ASSIGNMENT-1

Data Science & Analytics

BY

Anoop Nair

(R00223644)
Munster Technological University

INTRODUCTION

The Analysis is around the dataset Credit_Risk_25_final-2.csv. The Data is of a financial institution with 807 past loan customer cases and around 14 attributes of them. The main investigation around this data is to find out a model which can classify a loan application as bad or good before load approval. So that they can prevent any future losses due to bad loans. The description of data is as below.

Variable	Type	Description	Independent/Dependent/Identity
ID	Numeric	An Identity Column	Identiity
Checking.Acct	Categorical/factor	Type of Account ex: 0Balance, High etc.	Independent
Credit.History	Categorical/factor	CreditHistory in the past ex: All Paid, Bank Paid	Independent
Loan.Reason	Categorical/factor	Reason for Loan. Ex: Business,New Car etc.	Independent
Savings.Acct	Categorical/factor	Balance of the Account. Ex: High, Low etc.	Independent
Employment	Categorical/factor	How long has the person been employed? Ex: Long, Medium etc.	Independent
Personal.Status	Categorical/factor	Marital Status ex: Single, Divorced etc.	Independent
Housing	Categorical/factor	Typoe of Housing you live in. ex: Own	Independent
Job.Type	Categorical/factor	Type of Job. Ex: Management, Skilled etc.	Independent
Foreign.National	Categorical/factor	If a Foreign National / Not. Ex: Yes/No	Independent
Months.since.Checking.Acct.opened	Numeric	Time lasped in months since the account was open. Ex: 5,12 etc.	Independent
Residence.Time.In.current.district	Numeric	Time lasped in months since staying at the current residence. Ex: 5,12 etc.	Independent
Age	Numeric	Age of the person ex: 40.3,50.8	Independent
Credit.Standing	Categorical/factor	If the Loan was Good / Bad. Ex: Good, Bad etc.	Dependent

Fig1: Shows the variables in the Data and their description & datatypes

LOADING & CLEANING.

(Please refer from line 34 to 131 in the accompanying R -script file for the code)

- The data was loaded from csv file to data frame & names of certain columns were changed as in Fig 2. (Henceforth, we will be using new column names to explain the report)
- The categorical variables were converted into factors and the Age column was converted into integer (as we don't need decimal level precision for Age.)

- There were few missing values in columns like Employment,Housing & Martitalstatus, which was imputed using the mode value of the respective Variable.
- There was also a negative value for CurrentResidenceDuration Which was imputed with the median of that variable.

Old Column Names	New Column Names
ID	ID
Checking.Acct	Accounttype
Credit.History	CreditHistory
Loan.Reason	LoanReason
Savings.Acct	SavingsAccount
Employment	Employment
Personal.Status	MaritalStatus
Housing	Housing
Job.Type	JobType
Foreign.National	ForeignNational
Months.since.Checking.Acct .opened	AccountActivemonths
Residence.Time.In.current.district	CurrentResidenceDuration
Age	Age
Credit.Standing	CreditStanding

Fig2: Shows Old & New Column names.

EDA & DESCRIPTIVE STATISTICS.

(This address Question A, Please refer from lines 134 to 469 in the accompanying R - script file for the code)

Summary for Categorical variables:

- Firstly, let us look at the distibution of each class within each categorical variables.

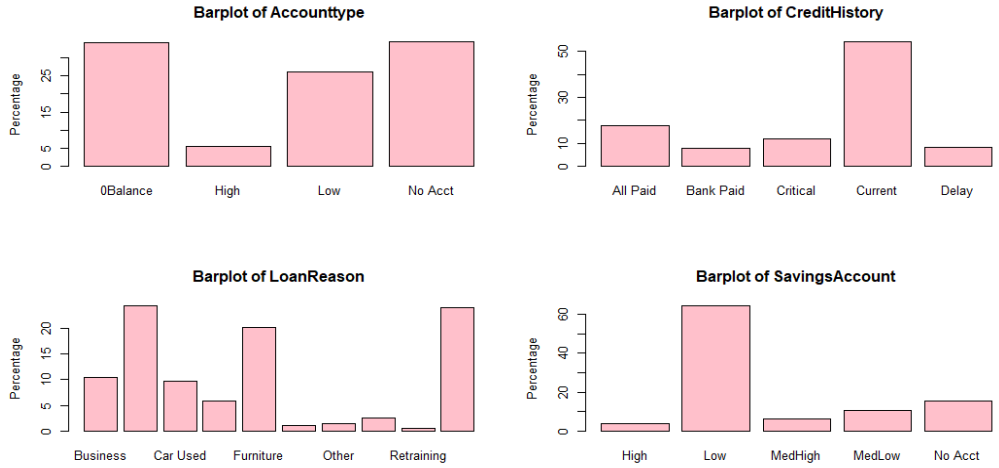


Fig3: Classwise % for variables Accountype, CreditHistory, Loan Reason and SavingsAccount.

