kf;

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY

School of Engineering and Technology

Academic Year: 2025-2026 **Sem:** V

Course: Foundations of Data Science Lab **Course code**: UBTML329

Lab Manual Prepared By,

Mr. Swapnil D. Magar

kf;

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Assignment No. 1

Title:

Loading Data from CSV, JSON, Excel, and Online APIs using Pandas and Exploring Dataset

Objective:

The main objective of this lab is to learn how to load datasets from multiple sources and formats into Python using the Pandas library. Students will also understand how to inspect dataset dimensions, identify data types, and generate descriptive summaries of the dataset. This will provide the foundational skills required for further data preprocessing and analysis.

Software Requirements:

1. Python 3.8 or above
2. Jupyter Notebook / JupyterLab or VS Code / PyCharm
3. pandas
4. numpy

Outcome:

After completing this lab, students will be able to:

1. Import and use the Pandas library for data handling.
2. Load datasets from CSV, JSON, and Excel files, as well as from online APIs.
3. Explore dataset dimensions (rows and columns).
4. Analyze data types of various attributes in the dataset.
5. Use Pandas functions such as info() and describe() to summarize data.
6. Understand the importance of dataset exploration as the first step in Data Science workflows.

Aim:

To load datasets from CSV, JSON, Excel, and online APIs into Python using Pandas and to explore dataset dimensions, data types, and summary statistics.

kf;

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Theory:

Data is the backbone of Data Science. In real-world applications, datasets come in various formats such as **CSV (Comma Separated Values)**, **JSON (JavaScript Object Notation)**, **Excel spreadsheets**, or data served through **APIs (Application Programming Interfaces)**. Efficient handling of these formats is necessary for smooth data preprocessing and analysis.

The **Pandas** library in Python provides powerful tools to load and manipulate data:

- **CSV files:** Most common file format in data science. Loaded using pd.read_csv().
- **JSON files:** A semi-structured data format widely used in web applications. Loaded using pd.read_json().
- **Excel files:** Popular for tabular data storage. Loaded using pd.read_excel().
- **APIs:** Online services often provide data through APIs in JSON format. This can be loaded using Pandas in combination with the requests library.

After loading the data, we must perform **data exploration** to understand its structure and quality. Important exploration steps include:

- **df.shape** → Gives dataset dimensions (rows × columns).
- **df.dtypes** → Displays data types of each column.
- **df.info()** → Provides concise summary including memory usage.
- **df.describe()** → Generates descriptive statistics for numerical data such as mean, median, min, max, and quartiles.

Without these steps, it is difficult to understand how to preprocess the dataset or apply further algorithms.

kf;

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Algorithm / Steps:

1. Import the Pandas library (import pandas as pd).
2. Load datasets from different sources:
    - Load a CSV file using pd.read_csv('filename.csv').
    - Load a JSON file using pd.read_json('filename.json').
    - Load an Excel file using pd.read_excel('filename.xlsx').
    - Load data from an online API by fetching with the requests library and converting to a Pandas DataFrame.
3. Display the first few rows of the dataset using df.head().
4. Check dataset dimensions using df.shape.
5. Explore data types of each column using df.dtypes.
6. Get dataset summary using df.info().
7. Generate statistical measures using df.describe().

Conclusion:

This experiment demonstrates how Pandas simplifies the process of loading datasets from multiple file formats and online sources. By using functions such as head(), shape, dtypes, info(), and describe(), we can quickly understand the structure, size, and nature of the dataset. Such exploration is crucial in the early stages of Data Science to identify missing values, detect data type inconsistencies, and prepare the dataset for further analysis.

Questions:

1. What are the differences between CSV, JSON, and Excel file formats? Give real-life examples of their use.
2. How does Pandas read_csv() function differ from read_json() and read_excel()?
3. Why is it necessary to check the dimensions and data types of a dataset before analysis?
4. What is the purpose of the describe() function in Pandas, and what kind of insights does it provide?
5. Explain the steps required to fetch and load data from an online API into a Pandas Data Frame.

kf;

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Assignment No. 2

Title:

Handling Missing Data, Removing Duplicates, Fixing Data Types, and Identifying & Treating Outliers

Objective:

The main objective of this assignment is to understand the importance of **data cleaning** in Data Science. Students will learn how to manage missing data, remove duplicate records, correct inconsistent data types, and detect as well as treat outliers to improve data quality and reliability for further analysis.

Software Requirements:

1. Python 3.8 or above
2. Jupyter Notebook / JupyterLab or VS Code / PyCharm
3. pandas
4. numpy

Outcome:

After completing this lab, students will be able to:

1. Understand the causes and consequences of missing data in datasets.
2. Apply statistical methods such as mean, median, and mode for handling missing values.
3. Recognize and remove duplicate records to ensure data integrity.
4. Fix inconsistent data types to ensure proper computations and analysis.
5. Identify outliers in data and apply different methods to treat them.
6. Appreciate the role of data cleaning as a crucial step in the data science pipeline.

