CMPE 255 - Data Mining

Loan Defaulter Classification and Prediction

Team 13

Group Members

Anish Shriram Kulkarni - 015246153

Anoushka Shailendra Gade - 015292108

Satish Dallaram Choudhary - 014488591

Saurabh Suryakant Sathe-015272023

Github link: https://github.com/satishc9757/loan-defaulter-prediction

Collab link:
https://colab.research.google.com/drive/1KLJ83eZdMq9b78EjgntrHUlPSBRCDfx5#scrollTo=eBu1snlNcS8o&uniqifier=1

1. Introduction

1.1 Motivation

Finance Organizations face huge losses due to defaulters. This has led to huge loan rejection rates. This has also adversely affected loan organizations and vehicle manufacturers by significantly reducing business. Hence there is a need for a more robust and efficient risk prediction system which can let the financial institutions know about the possibility of an individual being a defaulter.

1.2 Objective

In this project we aim to accurately predict the probability of borrower defaulting on a vehicle loan in the first EMI (Equated Monthly Instalments) on the due date. Doing so will ensure that clients capable of repayment are not rejected and important determinants can be identified which can be further used for minimizing the default rates.

The objectives are as follows:

- Analyse and train our classifier model based on the loan data.
- Predict and evaluate our model for accuracy of loan defaulter prediction

2. System design and Algorithms

2.1 Algorithms Selected:

2.1.1 XGBoost classifier

Extreme Gradient Boosting or XGBoost, is the first algorithm that we chose for the classification task. It has the following advantages:

a) Effective Tree Pruning :

Xgboost is just like random forest classifiers built trees. The trees in the Random Forest algorithm follow a greedy algorithmic approach where the nodes stop splitting once they encounter a negative loss. In the case of XGBoost, the trees are built till the maximum depth specified. The trees are then pruned upwards beyond which there is no positive gain. This helps in creating trees which have more efficient splitting criterion.

b) Regularization:

Xgboost has built in L1 and L2 norms which help in avoiding model overfitting. Hence Xgboost is also called as regularized gradient boosting.

c) Cross Validation:

Xgboost allows to run cross validation at each iteration and get optimum iterations in a single run

2.1.2 Random Forest Classifier

Random forest is a class of ensemble machine learning algorithms. It trains a number of decision trees and predicts labels by considering the predicted labels from all the individual decision trees. It is one of the most widely used machine learning algorithms for classification. We chose Random Forest Classifier for the following reasons:

a) Ensemble Nature:

Since we train a number of decision trees and consider their predictions, this algorithm considers and covers a number of possibilities while predicting the final label. These also result in a better model with better accuracy.

b) Highly Customizable

There are a number of hyper parameters that can be used to customize random forest algorithms like the number of trees, the impurity measure, maximum depth of trees and many more that help cater the needs of our dataset.

2.1.3 KNN classifier

KNN is a machine learning algorithm which classifies data based on K similar records. The similar records are called its neighbors hence the name K Nearest Neighbors. We chose KNN for the following reasons:

a) Easy Implementation:

KNN is a lazy learning algorithm which is extremely easy to train as it just makes predictions by dynamically considering its K neighbors.

b) Fast:

As KNN is a fairly fast algorithm, it helps in parameter tuning. Trying out different values of K is easy and fast. This helps in improving model accuracy by choosing the right value of K

2.1.4 Artificial Neural Networks:

Neural Networks are a class of machine learning algorithms which mimic the functionality of the human brain. We chose artificial neural networks for the following reasons:

a) Ability to learn linear and non-linear relationships:

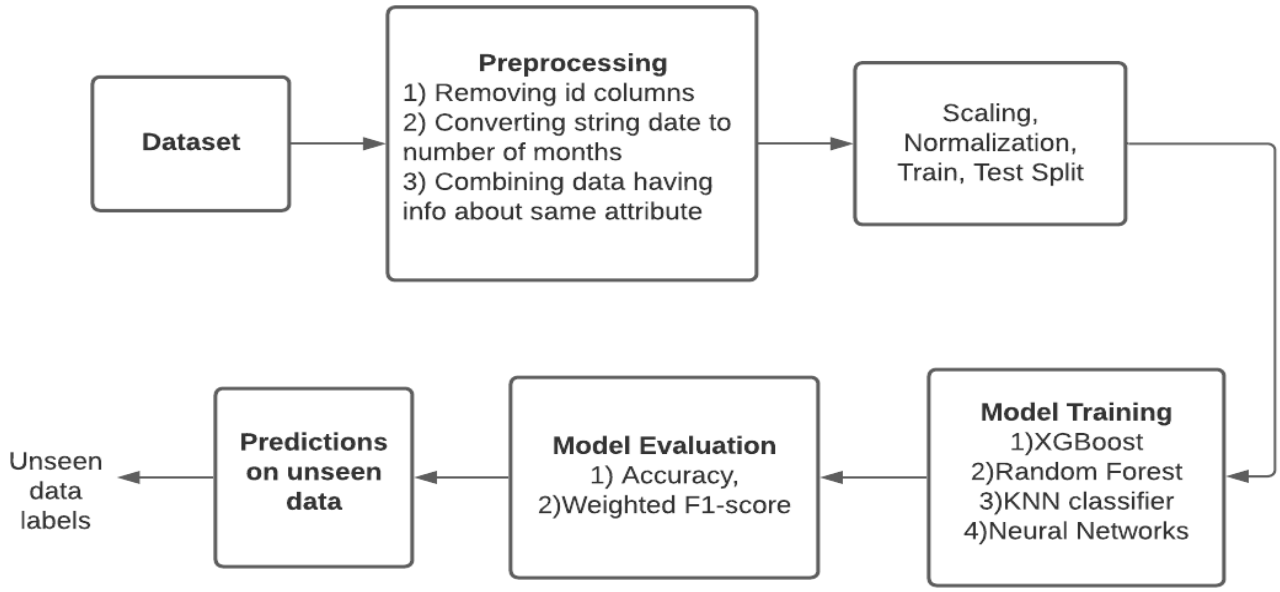
There can exist a linear or a non linear relationship between the data attributes and artificial neural networks can easily learn and infer from them

b) Multi layer architecture:

Due to the multi-layered architecture of Neural Networks ,the hidden layers are known to accurately capture the relationship between the attributes by optimizing the weights of the links between layers

c) Durability: Neural Networks are known to be highly durable. Even if one or more cells of the model are poorly trained, the network is known to give good results

2.2 System Design



2.3 Technologies and tools used

1. Python - For programming the classifier
2. Google Colab notebooks - For execution environment
3. Sklearn - For the inbuilt features such as test_train_split, scalers, metrics etc
4. Pandas - For data transformations and analyses
5. Numpy - For numerical calculations
6. XGBoost - For classification

3. Experiments and Proof of Concept evaluation

3.1 Dataset Used

The dataset contains the loan and borrower information as follows:

- Borrower Information includes demographic data, age, date of birth, employment type, and identity proof
- Loan information includes loan amount, disbursal date, supplier ID and few other parameters regarding loan
- History information of the borrower includes number of active accounts, previous loans and EMIs, Bureau history, number of inquiries

The dataset is in CSV format. It has a total 233155 rows and 41 columns representing the features and the class label where 0 means “Not a defaulter” and 1 means “Defaulter”

Link to dataset:
https://www.kaggle.com/mamtadhaker/lt-vehicle-loan-default-prediction?select=train.csv

3.2 Data Pre-processing

3.2.1 Removing Unique IDs

The unique ID columns like *Branch_Id, Current_Pincode, Employee_code_ID, UniqueID, State_ID, supplier_id, MobileNo_Avg_Flag* does not add any value to the classifier prediction model and hence removed from the training data.

3.2.2 Aggregating Disbursal Date and Date of Birth

The columns *Disbursal Date* and *Date of Birth* are clubbed into a single column by calculating the age of borrower at the time of loan disbursement.

3.2.3 Formatting the Average Account age and Credit History length to months

The columns *Average Account age* and *Credit History length* contain period information in the format “1yrs 11mon” which makes it difficult to process. These column values are parsed into a number of months.

