Income mobility within Italian households: an exploratory study on children's income decile prediction

Camilla Pelosi, Davide Bacigalupi, Matteo Dalle Luche, Gabriele Mole'

Contributing authors: c.pelosi@santannapisa.it; d.bacigalupi@santannapisa.it; m.dalleluche@santannapisa.it; g.mole@santannapisa.it;

Abstract

Italian households have long been thought as presenting a relatively stratified structure "in between" them, and a defined hierarchical structure "within". We attempt to exploit the National Income Registry provided by ISTAT to assess the grouping structure and the "within" and "between" factors driving heterogeneity in economic conditions across Italian households. In the present work, results show that, relying on a sufficiently low number of thoroughly defined variables regarding the structure and the economic backgroung, standard as well as ordinal classifiers do a satisfactory job in predicting economic wellbeing of following generations. We adopt both an unsupervised and a supervised learning approach to build upon this dataset and to highlight patterns across a highly-representative sample of the Italian population.

1 Introduction

Ever since their foundation, National Statistical Offices have been tracking and accounting for the distribution of national income across the households of the country they monitor. Such data provide evidence of evolving patterns within the national economy and the social system of a country.
In the present work, we try to manipulate the rich Italian Income Registry data provided by ISTAT to extract the emerging properties regarding the influence of the familiar background onto the future position in the social ladder of the pupils within

an household. In principle, income mobility and pre-distributive policies in a country should be in place to contrast the emergence of such patterns. We try to show that, within Italian households, there is a strong influence of household composition (the number and the position of the children with respect to the rest of his/her family) on the future economic position, but also parental education and disposable income have a very relevant role.
We first adopt an unsupervised learning approach: clustering shows that, from the most complete version of the data, Italian households have a grouping structure that is quite defined, while Principal Component Analysis shows that the aforementioned types of variables seem to explain most of the variability within the dataset.
We then exploit evidence from PCA select the features that should be the most explanatory to separate the deciles in the income distribution of pupils within the households. Our supervised learning exercise adopts a classification approach to show that classifiers such as the Quadratic Discriminant Analysis and the K-Nearest Neighbors manage to predict the income decile with satisfactory margins of error, though with very low accuracy given the complexity of the issue and the ordinal nature of our classes. Measures precisely tailored to deal with ordinal classification contexts show sufficient precision in assessing the income deciles, relying only on the six features we select (covering the aforementiones kinds of variables), though the approach of Frank and Hall (2001), standard in the literature about ordinal classification, does not improve results in a significant way. Cross-validation is then adopted to tune the models and better assess their precision.
Overall, we conclude that, though income is a complex variable to predict, household composition, as well as the economic and (most importantly) educational background (also thanks to the direct trasmission of the education level on the one of children) matters a lot in explaining future economic opportunities of the descendents. What is to grasp from this analysis is the crucial relevance of the position of children within the age ranking of the family.
This paper is organized as follows. Section 2 explains the background work to construct the dataset in a way that fits the need of the analysis: in particular, the information about the role of each individual within the household is first inferred from the age and sex structure. Section 3 explains the attempt to let grouping structure emerge from the data. Section 4 presents the results from PCA and the logical steps towards the selection of the predictors we will exploit in classification. Section 5 explains the standard and alternative models we used to attempt the classification of the income decile of the children. Section 6 validates such models to assess the improvement of results that are obtained adopting Frank and Hall (2001) approach, and concludes.

2 Data description

Every year, the Italian statistical office ISTAT collects data on income within Italian households, extracting information from modules available for the Income Registry: those are income data, either exposed to fiscal declarations or exempt from taxation but still traced by fiscal and social security sources, which include disposable income of every individual, before-tax gross disposable income, labour income (either wages

or from self-employment) and pensions income. These data exclude every form of rent, capital income (income from financial assets) and income coming from the shadow economy, which, to date, are not thoroughly traced in Italy (though they play a major role).
The data are provided in an horizontal panel structure across years 2015 to 2020 for each individual, accompanied with individual controls on sex, gender, citizenship and so on. Nonetheless, the unit of sampling is the household/family. Each individual, though anonymized, has a unique code within the sample, shares another unique identifier with all the components of his or her household, and all the components of each household of each individual in the sample are included as well in the dataset.
One of the main features of this dataset is its representativeness. The numerosity of the dataset that has been provided to us (almost 350.000 individuals, grouped in nearly 150.000 Italian households) is especially suited for our purpose, to apply supervised learning techniques and gather insight that are as robust as possible on the driving factors which influence intergenerational income transmission across generations within such households. The numerosity will be substantially reduced in the process of constructing our final dataset: nonetheless, it remains enough to apply more refined techniques (e.g. Quadratic Discriminant Analysis) with enough robustness.

2.1 Construction of the final family-level dataset

The purpose of this work is to apply unsupervised learning techniques to identify clusters across Italian households and to gather predictive insight on the factors, both individual and coming from the familiar background, driving children to belong to a specific income decile.
However, as pointed out before, the statistical unit within the dataset is the individual, not the family. Moreover, and most importantly, ISTAT does not disclose any information regarding the actual "role" or "position" of the individual within the household (whether the individual is a children, a parent, a grandparent and so on.).
The preliminary stages of the background work that has led to the dataset here used had two main purposes: first, to impute, through reasonable deduction, the role that each individual has within his or her household, and second, to rebuild the dataset to account in order to have each row corresponding to a household and allow household information to be plugged into each model we are going to estimate.
For the sake of brevity and since it is outside the scope of this paper, we will not go through the details of how these operations have been conducted. The general approach, given that there is no way to disentangle with certainty, for example, a parent from an uncle that lives in the same household, we have concentrated on the reconstruction of generational relations, rather than the precise parental bond between individuals. This said, we made sure not to label any individual on which there was significant uncertainty. For example, on a first approximation, a parent is an individual that has significant age difference with respect to one or more individuals below him/her: we would expect to have one individual or two individuals or discordant sex to be the parents within the same household. Thus, if there were three or more individuals within the same age range, this household has not been classified due to the uncertainty on who are the actual parents and who is the third non-parent person and

what is its actual role within that household (is him/her an uncle? A cousin of one of the two parents? Etc.). Therefore, households containing three or more individuals on within an age range referred to a later generation (parents, grandparents...) have not been classified. As a matter of fact, this restrictive, though necessary, criterion has not lead to significant losses of numerosity in the final dataset.
These operations excluded from the final dataset single-individual households, households composed by two individuals that have no significant age difference between them and households composed by nine or more individuals. The first two are self-explaining (they provide no information about intergenerational transmission of income within household), while for the last category of households, we stumbled upon the problem of household numerosity, which made our role imputation technique less and less reliable. This is due to the presence of more and more numerous age classes in later generation which, as mentioned before, lead to non-classification of that household. Of course, if an individual is labeled by the role classification algorithm, any other individual in the same household is classified as well.
The result is that each household corresponds to a pattern of parental bonds within those disposable for its numerosity (from two to eight). Within each households, we have identified at most two great-grandparents, at most two grandparents, at most two parents and at most seven children, mainly exploiting the age structure of the household. The final dataset has then been built "gluing" together the rows referring to each member of the family, with all their covariates (individual controls and values for each income in each year).
Each row then has a first part that contains some household-specific information which do not vary by individuals:

- `mem`: number of members of the household.
- `reg`, `zone`: geographical controls, the latter refers to the standard division of Italy into five macro-zones.
- `n_sons`: the number of children within the household, computed as the number of members minus the ranking of the age of the eldest children minus 1.
- `any_nonni`: a dummy indicating the presence of at least one grandparent living in the household.
- `onlychild`: a dummy indicating if there is only one child (an only-child).

The rest of each row is composed by 13 sets of 38 columns referring to each family member: the two great-grandparents, the two grandparents, the two parents and the seven children, ordered by seniority. If the household does not have a particular member, its values for the correspondent covariates of that member are all NAs (e.g. if the household has only three children, then all the values for the columns after those of the third child are all NAs).
Since the methodologies here applied do not explicitly exploit the time structure of income data, all the incomes for each individual have been averaged out across the six years available. If there are missing values in the time series of income, they have been ignored: meaning, there is a value for average income across years for each individual that has at least one non-missing observation.
The variable to be predicted in our work is the **income decile of each children** (in terms of disposable income) within each single classified family. We then took

together all the incomes of children in order of birth, storing them in a unique vector: we defined the threshold that defined the deciles. We then assigned to each children a number ranging from 1 (1st decile) to 10 (10th decile), to be predicted as a factor outcome variable, basing on his or her income lying between the threshold of the correspondent decile.
The final dataset will then be a slight rearranging of the one we have constructed during the previous passages. Children, in the present configuration, are in different columns, in order of birth: in order to use each of them as one single instance in training the model, we replicated family-level data for as much rows as the number of children in that household and assigned one children to each row of his or her corresponding family.
Lastly, we attempted to find a way to blend parents' information in one variable only, as a matter of uniformity: approximately, 40% of households have one single-parent only, therefore we searched in the literature for ways to harmonize information about parents' education levels and disposable income, while accounting for the diverse composition. We decided to take the average between disposable income of the two parents storing it into the variable `gYD`, since a substantial portion of households having both parents missed the information about the income of one of the two, therefore there was no reliable way to privilege households having both parents. Instead, we have the education level of all parents, therefore, we constructed the proxy for family's overall education level of parents `educ` which is equal to the education level of the single parent, if the household only has one parent, and instead, when they are both present, merges the education levels of parents e_1 and e_2 as follows:

$$\texttt{educ} = 1 + \max(e_1,\ e_2) + \frac{1}{2} \cdot \min(e_1,\ e_2) - \frac{1}{5} \operatorname{var}(e_1,\ e_2)$$

Doing so, we intended to create a proxy ranging from 1 to 10 of the overall education level, that privileged households having both parents, being endowed with an overall higher education level, and that penalized dispersion in education levels between parents (thus providing a higher premium when there are complementarities in education levels).
Lastly, we shall also mention the predictor `agediff`: it is the difference between the age of the individual and the average age of the household, proxying for hos or her role within the family. In principle, the higher it is, the more important the relative position of the individual is expected to be, also in economic terms: it can mean that he/she is the elder children, or that his/her parents are relatively young, meaning it is likely that he/she is expected to contribute to the economic life of the household.

