mathematics

MDPI

Article

Evaluation of Cost-Sensitive Learning Models in Forecasting Business Failure of Capital Market Firms

Pejman Peykani [1,*], Moslem Peymany Foroushany [2], Cristina Tanasescu [3], Mostafa Sargolzaei [2] and Hamidreza Kamyabfar [2]

[1] Department of Industrial Engineering, Faculty of Engineering, Khatam University, Tehran 1991633357, Iran
[2] Department of Finance and Banking, Faculty of Management and Accounting, Allameh Tabataba'i University, Tehran 1489684511, Iran; m.peymany@atu.ac.ir (M.P.F.); mostafa.sargolzaei@atu.ac.ir (M.S.); h.kamyabfar@gmail.com (H.K.)
[3] Faculty of Economic Sciences, Lucian Blaga University of Sibiu, 550324 Sibiu, Romania; cristina.tanasescu@ulbsibiu.ro
* Correspondence: p.peykani@khatam.ac.ir or pejman.peykani@yahoo.com

Abstract: Classifying imbalanced data is a well-known challenge in machine learning. One of the fields inherently affected by imbalanced data is credit datasets in finance. In this study, to address this challenge, we employed one of the most recent methods developed for classifying imbalanced data, CorrOV-CSEn. In addition to the original CorrOV-CSEn approach, which uses AdaBoost as its base learning method, we also applied Multi-Layer Perceptron (MLP), random forest, gradient boosted trees, XGBoost, and CatBoost. Our dataset, sourced from the Iran capital market from 2015 to 2022, utilizes the more general and accurate term business failure instead of default. Model performance was evaluated using sensitivity, precision, and F1 score, while their overall performance was compared using the Friedman–Nemenyi test. The results indicate the high effectiveness of all models in identifying failing businesses (sensitivity), with CatBoost achieving a sensitivity of 0.909 on the test data. However, all models exhibited relatively low precision.

Keywords: business failure forecasting; imbalanced data; cost-sensitive learning; machine learning; Multi-Layer Perceptron (MLP); random forest; gradient boosted trees; XGBoost; CatBoost; AdaBoost

MSC: 62M20; 62P05; 62P20; 68T05; 68T10; 90C90; 91B28

Academic Editor: Raymond Lee

Received: 18 December 2024
Revised: 16 January 2025
Accepted: 21 January 2025
Published: 23 January 2025

Citation: Peykani, P.; Peymany Foroushany, M.; Tanasescu, C.; Sargolzaei, M.; Kamyabfar, H. Evaluation of Cost-Sensitive Learning Models in Forecasting Business Failure of Capital Market Firms. *Mathematics* **2025**, *13*, 368. https://doi.org/10.3390/math13030368

1. Introduction

In recent decades, with advancements in ML algorithms and computational tools, their application has garnered significant attention among financial experts and researchers [1–11]. One of the most critical fields in which they have been applied is risk management, particularly credit risk management. Many studies have used ML models to identify firms or customers likely to default compared to others. Machine learning (ML)-based models offer specific advantages. They require fewer assumptions compared to traditional models and can process a wider range of data. Unlike traditional models, which typically rely on accounting or market data [12], ML models incorporate a broader set of factors, such as cash flow, national governance, and capital structure, making them more effective in credit assessment [13–16].

While their performance often surpasses that of traditional human-based or structural models, they encounter some challenges [17–19]. One of these challenges is the structure of datasets [20]. ML models work with data and are regularly developed to handle balanced data [21], while in many credit datasets, inherent imbalances exist.

This is logical because defaults are rare occurrences. As it is often stated, machine learning models are typically designed for balanced datasets. However, in credit risk management, it is crucial to predict defaulters as accurately as possible due to the high cost of missing a defaulter in a credit system. This is similar to other challenges, such as disease detection, where misclassifying a member of the minority class is costlier than misclassifying a member of the majority class [22,23].

In some studies, to address the performance challenges of ML models, the dataset is balanced by selecting an equal number of defaulters and non-defaulters. However, this approach is unrealistic and often results in models with high bias due to the artificially balanced dataset. In other cases, this imbalance has simply been ignored, resulting in models that achieve high accuracy but exhibit low sensitivity.

Imbalanced data have an impact on the performance of models, although it may not be visible initially. For instance, in a hypothetical credit dataset, an ML model might predict both good and bad payers with the same number of incorrect classifications. In this case, because of the low number of defaulters, the whole number of payers that their label predicts correctly is high, and, as a result, the machine learning model reached high accuracy or even a high AUC score.

However, this undermines the model's ability to identify defaulters effectively because even a very small portion of non-defaulter firms, which we defined as the number of falsely labeled firms in ML model performance, can be a vast majority of defaulter firms, and, as a result, the model exhibits a poor function in finding the defaulter firms.

This issue is particularly evident in metrics like sensitivity, which are based on defining defaulted borrowers as positive or negative and measure the rate of identifying each data label category.

As a result, it is important to note that due to the high number of good payers and their better identification rate, general metrics like accuracy and AUC may appear high while sensitivity remains low. This discrepancy can lead to credit disasters, especially considering the high correlation of defaults.

In response to this challenge, several solutions have been proposed. Some solutions focus on modifying the data distribution by creating artificial instances or reducing certain instances. The second brunch emphasizes developing new algorithms that are experts on learning imbalance data. The third category aims to allocate different weights to different classes. The last category is referred to as cost-sensitive approaches.

Cost-sensitive approaches are constructed based on real-world outcomes. As a matter of fact, it is inevitable that when a default occurs after credit risk management predicts that the firm will not default (false positive), it is more expensive than preventing allocating capital to a firm that is predicted to default although it will not (false negative).

Many cost-sensitive approaches have been introduced in recent years; however, few of them have been studied to detect defaulter firms. In this essay, we first review research on imbalanced datasets and the performance of notable studies in credit prediction using cost-sensitive ML models. Then, we explore the performance of notable papers in credit prediction using cost-sensitive machine learning models. Finally, we evaluate the performance of one of the most recent cost-sensitive models for imbalanced datasets, as proposed in work by Devi et al. [24], which combines several decision tree-based models for the first time.

In this paper, for the first time, Devi et al.'s [24] (CorrOV-CSEn) method is used in business failure prediction. Additionally, we employ one of the state-of-the-art algorithms introduced in recent years, CatBoost, in conjunction with a cost-sensitive approach. We believe that CatBoost's ability to prevent overfitting is expected to enhance our model's performance. Our third contribution is the use of Iranian capital market firms as our dataset. The Iranian capital market is one of the oldest in the Middle East and recently was described by Bloomberg as one of the most "unfamiliar" large capital markets in the world [25].

The remainder of this paper is organized as follows: Section 2 provides a brief review of methods developed to address imbalanced dataset problems and their applications in credit risk management. Section 3 details the methodology employed in this study. In Section 4, the case study—focused on business failure in the Iranian capital market—is analyzed. Section 5 presents the experimental results, discussing the performance of each machine learning model. Finally, Section 6 concludes the paper with recommendations for future research.

2. Literature Review

Our literature review is divided into two parts. The first part addresses previous work on imbalanced datasets, primarily that developed by computer science scholars. The second part explores the application of these models in finance, with a focus on credit risk management.

2.1. Imbalance Datasets Solution

Numerous methods have been proposed to address the issue of imbalanced datasets. These approaches are generally categorized into three types [26]: (A) data-level (or resampling) methods, (B) algorithm-level methods, and C) cost-sensitive learning, respectively.

Data-level (or resampling) methods address imbalanced datasets by modifying the structure of the data. This can be achieved through under-sampling, oversampling, or hybrid resampling methods. In under-sampling, only a subset of the majority class is used of the majority class are trained. Methods such as Tomek links [27], Kubat and Matwin [28], Japkowicz [29], Neighborhood Cleaning Rule (NCL) [30], Relevant Information-based Under Sampling (RIUS) [31], Lee & Seo [32], and EUStack [33] are examples of under-sampling approaches.

Oversampling methods, on the other hand, involve creating additional copies of the minority class to balance the training set. Solberg and Solberg [34], WK-SMOTE [35], MAHAKIL [36], GSMOTE-NFM [37], SMOTEFUN [38], SMOTE-tBPSO-SVM [39], and Approx—SMOTE [40] are examples of such methods.

Hybrid resampling methods usually combine oversampling and under-sampling. The Synthetic Minority Oversampling Technique (SMOTE), introduced in 2002 by Chawla et al. [41], is a widely used hybrid resampling approach. Other hybrid resampling methods include Ling and Li [42], RFMSE [43], RK-SVM [26], SA-CGAN [44], SMOTified-GAN [45], and Puri & Gupta [46].

Algorithm-level methods focus on developing algorithms specifically designed to classify imbalanced data. The RUSBoost algorithm [47], Weighted Ensemble with One-Class Classification with Over sampling and Instance Selection (WECOI) [48], and Lasso-Logistic Regression Ensemble [49] are examples of these methods.

Cost-Sensitive Learning addresses misclassification by assigning different costs to errors. In traditional ML models, misclassifications—such as false negatives (FN) and false positives (FP)—are often treated equally. However, in reality, the consequences of these errors can vary significantly, especially in domains like credit classification. In this context, there is a loss function that considers four possible outcomes in a binary classification problem, such as distinguishing between defaulters and non-defaulters (or

1 and 0). The matrix below illustrates the cost matrix used in a regular ML algorithm for credit classification.

$$\begin{bmatrix} C(1,1) = 0 & C(1,0) = 1 \\ C(0,1) = 1 & C(0,0) = 0 \end{bmatrix} \tag{1}$$

In the cost matrix, C(i,j) represents the cost of labeling an instance X, with an actual value of j as i. When the instance is correctly labeled, there is no cost. However, for both types of mislabeling (false positives and false negatives), the cost is typically set to 1.

Cost-Sensitive Learning incorporates the loss function through two main approaches: direct and indirect. In the direct approach, the loss function influences the training process itself by adjusting the model based on misclassification costs. In the indirect approach, the loss function is applied after training, either by modifying decision thresholds or using a Bayesian decision framework to minimize expected costs [50].

In Cost-Sensitive Learning, different misclassification costs are taken into account. In real-world scenarios, the cost of a false positive (e.g., incorrectly classifying an unhealthy firm as healthy) can be significantly different from the cost of a false negative. Misclassification costs can be assigned using various approaches. As a result, while traditional machine learning methods focus on minimizing overall misclassification and maximizing accuracy, Cost-Sensitive Learning methods aim to minimize the total costs associated with different types of misclassification errors.

One of the most pioneering cost-sensitive methods was ICET, introduced by Turney in 1995 [51]. It was built on genetic algorithms. Other cost-sensitive models based on decision trees were introduced by Ling et al. [52] and Drummond and Holte [53].

Some cost-sensitive methods use a threshold probability for algorithms, which produces probabilities for each instance classification, such as MetaCost [54], CostSensitiveClassifier [55], Cost-sensitive naïve Bayes [56], and Empirical Thresholding [57].

Khan et al. [58] proposed a cost-sensitive method based on the deep Convolutional Neural Network that focuses on feature selection. They did not alter data distribution. Unlike previous models, they set class dependent costs automatically during the learning procedure. The efficiency of their model has been demonstrated in subsequent works [59]. The Cost-sensitive General Vector Machine (CFGVM) was proposed by Feng et al., which combines feature selection and GVM [60]. Devi et al. [24], combined AdaBoost ensemble learning with correlation-based oversampling in their proposed model.

2.2. Imbalanced Learning in Finance

Using machine learning methods in credit risk assessment has already been extensively explored in the literature. However, the vast majority of these studies have not considered the imbalanced nature of datasets [22]. Among the notable works in utilizing machine learning tools for predicting defaults, Khandani et al. [61] evaluated machine learning-based models for predicting credit card default risk. They employed four classifier thresholds to classify the data, achieving sensitivity values of 65%, 78%, 83%, and 88% for each threshold, respectively.

