САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ЭНЕРГЕТИЧЕСКИХ СИСТЕМ

Каменская Елизавета Александровна

Выпускная квалификационная работа магистра

Диагностирование диабета на начальном этапе

Направление 010400

Прикладная математика и информатика

Магистерская программа «Математическое и информационное обеспечение экономической деятельности»

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Крылатов А. Ю.

Санкт-Петербург

2019

Содержание

Введение

Некоторые болезни очень сложно диагностировать на этапе их зарождения в человеческом организме. Именно они чаще всего пропускаются докторами. Это связано с тем, что на ранних этапах отсутствуют какие-либо симптомы, они неясные, либо слабо ощутимые. Другая причина – даже при полном спектре анализов даже квалифицированному специалисту будет сложно определить наличие такой болезни.

В данной работе речь пойдёт о диабете. Сахарный диабет — это состояние, при котором количество глюкозы (сахара) в крови слишком высокое, потому что организм не может правильно его использовать. Это происходит потому, что организм не в состоянии использовать или не производит гормон инсулин, который отвечает за переработку сахара из пищи, для клеток вашего тела. Различают два вида диабета: диабет 1 типа – врождённый диабет, диабет 2 типа – приобретенный. К сложно диагностируемому относится 2 тип. Дело в том, что диабет 2 не имеет ярко выраженных симптомов, требуются годы наблюдений для того, чтобы диагностировать данное заболевание. Часто сами больные не обращают внимания на симптомы этой болезни и не обращаются за медицинской помощью к профессионалам.

Сахарный диабет 2 типа — заболевание, затрагивающее практически все органы и системы в организме. Повышенный уровень сахара в крови отрицательно влияет на нервную систему, головной мозг, сердечно-сосудистую систему, а также вызывает определенные изменения уровня холестерина крови.

В настоящее время человечество живёт в веке высоких технологий, которые имеют огромные мощности и способны обрабатывать большие объемы информации. Уже сейчас существуют методы современной медицины, с использованием цифровых технологий, которые помогают докторам проводить анализы высокой точности и диагностировать те или иные болезни. Поэтому, можно поставить задачу диагностирования на раннем этапе такого сложно диагностируемого заболевания, как диабет 2 типа, используя современные методы машинного обучения [1-3].

Обзор литературы

За последние три десятилетия число людей с сахарным диабетом в мире более чем удвоилось, что делает его одной из наиболее важных проблем общественного здравоохранения для всех стран [12]. Сахарный диабет 2 типа (СД2) и предиабет все чаще наблюдаются у женщин, детей, подростков и молодых людей. Профилактика СД2 является задачей на всю жизнь и требует комплексного подхода [13-14].

В 2010 году, во всем мире, у 285 миллионов человек был сахарный диабет, у 90% из которых был диабет 2 типа (СД2) (Диаграмма 1). Прогнозируется, что число людей с сахарным диабетом во всем мире возрастет до 439 миллионов к 2030 году, что составляет 7,7% от общей численности взрослого населения мира в возрасте 20–79 лет [15] (Диаграмма 2).

Диабет оказывает существенное влияние на экономическую сферу здравоохранения государства. При оценке экономических последствий учитываются несколько факторов: заболеваемость и распространение заболевания, уровень развития системы здравоохранения и общий уровень экономического развития населения. Для оценки таких последствий были разработаны два подхода:

1. Первый подход измеряет нематериальные затраты, связанные с диабетом. Он сочетает в себе число лет здоровой жизни, потерянных в результате ранней смертности, и лет, потерянных из-за инвалидности.
2. Второй подход — это метод оценки стоимости болезни, который включает в себя концепции прямых, косвенных и нематериальных затрат.

Исследование, проведенное Всемирным банком, показало, что из 1362 миллионов лет жизней с поправкой на инвалидность, потерянных во всех болезнях в 1990 году, 7,97 миллионов лет были потеряны из-за диабета. В исследовании, проведенном в 1992 году, в котором оценивались прямые затраты на лечение диабета в США, Американская ассоциация диабета использовала подход, основанный на оценке стоимости заболевания, и обнаружила, что общая сумма расходов за 1 год составила 45,2 миллиарда долларов [10].

В эпидемиологических исследованиях Зиммета и Всемирной организации здравоохранения, проведенных в 1994 году, приводятся оценки увеличения распространенности диабета в результате увеличения численности населения. Оценки глобальной стоимости диабета на основе этих исследований показывают, что диабет составляет 2-3% от общего бюджета здравоохранения в каждой стране [11] (Диаграмма 3).

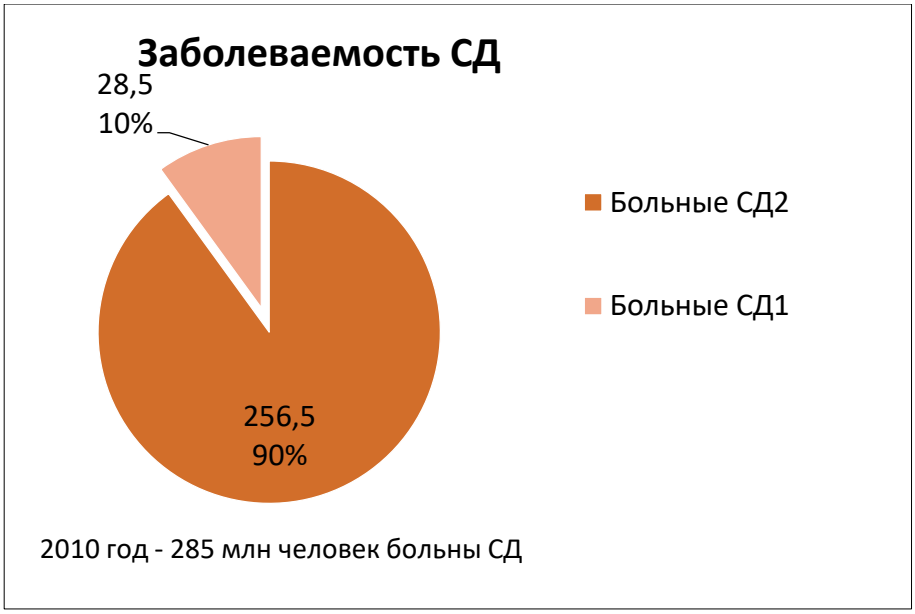


Диаграмма 1

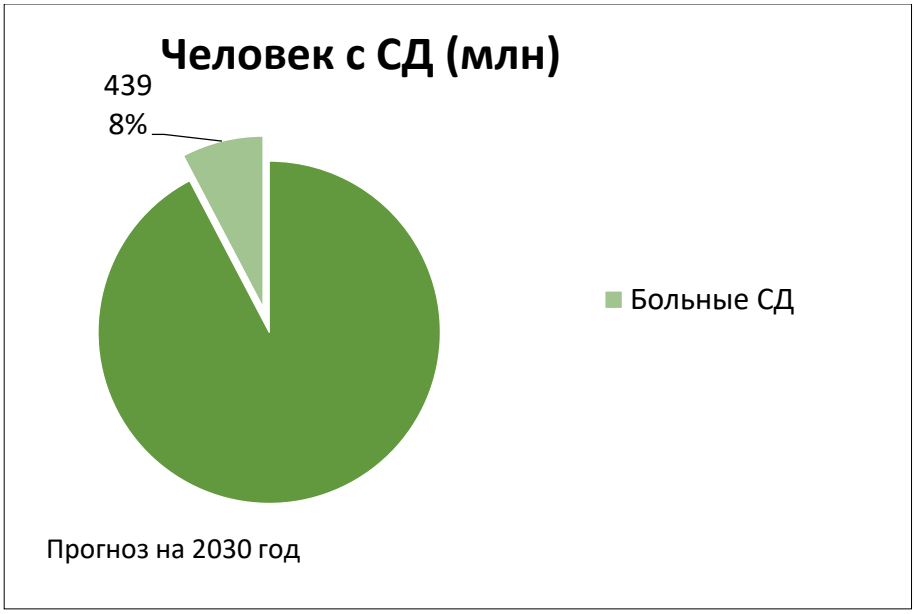


Диаграмма 2

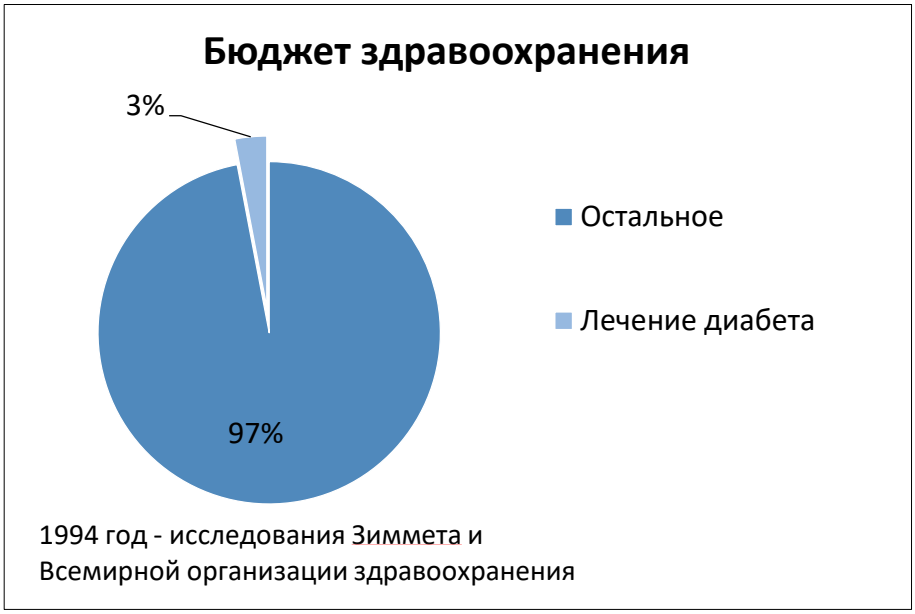


Диаграмма 3

Таким образом, осложнения СД2, увеличение заболеваемости и распространенности диабета приводят к значительным экономическим последствиям. Поскольку диагноз был поставлен достаточно поздно, затраты, которые могли бы идти на экономическое развитие, идут на лечение людей больных СД2. Итак, раннее диагностирование СД2 является важным этапом лечения и профилактики осложнений. Так как диагностирование диабета на раннем этапе является дорогостоящим, то в данной работе предлагается использовать машинное обучение для удешевления данного исследования.

Постановка задачи

В данной работе были поставлены следующие задачи:

1. изучение предметной области и анализ данных необходимых для диагностики диабета;
2. поиск и анализ необходимых данных;
3. приведение данных к особому виду для методов машинного обучения;
4. программное построение и сравнительный анализ методов машинного обучения для прогнозирования диабета у пациента;
5. анализ демографической ситуации, экономики и здравоохранения, связанный с заболеванием СД2.

Глава 1. Данные

1.1 Анализ данных

Для диагностирования диабета у женщин на начальном этапе были проанализированы данные исследований и измерений, влияющие на наличие данного заболевания.

Глюкоза

Главным анализом для диагностирования диабета (любого типа) является глюкозотолерантный тест – этот тест позволяет исследовать реакцию организма на введение дополнительного количества глюкозы. Данный анализ проводится в два этапа:

1. Сдается венозная кровь натощак. Результат – количество глюкозы, которое находится в венозной плазме. (Плазма – кровь без эритроцитов, тромбоцитов и лейкоцитов.)
2. После пациенту необходимо, в течении 5 минут, выпить 70 г растворённой глюкозы в 300 мл воды. Далее обследуемый вновь сдаёт кровь через 1 час и через 2 часа.

   Данные результаты измеряются в миллимоль/литр или в мг/дл: 1 ммоль/л = 18 мг/дл.

   Спустя 2 часа употребления глюкозы диагноз может быть следующим:

   - Норма – менее 7,8 ммоль/л (140,4 мг/дл)
   - Нарушенная толерантность – от 7,8 до 10,9 ммоль/л (от 140,4 до 196,2 мг/дл)
   - Диабет – более 10,9 ммоль/л (196,2 мг/дл)

На самом деле, глюкозотолерантный тест всегда требует перепроверки! Назвать его очень точным – нельзя. Именно поэтому необходимо оценить и собрать другие данные, влияющие на наличие диабета.

