STOCK PRICE PREDICTION

A PROJECT REPORT

Submitted in partial fulfillment for the award of the degree of

programme of B.Sc. Data Analytics

SUBMITTED BY

VISHAL A

2213141104047

Under the guidance of

Dr. M. LAVANYA, M.C.A., B.Ed., M.Phil., Ph.D.

Assistant Professor

Bachelor of Science in Data Analytics

GURU NANAK COLLEGE

(AUTONOMOUS)

Affiliated to University of Madras

Accredited at 'A++' Grade by NAAC | An ISO 9001 2015 Certified Institution

Guru Nanak Salai, Velachery, Chennai – 600 042.

MARCH-2025

GURU NANAK COLLEGE

(AUTONOMOUS)

Affiliated to University of Madras
Accredited at 'A++' Grade by NAAC | An ISO 9001 2015 Certified Institution
Guru Nanak Salai, Velachery, Chennai – 600 042.

BACHELOR OF SCIENCE IN DATA ANALYTICS

BONAFIDE CERTIFICATE

This is to certify that, this is a bonafide record of work done by **VISHAL A, 2213141104047** of for the Final Year Project during the Academic Year 2024-25.

PROJECT GUIDE **HEAD OF THE DEPARTMENT**

Submitted for the Project Viva Voce Examination held on _____

at GURU NANAK COLLEGE (Autonomous), Guru Nanak Salai, Velachery, Chennai - 600 042.

Internal Examiner
Date:

External Examiner
Date:

DECLARATION

I **VISHAL A, 2213141104047** studying III Year B.Sc. DATA ANALLYTICS at Guru Nanak College (Autonomous), Chennai hereby declare that this the Report of my Project entitled, **CREDIT CARD INTENT ANALYSIS** is the record of the original work carried out by me under the Guidance and Supervision of **Dr. M. Lavanya** towards the partial fulfillment of the requirements of the award of the Degree of B.Sc. DATA ANALYTICS**.** I further declare that this has not been submitted anywhere for the award of Degree/Diploma or any other similar to this before.

PLACE: CHENNAI **VISHAL A**

DATE: **2213141104047**

ACKNOWLEDGEMENT

I would like to thank the **Principal Dr. T. K. Avvai Kothai and Vice Principal Dr. P. V. Kumaraguru** for providing the necessary resources and facilities for the completion of this project.

I extend my deepest thanks to **Head of the Department Dr. S. Nirmala Devi** whose guidance, support, and encouragement were invaluable throughout this endeavor. Her expertise and insights have been instrumental in shaping this project and enhancing its quality.

I owe my Guide **Dr. M. Lavanya**, a debt of gratitude for her invaluable guidance, patience, and encouragement. Her mentorship has been a beacon of light, steering me through the complexities of this project and helping me realize my potential.

I also like to extend my thanks to the Faculty Members **Dr. K. Nalini, Dr. R. Ramya, Ms. V. Chellathai** for their valuable suggestion during the course of the study of my project.

Last but not least, I thank my **family and friends** for their unwavering encouragement and understanding during this journey.

ABSTRACT

INTRODUCTION:

Credit card intent analysis aims to understand the factors influencing an individual's decision to apply for a credit card. This study explores various demographic, financial, and behavioral attributes that affect credit card application intent. By analyzing key variables such as age, income, employment length, credit history, and past defaults, the project seeks to identify trends and patterns in credit card applications.

The analysis involves data preprocessing, statistical evaluation, and visualization techniques to uncover insights

ANALYSIS:

In **SAS**, **descriptive statistics (PROC MEANS, PROC UNIVARIATE)** provide insights into applicant age, income, employment length, and credit history. **Frequency analysis (PROC FREQ)** examines categorical variables, revealing trends in loan intent, homeownership status, and default history. In **Power BI**, interactive dashboards provide a visual representation of key insights. **Bar charts** illustrate the distribution of credit card intent across different demographic groups, while **pie charts** highlight approval rates. **Line and trend charts** analyze patterns in income vs. approval likelihood, and **filters** allow dynamic exploration of applicant characteristics. **Matrix visualizations** compare approval rates by employment length, credit history, and previous defaults. By combining **SAS statistical insights** with **Power BI's visual analytics**, this project enables better decision-making for financial institutions, helping optimize approval processes and reduce credit risk.

TABLE OF CONTENTS

Credit Card Intent Analysis

1.1 INRODUCTION :

Credit card intent analysis is a strategic approach that financial institutions and businesses use to understand customer behavior, predict their likelihood of applying for a credit card, and optimize marketing efforts. By leveraging data analytics and machine learning, companies can identify potential applicants, assess their creditworthiness, and tailor financial products to meet customer needs.

The rise of digital banking and financial services has generated vast amounts of customer data, including demographic details, spending habits, income levels, and credit histories. Analyzing this data helps institutions make data-driven decisions, reduce risks, and enhance customer acquisition strategies. This project focuses on identifying patterns in customer behavior and developing predictive models to determine the intent of customers in applying for a credit card.

1.2 Objective :

The main goal of this project is to analyze customer data to determine their intent regarding credit card applications. This includes:

- Identifying patterns in customer behavior that indicate interest in applying for a credit card.
- Segmenting customers based on their likelihood to apply for a card.
- Understanding the impact of financial factors such as income, credit score, and spending habits.
- Developing predictive models to classify potential applicants.

This project aims to bridge the gap between customer intent and business decision-making, helping financial institutions optimize their credit card offerings and improve customer acquisition strategies.

Introduction About SAS

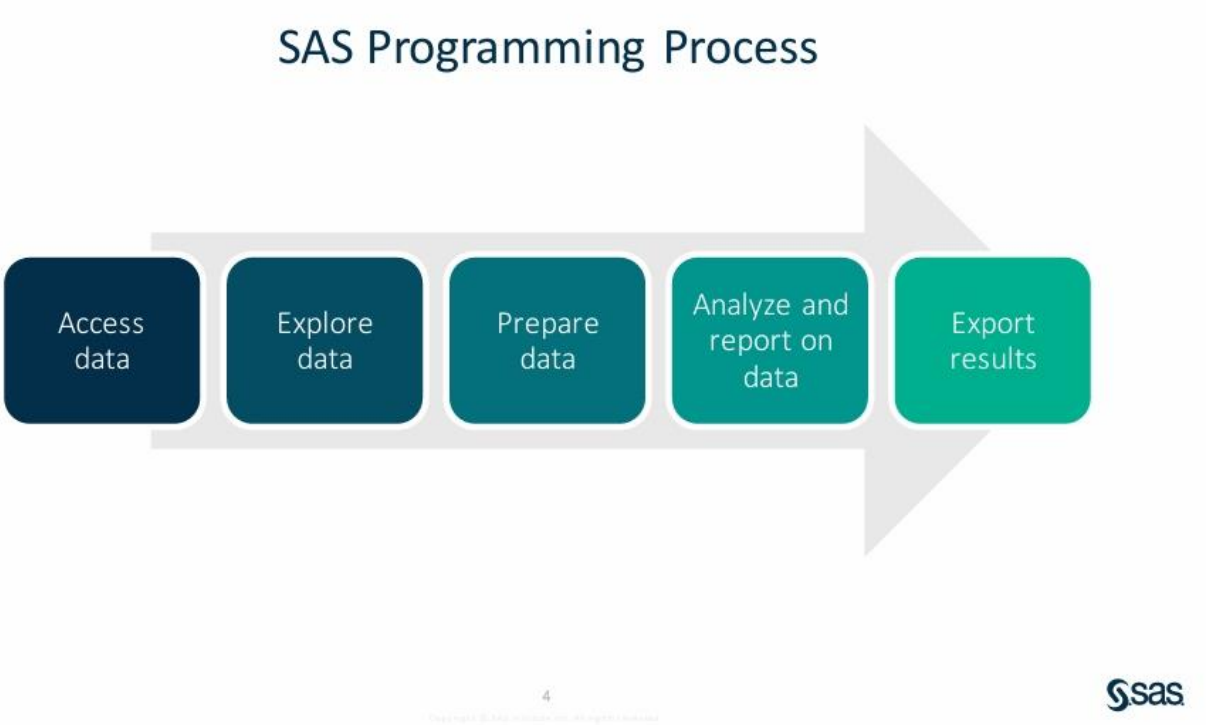
2.1 SAS (Statistical Analysis System)

SAS (Statistical Analysis System) is a powerful software suite used for data management, advanced analytics, business intelligence, and statistical analysis. Developed by SAS Institute Inc., it is widely used in industries such as healthcare, finance, and government for data-driven decision-making.

SAS Programming Process

As you go through the process of making data meaningful and actionable, you will likely follow these basic steps: access, explore, prepare, analyze and report, and export. SAS has programming tools for each of these steps in the process. You follow this process as you learn the fundamentals of the SAS programming language.

3.1 Steps in SAS Programming Process



The **SAS programming process** involves a series of steps to analyze and manage data. It follows a structured approach:

3.2 Access Data

In **SAS**, data can be accessed from various sources such as SAS datasets, external files (CSV, Excel, TXT), and relational databases (SQL, Oracle, MySQL). SAS provides different methods to import, read, and manipulate data for analysis.

SAS datasets are stored in libraries. The LIBNAME statement assigns a library reference to a folder containing SAS datasets.
SAS allows importing data from different file formats like CSV, Excel, and TXT.

3.3 Explore Data

After accessing data in SAS, the next step is to explore it to understand its structure, identify missing values, and summarize key statistics. SAS provides several procedures to explore datasets effectively.

1. **Viewing Data Structure**

- Check dataset details using PROC CONTENTS.
- Displays variable names, data types, and dataset properties.

2. Summarizing Data

- Descriptive Statistics using PROC MEANS
- Provides summary statistics (mean, min, max, standard deviation).

3.Checking Missing Values

- Identify missing values using PROC MEANS

4.Sorting and Filtering Data

- **Sorting Data** using PROC SORT
- Sorts data by default in Ascending order.

- **Prepare Data**

Data preparation is a crucial step in SAS to clean, transform, and structure data for analysis. This involves handling missing values, filtering data, creating new variables, merging datasets, and formatting data.

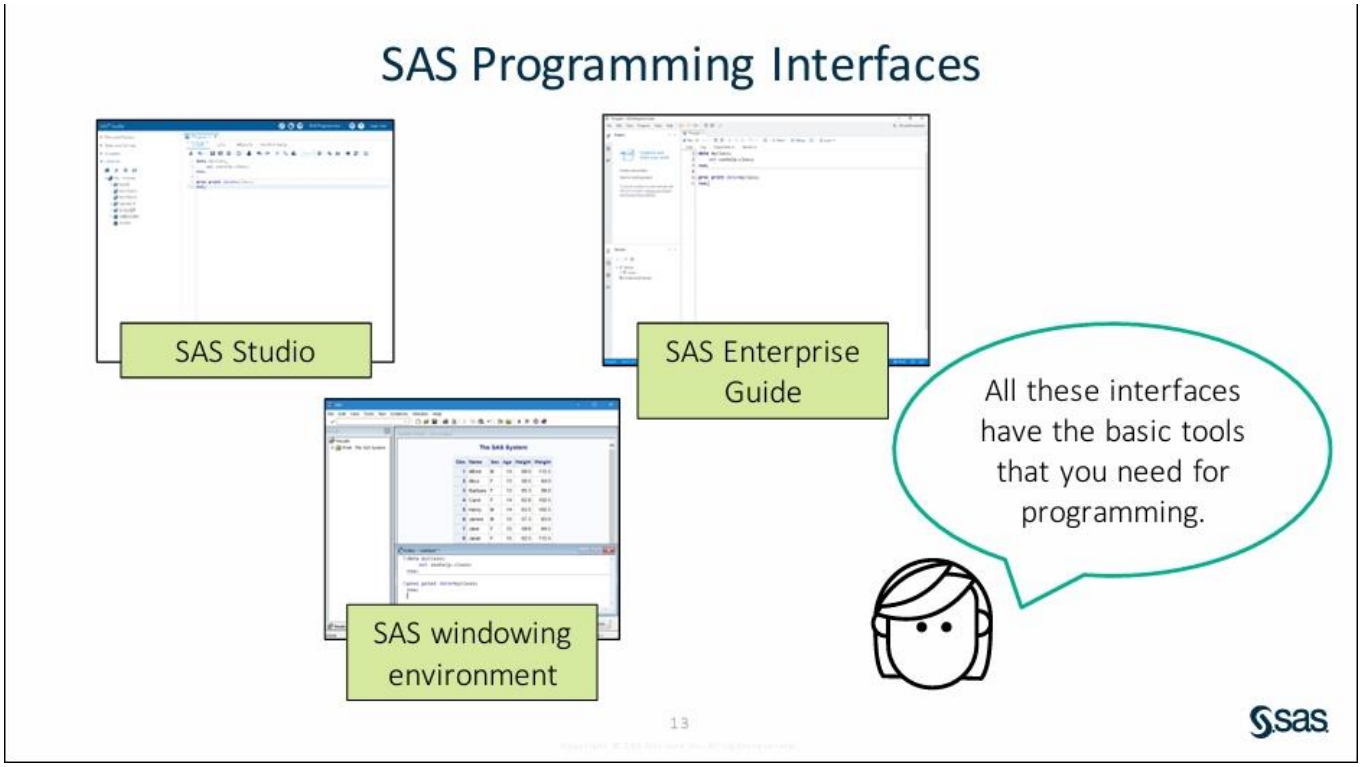
- **Analyse Data**

Once data is prepared, the next step is to analyze it using SAS procedures. SAS provides various tools for descriptive statistics, inferential analysis, and predictive modeling.

- **Export Results**

After analyzing data in SAS, you can export results to various formats like CSV, Excel, PDF, or databases for reporting and further use.

SAS Programming Interfaces



SAS Programming Interfaces

SAS provides multiple interfaces for programming, data analysis, and visualization. These interfaces allow users to write and execute SAS code, interact with data, and generate reports efficiently.

4.1. SAS Windowing Environment (SAS Display Manager System - DMS)

- Traditional **desktop-based** SAS interface.
- Includes **Editor, Log, and Output** windows.
- Used for writing and executing SAS programs.
- Supports interactive debugging and execution.

