Name: Eagle Xuhui Ying Email: yingx22@wfu.edu Date: 12/09/2022 Section: C

MODEL REPORT

Detailed Analysis & Steps

Problem Statement

Suppose I have recently joined a major financial institution and am tasked with developing and comparing several machine learning models to predict "loan status", Specifically, which loans are likely to default. My task is to build, tune and evaluate several different models, predicting (loan_default), identify any outliers and explain them, explain my top 10 correct predictions of a loan default (loan default = 1), top 10 predictions of loan default = 0, and my top 10 incorrect predictions. I have been provided with two datasets. One is loan_train.csv, which is used to train and evaluate my model. Another is loan_holdout.csv, which is used to assess the accuracy of my prediction.

I conducted data transformation and exploratory data analysis (EDA) to predict loan status, remove useless variables, and derive new ones. Also, I used an isolation forest for anomaly detection and removed five top anomalous records. Then, I used logistic regression, neural network, random forest, and XGBoost to predict the loan status and evaluate each model. My best model is the XGBoost model, and the highest AUC is 0.98, which means the model is a good classifier in this prediction problem. Finally, I further used global and local explanations to evaluate the variables' importance of my best model and used loan_holdout.csv to make my prediction.

File(s) Summary

Here are the summary of two csv files.

File Name	Record count	Column count	Numeric columns	Character columns
loan_train.csv	29777	52	36	16
loan_holdout.csv	12761	51	35	16

Data dictionary

LoanStatNew	Description
acc_now_delinq	The number of accounts on which the borrower is now delinquent.
addr_state	The state provided by the borrower in the loan application
annual_inc	The self-reported annual income provided by the borrower during registration.

application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
chargeoff_within_12_mths	Number of charge-offs within 12 months
collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
delinq_amnt	The past-due amount owed for the accounts on which the borrower is now delinquent.
desc	Loan description provided by the borrower
dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.
earliest_cr_line	The month the borrower's earliest reported credit line was opened
emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
emp_title	The job title supplied by the Borrower when applying for the loan.*
fico_range_high	The upper boundary range the borrower's FICO at loan origination belongs to.
fico_range_low	The lower boundary range the borrower's FICO at loan origination belongs to.
funded_amnt	The total amount committed to that loan at that point in time.
funded_amnt_inv	The total amount committed by investors for that loan at that point in time.
grade	LC assigned loan grade
home_ownership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.
id	A unique LC assigned ID for the loan listing.
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
installment	The monthly payment owed by the borrower if the loan originates.
int_rate	Interest Rate on the loan

issue_d	The month which the loan was funded
last_credit_pull_d	The most recent month LC pulled credit for this loan
last_pymnt_amnt	Last total payment amount received
last_pymnt_d	Last month payment was received
loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
loan_status	Current status of the loan
member_id	A unique LC assigned Id for the borrower member.
mths_since_last_delinq	The number of months since the borrower's last delinquency.
mths_since_last_record	The number of months since the last public record.
next_pymnt_d	Next scheduled payment date
open_acc	The number of open credit lines in the borrower's credit file.
out_prncp	Remaining outstanding principal for total amount funded
out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors
policy_code	publicly available policy_code=1 new products not publicly available policy_code=2
pub_rec	Number of derogatory public records
pub_rec_bankruptcies	Number of public record bankruptcies
purpose	A category provided by the borrower for the loan request.
pymnt_plan	Indicates if a payment plan has been put in place for the loan
revol_bal	Total credit revolving balance
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
sub_grade	LC assigned loan subgrade
tax_liens	Number of tax liens
term	The number of payments on the loan. Values are in months and can be either 36 or 60.

title	The loan title provided by the borrower
total_acc	The total number of credit lines currently in the borrower's credit file
total_rec_late_fee	Late fees received to date
url	URL for the LC page with listing data.
verification_status	Indicates if income was verified by LC, not verified, or if the income source was verified
zip_code	The first 3 numbers of the zip code provided by the borrower in the loan application.

Data Quality & Transformation

(1) ***Remove unique 'id' columns:*** id, member_id

(2) ***Use 'lubridate' package to transform months to the time difference between the event happend to today (transform it into years):*** issue_d_year, earliest_cr_line_year, last_pymnt_d_year, last_credit_pull_d_year

(3) ***Transform percentages to decimals:*** int_rate, revol_util

Remove categorical variables with too many categories / containing too many texts: emp_title, url, title, zip_code,

(4) ***drop variables with only one category / imbalanced variables:*** pymnt_plan, collections_12_mths_ex_med, policy_code, application_type, acc_now_delinq, chargeoff_within_12_mths, delinq_amnt, tax_liens

(5) **Drop columns with more than 20% missing values:** desc, next_pymnt_d, mths_since_last_delinq, mths_since_last_record

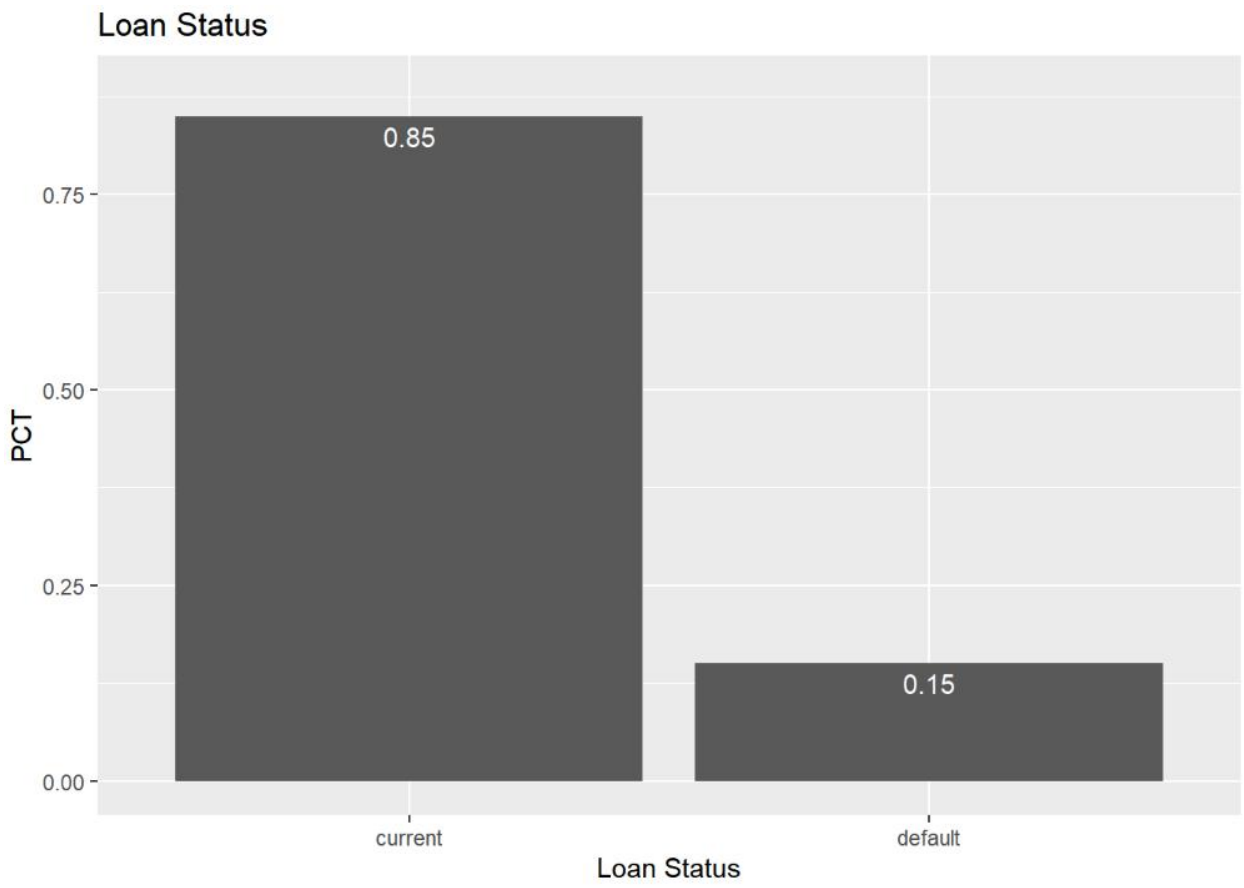
(5) ***Transform all categorical variables into factors***

(6) ***Make a recipe, specify a formula, impute mean/mode to numerical/categorical missing values, normalize the numeric variables, and dummy encode nominal predictors***

Target Summary

loan_status	n	pct
<chr>	<int>	<dbl>
current	25300	0.8496491
default	4477	0.1503509

