МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ЭКОНОМИКИ И СЕРВИСА

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

РЕКОМЕНДОВАНО
к защите

зав. кафедрой ИТС
канд. экон. наук, доцент

_____ Е.В. Кийкова

БАКАЛАВРСКАЯ РАБОТА

Разработка информационного сервиса прогнозирования фатального события от сердечно – сосудистых заболеваний

Б-ИС-16-126787. 3869-с. 05.00. БР

Студент _____ В.В. Костерин

Научный руководитель
доктор технических наук,
профессор кафедры ИТС _____ К.И. Шахгельдян

Нормоконтроль
кандидат технических наук,
доцент кафедры ИТС _____ М.А. Сачко

Владивосток 2020

Содержание

Введение

Сердечно – сосудистые заболевания (ССЗ) остаются одной из самых важных проблем большинства стран современного мира, что обусловлено преобладающим уровнем смертности населения от этих патологических форм. По рекомендациям Всемирной организации здравоохранения (ВОЗ), главным подходом к понижению и стабилизации сердечно-сосудистой смертности является активная деятельность по ограничению воздействия на организм факторов риска (ФР). Вместе с тем, низкая эффективность реализации мероприятий популяционной и индивидуальной коррекции ФР способствует прогрессированию ССЗ. [1]

Профилактика ССЗ оказывает положительное влияние на снижение уровня основных ФР, на уровне популяции может предотвратить до 80% преждевременных смертей от ССЗ. Выявляя людей, которые подвержены риску развития ССЗ, и предпринимая меры по уменьшению данного риска, большую часть летальных и не летальных сердечно – сосудистых событий можно предотвратить.

Один из методов профилактики является оценка ФР ССЗ.В России применяется шкала оценки риска летальности от ССЗ SCORE, так как в обучающей выборке данной модели присутствовали результаты наблюдений за пациентами из России. Существенным недостатком ранее разработанных шкал является отсутствие учета ряда индивидуальных особенностей обследуемых лиц.

Лучшие результаты предсказаний ССЗ реализуемы при разработке шкал и моделей оценки рисков летальности от ССЗ, получить которые можно только в результате долгих активных наблюдений за контингентом, который обследуется.

Данный проект проводится на данных исследования эпидемиологии сердечно – сосудистых заболеваний Российской Федерации. Основной целью исследования является исследование факторов риска ССЗ среди популяции Приморского края, разработка моделей по прогнозированию вероятности смерти в течение ближайших 7 лет от ССЗ.

Целью данного проекта является разработка технологии и информационного сервиса для оценки индивидуальных рисков развития сердечно – сосудистых заболеваний на основе методов искусственного интеллекта.

1 Оценка летальности в мире и России от сердечно-сосудистых заболеваний

1.1 Летальность от ССЗ в России и мире

Основной причиной смертности во всем мире являются ССЗ, которые становятся причиной смерти, почти, 18 миллионов человек (по оценке ВОЗ на 2016 года), что составляет 31% из всех летальных исходов населения планеты, 85% этих смертей произошло в результате сердечного приступа и инсульта. [1]]

На рисунке 1.1 демонстрируется глобальный график причин смерти.

Рисунок 1.1 – Распределение основных причин смертности [2]

В России ССЗ – основная причина смертности населения на протяжении длительного времени. По данным ВОЗ. большая половины из всех смертей произошла из – за ССЗ, что равняется, почти, 1 миллиону смертей, где мужчины составляют 44,9%, а женщины – 55,4%.

Коэффициенты Европейского стандарта смертности от ССЗ в России являются высокими:

- 703,6 у мужчин;
- 382,6 – у женщин.

Только в нескольких странах региона Европы показатели смертности превышают наши, это такие страны как:

- Украина;
- Беларусь;
- Болгария;
- Литва.

Распределение смертности от ССЗ изображено на рисунке 2.

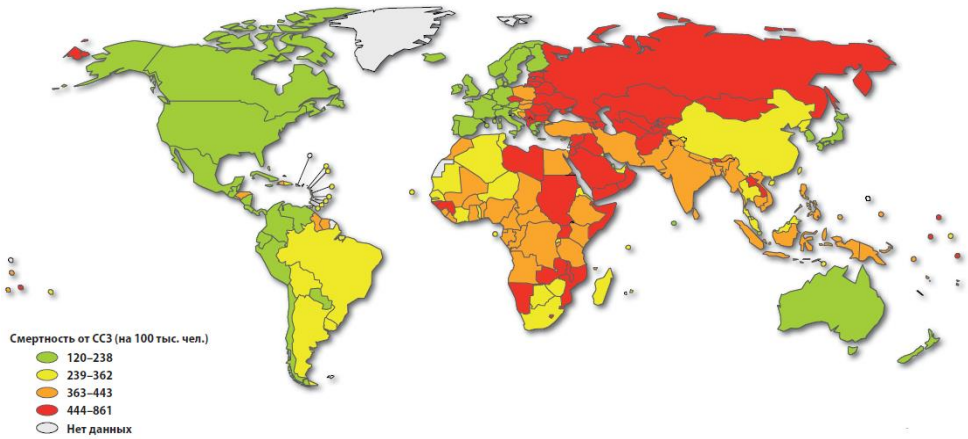

Рисунок 1.2 – Карта мира, демонстрирующая распределение смертности от ССЗ [2]

Множественные смерти от ССЗ считаются предотвратимыми за счет устранения или исправления определенных ФР. Из всех популяций, которые отмечали уменьшение смертности от ССЗ, подавление влияния ФР способствовало значительному улучшению уровня смертности.

Профилактика более чем эффективна: соблюдение правил и принципов здорового образа жизни и снижение уровня основных ФР на уровне популяции может предотвратить до 80% преждевременных смертей от ССЗ. Выявляя людей, которые подвержены риску развития ССЗ, и предпринимая меры по уменьшению данного риска, большую часть летальных и не летальных сердечно – сосудистых событий можно предотвратить. [3]

Существует много видов профилактики ССЗ. Один из видов кардиоваскулярной профилактики является оценка ФР ССЗ, суммарного ССЗ и понижение его модернизацией действующих факторов. В клинической медицине существуют различные модели оценки ФР летальности от ССЗ, которые успешно применяются на практике. К таким моделям относятся SCORE, ASSIGN SCORE, FRAMINGHAM, QRISK и так далее.

В России применяется модель оценки рисков летальности от ССЗ SCORE, так как она внесена, как инструмент оценки риска в национальные клинические рекомендации и порядки оказания медицинской помощи. Это связано с тем, что при ее разработке были учтены данные о многотысячной когорте длительно наблюдавшихся российских пациентов, а также последовательными рекомендациями использования этой шкалы Европейским и Российским обществами кардиологов.

У шкалы оценки риска летальности от ССЗ есть свои изъяны такие как:

- адаптация к популяциям Европейских стран, а не к этническим группам внутри этих популяций;

- шкала ограничена только основными ФР и не учитывает другие, влияющие на ССР;
- ограниченный диапазон возраста.

Шкала оценки рисков летальности от ССЗ SCORE показана на рисунке 3.

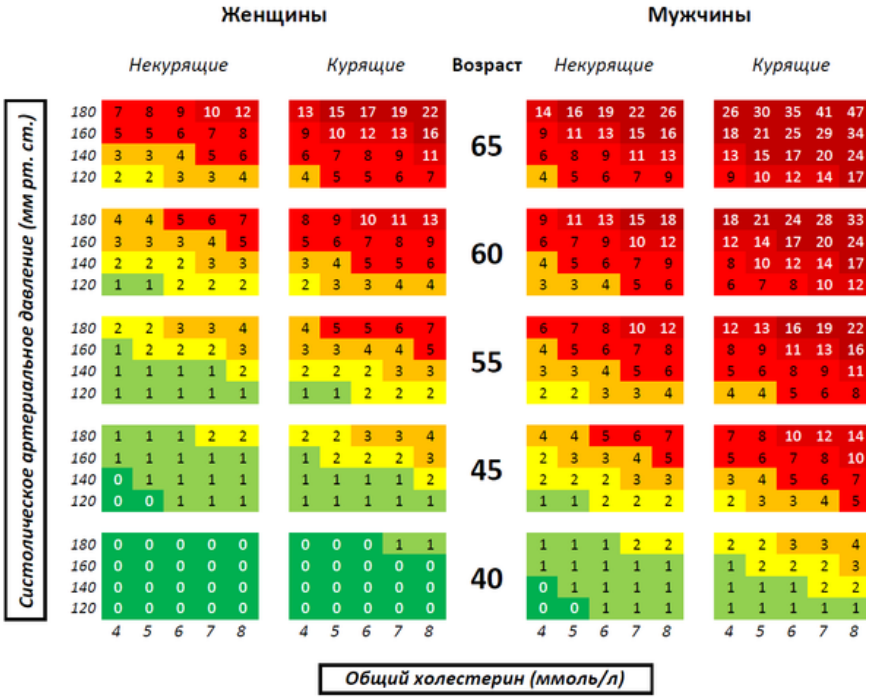

Рисунок 1.3 – Шкала SCORE для оценки риска летальности от ССЗ в % для мужчин и женщин

Основной особенностью этих моделей является их обучение на основе определенной популяционной выборки. Соответственно данные шкалы содержать ФР адаптированные для соответствующих регионов. SCORE – в западной Европе, FRAMINGHAM – в Америке, QRISK – Великобритания, ASSIGN SCORE – в Шотландии.

Согласно объединенным рекомендациям ESC (European Society of Cardiology) по оценке и профилактике кардиоваскулярного риска в реальной клинической практике существенным недостатком ранее разработанных шкал является отсутствие учета ряда индивидуальных особенностей обследуемых лиц.

Традиционных подход оценки риска ССЗ относительно популяции имеет свои недостатки. Лучшие результаты предсказаний ССЗ реализуемы при разработке шкал и моделей оценки рисков летальности от ССЗ, получить которые можно только в результате долгих активных наблюдений за контингентом, который обследуется, например, в рамках регионального этапа исследования ЭССЕ-РФ. [1]

1.2 Исследование эпидемиологии сердечно – сосудистых заболеваний в Российской Федерации

Исследование эпидемиологии сердечно – сосудистых заболеваний в Российской Федерации (ЭССЕ – РФ) было проведено по всей стране в 12 регионах: Приморский край, Вологодская область, Воронежская область, Ивановская область, Красноярский край, Оренбургская область, Самарская область, Волгоградская область, Северо – Западный федеральный округ, республика Северная Осетия (Алания), Томская область, Тюменская область в 2013 году. Данные были получены на основе анкетирования и объективных клинико – медицинских исследований.

Исследование ЭССЕ – РФ в Приморском крае обеспечило сбор клинических, антропометрических, лабораторных и инструментальных данных, всего 2132 обследованных в 2013 году. Кроме того, в 2019 году была собрана информация о смертности среди исследуемой когорты. Это исследование продолжается и объем данных для поиска факторов риска (ФР) и их верификации в процессе реализации проекта будет увеличиваться.

Оцифровка и деперсонификация фактических данных осуществляется в соответствии с требованиями современного законодательства и правил обработки первичных биомедицинских данных. [4]

Dataset ЭССЕ – РФ представляет собой .xlsx файл, который содержит 206 атрибутов, часть из которых результат объективных исследований, таких как: осмотр врачом, данные лаборатории и инструментальные исследования, а остальная часть – анкетирование обследуемого.

Анкетирование содержит различные модули, такие как:

- информация о респонденте;
- пищевые привычки и физическая активность;
- курение и употребление алкоголя;
- здоровье, сон и качество жизни;
- заболевания;
- экономические условия и работа;
- стресс, тревога и депрессия;
- данные об обращаемости за медицинской помощью и нетрудоспособности;
- вопросник, по оценке качества жизни.

Объективные данные содержат такие информацию, как:

- измерение артериального давления;
- антропометрические показатели;

- электроэнцефалография;
- биохимический и клинический анализ крови;
- электрокардиограмма покоя.

Набор данных для анализа содержит все основные и целый ряд дополнительных параметров, которые позволяют выполнить оценку для каждого индивидуума по традиционным методикам и подойти к анализу возможностей и разработке дополнительных шкал с использованием алгоритмов искусственного интеллекта. [1]

1.3 Научная цель и задачи проекта

Новые технологии хранения, обработки и сбора данных дали возможность собрать большие объемы биомедицинских данных, включая итоги популяционных исследований для профилактической медицины, бумажные и электронные истории болезни содержат информацию разного рода об обследуемых контингентах, а также о пациентах на различных этапах заболевания.

Эти данные содержат скрытые знания о взаимосвязях между функциональным и генетическим статусом обследованных, с одной стороны и вероятностью возникновения определенных заболеваний или их осложнений - с другой. Показатели, для которых выявлены такие взаимосвязи, относят к факторам риска (ФР) соответствующих заболеваний. Интерес представляют также разнообразные сочетания ФР, которые могут выступать в качестве предикторов развития определенных форм сердечно-сосудистых заболеваний.

Целью данного проекта является разработка технологии и информационного сервиса для оценки индивидуальных рисков развития сердечно – сосудистых заболеваний на основе методов искусственного интеллекта.

Основными задачами для реализации поставленной цели являются:

- проведение исследования моделей по оценке рисков летальности от ССЗ;
- разработка информационного веб-сервиса для оценки индивидуального десятилетнего риска летальности от ССЗ.

1.4 Описание научного коллектива гранта

Данный проект разрабатывается в рамках гранта «РФФИ_МК_901.1 Разработка методов верификации и прогнозирования рисков ССЗ» в ФГБОУ ВО "Владивостокский государственный университет экономики и сервиса", кафедра информационных технологий и систем, Лаборатория цифрового моделирования и анализа данных физики и биомедицины.

Название проекта – Разработка интеллектуальной технологии оценки факторов кардиоваскулярного риска и построения моделей прогнозирования сердечно-сосудистых событий.

Организация, через которую заключается договор – ВГУЭС. Руководитель и грантополучатель – Гельцер Борис Израилевич, профессор, член корреспондент Российской Академии Наук, доктор медицинских наук.

Исполнители:

- Шахгельдян К.И. (ВГУЭС)
- Невзорова В.А. (ТГМУ)
- Бродская Т.А. (ТГМУ)
- Кистенев Ю.В. (ТГУ)
- Москаленко Ф.М. (ИАПу)
- Петряева (ИАПУ)
- Шалфеева (ИАПУ)
- Тимченко (ИАПУ)
- Емцева Е.Д. (ВГУЭС)

Целью проекта является разработка самообучающейся технологии на данных исследования ЭССЕ – РФ, обеспечивающей разработку методов оценки индивидуальных рисков развития ССЗ, прогнозных моделей наступления сердечно-сосудистых событий и создание персонализированных рекомендаций по их профилактике на основе алгоритмов искусственного интеллекта.

Данный проект разрабатывается на базе Лаборатории цифрового моделирования и анализа данных физики и биомедицины.

Комплекс мер по поддержанию здорового образа жизни сотрудников ФГБОУ ВО "Владивостокский государственный университет экономики и сервиса" описан в Приложении А.[5-6]

Планировка рабочих мест в Лаборатории цифрового моделирования и анализа данных физики и биомедицины соответствует нормам (Приложение Б).[7]

2.Описание инструментария проекта

2.1 Выбор языка программирования

Выбор языка программирования (ЯП) определялся по популярности его для DataSience и MachineLearning на ресурсе Kaggle.

Kaggle – система организации конкурсов по исследованию данных, а также социальная сеть специалистов по обработке данных и машинному обучению. Принадлежит корпорации Google.

График популярности ЯП в DS за 2018 год показан на рисунке 2.1.

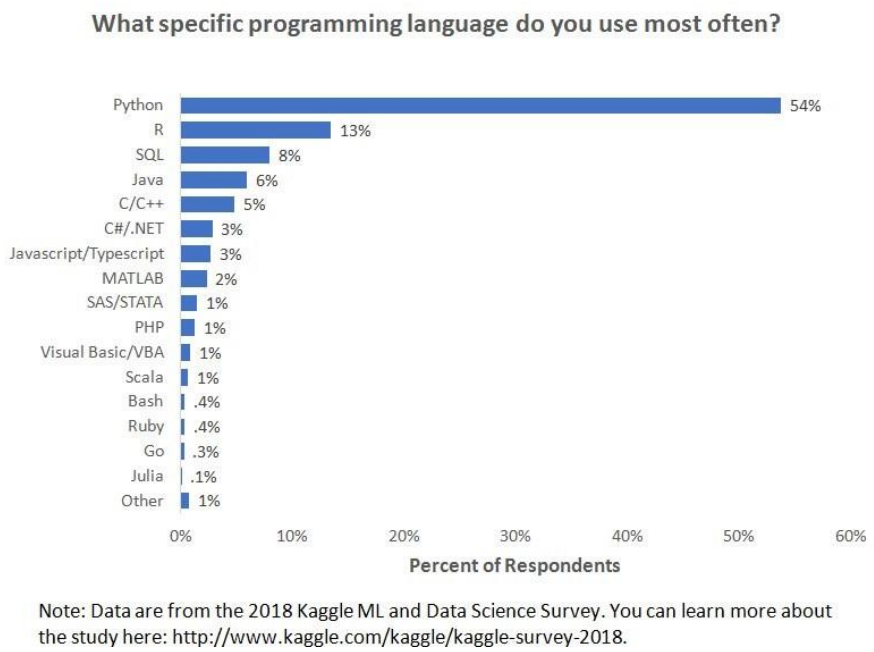

Рисунок 2.1 – График «Какой ЯП вы используете чаще всего?» в DS [8]

Самые популярные, простые и мощные ЯП для анализа данных это Python и R. Проведем аналитику между R и Python.

R – язык программирования для статистической обработки данных. Лицензия – GNU GPL 2.

Преимущества:

- большой выбор предметно – ориентированных пакетов. Сюда входят нейронные сети, нелинейная регрессия, филогенетика, построение сложных диаграмм, графиков и так далее;
- R прекрасно обрабатывает данные матричной алгебры.

Недостатки:

- низкая производительность;

- специфичность.

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Лицензия – Python Software Foundation License (совместим с GPL).

