The selection of potential customers for a telemarketing promotion

Brief Summary-Jinghan Yang

1. Introduction

This project is about telephone marketing. A bank wants to sell a financial product. The way to promote this product is through direct telephone call to a potential customer. We have a large data set of customers, containing some personal information, and purchase history etc. Given the large size of data set of clients, it is neither practical nor cost-efficient for a business to contact every customer. Therefore it is beneficial to select only a portion of the population that has a high probability to buy this product. The selection of the potential customers forms a typical classification problem with a binary response.

2. This project discussed four questions.

1) How to visualize the data?

PCA is one choice. PCA is not only a method to reduce feature dimensions, but also can be adopted to visualize data. It is very convenient to use the package in R to get some intuitive knowledge of data set by plotting scores of features and loadings of instances for PCA1 and PCA2.

2) How to preprocess the data set based on your knowledge about features?

Exploring knowledge of features, and then modifying and maybe designing new features are very important to final results of analysis. Sometimes it is much more important than the choice of algorithms.

I used two methods, one hot encoding and set education as an ordinal feature.

3) How select criterion to evaluate different models?

Usually the first measure is the test accuracy. But in the real industry, there are some cases, even though the test accuracy seems big, it is similar to the base line of predicting the majoring class. For example, we have a large data set of customers, there are only a little percentage, say 5%, will buy a product. Our job is to predict who will buy such product. Model test accuracy is not very valuable for this case. Even if test accuracy is 95%, it is just equal to to the base line of predicting the majoring class. In this case, we might also care about F1 score, a trade off combination measure of precision and recall. I discussed more details of it for this particular bank telemarketing data set.

4) Feature selection

Sometimes, it might cost a lot to collect some features. So we'd better know what features are important before collecting them. All four methods, lasso and three tree based methods, I used can be applied for feature selection. More details would be discussed in the following part.

3. Data visualization

- PCA

Simple description of the method

It is an unsupervised method to visualize the data. It is a good way to project data with high-dimension to a low-dimensional representation of the data that captures as much of the information as possible. We plotted the first two principal components. (Instance: score, feature: loading).

Result

We see that the first loading vector places approximately equal weight on some features like emp.var.rate (employment variation rate), cons.price.idx (consumer price index), euribor3m (3 month rate eribor) and nr.employed (number of employees). These four variables have relatively high correlation between each other. Therefore the first principal component PC1 represents the overall economic climate at the time of the individual telemarketing phone call. (The second principal component PC2 has no straightforward explanation.)

If some variables come close to each other, it indicates that they are correlated with each other – states with high value of one feature tend to have high values of other features.

4. Data processing

There are many categorical features in the data set. But logistic regression only accepts numerical variables. Although tree based algorithms can build up models on categorical variables, it is actually using one hot encoding when builds up trees. So it's more explicit to preprocess all categorical features to numerical ones. I used two methods to modify the original data set, one hot encoding and set feature, education, as an ordinary feature. One hot coding results 63 features, and ordinal method for education feature results 57 features.

5. Models

I applied four classification methods for this data set, lasso regularized logistic regression, Ada boost, decision tree, random forest.

Logistic regression with Lasso regularization is simple and very fast, and it can help us threw away useless useless features thanks to lasso penalty.

Ada Boost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers [wiki]. It can capture more variance, compared with a single tree. But it is very sensitive to noisy data and outliers [wiki].

Decision Tree. In this case, depth 6 leads to the best test accuracy.

Random forests. The special part for it is that it randomly selects a subsample of features to build up every single tree. And then average result of all trees. Random forests alleviate decision trees' tendency of overfit their training set.

1) Results based on one-hot encoding data set.

- Test accuracy

  Test accuracy for these four methods varies a little bit, some number around 90%. Among these lasso logistic regression is lowest 89.98%. Ada boost is the highest, 90.19%. The accuracy seems high. But in this case, we have around 40,000 instances, but only about 4,000 people brought the financial product. Actually the buy rate is 11.27%. So even the highest test accuracy for Ada boost is just a little bit higher than the base line. So the accuracy is not good. There are two possible explanations. First, the algorithm is not performing well, possibly due to the missing of informative features. Second, the algorithm is performing better than the base line, but the evaluation measure is not discriminative enough.

