Data science final project report

Car Insurance Cold Calls

Xinyi Cai,

Lisha Xie,

Jiaxun Li,

Yicheng Liu,

- # Session 1: Introduction/Business Question

A bank in the United States wants to improve the effectiveness of its car insurance sales campaigns by identifying potential customers who are most likely to buy insurance. Since the bank has potential customers' data, and bank employees will call them to advertise available car insurance options, it generates a dataset that consists of the train file and test file. And the business question becomes "What are the key factors that influence the purchase of car insurance by customers?". By targeting the right consumers, the bank can focus its efforts and resources on the customers who are most likely to convert, resulting in higher sales and increased revenue.

To date, the bank may have used traditional methods, such as random calling, mailing and mass advertising through media, to attract potential customers for car insurance. However, these methods may not be as effective as the targeted data-driven solution.

The analysis of the dataset let us predict the factors that significantly impact customers' likelihood to purchase car insurance. And our conclusion can provide information for business to improve its marketing campaigns, which could potentially lead to better customer targeting and enhanced business value.

- **About The Data**

The provided dataset includes information about potential customers of a bank. The dataset consists of two files:

* `train.csv` - This file contains data about 4,000 customers who were contacted during the last campaign. The file includes 20 columns, which provide general information about customers such as age, job, education, marital status, etc. It also includes more specific information about the campaign, such as communication type, last contact day, number of contacts, and outcome.

The data types in both files include numerical, categorical, and binary variables. The numerical variables include age, balance, day, month, duration, and so on. The categorical variables include job, education, marital status, contact, outcome. The binary variables include default and loans.

The data was collected during the bank's car insurance sales campaign, which took place over a period of time. The data also includes the dates on which customers were contacted, which can provide insights into the timing and frequency of contacts and their impact on customer outcomes.

Overall, the data provides lots of information that can be used to build predictive models. And this dataset aims to predict for 1000 customers(test dataset) who were contacted during the current campaign, whether they will buy car insurance or not.

URL:(https://www.kaggle.com/datasets/kondla/carinsurance)

- Session 2:EDA

In this section, we will show some characteristics of this data set and make EDA. The summary statistics provide insghts into the numerical variables, including Age, Default, Balance, HHInsurance, CarLoan, LastContactDay, NoOfContacts, DaysPassed, PrevAttempts, and CarInsurance. Additionally, there are categorical variables like Job, Marital, Education, Communication, LastContactMonth, and Outcome. The summary statistics offer information such as minimum, maximum, mean, and quartiles for numeric variables and the frequency of categories for categorical variables.

Also, the frequency counts of the Job variable were calculated, revealing the number of occurrences for each unique job category. The highest frequency jobs in the dataset are:
blue-collar: 759 occurrences
management: 893 occurrences
technician: 660 occurrences

These jobs appear most frequently in the dataset, indicating that individuals working in blue-collar, management, and technician roles are more represented in the data compared to other job categories.

I also performed additional analyses on the dataset df` to identify missing values and duplicate rows. This indicates that there are 4132 missing values, and no duplicated rows present in the dataset across various variables.

  - Data visualization

    - Skewness

      The resulting bar plot [See appendix 2]visualizes the skewness values of the variables in the df_num data-frame, providing an overview of the skewness distribution across the selected numeric variables. This can help identify variables that exhibit significant skewness, which may impact subsequent analysis or modeling approaches.As we can see from the value above, the balance, Noofcontacts,Prevattempts are skewed since it value excess 1.

    - Histogram

      A series of histograms are plotted also, each representing the distribution of the corresponding column in the **`df_num`** dataframe. These histograms provide insights into the shape, central tendency, and variability of the data for each variable.We can find balance, Noofcontacts,Prevattempts have long tails. It matches the skewness finding that the feature balance, Noofcontacts,Prevattempts is skewed.

    - Outliers

      By running this code, a series of boxplots will be generated, each representing the distribution and summary statistics (such as quartiles, outliers, and median) of the corresponding column in the **`df_num`** dataframe. And we can see that there are a lot of outliers in our dataset.

- # Session 3: Data preprocessing

  In the original Dataframes, we found that there were many missing values and other defects that prevented the model from being built. So, in this section, we'll address each of these issues.
  First, there are so many missing values for Outcome and communication that if we were to remove 80% rows from both, we would change the missing values in these two columns to 'None'. Then calculate the total missing values, there are 188. Since we have a total of 4000 records, the removal of 188 records will not affect the establishment of our model, so we choose to directly remove these 188 records.
  We also realize that there are fields in the data frame whose values are characters rather than numbers, so we need to encode these categorical variables.
  At the same time, we find that 'CallStart' and 'CallEnd' are two points in time, so we divide the two to get a new column called CallDuration as a numeric variable in our data frame. At this point, all of our preprocessing is over, and all of the values in the data frame have been converted into numbers that can be modeled.

