Statistical Analysis on Bank_Personal_Loan_Modelling Dataset

Alessio Gaia, Barrasso Marco, Longo Andrea, Ruoppolo Emanuele, Zampar Marco

2024-02-13

ABSTRACT

In the following we show the results of the analysis of the Bank Loan dataset. Five different models have been built: logistic regression, Generalized Additive Model (GAM), Random forest, support vector machine, with linear and radial kernels, and a logistic bayesian model. First we conduct a data exploration phase, that ended by removing redundant variables, that either were highly correlated with some other variables or were not significant. This first analysis showed that the response variable, that is a binary variable, was strongly imbalanced with an imbalance ratio IR=9,42. All models were initially trained on the imbalanced dataset and then retrained on a dataset balanced by oversampling to compare performances. In order to find the best model, for each we computed the four performance indexes: accuracy, true positive rate (TPR), true negative rate (TNR), and the Area Under the Curve (AUC). We decided to compare the models with their AUC concluding that the model that better performs is the GAM one.

DATA EXPLORATION

The Bank Loan contains information about 5000 customers of a bank, for each it presents 14 variables:

- 2 nominal variables:
  - **ID**
  - **Zip code**: postal code
- 2 ordinal categorical variables:
  - **Family**: family size of the costumer, between 1-4
  - **Educaton**: education level of the costumer between 1-3
- 5 numerical variables:
  - **Age**: age of the costumer
  - **Experience**: years of experience of the customer
  - **Income**: annual income in k$
  - **CCAvg**: average credit card spending per month in k$
  - **Mortgage**: value of House Mortgage in k$
- 5 binary categorical variables:
  - **CD.Account**: indicates if the costumer has a certificate account of deposit
  - **CreditCard**: indicates if the costumer uses a credit card
  - **Online**: indicates if the costumer uses online facilities
  - **Securities.Account**: indicates if the costumer has a a securities account
  - **Personal.Loan**: indicates if the costumer joined the last personal loan campaign promoted by the bank

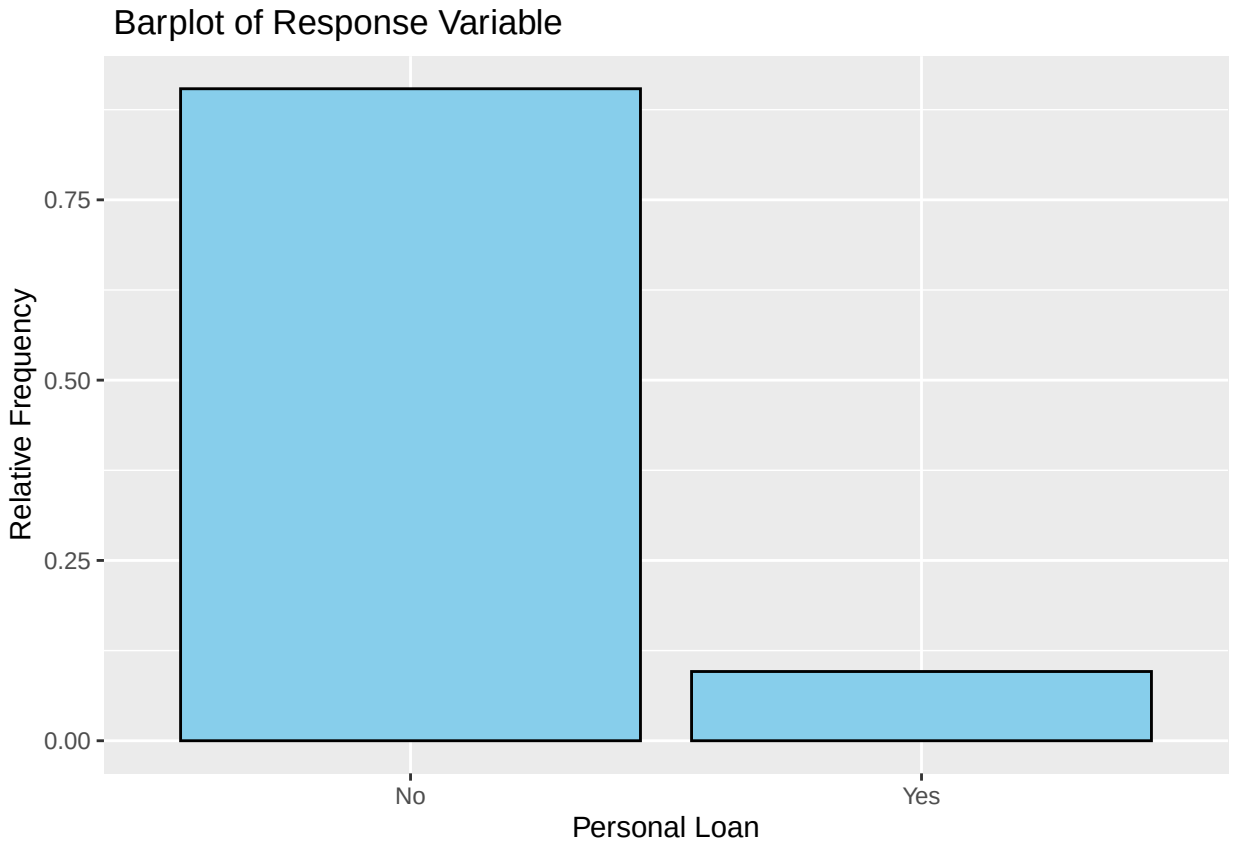
The dataset does not contain any missing value and all the models consider Personal.Loan as response variable.

In the pre-processing phase the nominal variables have been removed. The first, Id, because it does not contain any statistical information, it is simply a row index, a serial number between 1 and 5000. In addition the ZIP.Code variable has been removed because being a nominal variable it cannot be subjected to any

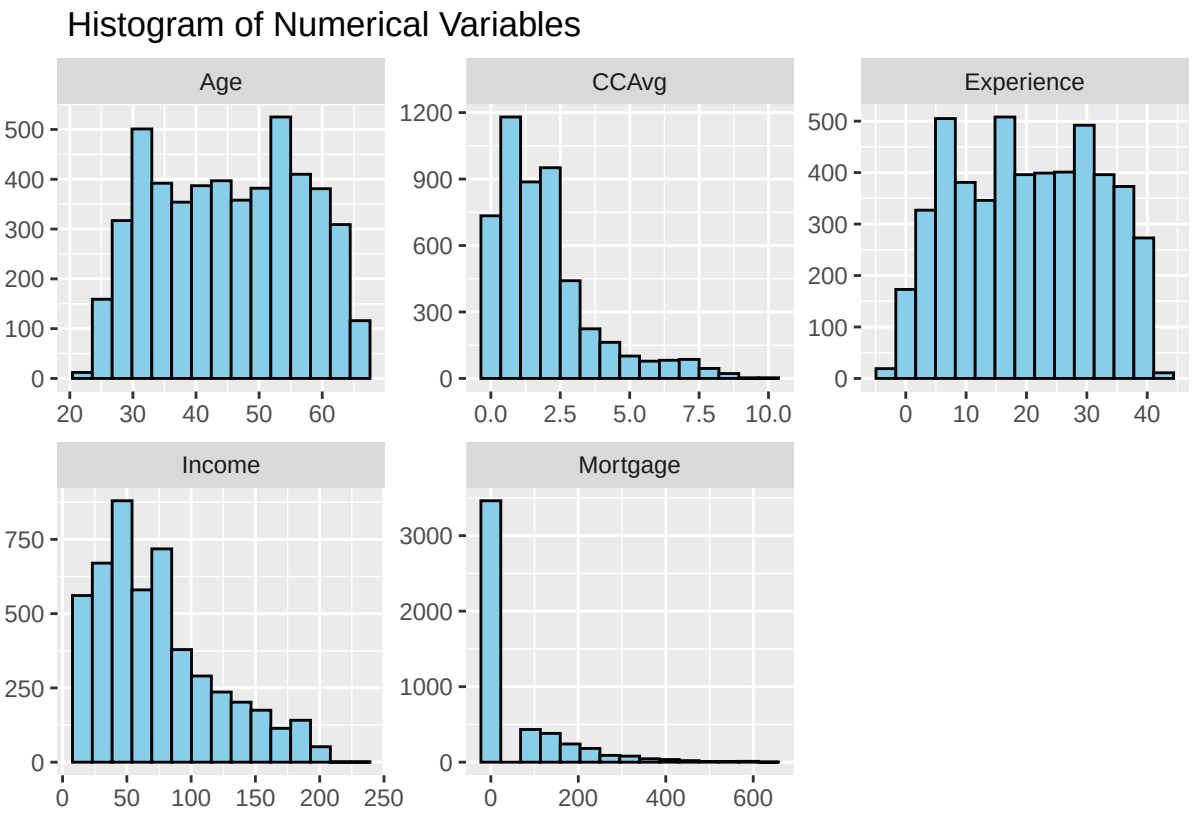
mathematical operation, such as mean or median. If used as a factor there are 467 unique values of this variable, so 467 different categories, that have been excluded to preserve the clarity of the analysis.

The following plots show the variables distributions, we see that the response variable Personal.Loan is highly imbalanced, with an imbalance ratio IR=9.42. During the analysis we'll analyze if the imbalance reduce the models' performances comparing the results with those of the same models applied on the dataset balanced with appropriate techniques. Also we see that both Age and Experience have a similar uniform distribution, while CCAvg and Income have a right skewed distribution.

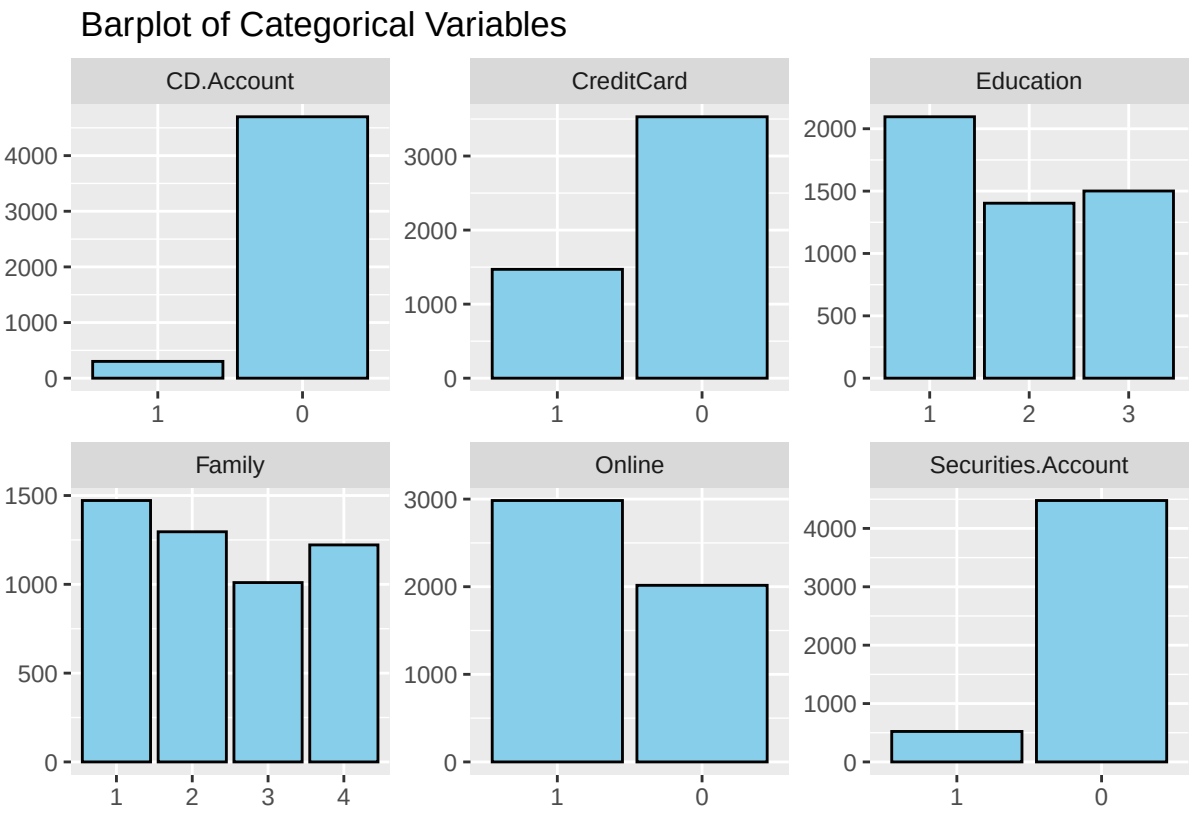
```
data %>%
  ggplot(aes(x = Personal.Loan, y = after_stat(count / sum(count)))) +
  geom_bar(fill = "skyblue", color = "black") +
  scale_x_discrete(labels = c("No", "Yes")) +
  xlab("Personal Loan") + ylab("Relative Frequency") +
  ggtitle(" Barplot of Response Variable")
```



```r
data %>%
  pivot_longer(cols = where(is.numeric)) %>%
  ggplot(aes(x = value)) +
  geom_histogram(bins = 15, fill = "skyblue", color = "black") +
  facet_wrap(~ name, scales = "free") + xlab("") + ylab("") +
  ggtitle(" Histogram of Numerical Variables")
```

