Applied Data Science for Leasing Score Prediction

Giuseppe Cianci[1], Roberto Goglia[1], Riccardo Guidotti[2], Matteo Kapllaj[1]
Roberto Mosca[1], Andrea Pugnana[3], Franco Ricotti[1], and Salvatore Ruggieri[2]

[1] Sadas S.R.L., Casalnuovo di Napoli, Italy
[2] Department of Computer Science, University of Pisa, Pisa, Italy
[3] Scuola Normale Superiore, Pisa, Italy

***Abstract*—We describe the design, the architecture, and the evaluation of the Leasing Score Prediction (LSP) system – a credit scoring and credit rating system for the leasing sector deployed at the Italian association of leasing companies. Due to its challenging objectives, the design and complexity of the LSP system represent a unique contribution to the best practices in the field. We cover requirements by managers, users, and regulations about rigorous backtesting, statistical validation, calibration, explainability, robustness and uncertainty self-assessment. LSP relies on a machine learning model trained on a mixture of data distributions contributed by many associated leasing companies. We describe the technical solutions adopted and report on their performance evaluation, including the management of the data shifts due to the COVID-19 pandemic.**

I. Introduction

Credit scoring is both a well-studied and an extremely relevant application of data science and advanced analytics [1]. Lease scoring, particularly in the application scenario tackled in this paper, brings a number of specific objectives that deserve the development of a tailored system, which we called Leasing Score Prediction (LSP). The complexity in the design of LSP, and the technical solutions adopted, represent a unique and novel contribution to the best practices in the field.

The LSP project started in January 2020 with the aim of developing a risk scoring and risk rating system based on supervised Machine Learning (ML), able to predict defaults of payments by the lessee at the time of contract definition. The available data, collected in a centralized credit information system, are contributed by more than 40 leasing companies associated to the Italian leasing association (Assilea). Data distributions are then a mixture resulting from different target markets, types of lessees/assets, contract conditions, and risk appetites. LSP is expected to outperform a scoring system trained on data of a single associate *(Objective O1)*. The available data may not represent the complete information used for decision-making, e.g., leasing companies do not contribute external credit bureau information or corporate data sources. LSP should impute missing data with external sources *(Objective O2)*. Risk score model performances, such as AUC and calibration error, can broadly vary over (subsets of) the input data distributions, depending on the quality of the input. LSP should provide an uncertainty self-assessment for both the input features and the output scores *(Objective O3)*. Managers aim at and regulation mandate that there is a clear understanding of the underlying ML models adopted, including the explainability of decision drivers and counterfactual analyses *(Objective O4)*. Moreover, users of LSP expect to use the features of the system interactively in what-if scenario, with real-time answers *(Objective O5)*. Finally, the financial capacity of lessees changes over time, e.g., due to the economic cycle or disruptive events such as floods, earthquakes, or pandemics. In particular, Italy was the first Western country to face the COVID-19 outbreak, and to implement social counter-measures, e.g., lock-downs, and fiscal counter-measures, e.g., suspension of credit payments[1]. LSP is then expected to adapt to prior distribution shifts *(Objective O6)*.

The LSP system entered production in April 2022 for a pilot subset of associate leasing companies, and from July 2022, it became gradually subscribable to all of them. We describe the design, the architecture, and the evaluation of LSP, with reference to the above specific objectives. The rest of this paper is structured as follows. First, we discuss the related work in Section 2. Section 3 summarizes the background on credit scoring, and Section 4 introduces the scenario of lease financing. Section 4 discusses the LSP framework. Section 5 focuses on experimental validation. Finally, we summarize lessons learned and possible extensions.

II. Related Work

Credit scoring models assess the creditworthiness of a lender [1], either w.r.t. an individual loan or w.r.t. a portfolio of loans. We tackle the former case: LSP predicts the probability of default w.r.t. a leasing contract when it is is about to start. Let us survey related work on a few key problems to be tackled in a credit scoring application.

Model selection. Statistical and machine learning models have been used for credit scoring in econometrics, banking, finance, and data science [2], [3]. The former methods focus on summarization and (causal) inference, while the latter focus mainly on prediction. Ensemble methods are the state-of-the-art machine learning classification models for credit scoring [4]. They exhibit the best predictive performances in benchmark results [5], [6], also when restricting to simple base classifiers [7]. The superior predictive performance of ensembles was confirmed on our leasing contract data by

[1] See https://en.wikipedia.org/wiki/COVID-19_pandemic_in_Italy.

	[26]	[11]	[27]	LSP
Domain	Loans	Bankruptcy	Mortgages	Leasing
Scoring	XGBoost	LightGBM	XGBoost	Ensemble
Rating		✓		✓
Calibration		✓		✓
Uncertainty				✓
Factual expl.	✓	✓	✓	✓
Counterfactuals				✓
Prior shift				✓
Self-learning				✓
Backtesting	Holdout	OOT 1y×1	Holdout	OOT 3m×23
Deployed	✓			✓

TABLE I: Related credit scoring systems. OOT $d \times n$ is Out of Time with d = test period duration, n = number of periods.

preliminary experiments, for which LSP core model has been set to an ensemble of gradient-boosting models.

Model calibration. Calibration of scoring models aims at producing risk scores that can be interpreted as the probability of default [8]. Calibration techniques can be model-agnostic (e.g., sigmoid, isotonic, beta calibration) [9], or model specific, as the one for decision tree models [10] used in a bank credit scoring systems [11]. The outer ensemble of LSP helps improve the calibration of the base classifier without the need for post-processing techniques.

Explainability. Complex ML scoring models lack the interpretability of their internal logic and the explainability of their outputs. Explainability of AI (XAI) methods [12] are being increasingly used in the context for credit scoring [13]. Post-hoc interpretability of models has been explored, e.g. by feature importance [14]. Explainability of the model's output has been explored, e.g. by feature attribution (such as the Shapley values [15]) in [16], [17]. We also resort to Shapley values for factual explanations of scores. Factual explanations answer the "*why?*" of a model's output. Counterfactuals explanations [18] answer the "*why not?*" question by providing instances that are similar to the one under analysis but that receive a lower risk score. Methods for searching counterfactuals have been proposed in the context of credit scoring [19], [20] and contrasted to expert-generated explanations [21]. The proposed methods do not account for a critical issue in a deployment scenario: computational efficiency, an enabling factor for interactive usage of the credit scoring system in what-if analysis. FasterRisk [22] (fast and accurate interpretable risk scores), markedly faster than its competitors, "produces a collection of high-quality risk scores within minutes". LSP produces risk scores, factual and counterfactual explanations in 1.1 seconds on average.

Data shifts. Probabilistic predictive models, including credit scoring models, suffer from shifts in the probability distribution from training to test data [23], [24]. Several approaches have been considered to mitigate the impact of data shifts for credit scoring [25]. Prior shifts, namely change in the distribution of defaults, directly impact on calibration of predictions. LSP deals with predictable prior shifts, for which an estimate of the new prior is available. Such an estimate is provided by official statistics forecasts.

Related systems. Table I compares LSP to a few complex credit scoring systems that are built and evaluated on real large-scale data. LSP covers the whole spectrum of features.

III. Background on Credit Scoring

Credit scoring and credit rating aim to estimate the lender's probability of default (PD) to meet the contractual obligations.

