Самостійна робота №2
з алгоритмів машинного навчання

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

27 квітня 2020 р.

Вступ. У даній роботі розглянуто підхід до розв'язання задачі бінарної класифікації. Дані були взяті з *crx.data*. Всього 16 колонок, серед яких у якості відгуку обрана остання, а факторною була тринадцята. Розглянуто такі моделі: наївний байєсівський класифікатор, метод k-найближчих сусідів, класифікація за допомогою логістичної регресії. Усі обчислення були проведені в *python* з використання таких пакетів, як: *sklearn*, *scipy* та ін.

Опрацювання даних на предмет пропущених значень. Перед використанням моделей класифікації на тих даних, що маємо, визначили пропущені значення набору. В таблиці нижче наведена кількість пропущених значень для кожної колонки, де такі випадки були зафіксовані.

	A1	A2	A4	A5	A6	A7	A14
Кількість	12	12	6	6	9	9	13

Загальний обсяг кожного об'єкта вибірки дорівнює 689. Кількість рядків, де відсутня інформація у деяких клітинах, рівна 37. Це не досить багато, але й не мало для такої вибірки. Однак такі рядки були видалені з набору: деякі дані були відсутні у тих колонках, які мали нечислову природу. Тому традиційні методи заміни значень неможливо використати.

Об'єкти з показниками нечислової природи були закодовані у вигляді натуральних чисел. Це було можливо реалізувати внаслідок скінченної кількості значень можливих результатів кожної такої колонки.

Надалі процес класифікації даних було розбито на три частини:

1. Побудова моделей на основі тих даних, що неперервно розподілені;
2. Побудова моделей з урахуванням всіх колонок вибірки;
3. Побудова моделей для кожної підмножині, яка залежить від значень факторної змінної.

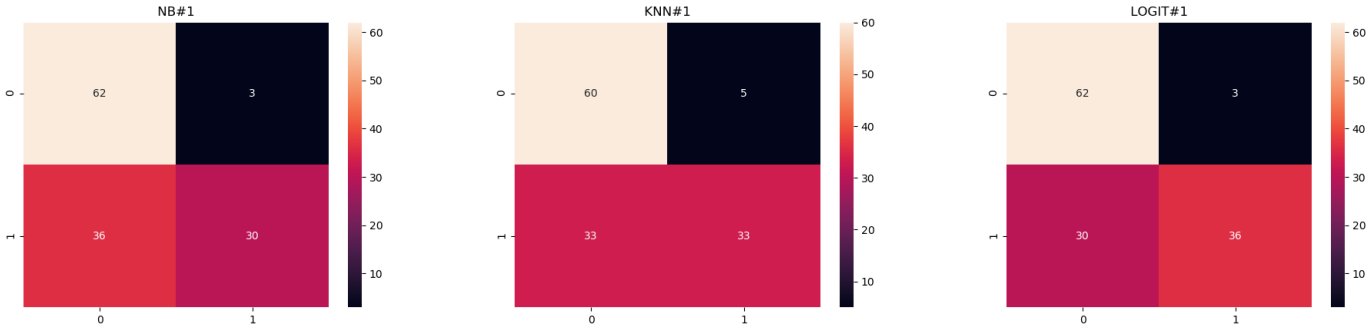

Рис. 1: таблиці спряженості (confusion matrix) для моделей (зліва-направо): $GaussianNB$, KNN, $LogisticRegression$.

Використання базових моделей. Перша частина. Як було зазначено раніше, тут беремо лише колонки з неперервно розподіленими даними. Результати хороші, але можна досягти кращих. На таблицях спряженості видно, що моделі досить вдало прогнозують значення '1', ніж '0'. Помилка другого роду є поширеною: майже половина тих елементів, які насправді носять значення '0', були помічені як '1'. Дещо краще проявила себе модель логістичної регресії, випереджуючи в кількості правильно вгаданих елементів з характеристикою '0' на 3-6 одиниць.

