Владивостокский государственный университет экономики и сервиса

Научный руководитель:
Гресько Александр Александрович, к.э.н.

ПОСТРОЕНИЕ ПРОГНОЗНОЙ МОДЕЛИ ДЛЯ ДИАГНОСТИКИ ИШЕМИЧЕСКОЙ БОЛЕЗНИ СЕРДЦА

Завалин Георгий Сергеевич
mescalin_z@mail.ru

Введение

Интеллектуальный анализ данных - это процесс извлечения скрытых знаний из данных. С помощью данного анализа можно выявлять закономерности и взаимосвязи между большими объемами. Некоторые отрасли, такие как банковское дело, страхование и маркетинг, используют интеллектуальный анализ данных для снижения затрат и увеличения прибыли. В настоящее время ишемическая болезнь сердца занимает первое место среди причин смертности людей от сердечно-сосудистых заболеваний. По данным Всемирной Организации Здравоохранения (ВОЗ) смертность от сердечно-сосудистых заболеваний составляет 31% и является наиболее частой причиной смертельных исходов во всем мире. На территории Российской Федерации этот показатель составляет 57,1%, из которых на долю ИБС выпадает более половины всех случаев (28,9%), что в абсолютных цифрах составляет 385,6 человек на 100 тысяч населения в год. [1] Наиболее достоверным способом диагностики ишемической болезни сердца принято считать коронарографию, которая позволяет определять место и степень сужения коронарной артерии. Но данный метод является затратным и при этом опасным для жизни. В связи с этим в данный момент на разных данных исследователями разрабатываются различные модели машинного обучения, помогающие в диагностике ишемической болезни сердца. В данном исследовании изложен вариант построения модели классификации для диагностики ишемической болезни сердца с применением языка программирования Python.

Основная часть

В данной работе будет использован набор данных Ализаде Сани (Alizade Sani), предоставленный турецкими исследователями. Датасет содержит в себе данные о 303 случайных пациентов сердечно-сосудистого медицинского и научно-исследовательского центра Shaheed Rajaei. 216 пациентов имели пораженные коронарные артерии, остальные были здоровы. Под здоровым человеком принято понимать пациента с сужением сосудов менее 50 процентов. Загружаем исходный датасет (массив данных) в Jupiter Notebook, используя библиотеку для работы с данными pandas (рисунок 1). Jupyter Notebook — это мощный инструмент для разработки и представления проектов анализа данных в интерактивном виде. Он объединяет код на языке программирования Python и вывод все в виде одного документа, содержащего текст, математические уравнения и визуализации. [2]

Каждый объект датафрейма(пациент) имеет 55 признаков, которые являются медицинскими показателями пациента, которые были получены перед прохождением коронарографии. Среди них можно встретить следующие показатели: пол, вес, возраст, курение, сахарный диабет, гипертония, глюкоза, триглицериды, одышка, гемоглобин и т.д. Категориальные признаки такие, как пол, курение, хрипы в легких, одышка и

остальные, были переведены в дискретный вид на этапе подготовки датасета к загрузке. Выходная переменная(Y) имеет два значения: 0 – сосуды в норме, 1 – ишемическая болезнь сердца (рисунок 2).

```
In [3]: import pandas as pd
        data=pd.read_csv('zade.csv',sep=';',decimal=',', encoding='Windows-1251')
        data.head()
```

Out[3]:

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	...	X47	X48	X49	X50	X51	X52	X53	X54	X55	Y
0	53	90	175	0	29.387755	0	1	1	0	0	...	4.7	141	5700	39	52	261	50	0	0	1
1	67	70	157	1	28.398718	0	1	0	0	0	...	4.7	156	7700	38	55	165	40	4	0	1
2	54	54	164	0	20.077335	0	0	1	0	0	...	4.7	139	7400	38	60	230	40	2	1	1
3	66	67	158	1	26.838648	0	1	0	0	0	...	4.4	142	13000	18	72	742	55	0	2	0
4	50	87	153	1	37.165193	0	1	0	0	0	...	4.0	140	9200	55	39	274	50	0	2	0

Рис. 1. Загрузка датасета

```
In [4]: data.Y.unique()
Out[4]: array([1, 0], dtype=int64)
```

Рис. 2. Выходная переменная

Далее проверим наши предикторы на статистическую значимость. Весь смысл статистической значимости заключается в том, чтобы определить, имеет ли под собой какое-то основание разница между двумя показателями, или же она случайна. [3] Уровень значимости возьмем p <0,05. Для непрерывных переменных будем использовать критерий Краскела-Уоллиса, а для категориальных – хи-квадрат.

