Classification Report

AUTHOR
Aaron Younger

PUBLISHED
December 10, 2025

Business Understanding

Business Problem

The data used in this analysis comes from direct market campaigns via phone calls by a Portuguese banking institution. The primary goal of this analysis is to build classification models to predict whether a client will subscribe to a term deposit. A term deposit represents a contract between an investor and a financial institution where a sum of money is locked away for a predetermined period in exchange for a fixed interest rate. The ability for banks to predict whether a client will subscribe to a term deposit is highly valuable for several reasons. First, banks allocate significant budget on marketing campaigns, and predictive models that helps identify clients who are most likely to say “yes” improves campaign efficiency and cost. Second, accurately targeting likely subscribers increases the volume of term deposits, which strengthens the bank’s funding base in a stable and predictable manner. Finally, these models can help support strategic decision-making as banks can tailor product offerings and interest rates based off specific customer segments.

Based off this information, the **Business Problem** is how to identify which clients are most likely to subscribe to a term deposit.

Two Research Questions

Along with the business problem, this report explores two research questions:

1. Which variables are most significant in predicting term deposit subscriptions?
2. Are there specific times of the year when clients are more likely to subscribe to a term deposit?

Now that the business problem and research questions have been clearly defined, the next step in this report is exploring the data to better understand its structure, key variables, and potential patterns relevant to term deposit subscriptions. This **Data Understanding** phase provides the foundational insight needed for effective data preparation for modeling.

Data Understanding

Import Dataset

```
# A tibble: 6 × 17
    age job        marital education default balance housing loan  contact   day
```

```
  <dbl> <fct>      <fct>   <fct>     <fct>     <dbl> <fct>   <fct> <fct>    <dbl>
1    30 unemployed married primary   no         1787 no      no    cellul…     19
2    33 services   married secondary no         4789 yes     yes   cellul…     11
3    35 management single  tertiary  no         1350 yes     no    cellul…     16
4    30 management married tertiary  no         1476 yes     yes   unknown      3
5    59 blue-coll… married secondary no            0 yes     no    unknown      5
6    35 management single  tertiary  no          747 no      no    cellul…     23
# ℹ 7 more variables: month <fct>, duration <dbl>, campaign <dbl>, pdays <dbl>,
#   previous <dbl>, poutcome <fct>, y <fct>
```

when importing the dataset, I converted the categorical variables stored as character strings to factors. This step is important because R treats factor variables differently from character variables. Factors allow for proper statistical analysis and model building.

EDA

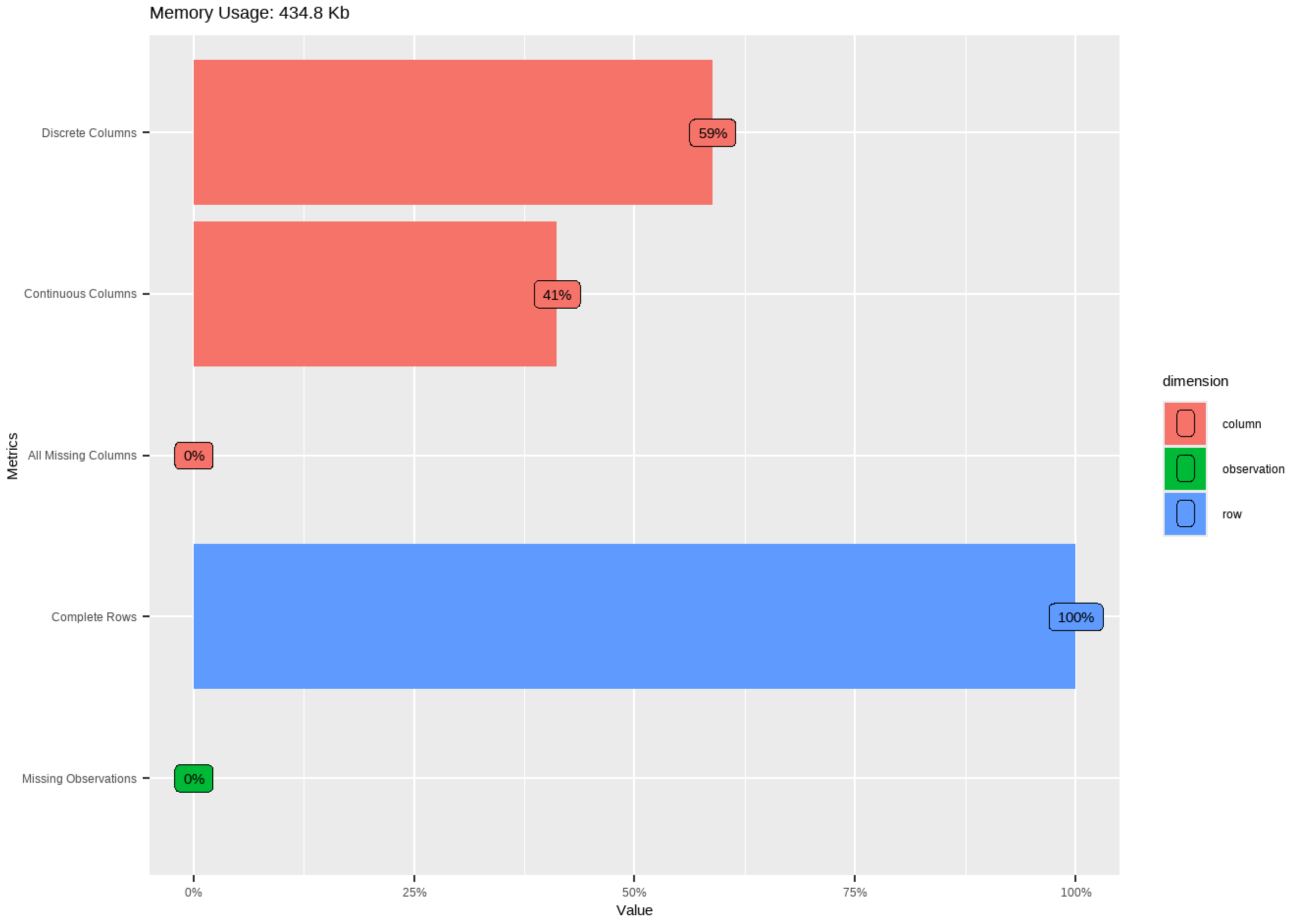
Dataset EDA

Number of Rows and Columns

```
# A tibble: 2 × 2
  Statistic         Value
  <chr>             <int>
1 Number of Columns    17
2 Number of Rows     4521
```

This dataset contains 17 variables, as indicated by the number of columns, and 4,521 observations, as indicated by the number of rows.

Data Structure and Completeness Overview



This graph shows that ten variables are categorical variables and the remaining seven variables are numeric. This dataset also contains no missing values.

Below is a short description of all the variables found in this dataset.
- Age: Represents the Age of the Client.
- Job: Represents the type of job the Client has.
- Marital: Represents the marital status of the client.
- Education: Represents the Education level of the client.
- Default: A binary variable that shows if the client has credit in default. - Balance: The Average yearly income the client is earning, value in Euros.
- Housing: A binary variable showing if the client has a house loan.
- Loan: A binary variable showing if the client has a personal loan.
- Contact: Represents the clients communication type.
- Day: Represents the last contact a client has had of the month.
- Month: Represents the last contact month of the year for a client.
- Duration: Represents the last contact duration of the client in seconds.
- Campaign: Represents the number of times the client was reached during a campaign, includes last contact.
- Pdays: Represents the number of days that has passed since a client has been last contacted from a previous campaign. (-1 means client was not previously contacted).try and convert this into a mutli layer nominal variable. (not contacted, low, medium, high contacted).
- Previous: Represents the total number of contacts before the campaign for the client.

- Poutcome: Outcome of the previous marketing campaign.
- Y (Dependent Variable): A binary variable showing if the client has subscribed a term deposit. (yes or no).

Categorical Variable EDA

Since the dataset contains both categorical and numerical variables, the exploratory data analysis (EDA) will be conducted in two parts: one focusing on categorical variables and the other on numeric variables. First, this analysis will explore the relationships and patterns found in categorical variables.

Create Categorical Variables

```
Distribution of Bins-Pdays



  0 - 150 Days 151 - 300 Days      300+ Days  Not Contacted 
           238            335            243           3705 



 Distribution of Bins-Previous



 High Contact   Low Contact Not Contacted 
           99           717          3705 
```

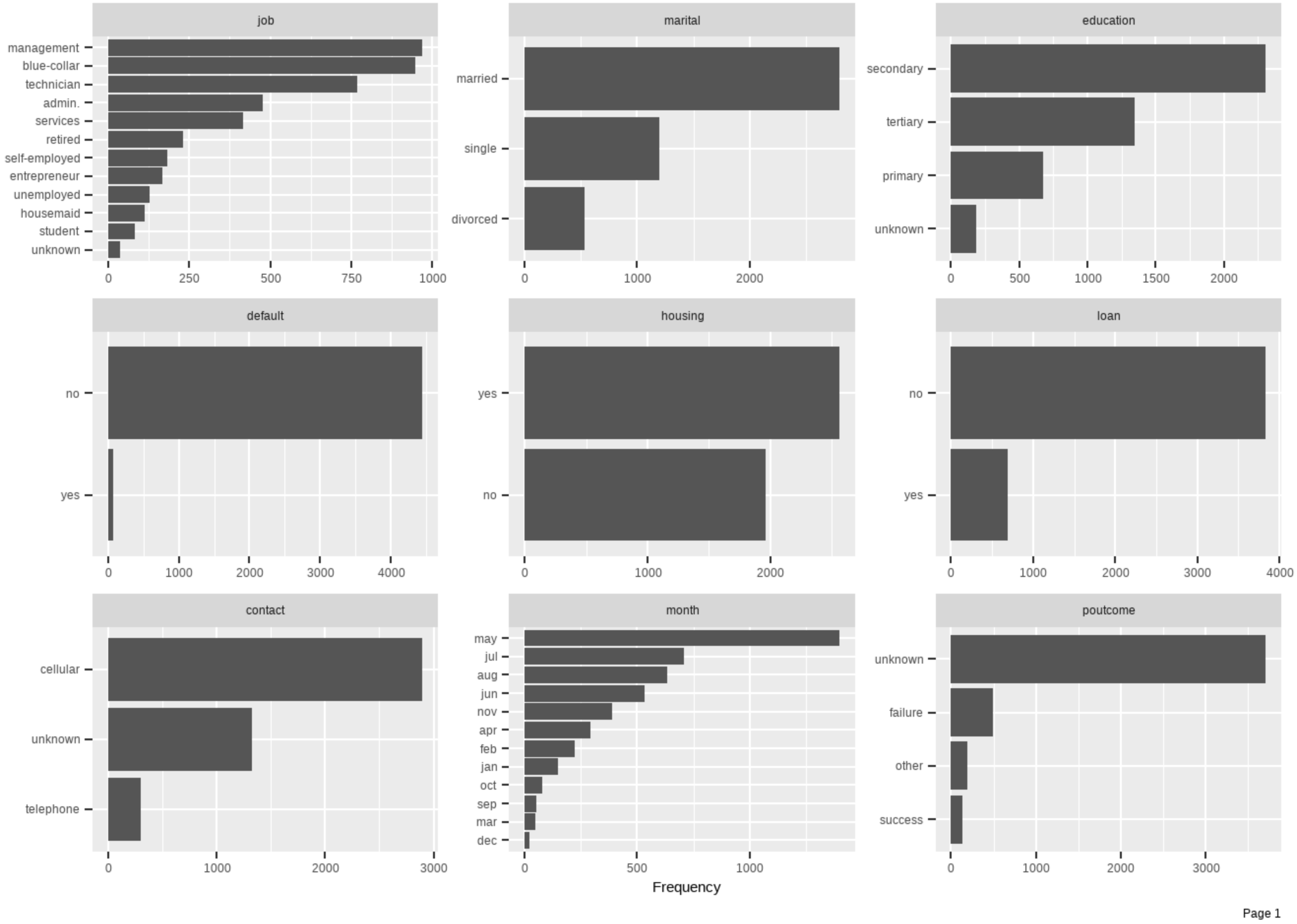
Before creating a subset containing only categorical variables, I first addressed two variables—pdays and previous—which were heavily dominated by single values (–1 and 0). To reduce this skewness and improve interpretability, both variables were grouped into binned categories. Binning these variables allowing their categories to more meaningfully reflect their influence on the dependent variable.

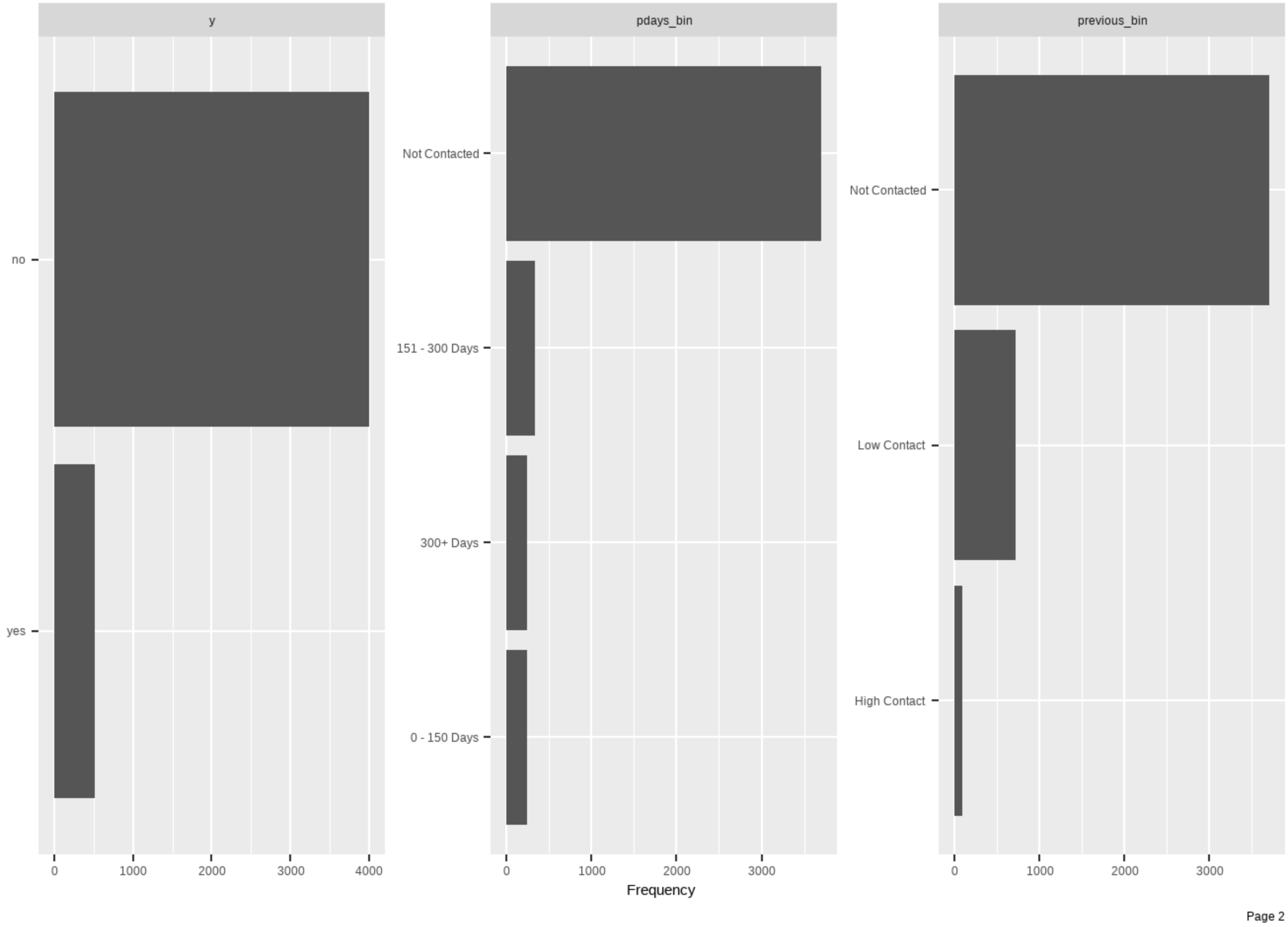
Create Categorical Subset

```
# A tibble: 6 × 12
  job       marital education default housing loan  contact month poutcome y    
  <fct>     <fct>   <fct>     <fct>   <fct>   <fct> <fct>   <fct> <fct>    <fct>
1 unemploy… married primary   no      no      no    cellul… oct   unknown  no   
2 services  married secondary no      yes     yes   cellul… may   failure  no   
3 manageme… single  tertiary  no      yes     no    cellul… apr   failure  no   
4 manageme… married tertiary  no      yes     yes   unknown jun   unknown  no   
5 blue-col… married secondary no      yes     no    unknown may   unknown  no   
6 manageme… single  tertiary  no      no      no    cellul… feb   failure  no   
# ℹ 2 more variables: pdays_bin <fct>, previous_bin <fct>
```

