Самостійна робота №3
з алгоритмів машинного навчання

Горбунов Даніел Денисович
3 курс бакалаврату
група "комп'ютерна статистика"

11 травня 2020 р.

Вступ. Маємо такі ж дані, що були використані в другій лабораторній роботі (*crx.data*, *crx.names*). Цього разу була проведена класифікація даних за допомогою чотирьох моделей: метод опорних векторів, вирішуче дерево, випадковий ліс та алгоритм бустінгу *AdaBoost*. Вказані лише найкращі результати. Кілька слів про невдачі, як у схемі Бернуллі, будуть у висновках.

Обробка даних. Врахуємо те, що було написано у висновках минулої роботи. Використовуємо усі колонки характеристик, проводимо центрування та нормування вибірки. Останний крок необхідних для отримання адекватних результатів з відносно "чутливих" моделей, як SVM.

Опорний метод векторів. Дані не є лінійно сепарабельними, тому звичайну модель SVM розглядати не дасть користі. Використаємо нелінійне перетворення з поліноміальним ядром п'ятого степеня та поклавши у відповідній оптимізаційній задачі $C = 12$.

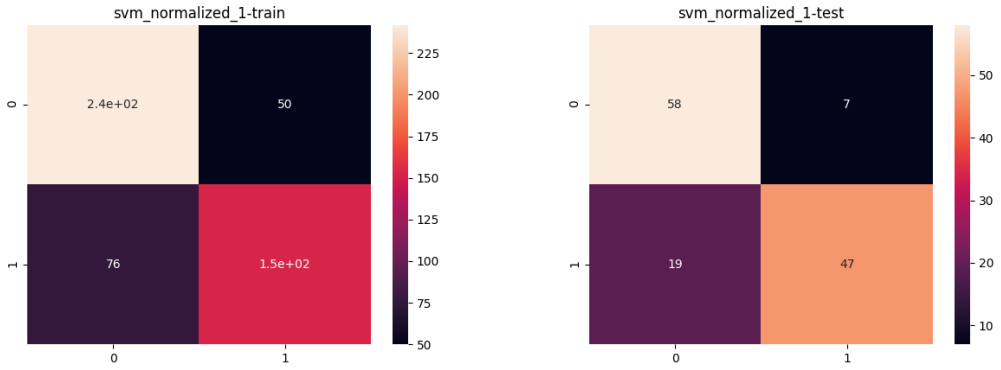

Рис. 1: Матриця спряженості для тренувальної і тестової вибірки.

	Train	precision	recall	f1-score	support	**Test**	precision	recall	f1-score	support
0.0	-	0.76	0.83	0.79	292	-	0.75	0.89	0.82	65
1.0	-	0.75	0.67	0.71	229	-	0.87	0.71	0.78	66
macro avg	-	0.76	0.75	0.75	521	-	0.81	0.8	0.8	131
weighted avg	-	0.76	0.76	0.75	521	-	0.81	0.8	0.8	131

$accuracy - train = 0.7581573896353166$
$accuracy - test = 0.8015267175572519$

Аналогічного результату можна добитися, використавши Гаусове ядро з параметрами $\gamma = 2.75$ та $C = 16$. В інших випадках, зазначені метрики набувають значень, для довільного тесту, в діапазоні $.6 \sim .75$. C оптимально підібрано, бо при більших значеннях маємо перенавчання моделі.

Вирішуче дерево. Перша спроба. Побудовано класифікатор на базі алгоритму вирішуючого дерева з ентропійним критерієм, без будь-яких модифікацій.

