Реферат

Тема работы: Использование гибридных схем машинного обучения для выделения области потенциальной опухоли в задаче диагностики рака молочной железы.

Объем работы составляет 35 страниц, список использованной литературы включает 14 источников, в работе приводится 12 таблиц и 4 рисунка.

Ключевые слова: машинное обучение, рак молочной железы, анализ данных, бинарная классификация.

Работа посвящена освоению метода определения наличия рака молочной железы у пациента на основе термографических снимков. Для каждой теплокарты в черно-белом формате вычисляется вектор значений некоторых характеристик изображения. Полученные вектора объединяются в табличную выборку, которая используется для решения задачи бинарной классификации. В ходе работы рассматриваются различные методы отбора признаков, такие как: статистические методы, t-тест Стьюдента, метод главных компонент, Permutation Importance, отбор признаков на основе их значимости в обученной модели и жадный алгоритм. В качестве моделей машинного обучения рассматриваются: логистическая регрессия, метод опорных векторов, случайный лес, градиентный бустинг. Производится отбор гиперпараметров моделей. Прилагаются сравнительные результаты метрик качества на независимых выборках. Основные результаты и выводы сопровождаются таблицами и графиками.

Содержание

Введение

По статистике, рак молочной железы (РМЖ) – наиболее частая форма онкологического заболевания среди женщин (около 23%). Заболеваемость РМЖ растет не только в России, но и во всем мире. На сегодняшний день существует ряд основных подходов к диагностированию РМЖ: термография, УЗИ, МРТ, томосинтез, маммография и другие. Термография – подход, основанный на регистрировании изменения температуры поверхности тела. Инфракрасное излучение, то есть тепло тела, улавливается тепловизионной камерой и преобразуется в соответствующие значения температуры. Тепловизионная камера отображает распределение температур в виде изображения, называемого термограммой [1]. Подход является эффективным, так как известно, что раковые клетки выделяют тепло [2].

Использование термограмм молочных желез – достаточно быстрый, безболезненный, недорогой и безопасный метод визуализации. Кроме того, данный подход можно использовать для обработки изображений и получения информации в режиме реального времени. Согласно некоторым исследованиям, термография, при определении четких протоколов, способна выявлять ранние симптомы РМЖ на 8-10 лет раньше, чем маммография [3]. С развитием области машинного обучения и задач компьютерного зрения, повысилась роль использования теплокарт, как эффективного начального инструмента при построении систем выявления аномалий на раннем этапе онкологии.

Цель данной работы – построить алгоритмическую систему, способную на основе термографических снимков определять наличие РМЖ у пациентов. Система должна содержать в себе выбранный заранее алгоритм машинного обучения, решающий задачу бинарной классификации «нет патологии – есть патология». Обучение модели необходимо проводить на выборке, полученной с помощью преобра-

зования каждого изображения в некоторый вещественный вектор фиксированного размера N (набор значений некоторых статистик над изображением). Таким образом, конкретной базе данных термоснимков (M снимков) будет соответствовать единая табличная выборка (M наблюдений, N признаков). Алгоритмическая система должна отвечать следующим требованиям:

1. Приемлемая точность;
2. Обобщенность;
3. Интерпретируемость.

П.1 означает, что прежде всего нам необходимо построить систему, с минимально возможным процентом ошибок при выставлении диагнозов. В п.2 имеется ввиду то, что построенная система должна удовлетворять точностью прогноза не только на данных, используемых при разработке, но и на новых снимках, которые будут прогоняться через систему в дальнейшем. Под п.3 понимается возможность наглядно продемонстрировать, какие именно факторы при построении прогноза указывают на наличие патологий. Построенная система должна быть способна "аргументировать" свой ответ так же, как и живой медицинский специалист.

Перед нами стоит задача построения комплексного алгоритма на языке Python с использованием таких библиотек, как numpy, open-cv, scikit-learn, hyperopt и т.д. Мы имеем последовательные фазы решений сформулированной задачи:

1. Построить алгоритмическую систему, принимающую на вход теплокарту молочной железы в виде черно-белого снимка формата PNG и выдающую результатом вектор значений разнообразных статистик и показателей, вычисленных на переданном изображении;

2. Использовать полученные выборки численных значений для применения различных подходов анализа данных и обучения моделей машинного обучения – решается задача бинарной классификации;

3. Сравнить полученные результаты и выбрать наилучший метод построения прогноза наличия РМЖ по термоснимку.

В данной работе предлагаются различные методы отбора и предобработки признаков, а также сравнительный анализ метрик качества алгоритмов машинного обучения.

1. База данных термоснимков. Вычисление признаков над изображениями

В работе представлены результаты применения гибридных схем машинного обучения на открытой базе данных (БД) проекта Visual Lab (DMR – Database For Mastology Research), основанного на базе Института Вычислений бразильского Федерального университета Флуминенсе [4]. Это онлайн-платформа, на которой хранится информация о свыше чем 400 пациентов с диагнозом РМЖ и без него. Термографические изображения были получены с соблюдением четкого протокола действий. Для каждого изображения были выделены области интереса, ограничивающие изображения лишь той частью, в которой содержатся молочные железы: по две области для здорового пациента и одна область с РМЖ для больного [5].

Открытая БД содержит термоснимки 51 пациента: 32 с патологией и 19 без каких-либо отклонений: всего это 1273 теплокарты молочных желез по-отдельности (с патологией 511 и 762 без патологии). Из всех пациентов, образующих базу данных, снимки 9-ых из них не участвовали в построении алгоритмических систем проекта Visual Lab и были отобраны специалистами как "контрольные" пациенты, а их теплокарты – "контрольными" снимками.

Теплокарты открытой БД проекта Visual Lab являются черно-белыми изображениями единого размера, с нулевыми пикселями вне областей интереса, где участкам тела с более высокой температурой соответствуют большие значения в матрице. Как уже было отмечено, места раковой опухоли имеют повышенную температуру, а следовательно такие аномальные области могут быть областью

РМЖ. Все, что нам остается, это с помощью какого-либо подхода или алгоритма выделить эту "горячую" область (см Рисунок 1.1).