2 rows

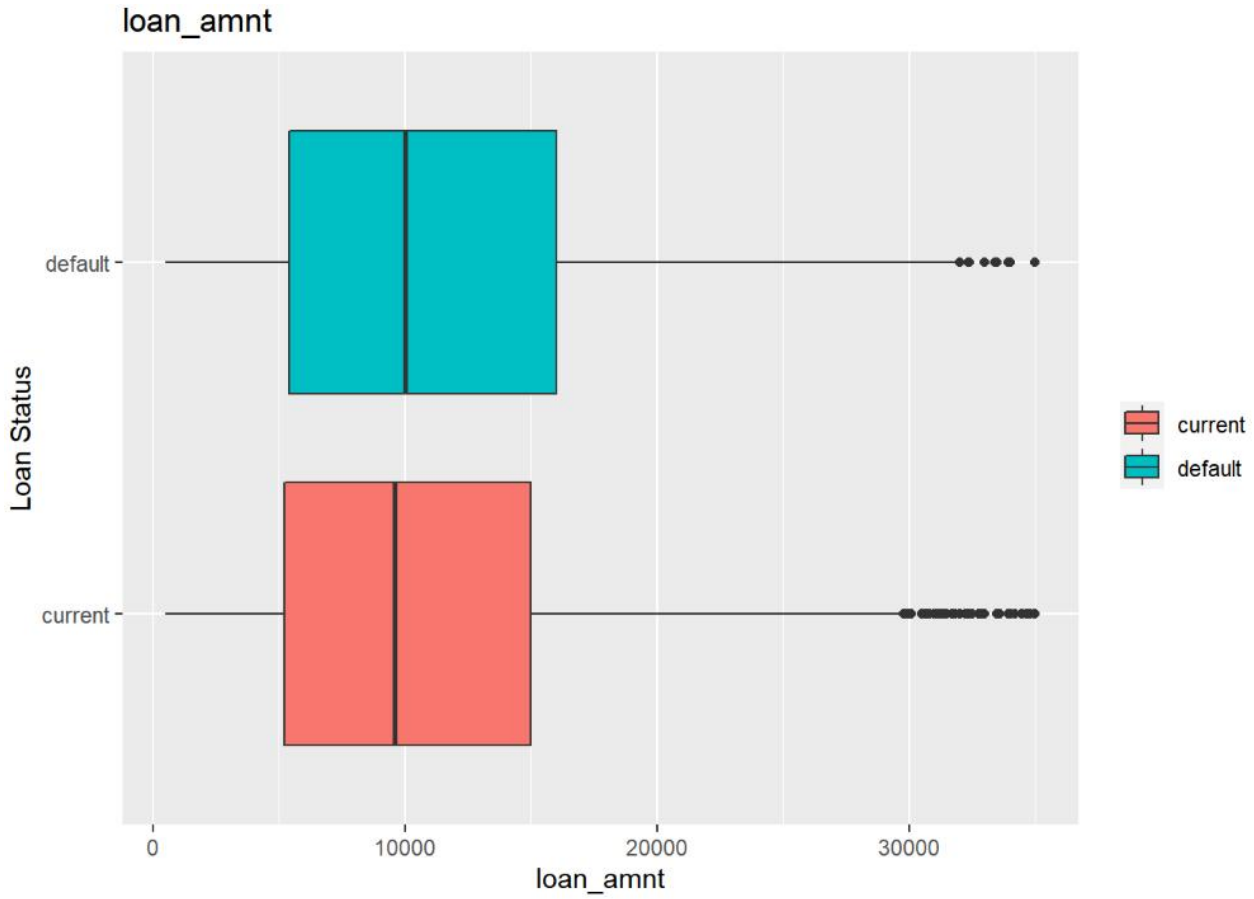


Since the task is a supervised modeling task, we should have an analysis of the target variable. I calculated the frequency and pct for two different loan statuses in this classification task. In this case, the loan default rate is 85%.

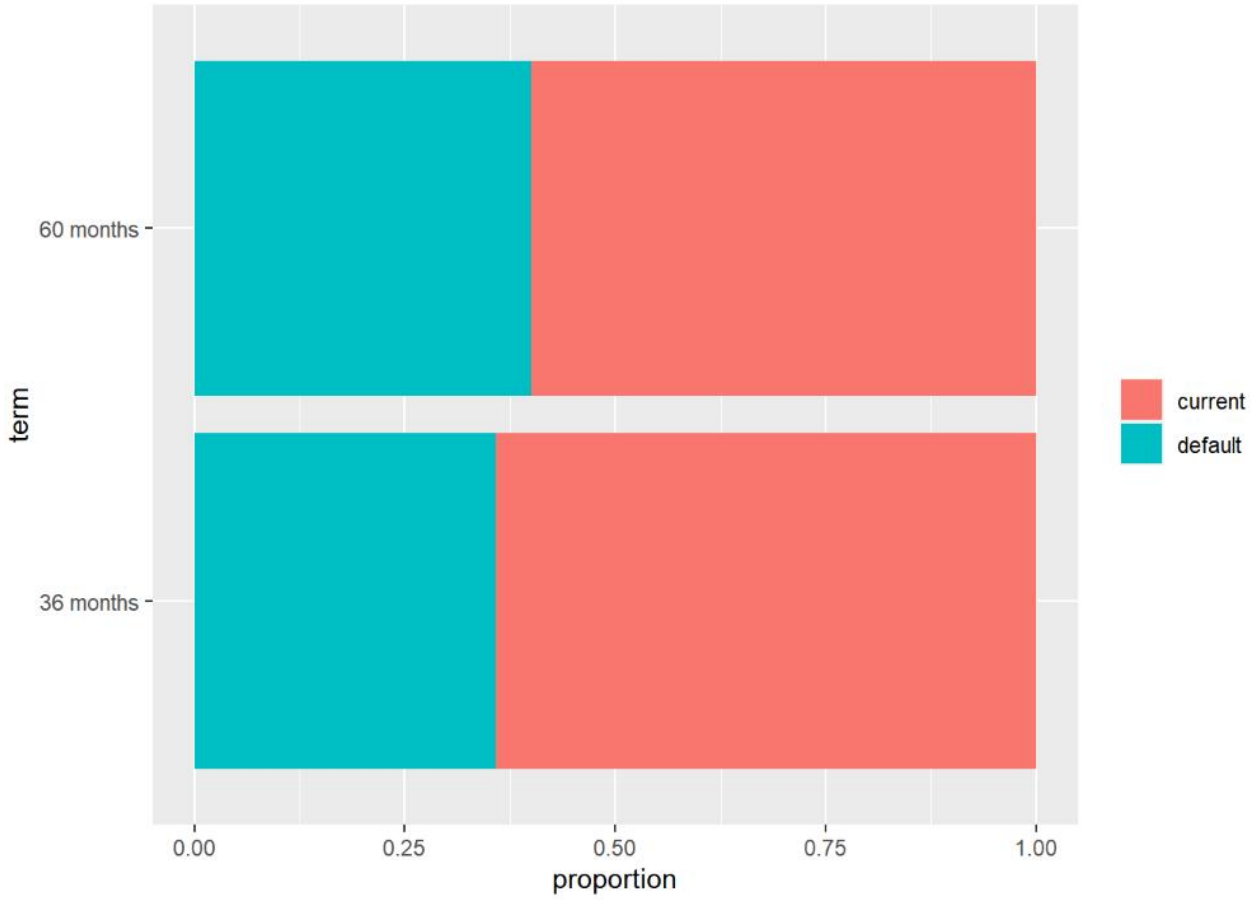
Exploratory Data Analysis (EDA)

Descriptive Statistics

For numerical variables, I used boxplots to explore the relation between explanatory variables and target variables.

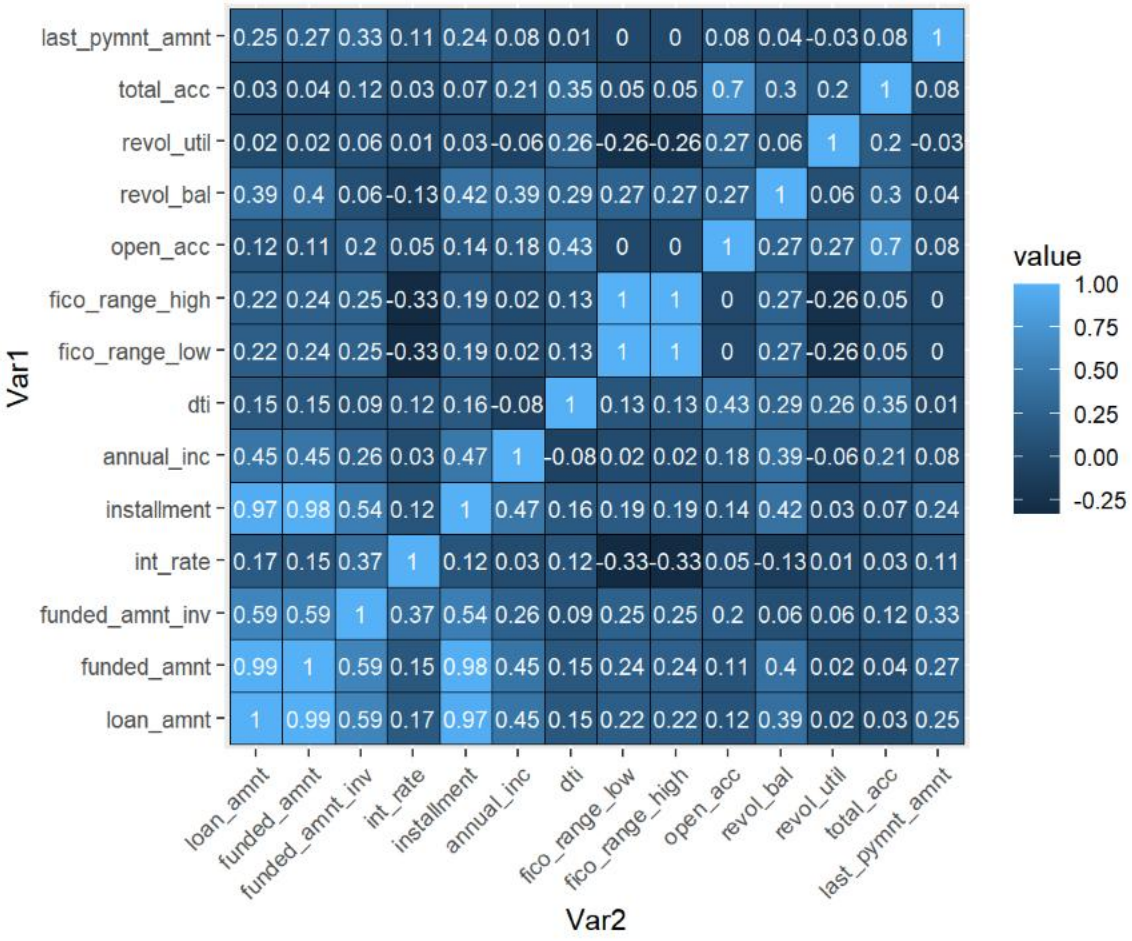


For categorical variables, I used bars charts to explore the relation between explanatory variables and target variables