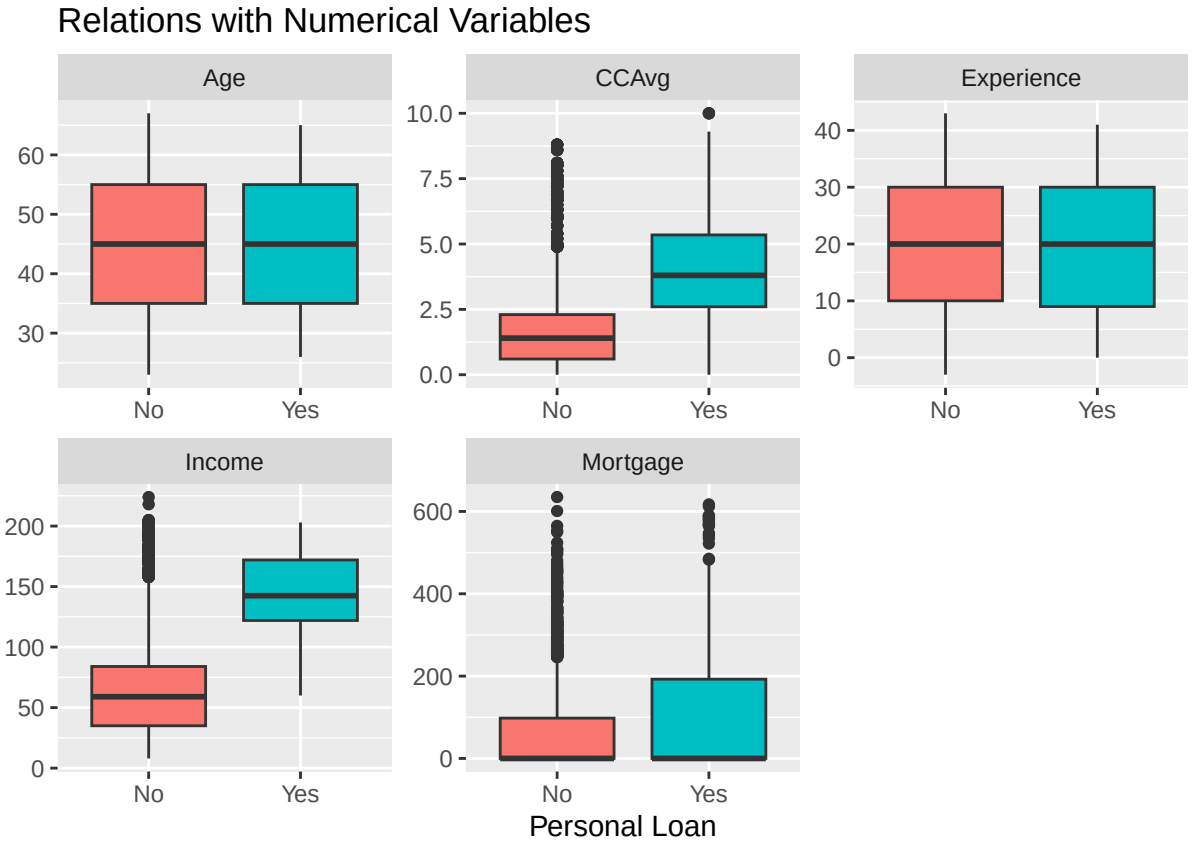


```r
data %>%
  dplyr::select(-Personal.Loan) %>%
  pivot_longer(cols = where(is.factor)) %>%
  ggplot(aes(x = value)) +
  geom_bar(fill = "skyblue", color = "black") +
  facet_wrap(~ name, scales = "free") +
  xlab("") + ylab("") +
  ggtitle(" Barplot of Categorical Variables")
```



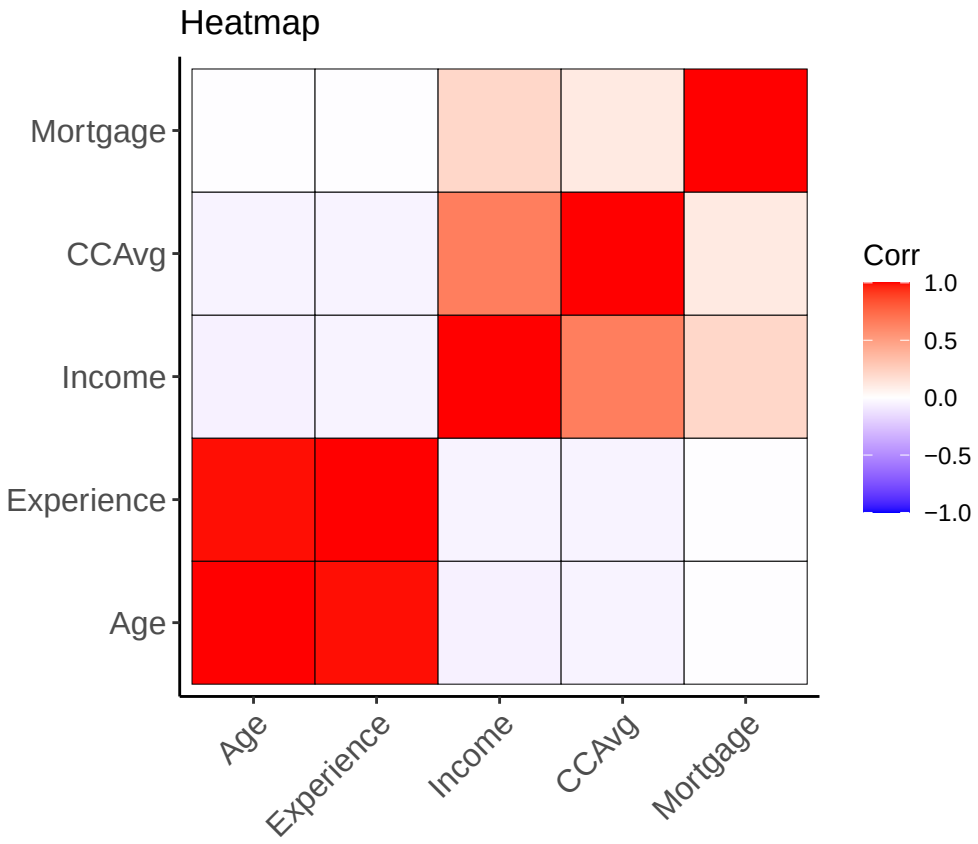
In the following box plots we can evaluate how the numerical variables are distributed with respect to the response variable. Two of the following seem to show different behaviors based on the response variable value: CCAvg and Income. Indeed the plots show that those clients that have joined the loan campaign have on average higher income and CCAvg.

```
data %>%
  pivot_longer(cols = where(is.numeric)) %>%
  ggplot(aes(x = Personal.Loan, y = value, fill = Personal.Loan)) +
  geom_boxplot() +
  facet_wrap(~ name, scales = "free")  +
  scale_x_discrete(labels = c("No", "Yes")) +
  labs(x = "Personal Loan", y = "", fill = "Personal Loan") +
  theme(legend.position = "none") + ggtitle("Relations with Numerical Variables")
```



We then looked for any correlation between the variables. The following heatmap is a representation of the correlation matrix. The first look shows that Age and Experience are highly correlated, that is clearly explainable by considering that the older the clients the higher is their experience. To avoid problems when building the models we decided to remove the Experience variable.

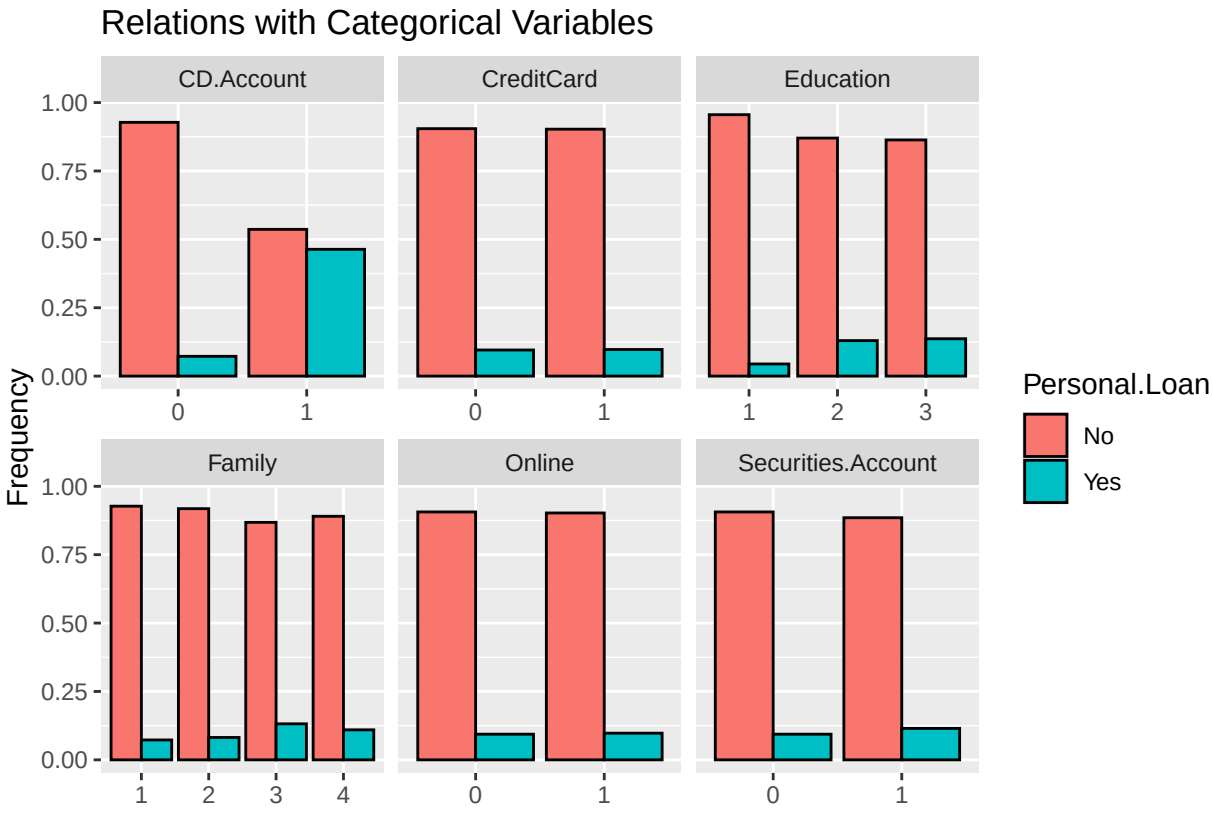
```
data %>%
  dplyr::select(where(is.numeric)) %>%
  cor() %>%
  ggcorrplot( ggtheme = theme_classic(), outline.color = "black") +
  ggtitle("Heatmap")
```



Also we can find an interesting correlation between CCAvg and Income, they could be correlated by considering that clients with higher income on average tend to spend more money. We decided to keep both considering that the models perform well without showing any problems with the standard errors.

In the last part of this analysis we show the relation between the categorical variables and the response variable. The most obvious differences are shown for the CD.Account variable, we see indeed an higher trend in joining the loan campaign for those clients with a certificate account of deposit. Moreover we see a difference based on education, those with higher education have been more inclined in joining the loan campaign.

```
data %>%
  pivot_longer(cols = c("CD.Account","Education", "Family",
                        "Securities.Account", "Online", "CreditCard")) %>%
  group_by(name, value, Personal.Loan) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  group_by(name, value) %>%
  mutate(Freq = Count / sum(Count)) %>%
  ggplot(aes(x = value, y = Freq, fill = Personal.Loan)) +
  geom_bar(stat = "identity", position = "dodge", color = "black") +
  facet_wrap(~ name, scales = "free_x") +
  scale_fill_discrete(labels = c("0" = "No", "1" = "Yes")) +
  labs(y = "Frequency") + xlab("") + ggtitle("Relations with Categorical Variables")
```



Logistic Regression

Analysis on Original Dataset

We began by fitting a logistic regression model incorporating all available variables, aiming to assess their individual contributions to predicting personal loan acceptance. This initial model served as a baseline for understanding the significance of each predictor.

```
full_model = glm(Personal.Loan ~ ., data = data, family = binomial)
summary(full_model)
```

```
##
## Call:
## glm(formula = Personal.Loan ~ ., family = binomial, data = data)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.310e+01  6.595e-01 -19.867  < 2e-16 ***
## Age                  7.867e-03  6.955e-03   1.131  0.25806
## Income               6.305e-02  3.123e-03  20.191  < 2e-16 ***
## Family2             -2.024e-01  2.336e-01  -0.867  0.38621
## Family3              1.953e+00  2.513e-01   7.772 7.70e-15 ***
## Family4              1.595e+00  2.397e-01   6.655 2.84e-11 ***
## CCAvg                1.922e-01  4.647e-02   4.135 3.54e-05 ***
## Education2           3.952e+00  2.784e-01  14.192  < 2e-16 ***
## Education3           4.109e+00  2.750e-01  14.943  < 2e-16 ***
## Mortgage             8.524e-04  6.067e-04   1.405  0.15999
## Securities.Account1 -8.566e-01  3.062e-01  -2.798  0.00515 **
## CD.Account1          3.725e+00  3.472e-01  10.729  < 2e-16 ***
## Online1             -7.232e-01  1.686e-01  -4.288 1.80e-05 ***
## CreditCard1         -9.809e-01  2.167e-01  -4.527 5.98e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3162.0  on 4999  degrees of freedom
## Residual deviance: 1131.5  on 4986  degrees of freedom
## AIC: 1159.5
##
## Number of Fisher Scoring iterations: 8
```