3 Clustering

3.1 Clustering across family structures

Dealing with NAs in this dataset is crucial. As it can be seen from the graph below (Figure 3.1), the actual majority of all observations is missing. This is because of the nature of the dataset, which shows only the values for existing individual in that household. For example, if the second parent does not exist in that household, then all his/her values would be an NA. In addition to this, NAs are present also among the existing individuals' data and should be addressed in order to run clustering.

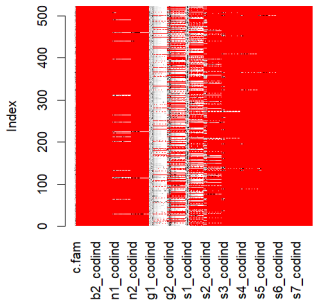


Fig. 1 *Dataset and missing values: red values indicate NAs. It is evident that at least one parent (g1) and a son (s1) exist all across the dataset. Moreover, we see that most parents have an income, while children do not so often.*

Exploring the dataset, it can be noticed that there are huge percentages of missing values among the great grandparents, the grandparents and the sons after the third. We decided to summarise their information creating two variables for each category: we built a counter which identifies the number of great grandparents, the number of grandparents and the number of sons based on the their presence in the dataset (signalled by the presence of their identification code). As far as income is concerned, the mean is taken out of the available income over the 6 years. The resulting two variables are then number and mean available income over the 6 years.
In general if an individual exists but has no average available income for a certain category a 0 is imputed, based on ISTAT guidelines, which stated that no declared income resulting in NA was meant a null income. Further operations on the dataset were conducted in order to have NAs only for non existing individuals in a household. In particular, income is averaged on the 6 years using the available data and divided into 5 types of categories, average available income, average total income, self-employment income, income from employment and income from pension. Then education, a value missing for individuals under 15 years old by ISTAT guidelines, is imputed based on the age.
Since the nature of the dataset is such that families with different structures have NAs left for non existing individuals, and considered that it would be interesting to

see if different patterns emerge depending on the number of people in the household, we identified 6 types:

- Type 1: single parent with only-child.
- Type 2: single parent with two children.
- Type 3: single parent with three or more children.
- Type 4: two parents with only-child.
- Type 5: two parents with two children.
- Type 6: two parents with three or more children.

So, the more people in the family, the more variables are taken into account when clustering.

3.2 Clustering analysis

Cluster analysis refers to idea of finding homogeneous subgroups in a dataset, which are similar within groups and distinct across groups. n observations are measured on a set of p continous features. In our case there are 51371 observations measured on 61 features. We first adopted the k-means algorithm, which identifies subgroups forcing every point into a definite number of clusters based on the minimization of the within cluster variation. Through this minimization, a local optimum is secured. We then opted for a hierarchical clustering, which instead a bottom-up approach, adopting a complete linkage function.
Generally speaking, clustering analyses are identified based on continuous variables. In our case, many variables are categorical and are included too. To cope with this, we use the Gower distance function, which is useful to find clusters in dataset with both categorical and continuous variables. The general form is this:

$$\text{Gowerdistance}(x_1, x_2) = 1 - \left(\frac{1}{p} \sum_{j=1}^{p} s_j(x_1, x_2) \right) \tag{1}$$

First introduced by Gower (1971), Gower's distance can be used to measure how different two records are. The records may contain combinations of logical, numerical, categorical or text data. The distance is always a number between 0 (identical) and 1 (maximal dissimilarity). In short, Gower's distance (or similarity) first computes distances between pairs of variables over two data sets and then combines those distances to a single value per record-pair. For qualitative descriptors, in particular, Dice distance is calculated. Whenever the values are equal, Dice Distance = 0 and when they're not equal an algorithm calculates the value as the ratio of number of non equal dimensions and the sum of the dimensions where the values are true and the number of non zero dimension (id est in the cases where at least one of the values is true). Because both types of distances are in a scale of 0 to 1, this makes it possible to compare categorical ones.
To evaluate the optimal number of optimal number of clusters to be used, average silhouette has been used fo k-means: it measures the similarity of each point to its cluster, and compares that to the similarity of the point with the closest neighboring

cluster. Overall the values of silhouette were quite low, but it depends on the type of family. Because of the risk of overlapping, we decided to proceed with hierarchical clustering and adopting a complete linkage function, which maximises the dissimilarity. These have been evaluated through the within-clusters dissimilarity.
The results for both k-means (Figure 3.2) and hierarchical clustering (Figure 3.2) are plotted thereafter. Because categorical variables have been used, the plots of the results of the k-means algorithm show on the x and y axes the results of the minimization of Kullback-Leibler divergences. This is because we adopted a different technique to visualise data, called t-Distributed Stochastic Neighbor Embedding (t-SNE). In brief, this plots high-dimensions data on a two-dimensions plot, based on the minimization of the difference of two probability distributions. The first one is the distribution built on the distance between points in the high-dimensions space: similar objects are assigned a higher probability while dissimilar points are assigned a lower probability. Then a similar a distribution over the points is defined by t-SNE in the low-dimensional map, and it minimizes the Kullback–Leibler divergence (KL divergence) between the two distributions with respect to the locations of the points in the map. The general tendency seems that in k-means clustering sub-groups become clearer as the number of individuals (and then of variables included) increases. The results of hierarchical clustering show, instead, a common trend of four clusters, despite not being so well defined: dendrograms are actually very dense and no clear pattern is immediately intuitable.

