Proceedings of Machine Learning Research 183:112–126, 2022 Full Paper - LIDTA 2022

Assessing the Robustness of Ordinal Classifiers against Imbalanced and Shifting Distributions

Thomas Bonnier THOMAS.BONNIER@SOCGEN.COM
Benjamin Bosch BENJAMIN.BOSCH@SOCGEN.COM
Société Générale

Editor: Nuno Moniz, Paula Branco, Luís Torgo, Nathalie Japkowicz, Michał Woźniak and Shuo Wang.

Abstract

Ordinal classification aims to categorize instances into ordered classes. An underrated or overrated prediction can have significant impacts in applications such as credit rating. Ordinal approaches based on Machine Learning (ML) algorithms can be employed to capture nonlinear patterns. However, under conditions such as lack of training data, their generalization power can be adversely impacted. In this paper, we propose to experimentally assess the robustness of various ordinal classifiers, with a focus on risk rating tasks. We suggest two types of scenarios to evaluate robustness in Machine Learning: lack of training data and data distribution shift. We also propose the *ordinal classifier chains*, an extension of the multi-label classifier chains to ordinal tasks. It uses a lightweight bit layout to encode the labels and employs the chain of classifiers to form a connected structure. Using various evaluation metrics, we compare a selection of ML models under different robustness tests. The models are evaluated on a specific risk rating dataset with significant class imbalance. This benchmark offers a picture of which ML models might be more robust in various data contexts.

Keywords: Ordinal classification, distribution shift, model robustness, data scarcity, class imbalance, risk rating, evaluation metrics.

1. Introduction

Risk assessment for credit or compliance activities in finance require classifying instances according to ordered classes, also called ratings. The objective of ordinal classification is to assign such an ordinal label to an instance, based on covariates $\boldsymbol{x}$. As mentioned by Frank and Hall (2001), ordinal variables distinguish from interval and ratio quantities as the difference between categories cannot be quantified. For instance, a risk rating could be assessed as *No Risk, Minor, Moderate,* or *High*.

Various modeling approaches can be employed for ordinal classification tasks. *Machine Learning* (ML) classifiers are one promising option. However, their inductive nature exposes them to specific limitations. First, training data scarcity can affect their ability to generalize well on new data (Prusa et al., 2015; Sordo and Zeng, 2005). Second, dataset shifts can arise between the Source dataset S (i.e., training/validation data) and the Target dataset T (i.e., real-world data) respective joint distributions: $P_S(\boldsymbol{x}, y) \neq P_T(\boldsymbol{x}, y)$ for a set of covariates $\boldsymbol{x}$ and label y. They are due to non-stationary environments and prove to be harmful as well (Moreno-Torres et al., 2012). These issues can be exacerbated in the context of class imbalance. For instance, the data distribution could shift towards risky classes after the

model has been trained on very few observations of those minority classes. Further, some modeling architectures for ordinal tasks are not parsimonious, e.g., methods based on a binary decomposition of the label (Gutiérrez et al., 2015). Lastly, other ML configurations, such as multinomial classifiers, do not consider the order of categories.

Scope and Commitment In this paper, we propose to experimentally assess and compare the robustness of 7 ML algorithms in performing ordinal classification tasks in the context of distribution shifts and data scarcity. In addition to existing ML classifiers, we extend the classifier chains, which originally target multi-label tasks (Read et al., 2021), in order to perform ordinal classification. The *ordinal classifier chain* model is based on a lightweight encoding of labels and a chained structure to reflect order by design. The 7 models are evaluated with specific metrics on a risk rating dataset with significant class imbalance. CatBoost, an algorithm for gradient boosting on decision trees (GBDT), has demonstrated to outperform some of the most popular GBDT packages (Prokhorenkova et al., 2018). During the experiments, it is thus used as a standalone model for the multiclass approach, and employed as a base learner in the ordinal binary decomposition approaches (ordinal classifier chain, nested dichotomy, and ordered partition models). This specific benchmark reveals some ML models' limitations when facing robustness tests. We show which approaches offer the best trade-off between accuracy, robustness, *2-notch error rate*, *Average Mean Absolute Error* (AMAE), *Maximum Mean Absolute Error* (MMAE), and computational cost. The class imbalance aspects are assessed thanks to the use of specific metrics, namely the AMAE and the MMAE.

2. Related Work

Ordinal approaches Gutiérrez et al. (Gutiérrez et al., 2015) propose a taxonomy for ordinal regression models and encoding of labels. There are 3 types of ordinal modeling approaches.

First: multinomial models, such as random forest (Breiman, 2001) or CatBoost classifiers, are called naive approaches as they ignore category order. To answer the explainability stake, *Explainable Boosting Machine* is a specific tree-based boosting *Generalized Additive Model* (GAM) (Nori et al., 2019), which automatically detects pairwise interactions. As a side note, the terms *multinomial* and *multiclass* are used interchangeably in this paper.

Next, the ordinal binary decomposition approach mostly depends on the encoding choice for the label. *Ordered partitions* split a *k-class* ordinal problem into $k-1$ classifications, as displayed in Figure 1(*a*). The labels are thus encoded using $k-1$ binary variables Y_i with values $I(Y_i > c_i)$, with I the indicator function. Various binary classifiers can be employed, for instance decision trees (Frank and Hall, 2001). *Nested dichotomies* can be used for ordinal classification as well (Frank and Kramer, 2004). They form a tree where at each node, a classifier partitions the data in two sets of disjointed classes. The final leaves contain only the data related to one class. A multi-output model is another option: neural networks can be trained with ordered partitions and a specific loss function to ensure consistent predicted ranks (Cao et al., 2020). A *classifier chain* model falls into that category as well. This is a multi-output model which exploits the relationships between classifiers. As displayed in Figure 2(*a*), each classifier's output is used as a feature for the subsequent classifiers along with the feature vector $\boldsymbol{x}$ (Read et al., 2021). In order to

