Project Name

Members' Names:

Bautista Lotes Kaye

Loyola, Dave Jarold

Marquez, Djullance

Salandanan, Jericho

Section: BS ECE - T4A

Date Submitted: January 23, 2024

Submitted To: Engr. Aisa Mijeno-Labastilla, PCpE

Major Assessment 3 / Capstone Assessment

Training and Evaluation of the Gathered Data

First Semester SY 2023-2024

I. Introduction / Problem Statement • Describe the problem you are trying to give a solution in doing the training and evaluation of the gathered data. State what SDG you are targeting

Unemployment is considered as a key measure of economic health. It becomes an indicator of the economic struggle of workers to obtain work and contribute to the economy of a country. It simply implies that the higher the unemployment rate, the less total economic production. The measure of unemployment excludes people who leave the labor force due to reasons like disability, pursuit of higher education, and retirement. Additionally, the unemployment rate is usually used as its key indicator, which is determined by dividing the number of unemployed individuals by the total labor force [1].

The project is about the analysis of world unemployment rates from 1991 to 2021. The aim of analyzing this problem is to gain insight into global employment trends over the last three decades and understand how various economic, social, and political issues influence employment rates. The Sustainable Development Goal (SDG) addressed by this project is SDG 8: "Decent Work and Economic Growth." This goal aims to foster long-term, inclusive, and sustained economic development, full and productive employment, and decent jobs for all [2].

Through the analysis of global unemployment rates, an in-depth analysis of the global employment landscape is provided as it correlates with this objective. It is also vital in promoting economic growth and ensuring that all individuals will have access to job opportunities.

II. Review of Related Literature • Describe here the technical know-how that you learned and researched related to your problem,

What Is Unemployment

Unemployment serves as a crucial economic indicator, reflecting the capacity of workers to secure gainful employment and contribute to overall economic productivity. A higher number of unemployed individuals corresponds to a reduction in total economic production. Conversely, a low unemployment rate indicates that the economy is likely operating close to its full capacity, fostering increased output, wage growth, and elevated living standards. Nevertheless, an excessively low unemployment rate may signal potential challenges, such as an overheated economy, inflationary pressures, and tight labor conditions for businesses seeking additional workers. Balancing unemployment rates is essential for maintaining a healthy economic equilibrium that encourages sustainable growth and prosperity [3].

Global unemployment rate 2003-2022

In 2022, the global unemployment rate exhibited a notable decline, decreasing by 0.4 percentage points (-6.45 percent) compared to the previous year. The unemployment rate represents the proportion of the economically active population actively seeking employment but currently without work. It is important to note that this metric excludes economically inactive groups, including long-term unemployed individuals, children, and retirees. This decrease in the unemployment rate suggests a positive trend in economic conditions, reflecting an increase in employment opportunities for the actively engaged workforce worldwide [4].

Youth Unemployment Causes and Solutions

Globally, the reported figure of 73 million unemployed youth belies the actual extent of the issue, with 620 million identified as not in employment, education, or training (NEET) by the World Bank. The origins of youth unemployment are complex, rooted in various factors. The aftermath of the financial crisis exacerbated pre-existing challenges in labor markets and education systems, leading to precarious job opportunities. Widespread skills mismatch, attributed to inadequate vocational training and curricula misaligned with industry needs, further impedes young people. Additionally, the absence of entrepreneurship and lifeskills education, coupled with challenges in accessing capital for aspiring entrepreneurs, exacerbates the problem. In low-income countries, a digital divide compounds the skills gap, restricting opportunities for those lacking access to technology. Addressing these intertwined challenges is imperative for developing comprehensive global solutions to youth unemployment [5].

Unemployment and mental health

Unemployment, characterized by the absence of employment while actively seeking work, consistently associates with detrimental health outcomes. The negative impact encompasses stress, diminished self-esteem resulting from job loss, financial hardships, insecurity, and reduced future earnings potential. The social security system, involving processes like claims, work capability testing, and job search conditions, can exacerbate mental health challenges. Extended periods of unemployment worsen these consequences, impacting mental health, life satisfaction, and physical well-being. Prolonged pandemic-related restrictions, causing extended periods of reduced income and job loss, amplify concerns regarding the enduring health effects of such circumstances [6].

Global Unemployment Crisis Continues

Within the world's wealthiest nations, encompassing members of the Organization for Economic Cooperation and Development (OECD), approximately 34 million individuals find themselves unemployed. In the European Union, there has been a notable rise in unemployment over the past year, reaching an average of 11.3 percent of the workforce, with significant increases observed in countries such as France, Germany, Italy, and Sweden. In contrast, the United States has experienced intensified job creation, leading to a dip in unemployment below 5 percent. While both the U.S. and the United Kingdom have seen declines in unemployment rates, there is a simultaneous trend of widening income disparities in these countries [7].

Employment is Needed

COVID-19 has led to massive job losses, particularly among youth and women In 2020, the global unemployment rate rose to 6.5 percent, marking a 1.1 percentage point increase from the previous year. The number of unemployed individuals worldwide surged by 33 million, reaching a total of 220 million, with an additional 81 million people exiting the labor market entirely. Latin America and the Caribbean, as well as Europe and Northern America, experienced unemployment rate hikes of at least 2 percentage points. Notably, youth and women faced disproportionate impacts, with employment losses of 8.7 percent and 5.0 percent, respectively, compared to 3.7 percent for adults and 3.9 percent for men. Before the pandemic, youth unemployment was already three times higher than that of adults. The crisis prompted more women than men to leave the labor force for childcare responsibilities, exacerbating longstanding gender gaps in labor force participation rates [8].

III. Gathered Data / Dataset • Describe here your dataset (type, size, repositories, links, etc.) • Link to the shared dataset

The "New Unemployment dataset" was gathered from the website Kaggle. Originally named "unemployment dataset", it was created by ANJALI PANT to describe the health of the world economy for the past 31 years in terms of its unemployment rate. The dataset contains about 235 countries, some countries in particular are divided based on their cardinal directions, which is why the countries in total are 235 instead of 195. The file contained about 40.7KB and was imported in a CSV format, but was later changed by the proponents to 203KB to better suit the lines of codes that the proponents will use in training and testing. Each country contains data regarding their rate of unemployment dating from 1991 to 2021 with its corresponding value per year. Furthermore, the four columns in the dataset are divided into Country Names, Country Codes, Year, and Value. Overall, the modified dataset contains four columns and 7,286 rows in a CSV format. The original dataset can be accessed through the link below and the modified dataset can be accessed in the link below the original dataset.

