

Министерство образования и науки РФ
Санкт-Петербургский Политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа искусственного интеллекта

**Построение портретов пациентов с инфарктом при различных
типах рака в разрезе выживаемости**

по дисциплине «Решение прикладных задач с применением методов
машинного обучения»

Выполнили:

студентка гр. 3540201/20301

_____ Климова О. А.

подпись, дата

Проверил:

д.т.н.

_____ Заборовский В. С.

подпись, дата

Санкт-Петербург

2024

Содержание

Введение.....	3
1 Постановка задачи.....	4
2 Описание данных	5
3 Описание модели обучения.....	7
3.1 Бинарная классификация.....	7
3.2 Полносвязная нейронная сеть.....	8
3.3 SHAP.....	9
4 Предобработка данных	11
5 Реализация модели обучения.....	13
5.1 Корреляция параметров.....	13
5.2 TargetEncoder	14
5.3 Полносвязная нейронная сеть.....	14
5.4 SHAP.....	17
6 Построение портретов пациентов.....	19
Заключение	20

Введение

В современном здравоохранении становится всё более ясным, что персонализированный подход к лечению играет ключевую роль в достижении успешных результатов. Комбинированные заболевания, такие как инфаркт и рак, представляют серьезные вызовы для медицинского сообщества, требуя инновационных подходов к анализу и прогнозированию.

Данная работа фокусируется на построении портретов пациентов с инфарктом, учитывая различные типы рака, исследуя их выживаемость. Цель нашей работы заключается в разработке модели машинного обучения, способной предсказывать вероятность смерти или выживания пациента на основе анализов, проводимых при его поступлении в медицинское учреждение. Мы стремимся создать эффективный инструмент для предсказания и ранней диагностики, который основывается на комплексном анализе медицинских данных.

Для достижения цели работы будут использованы методы машинного обучения, включая алгоритмы классификации и объяснительный интеллект. Построение модели машинного обучения позволит выявить закономерности и корреляции между различными параметрами пациентов и их исходами. Анализ таких данных поможет нам выделить ключевые факторы, влияющие на результаты лечения, и определить их влияние на вероятность выживания.

Эта работа не только способствует повышению точности диагностики, но также может быть полезной для оптимизации медицинских процессов, направленных на предоставление персонализированного и эффективного лечения для пациентов с инфарктом. Мы стремимся к созданию инновационного инструмента, который поможет медицинским специалистам принимать более обоснованные решения и улучшать прогнозы для пациентов, поступающих в медицинское учреждение.

1 Постановка задачи

Необходимо, используя модель машинного обучения, построить портреты типичных пациентов с инфарктом при различных типах рака в разрезе выживаемости на основе результатов анализов, проводимых при поступлении в медицинское учреждение.

Для этого необходимо:

- получить набор данных из ГБУ Санкт-Петербургского научно-исследовательского института скорой помощи им. И.И. Джанелидзе, включающий информацию о пациентах с инфарктом, учитывая тип рака, а также результаты медицинских анализов;
- провести тщательную предобработку данных, включая обработку пропущенных значений, стандартизацию и нормализацию числовых признаков, а также кодирование категориальных переменных;
- для реализации бинарной классификации, где целью является предсказание выживаемости пациента, использовать полносвязную нейронную сеть, обеспечивающую учет сложных взаимосвязей между характеристиками пациентов;
- обучить модель машинного обучения и выделить ключевые характеристики, влияющие на выживаемость, а для объяснения вклада каждого параметра в прогноз использовать shap (SHapley Additive exPlanations);
- оценить качество модели, используя метрики классификации, такие как точность и F1-мера;
- получить портреты пациентов, выделяя ключевые медицинские параметры и анализируя их влияние на прогноз выживаемости при инфаркте в сочетании с различными типами рака.

2 Описание данных

Данные были получены из ГБУ Санкт-Петербургского научно-исследовательского института скорой помощи им. И.И. Джанелидзе в виде эксель таблицы, фрагмент которой представлен на рисунке 2.1.

