Анализ и прогнозирование риска сердечных заболеваний на основе медицинских данных

Р. М. Куюков, email: renatkuyukov@gmail.com
В. В. Мокшин, email: vladimir_kgtu@mail.ru

Казанский национальный исследовательский технический университет имени А. Н. Туполева – Казанский авиационный институт

Аннотация. *Математический анализ данных и прогнозирование риска сердечных заболеваний с использованием методов машинного обучения.*

Ключевые слова: *Data Science, Machine Learning, здоровье, медицина, болезни сердца, анализ данных, прогнозирование данных.*

Введение

Термин “болезнь сердца” относится к нескольким типам сердечных заболеваний. Болезнь сердца описывает целый ряд состояний, которые влияют на сердце. Сердечные заболевания включают:

Заболевания кровеносных сосудов, такие как ишемическая болезнь сердца;

- Проблемы с сердечным ритмом (аритмии);
- Пороки сердца, при рождении (врожденные пороки сердца);
- Болезнь сердечного клапана;
- Заболевание сердечной мышцы;
- Сердечная инфекция.

Возникновение многих форм сердечных заболеваний возможно предотвратить. Необходимо проанализировать набор данных для прогнозирования сердечных заболеваний, чтобы выяснить причины и особенности, которые значительно влияют на вероятность сердечных заболеваний. Улучшение этих показателей может снизить риск возникновения сердечно-сосудистых заболеваний.

1. Описание и вывод данных

Данные будут обрабатываться в среде разработки Python. Перед работой необходимо подключить библиотеки математических функций и машинного обучения для сегментации полученных значений, а также для дальнейшего прогнозирования.

Данные представляют собой CSV-файл с 12 медицинскими атрибутами, представленные в виде текстовых и числовых значений. Ниже подробно описывается каждый из них.

Таблица 1

Атрибуты данных

Атрибут	Описание
Возраст (age)	возраст пациента [год]
Пол (sex)	пол пациента [M: Мужчина, F: Женщина]
Грудная боль (chest pain)	тип боли в груди [TA: Типичная стенокардия, ATA: Атипичная стенокардия, NAP: Нетипичная стенокардия, ASY: Бессимптомная]
Давление (Resting BP)	кровяное давление в состоянии покоя [мм рт. ст.]
Холестерин (Cholesterol)	холестерин в сыворотке крови [мм/дл]
Уровень сахара (FastingBS)	уровень сахара в крови натощак [1: если уровень сахара в крови > 120 мг/дл, 0: в противном случае]
ЭКГ (RestingECG)	результаты электрокардиограммы в состоянии покоя [Нормальный: Нормальный, ST: аномалия зубца ST-T (инверсии зубца T и/или подъем или понижение ST > 0,05 мВ), ГЛЖ: показывает вероятную или определенную гипертрофию левого желудочка по критериям Эстеса]
Макс. ЧСС (MaxHR)	максимальная достигнутая частота сердечных сокращений [Числовое значение от 60 до 202]
Стенокардия (ExerciseAngina)	стенокардия, в нагрузке [Y: Да, N: Нет]
Пиковый низкий ST (Oldpeak)	депрессия ST относится к обнаружению на электрокардиограмме, при котором след в сегменте ST аномально низок ниже базовой линии.
Макс. нагрузка ST (ST_Slope)	максимальная нагрузка ST [Вверх: наклон вверх, Плоский: плоский, вниз: наклон вниз]
Наличие сердечных заболеваний (HeartDisease)	наличие сердечных заболеваний [1: сердечные заболевания, 0: норма]

Далее подкючим csv-файл, в котором хранятся необходимые данные.

