



# Fair evaluation of classifier predictive performance based on binary confusion matrix

Amalia Vanacore<sup>1</sup> · Maria Sole Pellegrino<sup>1</sup> · Armando Ciardiello<sup>1</sup>

Received: 29 April 2022 / Accepted: 4 November 2022 / Published online: 29 November 2022  
© The Author(s) 2022

## Abstract

Evaluating the ability of a classifier to make predictions on unseen data and increasing it by tweaking the learning algorithm are two of the main reasons motivating the evaluation of classifier predictive performance. In this study the behavior of Balanced  $AC_1$  — a novel classifier accuracy measure — is investigated under different class imbalance conditions via a Monte Carlo simulation. The behavior of Balanced  $AC_1$  is compared against that of several well-known performance measures based on binary confusion matrix. Study results reveal the suitability of Balanced  $AC_1$  with both balanced and imbalanced data sets. A real example of the effects of class imbalance on the behavior of the investigated classifier performance measures is provided by comparing the performance of several machine learning algorithms in a churn prediction problem.

**Keywords** Class imbalance · Binary confusion matrix · Predictive performance measures · Customer churn prediction

## 1 Introduction

Classification algorithms are commonly adopted in service industry to manage problems related to various domains, such as cybersecurity systems, smart cities, telecommunication, healthcare, e-commerce, bank and finance, customer care, and many more. In such contexts, classifiers are built to handle binary and/or multi-class problems: *binary* means that the prediction outcomes (class labels) can be twofold

---

✉ Amalia Vanacore  
amalia.vanacore@unina.it

Maria Sole Pellegrino  
mariasole.pellegrino@unina.it

Armando Ciardiello  
armando.ciardiello@unina.it

<sup>1</sup> Department of Industrial Engineering, University of Naples Federico II, p.le Tecchio 80, Naples, Italy

and the two classes are usually indicated as Negative and Positive; *multi-class*, instead, means that the outcome is a value taken from a set of multiple non-overlapping classes.

The evaluation of classifier predictive performance is a relevant issue in order to assess the results of the classification process as well as to obtain a datum that must be optimized by tuning the classifier parameters. Several performance measures can be found in the scientific literature, some based on a threshold, others based on probabilities, while yet others based on ranks (Ferri et al. 2009). However, the most widely employed measures are those based on confusion matrix, a cross table that counts the cases that are properly predicted or classified (cells on the main diagonal) or not correctly predicted or classified (off-diagonal cells). Hereafter, we focus on binary confusion matrix for two reasons: binary classification is the most popular classification task and multi-class problems can be decomposed into a set of binary problems using the One-versus-All or the One-versus-One approach (Mehra and Gupta 2013). Accuracy, sensitivity (or recall), precision, and  $F_1$ -score are some of the commonly used predictive performance measures based on binary confusion matrix. Each measure deals with a specific performance aspect (Sammut and Webb 2011; Sokolova and Lapalme 2009), so that the appropriate performance measure for the problem at hand is generally chosen according to the performance aspect to be investigated. Specifically, Accuracy, by far the most widespread performance measure, focuses on overall classifier performance; whereas, Specificity, Sensitivity, Precision and  $F_1$ -score focus on the performance on one class.

In recent years, the scientific community working on classification algorithms has shown an increasing interest in the challenges that arise when imbalanced data sets are considered and the impact of class imbalance on classification performance measures has become a major issue. Class imbalance occurs in a wide range of scientific areas where unequal class distributions arise naturally in such a way that the rare cases are often difficult to separate from the most frequent ones although they are the most important ones to detect. Some real-world applications suffering from the class imbalance problem happen in telecommunication, web & email classification, ecology, biology, financial services, as well as in medical field for disease diagnosis, in industrial field for fault diagnosis or anomaly detection, and in fields of customer service and marketing for churn prediction (Ahn et al. 2020; Jo and Japkowicz 2004; Galar et al. 2011; Chawla et al. 2002; Dashtipour et al. 2016; Tang et al. 2015). In order to overcome the effect of imbalance on performance measures, several solutions have been suggested, among which the most cited is the use of the Balanced Accuracy.

Another criticism raised against all the above cited classifier predictive performance measures is that they fail to compensate the non-zero probability that some predictions match the actual class only by chance and thus they do not allow to estimate the classification improvement due to the classifier over chance classification. To handle this criticism, an alternative accuracy measure adopted in the last decades within the context of expert systems, machine learning and data mining communities (Ben-David 2008; Duro et al. 2012; Zhou et al. 2019) is the Cohen's  $K$  coefficient

(Cohen 1960), a  $\kappa$ -type coefficient introduced in social and behavioral sciences for measuring the degree of agreement between two raters. Cohen's  $K$  accounts for chance agreement by correcting the proportion of observed agreement with the proportion of agreement expected by chance alone, which is estimated through marginal frequencies. The adoption of Cohen's  $K$  as accuracy measure is still debated since the probability of classifications matching by chance converts a reasonably high proportion of observed agreement into a much smaller coefficient value when the marginal frequencies are unequal (Delgado and Tibau 2019). This means that the chance-agreement term of Cohen's  $K$  actually produces a penalization rather than a direct and verifiable correction for imbalance and thus it is not clear how the coefficient balances the predictive performance over majority and minority class. Moreover, it is worth to highlight that when the imbalance is asymmetrical between actual and predicted classes (i.e. the worst performance ever), Cohen's  $K$  value increases leading to a strongly misleading conclusion. For these reasons, Cohen's  $K$  should be avoided as measure of predictive performance, especially with imbalanced data sets. Another  $\kappa$ -type coefficient recently adopted as predictive performance measure is the  $AC_1$  proposed by Gwet (2002) who formulates the chance agreement term as independent from marginal frequencies (Labatut and Cherifi 2011).

A robust measure of predictive performance suitable for both balanced and imbalanced data sets while compensating the non-zero probability that some classifications match only by chance is obtained by correcting a balanced performance measure with the balanced proportion of classifications matching by chance. Specifically, a balanced measure averages the performance values estimated for each class so as to treat classes equally avoiding the dependency over the majority class, and formulates the chance-agreement term as independent from marginal frequencies. A novel accuracy measure able to treat classes equally while compensating the non-zero probability that some classifications match only by chance is the Balanced  $AC_1$ . This research work aims at investigating, via a Monte Carlo simulation study, the statistical behavior of the proposed Balanced  $AC_1$  for binary classification tasks with different class imbalance conditions and comparing it against other commonly adopted measures, that is Precision, Sensitivity,  $F_1$ -score, Accuracy, Balanced Accuracy, Cohen's  $K$  and  $AC_1$ .

Furthermore, a real example of the effects of class imbalance on the behavior of the investigated classifier performance measures is provided by comparing the performance of several machine learning algorithms in a problem of churn prediction, that is the customer propensity to stop doing businesses with a company (Mishra and Reddy 2017). Handling data imbalance is crucial in customer churn prediction since the number of churned customers generally accounts for a small proportion compared to the number of retained customers (Au et al. 2003; Burez and Van den Poel 2009; Nguyen and Duong 2021).

The remainder of the paper is organized as follows: in Sect. 2 the predictive performance measures under study are introduced; in Sect. 3 the simulation design is described and the main results are fully discussed; Sect. 4 is devoted to churn prediction case study; finally, conclusions are summarized in Sect. 5.

