Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра математических методов прогнозирования

Кулик Андрей

Прогноз исхода туберкулеза

Практическая работа

Научный руководитель:
к.ф.-м.н.
О.В.Сенько

Москва, 2020

Содержание

1 Введение

В данной работе рассматривается бинарная классификация. Цель определения - относится пациент к „1“-ой или ко „2“-ой группе. Были использованы разные методы машинного обучения для данной цели. Для выявления преймуществ и недостатков каждого из подходов производится сравнение предсказаний на разных метриках.

2 Исследуемые данные

Для проведения эксперементов был загружен датасет *«medicina3»*. В данных содержится целевая переменная *«gr»*(человек относится либо к 1-ой, либо ко 2-ой группе), а также 28 признаков, по которым будет производится обучение моделей. Дынные не имеют выбросов, а также пустых клеток. Целевая переменная распределена неравномерно, что вполне реалитично, для задачи диагностики болезни(Рис.1). Поэтому в эксперементах будут применены метрики, которые устойчивы к неравномерным данным.

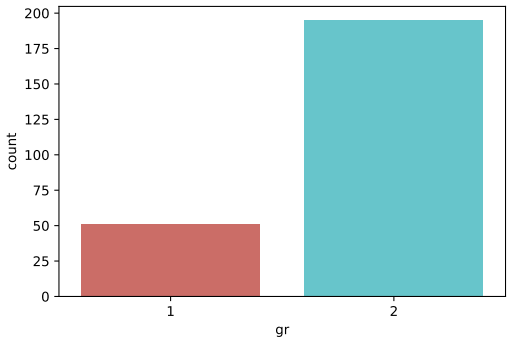


Рис. 1: Распеределение целевой переменной

3 Метрики классификации

Перед переходом к самим метрикам введем матрицу ошибок для бинарной классификации.

	y = 1	y = 0
$\hat{y} = 1$	True Positive (TP)	False Positive (FP)
$\hat{y} = 0$	False Negative (FN)	True Negative (TN)

Здесь $\hat{y}$ — это ответ алгоритма на объекте, а y — истинная метка класса на этом объекте. Таким образом, ошибки классификации бывают двух видов: False Negative (FN) и False Positive (FP). В данной работе True Positive будет является правильное определение человека ко „2“-ой группе.

3.1 Accuracy

Метрика показывающая долю правильных ответов алгоритма. $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ Эта метрика бесполелезна при случае неравных классов, в чем убедимся в эксперементах. Преодолеть это можно с помощью подсчета относительных пропорций типов ошибок.

3.2 Specificity, sensitivity, F1 score

Sensitivity(чувствительность), вычисляется по формуле $sensitivity = \frac{TP}{TP+FN}$ Поскольку формула не учитывает TN и FP, данная метрика может дать нам смещенную оценку, в случае, когда в целевой переменной преобладание отрицательных исходов.

Specificity(специфичность), вычисляется по формуле $specificity = \frac{TN}{TN+FP}$ Поскольку формула не учитывает TP и FN, данная метрика может дать нам смещенную

оценку, в случае, когда в целевой переменной преобладание положительных исходов.

F1 score(F-мера), вычисляется по формуле $F1 = \frac{precision \cdot sensitivity}{precision + sensitivity}$.
Здесь $precision = \frac{TP}{TP+FP}$ - метрика точности.
F1 score позволяет получить более сбалансированную характеристику модели, так как объеденяет в себе сразу 2 метрики. Однако может дать смещенную оценку из-за неучета TN.

3.3 AUC-ROC

Одним из способов оценить модель является AUC-ROC - площадь под кривой ошибок. Данная кривая представляет из себя линию от (0,0) до (0,1) в координатах TRP и FPR, где

$$TRP = \frac{TP}{TP + FN}$$

$$TFP = \frac{FP}{FP + TN}$$

TPR - это полнота, а FPR показывает, какую долю из объектов отрицательного класса модель предсказала неверно.
Критерий ROC-AUC устойчив к несбалансированным классам, что подходит для вышепоставленной задачи.

