CARAVAN INSURANCE POLICY PREDICTION

CAPSTONE PROJECT

Praveen Satyavarapu

01 January 2020

Contents

1 Overview

This project is the final requirement for HarvardX: PH125.9x Data Science: Capstone, the final course in the HarvardX Data Science Professional Certificate. The main goal of this project is to work independently on a data analysis project and apply the knowledge base and skills learned throughout the certification program to a real-world problem.

2 Acknowledgement

DISCLAIMER

The data set analysed in this project is owned and supplied by the Dutch data mining company Sentient Machine Research, and is based on real world business data. This data set and accompanying information is allowed to be used for non commercial research and education purposes only. It is explicitly not allowed to use this data set for commercial education or demonstration purposes. It is owned and donated by Peter van der Putten of the Dutch data mining company Sentient Machine Research, Baarsjesweg 224 1058 AA Amsterdam The Netherlands +31 20 6186927 putten@liacs.nl

This data set was used in the second edition of the Computational intelligence and Learning(CoIL) competition Challenge in the Year 2000, organized by CoIL cluster, which is a cooperation between four EU funded Networks of Excellence which represent the areas of neural networks (NeuroNet), fuzzy systems (ERUDIT), evolutionary computing (EvoNet) and machine learning (MLNet).

The data set was obtained from https://www.kaggle.com/uciml/caravan-insurance-challenge

3 Introduction

Insurance is a way to protect from financial loss of anything or any person. In the insurance industry, there are different kinds of products that are carefully put together by firms. These insurance products are financial agreements between a customer and an insurance provider where the latter will pay on any incurring claims on something based on certain terms and conditions. One such example of an insurance product is the coverage for caravans or mobile homes.

Insurance firms sell their products to customers through various kinds of marketing. It would help them if they can identify or classify the types of customers that will most likely buy their product. This would reduce their costs by not wasting time and resources approaching customers that will most likely not buy their product.

The analysis in this project aims to get a particular customer profile for those having caravan insurance. It is based on the data set that contains information about customers of an insurance company, and also the results of a marketing campaign "Caravan Insurance Policy", which has already been performed, that tells if customers were interested in this insurance policy or not. R will be used for the analysis and modelling. The prediction problem we are trying to answer is, given the data, what kind of profile will a customer have that most likely purchase caravan insurance. In addition, we are trying to develop a classification algorithm that can predict which customers from an unseen data set will get caravan insurance so that they can be targeted more by the insurance company while marketing

The data set is imbalanced which will be seen in the data exploration section. There are much more customers who did not want caravan insurance than those who were interested. This is a common issue in a lot of insurance data. Since we are trying to solve a classification problem, the main measures that will be used to check how effective a classifier model are the sensitivity, F1 score and AUC.

The sensitivity or the recall of a model measures the proportion of actual positives that are correctly identified. F1 Score is the weighted average of precision and recall. Therefore, this score takes both false positives and false negatives into account. F1 is usually more useful than the overall accuracy of a model, especially if you have an uneven class distribution. An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. AUC stands for "Area under the ROC Curve." One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example. This is what we need for the data set since we want to predict customers who would choose to get caravan insurance.

The key steps that were undertaken in this project have been summarised as below.

• Data Preparation: Here, we install the required R libraries, download the data file and create the required partitioned data sets.

• Data Exploration: Using various methods including visualisation and analysis techniques to understand the data set.

• Data Preprocessing: The data set consists of 85 predictors making it highly dimensional. Techniques such as eliminating variables with near-zero variance and checking for correlations between them are used to reduce the number of predictors.

• Modelling Process: Different machine learning techniques are explored to try find a suitable customer classification model. Some of the models are progressively built using cross validation and getting optimised parameters.

• Results and final testing: We apply the final model on the test data set, calculate the sensitivity, F1 score and AUC and check if we have achieved a suitable score.

• Conclusion: We highlight how the final model can be used, the limitations of the project and what could be the enhanced in the chosen model in the future.

4 Data Preparation

In order to do this project, a few R packages are installed and loaded as well. The data set is downloaded from the author's GitHub account as one of the requirements of the project is for it to be automatically downloaded or provided in a GitHub repo. The format of the data set is .csv.

```
urlfile="https://raw.githubusercontent.com/praveen556/DataScienceCapstone/master/caravan-insura
CIdata <- read_csv(url(urlfile))
head(CIdata)
```

```
## # A tibble: 6 x 87
##   ORIGIN MOSTYPE MAANTHUI MGEMOMV MGEMLEEF MOSHOOFD MGODRK MGODPR MGODOV MGODGE
##   <chr>    <dbl>    <dbl>   <dbl>    <dbl>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 train       33        1       3        2        8      0      5      1      3
## 2 train       37        1       2        2        8      1      4      1      4
## 3 train       37        1       2        2        8      0      4      2      4
## 4 train        9        1       3        3        3      2      3      2      4
## 5 train       40        1       4        2       10      1      4      1      4
## 6 train       23        1       2        1        5      0      5      0      5
## # ... with 77 more variables: MRELGE <dbl>, MRELSA <dbl>, MRELOV <dbl>,
## #   MFALLEEN <dbl>, MFGEKIND <dbl>, MFWEKIND <dbl>, MOPLHOOG <dbl>,
## #   MOPLMIDD <dbl>, MOPLLAAG <dbl>, MBERHOOG <dbl>, MBERZELF <dbl>,
## #   MBERBOER <dbl>, MBERMIDD <dbl>, MBERARBG <dbl>, MBERARBO <dbl>, MSKA <dbl>,
## #   MSKB1 <dbl>, MSKB2 <dbl>, MSKC <dbl>, MSKD <dbl>, MHHUUR <dbl>,
## #   MHKOOP <dbl>, MAUT1 <dbl>, MAUT2 <dbl>, MAUT0 <dbl>, MZFONDS <dbl>,
## #   MZPART <dbl>, MINKM30 <dbl>, MINK3045 <dbl>, MINK4575 <dbl>,
## #   MINK7512 <dbl>, MINK123M <dbl>, MINKGEM <dbl>, MKOOPKLA <dbl>,
## #   PWAPART <dbl>, PWABEDR <dbl>, PWALAND <dbl>, PPERSAUT <dbl>, PBESAUT <dbl>,
## #   PMOTSCO <dbl>, PVRAAUT <dbl>, PAANHANG <dbl>, PTRACTOR <dbl>, PWERKT <dbl>,
## #   PBROM <dbl>, PLEVEN <dbl>, PPERSONG <dbl>, PGEZONG <dbl>, PWAOREG <dbl>,
## #   PBRAND <dbl>, PZEILPL <dbl>, PPLEZIER <dbl>, PFIETS <dbl>, PINBOED <dbl>,
## #   PBYSTAND <dbl>, AWAPART <dbl>, AWABEDR <dbl>, AWALAND <dbl>,
## #   APERSAUT <dbl>, ABESAUT <dbl>, AMOTSCO <dbl>, AVRAAUT <dbl>,
## #   AAANHANG <dbl>, ATRACTOR <dbl>, AWERKT <dbl>, ABROM <dbl>, ALEVEN <dbl>,
## #   APERSONG <dbl>, AGEZONG <dbl>, AWAOREG <dbl>, ABRAND <dbl>, AZEILPL <dbl>,
## #   APLEZIER <dbl>, AFIETS <dbl>, AINBOED <dbl>, ABYSTAND <dbl>, CARAVAN <dbl>
```

From an initial analysis, the full Caravan Insurance data set has 9822 observations and 87 variables. There is no missing data. All the variables except ORIGIN are in a numeric format. From the data descriptions, certain fields were converted to categorical. The target variable is called Purchase. It is converted into a categorical variable that has two levels: "Yes" for those interested in caravan/ mobile home insurance and "No" for those who are not.

The data set is partitioned into two sets randomly, namely the train set and the test set. There is a column called ORIGIN that indicates which observations were provided initially as a training set in the competition. This is going to be ignored as there are now more observations available to train models especially since there are not many observations who are interested in caravan insurance. The train set will have 70% of the total observations. On this, all the descriptive and explorative

data analysis as well as the training of models will be done. The test set is kept aside to be used on the final model as unseen data to check how it fares with the chosen final model. Ideally, many machine learning exercises will try to have as much data observations in the training set to train models but since this data set is imbalanced, we do still need a sufficient size in the test set.

```
# Create training & test set
# These will be used for testing models
# The same seed is set for reproducibility

set.seed(1)

test_index <- createDataPartition(y = CIdata$Purchase, times = 1,
                                  p = 0.3, list = FALSE)
train_set <- CIdata[-test_index,]
test_set <- CIdata[test_index,]
```

4.1 Description of the Data Set

Originally, for the CoIL Challenge 2000 data mining competition, which the Caravan Insurance data set was provided for, was split into two parts: the training set and the test set. The test set was only provided after the end of the competition. The data set we have is the combined version of these two files. The data includes product usage data and socio-demographic data derived from zip area codes. It means that customers with the same zip code are characterized with the same socio-demographic features. The data contained a range of information on customers, which included income, age range,vehicle ownership, number of policies held, and level of contributions (premiums) paid as well as more qualitative information on lifestyle and type of households.

The field ORIGIN in the Caravan Insurance data set has the values train and test, corresponding to the training and test sets, respectively. The field CARAVAN is the target variable which indicates whether the customer purchase a caravan insurance policy or not.

The data file contains the following fields:

- ORIGIN: train or test, as described above
- MOSTYPE: Customer Sub type; see L0
- MAANTHUI: Number of houses; 1 - 10
- MGEMOMV: Average size household; 1 - 6
- MGEMLEEF: Average age; see L1
- MOSHOOFD: Customer main type; see L2

Percentages in each group, per postal code (see L3)

- MGODRK: Roman Catholic; see L3
- MGODPR: Protestant; see L3
- MGODOV: Other religion; see L3
- MGODGE: No religion; see L3
- MRELGE: Married; see L3
- MRELSA: Living together; see L3
- MRELOV: Other relation; see L3
- MFALLEEN: Singles; see L3
- MFGEKIND: Household without children; see L3
- MFWEKIND: Household with children; see L3
- MOPLHOOG: High level education; see L3
- MOPLMIDD: Medium level education; see L3
- MOPLLAAG: Lower level education; see L3
- MBERHOOG: High status; see L3
- MBERZELF: Entrepreneur; see L3
- MBERBOER: Farmer; see L3

- MBERMIDD: Middle management; see L3
- MBERARBG: Skilled labourers; see L3
- MBERARBO: Unskilled labourers; see L3
- MSKA: Social class A; see L3
- MSKB1: Social class B1; see L3
- MSKB2: Social class B2; see L3
- MSKC: Social class C; see L3
- MSKD: Social class D; see L3
- MHHUUR: Rented house; see L3
- MHKOOP: Home owners; see L3
- MAUT1: 1 car; see L3
- MAUT2: 2 cars; see L3
- MAUT0: No car; see L3
- MZFONDS: National Health Service; see L3
- MZPART: Private health insurance; see L3
- MINKM30: Income $<$ 30.000; see L3
- MINK3045: Income 30-45.000; see L3
- MINK4575: Income 45-75.000; see L3
- MINK7512: Income 75-122.000; see L3
- MINK123M: Income $>$123.000; see L3
- MINKGEM: Average income; see L3
- MKOOPKLA: Purchasing power class; see L3

Total number of variable in postal code (see L4):

- PWAPART: Contribution private third party insurance; see L4
- PWABEDR: Contribution third party insurance (firms); see L4
- PWALAND: Contribution third party insurance (agriculture); see L4
- PPERSAUT: Contribution car policies; see L4
- PBESAUT: Contribution delivery van policies; see L4
- PMOTSCO: Contribution motorcycle/scooter policies; see L4
- PVRAAUT: Contribution lorry policies; see L4
- PAANHANG: Contribution trailer policies; see L4
- PTRACTOR: Contribution tractor policies; see L4
- PWERKT: Contribution agricultural machines policies; see L4
- PBROM: Contribution moped policies; see L4

- PLEVEN: Contribution life insurances; see L4
- PPERSONG: Contribution private accident insurance policies; see L4
- PGEZONG: Contribution family accidents insurance policies; see L4
- PWAOREG: Contribution disability insurance policies; see L4
- PBRAND: Contribution fire policies; see L4
- PZEILPL: Contribution surfboard policies; see L4
- PPLEZIER: Contribution boat policies; see L4
- PFIETS: Contribution bicycle policies; see L4
- PINBOED: Contribution property insurance policies; see L4
- PBYSTAND: Contribution social security insurance policies; see L4
- AWAPART: Number of private third party insurance; 1 - 12
- AWABEDR: Number of third party insurance (firms)
- AWALAND: Number of third party insurance (agriculture)
- APERSAUT: Number of car policies
- ABESAUT: Number of delivery van policies
- AMOTSCO: Number of motorcycle/scooter policies
- AVRAAUT: Number of lorry policies
- AAANHANG: Number of trailer policies
- ATRACTOR: Number of tractor policies
- AWERKT: Number of agricultural machines policies
- ABROM: Number of moped policies
- ALEVEN: Number of life insurances
- APERSONG: Number of private accident insurance policies
- AGEZONG: Number of family accidents insurance policies
- AWAOREG: Number of disability insurance policies
- ABRAND: Number of fire policies
- AZEILPL: Number of surfboard policies
- APLEZIER: Number of boat policies
- AFIETS: Number of bicycle policies
- AINBOED: Number of property insurance policies
- ABYSTAND: Number of social security insurance policies
- Purchase: Number of mobile home policies 0 - 1

Keys (L1 - L4)

