Assignment4&5 Vehicle Loan Default

Meifang Li(2719570), Yuhao Qian(2684098)

1. Introduction

Financial institutions are exposed to credit default risk. The default of vehicle loans could lead to significant losses. As a result, these institutions might tighten up the underwriting of loans and increase loan rejections. Larsen&Toubro Financial Services(LTFS) is a non-banking financial company providing vehicle loan services. LTFS warrants a better credit risk scoring model, and it presented a "DataScience FinHack" in 2019. In this assignment, we participate in this former Hackathon and apply Logistic Regression(LR) and Random Forest(RF) to decision making on a sanctioning loan for a particular applicant.

2. Explanatory Data Analysis

The data set we use is from "DataScience FinHack" on the Kaggle page[1].

2.1. *General information*

The raw training data set has 233154 entries and 41 columns involved with float64(1)[2], int64(34), object(6). This data set is imbalanced as the positive instances(default) only accounts for 21.7% of the whole set. The missing values only exist in the *Employment.Type* column with 7661 Nan entries. We assume that missing employment type is useful so we construct a new class *Unknown_employ* to fill the Nans. Some columns have improper data format for machine learning, thus we need to transform these features before analysis. We calculate the *Age* according to *Date.of.Birth* as a numeric variable. Besides, we transform *AVERAGE.ACCT.AGE* and *CREDIT.HISTORY.LENGTH* into exact number of months for convenience. The features provided in the data set can be divided into three categories: Loanee information, Loan information and Bureau& history data.

We first compute the correlation between all the variables. Figure 1 shows that there is a relatively strong correlated relationship between *disbursed_amount* and *asset_cost*, *PERFORM_CNS.SCORE* and *PERFORM_CNS.SCORE.DESCRIPTION*, *PRI.ACTIVE.ACCTS* and *PRI.NO.OF.ACCTS*, *NEW.ACCTS.IN.LAST.SIX.MONTHS* and *PRI.ACTIVE.ACCTS*, *SEC.NO.OF.ACCTS* and *SEC.ACTIVE.ACCTS*, *SEC.CURRENT.BALANCE* and *SEC.SANCTIONED.AMOUNT* and *SEC.DISBURSED.AMOUNT* since their correlation values are all above 0.7.

2.2. *Loanee information*

Loanee information describes the features related to the loan borrower. Since we are very interested in the *PERFORM_CNS.SCORE.DESCRIPTION* describing the credit level of the loan borrower, and it is not straightforward to read the description, we decide to encode this categorical feature in this stage. We categorize the original description into six groups based on the risk level. Detail is in the Table 1.

As shown in Figure 2, self-employed is the largest group in both default and non-default situations, followed by the salaried group. Figure 3 reveals the loanee's CNS score description related to credit risk after encoding. We can see that the majority of this feature is 0, which means not rated for some reasons. The second largest group is the borrowers with low risk. Although this plot tells the imbalance of the data, we can not recognize a clear pattern indicating a specific group having a larger default possibility. And Figure 4 shows that most loan borrowers age from 20 to 60 but people under 40 are the majority.

[1] LoanDefault_LTFS_AV(ML_FINHACK):https://www.kaggle.com/lampubhutia/loandefault-ltfs-avml-finhack?select=train_LTFS.csv

[2] The number in the bracket is the number of corresponding features.

