Contents

Appendices

A Taxonomies of statistical models

Figure 1: Taxonomy as proposed by Hastie and Tibshirani (2009), reduced form

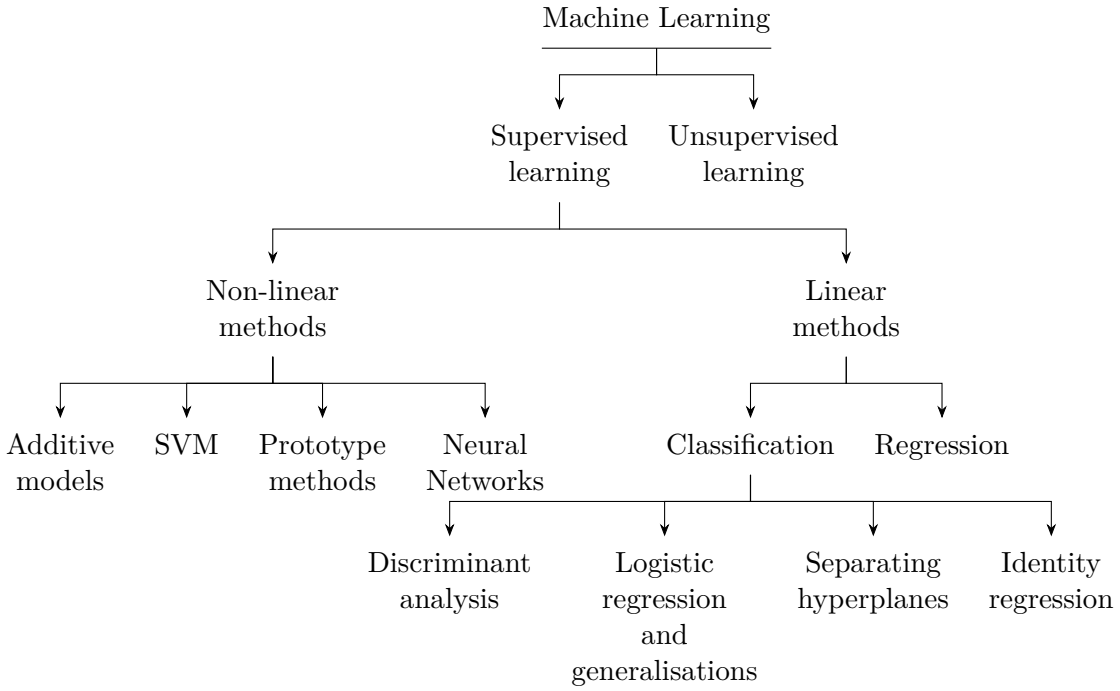


Figure 2: Taxonomy as proposed by Ayodele (2010)