4.2. SAS Enterprise Guide (SAS EG)

- A **point-and-click** GUI for SAS analysis.
- Supports **drag-and-drop** functionality for users with minimal coding knowledge.
- Generates SAS code automatically.
- Ideal for business users and analysts.

4.3 SAS Studio (Web-Based Interface)

- A **browser-based** SAS interface.
- Accessible from **any device** with an internet connection.
- Provides an interactive code editor with syntax highlighting and auto-completion.
- Allows users to run code without installing SAS locally.
- **Example: Running Code in SAS Studio**
- Open **SAS Studio** in a web browser.
- Write SAS code in the editor.
- Click **Run** to execute the program.

4.4 SAS Viya

- A cloud-based SAS **AI and analytics platform**.
- Supports **high-performance computing and machine learning**.
- Accessible via SAS Studio, Python, R, and REST APIs.

Introduction About POWER BI

5.1 Power BI is a business intelligence (BI)

Power BI is a business intelligence (BI) and data visualization tool developed by Microsoft that allows users to collect, analyze, and visualize data in an interactive and insightful way. It transforms raw data into actionable insights through dynamic reports and dashboards, making it an essential tool for data-driven decision-making in various industries.

Why Power BI?

In today's data-driven world, organizations generate vast amounts of data from multiple sources such as databases, cloud services, APIs, and spreadsheets. Power BI simplifies data analysis by enabling users to connect, clean, model, and visualize data—all within a single platform. Its intuitive drag-and-drop interface, AI-powered analytics, and real-time reporting capabilities make it accessible to both technical and non-technical users.

5.2 Key Features of Power BI:

1. Data Connectivity – Integrates with over 100 data sources, including Excel, SQL Server, Azure, Google Analytics, and more.
2. Power Query for Data Transformation – Cleans, shapes, and prepares data without complex coding.
3. Rich Visualizations – Provides bar charts, pie charts, maps, scatter plots, KPI indicators, and custom visuals.

4. DAX (Data Analysis Expressions) – Offers powerful formula-based calculations for advanced analytics.

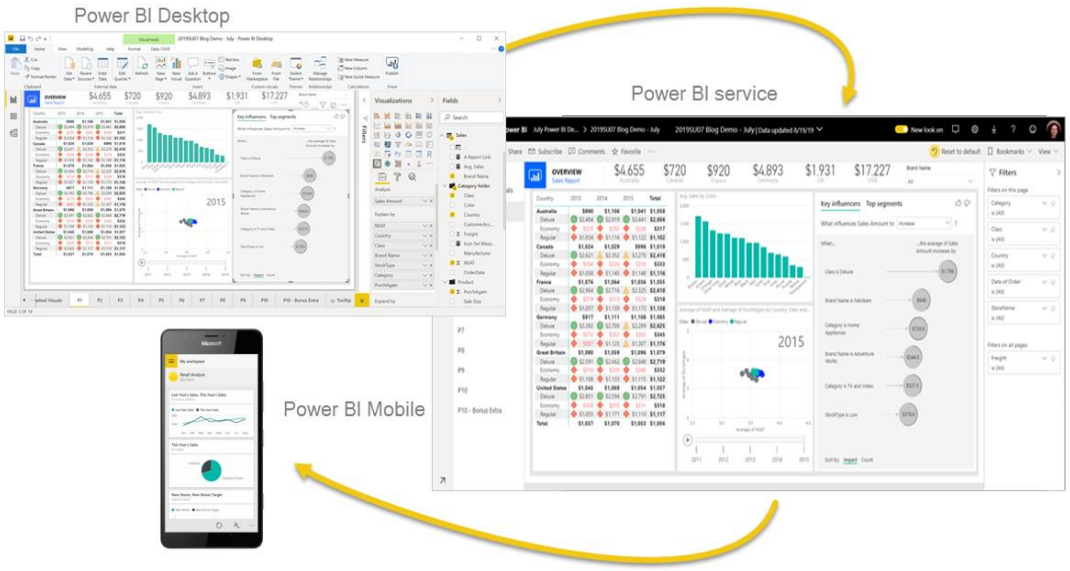
5. AI-Powered Insights – Uses machine learning for anomaly detection, forecasting, and sentiment analysis.
6. Real-Time Data Processing – Monitors key business metrics live with streaming data support.

7. Cloud & Mobile Accessibility – Enables report sharing and access from anywhere via Power BI Service and mobile apps.

8. Security & Governance – Ensures data security with role-based access, encryption, and compliance features.

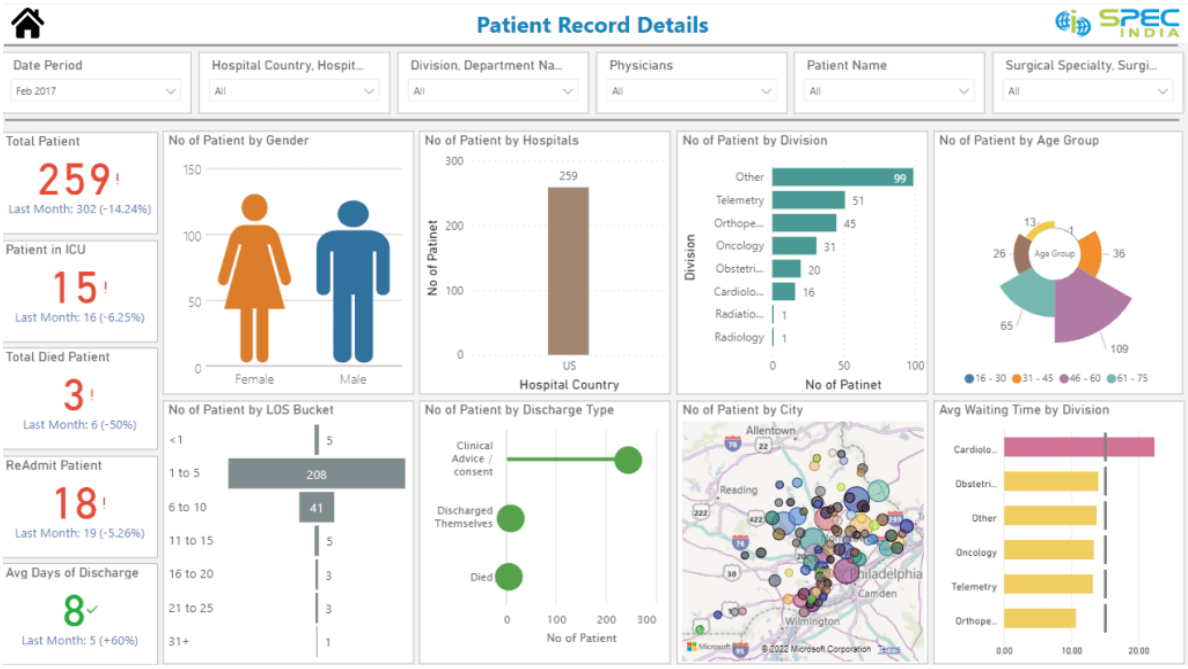
5.3 History :

The software was originally used as Power Pivot and Power Query in Microsoft Excel. This application was originally conceived by Thierry D'Hers and Amir Netz of the SQL Server Reporting Services team at Microsoft. It was originally designed by Ron George in the summer of 2010 and named Project Crescent Project Crescent was initially available for public download on 11 July 2011, bundled with SQL Server Codename Denali. Later renamed Power BI it was then unveiled by Microsoft in September 2013 as Power BI for Office 365. The first release of Power BI was based on the Microsoft Excel-based add-ins: Power Query, Power Pivot and Power View. With time, Microsoft also added many additional features like question and answers, enterprise-level data connectivity, and security options via Power BI Gateways. Power BI was first released to the general public on 24 July 2015. It has several versions for desktop, web, and mobile app.



5.4 Power BI Dashboards:

A Power BI dashboard is a single-page, interactive visual representation of data, providing key insights at a glance. It combines multiple reports, charts, and visualizations to help businesses monitor performance, track trends, and make data-driven decisions. Dashboards in Power BI are highly customizable, real-time, and shareable, making them an essential tool for business intelligence.

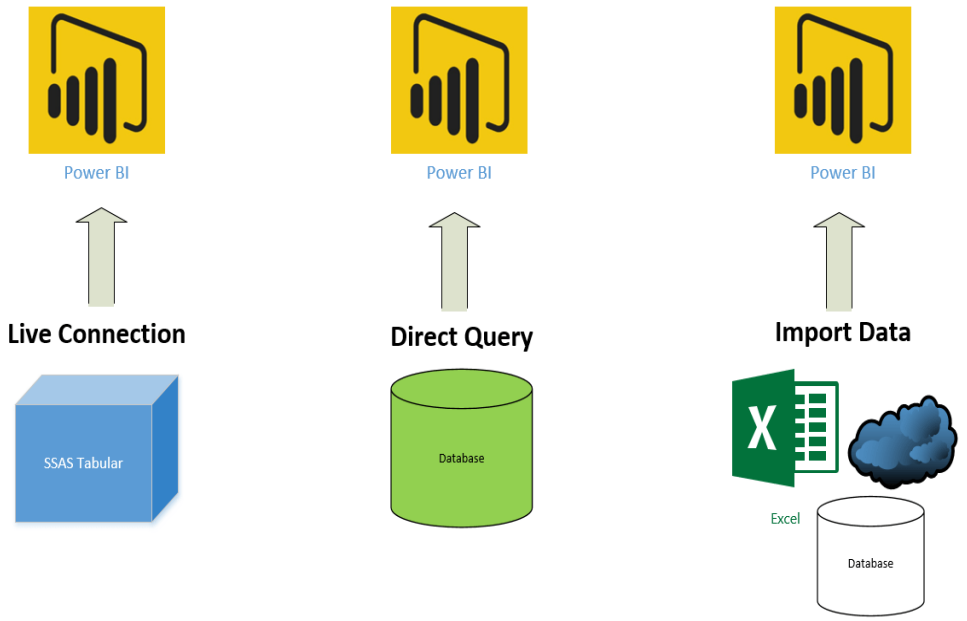


Types of Power BI Dashboards:

1. Executive Dashboards:
    - Used by business leaders to monitor KPIs, revenue, and company performance.
2. Sales & Marketing Dashboards:
    - Tracks customer behavior, lead conversion rates, sales performance, and campaign effectiveness.
3. Financial Dashboards:
    - Monitors profitability, expenses, budgeting, and revenue forecasts.
4. Operations & Supply Chain Dashboards:
    - Tracks inventory levels, logistics, supplier performance, and delivery status.
5. HR & Employee Dashboards:
    - Analyzes workforce demographics, employee performance, and attrition trends.

5.5 Power BI: Importing Data:

Power BI allows users to import data from various sources to create reports and dashboards. The data import process is crucial for building effective visualizations and performing analysis. Power BI supports multiple file formats, databases, and cloud services.

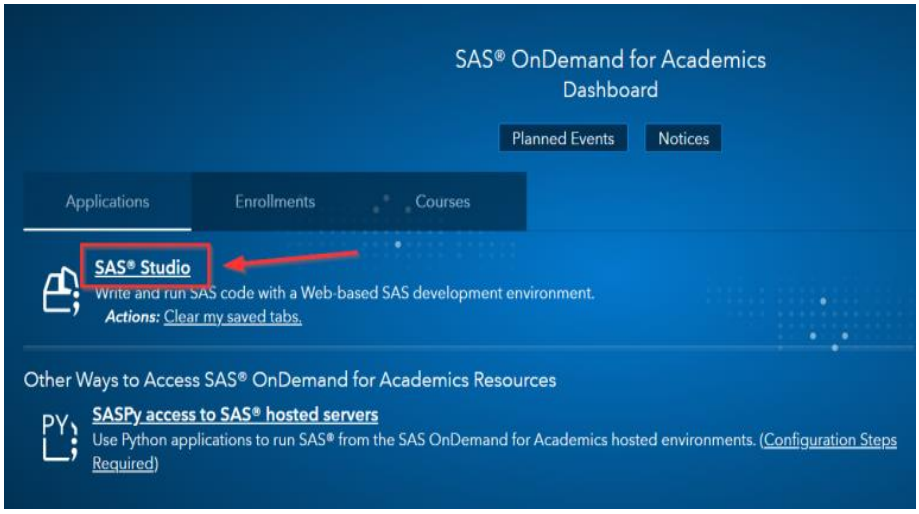


5.6 Power BI Visualization: Turning Data into Insights:

Power BI provides a wide range of visualization tools that help transform raw data into interactive charts, graphs, and reports. These visualizations enable businesses to track trends, compare data, and make informed decisions.

