

# «Система поддержки принятия медицинских решений»

Хмелёв Алексей

# Задача

Даны данные, хранящие информацию о пациентах:

- Показания с датчиков
- Количество прожитых дней
- Умер человек в отделении интенсивной терапии или нет

Необходимо:

- Создать приложение, которое по данным с датчиков пациента будет предсказывать, сколько дней проживет пациент и умрет ли он в отделении интенсивной терапии или нет

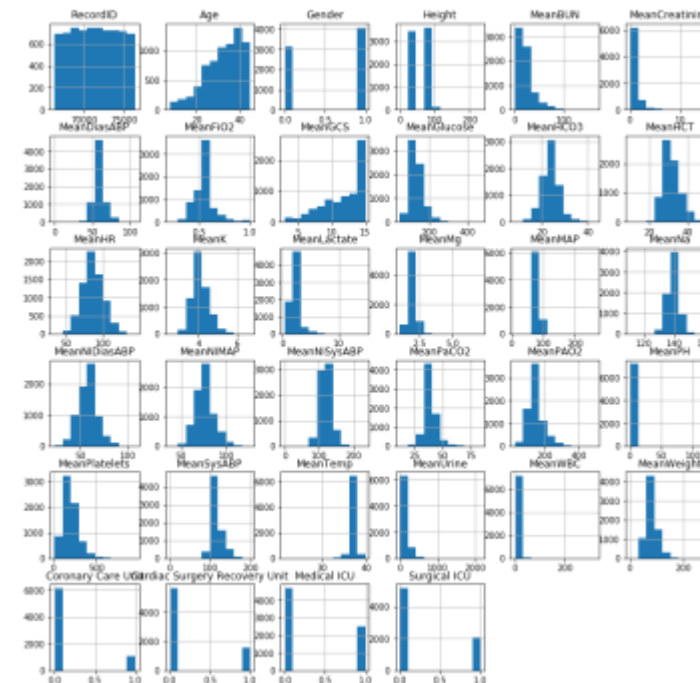
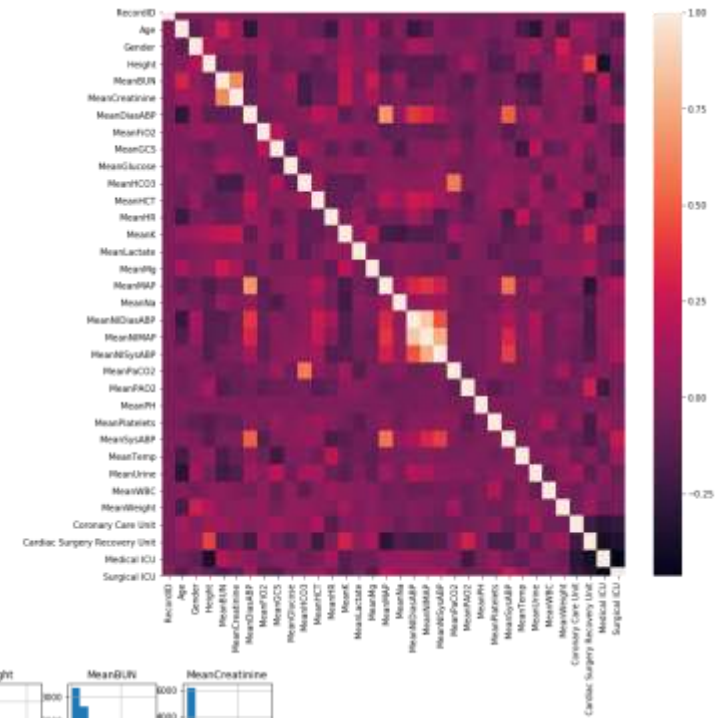
В ходе первой сессии данные были обработаны и от временных рядов для каждого параметра был осуществлен переход к некоторым фичам, которые не зависят от длительности расчета показателей пациента

