**Exploratory Data Analysis (EDA) Report – Loan Dataset**

Platform: Microsoft Azure Databricks

Language: PySpark (in Databricks notebooks)

### **1. Dataset Overview**

The dataset contains information about loan applicants, their demographic and financial details, and the status of their loan approval.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| Loan\_ID | String | Unique identifier for the loan |
| Gender | String | Applicant's gender |
| Married | String | Marital status |
| Dependents | String | Number of dependents (0, 1, 2, 3+) |
| Education | String | Graduate or Not Graduate |
| Self\_Employed | String | Employment status |
| ApplicantIncome | Integer | Income of the primary applicant |
| CoapplicantIncome | Double | Income of the co-applicant |
| LoanAmount | Integer | Loan amount requested (in thousands) |
| Loan\_Amount\_Term | Integer | Term of loan in months |
| Credit\_History | Integer | Credit history (1 = good, 0 = bad) |
| Property\_Area | String | Area where property is located (Urban/Rural/etc) |
| Loan\_Status | String | Target variable (Y = Approved, N = Rejected) |

**2. Data Loading & Preparation**

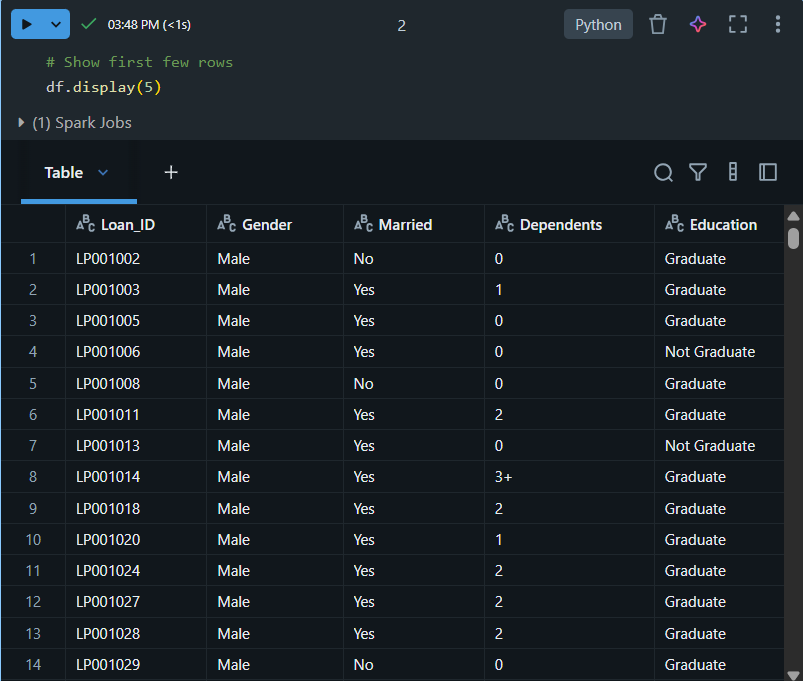
File was uploaded using Databricks UI → /FileStore/tables/LoanData\_\_1\_-1.csv

Loaded with PySpark using:

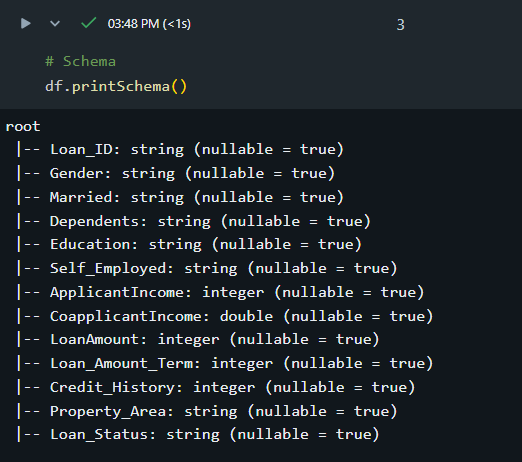
df = spark.read.csv("/FileStore/tables/LoanData\_\_1\_-1.csv", header=True, inferSchema=True)

# Show the first few rows

df.display(5)

