Task: Develop a Time Series Forecasting Model and Deploy it as a Web

Application

**Objective**: The project is developed as part of Internship requirement of the course **‘Python with Data Science’**. I have developed a time series forecasting project name ‘***Forecasting Nepal’s Economic and Social Development’*** and also deployed through the streamlit.

The project is tried to forecast Nepal Economic and Social Development using different indexes. It consists of **4 models with MAE and RMSE value below 16** which indicates that it is able to forecast accurately.

1. Data collection and Exploration:

**Data collection:**

Data is collected from the **pandas\_datareader** library which has the World Bank API interface inside it. So, we can directly access the data of Nepal as well from this library.

**Data Description**:

Project has 4 models.

1. Gross Domestic Product (GPD)
2. Life Expectancy
3. Total Population
4. GDP per capital

The data collected from the period of **1980 to 2024 that is 45 years**.

**Gross Domestic Product (GDP)**: Simply this is a total monetary value of all goods and services produced within a country’s boarder in one year.

It measures the country’s economy.

This index shows how much the country is earning using their resources. Higher GPD means country is earning more with its domestic products and services. So, this index will show the country’s economic status.

GPD increase from the 1.94B to 42.9 billion.

Life Expectancy: This index shows how long the Nepalese people live. From dataset we can see that life expectancy reaches from **48 to 70 years**.

**Total Population**: This index shows the total population of our country which is also increases from **1.55 corer to 2.97 corer**.

**GDP per capital**: This index shows the average income per person. It is calculated by total GDP divide by total population.

So, this index shows how wealthy the average person is 125 to 1447.

Note: The unit of GDP and total population is in billion and currency is in dollar. Also, we drop the Literacy rate.

1. Data Cleaning and Transformation:

**Drop Columns and fill the value:**

The dataset has 45 rows and 4 columns. We drop the **Literacy rate column** because it has 84.4% missing values.

There is one missing value in **Life Expectancy** of **2.22%** so we can fill the missing value. Since, the life expectancy column is linear means it increases year by year so we can’t **fill the value by mean and median** as it will break the trend so we will fill with **forward fill**

Also, we drop the **country column** as the dataset is about the Nepal

**Data type and set index:**

Now, the data type of ‘year’ column is change to datatime format and set the index year.

1. **Exploratory Data Analysis (EDA):**

For every column we plot the columns in line chart and bar chart.

GDP: The major trend here is a significant increase in GDP over this period. it appears that the graph becomes steeper after 2003, indicating a faster rate of GDP growth from that point onwards.

**Life Expectancy**: The major trend here is also increases life expectancy over a period. The value is gradually increases year by year

**Total Population**: The trend here is also increases as per year. In 45 years, the population is almost double.

**GDP\_per\_capital**: It follows the same trend as the GDP.

1. **Feature Engineering:**

The column GDP\_per capital is not directly access from the dataset. It created by dividing the total GDP by total population so we are able to create a new feature.

1. **Model Building:**

For modeling, the project uses the Prophet to forecasting the multivariable.

1. **Train Test split:**

**Manual train test split:**

The data is segregate into train set (from 1980 to 2015) and test set (2016 to 2024).

**Prophet style future forecast vs test:**

We make a future forecasting dataframe of 10 years with frequency yearly.

1. **Model Optimization:**

Note; Since the RMSE and MAE is already very low means models are performing very well. So, Cross validation isn’t necessary in this case.

1. **Pickling the file**:

All the models are dumps and use it to create an app using the vs code. Then the model is push in github.

1. **Model Deployment**: The model Deploy through the streamlit.