Predicting Household Composition by TV Viewing Behavior

Diploma of Advanced Studies in Applied Statistics at ETH Zurich

Rafael Lüchinger

30 April 2019

Contents

1 Summary **1**

2 Introduction **2**

3 TV Viewing Data **2**

4 Target and Feature **3**
4.1 Target: Household Composition . . . 3
4.2 Generating Features of Viewing Behavior . . . 3

5 Data Exploration **5**
5.1 Classification versus Regression . . . 5
5.2 Log Transformation . . . 5
5.3 Visualization of Features . . . 6
5.4 Dimensionality Reduction . . . 6
5.5 Probability to Correctly Classify by Chance . . . 6
5.6 Partitioning into Train and Test Data . . . 7

6 Model Specification **7**
6.1 Multinomial Logistic Regression . . . 7
6.2 Support Vector Machine . . . 8
6.3 Random Forest . . . 9

7 Results **9**
7.1 Performance . . . 9
7.2 Variable Importance . . . 10
7.3 Partial Dependence . . . 11

8 Discussion **11**

9 Appendix **13**

1 Summary

In this study television (TV) viewing data from individuals was aggregated to household level to test if the number of individuals in the household can be predicted based on the households TV viewing data. This situation simulates one particular obstacle when merging RPD data with traditional panel data. For each home the average daily viewing duration across an 8 week period was extracted and 53 different features of viewing behavior were derived such as daytime, weekday, type of channel or program genre. The sample of 2006 homes were split by a ratio of 6:4 and used for training and cross validating, respectively. Three different machine learning algorithms were used to compete each other for best prediction of household size. The results showed that all 3 classifier, multinomial linear regression, support vector machine and random forest,

performed similarly poor on test data with an overall accuracy of 42% to 44%, given a baseline accuracy of 27%. Separating small (household size 1 and 2) versus bigger homes (household size 3, 4 and 5) was the main pattern found and features like viewing on kids channel and kids programs among others was found to be particularly important. We conclude that TV viewing behavior is not informative enough to reliably derive the number of individuals living in a household.

2 Introduction

TV audience in Switzerland is measured by Mediapulse AG. A representative panel of roughly 2000 households is constantly under measurement. These homes were carefully selected by a complex sampling design and all household members have agreed to be part of the study. The TV viewing of each household member is individually recorded. For each home and individual detailed demographics are surveyed which allows the market to target TV audiences by relevant characteristics like age gender and many more.

A disadvantage of the panel approach is its poor granularity. Especially for small TV channels or unpopular daytimes chances are high that no person in the panel was watching TV. Such zero-ratings are a problem for analyzing program success, optimizing program schedules or prizing commercial time. Most likely, a few people in the Swiss population (~3.5 million households) are always watching even small channels at unpopular times of the day. A panel of 2000 homes is just not sensitive enough to capture all TV viewing behavior. This is not a bias of the system but poor resolution. Increasing the panel size is very expensive and inefficient.

A solution to this problem could be the inclusion of third party data. Set-Top-Boxes (STB) of TV-provider (Swisscom, UPC, etc.) are automatically recording the TV consumption in millions of homes in Switzerland every day and returned and stored on the providers servers (return path data, RPD). There are still many issues with these data that are currently addressed. One issue of RPD is that the viewing is recorded on household level, not on individual level like the Mediapulse TV-panel. Also almost no demographic information is surveyed from those homes. Household-level data is of little use to the market because it gives no insight in target groups based on age and gender and alike.

To integrate RPD data into panel data means to split up household viewing statements to individual viewing statements and assign demographics variables like age and sex. A first step to investigate the feasibility of such data fusion is to predict the number of individuals in a household based on the only available information, the household's TV viewing data.

In this study we will use Mediapulse TV-panel data to investigate to predict the household size by TV viewing behavior. First the panel data is aggregated from individual level to household level. For each household and a 8 week period different features of TV viewing behavior is extracted. Different supervised machine learning algorithms are competing for best prediction of the household size. The true household size for each home of the Mediapulse TV-panel is of course known.

3 TV Viewing Data

Mediapulse TV viewing raw-data comes in the form of daily text files. There are three types of files:

1. `dem`: all individuals with their demographics and daily weights
2. `view`: the TV viewing (live and time-shifted viewing)
3. `prog`: the program timetable with genre information

A commercial software allows to analyse this data via an easy to use software tool. The output of this software is the official Swiss TV audience measure published by Mediapulse AG. A in-house developed R-package allows to read and analyse the raw-data and output the very same results (e.g. daily estimates, the so called "Facts" like *Reach*, *Rating* or *Share*, etc.). The fact that the results between software and R-package match precisely, guarantees that the data processing in R is correct, plus it allows to check if all calculations and aggregations are implemented as intended.

Table 1: TV viewing raw-data (simplified). Reading example: On day `2018-01-01`, in household `2381` individual `1` is watching channel `SRF 1` from `18:04:21` to `18:13:02`. Later that day this person switches to channel `ARTE` and is joined by another household member individual `2`.

day	hh	ind	chn	start	end
2018-01-01	2381	1	SRF 1	18:04:21	18:13:02
2018-01-01	2381	1	ARTE	18:45:20	20:05:45
2018-01-01	2381	2	ARTE	18:45:20	19:45:03

Demographic information is simply joined on keys `day`, `hh` and `ind`. Program schedule is joined via an overlap join on keys `day`, `channel` and `start/end`. If a viewing statement overlaps with multiple programs, the statement gets copied and with `start/end` points cropped to the viewing interval boundaries.

4 Target and Feature

4.1 Target: Household Composition

The aim of this study is to predict the household composition in form of the household size, e.g. the number of people living in the household. Our sample for this study comprises 2006 homes in which a total of 4388 individuals are living in (average household size is 2.19).

The variable *hhsize* is given in the demographics file of the TV raw-data. *hhsize* is not necessarily equal to the sum of individuals for the following reasons:

- children 0-2 years old are not recorded (this is a market convention)
- guest are part of the TV-panel but not counted for household size
- household size is coded 1, 2, 3, 4, 5+, with 5+ meaning households with 5 or more members

Another detail is that household size is not necessarily constant over time. The number of people living in a household may change by natural reasons like birth, death, moving in or out. We use the hhsize of a single sample day and assume the household size is constant for the period of viewing data we are using.

Table 2: Household composition is the sociodemografic profile of a household, for example, household size, and age and gender of the household members. The last 3 of our sample of 2006 households are shown. For this study the target to predict is *hhsize*.

	hh	hhsize	age_1	age_2	age_3	age_4	age_5	sex_1	sex_2	sex_3	sex_4	sex_5
2004	6200	2	63	72	0	0	0	2	1	0	0	0
2005	6201	2	71	73	0	0	0	1	2	0	0	0
2006	6204	4	52	47	17	13	0	1	2	1	2	0

4.2 Generating Features of Viewing Behavior

4.2.1 Selecting 8 Weeks of Viewing Data

For this study, a sample day was fixed, and the viewing data of all panel member present at that day is collected 4 weeks prior and 4 weeks after that date. The sample day is the Sunday `2017-11-12` and comprises 2006 households and 4388 individuals, respectively.

The period of 8 weeks should be long enough to reflect individual viewing behavior. Autumn was chosen because during colder months people are watching more TV than in Summer. This period is free of holidays

or unusual TV events (FIFA World Cup, etc.). Within the 7 * 8 = 56 days, the weekdays Mondays to Sundays are balanced. This is important as TV viewing differs significantly between weekends and workdays (see Figure 1).

4.2.2 TV Viewing on Household Level

The TV raw-data described earlier shows that *Mediapulse TV data* is recorded for each individual. With RPD data however this is not the case. RPD data only provide viewing data on household level. Which person, or how many person are sitting in front of the TV set is unknown. Also, there is no demographic information accompanying RPD data. Here we study if it is possible to predict at least the number of household members if only the home's TV viewing data is available. To this end the *Mediapulse TV data* have to be aggregated form individual level to household level. This means, if more than one person is watching the same content, on household level, this is reflected by a single viewing statement. This household aggregation algorithm is somewhat more complex, but not detailed here.

