Forecasting Individual Brand Choice with Neural Networks: An Empirical Comparison

Bachelor Thesis Econometrics and Operations Research

Erasmus School of Economics, Erasmus University Rotterdam

Author: Amber Stoll (443139)   Supervisor: A. Castelein

Second assessor: F.J.L. van Maasakkers

July 10, 2019

Abstract

In this study we compare the forecasting performance of a multinomial logit (MNL) model with that of an artificial neural network (ANN). We also implement a hybrid model, which uses an ANN as a diagnostic tool to detect nonlinearities in the data and then incorporates the nonlinear relations found into an MNL model. We use a scanner data set containing information on 2798 Catsup purchases with four different brands. Price and whether the product was on display or featured in an advertisement at the time of purchase is given, and we also construct a brand loyalty term. Data is split up into a training and test set and different data partitions are evaluated. Forecasting performance is measured with accuracy, the negative prediction ratio and evaluation of confusion matrices. The MNL with an 80/20 data split yields an accuracy of 72.1% on the test set, while the ANN with data partitioning of 65/35 performs slightly better with an accuracy of 72.3%. The hybrid model outperforms both with an accuracy of 73.2% and thus we conclude that this is the most suitable model to forecast individual brand choices on Catsup.

The views stated in this thesis are those of the author and not necessarily those of Erasmus School of Economics or Erasmus University Rotterdam.

Contents

1 Introduction 2

2 Literature Review 3

3 Data 4

4 Methodology 5

4.1 Brand Loyalty 5

4.2 Multinomial Logit Model 5

4.3 Artificial Neural Network 7

4.4 Hybrid Model 11

4.5 Data Partitioning 12

4.5.1 Multinomial Logit Model 13

4.5.2 Artificial Neural Network 13

4.5.3 Hybrid Model 13

4.6 Performance Measures 14

5 Results 14

5.1 Multinomial Logit Model 14

5.2 Artificial Neural Network 16

5.3 Hybrid Model 17

5.4 Model Comparison 20

6 Conclusion 21

References 22

Appendix A Results 24

A.1 ANN Connection Weights 24

A.2 Scatterplot Utility of Hybrid Model 25

A.3 Hybrid Multinomial Logit Model Estimates 25

Appendix B Code 25

1 Introduction

Since the introduction of barcodes and barcode scanners, it has become very easy for grocery stores to keep track of their customers' purchase behaviour. The data they collect can tell them which brand of a product is bought, when it is bought - and since a lot of supermarkets use loyalty cards nowadays - by whom. When they combine this with their data on which promotional activities were happening at the time of each purchase, they have a very valuable data set on their hands. This data can be used to infer preferences of individual households and forecast their future purchase decisions. Naturally, being able to accurately predict these decisions is very valuable to grocery retailers in terms of maximizing profits and maintaining or improving their market position.

A traditional technique in the field of econometrics to model choice behaviour is the Multinomial Logit (MNL) model, which has been shown to be quite successful. It is popular because of the closed form of its choice probabilities. The MNL is based on a utility function that is assumed to be linear. This prevents the model from taking any nonlinearities in the data into account, which has a negative effect on its predictive capability.

Another approach is to make use of an Artificial Neural Network (ANN). This nonlinear statistical model is based on the human brain and can be trained to recognize data patterns. When nonlinear relations are present in the data, the neural network is expected to outperform the MNL as it can generalize these relations better. However, the ANN is sometimes referred to as a black box, because its structure does not reveal any information about the data itself.

In this study the goal is to compare the forecasting performance of a MNL with that of a three-layer ANN, and determine which is more suitable to model and forecast brand choice behaviour of individual households. For this we make use of a data set containing information on individuals' choices to purchase brands of Catsup. It contains data for 300 households and 2798 purchases. Moreover, a hybrid model is formed in an attempt to take advantage of the best qualities of both models. This is done by having the ANN detect nonlinearities in the data, which are then incorporated in the MNL model specification. As neural networks are being used more and more in the machine learning field, it is valuable to test and document its abilities in as many settings as possible.

Our approach to modeling the MNL and ANN is based on that of Agrawal and Schorling (1996), however, instead of brand shares, we forecast individual choices. Brand loyalty is also incorporated into both models, as opposed to just the MNL. Furthermore we improve on the training method used for the ANN by implementing regularization and softmax outputs and by making use of cross-validation to optimize parameters. The hybrid approach is based on a study by Bentz and Merunka (2000), and involves a partially connected three-layer neural network and shared weights. Forecasting performance is measured by splitting the data up into a training and test set and making use of metrics such as accuracy and negative prediction ratio.

We find that the data does contain nonlinear elements which cause the simple ANN to slightly outperform the MNL in predictive capabilities. The MNL and ANN manage to get a 72.1% and 72.3% accuracy on the test set respectively. The hybrid model detects nonlinear relations between variables and leads us to include price^2 in the MNL model. This new specification yields a 73.2% accuracy on the test set, meaning the hybrid model outperforms both the MNL and ANN.

2 Literature Review

The multinomial logit model is a well-known discrete choice model that has proven to be a useful forecasting tool in numerous applications. It has often been shown to be more suitable than other traditional statistical models such as multinomial probit (Dow and Endersby, 2004), regression (Gensch and Recker, 1979) and log-linear models (Green, Carmone, and Wachspress, 1977). In this study, we are interested in modeling individuals' choices of purchasing a specific brand of a grocery product. Previous studies on this topic using MNL models have been done by Guadagni and Little (1983) and Gonul and Srinivasan (1993).

Another model that can be used for this purpose is the artificial neural network. In the last few decades, ANN has become a popular machine learning tool. The model replicates the way the neurons in the human brain interact and learn from new data inputs. The reason why the neural network is so popular is because it is able to recognize patterns in data and classify unseen patterns by nonlinearisation. It can also be used for forecasting, which is the objective in this study.

The ANN consists of an input and output layer and, in between, one or more hidden layers. Each layer has a certain amount of nodes which are connected to nodes in other layers. Through these connections, of which each is assigned a weight, the input is transformed into an output signal. Often, biases are also included in a neural network, which are nodes that always send an output signal of 1 and do not depend on the output signals of the nodes in the previous layer. These bias nodes capture the intercept term present in a regular regression model. Without these, the nodes in the neural network would not be able to output anything other than zero, if the inputs were equal to 0. The ANN needs to be trained with a training data set before it can apply what it has learned. The most commonly used training algorithm is called the 'backpropagation' algorithm, which is based on gradient descent (Hastie, Tibshirani, and Friedman, 2009, p. 395).

An example of a paper where a discrete choice model is compared with ANN is Agrawal and Schorling (1996). The forecasted choice probabilities are aggregated to obtain market shares for each brand of the three grocery products. Another example, where the focus is on individual brand choice, is Kaya et al. (2010). This study also uses aggregated brand shares when evaluating predictive power. Both papers find that the ANN mostly outperforms the discrete choice model. However, this is not always the case. Sometimes no clear distinction in forecasting performance can be made, as is shown by Hensher and Ton (2000). It is therefore useful to compare the forecasting performance of discrete choice models and neural networks in as many settings as possible.

Other models that have been used to predict brand choices are the multinomial probit model (Paap and Franses, 2000), multiple regression (Queen, 1994) and models making use of discriminant analysis.

