Міністерство освіти та науки України
Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Кафедра обчислювальної математики

КУРСОВА РОБОТА

на тему:

Класифікація на основі матриці співпадінь

Виконав студент III курсу
групи ОМ-3
Кутузов Владислав Павлович
Керівник курсової роботи
кандидат технічних наук
доцент кафедри
обчислювальної математики
Голубєва Катерина Миколаївна

Київ – 2021

Зміст

1 Вступ

У цій роботі йтиметься про процесс створення методу класифікації, який на основі зображення розподіляє людей однієї з трьох груп: хворих на рак молочної залози і здорових людей. Для процесу були надані зображення ядер букального епітелію людей відповідних категорій. Ідея методу базуватиметься на побудові і подальшій обробці матриці відстаней, генерування на її основі параметрів та навчання алгоритмів.

На початку роботи буде зроблений короткий теоретичний екскурс у область обробки та маніпулювання даними, що міститиме усі визначальні ідеї та принципи подальшої роботи.

2 Характеристики даних

2.1 Види ознак

У цьому розділі мова піде про ознаки в машинному навчанні. Існує кілька класів, або типів ознак. І у всіх свої особливості - їх потрібно по-різному обробляти і по-різному враховувати в алгоритмах машинного навчання. Ознаки описують об'єкт в доступній та зрозумілій для комп'ютера формі. Далі множина значень j-ї характеристики позначатиметься через D_j. Усі ознаки (дані) розподіляють на два класи: **числові** та **категорійні**.

Числові дані – дані, що представляються у вигляді числового значення. Начастіше їх розділяються на два класи: *дискретні* та *неперервні* ($D_j = \mathbb{R}$). Прикладами таких ознак є:

- вік;
- площа квартири;
- діаметр ядра клітини;
- температура.

Категорійні дані (ознаки) – дані що можуть приймати скінченну кількість значеннь, яка називають категоріями. Категорійні ознаки поділяють на **бінарні** ($D_j = \{T, F\}$) і **ординальні**. Прикладами категорійних даних є:

- Освіта
- Колір очей
- Місто
- Тип населеного пункту

Відмітна особливість категоріальних ознак - *неможливість порівняння «більше-менше» значення ознак*, тому вони дуже важкі в обігу та досі з'являються способи обліку цих ознак в тих чи інших методах машинного навчання.

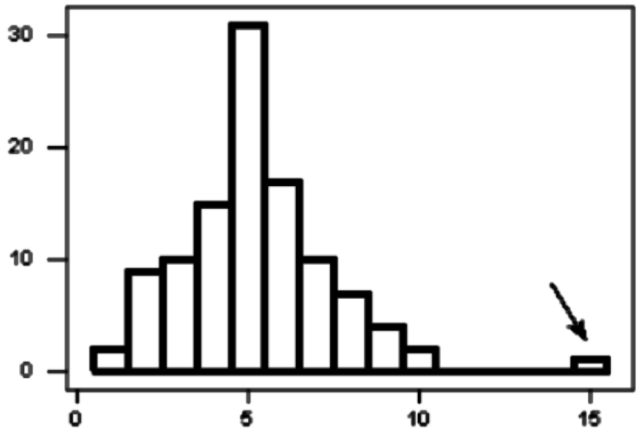
2.2 Розподіл ознаки

Одна з перших проблем з якими можна зіткнутися при роботі з ознаками – існування викидів.

Викидом називається такий об'єкт, значення ознаки на якому відрізняється від значення ознаки на більшості об'єктів. Наявність викидів представляє складність для алгоритмів машинного навчання, які будуть намагатися

врахувати і їх теж. Оскільки викиди описуються зовсім іншим законом, ніж основна множина об'єктів, викиди зазвичай виключають з даних, щоб не заважати алгоритму машинного навчання шукати закономірності в даних.