L0: Customer sub type

- 1: High Income, expensive child
- 2: Very Important Provincials
- 3: High status seniors
- 4: Affluent senior apartments
- 5: Mixed seniors
- 6: Career and childcare
- 7: Dinki's (double income no kids)
- 8: Middle class families
- 9: Modern, complete families
- 10: Stable family
- 11: Family starters
- 12: Affluent young families
- 13: Young all american family
- 14: Junior cosmopolitan
- 15: Senior cosmopolitans
- 16: Students in apartments
- 17: Fresh masters in the city
- 18: Single youth
- 19: Suburban youth
- 20: Ethnically diverse
- 21: Young urban have-nots
- 22: Mixed apartment dwellers
- 23: Young and rising
- 24: Young, low educated
- 25: Young seniors in the city
- 26: Own home elderly
- 27: Seniors in apartments
- 28: Residential elderly
- 29: Porch less seniors: no front yard
- 30: Religious elderly singles
- 31: Low income Catholics
- 32: Mixed seniors
- 33: Lower class large families
- 34: Large family, employed child

- 35: Village families
- 36: Couples with teens ‘Married with children’
- 37: Mixed small town dwellers
- 38: Traditional families
- 39: Large religious families
- 40: Large family farms
- 41: Mixed rurals

L1: average age keys:

- 1: 20-30 years
- 2: 30-40 years
- 3: 40-50 years
- 4: 50-60 years
- 5: 60-70 years
- 6: 70-80 years

L2: customer main type keys:

- 1: Successful hedonists
- 2: Driven Growers
- 3: Average Family
- 4: Career Loners
- 5: Living well
- 6: Cruising Seniors
- 7: Retired and Religious
- 8: Family with grown ups
- 9: Conservative families
- 10: Farmers

L3: percentage keys:

- 0: 0%
- 1: 1 - 10%
- 2: 11 - 23%
- 3: 24 - 36%
- 4: 37 - 49%
- 5: 50 - 62%
- 6: 63 - 75%
- 7: 76 - 88%

- 8: 89 - 99%
- 9: 100%

L4: total number keys:

- 0: 0
- 1: 1 - 49
- 2: 50 - 99
- 3: 100 - 199
- 4: 200 - 499
- 5: 500 - 999
- 6: 1000 - 4999
- 7: 5000 - 9999
- 8: 10,000 - 19,999
- 9: >= 20,000

5 Data Exploration and Analysis

Now that the Caravan Insurance data set is partitioned, we can do some exploration of the train data set to understand the data more. The main idea of data exploration and analysis is to check which variables have a significant effect on the target variable, Purchase, and reduce uncertainty caused by anomalies.

```
nrow(train_set)
```

```
## [1] 6875
```

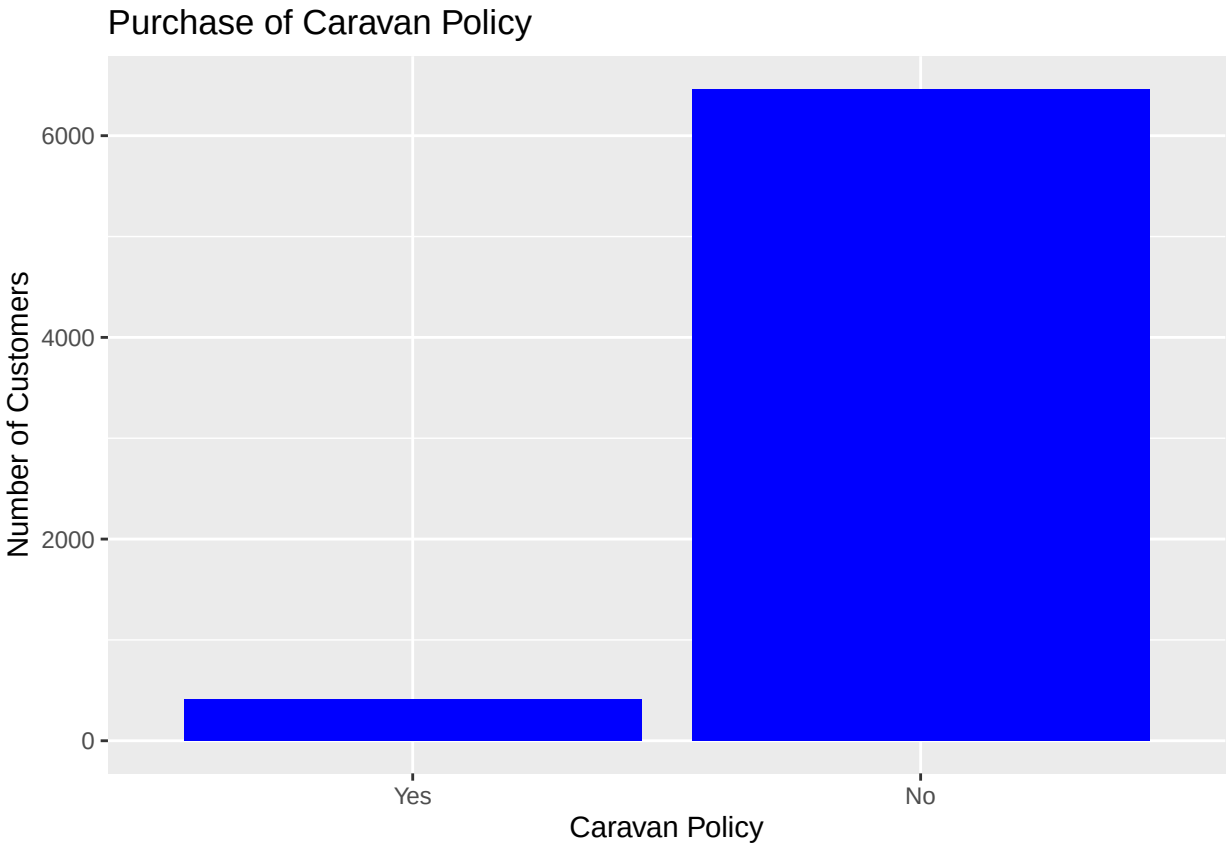
```
# Remove ORIGIN variable from the train and test sets
train_set$ORIGIN <- NULL
test_set$ORIGIN <- NULL
```

The train set has 6875 observations. The variable called ORIGIN is removed from both the train and test sets since the data set was partitioned randomly without using the classification given. So now, there are 85 predictors for the target variable, Purchase. This makes the data set highly dimensional. So, it needs to be explored whether some of the predictors can be removed which do not affect Purchase. It may be the case that probably not all of them are useful, or some of them presents the same information, or even some of them might not bring any value to the predictive model that will be built.

In the table below, it can be seen that the data set is very imbalanced. There is only 5.96% of the observations that are actually interested in caravan insurance. It is not surprising that the data is imbalanced because in most marketing campaigns for any product, there will be more non-buyers than ones who respond positively. Hence, the ROC curve and the F1 score will be used as measures to check how good a prediction model is later rather than the overall accuracy. If only roughly 6% are people that purchased insurance, simply predicting that everyone did not get caravan insurance will produce an accuracy of about 94% which is very good for a model based on a regular balanced data set. However, the aim is for the insurance company to get an idea of who will buy so that they can be targeted more in their marketing campaign.

Purchase	n	prop
Yes	410	0.0596364
No	6465	0.9403636

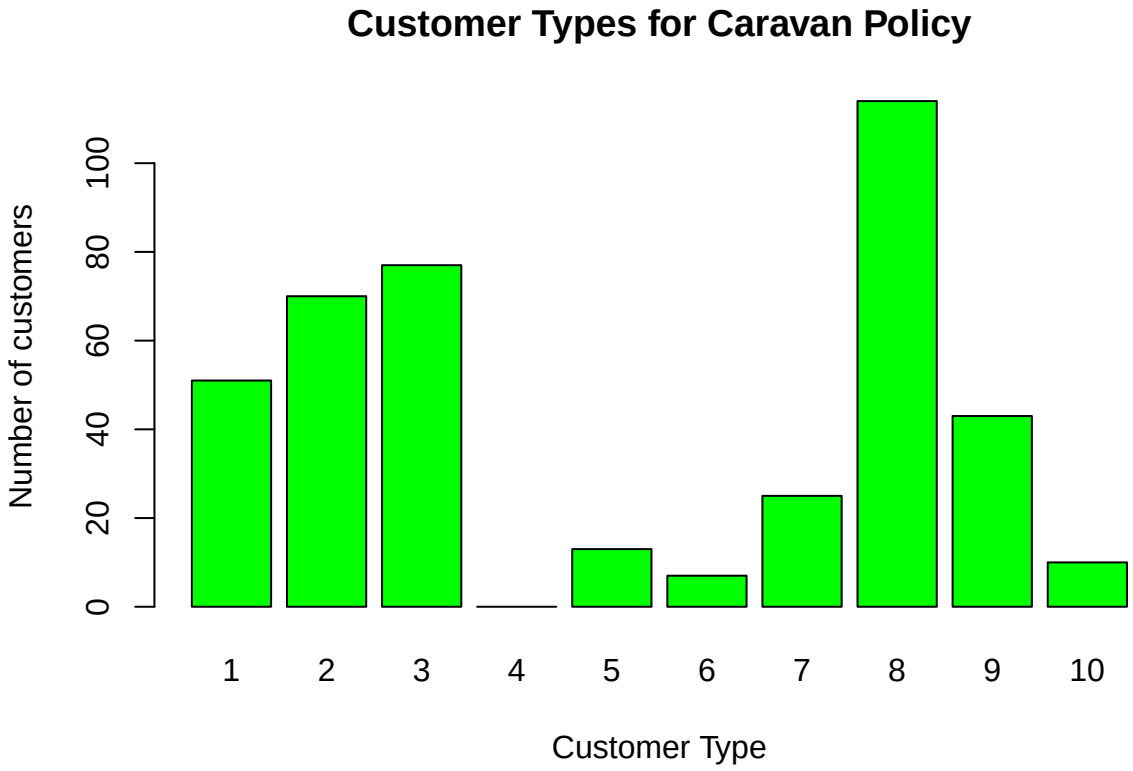
The following is a visualisation that confirms that the data set is indeed not equal by plotting the numbers of the two categories of the target variable, Purchase.



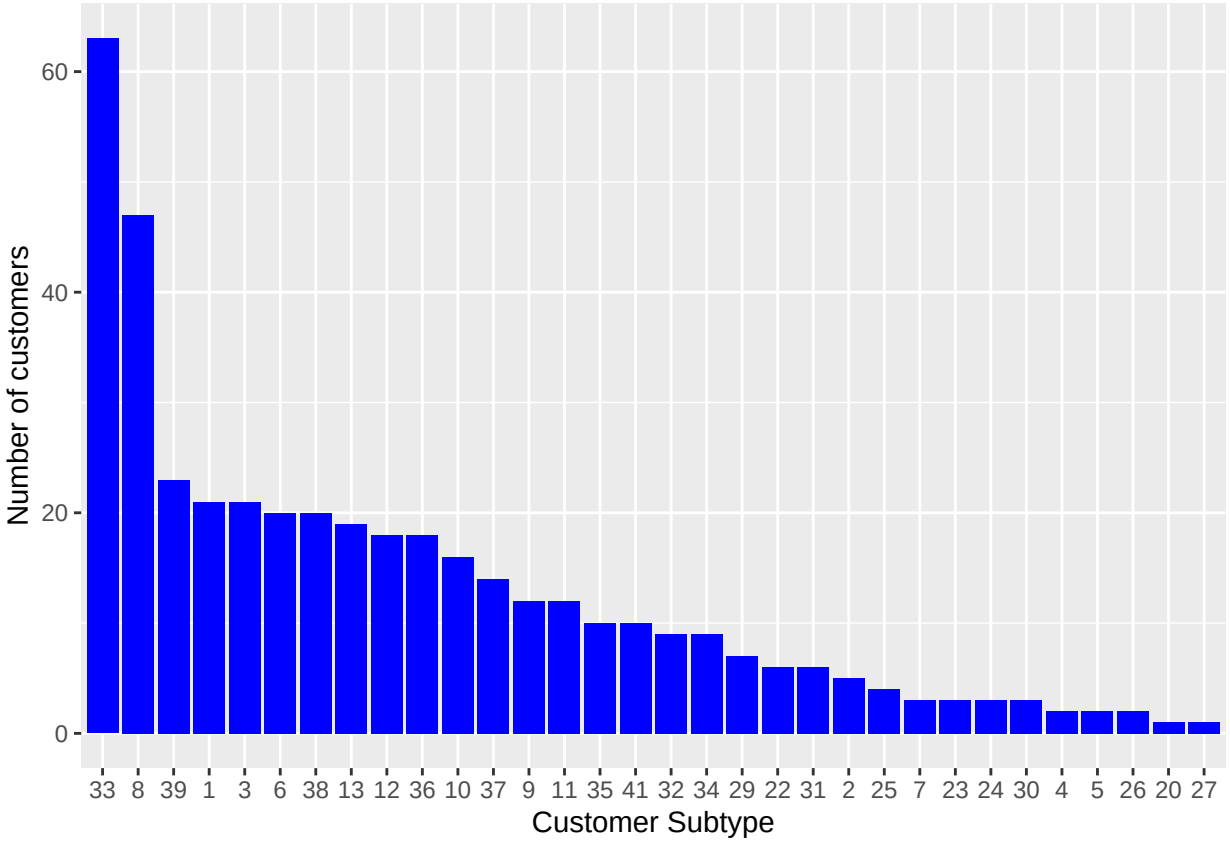
5.1 Data Visualisation

From just looking at the data, it is not possible to see if there are any relationships between the predictors and Purchase. So, mapping the variables against the target variable and seeing it as a plot helps. However, having this many variables proves it difficult to check each relationship especially since there is not much data on those that actually have caravan insurance. Going by intuition and using business sense, a few variables were selected to create plots.

