An Investigation on Fraud Detection Using Machine Learning Algorithms

Vishal Jain

Roll No : 2017IMT-090
ABV-IIITM Gwalior
Gwalior-474 010, MP, India

November 8, 2020

Table of Contents

Introduction

- Fraud detection have become a painful task for banking, ecommerce and medical.
- According to cyber source 83% businesses conduct manual reviews.
- India is among top 5 country with regard to credit/debit fraud detection.
- In 2007 credit/debit fraud cases jumped by 42%.
- As of 2019 there are around 52 million credit cards in India.

Contd...

- USA is the most prone country with 38.6%cases in 2018
- Identity theft is identified as one of the main reason
- Lack of consumer awareness.
- Use of skimmers to get information and later use them to produce fake cards
- 30-50 age group is the most vulnerable to these frauds.
- Leads to many health problem like sleep problems, increased stress, anger and frustration.

Background

- Credit card is a physical payment card provided by bank which allows users to pay after certain time.
  - Types of credit card:
  - Travel Credit Card
  - Fuel Credit Card
  - Reward Credit Card
  - Shopping Credit Card
  - Secured Credit Card

Contd...

- In 1994 fraudsters used fake names for fraud.
- In 1996 online status check of stolen cards.
- In 1999 hackers started using social engineering, honeypot etc to commit fraud.
- Introduction of ecommerce has lead to surge in demands and frauds.
- AI can play a key role by training it on a set of inputs to generalize in real world.

Motivation

- Problems with manual review in Fraud detection are:
  - Costly
  - Labour intensive task
  - Time consuming
  - High false positive
- Traditional approaches failed due to:
  - Increase in data
  - Variations in types of transactions.
  - Example: Rule based approaches.

Objective

- To study and analyze how machine learning algorithms perform on balance and unbalance dataset.
- Regression examples:
  - Logistic Regression
  - Logistic Regression with minmaxscaler
- Ensemblers example:
  - Decision Tree
  - Random Forest
- Boosting example:
  - Adaboost
  - Catboost
  - Xgboost
  - Lightgbm
- Class balancing algorithms:
  - Smote
  - Adasyn
  - Allknn

Literature

- Both supervised and unsupervised are used in real world scenario.
- Problem with supervised algorithms:
  - Requires accurate labelling of transactions.
  - Built to differentiate between legitimate and previous known frauds.
- Issues with dataset:
  - Evolving data
  - Changing data
  - May contain fraudster entry as legitimate due to similarity.

Dataset description

- Dataset contains 284,807 transactions.
- 0.172% transactions are fraudulent transactions.
- Most of the features in the dataset are transformed using principal component analysis (PCA).
- V1, V2, V3,..., V28 are PCA applied features.
- Other features include time, amount and class are non-PCA applied features.

Deliverables

- Analyze models on the basis of metrics such as accuracy, precision, recall, f1score, mcc and roc.
- Analyzing various machine learning algorithms like ensemblers, boosting, regression techniques etc
- Using 20% dataset as test data in all algorithms.
- To study effect of smote, allknn and adasyn on machine learning algorithms.