Проблема може бути і в тому, як розподілена ознака. Не завжди ознака має такий розподіл, яке дозволяє відповісти на необхідний питання. Наприклад, може бути занадто мало даних про клієнтів з невеликого міста, так як зібрати достатню статистику не є можливим.



Приклад викиду

3 Алгоритми машинного навчання

3.1 Навчання з учителем

У цьому розділі мова піде про те, які бувають типи завдань при навчанні на розмічених даних[1], або навчанні з учителем. Загальна постановка задачі навчання з учителем наступна. Для навчальної вибірки $X = (x_i, y_i)_{i=1}^{\ell}$ потрібно знайти такий алгоритм $a \in A$, на якому буде досягатися мінімум функціоналу (помилки):

$$Q(a, X) \to \min_{a \in A}$$

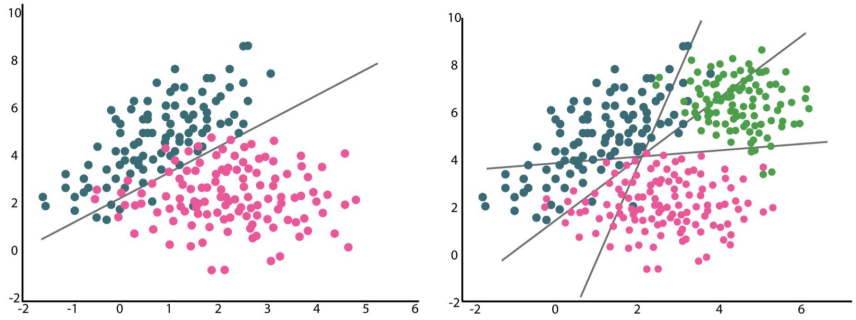
Залежно від множини можливих відповідей Y, задачі діляться на типи, найпоширенішими є:

- задач класифікації;
- задача регресії;

3.1.1 Задачі класифікації

Задана скінченна множина об'єктів X, для яких відомо, до яких класів вони належать. Ця множина називається вибіркою. До якого класу належать інші об'єкти невідомо. Необхідно побудувати такий алгоритм, який буде здатний класифікувати довільний об'єкт з вихідної множини.

Частковий випадок, коли множина відповідей Y є двоелементною, наприклад $Y = \{0, 1\} = \{T, F\}$ прийнято називати задачою **бінарної класифікації**. Якщо ж класів більше, ніж два, має місце задача **багатокласової класифікації**.



Задачі бінарної і багатокласової класифікації

3.1.2 Задача регресії

Коли множина можливих відповідей є дійсним числом, тобто $Y = \mathbb{R}$, кажуть про **задачу регресії**. Прикладом такої задачі є передбачення прибутку магазину базуючись на статистиці за попередні роки.