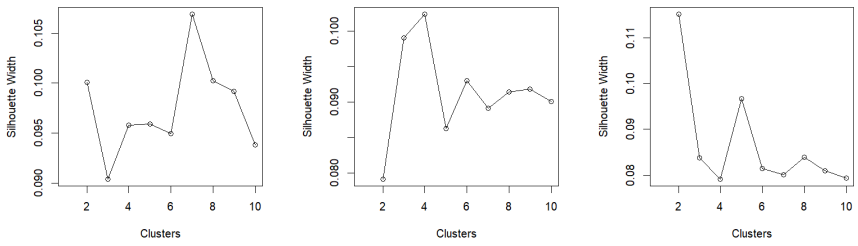


Fig. 2 *Silhouette width for family type 1, 2 and 3 respectively*

Fig. 3 *Clusters for family type 1, 2 and 3 respectively*

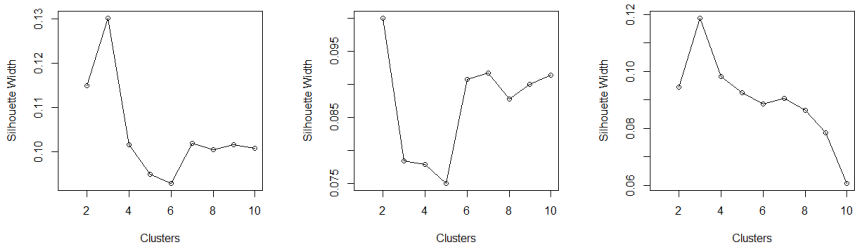


Fig. 4 *Silhouette width for family type 4, 5 and 6 respectively*

Fig. 5 *Clusters for family type 4, 5 and 6 respectively*

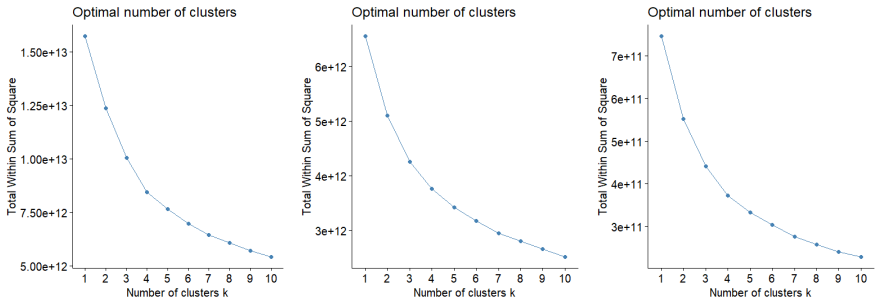


Fig. 6 *Within clusters dissimilarity for family type 1, 2 and 3 respectively*

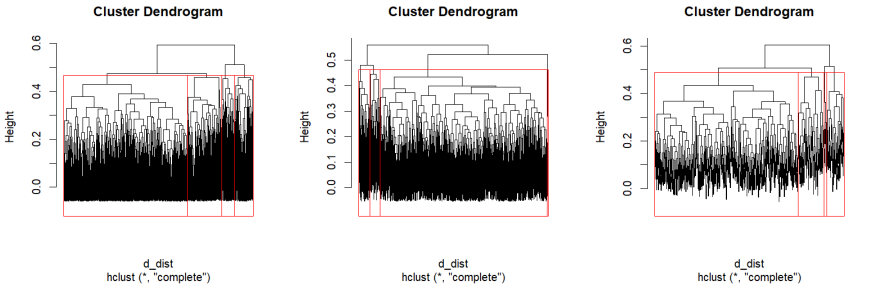


Fig. 7 *Clusters for family type 1, 2 and 3 respectively*

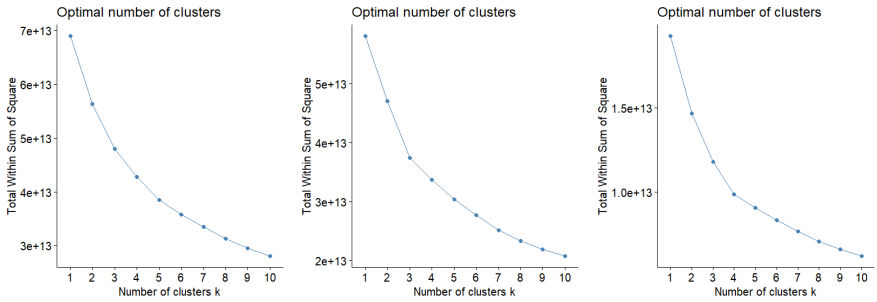


Fig. 8 *Within clusters dissimilarity for family type 4, 5 and 6 respectively*

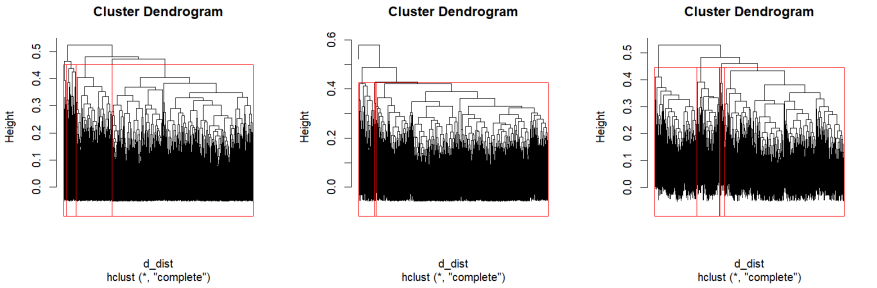


Fig. 9 *Clusters for family type 4, 5 and 6 respectively*

4 Features selection

Passing on, to address the aim of the present analysis to predict the income decile of children basing on the familiar background, we need to reduce the dimensionality of our dataset. We proceed first by logical deduction on which features to consider and which to eliminate with respect to previous section, then we perform Principal Component Analysis to further reduce the dimensionality.
The clustering section relied on a more complete version of our dataset (comprised of over 50 features for clustering analysis), considering the breakdown of each possible source of income, which could in principle be useful to explain the emergence of patterns on different types of households, but which cannot assume a predictive role, given that they are nested into the total disposable income of the individuals.
From now on, then, we will consider only total disposable income to assess the economic conditions of the family background of individuals. Of course, as far as children are concerned, this information is condensed in the income decile, which is the feature we would like to predict.
By that, we are left with some variables at family level, such as geographical location (`reg`, `zone`) and the variables accounting for the composition (`mem`, `n_sons`, `any_nonni`), and all the controls at individual level (sex, age, his/her ranking in terms of seniority within the household and `agediff`, citizenship, education level and disposable income).
We decided to retain only the variable `zone`, in order to avoid redundance with the region, which is too granular to have a predictive role. Then, `mem` was removed since it provides duplicate information that can be inferred from the other variables accounting for the composition (the number of children and grandparents living with them), so we retained only `any_nonni` and `n_sons`.
The sex of parents was dropped since it is trivial information. Absolute age was as well dropped, since it does not say anything about the role of that individual within an household (e.g. a 40-years old can both be a children of two elderly parents or a parent of two children): what matters, in predictive terms on income, is the position of the children within the household, meaning the his/her relative age. Hence we retained only the `agediff` of the children, dropping the one of the parents, since it is directly related to the one of their children. Since it provides the same information (though less refined), we also dropped the ranking of parents and children. The citizenship of parents and children is almost always the same (over 97% of the times), so we only retained the one of the children. Lastly, we considered the income and education level of parents as described in Section 2.1 and we also considered the education level of children.
By that, we are left with ten predictors and the income decile of children as the encoded variable to be classified. We deepened our analysis using unsupervised learning techniques with Principal Component Analysis, in order to further our dimensionality reduction procedure and features selection in a more rigorous way.

4.1 PCA

The selection of procedure presented so far used mainly logical deduction. We now want to do exploratory data analysis to see if there is some of the variables we retained that in principle can be dropped, due to its low "informational" value added to predict the income decile of the children. In order for the classification procedure to be as accurate as possible, we need to de-noise the data while losing the least amount of information possible. We conduct this operation of capturing the most important dimensions of variability through Principal Component Analysis.
We present our results using *unscaled* features. We also performed PCA on rescaled data (see the R code: first, rescaling only continous features, then also binary and categorical features), but our results do not change substantially, therefore we stick to the original version of our dataset.
The first two principal components, though they seem to explain a relatively low share of the variance present in our data, will guide most of the decisions we will do in what follows regarding features selection. We will prioritize the variables that have the highest projection on the first two principal components, focusing on the loading vectors and trying to observe which variables weight more.
From Figure 10, we see that the elbow in the Percentage of Variance Explained occurs on the sixth dimension, and we manage to explain more than 80% of the total variance. The first component, that explains most of the variance, combines mostly `n_sons` and `onlychild`: indeed, there variables are highly correlated, since of course living in a family with only one child means being an only-child. These two variables, together with `agediff`, explain the totality of the first component: this means that the first component is mostly influenced by the composition of the household and the relative position of the child considered. The biplot (Figure 11), confirms this evidence. From the second component, economic and educational variables emerge by importance (`gYD`, `educ` and `sons`). Individual controls, instead, weakly account for the variability in the data.
The six variables mentioned before will then be selected, in what follows, to perform our classification exercise.

	Eigenvalues	Variance explained (percent)	Cumulative variance explained
Dim. 1	2.539	25.391	25.391
Dim. 2	1.597	15.971	41.362
Dim. 3	1.239	12.388	53.750
Dim. 4	1.034	10.343	64.093
Dim. 5	0.974	9.737	73.829
Dim. 6	0.898	8.977	82.807
Dim. 7	0.677	6.772	89.579
Dim. 8	0.568	5.684	95.263
Dim. 9	0.244	2.438	97.701
Dim. 10	0.230	2.299	100

Table 1 *Outcome of PCA*

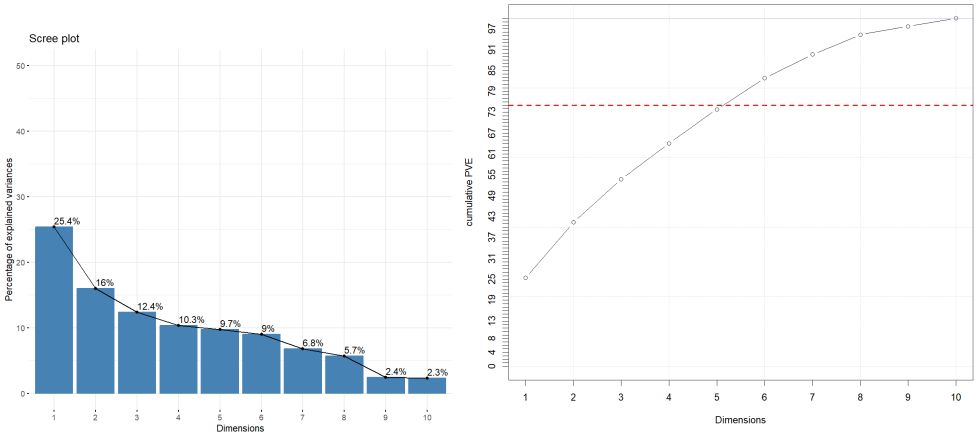


Fig. 10 *Percentage of variance explained: discrete and cumulative.*

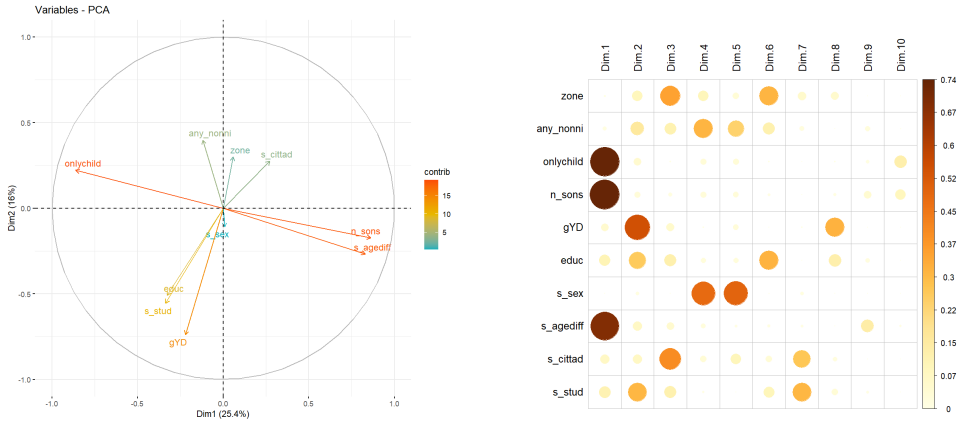


Fig. 11 *Relative contributions along the first two principal components (left) and along the other components (right).*

5 Classification

5.1 Choice of the classifier

The general idea behind classification is to train an algorithm to predict the categorical response of a unit (in our case, the income decile of the children of a family) on

the basis of the values of selected features. Given the results obtained from PCA, we selected the following variables as features (see section 2.1 for a detailed explanation of their content): `onlychild`, `n_sons`, `gYD`, `educ`, `s_agediff`, `s_stud`).
During the initial stages, we abstained from using Logistic Regression since our dataset comprised 10 ordinal classes, rather than a binary outcome as required by Logistic Regression. Secondly, we examined the differences between Quadratic Discriminant Analysis (QDA) and Linear Discriminant Analysis (LDA) to assess which strategy was more appropriate to our problem. Due to the non-Gaussian distribution of our features and the presence of trends in their variances, we did not have common variance-covariance structure across deciles. We see that breaking down the variance-covariance structure across deciles, the relative covariances change (see Figure 13) and the variance change across deciles (see Figure 14).

