GroupE Project

A. Gottardi, E. Corrolezzis, A. Minutolo, L.F. Palacios Flores

“Doubt the data until the data leave no room for doubt.” - Henri Poincaré

Problem statement

The dataset contains the data of the clients of an Insurance company that has provided Health Insurance. Our goal is to analyze the relationship between the features and the probability of the customers buying a vehicle insurance. Now, in order to predict whether the customer would be interested in Vehicle insurance, we have information about demographics (gender, age, region code type), Vehicles (Vehicle Age, Damage), Policy ins(Premium, sourcing channel) etc.

Our client is an Insurance company that has provided Health Insurance to its customers. Now they need the help in building a model to predict whether the policyholders (customers) from the past year will also be interested in Vehicle Insurance provided by the company.

An insurance policy is an arrangement by which a company undertakes to provide a guarantee of compensation for specified loss, damage, illness, or death in return for the payment of a specified premium. A premium is a sum of money that the customer needs to pay regularly to an insurance company for this guarantee.

Building a model to predict whether a customer would be interested in Vehicle Insurance is extremely helpful for the company because it can then accordingly plan its communication strategy to reach out to those customers and optimize its business model and revenue.

Data

We had three datasets to analyze ‘train.csv’, ‘test.csv’ and ‘sample.csv’. Among these datasets we only analyzed the first one. The ‘test.csv’ dataset lacked the ‘Response’ variable and the ‘sample.csv’ file contained observations of this variable but one for one category, making them unusable.

Our dataset is composed of the following variables:

Variable	Definition	Type
id	Unique ID for the customer	Numeric
Gender	Gender of the customer	Categorical
Age	Age of the customer	Numeric
Driving_License	0 : Customer does not have DL, 1 : Customer already has DL	Binary
Region_Code	Unique code for the region of the customer	Categorical
Previously_Insured	1 : Customer already has Vehicle Insurance, 0 : Customer doesn’t have Vehicle Insurance	Binary
Vehicle_Age	Age of the Vehicle	Categorical
Vehicle_Damage	1 : Customer got his/her vehicle damaged in the past. 0 : Customer didn’t get his/her vehicle damaged in the past.	Binary
Annual_Premium	The amount customer needs to pay as premium in the year	Numeric
Policy_Sales_Channel	Anonymized Code for the channel of outreaching to the customer ie. Different Agents, Over Mail, Over Phone, In Person, etc.	Categorical
Vintage	Number of Days, Customer has been associated with the company	Numeric
Response	1 : Customer is interested, 0 : Customer is not interested	Binary

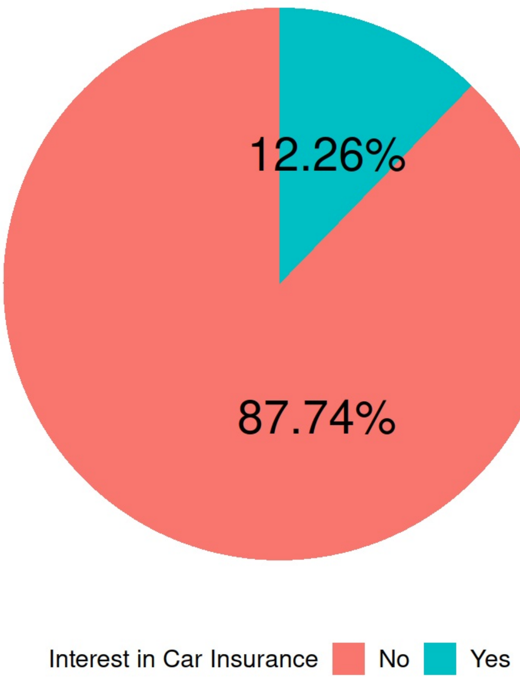
The first step is trying to get some insights about the dataset by plotting and analyzing the data. We present the barplots for the categorical variables and the density plots for the numerical ones.

The ‘id’ variable is just a discrete ordered variable with uniform distribution. Therefore, we just removed it from our analysis.

Exploratory Data Analysis

‘Response’ variable

The proportion for the categories of the response variable are the following:

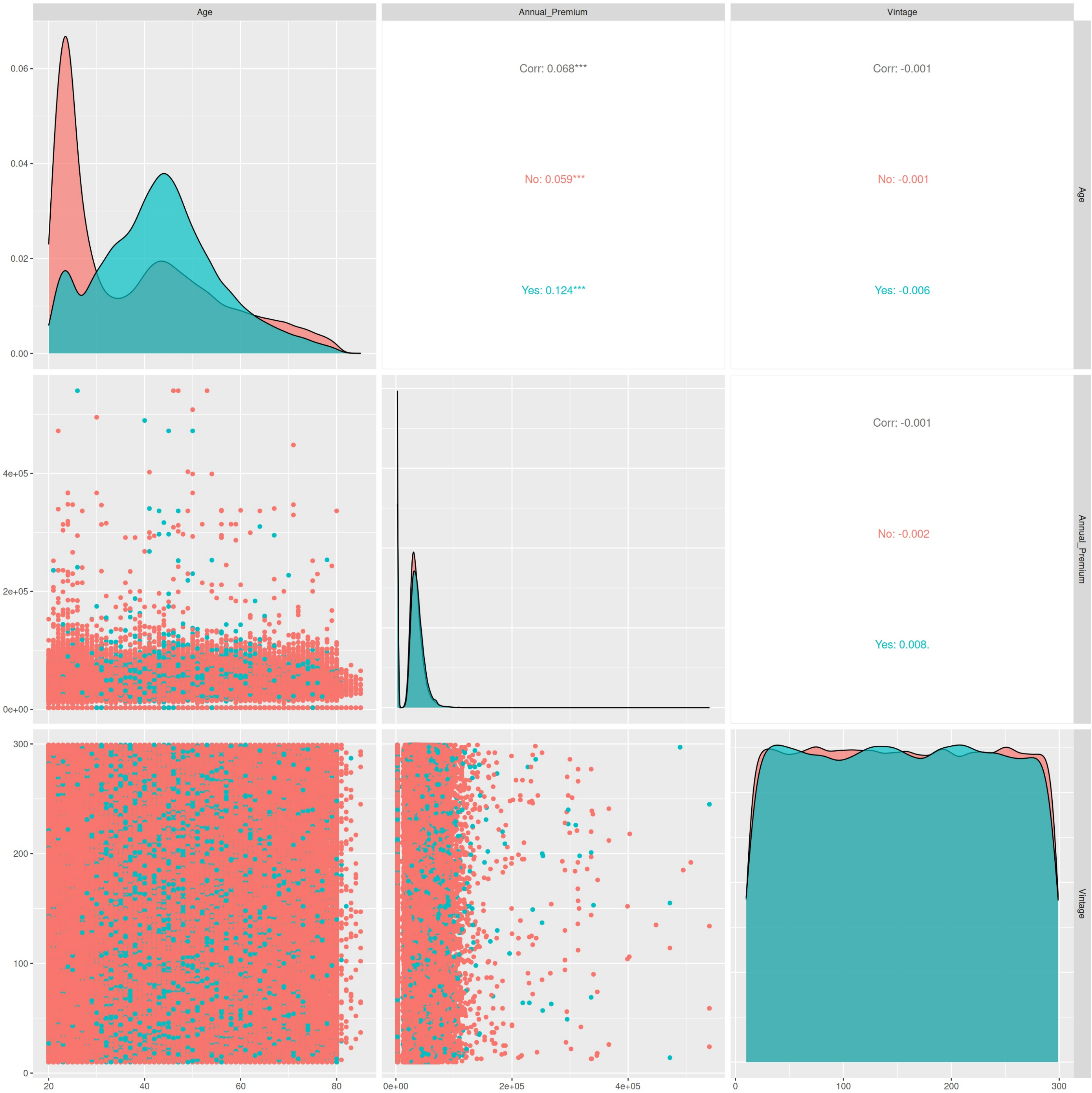


As we can see the dataset is unbalanced and the imbalance ratio is:

```
## [1] "IR: 7.159045"
```

This degree of imbalance is considered to be weak with respect to the reference level of 10 for slight imbalance, thus we decided to not perform any procedure to correct the imbalance.