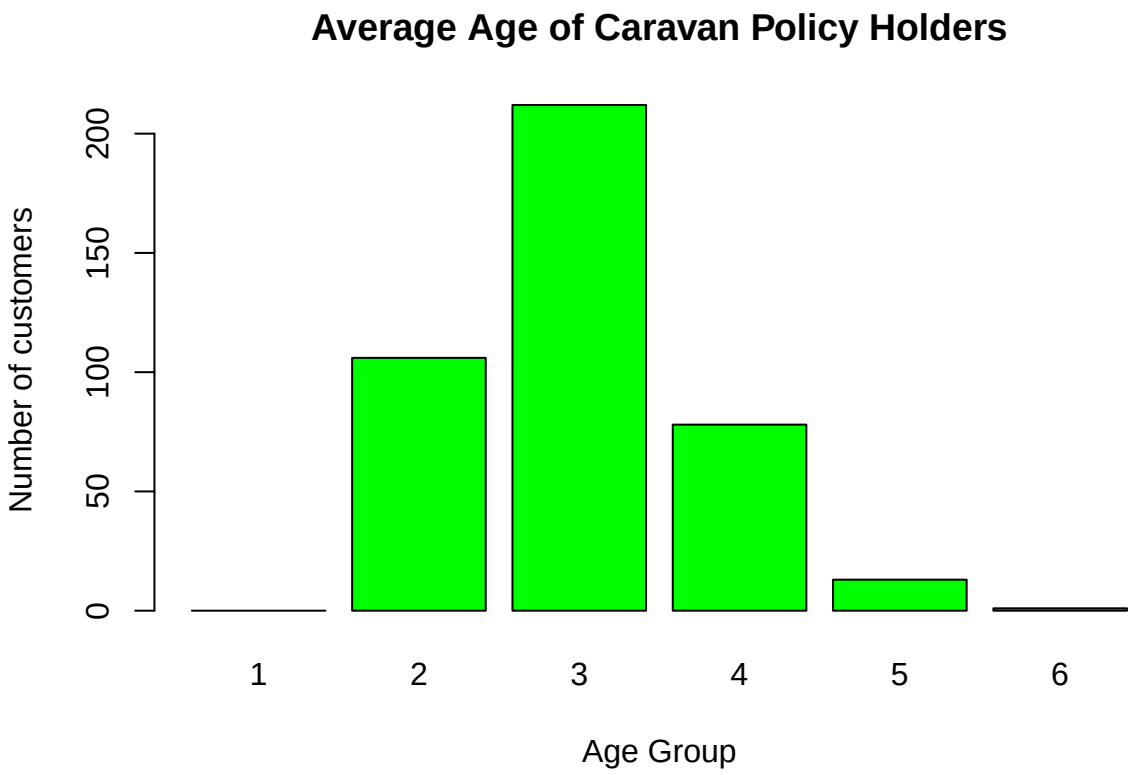
The following plot is a histogram showing which categories of Customer types are more likely to get caravan insurance. Using the key provided before in the description of the data set, we can see that the group that has the most insurance policies are Families with grown ups followed by Average Families and Driven Growers. Career Loners and Cruising Seniors are the most unlikely to have caravan insurance.



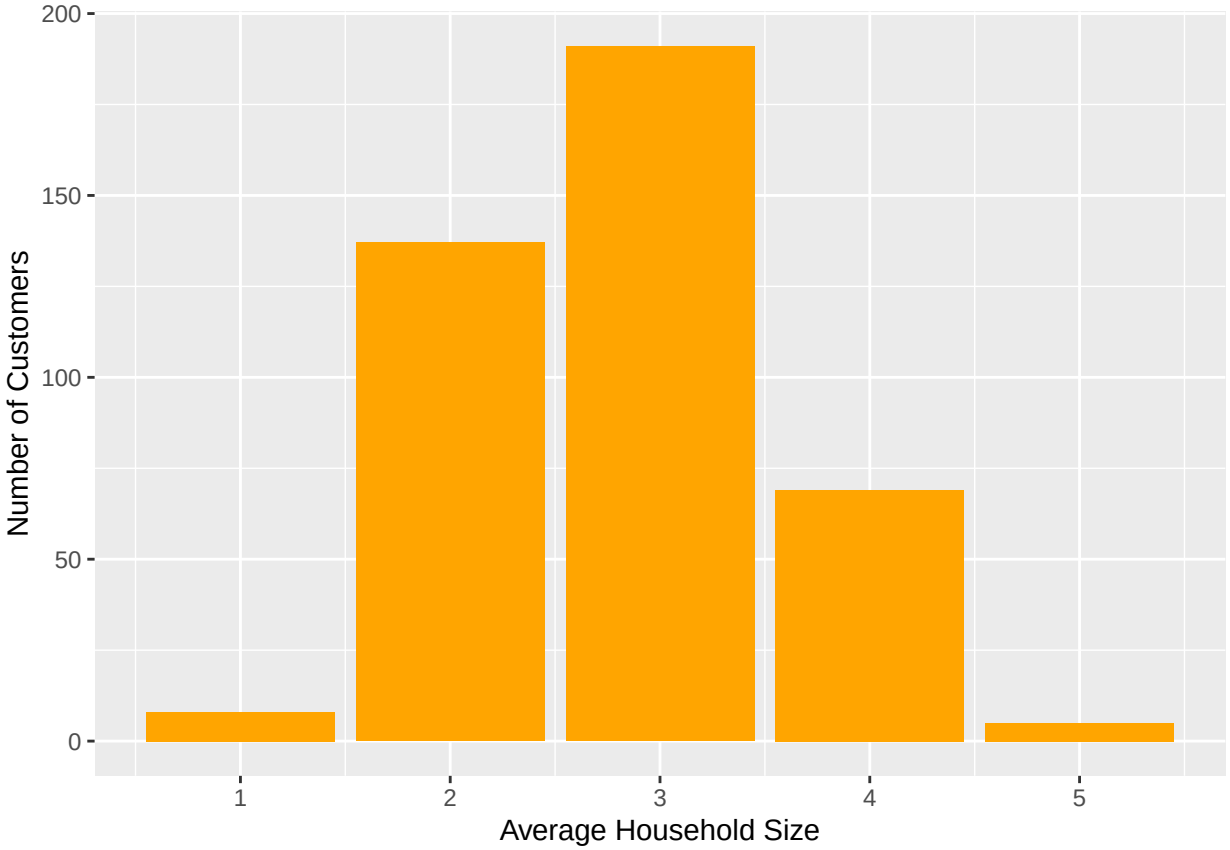
The following plot shows which categories of the 41 Customer sub types are more likely to get caravan insurance. It is plotted with the category having the highest policies first so that it is easier to analyse. It shows that Lower class large families and Middle class families have the most policies. Seniors that have affluent apartments and those who are ethnically diverse are the least likely to have insurance.



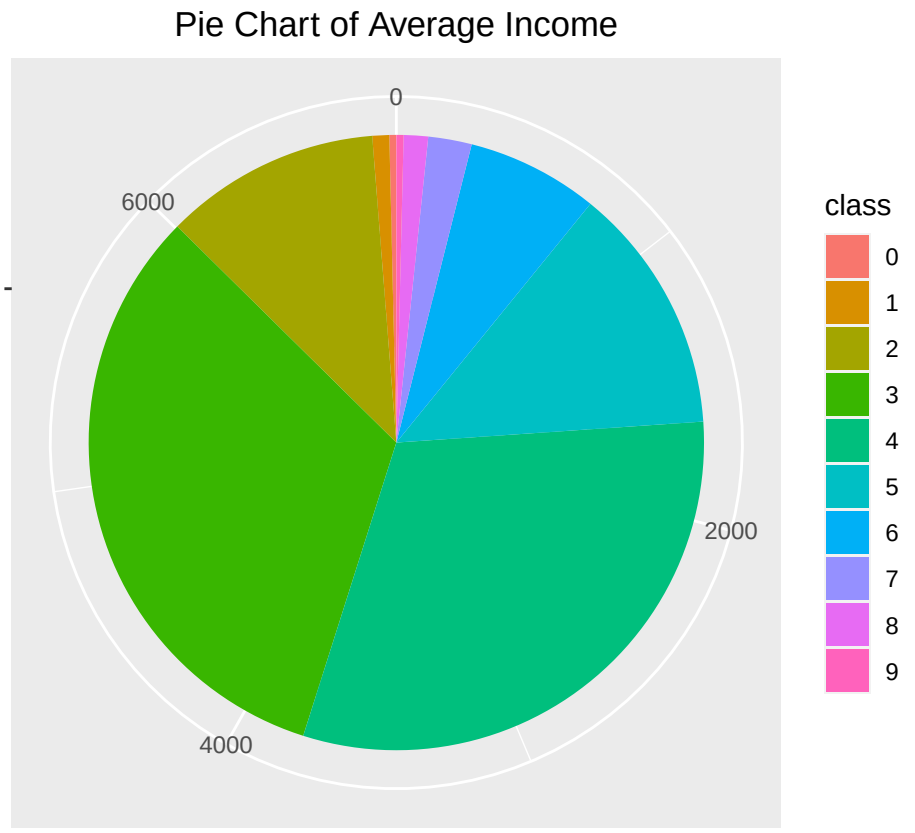
The following plot shows that middle aged people are the most likely to get caravan insurance especially in the age category of 40-50 years. People under 30 and over 70 years do not have a mobile home policies. This lines up with the observations made in the previous plots.



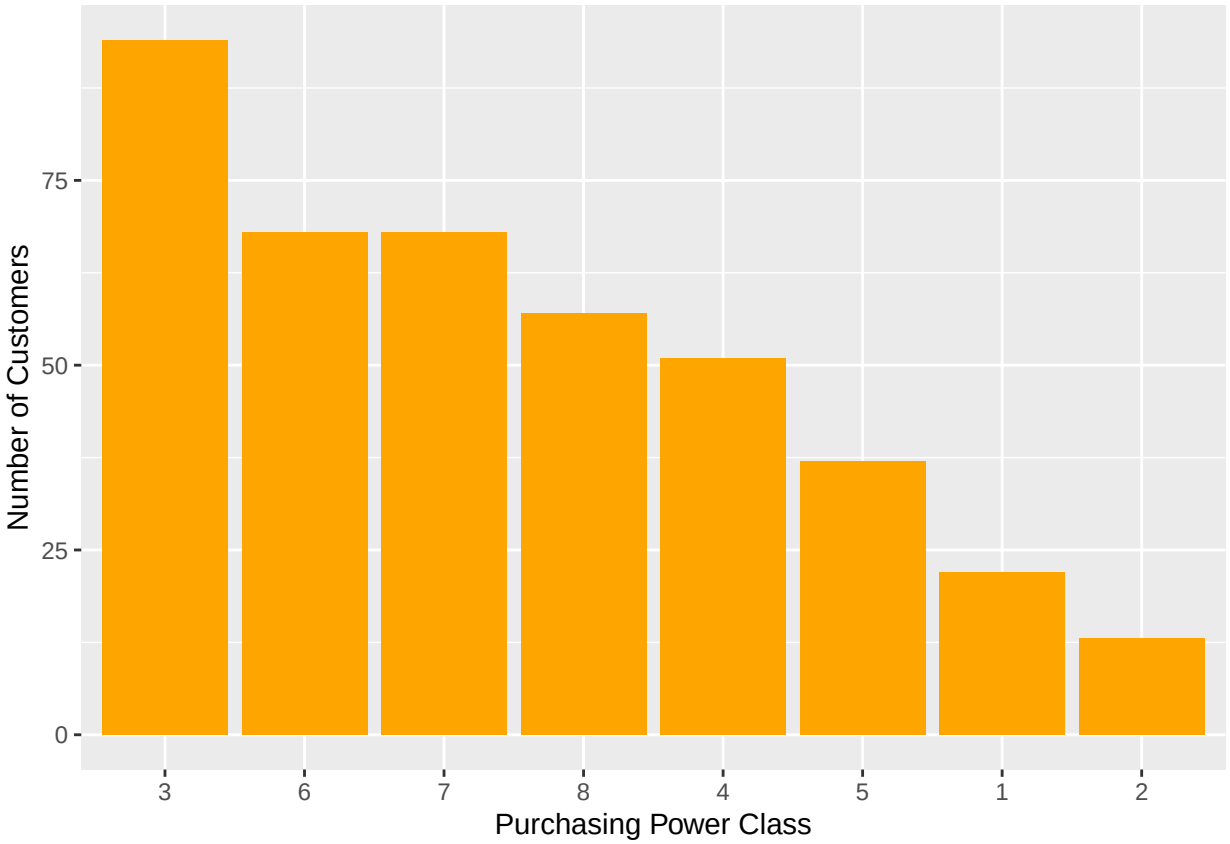
The following plot shows that households that have three people are the most likely to get caravan insurance which is surprising because it was noted that lower class large families are ones that have the most.



