



Credit Scoring Engine for Loan Disbursement

Arif Mohammad Asfe

ID: 1804068

Mohammad Akbar Bin Shah

ID: 1804075

Ahammed Zayed Uddin Rahat

ID: 1804088

Rajarshi Sen

ID: 1804092

Kowshik Chowdhury

ID: 1804119

Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
Chattogram-4349, Chattogram

October, 2023

Table of Contents

1. Introduction.....	3
2. Background Story.....	3
2.1. Upay Technology Departments.....	4
2.2. Upay Technological Ecosystem.....	7
3. Objectives.....	8
4. Project Overview.....	8
4.1. Project Requirements.....	9
5. Implementation.....	9
5.1. Data Generation.....	9
5.1.1. Types of Customers.....	10
5.2. Key Feature Findings.....	12
5.2.1. Payments and Bills.....	12
5.2.2. Fund Transfer and Add Money.....	12
5.2.3. Send Money.....	12
5.2.4. Cash in and Cash out.....	12
5.2.5. Make Request.....	13
5.2.6. Received Money.....	13
5.2.7. Donation.....	13
5.2.8. Remittance.....	13
5.2.9. Others.....	13
5.3. Data Preprocessing.....	14
5.3.1 Feature Engineering.....	14
5.3.2. Dataframe Merging.....	14
5.3.3. Data Augmentation.....	15
5.3.4. Label Encoding.....	15
5.3.5. Feature Selection.....	16
5.4 Split Dataset.....	16
5.5. Model Description.....	16
5.5.1. Multinomial Logistic Regression.....	16
5.5.2. Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimizer.....	17
5.5.3. Reason to Choose.....	18
5.5.4. Model Workflow.....	18

5.5.4. Model Training Performance.....	19
5.5.5. Confusion Matrix:.....	19
5.6. Integration into User Interface.....	20
5.7. User Interface.....	20
5.7.1. Workflow of User Interface.....	20
5.7.2. Home Page.....	20
5.7.3. Loan Eligibility Test Page.....	21
5.7.4. Result Page.....	22
6. Outcomes.....	23
7. Limitations.....	23
8. Future Improvements.....	24
9. Day Plan.....	24
9.1. Day 01.....	24
9.2. Day 02.....	24
9.3. Day 03.....	25
9.4. Day 04.....	25
9.5. Day 05.....	26
9.6. Day 06.....	26
9.7. Day 07.....	26
9.8. Day 08.....	26
9.9. Day 09.....	27
9.10. Day 10.....	27
10. Conclusion.....	27
Appendix:.....	28

1. Introduction

In the realm of academic and professional development, industrial attachment stands as a pivotal component, offering us a bridge between theoretical classroom knowledge and practical real-world experience. During this phase, we immerse ourselves within an industrial setting, engaging in hands-on activities, and often, real-world projects. As students of the Computer Science and Engineering (CSE) department, such an attachment provides a unique platform for us to acquire practical skills, witness industry practices firsthand, and deepen our understanding of the field.

Our selected organization for the industrial attachment will be Upay, a digital financial service brand of UCB Fintech Company Limited, a subsidiary of the United Commercial Bank, known for its user-friendly, secure, and innovative financial solutions.

Upay offers a valuable learning experience for undergraduate students in a number of ways, including the exposure to a variety of technologies which powers its digital financial services, including mobile banking, cloud computing, and blockchain, real-world problem-solving. Upay is constantly innovating and developing new products and services to meet the demands of its customers. The undergraduate students have the opportunity to receive mentorship from experienced and skilled professionals, who are successfully working in the industry for several years, and have hands-on experience of implementing their theoretical knowledge with their support.

2. Background Story

Bangladeshi people are really good at staying strong when things get tough and never giving up when faced with problems. Drawing inspiration from this positive spirit, Upay was established as a digital financial service platform. It aims to simplify and secure the customer journey while providing a wide range of financial services.

Operated under UCB Fintech Company Limited, a subsidiary of the United Commercial Bank, Upay began its journey in early 2021, with the approval of Bangladesh Bank. Its services consist of mobile transactions, bill payments, in-store and online payments, remittance handling, salary

disbursement and more, accessible through an extensive network of agents and merchants at reasonable rates.

Driven by the 'Digital Bangladesh' vision, Upay attempts to transform the country's financial landscape. It offers continuous and accessible digital financial solutions, encouraging financial inclusion among millions. With empathy as a core value, Upay is committed to prioritizing customer needs, promising to be a dependable problem solver and delivering a secure and innovative customer experience.

2.1. Upay Technology Departments

Upay Technology department comprises different sections, which are outlined in Table 1.

Table 1: Functions of Technology departments of Upay

Section	Functions and Remark
Quality Analysis	<ol style="list-style-type: none"> 1. Quality Assurance 2. Monitors the performance of existing products and services 3. Issue identification and provides feedback to backend and frontend section 4. Develops and implements quality standards and procedures <p>Remark: Checking how well products and services work, finding problems, and helping the teams improve them by making rules and guidelines.</p>
Service Operations	<ol style="list-style-type: none"> 1. Controls the application after handover from development team 2. Handles troubleshooting, customer service, service health 3. 24x7 duty and further support in API if needed

	<p>4. Automatic bug report as this section handle the application after publishing</p> <p>Remark: Taking care of the application once the developers finish their work, ensuring the app runs smoothly all the time, and reporting bugs automatically after launching.</p>
Project Management	<ol style="list-style-type: none"> 1. E2M or Engineer to management handover 2. End to end assigned project monitoring 3. Manages vendor, partner and stakeholders 4. Commercial and technical analysis to integrate the assigned project in supply chain management <p>Remark: Making sure the project moves smoothly from the engineers to management, overseeing the entire project, working with partners and others involved, and checking if the project fits well in the supply chain.</p>
Frontend	<ol style="list-style-type: none"> 1. Web and App(IOS/Android) development 2. Requirement Analysis 3. Backend and deployment support 4. Communicates backend using API and give feedback 5. Forwards application to quality analysis team <p>Remark: Understanding what the app requirements, receiving necessary responses and sending the app for quality testing.</p>
DevOps	<ol style="list-style-type: none"> 1. Uses kubernetes for open source container orchestration system for automating software deployment, scaling, and management