Original Dataset: https://www.kaggle.com/datasets/pantanjali/unemployment-dataset/data?fbclid=IwAR2nDIx9V6DhVvrf2i3hHAzlRxo2SLuMyv7kPb_mePkA4fSB-nDpvzp9KNk

Modified Dataset: https://drive.google.com/file/d/16DuzGWF388gDlliL9a687Xf9Z3KID2Nm/view?usp=drive_link

IV. Objectives • State here the main objective and the specific objectives to achieve the main objective

The main objective of the project is to comprehensively analyze world unemployment rates from 1991 to 2021, contributing to a better understanding of global employment trends and their implications for Sustainable Development Goal 8: Decent Work and Economic Growth. To achieve the main objective, the project specifically aims to:

· Import Data for training and testing.

· Analyze the data and create a graph.

· Filter the dataset to the chosen country (Philippines)

· Analyze the data from the chosen country.

· Write a code for a Linear Regression and Logistic Regression analysis.

· Train the models with the chosen dataset

· Display the graphical results of each model.

V. Conceptual • Describe here how you will manipulate your data to achieve the main objective of training and evaluating the gathered data

The provided code includes essential Python library imports for data analysis and visualization, with NumPy imported as 'np,' Pandas as 'pd,' Matplotlib for plotting as 'plt,' and Seaborn as 'sns.' Additionally, the code brings in the Linear Regression model from scikit-learn. Warnings are imported to manage warning messages, and the statement 'warnings.filterwarnings("ignore")' is employed to suppress these warnings. Finally, '%matplotlib inline' is a Jupyter Notebook magic command that ensures Matplotlib plots are displayed directly in the notebook. Together, these imports and configurations establish a comprehensive environment for conducting data analysis and creating visualizations.

To import a dataset in CSV format for analysis, the code snippet utilizes the Pandas library in Python. By specifying the path to the CSV file and using the pd.read_csv() function, the dataset is read and loaded into a Pandas DataFrame named 'df.' This DataFrame serves as a tabular representation of the data, providing a versatile structure for subsequent analysis and exploration. Adjust the file path within the code to match the location of the specific CSV file being used. Once the dataset has been imported, the checking for any null values is performed to ensure that there are no null values in the dataset. If there were any null values, adjustments would be needed to handle them appropriately.

Next, plotting the dataset initially results in a graph with several lines, making it crowded. To focus on the Philippines, the code filters the dataset using the statement 'country = df[df['Country Code'] == 'PHL'].'

A Linear Regression model is instantiated using scikit-learn's LinearRegression class. The model is then trained on the training data using the fit method, where 'X_train' represents the independent variable (features) and 'y_train' represents the dependent variable (target). This process involves adjusting the parameters of the linear regression model to minimize the difference between the predicted values and the actual values in the training set. After the execution of these lines, the 'model' variable holds a trained linear regression model ready for making predictions on new or unseen data based on the learned patterns from the training set.

Predictions on the test set are generated using the trained Linear Regression model. The model.predict(X_test) function applies the learned patterns from the training set to the independent variable 'X_test,' resulting in the predicted values stored in the 'y_pred' variable. The subsequent lines, where 'test_predict' is assigned the predicted values, represent an identical prediction procedure. The 'test_predict' variable now contains the model's predictions for the test set, which can be further evaluated and compared against the actual target values ('y_test') to assess the model's performance.

Scikit-learn's metrics modules are utilized to compute and evaluate the performance of the Linear Regression model on the test set. The mean absolute error (MAE) and mean squared error (MSE) are calculated using the mean_absolute_error and mean_squared_error functions, respectively. These metrics provide quantitative measures of the model's accuracy by comparing the predicted values ('y_pred') with the actual target values ('y_test'). The computed MAE and MSE values are then printed, offering insights into the average magnitude and squared differences between the predicted and actual values. Lower values for both metrics generally indicate better predictive performance.

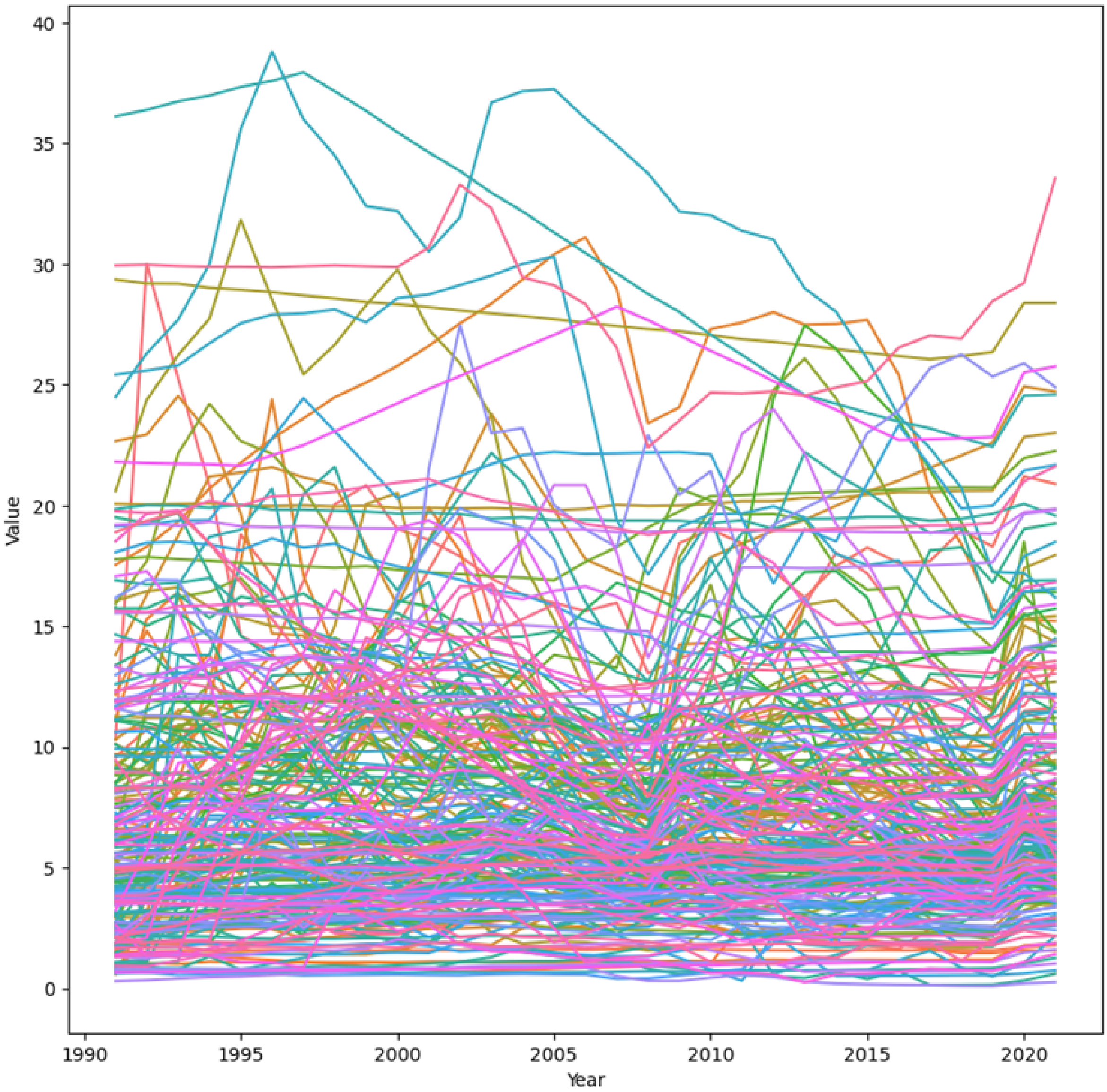
For Logistic Regression, a binary classification task is initiated by setting a threshold value of 2. The 'Values' column is created in the Pandas DataFrame 'df' to represent binary classifications (1 for values greater than the threshold, and 0 otherwise) based on whether the 'Value' column exceeds the specified threshold. The independent variable 'X' is assigned the 'Year' column, and the dependent variable 'y' is set to the newly created 'Values' column. The dataset is then split into training and testing sets using the train_test_split function, following the same principles as before. This sets the stage for training and evaluating a machine learning model for binary classification based on the selected features.