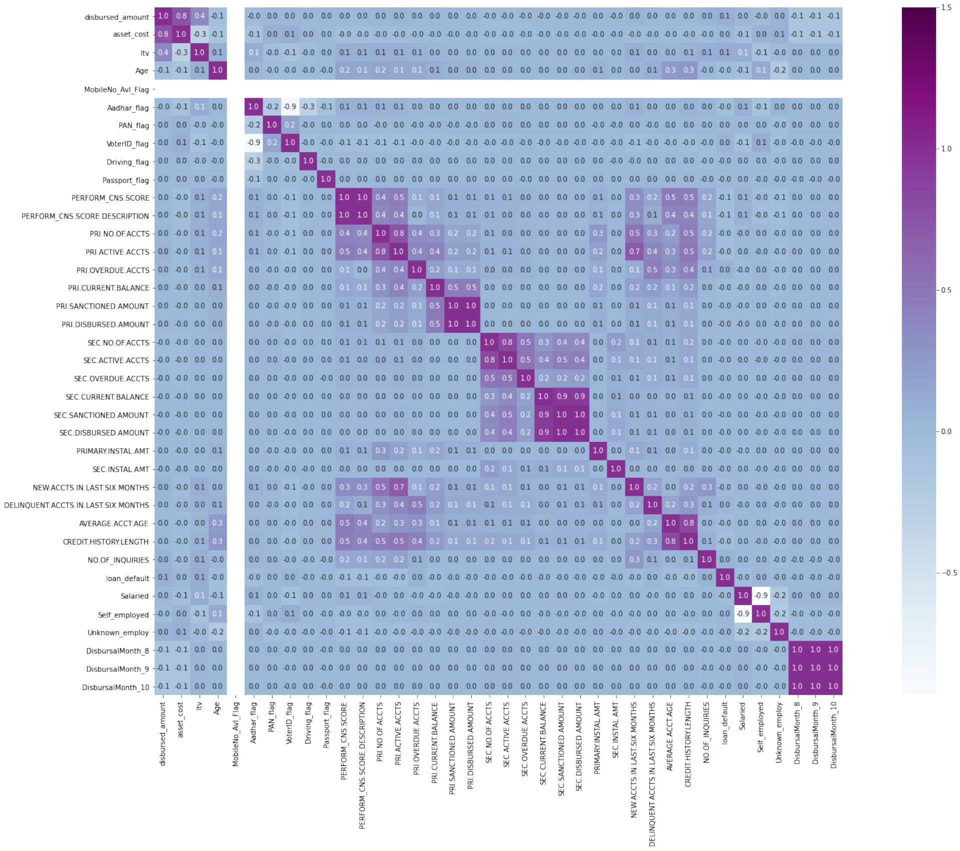


Fig. 1: The correlation matrix

Table 1: Detail of *PERFORM_CNS.SCORE.DESCRIPTION* encoding

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

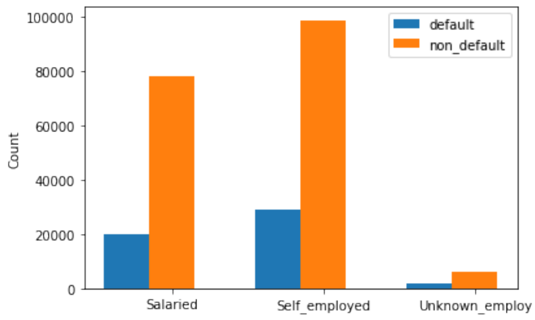


Fig. 2: The situation of employment

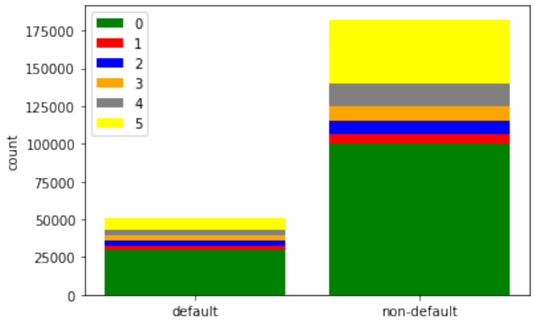


Fig. 3: The situation of performance score description

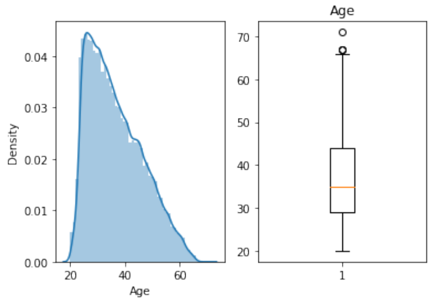


Fig. 4: Loanee Information-Age

2.3. *Loan information*

Figure 5 shows the distribution of the features related to loan information. It is clear that *disbursed_amount* and *asset_cost* are more skewed than *ltv*, and *ltv* is mainly clustered near 70 to 80.

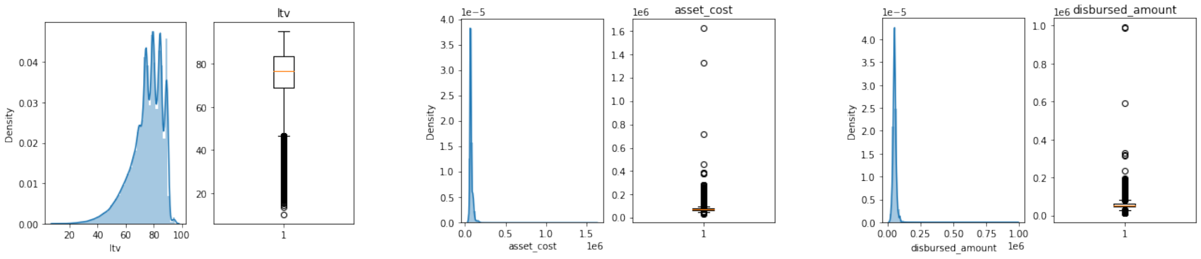


Fig. 5: Loan information

2.4. *Bureau& history data*

This category contains information from the Bureau and history data. As shown in Figure 6, we can see that all the other numeric features are highly skewed. There are extremely large values in the data set. We notice that there are some points that deviate from others immensely in *disbursed_amount*, *asset_cost*, *PRI.SANCTIONED.AMOUNT*, *PRI.DISBURSED.AMOUNT*.