Aim:

To clean and preprocess datasets by handling missing data, removing duplicates, fixing data types, and identifying and treating outliers to prepare data for accurate and meaningful analysis.

kf;

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Theory:

Data collected from real-world sources is rarely clean and ready for direct analysis. It often contains **missing values, duplicate entries, inconsistent data types, and extreme outliers**, all of which can negatively impact the results of data analysis or machine learning models. Effective data cleaning ensures that the dataset is consistent, complete, and accurate.

Duplicate records can occur due to repeated data entry, merging multiple datasets, or errors during data collection. These duplicates can bias results and must be removed. The process involves identifying identical rows and keeping only unique records to maintain dataset integrity.

3. Fixing Data Types

Sometimes data is stored in incorrect formats. For example:

- Numerical values stored as strings (e.g., "1000" instead of 1000).
- Dates stored as plain text instead of datetime objects.
- Categorical data stored as integers instead of categorical variables.

Fixing data types ensures correct operations, accurate computations, and efficient memory usage. Correct data types also allow libraries and tools to apply suitable methods for analysis.

4. Identifying and Treating Outliers

Outliers are values that are significantly different from the rest of the dataset. They may result from data entry errors, equipment malfunction, or genuine extreme observations. Outliers can distort statistical measures such as mean and standard deviation.

Methods to detect and handle outliers:

- **Visualization-based detection:** Using boxplots, scatter plots, and histograms to spot unusual values.
- **Statistical methods:** Identifying values beyond 1.5 times the interquartile range (IQR) or beyond ±3 standard deviations.

kf;

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

- **Treatment methods:**
  - Removing outliers if they are due to error.
  - Capping values at acceptable thresholds.

By handling missing values, duplicates, incorrect data types, and outliers, we ensure the dataset is accurate, consistent, and ready for meaningful analysis.

Algorithm / Steps:

1. Inspect the dataset for missing values, duplicates, data type inconsistencies, and potential outliers.
2. Handle missing values using mean, median, mode imputation, or deletion if appropriate.
3. Identify duplicate records and remove them to retain unique entries.
4. Check for incorrect data types and convert them into proper formats (e.g., string to numeric, text to datetime).
5. Detect outliers using visualization methods or statistical techniques.
6. Treat outliers by either removing, capping, or transforming values depending on context.
7. Verify the cleaned dataset for accuracy and consistency.

Conclusion:

This assignment highlights the significance of **data cleaning** in the Data Science lifecycle. Handling missing data prevents biased or inaccurate results, removing duplicates ensures dataset reliability, fixing data types allows for proper analysis, and treating outliers reduces distortions in statistical measures. Clean data forms the foundation for accurate data analysis, effective machine learning models, and reliable decision-making processes.

Questions

1. What are the common causes of missing data in real-world datasets? How can they affect analysis?
2. Differentiate between mean, median, and mode imputation. In what situations is each method most suitable?
3. Why is it important to remove duplicate records from datasets? Give an example.
4. Explain with examples how incorrect data types can affect analysis.
5. What are outliers? Discuss different methods to detect and treat them.

kf;

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Assignment No. 3

Title:

Normalization, Standardization, Encoding of Categorical Features, and Basic Feature Scaling

Objective:

The objective of this assignment is to understand and apply fundamental **feature engineering** techniques, including normalization, standardization, encoding categorical variables, and feature scaling. These methods prepare raw data for efficient and accurate use in machine learning models.

Software Requirements:

1. Python 3.8 or above
2. Jupyter Notebook / JupyterLab or VS Code / PyCharm

Outcome:

After completing this lab, students will be able to:

1. Explain the difference between normalization and standardization.
2. Apply normalization to bring data values within a defined range.
3. Use standardization to rescale features with zero mean and unit variance.
4. Encode categorical features into numerical representations suitable for models.
5. Perform basic feature scaling techniques to ensure comparability of variables.
6. Appreciate the importance of preprocessing in building robust machine learning models.

Aim:

To preprocess data by normalizing, standardizing, encoding categorical features, and applying feature scaling to improve data quality and model performance.

Theory:

Raw datasets often contain variables on different scales and types. Machine learning algorithms are sensitive to the magnitude and distribution of input data. For instance, distance-based algorithms like **K-Nearest Neighbors (KNN)** or gradient-based models like **Logistic Regression** and **Neural Networks** can perform poorly if features are not standardized. To overcome this, data preprocessing techniques such as normalization, standardization, encoding, and feature scaling are applied.

kf;

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

1. Normalization

Normalization is a technique used to rescale numerical data into a specific range, typically [0, 1]. This is useful when features have different units or scales.

