Introduction:

For this project, I chose an insurance-related dataset from Kaggle.com, and can be accessed with this link: https://www.kaggle.com/kondla/carinsurance.

This dataset contains information on clients/customers from one bank in the United States. Besides usual financial services, this bank also provides a car insurance service for their customers. The bank often organizes campaigns to attract new clients to this service, and its employees use their customers' data to call them to advertise their available car insurance options. One could say that this dataset contains results from cold calls. What I am trying to predict is whether or not the client being cold called will accept or decline the car insurance coverage. Predictors include the client's background information, such as their age, education level, and occupation, as well as more specific information about the bank's current and previous insurance selling campaign.

My motivation behind choosing this dataset is that I was always fascinated by movies featuring cold calling such as The Wolf of Wall Street, The Pursuit of Happyness, and The Boiler Room. The art of speech, especially public speaking, is something that I thought I have always struggled with so watching people from those movies gracefully convince someone that they have never met in their life to purchase a product/service seemed very impressive to me. If the background behind the dataset wasn't enough, the fact that it was banking/insurance related easily drew me in as well. Another reason that appealed to me about this dataset was the fact that the "y" or response variable is of a binary class. I always found that working with a categorical response variable to be more enticing than predicting something that is continuous.

As previously mentioned, the dataset was obtained from Kaggle.com, and the specific file that it is contained in is titled “carInsurance_train.csv”. There were actually two files included from Kaggle, and the other is titled “carInsurance_test.csv”. The train and test datasets both had the exact same features/variables, but there were 4000 observations in the train dataset and 1000 observations in the test dataset. The biggest difference, however, is that the 1000 entries of the response variable in the test dataset were all blank/missing. For this reason, I only used the “carInsurance_train.csv” file and not both, as it is possible to randomly split the observations into train/test purposes without having to use both of these files.

Data Cleaning & Exploratory Data Analysis:

The dataset in question has 4000 observations, with 19 variables. All of these variables along with their descriptions and a few example observations (numeric ranges are the minimum and maximum) are listed in Table 1. At a quick glance from Table 1, we can see that some of these variables can easily be dropped from the data analysis as they have little to no practical significance into predicting whether the customer being cold called will buy or decline the car insurance service. I will go over these variables in more detail below.

Table 1

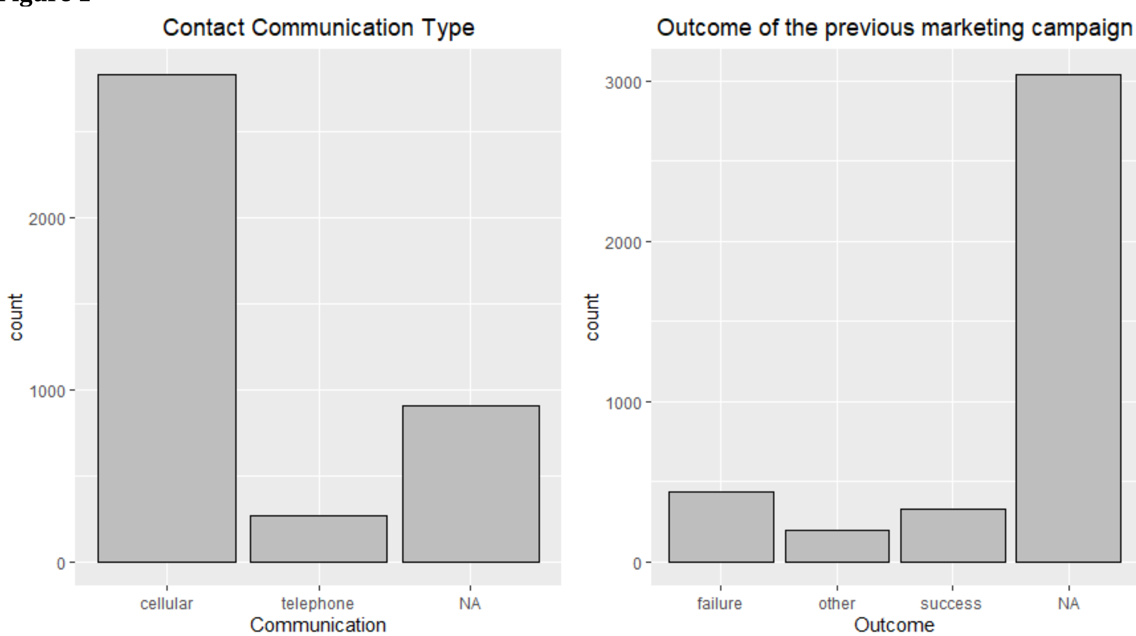
Variable	Description	Examples
Id*	Unique ID number	“1” … “4000”
Age	Age of the client	18, …, 95
Job	Job of the client	“admin.”, “blue-collar”, “management”, etc.

Marital	Marital status of the client	“single”, “married”, “divorced”
Education	Education level of the client	“primary”, “secondary”, “tertiary”
Default	Has credit in default?	“no”: 0, “yes”: 1
Balance	Average yearly balance, USD	-3058.0 … 98417.0
HHInsurance	Is household insured?	“no”: 0, “yes”: 1
CarLoan	Does client have a car loan?	“no”: 0, “yes”: 1
Communication*	Contact communication type	“cellular”, “telephone”, NA
LastContactMonth*	Month of the last contact	“jan” … “dec”
LastContactDay*	Day of the last contact	1 … 31
CallStart*	Starting time of the very last call for the client (hh:mm:ss)	09:00:00 … 17:59:58
CallEnd*	Ending time of the very last call for the client (hh:mm:ss)	09:02:20 … 18:25:31
NoOfContacts	# of times client was contacted in the current campaign	1 … 43
DaysPassed	# of days passed since last contact from a previous campaign (-1 means client was not previously contacted)	-1 … 854
PrevAttempts	# of times client was contacted before the current campaign	0 … 58
Outcome	Outcome of the **previous** marketing campaign	“failure”, “other”, “success”, NA

CarInsurance	Did the client end up purchasing car insurance?	"no": 0, "yes": 1

From Table 1, the variables indicated with a * were not considered in the data analysis. I removed Id for obvious reasons, and I believed that the month and day variables were not useful, so I discarded them. For the variable Communication and Outcome there were many missing variables, as can be seen in Figure 1. Also, I thought that whether the customer being called used a cell phone or home telephone was not going to have any influence on whether they would purchase or decline the car insurance. Next, CallStart and CallEnd were combined into 1 new variable, called "CallLength." This variable gave the total length of the very last call for the client (in minutes).

Figure 1

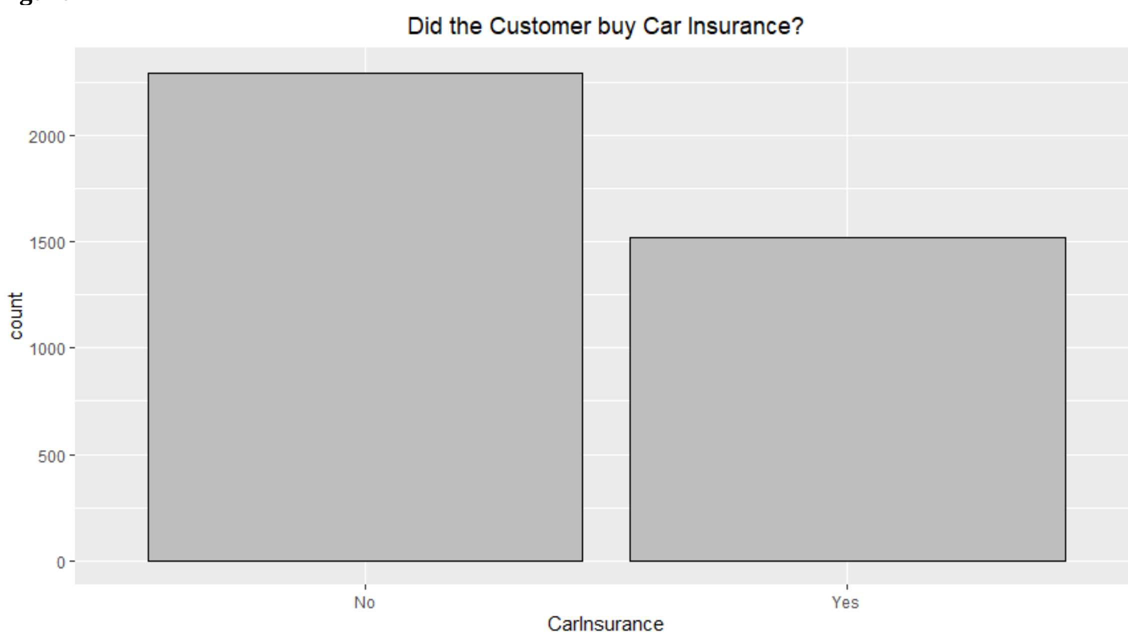


To address the issue of the over 3000 missing values for the Outcome variable, I noticed that observations with PrevAttempts equal to 0 also had a NA value for Outcome. This simply means that there was no outcome because these customers were not part of the previous campaign. Therefore, I introduced a 4th factor level into the Outcome variable and named it “noPrevious”.

