


Data Collection and Preprocessing Phase

Date	29 November 2024
Team ID	740108
Project Title	Unveiling Climate Change Dynamics Through Earth Surface Temperature Analysis.
Maximum Marks	6 Marks

Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Heatmap Visualization	Convert temperature data into heatmap visualizations for geographic comparisons.
Regional Segmentation	Segment temperature data by geographic regions or time intervals to focus on specific trends or anomalies.
Standardized Data Inputs	Integrate batch normalization layers in the neural network to speed up training and enhance the model's performance.
Data Preprocessing Code Screenshots	
Loading Data	

Spatial Data Aggregation	<pre>[16] data['month']=data.index.month data['year']=data.index.year pivot=pd.pivot_table(data, values='LandAverageTemperature', index='month', columns='year', aggfunc='n pivot.plot(figsize=(20,6)) plt.title('Yearly Tempertaures') plt.xlabel('Months') plt.ylabel('Temperatures') plt.xticks([x for x in range(1,13)]) plt.legend().remove() plt.show()</pre>
Temperature Standardization	<pre># Standardize the 'LandAverageTemperature' column from sklearn.preprocessing import StandardScaler scaler = StandardScaler() data['LandAverageTemperature'] = scaler.fit_transform(data[['LandAverageTemperature']])</pre>
Synthetic Data Generation	<pre>train = data[:-60].copy() y = train['LandAverageTemperature'] lags_plots=48 figsize=(22,8) y = pd.Series(y) fig = plt.figure() ax1 = plt.subplot2grid((3, 3), (0, 0), colspan=2) ax2 = plt.subplot2grid((3, 3), (1, 0)) ax3 = plt.subplot2grid((3, 3), (1, 1)) ax4 = plt.subplot2grid((3, 3), (2, 0), colspan=2) y.plot(ax=ax1, figsize=figsize) ax1.set_title('Temperature Variation') plot_acf(y, lags=lags_plots, zero=False, ax=ax2); plot_pacf(y, lags=lags_plots, zero=False, ax=ax3); sns.distplot(y, bins=int(sqrt(len(y))), ax=ax4) ax4.set_title('Distribution Chart') plt.tight_layout()</pre>
Trend Change Detection	<pre>print('Results of Dickey-Fuller Test:') adfulltest = adfuller(y) adftest = pd.Series(adfulltest[0:4], index=['Test Statistic', 'p-value', 'Lags Used', 'Number of Observations Used']) adftest = round(adftest, 4) for key, value in adftest[4].items(): adftest["Critical Value (%s)"%key] = value.round(4) print(adftest) if adftest[0].round(2) < adftest[5].round(2): print('\nThe Test Statistics is lower than the Critical Value of 5%. \nThe serie seems to be stationary') else: print("\nThe Test Statistics is higher than the Critical Value of 5%. \nThe serie isn't stationary")</pre>
Noise Reduction in Measurements	NO

Heatmap Visualization	<pre>[46] monthly_seasonality=pivot.mean(axis=1) monthly_seasonality.plot(figsize=(20,6)) plt.title('Monthly Temperatures') plt.xlabel('Months') plt.ylabel('Temperature') plt.xticks([x for x in range(1,13)]) plt.show()</pre>
Regional Segmentation	<pre>[48] plt.figure(figsize=(22,6)) sns.lineplot(x=data.index, y=data['LandAverageTemperature']) plt.title('Temperatur Variation from 1760 until 2000') plt.show()</pre>
Standardized Data Inputs	<pre>[52] X = data.drop('LandAverageTemperature', axis =1) y = data['LandAverageTemperature']</pre>