BANK LOAN ANALYSIS

**INTRODUCTION**

This report analyzes loan data to identify factors affecting performance and profitability. We explore borrower characteristics, loan products, geographic trends, and time-based patterns to recommend strategies for optimizing lending practices.

**OBJECTIVE:**

1. **Loan Performance & Risk:** Why do borrowers repay or default?
2. **Loan Products & Portfolio:** Find best loan types & borrowers for max profit.
3. **Geographical Analysis:** See how loans perform across different states.
4. **Time-based Analysis:** Track loan trends over months & quarters.

**Tools used in analysis:**

1. **Excel**: Initial CSV data inspection and handling of missing values.
2. **MySQL**: Store & clean data, run queries
3. **Power BI**: For visualization and dashboard creation.
4. **DAX Formulas**: Used in Power BI for calculations and metrics.

**DATA PREPARATION**

**Understand the key dataset features**

* address\_state: Borrower's state.
* emp\_length: Employment length.
* annual\_income: Borrower's annual income.
* dti: Borrower's Debt-to-income ratio.
* total\_acc: Total credit lines.
* home\_ownership: Home ownership status.
* Grade, sub\_grade: creditworthiness.
* issue\_date: Loan issue date.
* loan\_status: Current status of the loan.
* term: Loan term in months.
* verification\_status: Verification status of the borrower's information.
* int\_rate: Annual Interest rate charged on the loan.
* loan\_amount: Total amount of the loan borrowed.

**Data Inspection in CSV**

It's crucial to inspect the dataset's structure, any inconsistencies, and cleanliness before analysis.

* **Handling Missing Values**: Steps taken to handle missing values.
  + Issue: The emp\_title field contains 1,433 missing values (blank entries).
  + Solution: Did text parsing. Missing values in emp\_title was imputed with a placeholder value, "Other". This approach assumes missing titles are evenly distributed across various job categories.
* **Fix inconsistent date format**: Standardize date format to YYYY-MM-DD.
  + Applied formula: ‘=DATE(RIGHT(A2,4),MID(A2,5,2),LEFT(A2,2))’. After applying the formula, the date converted to a numerical value like '44645'.
  + Then converted the date cells to 'Date' format, set to 'yyyy-mm-dd', and updated dates in CSV.
* **Data Acquisition and Storage**

The next step involved importing this data into MySQL for further analysis and manipulation.

* + Creating a New Database. ‘CREATE DATABASE database\_name;’
  + Use 'Table Data Import Wizard’: Right-click database and select 'Table Data Import Wizard', browse CSV, set table name, click 'Next' for auto-detection, click 'Next' to import, then wait for confirmation.

**DATA EXPLORATION AND CLEANING IN MYSQL**

**1. Explore the Data**

Initial Inspection.

1.1  **Count the number of records:** Total number of records in the dataset.

Code: SELECT COUNT(\*) AS total\_records FROM loan\_data;

Output: 38,576

1.2 **Data types of columns:** Overview of data types.

Code: *DESCRIBE loan\_data;*

|  |  |  |
| --- | --- | --- |
| **Filed** | **Type** | **Null** |
| id | int | YES |
| address\_state | text | YES |
| … | … | … |
| total\_acc | int | YES |
| total\_payment | int | YES |

Output:

Key Takeaway: Loan data inconsistency. Key dates & categories stored as text.

We will convert to appropriate data type in data cleaning phase.

1.3 **Sample records:** Example records from the dataset.

Code: *SELECT \* FROM loan\_data LIMIT 5;*

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **st** | **...** | **tot\_pay** |
| 1077430 | GA | ... | 1009 |
| 1072053 | CA | ... | 3939 |
| 1069243 | CA | ... | 3522 |
| 1041756 | TX | ... | 4911 |
| 1068350 | IL | ... | 3835 |

Output:

Observation: Sample query previews data & structure.

1.4 **Summary Statistics:** numerical data

Created query using CTEs to calculate numeric summaries (total, avg, min, max, std dev), and quartiles (Q1, median, Q3).

Techniques used - CTEs for defining temporary result sets, Aggregations to group data, Window function to calculate quartiles by dividing data into four equal parts.

Code:

WITH loan\_stats\_cte AS (

SELECT

CONCAT(FORMAT(SUM(annual\_income) / 1000000,2), ' Million($)') Total\_Value,

ROUND(AVG(annual\_income)) AS Average,

MIN(annual\_income) AS Minimum,

MAX(annual\_income) AS Maximum,

ROUND(STDDEV(annual\_income)) AS std\_dev

FROM loan\_data

),

quartile\_cte AS (

SELECT

MAX(CASE WHEN quartile = 1 THEN annual\_income END) AS "Q1\_25%",

MAX(CASE WHEN quartile = 2 THEN annual\_income END) AS Median,

MAX(CASE WHEN quartile = 3 THEN annual\_income END) AS "Q3\_75%"

FROM (

SELECT annual\_income, NTILE(4) OVER (ORDER BY annual\_income) AS quartile FROM loan\_data

) AS quartile\_groups

)

SELECT \* FROM loan\_stats\_cte, quartile\_cte;

Note: provided query will be changed for each numeric field.

Consolidated Output:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Numeric Field** | **Total\_Value** | **Avg** | **Min** | **Max** | **std\_dev** | **Q1** | **Median** | **Q3** |
| loan\_amount | 435.76M | 11.3k | 500 | 35k | 7.5k | 5.5k | 10k | 15k |
| total\_payment | 473.07M | 12.3k | 34 | 58.6k | 9k | 5.6k | 10k | 16.7k |
| installment | NA | 327 | 15.69 | 1.3k | 209 | 168 | 283 | 434 |
| int\_rate | NA | 12% | 5.4% | 24.6% | 4% | 9.3% | 11.9% | 14.6% |
| annual\_income | NA | 69.6k | 4k | 6M | 64k | 41.5k | 60k | 83k |
| dti | NA | 13% | 0 | 30% | 6.7% | 8% | 13% | 18.6% |
| total\_acc | NA | 22 | 2 | 90 | 11 | 14 | 20 | 29 |

Observation: Loan amounts: $500-$35k (avg. $11.3k). Income: $4k-$6M (avg. $69.6k). Avg DTI: 13.33% (0%-30%).

1.5  **Categorical Variables:** Explores categorical data distribution.

Query Execution: It counts loans by category, calculates % and sorts by most loans first.

SELECT address\_state, COUNT(\*) AS count,

CONCAT(ROUND(COUNT(\*) / (SELECT COUNT(\*) FROM loan\_data) \* 100, 2), '%') AS percentage

FROM loan\_data GROUP BY address\_state ORDER BY count DESC;

Note: provided query will be changed for each category.

Output of:

**Address state**

|  |  |  |
| --- | --- | --- |
| address\_state | count | percentage\_total |
| CA | 6894 | 17.87% |
| NY | 3701 | 9.59% |
| FL | 2773 | 7.19% |
| TX | 2664 | 6.91% |
| NJ | 1822 | 4.72% |
| IL | 1486 | 3.85% |
| PA | 1482 | 3.84% |
| VA | 1375 | 3.56% |
| GA | 1355 | 3.51% |
| MA | 1310 | 3.40% |

Observation: Loan applicants are geographically diverse (CA, NY, FL lead).

**Borrower’s Employee Length**

|  |  |  |
| --- | --- | --- |
| emp\_length | count | percentage\_total |
| 10+ years | 8870 | 22.99% |
| < 1 year | 4575 | 11.86% |
| 2 years | 4382 | 11.36% |
| 3 years | 4088 | 10.60% |
| 4 years | 3428 | 8.89% |
| 5 years | 3273 | 8.48% |
| 1 year | 3229 | 8.37% |
| 6 years | 2228 | 5.78% |
| 7 years | 1772 | 4.59% |
| 8 years | 1476 | 3.83% |
| 9 years | 1255 | 3.25% |

Observation: Emp. length is spread out: 10+ yrs (23%) most common, then <1 yr (12%) & 2 yrs (11%).

|  |  |  |
| --- | --- | --- |
| grade | count | percentage\_total |
| B | 11674 | 30.26% |
| A | 9689 | 25.12% |
| C | 7904 | 20.49% |
| D | 5182 | 13.43% |
| E | 2786 | 7.22% |
| F | 1028 | 2.66% |
| G | 313 | 0.81% |

**Loan Grade**

Observation: Most loans (55%) are B or A (fair-good credit), with fewer loans in lower grades D-G (higher default risk).

|  |  |  |
| --- | --- | --- |
| sub\_grade | count | percentage\_total |
| B3 | 2834 | 7.35% |
| A4 | 2803 | 7.27% |
| A5 | 2654 | 6.88% |
| B5 | 2644 | 6.85% |
| B4 | 2455 | 6.36% |
| C1 | 2089 | 5.42% |
| B2 | 1990 | 5.16% |
| C2 | 1972 | 5.11% |
| B1 | 1751 | 4.54% |
| A3 | 1740 | 4.51% |

**Sub Grade:** Top 10

Observation: Distribution across subgrades within each main grade is fairly even.

|  |  |  |
| --- | --- | --- |
| home\_ownership | Count | percentage\_total |
| RENT | 18439 | 47.80% |
| MORTGAGE | 17198 | 44.58% |
| OWN | 2838 | 7.36% |
| OTHER | 98 | 0.25% |
| NONE | 3 | 0.01% |

**home\_ownership:**

Observation: Borrowers split between renters (47.8%) and mortgage (44.6%), with few owning outright (7.4%).

|  |  |  |
| --- | --- | --- |
| loan\_status | count | percentage\_total |
| Fully Paid | 32145 | 83.33% |
| Charged Off | 5333 | 13.82% |
| Current | 1098 | 2.85% |

**Loan Status:**

Observation: Mostly Paid (83%), Defaults Exist (14%). (Current: 3%).

**Loan Purpose:**

|  |  |  |
| --- | --- | --- |
| purpose | count | percentage\_total |
| Debt consolidation | 18214 | 47.22% |
| credit card | 4998 | 12.96% |
| … | … | … |
| medical | 667 | 1.73% |
| moving | 559 | 1.45% |

Observation: Debt consolidation tops loans (47%), followed by credit cards (13%). Data covers diverse borrowing needs.

|  |  |  |
| --- | --- | --- |
| term | count | percentage\_total |
| 36 months | 28237 | 73.20% |
| 60 months | 10339 | 26.80% |

**Loan Term:**

Observation: Loans are mostly short-term (73% are 36 months).

|  |  |  |
| --- | --- | --- |
| verification\_status | count | percentage\_total |
| Not Verified | 16464 | 42.68% |
| Verified | 12335 | 31.98% |
| Source Verified | 9777 | 25.34% |

**Loan verification\_status:**

Observation: Many loans (43%) lack income verification. This could be due to borrower challenges or loan types not requiring it.

Key categorical findings:

* Borrowers nationwide, with many having short (>1 yr.) or long (10+) work histories.
* Loan grades cluster around fair-good credit (B & A), with more defaults likely in lower grades.
* Debt consolidation is the most common loan purpose, and most loans are short-term (36 months).
* While most loans are fully repaid (83%), a concerning number are charged off (14%).
* Income verification is incomplete for many loans (43%).

1.6 **Date Range Analysis:** Date range covered in the dataset.

Fields: issue\_date, last\_credit\_pull\_date, last\_payment\_date, loan\_status, next\_payment\_date

Query :

SELECT

MIN(issue\_date) AS earliest\_issue\_date,

MAX(issue\_date) AS latest\_issue\_date

FROM loan\_data;

Note: provided query will be changed for each category.

Consolidated Output:

|  |  |  |
| --- | --- | --- |
| Date Field | earliest\_date | latest\_date |
| issue\_date | 01-Jan-21 | 12-Dec-21 |
| last\_credit\_pull\_date | 13-Jan-21 | 20-Jan-22 |
| last\_payment\_date | 13-Jan-21 | 15-Dec-21 |
| next\_payment\_date | 13-Feb-21 | 01-Dec-22 |

This analysis gives us a clear picture of the timeframe for the loan issuance, credit pull, last payment, and upcoming payment dates.

**2. Data Cleaning**

* 1. **Handle Missing Values:**

Techniques used - Coalescing for NULL check.

we use the ‘COALESCE’ function in MySQL. The COALESCE function returns the first non-NULL value in the list of arguments. If all arguments are NULL, then it returns NULL. Here, we are checking each column for NULL values.

Query:

SELECT \* FROM loan\_data

WHERE COALESCE(id, address\_state, application\_type, emp\_length, emp\_title, grade,

home\_ownership, issue\_date, last\_credit\_pull\_date, last\_payment\_date,

loan\_status, next\_payment\_date, member\_id, purpose, sub\_grade, term,

verification\_status, annual\_income, dti, installment, int\_rate,

loan\_amount, total\_acc, total\_payment) IS NULL;

Observation: No missing value in dataset.

**2.2 Check for Duplicates:** Identifying and removing duplicate records.

Duplicates skew analysis. Finding and removing them is essential for clean, reliable data.

Identify Duplicate Records:

Using GROUP BY and HAVING, we identify duplicate records by grouping all dataset columns and counting occurrences. A count exceeding 1 in any group signals duplicate records.

