

Разработка методов машинного обучения для анализа и прогнозирования сердечного приступа

Языков Кирилл Александрович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Целью исследования является применение различных методов машинного обучения для построения модели с целью прогнозирования сердечного приступа. В результате исследования выбрана модель и переменные, обеспечивающие лучшие результаты для текущей задачи. Сравнение моделей произведено на основании основных метрик: AUC, F1, Precision, Recall.

Ключевые слова: модель обучения, датасет, сердечный приступ.

Development of a monitor price prediction system based on machine learning technologies

Yazykov Kirill Alexandrovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The aim of the study is to apply various machine learning methods to post-process the model in order to predict a heart attack. As a result of the research, a model and variables were selected that give the best results for the current task. The comparison of models is based on the main metrics: AUC, F1, Precision, Recall.

Keywords: learning model, dataset, heart attack.

1 Введение

1.1 Актуальность

В современном мире объемы данных растут в геометрической прогрессии, и умение извлекать ценную информацию становится все более критическим для успешного функционирования сферы медицины. Одной из важнейших задач в этом контексте является анализ и понимание факторов, влияющих на прогнозирование сердечного приступа. От решения данной задачи могут зависеть жизни множества людей

1.2 Обзор исследований

Е. М. Аксютин и Ю. С. Белов в своей статье рассматривают основные архитектуры, применяемые для анализа больших данных, их особенности и ограничения, а также выявляются требования к методам машинного обучения, выполнение которых позволит применять их при анализе больших данных [1].

В работе «Применение методов машинного обучения для прогнозирования сердечного приступа» описывается применение методов машинного обучения для прогнозирования сердечного приступа [2]. Автор «Прогнозирование сердечного приступа с помощью технологий глубокого обучения» представляет модель, основанную на искусственных нейронных сетях, которая способна предсказывать вероятность возникновения приступа в течение ближайших 48 часов. Модель была обучена на данных пациентов с инфарктом миокарда и показала высокую точность прогнозов [3]. В статье «Искусственный интеллект в медицине» раскрывается тема использования искусственного интеллекта в медицинской диагностике и прогнозировании заболеваний, рассматриваются различные методы и подходы, основанные на искусственном интеллекте, которые применяются для анализа медицинских данных, выявления закономерностей и создания прогностических моделей [4]. Статья «Прогнозирование заболеваний с помощью шкал» помогает рассмотреть преимущества и недостатки прогностических шкал, предлагает новую классификацию и формулирует принципы практического использования шкал. Также обсуждаются возможности и перспективы разработки более точных моделей прогноза заболеваний с использованием математических инструментов [5].

1.3 Цель исследования

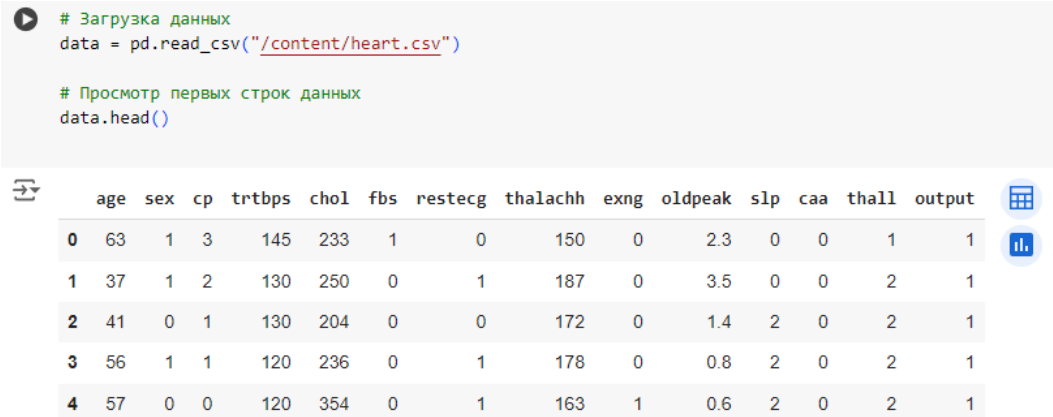
Целью исследования является применение различных методов машинного обучения для построения модели с целью прогнозирования сердечного приступа.

