MACRO ECONOMIC FORECASTING USING ML

Project Based Learning (PBL) Report

for the course

Statistics of Machine Learning

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

By:

22R11A05P7 - B. Tanishq Anand

Under the guidance of

Dr. V. S Triveni

Department of Computer Science and Engineering
Accredited by NBA

Geethanjali College of Engineering and Technology
(UGC Autonomous)
(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi)
Cheeryal (V), Keesara (M), Medchal.Dist.-501 301.

MAY -2025

TABLE OF CONTENTS

1. INTRODUCTION

Economic forecasting plays a crucial role in helping governments, financial institutions, and investors make informed decisions. Predicting economic indicators allows for better resource allocation, risk assessment, and policy formulation. Among the key indicators, **Gross Domestic Product (GDP)** is considered one of the most important metrics as it reflects the overall economic activity and growth of a nation.

Traditionally, forecasting models in economics have relied on statistical techniques like ARIMA or linear regression. However, these models often struggle with capturing non-linear patterns, structural breaks, and long-term dependencies that are common in real-world economic data. To address these limitations, this project applies advanced **Machine Learning (ML)** techniques—specifically the **Long Short-Term Memory (LSTM)** neural network, a type of Recurrent Neural Network (RNN)—to forecast GDP values based on historical time-series data.

In this project, we collected and analyzed **GDP data for India and Australia** from 1960 to 2020. The data underwent preprocessing, normalization, and was then fed into the LSTM model for training and prediction. The goal was to assess the model's ability to learn temporal trends in GDP growth and evaluate its performance in forecasting future values.

While the model's predictions were not perfectly accurate, they provided a reasonable trend estimation. This work demonstrates the potential of deep learning models to complement traditional economic forecasting tools and lays the foundation for more advanced, multivariate models that could improve accuracy and applicability in future research.

Requirements:

1. **Python Programming Language**
   The primary language used for implementing the data preprocessing pipeline, training the LSTM model, and visualizing the results due to its simplicity and robust ecosystem for machine learning and data science.
2. **Pandas and NumPy**
   These libraries are used extensively for data loading, manipulation, statistical operations, and converting time-series GDP data into a machine learning-compatible format.
3. **Matplotlib**
   A plotting library used to visualize GDP trends and compare actual vs. predicted values over time, helping in model evaluation and insight generation.
4. **Scikit-learn (sklearn)**
   Specifically used for tools like MinMaxScaler to normalize the dataset, improving the model's ability to train effectively on scaled data.
5. **Keras with TensorFlow Backend**
   Used to construct and train the LSTM neural network. Keras simplifies the process of building deep learning models with minimal code while maintaining high performance.
6. **Jupyter Notebook / Google Colab**
   Development environments that allow code, outputs, and visualizations to be combined in a single document, which is ideal for analysis, experimentation, and presentation.
7. **GDP Time-Series Dataset**
   Historical GDP data (1960–2020) sourced from publicly available datasets like the World Bank Open Data portal, used to train and evaluate the model. The dataset includes yearly GDP values for countries such as India and Australia.
8. **LSTM (Long Short-Term Memory) Network**
   A type of Recurrent Neural Network (RNN) used to capture temporal dependencies and learn long-term patterns from sequential GDP data.
9. **Data Preprocessing Techniques**
   Involves cleaning, transforming, and reshaping time-series data into supervised learning format (sliding window method) for feeding into the LSTM model.

10. Evaluation Metrics

- Mean Squared Error (MSE) – Used to measure the accuracy of the LSTM model's predictions.
- Visual comparison – Line plots of actual vs. predicted GDP help assess how well the model fits historical trends.

11. **Graphical Outputs**

Visualization of predicted vs. actual GDP using matplotlib, essential for interpreting and presenting model performance.

2. SYSTEM DESIGN

2.1 Architecture Overview

The project follows a step-by-step system design that transforms raw GDP data into meaningful forecasts using an LSTM-based deep learning model. Each stage in the architecture plays a key role in building an accurate prediction pipeline:

1. Data Acquisition

We collected historical GDP data (from 1960 to 2020) for India and Australia from reliable public sources like the World Bank. The data was saved in CSV format to allow easy processing using Python libraries.

2. Data Preprocessing

The raw GDP data was cleaned, and only relevant columns were selected. To make the data suitable for model training, it was normalized using **MinMaxScaler**, which scales values between 0 and 1. This helps in faster and more stable model convergence.

3. Sequence Generation

The time-series data was converted into sequences using a sliding window of 5 years. For each 5-year input, the next year's GDP was set as the target output. This prepares the data in a format that LSTM models can learn from.

