Erasmus University Rotterdam

Erasmus School of Economics

Seminar Case Studies in Quantitative Marketing (FEM21001)

MSc Econometrics and Management Science

Predicting the Bankruptcy Probability of Restaurants in Zip Code Regions using a Two-Stage Clustering and a Generalized Linear Hierarchical Model Approach

Authors: (Group 19)
Max Broers (471390)
Bart Dubbeldam (480675)
Stefan Lam (481922)
Yin Liu (473979)

Supervisor:
Dr. Andreas Alfons
Second assessor:
Dr. Wendun Wang

July 5, 2022

Abstract

Since location plays an important role in a restaurant's success, determining the potential of a region is valuable information for both restaurant owners and sales representatives. This paper aims to forecast the probability of bankruptcy of restaurants in certain zip code regions. We apply a two-step approach, where we first cluster the zip codes into different types of zip code regions. This is done by a two-stage clustering approach using a Self Organising Map (SOM) and a partitional clustering algorithm such as k-means or Gaussian Mixture Models (GMM). Next, we use the resulting clusters in various Generalized Linear Hierarchical (GLH) multinomial logit models, which are estimated using a Bayesian approach, to forecast the bankruptcy of restaurants. We use regional and restaurant data provided by the Central Bureau of Statistics and Unilever Food Solutions, respectively. We find that our clustering procedure is able to successfully distinguish the characteristics of urban agglomerations from other zip code regions in the Netherlands. However, the GLH models turn out to perform poorly and are thus deemed as unreliable.

Contents

1 Introduction

One of the most important characteristics that determine the success of food serving establishments (restaurants, hotels, caterers, etc) is their location. In particular, the long-term commitment of location choice must not be overlooked (Yang and Lee, 1997), since a good choice of location can lead to an increase in market share and higher profitability (Chou et al., 2008; Chen and Tsai, 2016). Therefore, it is of high value to determine the potential of a region, when deciding where to open a new restaurant.

This information is useful for both restaurant owners to determine where to (re-)locate their business and for marketers or sales managers to identify new marketing opportunities and develop more locally targeted strategies. For this reason, we aim to construct models that predict the probability of a new restaurant going bankrupt in a certain zip code region, since this probability is a good indication of the potential of a zip code region. We thus formulate the following research question: *"In what type of zip code region do restaurants have a smaller probability of going bankrupt?"*. Our research question will be answered using the following subquestions:

1. *"How many different types of zip code regions can we divide our zip codes into?"*
2. *"How does the probability of bankruptcy vary between the different types of zip code regions?"*

These subquestions are answered by implementing a two-step approach. In the first step, we answer subquestion (1) by finding an optimal clustering for the zip codes based on demographics and geographical descriptives. In particular, we implement k-means and Gaussian Mixture Models (GMM) to cluster zip codes. In addition, these methods are extended to a two-stage procedure as proposed by Vesanto and Alhoniemi (2000). The first stage of the two-stage procedure consists of mapping the samples to prototypes using a Self Organising Map (SOM). Then, in the second stage k-means or GMM is used to cluster these prototypes. We refer to these two variants of Two-Stage (TS) clustering procedures as TS k-means and TS GMM respectively.

Next, in the second step, we answer subquestion (2) by using the obtained zip code regions from the first step in a Generalized Linear Hierarchical (GLH) multinomial logit model, which aims to predict whether a restaurant is open, temporarily closed or permanently closed. We propose three different models which are "built-up" from each other in the following order: the Pooled model, the Unpooled model and the Simultaneous Unpooled (SU) model. These models are estimated using a Bayesian approach.

We illustrate our research by using open source regional zip code specific data provided

by the Central Bureau for Statistics (CBS), as well as restaurant-specific data provided by Unilever Food Solutions (UFS). First, using the regional data we find for the first step that TS k-means is the best clustering method and conclude that this clustering method is able to divide the zip codes into two zip code regions, where the two regions are able to distinguish the characteristics of the four biggest urban agglomerations (Randstad) from other areas in the Netherlands. Secondly, using the restaurant and regional data, we observe that in the second step our GLH models perform poorly, which leaves us unable to reliably answer our second subquestion. Further inspection of the data and our models is required.

This paper is structured as follows. In Section 2, relevant related work is discussed. Section 3 provides an overview of the used data and the undertaken preprocessing steps. In Section 4 we describe the used framework. The results of our research are presented in Section 5. Finally, concluding remarks, limitations and future research possibilities are given in Section 6.

2 Literature review

In this section we provide a theoretical framework for the topics under investigation. First, in Section 2.1 we discuss factors that determine the success of a restaurant. Next, Section 2.2 summarises the clustering techniques considered in this paper. Section 2.3 and 2.4 provide a summary on hierarchical linear models and Bayesian estimation methods. Lastly, Section 2.5 discusses how to use imputation to deal with missing data.

2.1 Factors of restaurant success

There exist several factors that determine whether a restaurant will become successful. For example, Baraban and Durocher (2010) note that factors such as interior design, service and quality of the food are extremely important for determining restaurant success, they also mention location as an important factor. Moreover, Jain et al. (1979) point out that an establishment's location is hardly a feature that can be undermined, as opposed to, for example promotions which can be copied. Apart from being an important factor that leads to restaurant success, it is much more difficult to adjust the location of a restaurant than adjusting the price, quality of the food, service and promotions. Given that around 60% of restaurants fail by the end of their third year (Agarwal and Dahm, 2015), the importance of optimally selecting the location for a restaurant should not be underestimated. In particular, Yang et al. (2017) show that there is a significant relationship between the location of restaurants and several neighbourhood sociodemographic characteristics, such as average household size, median income, age, race, gender and neighbourhood urbanization.

Current literature highlights several key factors in determining a successful location for a

restaurant. For example, Parsa et al. (2015) note that a substantial population of apartment dwellers and transient residents increases the probability of success of a restaurant. Apart from this, they also find that it is best to avoid locations where restaurants have already failed. In their study the authors split up a city into zip code regions and identify notable factors for the zip codes with a high or low failure rate. Another common approach in current literature is to conduct a survey with demographic factors that could potentially lead to the success of a restaurant. For example, Tzeng et al. (2002) propose a multicriteria ranking solution based on the opinions of multiple experts. However, quantitative approaches seem to be rarely used in current literature.

2.2 Clustering

Cluster analysis involves grouping similar objects into distinct and mutually exclusive subsets, referred to as clusters (Mangiameli et al., 1996). There are two popular types of clustering techniques: hierarchical and partitional clustering techniques. Hierarchical techniques seek to build a hierarchy of clusters in a greedy manner (Mirkin, 2012), while partitional techniques aim to minimise a cluster criterion by iteratively relocating data points between clusters until an optimal partition is attained (Popat and Emmanuel, 2014).

In the current literature, two commonly used partitional unsupervised clustering techniques are k-means (MacQueen et al., 1967) and GMM (McLachlan and Basford, 1988), which correspond to hard and soft clustering, respectively. These techniques have a time complexity of $\mathrm{O}(kN)$ and both work well for spherical shaped clusters (Popat and Emmanuel, 2014). The difference between these techniques is that k-means assumes that each cluster is represented by the mean of its cluster members, while GMM assumes that each cluster is represented by a Gaussian distribution with a corresponding mean and covariance. This results in more flexible decision boundaries with GMM, since the boundaries can be elliptical due to the covariance as opposed to only circular boundaries with k-means.

Hierarchical clustering methods can be categorised into agglomerative or divisive clustering methods. In the general case, the time complexity of agglomerative clustering is $\mathrm{O}(N^3)$, while divisive clustering is $\mathrm{O}(2^N)$ (Popat and Emmanuel, 2014). For this reason, agglomerative techniques are more common than divisive techniques. However, in practice, these techniques are still too slow for large data sets compared to the more efficient partitional clustering methods. Moreover, a major advantage of partitional clustering is the ability to gradually improve the clustering quality through an iterative optimization process, while this cannot be done in standard hierarchical clustering, since we cannot revisit the merges that were already completed due

to the hierarchical structure as discussed by Popat and Emmanuel (2014).

A method similar to hierarchical clustering methods is the SOM network as proposed by Kohonen (1990), which is a two-layer artificial neural network, consisting of an input and a Kohonen (or output) layer as illustrated in Figure 1. The Kohonen layer is a 2-dimensional discrete grid of M map units, where each unit is connected to adjacent ones by a neighbourhood relation. Specifically, the SOM network is able to non-linearly cluster groups of similar input patterns from a high-dimensional input space onto the low-dimensional map units in the Kohonen layer, such that the topological structure of the data is preserved. This is done by using a competitive unsupervised learning procedure based on the Kohonen learning rule (Kohonen, 1990), which gives similar input patterns to map units located close to each other in the Kohonen Layer. The discrete output lattice of SOM can be seen as a hierarchical representation of the data (Popat and Emmanuel, 2014).

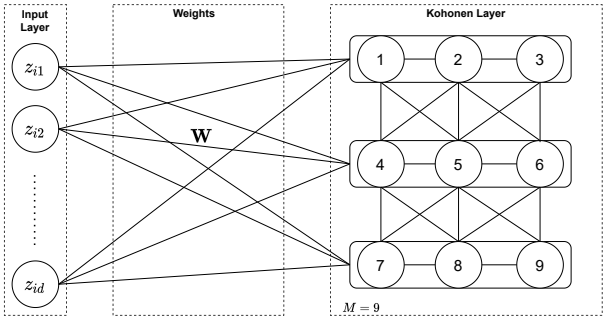

Figure 1: Schematic overview of the SOM network with $M = 9$ map units, where $\mathbf{W}$ is the weight matrix of the network.

Furthermore, the SOM network, contrary to standard hierarchical clustering methods, can improve the quality of decisions that require the cluster analysis of "messy data", e.g. in market segmentation, credit analysis or operations problems according to Mangiameli et al. (1996). This is due to the superiority of the SOM network over hierarchical techniques, when the data has (1) a high dispersion of clusters, (2) outliers or (3) irrelevant variables as discussed by Mangiameli et al. (1996).

Moreover, Vesanto and Alhoniemi (2000) suggest to use the SOM network in a two-stage approach, where the data is first clustered into "prototypes" using the SOM network, and then the SOM is clustered using a partitional clustering technique, such as k-means or GMM as shown in Figure 2. As discussed by Vesanto and Alhoniemi (2000), the two advantages of this approach is (1) the considerable decrease in computational load making it possible to cluster large data sets in a limited amount of time, and (2) the noise reduction as the prototypes are local averages of the data making it less sensitive to random variations than the original data,

which results in a more robust clustering method against outliers.

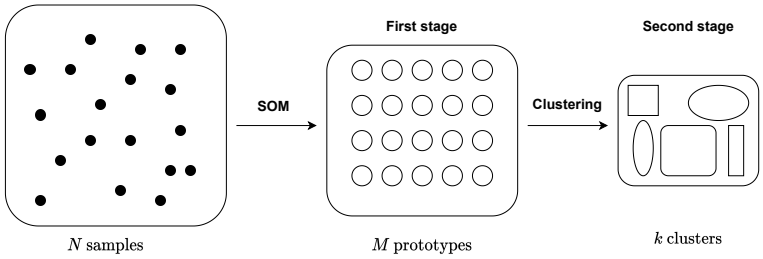

Figure 2: Schematic overview of the two-stage procedure for clustering.

2.2.1 Cluster validation

One of the main challenges in cluster analysis is determining the optimal number of clusters in the data by assessing the quality of the formed clusters. Since there are no pre-defined classes to which our data belongs, we focus on validating our clusters through relative clustering validation techniques, which vary the number of clusters to determine the optimal number of clusters (Schalkoff, 2007; Charrad et al., 2014).

A commonly used clustering validation technique is the elbow method (Bholowalia and Kumar, 2014), where the distance between observations of the same cluster is examined using the within-cluster sum of squared residuals (WSS). The WSS is computed by the sum of the distance of each point to its predicted cluster center. The optimal number of clusters is found when the marginal gain of adding another cluster, the decrease in WSS, drops drastically. The drastically decrease in WSS is observed as the inflection point on the curve, which leads to an elbow-like shape. However, this inflection point may be hard to identify (Ketchen and Shook, 1996), which could cause the obtained number of clusters to be unreliable. Furthermore, the elbow method does not take the distance between different clusters into account, unlike methods like the Silhouette coefficient (Rousseeuw, 1987) and the Davies-Bouldin (DB) index (Davies and Bouldin, 1979). In particular, the Silhouette coefficient is used as an indication of how accurately individual data points are assigned to their clusters, whereas the DB index evaluates intra-cluster similarity and inter-cluster differences.

To visually validate the resulting clusters, dimensionality reduction can be used to project high-dimensional input data onto two dimensions. Two commonly used dimensionality reduction techniques are Principal Component Analysis (PCA) (Hotelling, 1933) and Student t-distributed Stochastic Neighbourhood Embedding (t-SNE) (Maaten and Hinton, 2008), since these techniques are able to produce 2-dimensional projections, while preserving the structure of the high-dimensional data in their own way. In particular, PCA is a linear technique which pre-

serves the large dissimilarities between observations in the 2-dimensional projection; it preserves as much of the variance in the data as possible. In contrast, t-SNE is a nonlinear technique which focuses on placing similar observations closer to each other, that is, it preserves the local neighbourhoods in the data. It is shown by Van Der Maaten et al. (2009) and Maaten and Hinton (2008) that both PCA and t-SNE are able to project a variety of real-life data sets better than several existing dimensionality reduction techniques.

