DSA 5013 Intelligent Data Analytics

Assignment # 5

Saeid Hosseinipoor

Problem 1.

Problem 1 – part a.

A user defined function was developed to analyze a model based of different statistics and plot. The function is CPE which instance for Classification Performance Evaluation. The following syntax is the form of usage:

CPE (Observed, Predicted, ...)

Inputs:

Observed:	is a data set of true value which is called observed values.
Predicted:	is a data set of predicted values where predicted by a model.

Outputs:

Confusion matrix and statistics:	Uses InformationValue package.
ROC curve and AUC:	Uses ROCR package.
Concordant Pairs:	Uses InformationValue package.
D statistics:	Calculates three types of D statistics: Hoeffding's, Somer's and mean difference.
K-S chart and statistics:	Uses InformationValue package.
Distribution:	Uses sm package.
Lift Chart:	Uses prediction function.

Example:

```
honors <- read.csv("honors.csv")
fit <- glm(data=honors, hon ~ math + read + female , family="binomial")
Predicted = predict(fit)
Predicted = as.numeric (Predicted >= 0)
Observed = honors$hon
test = CPE (Observed, Predicted)
test$Concordant
test$AUROC
```

```
> [1] 0.4476281
> [2] 0.6807677
```

Problem 2.

Problem 2 – Part a.

The paper was read.

Problem 2 – Part b.

Marketing selling campaigns use different approaches to promote the business. One the strategies is they call a part of customers whom are more likely to get involved in the business. In this paper authors tried to find this types of customer. They used the data collected about the successful and non-successful contact. The contacts were two classes of inbound or outbound. The outbound contacts are those that agents call the customer and offer the services. In inbound contacts, they redirect the customers who called the company and make an offer for them.

Decision support systems (DSSs) use information technology to support managerial decision making. There are two types of personal and intelligent DSSs. In personal DSS, experts and managers use available information and process this information based on their experiments and personal judgement. Intelligent Dss tries to establish an algorithm to make this type of decision. It helps to develop a reliable and fast decision process that could be applied by any person.

There are several classification models, such as the classical Logistic Regression (LR), Decision Trees (DTs), and more recent neural networks (NNs) and Support Vector Machines (SVMs).

The paper evaluates the data mining models to predict the success of telemarketing calls for selling bank long term deposits. A large dataset with 150 features related to bank client, product, and socio-economic attributes has been analyzed with logistic regression, decision trees, neural network, and support vector machine as implemented in the rminer package of R.

The three major contributions of this research are:

- feature selection and engineering – the paper demonstrates and effective approach to narrow down bank attributes data using business knowledge –as a semi-automated feature selection process.
- Rolling Window Evaluation – the paper demonstrates an approach in which four models, i.e. Logistic Regression, Decision Trees, Neural Network, and Support Vector Machine are evaluated with respect to AUC and area under cumulative lift chart using moving window of data.
- Results comparison and effectiveness of neural network modeling strategy for the problem

The paper deals with an unbalanced dataset since out of 52,944 contacts only 6,557 are successful (12.38%). The findings indicate that the statistics for logistic regression, decision trees, support vector machines, and neural network classifiers progressively get better with respect to AUC from 0.715 for LR and 0.794 for NN. Similarly, with respect to ALIFT the result of 0.626 for LR improves to 0.672 for NN.

The authors claim that NN model is best suited for optimizing telemarketing costs for long term bank deposits, which produces 80% success rate by reducing the current phone contacts by 50%. Also, two knowledge extraction methods: sensitivity analysis and DT applied to NN model to reveal Euribor rate, direction of the call and bank agent experience as key attributes for success. The conclusions of the paper are sound and will prove effective on very large scale for large banks.

Problem 3.

Problem 3 – Part a.

The data set was loaded into R variable. The data set was investigated in terms of missing values. There were no missing values in the observations. Some values were marked as unknown. It seems that these values should be missing. Therefore; after considering these values as missing values, housing variable has more than 20% values missed. This variable was removed from data set. In the next step, the correlations between numeric values were investigated. Some variables which showed high multicollinearity were deleted from the data set. Detailed process is available in attached R script.