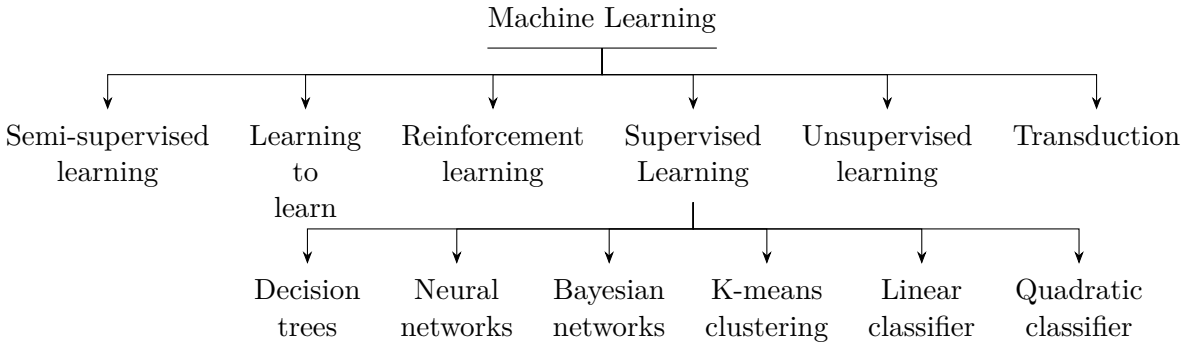
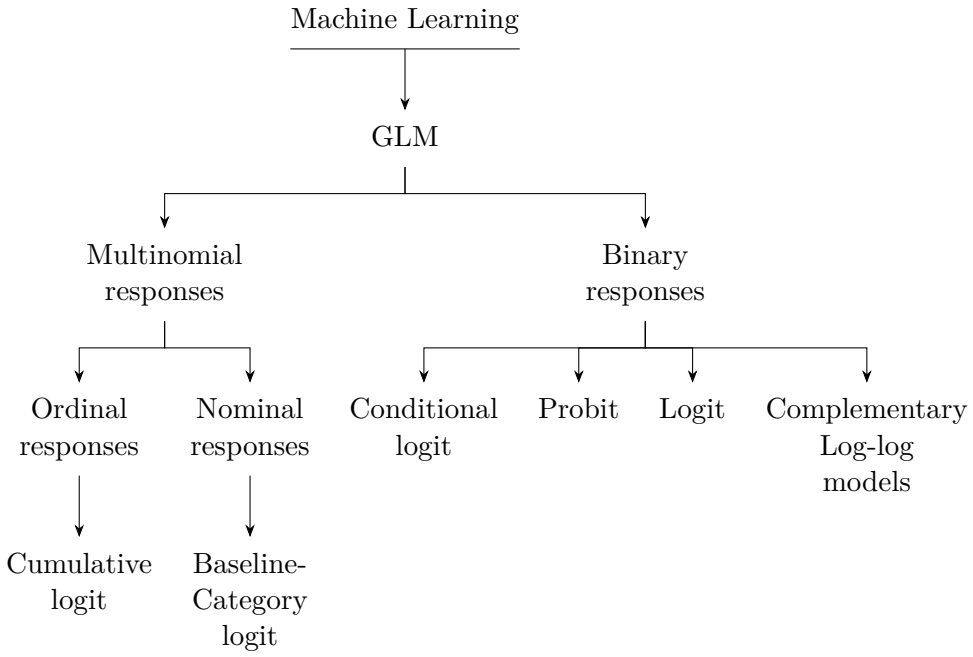
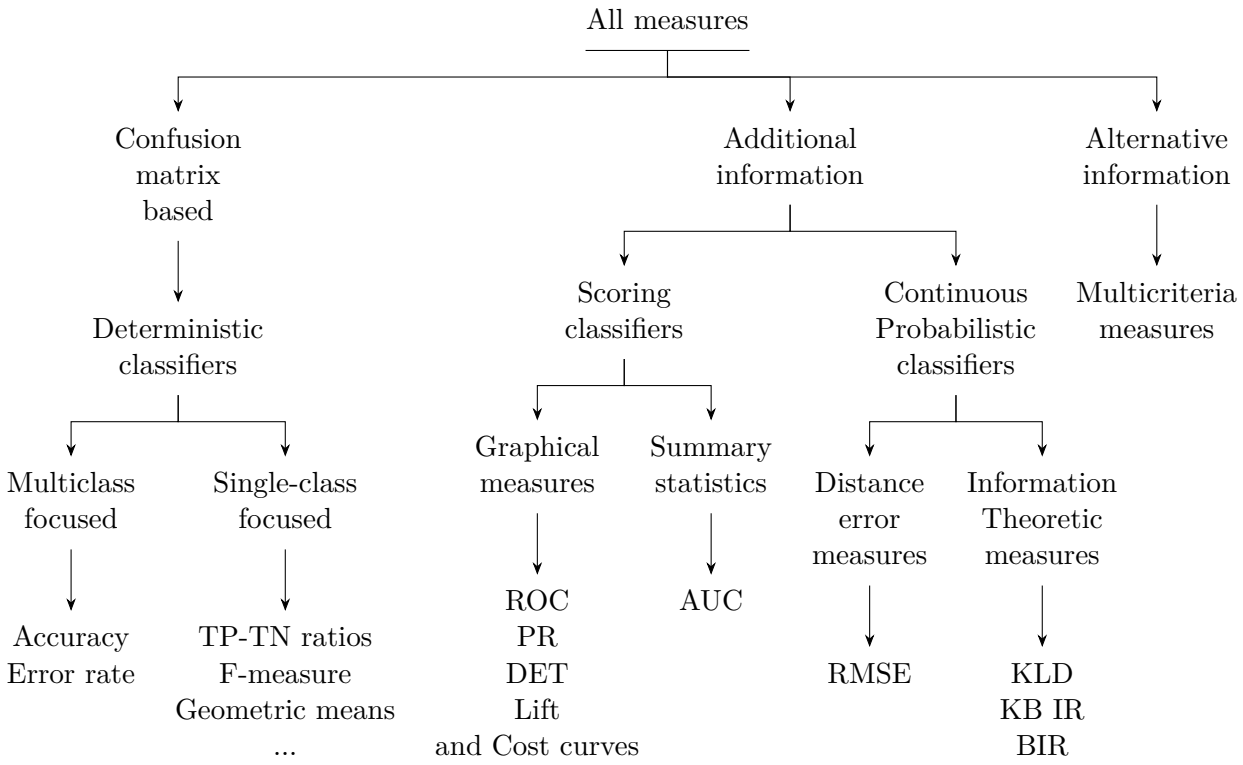


