DOI https://doi.org/10.15589/znp2020.3(481).8
УДК 330.4+004.(4+6)

PERSONAL CREDIT RATING FORECASTING USING ML.NET

ПРОГНОЗУВАННЯ ОЦІНКИ КРЕДИТОСПРОМОЖНОСТІ ФІЗИЧНИХ ОСІБ ІЗ ВИКОРИСТАННЯМ МОЖЛИВОСТЕЙ ML.NET

Dmytro S. Antoniuk
dmitry_antonyuk@yahoo.com
ORCID: 0000-0001-7496-3553

Tetiana A. Vakaliuk
tetianavakaliuk@gmail.com
ORCID: 0000-0001-6825-4697

Galyna V. Marchuk
pzs_mgv@ztu.edu.ua
ORCID: 0000-0003-2954-1057

Vladyskav V. Didkivskyi
v.didkivskyi@sana-commerce.com
ORCID: 0000-0002-4615-7578

Д. С. Антонюк,
канд. пед. наук, доцент

Т. А. Вакалюк,
докт. пед. наук, доцент

Г. В. Марчук,

В. В. Дідківський,
магістр

Zhytomyr Polytechnic State University, Zhytomyr
Державний університет «Житомирська політехніка», м. Житомир

Abstract. Computer technologies with intellectual analysis and calculations are on the raise. This is caused by the flow of the new ideas and approaches, formed on the intersection of computer science, artificial intelligence, statistics, and databases. The rapid growth of the software that are using newest technologies for solving new types of the tasks and bring significant economic effects are happening now.

The usage of the machine learning techniques in the retail crediting provides financial institutions optimization of spending of up to 25%. The usage of machine learning for credit rating assessment and forecasting in "internet crediting" is especially relevant now. In the same time the analysis of the publications revealed the needed in the broader research in the area of economic and credit forecasting.

Purpose. The purpose of the work is the design and development of the system that assesses and forecasts credit rating of the persons based on the data analysis conducted.

Method. The next scientific methods of research were used in the work: abstraction, formalization, analysis and modelling. The object of the study is to forecast the assessment of the creditworthiness of individuals. The subject of the work is the possibilities of ML.NET platform to assess and forecast credit rating of the person.

Results. Execution of the credit rating assessment and forecasting of a person is the binary classification task that might be solved with the use of ML.NET – free machine learning platform for C# and F# programming languages. The open dataset of the banking institution with the database of anonymized transactions was used. Different kind of transactions were used to improve quality of assessment and forecasting.

Scientific novelty. The model has been built and assessment and forecasting of the credit rating of the people was completed in this work with the use of ML.NET platform.

Practical importance. The software application for the dataset formation and credit rating assessment and forecasting has been developed within the course of this work.

Key words: forecasting; assessment; person; credit; economic forecasting; machine learning; data science.

Анотація. Комп'ютерні технології з організацією інтелектуальних обчислень переживають свій розквіт. Насамперед це пов'язано з потоком нових ідей, які виходять з галузі комп'ютерних наук, що утворилася на перетині штучного інтелекту, статистики та теорії баз даних. Нині відбувається стрімке зростання кількості програмних продуктів, що використовують нові технології, а також типів задач, де їх застосування надає значного економічного ефекту.

Використання машинного навчання для оцінки фінансових ризиків у споживчому кредитуванні забезпечує фінансовим установам економію коштів до 25%. Особливої актуальності набуває використання засобів машинного навчання для

оцінки ризиків кредитоспроможності у проектах «Інтернет фінансів». Проте аналіз останніх досліджень дозволив встановити, що проблемі економічного прогнозування у сфері кредитів приділено не досить уваги.
Метою роботи є розробка системи, яка на основі проведеного аналізу даних визначатиме кредитоспроможність фізичних осіб.
Методика. У процесі дослідження були використані такі методи дослідження: абстрагування, метод формалізації, аналіз, моделювання. Об'єктом дослідження є прогнозування оцінки кредитоспроможності фізичних осіб. Предметом дослідження є можливості застосування платформи ML.NET для прогнозування оцінки кредитоспроможності фізичних осіб.
Результати. Встановлено, що виконання прогнозування оцінки кредитоспроможності фізичних осіб – це задача бінарної класифікації, яку дозволяє вирішити ML.NET – безкоштовна програма машинного навчання для мов програмування C# і F#. У цій роботі за основу була взята база даних банку, який пропонував послуги приватним особам. Послуги включають управління рахунками, надання кредитів тощо.
Наукова новизна. У дослідженні була побудована модель та проведено прогнозування оцінки кредитоспроможності фізичних осіб з використанням можливостей ML.NET.
Практична значимість. Реалізовано програму для формування набору даних, проведено інтелектуальний аналіз даних з використанням можливостей ML.NET.
Ключові слова: прогнозування; оцінка; фізичні особи; кредит; економічне прогнозування; аналіз даних.

