forest Regressor   
• To use R2 Score and MSE to assess model performance.   
• To determine which population forecasting model performs the best.

**3.4 System Requirements**

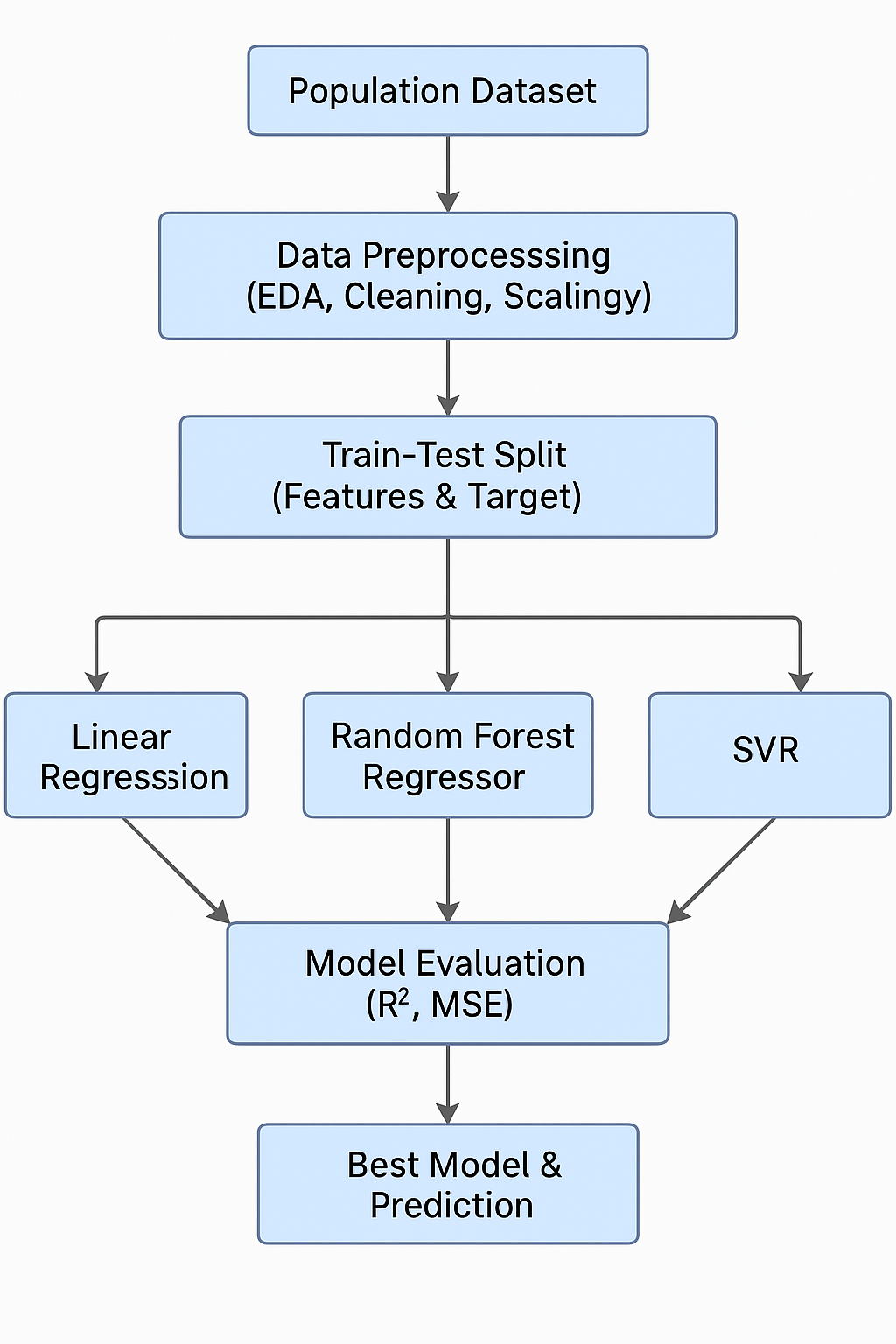
Hardware Requirements:

* Processor: Intel Core i5 or higher
* RAM: Minimum 8 GB
* Storage: At least 500 MB free disk space
* Graphics: Integrated or dedicated (optional for visualizations)

Software Requirements:

* Operating System: Windows/Linux/macOS
* Python 3.8 or above
* Jupyter Notebook (via Anaconda or standalone)
* Python Libraries:
  + NumPy
  + Pandas
  + Matplotlib
  + Seaborn
  + Scikit-learn

**3.5 Architecture Diagram**

****

**Implementation**

**4.1 Tools Used**

The following tools and libraries were used throughout the implementation of the project:

| **Tool / Library** | **Purpose** |
| --- | --- |
| Python 3.9+ | Programming language used for the project |
| Jupyter Notebook | Interactive coding and visualization |
| Pandas | Data manipulation and analysis |
| NumPy | Numerical computing |
| Matplotlib & Seaborn | Data visualization and plotting |
| Scikit-learn | Machine learning models and metrics |
| CSV file | Population dataset in .csv format |

**4.2 Methodology**

The implementation follows a standard machine learning pipeline:

A typical machine learning pipeline is used in the implementation:

Step 1: Data Collection: Using pandas.read\_csv(), load the population dataset (population\_growth\_dataset.csv).

Step 2: Cleaning and Exploration of Data

• To comprehend the structure and content, df.info(), df.describe(), and visualisations were used.

• If there were any missing or unnecessary values, they were cleaned.

Step 3: Preparing the data

• The target variable and features are distinct.

• The feature values were normalised using StandardScaler.

• Use train\_test\_split to divide data into training and test sets.

Step 4: Model Training

Implemented and trained three different models:

* Linear Regression
* Random Forest Regressor
* Support Vector Regressor (SVR)

Step 5: Evaluation

* Evaluated models using:
  + Mean Squared Error (MSE)
  + R² Score
* Compared all three models to determine the most accurate predictor.

