INTRODUCTION TO MACHINE LEARNING (NPFL054)
Homework #2

Name: Ayaz Dyshin

Date:

Overview:

Used packages: ISLR, rpart,ROCR, randomForest, glmnet, writexl.

Target dataset: Caravan

Target attribute: Purchase – a binary attribute indicating if a customer has purchased a Caravan insurance policy.

Number of attributes (including target attribute): 86

Dataset size (number of rows): 5822

Description of attributes:

1-43: Contains general information about customers ie: Social class, income, education, age, religion etc.

44-85: Contains information about customer’s contribution to different policies and number of other types of insurance or policy.

86: target attribute Purchase.

More on the feature description:
https://ufal.mff.cuni.cz/~holub/2021/docs/caravan.attributes.pdf

Description of tuning process: initial data set will be split into a train set of size 4822 and test set of size 1000. Estimation of train set will be done using cross-validation. There methods will be used and compared to decide which one fits better: Decision Tree, Random Forest, Regularized Logistic Regression. Parameters to tune (respectively) complexity parameter cp, number of trees (ntree) and feature sample size (mtry), regularization parameter lambda, elasticity parameter alpha.

After the best method with optimized parameters is found, it will then be retrained on the whole (5822 rows) data set and used to predict with 1000 values of blind test dataset.

Task 1 – Data analysis

Pre question:

If we select 100 random entries form the dataset, within those 100 entries on average we will have 6 customers that purchased the insurance and 94 customers that didn't purchase the insurance. Therefore, the precision will be:

Precision = 6/(6+94) = **0.06**

a)

Now we will analyze customer type **MOSHOOFD**, which is **Customer main type**. It has 10 different levels. Here is a table with each level's name, number of customers in each level and percentage of customers that have purchased the insurance from each level:

Table for **MOSHOOFD**:

	Group name	Size	Purc. Freq.
1	Successful hedonists	552	8.7
2	Driven Growers	502	13.1
3	Average Family	886	6.66
4	Career Loners	52	0
5	Living well	569	2.64
6	Cruising Seniors	205	1.95
7	Retired and Religeous	550	3.64
8	Family with grown ups	1563	5.69
9	Conservative families	667	6.3
10	Farmers	276	1.81

From this table we can see that the highest percentage of purchases is in the **Driven growers** group.

Runner up for the highest percentage of purchases is the **Successful hedonists** group.

Career loners group has the smallest percentage of purchases which is in fact 0.

Now we will analyze **MOSTYPE** which is **customer subtype** and it has 41 different levels.

Here is a table with each level's name, number of customers in each level and percentage of customers that have purchased the insurance from each level:

Table for **MOSTYPE**:

	Group name	Size	Purc. Freq.
1	High Income	124	10.5
2	Very Important Provincials	82	7.32
3	High status seniors	249	10
4	Affluent senior apartments	52	3.85
5	Mixed seniors	45	4.44
6	Career and childcare	119	10
7	Dinki's (double income no kids)	44	6.82
8	Middle class families	339	15
9	Modern	278	4.32
10	Stable family	165	5.45
11	Family starters	153	5.88
12	Affluent young families	111	14.4
13	Young all american family	179	7.26
14	Junior cosmopolitan	0	0
15	Senior cosmopolitans	5	0
16	Students in apartments	16	0
17	Fresh masters in the city	9	0
18	Single youth	19	0
19	Suburban youth	3	0
20	Etnically diverse	25	8
21	Young urban have-nots	15	0
22	Mixed apartment dwellers	98	4.08
23	Young and rising	251	1.59
24	Young	180	2.78
25	Young seniors in the city	82	2.44
26	Own home elderly	48	2.08
27	Seniors in apartments	50	2
28	Residential elderly	25	0
29	Porchless seniors: no front yard	86	2.33
30	Religious elderly singles	118	3.39
31	Low income catholics	205	2.93
31	Mixed seniors	141	5.67
33	Lower class large families	810	5.68
34	Large family	182	4.95
35	Village families	214	3.74
36	Couples with teens 'Married with children	225	7.11
37	Mixed small town dwellers	132	7.58
38	Traditional families	339	6.78
39	Large religous families	328	5.79
40	Large family farms	71	0
41	Mixed rurals	205	2.4

From this table we see that groups: **Senior cosmopolitans, Students in apartments, Fresh masters in the city, Single youth, Suburban youth, Young urban have-nots, Residential elderly, Large family farms** have no people who purchased the insurance.

While group **Junior Cosmpolitan** doesn‘t have any people.