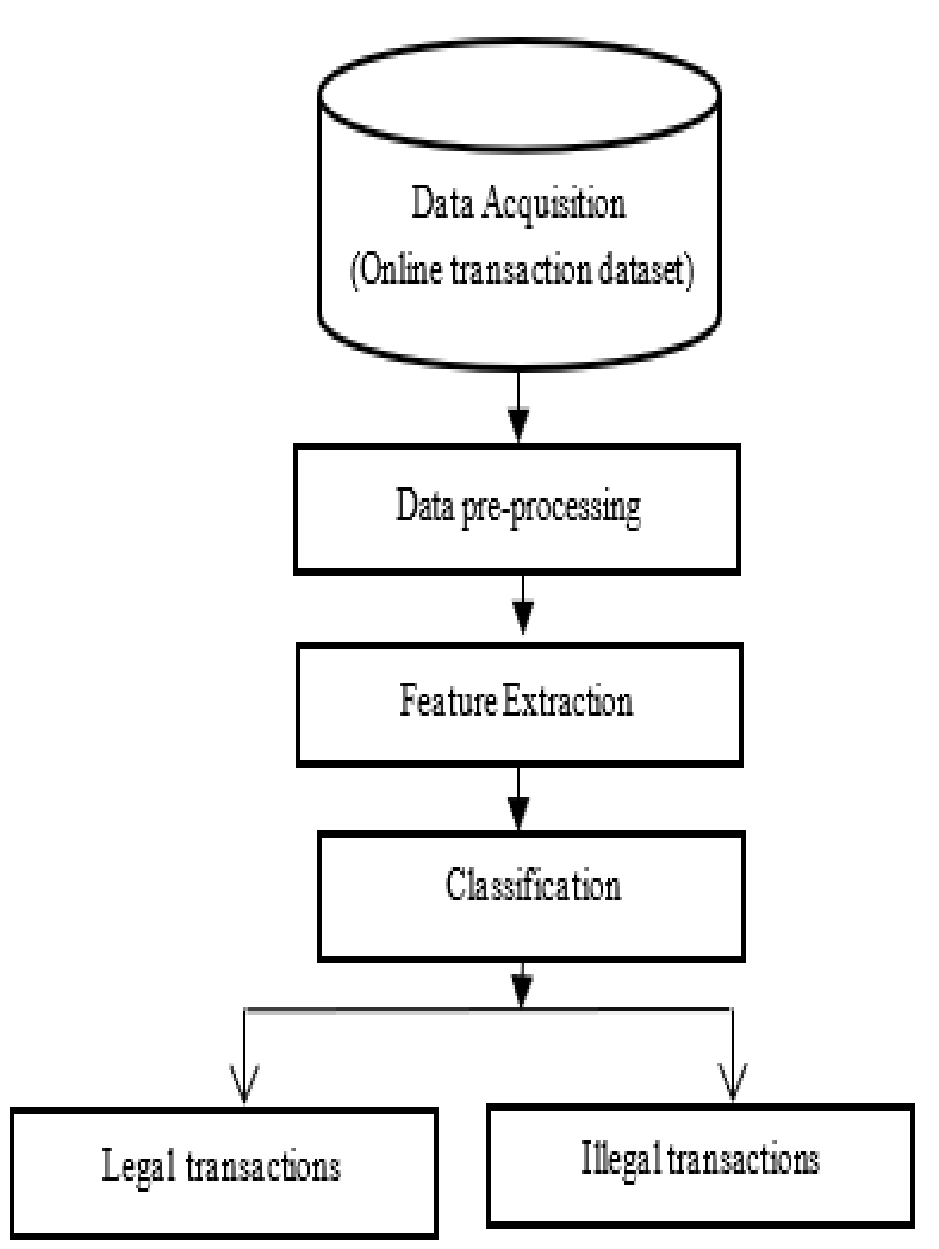
Salient Features

- Only supervised machine learning algorithms are used for analysis purpose.
- Decision trees take decisions at each step and random forest take account of various decision trees while making a conclusion.
- Most of the algorithms used for the analysis requires pretraining on a dataset to generate an output.

System architecture

- It is divided into 4 steps:
  - Data Acquisition - Collecting data from various resources. For our use we have used Kaggle credit card fraud detection.
  - Data Preprocessing - It involves cleaning and organizing of raw data to make it meaningful for the model to feed.
  - Feature Extraction - It's a way of representing data in a condensed form thus reducing data for computation and preserving description of data.
  - This step involves classification of a data point into legal and illegal by use of a trained model.

Contd...