ПОСТАНОВКА ЗАДАЧІ

Комп'ютерні технології з організацією інтелектуальних обчислень переживають свій розквіт. Насамперед це пов'язано з потоком нових ідей, що виходять з галузі комп'ютерних наук, яка утворилася на перетині штучного інтелекту, статистики та теорії баз даних. Нині відбувається стрімке зростання кількості програмних продуктів, що використовують нові технології, а також типів задач, де їх застосування надає значного економічного ефекту. Одним із наслідків стрімкого розвитку сфери машинного навчання є розширення стандартних можливостей .NET новою платформою ML.NET, яка дозволяє створювати власні моделі машинного навчання та інтегрувати їх у .NET додатки.

Інтелектуальний аналіз даних – це обробка даних і виявлення у них моделей і тенденцій, що допомагають приймати рішення. Одним із результатів розвитку методів інтелектуального аналізу є можливість використання раніше зібраних даних для прийняття кращих рішень, наприклад, рішення щодо видачі кредиту.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Проблемам економічного прогнозування приділяли увагу багато науковців, зокрема Ю. Архангельський, В. Бесєдін, В. Вітлінський, В. Геєць, М. Михалевич, О. Черняк та інші.

Використання машинного навчання для оцінки фінансових ризиків у споживчому кредитуванні забезпечує фінансовим установам економію коштів до 25% [14]. Особливої актуальності набуває використання засобів машинного навчання для оцінки ризиків кредитоспроможності у проектах «Інтернет фінансів», які забезпечують оперативність і знижують витрати на обслуговування процесу видачі кредитів, але через мінімізацію контактів із потенційним клієнтом зменшують можливості якісної оцінки ризиків людиною-менеджером фінансової установи [15].

ВІДОКРЕМЛЕННЯ НЕ ВИРІШЕНИХ РАНІШЕ ЧАСТИН ЗАГАЛЬНОЇ ПРОБЛЕМИ

Аналіз останніх досліджень дозволив встановити, що проблемі економічного прогнозування у сфері кредитів приділено не досить уваги.

МЕТА ДОСЛІДЖЕННЯ

Метою роботи є розробка системи, яка на основі проведеного аналізу даних визначатиме кредитоспроможність фізичних осіб.

МЕТОДИ, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

У процесі дослідження були використані такі *методи* дослідження:

Абстрагування – це «визначення, відділення та виокремлення однієї якої-небудь істотної сторони, властивості, ознаки явища або предмета й відволікання від всіх інших сторін, властивостей» [16, с. 19].

Метод *формалізації* – це «представлення найрізноманітніших об'єктів шляхом відображення й зображення їхнього змісту і структури у знаковій формі за допомогою найрізноманітніших «штучних» мов, до яких належить мова математики, математичної логіки, хімії й інших наук» [16, с. 20].

Аналіз передбачає «роздроблення цілого на складові елементи, тобто виділення ознак предмету для вивчення їх окремо як частини єдиного цілого» [16, с. 24].

Моделювання – це «метод створення й дослідження моделі» [16, с. 29].

Об'єктом дослідження є прогнозування оцінки кредитоспроможності фізичних осіб.

Предметом дослідження є можливості застосування платформи ML.NET для прогнозування оцінки кредитоспроможності фізичних осіб.

ОСНОВНИЙ МАТЕРІАЛ

Виконання прогнозування оцінки кредитоспроможності фізичних осіб – це задача бінарної класифі-

кації. Бінарна класифікація – «клас задач класифікації елементів набору даних на дві групи на підставі правила класифікації. Важливим моментом бінарної класифікації є те, що два класи здебільшого не симетричні як за обсягом відмінних наборів даних із кожного класу, так і за наслідками помилкової класифікації» [1].

Задачу бінарної класифікації дозволяє вирішити ML.NET – безкоштовна програма машинного навчання для мов програмування C# і F#. ML.NET дозволяє додавати в додатки .NET можливості машинного навчання [2]. Це дозволяє отримувати автоматичні прогнози на основі доступних даних. В основі ML.NET лежить модель машинного навчання. Ця модель визначає кроки, які необхідно виконати для отримання прогнозів на основі вхідних даних. За допомогою ML.NET можна навчити користувацьку модель, вказавши відповідний алгоритм. Створену модель можна додати в додаток і використовувати її для отримання прогнозів.

ML.NET Model Builder використовує автоматизоване машинне навчання (AutoML) для вивчення різних алгоритмів і параметрів машинного навчання, щоб допомогти знайти той, який найкраще відповідає певному сценарію [3].

ML.NET самостійно під час генерування моделі аналізу намагається визначити найкращий алгоритм. Для бінарної класифікації використовуються такі алгоритми [4]: averaged perceptron; fast forest; fast tree; LBFGS logistic regression; LightGBM; LinearSVM; SDCA logistic regression; SGD calibrated; symbolic SGD logistic regression. Розглянемо їх більш детально.

Averaged perceptron – це алгоритм класифікації, який робить свої прогнози, знаходячи роздільний гіперплан. Наприклад, зі значеннями функції f0,f1,...,fD−1f0,f1,...,fD−1 передбачення дається шляхом визначення того, на яку сторону гіперплану впадає точка. Це те саме, що ознака зваженої суми феодур, тобто ∑D−1i=0(wi∗fi)+b, де w0,w1,...,wD−1 – ваги, обчислені алгоритмом, а b – це зміщення, обчислене алгоритмом [5].

