Machine learning model solution

- CLUSTERING, CART & RANDOM FOREST

Objectives

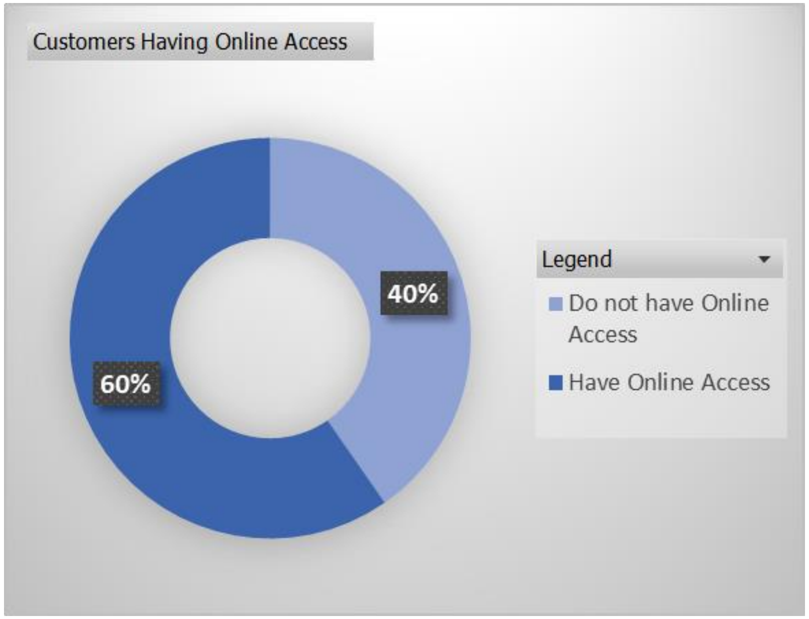
- Business problem: To identify amongst the existing customers of a bank who are more likely to take up personal loan , for effective cross-selling
- Modeling objectives:
  - Clustering : Create meaningful clusters from the population, to understand the nature of the customers
  - Classification Tree model: Find rules to classify the customers for cross-selling, techniques to be used
    - CART
    - Random forest

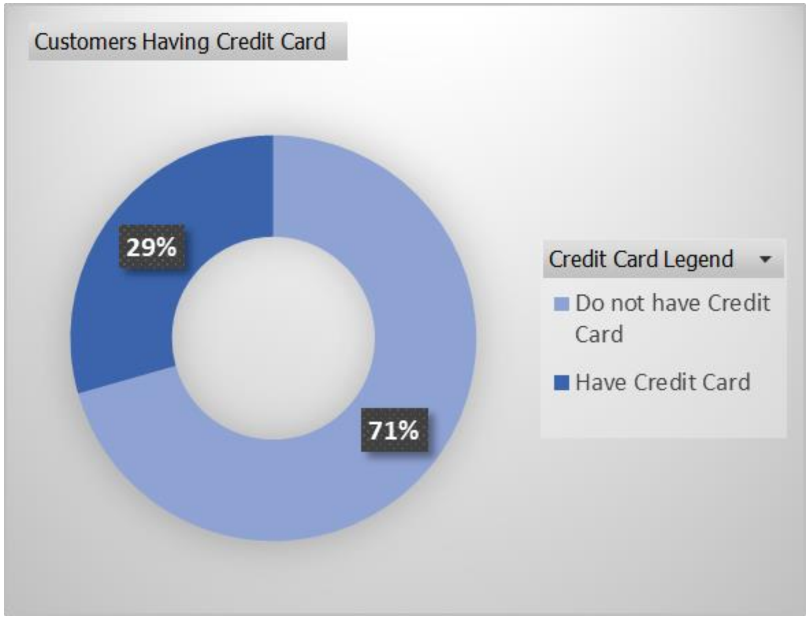
greatlearning

Exploratory data analysis

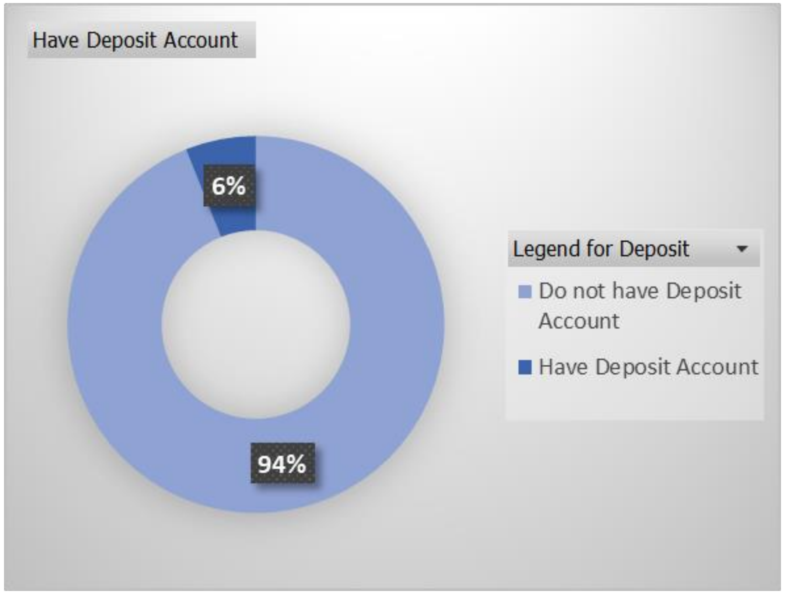
- Dataset provided has the following characteristics:
  - No. of records: 5000
  - No. of predictor variables : 13
    - 1 customer identifier column, which do not hold any statistical significance
    - 6 continuous numeric variables
    - 4 ordinal variables with 2 levels each
    - 2 ordinal variables with multiple levels
  - Target variable : with 2 distinct classes , 0 for non-responders and 1 for no response
- 480 of 5000 customers responded positively, i.e. only 9.6% success rate

