DETAILS OF ASSIGNMENT			
STUDENT NAME	Aseem Malhotra	SWINBURNE ID NUMBER	10155740 0
EMAIL ADDRESS	101557400@student.swin.edu.au	PHONE CONTACT	04305483 20
UNIT CODE & NAME	STA 30004 – Data Mining		
ASSESSMENT TITLE	Data Mining Report		
TUTOR'S NAME:	Dr Prahan Apputhurai	DATE OF SUBMISSION:	14th October 2018

.

DECLARATION

I declare that (the first four boxes must be completed for the assignment to be accepted):

☒ This assignment does not contain any material that has previously been submitted for assessment at this or any other university.

☒ This is an original piece of work and no part has been completed by any other student than signed below.

☒ I have read and understood the avoiding plagiarism guidelines at http://www.swinburne.edu.au/ltas/plagiarism/students.htm and no part of this work has been copied or paraphrased **from any other source** except where this has been clearly acknowledged in the body of the assignment and included in the reference list.

☒ I have retained a copy of this assignment in the event of it becoming lost or damaged.

☒ (optional) I agree to a copy of the assignment being retained as an exemplar for future students (subject to identifying details being removed).

Student acknowledgement (by signing or typing your name you agree to the above):	Aseem Malhotra	**Date:**	5th October 2018

DETAILS OF FEEDBACK

Office Use Only			
Date Received		Received by	
Total Mark / Grade		Marker	

Table of Content

1.0 Introduction.

The aim of data mining is to consider all the data available, separate the useful information and use that information to find ways to predict future outcome. We are moving in an age where we have unlimited amount of data but the skill of finding relevant pattern in it is still a challenge.

In this report, we use the skills we have learned to explore the two datasets. We will start by using simple descriptive statistics, use two and three dimensional graphs to explain the relationship between variables. Then we use simple decision trees and association analysis to male small prediction and use confusion matrix to give score to outcomes. Then we will use advanced support vector machines and neural networks to explore our data and look for patters and useful information to create prediction models.

The two datasets we explore for this report are motorAM.csv and MBAmotor2.csv.

motorAM is the data collected from an insurance company. It has record of 10,000 policies and for each policy there are 9 other variables such as age of car, number of drivers and how much a car is used each year and whether it's used for personal of commercial purpose. The data also tells us the excess of each policy or whether the policy holder made a claim or not. Our aim is to use this data on behalf of the insurance company and find ways to reduce their risk probability or to predict is as accurately as possible.

The MBAmotor2 contains the type of claim made by the policy holders. The claims are divided into four categories and a single patron can have more than one type of claim on each policy. The four categories are AD, FT. PD and PI. We can consider which claims are most and least likely and does one claim increases or decreases likelihood of other claims.

We will explore these two datasets and compare our findings in section 9 of this report and a conclusion in the last section.

2.0 Descriptive statistics.

We always start data mining by exploring the data. The first step is to understand the data, what variables are given and how can they be used. We start looking at a sample of 7,000 policies. The percentage of patrons who have filed for a claim in the last year (50.1%, 3504) is almost equal to those who have not filed for a claim. (49.9%, 3496). The cars usage is divided into 4 categories Figure 1.1 on the right, SC which accounts for more than half the policy holders with 3633 closely followed by S 3041. There are 323 patrons with categories SB and only 3 with ST. A detail of descriptive is given in the *Table 1.1* below.

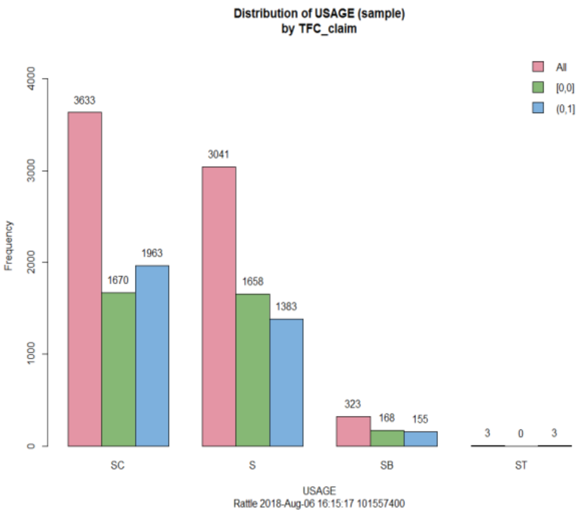


Figure 1.1: Distribution of Usage by Claim

Table 1.1: Descriptive statistics of variables

	MILAGE	EXCESS	DRIVERS	CAR AGE	EXPOUSURE	TOTAL	PRIMAGE
MIN	1,000	0	1	0	0	0	18
1ST QURT	5,000	75	1	4	0.5175	0	45
MEDIAN	8,000	75	2	7	0.9966	11.62	54
MEAN	7,810	85.31	1.75	6.977	0.7526	649.70	54.94
3RDQURT	10,000	100	2	10	0.9966	417.21	66
MAX	40,000	100	7	23	1.0021	151607.52	93

The above table gives an idea of the data at hand. We can see that the Milage, car age and number of drivers are all positively skewed as their mean is less then median and max is quite large. The excess amount even though is recorded as a numeric variable comprises of only 3 values, $0, $75 and $100. It would be more useful to treat it as a categorical variable than numeric variable. When looking at the total amount even though it is numerical it should be separated into 2 categories, because those who have not filled for a claim which accounts for about half of policy holders, have zero total and for the rest its anywhere between $0 to $151607.52. That's why a log transformation on total will be misleading as it will include those who have not files for claim and has zero total with those who have filed a claim. The median car age is about 7 years and there are about 2 drivers per car on average.

When looking at the correlation graph (*Figure 1.2*) we do not find any strong correlation (positive or negative) in any of the variables. There is a weak positive correlation between primage-car age and primage-exposure and between drivers-mileage. We found weak negative correlation between primage-excess, primage-drivers, primage-mileage, car age-mileage and exposure-drivers.

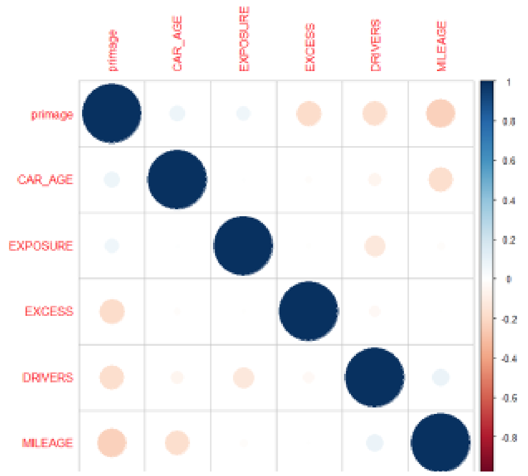
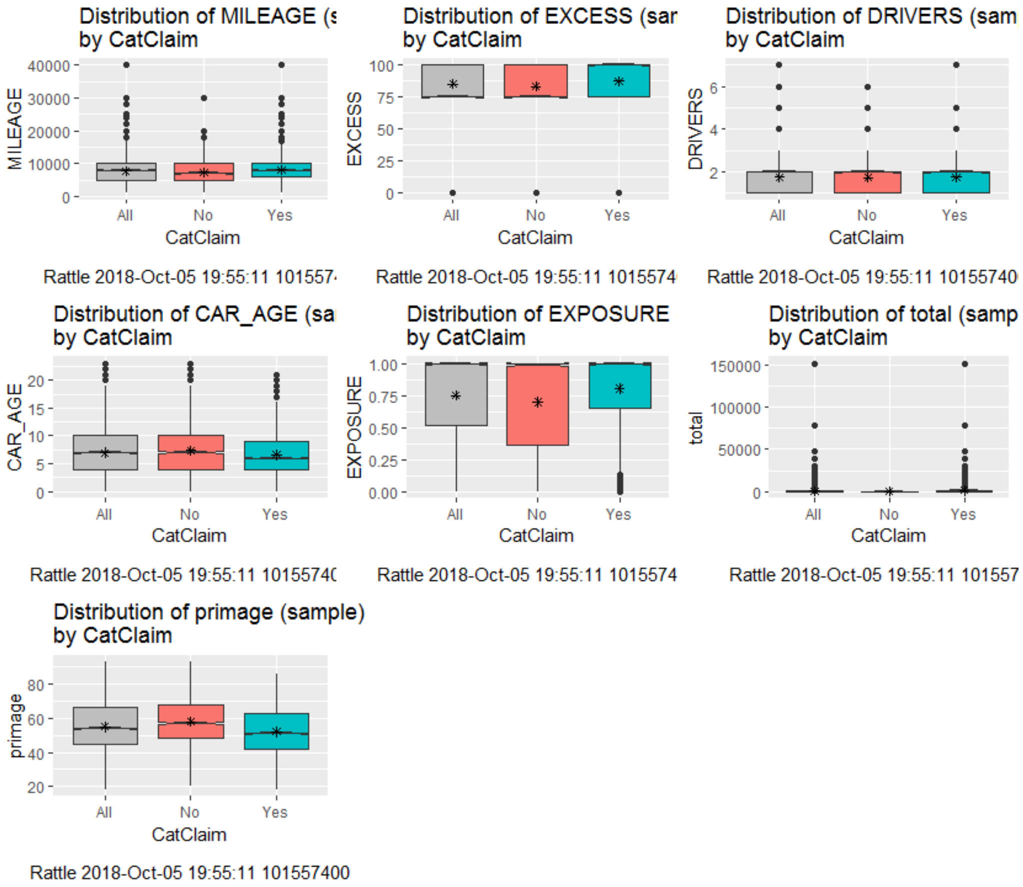