System Architecture

Requirements

REQUIREMENTS	SPECIFICATIONS
jupyter notebook	environment
python 3	programming language
sklearn	library used for machine learning algorithms
matplotlib/Seaborn	used to visual and plot the outputs
imblearn	library for data balancing
pandas/numpy	used for data analysis and manipulation
intel i5/8GB RAM	hardware used for computation work

Methodology

- Aims to answer how we can deal with unbalanced dataset in fraud detection.
- Plan to analyze how ensemblers, boosting and regression along with smote, adasyn, allknn effects unbalanced dataset.
- Metrics used for evaluation include accuracy, recall, precision, f1score, mcc and roc score.

Metrics used

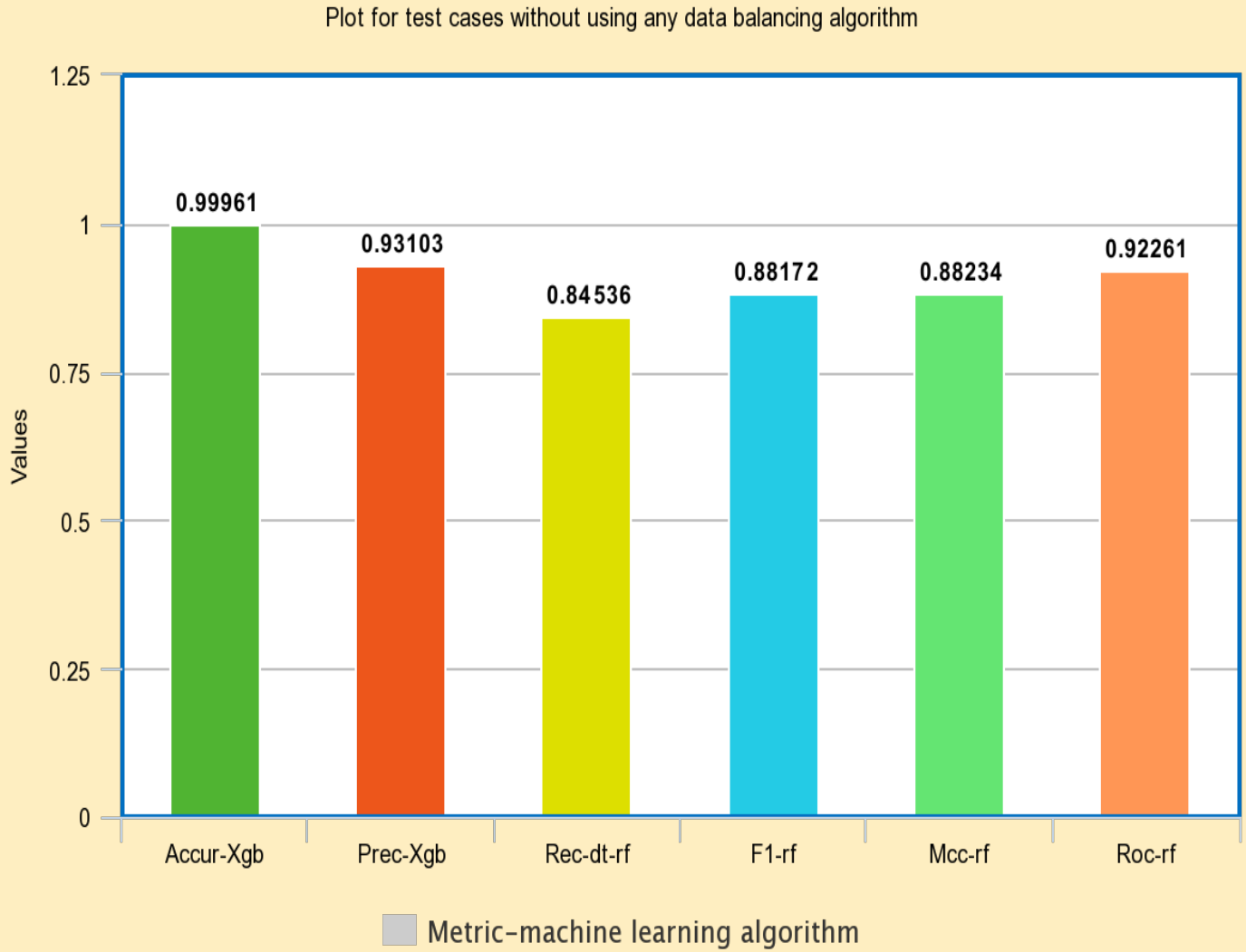
Metrics used for evaluation, (these parameters are used as base parameters for evaluation):