Инсулин

Глюкозотолерантный тест так же проводится для определения количества инсулина в крови. Через два часа после приёма глюкозы нормальным показателем инсулина будет: 17,8 – 173 мкМЕ/мл (микро международные единицы в 1

миллилитре). Высокий показатель инсулина может указывать на то, что человек предрасположен к диабету и рано или поздно столкнётся с данным заболеванием. Нулевой показатель инсулина в крови человека указывает на развитие сахарного диабета 2 типа. Данные отклонения обычно связаны с избыточной массой тела и могут осложняться ожирением, из-за чего толерантность к глюкозе будет серьезно нарушена.

Индекс массы тела (ИМТ)

Недостаточная масса тела, как и избыточная, способна стать причиной проблем со здоровьем. Сахарный диабет 2 типа чаще всего развивается на фоне избыточного веса и ожирения, однако нередко встречается недостаточная масса тела у людей, больных диабетом 1 типа. ИМТ позволяет рассчитать нормальный вес для любого человека. Формула для вычисления ИМТ:

$$\text{ИМТ} = \frac{\text{вес в кг}}{(\text{рост в м})^2}$$

- Недостаточная масса тела: ИМТ ≤ 18,5
- Нормальный вес: 18,5 < ИМТ ≤ 24,9
- Избыточный вес: 24,9 < ИМТ ≤ 29,9
- Ожирение: ИМТ ≥ 30

Толщина кожно-жировой складки над трицепсом (КЖСТ)

В продолжение расчетов ИМТ для диагностирования диабета важно добавить измерение толщины кожно-жировой складки над трицепсом, которое указывает на избыточную массу тела. Толщина КЖСТ определяет не только нарушения в весе, но также является одним из главных показателей неправильного питания и малой физической активности. Как известно, большинство людей, страдающих сахарным диабетом, придерживаются неправильного рациона и ведут пассивный образ жизни. Толщина КЖСТ измеряется в мм. Нормой для женщин считается: 13 мм ≤ КЖСТ ≤ 14,5 мм.

Диастолическое кровяное давления

Высокое давление при сахарном диабете – распространённая проблема, с которой сталкиваются больные. По статистике гипертонию выявляют у 60%

диабетиков. Около 25% людей с сахарным диабетом 1 типа и 80% людей с диабетом 2 типа имеют высокое артериальное давление. Патология сильно ухудшает самочувствие, усугубляет течение основного заболевания. На фоне повышенного АД повышается риск развития тяжёлых осложнений (инсульт, инфаркт), исход которых смертелен. Для больных диабетом 1, 2-го типа нормальным считается давление, не превышающее показателей 130/85 мм рт. ст. Особое внимание, уделяется диастолическому кровяному давлению, так как именно этот показатель служит связывающим между гипертонией и диабетом.

Количество беременностей

Немаловажным фактором диабета может служить количество беременностей у женщины. Самопроизвольное прерывание беременности происходит у 15—31% женщин в 20—27 недель беременности или раньше. Преждевременные роды часты, женщины, больные диабетом, редко донашивают до срока родов. У 20— 60% беременных может быть многоводие. При многоводии часто диагностируют пороки развития плода и мертворождаемость (у 29%). Несовместимые с жизнью пороки развития встречаются в 2,6% случаев. Внутриутробная гибель плода происходит обычно в 36—38 недель беременности. Чаще это случается при крупном плоде, проявлениях диабета и гестозе.

Родословная

Диабет может передаваться генетически. Признак диабета может проявиться у детей, даже если родители им не обладают. Если один из родителей является носителем данного признака, то данный признак может не проявиться, либо проявиться у половины потомства. Также диабет можете передаваться через одно или два поколения. Функция диабетической родословной (DPF) была разработана Смиттом и др. [4], чтобы обеспечить синтез истории сахарного диабета у родственников и генетическую связь этих родственников с субъектом исследования. DPF использует информацию от родителей, дедушек и бабушек, братьев и сестер, тетей и дядей, и двоюродных братьев и сестёр. Она обеспечивает меру ожидаемого генетического влияния родственников на возможный риск

диабета пациента. Чем выше данный показатель, тем больше вероятность того, что человек болен диабетом либо предрасположен к нему.

Возраст

Один из главных факторов риска заболеть сахарным диабетом является возраст. Чем старше человек, тем больше у него оснований опасаться данного заболевания. Сахарный диабет 1 типа возникает, как правило, в молодом возрасте (этой формой диабета в основном страдают молодые люди в возрасте до 30 лет). Сахарный диабет 2 типа - это болезнь зрелого возраста (им в основном страдают пожилые люди).

1.2 Входные данные

Для практической реализации алгоритма диагностирования диабета необходим поиск и предварительная обработка данных для формирования тренировочного набора.

Набор данных состоит из 768 наблюдений по 9 основным параметрам:

1. Уровень глюкозы в плазме (мг/дл)
2. Уровень инсулина в плазме (мкМЕ/мл)
3. Индекс масс тела (кг/м^2)
4. Толщина кожно-жировой складки над трицепсом (мм)
5. Диастолическое кровяное давление (мм рт)
6. Количество беременностей
7. Функция диабетической родословной
8. Возраст
9. Выходной параметр результата

Для наглядности были построены гистограммы по всем ранее перечисленным исследованиям для больных СД2 и здоровых (Рис. 1 – Рис. 8).

1 – больные диабетом, 0 – здоровые.

