UNIVERSITY OF BIRMINGHAM

School of Computer Science
College of Engineering and Physical Sciences

MSc. Project

Machine Learning & Deep Learning Approaches to Predict Credit Card Default

Submitted in conformity with the requirements
for the degree of MSc. Artificial Intelligence & Computer Science
School of Computer Science
University of Birmingham

Sarathkumar Padinjare Marath Sankaranarayanan
Student ID: 2359859
Supervisor: Dr.Kashif Rajpoot

September 2022

Abstract

The material contained within this report has not previously been submitted for a degree at the University of Birmingham or any other university. The research reported within this report has been conducted by the author unless indicated otherwise.

Keywords Credit Card Default Prediction, Ensemble Learning

Sarathkumar Padinjare Marath Sankaranarayanan

Declaration

The material contained within this report has not previously been submitted for a degree at the University of Birmingham or any other university. The research reported within this report has been conducted by the author unless indicated otherwise.

Signed Sarathkumar Padinjare Marath Sankaranarayanan

Sarathkumar Padinjare Marath Sankaranarayanan

"You have to learn the rules of the game.
And then you have to play better than anyone else"

Albert Einstein

MSc. Project
Machine Learning & Deep Learning Approaches to Predict Credit Card Default

Sarathkumar Padinjare Marath Sankaranarayanan

Contents

Table of Abbreviations

SVM	Support Vector Machine
ANN	Artificial Neural Network
GBDT	Gradient Boosting Decision Tree
GRU	Gated Recurrent Unit
LGBM	Light Gradient Boosting Machine
XGBoost	Xtreme Gradient Boosting Machine
GRU	Gated Recurrent Unit
CV	Cross Validation
SMOTE	Synthetic Minority Oversampling Technique
RAM	Random Access Memory
MSE	Mean Squared Error
ReLU	Rectified Linear Unit
CFS	Correlation Based Feature Selection
AUC	Area Under the Curve
PCA	Principle Component Analsys
RNN	Recurrent Neural Network
RF	Random Forest Classifier

List of Figures

List of Tables

1 Introduction

This section will introduce the user to definitions of terms relevant for understanding the problem, discuss the motivation behind the problem, the aim & approach taken to solve the problem, and the structure of this report.

1.1 Definitions

1.1.1 Credit Card Statement Date

The credit card statement date is the date on which the statement/bill is generated every month. Any transaction conducted on the card post billing date will reflect in the next month's credit card statement.

1.1.2 Delinquent Account

A credit card account is considered delinquent if the customer has failed to make the minimum monthly payment for 30 days from the original due date.

1.1.3 Delinquency Rate

The percentage of credit card accounts within a financial institution's portfolio whose payments are delinquent.

$$DelinquencyRate = \left(\frac{NumberOfDelinquentCreditCardAccounts}{TotalNumberOfCreditCardAccount} \right) * 100 \tag{1}$$

1.1.4 Credit Card Default

The customer is considered as defaulting customer in the event of nonpayment of the due amount in 120 days after the latest statement date.

1.2 Motivation

Delinquency rates & credit card default rates are directly proportional. According to the figure 1, the delinquency rates were at an all-time high just before the recession started in 2008; moreover, this was the same time when more & more customers began to default on credit card payments.

Predicting credit defaults is essential for managing risk in the consumer lending industry. Credit default prediction enables lenders to make the best possible lending decisions, improving customer satisfaction and fostering strong company economics.

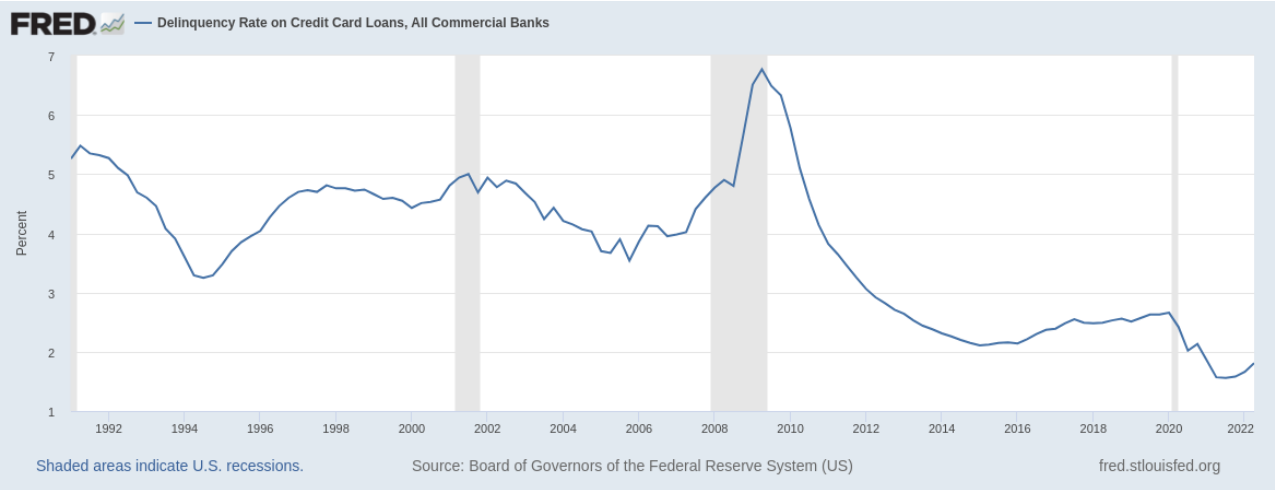


Figure 1: Delinquency rate on credit card loans for the period 1992-2022(Board of Governors of the Federal Reserve System (US) 2022).