3 Data

To empirically compare the forecasting performance of the MNL and ANN, this study uses a data set provided by Nielsen (Jain, Vilcassim, and Chintagunta, 1994) containing information on purchase decisions of individual households for several Catsup brands. The data can be obtained from the R package *Ecdat* (Croissant, 2016). There are 300 households, each indicated by an id number, with a total amount of 2798 purchases of Catsup. The number of observations available for each household varies between 5 and 44.

The customers could choose from four products: three different sizes of Heinz (41, 32 and 28 oz) and Hunt's 32 oz. For each observation, the price of all four products is given, whether the products were featured in a newspaper advertisement and/or on display in the store at the time of purchase, and the brand the customer ended up choosing. The price of the product that the customer has purchased is the actual price paid (minus any possible discounts or coupons), while the price for the other three brands is the shelf price. In Table 1 some summary statistics of the data can be found. The fraction of times each brand is on special display in the store and is featured in an advertisement is given. We note that the brand with the biggest market share, Heinz 32, has the lowest average price and is displayed the most in the store.

Table 1: Summary statistics of Catsup data per brand

	Brand			
	Heinz 41	Heinz 32	Heinz 28	Hunt's 32
Display	0.023	0.099	0.076	0.045
Feature	0.033	0.065	0.069	0.046
Price ($)	4.634	3.143	4.316	3.355
Market share	0.065	0.521	0.304	0.110

4 Methodology

In this study we deal with a multinomial discrete random variable, namely which of the available brands of a grocery product is bought in the supermarket. This random variable, given by Y_{it}, can take on the values $j = 1, ..., J$ and indicates which alternative j household i is predicted to buy at time t. y_{it} denotes the actual purchase made. As the number of observations available varies per household, we define the number of purchases of household i as T_i. We use K explanatory variables that inform us of the marketing-mix present for each brand at the time of purchase. These are denoted by the $(1 * K)$ alternative-specific vector x_{itj}.

This study will compare the forecasting capabilities of a multinomial logit model with that of a feed-forward neural network. Our approach is based on the study done by Agrawal and Schorling (1996). We build on this study by adding a measure of brand loyalty to both the MNL and ANN and improving on the neural network's training algorithm. Moreover, we also evaluate the performance of a hybrid model that is based on both the MNL and ANN. Results for each model are obtained in R (R Core Team, 2019) and descriptions of all programmes can be found in Appendix B. In this section we will elaborate on the models and performance measures used.

4.1 Brand Loyalty

For each household, multiple observations are available. It might therefore be valuable to include a measure of a customer's brand loyalty, as this could improve the forecasting performance of both the MNL and the ANN. Agrawal and Schorling (1996) do include a brand loyalty term in their MNL specification and this estimation procedure is explained in their study. In this study, we use another measure that is introduced by Guadagni and Little (1983). The brand loyalty is calculated by exponential smoothing of past purchase decisions consumers have made, as shown by the recursive function

$$\psi_{ij}(t) = a\psi_{ij}(t-1) + (1-a)I(y_{it} = j), \tag{1}$$

where $\psi_{ij}(t)$ denotes the brand loyalty to brand j of customer i at time t, a is a smoothing constant set by the user and $I(A)$ is the identity function which is equal to 1 if A is true and 0 otherwise. For the first observation, $\psi_{ij}(1)$ is set to a if brand j was the first brand bought by customer i and $(1-a)/(J-1)$ for all other brands. This brand loyalty term is then added to the set of explanatory variables.

4.2 Multinomial Logit Model

For the MNL, we first specify a utility function containing a brand-specific intercept and the alternative-specific variables x_{itj}. The utility that household i gets from purchasing brand j at time t, where $j = 1, ..., J$, is given by

$$u_{itj} = \alpha_j + \sum_k \beta_k x_{kitj} + \zeta_{itj}. \tag{2}$$

In (2), α_j is the brand-specific intercept, x_{kitj} the marketing-mix variables consumer i experiences for brand j at time t, and β_k the parameters for each variable with $k = 1, ..., K$. Household i will only purchase brand j at time t if $u_{itj} > u_{itl}$ for all $l \neq j$. Finally, MNL requires the error term of the utility function ζ_{itj} to be identically and independently distributed and to follow a type I extreme value distribution.

As shown by McFadden (1973), given that the error distribution assumption stated above holds, the choice probabilities of MNL are given by the following formula:

$$P(Y_{it} = j|x_{itj}) = P_{itj} = \frac{\exp(\alpha_j + \sum_k \beta_k x_{itj})}{\sum_l \exp(\alpha_l + \sum_k \beta_k x_{itl})}, \tag{3}$$

where P_{itj} denotes the probability of household i choosing to buy brand j at time t. For identification, we set the choice-specific intercept $\alpha_J = 0$.

Estimation of the coefficients is done with Maximum Likelihood Estimation (MLE). The likelihood function of the MNL model is given by

$$L(\beta) = \prod_{i=1}^{N} \prod_{t=1}^{T_i} \prod_{j=1}^{J} P(Y_{it} = j|x_{itj})^{I(y_{it}=j)}, \tag{4}$$

where I(A) is an indicator function that is equal to 1 if A is true and 0 otherwise. From this likelihood function, we get the following log-likelihood

$$l(\beta) = \sum_{i=1}^{N} \sum_{t=1}^{T_i} \sum_{j=1}^{J} \log P(Y_{it} = j|x_{itj}) I(y_{it} = j). \tag{5}$$

There are numerous optimisation algorithms that can be used with MLE. We use the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm, which is a quasi-Newton method that approximates the Hessian. It is a popular method, although convergence may sometimes be slow when estimating a large amount of parameters (Mai, Toulouse, and Bastin, 2014). In our case it seems suitable, as our model will not require a big amount of parameters to be estimated.

The predicted choice probabilities for the test set can then be calculated as

$$\hat{P}_{itj} = \frac{\exp(\hat{\alpha}_j + \sum_k \hat{\beta}_k x_{kitj})}{\sum_l \exp(\hat{\alpha}_l + \sum_k \hat{\beta}_k x_{kitl})}, \tag{6}$$

where $\hat{\alpha}_j$ and $\hat{\beta}_k$ are the ML estimates. Then, for each observation in the test set, the forecasted choice will be given by the brand that has the highest predicted choice probability, thus $\hat{y}_{it} = \arg \max_j \hat{P}_{itj}$. The variance of the estimated parameters is calculated as

$$\text{Var}(\hat{\theta}) = I(\hat{\theta})^{-1} = \left(-E \left[\frac{\delta^2 l(\hat{\theta})}{\delta\hat{\theta}\delta\hat{\theta}'} \right] \right)^{-1}, \tag{7}$$

where $I(\hat{\theta})$ is the Fisher information matrix, which is equal to the negative Hessian returned during optimization. Standard errors are computed by taking the square root of the diagonal elements of the variance matrix.

A reason the MNL is a popular choice model is the closed form of the choice probabilities. Moreover, the model parameters can easily be interpreted using the (log-)odds ratio. The log odds ratio for the MNL relative to base case J is as follows

$$\log \frac{P(Y_{it} = j|x_{itj})}{P(Y_{it} = J|x_{itJ})} = \alpha_j + \sum_k \beta_k (x_{kitj} - x_{kitl}), \tag{8}$$

When all variables are kept constant except the k-th variable for alternative j, then parameter β_k can be interpreted as the change in the log of the odds ratio when x_{kitj} increases with one unit. However, a disadvantage is the Independence of Irrelevant Alternatives (IIA) property, which implies that the odds ratio between two alternatives should not depend on the existence of another alternative. Naturally, this usually does not hold in practice, which is why the multinomial probit is sometimes used as a workaround. However, for J alternatives, a $J - 1$-dimensional integral will need to be calculated in the probit model, which makes it computationally difficult when $J > 2$. For this reason the simpler logistic model is usually preferred.

