Hernández Orallo, J.; Ferri Ramírez, C.; Lachiche, N.; Martínez Usó, A.; Ramírez Quintana, MJ. (2015). Binarised regression tasks: methods and evaluation metrics. Data Mining and Knowledge Discovery. 1-43. doi:10.1007/s10618-015-0443-9.

Noname manuscript No.
(will be inserted by the editor)

Binarised Regression Tasks: Methods and Evaluation Metrics

José Hernández-Orallo · Cèsar Ferri · Nicolas Lachiche · Adolfo Martínez-Usó · M.José Ramírez-Quintana

Received: date / Accepted: date

Abstract Some supervised tasks are presented with a numerical output but decisions have to be made in a discrete, binarised, way, according to a particular cutoff. This *binarised regression task* is a very common situation that requires its own analysis, different from regression and classification —and ordinal regression. We first investigate the application cases in terms of the information about the distribution and range of the cutoffs and distinguish six possible scenarios, some of which are more common than others. Next, we study two basic approaches: the *retraining* approach, which discretises the training set whenever the cutoff is available and learns a new classifier from it, and the *reframing* approach, which learns a regression model and sets the cutoff when this is available during deployment. In order to assess the binarised regression task, we introduce context plots featuring error against cutoff. Two special cases are of interest, the *UCE* and *OCE* curves, showing that the area under the former is the mean absolute error and the latter is a new metric that is in between a ranking measure and a residual-based measure. A comprehensive evaluation of the *retraining* and *reframing* approaches is performed using a repository of binarised regression problems created on purpose, concluding that no method is clearly better than the other, except when the size of the training data is small.

Keywords Regression · classification · reframing · mean absolute error · cutoff · binarisation.

José Hernández-Orallo · Cèsar Ferri · Adolfo Martínez-Usó · M.José Ramírez-Quintana
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí de Vera s/n, 46022, València, Spain
E-mail: {jorallo,cferri,admarus,mramirez}@dsic.upv.es

Nicolas Lachiche
ICube, Université de Strasbourg, CNRS
300 Bd Sebastien Brant - BP 10413, F-67412 Illkirch Cedex, France
E-mail: nicolas.lachiche@unistra.fr

1 Introduction

Data mining tasks are characterised by the available data and the decisions that have to be made according to those data. Supervised (or predictive) problems are defined over an input (or feature) space $\mathbb{X}$ and an output space $\mathbb{Y}$. If the output space is numeric (quantitative) we usually talk about regression problems, while if the output space is nominal (categorical) we usually talk about classification problems. However, things are more complicated than this. For instance, in this paper we consider the case where $\mathbb{Y}$ is numeric in the training data, but becomes nominal (actually Boolean, denoted by $\mathbb{Z}$) during the deployment of the model. Let us consider an example.

Example 1 An estate agent has a database of possible customers who are interested in buying a house. The estate agent collects information about each customer and learns a model about the maximum mortgage that the customer can get from a bank. This is our regression model. On an everyday basis, several new properties enter the estate agent's portfolio. Each of them has a different price. Obviously, the estate agent only offers a property to those customers that can afford it, i.e., those that can get a mortgage for at least the property price[1]. That means that each property represents a genuine *cutoff* of customers, those who can afford the property and those who cannot.

We will use the term *binarised regression problem* for this type of problem, as the data are given like a regression problem but decisions are like in (binary) classification, as the only thing that matters is whether each new example is above or below the cutoff. *Binarised regression problems* appear in many situations, especially when there is an all-or-nothing reward (or loss) if the actual value is above (or, respectively, below) a given cutoff. This is usual whenever there is a discrete cutoff that implies a qualification, entrance, label or accomplished objective or when the output space represents a count (number of sales, calls, payments, complaints, failures, or any other quantity [10]), either originally or as a result of an aggregation operation over an indicator value in a datamart [3].

The *binarised regression problem* appears naturally in many application areas. In fact, we have collected 20 datasets that will be used as *binarised regression problems* in our experimental section, apart from the running example shown above.

The basic idea arising in all these cases is that, for many applications, we are interested in telling whether the predictions are above or below a given cutoff. This cutoff c can vary depending on the context and will critically determine the overall performance. This problem leads to the consideration of two basic alternatives:

– *Reframing* (post-binarisation): A natural (eager) approach would be to train a model on the original training data just once and then *reframe* its

[1] Note that some people can buy a house that is much cheaper than its maximum mortgage, especially if they buy it as an investment or to refurbish it afterwards.

predictions using the cutoff c, once it is known, on the deployment data. In this paper, we understand reframing as follows: we train a regression model m_{regr} and we predict class labels as $\hat{z} = \mathbf{1}\{m_{regr}(x) \geq c\}$.

– *Retraining*[2] (pre-binarisation): A (lazy) alternative would be to wait until the cutoff c is known during deployment time, and only then the original output variable y in the training data is binarised as $z = \mathbf{1}\{y \geq c\}$. This makes a new classification training set $T' = \langle X, Z \rangle$, from which a classification model m_{clas} is trained. For each new instance x, the predicted class label $\hat{z} = m_{clas}(x)$ (1 or 0) is just used to make the decision. Note that this retraining has to be done each time the cutoff changes.

Basically, the differences between both alternatives are (1) whether the discretisation is performed before or after training and, as a consequence, (2) whether the training data must be kept and used whenever c changes (retraining) or is just used once and for all (reframing). Figure 1 shows these two alternatives graphically.

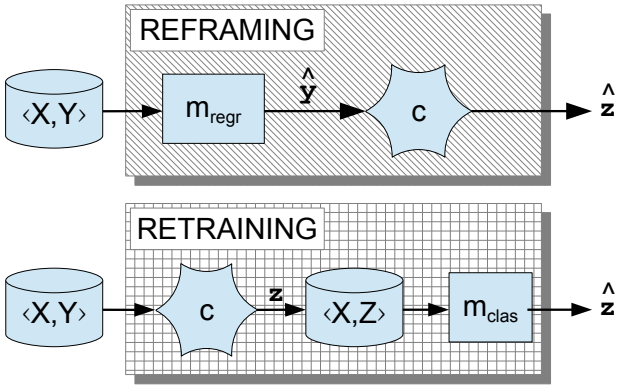

Fig. 1 Reframing vs. retraining. Reframing (top) shows how the training data $T = \langle X, Y \rangle$ is used just once to create a regression model m_{regr} that is applied to different operating contexts c by properly discretising its output $\hat{y}$ each time. Retraining (bottom) needs to convert the training data T for each context c into a new dataset $T' = \langle X, Z \rangle$ from which a classification model m_{clas} is learned.

In this work we present these two approaches to the binarised regression problem, which can be applied to many realistic problems. Knowledge reuse (reframing) seems a more efficient approach than the systematic generation of throw away models (retraining). This, and the possibility of using both classification and regression techniques, is the major reason for the study of these two options. These are two straightforward approaches. However, to our

[2] It is worth noting that the training process is entirely repeated in the retraining alternative, having nothing to do with any kind of incremental learning or adaptation of the previous model. This use of the term 'retraining', understood as building a different model each time a new cutoff is set, can often be found in the active learning research field [15,33].

knowledge, there is no systematic study in the literature about which method is best and why. There might be several reasons for this:

- The *binarised regression problem* we describe here has not been fully identified as a standalone problem in data mining.
- The learning techniques used for both approaches must be different (at most, we can use the same paradigm for both, such as a linear regression vs. a logistic regression, or a regression tree vs. a classification tree).
- It seems, a priori, that the reframing approach has more advantages: the model is trained with a richer version of the output variable and there is no need to keep the training data.
- While it is clear that the same performance metric should be used for both approaches, as the application is the same, the use of both regression and classification models may lead to confusion about this issue.

In the framework of the *binarised regression problems*, this work makes the following main contributions:

1. We identify and analyse a diverse collection of scenarios depending on the context knowledge for the *binarised regression problem*. These scenarios are illustrated with examples in our experimental section.
2. We formalise the notion of error for this kind of settings. We derive context plots that show the average error w.r.t the operating contexts c. We call them the cutoff error (CE) plots.
3. We introduce two versions of these plots. The first one considers a *uniform* range of cutoffs and we call them the uniformly-distributed cutoff error (UCE) plots. Interestingly, we prove that, for the reframing approach using regression models, the area under this curve (A_{UCE}) is the mean absolute error (MAE). Also, if we just focus on a partial region of cutoffs, we derive that the area of this region corresponds to a clipped version of the MAE.
4. The second kind of plots considers the x-axis is weighted by the observed distribution on output value. These are called output-distributed cutoff error (OCE) plots. In other words, the distribution of contexts is taken according to the observed prior $p(Y)$. We will discuss whether, in absence of knowledge about the distribution of operating contexts, this assumption is preferable or not over the uniform distribution, and hence its area, A_{OCE}, is preferable over the A_{UCE}.
5. We study the properties of both plots and their interpretation as evaluation metrics in this problem and also as general evaluation metrics for regression models.
6. We analyse the use of these plots and metrics in a case study, where we study both the reframing and retraining approaches, showing the usefulness of the newly introduced plots and metrics.
7. We perform a systematic study for the *reframing* and *retraining* approaches with a repository of binarised regression problems, analysing whether one approach is better than the other.

The rest of the paper is organised as follows. Section 2 states the problem and gives a taxonomy for the *binarised regression problems*. The error function

and the corresponding expected error expression are also introduced in this section. Section 3 derives the *UCE* curves and shows that the area under these curves for a regression model equals the *MAE*, including the correspondence for partial regions. Section 4 introduces the *OCE* curves. Section 5 briefly analyse the behaviour of the constant models, especially the role of the median model and the extreme models. Section 6 elaborates a case study with the example we have used in the introduction and illustrates the new plots and metrics with it, as well as the reframing vs. retraining dilemma. Section 7 performs a series of experiments to see the performance of the two different approaches. Section 8 discusses some related work and Section 9 closes the paper and outlines some future work.

2 Binarised regression problems

After the motivation in the previous section, we will now be more precise about the specification of the *binarised regression problem*. The input space is denoted by $\mathbb{X}$, the original (*numerical*) output space is denoted as $\mathbb{Y}$, its discretisation is denoted as $\mathbb{Z} = \{0, 1\}$, m_{regr} denotes a regression model trained from a dataset $T = \langle X, Y \rangle$, with vectors X and Y of the same size n taken from sets $\mathbb{X}$ and $\mathbb{Y}$ respectively, m_{clas} denotes a classifier trained with $T' = \langle X, Z \rangle$, where the vector Z, also of size n, is taken from $\mathbb{Z}$. We will use a somewhat abused set notations for elements in these vectors (as if they were ordered multisets). We hence refer to instances by their index, so $\hat{z}_i$ denotes the predicted class label for example i given by m_{clas} or by applying the cutoff c to $\hat{y}_i$, the estimation made by m_{regr}. With m we will denote whichever model (m_{regr} or m_{clas}).

2.1 Problem setting

We first define the notion of operating context:

Definition 1 Given a dataset $T = \langle X, Y \rangle$ where $\mathbb{Y} = \mathcal{R}$, the operating context $c \in \mathbb{Y}$ is defined as the value that determines a cutoff that splits the set Y (or its prediction $\hat{Y}$) into two disjoint sets: the values y_i (or $\hat{y}_i$) that are above and below c, respectively.

Therefore, when the operating context is applied to the actual values y_i, the original regression problem turns into a new classification problem which leads to the *retraining approach*, where a classification model m_{clas} can be learned. Alternatively, when the operating context is applied to the predicted values $\hat{y}_i$ given by the regression model m_{regr} we have the *reframing approach*. To be able to analyse and compare both approaches, we need to relate classification errors on $\mathbb{Z}$ and operating contexts.

Given the cutoff c, an error is any case where z and $\hat{z}$ differ (in the retraining approach, using a classification model) and the cases where y and $\hat{y}$ are on

different sides of the cutoff (in the reframing approach, using a regression model). These errors are known as false positives FP and false negatives FN.

The error function for each example is then defined as follows:

$$Q_i(c) \triangleq C_{FP} \cdot FP_i(c) + C_{FN} \cdot FN_i(c) \quad (1)$$

where C_{FP} and C_{FN} are the misclassification costs of false positives and false negatives respectively. Note that if $C_{FP} = C_{FN} = 1$, then Q_i would be 1 for misclassification and 0 otherwise.

Definition 2 The average error for a dataset with n examples with respect to a cutoff c is

$$Q(c) \triangleq \frac{1}{n} \sum_{i=1}^{n} Q_i(c)$$

For equal unitary misclassification costs, the above is equivalent to the misclassification rate. As we are considering that the context c changes, we are interested in the error for a range of contexts. If we know or assume a distribution of contexts, we can define the expected average error as:

Definition 3 The expected average error for a dataset under a context distribution $w(c)$ is

$$L \triangleq \int_{-\infty}^{\infty} Q(c)w(c)dc$$

It is then clear that if we are given an operating context c (or a context distribution $w(c)$) and we have to choose between several models, the model with lowest $Q(c)$ (respectively, L) will be preferred. However, a more useful and common question is when we want to do model selection and evaluation and we do not yet know the operating context or its distribution. Or, in other words, we would like to know how well a model behaves for a range of operating contexts. In the classical classification problem, this problem has been well-studied (works on ROC analysis and other related areas) and similar approaches exist for the regression problem. However, to our knowledge, this is not the case for the *binarised regression problem.*

When presenting a series of tools in the following sections, we need to bear in mind the distinction between *model selection* and *model evaluation.* Model selection can be done with some information or all information available about the context and the training dataset, whilst model evaluation will be done with a test dataset with all the information.

2.2 A Taxonomy of binarised regression problems

In this section we devise a taxonomy of binarised regression problems depending on the factors involved in the process. Then, we discuss for each case how to apply and evaluate the retraining and reframing approaches and to use the cost plots introduced in this paper.

The binarised regression problems are characterised by the following features that determine the context: the cutoff distribution and its range. According to the degree of knowledge about these two features of the context that we may have during training, we distinguish the following cases (from more to less information):

a) we know the exact cutoff.
b) we know the range and the expected distribution of cutoffs for deployment precisely.
c) we may have some information about the range or the distribution.
d) we may have complete absence of information about the range and the distribution.

Case **a** is the simplest one. In this scenario, there is no need of learning a regression model since the problem can be solved directly by using the cutoff in the training stage. First, the problem is binarised by using the cutoff and, then, a classifier is learnt. However, for the rest of the cases, the performance of the models for the range of possible cutoff values which we are interested in must be examined. For cases **c** and **d**, where no information about the cutoff is available, uniform-distributed or the output-distributed cutoff may be the most straightforward assumptions as we will discuss in the following sections, being the uniform distribution especially applicable when we want to focus on a small region[3] of cutoffs. Otherwise, the problem could not be solved.