Besides numeric features, we also have some identity features: *branch_id,supplier_id, manufacturer_id, State_id, employee_code_id*. The numbers of the unique ids are in Table 2:

Table 2: The numbers of the unique ids

ID	branch_id	supplier_id	manufacturer_id	State_id	employee_code_id
count	82	2953	11	22	3270

3. Preprocessing

3.1. *Feature engineering*

3.1.1. Merging feature

Since *disbursed_amount* and *asset_cost* are strong correlated as mentioned above and may have problem with collinearity in Figure 7 , we create a new feature *Downpayment* to represent the difference between these two features. We also drop *PERFORM_CNS.SCORE* since *PERFORM_CNS.SCORE.DESCRIPTION* after encoding has duplicated information for the credit score. Besides, we merge the information of the primary and secondary accounts and generate the total features:

```
['Total.NO.OF.ACCTS','Total.Active.ACCTS',
'Total.Overdue.ACCTS', 'Total.CurrentBalance','Total.SanctionedAmount',
'Total.DisbursedAmount','Total.InstalAmount']
```

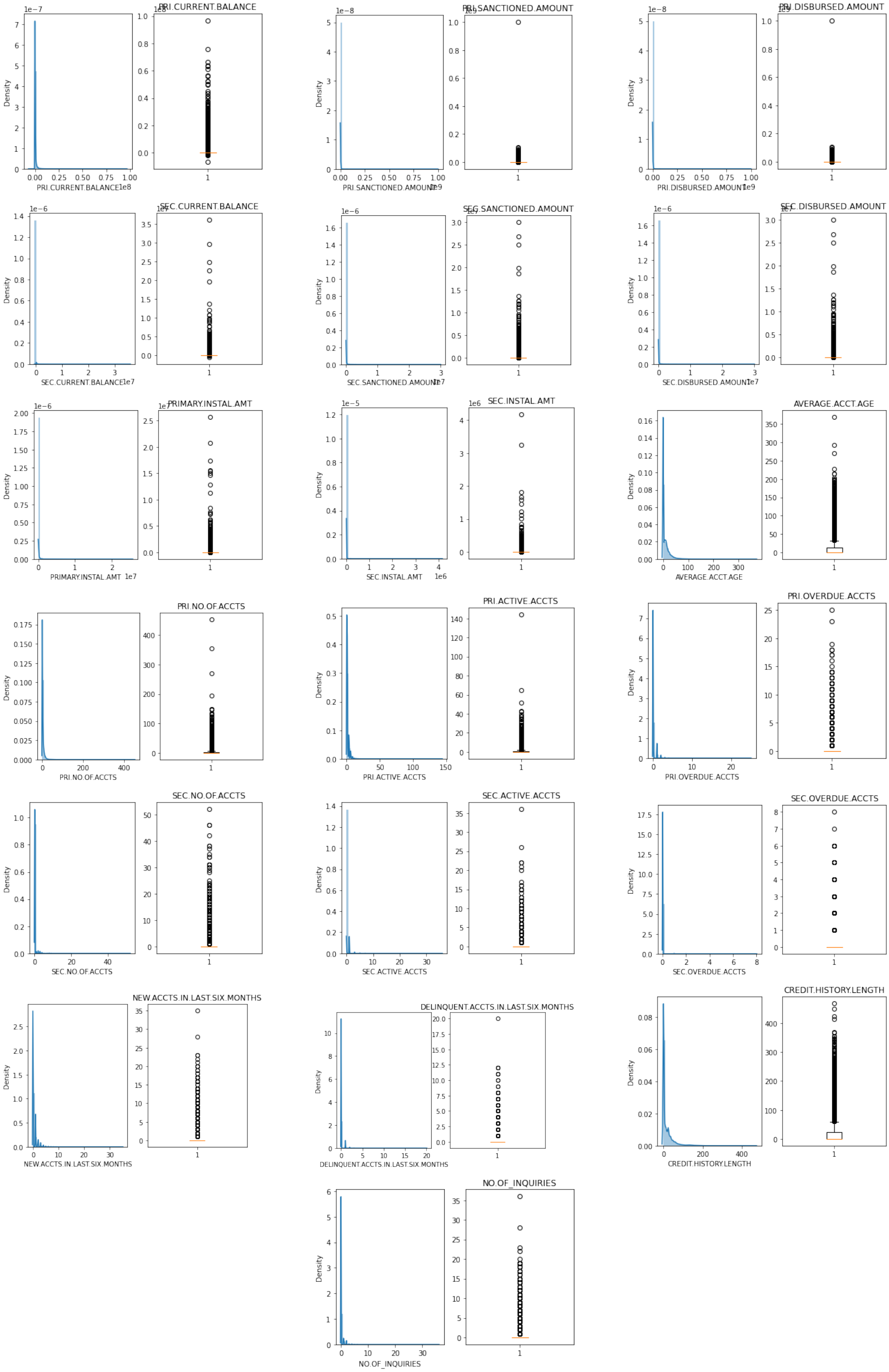


Fig. 6: Bureau data & history data

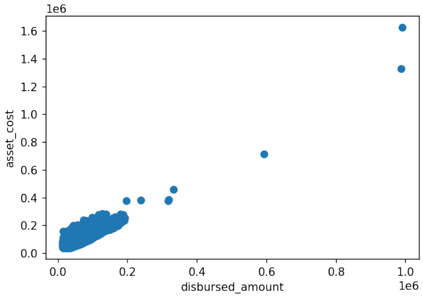


Fig. 7: The scatter plot for disbursed amount and asset cost.