Barboza et al. [62] conducted a comprehensive study examining the credit risk of North American companies from 1985 to 2013. The dataset included 10,000 companies and aimed to predict defaults one year in advance. They employed various models, including support vector machines, bagging, boosting, and random forests and compared these to statistical models such as discriminant analysis, logistic regression, and neural networks. Their findings indicated that machine learning models outperformed traditional ones in predicting corporate defaults by up to 10%, as measured by the ROC score. Notably, the random forest model demonstrated exceptional accuracy, achieving 87%, which surpassed other models. However, the sensitivity of the random forest remained in the range of 0.76 to 0.83.

Yildrim [63] conducted a study to develop two models for predicting corporate defaults using a sample of 1 million Turkish companies from 2010 to 2018. The study evaluated logistic regression, decision tree, random forest, and gradient boosted tree models. The average AUC scores for these models were 0.76, 0.80, 0.82, and 0.82, respectively. However, the sensitivity of the three tree-based models was notably low, at 0.15, 0.17, and 0.30, respectively.

In a similar study using the same dataset, Peykani et al. [64] employed two machine learning models—random forest and gradient boosted trees—to predict business failure in the Iranian capital market. Both models achieved exceptionally high ROC scores of 0.97. However, their sensitivity for defaulted firms was 0.66 for random forest and 0.77 for gradient boosted trees.

Chen & Ribeiro [65] combined multiple classifiers, including KNN, support vector machines, and decision trees, using a consensus approach for bankruptcy prediction. The dataset consisted of 37 French firms, and the ensemble method aimed to improve the robustness and accuracy of predictions by integrating results from several machine learning techniques.

Bahnsen et al. [66] presented a cost-sensitive decision tree algorithm designed to account for the varying costs associated with different instances by incorporating a cost-based impurity measure. They introduced a new performance metric called "Saving" to evaluate model performance. This algorithm is tested on various real-world datasets, including credit card fraud detection and credit scoring. The results indicate that it outperforms other methods across all datasets, achieving significant cost savings of up to 71 percent compared to 32 percent for the benchmark while constructing smaller trees that are faster to build, requiring only one-fifth of the time needed for traditional decision trees.

Zakaryazad and Duman [67] addressed the challenge of imbalanced data by developing an Artificial Neural Network (ANN) model optimized to maximize profit rather than traditional accuracy. Their profit-oriented ANN incorporates a customized penalty function that assigns variable penalties based on the financial impact of correctly or incorrectly classifying each instance, modifying the typical sum of squared errors (SSE) function to weigh misclassifications according to each instance's profit significance. The findings from datasets in fraud detection and bank marketing indicate that the ANN and Naïve Bayes classifier outperform other models.

Xia et al. [68] explored peer-to-peer lending datasets using a cost-sensitive weighted XGBoost approach. Their study examined both financial and non-financial factors, with the primary evaluation metric being the annualized rate of return (ARR). The model aimed to enhance loan evaluation by balancing risks and returns for lenders.

Fiore et al. [69] demonstrated that generative adversarial networks (GANs) can be employed as an alternative resampling technique to enhance credit card fraud modeling. Notably, early default has received less attention in the literature.

Papouskova and Hajek [70] proposed a two-stage ensemble learning model to evaluate default risk in consumer credit, particularly in P2P lending. In the first stage, they employed heterogeneous classification ensemble models to predict whether a P2P loan would default. In the second stage, they applied heterogeneous regression ensemble models to estimate the exposure at default for loans that had defaulted. Their findings demonstrated that the two-stage method outperformed single-stage approaches, with the ensemble method achieving greater predictive accuracy compared to traditional credit scoring models. They employed a diverse range of algorithms, including Decision Tree (C4.5), Logistic Regression, SVM, random forest, and AdaBoost.

De Bock et al. [71] addressed uncertainty in misclassification costs for business failure prediction through a heterogeneous ensemble framework. The model incorporated bagging, random forests, and multi-objective optimization and was evaluated on 21 datasets spanning various industries. The results highlighted the model's adaptability to scenarios involving unknown or dynamic misclassification costs.

Hou et al. [72] proposed an innovative approach to addressing imbalanced data in credit scoring. Recognizing the limitations of traditional static ensemble methods, they introduced a dynamic ensemble selection (DES) model specifically designed for imbalanced classification tasks. The model first applied SMOTE (Synthetic Minority Over-Sampling Technique) to balance the dataset, thereby creating a more effective candidate classifier pool. Additionally, they integrated DES-MI, a weighting mechanism that prioritizes minority instances during the evaluation of classifier competence. For further refinement, they applied META-DES for a comprehensive multi-criteria assessment and used DES-KNN to balance classifier competence with diversity. Testing on 15 imbalanced datasets demonstrated that the proposed model outperformed other DES approaches in terms of AUC performance. Moreover, when evaluated on real P2P loan data, it achieved a lower Type I error rate compared to XGBoost and LightGBM, highlighting its potential for more accurate credit risk predictions. This model is particularly valuable for applications where false positives carry significant financial consequences.

Li et al. [73] applied credit scoring tools to identify high-risk borrowers, including online loan fraudsters. Using ML-LightGBM, they aimed to more effectively identify early stage defaulters. To enhance prediction accuracy, the authors incorporated a cost-sensitive framework into the loss function of the classification model. Tested on a dataset of 1.6 million online loans, their method demonstrated that the cost-sensitive ML-LightGBM approach outperformed previous models in predictive performance, underscoring its effectiveness for fraud detection and credit scoring.

Barbaglia et al. [74] investigated default behavior in European residential mortgages leveraging a dataset of 12 million loans across multiple countries. They modeled loan default as a function of variables such as borrower profiles, loan characteristics, and regional economic conditions. By comparing cost-sensitive machine learning algorithms with traditional logistic regression, they demonstrated that machine learning methods significantly enhanced prediction accuracy. Their models included gradient boosted trees, XGBoost, and Logistic Regression. They employed both under-sampling and over-sampling techniques. Gramegna and Giudici [75] evaluated their model on real-world data from Italian small and medium enterprises, employing XGBoost with an under-sampling approach. Zou et al. [76] applied XGBoost with a cost matrix to predict business failures in the Chinese capital market. They utilized a diverse set of 47 financial ratios as features in their dataset. The model was compared to various other statistical and machine learning models, and the results indicated that XGBoost with a cost matrix excelled in minimizing Type II errors.

Chi et al. [77] introduced a novel instance-dependent, misclassification cost-sensitive algorithm for default prediction. The study proposed two classifiers—misclassification cost-sensitive Logistic Regression (MCSLR) and misclassification cost-sensitive Neural Network (MCSNN)—and evaluated their performance by minimizing Type I and Type II errors, thus improving prediction accuracy in financial decision making. Wang and Chi [78] utilized a cost-sensitive stacking ensemble learning method to predict financial distress among 3425 Chinese companies from 2000 to 2020. The study employed statistical tests, including T-tests and Wilcoxon non-parametric tests, to validate the significance of differences in financial distress predictions, underscoring the effectiveness of the ensemble method. Table 1. provides a summary of the discussion in this section.

Table 1. A summary of the studies conducted.

Year	Research	Method of Imbalanced Data	Machine Learning Model	Dataset
2013	Chen & Ribeiro [65]	Cost-sensitive	KNN Support Vector Machines Decision Trees	37 French firms
2015	Bahnsen [79]	Cost-sensitive	Decision Trees	Credit card transactions and customer data
2016	Zakaryazad and Duman [67]	Cost-sensitive	ANN	Credit card fraud detection
2017	Xia et al. [68]	Cost-sensitive	XGBoost	Two real-world P2P lending datasets
2017	Fiore et al. [69]	Resampling	GAN	credit card fraud
2019	Papouskova and Hajek [70]	Cost-sensitive	Decision Tree (C4.5) Logistic regression SVM Random forest AdaBoost	P2P lending consumer loans
2020	De Bock et al. [71]	Cost-sensitive	Bagging Random forests	21 datasets across various industries
2020	Hou et al. [72]	Resampling	XGBoost LightGBM	P2P loan
2021	Li et al. [73]	Cost-sensitive	LightGBM	1.6 million online loans
2021	Barbaglia et al. [74]	Cost-sensitive	XGBoost Gradient Boosted tree Logistic Regression	12 million loans
2021	Gramegna and Giudici [75]	Resampling	XGBoost	Italian small and medium enterprises
2022	Zou et al. [76]	Cost-sensitive	XGBoost	Chinese capital market
2022	Chi et al. [77]	Cost-sensitive	Logistic Regression Neural Network	
2024	Wang and Chi [78]	Cost-sensitive	Ensemble learning method	3425 Chinese companies from 2000 to 2020
2024	**Our Research**	Cost-sensitive and Resampling (CorrOV-CSEn)	Random forest Gradient Boosted tree AdaBoost XGBoost CatBoost	Iranian capital market firms

3. Methods

3.1. CorrOV-CSEn

In this study, we employed recently introduced Correlation-based Oversampling Aided Cost-Sensitive Ensemble learning (CorrOV-CSEn) technique. CorrOV-CSEn integrates two complementary approaches for handling imbalanced datasets. First, it applies correlation-based oversampling to better prepare the dataset. Then, the prepared data are used in a cost-sensitive ensemble algorithm, specifically Adaboost in some cases, but also in combination with other ensemble learning methods. The primary goals of CorrOV-CSEn are to reduce redundant data generation, prevent overfitting, and improve the classification accuracy of the minority class. Generally, CorrOV-CSEn follows a two-step process, as detailed below. Figure 1 describes an overview of the CorrOV-CSEn process.

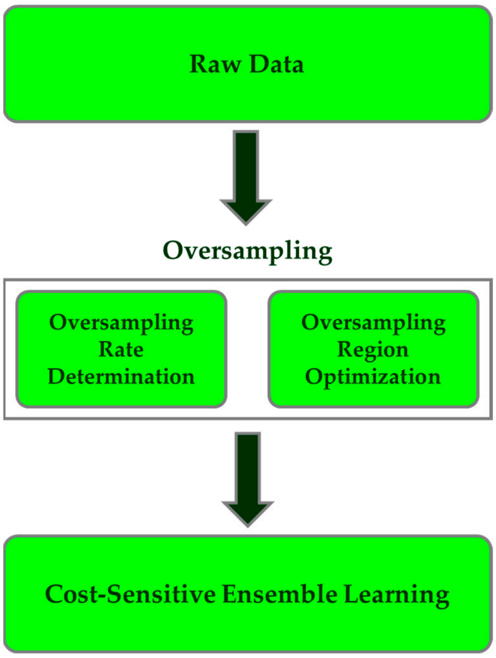
Raw Data
Oversampling
Oversampling Rate Determination
Oversampling Region Optimization
Cost-Sensitive Ensemble Learning

Figure 1. Overview of the CorrOV-CSEn process.

3.1.1. Correlation-Based Oversampling

This step enhances the performance of traditional oversampling methods like SMOTE by incorporating correlation information into the process. Specifically, we employ a Linear Covariance Matrix (LCM) [80] to determine the optimal level of oversampling. The LCM is calculated using the following equation:

$$\sum A = \frac{1}{|NN(X_a)|} \sum_{X \in NN(X_a)} \left(Y - \overline{Y}\right)\left(Y - \overline{Y}\right)^T \quad (2)$$

where

- $\sum A$ represents the Linear Covariance Matrix (LCM);
- X_a is a minority class instance;
- $NN(X_a)$ denotes the k-nearest neighbors (*K-NN*) of X_a;
- Y is the matrix of K-NN instances of X_a;
- $\overline{Y}$ is the centroid of the Y matrix.

The Linear Covariance Matrix (LCM) is utilized in two critical ways:

- Oversampling rate determination: Higher LCM values, particularly among the K-NN of the same minority class, indicate stronger correlation and guide a higher oversampling rate. This strategy reduces variance and generates synthetic instances in regions with higher minority class correlations, especially near borderline instances.
- Oversampling region optimization: For each minority instance, oversampling is performed only if its LCM with respect to the K-NN of the same class label is greater than its LCM with instances from other classes. This ensures that synthetic data are generated in the most relevant regions, enhancing both model robustness and the quality of the generated samples.