- Accuracy = TP + TN / TP + TN + FP + FN
- Precision = TP / TP + FP
- Recall = TP / TP + FN
- F1score = 2 * precision * recall / precision + recall
- Mcc = TP * TN - FP * FN / sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))
- Roc = Area under curve between false positive rate and true positive rate

- TP - True Positives
- TN - True Negatives
- FN - False Negatives
- FP - False Positives

Results

- Results for test cases without using any machine learning algorithm

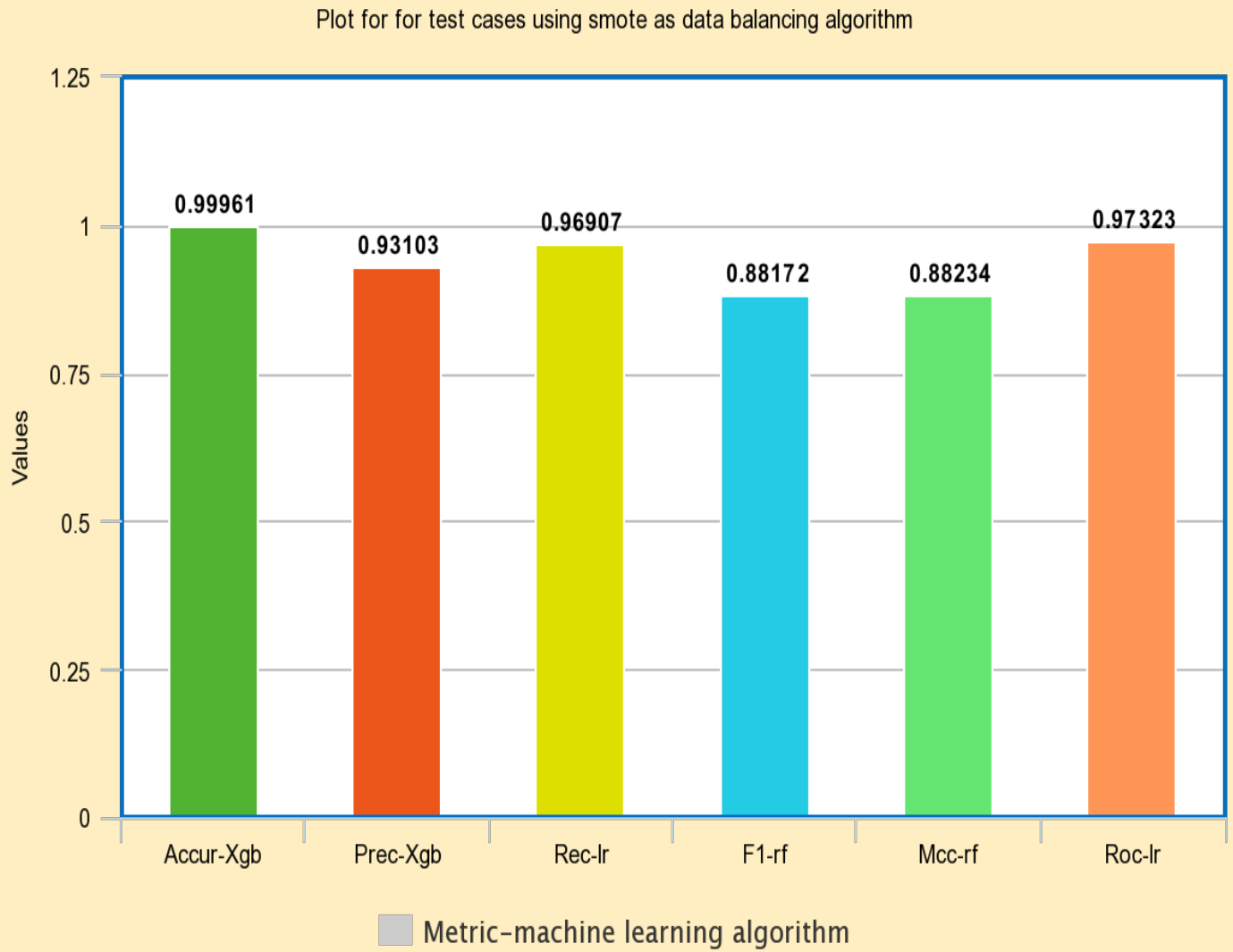


Contd...

- Boosting algorithms generally performed better than bagging in terms of accuracy.
- Random forest and xgboost are found to perform better in terms of precision
- Xgboost and random forest were found to perform best in terms of mcc and logistic regression worst

Contd...

