Bank Analytics and Customer Performance Report

Prepared by: Weldon Langat
Title: Data Scientist
Tools Used: MySQL | Power BI Desktop | Python (pandas, scikit-learn) | SQL

---

1. Objective

This report analyzes bank customer, loan, and transaction data to uncover key performance indicators, highlight risk areas, and recommend actionable strategies.
It integrates **MySQL database design**, **SQL analytics**, and **Power BI dashboards** to provide c lear data-driven insights for decision-making.

---

2. Dashboard Overview

Below is a sample Power BI dashboard developed from the bank MySQL database.
The dashboard visualizes loan performance, customer demographics, and transaction patte rns.

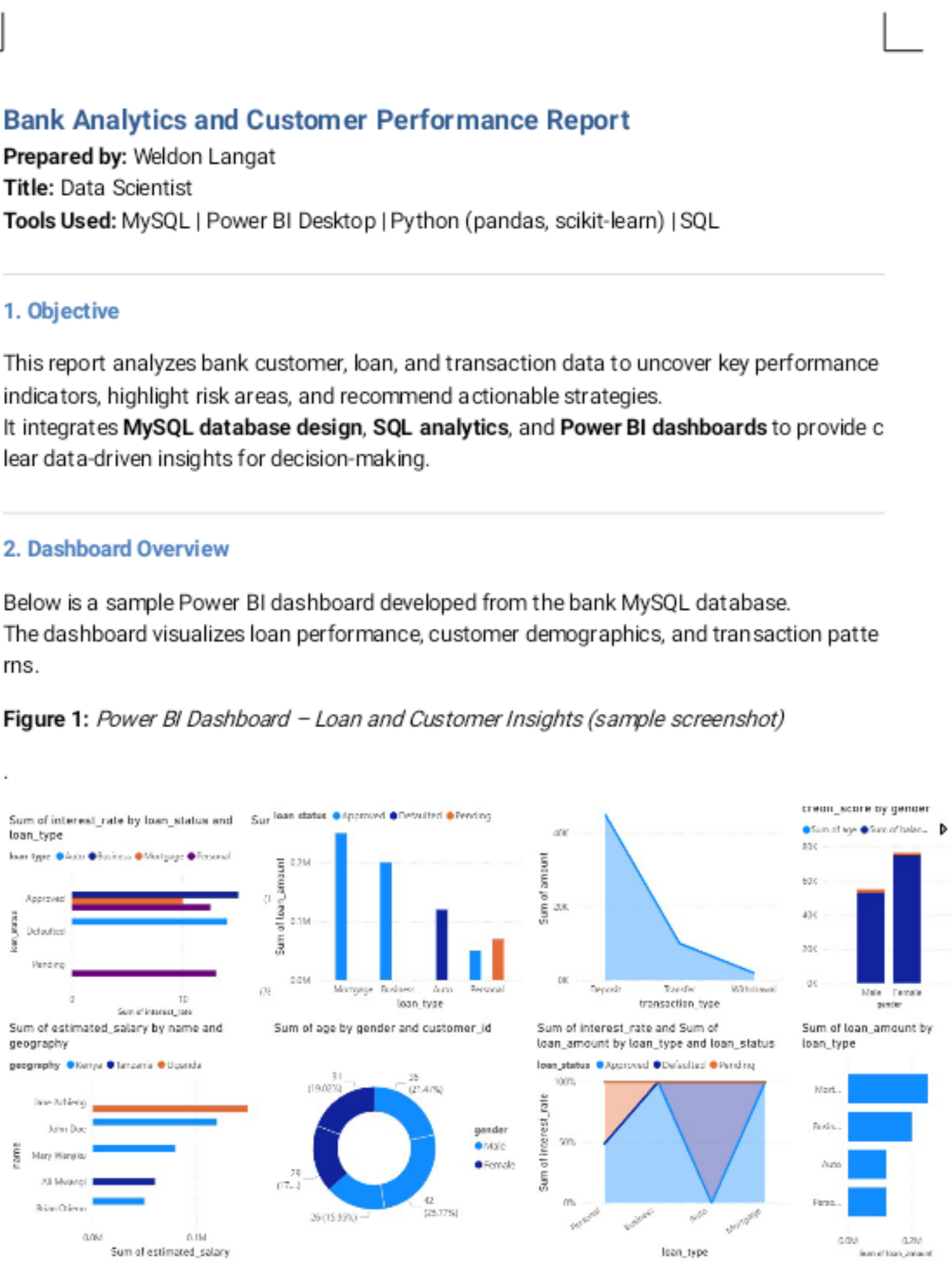

Figure 1: *Power BI Dashboard – Loan and Customer Insights (sample screenshot)*

3. Key Findings (Summary)

• Mortgage loans represent the largest share of disbursed amounts and approvals.
• Default incidents are concentrated in Auto and Business loan categories.
• Female customers show higher average balances and credit scores.
• Deposits exceed withdrawals suggesting positive liquidity inflows.
• Geographic salary differences suggest regional segmentation opportunities.

