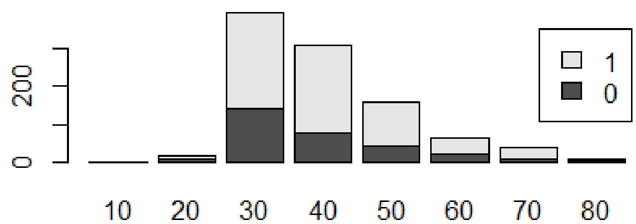
The German Credit dataset has data on 1000 past credit applicants, described by 30 variables. Each applicant is also rated as “Good” or “Bad” credit (encoded as 1 and 0 respectively in the Response variable). Objective is to develop a credit scoring rule that can be used to help determine whether a new applicant presents a good or bad credit risk.

Proportion of Good to Bad cases

GOOD CASES(1)	BAD CASES(0)
700	300

Column 'Age' has 9 missing values. Out of which 5 are Good case observation and 4 are bad cases. Distributing the missing values into the plot as shown below is not affecting the inference of the graph. So even if we do not handle these values, there won't be changes to the proportion of Good & Bad cases.



Predictor variables - DURATION, AMOUNT, INSTALL_RATE, AGE, NUM_CREDITS, NUM_DEPENDENTS

vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis
DURATION	1	1000	20.903	12.058815	18	19.4725	8.8956	4	72	68	1.0909038
AMOUNT	2	1000	3271.156	2822.625	2319.5	2754.5325	1627.1535	250	18424	18174	1.9438389
INSTALL_RATE	3	1000	2.973	1.1187147	3	3.09125	1.4826	1	4	3	-0.5297551
AGE	4	991	35.48335	11.37077	33	34.09206	10.3782	19	75	56	1.0285522
NUM_CREDITS	5	1000	1.407	0.5776545	1	1.33375	0	1	4	3	1.2687608
NUM_DEPENDENTS	6	1000	1.155	0.3620858	1	1.06875	0	1	2	1	1.90372

Frequencies of different categorical variables CHK_ACCT

	<0 DM	0<...<200 DM	=>200	No Checking A/c
Bad	135	105	14	46
Good	139	64	49	348

HISTORY

	No Credits	All Credits Paid Back Duly	Existing credits paid back duly	Delay in paying off in the past	Critical A/c
Bad	25	28	169	28	50
Good	15	21	361	60	243

EDUCATION

	No	Yes
Bad	278	22
Good	672	28

SAV_ACCT

	<100 DM	100<=…<500 DM	500<=…<1000 DM	=>1000 DM	Unknown/No Savings Acct
Bad	217	34	11	6	32
Good	386	69	52	42	151

EMPLOYMENT

	Unemployed	<1yr	1<=..<4 Years	4<=..<7 Years	>=7 Years
Bad	23	70	104	39	64
Good	39	102	235	135	189

PRESENT_RESIDENT

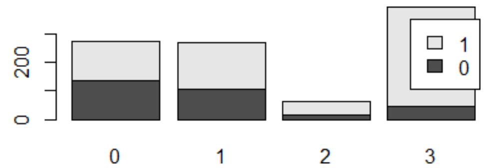
	<=1 Year	1<..<=2 Years	2<..<=3 Years	>4 Years
Bad	36	97	43	124
Good	94	211	106	289

JOB

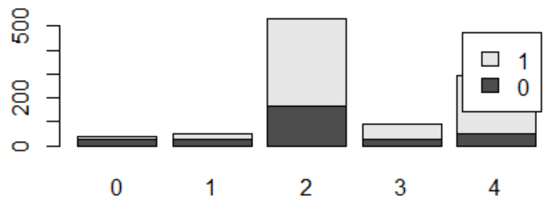
	Unemployed/Unskilled-Non-Resident	Unskilled-Resident	Skilled Employee/Official	Management/Self-Employed
Bad	7	56	186	51
Good	15	144	444	97

While examining variable plots following are some of the observation

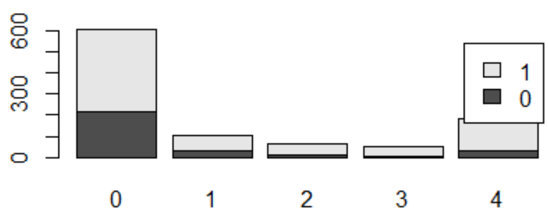
CHK_ACCT - We can see from the variable plot of CHK_ACCT that customers who don’t have any checking account(3) have larger proportion of Good cases which is something not expected.