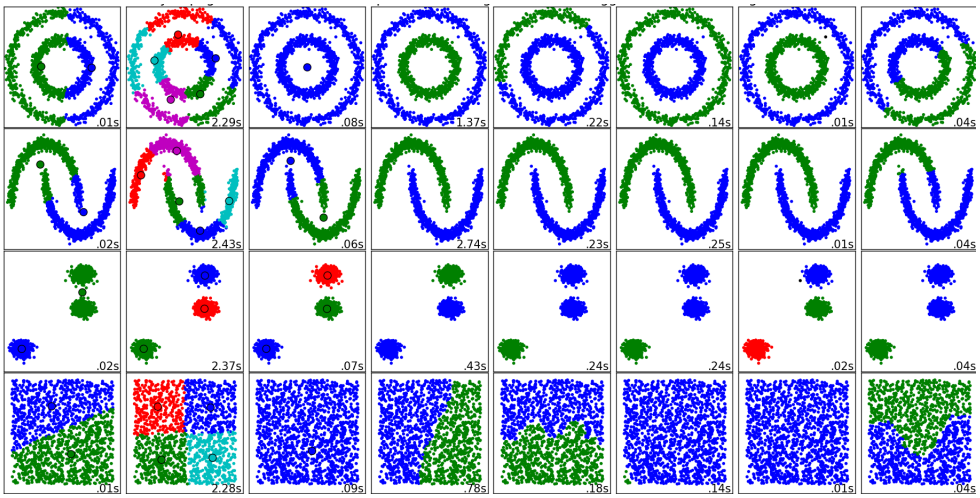
3.2 Навчання без вчителя

Завдання навчання без вчителя - це таке завдання, в якій є тільки об'єкти, а відповідей немає. Також бувають «проміжні» постановки. Прикладами таких задач є

- кластеризація;
- візуалізація;
- пошук аномалій

Кластеризація

Задача розбиття заданої вибірки об'єктів на підмножини, які називаються **кластерами**, так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися. Прикладами алгоритмів кластеризації є K-середніх (`K-means`), `DBSCAN`.



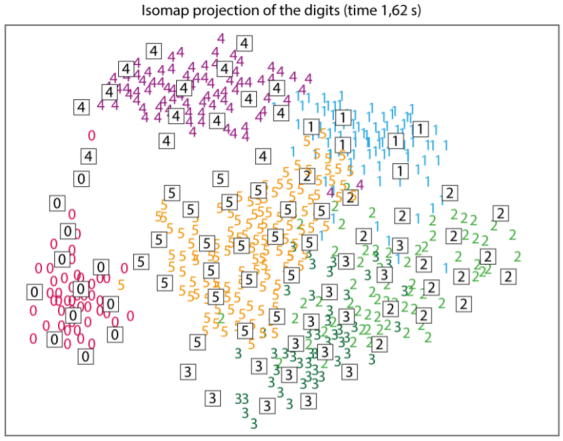
Кластеризація множин точок двовимірного простору

3.2.1 Візуалізація

Завдання візуалізації: необхідно намалювати багатовимірну (а конкретно, d-мірну) вибірку так, щоб зображення наочно показувало структуру об'єктів

Прикладом завдання візуалізації є завдання візуалізації набору даних `MNIST`. Цей набір даних був отриманий в результаті оцифровки рукописних

цифр. Кожен скан цифри характеризується вектором ознак - яркостей окремих пікселів. Необхідно таким чином відобразити цей набір даних на площину, щоб різні цифри виявилися в різних її областях.



Кластеризація множин точок двовимірного простору

3.2.2 Пошук аномалій

Третій приклад завдання навчання без учителя - пошук аномалій. Необхідно виявити, що даний об'єкт не схожий на всі інші, тобто є аномальним. При навчанні є приклади тільки звичайних, не аномальних, об'єктів. А прикладів аномальних об'єктів або немає взагалі, або настільки мало, що неможливо скористатися класичними методами навчання з учителем (методами бінарної класифікації). При цьому завдання дуже важлива. Наприклад, до такого типу завдань відноситься:

- визначення поломки в системах літака (за показниками сотень датчиків);
- визначення поломки інтернет-сайту;
- виявлення проблем в моделі машинного навчання.

Практичний аспект

На практиці Unsupervised Learning (навчення без вчителя) використовують зазвичай як методи аналізу та підготовки даних, а не як основний алгоритм, що вирішує конкретні завдання за допомогою цих даних [4].

У реальності добре розмічені дані – це велика рідкість, тому для їх розмітки зазвичай використовують або спеціальні сервіси, у яких реальні люди з країн з дешевою робочою силою (зазвичай Індії або Китаю) за мінімальну плату вручну класифікують дані, або спеціальні алгоритми для розмітки (які, в свою чергу, можуть також використовувати машинне навчання).