2.3 Hierarchical Linear Models

Hierarchical linear models are statistical models that take into account a hierarchical structure in the data. The model has a single dependent variable at the lowest level of aggregation and independent variables at each other level. In literature, this model is known under a variety of names, such as "random coefficient model", "variance component model", or "multilevel regression model" (Hox et al., 2017). In Bayesian literature this model is often referred to as "hierarchical linear model", following the lead of Lindley and Smith (1972), who laid out the framework for Bayesian estimation of a linear model with a hierarchical prior structure.

Hierarchical models are useful when the assumption that observations are independent of each other does not hold because of a shared context (Leyland and Groenewegen, 2003). Raudenbush (1988) acknowledges that problems like aggregation bias and misestimated precision are almost inevitable, when this hierarchical structure is ignored.

Apart from using a multilevel model to model hierarchical data, machine learning algorithms like neural networks, fuzzy sets and neurofuzzy networks could be trained to classify observations with great precision. However, a drawback of these kind of algorithms is the lack of interpretation on the relationships that exist between variables (Washington et al., 2009).

2.4 Bayesian estimation methods

In the current literature, hierarchical linear models are solved either in the classical way with maximum likelihood (ML) based methods or are designed in a Bayesian framework. Moreover, the literature on Bayesian estimation seems to be more popular for hierarchical linear models, due to certain disadvantages of ML estimation in a hierarchical framework. In particular, while ML based methods are faster than Bayesian methods, they tend to underestimate the uncertainty (Dosne et al., 2016) compared to Bayesian methods when the assumption of normality does not hold. Additionally, Stegmueller (2013) finds that Bayesian methods produce significantly less biased multi-level models than ML methods for a set of group-specific parameters. For each number of group-specific parameters that were selected, the ML models produced more biased results. Stegmueller (2013) also states that Bayesian methods have better frequentist

coverage than ML methods. Finally, ML methods are known to occasionally produce extreme estimates in small sample sizes, as described by Rouder et al. (2003). It follows that, compared to ML estimation, Bayesian estimation shrinks extreme estimates, making these estimators more robust. Rouder et al. (2003) also find that ML estimates are more biased compared to Bayesian estimates in a simulation study, coinciding with the results of Stegmueller (2013).

In a logistic regression context, normal priors have similar predictions performance when compared with other popular prior distributions as studied by Ghosh et al. (2018). The prior distributions under inspection are the less informative Cauchy and Students-t priors. Ghosh et al. (2018) explain that the normal priors have several computational advantages in Bayesian sampling. The normal priors are more informative than the aforementioned distributions, since the tails of these distributions are larger than that of a normal distribution.

2.5 Data imputation

A naive way to fix data sets with missing values is by deleting the corresponding observations. However, this has the risk of losing data with valuable information. A better way to solve this problem is by imputing the missing values, that is, inferring the missing values from the observed part of the data. For this reason, several imputation procedures have been proposed to fill in these missing values, such as single and multiple imputation (Rubin, 2004). Single imputation procedures could result in unbiased estimates (Donders et al., 2006). However, this procedure does not account for the uncertainty in the imputations, since once the imputation is completed, the analysis continues as if the imputed values were the true value. Consequently, this might lead to incorrect conclusions (Azur et al., 2011). In contrast, multiple imputation procedures do take into account the uncertainty of the imputations, since this involves creating multiple predictions for each missing value, resulting in accurate standard errors (Azur et al., 2011).

A multiple imputation procedure that has emerged in the statistical literature as a principled method for addressing missing data is Multivariate Imputation by Chained Equations (MICE) (Azur et al., 2011), which imputes the missing values by predicting it with a linear regression model fitted with the available data. These regression models are fitted multiple times for the same missing value, but with different available data as the missing values become available in the first imputation and will change with each new imputation. This chains the linear regression models, which creates very flexible results, since multiple imputation accounts for the statistical uncertainty in the imputations, as opposed to single imputations (Azur et al., 2011).

However, a drawback of MICE is that the linear regression models require some standard assumptions on the data (Heij et al., 2004), which in practice might not be able to be satisfied.

An imputation method that does not require such assumptions is Nearest Neighbour Imputation (NNI) (Chen and Shao, 2000). This method has the advantage that it is expected to be more robust than methods which are based on explicit relations in the data such as MICE (Chen and Shao, 2000). Moreover, Chen and Shao (2000) show that the bias of the NNI sample mean is asymptotically negligble. In particular, NNI imputes observations with missing values based on the mean of the non-missing values of its nearest neighbours.

3 Data

This section presents the data used to illustrate this research. Section 3.1 describes the characteristics of the regional data set, obtained from the Central Bureau of Statistics. Next, Section 3.2 presents the restaurant specific data set provided by Unilever Food Solutions .

3.1 Regional Data

The regional data set, which we denote by $\mathbf{Z}$, consists of open source data provided by the CBS[1]. The data consists of several neighbourhood demographics corresponding to all the 4.068 4-digit zip codes across the Netherlands. For our research, we will use the regional variables related to age, income, race, gender, household size and urbanization, since these variables influence the location of a restaurant, which is an important factor for the success of a restaurant as discussed in Section 2.1.

In particular, we include 23 regional variables related to age demographics, income demographics, general demographics such as average household size, gender or race, and some geographical descriptives such as address density, distance to the nearest train station or number of food facilities in a certain radius. We include these geographical descriptives as they are indicative of the urbanization in a zip code region. A complete description of the used variables and summary statistics can be found in Tables 7 and 8, respectively, in Appendix A.1.

3.1.1 Preprocessing the Regional Data

In the 4.068 zip codes from the regional data set there are 2.784 zip codes without missing values and 1.284 zip codes with missing values. Rather than excluding the zip codes with missing values and consequently losing valuable information, we opt to impute these missing values. To preprocess the regional data, we impute the missing values with Nearest Neighbor Imputation (NNI), since all used variables are non-negative. In particular, NNI is based on the means of the nearest neighbours and it thus does not result in negative imputed values, while,

[1] https://www.cbs.nl/nl-nl/dossier/nederland-regionaal/geografische-data/gegevens-per-postcode

for example MICE might result in negative imputed values as it uses linear regression models to predict the missing values.

However, NNI uses the Euclidean distance between the observations and is thus dependent on the scale of the variables. So, before imputing the missing values with NNI, we have to normalize the variables to the same scale. The results of the Jarque-Bera test (Thadewald and Büning, 2007) on the variables as shown in the results of Table 8 in Appendix A.1 reveal that all variables deviate from normality. For this reason, we normalize each feature vector $\mathbf{z}_j$, in the regional data $\mathbf{Z}$ via a minmax scale as follows:

$$\mathbf{z}_j^{norm} = \frac{\mathbf{z}_j - \min(\mathbf{z}_j)}{\max(\mathbf{z}_j) - \min(\mathbf{z}_j)}, \quad \text{for} \quad j = 1, ..., d, \tag{1}$$

such that all variables have the same scale in the range of [0,1] without making any assumptions about the data. Moreover, this normalisation may improve the numerical accuracy of the SOM network (Kohonen, 1990).

However, the minmax scale is sensitive to outliers. To determine potential outliers in the regional data set, we implement a commonly used outlier detection method called Local Outlier Factors (LOF) (Breunig et al., 2000). In short, LOF compares the local density of an observation with the density of its nearest neighbors to determine whether an observation is an outlier. In particular, we find 126 potential outliers by implementing LOF on the 2.784 zip codes without missing values. After normalising and imputing the regional data sets with and without outliers, we are left with 23 variables in an imputed regional data set with outliers containing 4.068 zip codes and an imputed regional data set without outliers containing 3.943 zip codes. We consider the complete regional data set with outliers, since excluding any zip codes would mean that we are not able to forecast bankruptcy of restaurants in these excluded zip codes. In contrast, we consider the regional data set without outliers, since this might lead to a better performance of the imputation and consequently our implemented methods.

3.2 Restaurant-specific Data

The restaurant-specific data set we use is provided by UFS, which we denote by $\mathbf{X}$. This data set consists of restaurant specific data for 45782 restaurants in the Netherlands. In particular, the data set contains variables concerning the location of the restaurants, such as the *latitude*, *longitude*, *address*, *city* and *postalCode*, but also variables on the type of restaurant: *globalChannel* and *cuisineType*. The *globalChannel* variable categorises the type of the restaurant, e.g., cafe, pizzeria or fast food restaurant, while *cuisineType* categorises the type of food served at restaurants, e.g., burgers, North American or Asian. Furthermore, the data set contains variables regarding the performance of a restaurant: *rating* and *nrRatings*, which represent the

average given Google-review ratings on a scale from 1 to 5 and the number of Google-review ratings, respectively. Lastly, the data set contains the categorical variable *closed* which indicates whether a restaurant is operational, permanently closed or temporarily closed.

The variables from this data set that are used in our research are: *postalCode, globalChannel, rating, nrRatings* and *closed*. We use the variable *postalCode* to link the restaurants to the corresponding zip code they are located in. Next, the variables *globalChannel, rating* and the log-transformation of *nrRatings* are used as independent restaurant specific variables, since these variables contain information regarding the type, performance and popularity of each restaurant. In contrast, *closed* is used as the dependent variable, since this variable is indicative of whether a restaurant is not successful, if the restaurant is temporarily or permanently closed.

Moreover, we do not include *cuisineType*, since this variable is too specific in characterizing the type of restaurant and thus has a large risk of not reflecting the true cuisine type. Many restaurants in the data set are not limited to the cuisine type they obtain, often serving more than one cuisine type. Hence, the variable *cuisineType* might cause a slight misrepresentation of reality. Consider for example a pizzeria: most pizzerias serve pasta and pizza. However, "pasta" and "pizza" are two distinct categories in the *cuisineType* variable.

3.2.1 Preprocessing the restaurant-specific data

For 22720 restaurants in the data set provided by UFS the variable *closed* is unknown. Since this is our dependent variable we do not impute it and thus remove these observations from the data. Next, for 9 restaurants, the variable *postalCode* is unknown, which we remove from the data set, since we are not able to link these restaurants to the corresponding zip code.

After removing these 22729 observations we are left with 395 missing values for *rating* and *log(nrRatings)*. As can be seen in Table 1, 265 of the missing values belong to the group of restaurants that are operational and 130 missing values correspond with the group of restaurants that are permanently closed. We replace the missing values in the operational and permanently closed group with the mean of the ratings and the log-transformation of the number of ratings for the corresponding group. This mean imputation per group might lead to a slight bias. However, since there are not a lot of observations with missing values relative to the total number of observations, the bias induced with mean imputation is limited. After imputing the missing values we have a data set of 23042 observations. In Table 2 we present the number of restaurants in category of *closed*. As we can see the data set is imbalanced: only 0.9% of the restaurants belong to the category temporarily closed, 16.7% of the restaurants is permanently closed and 82.4% is operational.

Furthermore, the eleven categories of the variable *globalChannel* are merged into four categories: "Dining", "No dining", "Fastfood" and "other", where the exact merging is given in Table 9 of Appendix A.2. To use *globalChannel* in our models we use the dummy variables: *globalChannel_Dining*, *globalChannel_no_dining* and *globalChannel_fastfood*, where we consider the category "*other*" as the baseline for our models.

Thus, after preprocessing we are left with a restaurant-specific data set containing 23042 observations with the dependent variable *closed* and the following independent variables: *rating*, *log(nrRatings)*, *globalChannel_Dining*, *globalChannel_no_dining* and *globalChannel_fastfood*

Class	**Missing values**	**Mean rating**	**Mean log(nr_rating)**
Operational	265	4.26 (0.34)	3.40 (1.53)
Temporarily Closed	0	4.18 (0.31)	5.84 (1.34)
Permanently Closed	130	4.08 (0.58)	5.02 (1.20)

Table 1: Missing values per class, mean of the ratings and mean of the number of ratings.

Class	**Number of restaurants**	**Percentage**
Operational	18976	82.4
Temporarily Closed	228	0.9
Permanently Closed	3838	16.7

Table 2: Number of observations per class

4 Methodology

This section elaborates on the methods used in this research. To answer our research questions, we define a two-step methodology: The clustering of the zip code regions, which is presented in Section 4.1, and forecasting bankruptcy using GLH Multinomial Logit models, which is elaborated on in Section 4.2.

4.1 Step 1: Two-stage clustering procedure

In our first step, we consider TS k-means and TS GMM as two-stage clustering procedures to cluster the zip code data. Specifically, we first use SOM to cluster the zip codes into M prototypes, then, in the second stage we use k-means or GMM to cluster these prototypes to the desired number of clusters k.

Moreover, we first discuss SOM, k-means and GMM in Sections 4.1.1, 4.1.2, and 4.1.3 respectively. Then, in Section 4.1.4 we discuss how to determine the optimal number of clusters based on two cluster validation measures. Lastly, in Section 4.1.5 we choose the hyperparameters for the two-stage clustering procedure.

