# BANK LOAN APPLICATION REPORT

Name : Ananya.V (4GW24CI401).

G-mail: ananya.v.acharya1268@gmail.com

Date : 03-09-2025

**Project Title :** Bank Loan Application Data Preprocessing and Outlier Detection

# INDEX :

## 1.Project Tittle :

**Bank Loan Application Data Preprocessing and Outlier Detection**

## 2.Problem Statement :

**A mid-sized commercial bank is experiencing inconsistencies in its loan approval process due to poor data quality and undetected anomalies in applicant records. The bank collects thousands of loan applications monthly, containing demographic details, financial metrics, credit history, and employment information. However, the raw data often includes missing values, inconsistent formats, and outliers — such as unusually high income or loan amounts — that skew risk assessments and lead to unreliable credit decisions.**

**This project aims to clean and preprocess the bank's loan application dataset and detect outliers that may represent fraud, data entry errors, or exceptional cases. By improving data quality and identifying anomalies, the bank can enhance its credit scoring models, reduce default rates, and make more informed lending decisions.**

## 3.Project Description :

**This project focuses on data preprocessing and outlier detection for bank loan application datasets to improve data quality and reliability. Raw datasets often include missing values, duplicate records, inconsistent formats, and unusual patterns in applicant information, which can mislead analysis and decision-making.**

The system is implemented as an interactive Streamlit-based web application that allows users to:

- Upload raw loan application data (CSV format).
- Automatically clean the dataset by handling missing values, removing duplicates, and standardizing categorical fields.
- Detect and display outliers in income and loan amounts using statistical methods.
- Identify high-risk loans where requested loan amounts exceed repayment capacity.
- Explore insights through visualizations such as loan distribution and employment-based trends.
- Download the cleaned dataset for further use in predictive modeling or analysis.

By ensuring that the dataset is accurate, consistent, and risk-aware, this project provides a practical solution for bankers, analysts, and researchers to make better-informed loan approval and risk management decisions.

## 4.Data and Output Purpose :

- Input Data (Raw Dataset)

The project takes a CSV file of bank loan applications as input. Typical columns include:

- **ApplicationID → Unique identifier for applicants**
- **ApplicantName → Name of applicant**
- **Income → Applicant's monthly/annual income**
- **CreditScore → Credit score of applicant**
- **LoanAmount → Requested loan amount**
- **EmploymentStatus → Employment type (e.g., Employed, Self-employed, Unemployed)**
- **LoanStatus → Approval status (Approved/Rejected)**
- **This dataset may contain missing values, duplicates, inconsistent categories, and outliers.**
- **Processing Steps (Using Pandas & Streamlit)**
- **Data Cleaning**
    - **Handle missing values (e.g., fill income with median, credit score with mean, drop null loan amounts).**
    - **Remove duplicate applications using ApplicationID.**
    - **Standardize categorical fields (e.g., normalize "approved", "APPROVED" → "Approved").**

- **Outlier Detection**
  - **Apply IQR (Interquartile Range) method to detect unusual values in Income and LoanAmount.**
  - **Highlight suspicious applications.**
- **High-Risk Loan Detection**
  - **Flag cases where LoanAmount > 2 × Income.**
- **Data Exploration & Visualization**
  - **Summary statistics of cleaned dataset.**
  - **Histogram of loan amounts by loan status.**
  - **Bar chart of average loan by employment status.**
  - **Search/filter option for specific applicant data.**
  - **Output (Results of Processing)**

**The system produces:**

- **Cleaned Dataset → Free from duplicates, missing values, and inconsistencies.**
- **Outlier Report → List of applicants with unusual income/loan values.**

- **High-Risk Loan Report → Applicants flagged for potential repayment issues.**

- **Visual Insights → Charts showing loan distribution, employment vs loan, and summary stats.**

- **Download Option → Export the cleaned dataset as a CSV for further modeling or analysis.**

## 5.Solution Plan :

**To address the issues of missing values, duplicates, inconsistent formats, and outliers in bank loan application data, the project implements the following solution:**

1. **Data Cleaning**

    - **Fill missing income values with the median and credit scores with the mean.**

    - **Drop records with missing loan amounts to maintain accuracy.**

    - **Remove duplicate entries based on ApplicationID.**

    - **Standardize categorical fields such as loan status to ensure consistency.**

2. **Outlier Detection**

- Use the Interquartile Range (IQR) method to identify abnormal values in Income and LoanAmount.
- Display detected outliers for analyst review.