2 Материалы и методы

Для реализации был использован язык программирования Python, средой для работы послужил сервис GoogleColab. Датасет был взят из социальной сети Kaggle [6]. Полное проведенное исследование также доступно по открытой ссылке [7].

3 Результаты исследования

Первым делом следует загрузить данные в GoogleColab (Рис. 1, Рис. 2).



The image shows a Google Colab notebook interface. At the top, there is a code cell with the following Python code:

```
# Загрузка данных
data = pd.read_csv("/content/heart.csv")

# Просмотр первых строк данных
data.head()
```

Below the code cell, the output of the `data.head()` command is displayed as a table. The table has 15 columns: `age`, `sex`, `cp`, `trtbps`, `chol`, `fbs`, `restecg`, `thalachh`, `exng`, `oldpeak`, `slp`, `caa`, `thall`, and `output`. The first five rows of data are shown, indexed from 0 to 4.

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Рисунок 1 – Фрагмент таблицы с данными

```
# Статистическое описание
# print(data.describe())
data.describe()
```

	age	sex	cp	trtbps	chol	fb	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

Рисунок 2 – Статистическое описание

Схематически зарисуем взаимосвязь между возрастом и уровнем холестерина (Рис. 3), артериальным давлением (Рис. 4), частотой сердечных сокращений (Рис. 5), а также риском сердечного приступа в соответствии с данными показателями

```
#уровень холестерина в зависимости от возраста
plt.figure(figsize=(10, 6))
sns.scatterplot(x='age', y='chol', hue='output', data=data, alpha=0.6)
plt.title('Взаимосвязь между холестерином и возрастом')
plt.xlabel('Возраст')
plt.ylabel('Холестерин')
plt.legend(title='Output')
plt.show()
```