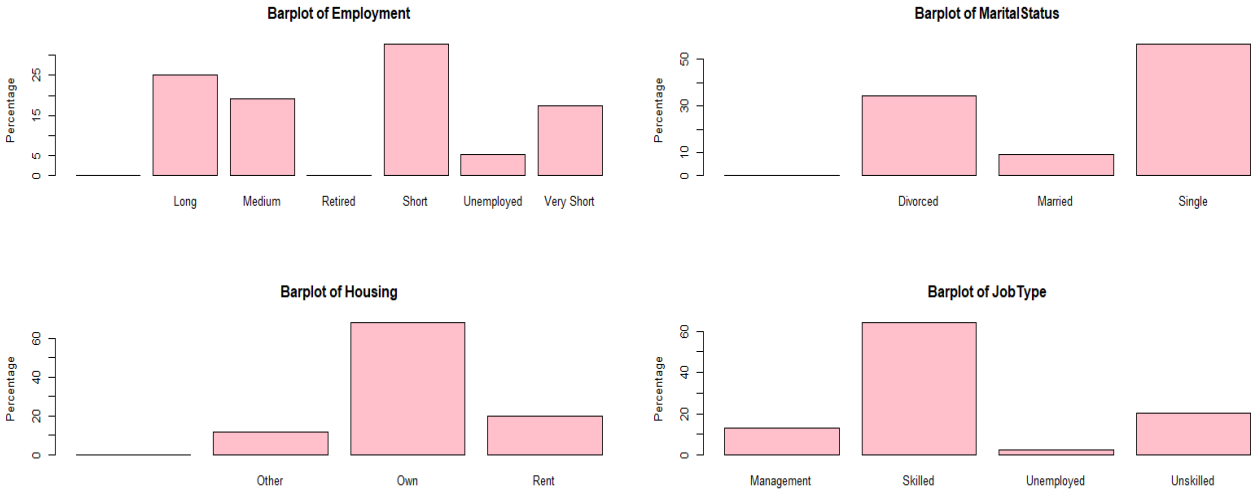


Fig4: Classwise % for variables Employment, MaritalStatus, Housing & JobType.

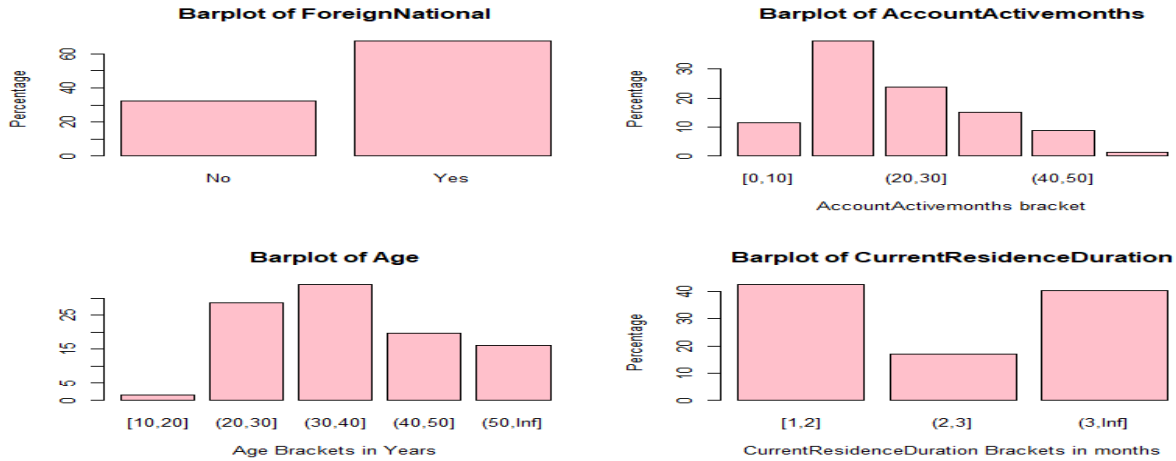
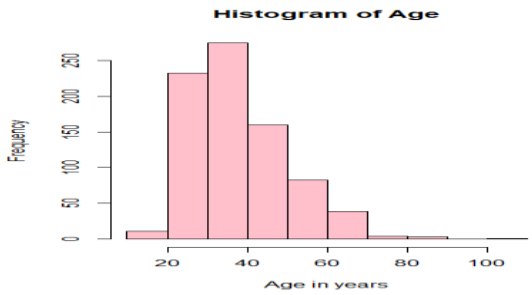


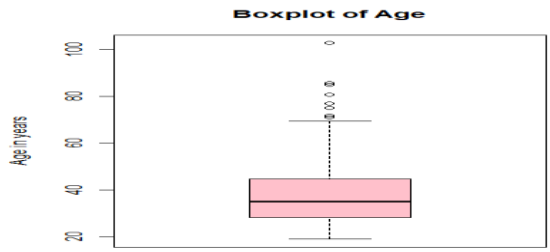
Fig5: Classwise % for variables ForeignNational , also for AccountActivemonths, Age & CurrentResidenceDuration, as they are grouped into bins.

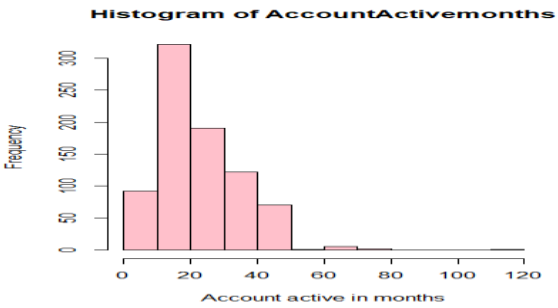
Few Observations from Figure 3,4 & 5:

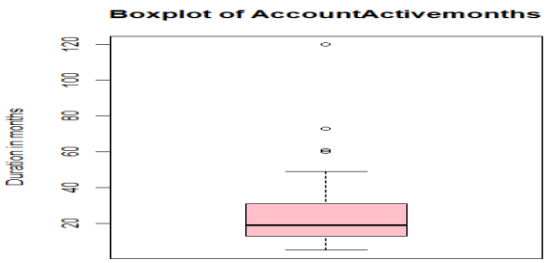
- Current class constitutes about 50% of the values in CreditHistory Class.
- Low Class constitutes >60% of values in SavingsAccount.
- About 60% of the values in JobType category is Skilled.
- >60% of the loan applications are from Foreign Nationals.
- Age group between 20-40 has about 60% of the Loan.
- More than 60% of the applications has their own House.