  So it's better for us not using test accuracy as the only criterion for a model.

- F1 score

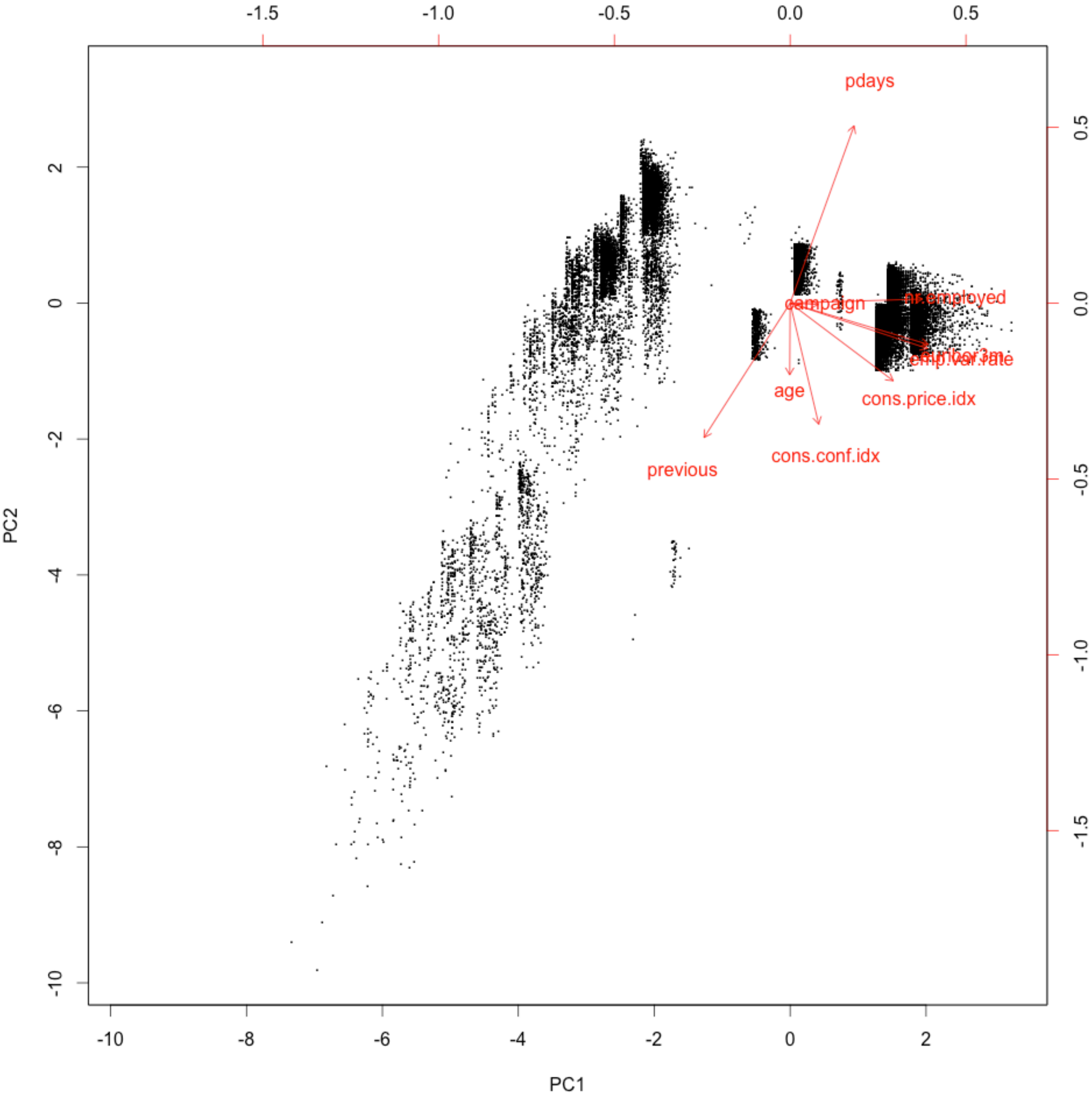
Another measure I considered for this case is F1, which is a combination of precision and recall. (Precision = TP/TP+FP = TP/# predicted positives, Recall = TP/TP+FN = TP/# positives in data). This is for the real cost consideration. For instance, this particular case, we have around 40,000 customers in our data set. The job for us is to find our targeting customers, and then only make phone calls to promote the financial product for targeting group. If we make phone calls for all customers, we can reach out all potential customers who would buy the financial product. But negative consequence for such behavior is that a lot people who have no interest in the financial product will get annoyed by promotion phone calls. A bad reputation of the bank might be carved in their heart. That is the price if we reach out a wrong person. Obviously, we should decide who are our targeting group, containing a trade off between precision and recall. F1 is a measure setting equal weights on precision and recall. In this case, lasso logistic regression has the lowest F1. Other three methods' F1 scores are similar. Random forest has the highest F1 0.50, and the F1 score of Ada boost quite approaches this number. Using the probability threshold which gives the best F1. Ada boost says we should call 2279 customers. Random Forest says we should call 2442 customers. Considering model accuracy and real economy indicator F1 score, the best algorithm for this data set is Random Forest. So the next step for the marketing group is to call these 2442 potential customers to promote this financial product.

