I have used python coding and Power BI to analyze the dataset and provided the insights:

1. Data Quality and Cleaning Observations

Currency Formatting - Loan amounts and deposits are in UK currency format, these are converted to numeric for analysis.

Missing Values - Many rows have NA for APR and funded status, especially for declined applications.

2. Application Outcomes

Approval Rate

Approved: The majority of records with a valid APR are approved.

Declined: Most declined applications have NA for APR and funded status.

Funded Rate

Not all approved loans are funded (Funded = Yes). Many approved applications are not funded, indicating drop-off after approval.

3. Key Variable Insights

A. Age range: 18–65 years.

**Approvals and funding occur across all ages, but younger applicants (18–25) have a higher decline rate.**

B. Car Type - Saloon, SUV, Convertible.

Saloon is the most common, followed by SUV and Convertible.

**Approval and funding rates are similar across car types, but Convertibles tend to have higher loan amounts**.

C. Area - Urban vs Rural:

No strong bias in approval rates, but **rural applications seem to have a slightly higher decline rate**.

D. APR (Interest Rate) - 5% to 25%

Lower APRs (0.05–0.15) are more common.

Higher APRs (0.2–0.25) are often associated with lower creditworthiness or higher risk, but approvals still occur at these rates.

E. Loan Amount and Deposit

Loan Amounts - ~£2,500 to over £26,000.

Deposits - ~£500 to over £1,700.

**Higher loan amounts are more likely for Convertibles and SUVs. Larger deposits may increase approval chances.**

4. Patterns and Trends

* Approval and Funding by Age

**Younger Applicants (18–25) - Higher decline rates, when approved, less likely to be funded.**

**Older Applicants (50+) - Higher approval and funding rates.**

df['Age\_group'] = pd.cut(df['Age'], bins=[17,25,35,45,55,65,100], labels=['18-25','26-35','36-45','46-55','56-65','66+'])

age\_summary = df.groupby('Age\_group').agg(

    applications=('Age', 'count'),

    approvals=('Application\_outcome', lambda x: (x == 'Approved').sum()),

    approval\_rate=('Application\_outcome', lambda x: (x == 'Approved').mean()),

    conversions=('Converted', 'sum'),

    conversion\_rate=('Converted', 'mean')

* Approval and Funding by Car Type

Convertibles: Higher average loan amounts, but approval and funding rates are similar to other types.

**SUVs: Slightly higher decline rate, possibly due to higher loan amounts or risk profiles.**

car\_type\_summary = df.groupby('Car\_type').agg(

    applications=('Car\_type', 'count'),

    approvals=('Application\_outcome', lambda x: (x == 'Approved').sum()),

    approval\_rate=('Application\_outcome', lambda x: (x == 'Approved').mean()),

    conversions=('Converted', 'sum'),

    conversion\_rate=('Converted', 'mean')

* APR vs funding

Lower APRs are more likely to be funded.

**Higher APRs - Approvals occur, but funding rates drop, possibly due to unattractive loan terms for the customer.**

* Area

**Rural - Slightly higher decline rate, possibly due to risk assessment or economic factors.**

area\_summary = df.groupby('Area').agg(

    applications=('Area', 'count'),

    approvals=('Application\_outcome', lambda x: (x == 'Approved').sum()),

    approval\_rate=('Application\_outcome', lambda x: (x == 'Approved').mean()),

    conversions=('Converted', 'sum'),

    conversion\_rate=('Converted', 'mean')

5. Actionable Insights

Consider reviewing approval criteria for **younger applicants**, as they are disproportionately declined.

**Loans with lower APRs are more likely to be funded**. Consider offering more competitive rates to increase funding rates.

**Larger deposits may improve approval chances**. Consider encouraging higher deposits for borderline applicants.

**Slightly higher risk in rural areas,** consider tailored risk models or additional verification for rural applicants.

6. Potential Next Steps

Investigate why approved loans are not funded? Are customers rejecting offers, or are there process issues?