Figure 3: Taxonomy as proposed by Agresti (2013), based on data types



B Performance measures positioning

Figure 4: Performance measures as described by Japkowicz (2011)



C Descriptive statistics

C.1 Comparing datasets over A alternative

Table 1: Alternatives' descriptive statistics by dataset, stratified by alternative

Alternative		Fixed Effects (N=320000)	Random Effects (N=320000)	Target (N=2372)	p value
A	**Alternative**				
	A	160000 (100.0%)	160000 (100.0%)	1186 (100.0%)	
	B	0 (0.0%)	0 (0.0%)	0 (0.0%)	
	Choice				< 0.001
	Mean (SD)	0.427 (0.495)	0.382 (0.486)	0.517 (0.500)	
	Range	0.000 - 1.000	0.000 - 1.000	0.000 - 1.000	
	Price				0.022
	Mean (SD)	3.069 (0.979)	3.069 (0.979)	2.990 (0.881)	
	Range	1.500 - 4.500	1.500 - 4.500	1.500 - 4.500	
	Carbon				< 0.001
	Mean (SD)	0.500 (0.500)	0.500 (0.500)	0.167 (0.373)	
	Range	0.000 - 1.000	0.000 - 1.000	0.000 - 1.000	
	Label				0.993
	Mean (SD)	0.500 (0.500)	0.500 (0.500)	0.502 (0.500)	
	Range	0.000 - 1.000	0.000 - 1.000	0.000 - 1.000	
	Price by group				< 0.001
	1.5	16000 (10.0%)	16000 (10.0%)	82 (6.9%)	
	2	24000 (15.0%)	24000 (15.0%)	223 (18.8%)	
	2.5	27000 (16.9%)	27000 (16.9%)	214 (18.0%)	
	3	23000 (14.4%)	23000 (14.4%)	175 (14.8%)	
	3.5	22000 (13.8%)	22000 (13.8%)	187 (15.8%)	
	4	21000 (13.1%)	21000 (13.1%)	219 (18.5%)	
	4.5	27000 (16.9%)	27000 (16.9%)	86 (7.3%)	

C.2 Comparing datasets over B alternative

Table 2: Alternatives' descriptive statistics by dataset, stratified by alternative