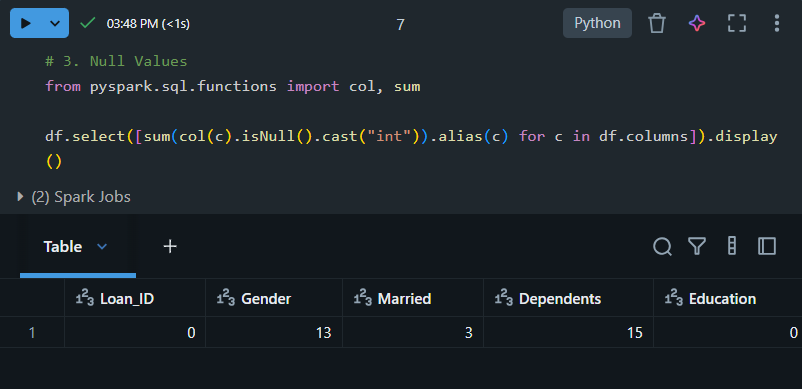


The schema was verified using df.printSchema().



Missing values and data consistency were checked using:

df.select([sum(col(c).isNull().cast("int")).alias(c) for c in df.columns]).display()



Duplicates removed and basic cleaning done (e.g., converting "3+" dependents to numeric value).

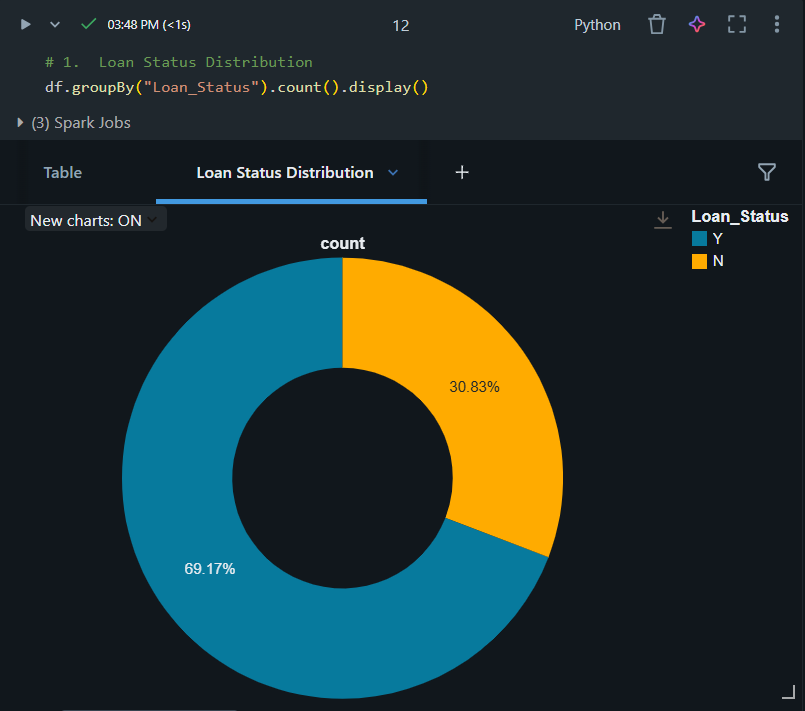
### **3. Visualizations and Insights**

Databricks built-in visualizations were used via display() and groupBy().count().show() methods.

#### **3.1 Loan Status Distribution**

* **Goal:** Understand the overall distribution of loan approvals vs rejections.
* **Chart:** Pie Chart (or Bar Chart)
* **Insight:** Majority of the applicants in the dataset had their loans approved (Loan\_Status = Y), indicating a favorable approval rate.