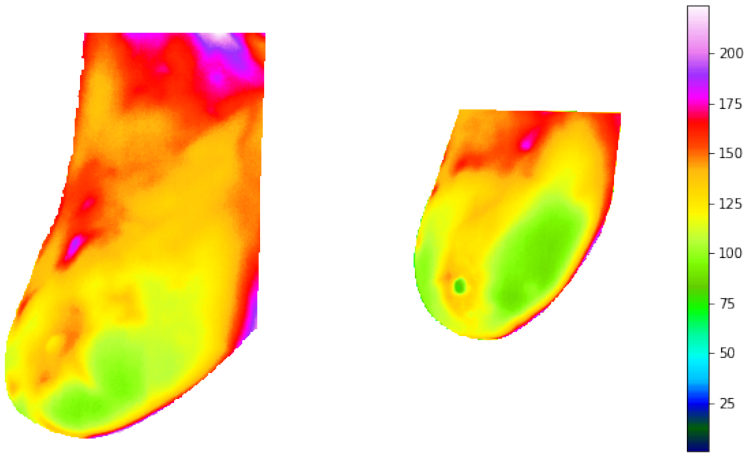


Рис. 1.1: Изображение двух молочных желез с патологией (слева) и без патологии(справа).

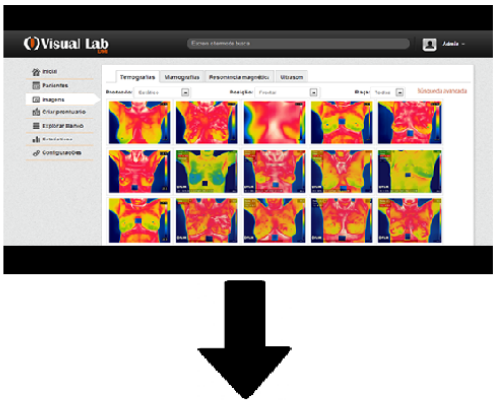
В данной работе представлен следующий подход (см. Рисунок 1.2). Мы самостоятельно выбираем для себя набор признаков, вычисление каждого из которых реализуется в виде отдельной функции. При прогоне матричных изображений через функцию, получается некоторый набор значений – дискретная случайная величина, т.ч. выполняются некоторые условия. Пусть имеется: некоторая функция f – функция вычисления признака над изображением, набор (I_i, y_i),$k = 1,...,M$, где I_i – матрица изображения, M – количество изображений. y_i выражается, как

$$y_i = \begin{cases} 1, & I_i \text{ содержит патологию,} \\ 0, & I_i \text{ не содержит патологии.} \end{cases}$$

Тогда для $V(i) = f(I_i)$ выполняется следующее:

$$\begin{cases} V(i) \in C(0), & \text{при } y_i = 0, \\ V(i) \in C(1), & \text{при } y_i = 1. \end{cases}$$

Здесь $C(0)$, $C(1)$ – некоторые распределения двух типов случайных величин из набора V. Более информативный признак влечет за собой большее различие этих двух распределений. В худшем случае $C(0) = C(1)$ – это означает, что конкретная функция $f(I)$ не способна выделять различия между здоровыми молочными железами и железами с диагнозом РМЖ.



	id	pathology	f_1_1_10__v1	f_1_1_1__v1	f_1_1_2__v1	f_1_1_3__v1	f_1_1_4__v1	f_1_1_5__v1	f_1_1_6__v1	f_1_1_7__v1
PAC_00_DN0_RAW_right	PAC_00	1	153.0	128.465223	19.200861	368.673072	130.0	0.087330	2.888025	4.314263
PAC_00_DN10_RAW_right	PAC_00	1	173.0	145.996855	20.726065	429.569786	147.0	0.380555	3.332089	4.384321
PAC_00_DN11_RAW_right	PAC_00	1	171.0	144.972555	19.626635	385.204788	146.0	0.251850	3.046579	4.335745
PAC_00_DN12_RAW_right	PAC_00	1	173.0	146.945692	20.425921	417.218249	148.0	0.314693	3.157280	4.368385
PAC_00_DN13_RAW_right	PAC_00	1	176.0	148.365707	20.948782	438.851456	149.0	0.381951	3.329843	4.393207
...	...	...	...	...	...	...	...	...	...	...
PAC_69_DN7_RAW_right	PAC_69	0	206.0	177.789264	21.475864	461.212743	177.0	0.034631	2.181427	4.392046
PAC_69_DN8_RAW_left	PAC_69	0	202.0	176.693163	17.467435	305.111268	174.0	0.239023	3.270608	4.195202
PAC_69_DN8_RAW_right	PAC_69	0	205.0	176.456169	20.941521	438.547308	176.0	0.029563	2.280503	4.373018
PAC_69_DN9_RAW_left	PAC_69	0	201.0	176.080986	17.228176	296.810059	173.0	0.470357	3.587042	4.184259
PAC_69_DN9_RAW_right	PAC_69	0	207.0	177.719607	21.439980	459.672762	177.0	0.078219	2.214795	4.396563

Рис. 1.2: База данных VisualLab преобразуется в табличную выборку.

Простыми примерами вышеобозначенной функции f являются разнообразные статистики: среднее, дисперсия, максимальная яркость изображения (см. Таблица 1.1).

Таблица 1.1: Примеры признаков над изображением $I(l,m)$ (H – высота изображения, W – ширина изображения в пикселях).

Формула	Описание
$\mu(I) = \frac{1}{H+W} \sum_{l=1}^{H} \sum_{m=1}^{W} I(l,m)$	Среднее набора яркостей изображения
$\sigma(I) = \frac{1}{H+W} \sum_{l=1}^{H} \sum_{m=1}^{W} (I(l,m) - \mu(I))^2$	Дисперсия набора яркостей изображения
$M(x) = \max_{l,m} I(l,m)$	Максимальное значение набора яркостей изображения

Всего алгоритмическая система вычисления признаков содержит 74 уникальных функций f. Некоторые из них содержат некоторые параметры вычисления (гиперпараметры), в зависимости от которых меняется и распределение значений $C(0)$ и $C(1)$, поэтому конечная размерность признакового пространства будет гораздо больше. Полное описание списка признаков представлено в онлайн-репозитории, где также содержится код работы решения задачи бинарной классификации на языке программирования Python [6].