Query Structure: SELECT column1, column2, COUNT(\*) FROM table\_name GROUP BY column1, column2 HAVING COUNT(\*) > 1;

Query Execution:

SELECT id, address\_state, application\_type, emp\_length, emp\_title, grade, home\_ownership, issue\_date, last\_credit\_pull\_date, last\_payment\_date, loan\_status, next\_payment\_date, member\_id, purpose, sub\_grade, term, verification\_status, annual\_income, dti, installment, int\_rate, loan\_amount, total\_acc, total\_payment,

COUNT(\*) AS duplicate\_count

FROM loan\_data

GROUP BY id, address\_state, application\_type, emp\_length, emp\_title, grade, home\_ownership, issue\_date, last\_credit\_pull\_date, last\_payment\_date, loan\_status, next\_payment\_date, member\_id, purpose, sub\_grade, term, verification\_status, annual\_income, dti, installment, int\_rate, loan\_amount, total\_acc, total\_payment

HAVING COUNT(\*) > 1;

Observation: No duplicate records were found in the dataset.

* 1. **Data Type Conversion:** Converting fields to appropriate data types.

Changing data types to appropriate ones as identified in step 1.2.

Techniques used - Data migration with a staging table.

To minimize the risk of data loss or corruption during the conversion process, changes are initially tested on a subset of the data. Once verified, the changes are applied to the entire dataset.

**Steps for Data Type Conversion:**

i. **Create a New Backup Table**

To preserve the original data, a backup table is created with the same structure as the original table.

CREATE TABLE loan\_data\_backup LIKE loan\_data;

ii. **Copy Subset of Data from Original Table to Backup Table**

A subset of the original data is copied to the backup table for testing the changes.

INSERT INTO loan\_data\_backup SELECT \* FROM loan\_data LIMIT 100;

**Recommended Data Type Changes:**

* VARCHAR: For purpose, sub\_grade, address\_state  
  Reason: Better performance and storage efficiency for shorter strings.
* ENUM: For categorical fields  
  Reason: Suitable for fields with a predefined set of values, providing data integrity.
* DATE: For date fields (issue\_date, last\_credit\_pull\_date, last\_payment\_date, next\_payment\_date)  
  Reason: Supports date-related operations and ensures consistency in date formats.
* DECIMAL: For numeric fields (dti, instalment, int\_rate)  
  Reason: Provides accurate representation and calculations for financial values.

**Code for Data Type Conversion:**

The following SQL statements demonstrate the changes made to the data types on the backup table. These changes were tested on the backup table and then executed on the actual table.

-- Change data type of address\_state from TEXT to VARCHAR

ALTER TABLE bank\_loan\_analysis.loan\_data\_backup

MODIFY COLUMN address\_state VARCHAR(50);

-- Change data type of application\_type from TEXT to ENUM

ALTER TABLE bank\_loan\_analysis.loan\_data\_backup

MODIFY COLUMN application\_type ENUM('INDIVIDUAL');

-- Change data type of emp\_length from TEXT to ENUM

ALTER TABLE bank\_loan\_analysis.loan\_data\_backup

MODIFY COLUMN emp\_length ENUM('< 1 year', '1 year', '2 years', '3 years', '4 years', '5 years', '6 years', '7 years', '8 years', '9 years', '10+ years');

-- Change data type of grade from TEXT to ENUM

ALTER TABLE bank\_loan\_analysis.loan\_data\_backup

MODIFY COLUMN grade ENUM('A', 'B', 'C', 'D', 'E', 'F', 'G');

-- Change data type of home\_ownership from TEXT to ENUM

ALTER TABLE bank\_loan\_analysis.loan\_data\_backup

MODIFY COLUMN home\_ownership ENUM('MORTGAGE', 'RENT', 'OWN', 'OTHER', 'NONE');

-- Change data type of issue\_date, last\_credit\_pull\_date, last\_payment\_date, next\_payment\_date from TEXT to DATE

ALTER TABLE bank\_loan\_analysis.loan\_data\_backup

MODIFY COLUMN issue\_date DATE,

MODIFY COLUMN last\_credit\_pull\_date DATE,

MODIFY COLUMN last\_payment\_date DATE,

MODIFY COLUMN next\_payment\_date DATE;

-- Change data type of loan\_status from TEXT to ENUM

ALTER TABLE bank\_loan\_analysis.loan\_data\_backup

MODIFY COLUMN loan\_status ENUM('Fully Paid', 'Charged Off', 'Current');

-- Change data type of purpose, sub\_grade from TEXT to VARCHAR

ALTER TABLE bank\_loan\_analysis.loan\_data\_backup

MODIFY COLUMN purpose VARCHAR(255),

MODIFY COLUMN sub\_grade VARCHAR(5); -- adjust length if needed

-- Change data type of term from TEXT to ENUM

ALTER TABLE bank\_loan\_analysis.loan\_data\_backup

MODIFY COLUMN term ENUM(' 36 months', ' 60 months');

-- Change data type of verification\_status from TEXT to ENUM

ALTER TABLE bank\_loan\_analysis.loan\_data\_backup

MODIFY COLUMN verification\_status ENUM('Verified', 'Not Verified', 'Source Verified');

-- Change data type of dti, installment, int\_rate from double to DECIMAL

ALTER TABLE bank\_loan\_analysis.loan\_data\_backup

MODIFY COLUMN dti DECIMAL(10, 4), -- adjust precision and scale as needed

MODIFY COLUMN installment DECIMAL(10, 2), -- adjust precision and scale as needed

MODIFY COLUMN int\_rate DECIMAL(6, 4); -- adjust precision and scale as needed

**Final Step:**

Once changes are confirmed on the backup table, applied them to the loan\_data table.

**DATA ANALYSIS AND QUERYING**

**1. Loan Performance & Risk Assessment**

Understand how well borrowers repay loans and identify factors that increase the risk of default.

**1.1** **Loan Repayment & Default Rates by Loan Segments:**

Segment the loan\_status by grade, term, emp\_length, verification\_status using cross-tabulation tables.

Repayment rate = % of loans fully paid. Default rate = % of loans charged off.

Repayment Rate = (Number of Fully Paid Loans ÷ Total Number of Loans) × 100

Default Rate = (Number of Defaulted Loans ÷ Total Number of Loans) × 100

|  |  |  |
| --- | --- | --- |
| Loan Status | Count | Total% |
| Fully Paid | 32145 | 83.33% |
| Charged Off | 5333 | 13.82% |
| Current | 1098 | 2.85% |

Here's an overview of the loan status distribution for our entire dataset:

We created a stored procedure to analyze loan performance across various segments (passed as parameters).

It calculates repayment rate, default rate, and loan counts for each category.

Techniques used - Dynamic SQL (constructing SQL statements at runtime using CONCAT) and stored procedures (takes a segment category as input. Inside the procedure, the dynamic SQL statement is constructed and then executed using PREPARE and EXECUTE statements).

Code:

DELIMITER //

CREATE PROCEDURE LoanPerformanceAnalysis(

IN segment\_category VARCHAR(50)

)

BEGIN

SET @sql = CONCAT('

SELECT

', segment\_category, ' AS segment\_category,

SUM(CASE WHEN loan\_status = ''Fully Paid'' THEN 1 ELSE 0 END) AS Fully\_Paid,

SUM(CASE WHEN loan\_status = ''Charged Off'' THEN 1 ELSE 0 END) AS Defaulted\_loan,

SUM(CASE WHEN loan\_status = ''Current'' THEN 1 ELSE 0 END) AS Active\_loan,

COUNT(\*) AS total\_loans,

CONCAT(ROUND(SUM(CASE WHEN loan\_status = ''Fully Paid'' THEN 1 ELSE 0 END) / COUNT(\*) \* 100, 2), ''%'') AS repayment\_rate,

ROUND((COUNT(CASE WHEN loan\_status = ''Charged Off'' THEN 1 END) / COUNT(\*)) \* 100, 2) AS default\_rate\_pct

FROM

loan\_data

GROUP BY

', segment\_category, '

ORDER BY

repayment\_rate DESC;

');

PREPARE stmt FROM @sql;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

END//

DELIMITER ;

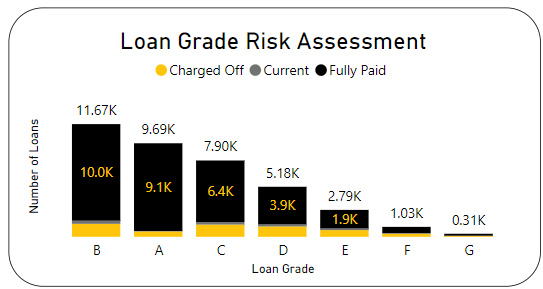
**Segment by Grade**:

Code: CALL LoanPerformanceAnalysis('grade');

Output:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Grade | Fully Paid Loans | Defaulted Loans | Active Loans | Total Loans | Repayment Rate | Default Rate |
| A | 9,102 | 552 | 35 | 9,689 | 93.94% | 5.70% |
| B | 10,004 | 1,343 | 327 | 11,674 | 85.69% | 11.50% |
| C | 6,381 | 1,266 | 257 | 7,904 | 80.73% | 16.02% |
| D | 3,894 | 1,072 | 216 | 5,182 | 75.14% | 20.69% |
| E | 1,920 | 691 | 175 | 2,786 | 68.92% | 24.80% |
| G | 198 | 98 | 17 | 313 | 63.26% | 31.31% |
| F | 646 | 311 | 71 | 1,028 | 62.84% | 30.25% |

Power BI Visual:



Observations:

* Loan grade strongly impacts repayment rates. Higher grades (A-C) have significantly higher repayment rates (>80%) and lower default rates (<17%) compared to lower grades (D-F).
* This suggests loan grade is a strong indicator of borrower creditworthiness and risk.

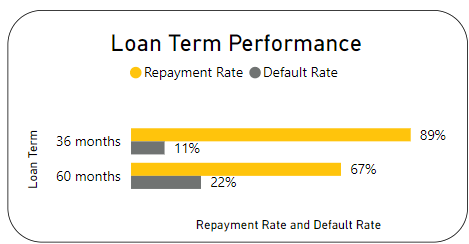
**Segment by Term**:

Coded: CALL LoanPerformanceAnalysis('term');

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Term | Fully Paid | Defaulted Loans | Active Loans | Total Loans | Repayment Rate | Default Rate |
| 36 | 25,214 | 3,023 | 0 | 28,237 | 89.29% | 10.71% |
| 60 | 6,931 | 2,310 | 1,098 | 10,339 | 67.04% | 22.34% |

Output:

Power BI Visual:



Observation: Shorter loans (36 months) have better repayment (89%) and lower defaults (11%) than longer loans (60 months).

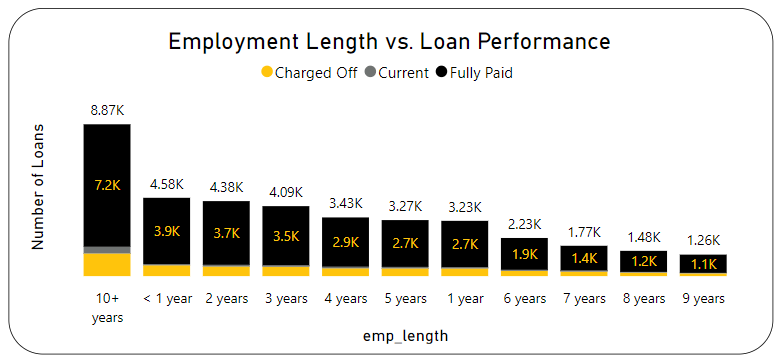
**Segment by emp\_length**:

Coded: CALL LoanPerformanceAnalysis('emp\_length');

Output:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Employment Length | Fully Paid Loans | Defaulted Loans | Active Loans | Total Loans | Repayment Rate | Default Rate (%) |
| 9 years | 1068 | 155 | 32 | 1255 | 85.10% | 12.35 |
| 2 years | 3724 | 561 | 97 | 4382 | 84.98% | 12.8 |
| … | … | … | … | … | … | … |
| 10+ years | 7157 | 1322 | 391 | 8870 | 80.69% | 14.9 |

Power BI Visual:



Observations:

Loan repayment rates hover around 82% to 85% across all employment lengths, with minimal impact on default rates. This suggests employment duration (within the observed range) has little influence on borrower creditworthiness.

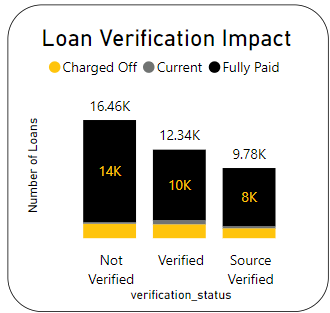
**Segment by verification\_status**:

Coded: CALL LoanPerformanceAnalysis('verification\_status');

Output:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Verification Status | Fully Paid Loans | Defaulted Loans | Active Loans | Total Loans | Repayment Rate | Default Rate |
| Not Verified | 14,229 | 2,015 | 220 | 16,464 | 86.42% | 12.24% |
| Source Verified | 8,098 | 1,382 | 297 | 9,777 | 82.83% | 14.14% |
| Verified | 9,818 | 1,936 | 581 | 12,335 | 79.59% | 15.70% |

Power BI Visual:



Observations:

Contrary to expectation, non-verified borrowers have the lowest default rate (12%) compare to verified and source-verified borrowers which show higher default rates (14-15%) and slightly lower repayment rates. Further analysis is needed to understand these counterintuitive findings.