4.3 Artificial Neural Network

To estimate the brand choice probabilities for each household in the testing set, we use a feed-forward single hidden layer neural network. Feed-forward means that the connections in the network do not form a cycle. The output layer contains J nodes, one for each brand, and the input layer Q nodes, one for each marketing-mix variable of each brand, the brand loyalty variables and a bias node. The optimal number of nodes in the hidden layer is determined with cross-validation. The hidden layer also contains a bias node. We make use of the backpropagation training algorithm. The input of each node will be the sum of the outputs of the previous nodes multiplied by their corresponding weights. The input of the input layer will simply be the data. This network is illustrated in Figure 1.

The transformation steps between the different layers can be summarized by the equations below:

$$z_{mit} = \sigma(x'_{it}\alpha_m), \quad m = 1, ..., M, \tag{9}$$

$$P(Y_{it} = j|x_{it}) = \phi_j(z'_{it}\beta_j), \quad j = 1, ..., J, \tag{10}$$

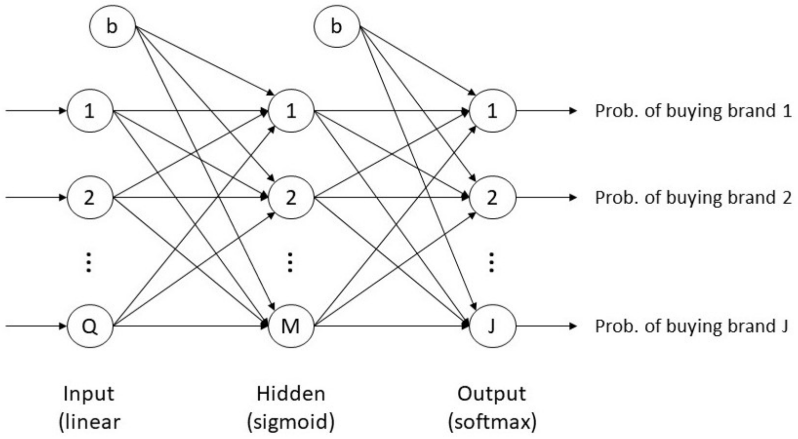

Figure 1: Three-layer feed-forward neural network with bias nodes

where x_{it} is a $(Q * 1)$ vector of input data for individual i at time t plus a bias, with corresponding weight vector α_m of size $(Q * 1)$. z_{mit} denotes the output of hidden node m, transformed by a certain activation function σ and z_{it} is the $(M * 1)$ vector $(z_{1it}, ..., z_{Mit})'$, where M is the total number of nodes in the hidden layer. In the second step, weights β_j of size $(M * 1)$ are used to transform the output of the hidden layer to input for node j in the final layer. That input is then transformed to class probabilities conditional on the input $P(Y_{it} = j|x_{it})$ using another function $\phi_j(T)$. Similar to the MNL, the forecasted choice will be the brand with the highest predicted choice probability.

For the hidden layer, we will use the sigmoid function as activation function, meaning that the input of the hidden nodes is transformed to an output value between 0 and 1.

$$\sigma(x) = \frac{1}{1 + \exp(-x)}, \tag{11}$$

This function is a popular function to use as it assures that a small change in input values will only cause a small change in output values.

It is common to use a different function for the output layer. The softmax function transforms the outputs to values summing up to one, which is very appropriate for our goal of classification, where we want the outputs of the ANN to be class probabilities. In the final layer, we will therefore make use of the softmax function

$$\phi_j(T) = \frac{\exp(T_j)}{\sum_{k=1}^{J} \exp(T_k)}. \tag{12}$$

The backpropagation algorithm updates the weights assigned to connections based on the 'loss' obtained in the previous run through the training set. The weights are updated from right to left, so starting with the connections between the last hidden layer and the output layer. As loss function we use the Cross-Entropy (CE), which is calculated as

$$R(\theta) = -\sum_{i=1}^{N}\sum_{t=1}^{T_i}\sum_{j=1}^{J} y_{itj} \log P_{itj}, \tag{13}$$

where θ denotes the set of all weights of the ANN, y_{itj} is equal to $I(y_{it} = j)$ and P_{itj} is the probability of individual i choosing to buy brand j at time t, $P(Y_{it} = j|x_{it})$, as estimated by the ANN.

To minimize $R(\theta)$ we make use of gradient descent. We therefore need to derive the gradient of (13). As θ consists of parameter vectors α and β, we take the partial derivative of $R(\theta)$ for both parameters. Below the derivation for the weights between the hidden and output layer can be seen

$$\begin{aligned}
\frac{\delta R(\theta)_{it}}{\delta \beta_{jm}} &= -\sum_{k=1}^{J} \frac{y_{itk}}{P_{itk}} \frac{\delta P_{itk}}{\delta \beta_{jm}}, \\
&= -\sum_{k=1}^{J} \frac{y_{itk}}{P_{itk}} \phi_k'(z_{it}'\beta_k) z_{mit}, \\
&= -\sum_{k=1}^{J} y_{itk}(\delta_{jk} - P_{itj}) z_{mit}, \\
&= (P_{itj} - y_{itj}) z_{mit}, \\
&= \Delta_{itj} z_{mit},
\end{aligned} \tag{14}$$

where β_{jm} is the connection weight between output node j and hidden node m, z_{mit} is the sigmoid output value of hidden node m for individual i at time t, and δ_{jk} is an indicator function that is 1 when $j = k$ and 0 otherwise.

The derivative of the loss function with respect to the weights between the input and hidden layer is given by

$$\begin{aligned}
\frac{\delta R(\theta)_{it}}{\delta \alpha_{ml}} &= -\sum_{k=1}^{J} \frac{y_{itk}}{P_{itk}} \frac{\delta P_{itk}}{\delta \alpha_{ml}}, \\
&= -\sum_{k=1}^{J} \frac{y_{itk}}{P_{itk}} \frac{\delta P_{itk}}{\delta z_{mit}} \frac{\delta z_{mit}}{\delta \alpha_{ml}}, \\
&= -\sum_{k=1}^{J}\sum_{j=1}^{J} \frac{y_{itk}}{P_{itk}} \phi_k'(z_{it}'\beta_k) \beta_{jm} \sigma'(x_{it}'\alpha_m) x_{itl}, \\
&= \sum_{j=1}^{J} \Delta_{itj} \beta_{jm} \sigma'(x_{it}'\alpha_m) x_{itl},
\end{aligned} \tag{15}$$

where α_{ml} is the connection weight between hidden node m and input node l, and x_{itl} is the input value of input node l for individual i at time t.

The initial weights are randomly drawn from the normal distribution with a mean of 0 and a standard deviation of 1 and then multiplied by 0.01 so they are close to 0, but not equal to 0. This is important, because if the weights are all equal to the same value, then the derivative of the loss function will be the same for every node. All weights are then updated with the same value causing the neural network to be symmetric. If this happens, the ANN will not perform better than a linear model. Therefore, careful selection of initial weights is crucial. It can also speed up the learning process of the ANN.

