Будуючи модель класифікатора необхідно визначити, яка подія буде позитивною, а яка негативною. Так як наша задача полягає у виявленні шахрайських випадків, то відповідно позитивною подією буде наявність факту шахрайства (клас 1), а негативною – його відсутність (клас 0).

Перед переходом до процесу безпосереднього моделювання необхідно визначитись з розподілом даних на набори для навчання та тестування.

Для оцінки достовірності моделей будемо користатися одноразовою перехресною перевіркою, яка реалізується розбиттям вибірки на взаємодоповнювані підвибірки: навчальну (чи тренувальну (training)) та тестову (testing). Перша слугує для тренування моделей, а друга для оцінки результатів [23].

Розділимо підготовані дані на підвибірки, виділивши на навчання 70 % випадків, та здійснимо перевірку такого розподілу, рис. 3.1.

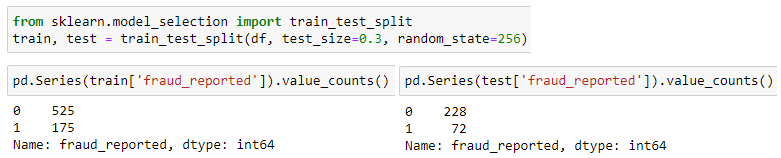


Рис. 3.1. Розбиття даних на набір для навчання та тестування (70/30)

*Джерело: авторська розробка*

Як і було зауважено раніше, ми маємо справу з незбалансованою вибіркою, і тестовий, і навчальний набір мають розподіл класів 75 % на 25 %. Клас шахрайських випадків, який цікавить нас найбільше, виступає класом меншості.

Незбалансовані набори даних поширені в багатьох областях і секторах, і, звичайно ж, це включає фінансові послуги. Фахівці з обробки даних стикаються з ними у багатьох контекстах – від шахрайства до проблемних кредитів. Алгоритми машинного навчання намагаються ідентифікувати ці поодинокі випадки у досить великих наборах даних. Через невідповідність класів у змінних алгоритм має тенденцію відноситись до класу більшості, водночас даючи помилкове відчуття високоточної моделі. Нездатність передбачити рідкісні події, що належать до меншості, так і точність, що вводить в оману, відволікають від побудованих прогностичних моделей [26]. Навчання моделей на таких даних може призвести до неправильної класифікації та великої кількості помилкових визначень, бо модель буде орієнтуватися на виявленні негативних випадків, у той час, коли нас цікавлять саме позитивні.

Щоб уникнути цієї проблеми, необхідно на навчання моделі подати набір даних з рівномірним розподілом класів чи перебалансувати навчальну вибірку таким чином, щоб якомога більше значень класу шахрайства потрапила на тренування моделей. Скористаємося цими способами перебалансування та створимо ще два варіанти розбиття вибірки.

Існує відомий варіант роботи з незбалансованими даними, який називається ресемплінг (передискретизація). Його суть полягає або у видаленні елементів з занадто великого набору (андерсемплінг) та/або додаванні більшої кількості елементів в недостатньо великий набір (оверсемплінг) [27]. Принцип роботи цього способу представлено на рис. 3.2.

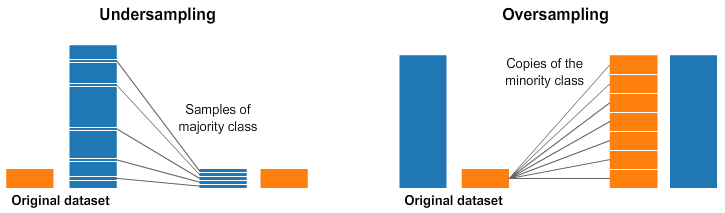


Рис. 3.2. Принцип роботи передискретизації

*Джерело: [28]*

У нашій ситуації, коли датасет складається з досить невеликої кількості випадків, користатися андерсемплінгом вважаю недоцільним, тому використаємо метод оверсемплінгу.

Розподіл класів навчального набору даних до проведення балансування вибірки наведено на рис. 3.3.

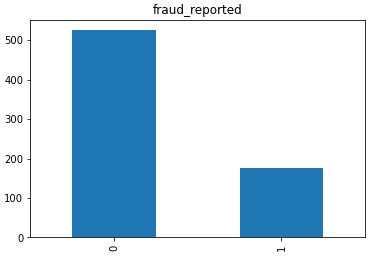


Рис. 3.3. Розподіл класів навчального набору даних

*Джерело: авторська розробка*

Найпростішим способом реалізації надмірної вибірки є дублювання випадкових записів із класу меншості. Слід зазначити, що недоліком такої вибірки може стати перенавчання моделі, що може призвести до поганого узагальнення тестового набору.

