DOI: 10.25140/2411-5363-2021-3(25)-202-212
УДК 004.65

Влада Ліпська
здобувачка вищої освіти
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського» (Київ, Україна)
E-mail: vladkalipskaya@gmail.com. **ORCID:** https://orcid.org/0000-0002-9847-7637

СПОСІБ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ДЕПЕРСОНІФІКАЦІЇ БАЗ ДАНИХ

Тема є актуальною через необхідність захисту персональних даних під час зберігання чи використання у різних системах, тому попит на анонімізацію даних закономірно з кожним днем зростає. Розглянуто відомі способи статичних замін, побудов реляційних даних та залежних від заданих математично кривих даних, і запропоновано метод покращення результатів – поєднання відомих способів замін із штучним синтезом даних на основі їх природи, враховуючи математичні показники. Метод було перевірено експериментально та висвітлено результати такого застосування з аналізом.

Ключові слова: *деперсоніфікація, анонімізація, дані.*
Рис.: 12. Табл.: 2. Бібл.: 8.

Актуальність теми дослідження. На сьогодні майже в усіх системах відбувається взаємодія з персональними даними користувача. Таких прикладів є безліч, щонайменше деякі з них можна зустріти щодня - режим “не турбувати” на телефоні, дані здоров’я на трекерах під час занять спортом, заміри для додатків здорового харчування, геолокації на картах, збереження історій поїздок, відмітки улюблених місць, збереження вкладень у різних браузерах, інформація банківських карток чи обраний набір товарів. То ж, відповідно відбувається стрімке розповсюдження та зберігання персональних даних, які потім часто використовуються для цілей бізнесу (застосування таргетованої реклами для специфічної групи осіб, надсилання повідомлень зі знижкою на певний товар користувачам зі схожими вподобаннями, рекомендації при виборі послуги чи товару...). Існує декілька можливих напрямків використання даних користувача. Один з них - це передбачення поведінки конкретних осіб на основі аналізу їх попередніх дій, а інший - передбачення поведінки нової групи схожих осіб. Схожість однієї множини на іншу визначається за допомогою характеристик («features»), які були обрані для аналізу. Яскравою ілюстрацією є платформа Netflix. Рекомендації системи для користувача застосовуються майже всюди [1]. Працівники компанії емпіричним шляхом з’ясували, що чим більше персональних показників застосовуються, тим більш точні результати отримуються [1]. То ж, у проектах компанії навіть почали використовувати дані про друзів з Facebook, щоб зрозуміти соціальний стан людини та з урахуванням кола спілкування пропонувати відповідний матеріал. Після декількох експериментів було з’ясовано, що рекомендації на основі персональних даних працюють краще в близько 5 разів, ніж на основі загальних рейтингів. Проте цікавим спостереженням і ефективним рішенням виявилось поєднання персональних даних та загальних рейтингів. Тобто спершу обиралась вибірка найкращих фільмів певної кількості, а потім рейтинг не мав значення і відбувався аналіз на основі персональних характеристик користувачів, яким рекомендуються фільми. Працівники компанії пов'язують це з соціальними звичками людей - здебільшого люди тяжіють до перегляду популярних фільмів через більшу вірогідність спільних вподобань з друзями, знайомими [1]. Отже, з точки зору бізнесу було ефективним поєднати метрику популярності з персональною метрикою потенційного задоволення від перегляду контенту. Таким чином, описаний випадок є підтвердженням, що інформація про користувача може бути використана у не зовсім очевидних галузях. Розглядаючи, таке застосування, можемо говорити про машинне навчання, а при тренуванні моделі не потрібна ідентифікація певної людини, достатньо знати, що люди зі схожими параметрами, ймовірно, будуть поводитись очікуваним чином з певним відсотком ймовірності (або лише «будуть поводитись так» та

«не будуть поводитись так» - значення у випадку задачі класифікації). Під задачею класифікації прийнято вважати необхідність віднесення до певного класу той чи інший набір даних, поведінку якого потрібно передбачити.