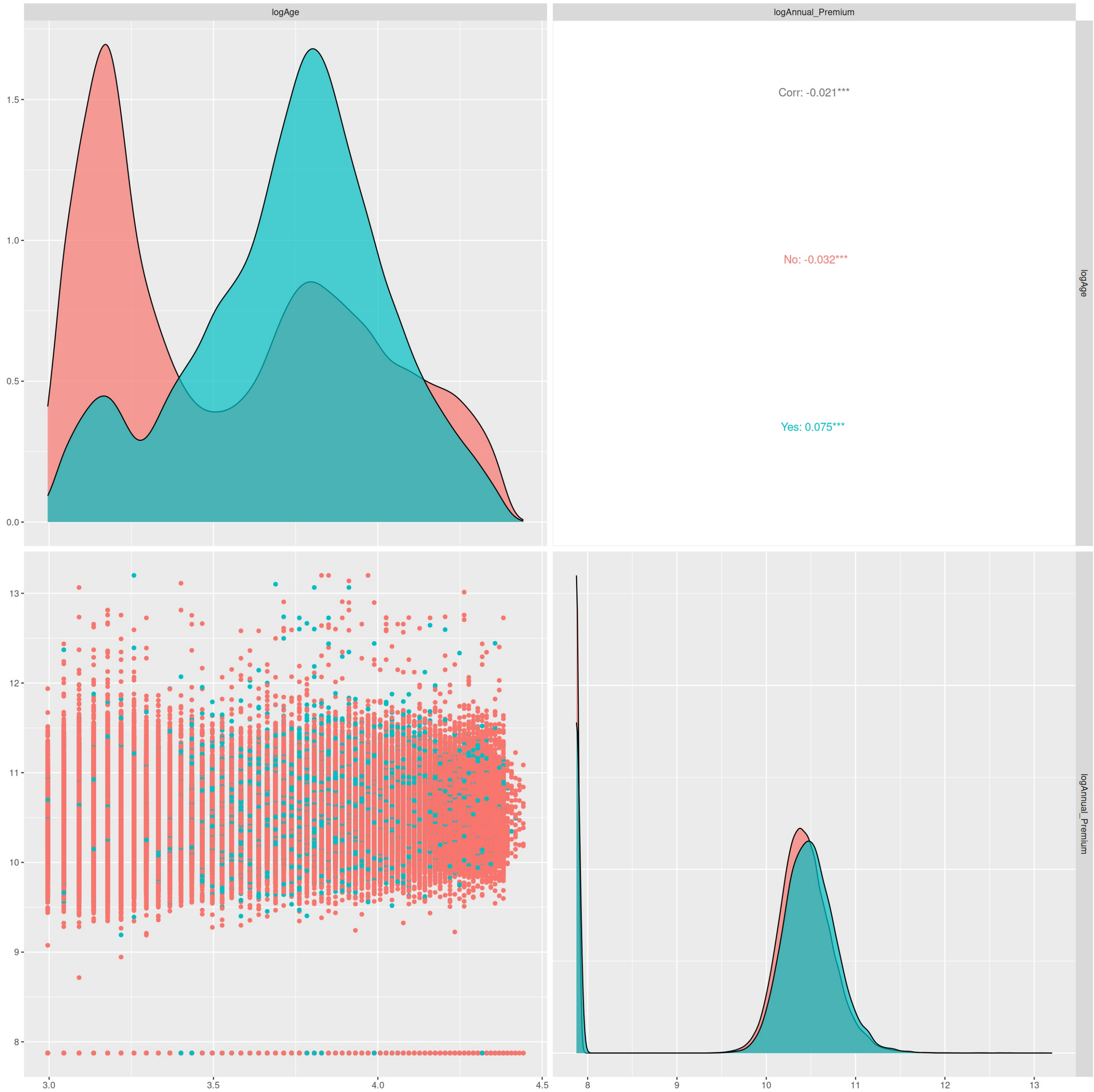
Numerical variables



The distribution of the Age variable with respect to the Response variable shows that the majority of the customers who are interested in car insurance are middle-aged (between 30 and 60 years old), which coincides with the age people are more likely to own a car. The customers not interested in acquiring a car insurance policy are mostly distributed among younger people and some middle-aged adults in their 50s. The ditribution is skewed to the right.

The plot for Annual Premium suggests that the costs of the car insurance policy is independent of the interest if the customers to buy the product. It has a highly right-skewed distribution, with most of the data concentrated on the lower end of the premium scale and a long tail extending to higher premium values. The lower tail shows a high values as a consequence of entry level health insurance policy as expected.

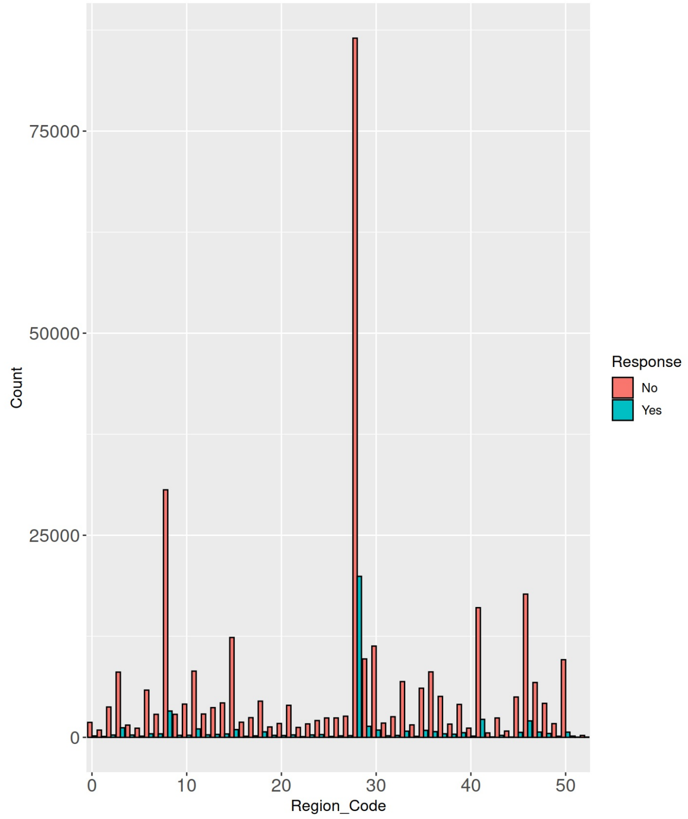
Since both Age and Annual Premium are skewed to the right, we considered to apply a logarithm transformation for both the variables.

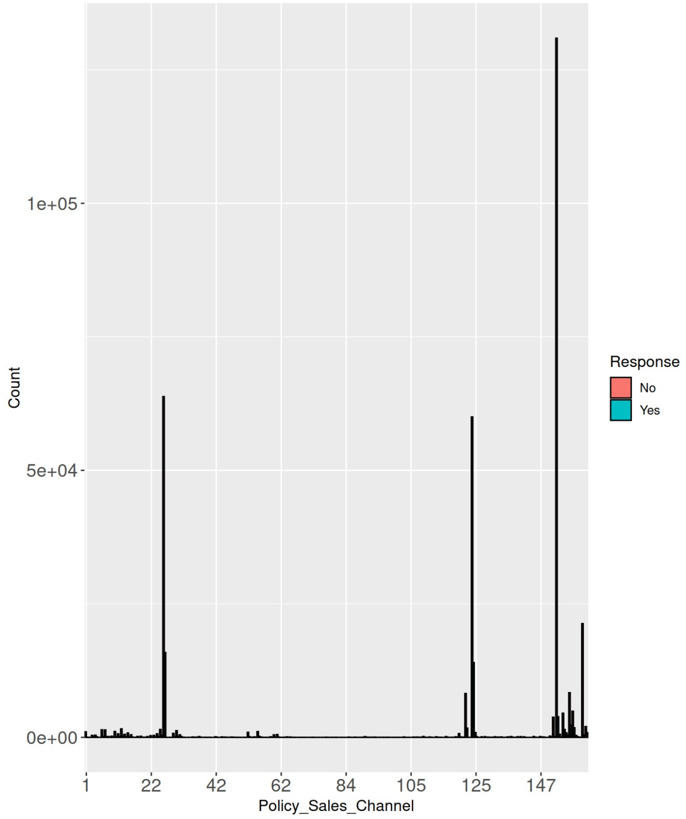


The plot for Vintage shows a nearly uniform distribution, with a slight increase in frequency towards the middle range of the Vintage variable. It may not be significant in the explanation of the Response.

The variables show negligible linear correlation between them, which is clearly shown in the scatter plot and in the correlation coefficients.

Categorical variables



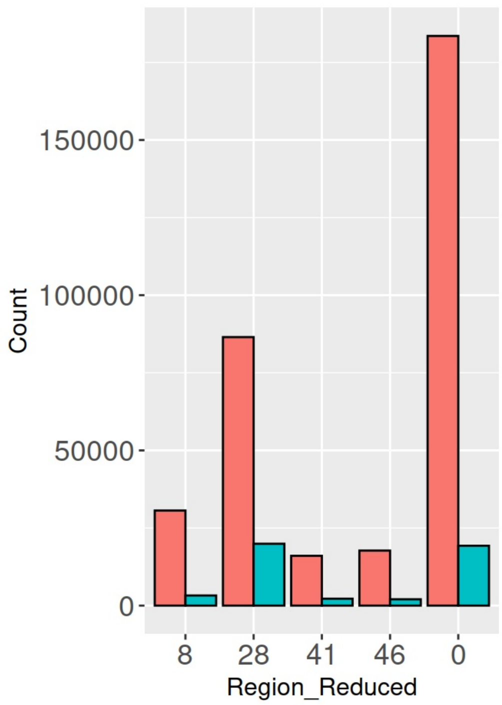


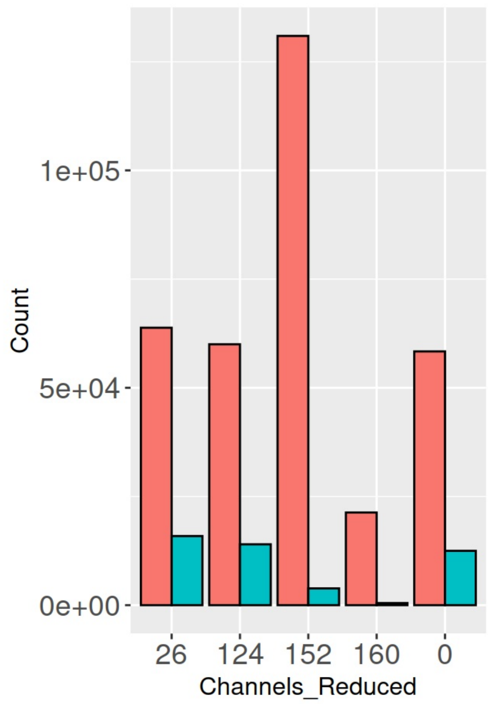
Regarding the variable Region_Code, we can notice that the vast majority of the customers are from region 28. The customers from region 28 are also the ones who are most interested in car insurance. Almost half of the customers are distributed among regions 8, 28, 41, 46 accounting for ~47% of the total customers. Since this variable has a lot of labels with low frequency, we decided to consider only the major four ones mentioned above and an additional one with the remaining labels as a unique category.