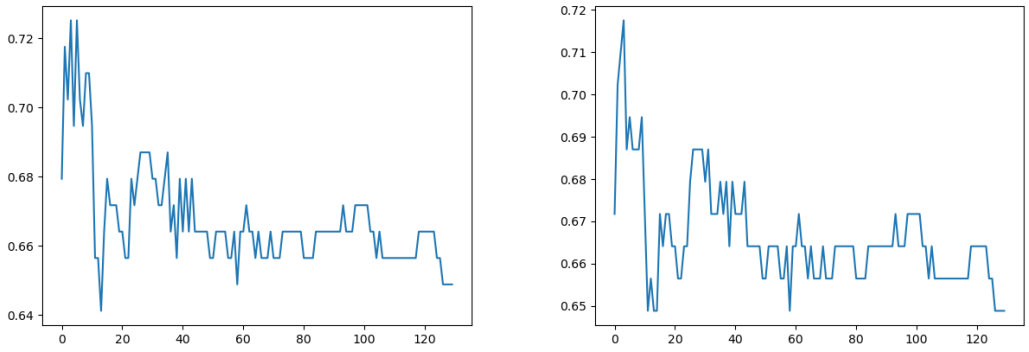

Рис. 2: Графік залежності метрики точності (accuracy) моделі KNN від кількості найближчих сусідів. Графік зліва - для моделі з метрикою Махаланобіса, справа - з використанням евклідової

Методом перебору чисел в певному діапазоні вдалося визначити оптимальну кількість сусідів для KNN. Починаючи з п'яти сусідів, метрики моделі спадають з 0.7 до 0.6.

Naive Bayes

	precision	recall	f1-score	support
0.0	0.63	0.95	0.76	65
1.0	0.91	0.45	0.61	66
accuracy			0.70	131
macro avg	0.77	0.70	0.68	131
weighted avg	0.77	0.70	0.68	131

Accuracy of Gaussian NB: 0.7023

KNN

	precision	recall	f1-score	support
0.0	0.65	0.92	0.76	65
1.0	0.87	0.50	0.63	66
accuracy			0.71	131
macro avg	0.76	0.71	0.70	131
weighted avg	0.76	0.71	0.70	131

Metric: Mahalanobis
Optimal number of neighbors: 6
Accuracy of opt KNN: 0.7023
Metric: Euclidean
Optimal number of neighbors: 4
Accuracy of opt KNN: 0.7176

Logistic

	precision	recall	f1-score	support
0.0	0.67	0.95	0.79	65
1.0	0.92	0.55	0.69	66
accuracy			0.75	131
macro avg	0.80	0.75	0.74	131
weighted avg	0.80	0.75	0.74	131

Accuracy of logistic regression classifier: 0.7481

Найкраща модель в даному випадку є класифікатор з використанням логістичної регресії. Спробуємо такі ж самі моделі, але тепер додамо ті колонки, які не використовували у першій частині.

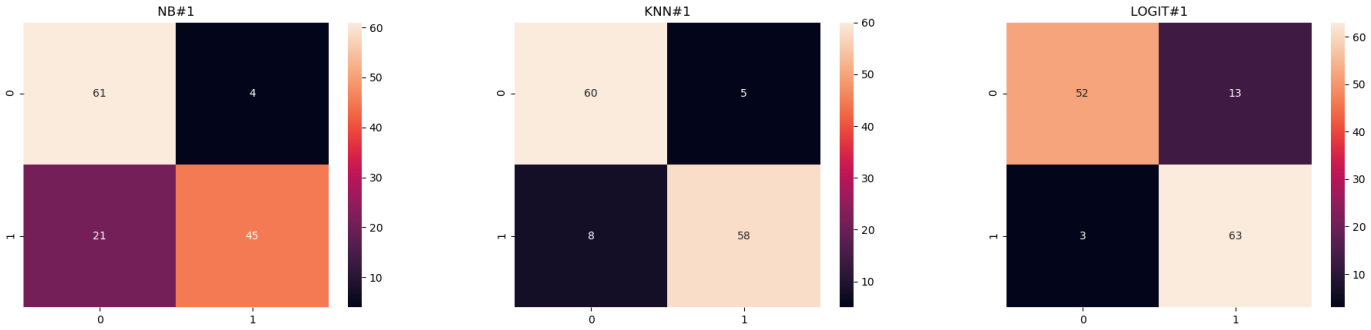

Рис. 3: таблиці спряженості (confusion matrix) для моделей (зліва-направо): $GaussianNB$, KNN, $LogisticRegression$.

Використання базових моделей. Друга частина Порівняно з результатами моделей першої частини розв'язку задачі класифікації, маємо кращі. Суттєво зменшилась похибка другого роду, але бачимо, що для логістичної моделі збільшилася похибка першого роду. У цьому разі досконалі результати показує модель KNN з використанням метрики Махаланобіса. Цікаво звернути увагу на зміну значень метрики при збільшенні кількості сусідів - графік подібний до параболи з вітками вниз.