Великі корпорації, які мають величезний потік користувачів, можуть використовувати для розмітки своїх клієнтів. Наприклад, Google і його anti-captcha: знаходячи автобуси на фото, ви не тільки підтверджуєте, що «ви не робот», а й навчаєте нейронну мережу тому, який вигляд має автобус. У випадках, коли розмітка даних неможлива, вдаються до методів навчання без учителя. До таких алгоритмів як раз належать задачі кластеризації, зменшення розмірності і пошуку правил.

3.3 Проблеми алгоритмів

Припустимо при вирішенні задачі класифікації був побудований деякий алгоритм причому частка помилок на об'єктах з навчальної вибірки дорівнювала 0.2, і така частка помилок є допустимою. Але оскільки алгоритм не володіє узагальнюючої здатністю, немає ніяких гарантій, що така ж частка помилок буде для нової вибірки. Цілком може виникнути ситуація, що для нової вибірки помилка стане рівною 0.9. Це означає, що алгоритм не зміг узагальнити навчальну вибірку, і витягти з для класифікації нових об'єктів. Проте алгоритм якось налаштувався на навчальну вибірку і показав хороші результати при навчанні без вилучення справжньої закономірності, це і є **проблема перенавчання**. Основними причинами, що призводять до перенавчання є

- незбалансованість вибірки;
- невдало обраний функціонал помилки $Q(a, X)$;
- невдало обрана модель;
- неінформативні ознаки;

Гарними інструментами для боротьби з перенавчанням є *регуляризація*, *крос-валідація* і використання *відкладеної вибірки*[2].

4 Хід роботи

4.1 Початкова обробка

Використовуватимемо схему запропоновану в роботі [12]. Будемо розглядати цифрові зображення в кольоровій моделі RGB.

Нехай $I = \|I(x,y)\|_{N\times M}, x = \overline{1,N}, y = \overline{1,M}$ — зображення розміром $N \times M$. Кожен піксель характеризується яскравостями трьох кольорових компонент

$$I(x,y) = p(R,G,B),$$
$$R = \overline{0,255}, G = \overline{0,255}, B = \overline{0,255}$$

де R, G, B — яскравості червоної, зеленої та блакитної компоненти відповідно.

Для побудови матриці співпадінь зображення будемо розглядати його чорно-білий аналог $I(x,y) = p(Gray, Gray, Gray)$. Для перетворення використаємо

$$Gray = 0.299R + 0.587G + 0.114B$$

Матрицю співпадінь будемо позначати через $P_{(\Delta x,\Delta y)}$.

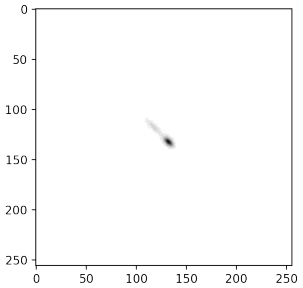
$$P_{(\Delta x,\Delta y)} = \|P(i,j)\|_{256\times 256}, i = \overline{0,255}, j = \overline{0,255}$$

, де її елементи $P(i,j)$ визначаються з формули

$$P(i,j) = \sum_{x=1}^{N}\sum_{y=1}^{M}\begin{cases} 1, & \text{if } I(x,y) = i \text{ and } I(x+\Delta x, y+\Delta y) = j \\ 0, & \text{otherwise .} \end{cases} \quad (1)$$