3.1.2. Changing variable distribution

Strictly speaking, almost all the numeric features have many outliers, but we can only consider these cases as extreme values; otherwise, we would remove too much data. However, we still remove the ”outliers” that deviate significantly. We drop the maximum value of *PRI.NO.OF.ACCTS*, *PRI.SANCTIONED.AMOUNT*, *PRI.DISBURSED.AMOUNT*, *DELINQUENT.ACCTS.IN.LAST.SIX.MONTHS* and the two maximum values of *Downpayment* and *SEC.INSTAL.AMT*. To mitigate the effect of extreme values, we apply logarithmic transformation($\log(1+x)$) to the following numeric features:

```
[’ltv’,’Total.NO.OF.ACCTS’,’Total.Active.ACCTS’,
’Total.Overdue.ACCTS’, ’Total.CurrentBalance’,’Total.SanctionedAmount’,
’Total.DisbursedAmount’,’Total.InstalAmount’,
’NEW.ACCTS.IN.LAST.SIX.MONTHS’, ’DELINQUENT.ACCTS.IN.LAST.SIX.MONTHS’,
’AVERAGE.ACCT.AGE’, ’CREDIT.HISTORY.LENGTH’, ’NO.OF_INQUIRIES’,’Downpayment’]
```

When taking logarithmic, we notice that *PRI.CURRENT.BALANCE* and *SEC.CURRENT.BALANCE* have negative values; hence, we transform the original *PRI.CURRENT.BALANCE* and *SEC.CURRENT.BALANCE* by adding the absolute value of the minimum value in the whole data series before applying the log transformation.

3.1.3. Downsampling or oversampling

We still need to deal with the imbalance in the training set. We have two options: downsampling and oversampling. In downsampling, we keep all the positive instances(default) while randomly select the same amount of positive instances in the negative instances(nondefault). As for oversampling, we use SMOTE introduced in the *imblearn* python package[3].

3.1.4. Encoding

Apart from the *PERFORM_CNS.SCORE.DESCRIPTION* which we has already encoded before, we also implement one hot encoding to *DisbursalMonth* and *Employment.Type*. It is worth to be mentioned that we ignore identity features(*branch_id*, *supplier_id*, *manufacturer_id*, *Current_pincode_ID*, *State_ID* and *Employee_code_ID*) when trying for Logistic Regression model but we apply one hot encoding to these features when trying for Random Forest model.

3.2. Feature selection

Although we have a large number of candidate variables, not all of them are useful in the LR model. Thus, we implement stepwise feature selection to reduce the number of explanatory variables. Stepwise regression is a method that iteratively examines the statistical significance of each independent variable in a regression model [1]. There are three types of stepwise regression: forward, backward and hybrid. We apply the hybrid stepwise regression in LR, which means we use forward selection, but we check the chosen variables using backward selection after each step. Unfortunately, there is no available stepwise feature selection function in Python, and we have to create a custom program [2]. We start with nothing and add the explanatory variables with the most significant p-value which is smaller than 0.2; then we double-check the chosen variables, eliminating the insignificant variables.

After stepwise feature selection, there are eighteen explanatory variables left:

[3] imbalanced-learn documentation:https://imbalanced-learn.org/stable/

```
[ltv, MobileNo_Avl_Flag, PERFORM_CNS.SCORE.DESCRIPTION, VoterID_flag,
NO.OF_INQUIRIES, Total.Overdue.ACCTS, Total.ACCTS, Age, Self_employed,
Total.CurrentBalance, Total.InstalAmount, DELINQUENT.ACCTS.IN.LAST.SIX.MONTHS,
AVERAGE.ACCT.AGE, CREDIT.HISTORY.LENGTH, Downpayment,
Passport_flag, Aadhar_flag, Driving_flag]
```

4. Model building

4.1. *Logistic Regression*

Logistic regression(LR) is a classic model for binary classification. It has few hyper-parameters and rapid training speed, indicating it is an intuitive model to start with. Furthermore, as a linear model, LR can use linear correlations between features and dependent variables. Considering their non-linear correlations to the dependent variable, we exclude identity features concerning ID. Since the default data set is highly imbalanced, we decide to compare downsampling and oversampling using MOTE. When applying SMOTE, we only oversample the training set. We can not oversample the validation set, otherwise we would get unseen observations and fake result [3]. Using the selected features through the above-mentioned stepwise selection, we build the LR model using the default parameter in *sklearn*.

4.2. *Random Forest*

Random forest (RF) works well on missing data, continuous and discrete features, also can capture the complex and non-linear interactions between them. Random forest tries to exploit information gain of features rather than linear correlations, presenting another perspective of this task.

