# ETC3250 Project

*Elizaveta Koshenko 25117106, Eileen Dzhumasheva 27803376, Eshan Gaindhar 26003155, Daniel Barrett 25981803, Jeremy Forbes 24251038, Jacky Yang 25097644*

*17/05/2018*

## Contents

# 1 Introduction

## 1.1 Background and Motivation

Analysing consumer data can open up an endless world full of numerous possibilities for companies across many different industries. This can help companies utilise resources in a more efficient manner in order to improve performance. The banking sector in particular has realised the importance of this, and banks now enlist consultants to uncover consumer trends in collected data.[1] In this project, we have been given a data set of past marketing campaigns of a banking institution, that contains customer and campaign characteristics. Our goal is to use this information, identify what types of customers are most likely to subscribe to a term deposit, and in doing so, be able to roll out a new marketing campaign that is more efficient and effective. A good predictive model will allow the bank to focus resources at obtaining the most interested customers, rather than using a blind, all-round approach that would also involve calling up uninterested customers.

The data set is related to direct marketing campaigns of a banking institution, which was based on multiple phone calls to 30,436 prospective clients. The data set contains fourteen explanatory variables including *age*, *job*, *education*, *default*, *housing*, *loan*, *contact*, *month*, *week day*, *campaign*, *pdays*, *previous* and *poutcome*. These are both continuous and multivalued categorical variables. The dependent binary variable, *y*, defines whether or not a client subscribed to a bank deposit. Further details about the data set as well as some preliminary analysis are included in Appendix A.

# 2 Methodology

## 2.1 Data issues

### 2.1.1 Missing Data

Figure 2 in appendix A illustrates the maginitude of "unknown" responses and their respective locations in the data set. The existence of missing data causes problems in many algorithms and may blur the real pattern hidden in the data, making it more difficult to extract information. Three methods of dealing with the missing data are explored.

The first method is complete case analyses. However, dropping the observations with missing data will reduce the amount of data we have significantly, and if the data is not missing at random, this can produce bias in the model.

The second method is recoding the missing values as a non-response. Due to the private nature of the the information being asked, we suspect there is a pattern in people not responding to questions, such that the data are not missing at random–a response of "unknown" is treated as a legitimate response in analysis.

The third method involves imputation using the MICE package. In the preliminary stages we compared the performance of the dataset with unknowns and the imputed dataset using 10-fold cross validation on a logistic model with all variables, and using the mean absolute error as the evaluation metric. There did not appear to be any improvement in performance when using the imputed data set. We suspect this is because the MICE imputation method makes the assumption that data is missing at random. As a result, we proceeded with the second method, where the non-responses were treated as their own category in the analysis.

### 2.1.2 Imbalanced data

Table 1: Counts and proportions of classes in data

| | Count | Percentage, % |
|---|---|---|
| Subscribed | 2,342 | 8 |
| Did not subscribe | 28,094 | 92 |
| Total | 30,436 | 100 |

It is clear from table 2 that the binary variable y is imbalanced, with clients who subscribed to bank term deposits being underrepresented in the training set. This is expected as bank term deposits are not favoured due to their lack of liquidity.

[1] http://www.afr.com/technology/banks-planning-big-data-deals-to-target-customers-20160521-gp0pnq

This leads to a class imbalance problem, making classification difficult as standard classifiers are driven by overall accuracy, hence the minority class, $y = 1$, may be ignored.

## 2.2 Evaluation Method

### 2.2.1 Log loss

The classification evaluation metric used in the remainder of our analysis is log loss. Log loss evaluates a model in terms of the probabilistic weight assigned to each of the corresponding class predictions.

$$LogLoss = -\frac{1}{N}\sum_{i=1}^{n}[y_i log(\hat{p}_i) + (1 - y_i) log(1 - \hat{p}_i)]$$

Log loss is minimised when $y_i = 1$ and the predicted probability of the model is large or conversely when $y_i = 0$ and the predicted probability for the model is small. A lower log loss generally indicates a better fit for a model. A potential disadvantage of using the log loss as an evaluation metric lies in its non-robustness to class imbalances: since there are very few $y_i = 1$ outcomes in the training data, a model can give lower probabilistic weight to the majority of the clients who will subscribe to a term deposit and still appear to have a relatively small log loss score.

## 2.3 Model selection

The main candidate models we considered were Logistic regression and Random Forest. For Logistic regression, we assume that the independent variables are related linearly through log-odds. This does not impose any direct linear constraints on the relationship between our predictors and response. Random forest does not carry with it any strong assumptions, but its computational intensity was considered as a potential drawback. With the ability to compute a random forest, we certainly wanted to apply it to our data given its ability to capture highly complex, non-linear dependencies.

## 2.4 Logistic regression

We model the conditional expectation of y given x, using the logistic density. We choose parameters $\beta$ using maximum likelihood estimation.

$$P(Y = y|x) = \frac{1}{1 + exp(-\beta'\mathbf{x})} = Logistic(\beta'\mathbf{x})$$

Where $\mathbf{x}$ is a matrix of predictors and $\beta$ is a vector of the parameters.

### 2.4.1 Candidate models

We were very mindful of overfitting the data, and decided that variable selection methods would be the best way for us to derive candidate models. If we include too many variables, the resultant model may fit the noise in the data.

Reducing the number of predictors used in a model will also reduce the variance of the model, and shortens training times (computation).

Based on our conclusions from the earlier visualisations, we will include variables: age, job, marital, default, loan, contact, campaign, pdays, poutcome, edu.

The three model candidates are as follows:

1. Modelling log-odds as a linear combination of the selected variables.
2. Add interactions between age & job, age & education, job & default and age & marital to model 1.
3. Backwards stepwise logistic regression, which is a different best subset selection technique. We run a backwards stepwise regression using `step` with `direction = "backwards"`.

The models were estimated using the entire training set. The code and and the regression output for the three models can be found in appendix D.

### 2.4.2 Variable interactions

Adding interaction terms in a regression model can greatly expand the ability of the model to capture the true relationships between variables in the data generating process. We suspect that interactions involving age, job, education and marital status will be particularly useful predictors. This flexibility allows the model to account for any effects associated with particular socio-demographic segments of the customers in the training set.

We have included the following interactions in our logistic model:

**Age and Marital: all levels**

As an individual ages, they are more likely to get married - a well known fact of life. The action of getting married at a young age (e.g. 20), as opposed to a middle age (e.g. 40) may be an indicator of an individual's decision making.

Figure 17 shows that amongst single people, those that are young and old individuals have a higher proportion of subscriptions than middle aged. Also, middle aged people with marital status as "unknown" have a higher incidence of subscriptions.

On this basis, we include the interaction age and marital status in our model.

**Age and Job: only age $\times$ student & age $\times$ retired**

The boxplot in figure 18 reveals that age of retirees and students is quite different to the rest of the job categories, and hence we will include interactions between age and these job categories.

**Age and Education: all levels**

This interaction will be included for the full set of education categories. We see that whilst subscription decisions are similar for ages under 30 and over 60, the middle age bracket varies across all education categories (see figure 20).

**Job and Default: all levels**

There is high variability in the incidence of "unknown" responses across job categories, shown in figure 19. Since different job categories have different pay scales, we suspect that having defaulted (and not responding to the question) will have different effects on likelihood of subscription across job categories.

Note: figures 18, 20 & 19 are found in Appendix B.

### 2.4.3 Cross-validation

In order to estimate test error, we used K fold cross-validation for each of the models. This meant that for the stepwise regression, the `step` function had to be embedded in the loop for cross-validation. The choice was made to use K=20 folds. Ideally, we would like to use a larger K, perhaps 100, but decided against it due to computation times.

The cross-validation errors for these models, using the log loss function, can be seen in figure 1, located in the Results and Discussion section.

## 2.5 Classifcation Trees and Random Forests

We began by briefly exploring classification trees using all features as potential candidates for stratifying the sample space.[2] In using the default inbuilt 10-fold cross-validation procedure in the `rpart` package and implementing the one standard error rule, the best tree size was zero – the root node. The root node simply gives the proportion of the training observations in each of the classes. This result may suggest that the predictor variables do not provide sufficient information to grow the tree such that the probability of observing whether a client will subscribe to the bank deposit is not conditional on the independent variables.

Due to the cross validation results, as well as their general non-robustness in terms of changes to the data set, classification trees were not further pursued. Further, as mentioned in the data analysis in appendix A, the test data and the training data have some distinct differences that would not be modelled well by a single tree (given we allowed the tree size to increase), which would try overfit the training data.

Random forests were subsequently considered due to their increased predictive power. Each individual tree in the ensemble is grown deep, and thus has low bias and high variance. By taking the average of these trees, the variance of the ensemble

[2] In cases where the data was not further manipulated, *default* was removed when running cross-validation.

is reduced at the expense of some bias. Initially, we expected that the random forest algorithm would suit our problem: We are using quite a large training sample and we believe it is reasonable to assume that there are no relationships between any of the observations– we are assuming that the data is IID. Both of these are important when working with the bootstrap.

Random forests are able to handle both continuous and multivalued categorical variables–they are able to partition the sample space on one or more of the levels of the categorical variable. However, both these variable types are more frequently chosen for sample splitting. We tried to overcome this problem by converting all the categorical and continuous variables in the data set into dummy variables, however, this resulted in consistenly higher log loss on our training data than a random forest using the variables as they have been defined thus. Similarly, since random forests can handle a large number of predictors, no dimension reduction was unnecessary.

Three tree ensemble algorithms were initially considered. More specifically:

1. Bagging
2. Random forest with default `mtry`[3]
3. Random forest with `mtry` chosen through 20-fold cross-validation – `mtry = 5` was ultimately chosen.

All algorithms were run using 500 trees. This decision was made primarily to improve computing efficiency while cross-validation was run. Further, when inspecting the out-of-bag error rates produced by the random forest algorithms, 500 trees was perceived as more than enough for the error rates to level off. In general, random forests are robust to overfitting, and therefore the fit may have marginally improved were we to further increase the number of trees.

# 3 Results and Discussion

## 3.1 Model assessment

The logistic regression models appear to perform much better than the classification trees and random forests, based on our cross-validation (log loss) errors.

The stepwise regression (logistic model 3), has the lowest cross-validation error of 0.2521. The model with interactions (logistic 2) slightly outperforms the basic model (logistic 1), with cross-validation errors of 0.2622 and 0.2624 respectively.