Преимущества:

- ЯП общего назначения, используется в том числе и для разработки web – сервисов;
- большой выбор предметно – ориентированных пакетов таких, как tensorflow, pandas и scikit – learn делают Python уверенным фаворитом для современных приложений в области анализа данных и машинного обучения.

Недостатки:

- возможность ошибки несоответствия типов;
- низкая производительность.

Так как после построения моделей, необходимо реализовать онлайн – калькулятор оценки риска смерти от ССЗ, а Python является ЯП общего назначения на котором это можно сделать благодаря своим framework – ам Django, Pyramid, Flask, кроме того код аналитики интегрируются в код калькулятора намного быстрее и проще, чем аналогичный код на R.

2.2 Используемые библиотеки

Во время работы над проектом были использованы библиотеки:

- pandas;
- matplotlib;
- numpy;
- seaborn;
- scipy;
- statistics;
- math;
- sklearn;

Рассмотрим каждую библиотеку подробнее.

NumPy – основной пакет для выполнения научных расчетов на Python. Возможности: поддержка многомерных массивов; поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами.

NumPy куда эффективнее встроенных в Python структур данных. Кроме того, библиотеки, написанные на низкоуровневом языке типа C или Fortran, могут работать с данными, хранящимися в массиве NumPy, вообще без копирования. [9]

Pandas – программная библиотека на языке Python для обработки и анализа данных. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами. [10]

Пример считывания данных из Excel и запись их в DataFrame при мощи библиотеки pandas демонстрируется на рисунке 2.2.

```
ESSE=pd.read_excel('C:/Users/kraia/Desktop/Диплом/Таблицы/esse.xlsx',sheetname_name='Sheet1')
df1=pd.DataFrame([ESSE['1_9'],ESSE['1_10']]).T
```

	1_9	1_10
0	2.0	2.0
1	2.0	1.0
2	2.0	2.0
3	1.0	2.0
4	2.0	1.0
5	2.0	1.0
6	2.0	2.0
7	2.0	3.0
8	1.0	2.0
9	2.0	1.0
10	2.0	2.0
11	1.0	3.0
12	1.0	2.0
13	2.0	2.0
14	2.0	2.0
15	1.0	1.0

Рисунок 2.2 – Создание DataFrame из 2 атрибутов dataset – а данного проекта

Pandas предоставляет высокую производительность средств работы с массивами, которая присуща numpy, с возможностями манипулирования данными, свойственными электронным таблицам и реляционным базам данных. [9]

Matplotlib – библиотека ЯП Python для визуализации данных двумерной графикой. Получаемые изображения могут быть использованы в качестве иллюстраций в публикациях.

Демонстрация построения графиков в Matplotlib – рисунок 2.3.

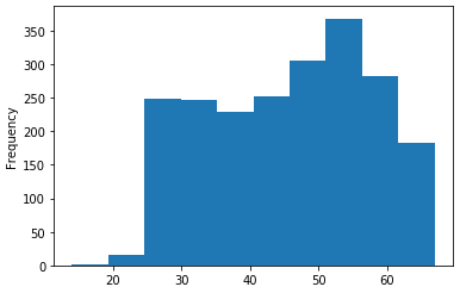

Рисунок 2.3 – Гистограмма возраста обследуемых, созданная при помощи библиотеки Matplotlib

Графики в библиотеке Matplotlib интерактивны – можно увеличить масштаб какого – то участка графика и выполнять панорамирование с помощью панели инструментов в окне графика.[9]

SciPy – библиотека для языка программирования Python с открытым исходным кодом, предназначенная для выполнения научных и инженерных расчётов.

В работе используется непосредственно SciPy.stats.

SciPy.stats – стандартные непрерывные и дискретные распределения вероятностей, различные статистические критерии и дополнительные описательные статистики. [11]

Seaborn – это библиотека визуализации данных Python, основанная на matplotlib. Он предоставляет высокоуровневый интерфейс для рисования привлекательной и информативной статистической графики. [12]

Демонстрация гистограммы с рисунка 10 используя библиотеку seaborn на рисунке 2.4.

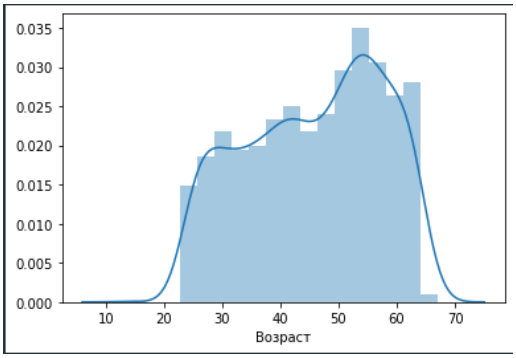

Рисунок 2.4 – Гистограмма возраста обследуемых, созданная при помощи библиотеки Seaborn

Используется, когда требуется отобразить графики более детализировано, чем отображает matplotlib.

Statistics – этот модуль предоставляет функции для расчета математической статистики числовых данных. [13]

Math – этот модуль обеспечивает доступ к математическим функциям, определенным стандартом C. [14]

Sklearn – библиотека для ЯП Python в свободном доступе для машинного обучения. Библиотека предоставляет широкий выбор алгоритмов обучения с учителем и без учителя. Обучение с учителем предполагает наличие размеченного dataset – а, в котором известно значение целевого признака. В то время как обучение без учителя не предполагает наличия разметки в dataset. [15]

2.3 Используемые Python frameworks

2.3.1 TensorFlow.Keras

TensorFlow - это комплексная платформа с открытым исходным кодом для машинного обучения. Он имеет всеобъемлющую, гибкую экосистему инструментов, библиотек и

ресурсов сообщества, что позволяет исследователям использовать новейшие технологии ML, а разработчикам легко создавать и развертывать приложения на базе ML. [16]

Keras является библиотекой Python, реализуемой поверх framework TensorFlow. Был разработан, чтобы сделать внедрение моделей глубокого обучения максимально быстрым и легким для исследований и разработок. Работает на Python 2.7 или 3.5 и может беспрепятственно выполняться на графических процессорах и процессорах с учетом базовых структур.

Основные принципы Keras:

- модульность – под моделью понимается последовательность или график автономных полностью настраиваемых модулей, которые могут быть подключены вместе с минимальными ограничениями. В частности, нейронные слои, функции затрат, оптимизаторы, схемы инициализации, функции активации, схемы регуляризации - это автономные модули, которые вы можете комбинировать для создания новых моделей.
- легкая масштабируемость – новые модули просто добавлять (как новые классы и функции), а существующие модули предоставляют множество примеров. Чтобы иметь возможность легко создавать новые модули, вы можете полностью выразить свою выразительность, что делает Keras подходящим для передовых исследований.
- работа с Python – нет отдельных файлов конфигурации моделей в декларативном формате. Модели описаны в коде Python, который компактен, легче отлаживается и обеспечивает простоту расширяемости

Основная структура данных Keras – это модель, способ организации слоев. В Keras доступны два основных типа моделей: последовательная модель Sequential и класс Model, используемый с функциональным API. [17]

Начало работы в Keras начинается с установки библиотеки командой pip install keras. Также необходимо установить Python версии 3.x и framework TensorFlow.

2.3.2 Django

Django – свободный framework для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation. Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. [18]

Django был выбран по таким причинам, как:

- калькуляторы для расчета рисков летальности от ССЗ SCORE и ASSIGN SCORE ранее были реализованы на языке программирования Python;

- нейросеть реализовывалась на TensorFlow.Keras, которая является framework Python, а Django позволит упростить интеграцию данной неросети;
- Django имеет простой веб-сервер разработки, который можно использовать для тестирования локальных веб-приложений Django на веб-браузере компьютера, а не на внешнем веб сервисе;
- Django может выдерживать высокую нагрузку, плюс имеет встроенные возможности кэширования и распределения нагрузки;
- в framework встроен автоматически генерируемый админский интерфейс;
- использование Python в качестве языка программирования, возможность использования обширных библиотек классов и его хорошая документацию.

Для начала работы необходимо установить Python версии 3.x. Далее при помощи установщика pip установить framework на рабочий компьютер командой pip install Django.

Проекты начинаются с создания проекта. Проект создается командой django – admin startproject «название проекта». После этого в выбранной папке для проекта создается структура проекта, состоящая из файлов:

- manage.py: утилита командной строки, которая позволяет взаимодействовать с проектом Django;
- __init__.py: пустой файл, который сообщает Python, что этот каталог следует рассматривать как пакет Python;
- settings.py: настройки/конфигурация для этого проекта Django;
- urls.py: объявления URL для этого проекта Django; «оглавление» сайта на платформе Django;
- asgi.py: точка входа для ASGI-совместимых веб-серверов для обслуживания проекта;
- wsgi.py: точка входа для WSGI-совместимых веб-серверов для обслуживания проекта. [19]

Каждое приложение, которое пишется в Django, состоит из пакета Python, который следует определенному соглашению. Django поставляется с утилитой, которая автоматически генерирует базовую структуру каталогов приложения. Приложение создается командой python manage.py startapp «название приложения».

3. Исследование моделей по оценке рисков летальности от ССЗ

3.1 Подготовка данных к машинному обучению

Предварительная обработка и очистка данных – это важные задачи, которые необходимо выполнить, прежде чем набор данных можно будет использовать для обучения модели. Необработанные данные зачастую искажены и ненадежны, и в них могут быть пропущены значения. Использование таких данных при моделировании может приводить к неверным результатам. Эти задачи являются частью процесса обработки и анализа данных группы и обычно подразумевают первоначальное изучение набора данных, используемого для определения и планирования необходимой предварительной обработки.

Реальные данные собираются для последующей обработки из разных источников и процессов. Они могут содержать ошибки и повреждения, негативно влияющие на качество набора данных. Вот какими могут быть типичные проблемы с качеством данных:

- неполнота: данные не содержат атрибутов, или в них пропущены значения;
- шум: данные содержат ошибочные записи или выбросы;
- несогласованность: данные содержат конфликтующие между собой записи или расхождения. [20]

Качественные данные – это необходимое условие для создания качественных моделей прогнозирования. Чтобы избежать появления ситуации «мусор на входе, мусор на выходе» и повысить качество данных и, как следствие, эффективность модели, необходимо провести мониторинг работоспособности данных, как можно раньше обнаружить проблемы и решить, какие действия по предварительной обработке и очистке данных необходимы.

Для проверки качества данных необходимо оценить:

- число записей;
- количество атрибутов;
- типы данных атрибута;
- число отсутствующих значений;
- правильность формирования данных;
- проверить допустимость диапазона значений.

Главные задачи предварительной обработки данных

- очистка данных: заполнение отсутствующих значений, обнаружение и удаление шума данных и выбросов;
- преобразование данных: нормализация данных для уменьшения размеров и шума;

- уплотнение данных – создание выборки данных или атрибутов для упрощения обработки данных.

При обнаружении проблем с данными необходимо выполнить действия по обработке.

3.2 Понятие машинного обучения

Первостепенная цель реальной жизни и науки – получение верных предсказаний о поведении сложных систем в будущем на основании их поведения в прошлом.

Множество задач, которые возникают на практике, нельзя решить заранее известными алгоритмами и методами. Это происходит из – за того, что заранее не известны механизмы происхождения исходных данных или же информации, которая является известной, нам мало для создания модели, генерирующей данные.

Необходимо изучать последовательность доступных исходных данных пробовать создавать предсказания, усовершенствуя схему в процессе предсказания. Метод, при котором данные, которые были в прошлом, или параметры применяются для первоначального формирования и совершенствования схемы предсказания, называется методом машинного обучения. [21]

Машинное обучение - чрезвычайно широкая и динамически развивающаяся область исследований, использующая огромное число теоретических и практических методов.

С данным подходом тесно связана задача универсального предсказания. В том случае, когда нет достаточной информации для того чтобы построить модель источника, генерирующего наблюдаемые данные, приходится учитывать, как можно более широкие классы таких моделей и строить методы, которые предсказывают "не хуже", чем любая модель из данного класса.

3.3 Типы машинного обучения

При построении моделей используется две парадигмы обучения: с учителем и без учителя.

Обучение с учителем предполагает, что модели предъявляются примеры, состоящие из пар «известный вход — известный выход». Обучение производится на основе заранее определенных (целевых) значений, с которыми сравниваются реальные значения, сформированные моделью на выходе. При этом вычисляется значение выходной ошибки модели, на основе которого по определенному алгоритму обучения рассчитывается коррекция параметров модели (например, весов нейронной сети). Типичным приложением, в котором необходимо обучение с учителем, является классификация, поскольку классы предполагаются заданными заранее. В этом случае целевой переменной будет метка класса. [20]

Типы входных данных:

- признаковое описание или матрица объекты-признаки — наиболее распространённый случай. Каждый объект описывается набором своих характеристик, называемых признаками. Признаки могут быть числовыми или нечисловыми;
- матрица расстояний между объектами. Каждый объект описывается расстояниями до всех остальных объектов обучающей выборки. С этим типом входных данных работают немногие методы, в частности, метод ближайших соседей, метод парзеновского окна, метод потенциальных функций;
- временной ряд или сигнал представляет собой последовательность измерений во времени. Каждое измерение может представляться числом, вектором, а в общем случае — признаковым описанием исследуемого объекта в данный момент времени;
- изображение или видеоряд;
- графы, текст, результатов запросов к базе данных, и т. д. Как правило, они приводятся к первому или второму случаю путём предварительной обработки данных и извлечения признаков.

Типы откликов:

- задачи классификации – множество возможных ответов конечно. Их называют идентификаторами (именами, метками) классов;
- задачи регрессии – ответы являются действительными числами или векторами. [23]

В процессе обучения без учителя исследуется набор данных и вычисляются скрытые взаимосвязи корреляции между разными переменными. Данный способ может быть применен для кластеризации данных на основе их статистических свойств.

Хорошее применение обучения без учителя — алгоритм кластеризации, используемый для вероятностного соединения записей. Определяются связи между элементами данных, и на основании этих отношений выявляются связи между людьми и организациями в физическом или виртуальном мире. [24]

В данном исследовании данные имеют известный вход и известный выход, поэтому далее будут рассмотрены только методы обучения с учителем.

3.4 Рассмотренные модели обучения с учителем

Так как основная цель исследования создание более эффективной модели оценки риска смерти от ССЗ, а наиболее эффективное решение задачи не известно, необходимо рассмотреть, как можно больше моделей обучения с учителем, для выявления более чувствительной из них.

3.4.1 Логистическая регрессия

Логистическая регрессия является подходящим регрессионным анализом, который необходимо проводить, когда зависимая переменная является дихотомической. Как и все регрессионные анализы, логистическая регрессия является прогностическим анализом. Логистическая регрессия используется для описания данных и для объяснения взаимосвязи между одной зависимой двоичной переменной и одной или несколькими номинальными, порядковыми, интервальными или независимыми переменными уровня отношения. [25]

Логистическая регрессия используется, когда зависимая переменная является категориальной.

Учебный набор – это входные данные, в которых для каждого предопределенного набора функций x, имеется правильная классификация y, что показано в формуле (3.1).

$$\{\{(x^1, y^1)\}, \{(x^2, y^2)\}, \{(x^3, y^3)\}, \dots, \{(x^m, y^m)\}\}, \tag{3.1}$$

где m – количество примеров учебных наборов.

Вероятность вычисляется по формуле (3.2):

$$P = \frac{1}{1 + e^{-z}}, \tag{3.2}$$

где $z = x_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$ – стандартное уравнение регрессии, x и b – векторы – столбцы значений независимых переменных $1, b_1, b_2, \dots, b_n$, и параметров (коэффициентов регрессии) – вещественных чисел $x_0, x_1, x_2, \dots, x_n$ соответственно. Графически сигмоида представлена на рисунке 3.1.

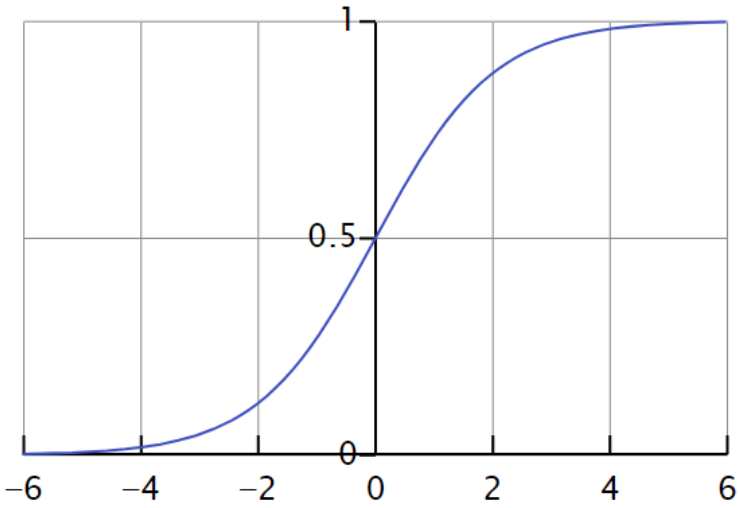

Рисунок 3.1 – График сигмоиды под формулой (2)

Благодаря тому, как обучается модель, предсказания логистической регрессии можно использовать для отображения вероятности принадлежности образца к классу 0 или 1. Это полезно в тех случаях, когда нужно иметь больше обоснований для прогнозирования.

3.4.2 Random forest

Random forest – алгоритм машинного обучения, предложенный Лео Брейманом и Адель Катлер, заключающийся в использовании комитета решающих деревьев. Алгоритм применяется для задач классификации, регрессии и кластеризации. [26]

Пусть обучающая выборка состоит из N образцов, размерность пространства признаков равна M, и задан параметр, как неполное количество признаков для обучения.

Алгоритм обучения:

- генерация случайной подвыборки с повторениями размером N из обучающей выборки;
- отстройка решающего дерева, классифицирующего образцы данной подвыборки, причём в ходе создания очередного узла дерева выбирается набор признаков, на основе которых производится разбиение, выбор наилучшего из этих m признаков осуществляется критерием Джини. Для вычисления критерия Джини для набора элементов с J классами, предположим, что $i \in \{1,2,\dots,J\}$ и пусть p_i будет долей элементов, помеченных классом i в наборе, тогда критерий Джини вычисляется по формуле (3.3):

$$I_G(p) = 1 - \sum_{i=1}^{J} {p_i}^2 ; \tag{3.3}$$

- дерево строится до полного исчерпания подвыборки и не подвергается процедуре отсечение ветвей.