Fast forest – непараметричні моделі, які виконують послідовність простих тестів на входах. Цей порядок прийняття рішення наближає їх до результатів, знайдених у навчальному наборі даних, вхідні дані яких були аналогічні екземпляру, який обробляється. Рішення приймається на кожному вузлі структури даних бінарних дерев на основі міри подібності, яка відображає кожен екземпляр рекурсивно через гілки дерева до тих пір, поки не буде досягнуто відповідного вузла листів і повернення висновку про вихід [6].

Fast tree – це ефективна реалізація алгоритму збільшення градієнта MART. Підвищення градієнта – це технологія машинного навчання для проблем регресії. Він будує кожне дерево регресії поетапно, використовуючи заздалегідь задану функцію втрат для вимірювання помилки для кожного кроку та виправляє її на наступному. Тож ця модель передбачення насправді є сукупністю слабших моделей прогнозування. При проблемах регресії прискорене будівництво послідовно будує серію таких дерев, а потім вибирає оптимальне дерево, використовуючи довільну диференційовану функцію втрат [7].

LBFGS logistic regression – це варіант лінійної моделі. На ньому зображено особливість вектора x∈Rn до скаляра через y(x)=wTx+b=∑nj=1wjxj+b, де b є навчальним відхиленням [8].

Light GBM – це реалізація з відкритим вихідним кодом дерева рішень, що збільшує градієнт [9].

Linear SVM– алгоритм, який знаходить гіперплан у просторі функцій для бінарної класифікації, вирішуючи задачу SVM [10].

SDCA logistic regression – сучасна методика оптимізації для опуклих цільових функцій [11].

SGD calibrated – одна із популярних процедур стохастичної оптимізації, яка може бути інтегрована в кілька завдань машинного навчання для досягнення найсучасніших показників. Цей тренер реалізує стохастичний градієнт Hogwild Gradient Descent для двійкової класифікації, яка підтримує багатопотоковість без блокування. Якщо пов'язана проблема оптимізації є рідкою, Stohastic Gradient Descent Hogwild досягає майже оптимальної швидкості конвергенції [12].

Symbolic SGD logistic regression – це алгоритм, який робить свої прогнози, знаходячи роздільну гіперплану [13].

У цій роботі за основу була взята база даних банку, який пропонував послуги приватним особам. Послуги включають управління рахунками, надання кредитів тощо. Режим доступу до даних: https://data.world/lpetrocelli/czech-financial-dataset-real-anonymized-transactions.

Банк хоче покращити свої послуги. Наприклад, менеджери банків мають лише розпливчасте уявлення, хто хороший клієнт (кому запропонувати якісь додаткові послуги), хто поганий клієнт (за ким уважно слідкувати, щоб мінімізувати втрати банку). На щастя, банк зберігає дані про своїх клієнтів, рахунки (транзакції протягом декількох місяців), вже надані позики, видані кредитні картки. Керівники банку сподіваються покращити розуміння клієнтів і шукають конкретних дій для покращення послуг.

Дані про клієнтів та їхні рахунки складаються з таких відносин:

1) облікові записи (4500 об'єктів у файлі "account.csv"), де кожен запис описує статичні характеристики облікового запису,

2) клієнти (5369 об'єктів у файлі "client.csv") – кожен запис описує характеристики клієнта,

3) диспозиція (5369 об'єктів у файлі "disp.csv") – кожен запис пов'язує разом клієнта з обліковим записом, тобто це відношення описує права клієнтів на управління рахунками,

4) замовлення (6471 об'єкт у файлі "order.csv") – кожен запис описує характеристики платіжного доручення,

5) транзакція (1 056 320 об'єктів у файлі "trans. csv") – кожен запис описує одну транзакцію в обліковому записі,

6) позики (682 об'єкти у файлі "loan.csv") – кожен запис описує позику, надану для певного рахунка,

7) кредитні картки (892 об'єкти у файлі "card. csv") – кожен запис описує кредитну картку, видану на рахунок,

8) відношення демографічних даних (77 об'єктів у файлі "district.csv") – кожен запис описує демографічні характеристики району.

Кожен рахунок має як статичні характеристики (наприклад, дату створення, адресу відділення), задані стосовно рахунку, так і динамічні характеристики (наприклад, дебетові або кредитовані платежі, залишки на рахунку), наведені у таблицях "Orders" і "Transactions".

Таблиця "Clients" описує характеристики осіб, які можуть маніпулювати рахунками. В одного клієнта може бути багато рахунків, кілька клієнтів можуть маніпулювати одним рахунком; клієнти та рахунки пов'язані між собою відношеннями, які описані у таблиці "Dispositions".

Таблиці "Loans" і "Cards" описують деякі послуги, які банк пропонує своїм клієнтам; на рахунок може бути видано багато кредитних карток, максимум одна позика може бути надана на рахунок. Таблиця "Districts" дає деяку загальнодоступну інформацію про регіони (наприклад, рівень безробіття), з цього можна отримати додаткову інформацію про клієнтів.

Для здійснення інтелектуального аналізу даних було створено консольний додаток .NET Core 3.1 і встановлено пакет NuGet для Microsoft.ML. Для побудови моделі аналізу було створено клас "AIAccount", який містить властивості, наведені у табл. 1.