пол	возраст	Вид рака	Активность	Тип	метастазы	операция	лучевая терапия	химиотерапия	поступил	смертная дата
Мужчина	70	ЗНО пищеварительной системы	Перенесенный рак		отсутствует	перенес	отсутствует	отсутствует	03/11/201	выжившие
Женщина	72	ЗНО репродуктивной системы	Перенесенный рак	1	отсутствует	перенес	отсутствует	отсутствует	05/28/201	выжившие
Мужчина	83	ЗНО соединительной ткани+меланома	Активный рак		наличие	без опер	отсутствует	отсутствует	04/07/201	выжившие
Женщина	84	ЗНО репродуктивной системы	Перенесенный рак		отсутствует	перенес	наличие	наличие	01/06/201	выжившие
Женщина	84	ЗНО эндокринной системы	Перенесенный рак		отсутствует	перенес	наличие	наличие	01/06/201	выжившие
Женщина	84	ЗНО пищеварительной системы	Перенесенный рак		отсутствует	перенес	наличие	наличие	01/06/201	выжившие
Мужчина	58	ЗНО дыхательной системы	Перенесенный рак		отсутствует	перенес	отсутствует	отсутствует	02/03/201	умершие
Женщина	57	ЗНО пищеварительной системы	Активный рак		отсутствует	без опер	отсутствует	отсутствует	02/04/201	умершие
Мужчина	77	ЗНО репродуктивной системы	Перенесенный рак		отсутствует	перенес	отсутствует	отсутствует	02/08/201	умершие
Женщина	65	ЗНО репродуктивной системы	Активный рак		наличие	без опер	отсутствует	отсутствует	02/18/201	умершие
Мужчина	78	ЗНО пищеварительной системы	Активный рак		наличие	без опер	отсутствует	отсутствует	06/10/201	выжившие
Мужчина	71	ЗНО пищеварительной системы	Перенесенный рак		отсутствует	перенес	отсутствует	отсутствует	02/14/201	умершие
Мужчина	70,8	ЗНО дыхательной системы	Перенесенный рак		наличие	без опер	отсутствует	наличие	01/25/201	выжившие
Женщина	68	ЗНО пищеварительной системы	Перенесенный рак		отсутствует	перенес	наличие	отсутствует	05/17/201	выжившие
Женщина	68	ЗНО репродуктивной системы	Перенесенный рак		отсутствует	перенес	наличие	отсутствует	05/17/201	выжившие
Женщина	68	ЗНО пищеварительной системы	Перенесенный рак		отсутствует	перенес	наличие	отсутствует	05/17/201	выжившие
Женщина	68	ЗНО пищеварительной системы	Перенесенный рак		отсутствует	перенес	наличие	отсутствует	05/17/201	выжившие
Женщина	78	ЗНО репродуктивной системы	Перенесенный рак		отсутствует	перенес	отсутствует	отсутствует	11/29/201	выжившие
Женщина	73	ЗНО репродуктивной системы	Перенесенный рак		отсутствует	перенес	наличие	наличие	03/08/201	выжившие

Рисунок 2.1 – Таблица с данными

Данная таблица включает информацию о пациентах с инфарктом, типе рака, их выживаемости, а также результатах медицинских анализов в виде 385 параметров для 356 пациентов.

Параметры, представленные на рисунке 2.1:

- пол (мужчина/женщина);
- возраст (50–100);
- вид рака (ЗНО пищеварительной системы/ ЗНО репродуктивной системы/ ЗНО соединительной ткани+меланома/ ЗНО эндокринной системы/ ЗНО дыхательной системы);
- активность (перенесенный рак/активный рак);
- тип (0/ 1/ 4/ T2N1M1/ 2,1,1/ T4NxMx/ T4N1M1/ T3N1M1/ T4N1M0/ TхТхх0/ T2N0M0/ TхMхN0/ T3N1M1/ T1NхM1/ T3NхMх/ T3N0M1/ T3N0M0/ T2N2bM0/ T2N1M1/ T4NxMx);
- метастазы (наличие/отсутствие);
- операция (без операции/перенесенная операция);
- лучевая терапия (наличие/отсутствие);

- химиотерапия (наличие/отсутствие);
- поступление (дата);
- госпитальная летальность (выжившие/умершие).

Цель анализа данных будет заключаться в выявлении взаимосвязей между указанными параметрами и выживаемостью пациентов. Предварительная обработка данных будет ключевым этапом для эффективного построения модели машинного обучения и извлечения значимых выводов для медицинской практики.

3 Описание модели обучения

3.1 Бинарная классификация

Бинарная классификация представляет собой тип задачи машинного обучения, где модель обучается разделять объекты на два класса. Каждому объекту в обучающей выборке присваивается одна из двух меток - "положительный" или "отрицательный" класс (рис. 3.1.1). Целью бинарной классификации является построение алгоритма, способного правильно отнести новые, ранее не встреченные объекты к одному из указанных классов.

