Argomenti di discussione

(Discussion topics)

Data mining application issues in income indicators audits

M. Barone, S. Pisani, A. Spingola

N. 02/2017

Agenzia Entrate

Argomenti di discussione è una pubblicazione che intende divulgare contributi ed analisi su argomenti di economia, statistica, econometria e scienza delle finanze, che abbiano rilevanza per la missione dell’Agenzia delle Entrate, al fine di alimentare il dibattito scientifico sui temi di interesse strategico dell’Agenzia e favorire lo scambio di opinioni. La pubblicazione ospita contributi in lingua italiana o inglese proposti da autori sia interni sia esterni all’Agenzia. Le opinioni espresse negli articoli sono lasciate alla discrezionalità degli autori e non impegnano la responsabilità dell'Agenzia delle Entrate.

Discussion topics *is a publication with the purpose of disseminating contributions and analyses in economics, statistics, econometrics and public finance, which are relevant to the mission of the Italian Revenue Agency (Agenzia delle Entrate) in order to bolster the scientific debate on topics of strategic interest for the Agency and to facilitate the exchange of opinions. This publication accepts contributions in Italian or in English proposed by authors who are internal or external to the Agency. The views expressed in the articles are those of the authors and do not involve the responsibility of the Revenue Agency.*

ISSN: 2420-773X

Tel. +39-06-5054-5037/3874 Fax +39-06-5056-2612
E-mail: ae.argomentididiscussione@agenziaentrate.it
Web: http://www.agenziaentrate.gov.it/wps/content/Nsilib/Nsi/Agenzia/Agenzia+comunica/Prodotti+editoriali/Rivista/Argomenti+di+discussione/

Data mining application issues in income indicators audits[1]

Mauro Barone, Stefano Pisani, Andrea Spingola

Agenzia delle Entrate, mauro.barone@agenziaentrate.it

Abstract

This paper provides a methodology to design an effective learning scheme intended to improve the Italian Revenue Agency's ability to identify non-compliant taxpayers in the context of income indicators audits (i.e. redditometro [income meter]). This methodology follows some well-defined steps, including data selection and its preliminary preprocessing, data mining model building, validating and testing, and successful model incorporation into standard used applications, subject to periodic review. Our analysis shows that data mining techniques could actually enhance fiscal audit quality, by increasing both the expected audit positivity rate and the expected average tax claim. The proposed methodology is currently being validated on real cases: a number of taxpayers have been selected according to classification criteria developed in this paper, and actual audits will be performed in order to assess their predictive accuracy. At the writing of this paper, no results are yet available.

Sommario

Il presente articolo propone una metodologia di implementazione di un modello predittivo utilizzabile per definire un profilo di rischio che consenta di selezionare i contribuenti da sottoporre ad accertamento. Il modello teorico è stato applicato alla tipologia di accertamento "sintetico" basato sulla verifica della coerenza tra la capacità di spesa e il reddito dichiarato dal contribuente. La scelta del campo di applicazione è stata dettata dal fatto che l'efficacia dello strumento di accertamento considerato ha presentato delle specifiche criticità in passato. Si intende verificare se tale criticità siano, almeno in parte, riconducibili al processo di selezione dei contribuenti.

Detta metodologia si sviluppa in fasi standardizzate che includono la selezione dei dati ed il loro *pre-processing*, la costruzione, validazione e test di uno o più modelli di *data mining* ed infine l'internalizzazione dei modelli maggiormente performanti nell'ambito dell'infrastruttura informatica già utilizzata dall'Agenzia delle Entrate, soggetti a revisione periodica. La nostra analisi mostra che l'impiego di tecniche di *data mining* potrebbe effettivamente migliorare la qualità dell'attività di controllo posta in essere, incrementandone sia il tasso di positività che i maggiori imponibili accertati. La metodologia proposta è attualmente impiegata in fase sperimentale su casi reali: un certo numero di contribuenti è stato selezionato secondo i criteri mostrati nel presente articolo. L'esito dei controlli che ne seguiranno fornirà una stima della capacità predittiva dei modelli impiegati. Al momento in cui il presente articolo viene scritto, non sono ancora noti i risultati di tale sperimentazione.

[1] Data analyses were performed using WEKA, the data mining workbench developed at Waikato University in Hamilton, New Zealand, released under GNU GPL license .

JEL classification: C18, C38, H26

Keywords: Data mining application, decision trees, ensemble learning, tax fraud detection

Index

1. Introduction

Fraud detection systems are designed to automate and help reduce the manual parts of a screening / checking process. Data mining plays an important role in fraud detection as it is often applied to extract fraudulent behavior profiles hidden behind large quantities of data, and, thus, may serve in decision support systems for planning effective audit strategies. Indeed, huge amounts of resources may be recovered from well–targeted audits. This explains the increasing interest and investments of both governments and fiscal agencies in intelligent systems for audit planning. The Italian Revenue Agency itself has been studying data mining application techniques in order to detect tax evasion, focusing, for instance, on the tax credit system, supporting investments in disadvantaged areas, as in [1], or on VAT frauds related to credit mechanism, as in [2].

This paper presents a case study focusing on another kind of audit, known as “redditometro”, or *income meter*.

Income meter, as a type of fiscal audit, is provided by art. 38, paragraphs 4 and following of Presidential Decree 600/73, which allows tax authorities to assess a taxpayer's total income according to his spending power. Briefly, given the possession or availability of a certain amount of goods and services, the corresponding income is computed by usage of appropriate coefficients. Individual incomes associated to these goods and services are added together and summed up to form the *total synthetic income* or *estimated income*. If, for a given taxpayer, *estimated income* is greater than *declared income* (plus exempt income and other income subject to definitive withholding) for more than 20%, it is up to that taxpayer to explain how he could spend, in a year, more than he had earned. If he doesn’t, then tax authorities are entitled to send him a *tax assessment notice,* a formal written act through which tax administration assesses a higher taxable income with respect to that declared. Before sending a *tax assessment notice* to a taxpayer who satisfies the above mentioned risk condition, however, tax authorities must invite him to explain how he could afford all goods and services he had bought, given his relative low income.

However, not all the invitations sent to taxpayers by the Italian Revenue Agency in the years 2014-2015 ended up in *tax assessment notices*, that is, for only a small part of the invited people, a higher income tax burden was actually determined. That is mainly because many taxpayers managed to show either they had other income not taken into account by the tax administration, or that the expenses they were charged were actually incurred, fully or partially, by other people.

The taxpayer selection process efficiency may be evaluated by dividing the number of *tax assessment notice*s by the number of invitations sent. This ratio represent the outcome that should be improved by Italian Revenue Agency.

For this purpose, it becomes of key relevance to design a predictive analysis tool, such as a classification scheme able to increase the invitation positivity rate, given the regulatory environment. In other words, such a tool should be able to recognize, among all taxpayers showing a difference between estimated and declared income higher than 20%, those with a risk profile that maximizes the probability of a positive *tax assessment notice* outcome.

Basically, taking into account a set of taxpayer characteristics, such as demographic profile and expenditures incurred, such a tool should be able to find *recurrent patterns* in data that may help tax offices perform their screening tasks more efficiently, taking advantage of data mining techniques.

In essence, data mining techniques would be useful in that they would allow tax auditors to learn from the past, enhancing the positive aspects and discarding the negative signals. Indeed, such techniques, by training on a given dataset of records concerning the past audit activity, try to discover illegal taxpayer profiles that, until now, have not been identified by traditional techniques.

In the context we are interested in, the very practical goal data mining techniques would like to achieve is, therefore, that of isolating, from among taxpayers that “square off” to the *income meter*, those who "more likely" are actual fraudsters, establishing a relationship between the positive outcome of already carried out audits and the predictive variables contained in the dataset on hand.

Of course, result reliability depends on the training sample’s representativeness, on overall data quality and on its inherent discriminatory nature. We need to point out that auditing is the only way to produce a dataset. Since tax auditors focus only on taxpayers thought to be particularly suspicious according to some clues, data on hand may be biased and may not represent the whole set of Italian taxpayers, being the result of a selection process that may have systematically excluded some groups of taxpayers.

Summing up, this paper, however, provides a methodology to design a suitable and effective learning scheme intended to improve the Italian Revenue Agency’s ability to identify non-compliant taxpayers in the context of income indicators audits. Hence, after presenting some previous studies focusing on tax fraud detection by means of data mining techniques, this paper’s structure reflects some well-defined steps indicated in [3], typical of any classification project, that can be summarized as follows:

Defining inputs, outputs, and evaluation metrics: identifying inputs includes data selection. This is mainly done by retrieving data from VERDI application, part of the technological workbench used by the Italian Revenue Agency in its daily operations, specifically used in the context of income indicators audits[2]. Outputs typically include a ranked list of taxpayers suspected of having a particular compliance issue. Evaluation metrics are defined to measure the success of the project—for example, measuring audit positivity rates and their average recovery.

Obtaining, exploring, and preprocessing data: this phase typically involves descriptive statistics in order to gain familiarity with data used for classification.

Building, validating, and testing models: validation consists in selecting optimal model parameters to assess fraudulent risk, while testing actually provides an estimate of the models’ accuracy.

Deploying risk-analysis models: successful models should be incorporated into standard use applications (such as the above mentioned VERDI), and be subjected to periodic review for re–evaluating accuracy and performance.

2. Related work

As pointed out in [4], generally, manual case selection, computer-based case selection (data mining based methods [5], [6]) and whistle-blowing-based selection are three frequently used methods of tax inspection. However, many researchers believe that data mining techniques used by tax administrations to detect tax fraud are the most promising approaches [6]. Mechanisms such as neural networks, decision trees [7], logistic regression, SOM (self-organizing maps), K-Means, support vector machines, Bayesian networks, K-nearest neighbor and many others have been used to check tax evasion.

[2] VERDI is the acronym of **VE**erifica **R**eddito **DI**chiarato, that is, Declared Income Audit. As a matter of fact, it recalls the famous 19th-century italian opera composer, Giuseppe Verdi. This application allows tax offices to manage income indicators audits, by providing a complete editable dashboard, supported by a large database, on which our analysis is based.

Moreover, many researchers examine data mining approaches as effectively adopted in their countries: the Moroccan case is described in [8], the Chinese case in [5]; the Greek case in [7], the Taiwanese case in [9].

Thus, there is great interest in the possible applications of data mining techniques to the context of tax fraud detection, and Italy is no exception.

3. Defining inputs

3.1 The dataset: an overview

The analyzed dataset gathers information both from the VERDI application and from the database referred to invitations and *tax assessment notices*.

Data on hand refer to taxpayers invited by tax offices over the years 2014 and 2015. So, data reports the amounts of their expenses referred to 2009 and 2010[3] (see Appendix 1 for details on the kind of expenses that were taken into consideration). Furthermore, administrative details about sent invitations and subsequent *tax assessment notices* are available, plus some other personal details, such as declared income (the taxpayer's and his family's), reconstructed income based on incurred expenses (also called *selection value*), family type, geographical residence area and possession (or absence) of a VAT number.

This dataset includes **39,757** records, which form the initial database for data mining analyses.

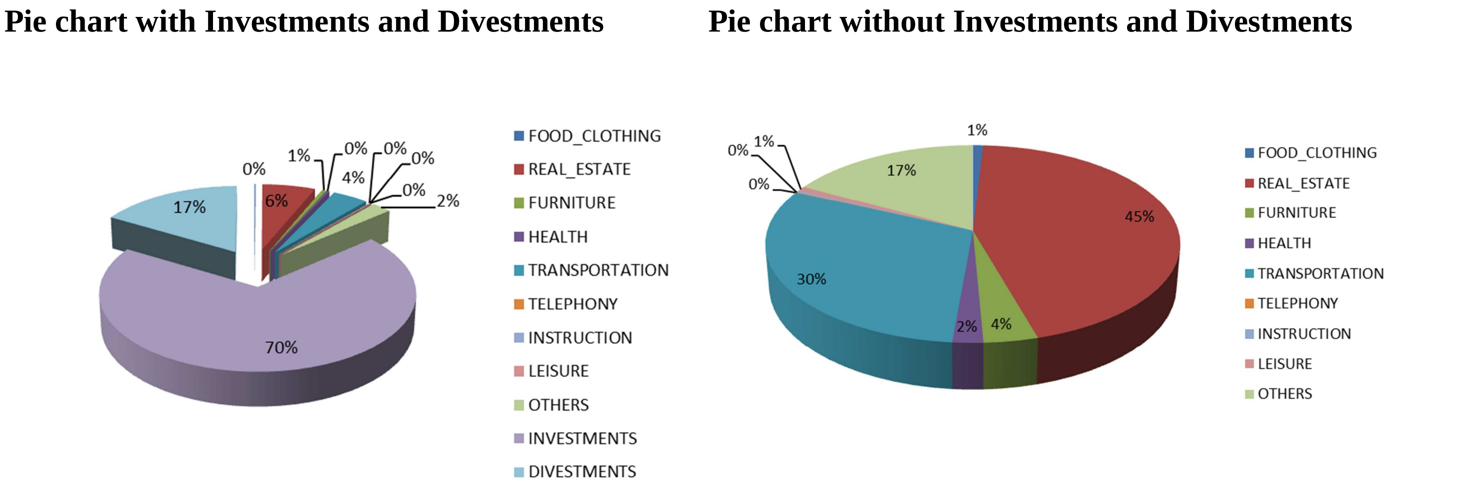
To better understand the nature of the data, some preliminary considerations are listed below.

First, we show how taxpayers' expenses are distributed, including investments and divestments, as follows:

[3] Some expenses data is only available starting from year 2010. In these cases, for year 2009, the corresponding value will not be zero, but null, which means information is not available.

Figure 3.1 – Taxpayers' expenditure types

TYPE	Amount
FOOD_CLOTHING	€ 11,361,682
REAL_ESTATE	€ 555,648,707
FURNITURE	€ 46,523,838
HEALTH	€ 26,266,655
TRANSPORTATION	€ 378,633,661
TELEPHONY	€ 1,380,152
INSTRUCTION	€ 3,906,078
LEISURE	€ 9,639,412
OTHERS	€ 211,529,852
INVESTMENTS	€ 6,286,457,635
DIVESTMENTS	€ 1,496,464,196



Pie chart with Investments and Divestments

Pie chart without Investments and Divestments

Figure 3.1 clearly shows that some expenses are more substantial than others, such as investments, building expenses, transportation expenses and expenses for "other" goods and services. Indeed, data related to many of these expenses is gathered by the Revenue Service in a very accurate way: for instance, data concerning investments is available, to a large extent, from the Land registry, as well as mortgages details; moreover, utilities companies periodically transmit streams of data to the Revenue Service (so that information on water, electricity and gas consumption is very precise) and a direct connection to the Department of Motor Vehicles databases allows for precise estimates on expenses related to cars, motorcycles, and so on. Moreover, deductible expenses from taxable income are drawn directly from submitted tax returns.

Lastly, we recall that Decree Law n. 78/2010 introduced an obligation, for all people or companies liable to VAT registration, to transmit to the Revenue Service a list containing details about their purchase and sale operations worth not less than 3,600 euros (this kind of communication is known as "*Spesometro*"): so, the operation amounts and the customer or supplier name had to be reported in such lists. As a result, for 2010 only, expenses such as those concerning food and clothing, furniture and instruction have been collected simply by retrieving taxpayers names in VERDI database, as customers, from these lists. However, data concerning these latter expenditures is not as accurate as data concerning the afore mentioned ones: indeed, some lists may have not been sent at all or may contain some errors, all the expenses lower than 3,600 euros have not been taken into account[4]. For all of these reasons, it turns out that these latter expenses account for a relative small part, if compared with the overall expenses incurred by invited taxpayers.

Some basic statistics referred to *Figure 3.1* expenses are shown in *Table 3.1*.

Table 3.1 – Statistics of taxpayers' expenses

	FOOD, CLOTHING	REAL ESTATE	FURNITURE	HEALTH	TRANSPORTATION	TELEPHONY	INSTRUCTION	LEISURE	OTHERS	INVESTMENTS	DIVESTMENTS
minimum	**€ 0**	**-€ 217**	**€ 0**	**€ 0**	**€ 0**	**€ 0**	**€ 0**	**€ 0**	**€ 0**	**€ 0**	**€ 0**
1 quart	€ 0	€ 1,249	€ 0	€ 0	€ 866	€ 0	€ 0	€ 0	€ 0	€ 0	€ 0
median	**€ 0**	**€ 3,659**	**€ 0**	**€ 0**	**€ 2,632**	**€ 0**	**€ 0**	**€ 0**	**€ 935**	**€ 14,502**	**€ 0**
3 quart	€ 0	€ 11,436	€ 0	€ 348	€ 5,613	€ 0	€ 0	€ 0	€ 5,028	€ 196,350	€ 0
80 perc	€ 0	€ 16,222	€ 0	€ 528	€ 7,022	€ 0	€ 0	€ 0	€ 6,104	€ 255,000	€ 3,900
90 perc	€ 0	€ 42,881	€ 0	€ 1,261	€ 16,059	€ 0	€ 0	€ 0	€ 11,467	€ 470,892	€ 104,402
95 perc	€ 0	€ 72,465	€ 5,126	€ 2,403	€ 40,538	€ 0	€ 316	€ 90	€ 20,407	€ 689,644	€ 200,000
99 perc	€ 0	€ 120,557	€ 21,052	€ 9,137	€ 138,230	€ 0	€ 2,255	€ 420	€ 67,384	€ 1,456,840	€ 588,254
max	**€ 1,250,000**	**€ 1,253,872**	**€ 3,052,880**	**€ 611,743**	**€ 1,388,757**	**€ 330,000**	**€ 200,994**	**€ 509,820**	**€ 2,884,125**	**€ 28,170,000**	**€ 15,090,000**
dev std	€ 15,833	€ 29,263	€ 18,652	€ 4923	€ 34,739	€ 2,777	€ 1,269	€ 6,123	€ 21,532	€ 379,732	€ 183,260
mode	€ 0	€ 0	€ 0	€ 0	€ 0	€ 0	€ 0	€ 0	€ 0	€ 0	€ 0
average	**€ 745**	**€ 13,976**	**€ 1,170**	**€ 661**	**€ 9,524**	**€ 35**	**€ 98**	**€ 242**	**€ 5,321**	**€ 158,122**	**€ 37,640**

Table 3.1 shows that for all the reported expenses, the most frequent value is zero, value often assumed by over half of the population. In addition, for each expense type, there must be a certain number of outliers, given the gap between the value of the 99th percentile and the maximum one.

[4] Consider also that expense types have been derived from the activity carried on by the sender, which could have led to errors or inaccuracies.

As a result, each expense frequency distribution looks highly skewed, with the median always below the average value (recall that a distribution's median is not affected by outliers, while its average value is).

Based on taxpayers' expenses, tax offices derive, for each of them, the selection value, i.e., the estimated income. Its frequency distribution is as follows:

Figure 3.2 – Selection value frequency distribution

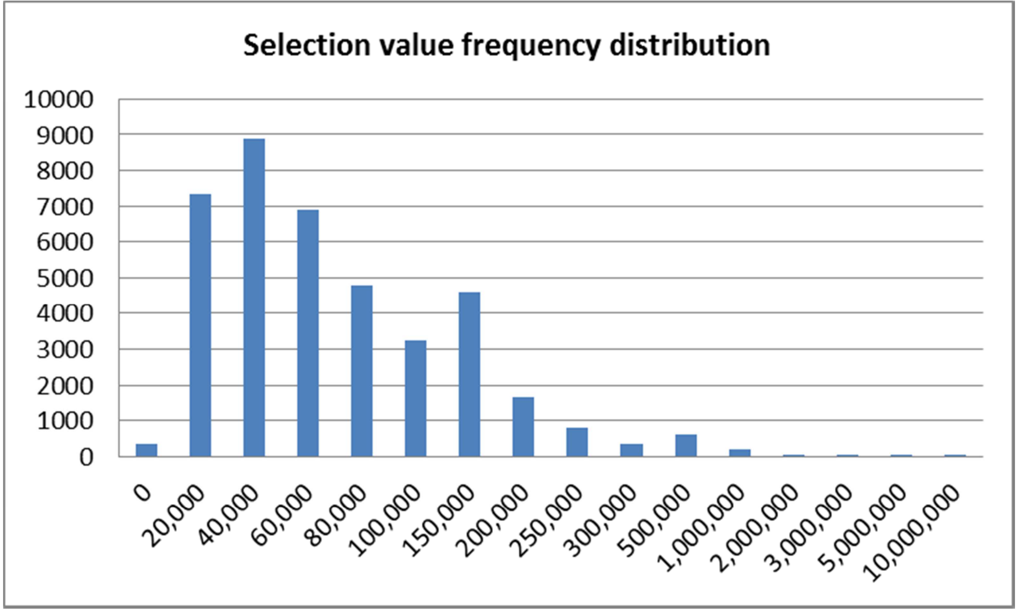


Figure 3.2 shows that nearly 80% of taxpayers have been estimated to have an income below € 100,000 and 3% above € 250,000; at the same time, 80% of taxpayers hold a selection value greater than € 20,000.

Furthermore, we investigated how many expenses types, on average, were needed to compute taxpayers' expected incomes. We found out that each taxpayer's overall consumption is highly concentrated in only a few items, as can be proved by calculating the Gini coefficient on their expense distributions[5]. Given the dominant role assumed by capital expenditures and divestments, we calculated two Gini indices, one including such items and one excluding them[6].

Gini coefficients for all taxpayers are shown in *Table 3.2*, in which data are divided into 16 intervals. We have calculated two indices: the first takes into account both investments and divestments (index called *GiniWith*); the second, instead, does not consider these items (index called *GiniWithout*). In both cases, we show the right end of each interval. So, for example, in the first interval of the index *GiniWith*, there are 5 taxpayers with a value of this index lower than

[5] Gini coefficient measures the inequality among values of a frequency distribution (for example, levels of income). A Gini coefficient of zero expresses perfect equality, where all values are the same (for example, where everyone has the same income). A Gini coefficient of 1 (or 100%) expresses maximal inequality among values (e.g., for a large number of people, where only one person has all the income or consumption, and all others have none, the Gini coefficient will be very nearly one).