4.1.1 Self Organising Maps

The SOM is obtained by training a two-layer artificial neural network consisting of an input and a Kohonen layer (Kohonen, 1990) as shown in Figure 1, where the weight matrix of the

network is denoted by $\underset{M\times d}{\mathbf{W}} = \begin{bmatrix} \mathbf{w}_1^\intercal \\ \vdots \\ \mathbf{w}_M^\intercal \end{bmatrix}$, where d is the input dimension and the j'th map unit in the Kohonen layer is represented by the weight vector $\mathbf{w}_j = [w_{j1}, ..., w_{jd}]^\intercal$. These weight vectors are the prototypes, and are connected to each other to preserve the neighbourhood relations.

The SOM network is trained iteratively using a competitive unsupervised learning procedure (Du, 2010). The procedure is as follows: at each iteration, first find the b'th map unit which corresponds to the Best Matching Unit (BMU) for an observation $\mathbf{z}_i = [z_{i1}, ..., z_{id}]^\intercal$. In particular, the BMU is the map unit, where its prototype is closest to $\mathbf{z}_i$ according to the Euclidean distance. Thus, it holds that

$$b = \arg\min_j \{||\mathbf{z}_i - \mathbf{w}_j||\}, \tag{2}$$

where $||\cdot||$ denotes the Euclidean distance.

Then, secondly the prototypes are updated by moving the BMU and its topological neighbours closer to the input vector in the input space. This is achieved using the Kohonen learning rule (Kohonen, 1990) given by

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \eta(t)h_{jb}(t)[\mathbf{z}_i - \mathbf{w}_j(t)] \quad \text{for} \quad j = 1, 2, ..., M, \tag{3}$$

where t is the current iteration step, $\eta(t)$ is a monotonically decreasing learning rate, $h_{jb}(t)$ is the neighbourhood kernel centered on the b'th unit which defines the relation between the j'th and b'th map units. A short description of this training procedure is provided in Algorithm 1. In particular, the exponential decay function is commonly used as the learning rate for SOM according to Zhang et al. (2018), which is given by

$$\eta(t) = \eta_0 \exp(\frac{t}{T}), \tag{4}$$

where η_0 is the initial positive constant and T the number of iterations. Next, the neighbourhood kernel is chosen to be decreasing with the increasing distance between the j'th and b'th map units, and the iteration step. Let n_{row} and n_{col} denote the number of rows and columns of the discrete 2-dimensional grid of the Kohonen layer respectively, such that $n_{row} * n_{col} = M$, then $\mathbf{c}_j \in \mathbb{N}^{n_{row}\times n_{col}}$ denotes the coordinate of the j'th map unit in the grid. The neighbourhood relation between the j'th and b'th map units is typically given by a Gaussian kernel defined as

$$h_{jb}(t) = h_0 \exp(-\frac{||\mathbf{c}_b - \mathbf{c}_j||^2}{2\sigma^2(t)}), \tag{5}$$

where h_0 is an initial positive constant and the radius $\sigma(t)$ has to be a decreasing function in t, such that the map will stabilize into its final organization (Zhang et al., 2018). In particular,

the radius is given by

$$\sigma(t) = \sigma_0 e^{-\frac{t}{\tau}}, \tag{6}$$

where σ_0 is an initial positive radius and τ a time constant (Zhang et al., 2018) defined as

$$\tau = \frac{T}{\ln \sigma_0}. \tag{7}$$

Algorithm 1 Training procedure of the SOM network

Input: The regional data set $\mathbf{Z}$, the number of rows n_{row} and columns n_{col}, the number of iterations T, and the constants η_0, h_0 and σ_0.

Output: The M optimized prototypes

1: Initialize the prototypes $\mathbf{w}_j$ with small random values for $j = 1, 2..., M$.
2: Randomly pick a region $\mathbf{z}_i$ from $\mathbf{Z}$.
3: Obtain the b'th map unit of $\mathbf{z}_i$ such that it satisfies (2).
4: Update the prototypes corresponding to the map units in the neighbourhood of the b'th map unit by pulling them closer to the observation $\mathbf{z}_i$ using the Kohonen learning rule given in Equation (3).
5: Increment t and repeat from step 2 while $t < T$.

4.1.2 k-means

k-means is a non-probabilistic clustering method which, in our case, aims to partition the M prototypes, obtained from the first stage, into k clusters $C_1, ..., C_k$, where each prototype belongs to the cluster with the nearest mean μ_c for $c = 1, ..., k$. This is done by finding the centers μ_c that minimise the following objective function:

$$E_{k-\text{means}} = \sum_{j=1}^{M} \min_{c} ||\mathbf{w}_j - \mu_c||^2. \tag{8}$$

The objective function can be minimised with the Lloyd's algorithm (Lu and Zhou, 2016), which can be seen as an Expectation-Maximization (EM) algorithm. The procedure of the Lloyd's algorithm is shown in Algorithm 2, where the algorithm is said to converge when either there is no change in cluster assignment or a maximum number of iterations T is reached.

Algorithm 2 The Lloyd's algorithm which optimizes the k-means objective function.

Input: The prototypes $\mathbf{w}_1, ..., \mathbf{w}_M$, the number of iterations T.

Output: The clusters $C_1, ..., C_k$ and their corresponding centers $\mu_1, ..., \mu_k$.

1: Randomly initialize the centers μ_c for $c = 1, 2..., k$.
2: Repeat until convergence:
a: **E-step**: For each prototype $\mathbf{w}_j$, compute the distance to each cluster center μ_c for $c = 1, ..., k$, and assign each prototype to the cluster with the nearest center.
b: **M-step**: Using the new cluster assignments of the prototypes, compute the new means for each cluster and update the cluster centers μ_c for $c = 1, ..., k$.

4.1.3 Gaussian Mixture Models

As opposed to k-means, GMM is a probabilistic clustering method, which in our case aims at partitioning the M prototypes into k clusters $C_1, ..., C_k$, where GMM assumes that each cluster is independently represented by a Gaussian distributions with a mean μ_c, covariance Σ_c and a cluster probability $p(C_c) = \pi_c$ representing the unconditional probability of a prototype falling in cluster C_c, such that $\sum_{c=1}^{k} \pi_c = 1$. The GMM can thus be represented as a linear combination of Gaussian probability distribution expressed as

$$p(\mathbf{W}) = \sum_{c=1}^{k} \pi_c \phi(\mathbf{W}; \mu_c, \Sigma_c), \tag{9}$$

where $\phi(\cdot; \mu_c, \Sigma_c)$ is the normal p.d.f with mean μ_c and covariance Σ_c. To estimate the means and covariances, we can make use of Maximum Likelihood (ML) estimation by maximizing the loglikelihood of Equation (9). However, there is no analytical solution for this log likelihood (Gupta and Chen, 2011). Thus, GMM makes use of an EM-algorithm to find an approximate solution as described by Gupta and Chen (2011). The procedure of this EM-algorithm is given in Algorithm 3, where the algorithm is said to converge when there is either no change in the Maximum Loglikelihood value or a maximum number of iterations T is reached.

Algorithm 3 EM-algorithm for GMM.

Input: The prototypes $\mathbf{w}_1, ..., \mathbf{w}_M$ and the number of iterations T.

Output: The clusters $C_1, ..., C_k$ and their Gaussian distributions $N(\mu_1, \Sigma_1), ..., N(\mu_k, \Sigma_k)$.

1: Randomly initialize the μ_c, Σ_c and π_c for $c = 1, 2..., k$.

2: Repeat until convergence:

a: **E-step**: Compute the probability of the prototype $\mathbf{w}_j$ belonging to any cluster C_c as

$$p(C_c|\mathbf{w}_j) = \frac{\phi(\mathbf{w}_j|\mu_c, \Sigma_c) * \pi_c}{\sum_{s=1}^{k} \phi(\mathbf{w}_j|\mu_s, \Sigma_s)\pi_s} \tag{10}$$

b: **M-step**: Using the probabilities from the E-step update μ_c, Σ_c and π_c with the following equations for $c = 1, 2..., k$:

$$\mu_c = \frac{\sum_{j=1}^{M} p(C_c|\mathbf{w}_j) * \mathbf{w}_j}{\sum_{j=1}^{M} p(C_c|\mathbf{w}_j)} \tag{11}$$

$$\Sigma_c = \frac{\sum_{j=1}^{M} p(C_c|\mathbf{w}_j) * (\mathbf{w}_j - \mu_c)(\mathbf{w}_j - \mu_c)^\intercal}{\sum_{j=1}^{M} p(C_c|\mathbf{w}_j)} \tag{12}$$

$$\pi_c = \frac{\sum_{j=1}^{M} p(C_c|\mathbf{w}_j)}{M} \tag{13}$$

4.1.4 Cluster validation and visualisation

To determine the optimal number of clusters k^*, we use two different methods to assess the quality of the clusters resulting from our two-stage clustering procedure.

- **The Silhouette Coefficient** (Rousseeuw, 1987): this metric examines how accurately individual data points are assigned to their respective clusters. The Silhouette Coefficient for the i'th observation is computed as

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{14}$$

  where $a(i)$ is the average distance of the i'the observation to the other observations within the same cluster, and $b(i)$ represents the smallest average distance of the i'th observations to all observations from other clusters, The value of the Silhouette Coefficient ranges from -1 to 1. If $S(i)$ is 1, an observation is clustered correctly. If $S(i)$ is close to -1, this observation should have been assigned to a different cluster. Lastly, if S(i) is close to 0 the observation resides between two clusters. Thus, an optimal clustering method would achieve a Silhouette Coefficient of 1. Furthermore, to assess the quality of the clusters, we compute the average Silhouette Coefficient over all observations.

- **Davies-Bouldin (DB) index** (Davies and Bouldin, 1979) The Davies-Bouldin Index combines the intra-cluster similarity and inter-cluster differences between data points into one measure calculated as

$$DB = \frac{1}{k} \sum_{c=1}^{k} \max_{s \leq k, s \neq c} \left(\frac{\gamma_c + \gamma_j}{d(\mu_c, \mu_s)}\right), \tag{15}$$

  where γ_c denotes the average distance of all observations in cluster c to the corresponding cluster center, and $d(\mu_c, \mu_s)$ is the Euclidean distance between the centers of cluster c and s. Thus, the optimal clustering method achieves the lowest DB Index, as this indicates a small intra-cluster distance, while having a large inter-cluster distance.

We compute the average Silhouette Coefficient and the Davies-Bouldin Index for each model, where we vary the number of clusters k in a range $\{1, ..., k_{max}\}$. The optimal number of clusters is the value of k that corresponds with the highest average Silhouette Coefficient and the lowest Davies-Bouldin Index. Here, we focus on clustering the zip codes, in order to obtain zip code regions that are clearly distinguishable based on their regional characteristics. These different types of regions can then be used in the GLH Multinomial Logit Models of the second step of our two step approach, to investigate the difference of bankruptcy risk between these zip code regions.

Furthermore, to provide a visual representation to validate the performance of our clustering algorithms, we implement two dimensionality reduction techniques, t-SNE (Maaten and Hinton, 2008) and PCA (Hotelling, 1933), to obtain two-dimensional projections of the data. The projections of these techniques are able to preserve the local neighbourhood and variance of the data respectively. Thus, we can use the projections from t-SNE to visually determine whether the obtained clusters are able to preserve the similarities between observations, while the projections from PCA be used to visually determine whether the obtained clusters are able to preserve dissimilarity between observations.

4.1.5 Hyperparameters two-stage clustering procedure

The two-stage clustering procedure is implemented on the regional data set provided by the UFS. For the SOM network we set the maximum number of iterations T to 10000. Furthermore, we have to determine the map shape by setting n_{row}, n_{col} and M. Canetta et al. (2005) propose to set the number of map units M to be proportional to $\sqrt{N}$, where $N = 4068$ is the number of observations in the complete regional data set. Thus, we set $M = \lceil\sqrt{4068}\rceil = 64$. Consequently, we set n_{row} and n_{col} both to 8, such that $n_{row} * n_{col} = M$. Next, we have to set the initial constants η_0, h_0 and σ_0. We set $\eta_0 = 0.1$, since Zhang et al. (2018) find that a higher learning rate would cause the prototypes to swing more towards the most recent matched input observation than it should. This means that the prototypes forget the impact of the other input observations, that the prototypes were matched to before, too quickly. Furthermore, similar to Zhang et al. (2018), we set $h_0 = 1$ and σ_0 equal to half of the size of the smallest dimension in the SOM map which can be computed as

$$\sigma_0 = \frac{\min(n_{row}, n_{col})}{2}. \tag{16}$$

Next, we set the maximum number of iterations T to 5000 for both the k-means and GMM algorithms. However, these algorithms are both an EM-algorithm and are thus susceptible to be stuck in a local minimum depending on the random initialisation. For this reason, we consider 10 different random initialisations for these algorithms and use the initialisation with the best results.

4.2 Step 2: GLH Multinomial Logit models

In this section we describe the models that we use and the priors we specify for them. First, we provide general information about these models in Section 4.2.1. Moreover, to answer our research question and second subquestion, we use a build-up approach which allows us to investigate the effects of making small adjustments to the models. We propose a Pooled model

with common parameter estimates between clusters, an Unpooled model which allows for the parameter estimates to vary between clusters and finally a Simultaneous Unpooled (SU) model, where the clusters are treated as dependent on each other. These models are further described in Sections 4.2.2, 4.2.3 and 4.2.4. Next, Section 4.2.5 provides performance measures for the GLH models. Lastly, Section 4.2.6 provides the sampling and modelling procedure of the GLH models.