3.2.4 One hot encoding of Employment.Type column

One hot encoding is a process of converting categorical data into some form that is better suited for the machine learning algorithms. In this project, the Employment.Type column contains values “Self employed” and “Salaried”. These are encoded into 0 and 1 respectively. Any other value is counted as 2.

3.2.5 One hot encoding of PERFORM_CNS.SCORE.DESCRIPTION column

The PERFORM_CNS.SCORE.DESCRIPTION column gives useful data about the bureau history of the borrower. This data is categorical and hence a good candidate for encoding into numerical value.

We have done below encoding for this column:

```
def clean_credit_risk(row):
  risk_category = {'unknown':-1, 'A':13, 'B':12, 'C':11,'D':10,'E':9,'F':8,'G':7,'H':6,'I':5,'J':4,'K':3, 'L':2,'M':1}
  value = row['PERFORM_CNS.SCORE.DESCRIPTION'].split("-")
  if len(value) == 1:
    # No Bureau History Available
    return -1
  else:
    rc = risk_category[value[0]]
    return rc
```

The encoding is based on the following definitions of bureau description, where A is given the highest value and M the lowest. The unknown columns are encoded as -1

- ☑ A-Very Low Risk
- ☑ B-Very Low Risk
- ☑ C-Very Low Risk
- ☑ D-Very Low Risk
- ☑ E-Low Risk
- ☑ F-Low Risk
- ☑ G-Low Risk
- ☑ H-Medium Risk
- ☑ I-Medium Risk
- ☑ J-High Risk
- ☑ K-High Risk
- ☑ L-Very High Risk
- ☑ M-Very High Risk
- ☑ No Bureau History Available
- ☑ Not Scored: More than 50 active Accounts found
- ☑ Not Scored: No Activity seen on the customer (Inactive)
- ☑ Not Scored: No Updates available in last 36 months
- ☑ Not Scored: Not Enough Info available on the customer
- ☑ Not Scored: Only a Guarantor
- ☑ Not Scored: Sufficient History Not Available

3.2.6 Merging of account columns

The dataset contains information about primary and secondary accounts. Since the bank cannot disregard the number of secondary accounts while making a classification decision, the 2 columns are added and to a third column prefixed with ‘total’
The following columns are involved in the account merging operation -

```
data['all_accounts'] = data['PRI.NO.OF.ACCTS'] + data['SEC.NO.OF.ACCTS']
data['primary_inactive_accounts'] = data['PRI.NO.OF.ACCTS'] - data['PRI.ACTIVE.ACCTS']
data['secondary_innactive_accounts'] = data['SEC.NO.OF.ACCTS'] - data['SEC.ACTIVE.ACCTS']
data['total_inactive_accounts'] = data['primary_inactive_accounts'] + data['secondary_innactive_accounts']
data['total_overdue_Accounts'] = data['PRI.OVERDUE.ACCTS'] + data['SEC.OVERDUE.ACCTS']
data['total_balance'] = data['PRI.CURRENT.BALANCE'] + data['SEC.CURRENT.BALANCE']
data['total_sanctioned_amount'] = data['PRI.SANCTIONED.AMOUNT'] + data['SEC.SANCTIONED.AMOUNT']
data['total_disbursed_amount'] = data['PRI.DISBURSED.AMOUNT'] + data['SEC.DISBURSED.AMOUNT']
data['total_installment'] = data['PRIMARY.INSTAL.AMT'] + data['SEC.INSTAL.AMT']

data=data.drop(['PRI.NO.OF.ACCTS','SEC.NO.OF.ACCTS','PRI.CURRENT.BALANCE',
                'primary_inactive_accounts','secondary_innactive_accounts',
            'PRI.SANCTIONED.AMOUNT','SEC.NO.OF.ACCTS','PRI.NO.OF.ACCTS',
            'PRI.DISBURSED.AMOUNT','PRI.ACTIVE.ACCTS', 'PRI.OVERDUE.ACCTS',
            'SEC.CURRENT.BALANCE','SEC.SANCTIONED.AMOUNT', 'SEC.OVERDUE.ACCTS',
            'SEC.DISBURSED.AMOUNT','PRIMARY.INSTAL.AMT','SEC.INSTAL.AMT',
            'disbursed_amount','SEC.ACTIVE.ACCTS'],axis=1)

```

3.2.7 Scaling of numerical columns

Banking data heavily depends on outliers. So we tried to preserve the outlier records instead of removing them. We used RobustScaler to bring the observations closer to the median value.