A Logistic Regression model is instantiated using scikit-learn's LogisticRegression class. The model is then trained on the training data using the fit method, where 'X_train' represents the independent variable (features) and 'y_train' represents the binary target variable (0 or 1). Logistic Regression is suitable for binary classification tasks, and the fit process involves learning the parameters that best fit the logistic regression equation to the training data. After this step, the 'model' variable holds a trained logistic regression model ready for making predictions on new or unseen data.

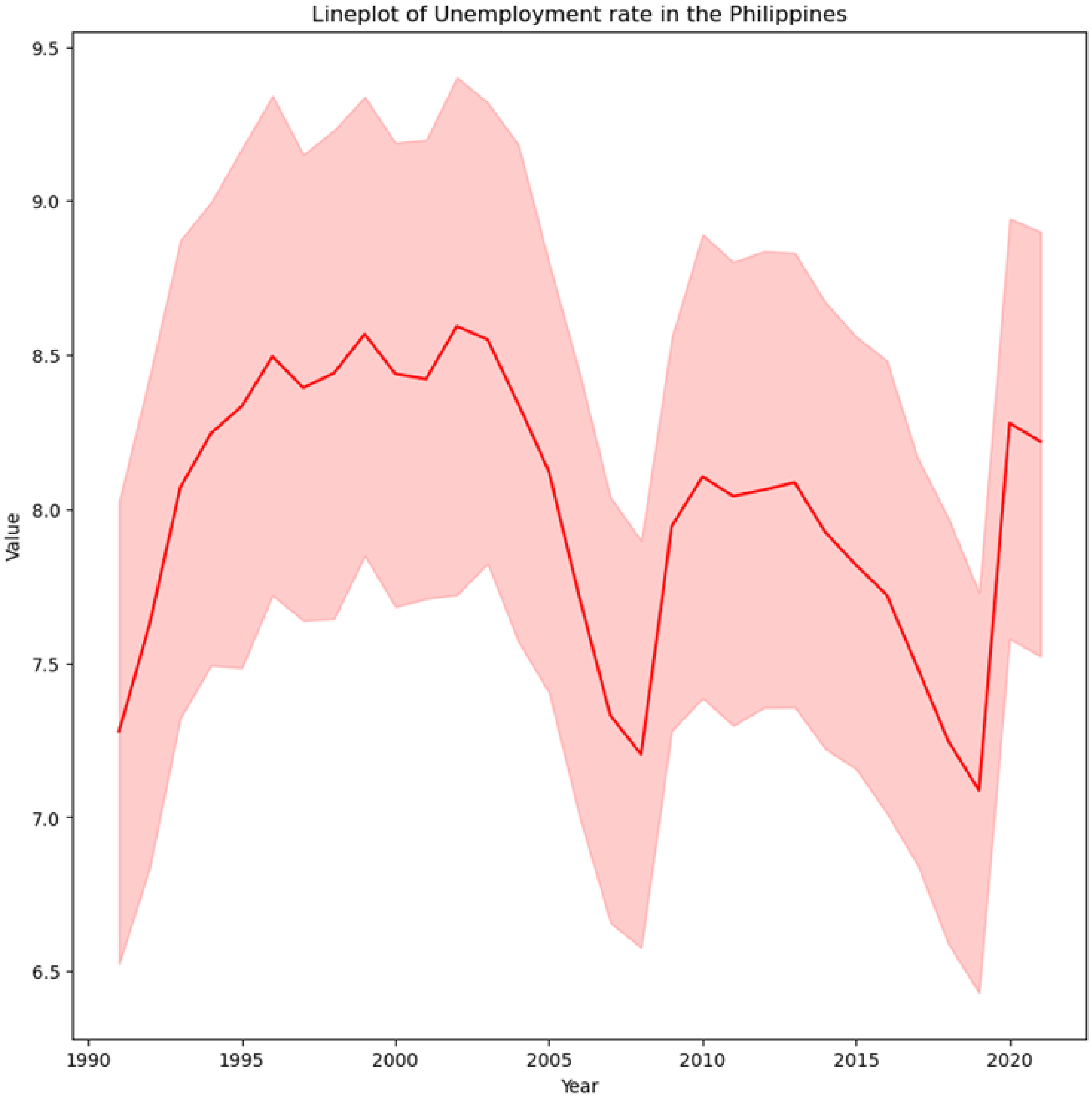
The Logistic Regression model is evaluated on the test set using key classification metrics. The predict method is employed to obtain model predictions, and scikit-learn's metrics modules are utilized to compute accuracy, precision, recall, and F1 score. Accuracy measures the overall correctness of predictions, precision assesses the accuracy of positive predictions, recall evaluates the ability to capture positive instances, and F1 score provides a balanced metric between precision and recall. The resulting scores are printed, offering insights into the model's effectiveness in binary classification.

A new data point is represented by the value '2.7' for the 'Year' feature. The Logistic Regression model is then used to predict the corresponding binary classification, indicating whether the 'Value' (unemployment rate) is above the threshold set earlier. The prediction is obtained using the predict method with the new data point, and the result is printed to the console. This allows for the model's application to unseen data, providing a binary classification prediction for the given 'Year' value.