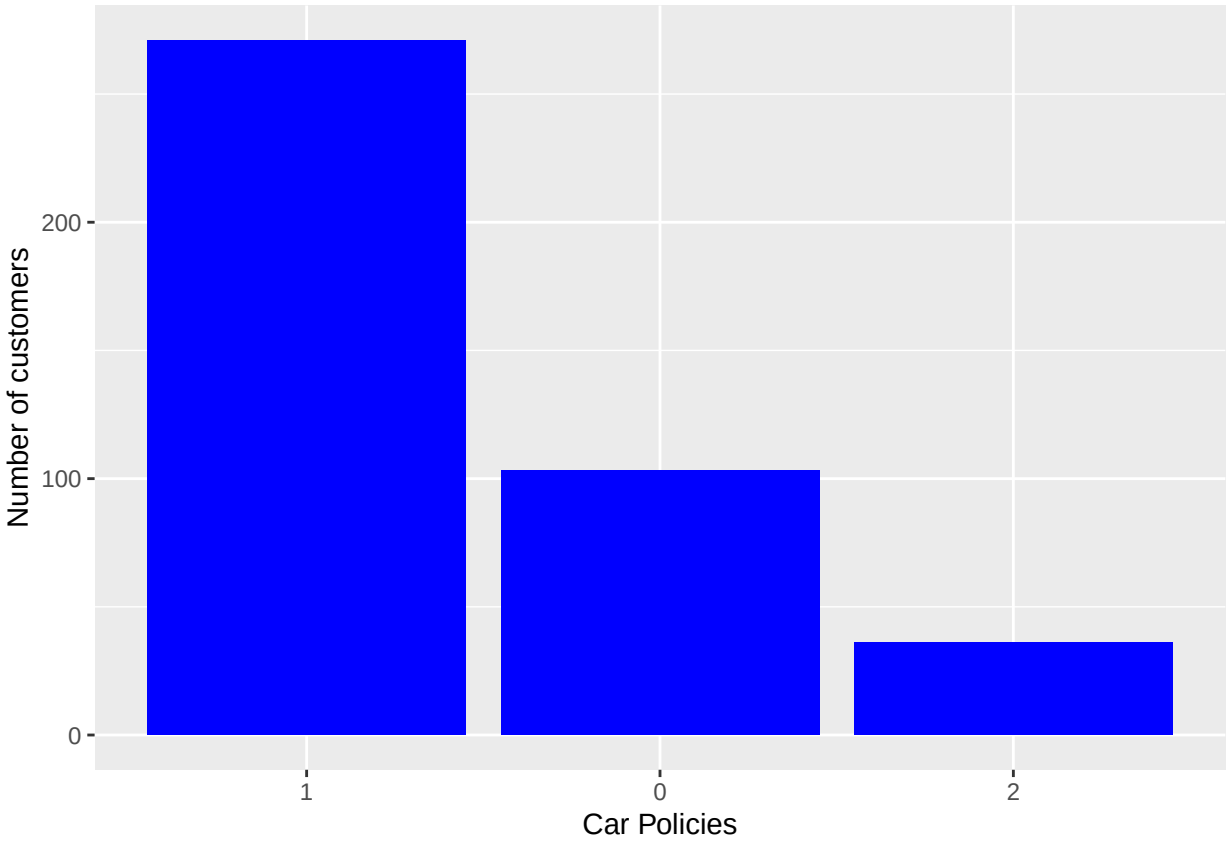
The following plot is a pie chart showing the different average income levels against the number of people who have insurance. It can be noted that those with average middle scale income (24 to 49%) are more likely to get a policy. It makes up for more than 50% of the existing polices.



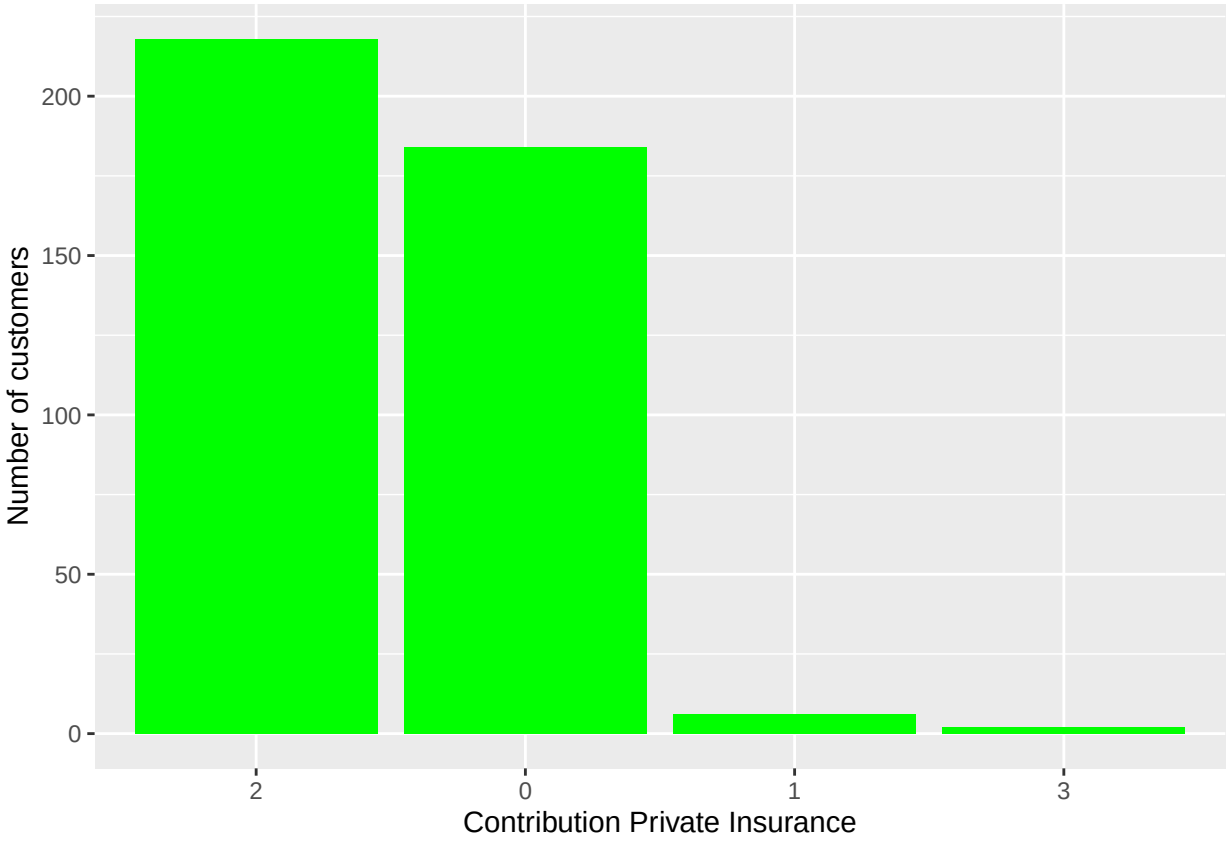
The following is a plot that shows the relationship between purchasing power class and Purchase. Individuals who can afford to buy high cost products such as caravan insurance which is not an essential need are more likely to get a policy.



One of the most common insurance products that people need to get are for their cars. The following plot shows that people who own one car is the most likely to get caravan insurance as well. It shows their purchasing power and higher income as well as their general interest to drive. So they would be more inclined to get a caravan and its corresponding insurance as well. People who have more than a car are less likely to have a caravan.



The following plot shows that those who have caravan insurance will be holders of other private insurance policies as well.

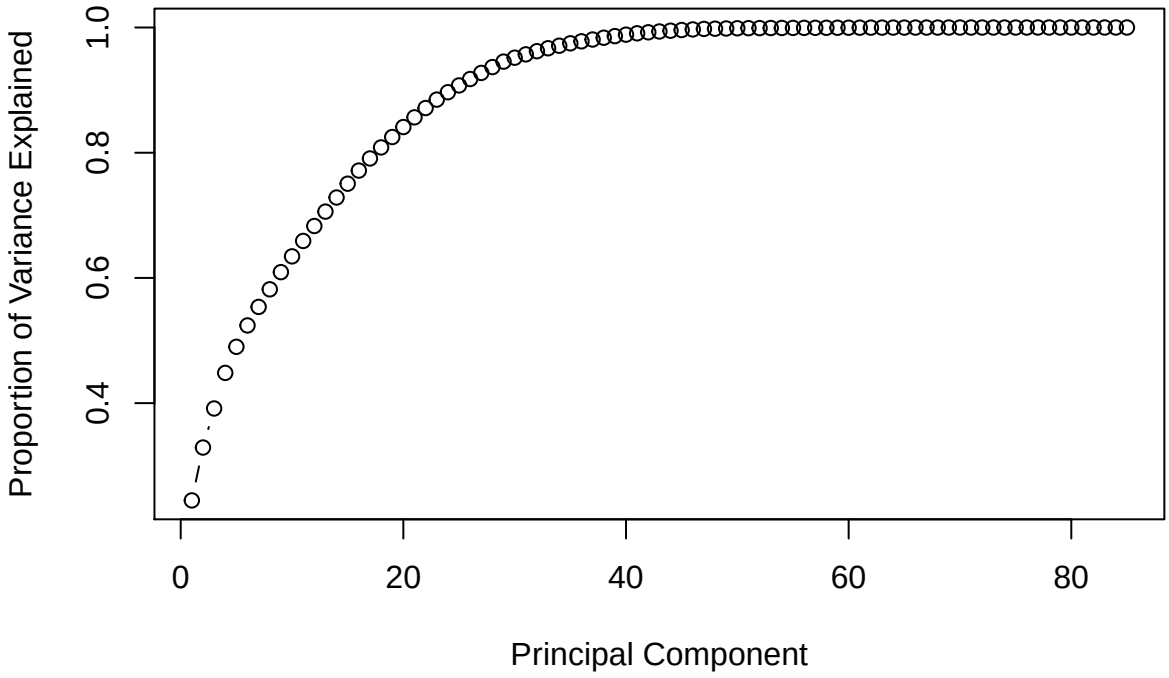


5.2 Data Preprocessing

One of the challenges of the Caravan Insurance data set is that it has a large number of predictors. In the earlier section, a few plots were shown as part of data visualisation. However, to reveal relationships between many variables are much more complicated in higher dimensions. In machine learning, predictors that are not useful are removed before running the machine algorithm. The steps taken are called preprocessing.

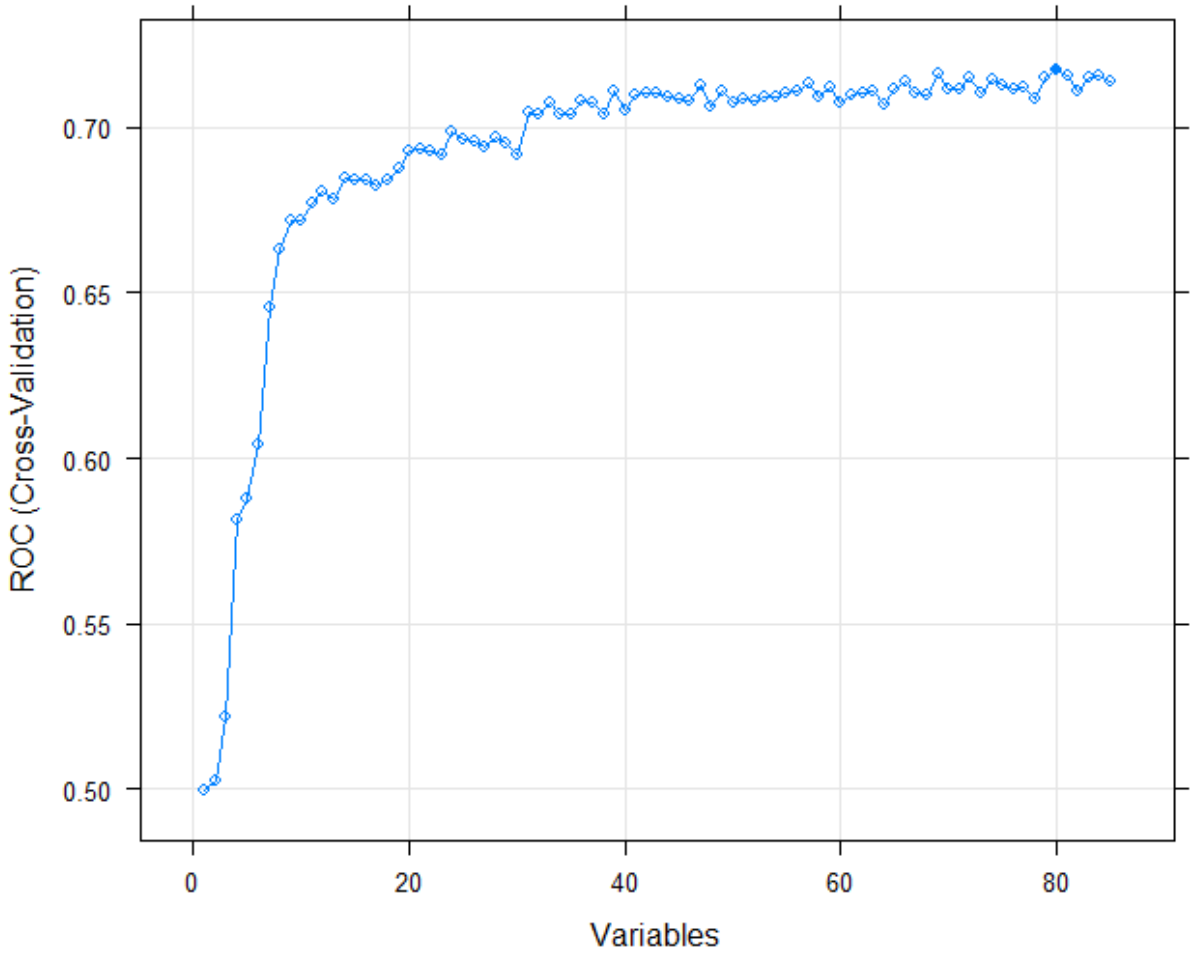
One of the steps would be to objectively test the impact of the variables on Purchase. This will be done using Principal Component Analysis (PCA). PCA is a method of linear transformation that aims to detect the correlation between variables. This is done by computing the variance "explained" by each of the variables in multidimensional data. If such a strong correlation exists, then the logical course of action would be a reduction in the number of dimensions while still retaining the majority of the pertinent information.

The first step in the PCA process would be to standardise the data. So, all the variables are converted into a numeric format. The following plot shows the number of principal components against the propotion of variance explained in the data. We can see that the first 44 dimensions account for 99.5% of the variability. So, we can conclude that there is definitely a scope to reduce the number of features.



The next feature selection technique that was used was the Recursive Feature Elimination method. This technique begins by building a model on the entire set of predictors and computing an importance score for each predictor. The cross-validation tries to maximise the ROC measure. The

least important predictor(s) are then removed, the model is rebuilt, and the importance scores are computed again. This resulted in the optimal subset of 80 predictors. However, a good ROC value is above 0.70. This can be achieved with 31 variables. The plot of the number of variables against the values of ROC can be seen below.