Figure 1.2: Correlation chart

The *Figure 1.3* below tells us the distribution of data based on whether the patron has made a claim or not. The mileage for patrons who made a claim is slightly higher than those who did not make any claim. Excess amount is decided upon taking the policy so it has no relationship with the claim. The number of drivers seems to have little impact too on whether a claim is made or not. Patrons with a newer car seems to be more likely to make a claim than people with older cars and the distribution of exposure for those who do not make a claim seems to have higher variance than those who do make a claim. Obliviously those who did not make a claim has zero claim account and the primage for those who made a claim is slightly lower than those who did not.



Distribution of MILEAGE (s
by CatClaim

Rattle 2018-Oct-05 19:55:11 101557

Distribution of EXCESS (sar
by CatClaim

Rattle 2018-Oct-05 19:55:11 1015574

Distribution of DRIVERS (sam
by CatClaim

Rattle 2018-Oct-05 19:55:11 10155740

Distribution of CAR_AGE (sa
by CatClaim

Rattle 2018-Oct-05 19:55:11 1015574(

Distribution of EXPOSURE
by CatClaim

Rattle 2018-Oct-05 19:55:11 1015574

Distribution of total (samp
by CatClaim

Rattle 2018-Oct-05 19:55:11 101557

Distribution of primage (sample)
by CatClaim

Rattle 2018-Oct-05 19:55:11 101557400

Figure 1.3: Boxplot of each variable by claim

3.0 Exploratory Analysis

Another way to explore data is through graphical data analysis. Often the important characteristic we miss numerically can be explored on a chart or boxplot. It’s also the best way to explain our analysis to people with no statistical background.

We start with a summary statistics of the 70% training data. Out of total 7,000 policies 3,496 did not received any claim and 3.504 had a claim. When we look at the box-plot for excess on left and compare the chart produced by GGobi display (purple for claim Yes) we notice only a small difference in mean excess for those with claim Yes and No.

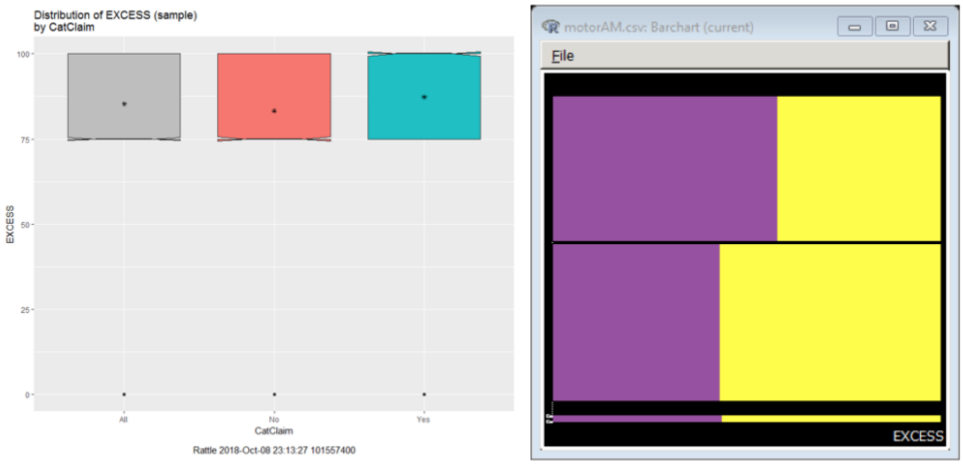


Figure 2.1: Distribution of Excess

Figure 2.2 on the right shows scatter plots between different variables with Yes claim in purple and No claim in yellow. The diagonals are scatter plots of individual variables and we can see that excess has just 3 values and cat claim only 2. Car age has a slightly positively skewed chart and primage has normal distribution. Exposure has a low value for most part and then it just jumps towards the end.

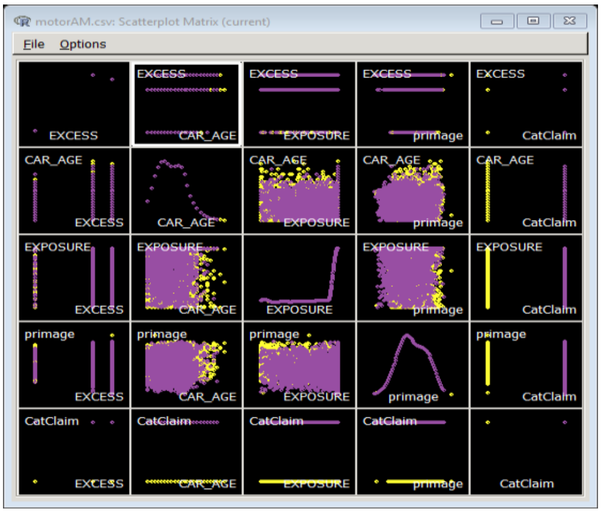


Figure 2.2: scatter plots

In the *Figure 2.3* below we look for relationship between the drivers who have made a claim and their age. We can see that the purple bars (claim Yes) are spread across evenly on the primage side which says that people of all ages applied for a claim and not just young ones.

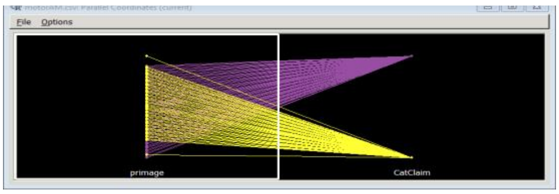


Figure2.3: distribution of claim by primage

We would use features like brush to investigate the correlation between age of drivers and age of car. We were looking for a negative relationship (young drivers with old cars) which we can tell from the scatter matrix is not there. In fact, we see *Figure 2.4* a weak positive correlation.

The figure below is a 3D rotating plot which gives us a unique look in the distribution of data based on excess, car age and exposure and where the older cars brushed in red lie on that plot.