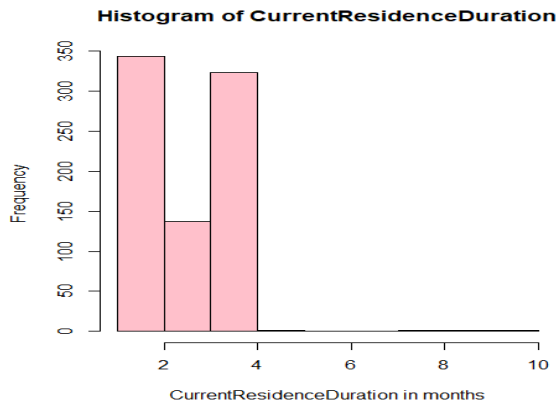
Summary for Continuous variables:











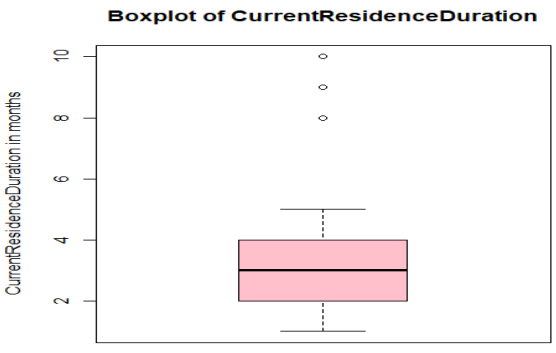


Fig6: Boxplot & Histogram for continuous variables

Few Observations from figure 6:

- The Mean and median of AccountActivemonths are ~23 and ~19 respectively. Since the mean > median it's left skewed. Also, there are few outliers for it on the higher side. Standard deviation is 12.8.
- The Mean and Median for Age is 37.7 & 35 respectively. Since the mean> median its slightly left skewed. Age also has few outliers around >70. The standard deviation is 12.15.
- The Mean and median for CurrentResidenceDuration are 2.8 and 3. The standard deviation is 1.15. There are few outliers as we can see in the bottommost boxplot.The boxplot looks symmetrical on either side of median.

Bi-variate Analysis:

- The distribution of Bad and Good loans in the dataset is 40.8% & 59.2% respectively.
- Let us try doing a Bivariate analysis i.e. (CreditStanding vs All other variables) where we check the distribution of Bad and Good loans against each class to see which class of a predictor contribute to what % of Bad or Good Loans.

Accounttype				
	OBalance	High	Low	No Acct
Bad	40.4	6.1	29.8	23.7
Good	29.5	5.2	23.4	41.8

SavingsAccount					
	High	Low	MedHigh	MedLow	No Acct
Bad	2.7	68.4	6.1	7.6	15.2
Good	4.2	61.3	6.3	12.3	15.9

Employment						
	Long	Medium	Retired	Short	Unemployed	Very Short
Bad	19.5	10.6	0.3	46.8	4	18.8
Good	28.9	24.9	0.2	23.2	6.3	16.5

JobType				
	Managem	Skilled	Unemployed	Unskilled
Bad	10.6	66.9	0.6	21.9
Good	14.9	62.1	3.6	19.5

Credit History					
	All Paid	Bank Paid	Critical	Current	Delay
Bad	1.5	3	28.9	58.7	7.9
Good	28.9	11.3	0.2	51	8.6

Loan Reason										
	Business	New Car	Used	Education	Furniture	Large App	Other	Repairs	Retraining	Small Appliance
Bad	9.1	27.1	8.2	7.9	23.7	0.9	1.2	1.2	0	20.7
Good	11.5	22.4	10.7	4.4	17.6	1	1.7	3.6	0.8	26.4

Marital Status			
	Divorced	Married	Single
Bad	39.8	9.1	51.1
Good	30.5	9	60.5

Foreign National		
	No	Yes
Bad	32.5	67.5
Good	32.2	67.8

Housing			
	Other	Own	Rent
Bad	14.9	62	23.1
Good	10	72.2	17.8

CurrentResidenceDuration (bins)			
	[1,2]	(2,3]	(3,Inf]
Bad	45	15.5	39.5
Good	40.8	18	41.2

AccountActiveMonths (bins)						
	[0,10]	(10,20]	(20,30]	(30,40]	(40,50]	(50,Inf]
Bad	7	41.6	21.9	17	11.2	1.2
Good	14.4	38.5	24.9	13.8	7.1	1.3

Age (bins)					
	[10,20]	(20,30]	(30,40]	(40,50]	(50,Inf]
Bad	3.3	45.6	26.1	16.1	8.8
Good	0	17.2	39.3	22.4	21.1

Fig7: Bivariate distribution between Credit standing and other variables.

Few Observations from Figure 7:

- CreditHistory has an interesting distribution where it contributes to 28% of the total Bad from the group, but only could contribute 0.2% towards the Good Loans.It is highly pure split. And a very good indicator of Bad Loans.
- In Accounttype Variable 0Balance account contributed 40% of Bad loans,but only contributes 29% towards Good.
- BankPaid class of Credit History Variable contribute 11.3% to towards Good Loans, but only 3% towards Bad.
- People who have a Short Employment contributes 47% of the Bad loans,but only contributes 25% of the Good loans.
- Similarly, 24.9% of those with medium employment contributes towards Good Loans, but only 10.6% towards Bad loans.
- What is interesting is from those who don't have a job 3.6% contributes to Good Loans but 0.6% to Bad.This is a small number still an interesting fact.
- People between the age of 20-30 contributes 45.6% of Bad loans but only 17.2 % of good loans.

Tri-variate Analysis:

- Critical History points a decent split between Bad and Good loans.
- Next step would be to see if any other variables provide an additional knowledge on top of what critical history can provide. So we try doing various trivariate analysis with Credit history & CreditStanding as constants.

	Credit History				
	All Paid	**Bank Paid**	**Critical**	**Current**	**Delay**
Bad	1.5	3	28.9	58.7	7.9
Good	28.9	11.3	0.2	51	8.6

Credit History	All Paid	Bank Paid	Critical	Current	Delay
	Age Grp				
Bad	[0,10]	4.3	8.7	13	69.6	4.3
	(10,20]	2.9	0	33.6	60.6	2.9
	(20,30]	0	6.9	27.8	51.4	13.9
	(30,40]	0	3.6	25	60.7	10.7
	(40,50]	0	0	32.4	54.1	13.5
	(50,Inf]	0	25	0	75	0
Good	[0,10]	29	11.6	0	55.1	4.3
	(10,20]	35.9	9.2	0.5	52.2	2.2
	(20,30]	26.9	10.9	0	44.5	17.6
	(30,40]	24.2	4.5	0	54.5	16.7
	(40,50]	11.8	29.4	0	52.9	5.9
	(50,Inf]	0	50	0	50	0

Fig8: Trivariate distribution between Credit standing and Credit History & Age_grp

- From the below graph Fig 8 we can see that , Even though Bank paid is generally considered good. 25% of those with age above >50 years are bad Loan.