Focussing on cases **b** and **c**, we have considered six possible binarised regression scenarios (types) which are shown in Table 1. In the second column, we consider different cases for the *true* distribution of cutoffs (w), which can follow a uniform distribution, be similar to the output distribution observed in the training set or any other distribution. The third column shows the range of this distribution as *Full* or *Region*, depending on whether we expect a wide range or a narrow range of cutoffs, respectively. The fourth column shows whether each type of problem can be considered common or not whereas the fifth column shows which error measures would be recommended for each case (which will be seen in the next sections).

A collection of datasets from common repositories [1,2] can be found at http://www.dsic.upv.es/~flip/BinarisedRegression/. For each problem, we have included information about the binarised regression task, the distribution and regions of cutoffs that make more sense for the problem, as well as the C_{FP} vs C_{FN} costs. These datasets have also been used in the experiments in Section 7. The repository features three datasets for which the cutoff has a narrow uniform range (type 2), seven with a wide output-distributed range (type 3), one with a narrow output-distributed range (type 4), eight with a wide range with any other distribution (type 5) and one with a narrow range with any other distribution (type 6). There are no type-1 problems in our repository since this situation is quite unlikely. In this collection, there

[3] Note that *region* is here used to refer to an interval (continuous subset of values) within all the possible cutoff values. This interval will usually be narrow.

Type	Cutoff Distribution (w)	Range	Common?	Measure Method
1	Uniform	Full	No	MAE
2	Uniform	Region	Yes	Clipped MAE ($cMAE$)
3	Output	Full	Yes	A_{OCE}
4	Output	Region	Yes	A_{OCE} for the region (or $cMAE$)
5	Other	Full	Yes	A_{CE}
6	Other	Region	Yes	A_{CE} for the region (or $cMAE$)

Table 1 Types of cutoff contexts, including the distribution and range of cutoffs. *Common?* indicates how realistic each kind of problem seems to be, and *Measure Method* shows which evaluation measure is recommended, where A_P denotes the area under the plot P. For cases 4 and 6, if the region is small it is acceptable not to use the true distribution, and use a flat uniform distribution instead, so we can ultimately use $cMAE$.

are situations where a uniform distribution of cutoffs is observed for a region of interest (type 2). For instance, some problems depend on cutoffs that are established according to certain local or national normative or regulation such as the allowed acoustic noise level or the energy consumption required for building efficiency qualification. Usually, the interest in this kind of problems focusses on a small region of values for which we usually lack a clear information about whether it is 2, 4 or 6.

Type-3 problems have a wide range of cutoffs that are approximately distributed as the output variable. The running example introduced in Section 1 belongs to this type. Niche applications detected for type 3 have some common problem typologies:

- *Supply-demand regulation*, which includes all problems where we predict sales, trade, stocks, etc., and there is a supply-demand equilibrium, which aligns the cutoffs with the output values.
- *Trend sign detection*, which includes all problems where we predict the change of a quantity in time and the decision is whether the value is higher or lower than the current value (so cutoffs are taken from the output values).
- *Who's above me* niche, which includes all problems where an individual is interested in knowing the examples that are above itself, independently of the magnitude. Again, the cutoffs are samples from the output distribution.

For instance, Figure 2 shows the distributions of mortgage amounts (which we called output value from the model) and property prices (the true cutoffs) for the running Example 1. In this case, both distributions are very similar and we can use the actual distribution of outputs as a surrogate of the distribution cutoffs. Hence, this example belongs to case **b**, is of typology 3 and corresponds to a 'supply-demand regulation' case.

2.3 *CE* curves

We first present a graphical way of analysing binarised regression problems for a range of possible cutoff values (cases from **b** to **d**):

Fig. 2 Comparing the true cutoff distribution (top) with the output distribution (bottom) for the running Example 1. The top figure shows the distribution of the mortgage amounts (source [11]) whereas the bottom figure shows the house prices (source [38]) in USA for year 2013. As it can be seen, both mortgage amounts and house prices exhibit approximately the same distribution. This belongs to case **b**, is of typology 3 and corresponds to a 'supply-demand regulation' case.

Definition 4 The cutoff error (CE) curve plots the error (value of $Q(c)$) on the y-axis as a function of the cutoff c on the x-axis.

Note that the same plot can be drawn for the retraining approach, where each point of the curve would correspond to a different classification model m_{clas} for the cutoff on the x-axis. That means that the CE curve for the retraining approach would not show the performance for one classifier, but for a procedure that generates a possibly different classifier for each point.

Figure 3 shows the CE plot for the running Example 1. The x-axis is drawn according to the cutoff distribution (i.e., the actual mortgage distributions shown on top of Figure 2). For the reframing approach, we have applied five regression algorithms (linear regression, kNN-median, kNN-mean, a regression tree, and quantile regression). For the retraining approach we have applied three algorithms (logistic regression, kNN and a decision tree). We have used a partition 50%-50% for training and test. For every curve, the area under the curve is the expected average error given in definition 3. This is the case because we are drawing the magnitudes on the x-axis using the cutoff distribution w, which magnifies (widens) those regions where cutoffs are more likely according to w. Note that depending on the cutoff the best model

changes and we see different *dominance* regions. In Section 6 we carry out a more detailed analysis of this case.

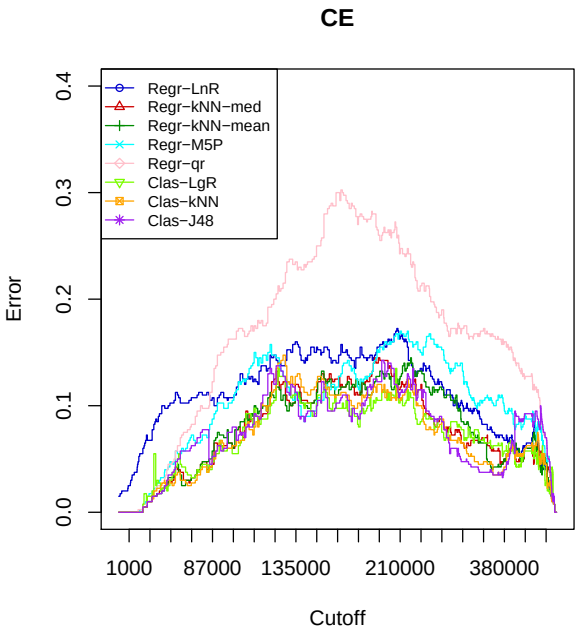

Fig. 3 *CE* curves for five regression models (linear regression, kNN-median, kNN-mean, the regression tree M5P, and quantile regression) and three classification models (logistic regression, kNN and the decision tree J48) for the case study in Example 1.

Note that if we chose the best model for each cutoff, i.e., the lower envelope of the *CE* plot, we would take the most of all models, by selecting the one that dominates in each region. We call this solution a *hybrid* model, and it is theoretically optimal if these regions can be perfectly determined for the test set. Also, these dominance regions are *independent on the distribution*. The problem, as we will see in the experimental section, is that it is not so easy to estimate these regions from the training (or validation) set and extrapolate them for the test set.

It is easy to see that there is no need to plot the curves from $-\infty$ to ∞. If we are dealing with classification models, and if we assume that a classifier that is learned with all positives always predicts positive and a classifier that is learned with all negatives always predicts negative, we have that we only need to look at the plot between c_{min} and c_{max}, where $c_{min} = \min\{y : \langle x, y\rangle \in T \cup D\}$ and $c_{max} = \max\{y : \langle x, y\rangle \in T \cup D\}$, where T is the training dataset and D is the deployment dataset. In other words, there is no need to plot beyond any of the observed (true) values in both the training and deployment datasets. Somewhat similarly, if we are dealing with a regression model, we only need to consider the set of actual values in the deployment dataset and the set of predictions for this dataset. $c_{min} = \min(\min\{m_{regr}(x) : \langle x, y\rangle \in D\}, \min\{y :$

$\langle x, y\rangle \in D\}$), and $c_{max} = \max(\max\{m_{regr}(x) : \langle x, y\rangle \in D\}, \max\{y : \langle x, y\rangle \in D\})$. In other words, there is no need to plot beyond any of the observed (true) and estimated values in the deployment dataset. This suggests that plots only need to be drawn in this range, as above, since for any $c < c_{min}$ or $c > c_{max}$ we have that $Q(c) = 0$.

3 *UCE* curves

In the previous section we have seen that we can modify the x-axis of the *CE* plot by using any distribution w. However, it is interesting to analyse the case where w is uniform. This is not because we assume w uniform for selection or evaluation purposes, but just because the plot is much easier to understand as the x-axis is linear w.r.t. the cutoff magnitude. To highlight this choice of distribution for the x-axis, we use the term *UCE* plots (uniform-distributed cutoff error plots).

Let us illustrate the *UCE* plots with an example.

Example 2 Consider a regression model m_{regr} which is applied to a dataset with $n = 6$ instances with true values y_i, producing the predicted values $\hat{y}_i$ and leading to absolute error AE_i, shown in the following table:

i	1	2	3	4	5	6
y_i	3	5	6	8	11	12
$\hat{y}_i$	9	4	7	10	16	13
AE_i	6	1	1	2	5	1

Without loss of generality, examples are sorted by increasing values of y for convenience. The Absolute Error is $AE = 16$, and the Mean Absolute Error is $MAE = 16/6 = 2.66$. Figure 4 (left) shows what we call the *UCE* curve for Example 2. Note that, in this case the values in the x-axis range from $c_{min} = 3$ and $c_{max} = 16$.

We can see that the curve in Figure 4 configures 16 rectangles of width 1 and height 1/6. This leads to the following straightforward but interesting result:

Theorem 1 *The area under the UCE curve (A_{UCE}) for a regression model equals the mean absolute error MAE.*

Proof We just operate with the area under the curve as:

$$\int_{-\infty}^{\infty} Q(c)dc = \int_{-\infty}^{\infty} \frac{1}{n}\sum_{i=1}^{n} Q_i(c)dc \;=\; \frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{\infty} Q_i(c)dc$$

It is easy to see that for each example, when $\hat{y}_i < y_i$, there is a region $c < \hat{y}_i < y_i$ where $Q_i(c) = 0$, there is another region $\hat{y}_i < c < y_i$ where $Q_i(c) = 1$ and, finally, there is another region $\hat{y}_i < y_i < c$ where $Q_i(c) = 0$. As the width of the second region is $|y_i - \hat{y}_i|$, this is the area contributed by example i. A

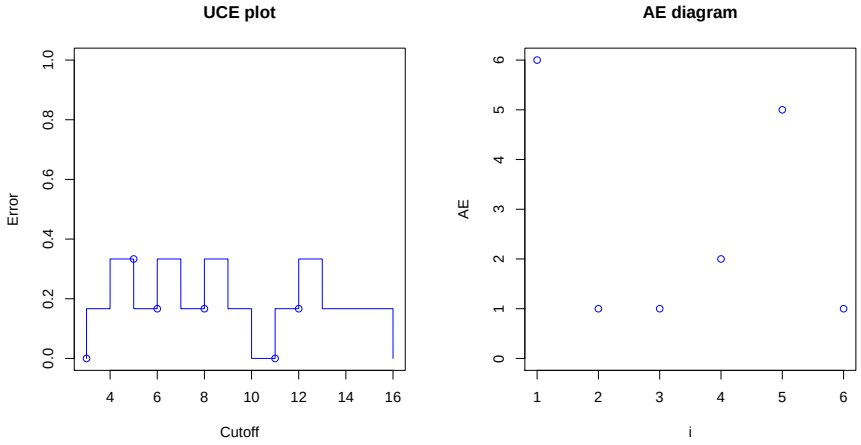

Fig. 4 Left: *UCE* curve for Example 2 with c on the x-axis and $Q(c)$ on the y-axis. The continuous line shows $Q(c)$. The small circles show the exact value of $Q(c)$ for the values of y (borderline cases). Right: The absolute error is shown with a very different (example-wise) decomposition.

similar rationale can be used when $\hat{y}_i > y_i$. And the area is 0 when $\hat{y}_i = y_i$. So, the inner integral above simplifies:

$$\int_{-\infty}^{\infty} Q_i(c)dc = \int_{\min(\hat{y}_i,y_i)}^{\max(\hat{y}_i,y_i)} Q_i(c)dc = |y_i - \hat{y}_i| = AE_i$$

Now, by just plugging this into the original equation we get:

$$\int_{-\infty}^{\infty} Q(c)dc = \frac{1}{n}\sum_{i=1}^{n} AE_i \;\; = \;\; MAE$$

□

These curves show how the *MAE* is distributed for different ranges of the output variable. In Example 2, we see that both *MAE* and the area are 2.66. The previous proof and the traditional examplewise decomposition of the *MAE* as $\sum_{i=1}^{n} AE_i$ suggest an interesting comparison between a diagram where we show each example in the x-axis (sorted by y) with AE_i on the y-axis. This is what can be seen in Figure 4 (right). While the area and the sum on the left plot and the right diagram respectively are the same, they decompose the *MAE* very differently.

So is *MAE* an expected error for our binarised regression problem? In order to make this link exact, we have to assume a uniform distribution of the context c, $w_u(c) = \frac{1}{c_{max}-c_{min}}$. If we consider continuous data bounded by c_{min} and c_{max}, the expected error function defined in Definition 3 can be calculated as follows:

$$L_u \triangleq \int_{-\infty}^{\infty} Q(c)w_u(c)dc = \int_{c_{min}}^{c_{max}} Q(c)w_u(c)dc$$

This leads to the following corollary of Theorem 1:

Corollary 1 *For a regression model,*

$$L_u = \frac{MAE}{c_{max} - c_{min}}$$

Proof This just derives from the fact that $Q(c)$ is always 0 beyond the interval c_{min} and c_{max} and that the uniform distribution is independent of c inside this interval. □

In Example 2, this is $L_u = \frac{2.66}{16-3} = 0.208333$.

What is the interpretation of this result? While it is a straightforward connection, this gives a new interpretation of the *MAE* as aggregate performance (more precisely, error) under a set of operating contexts (the cutpoint c). Another possible interpretation of the *MAE* is probabilistic: "Take a random cutoff c uniformly between c_{min} and c_{max}. The probability that for a random example its y and $\hat{y}$ are not on the same side of the cutoff is linearly related to *MAE*".

What happens with the retraining approach? What is the interpretation of A_{UCE} for a set of classification models constructed under this procedure? We conclude that the A_{UCE} is the expected error for this procedure as well, but we do not have the connection with any metric for classification. It is important to clarify here that we can actually calculate the *MAE* of a *probabilistic* classifier, but this is done by comparing the scores of a classifier with the actual values (0 or 1). The A_{UCE}, in contrast, represents the expected error for the infinite set of *crisp* classifiers that are generated for all possible cutoffs. In other words, for the retraining approach, A_{UCE} gives the expected error of a procedure, not the performance of a single classifier[4].