2. Предобработка табличных данных

2.1. Разведочный анализ данных. Первичный отбор признаков

После применения алгоритма системы вычисления признаков к каждому изображению, мы получаем табличную выборку вещественных значений размера 1273 на 550, где 1273 - количество молочных желез, 550 - количество признаков. Каждому наблюдению соответствует метка целевой переменной y, принимающая значения 0 и 1 – отсутствие или наличие патологии соответственно. Также, каждому снимку молочной железы соотносится свой идентификационный номер пациента (ID). Как уже было отмечено ранее, база данных Visual Lab содержит "контрольную" группу пациентов $X^{control}$. В нашей табличной выборке это 197 наблюдений. Следовательно в нашем распоряжении имеется полная таблица X размера 1076 на 550 – выборка для разработки.

Перед обучением моделей машинного обучения необходимо проанализировать набор признаков на наличие в нем "шумовых" – то есть тех, которые являются неинформативными в имеющемся признаковом пространстве. Также, нельзя упускать тот факт, что 550 признаков для ряда алгоритмов машинного обучения – слишком большой набор. Поэтому первой нашей задачей будет провести первичный анализ данных для "грубого" удаления подмножества признаков из имеющегося множества.

Пусть N – количество имеющихся векторов-признаков V_j ($\bigcup_{j=1}^{N} V_j = X$; $V_j \in \mathbb{R}^M$), V_j^{norm} – нормализованный вектор следующим образом:

$$V_j^{norm} = \frac{V_j - \min\limits_j V_j}{\max\limits_j V_j - \min\limits_j V_j}.$$

Первичный анализ данных заключается в анализе статистических характеристик массива, поиска некоторых закономерностей эмпирическим путем. Предлагаемый в данной работе метод состоит из трех этапов:

1. Отбор признаков по дисперсии;
2. Отбор признаков по количеству выбросов;
3. Отбор признаков по парной корреляции.

Дисперсионный критерий необходим для избавления от константных или околоконстантных признаков. Серьезные алгоритмы, такие как градиентный бустинг или случайный лес справляются с влиянием таких факторов, однако линейные модели (логистическая регрессия) при сильном дисбалансе одного из классов будут ошибочно передавать больший весовой коэффициент такому признаку. Для каждого вектора вычисляется стандартное отклонение $s_j = \sigma(V_j^{norm})$.

В итоге формируется некоторый набор значений стандартных отклонений для каждого признака. Далее, воспользуемся так называемым "правилом трех сигм" (также изестное как "правило 68–95–99,7"), утверждающим, что вероятность того, что случайная величина отклонится от своего математического ожидания более чем на три среднеквадратических отклонения, практически равна нулю. Мы же будем рассматривать отклонение два среднеквадратичных отклонения. Данный факт справедлив только для случайных величин, распределенных по нормальному закону, однако на практике редко когда предстает возможность работать с такими идеальными распределениями, поэтому мы воспользуемся именно идеей: вычеркнем из дальнейшего рассмотрения те признаки, стандартное отклонение

которых будет превышать значение V_j^α – квантиль уровня α (в нашем случае $\alpha = 0.9544$). При работе с дискретными конечными наборами значений V_j^α вычисляется как некоторое число, при котором доля всех остальных значений меньших V_j^α не превышает α.

Критерий отбора по количеству выбросов имеет ту же идею, однако теперь вместо вычисления набора значений вычисляется o_j как сумма функций-индикаторов

$$o_j = \sum_{i=1}^{M} [V_j(i) \notin [A_j, B_j]].$$

Здесь A_j, B_j - края статистически значимой выборки для каждого признака. Данные значения вычисляются так:

$$\begin{cases} A_j = V_j^{0.25} - 1.5 \cdot (V_j^{0.75} - V_j^{0.25}), \\ B_j = V_j^{0.75} + 1.5 \cdot (V_j^{0.75} - V_j^{0.25}). \end{cases}$$

То есть, o_j – это количество наблюдений, являющихся статистическими выбросами для признака V_j. Выбросами в анализе данных являются некоторые аномалии, статистически несоответствующие некоторому общему закону распределения всех значений. В данном случае мы считаем слишком малые, либо слишком большие значения по каждому наблюдению. По той же логике, что и со стандартным отклонением, избавимся от части признаков по "правилу двух сигм".

Корреляционный критерий основывается на вычислении корреляционной матрицы C размера M на M, где

$$C(i, j) = r(V_i, V_j).$$

Причем $C(i, j) = C(j,i)$. Здесь $r(V_i, V_j)$ - коэффициент корреляции Пирсона.

Из одной пары признаков, корреляция которых превышает некоторое пороговое значение, случайным образом оставляем только один из них. Приведена сравнительная таблица (см. Таблица 2.1), сколько удаляется признаков в зависимости от порога.

Таблица 2.1: Количество удаленных признаков по пороговому значению коэффициента корреляции Пирсона.

Абсолютное значение порога	Количество удаленных признаков
0.99	293
0.95	352
0.9	381
0.8	419

Такое большое количество удалений связано с тем, что большая часть набора признаков – это различные реализации одной и той же функции вычисления факторов с разным набором начальных параметров. Результатом применения первичного отбора признаков становится сокращение признакового пространства с 550 до 180 (см Таблица 2.2).

Таблица 2.2: Последовательное изменение размера выборки для разработки после применения порогового отбора признаков.

Метод отбора	Значение порога	Размер выборки
До предобработки	-	(1076, 550)
Порог по значению стандартного отклонения	0.0003454	(1076, 524)
Порог по количеству выбросов	113	(1076, 500)
Порог по абсолютному значению корреляционного коэффициента	0.95	(1076, 180)

2.2. Основной отбор признаков

Благодаря первичному отбору признаков размер признакового пространства был сокращен более чем в два раза. Идея уменьшения количества признаков следует из двух идей: сокращение времени обучения модели машинного обучения и снижение риска ухудшения качества модели на новых данных. На самом деле, на современных ЭВМ время обучения модели начинает сильно влиять на производительность при количестве наблюдений свыше нескольких сотен тысяч и если это действительно сложные модели (ансамбли базовых моделей). Однако, достаточно весомой является идея с ухудшением качества модели: чем больше признаков участвует в построении прогноза модели, тем больше риск того, что один из них окажется неинформативным при построении прогноза в будущем.