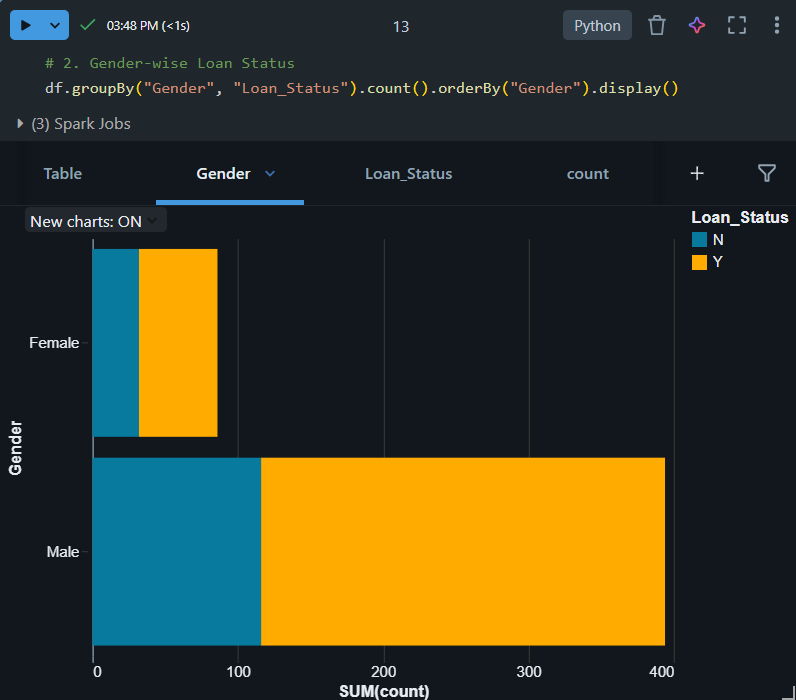
df.groupBy("Loan\_Status").count().display()

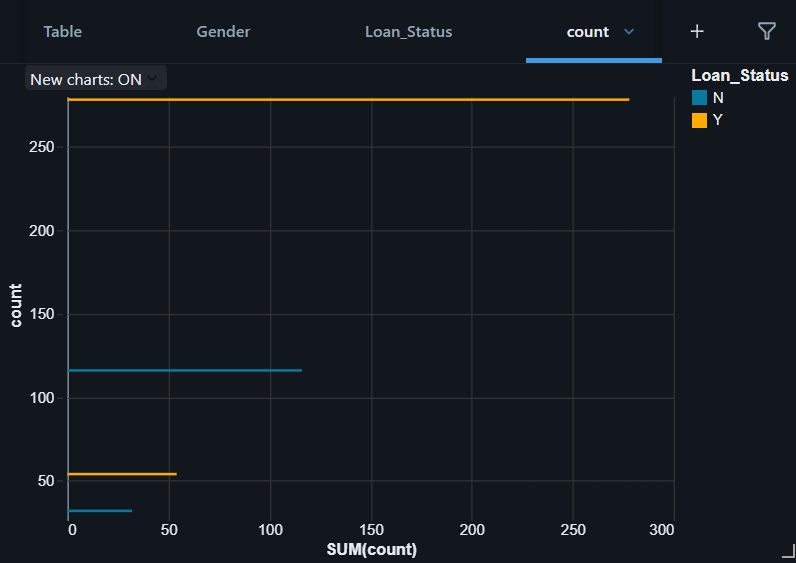


**3.2 Gender-wise Loan Approval**

* **Goal:** Analyze whether gender has any effect on loan approval.
* **Chart:** Grouped Bar Chart
* **Insight:** Males apply for more loans than females. The approval rate is roughly similar across genders, but male applicants dominate in volume.

df.groupBy("Gender", "Loan\_Status").count().orderBy("Gender").display()