	<ol style="list-style-type: none"> 2. Works on healing a system and app segregation 3. Uses CI/CD pipeline for automation development 4. Employs microservice architecture approach in app development <p>Remark: Use kubernetes for open source containers and CI/CD pipelines in development. Use microservice instead of monolithic approach for efficiency.</p>
Network Observation Center	<ol style="list-style-type: none"> 1. Monitors the performance of the network 2. Identifies and resolves network problems 3. Staffed 24/7/365 to ensure that the network is always being monitored and managed 4. Coordinates with other Sections <p>Remark: Monitor the network's performance, solve problems if detected, physically observe the health of the database using different parameters like temperature, humidity or water flow etc.</p>
Backend	<ol style="list-style-type: none"> 1. Project development and management using best fitted language 2. Tests load and parameter of API 3. Develops a feasible feature from requirements 4. Tests app integration and user acceptance with stakeholders 5. Designs KYC(Know Your Customer) to protect financial system <p>Remark: Develop the project based on feasible requirements, use the best fitted language and integrate app on the internet.</p>

Project Delivery	<ol style="list-style-type: none"> 1. Aligns the project with the overall business strategy 2. Develops a detailed project plan, establish and negotiate deadlines, prioritize projects 3. Monitors assigned project based on deadline 4. Maintains multiple project in parallel if required <p>Remark: Making sure the project fits with our business goals, creating a clear plan and following it, keeping an eye on deadlines, and handling multiple projects at once if needed.</p>
------------------	---

2.2. Upay Technological Ecosystem

Upay's Technological Ecosystem is organized into distinct layers which work together as a cohesive system. The network enables communication, infrastructure hosts and supports the different layers, databases store critical data, applications process data and transactions, and the presentation layer ensures users can interact with the system. These layers are interconnected and rely on each other for seamless operation. To operate efficiently, all layers must maintain stability and reliability.

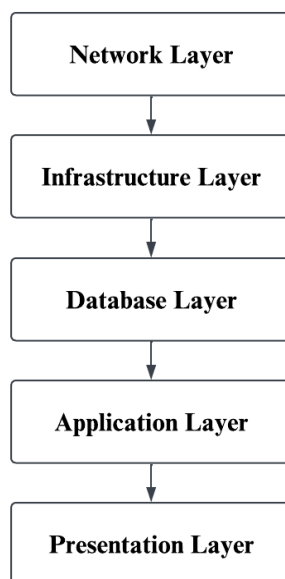


Figure 1: Technological Ecosystem

3. Objectives

- Gain practical experience by actively participating in real-world projects and tasks
- Develop a deeper understanding of industry practices, work culture, and the application of technology and establishing connections
- Complete the assigned project based on loan disbursement named ‘Credit Scoring Engine for Loan Disbursement’

4. Project Overview

As part of our industrial attachment, our assigned project revolves around the development of a credit scoring system aimed at assessing the eligibility of Upay users for loans. This system leverages machine learning techniques to use multiple data files, including transaction history and customer information, to predict the likelihood of a user receiving a loan. The ultimate goal of this project is to deliver an efficient and precise credit scoring system that facilitates loan disbursement decision-making. Additionally, it will offer decision makers some insights into their customer’s loan eligibility and the potential loan amount they could receive. We are also developing a user interface where users can upload data files to receive loan decisions, loan probability assessments, and suggested loan amounts for eligible applicants.

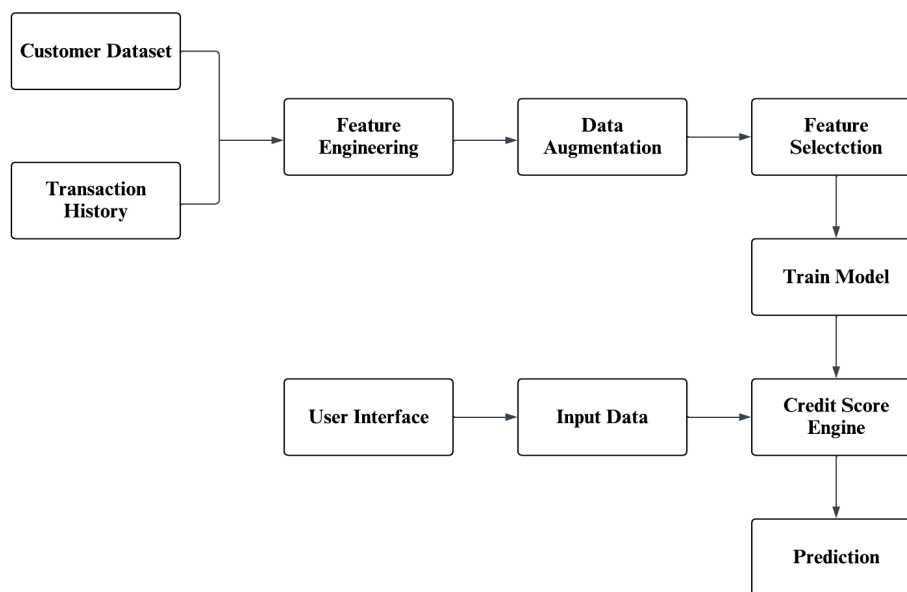


Figure 2: Project Overview

We will take a Machine Learning approach to automate Upay's loan disbursement decision-making process for faster decision making. In Figure 2, we have shown a brief overview of the workflow of building the engine.

4.1. Project Requirements

❖ Front-End

- React
- Tailwind CSS

❖ Back-End

- Django

❖ Machine Learning Algorithm

- Multinomial Logistic Regression

❖ Libraries

- Pandas
- NumPy
- Matplotlib
- Seaborn

❖ IDE

- VS Code
- Jupyter Notebook

5. Implementation

5.1. Data Generation

We have generated our dataset on the basis of different parameters.

