भारतीय प्रबंध संस्थान बेंगलूर
INDIAN INSTITUTE OF MANAGEMENT
BANGALORE

Project Report

MLOps & Systems Design

PGP Term 5
2021

Submitted to:
Prof. Manaranjan Pradhan

Submitted by GROUP 7

Arjun Mehra	2017010
Madhur Butke	2017030
Prafulla Sangle	2017040
Raja Ram Pandey	2017011

Business Context

To become a $5 Trillion economy with a rapid pace, growth of every stratum of society across urban areas and the hinterland region is pertinent for India. This growth cannot manifest without the fulfilment of necessary capital formation. Now since most Indians in the small towns and rural India are not left with large amount of savings after their day-to-day expenditures, access to affordable finance is critical to improve productivity across the agriculture sector and small business units.

The group chose to limit the scope of the problem to credit availablity for purchasing two wheelers, used cars, three wheelers or tractors. For such an application, the small finance companies start the credit underwriting process with a manual evaluation of the application by a credit underwriting officer. Next, the loan application involves a personal interview with the borrower. Upon a satisfactory response from both these media, the loan amount is finally disbursed. However, post the disbursal, to make sure the loan amount has been used in the right purchase as quoted in the application, Post Asset verification agents are sent to the borrowers' home who testify the vehicle purchase and whether the exact model of the vehicle is bought or not.

Since the process involves a lot of manual intervention, it leaves significant scope for biases to kick in, leading to an incorrect perception of the default risk. Further, the physical visits also involve a huge cost in terms of the time invested and the wages that would be needed to be paid to Post Asset Verification agents. Therefore, the group aimed to create an analytics-based underwriting engine which will output a credit risk probability of a given loan application. This probability will account as an additional factor which the human underwriters shall consider, thus making their judgement more robust. Further, the group has also tried to evaluate the credit default risk of the borrowers' post loan disbursal so that the Post Asset Verification agents have a ranked order list of probable defaulters which they can use to create their sequence for house visits. This will help make the verification process faster and cost efficient too.

Dataset Overview

The dataset used for the project contains loan disbursal information and fraud check for 11702 customers. For each of the disbursals, the dataset captures 23 features - location information, financial information of a loan and demographic information of customers. Given below are the features that have been used in our analysis, with description for each of them:

Customer_id	Unique identifier for each customer
Date_disb	Date of disbursal of loan
pay_type	Payment type
area_code	Area code of customer's location
pin_code	Pin code of customer's location
state	State in which the customer is residing
dealer	Type of dealer from whom the vehicle is being bought (DEALER/ASC)
product_code	Product Code (SC: Scooter, MO: Moped, MC: Motorcycle, EB: Electric Bike)
tenure	Standardized tenure of the loan issued
roi	Standardized ate of interest of the loan issues
emi	Standardized monthly installments to be paid by the customer
proc_fee	Standardized processing fees charged for the loan
asset_cost	Cost of the asset to be bought
loan_amt	Amount of loan disbursed
gender	Gender of the loan applicant
qualification	Educational qualification of the loan applicant (HSC/SSC/UG/PG/OTHERS)
employ_type	Employment type (SAL: Salaried, AGR: Agriculture, HOW: Housewife, OTH:

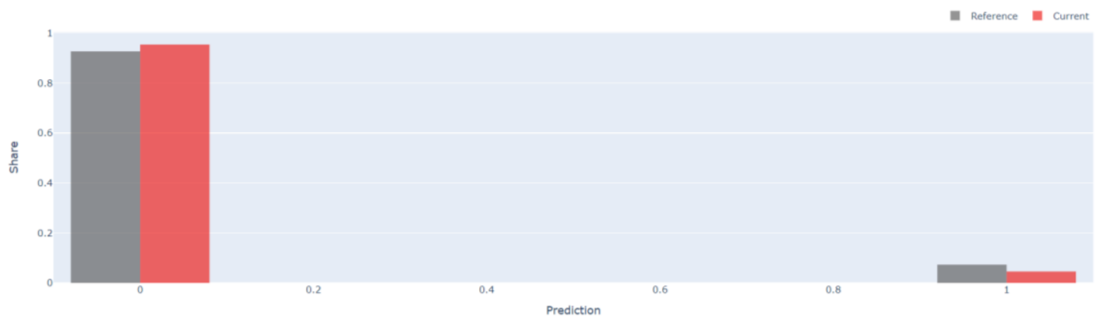
	Others, STU: Student, SEP: Self-employed, PEN: Pension)
resid_type	Residence Type (O: Owned, R: Rented, L: Leased)
age	Standardized age of the loan applicant
cibil_score	Standardized CIBIL score of the loan applicant (left blank if not available)
net_salary	Standardized net salary of the loan applicant
net_irr	Standardized net internal rate of return for the company on the loan
fraud	Target Variable: 1/0 (1: Fraudulent; 0: Not fraudulent)