- # Session 4: Modeling

  - Modeling preparation

  For our modeling convenience, we replace all 1s with "Yes" and all 0s with "No" for the CarInsurance variable。And we change all the hyphens into underscores.

  We select 955 indices out of 3820 (25% of dataset). Using these indices, we create a test dataset(955 indices) and a training dataset(2865 indices)

  Then we select the data instances in the rows in `index` and save them as `test`. The rest will be saved as `training`. By observing the heatmap, we choose 10 variables with high positive or negative correlation values.

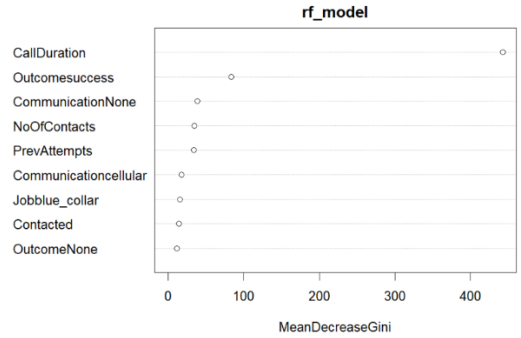
  - Random Forest

  First, we build a random forest model. The first step we do is converting the target variable "CarInsurance" in our training data into a factor. This is necessary because random forest models require the target variable to be a factor for classification tasks.Then we set the seed for the random number generator in R to 1 to make sure we will get the same result each time.We create a random forest model with 500 trees (ntree = 500) using the randomForest function. The target variable is "CarInsurance", and the predictor variables are CallDuration, Outcomesuccess, Communicationcellular, Contacted, CommunicationNone, OutcomeNone, Jobblue_collar, NoOfContacts, and PrevAttempts. The cutoff = c(0.5,0.5) argument sets the threshold for classifying observations into each class. With equal cutoffs (0.5), an observation is classified into the class that gets more than 50% of the votes from the trees. After that, rf_model will contain a trained random forest model, which can be used for making predictions on new data.

  By checking the performance of our rf model. The OOB error rate is 20.21%. It means roughly one in five predictions the model made on data it was not trained on were incorrect. The OOB error rate might seem relatively low. It gives a class error rate of about 14.8% for the "No" class and 28.37% for the "Yes" class. This might suggest an imbalance in the training data, where there might be

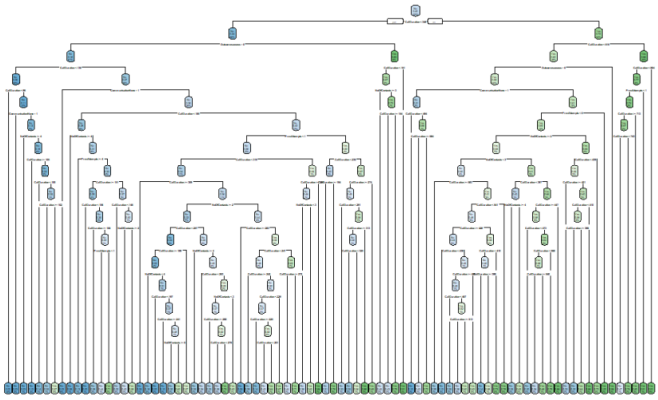
fewer "Yes" instances than "No" instances, making it harder for the model to learn to correctly classify the "Yes" instances.

We create a plot of variable importance. This can tell us which variables are most useful for making accurate predictions with our model. It shows that CallDuartion, Outcomesuccess, CoomunicationNone, NoOfContacts and Prevattempts are the most important variables.



- Classification Tree Model

Now we build a classification tree model with the five most important variables. We set cp=0 to include all the variables possible for now.