Критерий Краскела-Уоллиса - это непараметрическая альтернатива одномерному (межгрупповому) дисперсионному анализу. Он используется для сравнения трех или более выборок, и проверяет нулевые гипотезы, согласно которым различные выборки были взяты из одного и того же распределения, или из распределений с одинаковыми медианами. [4] Для реализации критерия датафрейм с данными был разбит на два датафрейма: первый содержит данные только об пациентах с ишемической болезнью сердца, второй – данные об пациентах с здоровыми сосудами. В Python критерий Краскела-Уоллиса реализуется с помощью функции kruskal из библиотеки scipy.stats. Пример реализации критерии для признака «Возраст» представлен на рисунке 3. Далее для всех остальных непрерывных признаков при помощи цикла были получены соответствующие значения p-value.

```
In [9]: from scipy.stats import kruskal
        stat, p = kruskal(Y0['X1'],Y1['X1'],nan_policy='omit')
        p
Out[9]: 1.0492201824580363e-09
```

Рисунок 3 – Критерий Краскела-Уоллиса

Хи-квадрат Пирсона один из самых популярных статистических критериев для анализа качественных данных (номинальных, порядковых, ранговых), анализа частот. Однако, как и у каждого статистического критерия у хи-квадрата есть свои собственные правила применения метода, его интерпретации. [5] Подобно предыдущему критерию в библиотеке scipy.stats имеется функция для расчета критерия хи-квадрат. При этом для

этой функции следует подготовить таблицы сопряженности, необходимые для реализации данного критерия. Пример расчета p-value для признака «Пол» представлен на рисунке 4.

```
In [12]: from scipy.stats import chi2_contingency
         tbl=pd.concat((X['X4'],Y), axis=1)
         tbl=tbl.dropna()
         tik=pd.crosstab(tbl['X4'],tbl['Y'])
         stat, p, dof, expected = chi2_contingency(tik)
         p

Out[12]: 0.29912748032994074
```

Рис. 4. Критерий хи-квадрат

После получения значений p-value для всех признаков была составлена итоговая таблица со значимыми предикторами. Из 55 признаков значимыми оказались только 22 (таблица 1).

Таблица 1. Значимые предикторы

Предиктор	Значение p-value
Возраст	1.053992e-09
Сахарный диабет	0.00002
Гипертония	1.061399e-06
Систолическое артериальное давление	0.00001
Диастолическое артериальное давление	0.00324
Диастолический шум	0.02918
Типичная стенокардия	1.120876e-20
Одышка	0.04019
Атипичная стенокардия	1.22741e-12
Кардиалгия	0.0000071
Зубец Q	0.02009
Элевация ST	0.03325
Депрессия ST	0.01807
Инверсия зубца T	0.000067
Глюкоза	0.00004
Триглицериды	0.00131
Скорость оседания эритроцитов	0.00026
Калий	0.00173
Лимфоциты	0.02996
Нейтрофилы	0.03111
Поражение клапана	0.00103
Фракция выброса	1.627515e-07

Из датафрейма с признаками были удалены все незначимые предикторы. После чего было выполнено разбиение всего датасета на обучающую и тестовую выборки в равном соотношении. Разбиение реализовано с помощью функции train_test_split из библиотеки sklearn (рисунок 5).

```
In [15]: X_update=X
         dr=[2,3,4,5,8,9,10,11,12,13,14,15,16,17,20,21,22,23,27,30,31,36,37,38,40,42,43,44,46,48,49,52,54]
         for i in dr:
             X_update=X_update.drop(['X'+str(i)],axis=1)
         X_update
```

Out[15]:

	X1	X6	X7	X18	X19	X24	X25	X26	X28	X29	...	X34	X35	X39	X41	X45	X47	X50	X51	X53	X55
0	53	0	1	110	80	0	0	0	0	0	...	1	1	90	250	7	4.7	39	52	50	0
1	67	0	1	140	80	0	1	0	0	0	...	1	1	80	309	26	4.7	38	55	40	0
2	54	0	0	100	100	0	1	0	0	0	...	0	0	85	103	10	4.7	38	60	40	1
3	66	0	1	100	80	1	0	1	0	1	...	1	0	78	63	76	4.4	18	72	55	2
4	50	0	1	110	80	0	0	1	0	0	...	0	0	104	170	27	4.0	55	39	50	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	58	0	0	100	76	0	1	0	0	0	...	0	0	92	112	13	4.8	34	58	45	0
299	55	0	0	100	60	0	0	1	1	0	...	0	0	86	111	3	4.0	16	80	40	1
300	48	0	1	130	70	0	0	0	0	1	...	0	0	83	93	20	4.0	35	55	55	0
301	57	1	0	100	60	0	0	1	1	0	...	0	0	96	116	31	3.8	48	40	55	0
302	56	0	1	120	80	0	1	0	0	0	...	0	1	78	139	13	4.4	32	55	55	0

303 rows × 22 columns

```
In [17]: from sklearn.model_selection import train_test_split
         X_train, X_test, Y_train, Y_test = train_test_split(X_update, Y, test_size=0.5, random_state=0)
```

Рис. 5. Разбиение на обучающую и тестовые выборки

Задача заключается в предсказании выходной переменной (Y) по имеющимся признакам у пациента. В качестве алгоритма классификации была взята логистическая регрессия. Логистическая регрессия — это алгоритм классификации машинного обучения, используемый для прогнозирования вероятности категориальной зависимой переменной. В логистической регрессии зависимая переменная является бинарной переменной, содержащей данные, закодированные как 1 (да, успех и т.п.) или 0 (нет, провал и т.п.). [6] В Python модель логистической регрессии задается с помощью функции LogisticRegression из библиотеки sklearn (рисунок 6). В нашем случае созданная модель (logreg) обучается на данных обучающей выборки (X_train, Y_train). После обучения модели при помощи функции predict предсказываем значения выходной переменной для тестовой выборки (y_pred). После предсказания рассчитаем точность прогноза (accuracy) с помощью функции score. Точность рассчитывается как сумма верно предсказанных 0 и 1, деленная на количество объектов в выборке.

```
In [19]: from sklearn.linear_model import LogisticRegression
         logreg = LogisticRegression(max_iter=10000)
         logreg.fit(X_train,Y_train.values.ravel())
         logreg

Out[19]: LogisticRegression(max_iter=10000)

In [21]: y_pred = logreg.predict(X_test)
         print('Accuracy: {:.2f}'.format(logreg.score(X_test, Y_test)))

         Accuracy: 0.88
```

Рис. 6. Реализация модели логистической регрессии

Точность классификатора логистической регрессии для тестовой выборки оказалась равной 0,88. Для оценки качества созданной модели рассчитаем соответствующие метрики с помощью функции classification_report, а также посмотрим на матрицу ошибок нашей модели. Матрица ошибок показывает нам, сколько 0 и 1 алгоритм предсказал верно, а сколько неверно (рисунок 7). В строках матрицы, предсказанные 1 и 0 классификатором, а в столбцах 0 и 1 являются истинными метками классов.

```
In [22]: from sklearn.metrics import confusion_matrix
         confusion_matrix = confusion_matrix(Y_test, y_pred)
         print(confusion_matrix)

         [[36  9]
          [10 97]]
```

Рис. 7. Матрица ошибок

Метрики recall и precision используются для оценки качества модели по каждому из классов (рисунок 8). Высокие показатели по каждой из метрик говорят нам о том, что реализованная модель является классификатором хорошего качества для задачи диагностики ишемической болезни сердца.

```
In [23]: from sklearn.metrics import classification_report
         print(classification_report(Y_test, y_pred))

                       precision    recall  f1-score   support

                    0       0.78      0.80      0.79        45
                    1       0.92      0.91      0.91       107

             accuracy                           0.88       152
            macro avg       0.85      0.85      0.85       152
         weighted avg       0.88      0.88      0.88       152
```

Рис. 8. Оценка качества модели

В завершении оценки качества модели построим ROC-кривую, которая является популярным инструментов при работе с бинарными классификаторами (рисунок 9). Пунктирная линия представляет ROC-кривую полностью случайного классификатора. Хороший классификатор остаётся от неё максимально далеко (по направлению к верхнему левому углу) [6].

