Challenge Report

Mozhdeh, Naghmeh, Rozita, Samim

January 2015

1 Data Pre-processing

We took the first 5000 records of the original data as a CSV file and read it into R.

```
ds = read.csv("new_train.csv")
ds[] <- lapply(ds, factor)
target <- "click"
```

After some inspection we found that we can ignore some attributes. `hour` attribute is constant in this portion of data.

```
summary(ds$hour)

:     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
: 14100000 14100000 14100000 14100000 14100000 14100000
```

It makes sense that `id`, `site_id`, `site_domain` and `device_model` do not give any useful information. Also `device_ip` has so many different values.

```
str(ds$device_ip)

:  Factor w/ 4016 levels "0007786f","000dcf99",..: 3451 2361 2824 ...
```

```
str(ds$device_model)

:  Factor w/ 841 levels "00b08597","00b1f3a7",..: 232 353 432 ...
```

So we ignored these attributes and created the test and train datasets.

```
ignore <- c( "hour",
             "id",
             "site_id",
             "site_domain",
             "device_ip",
             "device_id",
             "device_model") # Coloumns to ignore
vars <- setdiff(names(ds), ignore)
inputs <- setdiff(vars, target)
form <- formula(paste(target, "~ ."))
nobs <- nrow(ds)
train <- sample(nobs, 0.7*nobs)
test <- setdiff(seq_len(nobs), train)
actual <- ds[test, target]
data <- ds[train,vars]
```

Here is the structure of the data after data pre-processing.

```
str(data)

'data.frame':   3500 obs. of  17 variables:
$ click           : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 1 1 2 2 ...
$ C1              : Factor w/ 5 levels "1001","1002",..: 3 3 3 3 3 3 3 ...
$ banner_pos      : Factor w/ 3 levels "0","1","4": 2 2 1 1 1 1 1 1 1 2 ...
$ site_category   : Factor w/ 14 levels "0569f928","110ab22d",..: 13 ...
$ app_id          : Factor w/ 210 levels "00848fac","03528b27",..: 197 19 ...
$ app_domain      : Factor w/ 22 levels "0654b444","18eb4e75",..: 10 10 ...
$ app_category    : Factor w/ 10 levels "07d7df22","09481d60",..: 1 1 1 ...
$ device_type     : Factor w/ 4 levels "0","1","4","5": 2 2 2 2 2 2 ...
$ device_conn_type: Factor w/ 4 levels "0","2","3","5": 1 1 1 2 1 1 ...
$ C14             : Factor w/ 216 levels "375","377","380",..: 71 47 ...
$ C15             : Factor w/ 4 levels "216","300","320",..: 3 3 3 3 ...
$ C16             : Factor w/ 5 levels "36","50","90",..: 2 2 2 2 ...
$ C17             : Factor w/ 104 levels "112","122","153",..: 42 26 5...
$ C18             : Factor w/ 4 levels "0","1","2","3": 3 1 3 4 1 1 ...
$ C19             : Factor w/ 32 levels "35","39","41",..: 2 1 2 2 ...
$ C20             : Factor w/ 91 levels "-1","100000",..: 1 1 1 1 3...
$ C21             : Factor w/ 28 levels "13","15","16",..: 3 16 6 5 1...
```

Let us explore our data a little. Displaying distribution of data based on $\text{site}_{\text{category}}$ for all data and clicked data. For both all data and clicked data major site category is 28905ebd:

```
table(ds$site_category)

0569f928 110ab22d 28905ebd 335d28a8 3e814130 50e219e0 72722551 75fa27f6
      35        1     1909       57      604     1244       12       11
76b2941d a818d37a bcf865d9 c0dd3be3 f028772b f66779e6
     116        1        1        3      994       12
```

Displaying distribution of data based on $\text{app}_{\text{category}}$ for all data and clicked data. For both all data and clicked data major app category is 07d7df22:

```
table(ds$app_category)

 07d7df22 09481d60 0f2161f8 4ce2e9fc 75d80bbe 8ded1f7a cef3e649 d1327cf5
     3955        1      751        4        6       66       70        5
 f95efa07 fc6fa53d
      141        1
```

2 SVM

First we can use the `tune` function to determine our constants in using SVM.

```
library(e1071)
tuned <- tune.svm(form, data = data, gamma = 10^(-6:-1), cost = 10^(1:2))
summary(tuned)

Parameter tuning of svm:

- sampling method: 10-fold cross validation

- best parameters:
 gamma cost
 1e-06   10
```

Using the constants above we can train our model.

```
model  <- svm(form, data = data, gamma = 10^(-6:-1), cost = 10)
```

Here is the confusion matrix of our model

```
svmPred <- predict(model, ds[test,vars])
tab <- table(pred = svmPred, true = ds[test,target])
print(tab)

     true
 pred    0    1
    0 1244  256
    1    0    0
```

3 Naïve Bayes

```
library(e1071)
classifier <- naiveBayes(data[train, vars], data[train, target])
table(predict(classifier, data[test, vars]), data[test, target])

       0   1
   0 861   5
   1  15 178
```

4 kNN

```
library(RWeka)
classifier <- IBk(form, data = data, control = Weka_control(K = 2, X = TRUE))
evaluate_Weka_classifier(classifier, numFolds = 10)

=== 10 Fold Cross Validation ===

=== Summary ===

Correctly Classified Instances        2866               81.8857 %
Incorrectly Classified Instances       634               18.1143 %
Kappa statistic                          0.0876
Mean absolute error                      0.2578
Root mean squared error                  0.379
Relative absolute error                 90.5826 %
Root relative squared error            100.4847 %
Coverage of cases (0.95 level)          96.9429 %
Mean rel. region size (0.95 level)      85.3143 %
Total Number of Instances             3500

=== Confusion Matrix ===

   a    b   <-- classified as
2811   88 |    a = 0
 546   55 |    b = 1
```

5 Conclusion