Machine Learning in R

Serena Huang

Executive Summary

Our analysis identifies key factors influencing call conversion likelihood. By leveraging predictive analytics and averaging the conversion likelihood from logistic regression and random forest models, we identified key trends. Specifically, individuals aged 60-69, households earning between 100K-200K annually, and smaller households (size 2) demonstrate the highest likelihood of conversion. In contrast, we found that call attributes had a mixed impact on conversion likelihood compared to caller characteristics.

Overview

The objective of this project is to estimate the conversion likelihood for each call in the dataset and identify the call types most likely to convert. To accomplish this, we will conduct exploratory data analysis and preprocess the data accordingly. Subsequently, we will develop and train various machine learning classification models, including logistic regression, decision trees, and random forests. We will then evaluate these models based on metrics such as confusion matrix, ROC curve, and AUC to select the most effective model. Additionally, we will visualize our findings using multiple box plots to illustrate the relationship between conversion likelihood and call and caller attributes.

Data Setup

```
# Set up libraries and working directory
rm(list=ls())
library(tidyverse)
library(caret)
library(ggpubr)
library(ggplot2)
library(fastDummies)
library(glmnet)
library(randomForest)
library(pROC)
library(ROCR)
library(tree)
library(rpart)
library(rpart.plot)
setwd("C:/Users/seren/OneDrive/Desktop/Machine_Learning")

# Read in datasets
health_calls_2023 <- read.csv("health_calls_2023.csv")
health_calls_2024 <- read.csv("health_calls_2024.csv")
source_data <- read.csv("source.csv")
user_data <- read.csv("user_provided_data.csv")
```

```r
# Merge datasets
calls <- health_calls_2023 %>%
  mutate(season="Q4_2023") %>%
  bind_rows(health_calls_2024) %>%
  mutate(season = ifelse(is.na(season), "Q1_2024", season)) %>%
  left_join(source_data, by = "source_id") %>%
  left_join(user_data, by = "call_id")
```

Data Exploration

After taking a quick look at the summary data, we can see the quantile percentages for each variable. Besides this, we also notice a lot of missing data, which will be addressed in the 'Data Processing' section.

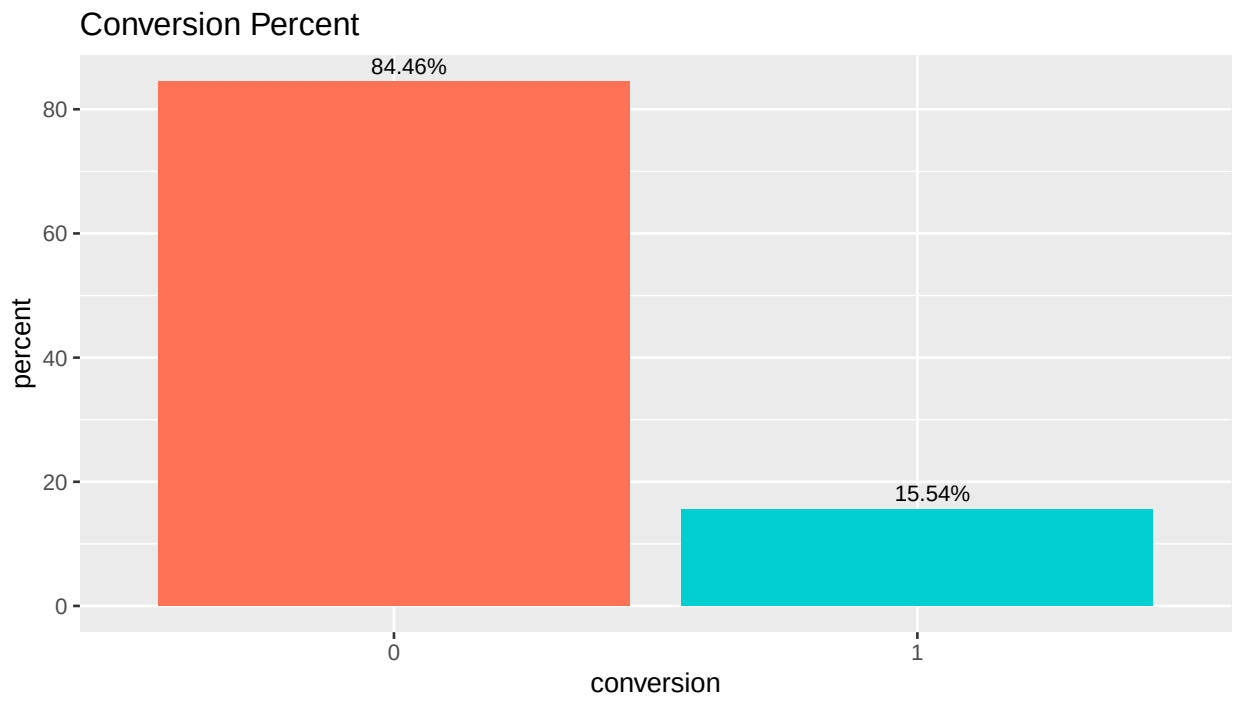
```r
# Simple summary of dataset
summary(calls)
```

```
##    call_id          phone_num_hash        buyer_id        source_id
##  Length:99546       Length:99546       Min.   :   171   Min.   :    283
##  Class :character   Class :character   1st Qu.:  2282   1st Qu.:   3329
##  Mode  :character   Mode  :character   Median : 34844   Median :   3330
##                                        Mean   : 38549   Mean   : 226806
##                                        3rd Qu.: 63253   3rd Qu.:   5531
##                                        Max.   :106603   Max.   :2410066
##
##      time           call_duration       conversion       season
##  Length:99546       Min.   :    0.0   Min.   :0.00    Length:99546
##  Class :character   1st Qu.:   77.0   1st Qu.:0.00    Class :character
##  Mode  :character   Median :  327.0   Median :0.00    Mode  :character
##                     Mean   :  679.1   Mean   :0.16
##                     3rd Qu.:  960.0   3rd Qu.:0.00
##                     Max.   :10666.0   Max.   :1.00
##                                       NA's   :85927
##    seller_id      seller_type          state           currently_insured
##  Min.   :   1.0   Length:99546       Length:99546       Min.   :0.00
##  1st Qu.: 506.0   Class :character   Class :character   1st Qu.:0.00
##  Median : 507.0   Mode  :character   Mode  :character   Median :0.00
##  Mean   : 876.5                                        Mean   :0.01
##  3rd Qu.: 782.0                                        3rd Qu.:0.00
##  Max.   :4127.0                                        Max.   :1.00
##                                                        NA's   :98730
##  connection_type      browser          browser_platform      device
##  Length:99546       Length:99546       Length:99546       Length:99546
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##       age              bmi         coverage_type      credit_rating
##  Min.   :11.40   Min.   : 8.80   Length:99546       Mode:logical
##  1st Qu.:28.70   1st Qu.:24.30   Class :character   NA's:99546
##  Median :40.60   Median :28.20   Mode  :character
```

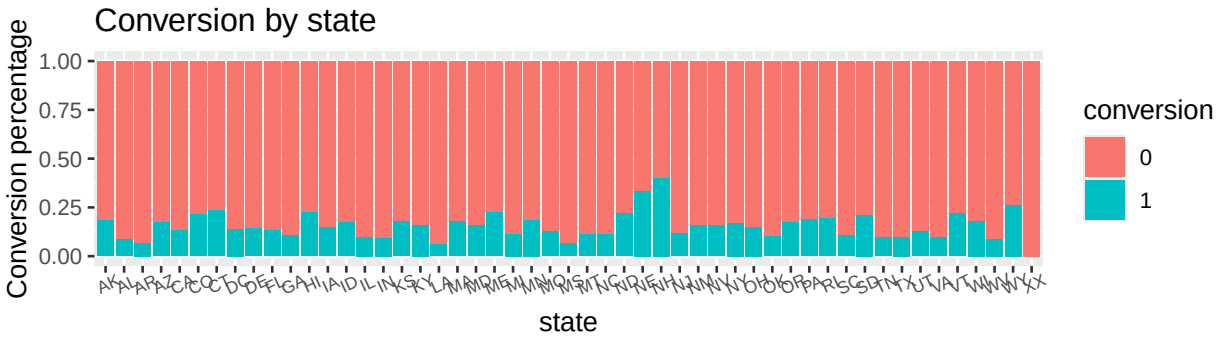
```
##  Mean   :41.43   Mean   :29.89
##  3rd Qu.:54.20   3rd Qu.:34.02
##  Max.   :89.50   Max.   :99.90
##  NA's   :813     NA's   :99232
##  current_company       gender              height      household_income
##  Length:99546       Length:99546       Min.   : 5.00   Min.   :0.000e+00
##  Class :character   Class :character   1st Qu.:64.00   1st Qu.:2.050e+04
##  Mode  :character   Mode  :character   Median :66.00   Median :3.700e+04
##                                        Mean   :66.57   Mean   :8.308e+04
##                                        3rd Qu.:70.00   3rd Qu.:4.000e+04
##                                        Max.   :77.00   Max.   :4.295e+09
##                                        NA's   :99232   NA's   :577
##  married        household_size    subsidy              weight
##  Mode:logical   Min.   :1.000   Length:99546       Min.   : 48.0
##  NA's:99546     1st Qu.:1.000   Class :character   1st Qu.:150.0
##                 Median :1.000   Mode  :character   Median :185.0
##                 Mean   :1.889                      Mean   :188.6
##                 3rd Qu.:2.000                      3rd Qu.:215.0
##                 Max.   :7.000                      Max.   :379.0
##                                                    NA's   :99232
```

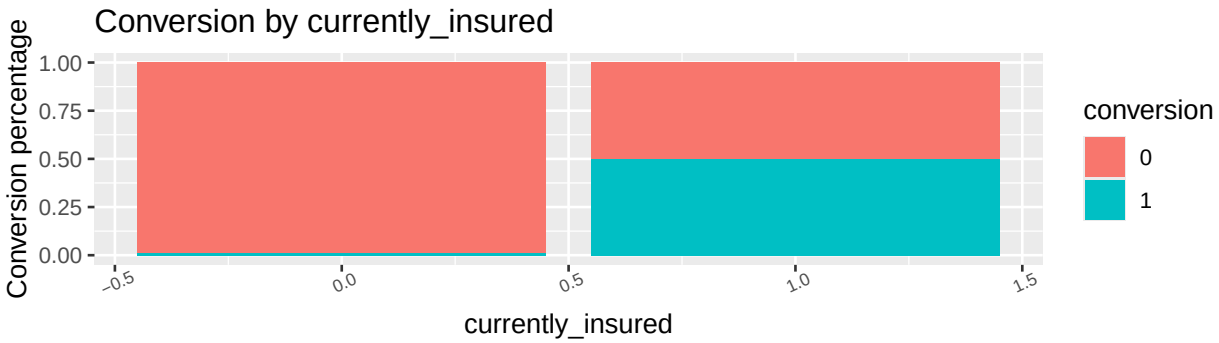
Despite the presence of 85,927 missing values out of 99,546 total records for the conversion variable, our exploratory data analysis focuses on the remaining 13,619 rows where conversion data is available. We aim to explore how conversion is distributed within this subset and its variations across different features.

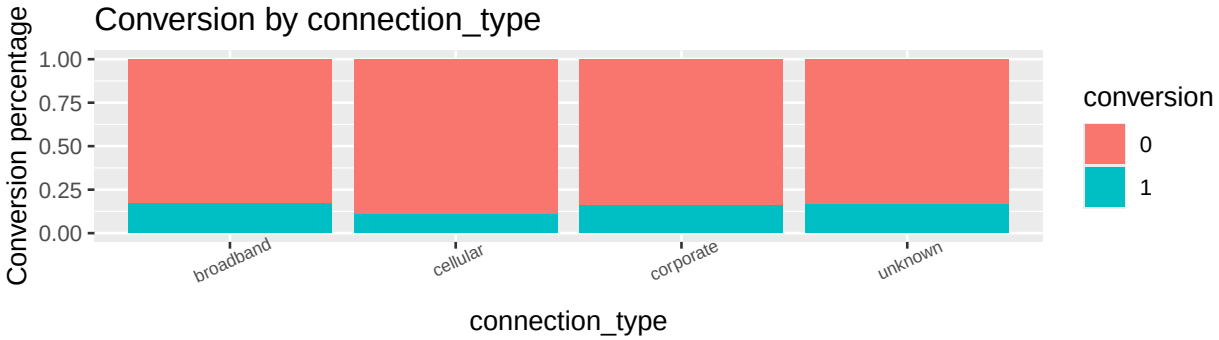
We will draw stacked bar plots for categorical features and box plots for numeric features.