Groups with biggest percentage of purchases is:is **middle class families** and **Affluent young families**.

b)

Now we will analyze the intersection of these two level. In the following table we can we can see intersection of these two level.

How to interpret the table: Columns are groups of **Customer main** type and rows are groups of customer subtype. Ie if we take cell with coordinates 1 1, that means that we have 124 customers with main type 1 (Successful hedonists) and subtype 1 (High income).

L0 vs L2	1	2	3	4	5	6	7	8	9	10
1	124	0	0	0	0	0	0	0	0	0
2	82	0	0	0	0	0	0	0	0	0
3	249	0	0	0	0	0	0	0	0	0
4	52	0	0	0	0	0	0	0	0	0
5	45	0	0	0	0	0	0	0	0	0
6	0	119	0	0	0	0	0	0	0	0
7	0	44	0	0	0	0	0	0	0	0
8	0	339	0	0	0	0	0	0	0	0
9	0	0	278	0	0	0	0	0	0	0
10	0	0	165	0	0	0	0	0	0	0
11	0	0	153	0	0	0	0	0	0	0
12	0	0	111	0	0	0	0	0	0	0
13	0	0	179	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	5	0	0	0	0	0	0
16	0	0	0	16	0	0	0	0	0	0
17	0	0	0	9	0	0	0	0	0	0
18	0	0	0	19	0	0	0	0	0	0
19	0	0	0	3	0	0	0	0	0	0
20	0	0	0	0	25	0	0	0	0	0
21	0	0	0	0	15	0	0	0	0	0
22	0	0	0	0	98	0	0	0	0	0
23	0	0	0	0	251	0	0	0	0	0
24	0	0	0	0	180	0	0	0	0	0
25	0	0	0	0	0	82	0	0	0	0
26	0	0	0	0	0	48	0	0	0	0
27	0	0	0	0	0	50	0	0	0	0
28	0	0	0	0	0	25	0	0	0	0
29	0	0	0	0	0	0	86	0	0	0
30	0	0	0	0	0	0	118	0	0	0
31	0	0	0	0	0	0	205	0	0	0
32	0	0	0	0	0	0	141	0	0	0
33	0	0	0	0	0	0	0	810	0	0
34	0	0	0	0	0	0	0	182	0	0
35	0	0	0	0	0	0	0	214	0	0
36	0	0	0	0	0	0	0	225	0	0
37	0	0	0	0	0	0	0	132	0	0
38	0	0	0	0	0	0	0	0	339	0
39	0	0	0	0	0	0	0	0	328	0
40	0	0	0	0	0	0	0	0	0	71
41	0	0	0	0	0	0	0	0	0	205

.

Task 2 – Model fitting, optimization, and selection

2a)

Now we will start tuning our models. But before that, we need to prepare our data.

As our data set has a size of 5822, we will split it as 4822 for train set and 1000 for test set.

As we are going to do a 10-fold cross-validation we need to split our train data in 10 parts for evaluation of each model.

As data is unbalanced, we cannot simply split the train set randomly in 10 parts. To make sure that each fold has approximately the same ratio of positives to negatives, we will split the whole train set in two parts: only positives and only negatives. Then we will split both of these parts (randomly) in 10 equal parts, and randomly combine them. In this way we can make sure that we have equally balanced 10 folds.

In the following section we will be testing several parameters for 3 different methods, but some things we will do will be in common for all of them, such as:

For each value of our parameter in specified range, we run 10-fold cross-validation and report it's: Mean AUC 0.2, standard deviation and confidence interval for the mean.

We measure mean AUC 0.2 (meaning that we optimize it up to FRP <= 20%) the reason for that is because precision drops the higher FPR is.

After we obtain the results of each parameter for a specific method, we pick the one that gives us the highest AUC 0.2. We maximize AUC 0.2 because of the fact that data is unbalanced. Out of 5822 the number of rows where we have Purchase = "Yes" is 348, while rows where we have Purchase = "No" is 5474. Giving us the ratio around 1:16 (Yes to No). This means that we need to adjust our models so that they are good at predicting true positives correctly. This is why we maximize AUC because ROC-curve is plotted as: TPR on y-axis and FPR on x-axis, meaning that the bigger the AUC is the bigger the TPR is.

The first method that we are going to tunes is going to be a Decision Tree. Here we will be tuning cp (complexity parameter). Range for cp values that was chosen is from 0.001 up to 0.02 with a step of size 0.001.