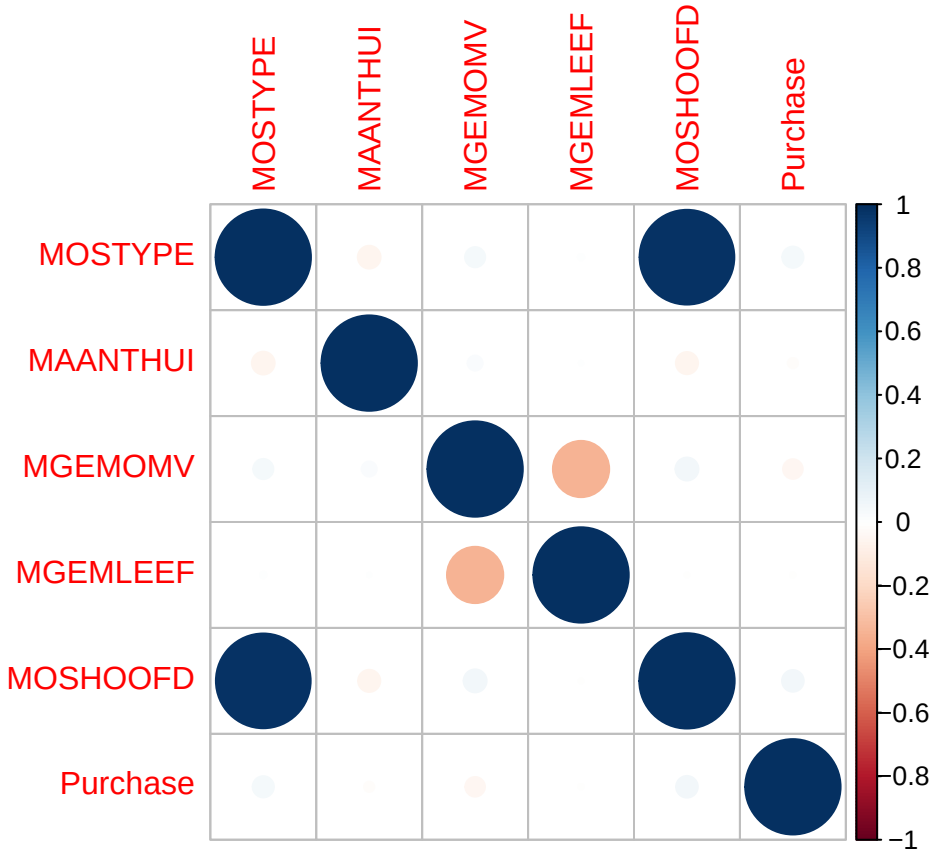
The caret package in R includes a function called nearZero that recommends features to be removed due to the fact that several predictors do not vary much from observation to observation. Using this function resulted in a list of 35 variables from the total of 85 that were suggested for elimination.

Stepwise regression analysis was performed on the train data set. It is a combination of the forward and backward selection techniques. It uses a statistic called Akaike information criterion (AIC). The AIC statistic is used for models that use the likelihood as the objective (we use logistic regression) and penalises the likelihood by the number of parameters included in the model. The goal is to minimise the AIC value. Using this method resulted in a selection of 35 variables.

One of the faults of using stepwise regression analysis is if highly correlated predictors are there in the data set, it will result in an over-selection of features. Therefore, the variables were divided into groups to check for high inter-correlation. The method of Spearman correlation was used since the

variables are not normally distributed and the relationship between the variables is not linear. It can be seen in the correlation matrices and plots below that there is high correlation between the following pairs: the Customer main type and sub type, renting a home versus owning a home and having private versus public health insurance.

```
##                MOSTYPE     MAANTHUI      MGEMOMV      MGEMLEEF      MOSHOOFD
## MOSTYPE    1.000000000 -0.056919667  0.04451774  0.004383286  0.987532072
## MAANTHUI  -0.056919667  1.000000000  0.02464779  0.002110357 -0.053779458
## MGEMOMV    0.044517736  0.024647787  1.00000000 -0.348062558  0.057987692
## MGEMLEEF   0.004383286  0.002110357 -0.34806256  1.000000000 -0.002581397
## MOSHOOFD   0.987532072 -0.053779458  0.05798769 -0.002581397  1.000000000
## Purchase   0.049305864 -0.011484557 -0.04228718 -0.003180497  0.051580832
##                Purchase
## MOSTYPE    0.049305864
## MAANTHUI  -0.011484557
## MGEMOMV   -0.042287185
## MGEMLEEF  -0.003180497
## MOSHOOFD   0.051580832
## Purchase   1.000000000
```




```
##                MHHUUR      MHKOOP    Purchase
## MHHUUR     1.00000000 -0.99958499  0.07947762
```

```
## MHKOOP   -0.99958499  1.00000000 -0.07906197
## Purchase  0.07947762 -0.07906197  1.00000000
```

```
##               MZFONDS      MZPART    Purchase
## MZFONDS    1.00000000 -0.99926592  0.06465842
## MZPART    -0.99926592  1.00000000 -0.06390042
## Purchase   0.06465842 -0.06390042  1.00000000
```

The aim of variable selection is parsimony. This is try to achieve a balance between simplicity (as few predictors as possible) and fit (as many predictors as needed). Using the combination of all these feature selection methods, a subset of the train data set was created that has 35 predictors and the target variable. So, a total of 50 variables were removed which equates to a 58.8% reduction.

```
stepwise_train_set <- train_set[, c("MGODOV", "MGODGE", "MOPLMIDD", "MOPLLAAG",
                                    "MRELGE","MBERZELF", "MBERBOER", "MBERMIDD",
                                    "MHHUUR", "MFWEKIND", "MAUT1", "MINK123M",
                                    "PWAPART", "PPERSAUT", "PLEVEN", "PPERSONG",
                                    "PGEZONG", "PWAOREG", "PBRAND", "PZEILPL",
                                    "PPLEZIER", "PINBOED", "PBYSTAND", "AWAPART",
                                    "AWALAND", "ABESAUT", "AWERKT", "ALEVEN",
                                    "APERSONG", "AGEZONG", "ABRAND", "AFIETS",
                                    "AINBOED", "ABYSTAND", "MOSHOOFD", "Purchase")]

train_set2 <- stepwise_train_set
```

6 Modelling Process

The aim of this project is a classification of a type of customer profile that will respond positively to marketing campaigns of caravan insurance policies. So, a variety of different classification algorithms were employed: Logistic Regression, K nearest neighbour classification and Decision Trees. The ensemble method, Random Forest, was used for improving on the single tree classifier model: Decision Tree. The main measures that will be used to assess the different models are the sensitivity, F1-score and the AUC. For each model that is tried, a copy of the training set is created so that by the end of all the training and validation, it does not become over-tuned.

6.1 Further Partitioning

The train data set is further partitioned into separate training and validation sets to design and test the machine learning algorithms. This will allow the test data set to be only used to check the efficacy of the final model. Again, the train set will be split with 70% of its observations in the training set and 30% in the validation set.

```
# Create training & validation set from the original train data set
# These will be used for testing models
# The same seed is set for reproducibility

set.seed(1)

test_index <- createDataPartition(y = train_set2$Purchase, times = 1,
                                  p = 0.3, list = FALSE)
training_set <- train_set2[-test_index,]
validation_set <- train_set2[test_index,]

nrow(training_set)
```

```
## [1] 4812
```

```
nrow(validation_set)
```

```
## [1] 2063
```

```
mean(training_set$Purchase == "Yes")
```

```
## [1] 0.05964256
```

The training set has 4812 observations and the validation set has 2063. In the training set, 5.96% of the observations are those that have caravan insurance.

6.2 Logistic Regression

The first classifier model that will be tried is Logistic Regression, a specific case of a set of generalised linear models. This is used because the target variable is categorical. The model involves transforming the output of a linear regression by using a logit link function and the t-statistic will be used as the criteria for selecting variables.

```
training_glm <- training_set

set.seed(3) # for reproducibility

control <- trainControl(summaryFunction= twoClassSummary, classProbs = TRUE)

train_glm <- train(Purchase ~ .,
                   method = "glm",
                   family = binomial,
                   data = training_glm,
                   trControl = control,
                   metric = "ROC")

train_glm
```

```
## Generalized Linear Model
##
## 4812 samples
##   35 predictor
##    2 classes: 'Yes', 'No'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 4812, 4812, 4812, 4812, 4812, 4812, ...
## Resampling results:
##
##   ROC       Sens        Spec
##   0.700828  0.01585469  0.9978907
```

```
y_hat_glm <- predict(train_glm, validation_set)
summary(y_hat_glm)
```

```
##  Yes   No
##    4 2059
```

```
cm_glm <- confusionMatrix(y_hat_glm, validation_set$Purchase)
cm_glm$table
```

```
##           Reference
```

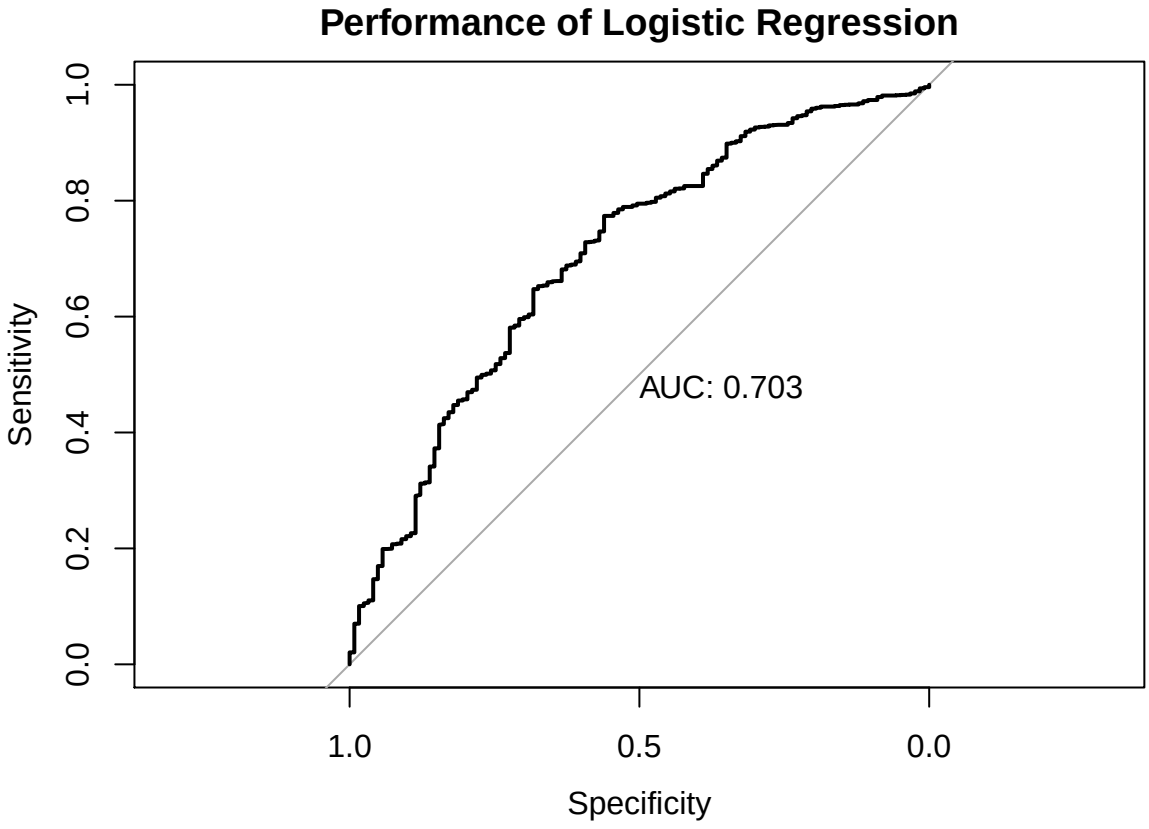
```
## Prediction  Yes   No
##        Yes    0    4
##        No   123 1936
```

Above, the confusion matrix for Logistic Regression can be seen. This shows all the true positive (TP), the false negative (FN), the false positive (FP) and the true negative (TN) values of the classifier model. Based on this, the measures for the model can be calculated. This can be seen below.

Method	Accuracy	Sensitivity	Specificity	F1_score
Generalised Linear Model	0.9384392	0	0.9979381	NaN

The measures show that the sensitivity and the F1 score is quite close to 0 which is not what is needed. It shows that the model is not capable to identify the people that would get insurance well.

The plot below shows the ROC curve and the AUC of the model. The AUC takes values from 0 to 1, where a value of 0 indicates a perfectly inaccurate test and a value of 1 reflects a perfectly accurate test. So the curve should be as far up from the diagonal line. It can be seen that the model results in a good AUC score but since its sensitivity and the F1 score is very low, other models need to be considered.