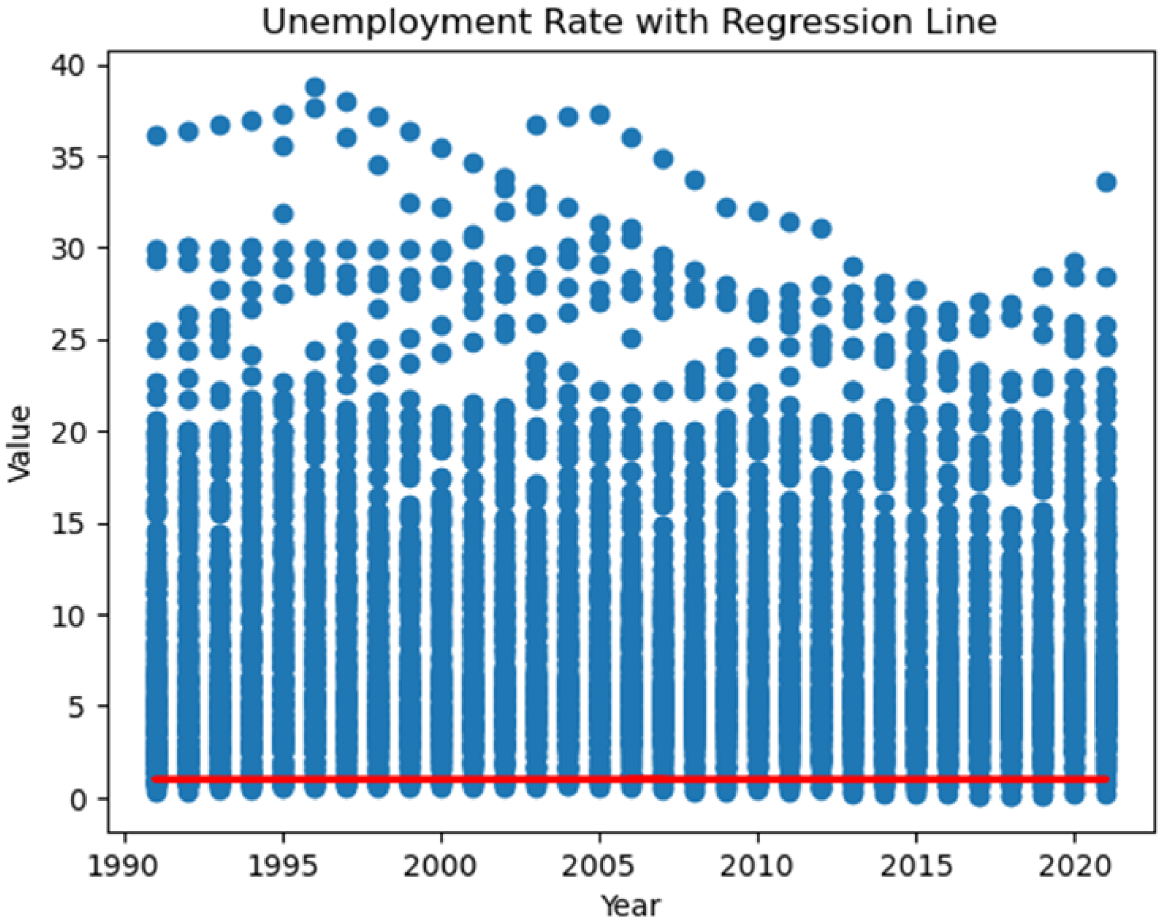
VI. Data and Results • Discuss here the results of each conducted training and evaluation



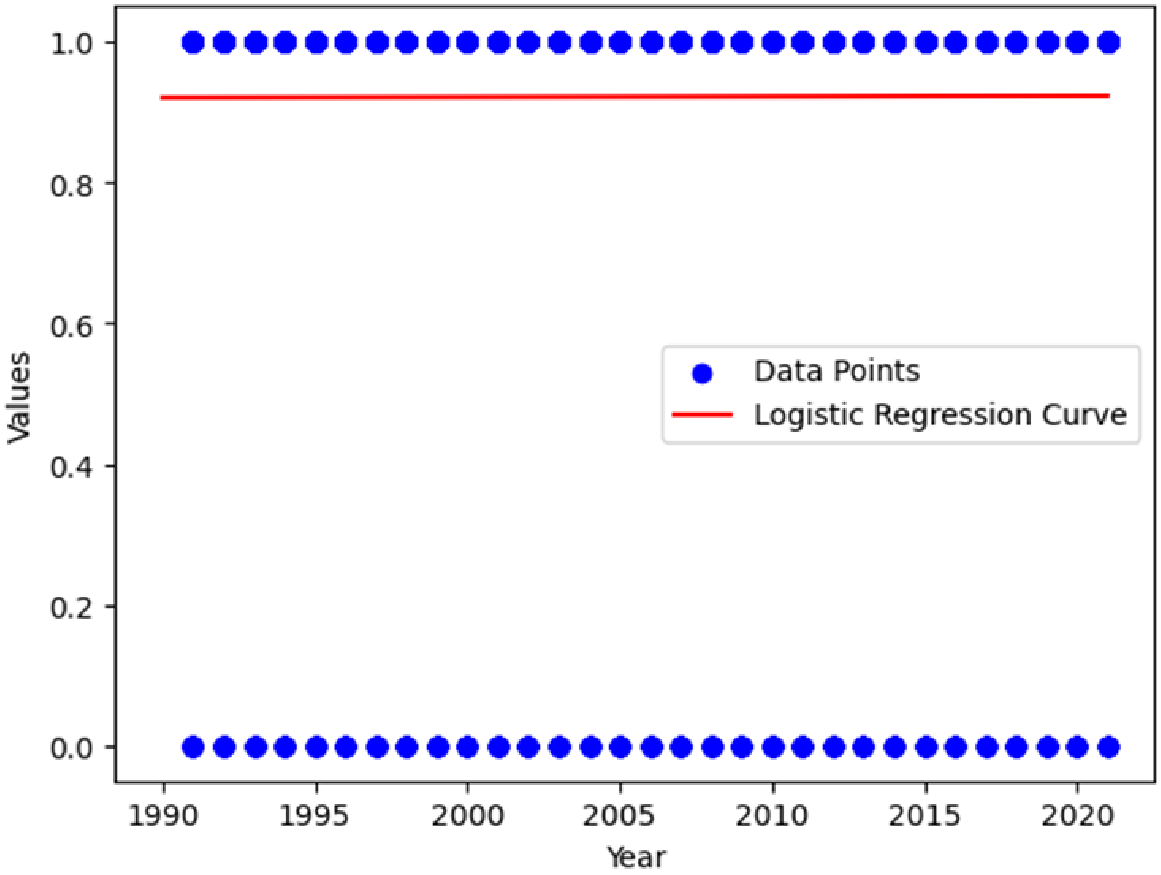
The visual complexity observed in the graph is due to the inclusion of data for every country in the dataset. Each line on the plot represents the unemployment rates of different countries, resulting in a dense and cluttered appearance. This makes it challenging to discern distinct trends or patterns, as the numerous overlapping lines create a graph that appears chaotic and lacks clear organization.



The analysis will specifically concentrate on the Philippines. To enhance the clarity and interpretability of the graph, the code snippet 'country = df[df[‘Country Code’] == ‘PHL’]' is employed. This filters the dataset to isolate data pertaining solely to the Philippines. Consequently, the resulting graph will be more organized and easier to understand, exclusively representing the unemployment rates of the selected country.