5.1.1. Types of Customers

On the basis of different scenarios, we have divided customers into different types. Here, we looked at their balance or quantity as well as how frequently they engaged in transactions. Types are given below:

❖ Type A: Regularly Active User

A small business owner who uses Upay to process payments, pay his suppliers and employees, and receive remittances from his suppliers abroad is an eligible candidate for a Upay loan. He is a frequent user of Upay, and his balance is stable, indicating that his business is financially healthy. While his occupation may vary, he is a good candidate for a loan because he has a demonstrated track record of using Upay responsibly and has a stable financial situation.

❖ Type B: Financially Vulnerable User

In the next scenario, suppose Bob, a single earning parent, is under consideration for a Upay loan. His full-time job does not fully cover his family's expenses, and he relies on financial assistance from relatives and friends. Upay will consider Bob's credit history when making a decision on his loan application.

❖ Type C: Elite Class User

Suppose a customer named Charlie, a business owner who relies on Upay to manage his business finances, is eligible for a Upay loan. He has a high and stable balance history and a positive transaction history, which indicates that his business is financially healthy and that he is a responsible Upay user.

❖ Type D: Middle Class User

In this type, a user, Dan, uses Upay to manage his daily finances, receiving his salary and other income payments through Upay and using it to pay his bills and transfer money to his savings accounts. As such, he is a responsible Upay user and is likely to be eligible for a loan from Upay as long as there is a good transaction history and flow of balance.

❖ **Type E: Low-Income Business User**

This type basically consists of small business owners, vendors, etc. Suppose Jim, a small online business owner with a good transaction history but a lower incoming flow of transactions than outgoing flow, is under consideration for a Upay loan. His credit history suggests that he is a responsible borrower, but his lower income flow may make it more difficult for him to repay a loan.

❖ **Type F: Student User**

Here we considered two types of students: ‘Elliot’ and ‘Elisen’. Elliot has a stable balance and a history of regular cash flow, indicating that he is financially responsible. Elisen, on the other hand, is more dependent on others for financial support and has a history of making requests, cashing out, and paying bills. This suggests that he may be more vulnerable to defaulting on a loan. In this case, Elliot can be considered eligible for a loan.

❖ **Type G: Inactive User**

We have considered four idle customers, W, X, Y, and Z, who use Upay for transactions. W transacts regularly at the beginning and end of the year. Given he has a stable financial situation, he will not be prone to loan default, which makes him under consideration for a loan. X, Y, and Z are not eligible for loans because they only transact regularly at the beginning, middle, or end of the year, respectively, suggesting that their financial situations are less stable.

❖ **Type H: Unsteady Customer**

We have assumed Eric is a freelancer whose behavior on the Upay app fluctuates in the sense that he does not have a consistent balance, that his balance fluctuates, that his app activity fluctuates, and that his wallet fluctuates based on when he receives his payments. As his pattern of transactions is very unstable, he will not be eligible for a loan.

❖ **Type I: Declining Customer**

Suppose a customer named Davies uses Upay to pay for goods and services online. This could include things like shopping, booking travel, or paying for streaming services. If the customer is

spending more money online than they are earning, their Upay balance will decline over time. Although his transactions are regular, his balance has always been declining, which makes him not eligible in case of getting a loan.

5.2. Key Feature Findings

5.2.1. Payments and Bills

If a person has sufficient amounts of transactions under “Make Payment” to pay daily or monthly expenses or “Pay Bill” to pay utility bills, we can consider this person has the ability to pay back the loan timely.

5.2.2. Fund Transfer and Add Money

The ability to repay the loan will be evident if a person can send enough money to his or her online wallet from his or her bank account using "Fund transfer" and enough money from his or her bank account to his or her wallet through "Add Money" and if they carry out these types of transactions with a certain amount of consistency. This is because frequent transactions will raise the person's credit scoring.

5.2.3. Send Money

Sending money means when one person transfers funds to another person's Upay wallet. This is a key factor because it depends on how much money someone has in their wallet. The wallet balance shows a person's financial situation and their ability to pay back a loan if they decide to take one. In simple terms, this one-way transaction is crucial in the engine's assessment of whether someone can get a loan.

5.2.4. Cash in and Cash out

“Cash in” and “Cash Out”, indicating the extent to which an individual deposits or withdraws funds from their Upay wallet, are significant factors. The more an individual engages in regular two-way transactions, the higher their credit score will rise. This increasing credit score will factor into the decision-making process for loan eligibility.

5.2.5. Make Request

Another crucial consideration when making a decision is whether an individual qualifies for a loan or should be taken into account. A person's susceptibility to loan eligibility is considered when they request a sum of money that exceeds their fund limit or transfer activity. The system will classify them based on their individual circumstances in this scenario.

5.2.6. Received Money

“Received Money” refers to transactions in which one person gets money from another user. A user may receive money in response to a request for payment from another user or when that user transfers money to the person in question. It is an important factor since the more money a user receives from another user, the less likely it is that they will default on a loan.

5.2.7. Donation

“Donation” is an important requirement for receiving a loan because when someone donates several times throughout the year and still has money in their wallet to cover the loan in the event of failure, they might be regarded as eligible for a loan.

5.2.8. Remittance

When a person receives remittances multiple times a year, it can be implied that this person has a significant amount of money in his/her wallet due to the influx of money. As a result, remittance can potentially be seen as a key consideration when deciding eligibility for a loan.

5.2.9. Others

There are other government services like Land Tax, DNCC Holding Tax, E-Porcha, E-Mutation that can be paid using the UPAY wallet. These services can be a crucial factor in deciding the eligibility of a loan. If a person uses his/her UPAY wallet regularly to pay the government services, this can play a crucial role in his/her eligibility to receive a loan.

5.3. Data Preprocessing

5.3.1 Feature Engineering

❖ Coefficient of Variation

The coefficient of variation (CV) is a statistical measure of the dispersion of a data set relative to its mean. It is calculated by dividing the standard deviation of the data set by its mean.

$$CV = \frac{\text{Standard Deviation}}{\text{Mean}}$$

The CV is useful for categorizing customers for loan disbursement using transaction history because it can provide a measure of how variable a customer's income or spending is. A customer with a high CV has a more variable income or spending stream than a customer with a low CV.

Customers with a high CV may be more likely to default on a loan because their income or spending may not be sufficient to cover their loan payments on a consistent basis. Therefore, lenders may use the CV to identify customers who are at a higher risk of default.