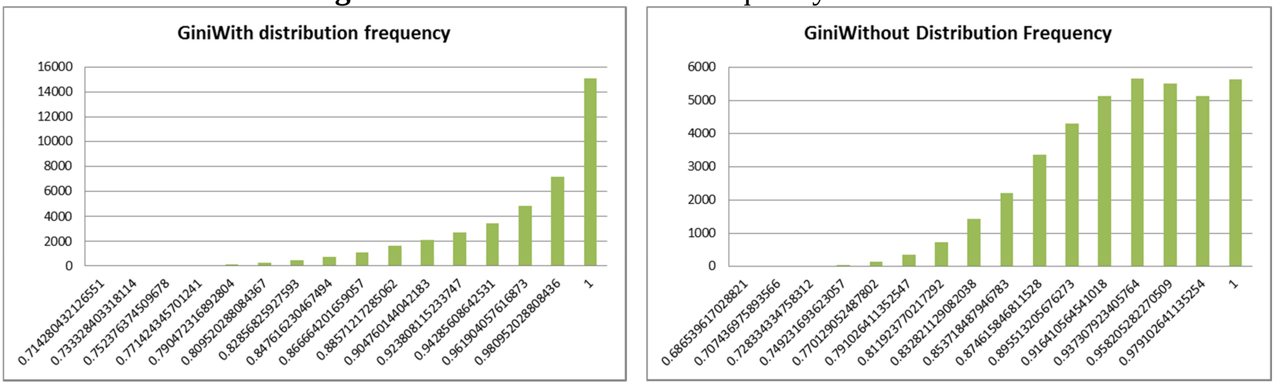
[6] The frequency distributions have been obtained by splitting data in k groups, where k is equal to the classic value: $1+\frac{10}{3}*\log_{10} n$, where n represents the observed sample cardinality, that is, 39,757. Thus k is equal to 16. The class interval, for each expense type, has been calculated on the ratio $\frac{Val_{max}-Val_{min}}{number of classes}$and the frequency diagram shows, on the x-axis, the right end of each interval (see Figure 3).

0.7142, and in the tenth interval of the index *GiniWithout*, there are 3,371 taxpayers with values between 0.8537 (right value of the previous interval) and 0.8746 (right value of the tenth interval).

Table 3.2 – Frequency values of Gini Indices

Classes	Gini with (right end interval)	Frequency	Gini without (right end interval)	Frequency
1	0.714280432126551	5	0.686539617028821	2
2	0.733328403318114	6	0.707436975893566	4
3	0.752376374509678	14	0.728334334758312	18
4	0.771424345701241	45	0.749231693623057	51
5	0.790472316892804	104	0.770129052487802	146
6	0.809520288084367	255	0.791026411352547	375
7	0.82856825927593	445	0.811923770217292	738
8	0.847616230467494	725	0.832821129082038	1,447
9	0.866664201659057	1,137	0.853718487946783	2,211
10	0.88571217285062	1,669	0.874615846811528	3,371
11	0.904760144042183	2,123	0.895513205676273	4,318
12	0.923808115233747	2,714	0.916410564541018	5,136
13	0.94285608642531	3,457	0.937307923405764	5,654
14	0.961904057616873	4,834	0.958205282270509	5,518
15	0.980952028808436	7,146	0.979102641135254	5,134
16	1	15,078	1	5,634
	Total	**39,757**		**39,757**

Figure 3.3 – Gini coefficients frequency distributions



The effect of investments and divestments is clear from *Figure 3*: *GiniWith* coefficients' values are higher, on average, than *GiniWithout* ones. Furthermore, in the first case, about 15,000 taxpayers are placed in the last class while in the latter, the last five classes are substantially equivalent.

Table 3.2 shows that in both cases, over 30,000 taxpayers have a Gini coefficient greater than 0.9. Thus, many taxpayers in the dataset had only a few anomalous spending elements which encouraged tax authorities to invite them.

So far, we've only focused on taxpayers' expenditures. To assess the audit's expected profitability, however, we should also consider taxpayer's income, as follows:

Figure 3.4 – Income frequency distribution

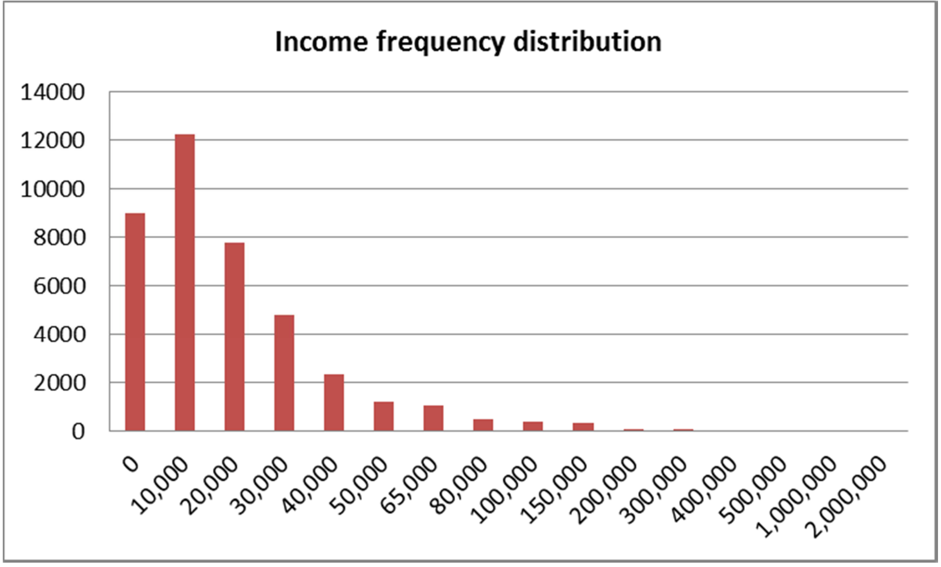


Figure 3.4 shows that more than 70% of taxpayers declared an income less than € 20,000.

Before proceeding through tax claim and class values, we would like to make some concluding remarks, highlighting that, of course, result reliability depends on the training sample's representativeness, on overall data quality and on its inherent discriminatory nature. As already pointed out, auditing is the only way to produce a dataset. Since tax auditors focus only on taxpayers thought to be particularly suspicious according to some clues, data on hand may be biased and may not represent the whole set of Italian taxpayers, being the result of a selection process that may have systematically excluded some groups of taxpayers. For instance, taxpayers might have been chosen mainly according to the difference between their expenses amount and declared income or to the score value (see section 7 for more details on these issues), so that taxpayers showing low score values or small differences between their income and expenses might have been systematically excluded from audits. Nonetheless, these taxpayers could have shown interesting features or patterns that data mining models will not be able to detect, simply because they are not present in the dataset. That is why learning models should also be fed with taxpayers selected without using the rules they produce.

Finally, we point out that, among the expenditure items, paid taxes were absent; nonetheless, they represent a substantial and widespread expense component among families.

3.2 Defining tax claim and class values

For each taxpayer in the dataset, his *tax assessment notice* status is known, as well as his additional due tax (i.e. the additional requested tax amount) and his additional settled tax (i.e. the additional tax requested after a tax settlement agreement). Depending on the status value, *tax claim*, which is defined as the tax debt owed by a taxpayer, will assume values proportional to the higher assessed or settled taxes or to a combination of them, or will be equal to zero.

We have therefore the following possible *tax claim* values:

€ zero: when the tax office did not notify a *tax assessment notice* or when the taxpayer (or a tax court) demonstrated that the *tax claim* was unfounded.

"HAT_TOT": basically, *higher assessed tax* is considered as *tax claim* when the assessed amounts were either paid by the taxpayer or given to the tax collection agent. Besides these cases, where the tax claim value is definitely correct, there are some other situations in which higher assessed taxes may not correspond to the actual tax claim (for instance, when it is only known that the *tax assessment notice* has been notified) or it could also prove to be incorrect (for example when it is known that, at the moment data was gathered, there was an ongoing settlement agreement: if it ended up positively, the correct tax claim value would then be the settled due amount, not the original assessed one). Nonetheless, in all these dubious situations, tax claim is estimated according to higher assessed tax, this being the most neutral choice. It is important to point out that tax claims do not necessarily represent money that will be collected by the Revenue Agency, especially when tax collection agents are involved.

"HST_TOT": *higher settled tax* is only taken into account when a taxpayer and tax authorities reach an agreement on the determination of the due amounts.

“HAT_TOT/2": half of the higher assessed tax is only considered when a taxpayer appeals against his *tax assessment notice* and a court takes a partially favorable decision for the Revenue Agency.

Some figures may describe the audit activity carried out in 2014 and 2015 by the Italian Revenue Agency, in the context of income indicators audits.

To start with, **39,757** invitations were sent, of which 24,507 concerning fiscal year 2009 and 15,250 concerning fiscal year 2010, while only **16,195** *tax assessment notices* were issued, of which 3,190 with a so called *negative outcome* (that is, they did not end up with a payment due).

Total higher due taxes assessed has been of about **€ 376,815,209**, of which € 222,064,219 referred to the year 2009 and € 154,750,990 to the year 2010.

So, given our sample of **39,757** taxpayers, we show some statistics about tax claims, higher assessed taxes and higher settled taxes in *Table 3.3.*

Table 3.3 – Statistics for HAT, HST and tax claim

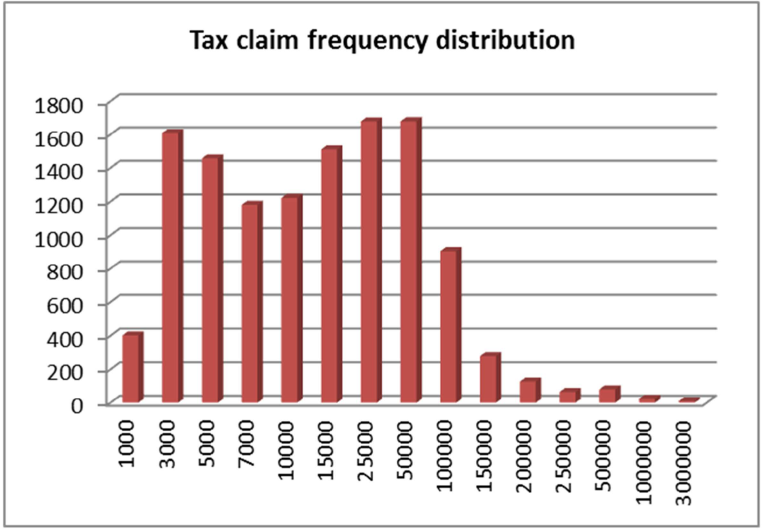
	HAT	**HST**	**Tax claim**
Sum	€ 367,584,437	€ 169,790,828	€ 326,018,991
25 percentile	€ 1,619	€ 0	€ 0
median	€ 8,177	€ 1,347	€ 0
75 percentile	€ 23,103	€ 9,523	€ 3,354
90 percentile	€ 53,285	€ 26,110	€ 19,412
99 percentile	€ 220,818	€ 138,223	€ 127,641
min	€ 0	€ 0	€ 0
max	€ 2,633,283	€ 1,179,573	€ 2633283
avg	€ 23,165	€ 10,700	€ 8,200
mode	€ 0	€ 0	0
std.dev.	€ 60,850	€ 31,017	€ 38,749

Table 3.3 highlights a strongly skewed *dataset*, with more than 50% of zero tax claim taxpayers and only 25% owing a debt greater than **€ 3,354**.

Without considering **27,529** zero tax claim taxpayers, the tax claim frequency distribution is depicted in *Figure 3.5.*

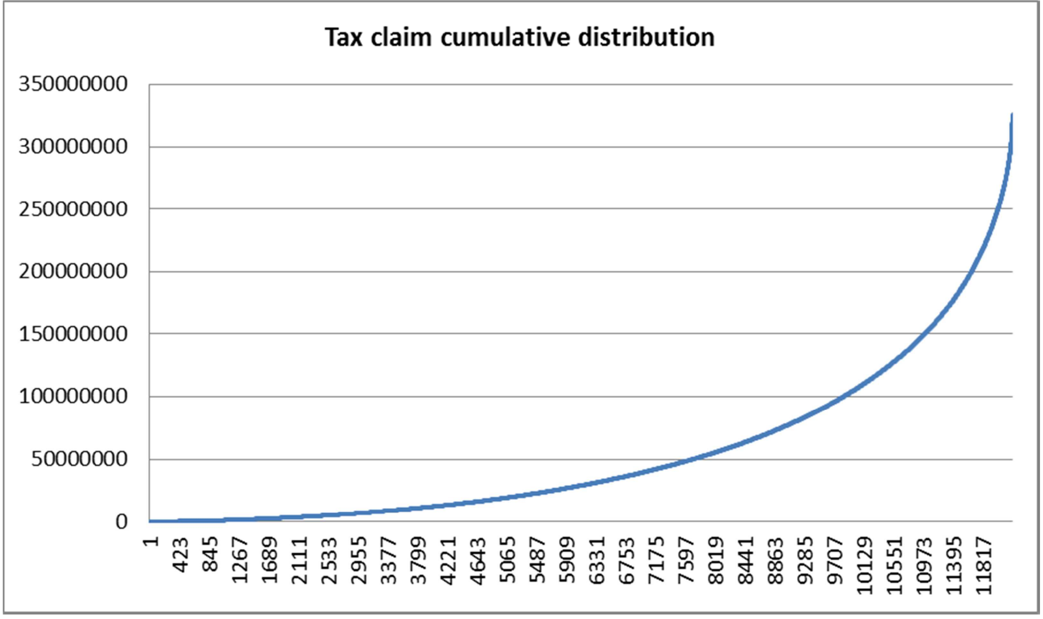
x–axis values represent the right end of each interval: therefore, there are about 400 individuals owing less than € 1,000, 1,600 having tax claims ranging between € 1,000 and € 3,000, and so on. About half the taxpayers in the dataset owe the Revenue Agency less than € 10,000, and almost all of them, 95%, less than € 100,000.

Figure 3.5 – Tax claim frequency distribution



The following chart shows the cumulative tax claim trend: by ordering taxpayers increasingly with respect to their tax claim, it turns out that half of the tax revenue (€ 163 million) would be recovered by identifying the 1,000 most unfaithful taxpayers:

Figure 3.6 – Tax claim cumulative distribution



As *figure 3.6* shows, almost 70% of taxpayers managed to justify the difference between declared income and incurred expenses; as for the remaining 30%, the 25% most unfaithful are liable for 75% of the total revenue.

Tax claims assume numerical real values, while classification models require the class attribute to be nominal. Since the purpose of this paper is to provide some selection criteria, we define a binary class, whose values are "interesting" and "not interesting": indeed, only "interesting" classified taxpayers will be invited, while the others won't.

A natural threshold with which interesting taxpayers could be separated from uninteresting ones could be **€ 0.00**: that is, only when a tax claim is greater than zero, then taxpayers would be classified as "interesting", otherwise not.

However, the threshold could also become one of the data mining process parameters. We will then have model learning schemes in which the tax claim threshold is set at € 0, € 1,000, € 5,000, € 10,000 and € 20,000. This avoids multiplying class values, which would raise complex issues about some other model parameters (e.g., the cost matrix)and make interpreting results more difficult (e.g., confusion matrix with many entries).

Indeed, models incorporating different thresholds behave quite differently: it seems that higher thresholds allow models to select the most profitable taxpayers, although at the expense of a worsening precision rate. However, we will return to this point later on.

We now need to proceed through a data pre-processing stage, topic of the following section.

4. Data pre-processing

Machine learning methods, such as decision trees, are sound and robust techniques, applicable to practical data mining problems. They usually have many parameters, for which suitable values must be chosen, according to the data on hand.

However, other important processes, which constitute a kind of input data engineering, can actually improve success when applying machine learning techniques, as described in [4]. In the following section, we'll show how we addressed issues such as *data cleaning*, adding calculated attributes and *feature selection*.

4.1 Data cleaning

Taxpayers with abnormal values have already been detected, while observing huge differences often existing between the maximum value and the 99th percentile in many expenses types. Such taxpayers are called outliers, i.e. data objects having values of one or more attributes that are unusual with respect to typical values for those attributes. Very often in data analysis activities, outliers are removed from the dataset being studied, as they may negatively affect most parametric statistics, such as means, standard deviations and correlation, since they are highly sensitive to outliers.

Records are commonly considered outliers with respect to an attribute when they deviate from the average value for more than 3 or 5 standard deviations. In our dataset, each record could be considered an outlier with respect to one or more attributes, and not to others. For our purposes, a taxpayer is considered as an outlier if he is so for at least one attribute, i.e.:

$$\{outliers\} = \bigcup_i \{record_i | record_i outlier\ at\ least\ on\ one\ attribute\}$$

As a matter of fact, by identifying as outliers those taxpayers showing, for at least an attribute, a value differing from its average by more than 5 standard deviations, 4,366 records should be removed, i.e. 11% of data. Indeed, they represent a significant part of the entire dataset. We hardly believe they're all misleading records and we are not willing to drop all of them from the entire dataset, because we could lose some precious information about invited taxpayers. Therefore, many extreme records do not actually represent anomalous data, but rather we need to find a more restrictive criterion to identify outliers. To keep things simple, we shall identify as *outliers* those records that, for at least one attribute, differ from the average value by more than 6 and 10 standard deviations (which identifies 3,342 and 1,413 outliers, respectively). We point out that this criterion is based exclusively on data characteristics and has been, thus, empirically derived.

4.2 Data transformation

The dataset on hand needs some calculated attributes, such as taxpayer age (divided in ranges: [0-30]; [31-40]; [41-50]; [51-60]; [61-70]; [71-80]; [81-100]), called "decades" and the Gini coefficients multiplied by 100, with respect to each taxpayer's expenses, computed both considering and excluding investments and divestments: these latter are called *GiniWith* and *GiniWithout* respectively.

The new attribute "decade" frequency distribution is depicted below:

Figure 4.1 – Decade frequency distribution

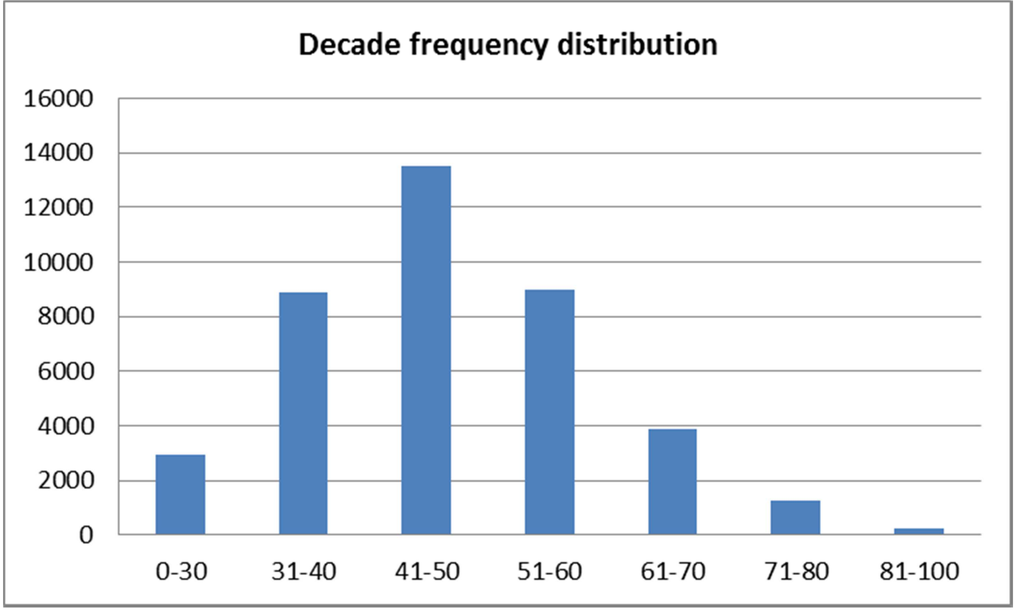


Figure 4.1 shows that tax offices mainly invited mature people (the modal class being that of people in their forties) for whom it can be assumed, on one hand, their economic independence and, on the other, that they have had time to create their own patrimonial situation, which turns out to be inconsistent with their declared incomes.

Table 4.1 – Statistics for Decade

Decade	Num.	Num. tax claim>0	Positivity rate	Total tax claim	Average tax claim
0-30	2,923	858	29.35%	24,069,914,40	8,234.66
31-40	8,909	2,932	32.91%	71,899,084,05	8,070.39
41-50	13,495	4,652	34.47%	118,881,552,43	8,809.30
51-60	8,998	2,780	30.90%	68,217,907.50	7,581.45
61-70	3,892	815	20.94%	29,252,256.00	7,515.99
71-80	1,283	168	13.09%	11,131,356.00	8,676.04
81-100	257	23	8.95%	2,566,921.00	9,988.02
Total	**39,757**	**12,228**		**326,018,991.38**	

Table 4.1 shows that the highest *tax assessment notices* positivity rate is registered among those notified in the forty years old population segment (34.47%) and, even because of this, their resulting average tax claim is among the highest in the data set.

4.3 Feature selection (data reduction)

We have previously pointed out that the expenses' most common value is zero, and that there are many outliers present. Such data features pose some questions about the overall data quality level, given that redundant or irrelevant information or data with "noise" may confuse machine learning algorithms.

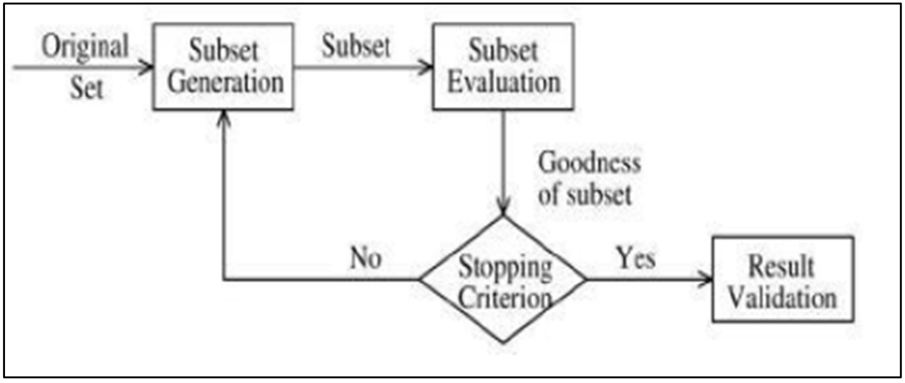
Feature selection, as a preliminary step to machine learning, is concerned with identifying and removing as many irrelevant or redundant attributes as possible from datasets. In general terms, learning algorithms differ greatly as to their ability to select attributes to use: there are those who use them all (e.g. Simple K Nearest Neighbours, Naïve Bayes) and others that, on the contrary, focus on a single one (e.g. OneR). Decision trees are "selective": while testing attribute values, they try to split the training set into subsets containing strong majorities of a single class. This process generally takes place by selecting a small number of attributes considered as the most predictive.

Feature selection has been a fertile field of research since 1970s and has proven to be effective in removing irrelevant and redundant features, increasing efficiency in learning tasks, improving learning performances such as predictive accuracy, reducing computation time and enhancing comprehensibility of learned results.

Feature selection is generally performed by filtering attributes in order to get a good subset of them, prior to applying data mining tasks. The appropriateness of the generated subset is evaluated using an evaluation criterion. Two, from among many possible criteria, will be shown later.