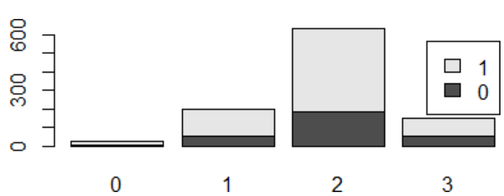
HISTORY: The Credit History variable plot below indicates that customers with existing credits paid back duly (2) have larger proportion of Good Cases. Also customers who have critical account(4) have larger proportion of Good Cases compared to Bad Cases.



SAV_ACCT - When average balance in savings account is less than 100DM(0) and when the savings account is unknown or there is no savings account(4) the proportion of Good cases is higher which is odd.



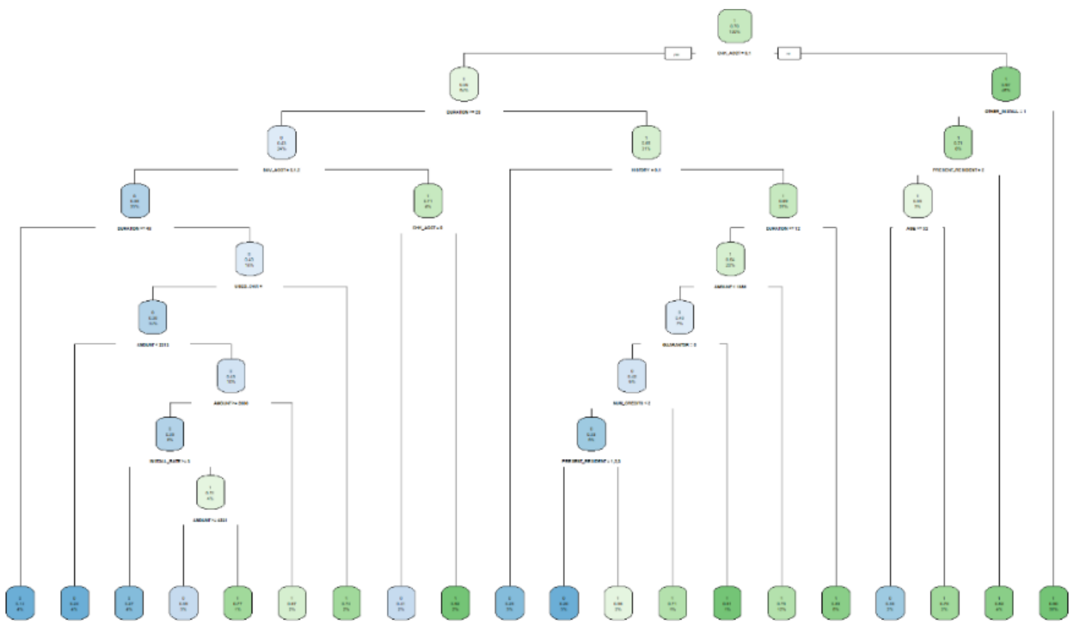
JOB - When nature of job is skilled employee/official(2) the proportion of Good cases is high as compared to customers with unskilled jobs.



Most relevant variables for the outcome of interest is as following based on correlation coefficient

HISTORY	0.2287847
DURATION	-0.2149267
SAV_ACCT	0.1789427
AMOUNT	-0.1547584

Decision Tree (on full data using rpart package)



Critical parameters in building the model are: a) Minsplit – minimum number of splits at a node in order for a split b) Cp – complexity parameter which helps to reduce the complexity of the tree.