## 2 Predictive performance measures based on binary confusion matrix

The framework where our investigation is set is a machine learning task requiring the solution of a binary classification problem. Specifically, the data set describing the task is composed by  $n$  cases classified on  $k = 2$  non-overlapping classes. The provided classifications are arranged in a  $2 \times 2$  confusion matrix reported in Table 1, whose cells count the number of correctly classified cases belonging (true positives,  $tp$ ) or not belonging (true negatives,  $tn$ ) to the positive class and the number of cases that are either incorrectly assigned to the positive class (false positives,  $fp$ ) or that are not assigned to the positive class (false negatives,  $fn$ ).

These counts are the basis for some of the often used classifier performance measures hereafter introduced. Specifically, Accuracy is defined as the number of both positive and negative successful predictions relative to the total number of classifications. Precision is the proportion of cases predicted as positive that are truly positive related to the total count of cases predicted as positive. Sensitivity is the proportion of cases predicted as positive that are truly positive related to the total count of truly positive cases. Specificity is the proportion of cases predicted as negative that are truly negative related to the total count of truly negative cases. Sensitivity and Specificity can be considered as two kinds of Accuracy, for actual positive cases and actual negative cases, respectively (Tharwat 2020).

The main goal of all classifiers is to improve the Sensitivity, without sacrificing Specificity and Precision. However, the aim of Sensitivity often conflicts with the aims of Specificity and Precision, which may not work well, especially when the data set is imbalanced. Hence, the Balanced Accuracy aggregates both Sensitivity and Specificity measures; whereas  $F_1$ -score has been specifically defined to seek a trade-off between Precision and Sensitivity.

Cohen's  $K$  coefficient (Cohen 1960), belonging to  $\kappa$ -type coefficient family, compensates the effect of classifications matching by chance by correcting the observed agreement,  $p_a$  (given by the proportion of correctly classified cases and thus coinciding exactly with Accuracy), with the proportion of agreement expected by chance,  $p_{a|c}$ . The coefficients belonging to  $\kappa$ -type family are all formulated as follows:

$$\kappa = \frac{p_a - p_{a|c}}{1 - p_{a|c}} \tag{1}$$

**Table 1**  $2 \times 2$  confusion matrix

		Predicted		
		Positive (+)	Negative (-)	Total
Actual	Positive (+)	$tp$	$fn$	AP
	Negative (-)	$fp$	$tn$	AN
	Total	PP	PN	$n$

$\kappa$ -type coefficients differ each other for the adopted notion of chance agreement and thus for the formulation of  $p_{alc}$  term.

Specifically Cohen’s  $K$  formulates  $p_{alc}$  by means of the marginal frequencies:

$$p_{alc}^K = \frac{AP \cdot PP + AN \cdot PN}{n^2} \tag{2}$$

Gwet’s  $AC_1$  coefficient (Gwet 2002) quantifies the probability of agreement between two series of evaluations *exclusively* on the cases not susceptible to agreement by chance. The cases whose classification is not certain are difficult to classify and these are the only cases that could lead to chance agreement. This notion of chance agreement let the  $p_{alc}^{AC_1}$  be formulated as follows:

$$p_{alc}^{AC_1} = 2 \left[ \frac{AP + PP}{2n} \left( 1 - \frac{AP + PP}{2n} \right) \right] \tag{3}$$

The Balanced  $AC_1$ , instead, is formulated as a relative measure that corrects the Balanced Accuracy with the balanced proportion of classifications matching by chance obtained by averaging the values estimated for each class. Specifically, the probability of classifications matching by chance,  $p_{alc}^{Bal AC_1}$ , is given by the probability that the classifications are correctly predicted (i.e. Agreement, event  $A$ ) under the assumption of random classifications (event  $R$ ):

$$p_{alc}^{Bal AC_1} = P(A \cap R) = \frac{P(A|R)^+ \cdot P(R)^+ + P(A|R)^- \cdot P(R)^-}{2} \tag{4}$$

The probability of agreement between actual and randomly predicted classifications for each class is estimated under the assumption that classifications are uniformly distributed between classes:

$$\begin{aligned} P(A|R)^+ &= \frac{1}{2} \frac{tp}{AP} \\ P(A|R)^- &= \frac{1}{2} \frac{tn}{AN} \end{aligned} \tag{5}$$

whereas the probability of providing random classifications is approximated with a normalized measure of randomness defined by the ratio of the observed variance to the variance expected under the assumption of totally random classifications:

$$\begin{aligned} P(R)^+ &= 2 \left[ \frac{tp}{PP} \left( 1 - \frac{tp}{PP} \right) + \frac{fn}{PN} \left( 1 - \frac{fn}{PN} \right) \right] \\ P(R)^- &= 2 \left[ \frac{fp}{PP} \left( 1 - \frac{fp}{PP} \right) + \frac{tn}{PN} \left( 1 - \frac{tn}{PN} \right) \right] \end{aligned} \tag{6}$$

Under these assumptions, the probability of correct classifications matching by chance can be estimated as follows:

$$P_{a|c}^{Bal AC_1} = \frac{\frac{tp}{AP} \cdot \left[ \frac{tp}{PP} \left( 1 - \frac{tp}{PP} \right) + \frac{fn}{PN} \left( 1 - \frac{fn}{PN} \right) \right] + \frac{m}{AN} \cdot \left[ \frac{fp}{PP} \left( 1 - \frac{fp}{PP} \right) + \frac{m}{PN} \left( 1 - \frac{m}{PN} \right) \right]}{2} \tag{7}$$

The formulation of the investigated predictive performance measures based on binary confusion matrix are reported in Table 2.

### 3 Monte Carlo simulation

The statistical behavior of Balanced  $AC_1$  is investigated via a Monte Carlo simulation study under different class imbalance conditions and compared against the behavior of the other performance measures listed in Table 2.

#### 3.1 Study design

The simulated data sets are the  $2 \times 2$  confusion matrices cross-classifying actual and predicted classifications of  $n = 100$  cases under the assumption of a prevalence rate (Pr) of class ‘+’. The study has been designed as a multi-factor experimental design with four factors: prevalence rate Pr, classifier sensitivity  $\alpha$ , classifier specificity  $\beta$  and classifier propensity of randomly classifying cases into classes  $\theta$ . A total of 576 different scenarios have been investigated, corresponding to balanced and imbalanced data sets, obtained by assigning 3 levels (i.e. 0.5, 0.7, 0.9) to factor Pr; moreover, the factors  $\alpha$  and  $\beta$  have 8 levels: 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95; and the factor  $\theta$  has 3 levels: 0.05, 0.20, 0.50. The behavior of each performance measure is

**Table 2** Formulation of the investigated predictive performance measures based on binary confusion matrix

Name	Formulation
Accuracy	$\frac{tp+tn}{n}$
Precision	$\frac{tp}{tp+fp}$
Sensitivity	$\frac{tp}{tp+fn}$
Specificity	$\frac{tn}{fp+tn}$
Balanced Accuracy	$\left( \frac{tp}{AP} + \frac{tn}{AN} \right) \cdot \frac{1}{2}$
F <sub>1</sub> -score	$\frac{2 \cdot tp}{2 \cdot tp + fn + fp}$
Cohen’s K	$\left[ \frac{tp+tn}{n} - \frac{AP \cdot PP + AN \cdot PN}{n^2} \right] / \left[ 1 - \frac{AP \cdot PP + AN \cdot PN}{n^2} \right]$
AC <sub>1</sub>	$\left[ \frac{tp+m}{n} - \frac{AP+PP}{n} \left( 1 - \frac{AP+PP}{2n} \right) \right] / \left[ 1 - \frac{AP+PP}{n} \left( 1 - \frac{AP+PP}{2n} \right) \right]$
Balanced AC <sub>1</sub>	$\left[ \left( \frac{tp}{AP} + \frac{tn}{AN} \right) \cdot \frac{1}{2} - P_{a c}^{Bal AC_1} \right] / \left[ 1 - P_{a c}^{Bal AC_1} \right]$

assessed by looking at the the bias of the measure related to the true predictive performance value.