Observations from the initial model indicated that *Age* and *Mortgage* were not statistically significant predictors of loan acceptance. Thus we fitted another model by excluding these variables to enhance model simplicity and focus on more impactful factors.

```
model = glm(Personal.Loan ~. ,data = data2, family = binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Personal.Loan ~ ., family = binomial, data = data2)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -12.683437   0.558281 -22.719  < 2e-16 ***
## Income                0.063402   0.003111  20.378  < 2e-16 ***
## Family2              -0.186971   0.232577  -0.804  0.42145
## Family3               1.965261   0.251737   7.807 5.87e-15 ***
## Family4               1.604021   0.239795   6.689 2.24e-11 ***
## CCAvg                 0.179668   0.045953   3.910 9.24e-05 ***
## Education2            3.929341   0.276866  14.192  < 2e-16 ***
## Education3            4.075672   0.272875  14.936  < 2e-16 ***
## Securities.Account1  -0.860245   0.306122  -2.810  0.00495 **
## CD.Account1           3.757164   0.346992  10.828  < 2e-16 ***
## Online1              -0.716946   0.168291  -4.260 2.04e-05 ***
## CreditCard1          -0.982726   0.216145  -4.547 5.45e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3162.0  on 4999  degrees of freedom
## Residual deviance: 1134.6  on 4988  degrees of freedom
## AIC: 1158.6
##
## Number of Fisher Scoring iterations: 8
```

Subsequent analysis explored the inclusion of interaction terms, specifically between *Income* and *Education* and between *Income* and *Family*.

```
inter_model = glm(Personal.Loan ~. + Income:Education + Income:Family,
                  data = data2, family = binomial)
summary(inter_model)
```

```
##
## Call:
## glm(formula = Personal.Loan ~ . + Income:Education + Income:Family,
##     family = binomial, data = data2)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -3.736148   0.619120  -6.035 1.59e-09 ***
## Income               -0.012572   0.006189  -2.031 0.042215 *
## Family2               0.199341   0.847456   0.235 0.814036
## Family3              -8.731259   1.793349  -4.869 1.12e-06 ***
## Family4             -10.945952   2.005202  -5.459 4.79e-08 ***
## CCAvg                 0.489934   0.076906   6.371 1.88e-10 ***
## Education2           -9.582707   1.572594  -6.094 1.10e-09 ***
## Education3          -11.097585   1.639592  -6.769 1.30e-11 ***
## Securities.Account1  -0.888110   0.406529  -2.185 0.028917 *
## CD.Account1           3.758910   0.465702   8.071 6.94e-16 ***
## Online1              -0.860485   0.243561  -3.533 0.000411 ***
## CreditCard1          -1.197705   0.304475  -3.934 8.37e-05 ***
## Income:Education2     0.119550   0.014715   8.124 4.50e-16 ***
## Income:Education3     0.133306   0.015324   8.699  < 2e-16 ***
## Income:Family2       -0.003529   0.007433  -0.475 0.634977
## Income:Family3        0.104103   0.016769   6.208 5.37e-10 ***
## Income:Family4        0.118084   0.018172   6.498 8.13e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3162.04  on 4999  degrees of freedom
## Residual deviance:  618.81  on 4983  degrees of freedom
## AIC: 652.81
##
## Number of Fisher Scoring iterations: 9
```

The summary show that the effect of *Income* on loan acceptance is not uniform across all levels of *Education* and *Family*.

To check the contribution of each independent variable in the model we used the anova function.

```
anova(inter_model, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Personal.Loan
##
## Terms added sequentially (first to last)
##
##
##                     Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                 4999    3162.04
## Income               1  1145.81      4998    2016.23 < 2.2e-16 ***
## Family               3   279.08      4995    1737.15 < 2.2e-16 ***
## CCAvg                1     8.53      4994    1728.62 0.0034932 **
## Education            2   451.10      4992    1277.52 < 2.2e-16 ***
## Securities.Account   1     4.38      4991    1273.14 0.0363656 *
## CD.Account           1   102.34      4990    1170.80 < 2.2e-16 ***
## Online               1    13.12      4989    1157.68 0.0002915 ***
## CreditCard           1    23.03      4988    1134.64 1.593e-06 ***
## Income:Education     2   339.31      4986     795.33 < 2.2e-16 ***
## Income:Family        3   176.52      4983     618.81 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To determine the most effective model, we firstly relied on the Akaike Information Criterion (AIC) and Residual Deviance as our primary selection metrics. These criteria allowed us to balance model fit and complexity.

```
res_dev = c(full_model$deviance, model$deviance, inter_model$deviance)
as.matrix(AIC(full_model, model, inter_model) %>% mutate(res_dev = res_dev))
```

```
##             df       AIC   res_dev
## full_model  14 1159.4675 1131.4675
## model       12 1158.6434 1134.6434
## inter_model 17  652.8116  618.8116
```

Given the observation of notably high coefficients and standard errors in the model that incorporated interaction terms, we used the Variance Inflation Factor (VIF) analysis to check the presence of multicollinearity among the independent variables.

```
vif(inter_model)
```

```
##                           GVIF Df GVIF^(1/(2*Df))
## Income                3.115810  1        1.765166
## Family            15342.159662  3        4.984800
## CCAvg                 2.029217  1        1.424506
## Education          1520.302326  2        6.244282
## Securities.Account    1.419873  1        1.191584
## CD.Account            2.264790  1        1.504922
## Online                1.220416  1        1.104725
## CreditCard            1.459672  1        1.208169
## Income:Education   1588.724858  2        6.313384
## Income:Family     17934.390697  3        5.116203
```

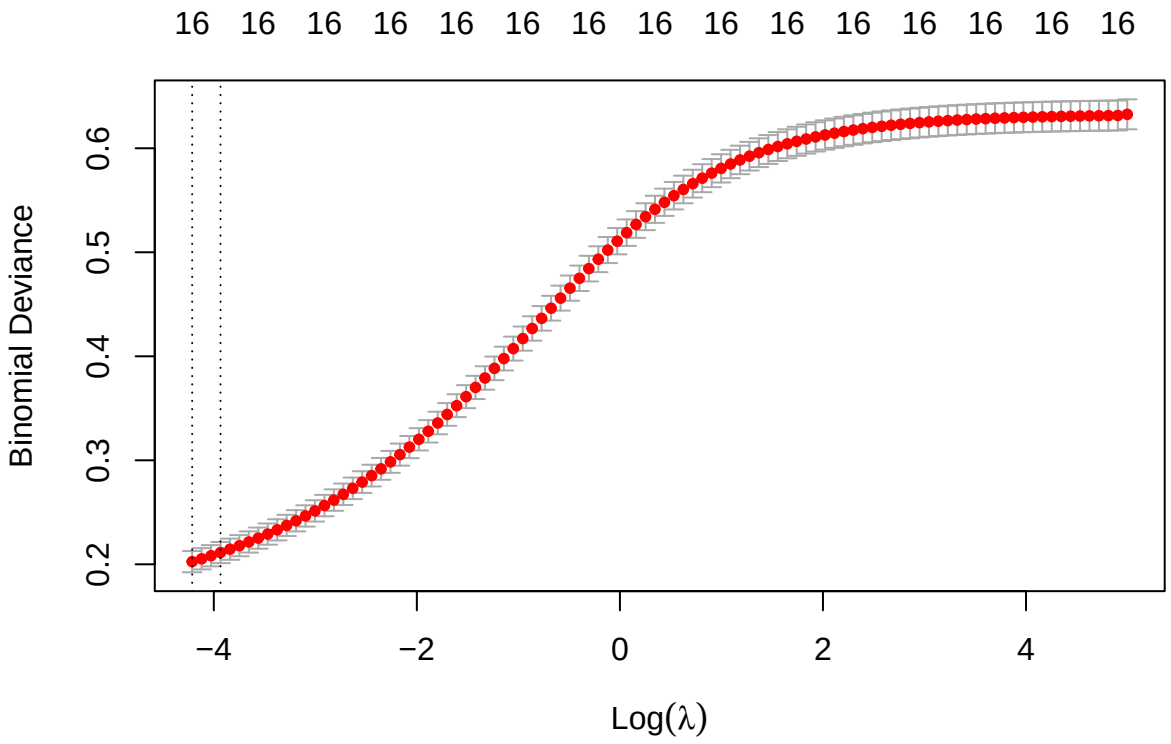
Upon identifying values of Generalized Variance Inflation Factor (GVIF) that raised concerns regarding multicollinearity, we proceeded to fit a Ridge regression model.

```
#Preparing the data
X = model.matrix(Personal.Loan ~ . + Income:Education + Income:Family, data = data2)[, -1]
Y = data$Personal.Loan

ridge_model = cv.glmnet(x = X, y = Y, family = "binomial", alpha = 0)
ridge_model
```

```
##
## Call:  cv.glmnet(x = X, y = Y, family = "binomial", alpha = 0)
##
## Measure: Binomial Deviance
##
##      Lambda Index Measure      SE Nonzero
## min 0.01480   100  0.2026 0.01019      16
## 1se 0.01957    97  0.2113 0.01018      16
```

```
plot(ridge_model)
```




```
coef(ridge_model, s = "lambda.min")
```

```
## 17 x 1 sparse Matrix of class "dgCMatrix"
##                                s1
## (Intercept)         -6.733889841
## Income               0.021407943
## Family2              0.096974469
## Family3             -0.225548522
## Family4             -0.332919802
## CCAvg                0.226260627
## Education2           0.014507601
## Education3           0.013655380
## Securities.Account1 -0.305116751
## CD.Account1          2.059839469
## Online1             -0.296651425
## CreditCard1         -0.404092132
## Income:Education2    0.021144137
## Income:Education3    0.021924182
## Income:Family2      -0.000403825
## Income:Family3       0.017676082
## Income:Family4       0.015598014
```

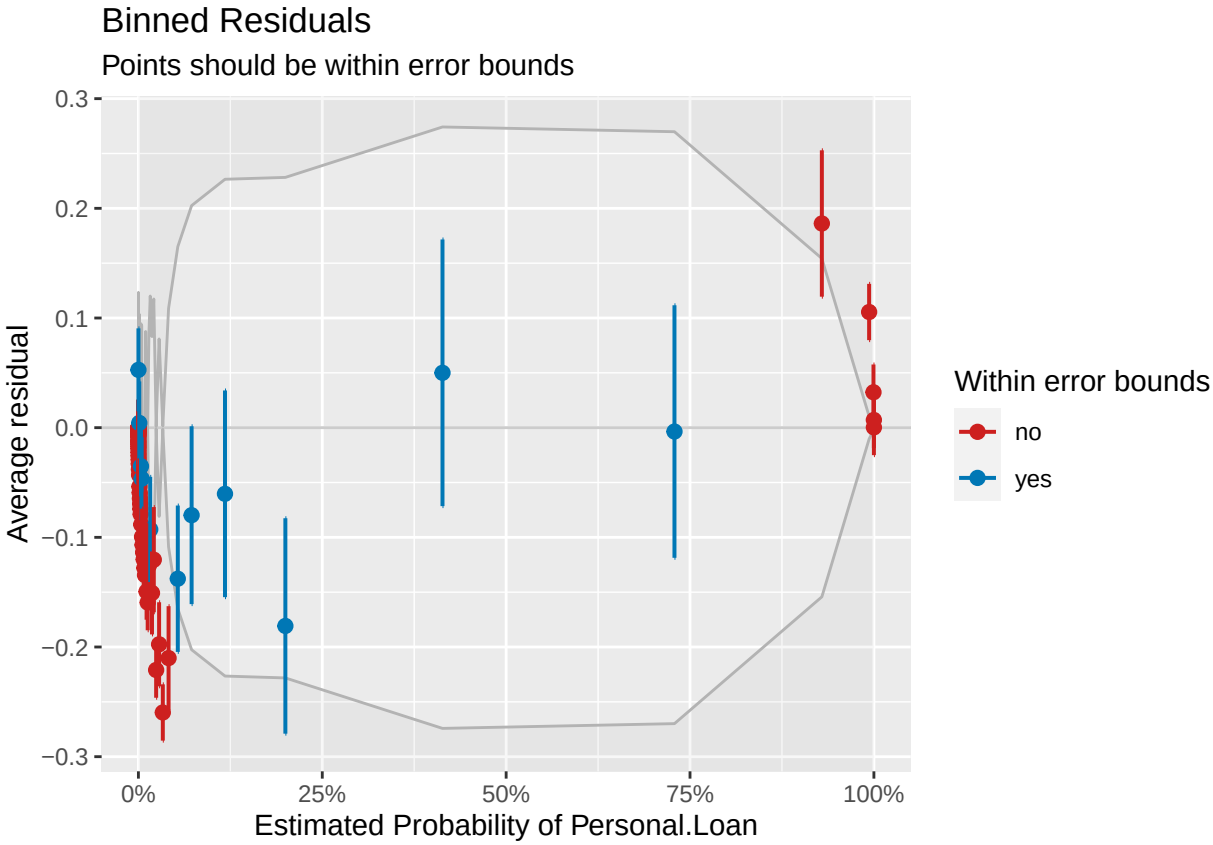
To validate our model selection and assess predictive performance, we implemented ten-fold cross-validation. This method provided a way for evaluating model accuracy, true positive rate (TPR), true negative rate (TNR), and the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC), offering insights into the model's generalizability and effectiveness in predicting personal loan acceptance.