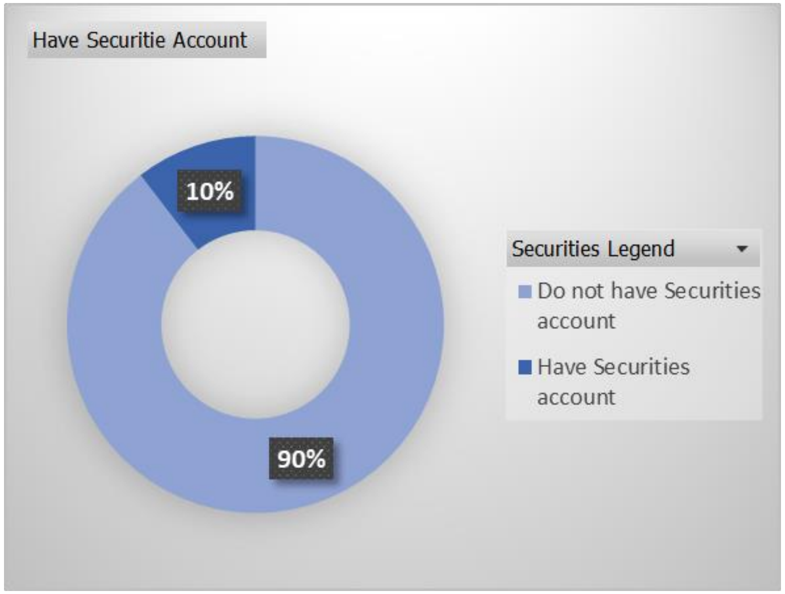
Data analysis : univariate – ordinal variables