Amongst the three tree-based candidates, random forest using `mtry = 5` (tree-based model 3), performed marginally best, with a cross-validated log loss of 0.3068. All three of these candidates produced similar cross-validated errors.

Table 2: Cross-validation errors

| Model | Training log loss | Test log loss |
|---|---|---|
| Step Logistic | 0.252 | 0.609 |
| Simple Logistic | 0.262 | 0.5413 |
| Logistic with interactions | 0.262 | 0.535 |
| Random Forest CV | 0.307 | 0.601 |
| Random Forest Default | 0.331 | |
| Bagging | 0.324 | |

Figure 1 contains the cross-validation errors for all models.

Each of the logistic regression models outperform all of the tree-based candidates, and the logistic models are within one standard error of the lowest cross-validation error (logistic model 3). This indicates that the estimated errors are not significantly different across the three models, and that we are justified in choosing the simplest of the three, by the one standard error rule.

The stepwise regression model (logistic 3) is simplest model, containing 9 of a possible 14 predictors in our training set. Therefore, we select this to be our best and final model. It can be obtained by calling: `lm(formula = y ~ age + job + default + contact + month + day_of_week + campaign + pdays + previous, family = binomial, data = train)`.

[3] `mtry` is the number of randomly selected variables at each node. Default for a classification problem (`floor(sqrt(p))`where p is the number of predictors).

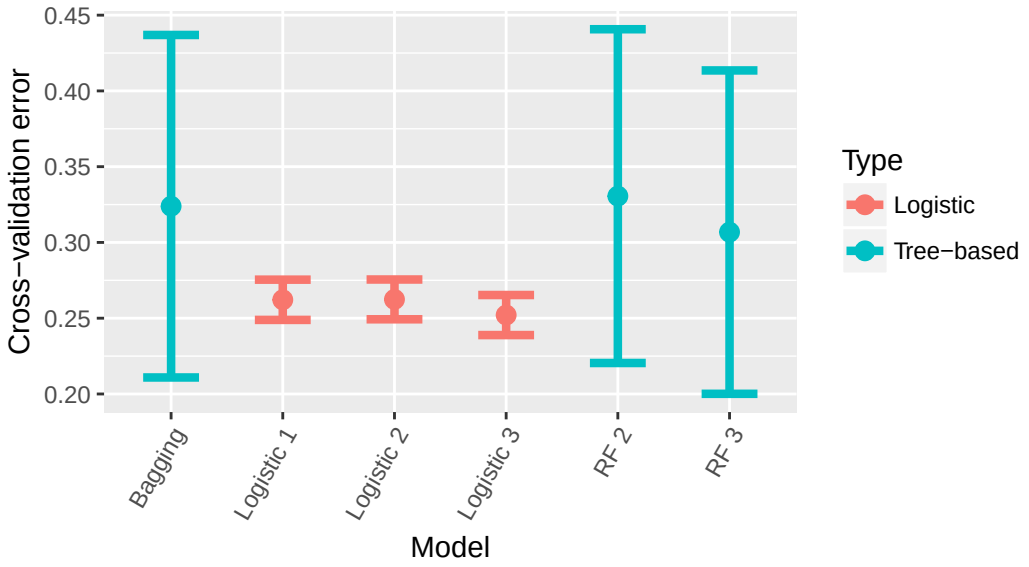


Figure 1: 20-fold cross-validation results for all candidate models. Log loss used as a proxy for the error rate to choose between the three models.

### 3.2 Performance on the test set

Interestingly, our preferred stepwise logistic regression did not perform as we expected on the kaggle test set, scoring a 0.609 log loss. This was trumped by the RF-CV[4] model, scoring a log loss of 0.601. We were not entirely surprised by this particular result, as the RF-CV standard error of cross-validation covered the log loss interval from the stepwise cross-validation (see figure 1). The random forest models also performed poorly.

Our best results came from our logistic models with the reduced variable set. Logistic model 2, which contained interactions between variables, performed best, with a score of 0.535. The model without interactions (logistic 1) scored 0.541.

Our conclusions on model preference, based on cross-validation, do not match the corresponding test performance. We suspect that this is caused by discrepencies that exist between the test and training data sets.

### 3.3 Caveats and improvements

The analysis in this report leads to the conclusion that a logistic model with a reduced predictor set is preferred to predit bank subscriptions. Typically logistic regression deals poorly with a significant class imbalance, as we have in our problem with only 7.69% of individuals in the training set having $y = 1$. This means that any models will tend to over-predict $y = 0$.

Further investigation into oversampling techniques could yield significant model improvements, so that the model does not over-predict the majority class, $y = 0$. Undersampling could also be considered, but we would lose information by dropping observations.

The curse of learning from an extremely imbalanced training data set is not limited to logistic regression. Random forests also will tend to focus more on the prediction accuracy of the majority class, which often results in poor accuracy for the minority class, as it minimizes the overall error rate. It is also likely that a bootstrap sample will contain few or even none of the minority class, resulting in a tree with poor performance for predicting $y = 1$.

There may also be opportunities to model using higher values of K using the K nearest neighbour (KNN) technique, and using linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA). We found that cross-validation error using KNN decreased up to $K = 10$, but then sagnated. Looking at higher values (e.g. $K = 90$) may reveal different results. We also explored ridge and LASSO regression techniques, but did not pursue them because their cross-validation errors were not lower than our candidate models.

[4]Random forest with the "mtry" parameter chosen via cross-validation

# 4 Conclusion

Our final model is the backwards stepwise logistic regression (logistic model 3), as it had the lowest cross-validation error, and therefore we expect it to perform best on the test data. Initially, we expected random forest to perform well, given the highly complex, non-linear dependencies in the problem, but were underwhelmed by its performance on the imbalanced training data set.

Surprisingly, our approach of using data visualisation to inform our model specification was fruitful. The logistic regression using interactions and a reduced predictor set (logistic 2) produced the lowest test log loss.

A benefit of using logistic regression is also its interpretability. In our chosen stepwise regression model, we find that job $\times$ retired, job $\times$ student, campaign and contact are some of the most significant predictors in bank deposit subscription (see figure **??** in Appendix C for the full regression output).

# 5 Appendix

## 5.1 Appendix A

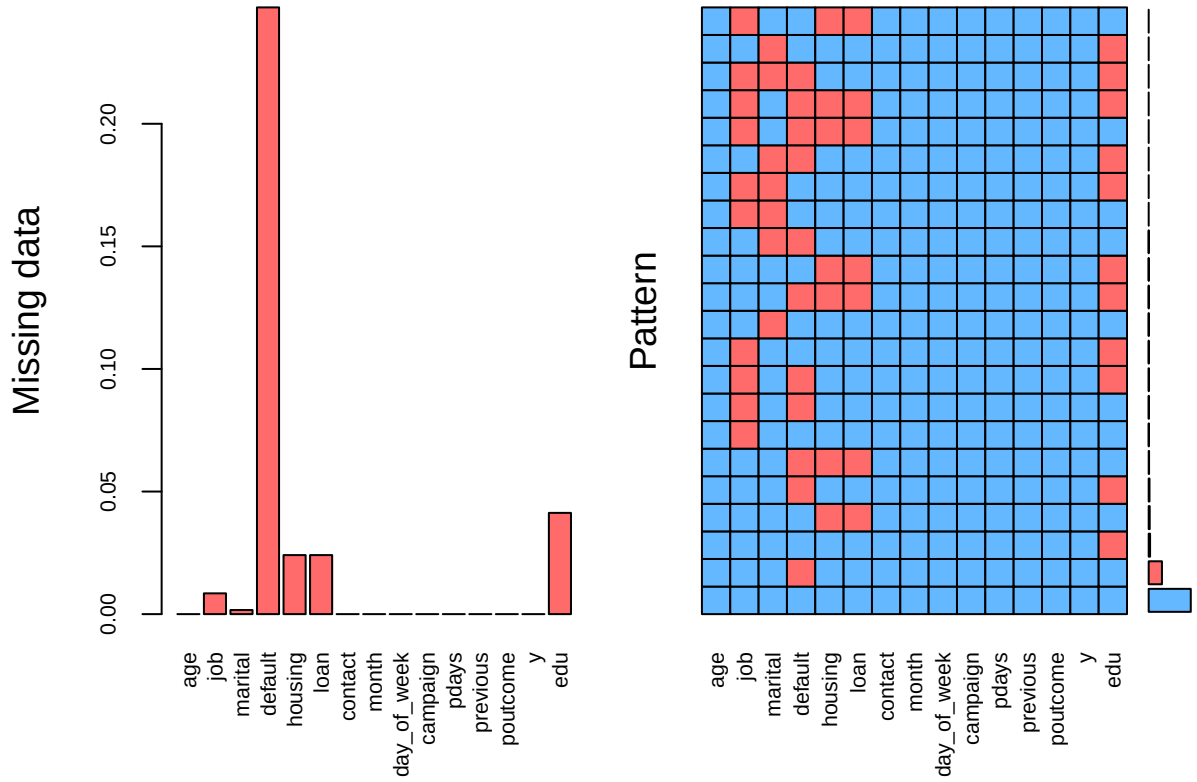


Figure 2: This plot illustrates the location of missing unknown responses in the data set. In particular, we observe many of these in variables such as job, marital, default, housing, loan.