To incorporate identity and categorical features, we transform some of them into one-hot encoding [4] according to their contribution to the model. If we encode all the identity features, we would have ten thousand features and a very sparse matrix, which causes an extremely long training time. We finally decide to encode *manufacturer_id* and *state_id*, and withdraw all the other ID features. Although the features are sparse, according to the construction procedure of sub-trees, uninformative features will be neglected in the final structure.

Compared to LR, RF has more parameters to tune but still can be searched in an acceptable time if we choose several critical parameters. We are interested in the following parameters: n_estimators, max_features, max_depth, min_samples_split, min_samples_leaf, bootstrap. We use a random grid search to exploit parameter combinations producing the best classification accuracy on the validation set. The parameter configurations are shown in Table 3.

Table 3: Parameters to tune. Text in bold are default values in *sklearn*

Parameter	Description	Values
n_estimators	The number of trees in the forest.	[**100**, 200... 1000]
max_features	The number of features for the best split.	**auto**, sqrt
max_depth	The maximum depth of the tree.	[10, 20,...,110, **None**]
min_sample_split	The minimum samples to split an internal node.	[**2**,5,10]
min_samples_leaf	The minimum number to be a leaf node.	[**1**,2,4]
bootstrap	Bootstrap samples are used when building trees.	**True**, False

According to the result of the grid search, we choose a parameter combination with the highest AUROC score, which is n_estimators= 600, min_sample_split=10, min_samples_leaf=2, max_depth=20, auto max_features and bootstrap.

5. Evaluation

We mainly use the confusion matrix, ROC curve and SHAP value to evaluate the two models in this assignment.

5.1. *Confusion matrix*

Accuracy is the simplest metric for the classification model, but it does not consider imbalance classes or differing costs of false negatives and false positives. Thus, we opt for a better metric: F1 score. We first introduce precision and recall which are defined as follows:

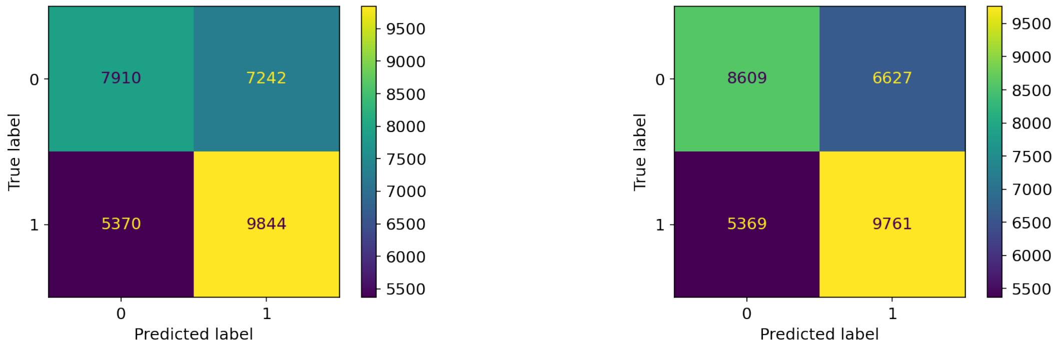
$$precision = \frac{TP}{TP+FP}, recall = \frac{TP}{TP+FN} \tag{1}$$

The F1 score is the geometric mean of precision and recall:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{2}$$

Since the main goal is detecting loan default behaviour, we should focus on identifying positive cases. Extremely large precision or recall is unacceptable; hence, we warrant an optimal blend of precision and recall using F1 score.

The confusion matrix helps calculate the precision and recall. A confusion matrix for binary classification has four outcomes: true positive, false positive, true negative, and false negative. In this task, we prefer to have more true positive cases and fewer false negative cases. Figure 8 shows confusion matrices for LR and RF respectively. As we can see, both plots have more true positive cases(bottom right) than false positive cases(bottom left), which means we can detect loan default correctly.



(a) Logistic Regression

(b) Random Forest

Fig. 8: The confusion matrix. Left is for LR and right is for RF. Both confusion matrices are obtained from balanced training set using downsampling.

Table 4 gives classification report of LR and RF. We can see that besides recall for default class, where both models have 0.65, RF outperforms LR in all the other metrics. RF has higher precision for default than that of LR, which means in this case, RF predicts fewer false positive cases. Consequently, RF has a higher F1 score than LR.

Table 4: The classification report for downsampled LR and RF

Model	Class	Precision	Recall	F1 score
LR	0	0.60	0.52	0.56
	1	0.58	0.65	0.61
RF	0	0.62	0.57	0.59
	1	0.60	0.65	0.62

We also compute the precision, recall and F1 score for LR and RF using oversampling with SMOTE. The result is in Table 5. We notice that LR has a significant decrease in precision of default class compared with metrics in the previous table. This means there are increasing false positive cases, which is acceptable. Intriguingly, we find that both precision and recall for RF reduce in half. A possible reason is that we only oversample the training set while the validation set is still imbalanced, and RF can not handle the imbalance well.