The simulation procedure works as follows:

1. set the factors  $Pr$ ,  $\alpha$ ,  $\beta$  and  $\theta$  at the beginning of each experiment;
2. assign the  $n$  cases to the actual class in such a way that the probability of belonging to class ‘+’ is  $Pr$ ;
3. choose a proportion of  $\theta$  cases and randomly assign them into ‘+’ or ‘-’ class with the same probability  $1/2$ , assuming that non random classifications lead to a correct classification;
4. assign the  $(1 - \theta)\%$  not-randomly-classified cases to the predicted class in such a way that the probability of belonging to class ‘+’ is  $\alpha$  for cases with actual class ‘+’, and  $1 - \beta$  for cases with actual class ‘-’;
5. match actual and predicted classifications for each case and fill the  $2 \times 2$  confusion matrix;
6. assess the predictive performance via Precision, Sensitivity,  $F_1$ -score, Accuracy, Balanced Accuracy, Cohen’s  $K$ ,  $AC_1$ , and Balanced  $AC_1$ ;
7. repeat  $R$  times steps 2 through 6;
8. for each measure under comparison, the performance — expressed in terms of relative bias — is estimated as follows:

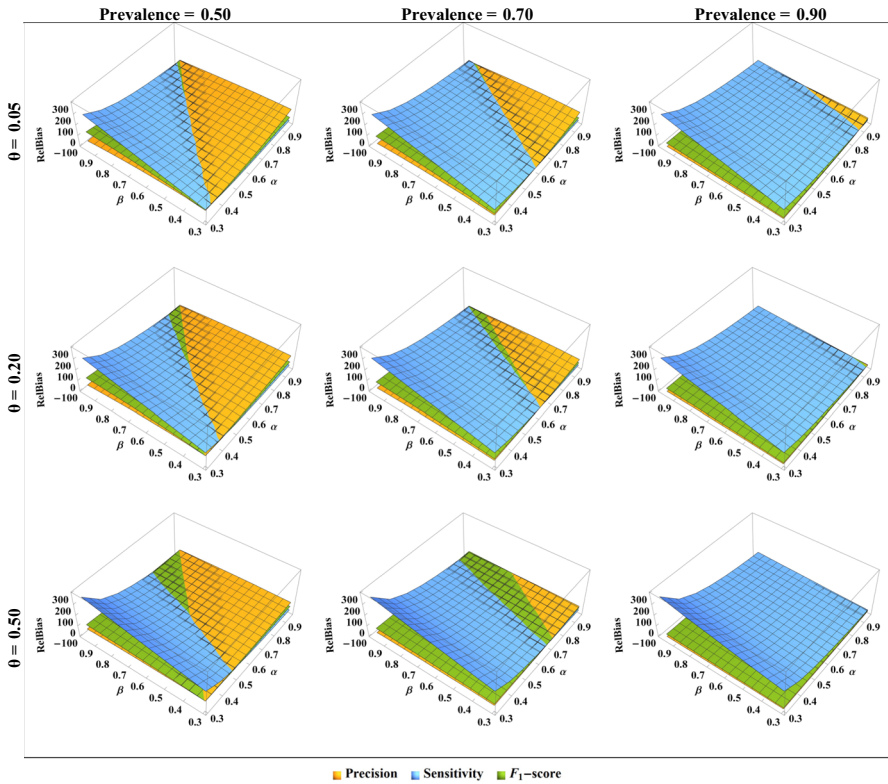
$$\text{RelBias} = \frac{\frac{1}{R} \sum_{r=1}^R \hat{p}_r - p^*}{p^*} \quad (8)$$

where  $\hat{p}_r$  is the classifier performance estimated for the  $r$ th data set, and  $p^*$  is the ‘true’ value of classifier performance, which can be defined for each combination of  $\alpha$ ,  $\beta$  and  $\theta$ .

### 3.2 Simulation results

The values of relative bias obtained for each performance measure and every combination of  $\alpha$  and  $\beta$  are represented in Figs. 1 through 3 for each propensity for random classification  $\theta$  (on row) and prevalence rate (on column). Simulation results highlight the different behavior of the predictive performance measures under comparison: for most scenarios, the performance measures with the worst and best behavior in terms of relative bias are Sensitivity and Balanced  $AC_1$ , respectively.

Specifically, simulation results for performance measures focusing on one class (see Fig. 1) highlight that when classes are balanced (i.e.  $Pr = 0.50$ ) and a very low percentage of cases are randomly classified (i.e.  $\theta = 0.05$ ), Precision tends to overestimate the predictive performance when  $\alpha > 0.6$ ; Sensitivity and  $F_1$ -score overestimate predictive performance when  $\alpha < \beta$ . Vice-versa, the relative bias is underestimated in the other scenarios. The relative bias of Precision, Sensitivity and  $F_1$ -score gets greater as the difference between  $\alpha$  and  $\beta$  increases.



**Fig. 1** Relative bias of Precision, Sensitivity and  $F_1$ -score

For increasing prevalence value, Precision and  $F_1$ -score underestimate the predictive performance, as revealed by their relative bias equal respectively to  $-90\%$  and  $-70\%$ ; Sensitivity, as expected, is not affected by changes in prevalence rate. Vice-versa, the propensity for random classifications impacts only on Sensitivity, whose relative bias strongly increases with  $\theta$ , especially when  $\alpha \leq 0.3$ .

Simulation results for overall performance measures (see Fig. 2) reveal that when  $\theta = 0.05$ , an increase in class imbalance produces no change in Balanced Accuracy; instead Accuracy overestimates (resp. underestimates) up to  $40\%$  (resp.  $-40\%$ ) when  $\alpha > \beta$  (resp.  $\alpha < \beta$ ). When the classifier propensity for random classification  $\theta$  increases, the behavior of Accuracy and Balanced Accuracy gets worse. It is worth to note that the relative bias tends to infinity if  $\alpha + \beta$  approaches  $\theta$ , which is evident in the scenario with both  $\alpha$  and  $\beta$  equal to  $0.3$  and  $\theta = 0.50$ , where the relative bias reaches  $470\%$ .