3.1.2. Cost-Sensitive Ensemble Learning

After applying correlation-based oversampling, the prepared data are fed into an ensemble learning framework. While previous studies, such as those by Devi et al. [24], used AdaBoost [81], this study, in addition to AdaBoost, explores a broader range of ensemble methods to assess their performance. These methods include Multi-Layer Perceptron (MLP), random forest [82], gradient boosted trees [83], XGBoost [84], and CatBoost [85]. Each of these ensemble models is adapted to be cost-sensitive, focusing on minimizing the misclassification costs associated with the minority class, which is crucial for handling imbalanced datasets. We describe these methods in detail.

Multi-Layer Perceptron (MLP)

The Multi-Layer Perceptron (MLP) [86], a type of feedforward artificial neural network, is widely used for both classification and regression tasks due to its flexibility and ability to model complex, non-linear relationships. The MLP consists of multiple layers of neurons, where each neuron is connected to the neurons in the subsequent layer through weighted connections. The learning process involves adjusting these weights to minimize prediction error. The algorithm's process can be summarized as follows [87]:

1. An MLP consists of an input layer, one or more hidden layers, and an output layer. Each layer is composed of several neurons (nodes). If the dataset contains M features, the input layer will have M neurons. The number of neurons in the hidden layers can be chosen based on the complexity of the task. Each neuron applies a weighted sum of inputs followed by a non-linear activation function such as ReLU or sigmoid. Mathematically, the output of a neuron can be expressed as

   $$z = \sum_{i=1}^{M} w_i x_i + b$$

   where w_i are the weights of the connections, x_i are the input features, and b is the bias term. The neuron output after applying the activation function f is

   $$a = f(z)$$

2. During forward propagation, inputs pass through the network from the input layer to the output layer. Each hidden layer neuron processes the weighted sum of inputs and applies the activation function. The final output layer provides predictions, which can be either Classification or Regression.
3. The loss function quantifies the error between the predicted output and the actual target. For regression, the Mean Squared Error (MSE) is often used.
4. Backpropagation and Weight Update: The gradient of the loss function is calculated using the chain rule, and weights are updated using gradient descent.

3.2. Random Forest

The random forest algorithm, introduced by Leo Breiman in 2001 [82], is among the most widely used and accurate machine learning techniques, including applications in credit risk management [88–91]. It constructs an ensemble of decision trees by drawing random subsets from the dataset and combines predictions from multiple "weak" models to create a robust "strong" model. Based on CART (Classification and Regression Trees), each tree is independently trained on a bootstrapped sample—a random subset chosen with replacement. The algorithm's process can be summarized as follows [92]:

1. Bootstrap Sampling: For each of the T trees in the forest, a random subset of the data is drawn with replacement. If there are N total samples, then each tree is built from a subset D_t of N samples drawn randomly with replacement, resulting in different training sets for each tree:

   $$D_t = \{x_i, y_i\} \; where \; i \in \{1, 2, \ldots, N\} \tag{3}$$

2. Feature Selection: At each node of the decision tree, a random subset of features is chosen, typically equal to the square root of the total number of features M in classification tasks (i.e., $\sqrt{M}$). This helps reduce the correlation between trees and improve model variance. For regression, the number of selected features is often $M/3$. This features minimizes correlations among the trees [60].
3. Splitting Criterion: From the selected subset of features at each node, the feature that best splits the data is chosen using a splitting criterion, often the Gini index or entropy. For example, the Gini index G for a split can be calculated as

$$G = 1 - \sum_{i=1}^{C} p_i^2 \tag{4}$$

4. Building the Forest: Each tree is grown to its full depth without pruning, resulting in a collection of deep, unpruned trees. By default, 500 trees are built, though this number can be adjusted for specific applications.
5. Prediction Aggregation: For classification tasks, the final prediction for each data point is determined by majority voting across all trees. Let $h_t(x)$ represent the prediction of the $t - th$ tree for a data point x. Then, the final prediction $H(x)$ is given by

$$H(x) = mode\{h_1(x), h_2(x), \ldots,\ h_T(x)\ \} \tag{5}$$

For regression tasks, the final prediction is the average of all tree outputs:

$$H(x) = \frac{1}{T} \sum_{t=1}^{T} h_t(x) \tag{6}$$

Random sampling and feature selection in random forest reduce the variance of individual trees while minimizing correlations among them, producing an ensemble with lower variance and higher accuracy. Each tree in the forest is uncorrelated with the others, enhancing the model's robustness.

3.3. Gradient Boosted Trees

Gradient boosted trees (GBT), introduced by Friedman in 2000 [83], extend the boosting concept to decision trees by building a sequence of models that iteratively minimize errors. Each model focuses on correcting the errors of its predecessor, creating a strong learner from a series of weak learners. Unlike bagging, which trains independent models on random subsets of data (as used in random forest), boosting involves sequential training where each model improves upon the previous one [93].

Boosting operates on the principle that a robust learning model can be constructed by combining multiple complementary weak models. Unlike bagging [94], boosting does not divide the dataset into random subsets. Instead, it assigns higher weights to samples that were misclassified in previous iterations, refining the model step-by-step. This process continues until the model achieves a desired level of accuracy or the error is minimized [95].

In GBT, the first decision tree $T_1(x)$ is trained on the original target values y. Subsequent trees are trained on the residuals (errors) of the preceding models to progressively reduce the remaining error. For example, if y is the target value, the residuals for the first tree are calculated as

$$r_i^{(1)} = y_i - T_1(x_i) \tag{7}$$

In each successive step m, a new tree $T_m(x)$ is trained to predict the residuals from the prior model. The model update process can be summarized as follows:

1. Initialize the model: Start with an initial estimate, often taken as the mean value of the target variable for regression tasks or a single weak classifier for classification.

$$F_0(x) = argmin_{\gamma} \sum_{i=1}^{N} L(y_{i,}\ \gamma) \tag{8}$$

where L is the loss function, such as squared error for regression or log-loss for classification.

2. Iterative Model Updates: For each iteration $m = 1,\ 2,\ \ldots,\ M$:
   - Compute the Residuals: Calculate the residuals $r_i^{(m)}$ for each sample based on the current model $F_{m-1}(x)$:

$$r_i^{(m)} = -\frac{\partial L(y_i,\ F_{m-1}(x_i))}{\partial F_{m-1}(x_i)} \tag{9}$$

   - Fit a New Tree: Train a new decision tree $T_m(x)$ to predict the residuals $r_i^{(m)}$.
   - Update the Model: Add the new tree to the model with a learning rate η (to control the contribution of each tree), yielding an updated model:

$$F_m(x) = F_{m-1}(x) + \eta T_m(x) \tag{10}$$

3. Final Prediction: After M iterations, the final model $F_M(x)$ is an ensemble of the trees, each adjusted to reduce the error from prior steps. For regression, the final prediction is

$$\hat{Y} = F_M(x) = F_0(x) + \sum_{m=1}^{M} \eta T_m(x) \tag{11}$$

The sequential nature of boosting, combined with gradient descent optimization, allows gradient boosted trees to achieve high accuracy and performance on various datasets. This algorithm is well-known in credit risk prediction [89].

3.4. XGBoost

XGBoost, introduced by Tianqi Chen in 2016 [84], is an optimized implementation of gradient boosted trees (GBT) designed to be both efficient and scalable. XGBoost enhances traditional gradient boosting by adding regularization techniques, tree pruning, and advanced handling of missing data, making it well-suited for high-dimensional datasets [96]. These improvements help XGBoost achieve high predictive accuracy and robustness while avoiding overfitting [97].

One of the key differentiators of XGBoost from other GBT methods is its use of both L1 (Lasso) and L2 (Ridge) regularization. These regularization terms penalize the complexity of the model, ensuring that the final model generalizes well even with large datasets:

1. Objective Function: The objective of XGBoost is to minimize a regularized loss function that combines the traditional loss function with regularization terms for complexity control. For T trees, the objective function Obj is defined as

$$Obj = \sum_{i=1}^{N} L(y_i,\ \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t) \tag{12}$$

where

   - $L(y_i,\ \hat{y}_i)$ is the loss function, such as mean squared error for regression or log-loss for classification;
   - $\Omega(f_t) = YT + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$ is the regularization term with parameters γ and λ, controlling the complexity of each tree.

2. Tree Structure and Growth: Each tree in XGBoost is built to minimize the residuals from the previous trees, following the same general structure as GBT. However, XGBoost introduces a tree-pruning technique, where trees are pruned based on their impact on the objective function rather than growing to full depth. The max_*depth* parameter controls the maximum depth of each tree, preventing the model from overfitting by limiting tree complexity.
3. Update Process: In each iteration, the algorithm calculates the best tree structure to minimize the residuals of the previous ensemble. The updates are computed using second-order gradients (Hessian) of the loss function, making it more efficient. The model update at each step t is given by

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i) \tag{13}$$

where η is the learning rate and $f_t(x_i)$ is the output of the $t - th$ tree.
4. Handling Missing Data: XGBoost automatically manages missing data by learning optimal paths for instances with missing values during training. It assigns missing values to the most suitable branch, improving model accuracy when dealing with incomplete datasets.
5. Final Prediction: The final prediction is an aggregation of all trees, represented as

$$\hat{y} = F(x_i) = \sum_{t=1}^{T} f_t(x_i) \tag{14}$$

where x_i represents the input features, and $f_t(x_i)$ is the output from the $t - th$ tree. For classification, the final output is often determined by applying a softmax function to convert the aggregated score to class probabilities.

By integrating these innovations, XGBoost achieves a high degree of accuracy and efficiency, making it particularly effective for complex tasks such as handling imbalanced datasets and financial failure prediction [88,89,98,99].

3.5. AdaBoost

Adaboost, short for Adaptive Boosting, is an ensemble learning method designed to create a strong classifier by combining multiple weak classifiers. The core idea behind Adaboost, like Boosting, is to iteratively adjust the weights of the training samples, placing greater emphasis on those that were misclassified in previous rounds. This approach enhances the overall model's accuracy by forcing each weak classifier to focus more on challenging cases.

Initially, Adaboost assigns equal weights to all training sample. In each iteration, it selects the weak classifier that performs best on the current weighted dataset and updates the sample weights based on its classification results. Misclassified samples receive higher weights in the next round, while correctly classified samples are assigned lower weights. This ensures that previously misclassified samples receive more attention in subsequent rounds, improving the model's overall accuracy.

The Adaboost process can be formalized as follows [81]:

1. Initialize sample weights: Each sample i in the training set receives an initial weight:

$$w_i^{(1)} = \frac{1}{N} \tag{15}$$

where N is the number of training samples.

2. Train a weak classifier: In each round t, a weak classifier $h_t(x)$ is trained on the weighted samples, and its error rate ϵ_t is calculated as

$$\epsilon_t = \sum_{i=1}^{N} w_i^{(t)}.1(h_t(x_i) \neq y_i) \tag{16}$$

3. Calculate the classifier's weight: The weight of the weak classifier is determined based on its accuracy:

$$a_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{17}$$

4. Update sample weights: Sample weights are updated to reflect the classifier's performance, giving more weight to misclassified samples:

$$w_i^{(t+1)} = w_i^{(t)}.\exp(a_t.1(h_t(x_t) \neq y_i)) \tag{18}$$

5. Combine weak classifiers: The final strong classifier $H(x)$ is a weighted sum of all weak classifiers:

$$H(x) = sign\left(\sum_{i=1}^{T} a_t.h_t(x)\right) \tag{19}$$

Through these iterations, Adaboost creates a robust ensemble model capable of generalizing well across various datasets, improving classification accuracy significantly, especially for imbalanced datasets.