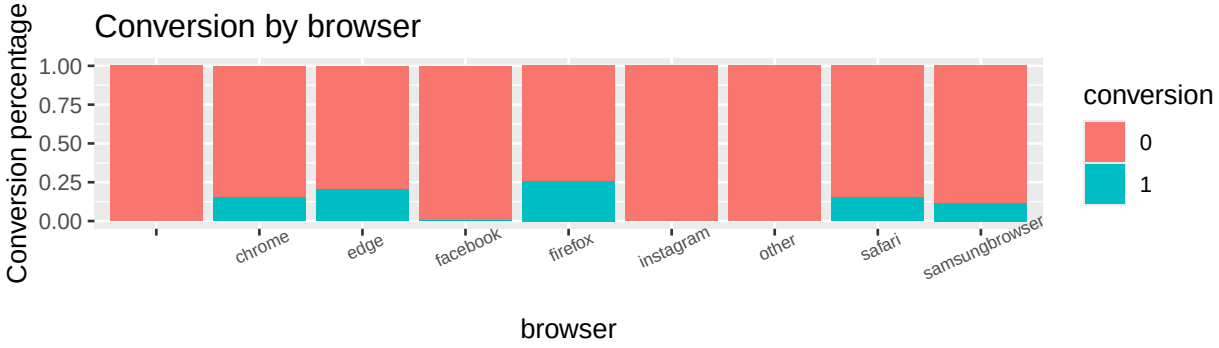



```
## $`1`
```

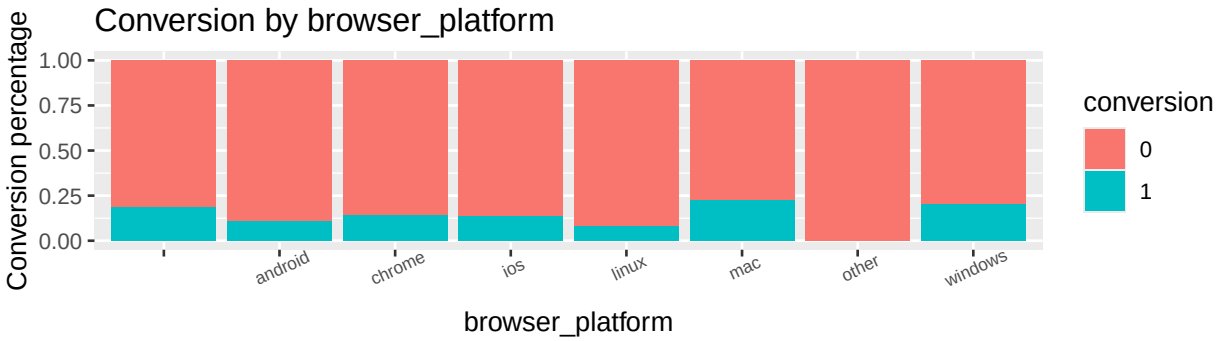

Conversion by state
Conversion percentage
1.00
0.75
0.50
0.25
0.00
AK AL AR AZ CA CO CT DC DE FL GA HI IA ID IL IN KS KY LA MA MD ME MI MN MO MS MT NC ND NE NH NJ NM NV NY OH OK OR PA RI SC SD TN TX UT VA VT WA WI WV WY
state
conversion
0
1

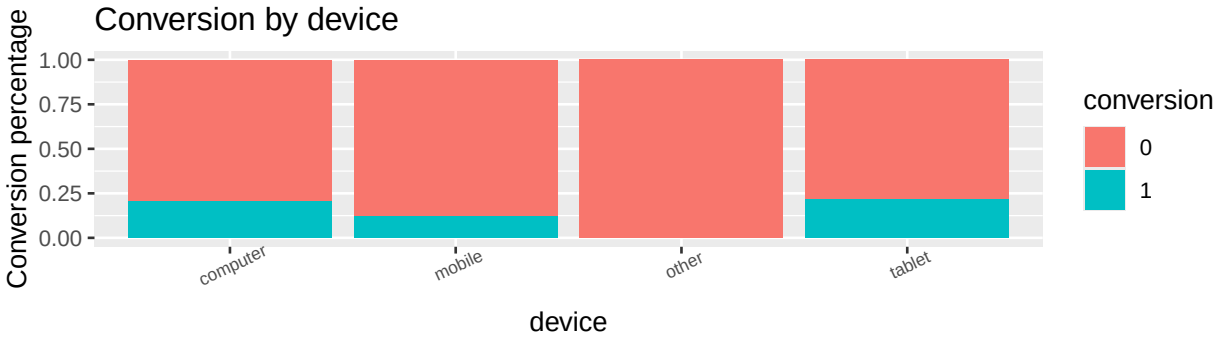

Conversion by currently_insured
Conversion percentage
1.00
0.75
0.50
0.25
0.00
−0.5
0.0
0.5
1.0
1.5
currently_insured
conversion
0
1

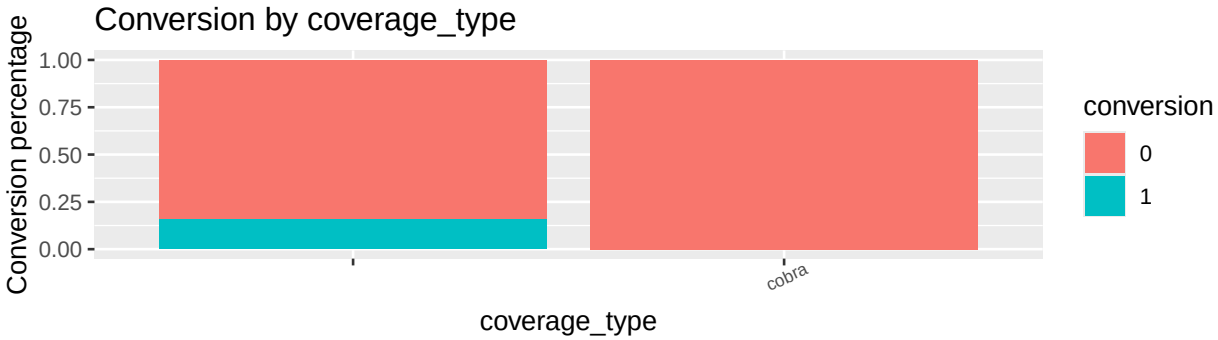

Conversion by connection_type
Conversion percentage
1.00
0.75
0.50
0.25
0.00
broadband
cellular
corporate
unknown
connection_type
conversion
0
1

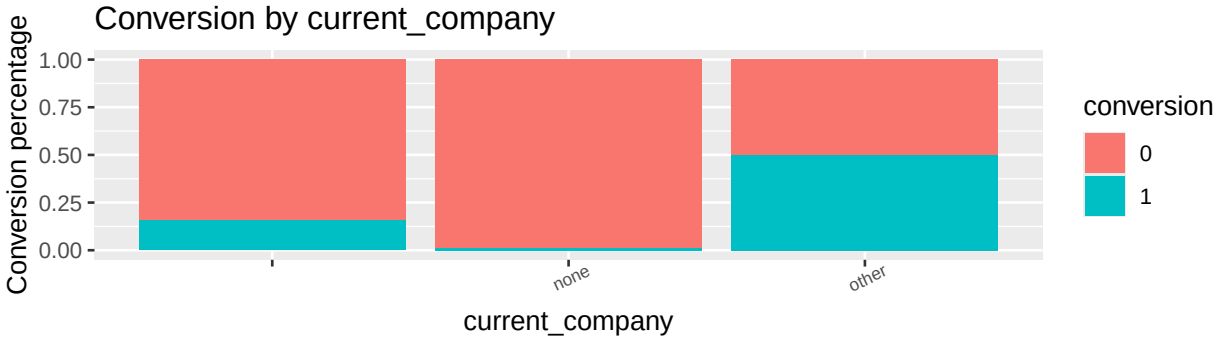

Conversion by browser
Conversion percentage
1.00
0.75
0.50
0.25
0.00
chrome
edge
facebook
firefox
instagram
other
safari
samsungbrowser
browser
conversion
0
1

```
##
## $`2`
```

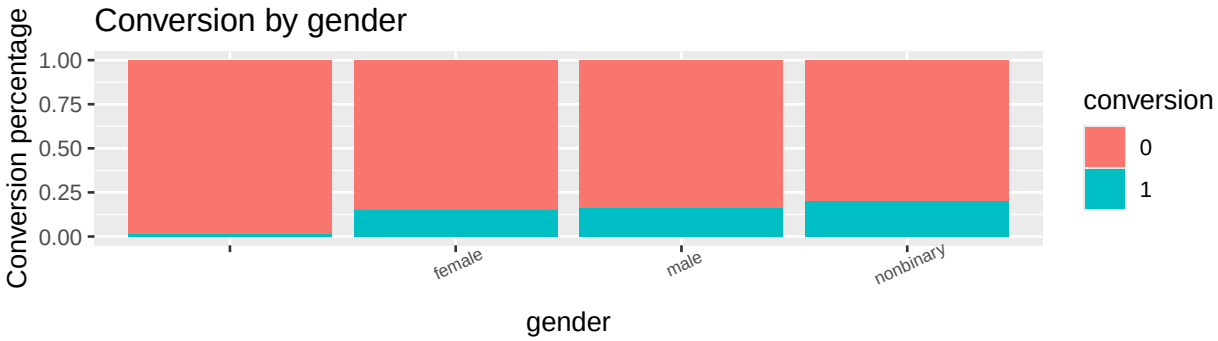

Conversion by browser_platform
Conversion percentage
1.00
0.75
0.50
0.25
0.00
android
chrome
ios
linux
mac
other
windows
browser_platform
conversion
0
1

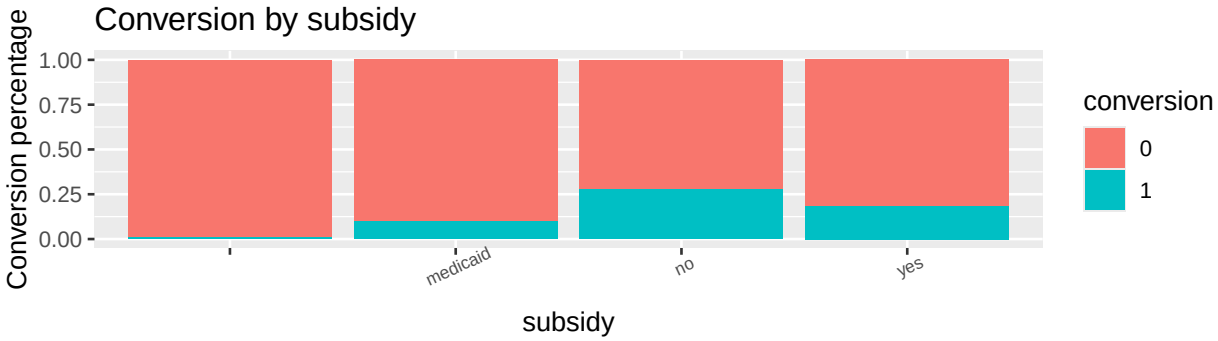

Conversion by device
Conversion percentage
1.00
0.75
0.50
0.25
0.00
computer
mobile
other
tablet
device
conversion
0
1

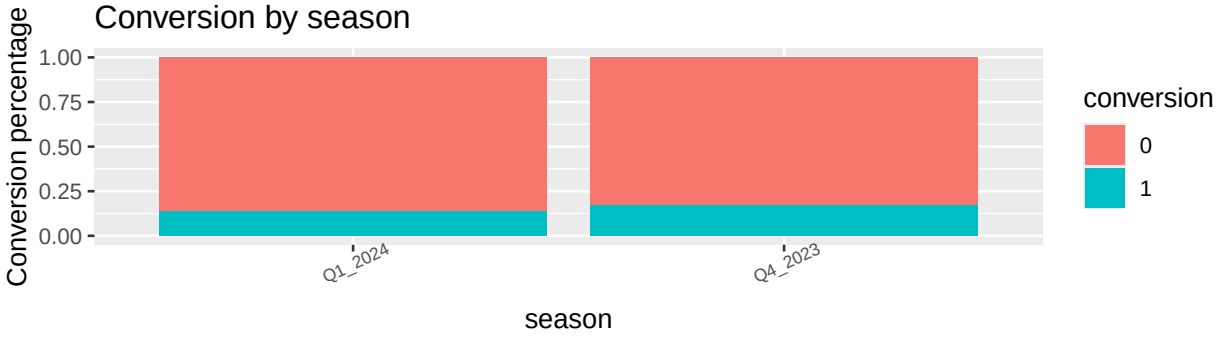

Conversion by coverage_type
Conversion percentage
1.00
0.75
0.50
0.25
0.00
cobra
coverage_type
conversion
0
1