- **Formula:**
- Xnorm=X−Xminf_0Xmaxf_0−Xminf_0X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} .
- **Example:** Converting salaries ranging from 20,000–200,000 into a range of 0–1.
- **Advantages:** Maintains relationships between values and is simple to apply.
- **When to use:** Useful in distance-based models like KNN and K-Means where feature magnitude directly influences similarity.

2. Standardization

Standardization transforms data to have a mean of 0 and a standard deviation of 1. Unlike normalization, it does not bound values within a fixed range but rescales them relative to distribution.

- **Formula:**
  Xstd=X−μσX_{std} = \frac{X - \mu}{\sigma}Xstd=σX−μ where μ\muμ = mean, σ\sigmaσ = standard deviation.
- **Advantages:** Useful when features have different units (e.g., weight in kg and height in cm).
- **When to use:** Essential for algorithms like Logistic Regression, Support Vector Machines (SVMs), and Neural Networks, which assume data is centered.

3. Encoding Categorical Features

Machine learning models generally require numerical input. Categorical variables (e.g., gender, color, city) must be converted into numbers using encoding techniques:

- **Label Encoding:** Assigns a unique integer to each category. Suitable for ordinal data (e.g., education levels: Primary=1, Secondary=2, Graduate=3).
- **One-Hot Encoding:** Creates binary columns for each category. Suitable for nominal data with no inherent order (e.g., colors: red, blue, green).

- **Ordinal Encoding:** Explicitly encodes ordered categories with a ranking.
- **Why encoding matters:** Without encoding, algorithms cannot interpret categorical data meaningfully.

4. Feature Scaling

Feature scaling ensures that features with larger magnitudes do not dominate features with smaller magnitudes. Scaling makes features comparable and helps models converge faster.

- **Types of Scaling:**
  - **Min-Max Scaling (Normalization)** → Scales features to [0, 1].
  - **Z-score Scaling (Standardization)** → Mean = 0, Standard deviation = 1.
  - **Robust Scaling:** Uses median and IQR, making it less sensitive to outliers.
- **Importance:** Algorithms like gradient descent converge faster and yield better results when features are scaled appropriately.

Algorithm / Steps:

1. Load the dataset and identify numerical and categorical features.
2. Apply normalization to rescale numerical data within a range (0–1).
3. Apply standardization to adjust features around mean 0 with unit variance.
4. Identify categorical features and apply appropriate encoding:
   - Use label encoding for ordinal variables.
   - Use one-hot encoding for nominal variables.
5. Scale features to ensure comparability and stability in models.
6. Verify preprocessing by checking transformed values and distributions.

kf;

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Conclusion:

Data preprocessing through normalization, standardization, encoding, and feature scaling is critical in preparing datasets for analysis. Normalization and standardization help align features with different units and scales, while encoding converts categorical variables into machine-readable formats. Feature scaling ensures that no feature dominates others due to differences in magnitude. Collectively, these techniques improve the efficiency of learning algorithms, enhance model accuracy, and ensure stable performance across diverse datasets.

Questions:

1. Differentiate between normalization and standardization. Provide examples of when each should be used.
2. Why do machine learning models require categorical features to be encoded into numerical values?
3. Explain the difference between label encoding and one-hot encoding. Which situations are they best suited for?
4. What is feature scaling, and why is it essential in algorithms like KNN and SVM?
5. How does robust scaling differ from standardization and normalization? When would you prefer robust scaling?

kf;

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Assignment No. 4

Title

Use NumPy/pandas to calculate mean, median, mode, variance, std dev, correlation, and covariance.

Objective

The objective of this assignment is to enable students to apply fundamental statistical concepts in Python using NumPy and pandas. By performing calculations of mean, median, mode, variance, standard deviation, correlation, and covariance, students will gain hands-on experience in analyzing datasets for meaningful insights.

Software Requirements:

1. Python 3.8 or above
2. Jupyter Notebook / JupyterLab or VS Code / PyCharm
3. pandas
4. numpy

Outcome

- Understand and compute descriptive statistics.
- Differentiate between measures of central tendency and measures of dispersion.
- Analyze relationships between two or more variables using correlation and covariance.
- Apply statistical operations on real datasets for data-driven decision making.

Aim

To calculate measures of central tendency, dispersion, and relationships using **NumPy** and **pandas**, and to understand their significance in data analysis.

Theory

1. Measures of Central Tendency

These are values that represent the center or average of a dataset.

- **Mean (Average):** The sum of all values divided by the total number of values. It is sensitive to extreme values (outliers).
- **Median:** The middle value when the dataset is ordered. It is resistant to outliers and skewed data.

- **Mode:** The most frequently occurring value in the dataset. Useful for categorical data.

