Credit Score Analysis

Aabhash Shrestha, Garvit Sakhuja, Joseph Vazhiyil

Objective

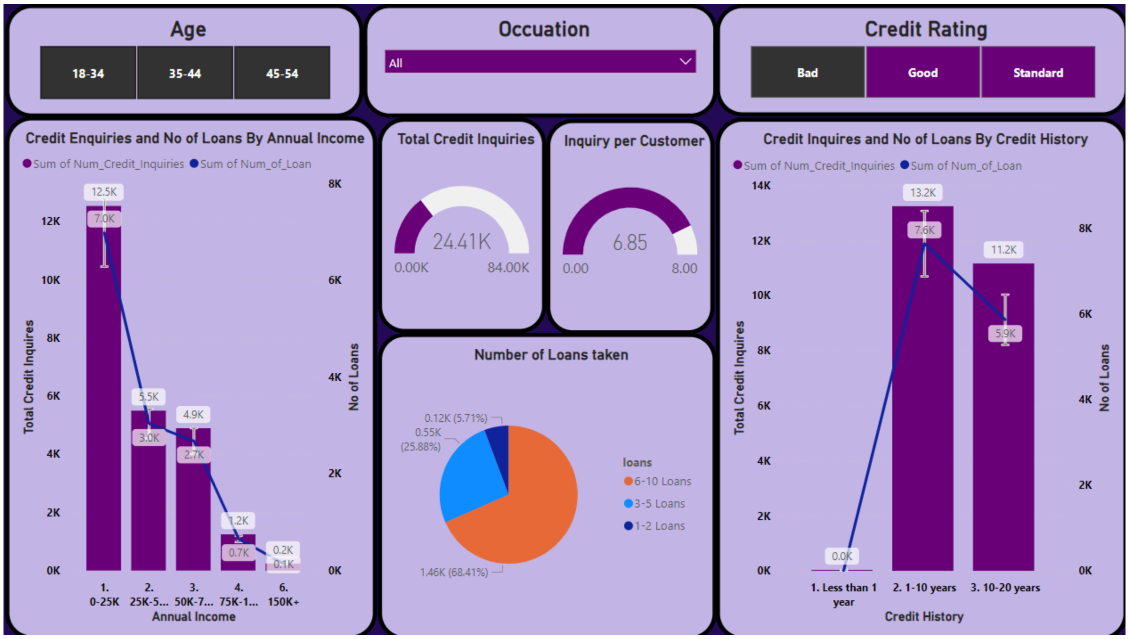
This visualization aims to provide in depth analysis of credit inquiries and loan behaviours which are separated by annual income, age groups, credit history age and occupation. The goal of this visualization is to facilitate decision-making by identifying relationships between certain demographics and financial factors. Finding these relationships will allow for better financial solutions and deeper understandings on each customer demographic.

Dataset

The data was obtained from Kaggle.
https://www.kaggle.com/datasets/parisrohan/credit-score-classification/code.
This dataset contains customer's credit related information, such as Customer's Name, Age, credit inquiries, credit rating, no of loans taken, etc.