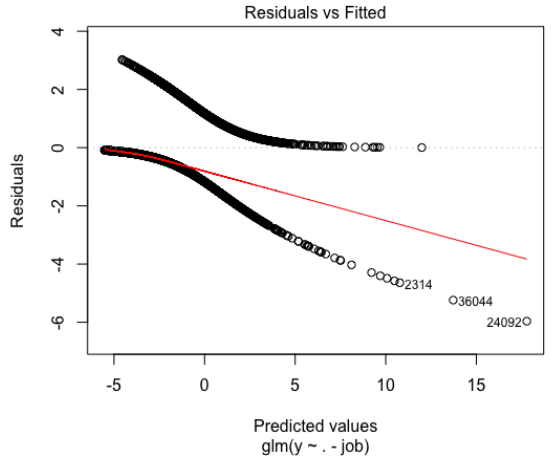
```
Coefficients:
                      Estimate Std. Error  t value Pr(>|t|)
(Intercept)          1.356e+04  1.955e+01  693.799  < 2e-16 ***
cons.price.idx      -9.012e+01  2.059e-01 -437.771  < 2e-16 ***
euribor3m            1.199e+01  1.442e-01   83.137  < 2e-16 ***
emp.var.rate         5.255e+01  1.956e-01  268.568  < 2e-16 ***
previous            -5.845e-01  1.535e-01   -3.806 0.000141 ***
monthaug            -4.461e+01  1.986e-01 -224.592  < 2e-16 ***
monthdec            -3.620e+01  5.226e-01  -69.256  < 2e-16 ***
monthjul            -4.644e+00  1.810e-01  -25.658  < 2e-16 ***
monthjun             3.931e+01  1.914e-01  205.390  < 2e-16 ***
monthmar            -3.700e+01  3.193e-01 -115.880  < 2e-16 ***
monthmay            -1.656e+01  1.533e-01 -108.031  < 2e-16 ***
monthnov            -1.369e+01  2.024e-01  -67.648  < 2e-16 ***
monthoct            -3.343e+01  3.022e-01 -110.609  < 2e-16 ***
monthsep            -5.696e+01  3.166e-01 -179.926  < 2e-16 ***
contacttelephone     1.359e+00  1.251e-01   10.864  < 2e-16 ***
campaign             4.784e-02  1.220e-02    3.922 8.80e-05 ***
poutcomenonexistent -1.165e+00  2.137e-01   -5.451 5.03e-08 ***
poutcomesuccess     -3.575e+00  2.230e-01  -16.036  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.723 on 41170 degrees of freedom
Multiple R-squared:  0.9913,    Adjusted R-squared:  0.9913
F-statistic: 2.774e+05 on 17 and 41170 DF,  p-value: < 2.2e-16
```

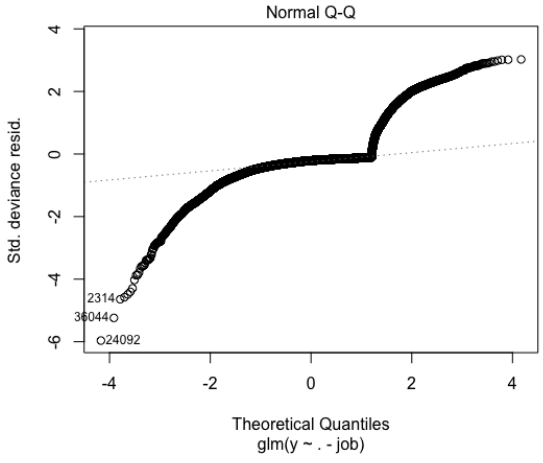
The remaining missing values were imputed by kNN technique using k = 10. Totally 11 predictors were remained to make the models. 80% of the observations were randomly selected as a training set, and the rest were hold for testing purposes.

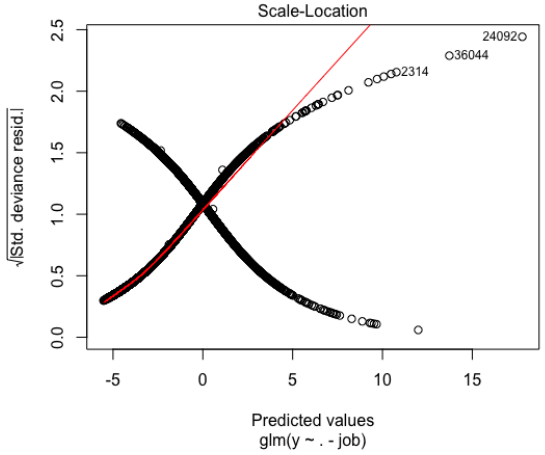
Problem 3 – Part b.