Results of cross validation for Decision tree method:

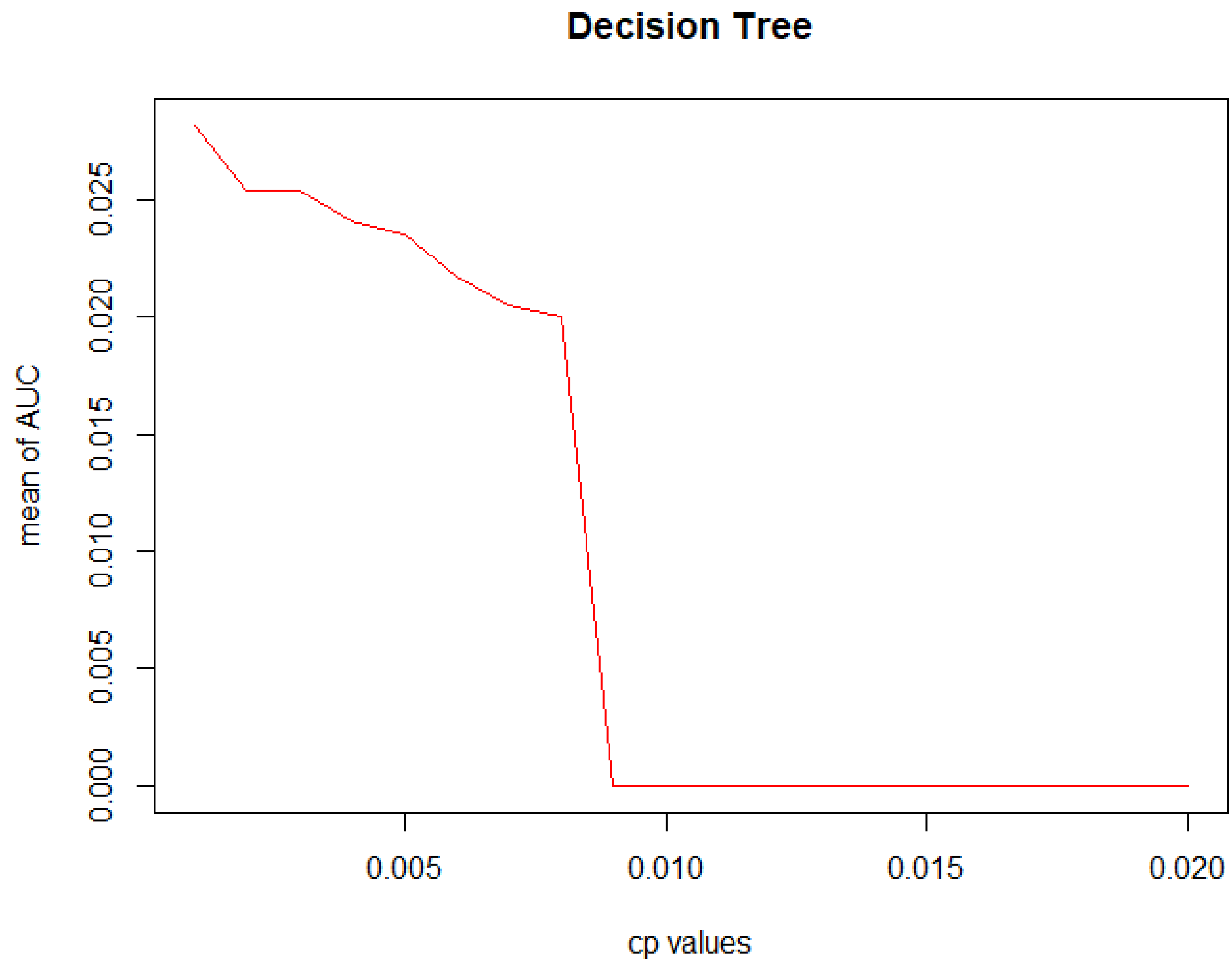


Table of other results:

cp value	mean of AUC	std. div.	CI s	CI e
0,001	0,028	0,008	0,0223	0,0340
0,002	0,025	0,007	0,0202	0,0305
0,003	0,025	0,006	0,0209	0,0299
0,004	0,024	0,005	0,0205	0,0276
0,005	0,023	0,004	0,0205	0,0265
0,006	0,022	0,003	0,0197	0,0237
0,007	0,021	0,002	0,0193	0,0218
0,008	0,02	0	0	0
0,009	0	0	0	0
0,01	0	0	0	0
0,011	0	0	0	0
0,012	0	0	0	0
0,013	0	0	0	0
0,014	0	0	0	0
0,015	0	0	0	0
0,016	0	0	0	0
0,017	0	0	0	0
0,018	0	0	0	0
0,019	0	0	0	0
0,02	0	0	0	0

We pick cp value with the highest AUC 0.2 = 0.028 for our best model which is: 0.001

2b)

Now we will tune the Random forest method. For Random forest we will be tuning two parameters: number of trees (ntree) and feature sample size (mtry). We will start with ntree, pick the best value of ntree, and then with this best value of ntree we will adjust the mtry paramter.