4.2.3 Feature Generation

Features are a set of variables that are used as input data for Machine Learning classifiers or as predictors for statistical models. In both cases we aim to predict the target variable *hhsize*. To create features of viewing behavior the TV data on household level is summed up for each household and day, by different characteristics. Then, for each household the average across the 56 days was calculated. TV viewing is expressed as the duration of viewing in seconds.

The characteristics underlying the feature generation is guided by industry knowledge and intuition about TV viewing behavior we believe would carry information about the household composition, i.e:

1. Dimension time
    - weekend vs. working days
    - time of the day
2. Dimension content
    - type of channel
    - type of program genre

Figure 1, and Figure 2 in the Appendix are illustrating the effect of the dimension *time* on total TV viewing.

There are over 300 TV channels received in Switzerland. In comparison to other countries this so called *inspill* is substantial. Because of the country's small size and its different linguistic regions there are many channels from neighboring countries watched in Switzerland. For simplification the channels have been mapped to channel groups. There are 3 type of groups: channel type, language, and country of origin.

The program genre is a pre-specified variable in the program schedule files. Each program is categorized to one of the 14 different genres. The TV viewing data is overlapped and split up by the program schedule. The viewing duration is than summed up by each program genre within each household.

Table 3: The sets of features of TV viewing behavior to predict household composition.

weekpart by time of day	channels	programs
day_weekend_02to06	chn_arts	prg_commercial
day_weekend_06to08	chn_generalistprivate	prg_info
day_weekend_08to11	chn_generalistpublic	prg_kids
day_weekend_11to13	chn_kids	prg_missing
day_weekend_13to17	chn_livestileindoor	prg_movie
day_weekend_17to20	chn_livestileoutdoor	prg_music
day_weekend_20to22	chn_local	prg_news
day_weekend_22to24	chn_movieseries	prg_other
day_weekend_24to02	chn_music	prg_series

weekpart by time of day	channels	programs
day_workday_02to06	chn_nature	prg_service
day_workday_06to08	chn_news	prg_show
day_workday_08to11	chn_paytv	prg_sport
day_workday_11to13	chn_religion	prg_talk
day_workday_13to17	chn_sport	prg_trailer
day_workday_17to20	chn_foreign	
day_workday_20to22	chn_swiss	
day_workday_22to24	chn_english	
day_workday_24to02	chn_french	
	chn_german	
	chn_italian	
	chn_other	

Table 4: Final input data set. Shown are the first 6 rows and the first 5 columns. Each of the 2006 households is an observation on rows and identified by the household ID *hh*. The household size, the target to predict, is given in column *hhsize*. All 53 above mentioned features are given in the following columns. The values are the average daily viewing duration in seconds by feature on household level.

hh	hhsize	day_weekend_02to06	day_weekend_06to08	day_weekend_08to11
6	2	0.0000	0.0000	372.8750
9	4	88.3125	20.7500	621.5000
14	2	328.1250	39.2500	12.0000
20	2	1019.6667	555.1333	824.4667
21	2	607.3750	917.2500	3143.5000
35	1	72.0000	357.7500	2303.3125

5 Data Exploration

5.1 Classification versus Regression

Theoretically household size can be interpret both as categorical or as ordinal scale. For this study we prefer the categorical case and therefor apply classification not regression. One reason is that behind a specific household size very different compositions of individuals are possible. For example, a 2-person-household may consist of an elderly couple or a two young students of the same gender or a single parent with one child, etc. Accordingly, the TV viewing behavior in households of the same size may differ significantly given the possible variety of demographics profiles.

5.2 Log Transformation

Screening through the values of the 53 feature variables reveals that often the viewing duration is strongly right skewed and zero inflated (Appendix, Figure 3). That means most households have a relatively low value of TV viewing duration but for a few households the viewing is relatively high. A log transformation makes the data more symmetric. Although in general for Machine Learning algorithms such a transformation is not necessary, it should neither be harmful and we continue with log transformed data. Because of many zero values we use the transformation 'log(x + 1)'.

5.3 Visualization of Features

Before applying statistical methods, we can simply visually explore if there is a predictor variable that separates well the households by household size. Figure 4 in the Appendix shows some examples. For most features there is no such discrimination power apparent. But the amount of viewing kids channels seems to separate small and bigger households.

Another option is to visualize the correlation between feature and between features and the target variable. Figure 5 in the Appendix shows a heat map of the correlation matrix of the input data. The correlation is a standardized measure for the similarity between features. The correlation with the target variable reflects how strong a linear relation ship exists between a particular feature and household sizes 1 to 5. This is only of limited use as we're also interested in any non-linear relation with the five household sizes which we interprets as nominal classes here. The dendrogram attached to the heat map in Figure 5 uses the correlation structure to visualize the between-features association in a hierarchical order.

5.4 Dimensionality Reduction

Fifty three features are not a lot and definitely far less than the 2006 observations. Still, some of these variables may not carry much information or they are redundant to other variables (e.g. highly correlated). For an example of dimensionality reduction Principal Component Analysis (PCA) was chosen. PCA was calculated using the 'prcomp' package with centering but no scaling of the log transformed data matrix. All the features represent the same unit (log of viewing duration n seconds) of comparable magnitude. Figure 6 and 7 in the Appendix shows the results of the PCA.

If PCA finds a few principal components which together explain a substantial portion of the total variance, these synthesized features could used instead of the original high dimensional input data. Then an structure between predictor and target may be described along only 2 or 3 orthogonal axes.

However, Figure 6 shows no such simple decomposition. There is a dominant first component explaining 30% of the variance. Most likely this component reflects the magnitude of overall TV consumption. To reach a cumulative proportion of about 80% more than 10 components are needed. Given the general difficulty of a real-world interpretation of PCs, working with the PCA transformed data is of little use here. In contrast it shows that the features are heterogeneous and therefore well chosen.

Table 5: First 6 PCs. The Variance in the feature matrix is not easily separated in orthogonal components. To reach 80% cumulative explained variance the first 16 PCs would be needed.

	Standard deviation	Proportion of Variance	Cumulative Proportion
PC1	3.982955	0.29932	0.29932
PC2	2.272029	0.09740	0.39672
PC3	1.759215	0.05839	0.45511
PC4	1.517791	0.04347	0.49858
PC5	1.480727	0.04137	0.53995
PC6	1.324283	0.03309	0.57303

5.5 Probability to Correctly Classify by Chance

The probability to correctly classify by chance is 20%, if all five levels of household size were uniformly distributed (naive estimation).

```
x <- replicate(100, sample(1:5, nrow(hh.composition), replace = TRUE))
round(mean(apply(x, 2, function(y) mean(y == hh.composition$hhsize))), 2)
```

```
## [1] 0.2
```

However, we know the probability of each household size level in our sample:

```
p <- prop.table(table(hh.composition$hhsize))
c(hhsize = round(p*100, 2))
```

```
## hhsize.1 hhsize.2 hhsize.3 hhsize.4 hhsize.5
##    34.55    32.70    13.71    13.36     5.68
```

Therefore we can calculate the probability of correct classification by chance more precisely:

```
round(sum(p^2), 2)
```

```
## [1] 0.27
```

The estimation by simulation comes close:

```
x <- replicate(100, sample(1:5, nrow(hh.composition), replace = TRUE, prob = p))
round(mean(apply(x, 2, function(y) mean(y == hh.composition$hhsize))), 2)
```

```
## [1] 0.27
```

5.6 Partitioning into Train and Test Data

The 2006 households are split into two datasets, one for training and one for testing. The training data makes 60% of households and will be used to train the different models. The test data consists of the other 40% of households and will be used for cross validation, .i.e. to test how good the trained models perform when classifying new data.

```
library(caret)
set.seed(999)
train <- caret::createDataPartition(pred.log$hhsize, p = .6, list = FALSE)
d <- list(train = pred.log[train,-1], test = pred.log[-train,-1])
```

A stratified random sampling is used to split the households. Stratification by household size guarantees that the distribution of household size remains the same between train and test data and therefor the cross validation is more reliable.

