Automotive loans default forecast with ML classifiers

Natalja Talikova, 2023. __________

The global automotive finance market, projected to grow from $245.62 billion in 2021 to USD 519.21 billion by 2032(Precedence Research, 2023), is driven by the emergence of online finance applications that enable easy loan comparisons and applications(Fortune Business Insights, 2022). Developing countries present promising target markets as demand saturates in developed nations. With approximately 92% of new cars and a growing number of used cars purchased using finance agreements, concerns about potential defaults are rising due to escalating living costs.

Financial institutions face significant losses from vehicle loan defaults, leading to tightened loan underwriting and higher rejection rates(Automotive Management online magazine, 2022). This situation underscores the necessity for a more accurate credit risk scoring model and warrants an in-depth study to determine the factors contributing to vehicle loan defaults.

The project utilizes data from the Indian Financial Data Science Hackathon 2019(Analytics Vidhya, Mishra, S., Bhavsar, N., 2019), where 470 teams competed, and the dataset has since been downloaded 1,960 times on Kaggle, resulting in seven published projects. The hackathon's AUC-ROC evaluation method recorded a leader's result of 0.67317. Given that a few years have passed since 2019, this project aims to explore advancements that could improve predictions. As a first-time endeavor in Financial Data Science, the primary objective is to gain experience with Data Science tools and techniques while leveraging clean and well-researched data.

The goal is to learn and optimize three classifying algorithms at an intermediate level: Logistic Regression, Support Vector Machine, and the recently popularized GXBoost, which is known for its exceptional performance(Bentéjac, C., Csörgő, A. and Martínez-Muñoz, G., 2021). The project will compare results using the competition's metrics. While Logistic Regression and Support Vector Machines will not be explained in detail, a brief overview of GXBoost will be provided. GXBoost is a scalable ensemble technique based on gradient boosting that has proven to be a dependable and efficient machine learning challenge solver(Rao, C., Liu, Y. and Goh, M., 2022).

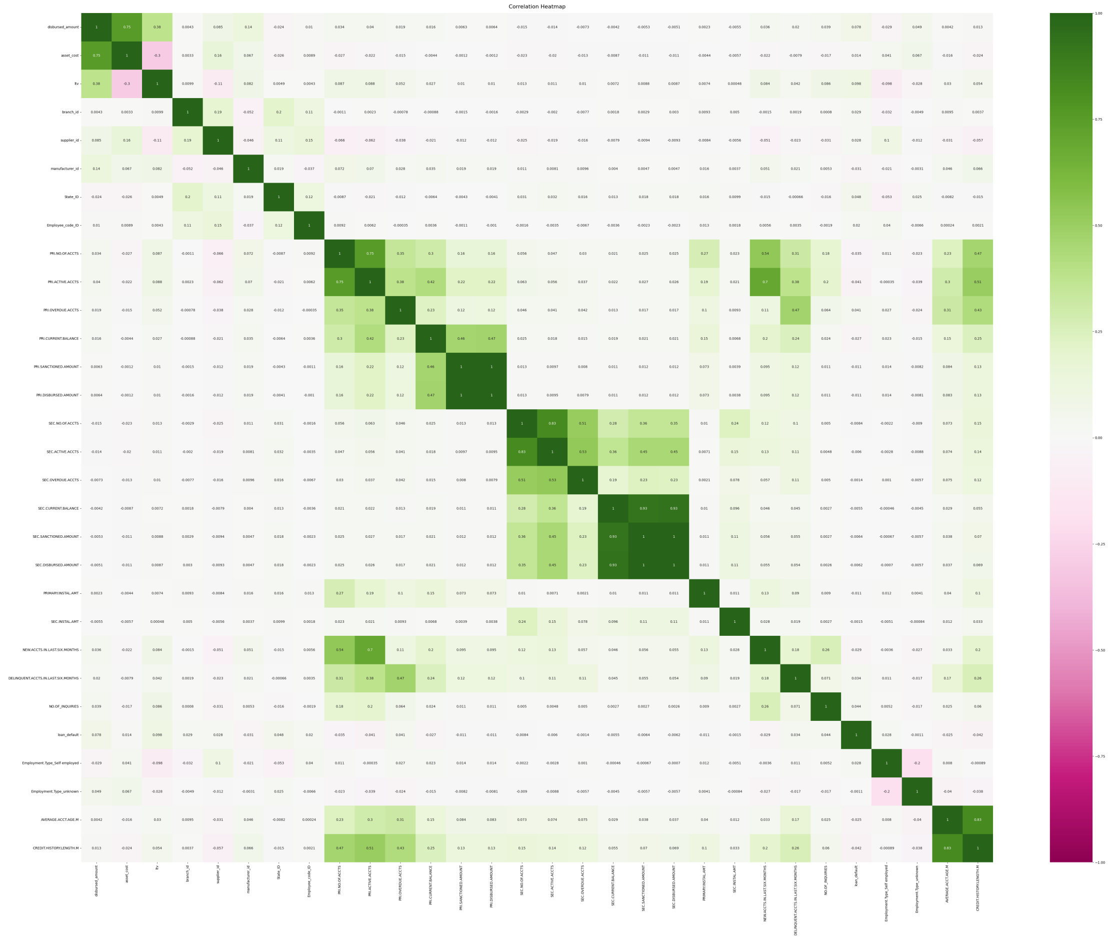
Data:

The full Data Dictionary

The data source for the project comes from the Analytics Vidhya website mentioned above. The original dataset contains 233,155 train and 112,393 test observations, 40 features, and a target vector 'loan_default'. It has one float64 type, five objects, and the rest are int64. Despite the datatype, 15 represent categories that require corresponding treatment before processing. Additionally, numerical data had many outliers. Therefore, 70% of the work involved data transformations, cleaning, and applying feature engineering methods compatible with mixed (numeric and categorical) data.

The data consists of three parts: identification documents of the loan recipients, demographic information, and financial information and risk scores. In the project, only the train dataset from the original data was used, as the purpose was to compare predictions to target variables, using distinct machine learning algoritms. The data was split into 67% train, 18% validation, and 15% test sets for better precision.

Initially, two columns with identifying documents were dropped as they are unique categories that cannot be used for inference. The data was then researched for duplicates and missing values and filled. Afterward, dates and categorical variables were dealt with, converting date features and object variables to appropriate formats. Dummy variables were used for 'Employment.Type' and 'manufacturer_id', dropping one subcategory to set the base. The 'PERFORM_CNS.SCORE.DESCRIPTION' feature was initially transformed into a categorical variable with a specific order subcategories and later on, it was treated as a numerical feature. Correlation heathmap 1 was used to examine mutual correlations between remaining 37 attributes and the loan_default.



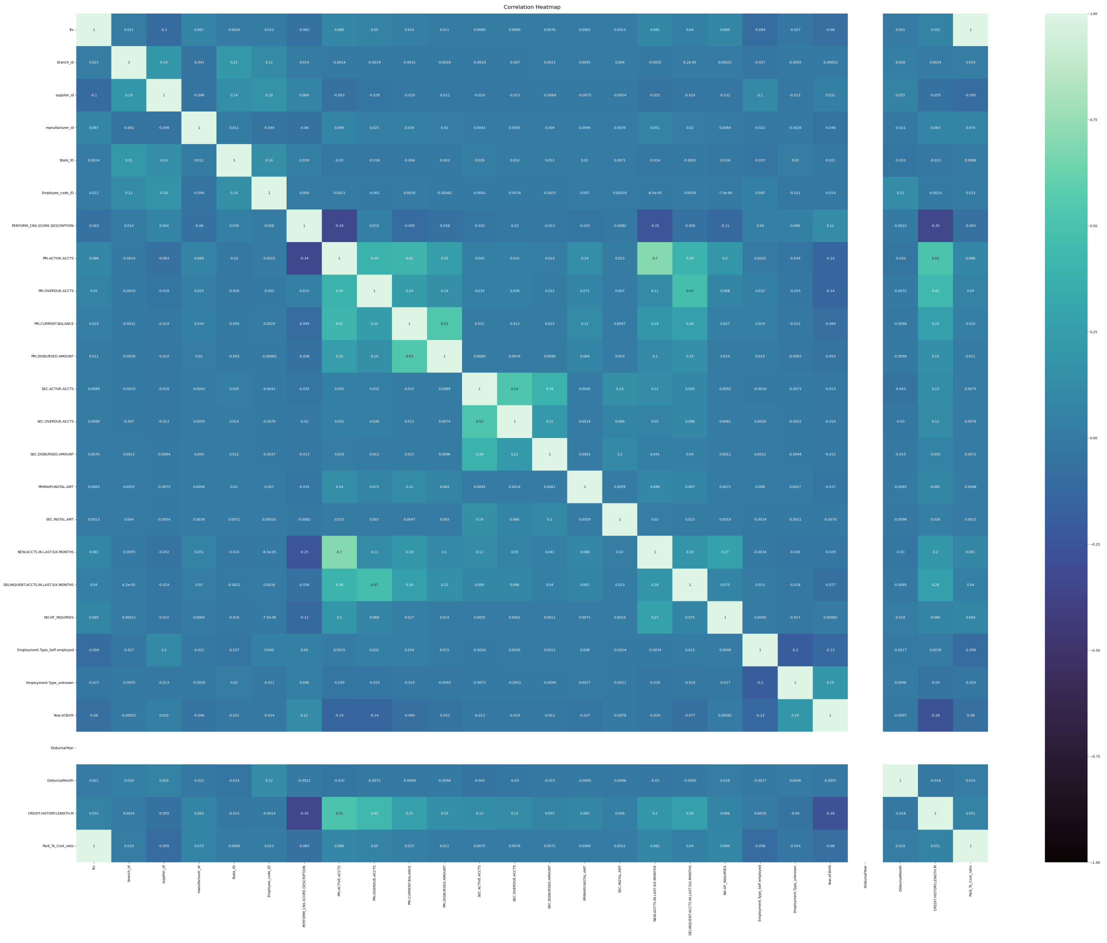
The attributes of ID, including 'branch_id', 'supplier_id', 'State_ID', 'Employee_code_ID', and 'manufacturer_id', were both Label and One-Hot Encoded before being analyzed using the Mutual Information method. This technique quantifies the association between two variables through mutual information, where higher values suggest a stronger dependency. It employs nonparametric approaches and estimates entropy based on knearest neighbors distances. However, the analysis did not yield significant values for these attributes. Although these attributes might be investigated further to ascertain if they can indirectly impact the target variable as factors for other characteristics, they have been omitted since the analysis goes beyond the scope and primary objective of the project.

