Министерство образования и науки Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

Интеллектуальный анализ больших данных

**Решение задачи классификации**

Первушина Анна Алексеевна

Выполнила:

студентка группы № 932028

\_\_\_\_\_\_\_\_\_\_\_А.А. Первушина

Томск 2021

**Содержание**

[Введение. Постановка задачи 3](#_Toc62428201)

[Раздел 1. Предварительная обработка данных 5](#_Toc62428202)

[Раздел 2. Обучение моделей 8](#_Toc62428203)

[Заключение. Выводы, сравнение точности используемых моделей 18](#_Toc62428204)

Введение. Постановка задачи

Имеется набор данных по доброкачественной и злокачественной опухоли груди. Характеристики в наборе данных вычисляются из оцифрованного изображения. Они описывают характеристики ядер клеток, представленных на изображении.

Информация об атрибутах:

1. Идентификационный номер;
2. Диагноз (M – злокачественное новообразование, B – доброкачественное новообразование);
3. Для каждого ядра клетки вычисляются 10 характеристик с действительным знаком:
   1. Радиус (среднее расстояние от центра до точек по периметру);
   2. Текстура (стандартное отклонение значений шкалы серого);
   3. Периметр;
   4. Площадь;
   5. Гладкость (локальное измерение длин радиуса;
   6. Компактность;
   7. Вогнутость (степень вогнутости участков контура);
   8. Точки вогнутости (количество вогнутых участков контура);
   9. Симметричность;
   10. Фрактальна размерность;

Среднее значение, стандартная ошибка и «худшее» или наибольшее (среднее из трех наибольших значений) этих характеристик были вычислены для каждого изображения, в результате чего было получено 30 функций.

Все значения функций перекодированы с использованием четырех значащих цифр.

Распределение по классам: 357 доброкачественные новообразование, 212 злокачественные новообразования.

Нам необходимо предсказать, является опухоль доброкачественной или злокачественной, используя алгоритмы машинного обучения требуется решить задачу классификации. Перед этим, необходимо провести первичный анализ данных, выбрать значимые признаки, избавиться от «мусора», и т.д. После обучения моделей, необходимо провести сравнительный анализ на предмет эффективности используемых моделей.

Раздел 1. Предварительная обработка данных

Данные, на которых мы обучаем модель – это объекты реального мира. Изначально мы не располагаем векторами и тензорами. Всё что у нас есть – это какое-то сложное описание каждого объекта в выборке. Это могут быть, например, номер телефона, цвет упаковки, рост и т.д. Эти данные могут быть количественными, ранговыми и номинальными. Мы извлекаем всю цифровую информацию, которая может как-то охарактеризовать каждый аспект нашего объекта. Для того, чтобы представить объект как цифровой вектор необходимо правильно интерпретировать признак, и, соответственно, правильно его запрограммировать или отбросить за ненадобностью, например, порядковый номер.

Рассмотрим решаемую нами задачу по анализу данных. Подключаем используемые библиотеки, считываем данные. С помощью встроенных функций определяем размер выборки, определяем тип каждого признака.

import numpy as np

import pandas as pd

df = pd.read\_csv("data.csv", sep=",")

df.head()

df.shape

df.dtypes

df.isna().sum()

Наша выборка следующего размера: 569 строк и 33 столбца. Анализируя наименование и тип признака, можно сделать вывод, что только целевой признак категориальный. В выборке также присутствует признак с номером (id), его можно будет исключить из анализа.

Следующим шагом определяем, пропущены ли у нас значения в вектор признаках. Находим, что один столбец полностью «пустой» (Unnamed: 32). Его также исключаем из анализа.

df = df.drop(['id','Unnamed: 32'], axis = 1)

Категориальный признак, который мы будем предсказывать, перекодируем следующим образом:

res\_map={'M':0,

'B':1}

df['diagnosis']=df['diagnosis'].map(res\_map)

Итак, имеем следующее: если 0, то злокачественное образование, если 1 – доброкачественное.

Следующим этапом делим выборку на обучающую и тестовую и с помощью функций VarianceThreshold и Mutual information выбрать в множестве признаков наиболее важные.