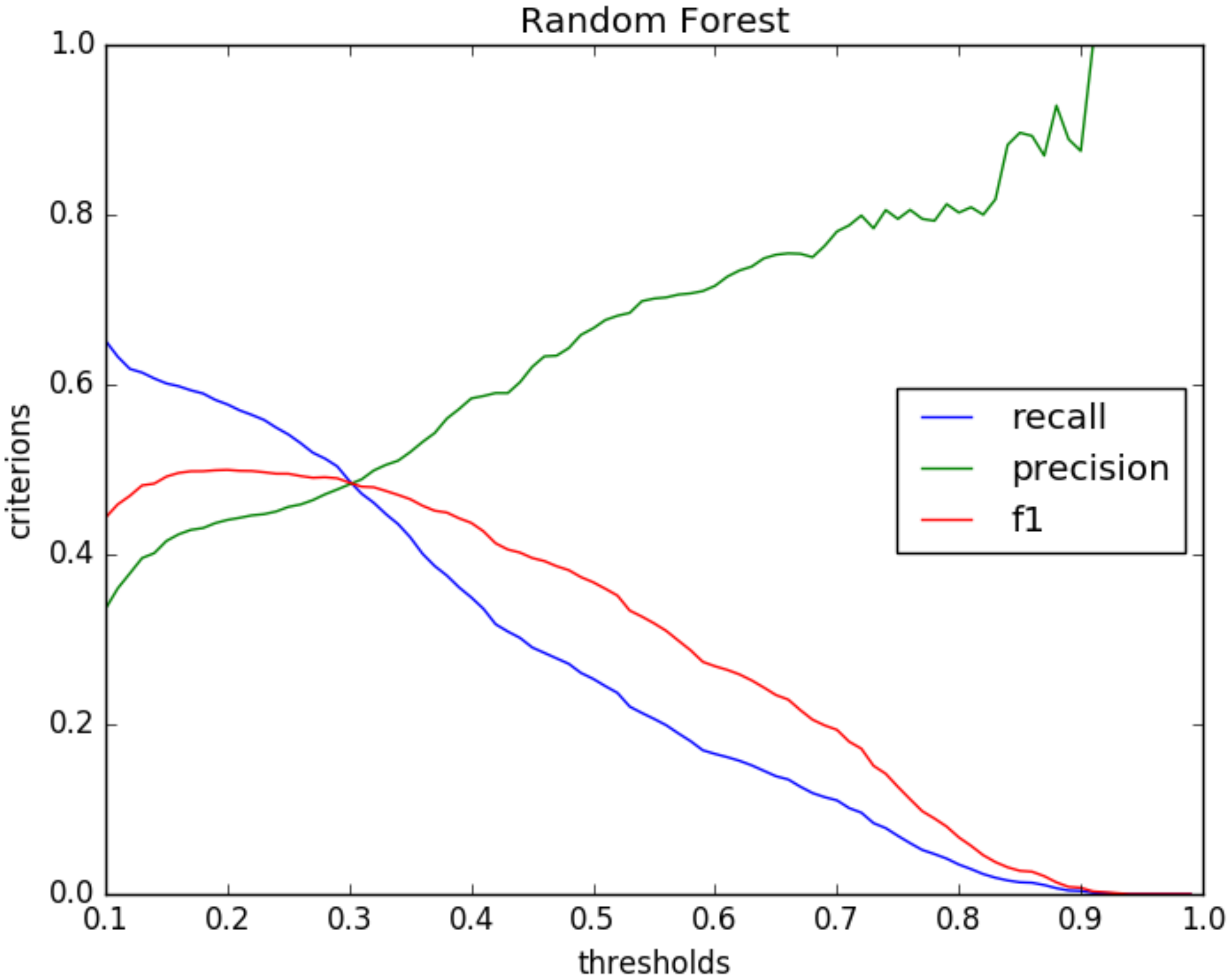
- Feature selection
  Ada boost and Decision tree can return the importance for each feature. For useless feature, it will return zero. For useful feature, the higher the number, the more important the feature is. Ada boost also often depends on tree, and it tempts to capture more variance than one single tree. So features selected by Ada boosting are more than one single tree. We can see that important features are customer's company features, economical climate features and previous campaign time and results. Customer's personal features generally mean very little to buying behavior of this financial product. Only age matters.