The variables Education and Job still had some missing values. I decided to simply exclude those observations from the data analysis. The final cleaned dataset now has 3820 observations and 14 total variables (13 predictors), with 180 observations excluded due to missing values for the Education and Job variables. Next, I will look into each of the features in more detail, separating the categorical and continuous variables.

<u>Response Variable:</u>

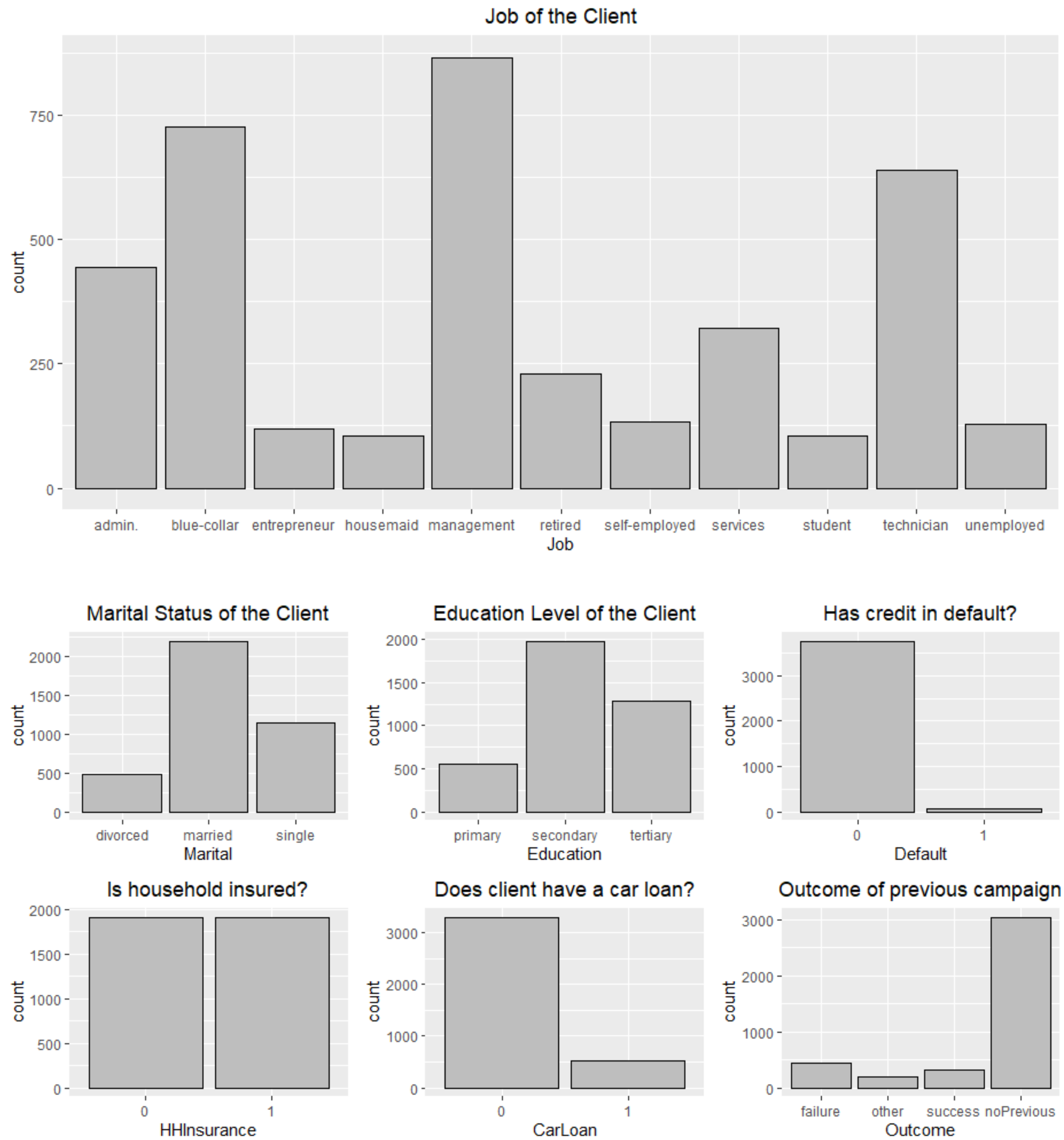
Figure 2



From Figure 2 - 2,299 clients declined the car insurance service and 1,521 chose to purchase. In other words, about 60% of total clients declined the car insurance and about 40% accepted and purchased the car insurance service.

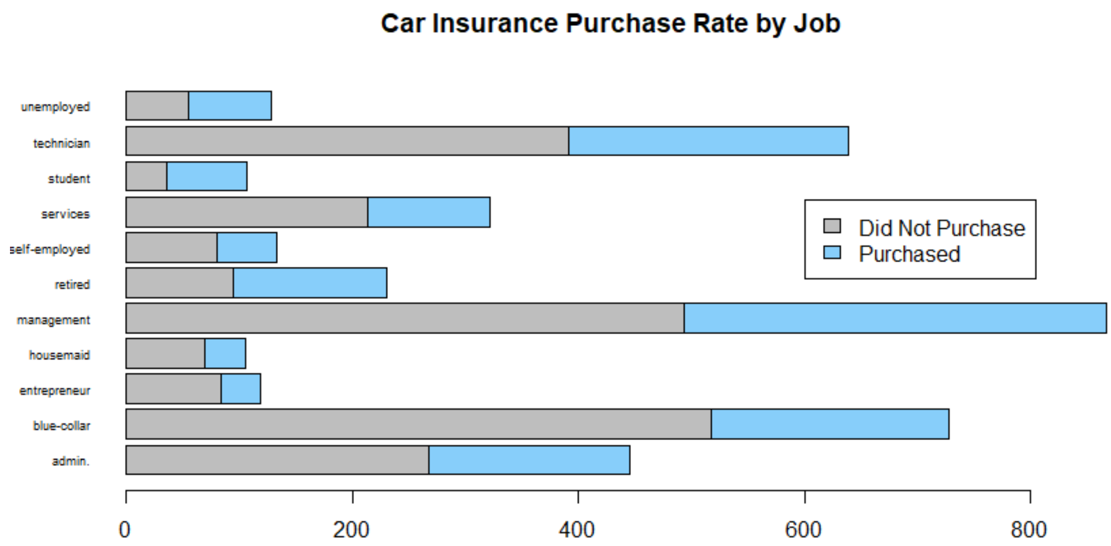
<u>Categorical Variables:</u>

Figure 3



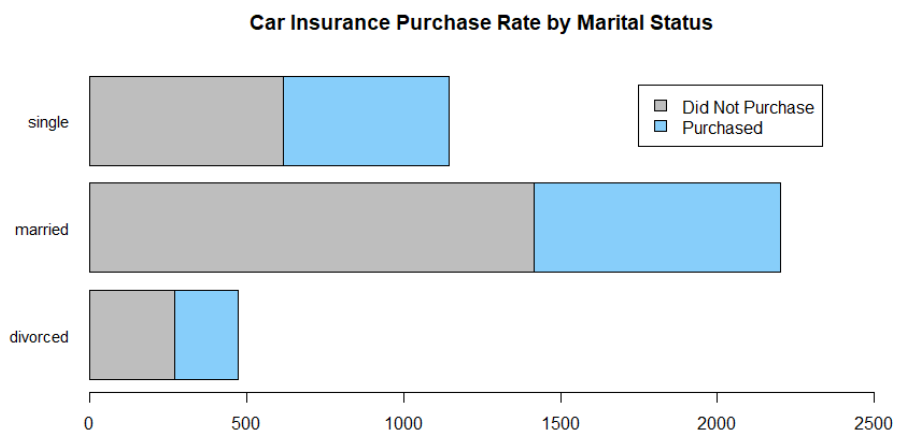
The distribution in the variables Job, Marital Status, Education, HHinsured, and Outcome variables warrant further investigation. I wanted to compare them alongside the response variable.

