Київський національний торговельно-економічний університет

Кафедра кібернетики та системного аналізу

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Визначення рейтингових показників студентів закладу вищої освіти за допомогою алгоритмів машинного навчання»

Студента 2 курсу, 1м групи,

спеціальності «Економіка» _____ *підпис студента* Стародубова Володимира Дмитровича

Освітня програма «Економічна кібернетика»

Науковий керівник кандидат економічних наук, ст. викладач _____ *підпис керівника* Кулаженко Володимир Валерійович

Гарант освітньої програми доктор фізико-математичних наук, професор _____ *підпис гаранта* Гамалій Володимир Федорович

Київ 2018

Київський національний торговельно-економічний університет

Факультет обліку, аудиту та інформаційних систем
Кафедра кібернетики та системного аналізу
Спеціальність 051 «Економіка»
Спеціалізація «Економічна кібернетика»

Затверджую
Зав. кафедри _____Роскладка А. А.
«05» листопада 2017р.

Завдання
на випускну кваліфікаційну роботу (проект) студенту

Стародубову Володимиру Дмитровичу
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)
«Визначення рейтингових показників студентів закладу вищої освіти за допомогою алгоритмів машинного навчання»
Затверджена наказом ректора від *«02» жовтня 2017 р. № 3035*
2. Строк здачі студентом закінченої роботи *15 листопада 2018 року*
3. Цільова установка та вихідні дані до роботи
Мета роботи: дослідження та побудова моделей машинного навчання для оптимізації процесів оцінювання та прогнозування рейтингу студентів.
Об'єкт дослідження: система рейтингового оцінювання
Предмет дослідження: методи, моделі та алгоритми машинного навчання.
4. Перелік графічного матеріалу *25 рисунків, 2 таблиці та 11 формул у основному тексті. У додатках 2 таблиці та 3 файли з програмним кодом на Python.*
_____
_____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Кулаженко В. В.	05.11.2017 р.	05.11.2017 р.
2	Кулаженко В. В.	05.11.2017 р.	05.11.2017 р.
3	Кулаженко В. В.	05.11.2017 р.	05.11.2017 р.

6. Зміст випускної кваліфікаційної роботи (проекту) (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 ОГЛЯД ТА АНАЛІЗ РЕЙТИНГОВИХ СИСТЕМ ОЦІНЮВАННЯ ЗНАНЬ СТУДЕНТІВ

1.1. Поняття та загальні засади рейтингових систем оцінювання знань студентів

1.2. Особливості рейтингових систем в навчальних закладах

1.3. Проблеми формування рейтингових оцінок і шляхи їх вирішення методами машинного навчання

Висновки до розділу 1

РОЗДІЛ 2. ТЕОРЕТИКО-МАТЕМАТИЧНІ ОСНОВИ ТЕОРІЇ МАШИННОГО НАВЧАННЯ ТА ПРОГНОЗУВАННЯ

2.1. Сучасні напрямки розвитку теорії машинного навчання

2.2. Огляд алгоритмів машинного навчання для рейтингових оцінок

2.3 Модель рейтингового оцінювання студентів на основі алгоритмів класифікації

Висновки до розділу 2

РОЗДІЛ 3. РОЗРОБКА АВТОМАТИЗОВАННОЇ СИСТЕМИ РЕТИНГОВОГО ОЦІНЮВАННЯ

3.1. Інструменти, методи та технології розробки автоматизованої рейтингової системи

3.2. Формування первинного набору даних та технології їх збагачення та очищення

3.3. Реалізація програмного забезпечення з розподілу стипендій

Висновки до розділу 3

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

7. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1	*Вибір теми випускної кваліфікаційної роботи*	*01.10.2017*	*01.10.2017*
2	*Розробка та затвердження завдання на випускну кваліфікаційну роботу*	*05.11.2017*	*05.11.2017*
3	*Вступ*	*01.04.2018*	
4	*Розділ 1. Огляд та аналіз рейтингових систем оцінювання знань студентів*	*01.05.2018*	
5	*Розділ 2. Теоретико-математичні основи теорії машинного навчання та прогнозування*	*20.06.2018*	
6	*Підготовка статті у збірник наукових статей магістрів*	*15.09.2018*	
7	*Розділ 3. Розробка автоматизованої системи рейтингового оцінювання*	*01.10.2018*	
8	*Висновки*	*01.11.2018*	
9	*Здача випускної кваліфікаційної роботи на кафедру науковому керівнику*	*15.11.2018*	
10	*Попередній захист випускної кваліфікаційної роботи*	*22.11.2018*	
11	*Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи*	*25.11.2018*	
12	*Представлення готової зшитої випускної кваліфікаційної роботи на кафедру*	*28.11.2018*	
13	*Публічний захист випускної кваліфікаційної роботи*	*За розкладом роботи ЕК*	

8. Дата видачі завдання «05» листопада 2017 р.

9. Керівник випускної кваліфікаційної роботи (проекту)

Кулаженко В.В.
(прізвище, ініціали, підпис)

10. Гарант освітньої програми Гамалій В.Ф.
(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Стародубов В.Д.
(прізвище, ініціали, підпис)

12. Відгук керівника випускної кваліфікаційної роботи (проекту)

Керівник випускної кваліфікаційної роботи (проекту)
_____ 15.11.2018 р.
(підпис, дата)

13. Висновок про випускну кваліфікаційну роботу (проект)

Випускна кваліфікаційна робота (проект) студента _____
(прізвище, ініціали)
може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми_____Гамалій В.Ф.____
(підпис, прізвище, ініціали)

Завідувач кафедри _____Роскладка А.А.___
(підпис, прізвище, ініціали)
«_____»_____2018 р.

Анотація

У даній роботі було розглянуто систему рейтингового оцінювання студентів, її базові принципи та проблеми реалізації в українських та зарубіжних вищих навчальних закладах. Розкрито базові поняття машинного навчання, сучасні напрямки його розвитку та алгоритми створення моделей. Проаналізовано сучасні програмні бібліотеки та існуючи шаблони мови програмування Python в сфері розв'язування задач машинного навчання. Представлено програмна реалізація системи прогнозування рейтингового оцінювання студентів на базі алгоритмів машинного навчання, а саме отримання студентами стипендій у майбутні періоди.

Ключові слова: рейтинг, оцінювання, машинне навчання, алгоритм, модель, класифікатор, набір даних.

Annotation

In this work are revealed questions about the rating assessment of students, its basic principles and problems of realization in Ukrainian and foreign higher educational establishments was considered. The basic concepts of machine learning, modern directions of its development and algorithms of model creation are revealed. The analysis of modern software libraries and existing Python programming language templates in the field of machine learning tasks are analyzed. The program realization of the system of prediction of rating estimation of students on the basis of machine learning algorithms, namely receiving scholarships for future periods by students, is presented.

Keywords: rating, assessment, machine learning, algorithm, model, classifier, data set.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА АБРЕВІАТУР

РСО – Рейтингова система оцінювань

КНТЕУ – Київський Національний Торгівельно-Економічний Університет

НТУУ – Національний Технічний Університет України

ХНЕУ – Харківський Національний Економічний Університет

ВСТУП

У зв'язку зі значними змінами в житті суспільства проблема вдосконалення професійної підготовки набуває особливого значення, а в результаті цих змін передбачається зміна змісту і методів діяльності викладача. Адаптивність системи освіти до наукового і технічного прогресу в умовах переходу від принципу навчання «освіта на все життя» до безперервної освіти досягається шляхом фундаменталізації змісту освіти, підвищення продуктивності методів навчання, формування професійної компетентності викладача. Особливістю підготовки кваліфікованого фахівця є необхідність досягнення не тільки стандартизованих освітніх результатів, але і творчих особистісних успіхів.

Шкільна система бальних оцінок (абсолютна кількісна шкала) не завжди відображає якісну зміну учня в процесі навчання, фіксуючи в ній лише результат навчального процесу. У зв'язку з чим, в процес освіти була введена нова система контролю і оцінки знань - система рейтингового контролю. Цей вид контролю не є чимось новим для європейських країн. У нашій же країні рейтинг став застосовуватися тільки в ряді вищих і середніх спеціальних навчальних закладах, а також в деяких середніх школах в якості експерименту.

Тема даної випускної кваліфікаційної роботи дуже актуальна на сьогоднішній день. Адже сьогодні змінюються цілі і завдання навчання, і, відповідно, змінюються освітні стандарти, навчальні плани, йде процес диференціації освітнього процесу за профілями і рівнями навчання. Вирішенню цих завдань сприяє застосуванням сучасних технологій які допомагають зменшити кількість роботи, виконувані викладачами та студентами.

Процес навчання передбачає наявність результатів навчання. Про відповідність цілей і результатів можна говорити, коли є необхідні механізми і інструменти вимірювання досягнення цих цілей і результатів. Самі ж вимірювальні інструменти багато в чому залежать від характеру критеріїв, які застосовуються при оцінці.

У зв'язку з цим з’являється необхідність комплексного вирішення таких питань як поєднання форм і методів контролю, розробка новітніх систем оцінювання, а також ролі системи рейтинг-контролю як одного з найважливіших мотивуючих факторів. Це дозволило сформулювати проблему дослідження, сутність якої полягає в аналізі навчальної успішності учнів.

Об'єкт дослідження: система рейтингового оцінювання.

Предмет дослідження: методи, моделі та алгоритми машинного навчання.

Мета дослідження: дослідження та побудова моделей машинного навчання для оптимізації процесів оцінювання та прогнозування рейтингу студентів.

В основу дослідження була покладена гіпотеза, згідно з якою найефективнішим способом моніторингу досягнень студентів є програмне забезпечення.

Відповідно до гіпотези, потрібно вирішити наступні завдання:

1) Визначити сутність рейтингової системи;
2) Проаналізувати наявний досвід України та зарубіжних вищих навчальних закладів;
3) Розкрити базові поняття машинного навчання;
4) Оглянути можливі алгоритми машинного навчання, які можна застосувати у рейтинговому оцінюванні студентів;
5) Розробити програмне забезпечення, яке дозволяє проводити моніторинг наявних оцінок та призначення стипендій.

Методи дослідження.

На різних етапах роботи для вирішення поставлених завдань використовувався комплекс методів, до яких відноситься:

1) теоретичний аналіз психолого-педагогічної літератури;
2) вивчення та аналіз діючих навчальних планів і робочих програм різних навчальних закладів;
3) статистико-математичні методи обробки даних;
4) алгоритмічні методи програмування.

РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ РЕЙТИНГОВИХ СИСТЕМ ОЦІНЮВАННЯ ЗНАНЬ СТУДЕНТІВ

1.1. Поняття та загальні засади рейтингових систем оцінювання знань студентів

Вітчизняні науковці під рейтингом (від англ. rating - оцінка, оцінювання, віднесення до розряду, категорії) визначають узагальнюючу порівняльну оцінку фінансово-господарського стану організацій та їх ранжування за певними критеріями. Підґрунтям рейтингу є узагальнена характеристика за певною ознакою, що дає можливість групувати організації у певній послідовності залежно від значення цієї ознаки. Отже, рейтинг - це метод порівняльної оцінки діяльності кількох установ однієї чи різних сфер діяльності залежно від обраних контрольних показників в управлінських цілях [4].

Термін «рейтинг» можна розкрити по різному, але потрібно акцентувати увагу на кількісній, вартісній природі рейтингу, визначає його як всеохоплюючу оцінку стану обраного об’єкта, що, в результаті, дає змогу віднести його до певного класу чи категорії. Також потрібно розмежовувати поняття «рейтинг» та «експертна оцінка». При цьому наголос ставиться на тому, що рішення про присвоєння рейтингів приймаються на базі застосування різноманітних методологічних підходів та прийомів економіко-математичного моделювання, а експертні оцінки, хоча й тісно переплітаються з рейтингами, є результатом висновків певного фахівця чи групи фахівців, котрі здатні приймати рішення про сучасний та майбутній стан досліджуваної організації, керуючись власними знаннями, досвідом, професійною інтуїцією та іншими суб’єктивним характеристиками [5].

Рейтингова система - сукупність правил, методичних вказівок і відповідного математичного апарату, реалізованого в програмному комплексі, забезпечує комплексне оцінювання досягнень студента у навчальній, науково-дослідній, культурно-масовій, соціальній та спортивній роботі, громадській діяльності [2].

Зазвичай під рейтингом розуміється «накопичена оцінка» як з окремих дисциплін, так і з циклу дисциплін за певний період навчання.

У практиці вищих навчальних закладів рейтинг - це деяка числова величина, виражена, як правило, по певною шкалою (наприклад, ECTS або п’ятибальна), яка інтегрально характеризує успішність і знання студента по одному або кількох предметів протягом певного періоду навчання (семестр, рік і т.д.)

Професор Дуканич Л.В. визначає рейтинг як вид інтегральної комплексної оцінки, до якої існує ряд вимог, зокрема [4]:

1) вона повинна враховувати сукупність критеріїв та показників, які всебічно характеризують об’єкт оцінки;
2) така оцінка має бути загальноприйнятою;
3) вона повинна бути транспарентною (має бути чітко зрозуміло, за якими саме характеристиками, в яких пропорціях і за якими правилами вона була визначена).

Мета рейтингового навчання полягає в тому, щоб створити умови для мотивації самостійності студентів засобами своєчасної та систематичної оцінки результатів їх роботи відповідно до реальними досягненнями.

В основі рейтингової системи контролю знань лежить комплекс мотиваційних стимулів, серед яких - своєчасна і систематична оцінка результатів в точній відповідності з реальними досягненнями студентів, система заохочення добре успішних учнів [6].

Запровадження системи рейтингового оцінювання діяльності студентів здійснюється з метою мотивації щодо отримання високого рівня знань, органічного поєднання в освітньому процесі навчальної, наукової та інноваційної складової, формування особистості шляхом патріотичного, правового, екологічного виховання, утвердження моральних цінностей, соціальної активності, громадянської позиції та відповідальності, здорового способу життя, бути креативним та самоорганізовуватися в сучасних умовах [2].

Кожен навчальний заклад має свою систему оцінювання, яка варіюється по різним критеріям, але в Україні дотримуються єдиного принципу Болонського процесу, яка приєдналася до нього 19 травня 2005 році [27].

Основні завдання Болонського процесу націлені на створення єдиної освітньої європейської системи. Крім цього, одне із завдань реформи – ввести систему співставлення дипломів і рівне визнання їх у всіх країнах так званої «Зони європейської вищої освіти». Також згідно з домовленістю між державами необхідно затвердити єдину рейтингову систему оцінювання студентів (ECTS).

Система ECTS – це єдиний порядок переведення і накопичення кредитів, який надає можливість вести облік загального обсягу годин, присвяченого дисципліні протягом всього навчального процесу, і при цьому дає студентам і викладачам свободу переведення з одного вищого навчального закладу до іншого без втрати цих кредитів [1]. За підсумком накопичення 180-240 кредитів студенту присвоюється ступінь бакалавра, а для диплома магістра потрібно ще заробити 60-120 кредитів.

Але кредит – це кількісна одиниця виміру пройденого матеріалу. Є ще і якісна, яка виражається в балах. A, B, C, D, E, FX, F – перші п’ять пунктів є задовільними для отримання кредитних балів, а два останніх – ні.

За шкалою ECTS студенти, які склали іспит або залік, за результатами успішності діляться на 5 груп (в порядку убування): 10% кращих студентів отримують позначку «A», 25% наступних студентів за рівнем академічної успішності – «B», 30% – «C», 25% – «D» і 10% -«E».

Основний алгоритм роботи рейтингової системи контролю знань поділяється на такі пункти:

1) курс навчання по кожному предмету розбивається на тематичні розділи, контроль за якими обов’язково проводитися викладачем протягом всього курсу;
2) після закінчення навчання по кожному розділу проводиться повний контроль знань студентів з оцінкою в балах;
3) в кінці курсу визначається сума, набраних за весь період, балів і виставляється загальна оцінка. Студенти, які мають підсумкову суму балів по рейтингу від 90 до 100 (A) можуть бути звільнені від заліків (іспитів);

4) загальні оцінки за всі предмети протягом періоду додаються та діляться на їхню кількість, демонструючи навчальну успішність студента.

1.2. Особливості рейтингових систем у вищих навчальних закладах

В Україні та Європі вищезазначена система ECTS використовується як основний параметр успішності студента, але він не використовується самостійно. Якщо брати, наприклад, Київський Національній Торгово-Економічний Університет, то згідно положенню про систему рейтингового оцінювання діяльності студентів, основними видами діяльності студентів є [2]:

1) навчальна робота;
2) науково-дослідна робота;
3) громадська діяльність;
4) культурно-масова, соціальна та спортивна робота.

Навчальна робота – основний вид діяльності студента у період навчання в університеті, спрямований на отримання відповідних професійних компетентностей за обраною спеціальністю. Кількість балів за навчальну роботу визначається за кожний семестр шляхом вирахування середнього арифметичного значення балової оцінки за шкалою оцінювання КНТЕУ, враховуючи бали з усіх форм контролю.

Науково-дослідна робота – одна із складових освітнього процесу, що формує навички творчого та ефективного вирішення наукових завдань, у тому числі інноваційного характеру.

Громадська діяльність є реалізацією права і можливості студентів брати участь в обговоренні та вирішенні питань удосконалення освітнього процесу, науково-дослідної роботи; призначення стипендій, організації дозвілля, оздоровлення, побуту, харчування, захисту прав та інтересів студентів, участі в управлінні вищим навчальним закладом, передбачених у статтях 40, 41 Закону України «Про вищу освіту». Громадська діяльність охоплює різні сфери суспільного життя молоді, сприяє розвитку ініціативності, організаторських, управлінських здібностей студентів.

Культурно-масова, соціальна та спортивна робота посідає чільне місце у формуванні духовності, культури та популяризації здорового способу життя студентської молоді. Цей вид роботи сприяє всебічному розвитку особистості, її 3 здібностей, зміцненню здоров'я та фізичного загартовування, сприяє залученню студентства до волонтерства та благодійності, виховує у студента почуття небайдужості до навколишнього світу та спонукає до саморозвитку.

Кількість балів за навчальну роботу вираховується автоматично за допомогою електронної системи «Деканат» (із застосуванням формули X/n, де X – сума отриманих балів за шкалою оцінювання КНТЕУ; n – кількість навчальних дисциплін).

Рейтингове оцінювання результатів своєї позанавчальної діяльності студент здійснює, використовуючи для розрахунків таблицю «Рейтингове оцінювання позанавчальної діяльності студента» (дод. А).

При визначенні балів за громадську діяльність бали зараховуються відповідно до займаної посади за умови, що студент займав цю посаду не менше 2/3 семестру.

Загальна кількісна оцінка (рейтинг) за семестр визначається сумою балів за навчальну роботу та за видами діяльності, зазначеними у додатку А. Для студентів, які складають сесію під час ліквідації академічної заборгованості, рейтинг може бути скорегований на початку наступного семестру.

Рейтинг студентів враховується при:

1) умові рівності балів при нарахуванні академічних стипендій;
2) призначенні іменних та персональних стипендій;
3) переведенні студентів на вакантні місця державного замовлення;
4) покращенні умов проживання у гуртожитку та поселенні на наступний навчальний рік;
5) заохоченні, в тому числі матеріальному;
6) формуванні наукового резерву та рекомендації до вступу до аспірантури;
7) інших випадках, які потребують порівняння результатів діяльності студентів.

Інформування студентів щодо проведення семестрового рейтингового оцінювання діяльності студентів забезпечує рада студентського самоврядування факультету.

Студент, за власним бажанням, на підставі написаної заяви має право відмовитися від участі в рейтинговому оцінюванні позанавчальної роботи за семестр. Якщо академічна група студентів у повному складі за власним бажанням хоче відмовитись від участі у рейтинговому оцінюванні позанавчальної роботи за семестр, то підставою для цього є заява написана старостою групи із зазначенням підписів усіх студентів групи. У такому випадку при формуванні рейтингу враховуватиметься лише оцінка за навчальну роботу студента [2].

Якщо взяти інші вищі навчальні заклади України, то в більшості застосовуються однакові принципи, відповідно до Закону України «Про вищу освіту». Більш складна система у Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

В засадах побудов рейтингової системи оцінювання передусім визначено систему контрольних заходів з кредитного модуля (за наявності змістових модулів – окремо з кожного з них): певне індивідуальне семестрове завдання та модульні контрольні роботи, що передбачені у робочому навчальному плані, колоквіуми, звіти та захист лабораторних робіт, а також поточний контроль на практичних і семінарських заняттях тощо. При плануванні контрольних заходів двогодинні модульні контрольні роботи можуть бути поділені на дві одногодинні або три 30-хвилинні контрольні роботи тощо.

Після побудови системи контрольних заходів визначаються максимальні бали з кожного контрольного заходу (r_k) з урахуванням важливості, трудомісткості та обсягу певної навчально-пізнавальної діяльності студента.

Визначення орієнтовних значень вагових балів з кожного контрольного заходу можливо на підставі розподілу навчального часу студентів згідно з тематичним планом робочої навчальної програми кредитного модуля [3]. Розрахунок проводиться за допомогою формули:

$$r_k = R\frac{t_k}{\sum_i t_i}, \quad (1.1)$$

де r_k - вагові бали;

t_k - навчальний час, запланований у робочій навчальній програмі для засвоєння навчального матеріалу (знань і умінь), який має контролюватися k -м контрольним заходом;

$\sum_i t_i$ - загальний навчальний час, призначений для засвоєння навчального матеріалу, який охоплюється всіма контрольними заходами, що заплановані рейтинговою системою оцінювання;

R - значення розміру шкали рейтингової системи оцінювання.

Значення шкали R у формулі (1.1) повинне дорівнювати 100, а для зручності доцільно використати округлення r_k до цілих чисел.

Рейтингова оцінка з кредитного модуля, семестрова атестація з якого передбачена у вигляді заліку (диференційованого заліку), а також з навчальних дисциплін обсягом менш ніж 3 кредити, з яких семестрову атестацію передбачено у вигляді екзамену, формується за допомогою формули (1.2) як сума всіх рейтингових балів r_k та заохочувальних балів r_s, яка повинна не перевищувати 0,1R.

$$RD_1 = \sum_k r_k + \sum_s r_s, \quad (1.2)$$

де RD_1 - рейтингова оцінка першого типу (залік);

$\sum_k r_k$ - сума усіх рейтингових балів за курс;

$\sum_s r_s,$ - сума усіх заохочувальних балів за курс.

Рейтингова оцінка з кредитних модулів обсягом більше трьох кредитів, семестрова атестація з яких передбачена у вигляді екзамену, формується у формулі (1.3) як сума балів поточної успішності навчання – рейтингової оцінки першого типу, яка розраховується формулою (1.2) та екзаменаційного балу r_e.

$$RD_2 = RD_1 + r_e, \quad (1.3)$$

де RD_2 - рейтингова оцінка першого типу (екзамен);
RD_1 - рейтингова оцінка першого типу (залік, формула 1.2);
r_e - бал за екзамен.

Остаточний рейтинговий бал формується як середнє значення балів, отриманих за заліки та екзамени.

Для повноти дослідження потрібно порівняти на яких засадах працюють виші навчальні заклади не європейського зразка, і наскільки сильно відрізняється система українських вищих навчальних закладів, яка працює по єдиній системі Болоньского процесу. На фоні цього розглянемо американську систему рейтингового оцінювання та її фактори, які дозволяють оцінити роботу студента максимально справедливо і об'єктивно.

