DSA211 Statistical Learning with R

INSURANCE COLD CALLS

Prepared by: G1 Group 4

Student	Matriculation Number
Chia Dehan	01332535
Chong Wen Kai Justin	01335794
Eugene Wong Wei Huan	01334933
Lim Jing Zhe Austin	01335331
Ryan Tay Zhi Hao	01333349

Prepared for:

Dr Kwong Koon Shin
Professor, School of Economics
Singapore Management University

27 March 2020

Table of Contents

Executive Summary

We aimed to model cold-calling success rates in the context of selling insurance. Traditional marketing managers often use cold calling as a part of their sales campaign in order to sell insurance policies to new generated leads. However, cold callers traditionally face a success rate of 1-2%, and a model predicting the probability of a successful cold call, given the recipient's characteristics, would increase the success rate of their cold calls, and subsequently the success of their marketing campaign.

Traditionally, a brute force approach is employed which not only wastes organisational resources and time but could affect the long term effectiveness of the caller through call reluctance as callers grow wary of rejection. We believe that we have proven that a data-centric approach to accurately predict the success of cold calls can help alleviate this industry challenge through our model.

The data was sourced from a data mining competition dataset from the Technical University of Munich and is provided by an anonymised bank that sells car insurance to clients through cold calling. Considering the fact that the bank has information regarding prospective clients, the data can be used to optimise the accuracy in identifying clients that are willing and unwilling to purchase car insurance to increase the effectiveness of the bank's cold calling campaign.

Multiple GLM models created using different selection methods for significant variables and a Classification Tree were evaluated based on their accuracy in identifying clients who were interested in purchasing car insurance. From our evaluation, the most accurate model was the GLM model:

$$log\left(\frac{p(call\ success)}{p(call\ failure)}\right) = \mathbf{-2.13781158 - 1.66715158} * \boldsymbol{Entrepreneur1} + \mathbf{1.45700202} * \boldsymbol{HHInsurance0} - \mathbf{0.62457171} * \boldsymbol{CarLoan} - \mathbf{0.08442889} * \boldsymbol{NoOfContacts} + \mathbf{1.89481397} * \boldsymbol{PrevSucc1} + \mathbf{0.30153617} * \boldsymbol{CallDuration}$$

Our practical recommendations are as follows:

1. Firstly, callers should try to engage more with their recipients to increase the call duration to over for better chance of success.
2. Callers should also not continuously approach a recipient multiple times to increase the success of cold calling.
3. Companies should also not target entrepreneurs as they may be more willing to take risks and thus more unwilling to buy insurance.
4. Customers who have a car loan may be more unwilling to buy insurance due to increased costs
5. Subsequently, companies should target customers who are already customers of existing complementary products, in this case, customers who already have household insurance will be more willing to purchase car insurance after the call.
6. Customers who have bought other products during previous marketing campaigns will also be more likely to buy after the cold call.

We believe that while this predictive model is an important step forward in the direction of digital transformation of traditional marketing, the predictive model can be improved further through actual deployment and obtaining more data tailored to the context of specific companies to increase the predictive accuracy and consistency of the model.

1. Project Background

This project aims to obtain a statistical model for banks to improve on their cold call success rates, in selling car insurance. Typically, cold calls result in about a 1-3% success rate (*Jantsch, 2010*). We believe that this value has the potential to be further increased through targeting customers based on factors which suggest higher success rates. As such, our objective is to obtain a model which is able to accurately predict the success rates of cold calls. To obtain this model, we analysed data of 4000 customers (*Data Mining Cup, 2017*) who were contacted during a campaign. There were a total of 18 explanatory variables given in the data, with the features shown in *Table 1*.

Feature	Description	Example
Id	Unique ID number. Predictions file should contain this feature.	"1" ... "5000"
Age	Age of the client	
Job	Job of the client.	"admin.", "blue-collar", etc.
Marital	Marital status of the client	"divorced", "married", "single"
Education	Education level of the client	"primary", "secondary", etc.
Default	Has credit in default?	"yes" - 1,"no" - 0
Balance	Average yearly balance, in USD	
HHInsurance	Is household insured	"yes" - 1,"no" - 0
CarLoan	Has the client a car loan	"yes" - 1,"no" - 0
Communication	Contact communication type	"cellular", "telephone", "NA"
LastContactMonth	Month of the last contact	"jan", "feb", etc.
LastContactDay	Day of the last contact	
CallStart	Start time of the last call (HH:MM:SS)	12:43:15
CallEnd	End time of the last call (HH:MM:SS)	12:43:15
NoOfContacts	Number of contacts performed during this campaign for this client	
DaysPassed	Number of days that passed by after the client was last contacted from a previous campaign (numeric; -1 means client was not previously contacted)	
PrevAttempts	Number of contacts performed before this campaign and for this client	
Outcome	Outcome of the previous marketing campaign	"failure", "other", "success", "NA"
CarInsurance	Has the client subscribed a CarInsurance?	"yes" - 1,"no" - 0

Table 1: Feature Summary

Our code is placed in Appendix A, and more information about the data source and the dataset itself is found in Appendix B.

2. Car Insurance Data

2.1 Data Preparation

Firstly, we load the data onto R and obtain the initial dataset summary as seen in *Figure 2.1*. We noticed that there were rows with NA, we then proceeded to omit those rows from our analysis, and also converted categorical variables to factors. Furthermore, we converted *CallStart* and *CallEnd* Variables into *CallDuration* via [*CallEnd* - *CallStart]*. The resultant dataset summary is shown in *Figure 2.2.*

```
> summary(coldcall)
      Age                  Job           Marital         Education     Default
 Min.   :18.00   management :231   single  :309   tertiary :346   0:903  
 1st Qu.:33.00   technician :152   married :492   primary  :101   1:  4  
 Median :38.00   blue-collar:131   divorced:106   secondary:460          
 Mean   :41.26   admin.     :114                                         
 3rd Qu.:48.00   services   : 69                                         
 Max.   :82.00   retired    : 64                                         
                 (Other)    :146                                         
    Balance      HHInsurance CarLoan   Communication  NoOfContacts      DaysPassed   
 Min.   : -982   1:462       0:818   telephone: 69   Min.   : 1.000   Min.   :  1.0  
 1st Qu.:  228   0:445       1: 89   cellular :838   1st Qu.: 1.000   1st Qu.:102.0  
 Median :  724                                       Median : 1.000   Median :182.0  
 Mean   : 1741                                       Mean   : 1.929   Mean   :204.8  
 3rd Qu.: 1947                                       3rd Qu.: 2.000   3rd Qu.:288.0  
 Max.   :52587                                       Max.   :12.000   Max.   :854.0  

  PrevAttempts        Outcome     CarInsurance  CallDuration   
 Min.   : 1.000   failure:417   0:381        Min.   : 0.1167  
 1st Qu.: 1.000   other  :185   1:526        1st Qu.: 2.4500  
 Median : 2.000   success:305                Median : 4.1667  
 Mean   : 2.988                              Mean   : 5.5958  
 3rd Qu.: 4.000                              3rd Qu.: 6.9333  
 Max.   :58.000                              Max.   :36.4000  
```