CreditHistory	All Paid	Bank Paid	Critical	Current	Delay
	Savings Account				
Bad	High	11.1	33.3	0	55.6	0
	Low	1.3	0	32.4	54.7	11.6
	MedHigh	0	10	40	50	0
	MedLow	0	0	24	76	0
	No Acct		2	10	16	72
Good	High	45	5	0	40	10
	Low	27	12.6	0.3	54.9	5.1
	MedHigh	13.3	6.7	0	66.7	13.3
	MedLow	15.3	22	0	44.1	18.6
	No Acct	48.7	1.3	0	38.2	11.8

Fig9: Trivariate distribution between Credit standing and Credit History & SavingsAccount

- From Fig 9 we can see that credithistory current and Medlow has a high 76% contributions towards Bad loans.
- Going through all the trivariate combinations and following all the numbers is a cumbersome task. So, let's, look for better methods to identify the best combinations amongst variables, to give greater classification power for categorizing a loan application as Bad or Good.Not just that,We can go wrong on the process of interpreting numbers, relationships and the story that they are planning to tell.
- Let's look out at an improved and more appropriate approach than the Bi-variate and Trivariate Analysis.

ANALYSIS

(Please refer from line 470 onwards in the accompanying R -script file to check the code).

- At first, if we need to create an algorithm to analyse data and later want to check the efficiency of the algorithm. So, the first step is to split the data into two, namely training and test set.
- The training set will be used to run the algorithm and build the model, and once the model is built. It is tested on the test data.
- The industry standard for splitting the data is 75% for training and 25% for test. (This addresses Question B, refer Code between lines 470 and 485)
- Our whole objective is to find a model that would best classify the predictive class CreditStanding, which has two classes bad and good for bad and good loans respectively.
- Few approaches which we will be considering in this report are as follows:
  i. Calculating Entropy in a recursive manner for all the variables and classes in the variables. To find out the best variable and best split class amongst all combinations, which gives the best purity in for the leaf nodes.
  ii. Use the Decision Tree to build a model and tune it, such that it gives you minimum classification error.
  iii. Use the Random Forest algorithm to improve upon the existing top two models.

Models:

i. Entropy Model:

✓ In entropy model we go about calculating the entropy for each variable and find the lowest entropy amongst them all. That becomes the most important variable for the prediction. If put in a tree structure that becomes our root node.

Entropy=-p1*log 2 p1-p2 * log 2 p2……-pn*log2pn

- ✓ Now, Entropy checks the purity of each class. The purer the class lesser is the entropy. For ex. In our case, the data is classified into bad or good loans. So, for every partition or split we try to calculate the purer node from the previous parent Node. If the Node after split is 70% bad & 30% food. Then it is considered purer in comparison to a split which gives 50% bad and 50% good.
- ✓ From Entropy we derive another concept called Information gain.
  Information gain=1-Entropy.
- ✓ Greater the information gain purer are the nodes after split.
- ✓ We build the entropy function to calculate the lowest possible amongst the variables.
- ✓ We will run the entropy on predictive variable credit standing, on the train dataset, before we look into the entropy of each variable. The entropy value for total train data is 0.9752181
- ✓ At first, we only consider the categorical variables in the data. They are Accountype, CreditHistory, LoanReason, SavingsAccount, Employment, Maritalstatus, Housing,JobType, ForeignNational.

VariableName	Entropy
Accountype	0.940409
CreditHistory	0.726727
LoanReason	0.952145
SavingsAccount	0.966109
Employment	0.915594
MaritalStatus	0.970432
Housing	0.96961
JobType	0.963646
ForeignNational	0.974868

Fig10: Non-Binary split entropy values for categorical variables

- ✓ We see that CreditHistory has the lowest entropy amongst all above categorical variables, with a value of 0.7267 .
- ✓ Hence CreditHistory becomes the rootnode. We need to note that this entropy was calculated by considering non-binary splits.i.e., if there are more than 2 classes within a variable. They were considered as it is while calculating the entropy.
  (This addresses Question C, refer Code between lines 487 and 560)

- ✓ For Binary Split, we will have only 2 classes while calculating the entropy.
- ✓ For example, let us consider the binary split entropy for the variable Marital status. Here, it has 3 classes Single, Married & Divorced.But Binary split can have only 2 classes. So, we will convert this data into 3 sets. One with Married and Others (which includes Single & Divorced), second will have Single and others (married & divorced), Third will have Divorced and Others (Married and Single).
- ✓ We will then calculate entropy with each of the three groups and calculate the lowest of them.
- ✓ If The group married and Others has the lowest entropy and if marital status is selected as the split node. Then the split will happen at Maritalstatus equal to Married.