Conversion by current_company
Conversion percentage
1.00
0.75
0.50
0.25
0.00
none
other
current_company
conversion
0
1

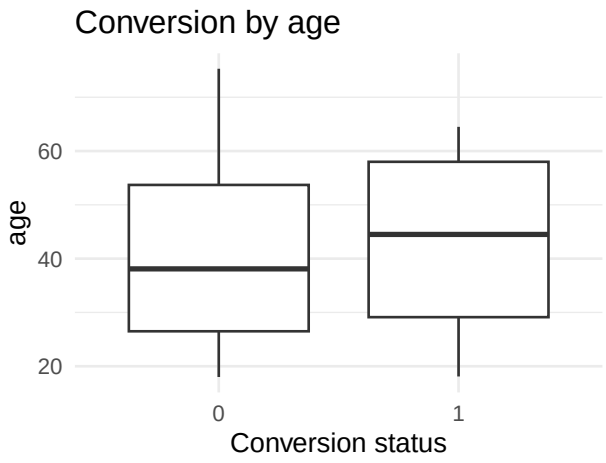
```
##
## $`3`
```

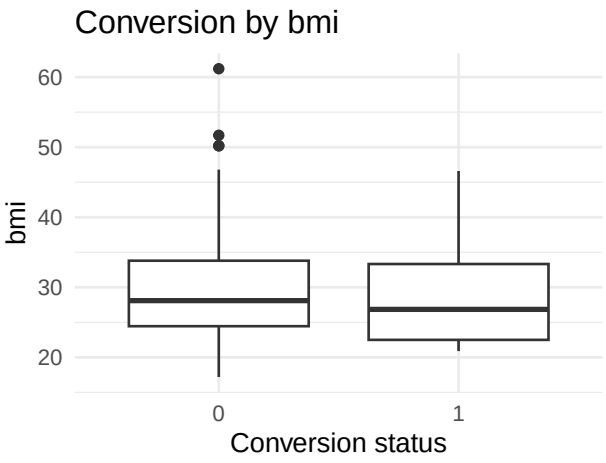


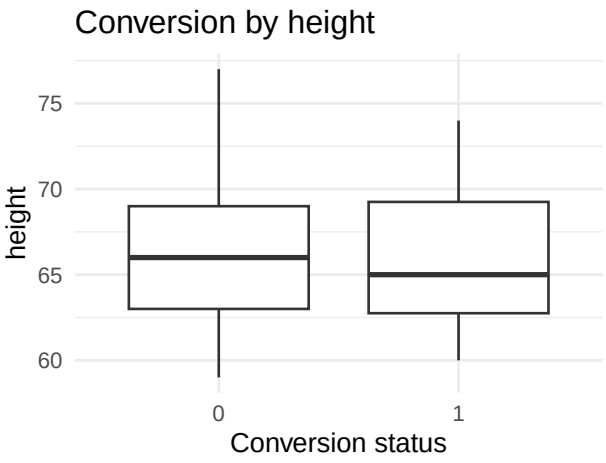


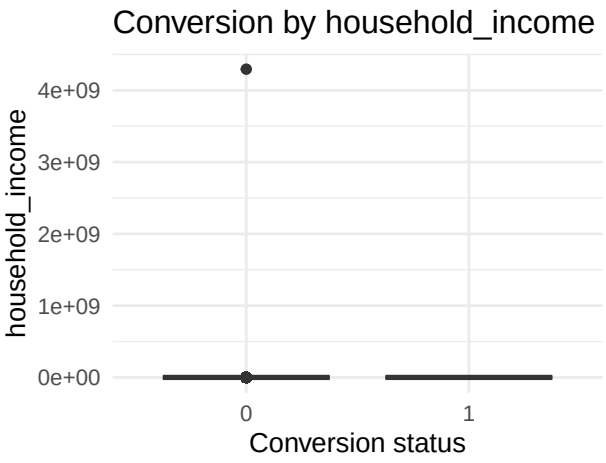


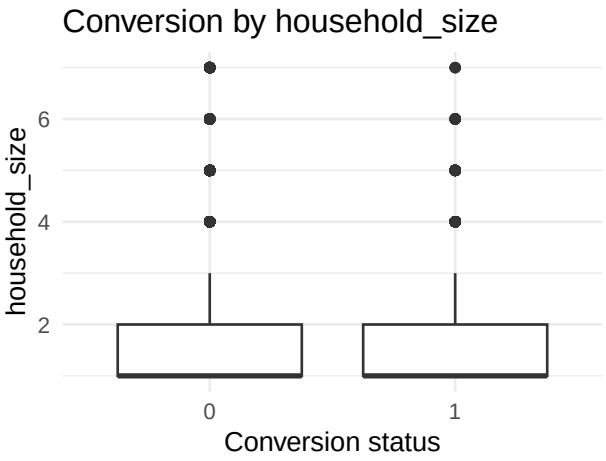

```
##
## attr(,"class")
## [1] "list"      "ggarrange"
```

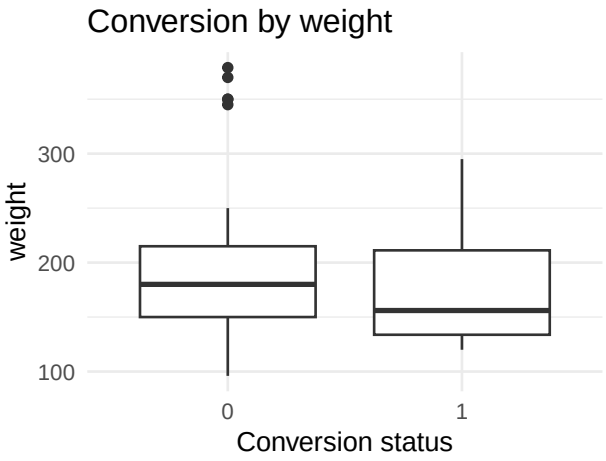













```
## # A tibble: 51 x 3
##    state convert_pct no_convert_pct
##    <chr>       <dbl>          <dbl>
##  1 AK          18.2            81.8
##  2 AL           8.51           91.5
##  3 AR           6.82           93.2
##  4 AZ          17.2            82.8
##  5 CA          13.0            87.0
```

```
##  6 CO          21.4            78.6
##  7 CT          23.5            76.5
##  8 DC          14              86
##  9 DE          14.3            85.7
## 10 FL          13.1            86.9
## # i 41 more rows
```

Findings:

1) The dataset contains a significantly larger number of non-converted callers compared to converted callers.

2) Conversion rates do indeed vary across different features. For instance, the browser analysis reveals that Firefox exhibits a higher conversion rate compared to other browsers.

3) Outliers are observed in columns such as household_income, suggesting potential anomalies in the data.

These findings highlight the need for careful data preparation and modeling, which will be done in the following sections.

Data Processing

As we mentioned above, there are a few issues with the data.

Missing Data

Variables with missing values can be categorized into four groups, to address these issues, we have a solution for each category.

1) Variables with constant/missing values throughout the entire dataset (e.g., credit_rating, married, seller_type)

   Solution: Drop these columns as they do not provide any additional information

2) Variables with more than 85% missing data (e.g., currently_insured, BMI, height, weight).

   Solution: Drop all these columns in the training and testing data

3) Variables with less than 1% missing data (e.g., age, household_income)

   Solution: Replace these missing values with the mean

4) Variables with blanks instead of NAs (e.g., browser, browser_platform, coverage_type, company, gender, subsidy)

   Solution: Replace them with NAs

Without addressing these, models may fail to converge, produce biased predictions, or lead to unstable results.

Outliers

Variables such as bmi, weight, height, household_income have outlier variables. We use the IQR method to establish lower and upper bounds to cap extreme values in bmi, weight, height. We then use log transformation to mitigate skewness in household income data. Last but no least, we standardize all numeric variable to ensure they are on the same scale.These practices will help mitigate bias in results and facilitate meaningful comparisons between variables

Categorical Variables

We use the one hot encoding method to transform categorical variables into dummies. This ensures that all categories are appropriately represented and considered during model training and prediction.

Model Imbalance

Given the higher number of non-converted callers compared to converted callers, we use the upsampling method to increase the representation of converted callers in the training dataset. This means randomly duplicating instances of converted callers. We do this because it aims to create a more balanced and representative training dataset, leading to better model performance and generalization.

After we address these issues, we will split the dataset into training (80%) and testing (20%) data.

```
calls_data <- calls

# Convert blanks to NA for all columns
# Adding this here so that it doesn't affect the creation of dummy vars
calls <- calls %>%
  mutate_all(~ ifelse(. == "", NA, .))

# Imputation
calls_data$age <-
  ifelse(is.na(calls_data$age), mean(calls_data$age, na.rm = TRUE), calls_data$age)
calls_data$household_income <-
  ifelse(is.na(calls_data$household_income),
         mean(calls_data$household_income, na.rm = TRUE), calls_data$household_income)

# One hot encoding
calls_all <- dummy_cols(calls_data, select_columns = categorical_columns, remove_first_dummy = TRUE)
calls_data <- calls_all[, !(names(calls_all) %in% categorical_columns)]

calls_data <- calls_data %>%
  select(-c("call_id", "phone_num_hash","buyer_id","source_id","time",
            "call_duration","seller_id","credit_rating","married","seller_type"))

# Log
calls_data$household_income <- log(calls_data$household_income + 1)

# Handling outliers
handle_outliers <- function(column) {
  Q1 <- quantile(column, 0.25, na.rm = TRUE)
  Q3 <- quantile(column, 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1
  lower_bound <- Q1 - 1.5 * IQR
  upper_bound <- Q3 + 1.5 * IQR
  column[column < lower_bound] <- lower_bound
  column[column > upper_bound] <- upper_bound
  return(column)
}

calls_data$weight <- handle_outliers(calls_data$weight)
calls_data$height <- handle_outliers(calls_data$height)
calls_data$bmi <- handle_outliers(calls_data$bmi)

# Standardizing columns
```

```r
calls_data[numeric_columns] <- scale(calls_data[numeric_columns])
```

```r
set.seed(1999)

# Get a dataset where conversion is not NA
calls_train_data <- calls_data %>%
  filter(!is.na(conversion))

# Get rid of missing values
calls_train_data <- calls_train_data %>%
  select(-c("currently_insured_1","currently_insured_NA","weight","bmi","height"))

train_index <- createDataPartition(calls_train_data$conversion, p = 0.8, list = FALSE)
train <- calls_train_data[train_index, ]
test <- calls_train_data[-train_index, ]

# Upsample
train <- upSample(
  x=train[,-1],
  y=train[,1]
)

train <- train %>%
  rename(conversion=Class)
```

Model Building

We implemented three machine learning classification models, logistic regression, classification trees, and random forests, to predict the conversion likelihood. We then use the confusion matrix to quantify classification accuracy, and ROC curve and AUC to visualize and quantify the model's discriminatory power.

