Embifi's Repayment Score

Introduction:

Credit Scoring is a statistical method which is used to predict the probability that a loan applicant, existing borrower, or counterparty will default or become delinquent. It provides an estimate of the probability of default (delinquency) which is widely used for consumer lending, credit cards and mortgage lending. Credit Scoring is widely used by lenders to decide whether to grant credit or not to borrowers who apply to them.

The goal of any credit scoring process is to be able to summarize all the borrower's available information into a score. If a borrower's score is found to be above a predetermined threshold credit is granted, otherwise it is denied.

Credit Scoring is applied both to new applicants in application scoring and to monitor existing borrowers to determine whether there are changes in their creditworthiness: behavioral scoring.

In both application scoring and behavior scoring, it is important for lending providers to have a large sample of previous customers together with their application details and subsequent credit history. It is from this large sample that the scoring techniques will establish connections between the characteristics(features) of a consumer and how good or bad their credit history is.

The **Embifi's Repayment Score** is a **Behavioral Scoring Machine Learning Model** based on the application details and repayment history of the Customer up to the last six days. It predicts the probability of default of a Borrower and returns a Credit Score between 300 – 850.

Dataset Used:

The Dataset used in this model was obtained from the UCI Machine Learning repository. The data was initially collected for research aimed at the case of customer default payments in Taiwan in October 2005. Among the total 30000 observations, 6636(22.1%) observations were the cardholders with default payments. This research employed a binary variable – default payment (Yes=1, No=0), as the response variable and used 23 variables as explanatory variables. The features have been discriminated into numerical and categorical variables are label encoded, whereby a category is replaced by a numerical value.

The above standard dataset was used for our case because in our case we don't have much data for our customers. As time proceeds we must keep adding

Attribute	Attribute ID	Variable Type	Categories
Amount of Given Credit	X1	Numerical	
Gender	X2	Categorical	1- Male 2-Female
Education	X3	Categorical	1-Graduate School 2-University 3-High School 4-Others
Marital Status	X4	Categorical	1-Married 2-Single 3-Others
Age	X5	Numerical	
Repayment status-Sep 2005	X6	Categorical	-1-Pay Duly 1-Payment Delay of 1 month ⋮ 9-Payment Delay of 9 months
Repayment status in August 2005	X7	Categorical	(Same as above)
Repayment status in July 2005	X8	Categorical	(Same as above)
Repayment status in June 2005	X9	Categorical	(Same as above)
Repayment status in May 2005	X10	Categorical	(Same as above)
Repayment status in April 2005	X11	Categorical	(Same as above)
Bill Statement-Sep 2005	X12	Numerical	
Bill Statement-Aug 2005	X13	Numerical	
Bill Statement-Jul 2005	X14	Numerical	
Bill Statement-Jun 2005	X15	Numerical	
Bill Statement-May2005	X16	Numerical	
Bill Statement-Apr 2005	X17	Numerical	
Amount paid in September 2005	X18	Numerical	
Amount paid in August 2005	X19	Numerical	
Amount paid in July 2005	X20	Numerical	
Amount paid in June 2005	X21	Numerical	
Amount paid in May 2005	X22	Numerical	
Amount paid in April 2005	X23	Numerical	

Table 4.1: Description of Dataset

new data to our training dataset. We shall also use **Data Augmentation** techniques to increase the number of training examples which we have and to create artificial data.

Machine Learning Model and its Implementations:

The Dataset which we have in hand is an imbalanced dataset and hence SMOTE (Synthetic Minority Oversampling Technique) has been employed to reduce misclassifications of the minority class. The Data set has been Mean and Variance Normalized.

Since our objective is to minimize the number of False Negatives (or Type II Error) we might want to make our model a little biased towards predicting someone as a default customer. We may allow some False Positives (Type-I error) but won't allow a larger number of False Negatives since default customers cause us larger loss. Hence class weights have been assigned to {1 : 0.55 , 0 : 0.45} making our model a bit biased (or overcautious).

An Artificial Neural Network with the following specifications has been trained using the oversampled and normalized training dataset:

- Input Layer: 23 units (corresponding to the number of features)
- Hidden Layer 1: 15 units
- Hidden Layer 2: 8 units
- Hidden Layer 3: 6 units
- Hidden Layer 4: 3 units
- Output Layer: 1 unit (Probability of Default)