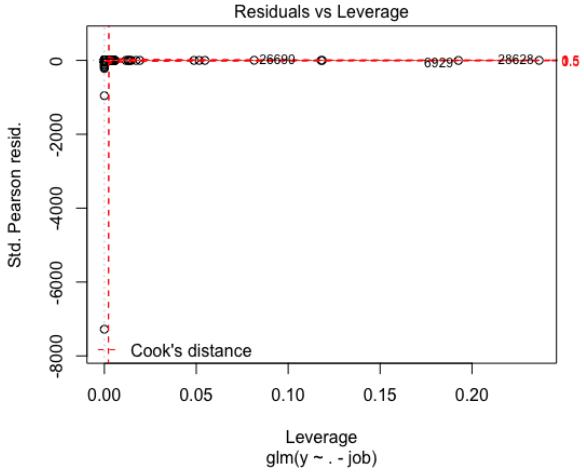
A model was designed based on the 11 predictors from last section. ‘glm’ function was used to establish this model. Model’s statistics and other measurements like AIC, and vif re checked. A variable was removed to get the better fit for this data set. The details are available in the R Script. Following graph shows the residuals behavior and relationship for this model.

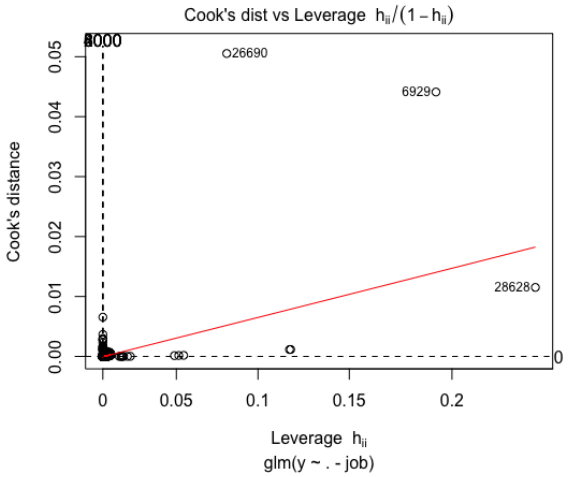
A stepwise AIC method also was performed. The results were same. The model seems to be the optimum model at this moment. A better model could be achievable if interaction between the predictors and nonlinearity is considered. Specially, Q-Q plot suggests a more complex model than a linear one. Cook’s distances also implies on some outliers.

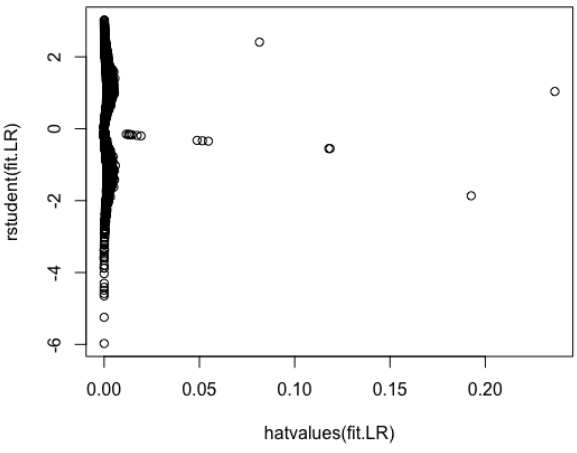

Residuals vs Fitted
Residuals
Predicted values
glm(y ~ . - job)
2314
36044
24092


Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
glm(y ~ . - job)
2314
36044
24092


Scale-Location
√|Std. deviance resid.|
Predicted values
glm(y ~ . - job)
24092
36044
2314


Residuals vs Leverage
Std. Pearson resid.
Leverage
glm(y ~ . - job)
Cook's distance
26690
6929
28628
0.5


Cook's dist vs Leverage hii/(1 − hii)
Cook's distance
Leverage hii
glm(y ~ . - job)
26690
6929
28628


rstudent(fit.LR)
hatvalues(fit.LR)

Problem 3 – Part c.

