**Data Collection and Preprocessing Phase**

|  |  |
| --- | --- |
| Date | 15 March 2024 |
| Team ID | SWTID1720707508 |
| Project Title | WarLens: Transfer Learning for Event Classification in Conflict Zones |
| Maximum Marks | 6 Marks |

**Data Exploration and Preprocessing Template**

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

|  |  |
| --- | --- |
| **Section** | **Description** |
| Data Overview | The image input size of the model is 224,224 |
| Univariate Analysis | Mean,model,media,range, variance and standard deviation, percentile and bar charts |
| Bivariate Analysis | Correlation coefficient and scatter plot |
| Multivariate Analysis | Correlation coefficient and scatter plot |
| Outliers and Anomalies | Identification and treatment of outliers. |
| **Data Preprocessing Code Screenshots** | |
| Loading Data | combat **=** "ImageData/Combat"  destroyed\_building **=** "ImageData/DestroyedBuildings"  fire **=** "ImageData/Fire"  humanterian **=** "ImageData/Humanitarian Aid and rehabilitation"  vehicles **=** "ImageData/Military vehicles and weapons" |
| Handling Missing Data | # Identifying missing values  missing\_values = df.isnull().sum()  print("Missing values:\n", missing\_values)  # Handling missing values by filling with the mean (for numerical columns)  df.fillna(df.mean(), inplace=True)  # Alternatively, drop rows with missing values  df.dropna(inplace=True) |
| Data Transformation | from sklearn.preprocessing import StandardScaler  # Scaling numerical features  scaler = StandardScaler()  numerical\_features = ['feature1', 'feature2'] # Replace with actual feature names  df[numerical\_features] = scaler.fit\_transform(df[numerical\_features]) |
| Feature Engineering | # Creating new feature by combining existing features  df['new\_feature'] = df['feature1'] \* df['feature2'] # Replace with actual feature names  # Converting categorical features to numerical (one-hot encoding)  df = pd.get\_dummies(df, columns=['categorical\_feature']) # Replace with actual feature names |
| Save Processed Data | # Save the cleaned and processed data to a new CSV file  df.to\_csv('processed\_data.csv', index=False)  print("Processed data saved successfully.") |