Classification_Report

AUTHOR
LUKE STUCKY

PUBLISHED
December 11, 2024

BUSINESS UNDERSTANDING

A high-end financial institution is working on reducing credit default rates while identifying high-risk customers. This will allow the institution to improve profitability and minimize financial losses. As the head data analyst, Luke Stucky is working on developing a reliable classification system that can predict whether a customer will default on their credit obligations. Doing this will allow the institution to be more effective in their selection of customers and increase their reputation as a high-end financial institution. To do this, Luke Stucky will create two models, a KNN and a Logistic Regression model to help answer the two questions:

What are the key predictors of credit default among customers, based on their financial and spending behaviors? How can the institution improve the accuracy of default prediction models to effectively mitigate financial risks while minimizing false negatives?

After answering these questions, Luke is hoping to implement one of the models starting the new year.

DATA UNDERSTANDING

Remove variables

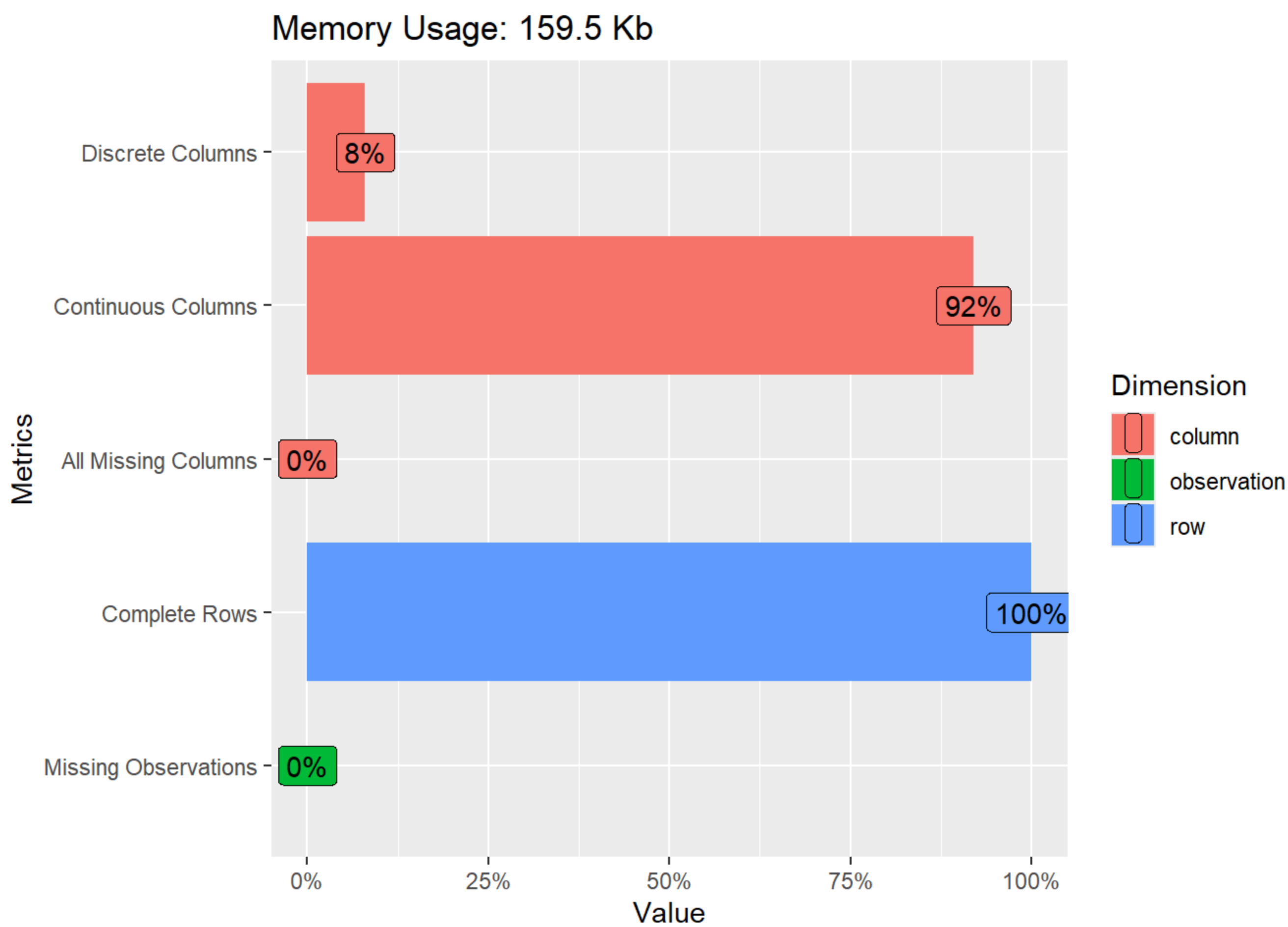
Create the dependent variable

```
Cross-Tabulation, Row Proportions
as.factor(mydata$default) * default

-------------------------- --------- ------------- ------------- ---------------
                             default             0             1           Total
  as.factor(mydata$default)
                          0             716 (100.0%)     0 (  0.0%)    716 (100.0%)
                          1               0 (  0.0%)   284 (100.0%)    284 (100.0%)
                      Total             716 ( 71.6%)   284 ( 28.4%)   1000 (100.0%)
-------------------------- --------- ------------- ------------- ---------------
```

EDA

Check for missing values, variable formats, and data load errors.

```
  default income savings   debt r_savings_income r_debt_income
1       1  33269       0 532304           0.0000       16.0000
2       0  77158   91187 315648           1.1818        4.0909
3       1  30917   21642 534864           0.7000       17.3000
  r_clothing_income r_education_income r_entertainment_income r_fines_income
1            0.0568                  0                 0.0922         0.0000
2            0.0754                  0                 0.2235         0.0000
3            0.0374                  0                 0.1168         0.0012
  r_gambling_income r_groceries_income r_health_income r_housing_income
1            0.0395             0.1458          0.0096           0.0904
2            0.0000             0.0677          0.0061           0.2118
3            0.0388             0.1402          0.0288           0.0892
  r_tax_income r_travel_income r_utilities_income r_expenditure_income
1       0.0000          0.5378             0.0280               1.0000
2       0.0256          0.2622             0.0369               0.9091
3       0.0000          0.5198             0.0277               1.0000
  cat_gambling cat_debt cat_credit_card cat_mortgage cat_savings_account
1         High        1               0            0                   0
2           No        1               0            0                   1
3         High        1               0            0                   1
  cat_dependents credit_score
1              0          444
2              0          625
3              0          469

     default income savings   debt r_savings_income r_debt_income
998        0      0   42428  30760           3.2379        8.1889
999        0  36011    8002 604181           0.2222       16.7777
1000       0  44266  309859  44266           6.9999        1.0000
     r_clothing_income r_education_income r_entertainment_income r_fines_income
998             0.0047             0.0000                 0.3664         0.0005
999             0.0553             0.2672                 0.0996         0.0000
1000            0.0356             0.0000                 0.1693         0.0000
     r_gambling_income r_groceries_income r_health_income r_housing_income
998              0.019             0.1656          0.0000           0.2051
999              0.000             0.1680          0.0242           0.0986
1000             0.000             0.1891          0.2054           0.0000
     r_tax_income r_travel_income r_utilities_income r_expenditure_income
998        0.0346          0.1382             0.0741               1.0668
999        0.0000          0.3678             0.0305               1.1111
1000       0.0084          0.4256             0.0776               1.1111
     cat_gambling cat_debt cat_credit_card cat_mortgage cat_savings_account
998            No        1               0            0                   1
999            No        1               1            0                   1
1000           No        1               0            0                   1
     cat_dependents credit_score
998               0          499
999               0          507
1000              0          657
```