- Results for test cases using smote as data balancing algorithm

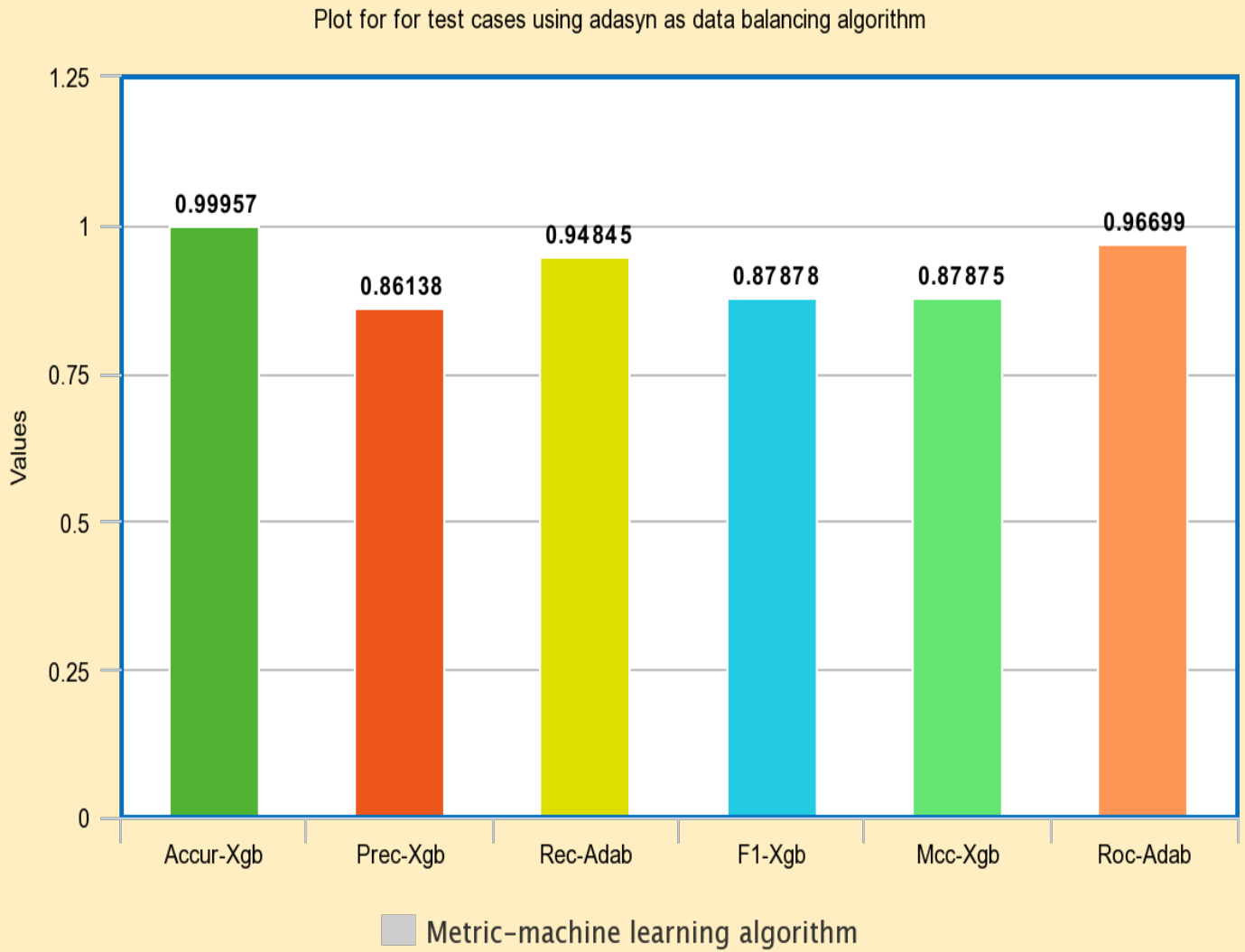


Contd...

- Performance degraded the most in logistic regression with minmaxscaler.
- Random forest and xgboost were found to perform better in terms of accuracy, precision, f1score and mcc.
- Logistic regression registered highest recall and roc value.

Contd...