Классификация объектов проводится путём голосования: каждое дерево комитета относит классифицируемый объект к одному из классов, и побеждает класс, за который проголосовало наибольшее число деревьев. [2]

Преимущества;

- эффективная обработка данных с множественными классами и признаками;
- нечувствительность к монотонным преобразованиям значений признаков;
- одинаково хорошая обработка категориальных значений и непрерывных;
- Есть методы оценки значимости признаков в модели по отдельности;
- тест по неотобранным образцам out – of – bag;
- высокая параллелизуемость.

Недостатки – размерность модели очень большая. O(K), где K количество деревьев.

Random Forest (по причине независимого построения глубоких деревьев) требует весьма много ресурсов, а ограничение на глубину повредит точности (для решения сложных задач нужно построить много глубоких деревьев). Можно заметить, что время обучения деревьев возрастает приблизительно линейно их количеству.

3.4.3 Нейронная сеть

Нейронный сети – это раздел искусственного интеллекта, в котором для обработки сигналов используются явления, аналогичные происходящим в нейронах живых существ. Важнейшая особенность сети, свидетельствующая о её широких возможностях и огромном потенциале, состоит в параллельной обработке информации всеми звеньями. [27]

При огромном количестве межнейронных связей это позволяет значительно ускорить процесс обработки информации. Во многихслучаях становится возможным преобразование сигналов в реальном времени.

Кроме того, при большом количестве межнейронных соединений сеть приобретает устойчивость к ошибкам, возникающих на некоторых линиях. Функции повреждённых связей берут на себя исправные линии, в результате чего деятельность сети не претерпевает существенных возмущений. Другое не менее важное свойство — способность к обучению и обобщению накопленных знаний.

Нейронная сеть обладает чертами искусственного интеллекта Натренированная на ограниченном множестве данных сеть способна обобщать полученную информацию и показывать хорошие результаты на данных, не использовавшихся в процессе обучения. ИНС в практических приложениях, как правило, используются в качестве подсистемы управления или выработки решений, передающей исполнительный сигнал другим подсистемам, имеющим иную методологическую основу.

Функции, выполняемые ИНС подразделяются на несколько групп:

- аппроксимация:
- классификация и распознавание образов;
- прогнозирование;
- идентификация и оценивание;
- ассоциативное управление.

В области прогнозирования задача сети формулируется как предсказание будущего поведения системы по имеющейся последовательности её предыдущих состояний. По информации о значениях переменной x в моменты времени, предшествующие прогнозированию, сеть вырабатывает решение о том, чему должно быть равно оцениваемое значение исследуемой последовательности в текущий момент времени.

3.5 Описание моделей для оценки рисков летальности от ССЗ используемые в современной медицине

3.5.1 SCORE

Шкала SCORE предназначена для оценки десятилетнего риска смерти от сердечно сосудистых осложнений, включая:

- коронарные события;
- мозговой инсульт;
- аневризма брюшной аорты.

SCORE также предназначен для оценки комбинации фатальных и нефатальных рисков. Оценка нефатальных событий менее определенная, так как постановка диагноза может различаться в разных странах, в разных учереждениях для диагностики и клинических подходах. Всё же оценка фатальных и нефатальных рисков осложнений ССЗ важна при профилактических мероприятий, а шкала SCORE даёт возможность это сделать. [28]

Есть 2 варианта шкалы SCORE для стран с низким уровнем сердечно сосудистой смертности,для стран с высоким и очень высоким уровнем сердечно сосудистой смертности.

Шкала SCORE для стран с низким уровнем уровнем риска летальности от ССЗ указана на рисунке 3.2.

Рисунок 3.2 Шкала SCORE для стран с низким риском ССЗ

Для России рекомендуется использовать шкалу SCORE для стран с высоким и очень высоким риском, так как смертность населения от ССЗ превышает 450 случаев на 100000 человек.[29]

SCORE для стран с высоким и очень высоким риском летальности от ССЗ использует такие предикторы как:

- общий холестерин;
- систолическое артериальное давление (САД);
- факт курения (Да / Нет);
- возраст;
- пол.

Градация рисков SCORE:

- риск менее 1% считается низким;
- в пределах больше либо равно 1% и до 5% – умеренным;
- от 5% включительно и до 10% – высоким;
- выше 10% включительно – очень высоким.

Пример реализации данного калькулятора SCORE для стран с высоким риском летальности от ССЗ показан на рисунке 3.3.

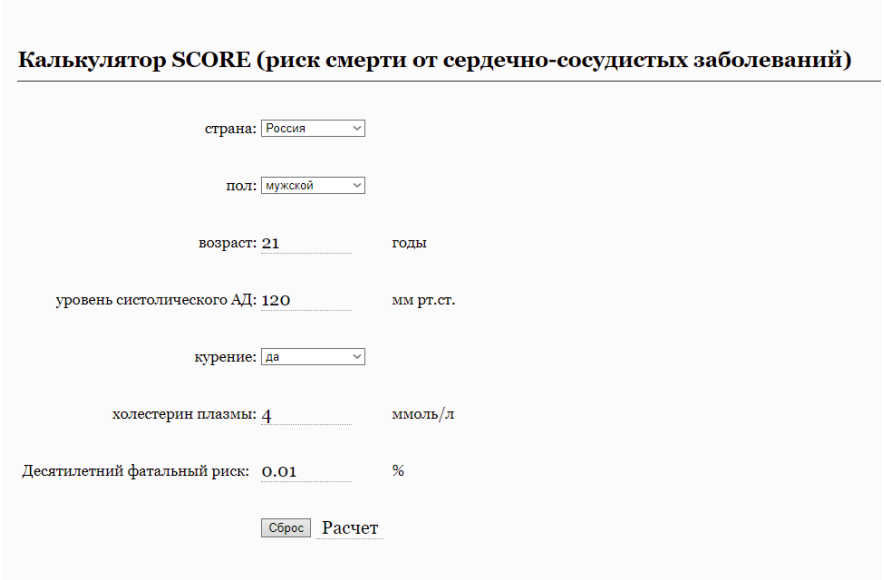

Рисунок 3.3 – Реализация калькулятора SCORE [30]

Результатом работы калькулятора является десятилетний фатальный риск в процентах.

3.5.2 ASSIGN SCORE

ASSIGN SCORE – оценка сердечно – сосудистого риска, разработанная в 2006 году в Dundee University, Шотландия. Разработан на основе модели Framingham.

ASSIGN может быть адаптирован для использования за пределами Шотландии, если не указывать Scottish Postcode.

Для оценки риска ASSIGN SCORE использует такие предикторы как:

- возраст;
- пол;
- диабет;
- ревматоидный артрит;
- количество выкуренных сигарет в день;
- САД;
- общий холестерин;
- HDL – холестерин.

Градация рисков ASSIGN SCORE:

- менее 10 % – умеренный;
- от 10% включительно и до 20% – высоким;
- выше 20% включительно – очень высоким.

Пример реализации калькулятора ASSIGN SCORE показан на рисунке 3.4.

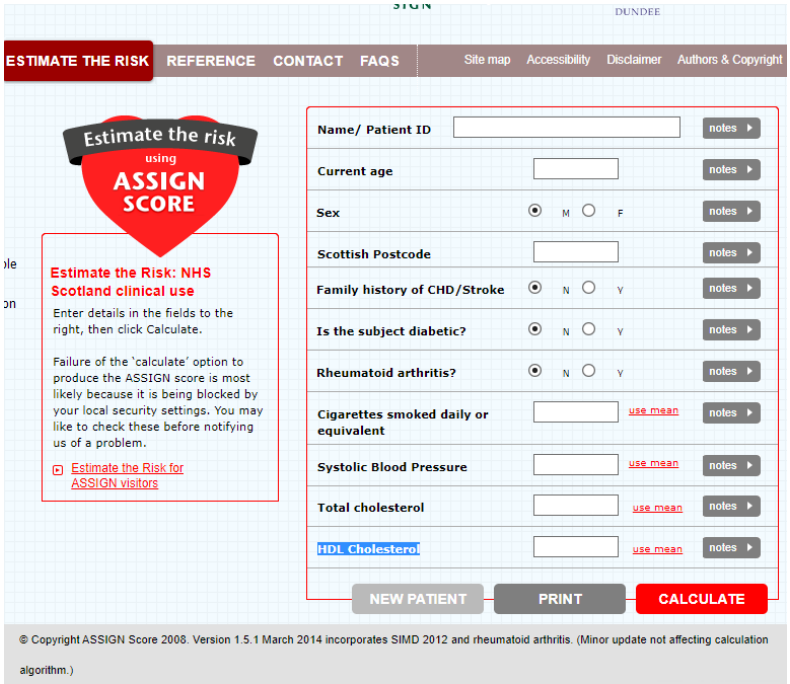

Рисунок 3.4 – Реализация калькулятора ASSIGN SCORE [31]

Результатом работы калькулятора является десятилетний фатальный риск в процентах.

3.6 Результаты исследования моделей по оценке рисков летальности от ССЗ

3.6.1 Предварительный анализ данных для машинного обучения

По результату были изучены все прилагающиеся документы к проекту:

- документ «Комментарии для базы.docx»;
- документ «Комментарии к датасету Корейцы и Славяне.docx»;
- статья «РФФИ_МК_901.1 Разработка методов верификации и прогнозирования рисков ССЗ.docx»;
- документ «Задание по анализу данных на жесткость сосудов.docx»;
- Документ «анкета.docx».

Вся документация находится на google drive «МК_Цифровая Медицина».

Были изучены все dataset – ы проекта:

- dataset «Данные ЭССЕ Приморский край.xlsx»;

- dataset «БАЗА корейцы и славяне жестк.xlsx».

Для того, чтобы считать данные из .xlsx документа применялась функция read_excel из библиотеки pandas. Данные принимают тип DataFrame. Код приведен на рисунке 3.5.

```
VLADIVOSTOK = pandas.read_excel('VLADIVOSTOK.xlsx', sheetname_name='Sheet1')
BD_cor_and_slav = pandas.read_excel('БАЗА корейцы и славяне жестк.xlsx', sheetname_name='Sheet1')
Data_ESSE = pandas.read_excel('Данные ЭССЕ Приморский край адаптированно.xlsx', sheetname_name='Sheet1')
ESSE_RF = pandas.read_excel('esse.xlsx', sheetname_name='Sheet1')
```

Рисунок 3.5 – Код обращения к dataset – ам

Чтобы получить количество атрибутов и количество кортежей вызывается метод shape функции DataFrame из библиотеки pandas. Код приведен на рисунке 3.6.

```
VLADIVOSTOK.shape
BD_cor_and_slav.shape
Data_ESSE.shape
ESSE_RF.shape
```

Рисунок 3.6 – Код обращения к размерности DataFrame

Результаты первичного просмотра dataset – ов приведены в таблице 1.

Таблица 3.1 – Размерности всех dataset – ов

Наименование	Количество атрибутов	Количество кортежей
Данные ЭССЕ Приморский край.xlsx	2132	206
БАЗА корейцы и славяне жестк.xlsx	489	61

После изучения всех dataset – ов и получения информации о размерности, dataset «Данные ЭССЕ Приморский край.xlsx» был выбран для дальнейшей работы над проектом, так как удовлетворял всем нужным требованиям для реализации следующих заданий.

3.6.2 Основные статистические данные по ключевым атрибутам

Для построения гистограммы по возрасту используем функцию distplot библиотеки seaborn.

Код для построения гистограммы приведен на рисунке 3.7.

```
sns.distplot(esse['Возраст'])
```

Рисунок 3.7 – Реализация построения гистограммы в библиотеке seaborn

Результат выполнения кода приведен на рисунке 3.8.

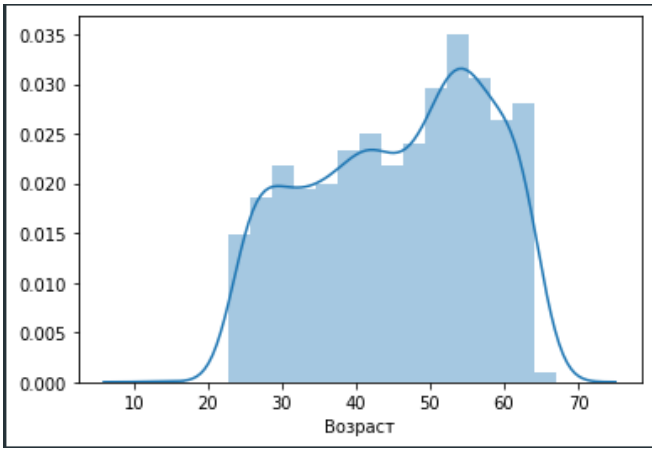

Рисунок 3.8 – Гистограммы обследуемых по возрасту

Исходя из гистограммы, мы видим, что у один из обследуемых очень молодой. Делаем вывод, что он попал в dataset по ошибке.

Построим диаграмму «Ящик с усами» по возрасту и полу обследуемых для того, чтобы увидеть медиану и точки экстремума. Для этого воспользуемся функцией boxplot библиотеки seaborn.

Код построения диаграммы «Ящик с усами» приведен на рисунке 3.9.

```
sns.boxplot(x=esse['Пол'],y=esse['Возраст'])
plt.ylabel('Возраст')
plt.xlabel('Пол')
plt.ylim(1,80)
plt.show()
```

Рисунок 3.9 – Построение диаграммы «Ящик с усами» при помощи библиотеки seaborn

Результат выполнения кода приведен на рисунке 3.10

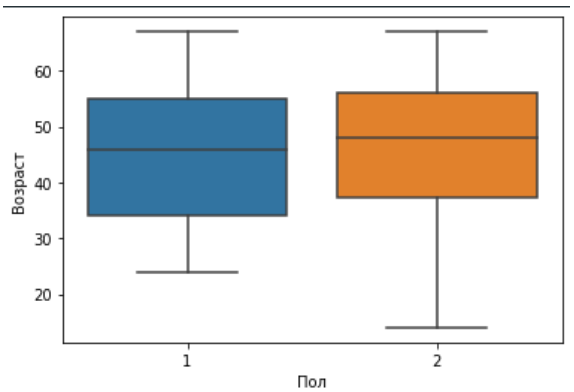

Рисунок 3.10 – Диаграмма «Ящик с усами» по Полу и Возрасту

Посмотрев на диаграмму делаем выводы, что медианы у мужчин и женщин примерно равны, а обследуемый с очень низким показателем возраста является женщиной.

Узнаем какое распределение имеет возраст у мужчин и возраст у женщин. Для этого воспользуемся сначала применим тест Шапиро – Уилка, реализованный в виде функции shapiro в библиотеке scipy.stats. Если результат больше или равен 0.05, то гипотезу о нормальном распределении принимаем, если нет – отвергаем. Используем тест Колмогорова – Смирнова для проверки гипотезы о том, что выборки принадлежат к какому – то одному распределению. Далее проверяем гипотезу о равенстве средних значений выборки при помощи тестов Стьюдента и Манн – Уитни.

Код реализации тестов приведен на рисунке 3.11.

```
sc.shapiro(man) (0.9474309086799622, 4.380256517052488e-17)
sc.shapiro(woman) (0.9529871344566345, 1.189857822879932e-19)
sc.ks_2samp(man,woman) Ks_2sampResult(statistic=0.0699686764433029, pvalue=0.012055834204006133)

sc.ttest_ind(man,woman) Ttest_indResult(statistic=-3.059539966782995, pvalue=0.0022444002411291285)
sc.mannwhitneyu(man,woman) MannwhitneyuResult(statistic=508477.0, pvalue=0.00157326868049270 78)
```

Рисунок 3.11 – Основные тесты для оценки статистических данных

Исходя из полученных результатов делаем вывод, что возраст у мужчин и возраст у женщин не имеет нормальное распределение исходя из результатов теста Шапиро – Уилка (результат больше или равен 0.05). Результаты тестов Стьюдента и Манна – Уитни показывают, что гипотезу о том, что средние 2 – х выборок равны отклоняем.

Создадим гистограмму по ИМТ. Для того, чтобы увидеть его распределение. Чтобы создать гистограмму используем метод hist объекта DataFrame.

Код создания гистограммы приведен на рисунке 3.12.

```
esse['BMI'].plot.hist()
plt.ylabel('Частота')
plt.xlabel('ИМТ')
plt.show()
```

Рисунок 3.12 – Реализация кода создания гистограммы атрибута «ИМТ»

На рисунке 3.13 показан результат выполнения кода.

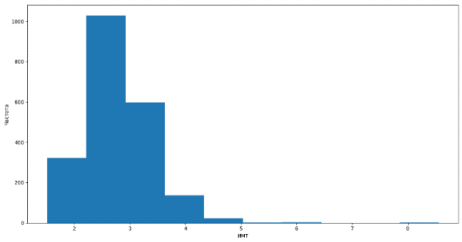

Рисунок 3.13 – Гистограмма по атрибуту «ИМТ»

На данной гистограмме видно, что есть аномалия у атрибута. Несколько обследуемых имеют слишком высокое значение ИМТ, настолько большое, что, скорее всего, это ошибка записи.

Построим диаграмму «Ящик с усами» по полу и ИМТ. Диаграмма указана на рисунке 3.14.

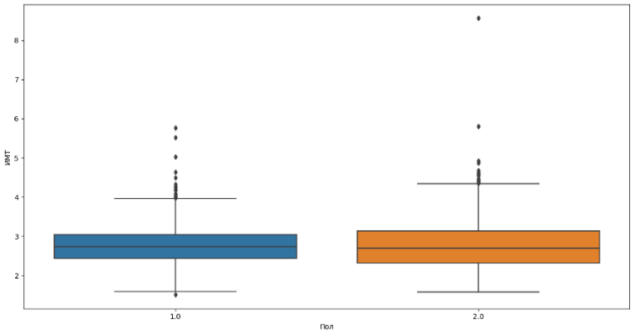

Рисунок 3.14 – Диаграмма «Ящик с усами» по ИМТ

На диаграмме видим, что медианы у мужчины и женщины примерно равны, как и точки экстремума. А также видим, что у нескольких людей действительно слишком завышенное значение ИМТ.