Figure 4



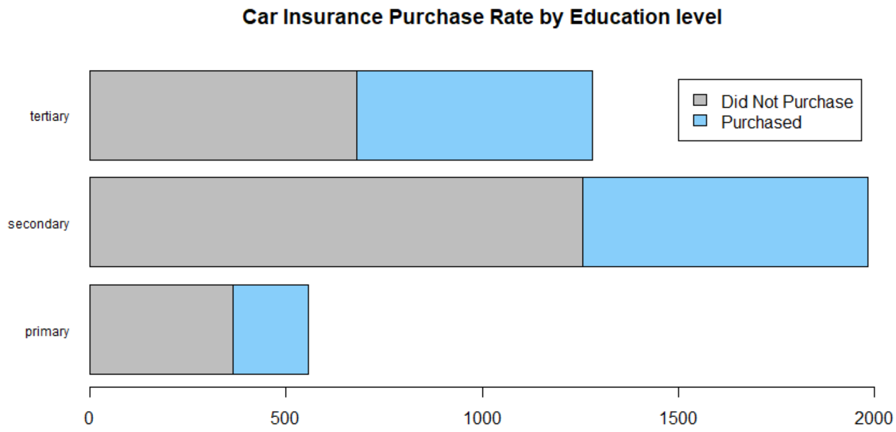
It is interesting that students, retired, and unemployed people are more likely to buy the car insurance.

Figure 5



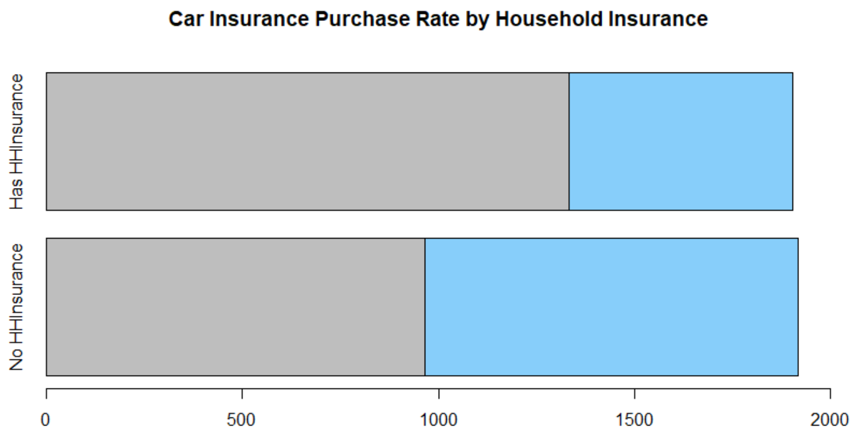
Married people are least likely to buy the car insurance, while single people are more likely to buy.

Figure 6



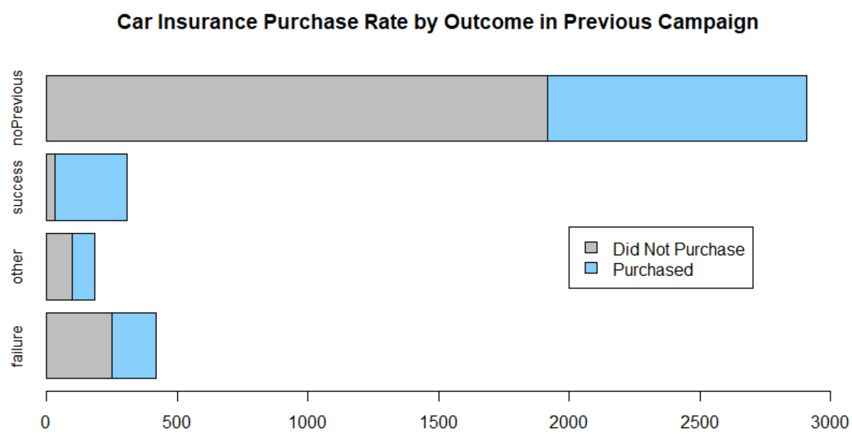
It seems like people with higher education levels are more likely to buy the car insurance.

Figure 7



If the client already had household insurance at the time of the campaign, they are far less likely to purchase car insurance. This makes sense as homeowners tend to already carry car insurance as well and would more than likely ignore a car insurance sales call.

Figure 8



It is clear to see from Figure 8 that success in the previous marketing campaign is largely associated with success in the current campaign.

Continuous/Numeric Variables

Figure 9

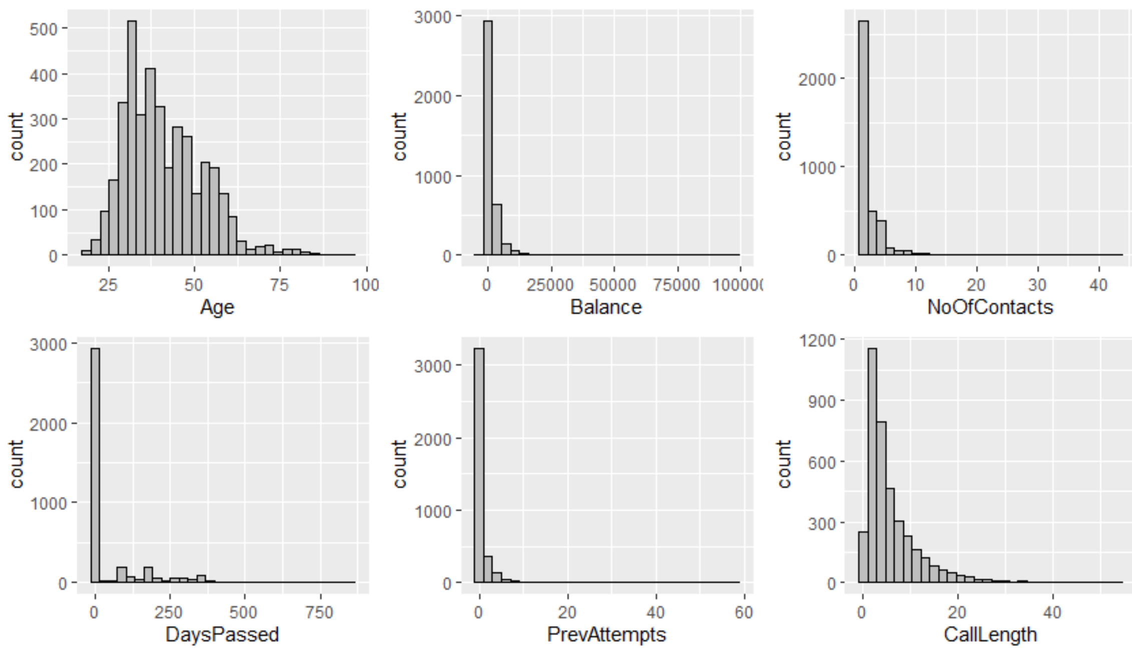
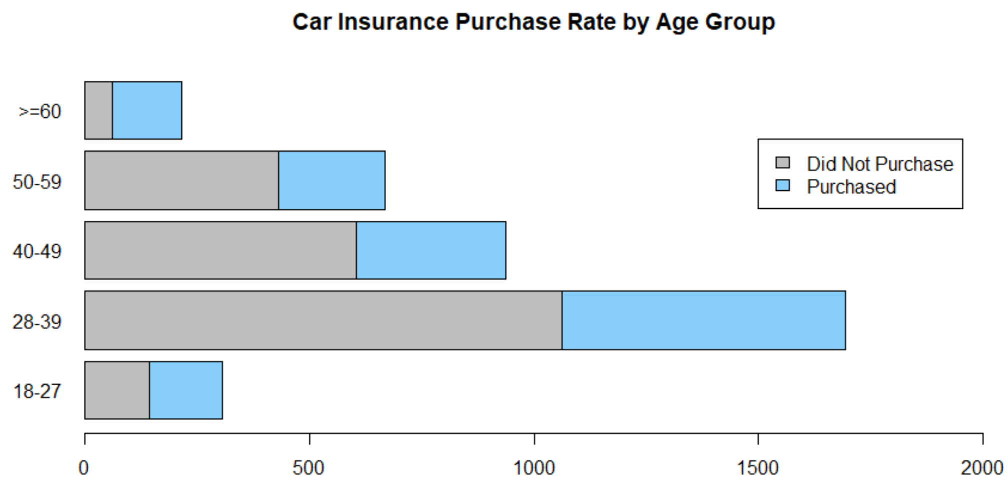


Figure 10



After exploring the numeric/continuous data, I then decided to investigate Age more thoroughly, and from Figure 10 I found that younger people and those older than 60 were more likely to purchase car insurance. This falls in line with the earlier discovery that students and people who were retired were more likely to purchase car insurance as well.