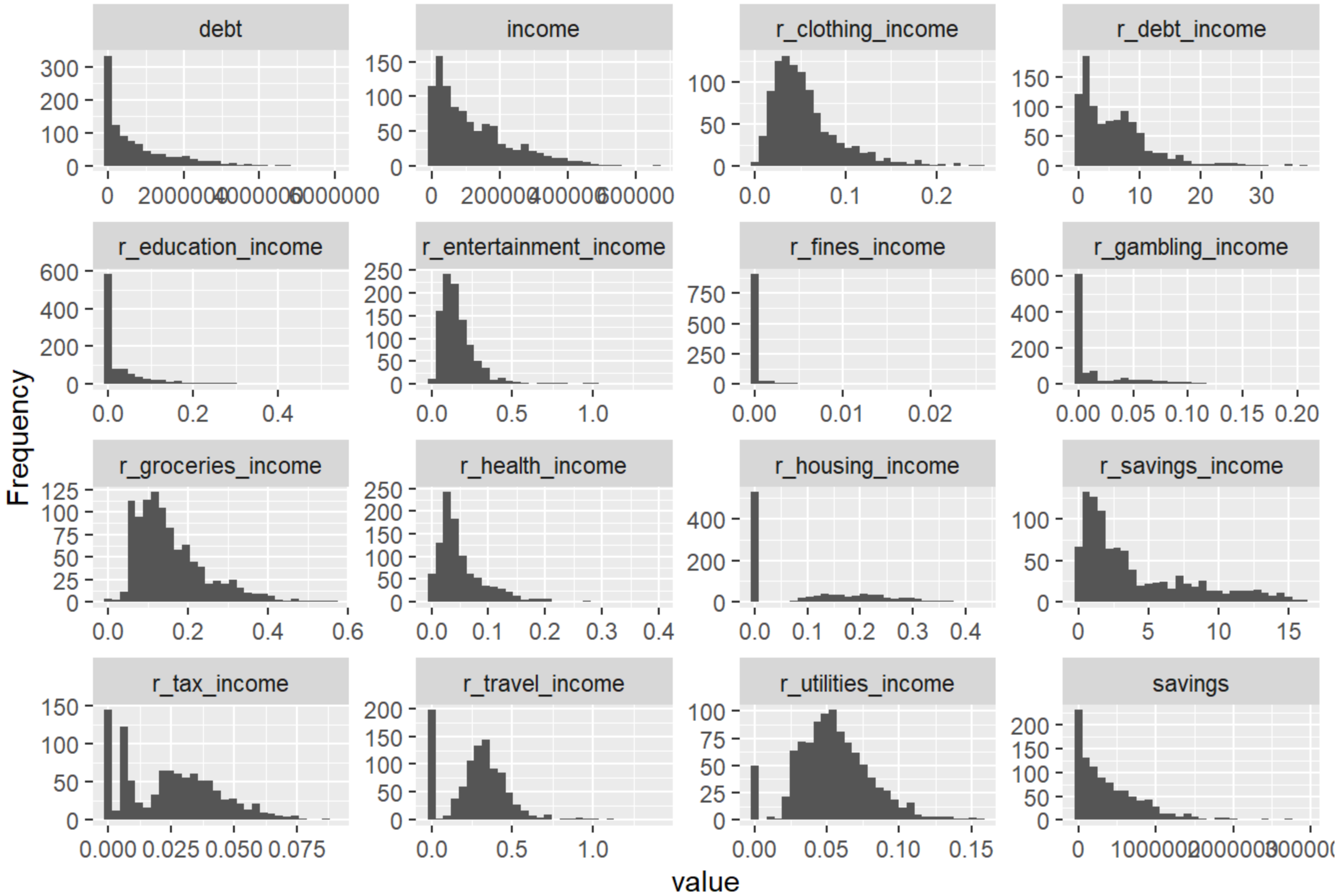


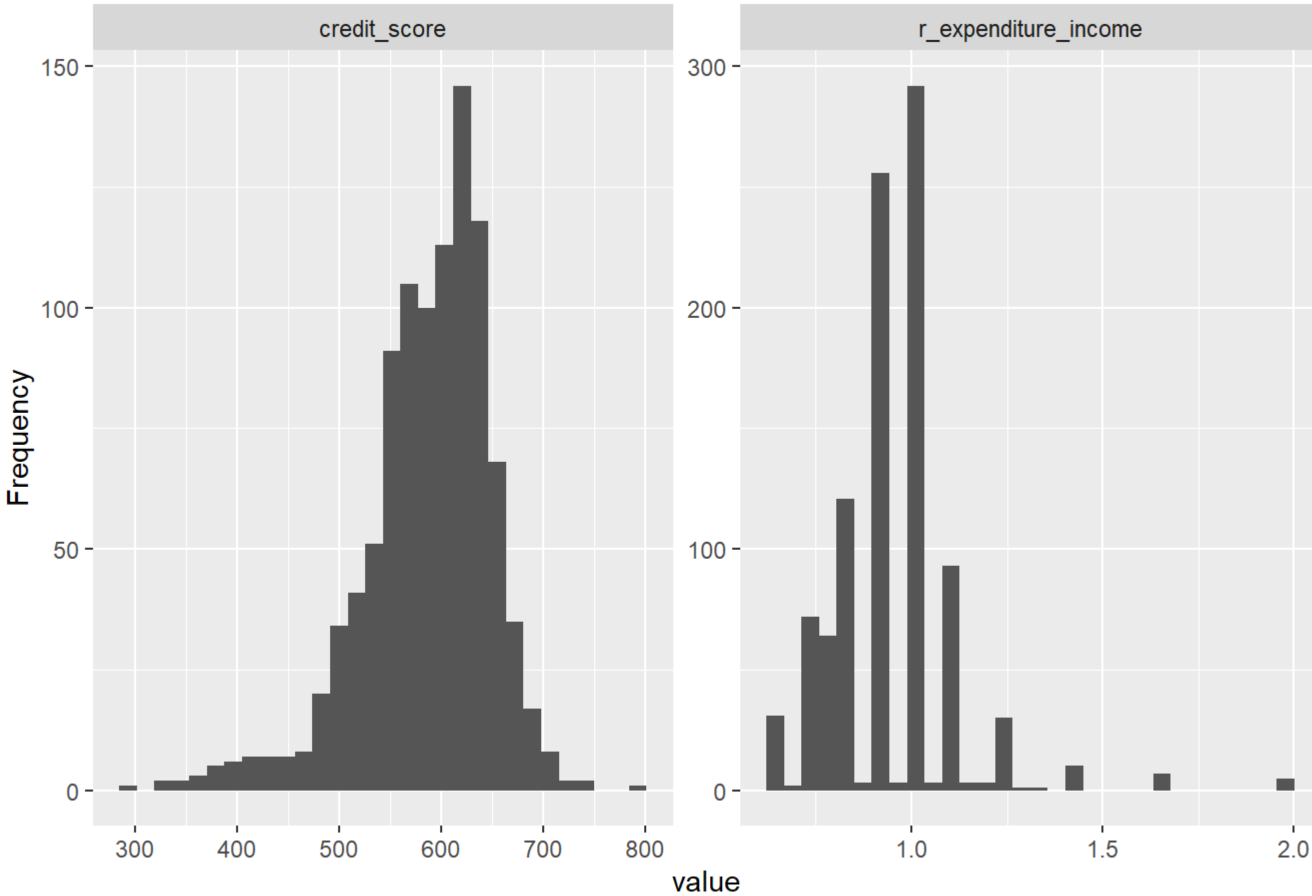
Check default proportion for balance

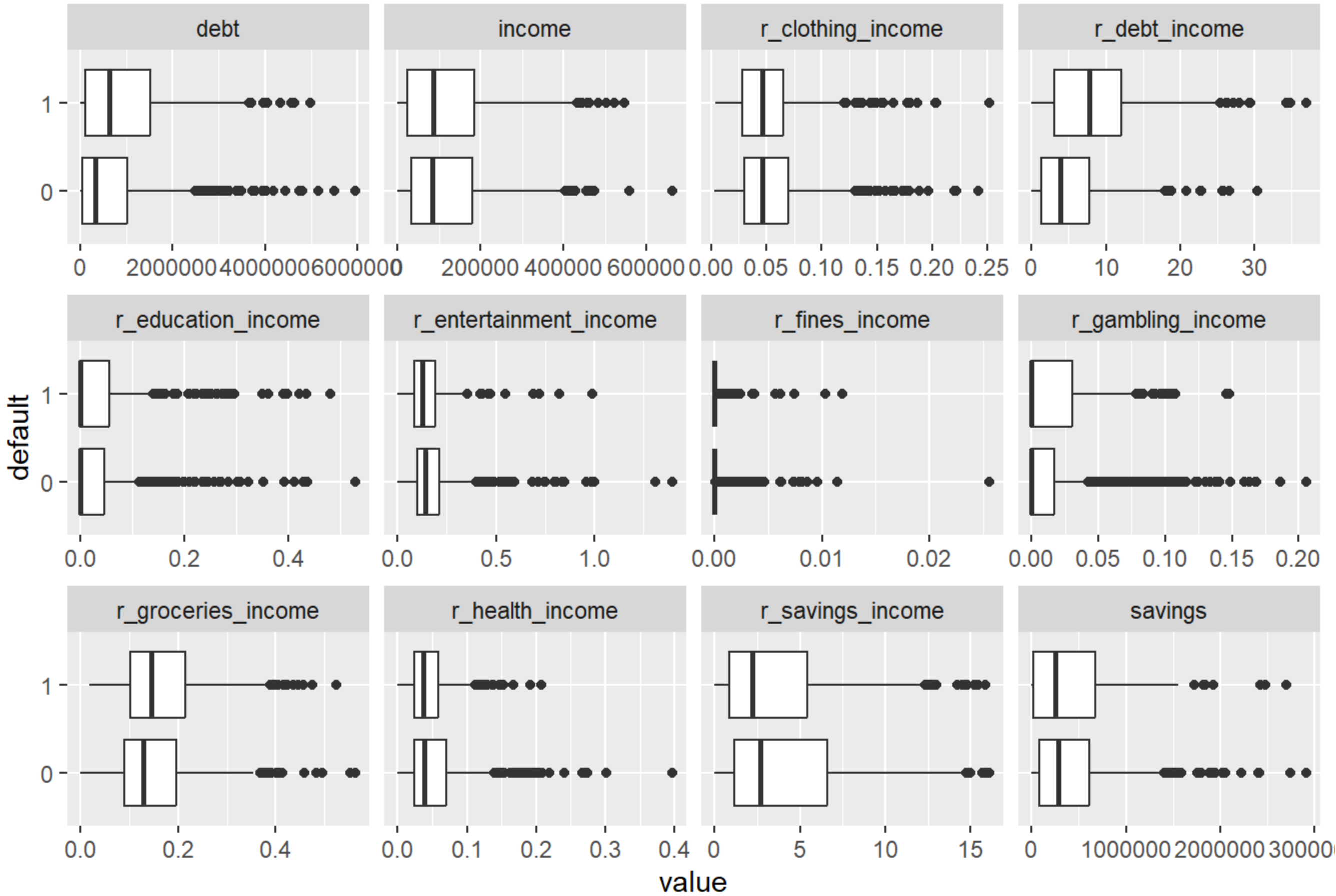
	vars	n	mean	sd	min	max
default	1	1000	NaN	NA	Inf	-Inf
income	2	1000	121610.02	113716.70	0.00	662094.00
savings	3	1000	413189.60	442916.04	0.00	2911863.00
debt	4	1000	790718.04	981790.39	0.00	5968620.00
r_savings_income	5	1000	4.06	3.97	0.00	16.11
r_debt_income	6	1000	6.07	5.85	0.00	37.00
r_clothing_income	7	1000	0.06	0.04	0.00	0.25
r_education_income	8	1000	0.04	0.07	0.00	0.53
r_entertainment_income	9	1000	0.17	0.14	0.00	1.40
r_fines_income	10	1000	0.00	0.00	0.00	0.03
r_gambling_income	11	1000	0.02	0.03	0.00	0.21
r_groceries_income	12	1000	0.16	0.09	0.00	0.56
r_health_income	13	1000	0.05	0.05	0.00	0.40
r_housing_income	14	1000	0.09	0.11	0.00	0.43
r_tax_income	15	1000	0.03	0.02	0.00	0.09
r_travel_income	16	1000	0.28	0.20	0.00	1.40
r_utilities_income	17	1000	0.05	0.03	0.00	0.16
r_expenditure_income	18	1000	0.94	0.17	0.67	2.00
cat_gambling	19	1000	NaN	NA	Inf	-Inf
cat_debt	20	1000	0.94	0.23	0.00	1.00
cat_credit_card	21	1000	0.24	0.42	0.00	1.00

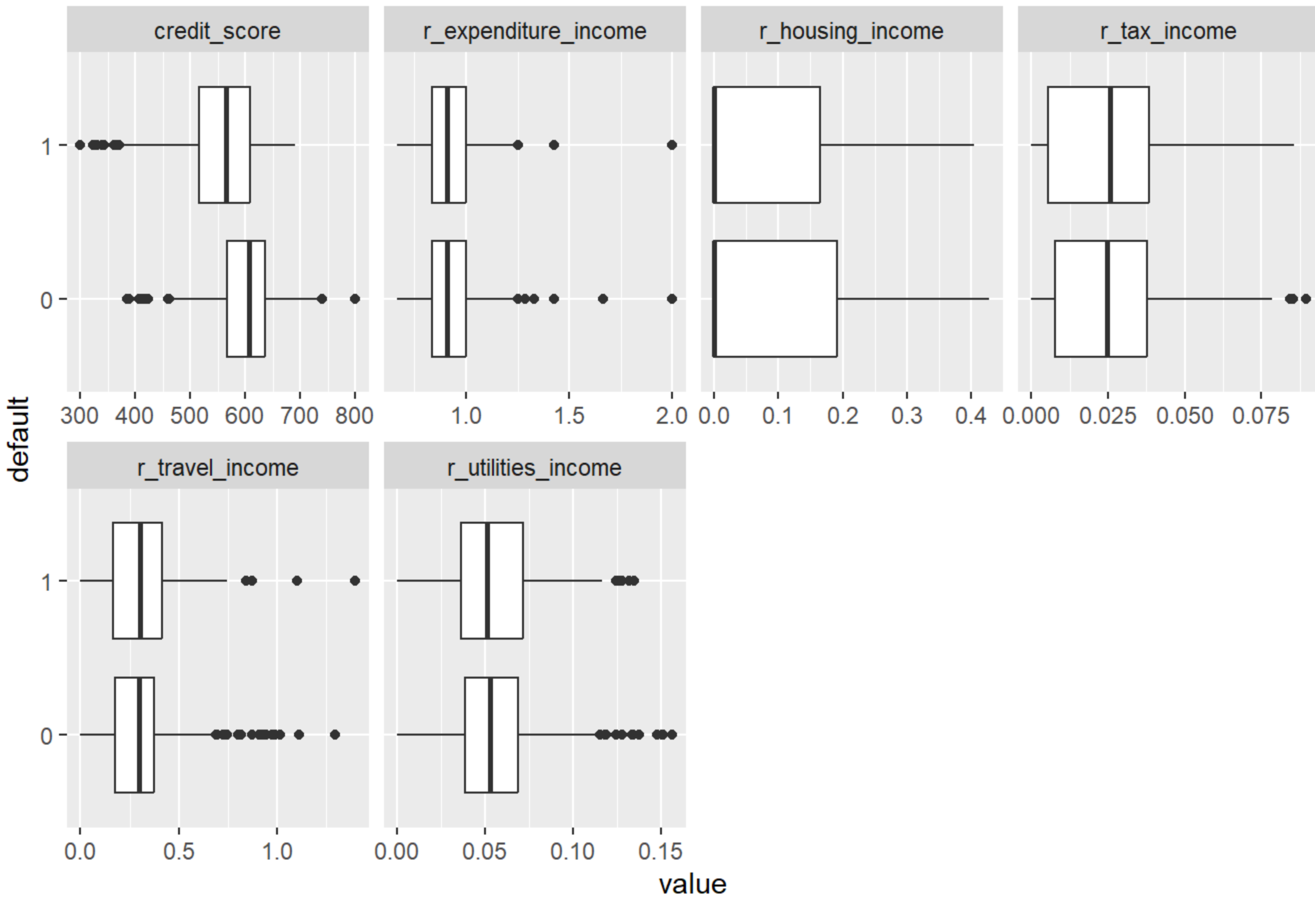
```
cat_mortgage            22 1000      0.17      0.38   0.00       1.00
cat_savings_account     23 1000      0.99      0.08   0.00       1.00
cat_dependents          24 1000      0.15      0.36   0.00       1.00
credit_score            25 1000    586.71     63.41 300.00     800.00
                             range       se
default                       -Inf       NA
income                   662094.00  3596.04
savings                 2911863.00 14006.23
