**Lab Guide:**

**Lab Exercise 1: Data Analysis**

Objective: Use Copilot to generate insights across different types of analytics.

Step 1: Descriptive Analytics – Understanding the Data

Prompt:

* Upload the dataset and list all columns with their data types.
* Identify missing values in the dataset.
* Show summary statistics (mean, median, min, max) for numerical columns.
* Count the number of approved (Loan\_Status=Y) and rejected (Loan\_Status=N) loans.
* Create a bar chart for the distribution of Property\_Area (Urban, Rural, Semiurban).

Step 2: Diagnostic Analytics – Identifying Patterns & Relationships

Prompt:

* Compare Applicant Income, Coapplicant Income, and Credit\_History for both groups.
* Is income a bigger factor than education in determining loan amount?
* Are self-employed applicants more likely to have a low Credit\_History score?
* Do self-employed applicants request **higher loan amounts** with lower declared incomes?
* Is there a strict **approval threshold** based on Credit\_History scores?
* Does income level compensate for poor credit history in approvals?
* Are there **exceptions** where people with poor credit history still get approved?
* Does the loan amount depend more on **property area** (urban vs. rural) rather than income?
* Are **loan term restrictions** (shorter terms for high-income applicants) affecting loan amounts?
* Do some high-income applicants have **co-applicants with low incomes**, reducing eligibility?
* Do married applicants apply with a co-applicant, increasing approval chances?
* Are married applicants more likely to have stable income sources?
* Do banks prefer married applicants due to lower default risks?

**Step 3: Predictive Analytics – Forecasting Loan Approval**

* Predict loan approval (Loan\_Status) based on income, credit history, and dependents.
* What factors most influence loan approval?
* Train a logistic regression model to predict Loan\_Status and show accuracy.
* Create a decision tree to classify whether a loan will be approved or not.
* Find the probability of approval for an applicant with ₹50,000 income, 2 dependents, and a credit score of 1.

**Step 4: Prescriptive Analytics – Recommending Actions**

* Recommend strategies to increase loan approval rates.
* What steps can applicants take to improve their approval chances?
* Suggest an automated rule to approve loans based on income and credit history.
* Identify the optimal loan amount for different income groups to minimize defaults.
* What changes in lending policy could improve loan acceptance rates by 10%?

**Lab Exercise 2: SQL Query Optimization**

**Objective:** Optimize SQL queries to improve performance in loan data analysis.

* **Write an SQL query** to retrieve only relevant columns (Loan\_ID, Loan\_Status, Applicant Income, Credit\_History) instead of selecting all.
* **Create an index in SQL** on Self\_Employed and Loan\_Status to speed up approval rate queries.
* **Use an SQL JOIN** to compare loan amounts for graduates and non-graduates instead of using a subquery.
* **Optimize an SQL query** to count approved and rejected loans without scanning the entire table.
* **Write an SQL query with indexing** to improve the performance of filtering loans based on Credit\_History.
* **Use SQL GROUP BY** to optimize the calculation of the average LoanAmount per Property\_Area.
* **Write an SQL query** to retrieve the top 10 highest loan amounts efficiently without using ORDER BY on the entire dataset.
* **Use an SQL index** to improve filtering for loan applications where Applicant Income is greater than ₹50,000.
* **Optimize an SQL GROUP BY query** for counting loans by Education category.
* **Improve SQL query performance** for filtering by Married and Coapplicant Income using indexing.

**Lab Exercise 3: Data Cleaning and Automation using Copilot**

**Objective:** Use **Copilot** to automate data cleaning tasks for the **loan dataset**.

**Step 1: Understanding and Preparing Data**

**Prompt:**

* Load the **loan dataset** and display the first five rows.
* Show column names and data types.
* Identify missing values in the dataset.

**Step 2: Handling Missing Values**

**Prompt:**

* Fill missing values in LoanAmount with the median loan amount.
* Replace missing values in Credit\_History with the most frequent value.
* Drop rows where Loan\_Status is missing.

**Step 3: Removing Duplicates**

**Prompt:**

* Check for duplicate Loan\_ID entries and remove them if any exist.

**Step 4: Transforming and Creating New Features**

**Prompt:**

* Create a new column Total Income as the sum of Applicant Income and Coapplicant Income.
* Convert Loan\_Status into numerical values (Y = 1, N = 0).
* Encode Property\_Area into categorical numerical values.

**Step 5: Standardizing and Scaling Data**

**Prompt:**

* Normalize Applicant Income and LoanAmount using Min-Max Scaling.
* Standardize Total Income using Z-score normalization.

**Step 6: Automating Data Cleaning**

**Prompt:**

* Write a function to perform all the above cleaning steps in one go.
* Save the cleaned dataset as a new CSV file.

**Step 7: Deployment**

**(Leave it to your choice)**