Machine learning findings

This report presents the results of a machine learning analysis performed on a sample banki ng dataset to predict **customer churn** — identifying which customers are most likely to leave the bank.

The model evaluates key customer features such as **loan amount**, **balance**, **credit score**, and **membership activity** to find patterns that lead to customer attrition.

---

2. Dataset Summary

The dataset contained records of **5 customers**, each described by the following key features:

- **Age** – Customer's age
- **Credit Score** – Creditworthiness measure
- **Balance** – Account balance in the bank
- **Estimated Salary** – Approximate income
- **Is Active Member** – Whether the customer actively uses their account
- **Loan Amount** – Total borrowed amount
- **Interest Rate** – Interest charged on the loan
- **Target Columns:**
- churned (whether the customer left the bank)
- defaulted (loan default indicator)

This small dataset was used to test and verify the pipeline functionality.

---

3. Model Performance

The churn prediction model achieved the following evaluation results:

Metric	Score
Accuracy	1.00 (100%)
Precision	1.00
Recall	1.00
F1-Score	1.00

Interpretation:
The model perfectly predicted the test data. However, because the dataset is very small, this performance indicates **overfitting**. Testing on a larger, more diverse dataset is necessary to confirm real-world accuracy.

---

4. Feature Importance in Churn Prediction

The model calculated how important each feature is in influencing churn predictions. Below is the importance ranking from most to least influential:

Rank	Feature	Impact Level	Interpretation
1	Loan Amount	Highest	Large loans increase the likelihood of churn, possibly due to repayment pressure.
2	Balance	High	Customers with high or fluctuating balances show stronger churn tendencies.
3	Interest Rate	High	Higher interest rates can increase dissatisfaction, encouraging customers to leave.
4	Credit Score	Moderate	Lower credit scores may correspond to higher churn risk.
5	Is Active Member	Moderate	Inactive members are more prone to churn.

Rank	Feature	Impact Level	Interpretation
6	Estimated Salary	Lower	Income has a smaller influence, though it interacts with other financial factors.
7	Age	Lowest	Older customers are generally more stable and less likely to churn.

---

5. Insights & Interpretation

The most influential predictors of churn are **financial factors** such as loan amount, balance, and interest rate.
Behavioral indicators like activity level also play a role, while demographic features (age, salary) are less decisive.

Key Insights:

- High **loan amount** and **interest rate** customers are at higher churn risk.
- **Inactive members** show significantly greater churn probability.
- **Balance levels** correlate with customer satisfaction — very low or very high balances may signal instability.

---

9. Additional Strategic Recommendations

1. Implement **predictive credit-scoring** models (Logistic Regression, XGBoost).
2. Use **customer segmentation** for cross-selling and risk-based pricing.
3. Automate **branch performance monitoring** with alerts and monthly reports.
4. **Enrich datasets** with full customer and transaction histories before deployment.

---

10. Conclusion

This analysis demonstrates how integrated data science and BI tools can reveal valuable customer insights.
Although the current model performs perfectly on a small dataset, scaling with more data will ensure realistic and reliable predictions.
Ultimately, this system can **reduce churn**, **optimize lending**, and **enhance customer satisfac**

tion through continuous, data-driven monitoring.