Existing models can be used to manage risk. However, developing models that perform better than those in use is feasible.

1.3 Aim & Approach

The objective of this project was to explore different machine learning algorithms & deep learning architectures on the American Express default prediction dataset(American Express 2022) to predict if a customer will default on the payment in the future. The project work started by developing a model using classic machine learning algorithm Support Vector Machine (SVM) followed by creating multiple models using Random Forest Classifier & Gradient Boosting Decision Tree (GBDT) algorithms. Artificial Neural Network (ANN), Gated Recurrent Unit (GRU) & Custom ensemble model created by model combining ANN, GRU and GBDT were created as part of exploring deep learning architectures. Finally created a lean model, using less features & optimzed parameters using GBDT which provided comparable performances to the previously explored models.

1.4 Structure of Report

The remainder of the report is structured as follows: in Section 2 background information on different machine learning & deep learning algorithms along with metrics explanation is provided. Then in Section 3 a literature review related to the credit card default prediction research is given. Section 4 & 5 provides detailed explanations on the dataset, tools & software used in the project, and methodology followed for creating the models. Model evaluation results and the comparison is given in Section 6. Finally Section 7 discusses the conclusion of the project.

2 Background Knowledge

This section provides the reader with the required background information on the machine learning algorithms, deep learning architectures & metrics.

3 Literature Review

3.1 Introduction

This section discusses the current techniques used to predict the credit card default. Since the dataset used for these studies differ, a direct comparison of results is not possible. However, an overall comparison of different techniques and its efficiency in predicting the credit card default will be discussed wherever possible.

3.2 Previous Work

(Sayjadah et al. 2018) developed logistic regression, rpart decision tree & Random Forest Classifier models on dataset generated from credit card operations. The dataset contains 30000 records and 24 features. A Correlation Based Feature Selection (CFS) technique was used to reduce the dimensionality of the dataset. 30% of the dataset was used as the test set for evaluating the performance. (Sayjadah et al. 2018) found that the Random Forest Classifier provided highest Area Under the Curve (AUC) among the models.

(Widyadhana & Prastyo 2021) developed Logistic Regressiion, SVM, ANN & Random Forest Classifier models to preduct credit card default on dataset containing 1000 records and 11 features. The dataset contains credit card data from cardholders from the territory of Indonesia;moreover, the authors used Principle Component Analsys (PCA) to do feature selection also. The dataset is split into 70:30 Train/Test set and the AUC was used to compare the results. The authors found that the Random Forest Classifier outperformed all other models by far and provided a 80% AUC score.

(Hsu et al. 2019) approached the credit card default prediction from a different perspective and proposed a model where dynamic features (time dependent features) were first passed through a Recurrent Neural Network (RNN) network to extract the time dependent features. Then the extracted dynamic features were concatenated with the static features and trained on a Random Forest Classifer. The dataset contained 30,000 samples credit card payment history with 23 features (5 static feature, 18 dynamic features). The authors compared the results of the proposed model with the SVM, Logistic Regression and KNN models and found that proposed method outperformed the others and provided a AUC score of 78%.

(Alam et al. 2020) investigated different approaches to solve the credit card default prediction problem with a specific focus on Class Imbalance. The credit card default prediction dataset are inherently imbalanced as only a small fraction of customers default on credit card. The authors employed different data under/over sampling techniques and evaluated performance on multiple credit card default dataset. They found that GBDT classifier when used with KMeans SMOTE provided the best results and the models performed significantly better on balanced datasets compared to imbalanced datasets.

(Faraj et al. 2021) research shows that ensemble methods consistently outperform Neural Networks and other machine learning algorithms in terms of F1 score. (Faraj et al. 2021) uses the same dataset as the (Sayjadah et al. 2018) which has 30,000 records and 24 features. The authors found that XGBoost provided maximum F1 score compared to Neural Networks, Random Forest Classifier and custom ensemble stacking model. Authors also concludes that the performance of XGBoost model did not improve on balanced datasets. This observation was in contrary to the observations made by (Emil Richard Singh & Sivasankar 2019), the authors of earlier had found that the best performance is achieved on GBDT with KMeans SMOTE method.