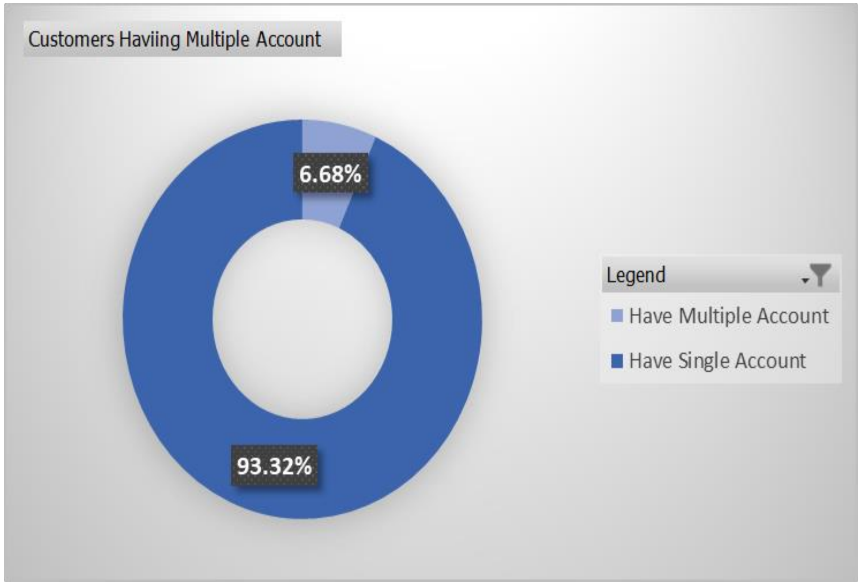


Data analysis : univariate – ordinal variables



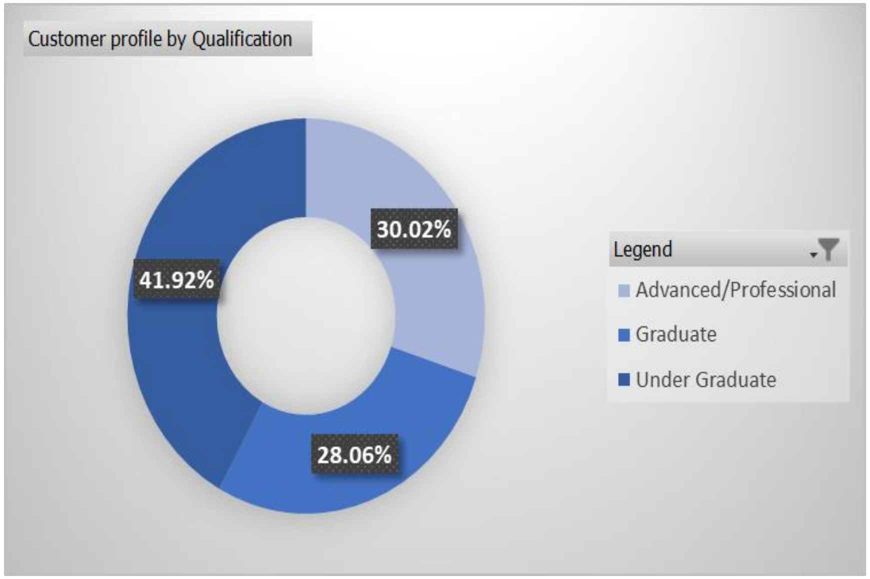


Data analysis : univariate – ordinal variables



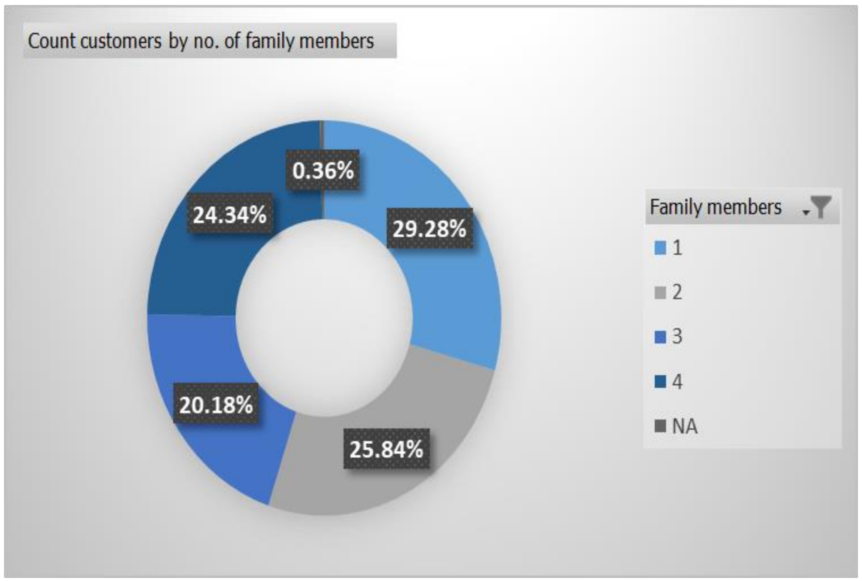
- Majority of the customers currently have only single relationship with the bank
- Securities and deposit accounts are not very popular with the customers
- Credit Card is the most popular product amongst all
- Customers are open to internet banking facilities

Data analysis : univariate – ordinal variables



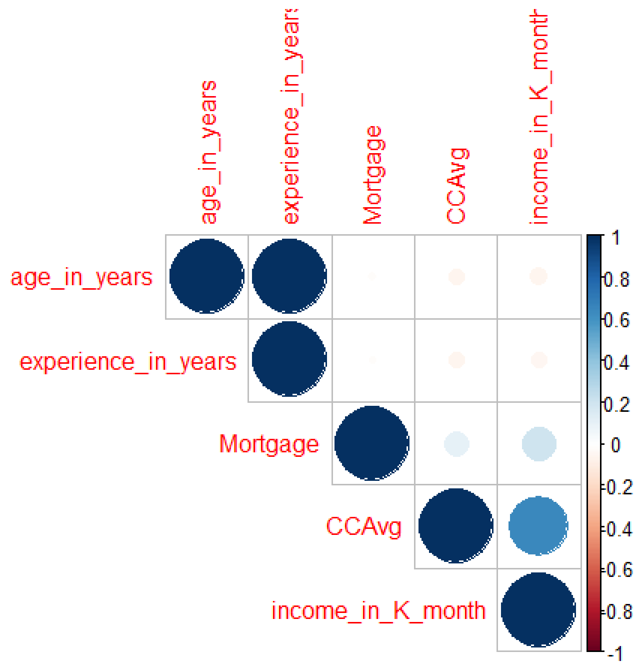
- Distribution of customers by their educational qualification is mostly uniform

Data analysis : univariate – ordinal variables



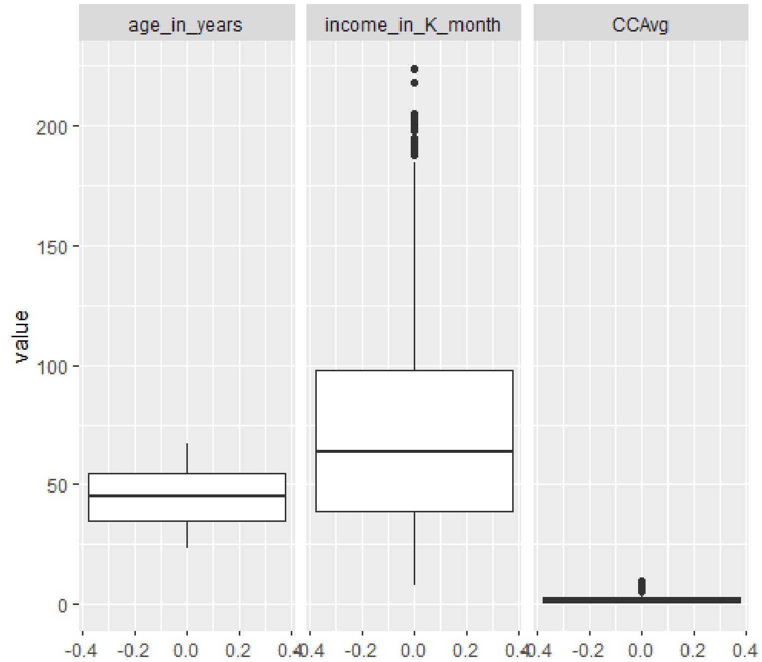
- Distribution of customers by number of family members they have
- Since only 0.36% percent of records have NA will exclude these records from further analysis