The weights are then updated for the $r+1$-th iteration as follows

$$\alpha_{ml}^{(r+1)} = \alpha_{ml}^{(r)} - \gamma \sum_{i=1}^{N} \sum_{t=1}^{T_i} \frac{\delta R(\theta)_{it}}{\delta \alpha_{ml}^{(r)}}, \tag{16}$$

$$\beta_{jm}^{(r+1)} = \beta_{jm}^{(r)} - \gamma \sum_{i=1}^{N} \sum_{t=1}^{T_i} \frac{\delta R(\theta)_{it}}{\delta \beta_{jm}^{(r)}}, \tag{17}$$

where γ denotes a specific learning rate. This rate can be set by the user. This should be done carefully, as a too large learning rate will cause the model to shoot past the minimum of the loss function, and bounce back and forth, perhaps even increasing the loss as the number of epochs increases, while a too small learning rate will require a large amount of epochs to approach the minimum.

To update the weights, the derivatives of both the sigmoid and softmax function are needed, as can be seen in equations (14) and (15). It can easily be shown that the derivative of the sigmoid is

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)). \tag{18}$$

The derivative of the softmax function with respect to its j^*-th input can be written as

$$\frac{\delta \phi_j(T)}{\delta T_{j^*}} = \phi_j(T)(\delta_{jj^*} - \phi_{j^*}(T)), \tag{19}$$

where δ_{jj^*} denotes the Kronecker delta which equals 1 if $j = j^*$ and 0 otherwise. This derivative is used to simplify (14).

It is important to note that we do not want to globally minimize $R(\theta)$, as we want to avoid overfitting. Our ANN should not be trained to get the highest accuracy on our training set, but to be able to classify unseen observations accurately. Therefore, it is necessary to implement something that will avoid this problem and thus improve model performance. Regularization is a popular way to do this. In this study we make use of the weight decay method, specifically L2 regularization. This method adds a term to the loss function that penalizes large weights. The total loss will then be $R(\theta) + \lambda J(\theta)$, with

$$J(\theta) = 1/2 \sum_{km} \beta_{km}^2 + 1/2 \sum_{ml} \alpha_{ml}^2. \tag{20}$$

The $\lambda \geq 0$ is the regularization rate, which can be set by the user. A larger value will penalize large weights more, thus making the weights shrink towards zero. To the weight update formulas (16) and (17) we then need to add the terms $-\gamma\lambda\alpha_{ml}$ and $-\gamma\lambda\beta_{jm}$ respectively. The maximum number of training epochs is set to 50000.

4.4 Hybrid Model

A big limitation of the MNL model is that the utility function is assumed to be linear. We therefore expect the ANN to outperform the MNL in forecasting when nonlinearities are present in the data. However, as Vroomen, Franses, and Nierop (2004) state, a drawback of ANN is that its weights are difficult to interpret. Unlike the MNL parameters, which can be interpreted using odds or log-odds ratios, the weights obtained after training are meaningless. This is a reason why some call the neural network a 'black box'. To circumvent these problems, we attempt to create a hybrid model that uses the best of the MNL and ANN. This hybrid approach is based on the study done by Bentz and Merunka (2000) and uses a neural network to detect nonlinearities in the data set. The nonlinearities found are then modeled by adding corresponding terms to the MNL model specification.

Bridle (1990) shows that his softmax output network with no hidden layer and shared weights is identical to the MNL. This network is partially connected, as each alternative has its own hidden nodes and connections which do not interact with each other. The weights corresponding to the same attribute for each alternative are kept equal, as the coefficients in a MNL utility function are also equal across alternatives. This model forms the base for the neural network that we use to detect nonlinear elements in the data. We consider a partially connected three-layer feed-forward neural network with shared weights and J parts for J brands. By adding a hidden layer, we gain the possibility to model choice probabilities derived from nonlinear utility functions. The ANN is therefore a generalization of the MNL. The setup of the neural network is shown in Figure 2.

We use the shared weights technique introduced by LeCun et al. (1989) where weights corresponding to the same attribute are initialized to be the same value. The weights are then updated with the average error across all alternatives. Apart from this, the same backpropagation approach is applied to this ANN as described in subsection 4.3. In principle, this means that J separate neural networks are trained at the same time, after which the outputs are merged. Naturally, for this training process we also need to find the optimal number of hidden nodes, which is equal for all alternatives, and the optimal learning and regularization rate. This is done with cross-validation.

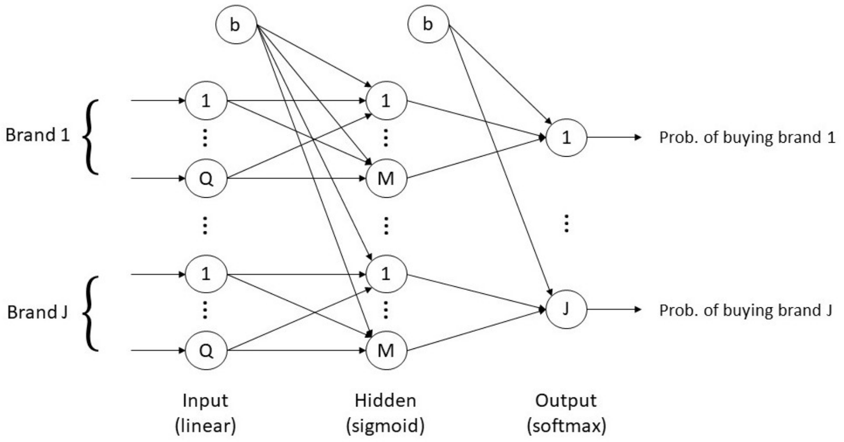

Figure 2: Partially connected neural network with shared weights and bias nodes (Q inputs and M hidden nodes for each brand)

Once the ANN has been trained, we can identify nonlinear relations between variables by plotting the utility per brand j which is given by the input of each output node against a variable, while keeping others fixed. To be able to clearly see the relations between variables we make use of Locally Weighted Scatterplot Smoothing (LOWESS), which fits a smooth line to the datapoints. After identifying which terms might be valuable additions to our model, we can confirm this by including them in the MNL and comparing the forecasting performance on the test set, in the same way as for the regular MNL model. This hybrid approach will provide us with interpretable coefficients as well as an expected increase in forecasting capabilities due to the nonlinear elements detected by the neural network.

4.5 Data Partitioning

To evaluate the forecasting performance of the MNL and ANN model, we need to compare the predictions made by both models with the actual outcomes. Therefore, we split up the data into a training set and a test set. The former will be used to estimate the coefficients of the MNL model and to train the neural network, and the latter will be used to compare the forecasts with the actual brand choices. We then use these comparisons to measure the forecasting ability of both models.

It is well-known that the amount of observations in the estimation set affects the forecasting performance of the MNL model, and the same goes for the size of the training set for ANN, as is shown by Foody, McCulloch, and Yates (1995). Therefore, three different data partitions (50/50, 65/35 and 80/20) are used to evaluate the effect that number of observations has on quality of predictions. To illustrate, with the 65/35 partitioning, the first 65% of available observations for each household will be used to estimate MNL coefficients and train the ANN, while the latter 35% will be used to determine the ability to make accurate predictions.

4.5.1 Multinomial Logit Model

To estimate the smoothing constant a in the brand loyalty term, we use the MNL with the 80/20 data split. The log-likelihood of the model is optimized using different values for a, namely 0.7, 0.75, 0.8, 0.85 and 0.9. As the goal of this study is to compare model performance, this rough estimation is sufficient (Bentz and Merunka, 2000). The optimal a found is then used to create brand loyalty terms for all MNL, ANN and hybrid models.