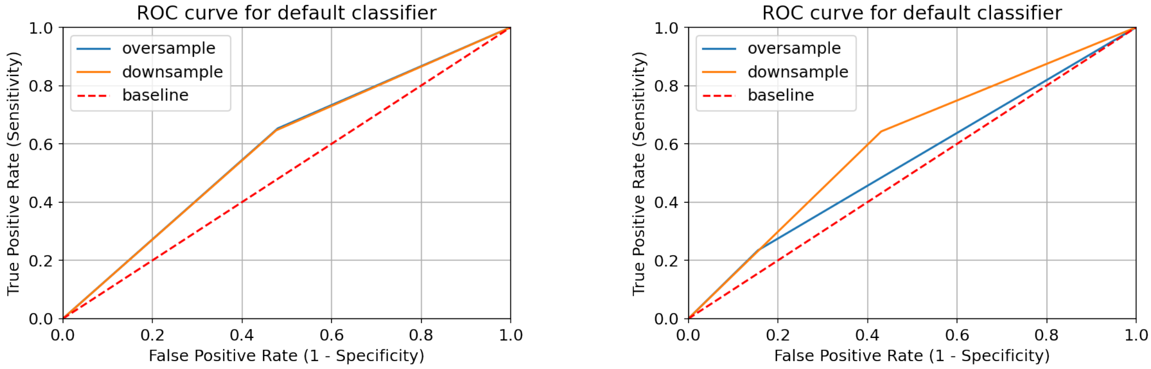
Table 5: The classification report for oversampled LR and RF

Model	Class	Precision	Recall	F1 score
LR	0	0.85	0.52	0.54
	1	0.27	0.65	0.38
RF	0	0.80	0.84	0.82
	1	0.29	0.23	0.26

5.2. *ROC*

ROC curve is a graphical plot used to show the diagnostic ability of a binary classifier. The curve shows the trade-off between sensitivity(true positive rate) and specificity(1- sensitivity). A model that has a curve closer to the top left indicates better performance. Hereby, we compare the effects of downsampling and oversampling using SMOTE in both models. In Figure 9a, the two methods dealing with imbalanced data have similar ROC curves while using LR. However, in Figure 9b, RF has a better ROC curve when using downsampling. All four scenarios outperform the baseline ROC curve, which is the red hash line with an AUROC score of 0.5.

Figure 10 compares the ROC curve between the training set and the validation set in RF. Again, it is clear that the training set has a significantly better ROC curve than the validation set, while the curve for the training set is not very close to the perfect curve(top left). This plot indicates that the RF, in this case, does not have a problem with overfitting.



(a) Logistic Regression

(b) Random Forest

Fig. 9: The ROC curve. The left plot is for LR using downsampling and oversampling, in which the AUROC score is 0.584 and 0.586. The right plot is for RF, yellow line is obtained from downsampling while blue line is from oversampling by SMOTE. The AUROC score using downsampling is 0.605, while using SMOTE is 0.540. The red dash line in both plot is the baseline ROC curve, having AUROC score 0.5.

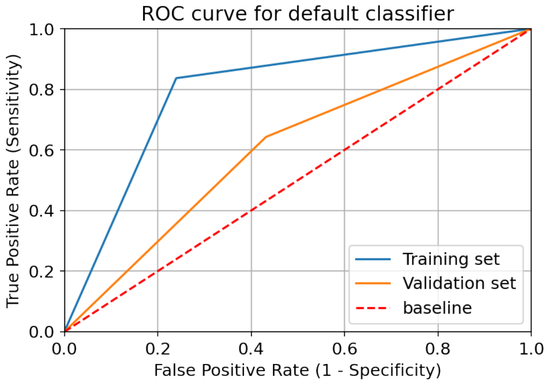


Fig. 10: ROC curve for training set and validation set in RF

5.3. *Residuals Analysis*

After applying the Logistic Regression model mentioned above, we assess the hypothesis of this model by plotting the deviance residuals. Figure 11 show that most residuals are on the scale of [-2, 2] and the residuals significantly deviate from the normal distribution in the QQ-plot. This is mainly because the data for training is so skewed and far from normal distribution even after log transformation that the deviance residuals cannot be guaranteed to have distributions close to normal. However, we might not conclude that our LR model is not useful under this situation.