Data analysis : multivariate – continuous variables



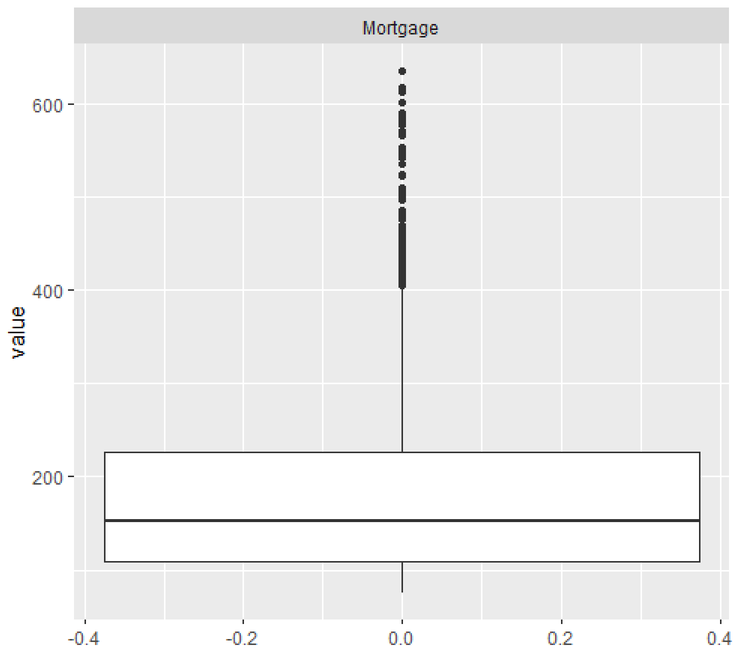
- There is high degree of correlation between customer age and years of professional experience, which is as per expectation
- Will exclude experience years from next steps f analysis
- People with high income tend to spend more on credit card per month , which is also as per expectation, but correlation is just about 0.4 so will keep both these variables

Data analysis : univariate – continuous variables



- Distribution of Age is normal
- Income and Credit card average spending per month have outliers
- These variables need to be scaled

Data analysis : univariate – continuous variables



- Check on records where home mortgage value is greater than 0 , shows that there are quite a number of outliers

Data analysis : Treatment of zip code

- There are unique 467 zipcodes in the dataset
- Since Zipcode as a number do not have any significance will treat it as a factor
- To reduce the cardinality of Zipcode will aggregate it to State level
- Since majority of the zipcodes are of 5 digits assuming them to be from US , hence using “zipcode” package to map them to States
- 33 records are found to have invalid Zipcode, ignoring them , found only one record belonging to state AE and rest 4966 to CA

	0	1
AE	1	0
CA	4489	477

- So we can safely assume that S te to target variable hence dropping zipcode column

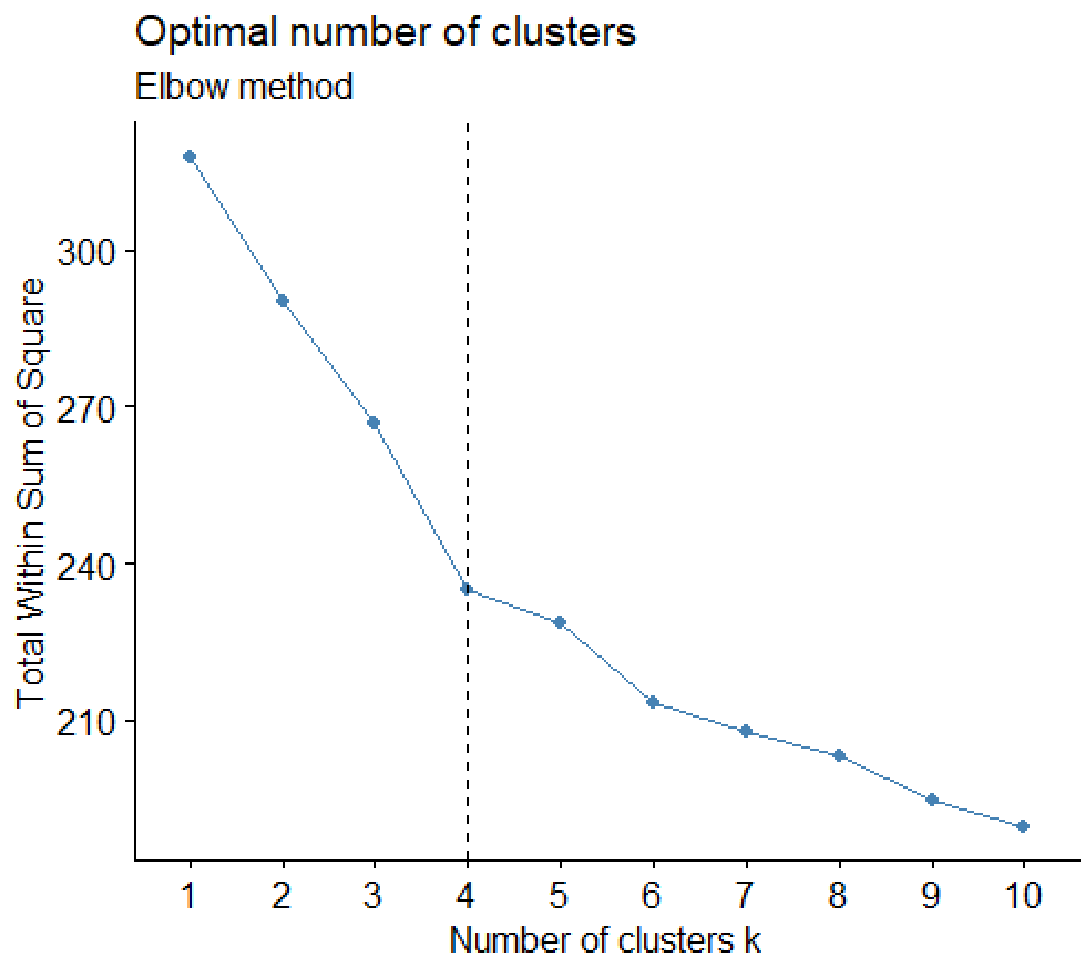
Data transformations done

- Row filters :
  - Records with family member count not specified (18 of 5000)
- Column Filter :
  - ID
  - Zipcode
  - Professional experience years
- Columns scaled for clustering:
  - Age
  - Income
  - CC Average spent
  - House Mortgage amount
- Columns created:
  - Have multiple relationships with bank (1 : Yes , 0:No)
  - Have House Mortgage (1:Yes , 0 :No)