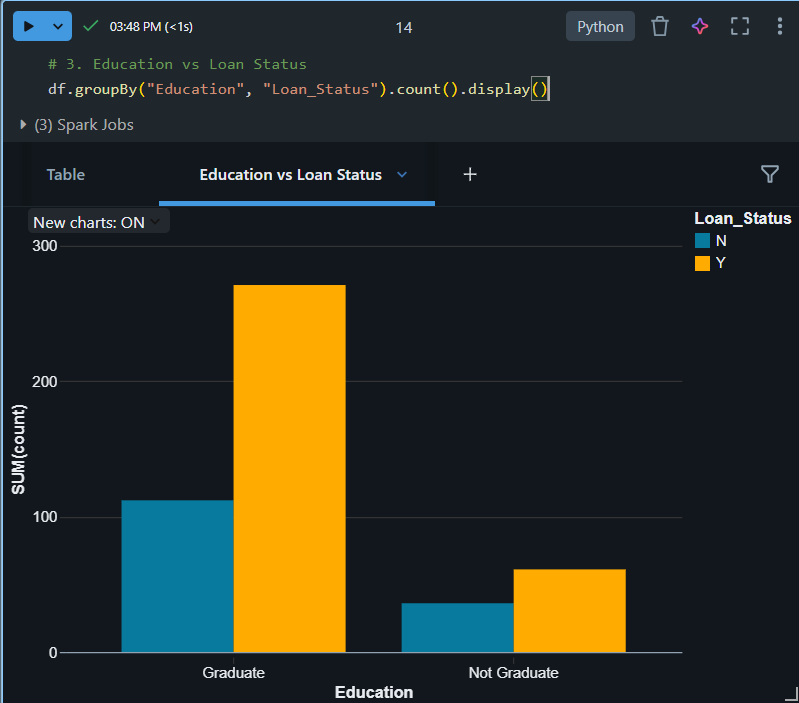




**3.3 Education vs Loan Status**

* **Goal:** Explore the impact of education level on loan approval.
* **Chart:** Grouped Bar Chart
* **Insight:** Graduate applicants have a higher number of approved loans, suggesting that education might have a positive impact on loan approval.

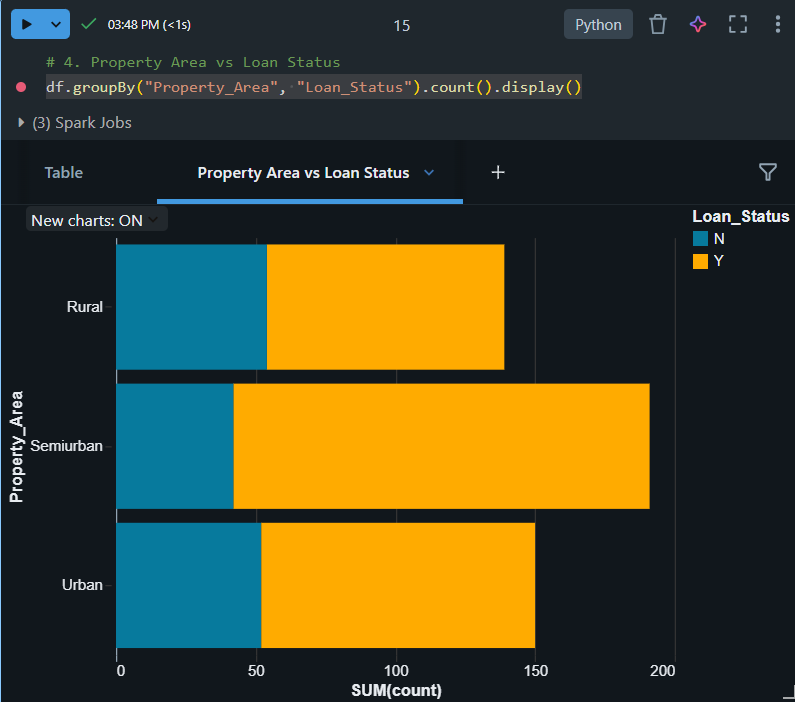
df.groupBy("Education", "Loan\_Status").count().display()



**3.4 Property Area vs Loan Status**

* **Goal:** Understand how the loan approval rate varies by property location.
* **Chart:** Grouped Bar Chart
* **Insight:** Applicants from semiurban areas have a noticeably higher approval rate, followed by urban, with rural areas seeing the least approvals.