Data Profiling

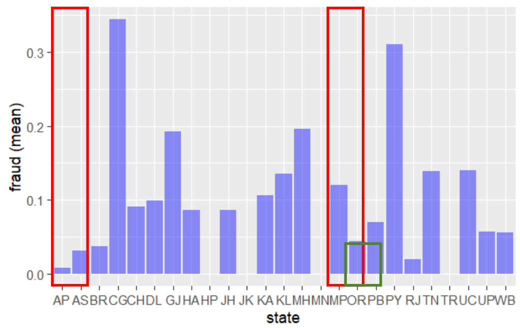
The dataset was randomly divided into a model development dataset (80%) and a production dataset (20%). The datasets were checked for the three types of drifts:

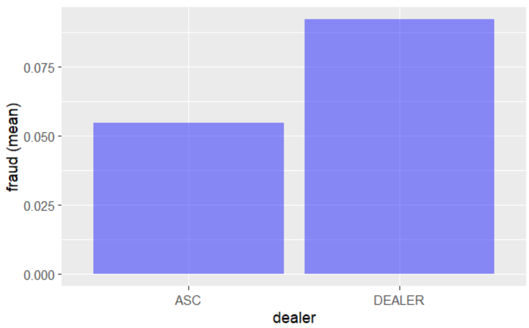
1. Data drift: It was detected for 12 out of 105 features (including encoded categorical variables). The affected variables, in the increasing order of their p-value, were state_JH, area_code_3053.0, area_code_3044.0, state_PB, area_code_3015.0, state_CG, state_TR, area_code_3087.0, area_code_3019.0 and area_code_3045.0.
2. Target drift: No target drift was detected between the two datasets (p-value = 0.31117).
3. Prediction drift: Very significant drift was noticed in the concept between the two datasets.

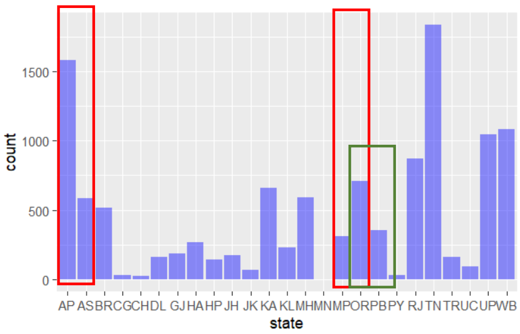
Prediction Drift: detected, p_value=4e-06

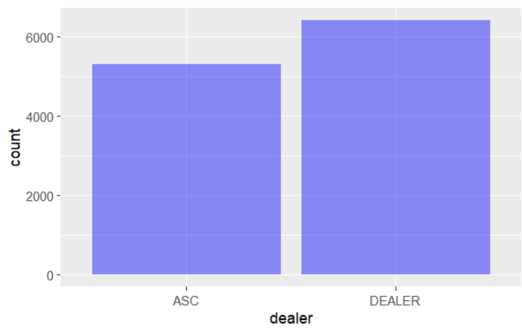


Further looking into the data, it was observed that certain states like Chhattisgarh and Pondicherry showed a remarkably high rate of frauds than other states. This could be attributed to the very low number of loans offered in this state – either causing inefficient default probability estimates or because of some fundamental reasons. However, states like Rajasthan have been performing very well and thus offer an opportunity to learn. Similarly, 'DEALERS' also showed higher fraud rates than ASC.