Additionally, from the Correlation Heatmap 1 follows that pairs of variables

- 'PRI.NO.OF.ACCTS' and '*PRI.ACTIVE.ACCTS*',
- 'PRI.SANCTIONED.AMOUNT' and '*PRI.DISBURSED.AMOUNT*',
- 'SEC.NO.OF.ACCTS' and '*SEC.ACTIVE.ACCTS*',
- 'AVERAGE.ACCT.AGE.M' and '*CREDIT.HISTORY.LENGTH.M*'
- three variables 'SEC.CURRENT.BALANCE', 'SEC.SANCTIONED.AMOUNT' and '*SEC.DISBURSED.AMOUNT*'

with correlation coefficients higher than 0.83 are approximately linearly correlated. To maintain a parsimonious model, each one of the variables(to the left) was from the group without a significant loss of information, while also reducing noise, as only one variable from the group carries new information.

Subsequently, numerical and count variables were addressed. Individual histograms and box plots were created, but they provided limited insight due to the differing distributions and abundance of outliers. Consequently, another correlation map was constructed to identify any missed crucial information, as the number of features had been reduced to 26 at this point.



Following that, interquartile ranges were determined, initially attempting to eliminate all tails above and below ±1.5 IQR. Totally 15 columns had outliers, from 1337 to 51660 each. After examining each feature, dropped 0.23-0.29% of the sample data, totally about 1.7% of observations.

__Column: ltv Count of outliers: 6170 Outliers to Sample Ratio: 0.026463195999210824 Mean: 74.74653001878589 lower_bound: 46.694999999999986 Min: 10.03 upper_bound: 105.85500000000002 Max: 95.0

Column: PRI.ACTIVE.ACCTS Count of outliers: 32534 Outliers to Sample Ratio: 0.13953867400945297 Mean: 1.0398963775015655 lower_bound: -1.5 Min: 0 upper_bound: 2.5 Max: 144

Column: PRI.OVERDUE.ACCTS Count of outliers: 26275 Outliers to Sample Ratio: 0.11269375605822761 Mean: 0.15654889043293274 lower_bound: 0.0 Min: 0 upper_bound: 0.0 Max: 25

Column: PRI.CURRENT.BALANCE Count of outliers: 41044 Outliers to Sample Ratio: 0.17603815503915868 Mean: 165900.07693627386 lower_bound: -52509.75 Min: -6678296 upper_bound: 87516.25 Max: 96524920

Column: PRI.DISBURSED.AMOUNT Count of outliers: 39712 Outliers to Sample Ratio: 0.1703251927910308 Mean: 218065.89865496624 lower_bound: -91200.0 Min: 0 upper_bound: 152000.0 Max: 1000000000

Column: SEC.ACTIVE.ACCTS Count of outliers: 3817 Outliers to Sample Ratio: 0.01637115382965765 Mean: 0.027702720090583905 lower_bound: 0.0 Min: 0 upper_bound: 0.0 Max: 36

Column: SEC.OVERDUE.ACCTS Count of outliers: 1337 Outliers to Sample Ratio: 0.005734407301611811 Mean: 0.007244139066882833 lower_bound: 0.0 Min: 0 upper_bound: 0.0 Max: 8

Column: SEC.DISBURSED.AMOUNT Count of outliers: 3704 Outliers to Sample Ratio: 0.01588649562092008 Mean: 7179.997872650694 lower_bound: 0.0 Min: 0 upper_bound: 0.0 Max: 30000000

Column: PRIMARY.INSTAL.AMT Count of outliers: 38868 Outliers to Sample Ratio: 0.16670526776293781 Mean: 13105.48172023641 lower_bound: -2998.5 Min: 0 upper_bound: 4997.5 Max: 25642806

Column: SEC.INSTAL.AMT Count of outliers: 2217 Outliers to Sample Ratio: 0.009508736714789367 Mean: 323.26844917951223 lower_bound: 0.0 Min: 0 upper_bound: 0.0 Max: 4170901

Column: NEW.ACCTS.IN.LAST.SIX.MONTHS Count of outliers: 51660 Outliers to Sample Ratio: 0.22157029259630973 Mean: 0.38183346629266496 lower_bound: 0.0 Min: 0 upper_bound: 0.0 Max: 35

Column: DELINQUENT.ACCTS.IN.LAST.SIX.MONTHS Count of outliers: 18195 Outliers to Sample Ratio: 0.07803854962814277 Mean: 0.09748063511670398 lower_bound: 0.0 Min: 0 upper_bound: 0.0 Max: 20