PROJECT USING INTERFACE

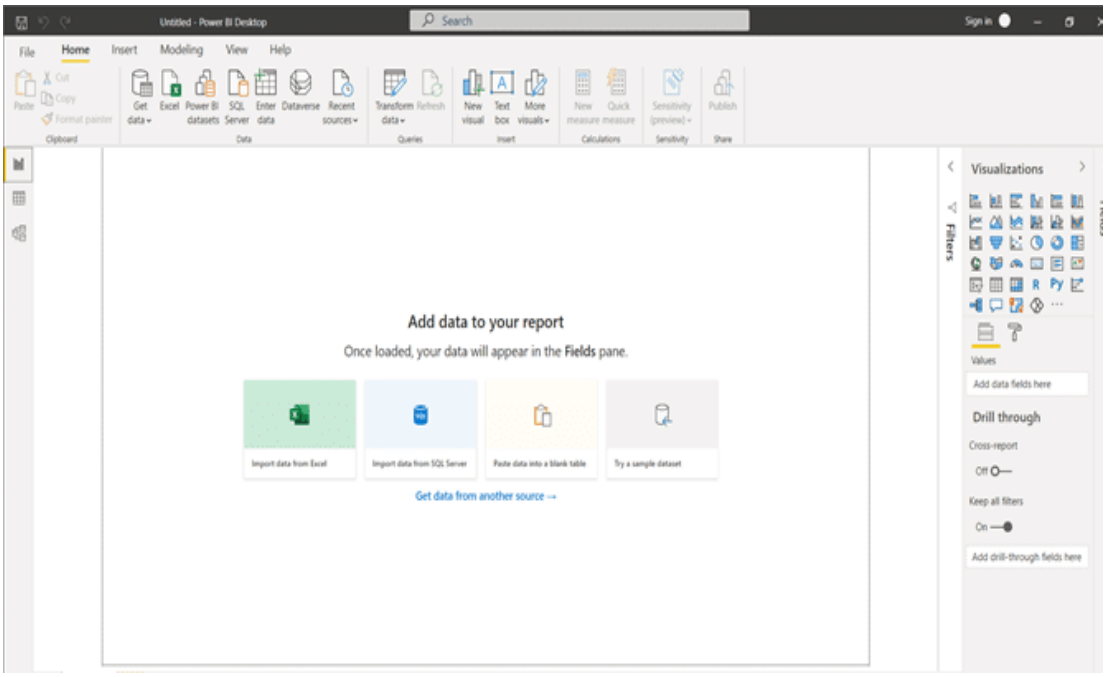
6.1 SAS Studio

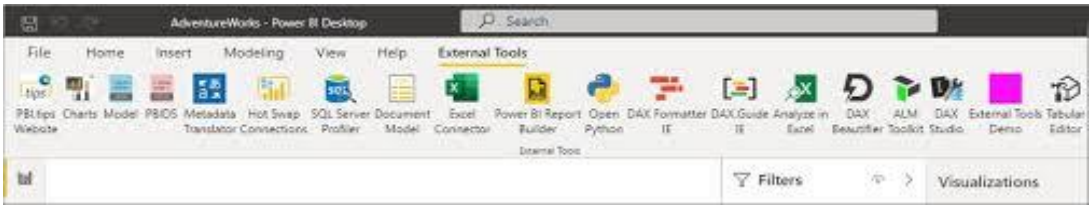


SAS Studio is a web-based interface for writing and executing SAS programs, making it a powerful tool for analyzing credit card intent data.

It provides capabilities for data preprocessing, statistical analysis, and visualization, helping to uncover insights into factors influencing credit card applications.

6.2 Power BI Desktop





Power BI is a powerful tool for data visualization and analysis, making it ideal for credit card intent analysis. It helps financial institutions identify patterns, trends, and key factors influencing loan approvals and defaults.

ABOUT THE DATTASET

7.1 Data set -: CREDIT CARD APPLICANT DATASET

The "Credit card Intent Analysis" dataset on Kaggle! This dataset provides essential information about loan applicants and their characteristics. Key variables include Age, Income, Home Ownership, Employment Length, Loan Intent, Amount, Interest Rate, Loan Status, and Credit Length.T he Credit Card Applicant Dataset contains 1,000 records with 13 key attributes related to credit card applications. It includes demographic, financial, and credit history factors, making it valuable for understanding loan approval trends and credit risk assessment.

ID : Unique identifier for each loan applicant.

Age: Age of the loan applicant.

Income: Income of the loan applicant.

Home: Home ownership status (Own, Mortgage, Rent).

Emp_Length: Employment length in years.

Intent: Purpose of the loan (e.g., education, home)

Amount: Loan amount applied for.

Rate: Interest rate on the loan.

Status: Loan approval status (Fully Paid, Charged Off)

Percent_Income: Loan amount as a percentage of income.

Default: Whether the applicant has defaulted on a loan previously (Yes, No).

Cred_Length: Length of the applicant's credit history.

Date: Shows the date of the applicant in dd/mm/yyy

7.2 Source of the Dataset:

Sources for credit card applicant datasets include:

1.Open-Source Datasets:

- **Kaggle** (https://www.kaggle.com) – Offers various financial datasets related to credit cards, loans, and customer intent.
- **UCI Machine Learning Repository** (https://archive.ics.uci.edu) – Provides datasets for credit scoring and loan approval.
- **Data.gov** (https://www.data.gov) – Contains public datasets, including financial and banking records.

2. Financial Institutions & Banks:

- Banks and credit card companies collect customer transaction data, which can be used for intent analysis.
- Credit bureaus (e.g., Experian, Equifax, TransUnion) may provide aggregated data for research purposes.

2. Company or Business Data:

- Internal customer records from financial service providers.
- CRM (Customer Relationship Management) systems tracking credit card applications.

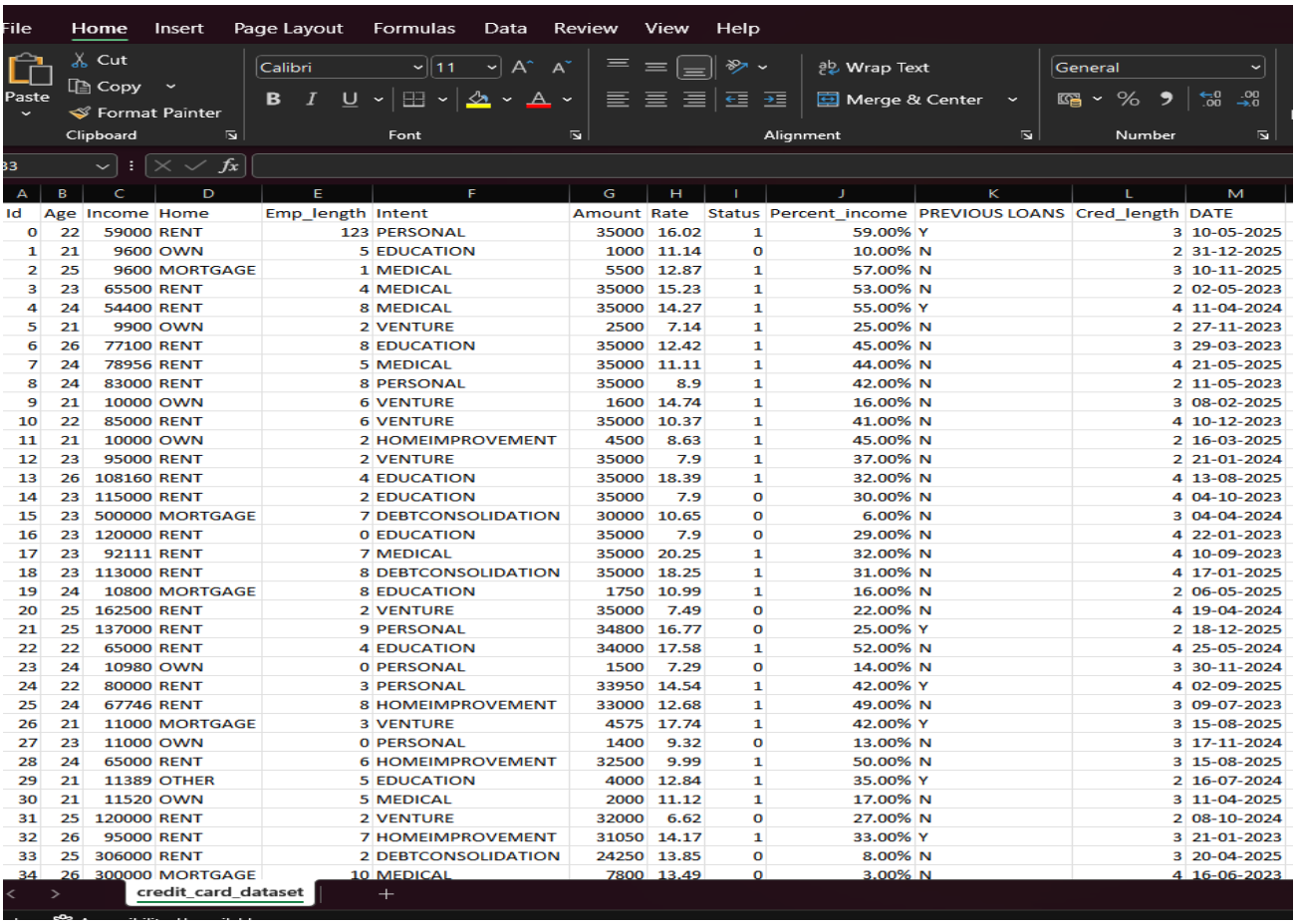
3. Web Scraping & APIs:

- Web scraping financial websites for trends in credit applications.
- APIs from financial institutions providing anonymized data (e.g., FRED Economic Data).

4. Simulated or Synthetic Data:

- If real data is unavailable, synthetic datasets can be created using tools like Python (Pandas, Faker) or SAS.

7.3 RAW DATA :



Id	Age	Income	Home	Emp_length	Intent	Amount	Rate	Status	Percent_income	PREVIOUS LOANS	Cred_length	DATE
0	22	59000	RENT	123	PERSONAL	35000	16.02	1	59.00%	Y	3	10-05-2025
1	21	9600	OWN	5	EDUCATION	1000	11.14	0	10.00%	N	2	31-12-2025
2	25	9600	MORTGAGE	1	MEDICAL	5500	12.87	1	57.00%	N	3	10-11-2025
3	23	65500	RENT	4	MEDICAL	35000	15.23	1	53.00%	N	2	02-05-2023
4	24	54400	RENT	8	MEDICAL	35000	14.27	1	55.00%	Y	4	11-04-2024
5	21	9900	OWN	2	VENTURE	2500	7.14	1	25.00%	N	2	27-11-2023
6	26	77100	RENT	8	EDUCATION	35000	12.42	1	45.00%	N	3	29-03-2023
7	24	78956	RENT	5	MEDICAL	35000	11.11	1	44.00%	N	4	21-05-2025
8	24	83000	RENT	8	PERSONAL	35000	8.9	1	42.00%	N	2	11-05-2023
9	21	10000	OWN	6	VENTURE	1600	14.74	1	16.00%	N	3	08-02-2025
10	22	85000	RENT	6	VENTURE	35000	10.37	1	41.00%	N	4	10-12-2023
11	21	10000	OWN	2	HOMEIMPROVEMENT	4500	8.63	1	45.00%	N	2	16-03-2025
12	23	95000	RENT	2	VENTURE	35000	7.9	1	37.00%	N	2	21-01-2024
13	26	108160	RENT	4	EDUCATION	35000	18.39	1	32.00%	N	4	13-08-2025
14	23	115000	RENT	2	EDUCATION	35000	7.9	0	30.00%	N	4	04-10-2023
15	23	500000	MORTGAGE	7	DEBTCONSOLIDATION	30000	10.65	0	6.00%	N	3	04-04-2024
16	23	120000	RENT	0	EDUCATION	35000	7.9	0	29.00%	N	4	22-01-2023
17	23	92111	RENT	7	MEDICAL	35000	20.25	1	32.00%	N	4	10-09-2023
18	23	113000	RENT	8	DEBTCONSOLIDATION	35000	18.25	1	31.00%	N	4	17-01-2025
19	24	10800	MORTGAGE	8	EDUCATION	1750	10.99	1	16.00%	N	2	06-05-2025
20	25	162500	RENT	2	VENTURE	35000	7.49	0	22.00%	N	4	19-04-2024
21	25	137000	RENT	9	PERSONAL	34800	16.77	0	25.00%	Y	2	18-12-2025
22	22	65000	RENT	4	EDUCATION	34000	17.58	1	52.00%	N	4	25-05-2024
23	24	10980	OWN	0	PERSONAL	1500	7.29	0	14.00%	N	3	30-11-2024
24	22	80000	RENT	3	PERSONAL	33950	14.54	1	42.00%	Y	4	02-09-2025
25	24	67746	RENT	8	HOMEIMPROVEMENT	33000	12.68	1	49.00%	N	3	09-07-2023
26	21	11000	MORTGAGE	3	VENTURE	4575	17.74	1	42.00%	Y	3	15-08-2025
27	23	11000	OWN	0	PERSONAL	1400	9.32	0	13.00%	N	3	17-11-2024
28	24	65000	RENT	6	HOMEIMPROVEMENT	32500	9.99	1	50.00%	N	3	15-08-2025
29	21	11389	OTHER	5	EDUCATION	4000	12.84	1	35.00%	Y	2	16-07-2024
30	21	11520	OWN	5	MEDICAL	2000	11.12	1	17.00%	N	3	11-04-2025
31	25	120000	RENT	2	VENTURE	32000	6.62	0	27.00%	N	2	08-10-2024
32	26	95000	RENT	7	HOMEIMPROVEMENT	31050	14.17	1	33.00%	Y	3	21-01-2023
33	25	306000	RENT	2	DEBTCONSOLIDATION	24250	13.85	0	8.00%	N	3	20-04-2025
34	26	300000	MORTGAGE	10	MEDICAL	7800	13.49	0	3.00%	N	4	16-06-2023

Raw data refers to unprocessed, unstructured, and original data collected from a source before undergoing any cleaning, transformation, or analysis. It is the first-hand data that has not been manipulated, formatted, or organized into a meaningful structure. Raw data serves as the foundation for data processing, where it is refined to extract insights and make informed decisions.