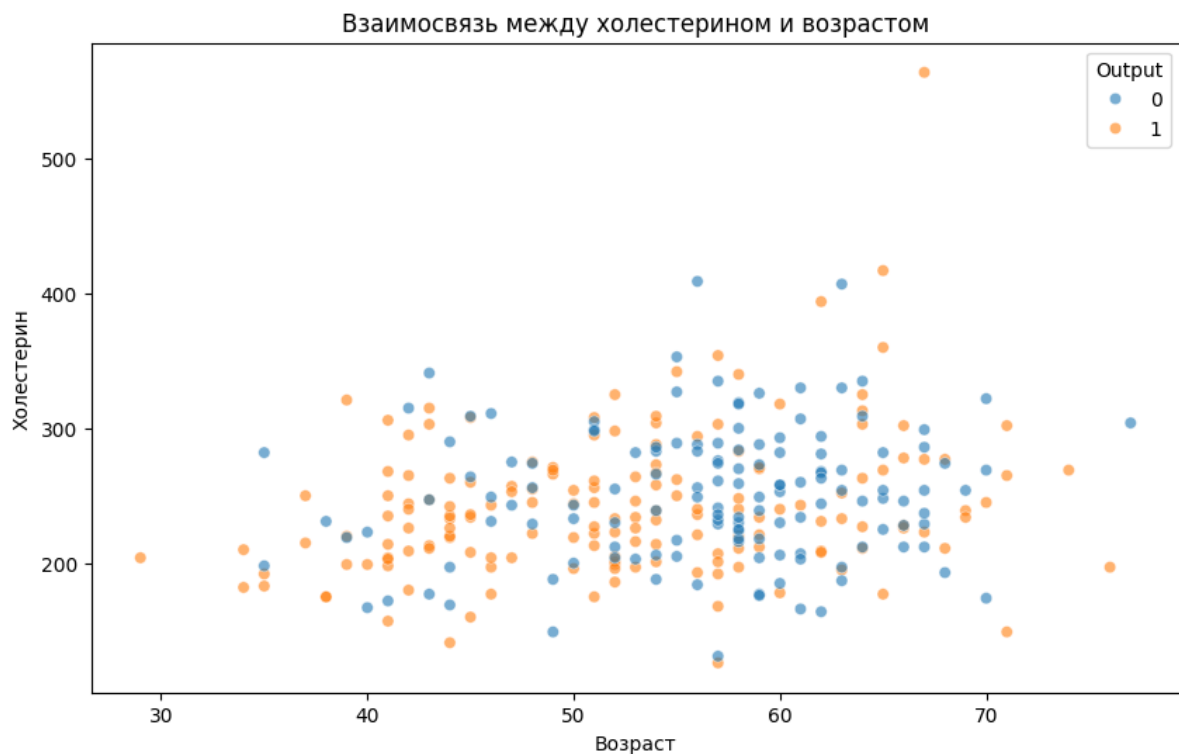


Рисунок 3 – Уровень холестерина в зависимости от возраста

```
#уровень артериального давления в зависимости от возраста
plt.figure(figsize=(10, 6))
sns.scatterplot(x='age', y='trtbps', hue='output', data=data, alpha=0.6)
plt.title('Взаимосвязь между артериальным давлением и возрастом')
plt.xlabel('Возраст')
plt.ylabel('Давление')
plt.legend(title='Output')
plt.show()
```



Рисунок 4 – Уровень артериального давления в зависимости от возраста

```
#уровень частоты сердечных сокращений в зависимости от возраста
plt.figure(figsize=(10, 6))
sns.scatterplot(x='age', y='thalachh', hue='output', data=data, alpha=0.6)
plt.title('Взаимосвязь между частотой сердечных сокращений и возрастом')
plt.xlabel('Возраст')
plt.ylabel('Сокращения')
plt.legend(title='Output')
plt.show()
```