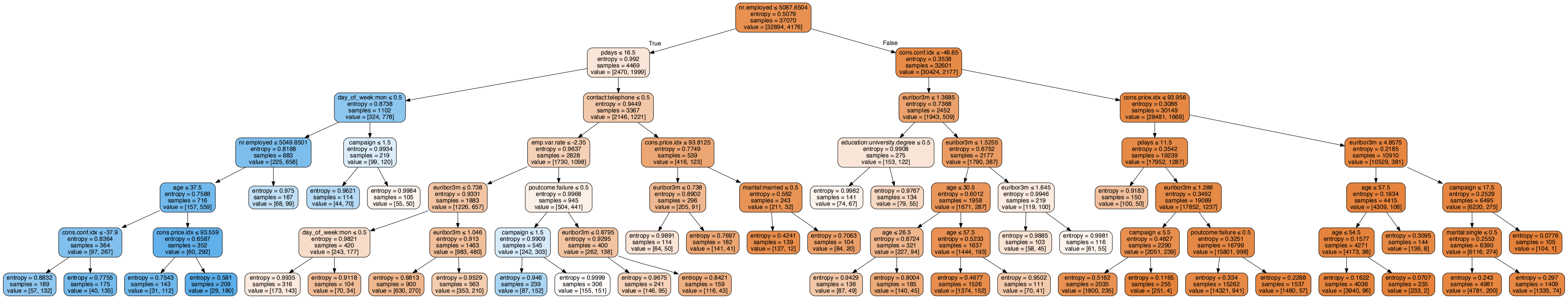
- Comparison
  All these four methods can do feature selection. And they can sort the useful features according to their importance. In this case, lasso selected the smallest number of useful features. And then a 6 depth decision tree gives the lowest test error. Ada boost and Random Forest gave us more features, since they captured more variance of the data set. I'd like to use the decision tree to visualize useful features. For important features, it could appears many time in the internal nodes and also decision tree could give us how many entropy, which is the purity of the node, it has for each node.

2) **Results based ordinal feature (education) data set.**
   I didn't change any parameters for building up models for each method.
   The results are almost same with one-hot encoding. The result is that education feature is not important for the final result. We can see that, for this data set, Ada boost selected over 40 features from 57 features, but education is not one of them.

So no matter how we encode this feature, result would not be affected very much.
But this encoding choice might be very important for other purposes.
Since the result doesn’t change very much. I put them in appendix part.