3.3 Summary

In conclusion, the ensemble boosting models generally provided better performance than the classic machine learning and deep learning techniques. The data under/over sampling had mixed performances depending on the dataset, some performed better with under/over sampling and some did not. Out of data under/over sampling techniques, KMeans SMOTE performed better. Some studies used feature selection techniques in the data preprocessing techniques which improved the model efficiency. The studies discussed in the section used dataset which were imbalanced and contained atmost 30,000 records; however, (American Express 2022) contains 5 Million records and exploring

the different techniques on such large scale dataset will help us to consolidate the understanding gained from these papers.

4 Materials

4.1 Primary Dataset

The primary dataset contains 190 aggregated profile features of 458913 American Express customers at each statement date for 13 months. Features are anonymized and normalized, and fall into the following general categories:

- D_* = Delinquency variables
- S_* = Spend variables
- P_* = Payment variables
- B_* = Balance variables
- R_* = Risk variables

This dataset(American Express 2022) was released as part of the "American Express - Default Prediction" hosted in Kaggle by the American Express team.

4.2 Secondary Dataset

The secondary dataset was derived from primary dataset by applying the below mathematical aggregate operations to the numerical features.

- Minimum
- Maximum
- Mean
- Last Value
- Standard Deviation

Aggregate for the categorical features were taken by the applying below operations.

- Last Value
- Count
- Unique Value Count

The secondary dataset contains 920 features and 458913 records.

4.3 Tools & Software

The primary programming language used for the implementation of this project is Python version 3.7. Data analysis and manipulation is done using Pandas(1.3.5), seaborn(0.11.2) & Dask(2.12.0) packages. Scikit Learn(1.0.2) package is used for create, train & evaluate machine learning models. ANN & GRU models were created using Tensorflow (2.8.2).Google colab was used to train the model in cloud and Github was used as the version control & project management software.

5 Methodology

5.1 Introduction

This section will first provide a brief overview of the overall strategy of the experiments performed as part of this project followed by providing detailed explanation on the data preprocessing techniques used. Then in subsequent sections each experiment/model will be presented along with the model specific explanations & details.

5.2 Overview of Methodology Followed

Figure 2 represents a overview of methodology in general followed for conducting experiments. The dataset was first split into chunks and stored in different files in parquet format to optimize the memory usage. Then, dataset was preprocessed to remove invalid values and encode categorical text variables to numerical values. Followed by data pre-processing, the dataset was split into Training & Test set, this ensures that none of the entries in test set will have an influence in model training and model selection process. Then the dataset was enhanced using oversampling techniques to resolve the class imbalance issue;in addition, feature selection techniques were used to eliminate the features from the dataset which were less important and hence contribute very little to model.