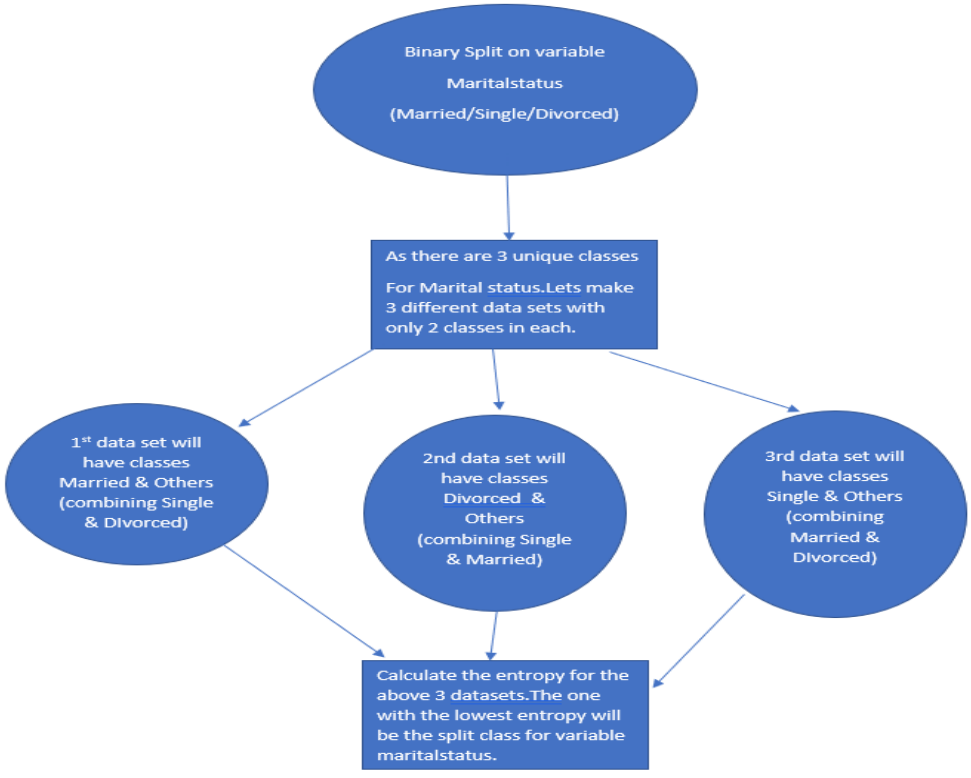


Fig11: Flow chart to explain Binary split on variable Maritalstatus.

- ✓ When we calculate the lowest entropy across each class, we get the below class and entropy for each.

Variable Name	Splitting class	Entropy
Accounttype	No Acct	0.943625284
CreditHistory	Critical	0.818396329
LoanReason	Retraining	0.970210663
SavingsAccount	Low	0.968723179
Employment	Short	0.929425904
MaritalStatus	Divorced	0.970435846
Housing	Own	0.969690102
JobType	Unemployed	0.96767588
ForeignNational	No	0.974868168

Fig12: Shows the class at which the entropy was lowest for each variable and their entropy value.

- ✓ Looking at the above value we can come to the conclusion. The root node is CreditHistory and the split will happen at class Critical. (This addresses Question D, refer Code between lines 561 and 623)
- ✓ As further additions to the model let us now add the continuous variables CurrentResidenceDuration, Age & AccountActivemonths to the entropy calculation.
- ✓ When we check the entropy we see that, Credithistory continues to have the lowest entropy in binary split at an entropy of 0.726726854088253.
- ✓ To get the split like section D we converted all the categorical variables and their classes into Binary splits and for numerical variables we tried iterating through all possible values.
- ✓ We continue to get the Credithistory rootnode splitting at the class Critical as it has the lowest entropy at 0.818396328923192. (This addresses Question E, refer Code between lines 624 and 669)
- ✓ Now to find the 2nd split after the rootnode has been split we need to repeat the entire process on two subset of data that we have after the split of rootnode.
- ✓ The first subset of data is where CreditHistory is equal to ”Critical” and the other subset is where CreditHistory is not equal”Critical”.
- ✓ Figure 13, Help us understand the 2nd split.

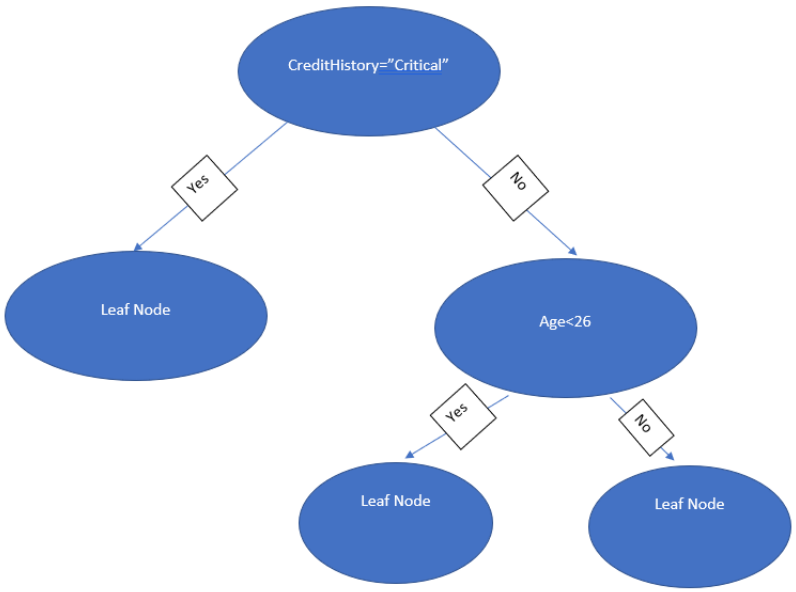


Fig13: Flow chart showing the 2nd split at Age<26.

For CreditHistory="Critical"		
Variable Name	**Splitting class**	**Entropy**
Accounttype	No Acct	0.072184486
CreditHistory	Critical	0.105591119
LoanReason	Car New	0.089165913
SavingsAccount	Low	0.100515588
Employment	Short	0.089165913
MaritalStatus	Single	0.092118137
Housing	Own	0.096534611
JobType	Skilled	0.097396601
ForeignNational	Yes	0.102572557
AccountActivemonths	19	0.092659755
CurrentResidenceDuration	2	0.085702238
Age	33	0.086444535

Fig14: Shows the class which has the lowest entropy for every predictor variable for the split credithistory = "Critical".

- When we further find the entropy for childnode where CreditHistory="Critical". We see from Fig 14 all the variables at their lower entropy class have a much lower entropy. Which is an indication that the split is almost pure and we need not further split that child node.