As described in [10], *feature selection* algorithms generally follow the path depicted in *Figure 4.2:* starting from the original set of attributes, they result in the selection of a candidate subset for evaluation. If the newly generated subset is better than the previous one, it replaces it, and the entire process is repeated until a stopping criterion is met. Only at this point are data mining tools applied.

Figure 4.2 – Feature selection process



We have applied two *feature selection* algorithms to our dataset, containing 64 predictive attributes. In what follows, tax claim threshold was set to € 1,000.

The first one, called *Correlation based feature selection*, was first described in [11]. This algorithm evaluates many feature subsets. Good feature subsets contain features highly correlated with the class, yet uncorrelated with each other. Results can be validated in many different ways: in general terms, this can be done over the entire dataset, or with a k-fold cross validation technique, as explained in [13].

Based on available data, this algorithm evaluated 894 different attribute subsets and ended up selecting 10 or 12 of them, depending on the validation method adopted, as shown below:

Figure 4.3 – Correlation based feature selection

```
=== Attribute Selection on all input data ===

Selected attributes: 3,11,12,17,18,30,39,45,47,64 : 10
GiniWith
CORPORATE OFFICES (from year 2010)
ADJUSTED DECLARED INCOME
FAMILY ADJUSTED DECLARED INCOME
SELECTION VALUE
TENANCY
HEALTHFROM DECLARATIONS
VEHICLES LEASING
KW MOTOR VEHICLES
INVESTMENTS

=== Attribute selection 10 fold cross-validation (stratified), seed: 1 ===
number of folds (%)  attribute
10(100 %)      3 GiniWith
10(100 %)     11 CORPORATE OFFICES (from year 2010)
10(100 %)     17 FAMILY ADJUSTED DECLARED INCOME
10(100 %)     18 SELECTION VALUE
10(100 %)     30 TENANCY
10(100 %)     39 HEALTH FROM DECLARATIONS
10(100 %)     45 VEHICLES LEASING
10(100 %)     64 INVESTMENTS
 9( 90 %)     12 ADJUSTED DECLARED INCOME
 5( 50 %)     46 NUM. MOTOR VEHICLES
 5( 50 %)     47 KW MOTOR VEHICLES
 1( 10 %)     13 OVERALL DECLARED INCOME
```

Figure 4.3 attribute subsets are quite similar. The ten-fold cross validation, though, tells us how many times each attribute was selected, while the ten runs were ongoing. In a way, it provides a ranking merit of individual attributes.

To rank attributes means to have on hand a way to order them, according to some criterion. This suggests a perspective change, in which the number of attributes to select becomes a model parameter. An algorithm following this philosophy is the *Information Gain Attribute Evaluation* presented in [12], which evaluates an attribute benefit by measuring its *gain ratio* with respect to the class, that is, the information gain that an attribute would permit if it was the only one in the dataset. Then, a *ranker* would order the selected attributes based on their individual evaluations.

By setting the number of attributes to select at 10, this algorithm chose the following features, ordered according their *Information Gain* benefit (the first list is generated by validating on the entire dataset, while the second by means of a *ten-fold cross validation*):

Figure 4.4 – Information gain feature selection

```
=== Attribute Selection on all input data ===

Ranked attributes:
0.0415    17 FAMILY ADJUSTED DECLARED INCOME
0.0373    64 INVESTMENTS
0.0371    12 ADJUSTED DECLARED INCOME
0.0361    13 OVERALL DECLARED INCOME
0.0306    18 SELECTION VALUE
0.0268     3 GiniWith
0.0241    19 FAMILY SELECTION VALUE
0.0195    39 HEALTH FROM DECLARATIONS
0.0149    47 KW MOTOR VEHICLES
0.0142    30 TENANCY

=== Attribute selection 10 fold cross-validation (stratified), seed: 1 ===

average merit      average rank  attribute
0.041 +- 0.001       1   +- 0        17 FAMILY ADJUSTED DECLARED INCOME
0.037 +- 0           2.5 +- 0.67     64 INVESTMENTS
0.037 +- 0.001       2.6 +- 0.49     12 ADJUSTED DECLARED INCOME
0.036 +- 0.001       3.9 +- 0.3      13 OVERALL DECLARED INCOME
0.03  +- 0.001       5   +- 0        18 SELECTION VALUE
0.026 +- 0.001       6   +- 0         3 GiniWith
0.024 +- 0           7   +- 0        19 FAMILY SELECTION VALUE
0.02  +- 0           8   +- 0        39 HEALTH FROM DECLARATIONS
0.015 +- 0           9.2 +- 0.4      47 KW MOTOR VEHICLES
0.014 +- 0          10.1 +- 0.7      30 TENANCY
0.014 +- 0          10.7 +- 0.46     46 NUM. MOTOR VEHICLES
```

Attribute subsets in *Figure 4.3* and *Figure 4.4* generated by the two algorithms don’t overlap perfectly, but there are some common elements, and we can therefore interpret this finding by saying that there is a hard core of attributes that are significantly more predictive than others.

In the following sections, we will test whether such algorithms are useful, compared to other models that use all existing attributes.

5. Obtaining models from the dataset

Given our set of predictive attributes (see Appendix 1 for more details) and a *class* attribute to predict, we now show how we extracted predictive models, in order to identify useful and interesting patterns among data.

In general terms, given a dataset, i.e. a collection of records, each described by a set of attributes, of which one is defined as the *class*, the aim of any classification process is that of learning a function (i.e. a classification model) that maps each attribute set to one of the predefined class labels, in order to assign the correct class label to unknown records (of the same species as the ones already analyzed), in the most accurate way.

A classification model can thus be treated as a black box that automatically assigns a class label when presented with the attribute set of an unknown record. Normally, when associating a class label to a *record*, the model also provides a probability, which highlights how confident the model is about its own prediction.

After having randomly partitioned the dataset records in two subsets, called *training set* and *test set*, the classification process takes place in two steps:

Training: step in which the model is built, based on the *training set* data: that is, the model trains and learns, by observing each attribute record, including the class value. This phase ends with model formalization in which existing relationships between the class and the other attributes are explained. In our case, this basically means that the model will find out how tax claim is related to data stored in VERDI database.

Test: step in which the newly born model performance is validated on the *test set*: during this step, what the model has learned in the previous phase is tested, on unseen data stored in the *test set*. Since each record class label is known, real values can be compared against predicted ones, using different evaluation metrics.

In the taxpayer selection process, easy to interpret and to understand models are preferred to more complex ones. Typically, decision trees meet the above requested conditions[7].

Instead of considering just one decision tree, both practical and theoretical reasons drove us to more sophisticated techniques, known as ensemble learning, with which different models learned from data are combined. One of these techniques is called *bagging*, that stands for *bootstrap aggregating*, with which many base classifiers are computed (in our case, many trees) as shown in [14]. Briefly, *bagging* consists of a two-steps procedure: first, several base classifiers, trained on different, equally sized, subsets of the original training set, obtained with a *bootstrap* method (i.e. by randomly sampling the training set with replacement) are built. Individual classifiers are then combined on a vote basis, i.e. a certain record is assigned to the class label that was predicted more often. This technique reduces prediction variance: in essence, by averaging predictions, the error rate is reduced.

As previously highlighted, we are designing a suitable and effective learning scheme intended to improve the Italian Revenue Agency's ability to identify non-compliant taxpayers, in the context of income indicators fiscal audits.

[7] The term **tree** is due to the similarity between the learning model representation and a tree, usually depicted upside down.
A tree shows a hierarchical structure, consisting of a finite set of elements called **nodes**, which depart from an initial node called **root node,** that has no incoming edges and zero or more outgoing edges, and are connected through oriented, labeled **edges**. We have two types of nodes: **leaf nodes**, labeled with the class label of the elements that satisfy all the conditions of the path root-leaf (thus they have exactly one incoming edge and no outgoing edges) and the **internal nodes**, labeled based on the splitting attribute, each having one incoming edge and two or more outgoing edges. The splitting criterion is represented by the label on the edges.
So, an instance is classified by starting from the root node of the tree, testing the attribute specified by this node, then moving down the tree edge corresponding to the value of the attribute itself. This process is then repeated for the subtree rooted at the new node, until a leaf is reached. Decision trees divide observations into mutually exclusive subsets, so an instance can only end in one leaf node.
Given a dataset, a tree model can be computed in many ways, depending on how the learning model tackles some fundamental issues, such as how to choose the internal nodes, how to decide when to stop the tree growth (pruning strategy) or how to assign a class label to a leaf.

Thus, we have empirically conducted a few experiments, each time varying the dataset (including / removing outliers and considering all the attributes or only a subset of them) and some fundamental model parameters, such as the *rebalancing of the classes* in the training set and tax claim thresholds, looking for the best possible model.

Anyway, scenarios shown in the next pages share the following common features:

- identical partitioning of the dataset in *training set* (containing 2/3 of *records*) and *test set* (containing the remaining *records*);
- *resampling* of the *training set*: this task increases the number of minority class records in the *training set*, by duplicating some of them until a new proportion between the class labels is reached. Often, this step boosts the predictive capabilities of the model, in case of unbalanced datasets. In our case, about 70%of records were referred to zero tax claim taxpayers and that is why this operation has actually proven to be very useful;
- use of a *cost matrix* while building the model. By giving different weights to different misclassification errors (i.e. to classify a taxpayer as interesting when he's not so – so called *false positive error* – and to classify a taxpayer as not interesting when, instead, he is so – a *false negative error*) turned out to dramatically affect classifier predictions. Intuitively, depending on the error weights settings, models will pay more attention to *false positive* errors rather than to *false negative* ones or viceversa.

  So, when building its base classifiers, the *bagging* algorithm may handle a cost matrix to specify misclassification error costs according to their consequences. In a binary class problem, we have a 2x2 matrix, whose values could be configured as follows:

$$\left[\begin{array}{ccc} \begin{matrix} pred \rightarrow \\ actual \downarrow \end{matrix} & I & NI \\ I & 0 & 1 \\ NI & 1 & 0 \end{array}\right]$$

  if the two types of error had identical consequences. However, in our context, this does not seem to be a realistic assumption because classifying as *interesting* an actual *not interesting* taxpayer is a much more serious error, based on the fact that, generally, tax offices human resources are barely sufficient to perform all the audits they are assigned, so optimizing the average working time of each audit is a relevant task. Clearly, incurring in false positive errors lengthens the average working time of a positive audit, while not checking an interesting taxpayer may not be so harmful, assuming that tax offices resources are never underutilized .

  The most appropriate cost matrix coefficients were empirically chosen after several trials, in which we observed the model's *confusion matrix* changes due to cost matrix weights variations, keeping the other model's parameters still (see Appendix 2 for more details on confusion matrices).

  As we expected, changes in the cost matrix highly affected the model confusion matrix values, as shown below[8]:

[8] The shown example has been obtained in the Scenario 1 context, keeping the *biasToUniformClass* parameter equal to 0.4, the tax claim threshold equal to € 0.00 and the *minNumObj* parameter equal to 500.

$$\begin{bmatrix} pred \rightarrow & & \\ actual\ \downarrow & I & NI \\ I & 0 & 1 \\ NI & 1 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} a & b & < classified\ as \\ 1233 & 2225 & a = Interesting \\ 1005 & 8791 & b = Not\ interesting \end{bmatrix}$$

$$\begin{bmatrix} pred \rightarrow & & \\ actual\ \downarrow & I & NI \\ I & 0 & 1 \\ NI & 1.5 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} a & b & < classified\ as \\ 803 & 2655 & a = Interesting \\ 488 & 9307 & b = Not\ interesting \end{bmatrix}$$

$$\begin{bmatrix} pred \rightarrow & & \\ actual\ \downarrow & I & NI \\ I & 0 & 1 \\ NI & 2 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} a & b & < classified\ as \\ 325 & 3133 & a = Interesting \\ 149 & 9646 & b = Not\ interesting \end{bmatrix}$$

$$\begin{bmatrix} pred \rightarrow & & \\ actual\ \downarrow & I & NI \\ I & 0 & 1 \\ NI & 2.5 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} a & b & < classified\ as \\ 0 & 3458 & a = Interesting \\ 0 & 9795 & b = Not\ interesting \end{bmatrix}$$

That is, by increasing the false positive error weight (ranging from 1 to 2.5), the confusion matrix shows that the model:

- selects fewer and fewer taxpayers to audit, until it decides not to invite anybody;
- becomes more and more precise;

Our results can easily be explained: as the false positive error weight increases, the model behaves like a sniper whose bullets are getting more and more expensive: so, before firing, he has to be quite sure he won’t miss the target.

The most suitable cost matrix appeared thus to be the one shown below:

$$\begin{bmatrix} pred \rightarrow & & \\ actual\ \downarrow & I & NI \\ I & 0 & 1 \\ NI & 2 & 0 \end{bmatrix}$$

A model built by using the above depicted cost matrix indeed satisfies two desirable requirements, as can be argued by observing the associated confusion matrix: it only suggests a small part of the test set records for audits (in the above mentioned example, less than 4%, which may appear quite arbitrary, but it reflects the percentage of sent invitations (about 40,000) compared to the taxpayers satisfying all the needed conditions to be assessed (about 800,000) in the years 2014-2015) and at the same time it ensures the best expected precision rate.

Such a cost matrix turned out to be the best one even in other scenarios, so it has always been used in what follows.

- use of ensemble learning techniques, such as bagging
- use of Ross Quinlan’s C4.5 decision tree model to build base classifiers

Finally, we recall that desirable models are easy to use in real life and easy to read and comprehend. To meet these requirements, trees must not be too deep. Thus, base classifiers used in the *bagging* process were built having the *minNumObj* parameter set to 500 (this parameter influences the tree leaves cardinality).

We point out that simplicity was not reached by sacrificing models accuracy and precision. Indeed, in the Scenario 1 context, varying the *minNumObj* parameter and keeping the *biasToUniformClass* parameter equal to 0.4, the tax claim threshold equal to € 0.00 and the cost matrix equal to:

$$\begin{bmatrix} \begin{matrix} pred \rightarrow \\ actual \downarrow \end{matrix} & I & NI \\ I & 0 & 1 \\ NI & 2 & 0 \end{bmatrix}$$

we obtained the following results:

- *minNumObj = 50:*

```
Correctly Classified Instances       10093               76.1563 %
Incorrectly Classified Instances      3160               23.8437 %

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.153     0.023      0.697     0.153     0.25       0.733    INTERESTING
                 0.977     0.847      0.766     0.977     0.858      0.733    NOT_INTERESTING
Weighted Avg.    0.762     0.632      0.748     0.762     0.7        0.733

=== Confusion Matrix ===

    a    b   <-- classified as
  528 2930 |    a = INTERESTING
  230 9565 |    b = NOT_INTERESTING
```

- *minNumObj = 100:*

```
Correctly Classified Instances       10045               75.7942 %
Incorrectly Classified Instances      3208               24.2058 %

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.124     0.018      0.705     0.124     0.211      0.715    INTERESTING
                 0.982     0.876      0.76      0.982     0.857      0.715    NOT_INTERESTING
Weighted Avg.    0.758     0.652      0.746     0.758     0.689      0.715
```

```
=== Confusion Matrix ===

    a    b   <-- classified as
  430 3028 |    a = INTERESTING
  180 9615 |    b = NOT_INTERESTING
```

- *minNumObj = 200:*

```
Correctly Classified Instances       10024               75.6357 %
Incorrectly Classified Instances      3229               24.3643 %

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.118     0.018      0.695     0.118     0.202      0.693    INTERESTING
                 0.982     0.882      0.759     0.982     0.856      0.693    NOT_INTERESTING
Weighted Avg.    0.756     0.657      0.742     0.756     0.685      0.693

=== Confusion Matrix ===

    a    b   <-- classified as
  408 3050 |    a = INTERESTING
  179 9616 |    b = NON_INTERESTING
```

- *minNumObj = 500:*

```
Correctly Classified Instances        9971               75.2358 %
Incorrectly Classified Instances      3282               24.7642 %

=== Detailed Accuracy By Class ===

                TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                    0.94     0.015      0.686     0.094     0.165      0.701    INTERESTING
                  0.985     0.906      0.755     0.985     0.855      0.701    NOT_INTERESTING
  Weighted Avg.    0.752     0.674      0.737     0.752     0.675      0.701

=== Confusion Matrix ===

    a    b   <-- classified as
  325 3133 |    a = INTERESTING
  149 9646 |    b = NON_INTERESTING
```

As it can easily be pointed out, the four models above described are quite similar as for precision rate (which ranges from 68.6% to 70.5%) and correctly classified instances rate (which ranges from 75.23% to 76.15%).

On the other hand, average number of leaves and average size of trees is greatly affected by the *minNumObj* parameter, as shown below:

Table 5.1 – Average number of leaves and average size of trees at different *numMinObj* parameter values

MinNumObj = 50	
Average Number of Leaves :	104,24
Average Size of the tree :	141,2
MinNumObj = 100	
Average Number of Leaves :	54,4
Average Size of the tree :	73,52
MinNumObj = 200	
Average Number of Leaves :	24,96
Average Size of the tree :	35,88
MinNumObj = 500	
Average Number of Leaves :	8,8
Average Size of the tree :	15,68

As a matter of fact, deeper trees are more difficult to read and to understand. Moreover, complex models are more likely to suffer from overfitting. Not being significantly more predictive than simpler ones, we won't rely on them and in what follows we'll always consider trees built with *minNumObj* parameter set at 500.

Of course, what follows does not claim to be exhaustive, but is rather intended to stimulate reflection and action, by providing a guideline on the use of some well-known data mining algorithms, in the context of income indicators audits.

The following section summarizes the experiments results.

5.1 Scenario 1

In the first scenario that we considered, all of the **64** predictive attributes, and all of the records in the dataset, including outliers, were used.

Three parameters have been varied during the experiment: cost matrix values, tax claim threshold and the proportion, within the training set, of interesting and not interesting taxpayers.

Tax claim threshold was initially set to **€ 0.00**.

The *training set*, a random sample of the original dataset had **26,504** records, corresponding to 2/3 of the total (the complete *dataset* having **39,757** *records*), of which **17,734** are *not interesting,* and **8,770** are *interesting* (thus, in the *training set*, interesting taxpayers accounted for 33% of the population).

The remaining records of the dataset (**13,253** *records*) were used to validate the model (i.e. they formed the *test set*) and were as follows:

- *# test set* = 13,253

- *# interesting taxpayers* = 3,458 (26%)
- *# not-interesting taxpayers* = 9,795 (74%)
- *Total tax claim = €126,114,464 (average tax claim € 9,515).*

We recall that to obtain legible trees (i.e. not too deep), base classifiers used in the *bagging* process were built having the *minNumObj* parameter set to 500. Individual base classifiers were built following a bagging scheme on different equally sized samples of the *training set.* Moreover, since the class label *interesting* was a minority in the training set, the balancing problem has been taken into account through the *biasToUniformClass* parameter, with which the proportion between the two classes in the dataset has been modified. This parameter ranges between 0 (no rebalancing) and 1 (rebalancing that leads to a 50:50 proportion). Balancing consists of replicating some minority class records and, at the same time, in removing an equal number of records belonging to the other class, leaving the *training set* size unaltered.

As mentioned before, the cost matrix used in this scenario is as follows:

$$\begin{bmatrix} \begin{matrix} pred \rightarrow \\ actual \downarrow \end{matrix} & I & NI \\ I & 0 & 1 \\ NI & 2 & 0 \end{bmatrix}$$

Class rebalancing provides the following results:

Table 5.2 – Classification processes with threshold set at € 0,00

biasToUniformClass value	Interesting taxpayers	Not interesting taxpayers	Confusion matrix	Precision rate
0.0	8,770	17,734	a b <-- classified as 0 3458 | a = INTERESTING 0 9795 | b = NOT_INTERESTING	0%
0.3	10,106	16,398	a b <-- classified as 0 3458 | a = INTERESTING 0 9795 | b = NOT_INTERESTING	0%
0.4	**10,618**	**15,886**	**a b <-- classified as** **325 3133 | a = INTERESTING** **149 9646 | b = NOT_INTERESTING**	**68.6%**
0.5	11,099	15,405	a b <-- classified as 357 3101 | a = INTERESTING 166 9629 | b = NOT_INTERESTING	68.3%
0.8	12,532	13,972	a b <-- classified as 713 2745 | a = INTERESTING 411 9384 | b = NOT_INTERESTING	63.4%

Table 5.2 shows that the *biasToUniformClass* parameter affects the model's performance, measured by the true positive ratio (precision). According to this criterion, the best model has the *biasToUniformClass* parameter set to 0.4.

The best model's most significant characteristics are shown below:

Figure 5.1 – Best model metrics, € 0.00 threshold

```
Scheme: CostSensitiveClassifier

Options: -cost-matrix "[0.0 1.0; 2.0 0.0]"

weka.classifiers.meta.Bagging -- -P 64 I 25 -weka.classifiers.trees.J48 -- -C 0.25 -M
500


Correctly Classified Instances        9971               75.2358 %

Incorrectly Classified Instances      3282               24.7642 %


Total Number of Instances            13253


=== Detailed Accuracy By Class ===


             P Rate  FP Rate Precision  Recall  F-Measure   ROC Area  Class

              0.094   0.015    0.686     0.094    0.165      0.701    INTERESTING

              0.985   0.906    0.755     0.985    0.855      0.701    NOT_INTERESTING

Weighted Avg. 0.752   0.674    0.737     0.752    0.675      0.701


=== Confusion Matrix ===

  a    b   <-- classified as

325 3133 |  a = INTERESTING

149 9646 |  b = NOT_INTERESTING
```

Figure 5.1 shows that this model would have suggested to invite **474** taxpayers out of 13,253 (3,6% of the total), of which **325** are actually *interesting* (about 69% – this is the model *precision*) representing almost 10% of interesting taxpayers in the entire *test set* (this is the model *recall*).

The high precision rate (69%) is counterbalanced by a low *recall* rate, since the model is quite cautious when it has to predict that someone is "*interesting*". As pointed out earlier, this may not be a problem, as long as suggested invitations are enough to cover the tax office's operational capacity.

The overall model accuracy rate is **75,23%**: that is, the model correctly identifies75% of the taxpayers in the *test set,* and this represents an *estimate* of the expected model *performance* when it has to classify new unseen *records.*

How confident can we be in this estimate? To answer this question, we follow [13]. First of all, we need some statistical reasoning. A succession of independent events that either succeed or fail is called a *Bernoulli process*. Our learning scheme is similar to a *Bernoulli process*, in that there are the "success" events, corresponding to a right taxpayer classification, and the "failure" events, corresponding to a misclassification error. The number of trials of the process is equal to the test set size. A *Bernoulli process* has one parameter, the success rate p. The question, then, is, what does the accuracy rate tell us about the true success rate p?