Use VarImp (model) function to figure out variable importance

Parameters	Overall
HISTORY	55.94
CHK_ACCT	55.43
AMOUNT	47.73
DURATION	41.63
SAV_ACCT	36.69
REAL_ESTATE	27.31
USED_CAR	13.82
PRESENT_RESIDENT	13.47
INSTALL_RATE	13.38
EMPLOYMENT	13.17
GUARANTOR	11.50
AGE	8.30

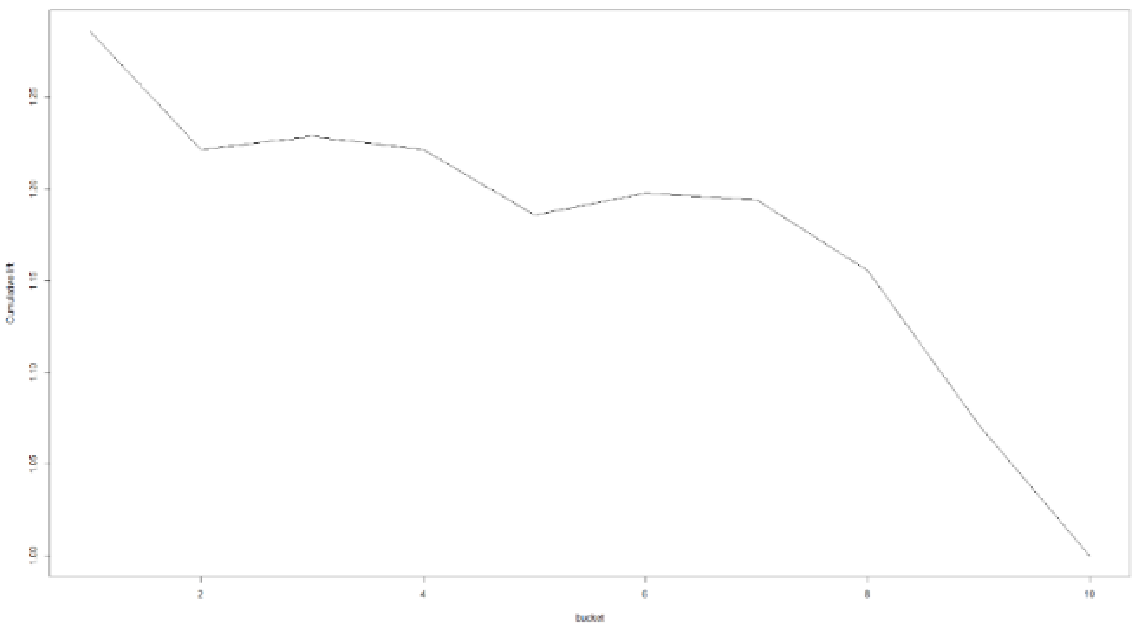
Confusion Matrix (Training data):

Performance Evaluation Metrics (Training data):

		Actual	
		0	1
Prediction	0	173	62
	1	127	638

Parameters	Values
Accuracy	81%
TP rate	91%
FP rate	58%

Lift Chart:

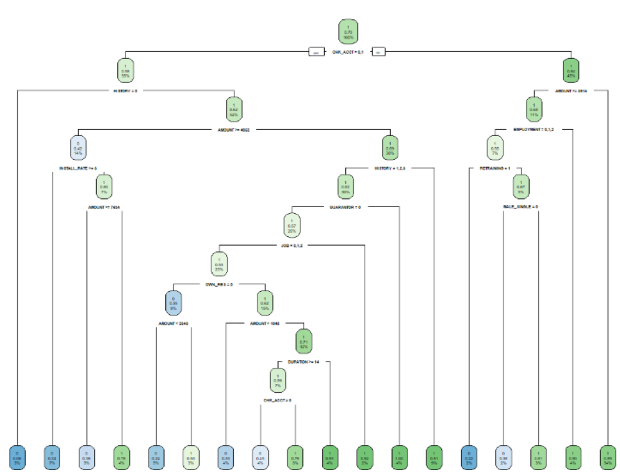


Decile	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Lift	1.29	1.22	1.23	1.22	1.18	1.19	1.19	1.15	1.08	1

Lift chart tells that top 10% of the cases have almost **30% higher chances** of being classified as 'Good' credit risk rather than choosing random target of 10% of cases. 10^{th} decile lift is 1, indicating that targeting all the cases will have no difference compared to our full set of cases.