debt                    5968620.00 31046.94
r_savings_income             16.11     0.13
r_debt_income                37.00     0.18
r_clothing_income             0.25     0.00
r_education_income            0.53     0.00
r_entertainment_income        1.40     0.00
r_fines_income                0.03     0.00
r_gambling_income             0.21     0.00
r_groceries_income            0.56     0.00
r_health_income               0.40     0.00
r_housing_income              0.43     0.00
r_tax_income                  0.09     0.00
r_travel_income               1.40     0.01
r_utilities_income            0.16     0.00
r_expenditure_income          1.33     0.01
cat_gambling                  -Inf       NA
cat_debt                      1.00     0.01
cat_credit_card               1.00     0.01
cat_mortgage                  1.00     0.01
cat_savings_account           1.00     0.00
cat_dependents                1.00     0.01
credit_score                500.00     2.01
```

Check summary statistics and variable distributions


debt
income
r_clothing_income
r_debt_income
r_education_income
r_entertainment_income
r_fines_income
r_gambling_income
r_groceries_income
r_health_income
r_housing_income
r_savings_income
r_tax_income
r_travel_income
r_utilities_income
savings
Frequency
value


credit_score
r_expenditure_income
Frequency
150
100
50
0
300
200
100
0
300
400
500
600
700
800
1.0
1.5
2.0
value


debt
income
r_clothing_income
r_debt_income
r_education_income
r_entertainment_income
r_fines_income
r_gambling_income
r_groceries_income
r_health_income
r_savings_income
savings
default
value