Використання рандомного оверсемплінгу дозволяє досягти балансу класів навчальної вибірки, збільшивши її на 450 шахрайських випадків, рис. 3.4.

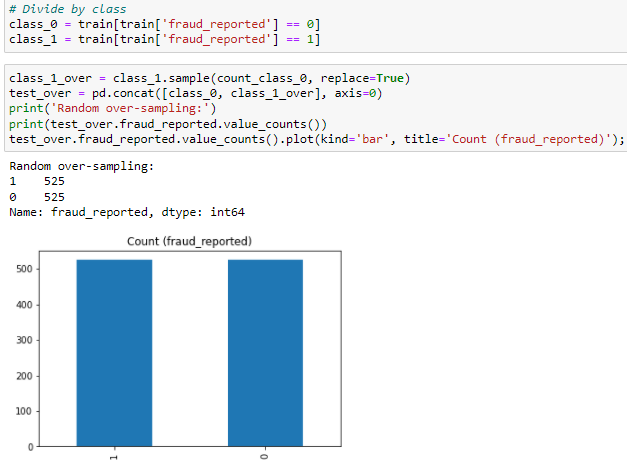


Рис. 3.4. Балансування вибірки методом рандомного оверсемплінгу

*Джерело: авторська розробка*

Другий варіант використання оверсемплінгу, було реалізовано завдяки методу SMOTE (Спосіб Передискретизації Синтезованих Меншин), що створює елементи в безпосередній близькості від тих, що вже існують у меншому наборі.

Для SMOTE обирається кілька спостережень та використовується міра відстані для синтетичного створення нового екземпляра з тими самими властивостями на доступних ознаках. Аналізуючи по одному об'єкту за раз, SMOTE враховує різницю між спостереженням та його найближчим сусідом. Він збільшує різницю на випадкове число від нуля до одиниці. Потім визначає нову точку, додаючи випадкове число до об'єкта. Таким чином, SMOTE не копіює спостереження, а натомість створює новий, синтетичний на основі наявних даних [26]. На рис. 3.5 приведено принцип роботи методу SMOTE.

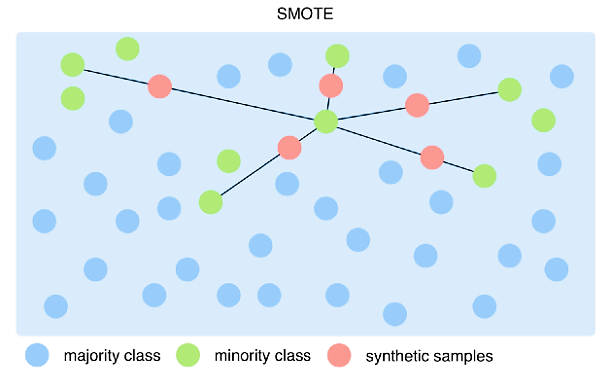


Рис. 3.5. Принцип роботи методу оверсемплінгу SMOTE

*Джерело: [26]*

Реалізуємо оверсемплинг, використовуючи бібліотеку imbalanced-learn (imblearn), що створена для боротьби з проблемами незбалансованих наборів даних. Результат представлено на рис. 3.6.

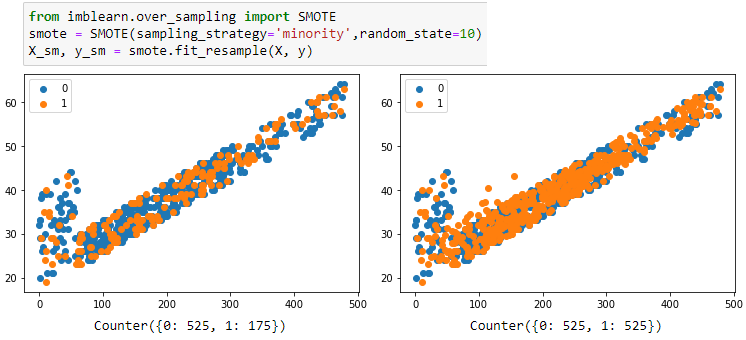


Рис. 3.6. Результат перебалансування методом SMOTE

*Джерело: авторська розробка*

З наведеної візуалізації можна побачити, що було створено 450 синтетичних шахрайських випадків, а класи відтепер рівно збалансовані 1:1.