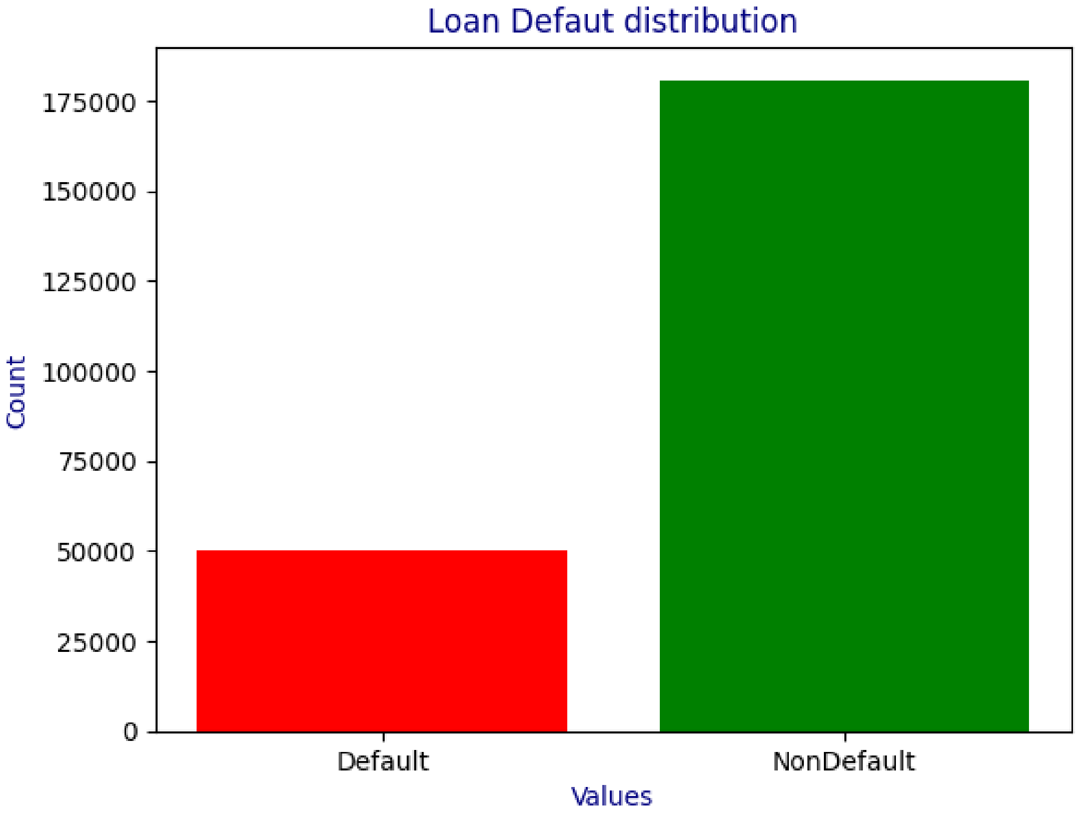
Column: NO.OF_INQUIRIES Count of outliers: 31193 Outliers to Sample Ratio: 0.133787110665054 Mean: 0.20661451229659367 lower_bound: 0.0 Min: 0 upper_bound: 0.0 Max: 36

Column: CREDIT.HISTORY.LENGTH.M Count of outliers: 16056 Outliers to Sample Ratio: 0.06886435574770323 Mean: 16.25240399049555 lower_bound: -36.0 Min: 0 upper_bound: 60.0 Max: 468

Column: Paid_to_cost_diff Count of outliers: 10373 Outliers to Sample Ratio: 0.04448990795783045 Mean: 21508.074615919093 lower_bound: -7194.0 Min: 3997 upper_bound: 46886.0 Max: 638420__

However, during the tuning of the logistic regression, the elimination was limited to four columns of features with non-zero values up to 2% of the observations and two particularly long tails, no more than 0.7% of the observations and it proved to be optimal.

After addressing numerical and count variables, the Standard Scaler was used to eliminate the dominance of one variable over another. The target variable distribution was then examined and found to be imbalanced.