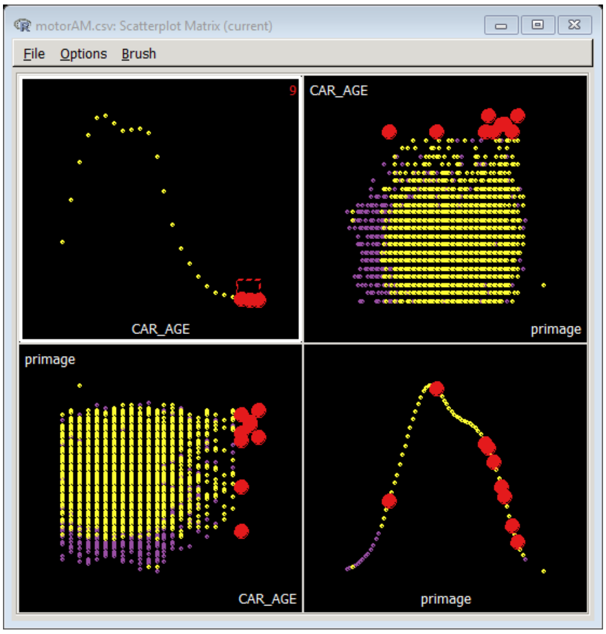


Figure2.4: Distribution of age of car and drivers

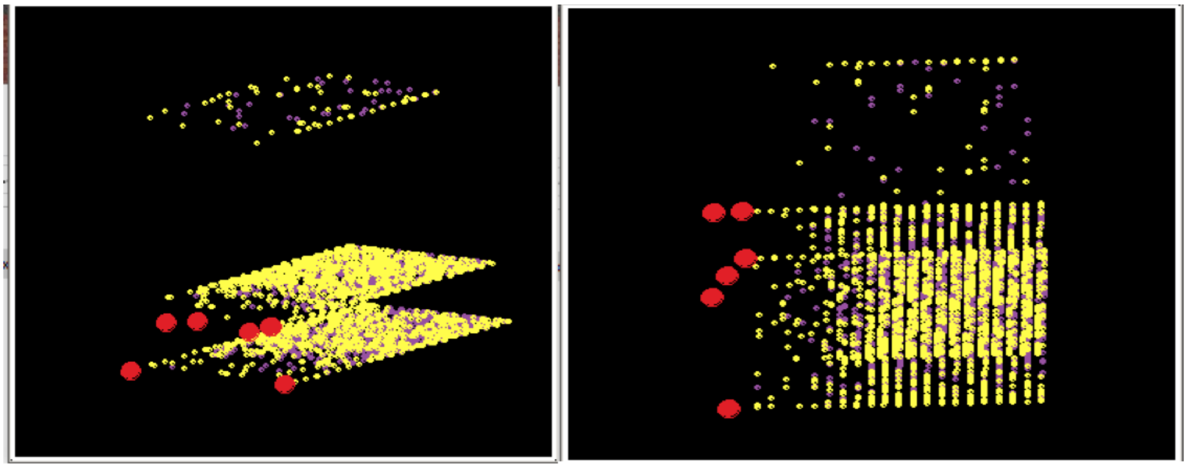


Figure 2.5: 3D rotating image

The *Figure 2.6* on the right shows a 1D tour of the binomial distribution of primage separated by catclaim. This shows we can graphically interpret the data in different ways that tells us more than by just looking at the numbers.

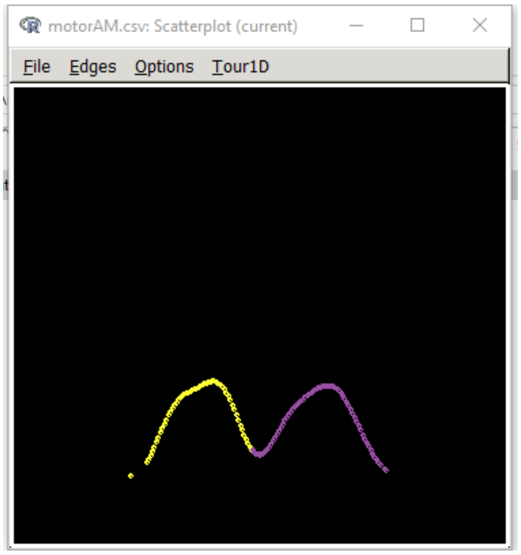


Figure 2.6: 1 D tour of age of drivers

4.0 Association Analysis

In this part of the report we explore MBAmotor2 dataset and look at how one claim impacts the other. We examine the support, confidence and lift values of one score on the other. We start by examining how many different claims there are and their frequency. *Figure 4.1* shows us that roughly half (50%) of the claims are for accidental damages (AD), followed by windshield claims (WS) around 38%. Personal damages (PD) accounts for about 25% of total claims and thefts (FT) about 10%. The least number of claims were personal injury (PI) at 5%. A patron can claim for more than one type of claim at a time hence the percentage of claims add up to more than 100.

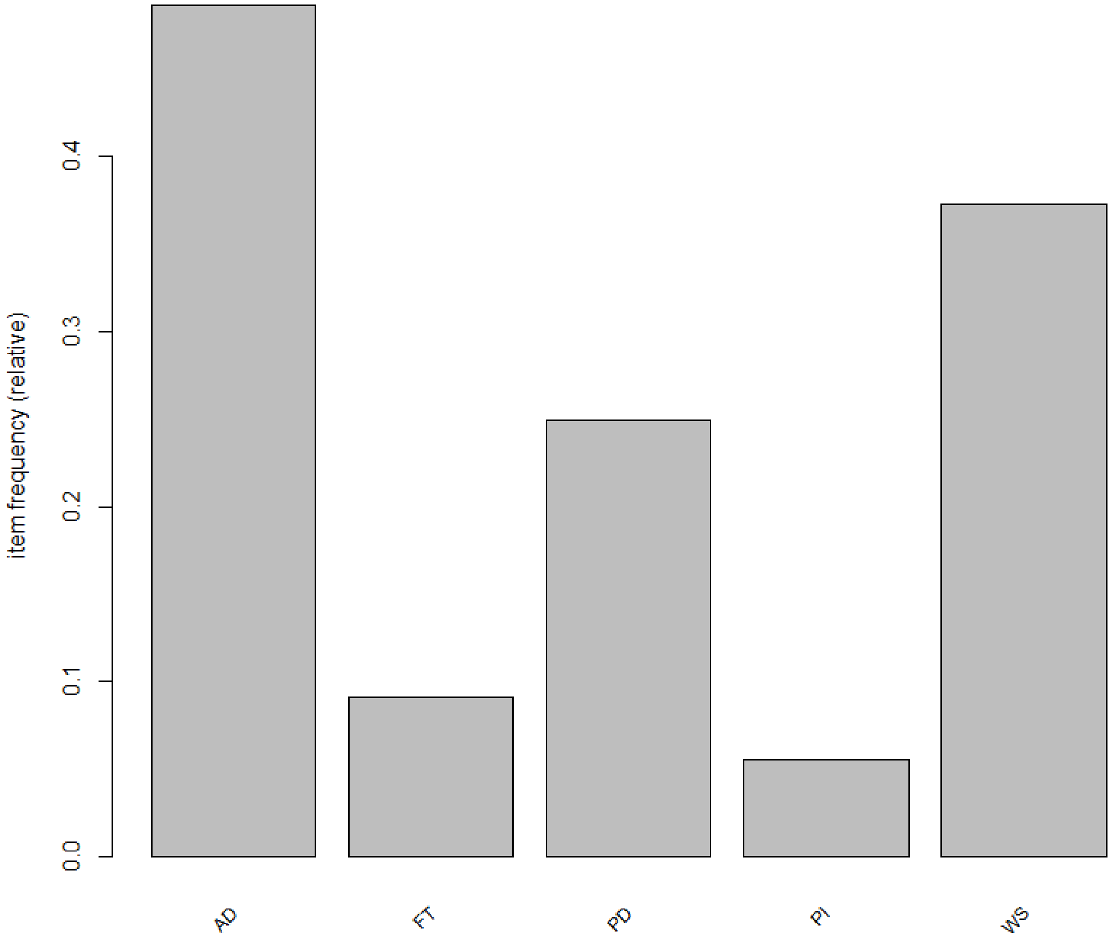


Figure 4.1: Frequency of claims

Table 4.1 below gives a summary of all the support, confidence and lift values for this dataset. We need to identify the high lift values and use them t0 find relationship between different values.

Table 4.1: Summary of support, confidence and lift

	Support	**Confidence**	**Lift**	**Count**
MIN	0.01476	0.01621	1	217
1stQURT	0.04027	0.2473	1.3427	592
MEDIAN	0.04285	0.6528	1.6747	630
MEAN	0.06700	0.5576	2.4219	985
3RDQURT	.04945	0.8143	3.5849	727
MAX	0.16285	0.9397	4.4715	2394

The most profound relationship we found was between accidental damages (AD), personal damages (PD) and personal injury (PI). When accidental damages (AD) and personal damages (PD) are given its 4.47 times more likely that the patron will seek to file a claim for personal injury (PI). And when personal injury (PI) and accidental damages (AD) are given the likelihood that a patron will claim for personal damages (PD) increases by 3.77 times.

