Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский Томский политехнический Университет»



Институт

ИЯТШ

01.03.02 «Прикладная математика и информатика»

**Отчет о практике**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Выполнил:** |  | | | | |
| студент группы | 0В91 |  | Змитрович Н.Л. |  |
|  |  |  |  |  |
| **Проверил:** | Чумаков А.Г. | | | | |
| преподаватель |  |  |  |  |  |
|  |  |  |  |  |  |

Томск – 2021

**Задание**

Используя данные и их описание с источника <https://www.kaggle.com/uciml/pima-indians-diabetes-database> сделайте анализ и обучите модель предсказывать наличие диабета.

**Ход работы**

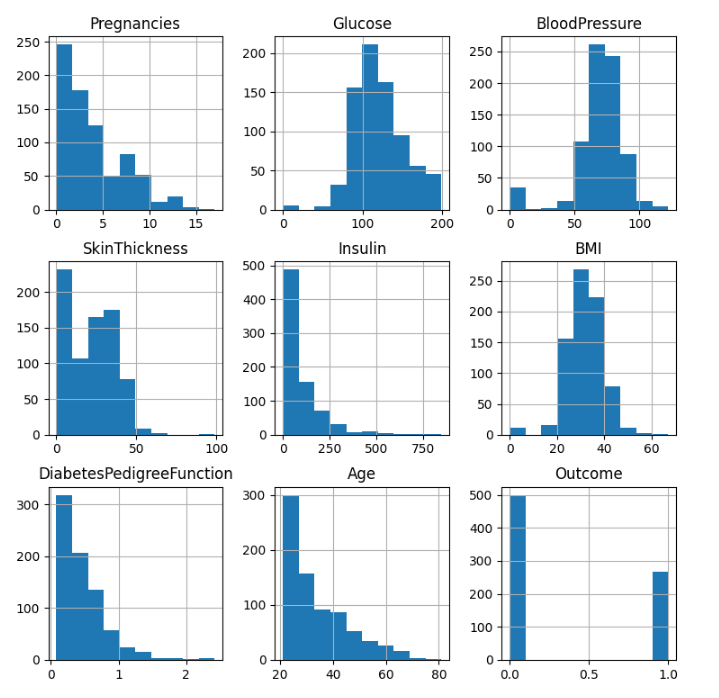
**Инициализация данных**

Для выполнения работы скачаем данные с сайта kaggle и создадим папку data в директории с jupyter notebook. Инициализируем библиотеки pandas для работы с данными и matplotlib, для построения графиков, а также применим функцию самого jupyter notebook %matplotlib notebook, для того, чтобы в дальнейшем можно было изменять размеры графиков. После чего средствами pandas импортируем базу данных и отобразим несколько первых элементов, для визуального понимания структуры данных.

**Первичный анализ**

Имеет восемь различных независимых переменных и одну зависимую – **Outcome.** Если значение Outcome пациента равно один, значит у него обнаружен диабет, если ноль, значит у пациента нет диабета. С помощью метода describe проверим основные параметры каждой из переменных, а также узнаем количество пациентов. При отображении первых нескольких строк таблицы, мы можем увидеть, что в таблице нет неопределенных значений, но для того, чтобы убедиться в этом применим комбинацию методов - isnull() и sum(). Метод isnull() вернет таблицу у которой этот метод вызван, только теперь все пустые ячейки (значение в которых равно null) будут иметь значение False, а непустые значение True. Так как в языке Python значение True при суммировании эквивалентно единице, False нулю, то после суммирования по каждой переменной, получим количество пустых ячеек в столбцах. По результату можем сделать вывод, что пустых ячеек в таблице нет.

Также при первичном ознакомлении с базой данных можно заметить что значение Insulin и SkinThicknes принимают значение 0, чего быть не может, значит значения просто пропущены. Построим гистограммы частот для каждой переменной, так как графиков много, чтобы они не наслаивались друг на друга и были легко читаемы в методе hist() передадим параметр figsize=(9,9), чтобы увеличить высоту и ширину основной фигуры в которой будут находится десять графиков. Действительно, по гистограммам частот параметров Insulin и SkinThicknes видим, что имеется достаточно большое количество нулевых значений. Попробуем посчитать это количество для каждого столбца, исключая параметры в которых нулевые значения теоретически могут присутствовать: Pregnancies (количество беременностей), Age (возраст) и Outcome (значения в котором в принципе могут быть равны либо ноль, либо один). В результате получаем, что у всех параметров, кроме DiabetesPedigreeFunction, имеются пропущенные значения, попробуем их заменить. По гистограммам частот можно сделать вывод, что параметры Glucose, BloodPressure и BMI имеют распределения похожие на нормальное, а остальные параметры, кроме Outcome, распределения похожие на параметрическое, значит заменять пропущенные значения на среднее было бы не очень корректно, так как среднее у параметрического распределение сильно смещено от максимального значения, а нам бы хотелось заменить пустые значения на такие, которые имеются у большинства, но при этом опять же из-за того, что некоторые параметры распределены экспоненциально, заменять пропущенные значения на моду, было бы так-же не совсем корректно, поэтому заменим их на медиану. Такая замена, как мне кажется не окажет критического влияния на обучение модели, и не вызовет переобучение на так называемых шумах, которыми по сути и будут являться пустые значения. Сделаем замечание, у параметра SkinThicknes и Insulin, количество пропущенных значений крайне велико относительно всех значений (30% для SkinThicknes и 49% для Insulin).