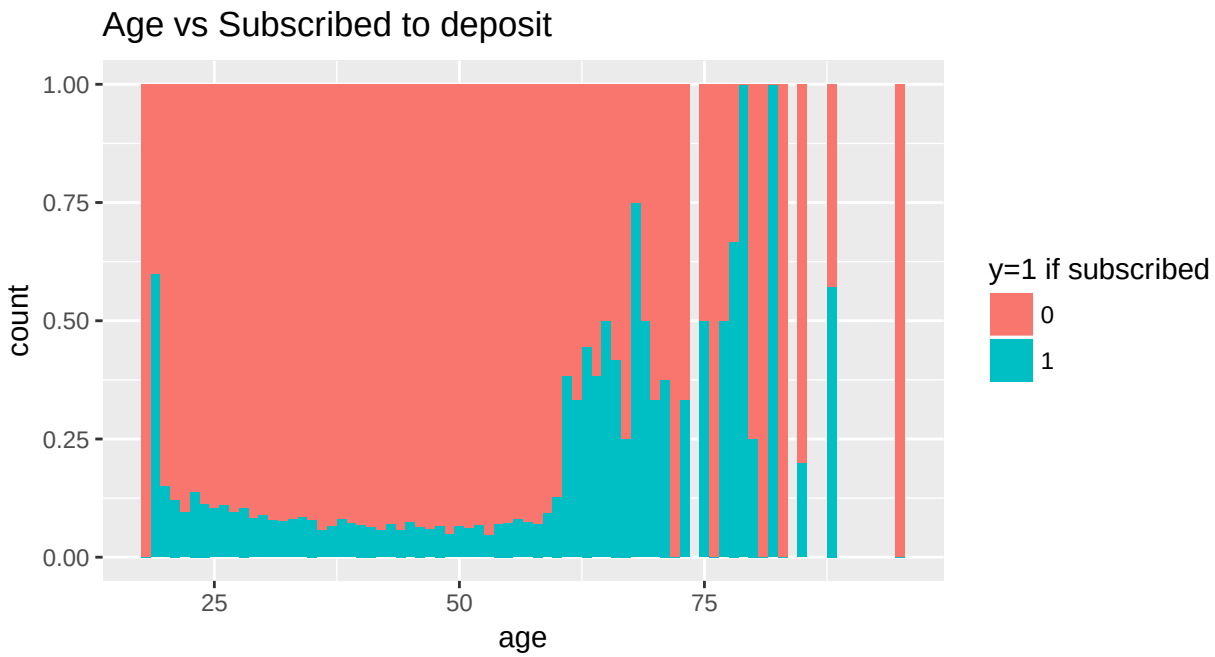


Figure 3: Older clients, as well as 19 year old's tend to have a higher proportion subscribed to bank-term deposits.

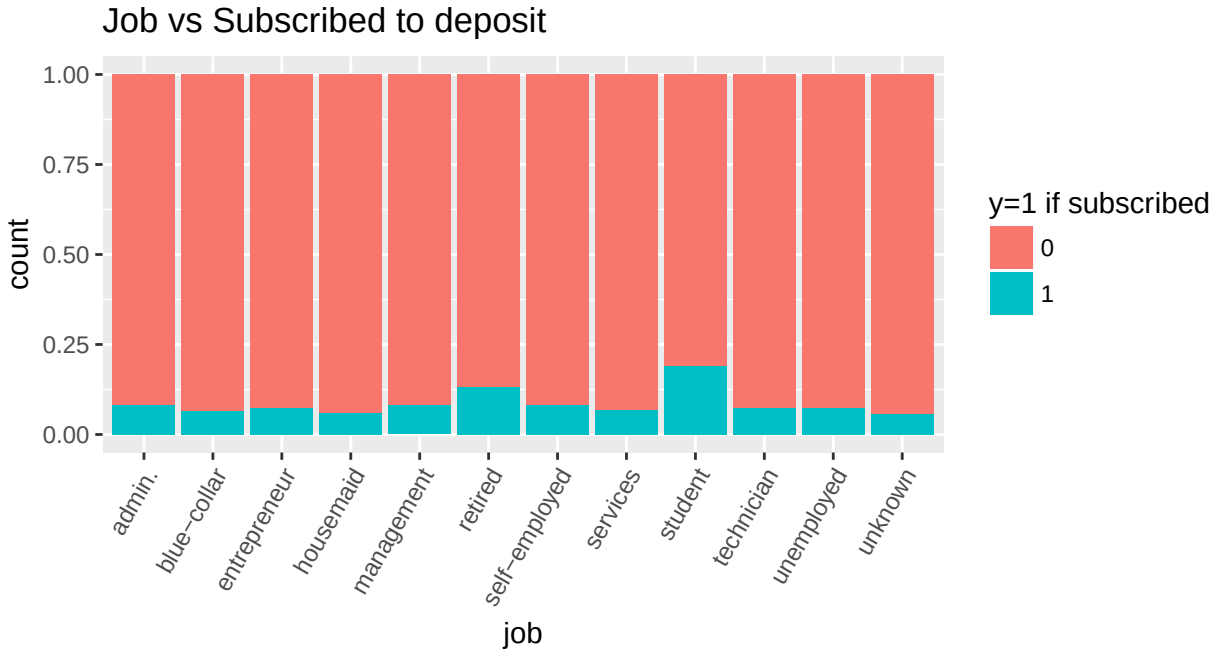


Figure 4: Students and retired clients tend to have a higher proportion subscribed to bank-term deposits.

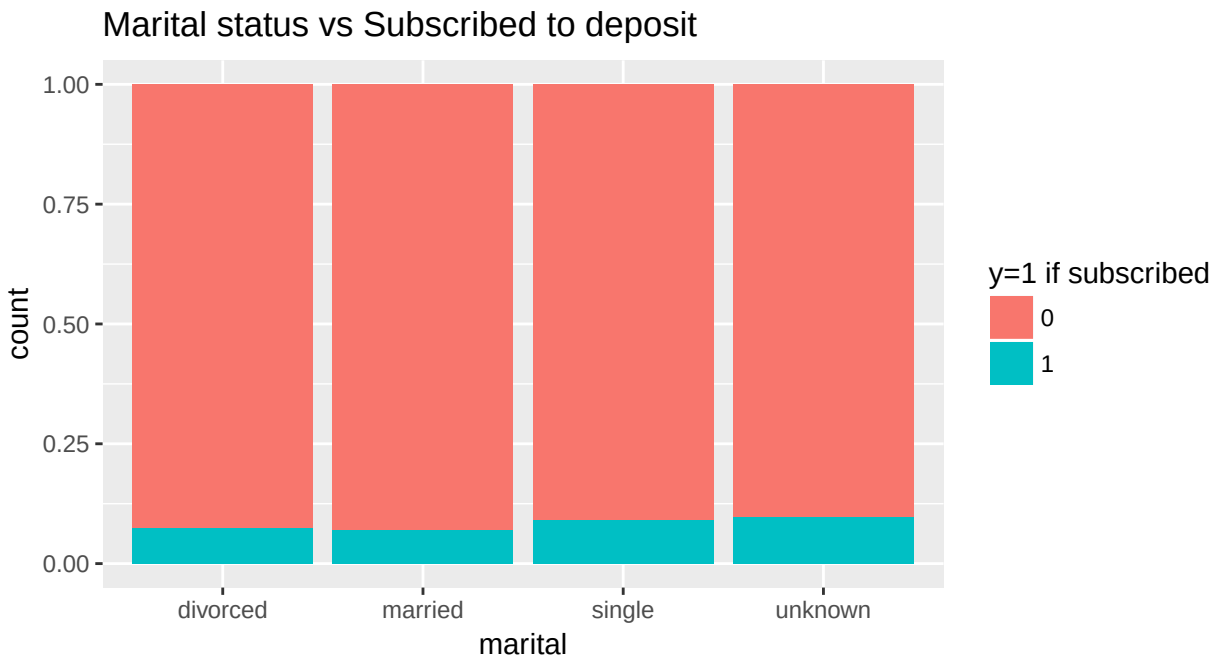


Figure 5: Single clients and those with an unknown marital status have a higher proportion subscribed to bank-term deposits, however the difference in proportions across marital classes is very small. Looking at a univariate distribution does not take into account interactions between marital and other predictors so as we suspect there may be some confounding between age and marital status, this will be explored.

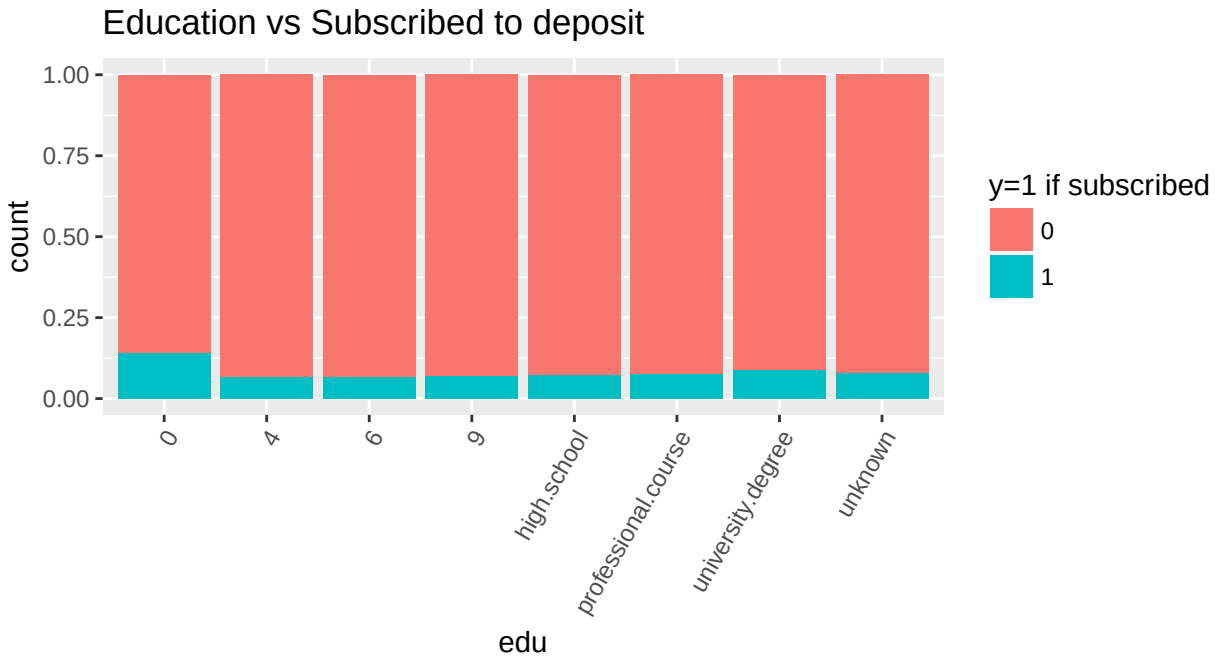


Figure 6: We notice that some of the education values have not been coded with their correct names, and instead coded with values 0, 4, 6 and 9. We treat these as distinct categories, on the assumption that only their names have been corrupted. Those with education 0 tend to have a higher proportion subscribed to bank-term deposits.