Model Deployment Strategy

Given the application we are targeting the model to serve, we would need both online and batch prediction deployment. Since the application credit score will have to be outputted whenever an underwriting officer adds the details, we would need to use online prediction for this application. However, for generating the ranked list of customers for Post Asset Verification, batch prediction will be a more suitable deployment strategy.

Model Performance Metrics

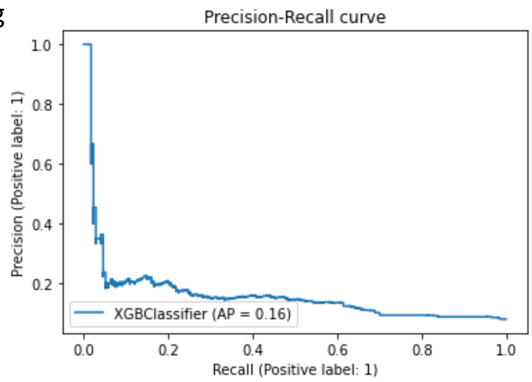
The average amount lent for a vehicle loan has been assumed to be equal to Rs. 70,000. The cost of capital was taken as 6% and the interest rate charged as 10%. For a 3-year tenure, the approximate profit from a successful lending operation comes out to be Rs. 8,400. Also, assuming the average amount lost in a default situation to be 50% of the lent amount, the cost of a wrong prediction was taken as Rs. 35,000. Hence, the final evaluation metric to be maximised is:

$$Expected\ profit\ per\ customer = \frac{8400 \times True\ Negatives - 35000 \times False\ Negatives}{No.of\ customers}$$

The above metric has been chosen over others since it represents a direct alignment with business goal (extending credit such as to maximise profits, making trade-off against customer acquisition) and provides a single number for comparison – a major advantage over choosing precision and recall metrics.

Model Selection

During the model development, first, features were selected using the sequential feature selection, embedded methods and recursive feature elimination approach. Next, we used the AutoML framework to understand on a broader level which models might give us the best results. It suggested us to use decision trees, random forest and XGBoost. To refine our process further, we implemented One-Hot Encoding, Target Encoding for encodings and used Simple Up-Sampling and SMOTE techniques to remove class imbalances. For all the possible combinations, we tracked our experiments using Weights & Biases and chose the model family which had the highest 'Expected profit per customer'. Finally, we used GridSearchCV to tune the hyperparameters of the chosen model.



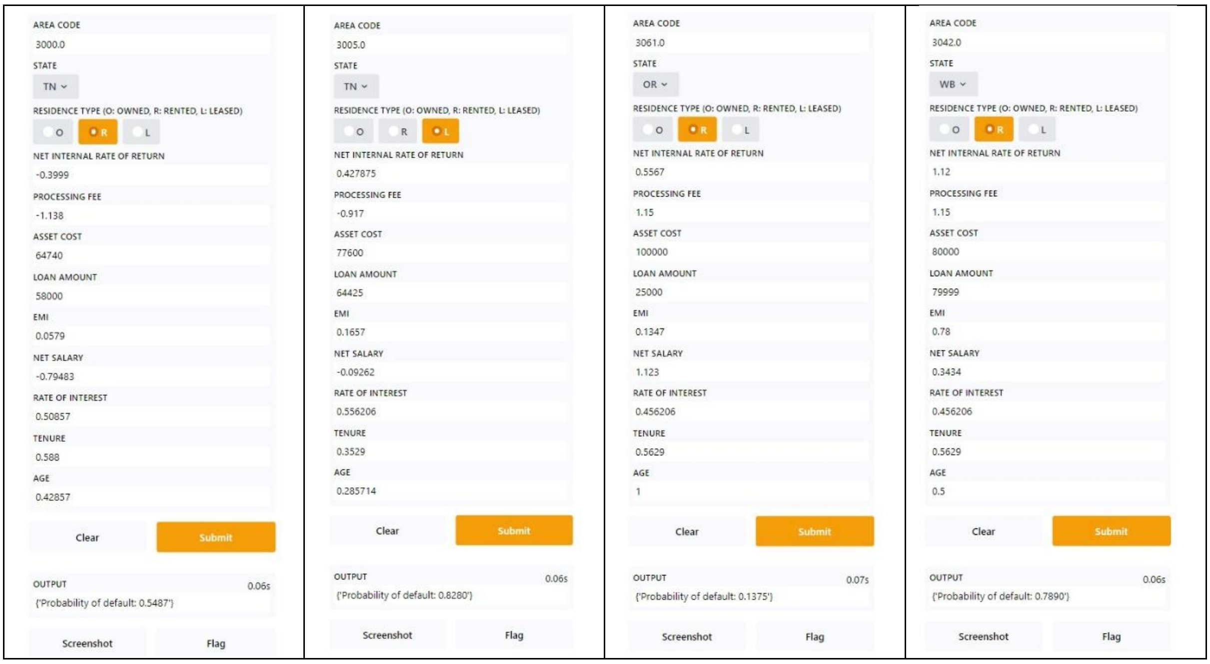
This was the PR curve for the selected model. As the data was highly imbalance and our business objective was to minimize custom defined cost function, we did not got the typically good Area under PR curve.

