**Eva Lipska**

**task 2, course 2**

**Customer brand Preferences Report**

**Goal of the project:**

Prediction of the customer brand preferences through the use of the two classifiers:

*K-Nearest Neighbors (KNN)* and *Random Forest (RF)*

**Execution:**

As the type of the predicted data is nominal, some of the variables required transformation into factors: brand, elevel, zipcode, car.

Dataset was divided into two sets: training (75% of all observations, which is 7501) and testing sets (25% of all observations – 2499). Both datasets contain 7 variables – salary, age, elevel, car, zipcode, credit and brand.

Cross-validation technique (10 folds, 3 repeats) was used for both classifiers in order to:

- test KNN algorithm with different values of K;

- test RF algorithm with different amount of mtry (mtry: number of variables randomly sampled as candidates at each split)

**Conducting two classification methods – KNN & RF**

The prediction of the customer brand preferences is based on a model that presents the highest accuracy.

Random Forest algorithm presents the highest accuracy, after parameters adjustment.

**Random Forest**

The highest accuracy obtained within the RF model – 92 %.

The same accuracy (92 %) was obtained in the testing set.

*Random Forest - tuning algorithm parameters:*

The search was executed with the use of the three searching for mtry strategies: default, random and grid.

Random search presented the best results: 92 % of accuracy.

1. **search = “random”**

accuracy: 92%

mtry = 13

ctrl2 <- trainControl(method = "repeatedcv", number = 10, repeats = 3, verboseIter = TRUE, search = "random")

rf\_random <- train(brand~., data = training, method = "rf", metric = "Accuracy", trControl = ctrl2, tuneLength = 15)

rf\_random

1. **default parameters**

accuracy: 72%

Tuning parameter “mtry” was held constant at a value of 2.645751 (which is a squared number of columns in dataset).

ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3, verboseIter = TRUE)

mtry <- sqrt(ncol(training))

tunegrid <- expand.grid(.mtry = mtry)

rf\_default <- train(brand~., data = training, method = "rf", metric = "Accuracy", tuneGrid = tunegrid, trControl = ctrl)

rf\_default

1. **search = “grid”**

accuracy 90%

mtry = 6

ctrl3 <- trainControl(method = "repeatedcv", number = 10, repeats = 3, verboseIter = TRUE, search = "grid")

mtry <- sqrt(ncol(training))

tunegrid <- expand.grid(.mtry = c(1:6))

rf\_grid <- train(brand~., data = training, method = "rf", metric = "Accuracy", tuneGrid = tunegrid, trControl = ctrl3)

rf\_grid

**K-Nearest Neighbors**

**The highest level of accuracy obtained within the KNN model was 69 %**

***K-Nearest Neighbors - tuning algorithm parameters:***

Accuracy of KNN was determined by the adjustment of the two parameters:

- data preprocessing (preProc) which improve the numerical stability of the model and

- tuneLength.

Data preprocessing was related to the choice between two options:

- normalization (preProc = “range”) – which scales all numeric variables in the range 0 – 1;

- standardization (preProc = c(“center”, “scale”) – centering is reducing the mean value of samples from all observations (mean value of 0). Scaling is dividing value of predictor for each observation by standard deviation of all samples. This will cause the transformed values to have a standard deviation of 1.

Better accuracy was obtained through the use of normalization: preProc = (“range”)

No outliers in the dataset were observed, which could be a case when normalizing method is used (the “normal” data would be scaled to a very small interval).

Best performance of KNN – accuracy 69 %, k = 57 – was obtained with:

preProc = “range”

tuneLength = 50

Code:

ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3, verboseIter = TRUE)

knn <- train(brand ~., data = training, method = "knn", trControl = ctrl, tuneLength = 50, preProc = "range")

knn

**Prediction of the customer brand preferences for the incomplete survey responses**

*The result of the RF classifier conducted on the testing set:*

New predictive model assessment: postResample()

Accuracy: 92%

*Final prediction:*

Data: incomplete dataset - "SurveyIncomplete.csv"

Acer: 1884

Sony: 3116

Chart presenting predicted customer brand preferences:

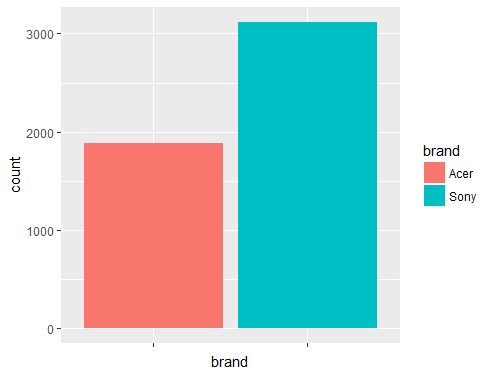


Chart presenting existing customer brand preferences:

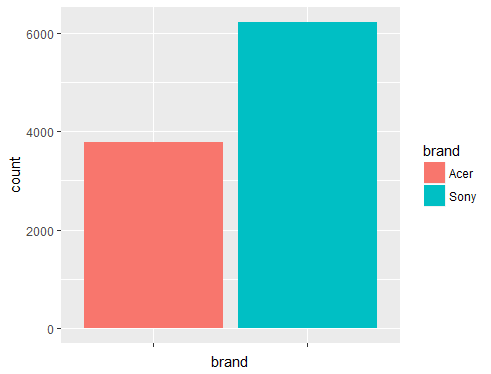
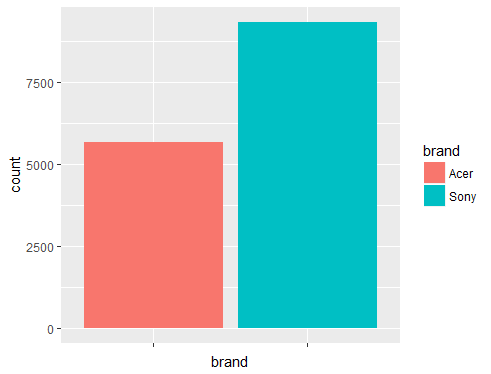


Chart presenting total customer brand preferences:



Acer: 5667

Sony: 9333