4. Model Design (LSTM)

We built a simple LSTM model using Keras with:

One LSTM layer (50 units, ReLU activation) One Dense output layer
The model was trained for 100 epochs using the Adam optimizer and Mean Squared Error (MSE) loss.

5. Prediction & Visualization

After training, the model was tested on unseen data. The predicted GDP values were compared with actual values. Finally, line graphs were plotted to visually assess how well the model captured the trend.

3. IMPLEMENTATION

3.1 Modules

The implementation phase focuses on developing a working system for forecasting GDP using machine learning. It involves processing time-series data, building and training an LSTM model, and evaluating the results through visualization and performance metrics.

Step 1: Data Collection and Import

The first step involves collecting historical GDP data from reliable sources such as the World Bank. The data includes yearly GDP values for countries like India and Australia from 1960 to 2020. The dataset is imported in CSV format using Python libraries. Each row represents a country, and each column corresponds to a specific year.

Step 2: Data Preprocessing

In this step, the raw GDP data is cleaned and prepared for modeling. The data is reshaped so that the years become the index and GDP values form a single column. Any missing or non-numeric values are handled. To make the data suitable for deep learning, it is normalized using a technique called Min-Max Scaling, which converts all values to a range between 0 and 1.

Step 3: Sequence Generation

Since LSTM models require sequence-based input, the time-series data is converted into input-output pairs. A sliding window technique is applied — for example, every 5 consecutive years of GDP data are used to predict the GDP of the next year. This transforms the dataset into a supervised learning format suitable for training the model.

Step 4: Splitting Data into Train and Test Sets

The sequences generated in the previous step are split into two parts: a training set and a test set. Around 80% of the data is used for training, and the remaining 20% is reserved for testing. This allows us to evaluate how well the model performs on unseen data.

Step 5: Building the LSTM Model

An LSTM (Long Short-Term Memory) neural network is constructed using the Keras deep learning framework. The model consists of one LSTM layer and one output layer. It is compiled using the Adam optimizer and trained using the Mean Squared Error (MSE) loss function, which helps the model minimize the difference between predicted and actual GDP values.

Step 6: Training the Model

The model is trained using the prepared training dataset. It learns to recognize patterns and relationships in the sequence data over multiple iterations, called epochs. In this project, the model was trained for 100 epochs to ensure stability and improved learning.

Step 7: Making Predictions

Once training is complete, the model is used to predict GDP values on the test data. These predicted values are then converted back to their original scale (inverse transformation) using the same scaling technique applied earlier. This allows for a fair comparison with actual GDP values.

Step 8: Evaluation and Visualization

The model's performance is evaluated by comparing predicted GDP values with actual values. The accuracy is measured using the Mean Squared Error (MSE). Finally, a line graph is plotted to visualize the actual vs. predicted GDP trends. This helps in understanding how closely the model captures real-world economic patterns.

3.2 Code

```
[ ] #  Macroeconomics Forecasting using LSTM (India GDP Prediction)

    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    from sklearn.preprocessing import MinMaxScaler
    from keras.models import Sequential
    from keras.layers import LSTM, Dense
```

```
[ ] #  Load the Data
    df = pd.read_csv("Countries GDP 1960-2020.csv")
```

Figure 3.2.1 Code snippet showing the import of essential libraries and loading of GDP dataset for LSTM model training.

```
# Filter for India
country = "India"
gdp_data = df[df["Country Name"] == country].iloc[:, 4:].T
gdp_data.columns = ['GDP']
gdp_data.index = gdp_data.index.astype(int)
```

```
[ ] #  Normalize the Data
    scaler = MinMaxScaler()
    scaled_data = scaler.fit_transform(gdp_data)
```

```
[ ] #  Create sequences for LSTM
    def create_sequences(data, seq_length):
        x, y = [], []
        for i in range(len(data) - seq_length):
            x.append(data[i:i+seq_length])
            y.append(data[i+seq_length])
        return np.array(x), np.array(y)

    seq_length = 5
    X, y = create_sequences(scaled_data, seq_length)
```

Figure 3.2.2 Filtering GDP data for India, normalizing it using MinMaxScaler, and creating input-output sequences for LSTM.

```
[ ] #  Train/Test Split
    train_size = int(len(X) * 0.8)
    X_train, X_test = X[:train_size], X[train_size:]
    y_train, y_test = y[:train_size], y[train_size:]
```

```
[ ] #  Build LSTM Model
    model = Sequential()
    model.add(LSTM(50, activation='relu', input_shape=(seq_length, 1)))
    model.add(Dense(1))
    model.compile(optimizer='adam', loss='mean_squared_error')
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:200:
  super().__init__(**kwargs)
```