- Results for test cases using adasyn as data balancing algorithm

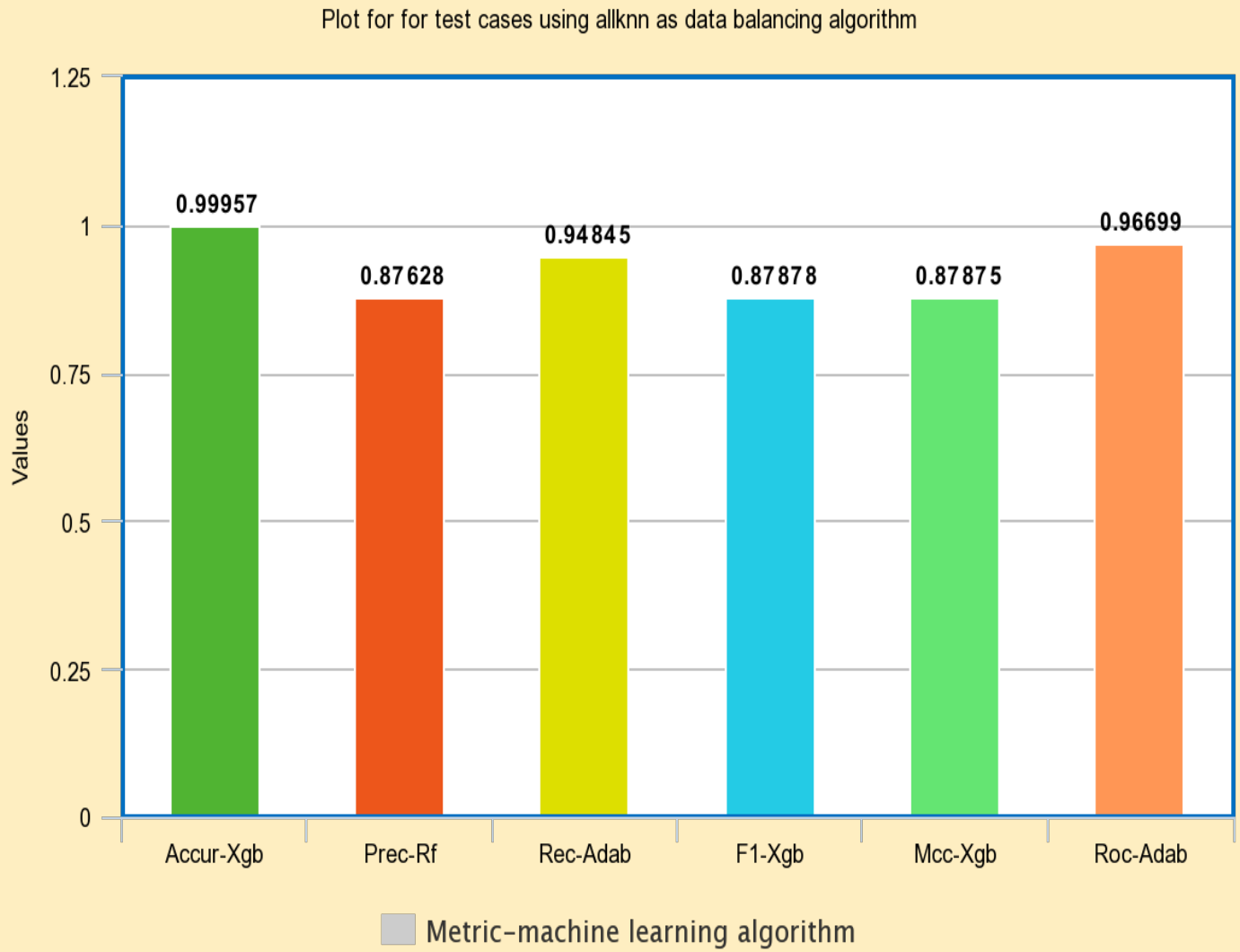


Contd...

- Logistic regression with minmaxscaler performed worst.
- Random forest and xgboost were found to perform better in terms of accuracy, precision, f1score and mcc.
- Adaboost performed dominated precision and roc metrics.

Contd...