For this section, different techniques have been applied to model the data set; Elastic net regularization, decision tree, random forest and boosted tree.

The first model was the elastic net regularization from package. The following commands were applied:

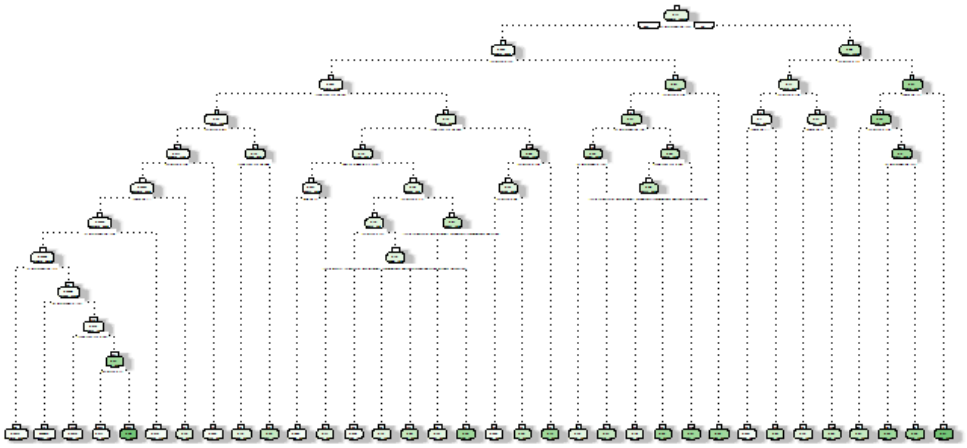
fitControl = trainControl (method="cv", number=10

fit.EN = train(y~., data=Bank.Training, method="glmnet", trControl=fitControl)

The following results were obtained by 10 fold cross validation:

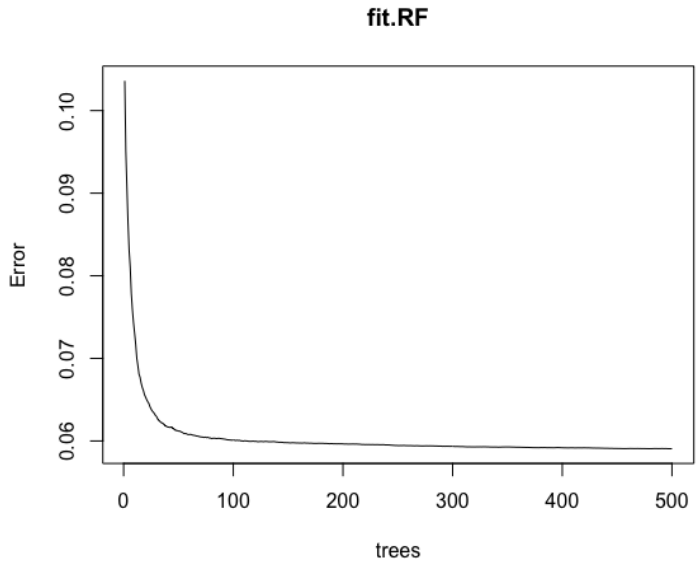
```
glmnet

32950 samples
   11 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 29655, 29655, 29655, 29655, 29655, 29655, ...
Resampling results across tuning parameters:

  alpha  lambda        RMSE       Rsquared
  0.10   0.0002604295  0.2608720  0.3242029
  0.10   0.0026042955  0.2608699  0.3242236
  0.10   0.0260429546  0.2612557  0.3242472
  0.55   0.0002604295  0.2608706  0.3242072
  0.55   0.0026042955  0.2608724  0.3242851
  0.55   0.0260429546  0.2632791  0.3189277
  1.00   0.0002604295  0.2608681  0.3242201
  1.00   0.0026042955  0.2609183  0.3241662
  1.00   0.0260429546  0.2653736  0.3160726

RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were alpha = 1 and lambda = 0.0002604295.
```

The second model was built implementing decision trees technique. Unfortunately, the fancy illustration from rattle package didn't work and the following ugly figure was the only available tree. The R script is attached.