Table 6: The 2006 household are split randomly into train (60%) and test (40%) data, with stratification by household size.

	train	test	train %	test %
hhsize1	416	277	0.35	0.35
hhsize2	394	262	0.33	0.33
hhsize3	165	110	0.14	0.14
hhsize4	161	107	0.13	0.13
hhsize5	69	45	0.06	0.06

6 Model Specification

6.1 Multinomial Logistic Regression

6.1.1 Full Model

The multinomial logistic regression models the log odds of the 5 household size levels as a linear combination of the predictor variables. We will use the full model with all features when assessing the prediction performance.

```
library(nnet)
m.mnr <- multinom(hhsize ~ ., data = d$train, trace = TRUE, maxit = 500)
```

6.1.2 AIC Optimized Model

In contrast to Machine Learning algorithms (ML), a statistical model describes the relationship between predictors and outcome variable as a mathematical formula. That equation is fitted to the data in the best possible way. However, if the formula does not describe the true relationship the result remains suboptimal. There are many possible ways of specifying a model, including transformation of predictors (non-linear relationship) and specification of interactions between predictors (depending on the state of other predictors). To find the best model in statistical terms, all models have to be fitted and compared against each other. A good model in statistical terms is a model that explains as much variance of the target variable as possible with the smallest number of predictors. The AIC criterion assesses each model by the explained variance subtracted by a punishing factor for each additional model term.

Each model is basically a hypothesis fitted and tested against the data. Running through all possible models to approach the best one is cumbersome and time consuming. If the true relationship between predictors and outcome variable is not among our pre-defined set of models we will not find the optimum and prediction will be sub optimal. But the advantage over ML is that once a satisfying model is found, the relation between predictors and outcome variable follows a mathematical equation. The modelling and prediction is parametric and fully transparent.

We will use an automated stepwise model search to find a good model. For simplicity we only model linear relationships. We will not use this model to compete against ML but to separate relevant from irrelevant predictors.

```
require(MASS)
m.mnr.best <- MASS::stepAIC(mnr)
```

	AIC	Terms
full model	2954.964	53
best model	2854.969	26

A chi-square test between the full model and the reduced model is far from significant. This means that the full model is not better than the reduced model. This is expected from the the AIC stepwise model search.

```
anova(m.mnr.best, m.mnr, test = 'Chisq')
```

6.2 Support Vector Machine

6.2.1 linear

Each household has a vector of 53 feature values. Let's interpret the input data as a 53 dimensional space in which the 2006 households are located. Support Vector Machine (SVM) iteratively constructs hyperplanes that separate the households by household size. It is a binary classifier. With 5 classes SVM runs through all classes, each time separating one against all others. The best separation out of many possible solutions the closest points to the hyperplane build the biggest margin. Usually there is no perfect separation and a few households are located on the wrong side of the hyperplane. Points within the margin and beyond are penalized as further apart they are located (hinge loss). Tuning the error distance with high costs, tends to improve the classification on training data with the risk of overfitting and bad classification on new data, et vice versa.

```
library(e1071)
m.svm.linear <- svm(hhsize ~ ., data = d$train, kernel = "linear", cost = 1)
```

6.2.2 Radial Kernel

Adding non-linearly transformed features to the input data, increases the chances that a well separating hyperplane can be found. In theory, adding enough well transformed features makes any training data

perfectly separable. However such a sophisticated fit to the training data carries a high risk of poorly performing on new data. We use a Gaussian kernel to add transformed features on the fly. In contrast to the non-transformed input data, this would allow to separate households if they were to form clusters by household size at some dimensions in the feature space.

```
m.svm.radial <- svm(hhsize ~ ., data = d$train, kernel = "radial", cost = 15,
                    gamma = 0.01)
```

Optimal values for parameters like cost and gamma were be found by tuning procedures. However what ever is optimized on training data is not necessarily beneficial for prediction accuracy on new data.

6.3 Random Forest

A decision tree is an iterative partitioning of the input data. For each feature the best binary split is found so that in the subsets the occurrence of the of household size classes is as homogeneous as possible. Among a subset of all features the one with the best split is chosen on each node. For each subset the procedure is repeated resulting in a tree like partitioning. For prediction, new data is run through the very same decision rules. The end leafs of the tree represent the final decisions to which class a new case belongs to. The end leaf might not be pure, not allowing a deterministic decision for one or the other household size class, so the majority class from training is taken. Such a single tree models very specific patterns and has a high risk of overfitting.

Random forest averages the results across many trees. For each tree only a random subset of observations is used. This bootstrapping increases bias and reduces variance in each tree, making the average more accurate for predicting new data. The aggregation across trees is only beneficial for accuracy gain if the trees are independent among each other. At each node the splitting variable is chosen from a random subsample of all features, introducing some degree of independence among the trees. Figure 10 shows the out of box (OOB) error rate across the 500 trees that were calculated.

Because the five household size classes are not uniformly distributed we instruct the Random Forest algorithm to use a stratified subsample for each tree.

```
library(randomForest)
m.rf <- randomForest(hhsize ~ ., data = d$train, imortance = TRUE,
                     strata = d$train$hhsize,
                     sampsize = rep(min(table(d$train$hhsize)), 5) * .7
                     )
```

7 Results

7.1 Performance

7.1.1 Accuracy

Accuracy was calculated as the average correctly classified households. Because this formula ignores the different frequencies of household size classes, the accuracy on small households are more influential in this overall score. A score of 1 (or 100%) means perfect match, 0 or (0%) not a single match. A random matching process would generate an accuracy of .27 (or 27%), as outlined before

The accuracy on training set is the model fit. More important is the cross validation with test data. If a trained model performs well on new data this means the rules for classification are of general importance and not restricted to train data. The cross validation on the a test data set also allows to directly compare different models.

Table 8: The Accuracy in train and test dataset for the different classifiers.

	train	test
multinomial	0.56	0.42
randomforest	0.44	0.42
svm.linear	0.57	0.43
svm.radial	0.95	0.44

In all cases accuracy on train data is better than on test data, as expected. On test data all four models perform very similar. Between 42% and 44% of all households in the test data set can be classified correctly. Given an accuracy of 27% by randomly guessing this performance is not good. Apparently it is hard to predict the household size by TV viewing behavior. At least with the input data used here.

An eye-catching result is the accuracy of the support vector machine using a radial kernel. With 95% correct classification this is close to perfection. The creation of non-linear feature space apparently enables SVM to find well separating hyperplanes. However this fine tuned pattern recognition becomes almost useless when applied to new data. The accuracy on test data is not much better compared to the linear SVM.

7.1.2 Cohen's Kappa

Cohen's kappa is an alternative to classical accuracy. Taking into account the probability of chance within each household size class, it is seem as somewhat more robust. A kappa of zero means zero match and a kappa of 1 means perfect match. The interpretation of Cohen's kappa is not easy because the magnitude changes also by other factors than agreement.

Cohen's weighted kappa punishes disagreement more as further apart the categories lie from each other. For example, incorrectly classifying a household of size 1 as 5 is worse than classifying it as size 2. Weighted kappa suggests to interpret household size as ordinal scale.

Table 9: Cohen's weighted Kappa in train and test dataset for the different classifiers.

	train	test
multinomial	0.64	0.48
randomforest	0.53	0.52
svm.linear	0.66	0.47
svm.radial	0.97	0.53

7.1.3 Confusion Matrix

The confusion matrix is the contingency table that results when counting matching classification between prediction and true condition. While accuracy and Cohen's kappa return a single overall score, the visualization of Cohen's agreement matrix gives an insight of the classification performance for each household size class.