Also in Policy_Sales_Channel, there are four categories more frequent than others: Channels 26, 124, 152 and 160 alone account for more than 80% of the customers. Channels 26 and 124 are the ones with the highest percentage of customers interested in the product. Only about 20% of the customers interested in the product are distributed in the rest of the channels of outreach. As we did for Region_Code, we grouped the remaining less frequent categories as one.

After the grouping:

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

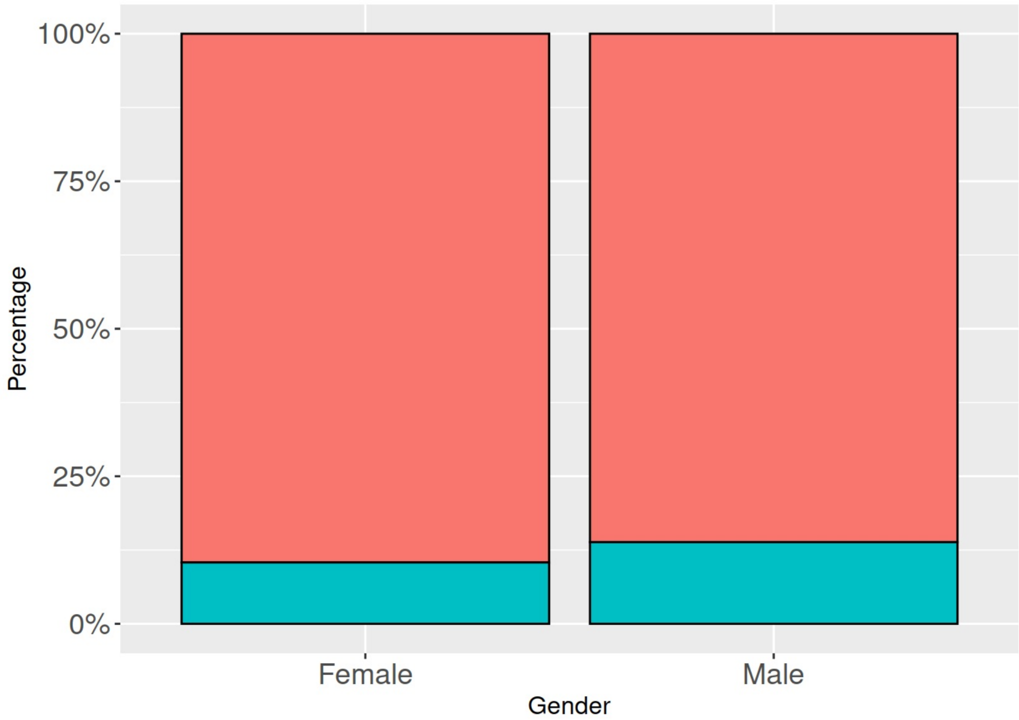




The data for Gender shows a similar trend for the interest of males and females customers in the product. However, there is a statistically significant difference in their proportions.

```
prop.test(table(train_reduced$Gender, train_reduced$Response))
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  table(train_reduced$Gender, train_reduced$Response)
## X-squared = 1047.7, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.03243785 0.03657948
## sample estimates:
##    prop 1    prop 2
## 0.8960976 0.8615889
```



MODELS

After exploring the data, now we proceed with the fitting and assessment of different models for binary classification. For training and testing the models we performed a static train/test split with 70% of train set and 30% of test set.

In order to understand which variables are more significant in the explanation of the response variable, we analyzed nested models with different combinations of selected explanatory variables.

GLM

stepAIC

The first approach we used consists of using the function stepAIC() from the MASS package to find the best combination of predictors with respect to AIC. The stepAIC() function must be applied to the full model, which serves as the starting point for the variable selection process. We chose the 'both' direction, that considers both adding and removing variables from the model.

```
full_model <- glm(Response ~ Gender + Age + Driving_License + Previously_Insured +
    Vehicle_Age + Vehicle_Damage + Annual_Premium + Vintage +
    Channels_Reduced + Region_Reduced, data = unbalanced_train, family = binomial)
stepAIC(full_model, direction = 'both')
```

```
## Start:  AIC=144269.1
## Response ~ Gender + Age + Driving_License + Previously_Insured + 
##     Vehicle_Age + Vehicle_Damage + Annual_Premium + Vintage + 
##     Channels_Reduced + Region_Reduced
## 
##                      Df Deviance    AIC
## - Vintage             1   144233 144267
## <none>                    144233 144269
## - Annual_Premium      1   144250 144284
## - Gender              1   144271 144305
## - Driving_License     1   144278 144312
## - Region_Reduced      4   144421 144449
## - Vehicle_Age         2   144580 144612
## - Age                 1   145993 146027
## - Channels_Reduced    4   146476 146504
## - Vehicle_Damage      1   148208 148242
## - Previously_Insured  1   149056 149090
## 
## Step:  AIC=144267.2
## Response ~ Gender + Age + Driving_License + Previously_Insured + 
##     Vehicle_Age + Vehicle_Damage + Annual_Premium + Channels_Reduced + 
##     Region_Reduced
## 
##                      Df Deviance    AIC
## <none>                    144233 144267
## + Vintage             1   144233 144269
## - Annual_Premium      1   144250 144282
## - Gender              1   144271 144303
## - Driving_License     1   144278 144310
## - Region_Reduced      4   144421 144447
## - Vehicle_Age         2   144580 144610
## - Age                 1   145994 146026
## - Channels_Reduced    4   146476 146502
## - Vehicle_Damage      1   148208 148240
## - Previously_Insured  1   149057 149089
```

```
## 
## Call:  glm(formula = Response ~ Gender + Age + Driving_License + Previously_Insured + 
##     Vehicle_Age + Vehicle_Damage + Annual_Premium + Channels_Reduced + 
##     Region_Reduced, family = binomial, data = unbalanced_train)
## 
## Coefficients:
##          (Intercept)             GenderMale                    Age  
##           -3.363e+00              8.284e-02             -2.644e-02  
##    Driving_LicenseYes  Previously_InsuredYes   Vehicle_Age> 2 Years  
##            1.106e+00             -3.855e+00              6.480e-01  
##   Vehicle_Age1-2 Year      Vehicle_DamageYes         Annual_Premium  
##            4.614e-01              1.975e+00              1.574e-06  
##   Channels_Reduced124    Channels_Reduced152    Channels_Reduced160  
##           -1.702e-01             -1.209e+00             -2.208e+00  
##     Channels_Reduced0       Region_Reduced28       Region_Reduced41  
##           -2.713e-01              2.573e-01              4.318e-01  
##      Region_Reduced46        Region_Reduced0  
##            1.484e-01              1.328e-01  
## 
## Degrees of Freedom: 266775 Total (i.e. Null);  266759 Residual
## Null Deviance:       198300 
## Residual Deviance: 144200    AIC: 144300
```

Performing the stepAIC() function we confirmed that the variable Vintage is not useful for the model, because of its uniform distribution. The stepAIC() function stops when the ranking of models built by removing and adding one variable at a time has an AIC greater than the default model. Hence, this procedure doesn't allow us to obtain a simple model based on the Occam's razor. Consequentially we also implemented a procedure to obtain a reduced model based on idea of the stepAIC().

```
best_model<-glm(Response ~ Gender + Age + Driving_License + Previously_Insured  + Vehicle_Age + Vehicle_Damage + 
Annual_Premium +
    Channels_Reduced + Region_Reduced, data = unbalanced_train, family = binomial)
summary(best_model)
```

```
##
## Call:
## glm(formula = Response ~ Gender + Age + Driving_License + Previously_Insured +
##     Vehicle_Age + Vehicle_Damage + Annual_Premium + Channels_Reduced +
##     Region_Reduced, family = binomial, data = unbalanced_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3554  -0.6389  -0.0496  -0.0289   4.1670
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -3.363e+00  2.012e-01 -16.713  < 2e-16 ***
## GenderMale             8.284e-02  1.345e-02   6.158 7.35e-10 ***
## Age                   -2.644e-02  6.460e-04 -40.928  < 2e-16 ***
## Driving_LicenseYes     1.106e+00  1.918e-01   5.768 8.01e-09 ***
## Previously_InsuredYes -3.855e+00  9.547e-02 -40.380  < 2e-16 ***
## Vehicle_Age> 2 Years   6.480e-01  3.613e-02  17.938  < 2e-16 ***
## Vehicle_Age1-2 Year    4.614e-01  2.774e-02  16.635  < 2e-16 ***
## Vehicle_DamageYes      1.975e+00  4.085e-02  48.334  < 2e-16 ***
## Annual_Premium         1.574e-06  3.774e-07   4.170 3.04e-05 ***
## Channels_Reduced124   -1.702e-01  1.687e-02 -10.091  < 2e-16 ***
## Channels_Reduced152   -1.209e+00  3.234e-02 -37.393  < 2e-16 ***
## Channels_Reduced160   -2.208e+00  6.253e-02 -35.315  < 2e-16 ***
## Channels_Reduced0     -2.713e-01  1.752e-02 -15.487  < 2e-16 ***
## Region_Reduced28       2.573e-01  2.648e-02   9.716  < 2e-16 ***
## Region_Reduced41       4.318e-01  3.944e-02  10.950  < 2e-16 ***
## Region_Reduced46       1.484e-01  3.959e-02   3.749 0.000178 ***
## Region_Reduced0        1.328e-01  2.660e-02   4.994 5.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 198262  on 266775  degrees of freedom
## Residual deviance: 144233  on 266759  degrees of freedom
## AIC: 144267
##
## Number of Fisher Scoring iterations: 9
```

```
exp(best_model$coefficients)
```

```
##           (Intercept)            GenderMale                   Age
##            0.03461490            1.08636867            0.97390775
##    Driving_LicenseYes Previously_InsuredYes  Vehicle_Age> 2 Years
##            3.02374597            0.02117363            1.91180217
##   Vehicle_Age1-2 Year     Vehicle_DamageYes        Annual_Premium
##            1.58628562            7.20333561            1.00000157
##   Channels_Reduced124   Channels_Reduced152   Channels_Reduced160
##            0.84346024            0.29835782            0.10990809
##     Channels_Reduced0      Region_Reduced28      Region_Reduced41
##            0.76237401            1.29337639            1.54008019
##      Region_Reduced46       Region_Reduced0
##            1.15998812            1.14202904
```