- Results for test cases using allknn as data balancing algorithm

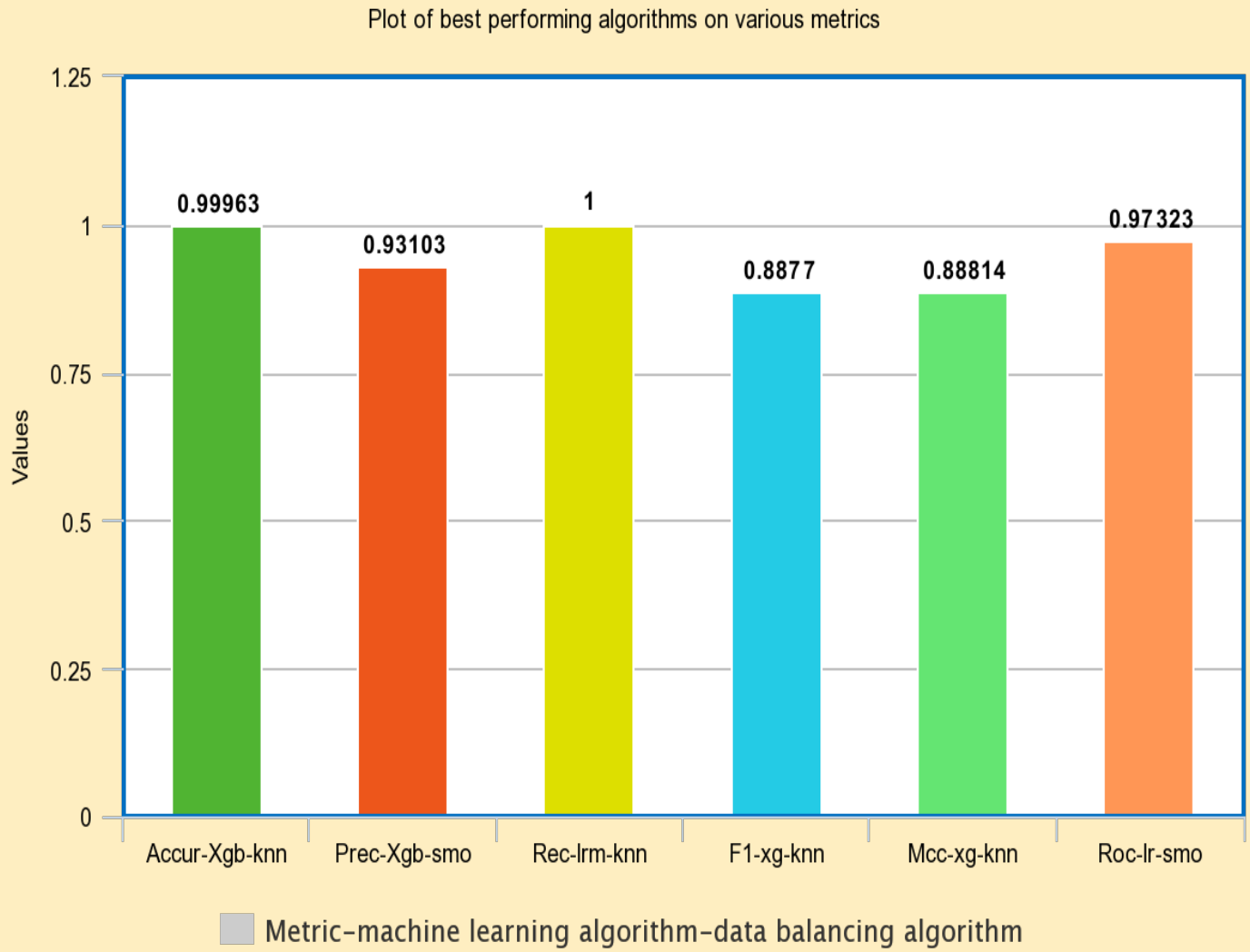


Contd...

- Logistic regression with minmaxscaler performed worst in all metrics except recall where a perfect score of 1 is observed.
- Random forest and xgboost were found to perform better in terms of accuracy, precision, recall, f1score and mcc.
- Decision tree outperformed all other algorithms in terms of roc score.

Conclusion

- Results for all machine learning algorithms on all metrics.



Contd...

- The dataset is highly imbalanced, according to observation 99.83% belongs to class 0 and 0.17% to class 1.
- Highest accuracy was observed for xgboost using allknn
- Highest recall was observed for logistic regression with minmaxscaler using allknn
- Highest precision and f1-score was observed for xgboost using allknn
- Highest roc and mcc score was observed for logistic regression using smote
- True positive was found to be zero for logistic regression with smote and adasyn
- True negative and false negative for logistic regression with minmaxscaler and logistic regression with minmaxscaler along with allknn is found to be 0

Contd...

- 'V14' is found to be the most dominating feature when smote and adasyn are used as data balancing algorithm
- 'V17' is found to be the most dominating feature when no data balancing algorithm is used

Future Work

- Performance improvement can be made by hyperparameter tuning techniques like grid searchCV, randomized searchCV, bayesian optimization.
- Study on comparison between machine learning techniques, neural network, hidden markov model and bayesian belief network could be done to built a better combined model.
- More data balancing techniques can be included like borderline-smote and other hybrid approaches to study the effect on metrics.

References

- R. Patidar, L. Sharma et al., Credit card fraud detection using neural network, International Journal of Soft Computing and Engineering (IJSCE), vol. 1, no. 3238, 2011
- S. Panigrahi, A. Kundu, S. Sural, and A. K. Majumdar, Credit card fraud detection: A fusion approach using dempstershafer theory and bayesian learning, Information Fusion, vol. 10, no. 4, pp. 354363, 2009.
- P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo, Distributed data mining in credit card fraud detection, IEEE Intelligent Systems and Their Applications, vol. 14, no. 6, pp. 6774, 1999.
- J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton,and O. Caelen, Sequence classification for credit-card fraud detection, Expert Systems with Applications, vol. 100, pp. 234245, 2018.

Thank You