Figure 8 shows a heatmap of the agreement matrix on test data for each classifier. All four classifier yield the same pattern. The prediction of small households (household size 1 and 2) is moderate while prediction of the household size larger than two persons is very poor.

7.2 Variable Importance

To asses which features are most important for the prediction of household size, the variable importance was calculated. For Random Forest the variable importance for a particular variable is the average score improvement across all nodes and all trees where that variable was chosen for splitting (MeanDecreaseGini).

For the multinomial logistic model the variable importance is assessed by the sum over the four absolute coefficients (there are 4 coefficients for 5 classes, household size 1 being the reference class). Although unbalanced, all five classes get the same weight for the overall importance in both the multinomial logistic model and Random Forest importance measure.

Figure 9 in the Appendix shows the 15 most important variables for the two multinomial logistic models and Random Forest. The reduced multinomial model shows almost identical ranking as the full model. This no surprise, as the stepwise model selection keeps important terms and drops irrelevant terms.

Overall, the set of important variables between Random Forest and multinomial model is quite similar, although the magnitude of importance differs substantially. Features of all the domains time, channels and programs are represented in the lists. Most significant difference are commercial programs and foreign channels that seem important for multinomial model but not for Random Forest. For Random Forest viewing children channels and programs is the most important feature for discriminating household size.

One reason for differences may be that multinomial regression did not model any interaction terms between features. The iterative binary partitioning by Random Forest on the other hand is able to model all possible interactions if opportune. Such interactions may differ form tree to tree and is not described mathematically. However we can visualize the overall relationship between two variables by partial dependence plot.

7.3 Partial Dependence

How exactly does one particular feature influence the probability of falling into one of household size class? Or how does the interaction between target and feature look like? This partial dependence can be visualized by fitting target variable for a range of values of the feature of interest while keeping the values of all other features in the model constant. Because we are modelling categorical classes, this has to be repeated for each class separately. When comparing the dependence of feature and target between different classifiers, we expect similar dependency pattern. For example, if viewing kids channels is negatively associated with the probability of household being classified as 1-person household, this negative relationship is expected to be apparent in all modelling processes.

Figure 11 to 14 show dependence plots for two features: viewing on kids channels and viewing on workdays between 17 and 20 o'clock. As more TV is watched on kids channels as more likely it becomes that the household is classified as a bigger household (size 3, 4 or 5) et vice versa. As more TV is watched during 17 and 20 o'clock on workdays, the probability increases of being classified as a small household (size 1 and 2).

In each case first the effects plot from multinomial linear regression is shown followed by the partial dependency plot from Random Forest. As expected the basic relation between target and feature is similar between models. Random Forest gives a much more precise interaction curve than multinomial linear regression which is defined to model harmonic log odd ratios transitions.

8 Discussion

In this study we have tried to predict the household size, e.g. the number of individuals living in a home by the TV viewing data that was recorded by the Mediapulse measurement. For the TV viewing raw data 53 features have been extracted and used as input data for learning algorithms. The features comprise viewing characteristics like the daytime, weekday, type of channel and program genre. This input data was first inspected, transformed and visualized. Three different models for statistical learning have been compared in terms of prediction performance: multinomial logistic regression, support vector machine and random forest. Their performance were assessed by accuracy and Cohen's kappa. The most important features were identified by variable importance measures and the dependency of a few features on household size were inspected by partial dependency plots.

It seems difficult to predict the number of individuals by the TV viewing behavior. An accuracy of about 43% with 27% baseline is rather poor. The prediction of small households (size 1 and 2) is significantly better than that of bigger households. To differentiate between a 3, 4 or 5-person home seems almost impossible.

The different classifier do all yield very similar results. Although they belong to different families of statistical learning algorithms (tree-based, kernel-based and linear regression model) none of them were able to outperform the others. It seems rather unlikely that any other algorithm would yield a significantly better performance.

To some degree the classifiers preferred different features in their variable importance ranking. Random Forest is much more flexible in pattern recognition as the multinomial linear regression, particularly as the latter was specified without interaction terms and nonlinear transformation terms (despite the log transformation of all features). It would be interesting to compare the variable importance of SVM, but no procedure to extract such scores form the SVM model was known to the author at time of writing.

The input data, 53 different characteristics of TV viewing duration in seconds was defined based on industry knowledge and intuition. Inspecting the structure of the input data matrix yielded good inter-feature variability and no dominant clustering. It would be possible to add more features such as time-shifted viewing, all 300 channels instead of channel groups or a more sophisticated partitioning of daytime and weekdays. Basically the performance of machine learning algorithms can only increase with more features. However we old not expect a significant improvement, believing that most of relevant characteristics of TV viewing is captured by the 53 features.

It would be interesting to replace the target variable household size by an alternative proxy of household composition. For example a classification of the type of household such as older couple, family with young / older kids, shared apartments, single mother / father household, etc. This type of household composition also includes age and gender information, which is not the case with household size. Maybe it is easier to distinguish these types. Although for the ultimate goal of assigning RPD TV viewing to individuals with age and sex attributes, it is not clear how such alternative household composition would be helpful.

9 Appendix

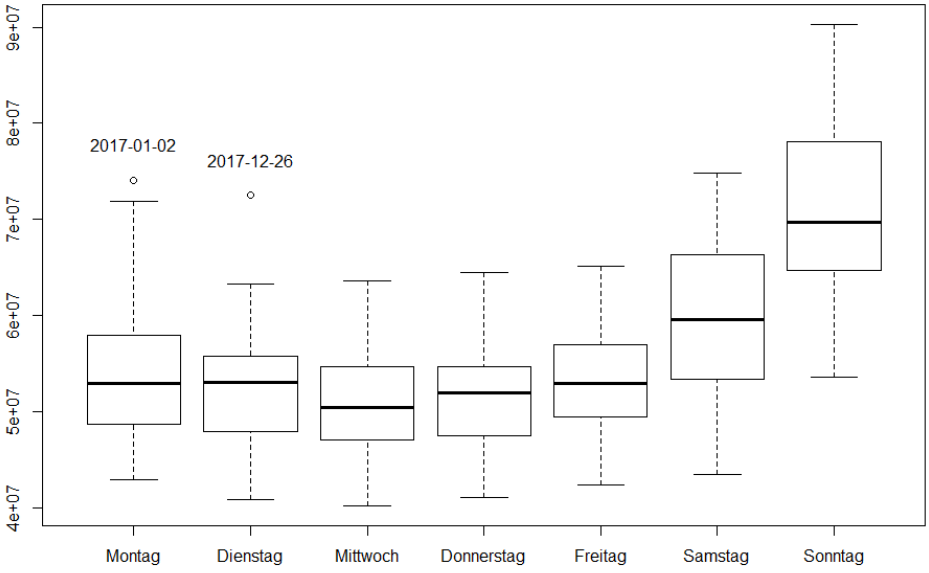

Figure 1: The sum of TV viewing duration [seconds] by weekdays during 2017. On weekends more TV is watched than during the rest of the week. Festival days often behave like Sundays.

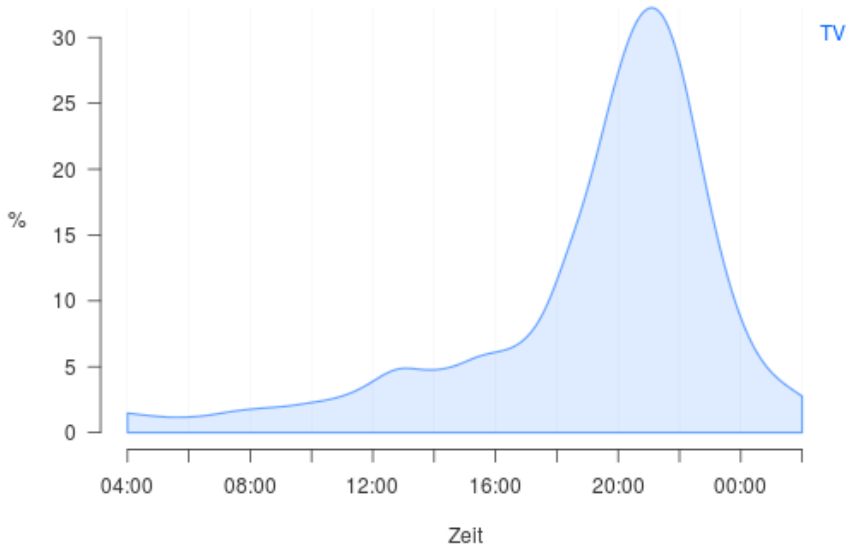

Figure 2: The relative amount of TV viewing across time of the day. The curve is the average of all 365 days in 2017. In the market the peak around 20:00 o'clock is called Primetime. On weekends the curve is flatter.