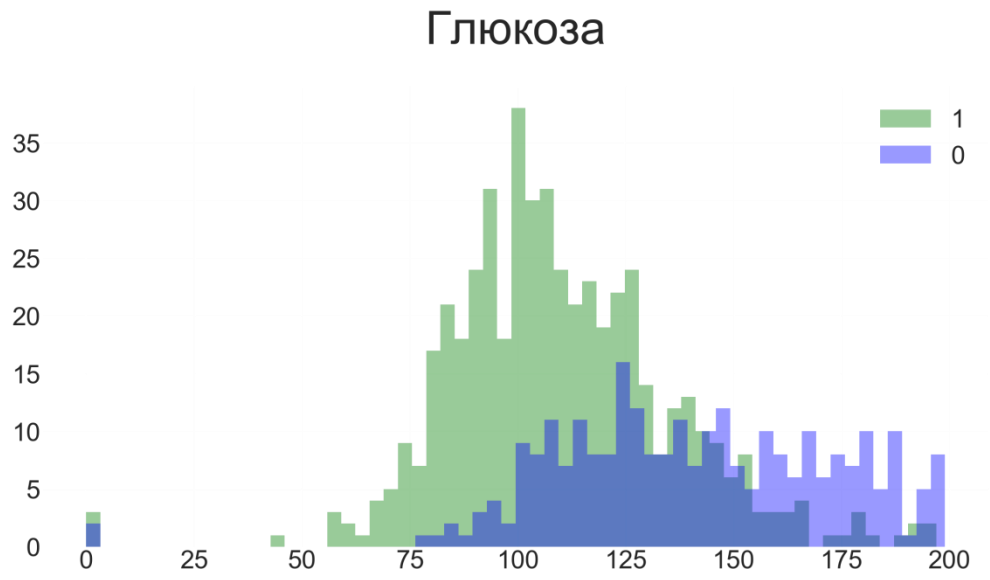


Рис. 1 Уровень глюкозы в плазме

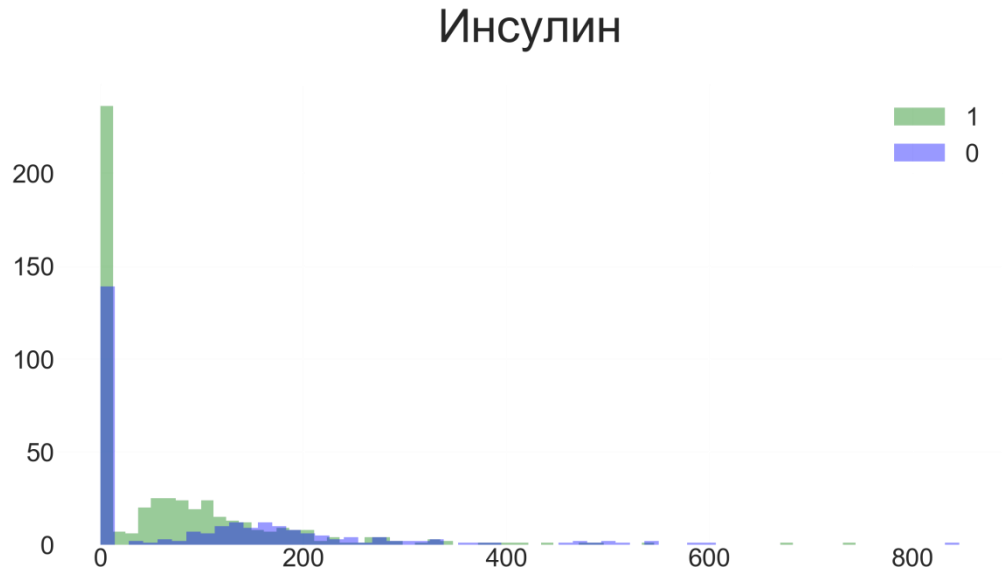


Рис. 2 Уровень инсулина в плазме

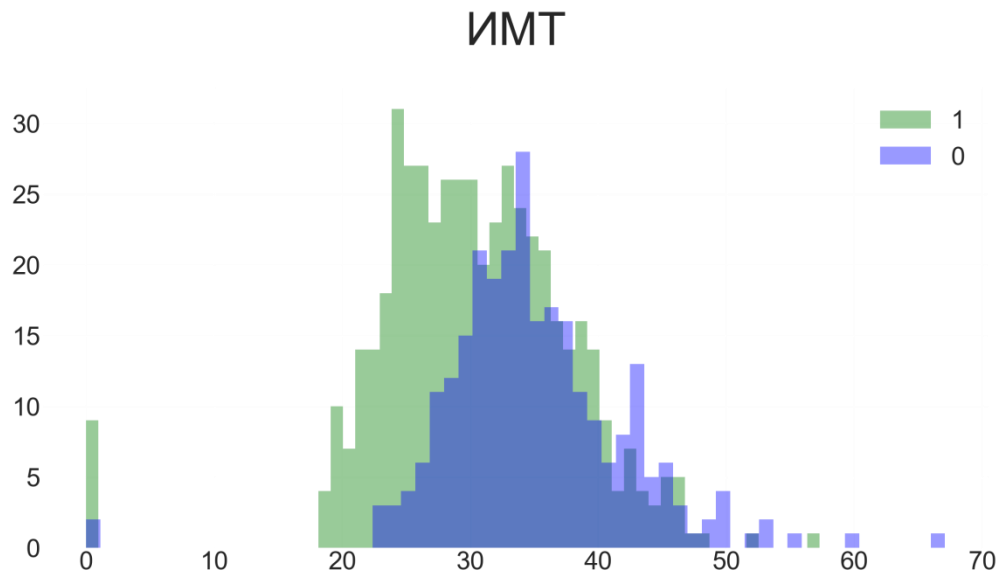


Рис. 3 Индекс массы тела

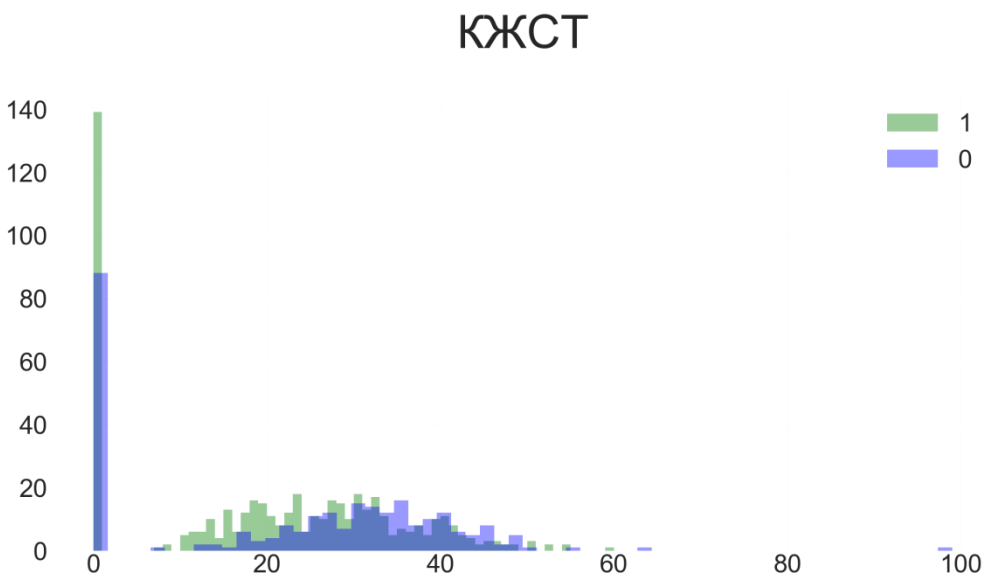


Рис. 4 Толщина кожно-жировой складки над трицепсом

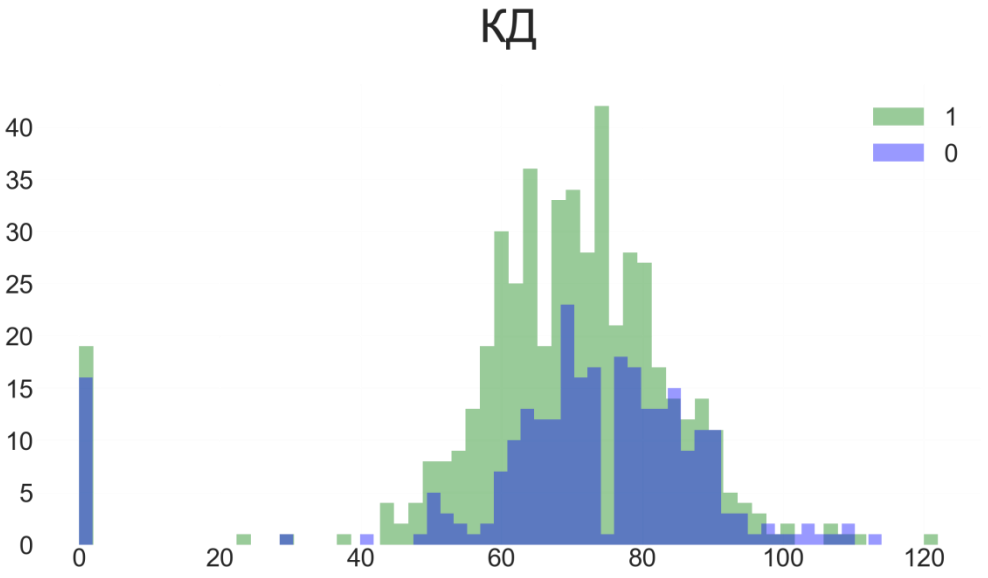


Рис. 5 Диастолическое кровяное давление

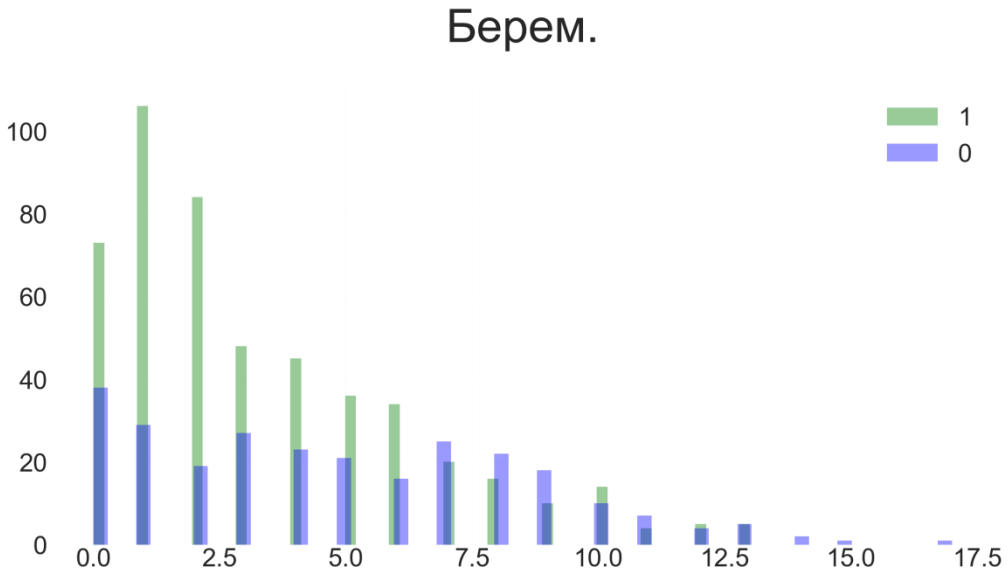


Рис. 6 Беременности

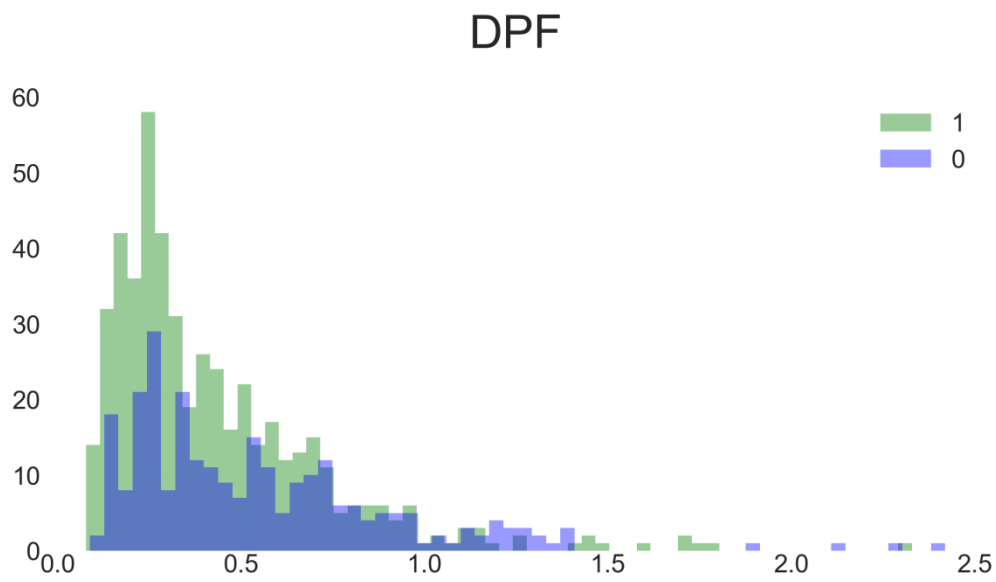


Рис. 7 Функции диабетической родословной

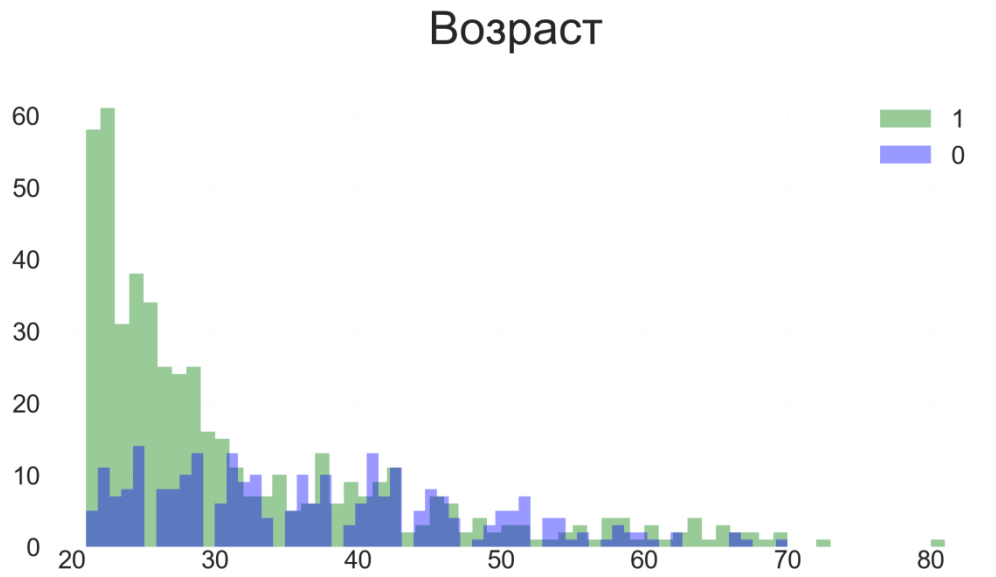


Рис. 8 Возраста

По данным графикам можно заметить, что рассмотренные признаки, безусловно, являются показателем диабета, но не могут каждый в отдельности гарантировать 100% результат.

Суммарная статистика для каждого признака (Таблица 1):

	Берем.	Глюкоза	КД	КЖСТ	Инсулин	ИМТ	DPF	Возраст	Результат
count	768,000000	768,000000	768,000000	768,000000	768,000000	768,000000	768,000000	768,000000	768,000000
mean	3,845052	120,894531	69,105469	20,536458	79,799479	31,992578	0,471876	33,240885	0,348958
std	3,369578	31,972618	19,355807	15,952218	115,244002	7,884160	0,331329	11,760232	0,476951
min	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,078000	21,000000	0,000000
25%	1,000000	99,000000	62,000000	0,000000	0,000000	27,300000	0,243750	24,000000	0,000000
50%	3,000000	117,000000	72,000000	23,000000	30,500000	32,000000	0,372500	29,000000	0,000000
75%	6,000000	140,250000	80,000000	32,000000	127,250000	36,600000	0,626250	41,000000	1,000000
max	17,000000	199,000000	122,000000	99,000000	846,000000	67,100000	2,420000	81,000000	1,000000

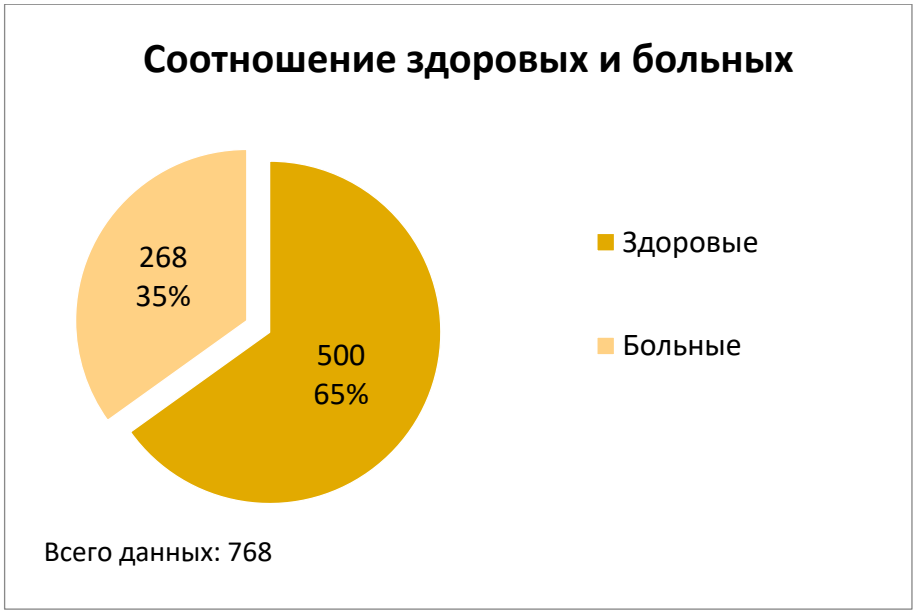
Таблица 1

Целевой признак:

0 – здоровые: 500;

1 – больные диабетом: 268.

Соотношение количества примеров, принадлежащих к каждому классу (Диаграмма 4):



Диарграмма 4

По данным признакам классы относительно сбалансированы (т.е. нет скошенных классов).

Для последующего анализа необходимо привести все входящие данные к единому виду. Используем класс StandartScaler, который преобразует данные следующим образом: среднее значение = 0, стандартное отклонение = 1. При распределении данных из каждого значения в наборе данных будет вычтено

среднее значение выборки (изначальное), а затем разделено на стандартное отклонение (изначальное) всего набора.