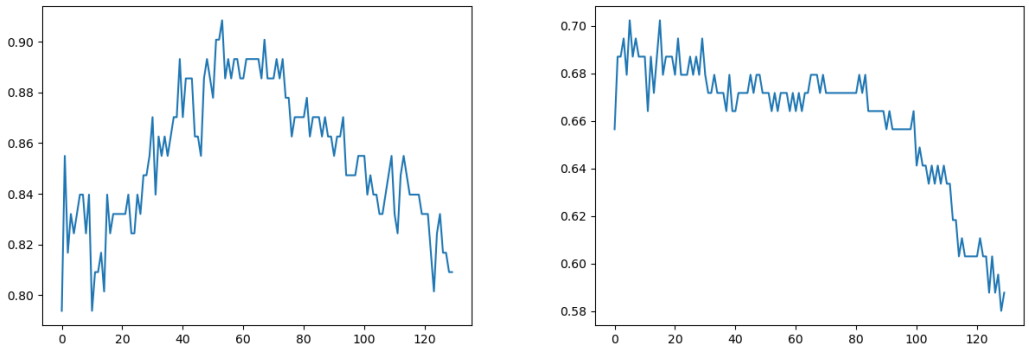

Рис. 4: Графік залежності метрики точності (accuracy) моделі KNN від кількості найближчих сусідів. Графік зліва - для моделі з метрикою Махаланобіса, справа - з використанням евклідової

KNN з евклідовою метрикою програє першій з суттєвим відривом, розглядаючи $recall$, $precision$, $accuracy$ (наведено лише значення останньої).

Naive Bayes

	precision	recall	f1-score	support
0.0	0.74	0.94	0.83	65
1.0	0.92	0.68	0.78	66
accuracy			0.81	131
macro avg	0.83	0.81	0.81	131
weighted avg	0.83	0.81	0.81	131

Accuracy of Gaussian NB: 0.8092

KNN

	precision	recall	f1-score	support
0.0	0.88	0.92	0.90	65
1.0	0.92	0.88	0.90	66
accuracy			0.90	131
macro avg	0.90	0.90	0.90	131
weighted avg	0.90	0.90	0.90	131

Metric: Mahalanobis
Optimal number of neighbors: 54
Accuracy of opt KNN: 0.9084
Metric: Euclidean
Optimal number of neighbors: 4
Accuracy of opt KNN: 0.7252

Logistic

	precision	recall	f1-score	support
0.0	0.95	0.80	0.87	65
1.0	0.83	0.95	0.89	66
accuracy			0.88	131
macro avg	0.89	0.88	0.88	131
weighted avg	0.89	0.88	0.88	131

Accuracy of logistic regression classifier: 0.8779

Для логістичної моделі менші значення *precision*, але кращий показник для F_1-міри.

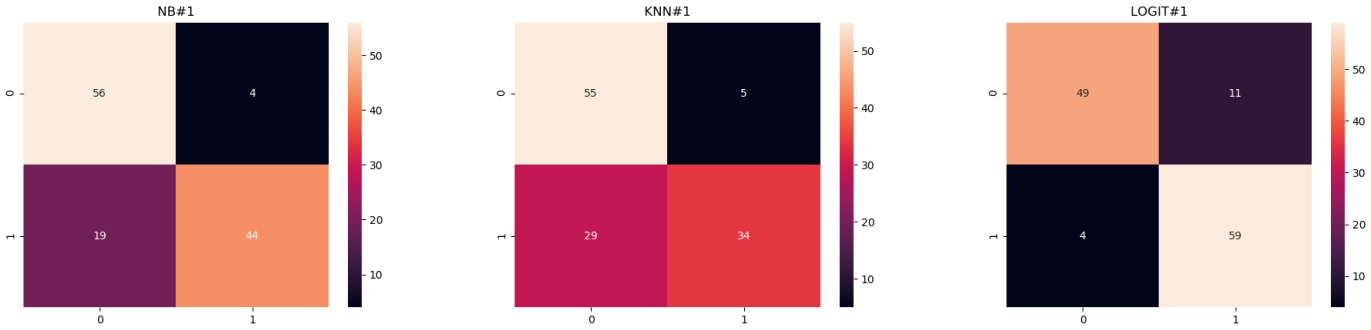

Рис. 5: таблиці спряженості (confusion matrix) для моделей (зліва-направо): $GaussianNB$, KNN, $LogisticRegression$. (пара p, g)

Використання базових моделей. Третя частина Була спроба задіяти факторну змінну "A13". Область значень вказаної змінної складається з трьох елементів, а саме: $\{s, g, p\}$. Проведення класифікації на основі підмножин потужності один неможливо, оскільки для деяким факторам відповідає зовсім мало елементів вибірки:

Number of rows after filtering by factor [s]:
Train: 45; Test: 8
Number of rows after filtering by factor [g]:
Train: 476; Test: 121
Number of rows after filtering by factor [p]:
Train: 0; Test: 2

Слушною думкою була ідея взяти дві підмножими потужності 2 та провести класифікацію даних для кожної. Допустимо було обрати $\{s, g\}$ та $\{p, g\}$.