Credit Scoring. A credit scoring model is a function $s_t : \mathcal{X}_t \rightarrow [0, 1]$ mapping instances from a feature space $\mathcal{X}_t$ to a risk score value between 0 and 1. The feature space distribution and the model can change over time: the subscript t specifies the point of time t at which the model is used. We omit the subscript if no ambiguity arises. The risk score $s(\mathbf{x})$, or simply the score, of a (credit application) instance $\mathbf{x}$ estimates the likelihood that a risk event occurs, such as, for instance, the insolvency within a certain time frame Δt from t. We denote by $y_\mathbf{x}$ the true outcome, namely $y_\mathbf{x} = 1$ if the event actually occurs (positive instance) and $y_\mathbf{x} = 0$ otherwise (negative instance). Notice that $y_\mathbf{x}$ is known only if the credit application is approved and the credit contract starts. We assume that $(\mathbf{x}, y_\mathbf{x})$ are drawn from an unknown distribution $\mathcal{D}_t$. Machine Learning models are built from a dataset of observations $\mathcal{TR}_t = \{(\mathbf{x}, y_\mathbf{x})\}$ available at time t called the *training set*. The credit scoring model is *calibrated* [9] if $P(y_\mathbf{x} = 1 | s(\mathbf{x}) = p) = p$, i.e., the score can be interpreted as the probability that the event will occur.

Quality metrics in credit scoring models can be classified based on the purpose [28], [29]. Metrics that evaluate the discriminative power to separate positives and negatives include AUC, Gini, and KS statistics. The Area Under the ROC Curve (AUC) [30] is the probability (the higher, the better) that a randomly drawn positive instance $\mathbf{x}^+$ receives a higher score than a randomly drawn negative instance $\mathbf{x}^-$:

$$AUC = \mathbb{E}_{\mathbf{x}^- \sim \mathcal{D}^0_t, \mathbf{x}^+ \sim \mathcal{D}^1_t}[\mathbb{1}(s(\mathbf{x}^+) > s(\mathbf{x}^-))].$$

where $\mathcal{D}^1_t$ (resp., $\mathcal{D}^0_t$) is the distribution $\mathcal{D}_t$ conditional to positives (resp., negatives). The Gini coefficient (also known as Accuracy Ratio) is linearly related to the AUC as follows $Gini = 2 \cdot AUC - 1$. The Gini coefficient amounts to the fraction of the difference in power between a perfect ranking (all positives scored higher than any negative) and a random ranking of scores. These and the other metrics are estimated on a hold-out dataset $\mathcal{TE}_t = \{(\mathbf{x}, y_\mathbf{x})\}$, called the *test set*, using the empirical counterparts of their definition at the population level. Metrics that evaluate the accuracy of the scores include BLS and Log-loss scores. The Brier Loss Score (BLS) [31] is the expected quadratic loss of scores:

$$BLS = \mathbb{E}[(s(\mathbf{x}) - y_\mathbf{x})^2]$$

A metric that evaluates the degree of calibration over a binning $b_1, \ldots, b_R$ of the score interval $[0, 1]$ is the Binary Expected Calibration Error (BIN-ECE) [9]:

$$BIN\text{-}ECE = \mathbb{E}[|P(y_\mathbf{x} = 1 | s(\mathbf{x}) \in b_i) - \mathbb{E}[s(\mathbf{x}) | s(\mathbf{x}) \in b_i]|]$$

This is estimated over the test set as the weighted absolute difference between the observed default rate OB_i in the bin

b_i (which estimates $P(y_{\mathbf{x}} = 1|s(\mathbf{x}) \in b_i)$) and the mean score in the bin b_i (which estimates $\mathbb{E}[s(\mathbf{x})|s(\mathbf{x}) \in b_i]$).

Finally, variants of those metrics have been proposed to take into account unbalanced distributions of positives and negatives. In particular, the Brier Skill Score [31] is defined as the error reduction (the higher, the better): $BSS = 1 - {}^{BLS}/_{BLS_{ref}}$ relatively to BLS_{ref}, which is the Brier Skill Score of a baseline classifier scoring any instance as the fraction of positives observed in the training set $\mathcal{TR}_t$.

Credit Rating. A credit rating model is a function $r_t : \mathcal{X}_t \rightarrow \{1, \ldots, R\}$, mapping instances to an ordered set of R rating classes. Typically R is in the range 5-10, with class 1 denoting low risk and class R denoting high risk. Each rating class i is assigned an apriori probability of default PD_i, increasing with i, for which the model is expected to be calibrated, namely $P(y_{\mathbf{x}} = 1|r_t(\mathbf{x}) = i) = PD_i$. A variant of BIN-ECE quantifies the calibration error of rating models:

$$\textit{BIN-ECE-Rating} = \mathbb{E}[|P(y_{\mathbf{x}} = 1|r(\mathbf{x}) = i) - PD_i|]$$

where the observed default rate is compared to the expected one, rather than the average score. BIN-ECE-Rating is estimated over the test set as the weighted mean value of $|OB_i - PD_i|$, where OB_i is the observed default rate for instances in the test set rated in class i. Moreover, the observed rate OB_i is expected to be lower or equal than the predicted rate PD_i – otherwise, the model underestimates the actual risk. A binomial test of the hypothesis $H_0 : P(y_{\mathbf{x}} = 1|r(\mathbf{x}) = i) \leq PD_i$ is adopted at some confidence level for such a purpose. Another statistical test based on the multinomial distribution is the Extended Traffic Light [32], which results in an immediately grasped colour (green, yellow, orange, red).

IV. The Leasing Scenario

A lease contract conveys the right to use an asset for a period of time in exchange for payment [33]. The leasing company, called the *lessor*, owns the asset. The asset user, called the *lessee*, can be a company or a natural person. The asset is bought by the lessor from a *vendor* based on the lessee's requirements. The payments of the contract include an initial payment, periodic installments, and an optional redemption price. Lease financing has many advantages. The asset is immediately available for the lessee, while its cost is split over the contract duration. The direct acquisition of expensive equipment, plant and machinery requires, instead, capital outlay or securities to access credit financing. For the lessor, the asset can be repossessed when the lessee defaults on payments; hence the lessor's interest is fully secured. These and other advantages have been driving a continuous growth of the leasing market in the last decade, with an estimated volume of ≈\$1,675B worldwide in 2023 (from ≈\$1,520B in 2022). The European market is estimated ≈€415B in 2022.

Assilea (https://www.assilea.it) is the Italian leasing association, with more than 70 associated companies, representing more than one half of the Italian market in 2021 (≈\$18B in value). Each associate runs its own business, possibly in competition with other associates, adhering to its own credit risk policies and procedures regarding target markets, type of assets, region of operation, appetite for risk, etc. More than 40 of the associated companies contribute monthly to Assilea raw data on newly signed contracts and on the payment status of active contracts. These data are integrated into a data warehouse of past and active lease contracts. Assilea provides back to the contributing associates a credit information service called BDCR ("Banca Dati Centrale Rischi del Leasing" in Italian). Information provided aggregates data from the various associates to summarize the number and status of a prospective lessee's past and active lease contracts. The benefit for the associates in contributing to the BDCR consists of more in-depth information on the past history and the current financial commitments of prospect lessees leading to a better evaluation of their creditworthiness. The contribution to the BDCR and its usage is ruled by a self-regulation code signed by the associates. The code is compliant to the European General Data Protection Regulation [34]. Prospect lessees can submit complaints about data processing by the BDCR to a national authority which monitors credit information systems.

V. The `LSP` Framework

The `LSP` project has followed the de-facto industry standard of the CRISP-DM process model [35]. Business objectives have been determined by involving a few representative associate companies. Users with different roles (risk managers, risk analysts, credit analysts, data analysts, head of organization and control, head of legal) have been interviewed several times at different project stages, both in person and through a questionnaire. Various definitions of default payments have been discussed with the users. The one adopted considers the occurrence of severe insolvency or legal default within a lookahead period of 12 months from the signing date of the contract. Severe insolvency is triggered in case of overdue payments for a total of 5% of the contract or a total of 35% of the payments due in the next semester. A legal default, causing contract termination, is triggered in case of overdue payments for at least six monthly installments for real estate assets or four monthly installments for other assets, even if not consecutive, or equivalent amount [36].