-1.5
-1.0
-0.5
0.0
0.5
pdays
campaign
nr.employed
euribor3m
emp.var.rate
age
cons.price.idx
previous
cons.conf.idx
2
0
-2
-4
-6
-8
-10
PC2
0.5
0.0
-0.5
-1.0
-1.5
-10
-8
-6
-4
-2
0
2
PC1


Random Forest
criterions
thresholds
1.0
0.8
0.6
0.4
0.2
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
recall
precision
f1


nr.employed ≤ 5087.6504
entropy = 0.5079
samples = 37070
value = [32894, 4176]
True
False
pdays ≤ 16.5
entropy = 0.992
samples = 4469
value = [2470, 1999]
cons.conf.idx ≤ -46.65
entropy = 0.3538
samples = 32601
value = [30424, 2177]
day_of_week:mon ≤ 0.5
entropy = 0.8738
samples = 1102
value = [324, 778]
contact:telephone ≤ 0.5
entropy = 0.9449
samples = 3367
value = [2146, 1221]
euribor3m ≤ 1.3685
entropy = 0.7368
samples = 2452
value = [1943, 509]
cons.price.idx ≤ 93.956
entropy = 0.3086
samples = 30149
value = [28481, 1668]
nr.employed ≤ 5049.8501
entropy = 0.8188
samples = 883
value = [225, 658]
campaign ≤ 1.5
entropy = 0.9934
samples = 219
value = [99, 120]
emp.var.rate ≤ -2.35
entropy = 0.9637
samples = 2828
value = [1730, 1098]
cons.price.idx ≤ 93.8125
entropy = 0.7749
samples = 539
value = [416, 123]
education:university.degree ≤ 0.5
entropy = 0.9908
samples = 275
value = [153, 122]
euribor3m ≤ 1.5255
entropy = 0.6752
samples = 2177
value = [1790, 387]
pdays ≤ 11.5
entropy = 0.3542
samples = 19239
value = [17952, 1287]
euribor3m ≤ 4.8575
entropy = 0.2185
samples = 10910
value = [10529, 381]
age ≤ 37.5
entropy = 0.7588
samples = 716
value = [157, 559]
entropy = 0.975
samples = 167
value = [68, 99]
entropy = 0.9621
samples = 114
value = [44, 70]
entropy = 0.9984
samples = 105
value = [55, 50]
euribor3m ≤ 0.738
entropy = 0.9331
samples = 1883
value = [1226, 657]
poutcome:failure ≤ 0.5
entropy = 0.9968
samples = 945
value = [504, 441]
euribor3m ≤ 0.738
entropy = 0.8902
samples = 296
value = [205, 91]
marital:married ≤ 0.5
entropy = 0.562
samples = 243
value = [211, 32]
entropy = 0.9982
samples = 141
value = [74, 67]
entropy = 0.9767
samples = 134
value = [79, 55]
age ≤ 30.5
entropy = 0.6012
samples = 1958
value = [1671, 287]
euribor3m ≤ 1.645
entropy = 0.9946
samples = 219
value = [119, 100]
entropy = 0.9183
samples = 150
value = [100, 50]
euribor3m ≤ 1.286
entropy = 0.3462
samples = 19089
value = [17852, 1237]
age ≤ 57.5
entropy = 0.1634
samples = 4415
value = [4309, 106]
campaign ≤ 17.5
entropy = 0.2529
samples = 6495
value = [6220, 275]
cons.conf.idx ≤ -37.9
entropy = 0.8364
samples = 364
value = [97, 267]
cons.price.idx ≤ 93.559
entropy = 0.6587
samples = 352
value = [60, 292]
day_of_week:mon ≤ 0.5
entropy = 0.9821
samples = 420
value = [243, 177]
euribor3m ≤ 1.046
entropy = 0.913
samples = 1463
value = [983, 480]
campaign ≤ 1.5
entropy = 0.9909
samples = 545
value = [242, 303]
euribor3m ≤ 0.8795
entropy = 0.9295
samples = 400
value = [262, 138]
entropy = 0.9891
samples = 114
value = [64, 50]
entropy = 0.7697
samples = 182
value = [141, 41]
entropy = 0.4241
samples = 139
value = [127, 12]
entropy = 0.7063
samples = 104
value = [84, 20]
age ≤ 26.5
entropy = 0.8724
samples = 321
value = [227, 94]
age ≤ 57.5
entropy = 0.5233
samples = 1637
value = [1444, 193]
entropy = 0.9885
samples = 103
value = [58, 45]
entropy = 0.9981
samples = 116
value = [61, 55]
campaign ≤ 5.5
entropy = 0.4827
samples = 2290
value = [2051, 239]
poutcome:failure ≤ 0.5
entropy = 0.3251
samples = 16799
value = [15801, 998]
age ≤ 54.5
entropy = 0.1577
samples = 4271
value = [4173, 98]
entropy = 0.3095
samples = 144
value = [136, 8]
marital:single ≤ 0.5
entropy = 0.2553
samples = 6390
value = [6116, 274]
entropy = 0.0776
samples = 105