STEPS TO ANALYZE THE DATASET AND VISUALIZATION

i. CREATING LIBRARIES

ii. IMPORT DATA

iii. EXPLORE THE DATASET

iv. UNDERSTANDING THE DATA

v. PRE-PROCESSING THE DATA

vi. DATA FILTERING AND CONDITIONAL QUERIES

vii. CREDIT INDENT DATA ANALYSIS & REPORTS

viii. POWERBI VISUALIZATION

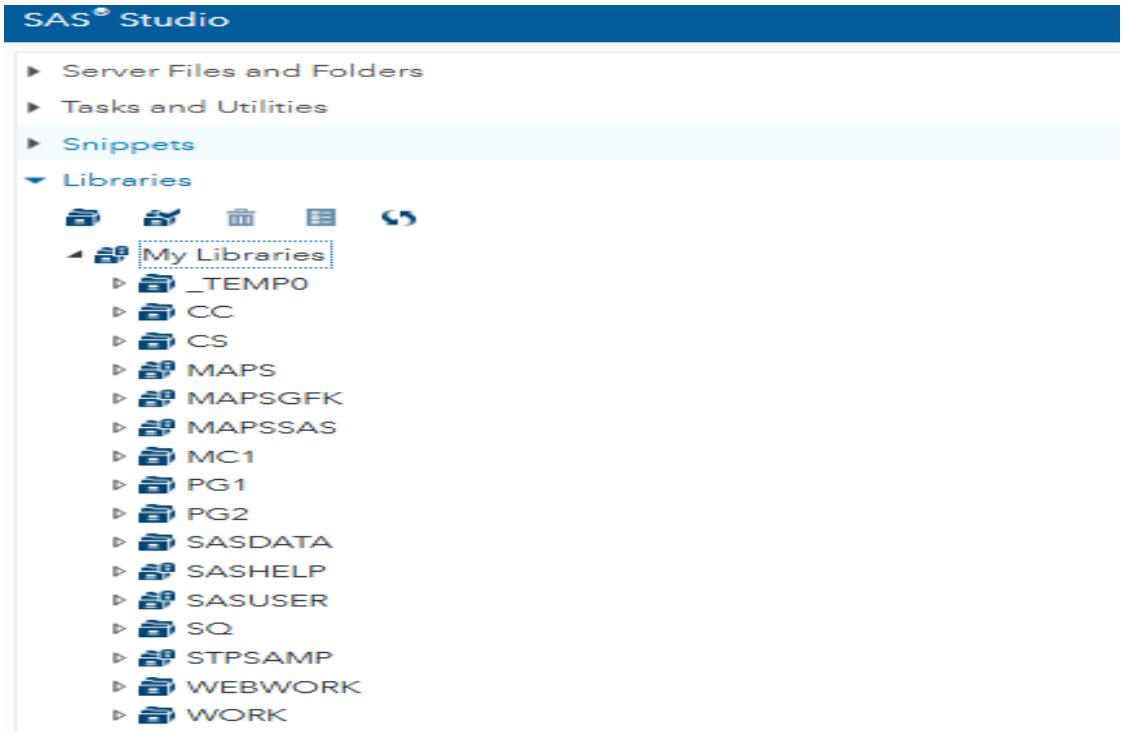
8.1 CREATING LIBRARIES:

In SAS, the LIBNAME statement is used to assign a library reference (libref) to a storage location where SAS datasets are stored. This allows you to reference datasets within that location without specifying the full path each time.

Code-:

```
/*CREATING LIBRARY */
LIBNAME CC "/home/u62467441/credit card project";
```

- **CC** → Library reference (libref) for the datasets in the specified location.
- **base** → Specifies that the library is a folder containing SAS datasets.
- "/home/u62467441/credit card project" → The physical path where SAS datasets are stored**.**



8.2 IMPORT DATA:

SAS allows importing data from various sources like CSV, Excel, text files, and databases using different methods.

1.Importing a CSV File

Use PROC IMPORT to load a **CSV file**.

Example -: SAS Code

```
/*IMPORT DATA*/

PROC IMPORT DATAFILE="/home/u62467441/credit card project/credit_card_dataset.csv"
   OUT=CC.credit_card_data
   DBMS=CSV
   REPLACE;
   GETNAMES=YES;
RUN;
```

- DATAFILE="/home/u62467441/credit card project/credit_card_dataset.csv" → Specifies the file location.
- OUT= CC.credit_card_data → Saves the imported data as a SAS dataset.
- DBMS=CSV → Defines the file type.
- REPLACE → Overwrites the dataset if it exists.

PROC IMPORT Statement:

This **procedure (PROC IMPORT)** is used to read external files and convert them into a SAS dataset.

DATAFILE Option

Datafile="/home/u62467441/credit card project/credit_card_dataset.csv"

- Specifies the **file path** of the **CSV file** being imported.
- The file is located in the **home directory** of the user ("/home/u62467441/credit card

project").

- The file name is “credit_card_dataset.csv”.

DBMS=CSV

- **Specifies the file type** (CSV format in this case).
- Other formats include **XLSX** (Excel), **TAB** (tab-delimited text files), etc.
- To load a CSV file into SAS, you can use the PROC IMPORT statement with DBMS=CSV. This allows SAS to recognize the file structure and automatically assign variable types

OUT = CC.credit_card_data

- Defines the **output SAS dataset** where the imported data will be stored.

- mp is likely a **library reference (LIBNAME statement required)**, meaning the dataset is stored in a specific library (folder).

REPLACE Option

- In **Base SAS**, the REPLACE option is used to **overwrite an existing dataset or file** with new data. It ensures that if a dataset or file with the same name already exists, it will be replaced instead of causing an error.

- If the dataset **CC.CREDIT_CARD_DATA** already exists, this option **overwrites it** with the new data.

GETNAMES = YES;

- **Automatically detects column data types** by scanning the **maximum number of rows** in the file.
- Helps in correctly identifying **numeric and character variables**.
- By default, SAS scans the first 20 rows, but max ensures **all rows** are considered.

OUTPUT DATA = CC.CREDIT_CARD_DATA;

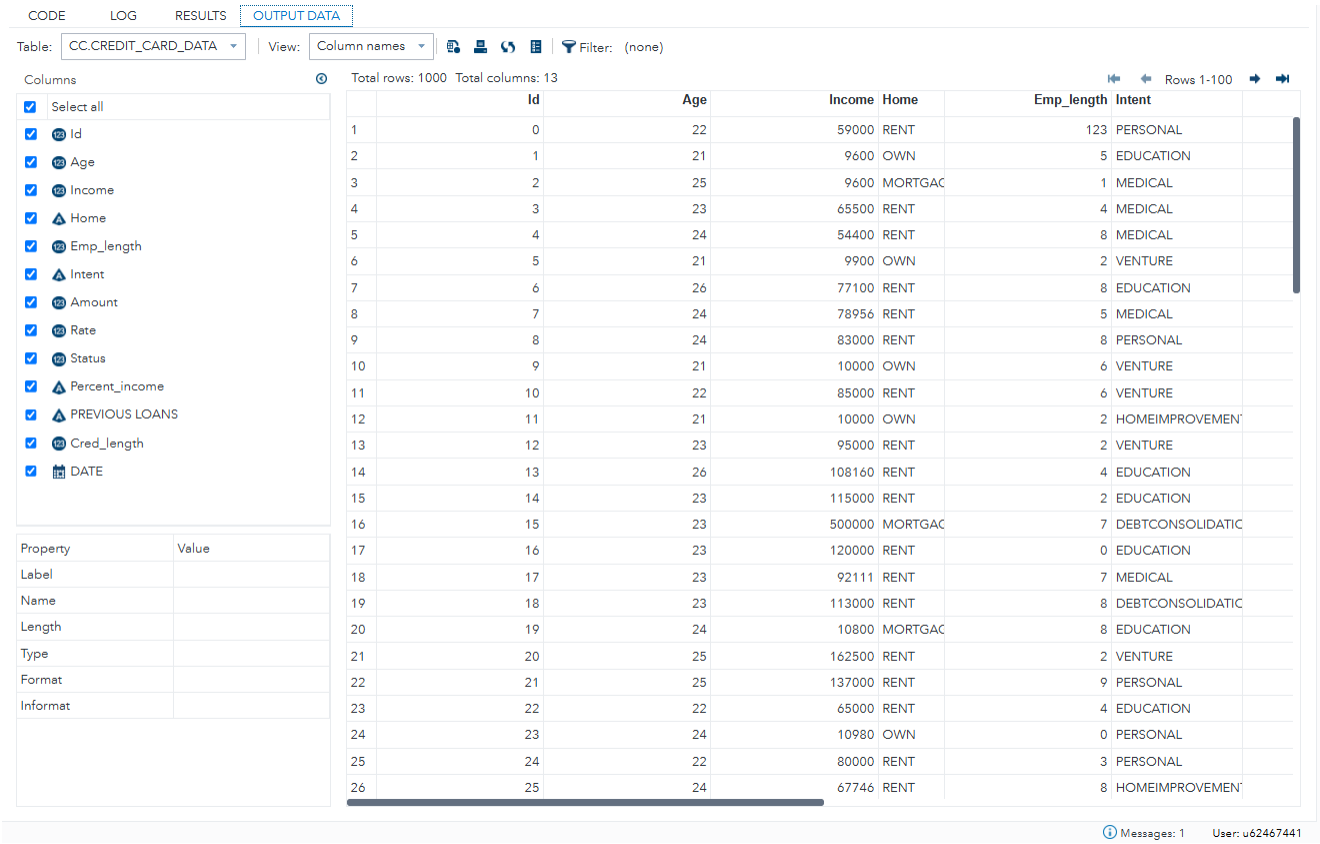

CODE LOG RESULTS OUTPUT DATA

Table: CC.CREDIT_CARD_DATA | View: Column names | Filter: (none)

Columns

Select all
Id
Age
Income
Home
Emp_length
Intent
Amount
Rate
Status
Percent_income
PREVIOUS LOANS
Cred_length
DATE

Property	Value
Label	
Name	
Length	
Type	
Format	
Informat	

Total rows: 1000 Total columns: 13 Rows 1-100

	Id	Age	Income	Home	Emp_length	Intent
1	0	22	59000	RENT	123	PERSONAL
2	1	21	9600	OWN	5	EDUCATION
3	2	25	9600	MORTGAC	1	MEDICAL
4	3	23	65500	RENT	4	MEDICAL
5	4	24	54400	RENT	8	MEDICAL
6	5	21	9900	OWN	2	VENTURE
7	6	26	77100	RENT	8	EDUCATION
8	7	24	78956	RENT	5	MEDICAL
9	8	24	83000	RENT	8	PERSONAL
10	9	21	10000	OWN	6	VENTURE
11	10	22	85000	RENT	6	VENTURE
12	11	21	10000	OWN	2	HOMEIMPROVEMEN
13	12	23	95000	RENT	2	VENTURE
14	13	26	108160	RENT	4	EDUCATION
15	14	23	115000	RENT	2	EDUCATION
16	15	23	500000	MORTGAC	7	DEBTCONSOLIDATIC
17	16	23	120000	RENT	0	EDUCATION
18	17	23	92111	RENT	7	MEDICAL
19	18	23	113000	RENT	8	DEBTCONSOLIDATIC
20	19	24	10800	MORTGAC	8	EDUCATION
21	20	25	162500	RENT	2	VENTURE
22	21	25	137000	RENT	9	PERSONAL
23	22	22	65000	RENT	4	EDUCATION
24	23	24	10980	OWN	0	PERSONAL
25	24	22	80000	RENT	3	PERSONAL
26	25	24	67746	RENT	8	HOMEIMPROVEMEN

Messages: 1 User: u62467441

8.3 EXPLORE THE DATASET:

In SAS, **exploring a dataset** refers to the process of examining and analyzing the structure, contents, and summary statistics of a dataset before performing further analysis.

This step helps in understanding data quality, identifying missing values, checking distributions, and detecting anomalies

Exploring a dataset in SAS means analyzing its structure and contents to understand the data before performing further analysis. Here are some key methods used in SAS to explore a dataset:

1. **PROC CONTENTS** – Provides information about the dataset, including variable names, types, formats, and dataset properties.
2. **PROC PRINT** – Displays rows and columns of the dataset to visually inspect the data.

PROC PRINT –

This procedure is used to display the contents of a dataset in a tabular format. It allows you to inspect data values directly.

Code-:

```
proc print data=CC.credit_card_data (OBS=10);
run;
```

 DATA=CC.credit_card_data – Specifies the dataset to be printed

Obs	Id	Age	Income	Home	Emp_length	Intent	Amount	Rate	Status	Percent_income	PREVIOUS LOANS	Cred_length	DATE
1	0	22	59000	RENT	123	PERSONAL	35000	16.02	1	59.00%	Y	3	10/05/2025
2	1	21	9600	OWN	5	EDUCATION	1000	11.14	0	10.00%	N	2	31/12/2025
3	2	25	9600	MORTGAGE	1	MEDICAL	5500	12.87	1	57.00%	N	3	10/11/2025
4	3	23	65500	RENT	4	MEDICAL	35000	15.23	1	53.00%	N	2	02/05/2023
5	4	24	54400	RENT	8	MEDICAL	35000	14.27	1	55.00%	Y	4	11/04/2024
6	5	21	9900	OWN	2	VENTURE	2500	7.14	1	25.00%	N	2	27/11/2023
7	6	26	77100	RENT	8	EDUCATION	35000	12.42	1	45.00%	N	3	29/03/2023
8	7	24	78956	RENT	5	MEDICAL	35000	11.11	1	44.00%	N	4	21/05/2025
9	8	24	83000	RENT	8	PERSONAL	35000	8.9	1	42.00%	N	2	11/05/2023
10	9	21	10000	OWN	6	VENTURE	1600	14.74	1	16.00%	N	3	08/02/2025

This code is useful for quickly viewing a sample of the dataset to check its structure and ensure the data is correctly loaded before performing further analysis.

PROC CONTENTS –

This procedure provides metadata (structural details) about a dataset. It does not display the actual data but gives information about the dataset structure.

Code-:

```
proc contents data=CC.credit_card_data;
run;
```

Results: credit_code.sas

Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Informat
10	Percent_income	Char	6	$6.	$6.
8	Rate	Num	8	BEST12.	BEST32.
9	Status	Num	8	BEST12.	BEST32.

The CONTENTS Procedure

Data Set Name	CC.CREDIT_CARD_DATA	Observations	1000
Member Type	DATA	Variables	13
Engine	V9	Indexes	0
Created	03/11/2025 23:01:30	Observation Length	104
Last Modified	03/11/2025 23:01:30	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Engine/Host Dependent Information	
Data Set Page Size	131072
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	1258
Obs in First Data Page	1000
Number of Data Set Repairs	0
Filename	/home/u62467441/credit card project/credit_card_data.sas7bdat
Release Created	9.0401M7
Host Created	Linux
Inode Number	14966475922
Access Permission	rw-r--r--
Owner Name	u62467441
File Size	256KB
File Size (bytes)	262144

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat
2	Age	Num	8	BEST12.	BEST32.
7	Amount	Num	8	BEST12.	BEST32.
12	Cred_length	Num	8	BEST12.	BEST32.
13	DATE	Num	8	DDMMYY10.	DDMMYY10.
5	Emp_length	Num	8	BEST12.	BEST32.
4	Home	Char	8	$8.	$8.
1	Id	Num	8	BEST12.	BEST32.
3	Income	Num	8	BEST12.	BEST32.
6	Intent	Char	17	$17.	$17.
11	PREVIOUS LOANS	Char	1	$1.	$1.