3.6. CatBoost

CatBoost, introduced by Prokhorenkova et al. in 2018 [100], is a powerful and efficient implementation of gradient boosted trees (GBT) designed to reduce overfitting and improve predictive accuracy, especially with categorical features. The primary innovation in CatBoost is the use of ordered boosting, a technique developed by Dorogush et al. [101], to address the target leakage problem that often arises in standard boosting algorithms. This feature makes CatBoost particularly effective on small- to medium-sized datasets, where target leakage can significantly impact model performance.

CatBoost offers several unique improvements over traditional GBT methods [85]:

1. Ordered Boosting to Avoid Target Leakage: In standard GBT, future data points might unintentionally influence earlier predictions, leading to target leakage. Ordered boosting solves this by using a permutation-based scheme, ensuring that only past information influences each iteration. This ordered approach is particularly useful in datasets where feature-target relationships are complex and dynamic, and it enhances CatBoost's accuracy.
2. Handling of Categorical Variables: CatBoost automatically handles categorical features without requiring extensive preprocessing. It converts categorical features into numeric representations through a series of random permutations, using them to guide the splitting criteria for each decision tree.
3. Objective Function: CatBoost minimizes a regularized loss function similar to other boosting methods, but with an emphasis on ordered boosting:

$$Obj = \sum_{i=1}^{N} L(y_i.\hat{y}_i) + \sum_{j=1}^{J} \Omega(f_j)) \tag{20}$$

where

- $L(y_i,\ \hat{y}_i)$ is the loss function (e.g., cross-entropy or log-loss for classification tasks);
- $\Omega(f_j)$ is the regularization term for tree complexity, helping to control overfitting.

4. Tree Structure and Decision Rule: CatBoost uses binary decision trees as base learners. For each input x_i, the decision tree assigns it to one of the leaf regions R_j based on a series of splits. The function for each tree can be represented as

$$H(X_i) = \sum_{j=1}^{J} C_j.1_{x \in R_j} \tag{21}$$

where

- $H(X_i)$ represents the decision function for each sample X_i;
- R_j is the disjoint region corresponding to each leaf in the tree;
- C_j is the predicted output value for region R_j.

5. Final Prediction: *The* final prediction is the aggregation of all the trees in the ensemble. For a dataset with T trees, the final output Z is given by

$$Z = F(X_i) = \sum_{t=1}^{T} f_t(X_i) \tag{22}$$

where $f_t(X_i)$ is the output of the $t - th$ tree for a given input X_i. For classification, the model often applies a sigmoid or softmax transformation to convert the output into class probabilities.

6. Regularization and Overfitting Prevention: CatBoost uses random permutations when selecting tree splits, which reduces overfitting and enhances model generalization. This, combined with ordered boosting, allows CatBoost to outperform traditional GBT methods on many complex tasks.

CatBoost have been applied in several papers in order to financial failure prediction [102,103], in this article, we applied a dost-sensitive approach toward them for the first time.

By combining correlation-based oversampling with cost-sensitive ensemble learning, the CorrOV-CSEn approach minimizes overfitting and significantly enhances the classification accuracy of the minority class compared to traditional methods.

3.7. Business Failure

In our study, we emphasize the concept of business failure rather than terms like default or bankruptcy. Business failure refers to a situation where a firm faces significant challenges in continuing its operations. It is a broader concept than default and bankruptcy. A firm experiencing business failure is likely to default, which may eventually lead to bankruptcy if it reaches specific legal thresholds and undergoes the legal process of resolution.

In countries like Iran, where the government plays a significant role in the economy [104,105] and the operation of major companies, firms are often prevented from defaulting and declaring bankruptcy in the capital and debt markets. However, the concept of business failure provides a valuable perspective for assessing credit risk. Business failure has been examined in other studies, particularly in relation to macroeconomic conditions.

In Iran's capital market, business failure is closely associated with "Article 141 of the Amended Commercial Code." This regulation requires companies that fall under Article 141 to present a detailed recovery plan. The correlation between Article 141 and business failure is evident in its focus on both financial losses and the proportion of those losses relative to the company's capital. A company falling under Article 141 has accumulated losses that exceed its equity, meaning its assets have dropped below its liabilities, which signals potential insolvency.

Figure 2 illustrates the percentage of firms in each year that failed under Article 141 as a proportion of the total number of firms in that year.

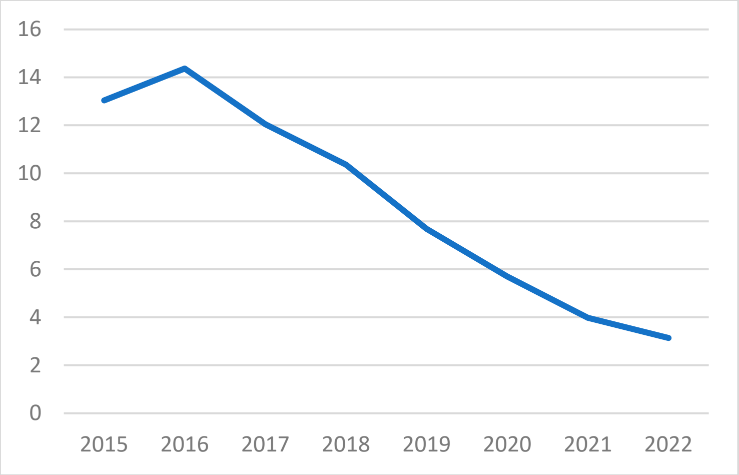

Figure 2. Percentage of firms failing under Article 141 each year from 2015 to 2022.

3.8. Evaluating Methods

In our research, we utilized ratios derived from the elements of the confusion matrix, which offers valuable insights into the overall performance of the model. The confusion matrix is commonly used to assess the performance of binary classification models, where the aim is to differentiate between failed companies (positive class) and healthy companies (negative class).

$$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \tag{23}$$

In the confusion matrix, TP or true positive refers to instances that are actually positive and were correctly identified by the model. TN or true negative indicates instances that are actually negative and correctly classified. FP or false positive represents instances that were predicted as positive but are actually negative, while FN or false negative refers to positive instances incorrectly classified as negative.

Based on the confusion matrix elements, various ratios are introduced to evaluate the performance of binary classification models. In this research, we used three key ratios: recall, precision, and F1 score, which will be explained in order of their significance.

Recall or sensitivity, calculated using Formula (3), measures the model's success in identifying failed companies. This metric is considered the most important, as a good credit model should be able to identify all failing companies and prevent misclassifying them as healthy.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{24}$$

Precision, calculated using Formula 4, evaluates the accuracy of the model in identifying failing companies. In other words, it indicates the likelihood that a company identified as failing by the model is indeed failing.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{25}$$

F1 score is a metric used to evaluate binary classification models, especially in cases where there is an imbalance between the positive and negative classes. The F1 score is the harmonic mean of precision and recall, calculated using the following formula:

$$\text{F1 Score } = \frac{2 * \text{Precision} * \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \tag{26}$$

It balances the two metrics, offering a comprehensive measure of a model's performance by considering both how well the model identifies failed companies (recall) and the accuracy of those predictions (precision). This score is particularly important when both false positives and false negatives carry significant costs.

3.9. Statistical Significance Test

We use the Friedman–Nemenyi test to detect significant differences among the models. This approach is commonly employed in research involving machine learning models, particularly those related to business failure. The Friedman test is suitable for comparing three or more groups, especially when the assumption of normality is violated. It extends the Wilcoxon signed-rank test by incorporating an additional assumption of sphericity [106]. The Friedman statistic is calculated as described by Friedman (1937) [107,108]:

$$X_F^2 = \frac{12}{nk(k+1)} \sum R_i^2 - 3n(k+1) \tag{27}$$

where

- n is the number of data sets (blocks);
- k is the number of models (groups);
- R_i^2 is the sum of ranks for each model.

H_0 is that there is no significance difference between the two models that have been compared, and if X_F^2 crosses the critical value, then H_0 is rejected. When H_0 is rejected, then the Nemenyi test is used.

4. Case Study

The statistical population of the research comprises all companies in the Iranian capital market from 2015 to 2022. Each instance represents a firm's annual information, with instances labeled as either "defaulted" or "healthy." In Iran's economy, the government prohibits large companies from declaring bankruptcy or default. Consequently, similar to most credit risk research in Iran, default and bankruptcy are defined based on Article 141 of the proposed amendment to a section of the Commercial Code. According to this article, if a company loses at least half of its capital due to incurred losses, the board of directors must promptly convene an extraordinary general meeting of shareholders to decide on the company's dissolution or survival. Article 141 effectively identifies conditions indicative of financial distress, and due to the accessibility of this information, it is used by researchers in the Iranian capital market. The following section reviews the models employed, detailing the parameters and calculation methods for each model.

We divided our sample into training and test datasets based on the years. Instances from 2015 to 2021 were considered as training datasets, and instances from 2021 to 2022 were also considered as training datasets.

In this research, as the focus of our investigation involves companies whose shares are traded in the capital market, we have made efforts to categorize variables into two main groups: financial statement-based variables and variables related to the company's stock price. These variables are considered the most fundamental information available for companies in the capital market [109].

Barboza et al. [62] conducted one of the most comprehensive studies investigating the default risk of companies in the North American capital market from 1985 to 2013. They employed two research approaches to determine their dataset variables. Firstly, they utilized the variables of the Altman model [110], a fundamental model designed to estimate the default risk of companies. Secondly, they also incorporated the variables used by Carton & Hofer [111], which are based on the growth rate of some fundamental company variables [62]. Our features are derived from the balance sheet, which is essential in credit studies [112].

It is essential to mention that the criterion used in this research for default is not the actual default but the inclusion in Article 141, which is measured based on the ratio of the retained earnings to the registered capital of the company. One of the variables used by Altman (variable X2), representing the ratio of retained earnings to registered capital, is excluded from the dataset variables list. The reason for excluding Altman's X2 is that the default criterion in this study already relies on the same ratio, thus avoiding redundancy and overlapping metrics. Additionally, one of the Carton & Hofer variables, GE, which measures the growth in the company employee count, was removed due to the lack of complete and reliable data.

The variables of the training and test datasets are as follows, as shown in Table 2, considering the aforementioned points.

Table 2. Features of the dataset and their respective formulas.

Variable	Formula
X1	$Net\ Working\ Capital\ /Total\ assets$
X3	$Earnings\ before\ interest\ and\ taxes\ /Total\ assets$
X4	$Market\ value\ of\ share * number\ of\ shares\ /Total\ debt$
X5	$Sales\ /Total\ assets$
OM	$Earnings\ before\ intrest\ and\ taxes\ /Sales$
GA	$Total\ assets_t - Total\ assets_{t-1}/Total\ assets_t$
GS	$Sales_{t-}Sales_{t-1}/Sales_{t-1}$
CROE	$ROE_{t-}ROE_{t-1}$
CPB	$Price - to - Book_{t-}Price - to - Book_{t-1}$

Table 3 shows the statistical description of our training and test data. The table provides a statistical summary of the training and test datasets, detailing key variables (e.g., X1, X3, and X4). Metrics such as the mean, standard deviation, minimum, maximum, and quartiles offer insights into the distribution of each variable. X4 and GS exhibit considerable variability, with large standard deviations and extreme maximum values. The training set shows more stability, while the test set includes outliers, particularly for X4 and GS. These variations could impact the model's predictive performance and generalizability.

Table 3. Statistical description of our training and test data.