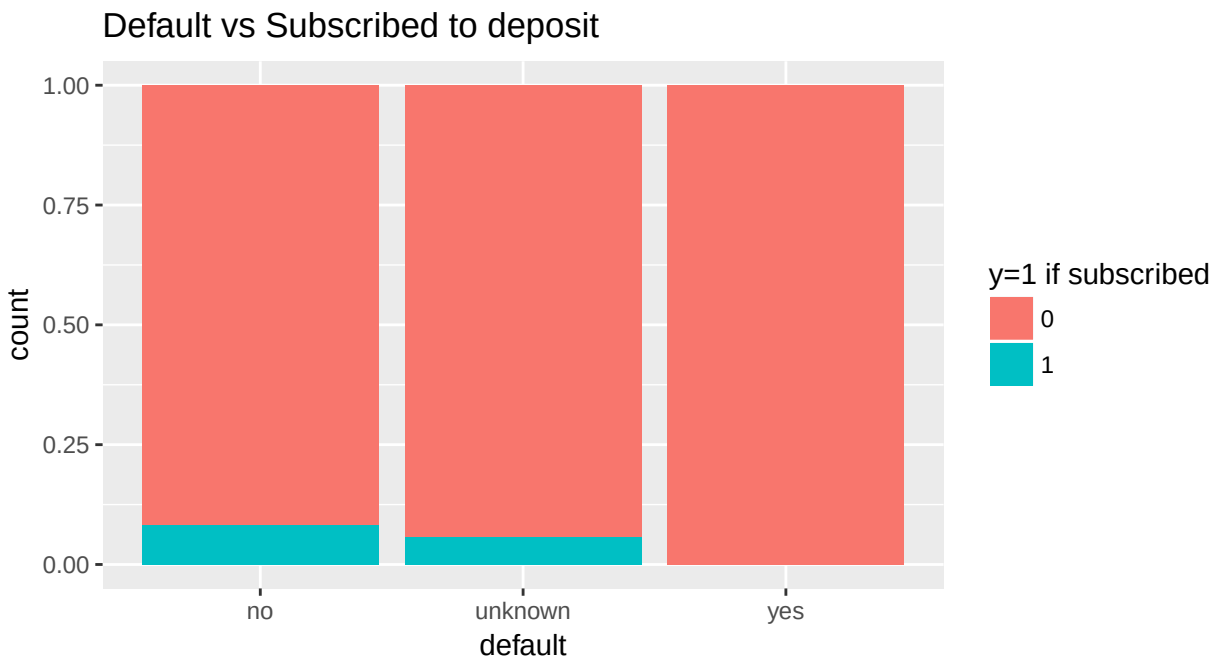


Figure 7: Those who do not have credit in default have a higher proportion subscribed to bank-term deposits. None of the clients who have credit in default subscribed to bank term deposits.

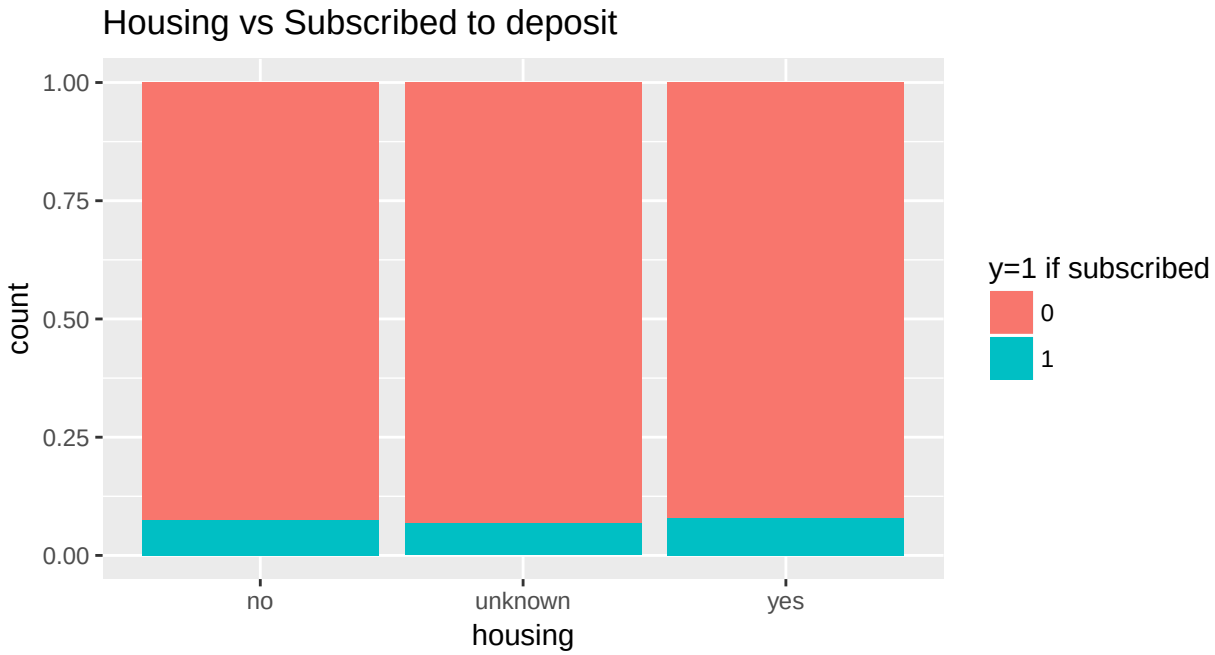


Figure 8: Appears to have same proportions across 3 levels.


Figure 9: Appears to have same proportions across 3 levels.

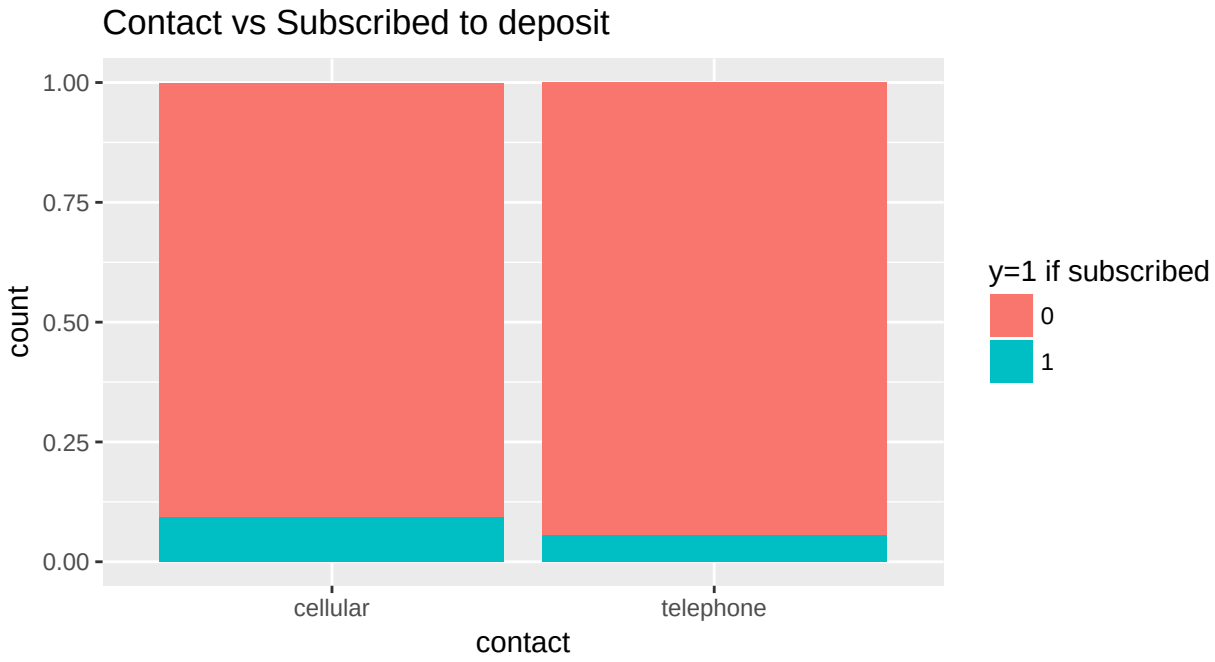


Figure 10: Those who were last contacted using a form of cellular communication have a higher proportion subscribed to bank-term deposits.

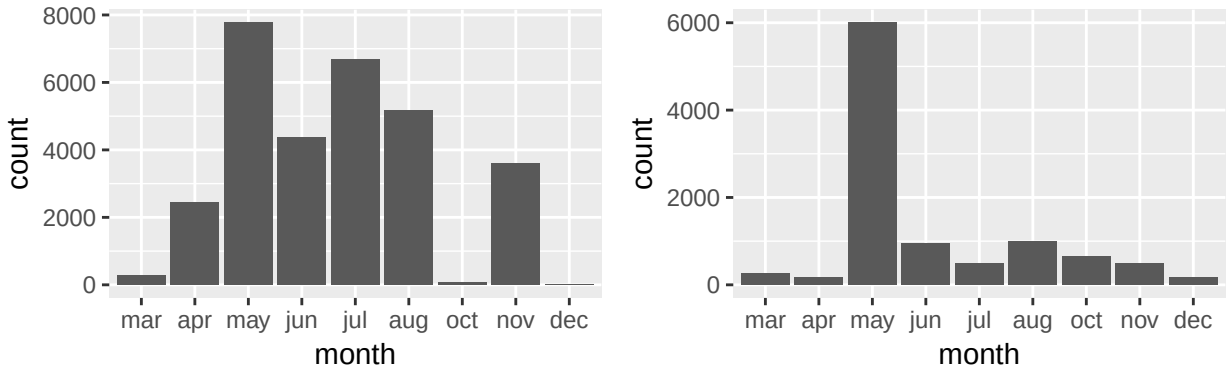


Figure 11: Ideally, training data should never influence testing data in any way as this results in a leakage of information. In theory, with sufficient data, the central limit theorem states that the distributions in the training data should converge on distributions in the testing data. However, it can be seen here there are is a very small number of observations in some levels of the month training data, particularly in October and December, so we suspected there is insufficient data for this convergence to take effect, suggesting that excluding month as a predictor may be a beneficial for predicting the outcome variable.