CV is calculated in this project by dividing the standard deviation and mean of the balances of the customers by their transaction history.

5.3.2. Dataframe Merging

Both customer data and transaction history are merged, and each transaction type is divided into multiple categories based on the amounts of transactions and the limits set by Upay. Each category is grouped by the wallet numbers of the customers, which is used to determine the frequency of transactions for each customer.

Table 1: Categorization Based on Transaction Types

Transaction	Category 1	Category 2	Category 3
Send Money	10-500	501-3000	3001-25000
Received Money	10-500	501-3000	3001-25000

Cash In	10-500	501-3000	3001-30000
Cash Out	50-500	501-3000	3001-25000
Make Payment	1-500	501-3000	3001-300000
Pay Bill	1-2000	2001-300000	
Add Money	50-500	501-3000	3001-50000
Fund Transfer	10-500	501-3000	3001-50000
Request Money	10-25000	-	-
Remittance	1-10000	10001-125000	-
Donation	10-1500	1501-300000	-
Others	1-1500	1501-3000	3001-300000

5.3.3. Data Augmentation

Data augmentation is a technique used to improve the performance of machine learning models by increasing the size and diversity of the dataset by creating new data points from the existing ones.

In this project, we construct a dataset of 13 examples of eight distinct sorts of customers. This dataset is insufficient to build an effective machine learning model. To overcome this issue, we have applied data augmentation to generate a new dataset including 36,000 instances of the same eight customer classes. We have been able to construct a dataset that is significantly larger and more diversified than the initial dataset by employing data augmentation. This allows us to develop a more accurate and generalizable machine learning model.

5.3.4. Label Encoding

As customer address, occupation, and loan approval types are categorical data in this project, one-hot encoding is applied to them to make the parameters appropriate for the machine learning model. One-hot encoding is a label encoding technique used to represent categorical variables as numerical vectors. To one-hot encode a categorical variable, we create a new binary variable for each unique category in the original variable. We then assign a value of 1 to the new binary

variable that corresponds to the category that the original variable is in, and a value of 0 to all other new binary variables.

5.3.5. Feature Selection

Out of all identified features, we chose to pass on to the model all features except wallet number and loan approval. Wallet number is redundant in a customer's transaction history, and loan approval will be the predicted outcome of the machine learning model. This was done to improve model performance and to reduce overfitting risk.

5.4 Split Dataset

This usually involves breaking down a larger dataset into smaller parts. In this project, we divided the dataset into two segments: one for training, constituting 70% of the main dataset, and the other, testing, making up 30% of the main dataset.

5.5. Model Description

5.5.1. Multinomial Logistic Regression

Multinomial logistic regression is a powerful statistical method used for modeling and predicting categorical outcomes with more than two categories.

Multinomial logistic regression is an appropriate choice for the following reasons:

Categorical Outcome: Our dependent variable contains multiple categories, and we need to model the relationships between independent variables and each category's likelihood.

Interpretable Coefficients: Multinomial logistic regression provides interpretable coefficients, allowing us to understand the impact of independent variables on each category relative to a reference category.

Probability Estimates: The model produces probability estimates for each category, which can be valuable for decision-making and classification tasks.

Multinomial logistic regression extends the principles of binary logistic regression to handle multiple categories. It operates as follows:

Model Equation: The probability of an observation belonging to each category is estimated using the softmax function, ensuring that the probabilities sum to 1. This is achieved by modeling a linear combination of independent variables for each category.

$$\log(p) = \ln\left(\frac{p}{1-p}\right) = a + b_1X_1 + b_2X_2 + b_3X_3 + \dots$$

or

$$p = \frac{\exp(a+b_1X_1+b_2X_2+b_3X_3+\dots)}{1+\exp(a+b_1X_1+b_2X_2+b_3X_3+\dots)}$$

- p = the probability that a case is in a particular category,
- \exp = the exponential (approx. 2.72),
- a = the constant of the equation and,
- b = the coefficient of the predictor or independent variables.

The complexity of multinomial logistic regression can vary depending on several factors, including the number of categories, the number of independent variables, and the sample size. Generally, as the number of categories and independent variables increases, the model's complexity also grows.

Interpretability: Interpreting the coefficients for each category can become more challenging as the number of categories and independent variables increases.

Computational Load: Large datasets and a high number of categories or features can increase the computational load and the time required for model estimation.

5.5.2. Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimizer

The **L-BFGS** (Limited-memory Broyden-Fletcher-Goldfarb-Shanno) optimization algorithm is used as a solver in multinomial logistic regression for several reasons:

Efficiency: L-BFGS is computationally efficient and well-suited for problems with a large number of parameters, making it a practical choice for multinomial logistic regression.

Convergence: It is known for its fast convergence properties, which can lead to quicker model estimation.

Memory Efficiency: L-BFGS is memory-efficient, making it suitable for large datasets and models with many parameters.

5.5.3. Reason to Choose

In our project, we employ the multinomial logistic regression model to determine the outcome labels because we are dealing with more than just binary outcomes. Since transaction data is a rich source of both categorical and numerical variables, multinomial logistic regression is very helpful in determining loan eligibility. This method covers multi-class outcomes (for example, loan approval categories), is interpretable, and grows efficiently to handle enormous datasets. Furthermore, it allows for feature selection to increase model accuracy, making it an excellent choice for understanding and predicting loan eligibility based on complex transaction patterns. Our model uses independent variables such as age, address, gender, balance, and transaction history to predict the dependent labels, which represent different categories for loan approval.