4.5.2 Artificial Neural Network

The ANN also contains a couple of parameters for which we need to find the optimal value, namely η and λ and the number of hidden nodes to use in the hidden layer. Therefore, we reserve a small part of the training data set for validation. This validation set will not be used for training, so to the ANN these observations will be new, or 'unseen'. As we are trying to optimize the ANN's predictive capabilities, and not how well it can fit the training data, we will use the validation set to determine the optimal number of hidden nodes, and the values of our learning rate and regularization rate. This will be done by training the ANN multiple times with different parameters, and then computing the accuracy on the validation set. The parameter combination with the maximum accuracy is then used to make the actual forecasts for the test set.

The validation set is taken to be the last 25% of available observations in the training set for all three data partitions. For the number of hidden nodes we try out the values 4 up to and including 13, as from preliminary analyses it is found that a smaller or larger number of hidden nodes does not improve model performance. The learning rates evaluated are 0.001, 0.005, 0.01, 0.05, 0.1 and 0.15, and the regularization rates 0.0001, 0.001, 0.01 and 0.1.

4.5.3 Hybrid Model

For the hybrid model, only the 80/20 data split is used so the ANN has the biggest training set possible, and thus the biggest opportunity to learn the nonlinear relations that might be present in the data. We then again create a validation set by taking 25% of the training set, and maximize the accuracy on this data set while varying the number of hidden nodes and two parameter values η and λ. As the hidden layer needs to contain the same amount of hidden nodes for each alternative, we try out 1 through 6 nodes per brand. For the learning and regularization rate, the same values as for the ANN are tested. After training the neural network and identifying nonlinear terms to include in the MNL, we again use the 80/20 data split to estimate the MNL coefficients and evaluate forecasting performance.

4.6 Performance Measures

To compare the predictive capabilities of the models, we make use of a couple of performance measures. First, we calculate the accuracy of our model, which is given by the formula

$$\text{Accuracy} = \frac{\#\text{correct predictions}}{\#\text{total predictions}} \tag{21}$$

Calculating the accuracy is a very intuitive way of measuring forecasting performance, however, solely using this measure is not a good idea. For example, when dealing with a highly skewed data set, a model predicting only the most common class would score a relatively good accuracy, while it is not a good model. In our data set, a model that solely predicts Heinz 32, would score an accuracy of over 50%.

The Negative Prediction Ratio (NPR) might therefore also be a useful performance measure, as it uses the number of times that the predicted probability of the actual choice was the lowest out of all predicted probabilities. It is calculated as

$$\text{NPR} = 1 - \frac{\#\text{times actual choice has lowest predicted probability}}{\#\text{total predictions}} \tag{22}$$

A confusion matrix, however, will most likely give us the most insight into the predictive capability of each model. The confusion matrix cross references the actual choice with the predictions, allowing us to see how many times each choice is correctly (or wrongly) predicted for each brand.

5 Results

In this section we present and discuss the results for each of the models separately. Then, making use of the earlier specified performance measures, we compare them and determine which is better suited to predict individuals' choices in purchasing Catsup. There are $J = 4$ Catsup brands, namely Heinz 41 (1), Heinz 32 (2), Heinz 28 (3) and Hunt's 32 (4).

5.1 Multinomial Logit Model

The brand loyalty smoothing constant a is found by estimating the MNL model multiple times for different values of a and choosing the one for which the negative log-likelihood is minimal. Using 80% of the data as the training set, we find $a = 0.75$ is the optimal smoothing constant and therefore use this to create four brand loyalty terms for each brand. These terms are used in all models that include brand loyalty that follow.

We estimate the MNL model multiple times for three different data partitions, both including and excluding the brand loyalty terms to assess the increase in forecasting performance that the addition of these terms brings forth. Table 2 shows the accuracy and negative prediction ratio on both the estimation and test set for all models.

Table 2: Forecasting performance of different multinomial logit model specifications

	Training set		Test set	
Data split	Accuracy	NPR	Accuracy	NPR
No brand loyalty				
80/20	0.619	0.952	0.617	0.951
65/35	0.619	0.956	0.614	0.946
50/50	0.624	0.963	0.600	0.938
Brand loyalty				
80/20	0.724	0.967	0.721	0.989
65/35	0.732	0.970	0.711	0.978
50/50	0.752	0.980	0.705	0.969

It is clear to see that the addition of brand loyalty terms has greatly increased the model's predictive capability. Furthermore, it is worth noting that the accuracy on the training set seems to go up when we use a smaller training set, while the test set accuracy goes down. This is most likely caused by the skewness of the data set and the way the data partitioning is done.

The 80/20 MNL including brand loyalty yields the highest accuracy and negative prediction ratio on the test set out of all models. We therefore take a closer look at this model's estimated coefficients, which can be found in Table 3.

Table 3: Multinomial logit model estimates for 80/20 data split

	α_1	α_2	α_3	Display	Feature	Price	Brand loyalty
Coeff.	1.80*	0.73*	2.31*	1.08*	1.25*	-1.39*	2.52*
S.E.	0.15	0.09	0.12	0.12	0.14	0.07	0.10
Log-lik.	1624.27						

Note: * Significant at 5% significance level

The intercept term for Hunt's 32 (α_4) is set to 0 for identification. All coefficient values are according to our expectations, as we only expect an increase in price to have a negative effect on the probability of choosing to buy a product. When the price of, for example, Heinz 41 goes up by one unit, then the log of the odds ratio of choosing Heinz 41 over Hunt's 32 goes down with -1.39.

In Table 4 the confusion matrix of the predictions on the test set is shown and we see that the brand with the smallest market share, Heinz 41, is almost never predicted. However, in the rare case that it is, the model usually gets it right. From all observations predicted to be Heinz 32, 80.5% is correct. Overall, the MNL does a relatively good job

Table 4: Confusion matrix test set predictions by 80/20 MNL with brand loyalty

		Reference			
		Heinz 41	Heinz 32	Heinz 28	Hunt's 32
Prediction	Heinz 41	6	0	1	0
	Heinz 32	8	215	31	13
	Heinz 28	25	44	148	22
	Hunt's 32	1	3	4	24

at modeling customer's preferences in buying Catsup.

5.2 Artificial Neural Network

Modeling the ANN is quite a computationally expensive task, as the model has to be trained multiple times to find the optimal combination of parameters and each training consists of 50000 epochs. We use a fully-connected three-layer feed-forward neural network with sixteen input nodes and a bias node in the input layer, and four output nodes, each giving the choice probability of the corresponding brand. The data is split up into three sets, namely the training, validation and test set. For each of these subsets we calculate the loss, accuracy and negative prediction ratio after training with the optimal parameter combination. These results are shown in Table 5.

Table 5: Forecasting performance of ANN with three data partitions

	Training set			Validation set			Test set		
Data split	Loss	Accuracy	NPR	Loss	Accuracy	NPR	Loss	Accuracy	NPR
80/20[1]	0.645	0.762	0.977	0.805	0.718	0.961	0.720	0.703	0.989
65/35[2]	0.522	0.801	0.986	0.848	0.700	0.944	0.777	0.723	0.980
50/50[3]	0.502	0.809	0.982	0.801	0.710	0.982	0.815	0.709	0.967

[1]: # hidden nodes = 10, $\eta = 0.05$, $\lambda = 0.001$
[2]: # hidden nodes = 7, $\eta = 0.1$, $\lambda = 0.0001$
[3]: # hidden nodes = 12, $\eta = 0.05$, $\lambda = 0.0001$

The loss calculated for the ANN models includes the regularization loss $J(\theta)$ which depends on the neural network connection weights. As these weights are different for each ANN and are randomly initialized, we have to be careful in comparing the loss across models.