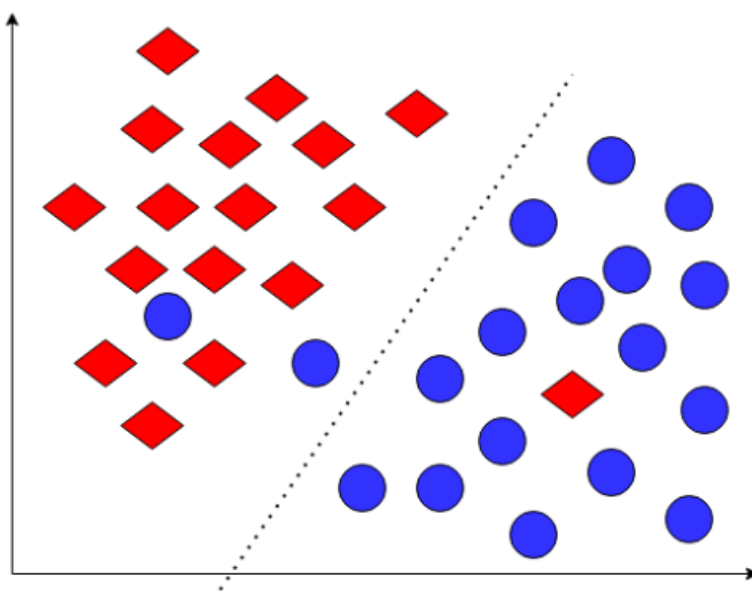


Рисунок 3.1.1 – Бинарная классификация

Целевой переменной в нашей задаче является бинарная метка выживаемости. Каждому пациенту присваивается метка "1" (выжил) или "0" (не выжил). Все параметры, представленные в медицинских анализах, включая пол, возраст, вид рака, активность, тип и т. д., служат признаками для обучения модели. Для решения задачи бинарной классификации в данной работе планируется использовать полносвязную нейронную сеть, которая способна учесть сложные взаимосвязи между различными характеристиками пациентов.

3.2 Полносвязная нейронная сеть

Полносвязная нейронная сеть (Fully Connected Neural Network), также известная как многослойный перцептрон (Multilayer Perceptron, MLP), представляет собой тип искусственной нейронной сети, состоящей из нескольких слоев нейронов, где каждый нейрон в одном слое связан с каждым нейроном в следующем слое. Эти связи образуют полносвязные слои, что означает, что каждый нейрон в одном слое соединен со всеми нейронами в следующем слое (рис. 3.2.1).

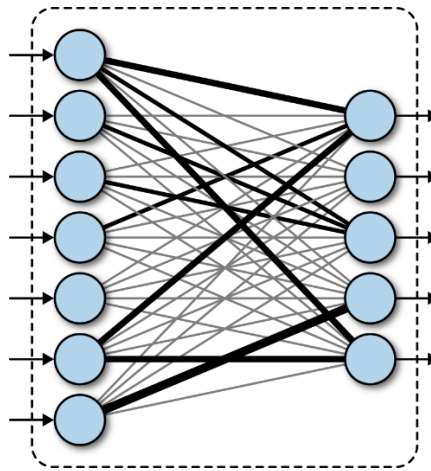


Рисунок 3.2.1 – Полносвязная нейронная сеть

Архитектура полносвязной нейронной сети включает:

- **входной слой** – нейроны в этом слое представляют входные признаки, количество нейронов в этом слое соответствует размерности вектора признаков;
- **скрытые слои** – "скрытые" вычислительные слои, в которых каждый нейрон связан со всеми нейронами предыдущего и следующего слоя, количество и размерность этих слоев являются параметрами модели и обычно определяются экспериментальным путем;
- **выходной слой** – представляет собой выход модели, количество нейронов в выходном слое зависит от числа классов или задачи, которую мы решаем, для бинарной классификации может – один нейрон.

Для каждого нейрона в скрытом или выходном слое формула прямого прохода имеет вид:

$$z_j = \sum_i^n (\omega_{ij} \cdot x_i) + b_j, \quad 3.2.1$$

где z_j – взвешенная сумма,

x_i – входной признак,

ω_{ij} – вес связи между нейронами i и j ,

b_j – смещение нейрона j .

Далее значение z_j подается в функцию активации, например, сигмоиду и тогда предсказанная вероятность принадлежности объекта к положительному классу будет описываться формулой:

$$P(y = 1|x) = \sigma(z_j) = \frac{1}{1 + e^{-z_j}}, \quad 3.2.2$$

где z_j – взвешенная сумма,

$\sigma(z_j)$ – сигмоидная функция активации.