В данном параграфе будет рассмотрено несколько вариантов отбора признаков: t-критерий Стьюдента, метод главных компонент, Permutation Importance,

весовые коэффициенты регрессионных моделей и случайного леса, а также жадный алгоритм отбора.

2.2.1. t-критерий Стьюдента

Двухвыборочный t-критерий для независимых выборок – статистический тест, применяемый к выборкам с нормальным распределением [7] (в нашем же случае это требование опускается). Для вектора-признака V_j рассмотрим две его подвыборки $V_j^1 \in C_j(1)$, $V_j^0 \in C_j(0)$, где $C_j(1)$, $C_j(0)$ – распределения случайных величин из набора V_j, относящихся к позитивному и негативному классу соответственно. Необходимо проверить по выборочным данным нулевую гипотезу равенства математических ожиданий этих случайных величин $H_0 : \bar{V}_j^1 = \bar{V}_j^0$. Для каждого V_j вычисляется статистическая значимость критерия (p-value), и удаляются те признаки, p-value которых больше некоторого порога (то есть не принимается гипотеза о равенстве средних двух классов).

Базовая идея отбора по критерию следующая: так как области патологий содержат более горячие участки температур, то при вычислении значения некоторого признака, большие значения матрицы изображения будут вносить большее отклонение в среднее значение этого фактора. В этом случае мы учитываем признаки, которые будут линейно разделять два класса в пространстве признаков с некоторой долей ошибки, поэтому данный вариант отбора считается достаточно простым.

Таблица 2.3: Сравнительное изменение размера выборки для разработки после применения t-критерия Стьюдента с разными пороговыми значениями p-value.

Значение порога p-value	Размер выборки
-	(1076, 180)
0.01	(1076, 141)
0.05	(1076, 130)
0.1	(1076, 121)

Результаты отбора представлены в Таблице 2.3. Как мы можем заметить, достаточно большое количество признаков было оставлено после отбора статистическим критерием, даже с p-value равным 10%. Это говорит о том, что большая часть признаков, разделяясь на два класса, распределяется вокруг различных средних значений.

2.2.2. Метод главных компонент

Метод главных компонент (PCA) – один из основных способов уменьшить размерность данных, потеряв наименьшее количество информации [8]. С математической точки зрения PCA является ортогональным линейным преобразованием, сжимающее признаковое пространство. При этом первая компонента новой системы координат (ось) строится таким образом, чтобы дисперсия данных вдоль неё была бы максимальна. В свою очередь, дисперсия, являющаяся мерой изменчивости данных, может отражать уровень их информативности. Поэтому для набора, состоящего из n новых компонент ($n < M$), можно вычислить остаточную информативность нового пространства, просуммировав значения относительных

дисперсий, и добавлять по одной компоненте до тех пор, пока информация не превысит некоторый порог.

Таблица 2.4: Сравнительное изменение размера выборки для разработки после применения метода главных компонент с разными пороговыми значениями информативности "сжатого" пространства признаков.

Значение порога информативности	Размер "сжатой" выборки
-	(1076, 180)
0.95	(1076, 66)
0.90	(1076, 53)
0.85	(1076, 44)

Результаты применения метода главных компонент представлены в Таблице 2.4. Следует отметить, что подход к уменьшению числа компонент не всегда даёт хорошие результаты. Это связано с тем, что часть дисперсии данных может быть обусловлена шумами, а не информативностью компонент.

2.2.3. Permutation Importance

Подход к вычислению Permutation Importance (PI) основан на вычислении значимости признака при построении прогноза некоторого алгоритма машинного обучения на независимой выборке. Эта значимость определяется как среднее снижение определенной метрики бинарной классификации, при случайной перестановке значений одного признака.

Пусть у нас имеется некоторое разбиение выборки X на X^{train} и X^{test}, т.ч. $X^{train} \cup X^{test} = X$, $X^{train} \cap X^{test} = \emptyset$. Аналогичное разбиение для целевого вектора y. На данных (X^{train}, y^{train}) обучим некоторую модель машинного обучения

$F(X^{train}, y^{train}) \to F_{fit}(X)$ и посчитаем метрику качества $S = S(y^{test}, y^{predict}) = S(y^{test}, F(X^{test}))$ на тестовой выборке.

Для каждого признака V_j ($j = 1,...,N$) посчитаем среднее снижение метрики относительно базового значения:

$$s_j = \{s_{i,j}\}_{i=1}^{K},$$

$$p_j = S - \mu(s_j).$$

Здесь K – количество случайных перестановок набора значений вектора V_j, $s_{i,j}$ – значение метрики $S(y^{test}, F(X_j^{test})$ при изменении j-го столбца матрицы X^{test} описанным выше способом, $\mu(x)$ – значение выборочного среднего.

Пороговым значением для PI будет нуль – то есть нас интересуют только те признаки, которые являются хоть сколько-нибудь значимыми для обученной модели. В качестве алгоритма F рассмотрим дерево решений, в качестве метрики S – ROC-AUC-меру (площадь под ROC-кривой). Результаты представлены в Таблице 2.5.

Таблица 2.5: Изменение размера выборки для разработки после применения отбора признаков по пороговому значению Permutation Importance.

Значение порога информативности	Размер выборки
-	(1076, 180)
0	(1076, 4)

Отбор по PI демонстрирует, что из большого набора признаков лишь малая часть действительно влияет на метрику качества модели.

2.2.4. Коэффициенты "значимости" обученных моделей

Прошлый метод отражал идею влияния одного признака на метрику качества при построении прогноза модели. Однако при таком подходе не совсем понятно, как именно фактор влияет на модель при перестановке значений. В данном случае алгоритм выдает результат и на его основе строится дальнейшее решение. Исправить такую неопределенность поможет следующий вариант.