Теперь построим линейные зависимости признаков, т.е. корреляционную матрицу (Рис. 9):

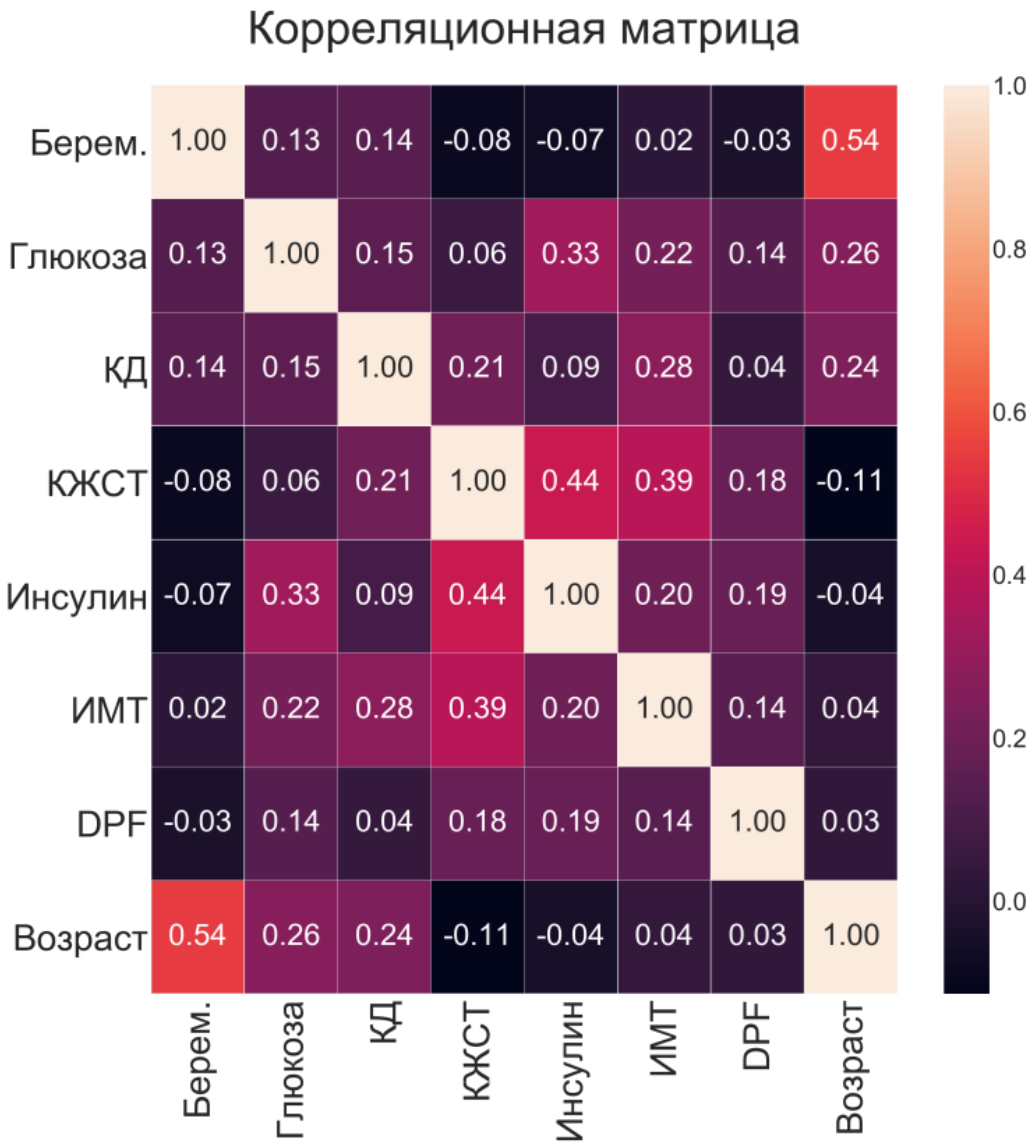


Рис. 9 Корреляционная матрица

Нетрудно заметить, что у нас нет сильно коррелированных признаков.

Построим график Violin (Рис. 10), показывающий медианы, верхние и нижние квартили:

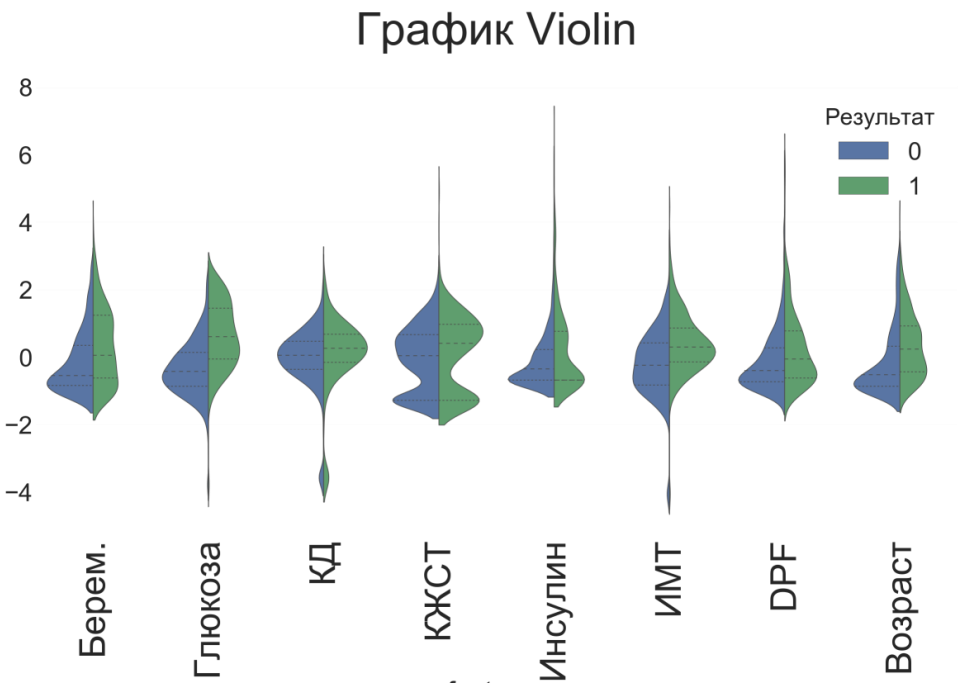


Рис.10 График Violin

По графику Violin можно заметить, что почти у всех признаков медианы (толстая пунктирная линия) для значений 0 и 1 отличаются, что говорит о том, что данные признаки могут быть полезны для классификации.

Построим график Ящик с усами (Boxplot) (Рис. 11), показывающий медианы, верхние, нижние квартили, 95% квантиль и 5% квантиль, а так же выбросы:

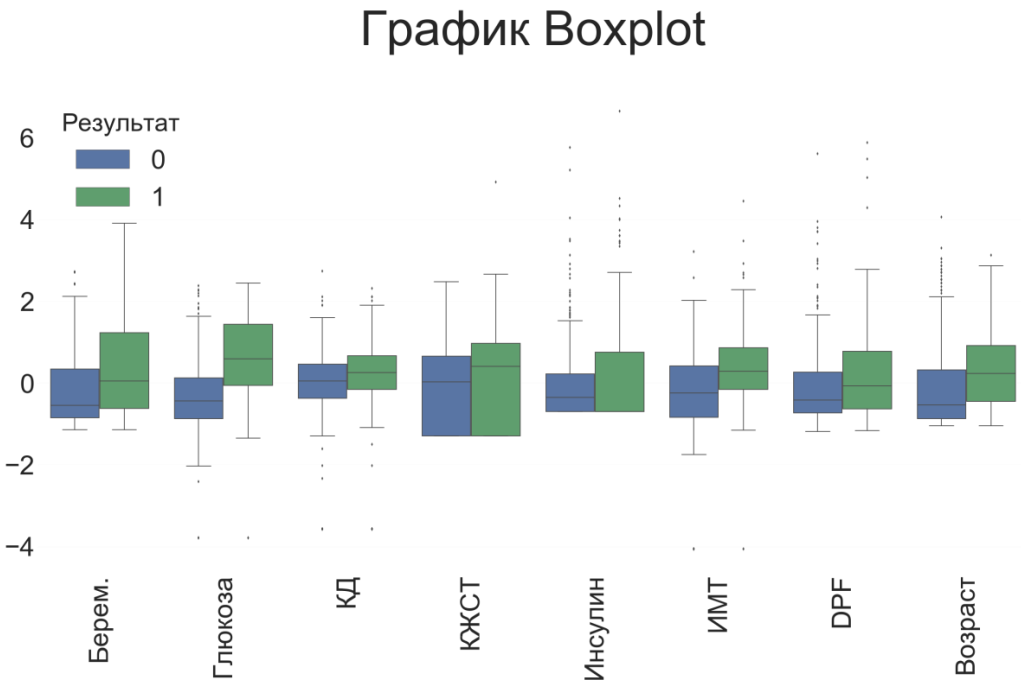


Рис. 11 График Boxplot

По данному графику видим, что в исходных данных много выбросов, что говорит о том, что перед тем, как начать обучение модели, необходимо их учесть. Т.е. как-то еще преобразовать или почистить входные данные.

Уменьшим размерность входных данных. Используем метод главных компонент и визуализируем (Рис. 12):

Рис. 12 Метод главных компонент, визуализация

После применения данного метода получили две главные компоненты (красные – больные, зелёные – здоровые). Нетрудно определить, что данные являются плохо разделимыми.

Итак, был проведен весь необходимый анализ данных для дальнейшего построения различных методов машинного обучения.

Глава 2. Программное построение моделей.

Построение прогноза о том, есть ли у человека диабет – задача бинарной классификации. Прежде всего, перед построением различных методов машинного обучения, необходимо выбрать метрику оценки качества модели.

2.1 Выбор метрики оценки качества

Accuracy

Проведенный ранее анализ с точностью указывает на то, что скошенных классов не наблюдается. Таким образом, можно взять метрику Accuracy в качестве оценки модели. Данная оценка достаточно проста и легко интерпретируется.

После обучения модели необходимо будет построить матрицу ошибок (Таблица 2), вида:

	$y = 0$	$y = 1$
$y' = 0$	True Negative (TN)	False Negative (FN)
$y' = 1$	False Positive (FP)	True Positive (TP)

Таблица 2

В данной таблице y – реальное значение класса, x – ответ алгоритма. Ошибки классификации: False Negative (FN) и False Positive (FP).

Расчёт метрики accuracy:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$accuracy = \frac{\sum \text{диагональных ячеек}}{\sum \text{всех ячеек}}$$

Так же для тестового набора мы рассчитаем метрики precision и recall.

Precision

Данную метрику можно интерпретировать, как «меткость» классификатора, т.е. как часто он попадает в фактическое значение, когда работает в данном классе.

Расчёт метрики precision:

$$precision = \frac{TP}{TP + FP}$$

$$precision = \sum \frac{\text{диагональный элемент}}{\sum \text{всех элементов соответсвующей строки}}$$

Recall

Интерпретируют, как «отзывчивость» классификатора, т.е. то, на сколько классификатор чувствует фактическое значение.

Расчёт метрики recall:

$$recall = \frac{TP}{TP + FN}$$

$$recall = \sum \frac{\text{диагональный элемент}}{\sum \text{всех элементов соответсвующего столбца}}$$

2.2 Предварительная обработка данных. Выбор параметров модели и кросс-валидация

Разбиваем данные на два класса: целевые признаки (0 и 1) и все остальные признаки (уровень глюкозы в плазме, уровень инсулина в плазме, индекс масс тела и т.д.).

Далее разбиваем все данные на тренировочные и тестовые. В результате получаем две выборки следующих размеров: тренировочная – 537, тестовая – 231.

Так же все значения, которые в классе всех остальных признаков = 0, заменим на -1, чтобы модель могла их лучше отличать.

Главная задача обучаемых алгоритмов – хорошо работать на новых данных. Поскольку на тестовых данных мы сразу не можем проверить качество построенной модели, то надо пожертвовать небольшой порцией тренировочных данных, чтоб на ней проверить качество модели. Для этого будем использовать кросс-валидацию (Рис. 13).

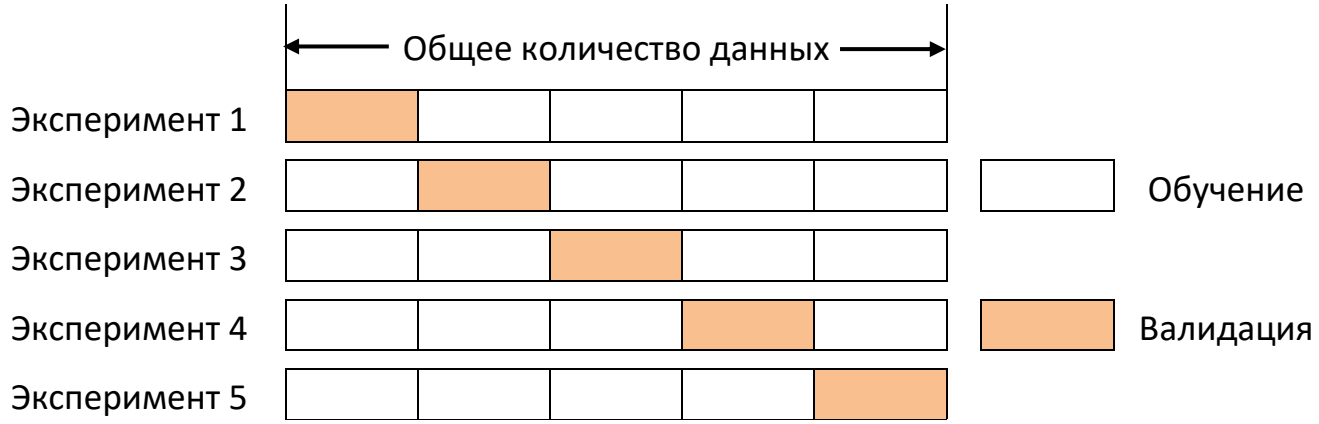


Рис. 13 Кросс-валидация

Модель обучается k раз на разных $k-1$ подвыборках исходной выборки (белый цвет), а проверяется на одной подвыборке (каждый раз на разной, оранжевый цвет). Получаются k оценок качества модели, которые обычно усредняются, выдавая среднюю оценку качества классификации/регрессии на кросс-валидации.