Training Set	**X1**	**X3**	**X4**	**X5**	**OM**	**GA**	**GS**	**CROE**	**CPB**
count	2987	2987	2987	2987	2987	2987	2987	2987	2987
mean	0.083	0.129	19.821	0.724	−2.688	0.374	349.054	0.698	0.057
std	0.682	0.182	104.381	0.720	129.034	1.536	19,013.508	8.601	3.794
min	−16.681	−2.109	0.002	−0.192	−6824.769	−0.786	−203.866	−181.728	−112.889
25%	−0.046	0.026	1.339	0.219	0.061	0.038	−0.014	−0.266	−0.077
50%	0.145	0.106	4.532	0.577	0.192	0.176	0.257	0.143	0.013
75%	0.341	0.222	13.310	1.001	0.463	0.429	0.671	1.505	0.124
max	0.982	0.842	4133.761	7.780	230.176	68.611	1,039,154.000	190.281	125.772
Test set	X1	X3	X4	X5	OM	GA	GS	CROE	CPB
count	1240	1240	1240	1240	1240	1240	1240	1240	1240
mean	0.224	0.192	1407.757	0.805	0.463	0.581	0.970	−0.771	−0.111
std	0.365	0.187	18,561.378	0.787	4.646	1.882	11.161	4.969	4.079
min	−3.494	−0.781	0.001	−0.579	−18.486	−0.637	−27.413	−99.452	−129.227

Table 3. *Cont.*

Training Set	X1	X3	X4	X5	OM	GA	GS	CROE	CPB
25%	0.057	0.061	3.283	0.271	0.123	0.174	0.162	−1.218	−0.143
50%	0.232	0.179	7.290	0.633	0.286	0.366	0.479	−0.173	−0.013
75%	0.402	0.313	16.479	1.132	0.622	0.612	0.828	0.425	0.083
max	1.000	0.838	387,142.019	7.467	159.588	44.695	385.756	32.016	47.076

Table 4 presents skewness and kurtosis values for variables in both the training and test sets. Skewness measures asymmetry, with values near zero indicating symmetric distributions. Many variables, especially in the training set (e.g., X1: −10.814, OM: −50.328), show high positive or negative skewness, indicating significant asymmetry.

Table 4. Skewness and Kurtosis values for variables in total datasets.

	Skewness		Kurtosis	
	Training Set	Test Set	Training Set	Test Set
X1	−10.814	−2.181	199.295	17.969
X3	−1.193	−0.008	15.695	0.939
X4	25.938	17.379	897.031	319.686
X5	2.584	2.417	12.077	10.372
OM	−50.328	32.369	2629.754	1113.145
GA	32.133	19.269	1337.749	426.320
GS	54.653	33.171	2986.988	1142.797
CROE	1.569	−7.078	172.100	136.141
CPB	5.855	−24.427	725.704	826.716

Kurtosis measures the "tailedness" of the distribution. High values, such as GS (2986.988 in the training set), suggest extreme outliers. The test set generally shows lower kurtosis, indicating more moderate outliers compared to the training set.

Table 5 shows the correlation matrix among features for both the training and test sets.

Table 5. Correlation matrix among features in the training and test datasets.

Training Set	X1	X3	X4	X5	OM	GA	GS	CROE	CPB
X1	1.000	0.529	0.103	0.097	0.255	0.040	−0.004	−0.173	0.008
X3	0.529	1.000	0.130	0.281	0.173	0.075	−0.015	−0.139	0.016
X4	0.103	0.130	1.000	−0.032	0.005	0.025	−0.003	0.201	0.002
X5	0.097	0.281	−0.032	1.000	0.022	−0.018	−0.016	0.003	0.008
OM	0.255	0.173	0.005	0.022	1.000	0.013	0.000	−0.013	0.001
GA	0.040	0.075	0.025	−0.018	0.013	1.000	0.003	−0.242	0.005
GS	−0.004	−0.015	−0.003	−0.016	0.000	0.003	1.000	−0.001	−0.003
CROE	−0.173	−0.139	0.201	0.003	−0.013	−0.242	−0.001	1.000	0.001
CPB	0.008	0.016	0.002	0.008	0.001	0.005	−0.003	0.001	1.000
Test set	X1	X3	X4	X5	OM	GA	GS	CROE	CPB
X1	1.000	0.445	0.127	0.017	−0.001	−0.046	−0.025	−0.029	0.024
X3	0.445	1.000	−0.115	0.321	0.026	−0.003	0.112	−0.045	0.042
X4	0.127	−0.115	1.000	−0.086	0.009	−0.028	−0.016	0.012	−0.001
X5	0.017	0.321	−0.086	1.000	−0.049	−0.027	0.009	0.054	0.035
OM	−0.001	0.026	0.009	−0.049	1.000	−0.007	−0.001	0.004	−0.008
GA	−0.046	−0.003	−0.028	−0.027	−0.007	1.000	−0.001	−0.606	−0.023
GS	−0.025	0.112	−0.016	0.009	−0.001	−0.001	1.000	−0.007	0.037
CROE	−0.029	−0.045	0.012	0.054	0.004	−0.606	−0.007	1.000	0.005
CPB	0.024	0.042	−0.001	0.035	−0.008	−0.023	0.037	0.005	1.000

5. Experimental Discussion

5.1. Evaluation Among Models

Table 6 shows the results of applying SMOTE and CorrOV-CSEn across different machine learning methods. We summarize all the results here and highlight the best result for each aspect among the models in bold.

Table 6. Performance metrics for different machine learning models.

Model	Sensitivity	Precision	F1 Score
CorrOV-CSEn			
Multi-Layer Perceptron (MLP)	0.841	0.327	0.471
Random Forest	0.886	0.375	0.527
Gradient Boosting	0.795	0.443	0.569
XGBoost	0.795	0.393	0.526
AdaBoost	0.750	**0.478**	**0.584**
CatBoost	**0.909**	0.201	0.329
SMOTE			
Multi-Layer Perceptron (MLP)	0.841	0.327	0.471
Random Forest	**0.795**	0.603	0.686
Gradient Boosting	0.727	0.603	0.660
XGBoost	0.772	0.554	0.645
AdaBoost	0.568	0.555	0.561
CatBoost	0.750	**0.717**	**0.733**

The performance evaluation of the Multi-Layer Perceptron (MLP), random forest, gradient boosting, XGBoost, AdaBoost, and CatBoost models reveals significant differences in their classification accuracy.

❖ CorrOV-CSEn Results:

- Multi-Layer Perceptron (MLP) shows good sensitivity (0.84). However, it struggles with precision (0.33), meaning a relatively small proportion of the predicted failure cases are actual failures. This imbalance results in a moderate F1 score of 0.47.
- Random forest demonstrates strong sensitivity (0.89), meaning it effectively detects failure cases. However, it struggles with precision (0.38), indicating that only a relatively small portion of the firms predicted as failures are actually failures. This results in a moderate F1 score of (0.53). On the other hand, when using SMOTE, it records (0.80) for sensitivity and loses much of its success rate for identifying default firms. However, precision got better ((0.60) and (0.69)).
- Gradient boosting offers balanced performance, with a sensitivity of (0.80) and higher precision (0.44), resulting in an F1 score of (0.57). This indicates better overall handling of both false positives and false negatives.
- XGBoost performs similarly to gradient boosting, with the same sensitivity (0.80) but slightly lower precision (0.39), resulting in an F1 score of (0.53). While still robust, it is slightly outperformed by gradient boosting in terms of precision.
- AdaBoost has the lowest sensitivity (0.75) but the highest precision (0.48), resulting in a competitive F1 score of (0.58). This indicates that while its failure predictions are more accurate, it misses some failure cases.
- CatBoost exhibits the highest sensitivity (0.91) but struggles the most with precision (0.20), leading to the weakest F1 score (0.33). This suggests that while CatBoost is highly effective at detecting failures, which is our primary objective, it produces more false positives.

❖ SMOTE Results:

- Multi-Layer Perceptron (MLP) maintains a similar performance pattern. Sensitivity remains high at 0.84, effectively capturing failure cases, while precision stays relatively low at 0.33, indicating that many predicted failure cases were not actual failures.

- Random forest sensitivity drops to 0.80 while precision improves to 0.60, leading to an F1 score of 0.69. However, the sensitivity reduction indicates some missed failure cases.
- Gradient boosting shows lower sensitivity (0.73) with a slight precision increase (0.60), resulting in an F1 score of 0.66, suggesting a modest trade-off.
- XGBoost sees a minor decrease in sensitivity (0.77) and an increase in precision (0.55), with an F1 score of 0.65.
- AdaBoost under SMOTE shows a significant drop in sensitivity (0.57) with minimal gain in precision (0.56), reducing its F1 score to 0.56.
- CatBoost improves precision (0.72) but its sensitivity remains lower than CorrOV-CSEn, with an F1 score of 0.73, showing more balanced results but still lower sensitivity.

These findings reveal that CatBoost reached the highest sensitivity, which is followed by random forest, Multi-Layer Perceptron (MLP), gradient boosting, XGBoost, and AdaBoost. On the other hand, CatBoost and random forest, despite their high sensitivity, achieve relatively poor precision and overall effectiveness.

When the SMOTE method is used, XGBoost records the highest sensitivity, followed by random forest, gradient boosting, CatBoost, and AdaBoost. Meanwhile, CatBoost has the best precision and F1 score.

CatBoost emerges as the strongest model in terms of sensitivity when combined with CorrOV-CSEn. This is primarily due to the features of CorrOV-CSEn, where the augmented data are generated based on correlations, leading to less noisy data being fed into the model. Additionally, the minority class receives more weight automatically, which is essential in imbalanced datasets. CatBoost, being highly adaptable to weighted data, can effectively handle the imbalance and emphasize the minority class.

Furthermore, CatBoost uses a gradient boosting framework with decision trees, leveraging the powerful combination of categorical feature processing and boosting to handle the weight distributions more efficiently. For recall specifically, CorrOV-CSEn generates data that clarifies the boundary between classes, reducing overlap and thus improving recall. This characteristic is particularly beneficial for models like CatBoost, which are well-equipped to learn from complex relationships in the data, including those between features that are more strongly correlated with default cases.

5.2. Significance Differences

For a more detailed comparison of our models, we divided the dataset into four subsets. The performance across these subsets reveals notable variations, highlighting the models' differing strengths and weaknesses in handling imbalanced data. Table 7 describes the performance of machine learning models across four datasets.

CatBoost achieves high sensitivity, particularly in Dataset-I (1.00) and Dataset-IV (1.00). It also performs reasonably well in Dataset-II (0.86) and Dataset-III (0.89), indicating its effectiveness in identifying positive cases. Gradient boosting and XGBoost demonstrate the highest and most consistent sensitivity across all datasets, both achieving perfect sensitivity (1.00) in Dataset-I and Dataset-IV. However, they experience moderate drops in Dataset-II (0.57 and 0.71, respectively) and Dataset-III (0.56 and 0.67, respectively). Random forest shows varied sensitivity, excelling in Dataset-I (0.95) and Dataset-IV (0.88) but dropping significantly in Dataset-II (0.71) and Dataset-III (0.67). The performance of Multi-Layer Perceptron (MLP), similar to random forest, varies significantly, ranging from 0.84 in Dataset-II to 0.67 in Dataset-IV. AdaBoost struggles more with sensitivity, particularly in Dataset-II (0.43) and Dataset-III (0.56), though it performs well in Dataset-I (0.80) and Dataset-IV (0.88).

Table 7. Performance comparison of machine learning models across four datasets.