Figure 2.1: Dataset Summary (before preparation)

```
> summary(coldcall)
      Age                Job           Marital         Education   Default    Balance       HHInsurance
 Min.   :18.00   management :231   single  :309   tertiary :346   0:903   Min.   : -982   1:462      
 1st Qu.:33.00   technician :152   married :492   primary  :101   1:  4   1st Qu.:  228   0:445      
 Median :38.00   blue-collar:131   divorced:106   secondary:460           Median :  724              
 Mean   :41.26   admin.     :114                                          Mean   : 1741              
 3rd Qu.:48.00   services   : 69                                          3rd Qu.: 1947              
 Max.   :82.00   retired    : 64                                          Max.   :52587              
                 (Other)    :146                                                                     
 CarLoan    Communication  NoOfContacts      DaysPassed     PrevAttempts        Outcome     CarInsurance
 0:818   telephone: 69   Min.   : 1.000   Min.   :  1.0   Min.   : 1.000   failure:417   0:381       
 1: 89   cellular :838   1st Qu.: 1.000   1st Qu.:102.0   1st Qu.: 1.000   other  :185   1:526       
                         Median : 1.000   Median :182.0   Median : 2.000   success:305               
                         Mean   : 1.929   Mean   :204.8   Mean   : 2.988                             
                         3rd Qu.: 2.000   3rd Qu.:288.0   3rd Qu.: 4.000                             
                         Max.   :12.000   Max.   :854.0   Max.   :58.000                             

  CallDuration   
 Min.   : 0.1167  
 1st Qu.: 2.4500  
 Median : 4.1667  
 Mean   : 5.5958  
 3rd Qu.: 6.9333  
 Max.   :36.4000  
```

Figure 2.2: Dataset Summary (after preparation)

2.2 Exploratory Data Analysis

Observing the univariate plots, frequency bar charts and qq plots of the features we noticed no serious outlying values in the data, refer to Appendix C for the plots. From the correlation matrix, in

Appendix D, we observe that there is generally a strong correlation between the dummy variables. Else, there were no further strong correlations observed.

3. Data Modelling

After examining the data set, we attempted a few techniques to generate appropriate models to predict whether a cold call will be successful in selling the insurance.. This included logistic modelling, best subset selection, ridge and lasso regressions and decision trees.

3.1 Logistic Regression Model (M1)

Looking at a logistic regression model, as the predicted outcome is a binary "Yes" or "No", it would be appropriate for analysing the data over a multiple linear regression model. From *Figure 3.1*, it is observed that the variables that are significant to a level of 0.05 are *Jobentrepreneur, HHInsurance0, CarLoan1, noOfContacts, Outcomesuccess, and CallDuration*. (*Refer to 1.0 Project Background for representation of these variables*)

Figure 3.1: Logistic model M1 summary

Looking at the chi-square test statistic from *Figure 3.2*, we find a p-value of 0.9688275 which is larger than the critical p-value at a 0.05 significance level. As such, we have insufficient evidence to reject the null hypothesis and it can be said that this logistic regression model (m1) is a good fitting model. The coefficients of m1 are shown in *Figure 3.3*.

```
> pvalue1 <- 1-pchisq(630.98, 699)
> pvalue1
[1] 0.9688275
```

Figure 3.2: Logistic model m1 p-value

```
> coef(m1)
    (Intercept)                Age     Jobblue-collar         Jobstudent      Jobtechnician          Jobadmin.              Jobservices
  -1.890228e+00       2.320602e-03      -5.022944e-01       2.390091e-01      -6.337630e-01      -1.455766e-01            -3.632631e-01
Jobself-employed        Jobretired       Jobhousemaid   Jobentrepreneur      Jobunemployed     Maritalmarried          Maritaldivorced
   5.533830e-02       1.967282e-02      -1.010248e+00      -1.777623e+00       6.930917e-02      -1.423838e-01            -1.134335e-01
Educationprimary Educationsecondary          Default1            Balance        HHInsurance0          CarLoan1 Communicationcellular
  -6.984882e-01      -4.330360e-01      -9.906830e-01      -1.405292e-05       1.358811e+00      -5.843215e-01            -6.142252e-02
    NoOfContacts         DaysPassed        PrevAttempts       Outcomeother     Outcomesuccess       CallDuration
  -1.211599e-01       8.934564e-04       5.344284e-02       2.520258e-01       2.031765e+00       3.104037e-01
```

Figure 3.3: Logistic model m1 coefficients

For the numerical variables, this means that an increase by 1 unit of Xi would result in the corresponding increase in the log odds-ratio by the corresponding coefficient Bi. For Categorical dummy variables, the log odds-ratio is affected by the corresponding coefficient Bi when the dummy variables are included or not included.

Building on m1, we re-ran a logistic regression on solely the significance at 5% variables mentioned above. Summary of m11 shown in *Figure 3.4*.

```
> summary(m11)

Call:
glm(formula = CarInsurance ~ Entrepreneur + HHInsurance + CarLoan +
    NoOfContacts + PrevSucc + CallDuration, family = binomial,
    data = new_coldcall[train, ])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.8607  -0.6422   0.2691   0.6795   2.2068

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2.13781    0.26124  -8.183 2.76e-16 ***
Entrepreneur1 -1.66715    0.76324  -2.184   0.0289 *
HHInsurance0   1.45700    0.20055   7.265 3.73e-13 ***
CarLoan1      -0.62457    0.33413  -1.869   0.0616 .
NoOfContacts  -0.08443    0.07442  -1.134   0.2566
PrevSucc1      1.89481    0.23756   7.976 1.51e-15 ***
CallDuration   0.30154    0.03217   9.373  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 990.32  on 725  degrees of freedom
Residual deviance: 655.61  on 719  degrees of freedom
AIC: 669.61

Number of Fisher Scoring iterations: 5
```

Figure 3.4: Logistic model m11 summary

```
> pvalue2 <- 1-pchisq(655.61, 719)
> pvalue2
[1] 0.9559716
```

Figure 3.5: Logistic model m11 p-value

Looking at the chi-square test statistic in *Figure 3.5*, we find a p-value of 0.9559716 which is larger than the critical p-value at a 0.05 significance level. As such, we have insufficient evidence to reject the null hypothesis and it can be said that this logistic regression model m11 is a good fitting model, with the coefficients shown in *Figure 3.6*.

```
> coef(m11)
 (Intercept) Entrepreneur1  HHInsurance0     CarLoan1  NoOfContacts    PrevSucc1  CallDuration
 -2.13781158   -1.66715158    1.45700202  -0.62457171   -0.08442889   1.89481497    0.30153617
```

Figure 3.6: Logistic model m11 coefficients

Thus our model is:
$log(\frac{p(call\ success)}{p(call\ failure)})$ = **-2.13781158 -1.66715158****Entrepreneur1*** **+ 1.45700202****HHInsurance0*** **- 0.62457171****CarLoan*** **- 0.08442889****NoOfContacts*** **+ 1.89481397****PrevSucc1*** **+ 0.30153617****CallDuration***

This would mean that if the call recipient has already bought household insurance from the bank, the log odds-ratio of buying car insurance from the campaign increases by 0.25819396. If the recipient has bought insurance due to a previous marketing campaign, then the log odds-ratio increases by 0.31622805. Lastly, if the call duration increases by 1 minute, the log odds-ratio will increase by 0.03281444.