Mean and variance of a single Bernoulli trial with success rate p are p and $p(1-p)$, respectively. If N trials are taken from a Bernoulli process, the expected success rate $f = S/N$ is a random variable with the same mean p and variance $p(1-p)/N$. For large N,the distribution of this random variable approaches the normal distribution.

Based on this, we are interested in determining an interval for p at a given confidence level. For a normal distribution Z with zero mean and variance equal to one, $Z \sim N(0{,}1)$, given a confidence level of 95%, we want to answer to our question.

It is well-known that:

$$\Pr(-1{,}96 \le Z \le 1{,}96) = 0{,}95$$

Thus

$$\Pr(-1{,}96 \le \frac{f-p}{\sqrt{p(1-p)/N}} \le 1{,}96)=0{,}95$$

which leads us to:

$$p = \left(f + \frac{1{,}96^2}{2N} \pm 1{,}96 \sqrt{\frac{f}{N} + \frac{f^2}{N} + \frac{1{,}96^2}{4N^2}} \right) \bigg/ \left(1 + \frac{1{,}96^2}{N} \right)$$

That is, at 95% of confidence we obtain the interval [0,745, 0,760] for p. Since the interval size is, indeed, very small, 1.5 percentage points between maximum and minimum, we are confident about the goodness of the model.

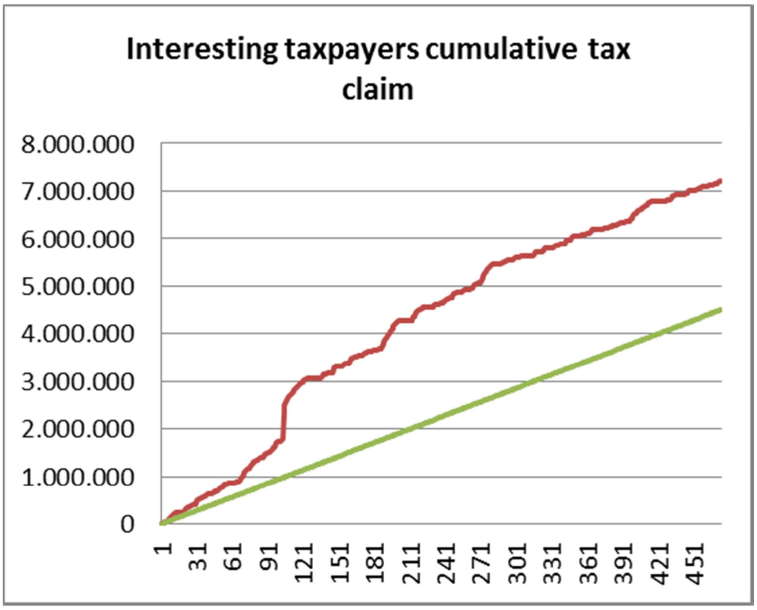
Other evaluation metrics, typically used in classification tasks, are described in Appendix 2.

In previous sections, we have stated that classifiers provide both a predicted value and a probability indicating how confident the model is on its prediction.

We can therefore order *test set* records according to their probability of being “interesting” given by the model: of course, if it is greater than 0.5, the record is classified as interesting, otherwise, as not interesting. By introducing probabilities, we also have different shades of “interesting” and “not interesting” taxpayers (i.e. an interesting taxpayer at a 0.8 level is more interesting than another taxpayer at a 0.6 level, who is still interesting).

This could allow us to investigate the model profitability, for any given set of selected taxpayers. For instance, let’s consider the first **474** more interesting taxpayers i.e. the ones the model would have invited. We can see the chart depicted in *Figure 5.2*, which reports, on the y–axis, the cumulative tax claims corresponding to the first n taxpayers reported on the x–axis (in this case, $n \le 474$). By way of comparison, we also depict the average cumulative tax claim (i.e. what we would expect from a random classifier):

Figure 5.2 – Cumulative tax claim for interesting taxpayers

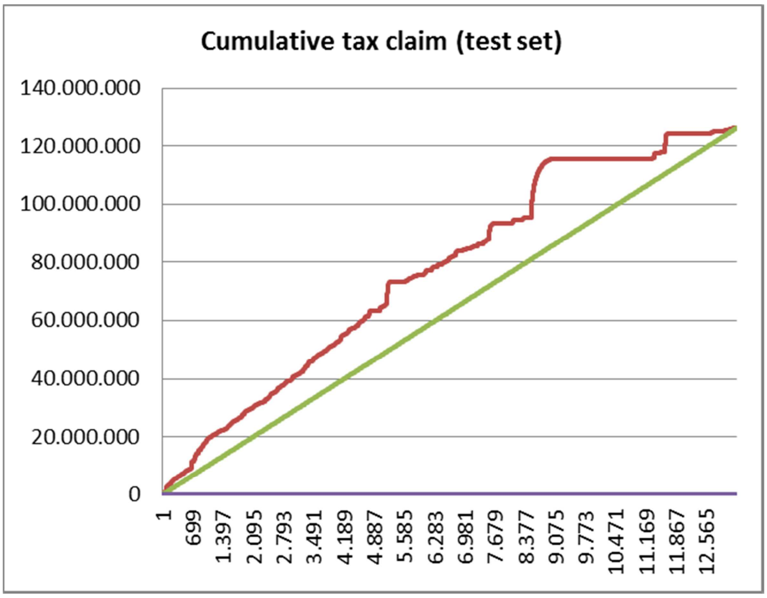


We recall that on the entire *test set* the average tax claim per taxpayer is equal to **€ 9,515.91**. Therefore, by multiplying this average value by the number of selected taxpayers, we would get a total of **€ 4,510,545**.

Is the model we've built more profitable? The answer appears to be "yes". Selected taxpayers would have led to a total recovery of **€ 7,215,227,** corresponding, on average, to **€ 15,222** per taxpayer.

Since all taxpayers could be ordered according to their probability of being interesting, we can introduce it as another possible selection criterion, without being strictly conditioned by the confusion matrix. Indeed, without any probabilistic information, 474 taxpayers would be invited. But if a tax office had to invite, say, 800 of them, it would not know how to select the remaining ones. Knowing each taxpayer's probability of being interesting, it would then select the first 800 taxpayers according to this *ranking* (noticing that taxpayers from the 475^ up to the 800^ position have a probability $p < 0{,}5$ of being "interesting", but nonetheless, higher than the other taxpayers in the test set). *Figure 5.3* chart can then be extended to an arbitrary number n of taxpayers, or even to the entire test set, as shown below:

Figure 5.3 – Cumulative tax claim on the test set



For every n value, we have an expected tax recovery level: the chart shows the model performs better than the random classifier, being its cumulative tax claim constantly above the other. Obviously, for the last taxpayer in the test set, the two cumulative tax claims coincide.

In what we've just seen, interesting taxpayers were those who had a positive tax claim. However, specific tax offices profitability needs may suggest to vary that threshold, in order to consider as interesting only taxpayers with higher tax claims.

So, let's set a new threshold at € 1,000, using the same cost matrix:

$$\begin{bmatrix} \begin{matrix} pred \rightarrow \\ actual\ \downarrow \end{matrix} & I & NI \\ I & 0 & 1 \\ NI & 2 & 0 \end{bmatrix}$$

and varying the *biasToUniformClass* parameter within the *training set*, as follows:

Table 5.2 – Classification processes with threshold set at € 1,000

biasToUniformClass value	Interesting taxpayers	Not interesting taxpayers	Confusion matrix	Precision rate
0.0	8,434	18,070	a b <-- classified as 0 3458 | a = INTERESTING 0 9795 | b = NOT_INTERESTING	0%
0.4	**10,419**	**16,085**	**a b <-- classified as** **276 3116 | a = INTERESTING** **105 9756 | b = NOT_INTERESTING**	**72.4%**
0.5	10,923	15,581	a b <-- classified as 434 2958 | a = INTERESTING 247 9614 | b = NOT_INTERESTING	63.7%

The best model, according to its precision rate, is the one where the *biasToUniformClass* parameter is set to 0.4, detailed below:

Figure 5.4 – Best model metrics, € 1,000 threshold

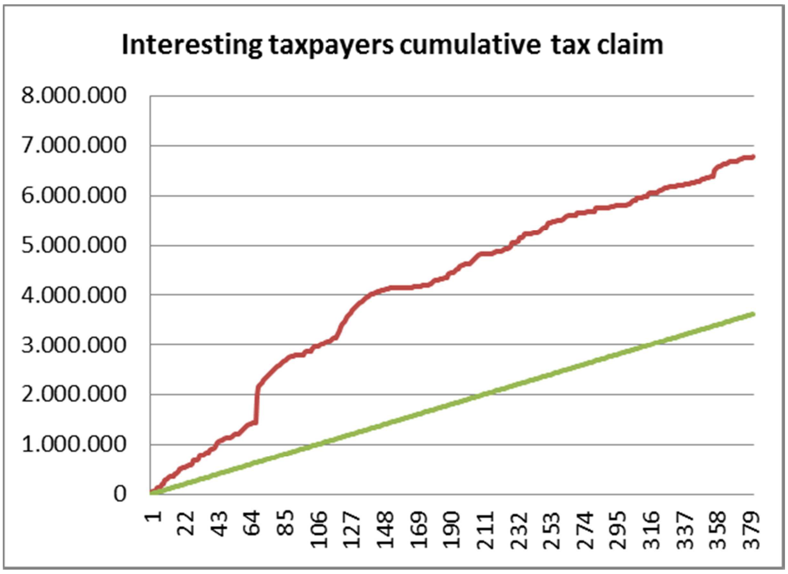
```
Scheme: CostSensitiveClassifier
Options: -cost-matrix "[0.0 1.0; 2.0 0.0]"
weka.classifiers.meta.Bagging -- -P 64 I 25 -weka.classifiers.trees.J48 -- -C 0.25 -M
500


Correctly Classified Instances       10032              75.6961 %
Incorrectly Classified Instances      3221              24.3039 %


=== Detailed Accuracy By Class ===


             TP Rate  FP Rate  Precision  Recall  F-Measure   ROC Area  Class
              0.081    0.011     0.724    0.081     0.146      0.704    INTERESTING
              0.989    0.919     0.758    0.989     0.858      0.704    NOT_INTERESTING
Weighted Avg. 0.757    0.686     0.749    0.757     0.676      0.704


=== Confusion Matrix ===
    a    b   <-- classified as
  276 3116 |    a = INTERESTING
  105 9756 |    b = NOT_INTERESTING
```

Setting a € 1,000 threshold has led us to a *training set* with 18,070 “*not interesting*” taxpayers (68% of total), and 8,434 interesting (32% of total): compared with the previous case, we register only minimal differences, as only a few taxpayers received a tax assessment notice with a tax claim lower than 1,000 euro.

However, the precision rate has slightly increased (**72,4%**against **69%**), and the model would have suggested **381** taxpayers to be invited. As we’ve just seen, though, the ranking given by their probability would allow tax officers to wisely select even more (or less) taxpayers, as explained before.

Considering the first **381** selected taxpayers, we can draw a chart as depicted in *Figure 5.5*, which reports, on the y–axis, the cumulative values of the tax claim corresponding to the first *n* taxpayers on the x–axis:

Figure 5.5 – Cumulative tax claim of “interesting” taxpayers



On the entire *test set* the average tax claim per taxpayer is equal to **€ 9,515.91**, so the first 381 taxpayers would yield a total recovery of **€ 3,625,564**. The model, on the other hand, would have led to **€ 6,774,595**, corresponding to an average of **€ 17,781** per taxpayer.

We can further increase the threshold, say to **€ 5,000**, obtaining thus a *training set* with 20,487 *not interesting* records and 6,017 *interesting ones*. The unbalancing problem thus becomes more severe, and to improve model performances, we need to set the *biasToUniformClass* parameter to higher values. Indeed, the best results are obtained setting it at 0.7 and 0.8, as shown below:

Table 5.4 – Classification process with threshold set to € 5,000

biasToUniformClass value	Interesting taxpayers	Not interesting taxpayers	Confusion matrix	Precision rate
0.7	11,220	15,284	a b <-- classified as 327 2417 | a = INTERESTING 189 10320 | b = NOT_INTERESTING	63.4%
0.8	11,950	14,554	a b <-- classified as 377 2367 | a = INTERESTING 257 10252 | b = NOT_INTERESTING	59.5%

Table 5.4 shows that the best model, which is detailed below, has the *biasToUniformClass* parameter set to 0.7:

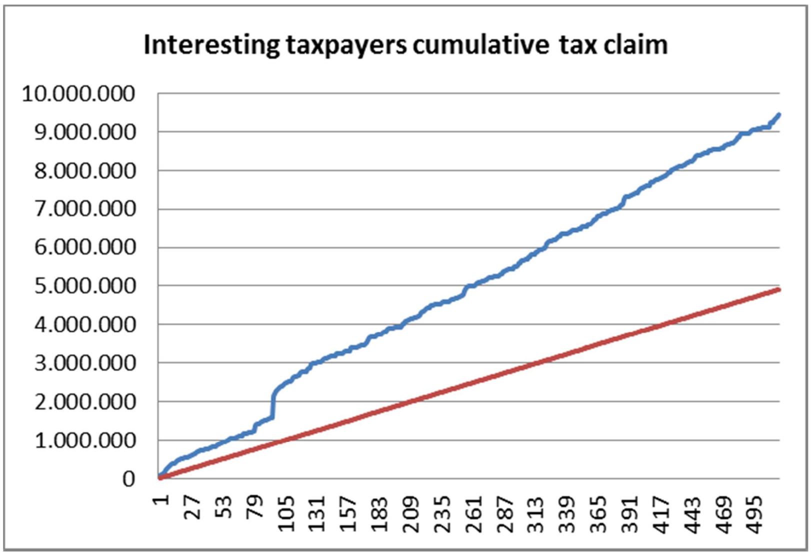
Figure 5.6 – Best model metrics - € 5,000 threshold

```
=== Evaluation result ===
Scheme: CostSensitiveClassifier
Options: -cost-matrix "[0.0 1.0; 2.0 0.0]"
weka.classifiers.meta.Bagging -- -P 64 I 25 -weka.classifiers.trees.J48 -- -C 0.25 -M 500

Correctly Classified Instances       10647               80.3365 %
Incorrectly Classified Instances      2606               19.6635 %

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure  ROC Area  Class
                 0.119     0.018      0.634     0.119     0.201     0.645    INTERESTING
                 0.982     0.881      0.81      0.982     0.888     0.645   NOT_INTERESTING
Weighted Avg.    0.803     0.702      0.774     0.803     0.746     0.645

=== Confusion Matrix ===
     a     b   <-- classified as
   327  2417 |     a = INTERESTING
   189 10320 |     b = NOT_INTERESTING
```

According to *Figure 5.6*, the model predictive capability seems to have worsened (precision rate of 63,4%), but taxpayers that were previously judged as interesting (when the threshold was lower), now may have been considered as not interesting: in a sense, the model is judged by a more severe arbiter. We'll come back to this issue later on.

Considering the first **516** selected taxpayers, we can draw a chart, as depicted in *Figure 5.7*, which reports, on the y–axis, the cumulative tax claim corresponding to the first *n* taxpayers on the x–axis:

Figure 5.7: cumulative tax claim of "interesting" taxpayers



On the entire *test set,* the average tax claim is equal to **€ 9,515.91**, so the first 516 taxpayers would yield total recovery of **€ 4,909,740.** The model, on the other hand, would have led to **€ 9,446,875**, corresponding to an average of **€ 18,308** per taxpayer: the average tax claim, thus, is still increasing.

Let's see what happens with further threshold increases, to € 10,000 and € 20,000. The best models in these cases are reported below:

Table 5.5 – Best classification processes, threshold € 10.000 and € 20.000

Threshold	Confusion matrix	Precision rate	Correctly classified instances	Average tax claim on interesting taxpayers
€ 10,000	a b <-- classified as 327 2417 | a = INTERESTING 189 10320 | b = NOT_INTERESTING	57.7%	83.77%	€ 23,757
€ 20,000	a b <-- classified as 377 2367 | a = INTERESTING 257 10252 | b = NOT_INTERESTING	47.7%	88.53%	€ 31,676

A threshold set at € 10,000 leads to a model which would suggest 333 taxpayers with an overall recovery of **€ 7,911,312**, corresponding to an average of **€ 23,747** per taxpayer. A threshold set at € 20,000, instead, leads to a model which would suggest 342 taxpayers, with an overall recovery of **€ 10,833,236**, corresponding to an average of **€ 31,676** per taxpayer.

As threshold values increase, positivity-rates-derived confusion matrices lose significance, as precision rates measure the models' "success" rate, but what a "success" actually is depends on the threshold value. That is, when the threshold is set to € 0.00, "success" means "tax claim greater than € 0.00", when the threshold is set to €20,000, "success" means "tax claim greater than € 20,000". As a result, models with different threshold levels cannot be compared on a precision rate basis. We can overcome this issue by comparing the computed models on the basis of a fixed percentage of the *test set* (say, 5%[9]), and find both the number of positive tax claim taxpayers and the average tax claim each model would ensure, as in the table that follows:

Table 5.6 – Models comparison

Best model per threshold	Positive tax claim taxpayers on 5% of test set	Expected recovery on 5% of test set	Average tax claim
€ 0, bias 0.4	441 (66.52%)	€9,145,803	€ 13,794
€ 1,000, bias 0.4	453 (68.33%)	€ 10,928,364	€ 16,483
€ 5,000, bias 0.7	452 (68.17%)	€ 11,854,572	€ 17,880
€ 10,000, bias 0.7	405 (61.09%)	€ 16,974,595	€ 25,602
€ 20,000, bias 0.7	365 (55.05%)	€ 24,430,834	€ 36,848

[9] A 5% threshold may appear quite arbitrary, but, as recalled earlier, it reflects the percentage of sent invitations (about 40,000) compared to the taxpayers satisfying all the needed conditions to be assessed (about 800,000) in the years 2014-2015.

Table 5.6 seems to suggest the following relationships:

- as threshold increases, precision rate decreases. Notice that since models' precision rates are now always computed in the same way, models themselves are actually comparable
- as threshold increases, average tax claim return increases

The above depicted situation shows a clear *tradeoff* between precision rate and tax claim returns, to which we'll come back later on.

5.2 Scenario 2

In this second scenario, outliers are removed from the training set. Outliers were defined as values differing more than 6 (case **a**) or 10 (case **b**) standard deviation from the mean value in at least one attribute. In the former, the *training set* has **24,138** records and in the latter,**25,488** (while the original *training set* was made up of 26,504). The *test set* will continue having **13,253** records, as seen before.

We point out that, instead of removing the outliers from the training set, we could also have kept them and erased their extreme values (we recall that Ross Quinlan's 4.5 model is able to handle null values). We preferred not to do this, considering the Gini Index analysis results (see section 3), which showed that many taxpayers in the dataset had only a few anomalous spending elements which encouraged tax authorities to invite them. So, we could easily erase the only expenditure value for which these taxpayers had been invited. In such cases, thus, the model could have observed some tax claim amounts not corresponding to any particular expenditure value, which may have negatively affected its overall predictive analysis.

The usual cost matrix is used, as well as and the same *bagging* algorithm.

Results of case **a)** are shown below:

Table 5.7 – Classification results, outliers at 6 std. dev.

<table>
<tr><th>Threshold</th><th>Original partitioning</th><th>Partitioning after resampling</th><th>Confusion matrix</th><th>Precision rate</th><th>Correctly classified instances</th><th>Average tax claim on interesting taxpayers</th></tr>
<tr><td>€ 0.00
bias 0.4</td><td>NI:15,977
I: 8,161</td><td>NI:14,400
I: 9,738</td><td>a b &lt;-- classified as
310 3148 | a = INTERESTING
125 9670 | b = NOT_INTERESTING</td><td>71.3%</td><td>75.3%</td><td>€ 14,393</td></tr>
<tr><td>€ 1,000
bias 0.4</td><td>NI:16,304
I: 7,834</td><td>NI:14,607
I: 9,531</td><td>a b &lt;-- classified as
79 3313 | a = INTERESTING
19 9842 | b = NOT_INTERESTING</td><td>80.6%</td><td>74.85%</td><td>€ 16,267</td></tr>
<tr><td>€ 5,000
bias 0.7</td><td>NI:18,645
I: 5,493</td><td>NI:14,064
I: 10,074</td><td>a b &lt;-- classified as
376 2368 | a =INTERESTING
236 10273 | b =NOT_INTERESTING</td><td>61.4%</td><td>80.35%</td><td>€ 24,573</td></tr>
<tr><td>€ 10,000
bias 0.7</td><td>NI:20,425
I: 3,713</td><td>NI:14,557
I: 9,581</td><td>a b &lt;-- classified as
271 1931 | a = INTERESTING
242 10809 | b =NOT_INTERESTING</td><td>52.8%</td><td>83.60%</td><td>€ 30,834</td></tr>
<tr><td>€ 20,000
bias 0.7</td><td>NI:22,108
I:2,030</td><td>NI:15,155
I: 8,983</td><td>a b &lt;-- classified as
162 1342 | a = INTERESTING
188 11561 | b =NOT_INTERESTING</td><td>46.3%</td><td>88.45%</td><td>€ 39,285</td></tr>
</table>

Similarly, in case **b)**, we have the following results:

Table 5.8 – Classification results, outliers at 10 std. dev.