Если вероятность превышает 0.5, то объект относится к положительному классу (1), а в противном случае он относится к отрицательному классу (0).

Таким образом, полносвязная нейронная сеть применяет последовательность операций линейных преобразований и нелинейных функций активации для извлечения сложных зависимостей из данных.

Для интерпретации результатов бинарной классификации может быть использован метод SHAP (SHapley Additive exPlanations), который позволяет понять вклад каждого признака в принятое решение

3.3 SHAP

SHAP (SHapley Additive exPlanations) – это метод интерпретации моделей машинного обучения, разработанный на основе теории кооперативных игр и концепции вклада Шепли. Он предоставляет объяснения для предсказаний модели, позволяя понимать влияние каждого признака на конечный результат.

SHAP аддитивен, что означает, что объяснение предсказания модели представляется в виде суммы вкладов каждого признака. Локальные

объяснения для конкретных примеров данных позволяют понять влияние признаков на предсказания для конкретных наблюдений.

Этот метод применим к различным моделям машинного обучения и может быть использован для отладки, оптимизации признаков и повышения доверия к модели. Библиотека `shap` в Python предоставляет инструменты для расчета и визуализации значений, полученных в результате работы объяснительной модели, делая процесс объяснения более доступным и интерпретируемым.

4 Предобработка данных

После консультации с врачом кардиологом были сделаны выводы о нецелесообразности использования некоторых критериев для обучения. Например, параметры 'срок наступления события' и 'койкодень' описывают через сколько дней пациент выздоровел и сколько из них он провел в больнице. Так как мы хотим определить умрет человек или выживет на момент поступления в больницу, то данные параметры не нужны для обучения модели.

По ряду других причин были исключены из рассмотрения еще 12 параметров: 'тип2', 'дни в орит', 'дата поступления', 'Код', 'п4т госпит', 'Отд_лет', 'Комп.номер', '№ иб', 'фамилия', 'Стентир4аниесейчас', 'эозинофилы4', 'Ишемия'. В коде данный процесс был реализован следующим образом:

```
not_to_use = ['тип2', 'дни в орит', 'дата поступления', 'Код', 'п4т госпит', 'Отд_лет',  
'Комп.номер', '№ иб', 'фамилия', 'Стентир4аниесейчас', 'эозинофилы4', 'Ишемия',  
'срок наступления события', 'койкодень']  
  
use_col = [i for i in use_col1 if i not in not_to_use]
```

Из набора данных также были исключены пустые колонки, а также колонки, заполненные одним значением:

```
columns = list(df.columns)  
  
len_0 = []  
len_1 = []  
  
for i in columns:  
    if len(df[i].value_counts()) == 0:  
        len_0.append(i)  
    if len(df[i].value_counts()) == 1:  
        len_1.append(i)  
  
df.drop(len_0 + len_1, axis=1, inplace=True)
```

После удаления неинформативных для обучения колонок, был осуществлен отбор колонок, в которых заполнено хотя бы 150 ячеек:

```
use_col1 = list(df.columns[df.count().values >= 150])
```

Далее в отобранных колонках заполнялись пропущенные значения.