VarianceThreshold – в этом методе удаляются те признаки, значение дисперсии которых не достигает установленного порога.

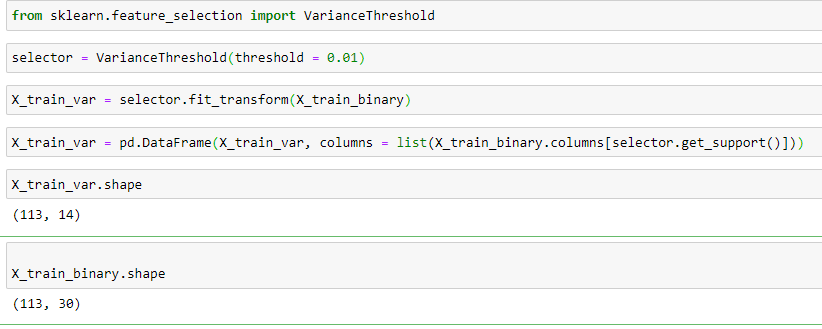
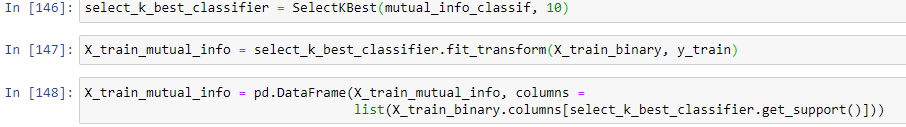


Рисунок 1 Уменьшение размерности признакового пространства с помощью VarianceThreshold

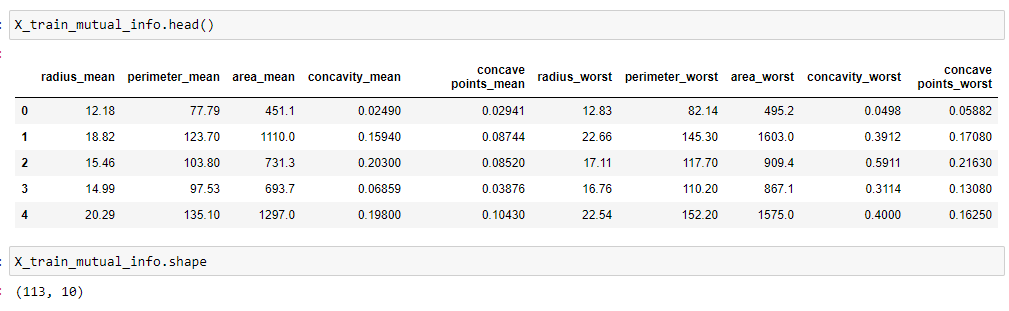
Можно отметить, что в результате выполнение функции признаковое пространство сократилось более чем в 2 раза.

Mutual information – метод, основанный на композиции двух функций, которые отбирают признаки, наиболее зависимые друг с другом (метод k ближайших соседей).



*Рисунок 2 Уменьшение размерности признакового пространства Mutual information*

В результате выполнения работы программы признаковое пространство сократилось до 10 значимых атрибутов.



*Рисунок 3 Результат выполнения Mutual information*

Итак, в данном разделе мы провели предварительную обработку данных и произвели отбор значимых признаков. В следующем разделе мы сравним качество обучения модели на разных наборах данных.

Раздел 2. Обучение моделей

Для решения нашей задачи выбираем следующие алгоритмы: RandomForestClassifier, GradientBoostingClassifier, LogisticRegression.

RandomForestClassifier (случайный лес) – один из самых универсальных алгоритмов машинного обучения, придуманные Лео Брейманом и Адель Катлер. Алгоритм применим в 70% задач, встречающихся на практике. Случайный лес представляет из себя множество решающих деревьев. В задаче регрессии их ответы усредняются, в задаче классификации принимается решение голосованием по большинству. Все деревья строятся независимо по следующей схеме:

* Выбирается подвыборка обучающей выборки размера samplesize (м.б. с возвращением) – по ней строится дерево (для каждого дерева — своя подвыборка).
* Для построения каждого расщепления в дереве просматриваем max\_features случайных признаков (для каждого нового расщепления — свои случайные признаки).
* Выбираем наилучшие признак и расщепление по нему (по заранее заданному критерию). Дерево строится, как правило, до исчерпания выборки (пока в листьях не останутся представители только одного класса), но в современных реализациях есть параметры, которые ограничивают высоту дерева, число объектов в листьях и число объектов в подвыборке, при котором проводится расщепление.