Simulation results for chance-corrected performance measures (see Fig. 3) reveal that when classes are balanced (i.e.  $Pr = 0.50$ ) and a very low percentage of cases are randomly classified (i.e.  $\theta = 0.05$ ), Cohen's  $K$ ,  $AC_1$  and Balanced  $AC_1$  underestimate predictive performance. The relative bias of Cohen's  $K$  and  $AC_1$  is equal to



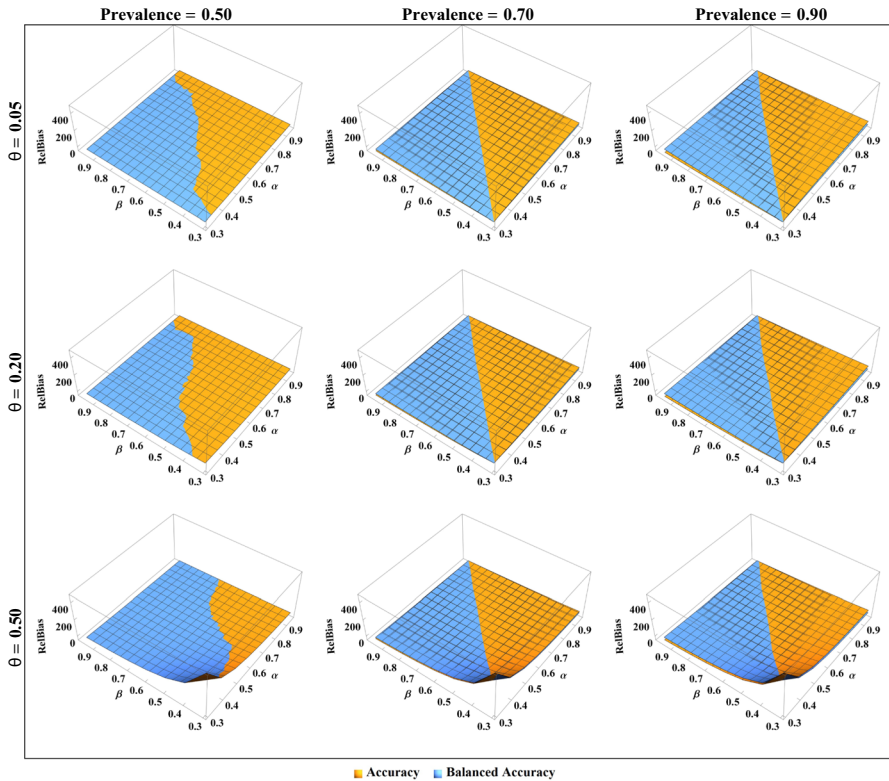


Fig. 2 Relative bias of Accuracy and Balanced Accuracy

−45% when  $\alpha$  and  $\beta$  are no more than 0.5 and it gets smaller as  $\alpha$  and  $\beta$  increase; whereas the relative bias of Balanced  $AC_1$  is always no more than − 10%.

When prevalence value and propensity for random classification  $\theta$  increase, the relative bias of Cohen’s  $K$  and  $AC_1$  worsens for  $\alpha$  and  $\beta$  lower than 0.7, indeed in the presence of imbalanced data sets the Cohen’s  $K$  value decreases since the probability of classifications matching by chance is estimated through marginal frequencies and it is not clear how the predictive performance is balanced over majority and minority classes. Moreover, when  $\alpha \geq 0.7$ , the  $AC_1$  overestimates classifier predictive performance. Balanced  $AC_1$  values, instead, get closer to the ‘true’ value of predictive performance for increasing prevalence and the relative bias becomes no more than − 4%.

While the behavior of Precision,  $F_1$ -score, Accuracy, Cohen’s  $K$  and  $AC_1$  changes with class prevalence and  $\theta$  being often far from the ‘true’ value of predictive performance, Balanced Accuracy and Balanced  $AC_1$  are always close to such ‘true’ value whatever class prevalence and  $\theta$ , with the only exception of those scenarios with  $\alpha + \beta \approx \theta$ . Thus, these latter performance measures are recommended for both balanced and imbalanced data sets.

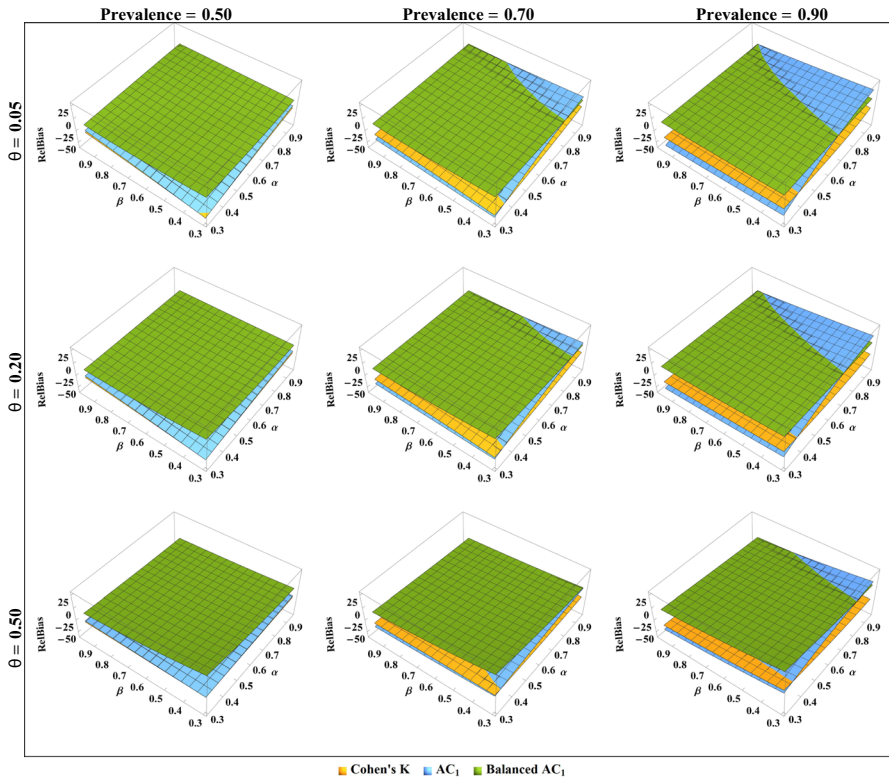


Fig. 3 Relative bias of Cohen’s  $K$ ,  $AC_1$  and Balanced  $AC_1$

### 4 Customer churn prediction

Many subscription-based service industries are constantly striving to recognize customers that are looking to switch providers (i.e. customer churn). Reducing churn is extremely important in competitive markets since acquiring new customers is very difficult (Verbeke et al. 2011). For this reason, in many service fields such as banks, telecommunication and internet services, games, and insurance, just to name a few (De Bock and Van den Poel 2011; Dechant et al. 2019; Lee et al. 2018; Ngai et al. 2009; Xie et al. 2015; Zhang e al. 2017), churn analysis is one of the most important personalized customer management techniques.

The ability to predict that a particular customer is at a high risk of churning represents a huge additional potential revenue source for companies; indeed, the increase of retention rate of loyal customers is more efficient than acquiring new customers, which can cost up to six times more than what it costs to retain the current customers by taking active steps to discourage churn behavior (Mishra and Reddy 2017).

Because of customer churn impact on business performance, churn-prone industries typically maintain customer relationship management (CRM)

databases; however, knowledge discovery in such rich CRM databases, which typically contains thousands or millions of customers information, is a challenging and difficult task (Amin et al. 2016). As a consequence, several competitive industries have implemented a wide range of statistical and intelligent machine learning (ML) techniques to develop predictive models that deal with customer churn (Burez and Van den Poel 2009).

However, churn prediction algorithms often fail to handle the imbalance between churn and non-churn groups since they put emphasis on the majority of non-churn customers, leaving the prediction of churn customers vulnerable (Nguyen and Duong 2021).

In this case study, several churn prediction models have been evaluated using the investigated classifier predictive performance measures in order to assess the effects of class imbalance on the behavior of each measure.