Y	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
c_1	0	0	0	0	0	0	0
c_2	1	0	0	0	0	0	0
c_3	1	1	0	0	0	0	0
c_4	1	1	1	0	0	0	0
c_5	1	1	1	1	0	0	0
c_6	1	1	1	1	1	0	0
c_7	1	1	1	1	1	1	0
c_8	1	1	1	1	1	1	1

(*a*) Ordered partitions

Y	Y_1	Y_2	Y_3
c_1	0	0	0
c_2	0	0	1
c_3	0	1	0
c_4	0	1	1
c_5	1	0	0
c_6	1	0	1
c_7	1	1	0
c_8	1	1	1

(*b*) Bit encoding

Y	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
c_1	0	0	-	0	-	-	-
c_2	0	0	-	1	-	-	-
c_3	0	1	-	-	0	-	-
c_4	0	1	-	-	1	-	-
c_5	1	-	0	-	-	0	-
c_6	1	-	0	-	-	1	-
c_7	1	-	1	-	-	-	0
c_8	1	-	1	-	-	-	1

(*c*) Nested dichotomy

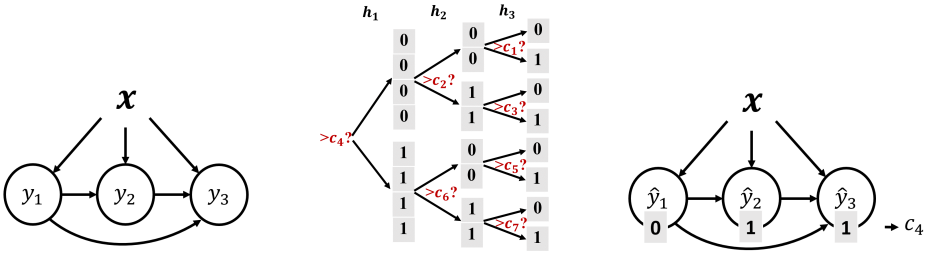
Figure 1: Encoding of labels with 8 ordered categories.

address the problem of class imbalance in ordinal classification, Cruz et al. (2018) introduce a pairwise scoring ranking approach where observations are trained in pairs. Lastly, it is worth noting that while some binary decomposition schemes consider the order of classes (e.g., ordered partition, nested dichotomy), other approaches don't (e.g., one-versus-all).

Third, threshold models are regressions using intervals to specify the order. The *ordinal logistic regression* with proportional odds is one of these (Agresti, 2010; McCullagh, 1980). Lastly, ordinal forests use latent scores instead of class values by minimizing the out-of-bag errors (Hornung, 2020).

Rating with ML The *rating* model is a financial tool used in risk management. For instance, credit rating is a forward-looking credit risk assessment of a borrower in terms of default likelihood, through an ordinal scale from low to high risk. It thus supports credit granting decisions such as loan approval or rejection, and interest to charge. To better understand the drivers of firm's creditworthiness, Hirk et al. (Hirk et al., 2019) consider multivariate ordinal regression models with a latent variable specification and correlated error terms. In that case, the ratings from various credit rating agencies are employed as dependent variables, and firm-level and market information as covariates. Despite their lack of interpretability, ML models can perform just as well or even better in credit rating prediction (Gambacorta et al., 2019). Different ML algorithms have been investigated, such as gradient boosting and deep learning (Petropoulos et al., 2019).

Robustness in Machine Learning ML model performance is sensitive to training data size (Althnian et al., 2021; Prusa et al., 2015; Sordo and Zeng, 2005) . This is exacerbated with over-parametrized models such as deep neural networks (Thompson et al., 2020). After model deployment, dataset shifts can surface, generate model uncertainty, and damage its performance (Rabanser et al., 2019; Szegedy et al., 2013). In that case, training data is not representative of real-world data anymore. Two examples are covariate shift and prior probability shift (Moreno-Torres et al., 2012). The former occurs when, for a set of covariates $\boldsymbol{x}$ and label y, $P_S(y|\boldsymbol{x}) = P_T(y|\boldsymbol{x})$ and $P_S(\boldsymbol{x}) \neq P_T(\boldsymbol{x})$. The latter is defined as $P_S(\boldsymbol{x}|y) = P_T(\boldsymbol{x}|y)$ and $P_S(y) \neq P_T(y)$, with S and T the Source and Target datasets respectively. $P(y|\boldsymbol{x})$ denotes the conditional probability of $Y = y$ knowing $\boldsymbol{X} = \boldsymbol{x}$. Risk



(a) Multi-output classifier chains (b) Classifiers' tasks on bit encoding (c) Ordinal classifier chains during inference

Figure 2: Classifier chains.

rating data is usually prone to class imbalance as severe events, such as fraud or credit default, might rarely occur (Bischl et al., 2016; Liu et al., 2020). Data scarcity could hit high-risk classes or data distributions could shift towards those classes. The potential impacts of these issues can be amplified by class imbalance.

3. Methodology

In this section, we extend the multi-label classifier chains in order to perform ordinal classification. With this goal in mind, we propose a specific binary decomposition, coined *bit encoding*, which minimizes the number of base learners.

Notations and objectives An instance $\boldsymbol{x} \in F$ the instance space, while $y \in \Upsilon$ a set of k rating categories $c_1 \prec ... \prec c_k$. When the ratings are encoded in p binary variables (e.g., ordered partition case), $\boldsymbol{y} = (y_1, ..., y_p) \in \{0,1\}^p$ denotes the corresponding binary vector. Observations $\{(\boldsymbol{x}, y)\}$ are assumed to be drawn independently according to an arbitrary probability distribution $P(\boldsymbol{X}, Y)$ on $F \times \Upsilon$. Our goal is to learn $\hat{p}(y|\boldsymbol{x})$; an accurate, consistent, and parsimonious ordinal predictor of true unknown $P(Y = y|\boldsymbol{X} = \boldsymbol{x}), \forall \boldsymbol{x} \in F$, for any $y \in \Upsilon$. Such a classifier should generalize well when tested on an independent dataset.