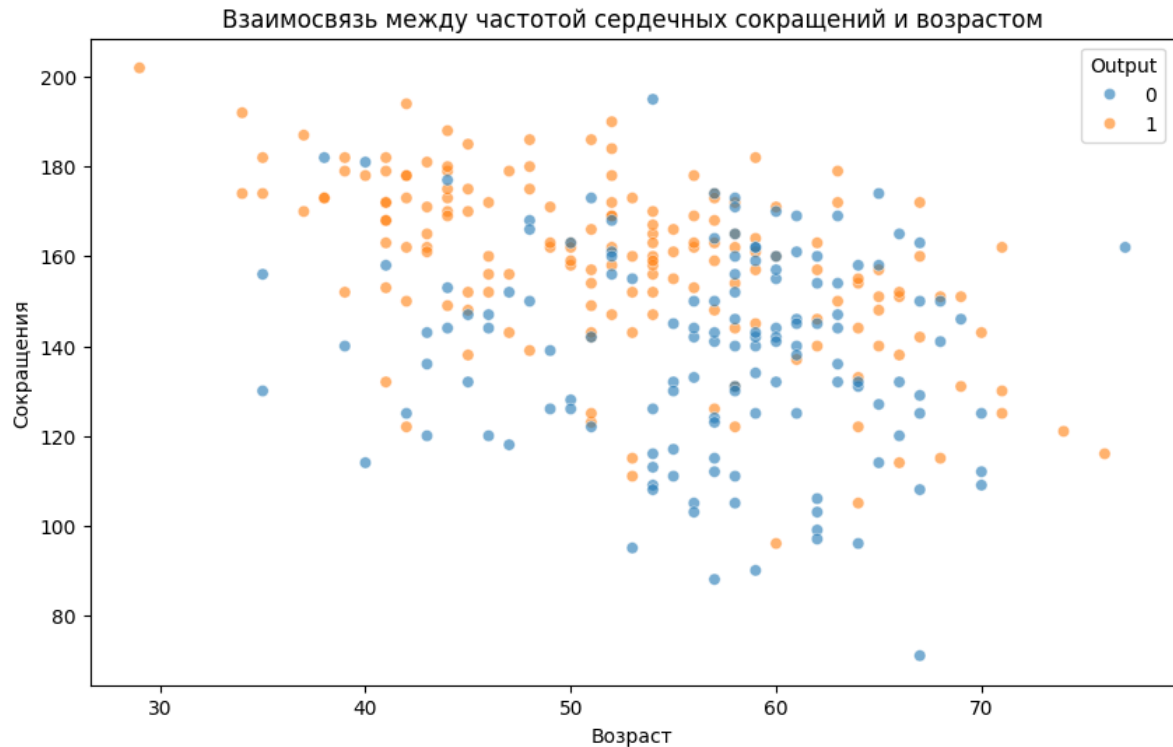


Рисунок 5 – Уровень частоты сердечных сокращений в зависимости от возраста

Затем выполняем предобработку данных для моделирования, разделение на признаки и целевую переменную, а также на обучающую и тестовую выборки (Рис. 6).

```
[316] # Разделение на признаки и целевую переменную
# За основу возьмем такие признаки, как возраст, холестерин, давление, сердечные сокращения и талий
X = data[['age', 'chol', 'trtbps', 'thalachh', 'thall']]
y = data['output']

[317] # Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Рисунок 6 – Деление данных на обучающую и тестовую выборки

Далее проведем масштабирование признаков (Рис. 7).

```
[318] # Масштабирование признаков
      scaler = StandardScaler()
      X_train = scaler.fit_transform(X_train)
      X_test = scaler.transform(X_test)

[319] models = {
      'KNN': KNeighborsClassifier(),
      'Random Forest': RandomForestClassifier(),
      'Decision Tree': DecisionTreeClassifier(),
      'Logistic Regression': LogisticRegression(),
      'XGBoost': xgb.XGBClassifier(use_label_encoder=False, eval_metric='logloss')
    }

[320] results = {}
```

Рисунок 7 – Масштабирование признаков

Выполняем обучение модели, вывод метрик и анализируем результаты (Рис. 8, Рис. 9).

```
for model_name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    # Вычисление метрик
    auc = roc_auc_score(y_test, model.predict_proba(X_test)[: , 1])
    f1 = f1_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)

    results[model_name] = {
        'AUC': auc,
        'F1': f1,
        'Precision': precision,
        'Recall': recall,
        'Confusion Matrix': confusion_matrix(y_test, y_pred)
    }
```

Рисунок 8 – Обучение модели

```
for model_name, metrics in results.items():
    print(f"Model: {model_name}")
    print(f"AUC: {metrics['AUC']:.4f}, F1: {metrics['F1']:.4f}, Precision: {metrics['Precision']:.4f}, Recall: {metrics['Recall']:.4f}")
    print("Confusion Matrix:")
    print(metrics['Confusion Matrix'])
    print()
```

```
Model: KNN
AUC: 0.8669, F1: 0.7931, Precision: 0.8846, Recall: 0.7188
Confusion Matrix:
[[26  3]
 [ 9 23]]
```

```
Model: Random Forest
AUC: 0.8858, F1: 0.8000, Precision: 0.9565, Recall: 0.6875
Confusion Matrix:
[[28  1]
 [10 22]]
```

```
Model: Decision Tree
AUC: 0.7107, F1: 0.6786, Precision: 0.7917, Recall: 0.5938
Confusion Matrix:
[[24  5]
 [13 19]]
```

```
Model: Logistic Regression
AUC: 0.8685, F1: 0.8438, Precision: 0.8438, Recall: 0.8438
Confusion Matrix:
[[24  5]
 [ 5 27]]
```

```
Model: XGBoost
AUC: 0.8578, F1: 0.7586, Precision: 0.8462, Recall: 0.6875
Confusion Matrix:
[[25  4]
 [10 22]]
```

Рисунок 9 – Вывод метрик

Из полученных данных следует, что модель RandomForest по метрике Precision показал наиболее точный результат.

Далее, визуализируем важность выбранных признаков (Рис. 10).