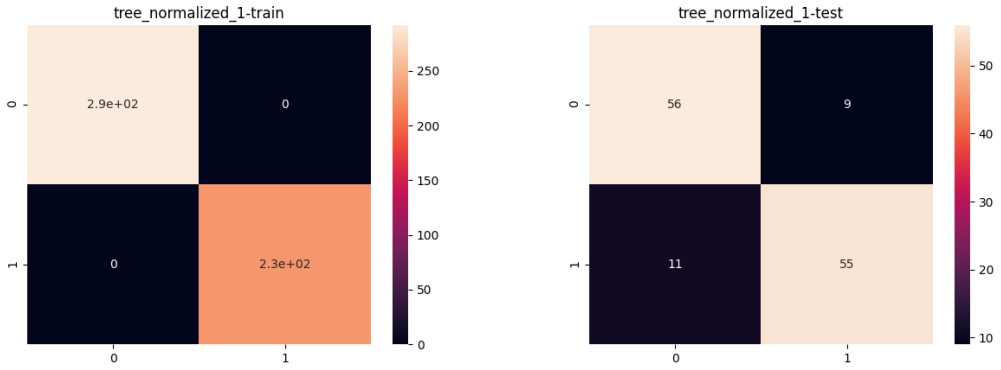

Рис. 2: Матриця спряженості для тренувальної і тестової вибірки.

	Train	precision	recall	f1-score	support	**Test**	precision	recall	f1-score	support
0.0	-	1	1	1	292	-	0.84	0.86	0.85	65
1.0	-	1	1	1	229	-	0.86	0.83	0.85	66
macro avg	-	1	1	1	521	-	0.85	0.85	0.85	131
weighted avg	-	1	1	1	521	-	0.85	0.85	0.85	131

$accuracy - train = 1.0$
$accuracy - test = 0.8473282442748091$

З результатів для тренувальної вибірки можна вважати, що алгоритм зміг підібрати непогане розбиття простору даних. Показники точності та ефективності цієї моделі суттєво кращі порівняно з тими, що мали для опорних векторів.

Покращення моделі вирішучого дерева. Підбір оптимального α. Визначення оптимального значення α у формулі цінового критерію, яке б мінімізувало задану функцію, буде виконано за допогою алгоритму визначення оптимального шляху відповідності між α та мірою забрудненості вузла в моделі.

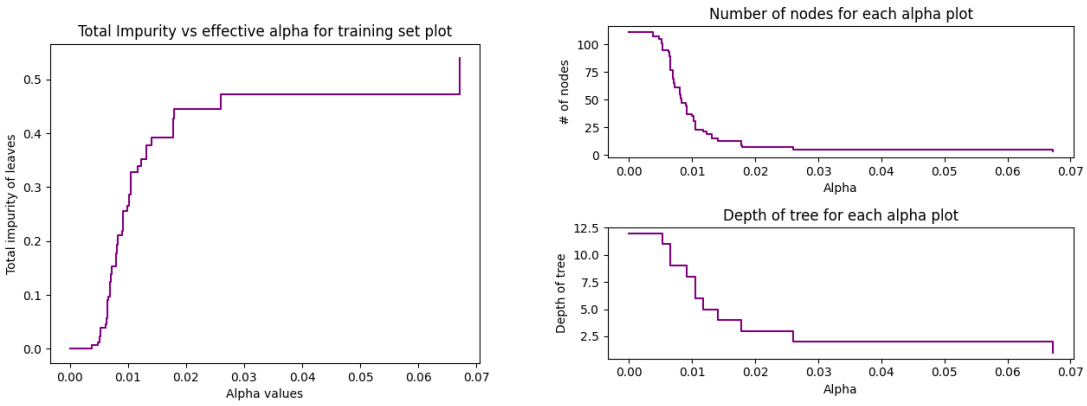

Рис. 3: Зліва: Оптимальний шлях відповідності між необхідним параметром та значенням забрудненості. Справа: Графіки залежності глибини дерева та його кількості вузлів в залежності від значення параметра.

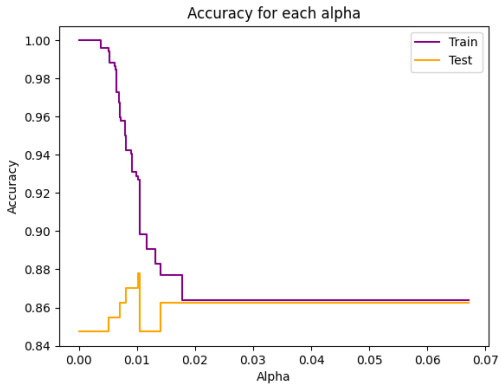

Рис. 4: Графік залежності якості моделі від значення параметра для тренувальної та тестової вибірки.

За допомогою вищезазначеної техніки отримали оптимальне $\alpha = 0.01$. З рисунків видно різкий злам показників при незначному збільшенні α.