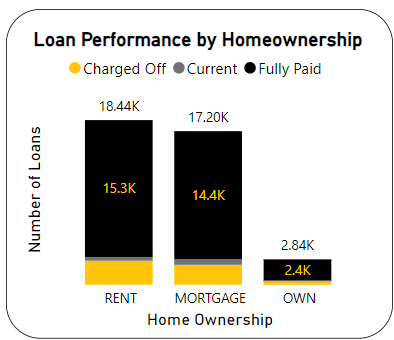
**Segment by home\_ownership**:

Coded: CALL LoanPerformanceAnalysis('home\_ownership');

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Homeownership | Total Loans | Fully Paid | Defaulted Loan | Active Loan | Repayment Rate | Default Rate |
| OWN | 2,838 | 2,370 | 397 | 71 | 83.51% | 13.99 |
| MORTGAGE | 17,198 | 14,350 | 2,231 | 617 | 83.44% | 12.97 |
| RENT | 18,439 | 15,342 | 2,687 | 410 | 83.20% | 14.57 |
| OTHER | 98 | 80 | 18 | 0 | 81.63% | 18.37 |
| NONE | 3 | 3 | 0 | 0 | 100.00% | 0 |

Output:

Power BI Visual:



Observations:

* Homeownership status has minimal impact on repayment rates (around 83%) across categories (OWN, MORTGAGE, RENT).
* Renters have a slightly higher default rate (14.57%) compared to homeowners, possibly due to income stability or investment in property.

Loan Repayment & Defaults by Segment Summary**:**

* Loan grade is a strong indicator of loan performance. Higher grades have lower default rates.
* Shorter loan terms also lead to better performance
* while employment length and homeownership have minimal impact within this data set.
* Interestingly, non-verified borrowers have the lowest default rate, requiring further investigation.

**1.2 Default Rates:** Identify future loan risk factors.

Default Rate = (Number of Defaulted Loans ÷ Total Number of Loans) × 100

Technique used - CASE statement with binning.

**Default Rates by Debt-to-income ratio**:

We'll first examine the relationship between a borrower's DTI ratio and default rates.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Statistic | Avg | Min | Max | Std | Q1 | Median | Q3 |
| Value | 0.13 | 0 | 0.3 | 0.07 | 0.08 | 0.13 | 0.19 |

DTI statistics:

Query: It segments loan data by DTI range (Low, Medium, High, Very High) and calculate the default rate for each segment.

SELECT

CASE

WHEN dti < 0.1 THEN 'Low (< 10%)'

WHEN dti < 0.15 THEN 'Medium (10% - 14%)'

WHEN dti < 0.2 THEN 'High (15% - 19%)'

ELSE 'Very High (> 20%)'

END AS dti\_range,

COUNT(\*) AS total\_borrowers,

COUNT(CASE WHEN loan\_status = 'Charged Off' THEN 1 END) AS charged\_off\_loans,

ROUND((COUNT(CASE WHEN loan\_status = 'Charged Off' THEN 1 END) / COUNT(\*)) \* 100, 2) AS default\_rate\_pct

FROM loan\_data

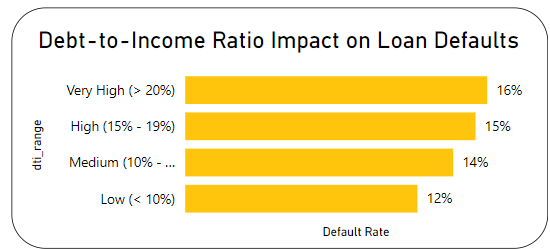
GROUP BY dti\_range

ORDER BY default\_rate\_pct desc;

|  |  |  |  |
| --- | --- | --- | --- |
| DTI Ratio Range | Total Borrowers | Charged-Off Loans | Default Rate |
| Very High (> 20%) | 7,291 | 1,137 | 15.59% |
| High (15% - 19%) | 8,864 | 1,329 | 14.99% |
| Medium (10% - 14%) | 9,646 | 1,335 | 13.84% |
| Low (< 10%) | 12,775 | 1,532 | 11.99% |

Output:

Power BI Visual:



Observation:

The default rate increases steadily as the DTI ratio increases. Borrowers with a very high DTI (over 20%) have the highest default rate (over 15%). This suggests that DTI can be a valuable indicator of loan risk.

**Default Rates by Annual Income**:

we'll explore the relationship between a borrower's annual income and default rates.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Statistic | Avg | Min | Max | std | Q1 | Median | Q3 |
| Value | 69,645 | 4,000 | 6,000,000 | 64,293 | 41,500 | 60,000 | 83,200 |

Annual Income statistics:

Analysis: Since income data can have a wide range, we defined income brackets for analysis.

Query: It segments the data by income bracket and calculates the avg DTI, # of borrowers, charged-off loans, and default rate for each bracket.

SELECT

CASE

WHEN annual\_income < 41500 THEN '< $41,500'

WHEN annual\_income < 60000 THEN '$41,500 - $59,999'

WHEN annual\_income < 83200 THEN '$60,000 - $83,199'

WHEN annual\_income < 100000 THEN '$83,200 - $99,999'

ELSE '$100,000+'

END AS annual\_income\_range,

COUNT(\*) AS total\_borrowers,

ROUND(AVG(dti\*100), 2) AS Avg\_DTI,

COUNT(CASE WHEN loan\_status = 'Charged Off' THEN 1 END) AS charged\_off\_loans,

ROUND((COUNT(CASE WHEN loan\_status = 'Charged Off' THEN 1 END) / COUNT(\*)) \* 100, 2) AS default\_rate\_pct

FROM loan\_data

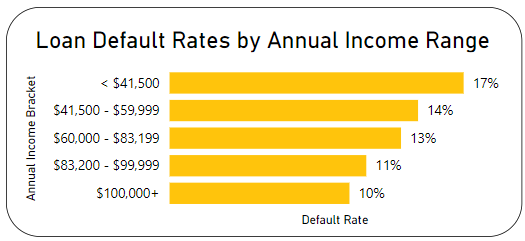
GROUP BY loan\_amount\_range

ORDER BY default\_rate\_pct desc;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Annual Income Range | Total Borrowers | Average DTI | Charged-Off Loans | Default Rate |
| Below $41,500 | 9,643 | 13.81 | 1,634 | 16.94% |
| $41,500 - $59,999 | 9,470 | 14.06 | 1,356 | 14.32% |
| $60,000 - $83,199 | 9,807 | 13.56 | 1,308 | 13.34% |
| $83,200 - $99,999 | 3,511 | 12.81 | 398 | 11.34% |
| Over $100,000 | 6,145 | 11.36 | 637 | 10.37% |

Output:

Power BI Visual:



Observation:

There seems to be an inverse relationship between annual income and default rates. Borrowers with lower incomes (< $41,500) have the highest default rate (16.94%), while those with the highest income bracket ($100,000+) have the lowest default rate (10.37%). However, the average DTI also follows a similar pattern.

**Default Rates by loan\_amount**:

Finally, we'll analyze default rates based on the loan amount.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Statistic | Avg | Min | Max | std | Q1 | Median | Q3 |
| Value | 11296 | 500 | 35000 | 7461 | 5500 | 10000 | 15000 |

Loan Amount Statistics:

Query: This query buckets loans by amt range, calcs borrower counts, defaults, default rates, and sorts by highest default rate.

SELECT

CASE

WHEN loan\_amount < 5500 THEN 'Less than $5.5K'

WHEN loan\_amount < 10000 THEN '$5.5K - $10K'

WHEN loan\_amount < 15000 THEN '$10K - $15K'

WHEN loan\_amount < 20000 THEN '$15K - $20K'

WHEN loan\_amount < 25000 THEN '$20K - $25K'

WHEN loan\_amount < 30000 THEN '$25K - $30K'

ELSE 'More than $30K'

END AS loan\_amount\_range,

COUNT(\*) AS total\_borrowers,

COUNT(CASE WHEN loan\_status = 'Charged Off' THEN 1 END) AS charged\_off\_loans,

ROUND((COUNT(CASE WHEN loan\_status = 'Charged Off' THEN 1 END) / COUNT(\*)) \* 100, 2) AS default\_rate\_pct

FROM bank\_loan\_analysis.loan\_data

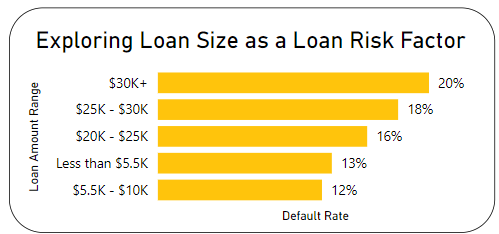
GROUP BY loan\_amount\_range

ORDER BY default\_rate\_pct desc;

Output:

|  |  |  |  |
| --- | --- | --- | --- |
| Loan Amount Range | Total Borrowers | Charged-Off Loans | Default Rate |
| More than $30K | 1,186 | 241 | 20.32% |
| $25,000 - $30,000 | 1,766 | 318 | 18.01% |
| $15,000 - $20,000 | 4,770 | 766 | 16.06% |
| $20,000 - $25,000 | 3,181 | 499 | 15.69% |
| Less than $5,500 | 9,418 | 1,228 | 13.04% |
| $10,000 - $15,000 | 8,743 | 1,111 | 12.71% |
| $5,500 - $10,000 | 9,512 | 1,170 | 12.30% |

Power BI Visual:



Observation:

The default rate appears to be higher for loans with larger amounts (more than $25,000) compared to smaller loan amounts (less than $5,500). This suggests a potential risk factor associated with higher loan amounts due to a higher financial burden.

Default Rate Analysis Summary:

* This analysis identified DTI ratio and loan amount as significant factors influencing loan default risk.
* Borrowers with higher DTI and larger loan amounts are more likely to default on their loans.
* The relationship between annual income & default rates is less clear and might be influenced by DTI.

**1.3 Correlation Analysis:** Relationships between numerical variables.

Technique used: Pearson correlation coefficient to measures how linear two variables are.

The correlation coefficient ranges from -1 to 1, where:

1 is perfect positive correlation, -1 is perfect negative correlation and 0 is no correlation.

Formula and Calculation: r = cov(X, Y) ÷ (std(X) \* std(Y))

Where cov(X, Y) = AVG(x \* y) - AVG(x) \* AVG(y)

Query Approach: SELECT (AVG(x \* y) - AVG(x) \* AVG(y)) / (STD(x) \* STD(y)) AS correlation FROM table\_name;

Implementation:

SELECT

(AVG(loan\_amount \* (CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END)) - AVG(loan\_amount) \* AVG(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END)) / (STD(loan\_amount) \* STD(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END)) AS correlation\_loan\_amount\_default\_rate,

(AVG(dti \* (CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END)) - AVG(dti) \* AVG(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END)) / (STD(dti) \* STD(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END)) AS correlation\_dti\_default\_rate,

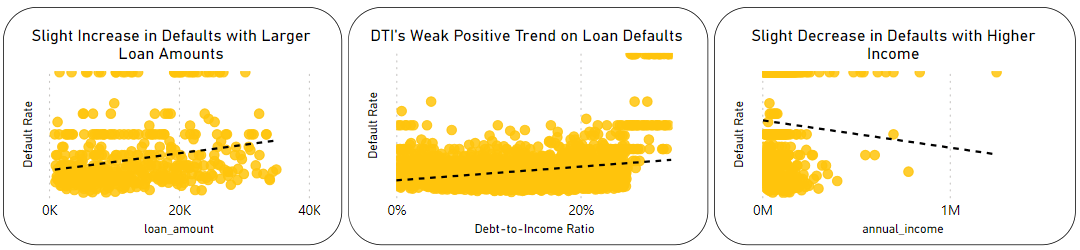
(AVG(annual\_income \* (CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END)) - AVG(annual\_income) \* AVG(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END)) / (STD(annual\_income) \* STD(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END)) AS correlation\_annual\_income\_default\_rate

FROM loan\_data;

|  |  |  |
| --- | --- | --- |
| Correlation (Loan Amount, Default Rate) | Correlation (DTI, Default Rate) | Correlation (Income, Default Rate) |
| 0.053 | 0.041 | -0.038 |

Output:

Power BI Visual:



Overall Interpretation:

* All three correlation coefficients are very close to zero, indicating weak relationships between the borrower characteristics and default rates.
* This suggests that loan amount, DTI, & annual income, in isolation, are not very strong predictors of loan default on their own. There are likely other factors that play a more significant role in default risk.

Key takeaways from Loan Performance & Risk Assessment:

* Higher loan grades (A-C) & short terms (36 months) best for repayment.
* Employment length & homeownership have little impact. Surprisingly, non-verified borrowers default less (needs more study).
* High DTI & loan amount correlate with increased default risks. Lower incomes also associate with higher default rates.
* Focus on loan grade & term for performance, DTI & loan size for risk.

**2. Loan Product & Portfolio Analysis**

Focuses on the loan products themselves and how to optimize the loan portfolio for profitability.

* 1. **Loan Popularity:**

Uncover popular loan terms & amounts, then see how these trends change over time.

2.1.1 Loan Popularity by Term lengths:

Loan Term popularity based on Loan Amount Issue to borrowers.

Query: It calculates loan amount quartiles for each term and then summarizes key loan amount statistics for each term (total loan %, borrower preference, distribution: avg, min, max, quartiles).

MySQL Techniques: used Window Function, Subquery, Aggregation & Summarization, Case Statement.