```
for model_name in ['Random Forest', 'XGBoost']:
    model = models[model_name]
    if model_name == 'Random Forest':
        importances = model.feature_importances_
    else:
        importances = model.feature_importances_

    # Визуализация важности признаков
    plt.figure(figsize=(8, 6))
    plt.barh(X.columns, importances)
    plt.title(f'Feature Importance for {model_name}')
    plt.xlabel('Importance')
    plt.show()
```

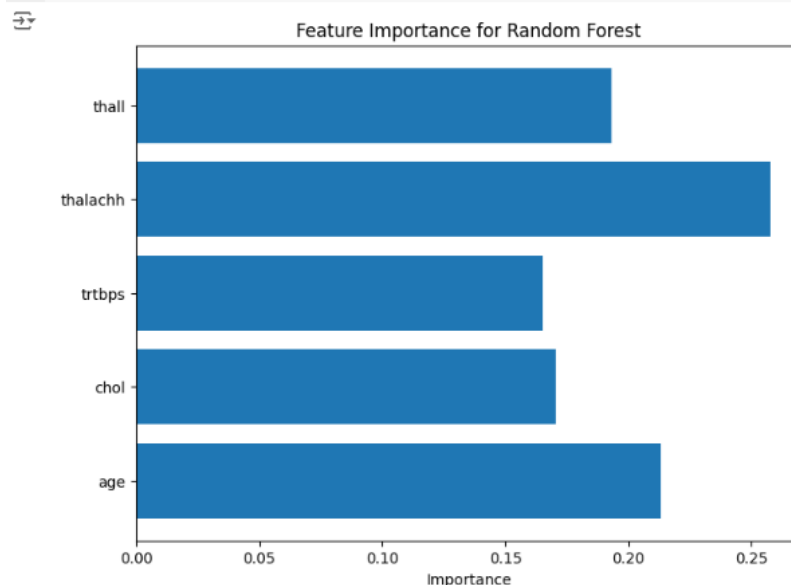


Рисунок 10 – Визуализация важности признаков

Исходя из графика можно сделать вывод, что в модели RandomForest признак *thalachh* (максимальная частота сердечных сокращений) наиболее сильно влияет на вероятность прогнозирования сердечного приступа. Наименьшее же влияние оказывает признак *trtbps* (уровень артериального давления в покое). Выводим итоговую таблицу (Таблица 1).

Таблица 1. Сравнение точности предугадывания результата

Метод	AUC	F1	Precision	Recall
KNN	0.8669	0.7931	0.8846	0.7188
Random Forest	0.8858	0.8000	0.9565	0.6875
Decision Tree	0.7107	0.6786	0.7917	0.5938
Logistic Regression	0.8685	0.8438	0.8438	0.8438
XGBoost	0.8578	0.7586	0.8462	0.6875

Выводы

Таким образом, в результате проведенного анализа и экспериментов, на основании метрики F1 можно рекомендовать использовать модель Logistic Regression для прогнозирования сердечного приступа. Однако, для дальнейшего улучшения модели, возможно, потребуется дополнительное исследование и настройка параметров.

Библиографический список

1. Аксютин Е. М., Белов Ю. С. Обзор архитектур и методов машинного обучения для анализа больших данных // Электронный журнал: наука, техника и образование. 2016. №. 1. С. 134-141.
2. Старцева О. Г., Ключа Д. И. Применение методов машинного обучения для прогнозирования сердечного приступа // Вестник Башкирского государственного педагогического университета им. М. Акмуллы, 2023. 141-145 с.
3. Волынкин В. А. Прогнозирование сердечного приступа с помощью технологий глубокого обучения // Математика и математическое моделирование. 2019. С. 333-334.
4. Беялов Ф. И. Прогнозирование заболеваний с помощью шкал // Комплексные проблемы сердечно-сосудистых заболеваний. 2018. Т. 7. №. 1. С. 84-93.
5. Датасет для исследования // Kaggle URL: <https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset/data>
6. Ноутбук с исследованием // GoogleColab URL: <https://colab.research.google.com/drive/1pSA7znk-3CLDL86CdSHe0j3aJ3nnZxHP>