Бально-рейтингова система в США розроблена в умовах індивідуально-орієнтованої організації навчального процесу і, незважаючи на безліч альтернативних оцінок досягнень студентів, є двоступеневою і включає: оцінки з дисциплін та оцінку за середнім показником успішності.

В американських університетах існує декілька систем оцінки знань студентів, залежно від предмета, від побажань студентів, від виду і тривалості курсу [7]:

1) Літерна система A-F, де оцінка «A, A-» означає виключне якість виконання роботи (у балах — 4; 3,7), «B+, B, B-» — висока якість (3,3; 3,0; 2,7), «C+ C, C-» — середнє (2,3; 2,0; 1,7), «D+, D, D- » — погана якість (1,3; 1,0; 0,7), «F» — провал.
2) Система I-U, де оцінка «I» (англ. Incomplete — незакінчена робота) ставиться, коли вимоги курсу не до кінця виконані, але робота ведеться і викладач не проти того, щоб дати додатковий час для завершення роботи, вказуючи крайній термін подання закінченої роботи. Оцінка «W» (англ. Withdrawal — відмова від курсу) ставиться, коли студент відмовляється від курсу задовго до його закінчення, і тоді середній бал з даного курсу не

проставляється, він просто не враховується у переліку пройдених курсів. Студент може відмовитися від курсу, якщо йому загрожує оцінка «FN» (англ. Failure for Non attendance) — не атестований за невідвідування занять. R (англ. thesis in progress) — робота над дисертацією (науковим проектом) триває, P (англ. Pass) — здано, N (англ. No credit) — не здано, S (англ. Satisfactory) — задовільно, U (англ. Unsatisfactory) — незадовільно. Студенти можуть зареєструватися на аудиторный курс, тобто курс, за який не ставиться ні кредитів, ні оцінок, а у відомості буде зазначено «AUD» (англ. audited course — прослухав курс).

3) Студент може вибрати курси, які будуть оцінюватися по системі S/U — задовільно/незадовільно, але кількість курсів по даній системі не повинно перевищувати 36 кредитів у студента, що навчається на ступінь бакалавра. Студент, який навчається в аспірантурі (магістранти та докторанти), також може вибрати цю систему оцінок, але для курсів програми бакалавра або курсів градуйованою школи, але не входять в обов'язкову програму даного студента. Оцінка «S» (задовільно) ставиться, якщо студент отримав A, A-, B+, B, B-, C+, C протягом курсу, а оцінка «U», якщо студент отримав D+, D, D, F. Причина вибору даної системи може бути відома тільки студенту і його навчального консультанту, а викладач самих курсів про те, яку систему оцінок вибрав той чи інший студент, зазвичай не обізнаний, він оцінює по традиційній системі A-F. Переведення оцінок в систему «S/U» буде проходити автоматично, згідно з проставленим оцінками в системі on-line.
4) Оцінка «P/N» (зданий/не зданий) дається за курси, які важко оцінити, якої немає необхідності оцінювати за системою A-F. Це курси, пов'язані з практикою, з накопиченням (нарощуванням) майстерності, які дають загальне уявлення про предмет або орієнтування в предметі. Студент, який отримав «P», заробляє кредити за курсом, а отримав «N» — не заробляє.

В кінці кожного семестру в університетах США підраховується середній бал пройдених предметів (GPA — Grade Point Average). Середній бал успішності

показується тільки в тому випадку, якщо студент навчається для отримання якої-небудь міри, і він не включає жодну іншу оціночну систему, крім A-F. Підрахунок ведеться в системі on-line.

За результатами середнього балу успішності студентів з GPA нижче 2,0 отримують «Попередження» (англ. Academic Warning). Студенти, що закінчили два або більше семестру з середнім балом нижче 2,0, отримують «Випробувальний термін» (англ. Academic Probation), якщо ж вони виправляють свої бали на 2,0 і вище, то випробувальний строк знімається.

Система дуже прозора. Викладач на першій зустрічі зі студентами докладно пояснює систему бальної оцінки, хоча вимоги по кожному предмету студент може знайти на сайті університету. Студенти повинні знати, коли і якою сумою балів буде оцінюватися той чи інший вид їх праці; як викладач його буде оцінювати; коли, як і за якими темами будуть проводитися тестування і контроль виконання самостійної роботи. Регулярний контроль забезпечує зворотний зв'язок, що дозволяє викладачеві зрозуміти, яким темам або завдань слід приділити більше уваги.

Студент, працюючи з викладачем протягом семестру, оцінюючи свої успіхи, вже з першого тижня знає, як підвищується перша складова його оцінки з дисципліни, і в кінці семестру він може з високим ступенем ймовірності визначити її можливе підсумкове значення.

Друга складова оцінки з дисципліни — оцінка знань студента на іспиті.

Тестування, як правило, складається з трьох частин: «Загальні поняття», «Основна частина», «Вирішення проблеми». Перша виявляє знання основних, базових понять навчальної дисципліни. Студент ставиться в такі умови, при яких виключається можливість вгадування відповідей: за правильну відповідь нараховується 1 бал, за неправильну 1 бал знімається. Відсутність відповіді оцінюється в 0 балів. Друга частина вимагає від студента умінь і навичок рішення за обмежений час великої кількості відносно простих завдань в обсязі всього курсу. Цю частину тесту можна виконати тільки за умови регулярної роботи

протягом семестру. Третя частина тесту дозволяє оцінити вміння студента самостійно вирішувати конкретні проблемні завдання.

Для забезпечення об'єктивності і надійності результатів тестування приймається ряд спеціальних заходів: використання комп'ютерних генераторів варіантів тестових завдань, формування варіантів тестів безпосередньо перед іспитом або на самому іспиті в присутності экзаменуемых студентів [7].

І американська та європейська системи оцінювання студентів, активне становлення та розвиток яких відбувалося у другій половині XX століття, будувалися переважно як накопичувальні, при цьому головною метою було усунення недоліків традиційних систем.

1.3. Проблеми формування рейтингових оцінок і шляхи їх вирішення методами машинного навчання

Проблеми рейтингового оцінювання завжди йшли разом с наявністю людського фактору. Але сьогодні, не дивлячись на науково-технічний розвиток світу, досі актуальні проблеми, які пов’язанні з академічною недоброчестністю, підтриманням актуальності навчальних дисциплін, адаптацією студентів до нових систем оцінювання та адекватним ваговим оцінюванням кожного предмету.

Але сформовані проблеми є лише половиною шляху до їх вирішення, тому потрібно розглядати їх з точки зору сучасного, постіндустріального суспільства, де використовуються автоматизовані рішення.

Розглядаючи пункт з першої названої проблеми, яка описана словосполученням «академічна недоброчестність», доктор фізико-математичних наук Володимир Бахрушин виділяє, що одним із важливих видів порушень академічної доброчесності є необ’єктивне оцінювання. В широкому розумінні, це будь-яке оцінювання, що здійснюється в академічному середовищі – оцінювання здобувачів освіти, викладачів, науковців, закладів освіти, керівників, органів управління тощо [8].

Сьогодні актуальною проблемою в цій сфері є оцінювання студентів викладачами. Це, насамперед, зумовлено, запровадженням нової системи

призначення стипендій [9]. Попередня система мотивувала студентів боротися чесними і не дуже чесними способами за отримання конкретних оцінок. Оцінки одного студента формально ніяк не впливали на можливість отримання стипендії іншими. Науковець вважає, що студенти, іноді ще могли заступитися за товариша, якому незаслужено поставили двійку. Але майже ніколи не оскаржували незаслужено завищених оцінок одногрупників.

З 2017 року стипендії призначають не за середнім балом, а за рейтингом. Викладач може поставити максимальні бали усій групі, але стипендію все одне отримають лише 40 – 45% кращих з них. А з 2019 стипендія буде видаватись лише чверті студентів [10]. Тому значно зростає увага до об'єктивності оцінювання. В першу чергу необхідно створити прості і зрозумілі показники, критерії та процедури оцінювання з кожної дисципліни, а також процедури формування рейтингів. Враховуючи, що визначати їх якість буде не комісія з акредитації, якої все це не дуже цікаво, а кожний студент при кожному поточному чи підсумковому оцінюванні. Тому потрібно шукати нові відповіді на традиційні запитання: що оцінювати, як оцінювати, хто має здійснювати оцінювання.

В рамках того що і як оцінювати, можна порівняння дисциплінованість та лояльність студента, його збіг відповідей з текстом підручника або конспекту лекцій, зазначені у програмі дисципліни результати навчання та інше. Також необхідність перегляду відношення до порушення дисципліни студентами, їх критичне ставлення до конкретних викладачів, кафедр чи вищих навчальних закладів, та вплив цих критеріїв на рейтинговий бал.

Дивлячись на іншу проблему, пов'язану з актуальністю навчальних дисциплін, В. Бахрушин справедливо роздумує, що можливо проблеми у викладачах, які не завжди розуміють, що за 30-50 років, які минули після закінчення ними вищих навчальних закладів, Україна і світ змінилися, або проблема в інших, викладачах які намагаються вчити студентів тому, що самі знають не з власного досвіду (наукового, виробничого), а лише з чужих книжок [8].

А якщо оцінювати результати навчання, визначені, як передбачає сучасна модель через компетентності випускника, то відразу виникають інші запитання. Як формулювати ці результати, як їх оцінювати? Чи готові викладачі визнавати передбачені програмою результати навчання, здобуті не на їх лекціях, а в якійсь інший спосіб? Чи готові студенти визнавати такі результати навчання у своїх товаришів, яких вони не бачили на лекціях? І знов дискусія повертається до питання, про те, хто і як має оцінювати – свій викладач, інший викладач, декілька викладачів, чи комп'ютер. Очевидно, що комп'ютер може оцінити далеко не все.

Алгоритми машинного навчання можуть розгладити процес оцінювання студентів. Ми звикли до того, що машині чітко задаються відповіді і вона може видати лише «булеве» так чи ні, але з розвитком технологій машини стають все розумнішими. Людина теж в деякому сенсі комп'ютер, в якому нагромаджено великий тягар зі знань та досвіду.

В недалекому майбутньому може буде розроблений програмний продукт, який зможе оцінювати твори студентів гуманітарних спеціальностей. І в цей продукт буде потрібно вкласти, якщо не сотні, то тисячі творів, оцінені компетентним професіоналом від п'ятірки до двійки. Технічні спеціальності мають перевагу, бо їх відповіді мають більш чіткі критерії.

Рис. 1.1 Блок-схема дії програмного забезпечення для перевірки творів

Це дуже кропітка робота, оскілки ми ще не дійшли до епохи повної комп'ютеризації. Навіть в нашому університеті облік студентів ведеться на паперовому носії, який потім вручну вноситься методистами. А машину ще потрібно навчити розпізнавати почерк кожного студента.

Третьою проблемою, яка була досліджена в статті науковців [11], є проблемність переходу студентів від шкільної, 12-бальної шкали оцінювання до європейських, які включають в себе буквені позначення, та цифрову 100-бальну шкалу.

Вчорашні школярі, абітурієнти, вступаючи на шлях бакалавра натикаються на проблему адаптації до нових шкал оцінювання, які суттєво відрізняються від шкільних. На сьогодні використовується «проміжний» варіант переведення всіх оцінок у радянську п'ятибальну шкалу, як подано в табл. 1.1. За допомогою алгоритмічного програмування ця проблема вирішується дуже швидко.

Таблиця 1.1

Порівняння балів між різними бальними системами

12-бальна система	5-бальна система	100-бальна система
1	1	0-34
3-2	2	35-59
6-4	3	60-74
9-7	4	75-89
10-12	5	90-100

Але цим не вирішити проблему, тому потрібно на законодавчому рівні впроваджувати однакові системи оцінювання в «нижчих» навчальних закладах.

Четверта, проблема важливості тої чи іншої дисципліни завжди гостро стоїть у студентів, оскільки кожен предмет однаково впливає на рейтинговий бал в кінці навчального періоду. І іноді здається, що це не зовсім чесно, коли на спеціальності, пов'язаної з комп'ютерними технологіями, філософія та соціологія мають однакову вагу з програмуванням.

Все це вирішується програмними засобами, множниками, які будуть корегувати рейтинговий бал під кожну спеціальність, але це залишається викритим питання резонності того, чи іншого множника.

Науковець з Калькуттського університету [12] вважає, що похибка оцінювання студентом є наслідком багатьох критеріїв, але основним я би виділив ефект ореолу (гало-ефект), коли викладач може спаплюжити бачення студента на дисципліну.

Якщо говорити про щось реальне, то вже зараз можна впровадити систему, яка буде слідкувати за доброчесністю студента. Кожен тест, кожна відповідь має певну складність (вагу), яка задається викладачем. Програмне забезпечення може аналізувати студента під час тестування, на таки параметри:

1) активність протягом періоду навчання (оцінки та відвідування);
2) час на виконання кожного питання;
3) відповіді сусіда.

На підставі цього неповному переліку критеріїв, програмний продукт може запідозрити студента в недобросовісних відповідях та перевірити його за допомогою ситуативного тесту, коли сусідам одночасно видається однакове питання, з подальшим інформуванням викладача.

Також, на фоні розглянутих проблем, можна вести мову про створення програмного засобу, який зможе полегшити аналіз досягнень студентів. Використовуючи методи машинного навчання є можливість спрогнозувати успішність студента, але перед цим в програму потрібно «навчити», включив в навчальну вибірку велику кількість даних інших студентів. Для більшої точності прогнозу, у процесі написання, буде використовуватись метод навчання з учителем (англ. supervised learning), як один із основних.

Як результат, користувачі інформацією зможуть корегувати навчальний план згідно наявних даних, використовуючи свої можливості більш ефективно. Використання наведених рішень та наявних технічних можливостей дозволить у майбутньому знизити важливість наведених, у статті, проблем.

Висновки до розділу 1

У першому розділі представлено теоретична частина випускної кваліфікаційної роботи, яка розкриває базові засади та поняття рейтингової системи, та сутність терміну рейтинг. Оглянуто базові принципи відображення рейтингового балу, та його сучасні ітерації в вигляді системи ECTS.

Також було розглянуто сучасні системи рейтингового оцінювання на прикладі українських вищих навчальних закладів, таких як КНТЕУ та НТУУ «Київський політехнічний інститут імені Ігоря Сікорського», та американських навчальних закладів. У США система оцінювання у навчальних закладах відрізняється тим, що рейтингова система може відрізнятися від предмету до предмету, побажань студента та інших факторів. При розробці програмних рішень це може стати перешкодою, яку потрібно буде вирішувати окремо. В наших же

університетах досить уніфікована система, але у кожного закладу своя. Це теж ставить перешкоди, але якщо виходити із рамок одного університету.

Озвучені наявні проблеми в існуючому рейтинговому оцінюванні, які пов'язнанні більшою мірою з людським фактором. Відношення до студенту, його громадська позиція може відігравати ключову роль в отримуванні освіти. Але також у розгляді було запропоноване рішеня за допомогою програмних засобів, які на сьогоднішний день вже достатньо розвилися. І для їх подальшого впровадження вже потрібно накопичуватти набори даних в електронному виді.

РОЗДІЛ 2. ТЕОРЕТИКО-МАТЕМАТИЧНІ ОСНОВИ ВИРІШЕННЯ ЗАДАЧ ПРОГНОЗУВАННЯ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ

2.1. Сучасні напрямки розвитку машинного навчання

Теорія машинного навчання зародилася практично одночасно з появою перших комп'ютерів, а протягом останніх 70 років машинне навчання є дисципліною, що активно розвивається. Постійний розвиток пов’язаний зі зростанням можливостей сучасних обчислювальних систем, ще більш стрімким зростанням обсягів даних, доступних для аналізу, а також постійним розширенням області застосування методів машинного навчання на широкий клас задач обробки даних. [13]

Машинне навчання тісно пов'язане (та часто перетинається) з обчислювальною статистикою, яка також зосереджується на прогнозуванні шляхом застосування комп'ютерів. Воно має тісні зв'язки з математичною оптимізацією, яка забезпечують цю галузь методами, теорією та прикладними областями. Машинне навчання іноді об'єднують з добуванням даних, де друга підгалузь фокусується більше на розвідувальному аналізі даних, і є відомою як навчання без учителя. Машинне навчання також може бути спонтанним, і застосовуваним для навчання та встановлення базових характерів поведінки різних суб'єктів, а потім застосовуваним для пошуку виразних аномалій.

Основна мета системи, яка навчається, — це робити узагальнення зі свого досвіду. Узагальнення в цьому контексті є здатністю машини, яка вчиться, працювати точно на нових, не бачених прикладах/задачах після отримання досвіду навчального набору даних. Тренувальні приклади походять з якогось загалом невідомого розподілу ймовірності (який вважається представницьким для простору випадків), і система, яка вчиться, має будувати загальну модель цього простору, яка дозволяє їй виробляти достатньо точні передбачення в нових випадках [28].

З кінця 90-их років баєсовский формалізм при описі алгоритмів машинного навчання отримав загальне визнання [26]. В рамках нього вдалося розробити ряд

спільних методів для оцінки апостеріорного розподілів, байєсівського виведення, автоматичного вибору моделі та ін. Не менш важливим успіхом байєсівського формалізму стала можливість успішного узагальнення результатів і методів класичного машинного навчання на абсолютно нові завдання [14].

Методи глибинного навчання (англ. deep learning) є спробою реінкарнації нейронних мереж, з кінця 80-их років минулого століття, які переживають кризу. Причинами кризи традиційних нейронних мереж стали [15]:

1) критична залежність якості настройки ваг мережі від вибору початкового наближення і, як наслідок, проблеми з відтворюваністю «успішних» результатів, які публікувались в наукових журналах;
2) велика схильність перенавчання укупі зі слабкими можливостями контролю узагальнюючої здатності мережі;
3) велику кількість локальних мінімумів функціонала якості, більшість з яких виявлялися поганими. З іншого боку, незаперечною сильною стороною нейронних мереж стало відкриття методу зворотного поширення помилки (англ. backpropagation), що дозволяв відслідковувати вплив внутрішніх шарів мережі на якість прогнозу прихованих змінних об'єктів навчальної вибірки.

Цей напрямок почав розвиватися на початку 21 століття. В його основі лежать нейронні мережі, які зазнали значні зміни:

1) У найбільш поширеній постановці всі змінні об'єктів передбачаються бінарними. Це полегшує моделювання залежностей між змінними об'єкта.
2) Кожен шар нейронної мережі спочатку навчається незалежно, проходячи процедуру попереднього навчання (англ. pre-training). Це дозволяє знайти гарне початкове значення для подальшого запуску алгоритму зворотного поширення помилки.
3) Кожен шар, в залежності від обраної моделі, являє собою обмежену машину Больцмана (англ. restricted Boltzmann machine) або надточну мережу (англ. convolutional network).

4) Для навчання використовуються сотні тисяч і мільйони об'єктів. такі гігантські вибірки дозволяють налаштовувати мережі з десятками тисяч параметрів, без ризику перенавчання. Навчені таким чином мережі, не просто дозволяють моделювати складні об'єкти (наприклад, тексти або зображення), але і генерують в процесі навчання інформативні ознакові описи, які можуть бути використані іншими, більш простими алгоритмами машинного навчання в якості спостережуваних змінних об'єкта.

Методологія глибинного навчання дозволила добитися небачених раніше результатів при навчанні на великих і надвеликих обсягах даних. В даний час вона є одним з найбільш перспективних шляхів розвитку машинного навчання.

Наступним напрямом в теорії машинного навчання є непараметричні байєсовські методи (англ. non-parametric Bayes). Традиційно, методи непараметричної статистики визначалися як розділ статистики, в якій число параметрів, що описують дані (наприклад, параметри щільності розподілу об'єктів) не фіксували, а зростає з ростом числа об'єктів.

Щоб зрозуміти даний метод потрібно розглянути задачу визначення числа кластерів (скупчень об'єктів) в зростаючої вибірці об'єктів. Актуальність задачі зумовлена тим, що загальноприйнятих методів визначення кількості кластерів, з яких складається навіть зафіксована вибірка, на сьогоднішній день не існує. Чим більше об'єктів надходить в наше розпорядження, тим на більшому диапазоні ми можемо знаходити в них структуру, виділяючи кластери схожих між собою об'єктів. При досить неоднорідній вибірці число кластерів має поступово збільшуватися в міру надходження нових об'єктів.

Виникає питання, чи можна задати уявлення про те, як швидко має рости число кластерів з ростом даних (щоб їх не було занадто багато або занадто мало) і як, дивлячись на вибірку об'єктів, врахувати ці уявлення. Формально, відповідь може буди дана, використовуючи апостеріорну ймовірність Баєса ($P(A|B)$), яка як раз і об'єднує наші апріорні вистави за поточними спостереженнями у формулі 2.1.

$$P(A|B) = \frac{P(B|A)*P(A)}{P(B)}, \quad (2.1)$$

де $P(A)$ - апріорна ймовірність гіпотези А;

$P(B|A)$ - ймовірність настання події В при істинності гіпотези А;

$P(B)$ - повна ймовірність настання події В.

У непараметричному випадку, нам необхідно задати розподіл над всіма можливими розбивками довільній кількості об'єктів. Такий розподіл (як і в інших непараметричних баєсовских методах) задається за допомогою випадкових процесів. В даному випадку, це процес Діріхле (англ. Dirichlet process), також відомий як процес китайського ресторану (англ. Chinese restraunt process) [16]. З його допомогою, вдасться не тільки розрахувати для будь-якого розбиття довільного числа об'єктів на кластери їх апріорну ймовірність, але і врахувати характеристики об'єктів (їх змінні, що спостерігаються), щоб перейти до апостеріорного розподілу на різноманітні розбиття. Як це часто буває при застосуванні баєсовских методів, апостеріорний розподіл має гострий пік, який відповідає стійкому розпиленню вибірки об'єктів на деяке число кластерів. Фактично, процес Діріхле дозволяє задавати розподілу над різними дискретними розподілами. При виведенні використовуються наближені методи Монте-Карло з марковськими ланцюгами (англ. Markov chain Monte Carlo) і методи варіаційного виведення (англ. variational inference). Описана схема допускає численні узагальнення на випадок ієрархій кластерів, множинних вибірок, і інше.

Ще однією областю машинного навчання є навчання з підкріпленням, призначене для навчання агентів (автономних модулів, самостійно приймають рішення в реальному часу на підставі наявних даних) в умовах невизначеності, що породжується, як неповнотою інформації про навколишнє обстановці, так і можливими діями інших агентів. Залежно від поточного стану середовища і дій агентів розраховується функція вигоди, яку отримає агент в наступний момент часу. У ролі спостережуваних змінних об'єкта виступає інформація, що

розташовується агентом, а прихованими змінними є довгострокові оцінки отриманої вигоди. важливим гідністю алгоритмів навчання з підкріпленням є можливість навчання агенту «з нуля» за рахунок балансованого поєднання режимів «Дослідження-використання» (англ. exploration-exploitation) і вивчання стратегій, що дозволяють жертвувати малим Зараз заради отримання більшої вигоди в подальшому. алгоритми навчання з підкріпленням знайшли широке застосування не тільки в таких традиційних областях як робототехніка, але і, наприклад, на фондових ринках [17].