60 months
36 months
term
0.00
0.25
0.50
0.75
1.00
proportion
current
default

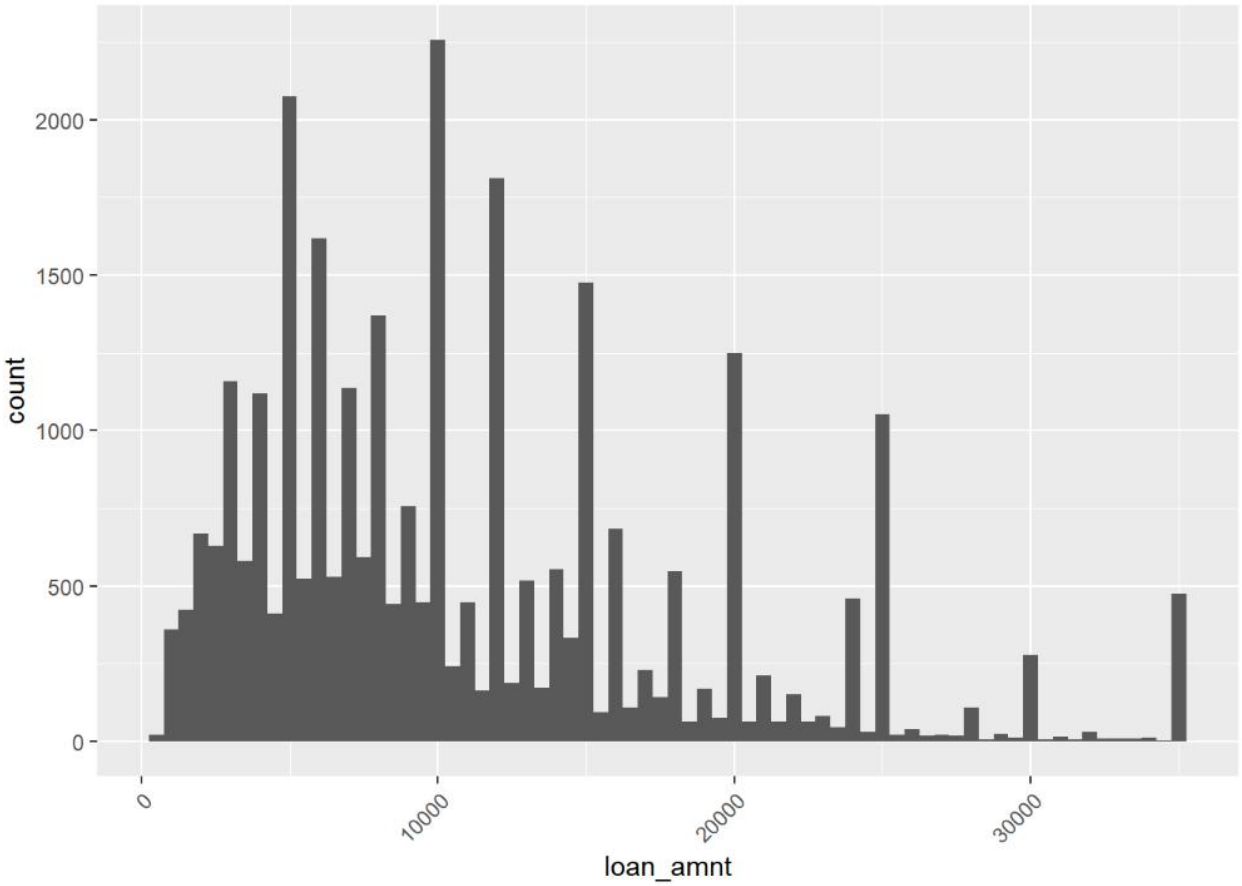
Correlation Analysis

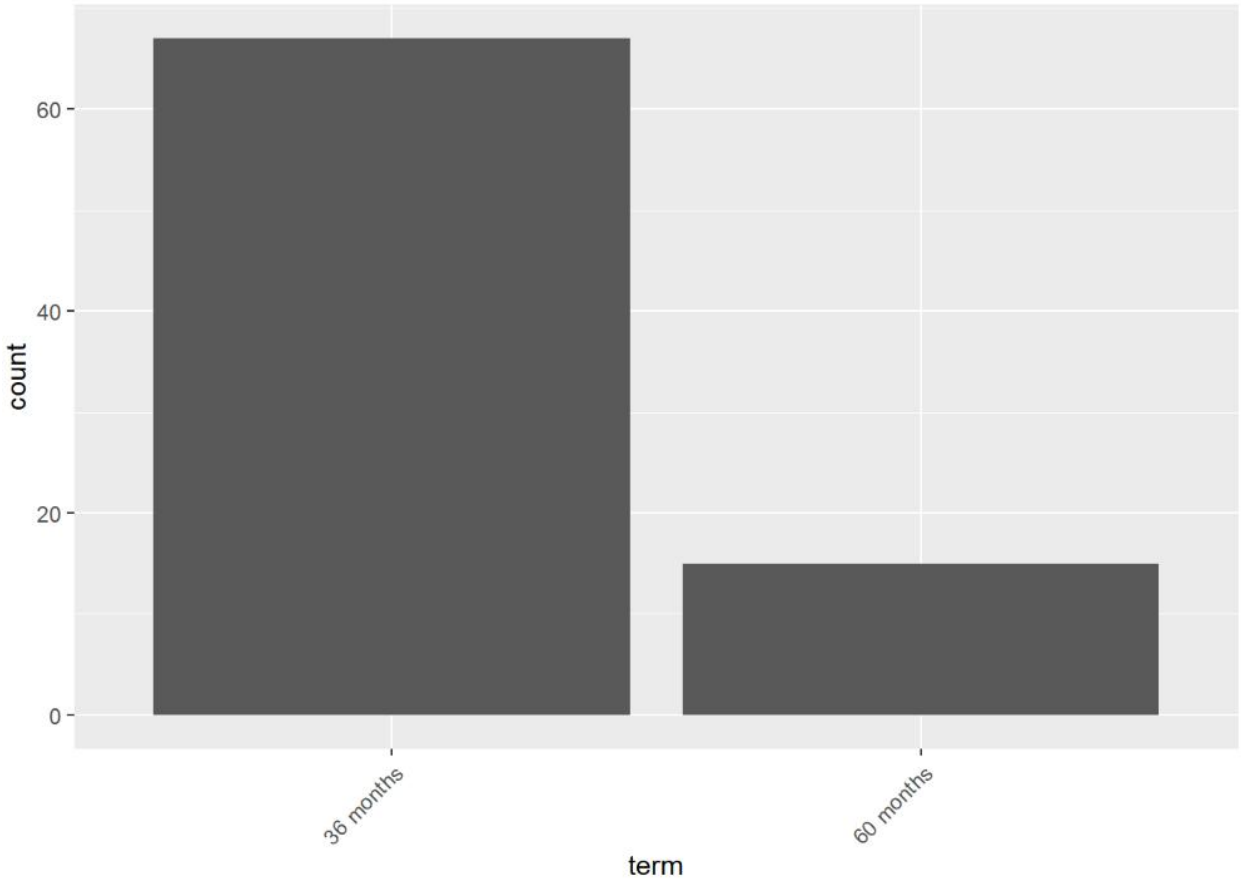
I used a heatmap to explore correlations between numerical variables.



Frequency Analysis

I used histograms and bar charts respectively to conduct frequency analysis of numerical and categorical variables.

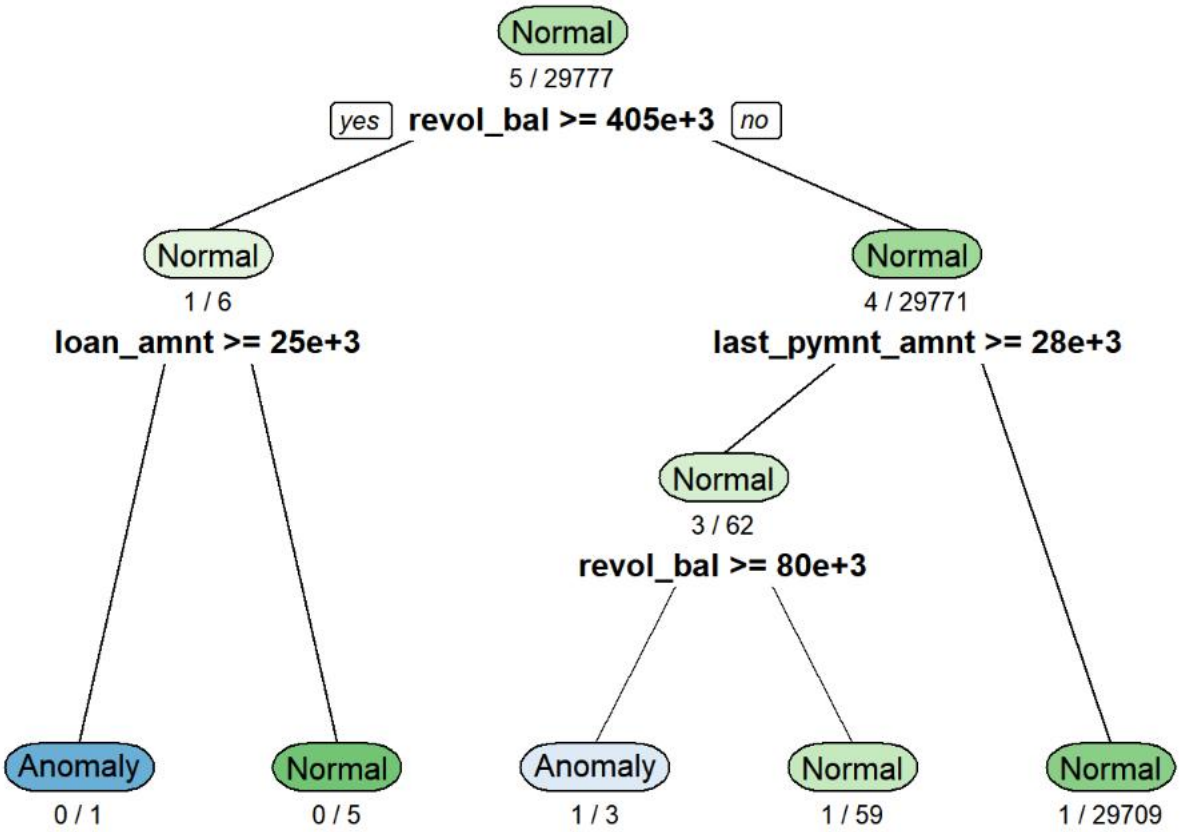




Anomaly Detection (Isolation Forest)

Splitting Criteria

Here is the decision tree to detect anomaly.



Also, We can see some criteria for determining some abnormal records..

	rule
	<chr>
13	IF revol_bal < 79777 & last_pymnt_amnt >= 27804 THEN Normal coverage 0%
7	IF revol_bal < 404868 & last_pymnt_amnt < 27804 THEN Normal coverage 100%
5	IF revol_bal >= 404868 & loan_amnt < 24500 THEN Normal coverage 0%

Anomaly Records

I detected top five anomaly records with top five highest anomaly scores and remove these five columns from the dataset.

id <int>	average_depth <dbl>	anomaly_score <dbl>	loan_amnt <dbl>	funded_amnt <dbl>	funded_amnt_inv <dbl>	int_rate <dbl>	installment <dbl>	annual_inc <dbl>
5259	6.92	0.6261297	35000	35000	31975.0000	0.2030	933.14	543000
29658	6.95	0.6248601	25000	25000	2450.0028	0.1533	870.68	165000
2737	6.99	0.6231713	35000	35000	34750.0000	0.2235	973.64	240000
2658	7.07	0.6198074	35000	35000	34977.3467	0.2167	960.11	250000
29546	7.08	0.6193882	25000	25000	474.9995	0.1375	851.41	320000

5 rows | 1-9 of 173 columns

Model Building & Training

I build four models for comparison purposes, including logistic regression, neural network, random forest, and XGBoost. I partitioned my data in a Train / Test split and used K-Fold cross-validation. I used ***hyper parameter tuning and k-fold cross-validation*** in my neural network, random forest, and XGBoost model, created a workflow, and fitted the model.