4.2.1 General notation

The dependent variable for our models is the nominal variable *closed*, this variable has three possible outcomes: Permanently Closed (PC), Temporarily Closed (TC) and Operational (O). To model the outcome of this variable we use a GLH model with a multinomial logit link function defined as

$$y_i|\theta_i \sim MN(\theta_i), \tag{17}$$

where the vector θ_i equals $\left[\theta_i^{[O]}, \theta_i^{[TC]}, \theta_i^{[PC]}\right]$. Here the individual elements $\theta_i^{[R]}$ represent the probability that the restaurant will end up in state R, and are obtained by the inverse multinomial logit link function $g(\cdot)$ which is given by

$$g(\theta_i^{[R]}) = ln\left(\frac{\theta_i^{[R]}}{\theta_i^{[r]}}\right) = \zeta_i^{[R]}, \tag{18}$$

where $R \in \{O, TC, PC\}$ is one of the outcomes of the dependent variable and r is the index of the baseline probability. This implies that the regression coefficients may differ for different outcomes. For our models we include restaurant-specific and regional data.

In particular, each observation from the restaurant-specific data $\mathbf{X}$ is denoted by $\mathbf{x}_i$. Moreover, to study if the clustering is useful for the hierarchical models we also include variables from the regional data set $\mathbf{Z}$ which are used for the clustering. We denote the combined restaurant-specific and regional data as $\mathbf{V}$ with $\mathbf{v}_i = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{z}_i \end{bmatrix}$, where $\mathbf{z}_i$ is the regional data of the zip code for the i'th restaurant. In particular, we compare the performance of the models using the variables in $\mathbf{x}_i$ with the variables in $\mathbf{v}_i$. The reasoning behind this comparison is that if the clustering is effective for the hierarchical models, the regional variables should not add a lot of information to the model, since they are indirectly already included via the clustering. We use different performance measures, described in Section 4.2.5, to compare the models and investigate this.

4.2.2 Pooled model

To start our build-up procedure we specify a Bayesian multinomial logit model as described by (Madigan et al., 2005). The model contains explanatory variables $\mathbf{r}_i$ and choice-specific

parameters $\beta^{[R]}$, where $\mathbf{r}_i$ is equal to $\mathbf{x}_i$ or $\mathbf{v}_i$. This specification does not take into account cluster specific effects. Instead the parameters $\beta^{[R]}$ determine the overall average effect per choice. If there are specific clusters in our data, we would expect this specification to be weak.

In particular, the pooled model is specified as

$$\zeta_i^{[R]} = [1 \quad \mathbf{r}_i^\intercal]\beta^{[R]}, \tag{19}$$

where $\zeta_i^{[R]}$ is the linear combination of the data, $\beta^{[R]} = [\beta_0^{[R]}, ..., \beta_m^{[R]}]^\intercal$ are the corresponding parameters and m are the number of independent variables. Next, we place the following informative prior on each of the parameters $\beta_j^{[R]}$:

$$\beta_j^{[R]} \sim N(\mu_j^{[R]}, \sigma_j) \quad \text{for} \quad j = 1, 2, ..., m, \tag{20}$$

where $\mu_j^{[R]}$ the corresponding mean parameter. Next, we specify the prior on each σ_j as $\sigma_j \sim Gamma(2, \frac{1}{10})$, which restricts σ_j to be non-negative. This prior specification is recommended by Chung et al. (2013) for hierarchical modelling and does not allow for correlation between variables.

Furthermore, since we are inspecting an informative prior, it is of importance to select good parameters. We propose a method, similar to Laud and Ibrahim (1996), which uses information of the independent variables to set the parameters. We use a subsample of the data, which we estimate using a standard multinomial logit model. From this model we obtain estimates of the parameters $\beta_j^{[R]}$, which we use to initialize our mean parameters $\mu_j^{[R]}$.

4.2.3 Unpooled model

The first extension we add to our pooled models is allowing for cluster-varying β parameters. In particular, the Unpooled model is specified as

$$\zeta_{ic}^{[R]} = [1 \quad \mathbf{r}_{ic}^\intercal]\beta_c^{[R]} \quad \text{for} \quad c = 1, 2, ..., k^*, \tag{21}$$

where $\beta_c^{[R]} = [\beta_{c0}^{[R]}, ..., \beta_{cm}^{[R]}]^\intercal$, $\mathbf{r}_{ic}$ is equal to the predictor variables that belong to the i'th restaurant that is contained in cluster c, and β_c are the corresponding coefficient estimates for cluster c. The model can be interpreted as k^* different pooled models, where k^* is equal to the optimal number of clusters. For a cluster c we only use the observations that are contained in that particular cluster c. Next, using similar reasoning as in the Pooled model, we specify the prior on each $\beta_{jc}^{[R]}$ as

$$\beta_{jc}^{[R]} \sim N(\mu_{jc}^{[R]}, \sigma_{jc}) \quad \text{for} \quad j = 1, ..., m, \text{ and } c = 1, 2, ..., k^*, \tag{22}$$

where the prior on each σ_{jc} is specified as $\sigma_{jc} \sim Gamma(2, \frac{1}{10})$. When initialising the parameters we group the observations corresponding to a cluster and use standard multinomial logit model to estimate the parameters $\beta_{jc}^{[R]}$, which we use to initialize our mean parameters $\mu_{jc}^{[R]}$.

4.2.4 Simultaneous Unpooled model

The extension that we add to the unpooled model is to simultaneously estimate the parameters between clusters. In particular, in the Unpooled model specification, we independently estimate the model for each cluster. In contrast, in the SU model specification we estimate all models simultaneously and allow for correlation between the variables within a model and between the cluster variables. In this model specification we study the relation between the clusters and their dependence on each other. Instead of assuming that they act independently of each other, we examine their relation through the covariance structure. To model this specification we make use of vectorization to ensure a single multivariate setting, that is, we define

$$\mu_j^{[R]} = \begin{bmatrix} \mu_{j,1}^{[R]} \\ \vdots \\ \mu_{j,k^*}^{[R]} \end{bmatrix} \quad \text{and} \quad \Sigma_{jl} = \begin{bmatrix} \sigma_{jl,11} & \cdots & \sigma_{jl,1k^*} \\ \vdots & \ddots & \vdots \\ \sigma_{jl,k^*1} & \cdots & \sigma_{jl,k^*k^*}, \end{bmatrix} \quad \text{for } j = 1, ..., m, \text{and } l = 1, ..., m. \tag{23}$$

Then, the model specification can be written down as follows:

$$\beta^{[R]} \sim N(\mu^{[R]} = \begin{bmatrix} \mu_1^{[R]} \\ \vdots \\ \mu_m^{[R]} \end{bmatrix}, \Sigma = \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1m} \\ \vdots & \ddots & \vdots \\ \Sigma_{m1} & \cdots & \Sigma_{mm}, \end{bmatrix}) \tag{24}$$

where we also introduce a prior on $\mu_{j,c}^{[R]}$ which is defined as

$$\mu_{j,c}^{[R]} \sim N(\mu_j^{[R]}, \sigma_j) \quad \text{for } j = 1, ..., m \text{ and } c = 1, ..., k^*. \tag{25}$$

We specify a normal prior with a mean $\mu_j^{[R]}$ that is similar between clusters. Similarly as for the Pooled model, we use a subsample of the data, apply a standard multinomial logit model to this subsample and obtain the estimates of the parameters $\beta^{[R]}$, which we use to initialize our means. We use a similar mean between clusters to further inspect the dependence between clusters. This dependence is captured by the covariance terms Σ_{jj} for $j = 1, ..., m$.

For Σ we propose to use an LKJ prior (Barnard et al., 2000). This prior has proven good for controlling the expected amount of correlation among the parameters. In particular, to use the LKJ prior we first define

$$\Sigma = \begin{bmatrix} \tau_1 & & \\ & \ddots & \\ & & \tau_m \end{bmatrix} \Omega \begin{bmatrix} \tau_1 & & \\ & \ddots & \\ & & \tau_m \end{bmatrix} \tag{26}$$

where Ω is a correlation matrix and $\tau = [\tau_1, ..., \tau_m]$ is the vector containing the coefficient scales. For the prior on τ we follow Carpenter et al. (2017), who recommend

$$\tau_j \sim Cauchy(0, 2.5) \text{ for } j = 1, ..., m, \tag{27}$$

where τ_j is constrained such that $\tau_j \geq 0$. The prior on the correlation matrix Ω is an LKJ prior

$$\Omega \sim LKJCorr(\iota), \tag{28}$$

where the LKJ correlation distribution is given by

$$LKJCorr(\Sigma|\iota) \propto \det(\Sigma)^{\iota-1}. \tag{29}$$

It is clear that for $\iota = 1$ this prior is equal to the uniform distribution, for $\iota > 1$ the prior favors less correlation, for $\iota < 1$ more correlation is favored (Carpenter et al., 2017).

4.2.5 Performance Measures

To compare the specified models with each other we make use of Bayes factors and prediction performance. Bayes factors, as described by Kass and Raftery (1995), compares the marginal likelihood of models with each other to determine which model is a better fit for the data. Advantages of Bayes factors include that they are easy to compute, do not favor more complex models and prevent overfitting. The authors also explain that Bayes Factors are useful and commonly used for guiding an evolutionary model-building process. The Bayes Factors are generally computed as

$$BF_{A|B} = \frac{p_A(y)}{p_B(y)}, \tag{30}$$

where $p_v(y)$ is the marginal Likelihood of model v. To draw conclusions from the Bayes Factors, we use the Jeffreys scale as described by Ly et al. (2016). A Bayes Factor that exceeds 1 implies that model A is preferred over model B. According to the Jeffreys scale a Bayes factor is barely worth mentioning when smaller than 3.16, there is substantial evidence when the value lies between 3.16 and 10, there is strong evidence when the value lies between 10 and 30 and above 30 the indication that model A is better becomes very strong and decisive.

Since it is not analytically tractable to compute the marginal likelihood, we approximate this via bridge sampling as described by Gronau et al. (2017). The authors find that bridge sampling gives accurate estimates of the marginal likelihood for hierarchical versions of a model. Moreover, bridge sampling has a low computational load.

To study the prediction performance, we measure the out-of-sample prediction accuracy. Since there is class imbalance prediction accuracy might give misleading results, therefore we

compute the precision and recall, the recall for class R is

$$\text{recall}^{[R]} = \frac{TP^{[R]}}{FN^{[R]} + TP^{[R]}}, \tag{31}$$

and precision for class R is

$$\text{precision}^{[R]} = \frac{TP^{[R]}}{TP^{[R]} + FP^{[R]}}, \tag{32}$$

where $TP^{[R]}, FP^{[R]}$ and $FN^{[R]}$ are the true positives, the false positives and the false negatives for class R, respectively. Precision is the fraction of relevant instances among the predicted instances, while recall is the fraction of relevant instances that were classified.

4.2.6 Sampling and modelling of the GLH models

We split the used data sets $\mathbf{X}$ and $\mathbf{V}$ in a training set that consists of 70% of the observations and a test set which consists of the remaining 30%. Due to time and resource constraints we use a subset of the training data, which consists of 3000 observations while using the full test data set consisting of 7604 observations. Next, to ensure identification of our GLH Multinomial logit models we restrict all estimates of the Operational group to zero. Thus, this group is the baseline probability. For choosing the parameters of the LKJ prior we propose to use $\iota = 0.5$, such that we induce more correlation between parameters.

Furthermore, we obtain the Bayesian inference through a Markov chain Monte Carlo method called No-U-turn sampling. In particular, No-U-Turn sampling is an adapted Hamiltonian Monte Carlo sampling method, which builds forth on Hamiltonian Monte Carlo methods as described by Nishio and Arakawa (2019). Hamiltonian Monte Carlo methods are known to avoid random-walk like behaviour when making posterior draws and achieve a more effective and consistent exploration of the probability space, when compared to other Markov chain Monte Carlo methods. However, a major shortcoming of previous Hamiltonian Monte Carlo methods is its sensitivity to certain hyperparameters. The No-U-turn sampler automates the tuning of these hyperparameters. In a study by Nishio and Arakawa (2019), when comparing Gibbs, Hamiltonian and No-U-turn sampling, they find optimal estimation performance for the Gibbs and No-U-turn sampler. When comparing the computational load between these two sampling methods, they find that the Gibbs sampler is slower for their examples.

In particular, for our sampling method, we use four chains and 2000 iterations when running our Bayesian models as advised by the Stan developers Carpenter et al. (2017). Of the 2000 iterations, the first 1000 are for warm-up. Each chain starts with a different initial value that converges towards a local optimum. When chains have not converged either the global optimum is not found or not enough iterations are run to reach convergence between the chains. To study

convergence between chains, the Stan developer team has created $\hat{R}$ statistic (Carpenter et al., 2017). The $\hat{R}$ statistic, as described by the authors, measures the ratio of the average variance of draws within each chain to the variance of the pooled draws across chains. If all chains have converged towards an equilibrium this variance for all chains will be the same, resulting in a statistic of 1. Each parameter estimate has a unique $\hat{R}$ value. It is advised by the Stan authors that the $\hat{R}$ values must not exceed 1.05, for a sample of draws to be trustworthy.