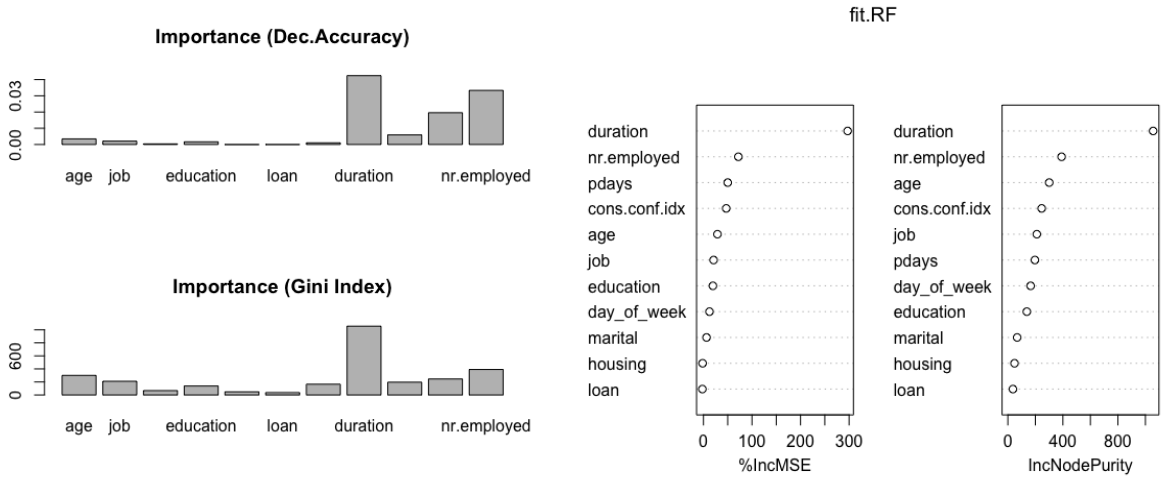


Rattle 2016-Nov-20 23:18:35 Saied

The third model was the random forest which is consisting of number of the decision trees. The following figure shows that increasing number of trees decreases the error. After 100 trees the improvement is not very considerable.



The variables in the decision trees are not as same as important. The philosophy of the random forest is to force the different variables on the root of the tree. The following figures show the importance of the variable in this model:



Duration is the most important variable, in the secod place number of employee is carying the information of some other variables that were deleted. This importance order is cosistant with logistic regression on previous section.

The last model was boosting which was applied as:

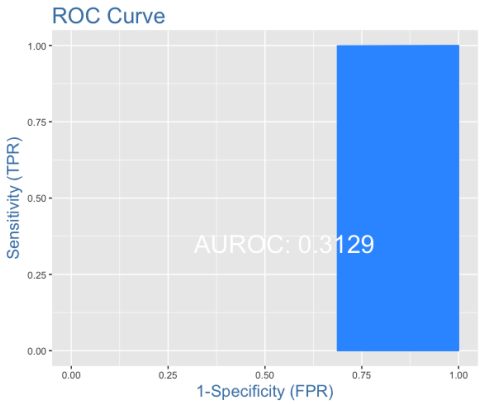
a = Bank.Training

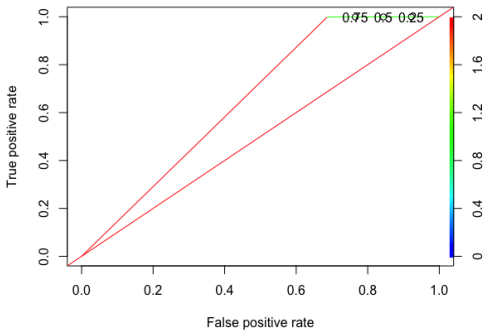
a$y = as.factor(a$y)

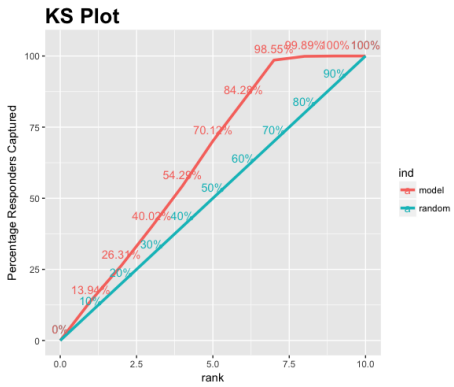
fit.BT <- boosting(y ~ ., data=a, boos=F, mfinal=20)

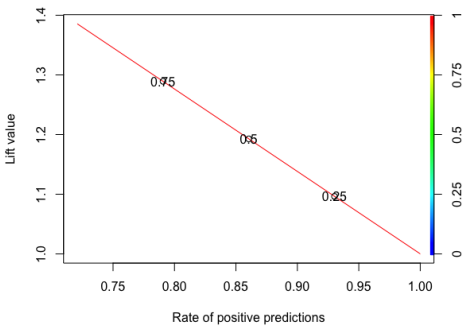
Problem 3 – Part d.