SELECT term,

SUM(loan\_amount) / (SELECT SUM(loan\_amount) FROM loan\_data) \* 100 AS Total\_Loan\_Pct,

ROUND(COUNT(\*) / (SELECT COUNT(\*) FROM loan\_data) \* 100, 2) AS Total\_Borrowers\_Pct,

SUM(loan\_amount) AS Total\_Loan,

AVG(loan\_amount) AS Avg\_loan,

MIN(loan\_amount) AS Min\_loan,

MAX(loan\_amount) AS Max\_loan,

MAX(CASE WHEN quartile = 1 THEN loan\_amount END) AS Q1\_Loan,

MAX(CASE WHEN quartile = 2 THEN loan\_amount END) AS Median,

MAX(CASE WHEN quartile = 3 THEN loan\_amount END) AS Q3\_Loan

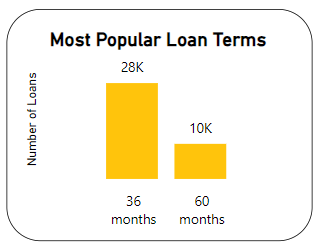
FROM (SELECT term, loan\_amount, NTILE(4) OVER (PARTITION BY term ORDER BY loan\_amount) AS quartile FROM loan\_data ) AS quartiles

GROUP BY term;

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Term | Total Loan % | Total Borrowers % | Total Loan Amount | Avg | Min | Max | Q1 | Median | Q3 |
| 36 Months | 62.66% | 73.20% | 273,041,225 | 9,669.62 | 500 | 35,000 | 5,000 | 8,000 | 12,500 |
| 60 Months | 37.34% | 26.80% | 162,715,850 | 15,738.06 | 1,000 | 35,000 | 9,500 | 15,000 | 20,675 |

Output:

Power BI Visual:



Observation:

Most loans (73%) are for 36 months, even though the average loan amount is smaller than 60-month loans ($9,669 vs. $15,738). This suggests borrowers prefer faster repayment, possibly due to lower interest rates or wanting to be out of debt quicker.

2.1.2 Most popular Loan Product

Identifying the top 10 loan amounts offered by the lender.

Query: This query finds the top 10 loan amounts by count and percentage issued.

MySQL Techniques: Group By, Window Function(subquery with aggregation).

SELECT loan\_amount, COUNT(\*) AS count,

ROUND((COUNT(\*) / (SELECT COUNT(\*) FROM loan\_data)) \* 100, 2) AS percentage\_share

FROM loan\_data

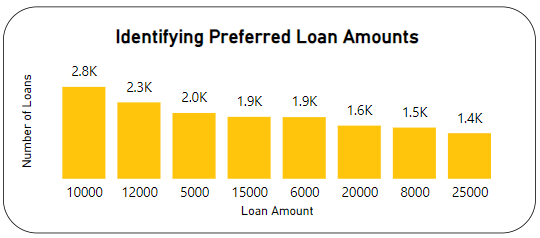
GROUP BY loan\_amount

ORDER BY count DESC LIMIT 10;

|  |  |  |
| --- | --- | --- |
| Loan Amount | Count | % Share |
| 10,000 | 2,761 | 7.16% |
| 12,000 | 2,295 | 5.95% |
| … | … | … |
| 7,000 | 989 | 2.56% |

Output:

Power BI Visual:



Observation:

Loan analysis shows borrowers prefer moderate amounts ($10k-$12k) and smaller sizes ($5k-$8k), suggesting affordability focus. Lower demand for larger loans could be due to stricter requirements or a preference to avoid debt.

2.1.3 Loan Popularity Over Time (Monthly Trend Analysis)

Investigates how loan issuance trends vary over time by analyzing the number of loans issued each month.

Query: It extracts year-month (YYYY-MM) from issue date and counts loans per month.

MySQL Techniques: Time-based extraction, Grouping, Aggregation.

SELECT DATE\_FORMAT(issue\_date, '%Y-%m') AS month, COUNT(\*) AS count

FROM loan\_data

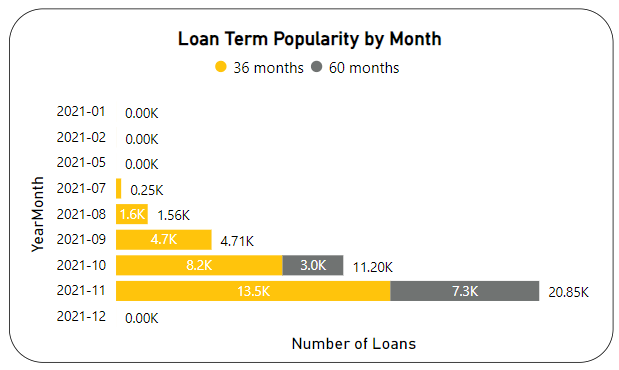
GROUP BY DATE\_FORMAT(issue\_date, '%Y-%m')

ORDER BY month;

|  |  |
| --- | --- |
| Month | Count |
| 2021-01 | 2 |
| 2021-02 | 4 |
| 2021-05 | 1 |
| … | … |
| 2021-12 | 1 |

Output:

Power BI Visual:



Observation: The data shows a rise in loan applications from July 2021 onwards, with a peak in November.

Loan Popularity Summary:

* Borrowers prioritize affordability with smaller loans ($10k-$12k) and faster repayment (36 months).
* Loan applications skyrocketed July-Nov 2021, suggesting seasonal shifts or product changes.

**2.2 Interest Rate Analysis:** Exploration of interest rates and their impact.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Avg | Min | Max | std | Q1 | Median | Q3 |
| 12% | 5.42% | 24.59% | 4% | 9.32% | 11.86% | 14.59% |

Interest Rate Statistics:

2.2.1 Average Interest Rates by Loan Grade and Term:

The goal is to understand how creditworthiness (reflected by loan grade) and loan term length influence the interest rate borrowers receive.

Query: This query calculates avg. interest rate by borrower grade & loan term.

SELECT grade, term, ROUND(AVG(int\_rate\*100),2) AS avg\_interest\_rate

FROM loan\_data

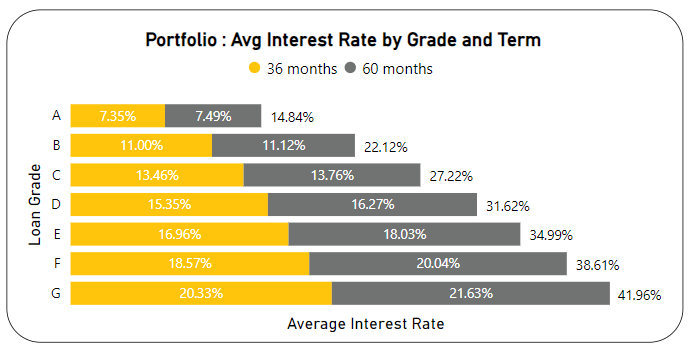
GROUP BY grade, term

ORDER BY grade, term;

|  |  |  |
| --- | --- | --- |
| Grade | Term (Months) | Avg Int Rate |
| A | 36 | 7.35% |
| A | 60 | 7.49% |
| B | 36 | 11% |
| … | … | … |
| G | 60 | 21.63% |

Output:

Power BI Visual:



Observation:

We see a clear trend, Borrowers with lower creditworthiness (higher grade) leads to higher interest rates. Longer terms (60 vs 36 months) see slightly higher rates, confirming the borrower risk-term-interest rate connection.

2.2.2 Exploring Interest Rate vs. Loan Performance:

See how interest rates affect loan repayment (default vs success).

Query: This query groups loans by int rate buckets and calcs loan status within each range. Finally, shows total loans per range.

MySQL Techniques: CASE statement with binning (Conditional aggregation)

SELECT

CASE

WHEN int\_rate < 0.0932 THEN 'Less than 9.32%'

WHEN int\_rate < 0.1186 THEN '9.32% - 11.85%'

WHEN int\_rate < 0.1459 THEN '11.86% - 14.58%'

ELSE 'Above 14.59%'

END AS int\_rate\_group,

SUM(CASE WHEN loan\_status = 'Fully Paid' THEN 1 ELSE 0 END) AS fully\_paid\_count,

SUM(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END) AS charged\_off\_count,

SUM(CASE WHEN loan\_status = 'Current' THEN 1 ELSE 0 END) AS current\_count,

count(\*) AS Total\_Loan

FROM loan\_data

GROUP BY

int\_rate\_group

ORDER BY

CASE

WHEN int\_rate\_group = 'Less than 9.32%' THEN 1

WHEN int\_rate\_group = '9.32% - 11.85%' THEN 2

WHEN int\_rate\_group = '11.86% - 14.58%' THEN 3

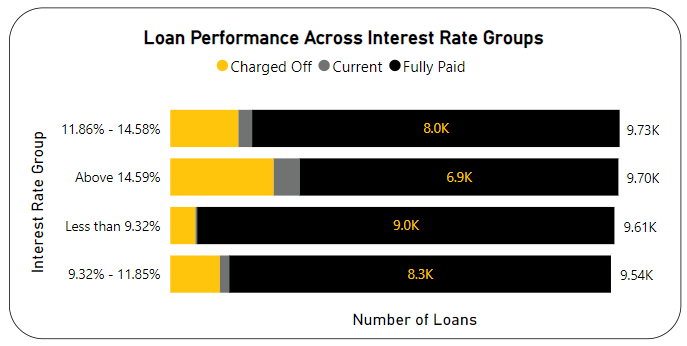
WHEN int\_rate\_group = 'Above 14.59%' THEN 4

END;

Output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Int Rate Group | Fully Paid Loans | CO Loans | Current Loans | Loan Counts |
| Less than 9.32% | 9,031 | 545 | 37 | 9,613 |
| 9.32% - 11.85% | 8,262 | 1,074 | 202 | 9,538 |
| 11.86% - 14.58% | 7,955 | 1,475 | 296 | 9,726 |
| Above 14.59% | 6,897 | 2,239 | 563 | 9,699 |

Power BI Visual:



Observation:

Lower interest rates (better credit) correlate with higher repayment rates, while higher rates (lower credit) see more defaults, confirming a creditworthiness-interest rate-performance link.

**2.3 Loan Purpose Analysis:**

Analyze top loan purposes and their correlation with loan amount and borrower profile.

MySQL Techniques: Descriptive statistics with group by.

Query: This query groups loans by purpose, calculates Loan Purpose Statistics (%, size, income, DTI, rate).

SELECT

purpose,

ROUND(COUNT(\*) / (SELECT COUNT(\*) FROM loan\_data) \* 100, 2) AS percentage\_total,

COUNT(\*) AS count,

AVG(loan\_amount) AS avg\_loan\_amount,

AVG(annual\_income) AS avg\_annual\_income,

AVG(dti\*100) AS avg\_dti,

ROUND(AVG(int\_rate\*100), 4) AS avg\_int\_rate

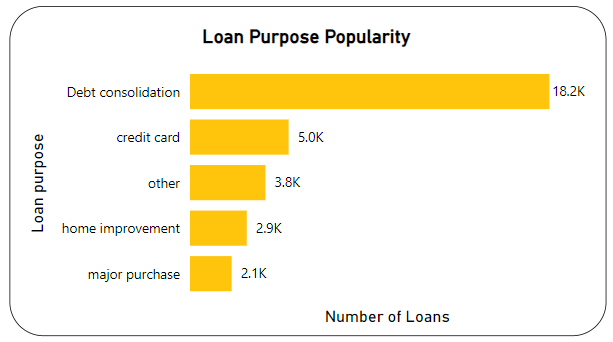
FROM loan\_data

GROUP BY purpose ORDER BY percentage\_total DESC;

Output:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Purpose | Loans % | Loan Count | Avg Loan Amount | Avg Annual Income | Avg DTI | Avg Int Rate |
| Debt consol. | 47.22% | 18,214 | $12,762.69 | $67,846.56 | 14.48% | 12.50% |
| Credit Card | 12.96% | 4,998 | $11,781.75 | $70,994.44 | 14.63% | 11.73% |
| … | … | … | … | … | … | … |
| Renewable En | 0.24% | 94 | $8,997.34 | $81,331.05 | 11.82% | 11.50% |

Power BI Visual:



Observation:

* Loans for larger needs (debt consolidation, home improvement) are bigger and used by borrowers with higher income.
* Small business loans have higher interest rates due to risk, while car loans and major purchases have lower rates potentially due to shorter terms or collateral.
* DTI ratio doesn't significantly differ across loan purposes.

**2.4 Profitability Analysis:**

Find most profitable loans by analyzing total profit against loan\_amount, interest rate, term.

2.4.1 Calculating profit

Quick view of the bank's loan portfolio profitability.

Query: This query summarizes loans - total value, interest profit, loan count, and overall margin.

SELECT

FORMAT(SUM(loan\_amount),0) AS total\_loan,

FORMAT(SUM(total\_payment - loan\_amount),0) AS profit\_earned,

FORMAT(COUNT(\*),0) AS num\_loans,

ROUND(SUM(total\_payment - loan\_amount) / SUM(loan\_amount) \* 100, 2) AS profit\_pct

FROM loan\_data;

|  |  |  |  |
| --- | --- | --- | --- |
| Total Loan Amount | Profit Earned | Number of Loans | Profit Percentage |
| 435,757,075 | 37,313,858 | 38,576 | 8.56% |

Output:

Observation: Loans generated $37.3M profit on a $435.8M portfolio (8.56% margin).

2.4.2 Profit vs term analysis

Understand which loan terms are most profitable.

MySQL Techniques: aggregation with grouping.

Query: It summarizes loan data by term, displaying total amount, profit, # loans, and profit %, sorted by most profitable.

SELECT term,

FORMAT(SUM(loan\_amount),0) AS total\_loan,

FORMAT(SUM(total\_payment - loan\_amount),0) AS profit\_earned,

FORMAT(COUNT(\*),0) AS num\_loans,

ROUND(SUM(total\_payment - loan\_amount) / SUM(loan\_amount) \* 100, 2) AS Profit\_Pct

FROM loan\_data

GROUP BY term

ORDER BY profit\_earned DESC;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Term | Total Loan Amount | Profit Earned | Number of Loans | Profit Percentage |
| 36 Months | $273,041,225 | $21,668,233 | 28,237 | 7.94% |
| 60 Months | $162,715,850 | $15,645,625 | 10,339 | 9.62% |

Output:

Observation: 60-month loans yield higher profits per loan (9.62% vs 7.94%) despite lower loan volumes ($162M vs $273M).