For Linear Regression, Matplotlib is used for data visualization. A scatter plot is created using the 'Year' and 'Value' columns from the Pandas DataFrame 'df' to display the actual data points. Additionally, a red regression line is overlaid on the scatter plot using the 'X_test' values and corresponding predictions ('y_pred') from the trained Linear Regression model. The resulting visualization provides a visual representation of the model's fit to the data, helping to assess how well the linear regression captures the underlying patterns. The title, xlabel, and ylabel functions are used to label the plot appropriately, enhancing its interpretability. The plt.show() function is then called to display the plot.



For Logistic Regression, a logistic regression curve is visualized alongside the scatter plot of the original data points using Matplotlib and NumPy. The x*values array is generated to cover a range of years (1990 to 2021), and the logistic function is applied using the model coefficients (model.coef* and model.intercept_) to calculate corresponding y_values. The scatter plot displays the original data points

VII. Conclusion • Must be parallel with the objectives

In conclusion, the proponents were able to import the dataset and create an initial graph depicting global unemployment rates for every country in the dataset. However, the graph appears visually complex and cluttered due to the multitude of overlapping lines, making it challenging to discern specific trends or patterns. To address this issue, the analysis focuses specifically on the Philippines, utilizing a code snippet to isolate and present the country's unemployment rates in a more organized and interpretable manner.

The proponents trained the models using the filtered dataset. The subsequent visualizations employ Linear Regression and Logistic Regression techniques using Matplotlib, Pandas, and NumPy. The Linear Regression plot showcases the model's fit to the data, aiding in assessing its ability to capture underlying patterns. Similarly, the Logistic Regression visualization features a logistic regression curve alongside the scatter plot of original data points, providing a clear representation of the model's application to the dataset.

These refined visualizations enhance the interpretability of the data and facilitate a more insightful analysis of unemployment trends in the Philippines. The capstone project proves highly beneficial by addressing the visual complexity of the initial global unemployment graph through a focused analysis on the Philippines. This approach enhances interpretability and showcases practical data manipulation skills.

The subsequent visualizations, employing Linear and Logistic Regression techniques, not only provide insights into the underlying patterns of unemployment data but also demonstrate the practical application of machine learning models. The integration of Matplotlib, Pandas, and NumPy in these visualizations adds real-world relevance, contributing to a comprehensive learning experience.

Overall, the capstone project serves as a valuable opportunity to apply and solidify data science skills, fostering a deeper understanding of data analysis and modeling techniques.

VIII. Share your thoughts individually and overall feedback after completing this course.

Bautista:

Engaging in the "Data Science with Artificial Intelligence" course was not only enjoyable but also incredibly enriching. I gained extensive knowledge in coding, with a particular focus on Python and the utilization of Jupyter Notebook. This platform proved to be invaluable for refining and advancing my coding skills. It became an important tool, enabling me to explore and conduct training and testing on datasets using various machine learning models. The course introduced me to Artificial Intelligence and Machine Learning, which further enhanced my technical skills. I'm able to understand real-world applications using Jupyter Notebook, through constructing predictive models and performing data analysis. The emphasis on manipulating data showcased its relevance for future applications. Overall, the course not only contributed to my technical skills but also helped me foster practical and experimental coding skills.

Loyola:

The "Data Science with Artificial Intelligence" course was a game-changer, especially with Jupyter notebook. This dynamic tool was crucial for improving coding skills, particularly in Python, and fostering an experimental mindset. The course focused on practical applications within Jupyter Notebook, like building predictive models and doing exploratory data analysis, which not only enhanced technical skills but also gave me a practical understanding of turning techniques into actionable insights. This experience equipped me with a versatile toolset and mindset, ready for the dynamic field of data science and artificial intelligence in my future work.

Marquez:

The course “Data Science with Artificial Intelligence” taught me many things in terms of coding. The use of Jupyter notebook has helped me create various ways of extracting and manipulating data that will help me in the future to come. Moreover, the course deepened my coding proficiency, particularly in Python, as Jupyter Notebook served as a dynamic platform for implementing data manipulation techniques. The ability to execute code in a cell-wise fashion allowed me to troubleshoot and iterate swiftly, fostering a more iterative and experimental coding mindset. The course's emphasis on real-world applications within Jupyter Notebook, such as building predictive models and conducting exploratory data analysis, has not only honed my technical skills but also provided me with a practical understanding of how these techniques translate into actionable insights.

Salandanan:

The "Data Science with Artificial Intelligence" course was beneficial, equipping me with substantial coding expertise, particularly in Python, and proficiency in utilizing Jupyter Notebook. This dynamic platform proved indispensable for honing my coding skills, facilitating exploration and application of various machine learning models on datasets. The course broadened my understanding of Artificial Intelligence and Machine Learning, elevating my technical capabilities. The practical focus on real-world applications, such as building predictive models and conducting exploratory data analysis within Jupyter Notebook, provided valuable insights. The emphasis on data extraction and manipulation underscored its practical relevance for future applications. Overall, the course not only significantly enhanced my technical skills but also cultivated an experimental and iterative mindset in coding, preparing me for the evolving landscape of data science and artificial intelligence in my future pursuits.

IX. Program Codes (Jupyter notebook)

In [3]:

```
# Import all libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.linear_model import LinearRegression

import warnings


warnings.filterwarnings("ignore")
%matplotlib inline
```

In [4]:

```
# Read the dataset into a DataFrame
df = pd.read_csv("C:\\data capstone\\new unemployment dataset.csv")

# Display the first few rows of the dataset
df.head(7286)
```

Out[4]:

	Country Name	Country Code	Year	Value
0	Africa Eastern and Southern	AFE	1991	7.80
1	Africa Eastern and Southern	AFE	1992	7.84
2	Africa Eastern and Southern	AFE	1993	7.85
3	Africa Eastern and Southern	AFE	1994	7.84
4	Africa Eastern and Southern	AFE	1995	7.83
...	...	...	...	...
7280	Zimbabwe	ZWE	2017	4.78
7281	Zimbabwe	ZWE	2018	4.80
7282	Zimbabwe	ZWE	2019	4.83
7283	Zimbabwe	ZWE	2020	5.35
7284	Zimbabwe	ZWE	2021	5.17

7285 rows × 4 columns

In [5]:

```
# Check for missing values
np.sum(df.isnull())
```

Out[5]:

```
Country Name    0
Country Code    0
Year            0
Value           0
dtype: int64
```

In [6]:

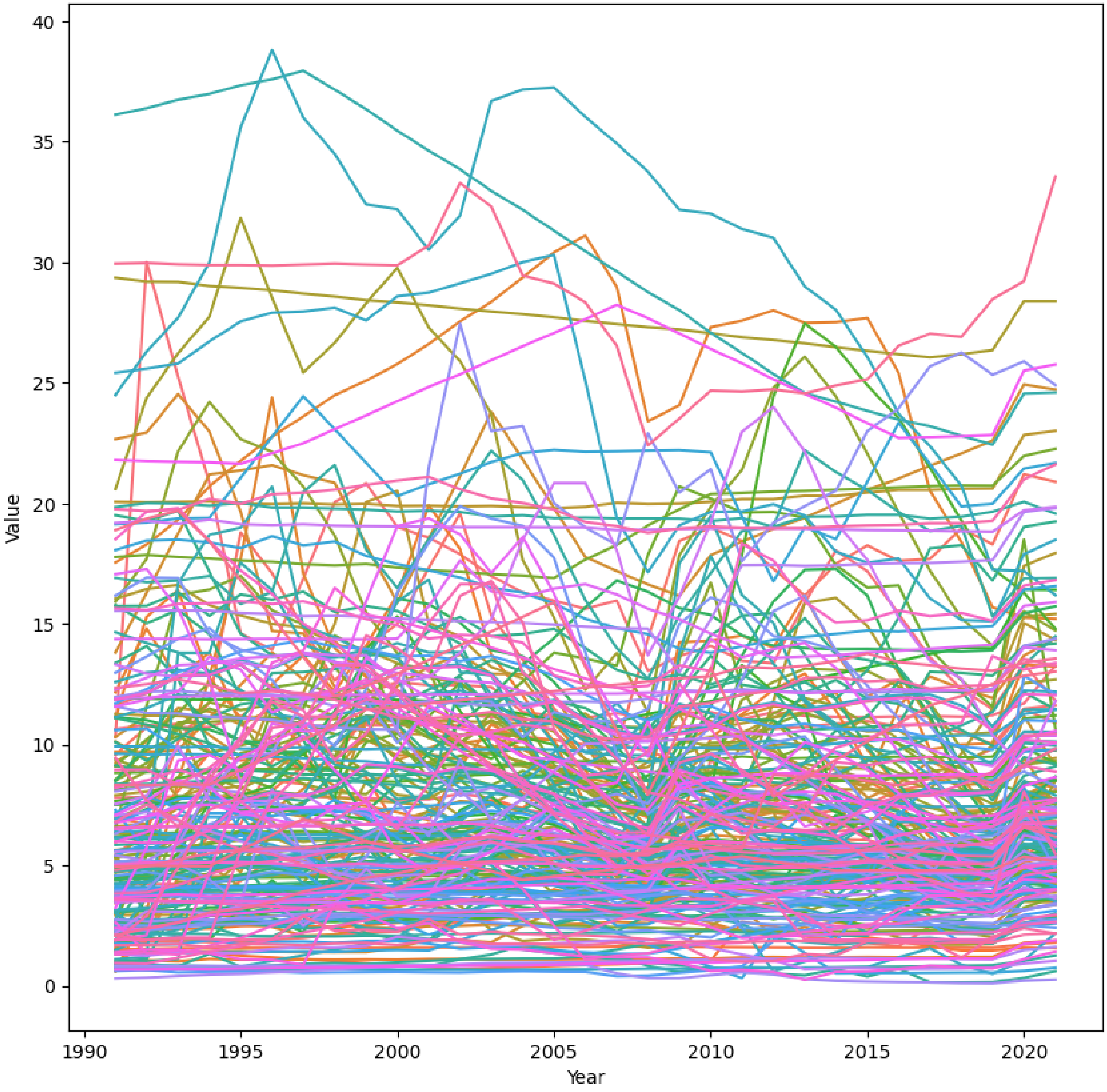
```
# Additional Information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7285 entries, 0 to 7284
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Country Name  7285 non-null   object 
 1   Country Code  7285 non-null   object 
 2   Year          7285 non-null   int64  
 3   Value         7285 non-null   float64
dtypes: float64(1), int64(1), object(2)
memory usage: 227.8+ KB
```

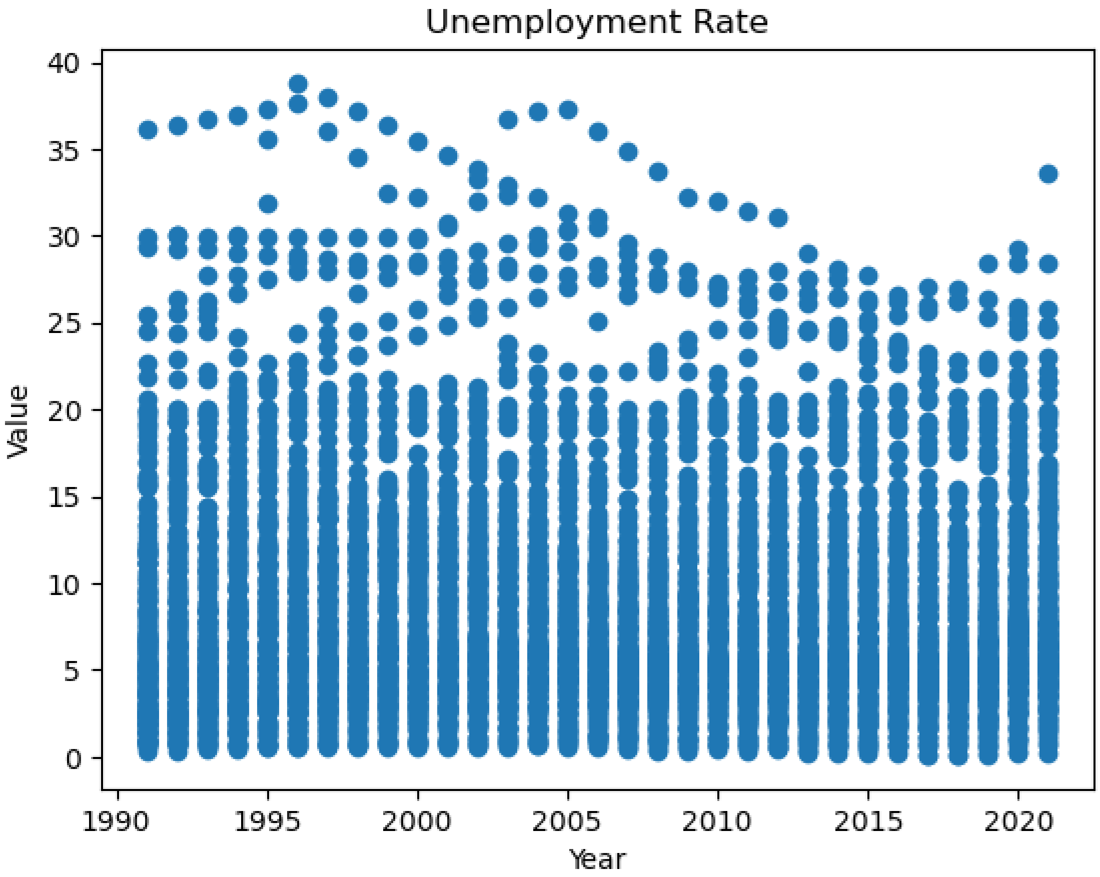
Visualization

In [7]:

```
# Average Unemployment rate per year in the world

plt.figure(figsize=(10, 10))
sns.lineplot(x='Year', y='Value', hue='Country Name', data=df, legend=False)
plt.show()
```