5 Results

In this section we provide the results of our two-step approach to forecast restaurant potential. In particular, the two-stage clustering procedure is implemented using Python 3.7, where we implemented the SOM networks ourselves, while the k-means and GMM algorithms are implemented using the scikit-learn library (Pedregosa et al., 2011). In contrast, to model our Bayesian hierarchical linear models we use a probabilistic programming language called STAN (Carpenter et al., 2017). The code can be found at: `https://github.com/StefanLam99/UnileverCase`, and a brief description of the Python and STAN files are given in Appendix D.

First, in Sections 5.1 and 5.2 we present the cluster performances and interpretation of our clustering methods with the complete regional data set respectively. The cluster performance and interpretation using the regional data set without the potential outliers determined by Local Outlier Factors (LOF) is presented in Appendix B. In particular, we opt to use the complete regional data set for our GLH models as we do not find a substantial difference between the performances and interpretations of our clustering methods using the complete regional data set and the regional data set without outliers. Moreover, in this research we would like to be able to forecast the bankruptcy of any given restaurant, which becomes impossible for restaurants that are in the excluded zip codes of the regional data set.

Next, in Section 5.3 we elaborate on the divergence of the GLH models using the complete regional data set. Then, in Section 5.4 the prediction performance of the GLH models is discussed. Lastly, in Section 5.5 we try to resolve the poor prediction performance by resampling the restaurant data set.

5.1 Cluster performance

First, we decide on the optimal number of clusters by plotting the average Silhouette Coefficient and DB Index against the number of clusters for the implemented clustering methods as shown in Figure 3, where the star marker corresponds to the number of clusters with the best performance for each method. This figure shows that overall both GMM and TS GMM perform worse than k-means and TS k-means. This implies that the assumption of GMM that the data follows a

Gaussian Mixture distribution might be violated, since k-means is non-parametric, but obtains a better performance than GMM. Moreover, TS GMM performs worse than GMM, which can be explained by the fact that GMM requires relatively more samples than k-means, as GMM has to estimate more parameters in the EM-algorithm. Thus, reducing the sample size by obtaining prototypes in the first stage might cause GMM to perform worse than using the original sample.

Furthermore, in Figure 3 we observe that for all implemented clustering methods the best values are obtained for $k = 2$, as shown by the star markers. It is thus likely that there are only two clusters in the data set. In particular, TS k-means obtains the best overall performance for $k = 2$. This is because, if there only exist two clusters in the data set, then the prototypes from SOM have less noise for these two clusters than the original data. Thus, clustering the prototypes rather than the original data set with k-means results in a better clustering quality, since k-means is sensitive to noisy data.

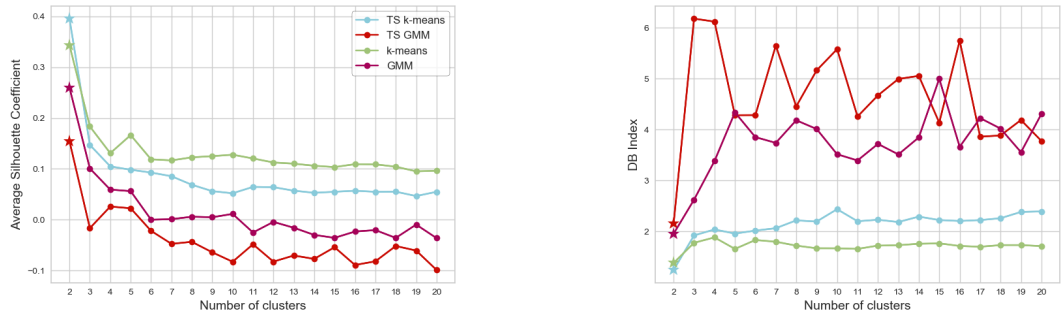

Figure 3: Plots of the average Silhouette Score and DB index against the number of clusters for the four implemented clustering methods using the complete regional data set

To validate the performances of the implemented clustering methods with their optimal number of clusters, we visualise the clusters in two-dimensional projections with t-SNE and PCA as shown in Figure 4. This figure shows two important observations. First, we observe that the clusters from TS GMM and regular GMM are not clearly separated in the visualisations from t-SNE and PCA as some points of the clusters are mixed together, which implies that these clusters are not able to preserve the similarities between observations and the variance between the clusters. Secondly, as opposed to TS GMM and GMM, TS k-means and k-means are able to preserve the similarities between observations and variance between clusters, as we observe that the clusters obtained from these methods are clearly separated in the visualisations from t-SNE and PCA. Thus, these results further illustrate that both TS GMM and regular GMM are inferior to TS k-means and regular k-means for our complete regional data set. Moreover, we conclude that TS k-means with $k = 2$ is the best clustering method, since it obtains the highest average Silhouette Coefficient and the lowest DB index.

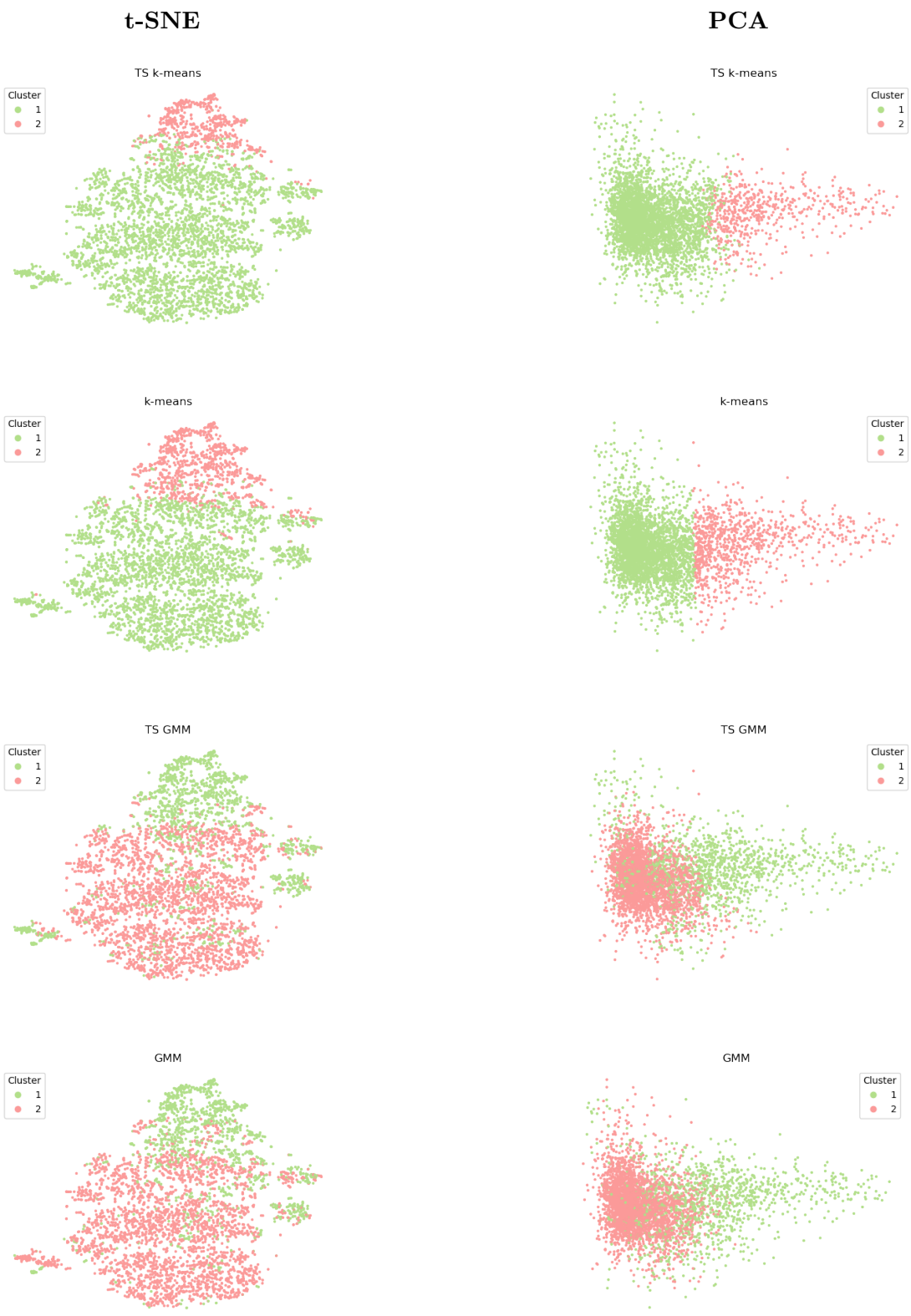

Figure 4: 2D visualisations of the complete regional data set by t-SNE (left column) and PCA (right column) with the optimal number of clusters from the 4 implemented clustering methods.

5.2 Cluster Interpretation

Table 3 presents the statistics of the clusters using TS k-means for $k = 2$. The exact descriptions of the displayed variables can be found in Table 7 in the Appendix.

	cluster 1			cluster 2			Overall		
General Demographics	mean	median	std	mean	median	std	mean	median	std
AMOUNT_HH	1520.196	820.000	1631.775	4674.014	4437.500	2316.512	1952.799	1145.000	2052.008
P_MEN	50.708	50.304	2.724	49.877	49.531	3.424	50.594	50.200	2.844
P_WOMEN	49.292	49.696	2.724	50.123	50.469	3.424	49.406	49.800	2.844
AV_HH_S	2.350	2.300	0.276	1.899	1.900	0.293	2.288	2.300	0.319
P_NL_BACK	87.944	90.000	8.468	57.724	60.000	15.658	83.799	90.000	14.268
P_WE_MIG_B	7.551	10.000	6.112	14.270	10.000	7.073	8.473	10.000	6.666
P_NW_MIG_B	3.654	0.000	5.764	27.466	20.000	15.370	6.920	0.000	11.322
Age Demographics	mean	median	std	mean	median	std	mean	median	std
P_INH_014	15.556	15.318	3.716	14.700	15.117	4.229	15.438	15.294	3.802
P_INH_1524	11.839	11.585	2.499	14.457	12.572	6.264	12.198	11.696	3.403
P_INH_2544	20.560	20.331	4.439	31.061	29.515	6.785	22.000	20.991	6.030
P_INH_4564	31.416	31.233	4.564	24.249	24.468	4.211	30.433	30.493	5.146
P_INH_65PL	20.630	20.455	5.816	15.534	14.714	5.832	19.931	19.968	6.077
Income Demographics	mean	median	std	mean	median	std	mean	median	std
log_median_inc	10.535	10.553	0.192	10.458	10.490	0.273	10.525	10.548	0.207
P_LINC_HH	33.082	32.000	9.518	48.645	49.400	11.292	35.217	33.100	11.150
P_HINC_HH	23.962	23.500	8.603	15.299	13.900	7.814	22.773	22.600	9.007
P_SOCIAL_BEN	6.722	6.042	3.247	11.256	11.145	4.196	7.344	6.421	3.734
Geographical Descriptives	mean	median	std	mean	median	std	mean	median	std
DIS_RAMP	2.012	1.500	2.705	1.960	1.800	0.934	2.005	1.600	2.536
DIS_TRNTR	15.760	13.700	10.435	4.976	3.700	4.243	14.281	12.100	10.497
DIS_TRAINS	7.785	5.900	7.414	2.584	2.000	1.957	7.071	5.100	7.152
AV1_FOOD	6.203	2.200	15.899	39.394	13.850	68.231	10.755	2.800	31.418
AV3_FOOD	33.814	11.400	64.399	299.025	142.300	408.760	70.192	16.050	186.606
AV5_FOOD	74.282	32.200	122.356	631.114	352.100	742.916	150.661	43.250	354.007
EAD	660.489	339.500	724.402	3417.138	2870.500	1974.754	1038.614	507.500	1373.705
Observations	3510			558			4068		

Table 3: Statistics of the overall zip codes and the clusters obtained by TS k-means on the complete regional data set.

In summary, from Table 3, we are able to distinguish between two types of zip code regions based on their characteristics. In particular, the first zip code region corresponding to cluster 1 consists of zip codes with a majority of Dutch natives, relatively less households, a relatively more ageing population. In contrast, the zip code region corresponding to cluster 2 contains zip codes with on average more citizens with an immigration background compared to cluster 1, more citizens who receive social benefits, more food facilities in the vicinity, more densely populated and easier access to various infrastructure.

In addition, in Figure 5 we plot the restaurants corresponding to the two clusters geographically on the Netherlands. Each restaurant is labeled with the cluster to which the zip code of the restaurant's location belongs to. In this figure we observe that most restaurants from cluster 2 are located in the "Randstad", which consists of four of the biggest urban agglomerations of

the Netherlands: Amsterdam, The Hague, Utrecht and Rotterdam. This is also in line with the interpretation of our clusters, as relative to other cities in the Netherlands, the cities in the "Randstad" have larger populations, more immigrants and better infrastructure.

Next, in Table 4 we present for each cluster the percentages of classes for the restaurant status and type. We notice in this table that cluster 2 has around 1% point more fastfood and dining restaurants than cluster 1, while having relatively similar percentages of no dining restaurants. Moreover, cluster 2 contains 6% point less operational restaurants compared to cluster 1. In summary, this table indicates that cluster 2 has more fastfood and dining restaurants, but less operational restaurants than cluster 1.

Figure 5: Map of the Netherlands, where the plotted points correspond to all the restaurants in the complete regional data set with the clusters obtained by TS k-means.