Листинг 1

```
data=pd.read_csv('../input/heart-failure-
prediction/heart.csv')
data.head()
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

Рис. 1. Описание типов структуры данных

2. Предварительная обработка данных

Предварительная обработка данных является неотъемлемым этапом машинного обучения, поскольку качество данных и полезная информация, которую можно извлечь из них, напрямую влияют на способность нашей модели к обучению.

Концепции для рассмотрения в данной статье следующие:

1. Обработка нулевых значений

В любом реальном наборе данных всегда есть несколько нулевых значений. На самом деле не имеет значения, является ли это регрессией, классификацией или любой другой проблемой, ни одна модель не может самостоятельно справиться с этими значениями NULL или NaN.

2. Масштабирование функций

На алгоритмы определения расстояний, такие как "KNN", "K-средние значения" и "SVM", в наибольшей степени влияет диапазон функций. Это происходит потому, что они используют расстояния между точками данных для определения их сходства. Когда два объекта имеют разные масштабы, есть вероятность, что больший вес будет придан объектам с большей величиной. Это повлияет на производительность алгоритма машинного обучения, и, очевидно, мы не хотим, чтобы наш алгоритм был ориентирован на одну функцию.

Следовательно, мы масштабируем наши данные, прежде чем использовать алгоритм, основанный на расстоянии, чтобы все функции в равной степени влияли на результат.

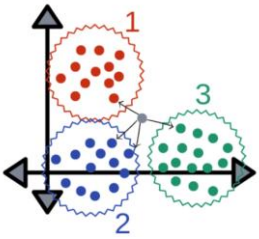

Рис. 2. Масштабирование функции

Нормализация - это метод масштабирования, при котором значения сдвигаются и масштабируются таким образом, чтобы в конечном итоге они находились в диапазоне от 0 до 1. Это также известно как минимальное-максимальное масштабирование.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Рис. 3. Формула нормализации значений

Где Xmax и Xmin являются максимальными и минимальными значениями функции соответственно, когда значение X является минимальным значением в столбце, числитель будет равен 0, и, следовательно, X’ равно 0 С другой стороны, когда значение X является максимальным значением в столбце, числитель равен знаменателю и, следовательно, значение X’ равно 1 Если значение X находится между минимальным и максимальным значением, то значение X’ находится между 0 и 1.

3. Категориальные переменные

Категориальные переменные - это любой тип объектов, который можно разделить на два основных типа:

– Номинальные переменные - это переменные, имеющие две или более категорий, с которыми не связан какой-либо порядок.

– Порядковые переменные имеют “уровни” или категории с определенным порядком, связанным с ними. Порядок важен.

Это проблема двоичной классификации: цель здесь не искажена, но мы используем наилучшую метрику для этой проблемы двоичной классификации. Следует знать, что компьютеры не понимают текстовые данные, и поэтому нам нужно преобразовать эти категории в числа.

3. Анализ и изучение входных значений

Для улучшения нашей модели необходимо удалить коррелированные переменные. Осуществим это с помощью визуализированной корреляционной матрицы. Можно найти корреляции с помощью функции pandas“.corr()” и визуализировать корреляционную матрицу с помощью plotly express.

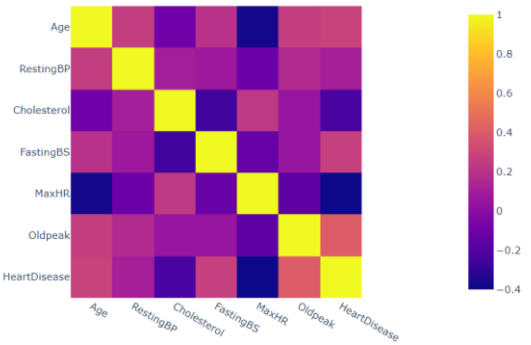

Рис. 4. Корреляционная матрица

Более светлые оттенки цветов на ячейках матрицы представляют собой положительную корреляцию, а более темные оттенки, в свою очередь, представляют собой отрицательную корреляцию.

На данном графике мы видим, что болезни сердца имеют высокую отрицательную корреляцию с "Макс.ЧСС" и отрицательную корреляцию с "Холестерином", положительная корреляция с " Пиковым низким ST ", "голоданием" и "отдыхом".

Покажем также распределение сердечных заболеваний среди мужчин и женщин.

Листинг 2