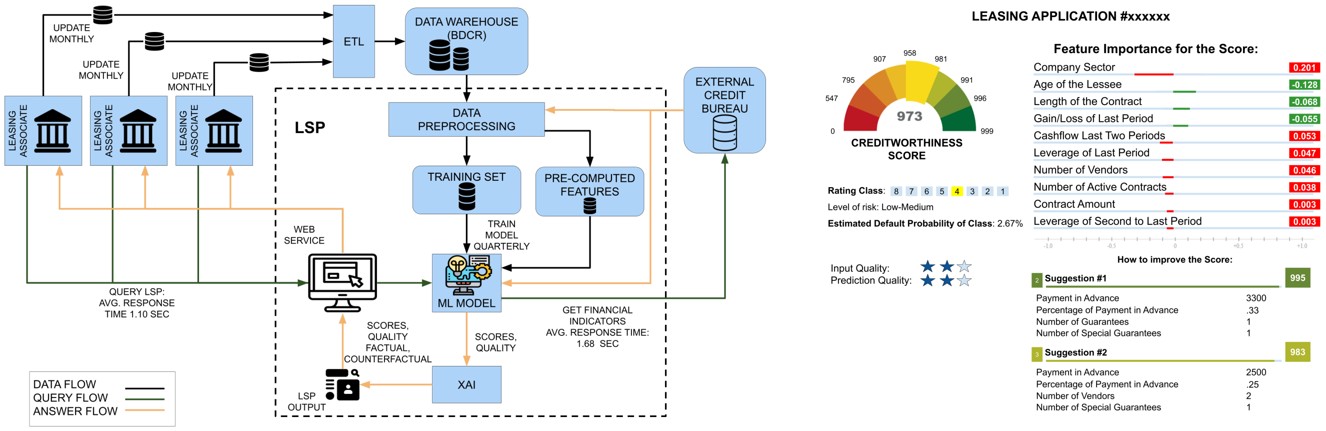
In the following, we describe the architecture and workflow of the `LSP` framework and detail the design rationale of the critical and novel components of the `LSP` system, also w.r.t. the objectives stated in the introduction. The components of the framework have been implemented in Python (`LSP`), Sadas Engine[2] (ETL, data warehouse), and Java (web service).

A. Architecture, workflow, inputs and outputs

Architecture and workflow. Figure 1a summarizes the architecture of the `LSP` system and the workflow of data and information from and towards the leasing associate companies. Each associate monthly contributes raw data to the BDCR credit information system, which stores in a centralized data

[2] A big data columnar database, see https://www.sadasdb.com/.

Fig. 1: The LSP system.



(a) LSP workflow.

(b) An example of LSP output.

LEASING APPLICATION #xxxxxx
0
547
795
907
958
981
991
996
999
973
CREDITWORTHINESS
SCORE
Rating Class: 8 7 6 5 4 3 2 1
Level of risk: Low-Medium
Estimated Default Probability of Class: 2.67%
Input Quality:
Prediction Quality:
Feature Importance for the Score:
Company Sector 0.201
Age of the Lessee -0.128
Length of the Contract -0.068
Gain/Loss of Last Period -0.055
Cashflow Last Two Periods 0.053
Leverage of Last Period 0.047
Number of Vendors 0.046
Number of Active Contracts 0.038
Contract Amount 0.003
Leverage of Second to Last Period 0.003
-1.0
-0.5
0
+0.5
+1.0
How to improve the Score:
2
Suggestion #1
995
Payment in Advance 3300
Percentage of Payment in Advance .33
Number of Guarantees 1
Number of Special Guarantees 1
3
Suggestion #2
983
Payment in Advance 2500
Percentage of Payment in Advance .25
Number of Vendors 2
Number of Special Guarantees 1

warehouse the data on new contracts and the status of payments of active contracts. Data preprocessing outputs a training set at the granularity of one instance per contract. Financial indicator features are collected through an external credit bureau service both in bulk (offline) and on-the-fly at the time of predictions. This is intended to answer the *Objective O2*. The credit scoring model is trained quarterly from the training set. Users at a leasing associate submit queries to the LSP system through a web interface. A query includes the prospect contractual information, including the (id of the) lessee, the vendor(s), the guarantor (s), the amount of the contract, its duration, the initial payment amount, and the redemption price. These data are joined with financial indicators and other pre-computed features, mainly behavioural indicators, not only about the lessee, but also about the vendors, and the guarantors – since good vendors tend to bring good lessees. The joined features values are passed in input to the scoring model, which produces a creditworthiness score, a rating class on an 8-scale, and two quality labels: one about the input feature values, and one about the output score. The quality labels are intended to answer the *Objective O3*. An eXplainability in AI (XAI) module adds factual and counterfactual explanations of the predicted score. The (counter)factual explanations are intended to answer the *Objective O4*. The outputs are arranged and sent back to the user by the web service. An example output is shown in Figure 1b. The average time for serving a user query is 1.1 seconds. The most time-consuming task is the invocation of the external credit bureau service (1.68 seconds on average), which, however, takes place only if the user query regards a lessee that is a company with some specific legal form, and such that there are no updated financial indicators cached about. These computational performances allow for interactively querying the LSP, e.g., by changing some value of the contractual data to re-evaluate the score and its explanation in a what-if scenario *(Objective O5)*. Also, the performances are optimized for scaling to a potential number of 600K calls per year, which is currently the number of calls to the BDCR credit information system from all associates.

Input features. Data preprocessing is run monthly after updating the BDCR data warehouse. Predictive features and a target feature are computed for each contract, with information available at the time of the contract starting date. These features are about:

- *the contract* including duration, total, initial payment, redemption price, number of installments, fraction of initial payment, fraction of redemption price, type of asset, state of the asset (e.g., used, new, to build);
- *the lessee,* including age of business (or age of natural persons), region/province of company headquarters (residence for natural persons), industry sector classification, legal form, financial indicators about the last and the second to last accounting closing year available;
- *the vendor(s),* including region/province of company headquarters, industry sector classification, legal form;
- *the guarantor(s),* including type of guarantee (e.g., surety or obligation to buy), amount guaranteed, legal form;
- *the current commitments,* including number/volume of active contracts by the lessee, by the vendor(s), by the guarantor(s);
- *the behavioural indicators,* including number/volume of contracts concluded by the lessee, by the vendor(s), by the guarantor(s), and number/volume of defaults for those contracts in the previous six months, 12 months, 24 months, ten years;
- *the target feature,* which is 1 (positives) if the default of the contract has been observed during the lookahead period, 0 (negatives) if no default has been observed in the lookahead period, and '*unknown*' if the lookahead period has not passed yet, but no default has been observed so far.