The values for ntree are in the range from 300 to 800 with a step size of 100.

Results of cross validation for Random forest method after changing ntree parameter (plot with mean of AUC 0.2 of y-axis and the value of ntree on x-axis):

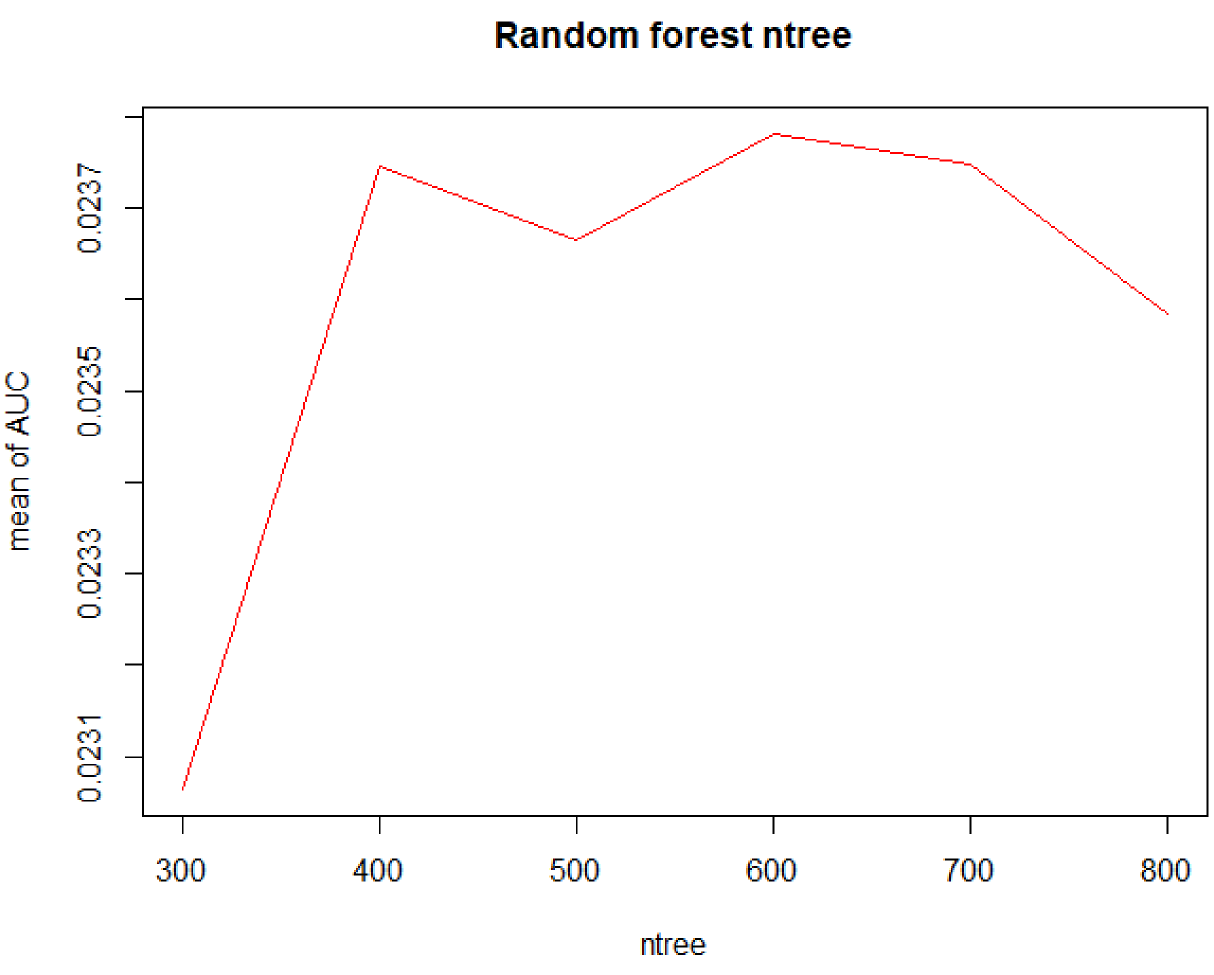
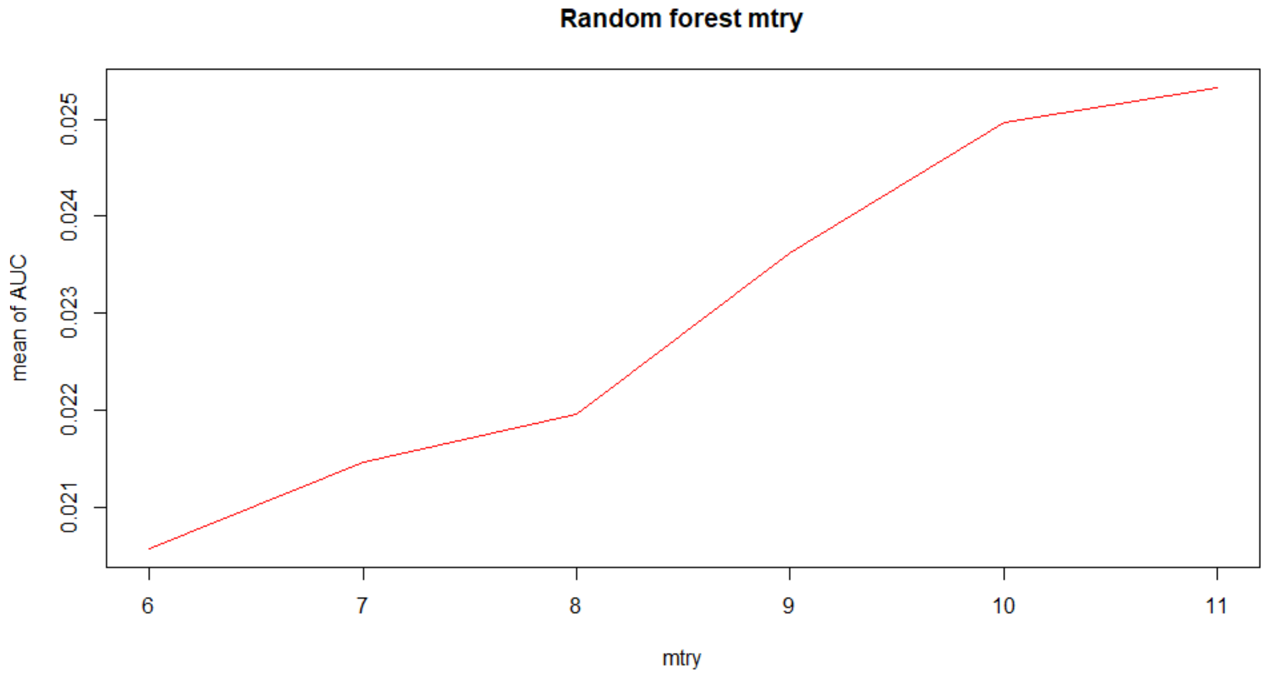