When the variable is numerical, the coefficient represents the increase in log odds-ratio when there is a unit increase in the variable by 1. If the recipient has 1 more contact from a previous marketing campaign, the logs odds-ratio decreases by 0.08442889. If the call duration increases by 1 minute, the log odds-ratio will increase by 0.30153617.

When the variable is categorical, the coefficient represents the increase in logs odds-ratio when the variable is present. If the recipient is an Entrepreneur, then the log odds-ratio decreases by 1.66715158. If the recipient is household insured, then the log odds-ratio will increase by 1.45700202. If the recipient is taking a car loan, then the log odds-ratio decreases by 0.62457171. If the recipient has bought insurance due to a previous marketing campaign, then the log odds-ratio increases by 1.89481497.

3.2 Best Subset Selection (M2)

As we obtained a model that has 26 independent variables in M1, including dummy variables, the variance of our model has increased, potentially reducing the accuracy of our model. Thus, to improve it, we used the best subset selection with BIC selection criterion to select a subset of predictors from our 26 predictors. We specifically chose the BIC criterion to select the best subset as it places a larger penalty on unnecessary features as our objective is to reduce the number of predictors. The resultant *BIC of the model vs the number of predictors* is shown in *Figure 3.7*. The coefficients are shown in *Figure 3.8*.

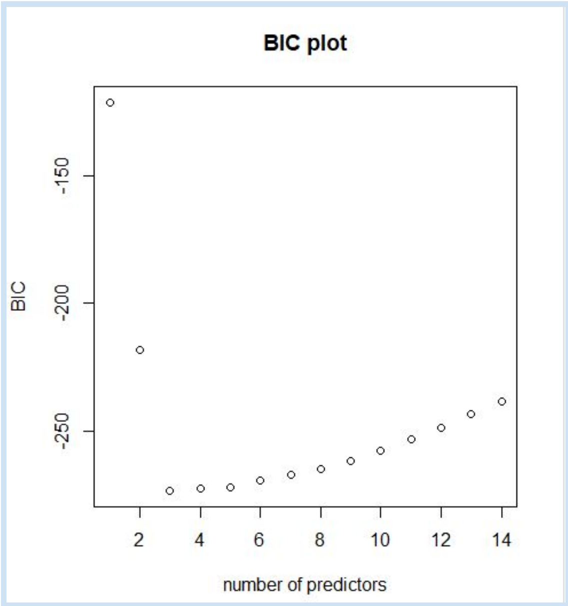


Figure 3.7: BIC Plot

```
Call:
glm(formula = CarInsurance ~ HHInsurance + PrevSucc + CallDuration,
    family = binomial, data = new_coldcall[train, ])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.7665  -0.6329   0.2862   0.6704   2.0748

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.39870    0.22071 -10.868  < 2e-16 ***
HHInsurance0  1.48586    0.19763   7.518 5.55e-14 ***
PrevSucc1     1.95980    0.23384   8.381  < 2e-16 ***
CallDuration  0.29583    0.03163   9.353  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 990.32  on 725  degrees of freedom
Residual deviance: 666.48  on 722  degrees of freedom
AIC: 674.48

Number of Fisher Scoring iterations: 5
```

Figure 3.8: Coefficients of the Best Subset Selected Model

3.3 Ridge and Lasso

Aiming to reduce the variance of the model, we used two shrinkage methods: Ridge Regression and The Lasso. Both models look to shrink the coefficient estimates to zero and we will find the best value of lambda using the cross-validation method.

3.3.1 Ridge Regression (M3)

To fit our data into a ridge regression, we placed the x variables of the trained data set into x and the y variable into y. In the case of a ridge regression, we are required to use the function *model.matrix()* and remove the y variable from the trained data set. Afterwhich, a 10-fold cross-validation error will be done in order to find out the best value of lambda which is determined by having the smallest cross-validation error. In this case, our best lambda in the model is 0.02033384, as shown in figure 3.9.

```
> bestlamb<-cv$lambda.min
> bestlamb
[1] 0.02033384
```

Figure 3.9: Best Lambda of Ridge Regression

After looking at the coefficients of the variables in the ridge regression, we find out that none of the coefficients of the 28 variables estimates have been shrunk to 0, with coefficients shown in *Figure 3.9*.

```
> coef(ridge,bestlamb)
29 x 1 sparse Matrix of class "dgCMatrix"
                                   1
(Intercept)            -1.476820e+00
Age                     1.838614e-03
Jobblue-collar         -3.869826e-01
Jobstudent              2.857873e-01
Jobtechnician          -5.178596e-01
Jobadmin.              -5.307468e-02
Jobservices            -2.301992e-01
Jobself-employed        1.143126e-01
Jobretired              1.153683e-01
Jobhousemaid           -7.929356e-01
Jobentrepreneur        -7.531241e-01
Jobunemployed           1.284677e-01
Maritalmarried         -1.109258e-01
Maritaldivorced        -7.511926e-02
Educationprimary       -5.441677e-01
Educationsecondary     -3.454464e-01
Default1               -7.444510e-01
Balance                -5.882120e-06
HHInsurance0            1.135277e+00
CarLoan1               -5.018646e-01
Communicationcellular  -2.812363e-02
NoOfContacts           -9.017545e-02
DaysPassed              6.889292e-04
PrevAttempts            4.335507e-02
Outcomeother            1.520428e-01
Outcomesuccess          8.915901e-01
CallDuration            2.234853e-01
Entrepreneur1          -7.493841e-01
PrevSucc1               8.871920e-01
```

Figure 3.10: Ridge Model (M3) Coefficients

3.3.2 The Lasso (M4)

When fitting our data set into The Lasso, we make use of the trained data set that was represented by our x and y from the ridge regression. However, in the case of the Lasso model, we set alpha to be equals to 1 instead of 0. In the Lasso model, we find out that the best lambda is 0.01035834, shown in figure 3.10, which is lower compared to the ridge regression model.

```
> bestlamb2<-cv2$lambda.min
> bestlamb2
[1] 0.01035834
```

Figure 3.11: Best Lambda for The Lasso

When comparing the coefficients of the estimated variables in the Lasso model with the ridge regression model, we observed that the Lasso model has shrunk 12 out of 28 of them to 0 which was what we aimed to achieve using the shrinkage methods. This gave the resultant coefficients as shown in *Figure 3.11*.

```
> coef(lasso,bestlamb2)
29 x 1 sparse Matrix of class "dgCMatrix"
                                1
(Intercept)           -1.68318317
Age                    .
Jobblue-collar        -0.22911840
Jobstudent             0.06697488
Jobtechnician         -0.32951677
Jobadmin.              .
Jobservices            .
Jobself-employed       .
Jobretired             .
Jobhousemaid          -0.46485684
Jobentrepreneur       -1.11038380
Jobunemployed          .
Maritalmarried        -0.02902297
Maritaldivorced        .
Educationprimary      -0.35586252
Educationsecondary    -0.28315213
Default1               .
Balance                .
HHInsurance0           1.23528454
CarLoan1              -0.41560106
Communicationcellular  .
NoOfContacts          -0.04350548
DaysPassed             .
PrevAttempts           0.02735097
Outcomeother           0.02873535
Outcomesuccess         1.64161086
CallDuration           0.25690714
Entrepreneur1          .
PrevSucc1              0.06070795
```

Figure 3.12:Model (M4) generated with Lasso Summary

We can observe that the Lasso method is better at achieving our goals of shrinking the coefficients of the estimated variables to 0.