|      | RecordID | Age  | Gender | Height | ICUType | MeanAlbumin | MeanALP | MeanALT | MeanAST | MeanBilirubin | ... | MeanSysABP | MeanTemp  | MeanTroponinI | Mei |
|------|----------|------|--------|--------|---------|-------------|---------|---------|---------|---------------|-----|------------|-----------|---------------|-----|
| 0    | 66269.5  | 27.0 | 0.0    | -0.50  | 2.0     | NaN         | NaN     | NaN     | NaN     | NaN           | ... | NaN        | 37.357143 | NaN           |     |
| 1    | 66270.0  | 38.0 | 0.5    | 87.65  | 1.0     | NaN         | NaN     | NaN     | NaN     | NaN           | ... | 113.411765 | 36.939130 | NaN           |     |
| 2    | 66270.5  | 22.0 | 0.0    | -0.50  | 1.5     | 2.5         | 116.0   | 83.0    | 199.5   | 2.9           | ... | 125.687500 | 37.800000 | NaN           |     |
| 3    | 66271.5  | 34.0 | 0.5    | 90.15  | 1.5     | 4.4         | 105.0   | 12.0    | 15.0    | 0.2           | ... | NaN        | 36.223077 | NaN           |     |
| 4    | 66272.5  | 44.0 | 0.0    | -0.50  | 1.5     | 3.3         | NaN     | NaN     | NaN     | NaN           | ... | NaN        | 36.880000 | NaN           |     |
| ...  | ...      | ...  | ...    | ...    | ...     | ...         | ...     | ...     | ...     | ...           | ... | ...        | ...       | ...           | ... |
| 7995 | 76424.5  | 39.0 | 0.5    | 90.15  | 1.0     | NaN         | NaN     | NaN     | NaN     | NaN           | ... | 129.271186 | 36.854348 | NaN           |     |
| 7996 | 76425.5  | 45.0 | 0.5    | 88.90  | 1.5     | NaN         | 74.0    | 12.0    | 25.0    | 1.0           | ... | 117.416667 | 37.092857 | NaN           |     |
| 7997 | 76429.0  | 35.0 | 0.0    | 76.20  | 1.0     | 2.8         | 88.0    | 11.0    | 21.0    | NaN           | ... | 120.272727 | 36.600000 | 1.0           |     |
| 7998 | 76431.0  | 24.5 | 0.0    | -0.50  | 1.5     | NaN         | NaN     | NaN     | NaN     | NaN           | ... | NaN        | 36.409091 | NaN           |     |
| 7999 | 76432.0  | 41.0 | 0.0    | -0.50  | 1.0     | NaN         | NaN     | NaN     | NaN     | NaN           | ... | 117.422535 | 36.927273 | NaN           |     |

8000 rows x 45 columns

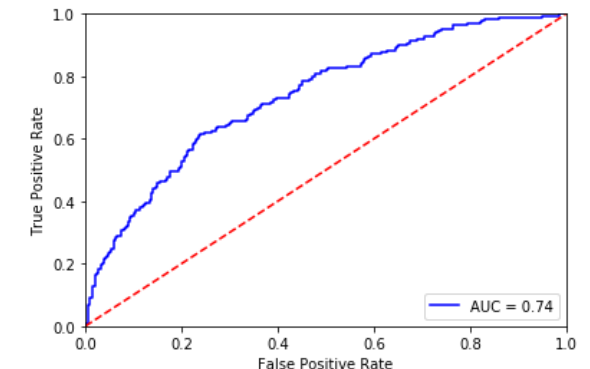
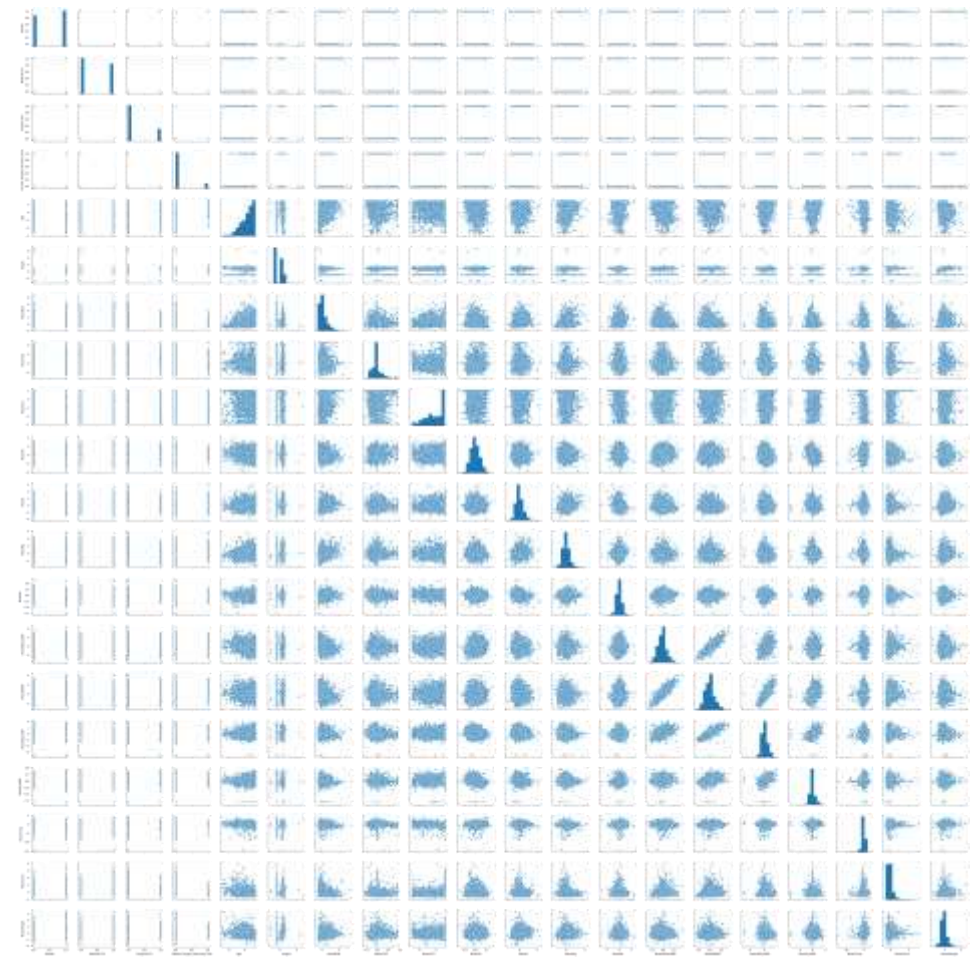
В ходе второй сессии:

- Из данных были убраны дубликаты и пустые столбцы
- Был убран параметр MeanMechVent, т.к. он состоял из одних единиц и пропусков
- Параметры Gender, Height, MeanWeight, ICUType были приведены к правильным значениям (в данных присутствовали отрицательный вес, гендер=0.5 и прочее)
- Были удалены столбцы, в которых больше половины пропусков
- Остальные пропуски были заменены средним значением столбца
- Были удалены выбросы используя IsolationForest
- Были построены гистограммы параметров, боксплоты и коррелогограмма, в результате анализа последней было получено, что заметна высокая корреляция признаков MeanNIMAP с MeanNiDiasABP и MeanNIMAP с MeanNiSysABP



В ходе третьей сессии:

- Данные были нормализованы
- Из датасета были получены два отдельных датасета (для задачи прогноза продолжительности жизни и для задачи предсказания выживаемости)
- В результате отбора признаков (отбор проводился используя обучение моделей с L1-регуляризатором, который зануляет коэффициенты перед малозначимыми признаками) в каждом датасете остались по 20 признаков из 34
- Поиск зависимостей в данных проводился используя коррелограмму Пирсона и Кендалла. Также зависимости были оценены визуально (построены парные графики всех зависимостей)
- В данных для классификации присутствовал дисбаланс классов (6328 наблюдений класса 0 против 872 наблюдений класса 1). Данные для тренировки и обучения делились в отношении 80/20. При обучении моделей использовалась кросс-валидация
- Для поиска лучшей регрессионной модели были рассмотрены линейная регрессия, Lasso-регрессия, Ridge-регрессия, ElasticNet и RandomForest. В результате лучшей регрессионной моделью оказалась Lasso-регрессия
- Для поиска лучшей классификационной модели были рассмотрены логистическая регрессия, случайный лес и SVM. В результате лучшей классификационной моделью оказался SVM (ROC-AUC = 0.74)



В ходе четвертой сессии было написано консольное приложение для:

- Прогноза продолжительности жизни пациентов
- Предсказания выживаемости пациента
- Вывода индикатора степени тяжести больного

Перед запуском необходимо выполнить:

`pip install -r requirements.txt` (установка необходимых библиотек)

Получение справки по работе с приложением:

`python3 app.py --help`

Приложение делает предсказания по данным из файла с информацией о пациентах и позволяет получить результат тремя разными способами:

1. Вывести его на экран
2. Получить в формате json из скрипта
3. Сохранить таблицу с результатом в файл

```
(venv) C:\Users\student\Desktop\C_4_2>python app.py --help
usage: app.py [-h] --path-to-data PATH_TO_DATA [--output-type OUTPUT_TYPE]
              [--path-to-result PATH_TO_RESULT]

API для формирования заключения по прогнозированию продолжительности жизни и
классификации выживаемости пациента

optional arguments:
  -h, --help            show this help message and exit
  --path-to-data PATH_TO_DATA
                        Путь до файла, содержащего данные о пациенте
  --output-type OUTPUT_TYPE
                        output - выводит результат на экран; json - возвращает
                        json; file - сохраняет результат в файл (необходимо
                        указать путь в --path-to-result)
  --path-to-result PATH_TO_RESULT
                        Путь, куда сохраняется результат работы моделей
```

Вывод результата на экран:

```
python3 app.py --path-to-data [ПУТЬ ДО ДАННЫХ] --output-type output
```

Получение результата в формате json:

```
python3 app.py --path-to-data [ПУТЬ ДО ДАННЫХ] --output-type json
```

Сохранение результата в виде таблицы в файл:

```
python app.py [ПУТЬ ДО ДАННЫХ] data.csv --output-type file --path-to-result [ПУТЬ ДЛЯ РЕЗУЛЬТИРУЮЩЕГО ФАЙЛА]
```

```
(venv) C:\Users\student\Desktop\C_4_2>python app.py --path-to-data data.csv --output-type output
```

|      | RecordID | Продолжительность жизни (в днях) | Степень тяжести | Наступит смерть |
|------|----------|----------------------------------|-----------------|-----------------|
| 0    | 66269.5  | 584                              | зеленая         | нет             |
| 1    | 66270.0  | 692                              | зеленая         | нет             |
| 2    | 66270.5  | 253                              | желтая          | нет             |
| 3    | 66271.5  | 588                              | зеленая         | нет             |
| 4    | 66272.5  | 553                              | зеленая         | нет             |
| ...  | ...      | ...                              | ...             | ...             |
| 7195 | 76424.5  | 675                              | зеленая         | нет             |
| 7196 | 76425.5  | 124                              | желтая          | нет             |
| 7197 | 76429.0  | 857                              | зеленая         | нет             |
| 7198 | 76431.0  | 376                              | зеленая         | нет             |
| 7199 | 76432.0  | 928                              | желтая          | нет             |

```
[7200 rows x 4 columns]
```