- For the split CreditHistory !=" Critical",when we checked the minimum split entropies for all the variables. We get the split at AGE<26
- As seen in Fig 15 Below. (This addresses Question F & G, refer code between lines 670 and 870)

For CreditHistory="NotCritical"		
Variable Name	**Splitting class**	**Entropy**
Accounttype	No Acct	0.891360167
CreditHistory	All Paid	0.827657063
LoanReason	Furniture	0.904035448
SavingsAccount	MedLow	0.90670903
Employment	Short	0.867998987
MaritalStatus	Single	0.911866671
Housing	Own	0.90961341
JobType	Unemployed	0.89913155
ForeignNational	No	0.91322693
AccountActivemonths	12	0.910817049
CurrentResidenceDuration	2	0.909351116
Age	26	0.82033672

Fig15: Shows the class which has the lowest entropy for every predictor variable for split Credithistory! =" Critical".

ii. Decision Tree:

- Now let us try building the model using the existing R package called Tree() and compare the results we got for the entropy model.
- When we run the Model we see the below result.

```
> summary (tree.CreditStanding) # misclassification error or the training error rate is 0.1815 i.e 18.15%

Classification tree:
tree(formula = CreditStanding ~ . - CreditStanding, data = dt.train)
Variables actually used in tree construction:
[1] "CreditHistory"           "Age"                     "LoanReason"              "Employment"              "Accounttype"
[6] "CurrentResidenceDuration" "SavingsAccount"
Number of terminal nodes:  12
Residual mean deviance:  0.7775 = 461.8 / 594
Misclassification error rate: 0.1815 = 110 / 606
```

Fig16: Summary of the Decision Tree Model

- We can see that misclassification error for the decision tree is 18.15% and like the entropy model, even the decision tree has the most important variables has CreditHistory and Age.
- When we plot the decision tree we get the below plot.

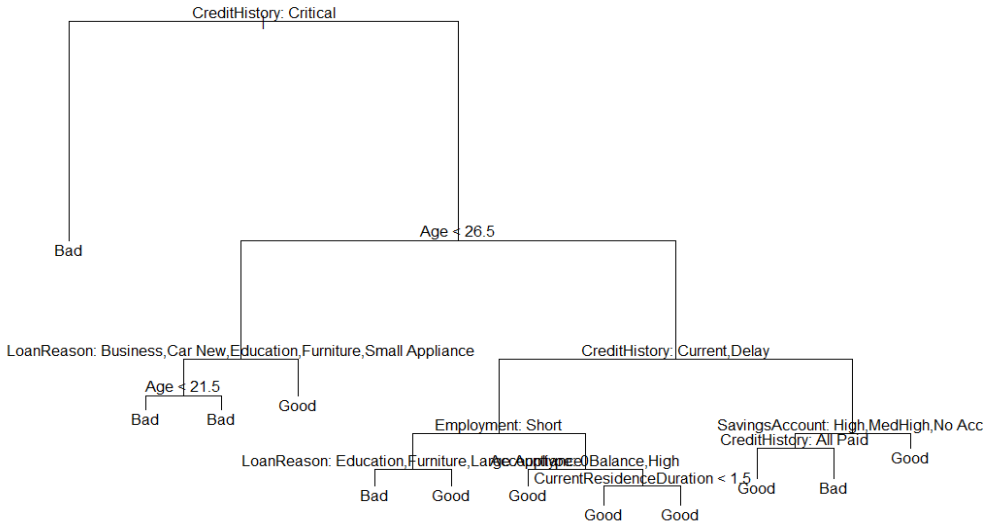


Fig17: The Decision Tree plot

- From the plot we can see that The Decision tree has the first split at CreditHistory=”Critical” and for the second splits , It did not split the left branch as we found out previously the entropy was decreased to ~0.07-0.1, which was as good as the pure Node.For the right branch the split happened at 26.5.
- The above decision tree is similar to what we found in Entropy, we dint further split the leftNode and for the right Node we split it on Age at 26.
- Now when we use the above decision tree to predict the test set, we get an accuracy of 78%.
- Now to simplify the model further /or to reduce the misclassification error we try to find various decision trees (trees with various # of leaf nodes) through cross validation and try finding the tree with minimum Misclassification error and later prune the tree accordingly.
- The tree with 9 terminal nodes has the best accuracy at 78% for the test set
- We see that there is not much improvement in the pruned tree as compared the normal decision tree. (This address Question H, refer code between lines 873 and 926)

```
> set.seed(644)
> prune.CreditStanding <-prune.misclass(tree.CreditStanding,best=9) ## pruned tree with 2 terminal nodes ###
>
> plot (prune.CreditStanding)
> text (prune.CreditStanding , pretty = 0)
> #### now to check the performance of the pruned tree with the test data lets use predict function ###
> prunedtree.pred <- predict (prune.CreditStanding,test,type = "class")
>
> table ( prunedtree.pred,CreditStanding.test)
              CreditStanding.test
prunedtree.pred Bad Good
          Bad   47    9
          Good  35  110
> (47+110)/(47+110+35+9) #78.1 % accuracy.
[1] 0.7810945
>
```

Fig18: Code for Pruned Tree

iii. Random Forest:

- Now to further improve the model let's try the RandomForest algorithm.
- Below is the result of the random forest model.

```
randomForest(formula = CreditStanding ~ . - CreditStanding, data = dt.train,      importance = TRUE, do.trace = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 18.65%
Confusion matrix:
     Bad Good class.error
Bad  184   63   0.2550607
Good  50  309   0.1392758
> (184+309)/(184+309+50+63)
[1] 0.8135314
> importance(rf.Credit)
                              Bad      Good MeanDecreaseAccuracy MeanDecreaseGini
Accounttype             14.641881 14.274089            20.423184        16.799062
CreditHistory           52.830516 46.647324            60.622290        60.077844
LoanReason              16.346393  8.362071            15.941315        24.694173
SavingsAccount          10.802812 11.400747            15.295353        14.606382
Employment              21.432207 18.006850            27.717157        27.422065
MaritalStatus            6.436373  5.182946             7.958950         8.099497
Housing                 13.601088  4.842073            13.210010         9.398819
JobType                 10.060737  8.250137            12.887874        10.425759
ForeignNational          2.929449  2.707564             4.276605         7.375289
AccountActivemonths     13.198093 11.221023            16.938653        24.257648
CurrentResidenceDuration 11.633390  7.082150            12.957104        13.728813
Age                     39.194099 30.871976            46.128360        61.671501
```

Fig19: Output for Random Forest.