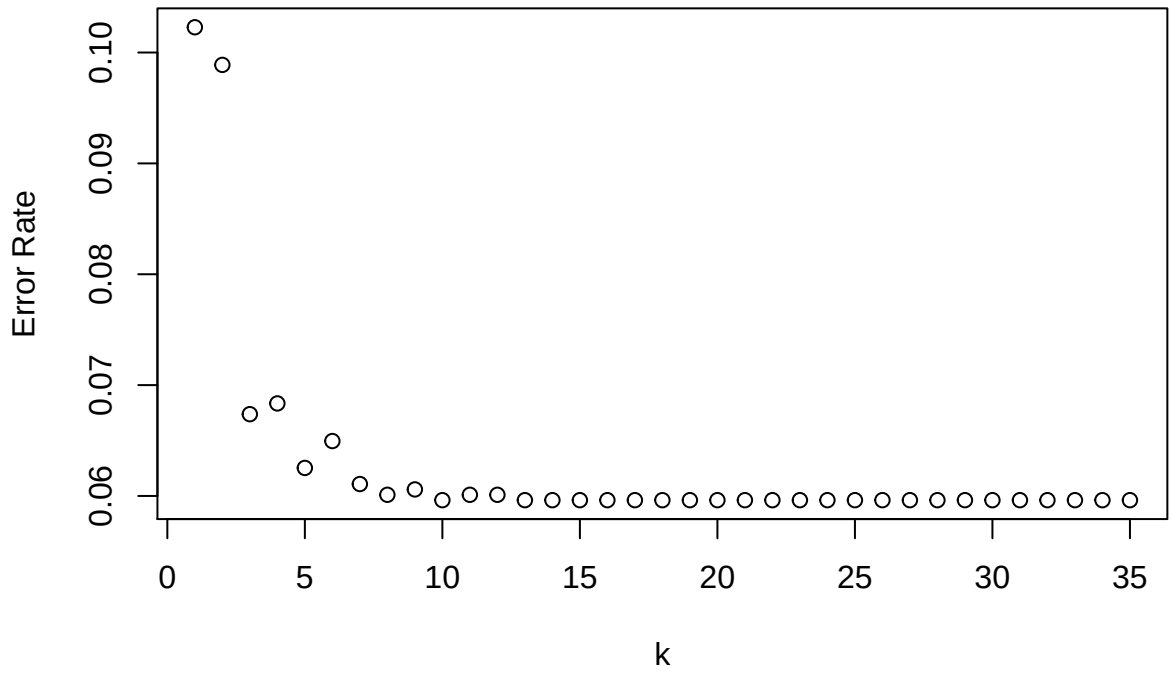
6.3 K Nearest Neigbour

The k-nearest neighbors algorithm (kNN) is a non-parametric method used for classification and regression. It uses the similarity of features to predict the values of new data points. The new data points will be allocated a distance value based on how closely it matches the points in the training set.

One major drawback in calculating distance measures directly from the training set is in the case where variables have different measurement scales or there is a mixture of numerical and categorical variables. So, the variables are standardised by normalising them. In addition, in order to not over-train or over-smooth the model, cross-validation is used to choose the value of the tuning parameter k. The value of k will be chosen based on which one minimises the error rates on the validation set. The plot of the k values and the error rates can be seen below.

```
training_knn <- training_set %>% mutate_all(as.numeric)
validation_knn <- validation_set %>% mutate_all(as.numeric)

#Normalising each variable so they are scaled
normalise <- function(x)
{
  u <- mean(x)
  o <- sd(x)
  (x - u)/(o)
}

train_knn_normalised <- apply(training_knn, 2, normalise)
val_knn_normalised <- apply(validation_knn, 2, normalise)

train_input <- as.matrix(train_knn_normalised[,-36])
train_output <- as.vector(train_knn_normalised[,36])
test_input <- as.matrix(val_knn_normalised[,-36])
test_output <- as.matrix(val_knn_normalised[,36])

set.seed(5) # for reproducibility

#pick the k in knn using cross validation

kmax <- 35
ER1 <- rep(0, kmax)
ER2 <- rep(0, kmax)
for (k in 1:kmax){
  prediction <- knn(train_input, train_input, train_output, k=k)
  prediction2 <- knn(train_input, test_input, train_output, k=k)

  #Confusion Matrix For Test Data
  cmtest <- table(prediction2, val_knn_normalised[,"Purchase"])

  #Error Rate on the validation sample
```

```
  ER2[k] <- (cmtest[1,2]+cmtest[2,1])/sum(cmtest)
}


# Minimum Validation Error k
plot(1:kmax, ER2, xlab = "k", ylab = "Error Rate")
```




```
best_k <- which.min(ER2)
best_k
```

```
## [1] 10
```

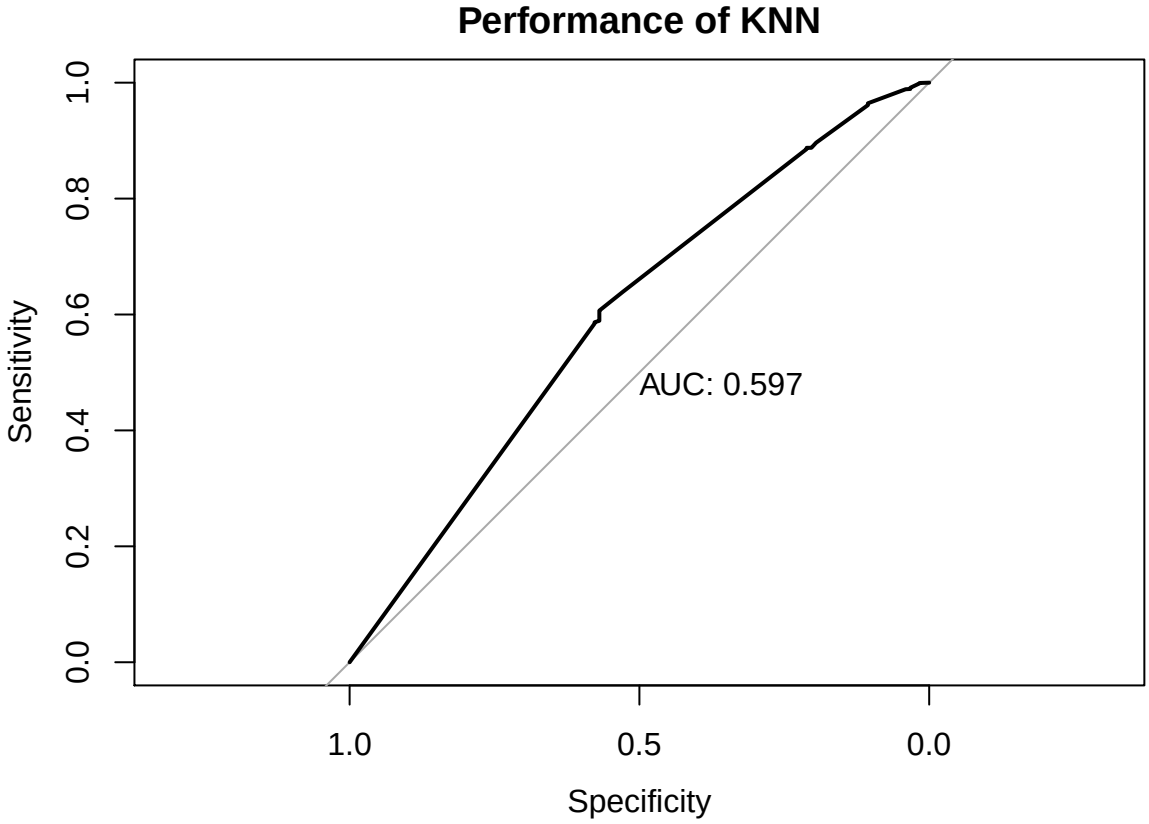
```
fit_knn <- knn(train_input, test_input, train_output, k= best_k)
actual <- ifelse(val_knn_normalised[,"Purchase"]> 0, 1, 0)
predicted <- ifelse(as.numeric(fit_knn)>0, 0, 1)
u <- union(predicted, actual)
t <- table(factor(predicted, u), factor(actual, u))
cm_test <- confusionMatrix(t)
cm_test$table
```

```
##
```

```
##         0    1
##    0  123 1940
##    1    0    0
```

The measures for the model can be seen below as well as the ROC curve. The sensitivity is a perfect score but the accuracy of the model is hugely compromised. So, other models will be considered.

Method	**Accuracy**	**Sensitivity**	**Specificity**	**F1_score**
Generalised Linear Model	0.9384392	0	0.9979381	NaN
K Nearest Neighbour	0.0596219	1	0.0000000	0.1125343



6.4 Decision Tree

Classification trees, or decision trees, are used in prediction problems where the outcome is categorical. A tree is essentially a flow chart of yes or no questions. The algorithm created uses the training data to create these trees with predictions at the ends, referred to as nodes. The decision tree will predict the target variable, Purchase, by partitioning the other variables.

Decision trees are widely used in machine learning because the output is easy to understand even for non-professional users. In this case, the trees will allow for customer segmentation so that we can see what characteristics a caravan insurance policy customer has.

```
training_rpart <- training_set

set.seed(7) # for reproducibility

training_set$Purchase = as.factor(training_set$Purchase)

train_rpart <- rpart(formula=Purchase ~ .,
                     data = training_rpart,
                     control = rpart.control(minsplit = 26, cp = 0.001),
                     method = "class")

print(train_rpart)
```

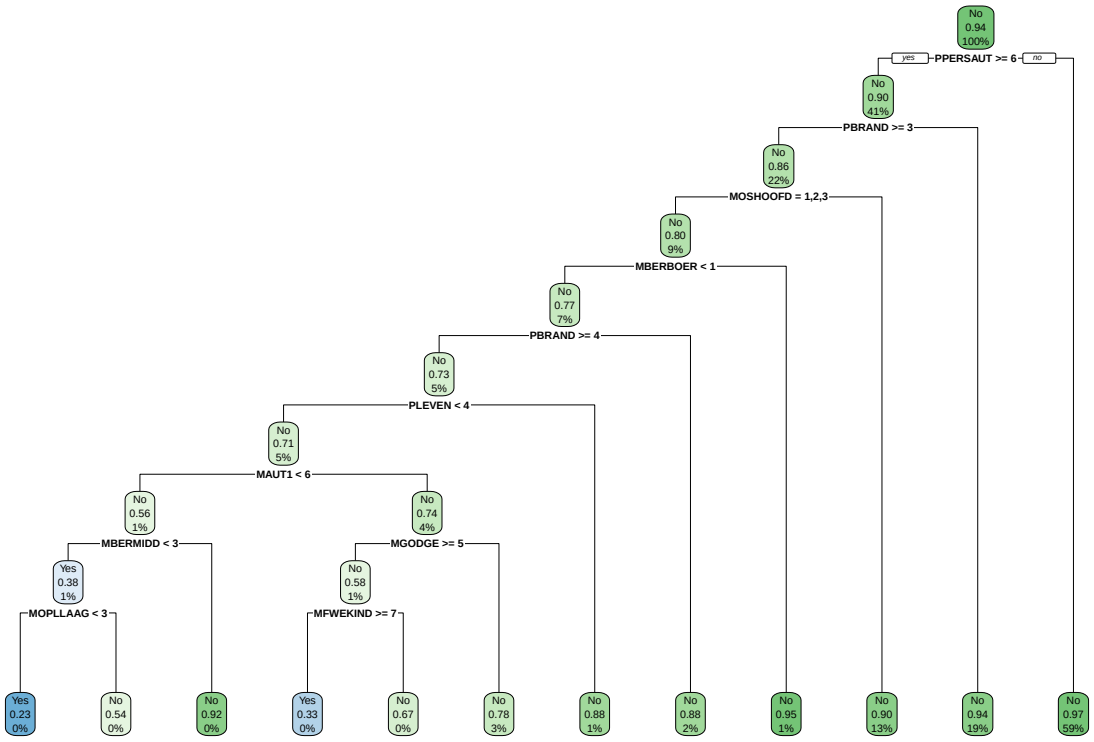
```
## n= 4812
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 4812 287 No (0.05964256 0.94035744)
##     2) PPERSAUT>=5.5 1973 204 No (0.10339584 0.89660416)
##       4) PBRAND>=2.5 1040 146 No (0.14038462 0.85961538)
##         8) MOSHOOFD=1,2,3 410  83 No (0.20243902 0.79756098)
##          16) MBERBOER< 0.5 353  80 No (0.22662890 0.77337110)
##            32) PBRAND>=3.5 258  69 No (0.26744186 0.73255814)
##              64) PLEVEN< 3.5 224  65 No (0.29017857 0.70982143)
##               128) MAUT1< 5.5 39  17 No (0.43589744 0.56410256)
##                 256) MBERMIDD< 2.5 26  10 Yes (0.61538462 0.38461538)
##                   512) MOPLLAAG< 3 13   3 Yes (0.76923077 0.23076923) *
##                   513) MOPLLAAG>=3 13   6 No (0.46153846 0.53846154) *
##                 257) MBERMIDD>=2.5 13   1 No (0.07692308 0.92307692) *
##               129) MAUT1>=5.5 185  48 No (0.25945946 0.74054054)
##                 258) MGODGE>=4.5 33  14 No (0.42424242 0.57575758)
##                   516) MFWEKIND>=6.5 9   3 Yes (0.66666667 0.33333333) *
##                   517) MFWEKIND< 6.5 24   8 No (0.33333333 0.66666667) *
##                 259) MGODGE< 4.5 152  34 No (0.22368421 0.77631579) *
##              65) PLEVEN>=3.5 34   4 No (0.11764706 0.88235294) *
##            33) PBRAND< 3.5 95  11 No (0.11578947 0.88421053) *
```

```
##           17) MBERBOER>=0.5 57    3 No (0.05263158 0.94736842) *
##         9) MOSHOOFD=4,5,6,7,8,9,10 630   63 No (0.10000000 0.90000000) *
##       5) PBRAND< 2.5 933   58 No (0.06216506 0.93783494) *
##     3) PPERSAUT< 5.5 2839   83 No (0.02923565 0.97076435) *
```

```
train_rpart$variable.importance
```

```
##    PPERSAUT      PBRAND    MBERMIDD    MOSHOOFD    MOPLLAAG      MHHUUR
## 12.80379201 10.01680300  7.18162872  6.05789790  5.95898129  5.85136716
##      ABRAND     PWAPART    MFWEKIND     AWAPART      MGODGE    MBERBOER
##  4.91498185  3.66677337  3.63504103  3.42017220  3.40820565  2.97153465
##      MRELGE       MAUT1      PLEVEN    MOPLMIDD      ALEVEN    MBERZELF
##  2.78115217  2.31053271  1.75741401  1.28335397  1.03377295  0.62976934
##      MGODOV    ABYSTAND    PBYSTAND      AFIETS
##  0.51562984  0.23362216  0.23362216  0.06723625
```

```
rpart.plot(train_rpart)
```




```
y_hat_rpart <- predict(train_rpart, validation_set, type = "class")
cm_rpart <- confusionMatrix(y_hat_rpart, validation_set$Purchase)
cm_rpart$table
```