Используем библиотеку sklearn, с параметрами модели, указанными на рисунке 1.

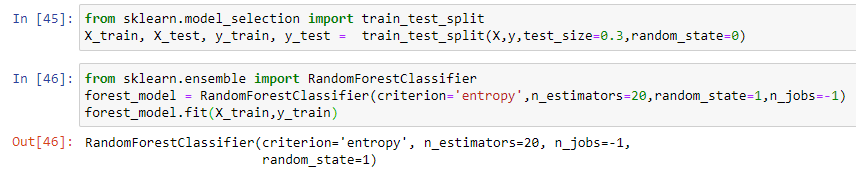


Рисунок 4 Модель машинного обучения "Случайный лес, обучение модели на тренировочных данных (полное пространство)

Обучаем модель на тренировочных данных forest\_model.fit(X\_train,y\_train), смотрим на эффективность выполнения forest\_model.predict(X\_test).

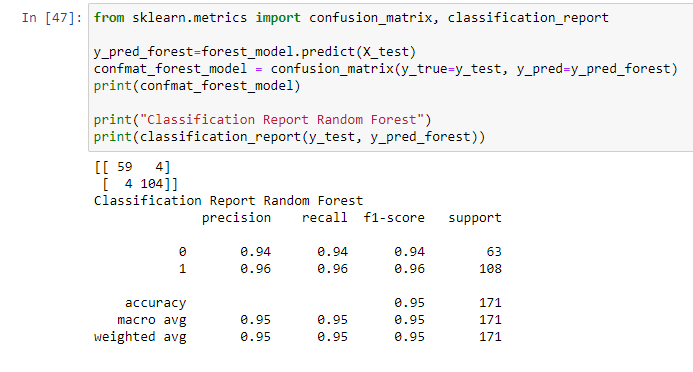
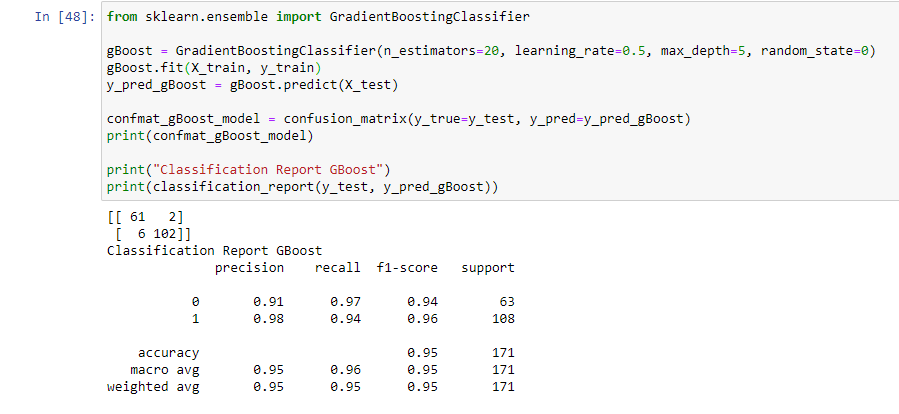


Рисунок 5 Предсказание класса моделью «Случайный лес» для тестовых данных (полное пространство)

Невооруженным глазом можно сказать, что модель работает верно, предсказывает верно для «0» с точностью 94% и «1» - с точностью 96%.