LSP outputs. Figure 1b shows an example of the output returned to the user. The following items are included:

Algorithm 1: LSP.fit()

```
Input  : $(\mathbf{X}, \mathbf{y})$ - training set,
         $H$ - LightGBM classifier,
         $k$ - number of folds
         $p_t$ - estimated positive rate in the test set
Output: $(h, g)$ - selective classifier
1  $S \leftarrow$ StratifiedKFold($(\mathbf{X}, \mathbf{y}), k$) // stratified $k$-fold partition
2  for $\mathbf{X}_i, \mathbf{y}_i k \in S$ do                                // for each fold
3      $(\mathbf{X}'_i, \mathbf{y}'_i) = (\mathbf{X} - \mathbf{X}_i, \mathbf{y} - \mathbf{y}_i)$          // training data
4      $h_i \leftarrow H.fit(\mathbf{X}'_i, \mathbf{y}'_i)$               // train $i$-th classifier
5      $\mathbf{s}_i \leftarrow h_i.score(\mathbf{X}_i)$                    // score test data
6  $\mathbf{s} \leftarrow \cup_{i=1}^{k} s_i$                                // store all scores
7  $\theta_{l,.9}, \theta_{u,.9} \leftarrow EstimateThetaAUC(s, .9)$  // bounds for $c = .9$
8  $\theta_{l,.5}, \theta_{u,.5} \leftarrow EstimateThetaAUC(s, .5)$  // bounds for $c = .5$
9  $h.score \leftarrow$ lambda $\mathbf{x}$ : $\frac{1}{k}\sum_{i=1}^{k} h_i.score(\mathbf{x})$  //  score function
10 $\gamma = mean(\mathbf{y})/(1 - mean(\mathbf{y}))/p_t * (1 - p_t)$       // odds factor
11 $h.score \leftarrow \frac{h.score}{h.score+\gamma(1-h.score)}$          // Bayesian correction
12 $g \leftarrow$ lambda $\mathbf{x}$ : $\begin{cases} * & \text{if } \theta_{l,.9} \leq h.score(\mathbf{x}) \leq \theta_{u,.9} \\ ** & \text{if } \theta_{l,.5} \leq h.score(\mathbf{x}) \leq \theta_{u,.5} \\ *** & \text{otherwise} \end{cases}$
     // prediction quality
13 return $(h, g)$
```

- *a creditworthiness score*, predicting the good outcome of the contract obligations w.r.t. the default notion. The creditworthiness score is in a three digit range 0-999, and it is calculated as $(1 - r) \cdot 999$ where r is the risk score of the LSP model;
- *a rating class*, in the range 1 to 8, where 1 is very low risk and 8 very high risk. Class boundaries are shown in the rainbow around the creditworthiness score, and the rating class assigned is larger than the others. Also, the rating class is shown as a number and with a textual description. On mouse-over the rainbow, the textual description of any class can be viewed. The estimated PD of the assigned rating class is also reported;
- *two quality labels*, on a 1 to 3-stars scale, referring to the quality of input and to a self-assessed quality of the predicted score.
- *a factual explanation*, of the score, using a feature importance plot, where for the top 10 features, the positive or negative contribution to the creditworthiness score is reported both as a bar and a value, in colour (green for positive, red for negative);
- *up to three counterfactual explanations*, consisting of changes in the features of the contract that lead to better creditworthiness score and, possibly, to a better rating class.

B. The scoring model

The core of the LSP system, shown in Algorithm 1, is a selective classifier with Bayesian correction of the prediction score. Its primary input $(\mathbf{X}, \mathbf{y})$ consists of the instances of the training set where the target value is known, i.e., 0 or 1. Instances with '*unknown*' target value will be considered later on in Section V-D.

Selective classification for uncertainty estimation. Objective O3 requires the uncertainty self-assessment of the risk score. We adopt a novel approach which relies on selective classification. Selective classification (or classification with a reject option) [37] pairs a classifier with a selection function to determine whether a prediction should be accepted or the classifier should abstain. The selection function assesses the trustworthiness/uncertainty of a prediction. Our selective classifier is a variant of the model-agnostic AUCROSS approach [38]. Algorithm 1 splits the training set into k stratified folds (line 1) – we set $k = 5$ in our implementation. For each fold i, a base classifier (h_i) is trained on the data of the other $k - 1$ folds (line 4). We use LightGBM[3] [41] as the base classification algorithm. The scores of h_i on the data of the i-th fold (line 5) are accumulated in s (line 6). The set s of predicted scores allows for estimating bounds θ_l, θ_u such that scores outside $[\theta_l, \theta_u]$ amount at a specified coverage, i.e., percentage of predictions (parameter c in lines 7 and 8), and such that they maximize the AUC of the predictions [38]. Scores inside the bounds $[\theta_l, \theta_u]$ are rejected (at coverage c), since predicting on such instances would lower the final AUC. Algorithm 1 computes two pairs of bounds, one at coverage $c = .9$ and one at $c = .5$. The output function g determines the quality/uncertainty of the output scores, ranking them into three levels: three stars for the top 50% of the predictions, two stars for the second top 40%, and 10% for the remaining bottom predictions w.r.t. AUC maximization. The scoring function of Algorithm 1 is obtained by averaging the scores of the k classifiers (line 9). An alternative would have been to train a final classifier on the whole training set, as in the AUCROSS approach [38]. However, the model's score averaging was beneficial w.r.t. calibration of the scores, also in comparison with using state-of-the-art calibration techniques.

Bayesian correction for prior data shifts. Objective O6 requires to address prior data shifts. A Bayesian correction of the scores is performed (line 11) for coping with the change in the proportion of positives from training to test data [23]. This correction takes into input a factor γ, which is the odds ratio of positive rate between the training and test population, i.e., $\gamma \approx (P_S(y = 1)/P_S(y = 0))/(P_T(y = 1)/P_T(y = 0))$ where P_S and P_T are the distributions over the training/source and test/target populations. $P_S(y = 1)$ is estimated on the training set. $P_T(y = 1)$ is estimated by p_t, which is an input parameter. The correction is suitable for coping with events potentially predictable in advance, such as those occurring after a policy change. The correction scales the output scores. Hence, it does not affect ranking metrics such as AUC, but it does affect accuracy scores, such as BLS and BSS, and the calibration of scores and ratings. We used the correction twice. First, the financial crisis of 2008 led to a long period of recession and stagnation in Italy, with an increasing fraction of non-performing loans (NPL). The peak was reached in 2014. Afterwards, the NPL fraction decreased over time for

[3] LightGBM adopts a gradient-boosting approach with state-of-the-art performances over tabular datasets, a fast parallel implementation, API's for a few programming languages, management of categorical attributes and missing values without encoding/imputation, and the calculation of TreeSHAP values used by LSP for factual explanations. We have experimented with other base classifiers, including *scikit-learn* models, XGBoost [39] and CatBoost [40]. CatBoost was the only classifier offering features comparable to LightGBM without statistically significant performance differences. We finally chose LightGBM as it was the fastest method.

all businesses, including leasing companies. Yearly forecasts of the Bank of Italy[4], and historical data from the BDCR data warehouse (see Figure 3 top), anticipated decreasing positive rates p_t. This information has been used in the backtesting of the LSP system to correct the prediction scores, setting $p_t = 0.02$ up to the first quarter of 2020. Second, in the first quarter of 2020, the government counter-measures against the COVID-19 outbreak included a suspension of loan repayments. Consequently, the positive rate dropped substantially (see Figure 3 top). As a forward-looking strategy for LSP, we set $p_t = 0.01$ for the suspension period.

C. Explanation methods

LSP provides post-hoc explanations of the scores, both in factual and in counterfactual terms, to address *Objective O4*.

Factual explanations. They are provided in terms of relative feature relevance w.r.t. the model's score. We rely on Shapley values [15], which are computed very fastly by the LightGBM implementation of the exact TreeSHAP algorithm [42]. Shapley values are a coalition game theory concept that aims to allocate the score generated by a coalition of features to each of the features. Shapley values are additive, hence the Shapley values of the LSP model, which averages the scores of k classifiers (line 9 in Algorithm 1) is the mean of the Shapley values of the k classifiers. TreeSHAP returns the Shapley values of the logit transformation of the risk score, i.e., the sum of Shapley values is the log odd of the risk score. By going back to the risk score (a.k.a., the probability space), we lose the additivity property, namely the sum of inverse log odds of Shapley values does not equal to the risk score probability. This may generate confusion in the reading by the user. We then choose to present to the user the *relative contribution* of each feature to the log odd of the risk score, calculated as the L1 normalized Shapley values. Normalized negative (resp., positive) values correspond to negative (resp., positive) contribution in risk, compared to the mean score. They are shown in red (resp., green) in Figure 1b).