```
##                    Accuracy       TPR       TNR       AUC
## Full Model           0.9582 0.6733271 0.9882993 0.9623707
## Restricted Model     0.9598 0.6799630 0.9891782 0.9625160
## Interaction Model    0.9774 0.8458088 0.9913829 0.9880953
## Ridge model          0.9482 0.5096023 0.9940367 0.9638978
```

Lastly, we examined the binned residuals, defined as:

dividing the data into categories (bins) based on their fitted values, and then plotting the average residual versus the average fitted value for each bin (Gelman, Hill 2007).

```
plot(binned_residuals(inter_model))
```



Analysis on Balanced Dataset

Dealing with imbalanced dataset can bias the model towards the majority class, leading to suboptimal classification of the minority class. To address this issue, we decided to balanced our dataset a priori through oversampling techniques. This process involved augmenting the minority class by replicating its instances, in order to have an equal representation of both classes in the dataset. This balanced approach aims to improve model sensitivity towards the minority class without compromising the overall predictive accuracy.

With the dataset balanced, we started with fitting a logistic regression model, mirroring the approach taken with the original dataset. This step served to establish a baseline.

```
full_model = glm(Personal.Loan ~ ., data = data_oversampled, family = binomial(link = logit))
summary(full_model)
```

```
##
## Call:
## glm(formula = Personal.Loan ~ ., family = binomial(link = logit),
##     data = data_oversampled)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -8.8327811  0.2729261 -32.363  < 2e-16 ***
## Age                  0.0039023  0.0033798   1.155    0.248
## Income               0.0550441  0.0014295  38.505  < 2e-16 ***
## Family2              0.0242674  0.1098439   0.221    0.825
## Family3              1.4361456  0.1163466  12.344  < 2e-16 ***
## Family4              1.4531724  0.1110884  13.081  < 2e-16 ***
## CCAvg                0.2514452  0.0243226  10.338  < 2e-16 ***
## Education2           2.6226968  0.1102450  23.790  < 2e-16 ***
## Education3           2.6662071  0.1090180  24.457  < 2e-16 ***
## Mortgage             0.0001218  0.0003260   0.374    0.709
## Securities.Account1 -1.0718214  0.1556182  -6.888 5.68e-12 ***
## CD.Account1          3.9720120  0.1861224  21.341  < 2e-16 ***
## Online1             -0.7729142  0.0843644  -9.162  < 2e-16 ***
## CreditCard1         -1.0490623  0.1018670 -10.298  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 12532.1  on 9039  degrees of freedom
## Residual deviance:  4361.2  on 9026  degrees of freedom
## AIC: 4389.2
##
## Number of Fisher Scoring iterations: 7
```

Following the initial assessment, we refined the model by excluding variables identified as statistically insignificant, specifically *Age* and *Mortgage*.

```
model = glm(Personal.Loan ~ . , data = data_oversampled2, family = binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Personal.Loan ~ ., family = binomial, data = data_oversampled2)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -8.648182   0.220677 -39.189   <2e-16 ***
## Income               0.055158   0.001415  38.993   <2e-16 ***
## Family2              0.022492   0.109796   0.205    0.838
## Family3              1.430347   0.115929  12.338   <2e-16 ***
## Family4              1.456200   0.110627  13.163   <2e-16 ***
## CCAvg                0.248120   0.024128  10.283   <2e-16 ***
## Education2           2.613799   0.109652  23.837   <2e-16 ***
## Education3           2.663756   0.108739  24.497   <2e-16 ***
## Securities.Account1 -1.083473   0.155274  -6.978    3e-12 ***
## CD.Account1          3.989091   0.184933  21.570   <2e-16 ***
## Online1             -0.769868   0.084359  -9.126   <2e-16 ***
## CreditCard1         -1.049702   0.101634 -10.328   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 12532.1  on 9039  degrees of freedom
## Residual deviance:  4362.7  on 9028  degrees of freedom
## AIC: 4386.7
##
## Number of Fisher Scoring iterations: 7
```

Again we investigated the role of interaction terms between *Income* and *Education*, and *Income* and *Family*.

```
inter_model = glm(Personal.Loan ~ . + Income:Education + Income:Family,
                  data = data_oversampled2, family = binomial)
summary(inter_model)
```

```
##
## Call:
## glm(formula = Personal.Loan ~ . + Income:Education + Income:Family,
##     family = binomial, data = data_oversampled2)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.879497   0.289631  -6.489 8.63e-11 ***
## Income               -0.020086   0.003194  -6.289 3.19e-10 ***
## Family2               0.838966   0.378672   2.216   0.0267 *
## Family3              -7.772398   0.718067 -10.824  < 2e-16 ***
## Family4              -9.804889   0.819295 -11.967  < 2e-16 ***
## CCAvg                 0.640312   0.042378  15.109  < 2e-16 ***
## Education2          -10.079187   0.717495 -14.048  < 2e-16 ***
## Education3          -10.346515   0.716994 -14.430  < 2e-16 ***
## Securities.Account1  -0.935539   0.200317  -4.670 3.01e-06 ***
## CD.Account1           3.800603   0.238765  15.918  < 2e-16 ***
## Online1              -0.681240   0.119434  -5.704 1.17e-08 ***
## CreditCard1          -1.122179   0.146479  -7.661 1.84e-14 ***
## Income:Education2     0.129382   0.007356  17.589  < 2e-16 ***
## Income:Education3     0.131410   0.007275  18.063  < 2e-16 ***
## Income:Family2       -0.008627   0.003634  -2.374   0.0176 *
## Income:Family3        0.100148   0.007361  13.606  < 2e-16 ***
## Income:Family4        0.119474   0.008210  14.552  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 12532.1  on 9039  degrees of freedom
## Residual deviance:  2284.6  on 9023  degrees of freedom
## AIC: 2318.6
##
## Number of Fisher Scoring iterations: 9
```

```r
anova(inter_model, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Personal.Loan
##
## Terms added sequentially (first to last)
##
##
##                     Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                 9039    12532.1
## Income               1   6112.6      9038     6419.5 < 2.2e-16 ***
## Family               3    370.5      9035     6049.0 < 2.2e-16 ***
## CCAvg                1    145.9      9034     5903.1 < 2.2e-16 ***
## Education            2    887.6      9032     5015.4 < 2.2e-16 ***
## Securities.Account   1     26.7      9031     4988.8 2.410e-07 ***
## CD.Account           1    464.0      9030     4524.8 < 2.2e-16 ***
## Online               1     48.9      9029     4475.8 2.664e-12 ***
## CreditCard           1    113.1      9028     4362.7 < 2.2e-16 ***
## Income:Education     2   1234.1      9026     3128.6 < 2.2e-16 ***
## Income:Family        3    844.0      9023     2284.6 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
res_dev = c(full_model$deviance, model$deviance, inter_model$deviance)
as.matrix(AIC(full_model, model, inter_model) %>% mutate(res_dev = res_dev))
```

```
##             df      AIC  res_dev
## full_model  14 4389.212 4361.212
## model       12 4386.713 4362.713
## inter_model 17 2318.602 2284.602
```

```r
coef(ridge_model, s = "lambda.min")
```

```
## 17 x 1 sparse Matrix of class "dgCMatrix"
##                                  s1
## (Intercept)          -4.177333e+00
## Income                2.041285e-02
## Family2               4.949862e-02
## Family3               3.286672e-02
## Family4              -4.789374e-02
## CCAvg                 2.218780e-01
## Education2            2.726659e-01
## Education3            2.990453e-01
## Securities.Account1 -2.257521e-01
## CD.Account1           1.577418e+00
## Online1              -2.381742e-01
## CreditCard1          -3.565485e-01
## Income:Education2     1.185321e-02
## Income:Education3     1.182772e-02
## Income:Family2        2.445573e-05
## Income:Family3        9.604026e-03
## Income:Family4        1.024587e-02
```

To validate model performance we used again a ten-fold cross-validation using the same metrics as before but now each iteration we split the original dataset into train set and test set and we make oversampling only on the train one. This validation step is critical, especially in the context of a balanced dataset, to ensure that the oversampling process does not lead to overfitting and that the models maintain their predictive power on unseen data.

```
##                    Accuracy       TPR       TNR       AUC
## Full Model           0.9064 0.8976568 0.9073048 0.9641418
## Restricted Model     0.9068 0.8998790 0.9075090 0.9645277
## Interaction Model    0.9616 0.9397615 0.9639330 0.9893060
## Ridge model          0.9038 0.8898984 0.9050072 0.9610008
```

From the results we can see that oversampling has significantly improved the True Positive Rate (TPR) across all models, indicating that it is effective in enhancing the models' ability to correctly identify positive cases. Despite the improvement in TPR, oversampling has not led to a significant increase in the Area Under the ROC Curve (AUC). This suggests that while oversampling improves the models' performance in terms of correctly identifying positive cases, it may not have a substantial impact on the models' overall discrimination ability between positive and negative cases.

Generalized Additive Models on Original Dataset

We started fitting a model that includes all variables; this first formulation includes smooth terms for Age, Income, CCAvg and Mortgage, allowing greater flexibility in the relationship between these variables and the response variable Personal.Loan, while also incorporating categorical factors representing Family, Education, Securities.Account, CD.Account, Online, and CreditCard, considered as linear.

```
full_model <- gam(Personal.Loan ~ s(Age) + s(Income) + s(CCAvg) + s(Mortgage) +
                    Family + Education + Securities.Account +
                    CD.Account + Online + CreditCard , data = data, family = binomial)
summary(full_model)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## Personal.Loan ~ s(Age) + s(Income) + s(CCAvg) + s(Mortgage) +
##     Family + Education + Securities.Account + CD.Account + Online +
##     CreditCard
##
## Parametric coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -10.9668     1.1184  -9.806  < 2e-16 ***
## Family2              -0.1008     0.2583  -0.390   0.6965
## Family3               3.0623     0.3250   9.422  < 2e-16 ***
## Family4               2.5265     0.3069   8.233  < 2e-16 ***
## Education2            4.5622     0.3288  13.876  < 2e-16 ***
## Education3            4.6085     0.3147  14.646  < 2e-16 ***
## Securities.Account1  -0.9789     0.4015  -2.438   0.0148 *
## CD.Account1           4.0252     0.4661   8.635  < 2e-16 ***
## Online1              -0.8717     0.2127  -4.098 4.17e-05 ***
## CreditCard1          -1.2134     0.2731  -4.444 8.83e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df  Chi.sq p-value
## s(Age)      3.226  4.018   6.492  0.1669
## s(Income)   6.096  6.966 327.652  <2e-16 ***
## s(CCAvg)    8.526  8.897 129.972  <2e-16 ***
## s(Mortgage) 1.001  1.002   3.254  0.0713 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.789   Deviance explained = 76.7%
## UBRE = -0.84098  Scale est. = 1         n = 5000
```

The second model excludes non-significant variables Age and Mortgage, and utilizes only Income and CCAvg as smooth terms. The other variables remain the same as in the first model.

```
model <- gam(Personal.Loan ~ s(Income) + s(CCAvg) +
                   Family + Education + Securities.Account +
                   CD.Account + Online + CreditCard , data = data, family = binomial)
summary(model)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## Personal.Loan ~ s(Income) + s(CCAvg) + Family + Education + Securities.Account +
##     CD.Account + Online + CreditCard
##
## Parametric coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -10.88189    1.08001 -10.076  < 2e-16 ***
## Family2              -0.06502    0.25746  -0.253   0.8006
## Family3               3.04439    0.32232   9.445  < 2e-16 ***
## Family4               2.53708    0.30543   8.307  < 2e-16 ***
## Education2            4.54996    0.32616  13.950  < 2e-16 ***
## Education3            4.53825    0.31112  14.587  < 2e-16 ***
## Securities.Account1  -0.98164    0.39742  -2.470   0.0135 *
## CD.Account1           4.02474    0.46017   8.746  < 2e-16 ***
## Online1              -0.86397    0.21012  -4.112 3.93e-05 ***
## CreditCard1          -1.22502    0.27115  -4.518 6.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df Chi.sq p-value
## s(Income) 5.987  6.886  332.5  <2e-16 ***
## s(CCAvg)  8.488  8.862  128.6  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.784   Deviance explained = 76.3%
## UBRE = -0.84028  Scale est. = 1         n = 5000
```

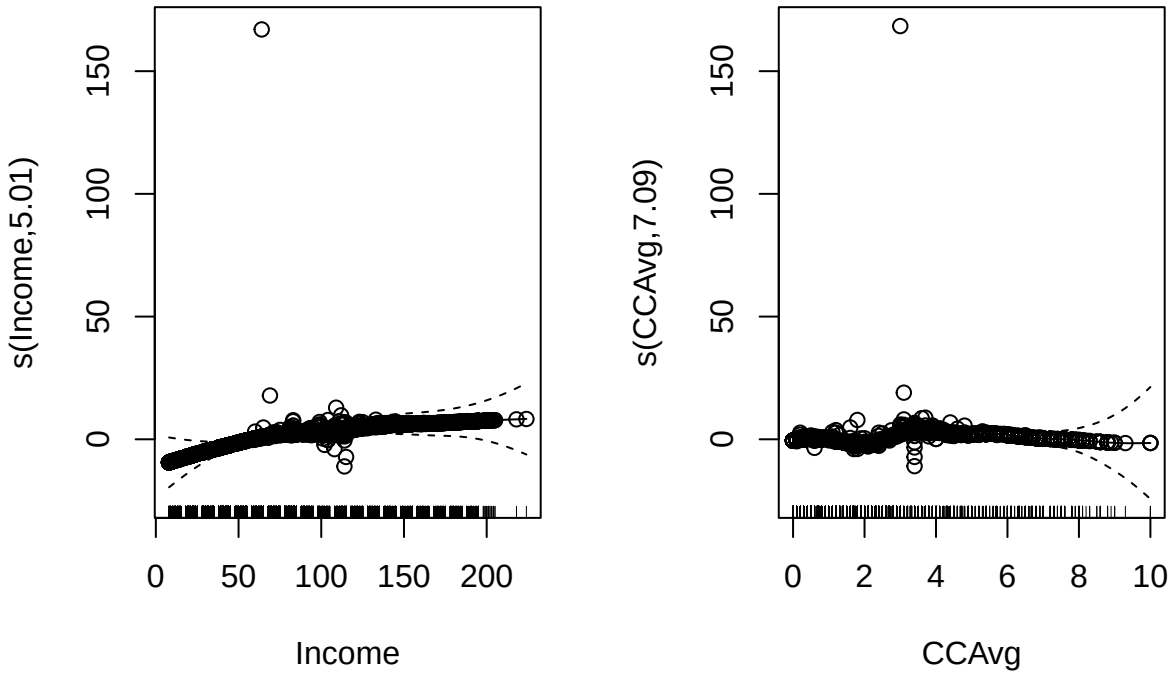
The third model includes also interactions between Income and Family or Education.They demonstrated significance, suggesting that the relationship between income and loan eligibility varies based on family size and education level.