value = [104, 1]
entropy = 0.8832
samples = 189
value = [57, 132]
entropy = 0.7755
samples = 175
value = [40, 135]
entropy = 0.7543
samples = 143
value = [31, 112]
entropy = 0.581
samples = 209
value = [29, 180]
entropy = 0.9935
samples = 316
value = [173, 143]
entropy = 0.9118
samples = 104
value = [70, 34]
entropy = 0.8813
samples = 900
value = [630, 270]
entropy = 0.9529
samples = 563
value = [353, 210]
entropy = 0.946
samples = 239
value = [87, 152]
entropy = 0.9999
samples = 306
value = [155, 151]
entropy = 0.9675
samples = 241
value = [146, 95]
entropy = 0.8421
samples = 159
value = [116, 43]
entropy = 0.9429
samples = 136
value = [87, 49]
entropy = 0.8004
samples = 185
value = [140, 45]
entropy = 0.4677
samples = 1526
value = [1374, 152]
entropy = 0.9502
samples = 111
value = [70, 41]
entropy = 0.5162
samples = 2035
value = [1800, 235]
entropy = 0.1165
samples = 255
value = [251, 4]
entropy = 0.334
samples = 15262
value = [14321, 941]
entropy = 0.2288
samples = 1537
value = [1480, 57]
entropy = 0.1622
samples = 4036
value = [3940, 96]
entropy = 0.0707
samples = 235
value = [233, 2]
entropy = 0.243
samples = 4981
value = [4781, 200]
entropy = 0.297
samples = 1409
value = [1335, 74]

```
One-hot encoding
key:1   value:set(['management', 'retired', 'self-employed',
'unknown', 'unemployed', 'admin.', 'technician', 'services',
'student', 'housemaid', 'entrepreneur', 'blue-collar'])
key:2   value:set(['unknown', 'single', 'married', 'divorced'])
key:3   value:set(['basic.9y', 'illiterate', 'basic.4y', 'unknown',
'basic.6y', 'high.school', 'professional.course',
'university.degree'])
key:4   value:set(['unknown', 'yes', 'no'])
key:5   value:set(['unknown', 'yes', 'no'])
key:6   value:set(['unknown', 'yes', 'no'])
key:7   value:set(['telephone', 'cellular'])
key:8   value:set(['mar', 'aug', 'sep', 'may', 'jun', 'jul', 'apr',
'nov', 'dec', 'oct'])
key:9   value:set(['fri', 'thu', 'wed', 'mon', 'tue'])
key:14  value:set(['failure', 'success', 'nonexistent'])
buyRate equals to 0.112651436063
***************lasso**************
start training for logistic regression...
useful = ['cons.price.idx', 'euribor3m', 'emp.var.rate',
'cons.conf.idx', 'age', 'nr.employed', 'pdays']
training done
training accuracy = 0.899724830042
test accuary = 0.898943918427
best threshold = 0.16
best F1 0.458470083312
*************************************
***************decisionTree**************
6
accuracy for test: 0.900383633056
['nr.employed', 'cons.conf.idx', 'pdays', 'euribor3m',
'cons.price.idx', 'age', 'poutcome:failure', 'campaign',
'contact:telephone', 'emp.var.rate', 'day_of_week:mon',
'marital:married', 'marital:single', 'education:university.degree']
Plot decision tree
Jun 11 01:44:29  dot[24578] <Error>: The function ‘CGFontGetGlyphPath’
is obsolete and will be removed in an upcoming update. Unfortunately,
this application, or a library it uses, is using this obsolete
function, and is thereby contributing to an overall degradation of
system performance.
Jun 11 01:44:29  dot[24578] <Error>: The function
‘CGFontGetGlyphPaths’ is obsolete and will be removed in an upcoming
update. Unfortunately, this application, or a library it uses, is
using this obsolete function, and is thereby contributing to an
overall degradation of system performance.

best threshold = 0.28
best F1 0.486540378863
***************************************
***************adaboost**************
```

```
start training for ada boosting...
training done
['nr.employed', 'euribor3m', 'cons.conf.idx', 'pdays', 'age',
'poutcome:success', 'cons.price.idx', 'campaign', 'default:unknown',