The assumption that the range of examples goes from the minimum to the maximum values is unrealistic, as we discussed in the previous section (type 1 in Table 1). On many occasions, we are only interested in a partial region of interest. While this can perfectly observed in any *CE* curve, if we want to calculate the expected area for a single region, we can no longer use *MAE* as an approximation. This is a pity, as when the region is small, the actual distribution is not so relevant, and a flat (uniform) distribution could be a choice, and the *UCE* plots would fit perfectly. Fortunately, we can still find a connection between *MAE* and partial areas. In order to see this, we have to clip the values y and its predictions $\hat{y}$ to fit in the region of interest. Let us first define the *clipping* of a value.

[4] For the interested reader, it is worth mentioning that Theorem 1 is connected to Theorem 11 (and corollary 12) in [20], where the expected loss of the score-uniform threshold choice method for a uniform distribution of operating contexts (cost proportions or skews) is shown to be equal to *MAE*. Two comments must be done, though. First, here we are talking about the *MAE* of a regression model while in [20] the result holds for a soft classifier with estimated probabilities between 0 and 1 —upon which the *MAE* is calculated. Second, here the decision rule is taking the operating context into account —the cutoff is used at each point of the curve, while in [20] the result is obtained by the score-uniform threshold choice method, which completely ignores the operating context. Nevertheless, this is still an interesting connection as both are assuming a uniform distribution of contexts.

Definition 5 Given two real numbers a and b with $a \leq b$, known as the low and high clipping limits, respectively, the *clipping* of a value $y \in \mathcal{R}$ with respect to a and b is defined as

$$[y]_a^b = \begin{cases} a & \text{if } y < a \\ y & \text{if } a \leq y \leq b \\ b & \text{if } y > b \end{cases}$$

Analogously, the difference between a clipped value and its clipped prediction is called the *clipped absolute error cAE*,

$$cAE = \left|[y]_a^b - [\hat{y}]_a^b\right|$$

From here, we can show the following result:

Theorem 2 *The expected error (area) for a region of cutoffs where we assume a uniform distribution corresponds to a clipped MAE (cMAE) where all the values (estimated and true) are clipped by the region of interest.*

Proof Using a similar rationale to Theorem 1, let us assume certain a and b values as the limits of our region of interest. So, all actual and predicted values have to be clipped w.r.t. the a and b clipping limits.

$$\int_a^b Q_i(c)dc = \int_{\max(\min(\hat{y}_i,y_i),a)}^{\min(\max(\hat{y}_i,y_i),b)} Q_i(c)dc = \left|[y_i]_a^b - [\hat{y}_i]_a^b\right| \triangleq cAE_i$$

$$\int_a^b Q(c)dc = \frac{1}{n}\sum_{i=1}^{n} cAE_i \triangleq cMAE$$

Note that the expected error is not given by taking only the subset of examples inside the cutoff interval but rather by using all the examples where both the estimated $\hat{y}$ and the actual y are clipped. Apart from the notation $cMAE$, we will use the equivalent term A_{PUCE}, referring to a *Partial UCE*.

4 *OCE* curves

While *UCE* plots allow for the analysis of dominance regions, and we can derive partial areas when we know the region of interest, what can we do in the case where we do not know the true cutoff distribution or the region (case **d** in Section 2.2), i.e., no information at all about the cutoffs? The use of A_{UCE} as an aggregated error under a range of operating contexts assumes a uniform distribution, which is unrealistic. When we are not given any information about $w(c)$ or any region, one interesting possibility would be to assume that the operating context distribution is induced by the data, i.e., $w_o(c) = f(c)$, where $f(c)$ is the *observed* probability density function of the data (the a priori

distribution for the *output* values y). This leads us to the following expected error under that observed distribution $w_o(c)$:

$$L_o \triangleq \int_{c_{min}}^{c_{max}} Q(c) w_o(c) dc \tag{2}$$

In other words, we could think of a discrete distribution where $w_o(c) = 1/n$ for those c that match a value in the observed data. This is actually what the circles in the *UCE* plots of Figure 4 (left) are showing. What if we just plot these points? This is precisely what we do next.

Let us define the *true rank ratio* as follows. Consider that the y_i are ordered by ascending order. The true rank ratio is defined as $R(i) \triangleq \frac{i-0.5}{n}$, where n is the number of examples. For the continuous case, $R(i)$ is invertible, but we need to work with empirical distributions and a finite set of points. The set of possible exact ratios for an example in a dataset D is denoted by $\mathcal{R}(D)$. Assume that there are no ties or there is a way to unequivocally sort the y_i by a criterion to resolve them. In this case, R is invertible and we can get a cutoff for every ratio $r \in \mathcal{R}(D)$, i.e., $y_{R^{-1}(r)}$. So now we could plot n points, with $r = R(i)$ for $i = 1..n$ on the x-axis and $Q(y_{R^{-1}(r)})$ on the y-axis. But, for the discrete case (the observed distribution), this is not actually a 'curve'. A 'curve' can be obtained by interpolation or by any other way of connecting the points. When n is large, how the points are connected becomes less and less relevant in terms of precision. Nonetheless, we will make a choice such that the area under the resulting curve is exact, i.e., equals the average of the points. In order to do this we draw horizontal segments of the same size around each point, by defining Q for all possible ratios in the continuum $[0, 1]$, i.e.:

Definition 6 The average error for a dataset D with n examples with respect to any true rank ratio $r \in [0, 1]$ is:

$$\bar{Q}(r) \triangleq Q(y_{R^{-1}(r_o)}) \text{ where } r_o = argmin_{\rho \in \mathcal{R}(D)} |r - \rho|$$

Basically, these segments are just built by looking for the closest ratio that corresponds to an example. And now we can define a curve for all points:

Definition 7 The output-distributed cutoff error (*OCE*) curve plots the error (value of $\bar{Q}(r)$) as a function of the true rank ratio $r \in [0, 1]$.

Note that the 'curve' is not continuous but it is now defined for all r in $[0, 1]$. Figure 5 (left) shows the *OCE* curve for Example 2.

We now need to check that our construction has led to a curve whose area A_{OCE} corresponds to L_o as for Eq. 2. This is shown next:

Theorem 3 *The area under the OCE curve (A_{OCE}) equals L_o.*

Proof As the ratios in $\mathcal{R}(D)$ are equally spaced at a distance of $1/n$ ($R(i) = \frac{i-0.5}{n}$), we can split the integral of the definition of the area under the *OCE*

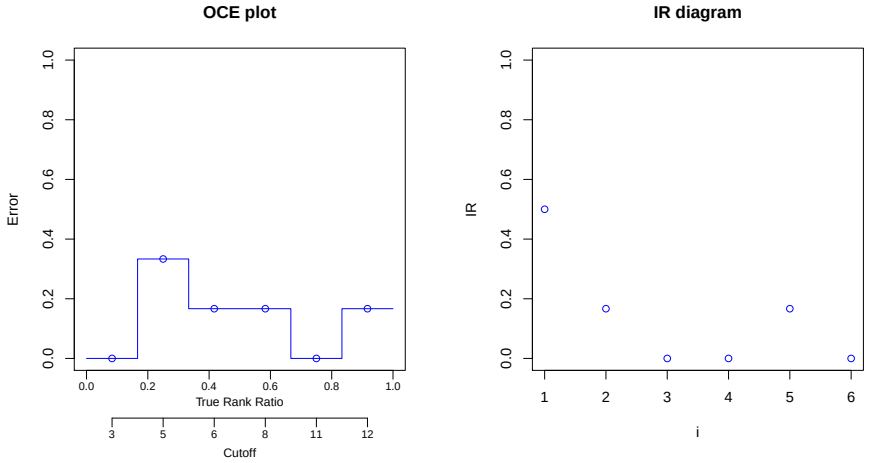

Fig. 5 Left: *OCE* 'curve' for Example 2 with the true rank rates r on the x-axis and $Q(y_{R^{-1}(r)})$ on the y-axis. Right: the interposition ratio for each example. Both left and right: note the correspondence with Figure 4.

curve into n intervals of constant $\bar{Q}$:

$$\int_0^1 \bar{Q}(r)dr = \int_{R(1)-1/2n}^{R(1)+1/2n} \bar{Q}(R(1))dr + \int_{R(2)-1/2n}^{R(2)+1/2n} \bar{Q}(R(2))dr + \cdots + \int_{R(n)-1/2n}^{R(n)+1/2n} \bar{Q}(R(n))dr$$

As all segments have the same width, we have:

$$\begin{aligned}\int_0^1 \bar{Q}(r)dr &= \frac{1}{n}\left\{\bar{Q}(R(1)) + \bar{Q}(R(2)) + \cdots + \bar{Q}(R(n))\right\} \\ &= \sum_{i=1}^{n}\left\{\bar{Q}(R(i))\frac{1}{n}\right\}\end{aligned}$$

Now we have to look at the empirical distribution $w_o(c)$, which is equal to $1/n$ iff $c \in \{y_{R^{-1}(r)} | r \in \mathcal{R}(D)\}$. Then, from Definition 6, the above expression can be rewritten as:

$$\int_0^1 \bar{Q}(r)dr = \sum\{Q(c)w_o(c)\}$$

which is the discrete version of:

$$\int_{-\infty}^{\infty} Q(c)w_o(c)dc = \int_{c_{min}}^{c_{max}} Q(c)w_o(c)dc \tag{3}$$

since $Q(c)$ is zero outside (c_{min}, c_{max}), so finally leading to the definition of L_o in Eq. 2. □

For Example 2, the area under the *OCE* curve is $L_o = (0/12 + 2/12 + 1/12 + 1/12 + 0/12 + 1/12) = 0.4166666$. Note that this value is different to L_u (0.208333), as we are using different distributions for the operating context c.

Therefore, if we are not given further information about the distribution for c or the range of cutoffs, the use of the prior distribution for y as the cutoff distribution is based on the fact that this is the only available knowledge to address the problem. In addition, a practical reason is that the calculation of the new measure is extremely simple. In the case of retraining, we just need to retrain a classification model and calculate its $Q(c)$ for every cutoff c found in the test set (i.e., those with $w_o(c) = 1/n$). But interestingly, for reframing, we just train one regression model and the curve can be calculated analytically, as we can calculate $Q(c)$ by comparing y and $\hat{y}$. This procedure has order in $\mathcal{O}(n^2)$. This original expression and this way of calculating the A_{OCE} correspond to the following interpretation:

Interpretation 1 *The area under the OCE curve (A_{OCE}) for a regression model can be interpreted as follows: "Take a random test example j and make the cutoff c equal its true value y_j. The probability that for another random example i its y_i and $\hat{y}_i$ are not on the same side of this cutoff c is the A_{OCE}".*

The above interpretation is related to the typology "who's above me" seen in Section 2. This is a common question whenever a group of people are given a score and we have an estimation of this score. If any individual uses the estimator for any other individual, the probability that the estimator places the second above or below the first incorrectly is the area under the OCE curve. The interpretation follows formally from expression in Eq. 3, which can be rewritten as $\int_{-\infty}^{\infty} \frac{1}{n} \sum_{i=1}^{n} Q_i(c) w_o(c) dc$. According to Eq. 1, if we assume equal unitary costs we have that $Q_i(c) = 0$ if the example is correctly classified and 1 otherwise, i.e., whether the example is misclassified or not according to cutoff c.

In the previous section we compared the *UCE* plots with an *AE* diagram (see Figure 4). We can do a similar decomposition for *OCE* plots:

Theorem 4

$$A_{OCE} = L_o = \frac{1}{n} \sum_{i=1}^{n} IR_i$$

where IR_i is known as the interposition ratio*, the proportion of actual values that can be found between $\hat{y}_i$ and y_i, more formally defined as follows:*

$$IR_i \triangleq \begin{cases} \frac{1}{n} \mid \{y_j : y_i < y_j \ \wedge \ y_j \leq \hat{y}_i\} \mid & \text{when } y_i < \hat{y}_i \\ \frac{1}{n} \mid \{y_j : y_i \geq y_j \ \wedge y_j > \hat{y}_i\} \mid & \text{when } y_i > \hat{y}_i \\ 0 & \text{when } y_i = \hat{y}_i \end{cases}$$

where the examples are assumed to be sorted by their true values.

Proof We start from the definition of L_o and proceed as we did in Theorem 1:

$$\int_{-\infty}^{\infty} Q(c)w_o(c)dc = \int_{-\infty}^{\infty} \frac{1}{n}\sum_{i}^{n} Q_i(c)w_o(c)dc = \frac{1}{n}\sum_{i}^{n}\int_{-\infty}^{\infty} Q_i(c)w_o(c)dc$$

As we argued in the proof of Theorem 1, $Q_i(c)$ is 1 only when c is between $\hat{y}_i$ and y_i. So, we can express this as:

$$\frac{1}{n}\sum_{i}^{n}\int_{\min(\hat{y}_i,y_i)}^{\max(\hat{y}_i,y_i)} w_o(c)dc$$

As w_o is defined from the distribution of actual values y, then the inner integral represents the proportion of actual values that can be found between $\hat{y}_i$ and y_i, which is exactly IR_i. □

This is a new interpretation of A_{OCE}:

Interpretation 2 *The area under the OCE curve (A_{OCE}) for a regression model can be interpreted as follows: "Take a random test example i, the expected proportion of examples j whose true value y_j is between $\hat{y}_i$ and y_i corresponds to the area under the OCE curve".*

Clearly the closer $\hat{y}_i$ and y_i are the less likely another example can be interposed in between. This interpretation further clarifies that the area under the *OCE* curves is independent of a linear transformation of the magnitudes of the problem (unlike the area under the *UCE*, which is *MAE* and obviously depends on the magnitudes). Imagine that we predict 10 but it is actually 15 (an error of -5). Then the number of actual values between 10 and 15 is interesting, because it tells us whether this error is high or low, in terms of how many of the other examples can go "misclassified" in between. If we do this for the whole dataset, we have the area under the *OCE*. This independence of the magnitude of the output variable (or the threshold) gives an extra practical advantage for the use of *OCE* curves and its areas, if we do not have any other better choice.

The values of IR_i for each example are shown on Figure 5 (right). This new way of calculating A_{OCE} is more efficient than the original one, as for each example, the calculation of IR_i is in $\mathcal{O}(\log n)$ using a binary search, as the examples are assumed to be sorted by y. For n examples we have $\mathcal{O}(n \log n)$, which is equal to the order of previously sorting the examples.