3.4 Classification Tree (M5)

The next model we attempted was a decision tree model. This modelling technique would be simple to understand and closely resemble human decision-making. To predict whether or not a cold call would be successful, we simply follow the tree down the internal nodes, if the condition of the internal node is "no", we go to the left of the node, and right if it is "yes".

It also removes the need to create dummy variables which is advantageous for the data we have due to the presence of multiple categorical variables, each with multiple categories. We obtain the summary results:

```
> mtree<-tree(CarInsurance~.,coldcall,subset = train)
> summary(mtree)

Classification tree:
tree(formula = CarInsurance ~ ., data = coldcall, subset = train)
Variables actually used in tree construction:
[1] "CallDuration" "HHInsurance"  "DaysPassed"   "Job"          "PrevSucc"     "Age"
Number of terminal nodes:  15
Residual mean deviance:  0.7935 = 564.2 / 711
Misclassification error rate: 0.1873 = 136 / 726
```

Figure 3.13: Classification Tree Summary

There are 15 terminal nodes on our tree diagram and we see that only *CallDuration, HHInsurance, DaysPassed, Job, PrevSucc and Age* were used in the tree construction. This gives us a residual mean deviance of 0.7935 and misclassification error rate of 0.1873. Plotting the tree in *Figure 3.13*:

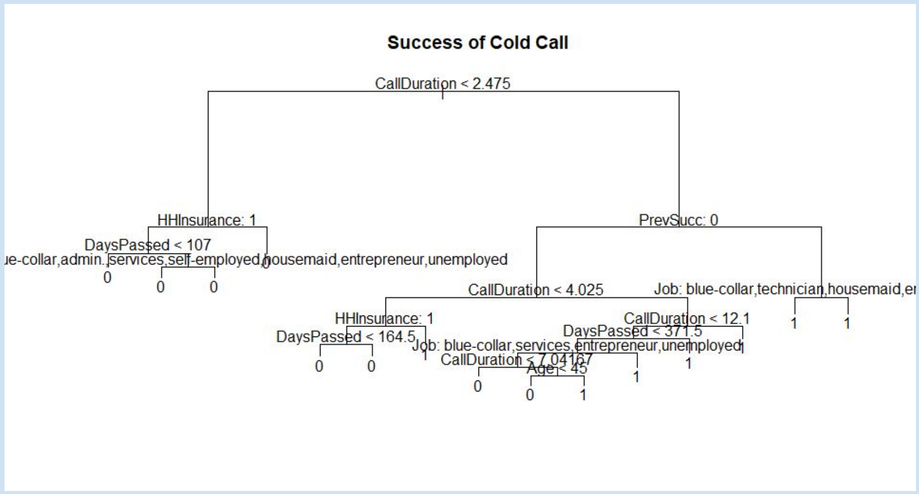


Figure 3.14: Classification Tree

To improve the model, we tried pruning the tree model to reduce the complexity of our model by removing the weakest links in the model, whilst trying to keep misclassification errors to a minimum. We obtain the result:

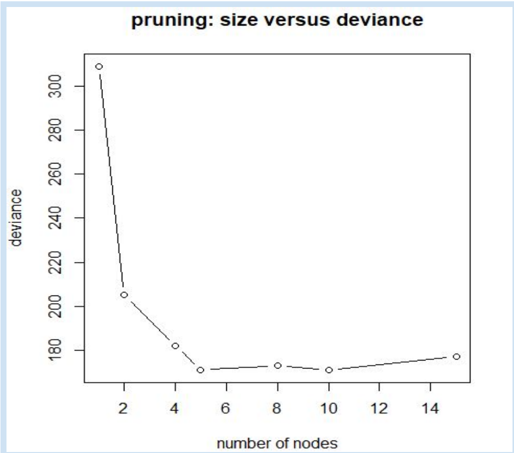


Figure 3.15: Deviance Plot

The best model thus uses only 5 terminal nodes compared to before. Thus the pruned model will look as such:

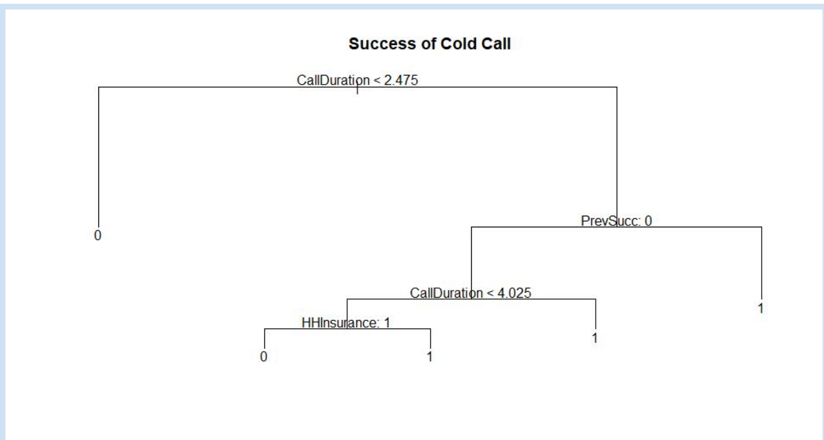


Figure 3.16: Pruned Decision Tree

In this pruned model, only *PrevSucc, CallDuration, HHInsurance* are used in the construction of the model. The variables used in the construction of the tree are similar to the variables in our best subset model. Thus, these variables are significant predictors.

To predict if a cold call was successful, we see that if *CallDuration* is less than 2.475min, then the cold call will most likely not succeed as the majority of the terminal nodes on the left are '0'. If not, we go down the tree to the right. If *PrevSucc* is '1' instead of '0', then the cold call would likely succeed as the majority of the terminal nodes on the right are '1' . The same method of following down the tree can be applied to the other internal nodes of the tree.

4. Evaluation

To compare the models, we use the model accuracy to evaluate which would be the 'best' model. We used a standard threshold of 0.5 to evaluate the models and their respective confusion matrices.