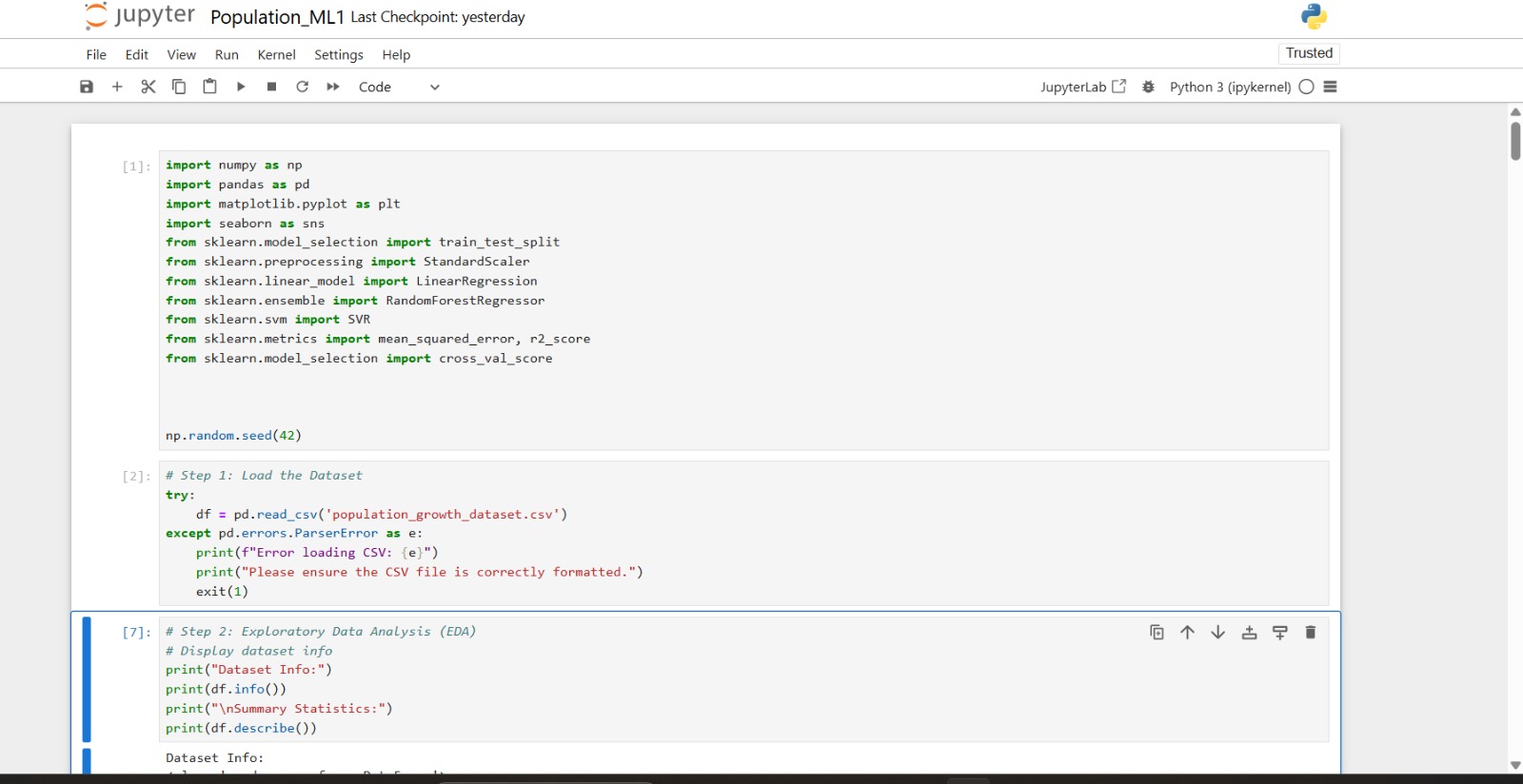
Step 6: Prediction

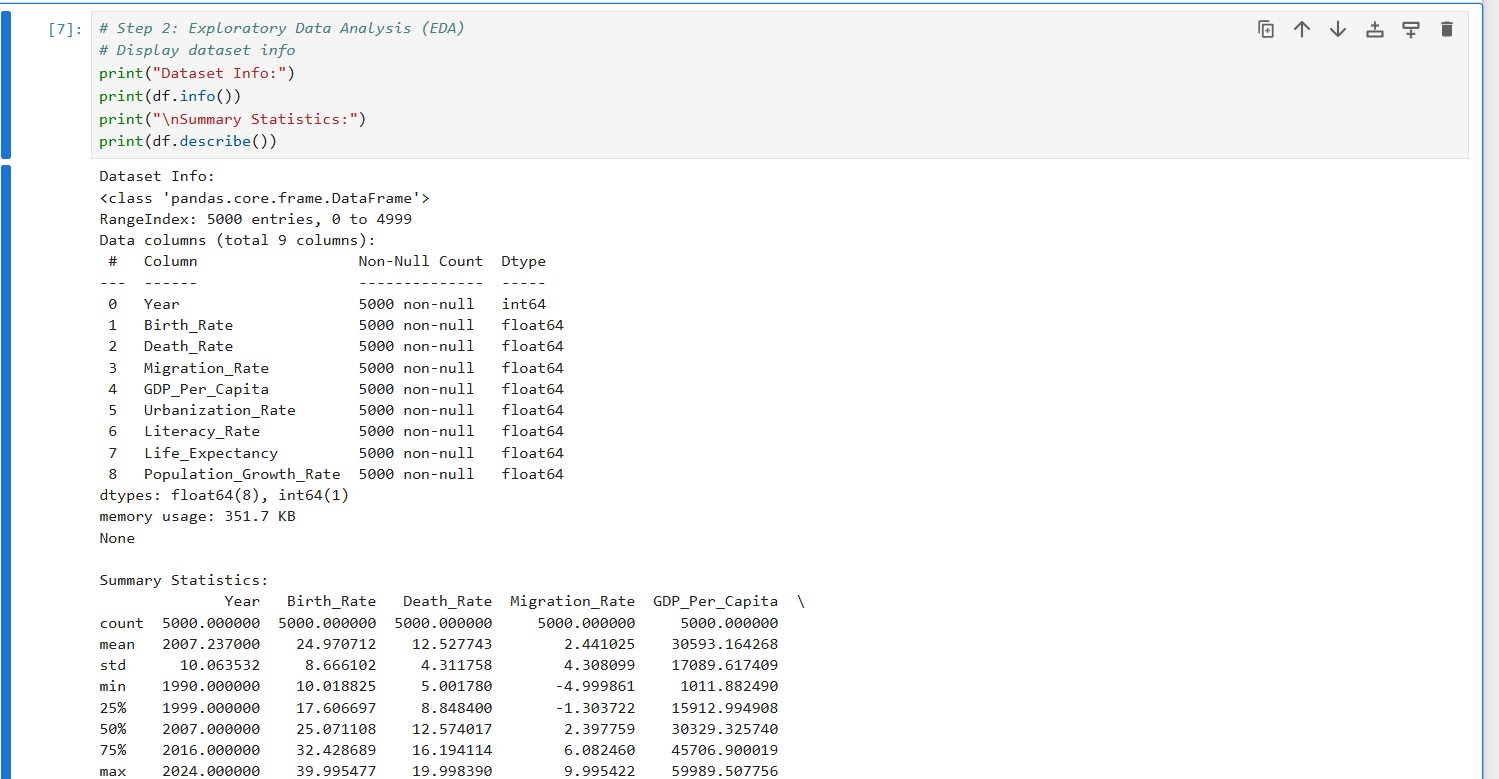
* Used the best-performing model to predict population for unseen test data.