CLUSTERING

Algorithm choice rationale

- Since the number of records are running into thousands we should be avoiding hierarchical clustering methods
- We have both continuous and factor data with more than one levels available within independent variables, hence Euclidean distance function cannot be used
- Gower distance function will be used, which takes care of both the data types
- K-means algorithms cannot be used since it assumed Euclidean distance , hence PAM (partitioning around medoids) will be used which is a variant of k-means , but instead of always defaulting to Euclidean distance it can refer to custom distance function
- We need to provide the number of clusters to start with as input to this algorithm


Optimal number of clusters
Elbow method
Total Within Sum of Square
300
270
240
210
1
2
3
4
5
6
7
8
9
10
Number of clusters k

profiling clusters – Cluster 1

```
[[1]]
  age_in_years    income_in_K_month     CCAvg           Mortgage        family_size_fact Education_fact have_securities_acct_fact
 Min.   :23.00   Min.   :  8.00     Min.   :0.000   Min.   :  0.00   1:107            1:165          0:1164                   
 1st Qu.:33.00   1st Qu.: 33.00     1st Qu.:0.600   1st Qu.:  0.00   2:131            2:909          1: 137                   
 Median :40.00   Median : 54.00     Median :1.300   Median :  0.00   3:293            3:227                                   
 Mean   :42.44   Mean   : 61.68     Mean   :1.551   Mean   : 27.25   4:770                                                    
 3rd Qu.:52.00   3rd Qu.: 80.00     3rd Qu.:2.100   3rd Qu.:  0.00                                                            
 Max.   :67.00   Max.   :194.00     Max.   :8.300   Max.   :612.00                                                            
 have_deposit_account_fact have_online_access_fact have_cc_fact have_multiple_relns_fact have_mortgage_fact
 0:1213                    0: 284                  0:941        0:1213                   0:1093            
 1:  88                    1:1017                  1:360        1:  88                   1: 208            
```

- Characterized by
  - Majority are Graduates
  - Majority have online access
  - Majority have 4 family members
  - Majority do not have mortgage account

profiling clusters – Cluster 2

```
[[2]]
  age_in_years   income_in_K_month     CCAvg            Mortgage      family_size_fact Education_fact have_securities_acct_fact
 Min.   :23.00   Min.   :  8.00    Min.   : 0.000   Min.   :  0.00   1:170            1:272          0:1110
 1st Qu.:38.00   1st Qu.: 34.00    1st Qu.: 0.600   1st Qu.:  0.00   2:602            2:212          1: 134
 Median :49.00   Median : 58.00    Median : 1.500   Median :  0.00   3:263            3:760
 Mean   :47.36   Mean   : 66.19    Mean   : 1.726   Mean   : 25.26   4:209
 3rd Qu.:57.00   3rd Qu.: 85.00    3rd Qu.: 2.200   3rd Qu.:  0.00
 Max.   :67.00   Max.   :203.00    Max.   :10.000   Max.   :581.00
 have_deposit_account_fact have_online_access_fact have_CC_fact have_multiple_relns_fact have_mortgage_fact
 0:1224                    0:1059                  0:869        0:1195                   0:1055
 1:  20                    1: 185                  1:375        1:  49                   1: 189
```

- Characterized by
  - Majority have advanced or professional degree
  - Majority do not have internet banking access
  - Majority have 2 family members
  - Majority do not have any house mortgage

profiling clusters – Cluster 3

```
[[3]]
  age_in_years   income_in_K_month      CCAvg           Mortgage family_size_fact Education_fact have_securities_acct_fact
 Min.   :23.00   Min.   :  8.00    Min.   : 0.000   Min.   :0   1:757            1:1008         0:1166
 1st Qu.:37.00   1st Qu.: 49.75    1st Qu.: 1.000   1st Qu.:0   2:269            2:  78         1: 134
 Median :45.00   Median : 84.00    Median : 2.000   Median :0   3:205            3: 214
 Mean   :45.71   Mean   : 90.79    Mean   : 2.471   Mean   :0   4: 69
 3rd Qu.:55.00   3rd Qu.:128.00    3rd Qu.: 3.300   3rd Qu.:0
 Max.   :67.00   Max.   :224.00    Max.   :10.000   Max.   :0
 have_deposit_account_fact have_online_access_fact have_CC_fact have_multiple_relns_fact have_mortgage_fact
 0:1197                    0: 284                  0:903        0:1193                   0:1300
 1: 103                    1:1016                  1:397        1: 107                   1:   0
```