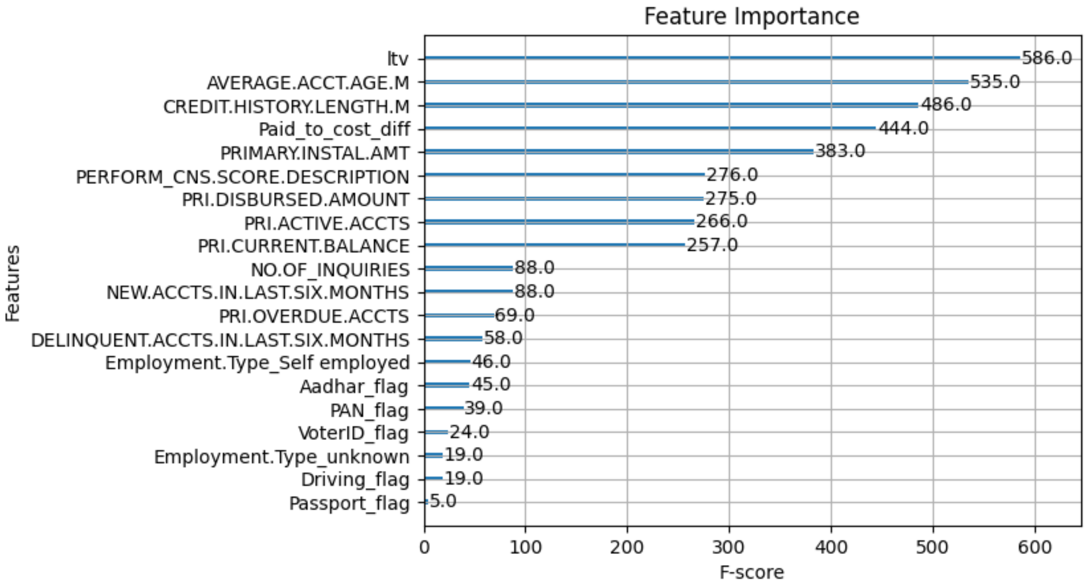


The graph show that the distribution of default and nondefault values is univen. To avoid bias caused by an imbalanced distribution, we can utilize techniques like oversampling the minority class or undersampling the majority class to balance the dataset. This is because Logistic Regression and Support Vector Machine algorithms are not robust to imbalanced distributions. Oversampling involves duplicating samples from the minority class at random to enhance its representation in the dataset. This can be achieved using methods such as random oversampling, Synthetic Minority Over-sampling Technique (SMOTE), or ADASYN(Adaptive Synthetic Sampling).

It is important to note that while SMOTE may have a higher accuracy score on this particular dataset, ADASYN may be more effective in other scenarios with different datasets. It is always important to experiment with different techniques and evaluate their effectiveness on the specific problem at hand. Additionally, it is crucial to assess the impact of data augmentation on the overall performance of the model and avoid overfitting.

As the dataset contains a mixture of categorical and numerical data, the SMOTENC version of the SMOTE method should be used. However, due to technical limitations, we will use SMOTE instead. To apply SMOTE for a mixed dataset, we will first transform all categorical variables to integers. Secondly, we will apply SMOTE, and finally, we will round all the synthetic sample categorical variables to integers.

The feature engineering was completed with the third algorithm, XGBoost tuning, and the feature importance analysis. It was found that the removal of any feature deteriorates the algorithm's performance, indicating optimal engineering. However, it might be worthwhile to perform factor analysis on the dropped Branch ID, Manufacturer ID, or Employee ID, though it is beyond this project's scope.



Methodology

Since a significant portion of the project tasks involved data transformations and feature engineering, the majority of the methodology employed is related to data modification and refinement.

1. Dummy variables inherited from statistics, also known as an indicator variable, is used to represent the different levels of a categorical explanatory variable through assigning parameters. It takes on values of either zero or one and is used to include or exclude the appropriate parameters for each observation.(Dobson, A.J. and Barnett, A.G.,2018)

2. One-Label encoding is a technique that converts categorical variables into numerical values by assigning a unique integer to each category. This method helps to represent categorical data in a format that can be processed by machine learning algorithms. On the other hand, One-Hot encoding creates binary features for each category, where only one feature is marked as '1' (hot) and the rest as '0' (cold) for each instance. This approach avoids the issue of assigning an arbitrary numerical order to categories and allows machine learning models to better understand the categorical data.(Scikit-learn.org.,n.d., LabelEncoder, OneHotEncoder)

3. The Standard Scaler is a preprocessing technique that standardizes dataset features to have zero mean and unit variance. This equalizes feature contribution and prevents dominance due to scale differences. It calculates each feature's mean and standard deviation, then scales the data accordingly. This process benefits algorithms sensitive to input scale, such as Support Vector Machines and Linear Regression.(Scikit-learn., n.d. StandardScaler.)

4. SMOTE description: This method addresses imbalanced datasets by combining over-sampling of the minority class and under-sampling of the majority class, resulting in enhanced classifier performance in ROC space. Imbalanced datasets often contain a small proportion of "abnormal" examples, making misclassification costly. By creating synthetic minority class examples, the approach outperforms alternatives like adjusting loss ratios in Ripper or class priors in Naive Bayes. Evaluated using AUC and the ROC convex hull strategy, the method proves effective with C4.5, Ripper, and Naive Bayes classifiers.(Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002)

5. Linear regression classifier: "Logistic Regression (also called Logit Regression) is commonly used to estimate the probability that an instance belongs to a particular class. If the estimated probability is greater than 50%, then the model predicts that the instance belongs to that class (called the positive class, labeled '1'), or else it predicts that it does not (i.e., it belongs to the negative class, labeled '0'). This makes it a binary classifier." (Géron, A., 2019, p. 141)

6. Support vector machine: "A Support Vector Machine (SVM) is a powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification, regression, and even outlier detection. It is one of the most popular models in Machine Learning, and anyone interested in Machine Learning should have it in their toolbox. SVMs are particularly well suited for classification of complex but small- or medium-sized datasets." (Géron, A., 2019, p. 155)