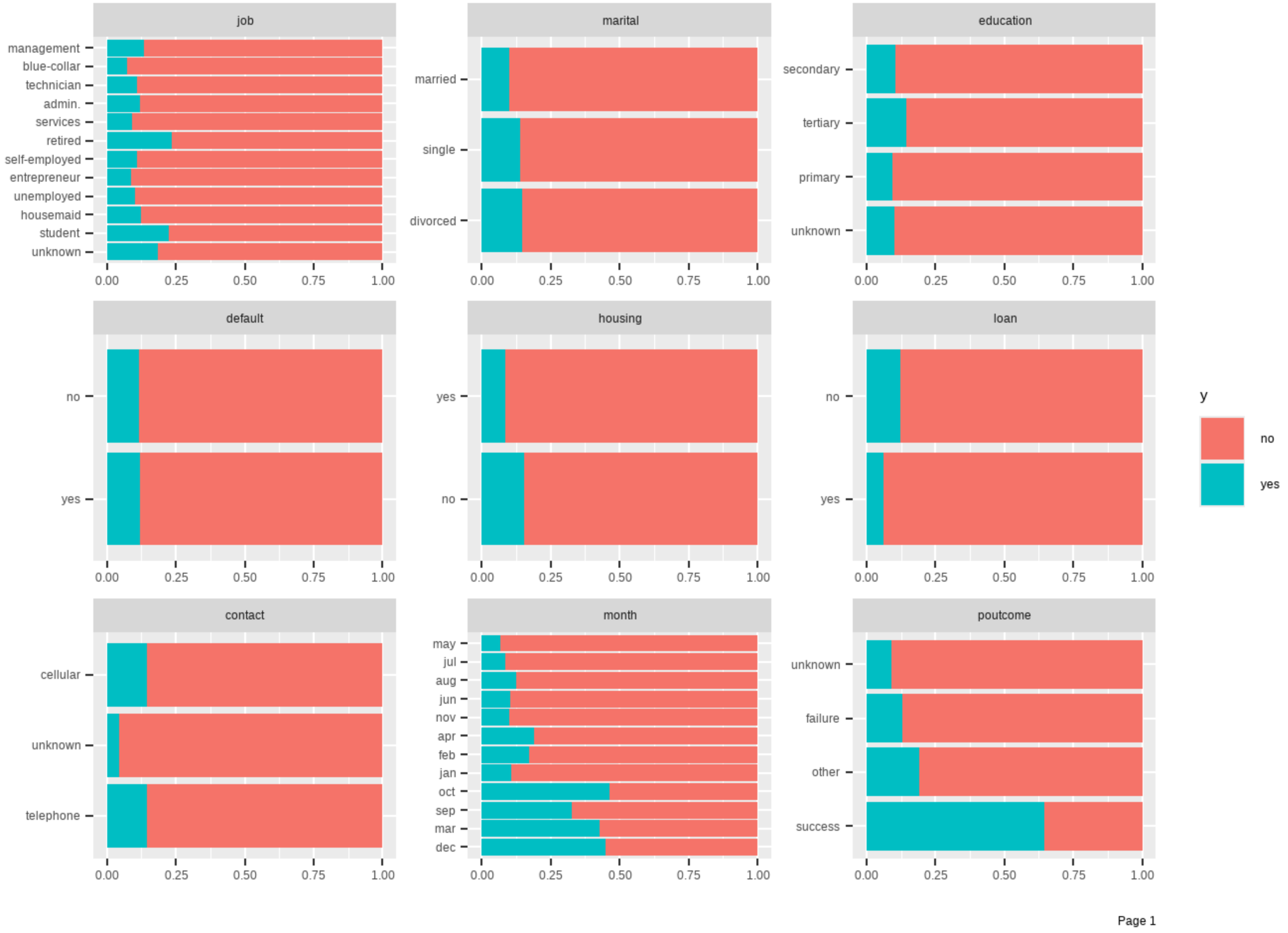
A subset from the original dataset was made just containing categorical variables.

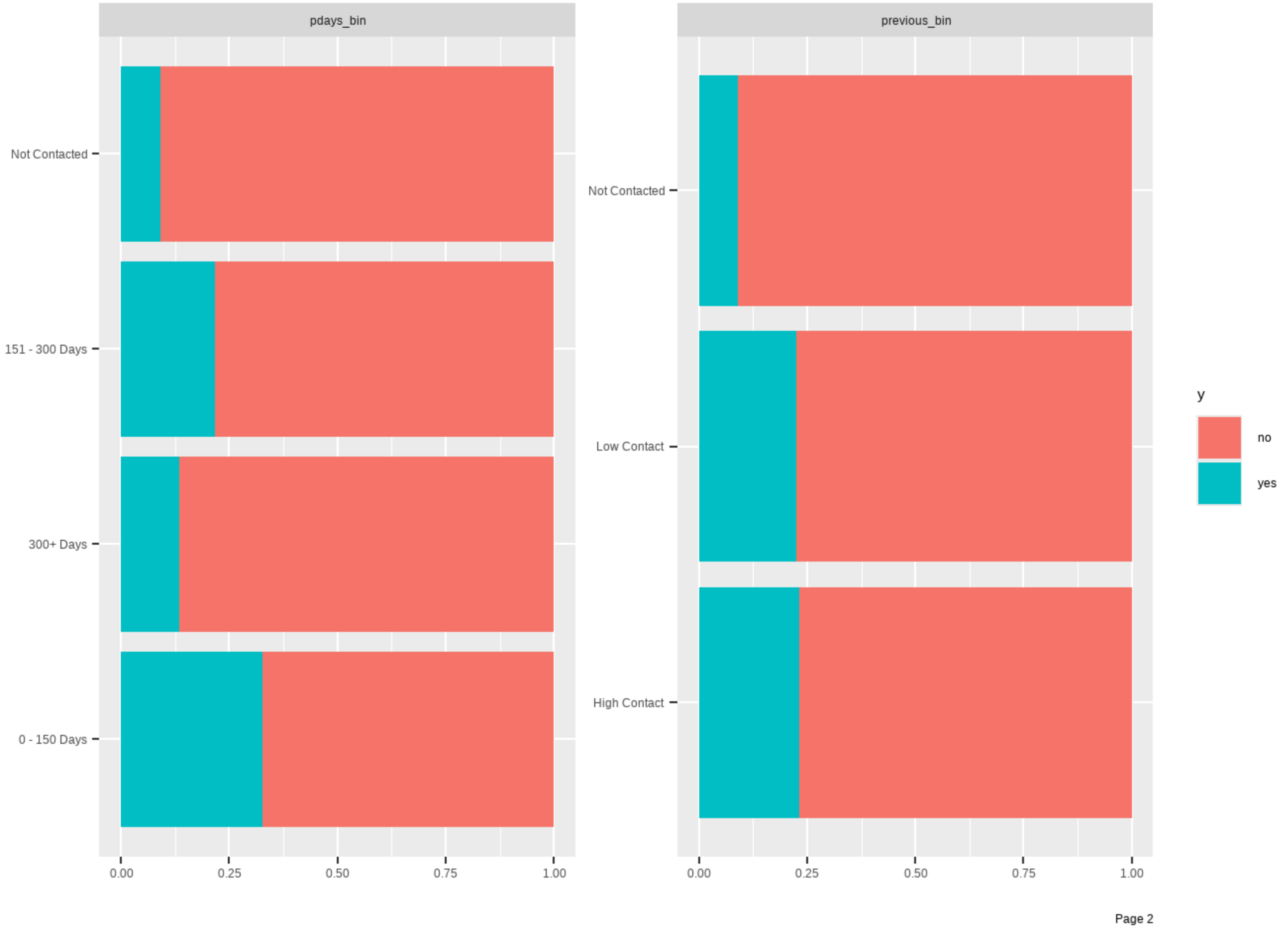
Explore Proportions of Categorical Variables


job
management
blue-collar
technician
admin.
services
retired
self-employed
entrepreneur
unemployed
housemaid
student
unknown
0
250
500
750
1000
marital
married
single
divorced
0
1000
2000
education
secondary
tertiary
primary
unknown
0
500
1000
1500
2000
default
no
yes
0
1000
2000
3000
4000
housing
yes
no
0
1000
2000
loan
no
yes
0
1000
2000
3000
4000
contact
cellular
unknown
telephone
0
1000
2000
3000
month
may
jul
aug
jun
nov
apr
feb
jan
oct
sep
mar
dec
0
500
1000
Frequency
poutcome
unknown
failure
other
success
0
1000
2000
3000
Page 1



An important step in exploratory data analysis is to check level proportions within categorical variables. If a categorical variable exhibits class imbalance, the minority category may be underweighted or overlooked by the model, potentially causing important relationships within smaller groups to be missed. As seen from the bar plots, class imbalance persists in almost every categorical variable. This problem will need to be addressed.


job
management
blue-collar
technician
admin.
services
retired
self-employed
entrepreneur
unemployed
housemaid
student
unknown
marital
married
single
divorced
education
secondary
tertiary
primary
unknown
default
no
yes
housing
yes
no
loan
no
yes
contact
cellular
unknown
telephone
month
may
jul
aug
jun
nov
apr
feb
jan
oct
sep
mar
dec
poutcome
unknown
failure
other
success
0.00
0.25
0.50
0.75
1.00
y
no
yes
Page 1



On top of checking for class level imbalance it is equally as important to check for distribution of subscription outcomes across each categorical variable. Based off the bar plots, there is clear class imbalance among the dependent variable. “No” responses substantially outnumber “yes” responses among all the categorical variable.

Proportion of Dependent Variable

```
Propotion of the Dependent Variable


     no     yes 
0.88476 0.11524 
```

This table showing the proportion of the dependent variable further confirm class imbalance, with 88.5% of clients not subscribing and only 11.5% subscribing. This class imbalance can be addressed using several modeling techniques, which will be explored later in the analysis.

Chi-Square Test of Signficance

```
                           Variable   Chi_Square DF    P_Value
job.X-squared                   job 6.898829e+01 NA 0.00019996
marital.X-squared           marital 1.903006e+01 NA 0.00019996
education.X-squared       education 1.523658e+01 NA 0.00139972
default.X-squared           default 7.671703e-03 NA 1.00000000
```

```
housing.X-squared           housing 4.954390e+01 NA 0.00019996
loan.X-squared                 loan 2.248136e+01 NA 0.00019996
contact.X-squared           contact 8.786986e+01 NA 0.00019996
month.X-squared               month 2.505001e+02 NA 0.00019996
poutcome.X-squared         poutcome 3.868774e+02 NA 0.00019996
pdays_bin.X-squared       pdays_bin 1.624645e+02 NA 0.00019996
previous_bin.X-squared previous_bin 1.187559e+02 NA 0.00019996
```

Along with checking class and level imbalance, a chi-square test is used to evaluate whether each categorical variable is significantly associated with the dependent variable. The table above presents the p-values for each test, and variables with p-values below 0.01 are considered statistically significant. All variables except default show a significant association with the subscription outcome. This is helpful because it indicates which categorical variables will most likely contribute to accurate classification of the dependent variable.

Numeric EDA

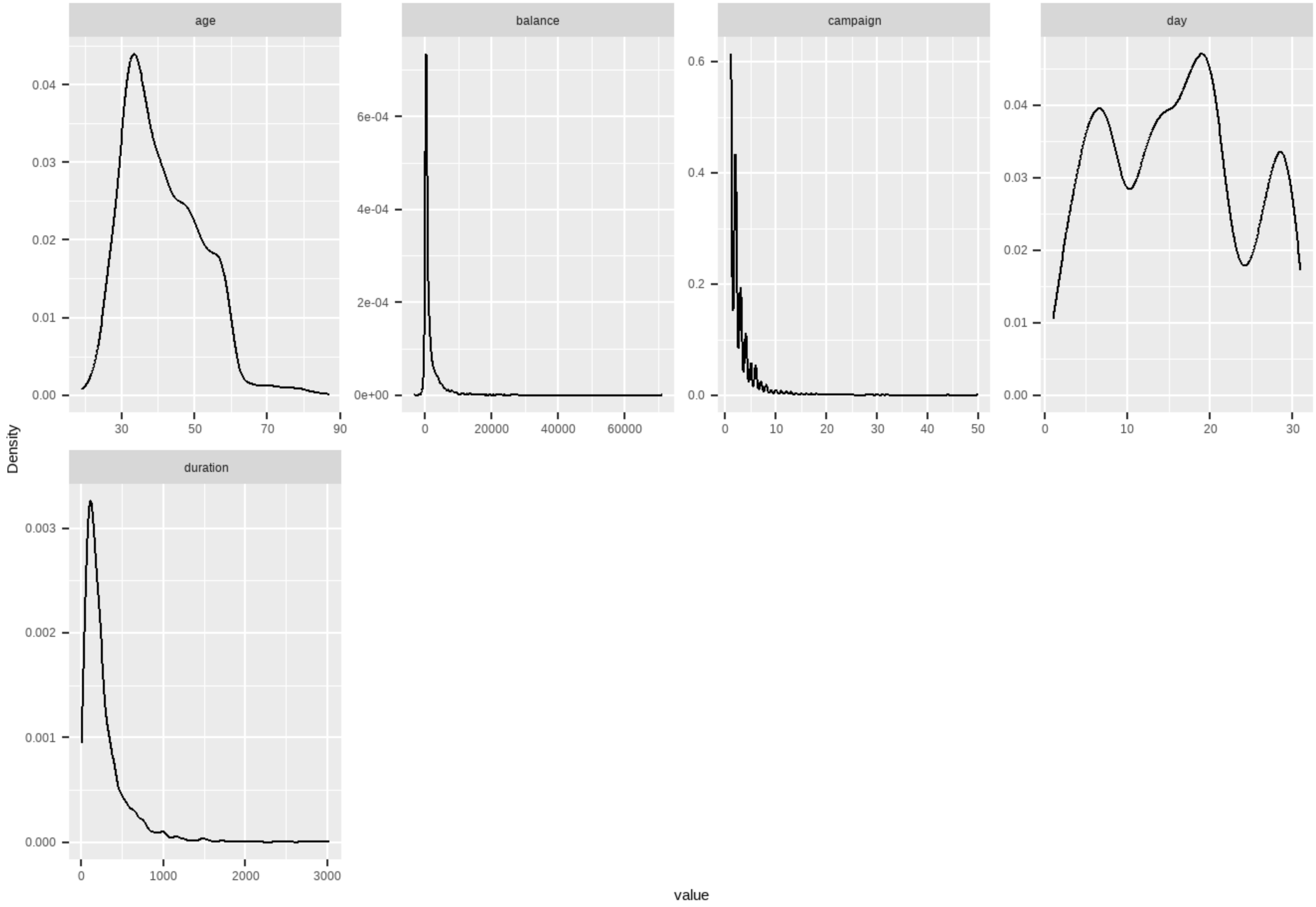
A subset from the original dataset was made just containing numerical variables.