Model Selection & Goodness of Fit:

Now that we have cleaned and explored our data to be used, we can start to model it. To start, I split the dataset into two: one for training and one for testing purposes. I wanted it to be an 80-20 split, so since we had 3820 observations, 80% of those were randomly selected to be in the training dataset, and the others were put into the testing dataset. This gave us a training dataset with 3056 observations, and a testing dataset with 764 observations. Both the train and test data still maintained the ratio of roughly 60% decline and 40% purchase for the response variable.

The stepwise model selection procedure was then used with the training data, starting with the null/empty model with only the intercept term as a predictor, and we consider all 13 predictor variables. Figure 11 shows the output of the stepwise selection procedure.

Figure 11

```
Start:  AIC=4096.6
CarInsurance ~ 1

               Df Deviance    AIC
+ CallLength    1   3206.4 3210.4
+ Outcome       3   3804.9 3812.9
+ HHInsurance   1   3960.2 3964.2
+ Job          10   3976.8 3998.8
+ PrevAttempts  1   4020.8 4024.8
+ DaysPassed    1   4044.2 4048.2
+ NoOfContacts  1   4049.9 4053.9
+ Marital       2   4056.9 4062.9
+ Education     2   4062.6 4068.6
+ CarLoan       1   4065.5 4069.5
+ Default       1   4089.3 4093.3
+ Age           1   4091.1 4095.1
+ Balance       1   4092.5 4096.5
<none>              4094.6 4096.6

Step:  AIC=3210.42
CarInsurance ~ CallLength

               Df Deviance    AIC
+ Outcome       3   2856.7 2866.7
+ HHInsurance   1   3010.1 3016.1
+ Job          10   3046.6 3070.6
+ PrevAttempts  1   3112.0 3118.0
+ DaysPassed    1   3134.6 3140.6
+ CarLoan       1   3141.0 3147.0
+ Education     2   3156.8 3164.8
+ NoOfContacts  1   3169.1 3175.1
+ Marital       2   3178.5 3186.5
+ Default       1   3197.2 3203.2
+ Age           1   3203.0 3209.0
+ Balance       1   3203.1 3209.1
<none>              3206.4 3210.4
- CallLength    1   4094.6 4096.6
```

```
Step:  AIC=2866.75
CarInsurance ~ CallLength + Outcome

               Df Deviance    AIC
+ HHInsurance   1   2686.1 2698.1
+ Job          10   2733.0 2763.0
+ CarLoan       1   2814.1 2826.1
+ Education     2   2819.3 2833.3
+ Marital       2   2836.1 2850.1
+ NoOfContacts  1   2838.7 2850.7
+ Default       1   2851.8 2863.8
<none>              2856.7 2866.7
+ Age           1   2854.8 2866.8
+ Balance       1   2855.1 2867.1
+ DaysPassed    1   2855.7 2867.7
+ PrevAttempts  1   2856.0 2868.0
- Outcome       3   3206.4 3210.4
- CallLength    1   3804.9 3812.9

Step:  AIC=2698.14
CarInsurance ~ CallLength + Outcome +
 HHInsurance

               Df Deviance    AIC
+ Job          10   2623.6 2655.6
+ CarLoan       1   2656.6 2670.6
+ NoOfContacts  1   2666.5 2680.5
+ Education     2   2665.4 2681.4
+ Marital       2   2667.0 2683.0
+ Default       1   2680.0 2694.0
<none>              2686.1 2698.1
+ PrevAttempts  1   2684.7 2698.7
+ Age           1   2685.4 2699.4
+ DaysPassed    1   2686.1 2700.1
+ Balance       1   2686.1 2700.1
- HHInsurance   1   2856.7 2866.7
- Outcome       3   3010.1 3016.1
- CallLength    1   3695.5 3705.5
```

```
Step:  AIC=2655.64
CarInsurance ~ CallLength + Outcome +
 HHInsurance + Job

               Df Deviance    AIC
+ CarLoan       1   2597.7 2631.7
+ NoOfContacts  1   2604.5 2638.5
+ Education     2   2602.6 2638.6
+ Marital       2   2613.2 2649.2
+ Default       1   2619.4 2653.4
+ Age           1   2621.5 2655.5
<none>              2623.6 2655.6
+ PrevAttempts  1   2621.9 2655.9
+ DaysPassed    1   2623.4 2657.4
+ Balance       1   2623.6 2657.6
- Job          10   2686.1 2698.1
- HHInsurance    1   2733.0 2763.0
- Outcome       3   2925.7 2951.7
- CallLength    1   3653.0 3683.0

Step:  AIC=2631.69
CarInsurance ~ CallLength + Outcome +
 HHInsurance + Job + CarLoan

               Df Deviance    AIC
+ NoOfContacts  1   2579.3 2615.3
+ Education     2   2578.8 2616.8
+ Marital       2   2587.7 2625.7
+ PrevAttempts  1   2595.2 2631.2
+ Age           1   2595.3 2631.3
<none>              2597.7 2631.7
+ Default       1   2595.9 2631.9
+ Balance       1   2597.2 2633.2
+ DaysPassed    1   2597.5 2633.5
- CarLoan       1   2623.6 2655.6
- Job          10   2656.6 2670.6
- HHInsurance   1   2699.7 2731.7
- Outcome       3   2884.2 2912.2
- CallLength    1   3645.9 3677.9
```

```
Step:  AIC=2615.28
CarInsurance ~ CallLength + Outcome +
 HHInsurance + Job + CarLoan +
    NoOfContacts

               Df Deviance    AIC
+ Education     2   2560.9 2600.9
+ Marital       2   2569.5 2609.5
+ PrevAttempts  1   2576.6 2614.6
<none>              2579.3 2615.3
+ Age           1   2577.4 2615.4
+ Default       1   2577.6 2615.6
+ Balance       1   2578.9 2616.9
+ DaysPassed    1   2579.1 2617.1
- NoOfContacts  1   2597.7 2631.7
- CarLoan       1   2604.5 2638.5
- Job          10   2637.5 2653.5
- HHInsurance   1   2682.5 2716.5
- Outcome       3   2847.8 2877.8
- CallLength    1   3620.1 3654.1

Step:  AIC=2600.9
CarInsurance ~ CallLength + Outcome +
 HHInsurance + Job + CarLoan +
    NoOfContacts + Education

               Df Deviance    AIC
+ Marital       2   2554.3 2598.3
+ PrevAttempts  1   2558.0 2600.0
<none>              2560.9 2600.9
+ Default       1   2559.4 2601.4
+ Age           1   2560.2 2602.2
+ Balance       1   2560.5 2602.5
+ DaysPassed    1   2560.5 2602.5
- Education     2   2579.3 2615.3
- NoOfContacts  1   2578.8 2616.8
- CarLoan       1   2584.0 2622.0
- Job          10   2618.1 2638.1
- HHInsurance   1   2658.7 2696.7
- Outcome       3   2824.2 2858.2
- CallLength    1   3607.0 3645.0
```

```
Step:  AIC=2598.28
CarInsurance ~ CallLength + Outcome +
 HHInsurance + Job + CarLoan +
    NoOfContacts + Education + Marital

               Df Deviance    AIC
+ PrevAttempts  1   2551.4 2597.4
<none>              2554.3 2598.3
+ Default       1   2552.7 2598.7
+ DaysPassed    1   2553.9 2599.9
+ Balance       1   2554.0 2600.0
+ Age           1   2554.3 2600.3
- Marital       2   2560.9 2600.9
- Education     2   2569.5 2609.5
- NoOfContacts  1   2572.1 2614.1
- CarLoan       1   2577.3 2619.3
- Job          10   2605.2 2629.2
- HHInsurance   1   2653.7 2695.7
- Outcome       3   2816.1 2854.1
- CallLength    1   3589.5 3631.5

Step:  AIC=2597.39
CarInsurance ~ CallLength + Outcome +
 HHInsurance + Job + CarLoan +
    NoOfContacts + Education + Marital
 + PrevAttempts

               Df Deviance    AIC
<none>              2551.4 2597.4
+ Default       1   2549.9 2597.9
- PrevAttempts  1   2554.3 2598.3
+ DaysPassed    1   2551.0 2599.0
+ Balance       1   2551.1 2599.1
+ Age           1   2551.4 2599.4
- Marital       2   2558.0 2600.0
- Education     2   2566.8 2608.8
- NoOfContacts  1   2569.4 2613.4
- CarLoan       1   2575.2 2619.2
- Job          10   2602.5 2628.5
- HHInsurance   1   2651.4 2695.4
- Outcome       3   2731.7 2771.7
- CallLength    1   3585.6 3629.6
```