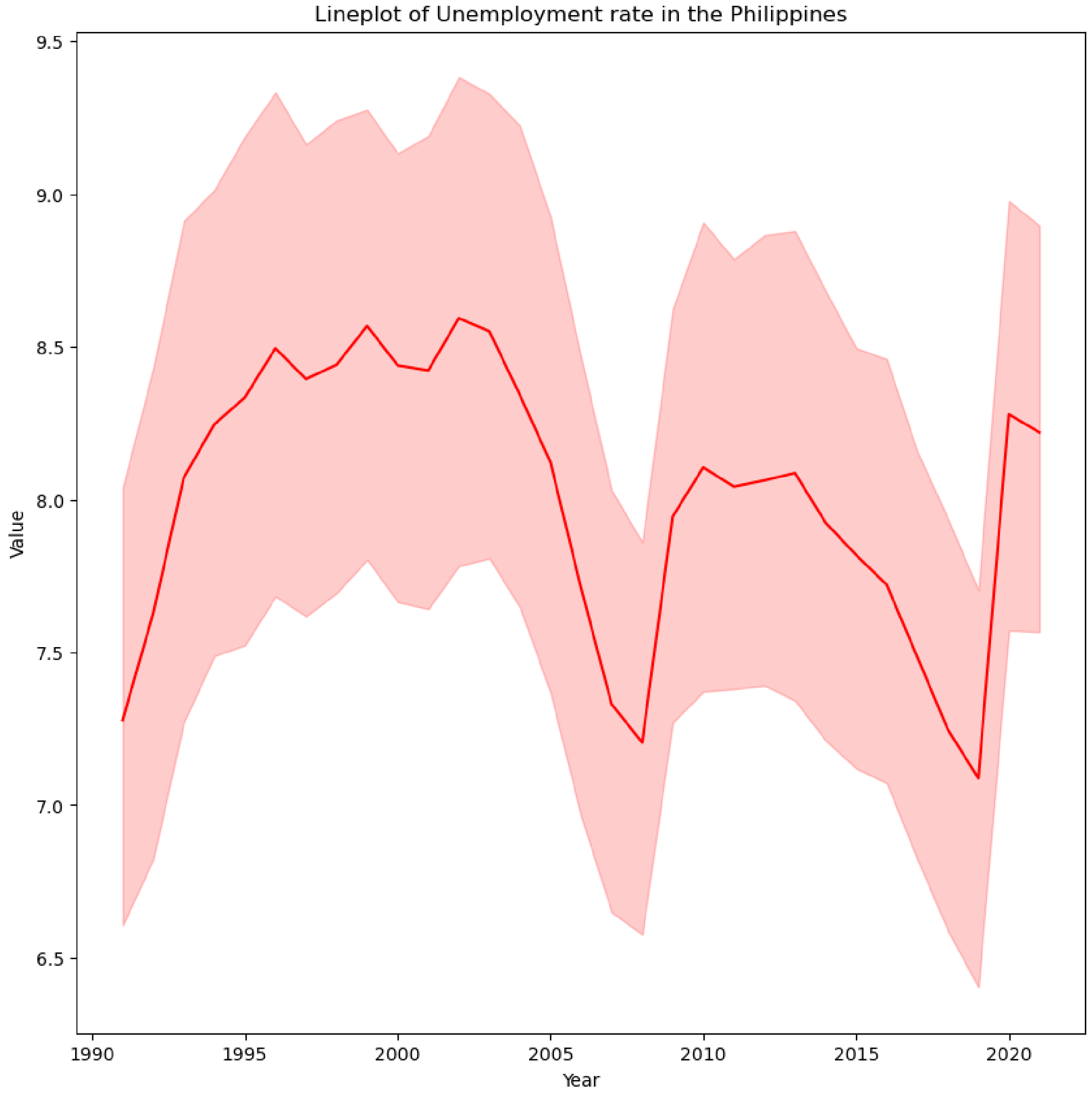


```
In [9]: plt.scatter(df['Year'], df['Value'])
        plt.title('Unemployment Rate')
        plt.xlabel('Year')
        plt.ylabel('Value')
        plt.show()
```



In [21]:

```python
# Average Unemployment rate per year in the Philippines

country = df[df['Country Code'] == 'PHL']

plt.figure(figsize=(10,10))
sns.lineplot(x='Year', y='Value', data=df,color='red', legend=True)
plt.title(f'Lineplot of Unemployment rate in the Philippines')

plt.xlabel('Year')
plt.ylabel('Value')

plt.show()
```



Linear Regression

In [10]:

```
# Splitting the Dataset
from sklearn.model_selection import train_test_split

X = df[['Year']]
y = df['Value']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

In [11]:

```
# Build and train the linear regression model
from sklearn.linear_model import LinearRegression

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

Out[11]:

```
▾ LinearRegression
LinearRegression()
```

In [12]:

```
# Making Predictions
y_pred = model.predict(X_test)
```

In [13]:

```
test_predict = model.predict(X_test)
test_predict
```

Out[13]:

```
array([7.99211308, 8.23162868, 7.75259748, ..., 7.79251675, 7.77255711,
       8.01207272])
```