Check for outliers

	variables	outliers_cnt	outliers_ratio	outliers_mean
1	income	25	2.5	464058.840000000
2	savings	28	2.8	1965481.785714286
3	debt	44	4.4	3736178.000000000
4	r_savings_income	21	2.1	14.974652381
5	r_debt_income	35	3.5	25.930257143
6	r_clothing_income	53	5.3	0.163069811
7	r_education_income	108	10.8	0.216386111
8	r_entertainment_income	50	5.0	0.603290000
9	r_fines_income	99	9.9	0.002939394
10	r_gambling_income	160	16.0	0.084101250
11	r_groceries_income	37	3.7	0.415181081
12	r_health_income	66	6.6	0.174146970
13	r_housing_income	0	0.0	NaN
14	r_tax_income	3	0.3	0.086933333
15	r_travel_income	28	2.8	0.897789286
16	r_utilities_income	17	1.7	0.133047059
17	r_expenditure_income	26	2.6	1.579546154
18	cat_debt	56	5.6	0.000000000
19	cat_credit_card	236	23.6	1.000000000
20	cat_mortgage	173	17.3	1.000000000
21	cat_savings_account	7	0.7	0.000000000

```
22        cat_dependents          150           15.0       1.000000000
23          credit_score           34            3.4     402.147058824
        with_mean     without_mean
1  121610.0190000 112829.280000000
2  413189.5970000 368473.361111111
3  790718.0450000 655152.942468619
4       4.0634772      3.829427477
5       6.0684492      5.348072746
6       0.0555572      0.049540127
7       0.0386945      0.017180269
8       0.1675136      0.144578000
9       0.0002910      0.000000000
10      0.0184709      0.005969881
11      0.1564751      0.146535202
12      0.0523004      0.043690257
13      0.0926080      0.092608000
14      0.0250889      0.024902808
15      0.2828336      0.265118827
16      0.0546550      0.053299288
17      0.9436065      0.926630698
18      0.9440000      1.000000000
19      0.2360000      0.000000000
20      0.1730000      0.000000000
21      0.9930000      1.000000000
22      0.1500000      0.000000000
23    586.7120000    593.208074534
```

It is important to not mess with any of the outliers in this dataset. Outliers can be used as a clear example of whether or not the person defaults.

DATA PREPARATION

Because there are no missing values and the outliers do not need fixed, we can move past data preparation.

MODELING AND EVALUATION

MODEL 1: KNN

Prepare Data

Partition

Partition 60/40 and check proportions

```
Full Dataset
```

```
    0     1
0.716 0.284
```

```
Train Dataset
```

```
        0         1
0.7154742 0.2845258
```

```
Test Dataset
```

```
       0        1
0.716792 0.283208
```

KNN Model

```
k-Nearest Neighbors

601 samples
 23 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 540, 541, 541, 541, 541, 541, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   1  0.5973224  0.010716302
   2  0.6006011  0.008154099
   3  0.6505464  0.048706043
   4  0.6839617  0.135998130
   5  0.6805191  0.089335111
   6  0.6855464  0.114772434
   7  0.6872131  0.073551120
   8  0.7021585  0.108902558
   9  0.7054918  0.096365643
  10  0.7187432  0.155356903

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 10.
```

Running the knn model with our cleaned data gives a k value of 10, meaning that we will use 10 neighbors to determine the classification.

Performance Metrics at Default Cutoff

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 269  96
         1  17  17

               Accuracy : 0.7168
                 95% CI : (0.6698, 0.7605)
    No Information Rate : 0.7168
    P-Value [Acc > NIR] : 0.5253

                  Kappa : 0.1154

 Mcnemar's Test P-Value : 0.0000000000002174

            Sensitivity : 0.15044
            Specificity : 0.94056
         Pos Pred Value : 0.50000
         Neg Pred Value : 0.73699
             Prevalence : 0.28321
         Detection Rate : 0.04261
   Detection Prevalence : 0.08521
      Balanced Accuracy : 0.54550

       'Positive' Class : 1


F1 Score:  0.2312925
```

Using a default cutoff shows an unbalanced specificity and sensitivity. Using a threshold-tuned cutoff will allow us to have a more balanced model that will be more useful.

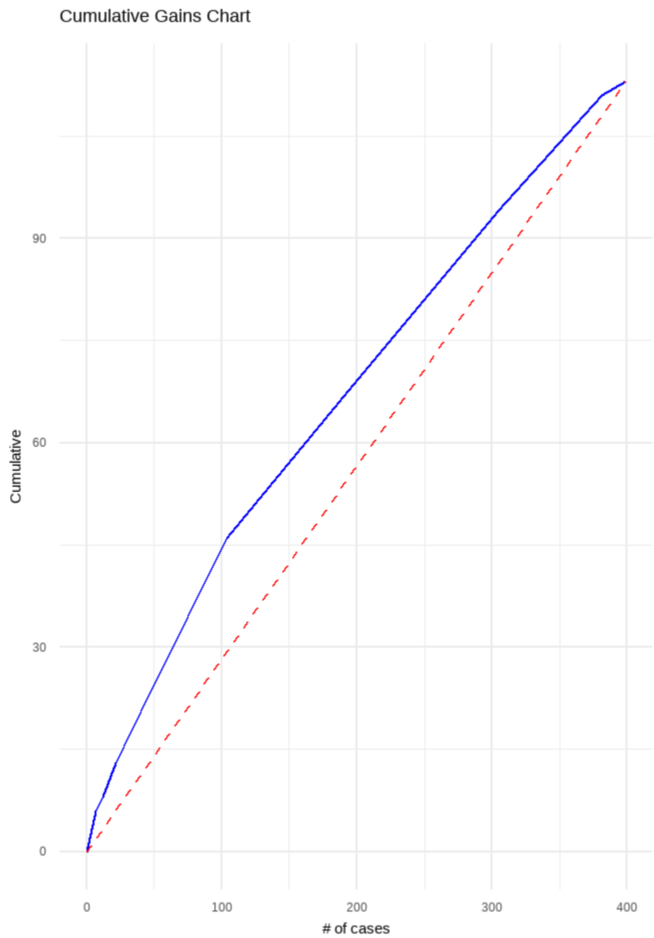
Performance Metrics at Threshold-Tuned Cutoff

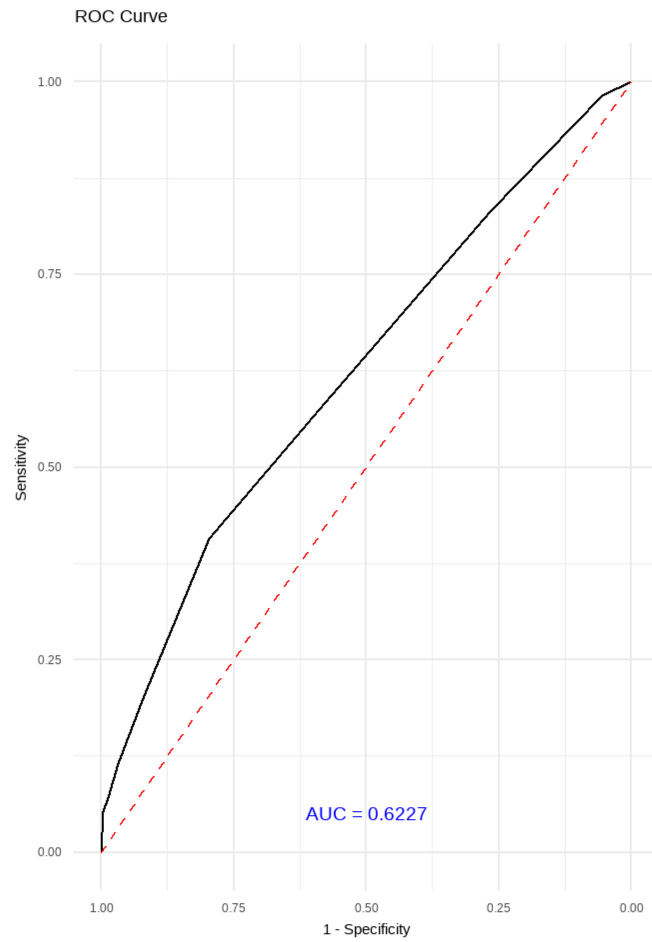
```
Optimal Cutoff

  threshold
1      0.35

CONFUSION MATRIX AT OPTIMAL CUTOFF VALUE OF: 0.35

Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 228  67
         1  58  46
```

```
               Accuracy : 0.6867
                 95% CI : (0.6387, 0.7319)
    No Information Rate : 0.7168
    P-Value [Acc > NIR] : 0.9165