The concept of interposition ratio is interesting as it is a component that corresponds to the *AE* in the *UCE* plots and leads to a more efficient calculation of the A_{OCE}. This exemplifies that A_{OCE} is a metric that is somewhat in between a residual-based metric and a ranking metric. However, an example-wise decomposition of the A_{UCE} loses our view of this metric as performance (or error) over a range of contexts c, which is well illustrated by the *OCE* plots.

So let us focus again on the *OCE* curve and its area. First, we will look at a more elaborate example than the previous toy examples. We will work with

the running Example 1 with the Zillow data of Figure 2. We randomly split the 799 examples in the dataset into 400 examples to train a linear regression, and the rest (399) were used for test (deployment). Figure 6 shows the *UCE* and *OCE* curves of this model. The first thing that we observe in these two

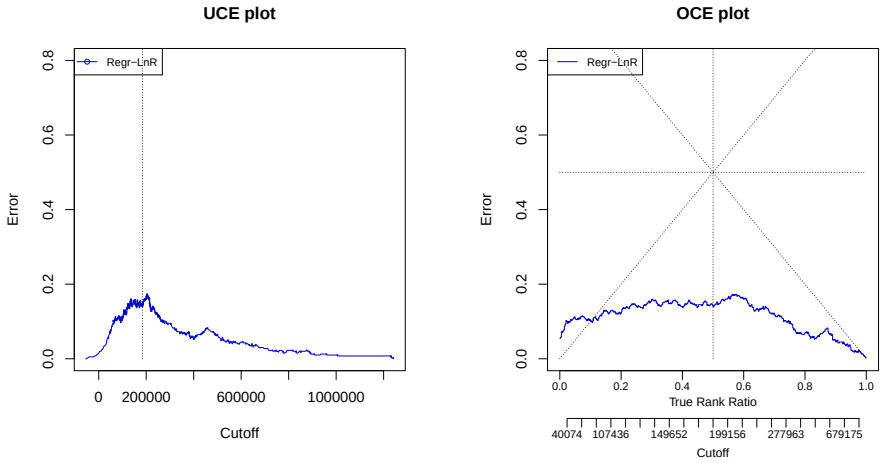

Fig. 6 *UCE* and *OCE* curves for a linear regression model for the Zillow data. The dotted vertical line shows the median. We see that half of the examples (left of the median) are extremely under-represented by the *UCE* plot.

plots is that they are significantly different. The plots show the median of the actual value as a vertical dotted line. This is very illustrative, as half of the examples are on the left, and they are clearly under-represented by the *UCE* plot, as this part is much narrower on the *UCE* plot than on the *OCE* plot. In fact, the contribution of this part for the A_{UCE} (i.e., the *MAE*) is small, while this is half of the plot for A_{OCE}. We can explain this by looking at a histogram of the output values again (Figure 2, bottom). The distribution is asymmetrical and far from uniform. Consequently, it is not very appropriate in this case to consider a uniform distribution of cutoffs from 0 to 1000000, when 75% of the examples are below 300000.

5 Trivial constant models

In order to better understand both the *UCE* and *OCE* plots, and *CE* plots in general, in this section we are going to examine some trivial models (see description in Table 2). The reframing approach used the five regression models, calculating the values of Q for each of them on the test set using the cutoffs. For the retraining approaches, the models Clas-Pos and Clas-Neg are fixed and always output positive and negative respectively. For the Clas-Maj, we converted the numerical output of the training set into a binary label for each cutoff and then we constructed (for each case) a new constant classifier

from each modified training set outputting the majority class (note that this actually leads to only two different classifiers that are just chosen depending on the cutoff). For the three trivial classifier models, the final step is that they are applied to the binarised test set directly for each cutoff.

Acronym	Type of *Constant* Model	Context-handling
Regr-Max	Regression: outputs ∞	Reframing
Regr-Min	Regression: outputs $-\infty$	Reframing
Regr-Avg	Regression: outputs the *mean* of the training set	Reframing
Regr-Med	Regression: outputs the *median* of the training set	Reframing
Regr-Bad	Regression: outputs ∞ for those below the *median* of the training set and $-\infty$ for the rest	Reframing
Clas-Pos	Classification: outputs positive	Retraining
Clas-Neg	Classification: outputs negative	Retraining
Clas-Maj	Classification: outputs the majority class	Retraining

Table 2 Seven different trivial models.

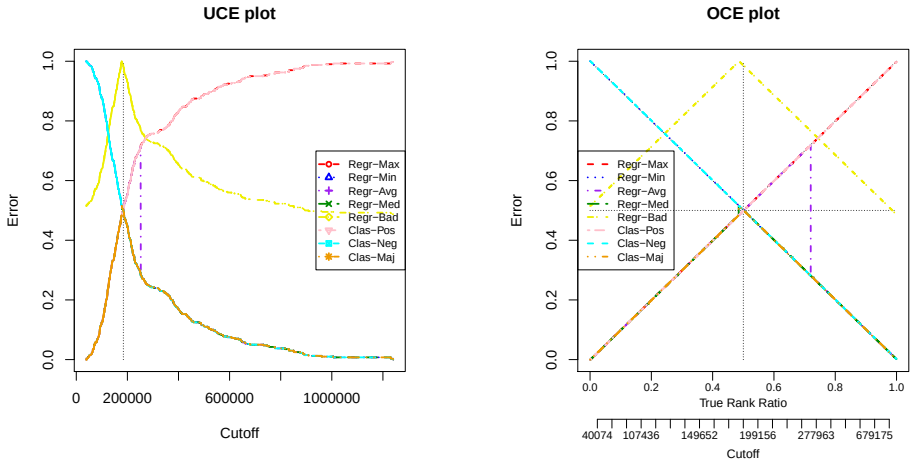

Fig. 7 *UCE* and *OCE* curves of the seven constant models in Table 2 for the Zillow Data (Example 1).

Figure 7 shows their plots. We see that the models Regr-Max (always predicting ∞) and Clas-Pos (always predicting the positive class) are equivalent (ascending diagonal in the *OCE* plot).

Similarly, we see that Regr-Min and Clas-Neg are equivalent (descending diagonal in the *OCE* plot). The Regr-Avg (mean=252485) performs very poorly here, as the distribution is highly asymmetrical and can only switch from the ascending diagonal to the descending diagonal very late. In contrast, the Regr-Med (median=177858) and Clas-Maj (the model outputting the majority class), which are equivalent, take about half of the ascending diagonal

and about half of the descending diagonal in the *OCE* plot. Note that the Regr-Bad approach has better performance with a worse estimation of the median, as it will get closer to either Regr-Max or Regr-Min. The inflection point for Regr-Med, Clas-Maj and Regr-Bad is not exactly where the diagonals cross (true rank ratio 0.5 in the *OCE* plot), because the median in the test set is slightly different (184128) from the training set.

Finally, Figure 7 shows the differences between *UCE* and *OCE* plots and the way in which the trivial models are shown as (combinations of) straight segments in the *OCE* plots. We also see clearly the difference between the mean model and the median model. There is plenty of information in these plots about the best and worst that can be done in this binarised regression problem. In general, this suggests the use of the median and majority models as baselines for this kind of problem.

From here, it is now the moment to show some easy properties of the *OCE* plots.

Theorem 5 *The OCE plots for Regr-Max and Clas-Pos are equivalent, Regr-Min and Clas-Neg are equivalent and Regr-Med and Clas-Maj are equivalent.*

Proof By Definition 6 we only need to consider ratios $r \in \mathcal{R}(D)$, where D is the deployment or test data. Let $|D| = n$. We denote the elements in $\mathcal{R}(D)$ as $r_1, \ldots, r_n$ such as $R^{-1}(r_i) = i$. Hence, $\bar{Q}(r_i) = Q(y_i)$.

Now, for each cutoff y_i, it holds that $\hat{y}_j = m_{Regr-Max}(x_j) > y_i$ for all $1 \leq j \leq n$, which means that $Q(y_i) = \frac{i-1}{n}$. Analogously, for each cutoff y_i we have $FN(y_i) = 0$ and $FP(y_i) = i - 1$ for the Clas-Pos model, and then, $Q(y_i) = \frac{i-1}{n}$. Hence the points in the *OCE* curves coincide for both models. We can apply the same rationale for the Regr-Min and Clas-Neg approaches.

Finally, for the Regr-Med approach, we have that $Q(y_i) = \frac{i-1}{n}$ for each $y_i < median(T)$, where T is the training dataset and $Q(y_i) = \frac{n-i}{n}$ for each $y_i \geq median(T)$. In the classification case, for each $y_i < median(T)$ the Clas-Maj model predicts all test examples as positive, so $Q(y_i) = \frac{i-1}{n}$. However, for each $y_i \geq median(T)$ the Clas-Maj approach predicts all test examples as negative, giving $Q(y_i) = \frac{n-i}{n}$. As a result, the *OCE* curves also coincide for both approaches. □

Theorem 6 *If the train and test medians are equal the median model is inflected at (0.5,0.5) in the OCE plot.*

Proof As we are assuming that the y_i values are in ascending order, the median corresponds to the cutoff placed at the middle point, which is $i = \frac{n+1}{2}$ (assuming n is even), whose true rank ratio is $R(i) = \frac{\frac{n+1}{2}-1}{n} = 0.5$. On the other hand, as $Q(y_i) = \frac{i-1}{n}$ for $i \in [1..\frac{n}{2}]$ and $Q(y_i) = \frac{n-i}{n}$ for $i \in]\frac{n}{2}..n]$, the *OCE* plot is an ascending curve for ratios $r \in [0..0.5]$ and a descending curve for $r \in]0.5..1]$. Finally, as the train and test medians are equal, for the cutoff at $r = 0.5$ half of the test examples are below and half are above the cutoff, that is $\bar{Q}(0.5) = \frac{\frac{n}{2}}{n} = 0.5$. □

This does not happen in Figure 7 as the medians are different (177858 vs. 184128). Finally, we are interested in the limits for A_{OCE}:

Theorem 7 *The area under the OCE curve, A_{OCE}, is always between 0 and 0.75 for regression models.*

Proof The minimum value is clearly obtained when $y_i = \hat{y}_i$ for every i, as Q equals 0. For the maximum values, we start with the results of Theorem 4, where $A_{OCE} = L_o = \frac{1}{n}\sum_{i=1}^{n} IR_i$. With this expression of A_{OCE} we can freely and independently choose the $\hat{y}_i$ for each y_i to see what the worst situation is. As we assume that the values of y_i are in ascending order, for y_1 we can actually put a very large $\hat{y}_1$, so we are in the case of Eq. 4 and we have that all the other $y_i < y_j$ are in between, so $IR_1 = \frac{n-1}{n}$. For y_2, we can only put $IR_2 = \frac{n-2}{n}$ elements in between. This can be generalised until $\frac{n}{2}$, assuming n is even, with $IR_i = \frac{n-i}{n}$. In a similar way, we can put a very small $\hat{y}_n$, and with the case of Eq. 4 we have that $IR_n = \frac{n}{n}$ as here $y_i \geq y_j$. So from $\frac{n}{2}+1$ we have $IR_i = \frac{i}{n}$. Putting both things together and making the change $j = n - i$ we have:

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} IR_i &= \frac{1}{n}\left\{\sum_{i=1}^{\frac{n}{2}} \frac{n-i}{n} + \sum_{i=\frac{n}{2}+1}^{n} \frac{i}{n}\right\} = \frac{1}{n}\left\{\sum_{j=\frac{n}{2}}^{n-1} \frac{j}{n} + \sum_{i=\frac{n}{2}+1}^{n} \frac{i}{n}\right\} \\
&= \frac{1}{n}\left\{\frac{1}{2} + 1 + \frac{2}{n}\sum_{j=\frac{n}{2}+1}^{n-1} j\right\} = \frac{1}{n}\left\{\frac{3}{2} + \frac{2}{n}\frac{(n-1)+(\frac{n}{2}+1)}{2}(\frac{n}{2}-1)\right\} \\
&= \frac{1}{n}\left\{\frac{3}{2} + \frac{1}{n}\frac{(2n-2+n+2)\frac{n-2}{2}}{2}\right\} = \frac{1}{n}\left\{\frac{3}{2} + \frac{3(n-2)}{4}\right\} = \frac{6+3(n-2)}{4n} = \frac{3}{4} = 0.75
\end{aligned}$$

A similar rationale works in approximately the same way when n is odd. Note that the above result does not depend on n, so the maximum is always achievable for any value of n. Nonetheless, for $n \to \infty$ we have the first half of the values decreasing smoothly from 1 to 0.5 and the second half of the values increasing smoothly from 0.5 to 1. □

The maximum value for regression models can be obtained as we have seen in the above theorem, provided we choose the test median or we estimate it perfectly. For the classification approach if we choose a very bad classifier for each cutoff we can theoretically have an A_{OCE} of 1.

Corollary 2 *When the train and test medians are equal, the A_{OCE} of the Regr-Med and Clas-Maj approaches 0.25.*

Proof From Theorem 6 we see that the inflection point is at 0.5. When the number of examples goes to infinity the curve of the Regr-Med converges to the ascending diagonal in the first part and the descending diagonal in the second part, by using the same rationale that we used in Theorem 5. Clearly, the area of this triangle is 0.25. □

6 Case study

In order to illustrate the binarised regression problem and the application of the plots and metrics we have seen in the previous sections, we are going to work with an illustrative example. This is the Zillow Data introduced in the running Example 1 in Section 1 with the cuttoff distribution extracted from [11] (see Figure 2, top).

We study the behaviour of eight techniques for this problem using RWeka [21,16] and R, as shown in Table 3. We chose five regression techniques for the reframing procedure (cutoff after prediction): linear regression (Regr-LnR), quantile regression (Regr-qr)[5] [24], k-nearest neighbours (kNN) [5] considering mean (Regr-kNN-mean) and median (Regr-kNN-median) among the neighbours for estimating predictions, and regression trees (Regr-M5P); and three classification techniques for the retraining procedure (cutoff before training): logistic regression (Clas-LgR), nearest neighbours (Clas-IBk) and decision trees (Clas-J48). All parameters were chosen by default in Weka, except for KNN, where we set the parameter k (number of neighbours) to 10. Our intention is to have three families: linear functions, lazy methods and trees, with two similar algorithms for regression and classification, so that the comparison could be more meaningful. The introduction of two linear regression methods was meant to compare one method that optimises MSE (Regr-LnR) and a method that optimises MAE (Regr-qr), given the connection seen between A_{UCE} and MAE. In a similar way, we compare two versions of the k-nearest neighbours: one method that optimises MSE (Regr-kNN-mean) and a method that optimises MAE (Regr-kNN-median) [17].