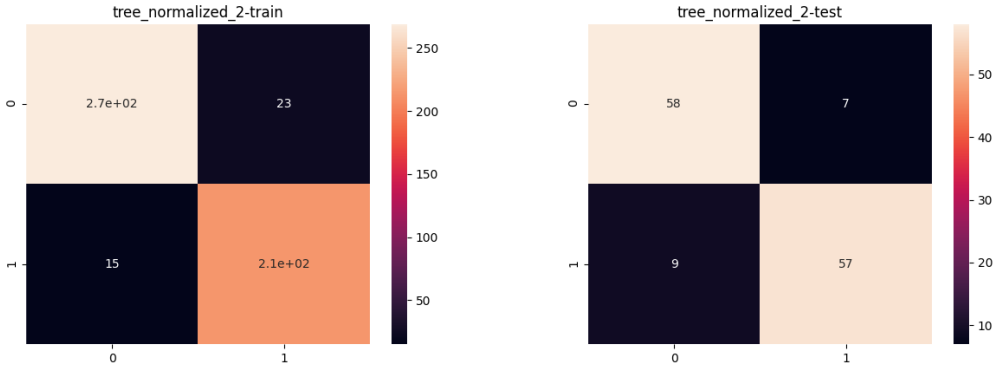

Рис. 5: Матриця спряженості для тренувальної і тестової вибірки.

	Train	precision	recall	f1-score	support	**Test**	precision	recall	f1-score	support
0.0	-	0.95	0.92	0.93	292	-	0.87	0.89	0.88	65
1.0	-	0.9	0.93	0.92	229	-	0.89	0.86	0.88	66
macro avg	-	0.93	0.93	0.93	521	-	0.88	0.88	0.88	131
weighted avg	-	0.93	0.93	0.93	521	-	0.88	0.88	0.88	131

$accuracy - train = 0.927063339731286$
$accuracy - test = 0.8778625954198473$

Метрики на тренувальній вибірці гірші за попередні, але модель краще "відносить" до різних класів на новій вибірці. Ймовірно що у першій модель дерева прийняття рішень трапилося перенавчання.

Випадковий ліс. Використали чисту модель з критерієм Джині. На відміну від попередньої моделі, є ефективнішою.

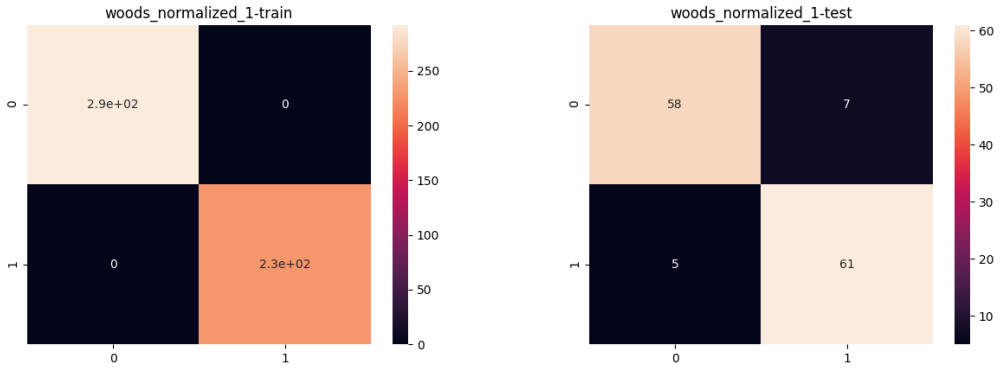

Рис. 6: Матриця спряженості для тренувальної і тестової вибірки.

	Train	precision	recall	f1-score	support	**Test**	precision	recall	f1-score	support
0.0	-	1	1	1	292	-	0.92	0.89	0.91	65
1.0	-	1	1	1	229	-	0.9	0.92	0.91	66
macro avg	-	1	1	1	521	-	0.91	0.91	0.91	131
weighted avg	-	1	1	1	521	-	0.91	0.91	0.91	131

$accuracy - train = 1.0$
$accuracy - test = 0.9083969465648855$

Нижче наведені результати класифікації у тому випадку, якщо вхідні дані не були пронормовані.