- Characterized by
  - Majority are undergraduates
  - Majority have online access
  - Majority have credit card accounts
  - Majority have family size of 1
  - None of them have any mortgage account

profiling clusters – Cluster 4

```
[[4]]
  age_in_years   income_in_K_month     CCAvg          Mortgage    family_size_fact Education_fact have_securities_acct_fact
 Min.   :23.00   Min.   :  8.00    Min.   :0.000   Min.   : 75   1:430            1:643          0:1023
 1st Qu.:36.00   1st Qu.: 38.00    1st Qu.:0.700   1st Qu.:113   2:290            2:200          1: 114
 Median :46.00   Median : 64.00    Median :1.500   Median :158   3:248            3:294
 Mean   :45.97   Mean   : 76.28    Mean   :2.012   Mean   :189   4:169
 3rd Qu.:56.00   3rd Qu.:104.00    3rd Qu.:2.700   3rd Qu.:234
 Max.   :66.00   Max.   :205.00    Max.   :9.000   Max.   :635
 have_deposit_account_fact have_online_access_fact have_CC_fact have_multiple_relns_fact have_mortgage_fact
 0:1048                    0:386                   0:804        0:1049                   0:   0
 1:  89                    1:751                   1:333        1:  88                   1:1137
```

- Characterized by
  - Majority are undergraduates
  - All of them have mortgage account
  - Majority have internet banking access

60
30
0
-30
-60
Y
-30
0
30
60
X
cluster
1
2
3
4

CLASSIFICATION

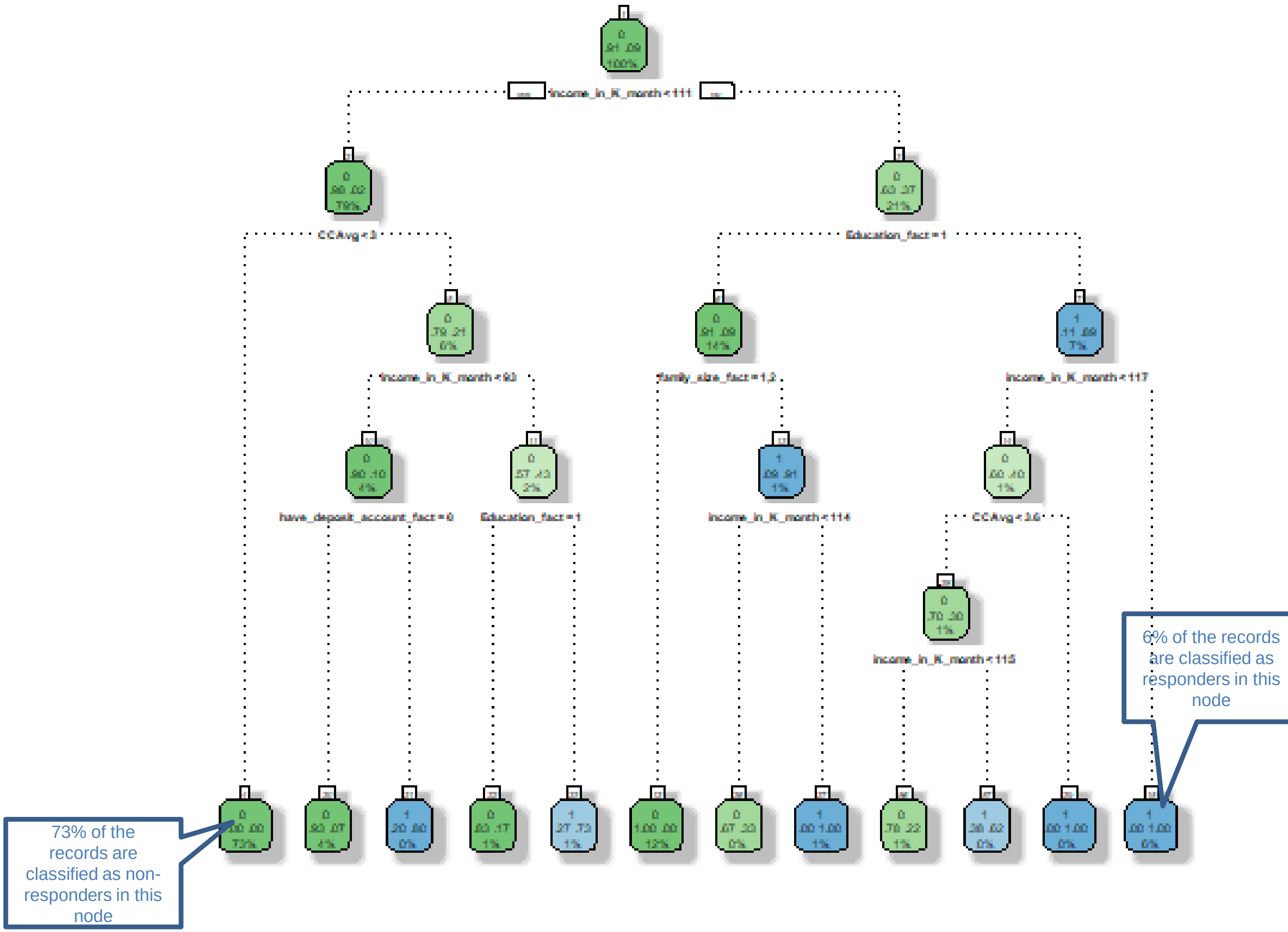
Test and train split

```
## 70% of the sample size
sample_size <- floor(0.7 * nrow(cust_base_orig_fact))

## set the seed to make the partition reproductible
set.seed(4)
train_ind <- sample(seq_len(nrow(cust_base_orig_fact)), size = sample_size)

train_dataset <- cust_base_orig_fact[train_ind, ]
test_dataset <- cust_base_orig_fact[-train_ind, ]

##Check for uniformity of data in test and train
> sum(as.integer(as.character(train_dataset$did_accept_personal_loan_offer_fact))) / nrow(train_dataset)
[1] 0.08976197
> sum(as.integer(as.character(test_dataset$did_accept_personal_loan_offer_fact))) / nrow(test_dataset)
[1] 0.1103679
> sum(as.integer(as.character(cust_base_orig_fact$did_accept_personal_loan_offer_fact))) / nrow(cust_base_orig_fact)
[1] 0.0959454
```