Alternative		Fixed Effects (N=320000)	Random Effects (N=320000)	Target (N=2372)	p value
B	**Alternative**				
	A	0 (0.0%)	0 (0.0%)	0 (0.0%)	
	B	160000 (100.0%)	160000 (100.0%)	1186 (100.0%)	
	Choice				< 0.001
	Mean (SD)	0.518 (0.500)	0.462 (0.499)	0.159 (0.366)	
	Range	0.000 - 1.000	0.000 - 1.000	0.000 - 1.000	
	Price				< 0.001
	Mean (SD)	2.803 (0.917)	2.803 (0.917)	3.020 (0.893)	
	Range	1.500 - 4.500	1.500 - 4.500	1.500 - 4.500	
	Carbon				< 0.001
	Mean (SD)	0.500 (0.500)	0.500 (0.500)	0.832 (0.374)	
	Range	0.000 - 1.000	0.000 - 1.000	0.000 - 1.000	
	Label				0.985
	Mean (SD)	0.500 (0.500)	0.500 (0.500)	0.497 (0.500)	
	Range	0.000 - 1.000	0.000 - 1.000	0.000 - 1.000	
	Price by group				< 0.001
	1.5	25000 (15.6%)	25000 (15.6%)	108 (9.1%)	
	2	28000 (17.5%)	28000 (17.5%)	192 (16.2%)	
	2.5	26000 (16.2%)	26000 (16.2%)	158 (13.3%)	
	3	30000 (18.8%)	30000 (18.8%)	204 (17.2%)	
	3.5	18000 (11.2%)	18000 (11.2%)	232 (19.6%)	
	4	23000 (14.4%)	23000 (14.4%)	195 (16.4%)	
	4.5	10000 (6.2%)	10000 (6.2%)	97 (8.2%)	

D *R* code for implemented models

D.1 MNL model

```
# Transform dataset to mlogit format
mnl_data = data %>%
    mlogit.data(
        choice = "Choice",
        alt.var = "Alternative",
        shape = "long", # Long format
        alt.levels = c("C", "A", "B") # Define order of alternatives
    )

# Function
utility = Choice ~ Sex + Age + Salary + Habit + # Individual characteristics
    Price + Buy + Label + Carbon + LC + 0 | 0 # Alternatives attributes

# Estimate MNL model
mnl_novar = mlogit(
        utility,
        data = mnl_data,
        reflevel = "C", # The No-buy option is the baseline
        print.level = 3, # Print estimation details
        iterlim = 1000
    )
```

D.2 MMNL model

```
# Transform dataset to mlogit format
mmnl_data = data %>%
    mlogit.data(
        choice = "Choice",
        alt.var = "Alternative",
        id = "ID", # Set individuals' index
        chid = "CHID", # Set choice sets index
        shape = "long",
        alt.levels = c("C", "A", "B")
    )

# Function
utility = Choice ~ Sex + Age + Salary + Habit + # Individual characteristics
    Price + Buy + Label + Carbon + LC + 0 | 0 # Alternatives attributes

# Estimate MMNL model
mmnl = mlogit(
```

```
        utility,
        data = mmnl_data,
        reflevel = "C", # The No-buy option is the baseline
        correlation = TRUE, # Include covariance (and not variance only)
        rpar =  c( # Normality assumption and four parameters
            "Buy" = "n",
            "Label" = "n",
            "Carbon" = "n",
            "LC" = "n"
        ),
        panel = TRUE, # Estimate dataset as panel
        print.level = 3, # Print estimation details
        iterlim = 1000
    )
```

D.3 CNN model with *Adam* algorithm

```
# Used libraries
library(tidyverse)
library(tensorflow)
library(keras)

# Define optimization algorithm to be used
adam_own = optimizer_adam(
    lr = 1e-1, # We adjust the learning rate, keeping the rest as defaults
    beta_1 = 0.9,
    beta_2 = 0.999,
    epsilon = NULL,
    decay = 0,
    amsgrad = FALSE,
    clipnorm = 6, # We limit as well the max value for weights
    clipvalue = NULL
)

# Set hyperparameters
## The number of epochs is a hyperparameter that defines the number times
## that the learning algorithm will work through the entire training
## dataset.
epoch = 50
## The batch size is a hyperparameter that defines the number of samples
## to work through before updating the internal model parameters.
batch = 16000

# Limit softmax weights
## (keras uses dense layer transformation inside softmax layer by default)
softmax_weights = list(
```

```
    matrix(
        c(  1,  0,  0,
            0,  1,  0,
            0,  0,  1),
        nrow = 3
    )
)

# Setup CNN model
model_cnn = keras_model_sequential() %>%
    # We reshape the dataset, as 1D convolution requires 3D tensor as input
    layer_reshape(
        target_shape = c(27, 1),
        input_shape = 27,
        trainable = FALSE
    ) %>%
    # 1D convolution layer
    layer_conv_1d(
        filters = 1L, # Dimentions of the output space
        kernel_size = 9L, # Number of parameters
        strides = 9L, # Strides of convolution equal to parameters side
        # The starting value is 0 to ensure reproducibility
        kernel_initializer = "zeros",
        # The constant is not added, because we already have "Buy" dummy
        use_bias = FALSE,
        # We want a linear activation function
        activation = "linear",
        input_shape = c(27, 1)
    ) %>%
    # An inverse transformation into a 2D tensor for softmax implementation
    layer_flatten(
        data_format = "channels_first"
    ) %>%
    # Softmax layer
    layer_dense(
        units = 3, # Number of units equal to categories (3 utilities)
        use_bias = FALSE, # The bias constant is not estimated
        weights = softmax_weights,
        trainable = FALSE, # This layer is fixed
        activation = "softmax" # Softmax layer (to obtain probabilities)
    ) %>%
    # Learning algorith definition
    compile(
        loss = "categorical_crossentropy", # Choice of loss function
        optimizer = adam_own, # Parametrised Adam
        metrics = c("accuracy") # Target metrics
    ) %>%
```

```
    # Training the model
    fit(
        X_train, Y_train, # To train the model we use 80% of our dataset
        epochs = epoch,
        batch_size = batch,
        validation_data = list(X_test, Y_test) # 20% for validation
    )
```