4.1. Confusion Matrix Comparison

We generated the following confusion matrices for the models, with their corresponding true positive and negative rates, overall error shown in *Table 4.1*.

Model	**GLM (M1)**	**Best Subset (M2)**	**Ridge (M3)**	**Lasso (M4)**	**Decision Tree (M5)**
Confusion Matrices	m11.pred 0 1 0 65 8 1 7 101	m22.pred 0 1 0 66 10 1 6 99	new.y prediction 0 1 0 62 13 1 10 96	new.y prediction2 0 1 0 62 12 1 10 97	prune_pred 0 1 0 47 10 1 25 99
True Positive Rate	92.66%	90.83%	88.07%	88.99%	90.83%
True Negative Rate	90.28%	91.67%	86.11%	86.11%	65.28%
Overall Error	8.29%	8.84%	12.71%	12.15%	19.34%

Table 4.1: Evaluating models

In reality, companies may use a lower threshold value than 0.5 as they may be more tolerant to false positives since the cost of cold calling would be lower than the return of a successful cold call.

5. Limitations

5.1 Reduction of Dataset

Given that the majority (77.5%) of the original data contained missing values, removing all rows with missing data may affect the prediction of models such as the classification tree that classifies observations based on the majority of each class. A possible alternative we considered in managing missing categorical data would be to use NA as a category. Additionally, datasets from different contexts would allow better tuning of the model as we can examine the performance of our model across different industries and products and obtain a threshold value that provides consistent accuracy, ensuring our model is not overfitted.

5.2 Polynomials and Interaction effect

With our high number of features present in the dataset, we did not explore the interaction effects between features nor multiple orders of polynomials as the resultant model would be even more complex and model interpretability is affected. This could be considered for future subsequent projects.

5.3 Assumptions on Profitability

In the case of profitability, we made the assumption that the risk levels of all clients are equal. In this case, profitability is only represented by whether a customer takes up an insurance after a call as a greater number of successes will directly translate to an increase in business for the company which leads to greater profits. We acknowledge the fact that bringing in customers of higher risk will also affect the profitability, hence the case for our assumption.

5.4 Possible Confounding Factors

Confounding factors may influence the outcome of the dependent variable but have not been accounted for in this research. Some possible confounding factors that may affect the success of a cold call are:

1. Household income: The higher the household income, they have a higher purchasing power and thus would more likely be able to afford the insurance since it takes up a lesser proportion of it. Job type, which had been considered in the data, alone does not explain household income.
2. Experienced car accident: One who has experienced an accident before would more likely portray risk-averse behaviour and thus more likely to purchase the car insurance. (Chiappori & Salanié, 2000)
3. Car ownership: Car ownership was measured roughly through whether clients had car insurance (HHInsurance). However this estimator does not account for car owners who own cars but do not have car insurance.

6. Conclusion

As the GLM model has the lowest total error rate, we conclude that it is the best model to use to predict the success of cold calling customers. This is probably due to lower bias in the model compared to the best subset and lower variance compared to the ridge and lasso regressions.

Therefore, our practical recommendations are as follows:

1. Firstly, callers should try to engage more with their recipients to increase the call duration for a better chance of success.
2. Callers should also not continuously approach a recipient multiple times to increase the success of cold calling.
3. Companies should also not target entrepreneurs as they may have a higher risk tolerance (Hvide, 2013) and thus more unwilling to buy insurance.
4. Customers who have a car loan may be more unwilling to buy insurance due to increased costs.
5. Subsequently, companies should target customers who are already customers of existing complementary products (PriceSpider, 2020), in this case, customers who already have household insurance will be more willing to purchase car insurance after the call.
6. Customers who have bought other products during previous marketing campaigns will also be more likely to buy after the cold call.

7. Reference List

Hvide, H. K. (2013). Risk Tolerance and Entrepreneurship. *Journal of Financial Economics*.

Jantsch, J. (2010, July 16). The Abusive Math of Cold Calling. Retrieved from https://ducttapemarketing.com/the-abusive-math-of-cold-calling/

Peake, G. (2016), "“Get busy talking or get busy dying”", *Industrial and Commercial Training*, Vol. 48 No. 1, pp. 33-37. https://doi-org.libproxy.smu.edu.sg/10.1108/ICT-07-2015-0047

Chiappori, P.-A., & Salanié, B. (2000, February). Testing for Asymmetric Information in Insurance Markets. Retrieved from http://public.econ.duke.edu/~hf14/teaching/socialinsurance/readings/fudan_hsbc/Chiappori_Salanie00(2.7).pdf

PriceSpider. (2020, March 25). The Benefits of Associating with Complementary Products. Retrieved March 27, 2020, from Adding Value: The Benefits of Associating with Complementary Products

Technical University of Munich (2017). Data Mining Cup 1. Retrieved from https://github.com/togiberlin/data_mining_cup/blob/master/dmc1/DMC1_description.pdf