3.2.8 Adding new feature for 0-value columns

For records with zero observations, we created a new feature column that counts the features having zero. This will act as a differentiator between people who have a credit history and those who don't. People with no credit history will likely have more than 9 features as zero, unlikely in case of customers with credit history.

3.3 Methodology

The training data (CSV file) was read and only the necessary columns were kept. The feature extraction, cleaning, aggregation operations were done. Then, the account-related primary and secondary features were merged. The categorical columns were encoded with numeric values, and other non-categorical numeric columns were scaled using RobustScaler.

For training the model, the data was split into a test and training set with 67:33 ratio. After that, a classifier was chosen for training the model.

We used four classifiers to predict the category of the loan defaulter. Out of the five classifiers, we evaluated each of the classifiers using the accuracy_score metric from sklearn.metrics. After evaluation, we got the best accuracy by using the Random Forest classifier with an accuracy score of 91 %.

3.4 Algorithm Comparison

Model	Accuracy	Weighted f1-score
Random Forest	**91%**	**0.9072**
XGBoost	**90.06%**	**0.89**
KNN	**88.3%**	**0.88**
Neural Network	**86.1%**	**0.86**

4. Discussions and Conclusions

4.1 Discussions and Decisions

- Our team discussed and searched for a dataset that we thought might be a perfect fit for our project.
- After we found the dataset, the team discussed and noted down key tasks that are to be undertaken for the project.
- We also thought out the strategies for preprocessing and cleaning of data which can be used so we can get the best possible data to work on to get the best possible results.
- Finally, the team narrowed down on the algorithms to be used for each task so as to get the required result in optimum steps.

4.2 Issues Faced

- The dataset that the team selected has many banking domain-intensive columns, which required knowledge of loan process, credit history and risk categories
- Selecting which columns to drop and which to keep was a highly challenging activity
- The date columns in the dataset were not in DD/MM/YYYY format, which had to be dealt with through additional logic.
- We needed a lot of time in cleaning, preprocessing and performing transformation techniques on the data.

4.3 Things that worked well

- Data Preprocessing went really well as the data was really complex and a lot of research went into identifying various categories in the data.
- The dataset contained a very few null values, which helped in better understanding and evaluation
- We also learned various terminologies in the business and financial services(BFSI) domain.

- Model training also went well as we got to experiment with various algorithms and tune them during the classification process
- During model evaluation we used weighted F1-score which is a great metric to measure the performance of classification models having disproportionate distribution of class labels. Choosing the weighted F1-score was one of the best decisions we made

4.4 Things that didn't work well

- The accuracy we achieved was great but could have been better with Neural Networks. We applied all the best classification algorithms available in the industry today. However we may not have chosen the best number of hidden layers for neural networks
- Our choices of layers were restricted may be due to insufficient domain knowledge of deep learning and the hardware restrictions
- Hence, we think that there was definitely room for improvement in neural networks.
- We are quite sure that given the appropriate number of hidden layers, it would have easily outperformed other algorithms.

4.5 Conclusion

- While handling this project, we learned how various data mining techniques and algorithms can enhance the real life applications and help giant corporations to maintain and serve a large user base.
- How to handle big datasets by cleaning and transforming the data according to your need was a big learning.
- We also learned to create and use classification and prediction models according to a certain use case.
- Lastly, we learned that the loan process (eligibility for disbursement) can be aided using proper data mining techniques and prediction models.

5. Project Distribution

Task	Role
Dataset	All
Data Cleaning	Anoushka, Satish
Preprocessing of Data	Anoushka, Satish
Algorithms	Saurabh, Anish
Performance Evaluation	Saurabh, Anish
Project Preparation	All
Report	All
PPT	All