Сьогодні також неможливо уявити машинне навчання без терміну «великі дані» (англ. big data). Словосполучення увійшло у вжиток наприкінці 2000-х років, коли став можливим збір і зберігання величезних обсягів даних. феномен великих даних можна наочно продемонструвати на прикладі великого адронного колайдеру (ВАК), який в минулому році виробив близько 25 петабайт експериментальних даних [18]. Традиційні методи машинного навчання не завжди застосовні для аналізу вибірок такого розміру, оскільки в них часто неявно передбачається, що вся вибірка поміщається в пам'яті комп'ютера, або ж вони мають недостатньо високі показники масштабованості (швидкості росту обчислювальної складності в залежності від розміру вибірки). Для подолання цих обмежень часто використовуються прийоми з наступних категорій:

1) Розпаралелювання. Незалежні частини алгоритму можуть виконуватися паралельними обробниками (в т.ч. на різних комп'ютерах) і в довільному порядку. У деяких випадках паралельної реалізації класичного алгоритму може бути досить для конкретної завдання. В тій чи іншій формі паралельність лежить в основі практично всіх обчислювальних систем, орієнтованих на великі дані. Цікаво, що паралельність накладає істотні обмеження на взаємодію між обробниками, так як накладні витрати на «спілкування» між ними може перевищувати виграш від використання великого обчислювального кластера.
2) Апроксимація. Відомо, що багато складних завдань можуть бути вирішені наближено з досить великою (а іноді і контрольованою) точністю,

достатньою для даного експерименту. Прикладами може служити фільтр Блума або наближений алгоритм пошуку найближчого сусіда, які допускають помилки першого роду, але мають істотно більше низьку обчислювальну складність ніж їх «точні» аналоги.

3) Стохастичність (рандомізація). При наявності великого числа незалежних об'єктів у вибірці, багато необхідні статистики можуть бути оцінені за випадковою підвибіркою, при цьому зберігаються теоретичні гарантії оптимальності та збіжності алгоритму. У разі, якщо вибирається підвибірка деякого фіксованого розміру, це дозволяє отримувати алгоритми з сублінійною масштабованістю. Найбільш відомим алгоритмом, де застосовується даний підхід, є метод стохастичного градієнтного спуску.

Останнім часом стали набирати популярність потокові алгоритми (англ. streaming algorithms, online learning), здатні навчатися у інкрементально, в режимі реального часу на постійно прибуваючих даних без необхідності зберігати їх в пам'яті. Попит на них виникає, як правило, в додатках, де дані надходять в таких кількостях і з такою швидкістю, що немає ніякий можливості зберігати їх, принаймні, надовго. З такими завданнями аналізу даних стикаються, наприклад, дослідники у Європейському центрі ядерних досліджень (CERN), де дані генеруються зі швидкістю 700 мегабайт в секунду [13].

Регіональний директор Oracle, Сергій Янчишин вважає, машинне навчання докорінно змінює навколишній світ, дозволяючи значно прискорити прийняття рішень. Можливо, це приваблює не так багато уваги ЗМІ, як роботи і автомобілі на автопілоті, але це зробить машинне навчання "технологією століття" для бізнесу. На думку аналітиків Gartner, до 2020 року технології штучного інтелекту будуть застосовуватися "майже в кожному новому програмному продукті" [19].

Потужний фактор, який допомагає в стрімкому рості для машинного навчання - це поширення хмарних технологій. Сьогодні ніхто вже не сумнівається в необхідності хмар, вони стають уже чимось само собою зрозумілим. Алгоритми машинного навчання потребують даних, в якомога більшій кількості даних з якомога більш широкого набору джерел. Чим більше вони навчаються по цим

джерелам, тим "розумніше" стають і тим більше їх потенціал при прийнятті рішень. І хмари дають ці великі дані.

Великі дані пропонують знайти багато цінного в процесі цифрової трансформації, в той час як хмара пропонує будівельні блоки для цього процесу. Машинне навчання, в свою чергу, стало першим по-справжньому промисловим інструментом для масштабного освоєння цих нових цінностей.

Привабливість машинного навчання в тому, що можливості його використання практично безмежні. Воно може застосовуватися всюди, де важливий швидкий аналіз даних, і надати просто-таки революційний ефект там, де важливо виявляти тенденції або аномалії в великих наборах даних - від клінічних досліджень до сфери безпеки і контролю за дотриманням стандартів [20].

2.2. Огляд алгоритмів машинного навчання для рейтингових оцінок

Методи машинного навчання можна розділити на 3 основні категорії: контрольоване (з учителем), неконтрольоване та з підкріпленням. Контрольоване навчання корисно в тих випадках, коли властивість (ярлик) є для певного масиву даних (навчального набору), але на даний момент воно відсутнє і повинно бути передбачене для інших випадків. Неконтрольоване навчання використовується для виявлення неявних відносин в даному немаркованих наборі даних. Навчання з підкріпленням - щось середнє між вищеописаними категоріями: є деяка форма зворотного зв'язку, доступна для кожного кроку або дії, але відсутня ярлик і повідомлення про помилку.

Для виконання поставленого завдання буде найбільш ефективно використовувати алгоритми першої категорії, оскільки маючи на руках рейтингові данні студентів можна «навчати» власний продукт робити прогнози.

Одним із перших алгоритмів, який можна застосувати для роботи з рейтинговими оцінками є дерево прийняття рішень (англ. Decision Tree Classifier). Це такий спосіб представлення правил в ієрархічній, послідовної структурі, де кожному об'єкту відповідає єдиний вузол, що дає рішення [29]. Область

застосування дерева рішень в даний час широка, але все завдання, які вирішуються цим апаратом можуть бути об'єднані в наступні три класи:

1) Опис даних: Дерева рішень дозволяють зберігати інформацію про дані в компактній формі, замість них ми можемо зберігати дерево рішень, яке містить точний опис об'єктів.
2) Класифікація: Дерева рішень відмінно справляються з завданнями класифікації, тобто віднесення об'єктів до одного з заздалегідь відомих класів. Цільова змінна повинна мати дискретні значення.
3) Регресія: Якщо цільова змінна має безперервні значення, дерева рішень дозволяють встановити залежність цільової змінної від незалежних (вхідних) змінних. Наприклад, до цього класу належать задачі чисельного прогнозування (передбачення значень цільової змінної).

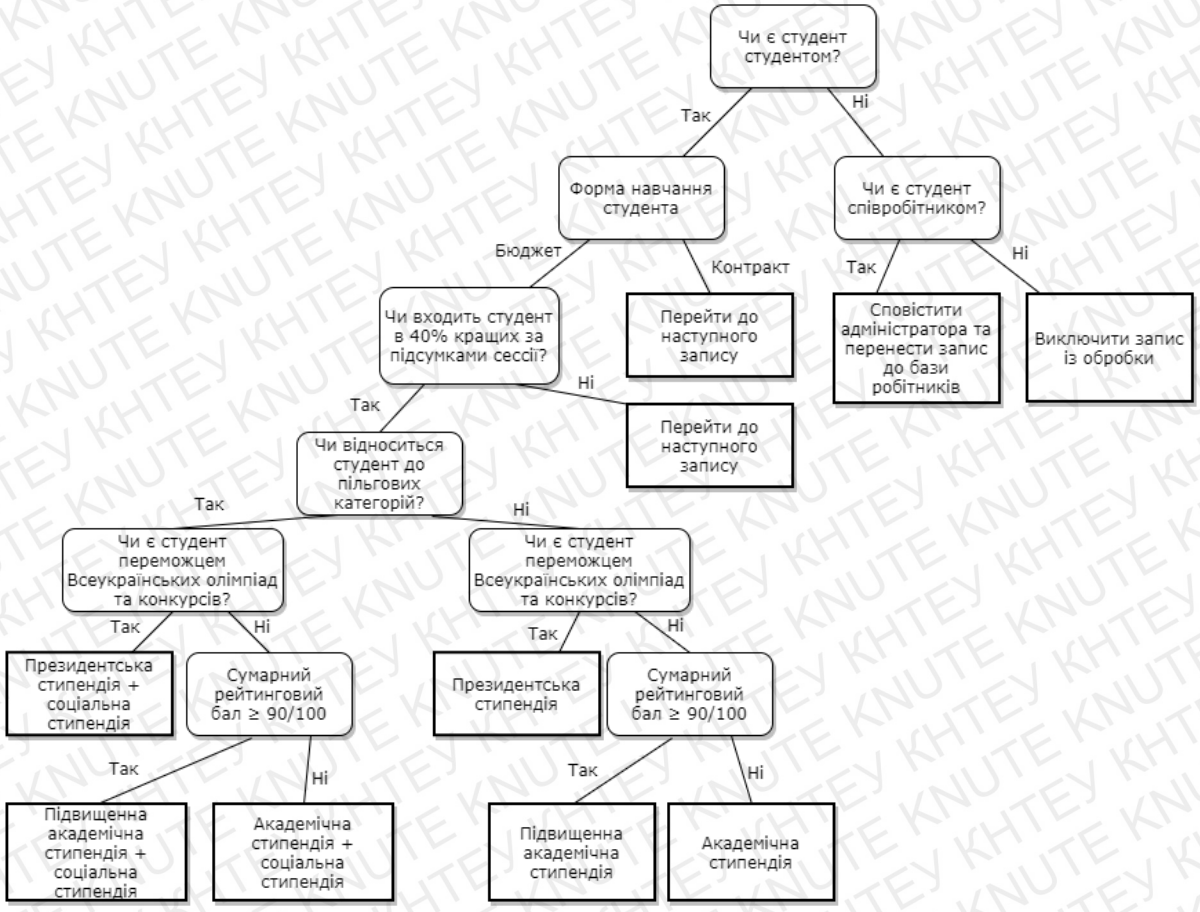

Рис. 2.1 Модель алгоритму дерева прийняття рішень для стипендіатів

Перевагами у використані дерев прийняття рішень для машинного навчання є:

1) швидкий процес навчання;
2) інтуїтивно зрозуміла класифікаційна модель;
3) висока точність прогнозу;
4) генерація правил в областях, де эксперту складно формалізувати свої знання;
5) побудова непараметричних моделей.

В силу цих та багатьох інших причин, методологія дерев рішень є важливим інструментом в роботі кожного фахівця, що займається аналізом даних, незалежно від того практик він або теоретик.

Наступним алгоритмом, який поверхово описувався в минулому розділі під заголовком непараметричних байєсовських методів є наївний баєсів класифікатор (англ. Naive Bayes). У порівнянні з іншими контрольованими методами машинного навчання, він є, мабуть, одним з найпростіших, але дивно потужним алгоритмом побудови моделей прогнозування з мічених навчальних наборів. Його прогностичні властивості часто були предметом теоретичних і практичних досліджень, і було доказано [25], що, незважаючи на наївне припущення байєсівського класифікатора умовної незалежності атрибутів даних класу, отримані моделі часто стійкі до такої міри, що вони збігаються або навіть перевершують інших більш складні методи машинного навчання. Для прикладу можна взяти завдання визначення статі на ім'я. Звичайно, щоб визначити стать можна створити великий список імен з мітками статі. Але цей список в будь-якому випадку буде не вичерпний. Для того щоб вирішити цю проблему, можна «натренувати» модель по маркованих іменам.

Нехай є рядок тексту O. Крім того, є класи C, до одного з яких ми повинні віднести рядок. Потрібно відшукати такий клас с, за якого ймовірність для цього рядка була би максимальною. Формулою можна записати так:

$$c = \arg max_c \, P(C|O), \tag{2.2}$$

де $\arg max_c$ - аргумент максимізації зі значенням с;

$P(C|O)$ - апостеріорна ймовірність гіпотези С при події О.

За допомогою основної формули (2.1) теореми Баєса [30], і переходимо до непрямих ймовірностей:

$$P(C|O) = \frac{P(O|C)*P(C)}{P(O)}, \quad (2.3)$$

Так як ми шукаємо максимум від функції, то знаменник нас не цікавить (він в даному випадку константа). Крім того, потрібно поглянути на рядок O. Зазвичай, немає сенсу працювати з усім рядком. Набагато ефективніше виділити з нього певні ознаки (англ. features). Таким чином формула (2.3) набуде вигляду:

$$P(C|o_1 o_2 \dots o_n) = \frac{P(o_1 o_2 \dots o_n|C)*P(C)}{P(o_1 o_2 \dots o_n)}, \quad (2.4)$$

де o_n - рядок тексту, під порядковим номером n;

Тут можна зробити «наївне» припущення про те, що змінні O залежать тільки від класу C, і не залежать один від одного. Це сильно спрощення, але найчастіше це працює. Чисельник набуде вигляду:

$$P(C)P(o_1|C)P(o_2|Co_1) \dots P(o_n|Co_1 o_2 \dots o_n) =$$
$$P(C)P(o_1|C)P(o_2|C) \dots P(o_n|C) = P(C)\prod_i (o_i|C), \quad (2.5)$$

І фінальна формула буде мати вигляд:

$$c = \arg max_{c \in C}\, P(c|o_1 o_2 \dots o_n) =$$
$$\arg max_{c \in C}\, P(c)\prod P(o_i|C), \quad (2.6)$$

Все, що потрібно зробити – порахувати ймовірності $P(C)$ та $P(O|C)$. Обчислення цих параметрів і називається тренуванням класифікатора. Аналогічним чином з його допомогою можна прогнозувати кілька різних класів на основі безлічі ознак. Цей алгоритм в основному використовується в області класифікації текстів і при вирішенні задач багатокласової класифікації.

Одним з обмеженням класифікатору є припущення про незалежність ознак. На практиці набори повністю незалежних ознак зустрічаються вкрай не часто.

Із плюсів можна виділити класифікацію, в тому числі багатокласову, яка виконується легко і швидко. Коли допущення про незалежність виконується, наївний Баєс перевершує інші алгоритми, такі як логістична регресія і при цьому вимагає менший обсяг навчальних даних. Класифікатор краще працює з категорійними ознаками, ніж з безперервними. Якщо знову говорити проо недоліки, то може трапитись те, що в тестовому наборі даних є певне значення категорійного ознаки, яке не зустрічалося в навчальному наборі даних. Тоді модель присвоїть нульову ймовірність цього значення і не зможе зробити прогноз. Це явище відоме під назвою «нульова частота» (англ. zero frequency). Дану проблему можна вирішити за допомогою згладжування.

Наступними двома алгоритмами, які можна оглянути в рамках роботи є використання логістичною регресії та методу лінійного дискримінантного аналізу. Алгоритм являє собою ще один статистичний метод класифікації, використовуючи лінійний дискримінант Фішера [31]. На відміну від звичайної регресії, в методі логістичної регресії не проводиться передбачення значення числової змінної виходячи з вибірки вихідних значень. Замість цього, значенням функції є ймовірність того, що дане початкове значення належить до певного класу. Для прикладу ми будемо використовувати бінарну класифікацію і ймовірність, яку ми будемо визначати. Тобто P+ ймовірності того, що деяке значення належить класу "+", та 1 - P+, що до класу "-". Таким чином, результат логістичної регресії завжди знаходиться в інтервалі (0, 1).

Основна ідея логістичної регресії полягає в тому, що простір вихідних значень може бути розділений лінійної кордоном (тобто прямою) на дві

відповідають класам області. Мається на увазі під лінійної кордоном те, що якщо ми маємо не більше двох вимірів - це просто пряма лінія. У разі трьох - площину, і так далі. Ця межа задається в залежності від наявних вихідних даних і навчального алгоритму. Щоб все працювало, точки вихідних даних повинні розділятися лінійної кордоном на дві вищезазначених області. Якщо точки вихідних даних задовольняють цю вимогу, то їх можна назвати лінійно розподіленими.

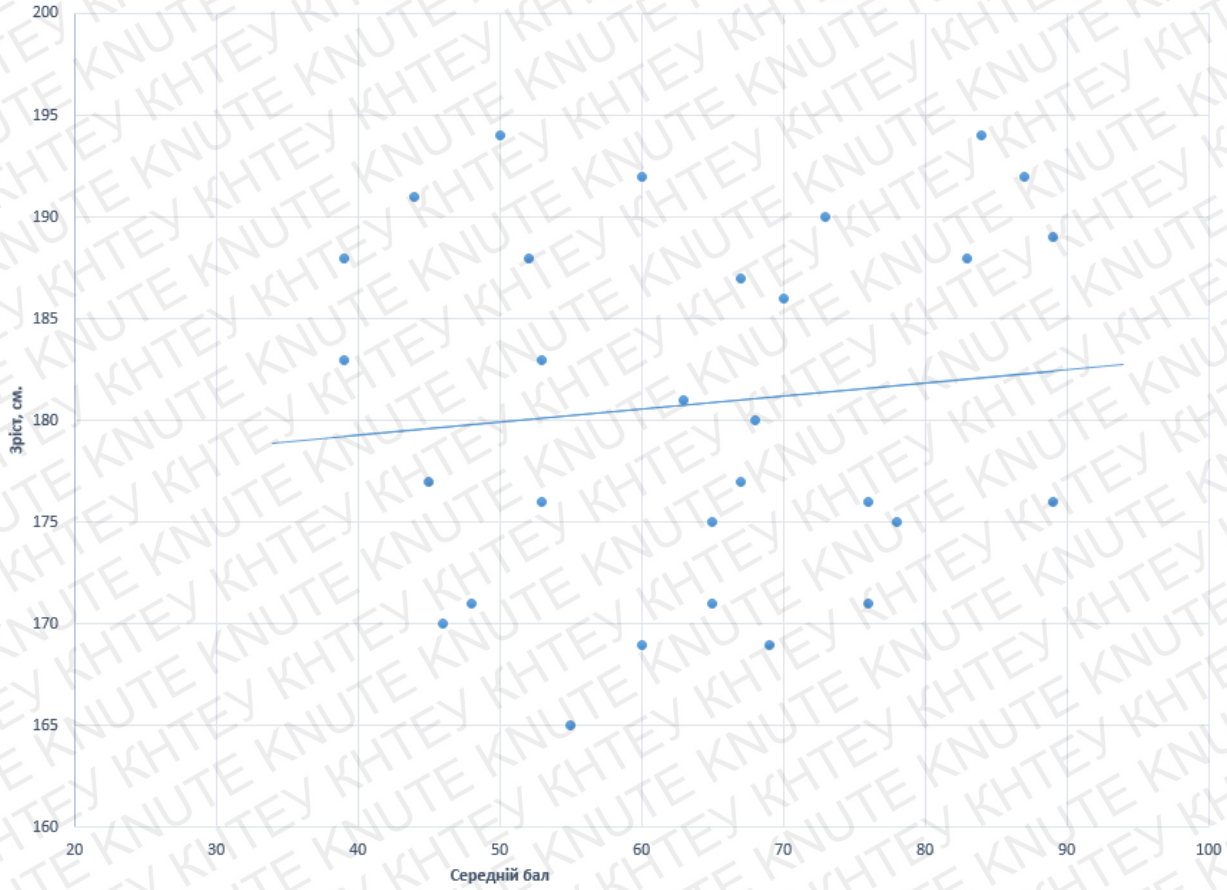

Рис 2.2 Поділ тестових даних лінійним дискримінантом

Зазначена пряма на рисунку 2.2 називається лінійним дискримінантом, так як вона є лінійної з точки зору своєї функції, і дозволяє моделі проводити розподіл, дискримінацію точок на різні класи.

Як тестову задачу візьмемо визначення майбутньої наявності студента за двома факторами – його зріст, та його середній бал. Для цього було згенеровано невеличку табличку, на 30 записів в яких міститься зріст студента (у сантиметрах), його середній бал (від нуля до сотні), та його наявність на минулій

парі (1 – наявний, 0 – відсутній). Таблиця з використаними даними знаходиться у додатку Б.

Для вирішення задачі була написана невеличка програма, яка демонструє точність, матрицю помилок, усереднені значення роботи моделі та прогнозоване значення. Результат її роботи з використанням методу логістичної регресії відображено на рисунку 2.3.

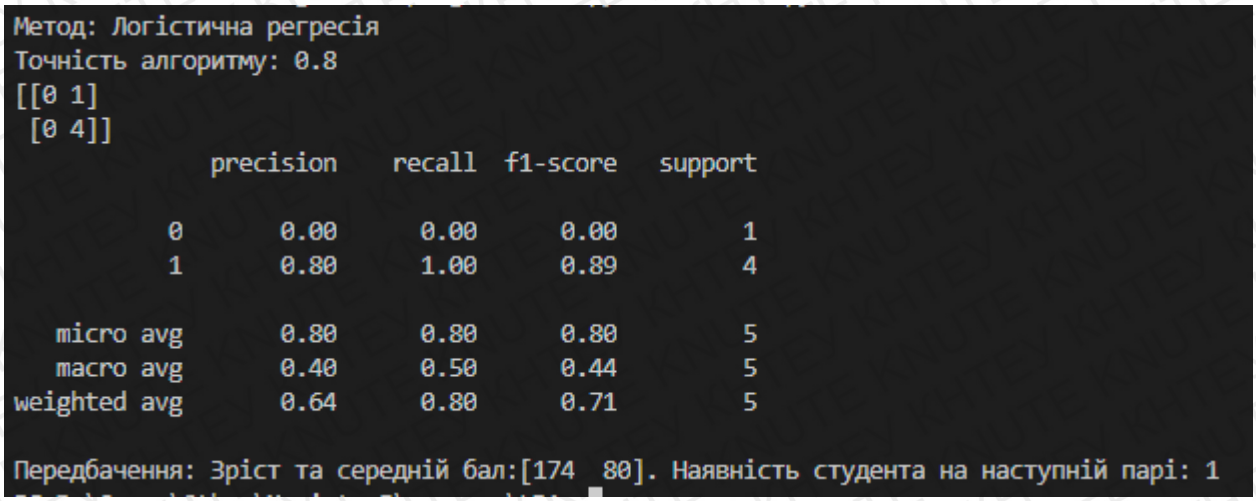

Рис 2.3 Робота алгоритму, який використовує логістичну регресію.

Розглянемо функцію *f(x)*, де *x* - точка даних навчальної вибірки. У простій формі *f(x)* можна описати так: якщо *x* є частиною класу "+", $f(x) = P_+$ (тут P_+ - вихідне значення, отримане з моделі логістичної регресії). Якщо *x* є частиною класу "-", то $f(x) = 1 – P_+$.

Функція *f(x)* проводить кількісну оцінку ймовірності того, що точка навчальної вибірки класифікується моделлю правильним чином. Тому, середнє значення для всієї навчальної вибірки показує ймовірність того, що випадкова точка даних буде правильно класифікована системою, незалежно від можливого класу.

Лінійний дискримінантний аналіз також використовується для пошуку лінійної комбінації змінних, найкращим чином поділяючої два або більше класів. Лінійний дискримінантний аналіз сам по собі не є алгоритмом класифікації, хоча і працює з інформацією про належність об'єкта до одного з класів. Однак найчастіше результат роботи лінійного дискримінантного аналізу

використовується, як частина лінійного класифікатора. Іншим можливим застосуванням є зниження розмірності вхідних даних перед застосуванням нелінійних алгоритмів класифікації [32]. Лінійний дискримінантний аналіз тісно пов'язаний з дисперсійним та, як було показано вище, регресійним аналізом, які також намагаються виразити будь-яку залежну змінну через лінійну комбінацію інших ознак або вимірювань. У цих двох методах залежна змінна - чисельна величина, а в методі вона є величиною номінальної (міткою класу). Крім того, лінійний дискримінантний аналіз має схожі риси з методом головних компонентів і факторного аналізу, які шукають лінійні комбінації величин, найкращим чином описуючі дані. Роботу за методом дискримінантним аналізом зображено на рисунку 2.4.