8. Appendix

Appendix A: Project Codes

```
######################
# load the dataset
######################
library(readr) #load readr package
coldcall <- read_csv("C:/Users/Dehan/Desktop/IMPT/4. SMU school/4, 2019
SEM 2 MODS/3. DSA211/proj/carInsurance_train.csv",
                 col_types = cols(CallEnd = col_time(format = "%H:%M:%S"),
                                  CallStart = col_time(format = "%H:%M:%S"),
                                  CarInsurance = col_factor(levels = c()),
                                  CarLoan = col_factor(levels = c()),
                                  Communication = col_factor(levels = c()),
                                  Default = col_factor(levels = c()),
                                  Education = col_factor(levels = c()),
                                  HHInsurance = col_factor(levels = c()),
                                  Id = col_skip(),
                                  Job = col_factor(levels = c()),
                                  LastContactDay = col_skip(),
                                  LastContactMonth = col_skip(),
                                  Marital = col_factor(levels = c()),
                                  Outcome = col_factor(levels = c())))
str(coldcall) #exploratory data analysis
#################################
# Data Cleaning and Preparation
#################################
coldcall<-na.omit(coldcall) #removing rows with NA values
library(dplyr) #load dplyr package
library(lubridate) #load lubridate package
coldcall<-coldcall%>%

mutate(CallDuration=time_length(interval(coldcall$CallStart,coldcall$CallEn
d),unit = 'minute'))%>%
  select(-CallStart,-CallEnd) #Removing CallStart and CallEnd which are not
useful since we have created the column CallDuration
str(coldcall)
summary(coldcall) #examining cleaned data set
##############################
# Split into training and test set
##############################
```

```
RNGkind(sample.kind = "Rounding") #standardising RNG process
set.seed(1) #set random seed for consistency in results
train<-sample(1:nrow(coldcall),round(0.8*nrow(coldcall))) #create train data
set using 80% of original data
test<-(-train) #create test dataset using 20% of original data

################################
# Logistic Regression
################################
m1<-glm(CarInsurance~.,data = coldcall[train,],family = binomial) #fitting
the logistic model
summary(m1) #looking at the fit of the model
pvalue1 <- 1-pchisq(630.98, 699)
pvalue1 #hypo test for whether the model is a good fit at a 5% level of
significance
coef(m1)
coldcall$Entrepreneur<-as.factor(ifelse(coldcall$Job=='entrepreneur','1','0'))
#converting Entrepeneur categorical data into factors and adding the new
column to the dataset
coldcall$PrevSucc<-as.factor(ifelse(coldcall$Outcome=='success','1','0'))
#converting PrevSuccess categorical data into factors and adding the new
column to the dataset
new_coldcall<-as.data.frame(coldcall) #creating a new data frame with which
is a copy of the cleaned dataset for ease of reference
m11<-glm(CarInsurance~Entrepreneur+HHInsurance+CarLoan+NoOfContac
ts+PrevSucc+CallDuration,data = new_coldcall[train,],family = binomial)
#running only the 5% significant models
summary(m11) #examining m11 fit and coeffs
pvalue2 <- 1-pchisq(655.61, 719)
pvalue2 #hypo test for whether the model is a good fit at a 5% level of
significance
coef(m11)

# Creating the m1 confusion matrix
# 0 = Cold Call not successful, 1 = Cold Call successful
m1.prob<-predict(m1,coldcall[test,],type = 'response')
m1.pred<-rep('0',nrow(coldcall[test,]))
m1.pred[m1.prob>0.5]<-'1'
table(coldcall[test,]$CarInsurance)
table(m1.pred,coldcall[test,]$CarInsurance) #m1 confusion matrix
```

```
#creating the m11 confusion matrix
# 0 = Cold Call not successful, 1 = Cold Call successful
m11.prob<-predict(m11,new_coldcall[test,],type = 'response')
m11.pred<-rep('0',nrow(new_coldcall[test,]))
m11.pred[m11.prob>0.5]<-'1'
table(m11.pred,new_coldcall[test,]$CarInsurance) #m11 confusion matrix

##############################
#Best Subset
##############################
library(leaps) #loading the leaps package
m2<-regsubsets(CarInsurance~.,coldcall[train,],nvmax=14) #forming the
model
m2.summary<-summary(m2) #examinging model m2 fit and coefficients
plot(m2.summary$bic,main = 'BIC plot',xlab = 'number of predictors',ylab =
'BIC') #plot number of predictors vs BIC
b<-which.min(m2.summary$bic) #find the lowest BIC generated
coef(m2,b) #find model m2 coeffs using best number of predictors
m22<-glm(CarInsurance~HHInsurance+PrevSucc+CallDuration, data =
new_coldcall[train,],family = binomial) #fitting log model m22
summary(m22) #looking at model m22 fit and coefficients

#creating a confusion matrix for m22
m22.prob<-predict(m22,new_coldcall[test,],type = 'response')
m22.pred<-rep('0',nrow(new_coldcall[test,]))
m22.pred[m22.prob>0.5]<-'1'
table(m22.pred,new_coldcall[test,]$CarInsurance) #confusion matrix created

################################################
#Ridge and Lasso
###############################################
library(glmnet) #load glmnet package
x<-model.matrix(CarInsurance~.,coldcall[train,])[,-1]# Split the categorical
variables into dummy variables
y<-coldcall[train,]$CarInsurance
new.x<-model.matrix(CarInsurance~.,coldcall[test,])[,-1]
new.y<-coldcall[test,]$CarInsurance#Creation of train and test datasets for
use, each dataset is seperated into dependant varaiables and indendant
variables
```

```
ridge<-glmnet(x,y,alpha = 0, nlambda = 100,family = 'binomial') #Creating
ridge model
cv<-cv.glmnet(x,y,alpha=0, family = 'binomial') #Cross validating ridge
model for different values of lambda
plot(cv)
bestlamb<-cv$lambda.min #Choosing the lambda that minimises cross
validation errors
bestlamb
coef(ridge,bestlamb) #Using the chosen lambda, find the coefficients of the
ridge model

ridge.pred<-predict(ridge,s=bestlamb, newx = new.x,type = 'response')
prediction<-rep('0',length(new.y))
prediction[ridge.pred>0.5]<-'1'
table(prediction,new.y) #Create Ridge model Confusion Matrix

lasso<-glmnet(x,y,alpha = 1, nlambda = 100,family = 'binomial') #Create
lasso model
cv2<-cv.glmnet(x,y,alpha=1, family = 'binomial') #Cross validating for the
lasso model for different values of lambda
plot(cv2)
bestlamb2<-cv2$lambda.min #Choosing the lambda that minimises cross
validation errors
bestlamb2
coef(lasso,bestlamb2) #Using the chosen lambda, find the coefficients of the
lasso model

lasso.pred<-predict(lasso,s=bestlamb2, newx = new.x,type = 'response')
prediction2<-rep('0',length(new.y))
prediction2[lasso.pred>0.5]<-'1'
table(prediction2,new.y) #Confusion Matrix for the lasso model

################################
#Decision Tree
################################
library(tree)
mtree<-tree(CarInsurance~.,coldcall,subset = train) #Create Tree Classifier
Model
summary(mtree)
mtree
plot(mtree)
```

```
title(main = 'Success of Cold Call')
text(mtree,pretty = 0) #Plot out Tree Classifier

prune_mtree<-cv.tree(mtree,FUN = prune.misclass) #Cross Validation for
tree complexity
plot(prune_mtree$size,prune_mtree$dev,type = 'b',main = 'pruning: size
versus deviance',xlab = 'number of nodes',ylab = 'deviance')
nn<-prune_mtree$size[which.min(prune_mtree$dev)] #Finding the number of
nodes that minimise misclassification errors
nn
prune_model<-prune.misclass(mtree,best = nn) #Pruning the tree classifier
based on optimal number of nodes
plot(prune_model)
title(main = 'Success of Cold Call')
text(prune_model,pretty = 0)

tree_pred<-predict(mtree,coldcall[test,],type = 'class') #Unpruned Tree
Confusion Matrix
table(tree_pred,coldcall[test,]$CarInsurance)

prune_pred<-predict(prune_model,coldcall[test,],type = 'class') #Pruned Tree
Confusion Matrix
table(prune_pred,coldcall[test,]$CarInsurance)
```

Appendix B: Data Mining Cup

Technische Universität München

Tutorial Business Analytics

Data Mining Cup (SS 2017)

Description

This is a dataset from one bank in the United States. Besides usual services, this bank also provides car insurance services. The bank organizes regular campaigns to attract new clients. The bank has potential customers' data, and bank's employees call them for advertising available car insurance options. We are provided with general information about clients (age, job, etc.) as well as more specific information about the current insurance sell campaign (communication, last contact day) and previous campaigns (attributes like previous attempts, outcome).

You have data about 4000 customers who were contacted during the last campaign and for whom the results of campaign (did the customer buy insurance or not) are known.