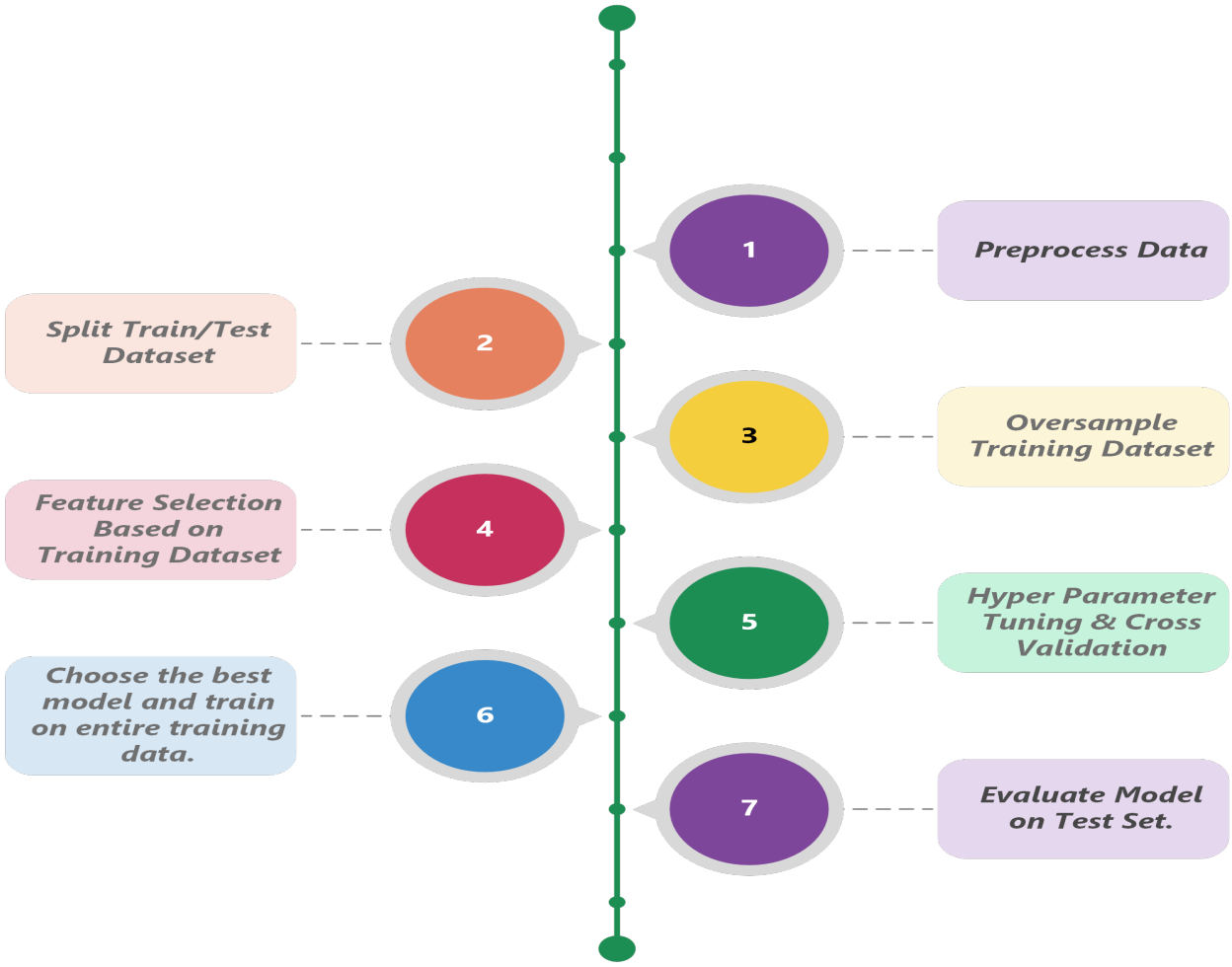


Figure 2: Methodology followed for the experiments

After the feature selection, the model was created and then passed through a Hyperparameter tuning pipeline which helps to find the best parameters for the model which would give highest cross validation score. Finally the entire training dataset was trained on the best model found using hyperparameter tuning and the model was evaluated using the test set set aside at the beginning of the experiment.

5.3 Data Preprocessing

This section discusses the common preprocessing techniques used in all experiments conducted as part of the project. The model specific data preporcessing techniques used will be discussed in respective

sections describing the model.

5.3.1 Default Values

NaN & NULL values in the dataset was replaced by Zero and if a column contains all values same, it was removed from the dataset. -1 was used as the default value for the categorical variables. The categorical variables were encoded using Ordinal Encoder before passing to the model training pipeline.

5.3.2 Normalization

The primary dataset from American Express is already normalized and all the values lies between zero to ten, hence none of the data normalization techniques were used to preporcess the data.

5.3.3 Handling Memory Issue

Google Colab provides 24 GB of Random Access Memory (RAM) in the virtual environment, though the Primary Dataset is 16 GB, the pandas library was unable to load the complete data into memory due to memory leakage issue in the framework. Dask library, which uses multiple Pandas dataframe under the hood, was used to overcome the memory. Dataframe API in Dask library splits the dataset into multiple chunks and loads each chunk on a need basis only (Lazy Loading), this ensured that the complete 16 GB dataset could be loaded even at a low memory of 4GB.

Additionally, the primary dataset was loaded using Dask framework and split the dataset month wise, ie one file for each month. The month wise files were saved in parquet format which helped to reduce the total size of the dataset from 16GB to 7GB. Similarly the primary dataset was also split customer wise, ie 1-50000 customers data in one file, 50001-100000 customers data in second file etc. These files were later used to build the Secondary Dataset.

5.4 Model 1 - Support Vector Machine (SVM)

The SVM model was created with parameters Regularziation Term = L2 Norm(Squared Error Loss), Alpha = 0.0001, Loss='hinge'(soft-margin), tolerance=0.001. The primary dataset was used to train the model and the model converged after 28 iterations. Early stopping was used to prevent overfitting of the model and 10% of the data from training set used as the validation set. Stochastic gradient descent was used to optimize the objective function, this ensured that even though the dataset contains millions of records, the training is able to proceed and finish in reasonable time. 20% of the Secondary Dataset was used as test set to evaluate the performance of the model.

5.5 Model 2 - Random Forest Classifier

The Random Forest Classifier model uses 100 Decision Trees trained in parallel on the primary dataset. Each decision tree uses a different subset of Primary Dataset with maximum number of records in a database set to 600,000. Gini impurity metric is used to measure the quality of the split while building decision tree. Finally the model predicts the target variable by taking mean of all the predictions from the 100 individual decision trees. 20% of the Secondary Dataset was used as test set to evaluate the performance of the model.

5.6 Model 3 - Gradient Boosting Decision Tree (GBDT)

GBDT model was created using 100 Decision Trees trained sequentially on the primary dataset. Friedman Mean Squared Error (MSE) is used to measure the quality of a split; additionally, model was set to use only 60% of the data for constructing each decision trees to avoid memory leakage issue. 10% of the training set was set for validation purpose; furthermore, the parameters were set to stop the training if the validation score does not improve to avoid overfitting. Loss function for the training was set to Deviance. 20% of the Secondary Dataset was used as test set to evaluate the performance of the model.