Acronym	Technique	Context-handling
Regr-LnR	Linear Regression	Reframing
Regr-qr	Quantile Regression	Reframing
Regr-kNN-med	Nearest Neighbours Median	Reframing
Regr-kNN-mean	Nearest Neighbours Mean	Reframing
Regr-M5P	Regression Tree	Reframing
Clas-LgR	Logistic Regression	Retraining
Clas-kNN	Nearest Neighbours	Retraining
Clas-J48	Decision Tree	Retraining

Table 3 Eight different approaches that are considered for the experiments throughout the paper.

We first split the dataset into 67% train and 33% test. For the five reframing approaches we learned the regression model once on the training set, and then we used the cutoffs for calculating the values of Q for each of them on the test set. For the three retraining approaches we converted the numerical output of the training set into a binary label for each cutoff and then we learned (for each

[5] Quantile regression aims at estimating either the conditional median or other quantiles of the goal variable.

case) a new classifier from each modified training set (note the computational cost of retraining so many models, one for each cutoff). Then each classifier was applied to the test set with each cutoff.

For the uniform distribution (the *UCE* plots), we generated n cutoffs (with n being the number of examples in the test set), but regularly distributed between the minimum and maximum values of all the true and predicted values. This mimics a uniform distribution. For the observed distribution (the *OCE* plots), we also generated n cutoffs, using the actual values on the test set as cutoffs. We calculated the areas under both curves for the eight methods as well as several additional metrics for the regression models in the reframing approach. Note that we cannot derive other metrics for the retraining approach, as there is no metric in the literature that evaluates a set of classifiers for a range of cutoffs. In fact, the area under the *UCE* and *OCE* curves, which both can be calculated for both the reframing (regression) and retraining (classification) approaches, is one of the main contributions of this paper, as they provide new measures for the binarised regression problem for both approaches (and the A_{UCE} is equivalent to *MAE* for the reframing approach, as we saw). Table 4 shows the areas of the *UCE* and *OCE* curves for each of the eight approaches and some other metrics for the reframing (regression) methods.

Approach	A_{UCE}	A_{OCE}	*MAE*	*MSE*	eVar	eBias	PCor	SCor	KCor
Regr-LnR	60308	0.107	60394	8191e6	8179e6	3510	0.896	0.898	0.734
Regr-kNN-med	50090	0.088	50317	8167e6	8165e6	-1508	0.895	0.897	0.744
Regr-kNN-mean	50691	0.091	51023	7101e6	7085e6	4037	0.910	0.899	0.747
Regr-M5P	66994	0.114	67101	12342e6	12339e6	-1809	0.838	0.860	0.685
Regr-qr	98303	0.175	98787	29390e6	27156e6	-47265	0.634	0.641	0.463
Clas-LgR	51105	0.081							
Clas-kNN	42653	0.076							
Clas-J48	54899	0.088							

Table 4 Results for the Zillow data and the approaches shown in Table 3. Metrics shown the area under the *UCE* and *OCE* curves for all approaches, and the mean absolute error (*MAE*), the mean squared error (*MSE*), the error variance (divided by n instead of $n-1$) and bias, as well as the Pearson, Spearman and Kendall correlation coefficients for the approaches based on regression models (reframing).

We can see some interesting things here. The methods based on classification (retraining) are equal or better than the methods based on regression. We cannot conclude anything from just one dataset (more datasets will be considered in Section 7). The results for Regr-kNN (mean and median) and Clas-kNN are close in A_{OCE}. In theory, as they use the same algorithm and the neighbours are calculated with the same features, the results should be very similar, but not always equal[6].

The metrics A_{UCE} and A_{OCE} are useful for comparing the two different procedures under the same rule, something that we cannot do with the other

[6] For example, consider three neighbours with outputs $y = \{1, 2, 6\}$, and a cutoff $c = 2.5$. If we consider equal weights and mean for the prediction, for regression, the average $\bar{y} = 4.5 \geq c = 2.5$ and predicts "above the cutoff", but for classification there is only one neighbour above the cutoff so it predicts "below the cutoff". Only for $k = 1$ would both approaches be equal.

metrics. However, the measures do not show whether the curves cross and whether there are different dominance regions. This is shown by the *CE*, *UCE* and *OCE* plots. Figure 8 shows the *UCE* and *OCE* plots, including the eight approaches. Given the different distributions, in the *OCE* plot, it is much easier to differentiate the eight models. We also have the diagonals for reference (representing the median model). In fact, we see that some models go beyond the diagonal on the left, which means that they are worse than the median model for low cutoffs (especially Regr-LnR, which has worse behaviour for the 10% lowest cutoffs (output value below 80000 approximately). The *OCE* plot and the corresponding areas are more meaningful (and clear) than the original uniform distribution in the *UCE* plots. Not only are the differences magnified but also the area gives more relevance to those cutoffs that correspond to more frequent values of the output values.

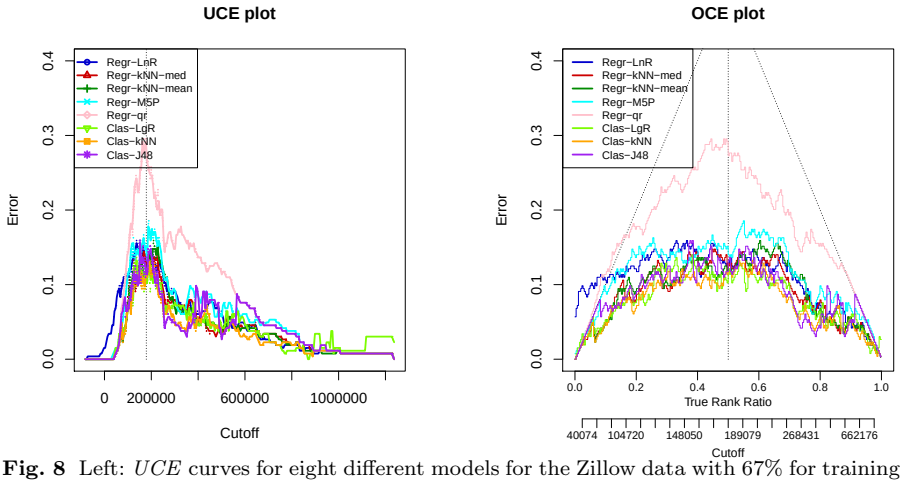

Fig. 8 Left: *UCE* curves for eight different models for the Zillow data with 67% for training and 33% for test. Right: Corresponding *OCE* curves.

If we look at the models, we see that the curves for Regr-kNN (median and mean) and Clas-kNN are very close. However, for Reg-LnR and Class-LgR and the two decision trees, the behaviour is notably different. In this particular case, it seems that Regr-LnR, Reg-M5P and Reg-qr can be safely discarded, as they are dominated for any possible cutoff.

We are also interested in the way dominance regions can be identified. Figure 9 shows the same eight models learned with 50% of the dataset and evaluated on different halves of the remaining data. This tries to emulate the common use of these curves to select and discard models and then use them for different deployment data. In other words, the left plot mimics the evaluation and selection with validation data and the right plot mimics the performance on deployment data. Despite the fact that this is a small dataset (799 examples), we can still identify which models present the better performance and the regions on one dataset and use them for the other. We can also determine which models can be directly discarded because they are dominated by

other models for all the range of possible cuttoffs. In this case we can reject Regr-LnR, Reg-M5P and Reg-qr (although Regr-M5P dominates for a small region on the right plot). We will explore this use of *OCE* and *CE* plots in the experiments of Section 7.

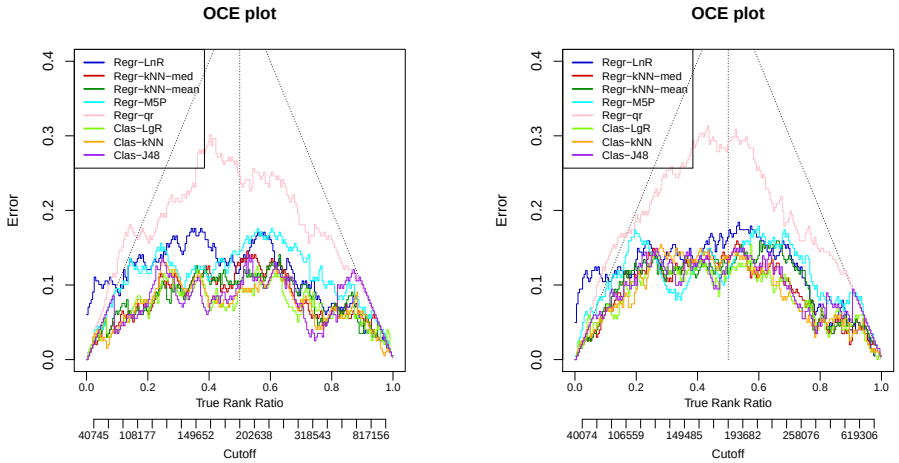

Fig. 9 *OCE* curves for eight different models for the Zillow case study with the same 50% of the data for training. The rest of the data is split into two halves, one is used for the plot on the left and the other for the plot on the right.

We can get more insight from this problem if we look at the behaviour when the training set becomes smaller. Figure 10 shows the *OCE* plots where the training set is now 33% (left) and 10% (right). We see a gradual trend if we also compare with Figure 8 (right). We see that performance is increasingly degraded for all the methods in general. We see a very dramatic degradation for Clas-J48, while its companion, Reg-M5P, stays relatively constant. We find a similar behaviour in the two methods based on linear models and in the regression versions of kNN. An explanation for this phenomenon is that when the dataset becomes smaller, the discretisation stage before training reduces the available information significantly, and the models are worse, especially those with risk of overfitting (linear models seem to be more robust here). This is related to the discretisation (binarisation) procedure, which creates imbalanced datasets for very low and very high cutoffs.

All this suggests that there is a dilemma for the retraining approach using classification. If the training dataset is small, the results may be poorer than for reframing. However, if the training dataset is large, retraining seems better, but keeping the training data forever and retraining each time a new cutoff is given becomes expensive.

In this section we have seen that the *OCE* plots are very informative about the behaviour of several approaches and techniques for a real dataset. Some intuitive rationales, however, have made upon one (illustrative) dataset. The following section analyses some of these issues more systematically.

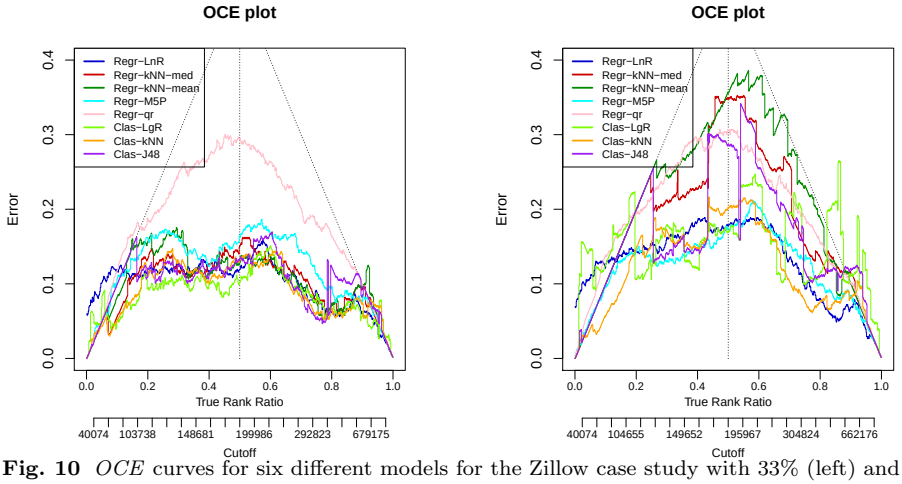

Fig. 10 *OCE* curves for six different models for the Zillow case study with 33% (left) and 10% (right) for training and 67% and 90% respectively for test.

7 Experiments comparing retraining and reframing

In this section we aim at establishing whether some of the observations in the previous section hold in general. In order to do that, we will analyse the performance of the reframing and retraining approaches over several datasets for different regression and classification techniques respectively. Table 5 includes a summary of the features of the twenty datasets that we use[7]: number of instances; number of attributes; average, standard deviation, median, maximum and minimum of the target value; and the parameters of the Beta distribution used to model the actual distribution of cutoffs (when $\alpha = \beta = 1$ the distribution is actually uniform), and the range of the target value employed for it. This range is in percentage w.r.t. the minimum and maximum values of the output value in the dataset.

In our analysis, we investigate the influence of the range of the cutoff region, considering narrow ranges those $< 50\%$ of the complete range, i.e., datasets 1, 2, 3, 8 and 19. We also analyse the influence of small vs. large datasets. We consider a dataset small if the number of instances is < 1000, i.e., datasets 2, 3, 6, 7, 8, 9, 12, 14, 17 and 20. Finally, we also distinguish those datasets that are evaluated with a uniform-distributed cutoff distribution (1, 2, 3), those with an output-distributed cutoff distribution (5, 7, 8, 9, 12, 13, 15, 20) using values of α and β that resemble the output distribution, and the rest (with other distributions using other parameters of α and β).

The goals of the experiments are:

- Analyse whether a method that optimises *MSE* is worse or better than a similar method that optimises *MAE*.
- Analyse the two alternative solutions for the binarised regression problem: retraining and reframing.

[7] See http://www.dsic.upv.es/~flip/BinarisedRegression/.

– Analyse whether the hybrid model (using the best model for each cutoff region it dominates) is best, or the selection of one single model using some aggregated measure can also be good.
– Analyse the items above depending on different actual cutoff distributions, wide vs narrow ranges, size of datasets, etc.

We consider the following experimental scenario. Each dataset is split into a training dataset (50%) and test dataset (50%). For Table 9, the partition is (50%) for training, (25%) for validation (model selection) and (25%) for test (model evaluation). As we did in Section 6 we use three different families of learning methods: linear models (linear regression and quantile regression), K-nearest neighbours (median and mean versions for regression) and Decision trees (M5P), all with the same configuration as shown in Table 3.