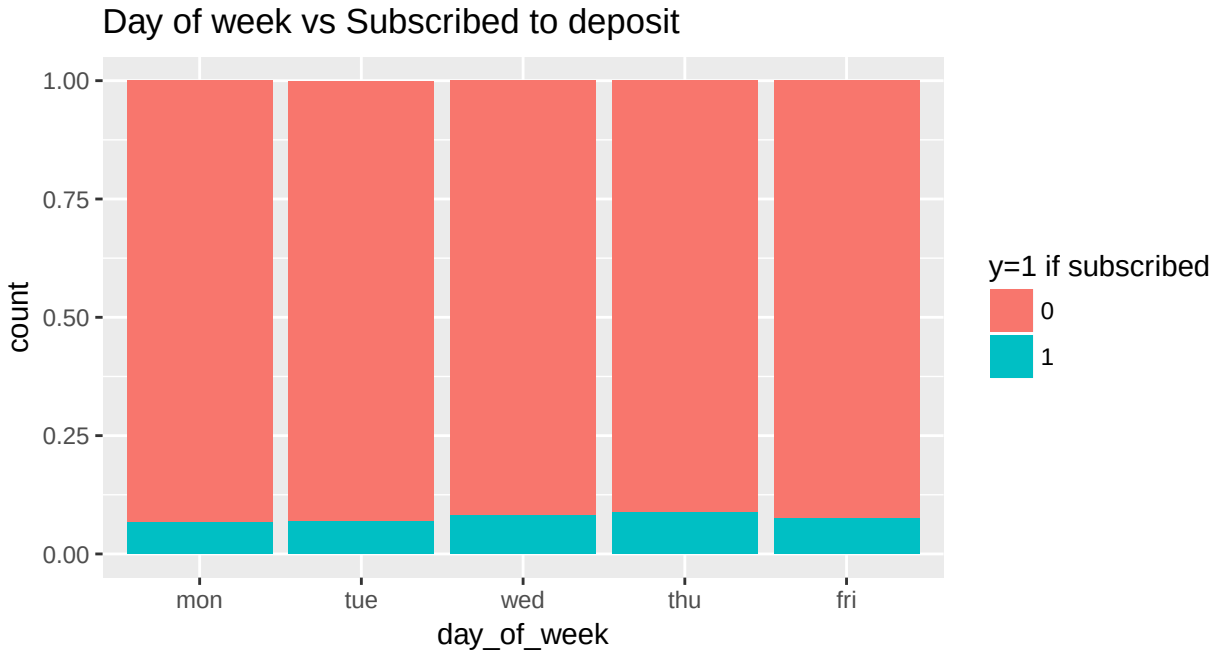


Figure 12: The day of the week does not appear to be influential, as proportion of subscriptions are very similar across all days.

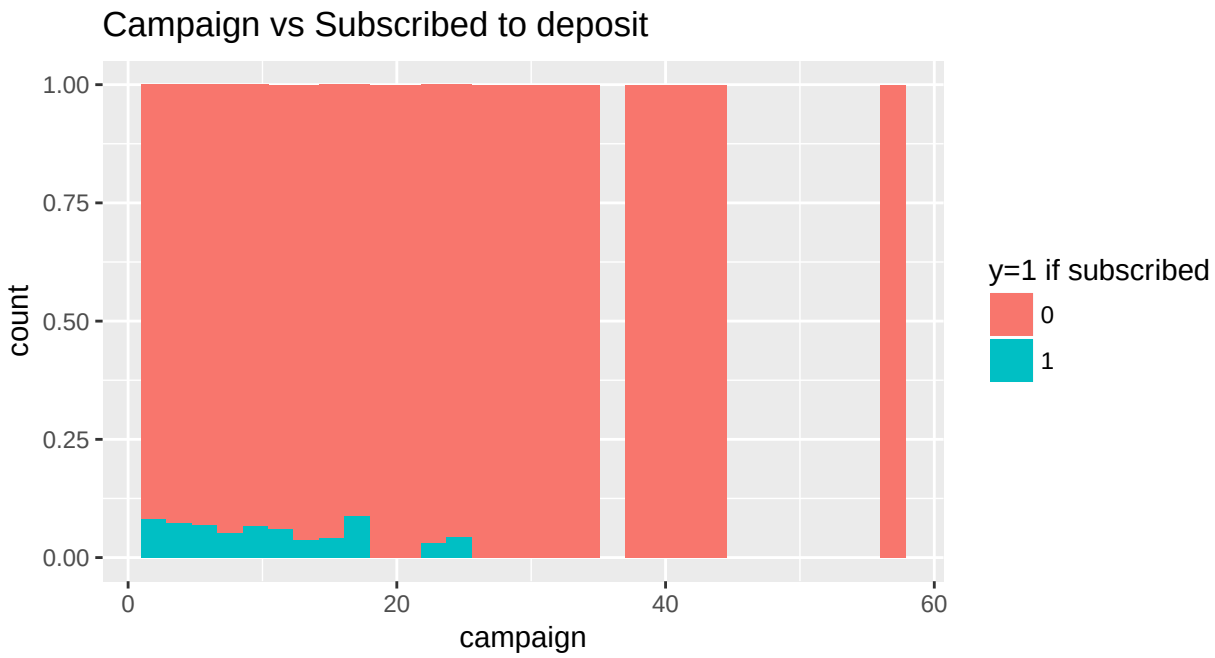


Figure 13: A lower number of contacts performed for clients tend to have a higher proportion subscribed to bank-term deposits.

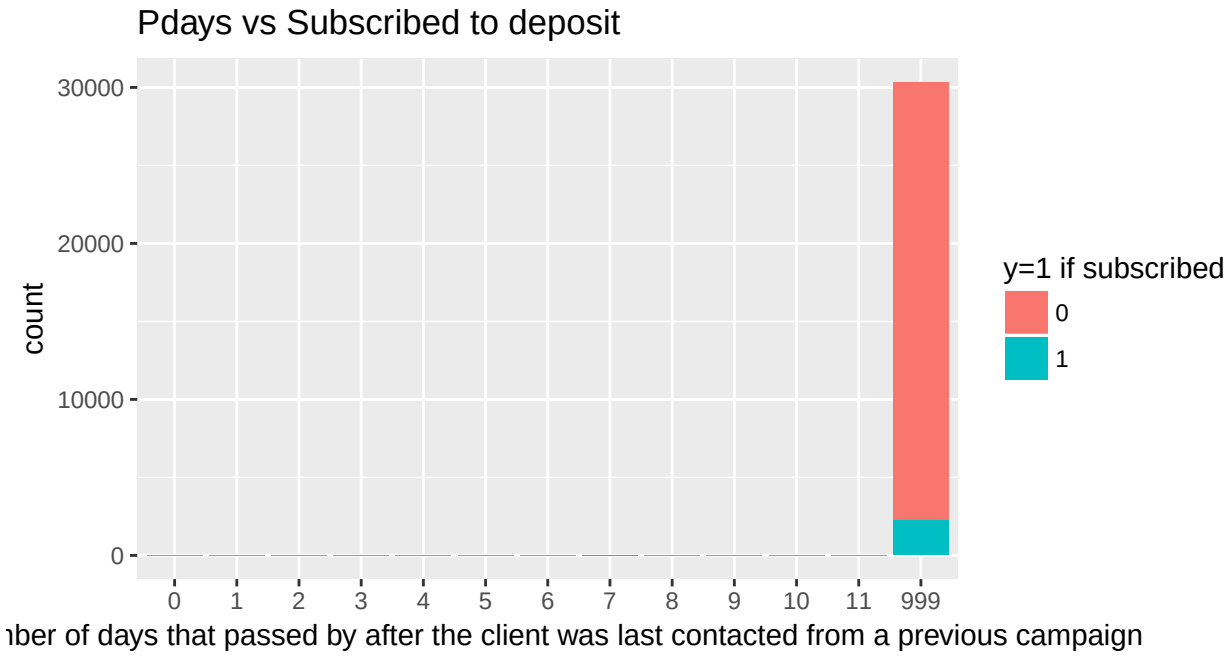


Figure 14: The pdays variable is imbalanced, with few clients being contacted from a previous campaign.

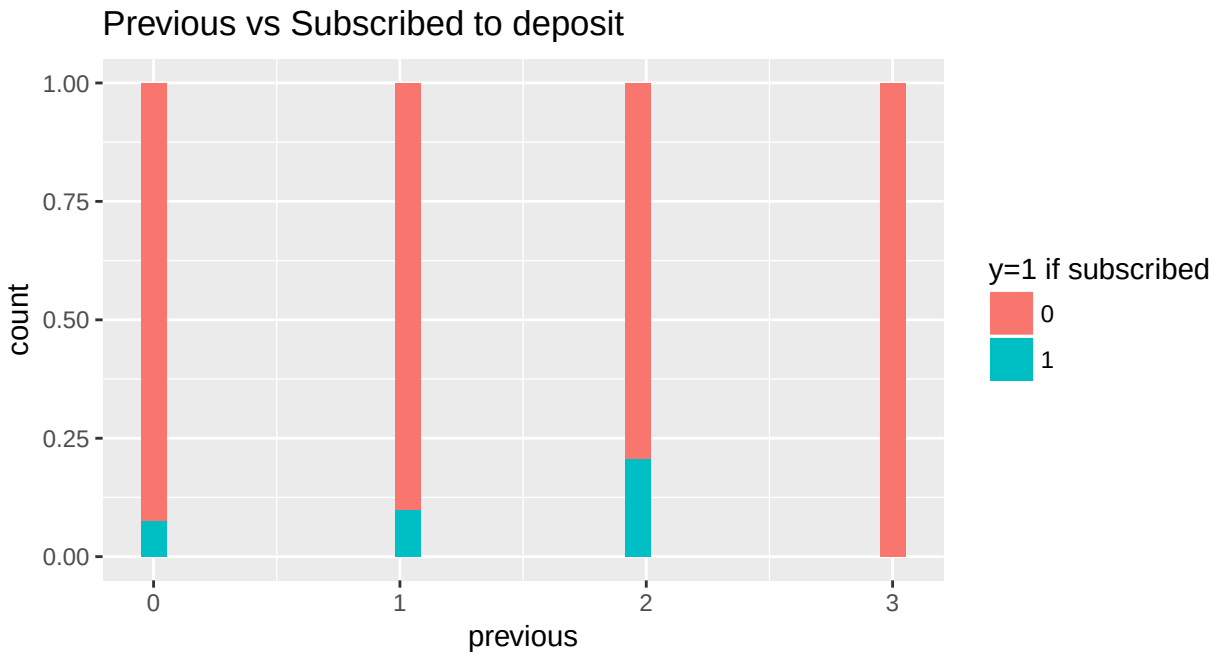


Figure 15: Clients on who were contacted twice before this campaign tend to have a higher proportion subscribed to bank-term deposits.