Classification Task

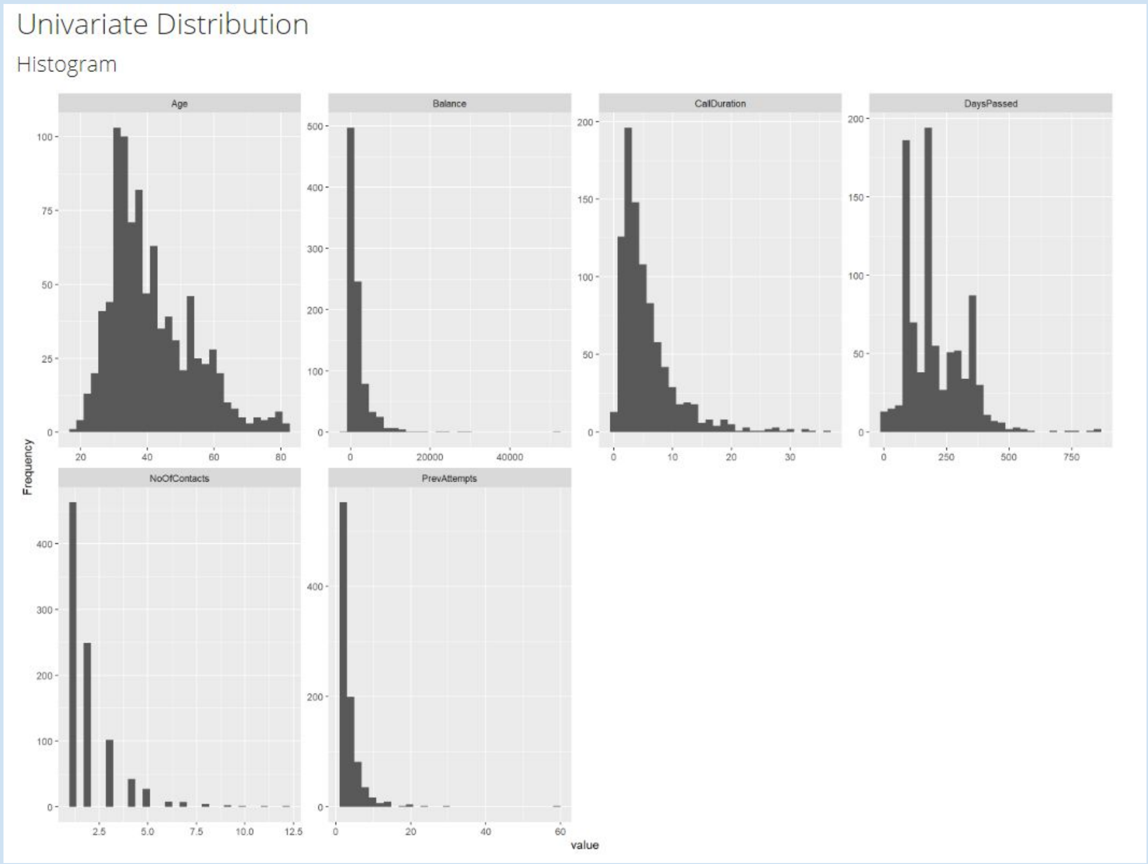
The task is to predict for 1000 customers who were contacted during the current campaign, whether they will buy car insurance or not.

Technische Universität München

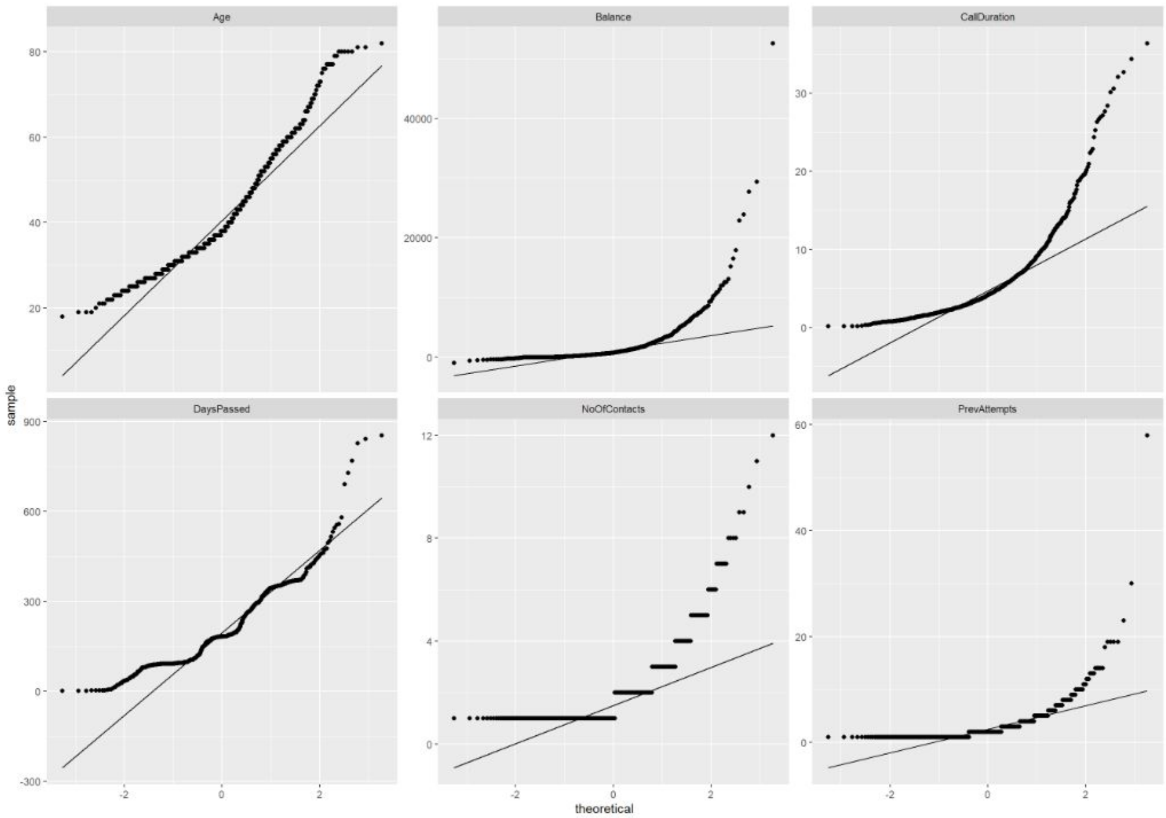
Feature Overview

Feature	Description	Example
Id	Unique ID number. Predictions file should contain this feature.	"1" ... "5000"
Age	Age of the client	
Job	Job of the client.	"admin.", "blue-collar", etc.
Marital	Marital status of the client	"divorced", "married", "single"
Education	Education level of the client	"primary", "secondary", etc.
Default	Has credit in default?	"yes" - 1,"no" - 0
Balance	Average yearly balance, in USD	
HHInsurance	Is household insured	"yes" - 1,"no" - 0
CarLoan	Has the client a car loan	"yes" - 1,"no" - 0
Communication	Contact communication type	"cellular", "telephone", "NA"
LastContactMonth	Month of the last contact	"jan", "feb", etc.
LastContactDay	Day of the last contact	
CallStart	Start time of the last call (HH:MM:SS)	12:43:15
CallEnd	End time of the last call (HH:MM:SS)	12:43:15
NoOfContacts	Number of contacts performed during this campaign for this client	
DaysPassed	Number of days that passed by after the client was last contacted from a previous campaign (numeric; -1 means client was not previously contacted)	
PrevAttempts	Number of contacts performed before this campaign and for this client	
Outcome	Outcome of the previous marketing campaign	"failure", "other", "success", "NA"
CarInsurance	Has the client subscribed a CarInsurance?	"yes" - 1,"no" - 0