After 9 steps, the model with the lowest Akaike Information Criterion score of 2597.39 was the one with the 6 categorical predictors Outcome, HHInsurance, Job, CarLoan, Education level, and Marital Status and the 3 numeric/continuous predictors NoOfContacts, PrevAttempts,and CallLength. The stepwise model selection removed the variables Default, DaysPassed, Balance, & Age. I agree with removing the variable Default, DaysPassed, and Balance as according to Figure 3 and Figure 9, the distribution of the Default variable shows us that over 98% of observations did not have credit in default, and the distributions of DaysPassed and Balance are extremely right skewed. However, I was surprised that it removed Age from the reduced model, but in the later goodness of fit tests section, we showed that this was the right move.

Logistic Regression

Before proceeding with the logistic regression, I wanted to reorder the levels in a few of the categorical variables so that the base/reference level is changed. For Marital status, I changed the reference level from “divorced” to “single.” In the Job variable, the reference level of “admin” was changed to “student.” Lastly for the Outcome of the previous campaign, the reference level was changed from “failure” to “noPrevious.” After everything is setup, we then get the logistic regression model shown in Figure 12.

Figure 12

```
Call:
glm(formula = CarInsurance ~ CallLength + Outcome + HHInsurance + 
    Job + CarLoan + NoOfContacts + Education + Marital + PrevAttempts, 
    family = binomial, data = train)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-5.4153  -0.6181  -0.3275   0.5611   2.5922  

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)    
(Intercept)        -0.97625    0.31760  -3.074 0.002113 ** 
CallLength          0.33479    0.01406  23.810  < 2e-16 ***
Outcomefailure      0.54248    0.16659   3.256 0.001129 ** 
Outcomeother        0.53024    0.24102   2.200 0.027810 *  
Outcomesuccess      2.92148    0.25309  11.543  < 2e-16 ***
HHInsurance1       -1.05184    0.10765  -9.771  < 2e-16 ***
Jobadmin.          -0.73809    0.31003  -2.381 0.017280 *  
Jobblue-collar     -1.35837    0.31310  -4.338 1.44e-05 ***
Jobentrepreneur    -1.45837    0.40674  -3.586 0.000336 ***
Jobhousemaid       -1.07810    0.42151  -2.558 0.010537 *  
Jobmanagement      -1.17637    0.30754  -3.825 0.000131 ***
Jobretired         -0.11948    0.34224  -0.349 0.727003    
Jobself-employed   -0.98857    0.38848  -2.545 0.010937 *  
Jobservices        -1.14101    0.32860  -3.472 0.000516 ***
Jobtechnician      -0.87761    0.30007  -2.925 0.003448 ** 
Jobunemployed      -0.74155    0.38975  -1.903 0.057089 .  
CarLoan1           -0.78165    0.16581  -4.714 2.43e-06 ***
NoOfContacts       -0.10192    0.02599  -3.922 8.80e-05 ***
Educationsecondary  0.11226    0.16976   0.661 0.508405    
Educationtertiary   0.62892    0.20089   3.131 0.001744 ** 
Maritaldivorced    -0.12413    0.16937  -0.733 0.463625    
Maritalmarried     -0.29466    0.11682  -2.522 0.011657 *  
PrevAttempts        0.04480    0.02792   1.605 0.108562    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4094.6  on 3055  degrees of freedom
Residual deviance: 2551.4  on 3033  degrees of freedom
AIC: 2597.4

Number of Fisher Scoring iterations: 5
```

We can then change these coefficients (log odds) into odds ratios so that they can be more easily interpreted. Figure 13 displays these odds ratios alongside their 95% confidence interval.

Figure 13

	Odds ratio	2.5 %	97.5 %
(Intercept)	0.3767200	0.2025140	0.7050918
CallLength	1.3976436	1.3604096	1.4375243
Outcomefailure	1.7202629	1.2357491	2.3776162
Outcomeother	1.6993461	1.0541340	2.7160422
Outcomesuccess	18.5687466	11.4726117	31.0530968
HHInsurance1	0.3492952	0.2824390	0.4307957
Jobadmin.	0.4780250	0.2586608	0.8738606
Jobblue-collar	0.2570798	0.1382123	0.4725471
Jobentrepreneur	0.2326145	0.1036656	0.5115814
Jobhousemaid	0.3402422	0.1466264	0.7674113
Jobmanagement	0.3083955	0.1675868	0.5607399
Jobretired	0.8873815	0.4515543	1.7304195
Jobself-employed	0.3721071	0.1725088	0.7923311
Jobservices	0.3194955	0.1665987	0.6051899
Jobtechnician	0.4157743	0.2293321	0.7452230
Jobunemployed	0.4763765	0.2208043	1.0192614
CarLoan1	0.4576509	0.3289211	0.6303685
NoofContacts	0.9031001	0.8570589	0.9487421
Educationsecondary	1.1188091	0.8034425	1.5636260
Educationtertiary	1.8755889	1.2674165	2.7867010
Maritaldivorced	0.8832638	0.6328367	1.2296608
Maritalmarried	0.7447825	0.5924275	0.9366840
PrevAttempts	1.0458232	0.9932403	1.1107996

Below are some interpretations (holding all other variables constant):

- For every minute that the length of the last call increases, the odds of the client purchasing car insurance increase by about 39.8%.
- The odds for someone who had an outcome of “success” in the previous campaign to purchase car insurance are 1757% higher than for someone who was not in the previous marketing campaign.
- The odds of someone to purchase the car insurance is about 65.1% less likely if they have household insurance vs if they do not have household insurance.
- The odds of purchasing the car insurance is lower for every other job compared to students.
- The odds of someone to purchase the car insurance is 54.2% less likely if they have a car loan vs if they don’t.

- The odds of purchasing car insurance decrease (but not by much) by about 9.7% for each additional time the client was contacted during the current marketing campaign.
- The odds of purchasing the car insurance increase the higher the client's education level is. The odds for someone who has a highest education level of "tertiary" is 87.51% higher to purchase the car insurance vs someone with education level "primary."
- The odds of purchasing car insurance increase (but not by much) by about 4.6% for each additional time the client was contacted in the previous marketing campaign.

<u>Wald Test</u>

Next, we can then perform Wald Tests on each of the predictors to check and see if they are needed in the model. The Wald test is conducted by taking the ratio of the square of the regression coefficient and the square of its standard error. It tells us the statistical significance of each coefficient in our model. From the Wald test results in Figure 14, the p-values indicate that each of the predictor variables are significant in predicting the odds that a customer being cold called will purchase the car insurance service, except for Marital status. However, it was still low enough, so I did not choose to remove it from the model.