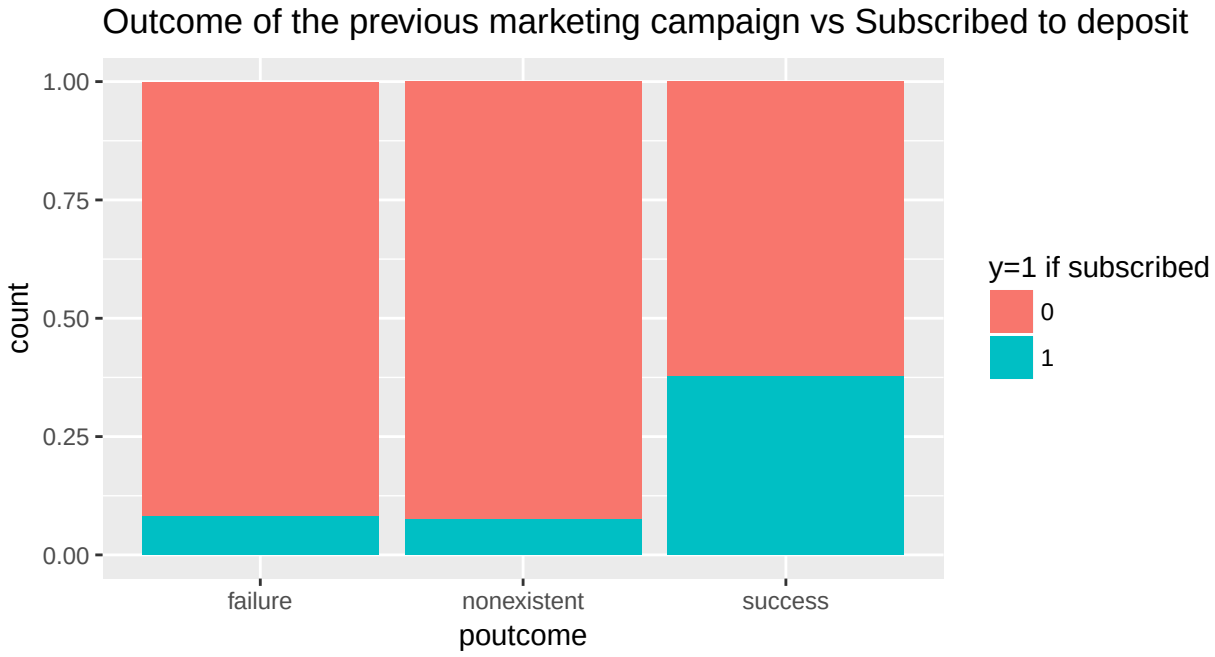


Figure 16: Clients on whom the previous marketing campaign was successful tend to have a higher proportion subscribed to bank-term deposits.

## 5.2 Appendix B

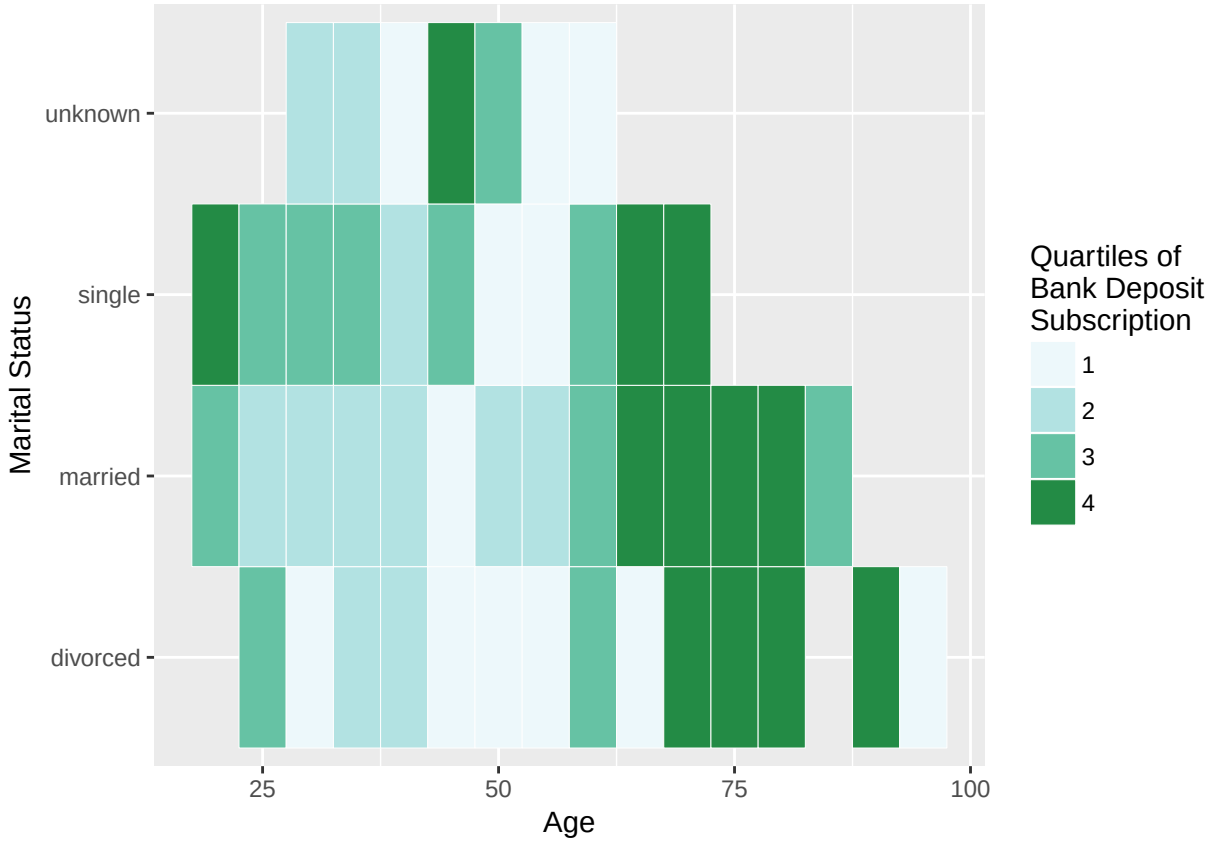


Figure 17: Propensity of bank deposit subscriptions, across grouping of age and marital status