#### 4.1 Data set

The analyzed Telco Customer Churn data set<sup>1</sup> deals with customer churn of a telecommunication company, that is the percentage of customers who stopped using company's service within the last month. The data set consists of 21 attributes and 7043 rows; each row represents a customer, while each column contains an attribute pertaining to customer that helps to deduce a comprehensible relation between customer behavior and churn. The attributes can be distinguished into three groups: services that customer has signed up for (i.e. phone, multiple lines, internet, online security, online backup, device protection, technical support, streaming TV and streaming movies); customer account information (i.e. how long she/he has been customer, contract, payment method, paperless billing, monthly charges, and total charges); and customer demographic information (i.e. gender, age range, and if she/he has partners and dependents). All these attributes can be further classified into two fundamental categories: numeric attributes and object type attributes. All information about attributes are reported in Table 3.

The last attribute is the binary output variable: the value "Yes" is for customers who churned, while "No" for the others. The frequency of the two classes is different, that is 73.5% for class "No" and 26.5% for class "Yes". The presence of class imbalance makes the problem suitable to display and compare the different behavior of the investigated predictive performance measures.

The data set has been analyzed using the Automated Machine Learning pipeline PyCaret (Ali 2020) in the Google Colab notebook environment (Bisong 2019). Different algorithms, among those most commonly adopted in churn prediction applications, have been applied: Regression models such as Logistic Regression (LR), Linear Discriminant Analysis (LDA) and Ridge Regression (R; Bhatnagar and Srivastava 2019); Boosted Tree techniques (De et al. 2021) such as Gradient Boosting (GB), Extreme Gradient Boosting (XGB), CatBoost (CAT), and Extra trees classifier

<sup>1</sup> <https://www.kaggle.com/datasets/blastchar/telco-customer-churn>

**Table 3** List of attributes (i.e. input variables) and main characteristics

Attribute	Description	Category
1	ID	Numeric
2	Gender	Yes, no
3	Senior Citizen	Yes, no
4	Partner exists	Yes, no
5	Dependents exists	Yes, no
6	Tenure	Numeric
7	Phone service	Yes, no
8	Multiple phone lines	Yes, no, other
9	Internet service	DSL, fiber optic, no
10	Online security	Yes, no, other
11	Online backup	Yes, no, other
12	Device protection	Yes, no, other
13	Technical support	Yes, no, other
14	Streaming TV	Yes, no, other
15	Streaming movies	Yes, no, other
16	Contract term type	Month-to-month, one year, two year
17	Paperless billing	Yes, no
18	Payment method	Electronic check, Mailed check, other
19	Monthly charges	Numeric
20	Total charges	Numeric
21	Churn	Yes, no

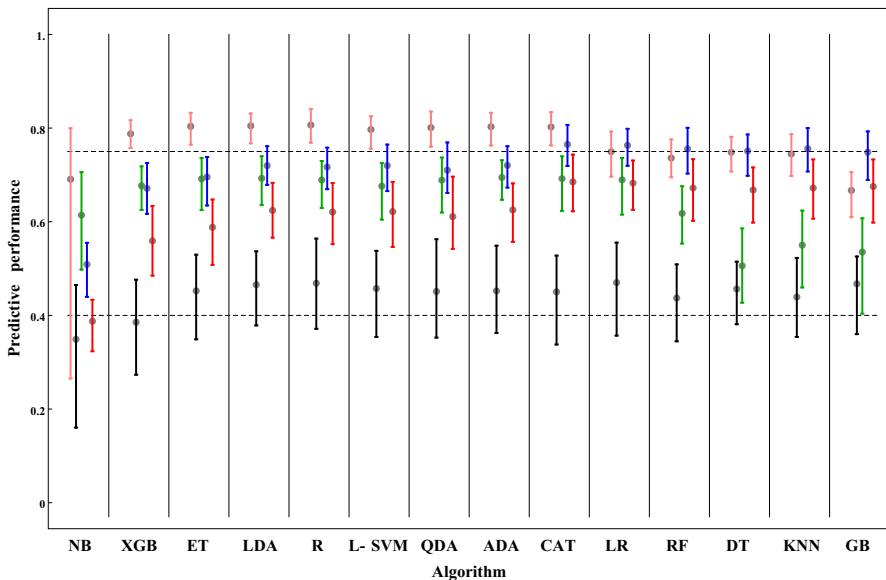
(ET); Linear Support Vector Machine (Liner-SVM; Coussement and Van den Poel 2008); Naïve Bayes (NB; Fei et al. 2017); K-Nearest-Neighbor (KNN; Hassonah et al. 2019); Quadratic Discriminant Analysis (QDA); Ada Boost (ADA); Decision Trees (DT; Qureshi et al. 2013) and Random Forest (RF).

The adopted strategy for estimating predictive performance is based on repeated stratified nested cross-validation (CV) that involves treating model hyper-parameter optimization as part of the model itself and evaluating it within the broader  $V$ -fold CV procedure for models evaluation and comparison. Namely, the CV procedure for model hyper-parameter optimization (i.e. inner loop responsible for model selection) is nested inside the CV procedure for model evaluation (i.e. outer loop responsible for generalization performance estimation). The algorithm adopted for implementing repeated stratified nested CV is detailed in the Appendix. A number of 10 repetitions are performed for both loops (i.e.  $r_I = r_O = 10$ ) with 10 folds for both inner and outer loop (i.e.  $V_I = V_O = 10$ ).

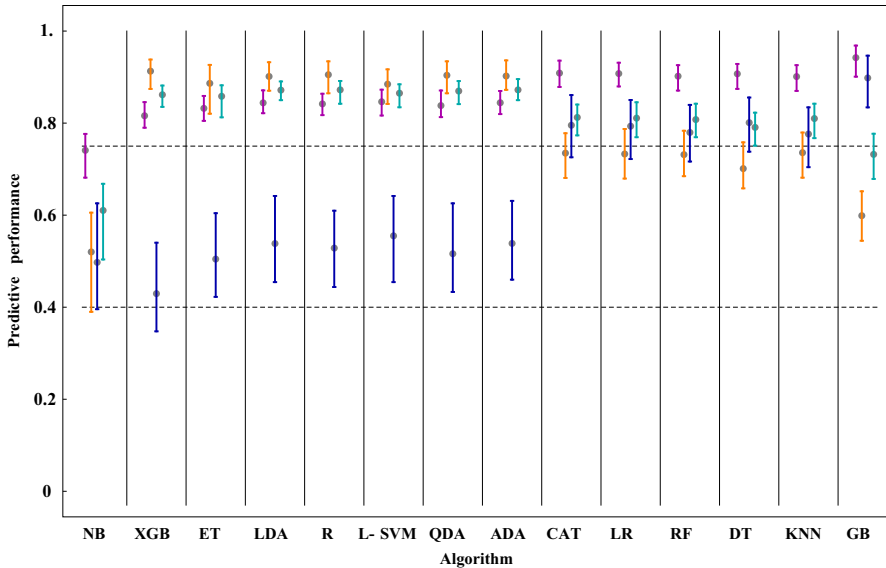
Classifier predictive performance has been assessed by means of the measures under comparison, that is Accuracy, Cohen's  $K$ ,  $AC_1$ , Balanced Accuracy, Balanced  $AC_1$ , Precision, Sensitivity, Specificity, and  $F_1$ -score.