Ordinal classifier chains Given we need 3 questions to guess a number between 1 and 8, we can use a bit encoding representation of the ordinal labels. With k ordered values, the labels are thus encoded as increasing binary numbers of length $\lceil \log_2(k) \rceil$ bits. This reduces the number of classifiers to be trained to $c = \lceil \log_2(k) \rceil$. Each rating category is thus represented by a specific $\boldsymbol{y} \in \{0,1\}^c$. For example, we just need *3-bit* numbers for 8 ordered categories. Figure 1(*b*) with k=8 shows that bit encoding is more parsimonious. Each binary variable Y_i plays a key role in separating specific categories or group of categories. Figure 2(*b*) shows that the role of the first classifier h_1 will be to separate (c_1, c_2, c_3, c_4) from (c_5, c_6, c_7, c_8). The objective of the second classifier h_2 will be to separate (c_1, c_2) from (c_3, c_4), and (c_5, c_6) from (c_7, c_8). Lastly, the final classifier h_3 will identify each category uniquely. For this type of encoding, a memory is thus required to secure the consistency in predicted bits. This role is ensured by the chains between the (Y_i)s. We define the ordinal

classifier chain model as $h(\boldsymbol{x}) : F \to \{0,1\}^{\lceil \log_2(k) \rceil}$ with classifiers $h_i : F \times \{0,1\}^{i-1} \to \{0,1\}$. The features $\hat{y}_{l<i}$ denote the predicted labels preceding classifier h_i. For $i \in [\![1, \lceil \log_2(k) \rceil]\!]$, we have:

$$\hat{y}_i = h_i(\boldsymbol{x}, \hat{y}_{l<i}) = \underset{y_i \in \{0,1\}}{\arg\max}\, \hat{p}(y_i | \boldsymbol{x}, \hat{y}_{l<i}) \tag{1}$$

During inference, the predicted label corresponds to the combination of the output of each classifier based on *argmax*, by following a single path through the chain. This *greedy* inference is a low-cost approximation of $\arg\max_{\boldsymbol{y} \in \{0,1\}^c} P(\boldsymbol{y}|\boldsymbol{x})$, with c denoting the number of classifiers in the chain (also $\arg\max_{\boldsymbol{y} \in \{0,1\}^c} \prod_{i=1}^{c} P(y_i | y_{l<i}, \boldsymbol{x})$). However, it has the advantage of remaining computationally tractable when there are many outputs (Read et al., 2021). This approach also corresponds to a rough minimization of the subset zero-one loss $I(\hat{\boldsymbol{y}} \neq \boldsymbol{y})$ (Dembczynski et al., 2012), or *exact match* maximization. Figure 2(*c*) shows an example during inference where the model predicts c_4.

Do the connections between the model outputs really add value? If we assume they offer no contribution to the model predictions, we can just employ the bit encoding structure without the chains between the outputs. We propose a simple example to compare the latter model to the ordinal classifier chains. We consider a training subset with 7 observations from a small feature region R, with true labels $\{c_1, c_1, c_1, c_2, c_2, c_4, c_4\}$, encoded as $\{00, 00, 00, 01, 01, 11, 11\}$. If, during inference, the model without chains is exposed to a new instance from R, the model will predict 01 (or c_2) in a greedy fashion. In fact, given the training subset, it will first estimate $\arg\max_{y_1 \in \{0,1\}} P(y_1|\boldsymbol{x})_{\boldsymbol{x} \in R}$ and then $\arg\max_{y_2 \in \{0,1\}} P(y_2|\boldsymbol{x})_{\boldsymbol{x} \in R}$. We remark that $\max P(y_1|\boldsymbol{x})_{\boldsymbol{x} \in R} = 5/7$ is reached when $y_1 = 0$, and $\max P(y_2|\boldsymbol{x})_{\boldsymbol{x} \in R} = 4/7$ is reached for $y_2 = 1$. On the other hand, the ordinal classifier chains will estimate $\arg\max_{y_1 \in \{0,1\}} P(y_1|\boldsymbol{x})_{\boldsymbol{x} \in R}$ and then $\arg\max_{y_2 \in \{0,1\}} P(y_2|y_1, \boldsymbol{x})_{\boldsymbol{x} \in R}$. Considering the training subset, we note that $\max P(y_1|\boldsymbol{x})_{\boldsymbol{x} \in R} = 5/7$ is reached when $y_1 = 0$, and $\max P(Y_2 = y_2 | Y_1 = 0, \boldsymbol{X} = \boldsymbol{x})_{\boldsymbol{x} \in R} = 3/5$ is reached for $y_2 = 0$. In that case, it will predict 00 (or c_1), which is different from the prediction of the model without chains. It is also an exact match of $\arg\max_{\boldsymbol{y} \in \{0,1\}^2} P(\boldsymbol{y}|\boldsymbol{x})_{\boldsymbol{x} \in R}$, which is 00 (or c_1) according to the training subset. This counterexample shows that the chains between the outputs of the model can provide value. These connections help to partition the rating space by considering the dependencies between the (Y_i)s. Going back to our training subset, it is obvious that $P(Y_2 = 0|Y_1 = 0) = 3/5 \neq P(Y_2 = 0|Y_1 = 1) = 0$ and $P(Y_2 = 1|Y_1 = 0) = 2/5 \neq P(Y_2 = 1|Y_1 = 1) = 1$, showing that Y_2 depends on Y_1. Lastly, another angle would be to experimentally demonstrate the role of classifiers' outputs in the decisions of subsequent classifiers by considering the feature contributions in the chain.