Recipe

I used ***lasso*** to select significant variables, and then put them into my recipe: loan_status ~ loan_amnt + annual_inc + inq_last_6mths + last_pymnt_amnt + term + int_rate + emp_length + purpose + issue_d_year + last_pymnt_d_year + grade + last_credit_pull_d_year + funded_amnt + fico_range_low + total_rec_late_fee

term <chr>	estimate <dbl>	penalty <dbl>
(Intercept)	-2.08913703	0.01
loan_amnt	0.12717119	0.01
int_rate	0.27842407	0.01
total_rec_late_fee	0.18804891	0.01
last_pymnt_amnt	-2.44474992	0.01
issue_d_year	-0.68560973	0.01
last_pymnt_d_year	1.67780666	0.01
last_credit_pull_d_year	-1.03085043	0.01
term_X36.months	-1.22695142	0.01
term_X60.months	0.32174797	0.01

1-10 of 12 rows Previous 1 2 Next

emp_length_n.a	0.28925935	0.01
purpose_small_business	0.04575169	0.01

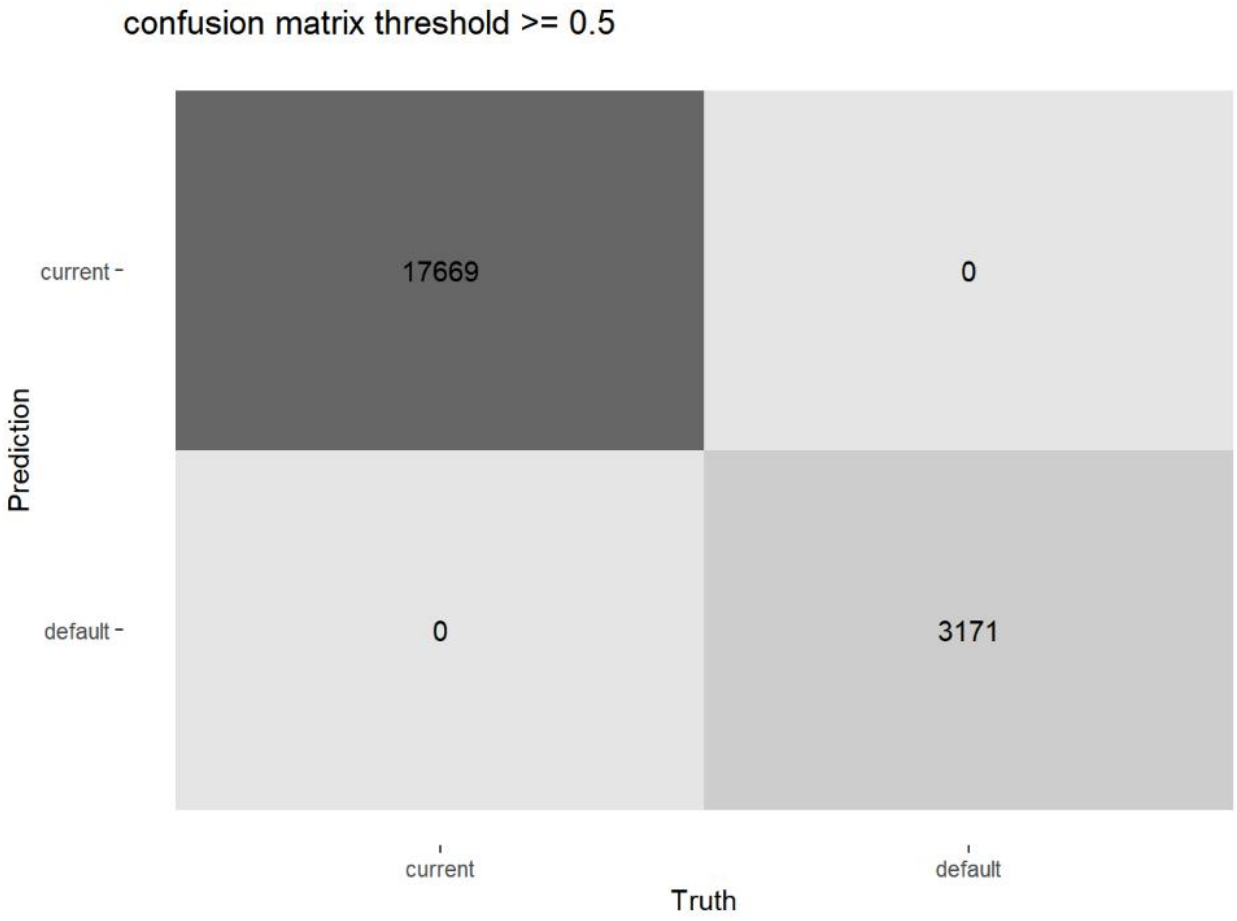
Model Evaluation

As for imbalanced data, accuracy is no longer a good indicator of evaluating whether the model is good or not. We use AUC (Area under the ROC Curve) as

the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. After comparing AUC, I found my XGBoost model is the best one with the highest AUC in the test set - 0.9824.

Confusion Matrix

Here is confusion matrices and my training and testing test. We can use these matrices to calculate the following matrices in the table of performance.



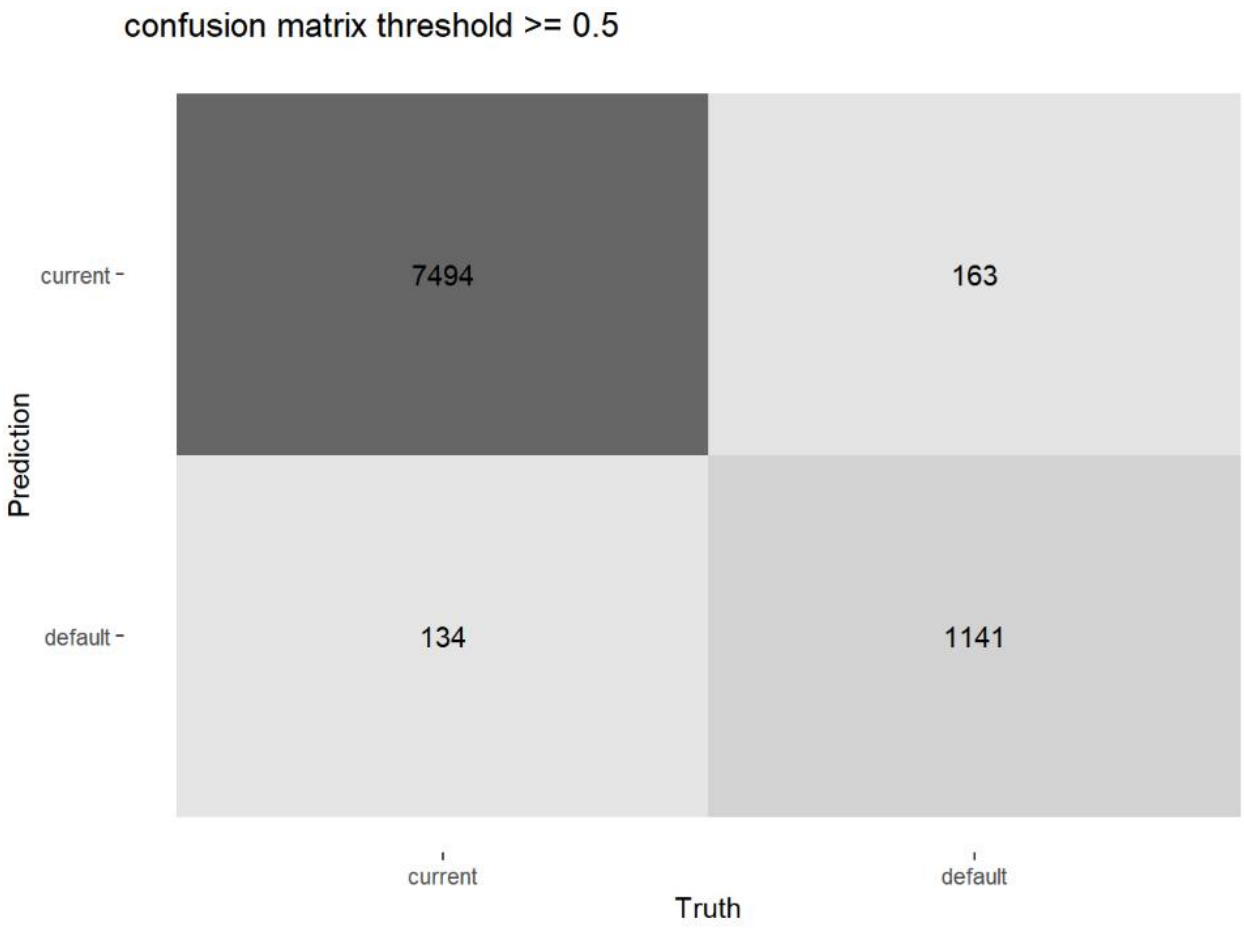


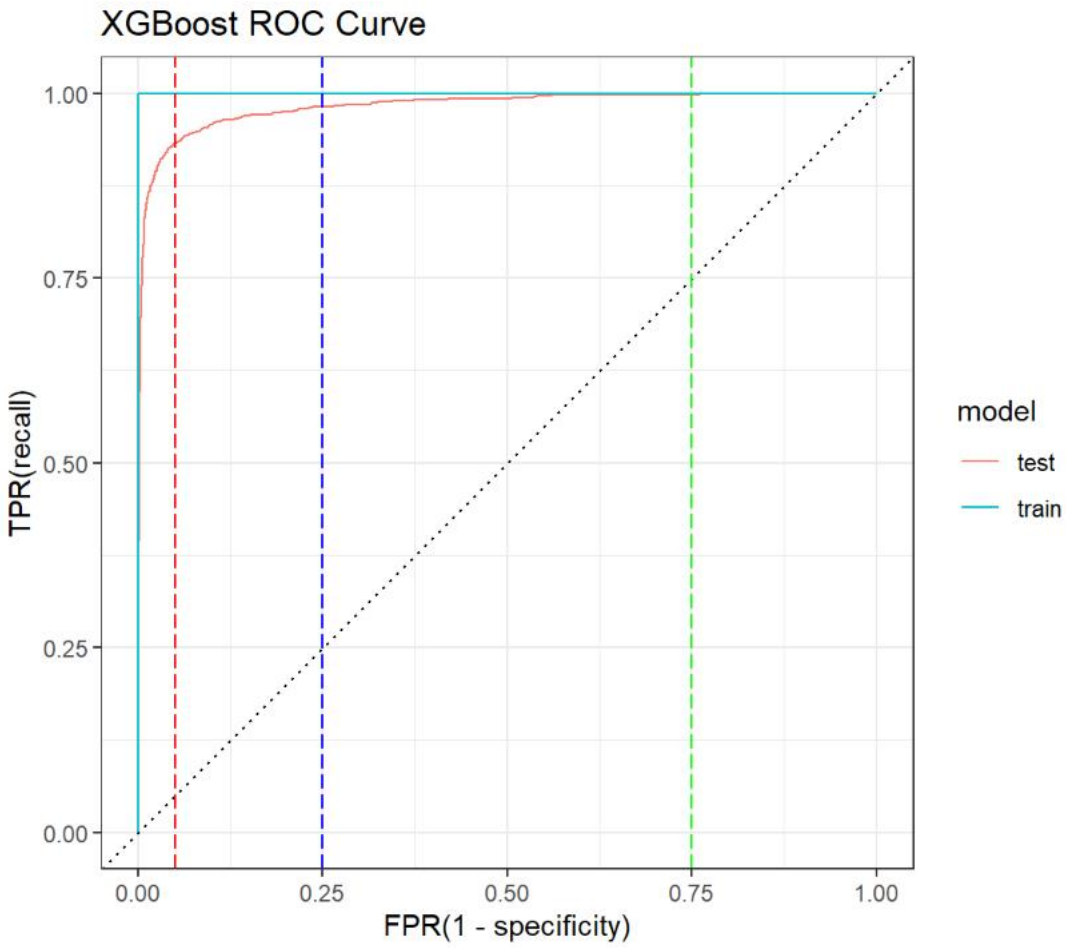
Table of performance

Here is the table summarizing AUC, LogLoss, Precision, Recall, F1 for my XGBoost on train and test sets.