Безумовно, велика кількість конкретних даних надає певну свободу бізнесу, проте зі свободою приходить і відповідальність. Через недбале користування можна завдати шкоди користувачам, що зумовлює відповідальність на рівні законодавства. Тоді з'являється поняття деперсоніфікації - процес вилучення характеристик даних, за якими можна визначити конкретну особу (номер картки платника податків, прізвище...). Розглянемо приклад задачі з описаною потребою: припустимо, що є дві системи. Одна система містить дані своїх користувачів за декілька років, а інша система наразі лише в розробці (тобто немає користувачів платформи ще), але дуже схожа на першу і має доступ до даних першої системи. При додаванні певних функцій до нової системи може бути важливим розуміння поведінки потенційних користувачів. Таким чином, дані потенційних користувачів можуть стати у нагоді для аналізу та розробки нової системи, проте саме персональні дані не надто важливі, бо їх важко застосувати в новій системі (такого користувача просто немає ще в ній). Така ситуація трапляється при переході зі старої версії послуги на нову та при погодженнях корпорацій про обмін досвідом чи даними між проектами, розробками.

Також у галузі машинного навчання інколи можна натрапити на проблему – недотренованості моделі (underfitting [2]) - що свідчить про те, що для тренування моделей і отримання якісних результатів недостатньо даних. Тоді вже на допомогу приходить один із методів - синтез штучних даних, який надає можливість отримання більшої вибірки і відповідно більш точних результатів.

Звісно, проблема недотренованості не завжди розв'язується додатковими даними, інколи варто побудувати графіки даних, подивитись чи немає аномалій чи певних виключних значень, а можливо є дисбаланс між значеннями самих ознак (features). Також не останню роль відіграє зберігання значень у єдиному масштабі. Проте якщо, виконавши усі кроки, проблема залишається, то необхідні додаткові пласти даних.

Отже, питання використання даних зараз розглядається у багатьох галузях та безлічі системах, для частини з них важливо деперсоніфікувати дані шляхом вилучення певних значень характеристик, а для іншої частини цього може бути недостатньо для результатів (точність не відповідає очікуваній), тоді пропонується застосування синтезу даних на основі існуючих.

Постановка проблеми. У зв'язку з важливістю охорони персональних даних та одночасним попитом на рекомендаційні системи чи системи передбачення поведінки, гостро стоїть питання деперсоніфікації та синтезу даних.

Аналіз останніх досліджень і публікацій. Наразі існує багато досліджень деперсоніфікації даних та їх способів обробки, оскільки зараз стрімко розвиваються системи, що побудовані з використанням машинного навчання, основний предмет дослідження якого - це дані. І відповідно інструменти для таких задач теж користуються великою популярністю. Є рішення, що створені як портативні системи, то ж їх можна отримати, склонувавши репозиторій та запустивши локально. Рішення здебільшого дозволяють обробити дані різними методами підстановки, маскування, підміни чи заміни за допомогою регулярного виразу. Також, є подібні додатки, заточені більше для допомоги розробки систем і перевірки на навантаженість чи заповнення хоча б первинними даними, проте вони так само пропонують генерацію даних для уникнення використання реальних персональних даних. Вони дозволяють вказати діапазон даних, тип та застосувати реляційний підхід як у SQL-баз даних. Є інший вид розв'язку у деяких викладках - основний принцип базується на підході відтворення заданої математично кривої даних, щоб зрозуміти загальний розподіл та спроектувати нові результати, не порушуючи загальну концепцію. Кожен з

додатків вирішує певну проблему і може використовуватись у специфічних завданнях, що інколи не перетинаються між собою (тренування моделей чи тестування нового додатку, як приклад). Проте дана галузь досить широка, і зробити одне рішення, яке підходить для усіх завдань, неможливо. Дана робота пропонує метод деперсоніфікації даних шляхом поєднання заміни та синтезу даних з урахуванням статистичних результатів, екстраполюючи на необхідну кількість.

Виділення недосліджених частин загальної проблеми. Наразі тема деперсоніфікації даних не є повністю вивченою, але вже активно застосовується у машинному навчанні. А саме при аналізі не вдалося знайти рішень, що пропонують вилучення персональних характеристик у поєднанні з синтезом даних, і при цьому зберігають природу існуючих даних. Окрім деперсоніфікації, синтез даних відбувається на основі математичних показників, тому природа даних зберігається, не порушуючи початкових параметрів, а нові дані опосередковуються та наближаються до існуючих під час генерації, що теж знижує шанси визначення конкретного користувача.

Мета дослідження. Мета цієї статті – визначити, чи може бути ефективною деперсоніфікація та синтез даних зі збереженням природи даних. Для досягнення цієї мети було проведено набір експериментів, які передбачають викладку аналітичних метрик, за якими здійснюється оцінка доцільності та якості проробленої роботи.