```
inter_model <- gam(Personal.Loan ~ s(Income) + s(CCAvg) +
                     Family + Education + Securities.Account +
                     CD.Account + Online + CreditCard + Income:Family +
                      Income:Education , data = data, family = binomial)
summary(inter_model)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## Personal.Loan ~ s(Income) + s(CCAvg) + Family + Education + Securities.Account +
##     CD.Account + Online + CreditCard + Income:Family + Income:Education
##
## Parametric coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            0.00000    0.00000     NaN      NaN
## Family2                0.14463    1.47389   0.098 0.921831
## Family3              -10.14593    2.11353  -4.800 1.58e-06 ***
## Family4              -12.56370    2.16071  -5.815 6.08e-09 ***
## Education2           -12.29673    1.89843  -6.477 9.34e-11 ***
## Education3           -15.51307    2.13719  -7.259 3.91e-13 ***
## Securities.Account1   -0.92338    0.47474  -1.945 0.051772 .
## CD.Account1            3.86261    0.55314   6.983 2.89e-12 ***
## Online1               -0.92024    0.27702  -3.322 0.000894 ***
## CreditCard1           -1.28755    0.34655  -3.715 0.000203 ***
## Family1:Income        -0.07911    0.01331  -5.944 2.78e-09 ***
## Family2:Income        -0.08141    0.02141  -3.802 0.000143 ***
## Family3:Income         0.04406    0.02455   1.795 0.072680 .
## Family4:Income         0.06383    0.02474   2.580 0.009870 **
## Education2:Income      0.15021    0.01843   8.152 3.58e-16 ***
## Education3:Income      0.17772    0.02069   8.590  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df Chi.sq p-value
## s(Income) 5.005  5.926  21.15 0.00136 **
## s(CCAvg)  7.090  7.800 116.97 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 33/34
## R-sq.(adj) =  0.864   Deviance explained =   86%
## UBRE = -0.90087  Scale est. = 1         n = 5000
```

There we plots the residuals of model with interactions to assess the model's fit.

```
plot(inter_model, residuals=TRUE, pch=1, pages=1)
```

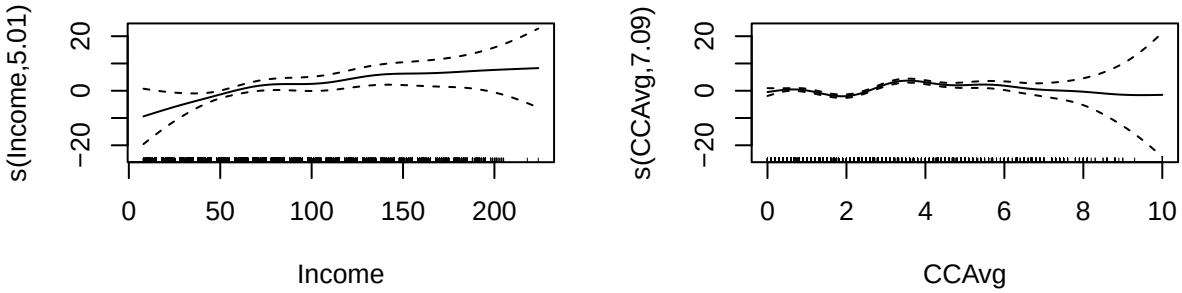


To determine the most effective model, we calculated and compared the Akaike Information Criterion (AIC) values of the full model, the model, and the model with interactions, alongside their deviance residuals.

```
##                   df      AIC  res_dev
## full_model  28.84875 795.0911 737.3936
## model       24.47439 798.6215 749.6727
## inter_model 27.09547 495.6602 441.4692
```

Finally we plotted the fitted smooth effects to visually assess the model's fit and identify any potential issues.

```
formula <- Personal.Loan ~ s(Income) + s(CCAvg) + Family + Education +
  Securities.Account + CD.Account + Online + CreditCard + Income:Family + Income:Education
model <- gam(formula, data = data, family = binomial)
num_variables <- length(model$terms)
num_rows <- ceiling(num_variables / 2)
par(mfrow = c(num_rows, 2))
for (i in 1:num_variables) {
  plot(model, select = i)
}
```



To evaluate the predictive performance, we implemented ten-fold cross-validation. This method allowed us to evaluate the model's accuracy, true positive rate (TPR), true negative rate (TNR) and area under the curve (AUC) of receiver operating characteristic (ROC).

```
## Full Model

## mean_accuracy      mean_tpr      mean_tnr      mean_auc
##     0.9846000     0.9562072     0.9875088     0.9940265

## Model

## mean_accuracy      mean_tpr      mean_tnr      mean_auc
##     0.9846000     0.9592666     0.9870795     0.9937915

## Model with interactions

## mean_accuracy      mean_tpr      mean_tnr      mean_auc
##     0.9832000     0.9501033     0.9864247     0.9941464
```

Generalized Additive Models on Balanced Dataset

We decided to balance our dataset through oversampling techniques. In this procedure, we balanced the dataset by increasing the representation of the minority class through replication of its instances. By doing so, we aimed to improve the model's sensitivity while maintaining overall predictive accuracy.

We repeated what we did in the first part with the new dataset starting with the model that include all variables.

```
full_model_oversampled <- gam(Personal.Loan ~ s(Age) + s(Income) + s(CCAvg) + s(Mortgage)
                    + Family + Education + Securities.Account +
                    CD.Account + Online + CreditCard , data = data_oversampled2,
                    family = binomial)
summary(full_model_oversampled)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## Personal.Loan ~ s(Age) + s(Income) + s(CCAvg) + s(Mortgage) +
##     Family + Education + Securities.Account + CD.Account + Online +
##     CreditCard
##
## Parametric coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -21.57435    7.28137  -2.963  0.00305 **
## Family2               0.02888    0.12973   0.223  0.82386
## Family3               2.47668    0.16256  15.236  < 2e-16 ***
## Family4               2.42443    0.15103  16.053  < 2e-16 ***
## Education2            3.60350    0.15124  23.827  < 2e-16 ***
## Education3            3.48170    0.13995  24.879  < 2e-16 ***
## Securities.Account1  -1.25153    0.20052  -6.241 4.34e-10 ***
## CD.Account1           4.22051    0.24525  17.209  < 2e-16 ***
## Online1              -0.93887    0.11056  -8.492  < 2e-16 ***
## CreditCard1          -1.15954    0.12926  -8.970  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                edf Ref.df   Chi.sq  p-value
## s(Age)       7.288  8.271   38.743 4.74e-06 ***
## s(Income)    8.865  8.982 1165.392  < 2e-16 ***
## s(CCAvg)     8.827  8.988  581.790  < 2e-16 ***
## s(Mortgage) 1.014  1.028    5.853   0.0159 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.831   Deviance explained = 78.3%
## UBRE = -0.69133  Scale est. = 1         n = 9320
```

Then, according to the summary, we decided not to use spline for Mortgage variable.

```
model_oversampled <- gam(Personal.Loan ~ s(Age) + s(Income) + s(CCAvg) + Mortgage +
                    Family + Education + Securities.Account +
                    CD.Account + Online + CreditCard , data = data_oversampled2,
                    family = binomial)
summary(model_oversampled)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## Personal.Loan ~ s(Age) + s(Income) + s(CCAvg) + Mortgage + Family +
##     Education + Securities.Account + CD.Account + Online + CreditCard
##
## Parametric coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -2.167e+01  7.286e+00  -2.974  0.00294 **
## Mortgage             9.235e-04  3.799e-04   2.431  0.01506 *
## Family2              2.887e-02  1.297e-01   0.223  0.82391
## Family3              2.476e+00  1.625e-01  15.236  < 2e-16 ***
## Family4              2.424e+00  1.510e-01  16.052  < 2e-16 ***
## Education2           3.603e+00  1.512e-01  23.827  < 2e-16 ***
## Education3           3.481e+00  1.399e-01  24.880  < 2e-16 ***
## Securities.Account1 -1.251e+00  2.005e-01  -6.241 4.34e-10 ***
## CD.Account1          4.220e+00  2.452e-01  17.209  < 2e-16 ***
## Online1             -9.387e-01  1.105e-01  -8.491  < 2e-16 ***
## CreditCard1         -1.159e+00  1.293e-01  -8.969  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df  Chi.sq  p-value
## s(Age)    7.288  8.271   38.72 4.82e-06 ***
## s(Income) 8.866  8.982 1165.78  < 2e-16 ***
## s(CCAvg)  8.820  8.987  581.84  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.831   Deviance explained = 78.3%
## UBRE = -0.69133  Scale est. = 1         n = 9320
```

Then, as before, we introduced interactions between Income and Family or Education.

```
inter_model_oversampled <- gam(Personal.Loan ~ s(Age) + s(Income) + s(CCAvg) + Mortgage +
                    Family + Education + Securities.Account +
                    CD.Account + Online + CreditCard + Income:Family + Income:Education ,
                    data = data_oversampled2, family = binomial)
summary(inter_model_oversampled)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## Personal.Loan ~ s(Age) + s(Income) + s(CCAvg) + Mortgage + Family +
##     Education + Securities.Account + CD.Account + Online + CreditCard +
##     Income:Family + Income:Education
##
## Parametric coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          0.000e+00  0.000e+00     NaN      NaN
## Mortgage             7.558e-05  6.521e-04   0.116 0.907739
## Family2              8.049e-01  8.256e-01   0.975 0.329597
## Family3             -9.795e+00  1.020e+00  -9.606  < 2e-16 ***
## Family4             -1.377e+01  1.128e+00 -12.211  < 2e-16 ***
## Education2          -1.253e+01  1.008e+00 -12.431  < 2e-16 ***
## Education3          -1.399e+01  1.001e+00 -13.970  < 2e-16 ***
## Securities.Account1 -1.056e+00  2.670e-01  -3.955 7.64e-05 ***
## CD.Account1          3.848e+00  3.274e-01  11.753  < 2e-16 ***
## Online1             -8.796e-01  1.560e-01  -5.640 1.70e-08 ***
## CreditCard1         -1.089e+00  1.760e-01  -6.185 6.19e-10 ***
## Family1:Income      -7.236e-01  1.458e-01  -4.964 6.91e-07 ***
## Family2:Income      -7.316e-01  1.475e-01  -4.961 7.01e-07 ***
## Family3:Income      -5.995e-01  1.456e-01  -4.116 3.85e-05 ***
## Family4:Income      -5.573e-01  1.448e-01  -3.848 0.000119 ***
## Education2:Income    1.590e-01  1.032e-02  15.401  < 2e-16 ***
## Education3:Income    1.693e-01  1.009e-02  16.774  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df Chi.sq p-value
## s(Age)    7.161  8.153  40.93  <2e-16 ***
## s(Income) 8.955  8.998  71.25  <2e-16 ***
## s(CCAvg)  7.758  8.336 452.67  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 43/44
## R-sq.(adj) =  0.913   Deviance explained = 88.8%
## UBRE = -0.83693  Scale est. = 1         n = 9320
```

```
##                               df      AIC  res_dev
## full_model_oversampled  35.99451 2876.801 2804.812
## model_oversampled       35.97375 2876.798 2804.851
## inter_model_oversampled 39.87363 1519.857 1440.110
```

To evaluate the predictive performance, we again implemented ten-fold cross-validation.