## 4.2 Study results

The large-sample predictive performance estimates have been obtained by averaging the  $r_O$  nested CV performance values and their variation has been assessed by the range over the  $r_O$  values. The results obtained by assessing predictive



**Fig. 4** Large-sample estimates and ranges of Accuracy (in orange), Cohen's  $K$  (in black),  $AC_1$  (in green), Balanced Accuracy (in blue) and Balanced  $AC_1$  (in red) (colour figure online)



**Fig. 5** Large-sample estimates and ranges of Precision (in purple), Sensitivity (in orange), Specificity (in blue) and  $F_1$ -score (in green) (colour figure online)

performance via Accuracy, Cohen's  $K$ ,  $AC_1$  and Balanced  $AC_1$  are represented in Fig. 4 and reported in Table 4; for comparative purpose, the results obtained via Precision, Sensitivity, Specificity and  $F_1$ -score are represented in Fig. 5.

Study results reveal the different behavior of algorithms and performance measures. Specifically, Cohen's  $K$  underestimates the predictive performance because of the effect of the penalization produced by the adopted chance-agreement term which makes the coefficient value strongly decrease in the presence of class imbalance; moreover, its variation is often larger than that of other measures. These results confirm that Cohen's  $K$  cannot be considered a trustable measure of predictive performance with imbalanced data sets.

The large-sample estimates generally take values between 0.4 and 0.75, with the exception of NB and XGB algorithms whose predictive performance is assessed lower than 0.4 via Cohen's  $K$ ; moreover for XGB, ET, LDA, R, L-SVM, QDA, ADA and CAT algorithms the large-sample predictive performance is greater than 0.75 when assessed via Accuracy, Precision, Sensitivity and  $F_1$ -score; for CAT, LR, RF, DT and KNN the large-sample predictive performance is greater than 0.75 when assessed via Balanced Accuracy, Precision, Specificity and  $F_1$ -score. In the light of simulation findings, case study results allow to group the algorithms as follows:

1. algorithms performing better on majority class, which obtain higher values of Accuracy, Cohen's  $K$ ,  $AC_1$ , Precision, Sensitivity and  $F_1$ -score, *viz.* NB, XGB, ET, LDA, R, L-SVM, QDA, ADA;
2. algorithms performing similarly on both classes, showing the same values for all performance measures, *viz.* CAT, LR;

**Table 4** Large-sample estimates and ranges (in square brackets) of predictive performance measures

Algorithm	Precision	Sensitivity	Specificity	F <sub>1</sub> -score	Accuracy	Cohen's K	Gwet AC <sub>1</sub>	Balanced Acc	Balanced AC <sub>1</sub>
NB	0.7409 [0.681, 0.776]	0.5201 [0.390, 0.605]	0.4974 [0.396, 0.626]	0.6103 [0.504, 0.668]	0.6906 [0.265, 0.800]	0.3490 [0.160, 0.465]	0.6140 [0.498, 0.706]	0.5088 [0.439, 0.555]	0.3875 [0.323, 0.433]
XGB	0.8159 [0.790, 0.846]	<b>0.9129</b> [0.874, 0.938]	0.4295 [0.347, 0.540]	0.8616 [0.835, 0.882]	0.7877 [0.757, 0.817]	0.3854 [0.273, 0.476]	0.6769 [0.625, 0.718]	0.6712 [0.617, 0.725]	0.5592 [0.485, 0.634]
ET	0.8321 [0.805, 0.859]	0.8864 [0.820, 0.926]	0.5045 [0.422, 0.604]	0.8583 [0.813, 0.882]	0.8036 [0.764, 0.833]	0.4525 [0.349, 0.529]	0.6916 [0.625, 0.736]	0.6955 [0.635, 0.738]	0.5882 [0.508, 0.647]
LDA	0.8381 [0.813, 0.871]	0.9039 [0.865, 0.934]	0.5161 [0.433, 0.626]	0.8697 [0.841, 0.891]	0.8048 [0.767, 0.831]	0.4655 [0.379, 0.537]	0.6928 [0.636, 0.740]	0.7198 [0.679, 0.761]	0.6242 [0.566, 0.683]
R	0.8417 [0.817, 0.864]	0.9052 [0.865, 0.934]	0.5285 [0.444, 0.610]	<b>0.8722</b> [0.842, 0.891]	<b>0.8065</b> [0.769, 0.841]	0.4687 [0.371, 0.564]	0.6892 [0.629, 0.730]	0.7168 [0.669, 0.758]	0.6206 [0.552, 0.683]
L-SVM	0.8440 [0.821, 0.871]	0.9014 [0.870, 0.932]	0.5383 [0.454, 0.642]	0.8717 [0.850, 0.890]	0.7970 [0.756, 0.825]	0.4573 [0.354, 0.538]	0.6760 [0.604, 0.725]	0.7198 [0.665, 0.765]	0.6216 [0.546, 0.685]
QDA	0.8442 [0.820, 0.870]	0.9023 [0.872, 0.936]	0.5386 [0.460, 0.631]	<b>0.8722</b> [0.850, 0.896]	0.8011 [0.760, 0.835]	0.4511 [0.353, 0.563]	0.6888 [0.620, 0.737]	0.7100 [0.661, 0.769]	0.6109 [0.542, 0.696]
ADA	0.8464 [0.816, 0.873]	0.8846 [0.842, 0.917]	0.5550 [0.454, 0.642]	0.8650 [0.834, 0.884]	0.8029 [0.763, 0.833]	0.4523 [0.362, 0.549]	0.6945 [0.647, 0.731]	0.7205 [0.673, 0.761]	0.6252 [0.557, 0.682]
CAT	0.9011 [0.870, 0.926]	0.7357 [0.681, 0.779]	0.7762 [0.704, 0.834]	0.8099 [0.768, 0.842]	0.8023 [0.763, 0.834]	0.4504 [0.338, 0.528]	0.6921 [0.623, 0.740]	<b>0.7651</b> [0.719, 0.807]	<b>0.6851</b> [0.622, 0.743]
LR	0.9020 [0.871, 0.926]	0.7315 [0.685, 0.783]	0.7797 [0.717, 0.839]	0.8077 [0.769, 0.842]	0.7496 [0.696, 0.793]	<b>0.4701</b> [0.357, 0.555]	<b>0.6896</b> [0.615, 0.736]	0.7631 [0.719, 0.798]	0.6825 [0.625, 0.731]
RF	0.9071 [0.874, 0.928]	0.7011 [0.658, 0.758]	0.8010 [0.738, 0.856]	0.7907 [0.751, 0.823]	0.7360 [0.695, 0.776]	0.4371 [0.345, 0.509]	0.6178 [0.553, 0.676]	0.7556 [0.703, 0.800]	0.6719 [0.602, 0.733]
DT	0.9077 [0.880, 0.931]	0.7329 [0.679, 0.787]	0.7934 [0.722, 0.850]	0.8108 [0.769, 0.845]	0.7484 [0.707, 0.781]	0.4564 [0.382, 0.515]	0.5059 [0.427, 0.586]	0.7511 [0.698, 0.786]	0.6678 [0.598, 0.716]

Table 4 (continued)

Algorithm	Precision	Sensitivity	Specificity	F <sub>1</sub> -score	Accuracy	Cohen's K	Gwet AC <sub>1</sub>	Balanced Acc	Balanced AC <sub>1</sub>
KNN	0.9087 [0.878, 0.936]	0.7348 [0.681, 0.778]	0.7953 [0.726, 0.861]	0.8124 [0.773, 0.840]	0.7448 [0.698, 0.787]	0.4393 [0.354, 0.523]	0.5499 [0.460, 0.624]	0.7560 [0.707, 0.800]	0.6722 [0.607, 0.733]
GB	<b>0.9422</b> [0.901, 0.968]	0.5988 [0.544, 0.652]	<b>0.8981</b> [0.834, 0.946]	0.7321 [0.679, 0.777]	0.6670 [0.610, 0.706]	0.4674 [0.360, 0.526]	0.5352 [0.404, 0.607]	0.7485 [0.689, 0.793]	0.6752 [0.598, 0.733]

The predictive performance value of the best performing algorithm is reported in bold for each measure



3. algorithms performing better on minority class, which obtain higher values of Balanced Accuracy, Balanced  $AC_1$  and Specificity, *viz.* RF, DT, KNN, GB.