Виклад основного матеріалу. Спершу варто зрозуміти, що саме розробляється і які результати оцінюються, оскільки сама тема передбачає вже певну проблему, яку можна вирішити різними методами. Також деперсоніфікація даних у нашому світі є доцільною через низку причин, таких як регламент захисту даних Європейського Союзу (GDPR) [3], можливі штрафи, кримінальна відповідальність осіб чи компаній, репутаційні збитки, суспільні резонанси, загальні віяння людей з намаганнями залишати менший "інформаційний слід" за собою в мережі. Тому перейдемо до частини, де можливі оцінки та викладки для аналізу, що свідчать про ефективність чи якість роботи. Такою частиною може стати область застосування інструментів машинного навчання. Вагомість анонімізації отриманих даних та синтезу на їх основі може бути суттєвою при зустрічі з проблемою «недотренування» (underfitting), що згадувалась раніше. Тезисно ця проблема описується як недостатня складність кривої, що формується відповідно поданих значень, таким чином уособлюючи в собі велику віддаленість від необхідних точок-значень, що були задані. Як нижче показано на рисунку 1, крива зображується таким чином, що занадто віддалена від потрібних значень параметрів. Або ще можна описати як наявність великої кількості виключень, що показано на рисунку 2.

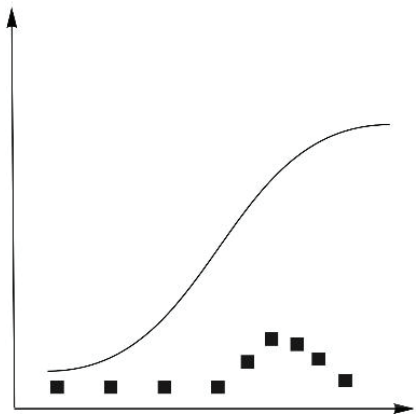

Рис. 1. Проблема недотренування

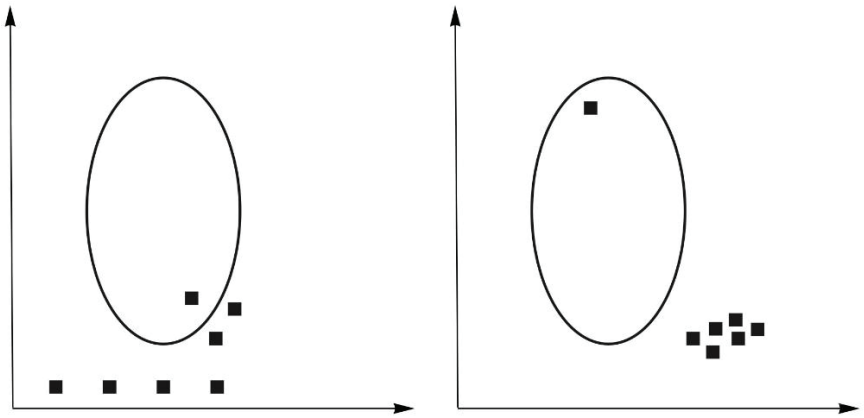

Рис. 2. Проблема великої кількості виключень

Отже, одне з можливих рішень може бути необхідність генерації додаткових даних, на основі яких модель натренується і буде відповідати більш точному графіку, якщо ми говоримо про проблеми регресії, наприклад. Відразу необхідно зазначити, що оскільки проводимо анонімізацію і синтез нових даних, то для точності результатів варто брати показники для передбачення первинних – справжніх отриманих даних, бо хоч і комплекс анонімізації і синтез даних зберігають природу проблеми і їх створення, проте вони все одно залишаються штучними. Перш, ніж заглиблюватися в проблему, треба описати певні деталі процесу. Для кращого розуміння весь алгоритм програми зображений на рисунку 3 нижче. Деякі блоки далі будуть описані детальніше та розширені.

