KNN model

```
##At first, csv file was modified using tool of text to columns separated by ; and then it was imported.
> library(caret)
> bank <- read.csv("C:/Users/rc_as/Desktop/Data_science/R_Training/Assignment/bank-additional-full.csv",h
eader=TRUE)
> anyNA(bank)
[1] FALSE
> bank = na.omit(bank)
> bank$age = as.numeric(bank$age) #changed following CASE STUDY file
> bank$duration = as.numeric(bank$duration)
> bank$campaign= as.numeric(bank$campaign)
> bank$pdays= as.numeric(bank$pdays)
> bank$previous= as.numeric(bank$previous)
> index = createDataPartition(y = bank$y, p = 0.7, list = FALSE)
> bank_train = bank[index, ]
> bank_test = bank[-index, ]
> trctrl = trainControl(method = "repeatedcv", number = 10, repeats = 3)
> knn_fit = train(y ~ ., data = bank_train, method = "knn",trControl = trctrl, preProcess = c("center", "
scale"),tuneLength = 10)
> knn_fit
k-Nearest Neighbors
28832 samples
   20 predictor
    2 classes: 'no', 'yes'
  k   Accuracy   Kappa
   5  0.8951167  0.3570606
   7  0.8971514  0.3501787
   9  0.8977873  0.3437551
  11  0.8982844  0.3378774
  13  0.8989666  0.3347738
  15  0.8993943  0.3323694
  17  0.8993712  0.3282199
  19  0.8992555  0.3231446
  21  0.8999261  0.3219199
  23  0.8999145  0.3190244

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 21.
> table(bank_train$y)/ length(bank_train$y)
       no       yes
0.8873474 0.1126526
> knn_test = predict(knn_fit, newdata = bank_test)
> confusionMatrix(knn_test, bank_test$y)
Confusion Matrix and Statistics
          Reference
Prediction    no   yes
       no  10775  1045
       yes   189   347
               Accuracy : 0.9001
                 95% CI : (0.8947, 0.9054)
            Sensitivity : 0.9828
            Specificity : 0.2493
```

Logistic regression

```
> log_fit = train(y ~ ., data = bank_train, trControl = trctrl, method = "glm",family = "binomial")
> bank_test$pred = predict(log_fit, newdata = bank_test)
> confusionMatrix(bank_test$pred, bank_test$y)
Confusion Matrix and Statistics
          Reference
Prediction    no   yes
       no  10658   842
       yes   306   550
               Accuracy : 0.9071
                 95% CI : (0.9018, 0.9122)
            Sensitivity : 0.9721
            Specificity : 0.3951
```