Personal damages (PD) and personal injury (PI) have a lift of 3.58 times between them but the confidence of Personal damages (PD) when personal injury (PI) is given is a lot higher than other way around. This difference needs to be considered when looking for relationships and likelihoods.

Claims which are list likely to appear together are windshield claims (WS) and accidental damages (AD) with a lift of just 0.1 and windshield claims (WS) and thefts claims (FT) with a lift of 0.43.

5.0 Classification Tree

Classification trees are and easy way to represent the decision process and the importance of variables in those decisions. The data starts at the top (root) and then divided at each stage by categorical and ordinal variables till all variables are accounted for at the bottom (leaves). We will also see how a loss matrix and prior weightage affects our decision tree.

The *Figure 5.1* shows us a basic decision tree without any weightage. The result claim Yes or No is based on its higher probability in dataset. E.g. for Nod 1 the percentage for Yes is 50.05% and no is 49.95% so decision is Yes. In Nod 6 the No percentage (50.47%) is higher than Yes (49.53%) so decision is no. The most important variable for this decision tree is exposure (above and below 0.16), followed by primage and then excess.

The ideal number of splits as per both relative error (0.7663) and cross-validation error (0.7869) are 4.

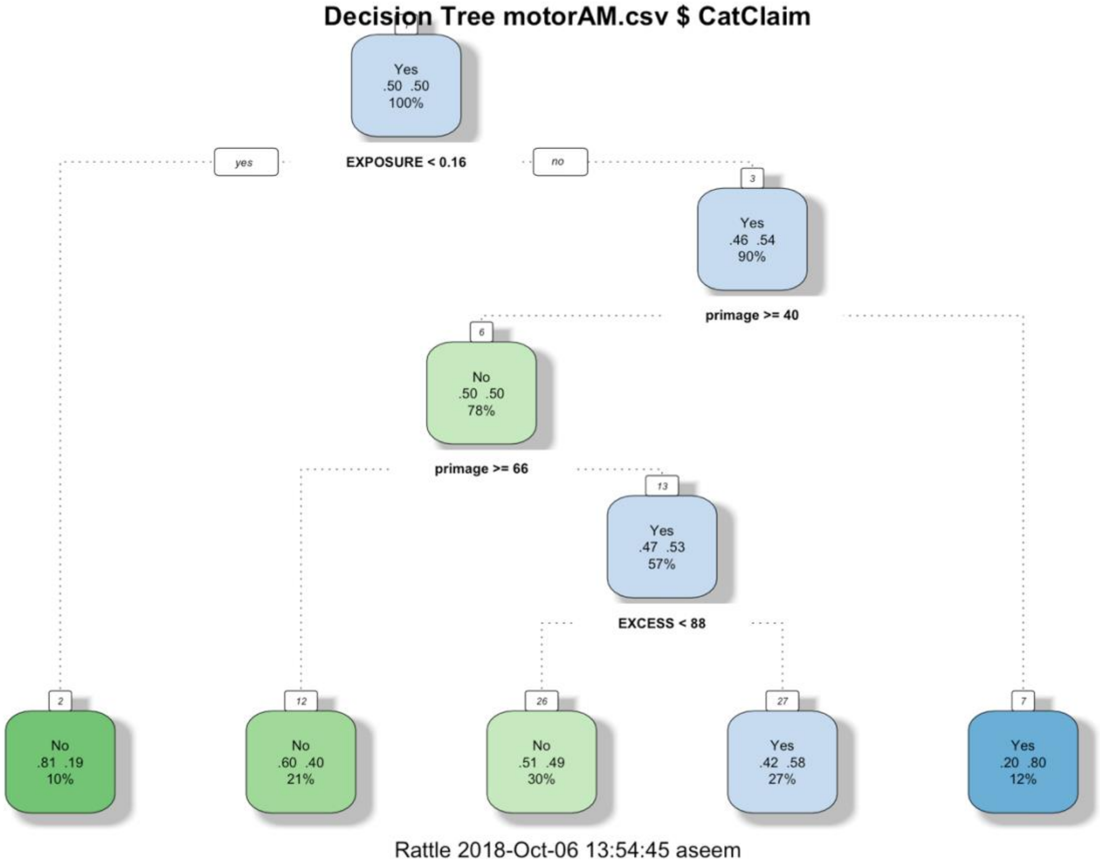


Figure 5.1: Decision tree

Loss Matrix

When we apply a loss matrix where false negative (No claim) is twice the false positive (yes) (i.e. loss matrix of 0, 2, 1, 0) the result is in the *Figure 5.2* below. The initial Nod is No as loss for false negative 99.9 (49.95*2) is bigger loss than loss for false positive (50.05). As per both relative error (0.91182) and cross-validation error (1.7705), one split is required i.e. primage at 35.5. 93% of claims (6505 claims) above 35.5 are in Nod 2 and it's better to assume them as No as loss for false negative is 1.06 (2*0.53) and loss for false positive is 0.47 (0.47*1)

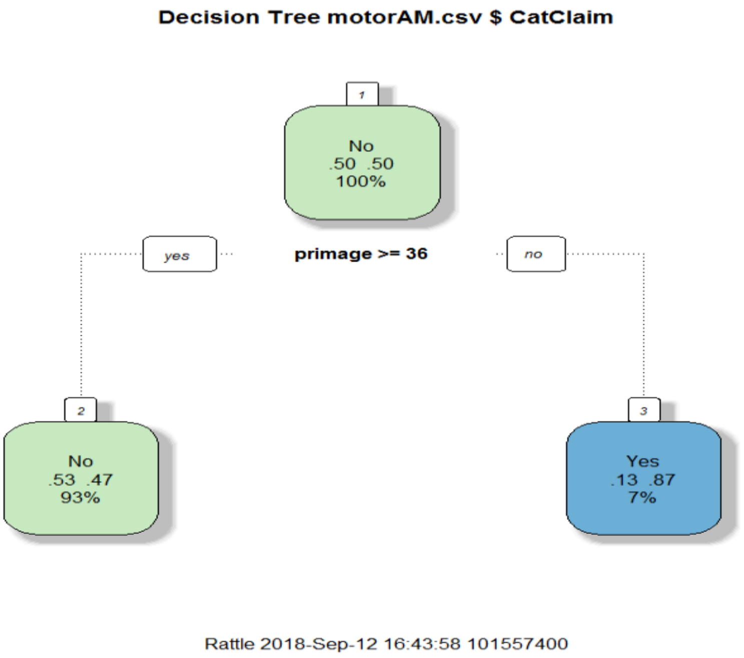

Decision Tree motorAM.csv $ CatClaim
1
No
.50 .50
100%
yes
primage >= 36
no
2
No
.53 .47
93%
3
Yes
.13 .87
7%
Rattle 2018-Sep-12 16:43:58 101557400

Figure 5.2: Decision tree with loss matrix (0, 2, 1, 0)

Prior weightage

We can give initial weightage to the categories of the target variable to reduce risk. In this case, we gave No a weight of 80% and Yes 20%. We get the same number of splits as the previous case but the proportion is different. In Nod 2 decision tree predict No with .82 to .18 and in Nod 3 it predicts Yes with .64 to .36.