Рассмотрим выборку для обучения X^{norm}, состоящую из нормированных столбцов-признаков V_j^{norm}. Обучим на ней модель логистической регрессии, применяемую для прогнозирования вероятности возникновения некоторого события. Логистическая регрессия – итерационный метод, основанный на вычислении весовых коэффициентов вектора w путем минимизации функционала ошибки градиентным спуском. При нормированном распределении значений каждого признака V_j вектор w также является вектором "значимости" признаков для наступления целевого события. Стоит отметить, что нас не сильно интересует метрика качества: здесь важен конечный результат оптимизации коэффициентов линейной модели. После получения вектора коэффициентов w по некоторому пороговому значению оставляются те признаки, чья "значимость" выше порогового значения коэффициентов (см Таблица 2.6).

Таблица 2.6: Сравнительное изменение размера выборки для разработки после применения отбора по пороговому значению коэффициентов логистической регрессии.

Пороговое значение	Размер выборки
-	(1076, 180)
0	(1076, 87)
0.5	(1076, 25)
1	(1076, 4)

Аналогичный, но более сложный подход рассматривается при использовании модели случайного леса (Random Forest) – ансамбля решающих деревьев. Метод основывается на вычислении коэффициента "значимости" w_j признака V_j (Random Forest Feature Importance) обученной модели случайного леса [10].

Создатели пакета BorutaPy предложили метод, основанный на последовательном обучении независимых ансамблевых моделей с одинаковыми гиперпараметрами и последовательном удалении из выборки "наихудших" и "наилучших" признаков относительно коэффициентов значимости w_j [11]. Также, на каждой итерации рассматривается случайное подмножество наблюдений, что позволяет уменьшить влияние выбросов и корреляции между признаками. Все это представляет из себя достаточно устойчивый алгоритм отбора признаков, который останавливается при достижении определенного номера итерации, либо пока все множество признаков не разделится на два подмножества (см Таблица 2.7), одно из которых мы и оставим.

Таблица 2.7: Последовательное изменение размера выборки для разработки после применения отбора признаков методом Boruta.

Число итераций	Размер выборки
-	(1076, 180)
50	(1076, 130)
100	(1076, 124)

2.2.5. Жадный отбор признаков

Последним методом отбора признаков является принцип жадного отбора на основе максимизации метрики качества. Для набора используется определенная конфигурация модели машинного обучения. Жадный отбор – итеративный метод, где на каждой итерации добавляется по одному признаку из списка длины $N - n$ в список факторов длины n и переобучении фиксированной модели на $n + 1$ признаке. Среди всех $N - n$ атрибутов выбирается наилучший с точки зрения качества классификации переобученной модели. Алгоритм отбора работает до тех пор, пока метрика качества на независимой выборке не стабилизируется.

После стабилизации полагается, что метрика не будет иметь импульсивного роста, так как жадным алгоритмом мы уже отобрали все значимые для модели признаки (см. Алгоритм 2.2.5).

Algorithm 1 Жадный отбор признаков

Result: v – индексы отобранных признаков (номера столбцов матрицы X)

Initialization:

1. c_{iter} – максимально допустимое количество итераций i, на которых метрика не улучшилась по сравнению с итерацией $i-1$
2. X_i^{train} $(i = 1,...,m)$: $\bigcup_{i=1}^{k} X_i^{train} = X^{train}$
3. $F(\gamma, X, y)$ - алгоритм машинного обучения с фиксированными гиперпараметрами, $S = 0$ – метрика качества.

$G := 0$; $R := \{1,...,N\}$, $c = 0$, $v = \emptyset$

while $c \neq c_{iter}$ **do**

  $max := 0$; $j^* := 0$

  for j *in* R **do**

    $v^* := v \cup j$

    for i *in* $\{1,...,k\}$ **do**

      $X^* := \bigcup_{\substack{j=1 \\ j \neq i}}^{k} X_j^{train}$; $y^* := \bigcup_{\substack{j=1 \\ j \neq i}}^{k} y_j^{train}$

      $F^i(\gamma, X^*[v^*], y^*) \rightarrow F_{fit}^i(X[v^*])$

      $S^i := S(y_i^{train}, F_{fit}^i(X_i^{train}[v^*]))$

    if $\frac{1}{k}\sum_{i=1}^{k} S^i > max$ **then**

      $max := S(y^{test}, F_{fit}(X^{test}[v^*]))$

      $j^* := j$

  $v := v \cup j^*$; $R := R \setminus j^*$

  if $G > max$ **then**

    $c := c + 1$

  else

    $G := max$

Таблица 2.8: Изменение размера выборки для разработки после применения жадного отбора признаков.

c_{iter}	Размер выборки
-	(1076, 180)
3	(1076, 7)

Данный алгоритм рассматривает среднее кросс-валидационное качество метрики (k-fold CV). При таком подходе отбор признаков и сами значения при добавлении факторов будут меньше зависеть от переобучения. В следующей главе эта проблема будет рассмотрена подробнее. После применения жадного алгоритма количество отобранных признаков получается сравнимым с отбором по перестановкам (см Таблица 2.8).

3. Обучение моделей. Оптимизация гиперпараметров

Получив некоторый короткий список факторов, мы переходим к следующему этапу – этапу обучения алгоритма машинного обучения F и оптимизации его набора гиперпараметров $\gamma = \{\gamma_1,...,\gamma_t\}$ – параметров контроля процесса обучения. В зависимости от сложности модели растет и зависимость от этих значений. Так, например, в модели логистической регрессии оптимизируется 2 параметра: коэффициент регуляризации и тип регуляризации, в то время, как у случайного леса их уже 5.

В данной работе был использован TPE алгоритм оптимизации (Tree Parzen Estimators), реализованный в пакете HyperOpt языка программирования Python [12]. Итерационный процесс необходимо проводить на основе максимизации некоторой метрики качества S на тестовой выборке при фиксированном разбиении выборки X на X^{train} и X^{test}.

$$\gamma^* = arg\max_{\gamma}\{S(y^{test}, F_{fit}(\gamma, X^{test}))\}.$$

Здесь $F_{fit}(\gamma, X^{test})$ – прогноз на X^{test} обученной на X^{train} модели с гиперпараметрами γ.

