Sri Sivasubramaniya Nadar College of Engineering, Chennai
(An autonomous Institution affiliated to Anna University)

Degree & Branch	B.E Computer Science & Engineering	Semester	VI
Subject Code & Name	UCS2612 & Machine Learning Algorithms Laboratory		
Academic year	2025-2026 Even	Batch:2023-2027	**Due date:**

Supporting document: Exploratory data analysis

Exploratory Data Analysis (EDA) refers to the process of visually and statistically summarizing and analyzing data to extract useful insights and patterns, detect outliers or missing values, and test assumptions that may influence the selection and performance of machine learning models.

Why is EDA Important in Machine Learning?

- Understanding Data Distribution – Know how data is spread (e.g., normal, skewed).
- Identifying Outliers & Noise – Helps clean the data for better model accuracy.
- Detecting Missing Values – Essential for data imputation or exclusion.
- Feature Selection – Reveal correlations and redundancies.
- Hypothesis Generation – Build intuition about relationships before modeling.
- Choosing the Right Model – Some models assume linearity or normality—EDA helps verify that.

1 Understanding Standardization and Normalization in Machine Learning

In machine learning and mathematical data processing, standardization and normalization are commonly used to scale features. This helps ensure that algorithms work efficiently and fairly treat each input feature.

1.1 Standardization (Z-score Scaling)

Standardization transforms data such that the resulting distribution has a mean of zero and a standard deviation of one. This is particularly useful when the data approximately follows a Gaussian distribution.

Mathematical Formula

Let x be a data point, μ be the mean, and σ be the standard deviation:

$$z = \frac{x - \mu}{\sigma}$$

This produces values with zero mean and unit variance.

When to Use Standardization

- When the data approximately follows a normal distribution.
- When using algorithms that assume standardized features, such as:
  - Linear regression
  - Logistic regression
  - Support Vector Machines (SVM)
  - Principal Component Analysis (PCA)
  - K-Means clustering
- When features have different units (e.g., cm vs kg).

Recommended Plots

- Histogram (before and after standardization)
- Boxplot (to visualize centrality and spread)
- Q-Q Plot (to check normality before standardization)

1.2 Normalization (Min-Max Scaling)

Normalization rescales the data to a fixed range, usually [0, 1] or [-1, 1]. It is useful when you want to bound the data values and preserve the shape of the original distribution.

Mathematical Formula

Let x be a data point, and $x_{\min}, x_{\max}$ be the minimum and maximum values in the data:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

This maps all data to the interval [0, 1].

When to Use Normalization

- When data does not follow a normal distribution.
- When features have different scales but need to be compared directly.
- In algorithms sensitive to magnitude and distance, such as:
  - K-Nearest Neighbors (KNN)
  - K-Means clustering
  - Neural networks (especially with sigmoid or tanh activation)
- When pixel values or bounded ranges are involved (e.g., image data).

Recommended Plots

- Histogram (to observe bounding effect)
- Line plot (to observe scaling transformation)
- Scatter plot (to visualize changes across features)

1.3 Summary of Use Cases

- Use **standardization** when you assume data to be normally distributed and when the algorithm relies on mean and variance.
- Use **normalization** when you want features on the same scale, especially for models using distance or bounded input functions.

2 Identifying Outliers and Noise in Data

Outliers are data points that significantly deviate from the rest of the observations, while noise refers to random variability in data that can obscure patterns. Detecting and handling these is essential for improving model robustness and accuracy.

2.1 Mathematical Techniques for Outlier Detection

1. Z-score Method (Standard Score)

Measures how far a data point is from the mean in terms of standard deviations.

$$z = \frac{x - \mu}{\sigma}$$

- If $|z| > 3$, the data point is typically considered an outlier.

2. Interquartile Range (IQR) Method

Uses the spread between the first and third quartiles.

$$\text{IQR} = Q_3 - Q_1$$

$$\text{Outlier if } x < Q_1 - 1.5 \cdot \text{IQR} \text{ or } x > Q_3 + 1.5 \cdot \text{IQR}$$

3. Mahalanobis Distance

Useful for identifying multivariate outliers by considering the correlation between features.

$$D_M(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}$$

4. Local Outlier Factor (LOF)

Measures the local deviation of a data point with respect to its neighbors. Higher LOF indicates higher outlier probability.