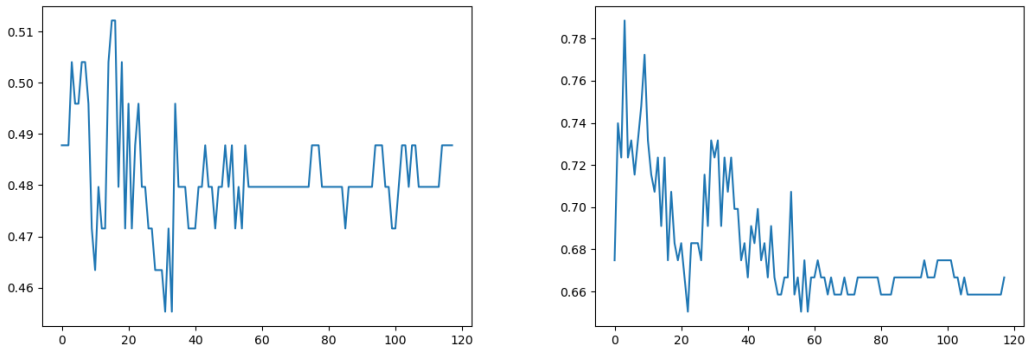

Рис. 6: Графік залежності метрики точності (accuracy) моделі KNN від кількості найближчих сусідів. Графік зліва - для моделі з метрикою Махаланобіса, справа - з використанням евклідової. (пара p, g)

Naive Bayes

	precision	recall	f1-score	support
0.0	0.74	0.94	0.83	64
1.0	0.92	0.68	0.78	65
accuracy			0.81	129
macro avg	0.83	0.81	0.80	129
weighted avg	0.83	0.81	0.80	129

Accuracy of Gaussian NB: 0.8062

KNN

	precision	recall	f1-score	support
0.0	0.88	0.92	0.90	64
1.0	0.92	0.88	0.90	65
accuracy			0.90	129
macro avg	0.90	0.90	0.90	129
weighted avg	0.90	0.90	0.90	129

Metric: Mahalanobis
Optimal number of neighbors: 54
Accuracy of opt KNN: 0.9070
Metric: Euclidean
Optimal number of neighbors: 4
Accuracy of opt KNN: 0.7209

Logistic

	precision	recall	f1-score	support
0.0	0.94	0.80	0.86	64
1.0	0.83	0.95	0.89	65
accuracy			0.88	129
macro avg	0.89	0.88	0.88	129
weighted avg	0.89	0.88	0.88	129

Accuracy of logistic regression classifier: 0.8760

Для першої підмножини показники кожної моделі майже співпали з минулими.

Naive Bayes

	precision	recall	f1-score	support
0.0	0.75	0.93	0.83	60
1.0	0.92	0.70	0.79	63
accuracy			0.81	123
macro avg	0.83	0.82	0.81	123
weighted avg	0.83	0.81	0.81	123

Accuracy of Gaussian NB: 0.8130

KNN

	precision	recall	f1-score	support
0.0	0.65	0.92	0.76	60
1.0	0.87	0.54	0.67	63
accuracy			0.72	123
macro avg	0.76	0.73	0.72	123
weighted avg	0.77	0.72	0.71	123

Metric: Mahalanobis
Optimal number of neighbors: 16
Accuracy of opt KNN: 0.5122
Metric: Euclidean
Optimal number of neighbors: 4
Accuracy of opt KNN: 0.7886

Logistic

	precision	recall	f1-score	support
0.0	0.92	0.82	0.87	60
1.0	0.84	0.94	0.89	63
accuracy			0.88	123
macro avg	0.88	0.88	0.88	123
weighted avg	0.88	0.88	0.88	123

Accuracy of logistic regression classifier: 0.8780

Гірші результати для KNN за Махаланобісом, але кращі для логістичної моделі, якщо порівняти з тією, що була побудована для першої підмножини. Наївний байєсівський класифікатор повертає стабільні, метрики.

Висновки. Використання запропонованих моделей на даних *crx.data* є доречним. Виявилося, що в цьому випадку кращі класифікатори можна отримати завдяки включенню усіх об'єктів вибірки. Рядками з фактором p можна знехтувати.