2.4.3 Most profitable interest rate. Top 10

Identify which interest rates generate the highest overall profit.

MySQL Techniques: aggregation with grouping.

Query: It groups loans by int. rate, calculates profit metrics (total loan, profit, # loans, profit %), and sort them by highest profit.

SELECT

ROUND(int\_rate \* 100, 2) AS interest\_rate,

FORMAT(SUM(loan\_amount), 0) AS total\_loan,

SUM(total\_payment - loan\_amount) AS profit\_earned,

COUNT(\*) AS num\_loans,

ROUND(SUM(total\_payment - loan\_amount) / SUM(loan\_amount) \* 100, 2) AS profit\_pct

FROM loan\_data

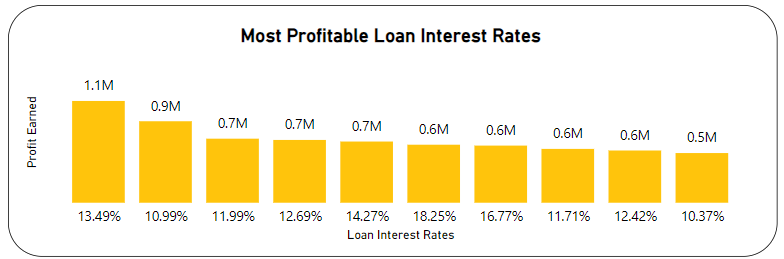
GROUP BY interest\_rate

ORDER BY profit\_earned DESC LIMIT 10;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| interest\_rate | total\_loan | profit\_earned | num\_loans | profit\_pct |
| 13.49 | 10,510,925 | 1,083,940 | 811 | 10.31 |
| 10.99 | 10,767,450 | 867,942 | 932 | 8.06 |
| … | … | … | … | … |
| 10.37 | 4,768,175 | 529,162 | 452 | 11.1 |

Output:

Power BI Visual:



Observation:

Higher interest rates generally lead to higher profit percentages. The sweet spot for profitability and loan volume is around 11-14% interest rate. There's a trade-off between profit percentage and loan volume. Balancing interest rates is key to maximizing profits.

2.4.4 Most profitable loan product. Top 10

MySQL Techniques: aggregation with grouping.

Query: This query groups the loan amount, calculates profit metrics (total loan, profit, # loans, profit %), and sorts by highest total profit, returning only the top 10.

SELECT

loan\_amount AS loan\_amount,

FORMAT(SUM(loan\_amount), 0) AS total\_loan,

SUM(total\_payment - loan\_amount) AS profit\_earned,

FORMAT(COUNT(\*), 0) AS num\_loans,

ROUND(SUM(total\_payment - loan\_amount) / SUM(loan\_amount) \* 100, 2) AS profit\_pct

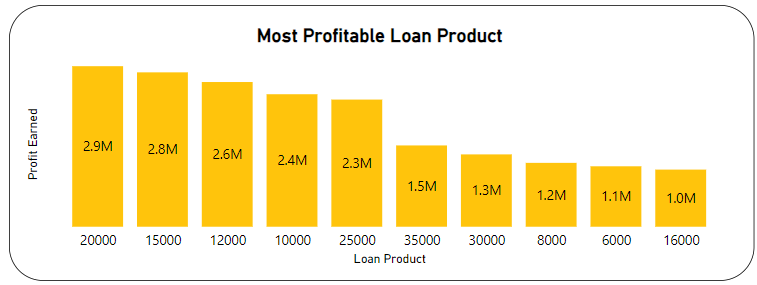
FROM loan\_data

GROUP BY loan\_amount ORDER BY profit\_earned DESC LIMIT 10;

Output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Loan Amount | Total Loan | Profit Earned | Number of Loans | Profit Percentage |
| $20,000 | $31,940,000 | $2,922,135 | 1,597 | 9.15% |
| $15,000 | $27,900,000 | $2,809,881 | 1,860 | 10.07% |
| … | … | … | … | … |
| $16,000 | $12,960,000 | $1,039,868 | 810 | 8.02% |

Power BI Visual:



Observation:

* $12k, $15k, $20k loans most profitable with good volume, generating the highest profit earnings.
* $30k loans also highly profitable despite lower volume, indicating high profit per loan.

Profitability Analysis Summary:

* The loan portfolio generates an 8.56% profit margin.
* Best loans: 60 months, 11-14% interest, $12k-$20k (volume & margin), $30k (high profit per loan).
  1. **Target Borrower Analysis:** Profiling profitable borrower segments.

Based on the findings on loan performance and risk, identify the characteristics of borrowers the bank should target for future loan approvals to maximize profit and minimize risk.

2.5.1 Identify borrower characteristics associated with low-risk and high-profit loans:

Low-risk loans are those that have been fully paid and have a low default rate, while high-profit loans are those with most profit.

MySQL Techniques: conditional aggregation, calculation of descriptive statistics.

Query: This query groups loans by borrower characteristics and calculates averages, counts. It sorts by most fully paid (low risk), least defaults (low risk), and most interest earned (high profit).

SELECT

grade, emp\_length, home\_ownership, verification\_status,

ROUND(AVG(annual\_income), 2) AS avg\_annual\_income,

ROUND(AVG(dti), 4) AS avg\_dti,

ROUND(AVG(int\_rate), 4) AS avg\_int\_rate,

COUNT(\*) AS num\_loans,

SUM(CASE WHEN loan\_status = 'Fully Paid' THEN 1 ELSE 0 END) AS fully\_paid\_count,

SUM(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END) AS charged\_off\_count,

SUM(total\_payment - loan\_amount) AS total\_interest\_earned

FROM loan\_data

WHERE loan\_status IN ('Fully Paid', 'Charged Off')

GROUP BY grade, emp\_length, home\_ownership, verification\_status

ORDER BY fully\_paid\_count DESC, charged\_off\_count ASC, total\_interest\_earned DESC;

Output: Top 5

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Grade | Empl Length | Home Own. | Verif. Status | Avg. Income | Avg. DTI | Avg. Int Rate | Loans | F.P | CO | Profit |
| A | 10+ years | MORTGAGE | Not Verified | 73,237.27 | 0.1178 | 0.0715 | 774 | 729 | 45 | 3,23,702 |
| B | 10+ years | MORTGAGE | Verified | 97,376.82 | 0.1405 | 0.1111 | 615 | 533 | 82 | 6,47,288 |
| B | 10+ years | MORTGAGE | Not Verified | 77,084.00 | 0.1318 | 0.1089 | 590 | 526 | 64 | 4,78,800 |
| A | 10+ years | MORTGAGE | Verified | 88,756.19 | 0.1269 | 0.0717 | 454 | 431 | 23 | 1,80,396 |
| B | 2 years | RENT | Not Verified | 50,398.48 | 0.1328 | 0.1104 | 433 | 382 | 51 | 2,41,969 |

Observations:

* Ideal borrowers: A/B grades, long work history, homeowners (any verification).
* Grade B borrowers with shorter employment and renting homes exhibit moderate potential.

2.5.2 Identify the most profitable loan segments based on purpose and term:

MySQL Techniques: Aggregation with grouping.

Query: It groups loans by purpose/term and calc. counts and profit, sorts by most fully paid loans, low defaults, then most profit.

SELECT

purpose, term, COUNT(\*) AS num\_loans,

SUM(CASE WHEN loan\_status = 'Fully Paid' THEN 1 ELSE 0 END) AS fully\_paid\_count,

SUM(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END) AS charged\_off\_count,

SUM(total\_payment - loan\_amount) AS total\_interest\_earned

FROM loan\_data

GROUP BY purpose, term

ORDER BY fully\_paid\_count DESC, charged\_off\_count ASC, total\_interest\_earned DESC;

Output: Top 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose | Term | Loans | Fully Paid | Charged-Off | Total Int Earned |
| Debt Consolidation | 36 | 12,823 | 11,432 | 1,391 | $12,279,507 |
| Credit Card | 36 | 3,982 | 3,686 | 296 | $4,114,553 |
| Debt Consolidation | 60 | 5,391 | 3,564 | 1,260 | $9,062,689 |
| Other | 36 | 3,009 | 2,626 | 383 | $1,326,409 |
| Home Improvement | 36 | 2,009 | 1,818 | 191 | $1,250,687 |

Observation:

* Best loans: Debt consolidation & home improvement (short terms).
* Riskier: Small business & longer terms.
* Least profitable: Vacation & education loans.

2.5.3 Identify the most profitable loan segments based on income and debt-to-income ratio:

MySQL Technique used - CASE statement with binning.

Query: It creates income/DTI brackets, groups loans by it, calc. counts, prof., and sorts most paid, least defaults, then most prof.

SELECT

CASE

WHEN annual\_income < 41500 THEN '< $41,500'

WHEN annual\_income < 60000 THEN '$41,500 - $59,999'

WHEN annual\_income < 83200 THEN '$60,000 - $83,199'

WHEN annual\_income < 100000 THEN '$83,200 - $99,999'

ELSE '$100,000+'

END AS income\_bracket,

CASE

WHEN dti < 0.1 THEN 'Low (< 10%)'

WHEN dti < 0.15 THEN 'Medium (10% - 14%)'

WHEN dti < 0.2 THEN 'High (15% - 19%)'

ELSE 'Very High (> 20%)'

END AS dti\_bracket,

COUNT(\*) AS num\_loans,

SUM(CASE WHEN loan\_status = 'Fully Paid' THEN 1 ELSE 0 END) AS fully\_paid\_count,

SUM(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END) AS charged\_off\_count,

SUM(total\_payment - loan\_amount) AS total\_interest\_earned

FROM loan\_data

GROUP BY income\_bracket, dti\_bracket

ORDER BY fully\_paid\_count DESC, charged\_off\_count ASC, total\_interest\_earned DESC;

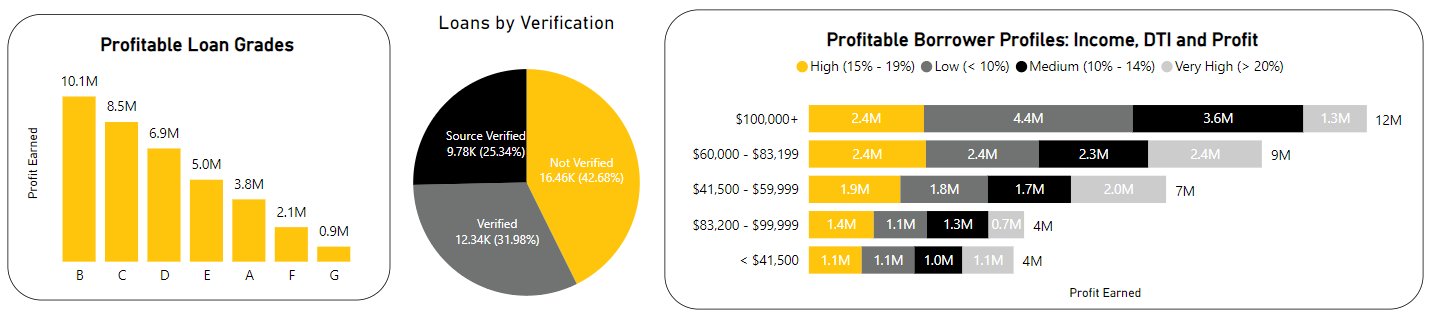
Output: Top 5

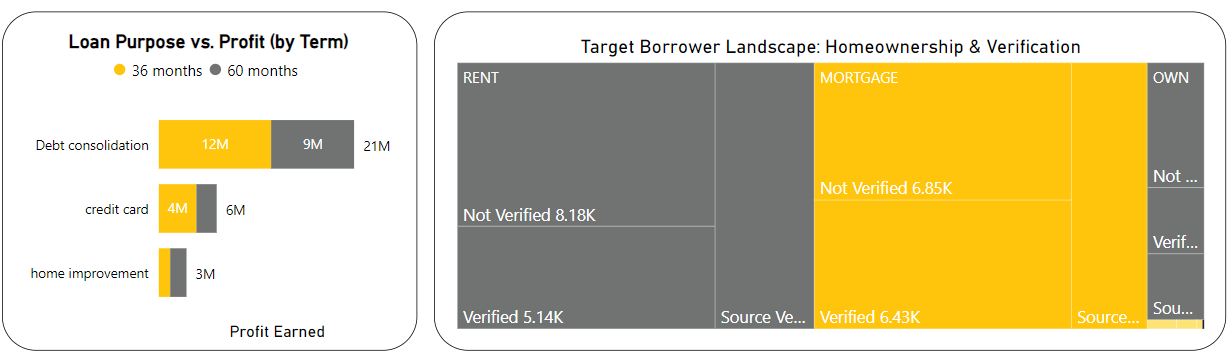
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Income Bracket | DTI Bracket | Number of Loans | Fully Paid Loans | Charged-Off Loans | Total Interest Earned |
| $60,000 - $83,199 | Low (< 10%) | 3,048 | 2,616 | 360 | $2,356,129 |
| < $41,500 | Low (< 10%) | 3,030 | 2,525 | 469 | $1,101,234 |
| $100,000+ | Low (< 10%) | 2,747 | 2,420 | 241 | $4,356,380 |
| $41,500 - $59,999 | Low (< 10%) | 2,716 | 2,318 | 338 | $1,823,171 |
| $60,000 - $83,199 | Medium (10% - 14%) | 2,555 | 2,163 | 333 | $2,284,185 |

Observation:

* Low income with good DTI (below 10%) profitable across the board. DTI seems more important than income for profitability.
* High DTI (>15%) risky for all income groups. Target lower income borrowers with strong DTI.