All the models from the previous section were examined by the user defined fuction CPE which was prepared in problem 1. Models were tested by using the test data set that were hold at the beginning. The results are demnstrated as follows:

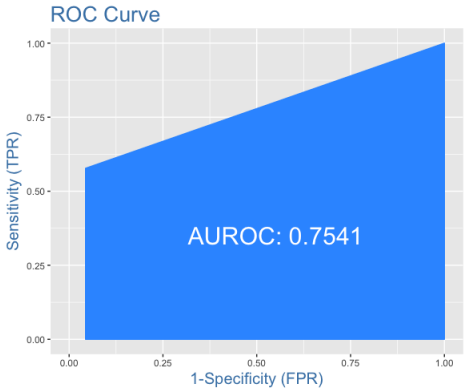


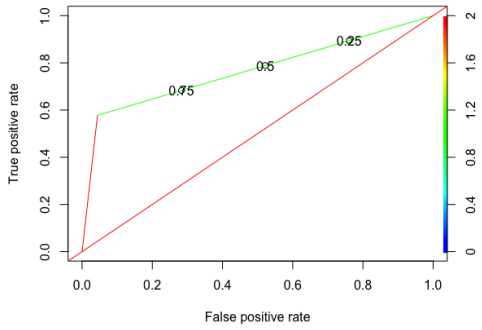


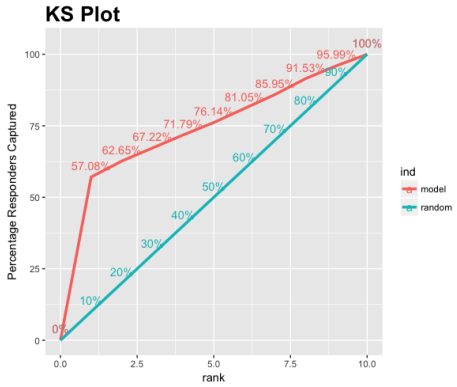


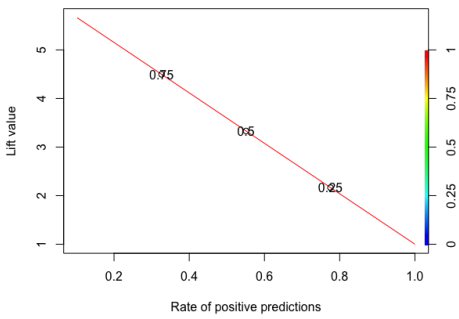


Elastic Net Regularization Model

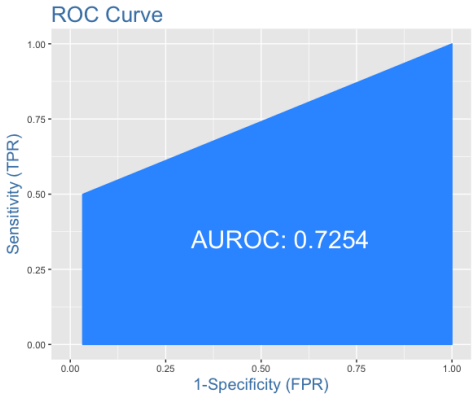