4 Модели обучения

4.1 Logistic model and SVM

4.1.1 Подготовка данных

Для данных методов важно использовать только **значимые переменные**. Одним из методов решения данной задачи является *recursive feature elimination*(RFE). Он основывается на повторяющемся конструировании модели и выборе лучше всех или хуже всех выполняемого признака, отделения этого признака и повторения цикла с оставшимися. Этот процесс применяется, пока в наборе данных не закончатся признаки. Цель RFE заключается в отборе признаков посредством рекурсивного рассмотрения всё меньшего и меньшего их набора.

Метод RFE импортируется из библиотеки *sklearn* и применяется для датасета. Затем строим таблицу и убираем признаки у которых высокое P-значение. В итоге получаем только значимые перемнные, которые малокорелируют между собой.(Рис 2.)

```
Optimization terminated successfully.
         Current function value: 0.303598
         Iterations 8
                         Results: Logit
================================================================
Model:              Logit            Pseudo R-squared: 0.399
Dependent Variable: gr               AIC:              120.4378
Date:               2020-10-28 16:26 BIC:              145.6177
No. Observations:   172              Log-Likelihood:   -52.219
Df Model:           7                LL-Null:          -86.894
Df Residuals:       164              LLR p-value:      1.9979e-12
Converged:          1.0000           Scale:            1.0000
No. Iterations:     8.0000
----------------------------------------------------------------
           Coef.    Std.Err.      z      P>|z|     [0.025    0.975]
----------------------------------------------------------------
v4         0.2171     0.0577    3.7662   0.0002    0.1041    0.3301
v7        -3.5177     0.6889   -5.1065   0.0000   -4.8679   -2.1676
v11        1.2645     0.5918    2.1369   0.0326    0.1047    2.4244
v16        0.3655     0.0943    3.8766   0.0001    0.1807    0.5504
v19        1.4870     0.9289    1.6008   0.1094   -0.3336    3.3075
v20       -1.4006     0.9230   -1.5175   0.1291   -3.2096    0.4084
v23       -1.3593     0.3645   -3.7297   0.0002   -2.0736   -0.6450
v28       -0.3522     0.1201   -2.9311   0.0034   -0.5876   -0.1167
================================================================
```

Рис.2 данные после выделения только значимых признаков

Данные подготовлены, можно приступать к обучению моделей.

4.1.2 Logistic regression

Logistic regression — метод построения линейного классификатора, позволяющий оценивать апостериорные вероятности принадлежности объектов классам.
Бинарный случай. Пусть $Y = \{-1, +1\}$. В логистической регрессии строится линейный алгоритм классификации $a : X \to Y$ вида

$$a(x, w) = sign\langle x, w\rangle = sign(\sum_{j=1}^{n} w_j f_j(x) - w_0)$$

Тогда задача классификатора заключается в настройке вектора весов w по обучающей выборке. Для этого решается задача задача эипречиского риска с функцией потерь специального вида:

$$Q(w) = \sum_{i=1}^{m} ln(1 + exp(-y_i\langle x_i, w\rangle)) \to \min_w$$

Далее логистическая регрессия может делать классификацию двух типов, определить метку класса $a(x) = sign\langle x, w\rangle$, а также вычислить апостериорные вероятности принадлежности классам:
$P\{y|x\} = \sigma(y\langle x, w\rangle)$, $y \in Y$, где $\sigma(z) = \frac{1}{1+e^{-z}}$ - сигмоидная функция.