Power BI Visuals:





Target Borrower Analysis Summary:

* Ideal borrowers: A/B Grade, long work history, own a home regardless of income verification.
* Focus on debt consolidation, credit card & home improvement loans (shorter terms).
* Low income with good DTI profitable, DTI more important than income. Avoid high DTI borrowers.

Key Takeaways from Loan Product & Portfolio Analysis:

* Borrowers prefer smaller ($10k-$12k), faster-to-pay (36-months) loans.
* Higher creditworthiness leads to lower interest rates and better repayment.
* Debt consolidation loans with shorter terms are profitable and less risky.
* Target borrowers with good credit history, home ownership, and low DTI for optimal profit.

**3.** **Geographical Analysis**

Investigates how loan characteristics and performance differ across geographical locations.

3.1 Analyze the distribution of loans across different states (address\_state).

MySQL Techniques: Aggregation and window function.

Query: This query groups loans by state, calculates counts & percentage, sorted by most loans first.

SELECT address\_state, COUNT(\*) AS total\_loans,

COUNT(\*)/(SELECT COUNT(\*) FROM loan\_data) \* 100 AS pct\_share

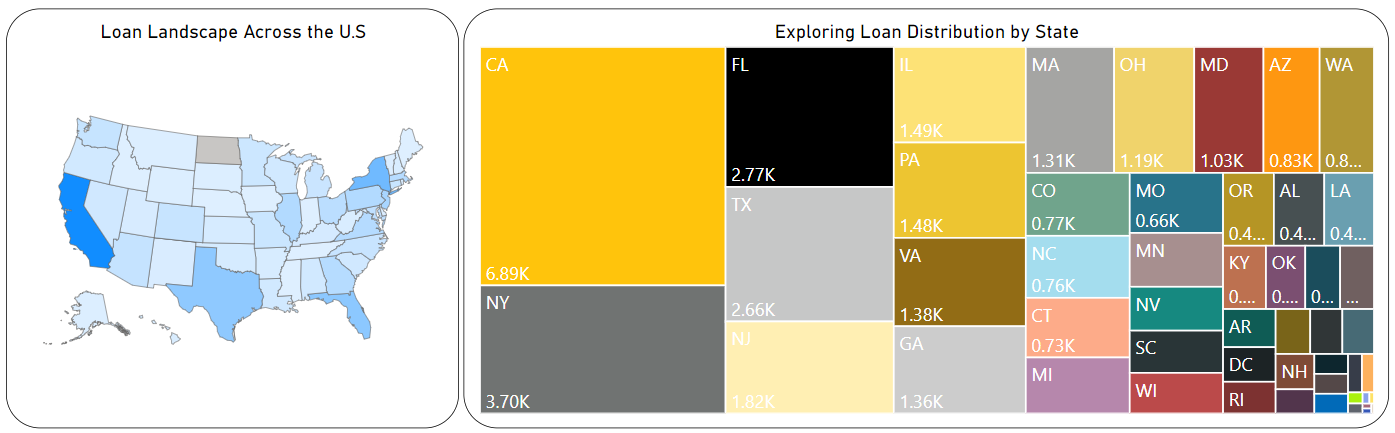
FROM loan\_data

GROUP BY address\_state ORDER BY total\_loans DESC;

Output:

|  |  |  |
| --- | --- | --- |
| **address\_state** | **total\_loans** | **pct\_share** |
| CA | 6894 | 17.8712 |
| NY | 3701 | 9.594 |
| FL | 2773 | 7.1884 |
| … | … | … |
| NE | 5 | 0.013 |
| IA | 5 | 0.013 |
| ME | 3 | 0.0078 |

Power BI Visuals:



Observation:

* CA leads (18%) followed by NY, FL, TX.
* Most states have 1-4% of loans.
* States like Maine, Nebraska, Iowa & some others have the lowest loan counts.

3.2 Analyze loan performance (loan\_status) based on address\_state.

This can reveal geographical trends in loan risk.

MySQL Techniques: Aggregation with conditional logic.

Query: This query groups loans by state, count loans, % total/repaid/charged-off, Avg.amt/int rate/DTI. Sorted by most loans first.

SELECT

address\_state states, COUNT(\*) AS loan\_count,

COUNT(\*)/(SELECT COUNT(\*) FROM loan\_data)\*100 AS total\_loan\_pct,

ROUND(SUM(CASE WHEN loan\_status = 'Fully Paid' THEN 1 ELSE 0 END) \* 100.0 / COUNT(\*), 2) AS fully\_paid\_pct,

ROUND(SUM(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END) \* 100.0 / COUNT(\*), 2) AS charged\_off\_pct

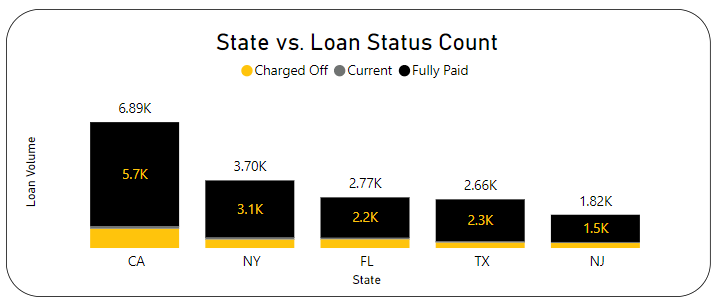
FROM loan\_data

GROUP BY address\_state ORDER BY loan\_count DESC;

Output: Top 10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **states** | **loan\_count** | **total\_loan\_pct** | **fully\_paid\_pct** | **charged\_off\_pct** |
| CA | 6894 | 17.8712 | 82.59 | 15.33 |
| NY | 3701 | 9.594 | 84.38 | 12.65 |
| FL | 2773 | 7.1884 | 79.77 | 17.27 |
| … | … | … | … | … |
| NE | 5 | 0.013 | 40 | 60 |
| IA | 5 | 0.013 | 100 | 0 |
| ME | 3 | 0.0078 | 100 | 0 |

Power BI Visual:



Observation:

* DC, DE top for fully paid loans.
* NV, AK, NE, SD charged-off loans higher.
* CA, NY, FL lead in loan counts.
* Balanced: NY, TX, PA, MA, OH, MD.

3.3 Analyze the profitability across geographical locations.

MySQL Techniques: Aggregate Query with Group By.

Query: It groups loans by state, calc. loan counts, total amt, profit & %, and profit share per state. Sorts by highest profit earned.

SELECT address\_state, COUNT(\*) AS loan\_count,

SUM(loan\_amount) Total\_loan\_issued,

SUM(total\_payment - loan\_amount) AS profit\_earned,

ROUND(SUM(total\_payment - loan\_amount) / SUM(loan\_amount) \* 100, 2) AS Profit\_Pct,

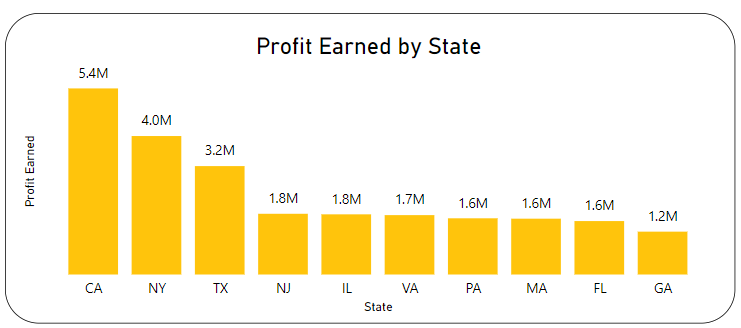
ROUND((SUM(total\_payment - loan\_amount) / (SELECT SUM(total\_payment - loan\_amount) FROM loan\_data)) \* 100, 2) AS profit\_percentage\_share

FROM loan\_data GROUP BY address\_state ORDER BY profit\_earned DESC;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **state** | **loan\_count** | **Total\_loan\_issued** | **profit\_earned** | **Profit\_%** | **profit\_%\_share** |
| CA | 6894 | 78484125 | 5417109 | 6.9 | 14.52 |
| NY | 3701 | 42077050 | 4031131 | 9.58 | 10.8 |
| TX | 2664 | 31236650 | 3156065 | 10.1 | 8.46 |
| … | … | … | … | … | … |
| IN | 9 | 86225 | -704 | -0.82 | 0 |
| NE | 5 | 31700 | -7158 | -22.58 | -0.02 |
| TN | 17 | 162175 | -20653 | -12.74 | -0.06 |

Output:

Power BI Visual:



Observation:

* Top profit earners (CA, NY, TX) differ from highest profit percentages (WY, DE, TX).
* NV has low profit percentage(2.7%) despite moderate volume(482).
* IN, NE, TN have negative profits, indicating a net loss.

Geographical Analysis Summary:

* CA leads in volume (18%) followed by NY, FL. Most others have 1-4% of loans.
* Repayment: High (DC, DE), Low (NV, AK, NE, SD). Balanced (NY, TX, etc.).
* Profit: Top earners (CA, NY, TX), highest profit % (WY, DE). Negative profit (IN, NE, TN).

**4. Time-based Analysis**

Looks at how loan performance and other metrics change over time.

Monthly performance Trends:

MySQL Techniques: Time-based grouping, Aggregation, Conditional Aggregation.

Query: It extracts year/month, counts total loans, fully paid/default loans, calculates fully paid %, default %, avg int rate/dti. Grouped by year/month and sorted by month.

SELECT

YEAR(issue\_date) AS year,

MONTH(issue\_date) AS month,

COUNT(\*) AS total\_loans,

SUM(CASE WHEN loan\_status = 'Fully Paid' THEN 1 ELSE 0 END) AS fully\_paid\_loans,

SUM(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END) AS charged\_off\_loans,

ROUND((SUM(CASE WHEN loan\_status = 'Fully Paid' THEN 1 ELSE 0 END) / COUNT(\*)) \* 100, 2) AS repayment\_rate,

ROUND((SUM(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END) / COUNT(\*)) \* 100, 2) AS default\_rate,

ROUND(AVG(int\_rate\*100),2) AS avg\_interest\_rate,

ROUND(AVG(dti\*100),2) AS avg\_dti

FROM loan\_data

GROUP BY year, month ORDER BY month;

Output:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Year | Month | Total Loans | Fully Paid Loans | Charged-Off Loans | Repayment Rate (%) | Default Rate (%) | Avg. Interest Rate | Avg. DTI |
| 2021 | 1 | 2 | 2 | 0 | 100 | 0 | 12.34 | 3.83 |
| 2021 | 2 | 4 | 3 | 1 | 75 | 25 | 11.44 | 5.6 |
| 2021 | 5 | 1 | 0 | 1 | 0 | 100 | 15.96 | 20.88 |
| 2021 | … | … | … | … | … | … | … | … |
| 2021 | 12 | 1 | 1 | 0 | 100 | 0 | 10.65 | 13.92 |

Observation:

* Repayment in 2021 averaged 79.67%, peaked at 87.63% in Sep, while defaults stayed below 18%.
* Interest rates (10.34% - 12.38%) & DTI (12% - 14%) remain fairly consistent throughout the year.
* Loan volume significantly increases from July onwards (253) to November (20,851).

Quarterly performance trends:

MySQL Techniques: Aggregation with grouping.

Query: similar to monthly query only for quarter segment.