Output Information:

- Dataset name and location
- Number of observations (rows) and variables (columns)
- Variable names, types (numeric or character), and formats
- Storage engine and dataset creation/modification date

This code helps in understanding the dataset's structure before performing analysis, ensuring the correct variables and data types are used.

8.4 UNDERSTANDING THE DATA :

Understanding the data in SAS refers to the process of examining and analyzing a dataset to gain insights into its structure, contents, and quality before performing further analysis.

This step ensures that the data is accurate, complete, and suitable for analysis.

 Ensures data quality before analysis.

 Helps in detecting errors, missing values, and outliers.

 Guides data cleaning, transformation, and preparation.

 Improves the accuracy of statistical modeling and decision-making.

Dictionary.columns :

Dictionary.columns is a metadata table that contains details about variables (columns) in SAS datasets, such as their names, types, lengths, formats, labels, and associated libraries.

Code :

```
proc sql ;
describe table dictionary.columns;
select name, type, length
from dictionary.columns
where libname="CC";
quit;
```

Column Name	Column Type	Column Length
Id	num	8
Age	num	8
Income	num	8
Home	char	8
Emp_length	num	8
Intent	char	17
Amount	num	8
Rate	num	8
Status	num	8
Percent_income	char	6
PREVIOUS LOANS	char	1
Cred_length	num	8
DATE	num	8

 NAME – The variable (column) name.

 TYPE – The type of the variable (Character or Numeric).

 LENGTH – The storage length of the variable.

DISTINCT:

SELECT DISTINCT Rate, Home, Intent –

- SELECT specifies the columns (Rate, Home, Intent) to retrieve.
-
- DISTINCT ensures that duplicate rows are removed, displaying only unique combinations.

Code-:

```
proc sql number outobs=20;
select distinct Rate,Home,Intent
From CC.credit_card_data;
quit;
```

Row	Rate	Home	Intent
1	.	MORTGAGE	DEBTCONSOLIDATION
2	.	MORTGAGE	EDUCATION
3	.	MORTGAGE	HOMEIMPROVEMENT
4	.	MORTGAGE	MEDICAL
5	.	MORTGAGE	PERSONAL
6	.	MORTGAGE	VENTURE
7	.	OWN	DEBTCONSOLIDATION
8	.	OWN	EDUCATION
9	.	OWN	HOMEIMPROVEMENT
10	.	OWN	MEDICAL
11	.	OWN	PERSONAL
12	.	OWN	VENTURE
13	.	RENT	DEBTCONSOLIDATION
14	.	RENT	EDUCATION
15	.	RENT	HOMEIMPROVEMENT
16	.	RENT	MEDICAL
17	.	RENT	PERSONAL
18	.	RENT	VENTURE
19	5.42	MORTGAGE	EDUCATION
20	5.42	MORTGAGE	MEDICAL

8.5 PRE-PROCESSING THE DATA:

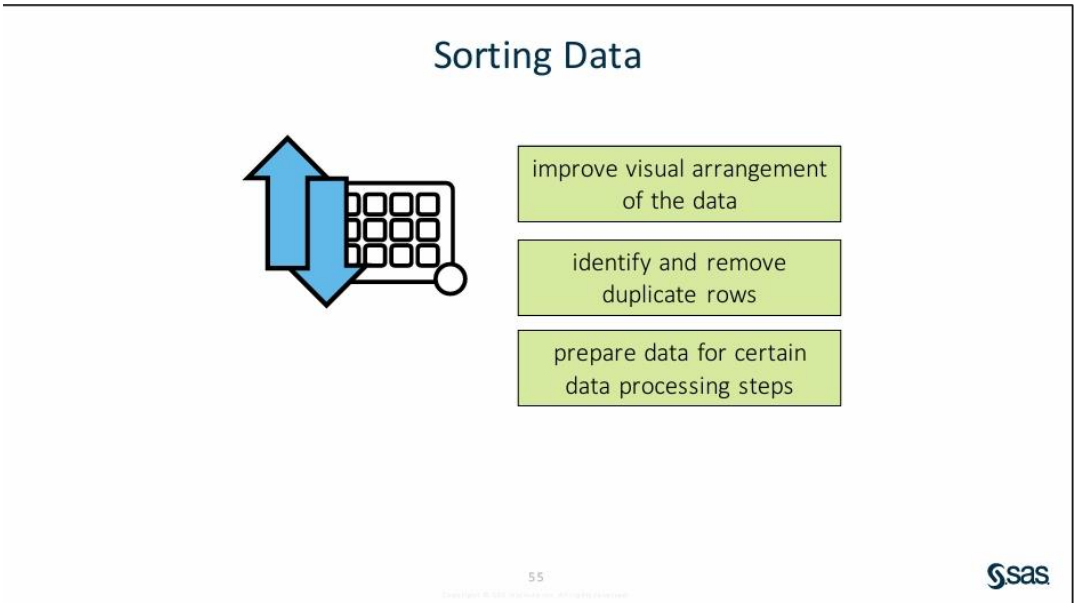
Pre-processing the data in SAS refers to the steps taken to clean, transform, and prepare raw data for analysis or modeling. This process ensures that the dataset is structured, complete, and free of errors before performing statistical analysis or machine learning.

SORTING

Sorting in SAS refers to the process of arranging data in ascending or descending order based on one or more variables. This helps in data organization, efficient retrieval, and better analysis.

Code-:

```
PROC SORT DATA=CC.credit_card_data OUT=sorted_data;
   BY DESCENDING Income;
RUN;
```



REPLACE MISSING VALUES

Replacing missing values in SAS is a data pre-processing step where missing or null values in a dataset are handled by assigning appropriate replacements. This is essential to ensure data completeness and accuracy before analysis.

Code :

```
DATA credit_data_clean;
   SET CC.credit_card_data;
   IF MISSING(Rate) THEN Rate = 0;
RUN;
```

Rate
15.23
10.65
7.9
15.96
11.48
0
7.29
0
5.99
11.58
0
15.7
15.99
14.84
10.99

- Rate column missing values where replaced with 0

8.6 DATA FILTERING & CONDITIONAL QUERIES:

1.Code:-

/*display only the customers who have an Income greater than $3,00,000*/

```
PROC SQL;
   SELECT *
   FROM CC.credit_card_data
   WHERE Income > 300000;
QUIT;
```

Output:

Id	Age	Income	Home	Emp_length	Intent	Amount	Rate	Status	Percent_income	PREVIOUS LOANS	Cred_length	DATE
15	23	500000	MORTGAGE	7	DEBTCONSOLIDATION	30000	10.65	0	6.00%	N	3	04/04/2024
33	25	306000	RENT	2	DEBTCONSOLIDATION	24250	13.85	0	8.00%	N	3	20/04/2025

 Filtering Data – The WHERE clause helps in selecting only relevant rows based on a condition.
 Comparison Operators (>) – The > operator is used to filter values greater than a specific threshold.

This output helps in segmenting high-income customers for further analysis, such as credit card offers, risk assessment, or marketing campaigns.

1.Code:-

/*count the number of Fully Paid vs. Charged Off loans*/

```
PROC SQL;
   SELECT Status, COUNT(*) AS Loan_Count
   FROM CC.credit_card_data
   GROUP BY Status;
QUIT;
```

Output:

Status	Loan_Count
0	447
1	553

 Aggregating Data – Uses COUNT(*) to count the number of loans in each status category.

 Grouping Data (GROUP BY) – Ensures that counts are calculated for each unique Status value (e.g., "Fully Paid" vs. "Charged Off").

 Labeling Output (AS Loan_Count) – Renames the column for better readability in results.

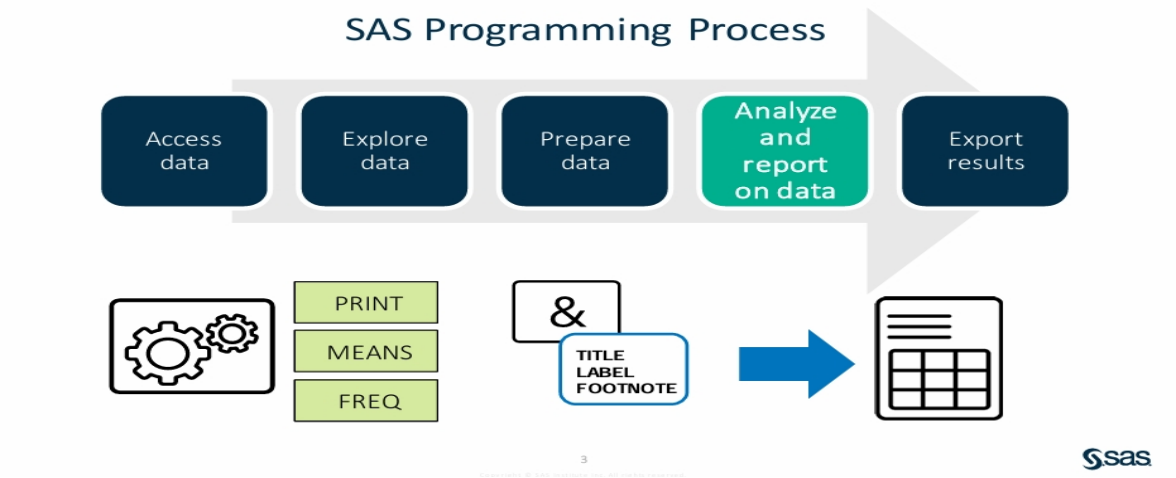
This query helps in analyzing loan performance by comparing the number of **fully paid loans = 1** vs. **charged-off loans = 0** which can be useful for risk assessment and financial decision-making.

8.7 CREDIT INDENT DATA ANALYSIS:

Credit Intent Data Analysis refers to the process of analyzing customer behavior, financial details, and other factors to determine their intent regarding credit applications.

This analysis helps financial institutions, banks, and lenders understand whether a customer is likely to apply for a credit card, take a loan, or default on payments.

- Identify potential credit card or loan applicants.
- Understand customer financial behavior.
- Assess risk and predict defaults.
- Segment customers based on their intent and creditworthiness.
- Improve marketing strategies for targeted offers.



CODE:

```
/* 1.avg of income and rate */
            TITLE "avg of income and rate";
            PROC SQL;
               SELECT AVG(Income) AS Avg_Income,
                    AVG(Rate) AS Avg_Rate
               FROM cc.credit_card_data;
            QUIT;
            title;
```

The given SAS code calculates the **average (mean) of Income and Rate** from the dataset cc.credit_card_data using SQL within SAS. The TITLE statement sets a heading for the output.

The PROC SQL procedure allows structured queryexecution. Inside the SELECT statement, AVG(Income) AS Avg_Income calculates the average income and renames the result as Avg_Income, while AVG(Rate) AS Avg_Rate does the same for the Rate variable.

The FROM clause specifies the dataset cc.credit_card_data, which is stored in the CC library. Finally, the QUIT; statement ends the SQL procedure, and TITLE; clears the title after execution.

This code helps in understanding the overall income and rate trends in the dataset, which can be useful for financial analysis and decision-making.

CODE :

```
 /* 2.max of amount */
TITLE "max of amount";
PROC MEANS DATA=cc.credit_card_data MAX;
   VAR Amount;
RUN;
TITLE;
```

The given SAS code calculates the **maximum value of the "Amount" variable** from the dataset cc.credit_card_data using the PROC MEANS procedure.

It starts by setting a title, "max of amount", which will be displayed in the output.

The PROC MEANS statement specifies the dataset to be analyzed and includes the MAX option,

which tells SAS to compute only the maximum value of the selected variable.

The VAR Amount; statement ensures that the calculation is performed specifically on the **"Amount"** column.

After executing the procedure with RUN;, the final TITLE; statement clears the title to avoid carrying it over to future outputs.

This analysis helps in identifying the highest transaction or loan amount in the dataset, which is crucial for financial insights, risk assessment, and decision-making.

CODE:

```
/* 3.max of amount by home*/

        TITLE "max of amount by home";
        PROC SQL;
           SELECT Home, MAX(Amount) AS Max_Amount
           FROM cc.credit_card_data
           GROUP BY Home;
        QUIT;
        TITLE;
```

The given SAS code calculates the maximum transaction or loan amount for each "Home" category in the dataset cc.credit_card_data using PROC SQL.

It begins with the TITLE statement, which sets the title "max of amount by home" to describe the output. The PROC SQL procedure is then used to perform the SQL query.

The SELECT statement retrieves two columns: Home, which represents different homeownership statuses, and MAX(Amount) AS Max_Amount, which calculates the highest amount for each home category and renames it as Max_Amount.

The FROM clause specifies the dataset being analyzed.

The GROUP BY Home statement ensures that the maximum amount is calculated separately for each unique value in the "Home" column. Finally, QUIT; ends the SQL procedure, and TITLE; clears the title after execution.

This analysis helps in understanding how transaction or loan amounts vary by homeownership status, which can be useful for financial decision-making and risk assessment.

CODE:

```
/* 4.sum of amount by intent*/

    TITLE "sum of amount by intent";
    PROC SQL;
      SELECT Intent, SUM(Amount) AS Total_Amount
      FROM cc.credit_card_data
      GROUP BY Intent;
    QUIT;
    TITLE;
```

The given SAS code calculates the **total transaction or loan amount for each "Intent" category** in the dataset cc.credit_card_data using PROC SQL.

It begins with the TITLE statement, which sets the heading **"sum of amount by intent"** to describe the output. The PROC SQL procedure is used to execute an SQL query.

The SELECT statement retrieves two columns: **Intent**, which represents different purposes for credit card usage or loan applications, and SUM(Amount) AS Total_Amount, which calculates the total amount for each intent category and renames it as Total_Amount.

The FROM clause specifies the dataset being analyzed. The GROUP BY Intent statement ensures that the sum calculation is performed separately for each unique value in the "Intent" column. Finally, QUIT; ends the SQL procedure, and TITLE; clears the title after execution.