Числовые не полностью заполненные колонки были заполнены средними значениями:

```
data.loc[data['Суммарный стеноз левой коронарной артерии'] == '0%', 'Суммарный стеноз  
левой коронарной артерии'] = 0  
data.loc[data['Суммарный стеноз левой коронарной артерии'] == '50-74%', 'Суммарный  
стеноз левой коронарной артерии'] = 62  
data.loc[data['Суммарный стеноз левой коронарной артерии'] == '1-49%', 'Суммарный  
стеноз левой коронарной артерии'] = 25  
data.loc[data['Суммарный стеноз левой коронарной артерии'] == '75-90%', 'Суммарный  
стеноз левой коронарной артерии'] = 83.5  
data.loc[data['Суммарный стеноз левой коронарной артерии'] == '91-96%', 'Суммарный  
стеноз левой коронарной артерии'] = 93.5  
data.loc[data['Суммарный стеноз левой коронарной артерии'] == '97-100%', 'Суммарный  
стеноз левой коронарной артерии'] = 98.5  
  
data.loc[data['Бифуркационный стеноз левой коронарной артерии'] == '0%',  
'Бифуркационный стеноз левой коронарной артерии'] = 0  
data.loc[data['Бифуркационный стеноз левой коронарной артерии'] == '50-74%',  
'Бифуркационный стеноз левой коронарной артерии'] = 62  
data.loc[data['Бифуркационный стеноз левой коронарной артерии'] == '1-49%',  
'Бифуркационный стеноз левой коронарной артерии'] = 25  
data.loc[data['Бифуркационный стеноз левой коронарной артерии'] == '91-96%',  
'Бифуркационный стеноз левой коронарной артерии'] = 93.5  
data.loc[data['Бифуркационный стеноз левой коронарной артерии'] == '97-100%',  
'Бифуркационный стеноз левой коронарной артерии'] = 98.5  
  
data.loc[data['Суммарный стеноз ПМЖВ'] == '0%', 'Суммарный стеноз ПМЖВ'] = 0  
data.loc[data['Суммарный стеноз ПМЖВ'] == '50-74%', 'Суммарный стеноз ПМЖВ'] = 62  
data.loc[data['Суммарный стеноз ПМЖВ'] == '1-49%', 'Суммарный стеноз ПМЖВ'] = 25  
data.loc[data['Суммарный стеноз ПМЖВ'] == '75-90%', 'Суммарный стеноз ПМЖВ'] = 83.5  
и т. д.
```

```
data.fillna(data.mean(), inplace=True)
```

Из-за человеческого фактора в таблице данных имели место опечатки, в соответствии с которыми также осуществлялось заполнение пустых ячеек:

```
data.loc[data['тропонин'] == '0,05-0,1', 'тропонин'] = 0.075  
data.loc[data['тропонин'] == '0,05-0,01', 'тропонин'] = 0.075
```

Категориальные столбцы были заполнены наиболее часто встречающимися элементами:

```
data.iloc[cat_col_id] = data.iloc[cat_col_id].apply(lambda x: x.fillna(x.value_counts().index[0]))
```

5 Реализация модели обучения

5.1 Корреляция параметров

В результате предобработки данных осталось 216 колонок, на основе которых будет производиться дальнейшая работа. Перед построением модели машинного обучения были рассмотрены корреляции параметров (рис. 5.1.1).

Рисунок 5.1.1 показывает несильную корреляцию параметров. Коррелированными являются лишь параметры АСТ и АЛТ. Оба данных параметра описывают функционирование печени, поэтому их корреляция вполне обоснована.