These results suggest that:

- Male customers are 9% more interested in car insurance than female customers.
- There is a slight decrease in the odds of interest in the product with each year of increasing customer age.
- Customers with a driver’s license are 3 times more interested in the product than those without a driver’s license.
- There is a dramatic decrease in the interest of the customers in the product for those who previously had their cars insured. The company should focus on improving its services because current policyholders lose their interest by 98% with respect to those without car insurance.
- The results show that customers with older cars show more interest than customers with new cars. Around 2 times for cars more than 2 years old compared to new cars.
- The interest in the Car Insurance Policy of the customers with health insurance who had their car damaged in the past is more than 7 times that of those who haven´t. This is expected and a variable the company should focus on to estimate the risk associated with these customers.
- The amount the customer needs to pay as a premium in the year doesn’t seem to be associated with any increase or decrease in the odds of the event.
- There is more interest in the product, from ~14% to ~54% more interest, for customers from Regions 28, 41, 46, and 0 (combination of low frequent regions) with respect to Region 8.

- The outreach Channel 26 (base category in the model) gets from 16% to 90% more interested customers in comparison with Channel 26. Channel 124 attracts a lot of clients as well and the company could attempt to increase its influence in this channel.
- The variable Vintage (number of days the customers have been associated with the company) was completely removed from the model because of low significance with respect to the AIC.

Nested models

This procedure builds a ranking of variables, similar to the one of the stepAIC(), meaning to create models by removing one variable at a time and sorting the variables by AIC. After that, we built nested models by adding one variable at a time based on the ranking of variables mentioned before. The aim is to find the optimal model based on both AIC and the Occam's razor.

```
#*LOAD THE DATA-----------------------------------------------------------------
# Define the path to the datasets
current_path <- dirname(rstudioapi::getActiveDocumentContext()$path)
datasets_dir <- paste(current_path,"datasets", sep = "/")

#*----------------------------------------------------------------------
#*FUNCTIONS
#*----------------------------------------------------------------------

# * This function assumes that `Policy_Sales_Channel` and `Region_Code` have been removed from the data
ranking_nested_models <- function(train_data, test_data, use_model = "glm", use_log = TRUE, use_splines = FALSE)
{
  if(use_model == "glm" && use_log == TRUE){
    numeric_variables <- c("I(log(Age))", "I(log(Annual_Premium))")
  } else if (use_model == "glm" && use_log == FALSE){
    numeric_variables <- c("Age", "Annual_Premium")
  } else if (use_model == "gam" && use_log == TRUE && use_splines == TRUE){
    numeric_variables <- c("s(I(log(Age)))", "s(I(log(Annual_Premium)))")
  } else if (use_model == "gam" && use_log == TRUE && use_splines == FALSE){
    numeric_variables <- c("I(log(Age))", "I(log(Annual_Premium))")
  } else if (use_model == "gam" && use_log == FALSE && use_splines == TRUE){
    numeric_variables <- c("s(Age)", "s(Annual_Premium)")
  } else if (use_model == "gam" && use_log == FALSE && use_splines == FALSE){
    numeric_variables <- c("Age", "Annual_Premium")
  }

  #*VARIABLES IMPORTANCE RANKING --------------------------------------------------

  # Sort variables by importance wrt AIC
  # We remove one variable at a time and by decreasing AIC we get the most important variables
  # i.e., the variables that when removed increase the AIC is important
  predictors <- colnames(train_data)
  predictors <- predictors[predictors != 'Response']
  ranking_variables_models <- list()
  sum_variables <- paste(predictors, collapse = " + ")

  for (predictor in predictors){
    if (predictor == "Age"){
      formula_string <- paste("Response ~", sum_variables ,"- Annual_Premium -", predictor, "+", numeric_variable
s[2])

    } else if (predictor == "Annual_Premium"){
      formula_string <- paste("Response ~", sum_variables ,"- Age -", predictor, "+", numeric_variables[1])
    } else {
      formula_string <- paste("Response ~", sum_variables ,"- Age - Annual_Premium -", predictor, "+", paste(nume
ric_variables, collapse = " + "))

    }
    model_formula <- as.formula(formula_string)

    if(use_model == "glm"){
      model <- glm(model_formula, data = train_data, family = binomial)
    } else if (use_model == "gam"){
      model <- gam(model_formula, data = train_data, family = binomial)
    }

    ranking_variables_models[[predictor]] <- model
  }

  # Compute AIC values
  ranking_variables_aic_values <- sapply(ranking_variables_models, AIC)

  # Sort variables by AIC values
  df_ranking_variables_aic <- data.frame(VariableRemoved = predictors, AIC = ranking_variables_aic_values)
  # Assuming df is your DataFrame
  df_sorted_ranking_variables_aic <- df_ranking_variables_aic[order(df_ranking_variables_aic$AIC, decreasing=TRUE
```

```r
), ]
  # df_sorted_ranking_variables_aic

  #* NESTED MODELS ----------------------------------------------------------------
  variables_order <- df_sorted_ranking_variables_aic$VariableRemoved
  # variables_order
  variables_nested <- c()
  nested_models <- list()

  for (variable in variables_order) {
    if (variable == "Age"){
      variables_nested <- c(variables_nested, numeric_variables[1])
    } else if (variable == "Annual_Premium"){
      variables_nested <- c(variables_nested, numeric_variables[2])
    } else {
      variables_nested <- c(variables_nested, variable)
    }
    formula_string <- paste("Response", "~", paste(variables_nested, collapse = " + "))
    print(formula_string)
    model_formula <- as.formula(formula_string)
    if(use_model == "glm"){
      model <- glm(model_formula, data = train_data, family = binomial)
    } else if (use_model == "gam"){
      model <- gam(model_formula, data = train_data, family = binomial)
    }

    nested_models[[variable]] <- model
  }

  # Compute AIC values
  raking_nested_models_aic_values <- sapply(nested_models, AIC)
  df_ranking_nested_models_aic <- data.frame(Model_Name = variables_order, AIC = raking_nested_models_aic_values)
  # df_ranking_nested_models_aic

  # Sort variables by AIC values
  df_sorted_ranking_nested_models_aic <- df_ranking_nested_models_aic[order(df_ranking_nested_models_aic$AIC, dec
reasing=TRUE), ]

  # COMPUTE AUC AND ACCURACY FOR EACH MODEL -------------------------------------

  # Apply models_assessment function to each model using map
  results_list <- map(nested_models, ~models_assessment(.x, test_data))

  # Compute AUC values
  auc_values <- list()
  accuracy_values <- list()
  tpr_values <- list()
  fpr_values <- list()
  tnr_values <- list()
  fnr_values <- list()
  precision_values <- list()
  threshold_values <- list()

  for (i in 1:length(results_list)){
    auc_values <- c(auc_values, as.numeric(results_list[[i]][1]))
    accuracy_values <- c(accuracy_values, as.numeric(results_list[[i]][2]))
    tpr_values <- c(tpr_values, as.numeric(results_list[[i]][3]))
    fpr_values <- c(fpr_values, as.numeric(results_list[[i]][4]))
    tnr_values <- c(tnr_values, as.numeric(results_list[[i]][5]))
    fnr_values <- c(fnr_values, as.numeric(results_list[[i]][6]))
    precision_values <- c(precision_values, as.numeric(results_list[[i]][7]))
    threshold_values <- c(threshold_values, as.numeric(results_list[[i]][8]))
  }

  result_df <- data.frame(
    Model_Name = df_ranking_nested_models_aic$Model_Name,
    AIC = df_ranking_nested_models_aic$AIC,
    AUC = unlist(auc_values),
    Accuracy = unlist(accuracy_values),
    TPR = unlist(tpr_values),
    FPR = unlist(fpr_values),
    TNR = unlist(tnr_values),
    FNR = unlist(fnr_values),
    Precision = unlist(precision_values),
    Threshold = unlist(threshold_values)
  )

  # Return ranking of variables, | dataframe
  # results | dataframe
```

```r
  # and nested models | list
  return(list(Ranking_Variables = df_sorted_ranking_variables_aic, Results = result_df, Models = nested_models))

}


#*FUNCTION TO PERFORM THE MODEL ASSESSMENT -------------------------------------
models_assessment <- function(model, test_data){
  # Predict probabilities
  probabilities <- predict(model, newdata = subset(test_data, select = -Response), type = "response")

  # Compute ROC curve
  roc_curve <- roc(test_data$Response, probabilities)

  # Calculate AUC
  auc_score <- auc(roc_curve)

  # Find optimal threshold using Youden's J statistic
  youdens_j <- coords(roc_curve, "best", best.method = "youden")
  optimal_threshold <- youdens_j$threshold


  # Obtain predicted classes based on the optimal threshold
  predicted_classes <- ifelse(probabilities > optimal_threshold, "Yes", "No")

  # Create the confusion matrix
  conf_matrix <- table(Actual = test_data$Response, Predicted = predicted_classes)

  conf_matrix_prop <- prop.table(conf_matrix, margin = 1)

  # Calculate accuracy
  accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)

  # Calculate true positive rate
  tpr <- conf_matrix[2, 2] / sum(conf_matrix[2, ])

  # Calculate false positive rate
  fpr <- conf_matrix[1, 2] / sum(conf_matrix[1, ])

  # Calculate true negative rate
  tnr <- conf_matrix[1, 1] / sum(conf_matrix[1, ])

  # Calculate false negative rate
  fnr <- conf_matrix[2, 1] / sum(conf_matrix[2, ])

  # Calculate precision
  precision <- conf_matrix[2, 2] / sum(conf_matrix[, 2])

  return(list(auc_score = auc_score, accuracy = accuracy,
              tpr = tpr, fpr = fpr, tnr = tnr, fnr = fnr,
              precision = precision, optimal_threshold = optimal_threshold))

}
```

```r
result_glm <- ranking_nested_models(unbalanced_train, unbalanced_test, use_model = "glm", use_log = FALSE, use_sp
lines = FALSE)
```

```
## [1] "Response ~ Previously_Insured"
## [1] "Response ~ Previously_Insured + Vehicle_Damage"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + Age"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + Age + Vehicle_Age"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + Age + Vehicle_Age + Region_Reduced"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + Age + Vehicle_Age + Region_Reduced +
Driving_License"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + Age + Vehicle_Age + Region_Reduced +
Driving_License + Gender"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + Age + Vehicle_Age + Region_Reduced +
Driving_License + Gender + Annual_Premium"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + Age + Vehicle_Age + Region_Reduced +
Driving_License + Gender + Annual_Premium + Vintage"
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
result_glm$Results
```

```
##            Model_Name      AIC       AUC  Accuracy       TPR       FPR
## 1  Previously_Insured 155905.7 0.7585627 0.5786343 0.9972279 0.4801025
## 2      Vehicle_Damage 150595.1 0.7890232 0.6372613 0.9781790 0.4105761
## 3    Channels_Reduced 146317.5 0.8263009 0.6568007 0.9686545 0.3869584
## 4                 Age 144931.4 0.8420756 0.6593197 0.9670197 0.3838566
## 5         Vehicle_Age 144565.1 0.8429728 0.6582177 0.9679437 0.3852430
## 6      Region_Reduced 144360.5 0.8442799 0.7067688 0.9070296 0.3213317
## 7     Driving_License 144318.7 0.8444795 0.7077397 0.9065321 0.3201548
## 8              Gender 144282.5 0.8444985 0.7109059 0.9023385 0.3159559
## 9      Annual_Premium 144267.2 0.8447612 0.6974189 0.9202502 0.3338486
## 10            Vintage 144269.1 0.8447582 0.6974627 0.9201791 0.3337888
##          TNR         FNR Precision  Threshold
## 1  0.5198975 0.002772052 0.2256824 0.11316378
## 2  0.5894239 0.021821025 0.2505462 0.14352241
## 3  0.6130416 0.031345511 0.2599474 0.09316676
## 4  0.6161434 0.032980311 0.2611725 0.09821276
## 5  0.6147570 0.032056294 0.2606615 0.10396336
## 6  0.6786683 0.092970360 0.2837102 0.13822661
## 7  0.6798452 0.093467908 0.2843448 0.13837295
## 8  0.6840441 0.097661525 0.2860910 0.14354108
## 9  0.6661514 0.079749805 0.2789100 0.13484162
## 10 0.6662112 0.079820883 0.2789305 0.13483145
```