- This Random Forest uses the default number of trees i.e. 500 and the number of predictor variable selected for each split is 3 (mtry).
- The classification error has also improved from the decision tree to 81.3 % accuracy.
- In the importance section of Fig 19, we can see that as expected and found in the Entropy and Decision Tree models. The two most important variables are the CreditHistory and Age.

- When we try fitting a different model with 1000 trees and mtry as 9, it doesn't not change the accuracy of the previous model by much.
- So, when we fit the model rf.Credit to the test data the accuracy remains at 81.5%.Which is good.

```
########## predict on test ########

tree.pred_rf2 <- predict(rf.Credit, test, type = "class") # note using type = "prob"

tree.pred_rf2

table(tree.pred_rf2, test$CreditStanding)
(59+105)/(59+105+14+23) #81.5%
```

Fig20: Fitting the RandomForest Model to the Test Data.

- When we plot a variable importance plot as Fig 21, we can see that Age and Credit History are by far the best predictors of the model.

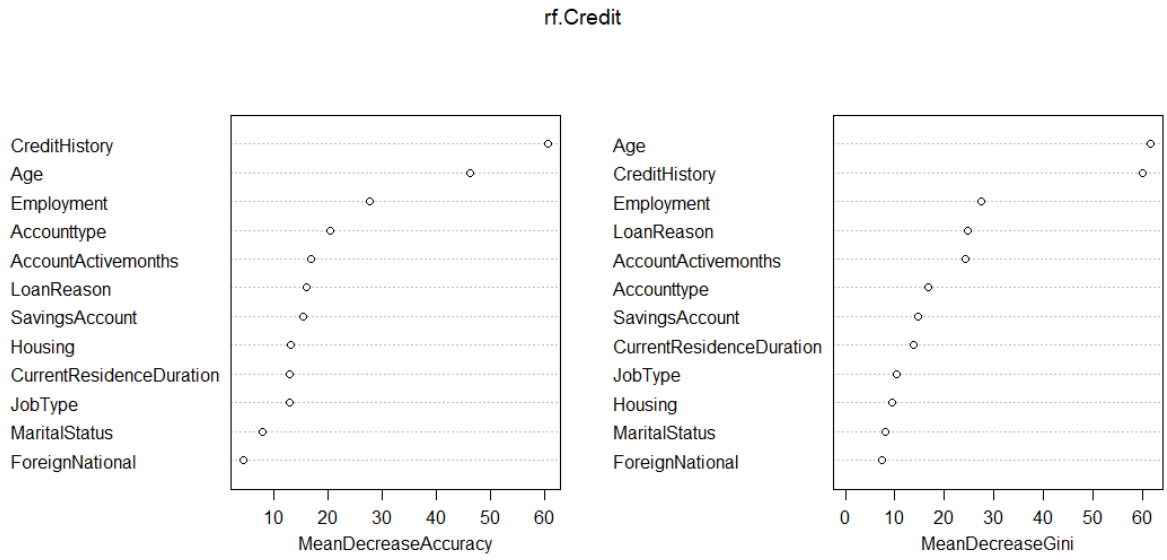


Fig21: Variable importance plot for randomforest model rf.credit

- So, we can safely Say that, out of the Entropy Model, Decision Tree and Random Forest.The Random Forest is a superior Model with 81.5% accuracy in predicting CreditHistory (Bad/Good) (These addresses Question I, refer code between lines 930 and 968)

.

Adherence to GDPR

- Due to GDPR, we were also asked to do the modelling by excluding the sensitive columns like Age, Maritalstatus and ForeignNational.
- The Decision Tree Model after removing the above columns is as below.

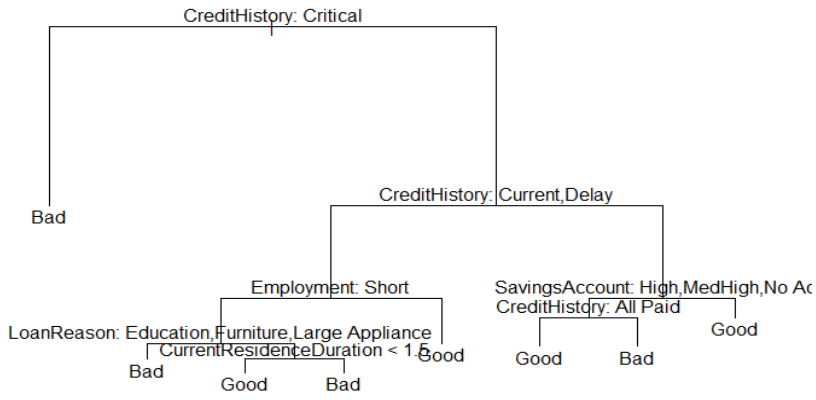


Fig22: Decision Tree graph after removing Age, Maritalstatus and ForeignNational

- We can see that the root node split remains the same, but the split on the right branch has changed from Age<26.5 to CreditHistory (Current,Delay).
- After we predict the fitted model on the test data. The accuracy is around 78.6%. It does not change much with respect to the previous decision tree.
- Hence, The exclusion of columns doesn't affect the accuracy of the decision tree by much.
- Wherease,when we fit the random forest after excluding the columns . We see that the accuracy for the test set reduced to 77.6% which was 81.5% initially.
- The variable importance plot for the randomforest model post GDPR exclusion is as below.
- From the below plot we can say that the top 2 important variables for the model are CreditHistory and Employment. (This address Question J, refer code between lines 970 and 1008)