5.5.4. Model Workflow

A detailed workflow of the proposed engine is shown in Figure 3

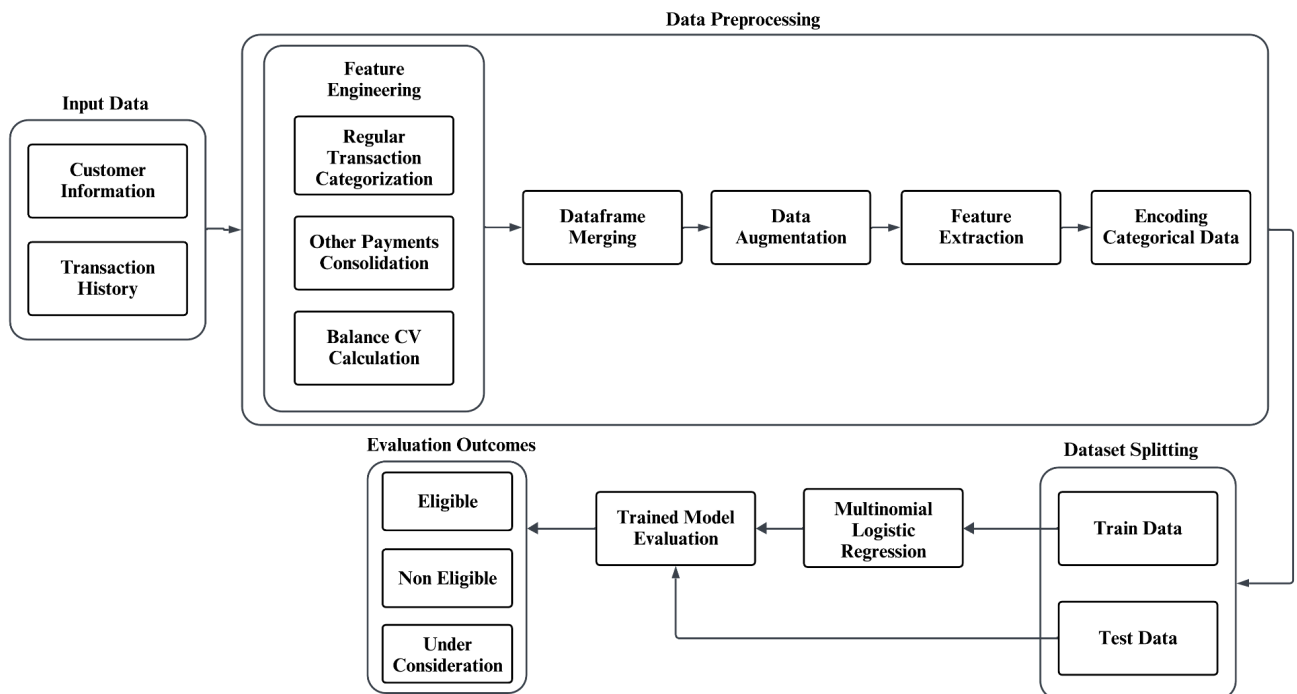


Figure 3: Model Workflow

5.5.4. Model Training Performance

The multinomial linear regression model was trained for 500 epochs on 36,000 instances of customers with selected features and achieved 95% accuracy. To ensure its effectiveness, we evaluated its performance using various metrics.

Table 2: Statistics of The Training Performance of The Model

Output Class	precision	recall	f1-score	accuracy	macro avg	weighted avg	support
Eligible	0.98	0.98	0.98	0.95	0.94	0.95	3576
Non Eligible	0.91	0.94	0.93				1631
Under Consideration	0.92	0.9	0.91				1996
Total Support							7203

5.5.5. Confusion Matrix:

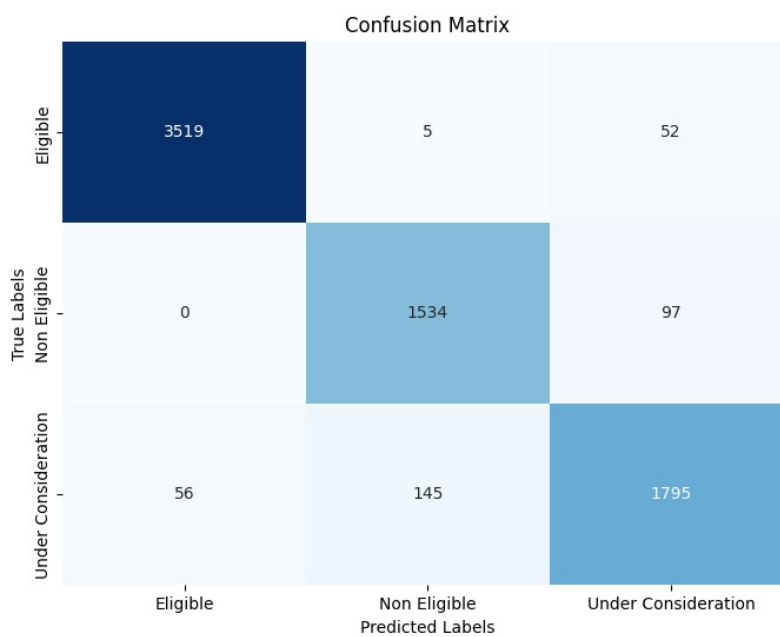


Figure 4: Confusion Matrix

5.6. Integration into User Interface

To deploy the trained model in our user interface, we saved it to a pickle file in binary format. Pickle files are a convenient and efficient way to store Python objects, including machine learning models. By saving the trained model to a pickle file, we can easily load it into our user interface and make it available to the users.