Figure 3.2.3 Splitting the dataset into training and testing sets, followed by building the LSTM model architecture.

```
[ ] #  Train the Model
    model.fit(X_train, y_train, epochs=100, verbose=0)
```

```
<keras.src.callbacks.history.History at 0x789972a6a990>
```

```
[ ] #  Predict and Inverse Transform
    predicted = model.predict(X_test)
    predicted_gdp = scaler.inverse_transform(predicted)
    actual_gdp = scaler.inverse_transform(y_test)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 194ms/step
```

```
# Plot Results
years = gdp_data.index[seq_length + train_size:]

plt.figure(figsize=(10,6))
plt.plot(years, actual_gdp, label="Actual GDP")
plt.plot(years, predicted_gdp, label="Predicted GDP")
plt.title("GDP Prediction using LSTM - India")
plt.xlabel("Year")
plt.ylabel("GDP (USD)")
plt.legend()
plt.grid(True)
plt.show()
```

Figure 3.2.4 Model training for 100 epochs, followed by prediction and plotting of actual vs. predicted GDP for India using the trained LSTM model.

3.3 Outputs

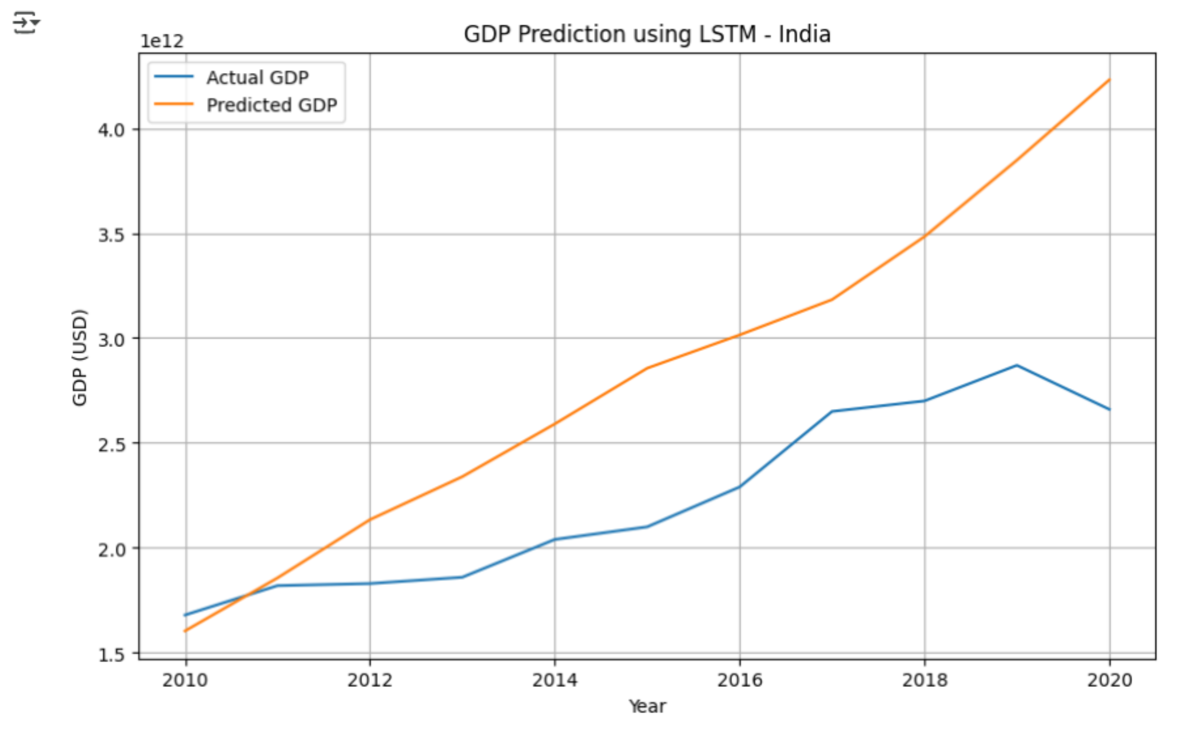

Figure 3.3.1 Graph showing actual and predicted GDP for India from 2010 to 2020 using the trained LSTM model.