В нашем распоряжении всего 51 пациент, снимки каждого из которых довольно схожи друг с другом. Разбиение необходимо проводить таким образом, что

$$(ID(i) \in ID(X^{train}) \wedge ID(i) \notin ID(X^{test}) \vee$$
$$(ID(i) \notin ID(X^{train}) \wedge ID(i) \in ID(X^{test}),$$

где $ID(i)$ – идентификатор пациента, относящийся к наблюдению i ($i = 1,..,M$), $ID(X^{train})$ – множество идентификаторов из выборки снимков для обучения, $ID(X^{test})$ – множество идентификаторов из выборки снимков для тестирования.

Так как каждый пациент содержит несколько десятков снимков, с таким разделением модель будет переобучена в сторону обучающей выборки, а метрика качества на тестовой выборке будет сильно зависеть от разбиения. На зависимость от разбиения еще достаточно сильно влияет малый размер выборки. Для наглядной демонстрации данного факта можно взглянуть на сглаженный график распределения метрики качества деревьев решений, обученных на разных подвыборках (см. Рисунок 3.1).

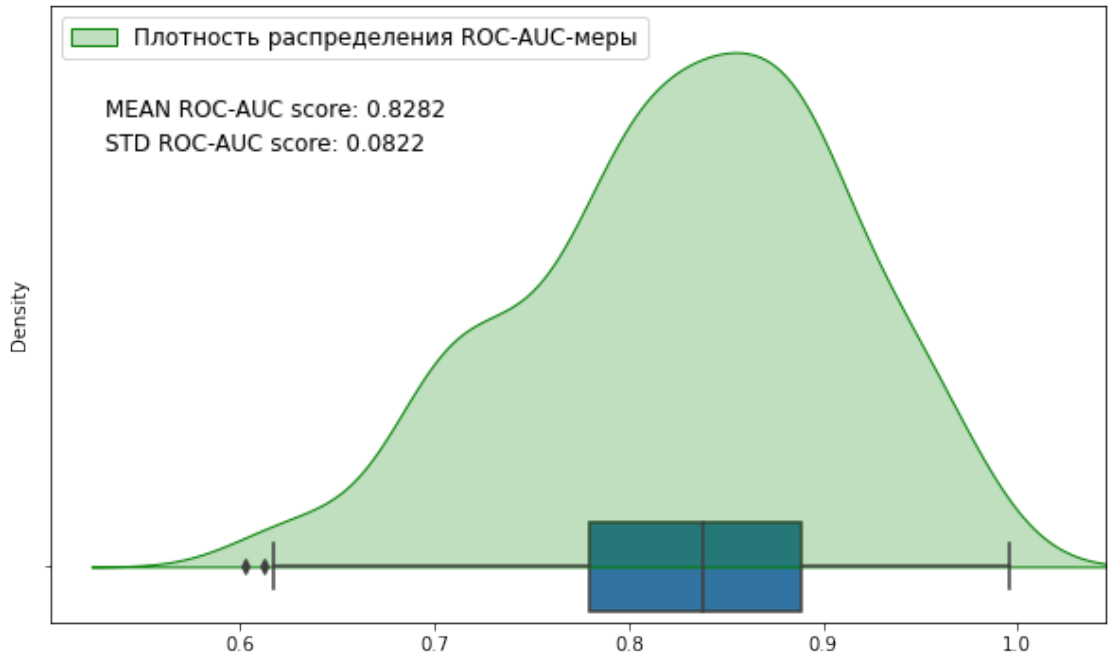


Рис. 3.1: Распределение значений ROC-AUC score деревьев решений, обученных и оцененных на разных подвыборках выборки X.

В связи с этим, довольно очевидно, что максимизация метрики качества на одной тестовой выборке влечет за собой переобучение. Поэтому на каждой итера-

ции оптимизации параметров мы будем максимизировать среднее по разбиению k-fold CV: этим мы избавимся от переобучения на конкретных наблюдениях, и оптимизация будет проводиться в среднем.

$$\gamma^* = arg\max_{\gamma}\{\frac{1}{k}\sum_{i=1}^{k} S(y_i^{train}, F_{fit}(\gamma, X_i^{train}))\},$$

$$\bigcup_{i=1}^{k} X_i^{train} = X^{train}.$$

После подбора гиперпараметров обучается модель с оптимальным набором гиперпараметров γ^* на полной выборке X^{train} и вычисляем контрольное значение метрики на независимой выборке X^{test}.

4. Сравнительный анализ качества классификации различных моделей

Было отмечено, что выборка недостаточно велика для оценки качества работы алгоритмов лишь на одном разбиении (X^{train}, X^{test}). Поэтому в данной работе все вышеописанные действия и процедуры будут применены к кросс-валиационному разбиению выборки X на k фолдов: начиная от первичного отбора данных, заканчивая обучением модели. Значение метрики S на X_i^{train}, будет являться одной из k компонент оценки результата применения пошаговых действий ($\bigcup_{i=1}^{k} X_i^{train} = X$). Получив k таких значений, высчитывается среднее – данная величина и есть результат работы системы. Для избавления от аномальных прогонов и зависимости от кросс-валидационного разбиения, вся эта последовательность действий для каждого варианта отбора признаков и алгоритма машинного обучения будет повторена K раз.

Формально это можно записать как сложную функцию

$$S = \frac{1}{K}\sum_{j=1}^{K}\frac{1}{k}\sum_{i=1}^{k} S(y_i^{train}, F_{fit}^{i}(\gamma^*, P_2(P_1(X_i^{train})))),$$

где F_{fit}^{i} – обученная на выборке $\bigcup_{\substack{j=1 \\ j\neq i}}^{k} X_j^{train}$ модель, γ^* – оптимизированный набор гиперпараметров на той же выборке, P_1, P_2 – функции применения соответственно первичного и основного этапа отбора признаков.