5.7. User Interface

5.7.1. Workflow of User Interface

A detailed workflow of the user interface is shown in Figure 5

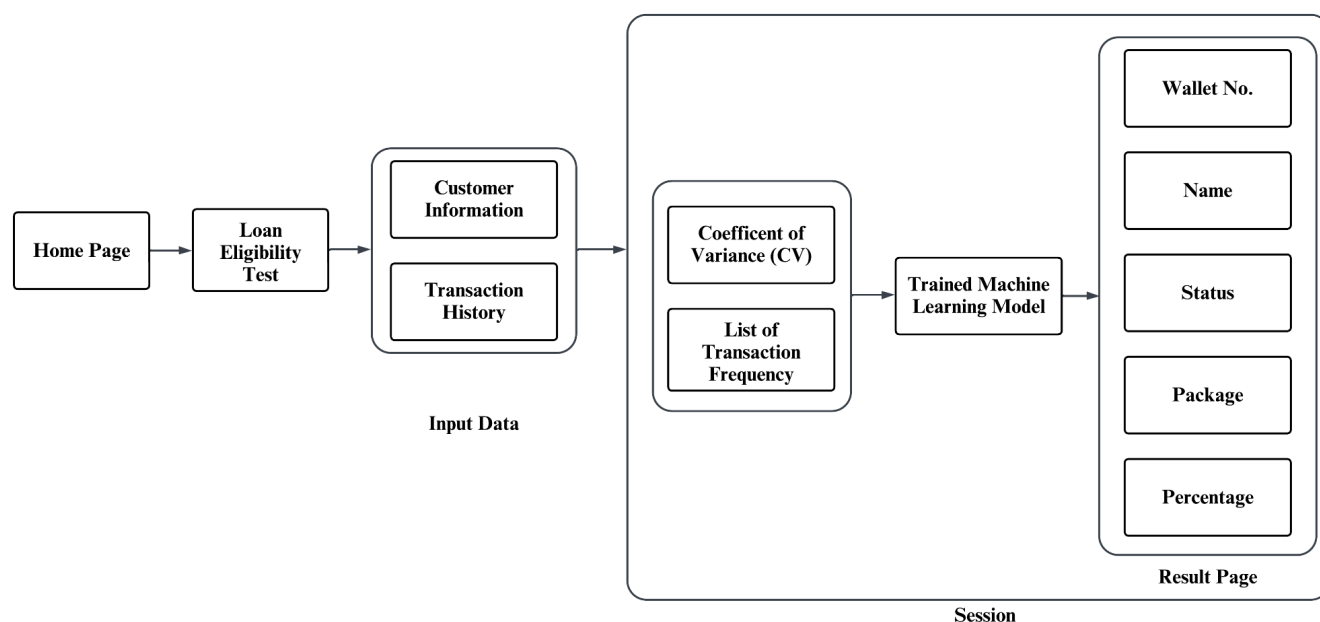


Figure 5: Overview of User Interface

5.7.2. Home Page

This is the project's homepage, where it outlines the project. In the navbar there is another button that will redirect the user to the loan eligibility test page.

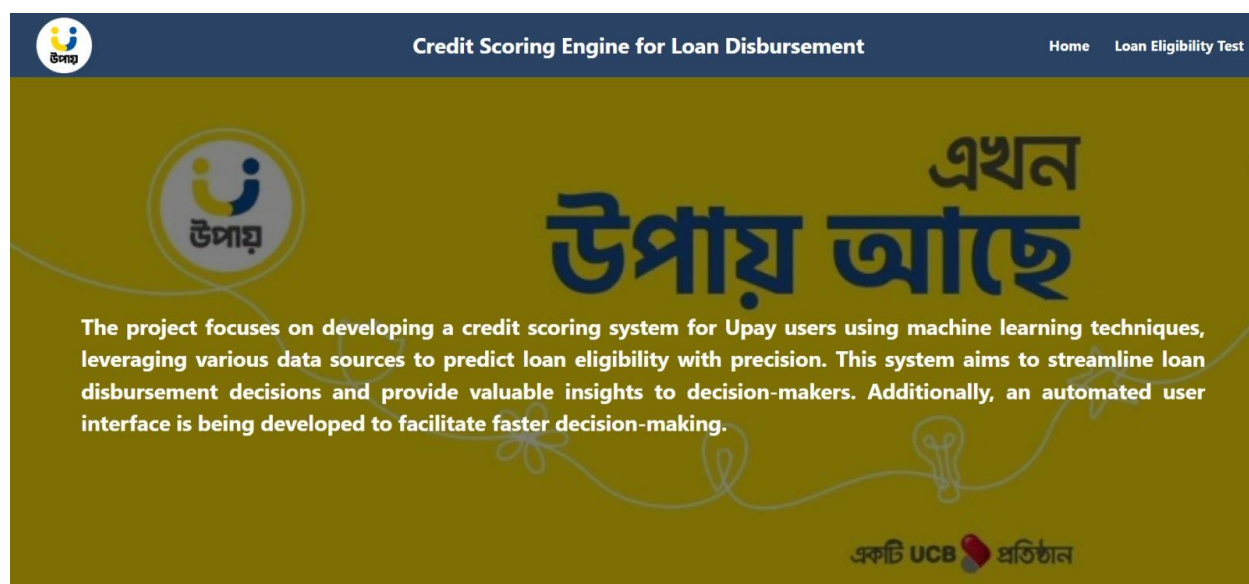


Figure 6: Homepage

5.7.3. Loan Eligibility Test Page

Users can submit customer information and transaction histories as a csv file to this page. The user will receive a success notification after uploading them. After inputting the information, the user will be sent to the loan prediction result page.

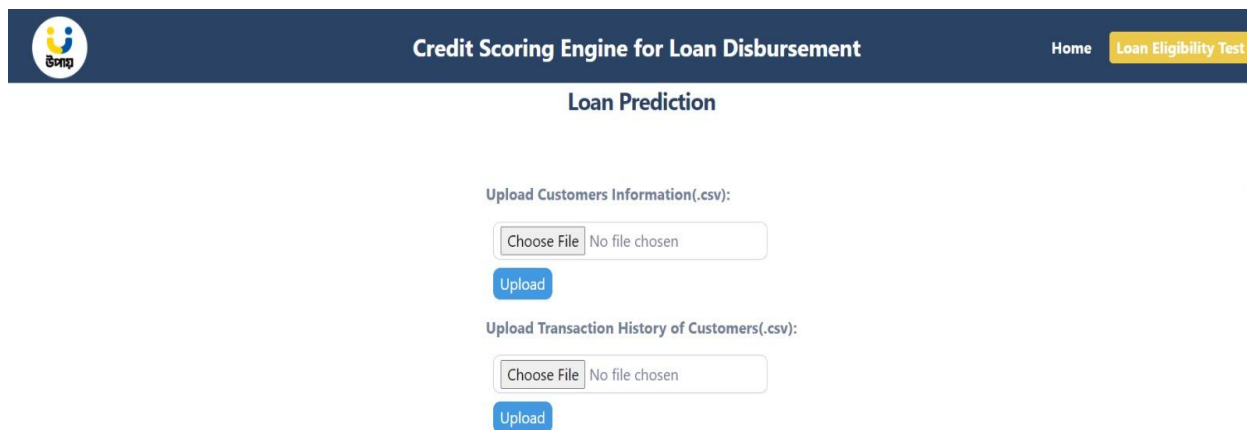



Figure 7: Loan Eligibility Test Page


Credit Scoring Engine for Loan Disbursement
Home Loan Eligibility Test

Success!
 Customer Information File has been uploaded successfully!