```
## Full Model
```

```
## mean_accuracy      mean_tpr      mean_tnr      mean_auc
##     0.9832000     0.9508799     0.9864562     0.9943261
```

```
## Model
```

```
## mean_accuracy      mean_tpr      mean_tnr      mean_auc
##     0.9838000     0.9551467     0.9866446     0.9943584
```

```
## Model with interactions
```

```
## mean_accuracy      mean_tpr      mean_tnr      mean_auc
##     0.9840000     0.9551473     0.9868724     0.9941049
```

By using oversampling, we fixed the problem of imbalanced classes, making our models perform better than those trained on the original dataset. Although the oversampled model had slightly lower accuracy than the initial ones, it still showed strong predictive ability and balanced performance. In summary, our models trained on the oversampled data handled class imbalance well and remained competitive in predicting loan eligibility.

In conclusion, our GAM models show promise in predicting whether an individual will join the last personal loan campaign promoted by the bank based on the given variables.

Support Vector Machine

Imbalanced case

We then built an SVM model, with both a linear and a radial kernel. Like in the previous cases the model has been first tested on the imbalanced dataset and then on a balanced with oversampling one. The procedure for both the models, balanced and imbalanced is the same.

For the radial kernel model we first trained a single SVM radial model in order to tune the value of gamma and cost parameters. Using a 10-fold cross validation tuning function the results are the following: gamma=0.25 and cost = 10, we can note that the best value for the cost parameter is similar to the imbalance ratio, that hence is a good trade-off value between achieving a low training error and a low model complexity. Once these parameters have been set we implemented a 10-fold cross validation to train and test the model each time computing the performance indexes of interest. We then considered the mean value of each as estimator.

In the imbalanced case we divided the dataset in train and test data, trained both the linear kernel model and the radial one on the train data and then made the predictions. For both the cases we built the confusion matrix to extract the TPR, the TNR and the Accuracy, we built the ROC curve and computed the AUC.

In the following summary we present the result coming from the imbalanced dataset. We see that the model has good performances and as expected, the radial kernel model performs better than the linear one, with an AUC=0.92 .

```
# Print the data frame
print(svm_summary)
```

```
##         AUC Accuracy  TPR  TNR
## Linear 0.81     0.96 0.91 0.96
## Radial 0.92     0.98 0.96 0.98
```

Balanced case

The same procedure has been done for the balanced case, with the only difference consisting in balancing for oversampling the train dataset before training the models.

We now show the summary of the balanced case.

```
# Print the data frame
print(bal.svm_summary)
```

```
##         AUC Accuracy  TPR  TNR
## Linear 0.90     0.91 0.51 0.99
## Radial 0.93     0.98 0.91 0.99
```

The resulting AUC is 0.93. We can note that the performances do not improve, and that the TPR for the linear kernel model falls at a lower value with respect to the imbalanced case. The reason for this fall could be due to a higher number of data out of the separating hyperplane. Indeed having less data, the loss function of the SVM model is able to penalize the data that cannot be separated and then generate a plan that better separates the remaining data. When increasing these data, the penalization could not be able to perform as the imbalanced case and then the resulting plane divides the data in a worse way such that TPR falls to TPR=0.5. We also tried to use oversampling methods that generate new data, not only using replacement of the existing data. With this aim we use the ROSE method, but the results are similar, as demonstration that increasing the number of points lead to an increase of non separable points hence the resulting hyperplane is worse than the one coming from the imbalanced case.

To be thorough we show the results obtained after applying the ROSE method.

```
# Print the data frame
print(ros.svm_summary)
```

```
##         AUC Accuracy  TPR  TNR
## Linear 0.88     0.90 0.50 0.98
## Radial 0.93     0.96 0.78 0.99
```

To conclude we can say that according to the Occam's Razor principle we should prefer the imbalanced model that requires a lower computational effort.

Random forest

We then built a Random Forest model with a 10-fold cross-validation in order to better identify the best number of variable to consider for each tree. We tested the model with both the unbalanced and balanced datasets and to evaluate their performances we divided the data in training and testing sets before carry out the balancing.

```
#Partitioning data in training and test datasets
ind <- sample(2, nrow(data), replace = TRUE, prob = c(0.7, 0.3))
train <- data[ind==1,]
test <- data[ind==2,]

#Dividing training dataset by the responce in order to perform oversampling

minority_train_data <- train[train$Personal.Loan == 1, ]
majority_train_data <- train[train$Personal.Loan == 0, ]
```

Unbalanced dataset

The cross-validation pointed out that the best number of variable to consider for each random forest, based on the training on the unbalanced dataset, is 7.

```
#10-fold CV model evaluation with the unbalanced dataset
train_control <- trainControl(method = "cv", number = 10)
rf0 <- train(Personal.Loan~., data = train,
               method = "rf",
               trControl = train_control)

#Model description and confusion matrix based upon OOB samples
print(rf0)
```

```
## Random Forest
##
## 3513 samples
##   10 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 3162, 3161, 3162, 3162, 3162, 3161, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9743808  0.8171069
##    7    0.9891827  0.9306748
##   13    0.9891803  0.9311163
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 7.
```

From the confusion matrix evaluated on the test set its worth noticing the TPR (sensitivity) of 99.92% and the TNR (specificity) of 88.48%.

```
#Model test
p0 <- predict(rf0, test)
```

```r
#Confusion matrix and statistics for the model assessment
#Sensitivity -> TPR
#Specificity -> TNR
confusionMatrix(p0, test$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1321   19
##          1    1  146
##
##                Accuracy : 0.9866
##                  95% CI : (0.9793, 0.9918)
##     No Information Rate : 0.889
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9284
##
##  Mcnemar's Test P-Value : 0.0001439
##
##             Sensitivity : 0.9992
##             Specificity : 0.8848
##          Pos Pred Value : 0.9858
##          Neg Pred Value : 0.9932
##              Prevalence : 0.8890
##          Detection Rate : 0.8884
##    Detection Prevalence : 0.9011
##       Balanced Accuracy : 0.9420
##
##        'Positive' Class : 0
##
```

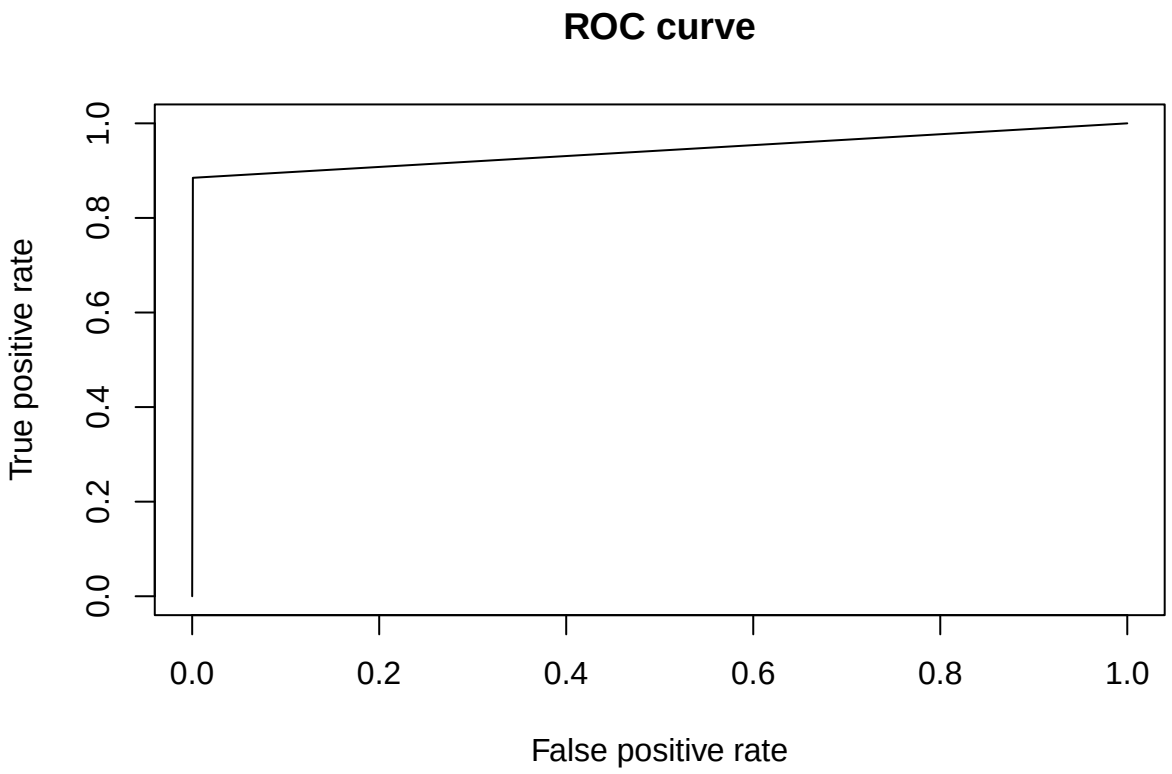
The resulting AUC for the unbalanced dataset is 94.20%.

```r
#AUC
pr0 <- prediction(as.numeric(p0), test$Personal.Loan)
auc0 <- performance(pr0, measure = "auc")@y.values[[1]]
print(paste("AUC : ", auc0))
```

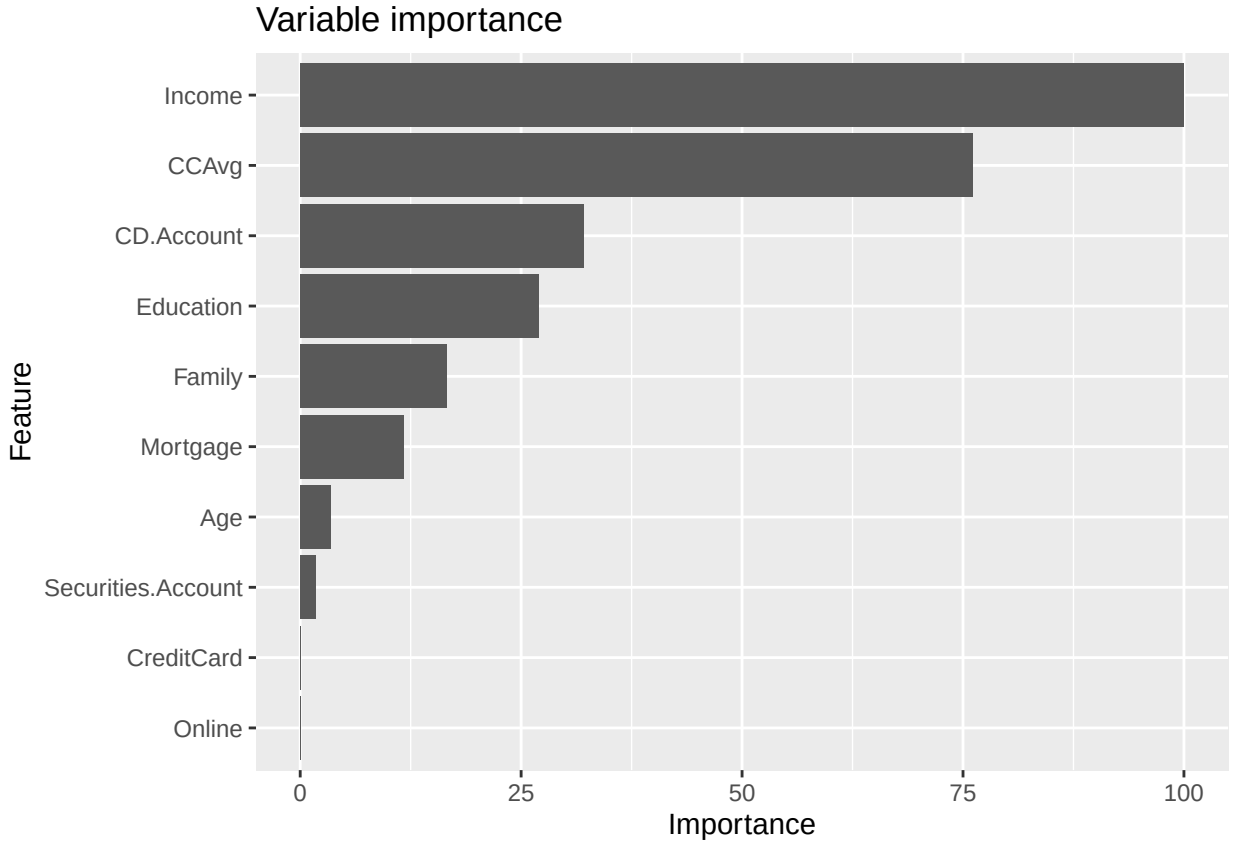
```
## [1] "AUC :  0.942046027598221"
```

Finally its possible to look at the ROC curve and the variable importance plot.

```r
#ROC curve plot
roc0 <- performance(pr0,"tpr", "fpr")
plot(roc0, main = "ROC curve")
```




```
#Variable importance plot
ggplot2::ggplot(varImp(rf0, useModel = FALSE))+ggtitle("Variable importance")
```



Balanced dataset

For the balanced dataset too the cross-validation pointed out that the best number of variable to consider for each random forest is 7.

```
#Balancing dataset with simple oversampling
train_data_oversampled = minority_train_data[sample(1:nrow(minority_train_data),
                                                    size = nrow(majority_train_data), replace = TRUE), ]
  rbind(., majority_train_data)

#10-fold CV model evaluation
rf1 <- train(Personal.Loan~., data = train_data_oversampled,
               method = "rf",
               trControl = train_control)

#Model description and confusion matrix based upon OOB samples
print(rf1)
```

```
## Random Forest
##
## 6396 samples
##   10 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 5756, 5756, 5757, 5756, 5757, 5756, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9931208  0.9862416
##    7    0.9965605  0.9931211
##   13    0.9962468  0.9924936
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 7.
```

From the confusion matrix evaluated on the test set its worth noticing the TPR (sensitivity) of 99.55% and the TNR (specificity) of 92.73%.