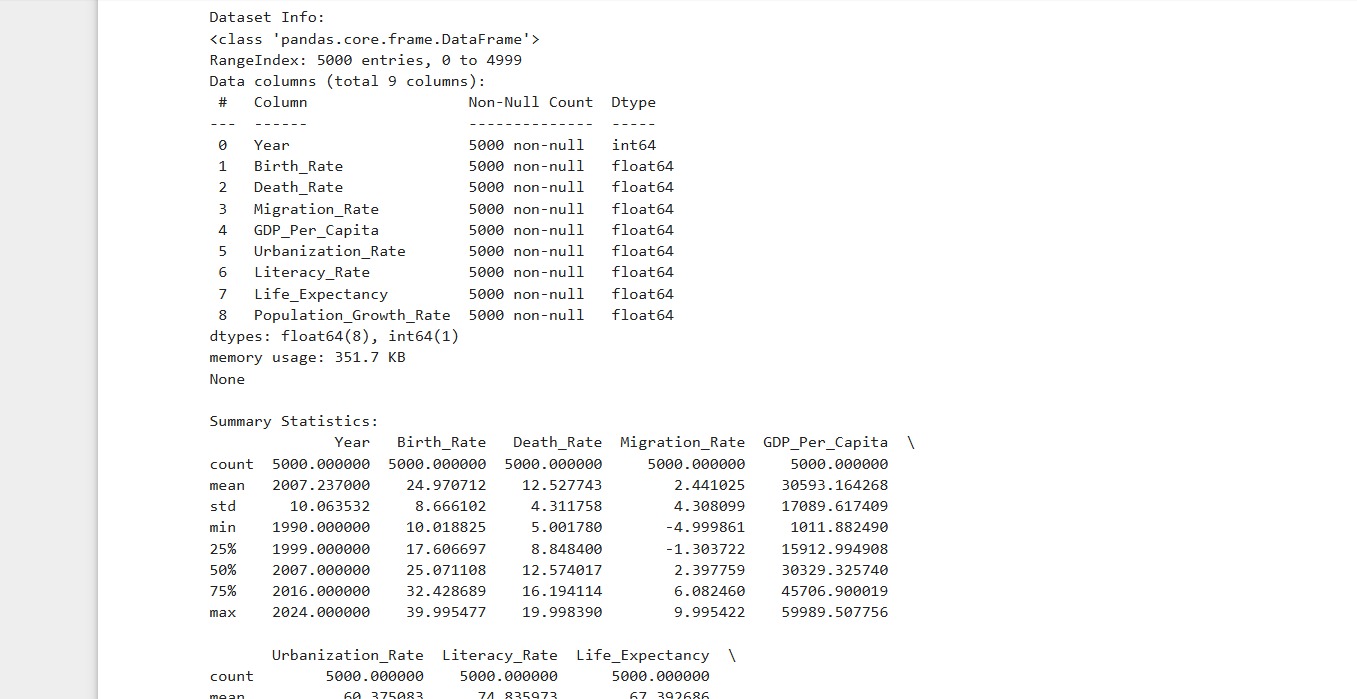
**4.3 Modules / Screenshots**

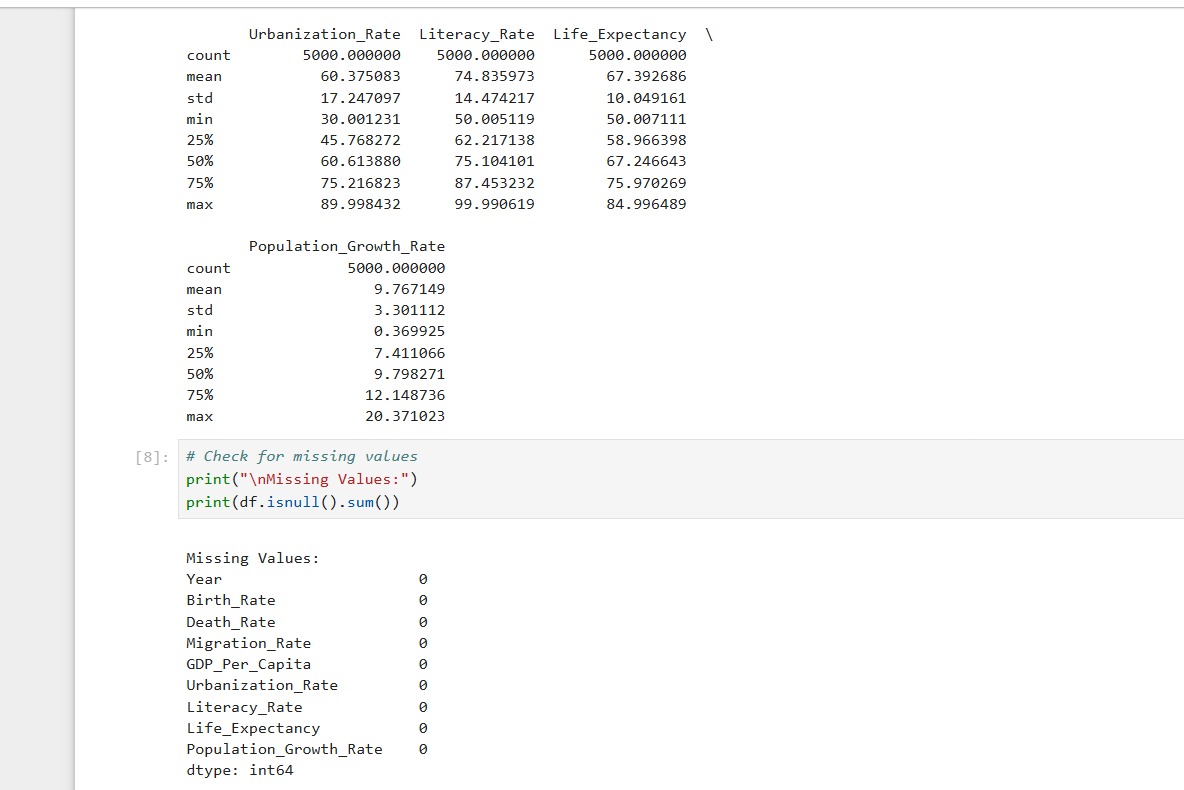
Here’s an overview of major sections (modules) from the implementation:

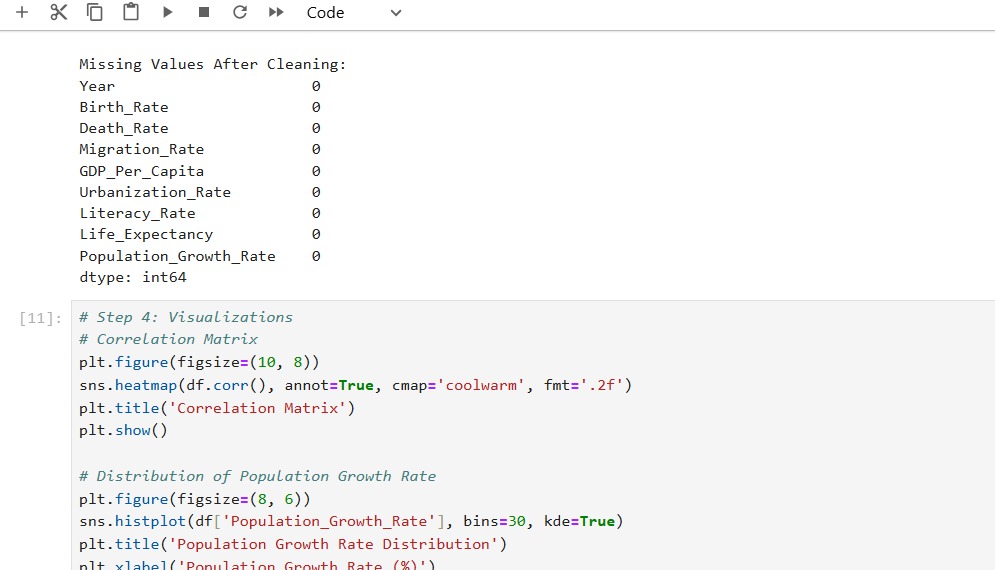
* Data Loading & Preprocessing
* Model Training
* Prediction Output