                  Kappa : 0.2093

 Mcnemar's Test P-Value : 0.4743

            Sensitivity : 0.4071
            Specificity : 0.7972
         Pos Pred Value : 0.4423
         Neg Pred Value : 0.7729
             Prevalence : 0.2832
         Detection Rate : 0.1153
   Detection Prevalence : 0.2607
      Balanced Accuracy : 0.6021

       'Positive' Class : 1

F1 Score:  0.4239631
```

Using the threshold-tuned cutoff, we are able to achieve a more balanced sensitivity and specificity of 40.71% and 79.72%. This is much better than the default cutoff. This allowed us to also improve our F1 score, which measures the balance of the precision and recall. While still being low and a poor performing model, it is improvement from the default cutoff.

Gains Chart and ROC Curve with AUC





The cumulative gains chart performs okay. Our lift is higher than the reference line, but it is not a strong lift. The ROC curve also performs decent as it gives and area under the curve of 62.27%, which is better than random probability. These charts show us that our model is performing alright, but it could use some improvement.

LOGISTIC REGRESSION MODEL

Prepare Data

	variables	outliers_cnt	outliers_ratio	outliers_mean	with_mean
1	r_savings_income	21	2.1	14.974652381	4.0634772
2	r_debt_income	35	3.5	25.930257143	6.0684492
3	r_clothing_income	53	5.3	0.163069811	0.0555572
4	r_education_income	108	10.8	0.216386111	0.0386945
5	r_entertainment_income	50	5.0	0.603290000	0.1675136
6	r_fines_income	99	9.9	0.002939394	0.0002910
7	r_gambling_income	160	16.0	0.084101250	0.0184709
8	r_groceries_income	37	3.7	0.415181081	0.1564751
9	r_health_income	66	6.6	0.174146970	0.0523004
10	r_housing_income	0	0.0	NaN	0.0926080
11	r_tax_income	3	0.3	0.086933333	0.0250889
12	r_travel_income	28	2.8	0.897789286	0.2828336
13	r_utilities_income	17	1.7	0.133047059	0.0546550

```
14   r_expenditure_income           26            2.6   1.579546154   0.9436065
15               cat_debt           56            5.6   0.000000000   0.9440000
16        cat_credit_card          236           23.6   1.000000000   0.2360000
17           cat_mortgage          173           17.3   1.000000000   0.1730000
18    cat_savings_account            7            0.7   0.000000000   0.9930000
19         cat_dependents          150           15.0   1.000000000   0.1500000
20           credit_score           34            3.4 402.147058824 586.7120000
21              logincome           51            5.1   0.145184271  10.7026882
22             logsavings           28            2.8   5.220401108  11.9068116
23                logdebt           56            5.6   0.000099995  11.9089361
    without_mean
1    3.829427477
2    5.348072746
3    0.049540127
4    0.017180269
5    0.144578000
6    0.000000000
7    0.005969881
8    0.146535202
9    0.043690257
10   0.092608000
11   0.024902808
12   0.265118827
13   0.053299288
14   0.926630698
15   1.000000000
16   0.000000000
17   0.000000000
18   1.000000000
19   0.000000000
20 593.208074534
21  11.270056685
22  12.099424249
23  12.615392446
```

The data preparation for the logistic regression model is more work than the KNN model. In order to reduce outliers and improve model effectiveness, we needed to take the log of some of the variables. Doing this reduced the outliers and will hopefully help in classifying the right clientele.

Partition

Partition 60/40 and check proportions

LR Model

Model Summary

```
Call:
NULL
```

```
Coefficients:
                         Estimate Std. Error z value  Pr(>|z|)    
(Intercept)              8.991586   2.887128   3.114   0.00184 ** 
r_savings_income         0.128917   0.063004   2.046   0.04074 *  
r_debt_income           -0.025960   0.043899  -0.591   0.55428    
r_clothing_income        4.491384   3.386473   1.326   0.18475    
r_education_income      -3.778179   1.953406  -1.934   0.05309 .  
r_entertainment_income  -2.354311   1.775298  -1.326   0.18479    
r_fines_income         -69.349248  76.565912  -0.906   0.36507    
r_gambling_income       -5.906648   5.646063  -1.046   0.29549    
r_groceries_income      -0.621242   3.235273  -0.192   0.84773    
r_health_income        -10.762271   4.301395  -2.502   0.01235 *  
r_housing_income        -1.959616   1.549732  -1.264   0.20606    
r_tax_income            -1.961339  10.766509  -0.182   0.85545    
r_travel_income         -0.970425   1.226901  -0.791   0.42897    
r_utilities_income       5.407117   9.777998   0.553   0.58027    
r_expenditure_income     0.338344   1.314192   0.257   0.79683    
cat_gamblingLow          0.410112   0.478731   0.857   0.39163    
cat_gamblingNo          -0.040319   0.403948  -0.100   0.92049    
cat_debt                -2.127669   1.928694  -1.103   0.26996    
cat_credit_card          0.350734   0.280516   1.250   0.21118    
cat_mortgage             0.546217   0.319541   1.709   0.08738 .  
cat_savings_account      3.580941   1.858588   1.927   0.05402 .  
cat_dependents          -0.525381   0.556531  -0.944   0.34516    
credit_score            -0.015399   0.003620  -4.253 0.0000211 ***
logincome               -0.002898   0.054633  -0.053   0.95769    
logsavings              -0.276409   0.161466  -1.712   0.08692 .  
logdebt                  0.101831   0.168544   0.604   0.54572    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 717.81  on 600  degrees of freedom
Residual deviance: 620.22  on 575  degrees of freedom
AIC: 672.22

Number of Fisher Scoring iterations: 4
                            GVIF Df GVIF^(1/(2*Df))
r_savings_income        5.876501  1        2.424150
r_debt_income           6.863141  1        2.619760
r_clothing_income       1.747696  1        1.322005
r_education_income      2.110874  1        1.452885
r_entertainment_income  4.861850  1        2.204960
r_fines_income          1.060272  1        1.029695
r_gambling_income       3.147495  1        1.774118
r_groceries_income      9.149869  1        3.024875
r_health_income         2.805334  1        1.674913
r_housing_income        2.975224  1        1.724884
```

```
r_tax_income            4.148744  1        2.036847
r_travel_income         5.999295  1        2.449346
r_utilities_income      6.663135  1        2.581305
r_expenditure_income    5.251266  1        2.291564
cat_gambling            4.764371  2        1.477411
cat_debt               17.874537  1        4.227829
cat_credit_card         1.665224  1        1.290436
cat_mortgage            1.654423  1        1.286244
cat_savings_account     2.967366  1        1.722604
cat_dependents          4.043635  1        2.010879
credit_score            4.734248  1        2.175833
logincome               2.856119  1        1.690006
logsavings             12.050272  1        3.471350
logdebt                30.730600  1        5.543519
```

After initially running this model, credit_score stands out as expected to be a statistically significant predictor. Looking at the VIF, there are quite a fiew variables that are showing signs of multicollinearity. We can run this model again and remove some of the variables that are showing multicollinearity as well as credit_score to see what other variables might be significant behind the scenes.

```
Call:
NULL

Coefficients:
                        Estimate Std. Error z value     Pr(>|z|)    
(Intercept)             -1.88775    1.53507  -1.230       0.2188    
r_savings_income         0.05485    0.04550   1.206       0.2280    
r_debt_income            0.12972    0.02400   5.405 0.0000000647 ***
r_clothing_income        4.19050    2.99245   1.400       0.1614    
r_education_income      -4.26752    1.89522  -2.252       0.0243 *  
r_entertainment_income  -0.60466    1.41135  -0.428       0.6683    
r_fines_income         -67.00081   78.28221  -0.856       0.3921    
r_gambling_income       -6.32357    5.42054  -1.167       0.2434    
r_health_income         -8.05290    4.06048  -1.983       0.0473 *  
r_housing_income        -2.22866    1.40064  -1.591       0.1116    
r_tax_income            -9.33573    8.80820  -1.060       0.2892    
r_travel_income         -0.91351    1.07803  -0.847       0.3968    
r_utilities_income      -0.26601    7.13849  -0.037       0.9703    
r_expenditure_income     1.02002    1.12019   0.911       0.3625    
cat_gamblingLow          0.12029    0.46742   0.257       0.7969    
cat_gamblingNo          -0.31588    0.38959  -0.811       0.4175    
cat_credit_card          0.35374    0.25778   1.372       0.1700    
cat_mortgage             0.44230    0.30489   1.451       0.1469    
cat_savings_account      0.87441    1.14056   0.767       0.4433    
cat_dependents          -0.53988    0.40967  -1.318       0.1876    
logincome               -0.05352    0.03769  -1.420       0.1556    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 717.81  on 600  degrees of freedom
Residual deviance: 645.32  on 580  degrees of freedom
AIC: 687.32