Create Numeric Subset

```
# A tibble: 6 × 6
    age balance   day duration campaign y
  <dbl>   <dbl> <dbl>    <dbl>    <dbl> <fct>
1    30    1787    19       79        1 no
2    33    4789    11      220        1 no
3    35    1350    16      185        1 no
4    30    1476     3      199        4 no
5    59       0     5      226        1 no
6    35     747    23      141        2 no
```

Similar to the reasoning for creating a categorical subset, I also created a numeric subset to focus the exploratory analysis on identifying patterns and assessing significance within the numeric variables.

Distribution of Numeric Values



The distribution of the numeric variables are checked using a density plot. Based off the density plots above, balance, campaign, duration show extreme right skewness. The age variable is right skewed and the day variable appears to be multimodal showing several peaks rather than a single dominant variable.

Descriptive Statistics

Vname	Group	TN	nNeg	nZero	nPos	NegInf	PosInf	NA_Value	Per_of_Missing	su
age	All	4,521	0	0	4,521	0	0	0	0	186,13
balance	All	4,521	366	357	3,798	0	0	0	0	6,431,83
campaign	All	4,521	0	0	4,521	0	0	0	0	12,63
day	All	4,521	0	0	4,521	0	0	0	0	71,95
duration	All	4,521	0	0	4,521	0	0	0	0	1,193,36

The descriptive statistics table provides a deeper examination of the spread and shape of the numeric variables. One notable observation is that balance contains both negative and zero values. This is important to note as negative numbers can restrict certain data transformations. This table also confirms the extreme right-skewness seen in the density plots for balance, campaign, and duration. The skewness values include 6.59 for balance, 4.47 for campaign, and 2.77 for duration which indicate long right tails being pulled by potential outliers. These three variables all show kurtosis greater than 3, suggesting there to be extreme values. Although age and day exhibit slight right skewness, the degree of skewness does not pose any issue.

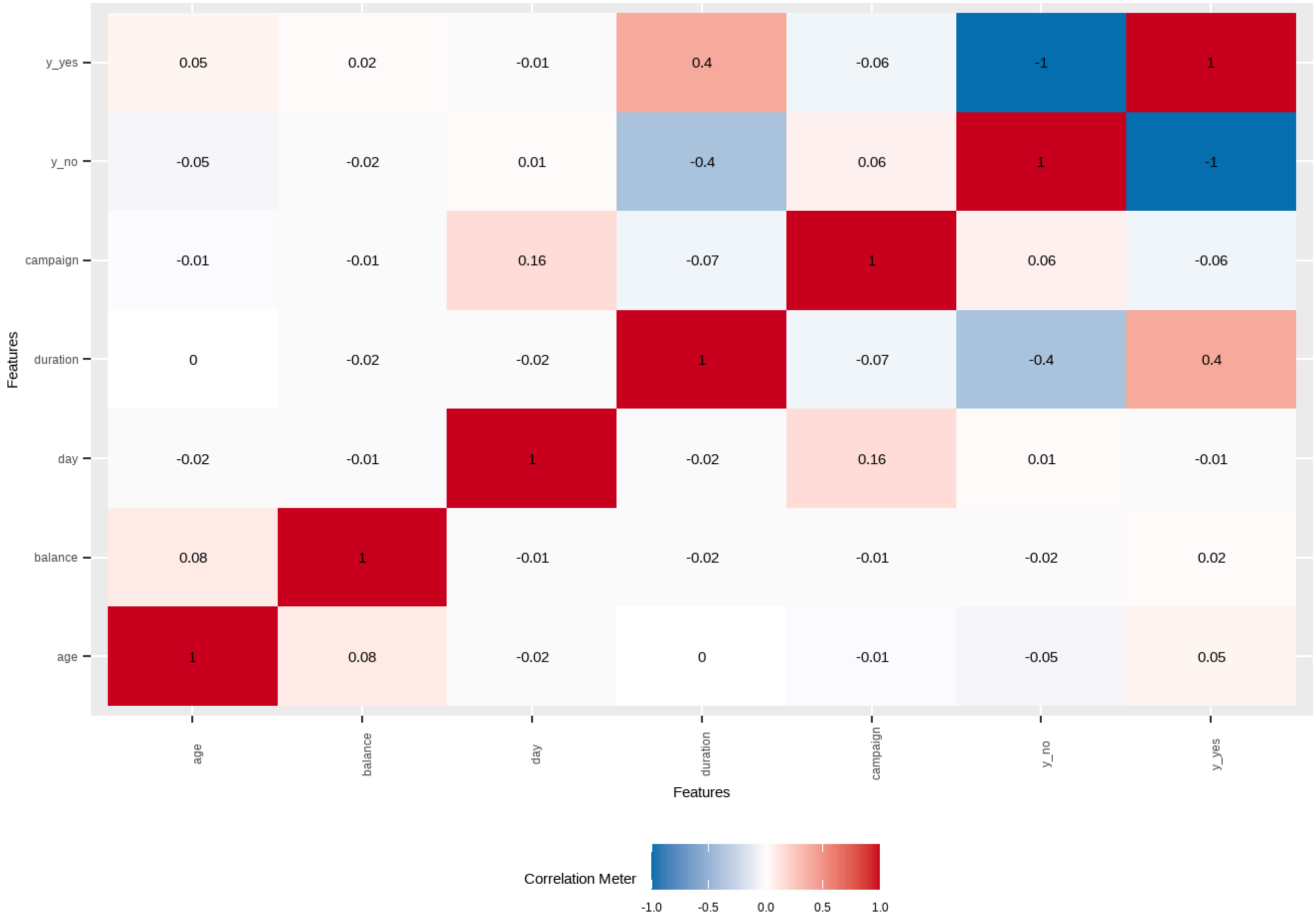
Diagnose Potential Outliers

```
# A tibble: 5 × 6
  variables outliers_cnt outliers_ratio outliers_mean with_mean without_mean
  <chr>            <int>          <dbl>         <dbl>     <dbl>        <dbl>
1 age                 38          0.841          78.6      41.2         40.9
2 balance            506         11.2          7587.    1423.         646.
3 day                  0          0             NaN       15.9         15.9
4 duration           330          7.30          970.     264.         208.
5 campaign           318          7.03           11.4       2.79         2.14
```

Comments on Diagnosing Outliers:
The three numeric variables that require further diagnosis are balance, duration, and campaign, as they contain a good amount of outliers. The descriptive table shows that these extreme values inflate the mean for each variable, indicating that the distribution is heavily influenced by outliers. However, outlier removal must be approached with caution. While these values are statistically extreme, they may still carry meaningful information in relation to the dependent variable. For example, removing all high balance values could eliminate an important relationship such as the possibility that clients with larger account balances are more likely to subscribe to a term deposit. To evaluate this properly boxplots will be generated to visually assess the outliers. If their are clusters of observations with a few reaching beyond it, those observations will be eliminated. This approach ensures that meaningful patterns are kept while minimizing distortion from set apart observations. This will be done in the Data Preparation part of the Report.

Age and day do not require outlier removal, as their potential outliers are minimal and do not meaningfully impact the overall mean or distribution.

Check Correlations



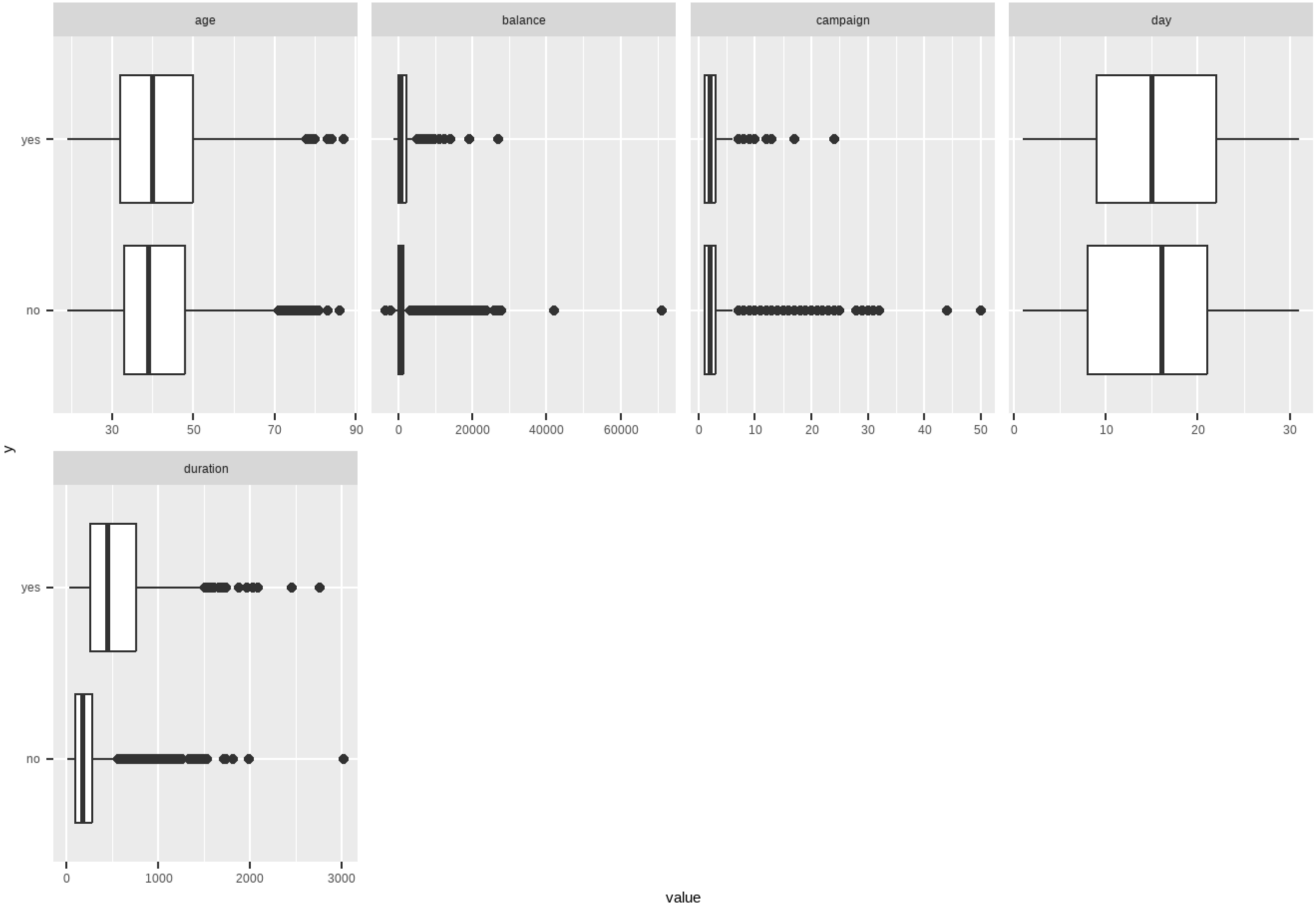
All numeric variables have weak correlation with the dependent variable except for duration which has moderately strong correlation. Although the numeric variables show low correlation to the dependent variable, it does not diminish their potential use in non-linear classification problems. This means that, depending on the modeling approach, certain variables may be directly useful in their current form, while others may require transformations to fully capture their predictive value.

Summary of Data Understanding

The Data Understanding phase revealed several important characteristics of the dataset that will guide the upcoming data preparation steps. The dependent variable exhibits class imbalance, with the majority of clients not subscribing to a term deposit, indicating that modeling techniques may need to address imbalance to avoid biased predictions. Categorical exploration showed uneven level distributions and identified strong associations between most categorical variables and subscription outcomes, with the exception of default, which offered little predictive value.
Numeric exploration highlighted significant skewness, heavy tails, and numerous outliers in variables such as balance, duration, and campaign, while age and day did not require cleaning. Correlation analysis confirmed that most numeric variables have weak linear relationships with the dependent variable, suggesting that nonlinear models may be more effective and that transformations could be beneficial for linear methods. Collectively, these findings emphasize the need for thoughtful preprocessing—such as handling outliers, addressing class imbalance, and possibly preparing numeric transformations to ensure that the subsequent modeling phase is both accurate and reliable.

Data Preparation

Look Into Outliers



Two boxplots were graphed for each numeric variable, one boxplot representing one class of the dependent variable, the other boxplot representing the second class of the dependent variable.
For the balance variable it was clear that two numbers extended beyond the normal cluster of outliers. These two balance values were 42,045 and 71,188.
For the campaign variable it was also clear that two campaign numbers extended beyond the normal cluster of outliers. The two campaing values were 44 and 50.
For the duration variable three values extended beyond the normal cluster of outliers. These campaign values were 2456, 2769.
These values will be removed to ensure that the models are not disproportionately influenced by values that do not represent typical customer behavior and to also help with underlying distribution.