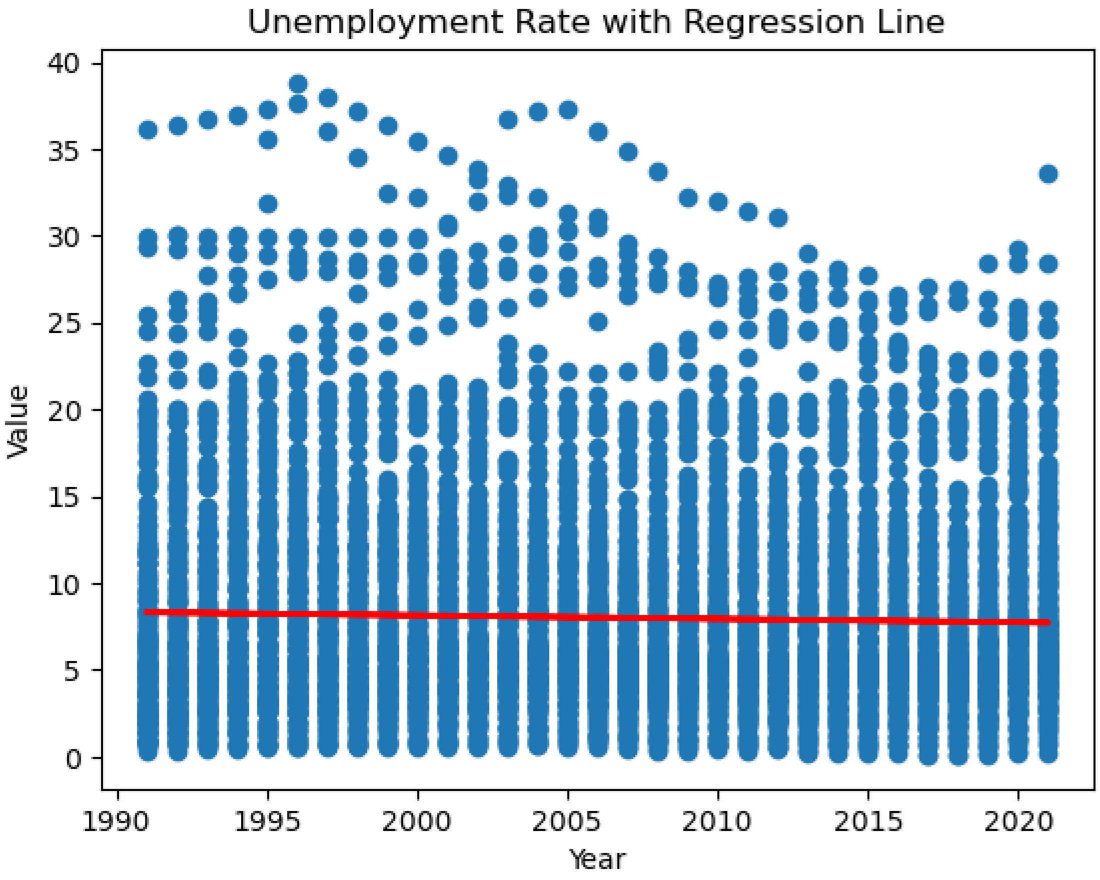
In [14]:

```
# Evaluationg the model
from sklearn.metrics import mean_absolute_error, mean_squared_error

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
print(f'MAE: {mae}')
print(f'MSE: {mse}')
```

```
MAE: 4.426717687856889
MSE: 36.27184496893957
```

In [15]:

```
# Visualizing the Regression Line
plt.scatter(df['Year'], df['Value'])
plt.plot(X_test, y_pred, color='red', linewidth=2)
plt.title('Unemployment Rate with Regression Line')
plt.xlabel('Year')
plt.ylabel('Value')
plt.show()
```



Logistic Regression

In [16]:

```
# Set a threshold to classify as high or low production
threshold = 2

# Converting to Binary values
df['Values'] = (df['Value'] > threshold).astype(int)

# Separate features (X) and  (y)
X = df[['Year']]
y = df['Values']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

In [17]:

```
# Build and train a logistic regression model
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model.fit(X_train, y_train)
```

Out[17]:

```
▾ LogisticRegression
LogisticRegression()
```

In [18]:

```
# Evaluating the model
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```

```
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1 Score: {f1}")
```

```
Accuracy: 0.9238160603980783
Precision: 0.9238160603980783
Recall: 1.0
F1 Score: 0.9603995718872637
```

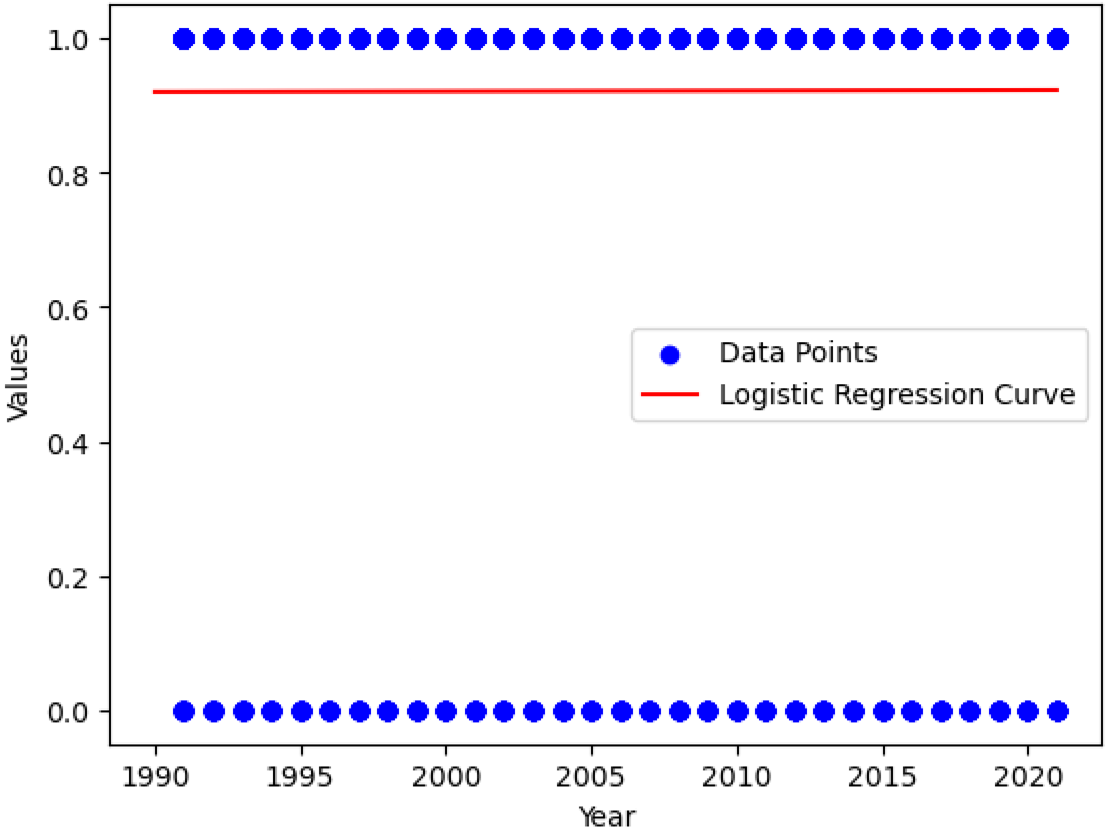
In [19]:

```
# Making Predictions
new_data_point = [[2.7]]
prediction = model.predict(new_data_point)
print(f"Prediction for unemployment rate in the world: {prediction[0]}")
```

```
Prediction for unemployment rate in the world: 1
```

In [20]:

```
# Visualizing the boundary of the logistic regression model:
import matplotlib.pyplot as plt
import numpy as np

x_values = np.linspace(1990, 2021, 100)  # Adjust the number of points (100 in this ex

# Use the logistic function with the model coefficients to calculate y_values
y_values = 1 / (1 + np.exp(-(model.coef_ * x_values + model.intercept_)))

# Scatter plot of the data points
plt.scatter(X, y, color='blue', label='Data Points')

plt.plot(x_values, y_values[0], color='red', label='Logistic Regression Curve')

# Set labels and legend
plt.xlabel('Year')
plt.ylabel('Values')
plt.legend()

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


1.0
0.8
0.6
0.4
0.2
0.0
Values
1990
1995
2000
2005
2010
2015
2020
Year
● Data Points
Logistic Regression Curve