Number of Fisher Scoring iterations: 4
```

Removing any variables with a VIF over 3 as well as credit_score gave us interesting results. This introduced new variables as significant that weren’t significant previously like the ratio of debt to income. This is good to know moving forward that these can have an impact outside of credit score. Moving into the odds ratio though, we cannot leave out credit_score, as it is the biggest predictor of default.

Coefficients as Odds Ratios

```
                             (Intercept)                         r_savings_income
1947.15474228333982864569406956434249880     1.0539188282685101327729171316605061
                             r_debt_income                       r_clothing_income
   0.9911996951844413983323534012015443   21.3323291792970515245997376041486859
                         r_education_income                  r_entertainment_income
   0.0215215547052686681506195043311891    0.5930990477234290292329887961386703
                            r_fines_income                       r_gambling_income
   0.0000000000000000000000000001534841    0.0006419648266865438405656685283418
                           r_health_income                        r_housing_income
   0.0000216274801893783705231122382884    0.1580723583929292408445377304815338
                              r_tax_income                        r_travel_income
   0.0018623492360325369539275630614839    0.4452781133950500236373670759348897
                        r_utilities_income                    r_expenditure_income
  15.9159801553841457888438526424579322    1.8901901012886122011025236133718863
                           cat_gamblingLow                          cat_gamblingNo
   1.3954356577287312379809236517758109    0.9210408664153727498202783863234799
                           cat_credit_card                            cat_mortgage
   1.3448954519679503505358297843486071    1.5974172525581409320949433094938286
                       cat_savings_account                          cat_dependents
   2.8994302267413960549902185448445380    0.5829257555546388802625301650550682
                              credit_score                               logincome
   0.9850762599450013645707713294541463    0.9528231012306567215830455097602680
```

credit_score: The odds ratio of 0.9850763 indicates that with every one unit increase in credit score, the odds of defaulting goes down by 1.492374 percent. This makes sense as individuals with higher credit scores are generally less likely to default.

r_debt_income: The odds ratio of 0.9911997 indicates that with every one unit increase in the ratio of debt to income, the odds of defaulting goes down by 0.8800305 percent. This is an interesting observation, as it would seem that someone with more debt would be more likely to default.

r_savings_income: The odds ratio of r_savings_income 1.0539188 means that for every one-unit increase in the savings-to-income ratio, the odds of defaulting increase by about 5.3918828 percent. This suggests that higher savings relative to income could be associated with increased odds of default.

Fit Metrics

```
McFadden
  0.1285

Nagelkerke
    0.2042
```

The McFadden R2 suggests the model explains about 12.85 percent of the variance, while the Nagelkerke R2, a more adjusted measure, indicates the model explains roughly 20.42 percent of the variation to the dependent variable. The Nagelkerke specifically indicates that we have a moderate performing model.

Performance Metrics at Default Cutoff

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 271  86
         1  15  27
                                          
               Accuracy : 0.7469          
                 95% CI : (0.7012, 0.7888)
    No Information Rate : 0.7168          
    P-Value [Acc > NIR] : 0.09969         
                                          
                  Kappa : 0.2302          
                                          
 Mcnemar's Test P-Value : 0.000000000003278
                                          
            Sensitivity : 0.9476          
            Specificity : 0.2389          
         Pos Pred Value : 0.7591          
         Neg Pred Value : 0.6429          
             Prevalence : 0.7168          
         Detection Rate : 0.6792          
   Detection Prevalence : 0.8947          
      Balanced Accuracy : 0.5932          
                                          
       'Positive' Class : 0               
                                          

F1 Score:  0.8429238
```

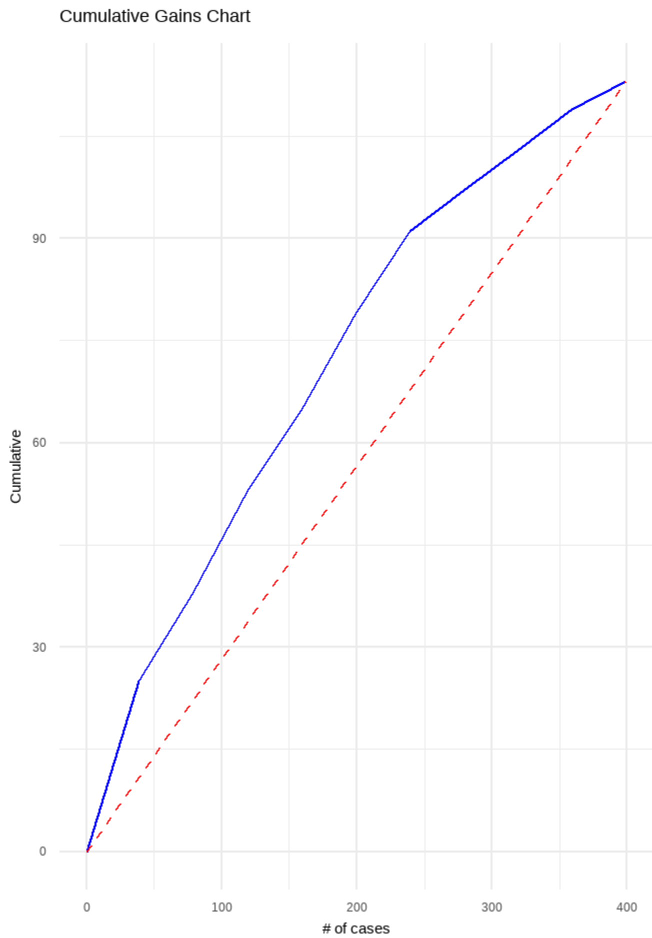
The logistic regression model achieves a pretty good accuracy and demonstrates high sensitivity, meaning it effectively identifies non-defaulters. However, its low specificity indicates it struggles to correctly classify defaulters, leading to a high rate of false negatives. This imbalance is reflected in a moderate Kappa statistic, highlighting limited agreement beyond chance. To improve performance, threshold tuning can be applied to achieve a better balance between sensitivity and specificity, addressing the trade-offs in misclassification.

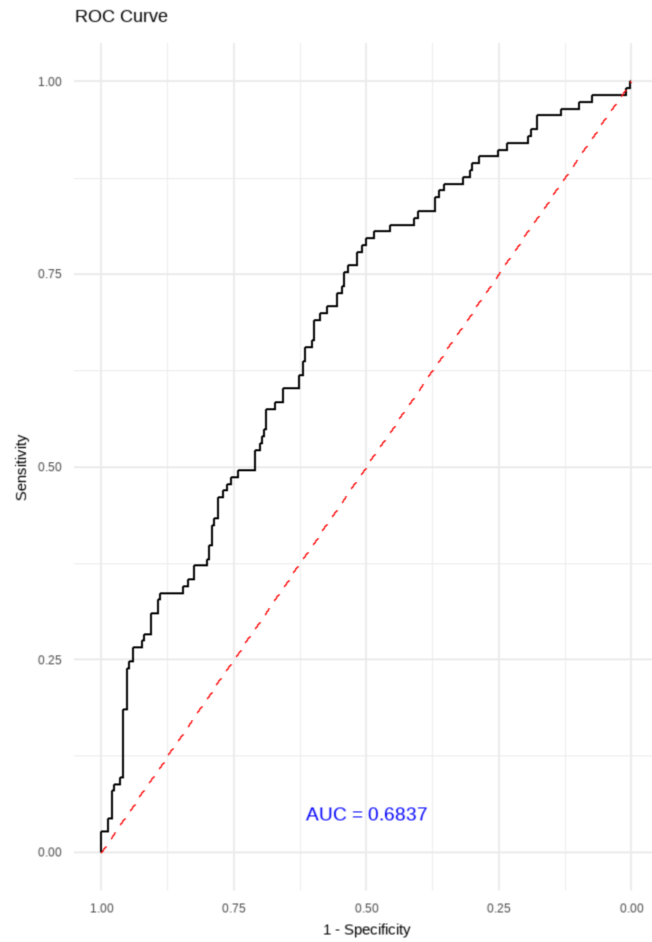
Performance Metrics at Threshold-Tuned Cutoff

```
Optimal Cutoff:  0.2143983

CONFUSION MATRIX AT OPTIMAL CUTOFF VALUE OF: 0.2143983

Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 143  23
         1 143  90

               Accuracy : 0.584
                 95% CI : (0.5339, 0.6328)
    No Information Rate : 0.7168
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2244

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.7965
            Specificity : 0.5000
         Pos Pred Value : 0.3863
         Neg Pred Value : 0.8614
             Prevalence : 0.2832
         Detection Rate : 0.2256
   Detection Prevalence : 0.5840
      Balanced Accuracy : 0.6482

       'Positive' Class : 1


F1 Score (Optimal):  0.5202312
```

The threshold value of 0.2143983 allows us to achieve a more balanced model. This make our sensitivity and specificity more balanced. Our optimal F1 score is 0.5202312 which is better than the KNN model.