7. XGBoost: XGBoost, an acronym for "Extreme Gradient Boosting," has emerged as a leading machine learning library in the ML community due to its unparalleled speed

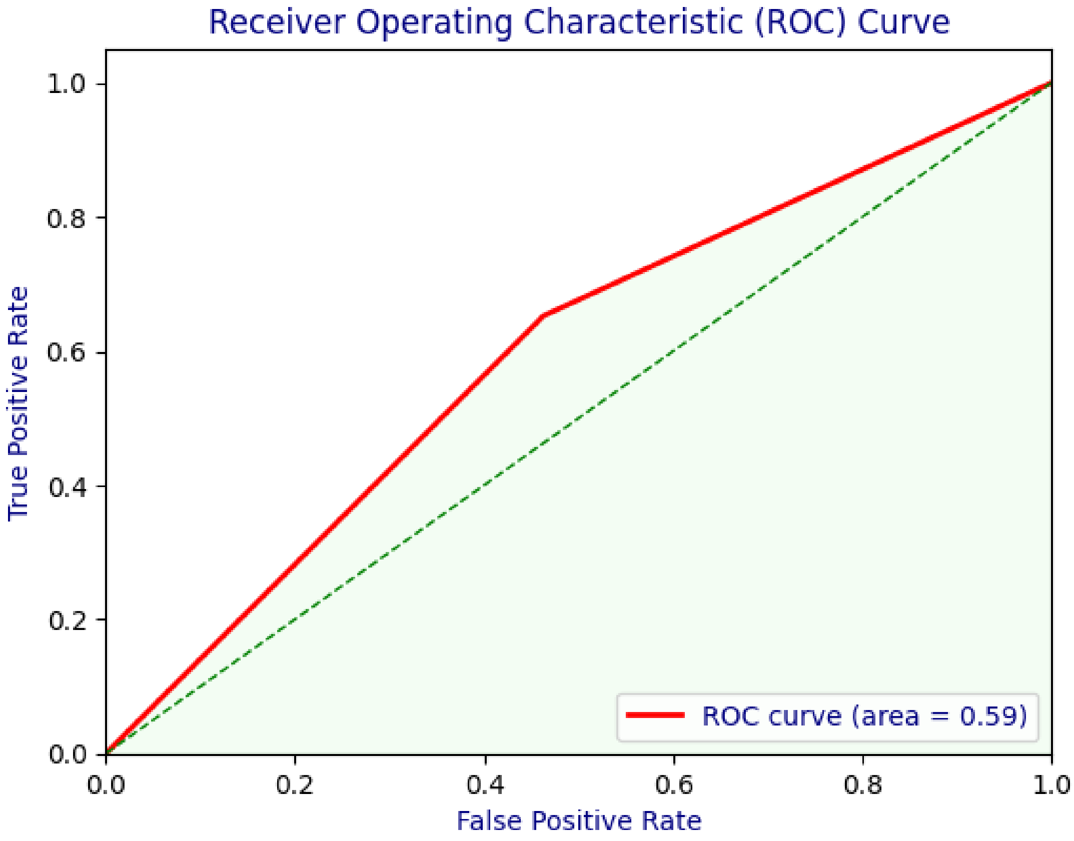
and performance. This optimized distributed gradient boosting library is designed for efficient and scalable training of machine learning models. It has become popular and widely used in a variety of applications, such as Kaggle competitions, recommendation systems, and click-through rate prediction. The library has APIs in multiple languages, including Python, R, and Julia, making it accessible to a wide range of users.

The key features of XGBoost include its efficient handling of missing values and built-in support for parallel processing. This allows the library to handle real-world data with missing values without requiring significant pre-processing and train models on large datasets in a reasonable amount of time. XGBoost is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. It is particularly effective in classification and regression tasks and can be customized for fine-tuning various model parameters to optimize performance.(Rao, C., Liu, Y. and Goh, M., 2022)

Results

Logistic Regression.