Проверим гипотезу о равенстве 2 – средних с помощью тестов Стьюдента и Манн – Уитни. Код реализации тестов приведен на рисунке 3.15.

```
sc.shapiro(Bman) (0.9585165977478027, 6.255844841175466e-15)
sc.shapiro(Bwoman) (0.9365643858909607, 1.1924758444699525e-22)
sc.ks_2samp(Bman,Bwoman) Ks_2sampResult(statistic=0.09926368840737329, pvalue=7.669474774230078e-05)

sc.ttest_ind(Bman,Bwoman) Ttest_indResult(statistic=0.37031684091495276, pvalue=0.7111835605096826)
sc.mannwhitneyu(Bman,Bwoman) MannwhitneyuResult(statistic=515449.5, pvalue=0.03648042513700399)
```

Рисунок 3.15 – Получение основных статистических данных по ИМТ

Результаты показывают нам, что среднее значение ИМТ у женщин и у мужчин равное, так как результаты теста Стьюдента больше 0.05.

3.6.3 Описание значимых для готовых моделей оценки риска от ССЗ факторов

В первоначальном наборе данных может быть много разных переменных, применение в алгоритме слишком большого их числа ведет к замедлению вычислений или к большим ошибочным предсказаниям из – за информационного шума.

Необходимо выполнить описание всех данных нужных для корректного функционирования готовых моделей, чтобы иметь представление с данными какого качества будет происходить дальнейшая работа. Требуется описать ключевые атрибуты модели оценки риска SCOREтакие как:

- пол;
- возраст;
- общий холестерин
- САД
- факт курения

Для ASSIGN SCORE необходимо дополнить описание такими факторами как:

- HDL – холестерин
- инфаркт миокарда
- инсульт
- ревматоидный артрит
- сахарный диабет

Для того, чтобы узнать показатели у живых и умерших делим DataFrame по атрибуту IsDeath. На рисунке 3.16 предоставлен код разделения на живых и мертвых.

```
VLADIVOSTOK=pd.read_excel('C:/Users/kraia/Desktop/Перрессия/Таблицы/VLADIVOSTOK.xlsx',sheetname_name='Sheet1')
ESSE=pd.read_excel('C:/Users/kraia/Desktop/Перрессия/Таблицы/esse.xlsx',sheetname_name='Sheet1')
ESSE['SEX']=VLADIVOSTOK['SEX']

Desse=ESSE['IsDeath']
GetS=SCORE_series.GetSer()
ESSE_DF=pd.DataFrame([Desse,GetS,ESSE['SEX'],ESSE['Возраст'],ESSE['общий холестерин'],ESSE['среднее САД'],ESSE['4_1']]).T.dropna()
ESSE_DF.columns = ['Death','SCORE','Пол','Возраст','TC','САД','Курение']
living=ESSE_DF.where(ESSE_DF['Death']==0).dropna()
dead=ESSE_DF.where(ESSE_DF['Death']==1).dropna()
```

Рисунок 3.16 – DataFrame разделенный на живых и умерших

Чтобы узнать диапазон значений атрибута используется метод объекта Series unique, который оставляет только уникальные значения выборки, возвращая массив значений numpy. array. Функция tolist преобразует array в list для удобства чтения.

Неполные данные мешают анализу и при любой возможности с ними нужно разобраться одним из следующих способов:

- приближение – если пропущено значение бинарного или категориального типа его можно заменить самым типичным значением (модой) переменной. А для целочисленных или непрерывных переменных используется медиана;
- вычисление пропущенные значения – также могут быть вычислены с применением более продвинутых алгоритмов обучения с учителем. Хотя такие вычисления требуют времени, они обычно приводят к более точным оценкам неполных значений;
- удаление – в качестве последнего средства строки с неполными значениями могут быть удалены. Тем не менее этого обычно избегают, чтобы не уменьшать объем данных, доступных для анализа.

Для получения значений NaN в Series используется метод объекта DataFrame isnull, который возвращает True / False в зависимости от того, является ли значение в ячейке NaN, метод sum суммирует все значения True.

Узнать количество уникальных значений можно с помощью метода DataFrame groupby, а метод size задает группировку по количеству.

Код описания атрибута «Пол» приведен на рисунке 3.17.

```
ESSE['SEX'].unique().tolist()
ESSE['SEX'].isnull().sum()
ESSE.groupby('SEX').size()
dead.groupby('Пол').size()
```

Рисунок 3.17 – Код для описания атрибута «Пол»

Результат описания атрибута «Пол» приведен в таблице 2.

Таблица 3.2 – Описание ключевых знаний о атрибуте «Пол»

Описание	Результат
английское наименование	SEX
русское наименование	пол
значение	дискретное
диапазон	1, 2
расшифровка	1 – мужчина; 2 – женщина
NaN	0
количество мужчин	874
количество женщин	1258
количество умерших мужчин	23
количество умерших женщин	19

Исходя из результатов таблицы 2 делаем вывод, что аномалий у атрибута «Пол» нет. Количество умерших мужчин преобладает над количеством умерших женщин.

Описывая возраст проверяем экстремумы, количество значений NaN, средний возраст мужчин и средний возраст женщин. Чтобы узнать средний возраст воспользуемся группировкой через метод mean, который группирует по среднему значению. Для нахождения экстремумов используем стандартные функции min и max.

Код описания атрибута «Возраст» приведен на рисунке 3.18.

```
ESSE['Возраст'].min()
ESSE['Возраст'].max()
ESSE['Возраст'].isnull().sum()
ESSE['Возраст'].mean()
sc.ttest_ind(living['Возраст'],dead['Возраст'])
tistic=-4.755351947493751, pvalue=2.1149673832898073e-06)

sc.mannwhitneyu(living['Возраст'],dead['Возраст'])
uResult(statistic=24241.5, pvalue=4.4835070240600875e-07)
```

Рисунок 3.18 – Код для описания атрибута «Возраст»

Результат описания атрибута «Возраст» приведен в таблице 3.

Таблица 3.3 – Описание ключевых знаний о атрибуте «Возраст»

Описание	Результат
русское наименование	возраст
значение	непрерывное
экстремумы	min – 14(*) max – 67
единица измерения	год жизни
NaN	0
средний возраст женщин	46.3
средний возраст мужчин	44.8

Исходя из результатов таблицы 3 делаем вывод, что возможный минимум = 23, а 14 ошибка записи (или человек попал в dataset по ошибке), так как диапазон исследования ЭССЕ – РФ от 25 до 65. Для проверки равенства средних значений возраста живых и умерших воспользуемся тестами Стьюдента и Манна – Уитнни реализованными в библиотеке scipy функциями ttest_ind() и mannwhitneyu().

t-критерий Стьюдента – общее название для класса методов статистической проверки гипотез (статистических критериев), основанных на распределении Стьюдента. Наиболее частые случаи применения t-критерия связаны с проверкой равенства средних значений в двух выборках. [32]

U-критерий Манна – Уитни – статистический критерий, используемый для оценки различий между двумя независимыми выборками по уровню какого-либо признака, измеренного количественно. Позволяет выявлять различия в значении параметра между малыми выборками. [33]

По результатам тестов p – value меньше 0.05, из этого делаем вывод, что гипотезу о равенстве средних отвергаем.

Чтобы описать атрибут «Общий холестерин» необходимо рассмотреть среднее значение холестерина у всей обследуемой когорты, среднее значение холестерина у живых/мертвых.

Код описания атрибута «Общий холестерин» приведен на рисунке 3.19

```
ESSE['общий холестерин'].min()
ESSE['общий холестерин'].max()
ESSE['общий холестерин'].isnull().sum()
ESSE['общий холестерин'].mean()
sc.ttest_ind(living['TC'],dead['TC']) Ttest_indResult(statistic=0.4151377031047831, pvalue=0.6780833277742389)
sc.mannwhitneyu(living['TC'],dead['TC']) MannwhitneyuResult(statistic=42762.0, pvalue=0.43032679886516834)
```

Рисунок 3.19 – Код для описания атрибута «Общий холестерин»

Результат описания атрибута «Общий холестерин» приведен в таблице 4.

Таблица 3.4 – Описание ключевых знаний о атрибуте «Общий холестерин»

Описание	Результат
русское наименование	Общий холестерин
значение	непрерывное
экстремумы	Min – 2.46 Max – 11.49
единица измерения	ммоль/л
NaN	15
нулевые значения	0
среднее значение холестерина	5.62

Исходя из результатов таблицы 4 делаем вывод, что аномалий у атрибута «Общий холестерин» нет. По результатам тестов Стьюдента и Манна – Уитни p – value больше 0.05, из этого делаем вывод, что гипотезу о равенстве средних принимаем.

Для того, чтобы описать атрибут «САД» необходимо рассмотреть среднее значение холестерина у всей обследуемой когорты, среднее значение артериального давления у живых/мертвых.

Код описания атрибута «САД» приведен на рисунке 3.20

```
ESSE['среднее САД'].min()
ESSE['среднее САД'].max()
ESSE['среднее САД'].isnull().sum()
ESSE['среднее САД'].mean()
sc.ttest_ind(living['САД'],dead['САД']) Ttest_indResult(statistic=-2.8910492784089237, pvalue=0.0038788627398563254)
sc.mannwhitneyu(living['САД'],dead['САД']) MannwhitneyuResult(statistic=32887.5, pvalue=0.003457141557984992)
```

Рисунок 3.20 – Код для описания атрибута «САД»

Результат описания атрибута «САД» приведен в таблице 5.

Таблица 3.5 – Описание ключевых знаний о атрибуте «САД»

Описание	Результат
русское наименование	среднее САД
значение	непрерывное
экстремумы	Min – 82 Max – 240
единица измерения	мм рт. ст.
NaN	6
нулевые значения	0
среднее значение САД	135

В результате данных таблицы 5 делаем заключение, что в атрибуте «САД», скорее всего, имеется аномалия, т.к. среднее верхнее давление 240 у человека, который фактически должен являться здоровым по условию проведения эксперимента ЭССЕ – РФ, крайне маловероятно. По результатам тестов делаем вывод, что гипотезу о равенстве средних значений отвергаем, т.к. p – value меньше 0.05.

Получая основные статистические данные по атрибуту «Факт курения» необходимо рассмотреть количество курящих и некурящих. И рассмотреть количество курящих среди обследуемых, которые считаются умершими.

Результат описания атрибута «Факт курения» приведен в таблице 6.

Таблица 3.6 – Описание ключевых знаний о атрибуте «Факт курения»

Описание	Результат
наименование	Факт курения
значение	дискретное
диапазон	1, 2, 3
расшифровка	1 – не курю 2 – бросил 3 – курю
NaN	0
количество некурящих	1666

Продолжение таблицы 6

количество курящих	466
количество курящих среди умерших	18 из 42

Отталкиваясь от результатов таблицы 6 делаем вывод, что аномалий в атрибуте «Факт курения» нет. Также считаем, что значение 2 – бросил приравниваем к 1 – не курю, потому что в готовых моделях будет проверять 1 – курю, 2 – не курю.

Фактор риска «HDL – холестерин» оценивается путем рассмотрения среднего значение HDL – холестерина у всей обследуемой когорты, количество значений NaN, количество нулевых значений и сравнением средних значений у живых и умерших с помощью тестов Стьюдента и Манна – Уитни.

Код описания атрибута «HDL – холестерин» приведен на рисунке 3.21.

```
ESSE['HDL'].min()
ESSE['HDL'].max()
ESSE['HDL'].isnull().sum()
ESSE['HDL'].mean()
sc.ttest_ind(living['HDL'],dead['HDL']) Ttest_indResult(statistic=1.83190716712[illegible]4205, pvalue=0.06710621109582554)
sc.mannwhitneyu(living['HDL'],dead['HDL']) MannwhitneyuResult(statistic=36249.0, pvalue=0.03280782689083844)
```

Рисунок 3.21 – Код для описания атрибута «HDL – холестерин»

Результат описания атрибута «HDL – холестерин» приведен в таблице 7.

Таблица 3.7 – Описание ключевых знаний о атрибуте «HDL – холестерин»

Описание	Результат
русское наименование	HDL
значение	непрерывное
экстремумы	min – 0.6 max – 2.98
единица измерения	ммоль/л
NaN	15
нулевые значения	0
среднее значение холестерина	1.44

По результатам таблицы 7 делаем вывод, что аномалий у атрибута «HDL – холестерин» нет. По результатам тестов видим, что p – value меньше 0.05, из этого делаем вывод, что гипотезу о равенстве средних между живыми и умершими отвергаем.

Рассмотрим такие факторы, как диагнозы «Ревматоидный артрит», «Инфаркт», «Инсульт» и «Сахарный диабет». Для калькулятора нам необходимо понимать количество положительных ответов и количество пропущенных данных.

Код описания диагнозов приведен на рисунке 3.22.

```
ESSE['Название болезни'].unique().tolist()
ESSE['Название болезни'].isnull().sum()
ESSE.groupby('Название болезни').size()
```

Рисунок 3.22 – Код для описания диагнозов

Результат описания диагнозов приведен в таблице 8.

Таблица 3.8 – Описание ключевых знаний о диагнозах

описание	«ревматоидный артрит»	«инсульт»	«сахарный диабет»	«инфаркт»
значение	дискретное	дискретное	дискретное	дискретное
диапазон	[1,2,3]	[1,2,9]	[1,2,9]	[1,2,3]
расшифровка	1-нет 2-да 3-не знаю	1-нет 2-да 9-не знаю	1-нет 2-да 9-не знаю	1-нет 2-да 3-не знаю
NaN	0	413	0	410
Кол-во положительных ответов	215	384	117	326

Из данной таблицы делаем выводы, что аномалий у диагнозов нет, но такие атрибуты, как «Инсульт» и «Инфаркт» имеют достаточно большое количество пропусков, что говорит о возможном некачественном вводе данных.

3.6.4 Показатели у 10% обследуемых с самыми большими рисками летальности по моделям SCORE и ASSIGN SCORE

Проводим проверку для модели SCORE. Для оценки среднего значения риска обследуемых в рамках ЭССЕ – РФ используем функцию mean библиотеки numpy к Series «Риск по шкале SCORE», для оценки медианы используем функцию median библиотеки statistics. Результат приведен на рисунке 3.23.

```
GetS=SCORE_series.GetSer()
np.mean(GetS) 2.1524774988157347
median(GetS) 2.975
```

Рисунок 3.23 – Среднее и медиана по шкале SCORE обследуемых в рамках ЭССЕ – РФ

Далее объединяем атрибут «Смерть», который показывает, кто из обследуемых умер за период проведения проекта, с Series «Риск смерти по шкале SCORE». Дальнейшие манипуляции с данными будут произведены с 10 процентами обследуемых с наивысшим значением риска смерти от ССЗ по шкале SCORE.

На рисунке 3.24 приведен код для оценки статистических данных обследуемых с добавленным риском по SCORE.

```
Mertv=df1.where(df1['IsDeath']==1).dropna()
Jiv=df1.where(df1['IsDeath']==0).dropna()
sort_df1=df1.sort_values(by=['SCORE'],ascending=False)
prst_10=sort_df1[1:215]
prst_10['IsDeath'].count() 214
min(prst_10['SCORE']) 5.94
max(prst_10['SCORE']) 56.68
np.mean(prst_10['SCORE']) 10.621074766355148
median(prst_10['SCORE'])
prst_10['IsDeath'].where(prst_10['IsDeath']==0).dropna().count() 199
prst_10['IsDeath'].where(prst_10['IsDeath']==1).dropna().count() 15
sc.ttest_ind(Mertv['общий холестерин'],Jiv['общий холестерин']) Ttest_indResult(statistic=-0.4151377031047831, pvalue=0.6780833277742389)
sc.mannwhitneyu(Mertv['общий холестерин'],Jiv['общий холестерин']) MannwhitneyuResult(statistic=42762.0, pvalue=0.43032679886516834)
sc.ttest_ind(Mertv['среднее САД'],Jiv['среднее САД']) Ttest_indResult(statistic=2.8910492784089237, pvalue=0.0038788627398563254)
sc.mannwhitneyu(Mertv['среднее САД'],Jiv['среднее САД']) MannwhitneyuResult(statistic=32887.5, pvalue=0.003457141557984992)
```

Рисунок 3.24 – Реализация кода для оценки статистических данных 10 процентов обследуемых с наивысшим значением SCORE

10 процентов с наивысшим процентом риска по SCORE – 214 человек из 2132.

Минимальное значение – 5.94

Максимальное значение – 56.68

Среднее – 10.62

Медиана – 8.98

Из этих 10 процентов реально умерли 15 обследуемых, остальные 199 живы.

На таблице 9 показатели умерших людей из 10 процентов обследуемых с наивысшим показателем риска смерти от ССЗ по SCORE.

Таблица 3.9 – Данные умерших из десятипроцентной выборки

ID	SCORE	Общий холестерин	САД
1318	44.68	5.34	215
1176	21.34	7.42	175

Продолжение таблицы 9

1471	16.78	7.98	128
817	15.79	6.76	240
1390	13.50	8.03	161
1656	12.43	6.77	165
348	12.23	5.82	169
1835	10.45	4.64	171
561	9.03	3.52	142
828	8.29	5.73	160
1639	8.19	6.34	138
591	7.60	7.40	123
964	6.97	5.25	135
298	6.19	3.18	160

Исходя из данных таблицы 9 видим, что разброс риска по SCORE среди умерших достаточно большой. Минимальное значение – 6.19, а максимальное – 44.68.

По результатам тестов Стьюдента и Манн – Уитни делаем вывод, что средние значения холестерина и средние значения артериального давления равны. Так как результат больше или равен 0.5.

Проводим проверку для модели ASSIGN SCORE. Получаем значения ASSIGN SCORE на обследуемой когорте. Реализованный series добавляем к общему DataFrame со всеми атрибутами готовой модели. Дальнейшие действия с данными будут произведены с 10 % обследуемых с наивысшим значением риска смерти от ССЗ по шкале ASSIGN SCORE.