5.7 Model 4 - Xtreme Gradient Boosting Machine (XGBoost)

XGBoost model was created using training 100 base learners on the Secondary Dataset and each base learner is constructed using 80% of the training dataset. Instead of using complete features to construct the base learner, parameters were set to use only 60% of features, this helped to eliminate the memory leakage/overflow issues while training. Moreover, L2 regularization parameter was set

to 0.9 to reduce the overfitting of the model. 20% of the Secondary Dataset was used as test set to evaluate the performance of the model.

5.8 Model 5 - Light Gradient Boosting Machine (LGBM)

LGBM model was trained on Secondary dataset and the 100 base learners were constructed using the entire features & training set. Tradition Gradient Boosting Decision Trees were used as the boosting type and learning rate was set to 0.1. 20% of the dataset were set aside as the test set for evaluating the model. Maximum depth is not set as to allow trees of any depth.

5.9 Model 6 - Artificial Neural Network (ANN)

Figure 3 depicts the architecture of the custom ANN model developed. The primary dataset was first split into training & test dataset, followed by oversampling the training dataset using KMeans SMOTE to make the percentage of defaulting & non defaulting customers equal. Then the training dataset was trained using the custom ANN model. The first & second layer uses Rectified Linear Unit (ReLU) as the activation function, however the final layer uses Sigmoid as the activation function. Adam optimizer was used to optimze the objective binary cross entropy loss function. The trained model was tested and evaluated on the test set.

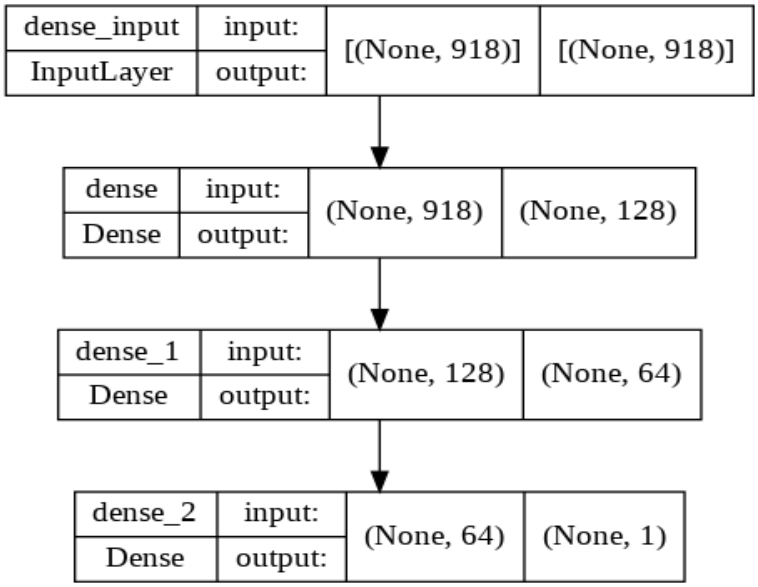


Figure 3: Custom Neural Network Architecture

5.10 Model 7 - Gated Recurrent Unit (GRU)

Figure 4 represents the architecture of the GRU based model for predicting credit card default. The primary dataset is used for training this model;furthermore, the tanh function is used as the activation function and sigmoid is used as the recurrent activation function for the GRU layer. Dropout of 10%, recurrent dropout of 50% added to reduce the overfitting problem. The output of the GRU layer is then fed to dense layer followed by another dense layer with activation sigmoid for making final prediction. Optimizer used is Adam & the loss function is Binary Cross Entropy. A dataset generator was created to provide input to the model in chunks, this helped to eliminate the memory issues while training. The final model contains 49165 trainable parameters.

5.11 Model 8 - Ensemble Stacking Model using ANN + GRU + GBDT

Figure 5 partially depicts the architecture of the custom ensemble stacking model. The primary dataset is first trained using GRU layer followed by a dense layer. In parallel, the secondary dataset is trained using a 2 dense layers. Then the output of these two parallel legs were combined to form the concatenation layer. The output of concatenation layer is then trained using a GBDT model to get the final prediction. Adam optimizer was used to optimze the objective binary cross entropy loss function. Instead of loading complete dataset into memory and train the entire dataset in one go,

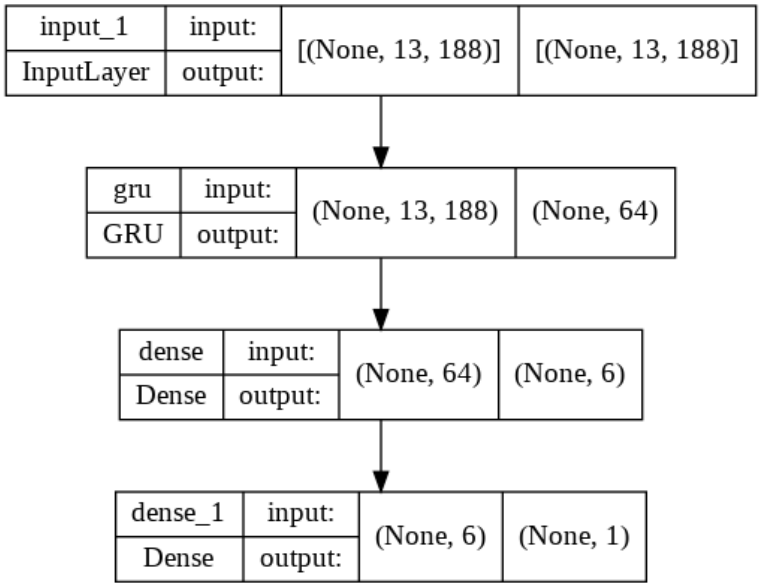


Figure 4: GRU Model Architecture

a dataset generator was written to return chunks of data for training, this helped to eliminate the memory overflow issues.