3. **High-Risk Loan Identification**
    - Flag applications where the loan amount requested is more than twice the applicant's income, indicating higher repayment risk.

4. **Interactive Exploration**
    - Build a Streamlit-based tool where users can:
        - Upload raw CSV files of loan data.
        - Search and filter applicants by name or ID.
        - View dataset summary, missing value report, and cleaned data table.

5. **Visualization & Insights**
    - Provide histograms and bar charts to show loan distribution and employment-based loan trends.
    - Help decision-makers understand patterns in applicant behavior.

6. **Downloadable Clean Data**

    - **Allow users to export the cleaned and preprocessed dataset as a CSV file for further statistical or machine learning analysis.**

## 6.Diagram Design :

### 1. Use Case Diagram

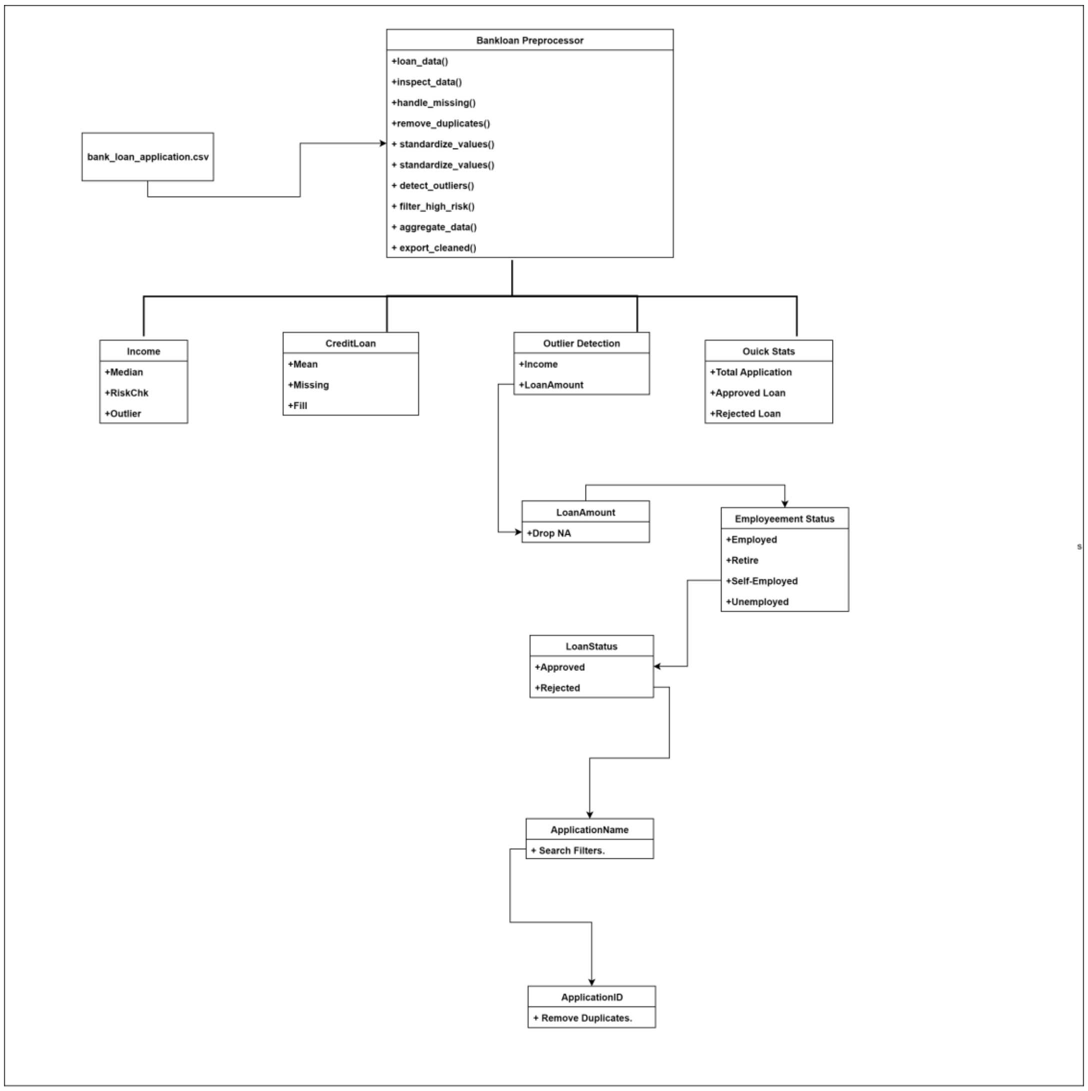
**Shows actors and how they interact with the system.**