Поиск гиперпараметров для каждой построенной модели производим по сетке (решётке).

2.3 Логистическая регрессия

Логистическая регрессия – статистический метод классификации, значением функции которой является вероятность того, что исходное значение принадлежит к определённому классу. Логит-регрессия – один из самых используемых алгоритмов, применяемых в машинном обучении и в науке о данных [5].

Основная идея логистической регрессии

В рамках поставленной задачи будем считать, что необходимо разделить всех исследуемых на два класса: 1 – больные диабетом, 0 – здоровые. Идея заключается в том, что пространство исходных данных может быть линейно разделено на две соответствующих классам области. Под линейной границей подразумевается прямая в случае двух измерений, плоскость в случае трёх и т.д. Эта граница задаётся в зависимости от исходных данных и обучающего алгоритма.

В результате поиска по сетке устанавливаем следующие гиперпараметры:

- L1-регуляризация;

- параметр L1-регуляризации (принудительное понижение весов для предотвращения переобучения модели) составляет 0,716.

Обучили модель и получили следующие результаты.

Важность признаков (Рис. 14):

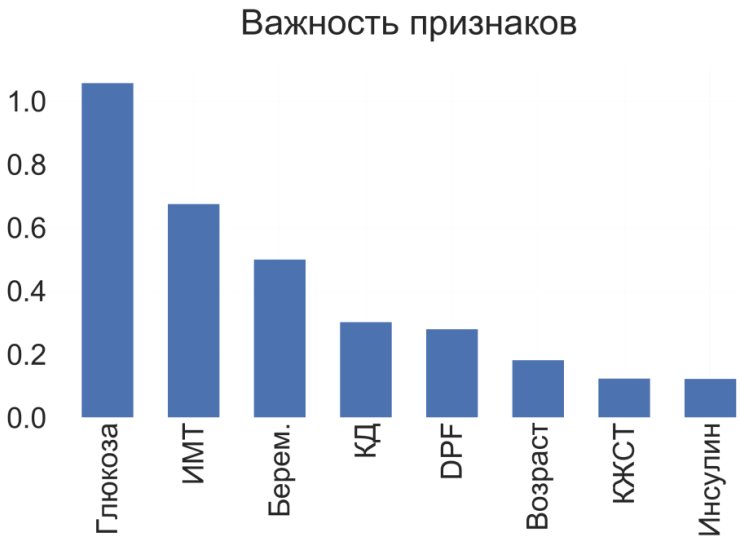


Рис. 14 Важность признаков логистической регрессии

Метрика качества accuracy:

Accuracy test score: 0.7575757575757576

Матрица ошибок (Рис. 15):

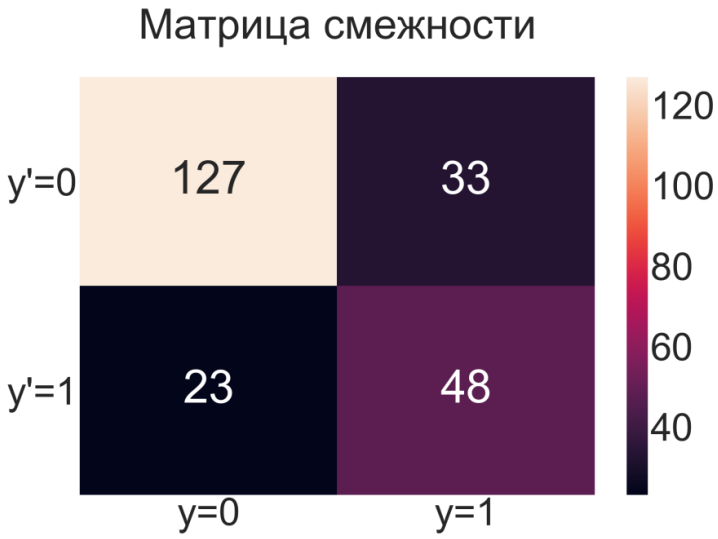


Рис. 15 Матрица смежности

Метрика качества precision:

Precision: 0.676056338028169

Метрика качества recall:

Recall: 0.5925925925925926

Логистическая регрессия определяет, болен человек диабетом или нет с точностью 75,8%.

2.4 Случайный лес

Случайный лес (Random Forest) является композицией (ансамблем) множества деревьев решений, что позволяет снизить проблему переобучения и повысить точность в сравнении с одним деревом. Прогноз получается в результате агрегирования ответов множества деревьев. Тренировка деревьев происходит независимо друг от друга (на разных подмножествах (используя бэггинг)).

В результате поиска по сетке устанавливаем следующие гиперпараметры:

- количество деревьев: 30;
- глубина каждого дерева: 5;
- максимальное количество признаков у каждого дерева, участвующих в разбиении sqrt(n);
- минимальное количество объектов в листовом узле: 2;
- минимальное количество объектов, необходимое для разделения внутреннего узла: 2.

Обучили модель и получили следующие результаты.

Важность признаков (Рис. 16):

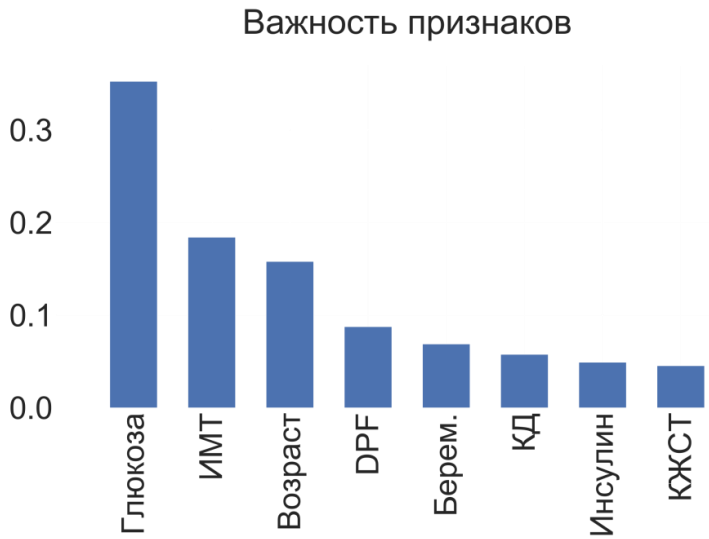


Рис. 16 Важность признаков случайного леса

Метрика качества accuracy:

Accuracy test score: 0.7748917748917749

Матрица ошибок (Рис. 17):

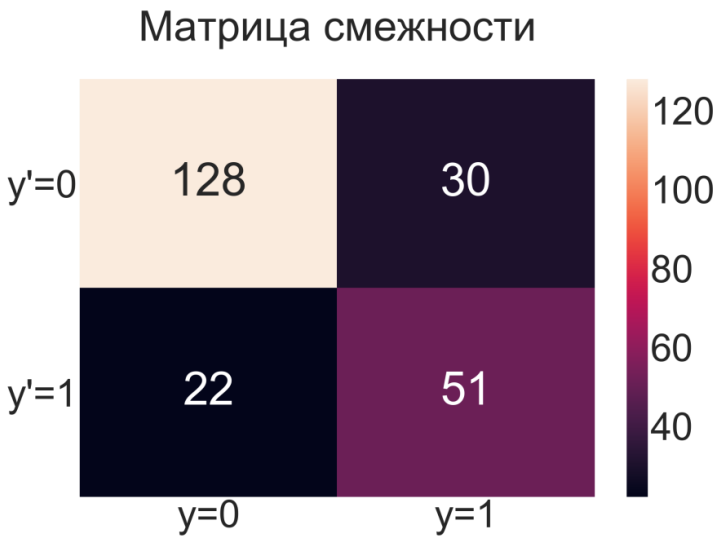


Рис. 17 Матрица смежности

Метрика качества precision:

Precision: 0.6986301369863014

Метрика качества recall:

Recall: 0.6296296296296297

Модель случайный лес определяет, болен человек диабетом или нет с точностью 77,5%.

2.5 Метод опорных векторов

Метод опорных векторов – алгоритм обучения с учителем, используется для задач классификации и регрессионного анализа. Данный метод направлен на уменьшение ошибки классификации и на максимизации полосы расширения между классами [6].

Основная идея SVM

Идея метода заключается в разделении данных на классы прямой. Такая прямая называется разделяющей прямой. Все новые данные не из обучающей выборки будут автоматически разбиваться на классы, определённые разделяющей прямой.

В результате поиска по сетке устанавливаем следующие гиперпараметры:

- параметр ошибки – 0.1, многомерное ядро;
- степень ядра – 2;
- свободный коэффициент – 0.67;
- использование эвристического алгоритма – true.

Обучили модель и получили следующие результаты.

Метрика качества accuracy:

Accuracy test score: 0.7662337662337663

Матрица ошибок (Рис.18):

Рис. 18 Матрица смежности

Метрика качества precision:

Precision: 0.7288135593220338

Метрика качества recall:

Recall: 0.5308641975308642

Метод опорных векторов определяет, болен человек диабетом или нет с точностью 76,6%.

2.6 Многослойный персептрон

Персептрон (нейрон) – это функция, которая принимает своё значение по нескольким переданным в неё параметрам [7].

Основная идея нейросети (НС)

Обучение персептрона

Обучение происходит следующим образом:

1. Вопрос нейрону. То есть подаём на него входной вектор.

2. Он выдает ответ на этот вопрос, сравниваем его с правильным ответом.

3. Далее проводим обучение нейрона на правильном ответе.

Чтобы обучить нейрон надо знать правильный ответ. То есть, какое правильное соответствие между входом и выходом. Если мы его знаем, то можем научить нейрон его реализовывать. В результате нейрон принимает решения, основываясь на полученных весах. Если ответ выдан неверный, то данные веса пересчитываются: повышаются или понижаются в зависимости от входящего значения.

Многослойный персептрон

Один нейрон (однослойный персептрон) не сможет решить задачи с большим количеством данных, нелинейные задачи, задачи с несовместными системами неравенств и другие. Несколько соединённых друг с другом нейронов, образующих большую сеть, вместе смогут запомнить, выучить и обработать даже самые сложные функции [8].

В данной работе многослойный персептрон решает задачу прогнозирования диабета на начальном этапе.

В результате поиска по сетке устанавливаем следующие гиперпараметры:

- скрытые слои: 4;
- нейронов в каждом слое: 10;
- L2-регуляризация;
- параметр L2-регуляризации (принудительное понижение весов для предотвращения переобучения модели) составляет 0,33.

Обучили модель и получили следующие результаты.

Метрика качества accuracy:

Accuracy test score: 0.7619047619047619

Матрица ошибок (Рис. 19):

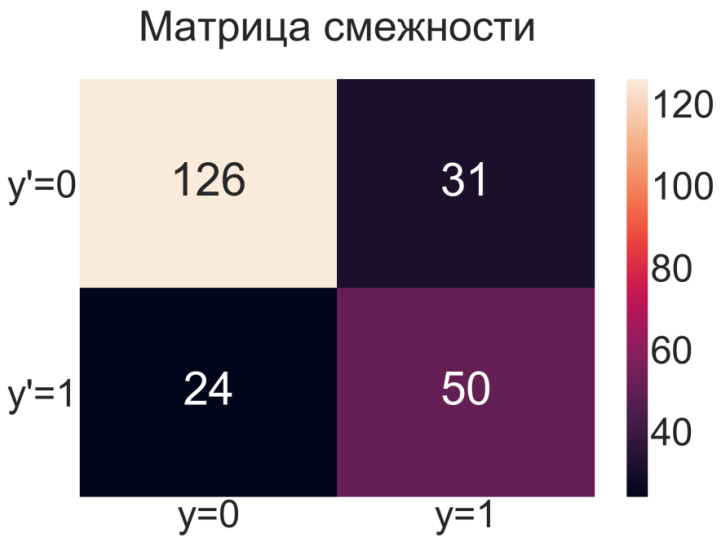


Рис. 19 Матрица смежности

Метрика качества precision:

Precision: 0.6756756756756757

Метрика качества recall:

Recall: 0.6172839506172839

Многослойный персептрон определяет, болен человек диабетом или нет с точностью 76,2%.

