Day22_EDA_and_ML_Theory_Introduction_Part_1

June 11, 2025

Exploratory Data Analysis (EDA) and Introduction to Machine Learning

1 What is EDA?

Exploratory Data Analysis (EDA) is the process of investigating datasets to discover patterns, spot anomalies, test hypotheses, and check assumptions using statistical summaries and visualizations.

1.1 Why is EDA Important?

- Understand the structure and meaning of data
- Detect errors, missing values, and outliers
- Highlight important relationships between variables
- Prepare the dataset for machine learning (ML) modeling " " "

2 Data Systems: Local vs Distributed

1. Local System (Database on Laptop)

- Runs on a single machine
- Limited storage and compute
- Suitable for small or medium data
- Example: SQLite, MySQL running on your laptop

2. Distributed System (Hadoop, Spark)

- Runs on multiple machines
- Can store and process big data
- Examples: Hadoop Distributed File System (HDFS), Apache Spark, Hive

3 Real-World Example: Local Shops & Central Database

Imagine: - 3 local grocery stores with individual databases - Each store performs ETL (Extract, Transform, Load) - All data is sent to a central master database or data warehouse - After cleaning and transformation, a final combined dataset is prepared

4 EDA Techniques – A Deep Dive into Feature Engineering

List of 7 Core EDA / Feature Engineering Techniques

1. Variable Identification
2. Univariate Analysis
3. Bivariate Analysis
4. Outlier Detection
5. Missing Value Treatment
6. Variable Transformation
7. Variable Creation

4.1 Variable Identification

- Classify features into:
  - Independent Variables (X): used to predict
  - Dependent Variable (Y): the target/output
- Types of Variables:
  - Categorical: Gender, City
  - Numerical: Age, Salary
  - Date/Time: Timestamp, DOB

Family Example:

Family has 4 members: - Dad (earns money) - Mom (housewife) - Son (student) - Daughter (student)

Only Dad earns → he is the dependent variable (Y) Others are independent variables: Mom (X1), Son (X2), Daughter (X3)

So, in ML form: Y = X1 + X2 + X3 (like Multiple Linear Regression)

4.2 Univariate Analysis

Analyzing one variable at a time.

- Categorical: use bar charts, value_counts()
- Numerical: use histograms, boxplots, describe()

4.3 Bivariate Analysis

Studying the relationship between two variables.

- Categorical vs Categorical: Stacked bar plot, crosstab
- Numerical vs Numerical: Scatter plot, correlation
- Categorical vs Numerical: Boxplot

4.4 Outlier Detection

Outliers are values far from the rest. - Boxplot (outside whiskers) - Z-score - IQR method

Example using IQR:

Q1 = df[‘Age’].quantile(0.25)

Q3 = df[‘Age’].quantile(0.75)

IQR = Q3 - Q1

outliers = df[(df[‘Age’] < Q1 - 1.5*IQR) | (df[‘Age’] > Q3 + 1.5*IQR)]

4.5 Missing Value Treatment

Ways to handle nulls: - Delete rows/columns - Impute (mean, median, mode) - Forward/Backward fill

Example: df['Age'].fillna(df['Age'].mean(), inplace=True)

4.6 Variable Transformation

Changing scale or format: - Scaling: MinMaxScaler, StandardScaler - Encoding: LabelEncoding, OneHotEncoding

Example: from sklearn.preprocessing import MinMaxScaler scaler = MinMaxScaler() df['Age_scaled'] = scaler.fit_transform(df[['Age']])

4.7 Variable Creation

Creating new features from existing ones: - Combine features - Extract info from date - BMI = weight / height^2

Example: df['BMI'] = df['Weight_kg'] / (df['Height_m'] ** 2)

4.8 Summary Table

Step	Technique Name	Purpose
1	Variable Identification	Define roles of features (X vs Y)
2	Univariate Analysis	Understand individual variable distributions
3	Bivariate Analysis	Analyze relationships between variables
4	Outlier Detection	Detect extreme values
5	Missing Value Treatment	Handle null values
6	Variable Transformation	Rescale/encode features
7	Variable Creation	Create informative new features

Introduction to Machine Learning (ML)

Why Learn ML After EDA?

- EDA helps us understand and prepare data.
- Without clean and understood data, ML models won't work well.
- EDA decides how data is transformed, which features are used, and helps choose the right ML model.

5 ML Categories

5.1 Regression

- **Used when the dependent variable is continuous** (e.g., price, temperature)
- Examples: Gold price, petrol price, house price, stock price, weather, crypto, etc.

We use **regression models** in such cases.

5.1.1 Regression Algorithms:

1. Simple Linear Regression
2. Multiple Linear Regression
3. Polynomial Regression
4. Gradient Descent
5. Stochastic Gradient Descent
6. Batch Gradient Descent
7. Lasso Regularization (L1)
8. Ridge Regularization (L2)
9. Elastic Net (L1 + L2)
10. K-Nearest Neighbor Regression (KNN)
11. Decision Tree Regression
12. Random Forest Regression
13. ANN Regression
14. Time Series Analysis
15. XGBoost Regression
16. LGBM Regressor
17. Support Vector Regressor (SVR)

5.2 Classification

- **Used when the dependent variable is categorical or binary**
- Examples: Win/Loss, Pass/Fail, Spam/Not Spam, Rain/No Rain, Yes/No

We use **classification models** in such cases.

5.2.1 Classification Algorithms (Only Names):

1. Logistic Regression
2. K-Nearest Neighbors (KNN)
3. Decision Tree Classifier
4. Random Forest Classifier
5. Naive Bayes
6. Support Vector Machine (SVM)
7. Stochastic Gradient Descent Classifier
8. Gradient Boosting Classifier
9. XGBoost Classifier
10. LGBM Classifier
11. AdaBoost
12. Extra Trees Classifier
13. ANN Classifier
14. CNN (for images)
15. RNN (for sequences)
16. CatBoost
17. Voting Classifier

5.3 Clustering

- **No dependent variable** (unsupervised learning)

- Use case: Grouping customers, documents, patterns, etc.

5.3.1 Clustering Algorithms (Only Names):

1. K-Means
2. DBSCAN
3. Agglomerative Clustering
4. Hierarchical Clustering
5. Mean Shift
6. OPTICS
7. Gaussian Mixture Model (GMM)

Important Tip:

> **Choosing the right dependent variable (Y) is CRUCIAL**

- It decides whether your problem is a regression or classification task
- Example:
  - Data: `x1 = name`, `x2 = soft`, `x3 = new/old`, `x4 = hospital`, `x5 = purchased`, `y = price`
  - If **y = price** → Regression
  - If **y = purchased (yes/no)** → Classification

Attribute Relevance - Two types: - **Relevant Attributes**: Improve model quality - **Irrelevant Attributes**: Cause noise, overfitting, multicollinearity

To build a strong ML model: - Use only relevant variables - Remove noise to reduce overfitting & errors

[ ]:

```

```