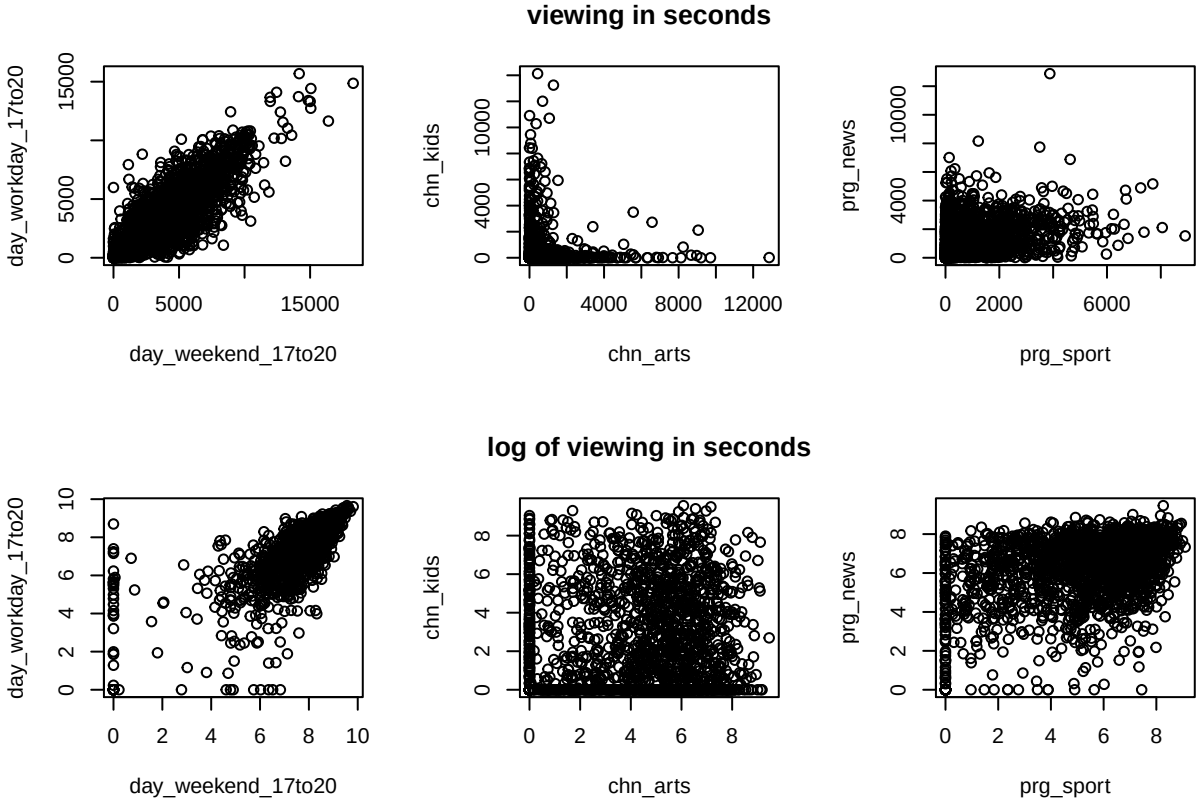

Figure 3: Illustration of log transformation. The upper row of scatterplots shows three examples of feature pairs. On of for each domain *time*, *channel* and *program*. In many cases viewing duration is not symmetrical distributed. The lower row shows the very same scatterplot with log transformed values.

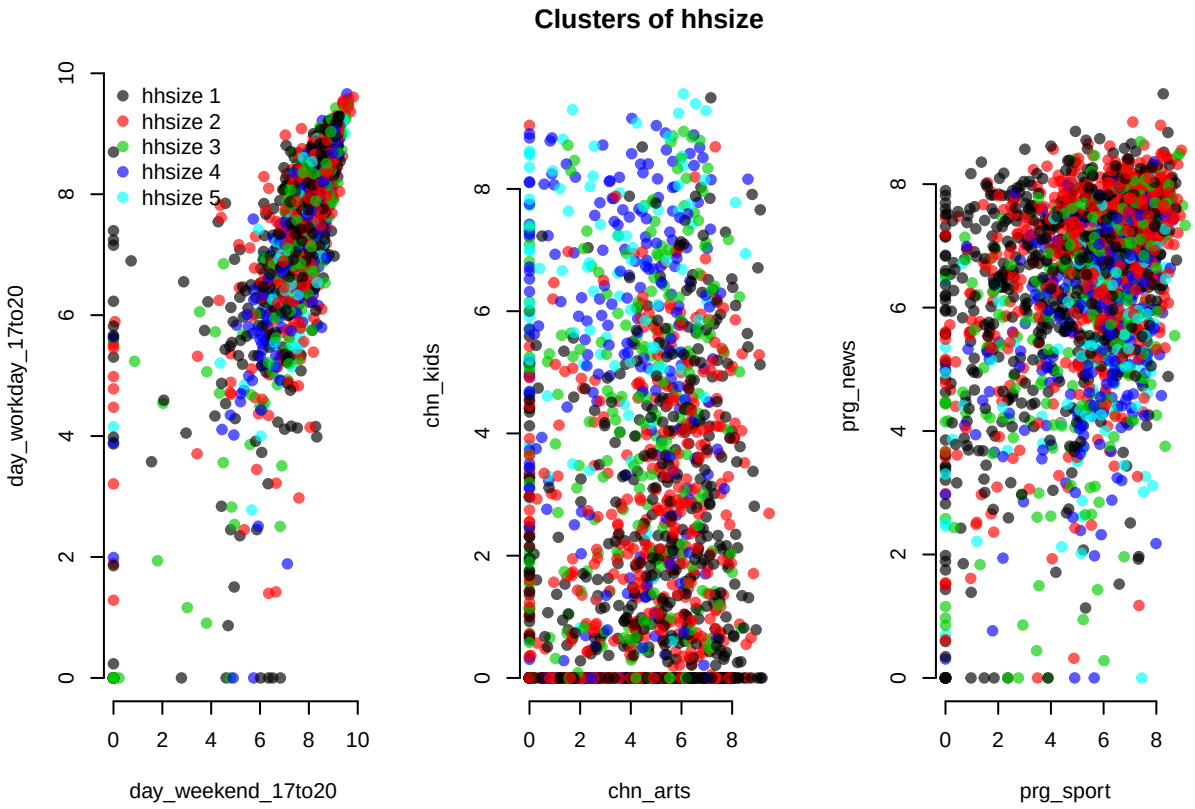

Figure 4: Shown are the same three scatterplots as in the Figure above but this time the corresponding household size is indicated by the color of the dots. If there was a feature that would separate the households (dots) into clusters of the same color, this would tell us that this particular variable is a good discriminator for household size. Apprently the variable *chn_kids* separates black and red dots from light and dark blue dots. This means there is a tendency that the more a household watches TV on typical kids channels, the more likely it is a 4 or 5 person household.