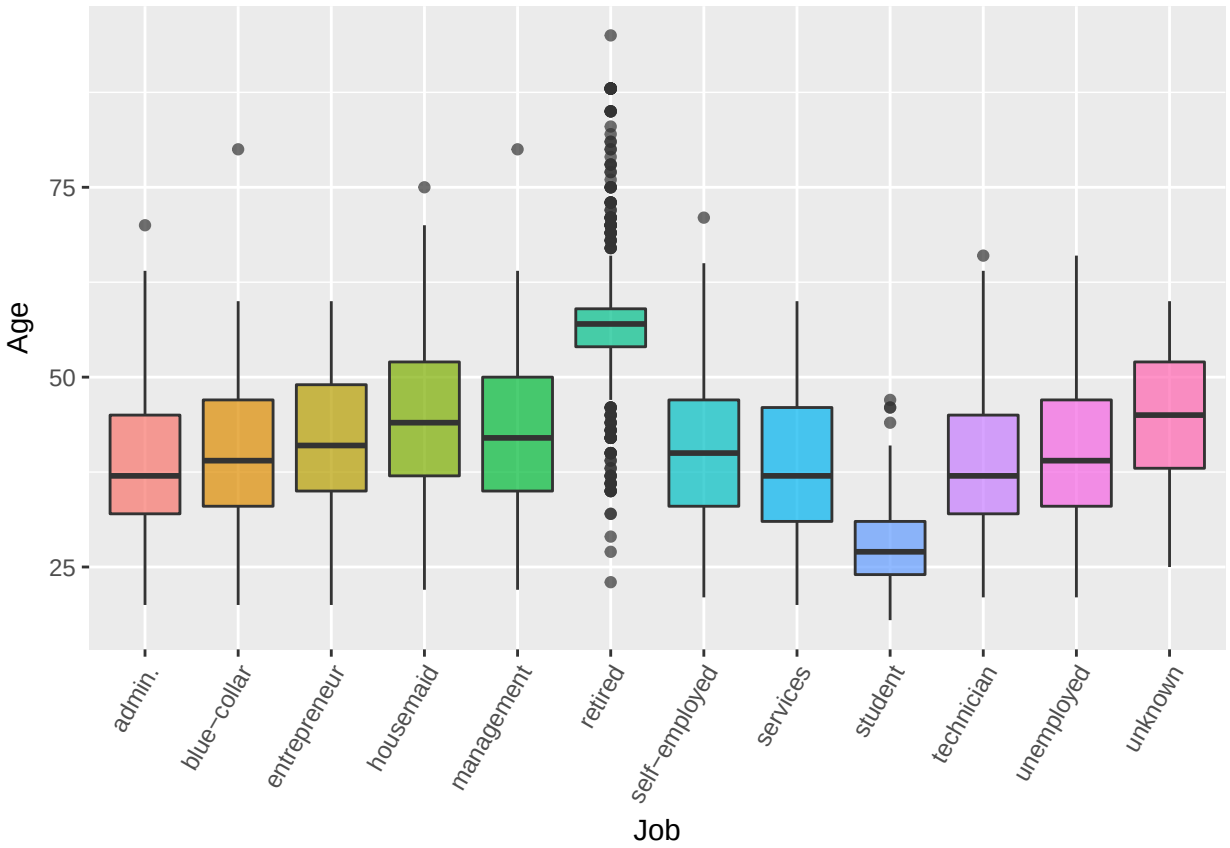


Figure 18: Boxplot of age by each job type

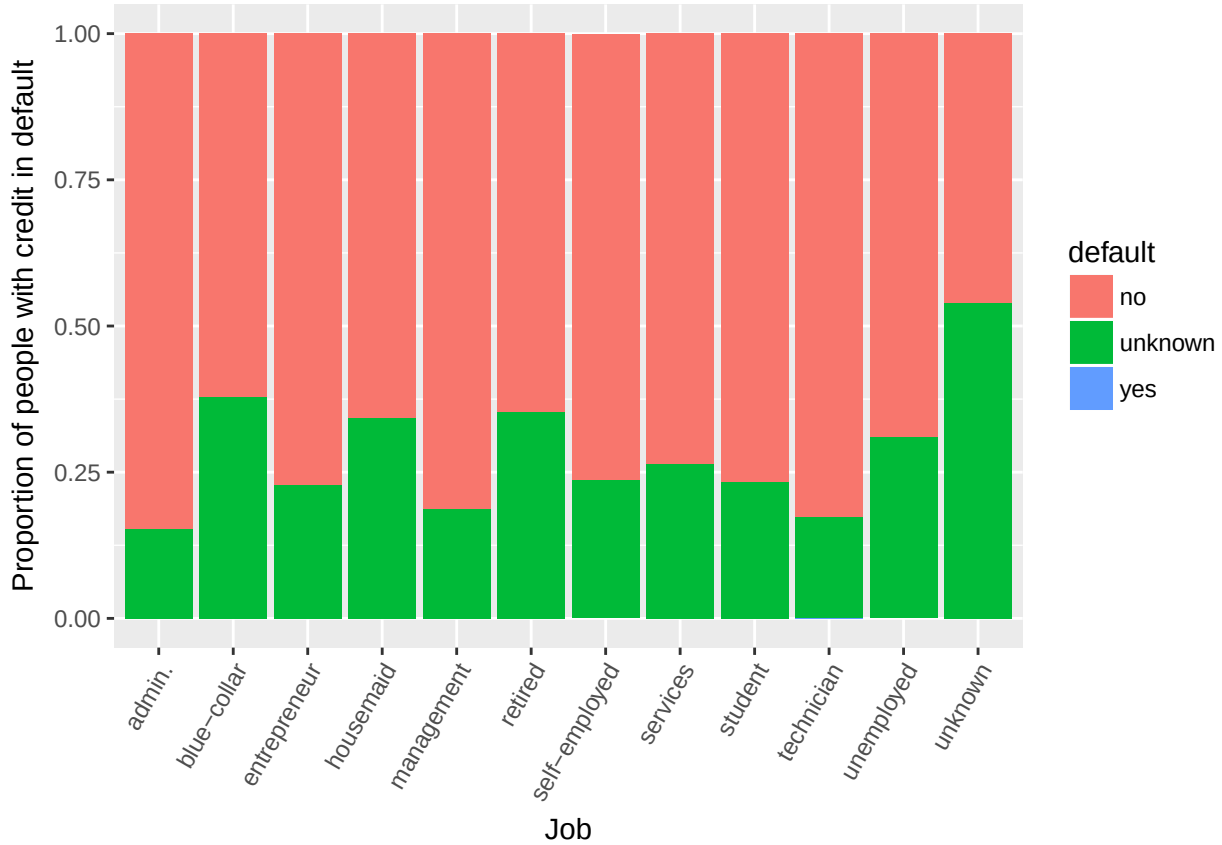


Figure 19: Proportion of individuals with credit in default, across job categories

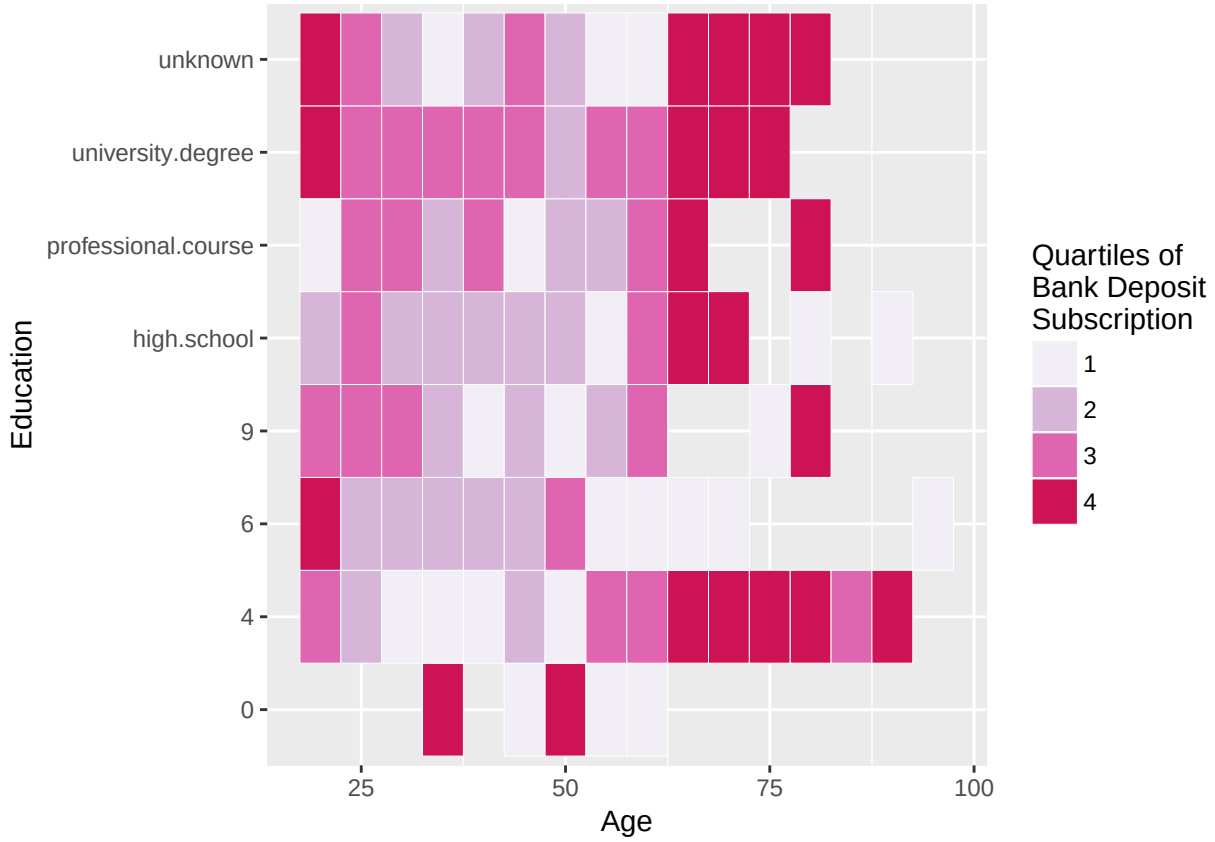


Figure 20: Subscription incidence for age and education groups

## 5.3 Appendix C

**Model 1:**

```
m1 <- glm(y ~ age + job + marital + default + loan + contact + campaign + pdays + poutcome + edu,
family=binomial, data = train)
```

**Model 2:**

```
m2 <- glm(y ~ age*factor(job=="retired") + age*factor(job=="student") + marital*age + default*job +
loan+contact + campaign+pdays + poutcome+edu*age, family = binomial, data = train)
```

**Model 3:**

```
m3 <- step(glm(y ~ ., family=binomial, data = train))
```

### 5.3.1 Model 1 - Logistic Regression with Selected Variables

% latex table generated in R 3.4.1 by xtable 1.8-2 package % Sun May 20 23:50:01 2018

### 5.3.2 Model 2 - Logistic Regression with Selected Variables and Interactions

% latex table generated in R 3.4.1 by xtable 1.8-2 package % Sun May 20 23:50:05 2018

### 5.3.3 Model 3 - Stepwise Logistic Regression

% latex table generated in R 3.4.1 by xtable 1.8-2 package % Sun May 20 23:51:31 2018

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | 0.2984 | 1.3111 | 0.23 | 0.8200 |
| age | -0.0031 | 0.0028 | -1.10 | 0.2733 |
| jobblue-collar | -0.0137 | 0.0820 | -0.17 | 0.8672 |
| jobentrepreneur | -0.0254 | 0.1215 | -0.21 | 0.8342 |
| jobhousemaid | -0.1629 | 0.1577 | -1.03 | 0.3015 |
| jobmanagement | 0.0457 | 0.0895 | 0.51 | 0.6098 |
| jobretired | 0.7396 | 0.1192 | 6.20 | 0.0000 |
| jobself-employed | 0.0716 | 0.1194 | 0.60 | 0.5485 |
| jobservices | -0.0386 | 0.0901 | -0.43 | 0.6685 |
| jobstudent | 0.8821 | 0.1592 | 5.54 | 0.0000 |
| jobtechnician | -0.1380 | 0.0768 | -1.80 | 0.0725 |
| jobunemployed | -0.0332 | 0.1528 | -0.22 | 0.8282 |
| jobunknown | -0.1278 | 0.2774 | -0.46 | 0.6452 |
| maritalmarried | -0.0066 | 0.0711 | -0.09 | 0.9256 |
| maritalsingle | 0.1519 | 0.0809 | 1.88 | 0.0603 |
| maritalunknown | 0.3546 | 0.4809 | 0.74 | 0.4609 |
| defaultunknown | -0.2705 | 0.0576 | -4.70 | 0.0000 |
| defaultyes | -9.1647 | 113.6942 | -0.08 | 0.9358 |
| loanunknown | -0.0721 | 0.1483 | -0.49 | 0.6268 |
| loanyes | -0.0334 | 0.0610 | -0.55 | 0.5836 |
| contacttelephone | -0.4992 | 0.0471 | -10.61 | 0.0000 |
| campaign | -0.0460 | 0.0092 | -4.99 | 0.0000 |
| pdays | -0.0018 | 0.0011 | -1.75 | 0.0799 |
| poutcomenonexistent | 0.1599 | 0.1047 | 1.53 | 0.1268 |
| poutcomesuccess | 0.0573 | 1.0650 | 0.05 | 0.9571 |
| edu4 | -0.7115 | 0.7721 | -0.92 | 0.3568 |
| edu6 | -0.6611 | 0.7747 | -0.85 | 0.3935 |
| edu9 | -0.7010 | 0.7709 | -0.91 | 0.3632 |
| eduhigh.school | -0.7057 | 0.7712 | -0.92 | 0.3601 |
| eduprofessional.course | -0.6395 | 0.7725 | -0.83 | 0.4078 |
| eduuniversity.degree | -0.5813 | 0.7706 | -0.75 | 0.4506 |
| eduunknown | -0.5678 | 0.7764 | -0.73 | 0.4646 |