We see that the accuracy on the validation and test set is always lower than that of the training set. This means that our ANN is still overfitting to some extent and can still improve on its ability to generalize. As with the MNL, we find that for a smaller training set the accuracy on the training set is higher than for a larger set. Noteworthy is that the neural network trained with the largest training set scores the lowest accuracy on the training and test set, but the highest on the validation set. Maximizing validation accuracy is used to determine the optimal combination of parameters, however it seems as if this leads to a neural network that is overfitting on the validation set.

While the 50/50 ANN does fit well onto the training set, it also has the largest decrease in accuracy compared with the test set out of the three models. The 65/35 neural network outperforms the other two data partitions on this performance metric. Therefore, we will use this ANN in the comparison across all models. With the help of regularization, the weights have stayed close to 0. The weight matrices for the 65/35 ANN can be found in Appendix A.1.

Table 6: Confusion matrix test set predictions by 65/35 ANN with brand loyalty

		Reference			
		Heinz 41	Heinz 32	Heinz 28	Hunt's 32
Prediction	Heinz 41	14	1	4	0
	Heinz 32	18	375	50	21
	Heinz 28	43	71	258	34
	Hunt's 32	2	14	15	65

In Table 6 the confusion matrix is shown for the neural network with the highest accuracy on the test set. From these results, and those for the 80/20 MNL, we see that correctly predicting the Heinz 41 and Hunt's 32 categories is a difficult task for both models. This makes sense as these brands have the smallest market shares, thus there are not many observations that the ANN can learn from.

5.3 Hybrid Model

The first step of the hybrid approach is to train the partially connected neural network that we specified earlier. This is done with the first 80% of all available observations for each household. Using cross-validation we find that the optimal number of hidden nodes to use per alternative is 3, meaning that the hidden layer contains 12 nodes. The optimal learning rate is $\eta = 0.15$ and the regularization rate $\lambda = 0.0001$. We train the neural network using a maximum of 50000 epochs. Then we extract the predicted utilities for all households and purchase occasions from the ANN by taking the input of each output node. By plotting these inputs on the explanatory variables, we can discover any nonlinear relations that are present in the data set.

We focus on plotting the utility function for Heinz 32 as this brand is purchased the most and is therefore the most evenly distributed. However, the same effects are found to a smaller extent for the other three brands. Multiple relations between variables have been assessed and the important and relevant ones have been highlighted in this section.

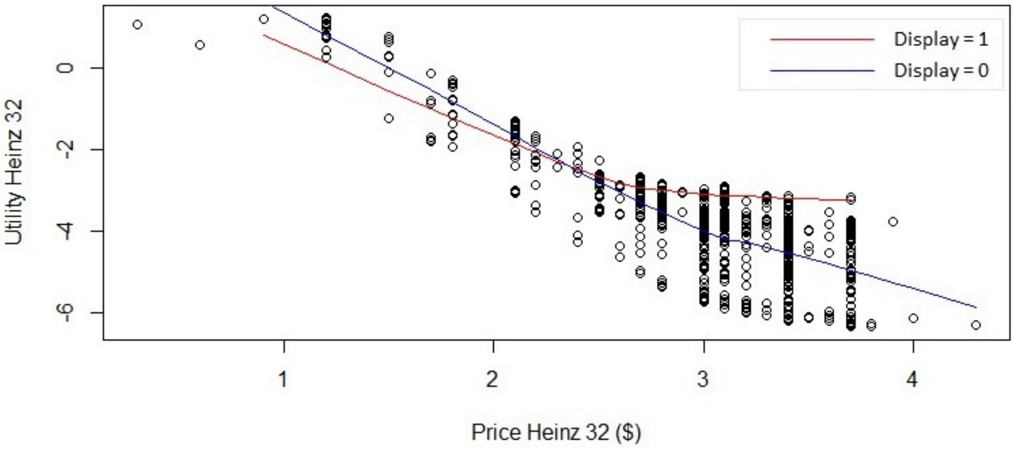

Figure 3: Utility for Heinz 32 plotted against price given whether product was on display during purchase

In Figure 3 the predicted utility of buying Heinz 32 would yield is plotted against its price. It is obvious to see that the LOWESS line is not linear, but instead starts to get less steep from a certain price point upwards. To capture this nonlinear aspect we can include price squared in the MNL as an additional explanatory variable.

The utility is plotted against price conditional on the display variable, which is 1 if the product was on display at the time of purchase and 0 if not. The difference in slopes of the lines indicates that there is some sort of relation between the two marketing-mix variables. When a product is on display then an increase in price has a smaller negative effect on utility than when it is not on display. Therefore another potential variable would be price multiplied by the display dummy.

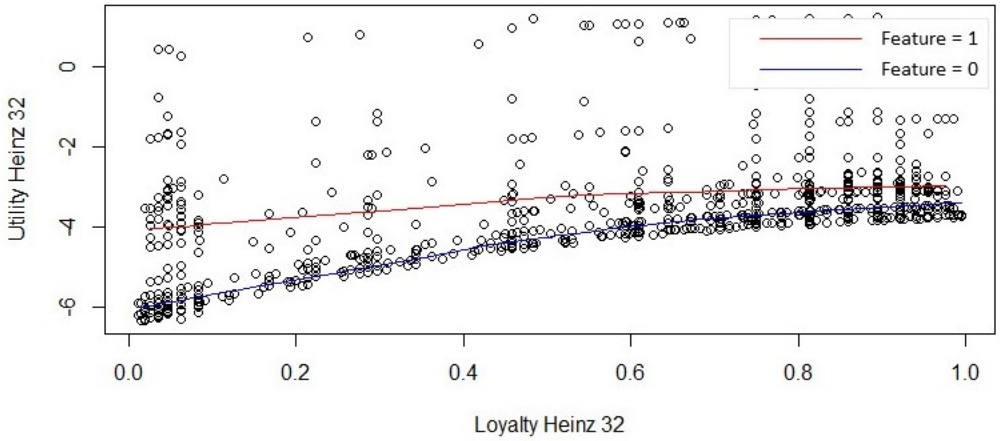

Figure 4: Utility for Heinz 32 plotted against brand loyalty given whether product was featured during purchase

Finally, from Figure 4 we conclude that when a product is featured in an advertisement the brand loyalty customers feel towards that brand matters less to their utility than when it is not featured. Therefore, loyalty multiplied by feature is another potential extra variable. The plot for utility against loyalty conditional on display, which can be seen in Appendix A.2, looks nearly identical to Figure 4. Loyalty multiplied by display is therefore also an option. As before, the nonlinear LOWESS lines indicate that loyalty squared might also be a good addition.

We have identified 5 potential additions to our MNL model, namely price^2, price*display, loyalty^2, loyalty*feature and loyalty*display. After some preliminary logistic regressions, we settle on evaluating the following model specifications, all with a 80/20 data partitioning:

a: Original model + price^2

b: Original model + brand loyalty^2

c: Original model + brand loyalty*feature

d: Original model + price^2 + brand loyalty*feature

Various model specifications have been analyzed and only those that did not decrease the training set accuracy have been selected to investigate further.