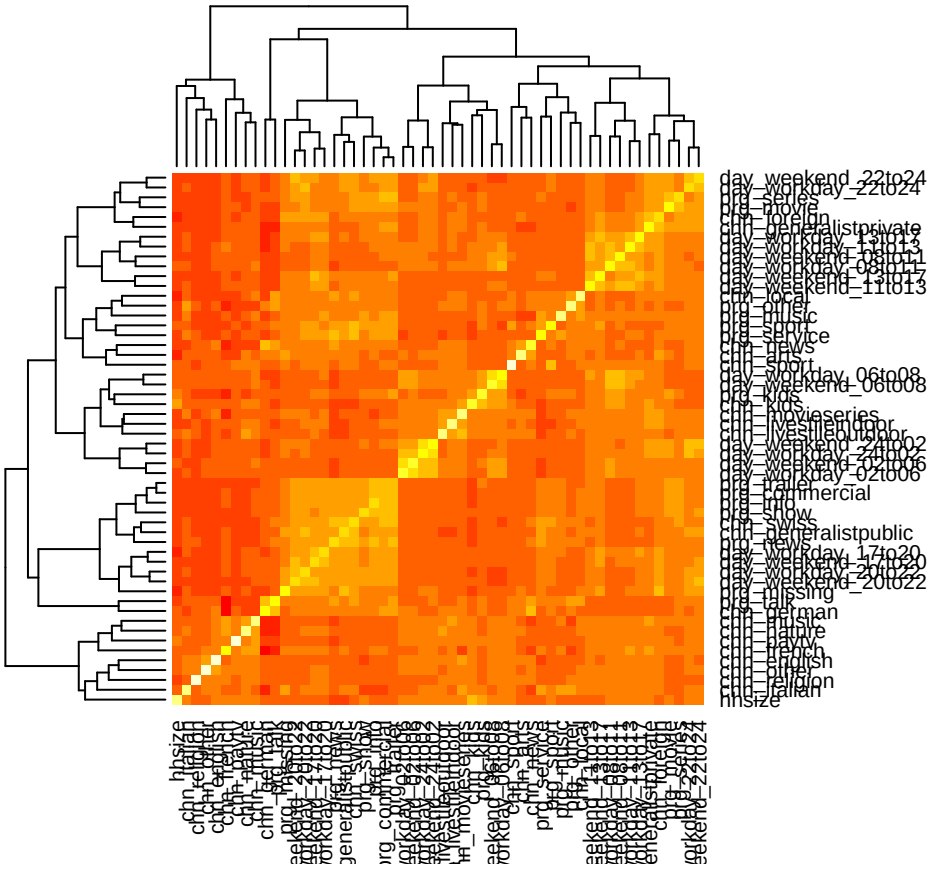

Figure 5: Heatmap of the correlation matrix of log transformed input data. Yellow is positive, red is negative correlation. Household size is included as numeric scale 1 to 5.

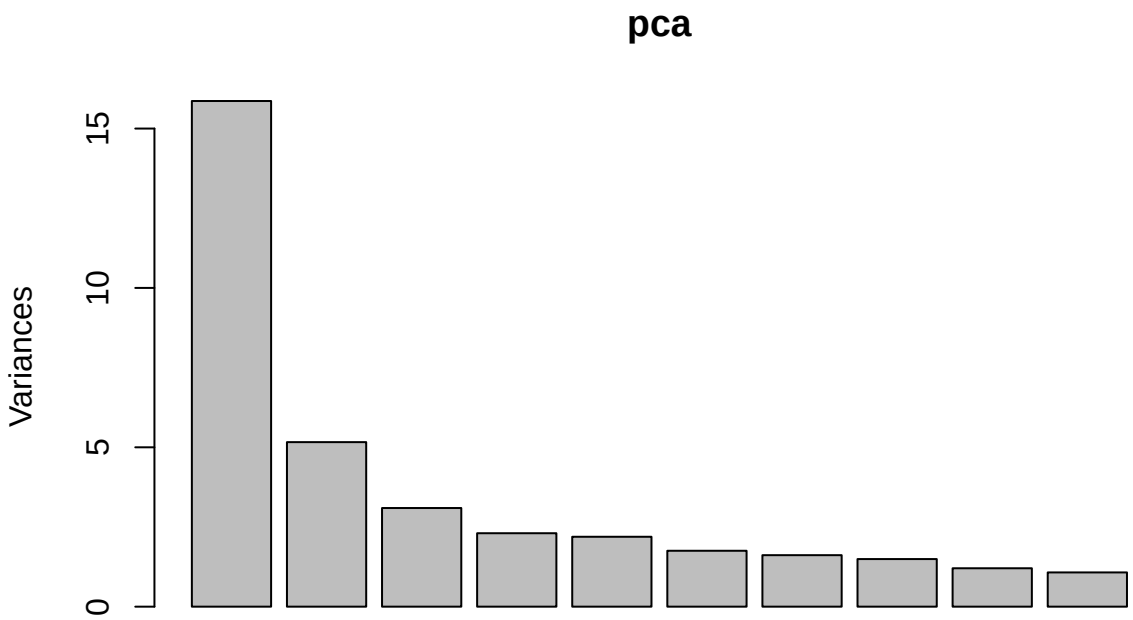

Figure 6: Principal Component Analysis (PCA). The screeplots shows the variance explained by the first 10 principal components (PCs). The first PC explaines 30% of the total variance.

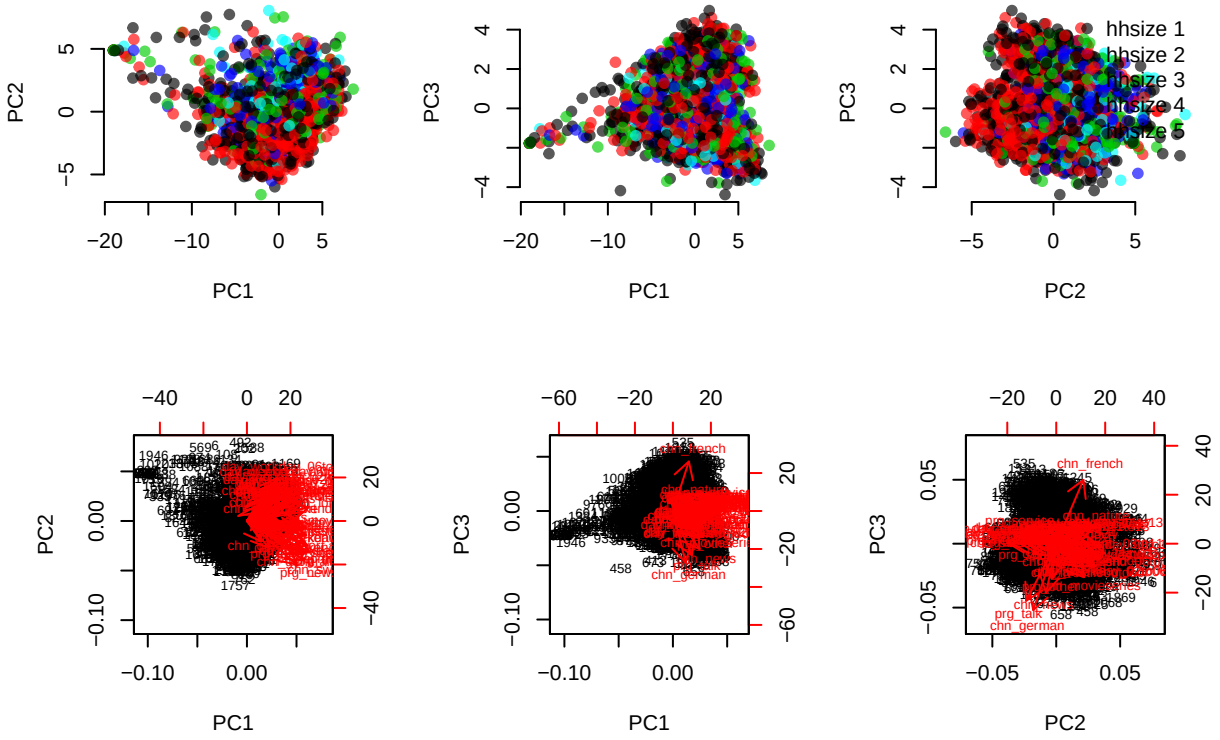

Figure 7: PCA scatterplots and biplots of the first 3 PCs.