Для спрощення згадки у тексті роботи отриманих наборів даних визначимо їх наступним чином:

* Вибірка № 0: дані розділені на набір для навчання та тестування (70/30), співвідношення класів для обох підвибірок складає 75 % випадків з відсутністю шахрайства на 25 % шахрайських претензій.
* Вибірка № 1: створена шляхом перебалансування класів рандомним оверсемплінгом таким чином, що навчальна вибірка збільшилась на 450 шахрайських випадків завдяки дублюванню існуючих записів.
* Вибірка № 2: дані навчального набору були перебалансовані шляхом синтетичного створення для класу меншості 450 випадків наближених до шахрайських.

Оцінка якості моделі логістичної регресії на усіх змінних вибірки № 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Модель | |  | |
| Негативно | Позитивно | *Точність* | 0,89 |
| Фактично | No | **212** | **16** | *Частка помилок* | 0,11 |
| *Чутливість* | 0,76 |
| Yes | **17** | **55** | *Специфічність* | 0,92 |
| *AUC* | 0,937 |

Провівши моделювання логістичної регресії на трьох наборах даних, порівняємо отримані результати:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Набір даних | Показник якості | | | | |
| **Accuracy** | **Error** | **Se** | **Sp** | ***AUC*** |
| Вибірка №0 | 0,89 | 0,11 | 0,76 | 0,92 | 0,937 |
| Вибірка №1 | 0,87 | 0,13 | 0,903 | 0,859 | 0,941 |
| Вибірка №2 | 0,866 | 0,13 | 0,902 | 0,855 | 0,941 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Модель | Вибірка №1 | | | | | Вибірка №2 | | | | |
| **Accuracy** | **Error** | **Se** | **Sp** | **AUC** | **Accuracy** | **Error** | **Se** | **Sp** | **AUC** |
| **LR** (усі змінні) | 0,87 | 0,13 | 0,903 | 0,859 | 0,941 | 0,866 | 0,13 | 0,902 | 0,855 | 0,941 |
| значущі | 0,87 | 0,13 | 0,916 | 0,855 | 0,939 | 0,88 | 0,12 | 0,93 | 0,864 | 0,945 |
| **SVM** (усі змінні) | 0,866 | 0,13 | 0,944 | 0,84 | 0,932 | 0,863 | 0,136 | 0,931 | 0,84 | 0,937 |
| значущі | 0,866 | 0,13 | 0,944 | 0,84 | 0,936 | 0,866 | 0,13 | 0,944 | 0,84 | 0,94 |
| **KNN** (усі змінні) | 0,787 | 0,21 | 0,77 | 0,79 | 0,84 | 0,63 | 0,37 | 0,7 | 0,6 | 0,843 |
| значущі | 0,857 | 0,14 | 0,944 | 0,82 | 0,889 | 0,84 | 0,16 | 0,931 | 0,811 | 0,906 |
| **NB** (усі змінні) | 0,62 | 0,37 | 0,88 | 0,54 | 0,852 | 0,646 | 0,35 | 0,875 | 0,57 | 0,85 |
| значущі | 0,73 | 0,27 | 0,875 | 0,68 | 0,873 | 0,86 | 0,14 | 0,916 | 0,84 | 0,926 |
| **DT** (усі змінні) | 0,866 | 0,13 | 0,944 | 0,84 | 0,913 | 0,866 | 0,13 | 0,944 | 0,84 | 0,907 |
| **RF** (усі змінні) | 0,846 | 0,15 | 0,83 | 0,85 | 0,883 | 0,85 | 0,146 | 0,71 | 0,899 | 0,909 |
| значущі | 0,873 | 0,126 | 0,93 | 0,855 | 0,914 | 0,87 | 0,13 | 0,875 | 0,868 | 0,922 |
| **CNN** (усі змінні) | 0,863 | 0,136 | 0,931 | 0,84 | 0,94 | 0,876 | 0,123 | 0,902 | 0,868 | 0,941 |
| значущі | 0,89 | 0,11 | 0,916 | 0,88 | 0,94 | 0,88 | 0,12 | 0,931 | 0,864 | 0,945 |