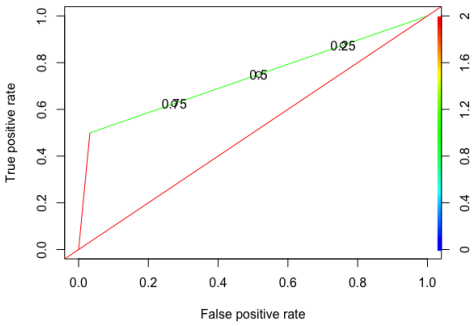


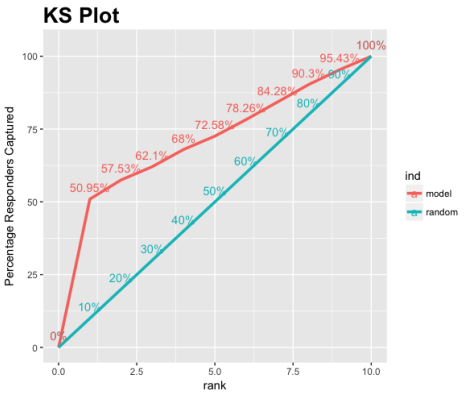


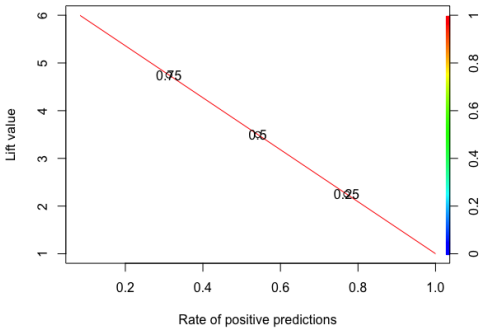


Decision Tree Model

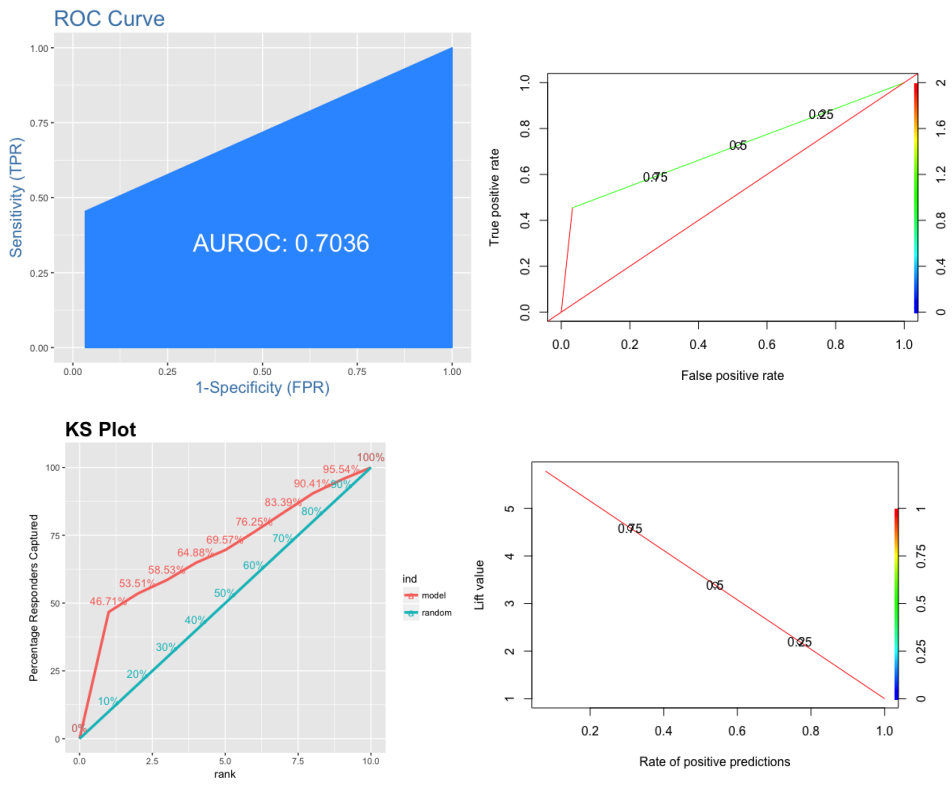








Random Forest Model



Boosting Model

Problem 3 – Part e.

According to the plots that are shown in previous section and the result and parameter from CPE function which is available in the attached R script, decision tree shows a better prediction based on the test data.