.metric	.estimator	.estimate	part
<chr>	<chr>	<dbl>	<chr>
accuracy	binary	0.9523033	training
mn_log_loss	binary	0.1623706	training
roc_auc	binary	0.9867277	training
precision	binary	0.9706010	training
recall	binary	0.7079786	training
F1_Score	binary	0.8187454	training
accuracy	binary	0.9444693	testing
mn_log_loss	binary	0.1797455	testing
roc_auc	binary	0.9732740	testing
precision	binary	0.9706010	testing
recall	binary	0.7079786	testing
F1_Score	binary	0.7791630	testing

ROC Chart

The ROC Curve is a graph showing the performance of a classification model at all classification thresholds. Here is the ROC Chart of my XGBoost model at 5% FPR, 25% FPR, and 75% FPR.



As we know, the threshold (.pred_default) can influence the precision and recall of a model. However, since the threshold has no influence on AUC, I just used the default threshold - 0.5. When threshold (.pred_default) = 0.5, the FPR is 0.018, and the TPR is 0.876.

Precision Recall Chart

Here is the precision recall chart. We can find that when the threshold (.pred_default) equals to 0.5, the FPR is 0.018, and the TPR is 0.876.

fpr <dbl>	threshold <dbl>	tpr <dbl>
0.010	0.8302	0.8369048
0.011	0.7913	0.8457222
0.012	0.7506	0.8509000
0.013	0.7192	0.8563333
0.014	0.6498	0.8635625
0.015	0.5966	0.8672000
0.016	0.5679	0.8680000
0.017	0.5267	0.8714706
0.018	0.4849	0.8760000
0.019	0.4535	0.8792727

Operating Table

Here is the operating table. We can find that when the threshold (.pred_default) equals to 0.5, the precision is 0.913, and the recall is 0.87.

recall <dbl>	precision <dbl>	.threshold <dbl>
0.80	0.946	0.909
0.81	0.944	0.894
0.82	0.943	0.880
0.83	0.941	0.841
0.84	0.936	0.798
0.85	0.929	0.728
0.86	0.920	0.649
0.87	0.913	0.508
0.88	0.896	0.379
0.89	0.876	0.273

Model Comparison

As for imbalanced data, accuracy is no longer a good indicator of evaluating whether the model is good or not. We use AUC (Area under the ROC Curve) as the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the model's performance in distinguishing between the positive and negative classes.

Here is the test AUC is my neural network model.

.metric	.estimator	.estimate	part
<chr>	<chr>	<dbl>	<chr>
accuracy	binary	0.9672265	training
mn_log_loss	binary	0.3556071	training
roc_auc	binary	0.9848973	training
precision	binary	0.9127405	training
recall	binary	0.8675497	training
F1_Score	binary	0.8895715	training
accuracy	binary	0.9611509	testing
mn_log_loss	binary	0.3607428	testing
roc_auc	binary	0.9770547	testing
precision	binary	0.8706429	testing
recall	binary	0.8619632	testing
F1_Score	binary	0.8662813	testing

Here is the test AUC is my random forest model.

.metric	.estimator	.estimate	part
<chr>	<chr>	<dbl>	<chr>
accuracy	binary	0.9523033	training
mn_log_loss	binary	0.1623706	training
roc_auc	binary	0.9867277	training
precision	binary	0.9706010	training
recall	binary	0.7079786	training
F1_Score	binary	0.8187454	training
accuracy	binary	0.9444693	testing
mn_log_loss	binary	0.1797455	testing
roc_auc	binary	0.9732740	testing
precision	binary	0.9706010	testing
recall	binary	0.7079786	testing
F1_Score	binary	0.7791630	testing

Here is the test AUC is my XGBoost model.

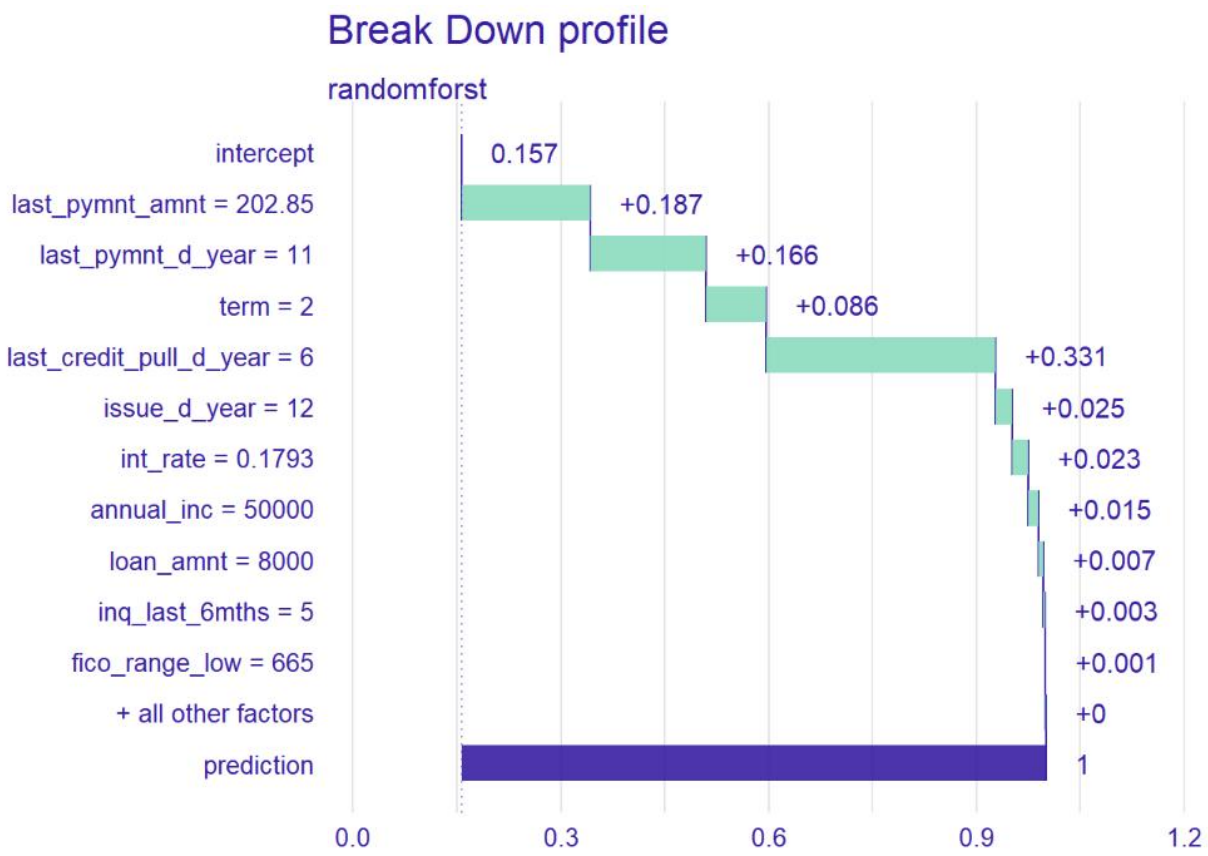
.metric	.estimator	.estimate	part
<chr>	<chr>	<dbl>	<chr>
accuracy	binary	1.000000000	training
mn_log_loss	binary	0.004986086	training
roc_auc	binary	1.000000000	training
precision	binary	0.970600951	training
recall	binary	0.707978556	training
F1_Score	binary	0.818745441	training
accuracy	binary	0.966748768	testing
mn_log_loss	binary	0.120248885	testing
roc_auc	binary	0.982414040	testing
precision	binary	0.970600951	testing

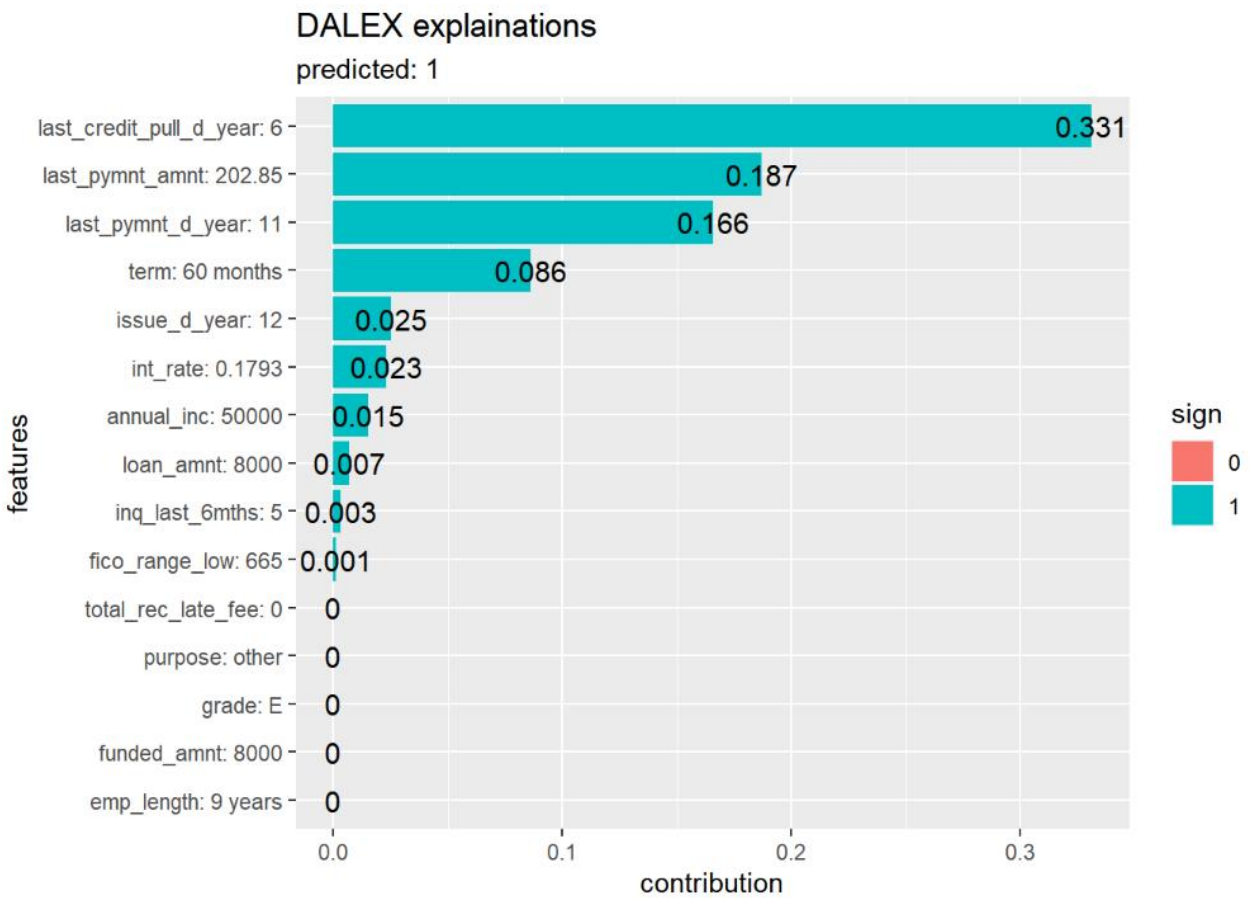
.metric	.estimator	.estimate	part
<chr>	<chr>	<dbl>	<chr>
recall	binary	0.707978556	testing
F1_Score	binary	0.779162956	testing