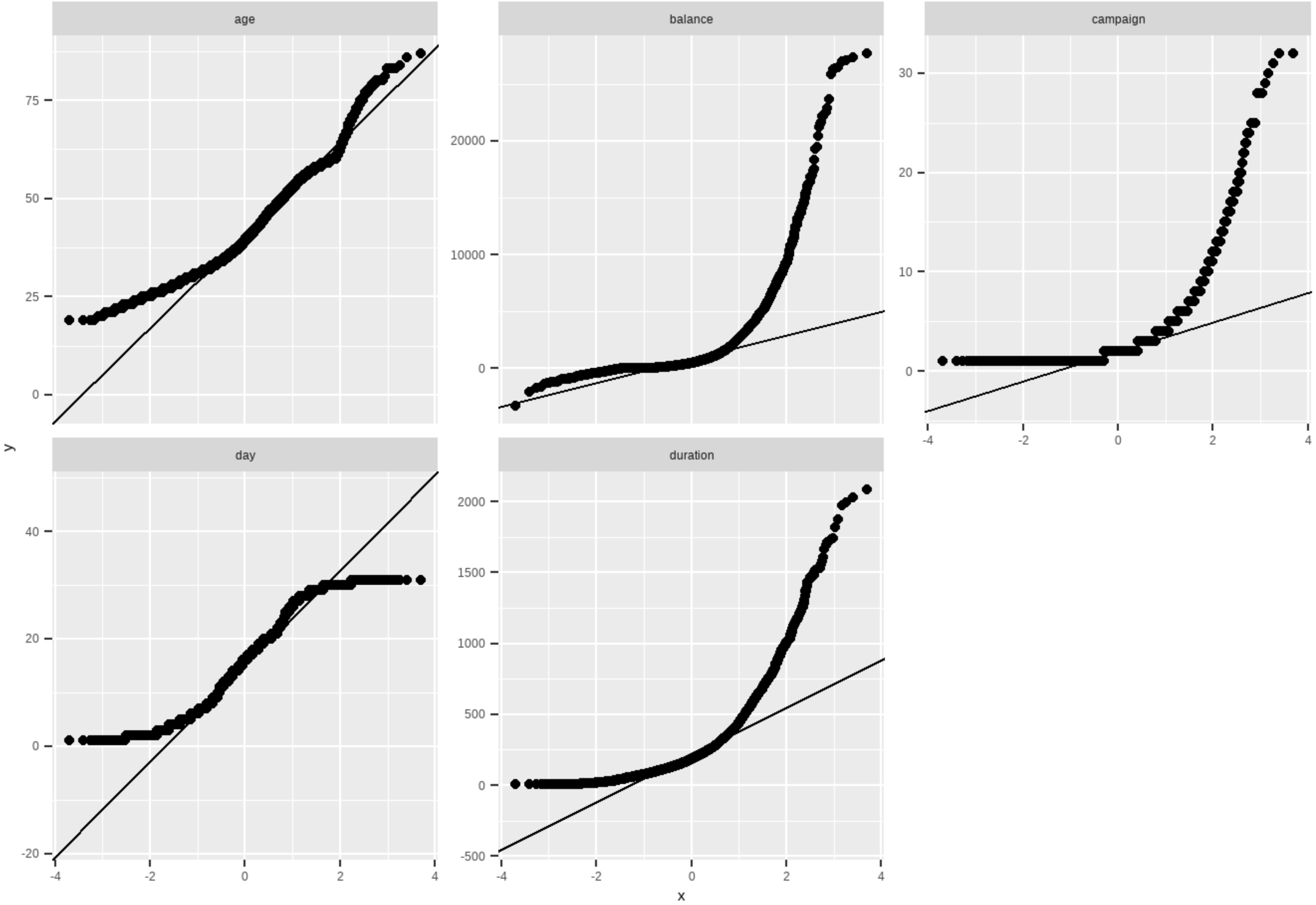
Remove Outliers

Revisit Skewness

```
  Variable Skewness
1  Balance 4.295765
2 Duration 2.408800
3 Campaign 3.993293
```

Removing the identified extreme outliers helped reduce the right skewness in the balance, duration, and campaign variables. Balance skewness decreased from 6.59 to 4.29, duration skewness decreased from 2.77 to 2.41, and campaign skewness decreased from 4.74 to 3.99. This demonstrates that removing only a small number of extreme observations was effective in reducing skewness while preserving the underlying patterns and overall structure of the data.

Plot Normality, Q-Q Plots



The Q-Q Plots shown above visualize the distribution of the numeric variables. All variables deviate form normality, some greater than others. Based off the models I will be using, normality can pose an issue. The models that will be used in this report are classification trees, KNN, and Logistic Regression.

Classification trees are not affected by normality and have no normality assumptions, so the normality of variables will not affect model performance.

KNN also is not affected by normality as it has no normality assumption, so normality of variables will not affect model performance.

Logistic Regression predictors do not need to be normal but the residuals need to be roughly symmetrical, so data transformations may be needed.

Check Data Types - Classification Tree

```
tibble [4,514 × 17] (S3: tbl_df/tbl/data.frame)
 $ age         : num [1:4514] 30 33 35 30 59 35 36 39 41 43 ...
 $ job         : Factor w/ 12 levels "admin.","blue-collar",..: 11 8 5 5 2 5 7 10 3 8 ...
 $ marital     : Factor w/ 3 levels "divorced","married",..: 2 2 3 2 2 3 2 2 2 2 ...
 $ education   : Factor w/ 4 levels "primary","secondary",..: 1 2 3 3 2 3 3 2 3 1 ...
 $ default     : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ balance     : num [1:4514] 1787 4789 1350 1476 0 ...
 $ housing     : Factor w/ 2 levels "no","yes": 1 2 2 2 2 1 2 2 2 2 ...
 $ loan        : Factor w/ 2 levels "no","yes": 1 2 1 2 1 1 1 1 1 2 ...
 $ contact     : Factor w/ 3 levels "cellular","telephone",..: 1 1 1 3 3 1 1 1 3 1 ...
 $ day         : num [1:4514] 19 11 16 3 5 23 14 6 14 17 ...
 $ month       : Factor w/ 12 levels "apr","aug","dec",..: 11 9 1 7 9 4 9 9 9 1 ...
 $ duration    : num [1:4514] 79 220 185 199 226 141 341 151 57 313 ...
 $ campaign    : num [1:4514] 1 1 1 4 1 2 1 2 2 1 ...
 $ poutcome    : Factor w/ 4 levels "failure","other",..: 4 1 1 4 4 1 2 4 4 1 ...
 $ y           : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ pdays_bin   : Factor w/ 4 levels "0 - 150 Days",..: 4 3 3 4 4 2 3 4 4 1 ...
 $ previous_bin: Factor w/ 3 levels "High Contact",..: 3 2 2 3 3 2 2 3 3 2 ...
```

The Data Types are correct and ready to model with for classification trees. All the categorical variables are factors and the numerical variables are numeric. The dependent variable “y” is a factor and is in a 1,0 binary format. 1 represents yes and 0 represents no.

Partition Data - Classification Tree

```
Distribution of DepVar-Trainset



        0         1 
0.8848466 0.1151534 
```

```
 Distribution of DepVar-Testset



        0         1 
0.8854398 0.1145602 
```

This dataset is partitioned into a 70/30 split so the data can be trained then tested. A set.seed of 1 was also given for reproducibility of the model results. The proportion of the dependent variable was very close between the trainset and testset as well indicating the data split preserved the original class distribution.

To address the dependent variable class imbalance, a weighted classification tree will be used to mitigate the disproportionate representation of the dependent variable. However, for the categorical predictors, there was no practical or meaningful way to bin or consolidate individual levels without distorting the underlying information. As a result, the level imbalance will simply be acknowledged and carried forward into the classification tree modeling, with the understanding that tree-based methods inherently downweight rare levels and that any remaining imbalance will be reflected in the natural structure of the data.

Convert Data for Modeling - KNN

Create dataset for KNN modeling

```
[1] "default"         "housing"         "loan"            "y"
[5] "job.admin."      "job.blue.collar"
```

Because KNN is a distance-based algorithm, proper preprocessing is essential to ensure that all predictors contribute appropriately to the model. Categorical variables must first be converted into numerical form through one-hot encoding so that the algorithm can compute distances between observations. Numeric variables require scaling because KNN is sensitive to differences in scale. Binary yes/no variables are also converted to 0/1 to maintain consistent numerical representation across the dataset. Finally, column names are standardized to avoid issues. These preprocessing steps allowed for the creating of an appropriate KNN dataset that can be used to model.

Eliminate Redundant Variables

```
Strongly Correlated Predictors:

 [1] "housing"                  "loan"
 [3] "y"                        "job.blue.collar"
 [5] "job.retired"              "marital.married"
 [7] "education.tertiary"       "contact.cellular"
 [9] "contact.unknown"          "month.apr"
[11] "month.dec"                "month.mar"
[13] "month.may"                "month.oct"
[15] "month.sep"                "poutcome.other"
[17] "poutcome.success"         "poutcome.unknown"
[19] "pdays_bin.0...150.Days"   "pdays_bin.151...300.Days"
[21] "pdays_bin.Not.Contacted"  "previous_bin.High.Contact"
[23] "previous_bin.Low.Contact" "previous_bin.Not.Contacted"
[25] "day.1"                    "day.10"
[27] "scaled_duration"          "scaled_campaign"
```

```
'data.frame':   4514 obs. of  28 variables:
 $ housing                   : num  0 1 1 1 1 0 1 1 1 1 ...
 $ loan                      : num  0 1 0 1 0 0 0 0 0 1 ...
 $ y                         : num  1 1 1 1 1 1 1 1 1 1 ...
 $ job.blue.collar           : num  0 0 0 0 1 0 0 0 0 0 ...
 $ job.retired               : num  0 0 0 0 0 0 0 0 0 0 ...
 $ marital.married           : num  1 1 0 1 1 0 1 1 1 1 ...
 $ education.tertiary        : num  0 0 1 1 0 1 1 0 1 0 ...
 $ contact.cellular          : num  1 1 1 0 0 1 1 1 0 1 ...
 $ contact.unknown           : num  0 0 0 1 1 0 0 0 1 0 ...
 $ month.apr                 : num  0 0 1 0 0 0 0 0 0 1 ...
 $ month.dec                 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ month.mar                 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ month.may                 : num  0 1 0 0 1 0 1 1 1 0 ...
 $ month.oct                 : num  1 0 0 0 0 0 0 0 0 0 ...
 $ month.sep                 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome.other            : num  0 0 0 0 0 0 1 0 0 0 ...
 $ poutcome.success          : num  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome.unknown          : num  1 0 0 1 1 0 0 1 1 0 ...
 $ pdays_bin.0...150.Days    : num  0 0 0 0 0 0 0 0 0 1 ...
 $ pdays_bin.151...300.Days  : num  0 0 0 0 0 1 0 0 0 0 ...
 $ pdays_bin.Not.Contacted   : num  1 0 0 1 1 0 0 1 1 0 ...
 $ previous_bin.High.Contact : num  0 0 0 0 0 0 0 0 0 0 ...
 $ previous_bin.Low.Contact  : num  0 1 1 0 0 1 1 0 0 1 ...
 $ previous_bin.Not.Contacted: num  1 0 0 1 1 0 0 1 1 0 ...
 $ day.1                     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ day.10                    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ scaled_duration           : num [1:4514, 1] -0.728 -0.168 -0.307 -0.252 -0.145 ...
  ..- attr(*, "scaled:center")= num 262
  ..- attr(*, "scaled:scale")= num 252
 $ scaled_campaign           : num [1:4514, 1] -0.598 -0.598 -0.598 0.413 -0.598 ...
  ..- attr(*, "scaled:center")= num 2.77
  ..- attr(*, "scaled:scale")= num 2.97
```

```
 [1] "housing"                    "loan"
 [3] "job.retired"                "marital.married"
 [5] "education.tertiary"         "contact.cellular"
 [7] "contact.unknown"            "month.apr"
 [9] "month.mar"                  "month.may"
[11] "month.oct"                  "month.sep"
[13] "poutcome.success"           "poutcome.unknown"
[15] "pdays_bin.0...150.Days"     "pdays_bin.151...300.Days"
[17] "pdays_bin.Not.Contacted"    "previous_bin.Low.Contact"
[19] "previous_bin.Not.Contacted" "day.1"
[21] "day.10"                     "scaled_duration"
[23] "scaled_campaign"            "y"
```

```
'data.frame':   4514 obs. of  24 variables:
 $ housing                   : num  0 1 1 1 1 0 1 1 1 1 ...
 $ loan                      : num  0 1 0 1 0 0 0 0 0 1 ...
 $ job.retired               : num  0 0 0 0 0 0 0 0 0 0 ...
 $ marital.married           : num  1 1 0 1 1 0 1 1 1 1 ...
 $ education.tertiary        : num  0 0 1 1 0 1 1 0 1 0 ...
 $ contact.cellular          : num  1 1 1 0 0 1 1 1 0 1 ...
 $ contact.unknown           : num  0 0 0 1 1 0 0 0 1 0 ...
 $ month.apr                 : num  0 0 1 0 0 0 0 0 0 1 ...
 $ month.mar                 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ month.may                 : num  0 1 0 0 1 0 1 1 1 0 ...
 $ month.oct                 : num  1 0 0 0 0 0 0 0 0 0 ...
 $ month.sep                 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome.success          : num  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome.unknown          : num  1 0 0 1 1 0 0 1 1 0 ...
 $ pdays_bin.0...150.Days    : num  0 0 0 0 0 0 0 0 0 1 ...
 $ pdays_bin.151...300.Days  : num  0 0 0 0 0 1 0 0 0 0 ...
 $ pdays_bin.Not.Contacted   : num  1 0 0 1 1 0 0 1 1 0 ...
 $ previous_bin.Low.Contact  : num  0 1 1 0 0 1 1 0 0 1 ...
 $ previous_bin.Not.Contacted: num  1 0 0 1 1 0 0 1 1 0 ...
 $ day.1                     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ day.10                    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ scaled_duration           : num [1:4514, 1] -0.728 -0.168 -0.307 -0.252 -0.145 ...
  ..- attr(*, "scaled:center")= num 262
  ..- attr(*, "scaled:scale")= num 252
 $ scaled_campaign           : num [1:4514, 1] -0.598 -0.598 -0.598 0.413 -0.598 ...
  ..- attr(*, "scaled:center")= num 2.77
  ..- attr(*, "scaled:scale")= num 2.97
 $ y                         : num  1 1 1 1 1 1 1 1 1 1 ...
                                    1          2 MeanDecreaseAccuracy
scaled_duration            46.0166138 60.9279989            60.856120
month.oct                  19.2578444 20.9889277            24.774599
contact.unknown            15.0134421  7.1329882            17.785824
month.apr                  13.6933407  3.7046106            13.777861
contact.cellular           13.6617780  1.6291981            15.164522
previous_bin.Low.Contact   11.2578379  0.2591165            11.307956
month.may                  10.8978293 14.0741552            16.874360
housing                    10.8639973  8.4275388            14.909353
pdays_bin.Not.Contacted     9.1738781  8.2650520            10.635235
month.mar                   8.3335099 16.2077767            14.456470
poutcome.unknown            8.1465752  9.0920055            10.530322
pdays_bin.151...300.Days    7.6271684  2.3955096             8.625869
education.tertiary          7.5263937  2.9763082             8.004400
previous_bin.Not.Contacted  7.4088212  8.4011612             9.560529
poutcome.success            5.4645041 44.4867124            28.286852
day.10                      5.3064619  8.6416527             8.756875
scaled_campaign             5.2353564 -0.1956031             3.892703
pdays_bin.0...150.Days      4.5438931  2.7720388             6.477212
job.retired                 4.2838278  9.6591142             9.246889
month.sep                   3.1851255 -1.6062974             1.927835
loan                        2.8562296  4.8510648             5.294345
marital.married             1.1093575  5.0714786             4.277174
day.1                      -0.5715345  5.5573073             2.101030
                           MeanDecreaseGini
scaled_duration                  215.323200
month.oct                         17.328798
contact.unknown                    9.034582
month.apr                         11.033848
```

```
contact.cellular                 9.999465
previous_bin.Low.Contact         7.221207
month.may                       11.460049
housing                         16.092902
pdays_bin.Not.Contacted          5.553144
month.mar                        9.799981
poutcome.unknown                 5.644474
pdays_bin.151...300.Days         5.725283
education.tertiary              13.486597
previous_bin.Not.Contacted       5.876688
poutcome.success                43.216056
day.10                           5.203309
scaled_campaign                 28.703231
pdays_bin.0...150.Days           9.076399
job.retired                     10.495790
month.sep                        5.093467
loan                             8.611733
marital.married                 13.215170
day.1                            3.956594
```

Because KNN is a distance-based algorithm, redundant or irrelevant predictors can introduce noise and significantly degrade model performance. To reduce redundancy and improve predictive efficiency, I applied two feature-selection steps: correlation filtering and Random Forest variable importance. Through this process, the dataset was reduced from 84 variables to 23, yielding a cleaner and more informative feature set for KNN modeling.