Example: If exam scores are [40, 50, 50, 60, 70], then:

- Mean = 54
- Median = 50
- Mode = 50

2. Measures of Dispersion (Spread)

Dispersion tells us how spread out the data values are.

- **Variance:** The average of the squared deviations from the mean. A higher variance means more spread.
- **Standard Deviation (σ):** The square root of variance. It represents the average distance of values from the mean in the same units as the data.

Example: If two students scored [50, 50, 50] and [30, 50, 70], both have the same mean (50), but the second dataset has higher variance and standard deviation.

---

3. Relationship Between Variables

- **Correlation:** Measures the strength and direction of the linear relationship between two variables.
    - Range: -1 to +1
    - +1 → perfect positive correlation (as one increases, the other increases)
    - -1 → perfect negative correlation (as one increases, the other decreases)
    - 0 → no correlation
- **Covariance:** Measures how two variables change together. Unlike correlation, covariance is not standardized, so its magnitude depends on the scale of the variables.

Example: Height and weight generally have a positive correlation (taller people tend to weigh more).

kf;

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

4. Real-Life Applications

- **Business:** Correlation is used in stock markets to measure relationships between assets.
- **Education:** Mean and median scores are used to evaluate class performance.
- **Healthcare:** Variance helps in analyzing variations in patient health metrics.
- **Weather Forecasting:** Standard deviation is used to measure climate variability.

Algorithm

1. Load dataset into Python using pandas.
2. Use pandas/NumPy functions to calculate:
    - Mean, Median, Mode
    - Variance, Standard Deviation
    - Correlation, Covariance
3. Interpret the results in terms of data distribution and relationships.
4. Compare results across different columns/features of the dataset.

Conclusion

Statistical measures form the backbone of data analysis. Central tendency (mean, median, mode) summarizes the dataset, while dispersion (variance, standard deviation) describes the spread. Correlation and covariance help understand relationships between variables. These concepts allow data scientists to draw meaningful insights, detect anomalies, and make informed predictions.

Questions

1. What is the difference between variance and standard deviation?
2. Why is the median preferred over mean in skewed datasets?
3. How do covariance and correlation differ?
4. Give a real-life example where mode is more useful than mean.
5. What does a correlation coefficient of 0 imply?

kf;

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Assignment No 5

Title

Data Visualization: Histograms, Boxplots, Scatterplots, Heatmaps, and Pairplots

Objective

The objective of this assignment is to understand the importance of data visualization in data science and to learn how to represent datasets using Matplotlib and Seaborn. By creating different types of plots, students will be able to interpret the distribution, relationships, and patterns within data.

Software Requirements:

1. Python 3.8 or above
2. Jupyter Notebook / JupyterLab or VS Code / PyCharm
3. pandas
4. numpy
5. matplotlib
6. seaborn

Outcome

- Learn the role of visualization in data analysis and decision making.
- Explore different visualization techniques for univariate and multivariate data.
- Understand how to use histograms, boxplots, scatterplots, heatmaps, and pairplots effectively.
- Develop the ability to interpret visual patterns and relationships between variables.

kf;

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Aim

To apply visualization techniques in Python using Matplotlib and Seaborn and interpret the results to extract meaningful insights from data.

Theory

Introduction to Data Visualization

Data visualization is the graphical representation of information and data. It allows analysts to see patterns, trends, and outliers more easily compared to raw tabular data. In data science, visualization is a critical step because it:

- Helps in exploratory data analysis (EDA).
- Provides intuitive understanding of data distribution and relationships.
- Aids in decision-making by making complex data comprehensible.

Two widely used Python libraries for visualization are:

- Matplotlib: A low-level, flexible library for creating static, animated, and interactive plots.
- Seaborn: A higher-level library built on Matplotlib, providing beautiful and **informative statistical graphics with less code.**

1. Histograms

- Represents the frequency distribution of a continuous variable.
- Data is divided into bins (intervals), and the height of each bar represents the number of observations in that bin.
- Usage: To understand the distribution of a single variable, detect skewness, and identify outliers.

Example: Exam scores distribution showing how many students scored between ranges like 40–50, 50–60, etc.

kf;

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

2. Boxplots (Whisker Plots)

- Displays the summary statistics of a dataset: minimum, first quartile (Q1), median, third quartile (Q3), and maximum.
- The box represents the interquartile range (IQR), while whiskers represent variability outside the upper and lower quartiles.
- Outliers are shown as individual points beyond whiskers.
- Usage: To detect outliers and understand data spread.

Example: Salary distribution in a company where extreme high or low values can be visualized as outliers.

3. Scatterplots

- Shows the relationship between two continuous variables.
- Each point represents one observation, plotted with respect to its x and y values.
- Usage: To identify correlations, clusters, and outliers.

Example: Plotting height vs. weight of individuals shows whether taller people generally weigh more.