Table 1: Comparison table of main ordinal techniques.

Methodology	**Advantages**	**Weaknesses**
Multinomial classifier	Wide range of classifiers, easy to implement	Naive approach, used for multiclass problems
Ordinal logistic regression	Model parsimony (features), explainable	Constraints such as proportional odds
Standard binary decomposition	Wide range of classifiers	Inconsistencies, lack of parsimony (estimators)
Nested dichotomy	Wide range of classifiers, specific nested models	Lack of parsimony (nested estimators)
Ordinal classifier chains	Wide range of classifiers, lightweight	Greedy inference

General versus specific model Table 1 compares the main ordinal approaches, with their respective strengths and weaknesses. It is worth highlighting the differences between the nested dichotomy and ordinal classifier chain models:

- During training: as displayed in Figure 1(*c*), the nested dichotomy trains $k-1$ classifiers on data partitions. On the other hand, the ordinal classifier chain model trains $\lceil \log_2(k) \rceil$ classifiers, and the whole dataset is used each time. The ordinal classifier chain model employs more features, the $(\hat{y}_{l<i})$s, while the nested dichotomy requires more classifiers. Therefore, with many rating categories, the number of classifiers would soar in the nested dichotomy case.

- During inference: nested dichotomies go through a limited number of classifiers while the ordinal classifier chain model goes through all of them.

The ordinal classifier chains and nested dichotomies thus follow two different paradigms: general versus specific classifiers. We will compare their performance in the experimental phase to understand which approach (general or specific) proves to be more accurate in the context of class imbalance.

Bit-output decoding and inconsistency correction Bit-output decoding can be carried out with a simple dictionary where, for instance, prediction 001 is mapped to category c_2. If the number of ordered categories k is strictly between 2^{n-1} and 2^n, inconsistencies in bit predictions may occur during inference. A prediction $\hat{\boldsymbol{y}}$ cannot be mapped to a category if it does not exist in the possible reference configurations. For instance, in a *3-class* ordinal problem, if 00, 01, and 10 have been used to respectively encode categories c_1, c_2, and c_3, a prediction such as 11 will not find any match. To correct inconsistencies in bit predictions, we suggest computing the probability $\prod_{j=1}^{c} \hat{p}(y_j|\hat{y}_{l<j}, \boldsymbol{x})$ for each of the k possible paths (k categories) and selecting the optimal one. Going back to our example, we compare $\hat{p}(y_2|y_1, \boldsymbol{x}) \times \hat{p}(y_1|\boldsymbol{x})$ for (y_1, y_2) in $\{(0,0),(0,1),(1,0)\}$, and select the most likely configuration. If the number of rating categories is significant, this could increase the computational cost. However, few inconsistencies were observed during the experiments.

Table 2: Model list and implementation packages.

Model	Acronym	Python 3 package
Ordinal logistic regression, proportional odds	OLR	*statsmodels* v0.13.0.dev0
Rank-consistent neural network (Cao et al., 2020)	NN	*Coral Ordinal* (Kennedy, 2020)
Ordinal classifier chains	OCC	*CatBoost* base learner
CatBoost (multiclass) (Prokhorenkova et al., 2018)	CAT	*CatBoost* 0.26.1
Nested dichotomy	ND	*nd* (Melnikov and Hüllermeier, 2018), *CatBoost* base learner
Ordered partitions	OP	*CatBoost* base learner
Explainable Boosting Machine (Nori et al., 2019)	EBM	*interpret* 0.2.7

4. Experiments

4.1. Settings

Use case The Lending Club dataset contains peer-to-peer lending data with personal and business loans for individuals (George, 2018). It is available under *CC0: Public Domain*

license. We limit the scope to 2018 accepted loan data, i.e., 56311 observations. The risk rating is the *grade* variable estimated by Lending Club at issuance time. In this use case, the *grade* validated and assigned by Lending Club is thus considered to be the true label. The rating has 7 levels from less risky A to riskier G. The minority classes are the high-risk ratings F and G. We employ 21 variables, with a blend of continuous and categorical types. The objective is to predict (i.e., replicate) the rating using only the information available at loan issuance time $t = 0$. Therefore, the loan status at $t > 0$ (default or not) is not used. Variables are financial ratios (e.g., *debt-to-income*), credit record data (e.g., *revolving line utilization rate*), borrower characteristics (e.g., *annual income*) or current loan information (e.g., *loan amount*).

Model inventory and evaluation A selection of 7 modeling approaches are evaluated. Table 2 describes the models and implementation packages. To make comparisons more relevant, the ordinal classifier chains, the nested dichotomy, and the ordered partition models use the same base learner, CatBoost classifier, which is an algorithm for gradient boosting on decision trees. We convert the ratings to numerical labels from 1 to 7 for the multiclass models (CAT, EBM). OLR employs ordered categories. Lastly, a specific label encoding scheme is used for the other models: ordered partitions for NN and OP, bit encoding for OCC, and nested dichotomy partitions for ND. Models are evaluated using various metrics:

- Accuracy is defined as the rating exact match by using the *0-1 loss*: $1 - (1/N)\sum_{i=1}^{N} I(\hat{c}_i \neq c_i)$, where the sum is performed over N test instances, and with c_i and $\hat{c}_i$ denoting the true and predicted ratings, respectively. The prediction of a multiclass or binary classifier is based on the $argmax$ of predicted class probabilities.

- The 2-notch error rate is defined as: $(1/N)\sum_{i=1}^{N} I(|\hat{c}_i - c_i| \geq 2)$. It corresponds to a weighted error rate, considering model under/over-rating of at least 2 categories from the true rating. We assume here that the ratings are numerical with $c_i \in [\![1, k]\!]$. This metric is a meaningful performance measure in rating tasks as it focuses on significant errors.