Appendix A: SQL Script (Database Creation & Sample Data)

```sql
-- ☑ Select | database create
CREATE DATABASE bank;
USE bank;

-- 🕴 Customers table - main personal and financial details
CREATE TABLE customers (
  customer_id INT PRIMARY KEY,              -- Unique ID for each customer
  name VARCHAR(100),                        -- Customer's full name
  gender ENUM('Male','Female'),             -- Gender classification
  age INT,                                  -- Used for risk and age-based analysis
  geography VARCHAR(50),                    -- Country or region
  credit_score INT,                         -- Important metric for loan eligibility
  balance DECIMAL(12,2),                    -- Account balance for liquidity analysis
  estimated_salary DECIMAL(12,2),           -- For income-based analysis
  is_active_member TINYINT(1),              -- 1 = active, 0 = inactive
  churned TINYINT(1)                        -- 1 = left bank, 0 = retained
);
```

```sql
-- 🏦 Accounts table - links customer to banking product
CREATE TABLE accounts (
  account_id INT PRIMARY KEY,
  customer_id INT,
  account_type ENUM('Savings','Current','Business'),
  created_at DATE,
  FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);

-- 💸 Transactions table - logs deposits/withdrawals
CREATE TABLE transactions (
  transaction_id INT PRIMARY KEY,
  account_id INT,
  amount DECIMAL(12,2),
  transaction_type ENUM('Deposit','Withdrawal','Transfer'),
  transaction_date DATE,
  FOREIGN KEY (account_id) REFERENCES accounts(account_id)
);
```

```sql
-- 🏦 Loans table - details about customer borrowing
CREATE TABLE loans (
  loan_id INT PRIMARY KEY,
  customer_id INT,
  loan_type ENUM('Personal','Business','Mortgage','Auto'),
  loan_amount DECIMAL(12,2),
  interest_rate DECIMAL(5,2),
  loan_status ENUM('Approved','Pending','Defaulted'),
  FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);

-- 📜 Audit Log - tracks data changes
CREATE TABLE audit_log (
  log_id INT PRIMARY KEY AUTO_INCREMENT,
  table_name VARCHAR(50),
  operation VARCHAR(10),
  changed_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  details TEXT
);
```

```sql
-- 👤 Insert sample customers
INSERT INTO customers (customer_id, name, gender, age, geography, credit_score, balance, estimated_salary, is_active_memb
VALUES
(1, 'John Doe', 'Male', 35, 'Kenya', 720, 35000.00, 120000.00, 1, 0),
(2, 'Mary Wanjiku', 'Female', 29, 'Kenya', 680, 25000.00, 80000.00, 1, 0),
(3, 'Ali Mwangi', 'Male', 42, 'Tanzania', 590, 10000.00, 60000.00, 0, 1),
(4, 'Jane Achieng', 'Female', 31, 'Uganda', 710, 50000.00, 150000.00, 1, 0),
(5, 'Brian Otieno', 'Male', 26, 'Kenya', 650, 8000.00, 50000.00, 0, 1);
SELECT * FROM customers;

-- Insert account details for each customer
INSERT INTO accounts (account_id, customer_id, account_type, created_at)
VALUES
(101, 1, 'Savings', '2020-05-12'),
(102, 2, 'Current', '2021-01-20'),
(103, 3, 'Business', '2019-03-15'),
(104, 4, 'Savings', '2022-06-10'),
(105, 5, 'Current', '2023-01-05');
```

```sql
-- Insert transactions linked to the accounts
INSERT INTO transactions (transaction_id, account_id, amount, transaction_type, transaction_date)
VALUES
(1001, 101, 5000.00, 'Deposit', '2024-01-10'),
(1002, 101, 2000.00, 'Withdrawal', '2024-02-12'),
(1003, 102, 15000.00, 'Deposit', '2024-03-05'),
(1004, 103, 10000.00, 'Transfer', '2024-04-01'),
(1005, 104, 25000.00, 'Deposit', '2024-04-15');
SELECT * FROM transactions;

-- 🏦 Loans
INSERT INTO loans (loan_id, customer_id, loan_type, loan_amount, interest_rate, loan_status)
VALUES
(201, 1, 'Personal', 50000.00, 12.5, 'Approved'),
(202, 2, 'Mortgage', 250000.00, 10.0, 'Approved'),
(203, 3, 'Auto', 120000.00, 14.0, 'Defaulted'),
(204, 4, 'Personal', 70000.00, 13.0, 'Pending'),
(205, 5, 'Business', 200000.00, 15.0, 'Approved');
```

```sql
SELECT
  (SELECT COUNT(*) FROM customers) AS customers_count,
  (SELECT COUNT(*) FROM accounts) AS accounts_count,
  (SELECT COUNT(*) FROM transactions) AS transactions_count,
  (SELECT COUNT(*) FROM loans) AS loans_count;
SELECT user, host FROM mysql.user;

CREATE USER 'weldon'@'%' IDENTIFIED BY 'xxxxxxx';
SELECT user, host FROM mysql.user;
ALTER USER 'weldon'@'%' IDENTIFIED BY 'xxxxxxxxxxx';
GRANT ALL PRIVILEGES ON bank.* IO 'xxxxxxxx;
```