	Restaurant type				Restaurant status			
	Fastfood	No dining	Dining	Other	O	PC	TC	Obs.
cluster 1	2.85	23.06	1.16	72.93	85.02	14.33	0.65	12719
cluster 2	3.83	23.21	2.36	70.60	79.07	19.53	1.40	10323
Overall	3.29	23.13	1.70	71.89	82.35	16.66	0.99	23042

Table 4: Characteristics of the restaurants in our resulting clusters from TS k-means on the complete regional data set.

5.3 Divergent transitions and convergence of the GLH models

When we run our GLH models, our model specifications contain divergent transitions. Divergent transitions arise when the simulated trajectory of the Hamiltonian sampler differs from the true trajectory that was measured from the initial value. When there are too many divergent transitions, the simulations of the parameter estimates can not be trusted. To remove the divergent transitions, Carpenter et al. (2017) advise to first check if lowering the initial step size or increasing the target acceptance rate have any effect. Otherwise, a reparameterisation, digging deeper into the data or specifying different priors might help. After some trial and error, we propose to use weaker prior specifications, since using a too informative prior can cause divergence when inferences are hindered by weakly identifiable likelihoods. For this reason, we propose to modify all GLH models by using weaker priors. In particular, the structure of the GLH models specifications remains the same, but the mean parameters μ are initialized at 0,

for each variable, in each model.

When applying this modification to our GLH models, we find that apart from the SU model containing restaurant and regional variables, all our models converge according to the $\hat{R}$ statistic. Multiple of these coefficients obtain a statistic larger than 1.05, indicating that the model has not converged. For the other models almost all the $\hat{R}$ statistics are exclusively 1.00. The characteristics of the $\hat{R}$ statistics for all the models can be found in Table 13 of Appendix C.

5.4 Prediction Performance of the GLH models

Table 5 presents the contingency tables, prediction rate, precision and recall of the classes of the GLH models for the out-of-sample predictions. Each row represents the predicted values of the observations, while the columns represent the true values. To present the Bayes factors we use the Pooled model with restaurant variables as the baseline model to compare our other models to. The prediction rate is almost equal across models, similarly as the precision and recall of classes between models. It seems that the models have a decent prediction rate. However, on closer inspection we observe that the models almost exclusively predict observations to be operational. This is reflected in the extremely low precision and recall values for the PC and TC classes.

	Restaurant variables									**Regional + Restaurant variables**								
	Pooled Model			**Unpooled Model**			**SU Model**			**Pooled Model**			**Unpooled Model**			**SU Model***		
Prediction\True Value	O	PC	TC	O	PC	TC	O	PC	TC	O	PC	TC	O	PC	TC	O	PC	TC
O	6019	972	74	6014	940	73	6012	933	73	6015	924	73	5998	914	71	6044	972	73
PC	224	315	0	229	347	1	231	354	1	228	363	1	245	373	3	199	315	1
TC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Performance																		
Prediction Rate	83.29%			83.65%			83.72%			83.87%			83.78%			83.63%		
Precision	0.85	0.58	0	0.86	0.60	0	0.86	0.60	0	0.86	0.61	0	0.86	0.60	0	0.85	0.61	0
Recall	0.96	0.24	0	0.96	0.27	0	0.96	0.28	0	0.96	0.28	0	0.96	0.29	0	0.97	0.24	0
Bayes Factors	1			1.96			1.02			0.99			1.99			0.922		

Note: ∗ This model has not converged

Table 5: Contingency tables,prediction performance and Bayes factors for the Generalized Linear Hierarchical Models using out-of-sample predictions

The corresponding prediction rates are thus the result of a large part of the data being operational. Our models are therefore not trustworthy and we do not analyse the results. Moreover, adding regional variables to the restaurant-specific data set has almost no influence to the results of our different model specifications. The Bayes factors show that we are not able to find strong evidence for a best model specification, since all Bayes factors are lower than 3.16. Since, these models are not trustworthy, we try to resolve these problems at hand as discussed in Section 5.5.

5.5 Resampling of the restaurant data

In this section we discuss the approaches which we use to try and fix the nearly exclusively operational predictions. First of all, our data set is heavily imbalanced, since the data consists of over 80% operational observations. As described by, Sáez et al. (2016), an imbalanced data set can cause a large bias to majority classes. They find that oversampling the class with the least observations may lead to a significant improvement. We thus apply a similar approach and run our models with the same number of observations for each class, by bootstrapping observations that are temporarily closed until we obtain a thousand observations. This number of observations is required, since we wish to use a training set that contains 3000 observations with an equal number of observations per group. Next, we randomly select a thousand of observations that are permanently closed and operational which we use as our training set. We call this approach the Bootstrap Oversampling technique.

In addition, we implement SMOTE, another sampling method (Chawla et al., 2002). This method is a combination of oversampling of the smallest class and undersampling of the largest. Chawla et al. (2002) find that SMOTE achieves better classification performance than only undersampling the largest group. We present and discuss the results of SMOTE, since SMOTE provides us with a better prediction performance than the Bootstrap Oversampling technique. The results of SMOTE are presented in Table 6, where the results of the Bootstrap Oversampling technique can be found in Table 12 in Appendix C.

In particular, Table 6 presents the out-of-sample prediction results and Bayes factors of the corresponding models. When we compare these results with the results of Table 5, convergence is once again reached for all models using SMOTE, except for the SU model with regional and restaurant variables. In addition, we observe that our models now predict PC and TC more often when using the modified data, obtaining a higher recall than the models using the unmodified data for these groups. However, the prediction rates are lower than the results of the unmodified data shown in Table 5.

Including regional variables allows our models to obtain a higher prediction rate, but precision for the PC and TC classes are again very poor. Furthermore, operational observations are often falsely classified. This shows that the GLH models are not able to capture the class distribution of the modified data obtained by SMOTE. The Unpooled model for both data sets obtains the most a posteriori evidence, but still not enough according to the Jeffreys scale.

To summarize, even though the application of SMOTE resolves the issue of our imbalanced data set and improves the recall of the TC and PC classes, the recall of operational predictions decreases drastically, which results in low prediction rates. Our models are unable to capture the

class distributions and are thus still not trustworthy. We thus refrain from drawing conclusions based on these models since the unreliability might lead to an incorrect representation of reality.

	Restaurant variables									Regional + Restaurant variables								
	Pooled Model			Unpooled Model			SU Model			Pooled Model			Unpooled Model			SU Model*		
Prediction\True Value	O	PC	TC	O	PC	TC	O	PC	TC	O	PC	TC	O	PC	TC	O	PC	TC
O	3027	331	30	3079	300	27	3180	315	22	3432	275	24	3716	304	27	3757	306	26
PC	1129	858	7	1098	882	7	1069	846	6	1100	876	6	1115	855	7	1087	851	5
TC	2191	123	41	2170	130	44	1994	126	46	1711	136	44	1412	128	40	1399	130	43
Performance																		
Prediction Rate	50.74%			51.76%			53.55%			57.23%			60.64%			61.17%		
Precision	0.89	0.43	0.02	0.90	0.44	0.02	0.90	0.44	0.02	0.92	0.44	0.02	0.92	0.43	0.03	0.92	0.44	0.03
Recall	0.48	0.65	0.53	0.49	0.67	0.56	0.51	0.66	0.62	0.55	0.68	0.59	0.59	0.66	0.54	0.60	0.66	0.58
Bayes Factors	1			1.95			1.01			1.04			1.99			1.07		

Note: ∗ This model has not converged

Table 6: Contingency tables,prediction performance and Bayes factors for the Generalized Linear Hierarchical Models using out-of-sample predictions using SMOTE

6 Conclusion

In this paper, we investigated in what types of regions restaurants have a smaller probability of going bankrupt by answering the following two subquestions: (1) *"How many different types of zip code regions can we divide our zip codes into?"* and (2) *"How does the probability of bankruptcy vary between the different types of zip code regions?"*. We aimed to answer these subquestions by the first and second step of our two-step approach respectively, which would help us answer our research question: *"In what types of zip code regions do restaurants have a smaller probability of going bankrupt?"*. In Section 6.1 we briefly summarize the main findings of this research and answer the subquestions and the research question. Next, in Section 6.2 we propose suggestions for future research.

6.1 Concluding remarks

From the results, we observe that in the first step TS k-means is the best performing clustering method. We thus answer our first subquestion by concluding that our clustering method is able to divide zip codes into two different types of zip regions. In particular, the clusters from TS k-means are able to clearly distinguish the characteristics of the four biggest urban agglomerations (Randstad) from other areas in the Netherlands. Moreover, the advantage of our two-stage clustering procedure as opposed to a more naive approach where for example entire cities would be considered as clusters, is that our segmentation is able to create more specific partitions for the zip code regions.

As for the second step, the results of our GLH models are inconclusive, since the models

almost exclusively classify observations to the group operational and are thus unreliable. We resolved a big part of this problem by using Bootstrap Oversampling and SMOTE of the minority group. However, this resulted in a poor prediction performance for these modified data sets. Consequently, we are not able to answer the second subquestion and propose suggestions for our GLH models to possibly answer this question for future research as discussed in Section 6.2.

In summary, due to the poor performance of our GLH models we are unable to give a reliable answer to the second subquestion and subsequently, our research question. However, we did manage to find the types of zip code regions which we could divide our zip codes into. Thus, while we are not able to give a reliable answer to our research question, we are able to recommend our implemented clustering method TS k-means to cluster zip codes from regional data. The results obtained with this clustering method can be used to further evaluate the probability of bankruptcy.

6.2 Limitations and future research

The main limitations that we faced during our research were the limited time and resources we had access to. For this reason, we were unable to use all observations for training our models and used 3000 only. These limitations left us unable to test the performance of other prior specifications or running more iterations for our GLH models.

Apart from factors like regional variables or the quality of the food, DiPietro et al. (2007) propose eight restaurant-specific factors that lead to the success. These factors include: single unit operations, standard operating procedures, multi-unit strategic planning, interpersonal and social responsibilities, travel and visiting units, human relations, effective leadership, and unit level finances. As can be seen, these factors are restaurant-specific and examine how well the restaurant is being run. It is thus, of interest to study if including restaurant-specific variables like these improve the prediction performance of our models. If this is the case a question that can be asked is: *"Can regional variables predict the success of a restaurant?"* .

Another future research direction is to study a different dependent variable than the bankruptcy of a restaurant for our models, as an indicator of the success of a restaurant. In particular, some restaurants that are operational, but are struggling and close to going bankrupt are not captured as unsuccessful in our data set. Hence, it seems logical to study the success of a restaurant through a different variable. For example, Lee et al. (2016) capture restaurant success through different variables such as profitability, volume of sales, growth, enterprise performance and achieving expectations. An interesting question to study is: *"What factor measures restaurant success most carefully?"* .

References

Agarwal, R. and Dahm, M. J. (2015). Success factors in independent ethnic restaurants. *Journal of Foodservice Business Research*, 18(1):20–33.

Azur, M. J., Stuart, E. A., Frangakis, C., and Leaf, P. J. (2011). Multiple imputation by chained equations: what is it and how does it work? *International Journal of methods in Psychiatric Research*, 20(1):40–49.

Baraban, R. S. and Durocher, J. F. (2010). *Successful restaurant design*. John Wiley & Sons.

Barnard, J., McCulloch, R., and Meng, X.-L. (2000). Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica*, pages 1281–1311.

Bholowalia, P. and Kumar, A. (2014). Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9).

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 93–104.

Canetta, L., Cheikhrouhou, N., and Glardon, R. (2005). Applying two-stage som-based clustering approaches to industrial data analysis. *Production Planning & Control*, 16(8):774–784.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M. A., Guo, J., Li, P., and Riddell, A. (2017). Stan: a probabilistic programming language. *Grantee Submission*, 76(1):1–32.

Charrad, M., Ghazzali, N., Boiteau, V., and Niknafs, A. (2014). Nbclust: An r package for determining the relevant number of clusters in a data set.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Chen, J. and Shao, J. (2000). Nearest neighbor imputation for survey data. *Journal of Official Statistics*, 16(2):113.

Chen, L.-F. and Tsai, C.-T. (2016). Data mining framework based on rough set theory to improve location selection decisions: A case study of a restaurant chain. *Tourism Management*, 53:197–206.

Chou, T.-Y., Hsu, C.-L., and Chen, M.-C. (2008). A fuzzy multi-criteria decision model for international tourist hotels location selection. *International Journal of Hospitality management*, 27(2):293–301.

Chung, Y., Rabe-Hesketh, S., Dorie, V., Gelman, A., and Liu, J. (2013). A nondegenerate penalized likelihood estimator for variance parameters in multilevel models. *Psychometrika*, 78(4):685–709.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

DiPietro, R. B., Murphy, K. S., Rivera, M., and Muller, C. C. (2007). Multi-unit management key success factors in the casual dining restaurant industry. *International journal of contemporary hospitality management*.

Donders, A. R. T., Van Der Heijden, G. J., Stijnen, T., and Moons, K. G. (2006). A gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology*, 59(10):1087–1091.

Dosne, A.-G., Bergstrand, M., Harling, K., and Karlsson, M. O. (2016). Improving the estimation of parameter uncertainty distributions in nonlinear mixed effects models using sampling importance resampling. *Journal of Pharmacokinetics and Pharmacodynamics*, 43(6):583–596.

Du, K.-L. (2010). Clustering: A neural network approach. *Neural networks*, 23(1):89–107.