	Dataset-I			Dataset-II		
Model	**Sensitivity**	**Precision**	**F1 Score**	**Sensitivity**	**Precision**	**F1 Score**
Multi-Layer Perceptron (MLP)	0.693	0.455	0.550	0.844	0.371	0.516
Random Forest	0.950	0.593	0.731	0.714	0.192	0.303
Gradient Boosting	1.000	0.666	0.800	0.571	0.500	0.533
XGBoost	1.000	0.606	0.755	0.714	0.385	0.500
AdaBoost	0.800	0.640	0.711	0.429	0.429	0.429
CatBoost	1.000	0.339	0.506	0.857	0.188	0.308
	Dataset-III			**Dataset-IV**		
Model	**Sensitivity**	**Precision**	**F1 Score**	**Sensitivity**	**Precision**	**F1 Score**
Multi-Layer Perceptron (MLP)	0.773	0.370	0.500	0.670	0.451	0.540
Random Forest	0.666	0.240	0.353	0.875	0.368	0.519
Gradient Boosting	0.556	0.227	0.323	1.000	0.444	0.615
XGBoost	0.667	0.300	0.414	1.000	0.333	0.500
AdaBoost	0.556	0.313	0.4	0.875	0.389	0.538
CatBoost	0.889	0.138	0.239	1.000	0.116	0.208

Gradient boosting delivers solid precision across all datasets, particularly in Dataset-I (0.67) and Dataset-IV (0.44). XGBoost also performs well in terms of precision, especially in Dataset-I (0.61), but suffers slightly in Dataset-II (0.38) and Dataset-IV (0.33), indicating a higher number of false positives in these datasets. Multi-Layer Perceptron (MLP) achieves a more stable performance, with scores ranging from (0.37) to (0.45) across the four datasets. Random forest shows a wide range of precision, performing strongly in Dataset-I (0.59) but struggling significantly in Dataset-II (0.19), Dataset-III (0.24), and Dataset-IV (0.37). This suggests that while random forest captures positive cases well, it is prone to misclassifying negative cases as positive. CatBoost exhibits the weakest precision across all datasets, with values of (0.34) in Dataset-I, (0.19) in Dataset-II, (0.14) in Dataset-III, and (0.12) in Dataset-IV, indicating consistent difficulty in accurately classifying failure cases and a higher rate of false positives. AdaBoost generally maintains moderate precision, performing best in Dataset-I (0.64) but falling to (0.43) in Dataset-II, with consistent but lower results in Dataset-III and Dataset-IV.

Gradient boosting achieves the highest and most consistent F1 scores, particularly in Dataset-I (0.80) and Dataset-IV (0.62). XGBoost also performs well, especially in Dataset-I (0.75), with solid F1 scores in Dataset-III (0.41) and Dataset-IV (0.50). However, its F1 score drops slightly in Dataset-II (0.50). Random forest delivers strong performance in Dataset-I (0.73) and Dataset-IV (0.52), but its lower F1 scores in Dataset-II (0.30) and Dataset-III (0.35) highlight its susceptibility to imbalanced class distributions, especially where precision is low. The Multi-Layer Perceptron (MLP) achieves stable performance, with scores consistently around (0.50). AdaBoost performs moderately well, with peak F1 scores in Dataset-I (0.71) and Dataset-IV (0.54), but faces challenges in Dataset-II (0.43) and Dataset-III (0.40). Despite its high sensitivity, CatBoost suffers the most in terms of F1 score due to poor precision, which may need tuning for scenarios where precision is more critical. Its F1 scores are (0.51) in Dataset-I, (0.31) in Dataset-II, and (0.21) in Dataset-IV.

We also used the Friedman–Nemenyi test to detect significant differences among the models. Table 8 shows the results of the Friedman–Nemenyi test for each of the three scores.

Table 8. Friedman test results for comparisons among machine learning models.

Precision						
Friedman Test Statistic	12.00					
p-value	0.03479					
	Random Forest	Multi-Layer Perceptron (MLP)	Gradient Boosting	XGBoost	AdaBoost	CatBoost
Random Forest	-	0.854075	0.635776	0.900000	0.635776	0.744925
Multi-Layer Perceptron (MLP)	0.854075	-	0.900000	0.900000	0.900000	0.136905
Gradient Boosting	0.635776	0.900000	-	0.900000	0.900000	0.052161
XGBoost	0.900000	0.900000	0.900000	-	0.900000	0.410222
AdaBoost	0.635776	0.900000	0.900000	0.900000	-	0.052161
CatBoost	0.744925	0.136905	0.052161	0.410222	0.052161	-
Sensitivity						
Friedman Test Statistic	10.04					
p-value [1]	0.07413					
No significant difference was found by the Friedman test because the p-value is greater than the significance level of 0.05.						
F1 Score						
Friedman Test Statistic	10.43					
p-value [2]	0.06396					

[1,2] No significant difference was found by the Friedman test because the p-value is greater than the significance level of 0.05.

Since the p-value is less than 0.05, the Friedman test indicates a significant difference in sensitivity and precision across the models:

- AdaBoost vs. CatBoost: This is the only comparison with a significant difference (p-value = 0.030), showing that CatBoost performs significantly better than AdaBoost in terms of sensitivity.
- Gradient boosting vs. CatBoost, AdaBoost vs. CatBoost, and MLP vs. CatBoost: All comparisons show significant differences with p-values of 0.030, indicating that CatBoost has significantly lower precision compared to gradient boosting, AdaBoost, and MLP.

All other comparisons have p-values above 0.05, indicating no significant differences in sensitivity and precision between these models.

5.3. Feature Importance

In the final stage, we present the importance of our feature set across the models used. Figure 3 illustrates the feature importance in our models. It is clear that X1 has the highest importance in all models expect MLP. This contrasts with other credit risk studies using the same feature set in the Iranian capital market [64].

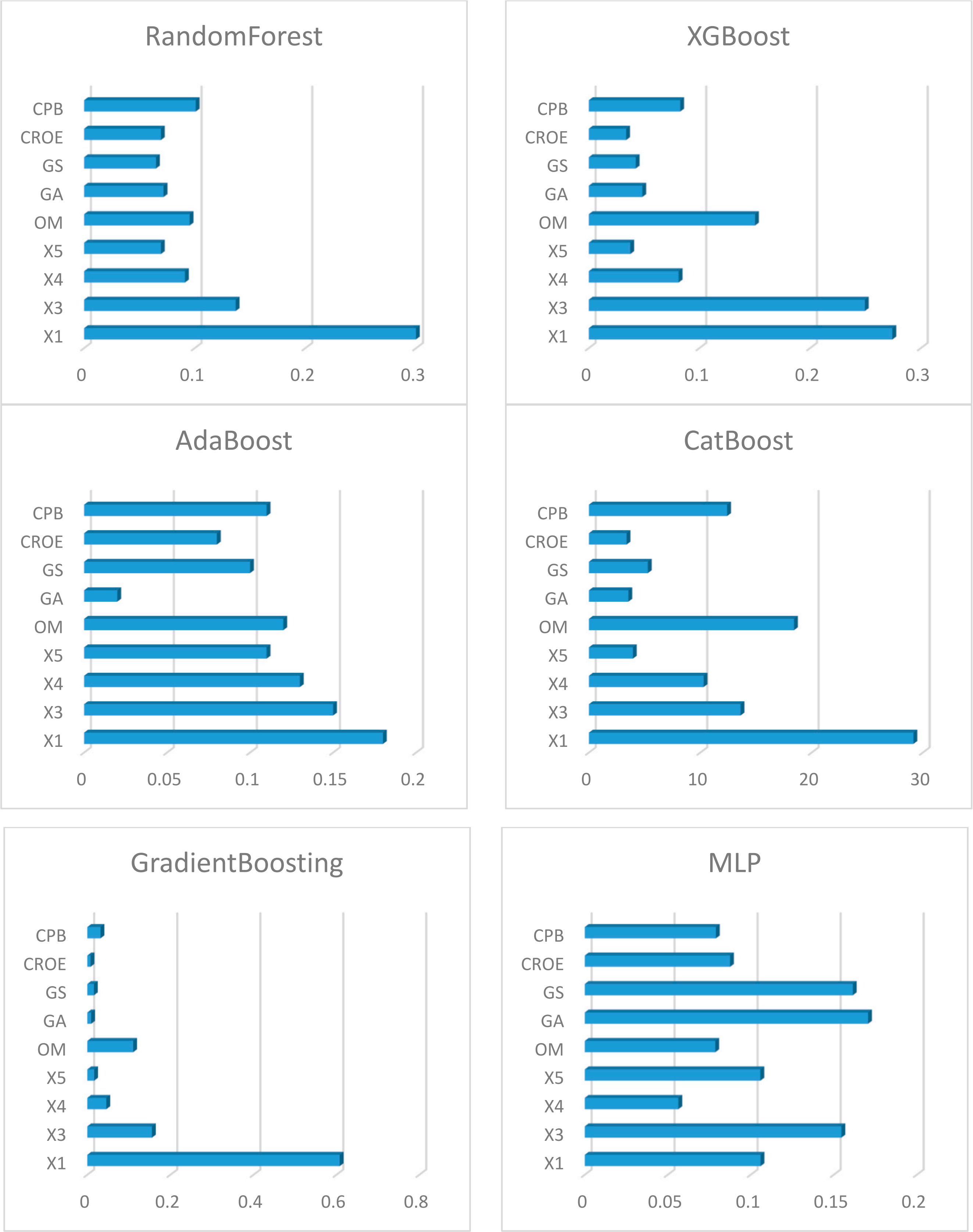

Figure 3. Feature importance in our machine learning models.

6. Conclusions

In this study, we employed recently introduced cost-sensitive methods to predict business failures in the Iranian capital market using five decision tree-based algorithms in addition to MPL. Our findings demonstrate that all models achieved improved sensitivity scores through this approach, with CatBoost outperforming the others.

While CatBoost showed clear superiority, there remains a tradeoff between extending credit to a broader range of customers to maximize revenue and minimizing the risk of default. Future research could focus on developing models that optimize creditor profits by balancing revenue generation with risk management rather than solely assessing default risk.

Additionally, other decision tree-based methods, such as Mondrian Forest, could be explored in this context. In addition to the models evaluated in this study, it is important to consider the role of hyperparameter optimization in improving model performance. While our current work focuses on assessing the effectiveness of various decision tree-based models, incorporating optimization techniques such as grid search or Bayesian optimization could lead to even better-performing models.

From a data perspective, incorporating new types of data, including sentiment analysis, textual data, and political indices, could significantly enhance model performance. This is especially relevant in countries like Iran, where political and economic conditions play a crucial role in credit risk management.

Our research focused on the Iran capital market, and due to the unique economic and political challenges facing the Iranian capital market, these findings might not exactly apply to other industries or nations, although many developing countries face similar challenges, like extensive governmental administration, challenges related to market efficiency, and regulatory frameworks and political instability. It is recommended to consider actual default instead of failure under Article 141 of the Amended Commercial Code.

Further, it is important to notice that the data analysis results may be affected by the global economic meltdown caused by the pandemic during the window period. Therefore, in the upcoming research, it is potential to conduct a sensitivity analysis to compare the results with the exclusion of the COVID-19 period.

Lastly, there is considerable potential in applying these methods to emerging fields, such as peer-to-peer (P2P) lending platforms, which have been growing rapidly in Iran in recent years.