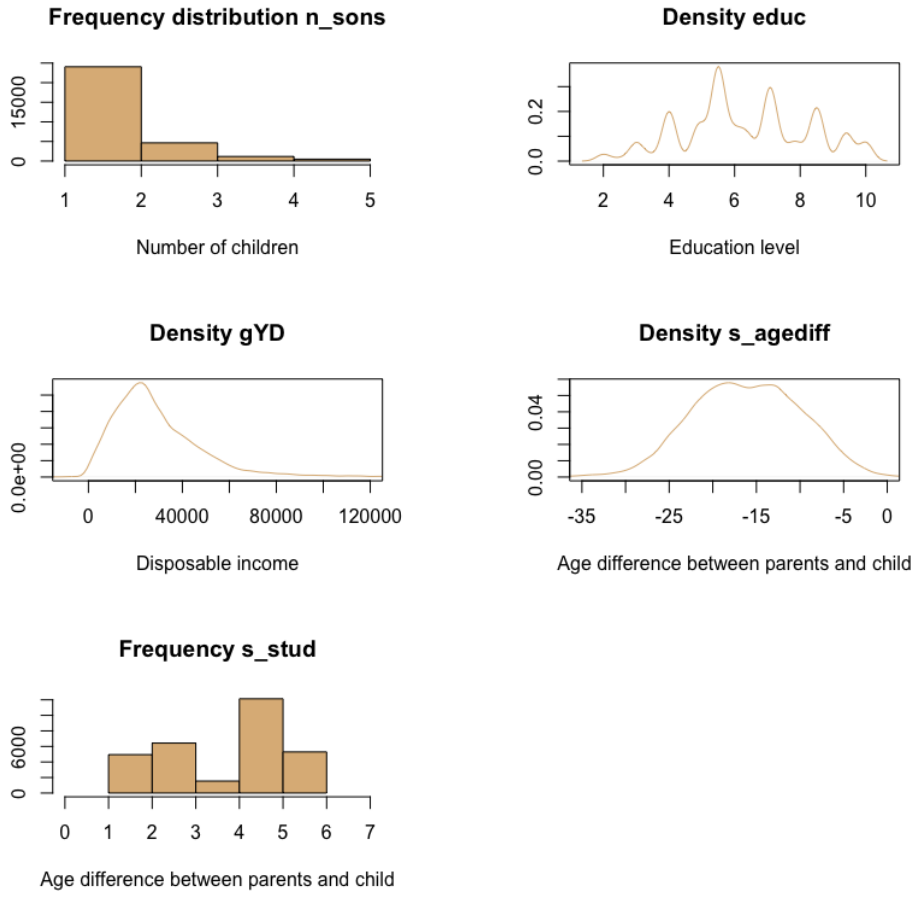


Fig. 12 *Distribution of selected features: we see clearly that most of them are not normally distributed, providing an argument against the use of LDA.*

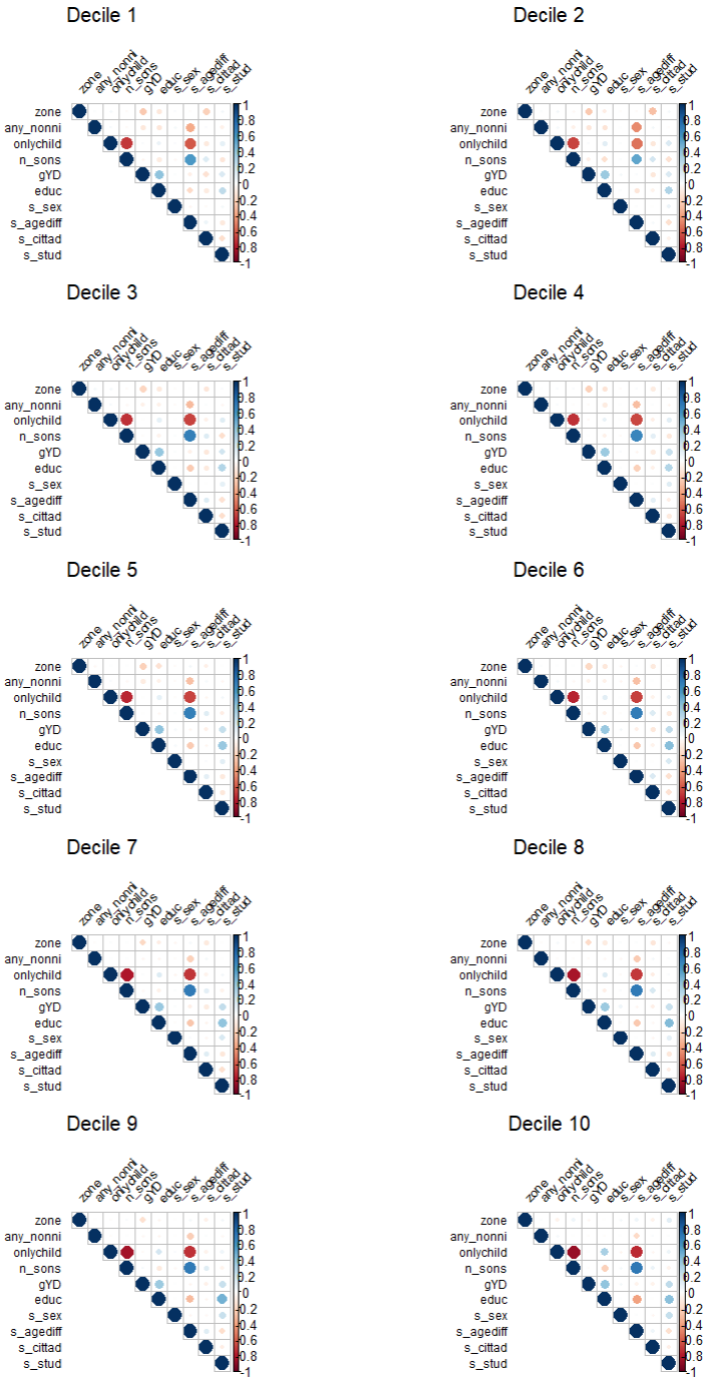


Fig. 13 *Correlation plots of the features: the relative structure of covariances changes.*

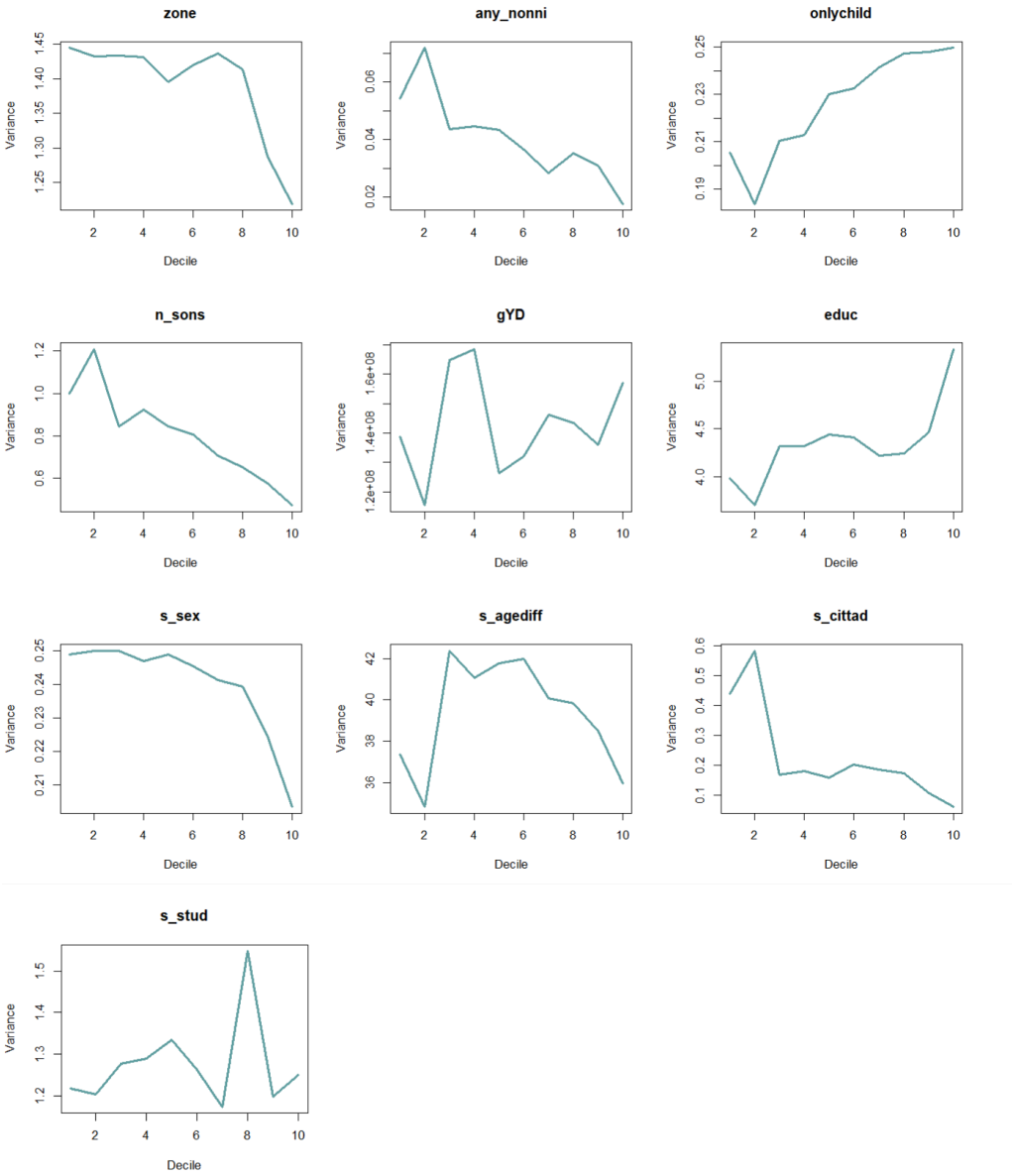


Fig. 14 *Trends in features' variances*

Also, the plots representing the frequency distributions (Figure 12) and densities of the selected features show that they do not behave as a Normal distribution.