Loan Prediction

Upload Customers Information(.csv):

No file chosen

Upload Transaction History of Customers(.csv):

No file chosen


Figure 8: Success Notification of Uploading Information

5.7.4. Result Page

This page shows the results of loan eligibility using customer information and transaction history of customers. This information is fed into the machine learning model, and it shows the wallet number, name, status, available packages, and amount of the loan based on their status, percentages of eligibility, non-eligibility and under consideration.

Table 3: Loan Eligibility Type

Status	Mean Balance(BDT)	Loan Amount (BDT)	Packages
Eligible	0-5000	1000	Type C
Eligible	5001-20000	5000	Type B
Eligible	20001+	10000	Type A
Not Eligible / Under Consideration	-	-	Not Applicable



Credit Scoring Engine for Loan Disbursement

[Home](#) [Loan Eligibility Test](#)

Loan Prediction Results

Wallet No	Name	Status	Packages	Loan Amount	% of Eligibility	% of Non Eligibility	% of Under Consideration
1376318942	Ali Bordhi Khan	Under Consideration	Not Applicable	-	18.97%	7.66%	73.37%
1376318871	Jagat Seth	Eligible	Type B	5000 tk	93.52%	0.02%	6.46%
1974318972	Ghosei Begum	Not Eligible	Not Applicable	-	0.02%	72.84%	27.13%
1876318879	Mir Jafar	Under Consideration	Not Applicable	-	0.02%	0.00%	99.98%
1878318891	Robert Clive	Eligible	Type B	5000 tk	99.64%	0.00%	0.36%

Figure 9: Result Page

6. Outcomes

Our project centers on the development of a credit scoring engine within the Upay app. Its primary objective is to determine the loan eligibility status for users, categorizing them as eligible, non-eligible, or under consideration. In the beginning, we synthesized a comprehensive dataset, combining customer information with transaction history data to train our machine learning model. Utilizing multinomial logistic regression, we applied this dataset to predict loan eligibility labels. Additionally, we designed a user-friendly interface that allows the upload of two files: one containing customer information and the other, the customer's transaction history. This interface generates a table displaying the customer's loan probability, the decision label, and the recommended loan package based on their average balance history.

7. Limitations

- A dataset was not provided for this project.
- In a real-world scenario, the provided features may be insufficient for accurate loan eligibility prediction.
- While our synthesized dataset is almost unbiased, actual real-world datasets often exhibit more biases.
- Transaction type, customer category, loan package type and loan disbursement decisions are fully based on assumptions

- Time constraints have been a limiting factor.

8. Future Improvements

- As this project is fully based on fake dataset, we can work on real time dataset in future to get more realistic results
- To attain the most optimal results, we have the potential to employ more advanced models in our future work.
- Enriching the set of features to broaden the scope of factors considered in loan eligibility assessments.
- Improving the user interface to provide more detailed and user-friendly loan eligibility information
- Developing the model dynamically to adapt user's changing behavior and financial pattern

9. Day Plan

9.1. Day 01

- Gaining familiarized with Upay's Office Culture
- Getting acquainted with Upay's Mobile App and Official Website
- Explored the Background of Upay
- Analyzed the existing features in both mobile application and website
- Classified those features in several categories
- Tested few existing features with real time transaction
- Getting acknowledged with the offers, transaction limitations, transaction history, regional availability, etc.

9.2. Day 02

- Establishing a standardized documentation structure for the project

- Completing critical documentation fields to ensure project clarity and accountability.
- Generating a sample dataset by selecting and filtering key features for analysis
 - “Customer_Info” contains a list of customer names, wallet numbers, addresses, genders, and balances
 - “Transaction_History” contains a record of all transactions for a wallet, and currently all transaction histories of two persons are enlisted.
- Developing a static website using the Django framework for project visibility and access

9.3. Day 03

- Variation in all customers (if possible, 10 to 12 instances)
- Initialization in designing UI using Tailwind CSS
- Synthesized Data creation for customer instances using Python and Bard
- Categorizing dataset and labeling

9.4. Day 04

- Based on transaction amount, transaction types are divided (e.g. Cash in, cash out, send money) into multiple categories (e.g. send money1, send money2, send money3)
- Created a new dataframe where each entry will be of an individual customer and the attributes contain the frequency of his/her transactions in those categories; it ensures linearity of data
- Based on this feature engineering, target variables for each entries have decided
- Studied and analyzed different numerical data augmentation techniques for increasing customer instances belonging to each of the established customer types
- The types of customers in the dataset was updated
- Successfully ran and tested a demo CSV file into designed UI

9.5. Day 05

- Determined the possible frequency ranges of each categories of the transaction types according to the customer types (Type A, Type B etc.) in the dataset
- Further categorized the transactions types on the basis of age, locality and user's occupation
- Augmented the data for all types of customers in the dataset to create the final dataset

9.6. Day 06

- Fixed some data augmentation errors in the dataset
- Dropped irrelevant features from the dataframe
- Attempted to train a “Multinomial Logistic Regression” model using the preprocessed dataset
- Need to solve overfitting issues of that model
- Created primary UI that will take the “Customer Information” and “Transaction History” files of customers which will later be tested into the selected model to get the prediction.

9.7. Day 07

- Finalized the model
- Fixed overfitting issues by avoiding normalization of the data
- Attempted different scenario to test the model's performance
- Created pickle file of the model to integrate into the UI backend, which will be later used to predict outcomes
- In UI, we merged “Customer Information” and “Transaction History” to create an instance for a particular user

9.8. Day 08

- Performing Django label encoding in the UI for transaction history

- Integration of the python program in UI using Django
- Improving the UI

9.9. Day 09

- Finalize the UI design
- Completing the project
- Finishing the documentation of the project
- Testing the trained model using different two csv files containing customer info and transaction history

9.10. Day 10

- Submitting the final documentation
- Presenting the final project

10. Conclusion

To conclude, our project focused on loan disbursement decisions using a credit scoring engine. Despite initial challenges, such as the absence of a dataset and limited knowledge of banking system loan processing, we acquainted ourselves with the Upay app's functionality. This enabled us to generate a synthetic dataset, merging customer information and transaction data. With the dataset in place, we selected a suitable model and successfully predicted loan eligibility.