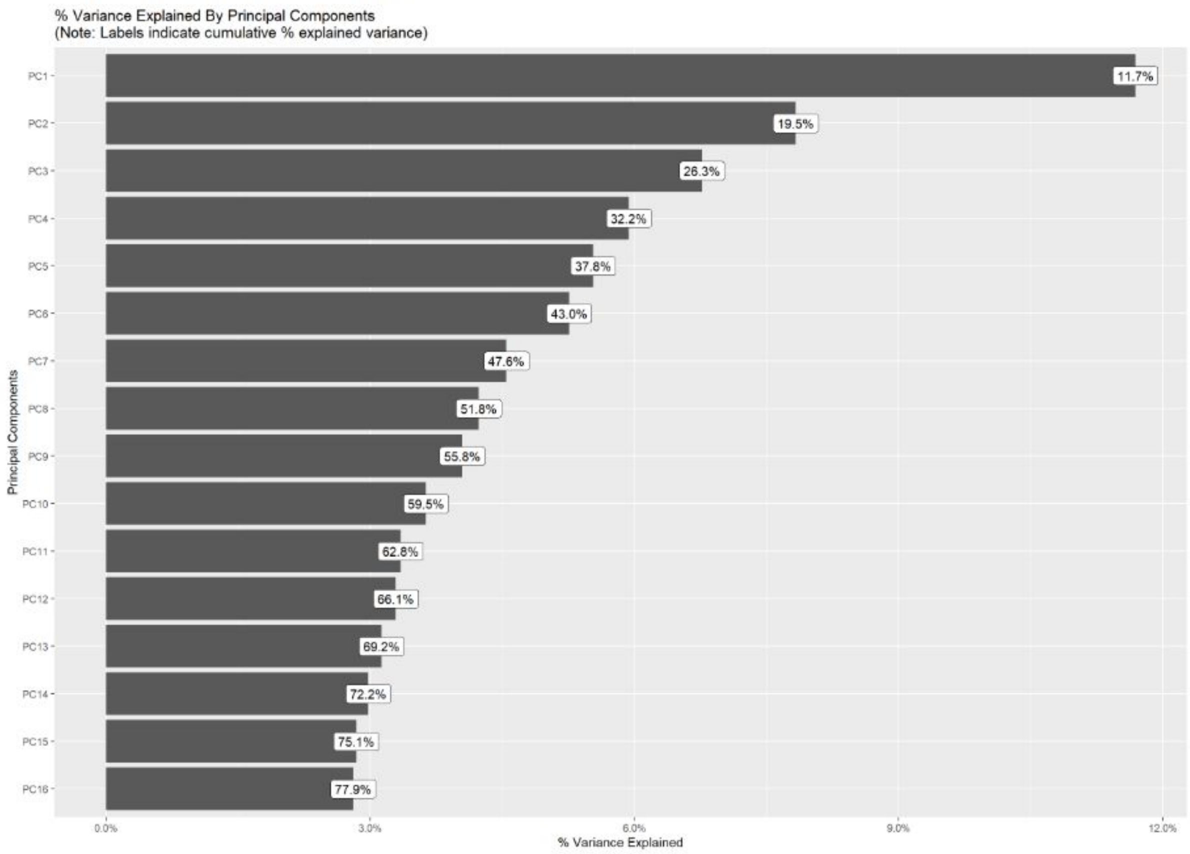
Appendix C: Exploratory Data Analysis Graphs



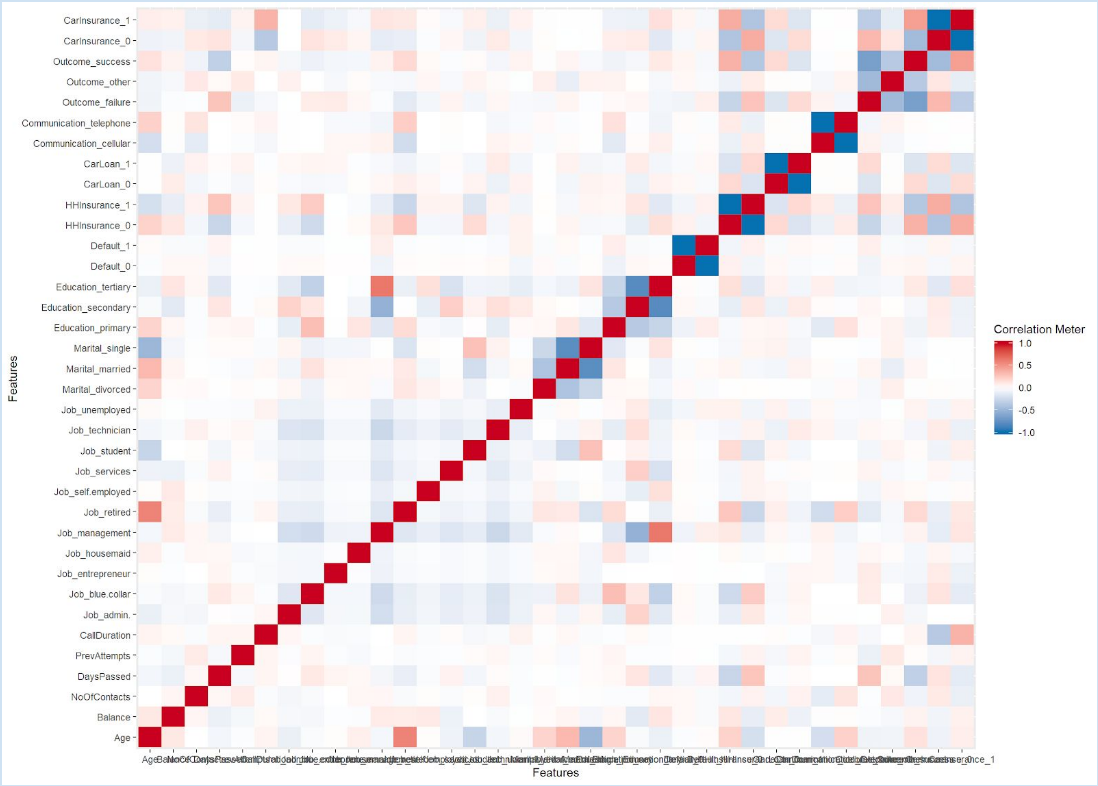
QQ Plot



Principal Component Analysis



Appendix D: Correlation Matrix



Appendix E: Ridge Regression & Lasso Plots