`onlychild` (not reported) is of course a Bernoulli variable. `n_sons` has a Poisson distribution, `educ` has a multimodal distribution and s_stud has a bimodal distribution. `gYD` in principle could be seen as distributing as a Normal distribution in logs (thus having a log-Normal distribution), but, since we also have negative values for disposable income, we cannot take the log transformation of a relevant chunk of data: hence, the distribution of this feature is skewed. Finally, `s_agediff` seems the only explanatory variable that could be approximated by a Normal distribution.
For all these reasons, we needed to provide our classifier with the highest possible flexibility: our large sample size allowed us to safely use QDA, which does not rely on normality assumptions on the features and accounts for non-linear boundaries between classes. Based on these observations, we opted to focus on QDA as it better suited for the characteristics of our dataset.
To explore various options, we also conducted trials with the K-Nearest Neighbors (KNN) algorithm. Initially, we experimented with reasonably low k values ranging from 1 to 10. In what follows, we then proceed to analyze a better suited range of ks for our kNN classifier.
During this initial analysis, we observed that the algorithms we used neglected the fact that the ten classes we employed were not independent, but were tied by a clear form of ordering (the deciles of the income distribution). As a result, these algorithms tended to classify observations into the extreme values of the ordinal scale, disregarding the nuances present within the intermediate classes. In what follows, we employ the standard approach (Frank and Hall, 2001) that are available in the literature to perform classification tasks on ordinal categories. Our objective then became to minimize the distance between the predicted and observed classes for each unit, considering the interdependence among the ten ordinal classes.

5.2 Frank and Hall (2001) approach on ordinal classes

To address this issue and better account for the ordinal nature of the response variable, we explored the literature to identify classification methods that consider the ordinal nature of labels. The goal was to find algorithms and validation measures for them that would account such setting, meaning, that would enable us to minimize the distance between the predicted and observed classes on individual units. This is also due to the granularity of our classes: being ten, classifiers accuracy tends to be very low, and we want to use such kinds of measures to better validate our algorithms with a different goal in mind, to get as close as possible to the actual income decile.
Following the findings in Frank and Hall (2001), their approach for handling ordinal classes relies on the construction of nine binary outcome variables by categorizing the observations as being above or below a certain income decile. Since we had $K = 10$ classes, this resulted in 9 outcome variables, attached to each observation in nine separate datasets (see Table 2).
Using the constructed binary datasets, we computed differential probabilities of being above each class, as suggested in the Frank approach, and we make predictions with all the proposed classifiers assigning to each observation the class for which it had the maximum predicted likelihood. We also included logistic regression, since under this

approach we have binary outcome variables that are properly constructed, taking into account the ordinal structure. Figure 15 illustrates the methodology.

Quantile		Q> 1	Q> 2	Q> 3	Q> 4	Q> 5	Q> 6	Q> 7	Q> 8	Q> 9
9		1	1	1	1	1	1	1	1	0
6		1	1	1	1	1	0	0	0	0
2		1	0	0	0	0	0	0	0	0
7		1	1	1	1	1	1	0	0	0
4		1	1	1	0	0	0	0	0	0
2	$\longrightarrow$	1	0	0	0	0	0	0	0	0
9		1	1	1	1	1	1	1	1	0
1		0	0	0	0	0	0	0	0	0
4		1	1	1	0	0	0	0	0	0
2		1	0	0	0	0	0	0	0	0

Table 2 *The transformation we adopted on our outcome classes: first ten instances of our dataset.*

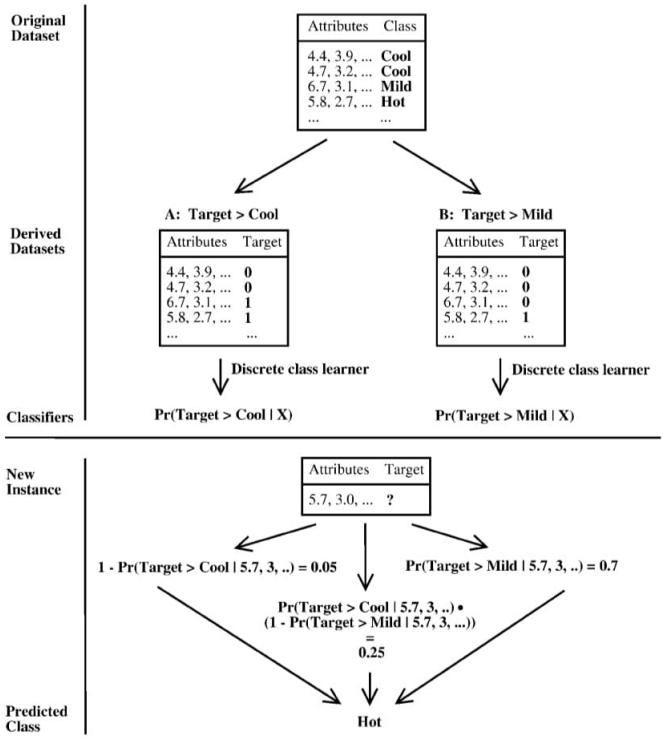


Fig. 15 *From Frank and Hall (2001): building a proper set up for ordinal classes classification.*

From the literature, it emerged that the most commonly utilized metrics for validating classification algorithms with ordinal outcomes are Accuracy, Mean Squared Error (MSE), and Mean Absolute Error (MAE) (see, for example, Cardoso and Sousa (2011)). Accuracy is of course extremely less powerful due to us using very disaggregated income decile classes and due to the tendency of standard algorithms to predict extreme classes. Therefore, onwards, we will rely mostly on ad-hoc measures being MAE and MSE: these are computed as

$$MAE = \frac{1}{N}\sum_{r=1}^{K}\sum_{c=1}^{K} n_{r,c}|r-c| \quad MSE = \frac{1}{N}\sum_{r=1}^{K}\sum_{c=1}^{K} n_{r,c}(r-c)^2$$

where each entry $n_{r,c}$ represents the number of points from the r-th class predicted as being from the c-th class, and K is the number of classes. These measures increase the more the classification has a bad outcome, therefore we adopted a normalization ($1 - MAE\%$ and $1 - MSE\%$) of such measures in order to have them increasing in the performance of the algorithm and constrained between 0 and 1, to allow for comparability with accuracy.
Since this approach aligned well with our framework, we focused on optimizing the parameter k for the KNN classifier under this set up. We observed that the three evaluation metrics (Accuracy, MSE, and MAE) were all monotonically increasing within the range of $k = 1$ to 10. To determine the optimal value of k, we built a loop to fit multiple kNN models with different target variables and values of k, to generate predictions and evaluate them against the true target values. The results are shown below. Through our experimentation, we achieved varying peak values for (1-MSE%) and (1-MAE%) with the same value, being $k = 418$: given our instances of training and test sets, a kNN classifier with $k = 418$ simultaneously maximised both of our metrics of interest, as it can be seen from Figure 16. Consequently, we selected $k = 418$ as the optimal value and obtained the relevant metric values for the KNN classifier for the six chosen classifiers.

In conclusion, in Table 3, we present the absolute performance outcomes on the three evaluation metrics under the five classifiers we eventually adopted (the kNN is employed considering only the tuned parameter). On our instances of training and test sets division, the kNN classifier with $k = 418$ under Frank and Hall (2001) approach outclasses the performance of the other classifiers, and the adoption of a set up that accounts for ordinal classes provides substantial improvement of performances. The QDA classifier is instead the weakest and suffers a loss of performance when the approach of Frank and Hall (2001) is adopted. Logistic regression also provides satisfactory results, approaching 75% precision under Mean Average Error and 89% using Mean Squared Error.

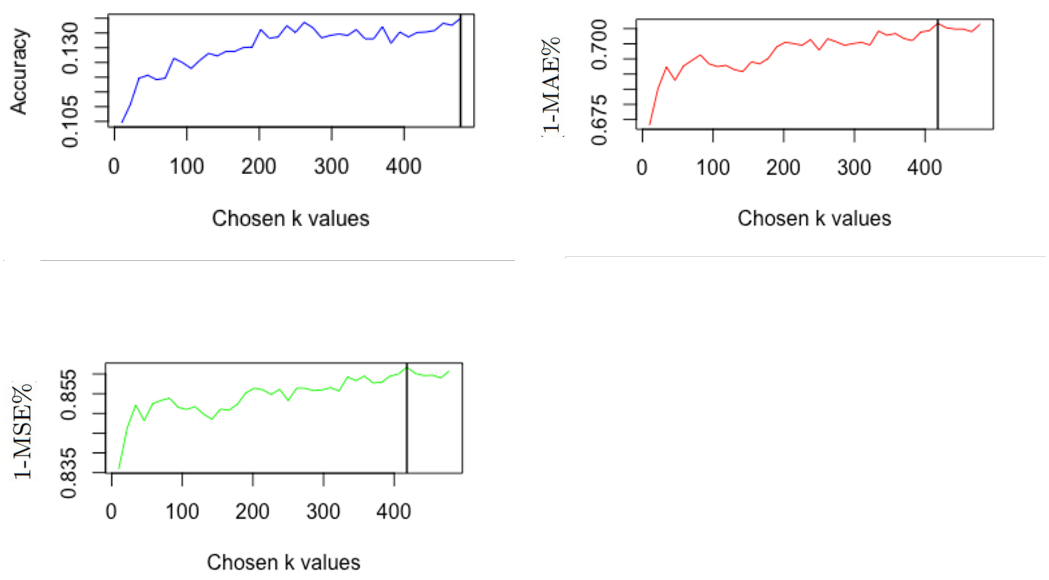


Fig. 16 *The performance of the kNN classifier under the three evaluation metrics under a selected set of different k values.*

	Accuracy	(1-MAE%)	(1-MSE%)
QDA (Ordinal)	0.173	0.720	0.865
Logistic (Ordinal)	0.191	0.751	0.891
KNN418 (Ordinal)	0.202	0.758	0.895
QDA (Standard)	0.183	0.728	0.871
KNN418 (Standard)	0.176	0.730	0.875

Table 3 *The final performance of the considered classifiers under the considered training and test sets.*