> #taking only significant variable manually

```
> log_fit1 = train(y ~ contact + month + default + duration + poutcome + poutcome + emp.var.rate+cons.pri
ce.idx, data = bank_train,trControl = trctrl, method = "glm",family = "binomial") #11 vaiables
> summary(log_fit1)
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)         -1.266e+02  6.014e+00 -21.041  < 2e-16 ***
contacttelephone    -3.664e-01  7.371e-02  -4.970 6.69e-07 ***
monthaug             9.716e-01  1.001e-01   9.711  < 2e-16 ***
monthdec             4.674e-01  2.245e-01   2.082 0.037323 *
monthjul             4.193e-01  1.023e-01   4.097 4.18e-05 ***
monthjun            -8.781e-02  1.022e-01  -0.859 0.390232
monthmar             1.870e+00  1.353e-01  13.827  < 2e-16 ***
monthmay            -5.752e-01  8.650e-02  -6.650 2.93e-11 ***
monthnov            -9.201e-02  1.056e-01  -0.872 0.383441
monthoct             4.865e-01  1.286e-01   3.783 0.000155 ***
monthsep             4.838e-01  1.373e-01   3.522 0.000428 ***
defaultunknown      -3.180e-01  7.797e-02  -4.078 4.54e-05 ***
defaultyes          -7.318e+00  1.390e+02  -0.053 0.958028
duration             4.662e-03  8.870e-05  52.562  < 2e-16 ***
poutcomenonexistent  4.782e-01  7.413e-02   6.451 1.11e-10 ***
poutcomesuccess      1.892e+00  1.011e-01  18.713  < 2e-16 ***
emp.var.rate        -1.018e+00  2.797e-02 -36.413  < 2e-16 ***
cons.price.idx       1.304e+00  6.405e-02  20.351  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> bank_test$pred1 = predict(log_fit1, newdata = bank_test)
> confusionMatrix(bank_test$pred1, bank_test$y)
Confusion Matrix and Statistics
          Reference
Prediction    no   yes
       no  10671   851
       yes   293   541
               Accuracy : 0.9074
                 95% CI : (0.9022, 0.9125)

            Sensitivity : 0.9733
            Specificity : 0.3886
```

```
> #Auto selection of significant variables
> library(MASS)
> model_logistic <- glm(y~., data=bank_train, family = "binomial")
> summary(model_logistic)
Call:
glm(formula = y ~ ., family = "binomial", data = bank_train)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.9281  -0.2990  -0.1849  -0.1345   3.1473

Coefficients: (1 not defined because of singularities)
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -2.947e+02  4.562e+01  -6.460 1.05e-10 ***
age                          1.872e-04  2.932e-03   0.064 0.949090
jobblue-collar              -1.853e-01  9.441e-02  -1.963 0.049641 *
jobentrepreneur             -1.566e-01  1.551e-01  -1.010 0.312636
jobhousemaid                -9.233e-02  1.846e-01  -0.500 0.616981
jobmanagement               -8.296e-02  1.045e-01  -0.794 0.427410
jobretired                   3.007e-01  1.287e-01   2.336 0.019498 *
jobself-employed            -1.300e-01  1.436e-01  -0.905 0.365435
jobservices                 -9.516e-02  1.011e-01  -0.941 0.346726
jobstudent                   2.403e-01  1.325e-01   1.813 0.069821 .
jobtechnician               -1.144e-02  8.632e-02  -0.133 0.894584
jobunemployed                1.833e-01  1.488e-01   1.232 0.217938
jobunknown                  -2.991e-01  3.011e-01  -0.993 0.320603
maritalmarried               3.695e-02  8.275e-02   0.447 0.655204
maritalsingle                5.522e-02  9.430e-02   0.586 0.558157
maritalunknown              -1.240e-01  5.910e-01  -0.210 0.833761
educationbasic.6y            2.261e-01  1.423e-01   1.589 0.112127
educationbasic.9y            8.037e-02  1.139e-01   0.706 0.480413
educationhigh.school         1.335e-01  1.107e-01   1.206 0.227833
educationilliterate          1.111e+00  9.743e-01   1.141 0.254075
educationprofessional.course 1.230e-01  1.234e-01   0.997 0.318893
educationuniversity.degree   2.261e-01  1.114e-01   2.030 0.042335 *
educationunknown             3.090e-01  1.437e-01   2.150 0.031547 *
defaultunknown              -2.873e-01  8.045e-02  -3.572 0.000355 ***
defaultyes                  -7.292e+00  1.391e+02  -0.052 0.958189
housingunknown              -7.796e-02  1.682e-01  -0.463 0.643023
housingyes                  -4.364e-02  4.961e-02  -0.880 0.379108
loanunknown                         NA         NA      NA       NA
loanyes                     -1.713e-02  6.823e-02  -0.251 0.801704
contacttelephone            -7.265e-01  9.345e-02  -7.774 7.61e-15 ***
monthaug                     8.974e-01  1.436e-01   6.250 4.11e-10 ***
monthdec                     2.387e-01  2.523e-01   0.946 0.344080
monthjul                     7.128e-02  1.158e-01   0.615 0.538351
monthjun                    -7.881e-01  1.506e-01  -5.234 1.66e-07 ***
monthmar                     2.166e+00  1.735e-01  12.481  < 2e-16 ***
monthmay                    -4.578e-01  9.873e-02  -4.636 3.55e-06 ***
monthnov                    -4.675e-01  1.449e-01  -3.227 0.001252 **