Id	dataset	#Inst	#Att	Ave	SD	Med	Max	Min	Alfa	Beta	MinR	MaxR	Type
1	airfoil	1503	6	124.84	6.90	125.72	140.99	103.38	1	1	0.30	0.75	2
2	data_cooling	768	9	24.59	9.51	22.08	48.03	10.90	1	1	0.30	0.65	2
3	data_heating	768	9	22.31	10.09	18.95	43.10	6.01	1	1	0.30	0.65	2
4	CCPP	9568	5	454	17.07	451	495	420	2	2	0.15	0.85	5
5	Concrete	1030	9	35.82	16.71	34.45	82.60	2.33	2	2	0.15	0.85	3
6	forestfires	517	13	12.85	63.66	0.52	1090	0	1	5	0	1	5
7	housing	506	14	22.53	9.20	21.20	50	5	2	2	0.20	0.70	3
8	autoMpg	398	8	23.51	7.82	23	46.60	9	2	4	0.20	0.60	4
9	yacht_hydr	308	7	10.50	15.16	3.06	62.42	0.01	1	1	0.30	0.60	3
10	wine-white	4898	12	5.88	0.89	6	9	3	2	2	0.10	0.90	5
11	wine-red	1599	12	5.64	0.81	6	8	3	2	2	0.10	0.90	5
12	solar-flare_1	323	11	0.29	0.77	0	6	0	2	6	0	0.70	3
13	solar-flare_2	1066	11	0.35	0.95	0	8	0	2	6	0	0.70	3
14	dee	365	7	2.97	0.97	2.79	5.12	0.77	2	2	0	1	5
15	plastic	1650	3	15	3.42	15	20	10	1	1	0	1	3
16	treasury	1049	16	7.52	3.38	6.61	20.76	3.02	1.5	3	0	1	5
17	wankara	321	10	48.92	14.98	47.70	81.60	16.20	2	2	0.20	0.80	5
18	wizmir	1461	10	61.51	14.38	60	89.90	29.40	2	2	0.20	0.80	5
19	cpu_small	8192	13	83.97	18.40	89	99	0	2	2	0.30	0.70	6
20	auto_price	159	16	11445	5877	9233	35056	5118	2	2	0	0.70	3

Table 5 Features of the 20 datasets employed in the experiments: number of instances, number of attributes; average, standard deviation, median, maximum and minimum of the target value; parameters of the Beta distribution used to model the actual distribution of cutoffs and range of the target value employed for it (in percentage w.r.t. the minimum and maximum values of the dataset). Last column shows which type each dataset belongs to (see Table 1).

In order to increase the significance of the results we perform ten iterations for each dataset. In these cases we estimate the same metrics we used for the case study in the previous section, as shown in Table 4 with the following variations. In order to take the range of cutoffs into account, we will use A_{PUCE} instead of A_{UCE}. Analogously, we also include the A_{CE} as a measure to better analyse case **b**. When not specified, the models in subsequent tables are evaluated with the true cutoff distribution, A_{CE}.

We first analyse in Table 6 the two alternative solutions for the binarised regression problem: retraining and reframing. For linear regression we only use LnR (not qr) and for kNN we only use the mean version (not the median version). The results of this table are difficult to synthesise, but we see that generally models based on decision trees give good results for both the retraining and the reframing approaches. However, we can find some datasets (e.g. 7, 8, 14, 15, 17, 18, 19) where Logistic Regression is obtaining a bet-

ter performance than J48 in the retraining scenario, whilst the situation is inverse in the reframing scenario. On the other hand, A_{PUCE} approximates the *MAE* quite well for the Regr-LnR approach and for those datasets with a wide range of cutoffs (type 1, 3 and 5), being the values slightly different because the A_{PUCE} is approximated with a finite number of points. Overall, for small datasets reframing methods seem competitive against retraining methods. For large datasets, however, the retraining approach looks generally better. Nonetheless, large datasets represent a situation where retraining requires more computational time. In summary, depending on the application a trade-off should be found.

In Table 7 we compare the default linear regression method (Regr-LnR, which uses quantitative response optimisation of *MSE*) with the quantile regression (Regr-qr, which optimises for *MAE*). Some values are missing because some predictions were null for Regr-qr. With this experiment we try to analyse whether the methods that are devised for *MAE* are better for the binarised regression problem, as the connection between this problem and the *MAE* metric may suggest. At first sight, we observe that it is true that linear regression and quantile regression optimise for different metrics. This is obvious in datasets 9, 12, 13, 19 and 20. But if we look at the columns A_{CE}, A_{PUCE} or A_{OCE}, we do not see that the results are better for Regr-qr always (it is only the case for 6, 12, 13, 15 and 20). In fact, Regr-LnR obtains better results than Regr-qr for *MAE* and *MSE* in datasets 1, 5 and 17. For the rest of datasets, the differences between both methods are small. Therefore, we think that we cannot yet conclude that optimising regression models for *MAE* is necessarily better than optimising them for *MSE* for the binarised regression problem.

A similar analysis is shown in Table 8. Here we compare two versions of the k-nearest neighbours: one method that optimises *MSE* (Regr-kNN-mean) and a method that optimises *MAE* (Regr-kNN-med) [17]. The results show the effect of these optimisations in most of the datasets: Regr-kNN-mean usually gets better performance in MSE while Regr-kNN-med obtains better results in MAE. In this case we have more evidence than for the linear regression case.

In Table 9 we aggregate all techniques per dataset and compare model selection when one single model is chosen using different aggregated measures (A_{CE}, A_{OCE} or A_{PUCE}) with hybrid models and the best for the test. The first three are chosen as options for model selection using the validation data for the cases **b** (the cutoff distribution is known), **d** (nothing is known so we assume output-distributed cutoffs) and **c** (the partial cutoff range is known) respectively, as described in Section 2.2. These are denoted by Val_{CE}, Val_{OCE} and Val_{PUCE} in the table. In addition, we compare with a hybrid model (using the best model for each cutoff region it dominates) combining the models where they are best (denoted by Val_{Hybrid}). For all of them, we use the true distribution for model evaluation with the test set. We also add two ideal (unreachable) models for comparison: the optimal hybrid model using the test set, denoted by $Test_{Hybrid}$, and the model with lowest area using the true cutoff distribution on the test set (denoted by $Test_{CE}$). What we see in these results is that the selection method using the aggregated measures for the

		Reframing			Retraining									
Id	Met	A_{CE}	A_{PUCE}	A_{OCE}	A_{CE}	A_{PUCE}	A_{OCE}	*MAE*	*MSE*	eVar	eBias	PCor	SCor	KCor
	Lr	2.97	2.97	0.18	2.77	2.78	0.16	3.73	23.28	23.25	-0.02	0.71	0.69	0.51
1	kNN	4.12	4.13	0.24	3.05	3.07	0.18	4.84	37.67	37.59	-0.16	0.49	0.45	0.31
	Trees	1.96	1.96	0.12	2.00	2.01	0.12	2.48	11.21	11.20	0.06	0.87	0.84	0.67
	Lr	0.99	1.00	0.08	0.83	0.84	0.06	2.26	10.28	10.26	0.02	0.94	0.93	0.78
2	kNN	0.85	0.86	0.07	0.92	0.93	0.07	1.99	7.29	7.26	0.07	0.96	0.92	0.76
	Trees	0.62	0.63	0.05	0.61	0.62	0.05	1.31	3.82	3.81	-0.01	0.98	0.98	0.89
	Lr	0.60	0.61	0.05	0.56	0.57	0.04	2.05	8.43	8.42	0.01	0.96	0.95	0.81
3	kNN	0.71	0.71	0.05	0.65	0.66	0.05	2.17	8.92	8.78	0.32	0.96	0.93	0.77
	Trees	0.37	0.37	0.03	0.17	0.17	0.01	0.88	1.52	1.52	-0.02	0.99	0.99	0.92
	Lr	3.14	3.24	0.06	2.91	2.99	0.05	3.63	20.74	20.73	0.01	0.96	0.96	0.83
4	kNN	2.86	2.91	0.05	2.69	2.75	0.05	3.18	18.76	18.75	-0.01	0.97	0.96	0.84
	Trees	2.80	2.87	0.05	2.69	2.77	0.05	3.14	16.67	16.66	0.02	0.97	0.97	0.84
	Lr	8.84	7.88	0.16	7.75	6.56	0.14	8.45	115.93	115.60	0.40	0.77	0.79	0.60
5	kNN	8.99	7.94	0.16	9.14	8.13	0.16	8.50	123.06	122.44	-0.73	0.75	0.76	0.56
	Trees	5.54	4.76	0.10	5.81	5.10	0.10	5.16	47.96	47.81	0.31	0.91	0.91	0.75
	Lr	80.13	19.77	0.07	75.79	40.86	0.07	21.13	3488	3462	0.43	0.04	0.04	0.03
6	kNN	45.84	12.01	0.04	45.54	11.95	0.04	13.02	3441	3312	-11	0.00	0.07	0.05
	Trees	81.08	20.65	0.07	45.91	12.13	0.04	21.81	4164	4121	1.22	0.00	0.03	0.02
	Lr	2.64	2.44	0.12	2.17	2.07	0.10	3.49	23.83	23.67	0.20	0.85	0.88	0.71
7	kNN	3.95	3.56	0.18	3.00	2.82	0.13	4.65	54.16	51.78	-1.51	0.63	0.70	0.53
	Trees	2.09	1.95	0.09	2.33	2.22	0.10	2.67	15.82	15.76	0.01	0.90	0.89	0.73
	Lr	1.43	1.46	0.09	1.46	1.46	0.10	2.43	10.18	10.13	0.15	0.91	0.93	0.78
8	kNN	1.73	1.95	0.12	1.45	1.58	0.10	3.05	17.61	17.39	-0.34	0.84	0.87	0.70
	Trees	1.37	1.39	0.09	1.62	1.81	0.11	2.28	9.63	9.56	0.13	0.92	0.94	0.79
	Lr	1.53	1.55	0.08	0.33	0.33	0.02	7.35	85.09	84.50	-0.42	0.81	0.99	0.94
9	kNN	2.81	2.84	0.15	2.64	2.71	0.14	8.25	238.52	195.89	-6.52	0.53	0.70	0.51
	Trees	0.19	0.19	0.01	0.21	0.22	0.01	1.28	5.00	4.92	-0.25	0.99	1.00	0.95
	Lr	0.78	0.58	0.16	0.70	0.53	0.15	0.59	0.57	0.57	0.01	0.52	0.55	0.43
10	kNN	0.84	0.62	0.17	0.68	0.51	0.14	0.63	0.78	0.78	-0.01	0.31	0.32	0.28
	Trees	0.76	0.57	0.16	0.69	0.52	0.14	0.57	0.54	0.54	0.01	0.56	0.58	0.46
	Lr	0.66	0.50	0.16	0.56	0.44	0.14	0.51	0.43	0.43	0.01	0.58	0.58	0.47
11	kNN	0.73	0.55	0.18	0.59	0.45	0.15	0.56	0.67	0.67	-0.07	0.30	0.31	0.28
	Trees	0.66	0.50	0.16	0.61	0.47	0.15	0.51	0.43	0.43	0.01	0.58	0.59	0.47
	Lr	0.65	0.40	0.15	0.61	0.39	0.14	0.42	0.66	0.65	-0.05	0.25	0.24	0.21
12	kNN	0.55	0.30	0.13	0.52	0.29	0.12	0.32	0.77	0.68	-0.30	0.08	0.06	0.06
	Trees	0.63	0.39	0.15	0.53	0.29	0.13	0.41	0.65	0.64	-0.06	0.27	0.26	0.24
	Lr	0.74	0.47	0.13	0.65	0.39	0.12	0.48	0.84	0.84	-0.03	0.40	0.36	0.31
13	kNN	0.63	0.36	0.11	0.62	0.36	0.11	0.37	1.09	0.97	-0.34	0.12	0.10	0.10
	Trees	0.75	0.47	0.13	0.64	0.37	0.12	0.48	0.86	0.85	-0.03	0.38	0.35	0.30
	Lr	0.40	0.32	0.09	0.39	0.32	0.09	0.32	0.18	0.17	-0.00	0.90	0.90	0.73
14	kNN	0.42	0.33	0.10	0.42	0.34	0.10	0.33	0.21	0.20	-0.01	0.88	0.88	0.71
	Trees	0.41	0.32	0.09	0.47	0.38	0.11	0.33	0.18	0.18	-0.01	0.90	0.90	0.72
	Lr	3.00	3.00	0.30	1.06	1.07	0.11	3.01	11.71	11.71	0.01	0.06	0.06	0.04
15	kNN	1.18	1.18	0.12	1.17	1.17	0.12	1.18	2.85	2.78	-0.25	0.88	0.88	0.76
	Trees	1.19	1.19	0.12	1.38	1.39	0.14	1.24	2.36	2.36	-0.02	0.89	0.90	0.76
	Lr	0.20	0.15	0.01	0.25	0.18	0.01	0.15	0.06	0.06	-0.00	1.00	1.00	0.95
16	kNN	0.37	0.32	0.02	0.27	0.20	0.01	0.32	0.49	0.49	-0.03	0.98	0.99	0.93
	Trees	0.20	0.14	0.01	0.24	0.18	0.01	0.14	0.05	0.05	-0.00	1.00	1.00	0.95
	Lr	1.01	1.01	0.03	1.73	1.78	0.04	1.26	2.75	2.73	0.01	0.99	0.99	0.94
17	kNN	1.23	1.25	0.03	2.02	2.21	0.05	1.69	4.85	4.79	0.06	0.99	0.99	0.93
	Trees	0.90	0.90	0.02	1.81	1.85	0.05	1.08	2.01	1.99	-0.01	1.00	1.00	0.95
	Lr	0.67	0.68	0.02	0.78	0.79	0.02	0.91	1.56	1.56	0.00	1.00	1.00	0.95
18	kNN	0.84	0.87	0.02	1.14	1.20	0.03	1.22	2.65	2.65	-0.00	0.99	0.99	0.93
	Trees	0.66	0.66	0.02	1.07	1.07	0.03	0.88	1.43	1.43	0.00	1.00	1.00	0.95
	Lr	0.44	0.54	0.01	0.13	0.17	0.00	6.21	98.22	98.13	-0.08	0.85	0.76	0.59
19	kNN	0.20	0.33	0.01	0.15	0.21	0.00	4.56	46.34	45.23	1.05	0.93	0.75	0.58
	Trees	0.14	0.19	0.00	0.17	0.23	0.00	2.38	12.17	12.17	-0.03	0.98	0.93	0.79
	Lr	1954	1849	0.09	2251	2120	0.11	2063	$83{\cdot}10^5$	$81{\cdot}10^5$	-54	0.88	0.90	0.73
20	kNN	1960	1784	0.09	2736	2486	0.13	1945	$112{\cdot}10^5$	$104{\cdot}10^5$	-825	0.85	0.92	0.75
	Trees	1805	1679	0.09	1867	1679	0.09	1864	$73{\cdot}10^5$	$72{\cdot}10^5$	-121	0.89	0.92	0.76

Table 6 Results for the 20 datasets of Table 5 as binarised regression problems. We use 3 techniques for reframing (qr and kNNmean are excluded here) and 3 techniques for retraining as in Table 3.

whole range (Val_{CE}, Val_{OCE}) are equal, i.e., they always choose the same model in all cases. The use of a partial region makes things slightly different for a few datasets (2, 6, 7, 8, 14 and 20), and indistinguishable (equal) for the rest. All the datasets showing difference are of small size, so this may be caused by a high variability in the curves for small validation sets, as we saw in the analysis of the test case. The interesting thing is when we compare the best single method with a hybrid method. We see that the hybrid method, which was supposed to be better and independent of the distribution, is indeed better in some cases but worse in others. There seems to be no pattern for