6 Cross-validation

6.1 Choice of the best CV approach

The main purpose of the cross-validation is to give a more robust assessment of the performance of the models seen above, helping us to choose the best one.
Before we used for the sake of simplicity and easiness of calculation the validation set approach, by randomly dividing the set of all observations into two parts: a larger training set and a relatively small validation set. Then we fitted the model on the training set and we used the validation set to evaluate its predictive performance. This approach is very important in order to give us a rough idea of the overall behaviour of

the models, but its results are not robust, because the estimate of the error rate/accuracy can be highly variable depending on which observations you choose to be on each set, and furthermore we use only a fraction of the data to estimate the model, wasting information and so making the estimate of the test error rate/accuracy potentially higher than the real one.
Thus, we can use several different approaches to solve these issues.
We can do the LOOCV (Leave-One-Out Cross-Validation) approach, in which we train the model using one single observation as a validation set and all the remaining $n-1$ observations as a training set; then we repeat the procedure for all n observations and the LOOCV estimate is the average of the n test error/accuracy estimates.
Or alternatively we can do the k-fold Cross Validation approach, in which we divide the dataset into k groups of approximately equal size; then we take one of the folds as a validation set and we fit the model on the remaining $k-1$ folds; we repeat the procedure on each fold and at the end the k-fold CV estimate is the average of these k test error/accuracy estimates.
In principle the LOOCV could give more correct results than k-fold CV because it uses almost all the information of the sample in each estimation, and thus it gives us truly unbiased estimates of the test error/accuracy.
However, since this dataset is quite large (30.343 observations), using LOOCV can be too much expensive in terms of computation (we would have to estimate 30.343 models!); furthermore, LOOCV has a very higher variance than k-fold CV, because of the so-called bias-variance trade-off (that arises from the fact that we repeat several times almost identical models), and this considerably dampens the advantage of using this method. Thus, for these reasons we choose not to use LOOCV approach.
Since we decide to use only k-fold CV, it remains to choose the number of folds in which we divide the dataset. In the literature, the values that typically are used are $k = 5$ and $k = 10$, since they have a reasonably low level of bias with respect to the validation set approach but also a reasonably lower variance with respect to the LOOCV.
Since this dataset is very large and the estimators obtained using the Frank and Hall (2001) approach are particularly onerous for R to calculate, we choose to use only $k = 5$ since this requires much less resources (and time) than $k = 10$.

6.2 Cross-Validation on the standard classifiers

First, we divide the entire dataset $supervised\_features2$ into $k = 5$ folds. Afterwards, we perform the 5-fold CV on the QDA estimator presented before, obtaining as a result the following values:

Table 4 QDA, standard approach

	Accuracy	(1-MAE%)	(1-MSE%)
QDA	0.1021570	0.9605275	0.7698306

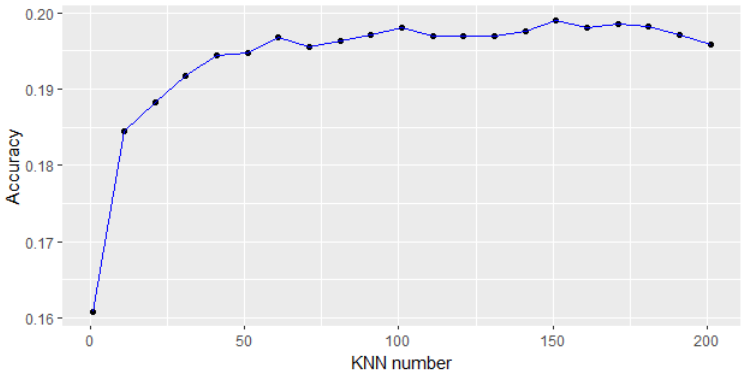
Here the results on the part of accuracy are clearly very low: only 10.2% of the observations are correctly predicted by the QDA model, while the (1-MAE%) and

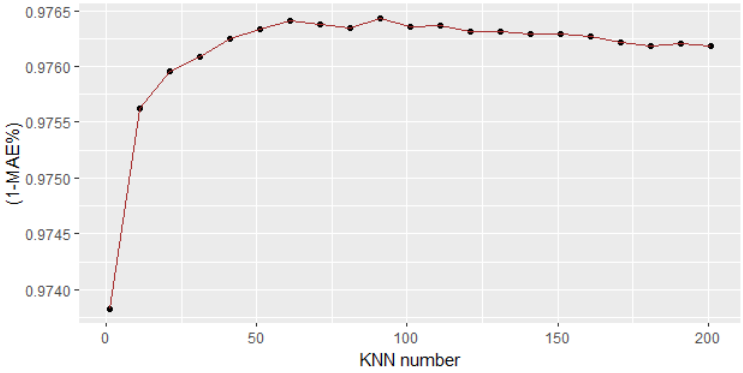
(1-MSE%) values seem to be very high in comparison.
This can be explained by the fact that we have a large number of income classes (10) here to which we must assign each observation.
The "Accuracy" measures how many of the observations are correctly predicted, regardless of the "distance" between the actual and the predicted income class. Instead, the "MAE" and "MSE" do measure this "distance". Thus, since the probability of choosing the classes just nearby the true one (say 9 or 7 instead of 8) is very high, it is perfectly normal that the accuracy of the predictor is relatively small, while the (1-MAE%) and (1-MSE%) are higher.

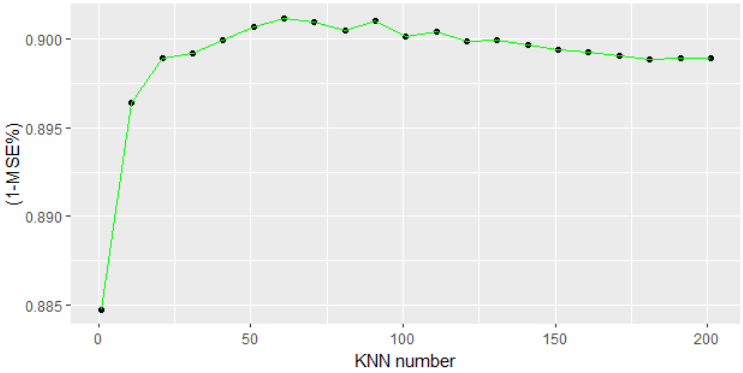
Afterwards, we can proceed by performing the cross-validation also on the KNN model. In this case we also have to choose the best level of k for our purposes, and the cross-validation can help us in this task.
The range of possible values of k is very huge, since it can vary from 1 to the total number of our observations (that in our case is 30.043).
Theoretically, it is not a good idea to use the larger values of k, because they tend to estimate a (biased) clear-cut linear decision boundary. Also, it is not a good idea to use the smallest values of k, because they typically produce overfitting. Thus, for these two effects combined the curves of the estimates of the test accuracy/reciprocal of test error exhibit a typical inverse-U-shape.
We can see if this is the case by simulating the model on the values of k going from 1 to 201. Since computationally it is a very expensive process, we begin by considering only some values in this interval: We can also see better the results by using plots:

Table 5

k	Accuracy	(1-MAE%)	(1-MSE%)
1	0.1608610	0.9738302	0.8847460
11	0.1845237	0.9756263	0.8964034
21	0.1883139	0.9759552	0.8989438
31	0.1916752	0.9760871	0.8992384
41	0.1943776	0.9762472	0.8999680
51	0.1947732	0.9763356	0.9007108
61	0.1968165	0.9764100	0.9011972
71	0.1955311	0.9763810	0.9009548
81	0.1962561	0.9763491	0.9004930
91	0.1971460	0.9764315	0.9010309
101	0.1980687	0.9763507	0.9001955
111	0.1969812	0.9763669	0.9004515
121	0.1969152	0.9763105	0.8999206
131	0.1970141	0.9763089	0.8999308
141	0.1976074	0.9762851	0.8996929
151	0.1990245	0.9762911	0.8994522
161	0.1981018	0.9762654	0.8992697
171	0.1985962	0.9762179	0.8990581
181	0.1982007	0.9761797	0.8988973
191	0.1970471	0.9762014	0.8989559
201	0.1958937	0.9761846	0.8989378


Accuracy
0.20
0.19
0.18
0.17
0.16
0
50
100
150
200
KNN number


(1-MAE%)
0.9765
0.9760
0.9755
0.9750
0.9745
0.9740
0
50
100
150
200
KNN number

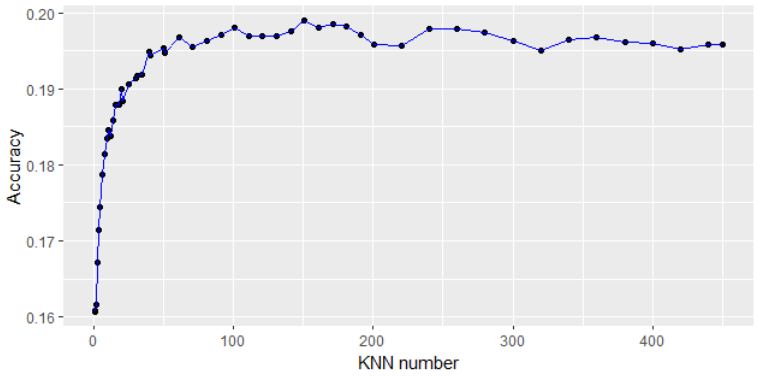

(1-MSE%)
0.900
0.895
0.890
0.885
0
50
100
150
200
KNN number

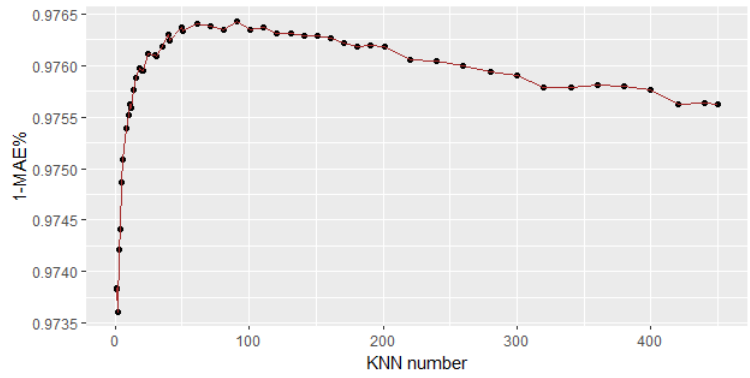
From the plots we clearly see that while the accuracy of the KNN estimate grows monotonically when we increase k (but with decreasing returns) until k=180, then decreases very slightly, the other measures (1-MAE% and 1-MSE%) start low at k=1, reach a peak at a k=60 and then are very slightly decreasing afterwards (a decline more pronounced for the 1-MSE%).
We can also note from the table that all these values (particularly the accuracy) are much greater than those of the QDA approach.
Nevertheless, we are not entirely satisfied with these results. Firstly, the accuracy is growing in this interval, and its decline at k=180 is relatively negligible: can we expect it to grow more when we increase k? To answer this question, we can consider greater values of k (until k=450, afterwards R is not able to compute the estimate), and do the cross-validated estimate of accuracy, 1-MAE% and 1-MSE%. The results are in the table below: As we can see from the tables above, the accuracy now is increasing