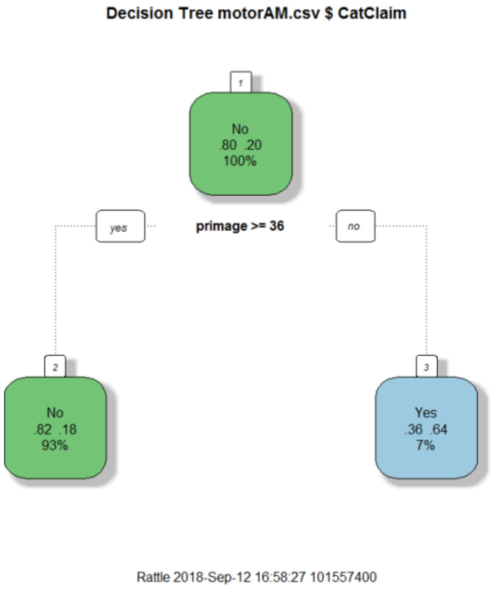

Decision Tree motorAM.csv $ CatClaim
1
No
.80 .20
100%
yes
primage >= 36
no
2
No
.82 .18
93%
3
Yes
.36 .64
7%
Rattle 2018-Sep-12 16:58:27 101557400

Figure 5.3: Weighted decision tree (0.8, 0.2)

6.0 Classification with random forest and boosting

Random forest was created to curtail the simplicity of decision trees. They are built by taking the average of hundreds of decision trees. They are very robust and extremely useful when there are many variables but few observations.

We run a random forest with 500 trees and 4 variables to predict whether a patron will make a claim or not. We get an out of bag error of 39.71%. out of bag error is the error recorded on each individual tee. The error rate confusion matrix gave us is 39.16% for Yes and 40.27% for No. We get an area under the curve of 0.65, which is below the minimum required of 0.7. *Figure 6.1* shows us that the optimum number of trees are around 150.

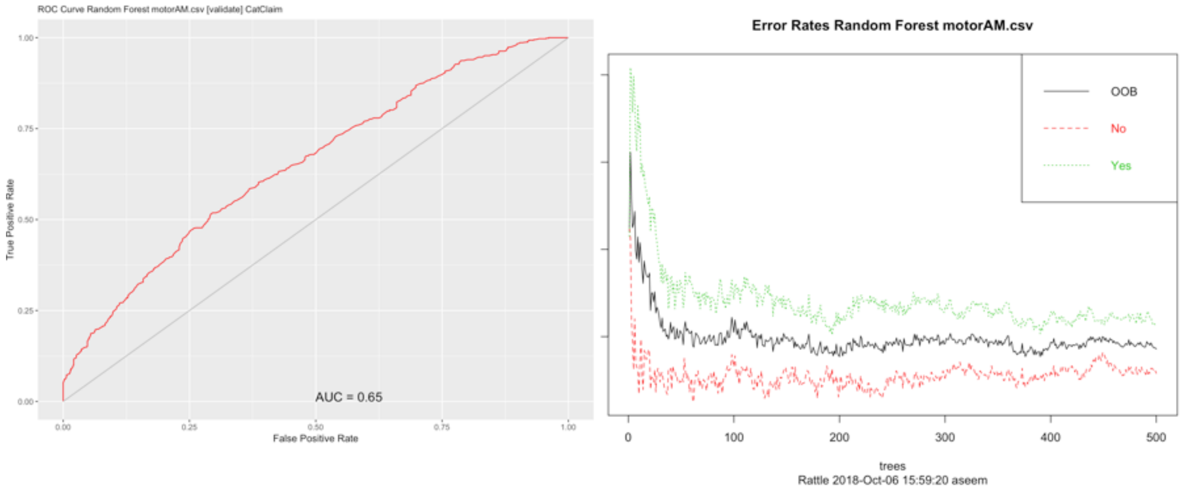


Figure 6.1: Area under the curve (left) and Optimum number of trees (right)

We can also rank variables on their importance. For the Gini measure the top-ranking variables are primage, exposure, car age, mileage and drivers. The variables important for accuracy measure are exposure, primage, excess, car age and mileage.

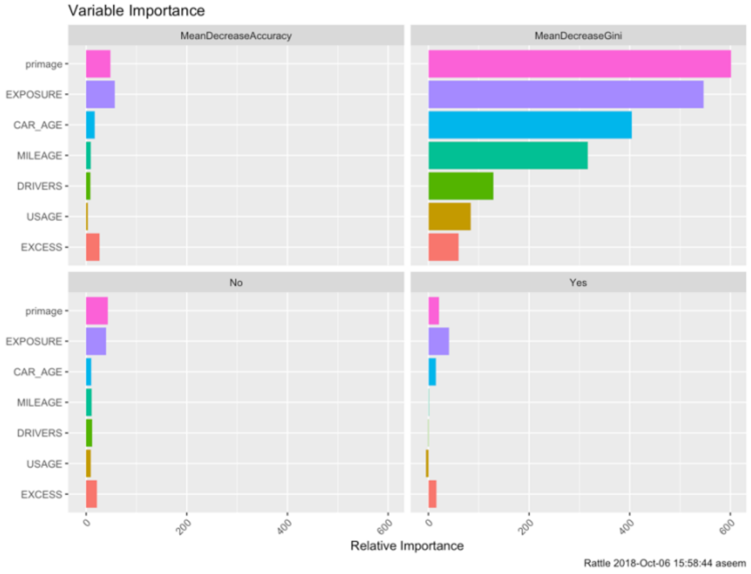


Figure 6.2: Variable importance chart

Boosting is a combination of several decision trees to create one models. Weak learners are boosted in weight to help create better prediction models. They perform better when we have many variables.

When we run an adaptive boosting on AMmotor we get an out of bag error of 35.6%. The error rate for claim Yes is 36.87% and for No is 34.10%. The area under the curve is 0.69 and the variable ranked per importance are usage, drivers and primage.

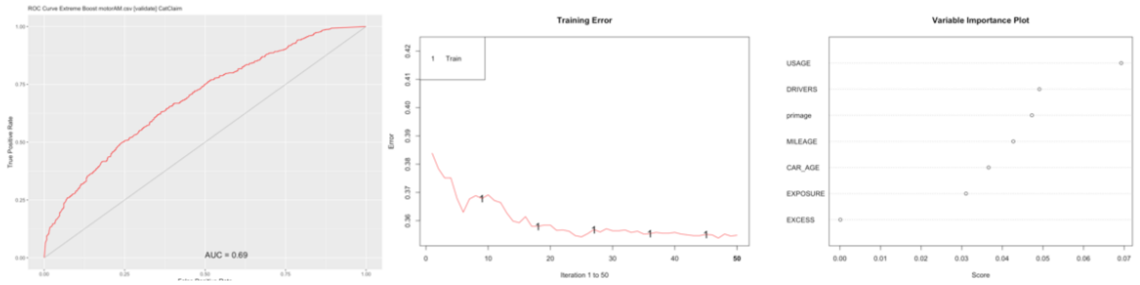


Figure 6.3: Area under the curve (left), Training error (centre), variable importance (right)

Below *Figure 6.4 and Figure 6.5* are the first five trees of adaptive boosting. It gives weights to weak learners and uses them in its predictive models.

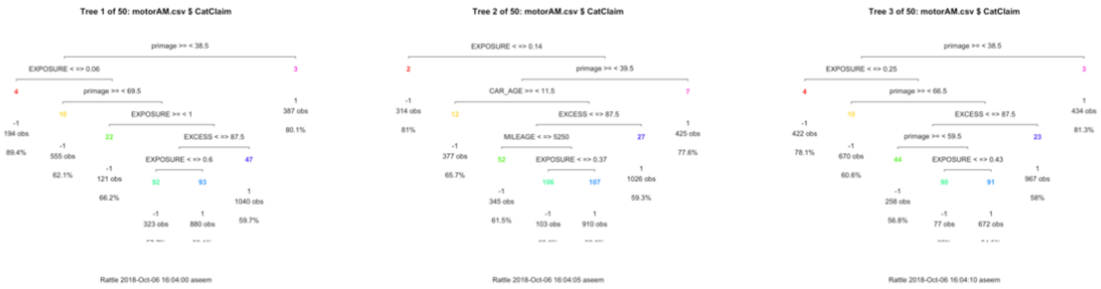


Figure 6.4: Tree 1 to 3

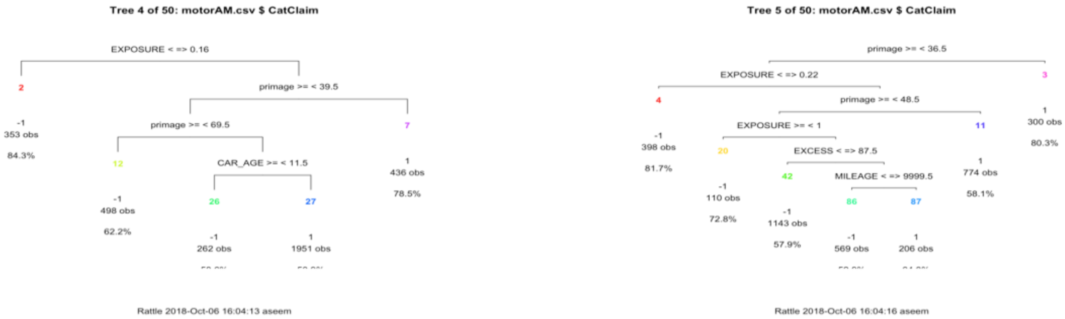


Figure 6.5: Tree 4 & 5

7.0 Classification with regression and support vector machines

Linear regression Model

We run a linear regression model with all the variables and realise the model is over fitted. We have used lots of non-significant variables in our model and our AIC is 9024.8. we try and refit the model by taking out Usage variable and the AIC gets worse by increasing to 9041.1. the odds for the new model are in *Table 7.1*.