2.7 Сравнительный анализ методов машинного обучения

Результаты по ранее построенным методам машинного обучения:

1. ***Логистическая регрессия*** определяет, болен человек диабетом или нет с точностью 75,8%.
2. ***Модель случайный лес*** определяет, болен человек диабетом или нет с точностью 77,5%.
3. ***Метод опорных векторов*** определяет, болен человек диабетом или нет с точностью 76,6%.
4. ***Многослойный персептрон*** определяет, болен человек диабетом или нет с точностью 76,2%.

В процессе сравнения методов машинного обучения для прогнозирования диабета у пациента на начальном этапе, установлено, что наиболее эффективным является модель «случайный лес». Данный метод показывает наилучшие результаты по следующим причинам:

— практически не чувствителен к выбросам в данных;

— не чувствителен к масштабированию признаков;

— не требует тщательной настройки параметров;

— обрабатывает непрерывные и дискретные признаки одинаково хорошо без дополнительной обработки;

— устойчив к переобучению;

— хорошо обрабатывает пропущенные данные.

Архитектурные особенности НС требуют более тщательной настройки параметров и обработки данных:

— удаление/обработка выбросов в данных;

— масштабирование непрерывных признаков;

— кодирование дискретных признаков (one-hot encoding);

— для устранения переобучения необходима регуляризация;

— удаление/заполнение пропущенных данных.

Так как в исходных данных присутствуют выбросы и пропущенные данные, имеются как непрерывные так и дискретные признаки, которые имеют разный масштаб, то модель случайного леса показывает более точный результат. Также, данных небольшое количество, поэтому случайный лес не переобучается, в отличие от НС, которая переобучается на тренировочном множестве. Однако, увеличение количества данных может позволить НС показывать более точный результат.

Анализ заболевания СД2 в РФ

Сахарный диабет 2 типа – сложно диагностируемое заболевание, которое является серьёзной медико-социальной и экономической проблемой как в Российской федерации, так и во всём мире. Самым тяжёлым последствием СД2 является высокая вероятность появления и распространения сердечно-сосудистых осложнений (ССО), что является основной причиной смерти людей с СД2 (вследствие инфаркта, инсульта, сердечно-сосудистой недостаточности).

СД прогрессивно растёт (Диаграмма 5):

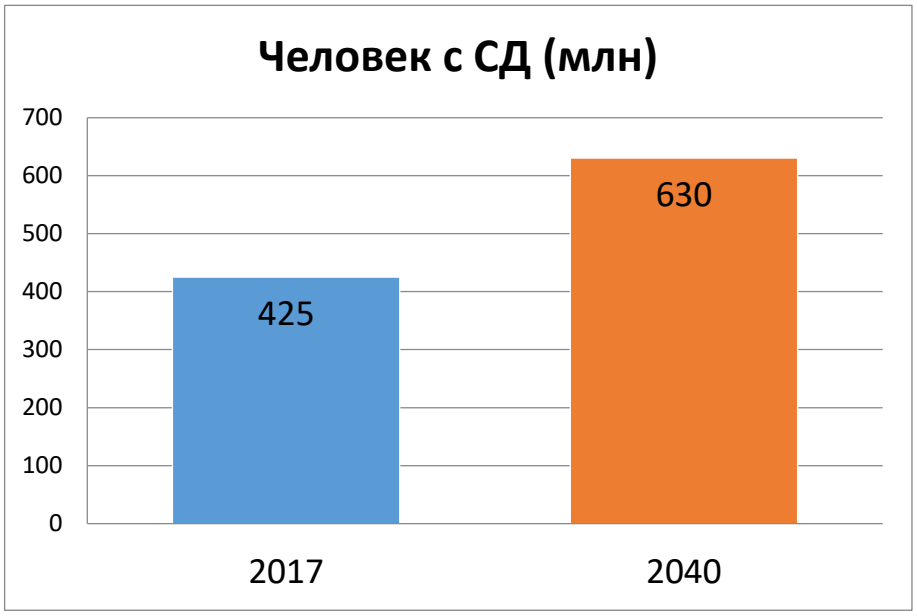


Диаграмма 5

- На 1 взрослого с выявленным СД приходится 1 человек, у которого не диагностировано СД;
- 4 млн смертей в год, связанных с осложнением СД;
- Каждые 8 секунд от осложнений СД погибает один человек;
- Основная причина смерти людей с СД – сердечно-сосудистые заболевания.

Проведем анализ затрат, связанных с СД2 для женщин в возрасте от 20 лет. Рассмотрим данные РОССТАТ на 2017 год:

- численность населения составляет 146,8 млн;
- численность женщин составляет 78,7 млн;
- численность женщин в возрасте от 20 лет составляет 63 млн.

По исследованию российских ученых, на 2017 год [16]:

- общая численность пациентов, в возрасте от 20 лет, с СД2 составила 4,15 млн;
- распределение мужчин/женщин (в возрасте от 20 лет): 29%/71%, то есть численность женщин с СД2 составляет 2,94 млн. (Диаграмма 6);

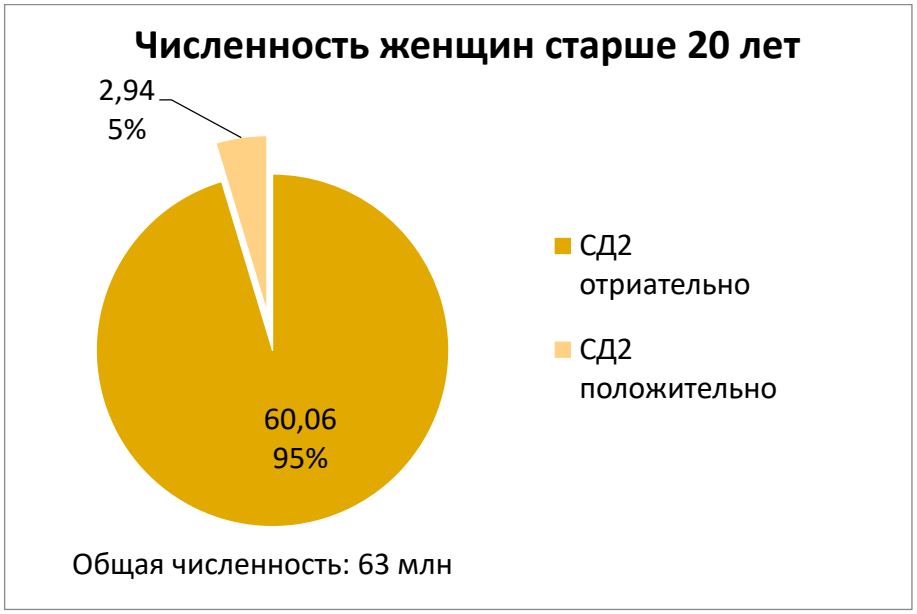


Диаграмма 6

- распределение СД2 своевременная/несвоевременная диагностика: 70%/30% , то есть численность женщин с несвоевременной диагностикой составляет 0,883 млн (Диаграмма 7);

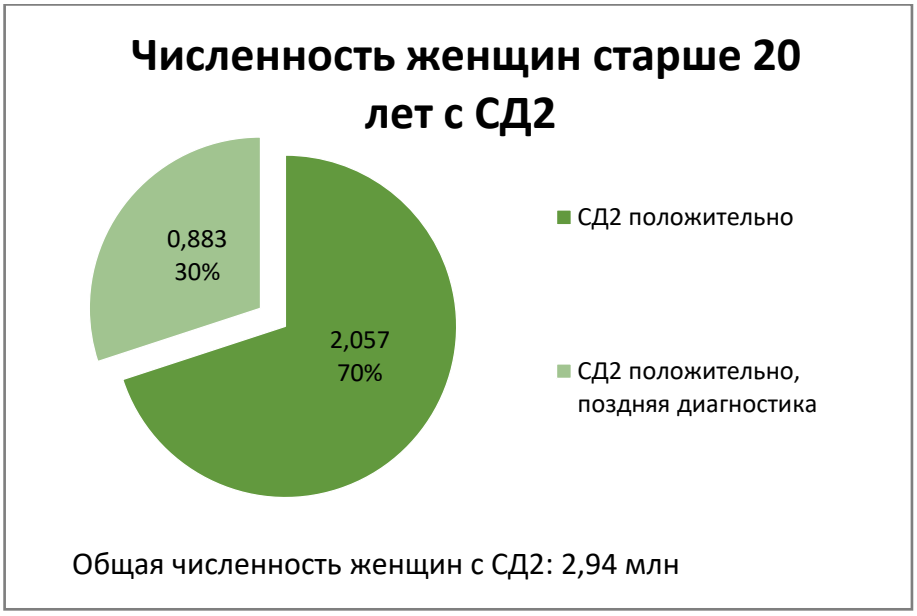


Диаграмма 7

На данный момент затраты, которые идут на СД2 составляют около 569 млрд в год, что соответствует 1% всего внутреннего валового продукта (ВВП) РФ. 34,7%

от этой суммы приходится на ССО, остальная сумма уходит на немедицинские затраты (потери ВВП) из-за временной нетрудоспособности, инвалидизации, преждевременной смертности. Таким образом, на женщин с данным заболеванием уходит порядка 403 млрд в год [17].

Затраты на СД2, при своевременном диагностировании, составляют 88 982 руб/пациент/год, в ином случае в 2,8 раз больше (то есть с ССО) составляют 249 149 руб/пациент/год (Диаграмма 8). Данные цифры указывают на важность решения задачи диагностирования СД2 на начальном этапе.

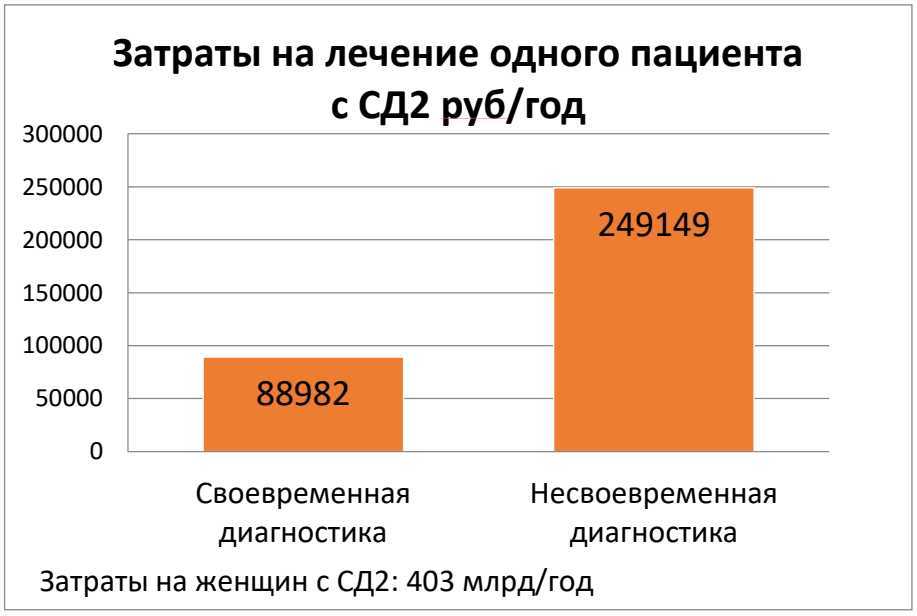


Диаграмма 8

Отметим, что в год на всех женщин от 20 лет с СД2 с несвоевременной диагностикой уходит 0,883 млн человек * 249149 руб = 220 млрд руб (Диаграмма 9).