Correlation filtering was used to identify predictors that had little to no association with the dependent variable. Variables with very weak correlation (|correlation coefficient| < 0.05) were considered uninformative, as they contribute minimal predictive power and act as noise in a distance-based model.

To further refine the feature set, a Random Forest model was used to evaluate variable importance. The Mean Decrease Accuracy metric identifies variables that meaningfully improve model accuracy, while low values indicate predictors that introduce noise or contribute very little.

Variables with a Mean Decrease Accuracy < 2 were removed, which refined the dataset from 29 variables down to 23 final predictors. These remaining variables represent meaningful contributors to model performance and form the basis of the KNN modeling dataset.

This two-step filtering process of correlation followed by Random Forest importance ensures that the final dataset used for KNN is both compact and highly informative, minimizing noise and maximizing model accuracy.

Data Partition - KNN

```
        0         1 
0.8848283 0.1151717 
```

```
        0         1 
0.8853186 0.1146814 
```

Comments on KNN Data Partitioning:
Before modeling the dataset is partitioned into a 60/40 split so the data can be trained then tested. A set.seed of 1 was also given for reproducibility of the model results. The dependent variable for both train and test set was coded into a binary so no = 0 and yes = 1. The proportion of the dependent variable was very close between the trainset and testset as well.

Data Partition - Logistic Regression

```
Depvar Proportion train set
```

```
        0         1 
0.8848539 0.1151461 
```

```
Depvar Proportion test set
```

```
        0         1 
0.8855368 0.1144632 
```

Before modeling the dataset is partitioned into a 75/25 split so the data can be trained then tested. A set.seed of 1 was also given for reproducibility of the model results. The proportion of the dependent variable was very close between the trainset and testset as well.

To address class imbalance seen in the dependent variable, a weighted logistic regression model will be used in the modeling phase to help mitigate that.

Summary of Data Preparation

The data preparation phase focused on ensuring that each modeling technique received an appropriately structured and optimized dataset. For logistic regression and classification trees, minimal preprocessing was required beyond the initial cleaning from the Data Understanding stage. To address the substantial class imbalance in the dependent variable, weighted modeling was planned for the classification tree and logistic regression. Categorical level imbalance was evaluated, but no meaningful grouping or binning could be applied without distorting the underlying information, so the imbalance was simply noted for interpretation. In contrast, KNN required extensive preprocessing due to its sensitivity to scale. Categorical variables were one-hot encoded, numeric predictors were standardized, and binary variables were converted to numeric format. Feature selection was performed through correlation filtering and Random Forest variable importance, reducing the predictor set from 84 variables to 23. Together, these preparation steps ensured that each model logistic regression, classification tree, and KNN, could be trained on data that was properly structured for its specific methodological requirements.

Modeling - Classification Tree

Non-Weighted Best Tree

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1149  108
         1   49   47

               Accuracy : 0.884
                 95% CI : (0.8657, 0.9005)
    No Information Rate : 0.8854
    P-Value [Acc > NIR] : 0.5886

                  Kappa : 0.3144

 Mcnemar's Test P-Value : 3.676e-06

            Sensitivity : 0.30323
            Specificity : 0.95910
         Pos Pred Value : 0.48958
         Neg Pred Value : 0.91408
             Prevalence : 0.11456
         Detection Rate : 0.03474
   Detection Prevalence : 0.07095
      Balanced Accuracy : 0.63116

       'Positive' Class : 1
```

```
F1 Score:  0.375
```

This model struggles due to imbalance in the dependent variable. This can be seen in the high specificity but low sensitivity score. The results of this model is to give a baseline for the weighted model.

Weighted Best Tree

Create Weights

```
Class Counts (n) for Admitted from the Training Dataset


Class Total: 3161
```

To address the imbalance in the dependent variable, class weights were created based on the inverse frequency of each class in the training dataset. This weighting scheme ensures that the minority class receives proportionally greater influence during model training, preventing the tree from being biased toward the majority class.

Model Weighted Tree

Test Weighted Tree

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 833  22
         1 365 133
                                          
               Accuracy : 0.714           
                 95% CI : (0.6891, 0.7379)
    No Information Rate : 0.8854          
    P-Value [Acc > NIR] : 1               
                                          
                  Kappa : 0.2819          
                                          
 Mcnemar's Test P-Value : <2e-16          
                                          
            Sensitivity : 0.8581          
            Specificity : 0.6953          
         Pos Pred Value : 0.2671          
         Neg Pred Value : 0.9743          
             Prevalence : 0.1146          
         Detection Rate : 0.0983          
   Detection Prevalence : 0.3681          
      Balanced Accuracy : 0.7767          
                                          
       'Positive' Class : 1               
                                          
```

```
F1 Score:  0.4073507
```

```
[1] "failure" "other"   "success" "unknown"
```

```
[1] "cellular"  "telephone" "unknown"
```

Although the weighted classification tree has a slightly lower overall accuracy than the unweighted tree, the balanced accuracy is substantially higher, and the sensitivity and specificity are much more comparable. This indicates that the weighted model is far better at distinguishing between the two classes of the dependent variable rather than being biased toward predicting the majority class. The higher F1 score further supports this conclusion, as it reflects improved performance on the minority “yes” class by balancing precision and recall. Together, these metrics demonstrate that the weighted tree provides a more reliable model making it the superior classification tree despite a marginally lower raw accuracy.

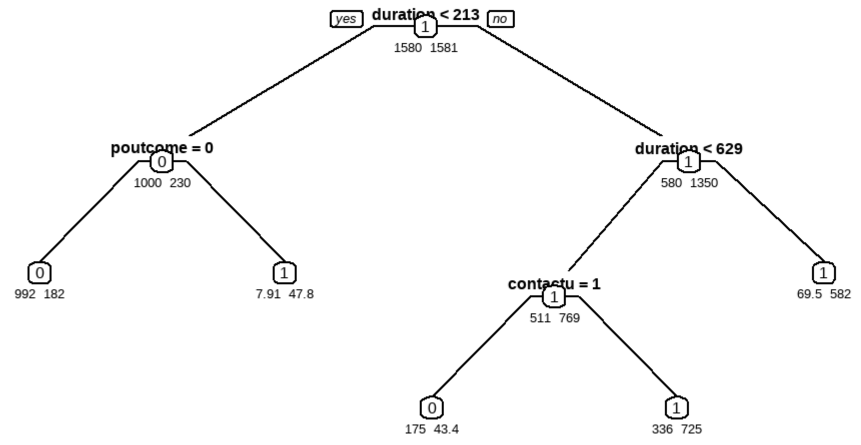
Variable Importance

```
Variable Importance

rpart variable importance

  only 20 most important variables shown (out of 47)

                          Overall
duration                  100.000
poutcomesuccess            52.861
contactunknown             52.154
poutcomeunknown            38.981
pdays_binNot Contacted     34.019
previous_binNot Contacted  12.383
age                         5.394
monthmar                    0.000
housingyes                  0.000
monthjul                    0.000
`pdays_binNot Contacted`    0.000
monthmay                    0.000
monthsep                    0.000
jobunemployed               0.000
maritalsingle               0.000
jobservices                 0.000
jobmanagement               0.000
monthoct                    0.000
jobstudent                  0.000
day                         0.000
```

The Variables that had the most importance in relation to the dependent variable for the weighted classification tree is duration, poutcomesuccess, and contactunknown. Variable importance signifies which predictors contribute the most to the model’s ability to correclty classify observatoins.

Print Weighted Tree

```
Tree Diagram with Counts
```



Best Tree Diagram Interpretation:
- Path 1: The left most path shows that calls less than 213 seconds and a failed campaign gives a very low chance for the client to subscribe.
- Path 2: The Right-Mid path shows that moderate call duration and telephone contact increases the likelihood of success. - Path 3: The Right most path shows that very long calls have a high probability of leading to a successful subscription.

Overall, the weighted classification tree demonstrated the strongest performance among the tree-based models and will be carried forward into the evaluation phase. It will be compared against the best-performing logistic regression and KNN models to determine which approach is most appropriate for addressing the business problem.

Modeling

Fit KNN - Default

```
k-Nearest Neighbors 

2709 samples
  23 predictor
   2 classes: '0', '1'
```

```
No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 2438, 2438, 2439, 2438, 2438, 2438, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   1  0.8645319  0.2993143
   2  0.8685923  0.2947420
   3  0.8874074  0.3343741
   4  0.8881509  0.3362722
   5  0.8907339  0.3280079
   6  0.8977463  0.3641726
   7  0.8936873  0.3195281
   8  0.8951620  0.3311455
   9  0.8936873  0.3164524
  10  0.8936887  0.3147053

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 6.
```

The chart above shows how many neighboring observations should be used when predicting the class of a new data point. The model evaluated values of k from 1 to 10, and accuracy was used to select the optimal number of neighbors. The highest accuracy occurred at k = 10, meaning that the model will classify each new observation based on the majority class among its 10 closest neighbors.

Predict KNN - Default

```
Default KNN Confusion Matrix

Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1547  145
         1   51   62

               Accuracy : 0.8914
                 95% CI : (0.8761, 0.9054)
    No Information Rate : 0.8853
    P-Value [Acc > NIR] : 0.2201

                  Kappa : 0.3335

 Mcnemar's Test P-Value : 3.077e-11

            Sensitivity : 0.29952
            Specificity : 0.96809
         Pos Pred Value : 0.54867
         Neg Pred Value : 0.91430
             Prevalence : 0.11468
         Detection Rate : 0.03435
   Detection Prevalence : 0.06260
      Balanced Accuracy : 0.63380

       'Positive' Class : 1

```

```
F1 Score:  0.388
```

Comments on Default KNN:
This model was made to be the baseline of the other KNN models which account for class imbalance in the dependent variable. The results highlight the model’s weak predictive power, as the positive class represents a minority of the data. The low sensitivity compared to the high specificity indicates that the model is biased toward predicting the majority class, performing well on non-subscribers but failing to accurately identify true subscribers. This demonstrates the impact of class imbalance on the model’s ability to generalize across both classes.

Fit KNN - Oversampling

```
k-Nearest Neighbors 

2709 samples
  23 predictor
   2 classes: '0', '1' 

No pre-processing
Resampling: Cross-Validated (10 fold) 
Summary of sample sizes: 2438, 2438, 2439, 2438, 2438, 2438, ... 
Addtional sampling using up-sampling

Resampling results across tuning parameters:

  k   Accuracy   Kappa    
   1  0.8645319  0.2993143
   2  0.8453410  0.3495339
   3  0.8235588  0.3620801
   4  0.8006765  0.3390980
   5  0.7855392  0.3379864
   6  0.7737310  0.3323858
   7  0.7567528  0.3103876
   8  0.7538007  0.3149414
   9  0.7508432  0.3056778
  10  0.7486292  0.3037790

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 1.
```

Oversampling is a technique used to address class imbalance by increasing the number of observations in the minority class. Since oversampling is being used the amount of neighbors needed to be recalculated to see which one lead to the highest accuracy. In this case the highest accuracy occured at k = 1, meaning that the model will classify each new observation based on the majoirty class among its closest neighbor.

Predict KNN - Oversampling

```
Oversampling KNN Confusion Matrix

Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1502  130
         1   96   77

               Accuracy : 0.8748
                 95% CI : (0.8586, 0.8897)
    No Information Rate : 0.8853
    P-Value [Acc > NIR] : 0.92376

                  Kappa : 0.3359

 Mcnemar's Test P-Value : 0.02815

            Sensitivity : 0.37198
            Specificity : 0.93992
         Pos Pred Value : 0.44509
         Neg Pred Value : 0.92034
             Prevalence : 0.11468
         Detection Rate : 0.04266
   Detection Prevalence : 0.09584
      Balanced Accuracy : 0.65595

       'Positive' Class : 1


F1 Score: 0.40526
```

This KNN model although better than the default KNN model still struggles with dependent variable class prediction. Threshold Tuning will now be used to see if a better model can be produced.

Fit KNN - Threshold Tuning

Optimal Threshold

```
OPTIMAL CUTOFF VALUE OF: 0.1547619
```

Threshold tuning is the process of adjusting the probability cutoff used to classify predictions in a binary classification model. The default cutoff value is 0.5, the optimal cutoff value for this KNN model is 0.04545.

Predict Threshold Tuning

```
KNN Threshold Tuning Confusion Matrix

Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1220   38
         1  378  169

               Accuracy : 0.7695
                 95% CI : (0.7494, 0.7888)
    No Information Rate : 0.8853
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3381

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.81643
            Specificity : 0.76345
         Pos Pred Value : 0.30896
         Neg Pred Value : 0.96979
             Prevalence : 0.11468
         Detection Rate : 0.09363
   Detection Prevalence : 0.30305
      Balanced Accuracy : 0.78994

       'Positive' Class : 1


F1 Score: 0.44828
```

This model is the best KNN model. The threshold-tuned KNN model shows the best overall predictive balance, with the highest balanced accuracy across all KNN models. Although class separation improves substantially, the model still slightly favors predicting non-subscribers over subscribers.

Overall, the threshold tuning KNN model demonstrated the strongest performance among the KNN models and will be carried forward into the evaluation phase. It will be compared against the best-performing logistic regression and KNN models to determine which approach is most appropriate for addressing the business problem.