Table of oher results:

ntree	mean o AUC 0.2	std.div	CI start	CI end
300	0,02306	0,00514	0,01939	0,02674
400	0,02375	0,00465	0,02042	0,02707
500	0,02367	0,00553	0,01971	0,02762
600	0,02378	0,00450	0,02056	0,02700
700	0,02375	0,00468	0,02040	0,02710
800	0,02359	0,00464	0,02027	0,02690

Now as we can see the highest mean AUC 0.2 (which is 0.024) is with ntree = 600. Thus we pick this value and adjust mtry using it.

Results of cross validation for mtry:



Other results:

mtry	mean o AUC 0.2	std.div	CI start	CI end
6	0,0210	0,0032	0,0200	0,0270
7	0,0215	0,0047	0,0196	0,0268
8	0,0220	0,0045	0,0199	0,0290
9	0,0240	0,0060	0,0200	0,0270
10	0,0250	0,0048	0,0191	0,0260
11	0,0250	0,0070	0,0190	0,0270

As the result we can see that the best value of mtry is 11, and thus:

The best parameters are: ntree = 600 and mtry = 11 with AUC 0.2 = 0.025

2c)

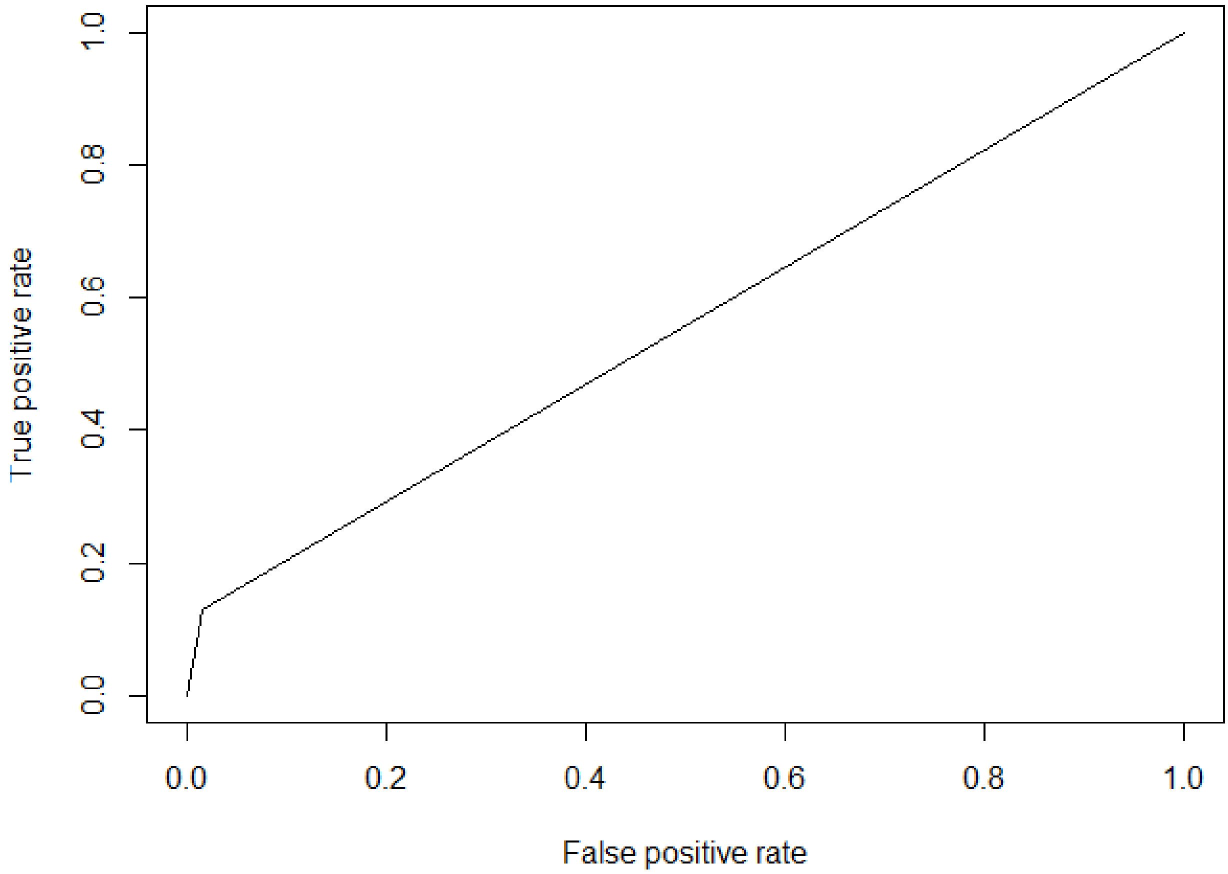
The next method that we are going to tune is Regularized Logistic Regression. Here we are going to tune two parameters: lamda and alpha. The process will be as follows: we will run 10-fold cross-validation adjusting alpha in the range from 0 to 1 with step size 0.1. After each cross validation ends we will determine the best lambda and use it to calculte AUC of the corresponding alpha. The way we decide the best lambda is by picking the lambda with minimum mean cross-validated error.

Here is a table of all alpha values with their AUC and best lambda:

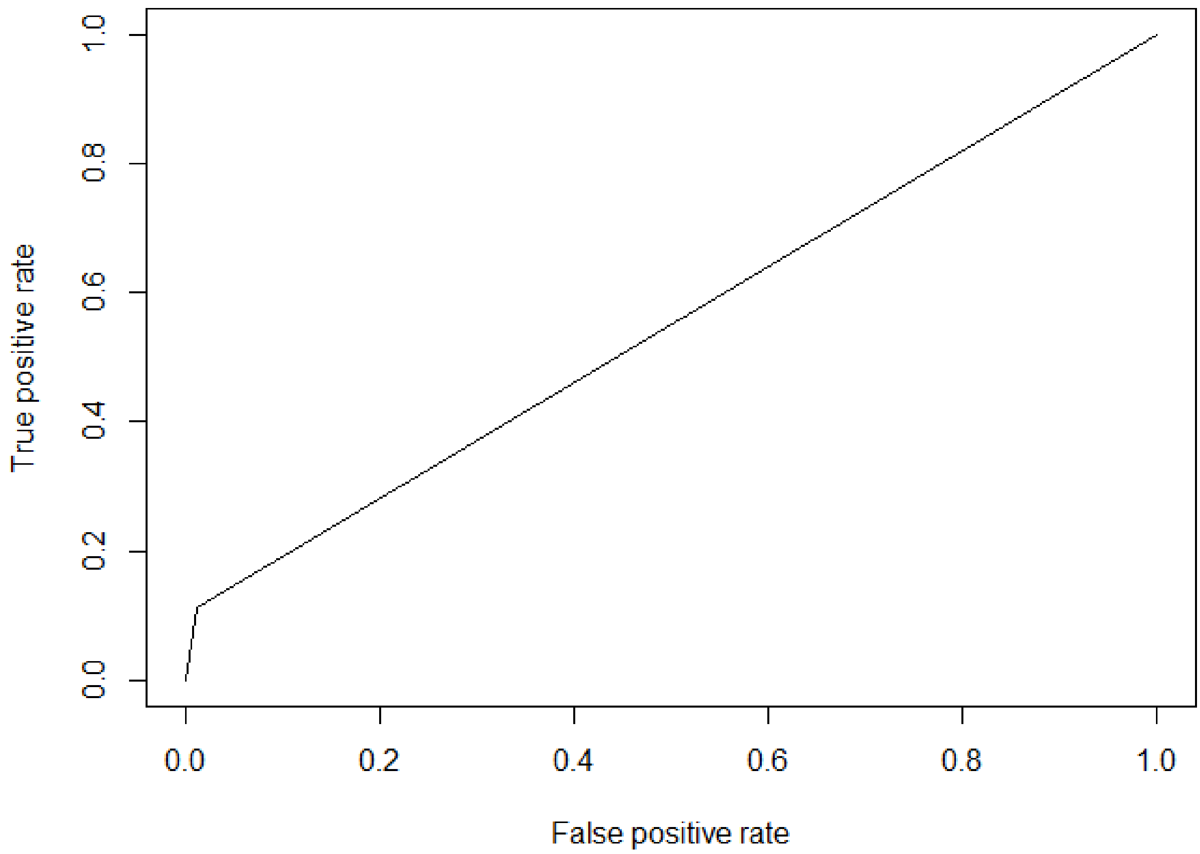
Alpha value	best Lambda	AUC 0.2
0	0,039	0,077
0,1	0,013	0,078
0,2	0,009	0,077
0,3	0,007	0,076
0,4	0,006	0,075
0,5	0,005	0,075
0,6	0,004	0,074
0,7	0,004	0,073
0,8	0,004	0,073