Table 7: Forecasting performance of several 80/20 MNL model specifications

	Original MNL	a	b	c	d
Accuracy	0.721	0.732	0.721	0.721	0.732
NPR	0.989	0.989	0.991	0.989	0.989
Log-lik.	1624.27	1591.83	1621.06	1623.55	1590.92

In Table 7 the accuracy and negative prediction ratio on the test set is shown for all 4 model specifications. Only model specifications a and d outperform the original model in terms of accuracy. However, the brand loyalty*feature term is not significant at a 5% significance level, as can be seen in Appendix A.3. Therefore, we will only take a closer look at specification a.

Table 8: Multinomial logit model estimates for 80/20 data split: specification a

	α_1	α_2	α_3	Display	Feature	Price	Brand loyalty	Price^2
Coeff.	1.77*	0.74*	2.31*	1.06*	1.26*	−3.70*	2.55*	0.31*
S.E.	0.16	0.10	0.13	0.12	0.15	0.38	0.10	0.05
Log-lik.	1591.83							

Note: * Significant at 5% significance level

A possible reason that the coefficient for price^2 is positive is that when the price of a product is relatively high, the customer might think this is an indication of quality and will therefore be willing to purchase the more expensive Catsup. With additional data on the perception of product quality that consumers have, this suspicion could be confirmed.

Table 9: Confusion matrix test set predictions by 80/20 hybrid model

		Reference			
		Heinz 41	Heinz 32	Heinz 28	Hunt's 32
Prediction	Heinz 41	6	0	2	0
	Heinz 32	8	215	25	11
	Heinz 28	25	43	154	24
	Hunt's 32	1	4	3	24

5.4 Model Comparison

After determining for the MNL, ANN and hybrid model which parameters, data partitions and additional variables lead to the best accuracy on the test set, we can now decide on which model is most appropriate for our goal of forecasting individual brand choice.

Table 10: Comparison forecasting performance MNL, ANN and hybrid model

	MNL[1]	ANN[2]	Hybrid[3]
Accuracy	0.721	0.723	0.732
NPR	0.989	0.980	0.989

[1]: Data partition: 80/20, including brand loyalty
[2]: Data partition: 65/35, # hidden nodes = 7, $\eta = 0.1$, $\lambda = 0.0001$
[3]: Data partition: 80/20, including brand loyalty and price^2

In Table 10 the three models and their forecasting performance metrics are shown. As the ANN slightly outperforms the MNL we can conclude that there are indeed nonlinearities in the data set that should be taken into account. The hybrid model outperforms both the MNL and ANN with respect to accuracy and has a negative prediction ratio equal to that of the MNL. This means that the hybrid approach, where nonlinear relations are first analyzed after which nonlinear terms can be added to the utility function specification of the MNL, is the most suitable for modeling individual brand choice with this data set. However, this does not mean that the hybrid model will outperform the other two every time when considering similar scanner data sets. Neural networks are very case-specific and sometimes certain data needs a specific ANN architecture. Nevertheless this result is still useful as empirical result for this specific data set, and as stepping stone for future research in the same domain.

6 Conclusion

The goal of this study was to compare the forecasting performance of a multinomial logit model, an artificial neural network and a hybrid model that combines the two. To do this, we used a scanner data set containing information on the Catsup purchasing behaviour of 300 households. Four different brands could be bought, namely Heinz 41, Heinz 32, Heinz 28 and Hunt's 32, and there were in total 2798 observations. The data was split into a training and a test set and three different data partitions were used to evaluate the effect of the length/size of the training set on forecasting performance. Out of three models, we found that the hybrid model based on the approach of Bentz and Merunka (2000) is most suitable for forecasting individual brandchoice with this data set.

The first model applied was the multinomial logit model. This discrete choice model assumes that the utility function is a linear function, and thus limits its own predictive capability when there are nonlinear relations between variables. The 80/20 data partition led to an accuracy on the test set of 72.1%. We then also trained an artificial neural network with one hidden layer containing 7 hidden nodes and a bias on the training set of the 65/35 data partition. This yielded an accuracy of 72.3%, which is only slightly better than the MNL model. However, this did tell us that nonlinearities were present in the data. This was confirmed by our third and final model: the hybrid model that combines the best of both worlds. First, a partially connected three-layer ANN with shared weights was used as a diagnostic tool that detects nonlinearities in the data. Plotting the ANN utilities against the explanatory variables confirmed the existence of several nonlinear relations between variables. After trying out several MNL model specifications, we found that the original MNL model plus the term price^2 yields 73.2% accuracy. Therefore, this hybrid model has greater predictive capabilities than the MNL and ANN, regarding this forecasting purpose and specific data set, and is therefore most appropriate for our aim.

One problem we ran into is the level of skewness in the data itself. Logically, with scanner data on grocery products, there are popular and less popular brands. However, this imbalance makes it hard for all models to properly identify the effects of each variable on the choice, and the relations between them. Sampling with replacement to create equal shares for all brands could be a solution to this problem.

Furthermore, the ANN we used in this study was relatively simple and thus biased on the training set. In future research, using a more complex ANN that can avoid overfitting is advised. Moreover, another data set or extra explanatory variables could help increase the accuracy of all three models. More customer characteristics or time and day of purchase would maybe further improve forecasting performance. Finally, different weight initialization techniques could be employed as the initial connection weights used in an ANN greatly affect the outcome. Incorporating these improvements would certainly improve forecasting performance even more.

References

Agrawal, D. and Schorling, C. (1996). “Market Share Forecasting: An Empirical Comparison of Artificial Neural Networks and Multinomial Logit Model”. In: *Journal of Retailing* 72.4, pp. 383–407.

Bentz, Y. and Merunka, D. (2000). “Neural Networks and the Multinomial Logit for Brand Choice Modelling: A Hybrid Approach”. In: *Journal of Forecasting* 19.3, pp. 177–200.

Bridle, J. S. (1990). “Probabilistic Interpretation of Feedforward Classification Network Outputs, With Relationships to Statistical Pattern Recognition”. In: *Neurocomputing*. Springer, pp. 227–236.

Croissant, Y. (2016). *Ecdat: Data Sets for Econometrics*. R package version 0.3-1. URL: `https://CRAN.R-project.org/package=Ecdat`.

Dow, J. K. and Endersby, J. W. (2004). “Multinomial Probit and Multinomial Logit: A Comparison of Choice Models for Voting Research”. In: *Electoral Studies* 23.1, pp. 107–122.

Foody, G., McCulloch, M., and Yates, W. (1995). “The Effect of Training Set Size and Composition on Artificial Neural Network Classification”. In: *International Journal of Remote Sensing* 16.9, pp. 1707–1723.

Gensch, D. H. and Recker, W. W. (1979). “The Multinomial, Multiattribute Logit Choice Model”. In: *Journal of Marketing Research* 16.1, pp. 124–132.

Gonul, F. and Srinivasan, K. (1993). “Modeling Multiple Sources of Heterogeneity in Multinomial Logit Models: Methodological and Managerial Issues”. In: *Marketing Science*, pp. 213–229.

Green, P. E., Carmone, F. J., and Wachspress, D. P. (1977). “On the Analysis of Qualitative Data in Marketing Research”. In: *Journal of Marketing Research* 14.1, pp. 52–59.