monthoct                     1.972e-01  1.836e-01   1.074 0.282600
monthsep                     4.913e-01  2.136e-01   2.300 0.021444 *
day_of_weekmon              -1.013e-01  7.922e-02  -1.279 0.200776
```

```
day_of_weekthu                5.727e-02  7.700e-02   0.744 0.456989
day_of_weektue                7.782e-02  7.929e-02   0.981 0.326391
day_of_weekwed                1.914e-01  7.889e-02   2.426 0.015280 *
duration                      4.667e-03  8.928e-05  52.277  < 2e-16 ***
campaign                     -3.577e-02  1.385e-02  -2.582 0.009812 **
pdays                        -4.094e-02  2.050e-02  -1.996 0.045883 *
previous                     -1.060e-01  7.173e-02  -1.477 0.139626
poutcomenonexistent           4.054e-01  1.147e-01   3.533 0.000410 ***
poutcomesuccess               8.304e-01  2.646e-01   3.138 0.001700 **
emp.var.rate                 -1.924e+00  1.688e-01 -11.400  < 2e-16 ***
cons.price.idx                2.605e+00  3.005e-01   8.670  < 2e-16 ***
cons.conf.idx                 2.968e-02  9.368e-03   3.168 0.001536 **
euribor3m                     2.430e-01  1.558e-01   1.560 0.118831
nr.employed                   9.357e-03  3.714e-03   2.519 0.011757 *
pdaysDummy1                   3.953e+01  2.026e+01   1.951 0.051069 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> logistic <- stepAIC(model_logistic, trace=FALSE, direction="backward")
> summary(logistic)
Call:
glm(formula = y ~ job + default + contact + month + day_of_week +
    duration + campaign + pdays + previous + poutcome + emp.var.rate +
    cons.price.idx + cons.conf.idx + euribor3m + nr.employed +
    pdaysDummy, family = "binomial", data = bank_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.9318  -0.2994  -0.1848  -0.1350   3.1252

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)         -2.947e+02  4.550e+01  -6.477 9.36e-11 ***
jobblue-collar      -2.616e-01  7.771e-02  -3.366 0.000763 ***
jobentrepreneur     -1.836e-01  1.534e-01  -1.197 0.231466
jobhousemaid        -1.817e-01  1.777e-01  -1.023 0.306477
jobmanagement       -6.695e-02  1.021e-01  -0.656 0.512067
jobretired           2.194e-01  1.013e-01   2.167 0.030212 *
jobself-employed    -1.353e-01  1.425e-01  -0.949 0.342550
jobservices         -1.367e-01  9.580e-02  -1.427 0.153616
jobstudent           2.353e-01  1.207e-01   1.949 0.051302 .
jobtechnician       -4.093e-02  7.672e-02  -0.533 0.593699
jobunemployed        1.236e-01  1.466e-01   0.843 0.399378
jobunknown          -2.911e-01  2.974e-01  -0.979 0.327717
defaultunknown      -2.930e-01  7.929e-02  -3.696 0.000219 ***
defaultyes          -7.328e+00  1.390e+02  -0.053 0.957970
contacttelephone    -7.252e-01  9.330e-02  -7.773 7.64e-15 ***
monthaug             9.097e-01  1.434e-01   6.345 2.23e-10 ***
monthdec             2.380e-01  2.520e-01   0.944 0.345104
monthjul             7.923e-02  1.154e-01   0.687 0.492205
monthjun            -7.791e-01  1.499e-01  -5.199 2.00e-07 ***
monthmar             2.169e+00  1.733e-01  12.519  < 2e-16 ***
monthmay            -4.606e-01  9.847e-02  -4.677 2.90e-06 ***
```

```
monthnov           -4.653e-01  1.447e-01  -3.216 0.001301 **
monthoct            1.881e-01  1.833e-01   1.026 0.304791
monthsep            4.982e-01  2.134e-01   2.335 0.019568 *
day_of_weekmon     -1.037e-01  7.911e-02  -1.310 0.190057
day_of_weekthu      5.715e-02  7.691e-02   0.743 0.457432
day_of_weektue      7.302e-02  7.918e-02   0.922 0.356433
day_of_weekwed      1.900e-01  7.885e-02   2.409 0.015999 *
duration            4.665e-03  8.917e-05  52.316  < 2e-16 ***
campaign           -3.531e-02  1.382e-02  -2.555 0.010623 *
pdays              -4.088e-02  2.046e-02  -1.998 0.045716 *
previous           -1.041e-01  7.155e-02  -1.455 0.145643
poutcomenonexistent  4.055e-01  1.146e-01   3.539 0.000401 ***
poutcomesuccess     8.382e-01  2.644e-01   3.170 0.001523 **
emp.var.rate       -1.927e+00  1.686e-01 -11.429  < 2e-16 ***
cons.price.idx      2.609e+00  2.998e-01   8.702  < 2e-16 ***
cons.conf.idx       3.031e-02  9.312e-03   3.255 0.001133 **
euribor3m           2.432e-01  1.554e-01   1.564 0.117733
nr.employed         9.350e-03  3.703e-03   2.525 0.011578 *
pdaysDummy1         3.949e+01  2.022e+01   1.953 0.050827 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> pred_logistic <- predict.glm(logistic, newdata = bank_test, type="response")
> le = levels(bank_test$y)
> pred_class <- ifelse(pred_logistic>0.5, le[2], le[1])
> pred_class <- as.factor(pred_class)
> confusionMatrix(pred_class, bank_test$y)
Confusion Matrix and Statistics
          Reference
Prediction    no   yes
       no  10663   835
       yes   301   557
               Accuracy : 0.9081
            Sensitivity : 0.9725
            Specificity : 0.4001
```