Рассмотрим основные статистические данные для модели ASSIGN SCORE, они указаны на рисунке 3.25.

```
df=pd.DataFrame([Desse,GetS,esse['Возраст'],esse['общий холестерин'],esse['среднее САД'],df1['4_5'],esse['HDL']
df.columns = ['Death','ASSIGN_SCORE','Возраст','TC','САД','Индекс курения','HDL','СД','PA','Инфаркт','Инсульт']
GetS.where(GetS==0).dropna().count() 0
df.shape (2111, 11)
mean(df['ASSIGN_SCORE']) 7.317039317858824
median(df['ASSIGN_SCORE']) 5.14

sort_df1=df.sort_values(by=['ASSIGN_SCORE'],ascending=False)
prst_10=sort_df1[1:215]
prst_10.count()

prst_10['ASSIGN_SCORE'].min() 16.18
prst_10['ASSIGN_SCORE'].max() 52.67
prst_10['ASSIGN_SCORE'].mean() 23.16939252336448
prst_10['ASSIGN_SCORE'].median() 21.134999999999998

d_prst_10=prst_10.where(prst_10['Death']==1).dropna()
d_prst_10['ASSIGN_SCORE'].min() 16.3
d_prst_10['ASSIGN_SCORE'].max() 34.95
```

Рисунок 3.25 – Реализация кода для оценки статистических данных 10 % обследуемых с наивысшим значением ASSIGN SCORE

10 процентов с наивысшим процентом риска по ASSIGN SCORE – 214 человек из 2132.

Минимальное значение – 16.18

Максимальное значение – 52.67

Среднее – 23.16

Медиана – 21.13

Из этих 10 процентов реально умерли 15 обследуемых, остальные 199 живы.

На таблице 10 показатели умерших людей из 10 % обследуемых с наивысшим показателем риска смерти от ССЗ по ASSIGN SCORE.

Таблица 3.10 – Данные умерших из десятипроцентной выборки

	death	ASSIGN SCORE	возраст	холестерин	САД	инд. курения	HDL	СД	PA	Семейная история
817	1	34.95	43	6.76	240	5	0.9	3	3	1
1176	1	33.46	56	7.42	175	20	1.27	1	1	0
1471	1	28.35	62	7.98	128	15	1.29	3	3	0
816	1	26.83	63	6.67	140	0	1.19	3	3	1
1390	1	26.55	60	8.03	161	0	1.41	2	1	0
590	1	26.31	62	7.61	153	0	1.07	2	1	1
348	1	25.37	62	5.82	169	0	0.98	2	1	0
1639	1	24.68	63	6.34	138	0	1.19	1	1	1
1318	1	21.24	61	5.34	215	10	2.44	1	1	0
402	1	20.44	50	6.30	144	25	1.26	1	1	1
1656	1	18.74	61	6.77	165	0	1.82	3	3	0
1835	1	18.58	64	4.64	171	2	0.92	1	1	0
591	1	18.19	63	7.4	123	0	1.43	1	1	0
964	1	16.57	56	5.25	135	15	0.81	3	3	0
828	1	16.30	52	5.73	160	15	1.17	3	3	0

Исходя из данных таблицы 10 видим, что риски по ASSIGN SCORE среди умерших имеют существенные отличия в показателях, сравнивая с рисками по SCORE. Минимальное значение – 16.3, а максимальное – 34.95.

3.6.5 Исследование моделей, используемых в медицинской практике

Для исследования необходимо реализовать алгоритм работы калькуляторов SCORE и ASSIGN SCORE, проверить статистически значимые данные, вычислить точность, специфичность, чувствительность и площадь под ROC кривой.

Статистически значимые данные проверяем при помощи тестов Стьюдента и Манн Уитни. Для реализации используются функции mannwhiteyu() и ttest_ind() из библиотеки scipy.

t – критерий Стьюдента обшее название для статистических тестов, в которых статистика критерия имеет распределение Стьюдента. Наиболее часто t криетрий применяется для проверки равенства средних значений в двух выборках. Нулевая гипотеза предполагает,что средние равны.

U критерий Манна Уитни статистический критерий, используемый для оценки различий между двумя независимыми выборками по уровню какого либо признака, измеренного количественно. Позволяет выявлять различия между малыми выборками.

Для вычисления точности строится confusion matrix. Для реализации используем функцию confusion_matrix() из библиотеки sklearn.

Confusion matrix – матрица пересечений, в которой указывается количество ложных положительных значений (FP), ложных отрицательных значений (FN), истинных положительных (TP) и отрицательных (TN) значений. P – количество положительных значений, N – количество отрицательных значений. Confusion matrix изображена на рисунке 3.26.

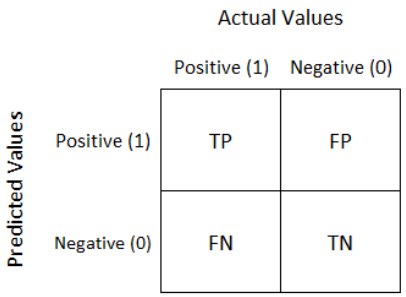

Рисунок 3.26 Представление confusion matrix

Точность модели рассчитывается по формуле (3.4):

$$accuracy = \frac{TP + TN}{P + N}. \tag{3.4}$$

Специфичность модели рассчитывается по формуле (3.5):

$$specificity = \frac{TN}{N}. \quad (3.5)$$

Чувствительность модели рассчитывается по формуле (3.6):

$$sensetivity = \frac{TP}{P}. \quad (3.6)$$

Площадь под ROC-кривой (AUC) является агрегированной характеристикой качества классификации, не зависящей от соотношения цен ошибок. Чем больше значение AUC, тем «лучше» модель классификации. [1]

AUC вычисляется по формуле (3.7):

$$AUC = \frac{1 + TPR - FPR}{2}, \quad (3.7)$$

где TPR – sensetivity, а FPR – 1-specificity.

SCORE применимы к 2111 из 2132 обследуемым соответственно из – за пропуска нужных для расчета данных в датасете. Реализация алгоритма модели SCORE указана в Приложении А. Данные по риску смерти обследуемой выборки указаны в таблице 11.

Таблица 3.11 риск смерти в ближайшие 10 лет по SCORE

Показатели	Значения
среднее SCORE	2.15 процентов
медиана SCORE	0.72 процентов
тест Стьюдента между живыми и мертвыми в выборке	p – value – 0.000000000006

По результатам из таблицы 11 тестов Стьюдента и Манна – Уитни видим, что p – value меньше 0.05, значит гипотезу о равенстве средних отклоняем.

ASSIGN SCORE применимы к 2111 из 2132 обследуемым соответственно из – за пропуска нужных для расчета данных в датасете. Реализация алгоритма модели ASSIGN SCORE указана в Приложении В. Данные по риску смерти обследуемой выборки указаны в таблице 12.

Таблица 3.12 риск смерти в ближайшие 10 лет по ASSIGN SCORE

Показатели	Значения
среднее ASSIGN SCORE	7.31 процентов
Медиана ASSIGN SCORE	5.14 процентов
Тест Манна – Уитни между живыми и мертвыми в выборке	p – value – 0.00000007

По результатам из таблицы 12 тестов Стьюдента и Манна – Уитни видим, что p – value меньше 0.05, значит гипотезу о равенстве средних отклоняем.

3.6.6 Разработка собственных моделей оценки риска летальности от ССЗ

Вычисляем при помощи confusion matrix точность моделей SCORE и ASSIGN SCORE. Результаты вычислений представлены в таблице 13.

Таблица 3.13 – Основные статистические данные моделей

	SCORE	ASSIGN SCORE
точность	0.86	0.92
специфичность	0.87	0.94
чувствительность	0.4	0.23
AUC	0.64	0.58

Исследование подразумевает разработку более чувствительных моделей отталкиваясь от результатов готовых моделей, приведенных в таблице 13, используемых в медицине.

Обучение происходит на предикторах шкалы SCORE. Основной качественной меркой для оценки модели будут средние показатели на тестовой выборке в цикле. В каждой итерации цикла строится confusion matrix, вычисляются чувствительность, специфичность и AUC.

Логистическая регрессия реализуется с помощью функции LogisticRegression() из библиотеки sklearn.. Цикл состоит из 300 итераций. Считаем, что обследуемый, возможно, мертв, если вероятность соответствию классу 1 – мертв более 5 процентов.

Подбираем параметры для логистической регрессии 3.27.

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='none',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

Рисунок 3.27 – Параметры для логистической регрессии

Тесты за весь цикл:

- средняя чувствительность на тесте – 0.2731
- средняя специфичность на тесте – 0.9178
- средняя AUC на тесте –0.5955
- максимальная чувствительность на тесте – 1.0
- максимальная специфичность на тесте – 0.9822
- максимальная AUC на тесте –0.9438

Максимальная средняя выборка на тесте – 0.7 (в тесте 7 умерших). На тесте: чувствительность – 0.333 специфичность – 0.904, AUC – 0.618. Матрица пересечений показана в таблице 14.

Таблица 3.14 – confusion matrix Логистической регрессии на тестах

Предсказанные Фактические	0	1
0	472	49
1	5	2

Применяем модель таблицы 14 на полном dataset. Матрица показана в таблице 15.

Таблица 3.15 – confusion matrix полной модели

Предсказанные Фактические	0	1
0	1862	207
1	21	21

Строя confusion matrix для модели из таблицы 15, получаем следующие результаты:

- accuracy – 0.89
- specificity – 0.88
- sensetivity – 0.5
- AUC – 0.69

Random forest реализовывается с помощью функции RandomForestClassifier() из библиотеки sklearn. Dataset разделяем на тренировочную часть и тестовую в соотношении 75% на 25% функцией train_test_split() из библиотеки sklearn. На рисунке 3.28 подбираем параметры для Random forest.

```
rand_forest=RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                   max_depth=None, max_features=2, max_leaf_nodes=None,
                                   min_impurity_decrease=0.0, min_impurity_split=None,
                                   min_samples_leaf=1, min_samples_split=2,
                                   min_weight_fraction_leaf=0.0, n_estimators=40,
                                   n_jobs=None, oob_score=False, random_state=9613,
                                   verbose=0, warm_start=False).fit(X_train, y_train)
```

Рисунок 3.28 – Параметры для Random forest

Проводим обучение Random forest на dataset в цикле из 300 итераций. Каждый цикл изменяем случайное число для параметра random_state в RandomForestClassifier() и в train_test_split(), чтобы каждый раз тренировочная выборка изменялась, а так же строим confusion matrix и вычисляем точность модели, чувствительность, специфичность и AUC, занося все данные в массивы для того, чтобы выбрать самую успешную модель. Мертвыми считаем тех, у кого риск выше 5 процентов.

Тесты за весь цикл:

- средняя чувствительность на тесте – 0.1959
- средняя специфичность на тесте – 0.8953
- средняя AUC на тесте –0.5456
- максимальная чувствительность на тесте – 0.75
- максимальная специфичность на тесте – 0.9530
- максимальная AUC на тесте –0.8130

Максимальная средняя выборка на тесте – 0.8130 (в тесте 4 умерших) На тесте: чувствительность – 0.976, специфичность – 0.94, AUC – 0.958. Матрица показана в таблице 16.

Таблица 3.16 – confusion matrix Random Forest на тестах

Предсказанные Фактические	0	1
0	459	65
1	1	3

Применяем модель таблицы 16 на полном dataset. Матрица показана в таблице 17.

Таблица 3.17 – confusion matrix полной модели

Предсказанные Фактические	0	1
0	1945	124
1	1	41

Строя confusion matrix для модели из таблицы 17, получаем следующие результаты:

- accuracy – 0.94
- specificity – 0.92
- sensetivity – 0.97
- AUC – 0.92

Нейросетевая модель реализовывается в Keras, собираем слои (layers) для построения модели (model). Модель – это обычно граф слоев. Наиболее распространенным видом модели является стек слоев: модель tf.keras.Sequential().

Построение простой полносвязной сети показано на рисунке 3.29.

```
neuro = tf.keras.Sequential()
neuro.add(layers.Dense(20, input_dim = X.shape[1], activation='relu'))
neuro.add(layers.Dense(29,  activation='relu'))
neuro.add(layers.Dense(2, activation='softmax'))
opt = tf.keras.optimizers.Adam(learning_rate=0.001)
neuro.compile(loss = 'binary_crossentropy', optimizer = opt, metrics = ['binary_accuracy'])
```

Рисунок 3.29 – код реализации стека слоев с 4 слоями

Регулировка гиперпараметров нейронной сети реализуется при помощи kerstuner. Keras Tuner – это библиотека, которая помогает вам выбрать оптимальный набор гиперпараметров для программы в TensorFlow.

Проводим обучение нейронной сети на dataset в цикле из 200 итераций. Каждый цикл изменяем случайное число для параметра random_state в train_test_split(), чтобы каждый раз тренировочная выборка изменялась, а так же строим confusion matrix и вычисляем точность модели, чувствительность, специфичность и AUC.

Тесты за весь цикл:

- средняя чувствительность на тесте – 0.79
- средняя специфичность на тесте – 0.92
- средняя AUC на тесте – 0.85
- максимальная чувствительность на тесте – 0.1
- максимальная специфичность на тесте – 0.96
- максимальная AUC на тесте –0.96

Максимальная средняя выборка на тесте – 0.96 (в тесте 8 умерших) На тесте: чувствительность – 1.0, специфичность – 0.93, AUC – 0.96. Матрица показана в таблице 18.

Таблица 3.18 – confusion matrix Random Forest на тестах

Предсказанные Фактические	0	1
0	389	26
1	0	8

Применяем модель нейронной сети на полном dataset. Матрица показана в таблице 19.

Таблица 3.19 – confusion matrix нейронной сети полной модели

Предсказанные Фактические	0	1
0	1974	95
1	6	36

Строя confusion matrix для нейросетевой модели, получаем следующие результаты:

- accuracy – 0.95
- specificity – 0.95
- sensetivity – 0.85
- AUC – 0.9

4 Разработка информационного сервиса прогнозирования фатального события от сердечно-сосудистых заболеваний

4.1 Задачи информационного сервиса

В рамках проведения исследования моделей были составлены требования к информационному сервису.

Основная аудитория пользователей:

- медицинские работники Приморского края;
- жители Приморского края.

Не считая нейросетевой модели, необходимо получить результаты по шкале SCORE, т.к. она рекомендована для использования в РФ Европейским и Российским обществами кардиологов. Результаты по ASSIGN SCORE будут доступны по желанию пользователя.

Основные ФР необходимые для получения результатов по моделям SCORE и нейросетевой модели (обязательные):

- пол;
- возраст;
- факт курения;
- САД;
- общий холестерин.

ФР для получения результатов по модели ASSIGN SCORE (необязательные):

- HDL – холестерин;
- семейная история;
- наличие ревматоидного артрита;
- наличие сахарного диабета.

Пользователю должна быть предоставлена возможность вводить свои данные в специально отведенные поля, возможность очищения данных в полях с помощью кнопки «Очистить», отправка данных на сервер с помощью кнопки «Рассчитать».

Информационный сервис реализуется в формате web из – за ряда причин:

- специфика сервиса. Нет необходимости поддерживать контакт с пользователем путем push – уведомлений.
- редкая частота использования сервиса. Пользователь не будет хранить на устройстве приложение, котором нет необходимости пользоваться часто;
- меньшие затраты человека – часов на разработку.

- простота перехода к информационному сервису. От пользователя не требуется таких действий, как установка или скачивание;
- адаптивность. Сайт с адаптивным дизайном выглядит корректно на всех платформах, ограничения вводятся только браузерами, которые использует пользователь;
- легкость обновлений. Сайт легко обновлять и все изменения будут тут же происходить у пользователя.

Результаты расчета моделей должны располагаться в пользовательском интерфейсе.

4.2 •Реализация клиентской стороны пользовательского интерфейса

Реализация начинается с создания макета главной страницы сайта для дальнейшего её создания средствами front end.

Макет главной страницы делается при помощи веб – приложения Figma.

Figma – кросс – платформенный онлайн-сервис для дизайнеров интерфейсов и веб-разработчиков.

В Figma есть все необходимые компоненты для создания макетов. Основным преимуществом является то, что в конце создания макета можно получить свойства CSS каждого объекта.

Макет страницы реализуемого информационного сервиса указана на рисунке 4.1.

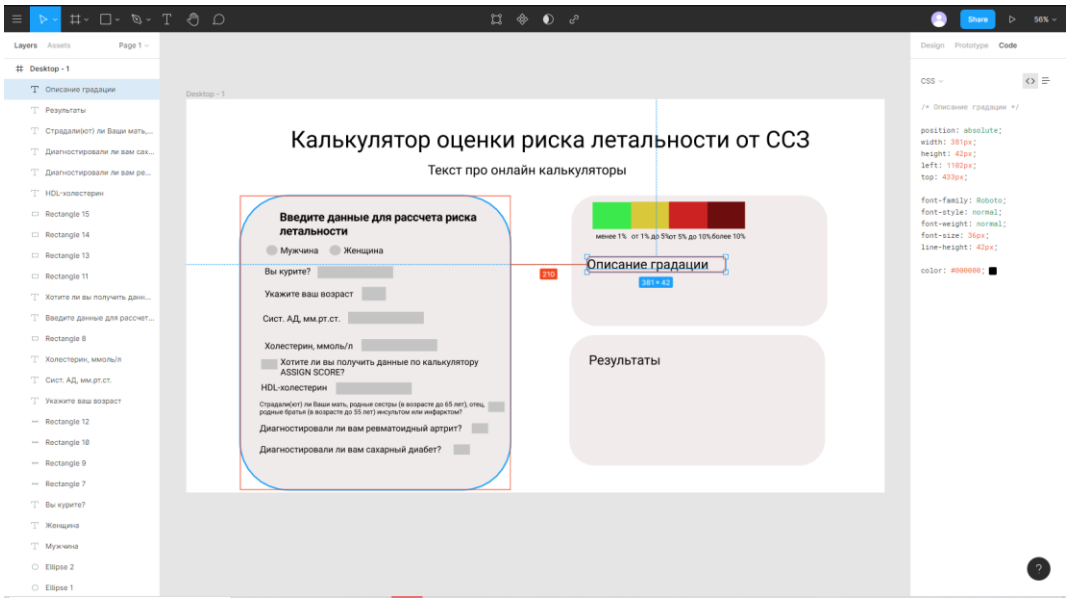

Рисунок 4.1 – макет страницы информационного сервиса

По макету создаётся структуры гипертекстового документа при помощи HTML5 при использовании CSS и клиентских сценариев на языке JavaScript, таким образом, чтобы элементы дизайна выглядели так же, как на макете.