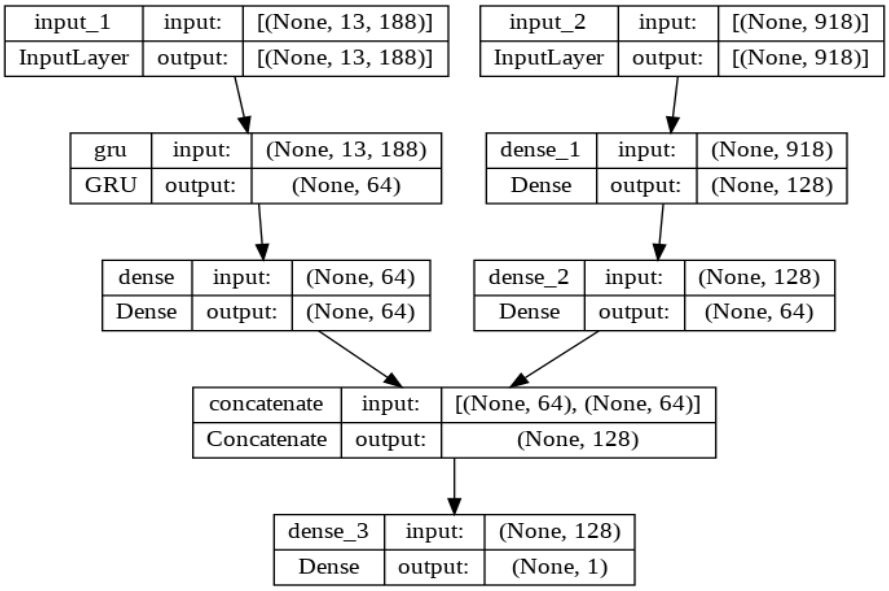


Figure 5: Custom Ensemble Stacking Model Architecture

5.12 Model 9 - Lean LGBM Model

Finally, a lean model was created using LGBM model which peformed on par with the other models but with less resources & data. Firstly the 20% of the dataset was set aside as test set and remaining 80% for training purpose. Training dataset was then oversampled using KMeans SMOTE method which resulted in the number of defaulting customers & non defaulting customers to become equal. Secondly the the training dataset was trained using a simple SVM model to extract the feature importances; in addition, the features with feature importance weight less than the mean of importance of weights were discarded. This helped to reduced the feature count from 920 in the secondary dataset to 279. This modified dataset was then passed through a Grid Search CV pipeline to choose the best parameters for maximum depth & maximum number of leafs for base learner trees, and boosting type. Cross Validation (CV) Recall score was used as the metric to choose the best model. Finally, a LGBM model was created using the best model parameters found using GridSearchCV and trained the same on the complete training dataset. The final model was tested and evaluated on the test set.

5.13 Summary

Firstly, this section presented overall methodology followed for the experiments followed by the data preprocessing techniques used. Handling of invalid feature values, normalization & handling memory leakage and memory overflow issue were discussed in the Data Preprocessing section. Secondly, the 9 different models created to solve the problem were discussed. Initially SVM model & Random Forest Classifier model were discussed followed by more advanced machine learning techniques such as GBDT, XGBoost & LGBM. Then 3 models using deep learning techniques such as Neural Network, GRU and Ensemble Stacking model were presented. Finally a lean model was developed and presented which used less features for training, was computationally less expensive and was explainable model. Before training the final model, the model was passed through a data oversampling pipeline, feature selection pipeline. The model was also tuned using GridSearchCV method to find the optimal hyper parameters. In all the experiments 20% of data was set aside before training for testing & evaluation purposes.

6 Results & Discussions

6.1 Introduction

Firstly this section presents the test result of the experiments conducted based on the methodology described in section 5. Secondly the main highlights or achievements of the project is discussed. Finally, limitations of this project as well as the future works are discussed.

6.2 Test Results

Table 1 depicts the result of running the models discussed in section 5 on the test set. 4 metrics, F1 Score, Recall, Accuracy & Precision is used to compare the performance of the models. Though 4 metrics are tracked, Recall & F1 Score are the metric which is used for Hyper Parameter Tuning, Early Stopping, Cross Validation across all experiments. This decision was taken based on observation that Accuracy will not be a good metric as the dataset is imbalanced; furthermore, the F1 score & Recall provides a better indictation on how well the model is able to predict the defaulting clients correctly.