Id	Approach	A_{CE}	A_{PUCE}	A_{OCE}	MAE	MSE	eVar	eBias	PCor	SCor	KCor
1	Regr-LnR	2.97	2.97	0.18	3.73	23.28	23.25	-0.02	0.71	0.69	0.51
	Regr-qr	2.95	2.95	0.17	3.78	24.52	24.46	-0.15	0.70	0.70	0.52
2	Regr-LnR	0.99	1.00	0.08	2.26	10.28	10.26	0.02	0.94	0.93	0.78
	Regr-qr	1.02	1.03	0.08	2.26	11.00	10.98	-0.01	0.94	0.92	0.76
3	Regr-LnR	0.60	0.61	0.05	2.05	8.43	8.42	0.01	0.96	0.95	0.81
	Regr-qr	0.69	0.69	0.05	2.05	9.06	9.06	-0.05	0.95	0.95	0.82
4	Regr-LnR	3.14	3.24	0.06	3.63	20.74	20.73	0.01	0.96	0.96	0.83
	Regr-qr	3.11	3.21	0.06	3.62	20.91	20.89	-0.10	0.96	0.96	0.83
5	Regr-LnR	8.84	7.88	0.16	8.45	115.93	115.60	0.40	0.77	0.79	0.60
	Regr-qr	8.43	7.61	0.15	8.55	143.86	140.90	1.66	0.74	0.81	0.62
6	Regr-LnR	80.13	19.77	0.07	21.13	3488.17	3462.34	0.43	0.04	0.04	0.03
	Regr-qr	45.69	12.18	0.04	12.97	3446.57	3307.60	-11.57	0.01	0.05	0.04
7	Regr-LnR	2.64	2.44	0.12	3.49	23.83	23.67	0.20	0.85	0.88	0.71
	Regr-qr	2.20	2.12	0.10	3.21	24.25	23.52	-0.79	0.85	0.88	0.72
8	Regr-LnR	1.43	1.46	0.09	2.43	10.18	10.13	0.15	0.91	0.93	0.78
	Regr-qr	1.73	1.76	0.11							
9	Regr-LnR	1.53	1.55	0.08	7.35	85.09	84.50	-0.42	0.81	0.99	0.94
	Regr-qr	2.81	2.85	0.15	6.55	133.61	117.73	-3.97	0.81	0.99	0.93
10	Regr-LnR	0.78	0.58	0.16	0.59	0.57	0.57	0.01	0.52	0.55	0.43
	Regr-qr	0.79	0.59	0.16	0.59	0.58	0.58	-0.03	0.52	0.54	0.42
11	Regr-LnR	0.66	0.50	0.16	0.51	0.43	0.43	0.01	0.58	0.58	0.47
	Regr-qr	0.65	0.49	0.16	0.50	0.44	0.44	-0.03	0.58	0.58	0.46
12	Regr-LnR	0.65	0.40	0.15	0.42	0.66	0.65	-0.05	0.25	0.24	0.21
	Regr-qr	0.53	0.29	0.13	0.31	0.72	0.65	-0.25	0.25	0.27	0.22
13	Regr-LnR	0.74	0.47	0.13	0.48	0.84	0.84	-0.03	0.40	0.36	0.31
	Regr-qr	0.62	0.36	0.11	0.37	1.02	0.93	-0.31	0.27	0.36	0.29
14	Regr-LnR	0.40	0.32	0.09	0.32	0.18	0.17	-0.00	0.90	0.90	0.73
	Regr-qr	0.41	0.33	0.09	0.33	0.18	0.18	-0.01	0.90	0.90	0.73
15	Regr-LnR	3.00	3.00	0.30	3.01	11.71	11.71	0.01			
	Regr-qr	1.15	1.15	0.12	1.23	2.37	2.34	-0.16	0.89	0.90	0.76
16	Regr-LnR	0.20	0.15	0.01	0.15	0.06	0.06	-0.00	1.00	1.00	0.95
	Regr-qr	0.19	0.14	0.01	0.14	0.06	0.06	-0.03	1.00	1.00	0.95
17	Regr-LnR	1.01	1.01	0.03	1.26	2.75	2.73	0.01	0.99	0.99	0.94
	Regr-qr	1.03	1.04	0.03	1.29	2.87	2.85	-0.08	0.99	0.99	0.93
18	Regr-LnR	0.67	0.68	0.02	0.91	1.56	1.56	0.00	1.00	1.00	0.95
	Regr-qr	0.67	0.69	0.02	0.91	1.56	1.56	-0.04	1.00	1.00	0.95
19	Regr-LnR	0.44	0.54	0.01	6.21	98.22	98.13	-0.08	0.85	0.76	0.59
	Regr-qr	0.63	0.67	0.02	4.13	168.42	167.10	1.11	0.78	0.91	0.77
20	Regr-LnR	1954	1849	0.09	2063	$83{\cdot}10^5$	$81{\cdot}10^5$	-54.00	0.88	0.90	0.73
	Regr-qr	1818	1758	0.09	1948	$86{\cdot}10^5$	$81{\cdot}10^5$	-298.67	0.88	0.91	0.75

Table 7 Results of Linear Regression and Quantile Regression. These results show the average of 10 iterations (50% train - 50 % test) for the 20 datasets.

this depending on the size of the dataset or the true cutoff distribution. This suggests that the choice of one single model using A_{CE}, A_{OCE} or A_{PUCE} can be a reasonably good option in comparison to keeping several models for different regions. In the end, this shows that these aggregated metrics show a good behaviour even if the distribution is different to the one used for the aggregation.

8 Related work

This work has focussed on a common problem where the dependent variable is given as numeric but the decision is binary according to a cutoff. This can be seen as a hybrid between regression and classification, which can actually be solved by regression and classification methods, as we have seen. The idea of solving one kind of problem by adapting techniques for other problems is not uncommon in the literature [28]. In this way, multiclass classification can be solved with binary classification, and ordinal regression is frequently addressed with classification and regression. Also, some classification techniques were originally regression techniques and vice versa (e.g., logistic regression, kNN, neural networks, SVM, etc.). Regarding the connection between regression and classification, we find in the literature that regression problems can

Id	Approach	A_{CE}	A_{PUCE}	A_{OCE}	*MAE*	*MSE*	eVar	eBias	PCor	SCor	KCor
1	Regr-kNN-med	4.12	4.13	0.24	4.84	37.67	37.59	-0.16	0.49	0.45	0.31
	Regr-kNN-mean	4.04	4.05	0.24	4.76	35.24	35.16	0.23	0.51	0.47	0.33
2	Regr-kNN-med	0.85	0.86	0.07	1.99	7.29	7.26	0.07	0.96	0.92	0.76
	Regr-kNN-mean	0.85	0.85	0.07	1.97	6.88	6.86	0.10	0.96	0.93	0.76
3	Regr-kNN-med	0.71	0.71	0.05	2.17	8.92	8.78	0.32	0.96	0.93	0.77
	Regr-kNN-mean	0.71	0.72	0.06	2.16	8.35	8.26	0.26	0.96	0.93	0.77
4	Regr-kNN-med	2.86	2.91	0.05	3.18	18.76	18.75	-0.01	0.97	0.96	0.84
	Regr-kNN-mean	2.85	2.91	0.05	3.18	17.73	17.72	-0.04	0.97	0.96	0.84
5	Regr-kNN-med	8.99	7.94	0.16	8.50	123.06	122.44	-0.73	0.75	0.76	0.56
	Regr-kNN-mean	8.77	7.80	0.16	8.36	114.05	113.50	-0.68	0.77	0.78	0.58
6	Regr-kNN-med	45.84	12.01	0.04	13.02	3441	3312	-11.12	0.00	0.07	0.05
	Regr-kNN-mean	75.54	18.54	0.07	19.71	3623	3605	-0.18	0.05	0.01	0.00
7	Regr-kNN-med	3.95	3.56	0.18	4.65	54.16	51.78	-1.51	0.63	0.70	0.53
	Regr-kNN-mean	4.40	3.89	0.20	4.97	51.87	51.70	0.07	0.63	0.67	0.49
8	Regr-kNN-med	1.73	1.95	0.12	3.05	17.61	17.39	-0.34	0.84	0.87	0.70
	Regr-kNN-mean	1.76	1.98	0.12	3.06	17.18	17.03	0.12	0.85	0.87	0.70
9	Regr-kNN-med	2.81	2.84	0.15	8.25	238.52	195.89	-6.52	0.53	0.70	0.51
	Regr-kNN-mean	2.79	2.82	0.15	9.46	196.18	193.93	-1.22	0.46	0.59	0.43
10	Regr-kNN-med	0.84	0.62	0.17	0.63	0.78	0.78	-0.01	0.31	0.32	0.28
	Regr-kNN-mean	0.88	0.65	0.18	0.66	0.68	0.68	0.02	0.38	0.39	0.31
11	Regr-kNN-med	0.73	0.55	0.18	0.56	0.67	0.67	-0.07	0.30	0.31	0.28
	Regr-kNN-mean	0.78	0.58	0.20	0.59	0.58	0.58	-0.03	0.35	0.36	0.29
12	Regr-kNN-med	0.55	0.30	0.13	0.32	0.77	0.68	-0.30	0.08	0.06	0.06
	Regr-kNN-mean	0.68	0.40	0.16	0.43	0.66	0.65	-0.07	0.21	0.23	0.19
13	Regr-kNN-med	0.63	0.36	0.11	0.37	1.09	0.97	-0.34	0.12	0.10	0.10
	Regr-kNN-mean	0.78	0.51	0.14	0.52	0.92	0.92	-0.05	0.26	0.27	0.23
14	Regr-kNN-med	0.42	0.33	0.10	0.33	0.21	0.20	-0.01	0.88	0.88	0.71
	Regr-kNN-mean	0.41	0.32	0.09	0.33	0.19	0.19	0.01	0.89	0.89	0.71
15	Regr-kNN-med	1.18	1.18	0.12	1.18	2.85	2.78	-0.25	0.88	0.88	0.76
	Regr-kNN-mean	1.28	1.28	0.13	1.28	2.51	2.50	-0.02	0.89	0.88	0.74
16	Regr-kNN-med	0.37	0.32	0.02	0.32	0.49	0.49	-0.03	0.98	0.99	0.93
	Regr-kNN-mean	0.41	0.34	0.02	0.34	0.49	0.49	-0.01	0.98	0.99	0.93
17	Regr-kNN-med	1.23	1.25	0.03	1.69	4.85	4.79	0.06	0.99	0.99	0.93
	Regr-kNN-mean	1.11	1.11	0.03	1.51	3.87	3.82	0.04	0.99	0.99	0.93
18	Regr-kNN-med	0.84	0.87	0.02	1.22	2.65	2.65	-0.00	0.99	0.99	0.93
	Regr-kNN-mean	0.79	0.81	0.02	1.14	2.38	2.37	-0.01	0.99	0.99	0.94
19	Regr-kNN-med	0.20	0.33	0.01	4.56	46.34	45.23	1.05	0.93	0.75	0.58
	Regr-kNN-mean	0.20	0.33	0.00	4.59	42.58	42.54	0.13	0.94	0.76	0.59
20	Regr-kNN-med	1960	1784	0.09	1945	$112{\cdot}10^5$	$104{\cdot}10^5$	-825	0.85	0.92	0.75
	Regr-kNN-mean	1809	1669	0.09	1828	$95{\cdot}10^5$	$90{\cdot}10^5$	-524	0.87	0.92	0.75

Table 8 Results of kNN using median and mean as the operator to predict new values from the selected neighbours. These results show the average of 10 iterations (50% train - 50 % test) for the 20 datasets.

be addressed by using classification techniques. Thus, 'quantile regression' is solved in [25] by learning a family of binary classifiers c_t with $t \in [0, 1]$ that are then used to make more accurate q-quantile estimations; 'regression' is solved by first discretising the problem for learning a classifier and then transforming their outputs into numeric values as in [36] and [23] (that uses cost-sensitive classification). Note that, in the mentioned approaches the original 'regression problem' does not change but is solved by using classification techniques whereas our approach addresses the more general situation in which a regression problem (at the learning time) is turned into a classification one (at the deployment time). A somewhat related approach for the opposite case, i.e., turning a classification problem into a regression one, is presented in [26], where the authors defined an algorithm to transform a binary classification problem into the problem of estimating class probability membership.

In our case the connection between regression and classification originates from a particular kind of problem. In addition, we have a volatile cutoff, so the mapping cannot be done once and for all. Either the context is applied to the training set or the decision rule has to take it into account. So our work is more closely related to an area of research that is concerned about model

Id	Val_{CE}	Val_{OCE}	Val_{PUCE}	Val_{Hybrid}	$Test_{Hybrid}$	$Test_{CE}$
1	1.81	1.81	1.81	1.85	1.76	1.81
2	0.59	0.59	0.60	0.59	0.51	0.58
3	0.16	0.16	0.16	0.16	0.14	0.16
4	2.69	2.69	2.69	2.66	2.55	2.68
5	5.55	5.55	5.55	5.45	5.10	5.46
6	43.92	43.92	44.23	43.69	42.51	43.16
7	2.02	2.02	2.04	2.06	1.69	1.96
8	1.40	1.40	1.31	1.43	1.06	1.26
9	0.21	0.21	0.21	0.21	0.12	0.20
10	0.68	0.68	0.68	0.68	0.66	0.68
11	0.57	0.57	0.57	0.58	0.55	0.56
12	0.53	0.53	0.53	0.53	0.49	0.51
13	0.63	0.63	0.63	0.63	0.60	0.61
14	0.40	0.40	0.41	0.40	0.32	0.38
15	1.08	1.08	1.08	1.10	1.06	1.08
16	0.19	0.19	0.19	0.19	0.11	0.18
17	0.98	0.98	0.98	1.14	0.66	0.98
18	0.65	0.65	0.65	0.70	0.55	0.64
19	0.12	0.12	0.12	0.13	0.11	0.12
20	1733	1733	1829	1656	1068	1532

Table 9 Results of four model selection strategies: three based on the aggregated measures using the whole range (Val_{CE}, Val_{OCE} and Val_{PUCE}), one hybrid model using the validation set. The last two columns show two ideal models for comparison: the optimal hybrid model using the test set and the model with lowest area using the true cutoff distribution on the test set.

generation, adaptation and evaluation when we consider volatile conditions at deployment time [22,37].