4.1.3 Support vector machine для линейно разделимой выборки

Support vector machine (SVM) основан на концепции гиперплоскостей, которые определяют границы гиперповерхностей. При этом они строятся так, чтобы 'зазор' между ними был максимальным. Получим систему описывающую данные гиперплоскости:

$$zx^t = b + 1$$

$$zx^t = b - 1,$$

Тогда расстояние δ между плоскостями равно $\frac{2}{|z|}$.
Таким образом задача обучения сводиться к оптимазицонной задаче с ограничениями

$$\delta = \frac{2}{|z|} \to max$$

$$zx_j^t \geq b + 1, \text{ при } s_j \in K_1 \bigcap \widetilde{S}_t,$$

$$zx_j^t \leq b - 1, \text{ при } s_j \in K_2 \bigcap \widetilde{S}_t.$$

При этом оптимизация производится по компонентам направляющего вектора $z = (z_1, ..., z_n)$ и параметру сдвига b. Без ограничения общности можно считать, что метки классов равны:

$$X(\omega) = \begin{cases} \mathrm{y}_j = 1 & \text{при } s_j \in K_1 \\ \mathrm{y}_j = -1 & \text{при } s_j \in K_2 \end{cases}$$

Тогда задача эквивалентна решению:

$$\frac{1}{2}\sum_{i=1}^{n} z_i^2 \to min$$

$$y_j(zx_j^t - b) \geq 1, j = 1, ..., m$$

Данная задача решается с помощью применения теоремы Каруша-Куна-Такера.

4.2 Методы основанные на потроении деревьев

В отличии от линейных методов, данные методы не требуют преподготовки значимых переменных. Данную проблему решает дерево принятия решений. Деревья решений - бинарное разбиение, при котором на каждом этапе количество вариантов будет уменьшаться примерно вдвое очень быстро сужая варианты.

4.2.1 Random forest

Главное проблемой деревья решений является переобучение, ведь можно дойти до слишком шлубокого уровня дерева,аппроксимируя таким образом нюансы конкретных данных вместо общих характеристик распределений, из которых они получены. Решением данной задачи называется *баггинг*.Багинг использует ансамбль параллельно работающих переобучаемых оценивателей и усредняет результаты методом *голосования* для получения оптимальной классификации. Ансамбль случайных деревьев принятия решений называется **random forest**.

Для более эффективной рандомизации деревьев принятия решений обеспечивается определенная стохастичность процесса выбора разбиений. При этом всякий раз в обучении участвуют все данные, но результаты обучения все равно сохраняют требуемую случайность.

4.3 Бустинги

Бустинг — это техника построения ансамблей, в которой предсказатели построены не независимо, а последовательно. Это техника использует идею о том, что следующая модель будет учится на ошибках предыдущей. Они имеют неравную вероятность появления

в последующих моделях, и чаще появятся те, что дают наибольшую ошибку. Предсказатели могут быть выбраны из широкого ассортимента моделей, например, деревья решений, регрессия, классификаторы и т.д. Из-за того, что предсказатели обучаются на ошибках, совершенных предыдущими, требуется меньше времени для того, чтобы добраться до реального ответа. Но мы должны выбирать критерий остановки с осторожностью, иначе это может привести к переобучению.

4.3.1 AdaBoost

Адаптивный бустинг, известный как AdaBoost - это алгоритма Бустинга. Метод, который использует этот алгоритм для минимизации ошибки предыдущих моделей, заключается в концентрации внимания на недообученности алгоритма. Что значит, при каждом новом предсказании алгоритм будет работать над сложными, неочевидными для предсказания подвыборками.
$Q(b, W^l) = \sum_{i=1}^{n} \omega_i [y_i b(x_i) < 0]$ - стандартный функционал качества алгоритма классификации b. Здесь $W^l = (\omega_1, ..., \omega_l)$ -вектор весов объектов, b_i - базовый алгоритм, возвращающий либо -1, либо 1.
Задачу оптимизации праметра a_t(прогноз) решаем, аппроксимируя пороговую функцию потерь $[z < 0]$ с помощью экспоненты $E(z) = exp(-z)$.