Model	F1 Score	Recall	Accuracy	Precision
SVM	74.66	75.57	87.24	73.78
RF	73.81	72.78	87.13	74.86
GBDT	74.33	74.13	87.25	74.52
XGBoost	80.22	80.11	89.79	80.34
LGBM	81.01	81.11	90.13	80.91
ANN	81.03	81.81	90.09	80.27
GRU	79.81	79.96	90.67	79.65
GRU+ANN+GBDT	80.09	79.71	90.87	80.49
Lean LGBM	80.95	80.68	90.15	81.23

Table 1: Comparison of metrics on various models.

The SVM provided an accuracy of 87.24% & F1 score of 74.66 which was impressive considering the model took very less time to train compared to other models; thus, SVM model was used in Lean LGBM model as part of feature selection pipeline. RF model performed worst on F1 Score, Recall & Accuracy among all the models experimented as part of this project. This reduced performance of RF could be due to the large dataset & dimensionality and the algorithm is not able to generalize well with 100 decision trees. GBDT model provided similar results to SVM model, however, GBDT provided better Precision while SVM provided better Recall scores. Ensemble boosting techniques XGBoost & LGBM performed really well the test set such that F1 score & Recall increased almost by 6% compared to SVM model. LGBM model performed slightly better than XGBoost model in all the metrics and LGBM model took less time to train compared to XGBoost.

ANN model provided the highest F1 Score & Recall among all the models, however, the difference in F1 Score of LGBM and ANN model is only 0.02. ANN model provided better Recall score while

LGBM provided better Precision score. GRU+ANN+GBDT model provided best accuracy score of 90.87 among all the models. GRU provided the second best accuracy score;moreover, considering that GRU model has 49,165 trainable parameters while the GRU+ANN+GBDT has 178,945, the GRU model is able to extract the features better than GRU+ANN+GBDT with less parameters. The table 2 shows the trainable parameter count for the deep learning models explored.

Model	Trainable Parameters
ANN	125,953
GRU	49,165
GRU+ANN+GBDT	178,945

Table 2: Trainable parameters comparison of Deep Learning Models

The Lean LGBM model provides a F1 Score of 80.95 which is only 0.08 less than the best performing ANN model; futhermore, the Lean LGBM provided the best Precision score among all the models. Lean LGBM model uses only 1/3rd of the features used by the ANN model and still provides comparable performance. The figure 6 represents the confusion matrix of all the Machine Learning models explored during this project.

6.2.1 Feature Importance

The table 3 represents the top 5 features based on the importance from model Lean LGBM. This is another advantage of Lean LGBM model over the ANN model, the Lean LGBM model is explainable. However, the features in the Primary Dataset & Secondary Dataset is anonymized, hence it is not possible to know the exact real life meaning of these features.

Feature	Importance Score
D_39_last	226
P_2_last	214
B_4_last	191
B_3_last	162
B_1_last	144

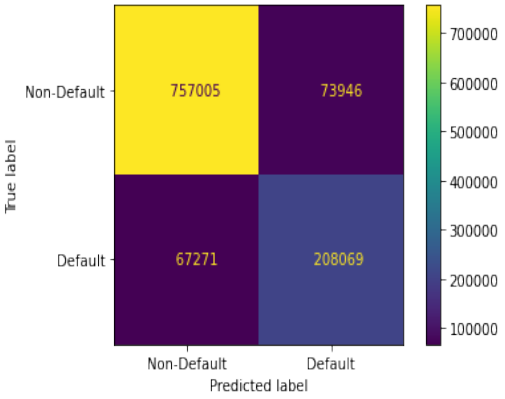
Table 3: Important Features Extracted Based on Lean LGBM model (Secondary Dataset)

6.3 Achievements

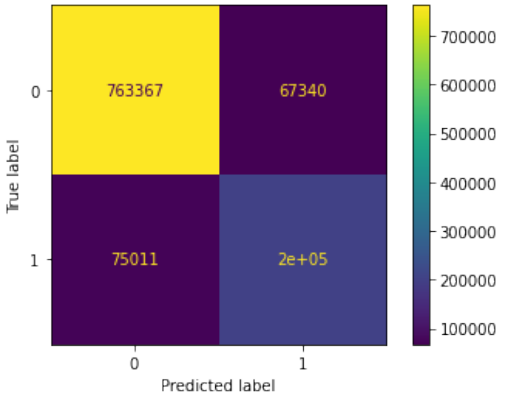
As mentioned in 3, most of the previous work on credit card default prediction was based on dataset containing records less than 100,000. However the Primary Dataset used in this work has around 5 Million records of 400,000 unique customers. Most of the previous works concluded that the Ensemble Boosting classifier such as XGBoost & GBDT performed better than the traditional machine learning algorithms, this is reconfirmed on large dataset with the work of this project. On the other hand, this work helped to identify that the difference in peformances between the Deep Learning Network & Ensemble Boosting Algorithms reduced on large dataset, however it the differences were significant in the smaller datasets. Finally, it was reconfirmed that the models performed better on balanced data and KMeans SMOTE is a suitable candidate for fixing the class imbalance.