4. Heatmaps

- Represents data in a matrix form, where values are shown as colors.
- Frequently used with correlation matrices to represent how features are related to each other.
- Usage: To quickly identify strong positive or negative relationships.

Example: A heatmap of correlation between stock prices of different companies.

kf;

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

5. Pairplots

- A pairplot is a collection of scatterplots arranged in a grid, showing relationships between all possible pairs of variables in a dataset.
- Diagonal plots show the distribution (often histograms or density plots) of each variable.
- Usage: Ideal for exploring multivariate datasets.

Example: Analyzing the Iris dataset with a pairplot to see relationships among petal length, petal width, sepal length, and sepal width.

Importance of Visualization in Data Science

1. Simplifies Data Understanding: Converts raw numbers into understandable visuals.
2. Pattern Recognition: Detects trends and recurring behaviors.
3. Outlier Detection: Visuals make it easier to spot unusual data points.
4. Decision Support: Helps stakeholders and decision-makers understand complex data quickly.
5. EDA (Exploratory Data Analysis): Visualization is often the first step before applying machine learning.

Algorithm

1. Import dataset using pandas.
2. Choose appropriate visualization library (Matplotlib or Seaborn).
3. For univariate analysis, plot histograms and boxplots.
4. For bivariate analysis, use scatterplots.
5. For correlation analysis, create a heatmap.
6. For multivariate analysis, create a pairplot.
7. Interpret each visualization in terms of data distribution, outliers, and relationships.

kf;

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Conclusion

Data visualization plays a vital role in making sense of data. Histograms provide insights into distributions, boxplots highlight outliers, scatterplots reveal relationships, heatmaps show correlations, and pairplots explore multivariate data comprehensively. By mastering visualization with Matplotlib and Seaborn, students can move beyond raw numbers and create intuitive, meaningful stories from datasets.

Questions

1. What is the difference between a histogram and a boxplot?
2. How can scatterplots be used to identify correlations?
3. Why are heatmaps commonly used with correlation matrices?
4. What unique advantage do pairplots provide over scatterplots?
5. How does data visualization improve decision making in real-world scenarios?

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Assignment No. 6

Title

Exploratory Data Analysis (EDA) on Titanic/Iris Dataset Using Statistical and Visual Summaries

Objective

The objective of this assignment is to perform **Exploratory Data Analysis (EDA)** on a real-world dataset such as the **Titanic** dataset (passenger survival data) or the **Iris** dataset (flower classification data). Students will learn how to explore, clean, and understand data using both **statistical measures** and **visualizations**.

Software Requirements:

1. Python 3.8 or above
2. Jupyter Notebook / JupyterLab or VS Code / PyCharm
3. pandas
4. numpy

Outcome

- Gain hands-on experience in exploring real-world datasets.
- Learn how to summarize data statistically and visually.
- Develop skills to detect missing values, outliers, and trends.
- Understand the importance of EDA in data preprocessing and model building.

Aim

To perform comprehensive exploratory data analysis (EDA) on Titanic or Iris datasets using **statistical summaries** and **visual plots**, and interpret meaningful insights from the data.

Theory

What is EDA?

Exploratory Data Analysis (EDA) is the process of examining datasets to summarize their main characteristics, often with the help of statistical tools and visualizations. It helps in understanding the structure, quality, and patterns of data before applying machine learning or predictive modeling.

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Steps in EDA

1. **Understanding the Dataset**
   - Identify dataset source (Titanic/Iris).
   - Understand what each column (feature) represents.
2. **Data Cleaning**
   - Handle missing values.
   - Remove duplicates.
   - Fix data types.
3. **Statistical Summaries**
   - **Central Tendency:** Mean, Median, Mode.
   - **Dispersion:** Variance, Standard Deviation, Range, Interquartile Range.
   - **Relationships:** Correlation and Covariance between variables.
4. **Visual Summaries**
   - **Univariate Analysis:** Histograms, Boxplots (distribution of single features).
   - **Bivariate Analysis:** Scatterplots, Barplots (relationships between two variables).
   - **Multivariate Analysis:** Heatmaps, Pairplots (interactions across multiple features).
5. **Insight Generation**
   - Identify key trends, patterns, and anomalies.
   - Interpret the real-world meaning of results.

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Dataset Examples

Titanic Dataset

- **Description:** Contains data of passengers aboard the Titanic ship. Features include passenger class, age, gender, fare, and survival status.
- **Possible Insights:**
  - Survival rate by gender and class.
  - Age distribution of passengers.
  - Relationship between ticket fare and survival.

Iris Dataset

- **Description:** Famous dataset containing sepal and petal dimensions of three species of Iris flowers.
- **Possible Insights:**
  - Distribution of petal lengths and widths among species.
  - Correlation between sepal length and petal length.
  - Classification boundaries between species.