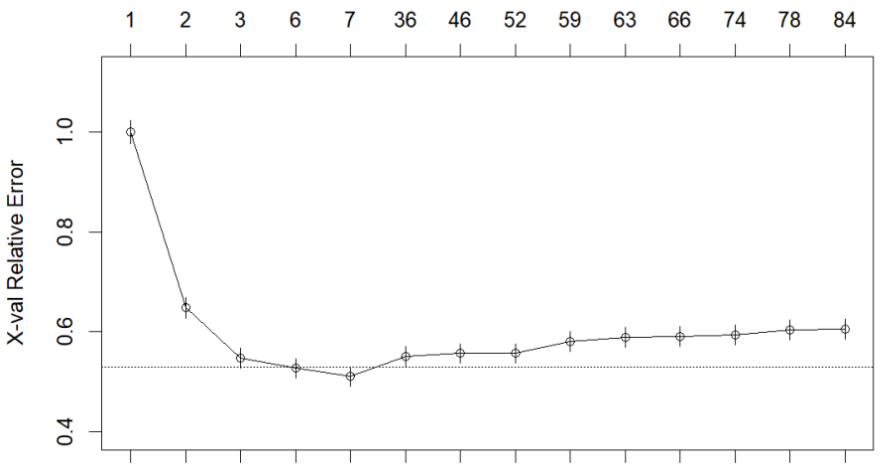


- predicting probabilities/class labels for test data

Then, we apply the model to the test dataset and get the predicted values. We check the accuracy of the model by comparing the the actual values (default) and the predicted values (ct_pred_class). The accuracy is 77.38%. This is a reasonable accuracy rate, but we may want to aim for a higher accuracy.By checking the confusion table, we get TPR = 0.85 and FPR = 0.28. The TPR suggesting that the model is effective at identifying "Yes" instances. The FPR is relatively high, suggesting that the model is making quite a few type I errors.

We check the error rate at the root node, it's about 39.86%. This is the baseline error rate, which the tree will try to improve upon by splitting the data.

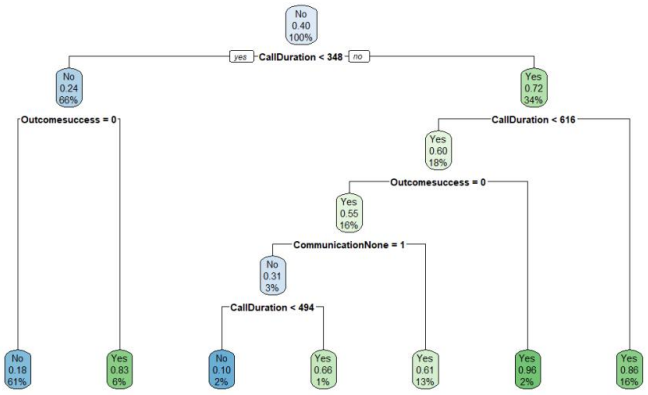
We check how the cross-validation error rate changes as the complexity of the model increases. The cross-validated error (xerror) starts to increase after a certain point, suggesting potential overfitting.



Classification trees are a good choice when you need models that are easy to understand and interpret. It is suitable for nonlinear relationship and complex interaction of data. The disadvantage is that overfitting is easy and requires pruning to mitigate overfitting.

So we use the information above to prune the tree to its optimal size using the prune() function with the CP value at which xerror is at its minimum. It chooses the one with 7 splits.

Let's consider min_xerror_tree as the best pruned tree, and get the prediction. We see the accuracy rate has been improved to approximately 80%. And the TPR has been raised to 85.07%.

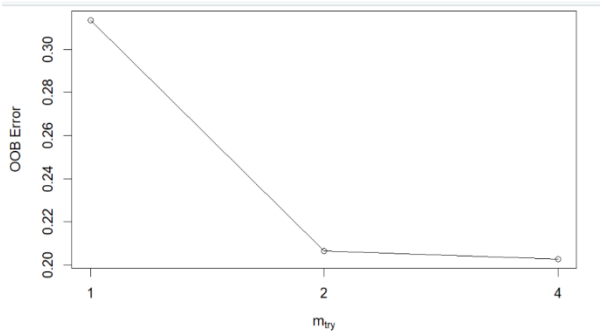


Following a similar process, we create a random forest model again with these five variables. The performance is similar to the first model we build.

Random forest performs well when the data features have nonlinear relationships and highly complex interactions. It can reduce overfitting risk and improve generalization performance. The disadvantages are that the model is relatively poor in interpretation and can be slow in training and prediction.

- hyperparameter tuning for Random Forest

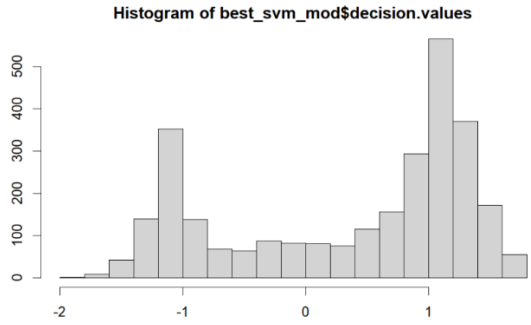
Now we try to tune our rf model by 500 Tree Tries. By looking at the graph, we observe the lowest OOB error.