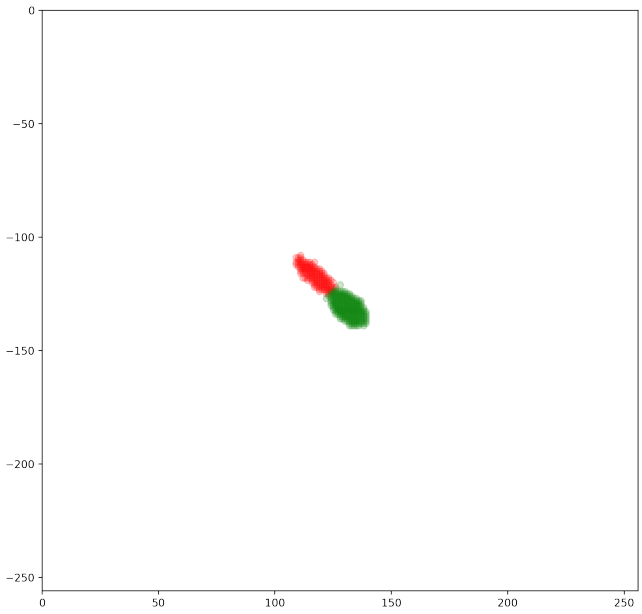
де (x,y) та $(x+\Delta x, y+\Delta y)-$ координати пікселей в зображенні I, а $\vec{d} = (\Delta x, \Delta y)-$ вектор зміщення. Складність алгоритму побудови матриці співпадінь дорівнює $O(N \times M)$.

Для кожного зображення I будуватимемо матрицю співпадінь. Для зручності можемо її візуалізувати: будемо вважати, що елемент матриці з найбільшим значенням відповідає чорному кольору з найменшим — білому, решта значень — градації сірого.



Таким чином кожен елемент матриці конвертується у яскравість відповідного пікселя зображення 256×256 (розмір матриці співпадінь). Приклад такої візуалізації наведений справа.

Описаним принципом візуалізації числової матриці будемо користуватися і в подальшому.



Приклад сегментованої матриці

Далі сегментуємо візуалізовану матрицю співпадінь, при цьому ледве помітні пікселі зображення враховувати не будемо: за 100% приймається найбільше значення в матриці співпадінь, якщо значення не перевищує 5%, то воно відкидається. Решта значень відносять до одної з двох можливих текстур T_1 чи T_2 для зручності позначатимемо їх різними кольорами.

Для сегментації матриці використовуватимемо EM-алгоритм (Expectation-maximization (EM) algorithm)[6] побудований на `Gaussian mixture model`. `Gaussian mixture model` – це імовірнісна модель, яка передбачає, що всі точки даних генеруються із суміші кінцевого числа розподілів Гауса з невідомими параметрами.

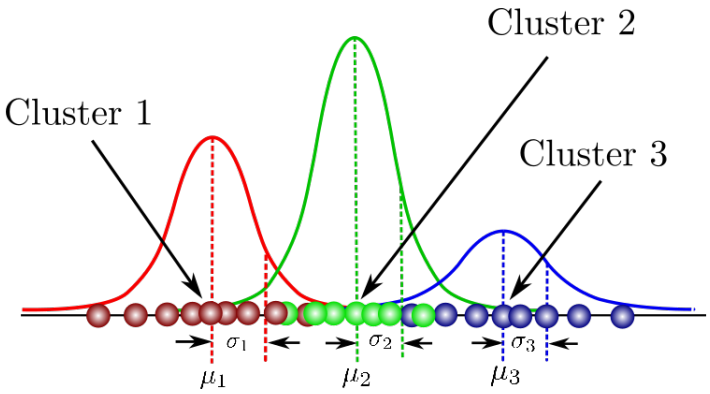
Кожна Гауссіана k описується відповідним вектором параметрів $\theta(k) = (\pi, \mu, \Sigma)$, де

- середнє μ_i визначає її центр;
- коваріація Σ_i визначає її ширину;
- імовірність змішування π_i, яка визначає, наскільки великою чи малою буде гауссіана;

Хід алгоритму:

1. На початку роботи алгоритму ми отримуємо початкові наближення $\theta(k) = (\pi, \mu, \Sigma)$[5]

$$\theta(k) = (\pi, \mu, \Sigma) = (\pi_1, \ldots, \pi_k; \mu_1, \ldots, \mu_k; \Sigma_1, \ldots, \Sigma_k) \tag{2}$$



Графічна ілюстрація параметрів, для трьох кластерів

в результаті роботи алгоритму k-Means (ця евристика застосовується, бо k-Means потрібно набагато менше ітерацій до досягнення стабільності, в той час як кожен крок EM вимагає великих обчислювальних витрат).