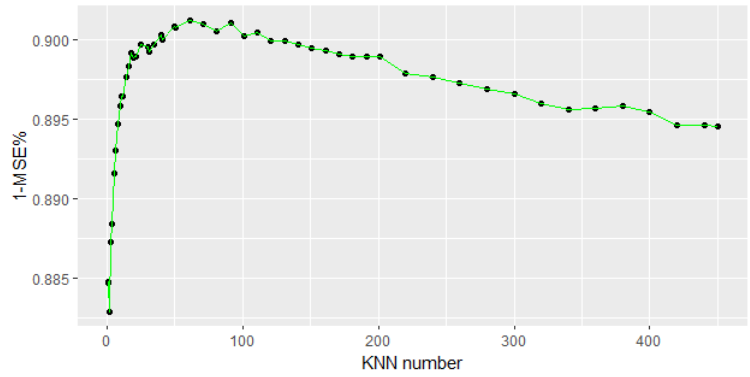
Table 6 KNN tuning, very large values

k	Accuracy	(1-MAE%)	(1-MSE%)
220	0.1956958	0.9760558	0.8978947
240	0.1978381	0.9760422	0.8976181
260	0.1979698	0.9760037	0.8972639
280	0.1973766	0.9759374	0.8968832
300	0.1963879	0.9758999	0.8965833
320	0.1950367	0.9757905	0.8959703
340	0.1964538	0.9757878	0.8956190
360	0.1967836	0.9758079	0.8957096
380	0.1961903	0.9758007	0.8957999
400	0.1960255	0.9757591	0.8954288
420	0.1952675	0.9756303	0.8946118
440	0.1957620	0.9756395	0.8946178
450	0.1957950	0.9756244	0.8945130

until $k = 260$, then it decreases a bit and afterwards it oscillates with low variations; this is an indication that around $k = 260$ we have the peak of this curve.
Secondly, it may be also interesting to look in details at the behaviour of these curves when k is small, because in this area we have a greater variability of the results (very high between $k = 1$ and $k = 11$, for example). We can merge these results and those seen before in a plot in order to see the overall dynamics of the data before commenting:







The results are in the table below:

Table 7

	Accuracy	(1-MAE%)	(1-MSE%)
1	0.1607291	0.9738368	0.8847757
2	0.1616848	0.9736107	0.8829387
3	0.1672216	0.9742125	0.8873239
4	0.1714730	0.9744139	0.8884375
5	0.1744064	0.9748660	0.8916285
6	0.1787236	0.9750836	0.8930034
8	0.1814918	0.9753930	0.8947174
10	0.1835021	0.9755212	0.8958454
12	0.1837985	0.9755911	0.8964103
14	0.1858091	0.9757641	0.8976183
16	0.1879185	0.9758781	0.8983394
18	0.1878854	0.9759698	0.8991639
20	0.1900276	0.9759543	0.8988466
25	0.1906207	0.9761177	0.8996929
30	0.1913457	0.9761016	0.8995468
35	0.1918730	0.9761830	0.8997054
40	0.1948392	0.9763023	0.9003118
50	0.1953335	0.9763695	0.9008417

The new observations here are perfectly in line with the behaviour of the more rough curves above: we have no "surprises" in the very low-k part of the curves, while in the high-k part the decrease of 1-MAE% and 1-MSE% continues in a more significant way; instead, the slight decline of Accuracy seen at $k = 180$ disappears. The shapes of the curves are here defined pretty well, with a sharp increase of 1-MAE% and 1-MSE% until $k \cong 50$, followed by a flatter decrease afterwards, while for Accuracy we have a well-shaped non-decreasing concave function, with a substantial plateau (with some oscillations) after $k \cong 150$.
In order to choose the best value of k, as we did previously, we can look at the sum of the three values of Accuracy, 1-MAE% and 1-MSE%.
Here, by using the "which.max" function in R, we obtain that the maximum of the sum of these values is obtained when $k = 151$, which we thus must regard as the benchmark value for KNN approach.

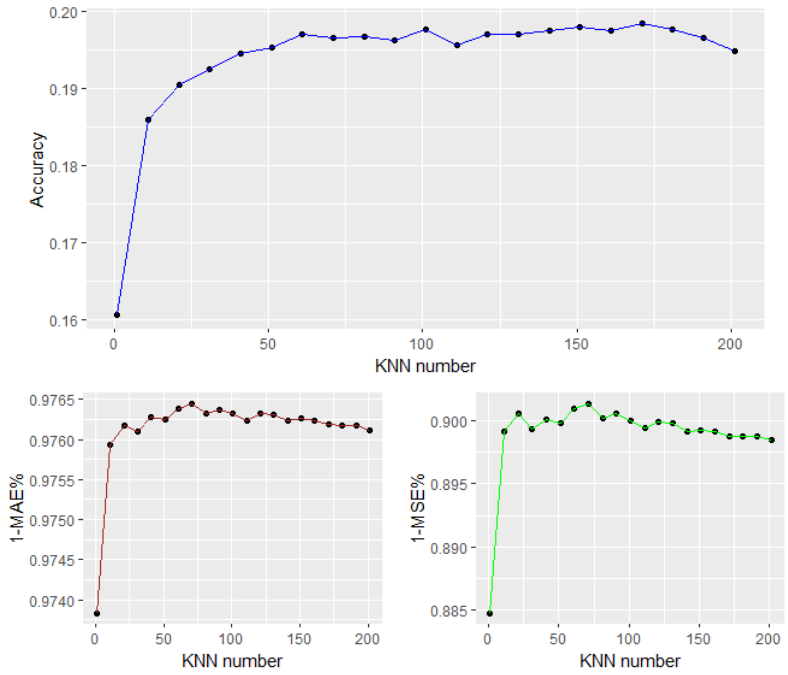
6.3 Cross-Validation on the Frank and Hall (2001) approach classifiers

Now we can proceed by estimating with 5-fold cross-validation the classifiers obtained using the Frank and Hall (2001) ordinal approach.
In this case, since we take into consideration the ordinal structure of the income classes, in addition to QDA and KNN as we have seen before we also can use the Logistic estimator.
We proceed, using the same folds as before in order to have a more precise comparison, by estimating first the values of accuracy, 1-MAE% and 1-MSE% of the QDA and Logistic estimators, which unlike KNN do not require the tuning of a parameter. The results are in the table below: Here it is clear from the table that the Logistic estimator performs better than the QDA in all metrics considered, but the QDA here does give

Table 8 QDA and Logistic, Frank (2001)

	Accuracy	(1-MAE%)	(1-MSE%)
QDA_Ord	0.1748672	0.9719164	0.8644483
Logistic_Ord	0.1888079	0.9749563	0.8902296

us considerably better results with respect to the standard approach.
The results of this table must then be compared to the ones of the KNN estimator, for which we have the same issue of "tuning" as seen before for the standard approach. So, like before, we begin by tuning the parameter k into a range of values going through 1 to 201. Again, we take into consideration only a small amount of values in this interval, also because this operation is computationally very expensive and doing a full-value tuning in this context can require a huge amount of time and power.
The results are presented in the following table and the plots:



From the table and the plot above we can clearly see that the results are not too different with respect to the previous approach KNN estimates. The shape of the curve is quite the same as before: accuracy is also here increasing with decreasing returns, with a very slightly decreasing region after $k = 180$, while 1-MAE% and 1-MSE% are very low at very small values of k, then they increase very sharply until they reach a peak, and eventually decreasing in a much flatter way afterwards.
To see the behaviour of the curve at greater values of k we do the same thing as before,

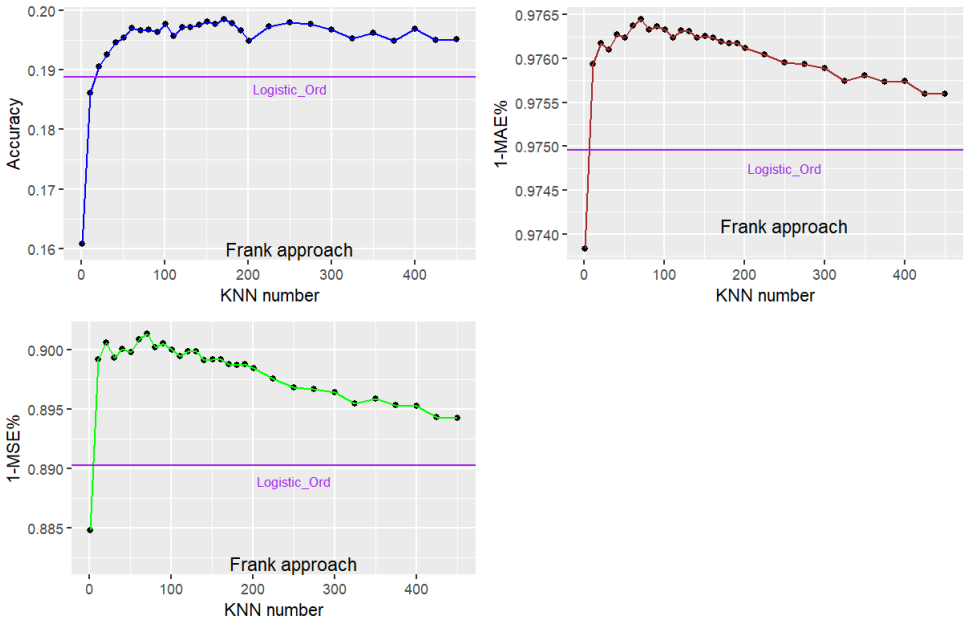
Table 9

	Accuracy	(1-MAE%)	(1-MSE%)
1	0.1607621	0.9738325	0.8847530
11	0.1860395	0.9759332	0.8991668
21	0.1905220	0.9761767	0.9006067
31	0.1925321	0.9760996	0.8993194
41	0.1945755	0.9762703	0.9000452
51	0.1953334	0.9762403	0.8997857
61	0.1970471	0.9763787	0.9009089
71	0.1965528	0.9764499	0.9013756
81	0.1967504	0.9763293	0.9002168
91	0.1962891	0.9763613	0.9005540
101	0.1976403	0.9763300	0.9000040
111	0.1956629	0.9762367	0.8994414
121	0.1971130	0.9763174	0.8998965
131	0.1970800	0.9763066	0.8998448
141	0.1975744	0.9762334	0.8991369
151	0.1980358	0.9762561	0.8991958
161	0.1976075	0.9762331	0.8991807
171	0.1984973	0.9761912	0.8987644
181	0.1977393	0.9761714	0.8987124
191	0.1966185	0.9761751	0.8987687
201	0.1948720	0.9761148	0.8984520

considering only few values below $k \cong 450$ as representatives of this behaviour. Instead, unlike the previous subsection, we do not consider the smaller values of k, mainly because this area is not quite interesting for our purposes and these computations require too much time to be performed.
The results of these estimates are in the tables and plots below:

Table 10 KNN tuning, very large values

k	Accuracy	(1-MAE%)	(1-MSE%)
225	0.1972449	0.9760459	0.8975407
250	0.1979370	0.9759526	0.8968549
275	0.1976731	0.9759322	0.8966749
300	0.1966845	0.9758877	0.8963872
325	0.1952015	0.9757413	0.8954743
350	0.1961902	0.9758079	0.8958520
375	0.1947732	0.9757321	0.8953576
400	0.1968825	0.9757450	0.8952585
425	0.1949051	0.9755993	0.8943528
450	0.1951687	0.9755940	0.8942777



The behaviour of the curve is not much changed with respect to the estimates of the smaller values done before. In particular, we have that the higher values of k forms a plateau region (with some variability) in the Accuracy plot and further decreasing in a more significant way in the 1-MAE% and 1-MSE% plots

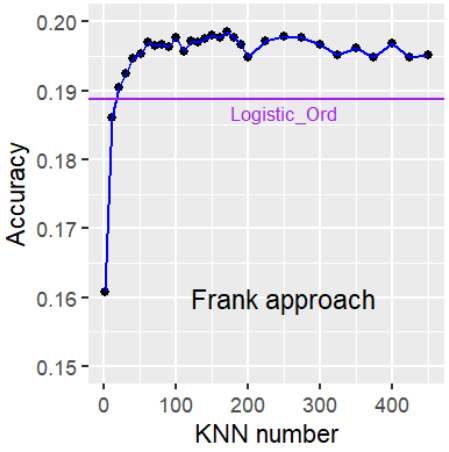
As in the previous subsection, in order to choose the best value of k we look at the sum of the three values of Accuracy, 1-MAE% and 1-MSE%; thus, by using the "which.max" function in R, we obtain as maximum of this sum the value of $k = 71$.

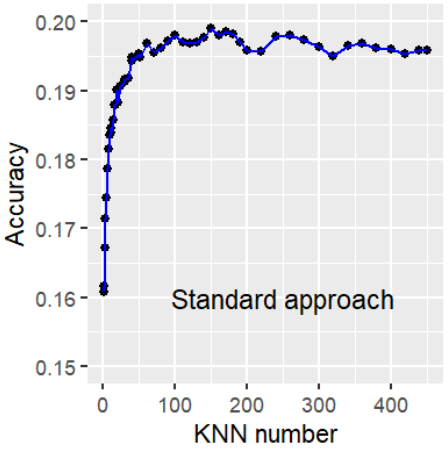
6.4 Overall results and conclusion

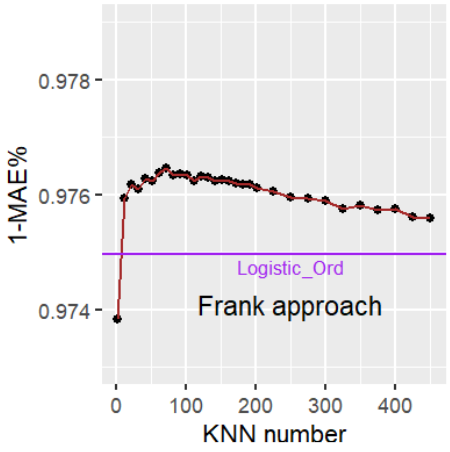
In order to compare the two approaches and to see what are the best estimators to use, we can see below some plots that sum up all the measures that we saw before, except from QDA which already from the tables is clearly suboptimal in both approaches.
From these plots we can see that the KNN estimator as $k = 1$ is less precise than the Logistic estimator, but then when the overfitting effect is ruled out and k increases the KNN becomes consistently better than the Logistic.
Regarding the comparison between the two approaches, while the QDA estimator is more accurate in the Frank and Hall (2001) approach than in the standard one ($accuracyQDA_{frank} = 0.174867$, $accuracyQDA_{standard} = 0.102157$), the values of KNN estimates are on average substantially the same in both approaches. However, the value of k resulted from the maximization of the sum of the three measures is different ($k = 71$), but with not so different values of the three estimates.
In conclusion, on the basis of these estimations we can conclude that both standard and Frank and Hall (2001) approaches have substantially the same predictive power in this context, while regarding the choice of the estimator it is clearly the best option to use the KNN approach with a reasonably high value of k (between 50 and 160), rather than the Logistic or QDA.
It is also worth noting that using cross validation we recovered the theoretical shape of the error functions for the KNN tuning, that was lost under the validation set approach.
To sum up everything, we present below the final table of results obtained by cross-validation:

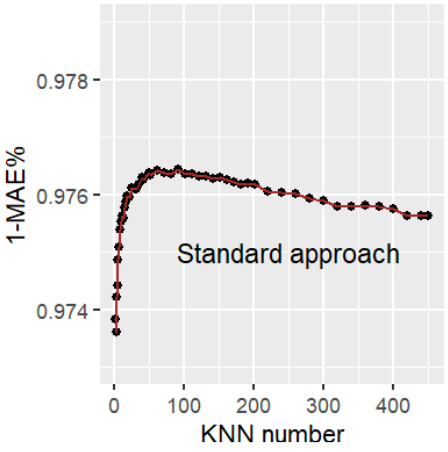
Table 11 Overall results

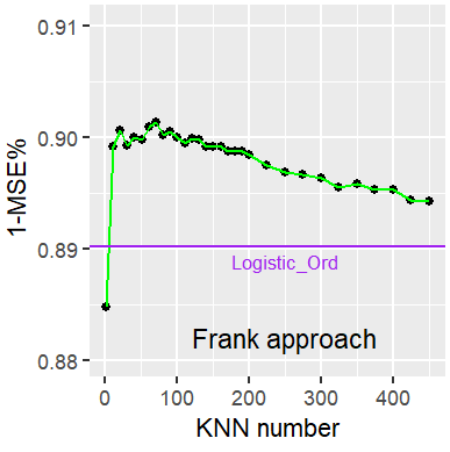
Classifier	Accuracy	(1-MAE%)	(1-MSE%)
QDA_norm	0.1021570	0.9605275	0.7698306
KNN_norm, k=151	0.1990245	0.9762911	0.8994522
QDA_Ord	0.1748672	0.9719164	0.8644483
Logistic_Ord	0.1888079	0.9749563	0.8902296
KNN_ord, k=71	0.1965528	0.9764499	0.9013756

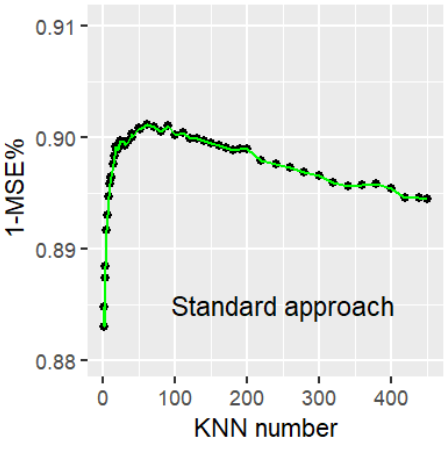

Accuracy
0.20
0.19
0.18
0.17
0.16
0.15
Logistic_Ord
Frank approach
0
100
200
300
400
KNN number


Accuracy
0.20
0.19
0.18
0.17
0.16
0.15
Standard approach
0
100
200
300
400
KNN number


1-MAE%
0.978
0.976
0.974
Logistic_Ord
Frank approach
0
100
200
300
400
KNN number


1-MAE%
0.978
0.976
0.974
Standard approach
0
100
200
300
400
KNN number


1-MSE%
0.91
0.90
0.89
0.88
Logistic_Ord
Frank approach
0
100
200
300
400
KNN number


1-MSE%
0.91
0.90
0.89
0.88
Standard approach
0
100
200
300
400
KNN number

References

Frank, E., Hall, M. (2001). A Simple Approach to Ordinal Classification. In: De Raedt, L., Flach, P. (eds) Machine Learning: ECML 2001. ECML 2001. Lecture Notes in Computer Science(), vol 2167. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-44795-4_13

Cardoso, Jaime Sousa, Ricardo. (2011). Measuring the Performance of Ordinal Classification.. IJPRAI. 25. 1173-1195. 10.1142/S0218001411009093.

Carrizosa, E., Restrepo, M. G., & Morales, D. R. (2021). On clustering categories of categorical predictors in generalized linear models. *Expert Systems with Applications*, 182, 115245.

Gower, J. C. (1971). A General Coefficient of Similarity and Some of Its Properties. *Biometrics*, 27(4), 857–871.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning. New York: Springer.

Preud'Homme, G., Duarte, K., Dalleau, K., Lacomblez, C., Bresso, E., Smaïl-Tabbone, M., ... & Girerd, N. (2021). Head-to-head comparison of clustering methods for heterogeneous data: a simulation-driven benchmark. *Scientific reports*, 11(1), 1-14.

Van der Maaten, L., Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(11).