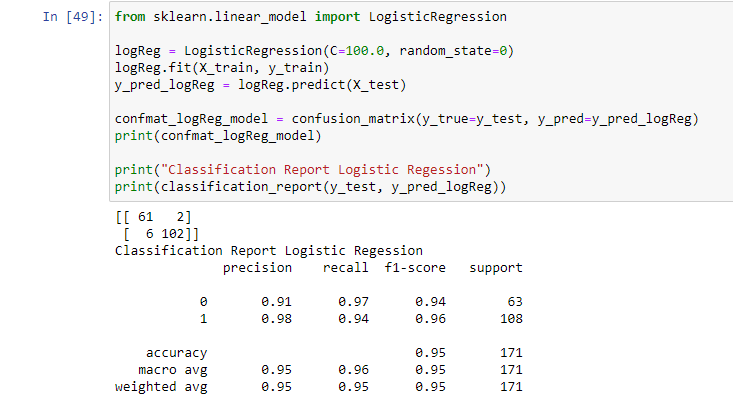
Следующей моделью используем GradientBoostingClassifier. Градиентный бустинг — это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей, обычно деревьев решений. Модель работает по индукции, таким образом, чтобы каждым следующим шагом минимизировать ошибку.



*Рисунок 6 Обучение модели «Градиентный бустинг» (полное пространство)*

Мы видим, что при заданных параметрах модель в отличии от случайного леса чуть лучше предсказывает одну переменную и чуть хуже другу. Однако, метрика оценка качества F1 совпадает с метрикой «Случайный лес». Это следует из того, каким образом строится модель градиентного бустинга.

Третьей моделью. Которую мы будем использовать, будет модель LogisticRegression. Логическая регрессия – это статистическая модель, используемая для прогнозирования вероятности возникновения некоторого события путем его сравнения с логистической кривой. Модель не просто прогнозирует, к какому классу отнести объект – 0 или 1, а определяет вероятность отнесения объекта к классу 0 или к классу 1.



*Рисунок 7 Обучение модели «LogisticRegression» на тестовых данных (полное признаковое пространство)*

Модель логистической регрессии провела анализ тестовых данных аналогично модели градиентного бустинга. Метрика F1 во всех трех алгоритмах

Проверим теперь, как будет обучаться выбранные нами модели на уменьшенном признаковом пространстве, отобранные методами Variance Threshold и Mutual information.

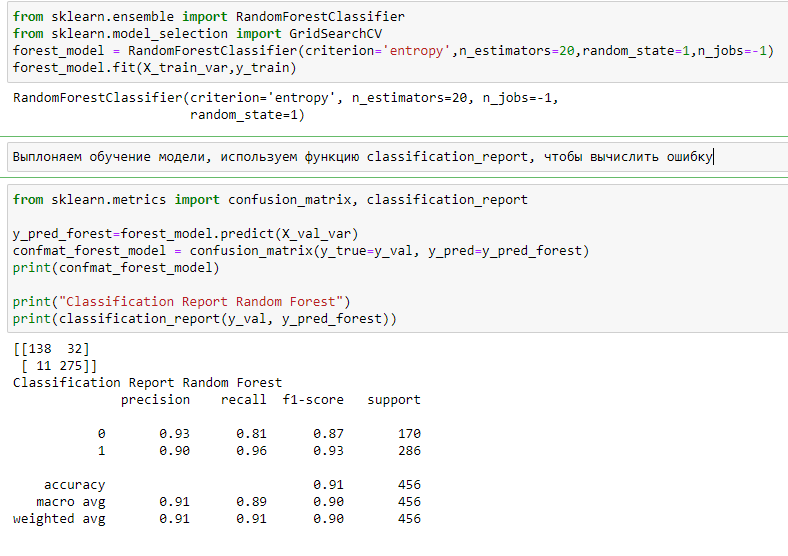


Рисунок 8 Обучение модели "Случайный лес" на данных, полученных методом Variance Threshold