Modeling - Logistic Regression

Default Logistic Regression

```
Call:
NULL

Coefficients: (2 not defined because of singularities)
                              Estimate Std. Error z value Pr(>|z|)    
(Intercept)                 -2.207e+00  7.424e-01  -2.973 0.002947 ** 
age                         -5.303e-03  8.310e-03  -0.638 0.523390    
`jobblue-collar`            -6.123e-01  2.918e-01  -2.098 0.035886 *  
jobentrepreneur             -6.320e-03  4.326e-01  -0.015 0.988343    
jobhousemaid                -1.319e-01  4.603e-01  -0.286 0.774497    
jobmanagement                1.621e-01  2.842e-01   0.570 0.568362    
jobretired                   5.787e-01  3.641e-01   1.590 0.111938    
`jobself-employed`          -1.202e-01  4.376e-01  -0.275 0.783548    
jobservices                 -1.230e-01  3.163e-01  -0.389 0.697324    
jobstudent                   4.670e-01  4.682e-01   0.997 0.318555    
jobtechnician               -1.363e-01  2.729e-01  -0.499 0.617444    
jobunemployed               -7.065e-01  5.227e-01  -1.351 0.176546    
jobunknown                   4.912e-01  6.723e-01   0.731 0.465042    
maritalmarried              -4.876e-01  2.009e-01  -2.428 0.015198 *  
maritalsingle               -3.231e-01  2.376e-01  -1.360 0.173856    
educationsecondary          -2.469e-01  2.292e-01  -1.077 0.281386    
educationtertiary           -1.665e-01  2.705e-01  -0.616 0.538060    
educationunknown            -7.100e-01  4.163e-01  -1.705 0.088133 .  
defaultyes                   6.467e-01  4.673e-01   1.384 0.166362    
balance                      4.653e-06  2.465e-05   0.189 0.850258    
housingyes                  -1.424e-01  1.630e-01  -0.873 0.382442    
loanyes                     -7.332e-01  2.427e-01  -3.021 0.002521 ** 
contacttelephone            -1.711e-02  2.706e-01  -0.063 0.949584    
contactunknown              -1.220e+00  2.629e-01  -4.640 3.48e-06 ***
day                          1.268e-02  9.393e-03   1.350 0.177079    
monthaug                    -2.469e-01  2.922e-01  -0.845 0.398096    
monthdec                     6.845e-02  7.764e-01   0.088 0.929744    
monthfeb                     1.173e-01  3.550e-01   0.330 0.741083    
monthjan                    -9.020e-01  4.429e-01  -2.037 0.041667 *  
monthjul                    -5.908e-01  2.921e-01  -2.022 0.043134 *  
monthjun                     4.449e-01  3.485e-01   1.277 0.201691    
monthmar                     1.542e+00  4.479e-01   3.442 0.000578 ***
monthmay                    -6.487e-01  2.815e-01  -2.304 0.021197 *  
monthnov                    -1.036e+00  3.295e-01  -3.143 0.001669 ** 
monthoct                     1.412e+00  3.779e-01   3.737 0.000186 ***
monthsep                     7.085e-01  4.919e-01   1.440 0.149753    
duration                     4.586e-03  2.497e-04  18.364  < 2e-16 ***
campaign                    -8.279e-02  3.196e-02  -2.590 0.009586 ** 
poutcomeother                6.422e-01  3.298e-01   1.947 0.051550 .  
poutcomesuccess              2.687e+00  3.311e-01   8.116 4.83e-16 ***
poutcomeunknown             -1.035e-01  4.354e-01  -0.238 0.812063    
`pdays_bin151 - 300 Days`   -2.204e-01  3.125e-01  -0.705 0.480526    
`pdays_bin300+ Days`        -3.084e-01  3.763e-01  -0.820 0.412497    
`pdays_binNot Contacted`            NA         NA      NA       NA    
```

```
`previous_binLow Contact`   -4.380e-02  3.758e-01  -0.117 0.907200    
`previous_binNot Contacted`         NA         NA      NA       NA    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2419.3  on 3386  degrees of freedom
Residual deviance: 1587.0  on 3343  degrees of freedom
AIC: 1675

Number of Fisher Scoring iterations: 6
```

```
   df      AIC
m1  4 3101.061
m2  3 3124.723
m3  4 2992.914
```

A logistic regression model was initially run using the full dataset with the goal of identifying and removing insignificant variables to reduce noise. During this process, the model produced NA coefficient estimates for pdays_binNot Contacted and previous_binNot Contacted. This occurred because these variables exhibited perfect multicollinearity, meaning they contained identical or near-identical information. Further investigation showed that pdays_binNot Contacted, previous_binNot Contacted, and poutcomeunknown were perfectly aligned, each indicating the same underlying condition: the client had not been contacted previously. To resolve this issue, three separate logistic regression models were made and run each using only one of the three predictors. The models were compared using AIC where lower values indicate better model fit. The variable poutcomeunknown produced the lowest AIC, making it the most informative of the three. Therefore, a revised dataset was made for future logistic regression modeling that excluded the redudant pdays_bin and previous_bin.

Data Partition - Logistic Regression (corrected)

```
Call:
NULL

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)    
(Intercept)        -2.532e+00  5.372e-01  -4.714 2.43e-06 ***
age                -4.077e-03  7.155e-03  -0.570 0.568798    
`jobblue-collar`   -4.098e-01  2.434e-01  -1.684 0.092199 .  
jobentrepreneur    -2.546e-01  3.834e-01  -0.664 0.506640    
jobhousemaid       -3.932e-01  4.199e-01  -0.936 0.349076    
jobmanagement      -7.809e-02  2.416e-01  -0.323 0.746505    
jobretired          6.173e-01  3.122e-01   1.977 0.048031 *  
`jobself-employed` -1.899e-01  3.551e-01  -0.535 0.592733    
jobservices        -1.449e-01  2.742e-01  -0.528 0.597171    
jobstudent          3.812e-01  3.765e-01   1.012 0.311345    
jobtechnician      -1.908e-01  2.312e-01  -0.825 0.409145    
jobunemployed      -5.273e-01  4.126e-01  -1.278 0.201217    
```

```
jobunknown          5.152e-01  5.868e-01   0.878 0.379960    
maritalmarried     -4.612e-01  1.750e-01  -2.635 0.008412 ** 
maritalsingle      -3.055e-01  2.049e-01  -1.491 0.135931    
educationsecondary  4.490e-02  2.021e-01   0.222 0.824155    
educationtertiary   2.855e-01  2.336e-01   1.222 0.221724    
educationunknown   -4.669e-01  3.580e-01  -1.304 0.192116    
defaultyes          5.685e-01  4.340e-01   1.310 0.190230    
balance             7.095e-06  2.092e-05   0.339 0.734449    
housingyes         -2.669e-01  1.382e-01  -1.932 0.053419 .  
loanyes            -6.311e-01  2.011e-01  -3.138 0.001700 ** 
contacttelephone   -8.565e-02  2.338e-01  -0.366 0.714107    
contactunknown     -1.433e+00  2.285e-01  -6.274 3.53e-10 ***
day                 1.660e-02  8.188e-03   2.027 0.042650 *  
monthaug           -3.032e-01  2.500e-01  -1.213 0.225190    
monthdec            9.990e-02  6.584e-01   0.152 0.879396    
monthfeb            2.033e-01  2.942e-01   0.691 0.489458    
monthjan           -1.023e+00  3.758e-01  -2.721 0.006511 ** 
monthjul           -7.515e-01  2.510e-01  -2.994 0.002754 ** 
monthjun            5.589e-01  3.017e-01   1.852 0.063991 .  
monthmar            1.516e+00  3.906e-01   3.882 0.000104 ***
monthmay           -4.886e-01  2.344e-01  -2.084 0.037126 *  
monthnov           -8.616e-01  2.734e-01  -3.152 0.001624 ** 
monthoct            1.390e+00  3.303e-01   4.210 2.56e-05 ***
monthsep            6.711e-01  4.123e-01   1.628 0.103626    
duration            4.330e-03  2.046e-04  21.161  < 2e-16 ***
campaign           -7.170e-02  2.842e-02  -2.523 0.011650 *  
poutcomeother       4.871e-01  2.695e-01   1.808 0.070659 .  
poutcomesuccess     2.444e+00  2.719e-01   8.991  < 2e-16 ***
poutcomeunknown    -7.956e-02  1.876e-01  -0.424 0.671580    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3221.1  on 4513  degrees of freedom
Residual deviance: 2155.8  on 4473  degrees of freedom
AIC: 2237.8

Number of Fisher Scoring iterations: 6
```

After removing the variables affected by multicollinearity, a full logistic regression model was run to identify predictors that were statistically insignificant in explaining the likelihood of subscription. Based on the model output, age, education, default, and balance were the only variables whose coefficients were not statistically significant at the chosen significance level. Because these variables did not meaningfully contribute to predicting the dependent variable, they were removed from the dataset. This resulted in a refined set of predictors consisting only of variables with demonstrated statistical significance, which was then used to develop the final unweighted logistic regression model.

```
Depvar Proportion Trainset
```

```
        0         1 
0.8848539 0.1151461 
```

```
 Depvar Proportion Testset
```

```
        0         1 
0.8855368 0.1144632 
```

The refined dataset was then partioned into a 75/25 split so the data can be trained and then tested. A set.seed of 1 was also given for reproducibility of the model results. The proportion of the dependent variable was very close between the trainset and testset as well.

Modeling - Refined Logistic Regression

```
Call:
NULL

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)    
(Intercept)        -2.9103674  0.4329322  -6.722 1.79e-11 ***
`jobblue-collar`   -0.5024290  0.2779467  -1.808 0.070662 .  
jobentrepreneur     0.0707059  0.4238022   0.167 0.867498    
jobhousemaid        0.0013748  0.4387535   0.003 0.997500    
jobmanagement       0.2437116  0.2483223   0.981 0.326379    
jobretired          0.5844365  0.3151444   1.855 0.063667 .  
`jobself-employed` -0.0447069  0.4288057  -0.104 0.916964    
jobservices        -0.0961054  0.3151655  -0.305 0.760415    
jobstudent          0.4848857  0.4530878   1.070 0.284538    
jobtechnician      -0.0845534  0.2698110  -0.313 0.753992    
jobunemployed      -0.6407254  0.5206592  -1.231 0.218471    
jobunknown          0.3623811  0.6569042   0.552 0.581188    
maritalmarried     -0.4874347  0.1992976  -2.446 0.014455 *  
maritalsingle      -0.2689187  0.2227439  -1.207 0.227317    
housingyes         -0.1382698  0.1610467  -0.859 0.390578    
loanyes            -0.7144590  0.2411101  -2.963 0.003045 ** 
contacttelephone   -0.0351389  0.2663543  -0.132 0.895043    
contactunknown     -1.2242754  0.2616660  -4.679 2.89e-06 ***
day                 0.0128567  0.0093329   1.378 0.168338    
monthaug           -0.2377580  0.2889528  -0.823 0.410607    
monthdec           -0.0321417  0.7731677  -0.042 0.966840    
monthfeb            0.1334588  0.3489297   0.382 0.702105    
monthjan           -0.9000361  0.4372136  -2.059 0.039535 *  
monthjul           -0.5822839  0.2887471  -2.017 0.043739 *  
monthjun            0.4580950  0.3461638   1.323 0.185720    
monthmar            1.5230318  0.4445658   3.426 0.000613 ***
monthmay           -0.6643918  0.2786124  -2.385 0.017096 *  
```

```
monthnov           -1.0155872  0.3262414  -3.113 0.001852 ** 
monthoct            1.3947765  0.3751871   3.718 0.000201 ***
monthsep            0.7058077  0.4846919   1.456 0.145338    
duration            0.0045532  0.0002479  18.367  < 2e-16 ***
campaign           -0.0843857  0.0317644  -2.657 0.007893 ** 
poutcomeother       0.6880945  0.3263343   2.109 0.034983 *  
poutcomesuccess     2.7339129  0.3235602   8.449  < 2e-16 ***
poutcomeunknown     0.1350526  0.2344697   0.576 0.564621    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2419.3  on 3386  degrees of freedom
Residual deviance: 1593.4  on 3352  degrees of freedom
AIC: 1663.4

Number of Fisher Scoring iterations: 6
```

An unweighted logistic regression model was run on the refined dataset.

```
  `jobblue-collar`    jobentrepreneur       jobhousemaid      jobmanagement 
          2.117792           1.316896           1.306916           2.705895 
        jobretired `jobself-employed`        jobservices         jobstudent 
          1.849610           1.295271           1.679231           1.364484 
     jobtechnician      jobunemployed         jobunknown     maritalmarried 
          2.219768           1.211429           1.121338           2.210878 
     maritalsingle         housingyes            loanyes   contacttelephone 
          2.327756           1.452054           1.065117           1.094757 
    contactunknown                day           monthaug           monthdec 
          1.962564           1.349122           2.595426           1.099631 
          monthfeb           monthjan           monthjul           monthjun 
          1.771036           1.366313           2.247946           2.671392 
          monthmar           monthmay           monthnov           monthoct 
          1.354986           2.730458           1.659886           1.478822 
          monthsep           duration           campaign      poutcomeother 
          1.292989           1.151322           1.155094           1.606395 
   poutcomesuccess    poutcomeunknown 
          1.712698           2.363231 
```

Predictors were again checked for multicollinearity and there was none.

```
                        Overall
`jobblue-collar`    1.807644788
jobentrepreneur     0.166836974
jobhousemaid        0.003133341
jobmanagement       0.981432620
jobretired          1.854503845
`jobself-employed`  0.104259172
jobservices         0.304936312
jobstudent          1.070180402
jobtechnician       0.313380149
```

```
jobunemployed       1.230604205
jobunknown          0.551649750
maritalmarried      2.445762916
maritalsingle       1.207299929
housingyes          0.858569554
loanyes             2.963206895
contacttelephone    0.131925238
contactunknown      4.678772335
day                 1.377562823
monthaug            0.822826547
monthdec            0.041571500
monthfeb            0.382480531
monthjan            2.058573026
monthjul            2.016587835
monthjun            1.323347703
monthmar            3.425885813
monthmay            2.384645187
monthnov            3.112992929
monthoct            3.717548869
monthsep            1.456198612
duration           18.366851598
campaign            2.656615873
poutcomeother       2.108557359
poutcomesuccess     8.449473380
poutcomeunknown     0.575991926
```

Variable Importance was also checked from the logistic regression model. Duration, poutcomesuccess, and contactunkown had the highest variable importance.