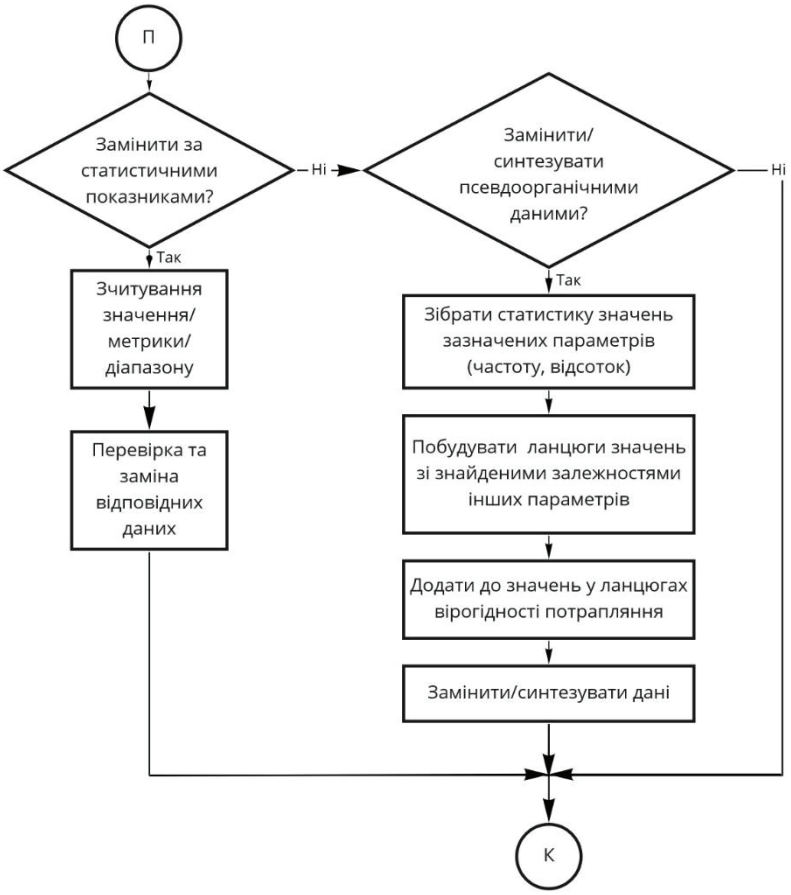

Рис. 3. Схема роботи алгоритму

А, отже, як бачимо, ми можемо замінювати дані статистично – за певними параметрами, заміною певних значень чи опираючись, на статистичні показники. А є й інша можливість – заміна залежно від інших даних. Одне з завдань машинного навчання може бути передбачення ціни на медичне страхування залежно від стану людини. З теорії основ машинного навчання ми знаємо, що співзалежні величини не завжди гарно включати, бо вони посилюють вагомість одне одного і нівелюють показники такої ж логічної вагомості, але не підкріплені схожим показником. Як приклад, не варто вказувати рік народження та кількість років людини одночасно (за умови, що певні роки не були специфічними для народження у виборці), бо ці величини співзалежні – знаючи рік народження, можемо порахувати вік без проблем і навпаки, тобто значення несуть один і той самий зміст, підкріплюючи одне одного. Проте є й закономірні значення, як такі що стан певних органів може бути кращим чи гіршим залежно від способу життя, харчування, наявності поганих звичок чи подібних параметрів. Тож, розуміємо, що дані досить пов'язані і обирати навмання значення ознак серед існуючих не може бути доцільним. Тому дана робота пропонує побудову взаємозв'язків між параметрами у багатовимірній матриці, де зберігається кожне значення параметра і будується ланцюг значень інших параметрів з можливими значеннями при використанні першого параметра. Також фіксується частота зустрічі певних значень при описаних ситуаціях.

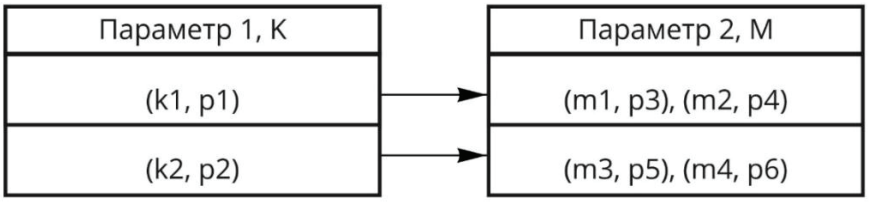

Рис. 4. Зображення взаємозв'язків між параметрами у ланках

Варто зазначити, що весь процес обробки та генерації відбувається з рядками таким чином, що відразу очікується, що дані нормалізовані, піддаються порівнянню, приведені до одного типу в колонках та якість відповідає очікуваним можливим результатам користувача, який працює над задачею. Адже значна частина відповідальності за успіх припадає саме на якість початкових даних. То ж, повернувшись до алгоритму, в результаті побудови матриці матимемо статистичні дані спершу про те, які можливі значення інших параметрів при використанні певного значення обраного параметра. Після побудови матриці вже відбудеться збереження статистичних даних про природу даних, які ми отримали. То ж наразі маємо можливість побудувати нові рядки, підтримуючи їх природність. Враховуючи те, що нам доступна частота входжень та загальна кількість, то можемо отримати вірогідність потрапляння у новий набір.