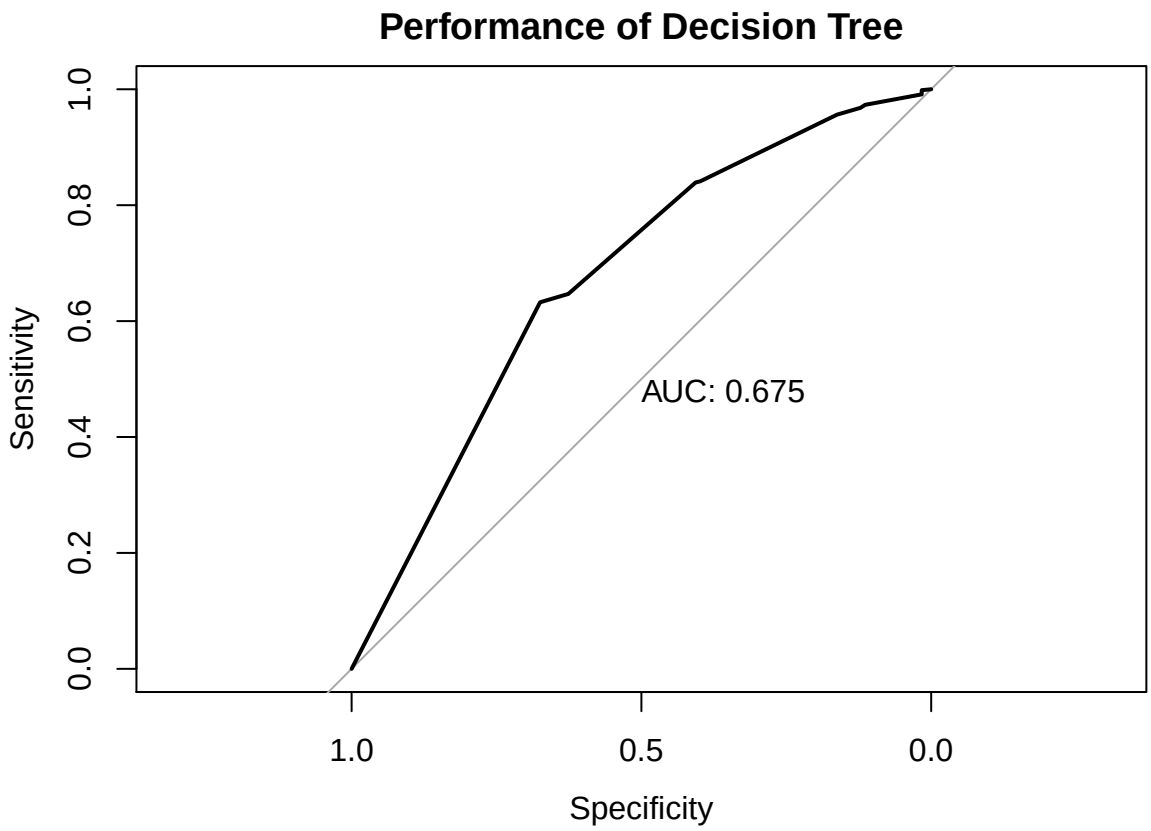
```
##            Reference
## Prediction  Yes   No
##        Yes    2    4
##        No   121 1936
```

The five most important variables that segments the customers are PPERSAUT, PBRAND, ABRAND, MOSHOOFD and PWAPART. These are contribution to car policies, contribution to and number of fire policies, customer type and contribution to third party insurance. The plot shows that the most likely customers that will get caravan insurance are:

Customer Profile: contribution to car policy >= 6, to fire policy >= 3, customer main types are Successful hedonists/ pleasure seekers or Driven Growers, up to 75% people living in the post code have no religion, less than 11% have incomes more than 123,000, most people not working in middle management, people who are not entrepreneurs and people who are married. Out of this profile, there are those that have low level education and those who do not.

The measures for the model can be seen below as well as the ROC curve.

Method	**Accuracy**	**Sensitivity**	**Specificity**	**F1_score**
Generalised Linear Model	0.9384392	0.0000000	0.9979381	NaN
K Nearest Neighbour	0.0596219	1.0000000	0.0000000	0.1125343
Decision Tree	0.9394086	0.0162602	0.9979381	0.0310078



The sensitivity and F1 score is better than logistic regression. The AUC is less and the ROC curve is not as smooth. In terms of accuracy, decision trees are rarely the best performing method since it is not very flexible and is highly unstable to changes in training data.

6.5 Random Forest

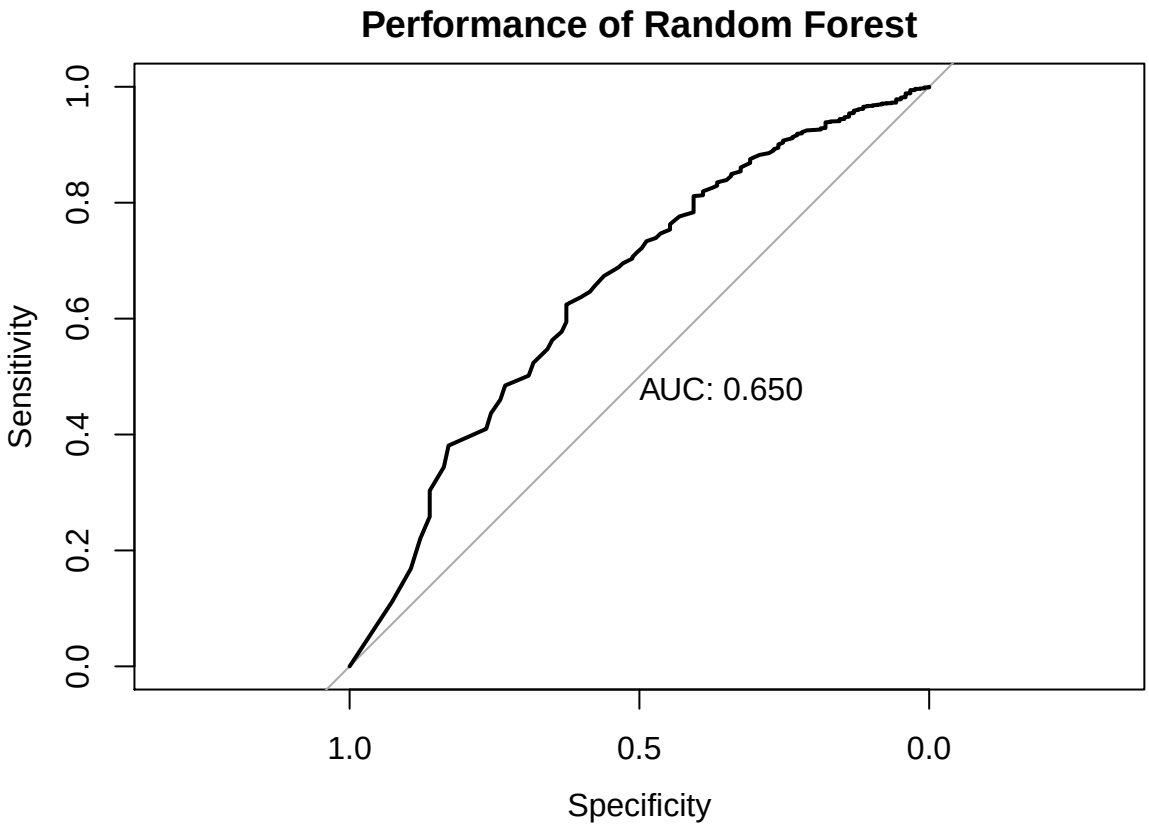
Random forests are a very popular machine learning approach that addresses the faults of decision trees. It is a supervised learning algorithm mainly used for classification problems. The goal is to improve prediction performance and reduce instability by averaging or combining the results of different decision trees.

```
training_rf <- training_set

set.seed(9) # for reproducibility

train_rf <- randomForest(Purchase ~ ., data = training_set)

cm_rf <- confusionMatrix(predict(train_rf, validation_set),
                         validation_set$Purchase)
cm_rf$table
```

```
##            Reference
## Prediction  Yes   No
##        Yes    4   11
##        No   119 1929
```

The measures for the model can be seen below as well as the ROC curve.

Method	Accuracy	Sensitivity	Specificity	F1_score
Generalised Linear Model	0.9384392	0.0000000	0.9979381	NaN
K Nearest Neighbour	0.0596219	1.0000000	0.0000000	0.1125343
Decision Tree	0.9394086	0.0162602	0.9979381	0.0310078
Random Forest	0.9369850	0.0325203	0.9943299	0.0579710



When comparing all the machine learning classifier models that have been tried till now, the Random Forest one has the highest sensitivity, F1 score and AUC.

6.6 Regularised Random Forest

There is still room for improvement in the Random Forest model. Regularisation will be used to try optimise the parameters of the algorithm. The parameter mtry is not automatically optimised by the caret package in R. So, a function is written to find the best value for mtry that will maximise the F1 score of the model. Mtry is the number of variables available for splitting at each tree node.

```
training_rrf <- training_set

set.seed(11) # for reproducibility

#pick the mtry in rf using cross validation

mtry <- seq(1, 33, 2)
rf <- map_df(mtry, function(m){

  fit_rf <- randomForest(Purchase ~ ., data = training_rrf, mtry = m)
  y_hat_rf <- predict(fit_rf, training_set, type = "class")
  cm_train <- confusionMatrix(y_hat_rf, training_set$Purchase)
  train_sens <- cm_train$byClass["Sensitivity"]
  y_hat_rf <- predict(fit_rf, validation_set, type = "class")
  cm_test <- confusionMatrix(y_hat_rf, validation_set$Purchase)
  test_acc <- cm_test$overall["Accuracy"]
  test_sens <- cm_test$byClass["Sensitivity"]
  test_spec <- cm_test$byClass["Specificity"]
  test_F1 <- cm_test$byClass["F1"]


  tibble(mtry = m, train = train_sens, test = test_sens, acc = test_acc,
         spec = test_spec, F1 = test_F1)
})
rf
```

```
## # A tibble: 17 x 6
##     mtry train   test   acc  spec       F1
##    <dbl> <dbl>  <dbl> <dbl> <dbl>    <dbl>
##  1     1 0      0      0.940 1      NA
##  2     3 0.111 0      0.940 0.999 NaN
##  3     5 0.585 0.0325 0.937 0.994   0.0580
##  4     7 0.774 0.0325 0.932 0.989   0.0537
##  5     9 0.829 0.0325 0.929 0.986   0.0516
##  6    11 0.836 0.0325 0.928 0.985   0.0510
##  7    13 0.836 0.0325 0.926 0.983   0.05
##  8    15 0.836 0.0325 0.925 0.981   0.0491
##  9    17 0.836 0.0325 0.924 0.980   0.0485
## 10    19 0.840 0.0325 0.924 0.980   0.0485
## 11    21 0.833 0.0325 0.923 0.979   0.0479
## 12    23 0.840 0.0325 0.923 0.979   0.0479
```

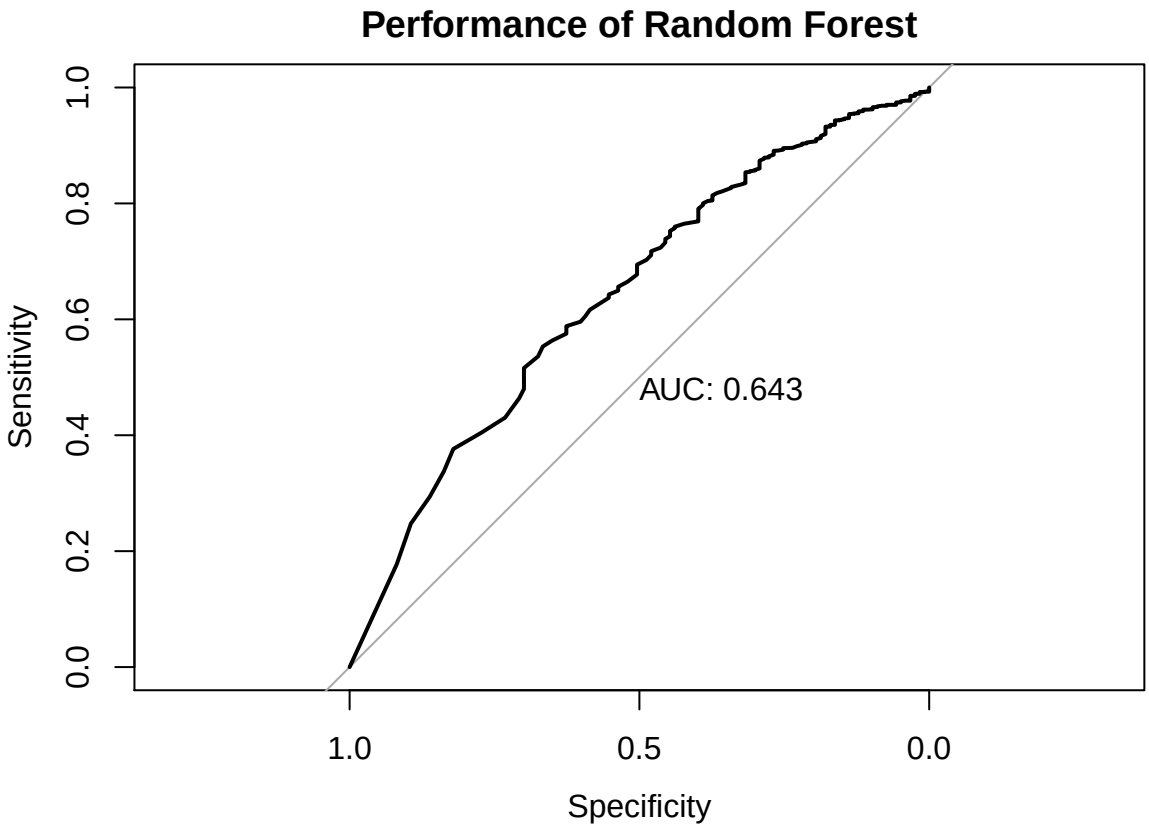
```
## 13    25 0.850 0.0407 0.924 0.980   0.0599
## 14    27 0.843 0.0325 0.922 0.979   0.0476
## 15    29 0.850 0.0325 0.923 0.980   0.0482
## 16    31 0.850 0.0325 0.923 0.979   0.0479
## 17    33 0.840 0.0325 0.921 0.978   0.0471
```

```
# Pick the mtry that maximises F1 using the estimates built on the test data
rf$mtry[which.max(rf$F1)]
```

```
## [1] 25
```

Now that the value of mtry has been found, the measures for the model can be calculated. They can be found below along with the ROC curve showing the AUC.