- The Average Mean Absolute Error (AMAE) is defined as $(1/k)\sum_{q=1}^{k} MAE_q$, where $MAE_q = (1/N_q)\sum_{i=1}^{N_q} |\hat{c}_i - q|$ on the partition related to true rating $q \in [\![1, k]\!]$ (Cruz et al., 2018; Perez-Ortiz et al., 2014). It is relevant for ordinal classification problems with class imbalance as the share of each class is not taken into account in the average computation. High MAE values on minority classes would thus influence AMAE. The fact that the variable is considered as a cardinal variable is an assumption of this measure.

- The Maximum Mean Absolute Error (MMAE) is defined as $max(MAE_q | q = 1, ..., k)$. It focuses on the class with the greatest mean distance between the predicted classes and the true classes. This metric is thus appropriate to assess the model performance for ordinal classification problems with class imbalance (Cruz et al., 2018; Perez-Ortiz et al., 2014). In fact, a high MAE value on the minority class would affect MMAE.

- Lastly, the computational cost is the time to perform pre-processing (e.g., label encoding), model training, optimization, and evaluation on 10000 observations (test set). Experiments are run with a CPU 1.9 GHz and 32 GB RAM.

For the ordinal logistic regression, variables with *Variance Inflation Factors* (VIF) greater than 10 are discarded (Hair et al., 2006). The neural network has 2 hidden layers (512 and 32 units) and 2 drop-out layers (0.2 rate). It employs *ReLu* activations. *Adam* optimizer (Kingma and Ba, 2014) is used with learning rate tuning initialized at 5e-5 and batch size of 32. Training is performed with early stopping on validation accuracy with a patience of 5.

For the 5 other models, the number of boosting iterations is optimized using cross-validation with a simple grid search: [10, 100, 500, 1500]. We select the model with the best mean accuracy over cross-validation folds. With 7 rating categories, the ordinal classifier chain model employs the first 7 *3-bit* binary numbers for label encoding: 000 (c_1), 001 (c_2), 010 (c_3), 011 (c_4), 100 (c_5), 101 (c_6), and 110 (c_7).

Robustness scenarios Two shift scenarios are tested:

- *Baseline* or *standard* scenario: no distribution shift occurs between training and test data. The distribution is noticeably skewed towards low-risk ratings.
- *Adverse* scenario: covariate shift occurs, and label proportions differ between training and test datasets. The distribution moves towards classes with higher risk.

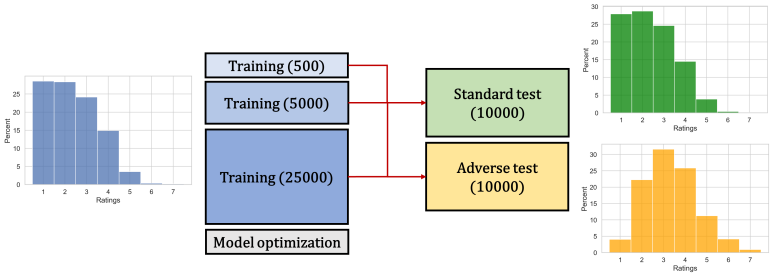


Figure 3: Training and evaluation pipeline with (dataset size), shift scenario and corresponding rating distribution. Model optimization includes cross-validation (number of iterations for the boosting algorithms) and validation (number of epochs through early stopping for the neural network) using folds of the training set.

The experiments are run over 2 dataset shift scenarios (standard, adverse), 3 training data sizes (500, 5000, 25000) and 5 random states. Random states correspond to different seeds in the dataset splits during the creation of training and test datasets. The 6 robustness scenarios are referred to as "shift scenario-training data size", e.g., *adverse-500*. We could have chosen a more progressive and granular approach for the training data sizes; this remains to be studied. As displayed in Table 3, a training data size of 500 corresponds to a situation of data scarcity, where volumes are low for most of the labels. For a given random state and training data size, all models are trained on the same data. For a given random state, all models are evaluated on 2 test datasets with distinct shift scenario: standard test and adverse test datasets. Figure 3 summarizes the training and evaluation pipeline for one

random state. As displayed in Table 4, the label distribution is thus significantly imbalanced. The training set includes very few observations related to ratings 6 and 7. However, in the adverse scenario, there is an increasing number of instances of those classes.

Table 3: Label distribution in training dataset (count), averaged over 5 random states, by data size.

Label /Volume	500	5000	25000
1	138.6	1391.0	7000.4
2	139.2	1420.2	7107.6
3	127.4	1219.0	6129.4
4	71.0	762.0	3711.2
5	21.4	187.6	944.2
6	1.8	16.0	86.2
7	0.6	4.2	21.0

Table 4: Rating distribution (%) averaged over 3 data sizes and 5 random states for each dataset: Training, Standard Test, Adverse Test.

Label	Training	Standard Test	Adverse Test
1	27.85	28.06	4.10
2	28.22	28.37	22.27
3	24.79	24.89	31.49
4	14.76	14.50	25.97
5	3.94	3.74	11.19
6	0.34	0.35	4.11
7	0.10	0.09	0.86

Shift generation As creating synthetic data can generate label inconsistencies, we keep the original dataset unchanged. Shift generation is performed through *k-means* clustering (Lloyd, 1982) applied to the whole (covariate) dataset. Categorical covariates are one-hot-encoded. Instances within each cluster will have their proper covariate distribution. Given these covariates are correlated with the target variable, the clusters will thus have different label distributions. In fact, distant clusters are more likely to exhibit significantly different label proportions. The adverse shift is thus produced by sampling (without replacement) heterogeneously over all the clusters for the training data and the adverse test data. Figure 4 describes the cluster sampling process. If the test dataset is constructed by sampling mainly from cluster 1 and the training set by sampling from the other clusters, significant covariate and rating shifts will be observed.