Threshold	Original partitioning	Partitioning after resampling	Confusion matrix	Precision rate	Correctly classified instances	Average tax claim on interesting taxpayers
€ 0.00 bias 0.4	NI:16,940 I: 8,548	NI:15,402 I: 10,086	a b <-- classified as 325 3133 | a = INTERESTING 152 9643 | b = NOT_INTERESTING	68.1%	75.21%	€ 15,672
€ 1,000 bias 0.4	NI:17,272 I: 8,216	NI:15,577 I: 9,911	a b <-- classified as 164 3228 | a = INTERESTING 49 9812 | b = NOT_INTERESTING	77.0%	75.27%	€ 21,157
€ 5,000 bias 0.7	NI:19,670 I: 5,818	NI:14,943 I: 10,545	a b <-- classified as 277 2467 | a = INTERESTING 147 10362 | b =NOT_INTERESTING	65.3%	80.27%	€ 30,134
€ 10,000 bias 0.7	NI:21,502 I: 3,986	NI:15,507 I: 9,981	a b <-- classified as 208 1994 | a = INTERESTING 162 10889 | b =NOT_INTERESTING	56.2%	83.73%	€ 30,100
€ 20,000 bias 0.7	NI:23,242 I: 2,246	NI:16,008 I: 9,480	a b <-- classified as 224 1280 | a = INTERESTING 482 11267 | b =NOT_INTERESTING	31.7%	86.70%	€ 33,679

To compare the just built models, we can proceed as in the previous scenario, by selecting a fixed percentage of the test set(e.g. the usual 5%), and thus comparing precision rate and average tax claim on these taxpayers. Data is shown below:

Table 5.9 – Comparing models of Scenario 2

6 std. dev.			
Best model per threshold	Positive tax claim taxpayers on 5% of test set	Expected recovery on 5% of test set	Average tax claim
€ 0.00; bias 0.4	444 (66.96%)	€ 9,815,033	€ 14,803
€ 1,000; bias 0.4	445 (67.11%)	€ 9,660,313	€ 14,570
€ 5,000; bias 0.7	436 (65.76%)	€ 16,283,886	€ 24,560
€ 10,000; bias 0.7	385 (58.06%)	€ 20,491,106	€ 30,906
€ 20,000; bias 0.7	292 (44.04%)	€ 24,243,344	€ 36,566
10 std. dev.			
Best model per threshold	Positive tax claim taxpayers on 5% of test set	Expected recovery on 5% of test set	Average tax claim
€ 0.00; bias 0.4	441 (66.51%)	€ 9,262,943	€ 13,971
€ 1,000; bias 0.4	461 (69.53%)	€ 10,242,995	€ 15,449
€ 5,000; bias 0.7	422 (63.65%)	€ 16,709,809	€ 25,203
€ 10,000; bias 0.7	379 (57.16%)	€ 18,483,105	€ 27,877
€ 20,000; bias 0.7	266 (40.12%)	€ 23,518,276	€ 35,472

For each sub-scenario, we have thus computed 5 models. A criterion to choose the best sub-scenario is now needed. A possible option could be that of computing the overall average tax claim relative to each sub-scenario. This would lead us to choose the 6 standard deviations scenario, as its average tax claim is equal to € 24,281, against € 23,594 of the other case.

5.3 Scenario 3

In this last scenario, decision trees are built on a training set containing all the records, but having only a subset of the original attributes, i.e. the ones selected by the *Correlation based feature selection* algorithm previously seen, as follows:

- GiniWith
- CORPORATE OFFICES (from year 2010)
- ADJUSTED DECLARED INCOME
- FAMILY ADJUSTED DECLARED INCOME
- SELECTION VALUE
- TENANCY
- HEALTH FROM DECLARATIONS
- VEHICLES LEASING
- KW MOTOR VEHICLES
- INVESTMENTS

We continue the same *bagging* algorithm.

Results are shown below:

Table 5.10 – Classification results of scenario 3

Threshold	Original partitioning	Partitioning after resampling	Confusion matrix	Precision rate	Correctly classified instances	Average tax claim on interesting taxpayers
€ 0.00 bias 0.4	NI:17,734 I: 8,770	NI:15,886 I: 10,618	a b <-- classified as 336 3122 | a = INTERESTING 153 9642 | b = NOT_INTERESTING	68.7%	75.28%	€ 14,989
€ 1,000 bias 0.4	NI:18,070 I: 8,434	NI:16,085 I: 10,419	a b <-- classified as 190 3202 | a = INTERESTING 62 9799 | b = NOT_INTERESTING	75.4%	75.37%	€ 19,233
€ 5,000 bias 0.7	NI:10,487 I: 6,017	NI:15,284 I: 11,220	a b <-- classified as 366 2378 | a = INTERESTING 254 10255 |b = NOT_INTERESTING	59.0%	80.14%	€ 16,372
€ 10,000 bias 0.7	NI:22,347 I: 4,157	NI:16,063 I: 10,441	a b <-- classified as 188 2014| a = INTERESTING 151 10900| b = NOT_INTERESTING	55.5%	83.66%	€ 21,688
€ 20,000 bias 0.7	NI:24,115 I: 2,389	NI:16,587 I: 9,917	a b <-- classified as 102 1402| a = INTERESTING 83 11666| b = NOT_INTERESTING	55.1%	88.79%	€ 32,217

To compare the just built models, we can proceed as in the previous scenarios, by selecting a fixed percentage of the test set (e.g. the usual 5%), and thus comparing precision rate and average tax claim on these taxpayers. Data is shown below:

Table 5.11 – Comparing classification models in Scenario 3

Best model per threshold	Positive tax claim taxpayers on 5% of test set	Expected recovery on 5% of test set	Average tax claim
€ 0.00; bias 0.4	425 (64.10%)	€ 8,951,287	€ 13,501
€ 1,000; bias 0.4	455 (68.62%)	€ 9,982,590	€ 15,056
€ 5,000; bias 0.7	439 (66.21%)	€ 11,194,008	€ 16,883
€ 10,000; bias 0.7	414 (62.44%)	€ 15,089,584	€ 22,759
€ 20,000; bias 0.7	297 (44.79%)	€ 24,515,249	€ 36,976

Results have slightly worsened, if compared with those of the first scenario, in which the learning scheme could work on all attributes. This could be due to the high Gini coefficients among taxpayers, which means that many of them had just one or two expense types originating their tax assessment. Thus, the removal of several attributes may have erased the tax assessment origin for a certain number of taxpayers, and this could have made it harder for the model to correctly predict the class labels.

6. Interpreting the results

So far, for each scenario and for each threshold we've considered, we have been able to choose the best model on a precision rate basis, as shown below:

Table 6.1 – Scenarios 1, 2, 3 best models

Best model per threshold	Positive tax claim taxpayers on 5% of test set		Average tax claim	Expected recovery on 5% of test set
Scenario 1				
€ 0.00; bias 0.4	441	66.52%	€ 13,794	€ 9,145,422
€ 1,000; bias 0.4	453	68.33%	€ 16,583	€ 10,994,529
€ 5,000; bias 0.7	452	68.17%	€ 17,880	€ 11,854,440
€ 10,000; bias 0.7	405	61.09%	€ 25,602	€ 16,974,126
€ 20,000; bias 0.7	365	55.05%	€ 36,848	€ 24,430,224
		Total	€ 110,707	€ 73,398,741
Scenario 2 (6 std. dev.)				
€ 0.00; bias 0.4	444	66.97%	€ 14,978	€ 9.930,414
€ 1,000; bias 0.4	445	67.12%	€ 14,570	€ 9,659,910
€ 5,000; bias 0.7	436	65.76%	€ 24,560	€ 16,283,280
€ 10,000; bias 0.7	385	58.07%	€ 30,906	€ 20,490,678
€ 20,000; bias 0.7	292	44.04%	€ 36,566	€ 24,243,258
		Total	€ 121,580	€ 80,607,540
Scenario 3				
€ 0.00; bias 0.4	425	64.10%	€ 13,501	€ 8,951,163
€ 1,000; bias 0.4	455	68.63%	€ 15,056	€ 9,982,128
€ 5,000; bias 0.7	439	66.21%	€ 16,883	€ 11,193,429
€ 10,000; bias 0.7	414	62.44%	€ 22,759	€ 15,089,217
€ 20,000; bias 0.7	297	44.80%	€ 36,976	€ 24,515,088
		Total	€ 105,175	€ 69,731,025

A criterion is now needed to select the best scenario. A possible option, again, could be that of maximizing the overall expected tax claim. According to this criterion, the second scenario should be chosen.

Finally, among the second scenario models, we ought to choose the best one, taking into account two conflicting needs: on one hand, the average tax claim recovery, and on the other one, the precision rate maximization. The latter also seems to be more important than the former, given the most recent Revenue Agency orientations (see, for more details, Circolare 16/E/2016).

To satisfy the two afore mentioned requirements, we compute, for each model, the following variable:

$$precision\ rate\ x\ \ln(\ expected\ tax\ claim)$$

because it makes it possible for low precision rates to penalize high tax claims. Moreover, tax claim logarithm dampens the effects of any further anomalous records:

Table 6.2 – Selecting the best model

Best model per threshold	Positive tax claim taxpayers on 5% of test set		Average tax claim	Expected recovery on 5% of test set	Prob x Ln (Avg tax claim)
Scenario 2 (6 std. dev.)					
€ 0.00; bias 0.4	444	66.97%	€ 14,978	€ 9.930,414	6.44
€ 1,000; bias 0.4	445	67.12%	€ 14,570	€ 9,659,910	6.43
€ 5,000; bias 0.7	**436**	**65.76%**	**€ 24,560**	**€ 16,283,280**	**6.65**
€ 10,000; bias 0.7	385	58.07%	€ 30,906	€ 20,490,678	6.00
€ 20,000; bias 0.7	292	44.04%	€ 36,566	€ 24,243,258	4.63

Hence, according to this criterion, the best *tradeoff* is given by the model in which the tax claim threshold is set to € 5,000, in a scenario without *outliers*, in which all available attributes are taken into consideration.

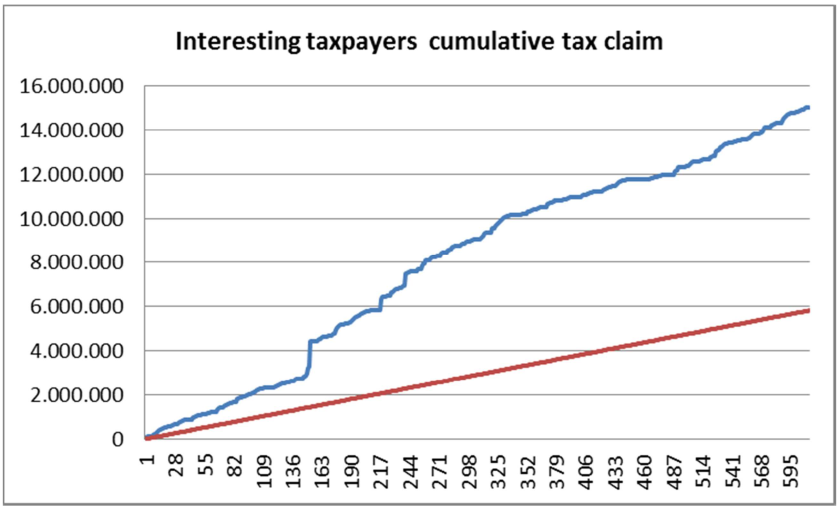
The best model's most significant characteristics are shown below:

Figure 6.1 – Best model

```
Correctly Classified Instances       10649               80.3516 %
Incorrectly Classified Instances      2604               19.6484 %
Total Number of Instances            13253

=== Detailed Accuracy By Class ===
             TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
              0.137     0.022      0.614     0.137     0.224      0.621    INTERESTING
              0.978     0.863      0.813     0.978     0.888      0.621    NOT_INTERESTING
Weighted Avg. 0.804     0.689      0.772     0.804     0.75       0.621

=== Confusion Matrix ===
     a     b   <-- classified as
   376  2368 |     a = INTERESTING
   236 10273 |     b = NOT_INTERESTING
```

Furthermore, the cumulative tax claim is as follows:

Figure 6.2 – Model recovery on interesting taxpayers

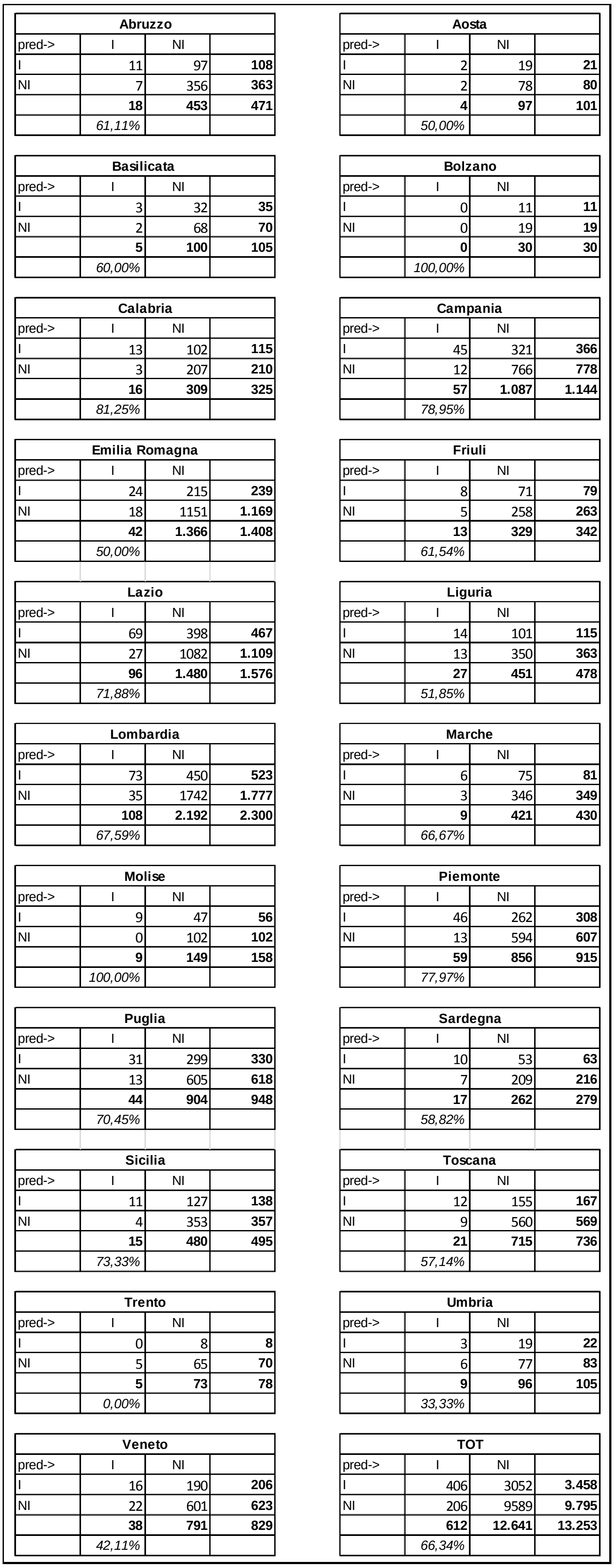


The model would have selected **612** taxpayers (corresponding to 4,61% of the test set), of whom **376** – i.e. 61,4% – are actually *interesting* (i.e. with a tax claim greater than € 5,000) who would have ensured an expected overall recovery of € 15,038,903: thus, on average, **€ 24,573** per taxpayer, while the actual average recovery has been equal to **€ 9,515.** Of course, actual fraudsters are a bit more. If we consider those with a tax claim x, with $0 < x < 5{,}000$ they sum up to **406** (66% of selected taxpayers).

We recall that the actual audit activity registered a positive rate of about 30% (i.e. only 12,228 invitations out of 39,757 ended up with a positive tax claim). Our methodology, on the other hand, leads to a positivity rate of about **66%,** more than the double.

This result appears to be recurrent in all Italian regions, as shown in *Figure 6.3*:

Figure 6.3 – Confusion matrices per region



Abruzzo			
pred->	I	NI	
I	11	97	**108**
NI	7	356	**363**
	18	**453**	**471**
	61,11%		

Aosta			
pred->	I	NI	
I	2	19	**21**
NI	2	78	**80**
	4	**97**	**101**
	50,00%		

Basilicata			
pred->	I	NI	
I	3	32	**35**
NI	2	68	**70**
	5	**100**	**105**
	60,00%		

Bolzano			
pred->	I	NI	
I	0	11	**11**
NI	0	19	**19**
	0	**30**	**30**
	100,00%		

Calabria			
pred->	I	NI	
I	13	102	**115**
NI	3	207	**210**
	16	**309**	**325**
	81,25%		

Campania			
pred->	I	NI	
I	45	321	**366**
NI	12	766	**778**
	57	**1.087**	**1.144**
	78,95%		

Emilia Romagna			
pred->	I	NI	
I	24	215	**239**
NI	18	1151	**1.169**
	42	**1.366**	**1.408**
	50,00%		

Friuli			
pred->	I	NI	
I	8	71	**79**
NI	5	258	**263**
	13	**329**	**342**
	61,54%		

Lazio			
pred->	I	NI	
I	69	398	**467**
NI	27	1082	**1.109**
	96	**1.480**	**1.576**
	71,88%		

Liguria			
pred->	I	NI	
I	14	101	**115**
NI	13	350	**363**
	27	**451**	**478**
	51,85%		

Lombardia			
pred->	I	NI	
I	73	450	**523**
NI	35	1742	**1.777**
	108	**2.192**	**2.300**
	67,59%		

Marche			
pred->	I	NI	
I	6	75	**81**
NI	3	346	**349**
	9	**421**	**430**
	66,67%		

Molise			
pred->	I	NI	
I	9	47	**56**
NI	0	102	**102**
	9	**149**	**158**
	100,00%		

Piemonte			
pred->	I	NI	
I	46	262	**308**
NI	13	594	**607**
	59	**856**	**915**
	77,97%		

Puglia			
pred->	I	NI	
I	31	299	**330**
NI	13	605	**618**
	44	**904**	**948**
	70,45%		

Sardegna			
pred->	I	NI	
I	10	53	**63**
NI	7	209	**216**
	17	**262**	**279**
	58,82%		

Sicilia			
pred->	I	NI	
I	11	127	**138**
NI	4	353	**357**
	15	**480**	**495**
	73,33%		

Toscana			
pred->	I	NI	
I	12	155	**167**
NI	9	560	**569**
	21	**715**	**736**
	57,14%		

Trento			
pred->	I	NI	
I	0	8	**8**
NI	5	65	**70**
	5	**73**	**78**
	0,00%		

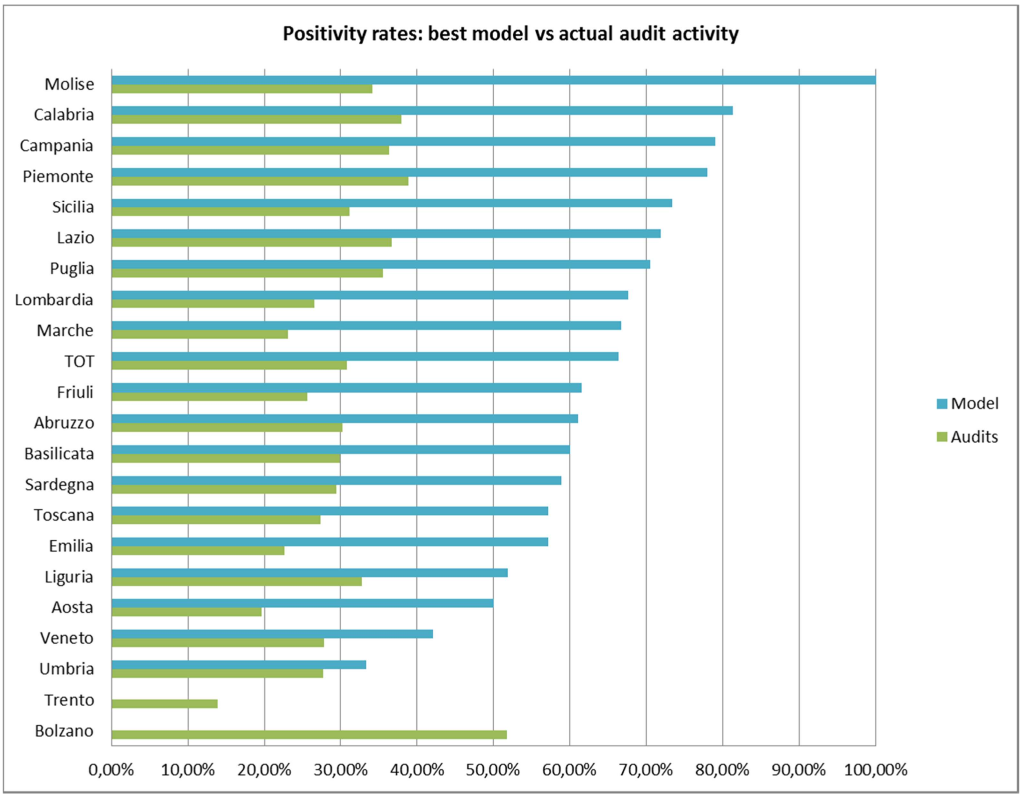
Umbria			
pred->	I	NI	
I	3	19	**22**
NI	6	77	**83**
	9	**96**	**105**
	33,33%		

Veneto			
pred->	I	NI	
I	16	190	**206**
NI	22	601	**623**
	38	**791**	**829**
	42,11%		

TOT			
pred->	I	NI	
I	406	3052	**3.458**
NI	206	9589	**9.795**
	612	**12.641**	**13.253**
	66,34%		

For each region, model and actual audits performances can thus be compared:

Figure 6.4 – Positivity rates per region, model vs audits



Let's now analyze what kind of trees have been built by this model, looking for interesting and recurrent *patterns* (See Appendix 3 for more details).

Our bagging learning scheme created 25 trees, each built according to Ross Quinlan's C4.5 algorithm, as implemented in Weka. Having set Weka *minNumObj* parameter at 500, each tree is not too deep, so it can easily be read (see Appendix 3).

As we expected (and hoped), we can, indeed, learn some recurrent patterns from the decision trees, such as:

if $selection\ value < 20{,}000$ then a taxpayer is more likely to be *not interesting* than *interesting*, because the attribute *selection value* always appears as a root node followed by a leaf labeled “not interesting” when the splitting condition on the edge is "<20,000".

Moreover, we could expect a taxpayer to be interesting if, given a high value of expenses, his income is quite low. This assumption is confirmed by our model, as we often find that, departing from a root node $selection\ value > 20{,}000$, if $overall\ income\ > x$ or$family\ income > y$ follow, the taxpayer is more likely to be not interesting. That is, to be interesting, a taxpayer should show a low income, if $selection\ value > 20{,}000$ holds.

Furthermore, some goods, such as motor vehicles, if in large quantities, suggest interesting taxpayers: indeed, paths including clauses such as $motor\ vhicles\ nr\ > n$ or $motor\ vehicles\ kw\ \ > m$ often end up in leaves labelled as *interesting*.

On the contrary, attribute *investments*, when in the form $investments > x$ suggests the taxpayer being more likely *not interesting*. This may happen because, if a taxpayer makes a significant investment (such as the purchase of an expensive house), he can often afford it because there is someone else helping (e.g. think about parents lending money to their sons).

Many attributes placed near to the root node (that is, those which ensure the higher information gain from the split) are the ones selected by *attribute selection* algorithms seen previously, such as: *selection value, overall income, overall family income, motor vehicles kw, investments*. This is encouraging, because it suggests that the learning scheme used the best attributes it could at first, and the others while approaching to the leaves.

Other expenses, including rent, insurance or medical expenses often appear near to the leaves, and are not easily interpreted, being of small entity and having to be evaluated within the path they belong to.

Quite surprisingly, attributes such as family type or taxpayer's age don't seem to be relevant.