Рис 2.4 Робота алгоритму бінарної класифікації за методом лінійного дискримінантного аналізу

При обчислені дискримінанту демонстраційна програма підраховує середні для кожного з двох класів, а потім використовує їх для розрахунку матриці помилок, а після цього вже нормалізує значення. Останнім кроком алгоритм порівнює нові данні з навчанню моделлю і видає значення, що хлопець із середнім балом на наступну пару буде присутній.

Дискримінантний аналіз працює шляхом створення однієї або більше лінійної комбінацій предикторов, отримуючи нову приховану змінну для кожної функції. Ці функції називаються дискримінантний функціями. Число можливих функцій дорівнює або *Ng-1*, де *Ng* = число груп, або *p* (числу предикторов), в

залежності від того, яке з чисел менше. Перша створена функція максимізує різницю між групами по цій функції. Друга функція максимізує різницю по цій функції, але не повинна корелювати з попередньою функцією. Процес триває створенням послідовності функцій з вимогою, щоб нова функція не корелювала з усіма попередніми.

Одним із найпростіших алгоритмів для класифікації є метод найближчих сусідів, або k-найближчих сусідів (анг. k-Nearest Neighbor). Він заснований на оцінюванні подібності об'єктів. Для нового об'єкта x метод передбачає знайти найближчі до нього об'єкти *x1, x2, ... xn* і побудувати прогноз по їх міткам.

Цей метод є прикладом навчання на основі екземплярів (англ. instance-based learning) або навчання без узагальнення (англ. non-generalizing learning): він не намагається побудувати загальну внутрішню модель, а просто зберігає екземпляри навчальних даних. Класифікатор обчислює більшість від загального числа найближчих сусідів кожної точки: точці запиту присвоюється клас даних, який має найбільшу кількість представників у межах найближчих сусідів точки [33].

Для класифікації кожного з об'єктів тестової вибірки алгоритм послідовно виконує наступні операції:

1) Обчислити відстань до кожного з об'єктів навчальної вибірки
2) Відібрати k об'єктів навчальної вибірки, відстань до яких мінімально
3) Об’єкту призначається той клас, який найчастіше трапляється серед k найближчих сусідів

Розглядаючи цей метод, можна взяти задачу розподілу стипендій, яка пізніше буде використовуватись у програмному забезпечені для розділу 3. Для аналізу використаємо данні з таблиці 2.1.

Таблиця 2.1

Таблиця розподілення стипендій по студентам

Студент	Рейтинговий бал	Додатковий бал	Стипендія, грн
A	92	0,5	1889

B	88	0	1330
C	78	5	0
D	86	8	1889
E	60	0	0
F	87	0,5	?

Потрібно знайти, яку стипендію отримає студент F. Для наочності будується графік для обчислення відстаней до кожного з об’єктів, зображений на рисунку 2.5.

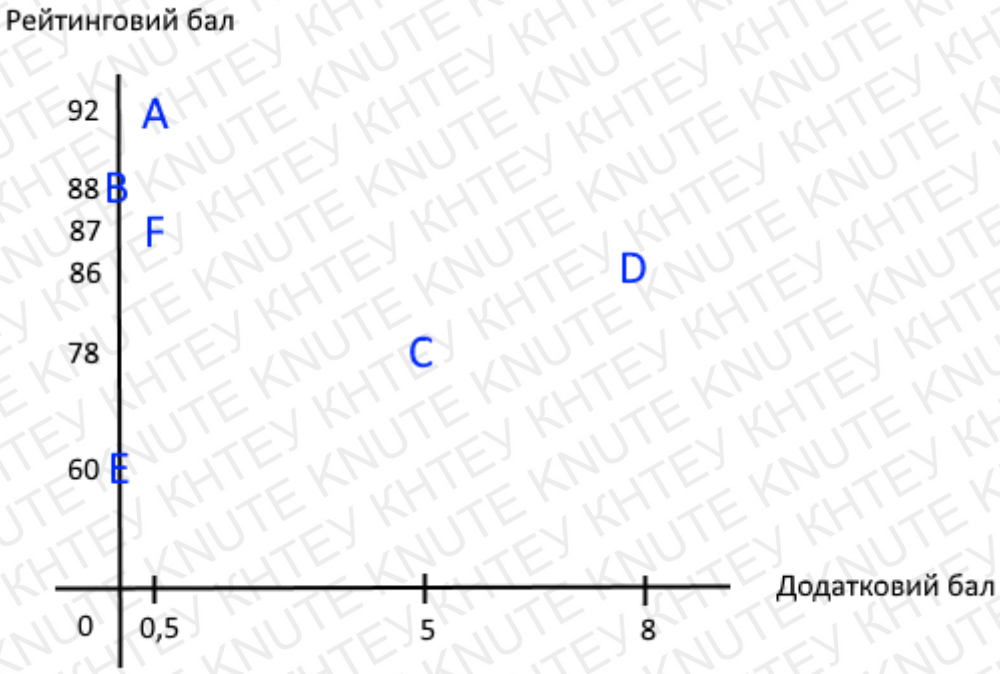

Рис 2.5 Розподілення відстаней між студентами табл. 2.1

Замість прямолінійного усереднення можна обчислити середньозважене з урахуванням того, наскільки далеко знаходяться зразки. Чим більше відстань, тим менше вага. В даному прикладі студенту B можна приписати найбільший ваговий коефіцієнт (0,5), а A і C - ваги поменше (0,4 та 0,1). Для розрахунку стипендії беремо формулу 2.7.

$$S = \sum_n (W_n * P_n), \quad (2.7)$$

де S - стипендія студента F;

W_n - ваговий коефіцієнт студента n ($\sum_n (W_n) = 1$);

P_n - стипендія студентів n;

n - порядковий номер рядка у табличному масиві.

Підставивши наші числа у формулу, S буде дорівнювати 1420,6 гривням. Оскільки розмір стипендії є фіксованою величиною, то алгоритм прирівняє до найближчого значення (1330-1889 грн) і тут, тому студент отримає звичайну стипендію. Звичайно, цю задачу можна розв'язати і аналітично, бо у нас є знання про те, що при сумі балів вище 90 студент отримує підвищену стипендію, а при меншій кількості балів – звичайну, або зовсім не отримує. Але подібний приклад є демонстрацією того, як працює цей алгоритм. І на більш високих вибірках він може проявити себе набагато краще.

Метод найближчих сусідів відноситься до числа оперативних методів, тобто дані можна додавати в будь-який момент, на відміну від наступного алгоритму, який буде розглянуто в роботі - методу опорних векторів, який потрібно заново навчати при кожній зміні даних. Більш того, при додаванні нових даних не потрібні взагалі ніякі обчислення; досить просто включити дані в набір. Основний недолік алгоритму полягає в тому, що для прогнозування йому потрібні всі дані, на яких проводилося навчання. Якщо в наборі мільйони зразків, то на це витрачається не тільки пам'ять, а й час - для вироблення кожного прогнозу доводиться порівнювати новий зразок з кожним з наявних, щоб знайти найближчі. Для деяких рішень це суттєве зниження продуктивності.

Ще один недолік – складність пошуку відповідних коефіцієнтів масштабування. Хоча існують способи автоматизації цієї процедури, але на великому наборі даних для оцінки тисяч можливих коефіцієнтів і перехресного контролю можуть знадобитися чималі обчислювальні ресурси. Якщо доводиться випробувати багато різних змінних, то для знаходження відповідного поєднання

масштабних коефіцієнтів, можливо, буде потрібно переглянути мільйони комбінацій.

Як зазначалося вище, для вирішення наявних задач можна використати метод (машина) опорних векторів (англ. support vector machine, SVM). Це такий набір алгоритмів, що використовуються для задач класифікації та регресійного аналізу. З огляду на те, що в N-вимірному просторі кожен об'єкт належить одному з двох класів, алгоритм генерує (N-1)-мірну гіперплоскість з метою поділу цих точок на 2 групи. Це можна зобразити на папері, малюючи точки двох різних типів, які можна лінійно розділити. Крім того, що метод виконує сепарацію об'єктів, алгоритм підбирає гіперплоскість так, щоб та характеризувалася максимальним віддаленням від найближчого елемента кожної з груп. Машина опорних векторів є однією з найбільш популярних методології навчання по прецедентах, запропонована Вапніком Н.В [21].

Для прикладу і демонстрації методу візьмемо дані із додатку Б, які використовувалися для лінійного дискримінанту, де в залежності від зросту та середньої оцінки прогнозувалась майбутня наявність студента на парі. Нехай на рисунку 2.6 точки, що належать різним класам, можна розділити за допомогою прямої лінії.

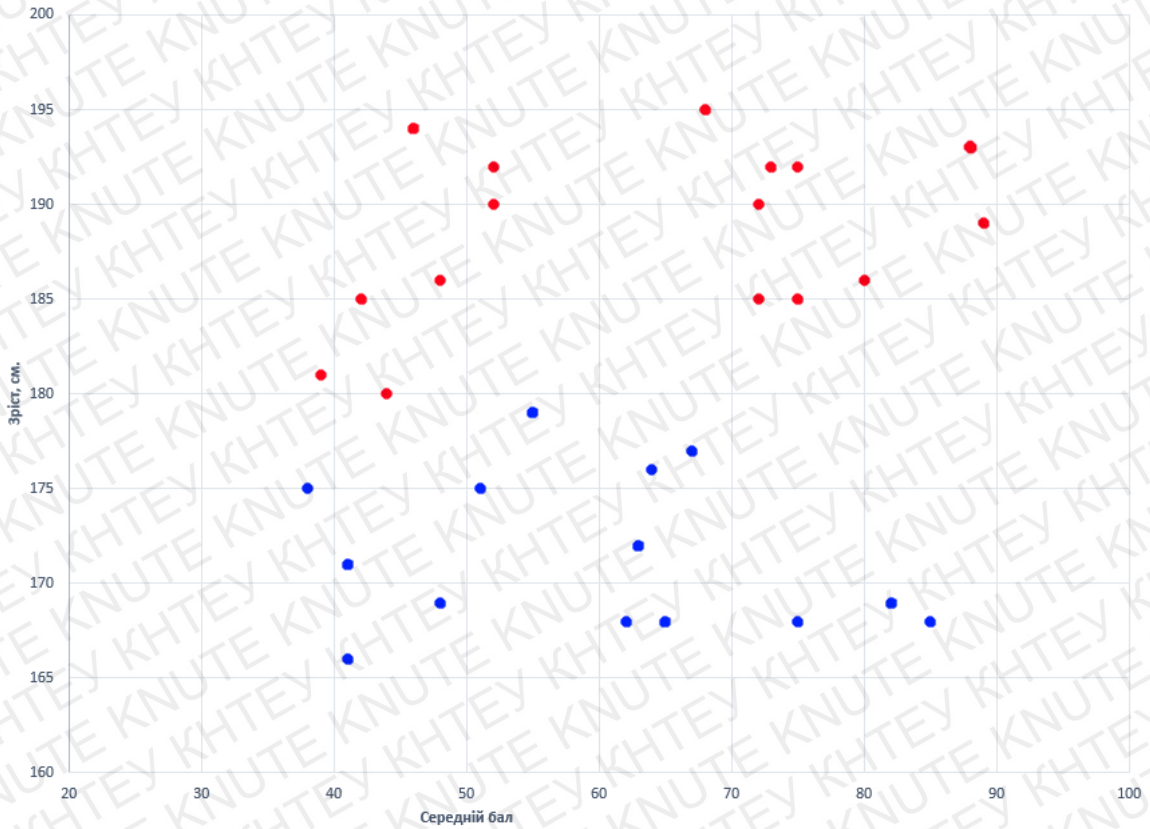

Рис 2.6. Точки поділенні на дві групи

Очевидний спосіб вирішення завдання в такому випадку провести пряму так, щоб всі точки одного класу лежали по одну сторону від цієї прямої, а всі точки іншого класу були на протилежному боці. Тоді щоб класифікувати невідомі точки нам потрібно просто подивитися з якого боку прямої вони виявляться. Звучить легко, але насправді все складніше, адже можна провести безліч прямих, які задовольняють умову. Деякі з них зображені на рисунку 2.7.

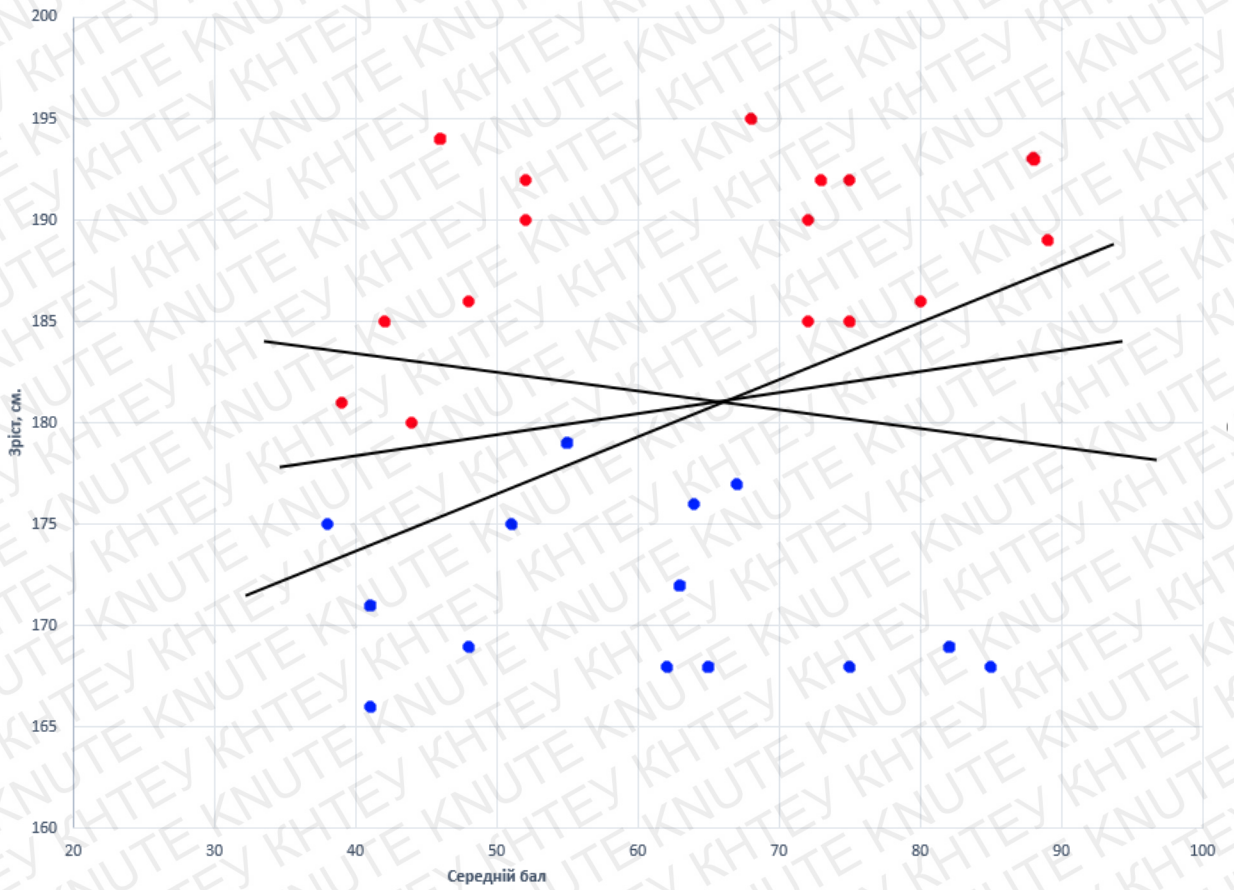

Рис 2.7 Прямі, які можна провести між поділеними точками

Інтуїтивно бачимо, що найкраще вибрати пряму максимально віддалену від наявних точок. Тут постає питання термінології, про відстань між прямою і безліччю точок. У методі опорних векторів цим відстанню вважається відстань між прямою і найближчою до неї точкою з безлічі інших. Саме таку відстань і максимізує в методі опорних векторів. Приблизна демонстрація його роботи знаходиться на рисунку 2.8.

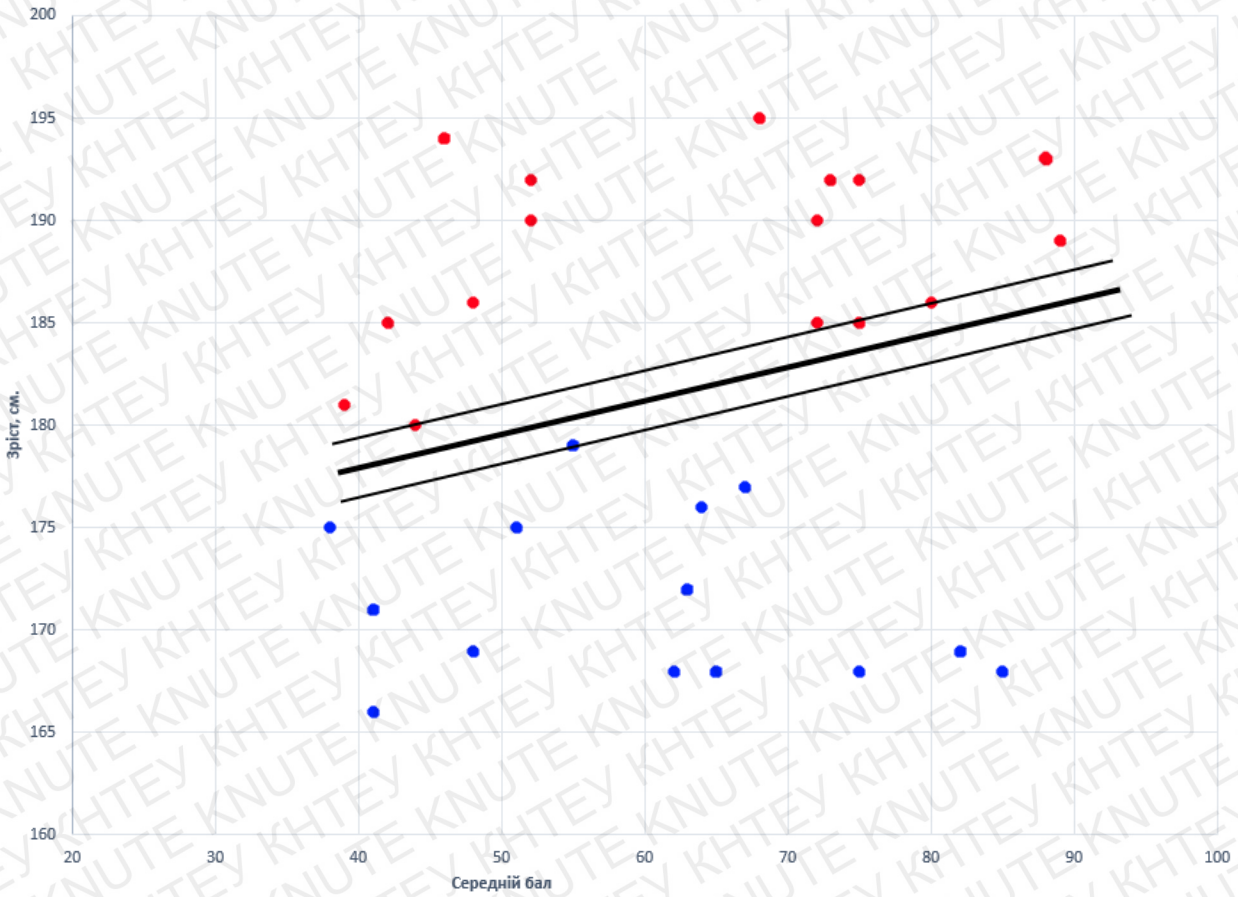

Рис 2.8. Відділена пряма, яка є максимально рівновіддаленою

Геометрично це виглядає так, як ніби ми намагаємося провести пряму чітко по центру, між двома множинами. Виявляється, що така пряма існує лише одна. І її не так вже складно знайти. Найближчі до цієї прямої точки з множин називаються опорними векторами. У найпростішому випадку, саме таку пряму і знаходить метод опорних векторів. Якщо обійтися без формул, все зводиться до деякої системи рівнянь, яка вирішується методами квадратичного програмування.

Якщо виділяти мінуси цього методу, то машина опорних векторів дуже чутлива до шумів – точок, які дуже тяжко віднести до якоїсь з категорій, та заважають проведенню прямої. У цьому випадку на допомогу до нас приходить, так званий, алгоритм з м'яким зазором (англ. soft marging). Математично це виражається введенням додаткових параметрів в систему рівнянь, а по суті, ми просто призначаємо якийсь штраф, за кожну точку, яка опинилася на чужині. Використовуючи такий підхід ми можемо працювати навіть з лінійно нероздільними даними.

2.3. Модель рейтингового оцінювання студентів на основі алгоритмів класифікації

Рейтингове оцінювання студентів проводиться з давніх часів, але на сьогоднішній день велику кількість аналітичної роботи можна перекласти на програмне забезпечення, яке з розвитком теорії машинного навчання може виконувати роботу з такою ж точністю, як і викладач. Завданням цієї випускної кваліфікаційної роботи стоїть розробка програмного забезпечення, яке зможе самостійно, без використання прямих алгоритмів, адаптуватися до наявних даних, аналізувати їх та видавати точний прогноз. Оскільки методи машинного навчання не можуть забезпечити 100% точність, потрібно вибрати такий алгоритм для певного набору даних, який буде забезпечувати максимально точний прогноз.

Однією з важливіших умов правильного функціонування моделі є правильно скомпонована база даних. Це може буди невеличкий текстовий файл у форматі csv (англ. Comma-Separated Values), або віддалена база даних успішності студентів по всій Україні. На цьому етапі йде формування першої проблеми, яка стоїть тими, хто використовує методи машинного навчання – недостача даних. Хоча теорія машинного навчання іде поруч з використанням теорії великих даних (англ. big data), наведений приклад з даними студентів є утопічним, тому що, нажаль, повсюдно не введена повна електронна звітність і більшість даних студентів залишаються на папері. Тому при формуванні моделі та подальшого програмного рішення, будуть використовуватись невеличкі масиви даних.

Маючи данні на руках, постає інша проблема, що полягає в правильному вибору категорій, які впливають на фінальний прогноз алгоритму. У минулому підрозділі використовувався приклад, в якому прогнозувалася майбутня присутність студента по його зросту та середньому балу. Фактично, ця задача використовувалася для демонстрації роботи алгоритмів, а самі дані були згенеровані алгоритмом випадкових чисел на заданій площині, тому результати роботи не несуть під собою наукової цінності. Але якщо дані були справжні? Ситуація би сильно не змінилася, оскільки немає прямої залежності росту студента від його бажання прийти на пару. Можна підібрати інший показник,

наприклад, наявність роботи, яка може корелювати з середнім балом. Прогноз по цим категоріям вже має право на існування, але виходячи з першої проблеми, навряд чи буде можливим отримати ці дані.

Для формування моделі оцінювання в цій роботі використовувалися наявні данні рейтингового оцінювання студенів двох навчальних закладів України – ХНЕУ та КНТЕУ. Два набори даних використовувалися на етапі підготовки фінальної бази даних та для порівняння ефективності роботи алгоритмів. В обох університетах використовуються різні підходи до розрахунку кінцевого рейтингового балу, тому на етапі підбору моделі буде віддане перевагу той, що продемонструє максимальну точність.