**Actors:**

- **User (Banker/Analyst/Researcher)**

**Use Cases:**

- **Upload Dataset**
- **Clean Data (handle missing values, remove duplicates, standardize)**
- **Detect Outliers**
- **Identify High-Risk Loans**
- **View Visualizations & Reports**
- **Download Cleaned Dataset**



•

•

## 2. Activity Diagram

**Represents the workflow step by step:**

**Flow:**
**Start → Upload Dataset → Clean Data → Detect Outliers → Identify High-Risk Loans → Show Results & Visualizations → Download Clean Dataset → End**

## 3. Class Diagram

**If you want to show the structure of your code:**

**Classes:**

- **Dataset (attributes: ApplicationID, Income, CreditScore, LoanAmount, EmploymentStatus, LoanStatus)**
- **Preprocessor (methods: handleMissingValues(), removeDuplicates(), standardizeCategories())**
- **OutlierDetector (methods: detectIQR(), flagHighRisk())**
- **StreamlitApp (methods: uploadData(), displayVisuals(), downloadCleanData())**

## 7.Code and Implementation :

- **loan_cleaned.py :**

**CODE SNIPPET :**

```
import pandas as pd

# Load and clean dataset

df = pd.read_csv("bank_loan_applications.csv")

df.columns = df.columns.str.strip()

# Display basic info

print("Dataset Shape:", df.shape)

print("Column Names:", df.columns.tolist())

print("Missing Values:\n", df.isnull().sum())

# Handle missing values

df['Income'] = df['Income'].fillna(df['Income'].median())

df['CreditScore'] = df['CreditScore'].fillna(df['CreditScore'].mean())

df = df.dropna(subset=['LoanAmount'])

# Remove duplicates
```

```
df = df.drop_duplicates(subset=['ApplicationID'])

# Standardize LoanStatus

df['LoanStatus'] = df['LoanStatus'].str.capitalize()

df['LoanStatus'] = df['LoanStatus'].apply(lambda x: 'Approved' if x ==
'Approved' else 'Rejected')

# Outlier detection using IQR

def detect_outliers(data, column):

   Q1 = data[column].quantile(0.25)

   Q3 = data[column].quantile(0.75)

   IQR = Q3 - Q1

   lower = Q1 - 1.5 * IQR

   upper = Q3 + 1.5 * IQR

   return data[(data[column] < lower) | (data[column] > upper)]


income_outliers = detect_outliers(df, 'Income')

loan_outliers = detect_outliers(df, 'LoanAmount')

# High-risk loans: LoanAmount > 2x Income

high_risk_loans = df[df['LoanAmount'] > 2 * df['Income']]

# Average loan by employment status

avg_loan_by_employment =
df.groupby('EmploymentStatus')['LoanAmount'].mean().reset_index()

# Export cleaned data
```

```
df.to_csv("cleaned_loan_data.csv", index=False)
```