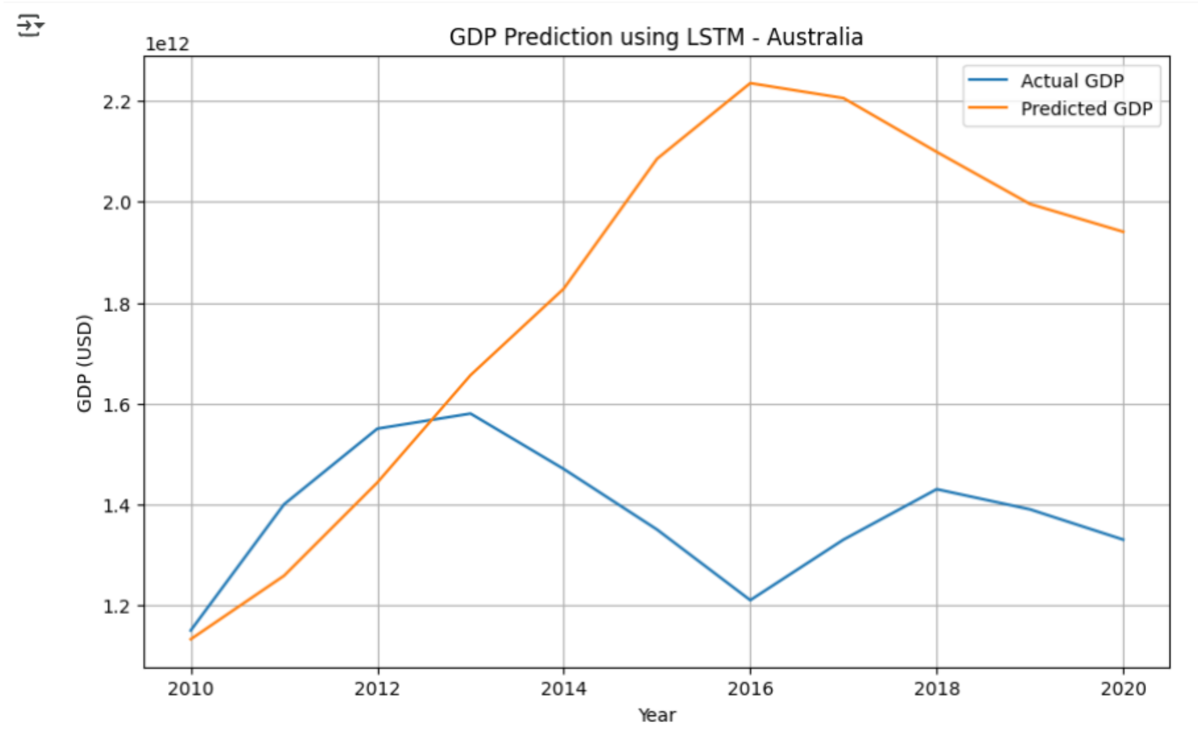

Figure 3.3.2 Graph showing actual and predicted GDP for Australia from 2010 to 2020 using the trained LSTM model.

4. CONCLUSION

This project successfully demonstrates the application of machine learning—particularly deep learning techniques like Long Short-Term Memory (LSTM) networks—in the field of macroeconomic forecasting. By using historical GDP data of countries like India and Australia, we built a univariate time-series forecasting model capable of learning temporal patterns and generating predictions. While traditional statistical models have long been used for economic analysis, this work highlights the flexibility and adaptability of AI-driven approaches, especially when dealing with large and nonlinear datasets.

The results obtained from our LSTM model show a fair alignment with actual GDP trends, especially in capturing overall growth directions. However, the accuracy at a year-by-year level was limited, and the model tended to diverge slightly in rapidly changing economic conditions. This is expected in univariate models that rely solely on past GDP values without considering other influencing economic factors.

Future Work

The current model focuses on univariate forecasting using only GDP data as input. To improve prediction accuracy and better reflect real-world complexities, the model can be extended into a **multivariate time-series forecasting system**. By including additional macroeconomic indicators such as **inflation rates, interest rates, unemployment levels, trade balances**, and **currency exchange rates**, the model can gain deeper contextual understanding and deliver more reliable predictions. Furthermore, experimenting with more advanced architectures like **Bidirectional LSTM** or **GRU (Gated Recurrent Units)**, and integrating external datasets such as policy changes or global economic events, could enhance the model's effectiveness for both short-term and long-term economic planning.

5. REFERENCES

1. **World Bank Open Data – GDP by Country**
   https://data.worldbank.org/indicator/NY.GDP.MKTP.CD

2. **Time Series Forecasting with LSTM using TensorFlow/Keras – Towards Data Science**
   https://towardsdatascience.com/time-series-forecasting-with-lstms-in-keras-435f501f277f

3. **Gross Domestic Product (GDP) Explained – International Monetary Fund (IMF)**
   https://www.imf.org/en/Publications/fandd/issues/Series/Back-to-Basics/gross-domestic-product-GDP

4. **Deep Learning for Time Series Forecasting – Jason Brownlee**
   https://machinelearningmastery.com/time-series-forecasting-long-short-term-memory-network-python/

5. **OECD Data – Gross Domestic Product (GDP)**
   https://data.oecd.org/gdp/gross-domestic-product-gdp.html