Згідно сформованому завданню та наявних даних, навчена модель буде прогнозувати отримання студентом стипендії. Використання методів машинного навчання не потребують знання точних коефіцієнтів та формул розрахунків рейтингового балу в закладах, які наведенні вище. Але алгоритми потребують правильних відповідей, оскільки буде використовуватись категорія навчання з вчителем, для більшої точності прогнозу.

Після закінчення робіт з формування бази для тренування моделі, потрібно приступити до написання програмного коду, який буде підставляти наші данні під шість алгоритмів, які описувались в підрозділі 2.1. Спочатку, ми даємо команду прочитати ці данні, після чого ми ділимо на дві частини – перші рядки відповідають за ознаки, яким будуть призначенні певні ваги у формуванні останнього рядка. Оскільки первинні дані використовуються для тренування, він буде заповнений правильною відповіддю. Після цього сформований масив поділяється, з певним відношенням, на данні для тренування та підтвердження (валідації) правильності роботи моделі. Відношення корегується в процесі вибору моделі окремою змінною та пізніше буде демонструвати точність роботи, після вибору моделі.

Данні для тренування відправляються в окрему функцію, після чого розпочинається цикл, де відбувається десятикратна перехресна перевірка для кожної з шести моделей. Перевірка ділить дані на треновані 10 частин, 9 з яких

використовуються для тренування, а 1 для тестування, і цей алгоритм повторюється для кожної комбінації відношень (8/2, 7/3, …). Кінцевим результатом є текстовий вивід точності роботи моделей. Для початку тестування пройшли дані ХНЕУ, результат якого на рисунку 2.9.

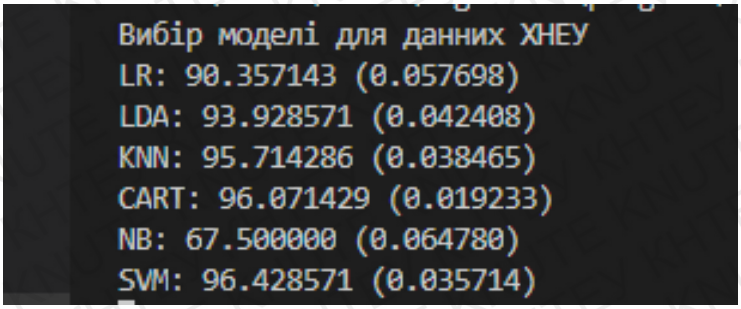

Рис 2.9 Результативна точність роботи шести моделей даних ХНЕУ

Для прогнозування в цій базі використовувалися три класифікатори:

1) Форма навчання (1 – контракт, 0 – бюджет)
2) Середній бал
3) Додатковий бал

Підвищена стипендія нараховується при рейтинговому балі >=90, звичайна >=75, або якщо менше – зовсім не нараховується. Сам рейтинговий бал (R) розраховується за формулою 2.7.

$$R = avg(B) * 0.9 + \frac{A}{10}, \tag{2.8}$$

де avg(B) - середній бал студента;
A - додатковий бал студента.

Звісно, ця формула не записана у програмі, а самого рейтингового балу в даних нема, тому модель буде намагатися це зробити сама. Майже всі моделі демонструють високу точність, окрім наївного Баєсу. Можливо алгоритм присвоїв нульові ймовірності деяким значенням. Як це вирішити було описано в минулому підрозділі, але при нормальній роботі інших моделей це не потрібно. Для наочності результати продемонстровані у графіку на рисунку 2.10.

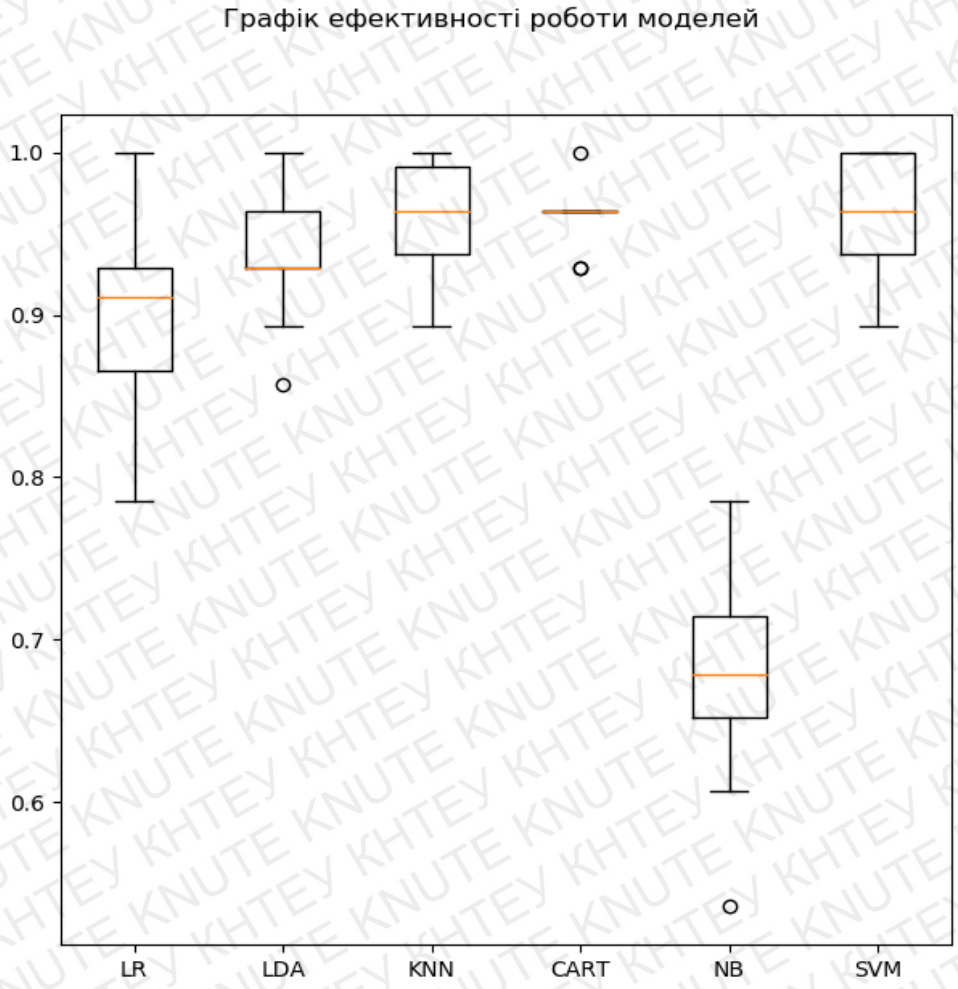

Рис 2.10 Графік ефективності роботи моделей з даними ХНЕУ

Найбільш точним для цих даних буде використання моделі з деревом прийняття рішень (~96% точність прогнозу).

Тепер черга бази з КНТЕУ. Алгоритм розрахунку рейтингового балу є стандартним, але при цьому стипендія видається певному проценту студентів у групі, і в наявній базі він дорівнює приблизно 45%, тому і набір класифікаторів відрізняється:

1) Номер групи (n = 1-9, n.1 – групи магістрів)
2) Додатковий бал
3) Середній бал

Попередньо тренувальна база мала оцінку студента по кожному предмету, але це лише ускладнювало навчання та знижувало процент точності, оскільки кожна група студентів має різну кількість навчальних предметів.

```
Вибір моделі для данних КНТЕУ
LR: 69.568106 (0.052587)
LDA: 74.252492 (0.052409)
KNN: 84.313400 (0.041753)
CART: 80.797342 (0.061583)
NB: 81.279070 (0.022134)
SVM: 81.976744 (0.038806)
```

Рис 2.11 Результативна точність роботи шести моделей даних КНТЕУ

На рисунку 2.11 видно, що точність роботи з цим набором даних нижче, ніж з попереднім. Моделям подібне розпізнавання далося складніше, оскільки в цьому наборі даних видача стипендій розподіляється по кількості найкращих в групі, тому точність впала. Найгірше впоралась модель з лінійною регресією, оскільки вона погано працює не з числовими класифікаторами, що також видно на графіку рисунка 2.12.

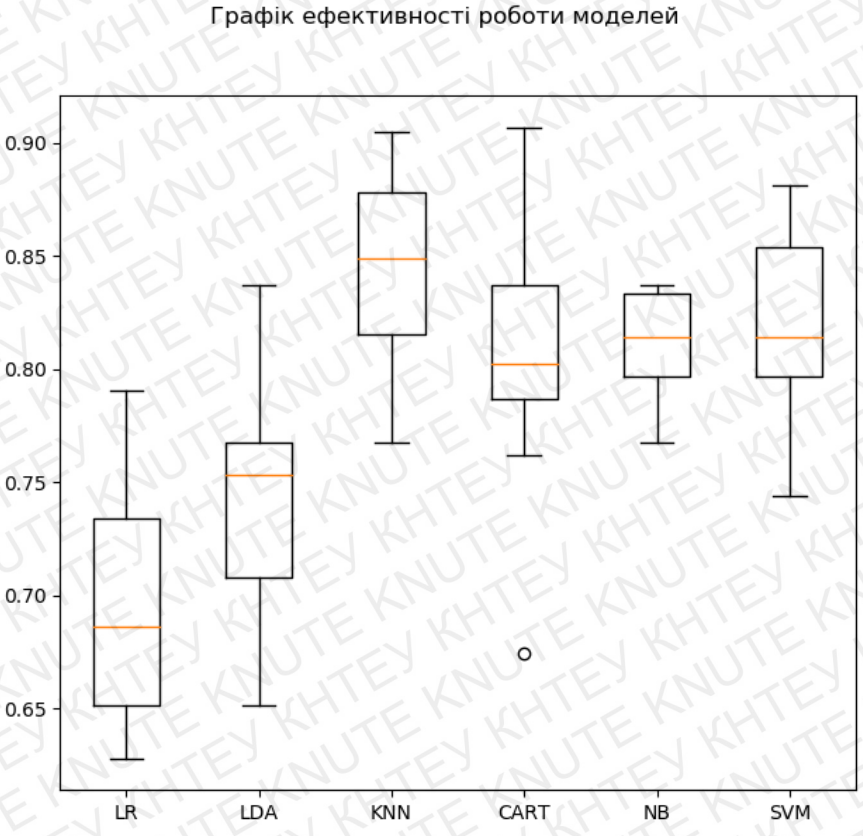

Рис 2.12 Графік ефективності роботи моделей з даними КНЕУ

Найкращим тут себе показала модель з методом k-найближчих сусідів (~84% точність прогнозу). Оскільки при подальшій роботі будуть використовуватись дані КНТЕУ, то ми використаємо цю модель для майбутніх

прогнозів. Однак, суттєвим мінусом є те, що процент отримувачів стипендій може змінюватись кожен рік [11], тому при змінах у законодавстві модель потрібно заново навчати.

Висновки до розділу 2

У другому розділі представлена аналітична частина випускної кваліфікаційної роботи, яка є продовженням теоретичного першого та закладає базові понятя про машинне навчання, використовані алгоритми та формуть модель. Було розглянуто суть машинного навчання, його появу, мету та розвиток.

Одним із основних напрямків розвитку теорії машинного навчаня стало глибинне навчання, яке виводить традиційні штучні нейронні мережі із кризи, в якій вони були до розвитку машинного навчання. Зараз воно тісно повязанне з нейронними мережами. Іншим напрямком є розвиток непараметричних статистичних алгоритмів, які використовуютть теорему Т. Баєса, і на його основі проводится навчання. Також було розглянуте навчання з підкріпленням (учителем), яке зараз є найбільш точним і буде використовуватись для написання програмного рішення. Потрібно не забувати й про великі данні, які на сьогодні ефективно аналізуються тількі завдяки машинному навчаню.

В рамках формування моделі розглянуто шість алгоритмів, два з яких є лінійними (логістична регрессія та лінійний дискримінантний аналіз), а інші чотири – нелінійними (найближчи сусіди, дерево рішень, наївний баєс та опорні вектори). До кожного цих алгоритмів була розглянута практична задача, для наочності, а також плюси з мінусами. Також проаналізован процесс формування найбільш результативної моделі машинного навчання для рейтингових оцінок. Були розглянуті основні проблеми, які можуть заважати ще на етапі формування тренувальної вибірки. При аналізі результативності навчання було використанно різні вибірки від двох університетів, в котрих відрізніявся спосіб вичислення рейтингового балу та критерії для видачі стипендіі. Було рокрито повний

алгоритм роботи програми, що вираховує ефективність роботи кожної моделі і вибирає найєфективнішу з них.

РОЗДІЛ 3. РОЗРОБКА АВТОМАТИЗОВАННОЇ СИСТЕМИ РЕТИНГОВОГО ОЦІНЮВАННЯ СТУДЕНТІВ

3.1. Інструменти, методи та технології розробки автоматизованої рейтингової системи оцінювання

На сьогоднішній день розробка програмного забезпечення з використанням технологій машинного навчання не є чимось новим. Раніше алгоритми та методи, описані в другому розділі випускної кваліфікаційної роботи використовувались як самописні набори математичних функцій, але з часом з'явилися комплексні шаблони (англ. Framework), які полегшили розробку і користуватись ними можна навіть початківцям з мінімальними математичними знаннями.

В роботі буде виділена лише невелика кількість тих шаблонів та рішень, які можуть використовуватись для написання автоматизованої рейтингової системи.

Першим буде Apache Spark - це фреймворк з відкритим сирцевим (джерельним) кодом для реалізації розподіленої обробки неструктурованих і слабкоструктурованих даних, що входить в екосистему проектів Hadoop. Завдяки своїй зростаючій бібліотеці алгоритмів, Spark став надійним інструментом машинного навчання, який можна застосувати для високошвидкісної обробки даних [34]. На відміну від класичного обробника з ядра Hadoop, що реалізує дворівневу концепцію MapReduce з дисковим сховищем, Spark використовує спеціалізовані примітиви для рекурентної обробки в оперативній пам'яті, завдяки чому дозволяє отримувати значний виграш в швидкості роботи для деяких класів задач, зокрема, можливість багаторазового доступу до завантажених в пам'ять призначених для користувача даних робить бібліотеку привабливою для алгоритмів машинного навчання [35].

Іншим рішенням від Apache є Singa [36], яка надає просту модель для тренування мереж глибокого навчання на ряді пристроїв. Вона підтримує багато поширених типи тренування: згорткові нейронні мережі, обмежені машини Больцмана та рекурентні нейронні мережі. Моделі можна тренувати синхронно або асинхронно, на кластерах CPU або GPU. Оскільки в роботі програмна

реалізація не використовує глибинне навчання, то цей фреймворк розглядається лише для ознайомлення.

Якщо говорити про фреймворки від світових лідерів IT-розробок, то одним із самих популярних рішень є Google TensorFlow, назва якого походить від операцій з багатовимірними масивами даних, які також називаються «тензорами». Ця бібліотека програмного забезпечення з відкритим кодом використовується для високопродуктивних чисельних розрахунків. Його гнучка архітектура дозволяє легко розгортати обчислення на різних кластерах (CPU, GPU, Google TPU (тензорний процесор)), від настільних комп'ютерів, серверних рішень та мобільних пристроїв [37]. Бібліотека зародилася серед дослідників та інженерів команди Google Brain, яка знаходиться в експериментальній організації AI Google. TensorFlow добре підходить для машинного та глибокого навчання, а гнучке ядро чисельного обчислення використовують в багатьох інших наукових областях. Самі обчислення виражаються у вигляді потоків даних через граф станів. Фреймворк чудово підходить для навчання генеративно-змагальних мереж, в яких дві штучні нейронні мережі змагаються одна з одною в рамках гри з нульовою сумою, а також використовується для автоматизованої анотації зображень [24] та систем збільшення релеватності ранжування пошукових видач [22].

Також існує рішення від іншого гіганта – Amazon. Його платформа для машинного навчання Amazon Machine Learning використовує потужні алгоритми машинного навчання спільно з інтерактивними візуальними інструментами, що дозволяє з легкістю створювати, оцінювати і розгортати моделі машинного навчання, такі як бінарна та мультикласова класифікація, а також регресії. Вбудовані алгоритми перетворення даних забезпечують ефективне перетворення вхідних наборів даних для забезпечення максимальної якості моделей прогнозування. Після того як модель створена, можна досліджувати її сильні і слабкі сторони і налаштувати продуктивність відповідно до бізнес-цілей. Для цього в сервісі передбачена інтуїтивно зрозуміла консоль оцінки і точної настройки моделей [38]. Нажаль, треновані моделі неможливо експортувати і вони використовуються лише в екосистемі Amazon.

Microsoft також не відстає від Google та Amazon, надаючи свої інструменти, такі як Azure ML Studio, Distributed Machine Learning Toolkit та Cognitive Toolkit. Перший дозволяє користувачам створювати та тренувати моделі, перетворювати їх в API і використовувати в інших сервісах. Одним із плюсів є те, що існує безкоштовна версія для навчання, яка включає в себе 10 гігабайт для зберігання навчальних даних [39]. Сервіс включає безліч навчальних алгоритмів як від Microsoft, так і від сторонніх компаній. Distributed Machine Learning Toolkit є окремою гнучкою системою, яка підтримує уніфікований інтерфейс для розпаралелювання даних, гібридну структуру даних для зберігання великих моделей, модельне планування для великих моделей навчання та автоматичне конвеєрне з'єднання для підвищення ефективності навчання.. Він створений як окремий фреймворк, а не як готове рішення, тому в нього за замовчуванням включена невелика кількість алгоритмів [40]. Cognitive Toolkit дозволяє користувачам створювати нейронні мережі способом орієнтованого графа, та може досягати високої швидкості роботи при використанні декількох CPU і GPU одночасно [41]. Так як і DMLT є окремим фреймворком.

Більшість з вище наведених фреймворків та бібліотек прив'язані до певної платформи, які використовуються клієнтами компаній для власних бізнес-процесів. Їх також можна використовувати у розробці автоматизованої рейтингової системи, але для написання автономної системи, яка буде працювати в рамках певної екосистеми краще використовувати самописні програмні рішення, написанні на популярних мовах програмування. Зазвичай, для цих цілей використовують такі мови, як C++, C#, Java, Ruby, Python, R та Lua. Також можна продемонструвати бібліотеки, з якими працюють на цих мовах.

Однією із таких бібліотек є mlpack2. Це бібліотека машинного навчання C + + з акцентом на масштабованість, швидкість та простоту у використанні. Його мета полягає в тому, щоб зробити машинне навчання простим для початківців, за допомогою простого та послідовного API, одночасного використання функцій мови C ++ і забезпечити максимальну продуктивність та гнучкість для досвідчених користувачів [42]. Це зроблено шляхом створення набору

командного рядка виконуваних файлів, які можна використовувати як чорні ящики, а також модульний API C ++ для досвідчених користувачів та дослідників, в якому легко внести зміни до внутрішніх частин алгоритмів.

Для мови програмування Java є свій фреймворк, під назвою Massive Online Analysis. Для неї це найпопулярніша платформа з відкритим вихідним кодом для видобування потоку даних, з дуже активною зростаючою спільнотою, вона включає в себе сукупність алгоритмів машинного навчання (класифікація, регресія, кластеризація, виявлення викидів, виявлення дрейфу концепції та рекомендаційних систем) та інструменти для оцінки [43].

Ще однім фреймворком, який був написанний на мові LuaJIT є Torch. Метою Torch є максимальна гнучкість та швидкість у створенні алгоритмів, роблячи цей процес надзвичайно простим. Він постачається з разом великою екосистемою пакетів, керованих спільнотою в області машинного навчання, комп'ютерного бачення, обробки сигналів, паралельної обробки, зображення, відео, аудіо та мереж, а також створюється на вершині спільноти Lua [44]. В основі Torch є бібліотеки нейронних мереж та оптимізації, які є простими у використанні, але мають максимальну гнучкість у застосуванні топологій складних нейронних мереж, на яких можна побудувати довільні графіки нейронних мереж і ефективно розпаралелювати їх над CPU та GPU.

Для платформи .NET Framework від Microsoft може використовуватись фреймворк Accord.NET. Це платформа навчання для .NET в поєднанні з аудіо та обробкою зображень бібліотек, повністю написаних на C#. Це повна база для побудови комп'ютерного зору на виробництві, комп'ютерного прослуховування, обробки сигналів та застосування статистики навіть для комерційного використання. Всеосяжний набір прикладних програма забезпечує швидкий початок швидкого запуску та роботи [45].

Для Python існує своя бібліотека, що швидко розвивається, під назвою scikit-learn. Вона включає в себе інструменти для багатьох стандартних задач машинного навчання (таких як кластеризація, класифікація, регресія і т. д.). І так як scikit-learn розробляється великим співтовариством розробників і експертів по

машинним навчання, перспективні нові методи, як правило, включаються в досить короткий термін. Бібліотека використовує багате середовище для забезпечення найсучасніших реалізацій багатьох добре відомих алгоритмів машинного навчання, одночасно підтримуючи простий у використанні інтерфейс, який тісно інтегрований з мовою Python. Це відповідає зростаючій потребі аналізу статистичних даних непрофесіоналами в галузях програмного забезпечення та Інтернету, а також у сферах за межами комп'ютерної науки, таких як біологія або фізика [23].

В процесі розробки програмного забезпечення для рейтингового оцінювання студентів буде використана мова Python, оскільки ця мова дуже проста в освоєні та має низький поріг для входу. Також в комплекті з Python йде широкий набір програмних пакетів, які можна використати для роботи з даними, деякі з них були використані при написані програмної реалізації для пункту 3.1:

1) NumPy. як програмна бібліотека мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами.
2) pandas, як ще одна бібліотека мови Python для обробки і аналізу даних. Робота pandas з даними будується поверх бібліотеки NumPy, що є інструментом нижчого рівня. pandas надає спеціальні структури даних і операції для маніпулювання числовими таблицями і тимчасовими рядами. Назва бібліотеки походить від економетричного терміна «панельні дані», використовуваного для опису багатовимірних структурованих наборів інформації.
3) Matplotlib для візуалізації даних двовимірної (2D) графікою. Бібліотека є гнучким, легко конфігурованим пакетом, який разом з NumPy надає можливості, подібні MATLAB.
4) PyQt, як інтеграція кросплатформного фреймворку Qt у середовище мови Python, яка являє собою набір інструментів для створення класичних і вбудованих графічних інтерфейсів користувача, а також програм, що працюють на різних програмних та апаратних платформах, які практично не

змінюють базову кодову базу, але як і раніше є нативною програмою з власними можливостями та швидкістю.

5) Scikit-learn, як основна бібліотека для побудування та використання моделей машинного навчання.

Також, при первинному аналізі завдання та первинних даних, було можливе використання такої мови, як R. Ця мова є спеціалізованою під статистичні дослідження та науки про дані, та вбудовані можливості для візуалізації даних. Але її вузька спеціалізованість є великим мінусом, оскільки при написанні програми код використовується не тільки для аналізу даних. Також мова R має дуже низьку продуктивність, у порівняні з іншими мовами програмування.