2. Далі ітеративно виконується наступна пара процедур

   (а) Використовуючи остання значення вектору θ знаходимо значення вектору "прихованих"змінних γ.

$$\gamma_{nk} = \frac{\pi_k \aleph\left(x_n \mid \mu_k, \Sigma_k\right)}{\sum_{j=1}^{K} \pi_j \aleph\left(x_n \mid \mu_j, \Sigma j\right)}$$

   (б) Переоцінка вектора параметрів, використовуючи поточне значення вектора прихованих змінних

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} x_n a$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \left(x_n - \mu_k^{new}\right) \left(x_n - \mu_k^{new}\right)^T$$

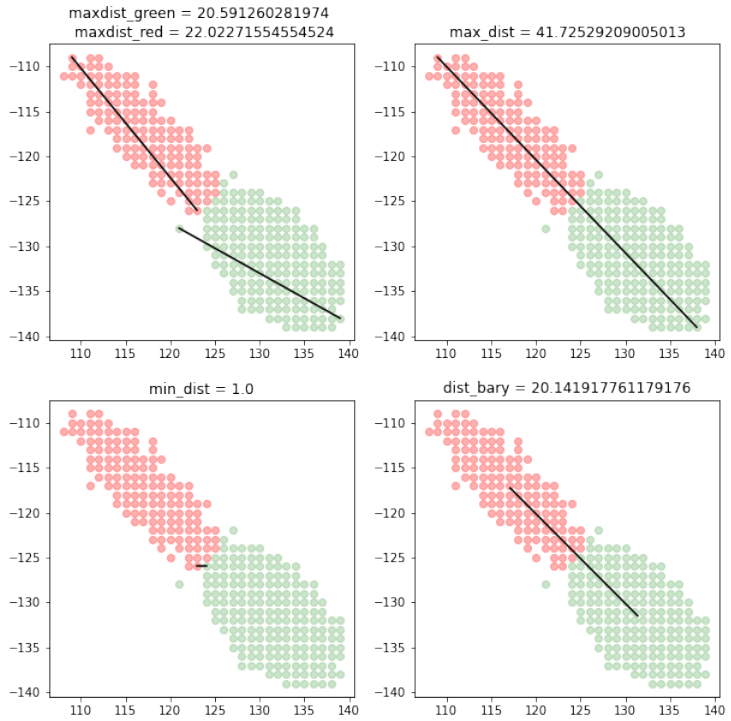
$$\pi_k^{new} = \frac{N_k}{N}$$

$$N_k = \sum_{n=1}^{N} \gamma_{nk}$$

   Процедура зупиняється коли, як норма різниці векторів $\|\gamma_{nk} - \gamma_{nk}^0\|$ прихованих змінних на кожній ітерації стане нижче заданої константи.

На основі сегментованої матриці розрахуємо 5 параметрів:

- максимальна відстані усередині кожної текстури (два значення);
- відстань між центрами текстур;

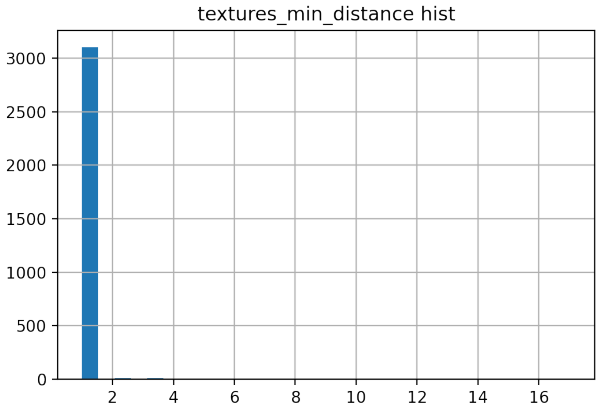


Іллюстрація параметрів зображення (збільшений масштаб)