HTML5 – язык разметки для структурирования и представления содержимого world web. [34]

CSS – формальный язык описания внешнего вида документа, написанного с использованием языка разметки. [35]

JavaScript – это легковесный, интерпретируемый или JIT-компилируемый, объектно-ориентированный язык программирования с функциями первого класса. Наиболее широкое применение находит, как язык сценариев веб-страниц. [36]

Для упрощения работы с формами далее используется framework Bootstrap.

Bootstrap – свободный и бесплатный HTML, CSS, JavaScript framework для быстрой проектировки и настройки адаптивных сайтов. Bootstrap включает переменные, Sass, адаптивную сеточную систему, обширные готовые компоненты и плагины JavaScript. [37]

Bootstrap включает:

- сетку;
- классы для стилизации контента;
- элементы для создания кнопок, форм, горизонтальных и вертикальных навигационных меню, слайдеров, выпадающих списков, модальных окон, всплывающих подсказок и других элементов интерфейса;
- классы для выравнивания текста, скрытия или отображения элементов, задания цвета и фона элементу, задание margin и padding отступов и т.д.

Для установки Bootstrap необходимо вставить stylesheet <link> в <head> перед всеми другими таблицами стилей, чтобы загрузить CSS.

Пример интеграции CSS стиля Bootstrap в HTML – код показан на рисунке 4.2.

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" crossorigin="anonymous">
```

Рисунок 4.2 – ссылка для добавления CSS стиля Bootstrap

Bootstrap придерживается парадигмы разработки mobile first, в которой сначала оптимизируем код под мобильные устройства, а затем масштабируем компоненты по мере необходимости, используя медиазапросы CSS. Чтобы обеспечить правильный рендеринг и масштабирование для всех устройств, необходимо добавить meta tag viewport в tag <head>.

Пример добавления viewport показан на рисунке 4.3.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Рисунок 4.3 – ссылка для добавления адаптивности под мобильные устройства через Bootstrap

Для реализации адаптивной сетки используется серия контейнеров/form – group, строк и столбцов для размещения и выравнивания содержимого. Построенная при помощи flexbox и полностью отзывчива.

Пример реализации адаптивной сетки в данной разработке показан на рисунке 4.4.

```
<div class="container">
    <div class="row">
        <div class="col-md">
          <div class="container">
              <div class="container-custom">
                  <form method="POST">
                      <h1 class="text-header"> Введите данные для рассчета риска летальности</h1>
                      <div class="form-group">
                          <p>Пол: </p>
```

Рисунок 4.4 – реализация адаптивной сетки в Bootstrap

Инструменты контроля текста – такие как <input>, <select> и <textarea> – стилизованы классом .form-control, который содержит основные стили внешнего вида, активного состояния, размерности и т.д.

Формы для ввода числовых значений с возможностью ограничения диапазона реализованы с помощью элементов input HTML5 типа number, которые получают обновленные стили от Bootstrap. Ограничение на ввод устанавливаются атрибутами max и min метки input. Число можно добавить непосредственно в форму или с помощью интерактивных стрелок.

Код реализации формы ввода возраста находится на рисунке 4.5.

```
<div class="form-group">
  <label for="inputAGE">Укажите ваш возраст:</label>
    <input type="number" class="form-control" id="inputage" name="age" placeholder="Возраст, лет" min=0 max=100>
</div>
```

Рисунок 4.5 – реализация формы ввода числовых значений средствами HTML5 и Bootstrap

Итоговая форма ввода показана на рисунке 4.6.

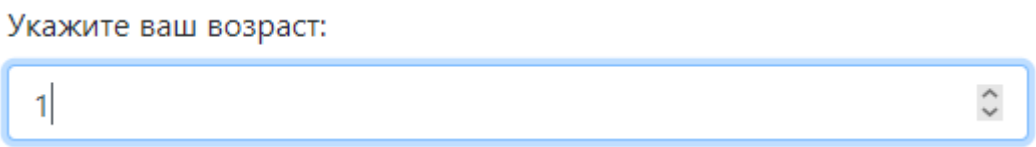

Рисунок 4.6 – форма для ввода числовых значений

Bootstrap Select – это элемент управления формы, который после щелчка отображает свернутый список из нескольких значений, которые можно использовать в формах.

Код реализации формы выбора факта курения из списка показан на рисунке 4.7.

```html
<div class="form-group">
  <label for="exampleFormControlSelect1"> Вы курите?</label>
  <select class="form-control" id="SmokeID" name="smoke">
    <option hidden> Выберите из списка</option>
    <option value="1"> Курю</option>
    <option value="2"> Бросил</option>
    <option value="3"> Не курю</option>
  </select>
</div>
```

Рисунок 4.7 – html код формы select

Форма для выбора атрибута из списка указана на рисунке 4.8.

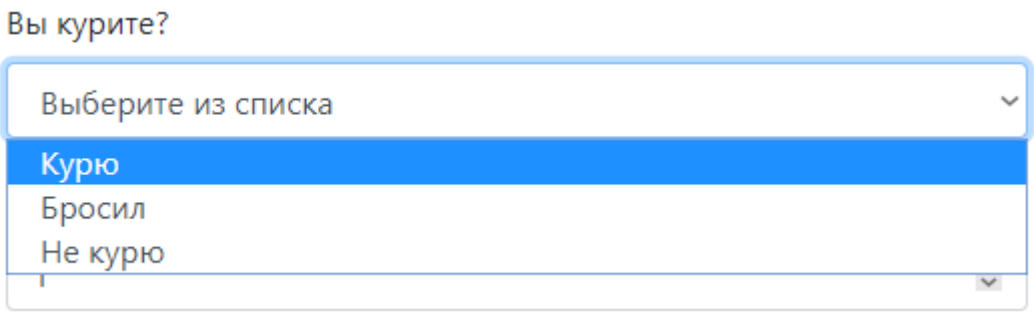

Рисунок 4.8 – форма выбора факта курения

Checkbox и radiobutton в Bootstrap улучшены с помощью класса form-check, единого класса для обоих типов ввода, который улучшает расположение и поведение их HTML-элементов. Checkboxes предназначены для выбора одного или нескольких параметров в списке, а radiobuttons – для выбора одного варианта из нескольких.

Код реализации checkbox и radiobutton находится на рисунке 4.9.

```html
<div class="form-group form-check">
  <input type="checkbox" class="form-check-input" id="checkAS" name="CHK" value=1>
  <label class="form-check-label" for="checkAS">Хотите ли вы получить данные по калькулятору ASSIGN SCORE?</label>
</div>

<div class="form-group">
  <p class="sbrosotstupov">Страдали(ют) ли Ваши мать, родные сестры (в возрасте до 65 лет), отец, родные братья (в возрасте до 55 лет)
  <div class="form-check form-check-inline">
    <input class="form-check-input" type="radio" name="FH" id="FamilyNo" value="1" checked disabled>
    <label class="form-check-label" for="FamilyNo">
      Нет
    </label>
  </div>
  <div class="form-check form-check-inline">
    <input class="form-check-input" type="radio" name="FH" id="FamilyYes" value="2" disabled>
    <label class="form-check-label" for="FamilyYes">
      Да
    </label>
```

Рисунок 4.9 – код реализация checkbox и radiobutton

Формы checkbox и radiobutton показана на рисунке 4.10.

☐ Хотите ли вы получить данные по калькулятору ASSIGN SCORE?

HDL-холестерин:

HDL-холестерин, ммоль/л

Страдали(ют) ли Ваши мать, родные сестры (в возрасте до 65 лет), отец, родные братья (в возрасте до 55 лет) инсультом или инфарктом?

◉ Нет ○ Да

Рисунок 4.11 – формы checkbox и radiobutton сайта

Класс .btn предназначен для использования с элементом <button>. Также можно использовать этот класс с элементами <a> или <input>. Для сайта необходимо создать 2 кнопки, кнопку «рассчитать» и кнопку «очистка форм». Кнопка «рассчитать» необходима для передачи данных серверу для дальнейшей манипуляции, а «очистка» для удобства очищения форм.

Код реализации кнопок показан на рисунке 4.12.

```
<div class="form-group col-md-24">
  <button type="Submit" name ="submit" value="Submit" class="btn btn-primary" onclick="handler_score()">Рассчитать</button>
  <button type="button" class="btn btn-default" onclick="handler_clear()">Очистить</button>
</div>
```

Рисунок 4.12 – код для создания функциональных кнопок

Итоговые формы для кнопок указана на рисунке 4.13.

Рассчитать Очистить

Рисунок 4.13 – формы для необходимых сайту кнопок

В <script> подключаем jQuery для удобной реализации функциональной зависимости между checkbox «Хотите ли получить данные по калькулятору ASSIGN SCOR?» и полей необходимых для рассчета. jQuery – набор функций JavaScript, фокусирующийся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими.

Скрипт по разблокировки и блокировки форм при клике на checkbox показан на рисунке 4.14.

```
$('#checkAS').on('change', function(){
  if($(this).is(':checked'))
  {
     $('#inputHDL').attr('disabled', false);
     $('#FamilyNo').attr('disabled', false);
     $('#FamilyYes').attr('disabled', false);
     $('#RANo').attr('disabled', false);
     $('#RAYes').attr('disabled', false);
     $('#SDNo').attr('disabled', false);
     $('#SDYes').attr('disabled', false);
   }
  else
  {
    $('#inputHDL').val("");
    $('#FamilyNo').prop("checked", true);
    $('#RANo').prop("checked", true);
    $('#SDNo').prop("checked", true);
    $('#checkRA').prop('checked',false);
    $('#checkSD').prop('checked',false);
    $('#FamilyNo').attr('disabled', true);
    $('#FamilyYes').attr('disabled', true);
    $('#RANo').attr('disabled', true);
    $('#RAYes').attr('disabled', true);
    $('#SDNo').attr('disabled', true);
    $('#SDYes').attr('disabled', true);
  }
});
```

Рисунок 4.14 – скрипт для разблокировки форм

В итоге по макету была реализована html страница со стилями Bootstrap – а. Итоговая html страница показана на рисунке 4.15.

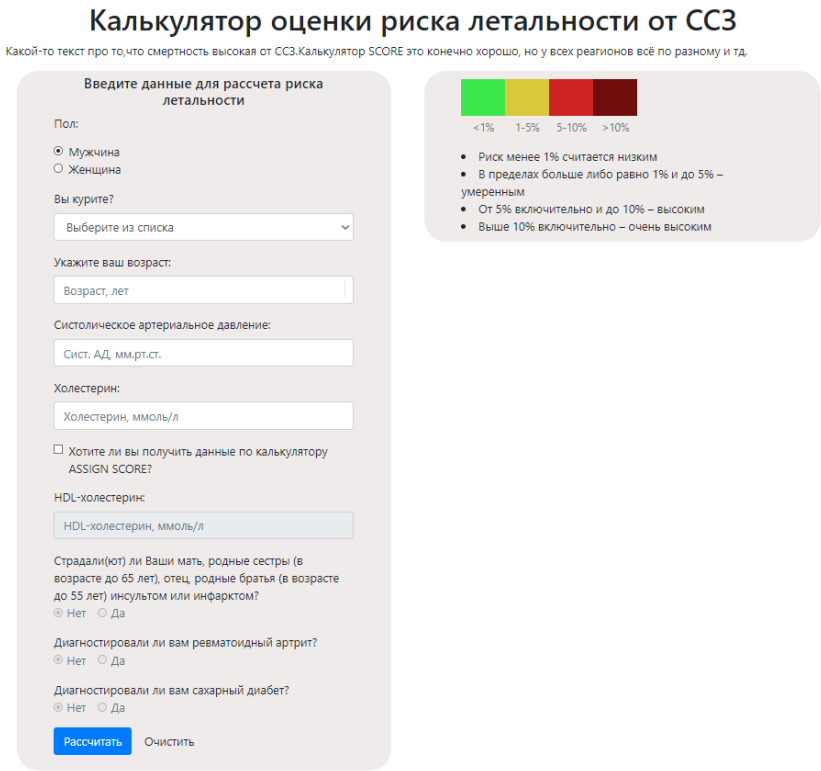

Рисунок 4.16 – готовый пользовательский интерфейс, созданный с помощью Bootstrap

Все дальнейшие манипуляции с пользовательскими данными будут проводится на серверной части на framework Python Django.

4.3 •Реализация программно – аппаратной части сервиса

Программно – аппаратная часть сервиса создается на Django.

Html документы по правилам Django хранятся в папке tamplates. Для того, чтобы Django понимал в какой директории шаблоны, необходимо прописать путь относительно файла settings.py.

Путь указывается командой PROJECT_DIR=os.path.dirname(__file__).

Код для определения местонахождения шаблонов показан на рисунке 4.17.

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [
        os.path.join(PROJECT_DIR,'templates')

        ],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

Рисунок 4.17 – определения местонахождения шаблонов в файле setting.py

Файлы css и javascript прописываются в папке static. Для того, чтобы Django понимал в какой директории они находятся, необходимо прописать путь относительно файла settings.py.

PROJECT_ROOT== os.path.abspath(os.path.dirname(__file__)).

Код для определения местонахождения файлов css и javascript показан на рисунке 4.18.

```
STATIC_URL = '/static/'
STATICFILES_DIRS=[
    os.path.join(PROJECT_ROOT,'static')
]
```

Рисунок 4.19 – определения местонахождения css и javascipt файлов в файле setting.py

Подключение файлов css и javascript к шаблону производится с помощью {% load static %} в HTML – коде.

Для получения значений форм используется метод POST. В файле приложения forms.py прописывается UserForm с определением типов полей.

Пример создание формы показан на рисунке 4.20.

```
class UserForm(forms.Form):
    sex = forms.IntegerField()
    smoke = forms.IntegerField()
    age = forms.IntegerField()
    CAD = forms.IntegerField()
    hol = forms.IntegerField()
    CHK=forms.IntegerField()
    HDL=forms.IntegerField()
    FH=forms.IntegerField()
    RA=forms.IntegerField()
    SD=forms.IntegerField()
```

Рисунок 4.20 – создание формы для сбора данных с полей ввода в файле forms.py

Далее в файле view.py подключается данная форма командой from .forms import UserForm. Далее к кнопке «рассчитать» класса submit привязывается получение значений формы POST методом get().

Обозначаются значения переменных по умолчанию для каждого элемента POST необходимого для работы калькуляторов.

В форму подаем значения POST запроса с кнопки «рассчитать».

Код передачи данных из полей ввода в форму показан на рисунке 4.21.

```
submitbutton= request.POST.get("submit")
sex = 1
smoke = 0
age = 0
CAD = 0
hol = 0
CHK=0
HDL=0
FH=0
RA=0
SD=0
SCORE=0
AS_SC=0
pred=array([])
form= UserForm(request.POST or None)
```

Рисунок 4.21 – передача результатов POST запроса в форму

Для разделения результатов запроса на переменные берем результаты поля ввода по его названию и передаем в соответствующую переменную методом form.get(). Для очистки результатов поля ввода используется метод cleaned_data.

Код присвоения значений полей ввода в соответствующие переменные показан на рисунке 4.22.

```
if form.is_valid():
    sex = form.cleaned_data.get("sex")
    smoke = form.cleaned_data.get("smoke")
    age = form.cleaned_data.get("age")
    CAD = form.cleaned_data.get("CAD")
    hol = form.cleaned_data.get("hol")
```

Рисунок 4.22 – присвоения значений полей ввода в соответствующие переменные

Далее через переменную context передаем значение форм обратно в шаблон для дальнейшей демонстрации результатов на главной странице сайта.

В HTML – коде для демонстрации получения результатов временно используем вывод в заголовки при условии, что кнопка «рассчитать» нажата.

Код вывода результатов манипуляции с формами показан на рисунке 4.23.