We use that model as the rf_best_model. We get 74.76% accuracy.Our OOB estimate of error rate is 25.93%.The TPR is 69.2% and FPR is 23.4%. The performance of our rf_model is worse than our ct model

- SVM

Then we build a SVM model. SVM model performs well when data sets are small or medium in size. It is suitable for linear and nonlinear classification problems, especially when the characteristic dimension is high. The disadvantage is that the interpretation of the model is relatively poor and the parameter tuning may be complicated. Here we tune SVM models using `tune` function. Set a range of search values for the parameter. It builds an SVM model for each possible combination of parameter values and evaluate accuracy. It will return the parameter combination that yields the best accuracy. The best performance of our SVM model is 0.2108148. The performance seems good. Then we build a SVM model. SVM model performs well when data sets are small or medium in size. It is suitable for linear and nonlinear classification problems, especially when the characteristic dimension is high. The disadvantage is that the interpretation of the model is relatively poor and the parameter tuning may be complicated. Here we tune SVM models using `tune` function. Set a range of search values for the parameter. It builds an SVM model for each possible combination of parameter values and evaluate accuracy. It will return the parameter combination that yields the best accuracy. The best performance of our SVM model is 0.2108148. The performance seems good.



- Logit regression

Next, we build a logit regression model. It returns 79.7% accuracy rate.

```
FALSE  TRUE
  194   761
        actual
predicted  No Yes
      No  511 129
      Yes  65 250
[1] 0.7968586
```

- Hold-out validation

Logistic regression is applicable to classification problems where there is a linear relationship between data features. It is simple and easy to interpret. Also, it can output probability values that help further analysis and business decisions. The disadvantage is that the ability to deal with nonlinear problems and feature interaction is weak. Here we apply hold-out validation. By checking the importance of variables, we drop one least important variable.

- Performance Visualization with ROC

At last, we build ROC graphs to check all four models' performances.

We draw ROC curves of all models and compare them. We found the logit regression model returns the highest AUC of 0.883.The second highest AUC is the SVM model. That suggest regression is a better approach to model our datasets.

• Session 5: Model evaluation

Our randomforest model after tunning has 74.9% accuracy, and Auc is 0.810. The Classification tree model has 78.01% accuracy and AUC is 0.865. Also, after pruning by applying the cp value when it reaches the min error, the accuracy can increase to 79.9%. Svm has 79.9% accuracy and AUC is 0.874.Last, logistic regression model has 79.6% accuracy and AUC is 0.884.

Based on our analysis, we will select logistic regression model as the best classifier because it has the highest AUC. We believe the AUC is the metric we should look into for the following reasons. First, our data is somewhat unbalanced. AUC is more robust to class imbalance and provides a comprehensive assessment of a model's performance across different classification thresholds.

Second, Auc is not affected by the threshold because it considers the performance of the model across all possible thresholds and provides an aggregated measure.

We believe that our model can have a pretty good performance if applied to the real company. First of all, we see that the accuracy and AUC is high enough. And then when we check the confusion matrix, we can see that its ability to predict the Positive case and Negative case are both good. Furthermore, logistic regression model is easy to interpret. If we would like to check which features contribute most to our prediction, we can simply check the coefficient of the variables.

To evaluate the result, we can generate confusion matrix and do calculations. By calculating the accuracy, we can see the overall correctness of the predictions and is calculated as the number of correct predictions divided by the total number of predictions. Also, we can calculate Precision, Sensitivity, Specificity, AUC-ROC, etc.

By setting up a specific goal, carrying out certain actions to achieve the goal, using the model to predict the outcome, and evaluating the results and determining what can be improved, we can develop a business case to project expected improvement.

- Session 6: Deployment and Conclusions

    - Specific Deployment Plan

Identify a business objective: When it comes to actually carrying out a business plan . We can identify a business objective to the bank such as increasing the conversion rate of car insurance sales.
Determine the baseline: Next step is to determine the baseline by accessing the current conversion rate of car insurance sales.
Set up a KPI: Then, we can set up a KPI for the bank, indicating success such as reaching a certain percentage of increase in a conversion rate.
Estimated the result: Then, the bank can fit the data into our model to estimate the expected sales.
Reflection and Improvement: Based on the predicted results, if it does not achieve the goal, the bank should do the reflection to see which part they do not perform well. They can basically check those important features in our model such as communication. It indicates that if the bank previously contacts clients, those people are more likely to purchase. Thus, bank many spending more money on communication expenses. Also, the bank can try to increase the call duration and try to convey more information to the clients, which may increase the sales as well.