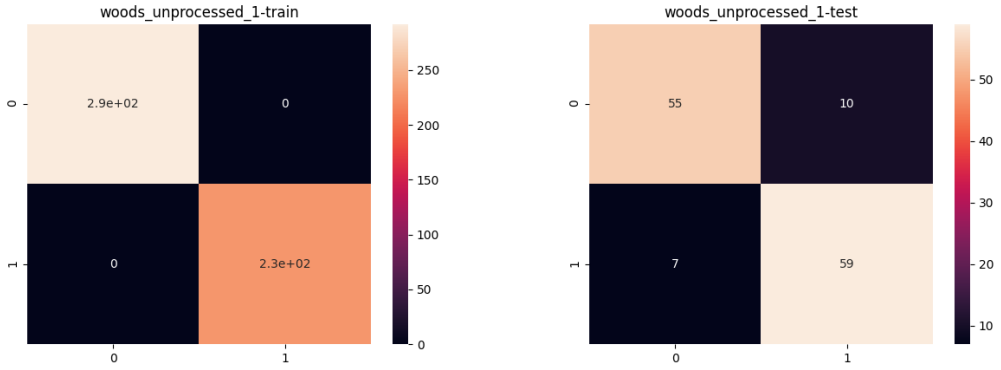

Рис. 7: Матриця спряженості для тренувальної і тестової вибірки.

	Train	precision	recall	f1-score	support	**Test**	precision	recall	f1-score	support
0.0	-	1	1	1	292	-	0.89	0.85	0.87	65
1.0	-	1	1	1	229	-	0.86	0.89	0.87	66
macro avg	-	1	1	1	521	-	0.87	0.87	0.87	131
weighted avg	-	1	1	1	521	-	0.87	0.87	0.87	131

$accuracy - train = 1.0$
$accuracy - test = 0.8702290076335878$

Покращення моделі вирішучого дерева. Бустінг за допомогою *AdaBoost*. Для бустінгу взяли вдосконалену модель вирішучого дерева. Встановлені параметри для алгоритму: ваговий параметр навчання (learning rate) рівний 0.25, кількість копій - 300.

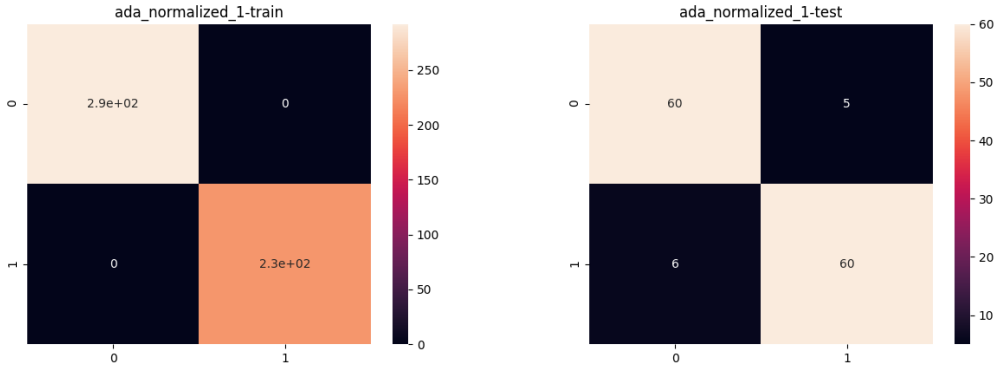

Рис. 8: Матриця спряженості для тренувальної і тестової вибірки.

	Train	precision	recall	f1-score	support	**Test**	precision	recall	f1-score	support
0.0	-	1	1	1	292	-	0.91	0.92	0.92	65
1.0	-	1	1	1	229	-	0.92	0.91	0.92	66
macro avg	-	1	1	1	521	-	0.92	0.92	0.92	131
weighted avg	-	1	1	1	521	-	0.92	0.92	0.92	131

$accuracy - train = 1.0$
$accuracy - test = 0.916030534351145$

Висновки. Використання моделей засованих на деревах було доцільним. Бустінг вирішучого дерева, за відповідних умов, покращив ситуацію в загальному. Метод опорних векторів частоко програє в якості класифікації в даному випадку.