Оскільки сам програмний код можна написати лише користуючись стандартним текстовим редактором, то питання вибору інтегрованої середи розробки (IDE) є другорядним питанням для написання автоматизованого рішення. Для Python може бути використані такі рішення, як JetBrains PyCharm та Microsoft Visual Studio. Але для програмної реалізації в цій роботі був використаний редактор Visual Studio Code, який є більш легкою та безкоштовною версією MS Visual Studio, з широкою базою плагінів та зручним підсвічуванням синтаксису. Він зображенний на рисунку 3.1.

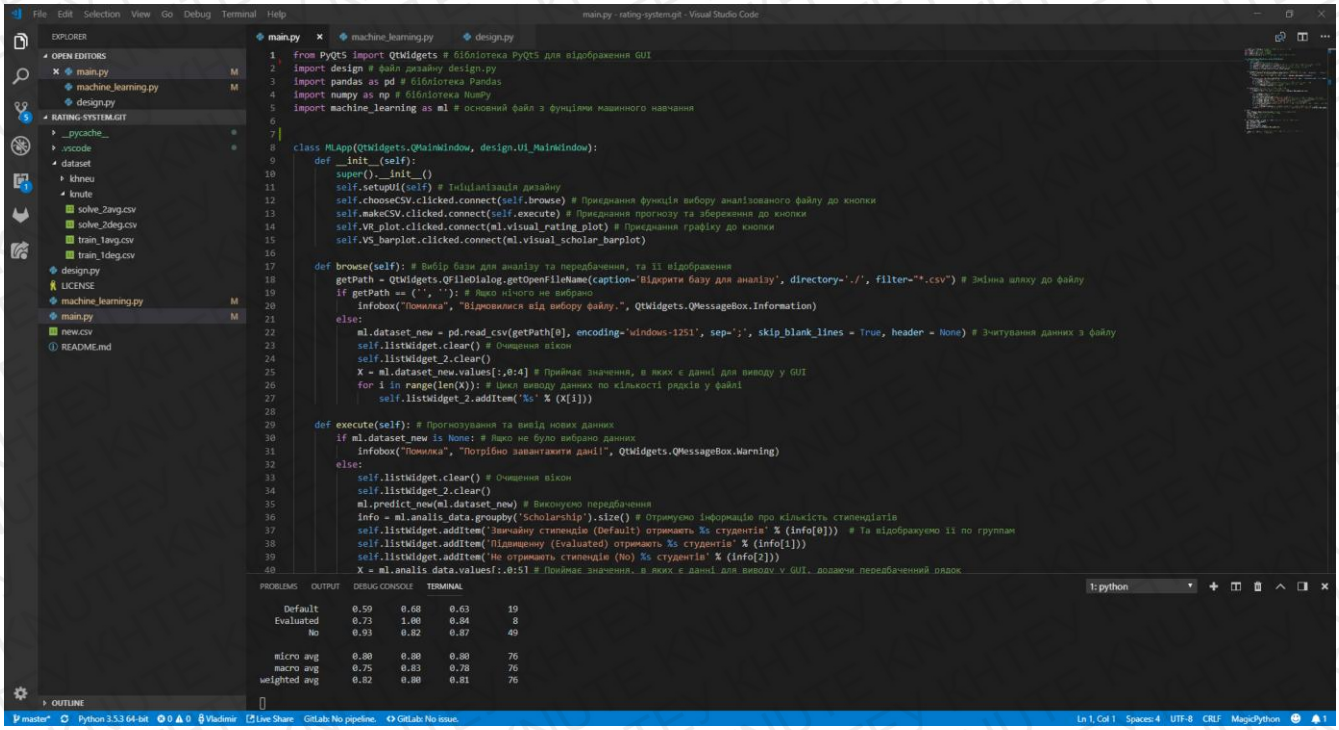

Рис 3.1 Відображення IDE MS Visual Code з підсвіченим синтаксисом

3.2. Формування первинного набору даних та технології їх збагачення і очищення

Набір даних, або датасет є однією з найважливіших частин у машинному навчанні, тому перед тим, як підбирати алгоритми і будувати моделі, потрібно чітко сформувати структуру, яка буде спочатку основою для первинного тренування, а вже потім для передбачення. В мережі інтернет, на сьогоднішній день, існує дуже широкий вибір із первинних наборів даних, які збираються вже не перший рік. Одним із проектів, який накопичує різні набори даних для подальшого використання у машинному навчанні має назву data.world [46]. Одне з ключових відмінностей data.world від інших «банків датасетів» - це інструменти, створення для спрощення роботи з даними. Система підтримує SQL-запити для вивчення даних і об’єднання кількох наборів даних, а також н власний набір із засобів розробки (англ. Software Development Kit, SDK), що спрощує роботу з даними і дозволяє працювати на популярних мовах не покидаючи сайту.

Більшість даних, які оцінюються та прогнозуються треба отримувати в процесі навчання студентів. Безперечно, статистично кожна оцінка та дія у навчальному процесі залежить від певного набору параметрів, але в рамках цієї роботи потрібно використовувати реальні данні студентів українських вищих навчальних закладів. Як приклад, сформований датасет буде користуватись протоколами розподілення стипендіатів КНТЕУ, зовнішній вигляд яких зображено на рисунку 3.2.

Реальні дані збираються для подальшої обробки з різних джерел і процесів. Вони можуть містити помилки і пошкодження, які негативно впливають на якість набору даних. Типові проблеми з якістю даних:

1) Неповнота: дані не містять атрибутів, або в них пропущені значення.
2) Шум: дані містять помилкові записи або викиди.
3) Неузгодженість: дані містять конфліктуючі між собою записи або розбіжності.

Київський національний торговельно-економічний університет

П Р О Т О К О Л № 34/-2018

стипендіальної комісії факультету обліку, аудиту та інформаційних систем

від "15" __01___ 2018__ р.

ПРИСУТНІ: Голова комісії Харченко О.А.

Члени комісії: 1. Гордополов В.Ю. 7. Гончаренко Ю.Ю.
2. Кутова О.А. 8. Щербань Ю.М.
3. Бродська О.Л. 9. Василега О.В.
4. Васильєва О.В. 10. Гаврилюк Я.М.
5. Куца К.К. 11. Молявін М.І.
6. Антоневич М.Є.

СЛУХАЛИ: Призначення стипендії за результатами підсумкового семестрового контролю за 1-й семестр 2017/18 н.р. студентам освітнього ступеня "магістр", 1 курсу, спеціальності "Облік і оподаткування" на період з 01.01.2018 по 30.06.2018.

ПОСТАНОВИЛИ:

1. Затвердити рейтинг успішності студентів за 1-й семестр 2017/18 н.р. освітнього ступеня "магістр", 1 курсу, спеціальності "Облік і оподаткування", денної форми навчання.

№ з/п	Прізвище, ім"я по батькові	Академічна група	ОС "Магістр" 1 курс, спеціальність "Облік і оподаткування"														Середній бал академічної успішності	Показник участі студента у науковій та науково-технічній діяльності, громадському житті та спортивній діяльності	Рейтинговий бал	Розмір академічної стипендії	Примітка
			Організація бухгалтерського обліку	Фінансовий аналіз	Облік і фінансова звітність за міжнародними стандартами	Управлінська звітність	Облік виплат та облік майна в матеріальній формі 1	Зовнішньоекономічна діяльність підприємства	Митне регулювання зовнішньоекономічної діяльності	Державне управління та регулювання економіки	Фінансовий корпоративний контроль	Інформаційні системи і технології в управлінні підприємством	Аудиторські послуги	Аналіз фінансових ризиків	Податкування соціально-економічних процесів						
1	Матейчук Юлія Костянтинівна	3м	92	95	98	90	99		90								94.0		**94.0**	1892.0	
2	Понкратова Дар'я Андріївна	6м		90	90					90		90		90	90		90.0	4.0	**94.0**	1892.0	
3	Слінчук Наталія Володимирівна	3м	93	93	98	90	99		90								93.8		**93.8**	1892.0	
4	Жадан Анастасія Олександрівна	3м	96	91	98	90	100		85								93.3		**93.3**	1300.0	
5	Мисник Анастасія Володимирівна	5м	90	92						95	93	90	90				91.7	0.5	**92.2**	1892.0	
6	Огданська Анастасія Валеріївна	6м		94	95					94		90		90	90		92.2		**92.2**	1892.0	
7	Штрайт Анастасія Олександрівна	2м	90	91	94	90	92	95									92.0		**92.0**	1892.0	
8	Миняйленко Анна Вікторівна	2м	95	90	95	90	96	85									91.8		**91.8**	1300.0	
9	Левицька Вероніка Віталіївна	4м	90	93	90					93	90	90					91.0		**91.0**	1892.0	

Рис 3.2 Зовнішній вигляд даних, які містяться в протоколі стипендіальної комісії

Оскільки програмні алгоритми не можуть працювати з наданим файлом у розширенні XLS, потрібні інформаційні поля витягуються з файлу, збагачуються, очищуються, та зберігаються у зручному форматі CSV. Тому формуємо перелік правил, які будуть використовуватись у програмній реалізації, у табл. 3.1. Програмні алгоритми видають помилку при роботі з українськими літерами, тому у рядку поля, по якому буде проводитись видача результату, будуть використані англомовні аналоги:

1) Підвищена стипендія = Evaluated;
2) Звичайна стипендія = Default;
3) Стипендія відсутня = No.

Таблиця 3.1

Форматування кінцевого набору даних

Назва стовпчика	Опис стовпчика	Метод очищення	Тип рядка
Номер групи	Належність студента до певної групи	Проведено очищення тексту і перетворено позначення групи магістрів з «nм» у «n.1», де n – номер групи	float (з рухомою точкою)
Додатковий бал	Показник участі студента у науковій та науково-технічній діяльності, громадському житті та спортивній діяльності	Без змін	float (з рухомою точкою)
Середній бал	Середнє значення усіх балів студента, отриманих протягом екзаменаційної сесії та виробничої практики	Перед цим планувалося використати перелік балів, але оскільки кожен студент має різну кількість балів, раціонально провести ущільнення.	float (з рухомою точкою)
Тип стипендії	Текстовий рядок, який містить категорію стипендії, яку отримає студент	Проведена дискретизація, оскільки розмір стипендії для кожного типу є сталою величиною	string (текстовий)

Якісні дані - це необхідна умова для створення якісного первинного набору даних для прогнозування. Щоб уникнути появи ситуації «сміття на вході, сміття на виході» і підвищити якість даних і, як наслідок, ефективність моделі, необхідно провести моніторинг працездатності даних, як можна раніше виявити проблеми і вирішити, які дії по попередній обробці і очищенню даних необхідні.

Під час виявлення неякісних даних використовуються такі методи:

1) Очищення даних - заповнення пропущених значень, виявлення і видалення перекручених даних та викидів.
2) Перетворення даних - нормалізація даних для зниження вимірювань і спотворень.
3) Ущільнення даних - створення вибірки даних або атрибутів для спрощення обробки даних.
4) Дискретизація даних - перетворення безперервних атрибутів в категоріальні, щоб простіше було використовувати деякі методи машинного навчання.
5) Очищення тексту: видалення впроваджених символів, які можуть порушувати вирівнювання даних, наприклад впроваджених символів табуляції в файлі з роздільником-табуляцією, впроваджених нових ліній, які можуть розбивати записи, тощо.

При роботі з пропущеними значеннями краще спочатку визначити причину їх появи в даних, що допоможе вирішити проблему. Ось які буває методи обробки пропущених значень:

1) Вилучення: видалення записів з пропущеними значеннями.
2) Фіктивна підстановка - заміна пропущених значень фіктивними, наприклад підстановка значення «unknown» (невідомо) замість категоріальних або значення 0, замість чисел.
3) Підстановка середнього значення: пропущені числові дані можна замінити середнім значенням.
4) Підстановка часто використовуваного елемента: пропущені категоріальні значення можна замінити найбільш часто використовуваних елементом.
5) Підстановка по регресії: використання регресійного методу для заміни пропущених значень регресійними.

	A	B	C	D
1	1	0.5	91.21429	Evaluated
2	1	0	86.28571	No
3	1	0	70.85714	No
4	2	0.5	90.35714	Default
5	3	1	89.42857	Default
6	4	0.5	89.35714	Default
7	4	0	89.14286	Default
8	5	0	88.85714	Default
9	4	0.5	88.78571	Default
10	2	0	88	Default
11	2	0	87.71429	Default
12	3	0	86.57143	Default
13	5	0	86	Default
14	3	0	85.71429	Default
15	4	0	85.28571	Default
16	2	0	84.85714	Default
17	4	0	84.85714	Default
18	2	0	84.71429	Default
19	2	0	84.71429	Default
20	2	0	83.85714	Default
21	4	0	83.57143	Default
22	2	0	83.42857	Default
23	3	0	82.57143	Default
24	5	0	82.28571	Default
25	3	0	81.42857	Default
26	4	0	81.28571	Default
27	3	0	81.14286	Default
28	3	0	81	Default
29	5	0	80.14286	Default
30	2	0	79.85714	Default

train_1avg

Рис 3.3 Сформований набір даних у редакторі MS Excel

Попередній перегляд даних можна зробити в Microsoft Excel, що зображенно на рисунку 3.3. Після чого сформовані дані можуть бути використані для тренування моделі і подальшого підтвердження (валідації) у рамках моделі рейтингового оцінювання студентів.

3.3. Програмна реалізація з розподілу стипендій

В процесі написання випускної кваліфікаційної роботи поставлена задача у розробці програмного продукту, який зможе розраховувати видачу стипендій методами машинного навчання. Актуальність, в порівнянні зі звичайною алгоритмізацією, полягає в тому, що методи машинного навчання не потребують прямої вказівки на розрахунок параметрів, що дозволяє працювати та аналізувати наявні дані, не вникаючи в процеси їх розрахунку.

Програма, як було зазначено у пункті 3.1, була написана на мові програмування Python та має користувацький інтерфейс, зображений на малюнку 3.4, для того, щоб з продуктом могли працювати звичайні користувачі персонального комп'ютера.

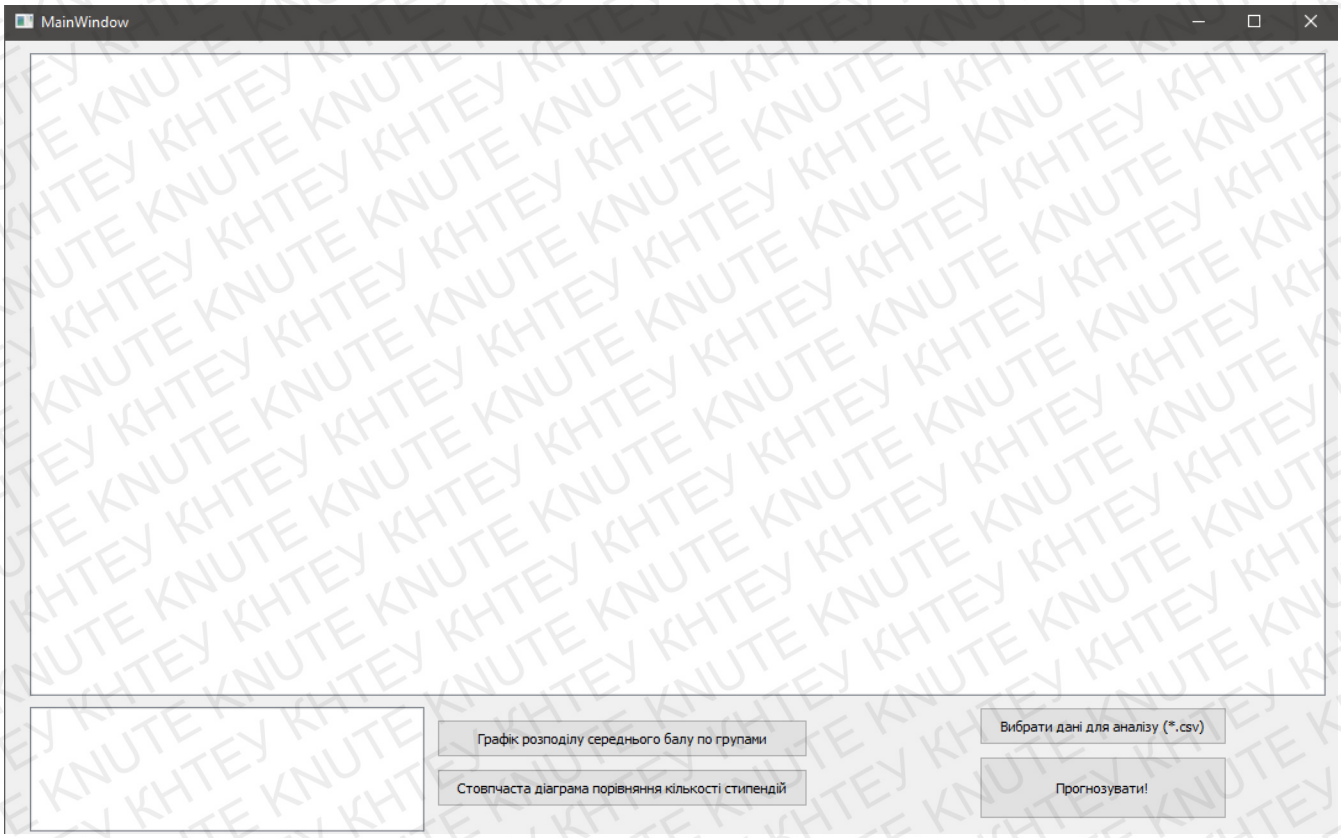

Рис 3.4 Головне вікно програми

Користувача зустрічає програмний інтерфейс, який виконаний в мінімалістичному стилі. При натисканні на будь-яку кнопку, окрім вибору даних для аналізу, з'явиться модальне вікно (рисунок 3.5), яке попередить користувача про те, що для роботи програми спочатку потрібно вибрати набір даних.

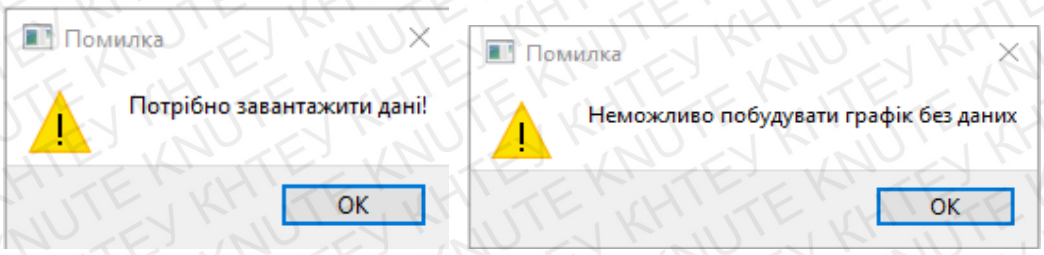

Рис 3.5 Вікна помилок при відсутності набору даних

Для початку роботи, користувачеві потрібно натиснути на кнопку вибору даних для аналізу, після чого відкривається стандартне вікно провідника операційної системи, відображене на рисунку 3.6, де потрібно вказати шлях до файлу з набором даних. Важливо те, що вибраний файл повинен бути правильно структурованим, та шлях до нього повинен містити лише англомовні символи.

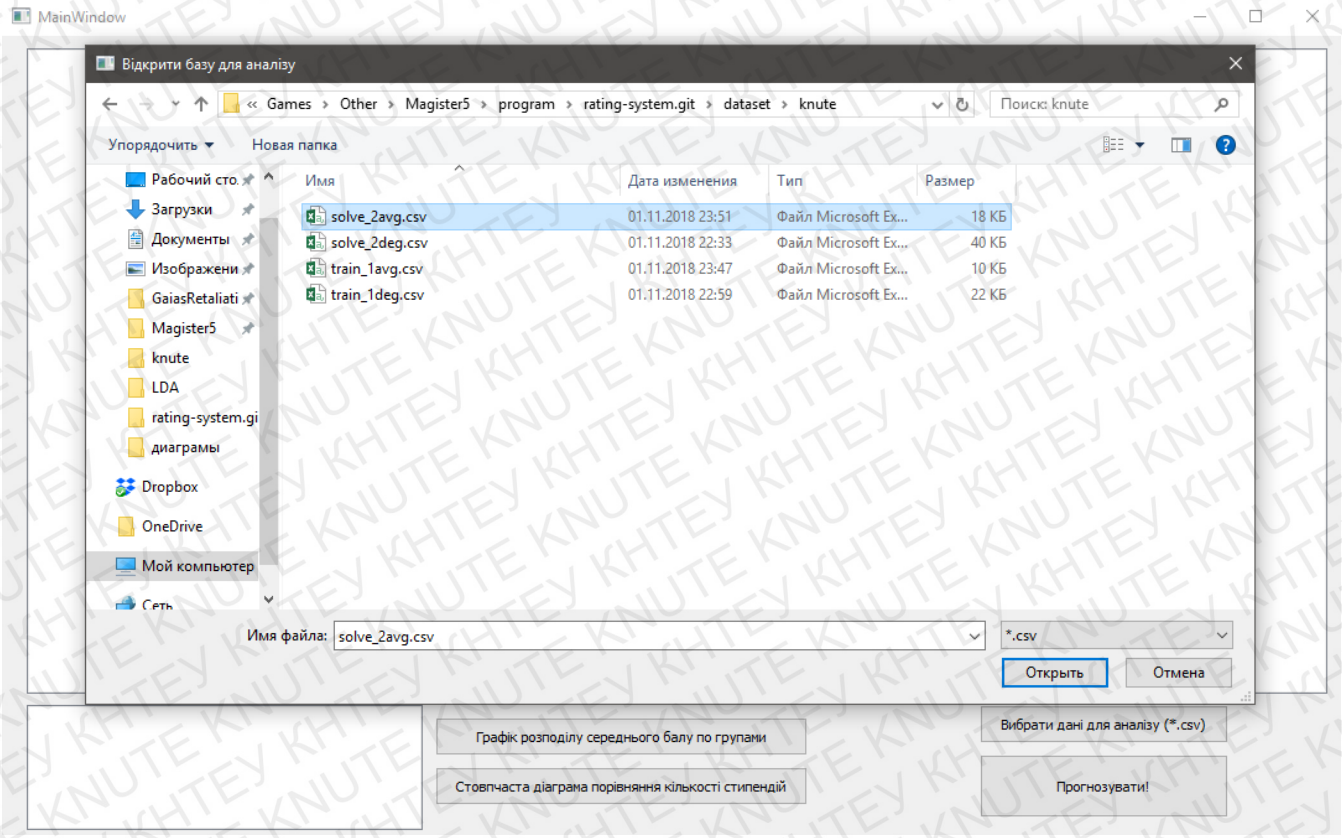

Рис 3.6 Вікно вибору файлу для аналізу

Користувач не може вибрати ніякий файл, окрім як у форматі CSV, тому при натисканні на кнопку «Відміна» висвітиться модальне вікно з інформацією, зображене на рисунку 3.7.

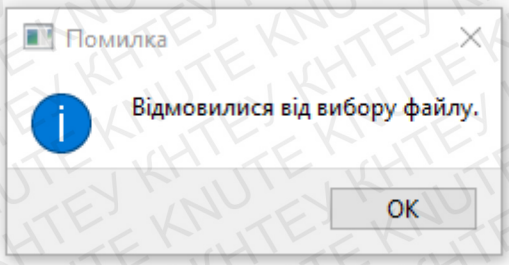

Рис 3.7 Вікно помилки при відміні вибору файлу

Після цього в головному вікні, на рисунку 3.8, відобразиться інформація, яка знаходиться в завантаженому наборі даних.