Guadagni, P. M. and Little, J. D. (1983). “A Logit Model of Brand Choice Calibrated on Scanner Data”. In: *Marketing Science* 2.3, pp. 203–238.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Second. New York, USA: Springer, pp. 389–414.

Hensher, D. A. and Ton, T. T. (2000). “A Comparison of the Predictive Potential of Artificial Neural Networks and Nested Logit Models for Commuter Mode Choice”. In: *Transportation Research Part E: Logistics and Transportation Review* 36.3, pp. 155–172.

Jain, D. C., Vilcassim, N. J., and Chintagunta, P. K. (1994). “A Random-Coefficients Logit Brand-Choice Model Applied to Panel Data”. In: *Journal of Business & Economic Statistics* 12.3, pp. 317–328.

Kaya, T. et al. (2010). “Modeling Toothpaste Brand choice: An Empirical Comparison of Artificial Neural Networks and Multinomial Probit Model”. In: *International Journal of Computational Intelligence Systems* 3.5, pp. 674–687.

LeCun, Y. et al. (1989). “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4, pp. 541–551.

Mai, A. T., Toulouse, M., and Bastin, F. (2014). *On Optimization Algorithms for Maximum Likelihood Estimation.* CIRRELT.

McFadden, D. et al. (1973). “Conditional Logit Analysis of Qualitative Choice Behavior”. In: *Frontiers of Econometrics*, pp. 105–142.

Paap, R. and Franses, P. H. (2000). “A Dynamic Multinomial Probit Model for Brand Choice with Different Long-Run and Short-Run Effects of Marketing-Mix Variables”. In: *Journal of Applied Econometrics* 15.6, pp. 717–744.

Queen, C. M. (1994). “Using the Multiregression Dynamic model to Forecast Brand Sales in a Competitive Product Market”. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 43.1, pp. 87–98.

R Core Team (2019). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. Vienna, Austria. URL: `https://www.R-project.org/`.

Vroomen, B., Franses, P. H., and Nierop, E. van (2004). “Modeling Consideration Sets and Brand Choice Using Artificial Neural Networks”. In: *European Journal of Operational Research* 154.1, pp. 206–217.

Appendix A Results

A.1 ANN Connection Weights

Table 11: 65/35 ANN connection weights after training: input and hidden layer

	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7
Display Heinz 41	−0.62	−0.41	−0.44	−0.46	−0.92	−0.53	−0.96
Display Heinz 32	−0.65	0.20	0.59	0.37	0.17	−0.25	0.00
Display Heinz 28	−0.52	−0.68	1.75	−0.85	−0.56	−1.19	−1.43
Display Hunt's 32	0.94	−2.36	0.90	0.45	−0.91	−0.12	−1.28
Feature Heinz 41	0.19	−0.27	0.14	1.22	1.02	0.43	0.25
Feature Heinz 32	−1.37	0.36	0.65	−0.62	−0.81	2.71	1.84
Feature Heinz 28	−1.11	0.92	1.97	0.78	−0.82	1.45	−1.83
Feature Hunt's 32	−1.08	−1.95	0.33	−0.95	−0.73	1.00	0.26
Price Heinz 41	0.91	−0.43	1.43	−2.65	1.08	0.55	1.54
Price Heinz 32	−4.73	−1.56	−0.11	−0.05	−1.38	−0.12	−0.83
Price Heinz 28	1.75	−0.36	−0.49	2.32	−0.12	−1.97	0.42
Price Hunt's 32	0.16	3.04	−0.17	0.68	−0.58	1.77	−1.02
Loyalty Heinz 41	0.29	0.94	−2.84	−0.12	0.66	0.07	−2.99
Loyalty Heinz 32	−0.26	0.54	−0.14	0.76	0.13	2.95	1.40
Loyalty Heinz 28	−0.92	0.43	0.51	−0.28	0.88	−0.30	−1.84
Loyalty Hunt's 32	0.43	−1.03	−0.56	−1.39	−1.12	−1.64	1.91
Bias	−0.47	0.89	−3.04	−1.01	0.57	1.09	−1.52

Table 12: 65/35 ANN connection weights after training: hidden and output layer

	Brand 1	Brand 2	Brand 3	Brand 4
Node 1	−0.16	3.77	−1.26	−2.33
Node 2	2.22	−0.22	1.57	−3.58
Node 3	−4.44	1.40	4.14	−1.11
Node 4	−0.96	3.31	−0.51	−1.84
Node 5	2.70	0.32	−1.38	−1.61
Node 6	0.86	3.02	−1.49	−2.40
Node 7	−2.85	2.41	−2.91	3.35
Bias	0.83	−5.06	0.90	3.33

A.2 Scatterplot Utility of Hybrid Model

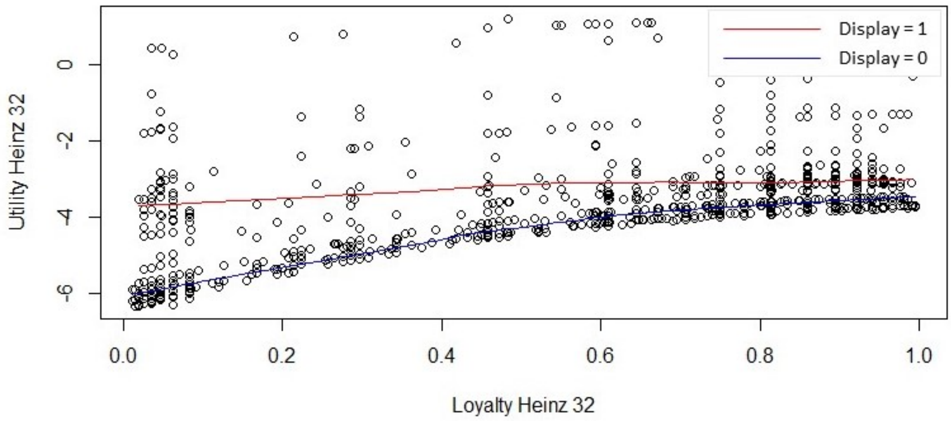

Figure 5: Utility for Heinz 32 plotted against brand loyalty given whether product was on display during purchase

A.3 Hybrid Multinomial Logit Model Estimates

Table 13: Multinomial logit model estimates for 80/20 data split: specification *d*

	α_1	α_2	α_3	Display	Feature	Price	Brand loyalty	Price^2	Brand loyalty*Feature
Coeff.	1.77*	0.74*	2.31*	1.06*	1.42*	−3.71*	2.59*	0.31*	−0.65
S.E.	0.16	0.10	0.13	0.12	0.18	0.39	0.10	0.05	0.47
Log-lik.	1590.92								

Note: * Significant at 5% significance level

Appendix B Code

All the code for this thesis was written in R and can be found in the zip-file “Thesis Code 443139.zip”. Below a list can be found of all programmes included:

- `MNL.R` Multinomial logit model that forecasts individual brand choice using Catsup scanner data and a data partitioning set by the user. Also optimizes brand loyalty smoothing constant.

- `ANN.R` Artificial neural network predicts individual brand choice with neural network with optimal parameter combination and a data partitioning set by the user.

- `Hybrid.R` Hybrid model programme trains several neural networks, chooses the one with best parameter combination and then plots consumer utility (given by ANN) against variable conditional on others. User can then add terms to MNL specification and make predictions. Uses 80/20 data split.