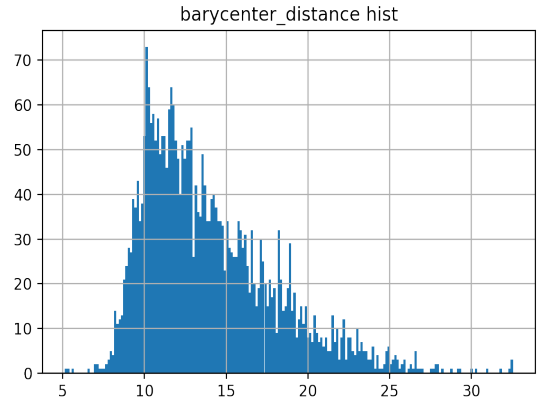
- максимальна відстань між двома текстурами;
- мінімальна відстань між двома текстурами

4.2 Аналіз вибірки

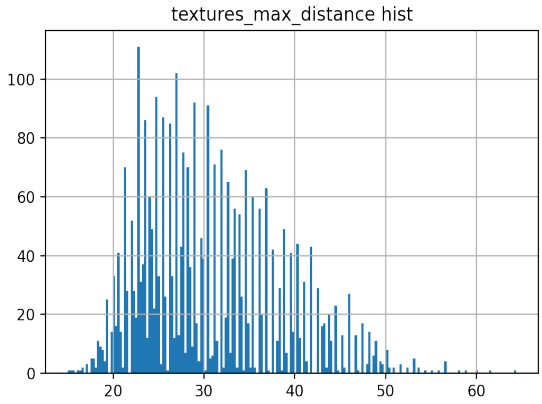
Для кожного з п'яти параметрів було побудовано гістограму, для розуміння розподілу і виявлення викидів.



Виявлено, що для переважної більшості зображень мінімальна відстань між текстурами `textures_min_distance` сегментованої матриці є рівною

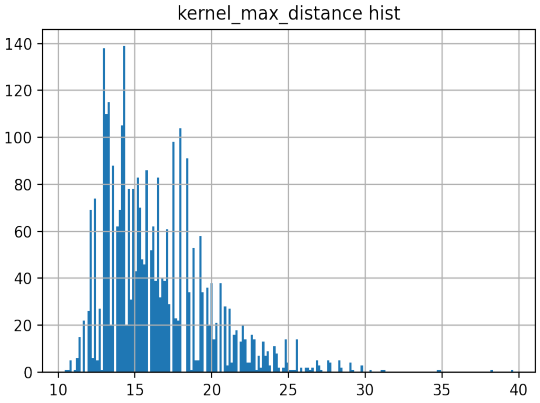


(а)

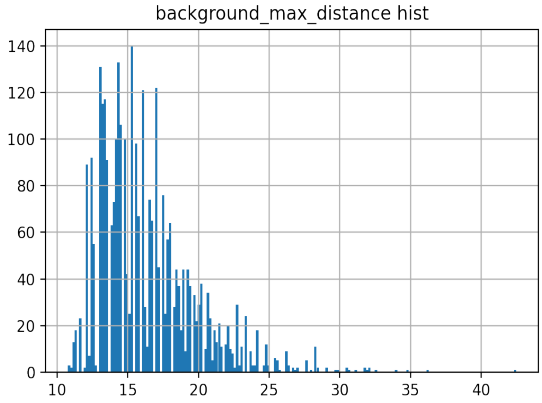


(б)

Рис. 1



(а)



(б)

Рис. 2

одиниці, отже дана ознака не є інформативною у контексті моделі, тому була прибрана. Також були прибрані записи, у яких `kernel_max_distance` > 27, `barycenter_distance` > 27 і `background_max_distance` > 33.