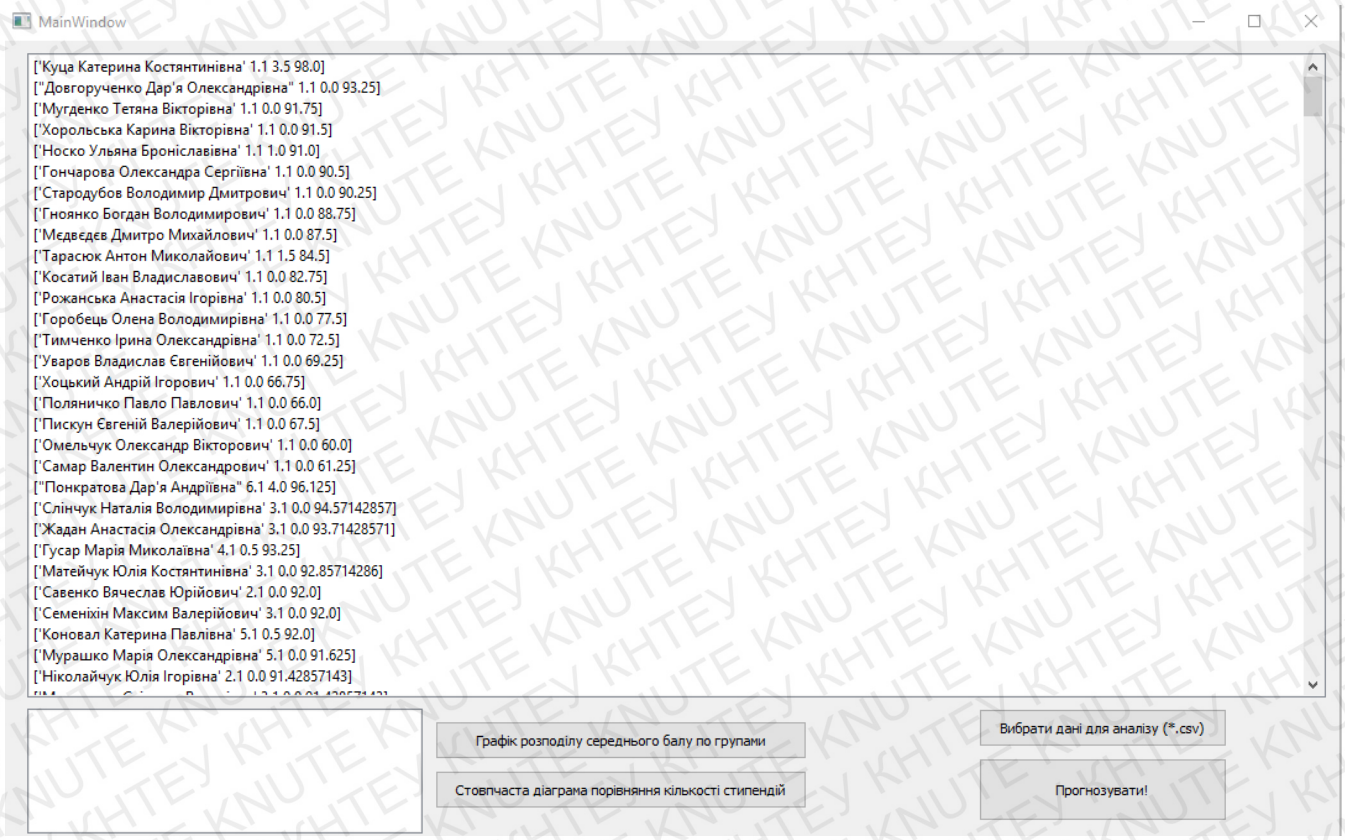

Рис 3.8 Головне вікно програми з завантаженими даними

Набір даних, що завантажується, повинен відповідати тим типам даних, на яких буда розроблена модель. Допускається наявність інформаційних стовпчиків, таких як ПІБ студента. Якщо вибрана база не відповідає натренованій моделі, то вона просто не буде працювати та завершить свою роботу.

Виконавши всі перевірки, користувач натискає кнопку «Прогнозувати!», після чого алгоритми машинного навчання починають створення нового

стовпчика, якій на основі тренованої моделі буде розподіляти стипендії по студентам. Фінальний результат зображено на рисунку 3.9.

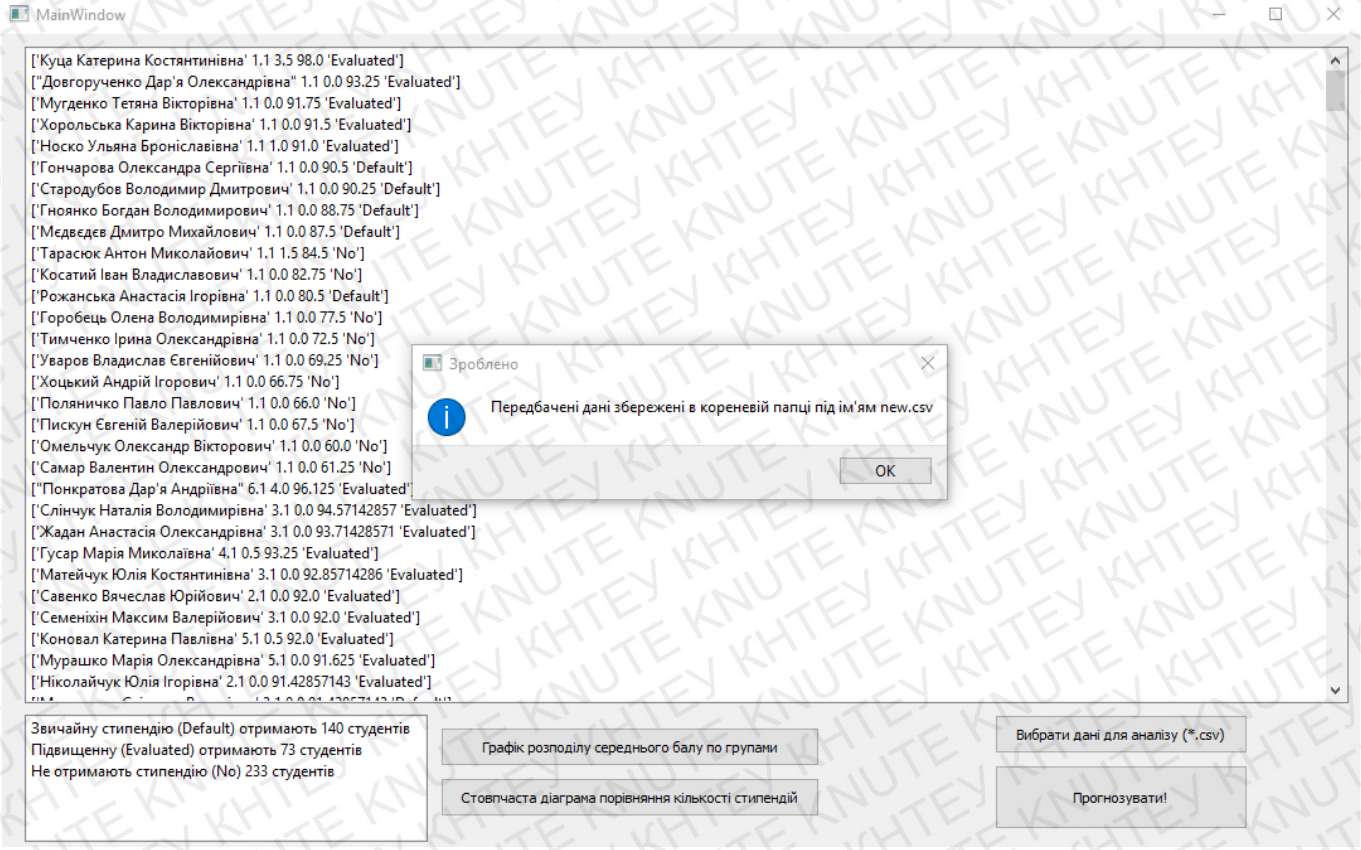

Рис 3.9 Вікно з прогнозованими значенням та вікном про успішне збереження прогнозу

Після закінчення роботи алгоритмів, програма видає повідомлення про успішне виконання прогнозу та збереження його в кореневій папці під ім’ям «new.csv». З даними можна ознайомитись у вікні програми, або працювати з сформованим файлом любим зручним редактором. Наприклад, у Microsoft Excel, як на рисунку 3.10.

	A	B	C	D	E
1	Куца Катерина Костянтинівна	1.1	3.5	98	Evaluated
2	Довгорученко Дар'я Олександрівна	1.1	0	93.25	Evaluated
3	Мугденко Тетяна Вікторівна	1.1	0	91.75	Evaluated
4	Хорольська Карина Вікторівна	1.1	0	91.5	Evaluated
5	Носко Ульяна Броніславівна	1.1	1	91	Evaluated
6	Гончарова Олександра Сергіївна	1.1	0	90.5	Evaluated
7	Стародубов Володимир Дмитрович	1.1	0	90.25	Default
8	Гноянко Богдан Володимирович	1.1	0	88.75	Default
9	Мєдвєдєв Дмитро Михайлович	1.1	0	87.5	Default
10	Тарасюк Антон Миколайович	1.1	1.5	84.5	No
11	Косатий Іван Владиславович	1.1	0	82.75	No
12	Рожанська Анастасія Ігорівна	1.1	0	80.5	Default
13	Горобець Олена Володимирівна	1.1	0	77.5	No
14	Тимченко Ірина Олександрівна	1.1	0	72.5	No
15	Уваров Владислав Євгенійович	1.1	0	69.25	No
16	Хоцький Андрій Ігорович	1.1	0	66.75	No
17	Поляничко Павло Павлович	1.1	0	66	No
18	Пискун Євгеній Валерійович	1.1	0	67.5	No
19	Омельчук Олександр Вікторович	1.1	0	60	No
20	Самар Валентин Олександрович	1.1	0	61.25	No
21	Понкратова Дар'я Андріївна	6.1	4	96.125	Evaluated
22	Слінчук Наталія Володимирівна	3.1	0	94.57143	Evaluated
23	Жадан Анастасія Олександрівна	3.1	0	93.71429	Evaluated
24	Гусар Марія Миколаївна	4.1	0.5	93.25	Evaluated
25	Матейчук Юлія Костянтинівна	3.1	0	92.85714	Evaluated
26	Савенко Вячеслав Юрійович	2.1	0	92	Evaluated
27	Семеніхін Максим Валерійович	3.1	0	92	Evaluated
28	Коновал Катерина Павлівна	5.1	0.5	92	Evaluated
29	Мурашко Марія Олександрівна	5.1	0	91.625	Evaluated
30	Ніколайчук Юлія Ігорівна	2.1	0	91.42857	Evaluated

new

Рис 3.10 Прогнозований набір даних, відкритий у MS Excel

Якщо було вирішено продовжити роботу з програмою, то вона може зробити швидку візуалізацію інформацію способами, такими як графік розподілу середнього балу по групам, точки якого демонструють бали в певній групі. По ньому можна проаналізувати кількість балів, які нижче певного порогу, та для вибраного набору даних результат зображено на рисунку 3.11.

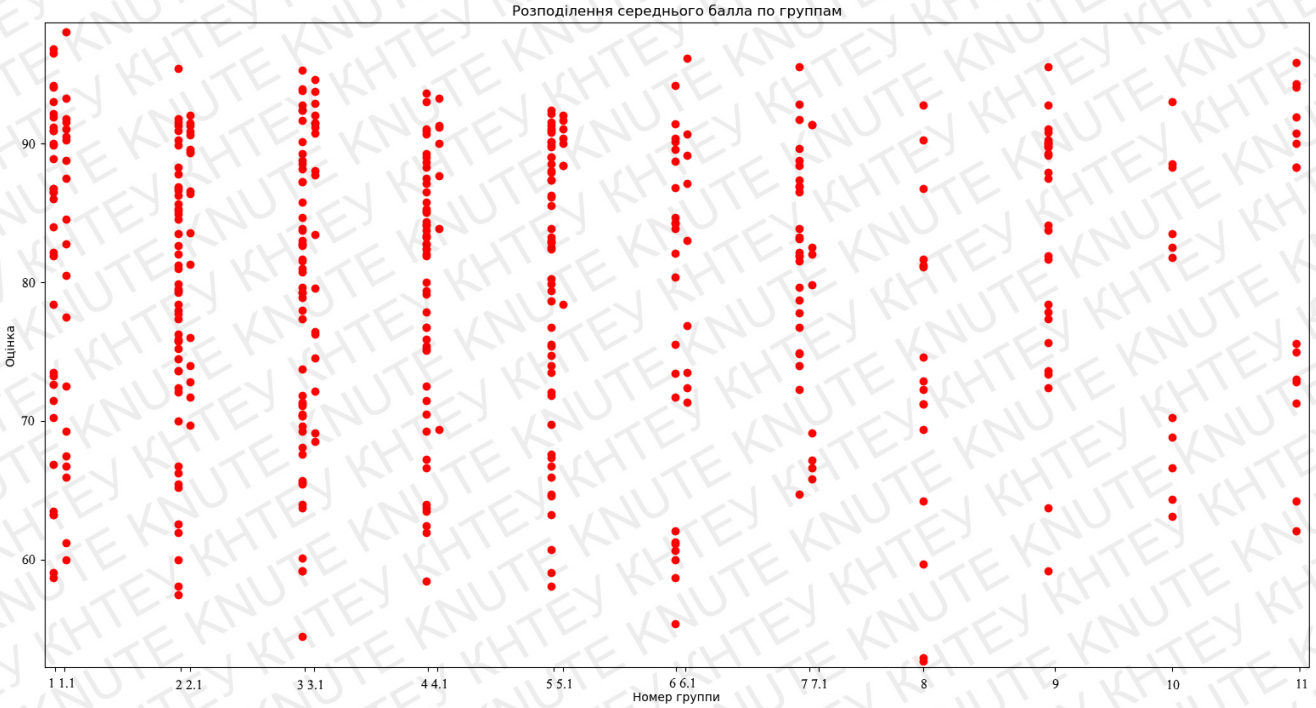

Рис 3.11 Графік розподілу середнього балу

А також побудувати стовпчасту діаграму з порівнянням кількості стипендій, по кількості студентів, як на рисунку 3.12.

Рис 3.12 Стовпчаста діаграма порівняння кількості стипендій

Сама програма працює з виконанням трьох файлів, які мають стандартне розширення мови Python:

1) main.py – головний виконавчий файл, який ініціалізує графічний інтерфейс користувача та запускає тренування моделі первинним набором даних. Код знаходиться у додатку В.1;
2) machine_learning.py – файл з функціями, які виконуються при роботі програми та містить базову логіку роботи програми, включаючи функції для підбору моделі. Код знаходиться у додатку В.2;
3) design.py – файл графічного інтерфейсу. Код знаходиться у додатку В.3.

Також в кореневій папці присутній файл new.csv, який при встановлені програми та першому запуску є пустим. У папці database заходяться набори даних, які використовувались у процесі написання роботи, в якій папка khneu містить дані для аналізу Харківського Національного Економічного Університету, а папка 2 – Київського Національного Торгово-Економічного Університету. Набори даних із останьої папки використовувались у демонстрації роботи програмної реалізації, а саме:

1) train_1avg.csv – первинний набір даних, який використовувався тренування та перевірки (валідації) моделі. За основу були взяті стипендіальні протоколи першого семестру за 2017-2018 роки;
2) solve_2avg.csv – набір даних, який аналізується в цьому розділі. Основою цього фалу були стипендіальні протоколи другого семестру за 2017-2018 роки.

Програмна реалізація захищена Загальною публічною ліцензією GNU, яка дає дозвіл на вільне копіювання, зміну та розповсюдження програмного коду, зі збереженням авторських прав на оригінал.

Висновки до розділу 3

У третьому розділі представлена практична частина випускної кваліфікаційної роботи, яка базується на поняттях та моделі другого розділу та розповідає про наявні можливості машинного навчання. Розглядаються комплексні системи та окремі фреймворки для роботи з машинним навчанням, які на сьогоднішній день полегшують роботу для початківців. Оглянуто як розробки крупних компаній, які мають власний інтерфейс прикладного програмування (API) для роботи з машинним навчанням, так і самостійні фреймворки, які можуть застосовуватись у інтегрованих середах розробки (IDE). Описано основний фреймворк та мова, на якій буде розроблена програмна реалізація та порівняння з ближчим аналогом. Також був проведено короткий огляд бібліотек Python, які використовувались у процесі розробки.

Було розглянуто важливість первинного набору даних, як і набору даних в цілому, а також описано основні методи збагачення та очищення даних. Після цього було продемонстровано роботу над первинними даними для програмної реалізації з використанням методів очищення та збагачення. Також ця робота була пророблена і для даних, які будуть прогнозуватись у третьому підрозділі.

Описано роботу програмної реалізації, яка виконує прогнозування стипендіатів без використання стандартних алгоритмів розрахунку. Продемонстровано програмний інтерфейс, запобіжні вікна з помилками та інформацію, пряму роботу моделі та візуалізацію. Реалізація є лише демонстраційною частиною, і має низку недоліків, які повинні дороблюватись з часом, а також описані в минулих розділах.

ВИСНОВКИ

В рамках випускної кваліфікаційної роботи було дано відповідь на завдання, які були поставлені метою дослідження, а саме:

1) визначенно сутність рейтингової системи, дано означення рейтингу, основні засади його існування та оцінювання студентів, а також системи ECTS, яка є основною шкалою у Болонському процесі;
2) проаналізовано наявний досвід України, в рамках рейтингових систем оцінювання Київського Національного Торгівельно-Економічного Університету та НТУУ «Київський політехнічний інститут імені Ігоря Сікорського», а також Харківського Національного Економічного Університету. Було порівнянно відчизняний досвід із американським, який використовується напротязі всього навчання студентів;
3) розкриті базові поняття машинного навчання, принцип його роботи та сучасні напрямки розвитку, які включають в себе штучні нейроні мережі, глибинне навчання, а також використання хмарних технологій та великих данних;
4) оглянуто можливі та найбільш популярні алгоритми машинного начання, які можна застосувати у рейтинговому оцінюванні студентів. На їх базі було проведенно дослідження із залученням даних КНТЕУ та ХНЕУ, які використовувались для підбору найбільш результативного алгоритма класифікації для роботи моделі рейтингового оцінювання та подальшого тренування;
5) розроблено програмне забезпечення на мові програмування Python, яке дозволяє проводити моніторинг наявних оцінок студентів та призначення стипендій. Також було оглянуто наявні програмні шаблони (фреймворки), та їх аналоги, які використовуються для інших мов програмування.

По отриманим результатам, які було оброблено в рамках дослідження, можна сказати, що метод машинного навчання є досить перспективним напрямком розвитку прогнозування, але на сьогоднішний день існує низка проблем, яка заважає впровадженню та єфективній роботі алгоритмів.

По-перше, це відсутність стандартизації систем рейтингового оцінювання, яка відрізняється у кожному вищому навчальному закладі. Якщо в американських навчальних закладах є певний вибір із декількох систем, но у відчизняних закладах воно залежить не тільки від закладу, але і від викладача. Для подальшого розвиту потрібно привести оцінювання студентів до певного стандартизованного вигляду.

По-друге, стоїть проблема відсутності потрібних даних для машинного навчання. Більшість з параметрів обліку студентів, на сьогоднішній день, ведеться у паперовому вигляді, що досить сильно затримує інтеграцію не тільки машинного навчання в процеси оцінювання студентів, а і комп'ютеризації вищих навчальних закладів в цілому.

По-третє, нестабільність законотворчих процесів у країні не дозволяє розпочати повноцінну інтеграцію методів машинного навчання у системи рейтингового оцінювання. На сьогоднішній день, як було описано при формуванні моделі на прикладі КНТЕУ, стипендію отримують лише 40-45% відсотків найкращих студентів у групі, а вже в наступному році відсоток зменшиться за рахунок підвищення грошових виплат.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України "Про вищу освіту" // Закон від 01.07.2014 № 1556-VII

2. Положення про систему рейтингового оцінювання діяльності студентів КНТЕУ, затверджене 25 травня 2017 р. [Електронний ресурс]. – Режим доступу: https://knteu.kiev.ua/file/NjY4NQ==/0600a257bc4ff1fcb774c2eb900eeb7c.pdf

3. Положення про систему рейтингового оцінювання результатів навчання студентів НУТТ "КПІ", затверджене 19 січня 2012 р. [Електронний ресурс]. – Режим доступу: http://kpi.ua/files/regulations-RSO.pdf

4. І. А. Лимар «Проведення прогнозно-аналітичних досліджень відповідності системи професійної освіти переспективам соціально-економічного розвитку України» [Текст] : монографія / [І. М. Грищенко, М. П. Денисенко, І. А. Ігнатьєва, В. В. Лойко, Т. Є. Воронкова, С. В. Бреус, Т. М. Власюк, Д. М. Лойко, О. Б. Моргулець, С. Г. Натрошвілі, Є. Б. Хаустова, О. В. Хоменко, З. Я. Шацька, Т. М. Янковець, І. А. Лимар, Д. А. Макатьора та ін.] ; за заг. ред. І. М. Грищенка. - К. : КНУТД, 2014. — С. 94-103

5. Ю.Л. Логвиненко «Сутність рейтингування підприємств та його значення в ринкових умовах» // Національний університет “Львівська політехніка” УДК 658.001.2:621, 2009

6. І.П. Тригуб «Мотивація студентів як один із основних факторів успішної професійної підготовки» // НУХТ, м.Київ УДК 37.013.46, 2014

7. Н.Л. Кивцова «Бально-рейтингова система в університетах США» // Вістник МДІКМ * 4 (6) липень-серпень, УДК 811.111:37.016, 2014

8. В. Бахрушин «Академічна доброчесність та об’єктивне оцінювання студентів» [Електронний ресурс]. – Режим доступу: http://saiup.org.ua/novyny/akademichna-dobrochesnist-ta-obyektyvne-otsinyuvannya-studentiv/

9. А. Панченко «Стипендия 2017: все по новому» [Електронний ресурс]. – Режим доступу: https://www.segodnya.ua/lifestyle/psychology/stipendiya-2017-vse-po-novomu--775241.html

10. С. Марченко «Мінфін хоче суттєво скоротити кількість отримувачів стипендій» [Електронний ресурс]. – Режим доступу: https://www.epravda.com.ua/news/2017/08/14/628032/

11. Л.В. Зубова, О.І. Ренер, Т.Д. Рожина, О.С. Степанова «Проблеми застосування бально-рейтингової системи у ВНЗ для контроля навчальних досягнень студентів» // УДК 378.146 ББК Ч448.02В, 2016

12. Pallabi Sarkar «What are the problems of rating system in performance appraisal?» [Електронний ресурс]. – Режим доступу: https://www.tuturself.com/posts/view?menuId=116&postId=933

13. Д.П. Ветров «Машине навчання – стан та перспективи» [Електронний ресурс]. – Режим доступу: http://ceur-ws.org/Vol-1108/paper1.pdf

14. D. Blei, A. Ng, M. Jordan «Latent Dirichlet Allocation» [Електронний ресурс]. – Режим доступу: http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

15. G. Hinton, S. Osindero, Y. Teh. «A fast learning algorithm for deep belief nets» [Електронний ресурс]. – Режим доступу: https://www.cs.toronto.edu/~hinton/absps/fastnc.pdf

16. C. E. Rasmussen «The Infinite Gaussian Mixture Model» [Електронний ресурс]. – Режим доступу: https://www.seas.harvard.edu/courses/cs281/papers/rasmussen-1999a.pdf

17. M. Kearns, Y. Nevmyvaka «Machine Learning for Market Microstructure and High Frequency Trading» [Електронний ресурс]. – Режим доступу: https://www.cis.upenn.edu/~mkearns/papers/KearnsNevmyvakaHFTRiskBooks.pdf

18. G. Brumfiel. "High-energy physics: Down the petabyte highway". Nature 469, 2011, pp. 282- 283

19. «Gartner Says AI Technologies Will Be in Almost Every New Software Product by 2020» [Електронний ресурс]. – Режим доступу: https://www.gartner.com/en/newsroom/press-releases/2017-07-18-gartner-says-ai-technologies-will-be-in-almost-every-new-software-product-by-2020

20. С. Янчишин "Чому машинне навчання - найважливіша технологія" [Електронний ресурс]. – Режим доступу: https://delo.ua/business/pochemu-mashinnoe-obuchenie-samaja-vazhnaja-tehnologija-343461/

21. Н.О. Баев «Використання методу опорних векторів в задачах класифікації» // Міжнародний журнал інформаційних технологій та енергоефективності. – 2017. – Т.2 №2(4) с. 17-21

22. V. Wollaston "Google releases TensorFlow: Search giant makes its artificial intelligence software available to the public" [Електронний ресурс]. – Режим доступу: https://www.dailymail.co.uk/sciencetech/article-3311650

23. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion "Scikit-learn: Machine Learning in Python" [Електронний ресурс]. – Режим доступу: http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

24!. M. Byrne "Google Offers Up Its Entire Machine Learning Library as Open-Source Software" [Електронний ресурс]. – Режим доступу: https://motherboard.vice.com/en_us/article/8q8avx/google-offers-up-its-entire-machine-learning-library-as-open-source

25. M. Mozina, J. Demsar, M. Kattan, B. Zupan «Nomograms for Visualization of Naive Bayesian Classifier» [Електронний ресурс]. – Режим доступу: https://pdfs.semanticscholar.org/a685/e42e7c2b7ed42d2a6f82cf518bc358f138ac.pdf

26. C. Bishop «Pattern Recognition and Machine Learning» // ISBN-10: 0-387-31073-8, ISBN-13: 978-0387-31073-2, Springer, 2006.