Runs (143)

Name (143 visualized)	State	User	Tags	Created	Runtime	Sweep	F_Score	accuracy	cost	precision	sensitivity	specificity
XGBoostWithTarget_HyPm3	finished	madhu	Target	60m ago	5s	-	[0.192]	[0.8885]	-121982	[0.2081]	[0.1782]	[0.9455]
XGBoostWithTarget_HyPm2	finished	madhu	Target	1h ago	6s	-	[0.0114]	[0.9256]	-121310	[0.5]	[0.0057]	[0.9995]
XGBoostWithTarget_HyPm3	finished	madhu	Target	1h ago	6s	-	[0.033]	[0.9248]	-121674	[0.375]	[0.0172]	[0.9977]
RandomForestWithTarget_HyPm3	finished	madhu	Rando	1h ago	5s	-	[0.1521]	[0.9047]	-122248	[0.2247]	[0.1149]	[0.9681]
RandomForestWithTarget_HyPm2	finished	madhu	Rando	1h ago	8s	-	[0.1566]	[0.8987]	-121604	[0.2056]	[0.1264]	[0.9608]
RandomForestWithTarget_HyPm1	finished	madhu	Rando	1h ago	7s	-	[0.0714]	[0.9222]	-122234	[0.3182]	[0.0402]	[0.9931]
XGBoostWithOHEwithSMOTE_HyPm3	finished	madhu	OheEn	1h ago	5s	-	[0.192]	[0.8885]	-121982	[0.2081]	[0.1782]	[0.9455]
DecisionTreeWithTarget_HyPm3	finished	madhu	Decisic	1h ago	5s	-	[0.2005]	[0.8705]	-120316	[0.1854]	[0.2184]	[0.9229]
DecisionTreeWithTarget_HyPm2	finished	madhu	Decisic	1h ago	6s	-	[0.1245]	[0.9038]	-121016	[0.1928]	[0.092]	[0.9691]
DecisionTreeWithTarget_HyPm1	finished	madhu	Decisic	1h ago	6s	-	[0.193]	[0.8821]	-121254	[0.1964]	[0.1897]	[0.9377]
XGBoostWithOHEwithSMOTE_HyPm1	finished	madhu	OheEn	1h ago	6s	-	[0.0718]	[0.9226]	-122318	[0.3333]	[0.0402]	[0.9935]
RandomForestWithOHEWithSMOTE_HyPm3	finished	madhu	OheEn	1h ago	5s	-	[0.0619]	[0.9222]	-121968	[0.3]	[0.0345]	[0.9935]
RandomForestWithOHEWithSMOTE_HyPm2	finished	madhu	OheEn	1h ago	7s	-	[0.0985]	[0.9218]	-122948	[0.3448]	[0.0575]	[0.9912]
RandomForestWithOHEWithSMOTE_HyPm1	finished	madhu	OheEn	1h ago	6s	-	[0.0729]	[0.9239]	-122570	[0.3889]	[0.0402]	[0.9949]
XGBoostWithOHE_HyPm3	finished	madhu	OheEn	1h ago	6s	-	[0.051]	[0.9205]	-121366	[0.2273]	[0.0287]	[0.9922]
XGBoostWithOHE_HyPm3	finished	madhu	OheEn	1h ago	6s	-	[0.0515]	[0.9214]	-121534	[0.25]	[0.0287]	[0.9931]
XGBoostWithOHE_HyPm2	finished	madhu	OheEn	1h ago	6s	-	[0.1775]	[0.897]	-123332	[0.2185]	[0.1494]	[0.9571]
XGBoostWithOHE_HyPm1	finished	madhu	OheEn	1h ago	7s	-	[0.1994]	[0.8868]	-122178	[0.2102]	[0.1897]	[0.9428]
RandomForestWithOHE_HyPm3	finished	madhu	OheEn	2h ago	6s	-	[0.1379]	[0.9145]	-123116	[0.2759]	[0.092]	[0.9806]
RandomForestWithOHE_HyPm2	finished	madhu	OheEn	2h ago	7s	-	[0.0622]	[0.9226]	-122052	[0.3158]	[0.0345]	[0.994]
RandomForestWithOHE_HyPm1	finished	madhu	OheEn	2h ago	7s	-	[0.1081]	[0.9154]	-12222(	[0.25]	[0.069]	[0.9834]