Ghosh, J., Li, Y., Mitra, R., et al. (2018). On the use of cauchy prior distributions for bayesian logistic regression. *Bayesian Analysis*, 13(2):359–383.

Gronau, Q. F., Sarafoglou, A., Matzke, D., Ly, A., Boehm, U., Marsman, M., Leslie, D. S., Forster, J. J., Wagenmakers, E.-J., and Steingroever, H. (2017). A tutorial on bridge sampling. *Journal of Mathematical Psychology*, 81:80–97.

Gupta, M. R. and Chen, Y. (2011). *Theory and use of the EM algorithm*. Now Publishers Inc.

Heij, C., Heij, C., de Boer, P., Franses, P. H., Kloek, T., van Dijk, H. K., et al. (2004). *Econometric methods with applications in business and economics*. Oxford University Press.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417.

Hox, J. J., Moerbeek, M., and Van de Schoot, R. (2017). *Multilevel analysis: Techniques and applications*. Routledge.

Jain, A. K., Acito, F., Malhotra, N. K., and Mahajan, V. (1979). A comparison of the internal validity of alternative parameter estimation methods in decompositional multiattribute preference models. *Journal of Marketing Research*, 16(3):313–322.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Ketchen, D. J. and Shook, C. L. (1996). The application of cluster analysis in strategic management research: an analysis and critique. *Strategic Management Journal*, 17(6):441–458.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480.

Laud, P. W. and Ibrahim, J. G. (1996). Predictive specification of prior model probabilities in variable selection. *Biometrika*, 83(2):267–274.

Lee, C., Hallak, R., and Sardeshmukh, S. R. (2016). Drivers of success in independent restaurants: A study of the australian restaurant sector. *Journal of Hospitality and Tourism Management*, 29:99–111.

Leyland, A. H. and Groenewegen, P. P. (2003). Multilevel modelling and public health policy. *Scandinavian Journal of Public health*, 31(4):267–274.

Lindley, D. V. and Smith, A. F. M. (1972). Bayes estimates for the linear model. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(1):1–41.

Lu, Y. and Zhou, H. H. (2016). Statistical and computational guarantees of lloyd's algorithm and its variants. *arXiv preprint arXiv:1612.02099*.

Ly, A., Verhagen, J., and Wagenmakers, E.-J. (2016). Harold jeffreys's default bayes factor hypothesis tests: Explanation, extension, and application in psychology. *Journal of Mathematical Psychology*, 72:19–32.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. Oakland, CA, USA.

Madigan, D., Genkin, A., Lewis, D. D., and Fradkin, D. (2005). Bayesian multinomial logistic regression for author identification. In *AIP Conference Proceedings*, volume 803, pages 509–516. American Institute of Physics.

Mangiameli, P., Chen, S. K., and West, D. (1996). A comparison of som neural network and hierarchical clustering methods. *European Journal of Operational Research*, 93(2):402–417.

McLachlan, G. J. and Basford, K. E. (1988). *Mixture models: Inference and applications to clustering*, volume 38. M. Dekker New York.

Mirkin, B. (2012). *Clustering: a data recovery approach*. CRC Press.

Nishio, M. and Arakawa, A. (2019). Performance of hamiltonian monte carlo and no-u-turn sampler for estimating genetic parameters and breeding values. *Genetics Selection Evolution*, 51(1):1–12.

Parsa, H., van der Rest, J.-P. I., Smith, S. R., Parsa, R. A., and Bujisic, M. (2015). Why restaurants fail? part iv: The relationship between restaurant failures and demographic factors. *Cornell Hospitality Quarterly*, 56(1):80–90.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Popat, S. K. and Emmanuel, M. (2014). Review and comparative study of clustering techniques. *International Journal of Computer Science and Information Technologies*, 5(1):805–812.

Raudenbush, S. W. (1988). Educational applications of hierarchical linear models: A review. *Journal of Educational Statistics*, 13(2):85–116.

Rouder, J. N., Sun, D., Speckman, P. L., Lu, J., and Zhou, D. (2003). A hierarchical bayesian statistical framework for response time distributions. *Psychometrika*, 68(4):589–606.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Rubin, D. B. (2004). *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons.

Sáez, J. A., Krawczyk, B., and Woźniak, M. (2016). Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition*, 57:164–178.

Schalkoff, R. J. (2007). Pattern recognition. *Wiley Encyclopedia of Computer Science and Engineering*.

Stegmueller, D. (2013). How many countries for multilevel modeling? a comparison of frequentist and bayesian approaches. *American Journal of Political Science*, 57(3):748–761.

Thadewald, T. and Büning, H. (2007). Jarque–bera test and its competitors for testing normality–a power comparison. *Journal of Applied Statistics*, 34(1):87–105.

Tzeng, G.-H., Teng, M.-H., Chen, J.-J., and Opricovic, S. (2002). Multicriteria selection for a restaurant location in taipei. *International Journal of Hospitality Management*, 21(2):171–187.

Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative. *Journal of Machine Learning Research*, 10(66-71):13.

Vesanto, J. and Alhoniemi, E. (2000). Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3):586–600.

Washington, S., Congdon, P., Karlaftis, M. G., and Mannering, F. L. (2009). Bayesian multinomial logit: Theory and route choice example. *Transportation Research Record*, 2136(1):28–36.

Yang, J. and Lee, H. (1997). An ahp decision model for facility location selection. *Facilities*.

Yang, Y., Roehl, W. S., and Huang, J.-H. (2017). Understanding and projecting the restaurantscape: the influence of neighborhood sociodemographic characteristics on restaurant location. *International Journal of Hospitality Management*, 67:33–45.

Zhang, W., Wang, J., Jin, D., Oreopoulos, L., and Zhang, Z. (2018). A deterministic self-organizing map approach and its application on satellite data based cloud type classification. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 2027–2034. IEEE.

A Data description

A.1 Regional data

Variable			
General Demographics	**Description**	**Applied Transformation**	**Source**
AMOUNT_HH	Number of households in a zip code area	Unchanged	CBS 2019
P_MAN	Percentage of men	$\frac{\text{MEN}}{\text{MEN + WOMEN}} * 100$	CBS 2019
P_VROUW	Percentage of women	$\frac{\text{WOMEN}}{\text{MEN + WOMEN}} * 100$	CBS 2019
AV_HH_S	Average household size	Unchanged	CBS 2019
P_NL_BACK	Percentage of people born with a Dutch background	Unchanged	CBS 2019
P_WE_MIG_B	Percentage of people with a western migration background	Unchanged	CBS 2019
P_NW_MIG_B	Percentage of people with a non-western migration background	Unchanged	CBS 2019
Age Demographics	**Description**	**Applied Transformation**	**Source**
P_INH_014	Percentage of inhabitants from age 0 to 14 years old	$\frac{\text{INH\_014}}{\text{INH\_014 + INH\_1524 + INH\_2544 + INH\_4564 + INH\_65PL}} * 100$	CBS 2019
P_INH_1524	Percentage of inhabitants from age 15 to 24 years old	$\frac{\text{INH\_1524}}{\text{INH\_014 + INH\_1524 + INH\_2544 + INH\_4564 + INH\_65PL}} * 100$	CBS 2019
P_INH_2544	Percentage of inhabitants from age 25 to 44 years old	$\frac{\text{INH\_2544}}{\text{INH\_014 + INH\_1524 + INH\_2544 + INH\_4564 + INH\_65PL}} * 100$	CBS 2019
P_INH_4564	Percentage of inhabitants from age 45 to 64 years old	$\frac{\text{INH\_4564}}{\text{INH\_014 + INH\_1524 + INH\_2544 + INH\_4564 + INH\_65PL}} * 100$	CBS 2019
P_INH_65PL	Percentage of inhabitants older than 65 years old	$\frac{\text{INH\_65PL}}{\text{INH\_014 + INH\_1524 + INH\_2544 + INH\_4564 + INH\_65PL}} * 100$	CBS 2019
Income Demographics	**Description**	**Applied Transformation**	**Source**
log_median_inc	Log of the median income	log(median_inc)	CBS 2017
P_LINC_HH	Percentage of households with low income	Unchanged	CBS 2017
P_HINC_HH	Percentage of households with high income	Unchanged	CBS 2017
P_SOCIAL_BEN	Percentage of inhabitants that receive social benefits	$\frac{\text{UITKMNINAOW}}{INHABITANTS} * 100$	CBS 2019
Geographical descriptives	**Description**	**Applied Transformation**	**Source**
DIS_RAMP	Average distance of all inhabitants to the nearest on-ramp of a highway, calculated over the road	Unchanged	CBS 2017
DIS_TRNTR	Average distance of all inhabitants to the nearest important transferstation, calculated over the road	Unchanged	CBS 2017
DIS_TRAINS	Average distance of all inhabitants to the nearest trainstation, calculated over the road	Unchanged	CBS 2017
AV1_FOOD	Number of food serving establishments (cafeterias, cafes and restaurants) within a radius of 1 km	AV1_CAFTAR + AV1_CAFE + AV1_RESTAU	CBS 2017
AV3_FOOD	Number of food serving establishments (cafeterias, cafes and restaurants) within a radius of 3 km	AV3_CAFTAR + AV3_CAFE + AV3_RESTAU	CBS 2017
AV5_FOOD	Number of food serving establishments (cafeterias, cafes and restaurants) within a radius of 5 km	AV5_CAFTAR + AV5_CAFE + AV5_RESTAU	CBS 2017
EAD	Average number of adresses per km^2 in a radius of 1 km around an adress.	Unchanged	CBS 2019

Table 7: Regional data description

General demographics	Missing Values	Min	Max	Median	Mean	Standard Deviation	JB test statistic	JB P-value
AMOUNT_HH	24	5.0	13475.0	1150.0	1958.953	2055.184	1689.194	0.0
P_MEN	25	0.333	0.917	0.502	0.506	0.028	141418.931	0.0
P_WOMEN	25	0.083	0.667	0.498	0.494	0.028	141418.931	0.0
AV_HH_S	24	1.1	4.3	2.3	2.287	0.319	558.697	0.0
P_NL_BACK	24	10.0	100.0	90.0	83.766	14.298	4777.643	0.0
P_WE_MIG_B	247	0.0	70.0	10.0	8.671	6.761	11421.515	0.0
P_NW_MIG_B	659	0.0	80.0	0.0	8.04	11.998	8870.273	0.0
Age demographics	Missing Values	Min	Max	Median	Mean	Standard Deviation	JB test statistic	JB P-value
P_INH_014	144	0.024	0.385	0.153	0.155	0.038	1671.351	0.0
P_INH_1524	144	0.037	0.621	0.117	0.122	0.034	180791.854	0.0
P_INH_2544	144	0.059	0.709	0.211	0.221	0.061	4288.148	0.0
P_INH_4564	144	0.065	0.526	0.303	0.303	0.052	300.828	0.0
P_INH_65PL	144	0.006	0.652	0.2	0.199	0.061	985.62	0.0
Income demographics	Missing Values	Min	Max	Median	Mean	Standard Deviation	JB test statistic	JB P-value
log_median_inc	540	8.732	11.471	10.558	10.525	0.221	1781.814	0.0
P_LINC_HH	521	2.9	88.4	33.3	35.5	11.821	360.646	0.0
P_HINC_HH	521	0.4	73.7	22.2	22.686	9.564	302.715	0.0
P_SOCIAL_BEN	241	0.006	0.568	0.065	0.074	0.038	76825.583	0.0
Geographical descriptives	Missing Values	Min	Max	Median	Mean	Standard Deviation	JB test statistic	JB P-value
DIS_RAMP	34	0.4	59.0	5.1	7.077	7.179	29786.305	0.0
DIS_TRNTR	34	0.5	71.8	12.1	14.283	10.534	2160.636	0.0
DIS_OPRIT	34	0.1	46.3	1.6	2.005	2.546	4473289.257	0.0
AV1_FOOD	34	0.0	696.6	2.8	10.8	31.544	1943694.504	0.0
AV3_FOOD	34	0.0	2664.2	15.8	70.464	187.346	757151.437	0.0
AV5_FOOD	34	0.0	3432.7	42.9	151.173	355.409	263603.856	0.0
EAD	2	2.0	11565.0	506.0	1038.855	1373.999	27150.293	0.0

Table 8: Summary statistics of the regional data

A.2 Restaurant Data

Variable			
Restaurant Variable	**Description**	**Applied Transformation**	**Source**
Rating	The average google review rating of the restaurant	Unchanged	UFS
log_nrRatings	The log-transformation of the number of google review ratings of the restaurant	log-transformation of *nrRatings*	UFS
Global_channel	Type of restaurant	Categorized in the categories: "Other", "Fastfood" and "No Dining"	UFS
Global_channel_no_dining	Contains the channels: retail bakery, cafe, bar, wine bar	Dummy variable for "No Dining" category	UFS
Global_channel_fastfood	Contains the channels: food delivery restaurant, fastfood restaurant	Dummy variable for "Fastfood" category	UFS
Global_channel_other	Contains the channel: other restaurant	Dummy variable for "other" category	UFS
Global_channel_dining	Contains the channels: bistro, sushi restaurant, pannenkoekenhuis, steakhouse, burger restaurant	Dummy variable for "Dining" category	UFS