#### 5.3.4 Random Forests- mtry selection

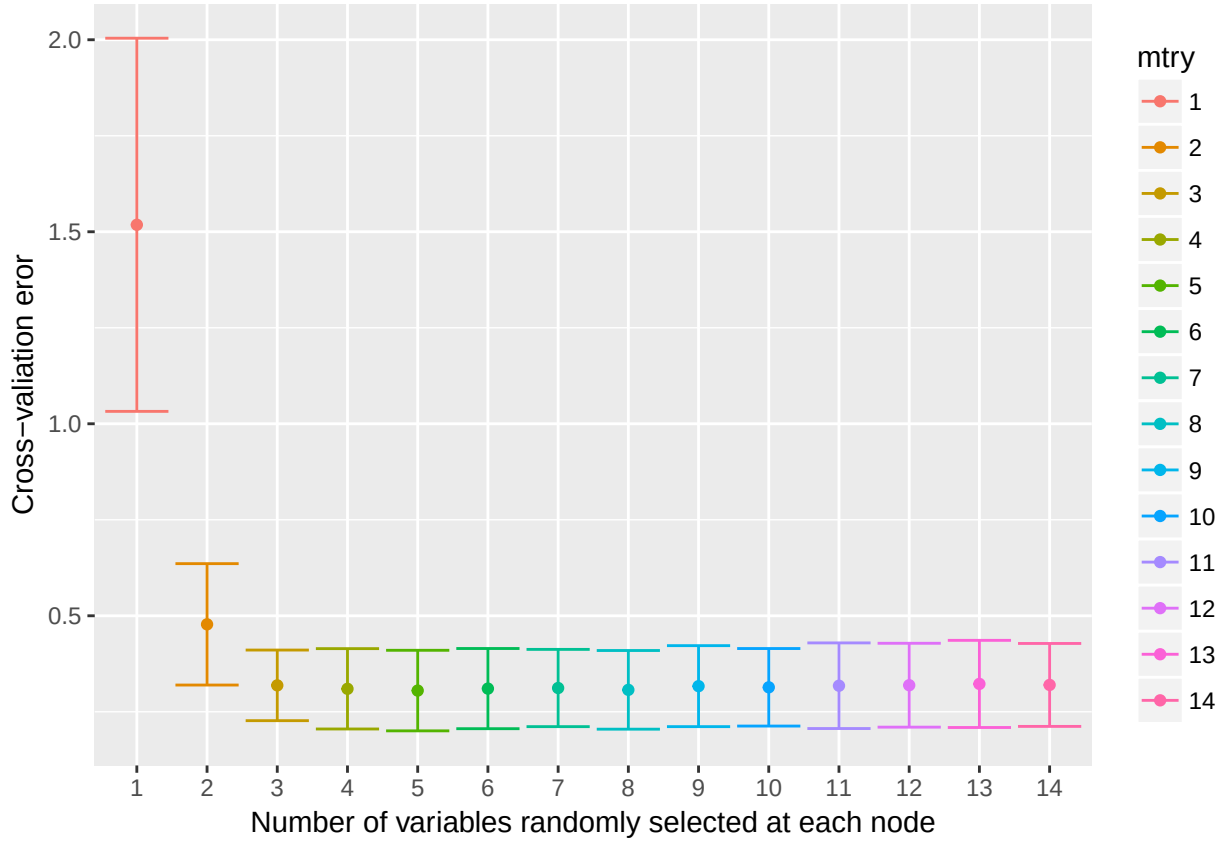


Figure 21: The plot illustrates the results from the 20-fold cross-validation conducted to select the best mtry parameter, given our training data. Ultimately, five was chosen as the number of variables randomly selected for partition at each node. In this case, the one standard error rule was not taken into account.

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | 3.1909 | 4.3438 | 0.73 | 0.4626 |
| age | -0.0695 | 0.0952 | -0.73 | 0.4653 |
| factor(job == "retired")TRUE | -2.4799 | 0.7903 | -3.14 | 0.0017 |
| factor(job == "student")TRUE | 0.8937 | 0.8135 | 1.10 | 0.2719 |
| maritalmarried | -0.1138 | 0.3188 | -0.36 | 0.7212 |
| maritalsingle | 0.7387 | 0.3687 | 2.00 | 0.0451 |
| maritalunknown | 0.4573 | 2.0617 | 0.22 | 0.8245 |
| defaultunknown | -0.2775 | 0.1325 | -2.09 | 0.0363 |
| defaultyes | -9.0442 | 196.9678 | -0.05 | 0.9634 |
| jobblue-collar | -0.0226 | 0.0909 | -0.25 | 0.8038 |
| jobentrepreneur | -0.0706 | 0.1363 | -0.52 | 0.6043 |
| jobhousemaid | -0.1351 | 0.1837 | -0.74 | 0.4622 |
| jobmanagement | 0.0184 | 0.0974 | 0.19 | 0.8498 |
| jobself-employed | 0.0293 | 0.1328 | 0.22 | 0.8252 |
| jobservices | -0.0332 | 0.0987 | -0.34 | 0.7366 |
| jobtechnician | -0.1400 | 0.0814 | -1.72 | 0.0855 |
| jobunemployed | -0.0646 | 0.1753 | -0.37 | 0.7124 |
| jobunknown | 0.2183 | 0.3275 | 0.67 | 0.5050 |
| loanunknown | -0.0743 | 0.1486 | -0.50 | 0.6170 |
| loanyes | -0.0367 | 0.0612 | -0.60 | 0.5488 |
| contacttelephone | -0.4799 | 0.0474 | -10.13 | 0.0000 |
| campaign | -0.0455 | 0.0092 | -4.93 | 0.0000 |
| pdays | -0.0018 | 0.0011 | -1.69 | 0.0919 |
| poutcomenonexistent | 0.1575 | 0.1053 | 1.50 | 0.1345 |
| poutcomesuccess | 0.1487 | 1.0687 | 0.14 | 0.8893 |
| edu4 | -4.6746 | 4.2195 | -1.11 | 0.2679 |
| edu6 | -2.6866 | 4.2282 | -0.64 | 0.5252 |
| edu9 | -3.4744 | 4.2106 | -0.83 | 0.4093 |
| eduhigh.school | -3.5795 | 4.2072 | -0.85 | 0.3949 |
| eduprofessional.course | -3.6054 | 4.2105 | -0.86 | 0.3918 |
| eduuniversity.degree | -3.3816 | 4.2045 | -0.80 | 0.4212 |
| eduunknown | -3.2227 | 4.2251 | -0.76 | 0.4456 |
| age:factor(job == "retired")TRUE | 0.0550 | 0.0133 | 4.14 | 0.0000 |
| age:factor(job == "student")TRUE | -0.0019 | 0.0295 | -0.06 | 0.9488 |
| age:maritalmarried | 0.0030 | 0.0071 | 0.42 | 0.6750 |
| age:maritalsingle | -0.0176 | 0.0092 | -1.92 | 0.0549 |
| age:maritalunknown | -0.0027 | 0.0482 | -0.06 | 0.9561 |
| defaultunknown:jobblue-collar | 0.1142 | 0.1680 | 0.68 | 0.4966 |
| defaultunknown:jobentrepreneur | 0.3191 | 0.3019 | 1.06 | 0.2905 |
| defaultunknown:jobhousemaid | 0.0841 | 0.3443 | 0.24 | 0.8071 |
| defaultunknown:jobmanagement | 0.2168 | 0.2486 | 0.87 | 0.3832 |
| defaultunknown:jobretired | -0.2203 | 0.2725 | -0.81 | 0.4189 |
| defaultunknown:jobself-employed | 0.2426 | 0.3059 | 0.79 | 0.4277 |
| defaultunknown:jobservices | -0.0490 | 0.2254 | -0.22 | 0.8280 |
| defaultunknown:jobstudent | -0.4703 | 0.4611 | -1.02 | 0.3078 |
| defaultunknown:jobtechnician | -0.0003 | 0.2066 | -0.00 | 0.9988 |
| defaultyes:jobtechnician | -0.1862 | 241.2353 | -0.00 | 0.9994 |
| defaultunknown:jobunemployed | 0.1773 | 0.3581 | 0.50 | 0.6205 |
| defaultunknown:jobunknown | -0.8642 | 0.6154 | -1.40 | 0.1602 |
| age:edu4 | 0.0857 | 0.0953 | 0.90 | 0.3684 |
| age:edu6 | 0.0425 | 0.0957 | 0.44 | 0.6572 |
| age:edu9 | 0.0611 | 0.0952 | 0.64 | 0.5208 |
| age:eduhigh.school | 0.0639 | 0.0951 | 0.67 | 0.5016 |
| age:eduprofessional.course | 0.0666 | 0.0952 | 0.70 | 0.4841 |
| age:eduuniversity.degree | 0.0622 | 0.0951 | 0.65 | 0.5126 |
| age:eduunknown | 0.0589 | 0.0955 | 0.62 | 0.5372 |

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | 0.3453 | 0.2915 | 1.18 | 0.2362 |
| age | -0.0052 | 0.0026 | -2.02 | 0.0433 |
| jobblue-collar | -0.1211 | 0.0671 | -1.80 | 0.0712 |
| jobentrepreneur | -0.0797 | 0.1232 | -0.65 | 0.5180 |
| jobhousemaid | -0.1557 | 0.1534 | -1.01 | 0.3101 |
| jobmanagement | 0.0437 | 0.0908 | 0.48 | 0.6303 |
| jobretired | 0.5210 | 0.1218 | 4.28 | 0.0000 |
| jobself-employed | 0.0181 | 0.1215 | 0.15 | 0.8813 |
| jobservices | -0.1148 | 0.0860 | -1.34 | 0.1816 |
| jobstudent | 0.4127 | 0.1671 | 2.47 | 0.0135 |
| jobtechnician | -0.1092 | 0.0697 | -1.57 | 0.1173 |
| jobunemployed | -0.0676 | 0.1538 | -0.44 | 0.6601 |
| jobunknown | -0.1413 | 0.2768 | -0.51 | 0.6097 |
| defaultunknown | -0.1331 | 0.0584 | -2.28 | 0.0226 |
| defaultyes | -8.7188 | 113.5324 | -0.08 | 0.9388 |
| contacttelephone | -0.2442 | 0.0991 | -2.46 | 0.0137 |
| month.L | -0.4219 | 0.5554 | -0.76 | 0.4474 |
| month.Q | 1.1890 | 0.5692 | 2.09 | 0.0367 |
| month.C | -1.8884 | 0.4900 | -3.85 | 0.0001 |
| month^4 | -0.3528 | 0.3427 | -1.03 | 0.3033 |
| month^5 | 0.8327 | 0.2125 | 3.92 | 0.0001 |
| month^6 | 1.2771 | 0.1910 | 6.69 | 0.0000 |
| month^7 | 1.9321 | 0.1454 | 13.29 | 0.0000 |
| month^8 | 0.8147 | 0.0834 | 9.76 | 0.0000 |
| day_of_week.L | 0.1628 | 0.0513 | 3.17 | 0.0015 |
| day_of_week.Q | -0.2165 | 0.0508 | -4.26 | 0.0000 |
| day_of_week.C | -0.0257 | 0.0499 | -0.52 | 0.6060 |
| day_of_week^4 | 0.0480 | 0.0486 | 0.99 | 0.3229 |
| campaign | -0.0323 | 0.0090 | -3.58 | 0.0003 |
| pdays | -0.0018 | 0.0002 | -7.45 | 0.0000 |
| previous | -0.5252 | 0.1030 | -5.10 | 0.0000 |