    - Matters of attention

Data Quality and Availability: It is important to ensure that the data used for training the model is representative, accurate, and up to date. Additionally, consider potential issues such as missing data, outliers, and data biases that may impact the performance and fairness of the deployed model.
Integration with Existing Systems: Ensure that the data science solution can be integrated smoothly with the existing systems and workflows of the firm.
Maintenance and Updates: It is crucial to always monitor the model's performance, addressing issues as they arise. Also, regularly update the model when needed.

    - Ethical considerations

It is vital to consider issues such as data privacy, data security, fairness, transparency, and compliance with relevant regulations. Implement appropriate safeguards to protect sensitive customer information and ensure that the model's predictions are fair and unbiased. Try to increase the sales through the phone call but not too largely impact clients daily life.

    - Risks

Inadequate maintenance and upgrades: It may result in model degradation, security flaws, or non-compliance with increasing legislation. In order to solve it, we can create a solid maintenance strategy that includes frequent model monitoring, performance review, and upgrades. Keep up to date on the latest advances in the sector in order to adopt new techniques or approaches as needed. Implement security measures to guard against potential risks to the deployed solution.
Largely rely on the model: While the model overall has a good performance, it may be

misleading sometimes. In order to avoid this problem, people should always incorporate in their own thoughts to interpret the result and use the prediction wisely.
Resistant to change: Employees may resist adopting the data science solution or face difficulties in using it effectively. To solve it, we can communicate the benefits of the solution, and provide comprehensive training and support to employees. Address their concerns and give timely feedback. Also, we can conduct an acceptance testing and iterate on the solution based on the feedback to improve usability and adoption.

- Conclusion

Overall, our logistic regression model can have an excellent performance on the current dataset provided. However, it can still be improved and updated by more data collected from the bank. Also, it is important to be careful of those things mentioned above while carrying out and try our best to avoid the risks or mitigate them when we discover them.

- Appendix:

Data Dictionary

Here is a data dictionary for the provided dataset:

* `ID` - Unique identifier for each customer

* `Age` - Age of the customer in years

* `Job` - Type of job of the customer (categorical: "admin.", "blue-collar", "entrepreneur", "housemaid", "management", "retired", "self-employed", "services", "student", "technician", "unemployed", "NA")

* `Marital` - Marital status of the customer (categorical: "divorced", "married", "single", "NA")

* `Education` - Education level of the customer (categorical: "primary", "secondary", "tertiary", "NA")

* `Default` - If the customer has credit in default (binary: "yes - 1", "no - 0")

* `Balance` - Average yearly balance in USD

* `HHInsurance` - is household insured (binary:"yes - 1", "no - 0")

* `CarLoan` - If the customer has a car loan (binary: "yes - 1", "no - 0").

* `Communication` - Contact communication type (categorical: "cellular", "telephone", "NA")

* `LastContactDay` - Day of the last contact

* `LastContactMonth` - Month of the last contact (categorical: "jan", "feb", "mar", ..., "nov", "dec")

* `CallStart` - Start time of the last call

* `CallEnd` - End time of the last call

* `CallDuration` - How long the call takes

* `NoOfContacts` - Number of contacts performed during this campaign for this consumer

* `Contacted` - Whether the consumer is contacted or not

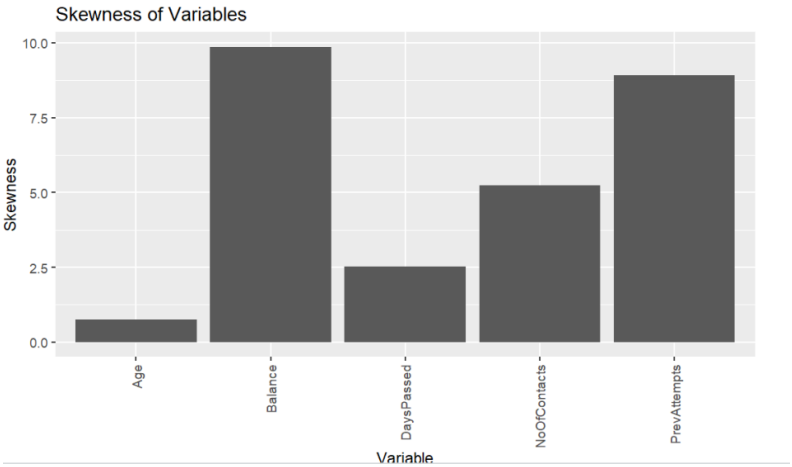
* `PrevAttempts` - Number of contacts performed before this campaign for this customer