Logistic Regression

We implement logistic regression because it's well-suited for predicting binary outcomes. This method provides probabilities of conversion based on various predictor variables and also allows for the interpretation of coefficients, helping to understand how each predictor contributes to the likelihood of conversion. Its simplicity and interpretability make it a suitable choice for this predictive modeling task.

```r
# Logistic regression
# Fit the logistic regression model
glm_model <- glm(conversion~., data = train, family = binomial)

# Summarize the model
summary(glm_model)
```

```
##
## Call:
## glm(formula = conversion ~ ., family = binomial, data = train)
##
## Coefficients: (4 not defined because of singularities)
##                             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)              -1.883285 721.667048  -0.003 0.997918
## age                       0.152786   0.016043   9.524  < 2e-16 ***
## household_income          0.543294   0.046722  11.628  < 2e-16 ***
## household_size           -0.199390   0.024311  -8.202 2.37e-16 ***
## state_AL                 -0.200776   0.313720  -0.640 0.522182
## state_AR                 -1.022615   0.500971  -2.041 0.041224 *
## state_AZ                  0.064458   0.297789   0.216 0.828634
## state_CA                  0.117426   0.301821   0.389 0.697233
## state_CO                  0.483871   0.277469   1.744 0.081181 .
## state_CT                  0.447700   0.283488   1.579 0.114278
## state_DC                  0.045607   0.326241   0.140 0.888822
## state_DE                 -0.918275   0.736190  -1.247 0.212275
## state_FL                  0.089661   0.276680   0.324 0.745892
## state_GA                  0.095200   0.290576   0.328 0.743195
## state_HI                  0.280857   0.352495   0.797 0.425585
## state_IA                  0.392381   0.318772   1.231 0.218355
## state_ID                  0.446810   0.338431   1.320 0.186755
## state_IL                 -0.281715   0.323665  -0.870 0.384088
## state_IN                  0.170589   0.373411   0.457 0.647785
## state_KS                  0.713958   0.345302   2.068 0.038675 *
## state_KY                  0.500031   0.300913   1.662 0.096570 .
## state_LA                 -0.366337   0.420801  -0.871 0.383989
## state_MA                  0.328650   0.276217   1.190 0.234116
## state_MD                  0.072034   0.278258   0.259 0.795732
## state_ME                  0.626327   0.318628   1.966 0.049333 *
## state_MI                 -0.175716   0.299360  -0.587 0.557222
## state_MN                  0.946475   0.423632   2.234 0.025470 *
## state_MO                  0.418055   0.332524   1.257 0.208675
## state_MS                  0.188655   0.387669   0.487 0.626513
## state_MT                 -0.722324   0.450393  -1.604 0.108766
## state_NC                  0.360706   0.301329   1.197 0.231288
## state_ND                  0.463758   0.399055   1.162 0.245179
## state_NE                  1.955445   1.153467   1.695 0.090023 .
## state_NH                  0.892497   0.893507   0.999 0.317858
## state_NJ                 -0.312544   0.292876  -1.067 0.285902
## state_NM                  0.005605   0.319015   0.018 0.985981
## state_NV                  0.120891   0.283067   0.427 0.669323
## state_NY                  0.197671   0.274847   0.719 0.472015
## state_OH                  0.248846   0.287948   0.864 0.387476
## state_OK                 -0.109563   0.331343  -0.331 0.740899
## state_OR                  0.220597   0.294488   0.749 0.453804
## state_PA                  0.227908   0.280051   0.814 0.415753
## state_RI                  0.509140   0.340038   1.497 0.134314
## state_SC                  0.061979   0.317326   0.195 0.845146
## state_SD                  0.161153   0.490070   0.329 0.742279
## state_TN                  0.390304   0.341564   1.143 0.253165
## state_TX                  0.080721   0.282969   0.285 0.775441
## state_UT                  0.118237   0.315062   0.375 0.707451
## state_VA                 -0.139688   0.321993  -0.434 0.664417
## state_VT                  0.869334   0.339327   2.562 0.010409 *
## state_WA                        NA         NA      NA       NA
## state_WI                  0.449049   0.307859   1.459 0.144670
## state_WV                 -0.368655   0.362076  -1.018 0.308597
## state_WY                  1.004447   0.473142   2.123 0.033760 *
```

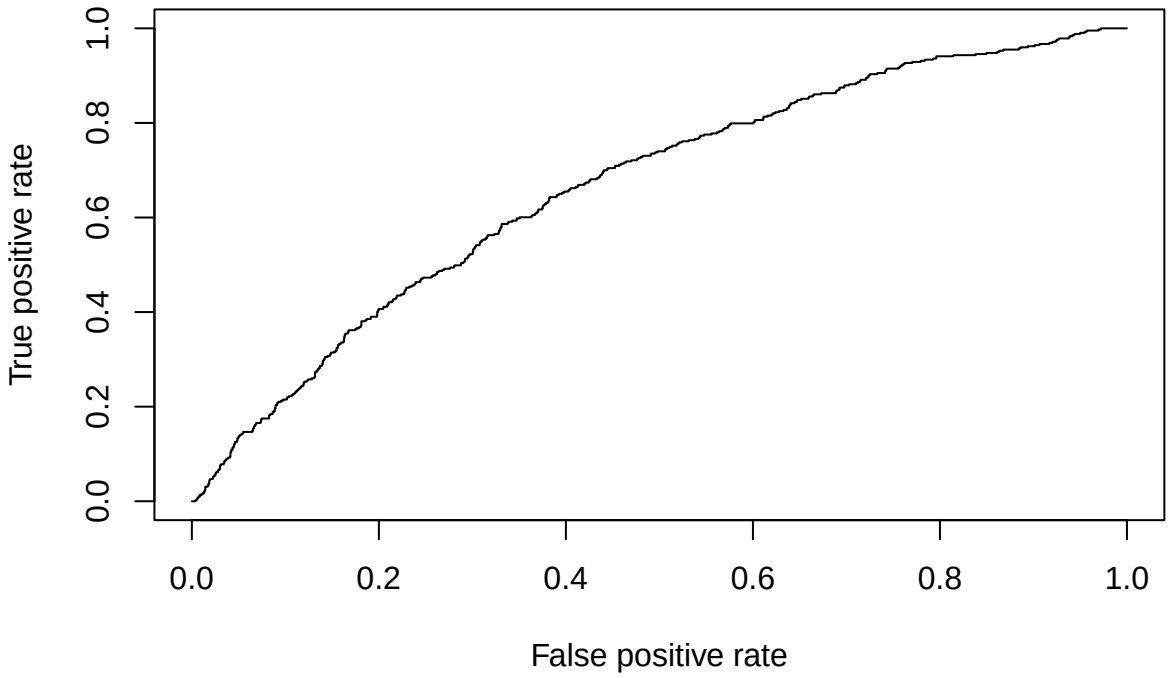
```
## state_XX                  -13.660173 806.644907  -0.017 0.986489
## connection_type_cellular   -0.222890   0.041763  -5.337 9.45e-08 ***
## connection_type_corporate  -0.202070   0.061390  -3.292 0.000996 ***
## connection_type_unknown     0.077192   0.197753   0.390 0.696281
## browser_chrome              0.154905   0.174287   0.889 0.374115
## browser_edge                0.181601   0.183557   0.989 0.322495
## browser_facebook           -3.412029   0.609055  -5.602 2.12e-08 ***
## browser_firefox             0.645659   0.213766   3.020 0.002524 **
## browser_instagram         -14.000816 138.390963  -0.101 0.919417
## browser_other             -14.711707 393.299680  -0.037 0.970161
## browser_safari              0.167652   0.186463   0.899 0.368589
## browser_samsungbrowser            NA         NA      NA       NA
## browser_platform_android   -0.188194   0.257373  -0.731 0.464649
## browser_platform_chrome   -13.361024 599.886288  -0.022 0.982231
## browser_platform_ios        0.157444   0.260400   0.605 0.545429
## browser_platform_linux    -14.300876 599.886182  -0.024 0.980981
## browser_platform_mac      -12.843172 599.886280  -0.021 0.982919
## browser_platform_other            NA         NA      NA       NA
## browser_platform_windows  -13.055701 599.886279  -0.022 0.982637
## device_mobile             -13.411295 599.886224  -0.022 0.982164
## device_other                      NA         NA      NA       NA
## device_tablet             -12.909384 599.886234  -0.022 0.982831
## coverage_type_cobra       -14.442812 882.743378  -0.016 0.986946
## current_company_none       13.606342 401.173337   0.034 0.972944
## current_company_other       1.439343   1.089548   1.321 0.186486
## gender_female              13.329518 401.173513   0.033 0.973494
## gender_male                13.385507 401.173513   0.033 0.973383
## gender_nonbinary           13.607307 401.174391   0.034 0.972942
## subsidy_medicaid            1.278227   0.801972   1.594 0.110968
## subsidy_no                  1.239845   0.797289   1.555 0.119928
## subsidy_yes                 1.415056   0.797130   1.775 0.075867 .
## season_Q4_2023              0.093024   0.032439   2.868 0.004134 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 25516  on 18405  degrees of freedom
## Residual deviance: 23497  on 18324  degrees of freedom
## AIC: 23661
##
## Number of Fisher Scoring iterations: 13
```

```
# Predict probabilities on the training data
phat_glm <- predict(glm_model,test, type = "response", probability=TRUE)

# Confusion matrix
test_cm <- factor(ifelse(test$conversion==1, "Yes","No"))
threshold_glm<- factor(ifelse(phat_glm>=0.5, "Yes","No"))
confusionMatrix(data=threshold_glm,reference=test_cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction   No  Yes
##        No  1456  166
##        Yes  844  257
##
##                Accuracy : 0.6291
##                  95% CI : (0.6106, 0.6473)
##     No Information Rate : 0.8447
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1455
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.6330
##             Specificity : 0.6076
##          Pos Pred Value : 0.8977
##          Neg Pred Value : 0.2334
##              Prevalence : 0.8447
##          Detection Rate : 0.5347
##    Detection Prevalence : 0.5957
##       Balanced Accuracy : 0.6203
##
##        'Positive' Class : No
##
```

```r
cm_glm <- confusionMatrix(data=threshold_glm,reference=test_cm)$overall['Accuracy']

# ROC
pred_glm <- prediction(phat_glm,test$conversion)
perf_glm <- performance(pred_glm,"tpr","fpr")
plot(perf_glm)
```




```
# AUC
perf_glm_auc <- performance(pred_glm,measure = "auc")
print(paste0(perf_glm_auc@y.values[[1]]))
```

```
## [1] "0.665374653098984"
```

Findings:

1) The logistic regression model revealed significant associations between conversion and several variables. Age and household income were positively associated with conversion (coefficients: 0.15326, $p < 0.001$; 0.53072, $p < 0.001$, respectively). State-level variables like AR, MN, VT, and WY also showed positive effects, indicating varying impacts across states. Categorical variables such as household size and Q4 2023 season had significant associations with conversion. However, variables related to browsers and platforms showed mixed significance levels. Overall, the model demonstrated a reasonable fit.

2) The confusion matrix accuracy was 0.63, indicating the model correctly predicted 63% of outcomes. The AUC of 0.67 suggests moderate capability in distinguishing between positive and negative classes. These metrics collectively indicate potential for improving the model's accuracy in predicting conversion outcomes and classifying them based on ROC curve analysis.

Classifcation Tree

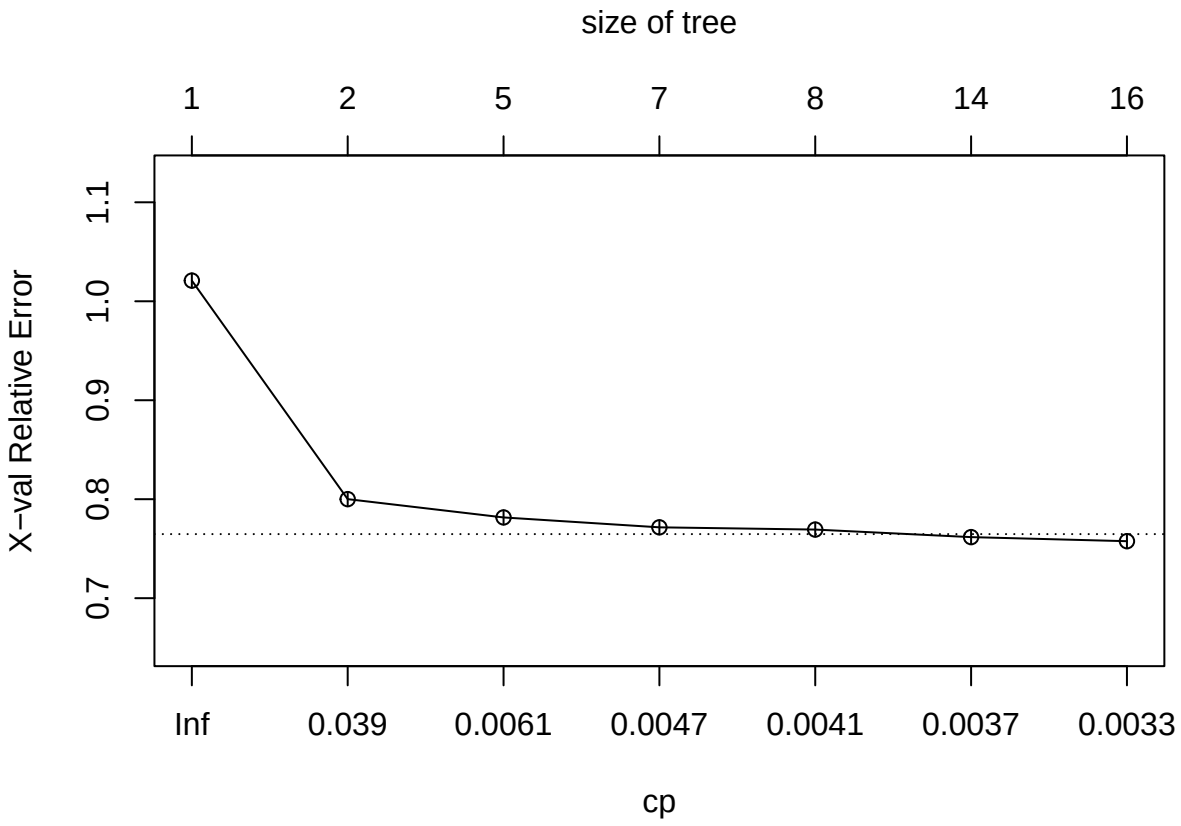
We implement classification trees due to their ability to handle complex predictor interactions and hierarchical decision rules. They are advantageous for their ability to handle both numerical and categorical data,