Annexes

I Simulation tool for performance comparison of discrete choice models

Author: Amirreza Talebijamalabad, M1 SIE (Grenoble INP)

Under supervision of: Iragaël Joly, HDR (GAEL, UGA, Grenoble INP)

Available at: https://github.com/Amirreza-96/sdcm

An experimental design is a plan which identifies the independent, dependent, and nuisance variables and indicates the way in which the randomization and statistical aspects of an experiment are to be carried out. Speaking of experimental design, we need to bear randomization, replication and blocking in our minds as three key elements of the experimental design (**???**). Randomization as a rather new concept in design of experiments, plays a pivotal role in distribution of idiosyncratic characteristics and variables' levels so that they do not selectively bias the outcome of the experiment. For example, in our designs, we applied randomization to avoid dominant alternatives as much as we can. Replication is the observation of two or more experimental units under the same conditions. Replication enables us to validate the proposed model and ensures the precise effects. Usually, in simulation, we run a very long replication or we make relatively many replications but small in dimensions, which we choose to replicate once but large enough. Blocking, on the other hand, is an experimental procedure for isolating variation attributable to a nuisance variable. Also, making blocks, we can randomly assign respondents to the choice sets or control the number of respondents in order to intimate the real scenarios; However, blocking is not of great importance when we are talking in the realm of simulation since we can control variations and variables.

Stated choice experiments present sampled respondents with a number of different choice situations, each consisting of a universal but finite set of alternatives defined on a number of attribute dimensions. Respondents are then asked to specify their preferred alternatives given a specific hypothetical choice context. In simulation, since the respondents are artificial, it will not be wiseful to sample the population, instead, replicationg the processes would be fruitful as the population will be generated repeatedly which is more close to reality. Moreover, to simulate the choice making process, based on decision rules such as utility maximization, utilities are calculated to reveal the choices of the individuals. SC data requires that the analyst designs the experiment in advance by assigning attribute levels to the attributes that define each of the alternatives which respondents are asked to consider(**???**).

To generate experimental designs for SC studies we need to find out how to allocate the attribute levels to the design matrix. Traditionally, researchers have relied on the principle of orthogonality to populate the choice situations shown to respondents. The orthogonality of an experimental design relates to the correlation structure between the attributes of the design. however, this class of designs may not be statistically efficient, as they do not take the SC model specification into account. These models are optimal for the linear models and assure the researcher that multicollinearity does not exist in design. Considering this, it is assumed that such designs can be used for the non-linear models by linear arrangements(**???**). It is important to note however, that the orthogonality of a design suggests nothing about whether two or more attributes are cognitively correlated in the minds of the respondents (e.g. price and quality attributes). As such, orthogonality is purely a statistical property of the design and not a behavioural property imposed upon the experiment(**???**). Moreover, by entreing non-design attributes such as socio-demographic variables, any covariate within the dataset will unlikely be orthogonal, not only amongst themselves,