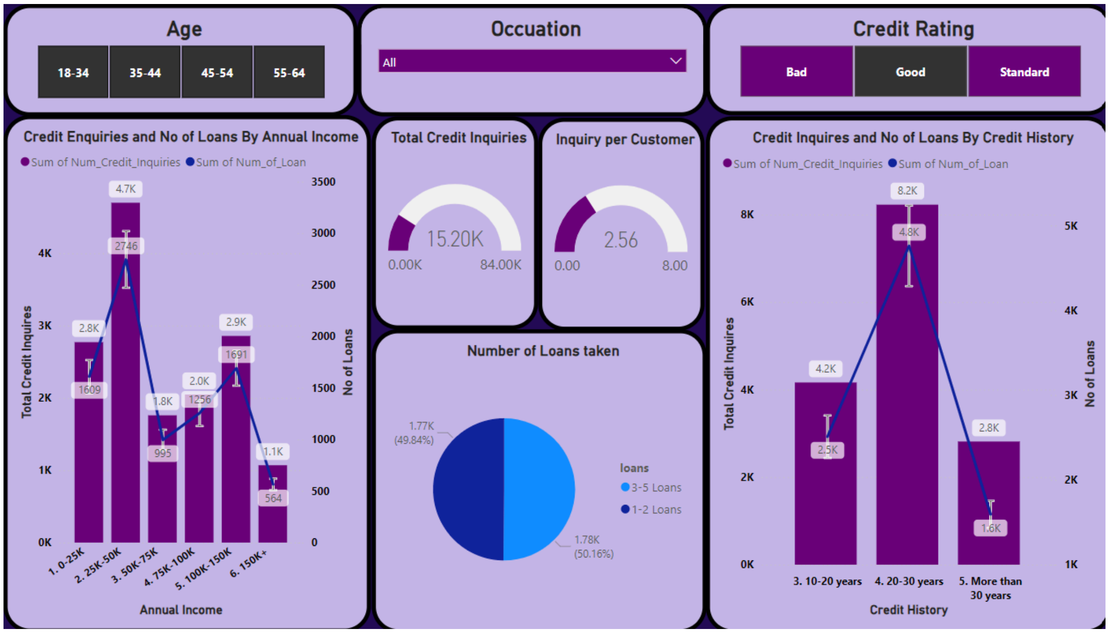
Data Storytelling and Analysis

- Credit Rating Filter: Bad



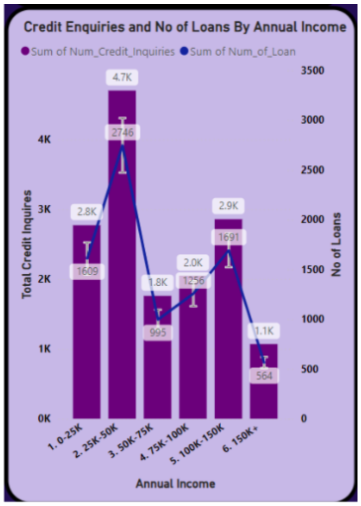
- Majority of loan takers with bad credit ratings have taken 6-10 loans which suggests that individuals with a bad credit rating may take a high number of loans
- More credit inquiries and the number of loans taken by people in the 0-25k salary bucket. Most likely due to their low salary, they often take micro loans which are susceptible to credit checks, thus the increase in both inquiries and number of loans taken
- Higher credit inquiry per customer indicating multiple credit checks from possible creditors, indicating possible financial instability among people with higher credit inquiries
- Most credit inquiries and loans are taken with people with 1-10 years of credit history indicating activity in short term loans rather than long term, possibly due to their inability to qualify

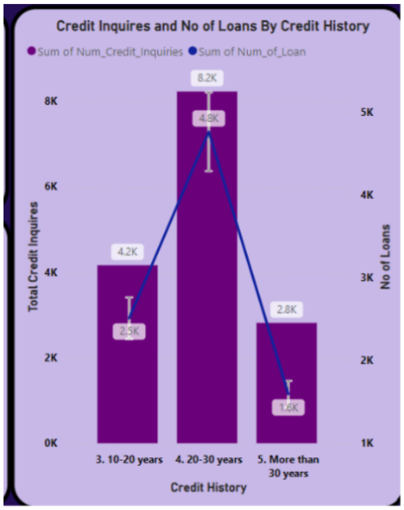
- Credit Rating Filter: Good



- Users with a good credit rating generally have a credit history exceeding 10-20 years, while those with a bad credit rating typically have a credit history of less than 10-20 years
- Users with a good credit rating generally have fewer loans compared to those with a bad credit rating. This pattern suggests that taking only a few loans (<5) may help improve credit score
- Low inquiry/customer also suggests that customers in the good cohort have taken less number of loans

Advanced Data Analysis:





Python Code:

```
import os
import pandas as pd
from kaggle.api.kaggle_api_extended import KaggleApi

# Authenticate Kaggle API
api = KaggleApi()
api.authenticate()

# Define dataset and download path
dataset = "parisrohan/credit-score-classification"
download_path = "."  # Current directory or specify your desired path

# Download and unzip the dataset
print("Downloading dataset...")
api.dataset_download_files(dataset, path=download_path, unzip=True)

# List downloaded files
print("Dataset downloaded and unzipped. Checking files...")
files = [f for f in os.listdir(download_path) if f.endswith('.csv')]
print("CSV files found:", files)

# Load the first CSV file into a Pandas DataFrame (if available)
```

```
if files:
    file_path = os.path.join(download_path, files[0])
    data = pd.read_csv(file_path)
    print("Preview of the raw dataset:")
    print(data.head())
else:
    print("No CSV files found in the specified path.")
    exit()

#cleaning
if 'Occupation' in data.columns:
    valid_occupations = [
        'Scientist', 'Teacher', 'Engineer', 'Entrepreneur',
        'Developer', 'Lawyer', 'Media_Manager', 'Doctor', 'Journalist',
        'Manager', 'Accountant', 'Musician', 'Mechanic', 'Writer',
        'Architect'
    ]
    data = data[data['Occupation'].apply(lambda x: x in valid_occupations if pd.notna(x) else
False)]

# Define valid ranges for each column
valid_ranges = {
    "Age": (18, 100),
    "Num_Bank_Accounts": (1, 8),
    "Num_Credit_Card": (1, 10),
    "Interest_Rate": (1, 33),
    "Num_of_Loan": (0, 9),
    "Num_Credit_Inquiries": (0, 17)
}

# Drop rows with NA values
data = data.dropna()

# Ensure all specified columns are within their valid ranges
for column, (min_val, max_val) in valid_ranges.items():
    if column in data.columns:
        data[column] = pd.to_numeric(data[column], errors='coerce')
        data = data[(data[column] >= min_val) & (data[column] <= max_val)]

# Drop rows with any negative values in the dataset
numeric_data = data.select_dtypes(include=['number'])
data = data[(numeric_data >= 0).all(axis=1)]
```

```
# Keep only rows with valid values in the 'credit_mix' column
if 'Credit_Mix' in data.columns:
    valid_credit_mix_values = ["Good", "Standard", "Bad"]
    data = data[data['Credit_Mix'].isin(valid_credit_mix_values)]
```