После отбора признаков, подбора гиперпараметров и обучения модели на $\bigcup_{\substack{j=1 \\ j\neq i}}^{k} X_j^{train}$ также делается прогноз на выборке $X^{control}$ – 197 снимков 9-ых пациентов, отобранных проектом VisualLab как контрольная группа для проверки каче-

ства классификаторов (120 наблюдений отрицательного класса и 77 класса "наличие РМЖ"). После получения k значений $S = S(y^{control}, F^i_{fit}(\gamma^*, X^{control}))$ также рассматривается среднее значение. Процесс оптимизации и усреднения метрик качества представлен на Рисунке 4.1.

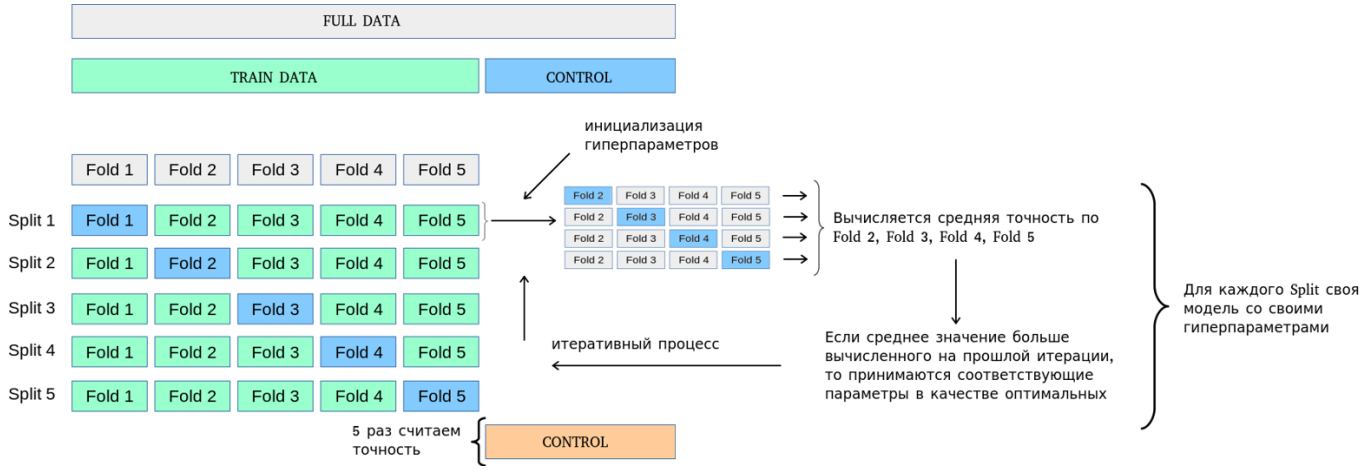


Рис. 4.1: Схема вычисления k-fold CV average score с оптимизацией гиперпараметров. Данная схема запускается K раз. *В данной схеме отбор признаков не отображен.*

5. Результаты

Итогом работы стала реализация комплексной системы оптимизации гиперпараметров и функций отбора признаков на яызке программирования Python с возможностью изменения начальной конфигурации: тип модели машинного обучения, выборка X для разработки и $X^{control}$ – независимая контрольная выборка, параметр k – количество разбиений выборки X методом k-fold CV, пороговые значения на улучшения средней метрики качества, возможность балансировки двух классов и другие параметры. Весь процесс построения гибридной схемы состоит в последовательном запуске этапов в различных вариациях:

1. Разбиение выборки для разработки на k фолдов;
2. Первичный отбор признаков;
3. Основной отбор признаков;
4. Оптимизация гиперпараметров модели;
5. Определение качества модели.

При запуске прогона схем было рассмотрено 4 модели машинного обучения: логистическая регрессия (Log Reg, см. Таблица 5.1), метод опорных векторов с ядром радиально-базисной функции (SVM, см. Таблица 5.2), случайный лес (RF, см. Таблица 5.3) и градиентный бустинг библиотеки XGBoost [13] (см. Таблица 5.4). Основной метрикой качества S на всех этапах кроме последнего будет являться ROC-AUC-мера как наиболее эффективная при выборе метрики для оптимизации модели [14]. Дополнительно рассмотрим значения средней точности (mean accuracy) алгоритмов как наиболее доступный и легко интерпретируемый результат – процент точных прогнозов. Метрику accuracy недопустимо использовать при

сильном дисбалансе классов, но в нашей выборке X отношение числа Negatives к Positives 3 к 2, что практически не влияет на оценку качества алгоритма.

В таблицах результатов рассматриваются средние значения метрик качества на тестовых выборках при разбиении выборки X на k-fold CV ("TEST ROC-AUC" и "TEST accuracy") и на независимой контрольной выборке $X^{control}$ ("CONTROL ROC-AUC" и "CONTROL accuracy"). В скобках после названий методов отбора признаков указан средний размер итогового короткого списка факторов, на котором обучались модели. Всего было произведено $K = 10$ независимых кросс-валидационных разбиений выборки X на k фолдов, где $k = 5$.

metric	t-test (133)	pca (65)	permut (5)	lr coefs (24)	boruta (126)	greedy (8)
TEST accuracy	0.805 ± 0.017	0.9103 ± 0.008	0.8152 ± 0.013	0.8122 ± 0.077	0.7778 ± 0.016	0.8078 ± 0.01
TEST ROC-AUC	0.9044 ± 0.025	0.972 ± 0.006	0.8998 ± 0.032	0.9437 ± 0.015	0.885 ± 0.016	0.9143 ± 0.017
CONTROL accuracy	0.5082	0.8175	0.6054	0.6369	0.5265	0.7066
CONTROL ROC-AUC	0.7017	0.9607	0.704	0.788	0.6922	0.7977

Таблица 5.1: Средняя оценка качества результатов прогноза модели Log Reg.