Counterfactual explanations. They consist of up to three examples in which changes in the contract features would result in a lower risk score (or higher creditworthiness score), possibly improving the rating class. Changes should be made only on certain actionable features, such as the initial payment amount, the duration of the contract, and the type of guarantee. Also, changes may be subject to some constraints, such as the range of possible values or the minimum amount of increase/decrease. These and several other requirements on counterfactual generation (minimal number of changes, plausible instances, diversity among counterfactuals, stability of results) are rarely dealt with by a single tool. We designed a novel ensemble of base counterfactual explainers, each one contributing several counterfactual instances. We filter (or modify) those not satisfying requirements not directly implemented by the tools, and from the collection of all remaining counterfactual instances, we finally select three instances (or less, if not possible) to return by following the diversity-maximization approach of [43]. Two types of base counterfactual explainers have been considered – the choice being driven by computational efficiency requirements. The first type is a brute force approach, generating all possible changes in actionable features (restricted to changing max two features, and considering a 10 equal-width binning of continuous features). The first type takes into account similarity and minimality. The second type projects instances into a lower-dimensionality space using PCA. Then it applies a random perturbation at that space, and converts back the perturbed instances to the original space, filtering only instances that result to be counterfactuals (lower risk score by the LSP model). Hence, it takes into account diversity. The counterfactual ensemble explainer runs four base classifiers of type one, three of type two using PCA for explaining 99% of variance, and three of type two using PCA for explaining 75% of variance. A total of 256 counterfactuals are produced, then filtered, and then 3 of them are finally selected. Compared to [43], the elapsed time for generating three counterfactuals lowered from 8 seconds to 0.5 seconds on average.

D. Miscellaneous

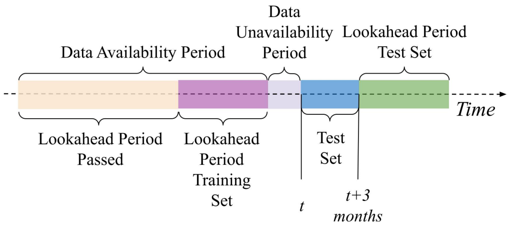
In this subsection, due to space restriction, we briefly outline a number of tasks and topics tackled by the LSP project.

The credit rating model. A standard approach to obtain a rating model from a credit scoring model is binning the $[0, 1]$ range of scores and then determining a PD for each bin. For example, [11] considers 9 rating classes, and it adopts a genetic algorithm for optimizing the choice of the class PDs w.r.t. BLS. LSP adopts a logarithmic binning of the $[0, 1]$ interval to adapt on the unbalanced distribution of defaults. The advantage of our choice is that the binning is data-independent. Hence it is not affected by random/small perturbations of the data distribution. The drawback is that, for the same reason, we need to closely monitor the binning quality. We rely on both binomial tests and the extended traffic light approach (see Section III) as well as on monitoring the stability of the model [44], [45].

Feature selection. The most relevant feature selection task was concerned with financial indicators [46]. The associated companies do not contribute such indicators, even if they might be available for decision-making in some cases. However, they are expensive to obtain, as the fees paid to the credit bureau service are proportional to the number of indicators requested. We collected more than 20 financial indicators for a sample of companies subject to mandatory submission of balance sheets. We selected six indicators by a variance inflation factor (VIF) stepwise variable elimination [47]. The selected indicators cover business performance (profit), efficiency in revenue generation (return on assets, and total assets turnover), indebtedness (current liabilities/total assets, and leverage), and liquidity (cashflow). Experiments showed a very narrow loss in performance when using the selected set of

[4] https://www.abi.it/studi/outlook-crediti-deteriorati/

Fig. 2: Backtesting framework.


Data Availability Period
Data Unavailability Period
Lookahead Period Test Set
Time
Lookahead Period Passed
Lookahead Period Training Set
Test Set
t
t+3 months

indicators w.r.t. the entire set. Another set of external features was considered, regarding market conditions (interest rates, inflation, GDP, public debt, etc.) and company demography (new/closed companies per region/type of business). However, these did not improve the performances of the scoring model.

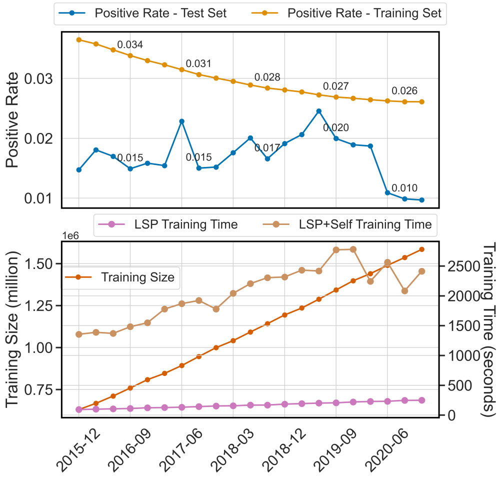
Missing values and quality of input. The legal nature of the data contributed to the data warehouse (contracts, payments, etc.) ensures a certain quality of the training set, with a few issues already solved by ETL (repeated entries, outliers, impossible values). Other issues may persist, however, e.g., due to missing contributions for technical reasons, for which the status of a contract could be missing in certain months. Financial indicators may also be missing since only certain company types have the legal obligation to submit balance sheets. Finally, a user query may not include all prospect contract features since only a few are mandatory. LightGBM natively deals with missing values in the training set and in the input instance. No imputation method was selected, yet a few were experimented with. LSP addresses *Objective O3* as per quality of the input by summarizing the impact of missing values in the input instance by a 1-star to 3-stars indicator (see Figure 1b). A missing value is weighted by the feature importance, calculated by the mean decrease in AUC when testing the base classifier h_i (line 5 in Algorithm 1) with such a feature set to missing (and all features calculated from that one, e.g., ratios, also set to missing).

Self-learning. On top of Algorithm 1, we experimented with a self-training approach close to the one described in [48], exploiting the instances with target feature value '*unknown*'. Such instances regard contracts started since less than 12 months for which the default event has not been observed. We iteratively re-train LSP by adding the '*unknown*' instances that are predicted with high confidence and assigning to them the predicted label (0 or 1). Such a procedure is computationally expensive. It resulted into a better calibration, but comparable with the impact of the Bayesian correction. Intuitively, the '*unknown*' instances were mostly predicted to be negatives, thus reducing the positive rate in the augmented training set.

VI. Evaluation

We experiment with a dataset of 2M leasing contracts with about 500 features (about 90 before one-hot encoding) ranging from 2011-01 (Jan 2021) to 2022-04 (Apr 2022). Let us first describe the backtesting framework and then present some performance results of LSP.

Fig. 3: Positive rate, training size and time.