As for the taxpayer's age, *Table 4.1* has shown that people aged 30 to 60 represent nearly 80% of the dataset, and that no significant average tax claim differences were pointed out among them. Thus, the learning scheme may have considered age not to be a significant selective attribute, or at least, much less than other attributes. The *Decade* attribute, however, appears in trees having a *numMinObj* parameter set to *100*, which are much deeper than the ones depicted in Appendix 3. Their extra detail, though, is not compensated by a better performance, as previously shown. Moreover, it adds a few drawbacks, as it increases both the model complexity and the risk of *overfitting*:

Figure 6.5 – NumMinObj = 100 model performance

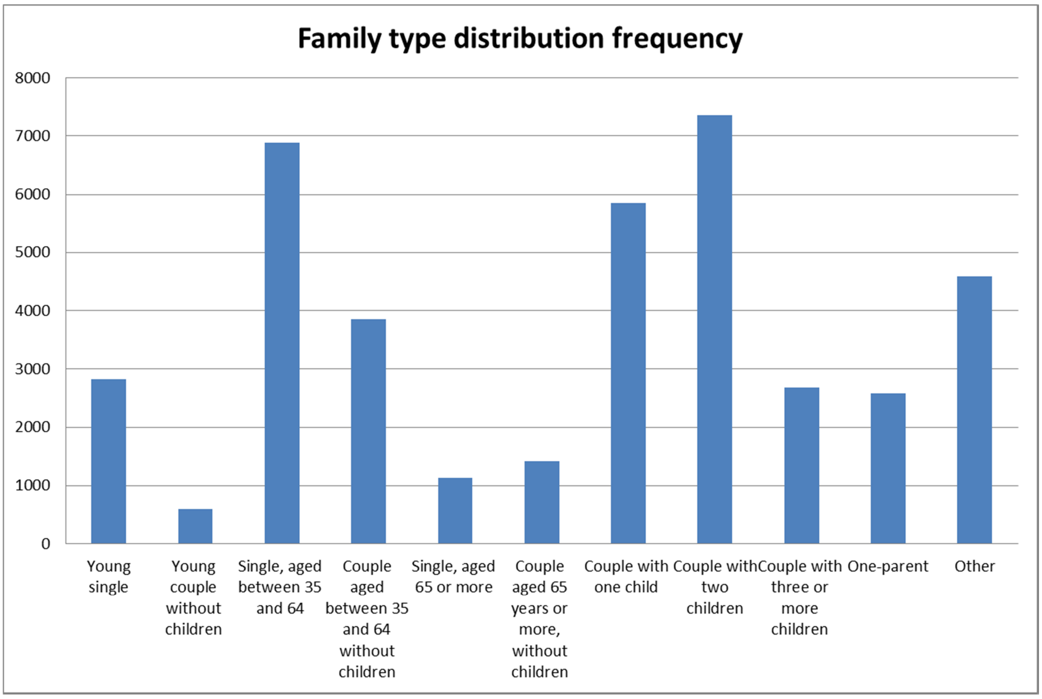
```
Correctly Classified Instances       10624               80.163  %
Incorrectly Classified Instances      2629               19.837  %
Kappa statistic                          0.2067
Mean absolute error                      0.3294
Root mean squared error                  0.3902
Coverage of cases (0.95 level)         100      %
Total Number of Instances            13253

=== Detailed Accuracy By Class ===
              TP Rate   FP Rate  Precision  Recall  F-Measure  ROC Area  Class
               0.198     0.041    0.559     0.198     0.293     0.69     INTERESTING
               0.959     0.802    0.821     0.959     0.885     0.69     NOT_INTERESTING
Weighted Avg.  0.802     0.644    0.767     0.802     0.762     0.69

=== Confusion Matrix ===
     a     b   <-- classified as
   544  2200 |     a = INTERESTING
   429 10080 |     b = NOT_INTERESTING
```

When predictive analyses concern individuals, their family type is often taken into account. Our dataset is no exception, and, for each taxpayer, this attribute has assumed one of the following values:

- `Couple aged 65 years or more, without children`
- `Single, aged between 35 and 64`
- `Couple with three or more children`
- `Couple with one child`
- `Couple with two children`
- `One-parent`
- `Couple aged between 35 and 64 without children`
- `Young single`
- `Single, aged 65 or more`
- `Other`
- `Young couple without children`

The frequency distribution of this attribute, in the entire dataset, is depicted in *Figure 6.6*:

Figure 6.6 – Frequency distribution of family type



As in the case of the age attribute, the taxpayer family type doesn’t seem to be particularly predictive, appearing in only 5 trees out of 25, only if the *numMinObj* parameter is set to 100, and often as one of the last attributes, near to the leaves (hence appearing only after other variables had already guided the most important classification choices).

However, this attribute often appears in paths in which the following conditions are satisfied:

$selection\ value > 20.000;$

$declared\ income < x$

$motor\ vehicle\ expense < y$

In trees having a *numMinObj* parameter set to 500, these clauses would have led directly to a “not interesting” outcome, while in these deeper ones, according to the family type value, the outcome is diversified as follows:

```
FAMILY_TYPE = Couple with one child: INTERESTING
FAMILY_TYPE = Couple with three or more children: INTERESTING
FAMILY_TYPE = Couple with two children: NOT_INTERESTING
FAMILY_TYPE = One-parent: INTERESTING
FAMILY_TYPE = Couple aged between 35 and 64 without children: NOT_INTERESTING
FAMILY_TYPE = Young couple without children: NOT_INTERESTING
FAMILY_TYPE = Couple aged 65 years or more, without children: NOT_INTERESTING
FAMILY_TYPE = Other: NOT_INTERESTING
FAMILY_TYPE = Young single: NOT_INTERESTING
FAMILY_TYPE = Couple aged 65 years or more, without children: NOT_INTERESTING
```

We can point out that, on average, having one or more children slightly increases the probability of being considered *interesting*. After all, since this attribute appears when incurred expenses are quite high and declared income quite low, and considering that children often involve significant expenditure, these results seem to be consistent.

In conclusion, even though attributes such as age and family type might be useful when trying to classify a taxpayer, they come out, however, only in complex models, more likely to suffer from overfitting, and not more predictive than simpler ones. As stated earlier, we won’t rely on them.

7. Deviation, score and tax claim

The VERDI application computes, among the others, two variables, for each taxpayer: *deviation* (i.e. difference between estimated and declared income) and *score* (which represents a sort of risk index of the taxpayer).

These variables have been ignored by all learning schemes we have built so far. However, it's interesting to see how they are related to the tax claim, using a linear regression model in which tax claim is the dependent variable and the independent variables are, in one case, *deviation* and in the other *score*.

It turns out that tax claim is hardly related to both deviation and score (correlation coefficient *R*equals to 0.06 when tax claim and score are considered, and to 0.20 in the other case), as the following charts clearly show:

Figure 7.1 – Linear regression: Tax claim on deviation

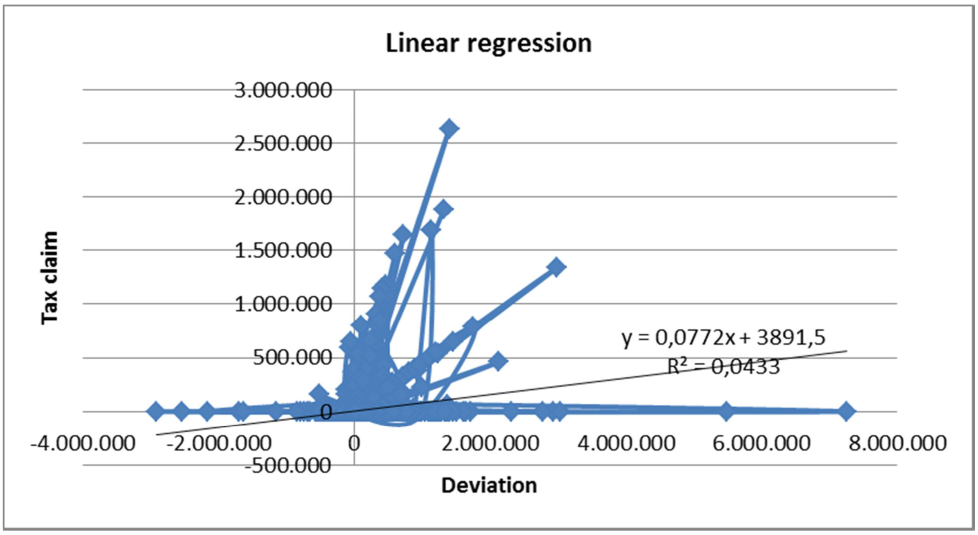
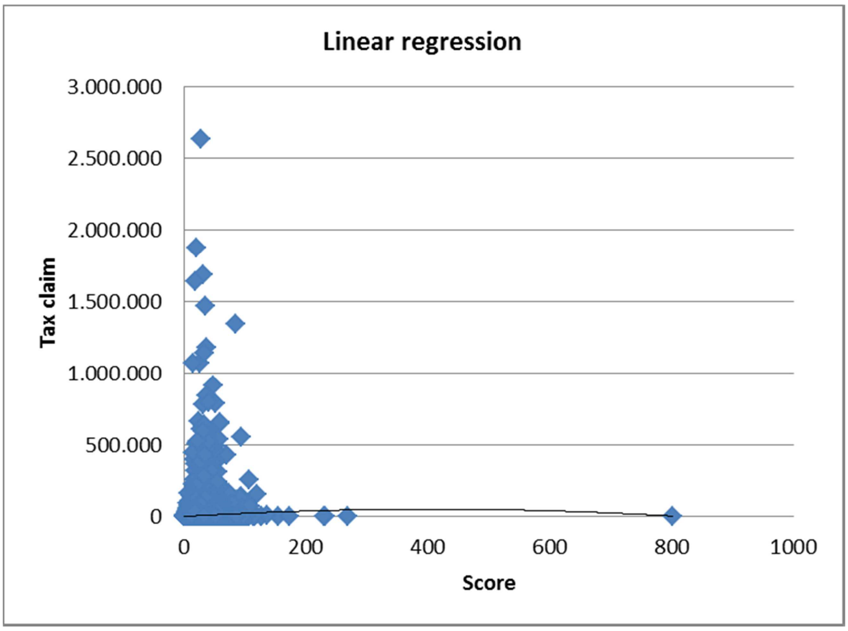


Figure 7.2 – Linear regression: Tax claim on score



These findings are interesting because they show that two variables such as score and deviation, which we could have expected to be strongly correlated to tax claim, are actually not so.

Therefore, the entire selection process should not rely on them, being two a-priori built coefficients, even if they were actually computed to help tax offices in their screening operations.

8. Concluding remarks and first operative guidelines

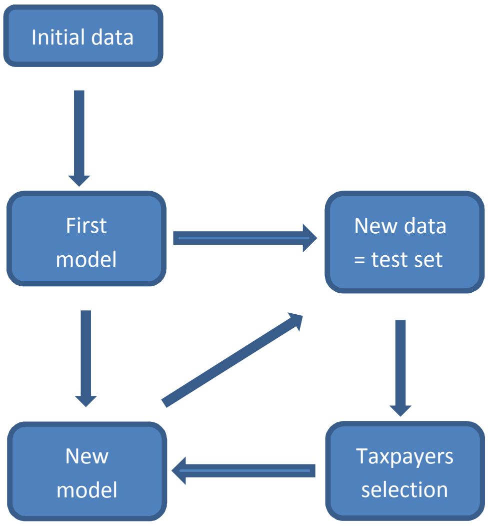
This paper gives three contributions.

First, the developed learning scheme methodology can effectively be used for filtering possible non–compliant taxpayers in the context of income indicators audits. Instead of relying on manual methods, on personal judgements or on a-priori built variables in selecting suspicious taxpayers, tax authorities may take advantage of a data mining tool to perform the same tasks, with both higher expected positivity rates and average tax claims.

Second, the proposed methodology is intended to stimulate reflection and action, by providing a guideline on the use of some well-known data mining algorithms. So, further analyses could be carried out, which may also improve presented results. Indeed, many parameters may be set in a different way, outliers could be defined differently, models built in different scenarios could be merged together according to some criteria, and so on.

Third, models applied to real data have identified some hidden patterns and significant features of illegal taxpayers. Thus, tax offices could successfully combine data mining methods with their professional experience to detect further cases of tax evasion. This would be desirable, considering that the proposed methodology is, in a sense, a dynamic process. Indeed, once a benchmark model has been set, it has to be employed on new, real data. And, more important, on the basis of new data, it has to produce new updated models, if necessary. To this purpose, a virtuous process, represented as follows, should be triggered:

Figure 8.1 – Data mining selection process



Basically, starting from data concerning an already concluded audit activity (such as the activity being carried out in the years 2014-2015), an initial learning scheme is built (which is what has been done so far).

Then, when new data is available, it will be used as a test set for the just built model, and some taxpayers will be selected accordingly. The audit activity will then be carried out, fully or partially, according to the model recommendations, and its results will become available at some time.

Results will be then collected to form a new dataset upon which build a newer learning scheme (this new dataset may also have a different set of attributes) which will be used on data available immediately after its definition.

The process will start again, by defining a new list of taxpayers that tax offices will invite and check.

Finally, successful models should be incorporated into standard use applications (such as the VERDI application), and be subjected to (annual) review for re-evaluating their accuracy and performance.

The proposed methodology is currently being validated on real cases: a number of taxpayers have been selected on the basis of the classification criteria we have formulated, and actual audits will be performed in order to assess their predictive accuracy. At the writing of this paper, no results are yet available.

References

[1] **P. De Sisti, S. Pisani.** Data mining e analisi del rischio di frode fiscale: il caso dei crediti d'imposta. Documenti di lavoro dell'Ufficio Studi – Agenzia delle Entrate, n. 4, 2007

[2] **S. Basta, F. Fassetti, M. Guarascio, G. Manco, F. Giannotti, D. Pedreschi, L. Spinsanti, G. Papi, S. Pisani.** High quality true positive prediction for fiscal fraud detection. 2009 IEEE International Conference on Data Mining Workshops

[3] **D. DeBarr, M. Harwood.** Relational Mining for Compliance Risk. Proceedings of the 2004 IRS Research Conference. The IRS Research Bulletin: Recent IRS Research on Tax Administration and Compliance, Publication 1500, pp. 175-186

[4] **F. Tian, T. Lan, K. Chao, N. Godwin, Q, Zheng, N. Shah, F. Zhang.** Mining suspicious tax evasion in big data. IEEE Transactions on knowledge and data engineering. October 2016

[5] **S. Xu.** An application on association rules data mining in the tax audit system. Economic supervision, vol. 13, pp. 43-44, November 2011

[6] **P.C. Gonzalez, J.D. Velasquez.** Characterization and detection of taxpayers with false invoices using data mining techniques. Expert systems with applications. Vol. 40, no. 5, pp. 1427-1436, April 2013

[7] **N. Goumagias, D. Hristu-Varsakelis, a. Saraidaris.** A decision support model for tax revenue collection in Greece. Decision support systems. Vol. 53, no. 1, pp. 76-96, April 2012

[8] **F. Ameur, M. Tkiouat.** Taxpayers fraudulent behavior Modeling the use of data mining in fiscal fraud detecting Moroccan case. Applied mathematics, no. 3, pp. 1207-1213, October 2012

[9] **R. Wu, C.S. Ou, H. Lin, S. Chang, D. Yen.** Using data mining technique to enhance tax evasion detection performance. Expert systems with applications, no. 39, pp. 8769-8777, 2012

[10] **K. Sutha, J.J. Tamilselvi.** A Review of Feature Selection Algorithms for Data Mining Techniques. International Journal on Computer Science and Engineering, Vol. 7 No.6 pp. 63-67, June 2015

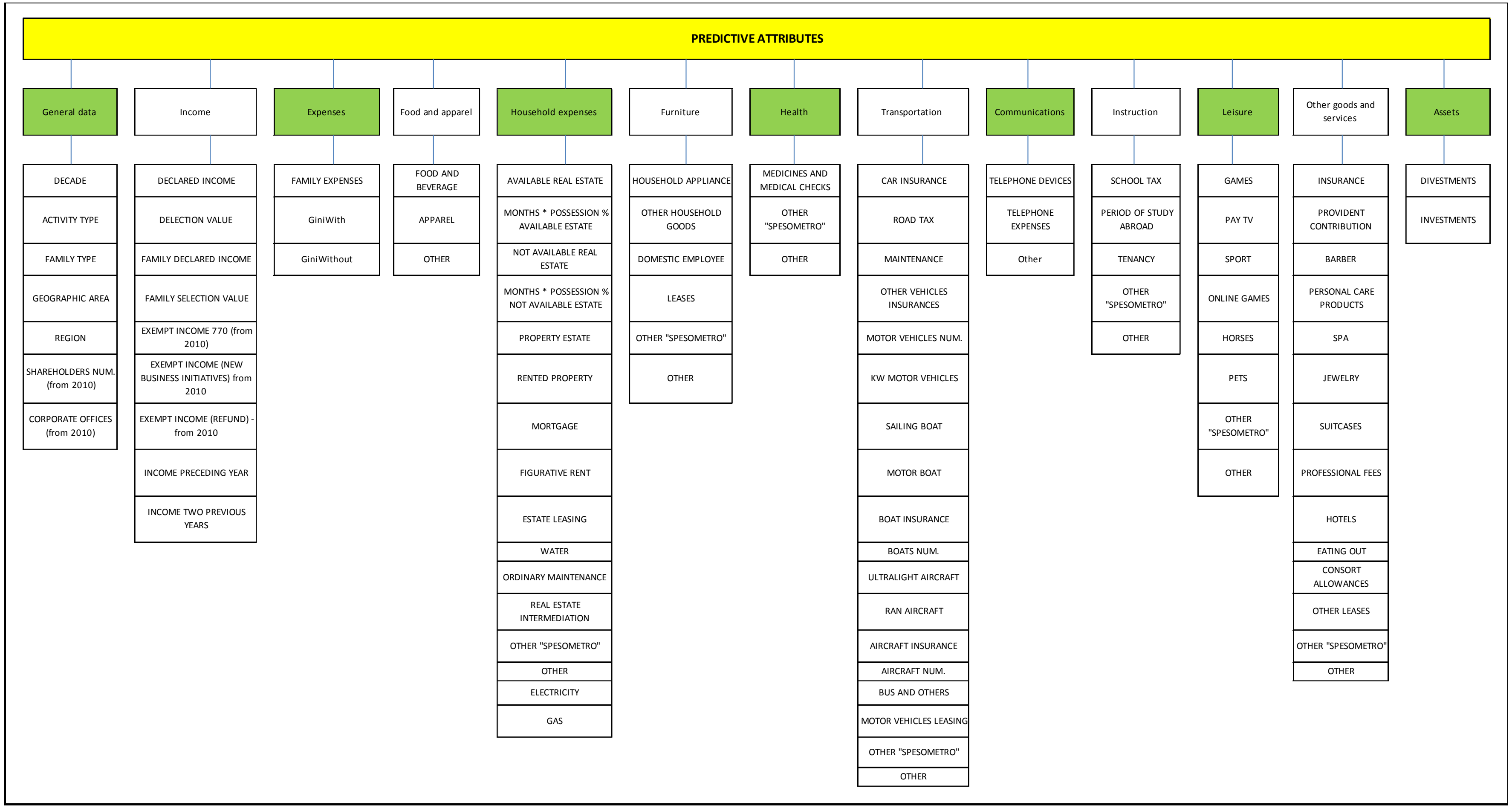
[11] **M. A. Hall.** Correlation-based Feature Subset Selection for Machine Learning. Hamilton, New Zealand, 1998

[12] **J. Novakovic.** Using Information Gain Attribute Evaluation to Classify Sonar Targets, 17th Telecommunications forum TELFOR, Belgrade, Serbia, 2009

[13] **Witten, Frank, Hall,** "*Data Mining, practical machine learning tools and techniques*", 3rd Edition, Morgan Kaufmann

[14] **L. Breiman.** Bagging predictors. Machine Learning, Vol. 24, no. 2, pp. 123-140, 1996

Appendix 1: Predictive attributes of the dataset



PREDICTIVE ATTRIBUTES

- General data
  - DECADE
  - ACTIVITY TYPE
  - FAMILY TYPE
  - GEOGRAPHIC AREA
  - REGION
  - SHAREHOLDERS NUM. (from 2010)
  - CORPORATE OFFICES (from 2010)
- Income
  - DECLARED INCOME
  - DELECTION VALUE
  - FAMILY DECLARED INCOME
  - FAMILY SELECTION VALUE
  - EXEMPT INCOME 770 (from 2010)
  - EXEMPT INCOME (NEW BUSINESS INITIATIVES) from 2010
  - EXEMPT INCOME (REFUND) - from 2010
  - INCOME PRECEDING YEAR
  - INCOME TWO PREVIOUS YEARS
- Expenses
  - FAMILY EXPENSES
  - GiniWith
  - GiniWithout
- Food and apparel
  - FOOD AND BEVERAGE
  - APPAREL
  - OTHER
- Household expenses
  - AVAILABLE REAL ESTATE
  - MONTHS * POSSESSION % AVAILABLE ESTATE
  - NOT AVAILABLE REAL ESTATE
  - MONTHS * POSSESSION % NOT AVAILABLE ESTATE
  - PROPERTY ESTATE
  - RENTED PROPERTY
  - MORTGAGE
  - FIGURATIVE RENT
  - ESTATE LEASING
  - WATER
  - ORDINARY MAINTENANCE
  - REAL ESTATE INTERMEDIATION
  - OTHER "SPESOMETRO"
  - OTHER
  - ELECTRICITY
  - GAS
- Furniture
  - HOUSEHOLD APPLIANCE
  - OTHER HOUSEHOLD GOODS
  - DOMESTIC EMPLOYEE
  - LEASES
  - OTHER "SPESOMETRO"
  - OTHER
- Health
  - MEDICINES AND MEDICAL CHECKS
  - OTHER "SPESOMETRO"
  - OTHER
- Transportation
  - CAR INSURANCE
  - ROAD TAX
  - MAINTENANCE
  - OTHER VEHICLES INSURANCES
  - MOTOR VEHICLES NUM.
  - KW MOTOR VEHICLES
  - SAILING BOAT
  - MOTOR BOAT
  - BOAT INSURANCE
  - BOATS NUM.
  - ULTRALIGHT AIRCRAFT
  - RAN AIRCRAFT
  - AIRCRAFT INSURANCE
  - AIRCRAFT NUM.
  - BUS AND OTHERS
  - MOTOR VEHICLES LEASING
  - OTHER "SPESOMETRO"
  - OTHER
- Communications
  - TELEPHONE DEVICES
  - TELEPHONE EXPENSES
  - Other
- Instruction
  - SCHOOL TAX
  - PERIOD OF STUDY ABROAD
  - TENANCY
  - OTHER "SPESOMETRO"
  - OTHER
- Leisure
  - GAMES
  - PAY TV
  - SPORT
  - ONLINE GAMES
  - HORSES
  - PETS
  - OTHER "SPESOMETRO"
  - OTHER
- Other goods and services
  - INSURANCE
  - PROVIDENT CONTRIBUTION
  - BARBER
  - PERSONAL CARE PRODUCTS
  - SPA
  - JEWELRY
  - SUITCASES
  - PROFESSIONAL FEES
  - HOTELS
  - EATING OUT
  - CONSORT ALLOWANCES
  - OTHER LEASES
  - OTHER "SPESOMETRO"
  - OTHER
- Assets
  - DIVESTMENTS
  - INVESTMENTS