6.4 Limitations & Future Work

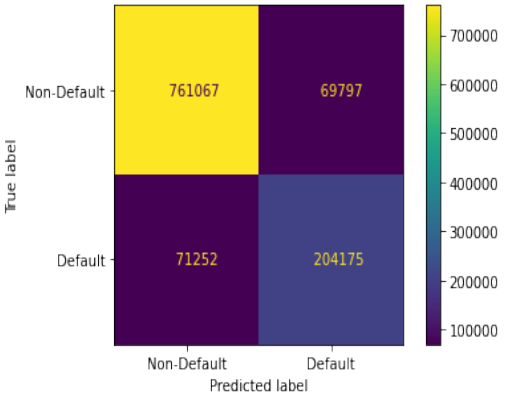
6.5 Summary



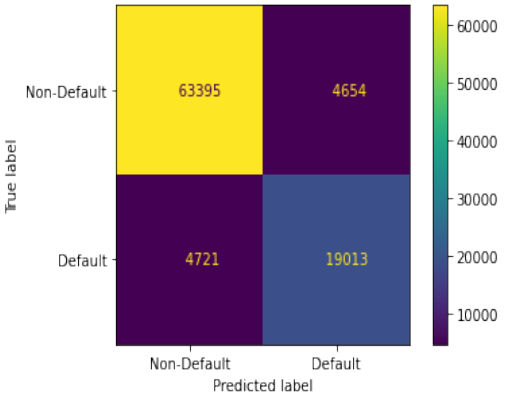
(a) SVM



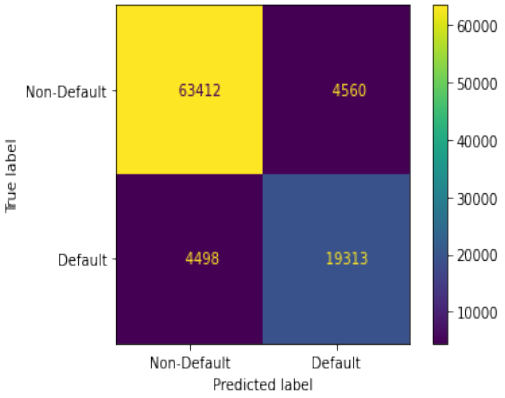
(b) RF



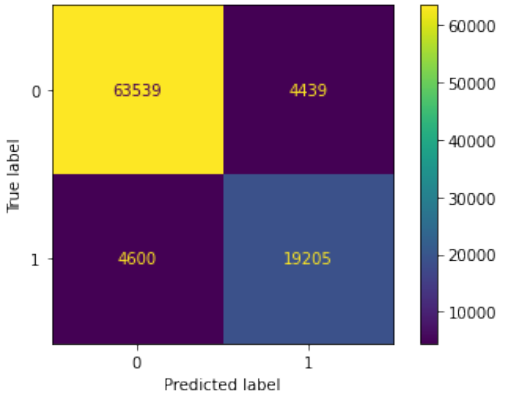
(c) GBDT



(d) XGBoost



(e) LGBM



(f) Lean LGBM

Figure 6: Confusion Matrix of Machine Leaning Models

7 Conclusion

References

Alam, T. M., Shaukat, K., Hameed, I. A., Luo, S., Sarwar, M. U., Shabbir, S., Li, J. & Khushi, M. (2020), 'An investigation of credit card default prediction in the imbalanced datasets', *IEEE Access* **8**, 201173–201198.

American Express (2022), 'American Express Default Precition Dataset', Available: `https://www.kaggle.com/competitions/amex-default-prediction/data`.

Board of Governors of the Federal Reserve System (US) (2022), 'Delinquency Rate on Credit Card Loans, All Commercial Banks [DRCCLACBS]', Available: `https://fred.stlouisfed.org/series/DRCCLACBS`. Data retrieved from FRED, Federal Reserve Bank of St. Louis;.

Emil Richard Singh, B. & Sivasankar, E. (2019), Enhancing prediction accuracy of default of credit using ensemble techniques, *in* 'First International Conference on Artificial Intelligence and Cognitive Computing', Springer, pp. 427–436.

Faraj, A. A., Mahmud, D. A. & Rashid, B. N. (2021), 'Comparison of different ensemble methods in credit card default prediction', *UHD Journal of Science and Technology* **5**(2), 20–25.

Hsu, T.-C., Liou, S.-T., Wang, Y.-P., Huang, Y.-S. et al. (2019), Enhanced recurrent neural network for combining static and dynamic features for credit card default prediction, *in* 'ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 1572–1576.

Sayjadah, Y., Hashem, I. A. T., Alotaibi, F. & Kasmiran, K. A. (2018), Credit card default prediction using machine learning techniques, *in* '2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)', IEEE, pp. 1–4.

Widyadhana, K. N. & Prastyo, D. D. (2021), 'Credit card default prediction using machine learning'.

8 Appendix One: Code

8.1 Directory Structure

8.2 Running the Provided Code