I actually fit a logistic regression with the same variables from the stepwise selection procedure, but added the Age variable. The Wald test for Age returned a p-value of 0.9911, so Age definitely needed to be excluded from the model.

Figure 14

```
> regTermTest(fit, "CallLength")
Wald test for CallLength
 in glm(formula = CarInsurance ~ CallLength + Outcome + HHInsurance +
    Job + CarLoan + NoOfContacts + Education + Marital + PrevAttempts,
    family = binomial, data = train)
F =  566.912  on  1  and  3033  df: p= < 2.22e-16
> regTermTest(fit, "Outcome")
Wald test for Outcome
 in glm(formula = CarInsurance ~ CallLength + Outcome + HHInsurance +
    Job + CarLoan + NoOfContacts + Education + Marital + PrevAttempts,
    family = binomial, data = train)
F =  44.57376  on  3  and  3033  df: p= < 2.22e-16
> regTermTest(fit, "HHInsurance")
Wald test for HHInsurance
 in glm(formula = CarInsurance ~ CallLength + Outcome + HHInsurance +
    Job + CarLoan + NoOfContacts + Education + Marital + PrevAttempts,
    family = binomial, data = train)
F =  95.46711  on  1  and  3033  df: p= < 2.22e-16
> regTermTest(fit, "Job")
Wald test for Job
 in glm(formula = CarInsurance ~ CallLength + Outcome + HHInsurance +
    Job + CarLoan + NoOfContacts + Education + Marital + PrevAttempts,
    family = binomial, data = train)
F =  5.010731  on  10  and  3033  df: p= 2.9503e-07
> regTermTest(fit, "CarLoan")
Wald test for CarLoan
 in glm(formula = CarInsurance ~ CallLength + Outcome + HHInsurance +
    Job + CarLoan + NoOfContacts + Education + Marital + PrevAttempts,
    family = binomial, data = train)
F =  22.22202  on  1  and  3033  df: p= 2.5382e-06
> regTermTest(fit, "NoOfContacts")
Wald test for NoOfContacts
 in glm(formula = CarInsurance ~ CallLength + Outcome + HHInsurance +
    Job + CarLoan + NoOfContacts + Education + Marital + PrevAttempts,
    family = binomial, data = train)
F =  15.37848  on  1  and  3033  df: p= 8.9933e-05
> regTermTest(fit, "Education")
Wald test for Education
 in glm(formula = CarInsurance ~ CallLength + Outcome + HHInsurance +
    Job + CarLoan + NoOfContacts + Education + Marital + PrevAttempts,
    family = binomial, data = train)
F =  7.654625  on  2  and  3033  df: p= 0.00048306
> regTermTest(fit, "Marital")
Wald test for Marital
 in glm(formula = CarInsurance ~ CallLength + Outcome + HHInsurance +
    Job + CarLoan + NoOfContacts + Education + Marital + PrevAttempts,
    family = binomial, data = train)
F =  3.289721  on  2  and  3033  df: p= 0.037397
> regTermTest(fit, "PrevAttempts")
Wald test for PrevAttempts
 in glm(formula = CarInsurance ~ CallLength + Outcome + HHInsurance +
    Job + CarLoan + NoOfContacts + Education + Marital + PrevAttempts,
    family = binomial, data = train)
F =  2.575038  on  1  and  3033  df: p= 0.10867
```

Next, to assess the predictive power of our model, we can use McFadden's pseudo R^2. This pseudo R^2 ranges from 0 to just under 1, and is defined as:

$$1 - \frac{log(fit)}{log(null)}$$

where log(fit) is the log likelihood value for our fitted model and log(null) is the log likelihood value for the null model.

```
fitting null model for pseudo-r2
 McFadden
0.3768885
```

A McFadden R^2 value between 0.2 and 0.4 indicates excellent model fit. Therefore, since our McFadden R^2 is 0.376885 from the R output above, we can say that the model selected is an excellent fit for predicting cold call results.

Cross Validation of Predicted Values

Table 2

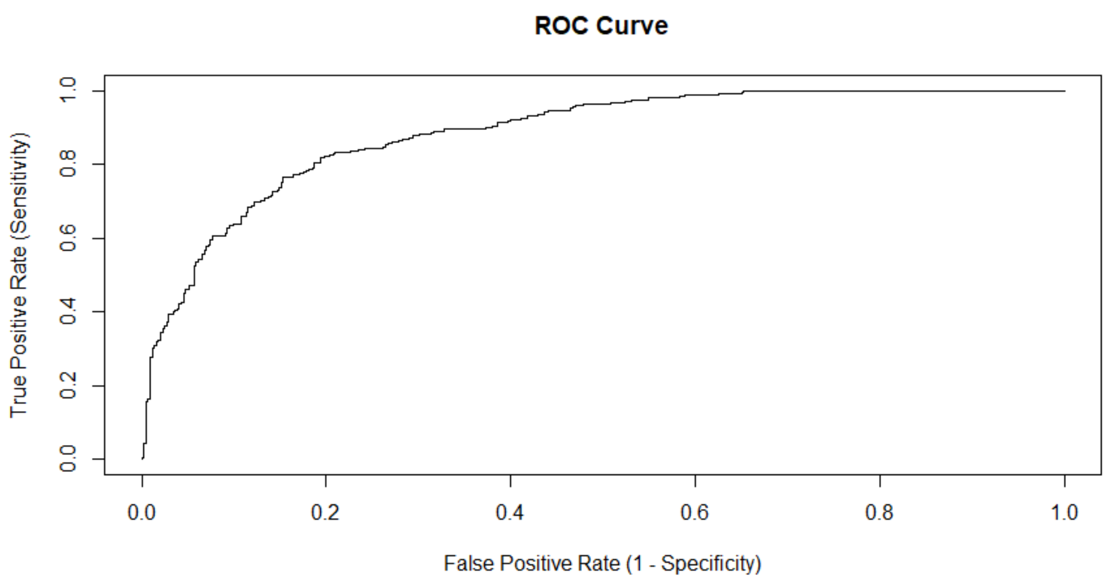
```
   Cell Contents
|-------------------------|
|                       N |
|-------------------------|


Total Observations in Table:  764


                  | test$CarInsurance
 CarInsurance.pred |         0 |         1 | Row Total |
------------------|-----------|-----------|-----------|
                0 |       393 |       106 |       499 |
------------------|-----------|-----------|-----------|
                1 |        50 |       215 |       265 |
------------------|-----------|-----------|-----------|
     Column Total |       443 |       321 |       764 |
------------------|-----------|-----------|-----------|
```

Now, we can use the test data set to see how well the fitted model does in predicting the cold call result. We will classify the observation's CarInsurance in the test data as "1" if the fitted logistic regression equation returns anything greater than or equal to 0.5 and classify as "0" if less than 0.5. Using this classification rule, we obtain the confusion matrix shown in Table 2. We can then use this confusion matrix to obtain the overall accuracy, sensitivity (true positive rate), and specificity (true negative rate). Our overall test accuracy is (393+215)/764 = 79.58%, sensitivity is 215/321 = 66.98 %, and specificity is 393/443 = 88.71%.

ROC Curve:

The Receiver Operator Characteristic curve plots (1 – Specificity) on the x-axis against Sensitivity on the y-axis and shows the trade-off between the rate of correct predictions with the rate of incorrect predictions. We are interested in the area under the ROC curve, which ranges from 0.5 to 1 with higher values indicating that the fitted model is great at distinguishing between the 2 classes of our response variable.

Figure 15



From the ROC curve in Figure 15, the area under the curve (AUC) is 0.884032, indicating that the model does a good job at discriminating between cold calls that resulted in the customer purchasing car insurance and those that resulted in the customer declining.

K-Nearest Neighbors

Next, I wanted to use the k-nearest neighbors classification algorithm on our model and see whether or not it would perform better than the logistic regression. To start, we need to normalize all of our continuous/numeric variables so that they are all on the same "scale." The scale() function in base R easily does that for us. If we did not do this, then the results would not be practical. This is because the K-NN classification algorithm relies on distances, and if any of our numeric variables were on a drastically different scale from each other, it would be problematic.