$$P(A) = \frac{m}{n},$$

де n – загальна кількість входжень, m – кількість входжень, які дорівнюють величині, для якої вираховується ймовірність, P(A) – ймовірність події A, де A – входження обраного значення.

Таким чином, побудова нового рядка виглядатиме як визначення ймовірностей і перегляд нових значень шляхом включень та виключень. Застосуємо один з типових підходів, який використовується у задачах машинного навчання, при необхідності визначення нового значення. А саме він полягає у тому, що ймовірність розподіляється на секторі від 0 до 1, а можливі значення займають відповідний простір на секторі у довжину своєї ймовірності.

Псевдовипадковим чином визначається значення від 0 до 1, знаходиться точка на секторі і визначається значення першого параметра. Тобто нехай вираховуємо значення серед a, b, c. Ймовірність потрапляння a – P(A), припустимо, дорівнює 0.25, P(B), припустимо, дорівнює 0.25, P(C), припустимо, дорівнює 0.5. Тоді значення на секторі від 0 до 0.25 відповідають a, від 0.25 до 0.5 – b і від 0.5 до 1 – відповідають C. Припустимо, що випадковим чином випало значення 0.75, тоді використовуємо значення c.

Тепер повертаємось до матриці, знаходимо відповідний рядок з відповідним значенням параметра і переходимо до його можливих значень другого параметра (до ланцюга, який вище був згаданий та описаний). Зберігаємо весь ланцюг, оскільки перший параметр був обраний саме з цього ланцюга. Аналогічним чином визначаємо значення другого параметра серед доступних (ті, що містяться у ланці збереженого рядка) через ймовірність. Тобто в даному випадку, знайшли ланцюг зі значенням параметра c. Знайшли можливі значення другого параметра. Припустимо, що це x, y, z з ймовірностями p1, p2, p3. Тоді перерахували шанси за вже відомою формулою і умовно зображуємо у довжину ймовірності на секторі і генеруємо псевдовипадкове значення – отримуємо нове значення другого параметра рядка. Записуємо до комірки новостворених даних обране значення параметра, і переходимо до визначення як будуть надалі визначатись дані. А, отже, нове значення другого параметра шукаємо у матриці відповідного стовпчика і забираємо отриманий ланцюг. Отже, тепер вже маємо визначатись з третім параметром. Беремо доступні значення третього параметра з першого та другого ланцюга. Перераховуємо ймовірності відповідно до нового можливого набору і аналогічним чином визначаємо третій параметр. Знову знаходимо ланцюг з третім визначеним параметром, зберігаємо і алгоритм повторюється знову, доки усі параметри не будуть визначені.

Рис. 5. Відображення ланок ланцюга, що впливають на вибір значень

Також при визначенні нових значень варто враховувати чи значення параметра можуть бути обрані з фіксованого набору, як булеве значення – true чи false (0 чи 1), чи як будь-яке значення з діапазону. У другому випадку, варто розглянути варіант визначення мінімального та максимального значень, та при визначенні ймовірності застосувати певну дельту залежно від порядку значень та порахованої частоти.

Результати. Тестування проводиться на моделях машинного навчання з розглядом завдань класифікації та регресії.

1. Завдання регресії, RandomForestRegressor, дані без змін.
2. Завдання регресії, RandomForestRegressor, дані зі статичними змінами.
3. Завдання регресії, RandomForestRegressor, дані зі змінами на основі матриці зі збереженням природи даних.

4. Завдання класифікації, RandomForestClassifier, дані без змін.
5. Завдання класифікації, RandomForestClassifier, дані зі статичними змінами.
6. Завдання класифікації, RandomForestClassifier, дані зі змінами на основі матриці зі збереженням природи даних.