0,9	0,003	0,073
1	0,003	0,073

We can the that the highest AUC 0.2 is with the following parameters: Alpha = 0.1 and lambda = 0.013 which is 0.078

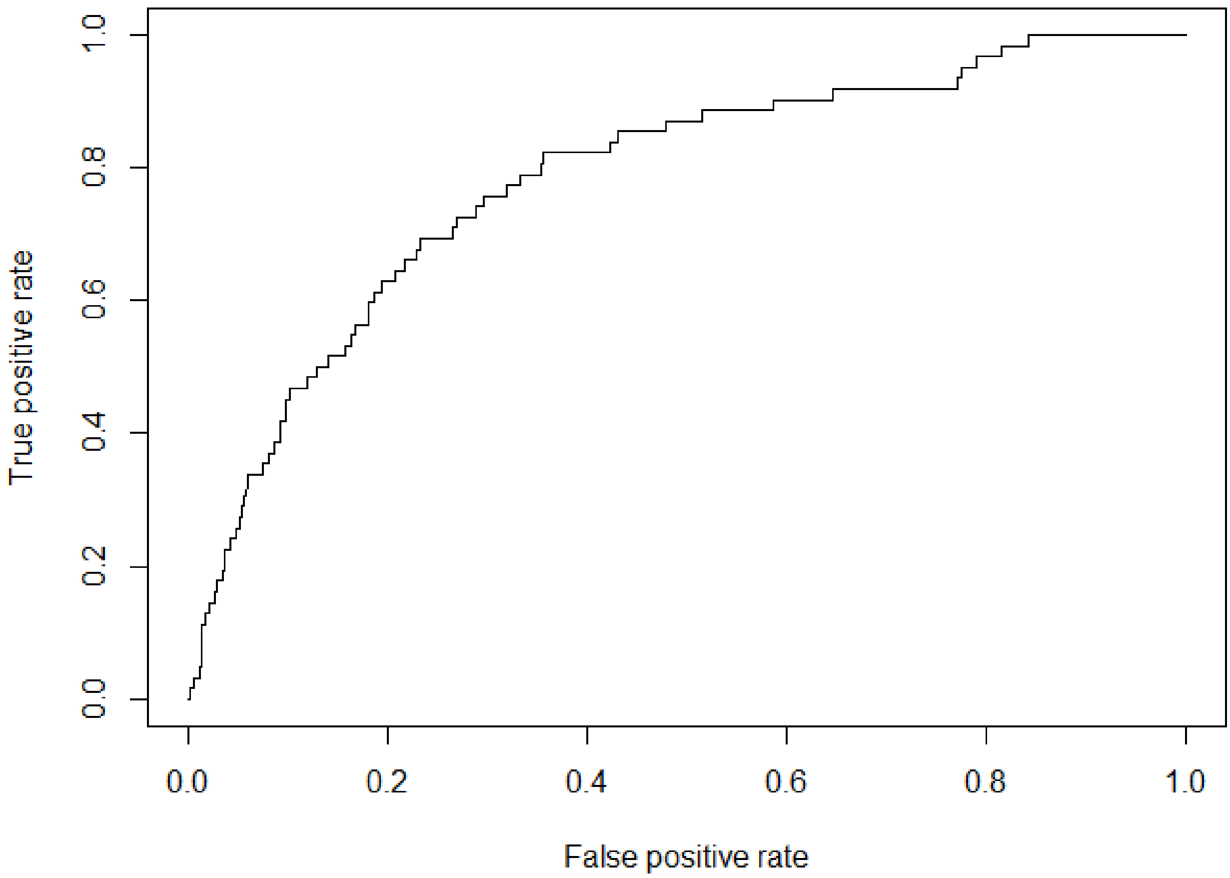
d) Roc curve for Decision tree :