Author Contributions: Conceptualization, P.P., M.P.F., C.T., M.S. and H.K.; methodology, P.P., M.P.F., C.T., M.S. and H.K.; software, P.P. and H.K.; validation, P.P., M.P.F., C.T. and H.K.; formal analysis, P.P., C.T., M.S. and H.K.; investigation, P.P., M.P.F., C.T. and M.S.; resources, P.P., M.P.F., C.T. and M.S.; data curation, M.P.F., M.S. and H.K.; writing—original draft preparation, P.P. and H.K.; writing—review and editing, P.P., M.P.F., C.T., M.S. and H.K.; visualization, P.P., M.S. and H.K.; supervision, P.P., M.P.F., C.T. and M.S.; project administration, P.P., C.T. and M.P.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank the anonymous reviewers and the editor-in-chief for their constructive comments and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Usmani, S.; Shamsi, J.A. LSTM based stock prediction using weighted and categorized financial news. *PLoS ONE* **2023**, *18*, e0282234. [CrossRef] [PubMed]
2. Zhang, Z.; Liu, X.; Niu, H. Financial crisis early warning of Chinese listed companies based on MD&A text-linguistic feature indicators. *PLoS ONE* **2023**, *18*, e0291818. [CrossRef]
3. Jezeie, F.V.; Sadjadi, S.J.; Makui, A. Constrained portfolio optimization with discrete variables: An algorithmic method based on dynamic programming. *PLoS ONE* **2022**, *17*, e0271811. [CrossRef] [PubMed]
4. Bi, W.; Zhang, Q. Forecasting mergers and acquisitions failure based on partial-sigmoid neural network and feature selec-tion. *PLoS ONE* **2021**, *16*, e0259575. [CrossRef]
5. Li, M. Financial investment risk prediction under the application of information interaction Firefly Algorithm combined with Graph Convolutional Network. *PLoS ONE* **2023**, *18*, e0291510. [CrossRef]
6. Dahal, K.R.; Pokhrel, N.R.; Gaire, S.; Mahatara, S.; Joshi, R.P.; Gupta, A.; Banjade, H.R.; Joshi, J. A comparative study on effect of news sentiment on stock price prediction with deep learning architecture. *PLoS ONE* **2023**, *18*, e0284695. [CrossRef] [PubMed]
7. Javid, I.; Ghazali, R.; Syed, I.; Zulqarnain, M.; Husaini, N.A. Study on the Pakistan stock market using a new stock crisis prediction method. *PLoS ONE* **2022**, *17*, e0275022. [CrossRef]
8. Cui, Y.; Liu, L. Investor sentiment-aware prediction model for P2P lending indicators based on LSTM. *PLoS ONE* **2022**, *17*, e0262539. [CrossRef]
9. Zhu, C.; Liu, X.; Chen, D. Prediction of digital transformation of manufacturing industry based on interpretable machine learning. *PLoS ONE* **2024**, *19*, e0299147. [CrossRef] [PubMed]
10. Khan, A.H.; Shah, A.; Ali, A.; Shahid, R.; Zahid, Z.U.; Sharif, M.U.; Jan, T.; Zafar, M.H. A performance comparison of machine learning models for stock market prediction with novel investment strategy. *PLoS ONE* **2023**, *18*, e0286362. [CrossRef] [PubMed]
11. Wei, X.; Ouyang, H.; Liu, M. Stock index trend prediction based on TabNet feature selection and long short-term memory. *PLoS ONE* **2022**, *17*, e0269195. [CrossRef] [PubMed]
12. Tran, T.; Nguyen, N.H.; Le, B.T.; Vu, N.T.; Vo, D.H. Examining financial distress of the Vietnamese listed firms using accounting-based models. *PLoS ONE* **2023**, *18*, e0284451. [CrossRef] [PubMed]
13. Laghari, F.; Ahmed, F.; López García, M.D.L.N. Cash flow management and its effect on firm performance: Empirical ev-idence on non-financial firms of China. *PLoS ONE* **2023**, *18*, e0287135. [CrossRef] [PubMed]
14. Almustafa, H.; Nguyen, Q.K.; Liu, J.; Dang, V.C. The impact of COVID-19 on firm risk and performance in MENA countries: Does national governance quality matter? *PLoS ONE* **2023**, *18*, e0281148. [CrossRef] [PubMed]
15. Tian, X.; Wang, Y.; Kohar, U.H.A. Capital structure, business model innovation, and firm performance: Evidence from Chinese listed corporate based on system GMM model. *PLoS ONE* **2024**, *19*, e0306054. [CrossRef]
16. Samour, A.; AlGhazali, A.; Gadoiu, M.; Banuta, M. Capital structure and financial performance of China's energy industry: What can we infer from COVID-19? *PLoS ONE* **2024**, *19*, e0300936.
17. Berloco, C.; Morales, G.D.F.; Frassineti, D.; Greco, G.; Kumarasinghe, H.; Lamieri, M.; Massaro, E.; Miola, A.; Yang, S. Predicting corporate credit risk: Network contagion via trade credit. *PLoS ONE* **2021**, *16*, e0250115. [CrossRef] [PubMed]
18. Hlongwane, R.; Ramaboa, K.K.K.M.; Mongwe, W. Enhancing credit scoring accuracy with a comprehensive evaluation of alternative data. *PLoS ONE* **2024**, *19*, e0303566. [CrossRef] [PubMed]
19. Ma, Z.; Hou, W.; Zhang, D. A credit risk assessment model of borrowers in P2P lending based on BP neural network. *PLoS ONE* **2021**, *16*, e0255216. [CrossRef] [PubMed]
20. Wang, H.; Liu, X. Undersampling bankruptcy prediction: Taiwan bankruptcy data. *PLoS ONE* **2021**, *16*, e0254030. [CrossRef] [PubMed]
21. Japkowicz, N. Learning from imbalanced data sets: A comparison of various strategies. In *AAAI Workshop on Learning from Imbalanced Data Sets*; AAAI Press: Menlo Park, CA, USA, 2000.
22. Groccia, M.C.; Guido, R.; Conforti, D.; Pelaia, C.; Armentaro, G.; Toscani, A.F.; Miceli, S.; Succurro, E.; Hribal, M.L.; Sciacqua, A. Cost-Sensitive Models to Predict Risk of Cardiovascular Events in Patients with Chronic Heart Failure. *Information* **2023**, *14*, 542. [CrossRef]
23. Natha, P.; RajaRajeswari, P. Advancing Skin Cancer Prediction Using Ensemble Models. *Computers* **2024**, *13*, 157. [CrossRef]
24. Devi, D.; Biswas, S.K.; Purkayastha, B. Correlation-based Oversampling aided Cost Sensitive Ensemble learning technique for Treatment of Class Imbalance. *J. Exp. Theor. Artif. Intell.* **2022**, *34*, 143–174. [CrossRef]
25. Alloway, B.T.; Weisenthal, J. *What's Been Happening with the Iranian Stock Market*; Bloomberg: New York, NY, USA, 2023.
26. Rawat, S.S.; Mishra, A.K. Review of Methods for Handling Class-Imbalanced in Classification Problems. *arXiv* **2022**, arXiv:2211.05456.
27. Tomek, I. Two Modifications of CNN. *IEEE Trans. Syst. Man Cybern.* **1976**, *11*, 769–772.
28. Kubat, M.; Matwin, S. Addressing the curse of imbalanced data sets: One-sided sampling. In Proceedings of the Fourteenth International Conference on Machine Learning, Nashville, TN, USA, 8–12 July 1997.

29. Japkowicz, N. The class imbalance problem: Significance and strategies. In Proceedings of the International Conference on Artificial Intelligence, Las Vegas, NV, USA, 26–29 June 2000.
30. Laurikkala, J. Improving identification of difficult small classes by balancing class distribution. In Proceedings of the Artificial Intelligence in Medicine: 8th Conference on Artificial Intelligence in Medicine in Europe, AIME 2001, Cascais, Portugal, 1–4 July 2001; Proceedings 8. Springer: Berlin/Heidelberg, Germany, 2001.
31. Hoyos-Osorio, J.; Alvarez-Meza, A.; Daza-Santacoloma, G.; Orozco-Gutierrez, A.; Castellanos-Dominguez, G. Relevant information undersampling to support imbalanced data classification. *Neurocomputing* **2021**, *436*, 136–146. [CrossRef]
32. Lee, W.; Seo, K. Downsampling for Binary Classification with a Highly Imbalanced Dataset Using Active Learning. *Big Data Res.* **2022**, *28*, 100314. [CrossRef]
33. Laveti, R.N.; Mane, A.A.; Pal, S.N. Dynamic Stacked Ensemble with Entropy based Undersampling for the Detection of Fraudulent Transactions. In Proceedings of the 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, 2–4 April 2021; pp. 1–7.
34. Solberg, A.S.; Solberg, R. A large-scale evaluation of features for automatic detection of oil spills in ERS SAR images. In Proceedings of the IGARSS '96. 1996 International Geoscience and Remote Sensing Symposium, Lincoln, NB, USA, 21–26 May 1996; pp. 1484–1486.
35. Mathew, J.; Pang, C.K.; Luo, M.; Leong, W.H. Classification of Imbalanced Data by Oversampling in Kernel Space of Support Vector Machines. *IEEE Trans. Neural Networks Learn. Syst.* **2017**, *29*, 4065–4076. [CrossRef] [PubMed]
36. Bennin, K.E.; Keung, J.; Phannachitta, P.; Monden, A.; Mensah, S. MAHAKIL: Diversity Based Oversampling Approach to Alleviate the Class Imbalance Issue in Software Defect Prediction. *IEEE Trans. Softw. Eng.* **2017**, *44*, 534–550. [CrossRef]
37. Cheng, K.; Zhang, C.; Yu, H.; Yang, X.; Zou, H.; Gao, S. Grouped SMOTE With Noise Filtering Mechanism for Classifying Imbalanced Data. *IEEE Access* **2019**, *7*, 170668–170681. [CrossRef]
38. Tarawneh, A.S.; Hassanat, A.B.A.; Almohammadi, K.; Chetverikov, D.; Bellinger, C. SMOTEFUNA: Synthetic Minority Over-Sampling Technique Based on Furthest Neighbour Algorithm. *IEEE Access* **2020**, *8*, 59069–59082. [CrossRef]
39. Almomani, I.; Qaddoura, R.; Habib, M.; Alsoghyer, S.; Al Khayer, A.; Aljarah, I.; Faris, H. Android ransomware detection based on a hybrid evolutionary approach in the context of highly im-balanced data. *IEEE Access* **2021**, *9*, 57674–57691. [CrossRef]
40. Juez-Gil, M.; Arnaiz-González, Á.; Rodríguez, J.J.; López-Nozal, C.; García-Osorio, C. Approx-SMOTE: Fast SMOTE for Big Data on Apache Spark. *Neurocomputing* **2021**, *464*, 432–437. [CrossRef]
41. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
42. Li, C. *Data Mining for Direct Marketing: Problems and Solutions*; National Library of Canada= Bibliothèque nationale du Canada: Ottawa, ON, Canada, 2000.
43. Xu, Z.; Shen, D.; Nie, T.; Kou, Y. A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data. *J. Biomed. Informatics* **2020**, *107*, 103465. [CrossRef]
44. Dong; Xiao, H.; Dong, Y. SA-CGAN: An oversampling method based on single attribute guided conditional GAN for multi-class imbalanced learning. *Neurocomputing* **2022**, *472*, 326–337. [CrossRef]
45. Sharma, A.; Singh, P.K.; Chandra, R. SMOTified-GAN for Class Imbalanced Pattern Classification Problems. *IEEE Access* **2022**, *10*, 30655–30665. [CrossRef]
46. Puri, A.; Gupta, M.K. Improved Hybrid Bag-Boost Ensemble With K-Means-SMOTE–ENN Technique for Handling Noisy Class Imbalanced Data. *Comput. J.* **2021**, *65*, 124–138. [CrossRef]
47. Seiffert, C.; Khoshgoftaar, T.M.; Van Hulse, J.; Napolitano, A. RUSBoost: A Hybrid Approach to Alleviating Class Imbalance. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2009**, *40*, 185–197. [CrossRef]
48. Czarnowski, I. Weighted Ensemble with one-class Classification and Over-sampling and Instance selection (WECOI): An approach for learning from imbalanced data streams. *J. Comput. Sci.* **2022**, *61*, 101614. [CrossRef]
49. Wang, H.; Xu, Q.; Zhou, L. Large Unbalanced Credit Scoring Using Lasso-Logistic Regression Ensemble. *PLoS ONE* **2015**, *10*, e0117844. [CrossRef] [PubMed]
50. Ariza-Garzón, M.-J.; Arroyo, J.; Segovia-Vargas, M.-J.; Caparrini, A. Profit-sensitive machine learning classification with explanations in credit risk: The case of small businesses in peer-to-peer lending. *Electron. Commer. Res. Appl.* **2024**, *67*, 101428. [CrossRef]
51. Turney, P.D. Cost-Sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm. *J. Artif. Intell. Res.* **1994**, *2*, 369–409. [CrossRef]
52. Ling, C.X.; Yang, Q.; Wang, J.; Zhang, S. Decision trees with minimal costs. In Proceedings of the Twenty-First International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004.
53. Drummond, C.; Holte, R.C. Exploiting the cost (in) sensitivity of decision tree splitting criteria. In Proceedings of the International Conference on Machine Learning, Stanford, CA, USA, 29 June 29–2 July 2000.