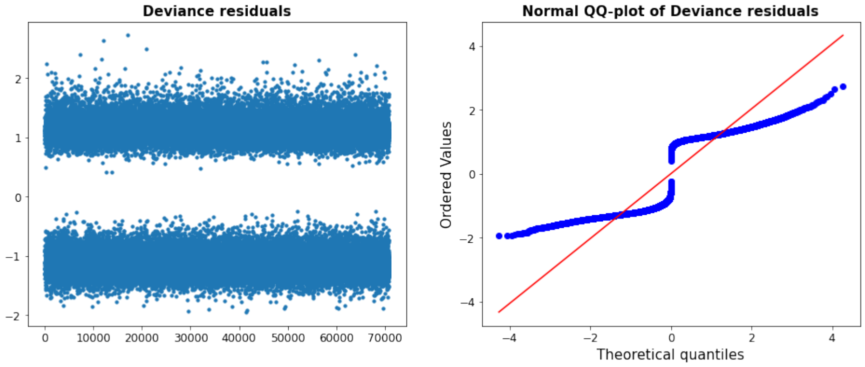
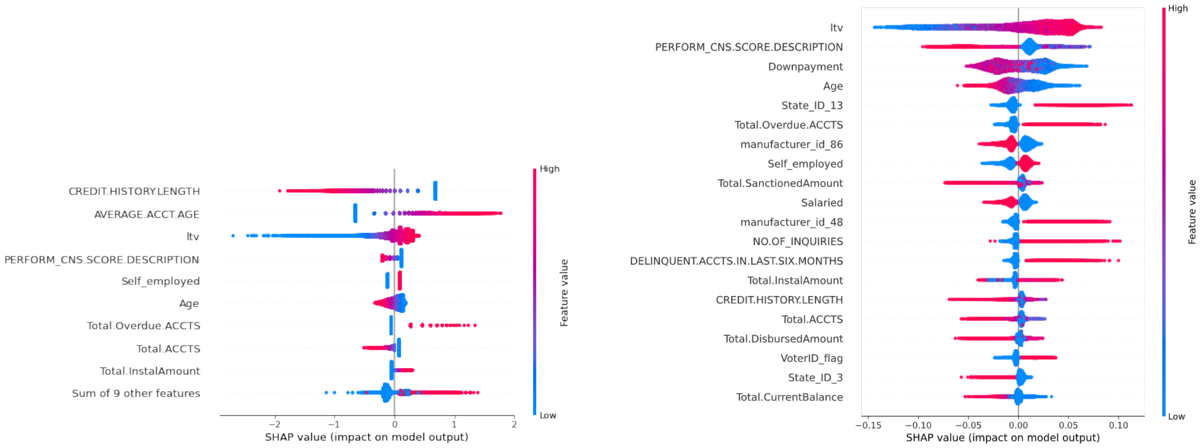


Fig. 11: The deviance residuals plot and QQ-plot of Logistic Regression model

5.4. SHAP

When it comes to feature importance, SHAP [5] value is a more informative choice than the classic feature importance in *sklearn*. We compute the SHAP value for each observation in LR and RF using the SHAP python package[4] and plot a summary plot as shown in Figure 12. Based on this plot, we can see the top 10 or 20 features and their indications of the relationship between the value of a feature and the impact on the classification model.



(a) Logistic Regression

(b) Random Forest

Fig. 12: The SHAP values for the important features in LR and RF. The y-axis is the features ordered by their importance and the x-axis is separate SHAP values. The color represents the value of the feature from low to high. Overlapping points are clustered in the y-axis direction.

Figure 12a reveals that the top three influential features of LR model are *CREDIT.HISTORY.LENGTH*, *AVERAGE.ACCT.AGE* and *ltv*. The *CREDIT.HISTORY.LENGTH* has a high range of values and it has a negative impact on the default situation, which means that as time since first loan increases, the probability of default decreases. In contrast, *AVERAGE.ACCT.AGE* has a positive influence, which means that it is more likely to default when the average loan tenure is larger. It is also worthwhile to note that *ltv* has a positive impact on the default situation. If the Loan to Value of the asset is larger, the risk of default will be higher.

Figure 12b shows that the top four influential features of RF model are *ltv*, *PERFORM_CNS.SCORE.DESCRIPTION*, *Downpayment* and *Age*. Also according to Figure 13, we can say that *ltv* has more than 15% of the RF model's explainability and these top 20 features provide more than 80% of the RF model's interpretation. Furthermore, Figure 14 reveals that the *ltv* has a positive impact on the default situation which corresponds with the analysis of LR model. The loanee tends to default when the loan to value ratio is larger than 4.25(after log transformation). And *PERFORM_CNS.SCORE.DESCRIPTION* has a negative influence since the lower score represents the higher risk, resulting in a higher chance of default. People in the

[4] https://shap-lrjball.readthedocs.io/en/docs_update/index.html

first four categories(0-3) are more likely to default. We can also tell from *Downpayment* that loan borrowers having downpayment lower than 9.5(after log transformation) are more willing to default. In other words, a larger down-payment would have a lower chance of default. Interestingly, we find that the larger of age, the lower tendency of default, which might due to the unstable financial situation of people under 35.