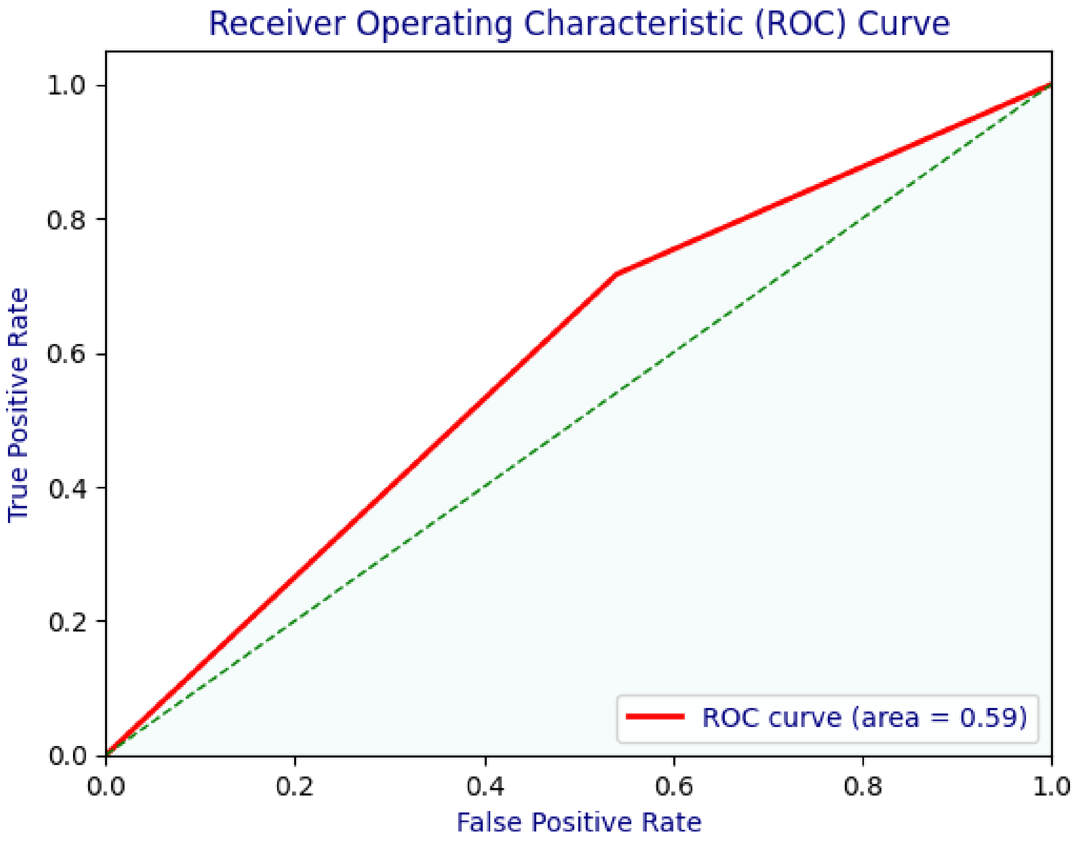
Accuracy of logistic regression model 0.5953338251567687 F1 Score 0.6171456490254585 Recall Score 0.6541392320781239 Balanced Accuracy Score 0.5954982189964665 [[14595 12591] [9350 17684]] AUC score: 0.5954982189964665



Regularised logistic regression yielded the best results at C = 0.007, with an AUC score of 0.5945125946367751. This score was slightly lower than that obtained using ordinary logistic regression (detailed results are available in the working notebook).

Support Vector Machines

Accuracy of the Support Vector Machine: 0.5879380302471413 F1 Score: 0.6343251824936986 Recall Score: 0.7168010653251461 Balanced Accuracy Score: 0.5882982741471607 AUC score for SVM: 0.5882982741471607



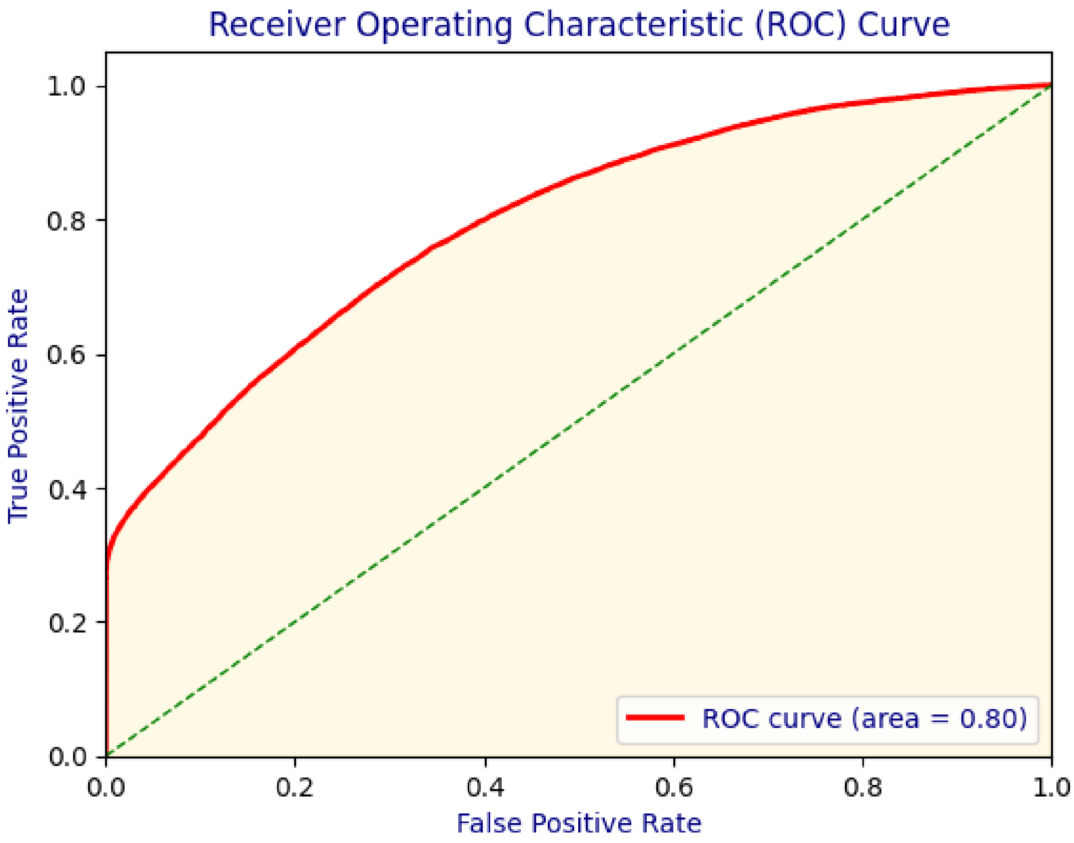
At first, the SVM with Radial Basic Function Kernel and default parameters yielded the most favorable metrics.