(Рис 1. Гистограммы распределения параметров)

**Подготовка данных**

Перед тем, как производить замену нулевых значений, сохраним индексы элементов с ненулевыми Insulin SkinThicknes. Теперь найдем медианные значения для каждого из параметров, который будем изменять. Важное замечание, при поиске медианы важно исключить пустые значения, так как они могут сильно сместить положение медианы в распределении. Так, например, для инсулина, если не исключать пропущенные значения, медиана будет равна 30.5, а если исключать, то уже 125. Теперь создадим список с «фичами» нашей базы данных и пройдемся по каждому столбцу, кроме Pregnancies, Age и Outcome (первого и двух последних), заменяем нулевые значения на значение медианы. Медиану будем искать на каждой итерации цикла с помощью маски, которая исключает нулевые значения, после чего применим метод median() и возьмем только текущий столбец. Проверим результат и увидим, что теперь пропущенных (нулевых) значений в таблице нет.

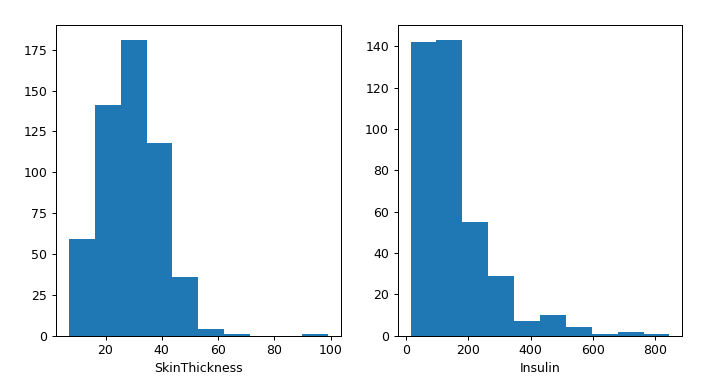
С помощью метода corr() посмотрим на значения корреляции между каждым из параметров. Некоторые параметры имеют достаточно большую корреляцию, например, значение корреляции между возрастом и количеством беременностей достаточно высока (0.544) или глюкозой и инсулином (0.419), значит не стоит использовать регрессионную модель, так как в такой модели нежелательно иметь высокую корреляцию между независимыми переменными. Также заметим, что корреляции SkinThicknes Outcome и Insulin Outcome, достаточно небольшие, а учитывая, что почти треть стобца SkinThicknes и половина столбца Insulin были пропущены, можно считать эти столбцы недостаточно значимыми, для обучения модели и просто не включать в обучающие данные.

Попробуем посмотреть корреляцию между SkinThicknes, Insulin и Outcome в изначальной базе данных, при этом исключая нулевые значения. Для этого воспользуемся двумя, сохраненными до этого списками, с элементами в которых не пропущены SkinThicknes и Insulin, Получаем значения корреляции между SkinThicknes и Outcome равное 0.303 и значения корреляции между Insulin и Outcome равное 0.258. Учитывая, что корреляция между SkinThicknes и Outcome и Insulin и Outcome, в таблице с заменой пропущенных значений соответственно равны 0.215 и 0.204. То есть добавив в данные 30% у SkinThicknes и почти 50% у Insulin медианных значений, мы достаточно сильно ухудшаем корреляцию.

Также учитывать специфику обучения модели методом решающих деревьев. Может получиться так, что пустые значения, которые мы заменили на медиану будут играть большое значение при обучении, так как дерево определяет некоторый барьер, и если значение меньше барьера, то оно будет определено в один лист, а если больше, то в другой. Таким образом мы рискуем определить значительное количество элементов в неправильный лист (так как изначально, значение было пропущено). Поэтому мне кажется, из базы данных стоит вычленить некоторых валидационный датасет, на котором можно проверить будут ли лучше метрики, если обучать модель на данных с заменой пропущенных значений, с заменой пропущенных значений и удалением параметров SkinThicknes и Insulin и с использованием базы данных с исключением пропущенных значений в этих колоннах (только в этих, потому что в остальных количество пропущенных значений мало).