Для генерації даних, які можуть бути використані в результаті ML.NET Model Builder було створено статичний клас "DataManager", який містить поля та методи, наведені у табл. 2.

У головному класі "Program" в методі "Main" було викликано методи "CreateDatabase" та "SplitDatabase" класу "DataManager" для того, щоб згенерувати дані для інтелектуального аналізу та розділити їх на дані для тренування моделі та дані для тестування моделі.

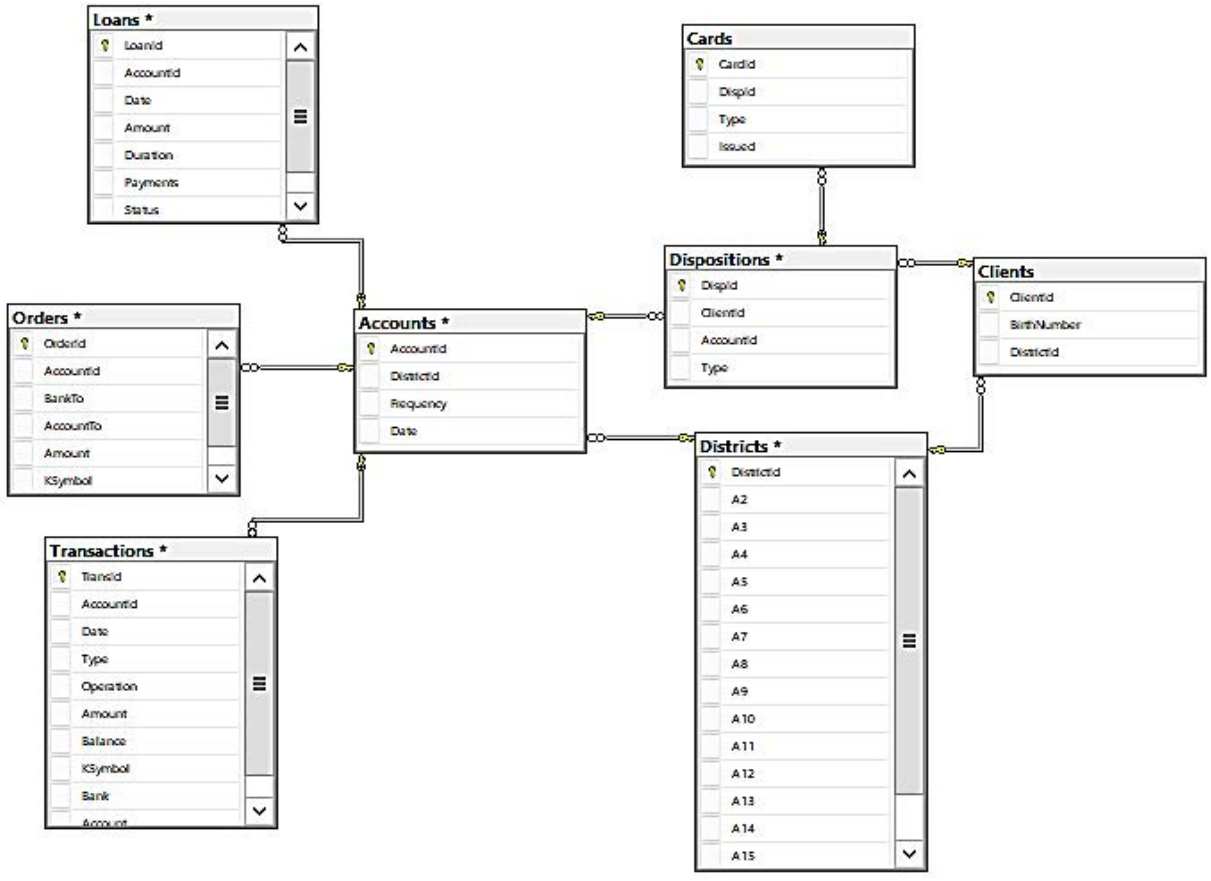

Рис. 1. Схема джерела даних

ОБГОВОРЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

У результаті роботи програми було опрацьовано дані 682 облікових записів – лише тих, за якими була оформлена хоча б одна позика. Згенеровані дані були збережені у файл із даними для тренування моделі – "TrainData.csv" і файл із даними для тестування моделі – "TestData.csv". Тепер можна здійснити інтелектуальний аналіз даних із використанням можливостей ML.NET ModelBuilder.

Вибравши сценарій і середовище тренування, ModelBuilder просить надати набір даних. Дані використовуються для навчання, оцінки та вибору найкращої моделі для цього сценарію. Наступний крок вимагає обрати файл із тренувальними даними, колонку для передбачення (в цьому випадку це "CanGetCredit") та колонки для аналізу.