2.2 When to Detect and Handle Outliers

- When the data contains measurement or recording errors.
- Before applying distance-based models (KNN, K-Means).
- In regression models where outliers can skew predictions.
- When reducing variance and improving generalization.

2.3 Recommended Plots for Visual Outlier Detection

- **Box Plot**: Visualizes outliers as individual points beyond the whiskers.
- **Histogram**: Can reveal gaps or extreme values.
- **Scatter Plot**: Useful in bivariate analysis to observe isolated points.
- **QQ Plot**: Identifies deviation from normality, especially in the tails.
- **Pair Plot (Multivariate)**: Helps spot outliers across feature combinations.

2.4 Noise Detection and Filtering

Noise refers to irrelevant or random variations in data that obscure the true underlying pattern.

- **Smoothing techniques**: Moving averages, Gaussian filters.
- **Clustering**: Small, isolated clusters may indicate noise.
- **Dimensionality reduction (e.g., PCA)**: Can reduce noise by removing low-variance components.

2.5 Summary of Use Cases

- Apply Z-score or IQR method for simple numerical outlier detection.
- Use Mahalanobis distance or LOF for multivariate and density-based detection.
- Always visualize your data before and after removing outliers to assess impact.

3 Detecting Missing Values – Essential for Data Imputation or Exclusion

Missing values are common in real-world datasets and can arise due to various reasons such as data entry errors, sensor failures, or non-responses in surveys. Detecting and handling missing values is a critical preprocessing step that influences model performance and bias.

3.1 Types of Missing Data

- **MCAR (Missing Completely At Random)**: The probability of a missing value is unrelated to any variable.
- **MAR (Missing At Random)**: The missingness is related to observed variables but not the missing one.
- **MNAR (Missing Not At Random)**: The missingness is related to the unobserved value itself.

3.2 Mathematical Notation and Representation

Let $X \in \mathbb{R}^{n \times d}$ be the dataset, where x_{ij} represents the value of the j^{th} feature in the i^{th} sample. Define an indicator matrix M as:

$$M_{ij} = \begin{cases} 1 & \text{if } x_{ij} \text{ is missing} \\ 0 & \text{otherwise} \end{cases}$$

The total missing rate for a feature j is:

$$\text{Missing Rate}_j = \frac{1}{n} \sum_{i=1}^{n} M_{ij}$$

3.3 Strategies for Handling Missing Values

- **Deletion Methods**:
  - **Listwise deletion**: Remove rows with any missing value.
  - **Column deletion**: Remove columns with high missing rate.
- **Imputation Methods**:
  - **Mean/Median/Mode Imputation**: Replace missing values with the mean, median, or mode of the column.
  - **K-Nearest Neighbors (KNN)**: Impute based on the values of nearest neighbors.
  - **Regression Imputation**: Predict missing values using regression models based on other variables.
  - **Multiple Imputation**: Generate multiple estimates to reflect uncertainty in imputed values.
- **Model-based Approaches**:
  - Algorithms like Decision Trees or XGBoost can inherently handle missing values.

3.4 Recommended Plots for Visualizing Missing Data

- **Heatmap**: Displays presence of missing values using binary color coding.
- **Bar plot**: Shows the count or percentage of missing values per column.
- **Matrix Plot**: Highlights missingness patterns across multiple variables.
- **Dendrogram or Clustering Maps**: Shows similarity in missing patterns across samples.

3.5 Summary of Use Cases

- Use deletion only when the proportion of missing values is small.
- Use simple imputation (mean/median) for numerical features with low missingness.
- Use advanced imputation (KNN, regression) for structured or correlated data.
- Always analyze the pattern of missingness before choosing a strategy.

4 Feature Selection – Reveal Correlations and Redundancies

Feature selection is the process of identifying and retaining the most relevant features from a dataset while removing those that are redundant or irrelevant. It helps improve model performance, reduce overfitting, and increase interpretability.

4.1 Mathematical Foundation

Let $X \in \mathbb{R}^{n \times d}$ be the dataset with n samples and d features, and let $y \in \mathbb{R}^n$ be the target variable. The goal is to select a subset $S \subseteq \{1, 2, \ldots, d\}$ such that the features X_S maximize predictive relevance and minimize redundancy.