their interpretability through visual representation, and are robust against outliers and non-linear relationships. This method was explored to capture non-linear relationships and interactions that logistic regression might overlook.

```
# Classification Tree
ct_model<- rpart(conversion ~., data=train, method="class", cp=0.003, minsplit=10, xval=10)
printcp(ct_model)
```

```
##
## Classification tree:
## rpart(formula = conversion ~ ., data = train, method = "class",
##     cp = 0.003, minsplit = 10, xval = 10)
##
## Variables actually used in tree construction:
## [1] age                      browser_facebook         connection_type_cellular
## [4] device_mobile            gender_female            household_income
## [7] household_size           state_OH
##
## Root node error: 9203/18406 = 0.5
##
## n= 18406
##
##          CP nsplit rel error  xerror      xstd
## 1 0.2000435      0   1.00000 1.02086 0.0073693
## 2 0.0074251      1   0.79996 0.80007 0.0072221
## 3 0.0049440      4   0.77768 0.78159 0.0071929
## 4 0.0044551      6   0.76779 0.77160 0.0071761
## 5 0.0036944      7   0.76334 0.76931 0.0071721
## 6 0.0036401     13   0.73737 0.76171 0.0071586
## 7 0.0030000     15   0.73009 0.75758 0.0071510
```

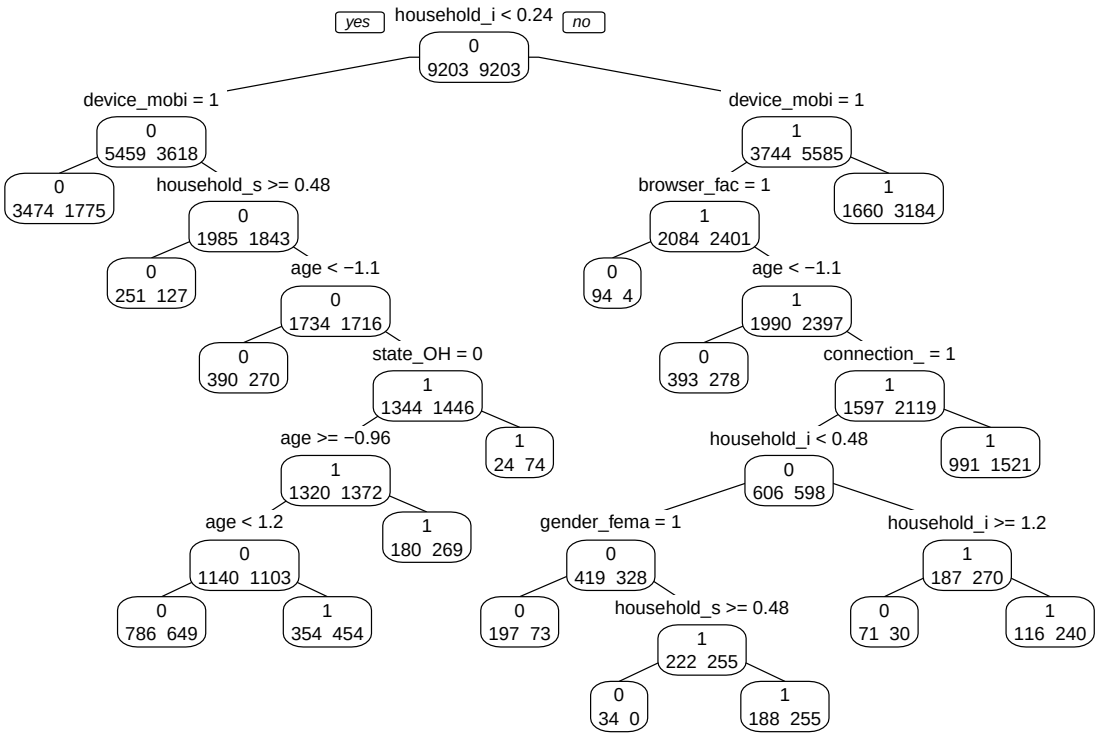
```
plotcp(ct_model)
```




```
pruned_ct_model <- prune(ct_model, cp=ct_model$cptable[which.min(ct_model$cptable[,"xerror"]),"CP"])
length(pruned_ct_model$frame$var[pruned_ct_model$frame$var=="<leaf>"])
```

```
## [1] 16
```

```
prp(pruned_ct_model, type=1, extra=1, split.font=1, varlen=-10)
```

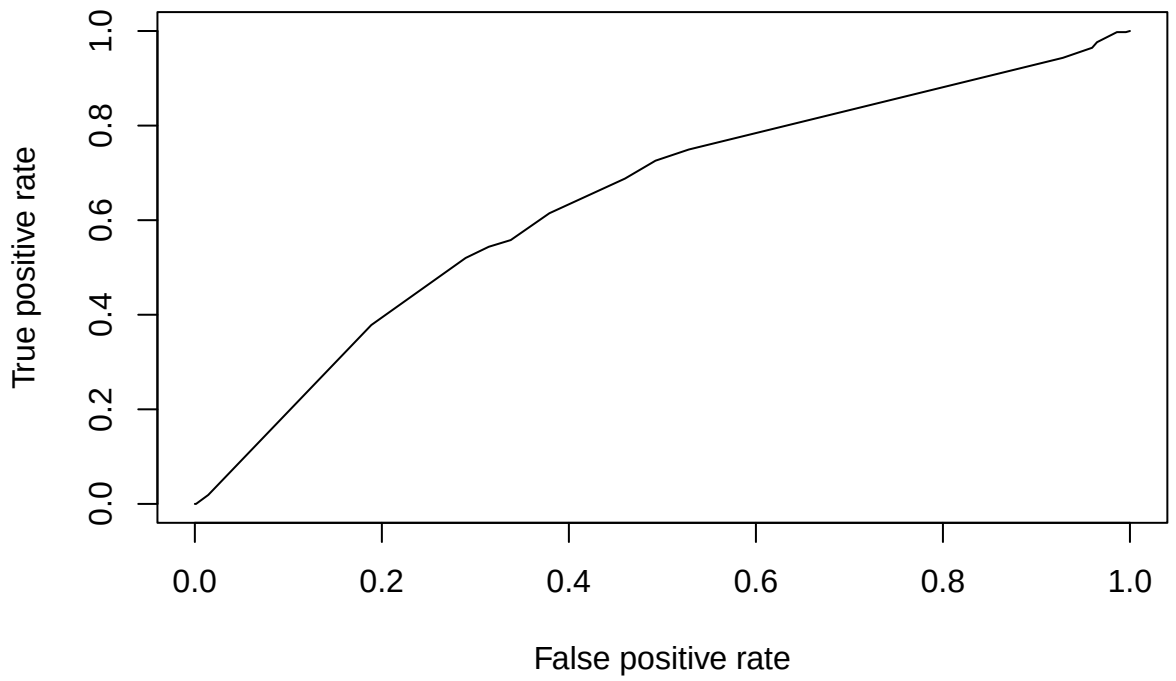



```
# Predict probabilities on the training data
phat_ct <- predict(pruned_ct_model,newdata=test, type="prob")[,2]

# Confusion matrix
test_cm <- factor(ifelse(test$conversion==1, "Yes","No"))
threshold_ct<- factor(ifelse(phat_ct>=0.5, "Yes","No"))
confusionMatrix(data=threshold_ct,reference=test_cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1428  163
##        Yes  872  260
##
##               Accuracy : 0.6199
##                 95% CI : (0.6014, 0.6382)
##    No Information Rate : 0.8447
##    P-Value [Acc > NIR] : 1
##
##                  Kappa : 0.1399
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.6209
##             Specificity : 0.6147
```

```
##          Pos Pred Value : 0.8975
##          Neg Pred Value : 0.2297
##              Prevalence : 0.8447
##          Detection Rate : 0.5244
##    Detection Prevalence : 0.5843
##       Balanced Accuracy : 0.6178
##
##        'Positive' Class : No
##
```

```
cm_ct <- confusionMatrix(data=threshold_ct,reference=test_cm)$overall['Accuracy']

# ROC
pred_ct <- prediction(phat_ct,test$conversion)
perf_ct <- performance(pred_ct,"tpr","fpr")
plot(perf_ct)
```




```
# AUC
perf_ct_auc <- performance(pred_ct,measure = "auc")
print(paste0(perf_ct_auc@y.values[[1]]))
```

```
## [1] "0.64077243293247"
```

Findings:

1) The classification tree identifies household_income as the root node, underscoring its substantial influence on predictions. Further down the tree, variables like device, browser, age, connection type, and gender are also pivotal in determining the likelihood of conversion. These results are consistent with the insights from our logistic regression model.

2) The confusion matrix accuracy was 0.62, indicating the model correctly predicted 62% of outcomes. The AUC of 0.64 suggests moderate capability in distinguishing between positive and negative classes. These metrics collectively indicate potential for improving the model's accuracy in predicting conversion outcomes and classifying them based on ROC curve analysis. Both metrics did slightly worse than the logistic regression.

Random Forest

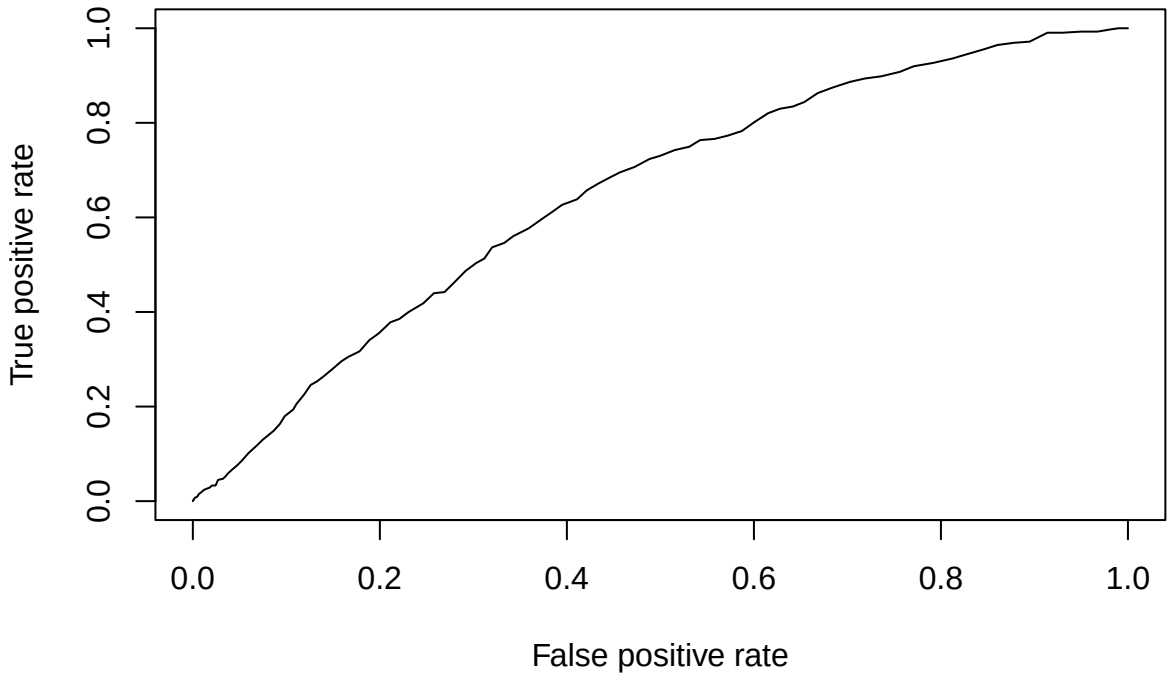
We implement random forests as they offer several advantages over individual decision trees. Random forests are an ensemble learning method that combines multiple decision trees to improve prediction accuracy and robustness. They mitigate overfitting by averaging multiple trees trained on different subsets of the data and features, thereby reducing variance and enhancing generalizability. Random forests can handle large datasets with many predictors and are less sensitive to outliers compared to single decision trees. They are effective in capturing complex interactions and non-linear relationships among variables, making them suitable for predicting conversion outcomes based on diverse user attributes and behaviors.

```
# Random Forest
rf_model <- randomForest(conversion ~ ., data=train, ntree=100,
                         nodesize=10, cv=5, importance=TRUE)

# Predict probabilities on the training data
phat_rf <- predict(rf_model,newdata=test, type="prob")[,2]

# Confusion matrix
test_cm <- factor(ifelse(test$conversion==1, "Yes","No"))
threshold_rf<- factor(ifelse(phat_rf>=0.5, "Yes","No"))
confusionMatrix(data=threshold_rf,reference=test_cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1656  227
##        Yes  644  196
##
##                Accuracy : 0.6801
##                  95% CI : (0.6622, 0.6976)
##     No Information Rate : 0.8447
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1308
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7200
##             Specificity : 0.4634
##          Pos Pred Value : 0.8794
##          Neg Pred Value : 0.2333
```

```
##              Prevalence : 0.8447
##          Detection Rate : 0.6082
##    Detection Prevalence : 0.6915
##       Balanced Accuracy : 0.5917
##
##        'Positive' Class : No
##
```