Наступний етап – це власне тренування моделі, на основі якої далі можна здійснювати передбачення. На

Таблиця 1. Опис класу "AIAccount"

Назва	Тип	Опис
Id	int	Унікальний ідентифікатор аккаунту
FrequencyOfIssuanceOfStatements	string	Періодичність видачі заяв
HasCreditCard	bool	Значення, яке вказує на те, чи має клієнт кредитну карту
NumberOfInhabitantsInDistrict	int	Кількість жителів у регіоні, в якому проживає клієнт
NumberOfCitiesInDistrict	int	Кількість міст у регіоні, в якому проживає клієнт
AverageSalaryInDistrict	int	Середня зарплата у регіоні, в якому проживає клієнт
NumberOfOrders	int	Кількість замовлень
DebitedAmount	double	Заборгована сума за зроблені замовлення
NumberOfCreditTransactions	int	Кількість кредитних операцій
NumberOfWithdrawalTransactions	int	Кількість операцій виводу коштів
ActualAccountBalance	double	Фактичний баланс рахунку
LoanAmount	int	Розмір позики
LoanDuration	int	Тривалість позики
LoanMonthlyPayments	int	Щомісячні платежі та позики
CanGetCredit	bool	Значення, яке вказує на те, чи може клієнт отримати кредит

Таблиця 2. Опис класу "DataManager"

Назва	Тип	Опис
accounts	приватне поле	Містить дані з файлу "account.csv"
cards	приватне поле	Містить дані з файлу "card.csv"
dispositions	приватне поле	Містить дані з файлу "disp.csv"
districts	приватне поле	Містить дані з файлу "district.csv"
loans	приватне поле	Містить дані з файлу "loan.csv"
orders	приватне поле	Містить дані з файлу "order.csv"
transactions	приватне поле	Містить дані з файлу "trans.csv"
Create Database ()	публічний метод	Основний метод цього класу, який збирає дані з усіх файлів і генерує новий файл "AIAccounts.csv" з нормалізованими даними для подальшого використання ML.NET Model Builder
Split Database ()	публічний метод	Розділяє дані файлу "AIAccounts.csv" на тренувальні дані в файл "TrainData.csv" і дані для тестування у файл "TestData.csv"
GetAccountFrequency (Accountaccount)	приватний метод	Нормалізує дані колонки "frequency" з файлу "account.csv"
GetCreditTransactions(IList<Transaction>transactions)	приватний метод	Визначає кількість кредитних операцій
GetWithdrawalTransactions(IList<Transaction>transactions)	приватний метод	Визначає кількість операцій виводу коштів
GetBalance(IList<Transaction>transactions)	приватний метод	Визначає фактичний баланс рахунку
GetCreditCards(intaccountId)	приватний метод	Повертає дані про кредитні карти для вказаного аккаунту
GetDistricts(intaccountId)	приватний метод	Повертає дані про регіони для вказаного аккаунту
GetAccountOrders(intaccountId)	приватний метод	Повертає дані про замовлення для вказаного аккаунту
GetAccountTransactions(intaccountId)	приватний метод	Повертає дані про транзакції для вказаного аккаунту
GetRecords<T>(stringfilepath)	приватний метод	Читає записи із вказаного файлу
WriteRecords<T>(IEnumerable<T>records, stringfilepath)	приватний метод	Створює новий або перезаписує існуючий файл вказаними вище записами