This analysis helps in understanding the total financial distribution across different credit intents, which can be valuable for business strategy, risk management, and financial planning

CODE:

```
/* 5.min of rate by home*/
    TITLE "min of rate by home";
    PROC MEANS DATA=cc.credit_card_data MIN;
      CLASS Home;
      VAR Rate;
    RUN;
    TITLE;
```

The given SAS code finds the **minimum interest rate for each homeownership category** in the dataset cc.credit_card_data.

The PROC MEANS procedure calculates summary statistics, and the MIN option ensures that only the lowest rate is retrieved. The CLASS Home; statement groups the data by homeownership type, so the minimum rate is calculated separately for each category.

The VAR Rate; statement specifies that the calculation is done on the **"Rate"** variable. Finally, RUN; executes the procedure, and TITLE; clears the title.

This helps in understanding the lowest interest rate for different homeownership types.

CODE:

```
/* 6.avg of income by age*/
        TITLE "avg of income by age";
        PROC MEANS DATA=cc.credit_card_data MEAN;
           CLASS Age;
           VAR Income;
        RUN;
        TITLE;
```

The given SAS code calculates the **average income for each age group** in the dataset cc.credit_card_data.

The PROC MEANS procedure is used to compute statistical summaries, and the MEAN option ensures that the average (mean) income is calculated.

The CLASS Age; statement groups the data by the **"Age"** variable, so the average income is calculated separately for each age group. The VAR Income; statement specifies that the calculation is performed on the **"Income"** variable.

Finally, RUN; executes the procedure, and TITLE; clears the title. This helps analyze income distribution across different age groups.

CODE:

```
/* 7.count of id by card_length*/
TITLE "count of id by card_length";
PROC SQL;
   SELECT Cred_Length, COUNT(ID) AS ID_Count
   FROM cc.credit_card_data
   GROUP BY Cred_Length;
QUIT;
TITLE;
```

The given SAS code calculates the **number of IDs for each credit card length category** in the dataset cc.credit_card_data

. The PROC SQL procedure is used to execute an SQL query. The SELECT statement retrieves two columns: **Cred_Length**, which represents the length of time a credit card has been held, and COUNT(ID) AS ID_Count, which counts the number of records (IDs) for each credit length category.

The GROUP BY Cred_Length; statement ensures that the count is calculated separately for each unique credit length. Finally, QUIT; ends the SQL procedure, and TITLE; clears the title.

This analysis helps understand how many customers fall into different credit card length categories.

CODE:

```
/* 8.avg of income by years*/
TITLE "avg of income by years";
PROC SQL OUTOBS=20;
   SELECT DATE, AVG(Income) AS Avg_Income
   FROM cc.credit_card_data
   GROUP BY DATE;
QUIT;
TITLE;
```

The given SAS code calculates the **average income for each year** in the dataset cc.credit_card_data using PROC SQL. The TITLE statement sets the heading **"avg of income by years"** to describe the output.

The SELECT statement retrieves two columns: **DATE**, representing the year or date of the record, and AVG(Income) AS Avg_Income, which calculates the average income for each year and

renames it as Avg_Income.

The GROUP BY DATE; statement ensures that the calculation is done separately for each unique year. The OUTOBS=20 option limits the output to **20 rows**. Finally, QUIT; ends the SQL procedure, and TITLE; clears the title.

This analysis helps understand **how income trends vary over different years**.

CODE:

```
/* 9.univariate of income */
TITLE "univariate of income";
PROC UNIVARIATE DATA=cc.credit_card_data;
   VAR Income;
RUN;
TITLE;
```

The given SAS code performs a **univariate analysis of the "Income" variable** in the dataset cc.credit_card_data using the PROC UNIVARIATE procedure.

The TITLE statement sets the heading **"univariate of income"** to describe the output. The PROC UNIVARIATE procedure provides detailed statistical summaries, including measures like **mean, median, standard deviation, skewness, and percentiles** for the selected variable.

The VAR Income; statement specifies that the analysis is conducted on the **"Income"** column. Finally, RUN; executes the procedure, and TITLE; clears the title.

This analysis helps in understanding **the distribution, spread, and central tendency of income** in the dataset.

OVERALL SAS CODE

```
/*CREATING LIBRARY */
LIBNAME CC "/home/u62467441/credit card project";

/*IMPORT DATA*/
PROC IMPORT DATAFILE="/home/u62467441/credit card project/credit_card_dataset.csv"
   OUT=CC.credit_card_data
   DBMS=CSV
   REPLACE;
   GETNAMES=YES;
RUN;

/*Reading the Data*/
proc print data=CC.credit_card_data (OBS=10);
run;
proc contents data=CC.credit_card_data;
run;

/*Understanding the Data*/
proc sql ;
describe table dictionary.columns;
select name, type, length
from dictionary.columns
where libname="CC";
quit;

proc sql number outobs=20;
select distinct Rate,Home,Intent
From CC.credit_card_data;
quit;


/*Preprocessing the Data*/
PROC SORT DATA=CC.credit_card_data OUT=sorted_data;
   BY DESCENDING Income;
RUN;

 DATA credit_data_clean;
   SET CC.credit_card_data;
   IF MISSING(Rate) THEN Rate = 0;
RUN;

/*Data Filtering & Conditional Queries*/
/*display only the customers who have an Income greater than $2,00,000*/
PROC SQL;
```

```
   SELECT *
   FROM CC.credit_card_data
   WHERE Income > 300000;
QUIT;

/*count the number of Fully Paid vs. Charged Off loans*/
PROC SQL;
   SELECT Status, COUNT(*) AS Loan_Count
   FROM CC.credit_card_data
   GROUP BY Status;
QUIT;

/*CREDIT INDENT DATA ANALYSIS*/

/* 1.avg of income and rate */
TITLE "avg of income and rate";
PROC SQL;
   SELECT AVG(Income) AS Avg_Income,
        AVG(Rate) AS Avg_Rate
   FROM cc.credit_card_data;
QUIT;
title;

/* 2.max of amount */
TITLE "max of amount";
PROC MEANS DATA=cc.credit_card_data MAX;
   VAR Amount;
RUN;
TITLE;

/* 3.max of amount by home*/
TITLE "max of amount by home";
PROC SQL;
   SELECT Home, MAX(Amount) AS Max_Amount
   FROM cc.credit_card_data
   GROUP BY Home;
QUIT;
TITLE;

/* 4.sum of amount by intent*/
TITLE "sum of amount by intent";
PROC SQL;
   SELECT Intent, SUM(Amount) AS Total_Amount
   FROM cc.credit_card_data
   GROUP BY Intent;
QUIT;
```

```
TITLE;

/* 5.min of rate by home*/
TITLE "min of rate by home";
PROC MEANS DATA=cc.credit_card_data MIN;
   CLASS Home;
   VAR Rate;
RUN;
TITLE;


/* 6.avg of income by age*/
TITLE "avg of income by age";
PROC MEANS DATA=cc.credit_card_data MEAN;
   CLASS Age;
   VAR Income;
RUN;
TITLE;

/* 7.count of id by card_length*/
TITLE "count of id by card_length";
PROC SQL;
   SELECT Cred_Length, COUNT(ID) AS ID_Count
   FROM cc.credit_card_data
   GROUP BY Cred_Length;
QUIT;
TITLE;

/* 8.avg of income by years*/
TITLE "avg of income by years";
PROC SQL OUTOBS=20;
   SELECT DATE, AVG(Income) AS Avg_Income
   FROM cc.credit_card_data
   GROUP BY DATE;
QUIT;
TITLE;

/* 9.univariate of income */
TITLE "univariate of income";
PROC UNIVARIATE DATA=cc.credit_card_data;
   VAR Income;
RUN;
TITLE;
```

OUTPUT AND RESULTS

LIBNAME STATEMENT :

Code-:

```
/*CREATING LIBRARY */
LIBNAME CC "/home/u62467441/credit card project";
```

Output-:

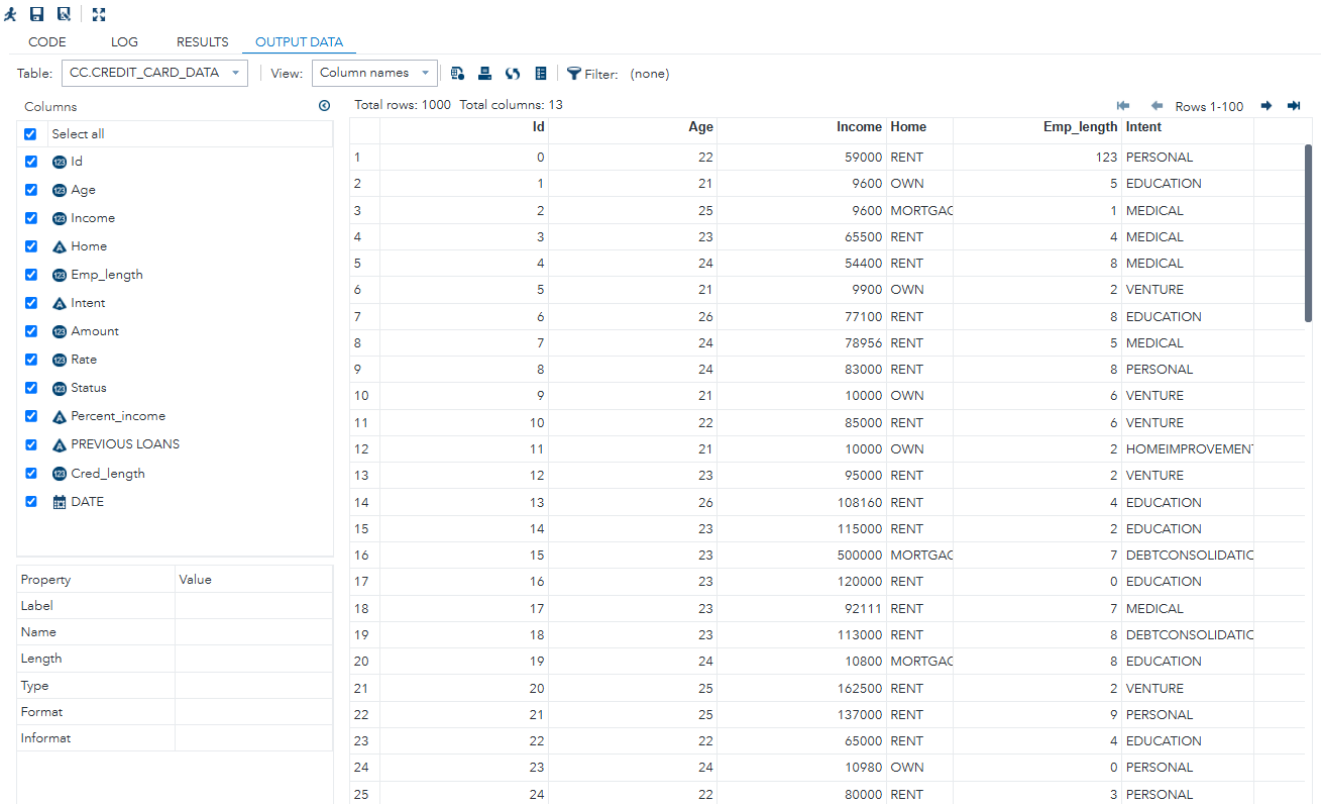

```
1          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
68
69         LIBNAME CC "/home/u62467441/credit card project";
NOTE: Libref CC refers to the same physical library as _TEMP0.
NOTE: Libref CC was successfully assigned as follows:
      Engine:        V9
      Physical Name: /home/u62467441/credit card project
70
71         OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
81
```

IMPORT DATA :

Code-:

```
/*IMPORT DATA*/
PROC IMPORT DATAFILE="/home/u62467441/credit card project/credit_card_dataset.csv"
    OUT=CC.credit_card_data
    DBMS=CSV
    REPLACE;
    GETNAMES=YES;
RUN;
```

Output-:



CODE LOG RESULTS OUTPUT DATA

Table: CC.CREDIT_CARD_DATA | View: Column names | Filter: (none)

Columns

- Select all
- Id
- Age
- Income
- Home
- Emp_length
- Intent
- Amount
- Rate
- Status
- Percent_income
- PREVIOUS LOANS
- Cred_length
- DATE

Property	Value
Label	
Name	
Length	
Type	
Format	
Informat	

Total rows: 1000 Total columns: 13 Rows 1-100

	Id	Age	Income	Home	Emp_length	Intent
1	0	22	59000	RENT	123	PERSONAL
2	1	21	9600	OWN	5	EDUCATION
3	2	25	9600	MORTGAC	1	MEDICAL
4	3	23	65500	RENT	4	MEDICAL
5	4	24	54400	RENT	8	MEDICAL
6	5	21	9900	OWN	2	VENTURE
7	6	26	77100	RENT	8	EDUCATION
8	7	24	78956	RENT	5	MEDICAL
9	8	24	83000	RENT	8	PERSONAL
10	9	21	10000	OWN	6	VENTURE
11	10	22	85000	RENT	6	VENTURE
12	11	21	10000	OWN	2	HOMEIMPROVEMEN
13	12	23	95000	RENT	2	VENTURE
14	13	26	108160	RENT	4	EDUCATION
15	14	23	115000	RENT	2	EDUCATION
16	15	23	500000	MORTGAC	7	DEBTCONSOLIDATIC
17	16	23	120000	RENT	0	EDUCATION
18	17	23	92111	RENT	7	MEDICAL
19	18	23	113000	RENT	8	DEBTCONSOLIDATIC
20	19	24	10800	MORTGAC	8	EDUCATION
21	20	25	162500	RENT	2	VENTURE
22	21	25	137000	RENT	9	PERSONAL
23	22	22	65000	RENT	4	EDUCATION
24	23	24	10980	OWN	0	PERSONAL
25	24	22	80000	RENT	3	PERSONAL

EXPLORE THE DATA :

DATASET -: Credit card applicant (Kaggle Dataset)

Code-:

```
/*Reading the Data*/
proc print data=CC.credit_card_data (OBS=10);
run;
proc contents data=CC.credit_card_data;
run;
```

Output-:

Obs	Id	Age	Income	Home	Emp_length	Intent	Amount	Rate	Status	Percent_income	PREVIOUS LOANS	Cred_length	DATE
1	0	22	59000	RENT	123	PERSONAL	35000	16.02	1	59.00%	Y	3	10/05/2025
2	1	21	9600	OWN	5	EDUCATION	1000	11.14	0	10.00%	N	2	31/12/2025
3	2	25	9600	MORTGAGE	1	MEDICAL	5500	12.87	1	57.00%	N	3	10/11/2025
4	3	23	65500	RENT	4	MEDICAL	35000	15.23	1	53.00%	N	2	02/05/2023
5	4	24	54400	RENT	8	MEDICAL	35000	14.27	1	55.00%	Y	4	11/04/2024
6	5	21	9900	OWN	2	VENTURE	2500	7.14	1	25.00%	N	2	27/11/2023
7	6	26	77100	RENT	8	EDUCATION	35000	12.42	1	45.00%	N	3	29/03/2023
8	7	24	78956	RENT	5	MEDICAL	35000	11.11	1	44.00%	N	4	21/05/2025
9	8	24	83000	RENT	8	PERSONAL	35000	8.9	1	42.00%	N	2	11/05/2023
10	9	21	10000	OWN	6	VENTURE	1600	14.74	1	16.00%	N	3	08/02/2025

The CONTENTS Procedure

Data Set Name	CC.CREDIT_CARD_DATA	Observations	1000
Member Type	DATA	Variables	13
Engine	V9	Indexes	0
Created	03/12/2025 12:51:21	Observation Length	104
Last Modified	03/12/2025 12:51:21	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Engine/Host Dependent Information	
Data Set Page Size	131072
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	1258
Obs in First Data Page	1000
Number of Data Set Repairs	0
Filename	/home/u62467441/credit card project/credit_card_data.sas7bdat
Release Created	9.0401M7
Host Created	Linux
Inode Number	14966475922
Access Permission	rw-r--r--
Owner Name	u62467441
File Size	256KB
File Size (bytes)	262144

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat
2	Age	Num	8	BEST12.	BEST32.
7	Amount	Num	8	BEST12.	BEST32.
12	Cred_length	Num	8	BEST12.	BEST32.
13	DATE	Num	8	DDMMYY10.	DDMMYY10.
5	Emp_length	Num	8	BEST12.	BEST32.
4	Home	Char	8	$8.	$8.
1	Id	Num	8	BEST12.	BEST32.
3	Income	Num	8	BEST12.	BEST32.
6	Intent	Char	17	$17.	$17.
11	PREVIOUS LOANS	Char	1	$1.	$1.
10	Percent_income	Char	6	$6.	$6.
8	Rate	Num	8	BEST12.	BEST32.
9	Status	Num	8	BEST12.	BEST32.

UNDERSTANDING THE DATA :

Code-:

```
/*Understanding the Data*/
proc sql ;
describe table dictionary.columns;
select name, type, length
from dictionary.columns
where libname="CC";
quit;

proc sql number outobs=20;
select distinct Rate,Home,Intent
From CC.credit_card_data;
quit;
```

Output-:

Id	Age	Income	Home	Emp_length	Intent	Amount	Rate	Status	Percent_income	PREVIOUS LOANS	Cred_length	DATE
15	23	500000	MORTGAGE	7	DEBTCONSOLIDATION	30000	10.65	0	6.00%	N	3	04/04/2024
33	25	306000	RENT	2	DEBTCONSOLIDATION	24250	13.85	0	8.00%	N	3	20/04/2025

Status	Loan_Count
0	447
1	553

CREDIT INDENT DATA ANALYSIS

- **AVG INCOME AND RATE**

Code-:

```
/*CREDIT INDENT DATA ANALYSIS*/

/* 1.avg of income and rate */
TITLE "avg of income and rate";
PROC SQL;
    SELECT AVG(Income) AS Avg_Income,
           AVG(Rate) AS Avg_Rate
    FROM cc.credit_card_data;
QUIT;
title;
```

Output-:

max of amount

The MEANS Procedure

Analysis Variable : Amount
Maximum
35000.00

- **MAX OF AMOUNT**

Code-:

```
/* 2.max of amount */
TITLE "max of amount";
PROC MEANS DATA=cc.credit_card_data MAX;
    VAR Amount;
RUN;
TITLE;
```

Output-:

max of amount

The MEANS Procedure

Analysis Variable : Amount
Maximum
35000.00

- **MAX OF AMOUNT BY HOME**

Code-:

```
/* 3.max of amount by home*/
TITLE "max of amount by home";
PROC SQL;
    SELECT Home, MAX(Amount) AS Max_Amount
    FROM cc.credit_card_data
    GROUP BY Home;
QUIT;
TITLE;
```

Output-:

max of amount by home

Home	Max_Amount
MORTGAGE	35000
OTHER	7500
OWN	35000
RENT	35000

- **SUM OF AMOUNT BY INTENT**

Code-:

```
/* 4.sum of amount by intent*/
TITLE "sum of amount by intent";
PROC SQL;
    SELECT Intent, SUM(Amount) AS Total_Amount
    FROM cc.credit_card_data
    GROUP BY Intent;
QUIT;
TITLE;
```

Output-:

sum of amount by intent

Intent	Total_Amount
DEBTCONSOLIDATION	2836750
EDUCATION	3616200
HOMEIMPROVEMENT	1880250
MEDICAL	2884800
PERSONAL	3029000
VENTURE	2996775

- **MINIMUM OF RATE BY HOME**

Code-:

```
/* 5.min of rate by home*/
TITLE "min of rate by home";
PROC MEANS DATA=cc.credit_card_data MIN;
    CLASS Home;
    VAR Rate;
RUN;
TITLE;
```

Output-:

min of rate by home

The MEANS Procedure

Analysis Variable : Rate		
Home	N Obs	Minimum
MORTGAGE	193	5.4200000
OTHER	4	11.9700000
OWN	136	5.4200000
RENT	667	5.9900000

- **AVERAGE OF INCOME BY AGE**

Code-:

```
/* 6.avg of income by age*/
TITLE "avg of income by age";
PROC MEANS DATA=cc.credit_card_data MEAN;
    CLASS Age;
    VAR Income;
RUN;
TITLE;
```

Output-:

avg of income by age

The MEANS Procedure

Analysis Variable : Income		
Age	N Obs	Mean
21	70	34872.51
22	189	56586.53
23	203	76223.60
24	203	78190.43
25	193	91862.79
26	138	107859.91
123	2	79002.00
144	2	225000.00

- **COUNT OF ID BY CARD LENGTH**

Code-:

```
/* 7.count of id by card_length*/
TITLE "count of id by card_length";
PROC SQL;
    SELECT Cred_Length, COUNT(ID) AS ID_Count
    FROM cc.credit_card_data
    GROUP BY Cred_Length;
QUIT;
TITLE;
```

Output-:

count of id by card_length

Cred_length	ID_Count
2	334
3	330
4	336

- **AVERAGE OF INCOME BY YEARS**

Code-:

```
/* 8.avg of income by years*/
TITLE "avg of income by years";
PROC SQL OUTOBS=20;
    SELECT DATE, AVG(Income) AS Avg_Income
    FROM cc.credit_card_data
    GROUP BY DATE;
QUIT;
TITLE;
```

Output-:

avg of income by years

DATE	Avg_Income
01/01/2023	50800
02/01/2023	53500
04/01/2023	22608
05/01/2023	96750
06/01/2023	74000
07/01/2023	49200
08/01/2023	20280
09/01/2023	69000
10/01/2023	48600
11/01/2023	22296
12/01/2023	68400
13/01/2023	21600
14/01/2023	75000
15/01/2023	86498
17/01/2023	74004
18/01/2023	45067.5
20/01/2023	175000
21/01/2023	95000
22/01/2023	120000
23/01/2023	62047

- **UNIVARIATE OF INCOME**

Code-:

```
/* 9.univariate of income */
TITLE "univariate of income";
PROC UNIVARIATE DATA=cc.credit_card_data;
    VAR Income;
RUN;
TITLE;
```

Output-:

univariate of income

The UNIVARIATE Procedure
Variable: Income

Moments			
N	1000	Sum Weights	1000
Mean	77704.169	Sum Observations	77704169
Std Deviation	61854.7976	Variance	3826015987
Skewness	1.55983426	Kurtosis	3.26550345
Uncorrected SS	9.86013E12	Corrected SS	3.82219E12
Coeff Variation	79.602933	Std Error Mean	1956.02045

Basic Statistical Measures			
Location		Variability	
Mean	77704.17	Std Deviation	61855
Median	62048.50	Variance	3826015987
Mode	60000.00	Range	490400
		Interquartile Range	71000

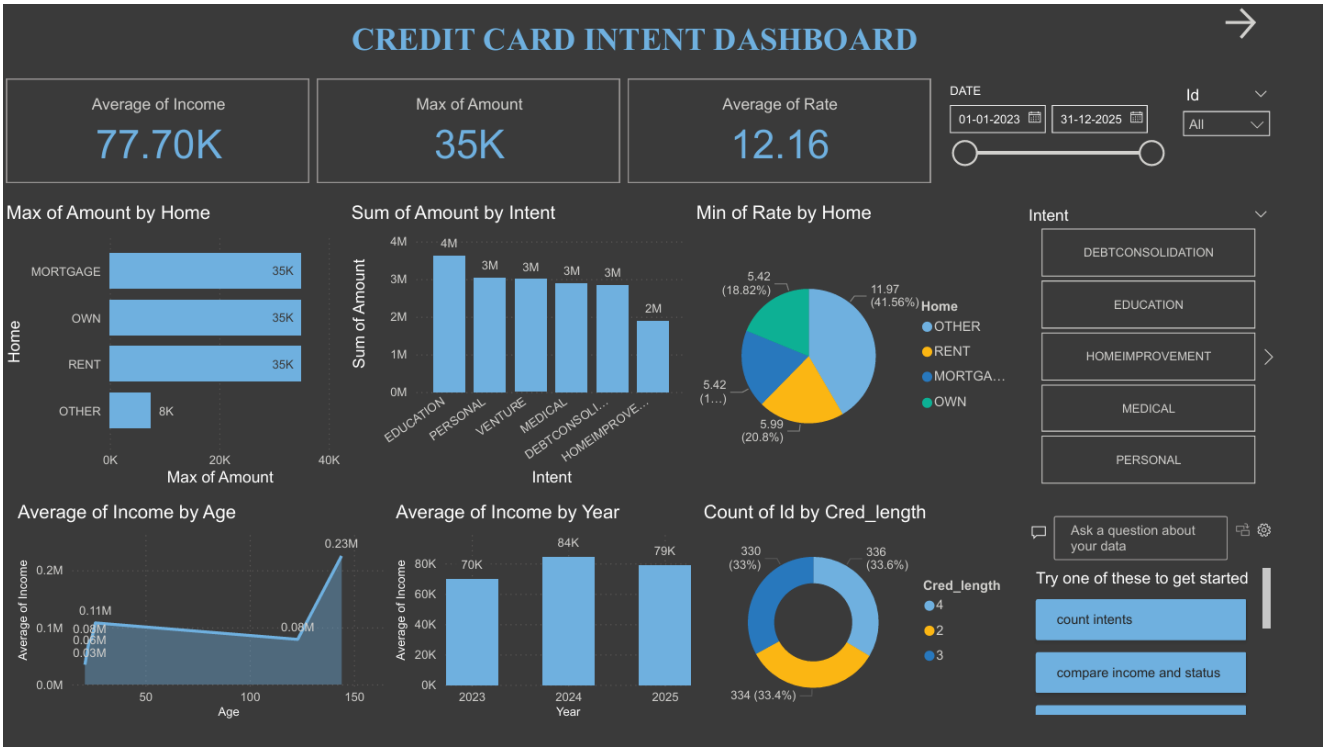
Tests for Location: Mu0=0				
Test	Statistic		p Value	
Student's t	t	39.72564	Pr > |t|	<.0001
Sign	M	500	Pr >= |M|	<.0001
Signed Rank	S	250250	Pr >= |S|	<.0001

Quantiles (Definition 5)	
Level	Quantile
100% Max	500000.0
99%	278552.0
95%	200000.0
90%	175000.0
75% Q3	95000.0
50% Median	62048.5
25% Q1	24000.0
10%	18000.0
5%	14400.0
1%	10890.0
0% Min	9600.0

Extreme Observations	
Lowest	Highest

VISUVALIZING THE ANALYZED DATA IN POWER BI

11.1 POWERBI DASHBOARD AND MARTIX :



The Credit Card Intent Dashboard shows key financial insights.

The average income is $77.70K, the maximum amount is $35K, and the average rate is 12.16%. Mortgage, Own, and Rent homes have the highest amounts ($35K), while Other is lower ($8K).

Education has the highest total amount ($4M), followed by other intents at around $3M. Owned homes have the lowest interest rate (11.97).

Income peaked at $84K in 2024 and dropped slightly in 2025.

Credit lengths of 2, 3, and 4 years are almost equal.

The dashboard has filters and a chatbot for quick analysis, helping in decision-making.