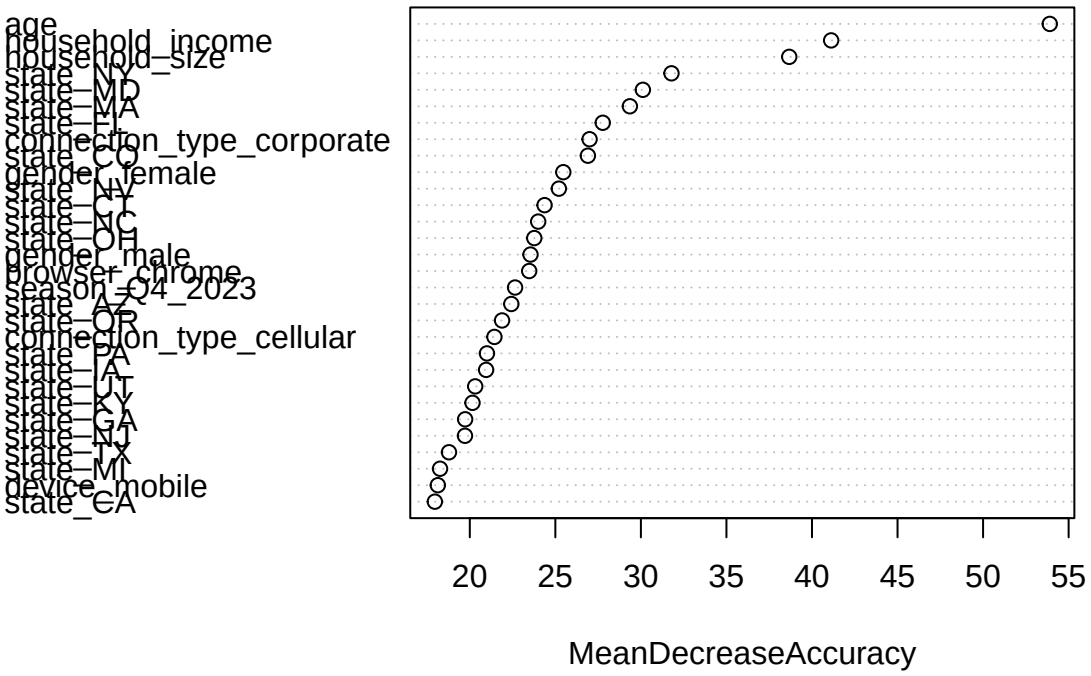
```
cm_rf <- confusionMatrix(data=threshold_rf,reference=test_cm)$overall['Accuracy']

# ROC
pred_rf <- prediction(phat_rf,test$conversion)
perf_rf <- performance(pred_rf,"tpr","fpr")
plot(perf_rf)
```




```
# AUC
perf_rf_auc <- performance(pred_rf,measure = "auc")
print(paste0(perf_rf_auc@y.values[[1]]))
```

```
## [1] "0.650267242265392"
```

```
# Variable importance plot
varImpPlot(rf_model,type=1,main="")
```



Findings:

1) The variable importance plot ranks features by their importance score, providing insight into which variables are most influential in predicting the outcome. From the plot, we can see that the most important factors are age, household income, and household size. These findings align with the results from the previous classification models, confirming the significant roles these variables play in predicting conversion.

2) The confusion matrix accuracy was 0.69, indicating the model correctly predicted 69% of outcomes. The AUC of 0.65 suggests moderate capability in distinguishing between positive and negative classes. These metrics collectively indicate potential for improving the model's accuracy in predicting conversion outcomes and classifying them based on ROC curve analysis. Both metrics did slightly better than the classification tree.

Model Evaluation

We now combine all the results from confusion matrix, ROC and AUC to choose the best model.

```
# Confusion Matrix
print(paste0("Logistic Regression Accuracy: ", cm_glm))
```

```
## [1] "Logistic Regression Accuracy: 0.629085567388909"
```

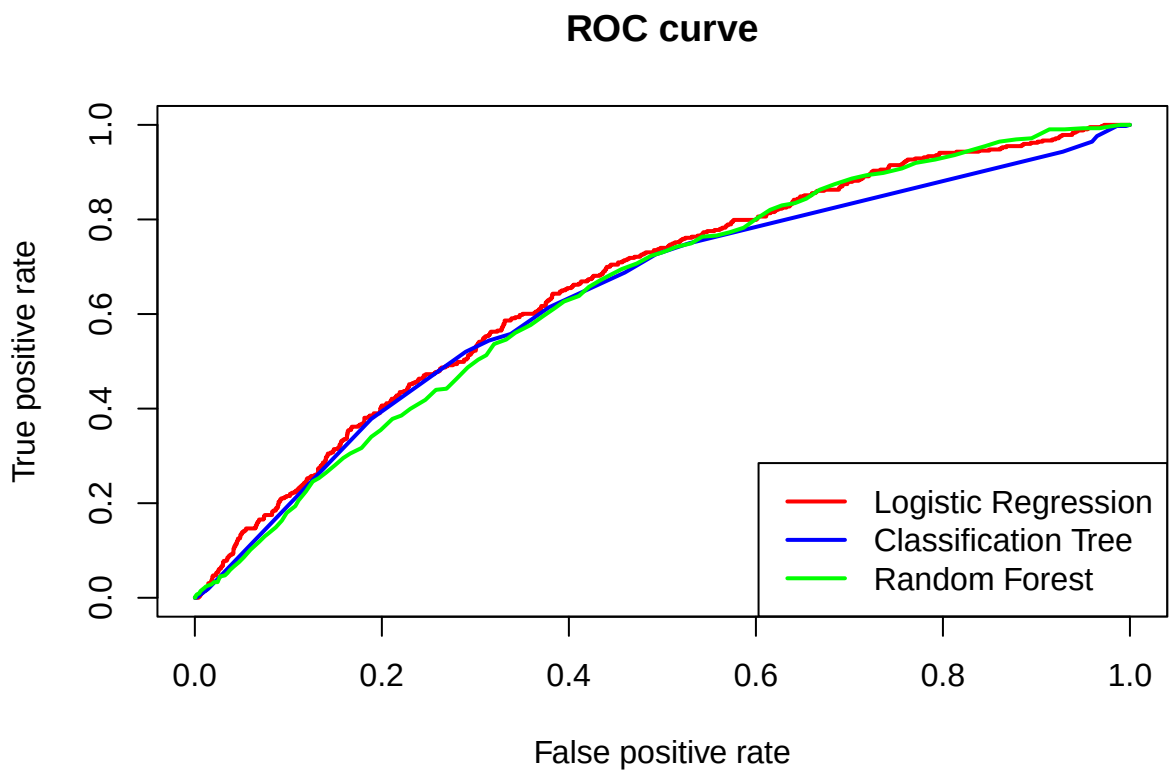
```r
print(paste0("Classification Tree Accuracy: ", cm_ct))
```

```
## [1] "Classification Tree Accuracy: 0.619904517076754"
```

```r
print(paste0("Random Forest Accuracy: ", cm_rf))
```

```
## [1] "Random Forest Accuracy: 0.680132207124495"
```

```r
# ROC
plot(perf_glm, col="red",lwd=2, main='ROC curve', cex.lab=1)
plot(perf_ct, add = T, col="blue",lwd=2)
plot(perf_rf, add = T, col="green",lwd=2)
legend("bottomright", legend=c("Logistic Regression","Classification Tree","Random Forest"),
       col=c("red","blue","green"), lty=1, lwd=2)
```




```r
# AUC
print(paste0("Logistic Regression AUC: ", perf_glm_auc@y.values[[1]]))
```

```
## [1] "Logistic Regression AUC: 0.665374653098984"
```

```r
print(paste0("Classification Tree AUC: ", perf_ct_auc@y.values[[1]]))
```

```
## [1] "Classification Tree AUC: 0.64077243293247"
```

```r
print(paste0("Random Forest AUC: ", perf_rf_auc@y.values[[1]]))
```

```
## [1] "Random Forest AUC: 0.650267242265392"
```

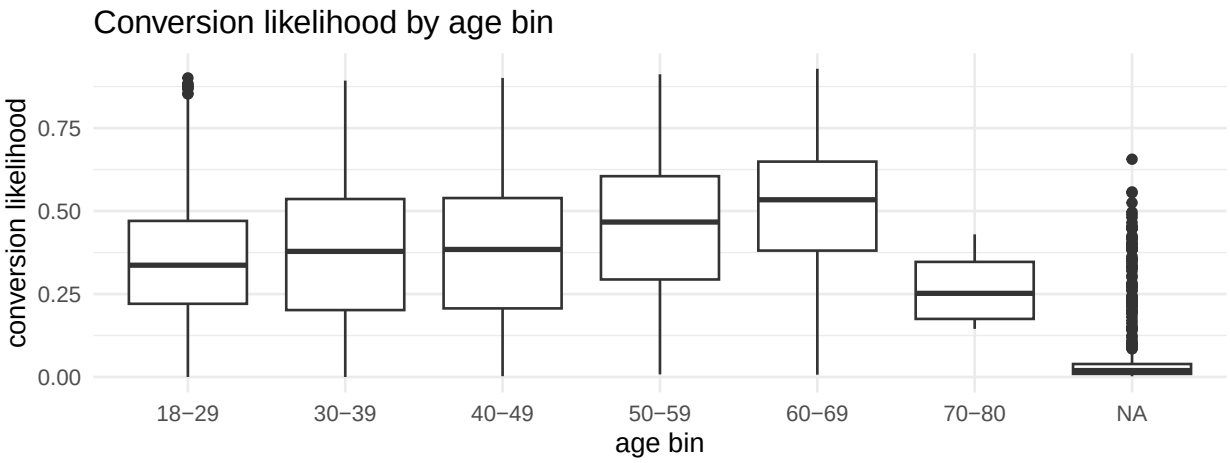
Findings:

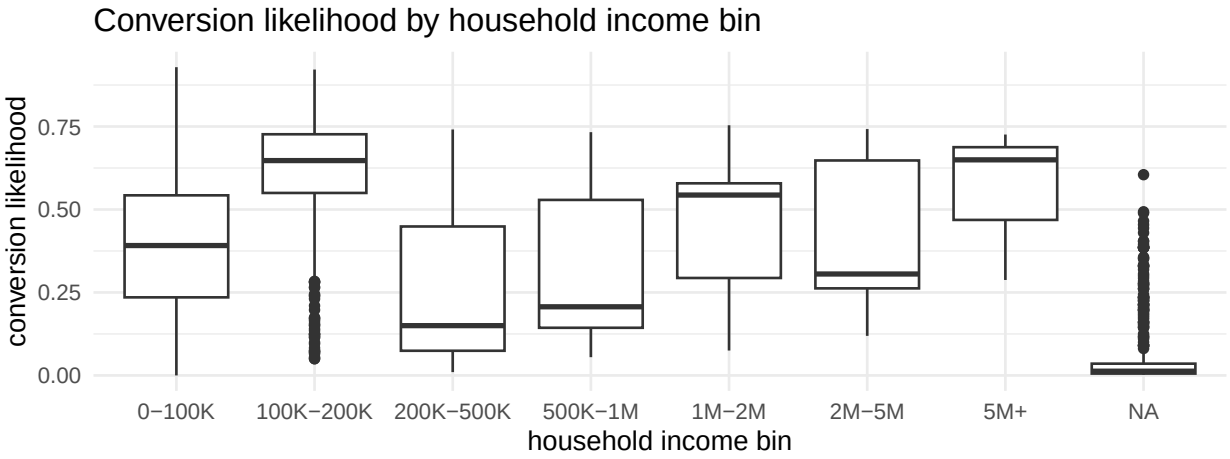
1) Random forest achieves the highest confusion matrix accuracy at 69%.
2) Logistic regression and random forest both outperform the classification tree in ROC analysis.
3) Logistic regression has the best AUC, with an area of 0.67.
4) Consequently, we will use both logistic regression and random forest models, and calculate the mean likelihood for predictions.