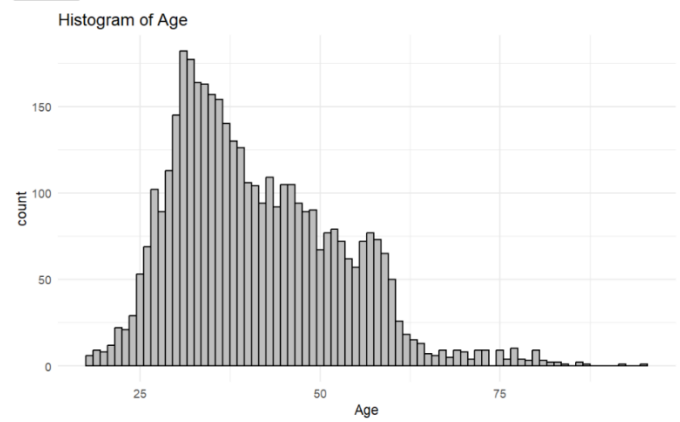
* `Outcome` - Outcome of the previous marketing campaign (categorical: "failure", “other”, "success", “NA”)

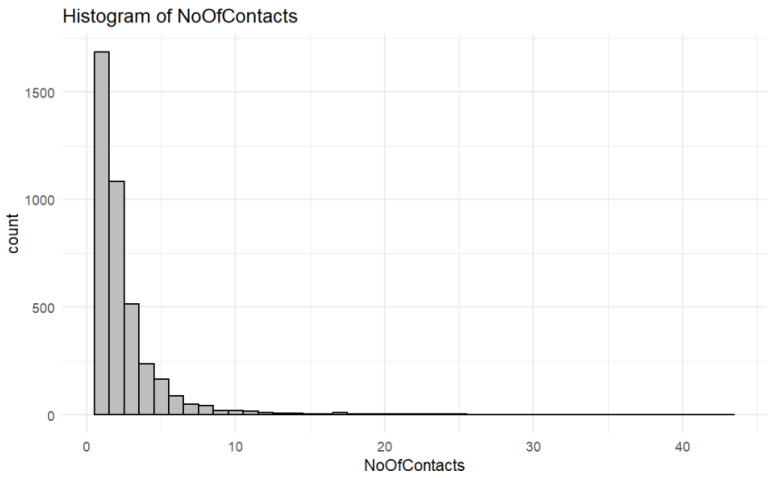
* `CarInsurance` - Target variable, indicating if the customer bought car insurance (binary: "yes -1", "no -0")

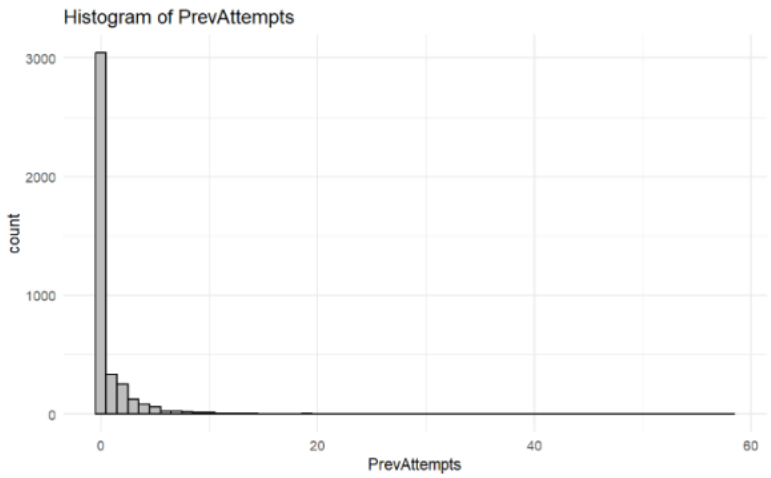
In summary, the data set includes detailed personal information about potential customers, as well as information about their previous interactions with the bank's sales campaigns. The target variable is whether the customer bought car insurance during the current campaign.

- Explanatory Data analysis graphs








Histogram of PrevAttempts
3000
2000
1000
0
count
0
20
40
60
PrevAttempts