The ranking of variables shown has similar results to the one of the stepAIC(). We also performed an Anova test in order to understand better the improvements of the nested models.

Anova test:

```
# Perform ANOVA on full model

anova_values <- anova(result_glm$Models$Vintage, test = "Chisq")
anova_values
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Response
##
## Terms added sequentially (first to last)
##
##
##                    Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                              266775     198262
## Previously_Insured  1    42360    266774     155902 < 2.2e-16 ***
## Vehicle_Damage      1     5313    266773     150589 < 2.2e-16 ***
## Channels_Reduced    4     4286    266769     146303 < 2.2e-16 ***
## Age                 1     1388    266768     144915 < 2.2e-16 ***
## Vehicle_Age         2      370    266766     144545 < 2.2e-16 ***
## Region_Reduced      4      213    266762     144333 < 2.2e-16 ***
## Driving_License     1       44    266761     144289 3.574e-11 ***
## Gender              1       38    266760     144250 6.436e-10 ***
## Annual_Premium      1       17    266759     144233 3.227e-05 ***
## Vintage             1        0    266758     144233    0.7003
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Observing the AIC and the anova test we noticed that there is not a significant decrease in the AIC or a significant improvement in deviance in the last three models, hence we decided to consider the nested model up to Region_Reduced.

```
summary(result_glm$Models$Region_Reduced)
```

```
##
## Call:
## glm(formula = model_formula, family = binomial, data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2551  -0.6412  -0.0497  -0.0291   4.1695
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -2.1557143  0.0562909 -38.296  < 2e-16 ***
## Previously_InsuredYes -3.8526359  0.0954613 -40.358  < 2e-16 ***
## Vehicle_DamageYes      1.9779266  0.0408438  48.427  < 2e-16 ***
## Channels_Reduced124   -0.1667655  0.0168539  -9.895  < 2e-16 ***
## Channels_Reduced152   -1.2152184  0.0322983 -37.625  < 2e-16 ***
## Channels_Reduced160   -2.2186733  0.0624895 -35.505  < 2e-16 ***
## Channels_Reduced0     -0.2833708  0.0173494 -16.333  < 2e-16 ***
## Age                   -0.0264099  0.0006419 -41.146  < 2e-16 ***
## Vehicle_Age> 2 Years   0.6560095  0.0360763  18.184  < 2e-16 ***
## Vehicle_Age1-2 Year    0.4626961  0.0276697  16.722  < 2e-16 ***
## Region_Reduced28       0.2585656  0.0264138   9.789  < 2e-16 ***
## Region_Reduced41       0.4242004  0.0393852  10.771  < 2e-16 ***
## Region_Reduced46       0.1349696  0.0394571   3.421 0.000625 ***
## Region_Reduced0        0.1156645  0.0262731   4.402 1.07e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 198262  on 266775  degrees of freedom
## Residual deviance: 144333  on 266762  degrees of freedom
## AIC: 144361
##
## Number of Fisher Scoring iterations: 9
```

As we can see from the summary, all the variables have a p-value close to 0, hence they are all statistically significant. We can compute the exponential of these coefficients to obtain the odds ratio:

```
exp(result_glm$Models$Region_Reduced$coefficients)
```

```
##          (Intercept) Previously_InsuredYes     Vehicle_DamageYes
##           0.11582043            0.02122372            7.22774119
##   Channels_Reduced124   Channels_Reduced152   Channels_Reduced160
##           0.84639804            0.29664521            0.10875330
##     Channels_Reduced0                   Age  Vehicle_Age> 2 Years
##           0.75324040            0.97393575            1.92708700
##   Vehicle_Age1-2 Year      Region_Reduced28      Region_Reduced41
##           1.58835054            1.29507105            1.52836787
##      Region_Reduced46       Region_Reduced0
##           1.14450194            1.12261915
```

we can comment these results as follows:

- The odds of a customer being interested in Vehicle insurance are 7 times higher for those who have had a vehicle damage compared to those who didn't.
- Compared to region 8, the odds of a customer being interested in Vehicle insurance are higher for those who live in Region 28, 41, 46 or 0.
- The odds of a customer being interested in Vehicle insurance get higher as the age of the vehicle increases.
- Compared to channel 26, the odds of a customer being interested in vehicle insurance are lower for those having a channel of outreaching with code 124, 160, 152 or 0.
- The odds of a customer being interested in vehicle insurance get lower for:
    - older customers;
    - customers that have been previously insured.

We also analyzed the model including a logarithm transformation for the Age:

```
result_glm_log <- ranking_nested_models(unbalanced_train, unbalanced_test, use_model = "glm", use_log = TRUE, use
_splines = FALSE)
```

```
## [1] "Response ~ Previously_Insured"
## [1] "Response ~ Previously_Insured + Vehicle_Damage"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age))"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age)) + Vehicle_Age"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age)) + Vehicle_Age + Region_Re
duced"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age)) + Vehicle_Age + Region_Re
duced + Driving_License"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age)) + Vehicle_Age + Region_Re
duced + Driving_License + Gender"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age)) + Vehicle_Age + Region_Re
duced + Driving_License + Gender + I(log(Annual_Premium))"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age)) + Vehicle_Age + Region_Re
duced + Driving_License + Gender + I(log(Annual_Premium)) + Vintage"
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
result_glm_log$Results
```