Data type	Name	Description	Data Origin
GENERAL DATA	DECADE	Taxpayer age grouped as follows: [0-30], [31-40], [41-50] and so on.	Computed
	ACTIVITY FAMILY	Information available only if the taxpayer is entrepreneur or professional	Computed
	FAMILY TYPE	01 Young single 02 Young couple without children 03 Single, aged between 35 and 64 04 Couple aged between 35 and 64 without children 05 Single, aged 65 or more 06 Couple aged 65 years or more, without children 07 Couple with one child 08 Couple with two children 09 Couple with three or more children 10 One-parent 11 Other	TAXPAYERS database
	GEOGRAPHIC AREA	Region Groups (NORTH-WEST, NORTH-EAST, CENTER, SOUTH, SICILIA+SARDEGNA)	TAXPAYERS database
	REGION		TAXPAYERS database
	SHAREHOLDERS NUM. (from 2010)		TAXPAYERS database
	CORPORATE OFFICES (from 2010)		TAXPAYERS database
INCOME	DECLARED INCOME		TAXPAYERS database
	SELECTION VALUE	Estimated income, equal to sum of incurred expenses	TAXPAYERS database
	FAMILY DECLARED INCOME		TAXPAYERS database
	FAMILY SELECTION VALUE	Family estimated income, equal to sum of incurred expenses	TAXPAYERS database
	EXEMPT INCOME 770 (from 2010)		TAXPAYERS database
	EXEMPT INCOME (NEW BUSINESS INITIATIVES) from 2010		TAXPAYERS database
	EXEMPT INCOME (REFUND) - from 2010		TAXPAYERS database
	INCOME PRECEDING YEAR		TAXPAYERS database
	INCOME TWO PREVIOUS YEARS		TAXPAYERS database
EXPENSES	FAMILY EXPENSES (sure and for sure items)	Family expenses sum	TAXPAYERS database
	GINI	GINI Coefficient on each taxpayer's expenses (with and without Investments/Divestments)	Computed
FOOD AND APPAREL	FOOD AND BEVERAGE	Data taken from "spesometro", starting from 2010	TAXPAYERS database
	APPAREL		TAXPAYERS database
	OTHER		TAXPAYERS database
HOUSEHOLD EXPENSES	AVAILABLE REAL ESTATE		TAXPAYERS database
	MONTHS * POSSESSION % AVAILABLE ESTATE		Computed
	NOT AVAILABLE REAL ESTATE		TAXPAYERS database

Data type	Name	Description	Data Origin
HOUSEHOLD EXPENSES	MONTHS * POSSESSION % NOT AVAILABLE ESTATE		Computed
	PROPERTY ESTATE		TAXPAYERS database
	RENTED PROPERTY	Data taken from Land Registry	TAXPAYERS database
	MORTGAGE	Since 2010 data transmitted by financial companies	TAXPAYERS database
	FIGURATIVE RENT		TAXPAYERS database
	ESTATELEASING	Data transmitted by leasing companies	TAXPAYERS database
	WATER	Expenses referred to available real estate	TAXPAYERS database
	ORDINARY MAINTENANCE	Expenses referred to available real estate	TAXPAYERS database
	REAL ESTATE INTERMEDIATION	Data transmitted in income declarations	TAXPAYERS database
	OTHER "SPESOMETRO"		TAXPAYERS database
	OTHER		TAXPAYERS database
	ELECTRICITY	Data transmitted by utility companies	TAXPAYERS database
	GAS	Data transmitted by utility companies	TAXPAYERS database
FURNITURE	HOUSEHOLD APPLIANCE		TAXPAYERS database
	OTHER HOUSEHOLD GOODS		TAXPAYERS database
	DOMESTIC EMPLOYEE	Data transmitted by INPS	TAXPAYERS database
	LEASES	Data transmitted by leasing companies	TAXPAYERS database
	OTHER "SPESOMETRO"		TAXPAYERS database
	OTHER		TAXPAYERS database
HEALTH	MEDICINES AND MEDICAL CHECKS	Data transmitted in income declarations	TAXPAYERS database
	OTHER "SPESOMETRO"		TAXPAYERS database
	OTHER		TAXPAYERS database
TRANSPORTATION	CAR INSURANCE	Data transmitted by insurance companies	TAXPAYERS database
	ROAD TAX	Data transmitted by ACI	TAXPAYERS database
	MAINTENANCE	Estimated data	TAXPAYERS database
	OTHER VEHICLES INSURANCES	Data transmitted by insurance companies	TAXPAYERS database
	MOTOR VEHICLES NUM.	Data transmitted by PRA/Motorizzazione	TAXPAYERS database
	KW MOTOR VEHICLES	Data transmitted by PRA/Motorizzazione	TAXPAYERS database
	SAIL BOAT		TAXPAYERS database
	MOTOR BOAT		TAXPAYERS database
	BOAT INSURANCE		TAXPAYERS database
	BOATS NUM.	Data transmitted by Coast Guard	TAXPAYERS database
	ULTRALIGHT AIRCRAFT		TAXPAYERS database
	RAN AIRCRAFT		TAXPAYERS database
	AIRCRAFT INSURANCE		TAXPAYERS database
	AIRCRAFT NUM.	Data transmitted by ENAC	TAXPAYERS database
	BUS AND OTHERS		TAXPAYERS database
	MOTOR VEHICLES LEASING	Data transmitted by leasing companies	TAXPAYERS database
	OTHER "SPESOMETRO"		TAXPAYERS database
	OTHER		TAXPAYERS database
COMMUNICATIONS	TELEPHONE DEVICES	Purchaseexpense (“spesometro”)	TAXPAYERS database
	TELEPHONE EXPENSES	Data transmitted by telephone companies	TAXPAYERS database
	OTHER		TAXPAYERS database

Data type	Name	Description	Data Origin
INSTRUCTION	SCHOOL TAX	Data transmitted in income declarations	TAXPAYERS database
	PERIOD OF STUDY ABROAD		TAXPAYERS database
	TENANCY	Data transmitted in income declarations	TAXPAYERS database
	OTHER "SPESOMETRO"		TAXPAYERS database
	OTHER		TAXPAYERS database
LEISURE	GAMES		TAXPAYERS database
	PAY TV		TAXPAYERS database
	SPORT	Data transmitted in income declarations	TAXPAYERS database
	ONLINE GAMES		TAXPAYERS database
	HORSES		TAXPAYERS database
	PETS	Data transmitted in income declarations	TAXPAYERS database
	OTHER "SPESOMETRO"		TAXPAYERS database
	OTHER		TAXPAYERS database
OTHER GOODS AND SERVICES	INSURANCE	Data taken from insurance contracts	TAXPAYERS database
	PROVIDENT CONTRIBUTION	Data taken from INPS or from income declaration	TAXPAYERS database
	BARBER	Data taken from Spesometro (from 2010)	TAXPAYERS database
	PERSONAL CARE PRODUCTS	Data taken from Spesometro (from 2010)	TAXPAYERS database
	SPA	Data taken from Spesometro (from 2010)	TAXPAYERS database
	JEWELRY	Data taken from Spesometro (from 2010)	TAXPAYERS database
	SUITCASES	Data taken from Spesometro (from 2010)	TAXPAYERS database
	PROFESSIONAL FEES	Data taken from Spesometro (from 2010)	TAXPAYERS database
	HOTELS	Data taken from “financial movements”	TAXPAYERS database
	EATING OUT	Data taken from “financial movements”	TAXPAYERS database
	CONSORT ALLOWANCES	Data taken from “financial movements”	TAXPAYERS database
	OTHER LEASES	Data taken from leasing companies	TAXPAYERS database
	OTHER "SPESOMETRO"		TAXPAYERS database
	OTHER		TAXPAYERS database
ASSETS	DIVESTMENTS	Divestments sum.	TAXPAYERS database
	INVESTMENTS	Investments sum	TAXPAYERS database

Appendix 2: Standard evaluation metrics

In binary class classification issues, a class can always be labeled as positive (in this paper, the interesting label) and the other as negative (not interesting). A test set consists of P positive records and N negative records (the following measures only make sense once we’ve decided which one is the positive class and which one is the negative class). A classifier assigns a class label to each of them, but some of these assignments will be, inevitably, wrong. To evaluate classifications results, we count how many true positive (TP), true negative (TN), false positive (FP) – truly negative records classified as positive and false negative (FN) – truly positive records classified as negative.

A confusion matrix can be defined as follows:

$$\text{Predicted class}$$

$$\text{Actual class}\begin{bmatrix} & \mathrm{C}_1 & \mathrm{C}_2 & \\ \mathrm{C}_1 & \mathrm{TP} & \mathrm{FN} & \mathrm{P} \\ \mathrm{C}_2 & \mathrm{FP} & \mathrm{TN} & \mathrm{N} \end{bmatrix}$$

These identities hold: $\mathrm{TP} + \mathrm{FN} = \mathrm{Pe}\ \mathrm{TN} + \mathrm{FP} = \mathrm{N}$

A classifier assigns $\mathrm{TP} + \mathrm{FP}$ recordsto the positive class and $\mathrm{TN} + \mathrm{FN}$records to the negative one.

Given a confusion matrix, some evaluation measures are defined as follows:

$$\mathrm{FP\ rate} = \ \mathrm{FP}/\mathrm{N}$$

$$\mathrm{TP\ rate} = \mathrm{TP}/\mathrm{P} = \mathrm{recall} = \mathrm{sensitivity}$$

$$\mathrm{TN\ rate} = \mathrm{TN}/(\mathrm{TN} + \mathrm{FP}) = \mathrm{specificity}$$

$$\mathrm{Y\ rate} = (\mathrm{TP} + \mathrm{FP})/(\mathrm{P} + \mathrm{N})$$

$$\mathrm{Precision} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$$

$$\mathrm{Accuracy} = (\mathrm{TP} + \mathrm{TN})/(\mathrm{P} + \mathrm{N})$$

$$\mathrm{F} - \mathrm{measure} = 2\mathrm{pr}/(\mathrm{r} + \mathrm{p})$$

Precision and accuracy are widely employed to measure binary classifiers predictions quality.

F-measure summarizes precision and recall, being their harmonic mean. Since the harmonic mean between two numbers x and y tends to be near to the smallest one, if it is high, both precision and recall are as well, thus we have only a few FP and FN.

Appendix 3: Best Decision Trees

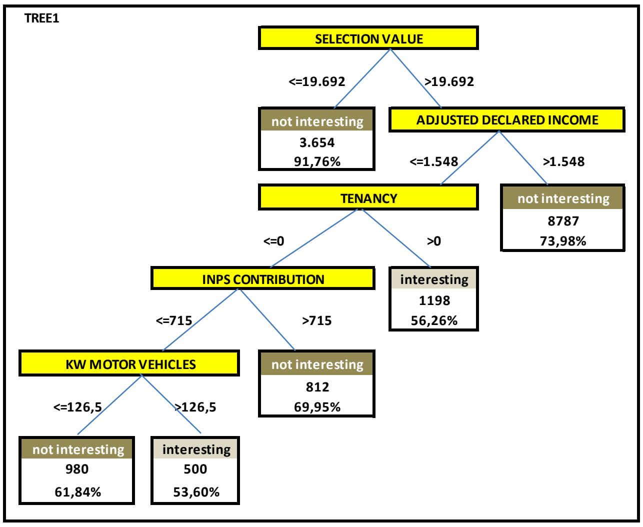
We recall that the best model was built in the context of Scenario 2, without outliers (recalling that, in this model, outliers differ from the average value by more than 6 standard deviations) and with a tax claim threshold set at € 5,000.

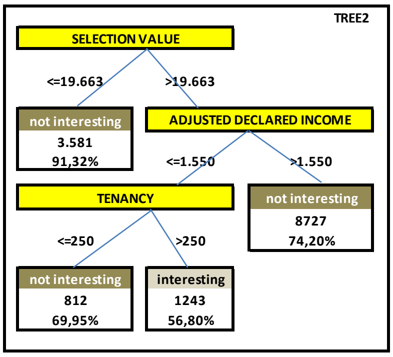
We now show the 25 individual classifiers generated in the bagging process.

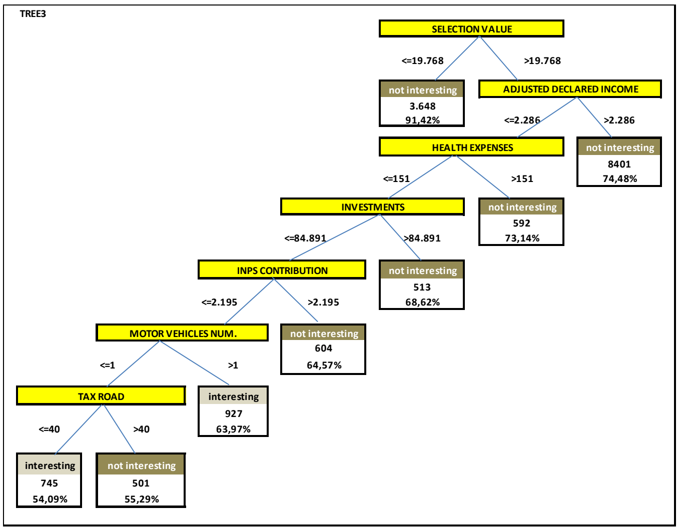
Take, for instance, tree n.1. It has to be read starting from the top, i.e. from the root node, which happens to be “selection value”. So, if a given record “selection value” is lower than € 19,692, this tree would classify it as a “not interesting” record. In such a case, we would have found a leaf node, a labeled node showing the class the analyzed record belongs to. Otherwise, if the attribute “selection value” is greater than € 19,692, we would go forward to the next node, which is “adjusted declared income”. So, we would not find a leaf node, but an internal one. Two edges depart from this node: according to the record’s “adjusted declared income” value, the first one or the second is taken. And so on, until a leaf node is met.

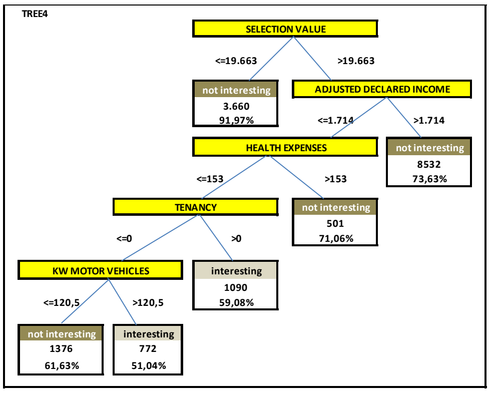
Each classifier (tree) classifies a record as interesting or not interesting according to the leaf node label.

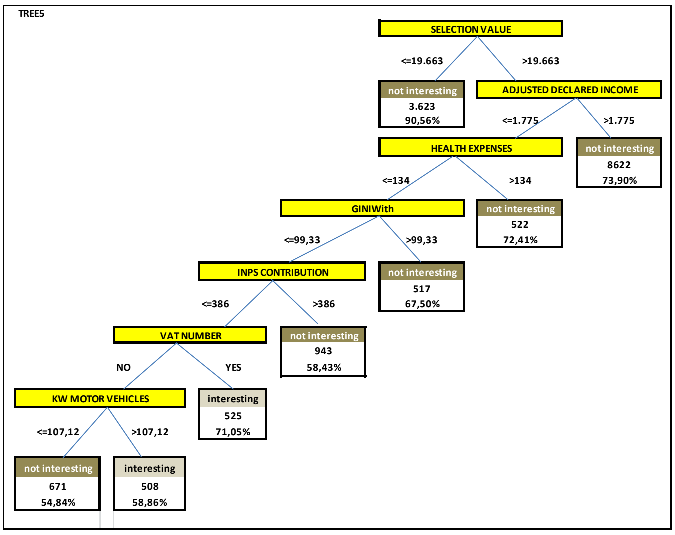
Then, individual classifiers are combined on a vote basis, i.e. a certain record is assigned to the class label that was predicted more often.

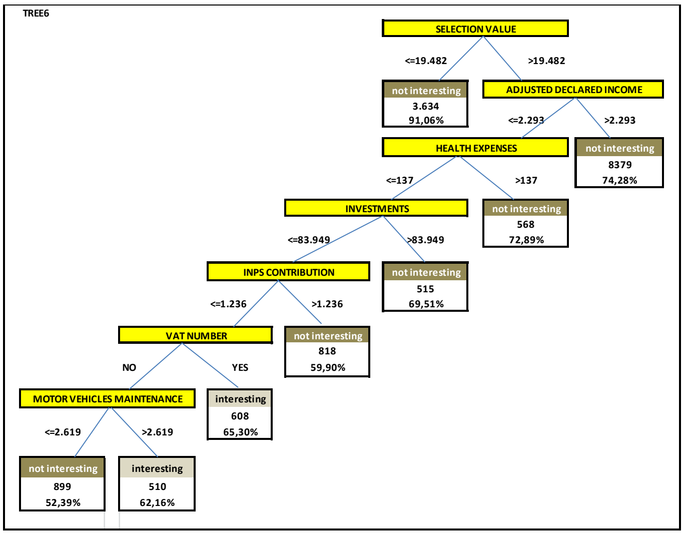


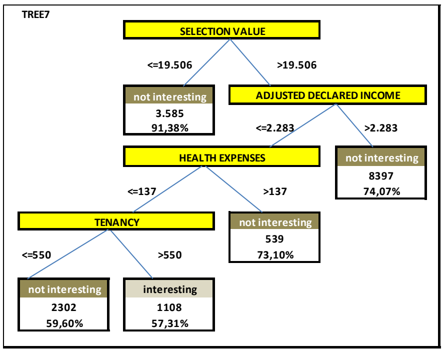


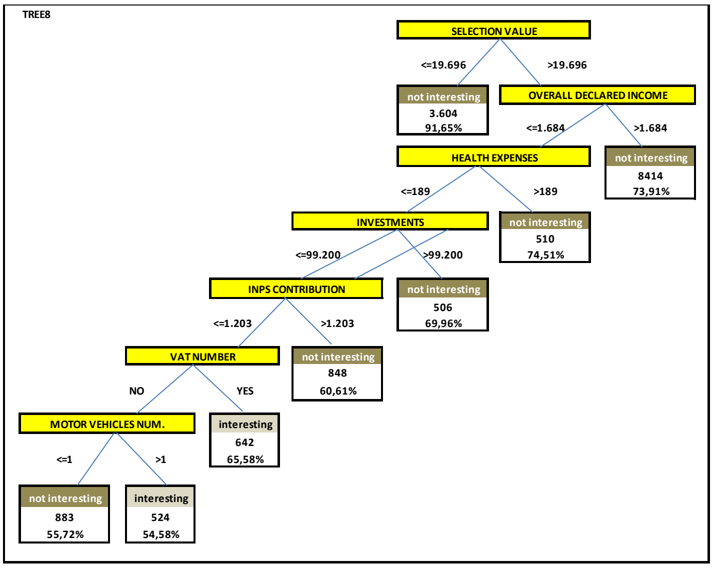


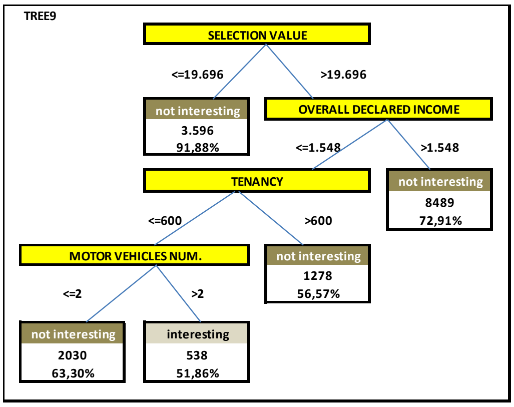


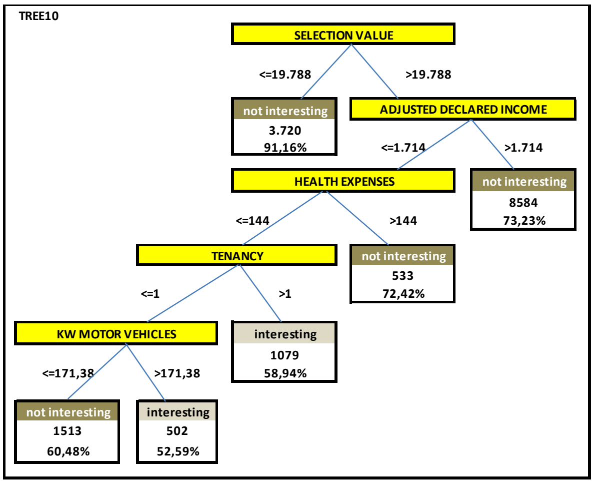

TREE5
SELECTION VALUE
<=19.663
>19.663
not interesting
3.623
90,56%
ADJUSTED DECLARED INCOME
<=1.775
>1.775
HEALTH EXPENSES
not interesting
8622
73,90%
<=134
>134
GINIWith
not interesting
522
72,41%
<=99,33
>99,33
INPS CONTRIBUTION
not interesting
517
67,50%
<=386
>386
VAT NUMBER
not interesting
943
58,43%
NO
YES
KW MOTOR VEHICLES
interesting
525
71,05%
<=107,12
>107,12
not interesting
671
54,84%
interesting
508
58,86%