```
fig=px.histogram(df,  x="HeartDisease", color="Sex",
                 hover_data=df.columns,
                 title="Distribution of Heart Diseases",
barmode="group")
fig.show()
```

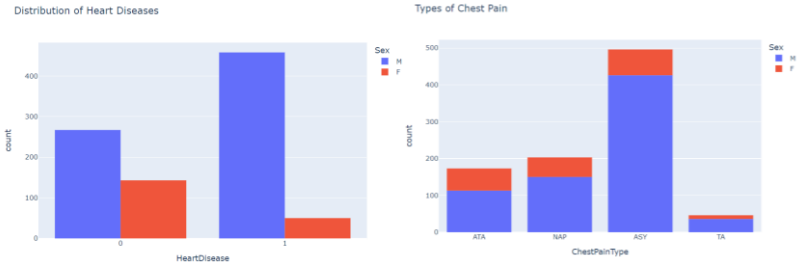

Рис. 5. Распределение сердечных заболеваний по половому признаку

Рассмотрим распределение зависимостей параметров. Для построения нескольких попарных двумерных распределений в наборе данных можно использовать функцию pair plot(). Это показывает взаимосвязь для (n, 2) комбинации переменных в кадре данных в виде матрицы графиков, а диагональные графики являются одномерными графиками.

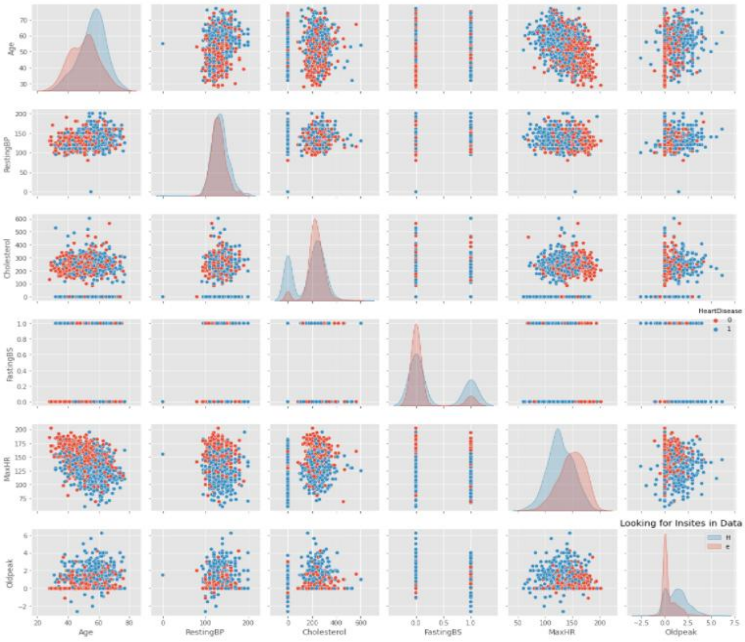

Рис. 6. Зависимость атрибутов данных

Теперь, чтобы проверить линейность переменных, построим график распределения и посмотреть на асимметрию функций. Оценка плотности ядра (kde) методом сглаживания данных является весьма полезным инструментом для построения графика формы распределения.

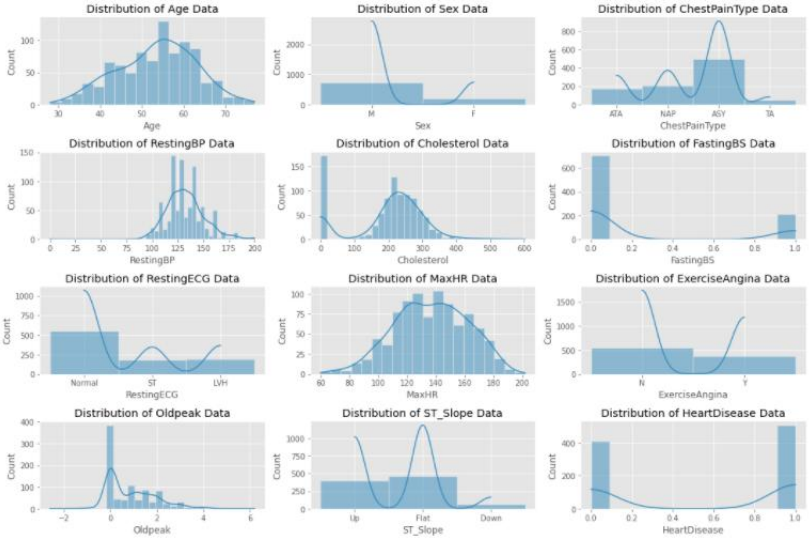

Рис. 7. Распределение атрибутов данных

4. Прогнозирование модели на основе методов машинного обучения

Будем использовать классификатор дерева решений. Дерево решений - это алгоритм контролируемого машинного обучения, который идеально подходит для задач классификации, так как он способен упорядочивать классы на точном уровне. Он работает как блок-схема, разделяя точки данных на две похожие категории одновременно от “ствола дерева” до “ветвей” и “листьев”, где категории становятся более похожими. Это создает категории внутри категорий, позволяя проводить органическую классификацию с ограниченным наблюдением со стороны человека.

Использование Random forest

Фундаментальная концепция, лежащая в основе random forest, проста, но мощна — безошибочность множества. С точки зрения науки о данных, причина, по которой модель случайного леса работает так хорошо, заключается в: большое количество относительно некоррелированных моделей (деревьев), работающих в качестве

комитета, будет превосходить любую из отдельных составляющих моделей.