Predict - Refined Logistic Regression

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 964  86
         1  34  43

               Accuracy : 0.8935
                 95% CI : (0.874, 0.9109)
    No Information Rate : 0.8855
    P-Value [Acc > NIR] : 0.2145

                  Kappa : 0.363

 Mcnemar's Test P-Value : 3.23e-06

            Sensitivity : 0.33333
            Specificity : 0.96593
         Pos Pred Value : 0.55844
         Neg Pred Value : 0.91810
             Prevalence : 0.11446
         Detection Rate : 0.03815
   Detection Prevalence : 0.06832
      Balanced Accuracy : 0.64963

       'Positive' Class : 1

```

```
F1 Score:  0.4174757
```

Comments on Refined Logistic Regression Model:
This model was made to be the baseline of the the upcoming weighted logistic Regression model which accounts for class imbalance in the dependent variable. The results highlight the model’s weak predictive power, as the positive class represents a minority of the data. The low sensitivity compared to the high specificity indicates that the model is biased toward predicting the majority class, performing well on non-subscribers but failing to accurately identify true subscribers. This demonstrates the impact of class imbalance on the model’s ability to generalize across both classes.

Modeling - Weighted Logistic Regression

Create Weights

To address the imbalance in the dependent variable, class weights were created based on the inverse frequency of each class in the training dataset. This weighting scheme ensures that the minority class receives proportionally greater influence during model training, preventing the logistic regression model from being biased toward the majority class.

Create Model

```

Call:
NULL

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)    
(Intercept)        -0.9364028  0.3075841  -3.044 0.002332 ** 
`jobblue-collar`   -0.8139609  0.1968765  -4.134 3.56e-05 ***
jobentrepreneur    -0.1329657  0.2967203  -0.448 0.654068    
jobhousemaid       -0.1110855  0.3158932  -0.352 0.725097    
jobmanagement      -0.0482501  0.1795400  -0.269 0.788127    
jobretired          0.5240346  0.2355661   2.225 0.026110 *  
`jobself-employed` -0.2367487  0.2972396  -0.796 0.425747    
jobservices        -0.3901042  0.2241786  -1.740 0.081833 .  
jobstudent          0.5955697  0.3427283   1.738 0.082258 .  
jobtechnician      -0.3052132  0.1906724  -1.601 0.109439    
jobunemployed      -0.8912270  0.3709313  -2.403 0.016276 *  
```

```
jobunknown          0.1475257  0.4934108   0.299 0.764946    
maritalmarried     -0.2944538  0.1501482  -1.961 0.049869 *  
maritalsingle       0.0424233  0.1662591   0.255 0.798596    
housingyes         -0.1604057  0.1123501  -1.428 0.153369    
loanyes            -0.9067313  0.1617079  -5.607 2.06e-08 ***
contacttelephone   -0.0085151  0.1942810  -0.044 0.965041    
contactunknown     -1.1178514  0.1636610  -6.830 8.47e-12 ***
day                 0.0112200  0.0066177   1.695 0.089989 .  
monthaug           -0.2398937  0.2050656  -1.170 0.242066    
monthdec            0.3841543  0.6012512   0.639 0.522872    
monthfeb            0.1978945  0.2532888   0.781 0.434626    
monthjan           -1.2387170  0.3184459  -3.890 0.000100 ***
monthjul           -0.6577868  0.2083929  -3.156 0.001597 ** 
monthjun            0.1603845  0.2418461   0.663 0.507223    
monthmar            1.8043146  0.3706641   4.868 1.13e-06 ***
monthmay           -0.8669394  0.2029454  -4.272 1.94e-05 ***
monthnov           -0.7453043  0.2248827  -3.314 0.000919 ***
monthoct            1.6939595  0.3260368   5.196 2.04e-07 ***
monthsep            0.8442097  0.4262381   1.981 0.047636 *  
duration            0.0060667  0.0002439  24.873  < 2e-16 ***
campaign           -0.1333340  0.0247262  -5.392 6.95e-08 ***
poutcomeother       0.7291476  0.2402802   3.035 0.002409 ** 
poutcomesuccess     2.6232942  0.2928452   8.958  < 2e-16 ***
poutcomeunknown    -0.2193644  0.1576808  -1.391 0.164167    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4695.2  on 3386  degrees of freedom
Residual deviance: 2721.2  on 3352  degrees of freedom
AIC: 3802.5

Number of Fisher Scoring iterations: 5
```

Comments on Weighted Logistic Regression:
This model takes into account the dependent variable class imbalance by using weighting. This model shows that jobretired, loanyes, contactunknown, monthmar, monthmay, monthnov, monthsep, duration, campaign, poutcomesuccess, jobself-employed, maritalmarried, monthjul are statistically significant to the dependent variable.

Assess weighted Logistic Regression Model

Model Validity and Coefficient evaluation

Baseline Logistic Regression Measures

predictors	odds_ratio	p_value
poutcomesuccess	13.7810467	0.0000

predictors	odds_ratio	p_value
monthmar	6.0758054	0.0000
monthoct	5.4409815	0.0000
monthsep	2.3261386	0.0476
poutcomeother	2.0733125	0.0024
jobstudent	1.8140641	0.0823
jobretired	1.6888277	0.0261
monthdec	1.4683719	0.5229
monthfeb	1.2188338	0.4346
monthjun	1.1739622	0.5072
jobunknown	1.1589631	0.7649
maritalsingle	1.0433361	0.7986
day	1.0112831	0.0900
duration	1.0060852	0.0000
contacttelephone	0.9915210	0.9650
jobmanagement	0.9528954	0.7881
jobhousemaid	0.8948622	0.7251
jobentrepreneur	0.8754951	0.6541
campaign	0.8751727	0.0000
housingyes	0.8517981	0.1534
poutcomeunknown	0.8030290	0.1642
`jobself-employed`	0.7891896	0.4257
monthaug	0.7867115	0.2421
maritalmarried	0.7449384	0.0499
jobtechnician	0.7369663	0.1094
jobservices	0.6769863	0.0818
monthjul	0.5179965	0.0016
monthnov	0.4745898	0.0009
`jobblue-collar`	0.4430995	0.0000
monthmay	0.4202358	0.0000
jobunemployed	0.4101522	0.0163
loanyes	0.4038421	0.0000
(Intercept)	0.3920355	0.0023
contactunknown	0.3269816	0.0000
monthjan	0.2897557	0.0001

Variable coefficients were also found and I will list the top 5 most influential predictors based on their coefficient value.
- poutcomesuccess (odds ratio = 13.78, p < 0.001): Clients with a previous campaign success are 13.8 times more likely to subscribe again compared to the reference group.
- monthoct (odds ratio = 5.44, p < 0.001): Contacts made in October are associated with 5.4× higher odds of a client subscribing. Suggests that October campaigns perform especially well.
- Duration (odds ratio = 1.006, p < 0.001): For each additional unit increase in call duration (seconds), the odds of subscription increase by about 0.6%. Longer conversations are linked to higher likelihoods of success.
- monthsep (odds ratio = 2.32, p = 0.047): Contacts made in September are about 2.3× more likely to result in a subscription.
- jobretired (odds ratio = 1.69, p = 0.026): Retired clients are about 1.7× more likely to subscribe.

Global likelihood Ratio Test

```
Analysis of Deviance Table

Model 1: y ~ 1
Model 2: y ~ job + marital + housing + loan + contact + day + month + 
    duration + campaign + poutcome
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)    
1      3386     4695.2                          
2      3352     2721.2 34     1974 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comments on LRT:
The LRT shows that the model as a whole has predictive value and the full regression model provides statistically significant improvement in fit than a null model with no predictors.

Model Fit Metrics

```
        Nagel
McFadden 0.632
   0.409     1
```

Comments on Goodness of Measures Fit:
The McFadden value is 0.402 which shows the logistic regression model is an excellent fit compared to a null model. The Nagelkerke variable is 0.625, this value is an adjusted version of the McFadden measure. A value of 0.625 suggests the model accounts for 62.5% of the variation in the outcome.

Predict - Weighted Logstic Regression Model

```
Confusion Matrix and Statistics


          Reference
Prediction   0   1
```

```
         0 817  19
         1 181 110

               Accuracy : 0.8225
                 95% CI : (0.799, 0.8444)
    No Information Rate : 0.8855
    P-Value [Acc > NIR] : 1

                  Kappa : 0.434

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8527
            Specificity : 0.8186
         Pos Pred Value : 0.3780
         Neg Pred Value : 0.9773
             Prevalence : 0.1145
         Detection Rate : 0.0976
   Detection Prevalence : 0.2582
      Balanced Accuracy : 0.8357

       'Positive' Class : 1

 F1 Score:  0.5238095
```

Comments on Predicting Weighted Logistic Regression Model:
This Weighted Logistic Regression Model is by far better than the raw logistic regression model. This Model’s Balanced Accuracy and F1 score is higher than the default model. This model also shows good predictive power for determining between dependent variable classes with both sensitivity and specificity being higher values and close together.

Overall, the Weighted Logistic Regression model demonstrated the strongest performance among the logistic regression models and will be carried forward into the evaluation phase. It will be compared against the best-performing classification tree and KNN model to determine which approach is most appropriate for addressing the business problem.

Summary of Modeling

The Modeling phase involved developing and tuning three classification models - classification trees, KNN, and logistic regression. These models were made to predict whether a client will subscribe to a term deposit. Two classification trees were made, unweighted and weighted, and were evaluated against the same metrics. The weighted classification tree proved to be the superior model. For KNN three types of KNN were made, unweighted, oversampling, and threshold tuning. These KNN models were evaluated all using the same metrics and the threshold tuning KNN model proved to be the superior model. Two types of Logistic regression models were made, unweighed and weighted, and were evaluated using the same metrics. The weighted logistic regression model proved to be the superior model. In the evaluation phase the weighted classification tree, threshold tuned KNN model, and weighted logistic regression model will be compared and evaluated to see which model is best suited for answering the business problem.

Evaluation

Compare Best Models Classification Models

Model Performance Comparsion

Metric	Weighted_Classification_tree	Knn_Threshold	LR_Weighted
Accuracy	0.714	0.770	0.823
Kappa	0.282	0.338	0.434
Sensitivity	0.858	0.816	0.853
Specificity	0.695	0.763	0.819
Pos Pred Value	0.267	0.309	0.378
Prevalence	0.115	0.115	0.114
Detection Rate	0.098	0.094	0.098
Balanced Accuracy	0.777	0.790	0.836
F1	0.407	0.448	0.524

Best Model

Comments on the Best Model:
Based on the best models presented, Weighted Logistic Regression is the best model. The Weighted Logistic Regression model has the highest balanced accuracy, meaning the model correctly identifies both classes of the dependent variable better than the other models. The model has strong sensitivity and specificity supporting the model has good predictive power of dependent variable classes despite the imbalanced dependent variable. The Weighted Logistic Regression model has the highest F1 score among models which shows it is the best at identifying the minority class and minimizing false positives. Although this logistic weighted regression model has slightly lower accuracy then the KNN threshold model, it is a minimal difference and this logistic regression model has significantly better class detection. The Logistic Regression Model will be the model used to deploy and answer the busines problem.

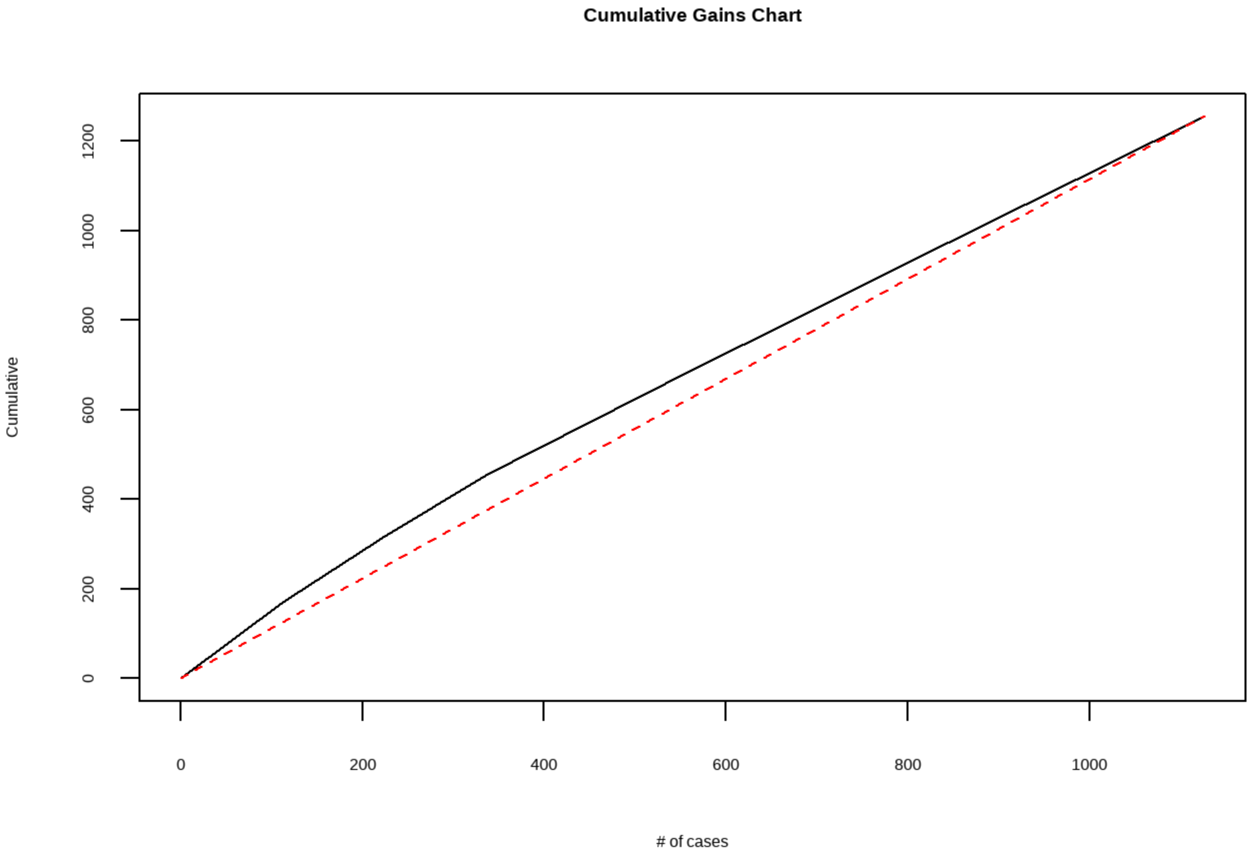
Model Evaluation Charts

Gains Table

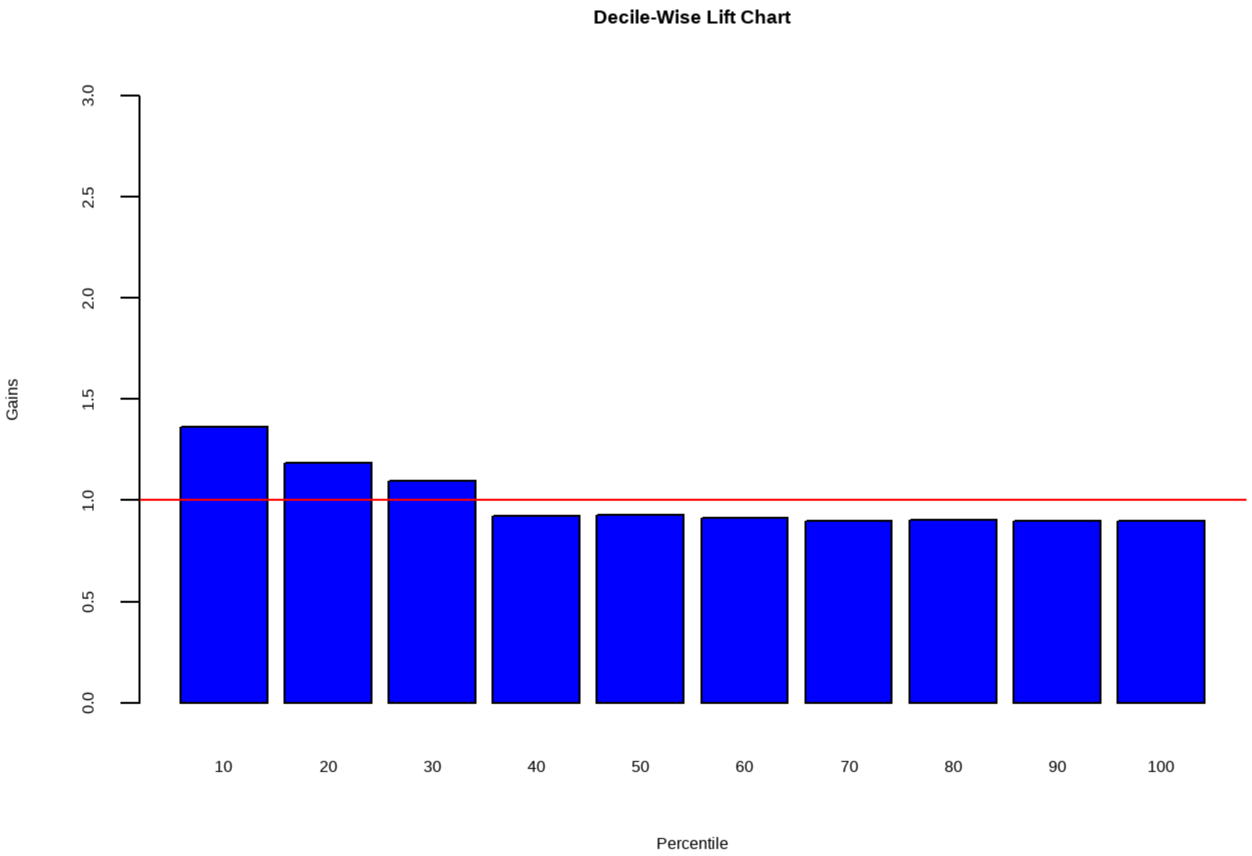
```
Depth                             Cume   Cume Pct                     Mean
 of          Cume      Mean      Mean   of Total    Lift   Cume     Model
File    N      N       Resp      Resp       Resp   Index   Lift     Score
-------------------------------------------------------------------------
  10   112    112       1.52      1.52      13.5%     136    136      0.95
  20   113    225       1.32      1.42      25.4%     118    127      0.74
  30   113    338       1.22      1.35      36.4%     110    121      0.51
```

```
     40   112    450       1.03       1.27       45.5%       92    114       0.35
     50   113    563       1.04       1.22       54.9%       93    110       0.25
     60   113    676       1.02       1.19       64.0%       91    107       0.19
     70   112    788       1.00       1.16       72.9%       90    104       0.13
     80   113    901       1.01       1.14       82.0%       91    103       0.09
     90   113   1014       1.00       1.13       91.0%       90    101       0.06
    100   113   1127       1.00       1.11      100.0%       90    100       0.02
```