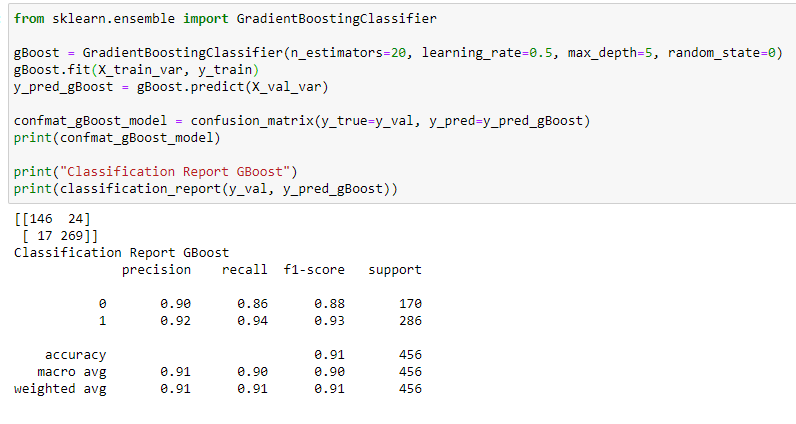


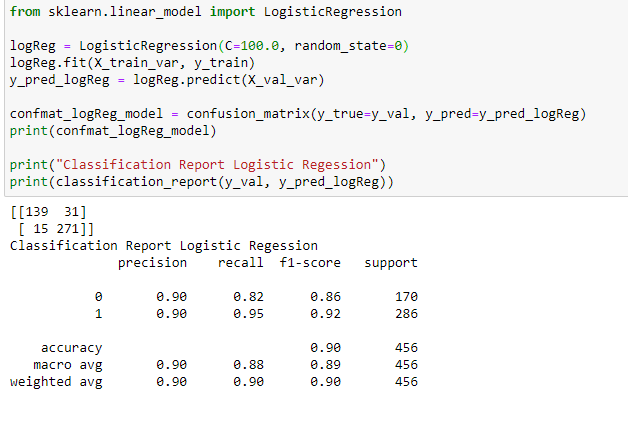
Рисунок 9 Обучение модели "Градиентный бустинг" на данных, полученных Variance Threshold 

Рисунок 10 Обучение модели "Логистической регрессии" на данных, полученных Variance Threshold

Далее, проведем обучение модели на данных, отобранных методом Mutual information.

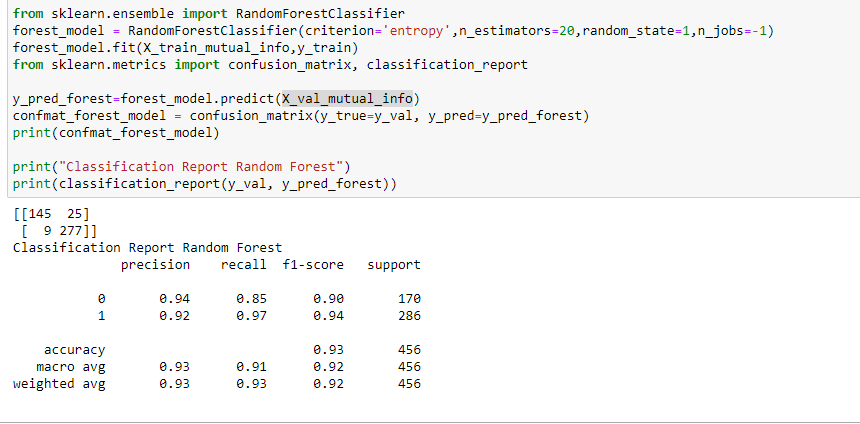


Рисунок 11 Обучение модели "Случайный лес" на данных, полученных методом Mutual information

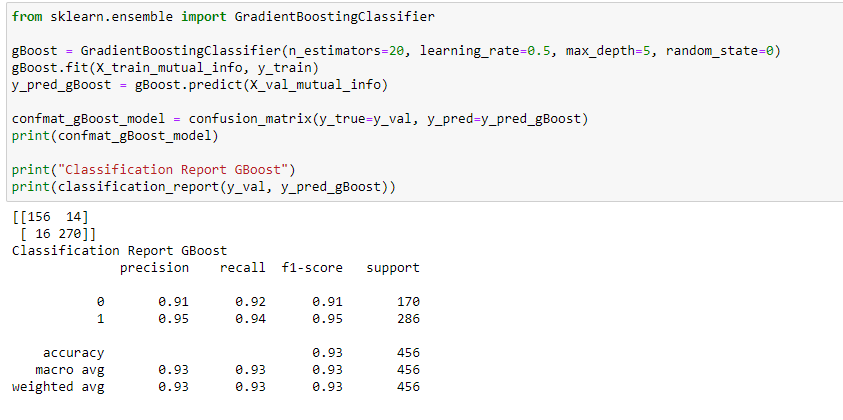


Рисунок 12 Обучение модели "Градиентный бустинг" на данных, полученных Mutual information