```
#Model test
p1 <- predict(rf1, test)

#Confusion matrix and statistics for the model assessment
#Sensitivity -> TPR
#Specificity -> TNR
confusionMatrix(p1, test$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1316   12
##          1    6  153
##
##                Accuracy : 0.9879
```

```
##                  95% CI : (0.9809, 0.9928)
##     No Information Rate : 0.889
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9377
##
##  Mcnemar's Test P-Value : 0.2386
##
##             Sensitivity : 0.9955
##             Specificity : 0.9273
##          Pos Pred Value : 0.9910
##          Neg Pred Value : 0.9623
##              Prevalence : 0.8890
##          Detection Rate : 0.8850
##    Detection Prevalence : 0.8931
##       Balanced Accuracy : 0.9614
##
##        'Positive' Class : 0
##
```

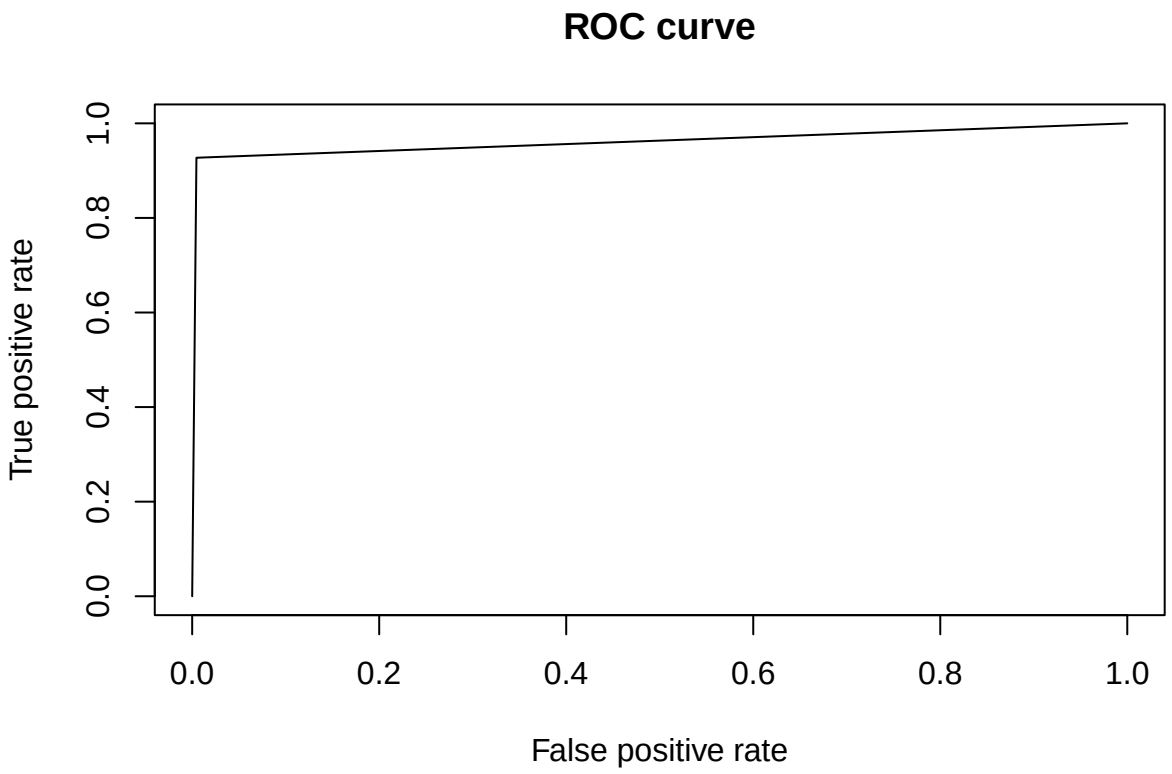
The resulting AUC for the balanced dataset is 96.14%.

```
#AUC
pr1 <- prediction(as.numeric(p1), test$Personal.Loan)
auc1 <- performance(pr1, measure = "auc")@y.values[[1]]
print(paste("AUC : ", auc1))
```

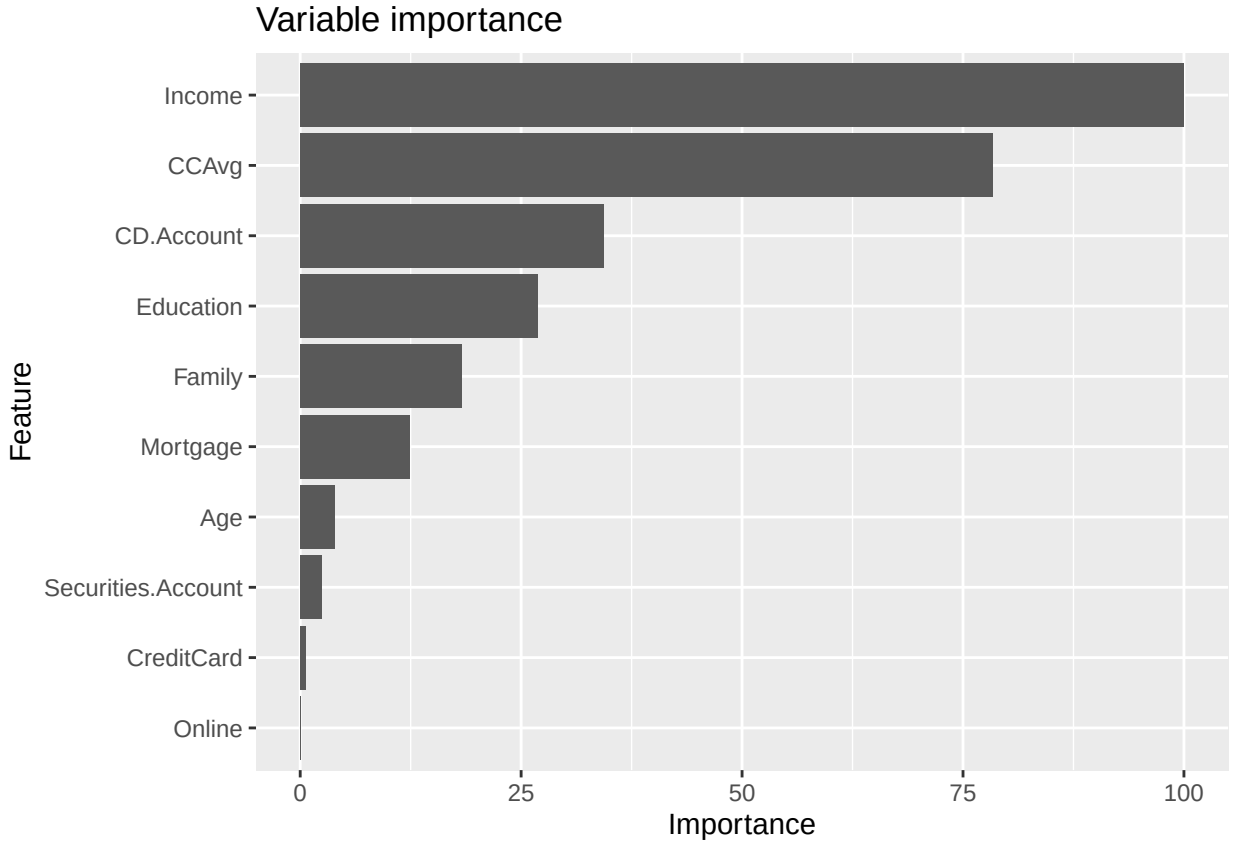
```
## [1] "AUC :  0.961367074680237"
```

Finally its possible to look at the ROC curve and the variable importance plot for the balanced dataset too.

```
#ROC curve plot
roc1 <- performance(pr1,"tpr", "fpr")
plot(roc1, main = "ROC curve")
```




```
#Variable importance plot
ggplot2::ggplot(varImp(rf1, useModel = FALSE))+ggtitle("Variable importance")
```



In conclusion the balanced dataset performed slightly better than the unbalanced one in terms of AUC; the variable importance didn't change significantly among the datasets.

A Bayesian Approach

The last model we analyzed is a Bayesian logistic regression model: the coefficients of the variables were studied in a Bayesian approach.

After trying another package (UPG), we opted to use the brms package: Bayesian Multilevel Models using Stan.

Since with many data the Bayesian model is very similar to the classic logistic regression one, we tried to balance the dataset through undersampling.

```
data_train <- data[1:4500,]
data_test <- data[4501:5000,]

majority_indices <- which(data_train$Personal.Loan == 0)
minority_indices <- which(data_train$Personal.Loan == 1)

num_majority <- length(majority_indices)
num_minority <- length(minority_indices)

undersampled_indices <- sample(majority_indices, size = 3*num_minority)

balanced_indices <- c(undersampled_indices, minority_indices)
balanced_data_train <- data[balanced_indices, ]
shuffled_indices <- sample(nrow(balanced_data_train))
balanced_data_train <- balanced_data_train[shuffled_indices, ]
```

```
# Glm model to compare with the Bayesian one
train_model = glm(Personal.Loan ~ Income + Family + CCAvg + Education + Securities.Account +
CD.Account + Online + CreditCard + Mortgage, data = balanced_data_train, family = binomial)
summary(train_model)
```

```
##
## Call:
## glm(formula = Personal.Loan ~ Income + Family + CCAvg + Education +
##     Securities.Account + CD.Account + Online + CreditCard + Mortgage,
##     family = binomial, data = balanced_data_train)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.111e+01  6.590e-01 -16.862  < 2e-16 ***
## Income               6.092e-02  3.946e-03  15.440  < 2e-16 ***
## Family2              5.212e-02  2.942e-01   0.177 0.859390
## Family3              1.990e+00  3.117e-01   6.382 1.74e-10 ***
## Family4              1.966e+00  2.965e-01   6.631 3.34e-11 ***
## CCAvg                2.146e-01  5.959e-02   3.601 0.000317 ***
## Education2           3.294e+00  3.195e-01  10.308  < 2e-16 ***
## Education3           3.556e+00  3.162e-01  11.245  < 2e-16 ***
## Securities.Account1 -1.244e+00  3.949e-01  -3.149 0.001637 **
## CD.Account1          4.300e+00  4.722e-01   9.106  < 2e-16 ***
## Online1             -7.626e-01  2.182e-01  -3.495 0.000473 ***
## CreditCard1         -1.146e+00  2.655e-01  -4.315 1.60e-05 ***
## Mortgage             6.356e-04  8.231e-04   0.772 0.440013
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2006.41  on 1783  degrees of freedom
## Residual deviance:  657.02  on 1771  degrees of freedom
## AIC: 683.02
##
## Number of Fisher Scoring iterations: 7

## [1] "Assessment for logistic regression model."

## [1] "Correct Predictions: 477"

## [1] "Accuracy: 0.954"

## [1] "True Positive Rate: 0.794117647058823"

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## [1] "AUC: 0.950896238323653"

## [1] "Dummy Accuracy: 0.932"
```

Let's delve into the Bayesian model. Since we are facing a binary classification problem, the likelihood is:

$$L(X, y|\beta) = \Pi_i p(X_i|\beta)^{y_i} * (1 - p(X_i|\beta))^{1-y_i}$$

where

$$p(X_i|\beta) = \frac{1}{1 + e^{-X_i\beta}}$$

Our posterior will be

$$\pi(\beta|X, y) \propto L(X, y|\beta)\pi(\beta)$$

One of the initial challenges encountered in this modeling process is setting the prior distributions for the coefficients: it forced us to develop our own idea about the underlying system that produces the outcome variable Personal.Loan.

It is interesting to note that, reasonably, without setting the prior we obtain a model equal to the one we get with the function glm().

```
capture.output({

# Define informative priors for each coefficient
 prior_spec <- c(
  # We don't set a Prior for the intercept, Age, CreditCard and Online,
  # since we don't have an idea about them
  # A person with high income has probably a stable financial situation and
  # can ask for a loan for long-term life projects
  prior(normal(0.3, 0.3), class = "b", coef = "Income"),
  # we expect that a family of 2 people doesn't need a loan
  prior(normal(-1, 0.5), class = "b", coef = "Family2"),
  # a family with a child may need a loan, the same for 2 children
    prior(normal(1, 0.5), class = "b", coef = "Family3"),
    prior(normal(1.5, 0.5), class = "b", coef = "Family4"),
  # Individuals with higher credit card usage may need money, so they could ask for a loan
  prior(normal(0.5, 0.5), class = "b", coef = "CCAvg"),
  # People with higher education more likely may take the "risk" of accepting a loan
```

```
  prior(normal(1, 0.5), class = "b", coef = "Education2"),
  prior(normal(2, 0.5), class = "b", coef = "Education3"),
  # A person with securities or cd account may need money and
  # can accept a loan because he knows he can repay it
  # securities account, investments account
  prior(normal(0.5, 0.5), class = "b", coef = "Securities.Account1"),
  # certificate of deposit account, which is a type of savings account
  # with a fixed term and interest rate
  prior(normal(0.5, 0.5), class = "b", coef = "CD.Account1"),
  # if you have a mortgage active, you may not want to ask for a loan
  prior(normal(-1, 0.3), class = "b", coef = "Mortgage")
)

# Fit the Bayesian logistic regression model with the specified priors
  model <- brm(
  formula = Personal.Loan ~ Age + Income + Family + CCAvg + Education + Mortgage
  + Securities.Account + CD.Account + Online + CreditCard,
  data = balanced_data_train,
  family = bernoulli(link = "logit"),
  prior = prior_spec,
  chains = 4, # controls parallel chains for convergence diagnostics.
  iter = 2000, # determines the total number of iterations per chain.
  warmup = 1000 # sets the number of warm-up iterations discarded before collecting posterior samples.
)

}, file = "/dev/null")
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: Personal.Loan ~ Age + Income + Family + CCAvg + Education + Mortgage + Securities.Account +
##    Data: balanced_data_train (Number of observations: 1784)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##                     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept              -9.94      0.62   -11.16    -8.72 1.00     2954     2601
## Age                     0.00      0.01    -0.01     0.02 1.00     6116     3187
## Income                  0.05      0.00     0.05     0.06 1.00     3236     3154
## Family2                -0.31      0.23    -0.76     0.13 1.00     4131     3206
## Family3                 1.55      0.23     1.10     2.02 1.00     3437     2983
## Family4                 1.60      0.23     1.15     2.05 1.00     3345     2751
## CCAvg                   0.19      0.05     0.08     0.29 1.00     4587     3300
## Education2              2.53      0.23     2.08     2.99 1.00     3150     3091
## Education3              2.81      0.23     2.37     3.27 1.00     3253     2921
## Mortgage                0.00      0.00    -0.00     0.00 1.00     4145     2922
## Securities.Account1    -0.28      0.27    -0.81     0.23 1.00     4719     2879
## CD.Account1             2.48      0.30     1.89     3.08 1.00     3918     3329
## Online1                -0.46      0.20    -0.85    -0.08 1.00     4804     3201
## CreditCard1            -0.71      0.23    -1.17    -0.26 1.00     4229     2836
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

We can see that the sign of the coefficients stay the same, meaning that our prior may be somehow correct.