```
{% if submitbutton == "Submit" %}
<br>
  <div class=row>
    <div class='container-custom-frame'>
        <div class='row'>
          <div class="d-block">
             <p class="sbrosotstupov">Риск по нейросетевой модели  - </p>
             <p class="sbrosotstupov"> {{Pred}}</p>
          </div>
          <p class='sbrosotstupov'> Риск по модели SCORE - {{SCORE}}</p>
          <p class='sbrosotstupov'> Риск по модели ASSIGN SCORE - {{AS_SC}}</p>
        </div>
    </div>
  </div>
  {% endif %}
```

Рисунок 4.23 – вывод результатов в шаблон при нажатии кнопки

Результат вывода данных указан на рисунке 4.24.

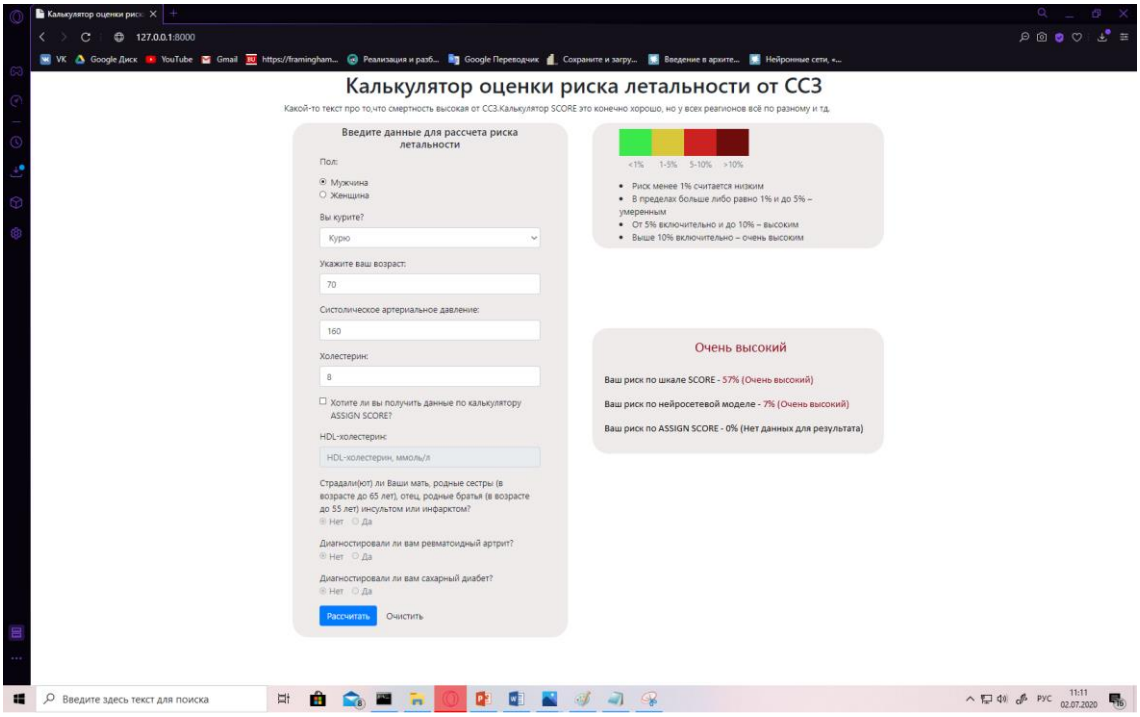

Рисунок 4.24 – форма вывода данных полей ввода

Данные возвращаются корректные. Применяем данные к калькулятору SCORE, получаем правильные значения. В результате получаем рабочую программно – аппаратную сторону сервиса, которая справляется с поставленными задачами.

5. Технико-экономическое обоснование эффективности вложений в разработку проекта

Для проведения расчета себестоимости разработки проекта учитывались трудовые ресурсы и прямые затраты на приобретение материальных ресурсов.

5.1 Расчет стоимости трудозатрат по проекту

Автор выпускной квалификационной работы выполнял роль аналитика, технического писателя и веб – разработчика в проекте. В таблице 5.1 указан расчет стоимости трудовых ресурсов проекта.

Таблица 5.1 – Расчет стоимости трудовых ресурсов

Наименование специалиста	Оклад, руб/мес	Оклад с учетом НДФЛ, руб	ЕСН, руб	Итого ФОТ с учетом ЕСН, руб	Месячный фонд рабочего времени, час	Стоимость 1 час работы, руб.
Аналитик данных	20000	22988,5	7356,3	30344,8	168	180,6
Технический писатель	15000	17241,3	5517,2	22758,5	168	135,4
Веб – разработчик	25000	28735,6	9195,4	37931	168	225,7

На рисунке 5.1 демонстрируется фрагмент диаграмма Ганта из Приложения В разработки проекта.

	Формирование документации, целей и задачь проекта	**21 дней**	**Пт 08.11.19**	**Пт 06.12.19**		
	Создание паспорта проекта	2 дней	Пт 08.11.19	Пн 11.11.19		Технический писатель
	Описание предметной облости	2 дней	Вт 12.11.19	Ср 13.11.19	2	Технический писатель
	Построение бизнес-модели Остервальдера	2 дней	Чт 14.11.19	Пт 15.11.19	3	Технический писатель

Рисунок 5.1 – План график работ

Расчет стоимости трудовых ресурсов для каждого этапа работ по разработке и внедрению проекта отображен в таблице 5.1. В расчете стоимости использованы данные реальных заработных плат по farpost.ru и по hh.ru.

В таблице 5.2 приведен расчет себестоимости по этапам работ по проекту.

Таблица 5.2 – Расчет себестоимости проектирования по этапам работ

Тип работ	Затраты
Формирование документации, целей и задачь проекта	22758
Исследование моделей оценки рисков летальности	124252
Разработка информационного веб-сервиса	106530
Всего	253540

В таблице 5.3 указаны расчеты стоимости работ в резерве трудовых ресурсов.

Таблица 5.3 – Расчет стоимости работ в разрезе трудовых ресурсов

Должность	Ставка, руб/ч	Фактические трудозатраты, ч	Фактические трудозатраты, руб
Аналитик данных	180,6	688	124252
Технический писатель	135,4	168	22758
Веб – разработчик	225,7	472	106530
Всего			253540

Из таблицы видим, что основные затраты идут на аналитика данных, т.к. самая объемная часть работы проведение исследования. Общие затраты на трудовые ресурсы 253540 рублей.

5.2 Смета прямых затрат на изготовление продукта

Стоимость ресурсов для приобретения технических средств и программного обеспечения указаны в таблице 5.4.

Таблица 5.4 – Расчет стоимости затрат на приобретение технических средств

Наименование	Цена, руб	Количество ,шт	Итог
Персональный компьютер (видеокарта RTX 2070 aero 8 GB, intel i7-8gen, 16 gb ORM)	75000	1	75000
Всего		1	75000

Для реализации проекта необходимо приобрести персональный компьютер из таблицы 5.4 с достаточно мощными компонентами для разработки моделей машинного обучения, а в частности нейросетевой модели. Видеокарта должна обладать высокими характеристиками для того, чтобы перенаправить часть вычислений на ядра видеокарты.

5.3 Технико-экономическое обоснование. Финансовая модель проекта

В финансовом расчете учтены все виды затрат, связанных с реализацией проекта: инвестиционные, постоянные, переменные, фонд заработной платы и социальные отчисления. Общая сумма инвестиционных затрат составляет 386 тыс. руб., недостаток в которых компенсируется за счет заемных средств, полученных в банке на срок 12 месяцев под 13% годовых. Наибольший объем денежных средств приходится на прямые затраты на разработку проекта и косвенные расходы при запуске проекта. Погашение кредита осуществляется ежемесячными аннуитетными платежами.

Часть финансовой модели из Приложения Г проекта приведена на рисунке 5.2.

Сервис прогнозирования летальных событий ССЗ															
Номера периодов			0	1	2	3	4	5	6	7	8	9	10	11	12
Период	Ед. измерения	параметров на единицу/ персональных параметров модели	окт.20	ноя.20	дек.20	янв.21	фев.21	мар.21	апр.21	май.21	июн.21	июл.21	авг.21	сен.21	окт.21
Отчет о фин. результатах															
ВЫРУЧКА			**10 800**	**16 200**	**21 600**	**29 700**	**40 500**	**43 200**	**43 200**	**45 900**	**45 900**	**51 300**	**62 100**	**72 900**	**78 300**
Количество продаж	шт.		4	6	8	11	15	16	16	17	17	19	23	27	29
Стоимость единицы продукта (лицензия, услуга)	руб.	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700
ПЕРЕМЕННЫЕ РАСХОДЫ			**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Себестоимость оказания услуги, непосредственно связаная с продажей услуги за единицу	руб.	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Себстоимость оказываемых услуг/продаж			0	0	0	0	0	0	0	0	0	0	0	0	0
Маржинальный доход			**10 800**	**16 200**	**21 600**	**29 700**	**40 500**	**43 200**	**43 200**	**45 900**	**45 900**	**51 300**	**62 100**	**72 900**	**78 300**
Маржинальность бизнеса			100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Общепроизводственные расходы			**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**
Аренда хостинг		5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000
Обслуживание сайта и наполнение контента		5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000
ВАЛОВАЯ ПРИБЫЛЬ			**800**	**6 200**	**11 600**	**19 700**	**30 500**	**33 200**	**33 200**	**35 900**	**35 900**	**41 300**	**52 100**	**62 900**	**68 300**
Косвенные расходы			**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**
Реклама		5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000
РКО		4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200
EBITDA			**-8 400**	**-3 000**	**2 400**	**10 500**	**21 300**	**24 000**	**24 000**	**26 700**	**26 700**	**32 100**	**42 900**	**53 700**	**59 100**
Амортизация			0	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689
Проценты по кредитам и займам			0	-3 797	-3 597	-3 270	-2 658	-2 616	-2 215	-1 962	-1 582	-1 308	-981	-633	-327
Прибыль до налогооблажения			**-8 400**	**-20 486**	**-14 886**	**-6 459**	**4 953**	**7 695**	**8 096**	**11 049**	**11 429**	**17 103**	**28 230**	**39 378**	**45 084**
Налоги (УСН "Доходы" - 3% для отрасли ИТ в первые 2 года с регистрации)			-324	-486	-648	-891	-1 215	-1 296	-1 296	-1 377	-1 377	-1 539	-1 863	-2 187	-2 349
ЧИСТАЯ ПРИБЫЛЬ			-8 724	-20 972	-15 534	-7 350	3 738	6 399	6 800	9 672	10 052	15 564	26 367	37 191	42 735
ЧИСТАЯ ПРИБЫЛЬ нарастающим итогом			-8 724	-29 696	-45 230	-52 580	-48 843	-42 444	-35 644	-25 972	-15 920	-357	26 010	63 201	105 936

Рисунок 5.2 – финансовая модель проекта

Чистая прибыль проекта после выхода на плановые показатели продаж составляет 105936 рубля за срок реализации проекта.

5.4. Выводы об окупаемости проекта

Предложенный проект является коммерчески эффективным. Чистая прибыль за год составила 106 тыс. рублей и поскольку объем первоначальных инвестиций на разработку продукта составил 330 тыс. рублей рентабельность возврата инвестиций 32% годовых, что гораздо выше ставок размещения инвестиций в другие более консервативные финансовые инструменты.

Однако, учитывая то, что данный проект является для региона относительно новым, возможны основные риски, связанные с продвижением и достижением плановых объемов продаж в установленные сроки.

Риски и мероприятия по их предотвращению приведены в таблице 5.5.

Таблица 5.5 – основные риски проекта и методы их предотвращения

Риски	Вероятность наступления	Тяжесть последствий	Мероприятия по предотвращению
Отклонение реального спроса от планируемого	Среднее	Тяжело	Планирование мероприятий по продвижению, мотивация партнеров к участию в мероприятиях по продвижению
Выход на рынок нового конкурента	Низко	Низко	Особое внимание укреплению на рынке в течение начального периода

Анализируя представленные данные, можно говорить о том, что реализация данного проекта связана с определенным уровнем риска по причине новизны услуги и слабого охвата. Однако, существующие перспективы развития достаточно высоки, поскольку, медицинский персонал Приморского края заинтересован в улучшение качества прогнозирования ССЗ жителей края. Таким образом, можно сделать вывод о высокой инвестиционной привлекательности данного проекта

Заключение

В результате работы были выполнены все поставленные задачи.

Проведен анализ готовых моделей для оценки риска летальности SCORE и ASSIGN SCORE. Были получены основные статистические данные, была рассчитана точность моделей на данных обследуемой когорты Приморского края.

На основе факторов риска летальности SCORE было построены такие модели, как логистическая регрессия, random forest и многослойный персептрон. Модели – логистическая регрессия и Random Forest показали в среднем результаты хуже, чем результаты устоявшихся моделей оценки риска летальности от ССЗ. Многослойный персептрон в среднем на тестовых данных показал лучшие результаты среди готовых моделей и также среди результатов эксплуатируемых моделей.

Разработан информационный веб-сервис для оценки индивидуального десятилетнего риска летальности от ССЗ. Сервис предоставляет пользователю все возможности расчета десятилетнего риска летальности от ССЗ по моделям SCORE, нейросетевой модели и модели ASSIGN SCORE. Модель ASSIGN SCORE может быть выбрана по желанию, т.к. она не рекомендована к использованию на территории РФ из – за обучающей выборки модели основанной исключительно на жителях Шотландии.

Проанализировав данные собранные во время разработки, можно говорить о том, что реализация данного проекта связана с определенным уровнем риска по причине новизны услуги и слабого охвата. Однако, существующие перспективы развития достаточно высоки, поскольку, медицинский персонал Приморского края заинтересован в улучшение качества прогнозирования ССЗ жителей края.

Веб-сервис полностью справляется с поставленной задачей оценки рисков летальности от ССЗ.

Список используемых источников

1. Cardiovascular diseases // World Health Organization [Electronic resources] – URL: https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

2. Global Atlas on cardiovascular disease prevention and control / editors: Mendis S, Puska P, Norrving B – World Health Organization, Geneva, 2016 –155 p.

3. Primary prevention and risk factor reduction in coronary heart disease mortality among working aged men and women in eastern Finland over 40 years: population based observational study / editors: J ousilahti P, Laatikainen T, Peltonen M, Borodulin K, Mannisto S, Jula A, Salomaa V, Harald K, Puska P, Vartiainen E. – BMJ, 2016 – 352 p.

4. Эпидемиология сердечно-сосудистых заболеваний в различных регионах России (ЭССЕ-РФ). Обоснование и дизайн исследования. / Научно-организационный комитет проекта ЭССЕ-РФ. // Профилактическая медицина. – 2013. – № 6. – С. 25-34.

5. Рабочие места, как неотъемлемая часть здорового образа жизни населения // Леонидова Г.В. – Проблемы развития территории, Федеральное государственное бюджетное учреждение науки Институт социально-экономического развития территорий Российской академии наук, 2015 – 125 с.

6. Забота о здоровье сотрудников – резерв роста производительности труда // Бурмистрова Н.О., Бурмистров Д.А. – Международный научно-исследовательский журнал, Индивидуальный предприниматель Соколова Марина Владимировна, 2016 – 75с.

7. Гриванв И.Ю., Гриванова О.В., Гриванова С.М. Безопасность жизнедеятельности. Учебно-практическое пособие. – Владивосток: Изд-во ВГУЭС, 2010. – 92 с.

8. Programming Languages Most Used and Recommended by Data Scientists // Business Broadway [Electronic resources] – URL: https://businessoverbroadway.com/2019/01/13/ programming-languages-most-used-and-recommended-by-data-scientists.

9. McKinney Wes. Python for Data Analysis / Wes McKinney – O’Reilly, 2015. - 482 p.

10. Pandas // Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia .org/wiki/Pandas.

11. SciPy // Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/SciPy.

12. Визуализация данных с использованием Seaborn // Машинное обучение, нейронные сети, искусственный интеллект [Электронный ресурс] – Режим доступа: https://www.machinelearningmastery.ru/data-visualization-using-seaborn-fc24db95a850/.

13. Основы статистики с Python: описательная статистика // Tproger [Электронный ресурс] – Режим доступа: https://tproger.ru/translations/basic-statistics-in-python-descriptive-statistics/.

14. Модуль Math // Python Scripts [Электронный ресурс] – Режим доступа: https://python-scripts.com/math.

15. Обзор методов классификации в машинном обучении с помощью Scikit-Learn // Tproger [Электронный ресурс] – Режим доступа: https://tproger.ru/translations/scikit-learn-in-python/.

16. Why TensorFlow // TensorFlow [Electronic resources] – URL: https://www .tensorflow.org.

17. Ф.М. Гафаров, А.Ф. Галимянов //Искусственные нейронные сети и их приложения Учебное пособие – .Казань: Издательство Казанского университета, 2018. – 120 с

18. Django // Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Django.

19. Новый проект // DjangoBook [Электронный ресурс] – Режим доступа: https://djbook.ru/ch02s05.html.