'education:university.degree', 'contact:cellular',
'contact:telephone', 'day_of_week:mon', 'poutcome:failure',
'previous', 'day_of_week:fri', 'loan:no', 'default:no',
'day_of_week:thu', 'marital:married', 'emp.var.rate',
'education:professional.course', 'month:apr', 'job:services',
'job:housemaid', 'job:technician', 'day_of_week:wed', 'housing:yes',
'loan:yes', 'marital:single', 'education:basic.4y', 'job:admin.',
'housing:no', 'loan:unknown', 'poutcome:nonexistent',
'job:unemployed', 'job:management', 'day_of_week:tue',
'education:basic.6y', 'education:basic.9y', 'month:aug', 'month:nov']
best threshold = 0.3
best F1 0.494571773221
best call number is 2279
accuracy for train:0.902719326643
accuracy for test:0.901857246905
***************************************
**************Random Forest************
start training for Random Forest
training done
useful = ['nr.employed', 'euribor3m', 'pdays', 'cons.price.idx',
'emp.var.rate', 'age', 'cons.conf.idx', 'poutcome:success',
'previous', 'campaign', 'contact:cellular', 'month:may', 'month:oct',
'contact:telephone', 'poutcome:nonexistent', 'default:no',
'poutcome:failure', 'day_of_week:tue', 'education:university.degree',
'day_of_week:wed', 'default:unknown', 'day_of_week:fri',
'education:basic.9y', 'month:mar', 'marital:single',
'marital:divorced', 'loan:no', 'job:admin.', 'marital:married',
'day_of_week:mon', 'housing:no', 'education:basic.4y', 'housing:yes',
'day_of_week:thu', 'job:entrepreneur',
'education:professional.course', 'job:technician',
'education:unknown', 'job:blue-collar', 'education:basic.6y',
'education:high.school', 'job:student', 'loan:unknown',
'job:services', 'job:retired', 'job:self-employed', 'month:jul',
'job:unknown', 'job:unemployed', 'job:management', 'job:housemaid',
'loan:yes', 'month:apr', 'housing:unknown', 'month:jun', 'month:aug',
'month:nov', 'marital:unknown', 'month:sep']
accuracy for train:0.910326966656
accuracy for test:0.901068220442
best threshold = 0.2
best F1 0.499535747447
best call number is 2442
**************************************
```

```
Ordinal Feature encoding
key:1   value:set(['management', 'retired', 'self-employed',
'unknown', 'unemployed', 'admin.', 'technician', 'services',
'student', 'housemaid', 'entrepreneur', 'blue-collar'])
key:2   value:set(['unknown', 'single', 'married', 'divorced'])
key:3   value:set(['basic.9y', 'illiterate', 'basic.4y', 'unknown',
'basic.6y', 'high.school', 'professional.course',
'university.degree'])
key:4   value:set(['unknown', 'yes', 'no'])
key:5   value:set(['unknown', 'yes', 'no'])
key:6   value:set(['unknown', 'yes', 'no'])
key:7   value:set(['telephone', 'cellular'])
key:8   value:set(['mar', 'aug', 'sep', 'may', 'jun', 'jul', 'apr',
'nov', 'dec', 'oct'])
key:9   value:set(['fri', 'thu', 'wed', 'mon', 'tue'])
key:14  value:set(['failure', 'success', 'nonexistent'])
buyRate equals to 0.112651436063
duration
campaign
['age', 'job:management', 'job:retired', 'job:self-employed',
'job:unknown', 'job:unemployed', 'job:admin.', 'job:technician',
'job:services', 'job:student', 'job:housemaid', 'job:entrepreneur',
'job:blue-collar', 'marital:unknown', 'marital:single',
'marital:married', 'marital:divorced', 'education',
'education:unknown', 'default:unknown', 'default:yes', 'default:no',
'housing:unknown', 'housing:yes', 'housing:no', 'loan:unknown',
'loan:yes', 'loan:no', 'contact:telephone', 'contact:cellular',
'month:mar', 'month:aug', 'month:sep', 'month:may', 'month:jun',
'month:jul', 'month:apr', 'month:nov', 'month:dec', 'month:oct',
'day_of_week:fri', 'day_of_week:thu', 'day_of_week:wed',