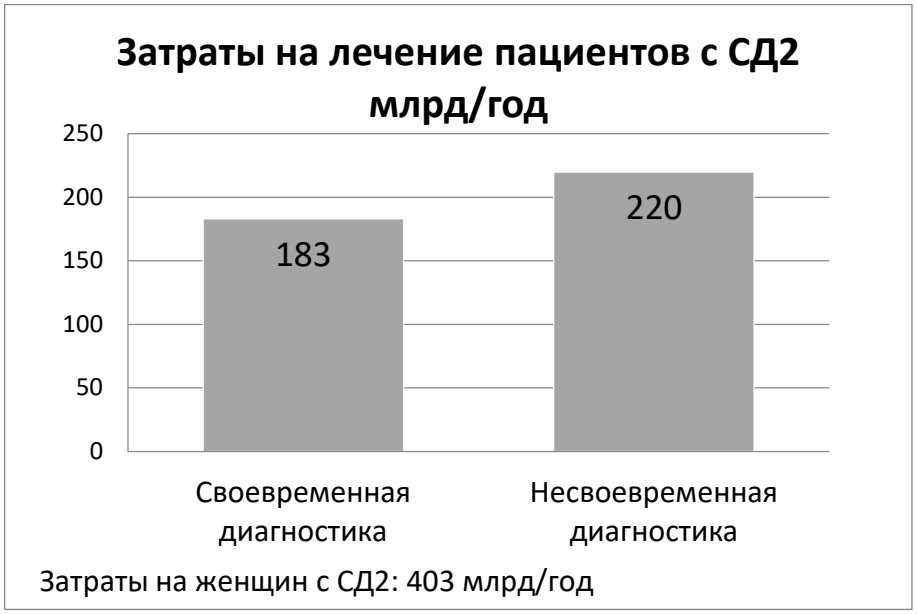


Диаграмма 9

В рамках данной работы ранее были выделены измерения и исследования, влияющие на развитие СД2. Рассмотрим затраты, которые необходимо потратить на одну женщину для получения этих параметров:

- Анализ на ***глюкозу***: средняя цена 1000 руб
- Анализ на ***инсулин***: средняя цена 1000 руб
- ***Толщина кожно-жировой складки трицепса*** (измерить самостоятельно): бесплатно
- ***Индекс массы тела (ИМТ)*** $\frac{\text{вес в кг}}{(\text{рост в м})^2}$ (пациенту необходимо знать свой вес в кг и рост в м): бесплатно
- ***Диастолическое давление***: бесплатно
- ***Количество беременностей***: бесплатно
- ***Функция диабетической родословной*** (пациенту необходимо знать: 1) был ли у кого СД2 в семье, указать у кого и в каком возрасте поставлен диагноз; 2) указать остальных родственников и возраст в котором они в последний раз сдавали/проводили обследование на СД2): бесплатно
- ***Возраст***: бесплатно

Средние затраты, которые необходимо потратить на одну женщину для проведения данного исследования составляют 2000 руб. Таким образом, чтобы

диагностировать СД2 для всех женщин в возрасте от 20 лет необходимо потратить 63 млн * 2000 руб = 126 млрд руб /год.

Полученная сумма в 126 млрд составляет 31,2% от суммы текущих затрат бюджета РФ на СД2 (т.е. от 403 млрд) (Диаграмма 10).

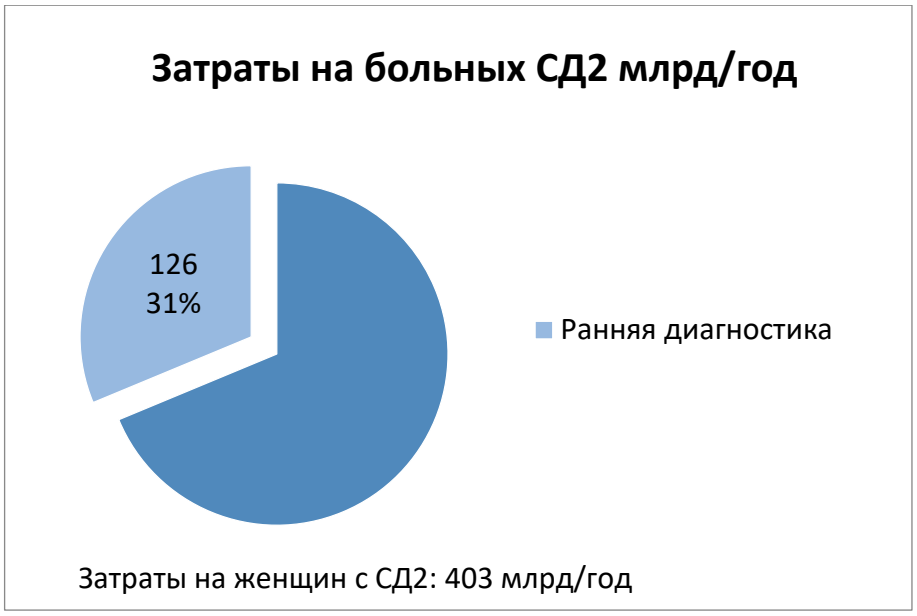


Диаграмма 10

На данный момент средние затраты, которые тратят на одну женщина для диагностирования диабета составляют 5000 руб. То есть на диагностику для всех, у кого выявлен СД2 было потрачено 5000 руб * 2,94 ≈ 15 млрд руб. Соответственно, если провести диагностику для всех женщин вовремя, то получим следующие затраты: 126 млрд (ранняя диагностика для всех 63 млн женщин от 20 лет) + 2,94 (количество больных СД2) * 88982 руб (стоимость лечения при своевременной диагностике) – 15 млрд руб (затраты на диагностику тех у кого на данный момент выявляют СД2 несвоевременно) = 373 млрд руб. Следовательно, используя предложенный подход диагностирования СД2 не только упростит прогнозирование данного заболевания, но и будет способствовать сокращению затрат на него. То есть, 403 млрд – 373 млрд = 30 млрд удастся сэкономить (Диаграмма 11).

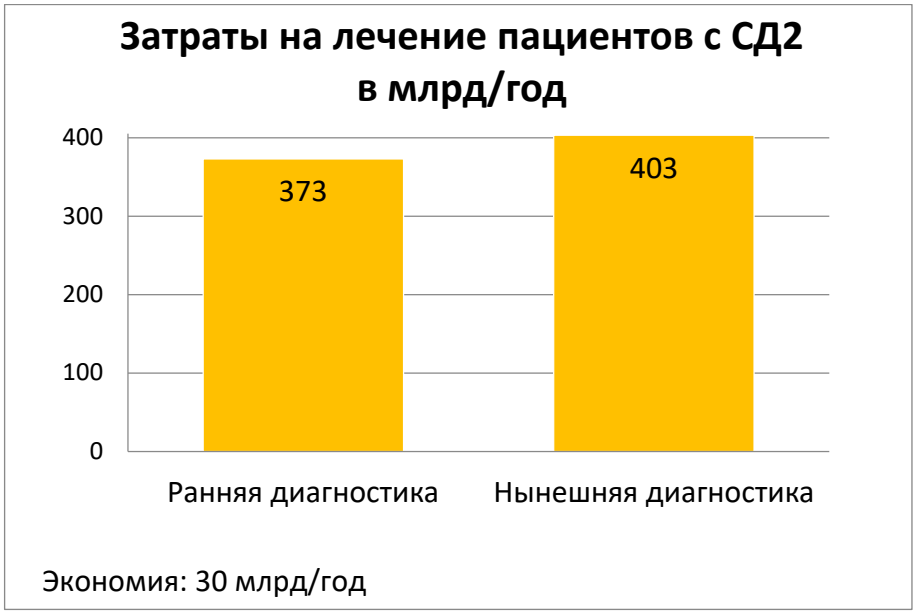


Диаграмма 11

В настоящее время в РФ происходит цифровизация отрасли здравоохранения. Многие компании, такие как «Росгосстрах», «ВТБ», «МТС», заинтересованы уходить в рынок персонализированной медицины, но необходимо корректно выстроить стратегию её продвижения, чтобы сформировать практику обращения к врачу и использования сервисов удалённого мониторинга вместо поисков в интернете, вопросов друзьям и самолечения. Соответственно, возникают новые источники продвижения, т.е. бюджет необходимый для решения задачи прогнозирования СД2 можно выделять из бюджета на цифровизацию, тем самым уменьшая нагрузку на бюджет здравоохранения. Переход на цифровые технологии привлекает компании, которые готовы инвестировать деньги, чтобы привлечь большее количество клиентов для услуг телемедицины в рамках реализации персонализированной медицины. Таким образом, сумму в 126 млрд можно разделить пополам: первую часть будет оплачивать государство, вторую – крупные инвесторы. В результате получим эффективное государственно-частное партнёрство, которое позволит улучшить диагностирование СД2 на ранних этапах, что снизит вероятность ССО и увеличит эффективность лечения, а так же позволит существенно уменьшить экономические затраты.

Заключение

Таким образом, в данной работе была изучена предметная область, а именно болезнь диабет. Были приведены все необходимые данные и их подробный анализ для её диагностирования. Программно построены различные методы машинного обучения для прогнозирования диабета у пациента на начальном этапе. Проведён их сравнительный анализ [9]. Установлено, что наиболее эффективным является метод «случайный лес». Однако, если увеличить количество входных данных НС сможет показать более точный результат.

В результате проделанной работы был сделан вывод о том, что вопрос диагностирования СД2 на начальном этапе, увеличение эффективности лечения, а также минимизации экономических затрат является актуальным и может быть решен путём программного построения методов машинного обучения для прогнозирования СД2, цифровизации отрасли здравоохранения и привлечение крупных компаний для инвестирования.

Список литературы

[1] Char, D. S., Shah, N. H., & Magnus, D. (2018). Implementing Machine Learning in Health Care—Addressing Ethical Challenges // New England Journal of Medicine,378(11), 981-983.

[2] Alpaydin, Ethem (2010). Introduction to Machine Learning.// London: The MIT Press.

[3] Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2012). Foundations of Machine Learning.// USA, Massachusetts: MIT Press.

[4] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, R. S. Johannes (1988). Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus// Proc Annu Symp Comput Appl Med Care. P. 261–265.

[5] David W. Hosmer, Stanley Lemeshow. Applied Logistic Regression.// 2nd ed. New York, Chichester, Wiley. 2002. 392 P.

[6] К. В. Воронцов (2007) Лекции по методу опорных векторов.// http://www.ccas.ru/voron/download/SVM.pdf

[7] Haykin, Simon (1998). Neural Networks: A Comprehensive Foundation (2 ed.).// Prentice Hall.

[8] Shanker M1, Hu MY, Hung MS (1999). Estimating Probabilities of Diabetes Mellitus Using Neural Networks.// SAR QSAR Environ Res.11(2):133-47.

[9] R. Collobert and S. Bengio (2004). Links between Perceptrons, MLPs and SVMs.// Proc. Int'l Conf. on Machine Learning (ICML).

[10] Jönsson B.(1998) The economic impact of diabetes.// Diabetes Care. Suppl 3: C7-10.

[11] Zimmet PZ. (1995) The pathogenesis and prevention of diabetes in adults. Genes, autoimmunity, and demography.// Diabetes Care. Jul;18(7):1050-64.

[12] Zimmet, P., Alberti, K. G. & Shaw, J. (2001) Global and societal implications of the diabetes epidemic. // Nature 414, 782–787.

[13] Chen, L, Magliano, DJ and Zimmet, PZ. (2011) The worldwide epidemiology of type 2 diabetes mellitus-present and future perspectives.// Nat Rev Endocrinol 8: 228-236

[14] Danaei, G. et al. (2011) National, regional, and global trends in fasting plasma

glucose and diabetes prevalence since 1980: systematic analysis of health examination surveys and epidemiological studies with 370 country-years and 2.7 million participants. // Lancet 378, 31–40.

[15] Shaw, J. E., Sicree, R. A. & Zimmet, P. Z. (2010) Global estimates of the prevalence of diabetes for 2010 and 2030. // Diabetes Res. Clin. Pract.87, 4–14.

[16] Дедов Иван Иванович, Шестакова Марина Владимировна, Викулова Ольга Константиновна, Железнякова Анна Викторовна, Исаков Михаил Андреевич (2018) Сахарный диабет в Российской Федерации: Распространённость, заболеваемость, смертность, параметры углеводного обмена и структура сахароснижающей терапии по данным федерального регистра сахарного диабета, статус 2017 г // Сахарный диабет. 2018. №3. URL: https://cyberleninka.ru/article/n/saharnyy-diabet-v-rossiyskoy-federatsii-rasprostranennost-zabolevaemost-smertnost-parametry-uglevodnogo-obmena-i-struktura (дата обращения: 21.04.2019).