MATRIX OF CREDIT INTENT BASED ON YEAR

Year	2023		2024		2025		Total	
Intent	Sum of Amount	Average of Rate	Sum of Amount	Average of Rate	Sum of Amount	Average of Rate	Sum of Amount	Average of Rate
⊟ DEBTCONSOLIDATION	858925	12.26	779300	12.85	1198525	12.41	2836750	12.50
MORTGAGE	74375	10.94	102500	12.96	168775	11.63	345650	11.87
OWN	22200	13.35	15750	11.58	14150	11.01	52100	11.88
RENT	762350	12.51	661050	12.93	1015600	12.85	2439000	12.77
⊟ EDUCATION	1174300	12.26	1441725	11.74	1000175	11.66	3616200	11.89
MORTGAGE	132025	12.02	279125	11.07	88750	11.00	499900	11.38
OTHER			4000	12.84	2400	12.21	6400	12.53
OWN	57225	10.14	54200	11.56	48675	10.91	160100	10.80
RENT	985050	12.74	1104400	11.94	860350	11.96	2949800	12.21
⊟ HOMEIMPROVEMENT	684550	12.27	352150	12.67	843550	12.58	1880250	12.49
MORTGAGE	40550	9.40	31150	16.17	130925	13.68	202625	12.61
OWN	7750	11.68	44200	10.76	66875	10.07	118825	10.57
RENT	636250	12.98	276800	12.87	645750	12.96	1558800	12.95
⊟ MEDICAL	949025	12.65	901825	12.73	1033950	12.39	2884800	12.58
MORTGAGE	33300	12.94	142400	13.95	143550	11.10	319250	12.54
OWN	51825	11.26	21675	9.55	39400	12.61	112900	11.45
RENT	863900	12.87	737750	12.65	851000	12.76	2452650	12.77
⊟ PERSONAL	859400	12.28	851175	11.74	1318425	11.73	3029000	11.88
MORTGAGE	68525	7.47	112100	10.73	136975	9.28	317600	9.60
OTHER	7500	11.97			6000	12.21	13500	12.09
OWN	26500	10.37	81800	10.31	128400	11.27	236700	10.67
RENT	756875	13.19	657275	12.69	1047050	12.50	2461200	12.77
⊟ VENTURE	780550	11.21	1170500	11.83	1045725	12.47	2996775	11.88
MORTGAGE	42150	10.14	194850	11.63	138775	11.55	375775	11.30
OWN	84325	10.54	28100	10.46	83950	11.41	196375	10.86
RENT	654075	11.74	947550	12.07	823000	13.06	2424625	12.34
Total	5306750	12.14	5496675	12.17	6440350	12.18	17243775	12.16

The Matrix of Credit Intent Based on Year shows the total amount and average interest rates for different credit intents from 2023 to 2025.

The total sum of amounts increased from $5.3M in 2023 to $6.4M in 2025, with an overall average interest rate of 12.16%.

Education had the highest total amount ($3.6M), followed by Medical ($2.88M), Personal ($3.02M), and Debt Consolidation ($2.83M).

Rent-based credit had the highest amounts across all intents, while mortgage and own categories had lower values. Interest rates varied, with the lowest for Personal (9.60%) and the highest for Venture (12.47%).

The trend shows an increase in borrowing amounts over time, with stable average interest rates.

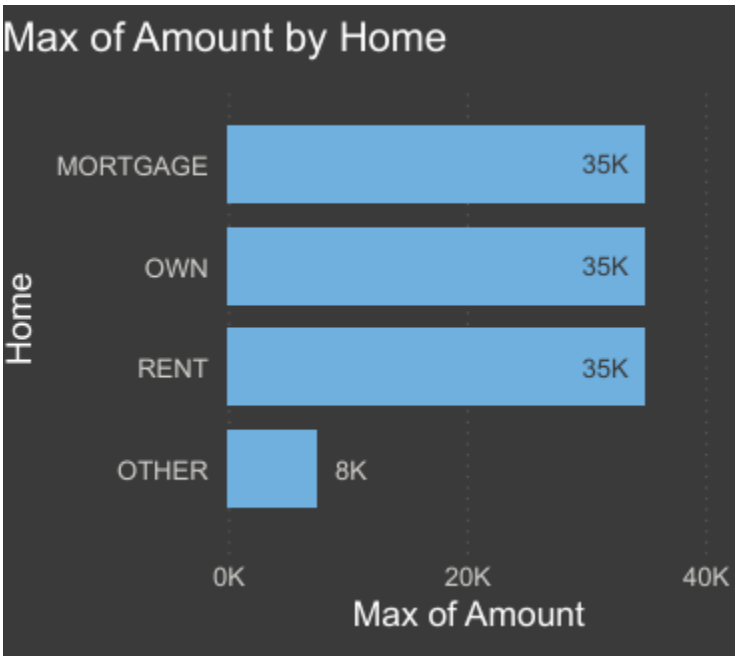
11.2 CHARTS USED FOR THESE VISUVAL

- AVG INCOME AND RATE & MAX OF AMOUNT

CARD NEW:

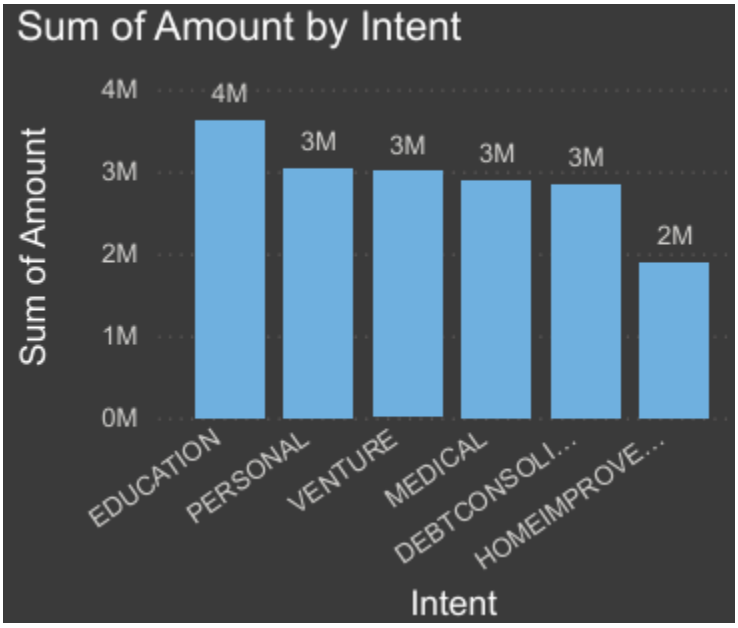
- MAX OF AMOUNT BY HOME

STACKED BAR CHART:



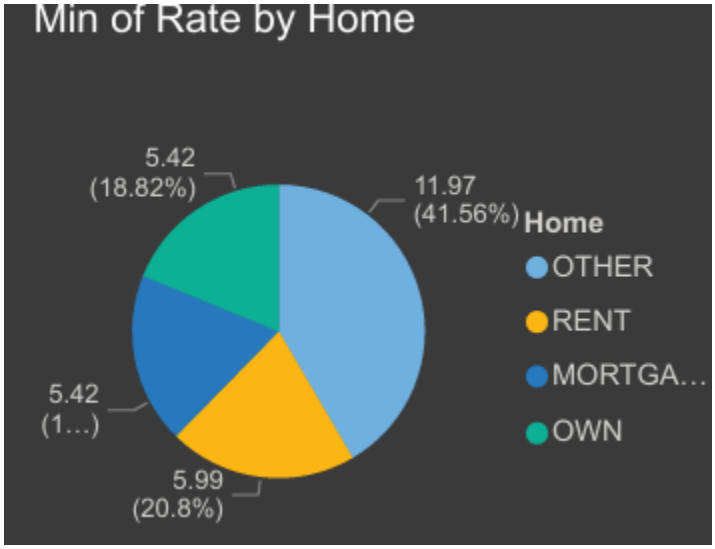
- SUM OF AMOUNT BY INTENT

CLUSTERED COLUMN CHAT:



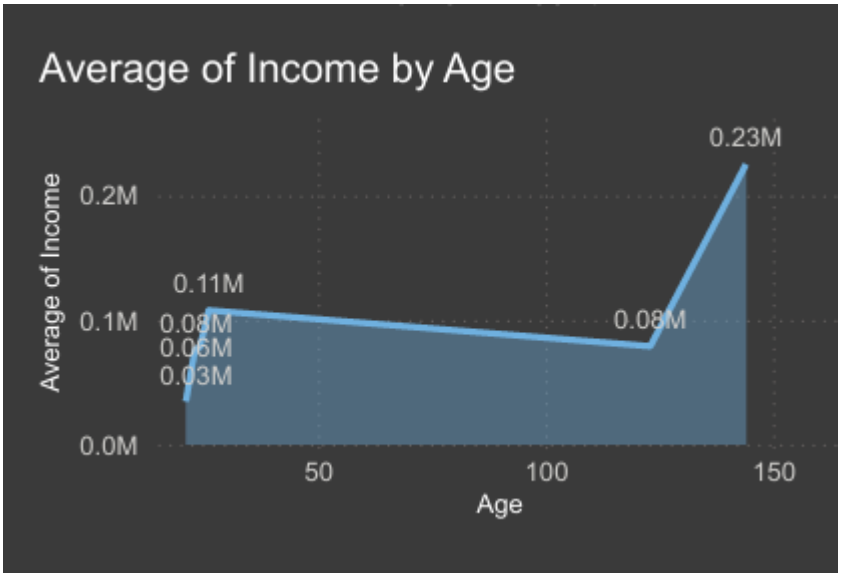
- MINIMUM OF RATE BY HOME

PIE CHART:



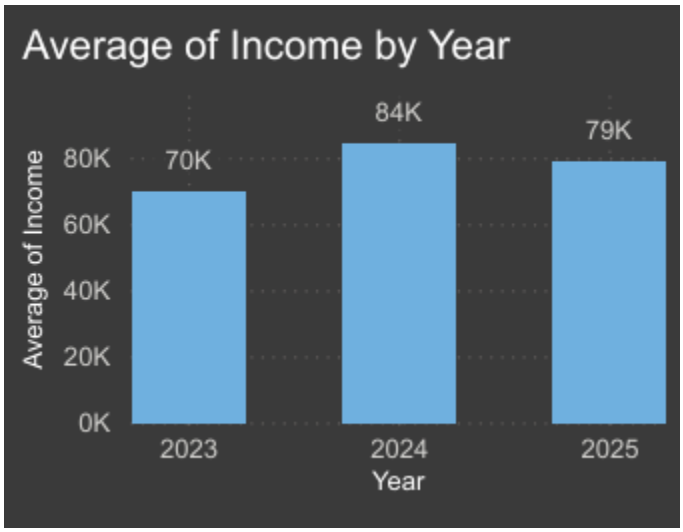
- AVERAGE OF INCOME BY AGE

STACKED AREA CHART:



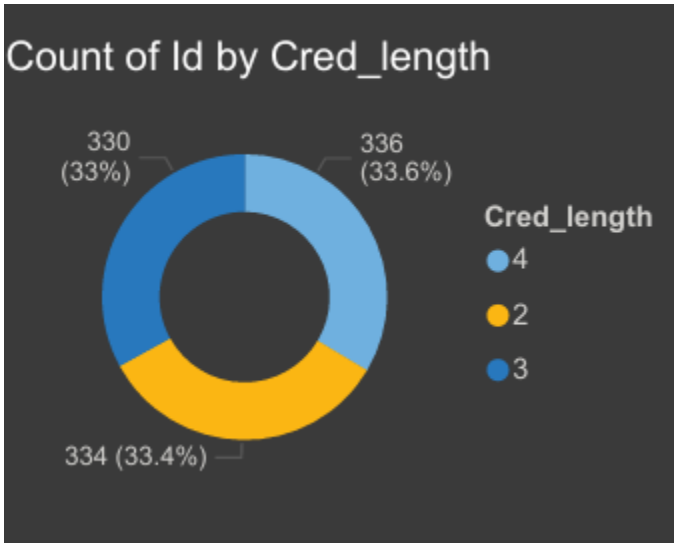
- COUNT OF ID BY CARD LENGTH

LINE AND STATCKET COLUMN CHART:

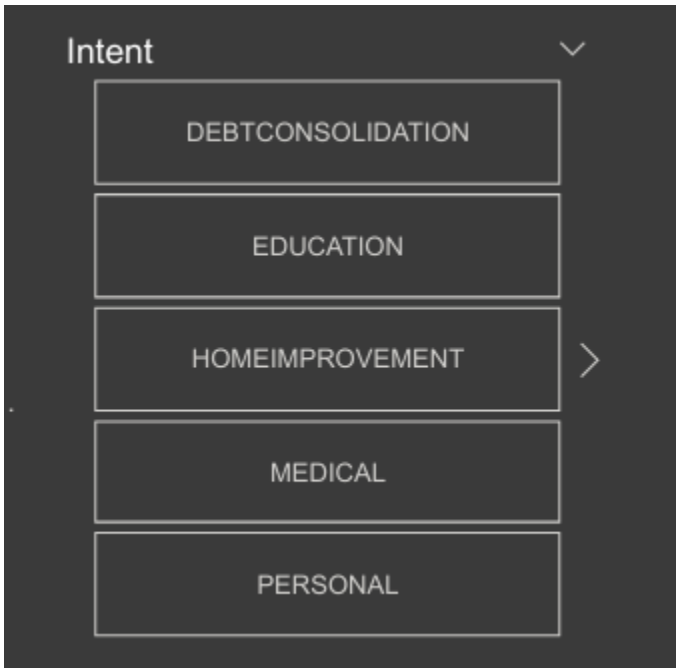


- AVERAGE OF INCOME BY YEARS

DONUT CHART:



11.3 FILTERS USED IN THE DASHBORAD:



- This Intent column slicer is used to filter the data by intent categrory.

- This date column slicer is used to filter the data by between Format.

- This ID column slicer is used to filter the data by each ID wise.

CONCLUSION

The Credit Card Intent Analysis project successfully identifies key factors influencing an individual's decision to apply for a credit card.

By leveraging SAS for statistical analysis and Power BI for data visualization, the study provides valuable insights into demographic, financial, and behavioral attributes affecting credit card applications.

The results indicate that factors like income, employment length, credit history, and loan intent play a significant role in determining approval rates.

The interactive dashboards help in visualizing trends and patterns, making it easier for financial institutions to refine their marketing strategies and reduce credit risk.

Overall, this project enhances data-driven decision-making in the credit industry.

REFERENCE

 SAS Documentation – SAS Institute Inc. (2025). *SAS® 9.4 Procedures Guide: Statistical Procedures*.
Retrieved from https://documentation.sas.com

 Power BI Documentation – Microsoft Corporation. (2025). *Power BI User Guide & Documentation*.
Retrieved from https://learn.microsoft.com/en-us/power-bi/

 Credit Card Intent Dataset – Kaggle. (2025). *Credit Card Applicant Dataset*.
Retrieved from https://www.kaggle.com

 Exploratory Data Analysis – Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley Publishing.

 Statistical Methods for Credit Risk – Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

 Business Intelligence and Decision Making – Davenport, T. H., & Harris, J. G. (2007). *Competing on Analytics: The New Science of Winning*. Harvard Business Press.

 Financial Institutions and Credit Analysis – Saunders, A., & Cornett, M. M. (2022). *Financial Institutions Management: A Risk Management Approach*. McGraw-Hill Education.