4.3.2 Gradient Boosting

Gradient boosting - это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей. Задача состои в том, чтобы минимизировать функцию потерь, используя метод градиентного спуска. Таким образом изменяя предсказания, основанные на *learning rate*, ищем значения, на которых сумма функций потерь стримиться к минимуму.

$$Q = \sum_{i=1}^{N} L(y_i, F_m(x_i)) \to min$$

5 Эксперементы

5.1 Cross validation

Cross validation - процедура эмпирического оценивания обобщающей способности алгоритмов, обучаемых по прецедентам. Фиксируется некоторое множество разбиений исходной выборки на две подвыборки: обучающую и контрольную. Для каждого разбиения выполняется настройка алгоритма по обучающей подвыборке, затем оценивается его средняя ошибка на объектах контрольной подвыборки. Оценкой скользящего контроля называется средняя по всем разбиениям величина ошибки на контрольных подвыборках.
Таким образом мы можем контролировать переобучение модели и получать более реальные оценки.
CV on q-Folds. Выборка случайным образом разбивается на q непересекающихся блоков одинаковой (или почти одинаковой) длины $k_1, ..., k_q$:

$$X^L = X_1^{k_1} \cap ... \cap X_q^{k_q}$$

$L = k_1 + ... + k_q$. Каждый блок по очереди становится контрольной подвыборкой, при этом обучение производится по остальным q-1 блокам. Критерий определяется как средняя

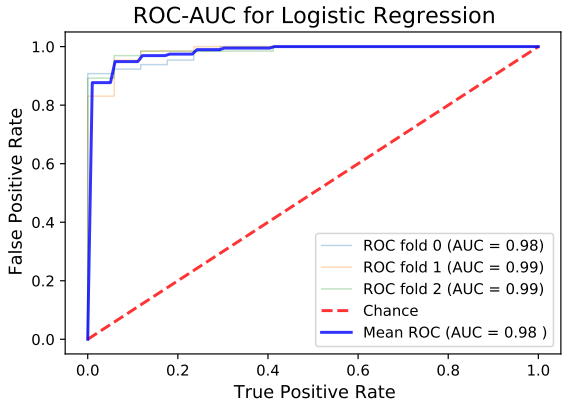
ошибка на контрольной подвыборке:

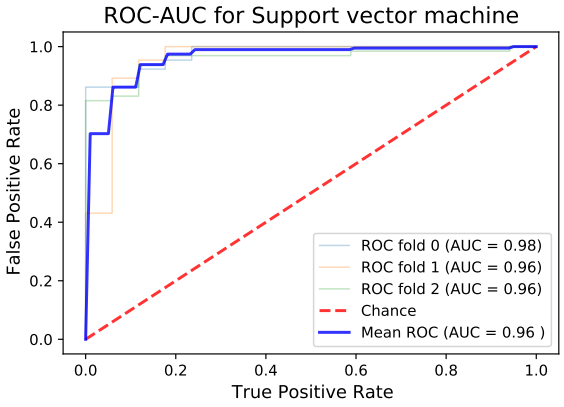
$$CV(\mu, X^L) = \frac{1}{q}\sum_{n=1}^{q} Q(\mu(X^L \setminus X_n^{k_n}), X_n^{k_n})).$$

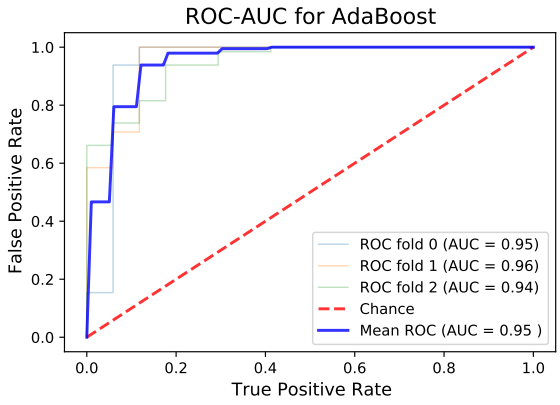
Все готово к проведению эксперементов. Методы SVM, LogicRegression, AdaBoostClassifier, RandomForestClassifier сымпортированы из библиотеки *sklearn*. Были проведены подсчеты метрик на кросс-валидации с **5**-тью фолдами для сравнения методов машинного обучения.(Табл. 1)