Model Consumption

1. **Batch prediction:**
   The group exported the model file in a pickle format to a GitHub repository. This pickle file can be loaded directly to Google Colab or any IPython notebook application. The production data can be uploaded. Here in the notebook named 'batch_prediction_pkl.py', we have used test data in a CSV format from a GitHub repository for batch deployment demonstration.
2. **Online prediction:**
   For online prediction we have used both the *app-based deployment* for model demo purpose and Library based deployment for deploying *model as a service*.
   - App based deployment: The model has been deployed using Gradio app for a UI based access. The screenshots of the typical usage have been pasted below.



Screenshots of the Gradio app predictions

- Online library-based deployment: A library was created on Github for deploying the model as a service. The github link for the library is shown below:

  https://github.com/rajap20/MLOpsProject

  The parameters for the library-based deployment are given below.

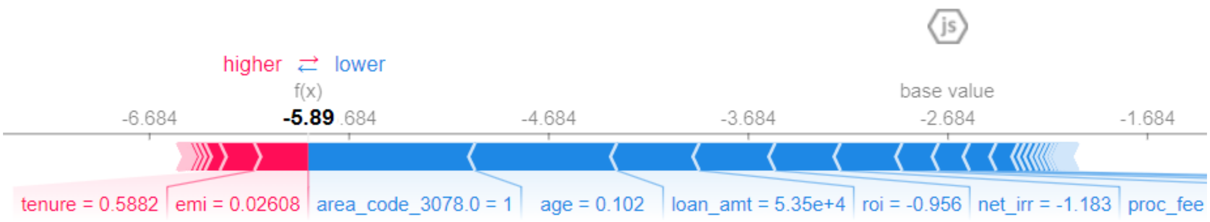
```
Parameters:
        area_code (str): "Area Code"
                 state (str): "State" ("AP", "AS", "BR", "CG", "DL", "UP", "HA", "GJ",
         "HP", "CH", "JH","JK", "KA", "KL", "MH", "MP", "OR", "TN", "PY", "PB", "RJ",
         "WB", "UC", "TR", "MN")
        tenure (float): (default=0.470588      , label="Tenure")
        roi (float): (default=-0.793282, label="Rate of Interest")
        emi (float): (default=0.124252, label="EMI")
        proc_fee (float):(default=-0.093758, label="Processing Fee")
        asset_cost (float): (default=87000.0, label="Asset cost")
        loan_amt (float): (default=71000.0, label="Loan Amount")
         resid_type (str): (["O", "R", "L"], default="O", label =    "Residence Type (O:
         Owned, R: Rented, L: Leased)")
        age (float): (default=0.306122, label="Age")
        net_salary (float): (default=   0.158169, label="Net Salary")
        net_irr (float): (default=-1.321153, label="Net Internal Rate of return")

Returns:
         float: The output shows the probability of default of the candidate
```