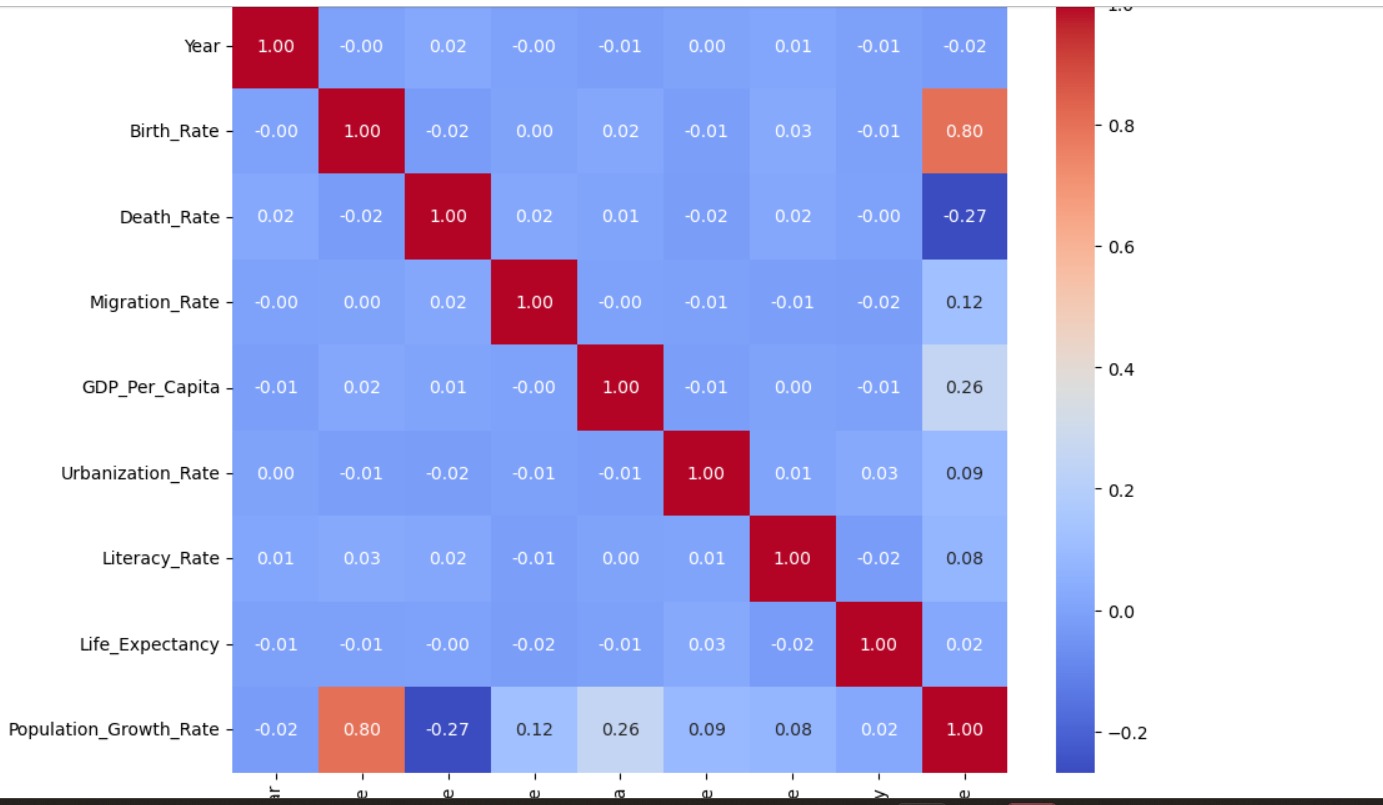


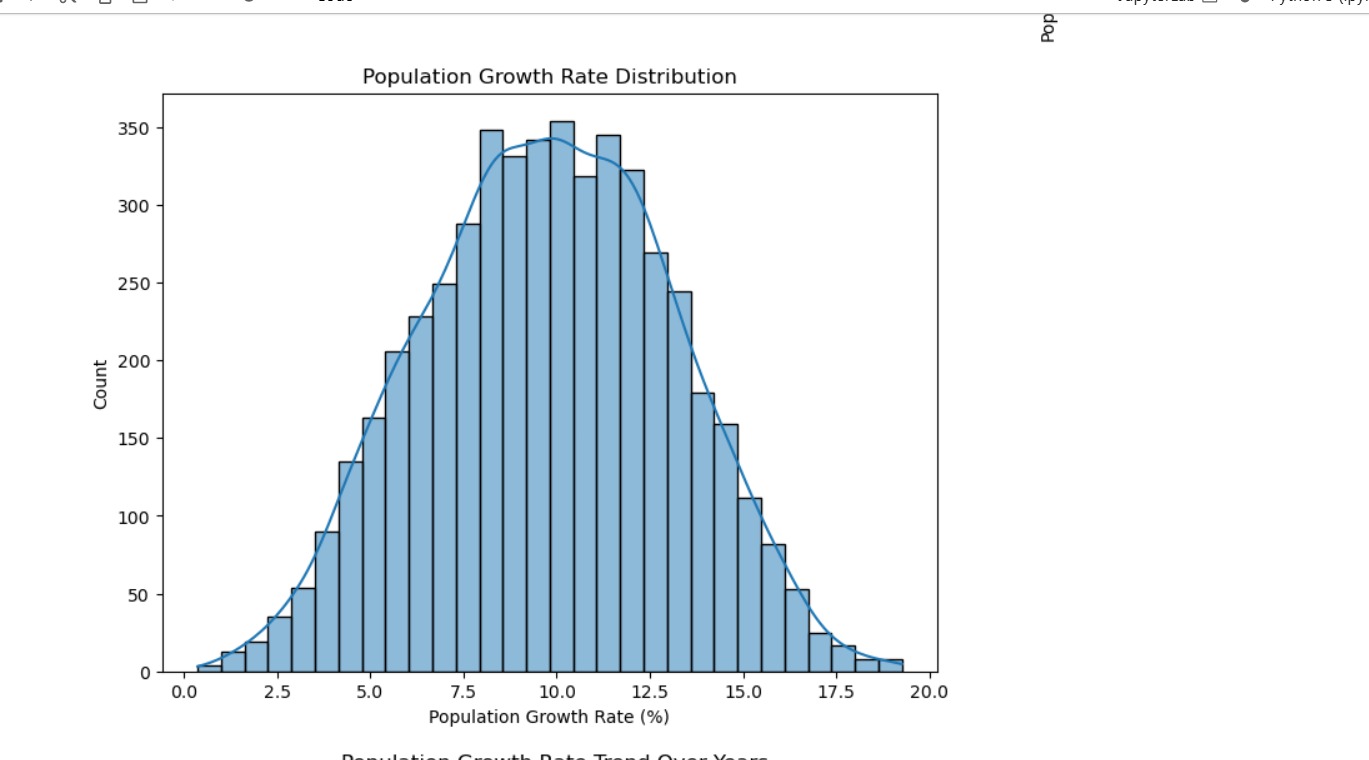


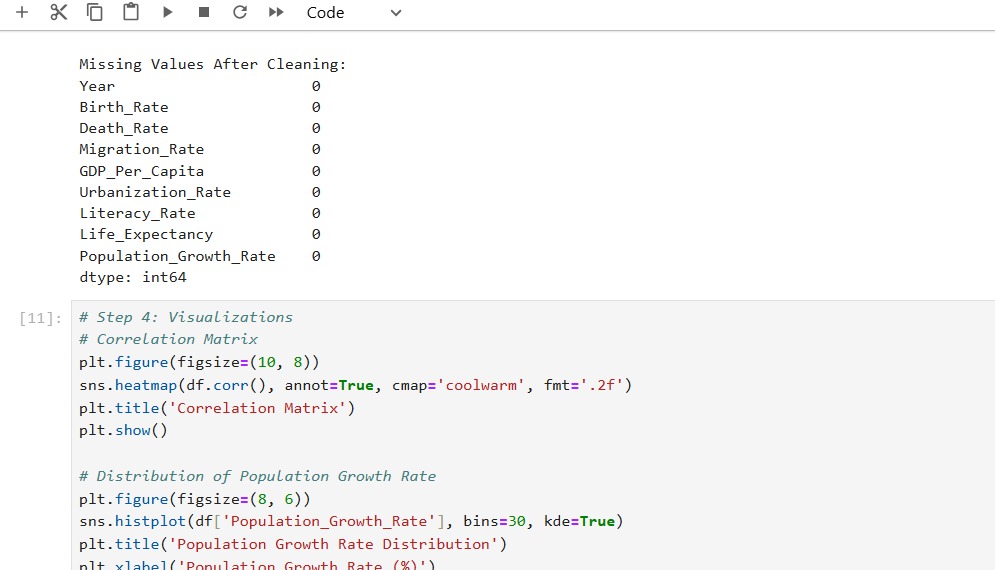


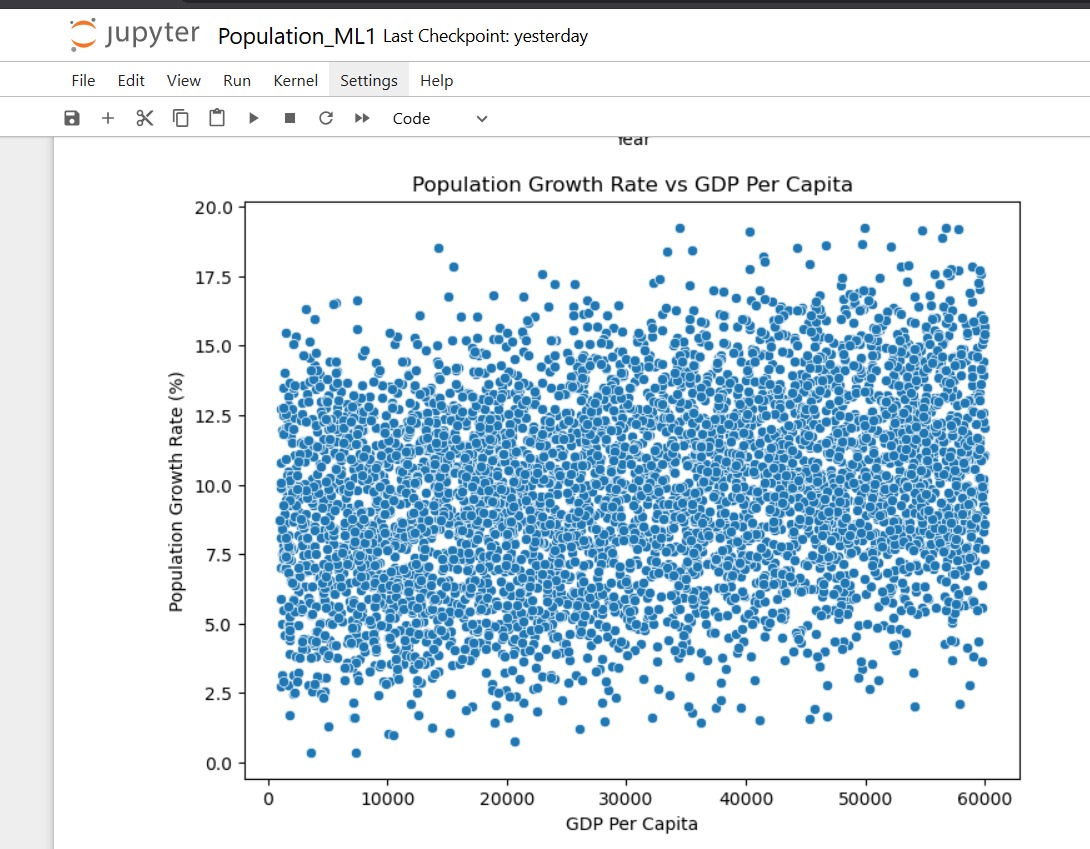
****

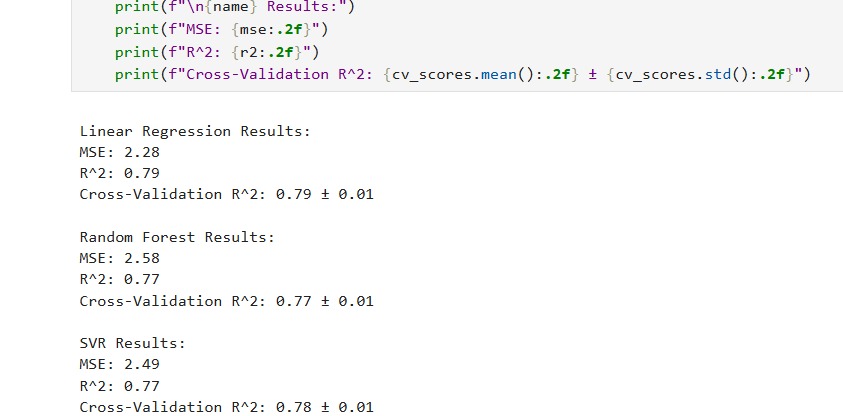
****

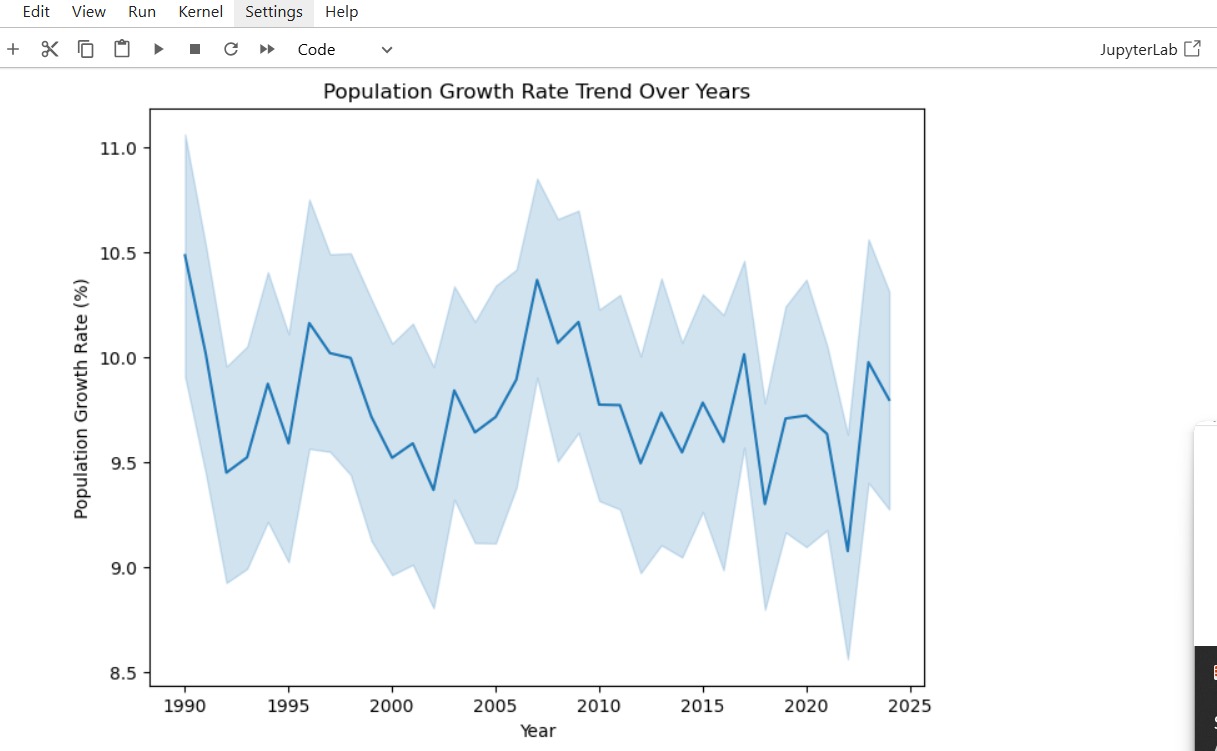
****

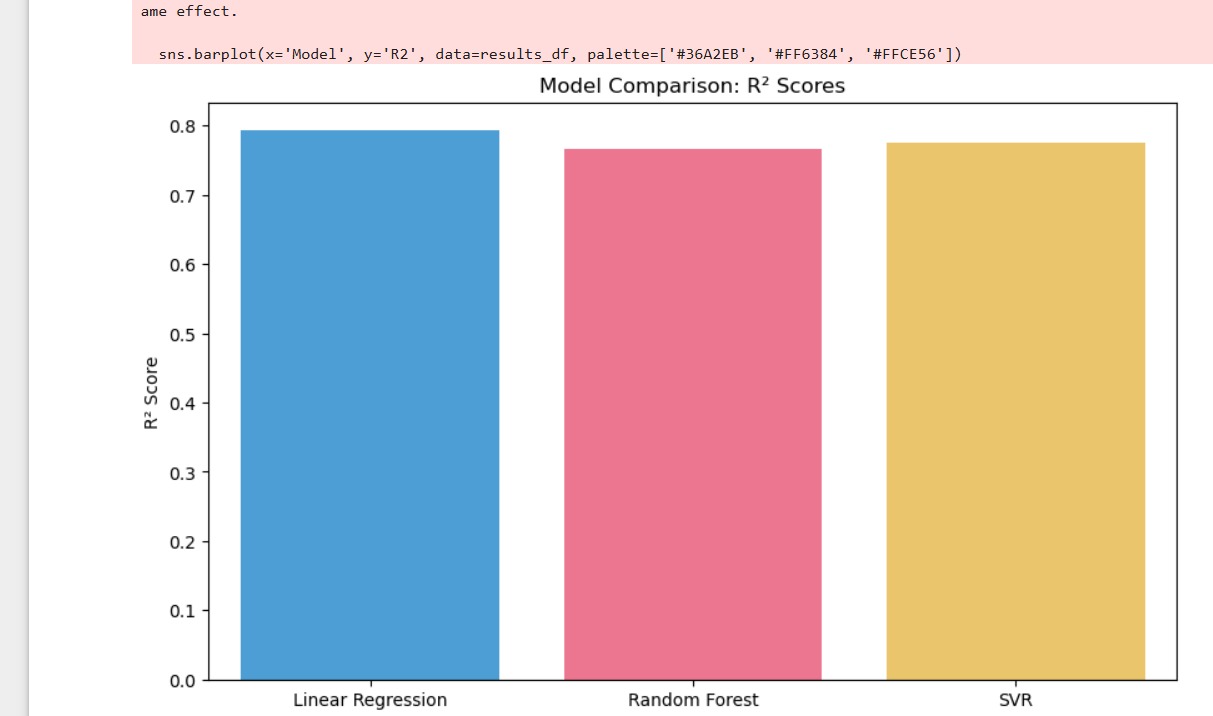
****

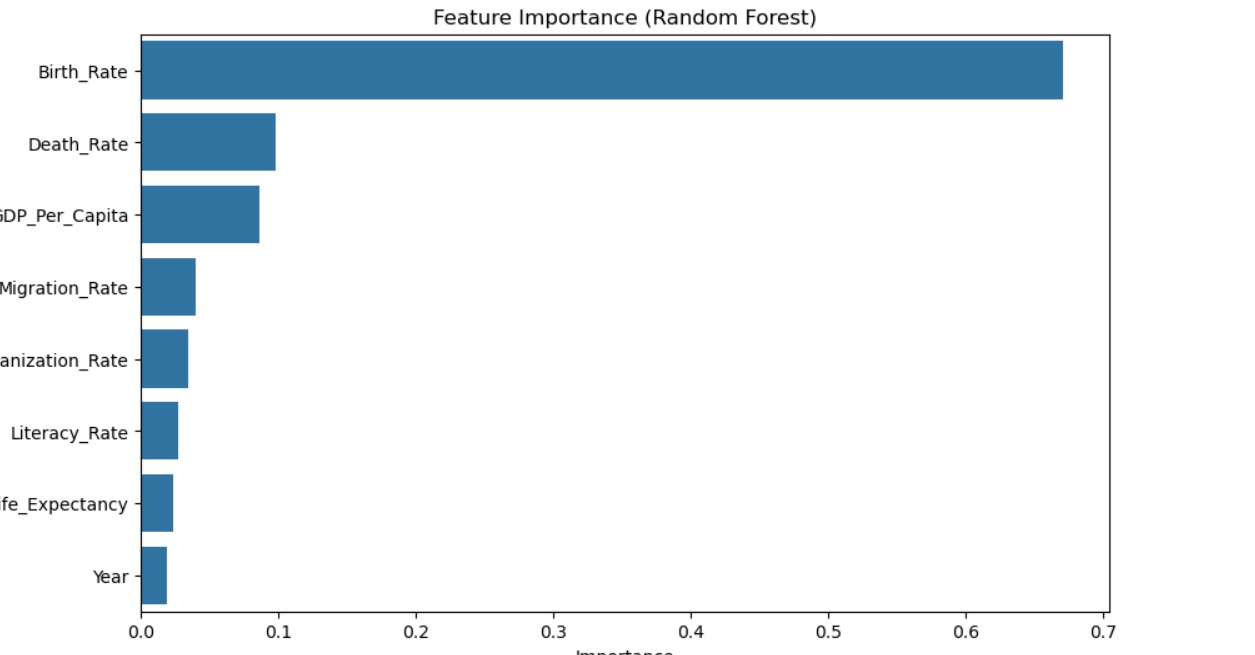
****

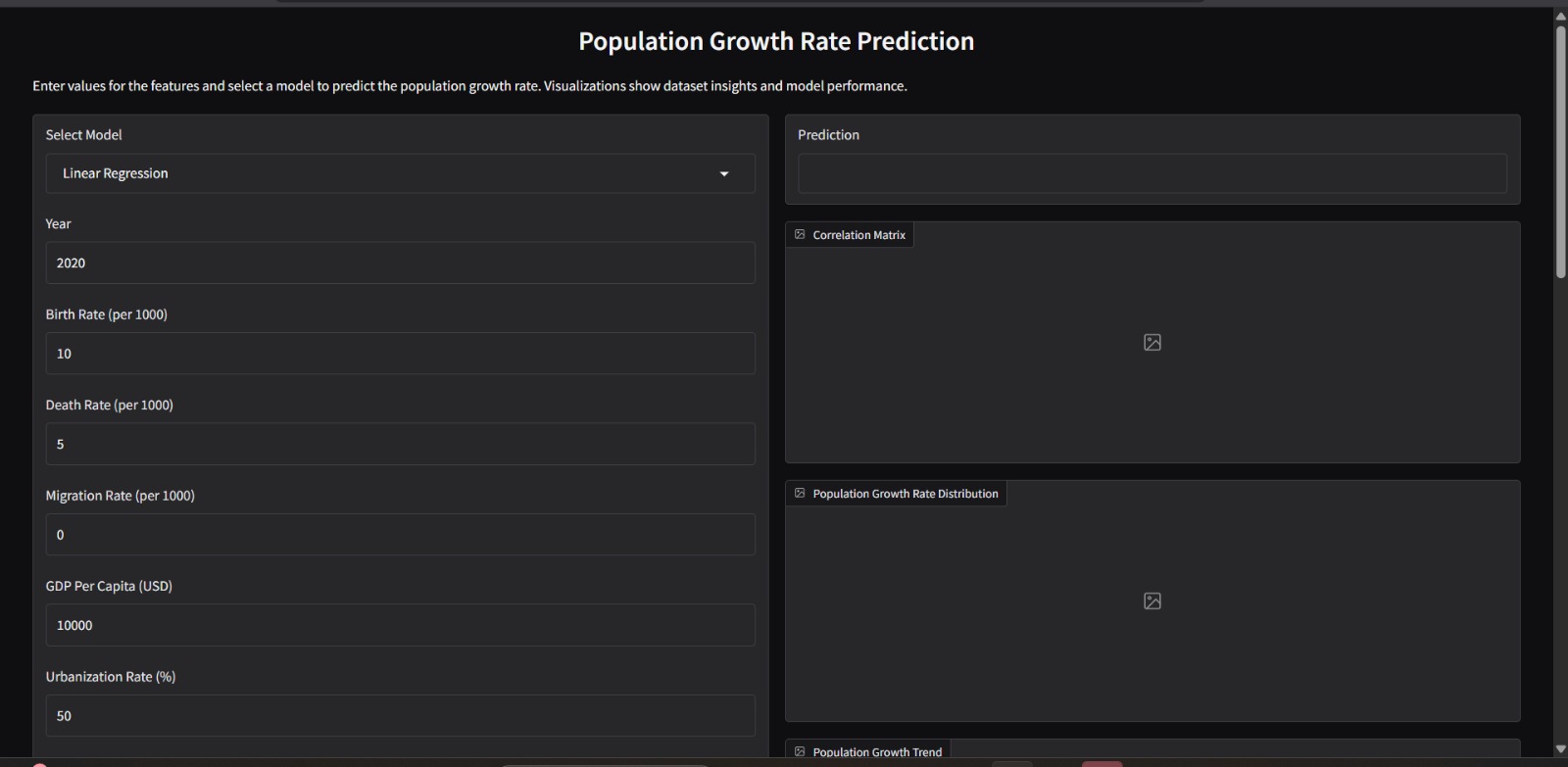
****

****

****

****

****

****

**4.4 Code Snippets**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

from sklearn.svm import SVR

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.model\_selection import cross\_val\_score

np.random.seed(42)

**# Step 1: Load the Dataset**

try:

    df = pd.read\_csv('population\_growth\_dataset.csv')

except pd.errors.ParserError as e:

    print(f"Error loading CSV: {e}")

    print("Please ensure the CSV file is correctly formatted.")

    exit(1)

**# Step 2: Exploratory Data Analysis (EDA)**

# Display dataset info

print("Dataset Info:")

print(df.info())

print("\nSummary Statistics:")

print(df.describe())

# Check for missing values

print("\nMissing Values:")

print(df.isnull().sum())

**# Step 3: Data Cleaning**

# Handle missing values (if any)

df = df.dropna()

print("\nMissing Values After Cleaning:")

print(df.isnull().sum())

# Handle outliers in Population\_Growth\_Rate using IQR

q1, q3 = df['Population\_Growth\_Rate'].quantile([0.25, 0.75])

iqr = q3 - q1

lower\_bound = q1 - 1.5 \* iqr

upper\_bound = q3 + 1.5 \* iqr

df['Population\_Growth\_Rate'] = df['Population\_Growth\_Rate'].clip(lower=lower\_bound, upper=upper\_bound)

**# Step 4: Visualizations**

# Correlation Matrix

plt.figure(figsize=(10, 8))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f')

plt.title('Correlation Matrix')

plt.show()

# Distribution of Population Growth Rate

plt.figure(figsize=(8, 6))

sns.histplot(df['Population\_Growth\_Rate'], bins=30, kde=True)

plt.title('Population Growth Rate Distribution')

plt.xlabel('Population Growth Rate (%)')

plt.show()

# Population Growth Rate vs Year (Trend)

plt.figure(figsize=(8, 6))

sns.lineplot(x='Year', y='Population\_Growth\_Rate', data=df)

plt.title('Population Growth Rate Trend Over Years')

plt.xlabel('Year')

plt.ylabel('Population Growth Rate (%)')

plt.show()

# Scatter Plot: Population Growth Rate vs GDP Per Capita

plt.figure(figsize=(8, 6))

sns.scatterplot(x='GDP\_Per\_Capita', y='Population\_Growth\_Rate', data=df)

plt.title('Population Growth Rate vs GDP Per Capita')

plt.xlabel('GDP Per Capita')

plt.ylabel('Population Growth Rate (%)')

plt.show()

**# Step 5: Prepare Data for Modeling**

# Features (X) and target (y)

X = df.drop('Population\_Growth\_Rate', axis=1)

y = df['Population\_Growth\_Rate']

# Scale features

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Split data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

**# Step 6: Train Multiple Regression Models**

models = {

    'Linear Regression': LinearRegression(),

    'Random Forest': RandomForestRegressor(n\_estimators=100, random\_state=42),

    'SVR': SVR(kernel='rbf')

}

results = []

for name, model in models.items():

    # Train model

    model.fit(X\_train, y\_train)