На наступних графіках зображено ознаки та правильність передбачення даних відносно їх справжніх значень. Прямі на графіку (рис. 8-10) відображають ідеальну ситуацію – передбачення повністю відповідають справжнім даним, тому чим ближче значення до прямої, тим краще. Графіки демонструють модель без змін, зі статичними змінами і зі змінами на основі матриці відповідно. Завдання полягало у визначенні ціни на медичне страхування.

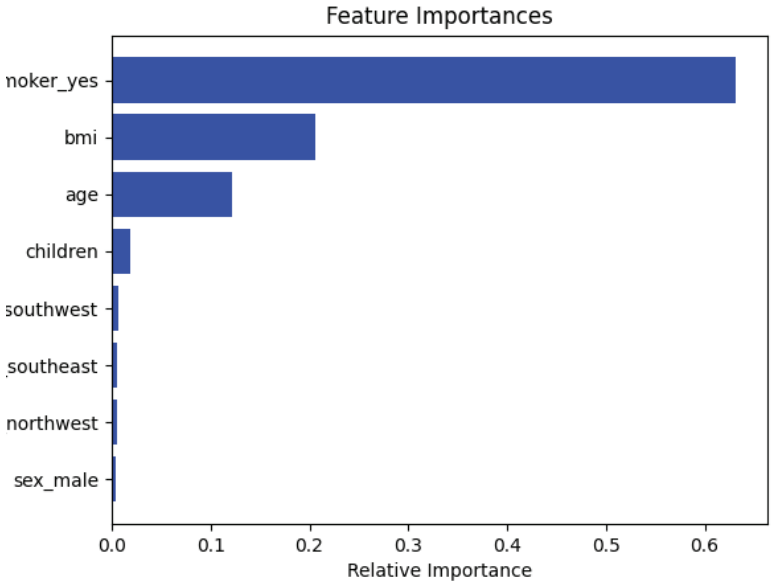

Рис. 6. Графік складових впливу ознак на точність моделі (завдання регресії)

```
Variable: smoker_yes          Importance: 0.63
Variable: bmi                 Importance: 0.21
Variable: age                 Importance: 0.12
Variable: children            Importance: 0.02
Variable: region_southeast    Importance: 0.01
Variable: region_southwest    Importance: 0.01
Variable: sex_male            Importance: 0.0
Variable: region_northwest    Importance: 0.0
```

Рис. 7. Значення складових впливу ознак на модель (завдання регресії)

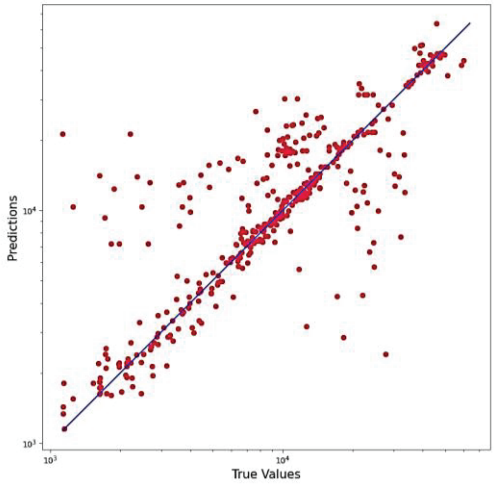

Рис. 8. Передбачення на даних без змін (завдання регресії)

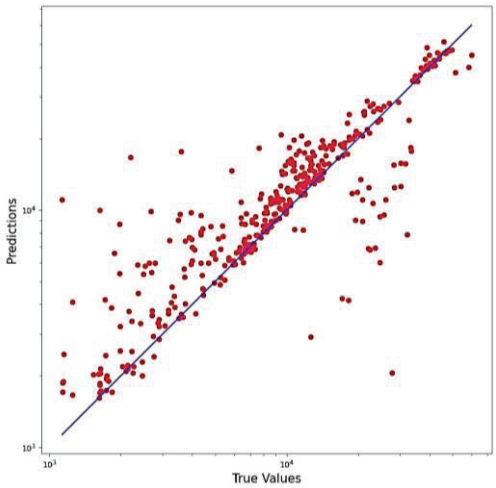

Рис. 9. Передбачення на даних зі статичними змінами (завдання регресії)