metric	t-test (133)	pca (65)	permut (5)	lr coefs (24)	boruta (126)	greedy (8)
TEST accuracy	0.9084 ± 0.017	0.8914 ± 0.034	0.9276 ± 0.003	0.9166 ± 0.013	0.9082 ± 0.018	0.9026 ± 0.015
TEST ROC-AUC	0.9739 ± 0.011	0.9721 ± 0.01	0.9851 ± 0.007	0.981 ± 0.007	0.9727 ± 0.009	0.9767 ± 0.011
CONTROL accuracy	0.8438	0.841	**0.8904**	0.8701	0.8363	**0.8724**
CONTROL ROC-AUC	0.9742	0.966	**0.9819**	0.9743	0.9716	**0.9761**

Таблица 5.2: Средняя оценка качества результатов прогноза модели SVM.

metric	t-test (133)	pca (65)	permut (5)	lr coefs (24)	boruta (126)	greedy (8)
TEST accuracy	0.8905 ± 0.014	0.8809 ± 0.017	0.878 ± 0.016	0.8891 ± 0.018	0.8938 ± 0.016	0.889 ± 0.016
TEST ROC-AUC	0.9649 ± 0.007	0.9491 ± 0.013	0.9732 ± 0.011	0.9798 ± 0.008	0.9641 ± 0.009	0.9744 ± 0.009
CONTROL accuracy	0.7323	0.7062	0.753	0.758	0.733	0.7492
CONTROL ROC-AUC	0.9448	0.8825	0.9371	0.9547	0.9462	0.9394

Таблица 5.3: Средняя оценка качества результатов прогноза модели RF.

metric	t-test (133)	pca (65)	permut (5)	lr coefs (24)	boruta (126)	greedy (8)
TEST accuracy	0.8733 ± 0.023	0.8536 ± 0.023	0.858 ± 0.022	0.8626 ± 0.032	0.8708 ± 0.019	0.8677 ± 0.022
TEST ROC-AUC	0.9636 ± 0.02	0.9428 ± 0.01	0.9429 ± 0.017	0.9679 ± 0.007	0.958 ± 0.019	0.9691 ± 0.017
CONTROL accuracy	0.8191	0.7371	0.7283	0.7783	0.8325	0.7645
CONTROL ROC-AUC	0.9407	0.893	0.8893	0.92	0.9428	0.9133

Таблица 5.4: Средняя оценка качества результатов прогноза модели XGBoost.

Согласно средним значениям по метрикам метод опорных векторов дает наиболее стабильный и наименее ошибочный прогноз по сравнению с другими алгоритмами. Также, как и предполагалось, жадный алгоритм и отбор по перестановкам (PI) используют минимально возможное количество признаков и сравнимы по качеству (как на тестовых выборках, так и на контрольной) с остальными методами отбора, где размерность признакового подпространства гораздо выше. Получив процент точности 89.04%, нами были получены практически сопоставимые результаты с работой [1], в которой предложено наилучшее решение задачи бинарной классификации SVM-моделью с точностью 91%. Однако примечателен тот факт, что, используя методы отбора признаков, в упомянутой работе было отобрано 36 лучших, в то время, как в нашей работе предлагается в среднем около 5 признаков.

Заключение

В ходе работ была реализована комплексная алгоритмическая система по переводу изображений в пространство векторов с вещественными значениями и решению задачи бинарной классификации по определению наличия РМЖ на снимках. После перевода БД термоснимков в табличные данные, был проведен сравнительный анализ качества различных комбинаций методов отбора признаков и моделей машинного обучения. Исходя из средних результатов по метрикам качества, был достигнут наилучший средний процент точности классификации на последовательной схеме "Первичный отбор"+ "Отбор по Permutation Importance"+ "$SVM_{optimized}$": 92.76% на тестовых выборках в разбиении 5-fold CV и 89.04% на независимой контрольной выборке.

Дальнейшим развитием работы является использование методов предобработки термоснимков (применение фильтров над изображениями, нормализация) и применение представленных методов на других открытых и закрытых базах данных.

Список литературы

1. Sathish D., Kamath S., et al. (2019). Role of normalization of breast thermogram images and automatic classification of breast cancer. Vis Comput 35, 57–70. https://doi.org/10.1007/s00371-017-1447-9

2. Kakileti, Siva Teja Manjunath, et al. (2017). Advances in Breast Thermography. 10.5772/intechopen.69198.

3. Singh, Deepika Singh, et al. (2019). Role of Image Thermography in Early Breast Cancer Detection- Past, Present and Future. Computer Methods and Programs in Biomedicine. 183. 105074. 10.1016/j.cmpb.2019.105074.

4. Database For Mastology Research http://visual.ic.uff.br/dmi/

5. Marques R. S. Segmentação automática das mamas em imagens térmicas. Master Thesis, Instituto de Computação, Universidade Federal Fluminense, Niterói, RJ, Brasil, 2012.

6. Ващенко И. А. Онлайн-репозиторий с данными и кодом, используемыми в ВКР. https://github.com/answerqu/BreastCancer

7. Лемешко Б. Ю., Лемешко С.Б. Об устойчивости и мощности критериев проверки однородности средних. Измерительная техника, 2008, №9. – С.23-28

8. Gorban A. N., Kegl B., et al. Principal Manifolds for Data Visualisation and Dimension Reduction, Series: Lecture Notes in Computational Science and Engineering 58, Springer, Berlin — Heidelberg — New York, 2007, XXIV, 340 p. 82 illus. ISBN 978-3-540-73749-0

9. Ho, Tin Kam (1995). Random Decision Forests. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282

10. Lewinson E. (2019). Explaining Feature Importance by example of a Random Forest. https://towardsdatascience.com/explaining-feature-importance-by-example-of-a-random-forest-d9166011959e. https://github.com/erykml/medium$_a$*rticles/blob/master/Machine*

11. Kursa M., Rudnicki W. (2010). Feature Selection with the Boruta Package. Journal of Statistical Software, 36(11), 1 - 13. http://dx.doi.org/10.18637/jss.v036.i11

12. Bergstra J., Yamins D., Cox D. D. (2013) Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. To appear in Proc. of the 30th International Conference on Machine Learning (ICML 2013).

13. Chen, Tianqi, et al. (2016). XGBoost: A Scalable Tree Boosting System.

14. Bradley A. P. The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recognition, 30 (1997), pp. 1145-1159