Algorithm

1. Load dataset (Titanic or Iris) into pandas DataFrame.
2. Perform **data cleaning**: check for missing values, duplicates, and incorrect data types.
3. Generate **statistical summaries** using pandas/NumPy.
4. Plot **histograms and boxplots** for univariate analysis.
5. Plot **scatterplots and barplots** for bivariate analysis.
6. Create **heatmaps and pairplots** for multivariate analysis.
7. Interpret results and highlight key insights.

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Conclusion

EDA is an essential step in the data science pipeline. By analyzing Titanic or Iris datasets, we gain insights into distributions, patterns, and relationships. Statistical summaries provide numerical evidence, while visualizations make patterns more intuitive. EDA not only improves understanding but also guides feature engineering, model selection, and better decision-making in real-world applications.

Questions

1. What is the role of EDA in the data science workflow?
2. How do statistical summaries and visual summaries complement each other?
3. Why is handling missing values important during EDA?
4. In the Titanic dataset, which factors most influence survival rates?
5. What advantage does pairplot offer when analyzing the Iris dataset?

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Assignment No 7

Title

Linear Regression: Model Building, Training, Testing, and Evaluation

Objective

The objective of this assignment is to understand how to build, train, and evaluate a **Linear Regression model** using Python. Students will learn the role of train-test split, generate predictions, and visualize predicted vs. actual values for performance evaluation.

Outcome

- Understand the concept of linear regression and its applications.
- Learn how to split data into training and testing sets.
- Build and evaluate a regression model using Python libraries.
- Interpret results using evaluation metrics and visualizations.

Software Requirements

- **Programming Language:** Python 3.8 or higher
- **Libraries:**
    - NumPy (for numerical computations)
    - pandas (for data manipulation)
    - scikit-learn (for regression modeling and evaluation)
    - matplotlib & seaborn (for visualization)
- **Environment:** Jupyter Notebook / Google Colab / Python IDE (PyCharm, VS Code, etc.)

Aim

To build a Linear Regression model using a real or synthetic dataset, evaluate its performance with train-test split, and plot predicted vs. actual values to assess accuracy.

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Theory

Introduction to Linear Regression

Linear Regression is a **supervised machine learning algorithm** used for predicting a continuous dependent variable based on one or more independent variables. It assumes a linear relationship between the variables.

- **Simple Linear Regression:** Involves one independent variable (predictor) and one dependent variable (target).
  Equation:
  $Y=b0+b1X+\epsilon$ Y = b_0 + b_1X + \epsilon
  where:
  - Y = Dependent variable
  - X = Independent variable
  - b_0 = Intercept
  - b_1 = Slope (coefficient)
- **Multiple Linear Regression:** Involves two or more independent variables.
  Equation:
  $Y=b0+b1X1+b2X2+...+bnXn+\epsilon$ Y = b_0 + b_1X_1 + b_2X_2 + ... + b_nX_n + \epsilon

Steps in Linear Regression Modeling

1. Load and preprocess dataset.
2. Define features (X) and target (Y).
3. Perform **train-test split** to avoid overfitting.
4. Train the Linear Regression model on training data.
5. Predict outcomes on test data.

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

1. Evaluate performance using metrics:
    - Mean Absolute Error (MAE)
    - Mean Squared Error (MSE)
    - Root Mean Squared Error (RMSE)
2. Visualize results by plotting **Predicted vs Actual values**.

Importance of Train-Test Split

- Ensures that the model generalizes well to unseen data.
- Prevents overfitting (where the model memorizes training data but fails on new data).

Interpretation of Predicted vs Actual Plot

- Points close to the diagonal line indicate accurate predictions.
- Large deviations suggest poor model performance.

Algorithm

1. Import necessary libraries.
2. Load dataset into pandas DataFrame.
3. Define independent variables (X) and dependent variable (Y).
4. Split dataset into training and testing sets (e.g., 80-20 split).
5. Initialize and fit the Linear Regression model.
6. Predict values using the test dataset.
7. Evaluate model using metrics (MAE, MSE, RMSE, R^2).
8. Plot **Predicted vs Actual** values.
9. Interpret the model's performance.

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Conclusion

Linear Regression is one of the simplest yet most powerful predictive modeling techniques. By applying train-test split, students can evaluate model performance and ensure generalization. Predicted vs. actual plots and error metrics provide meaningful insights into the accuracy and limitations of the model. Understanding these concepts builds a strong foundation for more advanced machine learning algorithms.

Questions

1. Differentiate between simple and multiple linear regression.
2. What is the role of the error term (ε) in the regression equation?
3. Why do we perform a train-test split in regression modeling?
4. Explain the significance of the R^2 score in regression evaluation.
5. How can a Predicted vs Actual plot help in interpreting model accuracy?