All the adopted performance measures agree in identifying NB as the algorithm with the worst predictive performance. This result is not surprising since NB generally shows lower performance than other classifiers (Akkaya and Çolakoğlu 2019), mainly because of the assumption that all predictors (i.e. attributes) are mutually independent, which rarely happens in real life. On the other hand, the performance measures do not agree on the selection of the best algorithm: Accuracy selects R, Cohen's  $K$  and  $AC_1$  select LR whereas Balanced Accuracy and Balanced  $AC_1$  select CAT as the best performing algorithm.

It is worth pointing out that in a churn model, the reward of true positives is often very different than both the cost of false positives and the missed gain of false negatives. Thus, assuming that  $C$  is the cost to retain a customer identified as churn, that  $\omega C$  is the customer lifetime value gained if the churn is stopped and missed if the churn is not predicted, a simple Profit measure can be derived as follows:

$$\text{Profit} = \omega K \cdot tp - K \cdot (tp + fp) - \omega K \cdot fn \quad (9)$$

The Profit has been assessed for all the algorithms under comparison and the results, reported in Table 5, reveal that the algorithm with the highest Profit value is GB, which is the algorithm with the highest difference between Balanced Accuracy and Balanced  $AC_1$  and the other performance measures, meaning the best performance on minority class of "No" churn and the lowest proportion of  $fn$  cases.

More interestingly, looking at Profit as a benchmark, Balanced  $AC_1$  is the predictive performance measure that ranks the algorithms more similarly to it; whereas

**Table 5** Large-sample estimates and range of profit (with  $C = 1000$  and  $\omega = 5$ )

Algorithm	Profit	Algorithm	Profit
NB	-346,060 [-505,000, -157,000]	ADA	-79,010 [-220,000, 71,000]
XGB	-257,120 [-409,000, -72,000]	CAT	266, 140 [136, 000, 376, 000]
ET	-144,560 [-285,000, 16,000]	LR	261, 840 [145, 000, 358, 000]
LDA	-80,020 [-215,000, 81,000]	RF	238, 160 [110, 000, 342, 000]
R	-94,630 [-236,000, 37,000]	DT	258, 260 [130, 000, 346, 000]
L-SVM	-60,520 [-234,000, 87,000]	KNN	234, 500 [99, 000, 338, 000]
QDA	-115,980 [-263,000, 73,000]	GB	<b>368,690</b> [233, 000, 459, 000]

The maximum profit value is reported in bold

the algorithms' rankings provided by Sensitivity,  $F_1$ -score, Accuracy and  $AC_1$  are inversely correlated to that provided by Profit.

## 5 Conclusions

This research study aims to investigate — via a Monte Carlo simulation — the statistical behavior of a new classifier performance measure, that is Balanced  $AC_1$  coefficient, under different scenarios of class imbalance conditions with binary classification tasks. The behavior of Balanced  $AC_1$  is compared against that of other performance measures, that is Precision, Sensitivity,  $F_1$ -score, Accuracy, Balanced Accuracy, Cohen's  $K$  and  $AC_1$ .

Simulation results reveal that Balanced  $AC_1$  has a smaller relative bias (i.e. generally no more than 10%) compared against the other performance measures. Among one-class performance measures, Sensitivity is the one with the worst predictive performance and it generally tends to overestimate classifier performance; in the group of chance-corrected measures, Cohen's  $K$  is that with the highest relative bias. As expected, the dependency of Accuracy on the performance over the majority class makes it overestimate (resp. underestimate) the predictive performance when the classifier predicts best the majority (resp. minority) class. More interestingly, simulation results reveal that  $F_1$ -score, although commonly considered a performance measure suitable for imbalanced data sets, has generally a greater relative bias than Balanced Accuracy and Balanced  $AC_1$ , tending to underestimate the predictive performance in the presence of high class imbalance.

The difference among the behavior of classifier predictive performance measures increases with class imbalance, which is a rule in many real-world classification problems, and with the propensity of random classification. Although Balanced Accuracy and Balanced  $AC_1$  seem to have similar behavior, it is recommended the adoption of Balanced  $AC_1$ , due to its ability to both deal with class imbalance and account for classifications matching by chance.

Moreover, the predictive performance measures under study have been applied to a real data set dealing with the problem of predicting customer churn in a telecommunication industry. The empirical results confirm the best suitability of Balanced  $AC_1$ , being the performance measure best correlated to a cost-sensitive criterion for the selection of the best performing algorithm.

## Appendix: Algorithm of repeated stratified nested CV

Let  $D$  be a data set of  $n$  realizations  $(Y, X_1, X_2, \dots, X_p)$ ,  $f_k$  be a classifier with a hyper-parameter vector  $\alpha_k$ ,  $r_I$  and  $r_O$  be the repetitions performed respectively for inner and outer loop,  $V_I$  and  $V_O$  be the number of folds in which inner and outer loop are respectively stratified.

Let us consider a grid of  $K$  points  $\alpha_1, \dots, \alpha_K$ ; the optimal one can be found via repeated stratified nested CV, whose protocol works as follows:

1. divide the data set  $D$  into  $V_O$  stratified folds;
2. for each  $i$ th stratified fold, with  $i$  from 1 to  $V_O$ :
  - (a) define the learning set  $L_{O_i}$  as the data set  $D$  without the  $i$ th fold;
  - (b) define the test set  $T_{O_i}$  as the  $i$ th fold of the data set  $D$ ;
  - (c) divide the data set  $L_{O_i}$  into  $V_I$  stratified folds;
  - (d) for each  $i$ th stratified fold of  $L_{O_i}$ , with  $i = 1, \dots, V_I$ :
    - i. define the learning set of the inner loop  $L_{I_i}$  as the data set  $L_{O_i}$  without the  $i$ th fold;
    - ii. define the test set of the inner loop  $T_{I_i}$  as the  $i$ th fold of the data set  $L_{O_i}$ ;
    - iii. for  $k$  from 1 to  $K$ :
      - build statistical model  $f_{ik} = f(L_{I_i}; \alpha_k)$ ;
      - apply  $f_{ik}$  on  $T_{I_i}$  and store the predictions;
  - (e) for each  $\alpha$  calculate the classifier performance on all elements in  $L_{O_i}$ ;
  - (f) repeat  $r_I$  times the steps from (c) to (e);
  - (g) for each  $\alpha$  calculate the mean over the  $r_I$  values of classifier performance;
  - (h) let  $\alpha^*$  be the hyper-parameter vector for which the average performance is maximal and select  $f^* = f(L_{I_i}; \alpha^*)$  as the optimal cross-validators model;
  - (i) apply  $f^*$  on  $L_{O_i}$ ;
  - (j) calculate the predictive performance of  $f^*$  on  $T_{O_i}$ ;
3. calculate the average predictive performance over all test sets  $T_{O_i}$ , hereafter referred to as the nested CV predictive performance;
4. repeat  $r_O$  times the process from step 1 to step 3.