Since the XGBoost model has the highest AUC (0.9824) among all these models, we chose XGBoost as my best model.

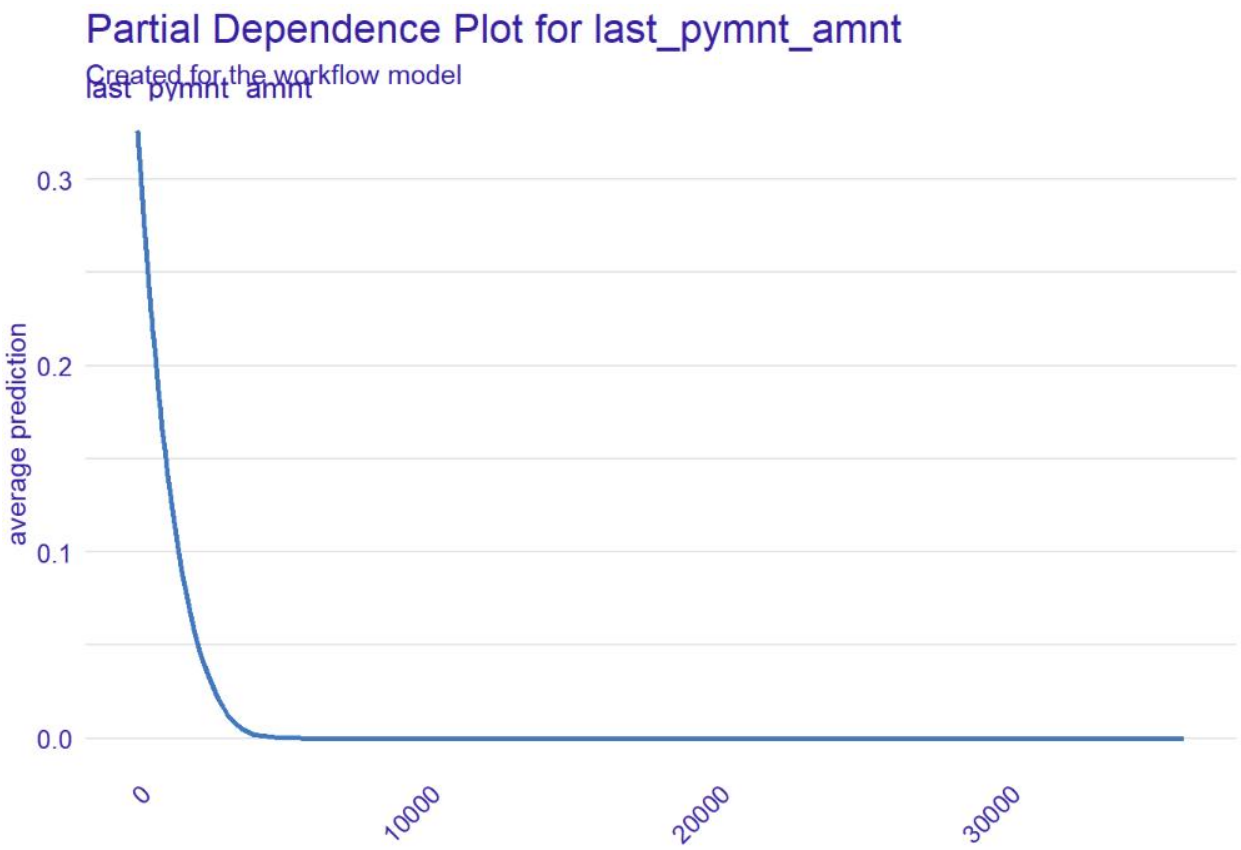
Global Explanations

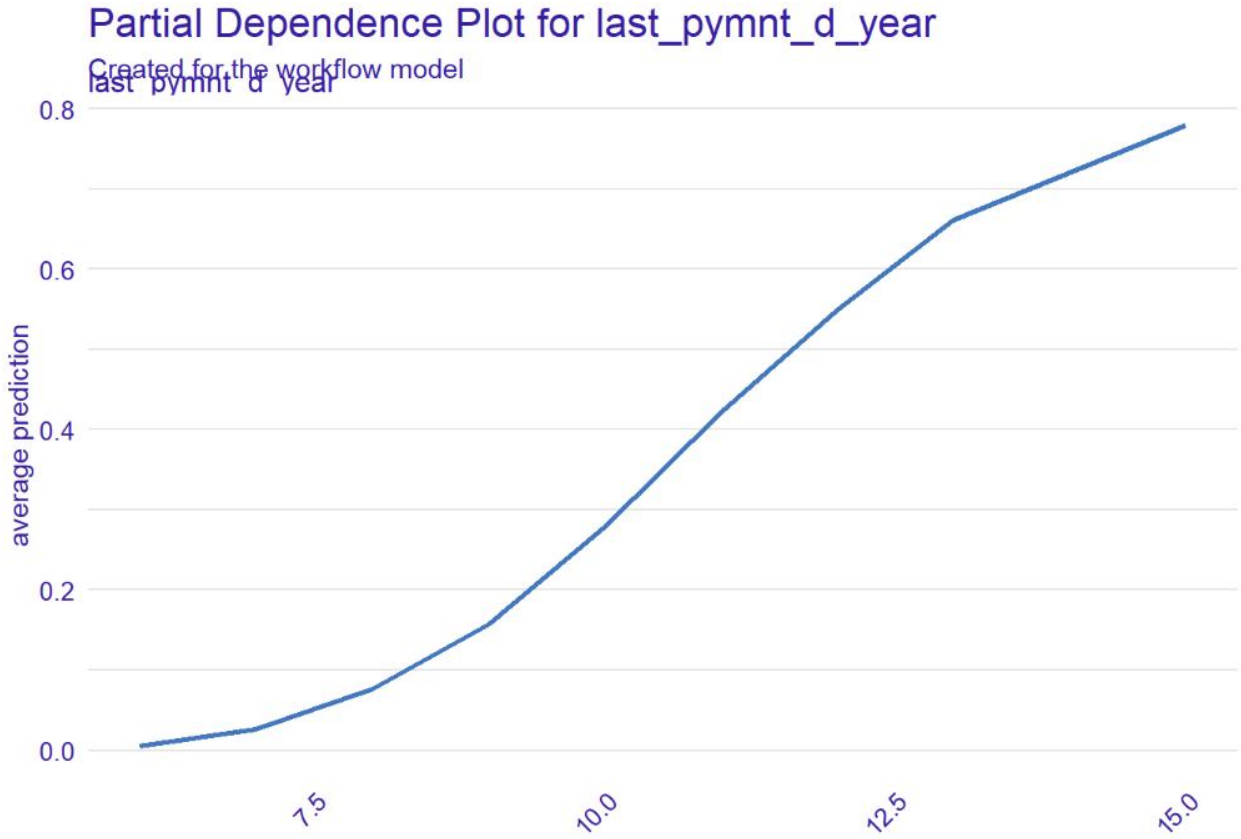
I evaluated variable importance using DALEX (Breakdown) for my XGBoost model.