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Assignment No 8

Title:

Build and Evaluate a Logistic Regression or Decision Tree Model on a Classification Dataset

Objective

To implement and evaluate Logistic Regression and Decision Tree models on a real-world classification dataset (e.g., Iris dataset) using Python libraries such as *scikit-learn*, and to analyze the model performance using various evaluation metrics.

Outcome

After successful completion of this experiment, students will be able to:

1. Load and preprocess a real-world dataset for classification.
2. Build and train **Logistic Regression** and **Decision Tree models**.
3. Understand the difference between linear and non-linear decision boundaries.
4. Evaluate classification models using metrics like accuracy, precision, recall, and F1-score.
5. Visualize classification results and compare performance between models.
6. Develop the skill to choose an appropriate classification algorithm based on dataset characteristics.

Software Requirements

- Python 3.8+
- Anaconda / Jupyter Notebook or Google Colab
- Libraries:
    - NumPy
    - pandas
    - matplotlib / seaborn
    - scikit-learn

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Aim

To classify data using Logistic Regression and Decision Tree algorithms, evaluate their performance, and visualize results.

Theory

1. Logistic Regression

- Logistic Regression is a supervised classification algorithm used when the target variable is categorical.
- It predicts the probability of belonging to a particular class using the logistic (sigmoid) function:

P(Y=1|X)=11+e−(b0+b1X1+b2X2+...+bnXn)P(Y=1|X) = \frac{1}{1 + e^{-(b_0 + b_1X_1 + b_2X_2 + ... + b_nX_n)}}P(Y=1|X)=1+e−(b0+b1X1+b2X2+...+bnXn)1

- Key features:
  - Suitable for binary and multi-class classification.
  - Decision boundary is linear.
  - Probability-based predictions.

2. Decision Tree

- A tree-structured model where data is split into subsets based on feature values.
- Nodes represent attributes, edges represent decisions, and leaves represent outcomes.
- Uses metrics like:
  - Gini Index
  - Entropy & Information Gain
- Advantages:
  - Easy to interpret and visualize.
  - Handles both numerical and categorical data.
- Limitations:
  - Can overfit the data → needs pruning.

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Applications of Logistic Regression & Decision Tree

- Logistic Regression: Spam detection, disease prediction, credit risk scoring.
- Decision Tree: Customer segmentation, fraud detection, recommendation systems.

Algorithm

Steps for Implementation

1. Import required libraries.
2. Load the classification dataset (e.g., Iris).
3. Preprocess the data (handle missing values, encode categorical features if any).
4. Split dataset into training and testing sets.
5. Build and train:
    - Logistic Regression model.
    - Decision Tree model.
6. Make predictions on the test data.
7. Evaluate models using:
    - Accuracy Score
    - Confusion Matrix
    - Classification Report (Precision, Recall, F1-score)
8. Compare Logistic Regression vs Decision Tree performance.
9. Visualize Decision Tree and classification boundaries (optional).

Conclusion

In this assignment, we implemented and compared Logistic Regression and Decision Tree models for classification tasks. Logistic Regression provided a linear decision boundary, while Decision Tree handled non-linear relationships. The choice of model depends on dataset complexity and interpretability requirements.

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329

Name: **Roll No:**

Questions

1. What is the difference between Logistic Regression and Linear Regression?
2. How does the sigmoid function help in classification?
3. What are the advantages and disadvantages of Decision Trees?
4. Explain Information Gain and Gini Index.
5. Which scenarios are more suitable for Decision Trees vs Logistic Regression?

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**

Course: Foundations of Data Science Lab **Course code**: UBTML329
Name: **Roll No:**

Assignment No 9

Title: Apply K-Means Clustering on Sample Data and Evaluate with Silhouette Score

Objective

To implement K-Means clustering on a sample dataset, visualize the clusters, and evaluate the clustering performance using the silhouette score.

Software Requirements

- **Python 3.8+**
- **Anaconda / Jupyter Notebook** or **Google Colab**
- **Libraries:**
  - NumPy
  - pandas
  - matplotlib / seaborn
  - scikit-learn

Aim

To group data points into meaningful clusters using the **K-Means algorithm**, represent them visually, and assess clustering quality using the **silhouette score**.

Theory

1. Introduction to Clustering

- Clustering is an **unsupervised learning technique** that groups data points such that those within the same cluster are **more similar** to each other than to those in other clusters.
- Unlike classification, clustering does not rely on predefined labels.

2. K-Means Clustering

- **Definition:** K-Means aims to partition a dataset into **k distinct, non-overlapping clusters**. Each cluster is defined by its centroid (mean point).
- **Working Principle:**
    - Start with k random points as cluster centers (centroids).
    - Assign each data point to the nearest centroid based on **distance (commonly Euclidean distance)**.
    - Recalculate centroids as the mean of the assigned points.
    - Repeat until cluster assignments stop changing or a maximum number of iterations is reached.