Linear SVM model

```
> svm_train = train(y ~ ., data = bank_train, method = "svmLinear",trControl = trctrl, tuneLength = 10)
> svm_train
  Accuracy   Kappa
  0.9055448  0.3954498
> svm_test = predict(svm_train, newdata = bank_test)
> confusionMatrix(svm_test , bank_test$y)
Confusion Matrix and Statistics
          Reference
Prediction    no   yes
       no  10709   932
       yes   255   460
               Accuracy : 0.9039
                 95% CI : (0.8986, 0.9091)
            Sensitivity : 0.9767
            Specificity : 0.3305
         Pos Pred Value : 0.9199
```

NON LINEAR SVM (svmRadial or svmPoly)

```
> svm_rad = train(y ~ ., data = bank_train, method = "svmRadial",trControl = trctrl, tuneLength = 10)
> rad_test = predict(svm_rad, newdata = bank_test)
> confusionMatrix(rad_test , bank_test$y)
Confusion Matrix and Statistics
          Reference
Prediction    no   yes
       no  10699   881
       yes   265   511

               Accuracy : 0.9073
                 95% CI : (0.902, 0.9123)
            Sensitivity : 0.9758
            Specificity : 0.3671
```

Decision Tree

```
> tree_fit = train(y ~ ., data = bank_train, method = "rpart",parms = list(split = "gini"), trControl = t
rctrl,tuneLength = 10)
> predict_test = predict(tree_fit, newdata = bank_test)
> confusionMatrix(predict_test, bank_test$y)
Confusion Matrix and Statistics
          Reference
Prediction    no   yes
       no  10588   670
       yes   376   722
               Accuracy : 0.9153
                 95% CI : (0.9103, 0.9202)
            Sensitivity : 0.9657
            Specificity : 0.5187
```