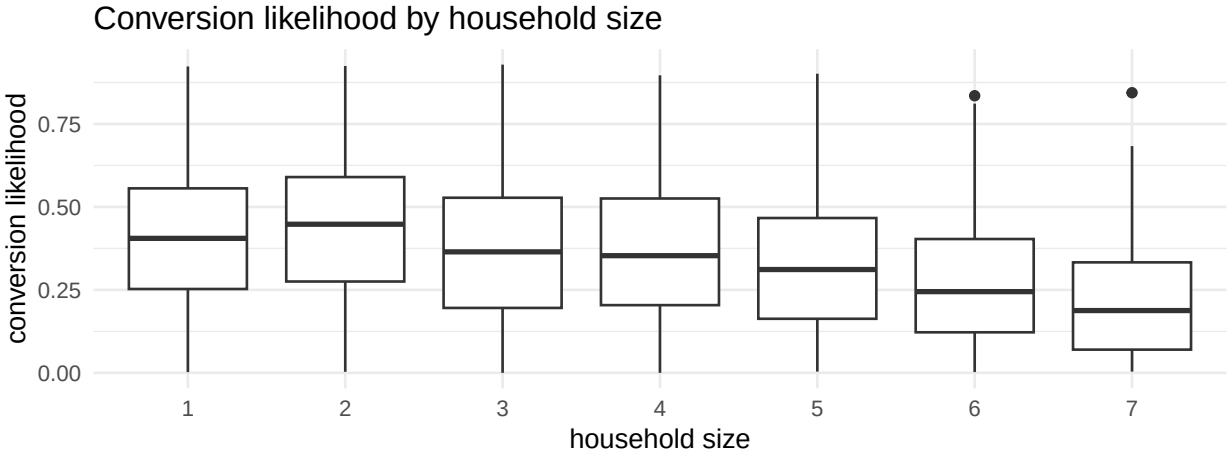
Predict Conversion Likelihood

To answer the first question, we use logistic regression and random forest to calculate conversion likelihood, and then average both likelihoods to obtain the desired probability.

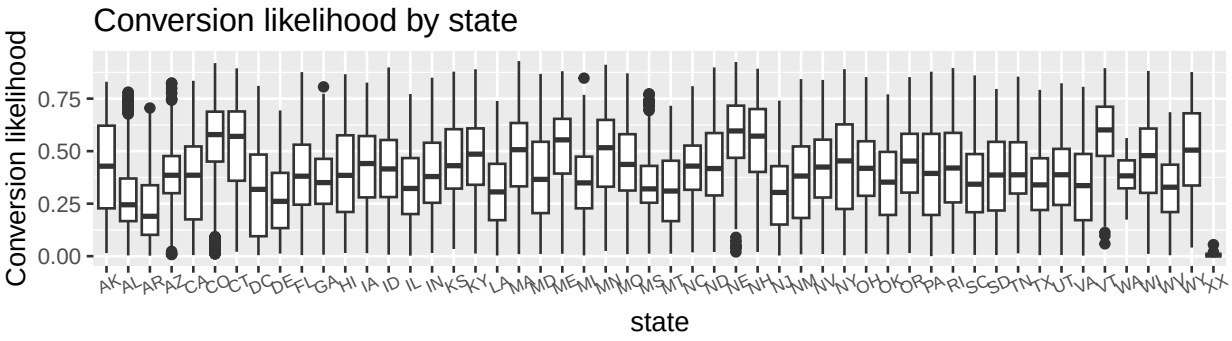
To answer the second question, we recreate the box plots using the averaged likelihood as the y-variable and other features as the x-variables. For numeric variables, we create bins for age and household income groups, and factors for household size. Categorical variables are also converted to factors. We then draw box plots to identify which types of callers have higher conversion likelihoods.

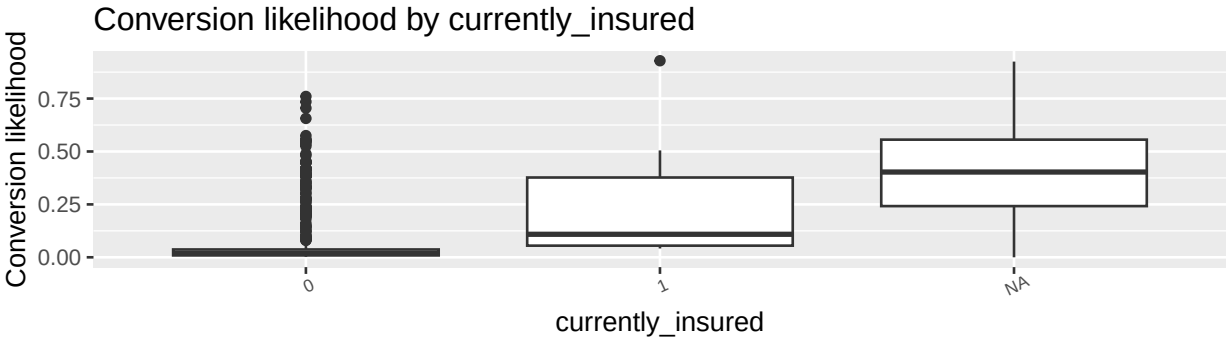

Conversion likelihood by age bin
conversion likelihood
0.00
0.25
0.50
0.75
18−29
30−39
40−49
50−59
60−69
70−80
NA
age bin

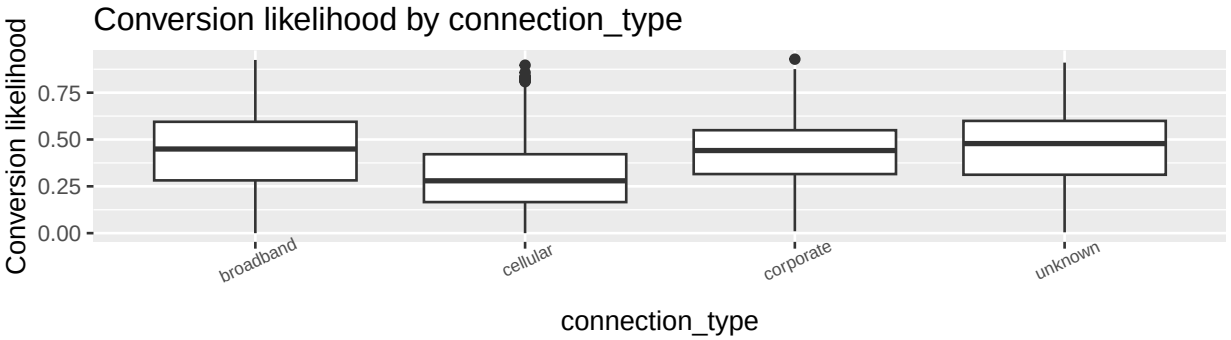

Conversion likelihood by household income bin
conversion likelihood
0.00
0.25
0.50
0.75
0−100K
100K−200K
200K−500K
500K−1M
1M−2M
2M−5M
5M+
NA
household income bin

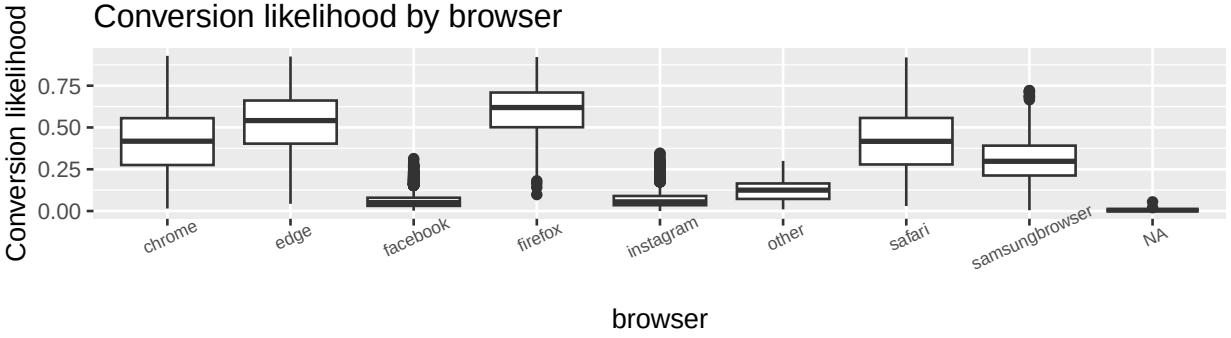

Conversion likelihood by household size
conversion likelihood
0.00
0.25
0.50
0.75
1
2
3
4
5
6
7
household size

```
## $`1`
```

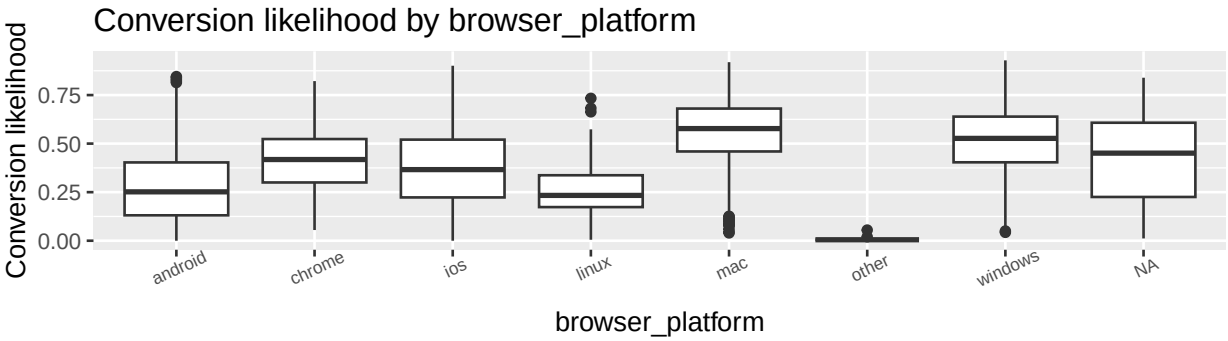

Conversion likelihood by state
Conversion likelihood
0.75
0.50
0.25
0.00
AK AL AR AZ CA CO CT DC DE FL GA HI IA ID IL IN KS KY LA MA MD ME MI MN MO MS MT NC ND NE NH NJ NM NV NY OH OK OR PA RI SC SD TN TX UT VA VT WA WI WV WY
state