Mathematical Representation:

argminC∑i=1k∑x∈Ci‖x−μi‖2\text{argmin}_C \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2argminCi=1∑kx∈Ci∑‖x−μi‖2

Where:

- CiC_iCi = cluster i
- μi\mu_iμi = centroid of cluster i
- ‖x−μi‖2\|x - \mu_i\|^2‖x−μi‖2 = squared distance between data point x and centroid

3. Distance Metrics in K-Means

- **Euclidean Distance:** Most common, measures straight-line distance.
- **Manhattan Distance:** Measures distance along axes, useful for high-dimensional data.
- **Cosine Similarity:** Measures angle between vectors, used for text data.
    - **Too many clusters** → overfitting (too detailed, meaningless).

4. Choosing the Number of Clusters (k)

- The choice of k is crucial:
  - **Too few clusters** → underfitting (not enough groups).
- **Methods to choose k:**
  - **Elbow Method:** Plot Within-Cluster Sum of Squares (WCSS) vs. number of clusters. The "elbow point" suggests optimal k.
  - **Silhouette Score:** Higher score = better-defined clusters.

5. Silhouette Score

- Measures how similar an object is to its own cluster compared to other clusters.
- Formula:

S=b−amax⁡(a,b)S = \frac{b - a}{\max(a, b)}S=max(a,b)b−a

Where:

- a = mean distance of a sample to points in the same cluster.
- b = mean distance of a sample to points in the nearest other cluster.

Interpretation:

- **+1** → Well-clustered
- **0** → Overlap between clusters
- **-1** → Misclassified

6. Strengths of K-Means

- Simple to understand and implement.
- Works well with large datasets.
- Fast and computationally efficient (linear time complexity).

7. Limitations of K-Means

- Requires pre-specifying **k**.
- Sensitive to **initialization of centroids**.
- Assumes clusters are **spherical and equally sized**.
- Struggles with outliers and noise.
- Not suitable for categorical features directly.

8. Applications of K-Means

- Business: Customer segmentation based on purchasing patterns.
- Healthcare: Patient grouping for treatment analysis.

Image Processing: Image compression by clustering pixel intensities

Algorithm

1. Import required libraries.
2. Load or generate dataset.
3. Preprocess the dataset (if required).
4. Select number of clusters (k).
5. Initialize centroids randomly.
6. Assign points to nearest centroid.
7. Recalculate centroids.
8. Repeat steps until centroids stabilize.
9. Plot the resulting clusters.
10. Evaluate clustering performance using **silhouette score**.

1. Apply techniques such as the **Elbow Method** and **Silhouette Score** to evaluate clustering quality.

PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year:2025-2026 **Sem: V**
Course: Foundations of Data Science Lab **Course code**: UBTML329
Name: **Roll No:**

Outcome

After completing this assignment, students will be able to:

1. Understand the concept of clustering and its difference from classification.
2. Implement **K-Means clustering** on real or synthetic datasets.
3. Visualize clusters effectively.
4. Recognize strengths and limitations of K-Means in real-world applications.

Conclusion

K-Means is a powerful unsupervised learning technique used to discover hidden structures in data. The silhouette score provides a quantitative measure of clustering quality. While K-Means is efficient, its performance heavily depends on the selection of k and sensitivity to outliers.

Questions

1. What is the difference between clustering and classification?
2. Why does K-Means require pre-defining k?
3. Explain the role of centroids in K-Means.
4. What are the advantages and disadvantages of K-Means clustering?
5. How do you evaluate clustering performance?

For **Assignment 10 (Case Study / Mini Project)**, we can structure it as follows:

- **Title & Domain Selection:** Choose a dataset based on the domain, for example:
  - Healthcare → Diabetes dataset
  - Sales → Retail dataset
  - Social Media → Twitter sentiment dataset
- **Objective & Aim**
- **Software Requirements**
- **Dataset Description**
- **Steps / Methodology:**

1. **Data Collection**
2. **Data Cleaning:** Handle missing values, duplicates, incorrect data types, etc.
3. **Exploratory Data Analysis (EDA):** Summary statistics and visualizations
4. **Feature Engineering:** Scaling, encoding, and feature selection
5. **Model Building:** Classification or regression depending on the dataset
6. **Evaluation:** Metrics such as accuracy, confusion matrix, RMSE, R^2, etc.
7. **Visualization of Results**

- **Conclusion & Findings**
- **Source Code (Python)**
- **Output Screenshots**

This format will naturally span around **10 pages** when detailed with explanations, code blocks, and visualizations.

Before finalizing, please verify the mini project topic or dataset and review the report with me before printing.