Table 7.1: Table of odds of linear regression model.

Variable	Coefficient (B)	Exp(B)	Interpret odds ratios
Mileage	0.000040358	1.000040359	Increase by a very small amount of about 0.004% in the odds of the events when Mileage is increased by 1, when other predictors are controlled.
Excess	0.011002816	1.01106357	Increase by 1.11% in the odds of the events when Excess is increased by 1, when other predictors are controlled.
Drivers	-0.020694188	0.9795185	2.05% reduction in the odds of the events when Drivers is increased by 1, when other predictors are controlled.
Car Age	-0.036810329	0.9638589	3.62% reduction in the odds of the events when Car age is increased by 1, when other predictors are controlled.
Exposure	1.124943976	3.080044	208% increase in the odds of the events when Exposure is increased by 1, when other predictors are controlled.
Primage	-0.031559180	0.9689336	3.11% reduction in the odds of the events when primage is increased by 1, when other predictors are controlled.

Apart from the above results linear model has an overall error rate of 40%. The area under the curve by this model in 0.6485, well below the 0.7 considered to be ideal and it coves No risk of about 76% and 51% of total risk.

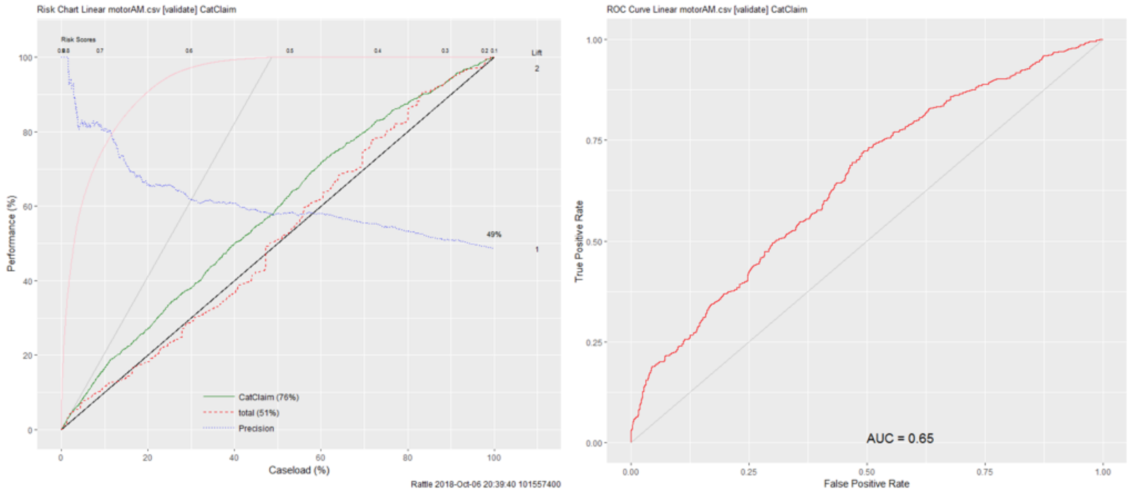


Figure 7.1: Risk chart (left) and area under the curve (right)

Support Vector Machines

When we use the same variables as we use in the regression model we get a training error rate of 35.05% with 5424 support vectors used. The overall error of SVM is 38.6% and the rick chart covers 77% of No risk and 51% of total risk. The area under the curve is 0.66.

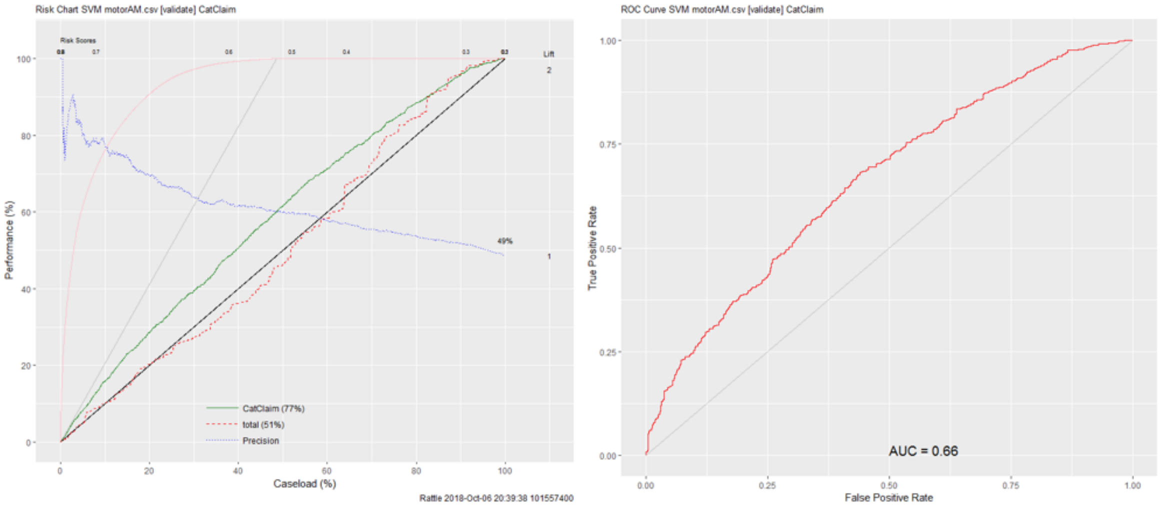


Figure 7.2: Risk chart (left) and area under the curve (right)

Table 7.2 below compared the results of both models.

Table 7.2. Comparison of liner regression with support vector machine.

	Linear Regression		**Support Vector Machine**	
Overall Error Rate	40%		38.6%	
Risk chart - No claim	76%		77%	
Risk chart – Total	51%		51%	
Area Under ROC Curves	0.6485		0.6623	
	Predict No claim	Predict Yes claim	Predict No claim	Predict Yes claim
Observe no claim	440	331	457	314
Observe at least one claim	268	461	265	464

The support vector machines produce slightly better result than linear regression. The overall error rate of SVM is lower and it covers more risk and area under the curve. If we compare the error matrix the false negative (FN) of SVN is 36.35% compared to 36.76% linear model and the false positive risk (FP) of SVN 40.73% is also better than linear model 42.93%.

So, support vector machines seem a better model than linear for this group of data.

8.0 Classification with neural network and Self-organising maps

Neural Network

To run a neural network model, we need to start by selecting all the numeric variable and scaling them to have a mean of zero and standard deviation of 1. We do this so that each variable has an equal effect on neural network and it’s not dominated by the variables measured on large scale.

We run our first neural network with two hidden nods, a 6-2-1 network with 23 weights. We get residual sum of squares of 1555.63 and an area under the curve of 0.66.

We than refit the model with 3 hidden nods, a 6-3-1 network with 31 weights. It reduces the residual sum of squares to 1516.51 and slightly improves area under the curve to 0.67.

When we try to run the model with 1 hidden nod, a 6-1-1 network with 15 weights the residual sum of squares goes back up to 1549.47 and area under the curve goes down to 0.66.