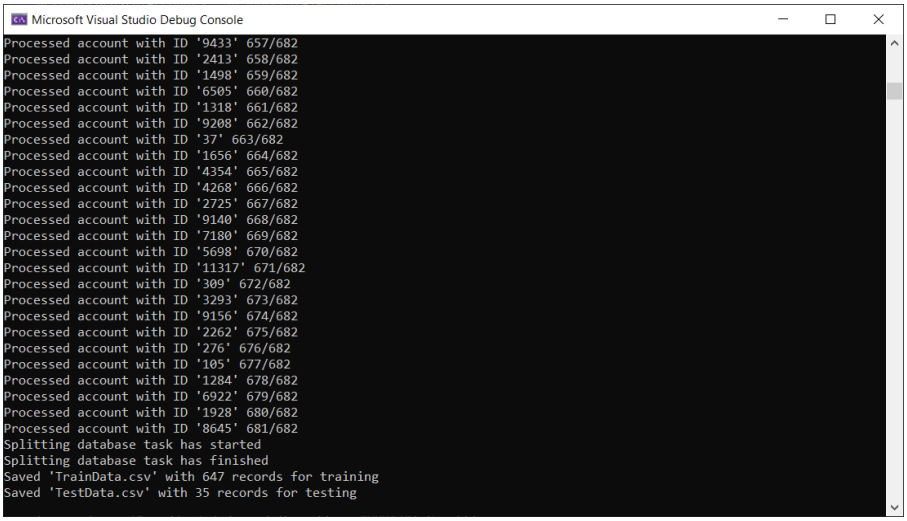

Рис. 2. Результат генерації даних для аналізу

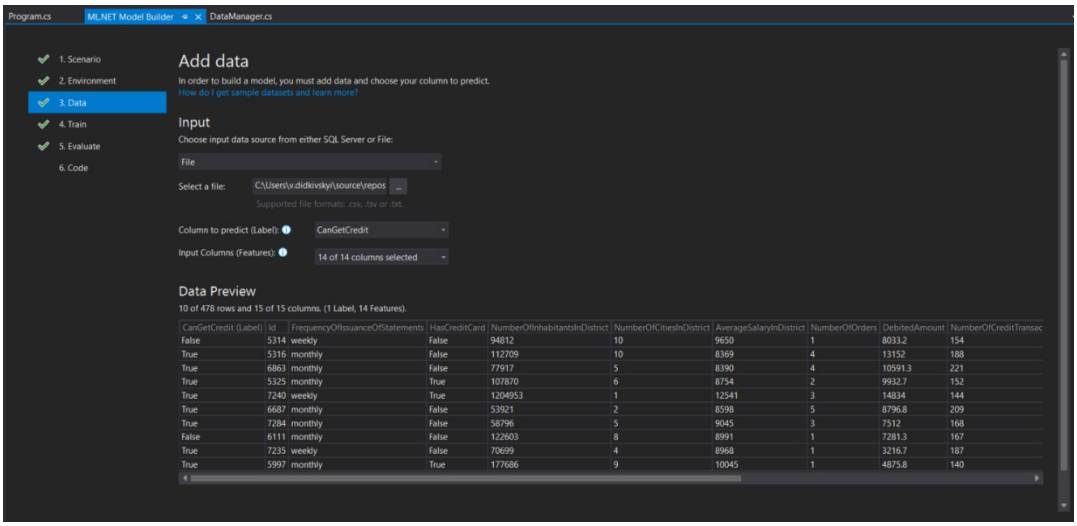

Рис. 3. Вибір даних для тренування моделі

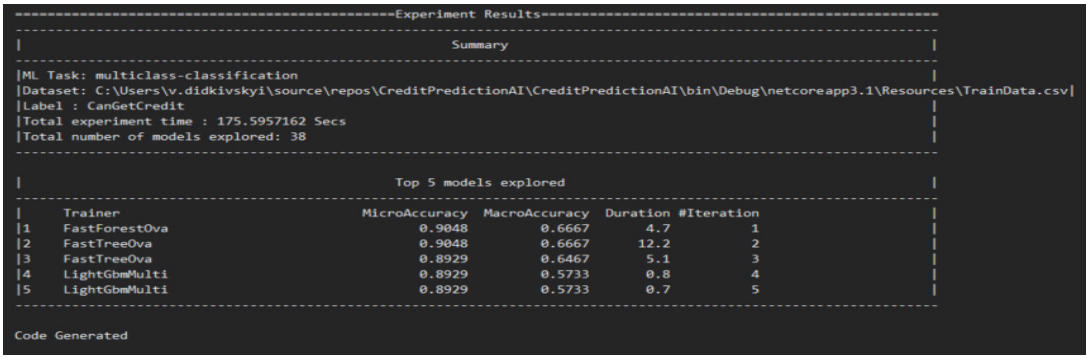

Рис. 4. Результат тренування моделі

```
5731;monthly;False;75232;5;8980;1;7683.2;34;65;-2376.9;460980;60;7683;False
```

Рис. 5. Рядок даних для тестового прогнозу

цьому етапі потрібно обрати час тренування моделі. Для того, щоб ModelBuilder міг протестувати всі можливі алгоритми, які були описані у першому розділі, встановимо час тренування 180 секунд (3 хвилини).

На рис. 4 видно, що алгоритмом із найвищою точністю для заданої вибірки виявився Fastforest, отже, він і буде використовуватися для виконання передбачень.

Після генерації моделі ML.NET ModelBuilder дає можливість виконати тестове передбачення. Введемо дані одного запису із файлу "TestData.csv" (рис. 5) та виконаємо прогноз (рис. 6).

На рис. 6 видно, що результат передбачення "False", що збігається зі значенням колонки "CanGetCredit" для цього запису у тестовому файлі. Після етапу оцінювання ML.NET ModelBuilder генерує файл моделі та код, який можна використовувати для додавання моделі у програмі. Моделі ML.NET зберігаються як zip файл. Код для завантаження та використання моделі додається як новий проект у рішення. ModelBuilder також додає зразок консольного додатка, який можна запустити, щоб побачити модель у дії (рис. 7). У результаті було отримано структуру рішення, представлену на рис. 8.

Для інтелектуального аналізу даних було згенеровано набір даних за допомогою реалізованого додатку. Загальна кількість отриманих тренувальних даних – 682 записи. ML.NET ModelBuilder використовує навчену модель для прогнозування з новими тестовими даними, а потім вимірює, наскільки прогнози є правильними.

За замовчуванням показник проблем класифікації – це точність. Точність визначає пропорцію