Низкая корреляция между моделями является ключевой. Некоррелированные модели могут давать совокупные прогнозы, которые являются более точными, чем любые отдельные прогнозы. Причина этого эффекта заключается в том, что деревья защищают друг друга от своих индивидуальных ошибок. В то время как некоторые деревья могут быть неправильными, многие другие деревья будут правильными, так что как группа деревья способны двигаться в правильном направлении.

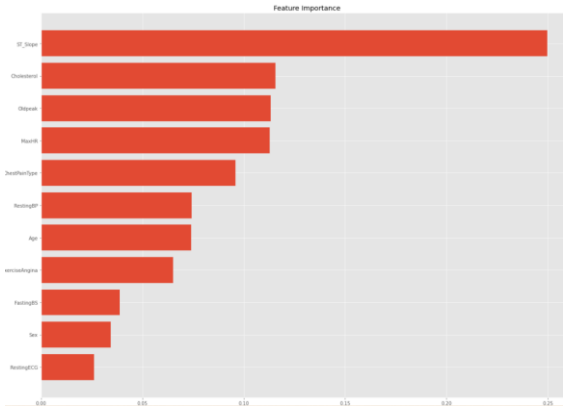

Рис. 8. График прогнозирования наиболее влиятельных атрибутов

На графике мы можем заметить, что Макс. нагрузка ST является самым значимым симптомом при сердечных заболеваниях, далее идут холестерин, пиковый низкий ST, Макс. ЧСС, Грудная боль и т.д.

Использование XGBoost

В отличие от многих других алгоритмов, XGBoost - это алгоритм ансамблевого обучения, означающий, что он объединяет результаты многих моделей, называемых базовыми учащимися, для прогнозирования.

Как и в «random forest», XGBoost использует деревья решений в качестве базового метода.

Однако следует заметить, что деревья, используемые XGBoost, немного отличаются от традиционных деревьев решений. Они называются деревьями «корзин» (деревья классификации и регрессии) и вместо того, чтобы содержать одно решение в каждом конечном узле,

они содержат оценки реального значения того, принадлежит ли экземпляр к группе. После того, как дерево достигнет максимальной глубины, решение может быть принято путем преобразования баллов в категории с использованием определенного порога.

Результатом программы является схема в виде дерева, где листья будут обозначать вероятность сердечного заболевания, а корни, которые ведут к листьям, условия атрибутов. Переход к атрибуту происходит при выполнении или не выполнении условия (да или нет), следовательно, каждому варианту соответствует свой путь (ребро графа).

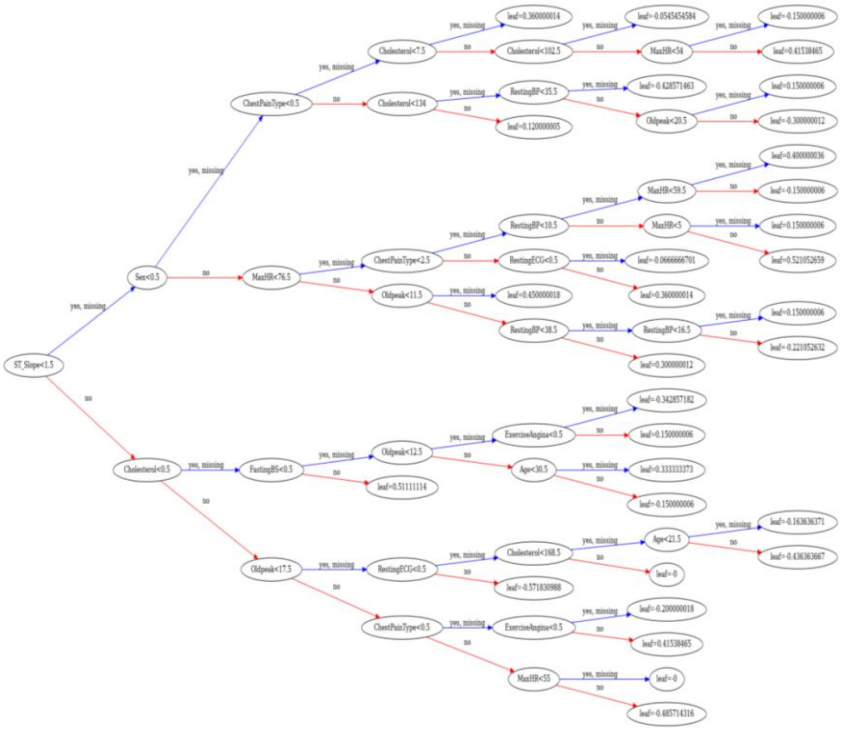

Рис. 9. Прогнозирование сердечных заболеваний с учетом всех полученных данных

В схеме рассмотрены всевозможные наборы атрибутов, поэтому мы можем оценить возможные риски любого пациента.

Листинг 3