Gains Chart and ROC Curve with AUC





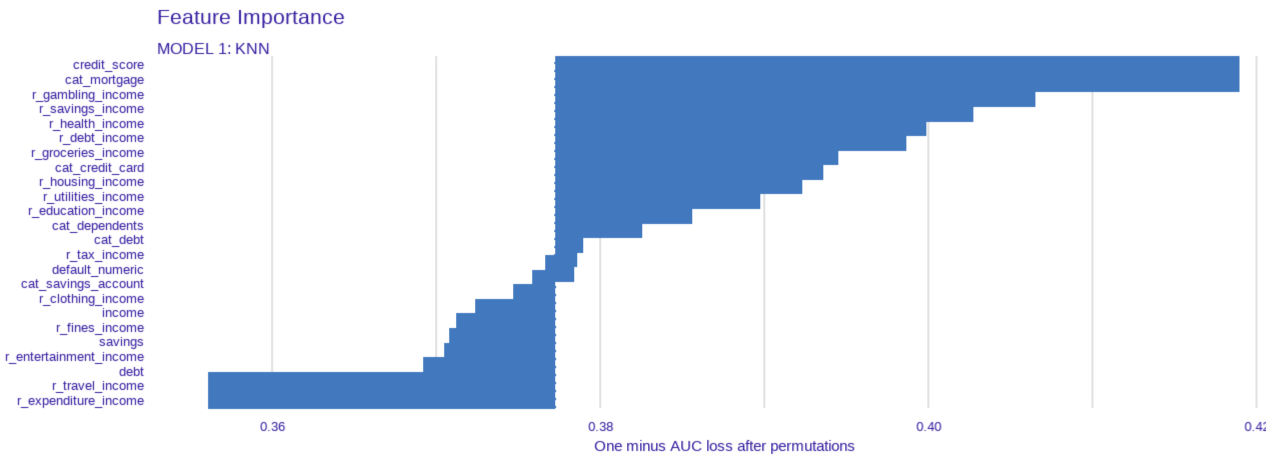
The gains chart and ROC curve chart are similar to the KNN model. Now our model predicts the employee performance 68.3674732 percent of the time.

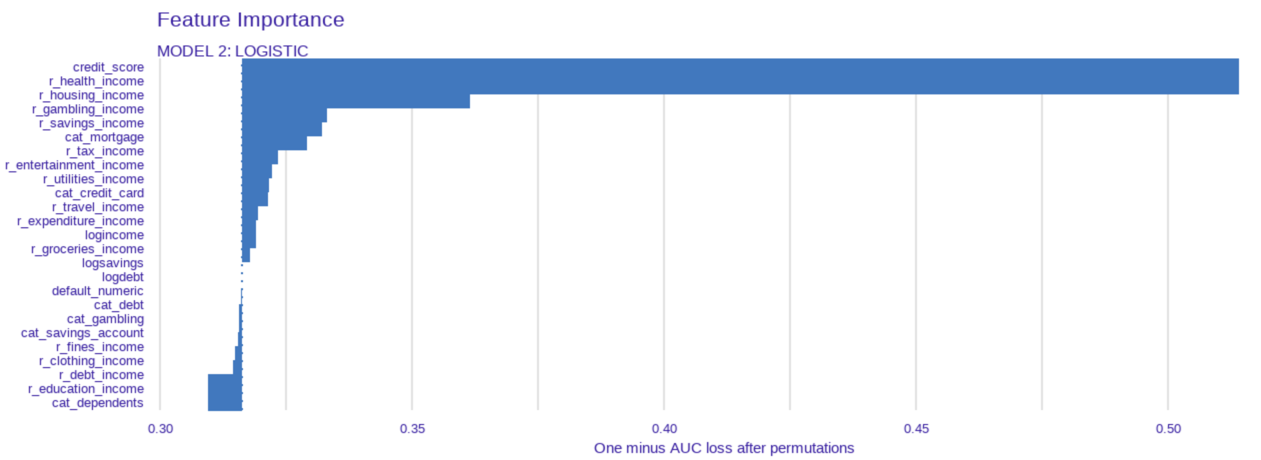
COMPARISON ACROSS MODELS

```
Preparation of a new explainer is initiated
  -> model label       :  MODEL 1: KNN
  -> data              :  399  rows  24  cols
  -> target variable   :  399  values
  -> predict function  :  yhat.train  will be used (  default  )
  -> predicted values  :  No value for predict function target column. (  default  )
  -> model_info        :  package caret , ver. 6.0.94 , task classification (  default  )
  -> predicted values  :  numerical, min =  0 , mean =  0.266416 , max =  0.8
  -> residual function :  difference between y and yhat (  default  )
  -> residuals         :  numerical, min =  -0.8 , mean =  0.01679198 , max =  1
  A new explainer has been created!

Preparation of a new explainer is initiated
  -> model label       :  MODEL 2: LOGISTIC
  -> data              :  399  rows  25  cols
  -> target variable   :  399  values
  -> predict function  :  yhat.glm  will be used (  default  )
  -> predicted values  :  No value for predict function target column. (  default  )
```

```
-> model_info        :  package stats , ver. 4.3.1 , task classification (  default  )
-> predicted values  :  numerical, min =  0.02075153 , mean =  0.2803786 , max =  0.9383668
-> residual function :  difference between y and yhat (  default  )
-> residuals         :  numerical, min =  -0.8997604 , mean =  0.002829429 , max =  0.9545483
A new explainer has been created!
```

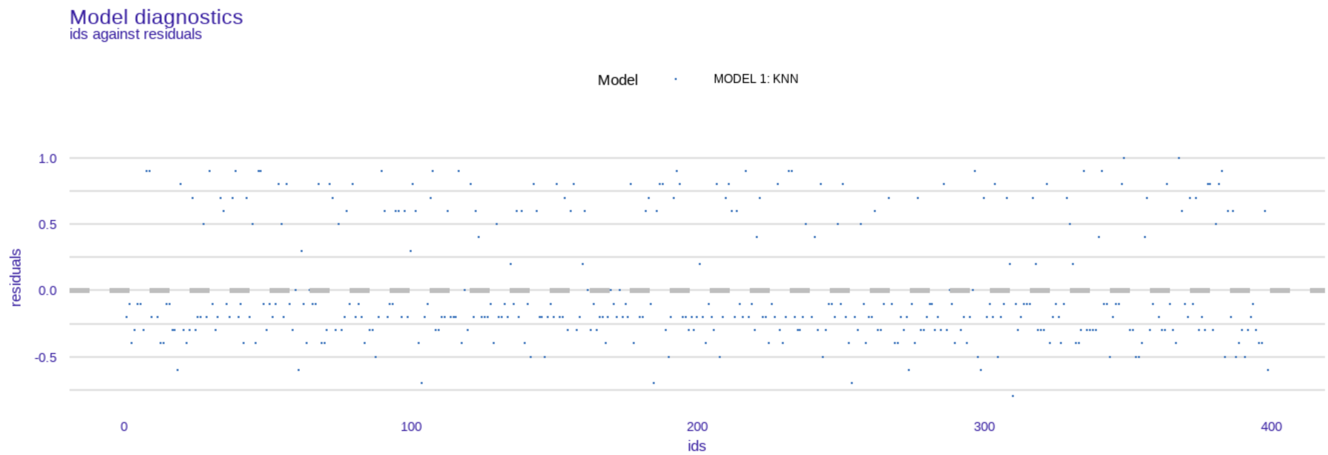
Variable Importance

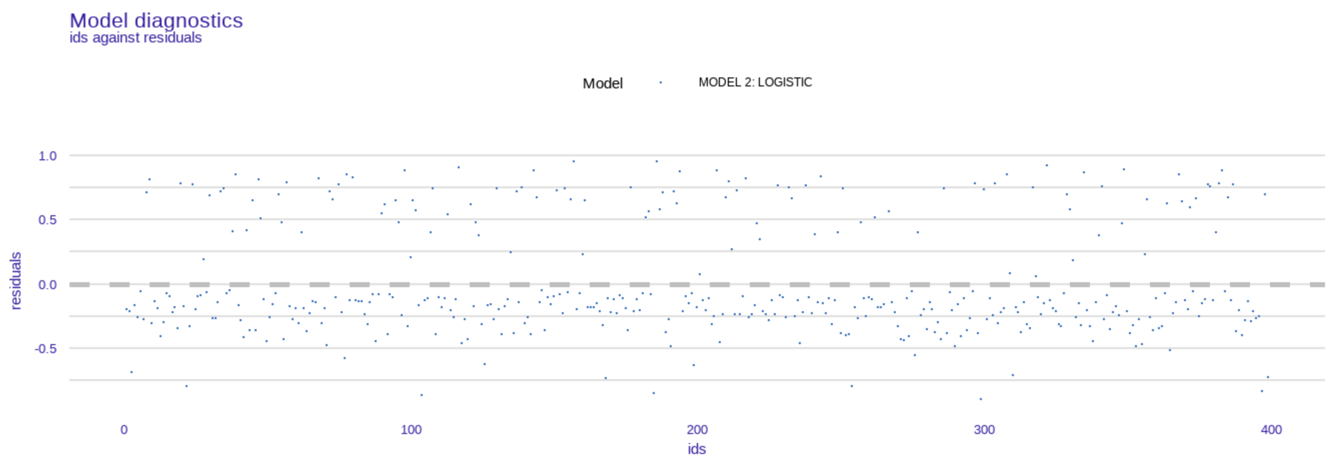




Both of our models show credit_score as the most important variable. After that, the rankings change significantly. KNN shows the categorical mortgage and the ratio of savings to income as the next top variables. The logistic model shows a cluster of the ratio measures as the next important variables.