Остаточно матимемо наступну статистику

4.3 Моделі

Для цієї задачі були перевірені наступні моделі

1. RandomForest Classifier[7]
2. Logistic regression[8]
3. k-Nearest Neighbors algorithm classifier[9]
4. Decision tree[10]

	background_max_distance	kernel_max_distance	textures_max_distance	barycenter_distance
count	3021.000000	3021.000000	3021.000000	3021.000000
mean	15.972734	16.044878	29.686976	13.570045
std	2.896136	2.891899	6.688843	3.492251
min	10.770330	10.440307	15.000000	5.138974
25%	13.601471	13.928388	24.207437	10.856553
50%	15.297059	15.524175	28.425341	12.828382
75%	17.804494	17.888544	34.655447	15.788653
max	25.000000	25.000000	49.497475	26.840578

Рис. 3: Базові характеристики навчальної вибірки

5. Naive Bayes classifier[11]

Для оцінки моделей використовувалися наступні метрики якості:

accuracy(a, X) – визначає долю правильно класифікованих об'єктів відносно усієї вибірки.

$$\text{accuracy}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) = y_i]. \tag{3}$$

precision(a, X) – показує, наскільки можна довіряти класифікатором в разі, коли він визначає пацієнта в групу `Control`:

$$\text{precision}(a, X) = \frac{TP}{TP + FP} \tag{4}$$

recall(a, X) – показує, на якій частці справжніх об'єктів першого класу алгоритм спрацьовує:

$$\text{recall}(a, X) = \frac{TP}{TP + FN} \tag{5}$$

де TP, TN, FP, FN – відповідні елементи матриці помилок

Табл. 1: Результати lkz

Metric	RFC	Log. reg.	KNN	DT	Naive Bayes
accuracy(a, X)	83%	66%	74%	74%	94%
precision(a, X)	88%	80%	79%	81%	96%
recall(a, X)	84%	63%	93%	74%	91%

5 Висновки

В результаті роботи ми створили алгоритм, що за зображенням букального епітелію класифікує людину до однієї з двох груп: BC or Control. В остаточний алгоритм увійшла модель Naive Bayes

Опис програми:

1. Обробка зображення;
2. Розрахунок матриці співпадінь
3. Сегментація двох текстур на основі матриці
4. Розрахунок п'яти параметрів базуючись на сегментованому зображені
5. Naive Bayes

Щодо результативності, то потрібно обережно відноситися до результатів, адже зображення двох груп помітно різняться між собою і поки що складно говорити про ефективність алгоритму у реальному житті, адже ми матимемо задачу багатокласової класифікації, і у такому випадку тестувати нашу модель потрібно буде ще раз, причому немає гарантій, що результати будуть не гірші.

Література

[1] Обучение на размеченных данных. Линейные модели. МФТИ
Coursera 2021

[2] Обучение на размеченных данных. Проблема переобучения и борьба с ней. МФТИ
Coursera 2021

[3] Обучение на размеченных данных.Метрики качества. МФТИ
Coursera 2021

[4] Вступ ло машинного навчання. КУНШТ
http://specials.kunsht.com.ua/machinelearning2

[5] Gaussian Mixture Models Explained
https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95

[6] Алгоритм кластеризации гауссовых моделей смесей
https://www.machinelearningmastery.ru/gaussian-mixture-models-d13a5e915c8e/

[7] RandomForestClassifier
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[8] LogisticRegression
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[9] KNeighborsClassifier
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[10] Decision Trees
https://scikit-learn.org/stable/modules/tree.html

[11] Naive Bayes
https://scikit-learn.org/stable/modules/naive_bayes.html

[12] MODIFIED TEXTURE METHOD OF IMAGE SEGMENTATION BASED ON THE GRAY-LEVEL CO-OCCURRENCE MATRIX
Kateryna Golubeva, Dmitry Klyushin 2018

[13] Обучение на размеченных данных. Cross-validation. МФТИ
Coursera 2021