1. Correlation Coefficient

To detect linear dependency between two features x_i and x_j, use the Pearson correlation coefficient:

$$\rho_{ij} = \frac{\text{Cov}(x_i, x_j)}{\sigma_{x_i}\sigma_{x_j}}$$

High absolute values (close to 1 or -1) suggest redundancy.

2. Mutual Information

For nonlinear dependencies, mutual information $I(x; y)$ measures the amount of information shared between a feature and the target:

$$I(x; y) = \sum_{x,y} p(x, y) \log\left(\frac{p(x, y)}{p(x)p(y)}\right)$$

Higher values indicate more informative features.

3. Variance Thresholding

Remove features with low variance, as they provide little discriminative power:

$$\text{Var}(x_j) < \theta \quad \Rightarrow \quad \text{Drop } x_j$$

4.2 Feature Selection Methods

- **Filter Methods**: Use statistical scores (e.g., correlation, chi-square, mutual information).
- **Wrapper Methods**: Use model performance (e.g., recursive feature elimination).
- **Embedded Methods**: Perform selection during training (e.g., Lasso, tree-based feature importance).

4.3 Recommended Plots for Visualizing Correlations and Redundancy

- **Correlation Heatmap**: Visualizes pairwise correlation between features.
- **Pair Plot**: Helps detect linear and nonlinear relationships.
- **Feature Importance Bar Plot**: Provided by models like random forest or XGBoost.
- **Variance Plot**: Highlights low-variance features.

4.4 Summary of Use Cases

- Remove highly correlated features to reduce multicollinearity.
- Use mutual information for detecting nonlinear associations.
- Apply feature selection before high-complexity models (e.g., SVM, neural networks) to avoid overfitting.
- Embedded methods like Lasso or tree models can automate feature selection.

5 Hypothesis Generation – Build Intuition About Relationships Before Modeling

Hypothesis generation in exploratory data analysis refers to the process of forming educated assumptions or ideas about the underlying structure, trends, and relationships within the dataset. These hypotheses guide the selection of models, features, and validation strategies.

5.1 Mathematical Perspective

Let $X = \{x_1, x_2, \dots, x_d\}$ be the set of input features and y be the target variable. A hypothesis H is an assumed relationship such as:

$$H_0 : \text{There is no relationship between } x_i \text{ and } y$$

$$H_1 : \text{There exists a relationship between } x_i \text{ and } y$$

Hypothesis generation aims to explore such relationships **before** formal statistical testing or modeling.

5.2 Techniques to Support Hypothesis Generation

- **Visual Analysis**:
  - Scatter plots to observe trends between variables.
  - Histograms and box plots to explore distributional differences.
  - Pair plots for multivariate interactions.
- **Statistical Summary**:
  - Mean and median comparisons across classes or categories.
  - Variance analysis to detect spread across subgroups.
- **Correlation Analysis**:
  - Pearson or Spearman correlation coefficients to detect linear or monotonic relationships.
- **Group-wise Aggregation**:
  - Comparing feature statistics for different class labels or conditions.

5.3 Recommended Plots for Hypothesis Generation

- **Scatter Plots**: Identify trends, clusters, or gaps.
- **Pair Plots**: Understand pairwise feature relationships.
- **Grouped Box Plots**: Compare distributions across classes.
- **Bar Charts**: Visualize categorical differences.
- **Violin Plots**: Combine distribution shape and central tendency.

5.4 Summary of Use Cases

- Hypothesis generation is an informal, visual-first step to guide deeper analysis.
- It helps prioritize which features or interactions to focus on during model development.
- Generated hypotheses can later be tested using formal statistical tests or predictive modeling.

6 Choosing the Right Model – EDA Helps Verify Model Assumptions

One of the key purposes of exploratory data analysis (EDA) is to assess whether the assumptions required by certain machine learning models hold true. This ensures better model selection, improved performance, and more valid inferences.

6.1 Mathematical Basis of Model Assumptions

Many models are based on specific mathematical assumptions about the data distribution or relationships. EDA allows us to visually and statistically evaluate whether these assumptions are satisfied.

1. Linearity Assumption

Linear models assume a linear relationship between independent variables $X = \{x_1, x_2, \ldots, x_d\}$ and the target variable y:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_d x_d + \epsilon$$

EDA helps check this using:

- Scatter plots (to observe linear trends).
- Residual plots (to check randomness of errors).