Backtesting framework. Backtesting aims at estimating the performances of a system by resampling historical data. We adopt an out-of-time testing procedure with rolling-origin [49], also known as walk-forward, forward-chaining, or nested cross-validation. In fact, standard cross-validation exhibits a positive bias error for time-related data [50]. Figure 2 shows the situation pretending to be at a time point t at which a new credit score model s_t is available (see Section III for notation). Such a model will be used to score instances (the test set $\mathcal{TE}_t$) for a specific test period, which, for LSP, is set to 3 months. After such a period, a new model will be available, and the time point t will roll forward by three months. As shown in Figure 1a, LSP re-trains a new model every quarter. For an instance $\mathbf{x}$, the true default outcome $y_\mathbf{x}$ is defined w.r.t. a lookahead period of 12 months. In order to evaluate the model performances, we need data from the lookahead period following $t+3$ months. Moreover, since the BDCR data warehouse is updated monthly, and updates regard the status of contracts at the previous month, we consider that data is actually available with a delay of two months. Therefore, the test periods in our experiments stop at $t =$ 2020-12. The last test period ranges from 2020-12 to 2021-02, the lookahead period for its third month (2021-02) ends on 2022-02, and the data regarding payments on 2022-02 are available in the data warehouse on 2022-04, which is our last month of raw data. Let us consider now which instance can be included in the training set. Reasoning as before, data on the two months prior to t should be considered unavailable at the time t. This is shown as *Data Unavailability Period* in Figure 2. Data in the availability period can be considered. Such instances $\mathbf{x}$ may have the target feature $y_\mathbf{x}$ known (1 for default, 0 for not default), or set to '*unknown*' (see Section V-A). The former instances are included in the training set $\mathcal{TR}_t$. The latter are included in the dataset for self-learning (see Section V-D). All instances for which the lookahead period has passed (shown as *Lookahead Period Passed* in Figure 2) are in the training set, which also includes instances for which the lookahead period has not passed yet, but the default event already occurred.

Fig. 4: Performance comparison: `LSP` vs `LSP-Cor` vs `LSP-Cor+Beta` vs `LGBM`.

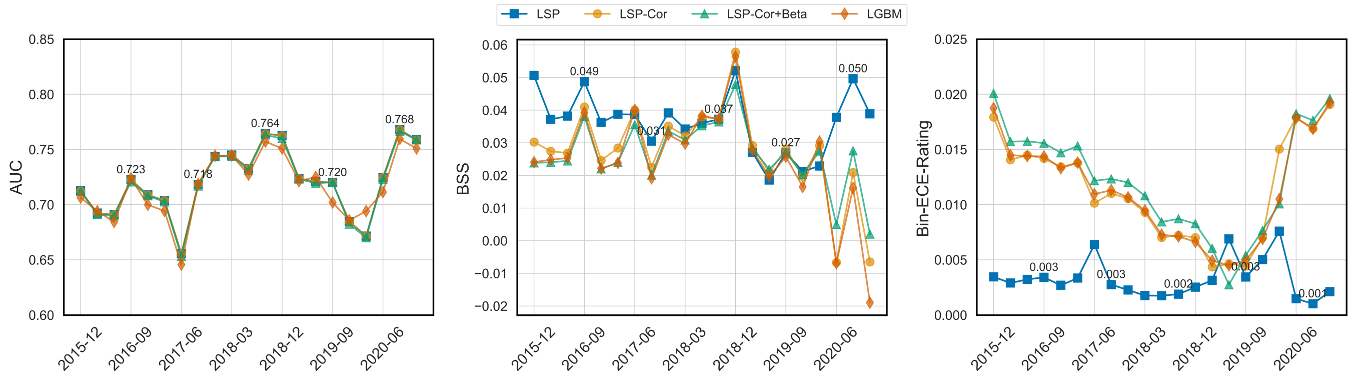


Fig. 5: Calibration plot for a single test period.

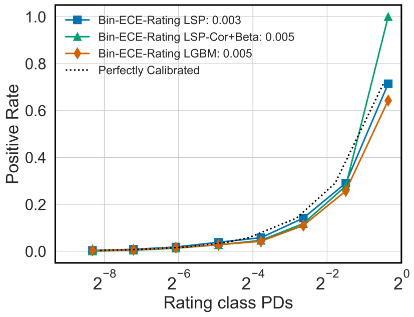


Fig. 6: Performance comparison: subsets of test data by prediction quality.

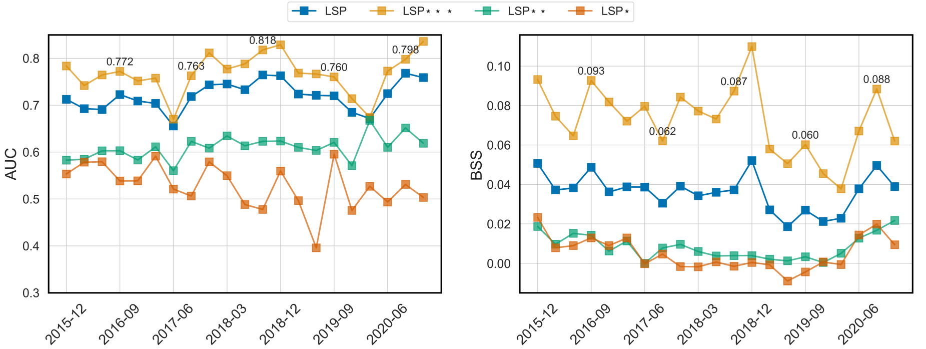
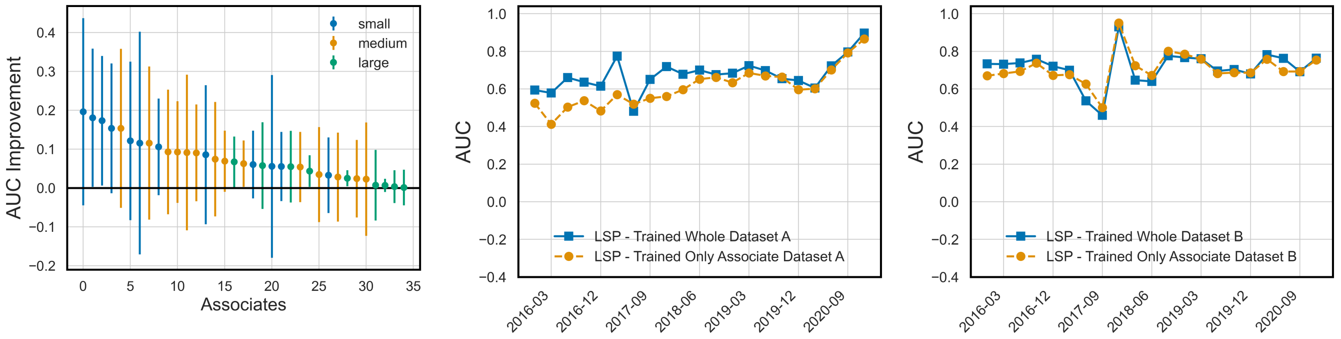


Fig. 7: `LSP` trained on all data w.r.t. `LSP` trained a single associate data.


(a) AUC improvement (mean ± stdev).

(b) Associate company A.

(c) Associate company B.

Hyper-parameter tuning. An extensive experimental phase of the project considered the tuning of the hyper-parameters of `LSP` and its base classifier LightGBM. In Algorithm 1, we fixed $k = 5$ as a trade-off between predictive performance and elapsed running time (see [38]). The quality bounds (lines 7, 8) were set by expert guidance. The hyper-parameters of LightGBM were initially tuned at each model re-training by relying on the Optuna framework (https://optuna.readthedocs.io/). During backtesting, however, we observed stability in the range of hyper-parameters chosen, for which we fixed these hyper-parameters for models trained after 2016. Model monitoring activities re-evaluate periodically such a choice.