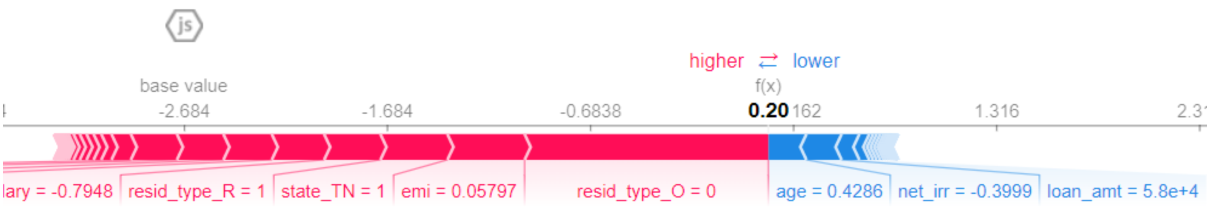
Model Explainability

The group implemented ELI5, Shapley Values, ICE and PDP for global explanation of the model and LIME and Shapley Values for local explanation of the predictions. Sample local explanations from the Shapley Values have been shown in the screenshots below:

1. Explaining a case of non-fraudulent application: The variables *tenure* and *emi* have been increasing the prediction probability of default where *emi* has the most impact. At the same time, *area code, age, loan_amt, roi, net_irr* etc decrease the predicted probability of default in that order.



2. Explaining a case of fraudulent application: In this case, the variables *age, net_irr* and *loan_amt* have been reducing the predicted probability of default. However, other variables in the decreasing order of importance, including *resid_type, emi, state_TN, resid_type_R* etc have been increasing the predicted probability of default.



Key Learnings

1. There were multiple options available for choosing the model selection metric. The group had thought conflicts between sensitivity, specificity, precision and recall, AUC etc. However, we realised that the whole purpose of the modelling activity is to generate business impact, and therefore we created a metric of our own which represented the 'expected profit per customer'. The whole further process was always aligned towards increasing it.
2. During the course of the projects, the group members were working on different parts of the same project. Initially, everyone was working individually and reconciling the work after 5-6 days. This caused a lot of merge conflicts leading to a lot of wastage of time. Therefore, we started reconciling the work as soon as possible – almost everyday to avoid merge conflicts piling up.
3. Initially the group members were having difficulties in completing their allocated tasks due to dependencies with each other's tasks. Hence, we learnt to divide tasks in a modular fashion to allow parallel development.
4. Tools like 'Weights & Biases' were a useful value addition. In all our initial projects, we always found it difficult to keep track of which models and with which hyperparameters were being used for what prediction accuracy. The tool made the tracking part very smooth.
5. Highlight of the project was model monitoring. As we learnt during the course, it is necessary to monitor model decay to avoid multiple types of risk. In our project context it is necessary to observe performance of model in identifying possibly fraudulent loan/credit applications. This requirement was fulfilled through monitoring of data drift, target Drift and concept drift. In our case we found significant data drift in 12 variables. For example, Data drift was found in loan application observations from Jharkhand and Punjab state. Hence, we
   have to focus on collecting latest new loan applicant data from these states.
6. As our project context involves FinTech area, we need to follow compliance standards set by Regulatory authorities like RBI. For our case, we had to explain how our model classifies loan application into genuine and fraudulent applications. As part of this, we tried multiple global explanation methods to understand the overall structure of how a model makes a decision. For example, how various features mentioned in a loan application form are impacting chance of fraud in loan applications submitted by customers. On the other hand, with local interpretation methods, we tried to understand how our model make decision for single instance along with behaviour of model in local neighbourhood. For example, using Shapely values we analysed why our model was classifying 37th record in our test data as a possibly fraudulent loan (credit) application.
7. The project should be run not in a waterfall approach, but with an agile methodology. Also, we should not be focussing on solely accuracy if it becomes a bottleneck in the whole development process. For example, it consumed a lot of efforts from the group to improve the model accuracy just a little bit – halting focus from all other tasks in the pipeline.