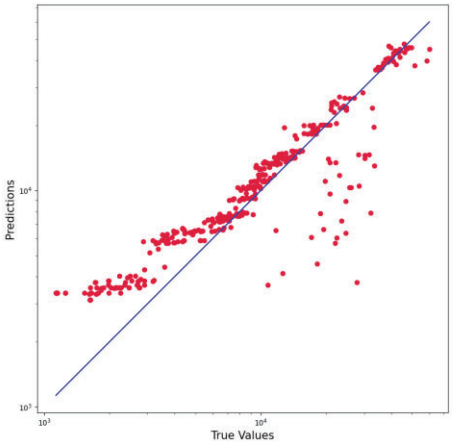

Рис. 10. Передбачення на даних зі змінами на основі матриці (завдання регресії)

Знайдемо показники моделі при різних даних:

Таблиця 1

Показники моделі (завдання регресії)

Дані	R2 score	Explained variance score	Mean absolute percentage
До змін	0,734668	0,739642	0,439573
Статичні зміни	0,822780	0,824976	0,350458
На основі матриці	0,856572	0,856581	0,301050

Джерело: розробка автора.

Отже, за показниками можемо переконатись, що обробка даних призвела до покращеного результату. Переглянемо чи будуть покращення при застосуванні методів, викладених у роботі, зі завданням класифікації. Для аналізу була обрана задача визначення надання кредиту особі. Оскільки, задача класифікації, тому зображено лише ознаки на графіках, а нижче, в таблиці, зафіксовані показники.

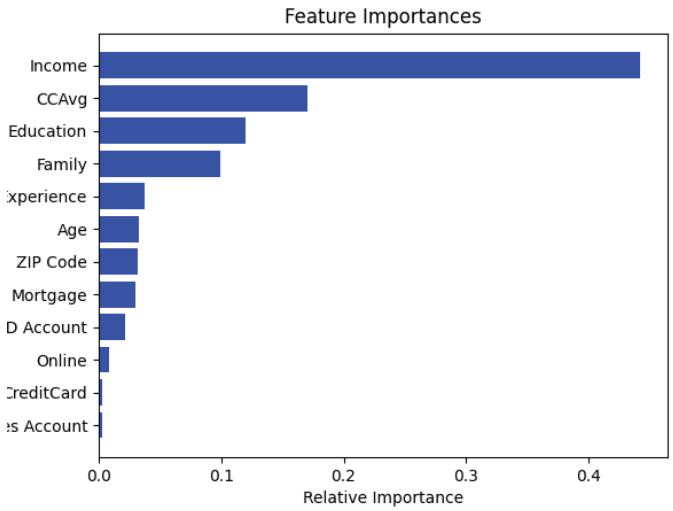

Рис. 11. Графік складових впливу ознак на точність моделі (завдання класифікації)

```
Variable: Income               Importance: 0.44
Variable: CCAvg                Importance: 0.17
Variable: Education            Importance: 0.12
Variable: Family               Importance: 0.1
Variable: Experience           Importance: 0.04
Variable: Age                  Importance: 0.03
Variable: ZIP Code             Importance: 0.03
Variable: Mortgage             Importance: 0.03
Variable: CD Account           Importance: 0.02
Variable: Online               Importance: 0.01
Variable: Securities Account   Importance: 0.0
Variable: CreditCard           Importance: 0.0
```

Рис. 12. Значення складових впливу ознак на модель (завдання класифікації)

Таблиця 2

Показники моделі (завдання класифікації)

Дані	**Accuracy**	**Precision**	**Recall**	**F1 score**	**False positive rate**
До змін	0,964000	0,933333	0,636364	0,756757	0,363636
Статичні зміни	0,976000	0,970588	0,750000	0,846154	0,250000
На основі матриці	0,984000	0,973684	0,840909	0,902439	0,159091

Джерело: розробка автора.

Завдання класифікації після застосування методів теж зазнало покращень і результати стали більш точні. Показово, що F1-score суттєво покращився – комплексна міра успішності моделі. Важливо орієнтуватися на F1-score у задачах класифікації через відому проблему «визначення терориста» [8], у якій може бути окремо високе значення precision або recall, проте програма буде неефективною. Тоді з'являється показник F1-score – міра успіху моделі (враховує і precision, і recall). Дані результати свідчать про врахування такої особливості, бо показники покращились.