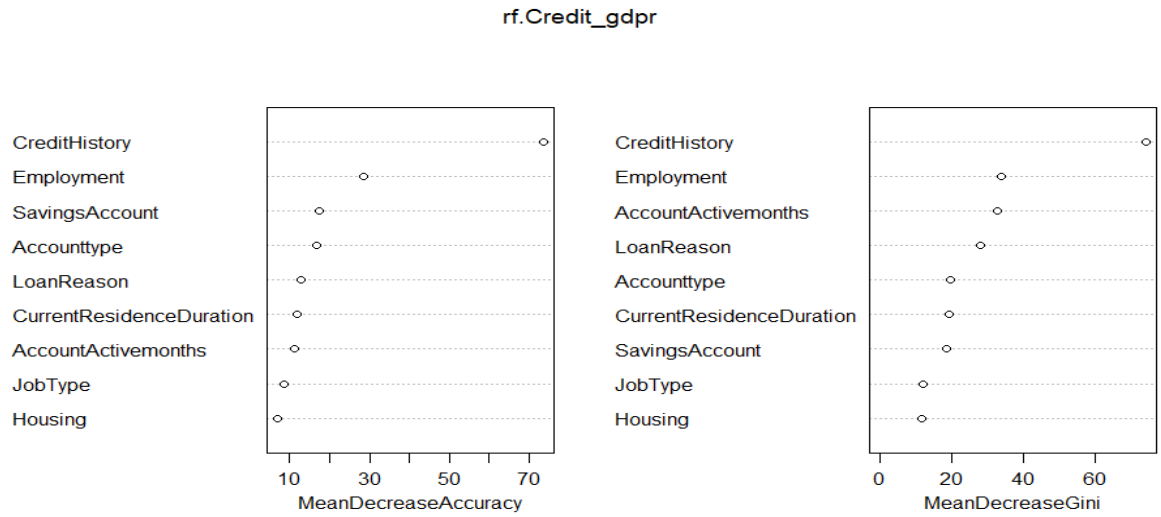


Fig23: Variableimportance plot for RandomForest post GDPR.

Anomaly Detection

- It was told to us by the manager that Maeve's company uses a process that is a mixture of a grading system and human input to grade each past loan as good or bad. Maeve is suspicious that during a particular time that this process performed very poorly and produced inaccurate results.
- So, the idea was to do a timeseries plot and check for anomalous trend.
- But challenge was we did not have a date variable.
- So, I did the following steps to create the time series plot
    i. At first, I converted the ID column to date with inception '2022-01-01', so that I get a continuous sequence of dates.
    ```
    CreditData$Dates <- as.Date('2022-01-01') + CreditData$ID
    ```
    ii. To catch the trend visible to naked eyes. I made a new column called creditstandingflag and replaced Good Loans with 1 and Bad loans with -1
    ```
    CreditData$creditstandingflag[CreditData$CreditStanding=="Good"]=1
    CreditData$creditstandingflag[CreditData$CreditStanding=="Bad"]=-1
    ```

iii. Next, I made a column storing the cumulative value of the variable CreditData$creditstandingflag,to catch hold of any trend that was anomalous.

CreditData[,"cum_flag"] <- cumsum(CreditData$creditstandingflag)

iv. Then I plot the below time series chart.

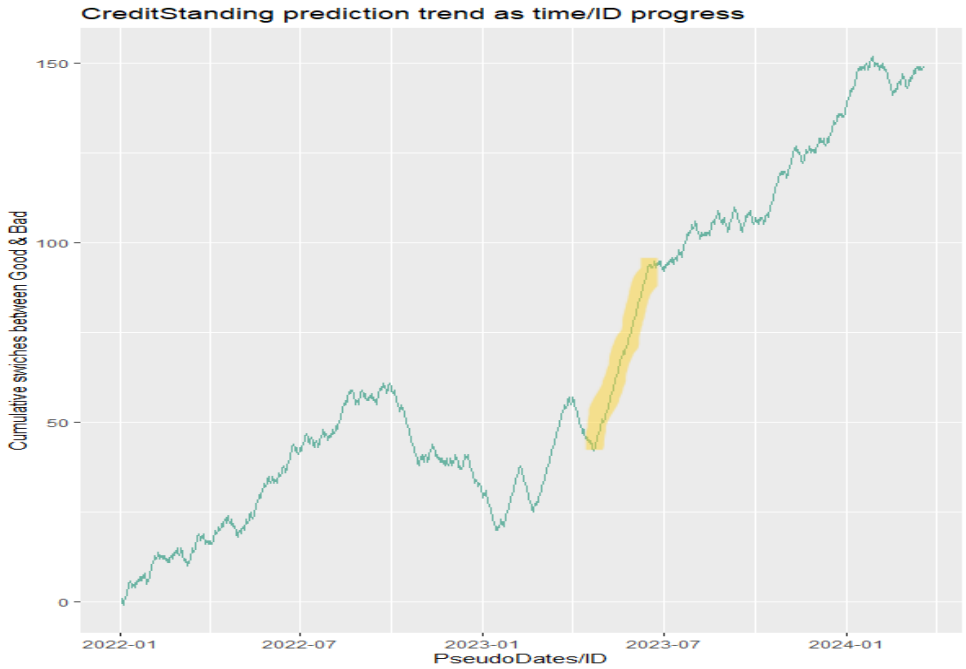


Fig24: Timeseries plot for anomaly detection.

v. From the highlighted region we can see that there was a period in between where abnormal number of good loans were approved at a stretch.

vi. We can see that from ID 477 till 533 we have large influx of good loans i.e., for those 56 records we have 54 Good loans and only 2 bad loans. This looks like something suspicious.

vii. This looks like a case of insider corruption or inefficient system, one case can be an insider person was bribed or forced to approve the loans during that period.

viii. It strongly suggests a case of Human involvement and Bias/ or ineffeciency. (This address Question K, refer code between lines 1009 and 1045)

CONCLUSION

From The above analysis we can come to the conclusion that.

1. Out of all the models the Random Forest was the most superior model with an accuracy of 81.5% for the complete model.
2. The two variables which has the highest predictive power are Credit History (@ Critical) and Age (@ 26).
3. GDPR exclusion of certain columns does bring down the accuracy by a little. But the impact is not substantial.
4. We found an anomalous set of records through time series plot. This can be looked upon and investigated further by the company.

REFERENCES

1. Labs and theory of Data Science and Analytics subject module.
2. ISLR book.