```
##             Model_Name      AIC       AUC  Accuracy       TPR       FPR
## 1  Previously_Insured 155905.7 0.7585627 0.5786343 0.9972279 0.4801025
## 2      Vehicle_Damage 150595.1 0.7890232 0.6372613 0.9781790 0.4105761
## 3    Channels_Reduced 146317.5 0.8263009 0.6568007 0.9686545 0.3869584
## 4                 Age 145448.6 0.8410972 0.6593022 0.9670197 0.3838766
## 5         Vehicle_Age 145124.1 0.8416569 0.6578503 0.9679437 0.3856619
## 6      Region_Reduced 144922.3 0.8426380 0.7072061 0.9026939 0.3202246
## 7     Driving_License 144868.9 0.8428380 0.7082032 0.9027649 0.3190976
## 8              Gender 144832.7 0.8427676 0.7097164 0.9002061 0.3170131
## 9      Annual_Premium 144804.0 0.8431696 0.6770661 0.9436349 0.3603387
## 10            Vintage 144805.9 0.8431671 0.6787717 0.9415026 0.3580946
##          TNR         FNR Precision  Threshold
## 1  0.5198975 0.002772052 0.2256824 0.11316378
## 2  0.5894239 0.021821025 0.2505462 0.14352241
## 3  0.6130416 0.031345511 0.2599474 0.09316676
## 4  0.6161234 0.032980311 0.2611625 0.09565707
## 5  0.6143381 0.032056294 0.2604521 0.09934074
## 6  0.6797754 0.097306134 0.2834379 0.14492984
## 7  0.6809024 0.097235056 0.2841705 0.14468299
## 8  0.6829869 0.099793873 0.2849269 0.15068889
## 9  0.6396613 0.056365058 0.2687177 0.12439630
## 10 0.6419054 0.058497406 0.2695015 0.12578227
```

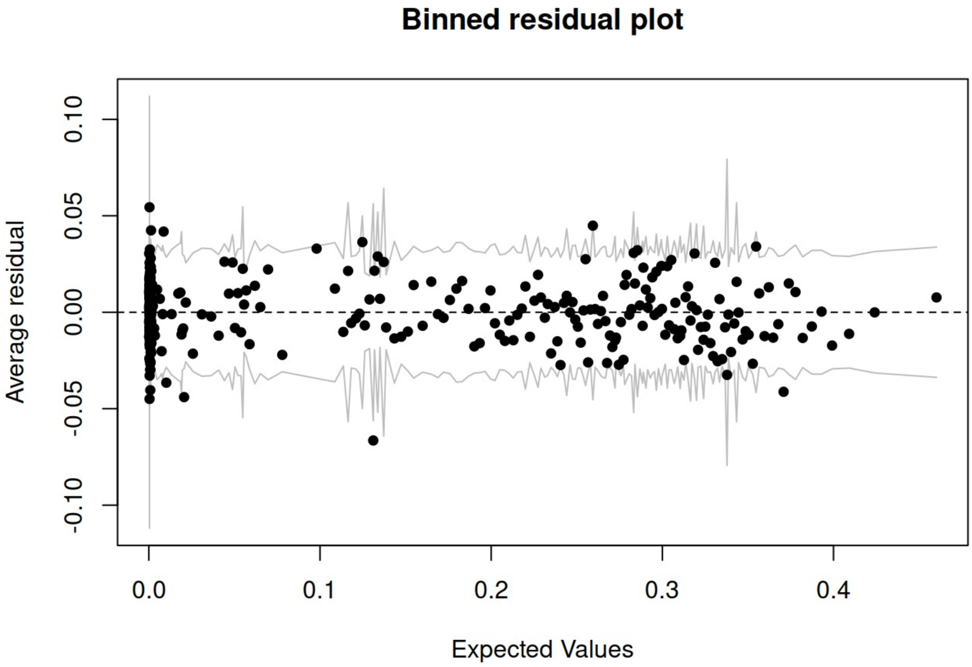
This analysis shows that the logarithm transformation doesn't produce a better AIC, actually it performes worse than the model without logarithm. Hence, we decided not to include the logarithm transformation.

```
vif(result_glm$Models$Driving_License)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## Previously_Insured 1.075696  1        1.037158
## Vehicle_Damage     1.080869  1        1.039649
## Channels_Reduced   2.190914  4        1.103007
## Age                1.774232  1        1.332003
## Vehicle_Age        2.701973  2        1.282095
## Region_Reduced     1.090994  4        1.010946
## Driving_License    1.002784  1        1.001391
```

Through the vif function we can see that there is no sign of multicolinearity, since the variance inflation factors are very small for every variable.

```
# Get predicted values from the model
predicted_values <- predict(result_glm$Models$Region_Reduced, unbalanced_test, type = "response")
# Calculate residuals
residuals <- residuals(result_glm$Models$Region_Reduced, type = "response")

# Plot the binned residuals
arm::binnedplot(predicted_values, residuals)
```



The binned residuals shown are contained in the confidence interval and they are evenly concentrated around zero. There seem to be a slight hint of heteroscedasticity.

GAM

In order to fit GAM models we performed the ranking of nested models used for GLMs:

```
result_gam <- ranking_nested_models(unbalanced_train, unbalanced_test, use_model = "gam", use_log = FALSE, use_sp
lines = TRUE)
```

```
## [1] "Response ~ Previously_Insured"
## [1] "Response ~ Previously_Insured + Vehicle_Damage"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + s(Age)"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + s(Age) + Channels_Reduced"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + s(Age) + Channels_Reduced + Region_Reduced"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + s(Age) + Channels_Reduced + Region_Reduced + Vehicle_Age
"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + s(Age) + Channels_Reduced + Region_Reduced + Vehicle_Age
+ s(Annual_Premium)"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + s(Age) + Channels_Reduced + Region_Reduced + Vehicle_Age
+ s(Annual_Premium) + Driving_License"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + s(Age) + Channels_Reduced + Region_Reduced + Vehicle_Age
+ s(Annual_Premium) + Driving_License + Gender"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + s(Age) + Channels_Reduced + Region_Reduced + Vehicle_Age
+ s(Annual_Premium) + Driving_License + Gender + Vintage"
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
result_gam$Results
```

```
##            Model_Name      AIC       AUC  Accuracy       TPR       FPR
## 1  Previously_Insured 155905.7 0.7585627 0.5786343 0.9972279 0.4801025
## 2      Vehicle_Damage 150595.1 0.7890232 0.6372613 0.9781790 0.4105761
## 3                 Age 144890.1 0.8417453 0.6732614 0.9481129 0.3653056
## 4    Channels_Reduced 143278.1 0.8501295 0.6979700 0.9226669 0.3335594
## 5      Region_Reduced 143075.8 0.8510111 0.6959058 0.9281399 0.3366812
## 6         Vehicle_Age 142992.3 0.8515785 0.7005327 0.9223115 0.3305872
## 7      Annual_Premium 142932.1 0.8521367 0.6972178 0.9274291 0.3350854
## 8     Driving_License 142901.6 0.8522854 0.6881215 0.9404364 0.3472832
## 9              Gender 142874.9 0.8522166 0.6935268 0.9337551 0.3401819
## 10            Vintage 142876.7 0.8522134 0.6942615 0.9326889 0.3391945
##          TNR         FNR Precision Threshold
## 1  0.5198975 0.002772052 0.2256824 0.1131638
## 2  0.5894239 0.021821025 0.2505462 0.1435224
## 3  0.6346944 0.051887128 0.2669615 0.1045971
## 4  0.6664406 0.077333144 0.2796123 0.1229755
## 5  0.6633188 0.071860118 0.2789277 0.1207088
## 6  0.6694128 0.077688535 0.2813408 0.1290902
## 7  0.6649146 0.072570901 0.2797299 0.1225083
## 8  0.6527168 0.059563580 0.2753533 0.1140271
## 9  0.6598181 0.066244936 0.2780612 0.1189352
## 10 0.6608055 0.067311110 0.2784155 0.1193174
```

Fitting a GAM model with a spline for the variable Age improves the AIC of the models. Therefore, the comparison of the AIC of the nested GAM models confirms the results obtained with GLM: that is, the model that strikes the best balance between AIC value and number of variables is the one that contains the variables up to Region_Reduced; adding other variables doesn't significantly reduce the AIC. It's worth to notice that the order of the variables changes: Age results to be slightly more significant adding the spline.