2. Normality Assumption

Some models assume that the features or residuals follow a normal distribution, particularly:

- Linear regression
- Linear Discriminant Analysis (LDA)
- Statistical tests like t-tests, ANOVA

Check this using:

- Histograms and KDE plots
- Q-Q (Quantile-Quantile) plots
- Shapiro-Wilk or D'Agostino test for normality

3. Homoscedasticity (Equal Variance)

For linear models, the variance of errors should be constant:

$$\text{Var}(\epsilon_i) = \sigma^2 \quad \forall i$$

Evaluate this using:

- Residual vs fitted value plots

4. Independence of Observations

Some algorithms assume that observations are independent and identically distributed (i.i.d). For time-series data, EDA helps verify autocorrelation using:

- Lag plots
- Autocorrelation function (ACF) plots

6.2 EDA-Guided Model Choices

- Use **linear models** (e.g., Linear Regression, LDA) when linearity and normality are satisfied.
- Use **nonlinear models** (e.g., Decision Trees, Random Forests) when data is highly skewed or interactions are complex.
- Use **distance-based models** (e.g., KNN, SVM with RBF kernel) when features are normalized and the geometry of data matters.
- Use **probabilistic models** (e.g., Naive Bayes) when feature independence is reasonable.

6.3 Recommended Plots for Model Selection via EDA

- **Scatter plots**: To detect linearity or curvature.
- **Residual plots**: To assess homoscedasticity and linearity.
- **Q-Q plots and Histograms**: To check normality.
- **Box plots**: To identify skewness and outliers.
- **Correlation matrix**: To assess multicollinearity.

6.4 Summary of Use Cases

- EDA helps ensure that model assumptions match data characteristics.
- Choosing a model without verifying assumptions may lead to biased or invalid results.
- Use EDA results to either select models or justify preprocessing techniques (e.g., transformations).

7 Commonly Used Plots in Exploratory Data Analysis

When performing Exploratory Data Analysis (EDA) on a dataset, visualizations play a crucial role in uncovering data characteristics, relationships, and anomalies. Below is a curated list of commonly used plots, categorized by their primary purpose.

1. Univariate Plots (Single Feature Analysis)

- **Histogram** – Displays frequency distribution of a numerical variable.
- **Box Plot** – Shows median, quartiles, and outliers.
- **Violin Plot** – Combines KDE with box plot to display distribution and density.
- **Bar Plot** – Used for categorical variables to show counts or proportions.
- **KDE Plot (Density Plot)** – Estimates the probability density function of a variable.

2. **Bivariate and Multivariate Plots**
   - **Scatter Plot** – Visualizes the relationship between two continuous variables.
   - **Pair Plot** – Matrix of scatter plots showing all pairwise relationships.
   - **Grouped Box Plot** – Compares distributions across different categories.
   - **Line Plot** – Useful for visualizing trends in time-series data.

3. **Correlation and Redundancy**
   - **Heatmap** – Displays correlation matrix using color gradients.
   - **Cluster Map** – Heatmap with hierarchical clustering applied to rows and columns.

4. **Outlier and Noise Detection**
   - **Box Plot** – Identifies outliers beyond the whiskers.
   - **Scatter Plot** – Highlights isolated points away from data clouds.
   - **Residual Plot** – Detects non-random patterns or heteroscedasticity.

5. **Missing Data Visualization**
   - **Missing Value Heatmap** – Shows presence or absence of data in the matrix.
   - **Bar Plot of Missingness** – Displays number or percentage of missing values per column.
   - **Matrix Plot** – Highlights missingness structure across rows and columns.

6. **Normality and Distribution Assumptions**
   - **Q-Q Plot (Quantile-Quantile Plot)** – Compares data distribution against a normal distribution.
   - **Histogram with Normal Curve Overlay** – Assesses how well the data fits a normal distribution.

7. **High-Dimensional or Categorical Data**
   - **Parallel Coordinates Plot** – Compares multivariate samples across dimensions.
   - **Count Plot** – Visualizes frequency of categories.
   - **Stacked Bar Plot** – Shows multiple categorical groupings.

Summary

These plots provide a foundational toolkit for visually exploring any dataset. Their appropriate selection depends on the data type (categorical or numerical), number of features, and the analysis goal (distribution, relationships, patterns, or anomalies).