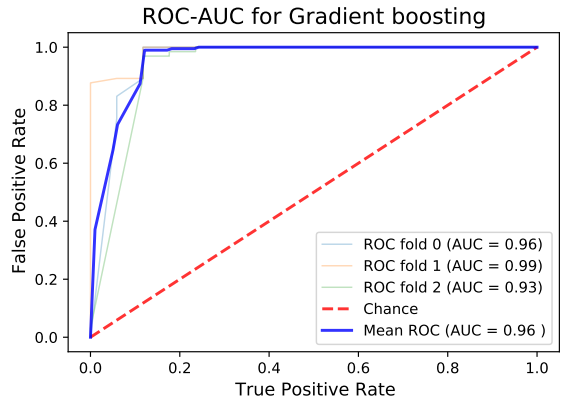
Табл. 1 Средняя точность метрик по 5 фолдам

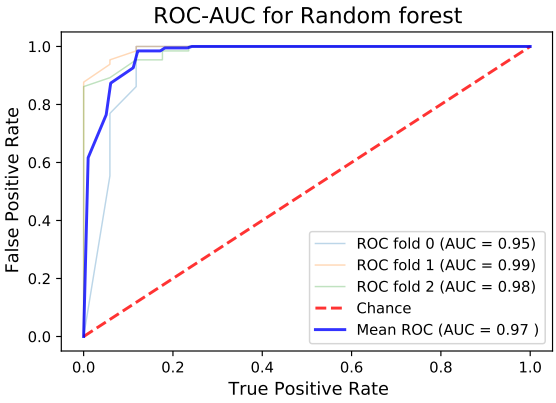
Model	Accuracy	Sensitivity	Specificity	F1-score
Logistic regression	0.92	0.913	0.937	0.946
SVM	0.95	0.965	0.875	0.965
Random forest	0.97	0.982	0.937	0.982
AdaBoost	0.95	0.965	0.875	0.965
Gradient boosting	0.965	1.0	0.875	0.983











В целом все модели показали хорошие результаты, поэтому выбор модели будет зависеть от цели поставленной задачи. Если важно, чтобы модель клиенту с состоянием болезни „2“всегда распозновала это, то лучшая модель будет градиентный бустинг, с показателем чувствительности 1.0. Если же важно, чтобы модель клиенту с состоянием болезни „1“верно указывала его диагноз, то лучше подойдет Random Forest или Logistic Regression. Также, эти 2 модели показали лучший результат на графиках ROC-AUC. Данный опыт проводился на 4 фолдах, и в дальнейшем брался средний показатель по всей кросс-валидации.Random Forest и Logistic regression оказались наиболее эффективными к несбалансированным данным, что показано на графиках.

6 Вывод

В данной работе была изучена проблема бинарной классификации. Были применены разные модели машинного обучения для данной задачи и сравнены между собой по результатам прогнозирования исхода туберкулеза.
Важно понимать, каких результатов вы ждете от модели. Если в приоритете зафиксировать „2“-ую группу человека, то нужно выбирать модель с высокой чувствительностью, относительно этого показателя. А если важно распозновать „1“-е состояние человека, то лучше выбрать модель с высоким показателем специфичности относительно показателя „2“.
Если же необходимо выбрать сбалансированную модель к любым ошибкам, то нужно смотреть на показатель ROC-AUC, данный показатель устойчив к несбалансированным данным и отражает качество всей модели.
Также применяя данные алгоритмы нужно учитывать тип данной модели. Если это линейная модель, то стоит сделать преподготовку данных и убрать лишние параметры, которые плохо коррелируют с целевой переменной. Для этого можно применить алгоритм RFE. Если это модель на основе деревевьев, то стоит правильно выбирать глубину, количество признаков и деревьев, чтобы моделе не была переобучена. Модели бустинга также сколны к переобучению, нужно смотреть на недостатки аппроксимируемой функции.