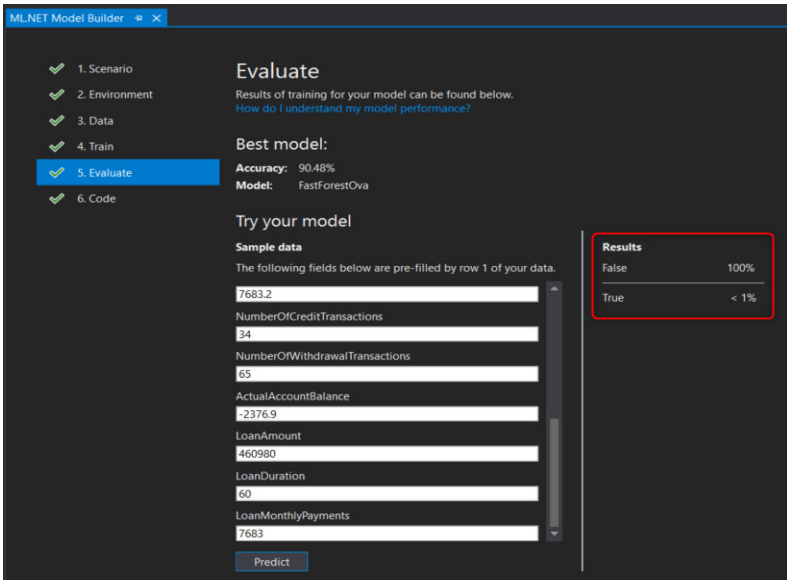

Рис. 6. Результат тестового прогнозу

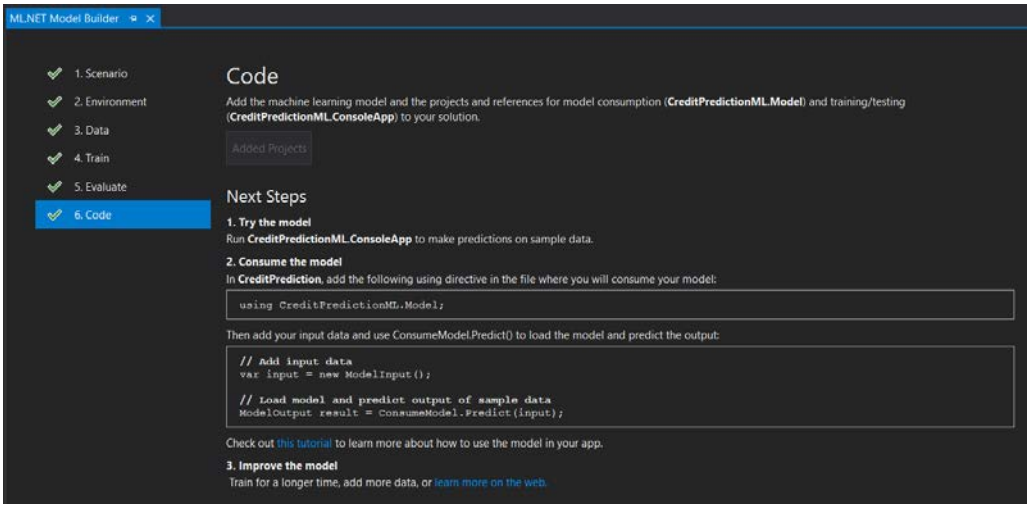

Рис. 7. Генерування файлу моделі та коду

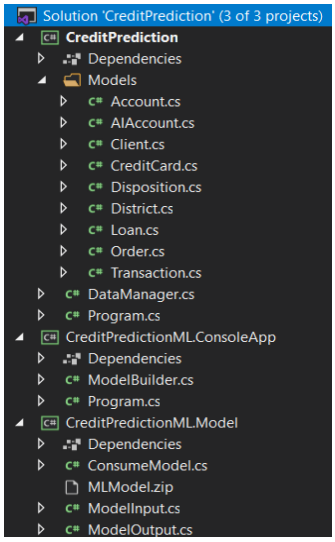

Рис. 8. Структура рішення після генерування моделі

```
5314;weekly;False;94812;10;9650;1;8033.2;154;235;5903.1;96396;12;8033;False
```

Рис. 9. Рядок даних для тестового прогнозу 1

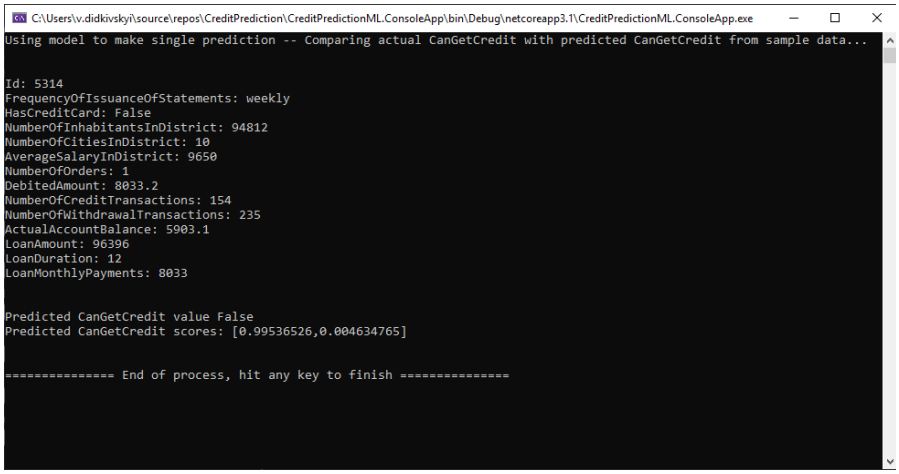

Рис. 10. Результат тестового прогнозу 1

правильних прогнозів, які модель робить над тестовим набором даних. Чим ближче до 100% або 1.0, тим краще. Використовуючи згенеровані реалізованим додатком дані, ML.NET ModelBuilder обрав алгоритм Fastforest і побудував модель із точністю д 90.48%, що може використовуватися для прогнозування оцінки кредитоспроможності фізичних осіб.