Experimental results. Figure 3 (bottom) shows the size of training sets over rolling time t, and the elapsed training times of `LSP` and of the `LSP` with the time-consuming self-learning option (`LSP+Self`). Both training set sizes and elapsed times grow linearly over time. Training set reaches more than 1.5M instances in the last period. The training time of `LSP` reaches a maximum of about 300 seconds, while the one of `LSP+Self` is an order of magnitude larger. The tests were performed on a machine with 18 cores, 36 threads, equipped with Intel(R) Core (TM) i9-10980XE CPU @ 3.00GHz, OS Ubuntu 20.04.3, programming language Python 3.8.12. In the experiments, we compare `LSP` to an ablation version without Bayes correction (`LSP-Cor`), to a version without Bayes correction but with Beta calibration [51] of scores (`LSP-Cor+Beta`), and to a baseline consisting of a single LightGBM classifier with no Bayes correction nor calibration (`LGBM`). Such a baseline approximates the related credit scoring system in Table I (but with the same hyperparameters as LSP). The first experiment is intended to highlight the contribution of the Bayes correction and the contribution of the "ensemble of ensembles" strategy of Algorithm 1. We follow the approach in [52] to test whether differences across classifiers are statistically significant at .01 significance level. Figure 4 displays the results for the AUC, BSS and Bin-Ece-Rating metrics. Regarding AUC, LSP performs slightly better than LGBM (mean ± stdev of $.72 \pm .03$ vs $.716 \pm .029$, not statistically significant), and ties with LSP-Cor. The latter is expected, as Bayesian correction is a monotonic transformation of the scores, which does not affect ranking metrics. The plots for BSS and Bin-Ece-Rating reveal, instead, the statistically significant advantage of using Bayesian correction over the other approaches – especially for calibration during the COVID-19 period starting in 2020-06. Figure 5 contrasts the calibration of LSP, LGBM, and LSP-Cor+Beta on a single test period t – the one with the lowest Bin-ECE-Rating gap between the models. It shows that LSP achieves calibration without post-processing methods.

The second experiment tests the uncertainty self-assessment performances of LSP. Figure 6 shows that the novel approach based on selective classification can separate instances with very accurate predictions (3 stars quality prediction, shown as $\text{LSP}^{\star\star\star}$), from instances with medium accuracy (2 stars, shown as $\text{LSP}^{\star\star}$), and from instances with low accuracy (1 star, shown as $\text{LSP}^{\star}$). The differences are statistically significant. Notice that LGBM is not natively a selective classifier. Hence, it cannot provide a quality prediction. Moreover, [38] shows that the strategy of Algorithm 1 outperforms existing methods also in terms of coverage guarantees.

The third experiment tests the improvement in performance for an associate to use the LSP model trained on all data against using the LSP model trained on its own data only *(Objective O1)*. Figure 7a shows the mean improvement (± stdev) over the rolling test periods. Only associates with data in at least 10 test periods are considered. The improvement is negatively correlated with the size of the associate. We categorize small (l<10K contracts in total), medium (10K to 50K) and large (>50K contracts) associates. Such a correlation is not surprising, as large associates operate across the whole country; hence their contracts are representative of the market. Small associates, instead, have a regional scope, and the model built on their instances only is less general. On average, small associates benefit from an absolute improvement in AUC of 11%, which reduces at 7% for medium and 3% for large associates. Figures 7b-7c show the detailed comparison over rolling test periods for two associates. Associate A is a large one, and the plot shows a continuous, yet decreasing, improvement over time. The decrease is due to accumulating a sufficiently large number of instances over time. Associate B is also large, but the improvement is instead fluctuating.

VII. Conclusions

The LSP project has faced several challenging objectives. *Objective O1* was achieved as a by-product of sharing data from several associate leasing companies. The increased performances w.r.t. using only their own data was marked for small and medium companies, and moderate for large companies. *Objective O2* required a complex workflow to collect in real-time (and to cache) the missing financial indicators from an external credit bureau, as well as a feature selection of those to be collected. *Objective O3* led to a novel usage of selective classification for uncertainty estimation of risk score by distinguishing top 50%, next 40%, and bottom 10% of the predictions w.r.t. AUC maximization. *Objective O4* was achieved by standard Shapley values, and a novel counterfactual ensemble algorithm aimed at producing results very fastly, also in light of *Objective O5* requiring interaction and what-if analysis. The average elapsed time for serving a user query is 1.1 seconds. Finally, regarding *Objective O6*, prior data shift was tackled by a Bayesian correction. The critical choice of the correction parameter is supported by default forecasts provided by the Bank of Italy.

The LSP system has been designed to be integrated into associate leasing companies' information and decision processes. Input data are collected from the existing BDCR data warehouse. Interaction with the user occurs at the same web interface already offering statistics and analytical reports based on the BDCR. Such a design choice is intended to reduce the information gap and the users' resistance to adopting the system in daily operations. Explanations of the output scores and uncertainty self-assessment aim at establishing trust towards the system beyond the backtesting evidence.

Finally, we mention two issues open for future work. *First,* data contributed to the BDCR pertain approved applications whose contract has actually started. Rejected applications are not contributed. If they were, we could have approached reject inference methods [53] in a similar way as we dealt with self-learning. *Second,* a pressing urge is to control for bias and unfairness in data-driven AI [54] to prevent discriminatory decisions against protected-by-law social groups. Leasing contracts in the BDCR regard mostly companies, with only 3% of the contracts having a natural person lessees (in our data, they were also indistinguishable from sole proprietorship/individual businesses). While EU anti-discrimination laws apply to natural persons only [34], other biases and forms of ethically unacceptable decisions may indirectly be generated through proxy features that relate social groups to e.g., type of business or region of operation.

Acknowledgements

We are grateful to Assilea and to its associates for their support and active contribution in all the phases of the design, development, and validation of the LSP system.

References

[1] A. Markov, Z. Seleznyova, and V. Lapshin, "Credit scoring methods: Latest trends and points to consider," *The Journal of Finance and Data Science*, vol. 8, pp. 180–201, 2022.

[2] J. Breeden, "A survey of machine learning in credit risk," *Journal of Credit Risk*, vol. 17, no. 3, 2021.

[3] X. Dastile, T. Çelik, and M. Potsane, "Statistical and machine learning models in credit scoring: A systematic literature survey," *Appl. Soft Comput.*, vol. 91, p. 106263, 2020.

[4] D. Tripathi, A. K. Shukla, B. R. Reddy, G. S. Bopche, and D. Chandramohan, "Credit scoring models using ensemble learning and classification approaches: A comprehensive survey," *Wirel. Pers. Commun.*, vol. 123, no. 1, pp. 785–812, 2022.

[5] S. Lessmann, B. Baesens, H. Seow, and L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," *Eur. J. Oper. Res.*, vol. 247, no. 1, pp. 124–136, 2015.

[6] V. Moscato, A. Picariello, and G. Sperlì, "A benchmark of machine learning approaches for credit score prediction," *Expert Syst. Appl.*, vol. 165, p. 113986, 2021.

[7] J. Abellán and J. G. Castellano, "A comparative study on base classifiers in ensemble methods for credit scoring," *Expert Syst. Appl.*, vol. 73, pp. 1–10, 2017.

[8] D. Tasche, "The art of probability-of-default curve calibration," *Journal of Credit Risk*, vol. 9, no. 4, pp. 63–103, 2013.

[9] T. S. Filho, H. Song, M. Perelló-Nieto, R. Santos-Rodríguez, M. Kull, and P. A. Flach, "Classifier calibration: a survey on how to assess and improve predicted class probabilities," *Mach. Learn.*, vol. 112, no. 9, pp. 3211–3260, 2023.

[10] X. He *et al.*, "Practical lessons from predicting clicks on ads at Facebook," in *ADKDD@KDD*. ACM, 2014, pp. 5:1–5:9.