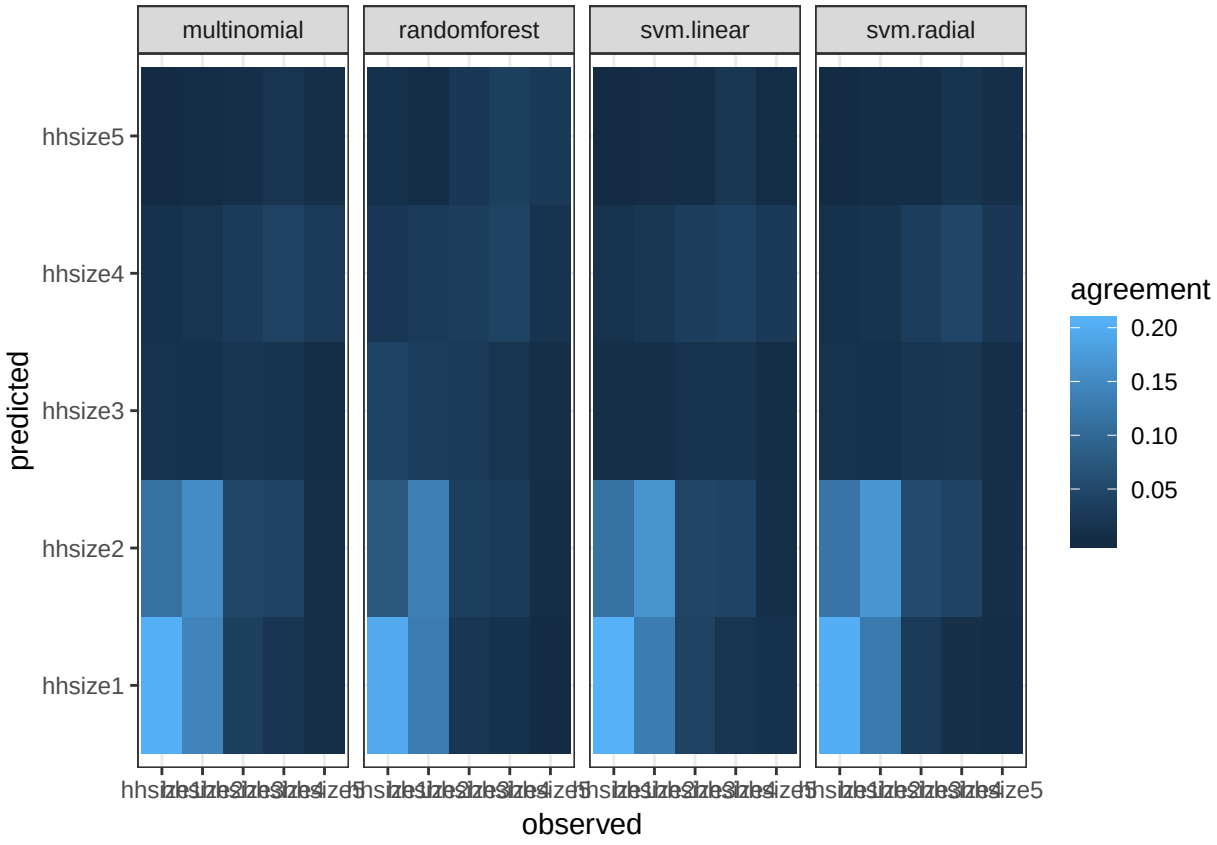

Figure 8: Heatmaps of Cohen's agreement matrix for each model.

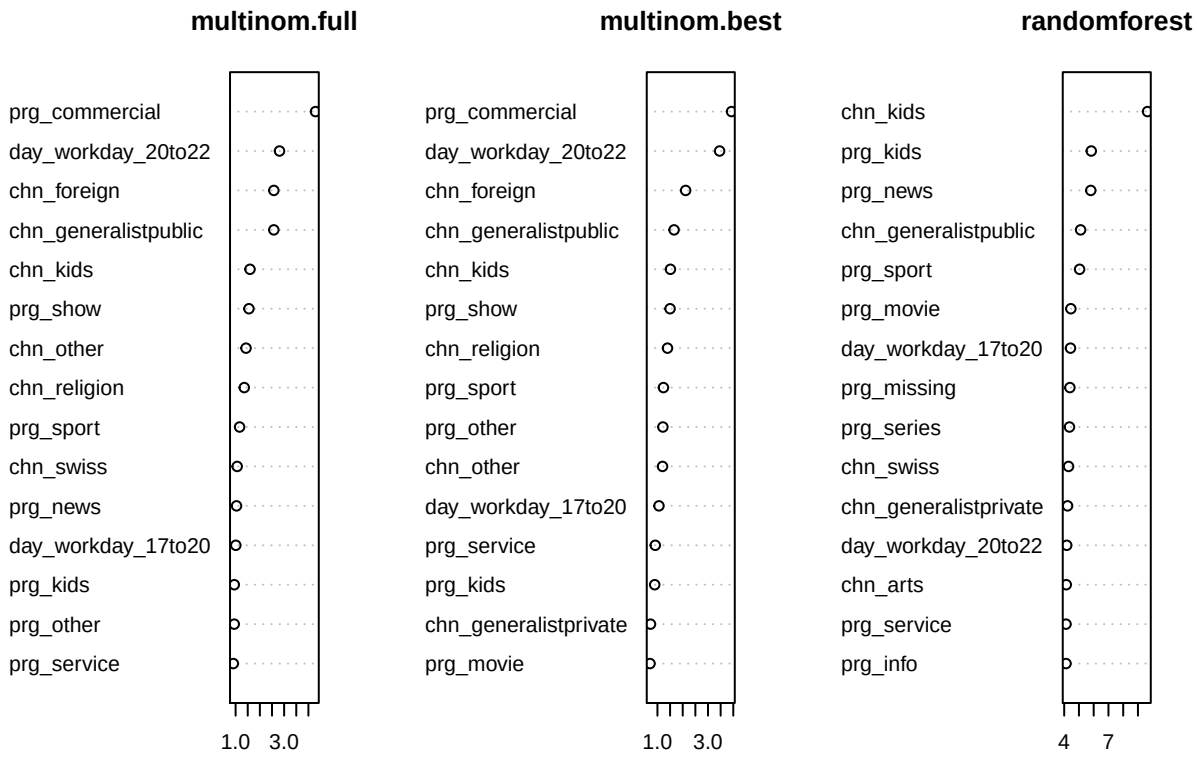

Figure 9: Variable importance plots for the multinomial full model, the reduced model and random forest.