Roc curve for random forest:



Roc curve for Regularized Logistic Regression:



Comparision of the best models of all 3 methods:

Method	AUC 0.2 value
Decision Tree	0,028
Random Forests	0,025
Regularized Logistic Regression	0,078

So comparing these ROC curves we can conclude that regularized logistic regression perfromce better, as it obiously has higher AUC <= 0.2 FPR.

Therfore our best model is Regularized logistic regression with alpha = 0.1 and lambda = 0.013

Task 3 – Model interpretation and feature selection

Variables that were used to build decision tree:

APERSAUT, MAUT1, MBERARBO, MBERMIDD, MFALLEEN, MGODOV, MINK7512 MINKGEM , MINKM30, MOPLHOOG, MOPLLAAG, MOPLMIDD, MOSTYPE, MRELOV, MRELSA, MSKA, MSKC, MSKD, MZFONDS, PBRAND, PMOTSCO, PPERSAUT , PPLEZIER

Variable chosen by lasso:

MGEMLEEF, MGODGE, MRELGE, MOPLHOOG, MOPLLAAG, MBERBOER, MBERMIDD, MHHUUR, MAUT1, MINK7512, MINK123M, MINKGEM, MKOOPKLA, PWAPART, PPERSAUT, PGEZONG, PBRAND, PFIETS, AWALAND, ATRACTOR, AZEILPL, APLEZIER, AFIETS, ABYSTAND.

And these are the best varibles (highest mean decresase gini) from our best Random Forest model:

	MeanDecreaseGini
MOSTYPE	17.777271463
PBRAND	16.338122065
PPERSAUT	15.202280519
APERSAUT	13.140433582
MOPLMIDD	11.380315553
MKOOPKLA	11.186532343
MBERMIDD	10.615777924
MGODGE	10.210050875
MOSHOOFD	10.111764649
MFWEKIND	10.081866828
MOPLLAAG	10.047828490
MINK3045	9.765350839
MFGEKIND	9.710596902
MBERARBG	9.500813743
PWAPART	9.442484541
MGODPR	9.390103498
MSKC	9.050409538
MBERARBO	8.983407750

Variables that appear in all three models are:

MOPLLAAG, PBRAND,PPERSAUT.

These are:

Attribute name	description
MOPLLAAG	Lower level education
PBRAND	Contribution fire policies
PPERSAUT	Contribution car policies

And therefore these are the most important variables in our evaluation.

Task 4 – Final prediction on the blind test set

Here we will use our best model, which is Regularized logistic regression with the following parameters: Alpha = 0.1 and lambda = 0.013. To select 100 most promising potential customers from given 1000 blind examples.

Before that we will train our model on the whole 5822 set and then use it to evaluate on this blind set.

And after that we will use it to predict on the test.

We will obtain a 1000 rows with values between 0 and 1. We will create a copy of it and sort them it descending order then we will record the value of the 100^{th} highest. After that we will use it to filter the original prediction such that all values that are higher or equal than remembered value will be assigned a value of 1 and the rest will be assigned a value of 0.

This will then be converted to a txt file with 1000 rows and only one symbol on this row, either 1 or 0.