SELECT

YEAR(issue\_date) AS year,

QUARTER(issue\_date) AS quarter,

COUNT(\*) AS total\_loans,

SUM(CASE WHEN loan\_status = 'Fully Paid' THEN 1 ELSE 0 END) AS fully\_paid\_loans,

SUM(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END) AS charged\_off\_loans,

ROUND((SUM(CASE WHEN loan\_status = 'Fully Paid' THEN 1 ELSE 0 END) / COUNT(\*)) \* 100, 2) AS repayment\_rate,

ROUND((SUM(CASE WHEN loan\_status = 'Charged Off' THEN 1 ELSE 0 END) / COUNT(\*)) \* 100, 2) AS default\_rate,

ROUND(AVG(int\_rate\*100),2) AS avg\_interest\_rate,

ROUND(AVG(dti\*100),2) AS avg\_dti

FROM loan\_data

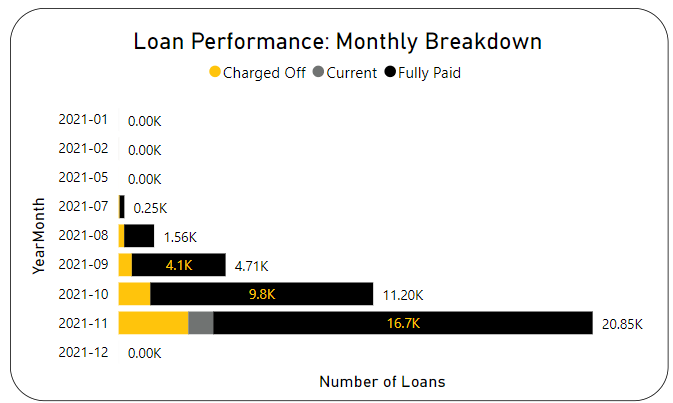
GROUP BY YEAR(issue\_date), QUARTER(issue\_date)

ORDER BY year, quarter;

Output:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Year | Quarter | Total Loans | Fully Paid Loans | Charged-Off Loans | Repayment Rate (%) | Default Rate (%) | Avg. Interest Rate | Avg. DTI |
| 2021 | 1 | 6 | 5 | 1 | 83.33 | 16.67 | 11.74 | 5.01 |
| 2021 | 2 | 1 | 0 | 1 | 0 | 100 | 15.96 | 20.88 |
| 2021 | 3 | 6520 | 5647 | 873 | 86.61 | 13.39 | 11.87 | 12.21 |
| 2021 | 4 | 32049 | 26493 | 4458 | 82.66 | 13.91 | 12.09 | 13.56 |

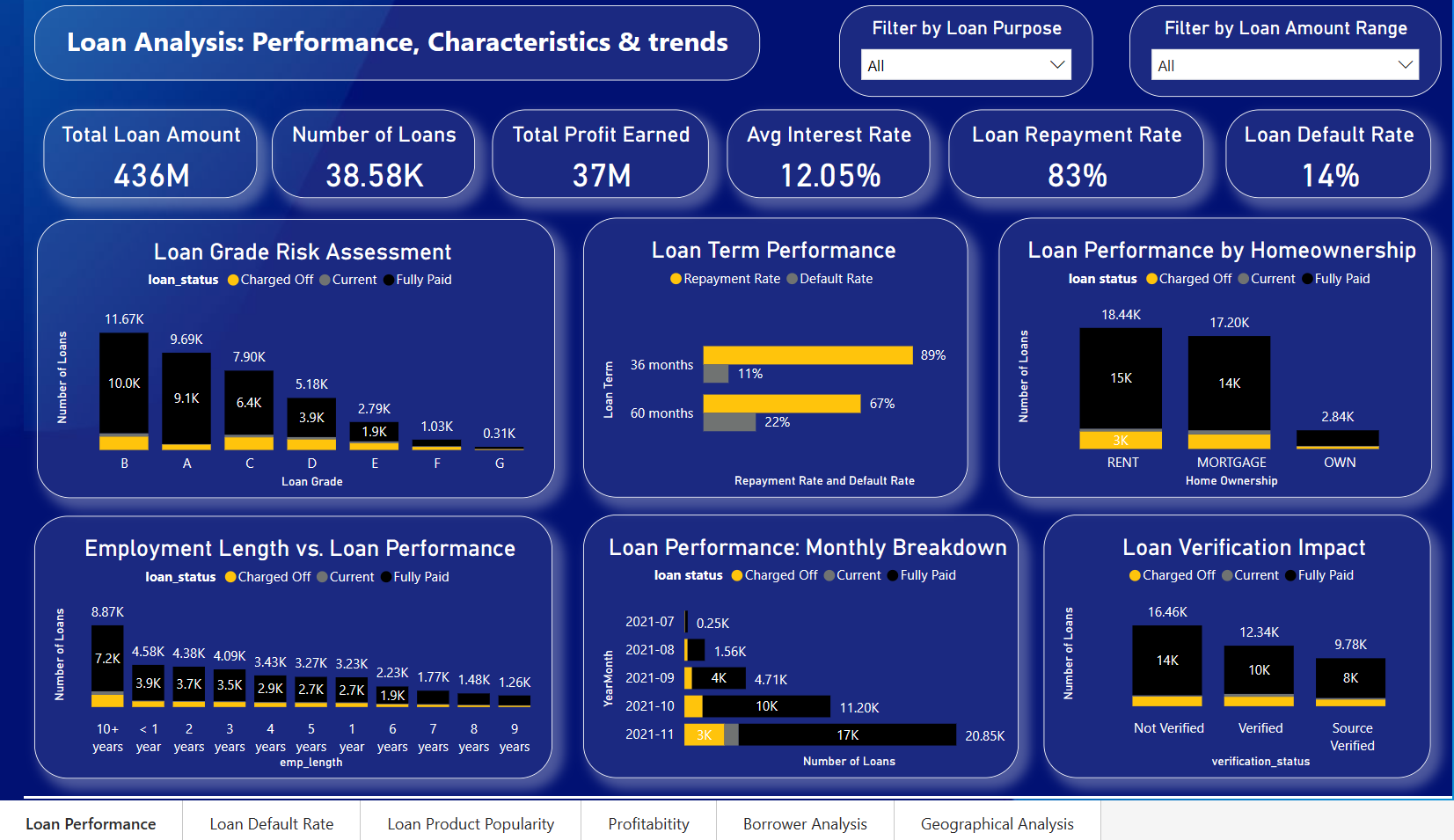
Power BI Visual:

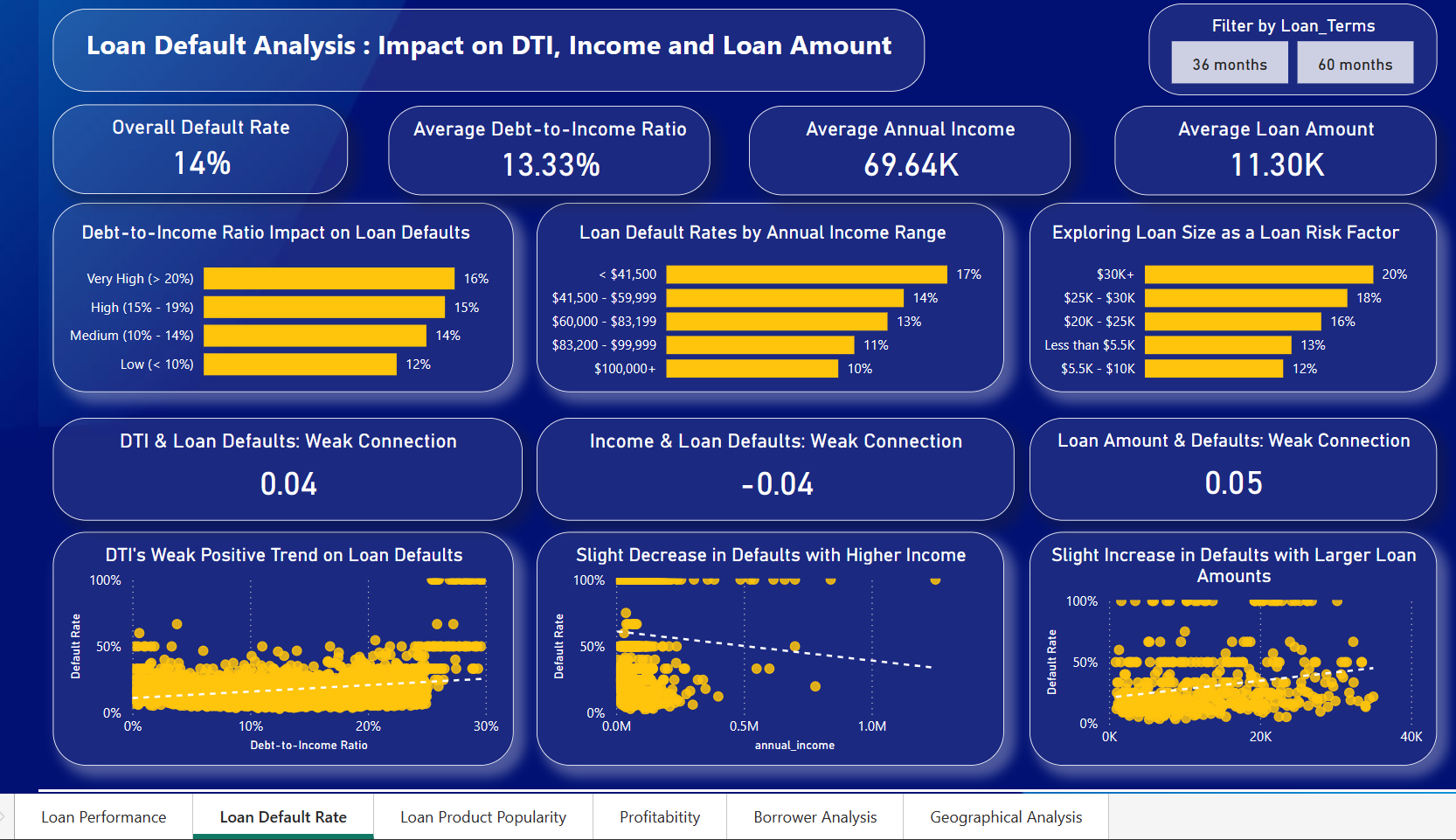


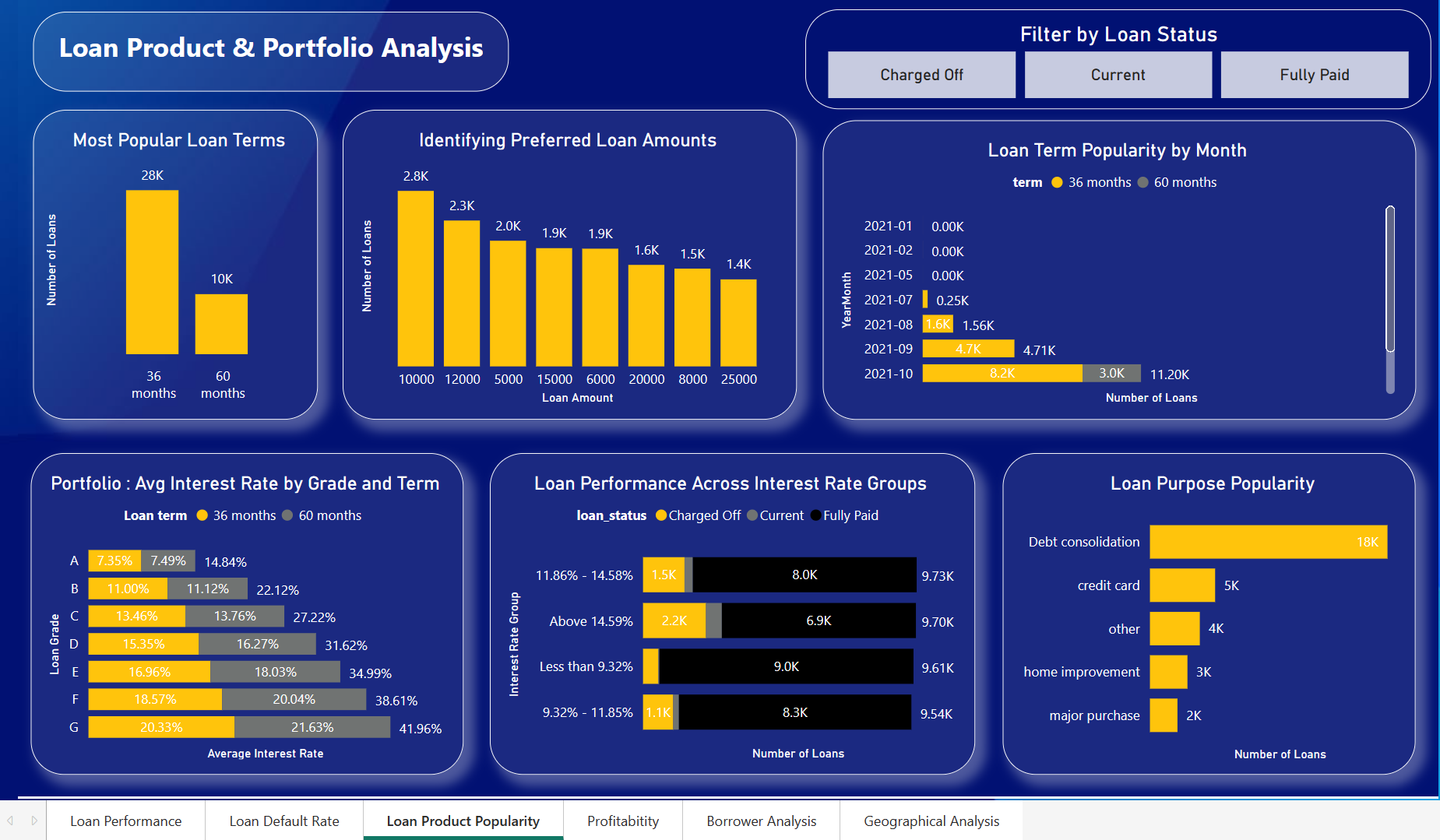
Observation:

* Loan volume peaks in Q4 (32,049).
* Loan repayment rates range from 82.66% to 86.61%, with the highest in Q3 2021.
* Interest rates steady (11.7% - 12.1%), DTI varies (high: 13.56 in Q4, low: 5.01 in Q1).