We can also extract the WAIC coefficient, similar to the AIC: lower values indicate better model performance.

```
##
## Computed from 4000 by 1784 log-likelihood matrix
##
##           Estimate   SE
## elpd_waic   -351.9 21.0
## p_waic        11.0  0.8
## waic         703.9 41.9
```

We use function hypothesis() of the brms package to test the significance of the predictors.

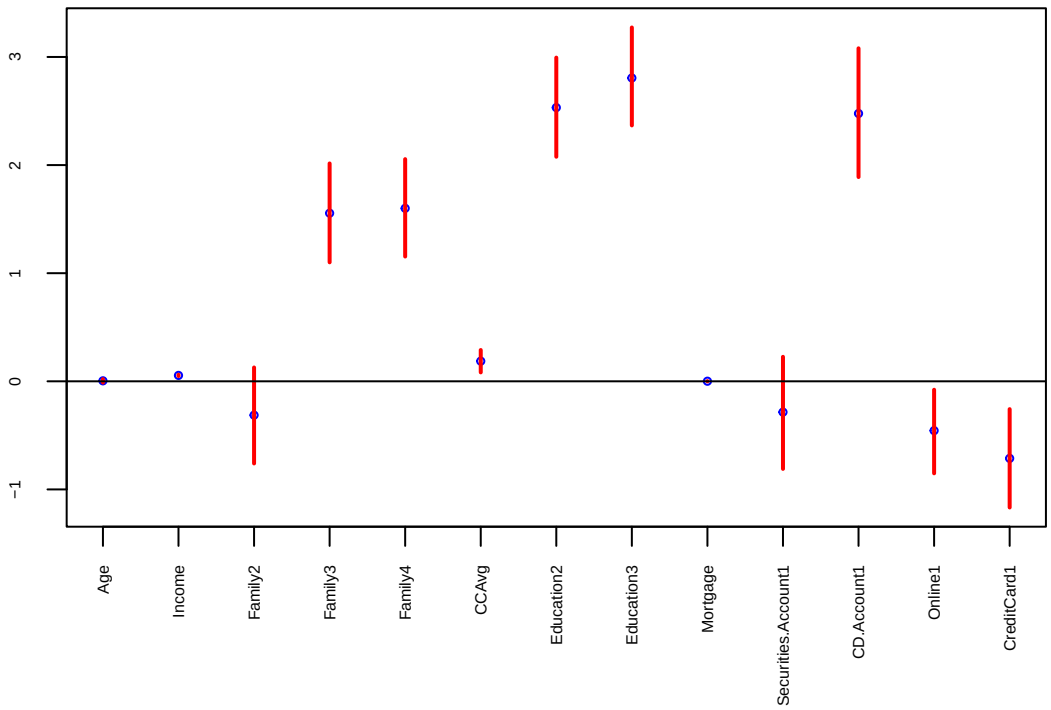
```
## Hypothesis Tests for class b:
##                 Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio
## 1                (Age) > 0     0.00      0.01    -0.01     0.02       2.36
## 2             (Income) > 0     0.05      0.00     0.05     0.06        Inf
## 3              (CCAvg) > 0     0.19      0.05     0.10     0.27     999.00
## 4            (Mortgage) > 0     0.00      0.00     0.00     0.00       4.44
## 5  (Securities.Accou... > 0    -0.28      0.27    -0.73     0.14       0.17
## 6         (CD.Account1) > 0     2.48      0.30     1.98     2.98        Inf
## 7             (Online1) > 0    -0.46      0.20    -0.79    -0.13       0.01
## 8         (CreditCard1) > 0    -0.71      0.23    -1.11    -0.34       0.00
## 9             (Family2) > 0    -0.31      0.23    -0.69     0.06       0.10
## 10            (Family3) > 0     1.55      0.23     1.19     1.93        Inf
## 11            (Family4) > 0     1.60      0.23     1.23     1.98        Inf
## 12         (Education2) > 0     2.53      0.23     2.16     2.93        Inf
## 13         (Education3) > 0     2.81      0.23     2.44     3.18        Inf
##    Post.Prob Star
## 1       0.70
## 2       1.00    *
## 3       1.00    *
## 4       0.82
## 5       0.15
## 6       1.00    *
## 7       0.01
## 8       0.00
## 9       0.09
## 10      1.00    *
## 11      1.00    *
## 12      1.00    *
## 13      1.00    *
## ---
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

```
## Hypothesis Tests for class b:
##                 Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio
## 1                (Age) < 0     0.00      0.01    -0.01     0.02       0.42
## 2             (Income) < 0     0.05      0.00     0.05     0.06       0.00
## 3              (CCAvg) < 0     0.19      0.05     0.10     0.27       0.00
## 4            (Mortgage) < 0     0.00      0.00     0.00     0.00       0.23
## 5  (Securities.Accou... < 0    -0.28      0.27    -0.73     0.14       5.83
```

```
## 6         (CD.Account1) < 0     2.48      0.30     1.98     2.98       0.00
## 7             (Online1) < 0    -0.46      0.20    -0.79    -0.13     104.26
## 8         (CreditCard1) < 0    -0.71      0.23    -1.11    -0.34    1332.33
## 9             (Family2) < 0    -0.31      0.23    -0.69     0.06      10.49
## 10            (Family3) < 0     1.55      0.23     1.19     1.93       0.00
## 11            (Family4) < 0     1.60      0.23     1.23     1.98       0.00
## 12         (Education2) < 0     2.53      0.23     2.16     2.93       0.00
## 13         (Education3) < 0     2.81      0.23     2.44     3.18       0.00
##    Post.Prob Star
## 1       0.30
## 2       0.00
## 3       0.00
## 4       0.18
## 5       0.85
## 6       0.00
## 7       0.99    *
## 8       1.00    *
## 9       0.91
## 10      0.00
## 11      0.00
## 12      0.00
## 13      0.00
## ---
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

The Posterior Probability is the posterior probability that the hypothesis is true, then we can state that Age, Mortgage and Securities.Account are not significant.

We can also plot credibility intervals for the coefficients.




```
## [1] "Assessment of the full Bayesian model."

## [1] "Correct Predictions: 429"

## [1] "Accuracy: 0.858"

## [1] "True Positive Rate: 0.911764705882353"

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## [1] "AUC: 0.947298661954049"

## [1] "Dummy Accuracy: 0.932"
```

The AUC, which is the coefficient we choose to compare the models, is sligthly lower than the one of the Glm model.

We also tried a 5-fold CV to compare the Bayesian and the classic Glm model, under-sampling on the train set, we got similar results, but the classic model is still better: Average AUC (glm): 0.963722250099581, Average AUC (brm): 0.958610378527511.

About the significance of the coefficients, we try to fit another model with less variables and compare the WAICs.

Let's try to modify the model cutting the variables that are not significant: Mortgage, Age, Securities.Accoun and compute again the WAIC.

```
capture.output({
# Fit the reduced Bayesian logistic regression model with the specified priors
# Define informative priors for each coefficient
prior_spec1 <- c(
  prior(normal(0.3, 0.3), class = "b", coef = "Income"),
```

```
  prior(normal(-1, 0.5), class = "b", coef = "Family2"),
  prior(normal(1, 0.5), class = "b", coef = "Family3"),
  prior(normal(1.5, 0.5), class = "b", coef = "Family4"),
  prior(normal(0.5, 0.5), class = "b", coef = "CCAvg"),
  prior(normal(1, 0.5), class = "b", coef = "Education2"),
  prior(normal(2, 0.5), class = "b", coef = "Education3"),
  prior(normal(0.5, 0.5), class = "b", coef = "CD.Account1")
)
# Fit the Bayesian logistic regression model with the specified priors
model1 <- brm(
  formula = Personal.Loan ~  Income + Family + CCAvg + Education +
    CD.Account + Online + CreditCard,
  data = balanced_data_train,
  family = bernoulli(link = "logit"),
  prior = prior_spec1,
  chains = 4, # controls parallel chains for convergence diagnostics.
  iter = 2000, # determines the total number of iterations per chain.
  warmup = 1000 # sets the number of warm-up iterations discarded before collecting posterior samples.
)

}, file = "/dev/null")
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: Personal.Loan ~ Income + Family + CCAvg + Education + CD.Account + Online + CreditCard
##    Data: balanced_data_train (Number of observations: 1784)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      -9.76      0.49   -10.75    -8.82 1.00     2759     2788
## Income          0.06      0.00     0.05     0.06 1.00     3372     3081
## Family2        -0.32      0.23    -0.76     0.12 1.00     3528     3145
## Family3         1.56      0.23     1.11     2.01 1.00     3615     2946
## Family4         1.61      0.22     1.19     2.05 1.00     3323     2976
## CCAvg           0.18      0.05     0.07     0.29 1.00     3849     2923
## Education2      2.53      0.23     2.08     3.00 1.00     3405     2881
## Education3      2.80      0.23     2.36     3.26 1.00     3126     2774
## CD.Account1     2.39      0.28     1.85     2.96 1.00     3376     2395
## Online1        -0.43      0.19    -0.80    -0.06 1.00     4005     3119
## CreditCard1    -0.68      0.23    -1.12    -0.24 1.00     3568     2977
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
##
## Computed from 4000 by 1784 log-likelihood matrix
##
##           Estimate   SE
## elpd_waic   -351.7 21.0
## p_waic         8.4  0.6
## waic         703.3 41.9
```

It is interesting to note that we get a lower WAIC, then we choose to take the reduced model and the variables we dropped are actually not significant.

Let's assess it.

```
## [1] "Assessment of the reduced Bayesian model."

## [1] "Correct Predictions: 430"

## [1] "Accuracy: 0.86"

## [1] "True Positive Rate: 0.911764705882353"

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## [1] "AUC: 0.949413027013378"

## [1] "Dummy Accuracy: 0.932"
```

We can see that the reduced model has a slightly better AUC than the full model, our choice to drop them was correct.

#RESULTS AND CONCLUSIONS

Family	Model	Unbalanced				Balanced			
		Accuracy	TPR	TNR	AUC	Accuracy	TPR	TNR	AUC
Logistic regression	Full Model	0.95820	0.67332	0.98829	0.96237	0.90640	0.89765	0.90730	0.96414
	Restricted Model	0.95980	0.67996	0.98917	0.96251	0.90680	0.89987	0.90750	0.96452
	Interaction Model	0.97740	0.84580	0.99138	0.98809	0.96160	0.93976	0.96393	0.98930
	Ridge Model	0.94860	0.51404	0.99403	0.96390	0.90380	0.88989	0.90500	0.96100
GAM	Full Model	0.98320	0.95010	0.98642	0.99414	0.98320	0.95087	0.98645	0.99432
	Restricted Model	0.98420	0.95690	0.98685	0.99418	0.98380	0.95514	0.98664	0.99435
	Interaction Model	0.98400	0.95477	0.98688	0.99354	0.98400	0.95514	0.98687	0.99410
SVM	Linear kernel	0.96000	0.91000	0.96000	0.81000	0.91000	0.51000	0.99000	0.90000
	Radial kernel	0.98000	0.96000	0.98000	0.92000	0.98000	0.91000	0.99000	0.93000
Random forest	Mtry = 7	0.98660	0.99920	0.88480	0.94204	0.98790	0.99550	0.92730	0.96136
Bayesian logistic regression	Full model					0.85600	0.91176	0.89241	0.94770
	Restricted model					0.86000	0.93753	0.91176	0.94887

By looking at the global results we can deduce that the best model in terms of AUC and accuracy is the GAM, in particular the interaction model, followed by the logistic regression models. In conclusion, even

though we had to analyze an unbalanced dataset, multiple tools and techniques have allowed us to obtain successful results with all the five models in which the dataset was analyzed.