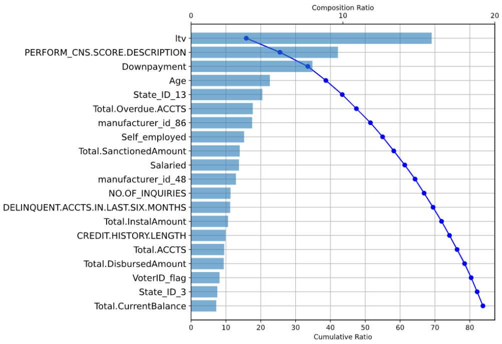


Fig. 13: The SHAP waterfall plot for RF model

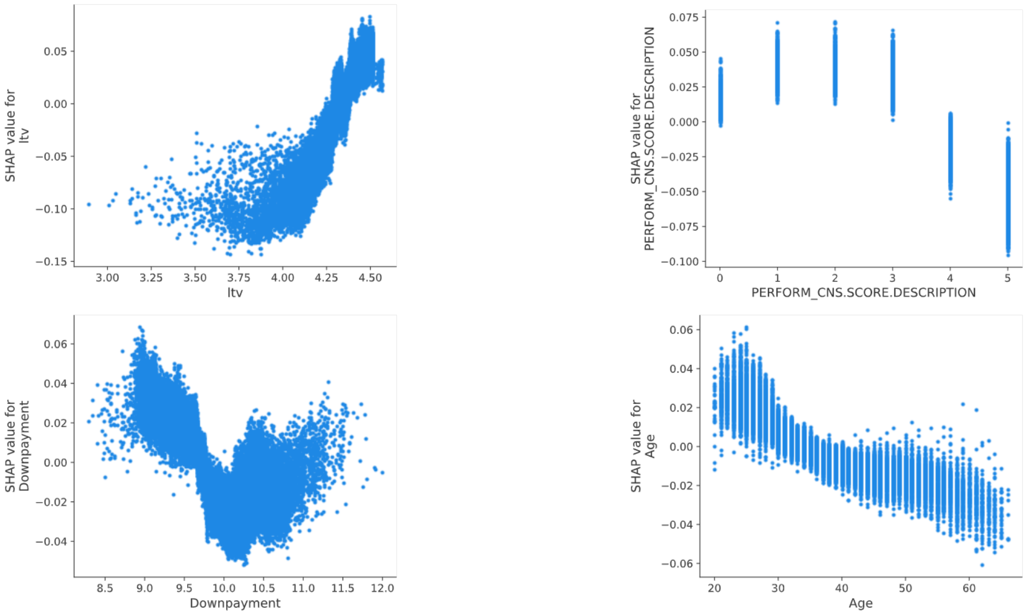


Fig. 14: The SHAP dependence plots for top four features of RF model

5.5. *Odds ratio in LR*

LR is a more interpretable model than RF since we can directly have the estimated coefficients for each variable. We show the coefficients and the subsequent odds ratios of the three most important features of LR in Table 6. The odds ratio is calculated by: e^{β_i}, where β_i is the coefficient of the i-th feature. The positive coefficients or odds ratio larger than 1 indicates that *AVERAGE.ACCT.LENGTH* and *ltv* have positive relation to default. To be specific, if we increase the value of “average loan tenure” and “loan to value ratio” by one unit, the estimated odds change by a factor of 1.3578 and 8.4470 respectively, indicating a larger default possibility. However, *CREDIT.HISTORY.LENGTH* has a negative relation to default, which means the one unit increase in the length would lead to the corresponding odds change by a factor of 0.7614, causing a lower default possibility.

6. Discussion

As mentioned in section 5.2, although the two methods dealing with imbalance have close performance in LR, downsampling has a better ROC curve than oversampling with SMOTE in RF. A possible explanation could be that LR is more robust than RF when predicting the loan default using the imbalanced validation set. Since we can

Table 6: The coefficients and odds ratios for the three most important features in LR.

Feature	Coefficient	Odds ratio
Credit History Length	-0.2726	0.7614
Average Acct Age	0.3059	1.3578
ltv	2.1338	8.4470

not oversample the validation set, we prefer downsampling the negative instances in the training set. According to the F1 score and ROC curve, we can conclude that RF generally performs better in detecting loan default in this task. This is because we encode more identity features in RF and it can reveal the non-linear relation between features and label, whereas LR can only handle linear relation. Besides, RF can automatically perform feature selection during building decision trees, while in LR, we apply stepwise feature selection manually. Finally, it has to be mentioned that LR is a more intuitive and efficient model as RF takes more training time. Therefore, we need to balance between performance and training time.

7. Conclusion

In this assignment, we implement LR and RF to detect vehicle loan default. We get the optimal result using RF with the AUROC score of 0.605 and the F1 score of 0.62. According to the SHAP value of each feature in RF, the three most important features are *ltv, PERFORM_CNS.SCORE.DESCRIPTION, Downpayment*.

References

1. Adam Hayes, “Stepwise regression,” 2021. https://www.investopedia.com/terms/s/stepwise-regression.asp.
2. Aakkash Bijayakumar, “Feature selection with python,” 2021. https://www.datasklr.com/ols-least-squares-regression/variable-selection.
3. Nick Becker, “The right way to oversample in predictive modeling,” 2019. https://beckernick\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{g\global\mathchardef\accent@spacefactor\spacefactor}\let\begingroup\endgroup\relax\let\ignorespaces\relax\accent95g\egroup\spacefactor\accent@spacefactorithubio/oversampling-modeling/.
4. Jason Brownlee, “Ordinal and one-hot encodings for categorical data,” 2020. https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/.
5. S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 4765–4774, Curran Associates, Inc., 2017.