Висновок. У викладеному матеріалі була висвітлена проблема використання даних та освітлені можливі шляхи їх анонімізації у разі потреби. Були зазначені можливі проблеми недбалого користування та способи уникнення таких варіантів. Розглянуто, що проблема притаманна різним галузям, тому і методи розв'язку можуть бути різними. Описано як статичні заміни, так і заміни на основі органічної природи даних. Варто пам'ятати, що відповідальність за якість даних лежить на користувачеві. Більш детально було висвітлено необхідність подібних утиліт у машинному навчанні та за його допомогою була виконана перевірка результатів. Не в останню чергу якість і необхідність залежить від потреб задачі та самої моделі, проте були взяті класичні проблеми машинного навчання і визначені показники.

Було з'ясовано, що результати, як завдання регресії, так і завдання класифікації, були покращені. Покращення сягають 20 % і особливості правильної оцінки – застосування F1-score – було враховано.

У замірах усі результати ставали кращими, проте з іншими моделями можуть відрізнятися. У задачах машинного навчання необхідний комплексний підхід до розв'язання, тому дана розробка теж не є виключенням і нею варто користуватися з розумом, аналізуючи та вдосконалюючи дані, моделі, результати.

Список використаних джерел

1. Big & Personal: data and models behind Netflix recommendations [Електронний ресурс]. – Режим доступу: https://xamat.github.io/pubs/BigAndPersonal.pdf.

2. Aurelien Geron. (2019). "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems", 2nd Edition (p.57)

3. General Data Protection Regulation [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/General_Data_Protection_Regulation.

4. If an app asks to track your activity [Електронний ресурс]. – Режим доступу: https://support.apple.com/en-us/HT212025.

5. Shai Shalev-Shwartz and Shai Ben-David. (2014). "Understanding Machine Learning: From Theory to Algorithms" [Електронний ресурс]. – Режим доступу: https://www.cs.huji.ac.il/~shais /UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf.

6. Andrew Ng. "Machine Learning by Stanford University", Coursera [Електронний ресурс]. – Режим доступу: https://www.coursera.org/learn/machine-learning.

7. Deep Learning – An MIT Press Book [Електронний ресурс]. – Режим доступу: https://www.deeplearningbook.org.

8. Beyond Accuracy: Precision and Recall [Електронний ресурс]. – Режим доступу: https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c.

References

1. Big & Personal: data and models behind Netflix recommendations. URL: https://xamat.github.io/pubs/BigAndPersonal.pdf

2. Aurelien Geron. (2019). "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems", 2nd Edition (p. 57).

3. GDPR. URL: https://en.wikipedia.org/wiki/General_Data_Protection_Regulation.

4. If an app asks to track your activity. URL: https://support.apple.com/en-us/HT212025.

5. Shai Shalev-Shwartz and Shai Ben-David. (2014). "Understanding Machine Learning: From Theory to Algorithms". URL: https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf.

6. Andrew Ng. "Machine Learning by Stanford University", Coursera. URL: https://www.coursera.org/learn/machine-learning.

7. Deep Learning – An MIT Press Book. URL: https://www.deeplearningbook.org.

8. Beyond Accuracy: Precision and Recall. URL: https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c.

Отримано 10.08.2021

UDC 004.65

Vlada Lipska
Student
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" (Kyiv, Ukraine)
E-mail: vladkalipskaya@gmail.com. **ORCID:** https://orcid.org/0000-0002-9847-7637

THE METHOD FOR INCREASING THE EFFICIENCY OF DATABASE DEPERSONIFICATION

The topic is relevant due to the need to protect personal data during its storage or use in different systems, so the demand for anonymization of data is growing every day.

The need for depersonalization is often mentioned, which is confirmed by the large number of competitors and the results of the materials. The materials offer static replacement, relational format replacement, data curve reproduction and replacement based on it.

The topic now needs to be deepened and it is proposed to consider depersonalization by various methods, one of which is by preserving the nature of data and use synthesis as a method of improving the results of the issue.

The aim is to develop a method of depersonalization and data synthesis for use without distortion. To achieve this goal, a set of experiments was conducted, which involve the calculation of analytical metrics, which are used to assess the feasibility and quality of the work done.

The principles of depersonalization methods are described, the emphasis is on anonymization with preservation of data nature. At the end, the analysis is performed and the results are presented.

The presented material highlighted the problem of data use and highlighted possible ways to anonymize them if necessary. It was found that the results of both the regression problem and the classification problem were improved. Improvements could reach 20 %.

Keywords: *depersonalization; anonymization; data.*

Fig.: 12. Tab.: 2. References: 8.