Python Code:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import os

pd.set\_option('display.max\_columns', None)

pd.set\_option('display.width', 120)

# Load data

df = pd.read\_csv('Case\_Study\_for\_Data\_Test\_25.csv')

# Data Cleaning

def clean\_currency(x):

    if isinstance(x, str):

        return float(x.replace('£', '').replace(',', '').replace('NA', '').strip() or np.nan)

    return x

df['Loan\_amount'] = df['Loan\_amount'].apply(clean\_currency)

df['Deposit'] = df['Deposit'].apply(clean\_currency)

# Convert APR to float, handle NA

if 'APR' in df.columns:

    df['APR'] = pd.to\_numeric(df['APR'], errors='coerce')

# Remove duplicate rows

initial\_shape = df.shape

df = df.drop\_duplicates()

print(f"Removed {initial\_shape[0] - df.shape[0]} duplicate rows.")

print("\n--- DATA OVERVIEW ---")

print(df.info())

print(df.head())

print("\nMissing values per column:\n", df.isnull().sum())

# Summary

print("\n--- SUMMARY STATISTICS ---")

print(df.describe(include='all'))

# Approval and Funding Rates

print("\n--- APPLICATION OUTCOMES ---")

if 'Application\_outcome' in df.columns:

    print(df['Application\_outcome'].value\_counts(normalize=True))

    if 'Funded' in df.columns:

        approved = df[df['Application\_outcome'] == 'Approved']

        print("\nFunding rate among approved:")

        print(approved['Funded'].value\_counts(normalize=True))

# Grouped Analysis

print("\n--- GROUPED ANALYSIS ---")

if 'Car\_type' in df.columns:

    print("\nBy Car Type:")

    print(df.groupby('Car\_type')['Application\_outcome'].value\_counts(normalize=True).unstack())

if 'Area' in df.columns:

    print("\nBy Area:")

    print(df.groupby('Area')['Application\_outcome'].value\_counts(normalize=True).unstack())

if 'Age' in df.columns:

    df['Age\_group'] = pd.cut(df['Age'], bins=[17,25,35,45,55,65,100], labels=['18-25','26-35','36-45','46-55','56-65','66+'])

    print("\nBy Age Group:")

    print(df.groupby('Age\_group')['Application\_outcome'].value\_counts(normalize=True).unstack())

# Visualizations

def plot\_count(col, title):

    plt.figure(figsize=(8,4))

    sns.countplot(x=col, data=df, order=df[col].value\_counts().index)

    plt.title(title)

    plt.xticks(rotation=45)

    plt.tight\_layout()

    plt.show()

def plot\_box(x, y, title):

    plt.figure(figsize=(8,4))

    sns.boxplot(x=x, y=y, data=df)

    plt.title(title)

    plt.xticks(rotation=45)

    plt.tight\_layout()

    plt.show()

# Countplots

if 'Car\_type' in df.columns:

    plot\_count('Car\_type', 'Applications by Car Type')

if 'Area' in df.columns:

    plot\_count('Area', 'Applications by Area')

if 'Application\_outcome' in df.columns:

    plot\_count('Application\_outcome', 'Application Outcomes')

if 'Funded' in df.columns:

    plot\_count('Funded', 'Funded Loans')

# Boxplots

if 'Loan\_amount' in df.columns and 'Car\_type' in df.columns:

    plot\_box('Car\_type', 'Loan\_amount', 'Loan Amount by Car Type')

if 'Loan\_amount' in df.columns and 'Area' in df.columns:

    plot\_box('Area', 'Loan\_amount', 'Loan Amount by Area')

if 'Loan\_amount' in df.columns and 'Application\_outcome' in df.columns:

    plot\_box('Application\_outcome', 'Loan\_amount', 'Loan Amount by Application Outcome')

if 'APR' in df.columns and 'Application\_outcome' in df.columns:

    plot\_box('Application\_outcome', 'APR', 'APR by Application Outcome')

# Correlation heatmap

num\_cols = df.select\_dtypes(include=np.number).columns

if len(num\_cols) > 1:

    plt.figure(figsize=(10,8))

    sns.heatmap(df[num\_cols].corr(), annot=True, cmap='coolwarm')

    plt.title('Correlation Heatmap')

    plt.show()

# Save cleaned data for further analysis

cleaned\_path = 'Case\_Study\_for\_Data\_Test\_25\_cleaned.csv'

df.to\_csv(cleaned\_path, index=False)

print(f"\nCleaned data saved to {cleaned\_path}")

**Power BI Reports**