54. Domingos, P. Metacost: A general method for making classifiers cost-sensitive. In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 15–18 August 1999.
55. Witten, I.H.; Frank, E. Data mining: Practical machine learning tools and techniques with Java implementations. *Acm Sigmod Rec.* **2002**, *31*, 76–77. [CrossRef]
56. Chai, X.; Deng, L.; Yang, Q.; Ling, C.X. Test-cost sensitive naive bayes classification. In Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04), Brighton, UK, 1–4 November 2004; IEEE: Piscataway, NJ, USA.
57. Sheng, V.S.; Ling, C.X. Thresholding for making classifiers cost-sensitive. In Proceedings of the Association for the Advancement of Artificial Intelligence, Boston, MA, USA, 16–20 July 2006.
58. Khan, S.H.; Hayat, M.; Bennamoun, M.; Sohel, F.A.; Togneri, R. Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 3573–3587. [CrossRef] [PubMed]
59. Lu, H.; Xu, Y.; Ye, M.; Yan, K.; Gao, Z.; Jin, Q. Learning misclassification costs for imbalanced classification on gene expression data. *BMC Bioinform.* **2019**, *20*, 1–10. [CrossRef] [PubMed]
60. Feng, F.; Li, K.C.; Shen, J.; Zhou, Q.; Yang, X. Using cost-sensitive learning and feature selection algorithms to improve the performance of imbalanced clas-sification. *IEEE Access* **2020**, *8*, 69979–69996. [CrossRef]
61. Khandani, A.E.; Kim, A.J.; Lo, A.W. Consumer credit-risk models via machine-learning algorithms. *J. Bank. Financ.* **2010**, *34*, 2767–2787. [CrossRef]
62. Barboza, F.; Kimura, H.; Altman, E. Machine learning models and bankruptcy prediction. *Expert Syst. Appl.* **2017**, *83*, 405–417. [CrossRef]
63. Yıldırım, M.; Okay, F.Y.; Özdemir, S. Big data analytics for default prediction using graph theory. *Expert Syst. Appl.* **2021**, *176*, 114840. [CrossRef]
64. Peykani, P.; Sargolzaei, M.; Sanadgol, N.; Takaloo, A.; Kamyabfar, H. The application of structural and machine learning models to predict the default risk of listed companies in the Iranian capital market. *PLoS ONE* **2023**, *18*, e0292081. [CrossRef] [PubMed]
65. Chen, N.; Ribeiro, B. A consensus approach for combining multiple classifiers in cost-sensitive bankruptcy prediction. In Proceedings of the Adaptive and Natural Computing Algorithms: 11th International Conference, ICANNGA 2013, Lausanne, Switzerland, 4–6 April 2013; Proceedings 11. Springer: Berlin/Heidelberg, Germany, 2013.
66. Bahnsen, A.C.; Aouada, D.; Ottersten, B. Example-dependent cost-sensitive decision trees. *Expert Syst. Appl.* **2015**, *42*, 6609–6619. [CrossRef]
67. Zakaryazad, A.; Duman, E. A profit-driven Artificial Neural Network (ANN) with applications to fraud detection and direct marketing. *Neurocomputing* **2016**, *175*, 121–131. [CrossRef]
68. Xia, Y.; Liu, C.; Liu, N. Cost-sensitive boosted tree for loan evaluation in peer-to-peer lending. *Electron. Commer. Res. Appl.* **2017**, *24*, 30–49. [CrossRef]
69. Fiore, U.; De Santis, A.; Perla, F.; Zanetti, P.; Palmieri, F. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Inf. Sci.* **2017**, *479*, 448–455. [CrossRef]
70. Papouskova, M.; Hajek, P. Two-stage consumer credit risk modelling using heterogeneous ensemble learning. *Decis. Support Syst.* **2019**, *118*, 33–45. [CrossRef]
71. De Bock, K.W.; Coussement, K.; Lessmann, S. Cost-sensitive business failure prediction when misclassification costs are uncertain: A heterogeneous ensemble selection approach. *Eur. J. Oper. Res.* **2020**, *285*, 612–630. [CrossRef]
72. Hou, W.-H.; Wang, X.-K.; Zhang, H.-Y.; Wang, J.-Q.; Li, L. A novel dynamic ensemble selection classifier for an imbalanced data set: An application for credit risk assessment. *Knowl.-Based Syst.* **2020**, *208*, 106462. [CrossRef]
73. Li, Z.; Zhang, J.; Yao, X.; Kou, G. How to identify early defaults in online lending: A cost-sensitive multi-layer learning framework. *Knowl.-Based Syst.* **2021**, *221*, 106963. [CrossRef]
74. Barbaglia, L.; Manzan, S.; Tosetti, E. Forecasting Loan Default in Europe with Machine Learning. *J. Financ. Econ.* **2021**, *21*, 569–596. [CrossRef]
75. Gramegna, A.; Giudici, P. SHAP and LIME: An Evaluation of Discriminative Power in Credit Risk. *Front. Artif. Intell.* **2021**, *4*, 752558. [CrossRef]
76. Zou, Y.; Gao, C.; Gao, H. Business Failure Prediction Based on a Cost-Sensitive Extreme Gradient Boosting Machine. *IEEE Access* **2022**, *10*, 42623–42639. [CrossRef]
77. Xing, J.; Chi, G.; Pan, A. Instance-dependent misclassification cost-sensitive learning for default prediction. *Res. Int. Bus. Financ.* **2024**, *69*, 102265. [CrossRef]
78. Wang, S.; Chi, G. Cost-sensitive stacking ensemble learning for company financial distress prediction. *Expert Syst. Appl.* **2024**, *255*, 124525. [CrossRef]
79. Correa Bahnsen, A.; Aouada, D.; Ottersten, B. Ensemble of Example-Dependent Cost-Sensitive Decision Trees. *arXiv* **2015**, arXiv:1505.04637.
80. Pandove, D.; Rani, R.; Goel, S. Local graph based correlation clustering. *Knowl.-Based Syst.* **2017**, *138*, 155–175. [CrossRef]

81. Freund, Y.; Schapire, R.E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
82. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
83. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
84. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; Association for Computing Machinery: San Francisco, Ca, USA; pp. 785–794.
85. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased boosting with categorical features. In Proceedings of the 32nd International Con-ference on Neural Information Processing Systems, Montréal, Canada, 3–8 December 2018; Curran Associates Inc.: Montréal, QC, Canada; pp. 6639–6649.
86. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
87. Kumar, V.; Kedam, N.; Sharma, K.V.; Mehta, D.J.; Caloiero, T. Advanced Machine Learning Techniques to Improve Hydrological Prediction: A Comparative Analysis of Streamflow Prediction Models. *Water* **2023**, *15*, 2572. [CrossRef]
88. Charoenwong, B.; Reddy, P. Using forensic analytics and machine learning to detect bribe payments in regime-switching environments: Evidence from the India demonetization. *PLoS ONE* **2022**, *17*, e0268965. [CrossRef] [PubMed]
89. Nandi, A.K.; Randhawa, K.K.; Chua, H.S.; Seera, M.; Lim, C.P. Credit card fraud detection using a hierarchical behavior-knowledge space model. *PLoS ONE* **2022**, *17*, e0260579. [CrossRef] [PubMed]
90. Carbo-Valverde, S.; Cuadros-Solas, P.; Rodríguez-Fernández, F. A machine learning approach to the digitalization of bank customers: Evidence from random and causal forests. *PLoS ONE* **2020**, *15*, e0240362. [CrossRef]
91. Hlongwane, R.; Ramabao, K.; Mongwe, W. A novel framework for enhancing transparency in credit scoring: Leveraging Shapley values for interpretable credit scorecards. *PLoS ONE* **2024**, *19*, e0308718. [CrossRef]
92. Quach, A.C. *A Extensions and Improvements to Random Forests for Classification*; Utah State University: Logan, Utah, 2017.
93. Wyrobek, J.; Kluza, K. Efficiency of Gradient Boosting Decision Trees Technique in Polish Companies' Bankruptcy Prediction. In Proceedings of the Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology–ISAT 2018: Part III, Wrocław, Poland, 16–18 September 2019; pp. 24–35.
94. Freund, Y. Boosting a Weak Learning Algorithm by Majority. *Inf. Comput.* **1995**, *121*, 256–285. [CrossRef]
95. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
96. Lu, M.; Hou, Q.; Qin, S.; Zhou, L.; Hua, D.; Wang, X.; Cheng, L. A Stacking Ensemble Model of Various Machine Learning Models for Daily Runoff Forecasting. *Water* **2023**, *15*, 1265. [CrossRef]
97. Ainan, U.H.; Por, L.Y.; Chen, Y.-L.; Yang, J.; Ku, C.S. Advancing Bankruptcy Forecasting with Hybrid Machine Learning Techniques: Insights from an Unbalanced Polish Dataset. *IEEE Access* **2024**, *12*, 1. [CrossRef]
98. Aiken, J.M.; De Bin, R.; Hjorth-Jensen, M.; Caballero, M.D. Predicting time to graduation at a large enrollment American university. *PLoS ONE* **2020**, *15*, e0242334. [CrossRef] [PubMed]
99. Du, H.; Lv, L.; Wang, H.; Guo, A. A novel method for detecting credit card fraud problems. *PLoS ONE* **2024**, *19*, e0294537. [CrossRef]
100. Jabeur, S.B.; Gharib, C.; Mefteh-Wali, S.; Arfi, W.B. CatBoost model and artificial intelligence techniques for corporate failure prediction. *Technol. Fore-Cast. Soc. Chang.* **2021**, *166*, 120658. [CrossRef]
101. Dorogush, A.V.; Ershov, V.; Gulin, A. CatBoost: Gradient boosting with categorical features support. *arXiv* **2018**, arXiv:1810.11363.
102. Lu, H.; Hu, X. Enhancing Financial Risk Prediction for Listed Companies: A Catboost-Based Ensemble Learning Approach. *J. Knowl. Econ.* **2023**, *15*, 1–17. [CrossRef]
103. Enkhtuya, T.; Kang, D.K. Bankruptcy Prediction with Explainable Artificial Intelligence for Early-Stage Business Models. *Int. J. Internet Broadcast. Commun.* **2023**, *15*, 58–65.
104. Peykani, P.; Sargolzaei, M.; Botshekan, M.H.; Oprean-Stan, C.; Takaloo, A. Optimization of Asset and Liability Management of Banks with Minimum Possible Changes. *Mathematics* **2023**, *11*, 2761. [CrossRef]
105. Peykani, P.; Sargolzaei, M.; Takaloo, A.; Sanadgol, N. Investigating the monetary policy risk channel based on the dynamic stochastic general equilibrium model: Empirical evidence from Iran. *PLoS ONE* **2023**, *18*, e0291934. [CrossRef] [PubMed]
106. Marino, M.J. Chapter 3—Statistical Analysis in Preclinical Biomedical Research. In *Research in the Biomedical Sciences*; Williams, M., Curtis, M.J., Mullane, K., Eds.; Academic Press: Cambridge, MA, USA, 2018; pp. 107–144.
107. Riffenburgh, R.H. Chapter Summaries. In *Statistics in Medicine*, 2nd ed.; Riffenburgh, R.H., Ed.; Academic Press: Burlington, MA, USA, 2006; pp. 533–580.
108. Friedman, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [CrossRef]
109. Hull, J. *Machine Learning in Business: An Introduction to the World of Data Science*; Amazon Distribution: London, UK, 2020.

110. Altman, E.I. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *J. Financ.* **1968**, *23*, 589–609. [CrossRef]
111. Carton, R.B.; Hofer, C.W. *Measuring Organizational Performance: Metrics for Entrepreneurship and Strategic Management Research*; Edward Elgar Publishing: Cheltenham, UK, 2006.
112. Peykani, P.; Sargolzaei, M.; Takaloo, A.; Valizadeh, S. The Effects of Monetary Policy on Macroeconomic Variables through Credit and Balance Sheet Channels: A Dynamic Stochastic General Equilibrium Approach. *Sustainability* **2023**, *15*, 4409. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.