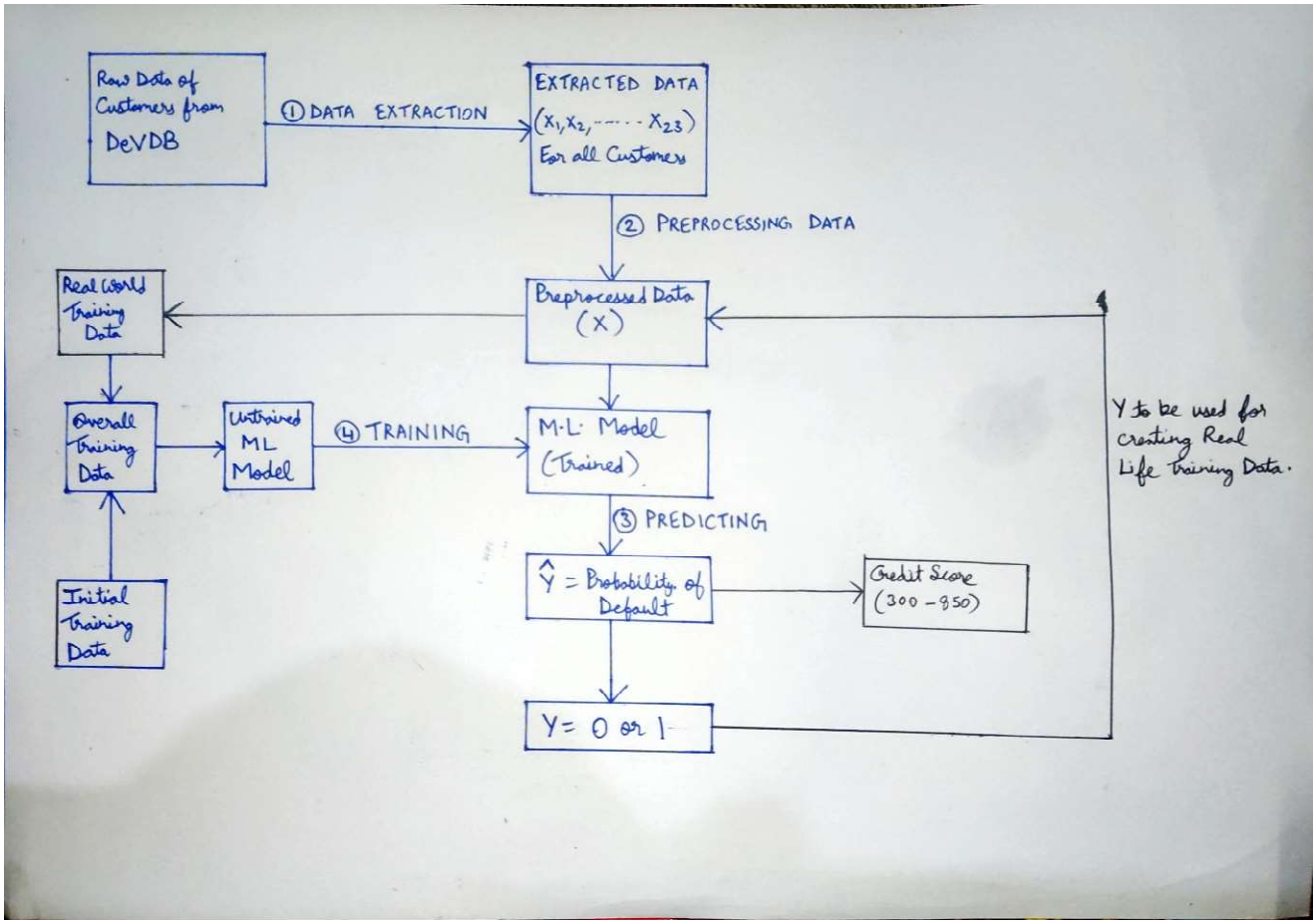
Sigmoid Activation has been used and the model has been trained using **Adam Optimizer.**

Since our aim is to minimize the **Type II error** (Miss), **Recall Value** has been taken as the deciding error metric for our Model. The higher the Recall value, the better the model is.

For the current set of Trained Parameters, we have obtained the following error metrics on the Test set:

Error Metrics	**Training Set**	**Test Set**
Recall	0.7765	0.7351
Precision	0.7101	0.6807
F-1 Score	0.7418	0.7069
Binary Accuracy	0.7298	0.6951
AUROC	0.8124	0.7683

Block Diagram for the whole Model:



End Goal of the Model:

The objective is to create a Model which would provide us with the Credit Score (a number between 300-850) on the basis of Probability of Default of an existing Borrower. The Model should take Customer ID as input. Based on the Customer ID it should be able to extract the Attributes of the Customer from the various databases of Embifi (**Data Extraction**). It should then be able to preprocess that data according to our model (**Data Preprocessing**). When these Attributes are feeded to our trained ML Model it should be able to predict the Credit Score of the Customer. The model should be deployed using AWS Sagemaker.

Tools and Technologies to be used:

Python, MongoDB, PyMongo, TensorFlow, AWS Sagemaker, Flask, ScikitLearn.

TimeLine for various parts of the Model:

1. Data Extraction (26th – 30th December 2022)
2. Data Preprocessing (15th – 18th December 2022)
3. Training the ML Model (13th – 18th December 2022)
4. Deploying the Model on AWS Sagemaker (26th December 2022 – 4th January 2023)