The mean over the  $r_O$  nested CV performance values is the estimate of the large-sample predictive performance of algorithm  $f^*$ , whereas the interval between the minimum and maximum over the  $r_O$  nested CV performance values is the estimated interval of the large-sample predictive performance of algorithm  $f^*$ .

**Acknowledgements** The authors would like to thank the anonymous reviewers for taking the time and effort necessary to review the manuscript. Their precious comments and suggestions allowed to significantly improve the quality of the manuscript.

**Funding** Open access funding provided by Università degli Studi di Napoli Federico II within the CRUI-CARE Agreement.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission

directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Ahn J, Hwang J, Kim D, Choi H, Kang S (2020) A survey on churn analysis in various business domains. *IEEE Access* 8:220816–220839
- Akkaya B, Çolakoğlu N (2019) Comparison of multi-class classification algorithms on early diagnosis of heart diseases
- Ali M (2020) PyCaret: an open source, low-code machine learning library in Python. PyCaret version 1.0.0
- Amin A, Anwar S, Adnan A, Nawaz M, Howard N, Qadir J, Hawalah A, Hussain A (2016) Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study. *IEEE Access* 4:7940–7957
- Au W-H, Chan KC, Yao X (2003) A novel evolutionary data mining algorithm with applications to churn prediction. *IEEE Trans Evol Comput* 7(6):532–545
- Ben-David A (2008) Comparison of classification accuracy using Cohen's weighted kappa. *Expert Syst Appl* 34(2):825–832
- Bhatnagar A, Srivastava S (2019) A robust model for churn prediction using supervised machine learning. In: 2019 IEEE 9th international conference on advanced computing (IACC), pp 45–49. IEEE
- Bisong E (2019) Building machine learning and deep learning models on Google cloud platform: a comprehensive guide for beginners. Apress
- Burez J, Van den Poel D (2009) Handling class imbalance in customer churn prediction. *Expert Syst Appl* 36(3):4626–4636
- Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357
- Cohen J (1960) A coefficient of agreement for nominal scales. *Educ Psychol Meas* 20(1):37–46
- Coussement K, Van den Poel D (2008) Churn prediction in subscription services: an application of support vector machines while comparing two parameter-selection techniques. *Expert Syst Appl* 34(1):313–327
- Dashtipour K, Poria S, Hussain A, Cambria E, Hawalah AY, Gelbukh A, Zhou Q (2016) Multilingual sentiment analysis: state of the art and independent comparison of techniques. *Cogn Comput* 8(4):757–771
- De S, Prabu P, Paulose J (2021) Effective ML techniques to predict customer churn. In: 2021 Third international conference on inventive research in computing applications (ICIRCA), pp 895–902. IEEE
- De Bock KW, Van den Poel D (2011) An empirical evaluation of rotation-based ensemble classifiers for customer churn prediction. *Expert Syst Appl* 38(10):12293–12301
- Dechant A, Spann M, Becker JU (2019) Positive customer churn: an application to online dating. *J Serv Res* 22(1):90–100
- Delgado R, Tibau X-A (2019) Why Cohen's Kappa should be avoided as performance measure in classification. *PLoS One* 14(9):e0222916
- Duro DC, Franklin SE, Dubé MG (2012) A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using SPOT-5 HRG imagery. *Remote Sens Environ* 118:259–272
- Fei TY, Shuan LH, Yan LJ, Xiaoning G, King SW (2017) Prediction on customer churn in the telecommunications sector using discretization and Naïve Bayes classifier. *Int J Adv Soft Comput Appl* 9(3):23–35
- Ferri C, Hernández-Orallo J, Modroi R (2009) An experimental comparison of performance measures for classification. *Pattern Recognit Lett* 30(1):27–38
- Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F (2011) A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Trans Syst Man Cybern Part C Appl Rev* 42(4):463–484
- Gwet K (2002) Kappa statistic is not satisfactory for assessing the extent of agreement between raters. *Stat Methods Inter-rater Reliab Assess* 1(6):1–6

- Hassonah MA, Rodan A, Al-Tamimi A-K, Alsakran J (2019). Churn prediction: a comparative study using KNN and decision trees. In: 2019 Sixth HCT information technology trends (ITT), pp 182–186. IEEE
- Jo T, Japkowicz N (2004) Class imbalances versus small disjuncts. *ACM SIGKDD Explor Newslett* 6(1):40–49
- Labatut V, Cherifi H (2011) Evaluation of performance measures for classifiers comparison. arXiv preprint [arXiv:1112.4133](https://arxiv.org/abs/1112.4133)
- Lee E, Jang Y, Yoon D-M, Jeon J, Yang S-I, Lee S-K, Kim D-W, Chen PP, Guitart A, Bertens P et al (2018) Game data mining competition on churn prediction and survival analysis using commercial game log data. *IEEE Trans Games* 11(3):215–226
- Mehra N, Gupta S (2013) Survey on multiclass classification methods
- Mishra A, Reddy US (2017) A comparative study of customer churn prediction in telecom industry using ensemble based classifiers. In: 2017 International conference on inventive computing and informatics (ICICI), pp 721–725. IEEE
- Ngai EW, Xiu L, Chau DC (2009) Application of data mining techniques in customer relationship management: a literature review and classification. *Expert Syst Appl* 36(2):2592–2602
- Nguyen NN, Duong AT (2021) Comparison of two main approaches for handling imbalanced data in churn prediction problem. *J Adv Inf Technol* 12(1)
- Qureshi SA, Rehman AS, Qamar AM, Kamal A, Rehman A (2013) Telecommunication subscribers' churn prediction model using machine learning. In: Eighth international conference on digital information management (ICDIM 2013), pp 131–136. IEEE
- Sammut C, Webb GI (2011) *Encyclopedia of machine learning*. Springer
- Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. *Inf Process Manag* 45(4):427–437
- Tang Z, Lu J, Wang P (2015) A unified biologically-inspired prediction framework for classification of movement-related potentials based on a logistic regression model. *Cogn Comput* 7(6):731–739
- Tharwat A (2020) Classification assessment methods. *Appl Comput Inform*
- Verbeke W, Martens D, Mues C, Baesens B (2011) Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Syst Appl* 38(3):2354–2364
- Xie H, Devlin S, Kudenko D, Cowling P (2015) Predicting player disengagement and first purchase with event-frequency based data representation. In: 2015 IEEE conference on computational intelligence and games (CIG), pp 230–237. IEEE
- Zhang R, Li W, Tan W, Mo T (2017) Deep and shallow model for insurance churn prediction service. In: 2017 IEEE international conference on services computing (SCC), pp 346–353. IEEE
- Zhou J, Li E, Yang S, Wang M, Shi X, Yao S, Mitri HS (2019) Slope stability prediction for circular mode failure using gradient boosting machine approach based on an updated database of case histories. *Saf Sci* 118:505–518

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.