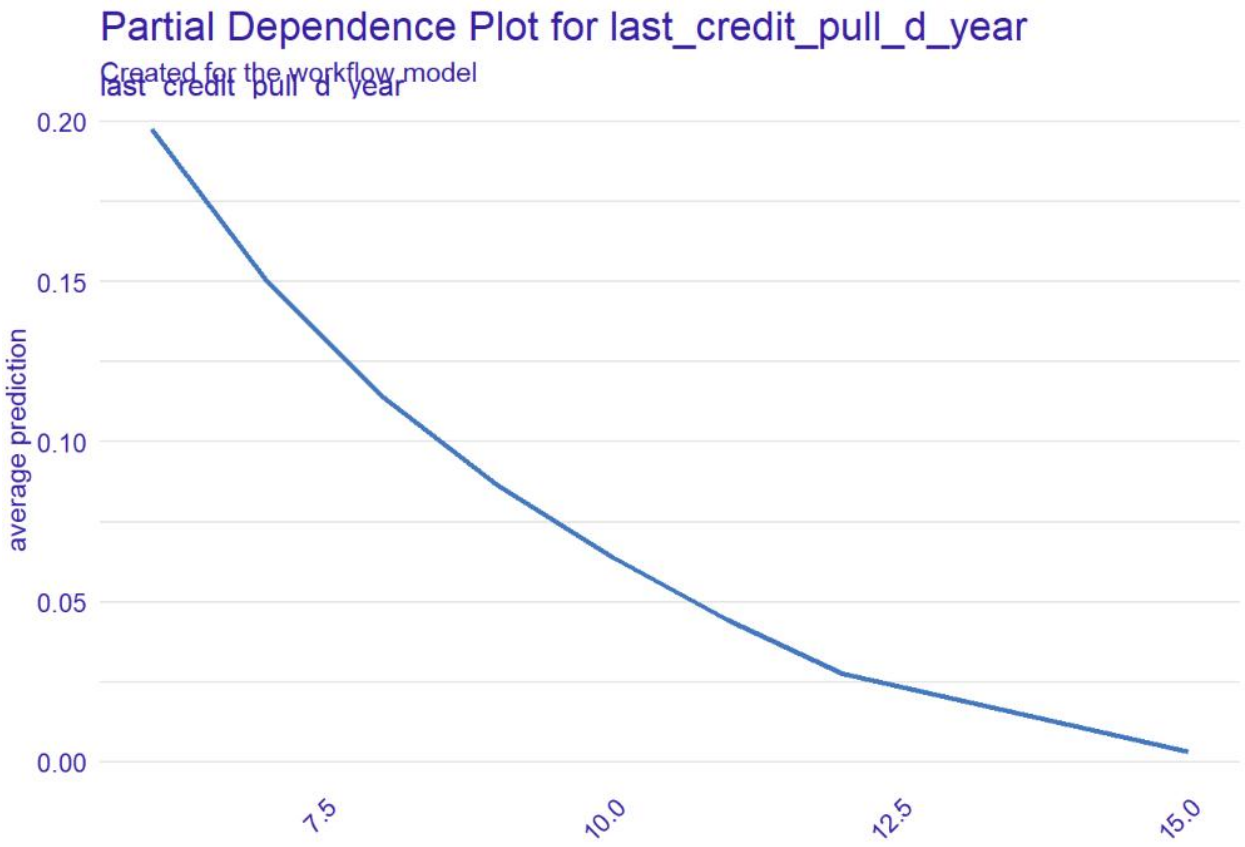

Break Down profile
randomforst
intercept
last_pymnt_amnt = 202.85
last_pymnt_d_year = 11
term = 2
last_credit_pull_d_year = 6
issue_d_year = 12
int_rate = 0.1793
annual_inc = 50000
loan_amnt = 8000
inq_last_6mths = 5
fico_range_low = 665
+ all other factors
prediction
0.157
+0.187
+0.166
+0.086
+0.331
+0.025
+0.023
+0.015
+0.007
+0.003
+0.001
+0
1
0.0
0.3
0.6
0.9
1.2

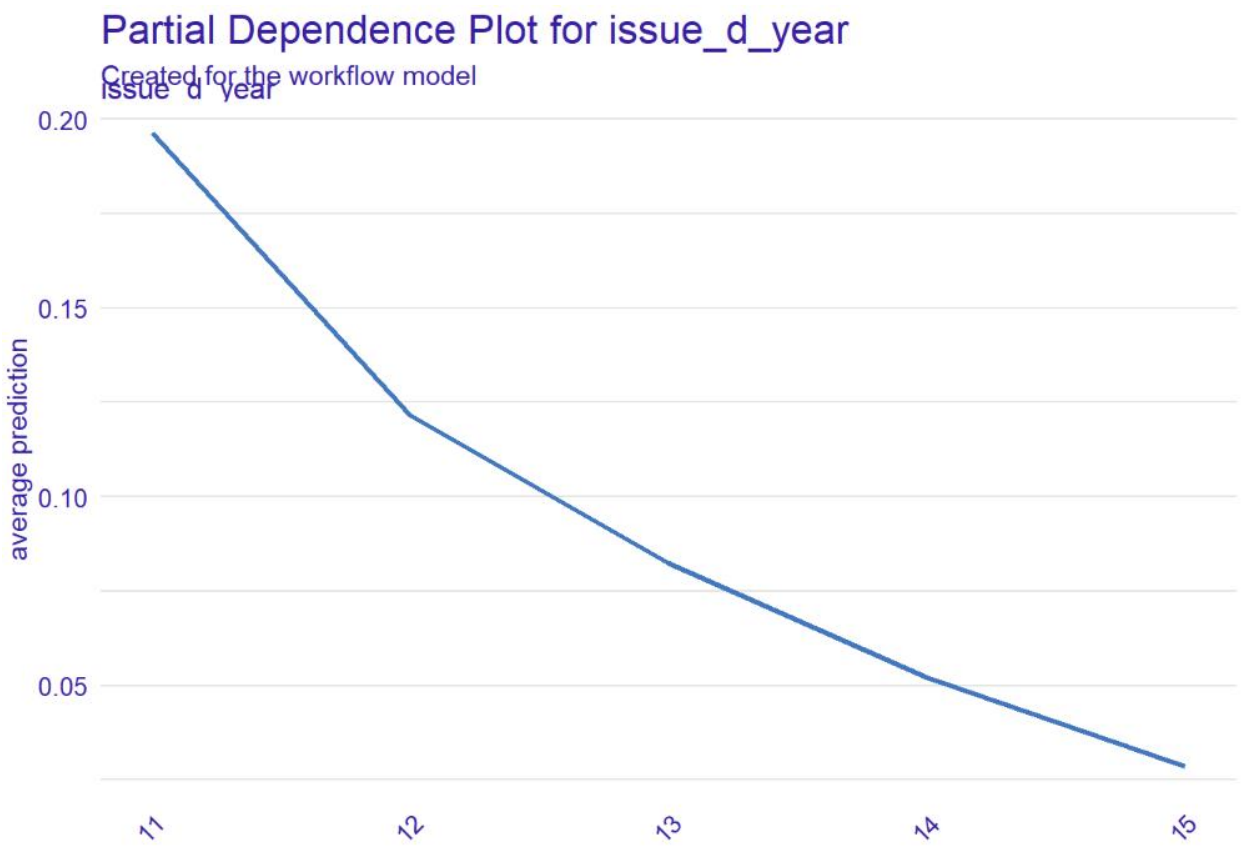


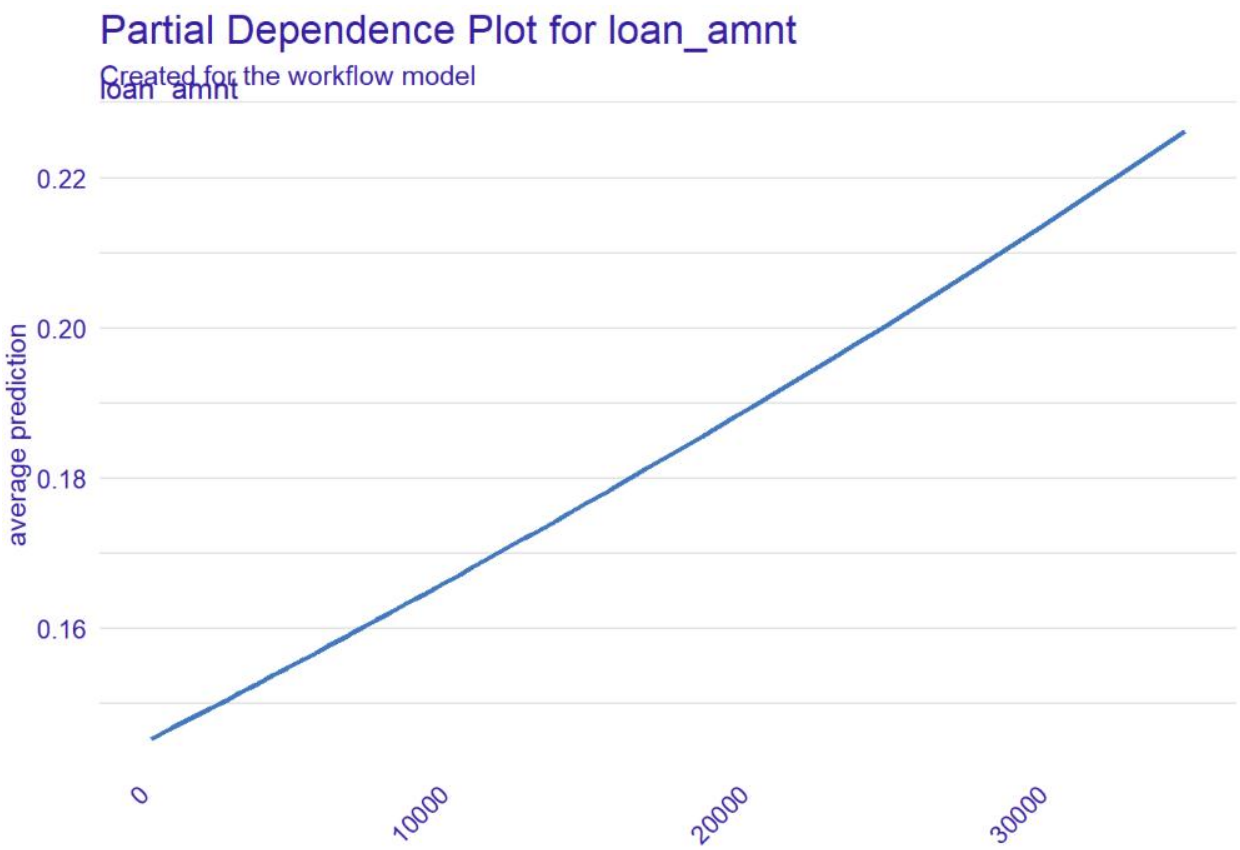
Also, I drew partial dependency plots of my top variables: last_pymnt_amnt, last_pymnt_d_year, last_credit_pull_d_year, issue_d_year, loan_amnt


Partial Dependence Plot for last_pymnt_amnt
Created for the workflow model
last_pymnt_amnt
average prediction
0.3
0.2
0.1
0.0
0
10000
20000
30000


Partial Dependence Plot for last_pymnt_d_year
Created for the workflow model
last_pymnt_d_year
average prediction
0.8
0.6
0.4
0.2
0.0
7.5
10.0
12.5
15.0