4.2. Results

4.2.1. Accuracy robustness trade-off

Figure 5(*a*) displays the average accuracy over all 6 scenarios versus the absolute change in accuracy computed between 2 opposite scenarios (*standard-25000* and *adverse-500*). 3 groups stand out. First, explainable models, i.e., the ordinal logistic regression and Explainable Boosting Machine, are more robust against data scarcity and distribution shift, but perform poorly on average. It is worth mentioning that in the *interpret* package, the *interaction* parameter is forcefully set to 0 for multiclass tasks. At the other end of the

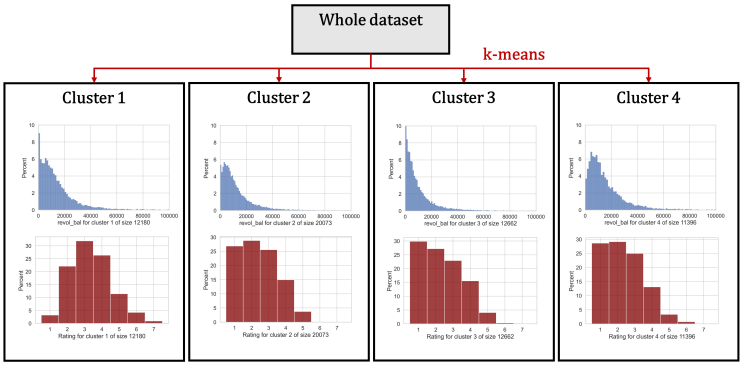
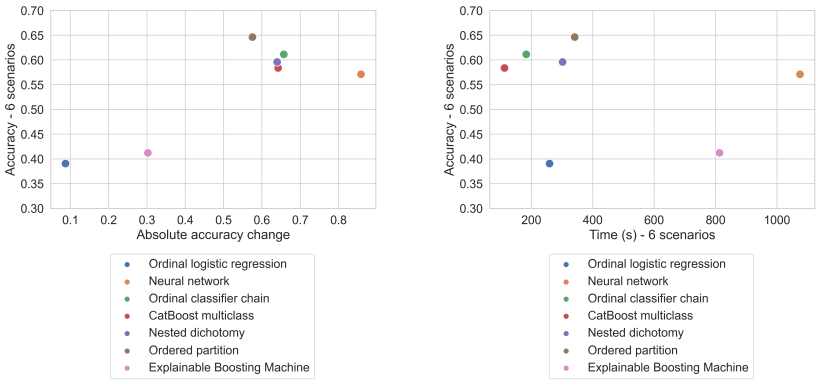


Figure 4: Shift generation based on cluster sampling. One covariate (*revolving balance*) and rating distributions are displayed respectively in blue and red by cluster.

spectrum, the neural network, as a complex model, achieves good average accuracy with poor adaptation to *adverse-500*. The last group, with 4 models, achieves superior accuracy and in-between robustness. Within this group, the ordered partition model has the best average accuracy over all scenarios, while the ordinal classifier chain model comes second. Overall, the *general* ordinal chain approach generalizes slightly better over different situations than the *specialized* nested dichotomy. Further, the former, as an *ordinal* approach, is more accurate than CatBoost *multiclass*. However, the ordinal classifier chain is a bit more sensitive to extreme changes when moving from *standard-25000* to *adverse-500*. Although designed for multinomial tasks, CatBoost multiclass proves to be quite competitive.



(*a*) (*b*)

Figure 5: (a) Mean accuracy over 6 scenarios versus absolute change between *standard-25000* and *adverse-500* accuracies, averaged over 5 random states. (b) Accuracy versus computation time in seconds, averaged over 6 scenarios and 5 random states.

Table 5 highlights that the ordinal logistic regression is robust in difficult conditions but does not really take advantage of a larger training dataset. As expected, it fails to capture nonlinear patterns. On the other hand, the neural network cannot generalize well when it is exposed to both training data scarcity and distribution shifts. It may overfit and require even more regularization. The ordered partition model confirms its good performance across various scenarios. Compared to the nested dichotomy and CatBoost multiclass models, the ordinal classifier chains mostly stand out when facing adverse dataset shifts provided that there is enough training data (5000 and 25000). In fact, the ordinal classifier chain model proves to be more robust to dataset shifts than training data scarcity.

Table 5: Accuracy (standard deviation), by model and scenario, averaged over 5 random states. Best performance for given scenario is in bold. Model acronyms are defined in Table 2.

Scenario	OLR	NN	OCC	CAT	ND	OP	EBM
adverse-500	**0.344** (0.01)	0.043 (2e-3)	0.247 (0.04)	0.247 (0.03)	0.258 (0.05)	0.334 (0.02)	0.280 (0.04)
adverse-5000	0.360 (0.01)	**0.606** (0.07)	0.519 (0.03)	0.478 (0.01)	0.477 (0.06)	0.574 (0.02)	0.360 (0.01)
adverse-25000	0.368 (4e-3)	0.840 (0.02)	0.830 (0.01)	0.716 (0.01)	0.786 (0.02)	**0.845** (0.01)	0.380 (0.01)
standard-500	0.413 (0.01)	0.281 (0.01)	0.409 (0.02)	0.412 (0.01)	0.411 (0.01)	**0.427** (0.01)	0.381 (0.02)
standard-5000	0.426 (4e-3)	0.755 (0.05)	0.759 (0.01)	0.761 (0.01)	0.745 (0.02)	**0.788** (0.01)	0.489 (0.02)
standard-25000	0.432 (0.01)	0.902 (0.02)	0.904 (5e-3)	0.889 (4e-3)	0.898 (0.01)	**0.909** (0.01)	0.583 (4e-3)

4.2.2. Computational cost

Figure 5(*b*) demonstrates that CatBoost multiclass is the most computationally efficient model, followed by the ordinal classifier chain model. The latter offers a good trade-off between accuracy and computation time thanks to its lightweight structure. On the other hand, the ordered partition approach is less parsimonious, due to a heavy encoding structure with more classifiers. The nested dichotomy model is also less efficient given the number of nested classifiers. Lastly, the neural network model would require more resources for a faster computation time, especially with large datasets.