Model Card for Vehicle Credit Assessment Model

Model Details

Overview

The model is meant to be used to generate additional credit default probability score to be used along with manual credit application assessment. It is also meant to generate a ranked list to be followed by Post Asset verification agents to optimise their house visit sequences.

Version

name: Version 1.0
date: 2021-12-07

Owners

Group 7, group7@google.com

Considerations

Intended Users

- Credit underwriting officers
- Post Asset Verification agents

Use Cases

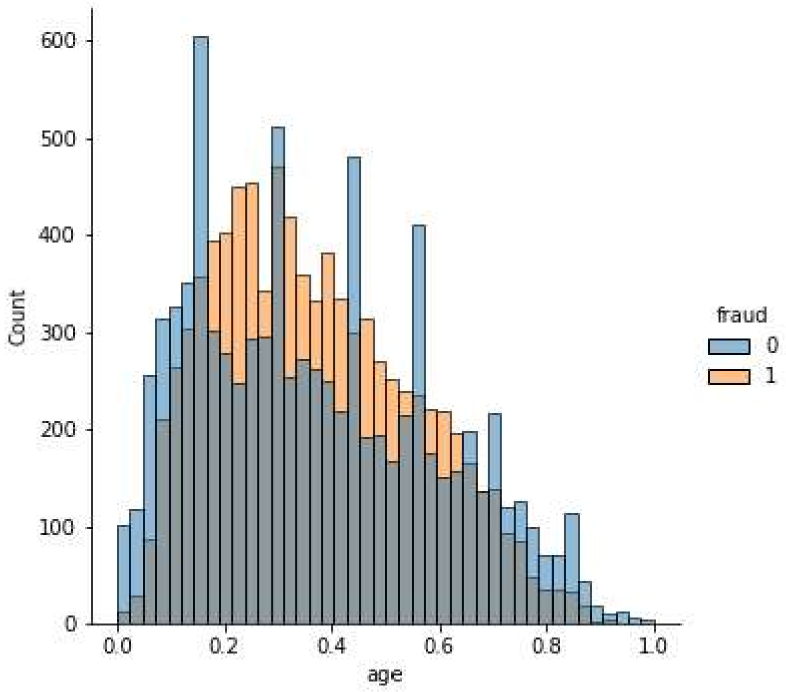
- Credit default probability evaluation; rank order sequencing for Post Asset Verification agents

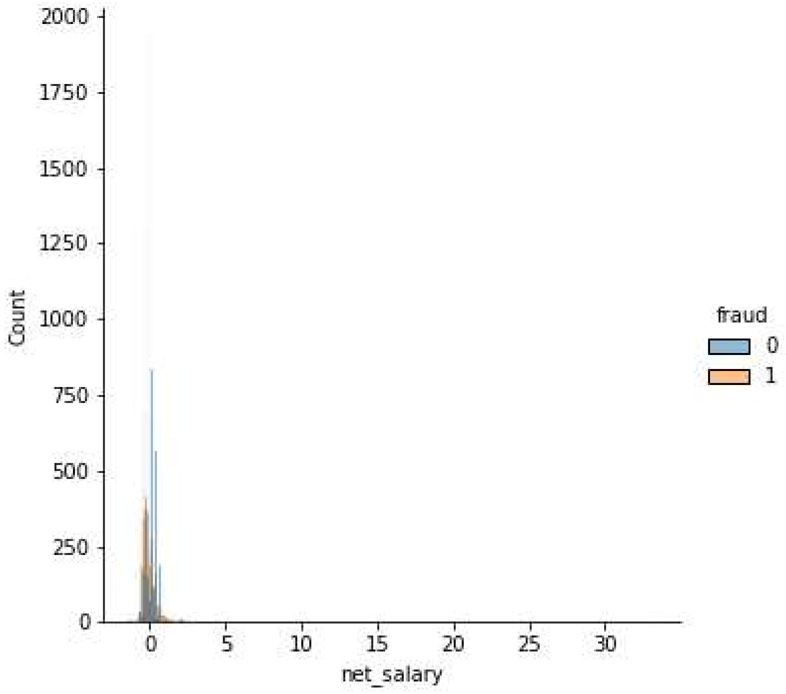
Ethical Considerations

- Risk: The model takes private information of the user and outputs a default risk, which is also a private information. Hence, the users of the model must be made to sign an NDA. Further, there is a possibility that the credit risk output probability might add bias to the human underwriter's own assessment, thus backfiring the intended use.
Mitigation Strategy: None