```
from xgboost import XGBClassifier
```

```
acc_XGB=[]
kf=model_selection.StratifiedKFold(n_splits=5)
for fold , (trn_,val_) in enumerate(kf.split(X=df_tree,y=y)):
    X_train=df_tree.loc[trn_,feature_col_tree]
    y_train=df_tree.loc[trn_,target]
    X_valid=df_tree.loc[val_,feature_col_tree]
    y_valid=df_tree.loc[val_,target]
    clf=XGBClassifier()
    clf.fit(X_train,y_train)
    y_pred=clf.predict(X_valid)
    print(f"The fold is : {fold} : ")
    print(classification_report(y_valid,y_pred))
    acc=roc_auc_score(y_valid,y_pred)
    acc_XGB.append(acc)
    print(f"The accuracy for {fold+1} : {acc}")
fig, ax = plt.subplots(figsize=(30, 30))
from xgboost import plot_tree
plot_tree(clf,num_trees=0,rankdir="LR",ax=ax)
plt.show()
```

Заключение

В данной статье был осуществлен анализ медицинской базы данных, и прогнозирование риска сердечных заболеваний с использованием машинного обучения. В итоге проделанной работы мы выявили причины и особенности, которые значительно влияют на вероятность болезней сердца. Улучшение этих показателей могут помочь людям сохранить свое здоровье, а также позволит медицинским работникам более подробно понять причины заболевания.

Список литературы

1. Нестеров, С. А. Базы данных. Интеллектуальный анализ данных : учеб. пособие / С. А. Нестеров – СПб. : Изд-во Политехн. ун-та, 2011. – 272 с.
2. База данных: Heart Failure Prediction Dataset - 11 clinical features for predicting heart disease events. [Электронный ресурс]. – Режим доступа : https://www.kaggle.com/fedesoriano/heart-failure-prediction
3. Мокшин, В. В. Метод формирования модели анализа сложной системы / В.В. Мокшин, И.М. Якимов // Информационные технологии. – 2011. – № 5. – С. 46-51.
4. Мокшин, В. В. Рекурсивный алгоритм построения регрессионных моделей сложных вероятностных объектов / В. В. Мокшин, И. Р. Сайфудинов, А. П. Кирпичников // Вестник Технологического университета. – 2017. – Т. 20. – №9. – С. 112-116.