After normalizing our 3 continuous variables, we needed to create dummy variables for our categorical variables that were not already coded in as "0s" and "1s" as the K-NN algorithm only takes numeric input. Here, we can make use of the dummy.code() function from the "psych" library. Next, we needed to choose our value for *k*. I decided to go with the square root of the total observations in the training data, rounded up. This gave us $k=\sqrt{3056}=55.28=56$. The resulting confusion matrix from the k-nearest neighbors classification algorithm is given in Table 3.

Table 3

```
   Cell Contents
|-------------------------|
|                       N |
|-------------------------|


Total Observations in Table:  764


                                    | class_comparison$Observed
class_comparison$`KNN Prediction`  |        no |       yes | Row Total |
-----------------------------------|-----------|-----------|-----------|
                                no |       402 |       125 |       527 |
-----------------------------------|-----------|-----------|-----------|
                               yes |        41 |       196 |       237 |
-----------------------------------|-----------|-----------|-----------|
                      Column Total |       443 |       321 |       764 |
-----------------------------------|-----------|-----------|-----------|
```

The resulting total accuracy is therefore (402+196)/764 = 78.27%, the sensitivity is 196/321 = 61.06% and specificity = 402/443 = 90.74%. The k-nearest neighbors algorithm resulted in a lower overall test accuracy and significantly lower sensitivity, but gave a slightly better specificity compared to the logistic regression. From these results, I would deem that logistic regression performs better than K-NN.