```
summary(result_gam$Models$Region_Reduced)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## Response ~ Previously_Insured + Vehicle_Damage + s(Age) + Channels_Reduced +
##     Region_Reduced
##
## Parametric coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -3.05287    0.04875 -62.625  < 2e-16 ***
## Previously_InsuredYes -3.85032    0.09558 -40.282  < 2e-16 ***
## Vehicle_DamageYes      2.01116    0.04089  49.189  < 2e-16 ***
## Channels_Reduced124   -0.19166    0.01699 -11.280  < 2e-16 ***
## Channels_Reduced152   -1.00391    0.03076 -32.639  < 2e-16 ***
## Channels_Reduced160   -1.65849    0.06397 -25.924  < 2e-16 ***
## Channels_Reduced0     -0.26664    0.01760 -15.147  < 2e-16 ***
## Region_Reduced28       0.26494    0.02651   9.994  < 2e-16 ***
## Region_Reduced41       0.40953    0.03952  10.362  < 2e-16 ***
## Region_Reduced46       0.12364    0.03963   3.120  0.00181 **
## Region_Reduced0        0.11996    0.02640   4.545 5.51e-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) 8.323  8.754   2803  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.184   Deviance explained = 27.9%
## UBRE = -0.46369  Scale est. = 1         n = 266776
```

Since the expected degrees of freedom of the variable Age is 8.34, we can consider the splines relevant for the variable.

From the summary we can see that all the variables are statistically significant, hence we may interpret their meaning by analyzing the exponential values:

```
exp(result_gam$Models$Region_Reduced$coefficients)
```

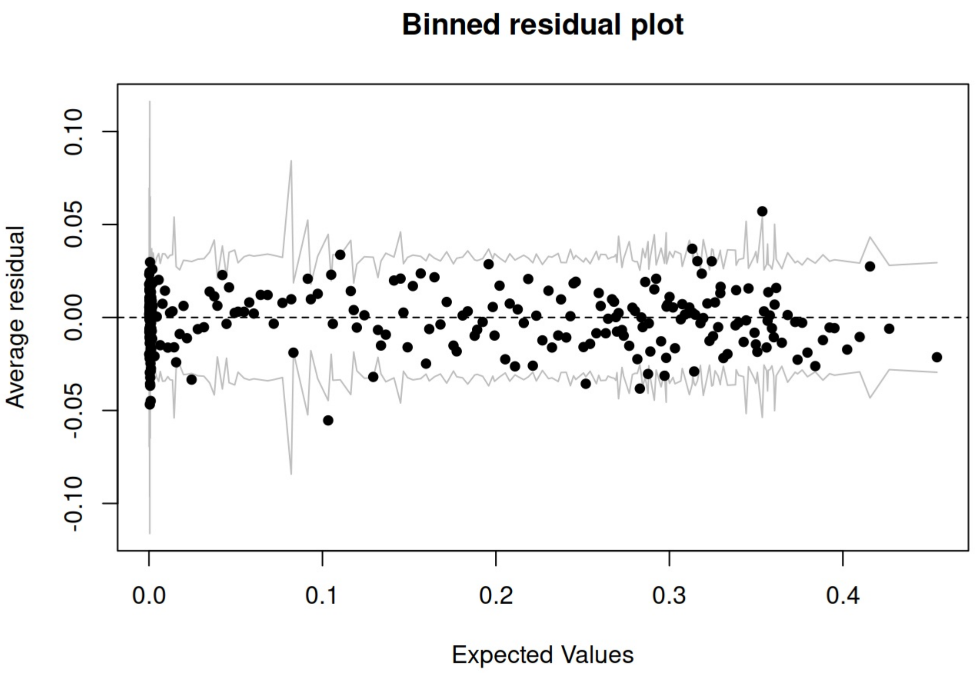
```
##           (Intercept) Previously_InsuredYes     Vehicle_DamageYes
##            0.04722329            0.02127285            7.47196369
##   Channels_Reduced124   Channels_Reduced152   Channels_Reduced160
##            0.82558527            0.36644516            0.19042592
##     Channels_Reduced0      Region_Reduced28      Region_Reduced41
##            0.76594544            1.30335084            1.50611374
##      Region_Reduced46       Region_Reduced0              s(Age).1
##            1.13160731            1.12745438            0.15714402
##              s(Age).2              s(Age).3              s(Age).4
##           22.42326158            5.18741501            0.13795396
##              s(Age).5              s(Age).6              s(Age).7
##            0.27268376            3.71529027            2.37795905
##              s(Age).8              s(Age).9
##           75.11888983            1.17098396
```

we can comment these results as follows:

- The odds of a customer being interested in Vehicle insurance are 7 times higher for those who have had a vehicle damage compared to those who didn't.
- Compared to region 8, the odds of a customer being interested in Vehicle insurance are higher for those who live in Region 28, 41, 46 or 0.
- The odds of a customer being interested in Vehicle insurance get higher as the age of the vehicle increases.
- Compared to channel 26, the odds of a customer being interested in vehicle insurance are lower for those having a channel of outreaching with code 124, 160, 152 or 0.
- The odds of a customer being interested in vehicle insurance get lower for:
    - older customers;
    - customers that have been previously insured.

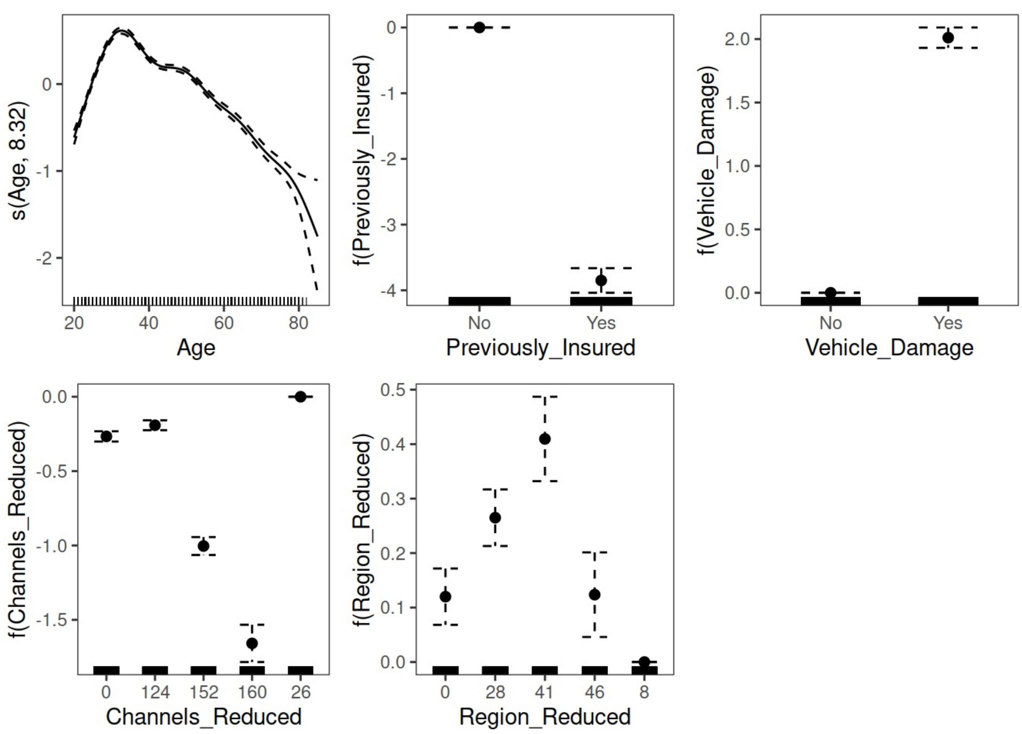
Trying to fit the model with gam and adding a spline for Age variable, we noticed that not only the AIC improves, but the expected degrees of freedom for the Age are significantly high, hence, so far, the best model seems to be the one using gam with splines on Age.

```
# Get predicted values from the model
predicted_values <- predict(result_gam$Models$Region_Reduced, unbalanced_test, type = "response")

# Calculate residuals
residuals <- residuals(result_glm$Models$Region_Reduced, type = "response")

# Plot the binned residuals
arm::binnedplot(predicted_values, residuals)
```



The binned residuals shown are contained in the confidence interval and they are evenly concentrated around zero. There seem to be a slight hint of heteroscedasticity.

```
# select the commands that actually convey relevant information
gam_sampledViz <- getViz(result_gam$Models$Region_Reduced)
print(plot(gam_sampledViz, allTerms = T), pages = 1)
```



This plot shows the log-odds coefficients of the GAM model for the categorical variables and the spline of Age.

Random Forest

Due to our limited computational resources, we had to sample our dataset in order to fit a random forest model with 500 trees.

```
sample <- train_reduced[sample(nrow(train_reduced), nrow(train_reduced)*0.6, replace = FALSE),]
```

```
trainIndex <- createDataPartition(sample$Response, p = .8, list = FALSE, times = 1)

trainSet <- sample[trainIndex,]
testSet <- sample[-trainIndex,]
```

```
# Drop columns Policy_Sales_Channel and Region_Code from unbalanced_train
trainSet <- trainSet[, !names(trainSet) %in% c("Policy_Sales_Channel", "Region_Code")]

# Drop columns Policy_Sales_Channel and Region_Code from unbalanced_test
testSet <- testSet[, !names(testSet) %in% c("Policy_Sales_Channel", "Region_Code")]
```

```
model_rf <- randomForest(Response ~ ., data = trainSet, importance = TRUE, ntree = 500)
print(model_rf)
```