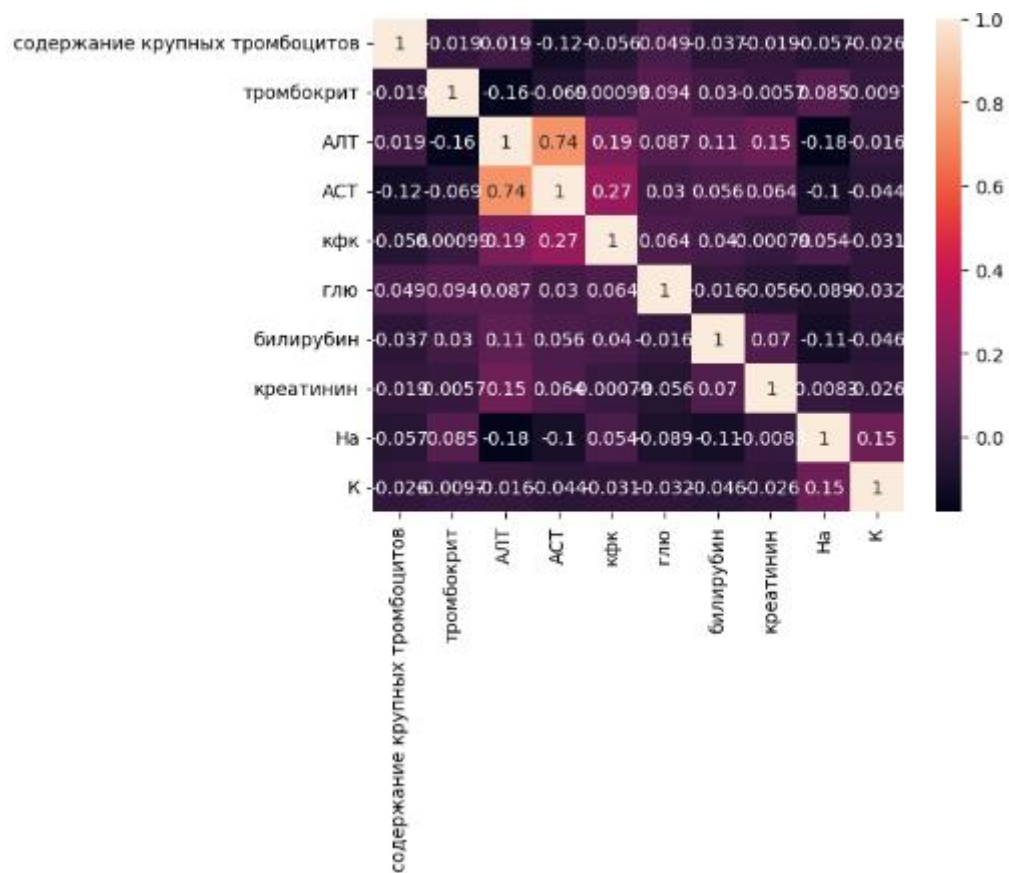


Рисунок 5.1.1 – Таблица корреляций параметров 80-90

Код для отображения корреляций параметров имеет следующий вид:


```
corr_matrix = data.iloc[:, num_col_id[80:90]].corr()
sn.heatmap(corr_matrix, annot=True)
plt.show()
```

В результате анализа корреляций параметров не было произведено дополнительное удаление колонок, поэтому набор данных для обучения все так же состоит из 216 параметров.

5.2 TargetEncoder

Для кодирования категориальных признаков был использован метод TargetEncoder. Данный метод использует целевую переменную для преобразования категорий в числовые значения вместо простого замещения категорий средним или медианным значением признака, как это делается в случае Label Encoding.

Для каждой уникальной категории категориального признака вычисляется вероятность отнесения к положительному классу, и эти вероятности присваиваются вместо категориальных данных в исходном датасете (рис. 5.2.1). Данный подход позволяет не увеличить число колонок, как это происходит при One Hot Encoding.



	Color	Target_1		Color_Target_1
0	Red	1	0	0.400000
1	Red	1	1	0.400000
2	Red	0	2	0.400000
3	Red	0	3	0.400000
4	Red	0	4	0.400000
5	Green	1	5	0.333333
6	Green	0	6	0.333333
7	Green	0	7	0.333333

Рисунок 5.2.1 – Кодирование с использованием TargetEncoder

В коде реализация TargetEncoder представляется в следующем виде:

```
for i in cat_col_id:
    encoder = ce.TargetEncoder(cols=data.columns[i])
    data.iloc[:, i] = encoder.fit_transform(data.iloc[:, i], data.iloc[:, 8])
```

5.3 Полносвязная нейронная сеть

Архитектура нейронной сети была реализована с использованием библиотеки PyTorch:

```

class MyModel(nn.Module):
    def __init__(self):
        super().__init__()
        self.hidden1 = nn.Linear(215, 256)
        self.act1 = nn.ReLU()
        self.hidden2 = nn.Linear(256, 128)
        self.act2 = nn.ReLU()
        self.hidden3 = nn.Linear(128, 64)
        self.act3 = nn.ReLU()
        self.hidden4 = nn.Linear(64, 32)
        self.act4 = nn.ReLU()
        self.output = nn.Linear(32, 1)
        self.act_output = nn.Sigmoid()

    def forward(self, x):
        x = self.act1(self.hidden1(x))
        x = self.act2(self.hidden2(x))
        x = self.act3(self.hidden3(x))
        x = self.act4(self.hidden4(x))
        x = self.act_output(self.output(x))
        return x

model = MyModel()
print(model)

```

В результате запуска кода была создана полносвязная нейронная сеть, имеющую структуру такую, что данные проходят через четыре скрытых слоя с функцией активации ReLU, а затем через выходной слой с функцией активации сигмоида. Параметры каждого слоя можно увидеть на рисунке 5.3.1.

```

MyModel(
  (hidden1): Linear(in_features=215, out_features=256, bias=True)
  (act1): ReLU()
  (hidden2): Linear(in_features=256, out_features=128, bias=True)
  (act2): ReLU()
  (hidden3): Linear(in_features=128, out_features=64, bias=True)
  (act3): ReLU()
  (hidden4): Linear(in_features=64, out_features=32, bias=True)
  (act4): ReLU()
  (output): Linear(in_features=32, out_features=1, bias=True)
  (act_output): Sigmoid()
)