Residuals





The residuals for both the KNN and logistic regression models are generally clustered around 0, indicating reasonable predictive performance, though some variability is present, particularly in the KNN model. Logistic regression appears to show slightly less dispersion in residuals, suggesting it may provide more consistent predictions.

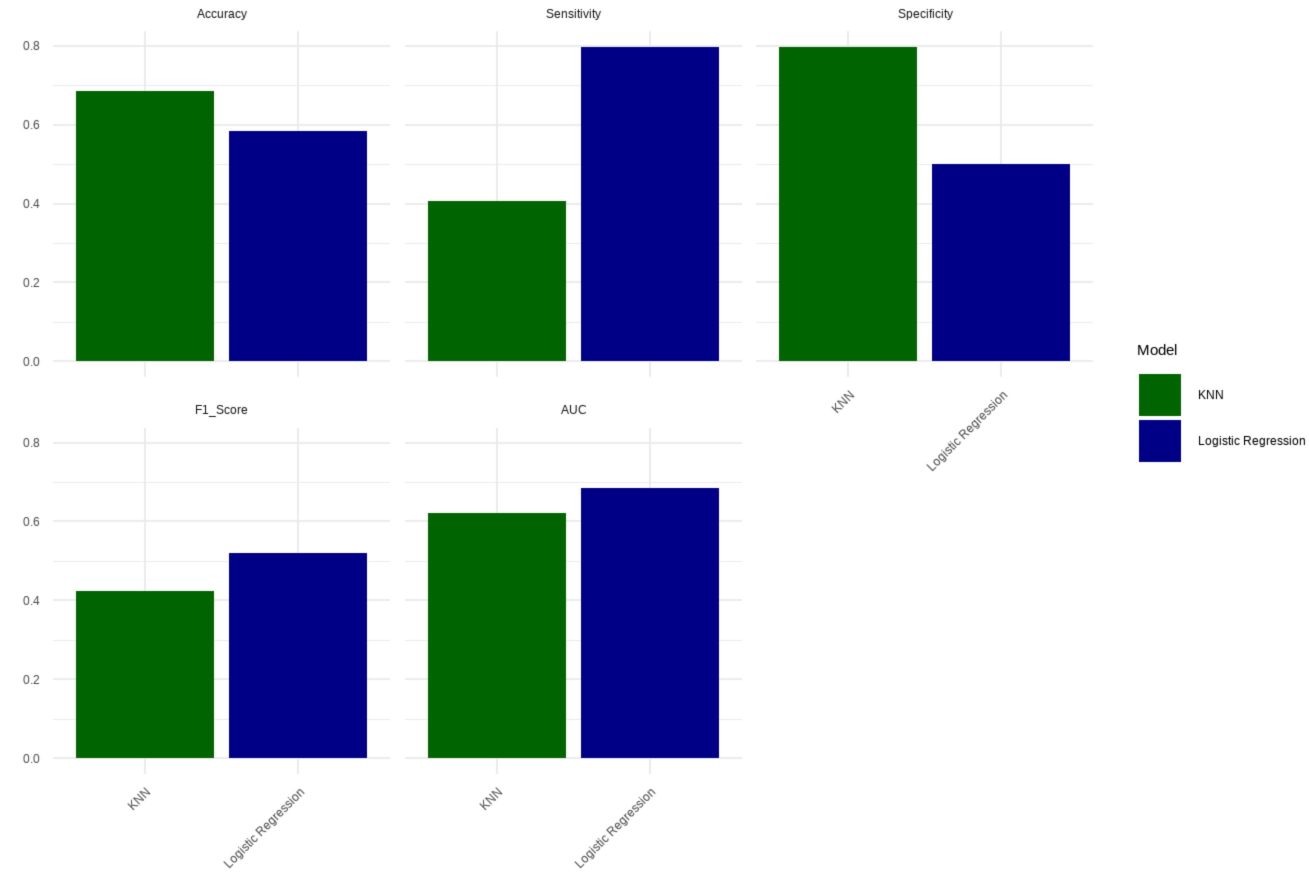
Performance Metrics

Table of Threshold-Tuned Metrics

```
                Model  Accuracy Sensitivity Specificity       AUC  F1_Score
1                 KNN 0.6867168   0.4070796   0.7972028 0.6227489 0.4239631
2 Logistic Regression 0.5839599   0.7964602   0.5000000 0.6836747 0.5202312
```

When selecting the right model, we first need to clarify whether we should prioritize preventing false negatives or false positives. In our case, we should try and prevent false negatives, as we do not want to assume someone will not default and they end up doing the opposite. Preventing false negatives means that we need to choose the model that has the higher sensitivity, which in our case, is the logistic regression model. This is a good choice as it has more balance than the KNN and a higher area under the curve.

Chart of Threshold-Tuned Metrics



Both models have their strengths, but logistic has higher results in the areas that we need.

Identify your Best, Final Model

After evaluating both the KNN and Logistic Regression models, the Logistic Regression model stands out as the best option for predicting credit defaults. The model provides clear performance metrics that demonstrate its ability to accurately distinguish between customers who are likely to default and those who are not. Key performance indicators, such as McFadden R^2 and Nagelkerke R^2, show that the model explains a decent amount of the variance in default behavior. The model's sensitivity and specificity at the optimal cutoff indicate a more balanced prediction capability, minimizing false negatives while correctly identifying high-risk customers. The optimal F1 score strengthens the reliability of the model. The logistic regression model also offers ease in interpreting the key predictors of credit default. Based on these metrics, the logistic regression model will provide meaningful insights into customer behavior and significantly help reduce financial risk.

DEPLOYMENT

Summarize Findings

The Logistic Regression model reveals several critical factors that influence the likelihood of credit default among customers. Key predictors include credit score, the ratio of debt to income, and the ratio of savings to income. The odds ratios show that with each increase in credit score, the odds of default decrease, which aligns with the expectation that individuals with higher credit scores are less likely to default. Interestingly, the odds of default decrease with a higher debt-to-income ratio, which may suggest better financial management despite higher debt. Conversely, the odds of default increase with a higher savings-to-income ratio, possibly indicating that individuals prioritize saving over debt repayment. By focusing on these key behaviors, the institution can reduce defaults while ensuring responsible lending practices.

Business Recommendations and Suggested client actions

Based on the findings from the Logistic Regression model, several strategic recommendations can be made to improve the institution's ability to predict and mitigate credit defaults. First, the institution should focus on enhancing its credit scoring models and ensure that applicants with lower credit scores are more thoroughly scored, as these customers are at a higher risk of default. There are many variables that can be explored and it would be good to do a deep-dive on each variable and their implications. This could uncover even more useful predictors for whether a customer will default or not. By implementing these recommendations, the institution can improve its risk management and reduce financial losses while fostering responsible lending practices.

REFERENCES

R and Packages

```
R version 4.3.1 (2023-06-16 ucrt)
```

```
R Packages Used:

 [1] "DALEX"        "formattable"  "DescTools"    "car"          "carData"
 [6] "dlookr"       "summarytools" "janitor"      "rpart.plot"   "rpart"
[11] "klaR"         "MASS"         "pROC"         "gains"        "caret"
[16] "lattice"      "gridExtra"    "flextable"    "DataExplorer" "lubridate"
[21] "forcats"      "stringr"      "dplyr"        "purrr"        "readr"
[26] "tidyr"        "tibble"       "ggplot2"      "tidyverse"
```

Other References

Jaggia, S., Kelly, A., Lertwachara, K., & Chen, L. (2023). *Business analytics: Communicating with numbers* (2nd Ed.). McGraw-Hill.