## TASK 2: Unemployment in India

## import pandas as pd

## import matplotlib.pyplot as plt

## import seaborn as sns

## from statsmodels.tsa.arima.model import ARIMA

## # Load the dataset

## file\_path = r"D:\User\Downloads\Unemployment in India (1).csv"

## try:

## # Load CSV file with appropriate encoding to handle BOM

## df = pd.read\_csv(file\_path, encoding='utf-8-sig')

## # Display the first few rows to inspect the content

## print(df.head())

## # Check columns to verify column names

## print(df.columns)

## # Remove leading and trailing whitespaces from column names

## df.columns = df.columns.str.strip()

## # Fill missing values with forward fill method

## df.ffill(inplace=True)

## # Convert date column to datetime

## if 'Date' in df.columns:

## df['Date'] = pd.to\_datetime(df['Date'])

## else:

## raise KeyError("'Date' column not found in DataFrame.")

## # Unemployment rate over time for different states

## plt.figure(figsize=(14, 8))

## for state in df['Region'].unique():

## state\_data = df[df['Region'] == state]

## plt.plot(state\_data['Date'], state\_data['Estimated Unemployment Rate (%)'], label=state)

## plt.title('Unemployment Rate Over Time by State')

## plt.xlabel('Date')

## plt.ylabel('Unemployment Rate (%)')

## plt.legend(loc='upper right', bbox\_to\_anchor=(1.15, 1))

## plt.xticks(rotation=45)

## plt.show()

## # Boxplot of unemployment rate by state

## plt.figure(figsize=(14, 8))

## sns.boxplot(x='Region', y='Estimated Unemployment Rate (%)', data=df)

## plt.title('Unemployment Rate by State')

## plt.xlabel('State')

## plt.ylabel('Unemployment Rate (%)')

## plt.xticks(rotation=90)

## plt.show()

## # Pairplot to explore relationships

## sns.pairplot(df[['Estimated Unemployment Rate (%)', 'Estimated Employed', 'Estimated Labour Participation Rate (%)']])

## plt.show()

## # Select a state for time series analysis

## state = 'Maharashtra'

## state\_data = df[df['Region'] == state]

## # Resample data to monthly average

## state\_data.set\_index('Date', inplace=True)

## monthly\_unemployment = state\_data['Estimated Unemployment Rate (%)'].resample('M').mean()

## # Plot the resampled data

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

## monthly\_unemployment.plot()

## plt.title(f'Monthly Average Unemployment Rate in {state}')

## plt.xlabel('Date')

## plt.ylabel('Unemployment Rate (%)')

## plt.show()

## # Fit the ARIMA model

## model = ARIMA(monthly\_unemployment.dropna(), order=(5, 1, 0))

## model\_fit = model.fit()

## # Forecast

## forecast = model\_fit.forecast(steps=12)

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

## plt.plot(monthly\_unemployment, label='Historical')

## plt.plot(forecast, label='Forecast', color='red')

## plt.title(f'Unemployment Rate Forecast in {state}')

## plt.xlabel('Date')

## plt.ylabel('Unemployment Rate (%)')

## plt.legend()

## plt.show()

## # Save DataFrame to CSV

## output\_file\_path = r"D:\User\Downloads\Unemployment\_in\_India\_cleaned.csv"

## df.to\_csv(output\_file\_path, index=False)

## print(f"DataFrame successfully saved to '{output\_file\_path}'")

## except FileNotFoundError:

## print(f"Error: File '{file\_path}' not found.")

## except KeyError as e:

## print(f"Error: {e}. Please check the structure of your CSV file.")

## except Exception as e:

## print(f"An error occurred: {e}")

### Documentation: Unemployment Analysis in India

#### Overview

This documentation outlines the methodology and design choices for analyzing unemployment data in India using Python. The objective is to provide a structured approach to loading, cleaning, analyzing, visualizing, and forecasting unemployment rates across different states.

#### Implementation Steps

##### 1. Loading the Dataset

The dataset "Unemployment in India (1).csv" is loaded into a Pandas DataFrame using pd.read\_csv(). Proper encoding (utf-8-sig) is specified to handle potential UTF-8 BOM characters at the beginning of the file. This step ensures that the data is read correctly into memory for subsequent analysis.

##### 2. Data Cleaning

Data cleaning involves handling missing values and ensuring data consistency. Missing values are filled using forward fill (ffill()), which propagates the last valid observation forward to maintain temporal integrity in the dataset.

##### 3. Data Exploration and Visualization

Exploratory data analysis (EDA) is performed to understand the distribution and trends in unemployment rates across different states. Visualization techniques such as line plots, box plots, and pair plots are employed to identify patterns, outliers, and correlations between variables.

##### 4. Time Series Analysis and Forecasting

Time series analysis is utilized to model and forecast unemployment rates over time. Techniques like ARIMA (AutoRegressive Integrated Moving Average) are applied to capture trends, seasonality, and other temporal patterns in the data. Monthly average unemployment rates are often resampled from daily or weekly data to facilitate modeling.

##### 5. Saving Cleaned Data

The cleaned and processed DataFrame is saved to a new CSV file ("Unemployment\_in\_India\_cleaned.csv"). This step ensures that the cleaned dataset is preserved for future analysis or reporting purposes.

#### Design Choices and Challenges

* **Encoding Handling**: The choice of utf-8-sig encoding ensures compatibility with UTF-8 BOM characters commonly found in CSV files. This prevents issues related to mismatched column names or encoding errors during data loading.
* **Data Cleaning Strategy**: Using ffill() for missing values propagation maintains the integrity of time-series data by preserving temporal sequences. This approach is preferred over arbitrary filling methods to avoid distorting underlying patterns in the data.
* **Visualization Techniques**: Matplotlib and Seaborn libraries are chosen for data visualization due to their flexibility in generating clear and informative plots. This facilitates effective communication of insights derived from the data exploration phase.
* **Model Selection**: ARIMA is selected for time series forecasting due to its capability to handle autocorrelation, trend, and seasonality in the data. The choice of model parameters (e.g., (p, d, q) in ARIMA) may vary based on the specific characteristics observed in the unemployment data.

#### Conclusion

This documentation provides a structured approach to conducting unemployment analysis in India using Python. By following these steps and design considerations, stakeholders can gain valuable insights into labor market dynamics, trends, and potential forecasting scenarios. The project serves as a foundation for further exploration and application of data science methodologies to address socio-economic challenges related to unemployment.