```

Рисунок 5.3.1 – Архитектура и параметры нейронной сети

Далее производилось обучение нейронной сети на 85% данных (302 примера) при 100 эпохах и размере батча 10:

```

n_epochs = 100
batch_size = 10

for epoch in range(n_epochs):
    for i in range(0, len(train_x), batch_size):
        Xbatch = train_x[i:i+batch_size]
        y_pred = model(Xbatch)
        ybatch = train_y[i:i+batch_size]
        loss = loss_fn(y_pred, ybatch)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
    print(f'Finished epoch {epoch}, latest loss {loss}')

```

В результате обучения модели функция потерь уменьшается, что свидетельствует об успешном обучении (рис. 5.3.2).

```

Finished epoch 82, latest loss 0.0013065843377262354
Finished epoch 83, latest loss 0.0011564919259399176
Finished epoch 84, latest loss 0.001083550276234746
Finished epoch 85, latest loss 0.0007141104433685541
Finished epoch 86, latest loss 0.0003291331813670695
Finished epoch 87, latest loss 0.00027609881362877786
Finished epoch 88, latest loss 0.0072949775494635105
Finished epoch 89, latest loss 9.102114563575014e-05
Finished epoch 90, latest loss 4.243968942319043e-05
Finished epoch 91, latest loss 0.0002759795752353966
Finished epoch 92, latest loss 0.0006691827438771725
Finished epoch 93, latest loss 0.0009926072089001536
Finished epoch 94, latest loss 0.0008222360629588366
Finished epoch 95, latest loss 0.0018311480525881052
Finished epoch 96, latest loss 0.0007724780589342117
Finished epoch 97, latest loss 0.0005682287737727165

```

Рисунок 5.3.2 – Функция потерь при различных эпохах

После обучения было произведено тестирование модели на обучающей и тестовой выборке, в результате чего были получены:

- для обучающей выборки Accuracy = 0.96; F1 = 0.97;
- для тестовой выборки Accuracy = 0.91; F1 = 0.92.

По полученным результатам точности и F1-мере можно сделать вывод об успешном функционировании построенной модели машинного обучения. Модель демонстрирует высокую эффективность как на обучающей, так и на тестовой выборках, а высокие значения Accuracy и F1 свидетельствуют о том, что модель хорошо обобщается на новые данные. Кроме того, соотношение

значений Accuracy и F1 говорит о том, что модель хорошо справляется с балансом между точностью и полнотой в классификации.

Код для вывода точности и F1-меры имеет вид:

```
with torch.no_grad():
    y_pred = model(train_x)

accuracy = (y_pred.round() == train_y).float().mean()
f1 = f1_score(train_y, y_pred.round())
print(f"Accuracy {accuracy}, F1 {f1}")
with torch.no_grad():
    y_pred = model(test_x)

accuracy = (y_pred.round() == test_y).float().mean()
f1 = f1_score(test_y, y_pred.round())
print(f"Accuracy {accuracy}, F1 {f1}")
```

5.4 SHAP

Реализация использования SHAP, который позволяет выделить ключевые характеристики, влияющие на выживаемость, и отобразить вклад каждого параметра в прогноз, была осуществлена с помощью кода:

```
explainer = shap.DeepExplainer(model, train_x)
shap_values = explainer.shap_values(test_x)

shap.summary_plot(shap_values, test_x, feature_names=X_train.columns)
```

В результате использования SHAP было выделено 20 наиболее влияющих параметров на прогноз выживаемости пациента (рис. 5.4.1).

Рассматривая полученную, объяснительную модель можно сделать вывод о том, что:

- с низкие тромбоциты свидетельствуют о смерти, а высокие – о выживаемости;
- высокий креатинин₂ ведет к смерти, а низкий к выживаемости;
- высокая частота сердечных сокращений при поступлении ведет к смерти, а их среднее значение или немного пониженное к выживанию;

- высокие лейкоциты ведут к смерти, а средние или слегка пониженные к выживанию;
- пониженные значения АЧТВ ведут к смерти, а повышенные к выживанию;
- низкий ПТИ ведет к смерти, а средний или слегка повышенный к выживанию;
- понижение лимфоцитов ведет к смерти, а их среднее значение или повышенное к выживанию;
- повышенный АЛТ ведет к смерти.