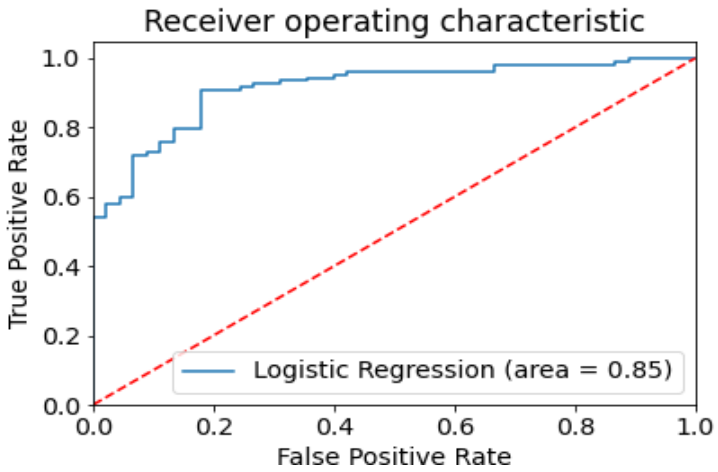

Рис. 9. ROC-кривая

Заключение

Таким образом, в данной работе на реальных медицинских данных была реализована модель логистической регрессии в качестве бинарного классификатора для диагностики ишемической болезни сердца. С помощью статистических критериев Краскела-Уоллиса и хи-квадрат для модели были отобраны значимые признаки. В

хорошем качестве реализованной модели нам помогли удостовериться различные метрики машинного обучения.

Источники:

1 Ишемическая болезнь сердца [Электронный ресурс] / Режим доступа: https://www.rusintervention.ru/пациентам/заболевания/ибс/

2 Введение в работу с Jupiter Notebook [Электронный ресурс] / Режим доступа: https://pythonru.com/baza-znanij/jupyter-notebook-dlja-nachinajushhih

3 Проверка статистической значимости [Электронный ресурс] / Режим доступа: http://datascientist.one/proverka-stat-znachimosti/

4 Критерий Краскела-Уоллиса [Электронный ресурс] / Режим доступа: http://statistica.ru/local-portals/medicine/kriteriy-kraskela-uollisa/

5 Критерий хи-квадрат Пирсона [Электронный ресурс] / Режим доступа: https://lit-review.ru/biostatistika/kriterijj-khi-kvadrat-pirsona/

6 Пошаговое построение логистической регрессии в Python [Электронный ресурс] / Режим доступа: https://medium.com/nuances-of-programming/пошаговое-построение-логистической-регрессии-в-python-a7c650ae77c2

СТАТИСТИКА КУЛЬТУРЫ И ИСКУССТВА

Раневская Анастасия Султановна
nastya_19_11@mail.ru.

Введение

Статистика культуры и искусства – одна из ветвей социальной сфер деятельности человек. Она периодически подвергается количественному анализу во всех развитых странах мира, так как играет немаловажную роль в статистической оценке государства. Культура и искусство является отраслью, отражающей деятельность человека по созданию, сохранение и потребление культурных ценностей. Она так же является одним из показателей качества жизни населения. Чем больше ресурсов, в частности, времени, денег, эмоций, готовы тратить люди на духовные ценности, тем выше считается качество жизни в стране. Таким образом оценка значимости культуры и искусства населения в стране является важной задачей современной статистики.

Целью данной работы является проведение статистический анализ показателей культуры и искусства в Российское Федерации на протяжении десяти лет в периоде с 2010 по 2019 года. Для достижения поставленной цели были сформулированы следующие задачи:

– произвести сбор статистических данных, используя соответствующий математический аппарат и программные средства;

– выполнить обработку, анализ и систематизацию информации по теме исследования;

– проанализировать динамику различных показателей.

Методологическая база исследования включает в себя статистико-социальной метод, который предполагается изучение статистических достоверных сведений. Для исследования данные были собраны с официальных статистических сайтов, таких как Федеральная служба государственной статистики, Единая межведомственная информационно-статистическая система, Сервер отраслевой статистики Минкультуры России.