The results of all the neural networks with one, two or three nods is better than a linear regression with same variables as it produces an area under the curve of 0.648.

Out of the three-neural network and linear regression model the neural network with 3 nods seems to be the best model as it produces the highest area under the curve and least residual sum of squares.

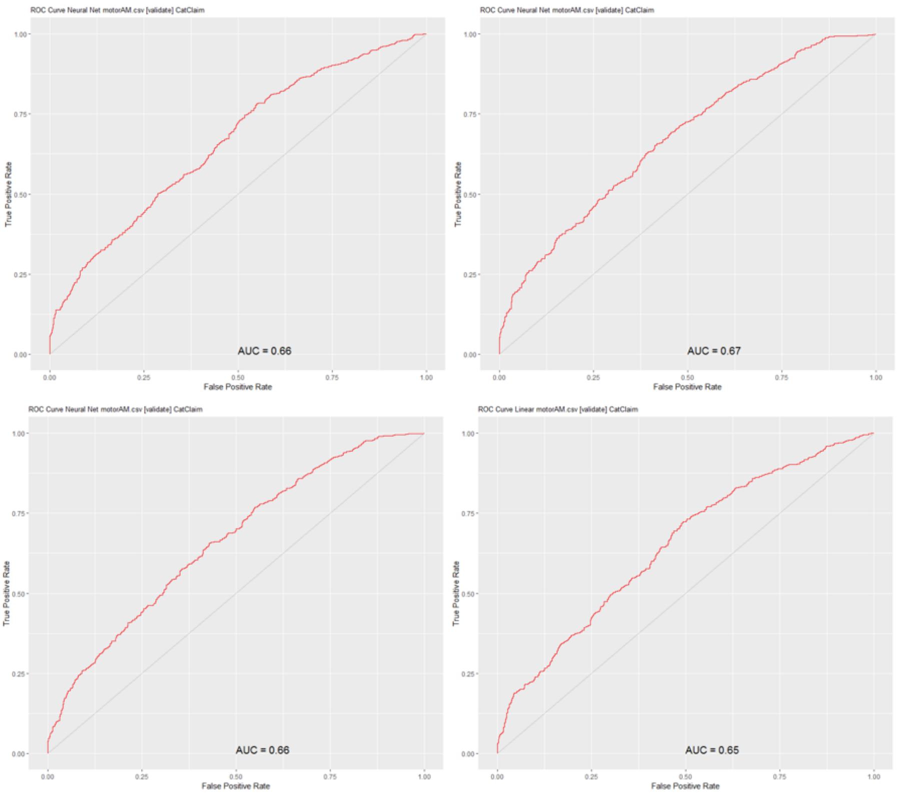


Figure 8.1: Area under the curve for Neural network 6-2-1 (top left), 6-3-1 (top right), 6-1-1(bottom left) and Liner model (bottom right)

Self-organising maps

Just like neural network we can use self-organising maps to train our model. They have an outer and inner layer but rely on unsupervised learning. It relies on clustering the data assigning a random weight to each neuron.

Figure 8.2 shows us that when we applied self-organising maps on our data we it takes roughly 85 iteration for our map to converge.

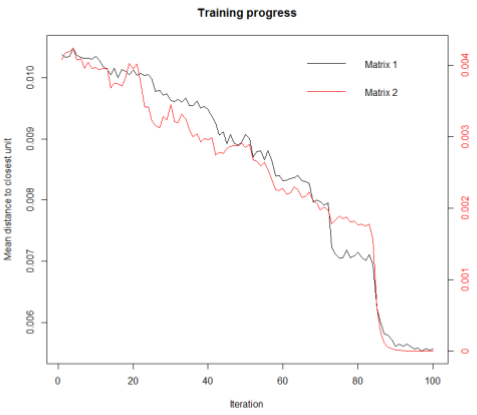


Figure 8.2: Iteration count

We can see from the count plot *Figure 8.3* the number of policies in each cell vary a lot. The light-yellow cells have far more policies in them then dark red cells. The cell count ranges from 20s policy per cell to more than 100, which is extremely high. There's also a grey cell with no policy in it.

The code plot below *Figure 8.4* shows the normalised value of each variables in Codes X and code Y shows whether its corresponding codes X nod was Yes or No claim.

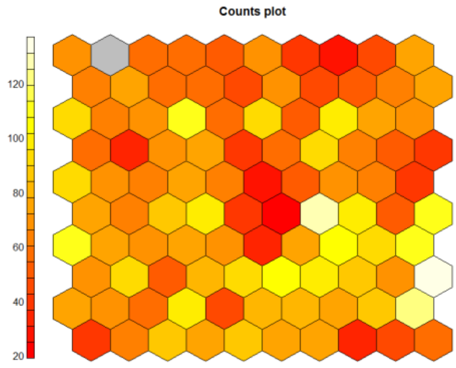


Figure 8.3: Policy count per cell

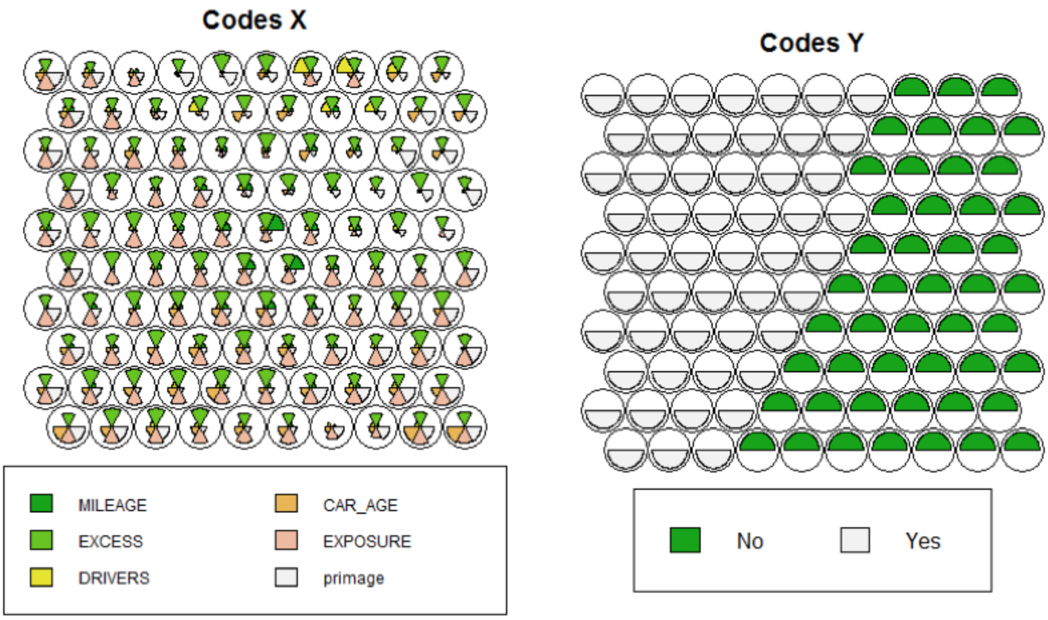


Figure 8.4: Variables distribution per cell (left), claim distribution per cell (right)

9.0 Comparison of classification results

We have explored the same data set through many different models and methods. There is no ideal way for analysing data. The data miners need to explore every model and function available to them in finding the most relevant and useful information.

There are many measures the data miner can look at to compare different prediction models. These are error matrixes, area under the ROC (receiver operation characteristics) curve, risk etc.

In this section, we will consider these measures of the models and methods we explored in this report.

Association analysis is not a predictive model. It just uses the information at hand calculates support, confidence and lifts between different products. We found which types to claims complements each other and which are least likely to appear together.

Comparison table

The *Table 9.1 below* shows all the important figures which can be used to compare different classification results. The figures in this table are derived from the validation dataset, hence the error will vary from the figures given above in section 5 to 8 of this report.