Decision trees are not robust because the model is extremely sensitive to data: any small change in data can result in a completely different tree. Moreover, complexity of the model tends to increase with the size of dataset.

Divided the data into Training and Validation sets. Consider a partition of the data into 50% for Training and 50% for Test



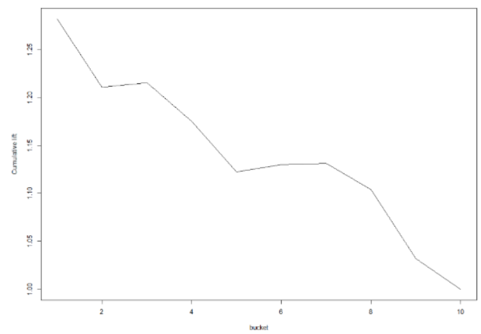
Confusion Matrix:

		Actual	
		0	1
Prediction	0	73	56
	1	76	295

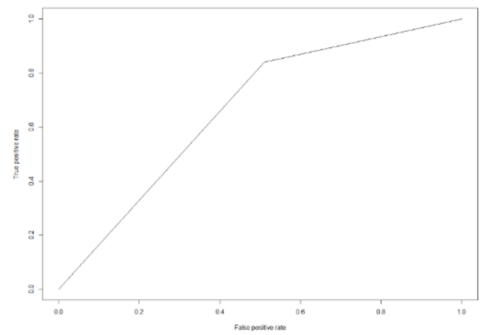
Performance Evaluation Metrics:

Parameters	Values
Accuracy	74%
Sensitivity	84%
Specificity	49%
Precision	80%

Lift Chart:



ROC curve:



Decile	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Lift	1.28	1.2	1.2	1.18	1.13	1.13	1.14	1.11	1.03	1

AUC (Area Under Curve): 0.67

Critical parameters which have been useful in deriving good model performance has been the following:

1. Split type – 'Gini' or 'Information' which helps to decide the purity of the node.
2. Minsplit – Minimum of cases to split at a node. Higher the number, smaller the tree size as there are more number of cases required to split the root node
3. Cp (Complexity parameter) – Helps to determine the complexity or size of the tree. As the number increases, size of the tree becomes smaller and hence impacts the overall accuracy of the outcomes.

Complexity parameter plays a crucial role in developing a good model performance. It helps to determine how the cost of the tree is compensated by number of terminal nodes, resulting in normalize cost. Small cp results in bigger tree and hence resulting in overfitting. Larger cp results in smaller tree and hence underfitting.

Variable Importance:

Using rpart

Parameters	Value
AMOUNT	53.92472
DURATION	30.909232
HISTORY	28.724536
OWN_RES	25.095145
PROP_UNKN_NONE	24.615467
SAV_ACCT	20.132758
CHK_ACCT	19.403457
GUARANTOR	14.537021
REAL_ESTATE	14.397315
AGE	12.501265
EMPLOYMENT	11.420922
NEW_CAR	7.267137
RETRAINING	7.166651
MALE_SINGLE	5.985569
OTHER_INSTALL	5.449124
NUM_CREDITS	5.134918
JOB	5.019453
INSTALL_RATE	4.587894
RENT	1.74434

Using C5.0

Parameters	Value
CHK_ACCT	100.0
HISTORY	94.0
GUARANTOR	45.6
REAL_ESTATE	19.8
SAV_ACCT	19.0
JOB	11.6
AMOUNT	6.2
NUM_DEPENDENTS	3.8
MALE_SINGLE	1.6

Variable importance in building models using rpart and C5.0 have common variables. But the importance value assigned to those variables between best models are different.

