**Exploratory Data Analysis (EDA) in Data Science**

**Introduction**

This notebook is structured approach to performing **Exploratory Data Analysis (EDA)** using Python. This document offers a detailed overview of the repository, outlining its purpose, structure, and functionality.

EDA is a critical step in data science that involves analyzing datasets to understand patterns, detect anomalies, identify relationships, and summarize key statistics before applying machine learning models.

**Repository Overview**

The repository consists of multiple scripts and Jupyter notebooks, each dedicated to a specific phase of EDA. The primary components include:

**1. Data Loading (dataLoader.py)**

* This script is responsible for loading datasets from various sources, such as CSV and json files.
* It ensures that data is properly formatted and structured before further processing.
* If errors occur (such as incorrect file paths or missing files), appropriate error-handling mechanisms are implemented.

**2. Data Preprocessing (dataProcessing.py)**

* This module focuses on cleaning and preparing data for analysis.
* Key preprocessing steps include:
  + Handling missing values by either filling them with appropriate statistics (mean, median, mode) or removing them.
  + Encoding categorical variables to convert text-based data into numerical formats suitable for analysis.
  + Normalizing and standardizing numerical features to maintain consistency across datasets.

**3. Exploratory Data Analysis (eda.py)**

* This script performs EDA by generating summary statistics, identifying trends, and creating visualizations.
* Important steps include:
  + Displaying statistical summaries such as mean, median, and standard deviation.
  + Creating graphical representations like histograms, box plots, and scatter plots.
  + Generating correlation matrices to understand relationships between numerical features.

**4. Outlier Detection (outlierDetection.py)**

* This module detects and removes anomalies from the dataset.
* Outliers can distort data analysis and lead to incorrect conclusions, so different statistical methods are used:
  + **Z-score method**: Identifies data points that deviate significantly from the mean.
  + **Interquartile Range (IQR)**: Identifies extreme values beyond the acceptable range.
  + **Visualization techniques**: Box plots and distribution plots help in identifying outliers.

**5. Regression Modeling (regressionModel.py)**

* This script builds and evaluates **Linear Regression** models to identify relationships between independent variables and the target variable.
* Key steps in the modeling process:
  + Splitting data into training and testing sets to prevent overfitting.
  + Training the regression model to establish predictive relationships.
  + Evaluating model performance using metrics like **Mean Squared Error (MSE)**.

**6. Main Script (main.py)**

* This script integrates all the individual modules and orchestrates the workflow.
* It ensures a streamlined execution of data loading, preprocessing, analysis, outlier detection, and model training.
* By running this script, users can automate the entire EDA process.

**7. Jupyter Notebook (F223635\_BSE6A\_Assignment2.ipynb)**

* The notebook serves as an interactive environment where EDA techniques are applied step by step.
* It allows users to visualize trends dynamically, explore insights, and interpret findings.
* The notebook also provides flexibility for modifications and custom analyses.

**Data Files**

The repository includes various datasets in CSV format, each representing different stages of the data pipeline:

* **cleaned\_data.csv**: Dataset after preprocessing, where missing values have been handled and data is formatted correctly.
* **normalized\_data.csv**: Dataset where numerical features have been scaled to a standard range.
* **processed\_data.csv**: Final dataset that is ready for modeling and further analysis.

**Key Functionalities**

**1. Handling Missing Data**

* The repository ensures missing data is handled appropriately using techniques like mean/mode imputation or deletion.
* This prevents inconsistencies and errors in subsequent analysis.

**2. Data Visualization**

* Graphical methods such as scatter plots, histograms, and heatmaps are used to understand feature distributions and correlations.
* These visualizations help in identifying patterns and trends within the dataset.

**3. Outlier Detection & Removal**

* Detecting outliers is crucial to avoid distorted statistical inferences.
* Statistical methods like Z-score and IQR are used to filter anomalies.

**4. Regression Analysis**

* The repository includes a regression model to study the relationship between features.
* It helps in making predictions based on historical data trends.

**5. Automated Workflow Execution**

* The repository is designed to run as an automated pipeline where each module executes sequentially.
* This allows users to perform end-to-end data analysis with minimal manual intervention.

**Key Notes**

1. **Normalization is performed only when required for regression.**
   * If normalization is applied before EDA, it may **disrupt meaningful patterns** and affect visualization insights.
   * Normalization is crucial for models sensitive to feature scaling but should not be done prematurely.
2. **Log transformation is applied to skewed data** to reduce extreme values and make distributions more normal.
   * This ensures that the data follows assumptions required for linear regression and other statistical models.

**Conclusion**

This notebook provides a comprehensive framework for performing **Exploratory Data Analysis (EDA)** using Python. By breaking down the process into well-structured modules, it ensures a systematic approach to data preparation, analysis, and modeling.

EDA is an essential step in any **data science pipeline**, and the tools included in this repository help in extracting meaningful insights before applying machine learning models.