20. Задачи по подготовке данных для расширенного машинного обучения // Microsoft [Электронный ресурс] – Режим доступа: https://docs.microsoft.com/ru-ru/azure/machine-learning/team-data-science-process/prepare-data.

21. Мюллер, Гвидо. Введение в машинное обучение с помощью Python / Мюллер, Гвидо - Москва 2017 – 393 с.

22. Обучение с учителем и без учителя // Studme [Электронный ресурс] – Режим доступа: https://studme.org/235648/informatika/obuchenie_uchitelem_uchitelya.

23. Обучение с учителем // Machine Learning [Электронный ресурс] – Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=Обучение_с_учителем.

24. Машинное обучение: методы и способы // Гид по технологиям цифровой трансформации [Электронный ресурс] – Режим доступа: https://www.osp.ru/ cio/2018/05/13054535.

25. Как легко понять логистическую регрессию// Habr [Электронный ресурс] – Режим доступа: https://habr.com/ru/company/io/blog/265007/.

26. Random forest // Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Random_forest.

27. Нейронная сеть // Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Нейронная_сеть.

28. European Guidelines on cardiovascular disease prevention in clinical practice // Massimo F. Piepoli, Arno W. Hoes, Stefan Agewall. – European Heart Journal, 2016, № 37, p. 2315–2381.

29. Кардиоваскулярная профилактика 2017 // М.Г. Бубнова, О.М. Драпкина, Н.Е. Гаврилова, Р.А. Еганян, А.М.Калинина, Н.С. Карамнова, Ж.Д. Кобалава, А.В. Концевая, В.В. Кухарчук, М.М.Лукьянов, Г.Я. Масленникова, С.Ю. Марцевич, В.А. Метельская, А.Н. Мешков, Р.Г.Оганов, М.В. Попович, О.Ю. Соколова, О.Ю. Сухарева, О.Н. Ткачева, С.А. Шальнова, М.В. Шестакова, Ю.М. Юферева, И.С. Явелов– Москва 2017 – 288 с

30. Калькулятор SCORE // Медицинская профилактика [Электронный ресурс] – Режим доступа: http://mpmo.ru/dop/risk.php.

31. ASSIGN SCORE // assign score [Electronic resources] – URL: http://www.assign-score.com/estimate-the –risk.

32. Критерий Стьюдента // Machine Learning [Электронный ресурс] – Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=Критерий_Стьюдента.

33. U-критерий Манна – Уитни // Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/U-критерий_Манна_—_Уитни.

34. HTML5 // Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/HTML5

35. Основы CSS // MDN web docs [Электронный ресурс] – Режим доступа: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics#:~:text=CSS%20(Cascading%20Style%20Sheets)%20—,для%20стилизации%20вашей%20веб-страницы.

36. JavaScript // MDN web docs [Электронный ресурс] – Режим доступа: https://developer.mozilla.org/ru/docs/Web/JavaScript

37. Build fast, responsive sites with Bootstrap // Bootstrap [Electronic resources] – URL: https://getbootstrap.com.

Приложение А

Поддержание здорового образа жизни персонала в ФГБОУ ВО "Владивостокский государственный университет экономики и сервиса"

На фоне усиления человеческого фактора в социально-экономическом развитии необходимо более акцентированное внимание к состоянию здоровья населения, в особенности его работающей части. Рабочие места как одна из важных сред, воздействующих на человека, рассматриваются ВОЗ приоритетным направлением здоровьесбережения [5].

Следствия инволюционных изменений не только ограничивают физическую активность. Существенно ухудшается настроение, снижается самооценка, замотивированность к труду и личностному развитию, повышается раздражительность и даже уровень агрессии, как следствие, в производственных коллективах повышается уровень конфликтности. Основополагающим фактором предупреждения и устранения физической детренированности в условиях монотонной работы являются занятия физической культурой. [6]

В таблице А.1 приведены примеры мероприятий в ФГБОУ ВО "Владивостокский государственный университет экономики и сервиса"

Таблица А.1 – Мероприятия по поддержанию здоровья персонала

Мероприятие	Описание
Ежегодная организация фестиваля фитнеса «Старт Чемпионов»	Ежегодный фестиваль спорта и фитнеса от ВГУЭС для популяризации ЗОЖ. Разделение на уровни подготовки гостей для новичков и для профессионалов.
Борьба с курением: регулярное проведение акции «Курить не модно»	Ежегодная акция, где главная цель призыв отказаться от зависимостей. Меняют сигареты на маленькие подарки.
Конкурс плакатов «Здоровое поколение»	Конкурс проводится в рамках масштабной работы, проводимой в университете по пропаганде здорового образа жизни. И среди его основных задач – формирование у студентов знаний о здоровом образе жизни и позитивного к нему отношения.

Поддержание здорового образа жизни сотрудников позволяет снизить риск приобретения хронических заболеваний, повышает удовлетворенность от работы, повысить производительность труда, снизить компенсационные выплаты, связанные с болезнью работника и издержек на поиск и подготовку новых кадров, в связи с уходом предыдущих (болезнь, смерть). Таким образом, забота о здоровье сотрудников и стимулирование здорового образа жизни персонала способствует не только замотивированности к труду, но и является дополнительным резервом для повышения его производительности [7].

Приложение Б

Безопасность жизнедеятельности в Лаборатории цифрового моделирования и анализа данных физики и биомедицины

Основным объектом в производственных условиях является рабочее место, представляющее собой в общем случае пространство, в котором может находиться человек при выполнении производственного процесса. Рабочее место является основной подсистемой производственного процесса. Согласно СанПиН 2.2.2/2.4.1340-03 (п.3.4) площадь рабочего места пользователей ПЭВМ должна быть S больше, либо равно 6 квадратным метрам.

Рекомендуемый проход слева, справа и спереди от стола 500 мм. Слева от стола допускается проход 300 мм. Рекомендуемое расстояние от спинки стула до границы должно быть не менее 300 мм. [8]

Предусматривается рекомендуемая и допустимая компоновка рабочего места. Рекомендуемая компоновка рабочего места по отношению к оконным проемам – свет падает с левой стороны, допустимая – свет падает с правой стороны. В случае чрезвычайных ситуаций человек, выходя с рабочего места должен двигаться только вперед к двери, не допускается делать никаких поворотов более, чем на 90 градусов.

Требования к эвакуационному пути:

- ширина принимается 0,6 м при одностороннем выходе на эвакуационный путь. При двустороннем выходе ширина увеличивается в два раза и составляет 1,2 м;
- в рабочих помещениях дверь должна открываться наружу, и при двустороннем движении по эвакуационному пути дверь – двухстворчатая.

Рабочие места не рекомендуется располагать вплотную к несущим конструкциям, поэтому при расстановке рабочих мест необходимо учитывать дополнительно следующие требования:

- расстояние от стенки (окна) до границы площади рабочего места – не менее 0,3 м;
- расстояние от передней стенки помещения до границы первого рабочего места – не менее 0,8 м;
- расстояние между боковыми поверхностями видеомониторов – не менее 1,2 м;
- при размещении рабочих мест с ПЭВМ расстояние от задней стенки видеомониторами до спины впереди сидящего должно быть не менее 2,0 м.

Площадь помещения – 24 квадратных метра. Рассчитав площадь помещения, определяется объем помещения (Б.1) и объем воздуха, приходящийся на человека (Б.2).

$$V_{пом} = S \times h. \quad (Б.1)$$

$$V_{\text{чел}} = \frac{V_{\text{пом}}}{n} \quad (Б.2)$$

При расчете объема необходимо учитывать:

- высота производственных помещений – 3.4 м.
- минимальный объем воздуха на 1 человека при работе с компьютерами в офисах, административных помещениях составляет 20 м3.
- оптимальный объем воздуха на 1 человека при отсутствии вредностей в воздухе рабочей зоны – 40 м3.

Объем помещения равен 81,6 кубометров. Объем воздуха на одного человека 27,2 кубометров.

На рисунке 1.1 представлена планировка рабочего помещения и рабочего места, на котором были пройдены практики.

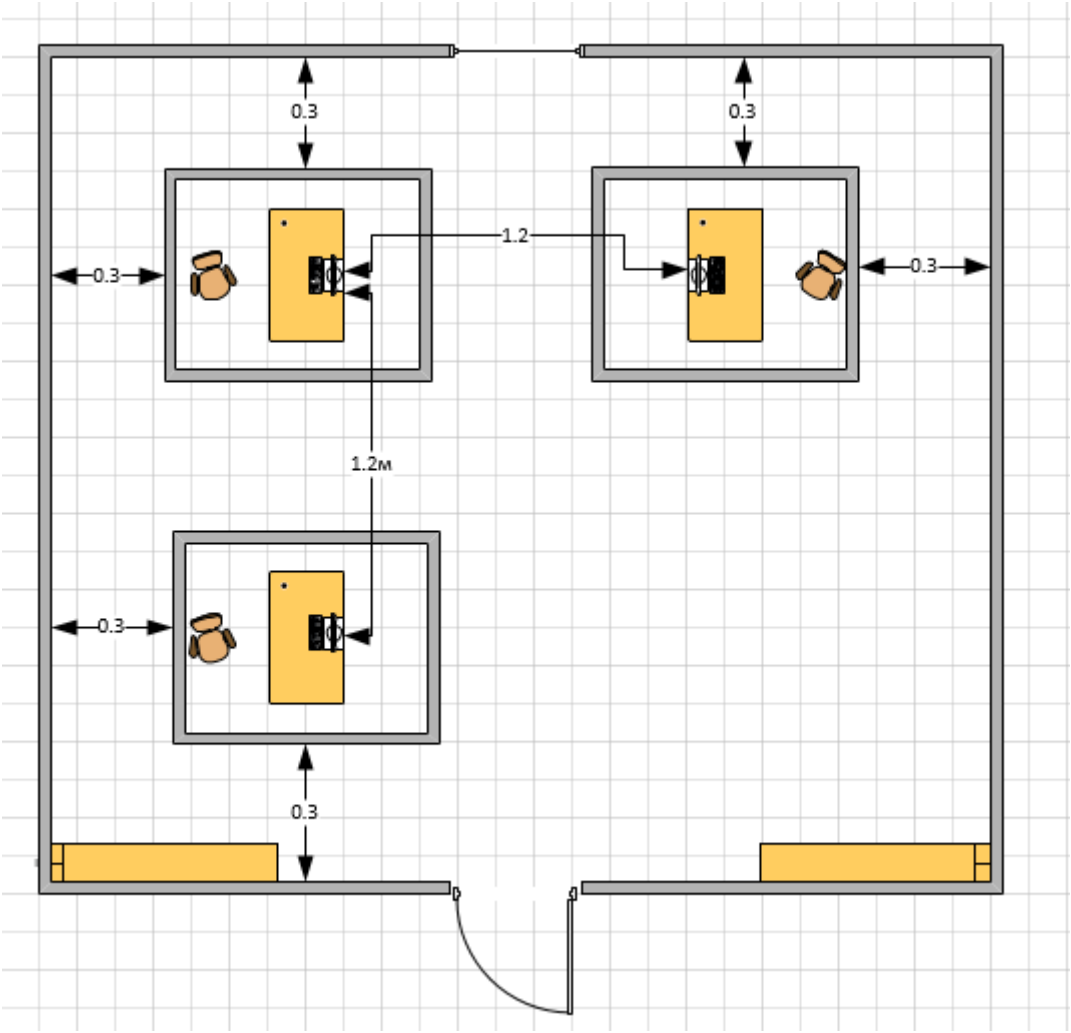

Рисунок Б.1 – Планировка рабочего помещения

Таким образом, рабочее помещение и рабочее место полностью соответствуют всем нормам.

Приложение В

Диаграмма Ганта проекта показана на рисунке Б.1

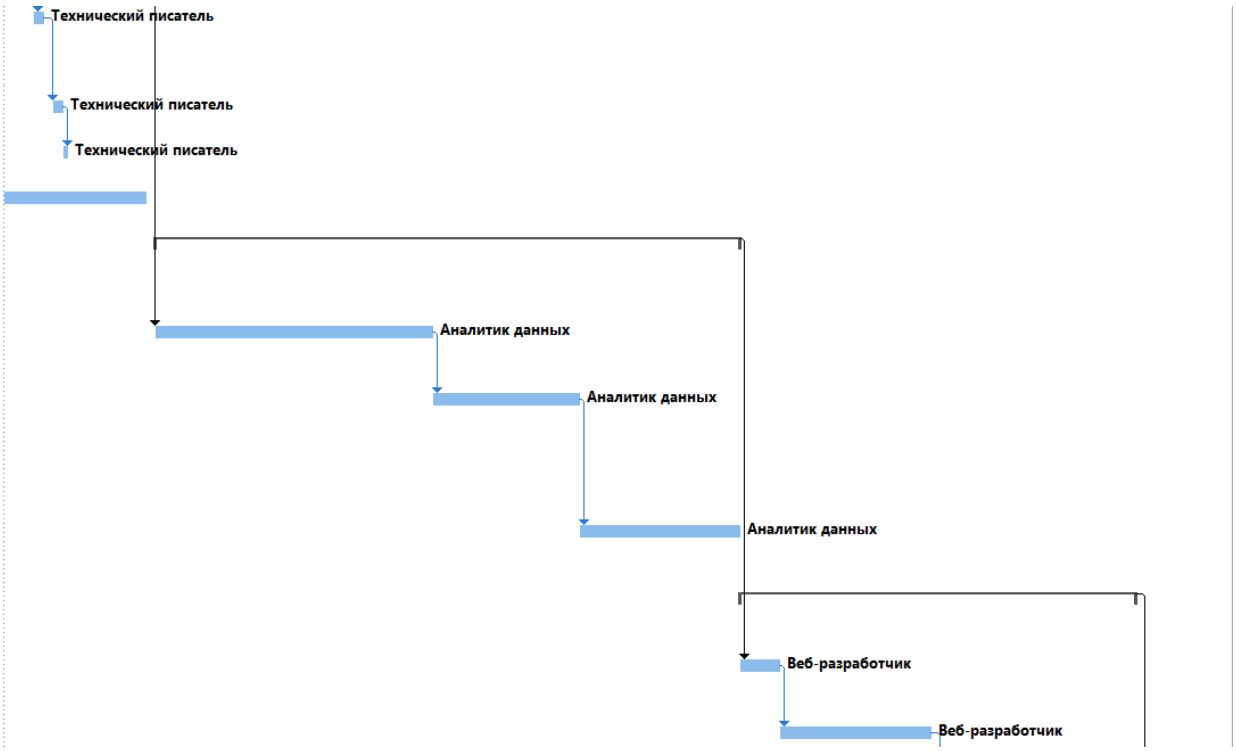

Рисунок Б.1 – диграмма Ганта проекта

Приложение Г

Финансовая модель проекта показана на таблице В.1

Таблица В.1 – финансовая модель проекта

Номера периодов			0	1	2	3	4	5	6	7	8	9	10	11	12
Период	Ед. измерения	параметров на единицу/ персональных параметров модели	окт.20	ноя.20	дек.20	янв.21	фев.21	мар.21	апр.21	май.21	июн.21	июл.21	авг.21	сен.21	окт.21
Отчет о фин. результатах															
ВЫРУЧКА			**10 800**	**16 200**	**21 600**	**29 700**	**40 500**	**43 200**	**43 200**	**45 900**	**45 900**	**51 300**	**62 100**	**72 900**	**78 300**
Количество продаж	шт.		4	6	8	11	15	16	16	17	17	19	23	27	29
Стоимость единицы продукта (лицензия, услуга)	руб.	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700	2 700
ПЕРЕМЕННЫЕ РАСХОДЫ			**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Себестоимость оказания услуги, непосредственно связаная с продажей услуги за единицу	руб.	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Себстоимость оказываемых услуг/продаж			0	0	0	0	0	0	0	0	0	0	0	0	0
Маржинальный доход			**10 800**	**16 200**	**21 600**	**29 700**	**40 500**	**43 200**	**43 200**	**45 900**	**45 900**	**51 300**	**62 100**	**72 900**	**78 300**
Маржинальность бизнеса			100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Общепроизводственные расходы			**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**	**-10 000**
Аренда хостинг		5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000
Обслуживание сайта и наполнение контента		5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000
ВАЛОВАЯ ПРИБЫЛЬ			**800**	**6 200**	**11 600**	**19 700**	**30 500**	**33 200**	**33 200**	**35 900**	**35 900**	**41 300**	**52 100**	**62 900**	**68 300**
Косвенные расходы			**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**	**-9 200**
Реклама		5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000	-5 000
РКО		4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200	-4 200
EBITDA			**-8 400**	**-3 000**	**2 400**	**10 500**	**21 300**	**24 000**	**24 000**	**26 700**	**26 700**	**32 100**	**42 900**	**53 700**	**59 100**
Амортизация			0	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689
Проценты по кредитам и займам			0	-3 797	-3 597	-3 270	-2 658	-2 616	-2 215	-1 962	-1 582	-1 308	-981	-633	469 067
Прибыль до налогооблажения			**-8 400**	**-20 486**	**-14 886**	**-6 459**	**4 953**	**7 695**	**8 096**	**11 049**	**11 429**	**17 103**	**28 230**	**39 378**	**514 478**
Налоги (УСН "Доходы" - 3% для отрасли ИТ в первые 2 года с регистрации)			-324	-486	-648	-891	-1 215	-1 296	-1 296	-1 377	-1 377	-1 539	-1 863	-2 187	-2 349
ЧИСТАЯ ПРИБЫЛЬ			-8 724	-20 972	-15 534	-7 350	3 738	6 399	6 800	9 672	10 052	15 564	26 367	37 191	512 129
ЧИСТАЯ ПРИБЫЛЬ нарастающим итогом			-8 724	-29 696	-45 230	-52 580	-48 843	-42 444	-35 644	-25 972	-15 920	-357	26 010	63 201	575 330
Движение денежных средст по инвест. деятельности			-328 540	0	0	0	0	0	0	0	0	0	0	0	0
Инвестиционная программа			**-328 540**												
Трудозатраты на разработку продукта			-253 540												
Оборудование сотрудников			-75 000												
ПО			0												
Амортизация, мес.		24													
Амортизация				-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689	-13 689
НАЛОГ УСН "Доходы"															
Ставка налога		3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%	3%
База налога			10 800	16 200	21 600	29 700	40 500	43 200	43 200	45 900	45 900	51 300	62 100	72 900	78 300
Налог			-324	-486	-648	-891	-1 215	-1 296	-1 296	-1 377	-1 377	-1 539	-1 863	-2 187	-2 349
Отчет о движении денежных средств (ДДС)															
Остаток денежных средств на начало периода			0	2 736	-32 047	-61 392	-82 553	-92 626	-100 038	-107 049	-111 188	-114 947	-113 194	-100 638	-77 258
Денежный поток по опер. деятельности			-8 724	-7 283	-1 845	6 339	17 427	20 088	20 489	23 361	23 741	29 253	40 056	50 880	525 818
Денежный поток по инвест. деятельности			-328 540	0	0	0	0	0	0	0	0	0	0	0	0
Денежный поток по фин. деятельности			340 000	-27 500	-27 500	-27 500	-27 500	-27 500	-27 500	-27 500	-27 500	-27 500	-27 500	-27 500	-27 500
Сальдо денежных средств по всем видам деятельности			2 736	-34 783	-29 345	-21 161	-10 073	-7 412	-7 011	-4 139	-3 759	1 753	12 556	23 380	498 318
Остаток денежных средств на конец периода			2 736	-32 047	-61 392	-82 553	-92 626	-100 038	-107 049	-111 188	-114 947	-113 194	-100 638	-77 258	421 060
Флаг кассового разрыва		11	0	1	1	1	1	1	1	1	1	1	1	1	0
Финансовая деятельность															
Вклад в акционерный капитал			10 000												
Выплата дивидендов															
Банковский кредит															
Остаток задолженности на начало периода			0	330 000	302 500	275 000	247 500	220 000	192 500	165 000	137 500	110 000	82 500	55 000	27 500
Поступление кредита			330 000												
Возврат основного долга				-27 500	-27 500	-27 500	-27 500	-27 500	-27 500	-27 500	-27 500	-27 500	-27 500	-27 500	-27 500
Остаток задолженности на конец периода			330 000	302 500	275 000	247 500	220 000	192 500	165 000	137 500	110 000	82 500	55 000	27 500	0
Выплата ссудного процента		14%	0	-3 797	-3 597	-3 270	-2 658	-2 616	-2 215	-1 962	-1 582	-1 308	-981	-633	469 067
CF проекта			-337 264	-3 486	1 752	9 609	20 085	22 704	22 704	25 323	25 323	30 561	41 037	51 513	56 751
CF total			-337 264	-340 750	-338 998	-329 389	-309 304	-286 600	-263 896	-238 573	-213 250	-182 689	-141 652	-90 139	-33 388
DCF			-337 264	-3 398	1 664	8 898	18 128	19 974	19 469	21 165	20 630	24 267	31 761	38 860	41 729
DCF total			-337 264	-340 662	-338 997	-330 099	-311 971	-291 997	-272 529	-251 364	-230 734	-206 467	-174 706	-135 846	-94 117
Ставка дисконтирования, год		36,0%	1,360												
Ставка дисконтирования, мес.		2,6%	1,026												
NPV (чистая приведенная стоимость)		-94 117													
IRR		380%													