27. «Болоньский процес в Україні» [Електронний ресурс]. – Режим доступу: http://www.osvita.org.ua/bologna

28."Machine Learning: What it is & why it matters" [Електронний ресурс]. – Режим доступу: https://www.sas.com/ru_ua/insights/analytics/machine-learning.html

29. Ш. Акобир «Дерево рішень – загальні принципи роботи» [Електронний ресурс]. – Режим доступу: https://is.gd/jomdA0

30. Bayes' Theorem [Електронний ресурс]. – Режим доступу: http://mathworld.wolfram.com/BayesTheorem.html

31. Cheng Li, Bingyu Wang "Fisher Linear Discriminant Analysis", [Електронний ресурс]. – Режим доступу: https://pdfs.semanticscholar.org/1ab8/ea71fbef3b55b69e142897fadf43b3269463.pdf

32. «ALGLIB User Guide - Класифікація, регресія, кластеризація, робота з даними - Лінійний дискримінантний аналіз» [Електронний ресурс]. – Режим доступу: https://is.gd/qOaE2w

33. Scikit-learn.org «Nearest Neighbors» [Електронний ресурс]. – Режим доступу: http://scikit-learn.org/stable/modules/neighbors.html

34. Apache Spark FAQ [Електронний ресурс]. – Режим доступу: http://spark.apache.org/mllib

35. M. Zaharia "Spark: In-Memory Cluster Computing for Iterative and Interactive Applications" [Електронний ресурс]. – Режим доступу: https://www.youtube.com/watch?v=qLvLg-sqxKc

36. Apache SINGA - History [Електронний ресурс]. – Режим доступу: https://singa.incubator.apache.org/en/index.html

37. About TensorFlow [Електронний ресурс]. – Режим доступу: https://www.tensorflow.org

38. Amazon Machine Learning - What is machine learning? [Електронний ресурс]. – Режим доступу: https://aws.amazon.com/aml/features

39. Azure Machine Learning - What is machine learning? [Електронний ресурс]. – Режим доступу: https://azure.microsoft.com/en-us/overview/machine-learning/

40. Distributed Machine Learning Toolkit [Електронний ресурс]. – Режим доступу: http://www.dmtk.io

41. Unlock deeper learning with the new Microsoft Cognitive Toolkit [Електронний ресурс]. – Режим доступу: https://www.microsoft.com/en-us/cognitive-toolkit

42. what is mlpack? [Електронний ресурс]. – Режим доступу: http://mlpack.org/about.html

43. MOA Machine Learning for Streams [Електронний ресурс]. – Режим доступу: https://moa.cms.waikato.ac.nz

44. What is Torch? [Електронний ресурс]. – Режим доступу: http://torch.ch

45. Accord - Machine learning made in a minute [Електронний ресурс]. – Режим доступу: http://accord-framework.net

46. Why Dataquest? [Електронний ресурс]. – Режим доступу: https://www.dataquest.io/why-dataquest

Додаток А

РЕЙТИНГОВЕ ОЦІНЮВАННЯ ПОЗАНАВЧАЛЬНОЇ ДІЯЛЬНОСТІ СТУДЕНТА

Факультет, курс, група *Прізвище, ім'я, по батькові*		*Загальна сума балів*		
№ пор.	**Вид роботи**	**Кількість балів**	**Відмітка про досягнення**	**Відмітка яка зазначає вид підтвердження яке додається**
1	**НАУКОВО-ДОСЛІДНА РОБОТА**			
1.1	Автор патенту, винаходу тощо, зареєстрованого за встановленим законом порядком; співавтор законопроекту	20		
1.2	Публікація наукової статті у міжнародному виданні іноземною мовою; співавтор монографії	18		
1.3	Участь у науково-дослідній роботі (НДР) кафедри	15		
1.4	Переможець міжнародного конкурсу наукових робіт (диплом І-ІІІ ступеня)	15		
1.5	Переможець всеукраїнського конкурсу наукових робіт (диплом І-ІІІ ступеня)	14		
1.6	Переможець міжнародної конференції, олімпіади (диплом І-ІІІ ступеня)	13		
1.7	Переможець всеукраїнської конференції, олімпіади (диплом І-ІІІ ступеня)	12		
1.8	Участь у міжнародній/всеукраїнській олімпіаді, конкурсі наукових робіт, конференції, круглому столі	10		
1.9	Публікація наукової статті	10		
1.10	Публікація тез доповіді у міжнародному виданні іноземною мовою	7		
1.11	Переможець загальноуніверситетської олімпіади, конференції, круглого столу, Інтернет-конференції	7		
1.12	Публікація тез наукової доповіді, участь у Всеукраїнській інтернет-конференції	5		
1.13	Участь у загальноуніверситетській олімпіаді, конференції, круглому столі, Інтернет-конференції	4		
1.14	Переможець міжнародного чемпіонату з кулінарного мистецтва	9		
1.15	Переможець всеукраїнського чемпіонату з кулінарного мистецтва	7		
1.16	Участь у міжнародному чемпіонаті з кулінарного мистецтва	6		
1.17	Участь у всеукраїнському чемпіонаті з кулінарного мистецтва	4		
1.18	Участь в інтелектуальних іграх («Брейн-ринг», «Своя гра», «Що? Де? Коли?», «Інтелектуальна битва факультетів» тощо)			
	Перемога у змаганні	5		
Примітка	Призове місце у змаганні	3		
	Участь у змаганні	2		
2	**ГРОМАДСЬКА ДІЯЛЬНІСТЬ**			
2.1	Голова РСС університету	20		
2.2	Голова РСС факультету, гуртожитку	18		
2.3	Заступник голови РСС університету	15		
2.4	Голова сектору РСС університету, секретар РСС університету	14		
2.5	Заступник голови РСС факультету, гуртожитку	14		
2.6	Голова сектору РСС факультету, секретар РСС факультету	10		
2.7	Заступник голови сектору РСС факультету	8		
2.8	Член РСС факультету, куратор академічної групи	5		
2.9	Голова сектору РСС гуртожитку	8		
2.10	Заступник голови сектору РСС гуртожитку	5		
2.11	Староста блоку	5		
2.12	Заступник старости блоку	2		
2.13	Член РСС гуртожитку	2		
2.14	Староста академічної групи (у т.ч. староста курсу + 2 бали)	8		
2.15	Заступник старости академічної групи	3		

Продовження додатку А

2.1	Голова НТСАД та МВ	15		
2.2	Заступник голови НТСАД та МВ	10		
2.3	Голова відділу НТСАД та МВ	8		
2.4	Член НТСАД та МВ	5		
2.5	Заступник голови профкому Профспілки працівників і студентів університету	14		
2.6	Профорг факультету	10		
2.7	Профорг курсу	6		
2.8	Профорг академічної групи	4		
2.9	Голова ДНД КНТЕУ	14		
2.10	Інші студентські громадські організації, клуби, рухи університету			
Примітка	Голова організації	8		
	Члени організації	3		
3	**КУЛЬТУРНО-МАСОВА, СОЦІАЛЬНА ТА СПОРТИВНА РОБОТА**			
3.1	**Культурно-масова робота**			
3.1.1	Учасник Народного студентського камерного академічного хору	10		
3.1.2	Учасник студії сучасного танцю КМЦ КНТЕУ	8		
3.1.3	Студія вокалу КМЦ КНТЕУ	6		
3.1.5	Інструментальний ансамбль КМЦ КНТЕУ	6		
3.1.6	CreativeMediaGroup (журнал «Кіото,19», КНТЕУ-TV)			
Примітка	Основний склад	8		
	Участь у підготовці випуску:			
	1-3 рази	3		
3.1.7	Учасник та організатор культурно-масових заходів на рівні університету (Дебют першокурсника, Міс та Містер КНТЕУ, День університету, День факультету, Ліга КВН КНТЕУ за кубок ректора, Міс та Містер факультету)			
	1 захід	5		
3.1.8	Інші заходи в університеті або заходи які проводяться на рівні факультету та за його межами як представник від університету			
	1 захід	2		
3.2	**Соціальна робота**			
3.2.1	Донор крові	7		
3.2.2	Волонтерська діяльність (участь у 3-х і більше заходах)	5		
3.2.3	Разова участь у соціальних заходах	2		
3.3	**Спортивна робота**			
3.3.1	Член збірної команди університету, яка бере участь у чемпіонатах України, міжнародних змаганнях	10		
3.3.2	Призер міжнародних змагань	8		
3.3.3	Участь у міжнародних змаганнях	6		
3.3.4	Призер всеукраїнських змагань	5		
3.3.5	Участь у всеукраїнських змаганнях	4		
3.3.6	Член збірної команди університету	5		
3.3.7	Член збірної команди факультету на Спартакіаді серед студентів			
Примітка	Перемога у змаганні	3		
	Призове місце у змаганні	2		
	Участь у змаганні	2		

«___»_____20_р. _____ _____

(підпис студент) *(підпис старости групи)*

Додаток Б

Таблиця порівняння зросту, середнього балу та відвідування (демонстраційна)

Зріст, см	Середній бал	Присутність на минулій парі
185	72	1
166	41	1
180	44	1
168	62	1
175	38	1
179	55	1
185	75	0
193	88	0
168	85	1
172	63	1
176	64	0
189	89	0
175	51	1
168	75	1
195	68	0
192	52	0
186	80	1
186	48	0
190	72	0
171	41	0
192	73	1
192	75	1
190	52	1
169	82	0
177	67	1
168	65	0
169	48	0
185	42	1
181	39	0
194	46	1

Додаток В

Сирцевий код програмної реалізації

1) main.py

```
from PyQt5 import QtWidgets # бібліотека PyQt5 для відображення GUI
import design # файл дизайну design.py
import pandas as pd # бібліотека Pandas
import numpy as np # бібліотека NumPy
import machine_learning as ml # основний файл з фунціями машинного навчання


class MLApp(QtWidgets.QMainWindow, design.Ui_MainWindow):
    def __init__(self):
        super().__init__()
        self.setupUi(self) # Ініціалізація дизайну
        self.chooseCSV.clicked.connect(self.browse) # Приєднання функція вибору аналізованого файлу до кнопки
        self.makeCSV.clicked.connect(self.execute) # Приєднання прогнозу та збереження до кнопки
        self.VR_plot.clicked.connect(ml.visual_rating_plot) # Приєднання графіку до кнопки
        self.VS_barplot.clicked.connect(ml.visual_scholar_barplot)

    def browse(self): # Вибір бази для аналізу та передбачення, та її відображення
        getPath = QtWidgets.QFileDialog.getOpenFileName(caption='Відкрити базу для аналізу', directory='./', filter="*.csv") # Змінна шляху до файлу
        if getPath == ('', ''): # Ящко нічого не вибрано
            infobox("Помилка", "Відмовилися від вибору файлу.", QtWidgets.QMessageBox.Information)
        else:
            ml.dataset_new = pd.read_csv(getPath[0], encoding='windows-1251', sep=';', skip_blank_lines = True, header = None) # Зчитування данних з файлу
            self.listWidget.clear() # Очищення вікон
            self.listWidget_2.clear()
            X = ml.dataset_new.values[:,0:4] # Приймає значення, в яких є данні для виводу у GUI
            for i in range(len(X)): # Цикл виводу данних по кількості рядків у файлі
                self.listWidget_2.addItem('%s' % (X[i]))

    def execute(self): # Прогнозування та вивід нових данних
        if ml.dataset_new is None: # Ящко не було вибрано данних
            infobox("Помилка", "Потрібно завантажити дані!", QtWidgets.QMessageBox.Warning)
        else:
            self.listWidget.clear() # Очищення вікон
            self.listWidget_2.clear()
            ml.predict_new(ml.dataset_new) # Виконуємо передбачення
            info = ml.analis_data.groupby('Scholarship').size() # Отримуємо інформацію про кількість стипендіатів
            self.listWidget.addItem('Звичайну стипендію (Default) отримають %s студентів' % (info[0]))  # Та відображуємо її по группам
            self.listWidget.addItem('Підвищенну (Evaluated) отримають %s студентів' % (info[1]))
            self.listWidget.addItem('Не отримають стипендію (No) %s студентів' % (info[2]))
            X = ml.analis_data.values[:,0:5] # Приймає значення, в яких є данні для виводу у GUI, додаючи передбаченний рядок
```

```
        for i in range(len(X)):
            self.listWidget_2.addItem('%s' % (X[i])) # Цикл виводу данних по кількості рядків у
файлі
        infobox("Зроблено", "Передбачені дані збережені в кореневій папці під ім'ям new.csv",
QtWidgets.QMessageBox.Information)

def main():
    app = QtWidgets.QApplication(sys.argv)  # Новий екземпляр QApplication
    window = MLApp()  # Створюємо об'єкт класу MLApp
    window.show()  # Показуємо вікно
    ml.train() # Тренуємо первісну модель
    sys.exit(app.exec_())  # Запуск додатку

def infobox(err_header, err_msg, err_type): # Контейнер для системних оповіщень
    msg = QtWidgets.QMessageBox()
    msg.setIcon(err_type)
    msg.setText(err_msg)
    msg.setWindowTitle(err_header)
    msg.setStandardButtons(QtWidgets.QMessageBox.Ok)
    msg.exec_()

if __name__ == '__main__':  # Якщо ми запускаємо файл безпосередньо, а не імпортуємо
    main()  # то запускаємо функцію main()
```

2) machine_learning.py

```
import numpy as np
import pandas as pd
import csv
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
from PyQt5 import QtWidgets

#Імпорт важливих бібліотек SKLearn
from sklearn import model_selection
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

path = "./dataset/2/train_1avg.csv" # Первісна база для навчання
dataset = pd.read_csv(path, encoding='windows-1251', sep=';', skip_blank_lines = True, header =
None)

a_path = "./new.csv" # Шлях до бази, яка буде зберігатися
names = ['Name', 'Group','Average', 'Activity', 'Scholarship']
analis_data = pd.read_csv(a_path, names=names, encoding='windows-1251', sep=';',
```

```
skip_blank_lines=True, header=None)

dataset_new = None # Для перевірки вибору
model = KNeighborsClassifier() # Модель передбачена програмістом и корегується вручну

def train(): # Вчимо модель
	X = dataset.values[:,0:3] # Массив ознак
	Y = dataset.values[:,3] # Масив правильних відповідей
	validation_size = 0.15 # 85% на навчання, 15% на валідацію
	X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y,
test_size=validation_size) # Навчанні данні та дані для валідації
	model.fit(X_train, Y_train) # Навчальні данні формують модель

	# Функції для програміста, які допомогают вибрати модель
	print("Вибір моделі для данних КНТЕУ")
	#model_pick(X_train, Y_train) # Вибір моделей, функція для розділу 2.3
	evaluate(model, X_validation, Y_validation) # Перевірка точності

def evaluate(model, X_validation, Y_validation):
	predictions = model.predict(X_validation)
	print(accuracy_score(Y_validation, predictions))
	print(confusion_matrix(Y_validation, predictions))
	print(classification_report(Y_validation, predictions))

def model_pick(X_train, Y_train):
	models = []
	models.append(('LR', LogisticRegression())) # Логістична регрессія
	models.append(('LDA', LinearDiscriminantAnalysis())) # Лінійний дискримінантний аналіз
	models.append(('KNN', KNeighborsClassifier())) # Метод k-найближчих сусідів
	models.append(('CART', DecisionTreeClassifier())) # Дерево рішень
	models.append(('NB', GaussianNB())) # Наївний баєсовский классифікатор
	models.append(('SVM', SVC())) # Метод опорних векторів

	results = []
	names = []
	for name, model in models:
		kfold = model_selection.KFold(n_splits=10) # 10-кратна перехресна перевірка
		# Ділимо вибірку на 10 частин, тренуємо на 9, та тестуємо на 1. Повторюється для
кожної комбінації тренування-тестування
		cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold)
		results.append(cv_results)
		names.append(name)
		msg = "%s: %f (%f)" % (name, cv_results.mean()*100, cv_results.std()) # Для кожної
моделі цикл виводить відсоток точності
		print(msg)
	fig = plt.figure() # Виводить порівняльний графік алгоритмів
	fig.suptitle('Графік ефективності роботи моделей')
	ax = fig.add_subplot(111)
	plt.boxplot(results)
```

```
        ax.set_xticklabels(names)
        plt.show()

def predict_new(table): # Передбачуємо нові дані
        X_new = table.values[:,1:4] # Массив нових ознак
        Y_new = model.predict(X_new) # На їх основі і моделі генеруємо наше передбачення
        save(table, Y_new) # Зберігаємо

def save(table, Y_new): # Функція збереження в *.csv
        arr_stack = np.column_stack((table.values[:,0:4],Y_new)) # Об'єднуємо все в один массив
        pd.DataFrame(arr_stack).to_csv("./new.csv", encoding='windows-1251', sep=';', header=None,
index=None) # Збереження


def visual_scholar_barplot(): #Функція візуалізації, порівняння кількості стипендіатів
        fig, ax = plt.subplots()
        type_of = ('Звичайна', 'Підвищенна', 'Без стипендії')
        y_pos = np.arange(len(type_of))
        count = analis_data.groupby('Scholarship').size()

        if dataset_new is None:
                error("Неможливо побудувати графік без даних",
QtWidgets.QMessageBox.Warning)
        else:
                ax.barh(y_pos, count, align='center', color='green', ecolor='black')
                ax.set_yticks(y_pos)
                ax.set_yticklabels(type_of)
                ax.set_xlabel('Кількість стипендіатів')
                ax.set_title('Розподілення по стипендіям')
                plt.show()

def visual_rating_plot(): # Функція візуалізації, графік розподілення середнього балу по групам
        if dataset_new is None:
                error("Неможливо побудувати графік без даних",
QtWidgets.QMessageBox.Warning)
        else:
                plt.plot(analis_data.values[:,1], analis_data.values[:,3], 'ro')
                plt.xlabel('Номер группи')
                plt.ylabel('Оцінка')
                plt.title('Розподілення середнього балла по группам')
                plt.show()

def error(err_msg, err_type): # Контейнер для оповіщень з помилками
  msg = QtWidgets.QMessageBox()
  msg.setIcon(err_type)
  msg.setText(err_msg)
  msg.setWindowTitle("Помилка")
  msg.setStandardButtons(QtWidgets.QMessageBox.Ok)
  msg.exec_()
```

3) design.py

```
# -*- coding: utf-8 -*-
# Form implementation generated from reading ui file './design.ui'
# Created by: PyQt5 UI code generator 5.11.3
# WARNING! All changes made in this file will be lost!

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_MainWindow(object):
  def setupUi(self, MainWindow): # Ініціалізація головного вікна
    MainWindow.setObjectName("MainWindow")
    MainWindow.resize(1082, 645)
    self.centralwidget = QtWidgets.QWidget(MainWindow)
    self.centralwidget.setObjectName("centralwidget") # Центральне вікно
    self.makeCSV = QtWidgets.QPushButton(self.centralwidget)
    self.makeCSV.setGeometry(QtCore.QRect(790, 580, 201, 51))
    self.makeCSV.setObjectName("makeCSV") # Кнопка для прогнозування та збереження
даних
    self.chooseCSV = QtWidgets.QPushButton(self.centralwidget)
    self.chooseCSV.setGeometry(QtCore.QRect(790, 540, 201, 31))
    self.chooseCSV.setObjectName("chooseCSV") # Кнопка вибору файлу
    self.VR_plot = QtWidgets.QPushButton(self.centralwidget)
    self.VR_plot.setGeometry(QtCore.QRect(350, 550, 301, 31))
    self.VR_plot.setObjectName("VR_plot") # Кнопка побудови графіку
    self.VS_barplot = QtWidgets.QPushButton(self.centralwidget)
    self.VS_barplot.setGeometry(QtCore.QRect(350, 590, 301, 31))
    self.VS_barplot.setObjectName("VS_barplot") # Кнопка побудови діаграми
    self.listWidget = QtWidgets.QListWidget(self.centralwidget)
    self.listWidget.setGeometry(QtCore.QRect(20, 540, 320, 101))
    self.listWidget.setObjectName("listWidget") # Для основного поля з інформацією
    self.listWidget_2 = QtWidgets.QListWidget(self.centralwidget)
    self.listWidget_2.setGeometry(QtCore.QRect(20, 10, 1051, 521))
    self.listWidget_2.setObjectName("listWidget_2") # Для додаткової інформації
    MainWindow.setCentralWidget(self.centralwidget)

    self.retranslateUi(MainWindow)
    QtCore.QMetaObject.connectSlotsByName(MainWindow)

  def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
    self.makeCSV.setText(_translate("MainWindow", "Прогнозувати!"))
    self.chooseCSV.setText(_translate("MainWindow", "Вибрати дані для аналізу (*.csv)"))
    self.VR_plot.setText(_translate("MainWindow", "Графік розподілу середнього балу по
групами"))
    self.VS_barplot.setText(_translate("MainWindow", "Стовпчаста діаграма порівняння кількості
стипендій"))
```