[11] A. R. Provenzano, D. Trifirò, A. Datteo, L. Giada, N. Jean, A. Riciputi, G. L. Pera, M. Spadaccino, L. Massaron, and C. Nordio, "Machine learning approach for credit scoring," *arXiv:2008.01687*, 2020.

[12] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 93:1–93:42, 2019.

[13] M. Bücker, G. Szepannek, A. Gosiewska, and P. Biecek, "Transparency, auditability, and explainability of machine learning models in credit scoring," *J. Oper. Res. Soc.*, vol. 73, no. 1, pp. 70–90, 2022.

[14] P. Bracke, A. Datta, C. Jung, and S. Sen, "Machine learning explainability in finance: an application to default risk analysis," 2019, Bank of England Working Paper.

[15] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *NIPS*, 2017, pp. 4765–4774.

[16] N. Bussmann, P. Giudici, D. Marinelli, and J. Papenbrock, "Explainable machine learning in credit risk management," *Computational Economics*, vol. 57, no. 1, pp. 203–216, 2021.

[17] N. L. Torrent, G. Visani, and E. Bagli, "PSD2 explainable AI model for credit scoring," *CoRR*, vol. abs/2011.10367, 2020.

[18] R. Guidotti, "Counterfactual explanations and how to find them: literature review and benchmarking," *Data Mining and Knowledge Discovery*, pp. 1–55, 2022.

[19] R. McGrath, L. Costabello, C. L. Van, P. Sweeney, F. Kamiab, Z. Shen, and F. Lécué, "Interpretable credit application predictions with counterfactual explanations," *CoRR*, vol. abs/1811.05245, 2018.

[20] A. C. Bueff, M. Cytrynski, R. Calabrese, M. Jones, J. Roberts, J. Moore, and I. Brown, "Machine learning interpretability for a stress scenario generation in credit scoring based on counterfactuals," *Expert Syst. Appl.*, vol. 202, p. 117271, 2022.

[21] X. Dastile, T. Çelik, and H. Vandierendonck, "Model-agnostic counterfactual explanations in credit scoring," *IEEE Access*, vol. 10, pp. 69 543–69 554, 2022.

[22] J. Liu, C. Zhong, B. Li, M. I. Seltzer, and C. Rudin, "FasterRisk: Fast and accurate interpretable risk scores," in *NeurIPS*, 2022.

[23] J. Quiñonero-Candela, M. Sugiyama, N. D. Lawrence, and A. Schwaighofer, Eds., *Dataset shift in machine learning*. The MIT Press, 2009.

[24] G. I. Webb, R. Hyde, H. Cao, H. Nguyen, and F. Petitjean, "Characterizing concept drift," *Data Min. Knowl. Discov.*, vol. 30, no. 4, pp. 964–994, 2016.

[25] J. P. Barddal, L. Loezer, F. Enembreck, and R. Lanzuolo, "Lessons learned from data stream classification applied to credit scoring," *Expert Syst. Appl.*, vol. 162, p. 113899, 2020.

[26] W. Wang, C. Lesner, A. Ran, M. Rukonic, J. Xue, and E. Shiu, "Using small business banking data for explainable credit risk scoring," in *AAAI*. AAAI Press, 2020, pp. 13 396–13 401.

[27] L. Barbaglia, S. Manzan, and E. Tosetti, "Forecasting loan default in Europe with Machine Learning," *Journal of Financial Econometrics*, vol. 21, no. 2, pp. 569–596, 2021.

[28] M. Rezac and F. Rezac, "How to Measure the Quality of Credit Scoring Models," *Czech Journal of Economics and Finance*, vol. 61, no. 5, pp. 486–507, 2011.

[29] B. Engelmann, "Measures of a rating's discriminative power: Applications and limitations," in *The Basel II Risk Parameters*, 2nd ed., B. Engelmann and R. Rauhmeier, Eds. Springer, 2011, pp. 269–291.

[30] T. Yang and Y. Ying, "AUC maximization in the era of big data and AI: A survey," *ACM Comput. Surv.*, vol. 55, no. 8, pp. 172:1–172:37, 2023.

[31] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly Weather Review*, vol. 78, no. 1, pp. 1–3, 1950.

[32] D. Tasche, "A traffic lights approach to PD validation," *arXiv preprint cond-mat/0305038*, 2003.

[33] International Financial Reporting Standard, "IFRS 16 – Leases," 2016, https://www.ifrs.org/issued-standards/list-of-standards/ifrs-16-leases/.

[34] European Parliament and the Council, "Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," 2016.

[35] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying CRISP-DM process model," *Procedia Computer Science*, vol. 181, pp. 526–534, 2021.

[36] L. Bertino, "The standardisation of the leasing contract under Italian law and the Unidroit principles," *European Business Law Review*, vol. 30, no. 5, pp. 841–852, 2019.

[37] K. Hendrickx, L. Perini, D. V. der Plas, W. Meert, and J. Davis, "Machine learning with a reject option: A survey," *CoRR*, vol. abs/2107.11277, 2021.

[38] A. Pugnana and S. Ruggieri, "AUC-based selective classification," in *AISTATS*, vol. 206. PMLR, 2023, pp. 2494–2514.

[39] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *KDD*. ACM, 2016, pp. 785–794.

[40] L. O. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," in *NeurIPS*, 2018, pp. 6639–6649.

[41] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *NIPS*, 2017, pp. 3146–3154.

[42] S. M. Lundberg, G. G. Erion, H. Chen, A. J. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S. Lee, "From local explanations to global understanding with explainable AI for trees," *Nat. Mach. Intell.*, vol. 2, no. 1, pp. 56–67, 2020.

[43] R. Guidotti and S. Ruggieri, "Ensemble of counterfactual explainers," in *DS*, ser. Lecture Notes in Computer Science, vol. 12986. Springer, 2021, pp. 358–368.

[44] A. Ben Soussia, C. Labba, A. Roussanaly, and A. Boyer, "Time-dependent metrics to assess performance prediction systems," *The Int. J. of Inf. and Learning Tech.*, vol. 39, no. 5, pp. 451–465, 2022.

[45] G. Castermans, D. Martens, T. V. Gestel, B. Hamers, and B. Baesens, "An overview and framework for PD backtesting and benchmarking," *J. Oper. Res. Soc.*, vol. 61, no. 3, pp. 359–373, 2010.

[46] P. du Jardin, "The influence of variable selection methods on the accuracy of bankruptcy prediction models," *Bankers, Markets & Investors*, vol. 116, pp. 20–39, 2012.

[47] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, 8th ed. Springer, 2017.

[48] J. Liao, W. Wang, J. Xue, A. Lei, X. Han, and K. Lu, "Combating sampling bias: A self-training method in credit risk models," in *AAAI*. AAAI Press, 2022, pp. 12 566–12 572.

[49] R. M. Stein, "Benchmarking default prediction models: pitfalls and remedies in model validation," *Journal of Risk Model Validation*, vol. 1, p. 77–113, 2007.

[50] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," *BMC Bioinform.*, vol. 7, p. 91, 2006.

[51] M. Kull, T. S. Filho, and P. A. Flach, "Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers," in *AISTATS*, vol. 54. PMLR, 2017, pp. 623–631.

[52] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[53] A. Ehrhardt, C. Biernacki, V. Vandewalle, P. Heinrich, and S. Beben, "Reject inference methods in credit scoring," *Journal of Applied Statistics*, vol. 48, no. 13-15, pp. 2734–2754, 2021.

[54] E. Ntoutsi *et al.*, "Bias in data-driven artificial intelligence systems - an introductory survey," *WIREs Data Mining Knowl. Discov.*, vol. 10, no. 3, 2020.