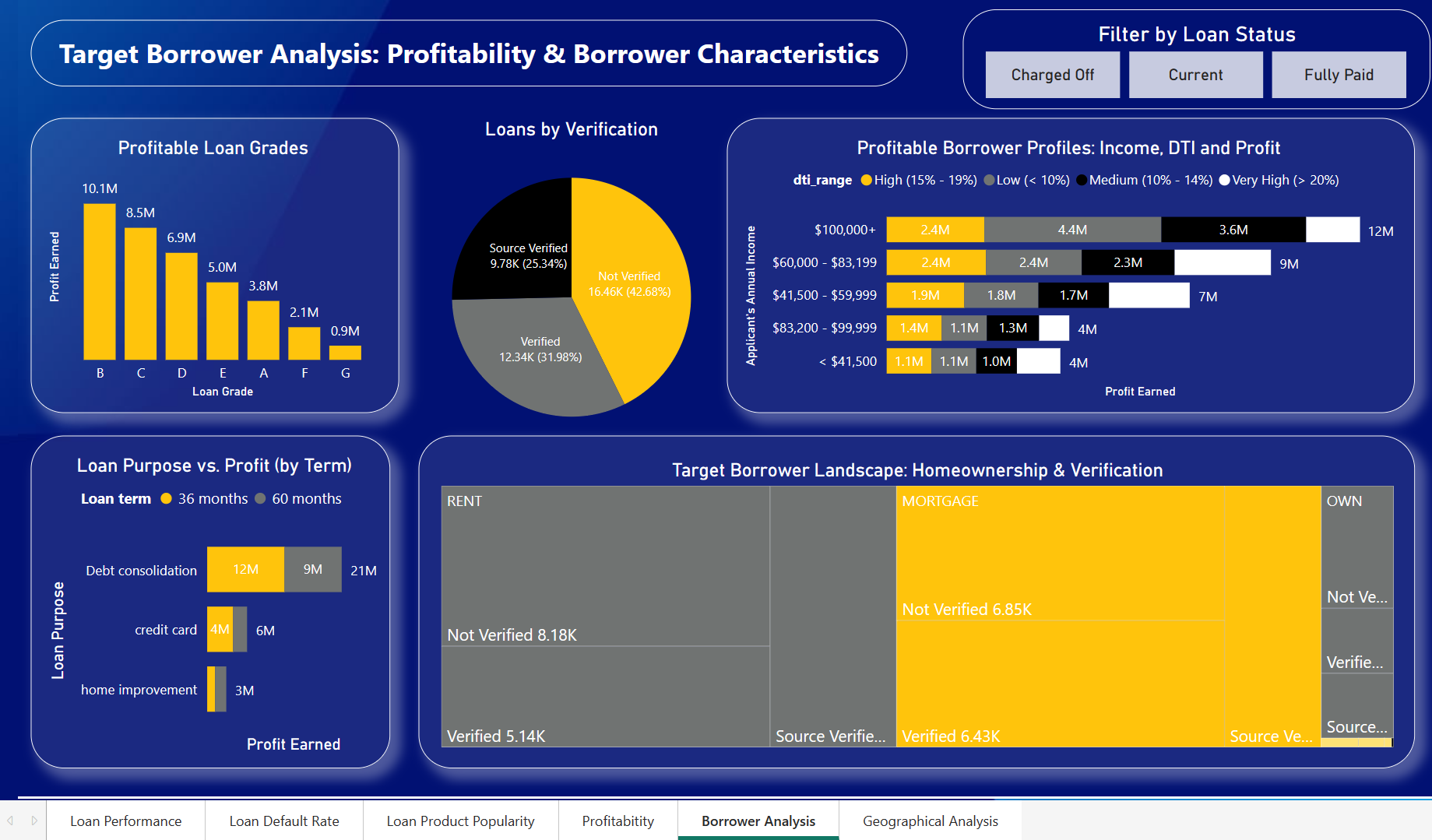
**POWER BI REPORT OVERVIEW**

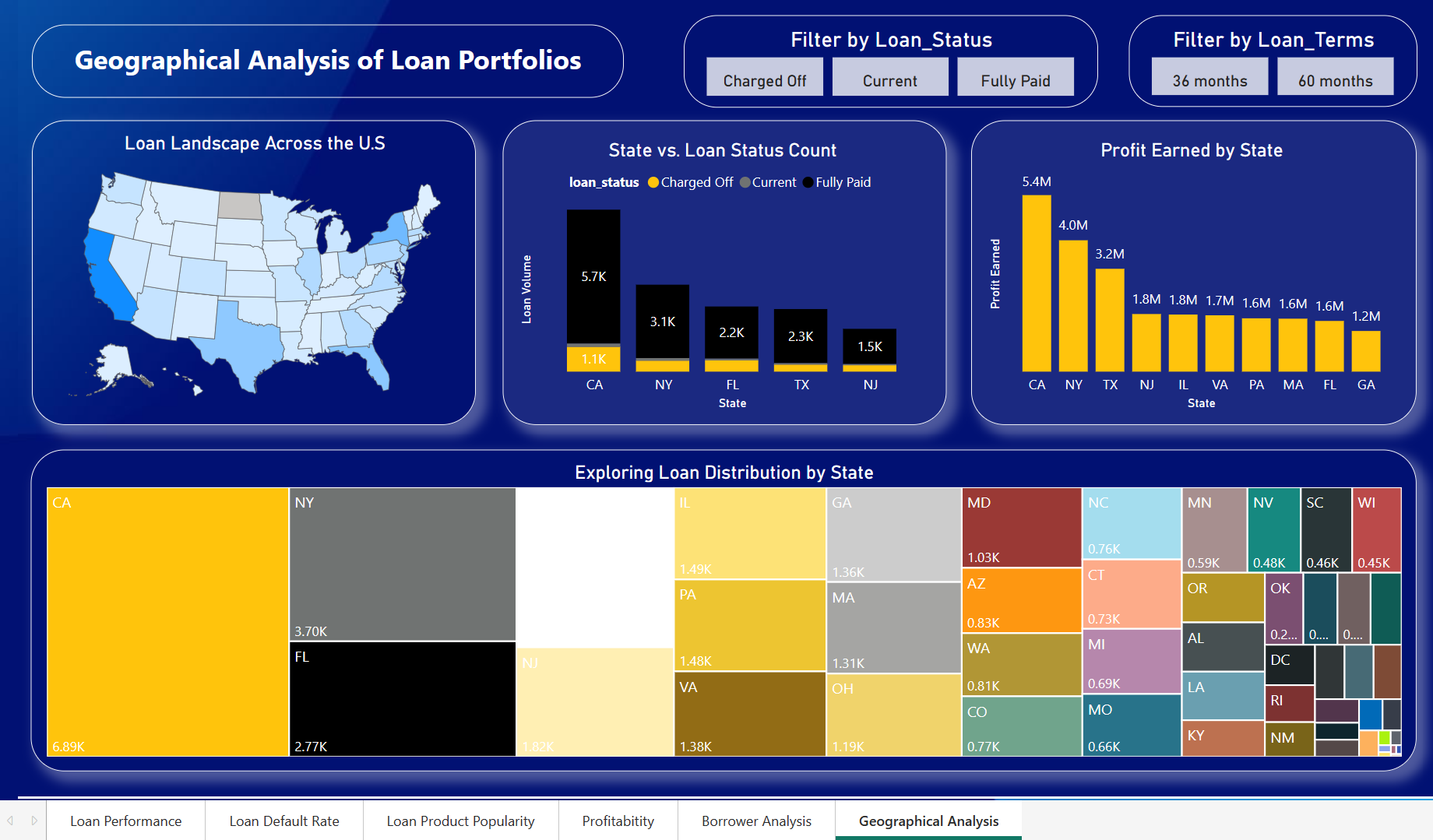
****

****

****

****

****

****

**CONCLUSION**

Loan analysis identified key factors for profitability and risk:

* Loan performance: Higher loan grades (A-C) & shorter terms (36 months) are best for repayment. DTI and loan amount correlate with defaults. Income may be less important than DTI. Investigate why non-verified borrowers default less.
* Loan products: Borrowers prefer smaller ($10k-$12k), faster-to-pay (36-months) loans. Debt consolidation loans with shorter terms are profitable and less risky.
* Target borrowers: Focus on good credit history, home ownership, and low DTI.

Profitability can be improved by:

* Approving loans to borrowers with strong credit history, long work history, and home ownership (regardless of income verification).
* Prioritizing debt consolidation, credit card refinance, and home improvement loans with shorter terms.
* Considering geographical trends when making loan decisions.
* Analysing data over multiple years to understand seasonal patterns.

**APPENDICES**

**Dax Formulas used in Power BI**

**Loan Performance Metrics:**

1. Calculated Measure with Variable.

Repayment Rate =

    VAR FullyPaidCount = CALCULATE(COUNTROWS(loan\_data), loan\_data[loan\_status] = "Fully Paid")

    RETURN

    DIVIDE(FullyPaidCount, COUNTROWS(loan\_data))

1. Calculated Measure with Variable.

Default Rate =

    VAR DefaultedCount = CALCULATE(COUNTROWS(loan\_data), loan\_data[loan\_status] = "Charged Off")

    RETURN

    DIVIDE(DefaultedCount, COUNTROWS(loan\_data))

1. Measure: Row-level calculation using SUMX iteration.

Profit Earned = SUMX(loan\_data, loan\_data[total\_payment] - loan\_data[loan\_amount])

1. Calculated Measure using SUMX iteration.

Profit\_Pct =

VAR TotalProfit = SUMX(loan\_data, loan\_data[total\_payment] - loan\_data[loan\_amount])

VAR TotalLoanAmount = SUM(loan\_data[loan\_amount])

RETURN

DIVIDE(TotalProfit, TotalLoanAmount)

**Correlation Coefficients:**

Corr\_AnnualIncome\_DefaultRate =

VAR \_avgX = CALCULATE(AVERAGE(loan\_data[annual\_income]))

VAR \_avgY = [Default Rate]

VAR \_numerator = SUMX(loan\_data, (loan\_data[annual\_income] - \_avgX) \* (IF(loan\_data[loan\_status] = "Charged Off", 1, 0) - \_avgY))

VAR \_denominator = SQRT(SUMX(loan\_data, (loan\_data[annual\_income] - \_avgX)^2) \* SUMX(loan\_data, (IF(loan\_data[loan\_status] = "Charged Off", 1, 0) - \_avgY)^2))

RETURN

DIVIDE(\_numerator, \_denominator)

Corr\_dti\_DefaultRate =

VAR \_avgX = CALCULATE(AVERAGE(loan\_data[dti]))

VAR \_avgY = [Default Rate]

VAR \_numerator = SUMX(loan\_data, (loan\_data[dti] - \_avgX) \* (IF(loan\_data[loan\_status] = "Charged Off", 1, 0) - \_avgY))

VAR \_denominator = SQRT(SUMX(loan\_data, (loan\_data[dti] - \_avgX)^2) \* SUMX(loan\_data, (IF(loan\_data[loan\_status] = "Charged Off", 1, 0) - \_avgY)^2))

RETURN

DIVIDE(\_numerator, \_denominator)

Corr\_LoanAmount\_DefaultRate =

VAR \_avgX = CALCULATE(AVERAGE(loan\_data[loan\_amount]))

VAR \_avgY = [Default Rate]

VAR \_numerator = SUMX(loan\_data, (loan\_data[loan\_amount] - \_avgX) \* (IF(loan\_data[loan\_status] = "Charged Off", 1, 0) - \_avgY))

VAR \_denominator = SQRT(SUMX(loan\_data, (loan\_data[loan\_amount] - \_avgX)^2) \* SUMX(loan\_data, (IF(loan\_data[loan\_status] = "Charged Off", 1, 0) - \_avgY)^2))

RETURN

DIVIDE(\_numerator, \_denominator)

**Loan Rate Buckets:**

1. Calculated Column for binning using SWITCH statement.

Interest\_Rate\_Group =

SWITCH(

    TRUE(),

    COUNTROWS(FILTER(loan\_data, loan\_data[int\_rate] < 0.0932)), "Less than 9.32%",

    COUNTROWS(FILTER(loan\_data, loan\_data[int\_rate] < 0.1186 && loan\_data[int\_rate] >= 0.0932)), "9.32% - 11.85%",

    COUNTROWS(FILTER(loan\_data, loan\_data[int\_rate] < 0.1459 && loan\_data[int\_rate] >= 0.1186)), "11.86% - 14.58%",

    COUNTROWS(FILTER(loan\_data, loan\_data[int\_rate] >= 0.1459)), "Above 14.59%"

)

**Borrower Characteristics:**

1. Calculated Column for binning using SWITCH statement.

Annual Income Bracket =

SWITCH(

    TRUE(),

    loan\_data[annual\_income] < 41500, "< $41,500",

    loan\_data[annual\_income] < 60000, "$41,500 - $59,999",

    loan\_data[annual\_income] < 83200, "$60,000 - $83,199",

    loan\_data[annual\_income] < 100000, "$83,200 - $99,999",

    "$100,000+"

)

1. Calculated Column for binning using SWITCH statement.

dti\_range =

SWITCH(

    TRUE(),

    loan\_data[dti] < 0.1, "Low (< 10%)",

    loan\_data[dti] < 0.15, "Medium (10% - 14%)",

    loan\_data[dti] < 0.2, "High (15% - 19%)",

    "Very High (> 20%)"

)

**Power BI Report Deployment:**

**1. Published Report to Power BI Service:**

* **Created Workspace:** In the Power BI service, navigate to "Workspaces" and click "New workspace." Fill out the form and create a new workspace.
* **Publish Report:** In Power BI Desktop, open loan report and click "Publish." Select the newly created workspace and publish the report.

**2. Created Personal Gateway:**

* **Download Gateway:** In the Power BI service, click the download icon (arrow pointing down) and select "Data Gateway." Choose "Download Personal Mode" and install it in system.
* **Sign-in and Verify:** Run the downloaded gateway installer, sign in with your account, and leave it running. You can verify the connection in the Power BI service under "Dataset settings" and "Gateway connections."

**3. Shared the Report with Users:**

* **Copy Embed Link:** In the Power BI service, navigate to your report and click "File" > Embed Report -> Website or Portal.
* Copy the generated link containing reportId and autoAuth=true parameters. This link allows secure access for the client.

**4. Created App in Power BI Service:**

* **Create App:** In the workspace, click "Create app" and fill out the form.
* **Configure App:** Select the loan report, choose the group to share it with, and click "Publish app." This creates a dedicated app for the report within the workspace.