– Data is split into 70:30 ratio for train to test

– Response rate in split and overall data are close

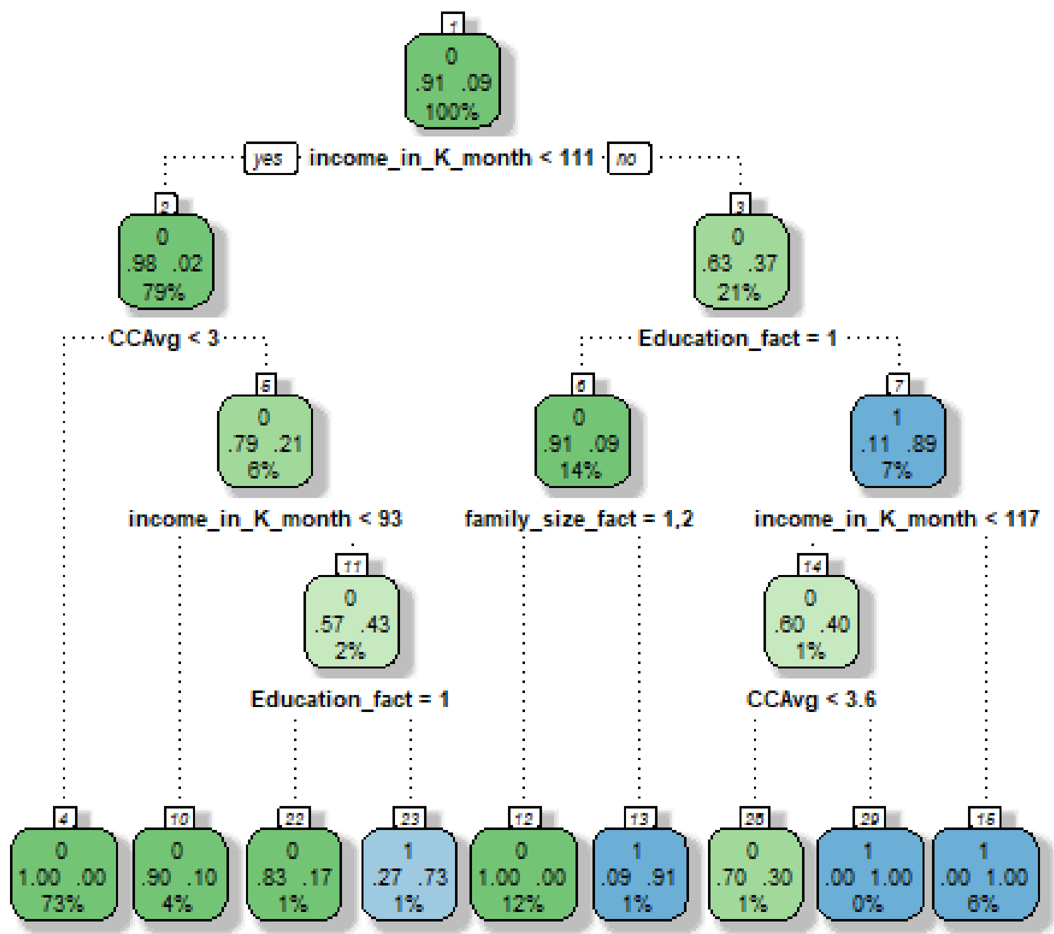
CART


6% of the records are classified as responders in this node
73% of the records are classified as non-responders in this node

```
Variables actually used in tree construction:
[1] CCAvg                    Education_fact             family_size_fact          have_deposit_account_fact
[5] income_in_K_month

Root node error: 313/3487 = 0.089762

n= 3487

         CP nsplit rel error  xerror     xstd
1 0.3210863      0   1.00000 1.00000 0.053927
2 0.1214058      2   0.35783 0.38339 0.034391
3 0.0287540      3   0.23642 0.23962 0.027369
4 0.0223642      4   0.20767 0.23003 0.026828
5 0.0149095      5   0.18530 0.21406 0.025899
6 0.0095847      8   0.14058 0.16613 0.022866
7 0.0063898      9   0.13099 0.17891 0.023716
8 0.0000000     11   0.11821 0.20128 0.025129
```

```
# Pruned tree

ptree<- prune(tree_iter1, cp= 0.0095847 ,"CP")
```

Pruned Classification Tree



Model performance measures

- Deciling and rank-ordering the deciles give us the following results
  - More than 90% of the records are accounted for in 10^{th} and 9^{th} deciles
  - KS statistics is 0.93 (benchmark: >0.45) in the first decile itself which indicates overfit
  - Lift calculated is also 9 in the first decile
- Checking on other performance measures:
  - AUC : 0.995953 (benchmark: >0.80)
  - Gini : 0.9028706 (benchmark : >0.60)

	deciles	cnt	cnt_resp	cnt_non_resp	lift
1	10	368	299	69	9.05
2	9	2687	14	2673	1.14
3	2	432	0	432	1.00