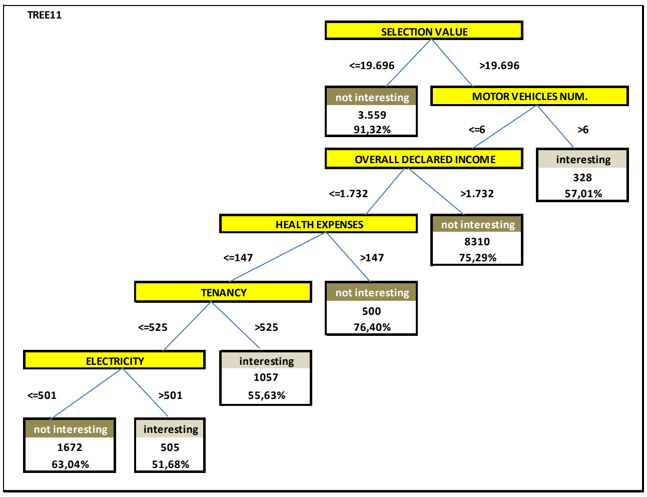

TREE6
SELECTION VALUE
<=19.482
>19.482
not interesting
3.634
91,06%
ADJUSTED DECLARED INCOME
<=2.293
>2.293
HEALTH EXPENSES
not interesting
8379
74,28%
<=137
>137
INVESTMENTS
not interesting
568
72,89%
<=83.949
>83.949
INPS CONTRIBUTION
not interesting
515
69,51%
<=1.236
>1.236
VAT NUMBER
not interesting
818
59,90%
NO
YES
MOTOR VEHICLES MAINTENANCE
interesting
608
65,30%
<=2.619
>2.619
not interesting
899
52,39%
interesting
510
62,16%

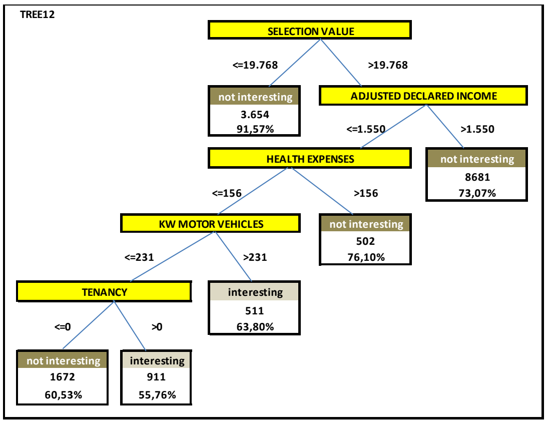

TREE7
SELECTION VALUE
<=19.506
>19.506
not interesting
3.585
91,38%
ADJUSTED DECLARED INCOME
<=2.283
>2.283
HEALTH EXPENSES
not interesting
8397
74,07%
<=137
>137
TENANCY
not interesting
539
73,10%
<=550
>550
not interesting
2302
59,60%
interesting
1108
57,31%

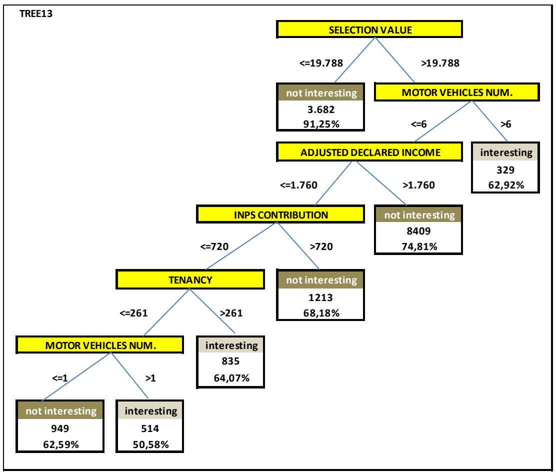

TREE8
SELECTION VALUE
<=19.696
>19.696
not interesting
3.604
91,65%
OVERALL DECLARED INCOME
<=1.684
>1.684
HEALTH EXPENSES
not interesting
8414
73,91%
<=189
>189
INVESTMENTS
not interesting
510
74,51%
<=99.200
>99.200
INPS CONTRIBUTION
not interesting
506
69,96%
<=1.203
>1.203
VAT NUMBER
not interesting
848
60,61%
NO
YES
MOTOR VEHICLES NUM.
interesting
642
65,58%
<=1
>1
not interesting
883
55,72%
interesting
524
54,58%

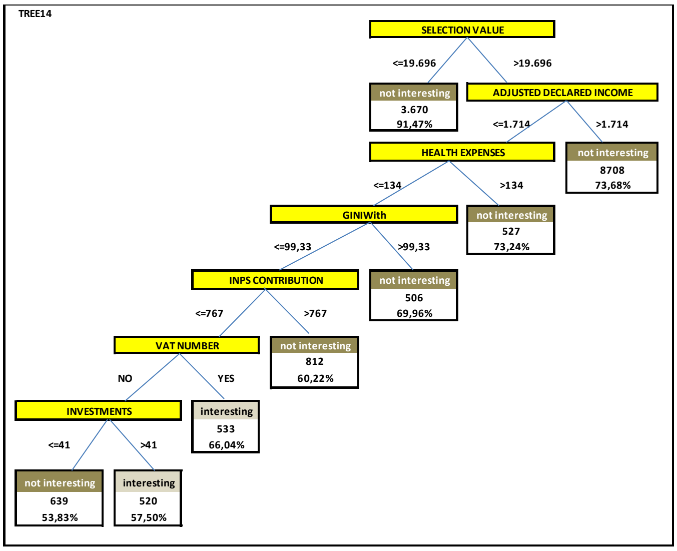

TREE9
SELECTION VALUE
<=19.696
>19.696
not interesting
3.596
91,88%
OVERALL DECLARED INCOME
<=1.548
>1.548
TENANCY
not interesting
8489
72,91%
<=600
>600
MOTOR VEHICLES NUM.
not interesting
1278
56,57%
<=2
>2
not interesting
2030
63,30%
interesting
538
51,86%

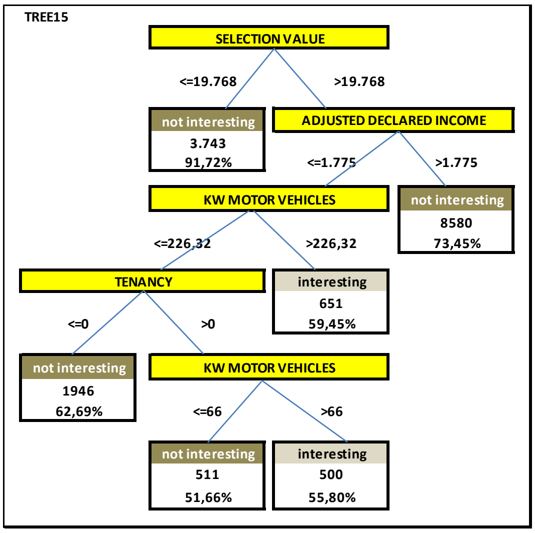

TREE10
SELECTION VALUE
<=19.788
>19.788
not interesting
3.720
91,16%
ADJUSTED DECLARED INCOME
<=1.714
>1.714
HEALTH EXPENSES
not interesting
8584
73,23%
<=144
>144
TENANCY
not interesting
533
72,42%
<=1
>1
KW MOTOR VEHICLES
interesting
1079
58,94%
<=171,38
>171,38
not interesting
1513
60,48%
interesting
502
52,59%

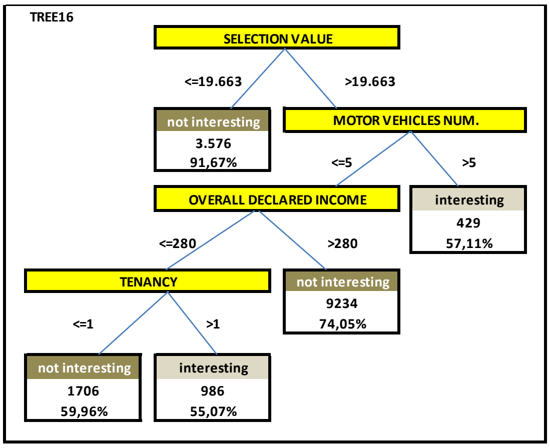

TREE11
SELECTION VALUE
<=19.696
>19.696
not interesting 3.559 91,32%
MOTOR VEHICLES NUM.
<=6
>6
OVERALL DECLARED INCOME
interesting 328 57,01%
<=1.732
>1.732
HEALTH EXPENSES
not interesting 8310 75,29%
<=147
>147
TENANCY
not interesting 500 76,40%
<=525
>525
ELECTRICITY
interesting 1057 55,63%
<=501
>501
not interesting 1672 63,04%
interesting 505 51,68%

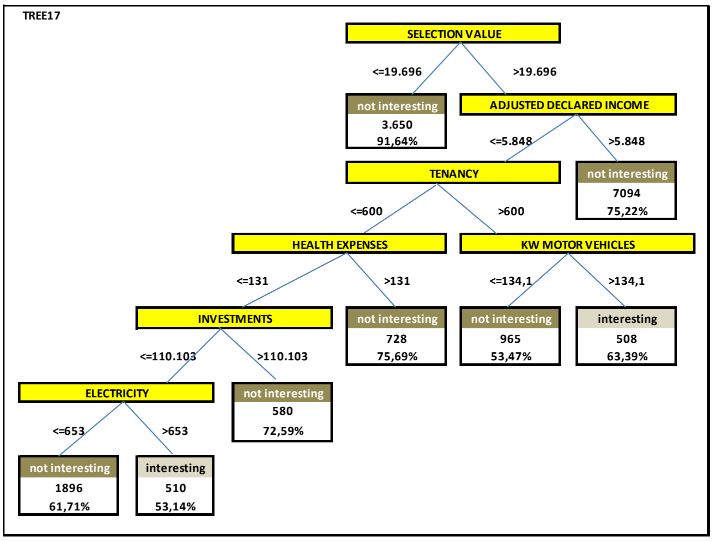

TREE12
SELECTION VALUE
<=19.768
>19.768
not interesting 3.654 91,57%
ADJUSTED DECLARED INCOME
<=1.550
>1.550
HEALTH EXPENSES
not interesting 8681 73,07%
<=156
>156
KW MOTOR VEHICLES
not interesting 502 76,10%
<=231
>231
TENANCY
interesting 511 63,80%
<=0
>0
not interesting 1672 60,53%
interesting 911 55,76%

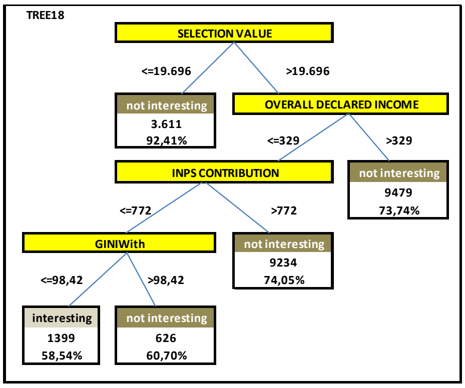

TREE13
SELECTION VALUE
<=19.788
>19.788
not interesting 3.682 91,25%
MOTOR VEHICLES NUM.
<=6
>6
ADJUSTED DECLARED INCOME
interesting 329 62,92%
<=1.760
>1.760
INPS CONTRIBUTION
not interesting 8409 74,81%
<=720
>720
TENANCY
not interesting 1213 68,18%
<=261
>261
MOTOR VEHICLES NUM.
interesting 835 64,07%
<=1
>1
not interesting 949 62,59%
interesting 514 50,58%

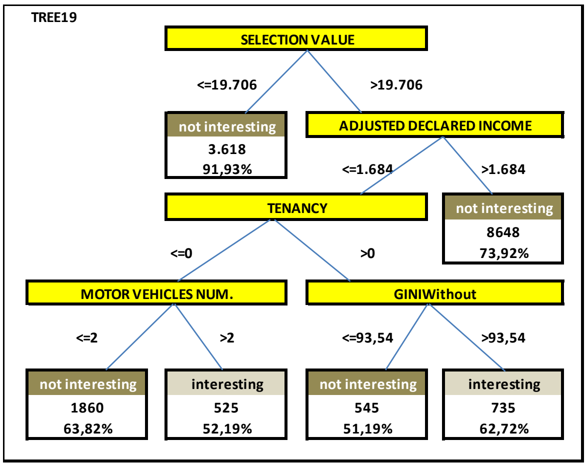

TREE14
SELECTION VALUE
<=19.696
>19.696
not interesting 3.670 91,47%
ADJUSTED DECLARED INCOME
<=1.714
>1.714
HEALTH EXPENSES
not interesting 8708 73,68%
<=134
>134
GINIWith
not interesting 527 73,24%
<=99,33
>99,33
INPS CONTRIBUTION
not interesting 506 69,96%
<=767
>767
VAT NUMBER
not interesting 812 60,22%
NO
YES
INVESTMENTS
interesting 533 66,04%
<=41
>41
not interesting 639 53,83%
interesting 520 57,50%

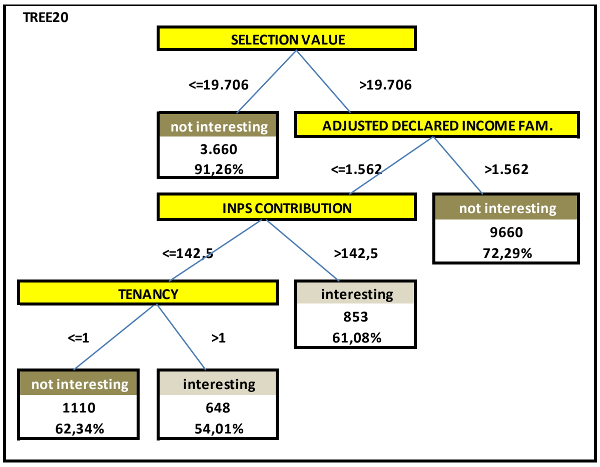

TREE15
SELECTION VALUE
<=19.768
>19.768
not interesting 3.743 91,72%
ADJUSTED DECLARED INCOME
<=1.775
>1.775
KW MOTOR VEHICLES
not interesting 8580 73,45%
<=226,32
>226,32
TENANCY
interesting 651 59,45%
<=0
>0
not interesting 1946 62,69%
KW MOTOR VEHICLES
<=66
>66
not interesting 511 51,66%
interesting 500 55,80%

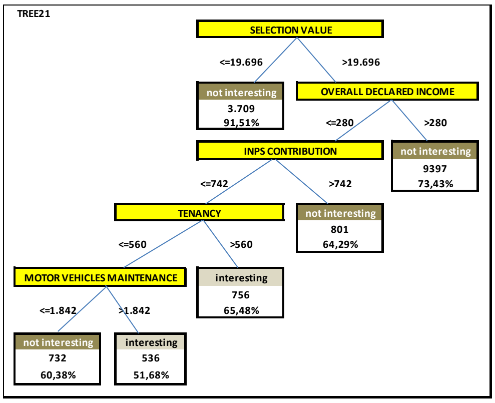

TREE16
SELECTION VALUE
<=19.663
>19.663
not interesting 3.576 91,67%
MOTOR VEHICLES NUM.
<=5
>5
OVERALL DECLARED INCOME
interesting 429 57,11%
<=280
>280
TENANCY
not interesting 9234 74,05%
<=1
>1
not interesting 1706 59,96%
interesting 986 55,07%

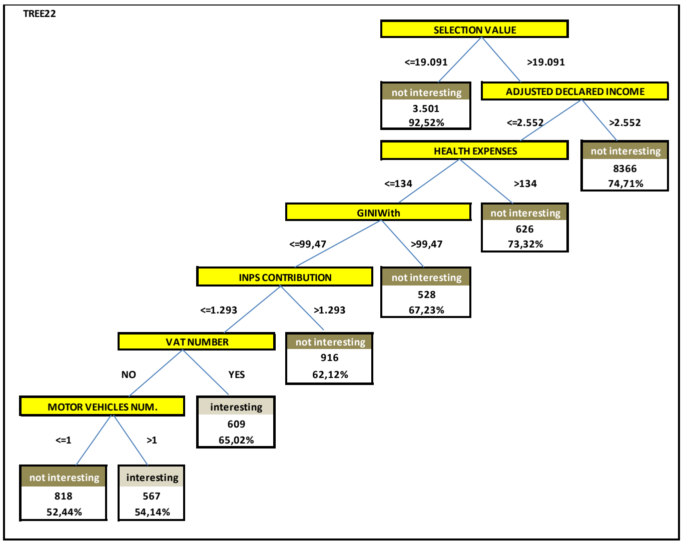

TREE17
SELECTION VALUE
<=19.696
>19.696
not interesting
3.650
91,64%
ADJUSTED DECLARED INCOME
<=5.848
>5.848
TENANCY
not interesting
7094
75,22%
<=600
>600
HEALTH EXPENSES
KW MOTOR VEHICLES
<=131
>131
<=134,1
>134,1
INVESTMENTS
not interesting
728
75,69%
not interesting
965
53,47%
interesting
508
63,39%
<=110.103
>110.103
ELECTRICITY
not interesting
580
72,59%
<=653
>653
not interesting
1896
61,71%
interesting
510
53,14%

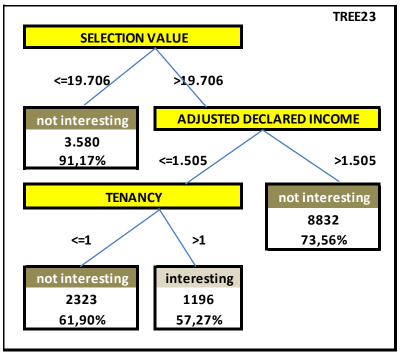

TREE18
SELECTION VALUE
<=19.696
>19.696
not interesting
3.611
92,41%
OVERALL DECLARED INCOME
<=329
>329
INPS CONTRIBUTION
not interesting
9479
73,74%
<=772
>772
GINIWith
not interesting
9234
74,05%
<=98,42
>98,42
interesting
1399
58,54%
not interesting
626
60,70%

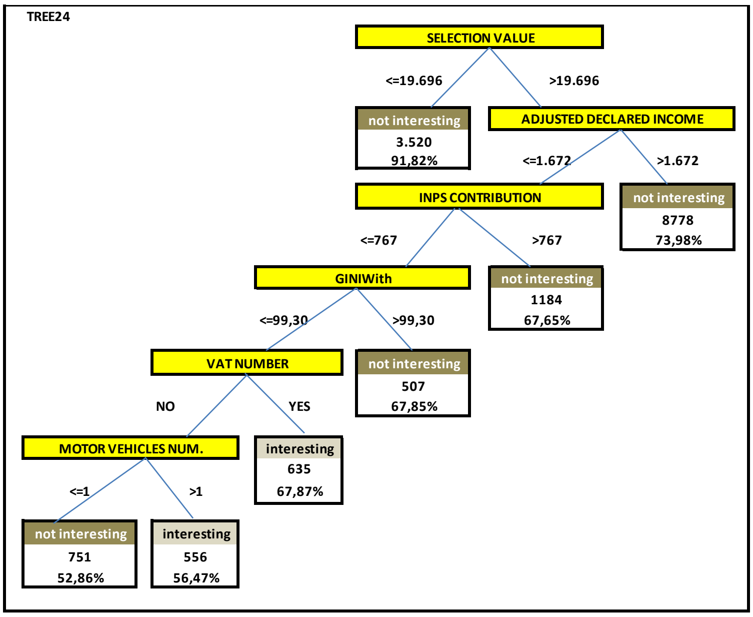

TREE19
SELECTION VALUE
<=19.706
>19.706
not interesting
3.618
91,93%
ADJUSTED DECLARED INCOME
<=1.684
>1.684
TENANCY
not interesting
8648
73,92%
<=0
>0
MOTOR VEHICLES NUM.
GINIWithout
<=2
>2
<=93,54
>93,54
not interesting
1860
63,82%
interesting
525
52,19%
not interesting
545
51,19%
interesting
735
62,72%

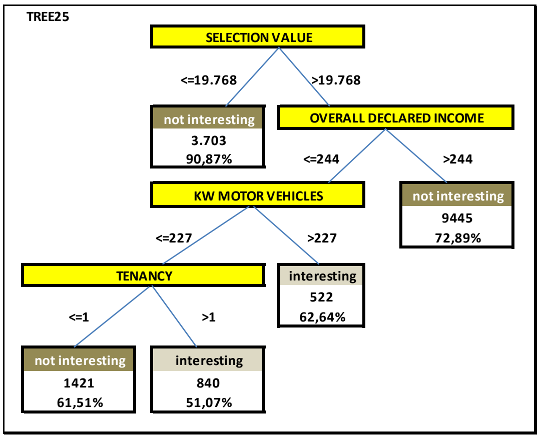

TREE20
SELECTION VALUE
<=19.706
>19.706
not interesting
3.660
91,26%
ADJUSTED DECLARED INCOME FAM.
<=1.562
>1.562
INPS CONTRIBUTION
not interesting
9660
72,29%
<=142,5
>142,5
TENANCY
interesting
853
61,08%
<=1
>1
not interesting
1110
62,34%
interesting
648
54,01%


TREE21
SELECTION VALUE
<=19.696
>19.696
not interesting
3.709
91,51%
OVERALL DECLARED INCOME
<=280
>280
INPS CONTRIBUTION
not interesting
9397
73,43%
<=742
>742
TENANCY
not interesting
801
64,29%
<=560
>560
MOTOR VEHICLES MAINTENANCE
interesting
756
65,48%
<=1.842
>1.842
not interesting
732
60,38%
interesting
536
51,68%


TREE22
SELECTION VALUE
<=19.091
>19.091
not interesting
3.501
92,52%
ADJUSTED DECLARED INCOME
<=2.552
>2.552
HEALTH EXPENSES
not interesting
8366
74,71%
<=134
>134
GINIWith
not interesting
626
73,32%
<=99,47
>99,47
INPS CONTRIBUTION
not interesting
528
67,23%
<=1.293
>1.293
VAT NUMBER
not interesting
916
62,12%
NO
YES
MOTOR VEHICLES NUM.
interesting
609
65,02%
<=1
>1
not interesting
818
52,44%
interesting
567
54,14%


TREE23
SELECTION VALUE
<=19.706
>19.706
not interesting
3.580
91,17%
ADJUSTED DECLARED INCOME
<=1.505
>1.505
TENANCY
not interesting
8832
73,56%
<=1
>1
not interesting
2323
61,90%
interesting
1196
57,27%


TREE24
SELECTION VALUE
<=19.696
>19.696
not interesting
3.520
91,82%
ADJUSTED DECLARED INCOME
<=1.672
>1.672
INPS CONTRIBUTION
not interesting
8778
73,98%
<=767
>767
GINIWith
not interesting
1184
67,65%
<=99,30
>99,30
VAT NUMBER
not interesting
507
67,85%
NO
YES
MOTOR VEHICLES NUM.
interesting
635
67,87%
<=1
>1
not interesting
751
52,86%
interesting
556
56,47%


TREE25
SELECTION VALUE
<=19.768
>19.768
not interesting
3.703
90,87%
OVERALL DECLARED INCOME
<=244
>244
KW MOTOR VEHICLES
not interesting
9445
72,89%
<=227
>227
TENANCY
interesting
522
62,64%
<=1
>1
not interesting
1421
61,51%
interesting
840
51,07%