Moreover, we designed a user-friendly interface that integrated with our trained model, outputting decision outcomes, loan probabilities, and packages based on average balance data. However, it's important to acknowledge that while our model provided results, its real-time accuracy for a realistic customer dataset, could be lessened. Looking ahead, we aim to enhance the model's realism by training it on real-time, more biased customer datasets, making it adaptable to the evolving nature of financial and user behavior patterns.

Appendix:

- **Accuracy:** Accuracy is a metric that measures the overall correctness of a model's predictions. It is calculated as the ratio of the total number of correct predictions (true positives + true negatives) to the total number of predictions.
- **Backend:** The backend refers to the server-side of a software application, where data processing, business logic, and database management occur.
- **Bias in Datasets:** Bias in datasets refers to systematic errors or prejudices present in the data, which can lead to biased machine learning models that favor certain groups or outcomes.
- **CI/CD Pipeline (Continuous Integration/Continuous Deployment):** A CI/CD pipeline is a set of automated processes used in software development to build, test, and deploy code changes to production environments efficiently.
- **Coefficient of Variation (CV):** CV is a statistical measure used to assess the variability of data relative to its mean. It is calculated by dividing the standard deviation by the mean and is useful for understanding the variability of customer income or spending patterns.
- **Computational Load:** Computational load is the amount of computing resources, such as CPU, memory, and processing power, required to perform a specific task or computation.
- **Confusion Matrix:** A confusion matrix is a table used to evaluate the performance of a classification model. It displays the number of true positives, true negatives, false positives, and false negatives.
- **Data Augmentation:** Data augmentation is a technique used to enhance the size and diversity of a dataset by generating new data points from existing ones. It helps improve machine learning model performance.
- **Data Preprocessing:** Data preprocessing involves cleaning, transforming, and organizing raw data into a suitable format for analysis, often including feature engineering.
- **Dataframe:** A data frame is a two-dimensional, tabular data structure commonly used in data analysis and manipulation. It is often used in libraries like Pandas in Python.

- **DevOps:** DevOps is a set of practices that combines software development (Dev) and IT operations (Ops) to automate and streamline the software delivery process, including continuous integration and continuous deployment (CI/CD).
- **F1 Score:** F1 score is the harmonic mean of precision and recall. It is a metric that balances the trade-off between precision and recall and is especially useful when dealing with imbalanced datasets.
- **Frontend:** The frontend refers to the user interface of a software application, including web and mobile app development, where user interactions take place.
- **IDE (Integrated Development Environment):** An IDE is a software application that provides tools and features to facilitate software development. In this case, VS Code and Jupyter Notebook are IDEs.
- **Label Encoding:** Label encoding is a method of converting categorical data into numerical format. Each category is assigned a unique numerical label, making it suitable for machine learning algorithms.
- **Libraries (Pandas, NumPy, Matplotlib, Seaborn):** These are Python libraries used for data manipulation (Pandas and NumPy) and data visualization (Matplotlib and Seaborn).
- **Machine Learning Algorithm:** Machine learning algorithms are mathematical models used to analyze data and make predictions or decisions without explicit programming. In this case, Multinomial Logistic Regression is used for credit scoring.
- **Macro Average:** Macro average is a way of calculating metrics like precision, recall, and F1 score by computing the metric for each class individually and then taking the average.
- **Network Observation Center:** This department is responsible for monitoring and managing the performance of a network, including identifying and resolving network problems to ensure uninterrupted service.
- **One-Hot Encoding:** One-hot encoding is a technique used to represent categorical variables as binary vectors, where each category corresponds to a binary variable. It is commonly used for machine learning with categorical data.
- **Optimizers:** Optimizers are algorithms used in machine learning to adjust the parameters of a model during training in order to minimize the error or loss function and improve the model's performance.

- **Overfitting:** Overfitting occurs when a machine learning model is trained too well on the training data and performs poorly on unseen data. It happens when the model captures noise and fluctuations in the training data.
- **Pickle File:** A pickle file is a binary file format used in Python to store and serialize objects, including machine learning models. It allows for easy storage and retrieval of objects.
- **Precision:** In machine learning, precision is a metric that measures the accuracy of positive predictions. It is calculated as the ratio of true positive predictions to the total number of positive predictions (true positives + false positives).
- **Project Delivery:** Project Delivery involves ensuring that a project aligns with the overall business strategy, creating a detailed project plan, and meeting deadlines.
- **Project Management:** Project Management involves overseeing the planning, execution, and monitoring of a project to ensure it is completed successfully within defined timelines and objectives.
- **Quality Analysis (QA):** QA is the process of ensuring the quality and reliability of software or products. It involves testing, identifying issues, and establishing quality standards to improve the software's performance.
- **Real-Time Data:** Real-time data refers to data that is continuously updated and reflects the current state of a system or process as it happens.
- **Recall:** Recall, also known as sensitivity or true positive rate, is a metric that measures the ability of a model to correctly identify all relevant instances. It is calculated as the ratio of true positives to the total number of actual positives (true positives + false negatives).
- **Service Operations:** Service Operations involve managing and maintaining the functionality and performance of a software application or system after it has been developed and deployed. This includes troubleshooting, customer support, and ensuring the service runs smoothly.
- **Split Dataset:** Splitting a dataset involves dividing it into two or more parts for various purposes, such as training and testing machine learning models. It helps assess model performance.

- **Synthetic Dataset:** A synthetic dataset is an artificially created dataset used for testing and experimentation, typically generated to mimic real data when real data is not available or insufficient.
- **User Interface (UI):** The user interface is the part of a software application or system that users interact with. It includes visual elements and functionality for user interaction.
- **Weighted Average:** Weighted average is a method of averaging where each data point is given a weight based on its importance. It is often used in the context of calculating metrics like precision, recall, and F1 score when dealing with imbalanced datasets.