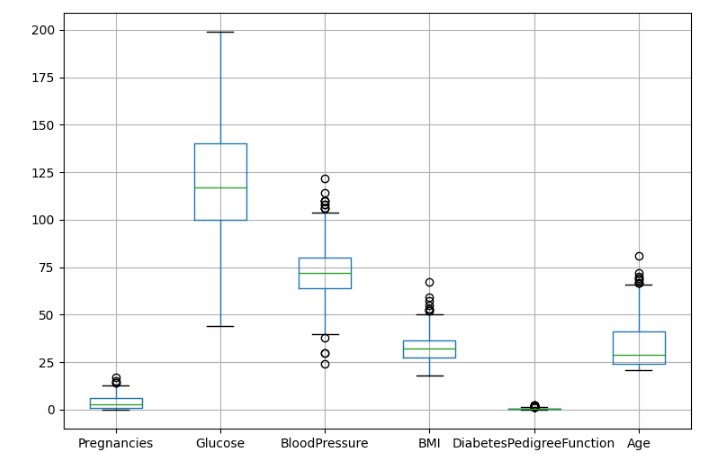
Помимо пропущенных значений в данных могут присутствовать так называемые выбросы. Если взять расстояние между первых и третьим квартилем выборки и отложить от этих квартилей отрезки равные этому расстоянию, умноженному на полтора, то все значения, которые не входят такой интервал называются выбросами, которые стоит исключать при обучении, так как модель может переобучиться на этих выбросах. При замене пустых значений на медианные мы изменяем саму первоначальную функцию распределения, изменяя стандартное отклонение и в некоторых случаях среднее значение. Но так-как во всех параметрах кроме толщины кожи и инсулина количество пропущенных значений мало относительно всех значений, пустые измененные значения можно не отбрасывать при поисках выбросов.

Посмотрим на графики гистограмм частот, и заметим, что скорее всего в каждом параметре имеются выбросы. Также для параметра Insulin и SkinThickness, стоит перестроить гистограммы без учета изменённых значения, которые первоначально были нулевыми. Используем массивы индексов исключающие пустые значение, которые мы сохранили до этого и построим гистограммы для двух параметров (рис. 2). Заметим, что данные распределения тоже имеют выбросы.



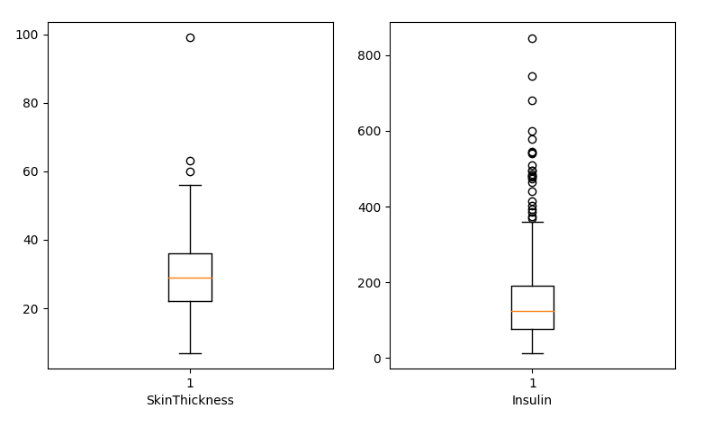
(Рис.2 Графики гистограмм частот без учета пустых значений)

Для того, чтобы точно убедиться в наличии выбросов построим диаграммы box plot. Причем для всех параметров, кроме SkinThickness и Insulin будем учитывать все значения, включая и измененные, так как их количество мало по сравнению с общим количеством значений.



(Рис.3 Графики box-plot для параметров в которых учитываем измененные значения)

Заметим, то практически во всех параметрах имеются выбросы. Теперь воспользуемся массивами индексов непустых изначально значений и построим графики box-plot отдельно для параметров SkinThickness и Insulin (рис.4).