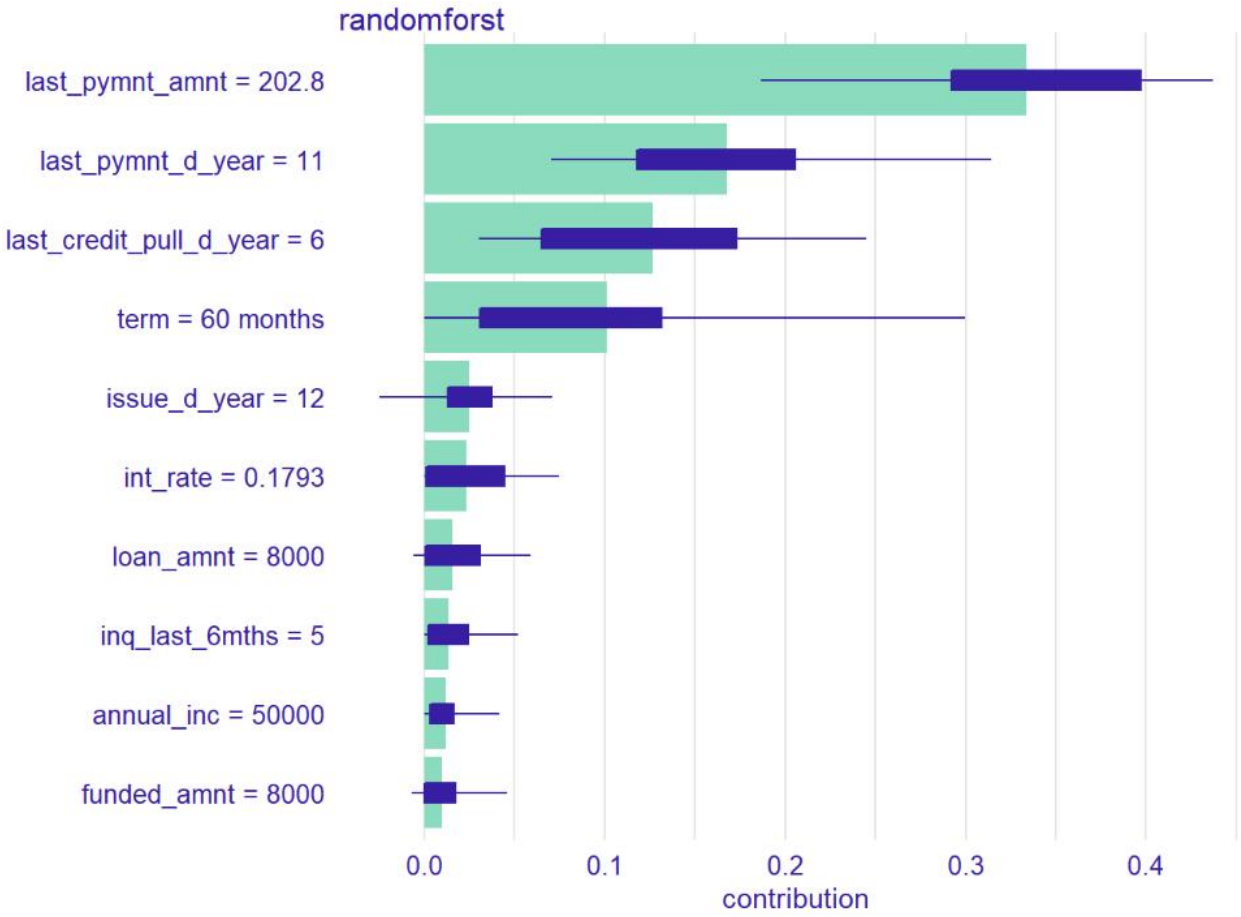

Partial Dependence Plot for last_credit_pull_d_year
Created for the workflow model
last_credit_pull_d_year
average prediction
0.20
0.15
0.10
0.05
0.00
7.5
10.0
12.5
15.0

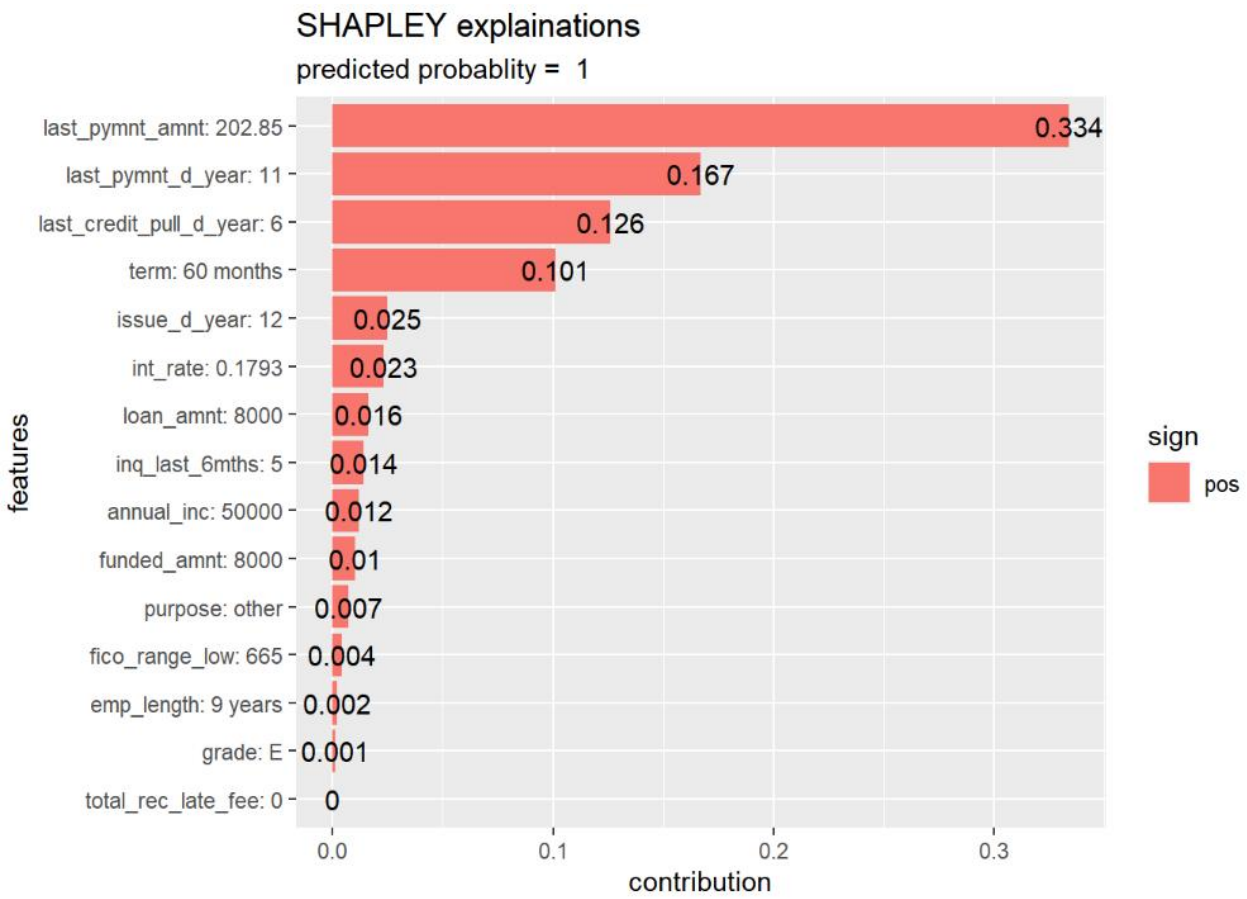

Partial Dependence Plot for issue_d_year
Created for the workflow model
issue_d_year
average prediction
0.20
0.15
0.10
0.05
11
12
13
14
15



Local Explanations

In this part, I used SHAP to evaluate variable importance.


randomforst
last_pymnt_amnt = 202.8
last_pymnt_d_year = 11
last_credit_pull_d_year = 6
term = 60 months
issue_d_year = 12
int_rate = 0.1793
loan_amnt = 8000
inq_last_6mths = 5
annual_inc = 50000
funded_amnt = 8000
0.0
0.1
0.2
0.3
0.4
contribution



Here are my top 10 true positives, loan default = 1 and ordered by pred_1 score DECENDING

.pred_current <dbl>	.pred_default <dbl>	.pred_class <fct>	loan_status <fct>	loan_amnt <dbl>	funded_amnt <dbl>	funded_amnt_inv <dbl>	int_rate <dbl>
0.000008220754	0.9999918	default	default	8000	8000	6725.0000	0.1793
0.000010558794	0.9999894	default	default	10000	10000	9175.0000	0.1316
0.000011241410	0.9999888	default	default	10000	10000	9975.0000	0.1311
0.000013776330	0.9999862	default	default	10000	10000	9950.0000	0.1385
0.000014257276	0.9999857	default	default	5000	5000	5000.0000	0.1348
0.000017836512	0.9999822	default	default	18000	13250	13250.0000	0.2264
0.000023066857	0.9999769	default	default	15000	15000	14628.4400	0.1459
0.000027022774	0.9999730	default	default	10000	10000	9900.0000	0.1218
0.000027126385	0.9999729	default	default	20000	20000	575.0019	0.1299
0.000027843398	0.9999722	default	default	6000	6000	5950.0000	0.1632

1-10 of 10 rows | 1-8 of 36 columns

Here are my top 10 false positives, loan default = 0 and ordered by pred_1 score DECENDING (high scoring but actually didn't default)

.pred_current <dbl>	.pred_default <dbl>	.pred_class <fct>	loan_status <fct>	loan_amnt <dbl>	funded_amnt <dbl>	funded_amnt_inv <dbl>	int_rate <dbl>
0.00009785406	0.9999021	default	current	2000	2000	2000.00	0.1316
0.00032366693	0.9996763	default	current	5000	5000	5000.00	0.0999
0.00078116619	0.9992188	default	current	12000	12000	9887.01	0.1316
0.00108001370	0.9989200	default	current	13000	13000	13000.00	0.1749
0.00110448664	0.9988955	default	current	4075	4075	4050.00	0.0832
0.00136160932	0.9986384	default	current	12000	12000	12000.00	0.1036
0.00142461131	0.9985754	default	current	20000	20000	19950.00	0.1479
0.00145760225	0.9985424	default	current	4000	4000	3750.00	0.1148
0.00201453571	0.9979855	default	current	12000	12000	12000.00	0.1599
0.00230993028	0.9976901	default	current	12000	12000	12000.00	0.1527

1-10 of 10 rows | 1-8 of 36 columns

Here are my top 10 true negatives, loan default = 1 and ordered by pred_1 score ASCENDING (low scoring that did default)

.pred_current <dbl>	.pred_default <dbl>	.pred_class <fct>	loan_status <fct>	loan_amnt <dbl>	funded_amnt <dbl>	funded_amnt_inv <dbl>	int_rate <dbl>
0.9999849	0.00001513958	current	default	3600	3600	3600.000	0.1398
0.9999803	0.00001966953	current	default	15850	15850	15844.212	0.1565
0.9999504	0.00004959106	current	default	6500	6500	6496.816	0.0579
0.9999201	0.00007987022	current	default	1000	1000	900.000	0.1148
0.9999171	0.00008285046	current	default	10000	10000	10000.000	0.1242
0.9999070	0.00009298325	current	default	25000	15925	15850.000	0.2077
0.9999046	0.00009536743	current	default	6000	6000	6000.000	0.0849
0.9998919	0.00010812283	current	default	8000	8000	8000.000	0.0890
0.9998821	0.00011789799	current	default	12250	12250	12250.000	0.0691
0.9997916	0.00020843744	current	default	4000	4000	4000.000	0.1399

1-10 of 10 rows | 1-8 of 36 columns

Predictions File

Finally, I used loan_holdout.csv to make predictions and saved my predictions file, which contains two columns: id, loan_status.

id	loan_status
<dbl>	<dbl>
1077175	0.19430386
1075358	0.16555820
1075269	0.13288385
1071570	0.77480957
1064687	0.56165513
1065775	0.04974112

6 rows