Appendix (R Code):

```
library(lubridate)
library(ggplot2)
library(Rmisc)
library(lmtest)
library(caret)
library(pscl)
library(survey)
library(ROCR)
library(psych)
library(class)
library(dplyr)
library(gmodels)
library(e1071)

# Read in and clean the data
setwd("C:/Users/tzhan/Google Drive/GSU Graduate School/STAT 8820 Research")
raw_data <- read.csv("carInsurance_train.csv")
raw_data$Id <- NULL
raw_data$LastContactDay <- NULL
raw_data$LastContactMonth <- NULL

x1 <- ggplot(raw_data, aes(x = Communication)) + ggtitle("Contact Communication
Type") +
  theme(plot.title = element_text(hjust = 0.5)) + geom_bar(colour="black", fill="gray")
x2 <- ggplot(raw_data, aes(x = Outcome)) + ggtitle("Outcome of the previous marketing
campaign") +
  theme(plot.title = element_text(hjust = 0.5)) + geom_bar(colour="black", fill="gray")
multiplot(x1, x2, layout = matrix(c(1,2), nrow = 1))

# "Outcome" variable
raw_data$Outcome <- addNA(raw_data$Outcome)
levels(raw_data$Outcome) <- c('failure', 'other', 'success', 'noPrevious')
raw_data[raw_data['PrevAttempts']==0,'Outcome']='noPrevious'
sum(raw_data['DaysPassed']==-1) == sum(raw_data['Outcome']=='noPrevious')

raw_data$Communication <- NULL

# Remove missing data
not_missing <- (apply(is.na(raw_data), 1, sum) == 0)
new_data <- raw_data[not_missing,]
```

```
# Combine time of Call Start & End into one variable - Total Length of the Call (in minutes):
CallStart <- hms(as.character(new_data$CallStart))
CallEnd <- hms(as.character(new_data$CallEnd))
new_data$CallLength <- as.numeric(as.duration(CallEnd - CallStart), "minutes")
new_data$CallEnd <- NULL
new_data$CallStart <- NULL

# Reorder the data so that the response variable is the last column
new_data <- new_data[, c(1,2,3,4,5,6,7,8,9,10,11,12,14,13)]

# Data Exploration:

# Response Variable:
new_data$CarInsurance[new_data$CarInsurance==0] <- "No"
new_data$CarInsurance[new_data$CarInsurance==1] <- "Yes"
ggplot(new_data, aes(x = CarInsurance)) + ggtitle("Did the Customer buy Car Insurance?") +
  theme(plot.title = element_text(hjust = 0.5)) + geom_bar(stat = "count", color="black", fill="gray")

# Recode response variable into a binary class
new_data$CarInsurance[new_data$CarInsurance=='No'] <- 0
new_data$CarInsurance[new_data$CarInsurance=='Yes'] <- 1
new_data$CarInsurance <- as.factor(new_data$CarInsurance)
str(new_data)
nrow(new_data[new_data$CarInsurance==0,]) #2299 customers did not purchase
nrow(new_data[new_data$CarInsurance==1,]) #1521 customers purchased

# Categorical variables

# Convert from integer to factor:
new_data$Default <- as.factor(new_data$Default)
new_data$HHInsurance <- as.factor(new_data$HHInsurance)
new_data$CarLoan <- as.factor(new_data$CarLoan)

# Relevel Job, Marital, and Outcome variables :
new_data$Job <- relevel(new_data$Job,'student')
new_data$Marital <- relevel(new_data$Marital,'single')
new_data$Outcome <- relevel(new_data$Outcome,'noPrevious')
str(new_data)

xtabs(~CarInsurance + Job,new_data)
xtabs(~CarInsurance + Marital,new_data)
xtabs(~CarInsurance + Education,new_data)
xtabs(~CarInsurance + Default,new_data)
```

```
xtabs(~CarInsurance + HHInsurance,new_data)
xtabs(~CarInsurance + CarLoan,new_data)
xtabs(~CarInsurance + Outcome,new_data)

p2 <- ggplot(new_data, aes(x = Job)) + ggtitle("Job of the Client") +
  theme(plot.title = element_text(hjust = 0.5)) + geom_bar(colour="black", fill="gray")
p3 <- ggplot(new_data, aes(x = Marital)) + ggtitle("Marital Status of the Client") +
  theme(plot.title = element_text(hjust = 0.5)) + geom_bar(colour="black", fill="gray")
p4 <- ggplot(new_data, aes(x = Education)) + ggtitle("Education Level of the Client") +
  theme(plot.title = element_text(hjust = 0.5)) + geom_bar(colour="black", fill="gray")
p5 <- ggplot(new_data, aes(x = Default)) + ggtitle("Has credit in default?") +
  theme(plot.title = element_text(hjust = 0.5)) + geom_bar(colour="black", fill="gray")
p7 <- ggplot(new_data, aes(x = HHInsurance)) + ggtitle("Is household insured?") +
  theme(plot.title = element_text(hjust = 0.5)) + geom_bar(colour="black", fill="gray")
p8 <- ggplot(new_data, aes(x = CarLoan)) + ggtitle("Does client have a car loan?") +
  theme(plot.title = element_text(hjust = 0.5)) + geom_bar(colour="black", fill="gray")
p9 <- ggplot(raw_data, aes(x = Outcome)) + ggtitle("Outcome of previous campaign") +
  theme(plot.title = element_text(hjust = 0.5)) + geom_bar(colour="black", fill="gray")
multiplot(p3, p4, p5, p7, p8, p9, layout = matrix(c(1,2,3,4,5,6), nrow = 2, byrow=T))

barplot(table(CarInsurance, Job),horiz=T,las=1,cex.names=0.55,col=c("gray",
"lightskyblue"),
        main='Car Insurance Purchase Rate by Job')
legend(600,9.5, legend = c("Did Not Purchase", "Purchased"), fill=c("gray",
"lightskyblue"))

barplot(table(CarInsurance, Marital),horiz=T,las=1,cex.names=0.95,col=c("gray",
"lightskyblue"),
        main='Car Insurance Purchase Rate by Marital Status',xlim=c(0, 2500))
legend(1750,3.5, legend = c("Did Not Purchase", "Purchased"), fill=c("gray",
"lightskyblue"))

barplot(table(CarInsurance, Education),horiz=T,las=1,cex.names=0.75,col=c("gray",
"lightskyblue"),
        main='Car Insurance Purchase Rate by Education level',xlim=c(0, 2000))
legend(1500,3.5, legend = c("Did Not Purchase", "Purchased"), fill=c("gray",
"lightskyblue"))

barplot(table(CarInsurance, HHInsurance),horiz=T,las=0,names.arg = c('No
HHInsurance', 'Has HHInsurance'),
        col=c("gray", "lightskyblue"),main='Car Insurance Purchase Rate by Household
Insurance',xlim=c(0, 2000))

barplot(table(CarInsurance, Outcome),horiz=T,las=0,cex.names=0.9,col=c("gray",
"lightskyblue"),
```

```
      main='Car Insurance Purchase Rate by Outcome in Previous Campaign',xlim=c(0,
3000))
legend(2000,2.5, legend = c("Did Not Purchase", "Purchased"), fill=c("gray",
"lightskyblue"))

# Continuous variables
p1 <- ggplot(new_data, aes(x = Age)) + theme(plot.title = element_text(hjust = 0.5)) +
  geom_histogram(colour="black", fill="gray")
p6 <- ggplot(new_data, aes(x = Balance))  + theme(plot.title = element_text(hjust = 0.5))
+
  geom_histogram(colour="black", fill="gray")
p10 <- ggplot(new_data, aes(x = NoOfContacts)) + theme(plot.title = element_text(hjust
= 0.5)) +
  geom_histogram(colour="black", fill="gray")
p11 <- ggplot(new_data, aes(x = DaysPassed)) + theme(plot.title = element_text(hjust =
0.5)) +
  geom_histogram(colour="black", fill="gray")
p12 <- ggplot(new_data, aes(x = PrevAttempts)) + theme(plot.title = element_text(hjust =
0.5)) +
  geom_histogram(colour="black", fill="gray")
p13 <- ggplot(new_data, aes(x = CallLength)) + theme(plot.title = element_text(hjust =
0.5)) +
  geom_histogram(colour="black", fill="gray")
multiplot(p1, p6, p10, p11, p12, p13, layout = matrix(c(1,2,3,4,5,6), nrow = 2, byrow=T))

attach(new_data)
# Look at Age Groups in more detail:
AgeGroup <- 0 # 0 = 18-27, 1 = 28-39, 2 = 40-49, 3 = 50-59, 4 = >=60
for (i in 1:dim(new_data)[1]){
  if (Age[i] <= 27)
    AgeGroup[i] <- 0
  else if (Age[i] > 27 & Age[i] <= 39)
    AgeGroup[i] <- 1
  else if (Age[i] > 39 & Age[i] <= 49)
    AgeGroup[i] <- 2
  else if (Age[i] > 49 & Age[i] <= 59)
    AgeGroup[i] <- 3
  else if (Age[i] > 59)
    AgeGroup[i] <- 4
}
barplot(table(CarInsurance, AgeGroup),horiz=T,las=1,col=c("gray", "lightskyblue"),
main='Car Insurance Purchase Rate by Age Group',
      names.arg=c('18-27', '28-39', '40-49', '50-59', '>=60'),xlim = c(0,2000))
legend(1500,5, legend = c("Did Not Purchase", "Purchased"), fill=c("gray",
"lightskyblue"))
```

```
# Data Analysis:
# Separate into train and test data

#set.seed(8820)
train_ind <- sample(nrow(new_data), 3056)
set.seed(8820)

train <- new_data[train_ind,]
test <- new_data[-train_ind,]
nrow(train[train$CarInsurance==0,]) #1856 cold calls failed in training data
nrow(train[train$CarInsurance==1,]) #1200 cold calls succeeded in training data

nrow(test[test$CarInsurance==0,]) #443 cold calls failed in testing data
nrow(test[test$CarInsurance==1,]) #321 cold calls accepted in testing data

str(train)
str(test)
# ---Model selection---
# Stepwise Model Selection / Logistic Regression
fit.null <- glm(CarInsurance ~ 1, data = train, family = binomial)
fit.full <- glm(CarInsurance ~ ., data = train, family = binomial)
select <- step(fit.null, scope = list(lower = fit.null, upper = fit.full), direction = "both")

# Removed "Default", "DaysPassed", "Balance" & "Age"
fit <- glm(CarInsurance ~ CallLength + Outcome + HHInsurance + Job + CarLoan +
NoOfContacts + Education + Marital +
        PrevAttempts, data = train, family = binomial)
fit_withAge <- glm(CarInsurance ~ CallLength + Outcome + HHInsurance + Job +
CarLoan + NoOfContacts + Education + Marital +
        PrevAttempts + Age, data = train, family = binomial)
summary(fit)
# Convert to odds ratios:
exp(coef(fit))
exp(cbind("Odds ratio"=coef(fit), confint(fit)))

# Goodness of Fit

# Wald Test:
regTermTest(fit, "CallLength")
regTermTest(fit, "Outcome")
regTermTest(fit, "HHInsurance")
regTermTest(fit, "Job")
regTermTest(fit, "CarLoan")
regTermTest(fit, "NoOfContacts")
regTermTest(fit, "Education")
regTermTest(fit, "Marital")
```

```
regTermTest(fit, "PrevAttempts")
regTermTest(fit_withAge, "Age") # p-value of 0.9911

# McFadden's pseudo R^2: = 0.3768885
pR2(fit)[4]

# Cross Validation
pred  <-  predict(fit, newdata=test, type="response")
CarInsurance.pred <- NULL
for (i in 1:length(pred)){ # Classify as success if >=0.5, and failure if <0.5
  if(pred[i] >= 0.5){
    CarInsurance.pred[i] <- 1}
  else{
    CarInsurance.pred[i] <- 0}
}
CarInsurance.pred <- as.factor(CarInsurance.pred)
#confusionMatrix(data=CarInsurance.pred, test$CarInsurance)
CrossTable(x = CarInsurance.pred, y = test$CarInsurance,
           prop.chisq=F, prop.c = F, prop.r = F, prop.t = F)
# Accuracy = (393+215)/764 = 0.7958, Sensitivity = 215/321 = 0.6698, Specificity =
393/443 = 0.8871

# ROC curve and AUC
pred_ROC <- prediction(pred, test$CarInsurance)
perf <- performance(pred_ROC, measure = "tpr", x.measure = "fpr")
plot(perf, main = "ROC Curve", xlab = "False Positive Rate (1 - Specificity)", ylab =
"True Positive Rate (Sensitivity)")

auc <- performance(pred_ROC, measure = "auc")
auc <- auc@y.values[[1]]
auc # = 0.884032

# ---K-nearest neighbors---
data_class <- new_data
data_class$Age <- NULL
data_class$Default <- NULL
data_class$DaysPassed <- NULL
data_class$Balance <- NULL
data_class$CarInsurance <- as.factor(ifelse(data_class$CarInsurance == 0, "no", "yes"))

CarInsurance_class <- data_class %>% select(CarInsurance) # Extract CarInsurance
variable from dataset, then remove it
data_class$CarInsurance <- NULL

# Normalize Continuous Variables:
```

```
data_class[, c('NoOfContacts','PrevAttempts','CallLength')] <- scale(data_class[,
c('NoOfContacts','PrevAttempts','CallLength')])

# Create Dummy Variables for all Categorical Variables not already coded as 0s & 1s:
data_class <- cbind(data_class, as.data.frame(dummy.code(data_class$Job)),
as.data.frame(dummy.code(data_class$Marital)),
                    as.data.frame(dummy.code(data_class$Education)),
as.data.frame(dummy.code(data_class$Outcome)))

# Remove original variables that had to be dummy coded:
data_class$Job <- NULL
data_class$Marital <- NULL
data_class$Education <- NULL
data_class$Outcome <- NULL

# Split into training and testing sets
train_class <- data_class[train_ind,]
test_class <- data_class[-train_ind,]

# Split CarInsurance into training and test sets using the same partition as above:
CarInsurance_train <- CarInsurance_class[train_ind, ]
CarInsurance_test <- CarInsurance_class[-train_ind, ]

train_class %>% nrow %>% sqrt %>% ceiling # k = 56
CarInsurance_pred_knn <- knn(train = train_class, test = test_class, cl =
CarInsurance_train, k=56)

CarInsurance_test <- data.frame(CarInsurance_test)
class_comparison <- data.frame(CarInsurance_pred_knn, CarInsurance_test)
names(class_comparison) <- c("KNN Prediction", "Observed")
CrossTable(x = class_comparison$`KNN Prediction`, y = class_comparison$Observed,
prop.chisq=F, prop.c = F, prop.r = F, prop.t = F)
# Accuracy = (400+184)/764 = 0.7644, Sensitivity = 184/321 = 0.5732, Specificity =
400/443 = 0.9029
```