Random Forest

```
> rforest = randomForest(y ~ . , data = bank_train)

> pred_forest = predict(rforest, newdata = bank_test)
> confusionMatrix(pred_forest, bank_test$y)
Confusion Matrix and Statistics
          Reference
Prediction    no   yes
       no  10574   651
       yes   390   741

               Accuracy : 0.9157
                 95% CI : (0.9107, 0.9206)
            Sensitivity : 0.9644
            Specificity : 0.5323
> varImp(tree_fit)
  only 20 most important variables shown (out of 54)
                             Overall
duration                    100.0000
euribor3m                    47.9276
nr.employed                  43.7329
pdays                        42.2605
poutcomesuccess              40.2001
cons.conf.idx                10.6716
```

```
emp.var.rate                    8.6870
cons.price.idx                  8.5421
monthmar                        4.4043
previous                        2.5295
contacttelephone                2.3897
poutcomenonexistent             1.9066
monthoct                        1.2547
day_of_weekthu                  0.5473
monthmay                        0.4432
age                             0.4317
day_of_weekmon                  0.2772
educationprofessional.course    0.2770
educationbasic.9y               0.2769
campaign                        0.2454
> important = as.data.frame(importance(rforest))
> important$var = rownames(important)
> important$var
 [1] "age"            "job"            "marital"        "education"      "default"        "housing"
"loan"           "contact"
 [9] "month"          "day_of_week"    "duration"       "campaign"       "pdays"          "previous"
"poutcome"       "emp.var.rate"
[17] "cons.price.idx" "cons.conf.idx"  "euribor3m"      "nr.employed"
> #important = important[, c(2,1)]
> important1 = arrange(important, desc(MeanDecreaseGini))
> View(important1)
> most_imp = important1[1:4, ]
> most_imp
  MeanDecreaseGini       var
1        1689.8571  duration
2         574.9418 euribor3m
3         415.9445       age
4         360.3558       job
> rforest1 = randomForest(y ~ duration + euribor3m + age + job, data = bank_train)
> #prediction and confusion matrix
> pred_2 = predict(rforest1, newdata = bank_test)
> confusionMatrix(pred_2, bank_test$y)
Confusion Matrix and Statistics
          Reference
Prediction    no   yes
       no  10455   674
       yes   509   718
               Accuracy : 0.9043
                 95% CI : (0.8989, 0.9094)
            Sensitivity: 0.9536
            Specificity: 0.5158
```

With 8 variables

```
> most_imp = important1[1:8, ]
> most_imp
  MeanDecreaseGini         var
1        1698.3344    duration
2         563.2928   euribor3m
3         417.0865         age
```

```
4         359.7723         job
5         330.8883 nr.employed
6         267.4001   education
7         244.4238 day_of_week
8         209.0730       pdays
> rforest1 = randomForest(y ~ duration + euribor3m + age + job + nr.employed + education + day_of_week + 
pdays, data = bank_train)
> #prediction and confusion matrix
> pred_2 = predict(rforest1, newdata = bank_test)
> confusionMatrix(pred_2, bank_test$y)
Confusion Matrix and Statistics
          Reference
Prediction    no   yes
       no  10554   671
       yes   410   721
               Accuracy : 0.9125
                 95% CI : (0.9074, 0.9174)
            Sensitivity : 0.9626
            Specificity : 0.5180
```

With 6 variables

```
> rforest1 = randomForest(y ~ duration + euribor3m + age + job + nr.employed + education, data = bank_tra
in)
> #prediction and confusion matrix
> pred_2 = predict(rforest1, newdata = bank_test)
> confusionMatrix(pred_2, bank_test$y)
Confusion Matrix and Statistics
          Reference
Prediction    no   yes
       no  10475   668
       yes   489   724
               Accuracy : 0.9064
                 95% CI : (0.9011, 0.9114)
            Sensitivity : 0.9554
            Specificity : 0.5201
```

Accuracy of RF was found to increase with the increase of predicting variables.

Comparison of all the models:

Model	Accuracy (%)	Sensitivity (%)	Specificity (%)
KNN	90.01	98.28	24.93
LOGISTIC REG.	90.91	97.16	41.67
LINEAR SVM	90.39	97.67	33.05
NON-LIN. SVM	90.73	97.58	36.71
DECISION TREE	91.53	96.57	51.87
RANDOM FOREST	91.57	96.44	53.23
RF with 4 vars.	90.43	95.36	51.58
RF with 6 vars.	90.64	95.54	52.01
RF with 8 vars.	91.25	96.26	51.80

Accuracy order from highest to lowest:

Random Forest, Decision Tree, Logistic Regression, Non-Linear SVM, Linear SVM

Conclusion: For the given dataset, random Forest is the best model with highest accuracy.