but also with the design attributes. For example, if age, gender and income are added as variables in an analysis, correlations are not only likely to exist for these variables, but given that the variables described are constant over all choice situations within individual respondents, correlations between these variables and other attributes of the design are also likely to exist. Simulation tool should allow us to enter or not such soci-demographic variables to the simulation process so that at least we have some control on correlations. Furthermore, more advanced data generation methods should be applied to generate correlated data with specific precision. In this research, we have made a very conventional and widespread design so called full factorial design. It contains all of the possible levels of factors, and allows us to estimate all of the main effects and two-way interactions. Main effects are independent of the levels of other attributes, however; interactions involve two or more factors in which, effect of one factor depends on the level of another(**???**). Furthermore, there are fractional orthogonal designs known as efficient designs providing ratherly small but efficient designs, also, there are algorithms to determine the correlations between columns as orthogonality is violated in these designs. It would be excellent if various kinds of designs were available in simulator.

(**???**) conducted an empirical work to figure out consumers' willingness to pay, and a price premium for two environmental attributes of a non-food agricultural product(Roses). In this research there are two unlabelled alternatives Rose A and B and one no choice alternative. The two attributes, Label and Carbon, have two levels which make four combinations, hence six pairs of alternatives can be drawn from these combinations. Price ranges from 1.5 to 4.5 and is randomly assigned to the combinations of two other attributes. Finally, each respondent is faced with twelve choice sets(24 combinations of all attributes or 12 questions), hence, considerring no choice mode, there are three alternatives in each question. Trying to simulate the paper's results, we made a design with the same attributes and attribute levels. We sample put alternatives two by two in choice sets (16 choice sittuation), hence, respondents are faced twice with six pairs of alternatives, but the price is randomly assigned to each of the choice sets. Moreover, as no-choice mode does not effect the design, we do not add this mode to the design but finally, when it comes to utility comparison and decision process, this alternative is taken into account. Furthermore, we have not put interaction variable in the design since as no-choice mode, it does not affect the combinations of the design. To add, it makes the tool more flexible if we allow the user to decide about these two options.

(**???**) considered four socioeconomic characteristics as well as sex, age, income and organic purchase habit. Since no information were available in regard to these features' correlation, we assumed that they are uncorrelated, and made each feature independently. This is a limitation for data generation process. Simulator must enable the user to specify whether data is correlated or not. Moreover, it should allow the user to enter the inputs and specifications as well as distributions and their parameters. In order to generate sex data, we draw samples out of a uniform distribution with parameters $a = 0, b = 1$, then we assume that there is a 0.49 chance that a respondent is female. Hence, if the random number is in range $(0, 0.49)$, hypothetical individual is female, otherwise, is male. The same procedure applies to the habit feautre. If the random number is among $(0, 0.35)$, organic habit is assumed to be zero. In order to generate age feature, the best distribution that we can draw samples which exactly could resemble the real data is truncated normal distribution. However, we just have the tnormal distribution's parameters and we need to have the underlying normal distribution's parameters(mean, and std.) to be able to draw samples. To tackle this, we solve a system of non-linear equations utilising numerical methods(Newton Raphson method) to find the underlying normal dist. parameters. Another way to generate such data is to draw samples from a log normal distribution. We still have a problem with this way since we need to have data ranging from 18 to 85, nevertheless positive values are generated. And finally, we simply take

draws from normal distribution with the same parameters. As future improvements, simulation tool should be able to generate data based on theoretical distributions or empirical ones. Hence, some curve fitting procedures to find the distribution best fitting the real datashould be installed in the tool.