Table 6: 2-notch error rate (standard deviation), by model and scenario, averaged over 5 random states. Best performance for given scenario is in bold.

Scenario	OLR	NN	OCC	CAT	ND	OP	EBM
adverse-500	**0.186** (0.013)	0.729 (0.005)	0.330 (0.058)	0.329 (0.043)	0.321 (0.065)	0.221 (0.026)	0.289 (0.070)
adverse-5000	0.153 (0.008)	**0.014** (0.007)	0.087 (0.032)	0.113 (0.038)	0.123 (0.048)	0.048 (0.020)	0.188 (0.018)
adverse-25000	0.151 (0.002)	**0.002** (0.000)	0.003 (0.001)	0.029 (0.003)	0.026 (0.013)	0.006 (0.001)	0.166 (0.009)
standard-500	0.157 (0.010)	0.445 (0.008)	0.192 (0.018)	0.190 (0.007)	0.187 (0.012)	**0.124** (0.004)	0.216 (0.016)
standard-5000	0.139 (0.003)	0.016 (0.007)	0.028 (0.002)	0.041 (0.002)	0.048 (0.016)	**0.013** (0.001)	0.155 (0.002)
standard-25000	0.139 (0.002)	**0.002** (0.001)	**0.002** (0.001)	0.005 (0.001)	0.007 (0.003)	**0.002** (0.001)	0.131 (0.003)

4.2.3. 2-notch error rate

Table 6 displays the 2-notch error rate across the 6 scenarios. For complex models, this metric is quite sensitive to training data size and distribution shifts. This trend is pronounced for the neural network. However, with more training data, this metric converges to 0, especially for the neural network, the ordinal classifier chains, and the ordered partition model, even when distribution shifts appear. We note that the ordinal logistic regression achieves the lowest rate if we focus on *adverse-500*, but its 2-notch error rate remains high around 14-20% across all the scenarios. For complex models exposed to *small data* contexts, the optimization of regularization hyperparameters to minimize this metric remains to be seen.

4.2.4. AMAE

AMAE is affected by high MAE values on minority ratings 6 and 7. Table 7 shows that AMAE values can be significant for the neural network in small data contexts. For complex models, AMAE values noticeably decrease with more training data. The adverse scenario sometimes displays a lower AMAE value than the standard case. In fact, in the standard case, there are often very few test instances with true ratings 6 or 7, which can produce pretty high MAE values on those classes.

Table 7: AMAE (standard deviation), by model and scenario, averaged over 5 random states. Best performance for given scenario is in bold.

Scenario	OLR	NN	OCC	CAT	ND	OP	EBM
adverse-500	**1.367** (0.07)	2.538 (0.23)	1.762 (0.09)	1.783 (0.07)	1.786 (0.08)	1.646 (0.07)	1.778 (0.16)
adverse-5000	1.332 (0.00)	**0.583** (0.10)	1.070 (0.13)	1.302 (0.04)	1.349 (0.18)	0.972 (0.08)	1.407 (0.03)
adverse-25000	1.314 (0.00)	**0.312** (0.07)	0.370 (0.02)	0.680 (0.02)	0.677 (0.19)	0.460 (0.02)	1.319 (0.01)
standard-500	**1.646** (0.07)	2.554 (0.15)	1.872 (0.08)	1.841 (0.03)	1.921 (0.04)	1.740 (0.05)	2.011 (0.09)
standard-5000	1.632 (0.06)	**0.741** (0.10)	1.012 (0.05)	1.183 (0.08)	1.250 (0.14)	0.899 (0.04)	1.619 (0.06)
standard-25000	1.659 (0.06)	**0.454** (0.06)	0.468 (0.05)	0.615 (0.08)	0.689 (0.13)	0.482 (0.05)	1.487 (0.08)

4.2.5. MMAE

MMAE is affected by a high MAE value on rating 7. MMAE values, displayed in Table 8, demonstrate that the greatest distance between predicted and true labels can be significant in small data contexts for all models. However, NN, OCC and OP can take advantage of more data to reach pretty good results. As mentioned previously for AMAE, MMAE can sometimes be lower in the adverse case, as the model is evaluated on more test instances with a true rating of 7, producing fewer erratic results than in the standard case. Lastly, the *general* OCC model achieves lower MMAE results than the *specific* ND model. A general approach can thus be relevant even for minority classes.

5. Conclusion

In this paper, we experimentally evaluate the robustness of various ML models against training data scarcity and adverse distribution shifts when predicting ratings in the context of class imbalance. We also extend the classifier chains to perform ordinal classification

Table 8: MMAE (standard deviation), by model and scenario, averaged over 5 random states. Best performance for given scenario is in bold.