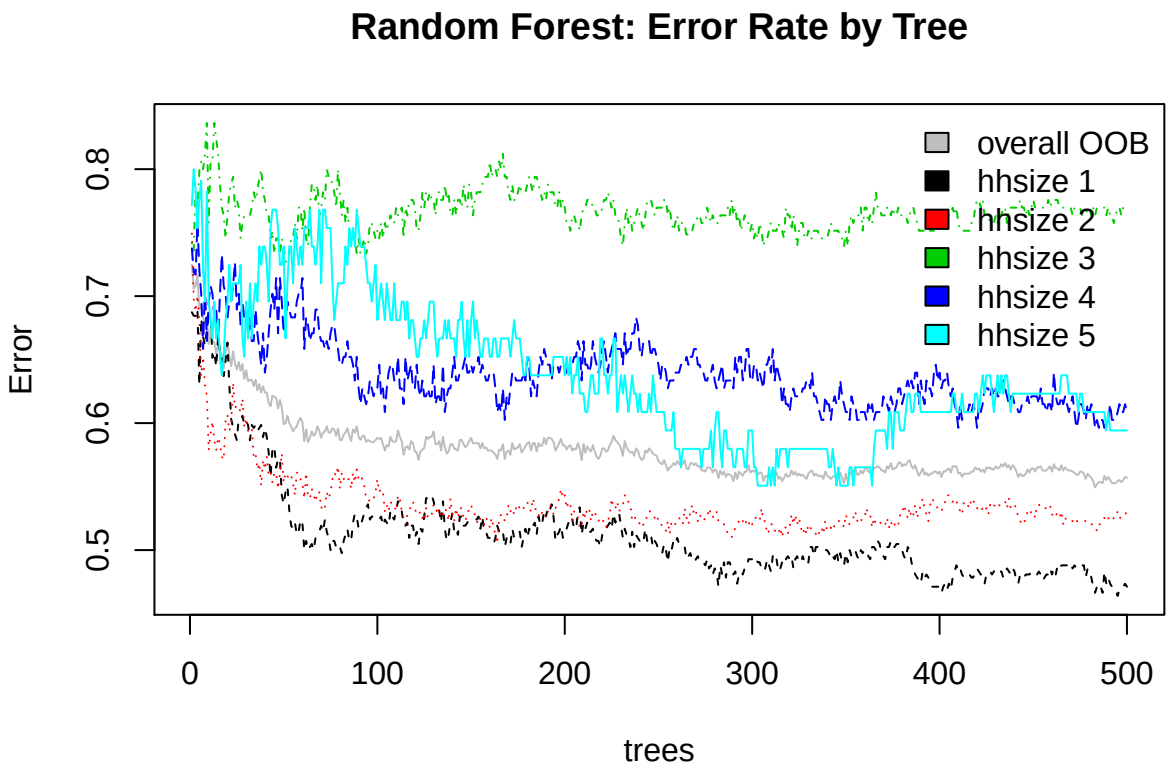

Figure 10: The Figure shows the Random Forest learning curve for predicting each household size class. The performance at each tree is calculated Out-Of-Bag (OOB), i.e. on the remaining cases that were not selected for training (cross validation).

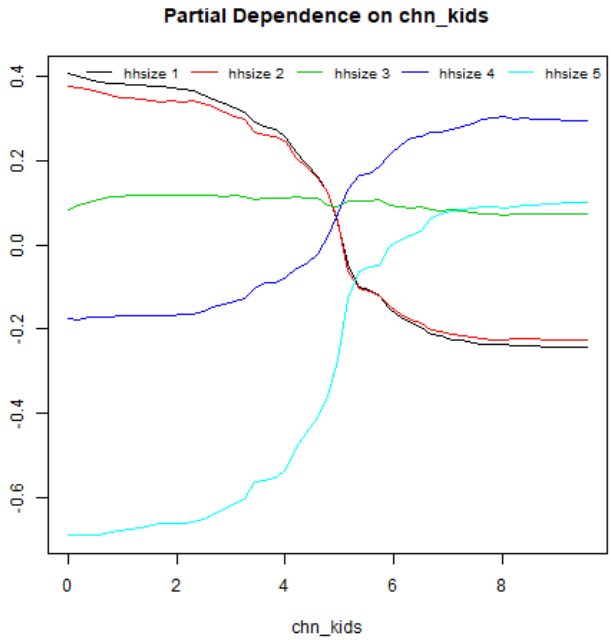

Figure 11: Random forest partial dependence plot for the variable kids-channel.

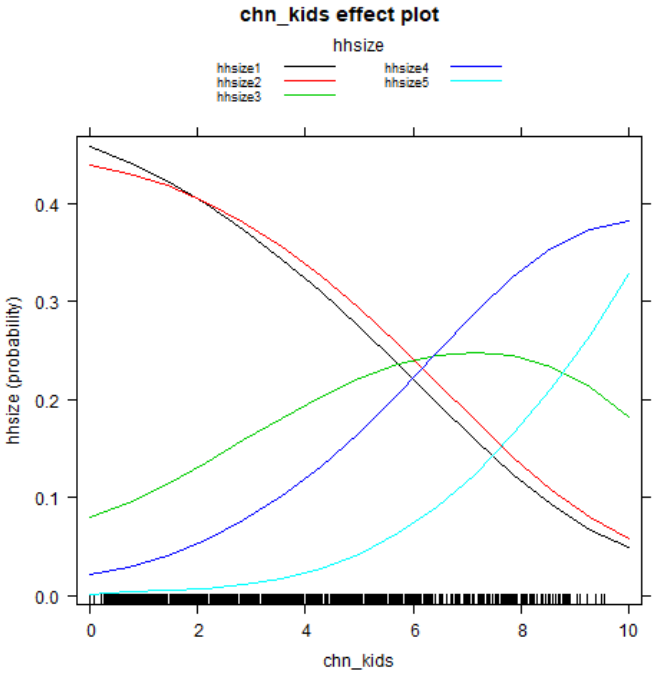

Figure 12: Multinomial Linear Model effects plot for the same variable as obove.

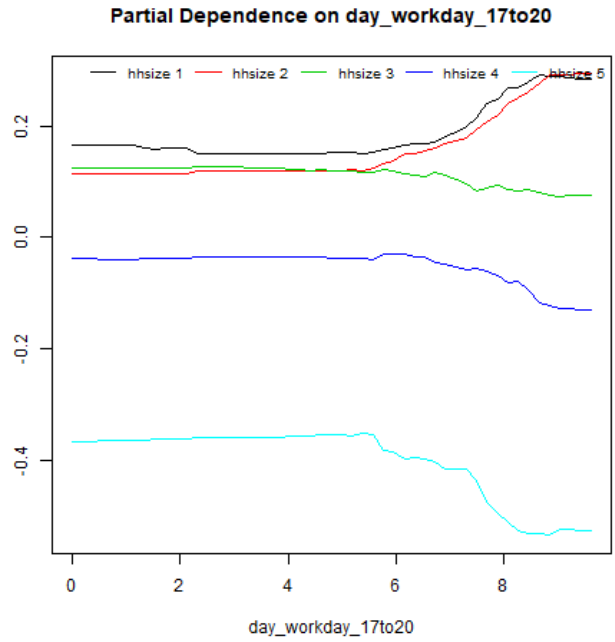

Figure 13: Random forest partial dependence plot for the amount of TV viewing between 17 and 20 o'clock during workdays.

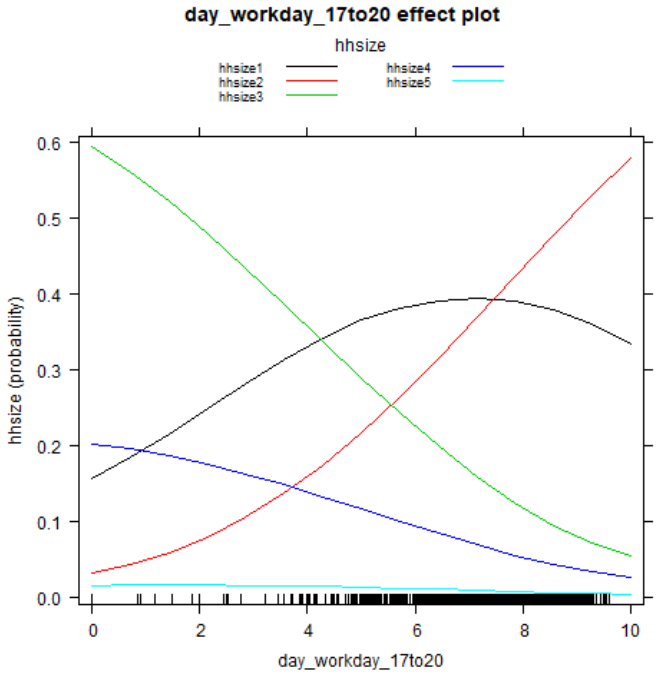

Figure 14: Multinomial Linear Model effects plot for the same variable as obove.