```
## 
## Call:
##  randomForest(formula = Response ~ ., data = trainSet, importance = TRUE,      ntree = 500) 
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
## 
##         OOB estimate of  error rate: 12.4%
## Confusion matrix:
##         No Yes class.error
## No  159874 522 0.003254445
## Yes  22164 372 0.983493078
```

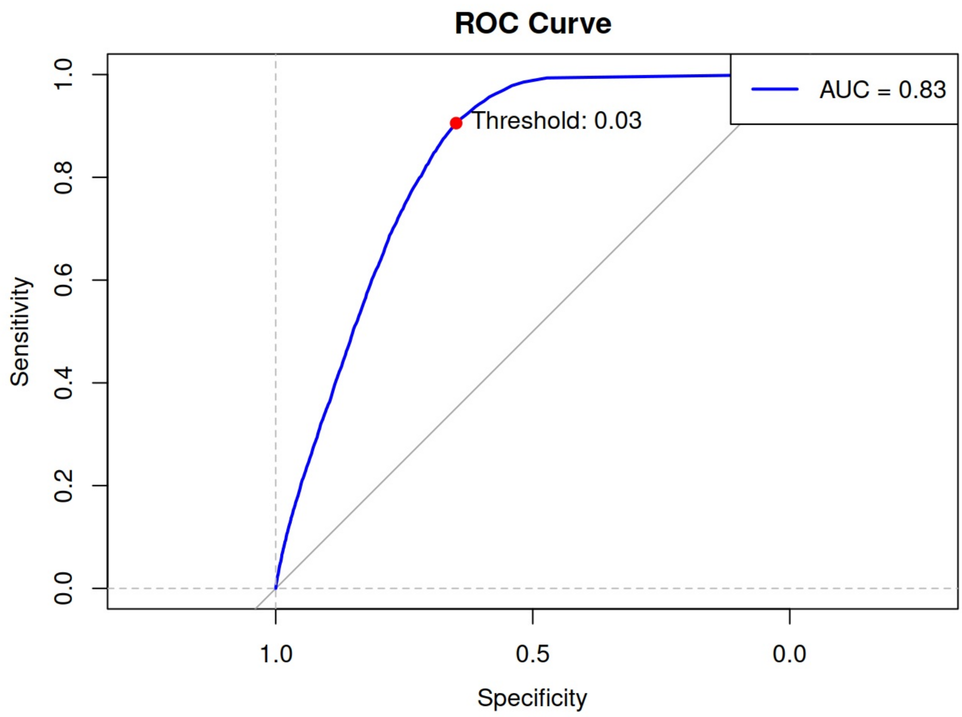
We can notice that the class error for the Yes category is very high, therefore we decided to use a threshold in order to avoid misclassification.

```
# Predict probabilities
probabilities <- predict(model_rf, newdata = subset(testSet, select = -Response), type = "prob")[, "Yes"]

# Compute ROC curve
roc_curve <- roc(testSet$Response, probabilities)
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
# Calculate AUC
auc_score <- auc(roc_curve)

# Find optimal threshold using Youden's J statistic
youdens_j <- coords(roc_curve, "best", best.method = "youden")
optimal_threshold <- youdens_j$threshold

# Plot the ROC curve using plot.roc from the pROC package
plot.roc(roc_curve, col = "blue", main = "ROC Curve", lwd = 2)

# Add a point for the best threshold
points(youdens_j$specificity, youdens_j$sensitivity, pch = 19, col = "red")

# Adding a legend or text to mark the point
text(youdens_j$specificity, youdens_j$sensitivity, labels = paste("Threshold:", round(optimal_threshold, 2)), pos
= 4)

# Add labels and legend
abline(h = 0, v = 1, lty = 2, col = "gray")
legend("topright", legend = paste("AUC =", round(auc(roc_curve), 2)), col = "blue", lwd = 2)
```



The model has an AUC of over 80% and the threshold which maximizes the difference between TPR and FPR is around 0.01; this small value could be justified by the imbalance of the response variable.

```r
#*FUNCTION TO PERFORM THE MODEL ASSESSMENT -------------------------------------
random_forest_assessment <- function(model_rf, testSet){
  # Probabilities prediction of the positive class
  probabilities <- predict(model_rf, newdata = subset(testSet, select = -Response), type = "prob")[, "Yes"]

  # Compute ROC curve
  roc_curve <- roc(testSet$Response, probabilities)

  # Calculate AUC
  auc_score <- auc(roc_curve)

  # Find optimal threshold using Youden's J statistic
  youdens_j <- coords(roc_curve, "best", best.method = "youden")
  optimal_threshold <- youdens_j$threshold

  # Save ROC curve plot if specified

  # Obtain predicted classes based on the optimal threshold
  predicted_classes <- ifelse(probabilities > optimal_threshold, "Yes", "No")

  # Create the confusion matrix
  conf_matrix <- table(Actual = testSet$Response, Predicted = predicted_classes)

  conf_matrix_prop <- prop.table(conf_matrix, margin = 1)

  # Calculate accuracy
  accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)

  # Calculate true positive rate
  tpr <- conf_matrix[2, 2] / sum(conf_matrix[2, ])

  # Calculate false positive rate
  fpr <- conf_matrix[1, 2] / sum(conf_matrix[1, ])

  # Calculate true negative rate
  tnr <- conf_matrix[1, 1] / sum(conf_matrix[1, ])

  # Calculate false negative rate
  fnr <- conf_matrix[2, 1] / sum(conf_matrix[2, ])

  # Calculate precision
  precision <- conf_matrix[2, 2] / sum(conf_matrix[, 2])

  # Store the results in a data frame
  results_df <- data.frame(AUC = auc_score,
                           Accuracy = accuracy,
                           TPR = tpr,
                           FPR = fpr,
                           TNR = tnr,
                           FNR = fnr,
                           Precision = precision,
                           Threshold = optimal_threshold)
  # Return the results
  return(results_df)

}

#*TRAIN THE RANDOM FOREST MODEL ------------------------------------------------

# Set the seed for reproducibility


# Train the random forest model
#rf_model <- randomForest(Response ~ ., data = unbalanced_train, ntree = 100)

# Assess the model using the test set and don't save the plots
rf_assessment <- random_forest_assessment(model_rf, testSet)
```

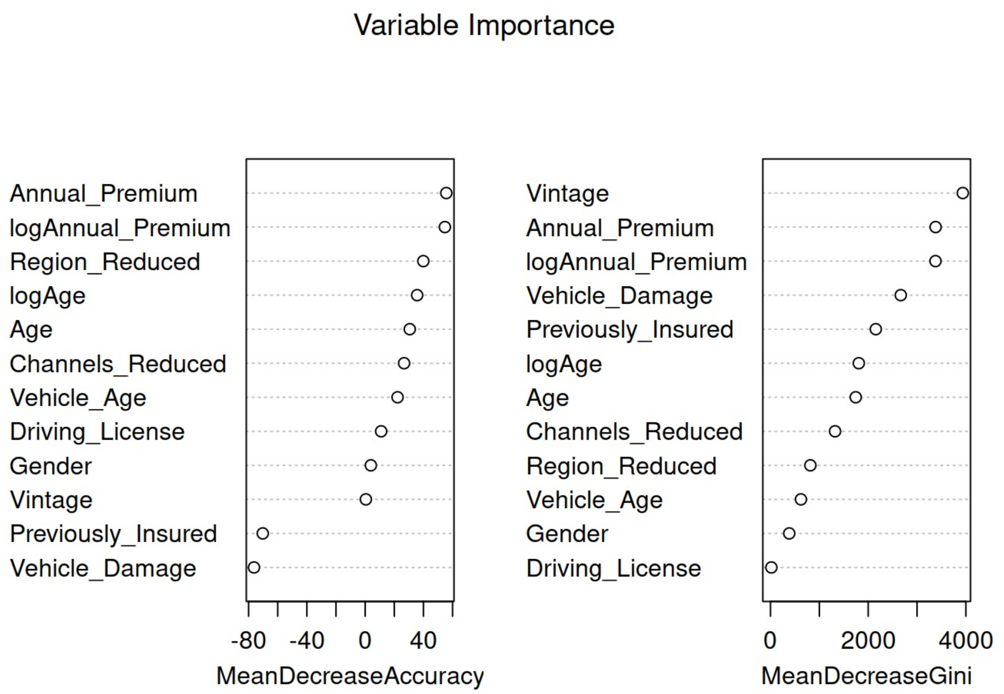
```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```r
rf_assessment
```

```
##         AUC  Accuracy       TPR       FPR       TNR        FNR Precision
## 1 0.8284575 0.6806245 0.9053958 0.3509564 0.6490436 0.09460419 0.2660373
##   Threshold
## 1     0.029
```

```
varImpPlot(model_rf, sort = T, main = "Variable Importance")
```



From this plot we can notice that the variable Vintage is taken into account as one of the most relevant with respect to Mean Decrease Gini. Apparently Vintage helps the model in finding pure nodes.

Performances and conclusion

Our goal was to determine the relationship between the response variable and the predictor. We selected the best models for each type of technique and computed the performance indexes which are illustrated in the table.

Models/Indexes	AUC	Accuracy	TPR	FPR	TNR	FNR	Precision
GLM	0.84	**0.71**	0.91	**0.32**	**0.68**	0.09	**0.28**
GAM	**0.85**	0.70	**0.93**	0.34	0.66	**0.07**	**0.28**
Random Forest	0.82	0.67	0.92	0.37	0.63	0.08	0.26

GLM: best model for Accuracy, FPR, TNR and Precision;

GAM: best for AUC, TPR, FNR and Precision;

Random Forest: doesn't perform better than the other kind of models under any index, but has similar performances.

After analyzing the upper table and taking into account all the previous considerations, we can conclude that the model that best explains this relationship is the GAM model. The relationship is well explained using the variables Previously_Insurance, Vehicle_Damage, Age, Channels_Reduced and Region_Code; other variable could be taken into account but they would over complicate the model while not affecting significantly the performances. Also we saw that it is significant to consider splines for the age variable.

The other models obtain good results in respect to the GAM, with GLM being the closest one and also the lightest to train.

It is worth noting that in the context of insurance, it is sensible to assume that the cost of the FNs is is higher than the cost of FPs, because the unrealized revenues of losing a potential client are greater than the cost of contacting a non interested customer. Therefore, given that the GAM model exhibits the highest AUC and the lowest FNR, this could provide an additional reason for selecting it over the other models.

Finally, it's worth mentioning that we also tried to balance the dataset, but the results were not significantly different from the unbalanced one. This is what we expected, considering that the imbalance ratio (IR) of the response variable is not too high.