[17] Иван Иванович Дедов, Концевая Анна Васильевна, Шестакова Марина Владимировна, Белоусов Юрий Борисович, Баланова Юлия Андреевна, Худяков Михаил Борисович, Олег Ильич Карпов (2016) Экономические затраты на сахарный диабет 2 типа и его основные сердечно-сосудистые осложнения в Российской Федерации // Сахарный диабет. 2016. №6. URL: https://cyberleninka.ru/article/n/ekonomicheskie-zatraty-na-saharnyy-diabet-2-tipa-i-ego-osnovnye-serdechno-sosudistye-oslozhneniya-v-rossiyskoy-federatsii (дата обращения: 21.04.2019).

Программный код

In [ ]:

```
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt

from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.model_selection import train_test_split, GridSearchCV, \
StratifiedKFold, learning_curve, validation_curve

from sklearn.neural_network import MLPClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC

from sklearn.metrics import accuracy_score, confusion_matrix, \
precision_score, recall_score
from sklearn.decomposition import PCA

from scipy.stats import mannwhitneyu

from scipy import sparse

sns.set(font_scale=1.5)
pd.options.display.max_columns = 50

import warnings
warnings.filterwarnings('ignore')
```

In [ ]:

```
names = [
    "Берем.",
    "Глюкоза",
    "КД",
    "КЖСТ",
    "Инсулин",
    "ИМТ",
    "DPF",
    "Возраст",
    "Результат"
]
```

In [ ]:

```
outcome = names[-1]
```

In [ ]:

```
data = pd.read_csv('../../data/raw/diabetes.csv', names=names, header=0)
```

Data analysis

General data overview

In [ ]:

```
data.head()
```

In [ ]:

```
data.info()
```

In [ ]:

```
data.shape
```

Missing values

In [ ]:

```
print("Are there missing values:", data.isnull().values.any())
```

Summary statistics

In [ ]:

```
data.describe()
```

Target feature

In [ ]:

```
data[outcome].value_counts()
```

Let's check the ratio of examples belong to each class:

In [ ]:

```
data[outcome].value_counts() / data[outcome].size
```

In [ ]:

```
scaler = StandardScaler()

scaled_data = scaler.fit_transform(data)
data_scaled = pd.DataFrame(scaled_data, columns=data.columns)
data_scaled[outcome] = data[outcome]
```

In [ ]:

```
data_z = pd.melt(data_scaled,
                 id_vars=outcome,
                 var_name="features",
                 value_name='value')
```

Linear dependecies of the features (correlation matrix):

In [ ]:

```
def plot_corr(data):
    sns.set(font_scale=6)

    plt.figure(figsize=[40, 40])

    ax = sns.heatmap(data.corr(), annot=True, fmt= '.2f', linewidths=.5)
    ax.set_xticklabels(ax.get_xticklabels(), size='large')
    ax.set_yticklabels(ax.get_yticklabels(), size='large')

    plt.xticks(rotation=90);
    plt.yticks(rotation=0);
    plt.title("Корреляционная матрица", pad=100, fontdict={'fontsize': 110});
    plt.savefig('corr.png', bbox_inches='tight', transparent=True)
    plt.show();

```

In [ ]:

```
plot_corr(data.drop([outcome], axis=1))
```

Conclusion: there are no highly correlated features.

Distribution of classes

In [ ]:

```
sns.set(font_scale=6)
plt.figure(figsize=(40, 20));
ax = sns.violinplot(x="features",
                    y="value",
                    hue=outcome,
                    data=data_z,
                    split=True,
                    inner="quartile");
ax.set_xticklabels(ax.get_xticklabels(), size='large');
plt.xticks(rotation=90);
plt.title("График Violin", pad=100, fontdict={'fontsize': 110});
plt.savefig('viol.png', bbox_inches='tight', transparent=True)
```

Conclusion: in some features, like Glussian, median of each class separated, so they can be useful for classification. Other features, like smoothness_se, are not so separated and my be less useful for classification. Most all the features have normal-like distribution with long tail.

Outliers

In [ ]:

```
sns.set(font_scale=6)
plt.figure(figsize=(40, 20));
ax = sns.boxplot(x='features', y='value', hue=outcome, data=data_z);
ax.set_xticklabels(ax.get_xticklabels());
plt.xticks(rotation=90);
plt.title("График Boxplot", pad=100, fontdict={'fontsize': 110});
plt.savefig('box.png', bbox_inches='tight', transparent=True)
```

Conclusion: there are a lot of variable with outliers. So before training we have to handle it.

In [ ]:

```
def plot_pair_hist(data, feature):

    sns.set(font_scale=6)

    plt.figure(figsize=(40, 20));
    plt.title(feature, pad=100, fontdict={'fontsize': 110});

    ax = sns.distplot(
        data[feature][data[outcome]==0],
        bins=60,
        kde=False,
        color='green'
    )

    ax = sns.distplot(
        data[feature][data[outcome]==1],
        bins=60,
        kde=False,
        color='blue'
    )

    plt.legend(data[outcome])
    ax.set_xlabel('')

    plt.savefig('%s.png'\
                % (feature), bbox_inches='tight', transparent=True)

    plt.show()
```

In [ ]:

```
for name in names:
    plot_pair_hist(data, name)
```

Dimensionality reduction

In [ ]:

```
pca_two_comp = PCA(n_components=2, random_state=24)
two_comp_data = pca_two_comp.fit_transform(scaled_data)

sns.set(font_scale=6)
plt.figure(figsize=(40, 30));

plt.scatter(
    x=two_comp_data[:, 0],
    y=two_comp_data[:, 1],
    c=data_scaled[outcome].map({ 1: 'red', 0: 'blue'}),
    s=900
)

plt.title("Метод главных компонент", pad=100, fontdict={'fontsize': 110});
plt.savefig('pca.png', bbox_inches='tight', transparent=True)
plt.show()
```

Conclusion: data is't good enough separable using only two components.

Metrics selection

Predict whether the positive for diabet or no is a binary classification task. Here we don't face the probem of skewed classes. So accuracy metric will be a good choice for model evaluation. Also this metric is simple enough, thus highly interpretable. Aslo for the test set we will calculate precision and recall.

Data preprocessing

In [ ]:

```
X = data.drop([outcome], axis=1)
y = data[outcome]
```

In [ ]:

```
print('Total number of examples:', X.shape[0])
```

Train/test split

In [ ]:

```
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.3,
    random_state=24,
    stratify=y
)

print('Train size:', X_train.shape[0])
print('Test size:', X_test.shape[0])
```

Feature scaling

Treating all featurese as non-catogorical.

In [ ]:

```
scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Result data

In [ ]:

```
X_train_res = pd.DataFrame(X_train_scaled, columns=X_train.columns)
X_test_res = pd.DataFrame(X_test_scaled, columns=X_test.columns)
```

Model training

In [ ]:

```
def linear_model():
    model = LogisticRegression(random_state=24)
    model_parameters = {
        'penalty': ['l1', 'l2'],
        'C': np.linspace(.1, 1, 20)
    }

    return model, model_parameters
```

In [ ]:

```python
def tree_model():
    model = RandomForestClassifier(random_state=24)

    model_parameters = {
        'n_estimators': range(10, 100, 10),
        'max_depth': range(1, 10, 2),
        'max_features': ['sqrt', 'log2'],
        'min_samples_split': range(2, 5, 1),
        'min_samples_leaf': range(2, 5, 1)
    }

    return model, model_parameters
```

In [ ]:

```python
def svm_model():
    model = SVC(random_state=24)

    model_parameters = {
        'C': np.linspace(0.1, 1, 5),
        'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
        'degree': range(1, 5, 1),
        'coef0': np.linspace(0, 2, 10),
        'shrinking': [True, False]
    }

    return model, model_parameters
```

In [ ]:

```python
def mlp_model():
    model = MLPClassifier(
        max_iter=1000,
        random_state=24
    )

    model_parameters = {
        'alpha': np.linspace(0.0001, 1, 10),

        'hidden_layer_sizes': [
            (10, 10, 10, 10),
            (10, 10, 10, 10, 10),
            (15, 15, 15, 15, 15),
            (8, 8, 8, 8, 8),
            (9, 9, 9, 9, 9, 9),
            (100, 100, 100, 100)
        ]
    }

    return model, model_parameters
```

In [ ]:

```
def get_model(type='linear'):

    if type == 'linear':
        return linear_model()

    elif type == 'tree':
        return tree_model()

    elif type == 'svm':
        return svm_model()

    elif type == 'mlp':
        return mlp_model()
```

In [ ]:

```
def train_model(X, y, type='linear'):
    model, model_parameters = get_model(type=type)

    cv = StratifiedKFold(n_splits=3, random_state=24)

    grig_search = GridSearchCV(
        model,
        model_parameters,
        n_jobs=-1,
        cv=cv,
        scoring='accuracy'
    )

    grig_search.fit(X, y);

    print('Model:', type)
    print('Best parameters:', grig_search.best_params_)
    print('CV accuracy:', grig_search.best_score_)

    return grig_search.best_estimator_
```

In [1]:

```
def forest_feature_importances(forest, labels=None):
    importances = forest.feature_importances_
    std = np.std([tree.feature_importances_ for tree in forest.estimators_],
                 axis=0)
    indices = np.argsort(importances)[::-1]

    features_count = importances.shape[0]
    feature_labels = indices if labels is None else labels[indices]

    # Print the feature ranking
    print("Feature ranking:")

    for f in range(features_count):
        print("%d. feature %s (%f)" \
              % (f + 1, feature_labels[f], importances[indices[f]]))

    # Plot the feature importances of the forest
    plt.figure()
    plt.figure(figsize=(40, 20));

    plt.title(
        "Важность признаков",
        pad=100,
        fontdict={'fontsize': 110}
    );

    plt.bar(
        range(features_count),
        importances[indices],
        align="center",
        width=0.6
    )

    plt.xticks(range(features_count), feature_labels, rotation=90)
    plt.xlim([-1, features_count])

    plt.savefig(
        'forest_feature_importances.png',
        bbox_inches='tight',
        transparent=True
    )

    plt.show()
```

In [ ]:

```python
def linear_feature_importances(estimator, labels=None):
    scores = np.abs(estimator.coef_[0])

    ttt = list(zip(labels, scores))
    ttt1 = sorted(ttt, key=lambda tup: tup[1], reverse=True)

    names_sorted = list(zip(*ttt1))[0]
    scores_sorted = list(zip(*ttt1))[1]

    sns.set(font_scale=9)
    plt.figure(figsize=(40, 20));
    plt.title("Важность признаков", pad=100, fontdict={'fontsize': 110});
    plt.bar(x=range(1, 9), height=scores_sorted, width=0.6)
    plt.xticks(range(1, 9), names_sorted, rotation=90);

    plt.savefig(
        'linear_feature_importances.png',
        bbox_inches='tight',
        transparent=True
    )
```

In [3]:

```python
def plot_confusion_matrix(y, predictions, type):
    mtrx = confusion_matrix(predictions, y);

    sns.set(font_scale=10)
    plt.figure(figsize=[30, 20])
    ax = sns.heatmap(mtrx, annot=True, fmt='d');

    ax.set_xticklabels(['y=0', 'y=1'], size='small')
    ax.set_yticklabels(["y'=0", "y'=1"], size='small')

    plt.yticks(rotation=0);

    plt.title(
        "Матрица смежности",
        pad=100,
        fontdict={'fontsize': 110}
    );

    plt.savefig(
        'conf-%s.png' % (type),
        bbox_inches='tight',
        transparent=True
    )

    plt.show();
```

In [ ]:

```
def evaluate(X, y, model, type):
    predictions = model.predict(X)

    accuracy = accuracy_score(y, predictions)
    precision = precision_score(y, predictions)
    recall = recall_score(y, predictions)

    print('Test accuracy:', accuracy)
    print('Test precision:', precision)
    print('Test recall:', recall)

    plot_confusion_matrix(y, predictions, type)
```

In [ ]:

```
def make_experiment(X_train, y_train, X_test, y_test):

    linear_model = train_model(X_train, y_train, type='linear')
    evaluate(X_test, y_test, linear_model, type='linear')
    linear_feature_importances(linear_model, X_train.columns.values)

    tree_model = train_model(X_train, y_train, type='tree')
    evaluate(X_test, y_test, tree_model, type='tree')
    forest_feature_importances(tree_model, X_train.columns.values)

    svm_model = train_model(X_train, y_train, type='svm')
    evaluate(X_test, y_test, svm_model, type='svm')

    mlp_model = train_model(X_train, y_train, type='mlp')
    evaluate(X_test, y_test, mlp_model, type='mlp')
```

In [ ]:

```
make_experiment(X_train_res, y_train, X_test_res, y_test)
```