Exploratory Data Analysis (EDA):

By: Loga Aswin

It refers to the method of Studying and explorating records to apprehend their predominant traits, discover patterns, locate outliers & identify relationships between Variables.

- **Goals of EDA:**

1. Data cleaning: Handle missing values, duplicates, outliers & handle Categorical Values.

2. Data Visualization: Visual techniques represent Statistics Graphically. Histogram, box plots, Scatter plot, line plot, heat map & bar charts to identify Styles, trends & relationships with facts.

3. Feature Engineering: Contain Scaling, Normalization, binning, encoding, Variables.

4. Correlation & Relationships: Allow discover relationship & dependencies between Variables. Correlation analysis, Scatter plot & pair tabulation.

5. Data Segmentation: Divide information into Significant Segments based totally on Sure-Standard or traits.

6. Hypothesis Generation: Generating hypothesis / Studies Questions based on preliminary exploration of data.

7. Data Quality Assessment: Permits assessing nice & reliability of the info.
   ↳ Involve checking record integrity, consistency of Accuracy to make into Suitable for Analysis.

⊙ Handling Missing Values :

- **isnull ()** - check any missing values in dataset.

  → df. isnull . sum ()

  O/p → Return sum of np.nan (NULL) values in each column.

- **fillna ()** - fill value at NULL Places.

  → df['column']. fillna (value, inplace = True)

- **replace ()** - used to replace values in dataset.

  → df ['col'] . ~~dropna~~ replace (np.nan, values)

- **dropna ()** - Drop records with NULL Values.

  → df['col'] . dropna (axis = 0, how = 'any')

- **duplicated ()** - Checks if duplicates Present in dataset.

  → df . duplicated ()

  O/p : Returns total no. of duplicate rows.

- **drop. duplicates** - Drop the Duplicate rows.

  → df. drop.duplicates (keep = 'first', inplace = True)

  O/p : keep first copy & remove all other duplicates.

⊙ **Data Encoding** :

→ Encode Categorical data into numerical values.

* One-hot Encoding / Label Encoding :

⇝ from sklearn.preprocessing import LabelEncoder

⇝ encoder = LabelEncoder()

⇝ df['col'] = encoder.fit.transform(df['col'])

O/p → Assign num to each Category starting from 0.

⊙ **Data Visualization** :

↳ Analyze data in the form of Graphs/maps, easy understand trends/patterns.

1. **Boxplot** -

   sns.boxplot(x='col1', y='col2', data=df)

2. **Pairplot ()** . pairwise distribution in dataset.

   sns.pairplot(df, hue='col', height=2)

3. **Histogram** : Count the numeric values in a dataset.

   sns.histplot(x='col', data=df)