This context can be parametrised in the form of loss functions, output variable distribution, attribute costs, etc. In this scenario, there is an increasingly common perspective of addressing this problem by using graphical representations where the context can appear in one or more of the axes of the plot. In these plots, we can determine dominance models if the context regions are given, or otherwise we can estimate the expected loss for a range or distribution of context. Many of these works are inspired by ROC analysis [34,9] and ROC isometrics [14], which were originally introduced for binary classification. Just for classification there are many other representations, such as cost curves [8,7], DET curves [27], lift charts [29], and calibration maps [6]. A survey on graphical methods of predictive performance evaluation for classification can be found in [31]. The connection of many of these graphical representation with the existing performance metrics for classification is becoming more clear throughout the years [13,20].

However, the interest in adapting graphical representations for regression is more recent, and we can find the so-called Regression Error Characteristic (REC) Curves [4] and the Regression Error Characteristic Surfaces [35], which rely on the concept of tolerance, in order to consider that an error (which can be the squared error, the absolute deviation or any other loss function) is admissible or not. The surfaces also include an additional dimension for the output value. The notion of context is then very different for REC curves and surfaces, as it is the tolerance to the *magnitude* of the error (and the range of the output value for the surfaces) and not the *sign* of the error, as with the binarised regression transformation to classification. In other words, in REC curves and surfaces, the regression model is evaluated in a discretised way as

for having an *error* below or above the tolerance or the output in a region, but not the *output* below or above a cutoff. The so-called partial REC curves, also introduced in [35], may look similar to our analysis of the *CE* curves for partial regions of the cutoff. However, partial REC curves show the behaviour of the model for a *subset* of examples according to a range of output values, whereas "partial" *CE* curves analyse *all* the examples with a partial range of cutoffs. This difference is well illustrated by the newly introduced notion of clipped *MAE*, which is not a *MAE* for a subset of examples but a *MAE* for which the absolute error is clipped for all examples according to the cutoff region. Overall, the notion of context in REC curves and surfaces is quite different to the notion of cutoff, and REC curves and surfaces are intended for the general problem of regression. Note that we can plot *CE* curves for a set of classification models, as for the retraining approach, which is obviously impossible for REC curves or surfaces.

Other graphical representations for regression that lead to a metric are the ROC curves for regression (RROC curves) [18]. In this case, the loss is the asymmetric cost and the area under the ROC curves for regression is shown to be the error variance. While there might be interesting connections to unveil between ROC curves for regression and the *UCE* and *OCE* curves, yet again the problem that the ROC curves for regression address is traditional regression, where the asymmetry of the cost is the operating context.

Finally, the approach that may look more closely related to this paper is an understanding of regression models as rankers [32]. Here, Rosset *et al.* stated the problem as being able to select the p percent of examples with higher output value, focussing on order and the magnitudes being consequently irrelevant. In fact, they establish connections with two rank correlation coefficients: Spearman and Kendall. The difference between our paper and the curves shown in [32] (e.g., the rank plots in Figure 1 in that paper) is very significant. There, the magnitudes of the predictions are not important, just the ranks. In our case, the magnitude of the prediction is very important. For instance, just adding a constant value to all predictions leaves all the metrics and analysis in [32] unchanged, while for our curves this clearly increases the expected loss as the binarised regression problem is clearly affected. But it is true that they understand regression in a different way, or as a different task, converting a regression dataset into a ranking decision problem. Nonetheless, on many occasions, that problem is not very realistic. Let us consider the case study proposed in [32] as an example of how the magnitude of the scores is important in order to reach the optimal solution. In this case study, the IBM wallet of customers for information technology products is used for evaluating and comparing models using a ranking-based measure. The authors suggest that the profit would be optimised by only considering the 10% of the total amount of companies (top companies). Let us now propose a different scenario where an important percentage of the spending in this top range (let us say 90%), is gathered together by the first half of these 10% of top companies, but with the same ranking previously obtained. In this case, it will surely be more convenient to the company to reduce the number of companies down to

5%. Therefore, differences between both scenarios (with the same ranking) are henceforth very significant.

Summing up, to the best of our knowledge, the area under the *OCE* curves is not related to any existing metric. It counts mismatches but takes the magnitude into account, so it is a sort of *hybrid* between a ranking metric (such as Spearman or Kendall coefficient) and a residual-based metric (such as *MAE* or *MSE*). The two interpretations that we have given to A_{OCE} reflect this hybrid character.

9 Conclusions

In this paper we have identified what is certainly not a new task but that, to our knowledge, has not been isolated as a standalone data mining task. We have used the term 'binarised regression task', as this problem is mostly characterised by a binarisation of a regression-like data when confronted by decision of whether the value is above or below a given cutoff. This means that once the cutoff is given, the loss metric is just a 0/1 loss, as in binary classification. By focussing on this task, its evaluation and two different kinds of solutions, we have better understood how to characterise it and address it more systematically.

We have seen with a case study that by using the graphical tools that we have proposed it is relatively easy to determine the dominance ranges where each model is best. However, the experiments for many datasets show that if we just focus on the overall metric these regions are not as decisive. Also, we have seen that none of the two kinds of families is best in general, although the retraining approach is more inefficient and behaves more poorly with small training sets. However, we cannot tell a priori what is small or big, as this depends on the problem. This means that each particular case requires a detailed examination. This lack of a clear pattern also supports the idea of having plots and metrics to evaluate each case, as one method can be good for one problem but very bad for other problems.

The main take-away message of this paper is that if a data mining or machine learning practitioner faces a binarised regression task, our recommendation is that s/he should not treat it as a classical regression task, relying exclusively on *MSE* or other classical regression metrics. We suggest to try, if possible, different techniques under the reframing and retraining procedures and plot their *UCE* and *OCE* curves. If we have some information about the regions of cutoffs that are most important, we should focus our selection on those areas. If we do not have such information and we want an overall metric to choose upon, we could calculate the integral using an expected distribution of cutoffs. If we do not have reliable information about the distribution of cutoffs, we suggest to use the observed distribution on the deployment situation. If a data mining or machine learning practitioner wants to do all this, it is really straightforward. A complete library in R for doing this is available at http://www.dsic.upv.es/~flip/BinarisedRegression/ and can tune their

techniques or compare with a repository of 20 binarised regression problems there.

There are clearly many possible continuations and connections of this work. For instance, while we have considered a 1/0 loss, there might be cases where an asymmetric misclassification cost could also be used. In other words, while the application areas we discussed in the introduction are characterised by a cutoff that is set as an all-or-nothing criterion, it is not uncommon that false positives and false negatives may have different costs. This suggests that the cost proportion could be an additional parameter of our context, and plots and metrics could be extended with an extra dimension, with two axes for the context. Nonetheless, the cutoff distribution and the cost distribution will rarely be independent, and we can analyse this dependency or even reduce this to a single parameter.

Also, the problem that we address in this paper is *binarised regression*. Why if we had more than two classes? For instance, the general *discretised regression* problem could be considered (which is not the same as, but might be related to, ordinal regression), with b categories or bins and $b-1$ cutoffs. This general problem is more complex as we would need to consider distributions for two or more cutoffs. As a result, more knowledge (or assumptions) would be required to evaluate the models for a range of cutoffs. This would affect the metrics and also the plots. For b categories, we would require $b-1$ dimensions. One option to simplify the problem would be the use of a cascade of binary decisions, which would be useful for those cases with b small, reusing much of this work. A second option could be considered for those cases with three bins where the middle bin is associated with no action (e.g., sell if below first cutoff, buy if above second cutoff and do nothing in between). This could be related to the problem of abstaining classifiers [12,30]. A third option would be to assume all bins with the same width. In this overly simplified case, we would have one parameter as context (and not a distribution) and we could try to understand the behaviour of models depending on how many bins are going to be considered in application time.

Finally, we have only analysed two possible approaches for this problem: retraining using crisp classification methods and reframing using regression models. However, we think that there are many other possibilities here to be explored, such as the exploration of techniques that try to improve MAE (such as the quantile regression explored here), the clipped MAE or A_{OCE}, the use of soft classification methods or soft regression methods [19] under the reframing paradigm, labelling the training dataset using the median as fixed cutoff and leaving a mapping of classifier scores to cutoffs for deployment time (by using a table, an approximate function or a calibration technique). This could relate this problem with some threshold choice methods in classification, most especially the scoredriven and the ratedriven methods [20].

Acknowledgements

We thank the anonymous reviewers for their comments, which have helped to improve this paper significantly. We thank Peter Flach and Meelis Kull for their insightful comments and very useful suggestions. This work was supported by the Spanish MINECO under grant TIN 2013-45732-C4-1-P and by Generalitat Valenciana PROMETEOII2015/013. This research has been developed within the REFRAME project, granted by the European Coordinated Research on Long-term Challenges in Information and Communication Sciences & Technologies ERA-Net (CHIST-ERA), and funded by the Ministerio de Economía y Competitividad in Spain (PCIN-2013-037) and the Agence Nationale pour la Recherche in France (ANR-12-CHRI-0005-03).

References

1. The keel-dataset repository (2002). URL http://http://www.keel.es/ 7
2. Bache, K., Lichman, M.: UCI machine learning repository (2013). URL http://archive.ics.uci.edu/ml 7
3. Bella, A., Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.J.: Aggregative quantification for regression. Data Mining and Knowledge Discovery **28**(2), 475–518 (2014) 2
4. Bi, J., Bennett, K.P.: Regression error characteristic curves. In: Twentieth International Conference on Machine Learning (ICML-2003). Washington, DC (2003) 33
5. Brooks, A.D.: knnflex: A more flexible KNN (2007). R package version 1.1.1 23
6. Cohen, I., Goldszmidt, M.: Properties and benefits of calibrated classifiers. Knowledge Discovery in Databases: PKDD 2004 pp. 125–136 (2004) 33
7. Drummond, C., Holte, R.: Explicitly representing expected cost: An alternative to ROC representation. In: Knowledge Discovery and Data Mining, pp. 198–207 (2000) 33
8. Drummond, C., Holte, R.: Cost Curves: An Improved Method for Visualizing Classifier Performance. Machine Learning **65**, 95–130 (2006) 33
9. Fawcett, T.: ROC graphs with instance-varying costs. Pattern Recognition Letters **27**(8), 882–891 (2006) 33
10. Fawcett, T., Provost, F.: Adaptive fraud detection. Data Mining and Knowledge Discovery **1**(3), 291–316 (1997) 2
11. Federal Financial Institutions Examination Council: Home Mortgage Disclosure Act (HMDA) (2013). http://www.ffiec.gov/hmda/ 9, 23
12. Ferri, C., Hernández-Orallo, J.: Cautious classifiers. Proceedings of the 1st International Workshop on ROC Analysis in Artificial Intelligence (ROCAI-2004) pp. 27–36 (2004) 36
13. Ferri, C., Hernández-Orallo, J., Modroiu, R.: An experimental comparison of performance measures for classification. Pattern Recognition Let. **30**(1), 27–38 (2009) 33
14. Flach, P.: The geometry of ROC space: Understanding machine learning metrics through ROC isometrics. In: Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), pp. 194–201 (2003) 33
15. Guo, Y., Schuurmans, D.: Discriminative batch mode active learning. In: J. Platt, D. Koller, Y. Singer, S. Roweis (eds.) Advances in Neural Information Processing Systems 20, pp. 593–600. Curran Associates, Inc. (2008) 3
16. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explor. Newsl. **11**(1), 10–18 (2009) 23
17. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York, NY, USA (2001) 23, 29
18. Hernández-Orallo, J.: ROC curves for regression. Pattern Recognition **46**(12), 3395–3411 (2013) 34

19. Hernández-Orallo, J.: Probabilistic reframing for context-sensitive regression. ACM Transactions on Knowledge Discovery from Data, 8(3), to appear (2014) 36
20. Hernández-Orallo, J., Flach, P., Ferri, C.: A unified view of performance metrics: Translating threshold choice into expected classification loss. Journal of Machine Learning Research (JMLR) **13**, 2813–2869 (2012) 13, 33, 36
21. Hornik, K., Buchta, C., Zeileis, A.: Open-source machine learning: R meets Weka. Computational Statistics **24**(2), 225–232 (2009). DOI 10.1007/s00180-008-0119-7 23
22. Hsu, C.N., Knoblock, C.A.: Discovering robust knowledge from databases that change. Data Mining and Knowledge Discovery **2**(1), 69–95 (1998) 33
23. Kocjan, E., Kononenko, I.: Regression as cost-sensitive classification. International multiconference on Information Society pp. 38–41 (2009) 32
24. Koenker, R.: Quantile regression. 38. Cambridge university press (2005) 23
25. Langford, J., Oliveira, R., Zadrozny, B.: Predicting conditional quantiles via reduction to classification. arXiv preprint arXiv:1206.6860 (2012) 32
26. Langford, J., Zadrozny, B.: Estimating class membership probabilities using classifier learners. In: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT05), pp. 198–205 (2005) 32
27. Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M.: The DET curve in assessment of detection task performance. In: Fifth European Conference on Speech Communication and Technology. Citeseer (1997) 33
28. Pan, S.J., Yang, Q.: A survey on transfer learning. Knowledge and Data Engineering, IEEE Transactions on **22**(10), 1345–1359 (2010) 31
29. Piatetsky-Shapiro, G., Masand, B.: Estimating campaign benefits and modeling lift. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, p. 193. ACM (1999) 33
30. Pietraszek, T.: On the use of ROC analysis for the optimization of abstaining classifiers. Machine Learning **68**(2), 137–169 (2007) 36
31. Prati, R.C., Batista, G.E., Monard, M.C.: A survey on graphical methods for classification predictive performance evaluation. IEEE Transactions on Knowledge and Data Engineering **23**, 1601–1618 (2011). DOI http://doi.ieeecomputersociety.org/10.1109/TKDE.2011.59 33
32. Rosset, S., Perlich, C., Zadrozny, B.: Ranking-based evaluation of regression models. Knowledge and Information Systems **12**(3), 331–353 (2007) 34
33. Sammut, C., Webb, G.: Encyclopedia of Machine Learning. Encyclopedia of Machine Learning. Springer (2011) 3
34. Swets, J.A., Dawes, R.M., Monahan, J.: Better decisions through science. Scientific American **283**(4), 82–87 (2000) 33
35. Torgo, L.: Regression error characteristic surfaces. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 697–702. ACM (2005) 33, 34
36. Torgo, L., Gama, J.: Regression by classification. In: Advances in artificial intelligence, pp. 51–60. Springer (1996) 32
37. Yang, Y., Wu, X., Zhu, X.: Mining in anticipation for concept change: Proactive-reactive prediction in data streams. Data Mining and Knowledge Discovery **13**(3), 261–289 (2006) 33
38. Zillow: Zillow API (2013). http://www.zillow.com/howto/api/APIOverview.htm 9