- Precision: 0.6004022899582238
- Recall: 0.7003320820098181
- F1 Score 0.6465285992768955

Nevertheless, the optimization of hyperparameters was not fruitful due to temporary and technical constraints.

XGBoost

Accuracy of XG for test data 0.7075986720767244 F1 Score 0.7097187637322396 Recall Score 0.7169120366945329 Balanced Accuracy Score 0.7076247081140581 AUC score for xgBoo for test data: 0.7971739089887604



In this project, three machine learning algorithms—Logistic Regression, Support Vector Machine, and GXBoost—were evaluated for their effectiveness in predicting vehicle loan defaults. GXBoost emerged as the clear winner, achieving an AUC score of 0.7960272504199912, which surpasses the 2019 DataScience FinHack winners by a significant 13% margin. Logistic Regression was primarily used to ensure appropriate model fitting due to its fast processing and ease of tuning; however, its performance was limited to 59% with the given feature setup. AUC-ROC graphs indicated that a linear model might not be the best fit for this dataset, suggesting the need to explore more flexible models.

Conclusion:

Despite the success of the GXBoost model, there is potential for further improvement. Incorporating ID features as factors and applying logarithmic transformations to numeric features before standardization could enhance the model's performance. Additionally, using cloud services such as AWS can help overcome hardware limitations and enable the use of Grid Search for hyperparameter tuning with SVM. Future work could also involve applying this powerful combination of algorithms to purely numerical datasets to refine tuning techniques and optimize computational efficiency.

References

(1) Precedence Research. (2023). *Automotive Finance Market (By Provider Type: Banks, OEMs, Others; By Finance Type: Direct, Indirect; By Purpose Type: Loan, Leasing, Others; By Vehicle Type: Commercial Vehicles, Passenger Vehicles) - Global Industry Analysis, Size,*

Share, Growth, Trends, Regional Outlook, and Forecast 2023-2032. [online] Available at: https://www.precedenceresearch.com/automotive-finance-market .

(2) Fortune Business Insights. (2022). *The global automotive finance market is projected to grow from $245.62 billion in 2021 to$385.42 billion in 2028 at a CAGR of 6.5% in forecast period, 2021-2028*. [online] Available at: https://www.fortunebusinessinsights.com/industry-reports/automotive-finance-market-100122 .

(3) Automotive Management online magazine. (2022). *UK car finance debt soars to £40bn*. [online] 2 Nov. Available at: https://www.am-online.com/news/finance/2022/11/02/uk-car-finance-debt-soars-to-40bn .

(4) Analytics Vidhya. (2019). *LTFS Data Science FinHack (ML Hackathon)*, 13-19 Apr. [online] Available at: https://datahack.analyticsvidhya.com/contest/ltfs-datascience-finhack-an-online-hackathon/#About .

(5) Mishra, S. (2019). *LTFS-Loan-Default-Prediction*. [online] GitHub. Available at: https://github.com/sauravmishra1710/LTFS-Loan-Default-Prediction .

(6) Bhavsar, N. (2019). *ltfs-vehicle-loan-default-prediction*. [online] GitHub. Available at: https://github.com/NishantBhavsar/ltfs-vehicle-loan-default-prediction

(7) Bentéjac, C., Csörgő, A. and Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54, pp.1937-1967. [online] Available at: https://doi.org/10.1007/s10462-020-09896-5

(8) Rao, C., Liu, Y. and Goh, M. (2022). Credit risk assessment mechanism of personal auto loan based on PSO-XGBoost Model. *Complex & Intelligent Systems*. https://doi.org/10.1007/s40747-022-00854-y

(9)Dobson, A.J. and Barnett, A.G. (2018). An Introduction to Generalized Linear Models. 4th ed. Boca Raton: CRC Press, Chapman & Hall. [online] Available at: https://doi.org/10.1201/9781315182780 [Accessed 26 Apr. 2023].

(10)Scikit-learn.org. (n.d.). LabelEncoder. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html Scikit-learn.org. (n.d.). OneHotEncoder. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

(11)Scikit-learn. (n.d.). StandardScaler. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

(12)Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16(1), pp.321-357. [online] Available at: https://doi.org/10.1613/jair.953 [Accessed 26 Apr. 2023].

(13)Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd Edition).

O'Reilly Media.

Project is completed with assistance of ChatGPT3.5 and 4: code debugging, grammar and syntaxis, text formatting.