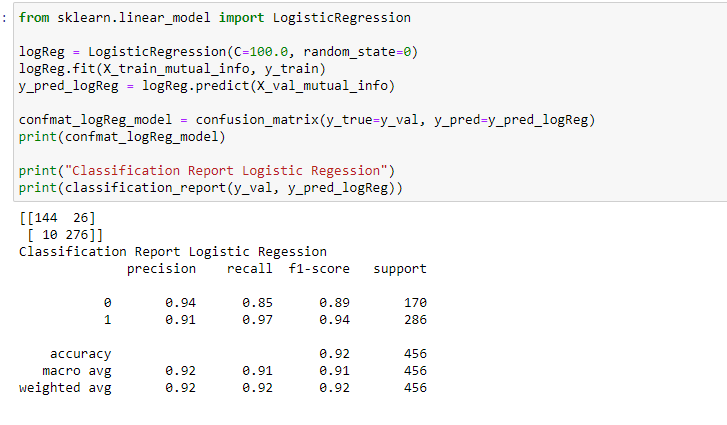


Рисунок 13 Обучение модели "Логистической регрессии" на данных, полученных Mutual information

Для наглядности, занесем полученные данные с метриками точности в таблицу.

Таблица 1 Сравнительная таблица метрик (обучение на 30 признаках)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №п/п | Модель | Precision 0 | Precision 1 | Recall 0 | Recall 1 | f1-score 0 | f1-score 1 | accuracy |
| 1 | RandomForestClassifier | 0,94 | 0,96 | 0,94 | 0,96 | 0,94 | 0,96 | 0,95 |
| 2 | GradientBoostingClassifier | 0,98 | 0,91 | 0,97 | 0,94 | 0,94 | 0,96 | 0,95 |
| 3 | LogisticRegression | 0,98 | 0,91 | 0,97 | 0,94 | 0,94 | 0,96 | 0,95 |

Таблица 2 Сравнительная таблица метрик (обучение на 14 признаках)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №п/п | Модель | Precision 0 | Precision 1 | Recall 0 | Recall 1 | f1-score 0 | f1-score 1 | accuracy |
| 1 | RandomForestClassifier | 0,93 | 0,90 | 0,81 | 0,96 | 0,87 | 0,93 | 0,91 |
| 2 | GradientBoostingClassifier | 0,90 | 0,92 | 0,86 | 0,94 | 0,88 | 0,93 | 0,91 |
| 3 | LogisticRegression | 0,90 | 0,90 | 0,82 | 0,95 | 0,86 | 0,92 | 0,90 |

Таблица 3 Сравнительная таблица метрик (обучение на 10 признаках)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №п/п | Модель | Precision 0 | Precision 1 | Recall 0 | Recall 1 | f1-score 0 | f1-score 1 | accuracy |
| 1 | RandomForestClassifier | 0,94 | 0,92 | 0,85 | 0,97 | 0,90 | 0,94 | 0,93 |
| 2 | GradientBoostingClassifier | 0,91 | 0,95 | 0,92 | 0,94 | 0,91 | 0,95 | 0,93 |
| 3 | LogisticRegression | 0,94 | 0,91 | 0,85 | 0,97 | 0,89 | 0,94 | 0,92 |

Очевидно, что обучение моделей на 10 признаках, отобранных методом Mutual information, дает более точный результат. Стоит отметить, что параметры моделей мы взяли случайно на свое усмотрение. Попробуем улучшить качество модели, настроя гиперпараметры моделей.