Method	Accuracy	Sensitivity	Specificity	F1_score
Generalised Linear Model	0.9384392	0.0000000	0.9979381	NaN
K Nearest Neighbour	0.0596219	1.0000000	0.0000000	0.1125343
Decision Tree	0.9394086	0.0162602	0.9979381	0.0310078
Random Forest	0.9369850	0.0325203	0.9943299	0.0579710
Random Forest with tuning mtry	0.9238972	0.0406504	0.9798969	0.0598802



Random forests work well for a large range of data observations and high dimensions than a single decision tree does. The scaling of data is not needed for the random forest algorithm. It maintains a good accuracy without scaling.

7 Results

The measures for all the models are provided again below. All the models were built on the data set that had the chosen predictors after preprocessing. When looking at the measures for each model tried along with each of their ROC curves, it can be noted that the Regularised Random Forest classifer model produces the best combination. This was a Random Forest model with an optimal value of mtry that maximised the F1 score.

Method	**Accuracy**	**Sensitivity**	**Specificity**	**F1_score**
Generalised Linear Model	0.9384392	0.0000000	0.9979381	NaN
K Nearest Neighbour	0.0596219	1.0000000	0.0000000	0.1125343
Decision Tree	0.9394086	0.0162602	0.9979381	0.0310078
Random Forest	0.9369850	0.0325203	0.9943299	0.0579710
Random Forest with tuning mtry	0.9238972	0.0406504	0.9798969	0.0598802

7.1 Final Model

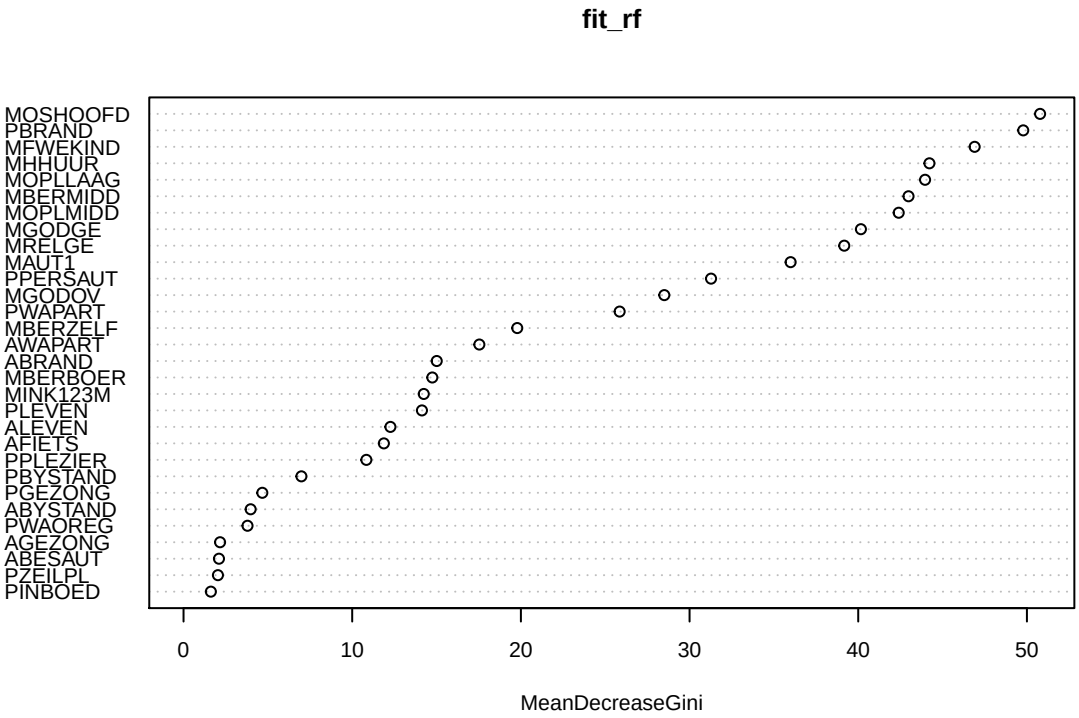
The final model that has been chosen is the Regularised Random Forest. Now, the test set can be used to check how the model performs against unseen data. First, the test set will be subsetted so that it only has the predictors that were chosen earlier after preprocessing.

```
# Regularised random forest

test_set <- test_set[, c("MGODOV", "MGODGE", "MOPLMIDD", "MOPLLAAG",
                         "MRELGE","MBERZELF", "MBERBOER", "MBERMIDD",
                         "MHHUUR", "MFWEKIND", "MAUT1", "MINK123M",
                         "PWAPART", "PPERSAUT", "PLEVEN", "PPERSONG",
                         "PGEZONG", "PWAOREG", "PBRAND", "PZEILPL",
                         "PPLEZIER", "PINBOED", "PBYSTAND", "AWAPART",
                         "AWALAND", "ABESAUT", "AWERKT", "ALEVEN",
                         "APERSONG", "AGEZONG", "ABRAND", "AFIETS",
                         "AINBOED", "ABYSTAND", "MOSHOOFD", "Purchase")]


set.seed(15) # for reproducibility

fit_rf <- randomForest(Purchase ~ .,
                       data = train_set2,
                       mtry = rf$mtry[which.max(rf$F1)])

y_hat_rf <- predict(fit_rf, test_set, type = "class")
cm_test <- confusionMatrix(y_hat_rf, test_set$Purchase)
cm_test$table
```

```
##            Reference
## Prediction  Yes   No
##        Yes   18   43
##        No   158 2728
```

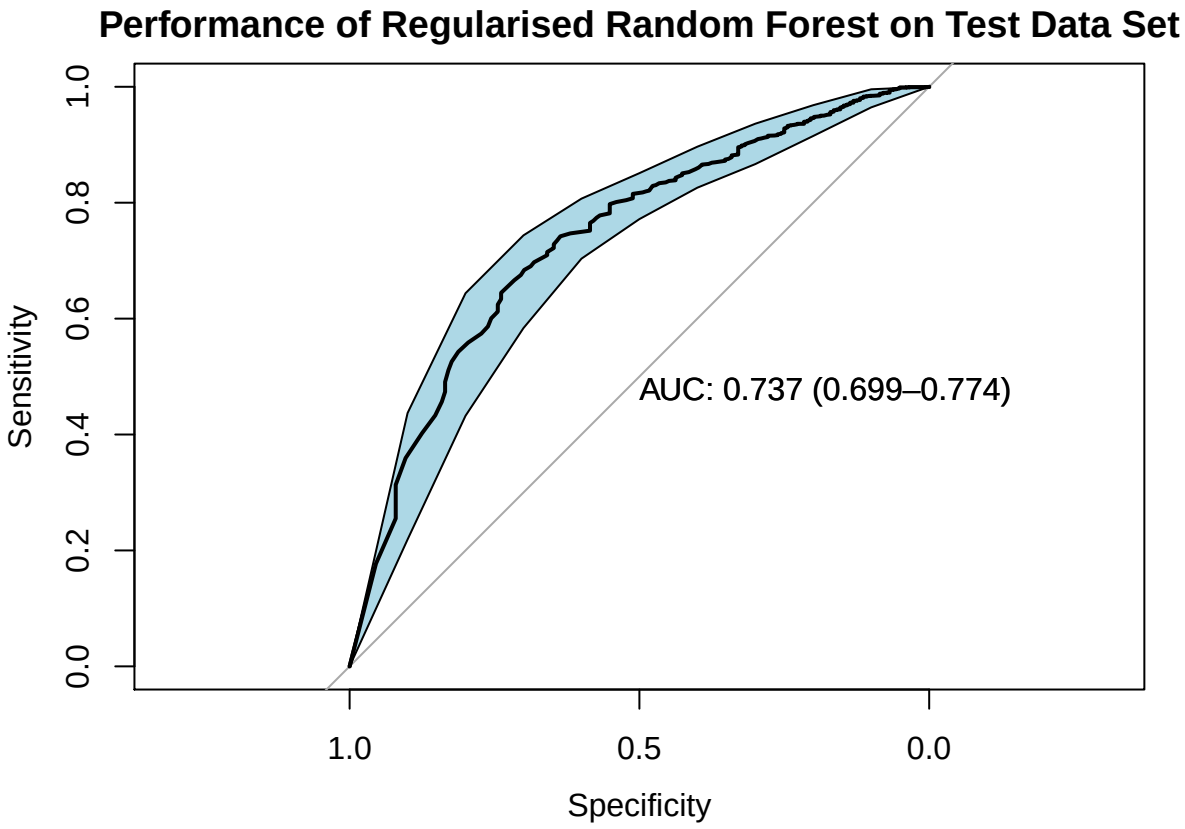
The confusion matrix does show that the model is able to identify a few of those customers who would get caravan insurance. However, it also classifies a good number of them as not interested in getting an insurance policy.

The following is a plot of the importance of the variables for the final model. It shows 30 features. As well as the features that were derived from the individual decision tree, there are other points that can be noted. The customers that have caravan insurance also have other insurance policies such as social security and property insurance cover. There is another segment which includes farmers. There are also people who have cover for outdoor things like bicycles, surfboards and cars.



The measures for the model can be calculated. They can be found below along with the ROC curve showing the AUC with a 95% confidence interval.

Method	**Accuracy**	**Sensitivity**	**Specificity**	**F1_score**
Random Forest with tuning mtry on test set	0.931795	0.1022727	0.9844821	0.1518987



As the final model is built on a significantly larger data set (train data set) compared to the smaller training data set, we do not get the same measures when the model is trained and applied to the smaller validation data set. It can be seen that the model on the test data set produces a higher sensitivity and F1 score. The AUC though is slightly smaller.

8 Conclusion

First, in order to build any prediction models, we first have to explore, understand and analyse the data. It was done by simple exploration techniques, creating proportions and generating visualisation plots. This was all done on the train data set so that the test data set was kept as unseen data for the testing of the final model.

Since the Caravan Insurance data set had high dimensionality, it was imperative that the number of variables should be reduced. A number of methods were employed to come up with an optimal subset of variables. These methods were Recursive Feature Elimination, Stepwise Selection, checking for variables that have near zero variance and looking at correlations between each of the variables as well as with the target variable, Purchase. We were able to eliminate 50 variables in total.

A number of classifier models were built and was tested using a smaller validation data set that was randomly partitioned from the train data set. These models were Logistic Regression, K Nearest Neigbour, Decision Trees, Random Forest and Regularised Random FOrest. The measures that were used to check the efficacy of each of the models were the sensitivity, F1 measure and the Area under the curve of the ROC curve. The approach taken was step-by-step, iterative and employing various analysis and model building techniques.

The Regularised Random Forest model was chosen as the best one based on the measures. The performance of the final model on the test data set resulted in quite a good sensitivity and F1 score. From the confusion matrix, it can be seen that it does correctly identify some customers who would take an insurance policy for their caravans. However, it is still not the most accurate prediction model. A good model should produce an AUC score of 0.70 or more. Considering that the final model only produced an AUC score of 0.681, it shows the limitation of its prediction power.

The final model is quite effective in identifying customer segments that the insurance company can focus on when they market the product of a mobile home insurance policy. It was noted that those who are pleasure seeking, love outdoor activities, educated, married, have families, are not that religious, have rented homes and have sufficient income are more likely to buy a policy. They also have cover for other things which shows that they do know the importance of having insurance cover. In addition, there is another segment where people have low level education, likely farmers, and lower income levels that might actually need a caravan as a necessity and so will look out to get the insurance policy.

8.1 Possible Improvements for the Final Model

There is definitely a scope for improvements of the final model. Since the data set was extremely imbalanced, it did provide limitations during the training of the models. Techniques such as up-sampling or downsampling the data set could make it much easier. It could also be optimised over time when there is more data that can be collected in the future when other customers purchase caravan insurance.

More variable transformations such as the possibility of interaction terms could be considered in the future. There were variables where there is contribution and number of insurance policies. So, it is worthwhile to check if an interaction term between them could be created. More statistical tests could be conducted when deciding what variables to be eliminated.

The final model does have other tuning parameters other than mtry. Functions could be written to see if these parameters can be refined or optimised further. Apart from this model, there are many more classifier models that could be built and tested to check if it produces a better AUC score than the Regularised Random Forest. The other models include Artificial Neural Networks, C5.0 Algorithm, Support Vector Machine and Boosted Ensembles.

At the end though, there is an aspect of free choice if someone wants to get insurance cover or not during a marketing campaign such as telephone calls or fliers. This is not easy to incorporate when building a prediction model so it will be difficult to get a perfectly accurate model.