Validate against holdout test data

- Running the tree for test dataset created from the initial split gives the following results
  - More than 90% of the records are accounted for in 10th and 9th deciles
  - KS statistics is 0.92 (for train: 0.93) in the first decile itself
  - Lift calculated is also 8.52 in the first decile
- Checking on other performance measures:
  - AUC : 0.9843769 (for train: 0.995953)
  - Gini : 0.8873045 (for train : 0.9028706)
- We see above that the performance measures in train and test are comparable , with not much deviation, hence we can conclude that this is a **good model.**

	deciles	cnt	cnt_resp	cnt_non_resp	lift
1	10	152	143	9	8.52
2	9	1176	22	1154	1.13
3	2	167	0	167	1.00

RANDOM FOREST

greatlearning

Initial model run

Started with these parameters given the guidelines

```
tuned_rf <- tuneRF(x = train_dataset[,-13],
                   y=train_dataset$did_accept_personal_loan_offer_fact,
                   mtryStart = 3,
                   ntreeTry=200,
                   stepFactor = 1.5,
                   improve = 0.001,
                   trace=T,
                   plot = T,
                   doBest = TRUE,
                   nodesize = 15,
                   importance=T
)
```

Suggests the out of bag error rate is lowest for mtry=12 so will run the model with that

```
mtry = 3  OOB error = 1.49%
Searching left ...
mtry = 2        OOB error = 1.84%
-0.2307692 0.001
Searching right ...
mtry = 4        OOB error = 1.35%
0.09615385 0.001
mtry = 6        OOB error = 1.26%
0.06382979 0.001
mtry = 9        OOB error = 1.2%
0.04545455 0.001
mtry = 12       OOB error = 1.12%
0.07142857 0.001
```

final model run

Through multiple iterations we see that these parameters give consistent output for multiple runs even on slight variations of the parameters

```
tuned_rf <- tuneRF(x = train_dataset[,-13],
                   y=train_dataset$did_accept_personal_loan_offer_fact,
                   mtryStart = 8,
                   ntreeTry=500,
                   stepFactor = 1.5,
                   improve = 0.001,
                   trace=T,
                   plot = T,
                   doBest = TRUE,
                   nodesize = 200,
                   importance=T
```

Suggests the out of bag error rate is lowest for mtry=8 so will stick to this

```
mtry = 8  OOB error = 1.81%
Searching left ...
mtry = 6        OOB error = 1.86%
-0.03174603 0.001
Searching right ...
mtry = 12       OOB error = 2.15%
-0.1904762 0.001
```

Checking for the importance of variables

	0	1	MeanDecreaseAccuracy	MeanDecreaseGini
Education_fact	45.45	40.05	46.17	145.67
income_in_K_month	33.95	34.27	34.27	123.52
family_size_fact	17.53	15.18	17.61	37.73
CCAvg	12.55	12.42	12.59	31.87
have_deposit_account_fact	5.54	1.39	4.51	10.07
Mortgage	2.13	0.30	2.10	0.49
have_multiple_relns_fact	2.92	-1.73	1.61	1.44
have_cc_fact	1.39	1.42	1.41	0.08
have_securities_acct_fact	-1.00	1.00	1.00	0.01
age_in_years	-1.00	1.00	0.00	0.02
have_online_access_fact	0.00	0.00	0.00	0.00
have_mortgage_fact	0.00	0.00	0.00	0.00

- We see that the 2 new variables added as part of feature engineering process do not have any significance
- Top 4 most important variables helping in reducing impurity in each node is same as in CART model

Model performance measures

- Deciling and rank-ordering the deciles give us the following results
  - More than 100% of the records are accounted for in top 3 deciles
  - KS statistics is 0.94 (benchmark: >0.45) in the first decile itself which indicates overfit
  - Lift calculated is also 9 in the first decile
- Checking on other performance measures:
  - AUC : 0.995788 (benchmark: >0.80)
  - Gini : 0.9005289 (benchmark : >0.60)

	deciles	cnt	cnt_resp	cnt_non_resp	lift
1	10	364	296	68	9.06
2	9	355	16	339	4.83
3	8	2768	1	2767	1.00

Validate against holdout test data

- Running the tree for test dataset created from the initial split gives the following results
  - All 100% of the records are accounted for in top 3 deciles
  - KS statistics is 0.92 (for train: 0.93) in the first decile itself
  - Lift calculated is also 8.42 in the first decile
- Checking on other performance measures:
  - AUC : 0.9809547 (for train: 0.995788)
  - Gini : 0.8822665 (for train : 0.9005289)
- We see above that the performance measures in train and test are comparable , with not much deviation, hence we can conclude that this is a **good model.**

	deciles	cnt	cnt_resp	cnt_non_resp	lift
1	10	150	139	11	8.40
2	9	164	22	142	4.65
3	8	1181	4	1177	1.00

conclusion

- CART and Random Forest have given the similar kind of variable importance output, so following variables play major role in identifying customers who are more likely to respond:
    - Count of Family Members
    - Education Level
    - Income per month
- For both the model performance measurements give us high accuracy indicating overfit, but the model when validated against holdout sample gives comparable measurements
- Imbalanced nature of the data might lead to these observations, though decision trees are theoretically more immune to imbalanced data
- We need to take this model with a pinch of salt, and more data will be good
- Following approaches can be taken to make the model more reliable:
    - Get more data from business
    - Run the models after synthetically correcting for imbalance

THANK YOU