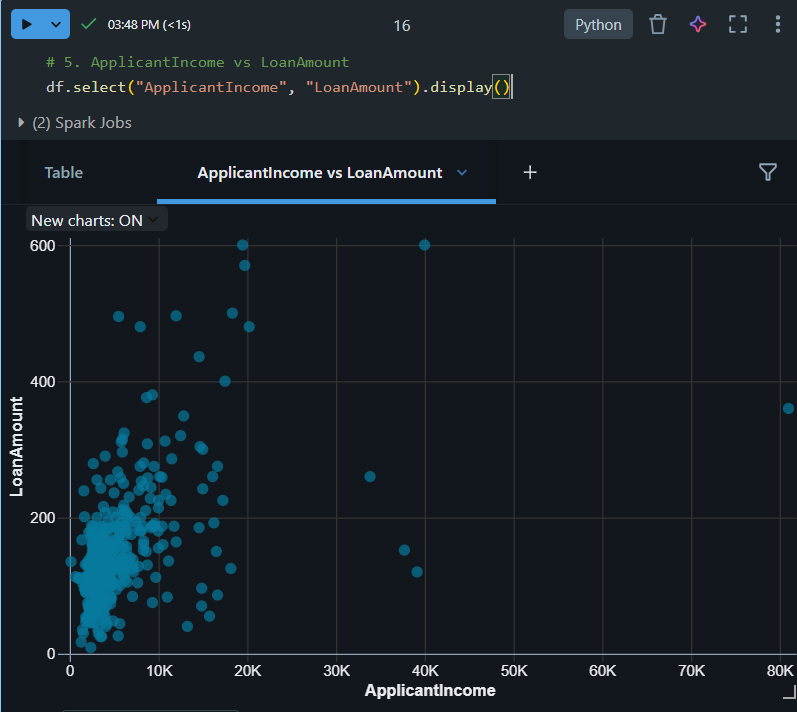
df.groupBy("Property\_Area", "Loan\_Status").count().display()



**3.5 Applicant Income vs Loan Amount**

* **Goal:** Analyze if there's a relationship between applicant income and the loan amount they request.
* **Chart:** Scatter Plot
* **Insight:** There is no strong correlation between income and loan amount. Some high-income applicants request low loans and vice versa, suggesting that loan size may depend on other factors too.

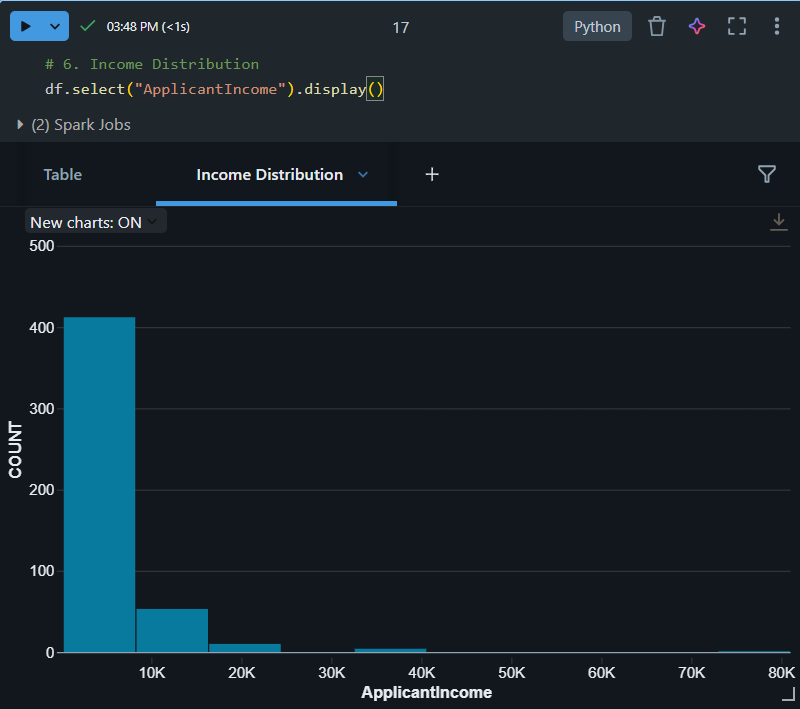
df.select("ApplicantIncome", "LoanAmount").display()



**3.6 Histogram – Income Distribution**

* **Goal:** Understand how applicant income is distributed.
* **Chart:** Histogram
* **Insight:** The income distribution is right-skewed, with a few outliers having very high incomes. Most applicants fall within a moderate income range.

df.select("ApplicantIncome").display()



**3.7 Box Plot – LoanAmount by Education**

* **Goal:** Compare loan amount distribution between graduates and non-graduates.
* **Chart:** Box Plot
* **Insight:** Graduates tend to apply for slightly higher loans, but both groups have outliers. There's slightly more variation among graduates.

df.select("Education", "LoanAmount").display()