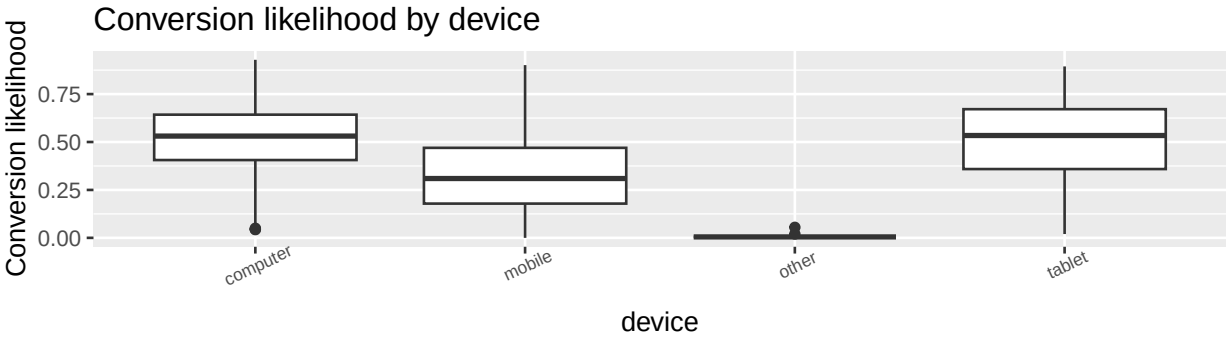

Conversion likelihood by currently_insured
Conversion likelihood
0.75
0.50
0.25
0.00
0
1
NA
currently_insured

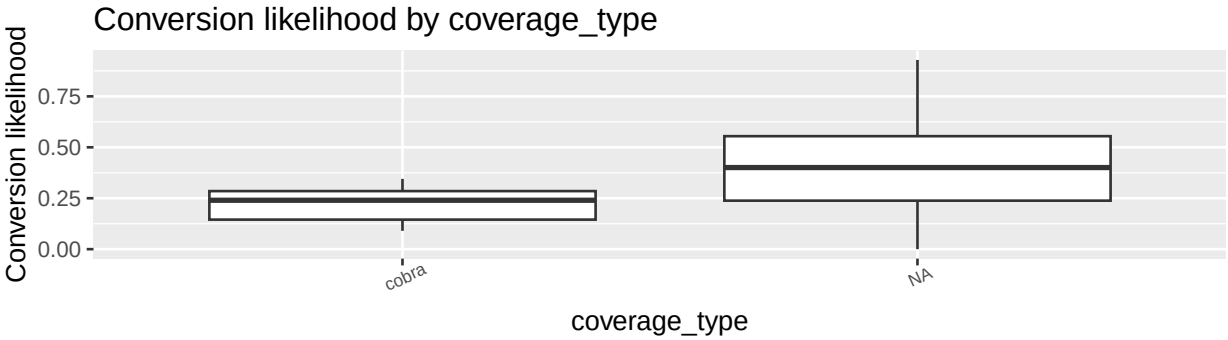

Conversion likelihood by connection_type
Conversion likelihood
0.75
0.50
0.25
0.00
broadband
cellular
corporate
unknown
connection_type

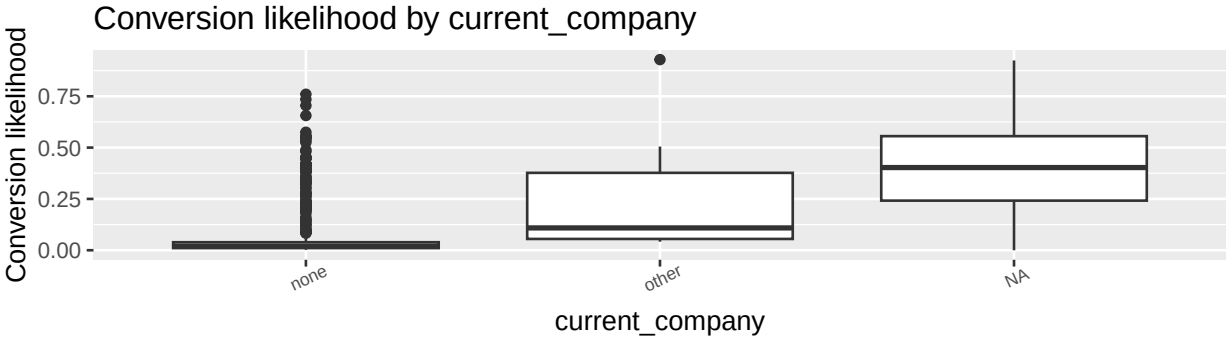

Conversion likelihood by browser
Conversion likelihood
0.75
0.50
0.25
0.00
chrome
edge
facebook
firefox
instagram
other
safari
samsungbrowser
NA
browser

```
##
## $`2`
```

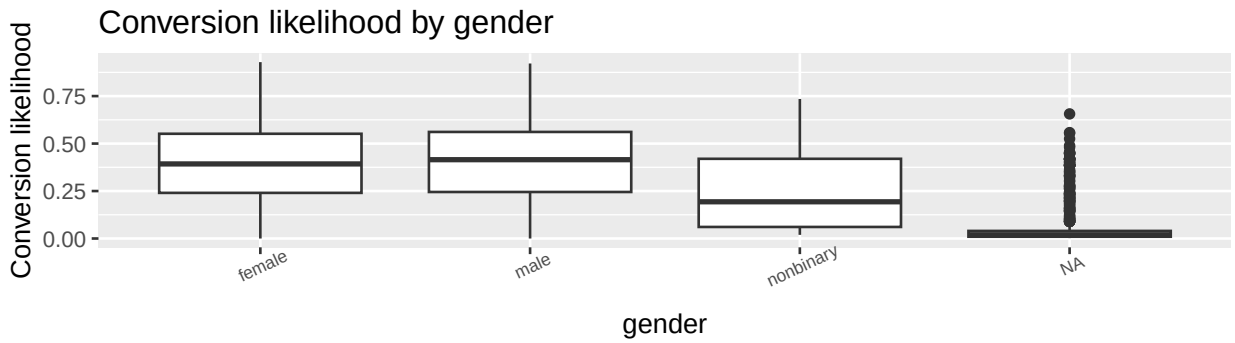

Conversion likelihood by browser_platform
Conversion likelihood
0.75
0.50
0.25
0.00
android
chrome
ios
linux
mac
other
windows
NA
browser_platform

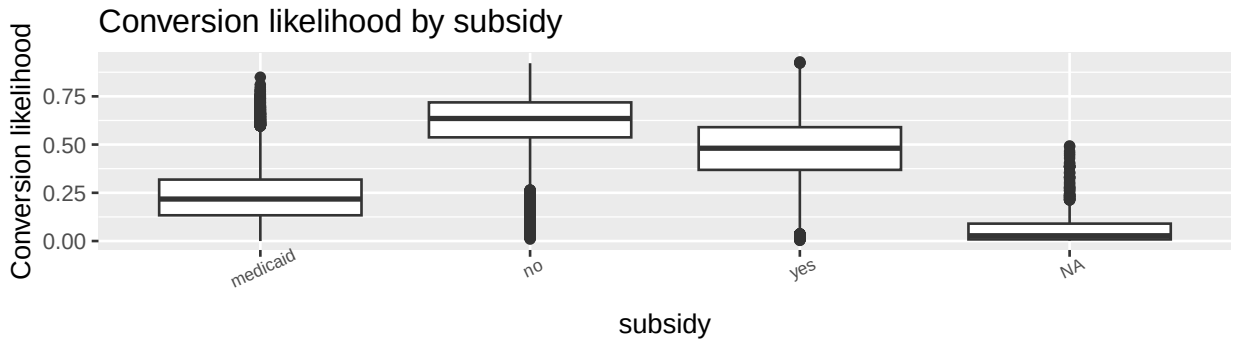

Conversion likelihood by device
Conversion likelihood
0.75
0.50
0.25
0.00
computer
mobile
other
tablet
device

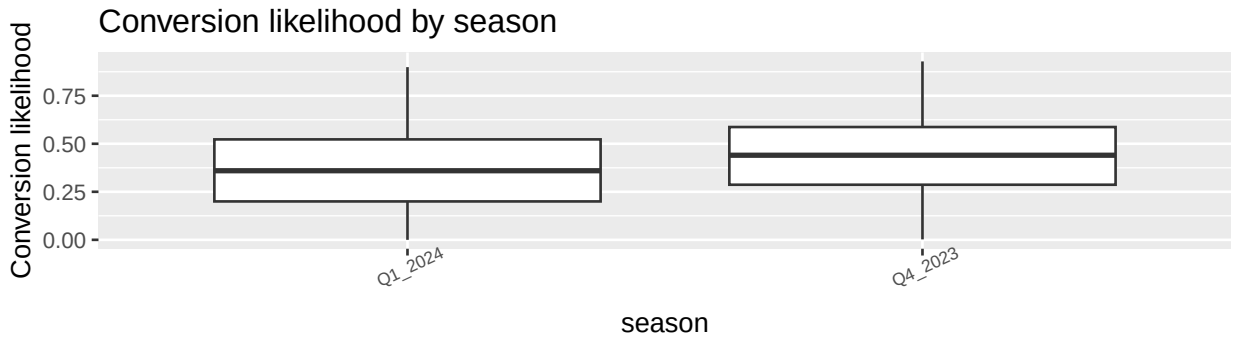

Conversion likelihood by coverage_type
Conversion likelihood
0.75
0.50
0.25
0.00
cobra
NA
coverage_type


Conversion likelihood by current_company
Conversion likelihood
0.75
0.50
0.25
0.00
none
other
NA
current_company

```
##
## $`3`
```








```
##
## attr(,"class")
## [1] "list"      "ggarrange"
```

From our previous findings, we know that age, household income and household size are the three most determining factors in predicting conversion likelihood. Other call attributes gave mixed signals throughout different machine learning models. Combined with the box plots above we can see which types of calls made by which type of callers are more likely to convert.

Findings:

1) *Age*: The likelihood of conversion shows an upward trend with increasing user age, reaching its highest point among callers aged 60-69. However, this trend reverses for callers aged 70-80, where the likelihood of conversion begins to decline. Therefore we believe callers aged 60-69 are most likely to convert. This trend may be influenced by factors such as financial stability and increased healthcare needs typical of this age group. Older adults nearing retirement age often have more disposable income and may be actively seeking new services or plans to cater to their evolving lifestyle and healthcare requirements.

2) *Household income*: The likelihood of conversion peaks among callers with household incomes ranging from 100K-200K. The trend initially rises, then dips, followed by a gradual increase. Therefore, we conclude that callers within this income bracket are most likely to convert. Factors influencing this trend could include purchasing power and affordability aligned with service offerings, where households in this income range may find the value proposition compelling enough to initiate conversion. Additionally, economic stability and disposable income within this bracket might facilitate easier adoption of new plans or services.

3) *Household size*: The likelihood of conversion decreases as household size increases, peaking when the household size is 2. Smaller households, like couples or individuals, may have more focused needs that align closely with conversion offerings, leading to higher likelihoods of adoption. In contrast, larger households, such as families, often face more complex decision-making dynamics, including existing commitments and diverse preferences among members, which can lower the likelihood of adopting new plans. Therefore, households of size 2 show a higher likelihood for conversion compared to larger household sizes.

Limitations

An accuracy of 60%+ suggests room for improvement, which could be influenced by several factors:

1) As the project mentioned, only one buyer (buyer_id = 63253) has conversion data in the dataset and we do not know the status of calls bought by the other buyers. This means we don't have information on whether calls bought by other buyers led to conversions or not. This imbalance in data could affect how well our models predict outcomes for different types of buyers. This could cause a bias problem especially when dealing with buyers who behave differently from the one we have conversion data on.

2) There were too many missing values in crucial columns such as weight and BMI, which are typically significant indicators of health status and could influence the likelihood of converting to a health plan. Since these columns had over 85% missing data, using mean imputation wasn't feasible, so removing the columns became necessary. To improve our model's accuracy, it would be beneficial to collect more data on weight, BMI, and possibly credit rating and seller type. This additional information could provide more insights and potentially enhance the predictive capabilities of our model.

3) Time constraints significantly influenced our approach during this project. Given more time, we could have explored additional methods and machine learning models. For instance, techniques like SMOTE could have been implemented to address imbalanced data issues. Moreover, leveraging advanced algorithms such as XGBoost could have enhanced the training and prediction accuracy of our models.

Conclusion

In conclusion, our analysis successfully predicted the likelihood of caller conversion using various machine learning models. Key features such as household income, household size, and age emerged as significant factors influencing conversion outcomes. These insights underscore the importance of demographic and socioeconomic variables in understanding user behavior. By accurately predicting conversion likelihood,

businesses can target high-conversion demographics, and enhance customer acquisition strategies. This can lead to improved marketing ROI, increased sales revenue, and better overall business performance. These findings not only validate the relevance of data-driven decision-making but also highlight the potential for leveraging predictive analytics to gain a competitive edge in the market.