Теперь подберем параметры обучаемых моделей с помощью GridSearchCV и RandomizedSearchCV. Код размещать в отчете будет слишком объемно, в программе приведены параметры на каждую модель, которые подбирают функции GridSearchCV и RandomizedSearchCV

Таблица 4 Сравнительная таблица метрик (обучение на 10 признаках) модель «Случайный лес»

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №п/п | Модель | Precision 0 | Precision 1 | Recall 0 | Recall 1 | f1-score 0 | f1-score 1 | accuracy |
| 1 | RandomForestClassifier (случайные параметры) | 0,94 | 0,92 | 0,85 | 0,97 | 0,90 | 0,94 | 0,93 |
| 2 | RandomForestClassifier (GridSearchCV) | 0,93 | 0,89 | 0,80 | 0,96 | 0,86 | 0,92 | 0,90 |
| 3 | RandomForestClassifier (RandomizedSearchCV) | 0,94 | 0,92 | 0,86 | 0,97 | 0,90 | 0,94 | 0,93 |

Таблица 5 Сравнительная таблица метрик (обучение на 10 признаках) модель «Градиентный бустинг»

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №п/п | Модель | Precision 0 | Precision 1 | Recall 0 | Recall 1 | f1-score 0 | f1-score 1 | accuracy |
| 1 | GradientBoostingClassifier  (случайные параметры) | 0,98 | 0,91 | 0,97 | 0,94 | 0,94 | 0,96 | 0,95 |
| 2 | GradientBoostingClassifier (GridSearchCV) | 0,93 | 0,92 | 0,86 | 0,96 | 0,90 | 0,94 | 0,93 |
| 3 | GradientBoostingClassifier (RandomizedSearchCV) | 0,91 | 0,95 | 0,92 | 0,94 | 0,91 | 0,95 | 0,93 |

Таблица 6 Сравнительная таблица метрик (обучение на 10 признаках) модель «Логистическая регрессия»

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №п/п | Модель | Precision 0 | Precision 1 | Recall 0 | Recall 1 | f1-score 0 | f1-score 1 | accuracy |
| 1 | LogisticRegression  (случайные параметры) | 0,94 | 0,91 | 0,85 | 0,97 | 0,89 | 0,94 | 0,92 |
| 2 | LogisticRegression (GridSearchCV) | 0,92 | 0,91 | 0,85 | 0,95 | 0,88 | 0,93 | 0,91 |
| 3 | LogisticRegression (RandomizedSearchCV) | 0,96 | 0,92 | 0,85 | 0,98 | 0,90 | 0,95 | 0,93 |

Нами были описаны принципы работы используемых алгоритмов машинного обучения, двумя методами мы сократили размерность признаков, которыми пользовались для обучения моделей. Также, двумя функциями подобрали гиперпараметры к исследуемым моделям. Анализируя общие результаты, можно сказать, что наибольшей точности мы добивались, когда использовали полное пространство признаков, исключая неинформативные (пустые). Метрики показывают, что качество работы моделей не сильно отличается друг от друга.

Заключение. Выводы, сравнение точности используемых моделей

В работе была произведена предварительная обработка данных, очистка от неинформативных признаков, перекодирование категориальных признаков в численные. Исследуемый набор данных был с достаточно небольшим набором признаков, поэтому сокращение размерности признакового пространства не улучшило используемые модели машинного обучения. Вероятно, из признакового пространства были удалены важные признаки, с точки зрения исследуемой области.

Для определения наилучшей модели и настройки параметров стоит сказать о матрице ошибок.

|  |  |  |
| --- | --- | --- |
|  | Реальность | |
| Прогноз | TP | FP |
| FN | TN |

Рисунок 14 Матрица ошибок

P – число истинных результатов, P=TP+TN

N - число ложных результатов, N=FP+FN

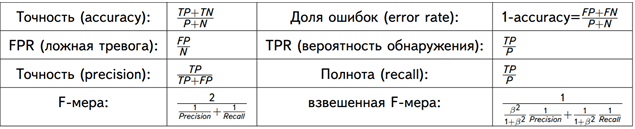


Рисунок 15 Метрики оценки качества

Все метрики качества дают оценки модели более 80%, что можно назвать хорошим результатом. Мы попытались настроить гиперпараметры каждой модели, но выбранные первоначально параметры показали наилучший результат практически по всем метрикам. Однако стоит учесть следующее: в некоторых задачах на практике выбор гиперпараметров зависит от цели исследователей. Можно привести в пример нашу задачу: нам важнее не назвать злокачественное образование доброкачественным. Соответственно, необходимо уменьшить FN.

Относительно качества моделей, даже на 10 параметрах модель GradientBoostingClassifier дает наиболее точные результаты.