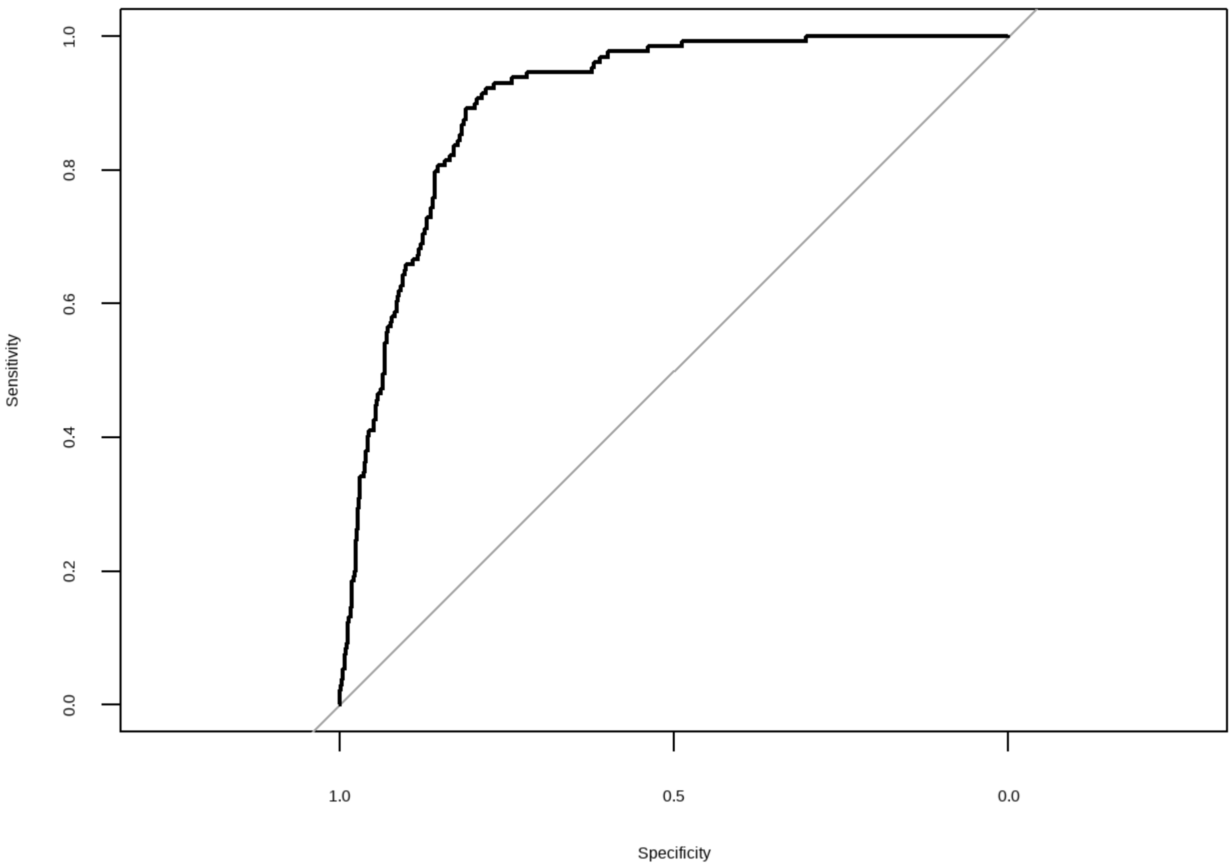
Cumulative Lift Chart



Decile Wise Lift Chart



ROC Curve With AUC Value

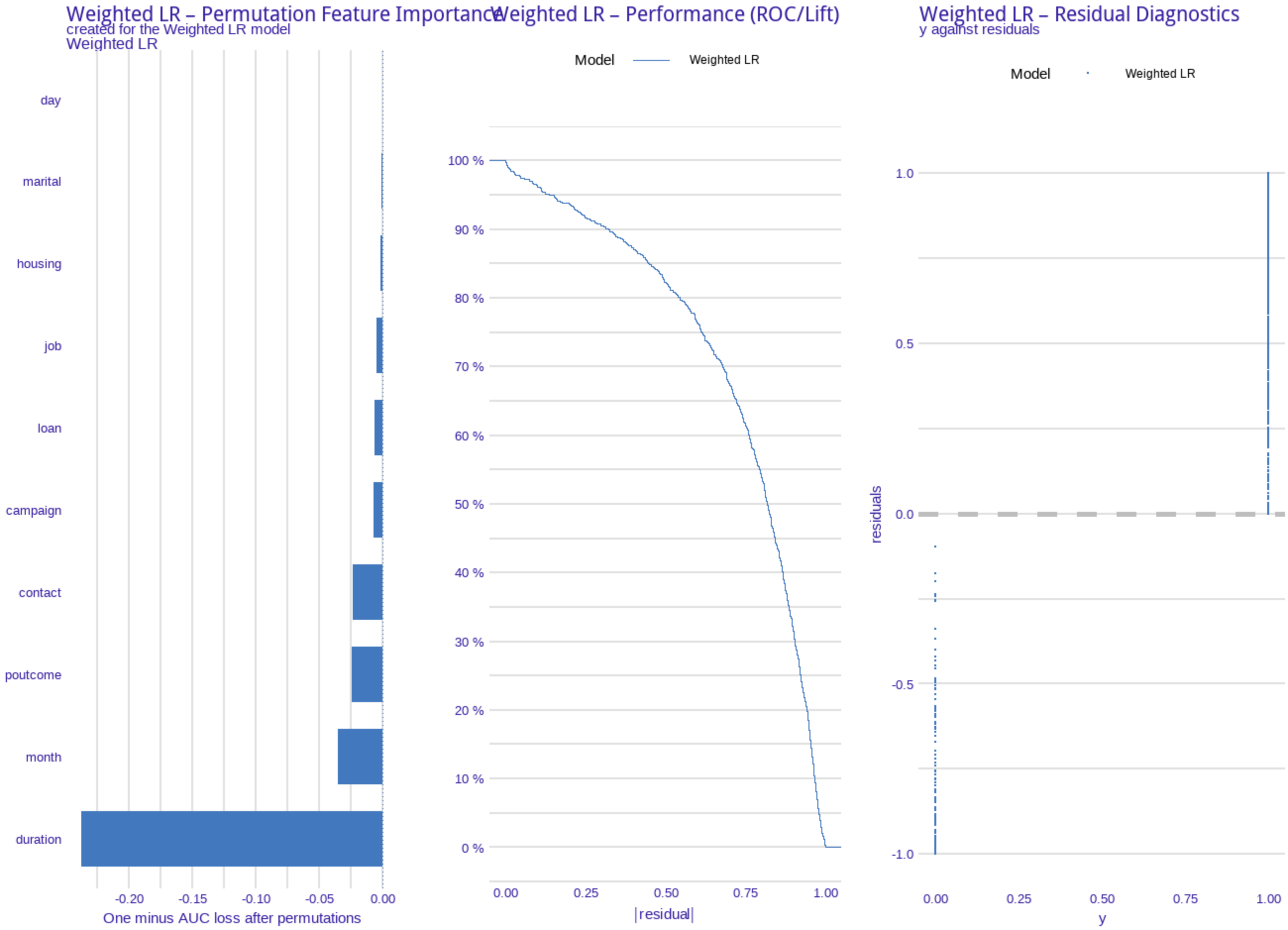


```
Area under the curve: 0.9011
```

Comments on Model Evaluation Charts:
Based on these evaluation charts, the model demonstrates strong predictive performance. The cumulative lift chart shows that the model performs better than random chance, as indicated by the black curve lying above the red diagonal line. The cumulative gains chart reveals that the top 30% of the dataset captures a large proportion of positive cases, supported by the high mean model score and cumulative lift values. For example, the first decile has a mean model score of 0.95 and a cumulative lift of 136, meaning the model is 36% more effective than random selection at identifying positives. The decile-wise lift chart confirms that the first three deciles outperform random chance, while the ROC curve, with an AUC value of 0.901, indicates excellent overall discrimination between the positive and negative classes.

Dalex Graph

```
Preparation of a new explainer is initiated
  -> model label       :  Weighted LR
  -> data              :  1127  rows  10  cols
  -> target variable   :  1127  values
  -> predict function  :  pfun
  -> predicted values  :  No value for predict function target column. (  default  )
  -> model_info        :  package caret , ver. 7.0.1 , task classification (  default  )
  -> predicted values  :  numerical, min =  0.001136535 , mean =  0.3290053 , max =  0.9999345
  -> residual function :  difference between y and yhat (  default  )
  -> residuals          :  numerical, min =  -0.9999345 , mean =  0.5565315 , max =  0.9988635
  A new explainer has been created!
```



Comments on the Dalex Graph:
The Dalex graph shows that duration is the dominant predictor in the weighted logistic regression model along with month, poutcome, and contact. The ROC style performance curve confirms strong discriminatory ability. The residual plot shows there is no pattern or systematic bias.

Summary of Evaluation

In the evaluation phase, the best-performing models from each modeling technique - weighted logistic regression, weighted classification tree, and KNN with threshold tuning - were compared using a set of metrics. Across the accuracy, error-based, and class imbalance sensitive metrics, weighted logistic regression consistently performed the best. Due to this weighted logistic regression was chosen to answer the business problem and will be used to help answer the two research questions. The evaluation charts further support these findings the cumulative gains and lift charts showed that the model identified positive cases far more effectively than random selection, with the top deciles capturing disproportionately high numbers of true positives. The ROC curve, with an AUC of 0.901, confirmed excellent discriminatory ability. The Dalex graph also showed information about the logistic regression supporting the logistic regression model had strong discriminatory ability and no systematic bias.

The Deployment phase will conclude the findings of this analysis and report in context of the business problem and research questions.

Deployment

Answer to Business Problem

The business problem guiding this analysis was to determine which clients are most likely to subscribe to a term deposit. Based on the full modeling and evaluation process, the weighted logistic regression model emerges as the most effective tool for this purpose. A model is only valuable if it performs meaningfully better than random chance, and the weighted logistic regression model meets this requirement with a balanced accuracy of 83.6%, reflecting strong performance across both outcome classes. A major concern in this project was the significant class imbalance in the dependent variable, where “no” responses greatly outnumbered “yes” responses. The weighted logistic regression model successfully mitigates this challenge, as evidenced by its high and closely aligned sensitivity and specificity values. This indicates that the model accurately identifies clients who are likely to subscribe while avoiding excessive misclassification of either class. The weighted logistic regression model is a reliable and practical solution for helping the bank target clients most likely to open a term deposit.

Answers to the Research Questions

1. What variables are most significant to people that will subscribe to a term deposit?

The Weighted Logistic Regression model shows which variables are most significant to people that subscribe to a term deposit. These Variables include:

- Duration: For each additional unit increase in call duration (seconds), the odds of subscription increase by about 0.6%. Longer conversations are linked to higher likelihoods of success.
- poutcomesuccesss: Clients with a previous campaign success are 13.8 times more likely to subscribe again compared to the reference group.
- jobretired: Retired clients are about 1.7× more likely to subscribe.

2. Are their certain times of the year where people are more likely to subscribe to a term deposit?

Yes, the weighted logistic model shows that there are certain times of the year people are more likely to subscribe to a term deposit. The two months that have the highest probability of clients subscribing are October and September. Contacts made in October are associated with 5.4× higher odds of a client subscribing. Suggests that October campaigns perform especially well. Contacts made in September are about 2.3× more likely to result in a subscription.

Business Recommendations

I have three recommendations that I would make based off this analysis. The first is to prioritize outreach during the high-conversion months of October and September. The second would be to target customer segments with higher likelihood of subscription, these include targeting clients that have had successful campaign outcomes in the past and retired clients. The third recommendation would be to increase engagement quality by extending call durations. Call duration was the strongest predictor in the weighted logistic regression model so call duration should be a priority.

Summary of Report

This analysis successfully identified the key factors that influence whether a client will subscribe to a term deposit and established a reliable, data-driven model to support targeted marketing efforts. By evaluating multiple classification techniques and accounting for the challenges of imbalanced data, the weighted logistic regression model emerged as the most effective and practical solution. Its strong predictive performance and interpretability make it well-suited for real-world application, enabling the bank to better allocate resources, refine outreach strategies, and improve conversion outcomes. With these insights and recommendations, the organization is now better equipped to engage the right clients at the right time, ultimately strengthening campaign effectiveness and driving higher subscription rates.

References

Citation of orignial data source with authors

Moro, S., Cortez, P., & Rita, P. (2012). Bank Marketing [Dataset]. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/222/bank+marketing

Citation of ChatGPT

OpenAI. (2025). ChatGPT (Version 5.1) [Large language model]. https://chat.openai.com/

Version of R

```
Version R version 4.5.2 (2025-10-31 ucrt)
```

R packages used during analysis

```
Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. Journal of Statistical
Software, 28(5), 1–26. https://doi.org/10.18637/jss.v028.i05
```

```
H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.
```

Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963

Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible Research in R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors, Implementing Reproducible Computational Research. Chapman and Hall/CRC. ISBN 978-1466561595

Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL https://www.jstatsoft.org/v40/i03/.

Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez and Markus Müller (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. BMC Bioinformatics, 12, p. 77.
DOI: 10.1186/1471-2105-12-77 <http://www.biomedcentral.com/1471-2105/12/77/>

End of Report