- app.py :

```
import pandas as pd
import streamlit as st
import plotly.express as px

st.set_page_config(page_title="Bank Loan Data Cleaning &
Analysis", layout="wide")
st.title("Bank Loan Application Data Preprocessing & Analysis")
st.markdown("This app cleans, preprocesses, analyzes, and
visualizes loan application data.")

st.sidebar.header("Upload Dataset")
uploaded_file = st.sidebar.file_uploader("Upload CSV file",
type=["csv"])

if uploaded_file:
   df = pd.read_csv(uploaded_file)
   df.columns = df.columns.str.strip()

   df['Income'] = df['Income'].fillna(df['Income'].median())
   df['CreditScore'] =
df['CreditScore'].fillna(df['CreditScore'].mean())
   df = df.dropna(subset=['LoanAmount'])

   df = df.drop_duplicates(subset=['ApplicationID'])

   df['LoanStatus'] = df['LoanStatus'].str.capitalize()
   df['LoanStatus'] = df['LoanStatus'].apply(lambda x: 'Approved' if x
== 'Approved' else 'Rejected')

   def detect_outliers(data, column):
       Q1 = data[column].quantile(0.25)
       Q3 = data[column].quantile(0.75)
       IQR = Q3 - Q1
       lower = Q1 - 1.5 * IQR
       upper = Q3 + 1.5 * IQR
       return data[(data[column] < lower) | (data[column] > upper)]

   high_risk_loans = df[df['LoanAmount'] > 2 * df['Income']]
   avg_loan_by_employment =
df.groupby('EmploymentStatus')['LoanAmount'].mean().reset_inde
x()

   st.subheader("View Dataset")
   search_term = st.text_input("Search by Applicant Name or ID:")
   if search_term:
       filtered_df = df[df.apply(lambda row:
row.astype(str).str.contains(search_term, case=False).any(),
axis=1)]
       st.dataframe(filtered_df, use_container_width=True)
   else:
       st.dataframe(df, use_container_width=True)

   st.subheader("Dataset Summary")
   st.write(df.describe())

   st.subheader("Missing Values After Cleaning")
   st.write(df.isnull().sum())

   st.subheader("Loan Amount Distribution")
   fig = px.histogram(df, x="LoanAmount", color="LoanStatus",
nbins=20)
   st.plotly_chart(fig, use_container_width=True)
   st.subheader("Average Loan by Employment Status")
   fig = px.bar(avg_loan_by_employment, x="EmploymentStatus",
y="LoanAmount", color="EmploymentStatus")
   st.plotly_chart(fig, use_container_width=True)

   st.subheader("Outlier Detection")
   option = st.selectbox("Choose column to detect outliers:",
["Income", "LoanAmount"])
   outliers = detect_outliers(df, option)
   st.write(outliers)

   st.subheader("High-Risk Loans (LoanAmount > 2x Income)")
   st.write(high_risk_loans)

   st.subheader("Download Cleaned Dataset")
   st.download_button(
       label="Download CSV",
       data=df.to_csv(index=False).encode("utf-8"),
       file_name="cleaned_loan_data.csv",
       mime="text/csv"
   )

else:
   st.info("Please upload a CSV file to begin.")
```

3. bank_loan_application.csv

```
ApplicationID,Name,Age,EmploymentStatus,Income,CreditScore,Lo
anAmount,LoanStatus
 A001,John Smith,29,Employed,45000,720,15000,Approved
 A002,Sarah Johnson,35,Self-Employed,60000,680,25000,Approved
 A003,Michael Lee,42,Employed,52000,,18000,Rejected
 A004,Emily Davis,28,Unemployed,,650,8000,Rejected
 A005,David Wilson,50,Employed,85000,710,40000,Approved
```

```
A006,Linda Martinez,31,Employed,47000,600,25000,Rejected
A007,James Anderson,38,Self-
Employed,120000,730,60000,Approved
A008,Mary Thomas,45,Employed,95000,710,100000,Approved
A009,Robert Jackson,52,Retired,30000,500,20000,Rejected
A010,Patricia White,33,Employed,48000,650,25000,Approved
A011,John Smith,29,Employed,45000,720,15000,Approved
```

Key Functions of app.py

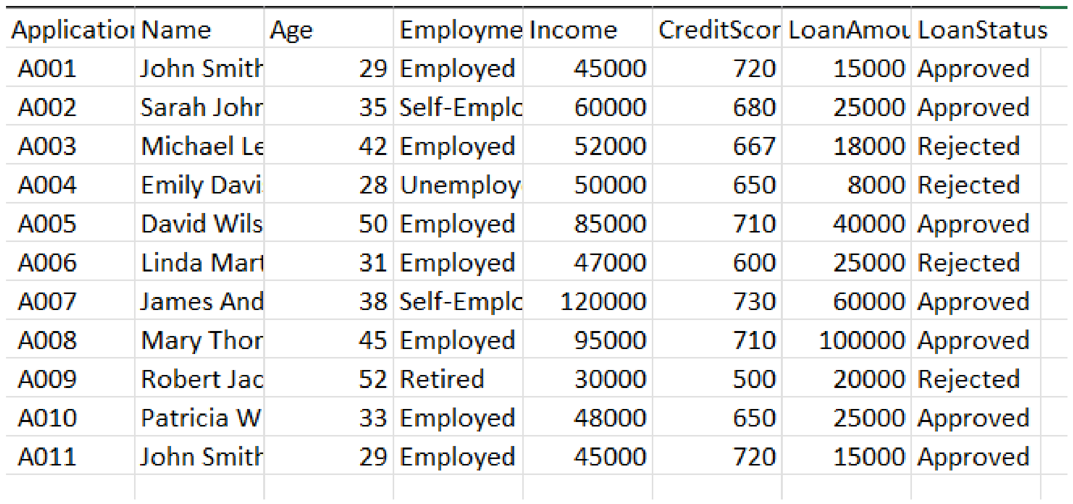
- File Upload
  Allows users to upload a CSV file via Streamlit sidebar.
- Data Cleaning
  - Strips column name whitespace
  - Fills missing Income and CreditScore
  - Drops rows missing LoanAmount
  - Removes duplicate ApplicationID entries
- Categorical Standardization
  Normalizes LoanStatus to either "Approved" or "Rejected"
- Outlier Detection
  Uses IQR method to detect outliers in Income and LoanAmount
- High-Risk Loan Identification
  Flags loans where LoanAmount > 2 × Income
- Group Aggregation
  Calculates average LoanAmount by EmploymentStatus
- Search & Filter
  Enables keyword search by applicant name or ID