Table 9: Restaurant data description

B Clustering results without outliers

B.1 Cluster performance

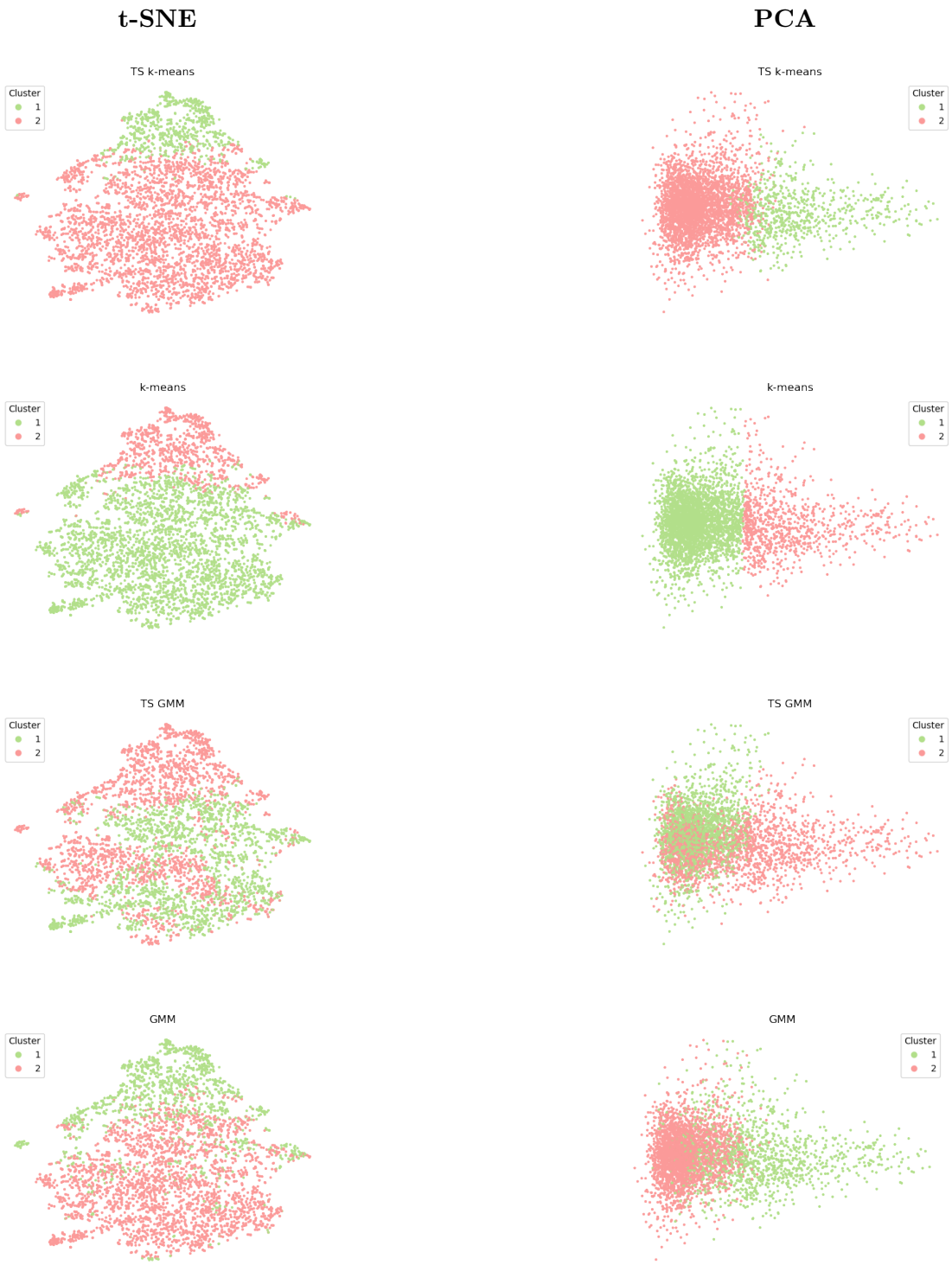

Figure 6: 2D visualisations of the regional data set without outliers by t-SNE (left column) and PCA (right column) with the optimal number of clusters from the 4 implemented clustering methods.

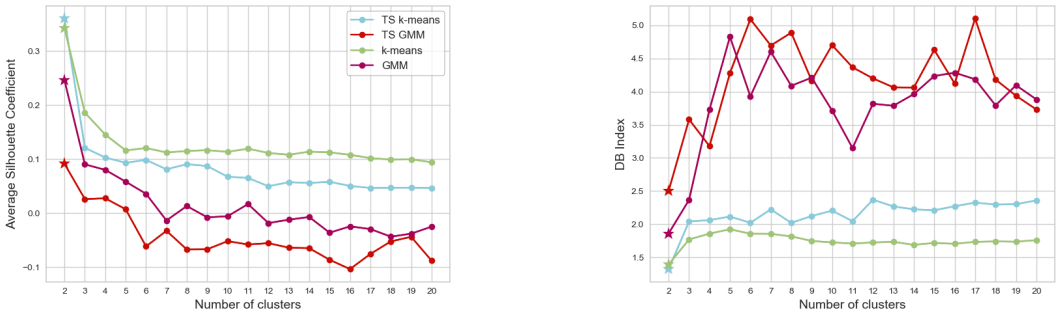

Figure 7: Plots of the average Silhouette Score and DB index against the number of clusters for the four implemented clustering methods using the regional data set without outliers

B.2 Cluster interpretation

	cluster 1			cluster 2			Overall		
General Demographics	mean	median	std	mean	median	std	mean	median	std
AMOUNT_HH	1468.375	780.000	1591.977	4472.423	4150.000	2154.415	1988.863	1215.000	2047.472
P_MEN	50.780	50.360	2.542	49.515	49.376	2.145	50.561	50.195	2.523
P_WOMEN	49.220	49.640	2.542	50.485	50.624	2.145	49.439	49.805	2.523
AV_HH_S	2.364	2.400	0.252	1.929	2.000	0.270	2.289	2.300	0.304
P_NL_BACK	88.595	90.000	7.784	61.654	60.000	15.797	83.928	90.000	14.046
P_WE_MIG_B	7.237	10.000	5.873	13.567	10.000	5.909	8.334	10.000	6.349
P_NW_MIG_B	3.208	0.000	5.342	24.321	20.000	15.013	6.866	0.000	11.247
Age Demographics	mean	median	std	mean	median	std	mean	median	std
P_INH_014	15.632	15.334	3.679	14.763	15.142	3.849	15.481	15.319	3.724
P_INH_1524	11.804	11.594	2.335	13.760	12.129	5.465	12.143	11.667	3.199
P_INH_2544	20.393	20.183	4.299	29.791	28.475	6.324	22.021	21.002	5.905
P_INH_4564	31.582	31.365	4.476	25.068	25.266	4.060	30.453	30.508	5.049
P_INH_65PL	20.590	20.462	5.538	16.618	16.313	5.922	19.902	19.981	5.804
Income Demographics	mean	median	std	mean	median	std	mean	median	std
log_median_inc	10.537	10.553	0.185	10.471	10.500	0.262	10.526	10.548	0.202
P_LINC_HH	32.556	31.700	8.851	47.726	48.600	11.565	35.184	33.000	10.996
P_HINC_HH	24.267	23.900	8.228	15.518	14.100	8.140	22.751	22.600	8.856
P_SOCIAL_BEN	6.491	5.962	2.693	11.184	10.900	3.831	7.304	6.435	3.419
Neighborhood Descriptives	mean	median	std	mean	median	std	mean	median	std
DIS_RAMP	1.890	1.500	1.912	1.948	1.800	0.935	1.900	1.600	1.781
DIS_TRNTR	15.863	14.000	10.010	5.617	4.000	4.852	14.088	12.000	10.097
DIS_TRAINS	7.772	6.000	7.090	2.790	2.100	2.245	6.909	5.100	6.782
AV1_FOOD	5.542	2.000	13.476	34.212	13.000	57.472	10.510	2.800	28.986
AV3_FOOD	30.525	10.200	59.354	258.941	124.000	369.633	70.101	15.800	184.550
AV5_FOOD	68.240	29.300	113.706	542.370	268.200	685.119	150.389	43.100	352.445
EAD	603.009	302.000	663.343	3170.669	2639.000	1854.858	1047.887	519.000	1379.954
Observations	3259			683			3942		

Table 10: Statistics of the overall zip codes and the clusters obtained by TS k-means on the regional data set without outliers

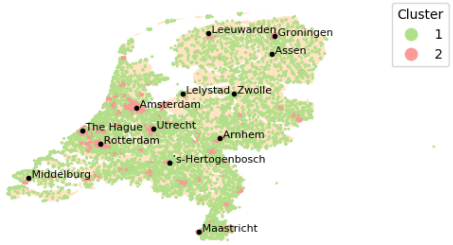

Figure 8: Map of the Netherlands, where the plotted points correspond to all the restaurants in the regional data set without outliers and with the clusters obtained by TS k-means.

	Restaurant type				Restaurant status			
	Fastfood	No dining	Dining	Other	O	PC	TC	Obs.
cluster 1	2.84	23.27	0.98	72.92	85.46	13.85	0.70	11064
cluster 2	3.78	22.91	2.37	70.94	79.29	19.46	1.25	11039
Overall	3.31	23.09	1.67	71.93	82.38	16.65	0.97	22103

Table 11: Characteristics of the restaurants in our resulting clusters from TS k-means on the regional data set without outliers. Note that the number of observations is less than in Table 4 as the restaurants corresponding to the excluded zip codes are also excluded.

C Bootstrap Oversampling technique

	Restaurant variables									**Regional + Restaurant variables**								
	Pooled Model			**Unpooled Model**			**SU Model**			**Pooled Model**			**Unpooled Model**			**SU Model***		
Prediction\True Value	O	PC	TC	O	PC	TC	O	PC	TC	O	PC	TC	O	PC	TC	O	PC	TC
O	2588	268	16	2907	231	18	2899	233	16	3186	244	19	3028	239	20	3434	262	22
PC	1265	911	7	1239	923	5	1246	922	5	1217	910	8	1207	898	8	1253	938	12
TC	2390	108	51	2097	133	51	2098	132	53	1840	133	47	2008	150	46	1660	112	44
Prediction performance																		
Prediction Rate	46.70%			51.04%			50.95%			54.49%			52.24%			57.1		
Precision	0.90	0.42	0.02	0.92	0.43	0.02	0.92	0.43	0.02	0.92	0.43	0.02	0.92	0.42	0.02	0.92	0.54	0.68
Recall	0.41	0.71	0.70	0.47	0.72	0.69	0.46	0.72	0.72	0.51	0.71	0.64	0.51	0.66	0.62	0.54	0.71	0.56
Bayes Factors	1			1.97			1.02			0.98			1.90			0.98		

Note: ∗ This model has not converged

Table 12: Contingency tables, prediction performance and Bayes factors for the Generalized Linear Hierarchical Models using out-of-sample predictions using the Bootstrap Oversampling technique

	Restaurant variables			**Regional + Restaurant variables**		
	Pooled Model	**Unpooled Model**	**SU Model**	**Pooled Model**	**Unpooled Model**	**SU Model**
Unmodified data	100%	100%	100%	100%	100%	18.5%
Oversampling	100%	100%	100%	100%	100%	13.4%
SMOTE	100%	100%	100%	100%	100%	15.9%

Table 13: percentage of variables, of which the $\hat{R}$ is ≤ 1.05, indicating convergence

D Code Description

In this appendix we present a brief description of the Python and STAN files used in this paper, which can be found at `https://github.com/StefanLam99/UnileverCase`. The descriptions of the files used for our clustering analysis and GLH models are given in Tables 14 and 15 respectively.

File	
Clustering methods	**Description**
SOM.py	Class to create an object which implements a SOM network as described by Kohonen (1990). It can be used to train a SOM network and for predictions of cluster labels.
two_stage_clustering.py	Class to create an object which implements the two-stage clustering procedure as described in this paper. The first stage is a SOM network and the second stage is either k-means or GMM. It can be used to train the two-stage clustering model and for prediction of the cluster labels.
Utility classes	**Description**
DataSets.py	Class which loads several data sets from the CBS and UFS to dataframes.
DataStatistics.py	Class which has functions to obtain statistics from several kinds of dataframes.
Utils.py	Class which contains simple functions used for the implemented clustering methods.
Main classes	**Description**
main_data_preprocessing	Main to preprocess the regional data set from the CBS.
main_cluster_validation.py	Main used to determine the optimal k for each method using our cluster validation measures.
main_clustering.py	Main to obtain the labels, plots and statistics of the implemented models with their optimal k determined from main_cluster_validation.py.
main_geopandas.py	Main to obtain the map of the Netherlands with as data points the restaurants corresponding to the label from the best clustering method.

Table 14: Description of the code used for our cluster analysis

File	
GLH Models	**Description**
multilog.stan	Stan model code of the Pooled model that is described in 4.2.2
unpooledmultilog.stan	Stan model code of the Unpooled model that is described in 4.2.3
su_model.stan	Stan model code of the SU model that is described in 4.2.4
Main classes	**Description**
main_data_preprocessing_GLH	Main to preprocess the restaurant data set from Unilever and merge it with the preprocessed zip code data set and the labels found in the clustering step.
main_stan_GLH.R	Main to run all the GLH models, obtain the predictions and create the confusion matrices.

Table 15: Description of the code used for our GLH models.