Performance Comparison (keeping the parameters the same)

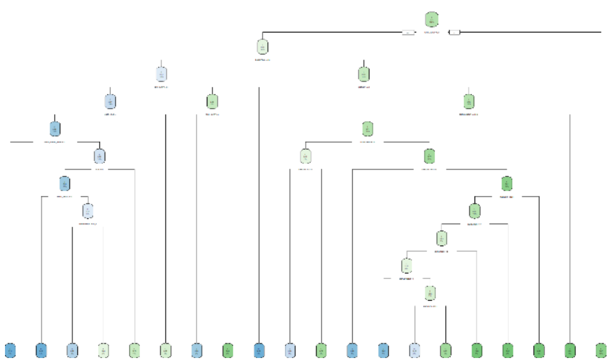
Using rpart

Parameters	50 – 50	70 – 30	80 - 20
Overall Accuracy	74%	76%	74%
Specificity	84%	89%	85%
Sensitivity	49%	48%	46%

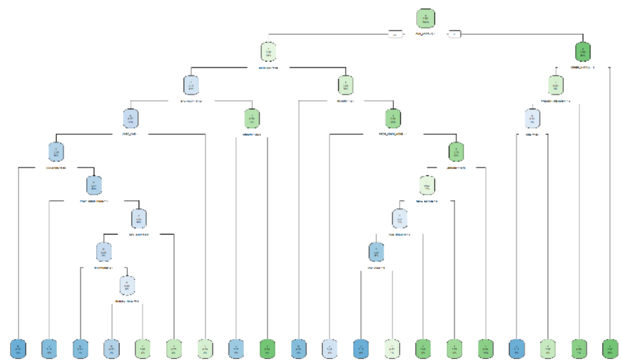
Precision	80%	78%	80%
Top Decile Lift	1.28	1.43	1.39
Area Under Curve	0.67	0.69	0.65

Keeping the decision tree learner parameters the same, it is evident that performance and model varies with the size of training and test data. Performance metrics keeps varying indicating that for different size of the training data, parameters have to changed accordingly in order to obtain the best model.

r-part (70/30)



r-part (80/20)



Considering the models with different size of the training data, we can infer the upper part of the tree are similar. Predictors with higher information gain / purity / correlation clearly explains the distribution of the dependent variable. Hence they are less affected by the size of training data. So they tend to be root nodes or non-leaf nodes below the root nodes. While predictors with lower information gain / purity have lesser details on distribution of the dependent variable. So they sit at the bottom of tree. With increase in size of the data, impurity increases and hence predictors sitting at the bottom of the tree are subjected to change.

Ideally, 70 – 30 combination of training and test data would be a preferred combination for building a model. This helps to build a stable model and provides sufficient data to test the model and gain a higher confidence level.

Comparing different threshold values on a model developed on full data to reduce FP count

Parameters	Threshold 0.5	Threshold 0.6	Threshold 0.7
Overall Accuracy	81%	81%	80%
Sensitivity	58%	60%	62%
Specificity	91%	89%	88%
Precision	83%	84%	85%
Top Decile Lift	1.28	1.29	1.29
Area Under Curve	0.74	0.75	0.75
FP count	127	119	114

It is clearly seen from the table that with change in threshold value, accuracy of the model reduces but at the same time, False Positive counts also reduce indicating that overall misclassification cost will go down.

Theoritical threshold = 500 / (500+100) = 0.81

Using r-part (on full data with threshold as 0.8)

Parameters	Without Costs
Overall Accuracy	71%
Sensitivity	79%
Specificity	68%

Confusion Matrix (with threshold as 0.8)

		Actual	
		0	1
Prediction	0	237	222
	1	63	478

Using misclassification costs in building tree models yields a completely different tree from the ones generated without the costs. Below is the summary of performance for tree models with 50% data with cut-off value as 0.5

Precision	89%
Top Decile Lift	1.31
Area Under Curve	0.74
FP count	63

Using r-part

Parameters	Without Costs	With costs
Overall Accuracy	74%	69%
Sensitivity	84%	65%
Specificity	49%	71%
Precision	80%	83%
Top Decile Lift	1.28	1.34
Area Under Curve	0.67	0.68

Confusion Matrix (with costs)

		Actual	
		0	1
Prediction	0	98	102
	1	51	249

Using C5.0

Parameters	Without Costs	With costs
Overall Accuracy	73%	58%
Sensitivity	88%	81%
Specificity	37%	48%
Precision	77%	85%
Top Decile Lift	1.22	1.28
Area Under Curve	0.63	0.64

Confusion Matrix (with costs)

		Actual	
		0	1
Prediction	0	120	182
	1	29	169