ModelBuilder самостійно розбиває навчальні дані на навчальний набір і тестовий набір. Дані про навчання (80%) використовуються для навчання моделі, а дані тесту (20%) – для оцінки моделі. У цій роботі із загального набору даних було також виділено 5% в окремий файл “TestData.csv” для того, щоб вручну переконатися на нейтральних даних, що модель дійсно працює. Проведемо тестування згенерованої моделі ML.NET.

У рядку даних відповідь знаходиться вкінці. У цьому випадку відповідь “False”.

На рис. 10. видно, що прогноз правильний. Провівши ще додатково 3 тестування, було встановлено, що згенерована модель дійсно вирішує проблемне питання – дає прогноз оцінки кредитоспроможності фізичних осіб.

ВИСНОВКИ

У дослідженні була побудована модель і проведено прогнозування оцінки кредитоспроможності фізичних осіб з використанням можливостей ML.NET. Було з’ясовано, що виконання прогнозування оцінки кредитоспроможності фізичних осіб – це задача бінарної класифікації. Внаслідок цього було реалізовано програму для формування набору даних, проведено інтелектуальний аналіз даних з використанням можливостей ML.NET.

Використання методів інтелектуального аналізу даних сприятиме не лише можливостям розробки додатків із прогнозування широкого спектру параметрів, а й дозволить покращити розуміння економічних концепцій в освітньому процесі, що сприятиме розвитку економічних компетентностей студентів і фахівців.

REFERENCES

1. Binarna klasyfikatsiia. [Binary classification]. Retrieved from: https://uk.wikipedia.org/wiki/Бінарна_класифікація.
2. ML.NET Retrieved from: https://en.wikipedia.org/wiki/ML.NET.
3. What is Model Builder and how does it work? Retrieved from: https://docs.microsoft.com/en-us/dotnet/machine-learning/automate-training-with-model-builder.
4. Binary Classification Trainer Enum. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.automl.binaryclassificationtrainer.
5. Averaged perceptron algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.averagedperceptrontrainer.
6. Fast forest algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.fasttree.fastforestbinarytrainer.
7. Fast tree algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.fasttree.fasttreebinarytrainer.
8. LBFGS logistic regression algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.lbfgslogisticregressionbinarytrainer.

9. Light GBMalgorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.lightgbm.lightgbmbinarytrainer.
10. Linear SVM algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.linearsvmtrainer.
11. SDCA algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.sdcalogisticregressionbinarytrainer.
12. SGD calibrated algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.sgdcalibratedtrainer.
13. Symbolic SGD logistic regression algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.symbolicsgdlogisticregressionbinarytrainer.
14. Aziz Saqib, Michael Dowling. (2019) Machine learning and AI for risk management. Disrupting Finance. Palgrave Pivot, Cham, p. 33–50.
15. Ma X., Lv S. (2019). Financial credit risk prediction in internet finance driven by machine learning. Neural Computing and Applications, 31(12), p. 8359–8367.
16. Grabchenko A.I,. Fedorovich V.O., Garashchenko Y.M. Metody naukovykh doslidzhen : Navch. posibnyk. [Research methods : Textbook. manual.] H. : NTU "KhPI", 2009. 142 p.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Бінарна класифікація. Retrieved from: https://uk.wikipedia.org/wiki/Бінарна_класифікація.
2. ML.NET. Retrieved from: https://en.wikipedia.org/wiki/ML.NET.
3. What is Model Builder and how does it work? Retrieved from: https://docs.microsoft.com/en-us/dotnet/machine-learning/automate-training-with-model-builder.
4. Binary Classification Trainer Enum. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.automl.binaryclassificationtrainer.
5. Averaged perceptron algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.averagedperceptrontrainer.
6. Fast forest algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.fasttree.fastforestbinarytrainer.
7. Fast tree algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.fasttree.fasttreebinarytrainer.
8. LBFGS logistic regression algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.lbfgslogisticregressionbinarytrainer.
9. Light GBMalgorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.lightgbm.lightgbmbinarytrainer.
10. Linear SVM algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.linearsvmtrainer.
11. SDCA algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.sdcalogisticregressionbinarytrainer.
12. SGD calibrated algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.sgdcalibratedtrainer.
13. Symbolic SGD logistic regression algorithm. Retrieved from: https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.symbolicsgdlogisticregressionbinarytrainer.
14. Aziz Saqib, Michael Dowling. (2019). Machine learning and AI for risk management. *Disrupting Finance.* Palgrave Pivot, Cham, p. 33–50.
15. Ma X., Lv S. (2019). Financial credit risk prediction in internet finance driven by machine learning. *Neural Computing and Applications*, 31(12), p. 8359–8367.
16. Грабченко А.І., Федорович В.О., Гаращенко Я.М. (2009). Методи наукових досліджень : навч. посібник. Х. : НТУ «ХПІ». 142 с.

Дата надходження статті до редакції: 09.10.2020
Дата затвердження статті до друку: 20.10.2020