'day_of_week:mon', 'day_of_week:tue', 'campaign', 'pdays', 'previous',
'poutcome:failure', 'poutcome:success', 'poutcome:nonexistent',
'emp.var.rate', 'cons.price.idx', 'cons.conf.idx', 'euribor3m',
'nr.employed']
***************lasso**************
start training for logistic regression...
useful = ['cons.price.idx', 'euribor3m', 'emp.var.rate',
'cons.conf.idx', 'nr.employed', 'age', 'pdays']
training done
training accuracy = 0.899805762383
test accuary = 0.899065307113
best threshold = 0.15
best F1 0.458752515091
*************************************
***************decisionTree**************
6
accuracy for test: 0.900383633056
['nr.employed', 'cons.conf.idx', 'pdays', 'euribor3m',
'cons.price.idx', 'age', 'poutcome:failure', 'campaign',
'contact:telephone', 'day_of_week:mon', 'education',
'marital:married', 'marital:single']
Plot decision tree
Jun 11 01:47:01  dot[24622] <Error>: The function ‘CGFontGetGlyphPath’
is obsolete and will be removed in an upcoming update. Unfortunately,
this application, or a library it uses, is using this obsolete
function, and is thereby contributing to an overall degradation of
system performance.
Jun 11 01:47:01  dot[24622] <Error>: The function
‘CGFontGetGlyphPaths’ is obsolete and will be removed in an upcoming
update. Unfortunately, this application, or a library it uses, is
using this obsolete function, and is thereby contributing to an
overall degradation of system performance.

best threshold = 0.28
best F1 0.486540378863
***************************************
***************adaboost**************
start training for ada boosting...
training done
['nr.employed', 'euribor3m', 'cons.conf.idx', 'pdays', 'age',
'poutcome:success', 'cons.price.idx', 'campaign', 'education',
'poutcome:failure', 'default:no', 'default:unknown',
'contact:telephone', 'day_of_week:mon', 'day_of_week:thu',
'day_of_week:fri', 'contact:cellular', 'previous', 'loan:no',
'day_of_week:wed', 'month:mar', 'emp.var.rate', 'marital:married',
'poutcome:nonexistent', 'marital:single', 'job:technician',
'job:services', 'job:management', 'housing:yes', 'loan:yes',
'job:blue-collar', 'housing:unknown', 'job:admin.', 'job:unemployed',
'month:may', 'day_of_week:tue', 'housing:no', 'month:nov',
'marital:divorced']
best threshold = 0.28
best F1 0.493209435311
best call number is 2331
accuracy for train:0.902597928132
accuracy for test:0.901189609128
***************************************
**************Random Forest************
start training for Random Forest
training done
useful = ['nr.employed', 'emp.var.rate', 'euribor3m',
'poutcome:success', 'pdays', 'cons.price.idx', 'age', 'campaign',
'month:may', 'cons.conf.idx', 'month:oct', 'education', 'previous',
'contact:telephone', 'month:mar', 'default:unknown',
'contact:cellular', 'day_of_week:fri', 'poutcome:failure',
'marital:divorced', 'loan:no', 'housing:no', 'day_of_week:tue',
'marital:single', 'poutcome:nonexistent', 'job:retired', 'month:aug',
'job:admin.', 'day_of_week:mon', 'housing:yes', 'marital:married',
'day_of_week:wed', 'default:no', 'job:blue-collar', 'job:technician',
'day_of_week:thu', 'job:housemaid', 'job:management', 'loan:unknown',
'month:apr', 'marital:unknown', 'job:unemployed', 'loan:yes',
'housing:unknown', 'job:student', 'month:jul', 'job:services',
'job:unknown', 'job:self-employed', 'education:unknown',
'job:entrepreneur', 'month:nov', 'month:jun', 'month:dec',
'month:sep']
accuracy for train:0.910326966656
accuracy for test:0.89936877883
best threshold = 0.2
best F1 0.498377375985
best call number is 2448
**************************************
```