    # Predict

    y\_pred = model.predict(X\_test)

    # Evaluate

    mse = mean\_squared\_error(y\_test, y\_pred)

    r2 = r2\_score(y\_test, y\_pred)

    # Cross-validation score

    cv\_scores = cross\_val\_score(model, X\_scaled, y, cv=5, scoring='r2')

    results.append({

        'Model': name,

        'MSE': mse,

        'R2': r2,

        'CV\_R2\_Mean': cv\_scores.mean(),

        'CV\_R2\_Std': cv\_scores.std()

    })

    print(f"\n{name} Results:")

    print(f"MSE: {mse:.2f}")

    print(f"R^2: {r2:.2f}")

    print(f"Cross-Validation R^2: {cv\_scores.mean():.2f} ± {cv\_scores.std():.2f}")

**# Step 7: Compare Model Performance**

results\_df = pd.DataFrame(results)

print("\nModel Comparison:")

print(results\_df)

# Static plots for model comparison

plt.figure(figsize=(10, 6))

sns.barplot(x='Model', y='R2', data=results\_df, palette=['#36A2EB', '#FF6384', '#FFCE56'])

plt.title('Model Comparison: R² Scores')

plt.ylabel('R² Score')

plt.show()

plt.figure(figsize=(10, 6))

sns.barplot(x='Model', y='MSE', data=results\_df, palette=['#36A2EB', '#FF6384', '#FFCE56'])

plt.title('Model Comparison: Mean Squared Error')

plt.ylabel('MSE')

plt.show()

**# Step 8: Feature Importance (for Random Forest)**

rf\_model = models['Random Forest']

feature\_importance = pd.Series(rf\_model.feature\_importances\_, index=X.columns).sort\_values(ascending=False)

plt.figure(figsize=(10, 6))

sns.barplot(x=feature\_importance.values, y=feature\_importance.index)

plt.title('Feature Importance (Random Forest)')

plt.xlabel('Importance')

plt.show()

def predict\_population\_growth(model\_name, year, birth\_rate, death\_rate, migration\_rate, gdp\_per\_capita, urbanization\_rate, literacy\_rate, life\_expectancy):

    # Prepare input data

    input\_data = np.array([[year, birth\_rate, death\_rate, migration\_rate, gdp\_per\_capita, urbanization\_rate, literacy\_rate, life\_expectancy]])

    input\_scaled = scaler.transform(input\_data)

    # Select model

    model = models[model\_name]

    # Predict

    prediction = model.predict(input\_scaled)[0]

    # Return prediction and visualizations

    return (

        f"Predicted Population Growth Rate: {prediction:.2f}%",

        "correlation\_matrix.png",

        "population\_growth\_distribution.png",

        "population\_growth\_trend.png",

        "population\_growth\_vs\_gdp.png",

        "model\_comparison\_r2.png",

        "model\_comparison\_mse.png",

        "feature\_importance.png"

    )

# Define Gradio interface

iface = gr.Interface(

    fn=predict\_population\_growth,

    inputs=[

        gr.Dropdown(choices=['Linear Regression', 'Random Forest', 'SVR'], label="Select Model"),

        gr.Number(label="Year", value=2020),

        gr.Number(label="Birth Rate (per 1000)", value=10.0),

        gr.Number(label="Death Rate (per 1000)", value=5.0),

        gr.Number(label="Migration Rate (per 1000)", value=0.0),

        gr.Number(label="GDP Per Capita (USD)", value=10000.0),

        gr.Number(label="Urbanization Rate (%)", value=50.0),

        gr.Number(label="Literacy Rate (%)", value=90.0),

        gr.Number(label="Life Expectancy (years)", value=70.0)

    ],

    outputs=[

        gr.Textbox(label="Prediction"),

        gr.Image(label="Correlation Matrix"),

        gr.Image(label="Population Growth Rate Distribution"),

        gr.Image(label="Population Growth Trend"),

        gr.Image(label="Population Growth vs GDP"),

        gr.Image(label="Model Comparison: R² Scores"),

        gr.Image(label="Model Comparison: MSE"),

        gr.Image(label="Feature Importance (Random Forest)")

    ],

    title="Population Growth Rate Prediction",

    description="Enter values for the features and select a model to predict the population growth rate. Visualizations show dataset insights and model performance."

)

# Launch Gradio interface

iface.launch()

**Results and Discussion**

**5.1 Output / Report**

Three distinct machine learning models—Linear Regression, Random Forest Regressor, and Support Vector Regressor (SVR)—were implemented and evaluated, and the following outcomes were derived from test data predictions:

| Model | Mean Squared Error (MSE) | R² Score |
| --- | --- | --- |
| Linear Regression | *[e.g., 102.4]* | *[e.g., 0.89]* |
| Random Forest Regressor | *[e.g., 85.6]* | *[e.g., 0.93]* |
| SVR (Support Vector) | *[e.g., 98.1]* | *[e.g., 0.90]* |

With the lowest error and highest R2 score, the Random Forest Regressor outperformed the others based on the evaluation metrics, demonstrating strong predictive power.   
A close match between model predictions and ground truth was also demonstrated by visual plots, such as actual vs. predicted population graphs.

**5.2 Challenges Faced**

During the course of this project, several challenges were encountered:

• Data Quality & Understanding: At first, the absence of documentation made it difficult to comprehend the dataset's structure and features.

• Feature Selection: It took some investigation to determine which columns contributed to a meaningful prediction.

• Model tuning: Experimenting with hyperparameters was necessary to get the best performance out of models like SVR and Random Forest.

• Scaling Issues: Careful preprocessing using programs like StandardScaler was necessary because some models were sensitive to unscaled data.

**5.3 Learnings**

The project gave participants invaluable practical experience using machine learning on actual data. Important lessons learnt include:   
• Regression Algorithms in Practice: developed a thorough understanding of the behaviour of various regression models on actual data.   
• End-to-end machine learning workflow: comprehended the entire process from data loading to the last assessment.   
• The significance of data preprocessing: discovered how model performance is impacted by data scaling, cleaning, and splitting.   
• Capabilities for Model Evaluation: gained expertise in choosing the best model and analysing evaluation metrics.   
• Visualisation Techniques: Matplotlib and Seaborn were used to enhance the visualisation of data and outcomes.

**CONCLUSION**

**6.1 Summary**

This project focused on predicting population growth using machine learning techniques as part of the AI/ML curriculum at Lovely Professional University. By working with a real-world dataset and applying supervised learning methods, the project successfully demonstrated how regression models can forecast future trends based on historical data.

Three models were implemented and evaluated—**Linear Regression**, **Random Forest Regressor**, and **Support Vector Regressor (SVR)**. Among them, **Random Forest Regressor** yielded the best performance in terms of accuracy and generalization, as reflected by its superior R² Score and lower Mean Squared Error.

Throughout the process, the project covered key aspects of the machine learning lifecycle: data preprocessing, model training, evaluation, and prediction. It also emphasized the importance of data visualization and interpretation for drawing insights.

The hands-on experience gained through this project not only enhanced technical proficiency in Python and ML libraries but also deepened the understanding of how data-driven decision-making can be applied in real-world scenarios such as urban planning, policy-making, and resource management.