So far, we have made the design and socioeconomic features for artificial individuals. Now, we need to specify utilities per each individual, and finally, due to the RUM model, we select the alternative with highest utility per each individual per each choice set. As a future improvement, we suggust that simulation tool should be able to simulate decision making process based on different approaches for example, regret minimization. In order to calculate utilities, we took parameters from the paper (*a priori*). All of the terms mentioned in the paper including ASC are used. We take no-choice as reference alternative. Firstly, a matrix of 4×1000 is constructed for socioeconomic characteristics parameters. Each column indicates a person , and all of the columns are similar since the parameters are constant for all of the people. Then, this matrix is pointly multiplied with the matrix of socioeconomic charachteristics matrix, and finally the sum of each column is the socioeconomic utility of each person and a vector of utility is achieved. Secondly, a matrix of 1000×5 is made to contain the parameters corresponding to the alternatives(price, label, carbon, label-carbon, constant). each row of this matrix is drawn from multivariate normal distribution, $\mu + L \times R$ where μ is a vector of means of parameters, L is derived from Cholesky decomposition($L \times L' = \sigma^2$) and R is a vector of K draws from a $N\ (0, 1)$. Finally, this matrix is multiplied by the inverse of design matrix which results in a matrix of 1000×36 in which each element shows the utility of an alternative for an individual. Consequently, we add up socioeconomic utility to each of the columns of this matrix. This makes the observed utility. In regard to unobserved utility, a matrix of 1000×36 is containing the draws of $Gumbel(0, 1)$, then we add this matrix to the previous one and this brings about the utility matrix. For the choice selection process, columns of utility matrix are compared pair by pair and also the max of these each of these paires is compared with an element of $Gumbel(0, 1)$ to specify whether the individual buys or not. Finally, we suggust that tool decode and clean the data. One important issue is the difference between real data and simulated data which arises from ommited variables. For example, when it comes to reality, time is a very important factor affecting the choices made by respondents, but when it comes to simulation, time is meaningless for artificial individuals. These issues also need to be taken into account specially when we are comparing estimation results of these two types of data.

II Reproducible research

This work was accomplished with implementation of the most advanced reproducible research techniques. First of all, a version control system (*git*) was used to track the changes and modification in the working tree from the start of the internship. The collaboration with other participants was organised through *GitHub*, where a common repository was maintained to store the data and document, as well as to keep every element of the code or text available to everyone. The report generation was automated with the use of a simplified markup language with embedded executable *R* code. For heavy tasks, such as data generation, model estimation or big data exploration separate source code files were used.

This short documents aims to introduce the reader to used research methodology, that was used in this work and during the internship. The used tool-set will be introduced.

Git is one of the version control tools alongside SVN and Mercurial-SCM, which allows to easily control changes and modifications within text documents. Unfortunately the proposed functionality does not function with more complex proprietary formats such as Word or image based documents, such as PDF. Consequently, this tool is not practical only for working with simple text documents: it remains absolutely impractical for working with typical office tasks. Several text editors for developers among which RStudio, VSCode, Atom and many other provide possibilities to integrate *git* functionality directly into the editor and drastically optimise the workflow. This makes interacting with *git* much more comfortable than through the command line or a standalone *git* client.

GitHub is an open source cooperative platform for developers offered by Microsoft making it easier to work with the *git* version control service. The platform has an entire ecosystem of extensions, expanding git functionality, as well as a set of project management and communication tools. In total this platform offers:

- A cloud space to host the working files and publish the results;
- A web interface to interact with *git* from browser or through a standalone app;
- A platform facilitating collaboration with other users, which gradually approaches in the functionality to a social network;
- An integrated project management system.

To write the scientific report it was decided to use the *LaTeX* complete markup language. There are several distributions of LaTeX, one of the verified versions to integrate well with *R* being *tinytex* (which is available as *tinytex* package in CRAN repositories). However, even if *LaTeX* produces well structured documents that are easy to manage, there exists the problem of its complexity in extending its functionalities. Consequently, it was decided to use an intermediary simplified markup language, which is easy to use and does not require advanced knowledge of *LaTeX*: Markdown. It allows to write documents with simple syntax, which could be later transformed into PDF, HTML and Word documents using the *pandoc* converter.

Finally, to embed the *R* code inside the document to automatically generate the figures and tables, we used *RMarkdown*, which is an extension for *Markdown* integrating *R* language inside. Such set-up ensured, that the documents will be easy to share and modify, preserving at the same time all their functionality. This work offers all the necessary elements to be fully reproducible.

The resulting compedium is available at: https://github.com/nikitagusarov/performance_exploration