Table 9.1: Comparison of different models

Method	**Overall error**	**Average class error**	**Area under ROC curve**	**Area under the recall**	**Area under the risk**
Decision Tree (No priors or loss matrix)	40.7%	41.05%	0.631	75%	69%
Decision Tree (loss matrix, 0, 2, 1, 0)	43.9%	45.1%	0.549	69%	68%
Decision Tree (priors 0.8, 0.2)	43.9%	45.1%	0.549	69%	68%
Random forest (4 variables)	39.2%	39.25%	0.654	76%	76%
Adaptive Boosting	36.8%	36.75%	0.688	79%	70%
Support Vector Machines	38.6%	38.55%	0.662	77%	67%
Linear regression model	40%	39.85%	0.649	76%	68%
Neural Network (6-3-1)	38.6%	38.6%	0.673	78%	69%
Neural Network (6-2-1)	40.8%	40.75%	0.661	77%	69%
Neural Network (6-1-1)	39.4%	39.25%	0.662	77%	68%

Error matrixes and Error rate

The error matrix gives the error amount and percentage from our validation dataset. It gives us a matrix of true negative, false negative, false positive and true positive. We can use these values to calculate the false negative error and false positive error proportion or percentage. In this case, they are false prediction of No claim when the claim was yes and false prediction of Yes claim when in fact there was no claim.

Overall error are the total wrong predictions out of all the predictions made, i.e. (FN + FP)/(TN+FN+FP+TP). It's a good way to check the overall accuracy of our prediction model. Of all the models, we used adaptive boosting had the lowest error rate of 36.8% and the weighted decision tree has the worst of 43.9%. *Table 9.2* shows the wrong and correct prediction cases in both boosting and prior weighted decision trees. The validation overall error is calculated by 552/1500 * 100 and 658/1500 * 100.

Table 9.2 Validation errors for Prior Weighted Boosting and decision tree

	Boosting		**Prior weighted decision tree**	
	Predict No claim	**Predict Yes claim**	**Predict No claim**	**Predict Yes claim**
Observe No claim	471	300	759	12
Observe Yes claim	252	477	646	83

Average class error is the average of false positive error and false negative error (FN% + FP%)/2. Often the number of either one of false negative error or false positive error will be quite large compared to the other, which will skew the overall error rate towards it. In our result, adaptive boosting has the lowest average class error of 36.75% and weighted decision tree had the highest of 45.1%.

In case of boosting the we can in *Table 9.3* the false negative and false positive error are quite even, 38.9% and 34.6%. so, the average error of 36.75% and overall error of 36.8% gives an accurate reflection of the results produced by this model.

But in the prior weighted decision tree the percentages are 1.6% and 88.6% so the average error rate of 45.1% and overall error rate of 43.9% are quite inaccurate.

Table 9.3 Percentage errors for Prior Weighted Boosting and decision tree

	Boosting			**Prior weighted decision tree**		
	Predict No claim	**Predict Yes claim**	**Error**	**Predict No claim**	**Predict Yes claim**	**Error**
Observe No claim	31.4%	20%	38.9%	50.6%	0.8%	1.6%
Observe Yes claim	16.8%	31.8%	34.6%	43.1%	5.5%	88.6%

Area under the recall and risk is the percentage of true positive and true negative. The higher the percentage the better the model. Adaptive boosting has the highest area under the recall of 79% and weighted decision tree have the lowest of 69%. Random forest has the highest area under the risk of 76% and Support Vector Machines has the lowest of 67%.

Area under the ROC curve is a chart of false positive and true positive. Its values between 0.5 and 1 with 0.5 being the worst result and any value above 0.9 considered excellent. The minimum requirement is a value of 0.7. if a prediction model results in an area under the ROC curve of less than 0.7 its, not considered a viable model. In our findings, we could not get a single model with area under curve of 0.7, adaptive boosting had the highest value of 0.688 and prior weighted decision tree the lowest 0.549. In *Figure 9.1* we can see the difference between the two area under the ROC curve for boosting model and prior weighted decision tree.

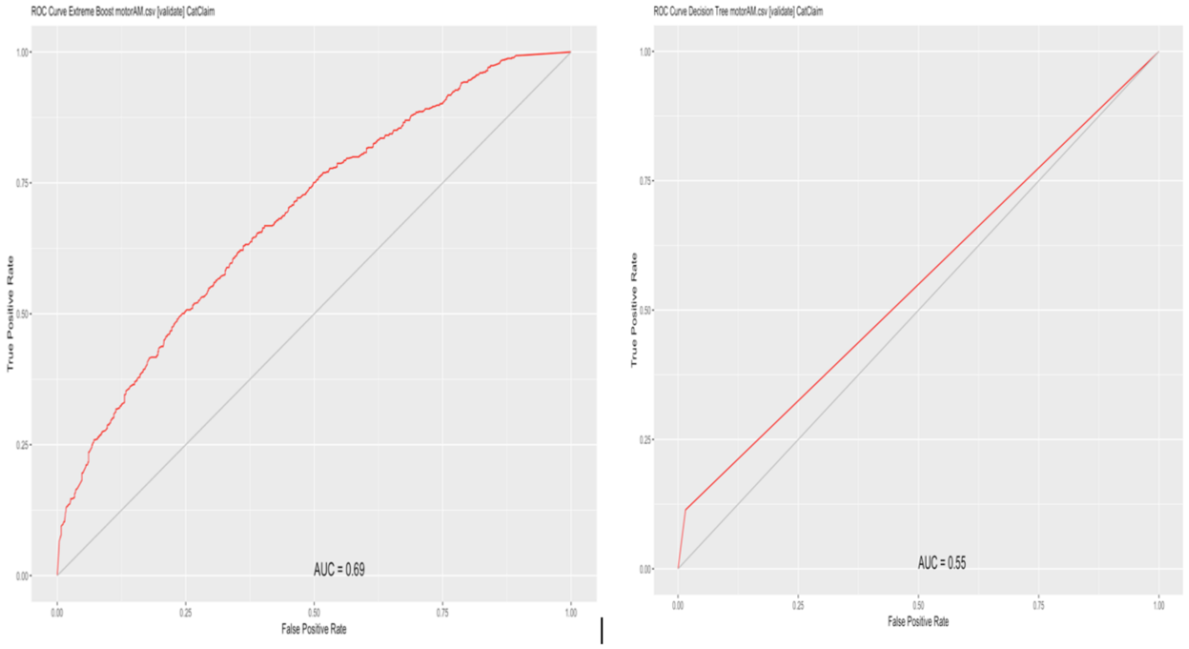


Figure 9.1: Area under the ROC curve for Boosting (left) and weighted decision tree (right).

After comparing all these methods and models adaptive boosting seems to be the most accurate predictor but with none of the models producing an area under the curve of 0.7 neither one of them could be of any use.

10.0 Conclusion

We started our report by looking at the summary statistics, looking for correlations between different variable and made changes like transforming our variables. Its, important to explore the data to get the best use of it. Often numerical exploration is not sufficient so we use Ggobi to visually explore the data and look for pattern that would help us in our prediction.

We use associative analysis to find relationship between different claims. The aim of association analysis is to find the likelihood of 2 claims occurring together. This can be used to train staff in claims which have higher likelihood of occurring together.

Then we use various models to make prediction weather a claim would be taken or not. We use many categorical and numerical variables to make this prediction. Often significant predictors are given importance over others and non-significant predictors are ignored. E.g. in the linear regression models we only selected the few significant variables. While developing a neural network we only used scaled (transformed) numerical variables.

We also explore deep learning methods like support vectors machines. We used different hidden nods to see which one will give us the lowest residual values. We used self-organising maps to cluster the data based on important variables.

Result of the comparison.

To compare the different methods, we used five common results, overall error, average class error, area under the ROC curve, and area under the recall and risk. We want the lowest error and highest area covered in each of the cases. We found out that adaptive boosting turns out to be the best methods out of the ones we tried. It produces the lowest error and highest area under the ROC curve. The weighted decision trees produced the highest error and the area under the ROC curve covered was least.

Which model will we select?

Even though we found that adaptive boosting to be the most efficient model of the ones we considered, the difference between them is very small and neither one of them produces a result which we would be confident enough to use for business decision making. The error rate for all of them is very high and no one can achieve the minimum acceptable value 0.7 for area under the ROC curve.

Hence as often in data mining the past data is not sufficient for creating prediction models.