**Exploratory Data Analysis (EDA) Full Report on Delinquency Prediction Dataset**

### **1. Introduction**

This report presents the full exploratory data analysis (EDA) of a financial dataset focused on identifying and understanding key patterns, risk factors, and data issues that may impact the prediction of account delinquency. The objectives include identifying data quality issues, exploring correlations between variables and delinquency, and engineering new risk indicators for modeling.

### **2. Dataset Overview**

* **Total Records:** 500
* **Key Variables:**
  + Customer\_ID: Unique identifier
  + Income: Annual income
  + Credit\_Score: Numerical score indicating credit health
  + Credit\_Utilization: Ratio of used vs. available credit
  + Missed\_Payments: Count of late payments
  + Debt\_to\_Income\_Ratio: Proportion of debt relative to income
  + Loan\_Balance: Outstanding loan amount
  + Employment\_Status, Credit\_Card\_Type, Location: Categorical features
  + Month\_1 to Month\_6: Payment behavior over 6 months
  + Delinquent\_Account: Binary target (0 = No, 1 = Yes)
* **Data Types:** Mix of numerical, categorical, and boolean fields

### **3. Missing Data Analysis**

#### **3.1 Summary of Missing Data:**

# 1. Check total missing values per column  
missing\_summary = df.isnull().sum()  
missing\_summary = missing\_summary[missing\_summary > 0].sort\_values(ascending=False)  
print("Missing values per column:\n", missing\_summary)  
  
# 2. Count rows with more than 1 missing field  
rows\_with\_multiple\_missing = df[df.isnull().sum(axis=1) > 1]  
print(f"Rows with >1 missing values: {len(rows\_with\_multiple\_missing)}")

#### **3.2 Handling Strategy and Justification:**

| Column | Handling Method | Justification |
| --- | --- | --- |
| Income | Synthetic generation using normal distribution | Preserves distribution and realism |
| Loan\_Balance | Median imputation | Robust to skew/outliers |
| Credit\_Score | Mean imputation | Simple and acceptable due to low missing count |

# Impute Income using synthetic normal distribution  
df['Income'] = df['Income'].fillna(  
 np.random.normal(  
 df['Income'].mean(),  
 df['Income'].std(),  
 df['Income'].isnull().sum()  
 )  
)  
  
# Impute Loan\_Balance using median  
df['Loan\_Balance'] = df['Loan\_Balance'].fillna(df['Loan\_Balance'].median())  
  
# Impute Credit\_Score using mean  
df['Credit\_Score'] = df['Credit\_Score'].fillna(df['Credit\_Score'].mean())

### **4. Feature Engineering**

#### **4.1 Derived Feature: Payment Risk Score**

* Monthly columns Month\_1 to Month\_6 were converted using:
  + On-time: 0
  + Late: 1
  + Missed: 2
* Payment\_Risk\_Score is the sum of these mapped values.

monthly\_cols = ['Month\_1', 'Month\_2', 'Month\_3', 'Month\_4', 'Month\_5', 'Month\_6']  
late\_score\_map = {'On-time': 0, 'Late': 1, 'Missed': 2}  
df['Payment\_Risk\_Score'] = df[monthly\_cols].replace(late\_score\_map).sum(axis=1)

### **5. Inconsistencies and Outliers Check**

# Check for Credit Utilization > 1.0 (should be <= 1)  
invalid\_utilization = df[df['Credit\_Utilization'] > 1.0]  
print(f"Records with Credit\_Utilization > 1.0: {len(invalid\_utilization)}")  
  
# Check for non-positive income  
invalid\_income = df[df['Income'] <= 0]  
print(f"Records with non-positive Income: {len(invalid\_income)}")

### **6. Key Findings and Risk Indicators**

#### **6.1 High-Risk Indicators:**

| Indicator | Explanation |
| --- | --- |
| Missed\_Payments | Direct sign of financial distress and late behavior |
| Payment\_Risk\_Score | Aggregated signal of month-to-month risk patterns |
| Credit\_Utilization | High ratios (>0.3) signal overextension on credit lines |
| Debt\_to\_Income\_Ratio | High values (>0.4) indicate limited capacity to repay debt |
| Low Credit\_Score | Lower scores often reflect past defaults and poor financial history |

#### **6.2 Correlation Insights:**

correlations = df.corr(numeric\_only=True)['Delinquent\_Account'].sort\_values(ascending=False)  
print(correlations)

* Payment\_Risk\_Score, Missed\_Payments, and Credit\_Utilization showed strong positive correlation with Delinquent\_Account
* Some high-income customers were still delinquent, suggesting income alone is not predictive

#### **6.3 Unexpected Findings:**

* Some accounts had Credit\_Utilization > 1.0 (103%), indicating behavior that exceeds credit limits
* Monthly payment data inconsistencies (e.g., “Missed” in monthly status but Missed\_Payments = 0)

### **7. AI & GenAI Usage**

AI tools (ChatGPT) were used to:

* Summarize key patterns and anomalies
* Suggest appropriate imputation strategies
* Design and generate synthetic data
* Create derived risk features (like Payment\_Risk\_Score)
* Draft structured reports and visual analysis plans

#### **Example Prompts Used:**

* “Suggest an imputation strategy for missing income values.”
* “Generate a Payment\_Risk\_Score from monthly payment behavior.”
* “Identify the top 3 features that might predict delinquency.”

### **8. Conclusion and Next Steps**

This EDA provided a comprehensive view of the dataset, identified and resolved missing data issues, and surfaced high-risk indicators such as missed payments and high credit utilization. Derived features like Payment\_Risk\_Score add valuable predictive power. The cleaned and enriched dataset is now ready for training machine learning models to predict delinquency risk.

**Next Steps:**

* Build classification models (e.g., logistic regression, decision trees)
* Validate model performance using metrics like AUC and recall
* Integrate results into decision-making systems