Scenario	OLR	NN	OCC	CAT	ND	OP	EBM
adverse-500	**2.968** (0.24)	4.246 (0.88)	3.834 (0.09)	3.902 (0.16)	3.919 (0.21)	3.811 (0.20)	3.922 (0.35)
adverse-5000	2.937 (0.05)	**1.204** (0.17)	2.476 (0.40)	3.348 (0.10)	3.334 (0.52)	2.453 (0.21)	3.124 (0.12)
adverse-25000	2.914 (0.04)	**0.986** (0.17)	1.104 (0.08)	1.952 (0.12)	2.213 (0.75)	1.585 (0.07)	2.954 (0.14)
standard-500	**3.955** (0.60)	4.526 (0.69)	4.348 (0.43)	4.304 (0.32)	4.642 (0.14)	4.248 (0.35)	4.642 (0.31)
standard-5000	3.974 (0.39)	**1.905** (0.44)	3.006 (0.25)	3.638 (0.45)	3.551 (0.43)	2.743 (0.26)	3.946 (0.36)
standard-25000	4.133 (0.42)	**1.304** (0.32)	1.791 (0.32)	2.207 (0.60)	2.655 (0.61)	1.788 (0.42)	3.923 (0.58)

tasks. This model leverages a lightweight bit encoding and connected structure. The ordinal logistic regression is less sensitive to distribution shifts than more complex models in small data contexts. Complex models are more likely to overfit but are more accurate when averaging over all scenarios, especially when they capture nonlinear patterns on larger training datasets. By design, tree-based methods or neural networks are able to capture these nonlinearities. The ordered partition model demonstrates good performance across diverse scenarios. It achieves the best accuracy, 2-notch error rate, and AMAE when each of these metrics is averaged over all scenarios. The ordinal classifier chain offers a relatively good compromise between the different evaluation metrics, especially when the training dataset is large. It achieves the second-best accuracy, 2-notch error rate, computational cost, AMAE and MMAE when each of these indicators is averaged over all scenarios. For all models, MMAE can disclose failure zones with a greater distance between predicted and true high-risk ratings in small data contexts. These results are specific to this use case and should be confirmed on additional datasets with more rating categories. Future work will also concentrate on testing a selection of regularization hyperparameters in small data contexts. Additional classifier families (e.g., random forest) could be evaluated.

References

Alan Agresti. *Analysis of ordinal categorical data*, volume 656. John Wiley & Sons, 2010.

Alhanoof Althnian, Duaa AlSaeed, Heyam Al-Baity, Amani Samha, Alanoud Bin Dris, Najla Alzakari, Afnan Abou Elwafa, and Heba Kurdi. Impact of dataset size on classification performance: an empirical evaluation in the medical domain. *Applied Sciences*, 11(2): 796, 2021.

Bernd Bischl, Tobias Kühn, and Gero Szepannek. On class imbalance correction for classification algorithms in credit scoring. In *Operations Research Proceedings 2014*, pages 37–43. Springer, 2016.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140: 325–331, 2020.

Ricardo Cruz, Kelwin Fernandes, Joaquim F Pinto Costa, María Pérez Ortiz, and Jaime S Cardoso. Binary ranking for ordinal class imbalance. *Pattern Analysis and Applications*, 21(4):931–939, 2018.

Krzysztof Dembczynski, Willem Waegeman, and Eyke Hüllermeier. An analysis of chaining in multi-label classification. In *ECAI: European Conference of Artificial Intelligence*, volume 242, pages 294–299, 2012.

Eibe Frank and Mark Hall. A simple approach to ordinal classification. In *European conference on machine learning*, pages 145–156. Springer, 2001.

Eibe Frank and Stefan Kramer. Ensembles of nested dichotomies for multi-class problems. In *Proceedings of the twenty-first international conference on Machine learning*, page 39, 2004.

Leonardo Gambacorta, Yiping Huang, Han Qiu, and Jingyi Wang. How do machine learning and non-traditional data affect credit scoring? new evidence from a chinese fintech firm. *BIS Working Paper*, 2019.

Nathan George. All lending club loan data, kaggle, 2018. URL https://www.kaggle.com/datasets/wordsforthewise/lending-club.

Pedro Antonio Gutiérrez, Maria Perez-Ortiz, Javier Sanchez-Monedero, Francisco Fernandez-Navarro, and Cesar Hervas-Martinez. Ordinal regression methods: survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1): 127–146, 2015.

Joseph F Hair, William C Black, Barry J Babin, Rolph E Anderson, and Ronald L Tatham. *Multivariate Data Analysis*. Pearson Prentice Hall, 2006.

Rainer Hirk, Kurt Hornik, and Laura Vana. Multivariate ordinal regression models: an analysis of corporate credit ratings. *Statistical Methods & Applications*, 28(3):507–539, 2019.

Roman Hornung. Ordinal forests. *Journal of Classification*, 37(1):4–17, 2020.

Chris Kennedy. Coral ordinal package, 2020. URL https://github.com/ck37/coral-ordinal.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yang Liu, Xiang Ao, Qiwei Zhong, Jinghua Feng, Jiayu Tang, and Qing He. Alike and unlike: Resolving class imbalance problem in financial credit risk assessment. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2125–2128, 2020.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

Peter McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 42(2):109–127, 1980.

Vitalik Melnikov and Eyke Hüllermeier. On the effectiveness of heuristics for learning nested dichotomies: an empirical analysis. *Machine Learning*, 107(8):1537–1560, 2018.

Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45 (1):521–530, 2012.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

Maria Perez-Ortiz, Pedro Antonio Gutierrez, Cesar Hervas-Martinez, and Xin Yao. Graph-based approaches for over-sampling in the context of ordinal regression. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1233–1245, 2014.

Anastasios Petropoulos, Vasilis Siakoulis, Evaggelos Stavroulakis, and Aristotelis Klamargias. A robust machine learning approach for credit risk analysis of large loan level datasets using deep learning and extreme gradient boosting. *IFC Bulletins chapters, BIS*, 49, 2019.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Joseph Prusa, Taghi M Khoshgoftaar, and Naeem Seliya. The effect of dataset size on training tweet sentiment classifiers. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 96–102. IEEE, 2015.

Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019.

Jesse Read, Bernhard Pfahringer, Geoffrey Holmes, and Eibe Frank. Classifier chains: a review and perspectives. *Journal of Artificial Intelligence Research*, 70:683–718, 2021.

Margarita Sordo and Qing Zeng. On sample size and classification accuracy: A performance comparison. In *International Symposium on Biological and Medical Data Analysis*, pages 193–201. Springer, 2005.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 2020.