- **Visualizations**
  **Displays histogram and bar chart using Plotly**

- **Download Feature**
  **Lets users export the cleaned dataset as a CSV file**

## 8. Output Screenshots & Explanation :

**This is the output of my project where it automatically creates a file name cleaned_loan.csv file :**



| Applicatio | Name | Age | Employme | Income | CreditScor | LoanAmou | LoanStatus |
|---|---|---|---|---|---|---|---|
| A001 | John Smith | 29 | Employed | 45000 | 720 | 15000 | Approved |
| A002 | Sarah John | 35 | Self-Emplc | 60000 | 680 | 25000 | Approved |
| A003 | Michael Le | 42 | Employed | 52000 | 667 | 18000 | Rejected |
| A004 | Emily Davi | 28 | Unemploy | 50000 | 650 | 8000 | Rejected |
| A005 | David Wils | 50 | Employed | 85000 | 710 | 40000 | Approved |
| A006 | Linda Mart | 31 | Employed | 47000 | 600 | 25000 | Rejected |
| A007 | James And | 38 | Self-Emplc | 120000 | 730 | 60000 | Approved |
| A008 | Mary Thor | 45 | Employed | 95000 | 710 | 100000 | Approved |
| A009 | Robert Jac | 52 | Retired | 30000 | 500 | 20000 | Rejected |
| A010 | Patricia W | 33 | Employed | 48000 | 650 | 25000 | Approved |
| A011 | John Smith | 29 | Employed | 45000 | 720 | 15000 | Approved |

**The output is a cleaned loan dataset where:**

- **Column names and categories are standardized.**
- **Duplicate records are removed.**
- **Missing values are handled.**
- **Outliers in Income and LoanAmount are detected.**
- **High-risk loans (Loan > 2× Income) are flagged.**
- **Visual insights (charts, summaries) help in better decision-making.**

## 9. Conclusion :

This project successfully demonstrates how data preprocessing and outlier detection improve the quality of bank loan application datasets. By handling missing values, removing duplicates, standardizing categories, and identifying high-risk loans, the system ensures that the data is accurate, consistent, and reliable. The interactive Streamlit tool further enhances usability by allowing users to upload raw data, explore insights, visualize trends, and download the cleaned dataset.

Overall, the project provides a practical solution for bankers, analysts, and researchers to make informed loan approval and risk management decisions, while also preparing the dataset for future predictive modeling.

## 10. Closure :

The Bank Loan Application Data Preprocessing and Outlier Detection project achieved its goal of transforming raw, inconsistent loan datasets into clean, structured, and reliable data. Through systematic preprocessing, duplicate removal, missing value handling, and outlier detection, the project ensures improved data quality and risk identification.

The Streamlit-based tool makes the process interactive and user-friendly, allowing easy dataset upload, exploration, visualization, and download of cleaned data. This project not only supports better loan decision-making but also prepares the dataset for advanced analytics and machine learning models.

In closure, the project successfully delivers a practical and efficient solution that bridges the gap between messy real-world data and accurate decision-making in the banking sector.

## 12.BIBLIOGRAPHY :

**The project referred to several key resources for implementation and understanding.**

***Python for Data Analysis* by Wes McKinney (O'Reilly Media, 2017) was used as a guide for data handling and preprocessing using the Pandas library. Official documentation from Pandas, Streamlit, and Plotly Express provided practical insights into coding and visualization. Concepts of data mining and preprocessing were further supported by *Data Mining: Concepts and Techniques* by Han, Kamber, and Pei (Morgan Kaufmann, 2011). Additional references included online tutorials and articles from GeeksforGeeks, Kaggle, and Medium, which helped in understanding techniques such as missing value handling, outlier detection, and risk identification. A sample loan dataset was created and adapted for academic purposes to demonstrate the application of these methods effectively.**