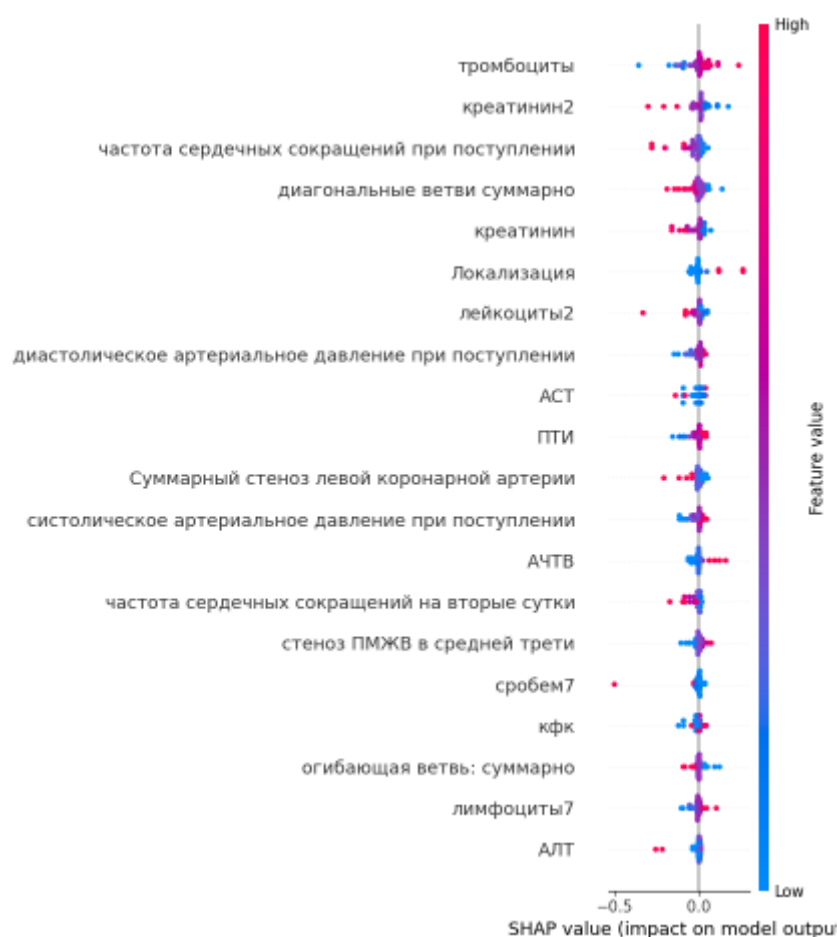


Рисунок 5.4.1 – Параметры, наиболее влияющие на прогноз выживаемости

6 Построение портретов пациентов

Исходя из полученных данных объяснительной модели SHAP и их влияния на выживаемость пациентов, можно составить портреты типичных пациентов:

Пациент, склонный к смерти:

- Низкие тромбоциты.
- Высокий уровень креатинина.
- Высокая частота сердечных сокращений при поступлении.
- Высокие лейкоциты.
- Пониженные значения АЧТВ.
- Низкий ПТИ.
- Пониженные лимфоциты.
- Повышенный АЛТ.

Пациент, склонный к выживанию:

- Высокие тромбоциты.
- Низкий уровень креатинина.
- Средняя или немного пониженная частота сердечных сокращений при поступлении.
- Средние или слегка пониженные лейкоциты.
- Средние или повышенные значения АЧТВ.
- Средний или слегка повышенный ПТИ.
- Средние или повышенные лимфоциты.
- Нормальный уровень АЛТ.

Построенные портреты базируются на предположениях о влиянии различных медицинских показателей на прогноз выживаемости пациентов при поступлении в больницу.

Заключение

В данной работе были успешно выполнены все поставленные задачи. Были получены данные из ГБУ Санкт-Петербургского научно-исследовательского института скорой помощи им. И.И. Джанелидзе, включая информацию о пациентах с инфарктом, типе рака и результатах медицинских анализов. Далее была произведена тщательная предобработка данных, включая обработку пропущенных значений, стандартизацию и нормализацию числовых признаков, а также кодирование категориальных переменных.

Для решения задачи бинарной классификации, направленной на предсказание выживаемости пациента, была использована полносвязная нейронная сеть, способная учитывать сложные взаимосвязи между характеристиками пациентов. Далее с использованием SHAP были выделены ключевые характеристики, влияющие на выживаемость, и получены вклады каждого параметра в прогноз.

Построенная модель машинного обучения была оценена с использованием метрик классификации, таких как точность и F1-мера, что обеспечило адекватность и надежность результатов.

На основе обученной модели были получены типичные портреты пациентов с инфарктом при различных типах рака в разрезе выживаемости.

Полученные результаты являются обобщением главных параметров, которые могут влиять на смертность пациентов с инфарктом при различных типах рака.