Problem 4

Problem 4 – Part a

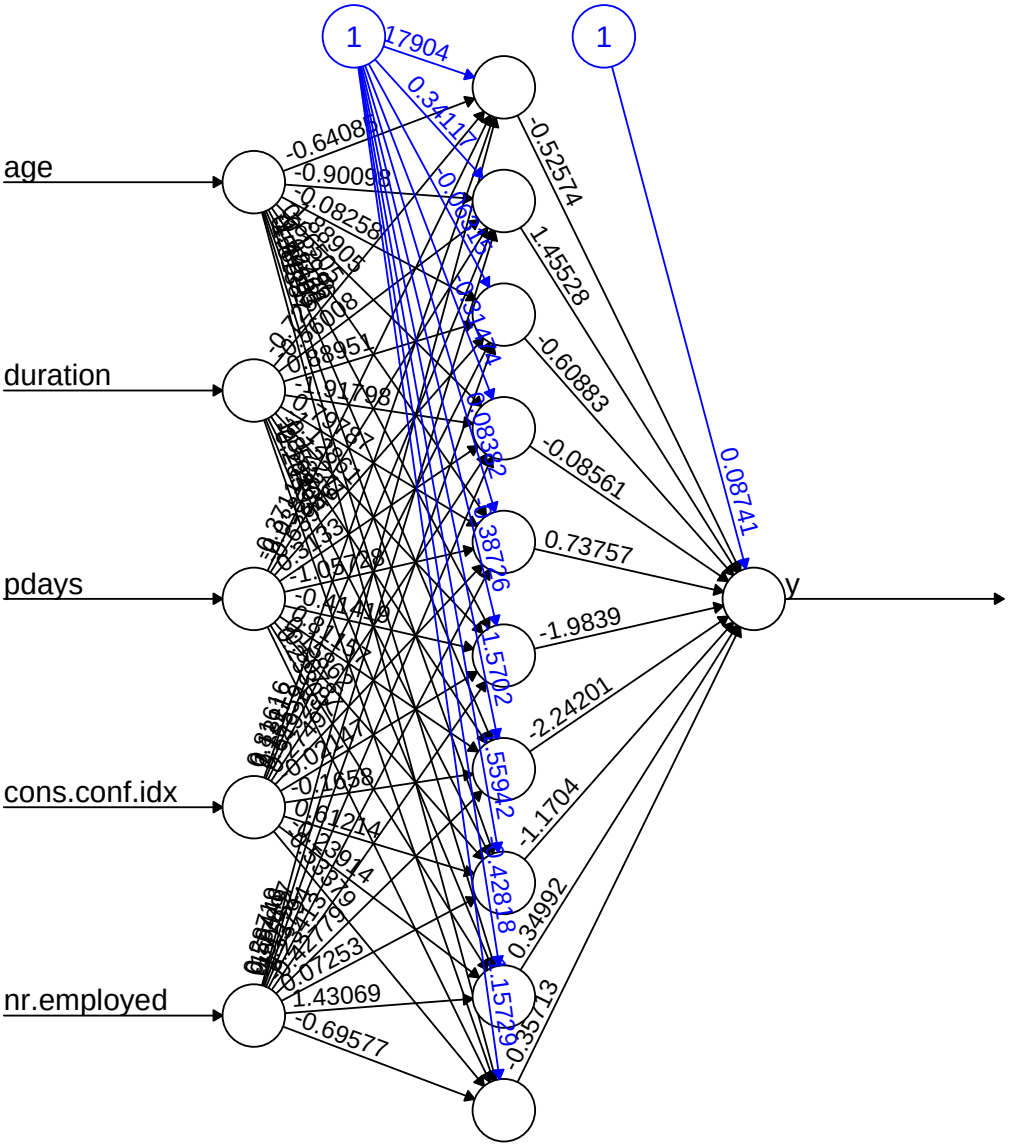
A support vector machine model also was developed. The details are available in the R script file.

fit.SVM <- svm(y ~ ., data = Bank.Training, cost = 100, gamma = 1)

Problem 4 – Part b.

A neural network model was developed based on the same data set. The details are available in the R script file.

```
Bank.NN = Bank.Training[,-(2:8)]
Bank.NN.Test = Bank.Test[,-(2:8)]
Bank.names <- names(Bank.NN)
f <- as.formula(paste("y ~", paste(Bank.names[!Bank.names %in% "y"], collapse = " + ")))
fit.NN <- neuralnet(f, data=Bank.Training, hidden=10, threshold=0.5)
fit.NN.pr <- compute(fit.NN,Bank.NN.Test[,1:5])$net.result
fit.NN.pr[fit.NN.pr >= 0] = 1
fit.NN.pr[fit.NN.pr < 0] = 0
```



Error: 1633.706502 Steps: 5047

In neural network model only numeric variables were used.