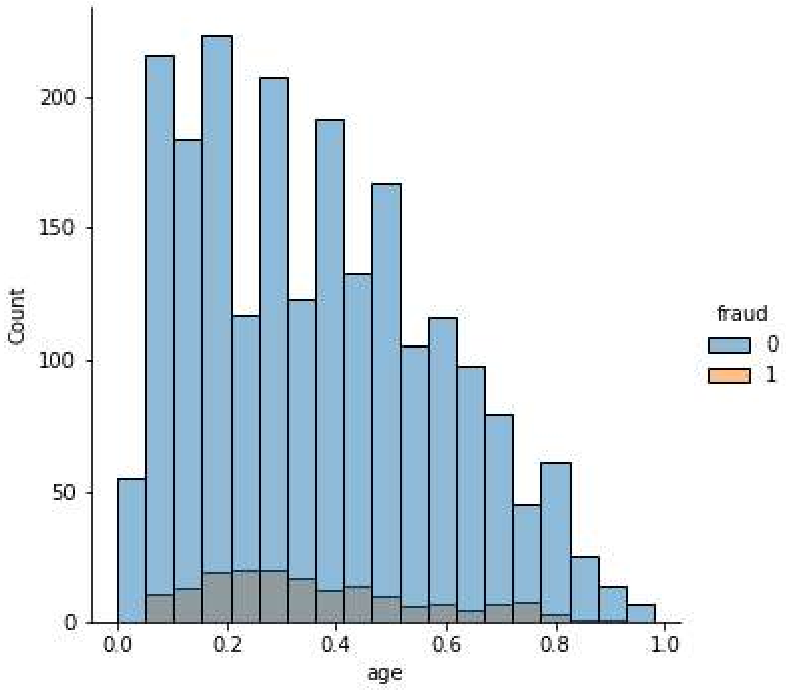
Datasets

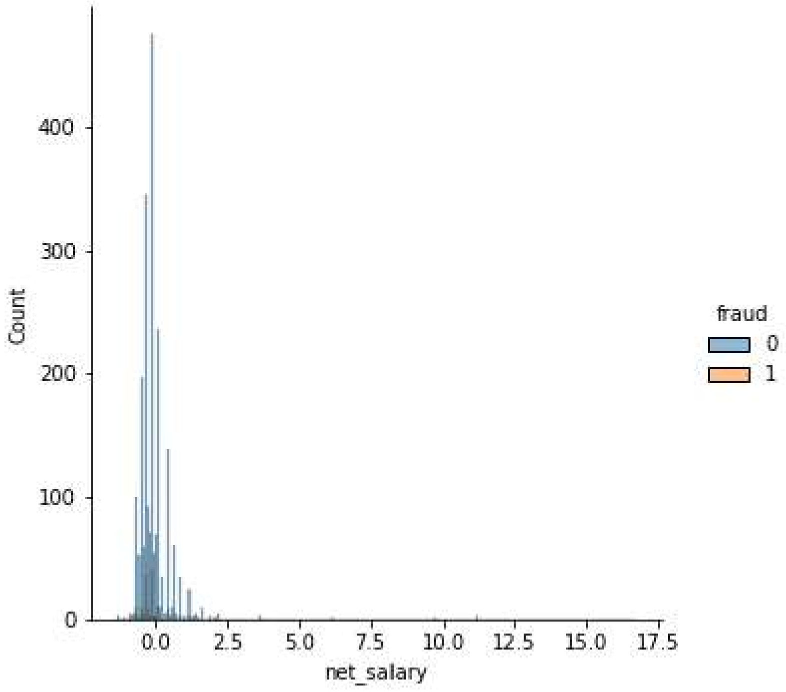
Training Dataset





Test Dataset





Quantitative Analysis

Performance Metrics

Name	Value
Expected profit per customer	Rs. 559.09

ROC curve and confusion matrix