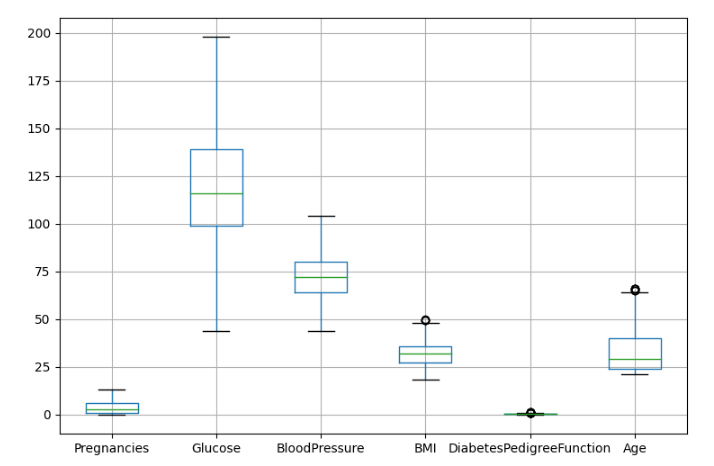
(Рис.4 Графики box-plot для параметров в которых не учитываем измененные значения)

В этих параметрах также имеются выбросы. Теперь попробуем исключить все строки таблицы, в параметрах которых есть выбросы. Так мы рискуем исключить большое количество входных данных, и если по итогу мы исключим большую часть таблицы, то потом вернемся к этому этапу и будем изменять значение выбросов на медианное значение.

Для начала создадим функцию, которая будет искать минимальное и максимальное значение границы значений, и если значения не будут входить в эту границу, то мы будем исключать их из таблицы. В функцию в качестве аргумента будет передаваться массив данных, у которого сначала найдем первый и третий квартили, после чего искать их разницу, умножать на 1.5 и отнимать это значение от первого квартиля и прибавлять ко второму, таким образом получим два значения – границы допустимых значений. На графике box-plot они обозначены усами. Заметим, что в некоторых случаях усы имеют разную длину, это происходит потому что параметр ограничен, например, в случае с инсулином, нижняя граница должна была быть отрицательна, но сам параметр ограничен значением близким к нулю снизу. Мы не будем вводить такие ограничения, так-как они никак не повлияют на наличие выбросов (если параметр имеет, например, минимальное значение до нижнего конца уса, то сам ус не продолжается, а останавливается на этом минимальном значении).

Теперь найдем границы допустимых значений для каждого из параметров кроме SkinThickness и Insulin. Для этого вызовем у таблице метод apply() и передадим функцию поиска границ, таким образом функция применится к каждой колонке и мы получим таблицу минимальных и максимальных допустимых значений.

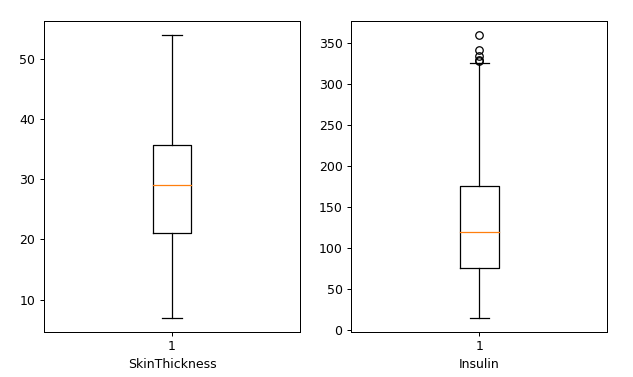
Для того, чтобы исключить выбросы воспользуемся маской размерности равной количеству строк в таблице, по итогу, например, если значение маски с индексом два будет ложью, значит строку с индексом два в таблице нужно исключить. Пройдемся в цикле по всем параметрам и на каждой итерации будем по каждому параметру создавать маску с истинным значением в том случае, если значение ячейки находится в пределах соответственного интервала. После применим логическое умножение к двум этим маскам. Таким образом в конце значение True будет лишь у тех строк в которых ни один параметр не имел выброса и просто перезапишем нашу базу данных. Построим графики box-plot для новой таблицы (рис. 5) и заметим, что действительно удалось избавиться от выбросов, присутствовавших на предыдущем графике, но при этом на границах появились новые выбросы. Так произошло потому что новый график строился без учета тех значений, которые мы исключили и соответственно значения, находившиеся вблизи границы на предыдущем графике теперь отображаются как выбросы. Но такие значения выбросами не считаются, потому что входят в интервал допустимых значений до исключения выбросов.



(Рис. 5 Графики box-plot, после того как исключили выбросы)

Важное замечание, теперь в таблице стало меньше строк и стоит обновить массивы с индексами непустых ячеек инсулина и толщины кожи, потому что некоторые из индексов, в которых значения были непустые могли быть потеряны при исключении выбросов. Для этого приведем списки с индексами к множеству set – мы можем так сделать, так как каждый элемент в массивах являлся уникальным индексом, после чего приведем оставшиеся индексы базы данных к множеству set и проведем логическое умножение между этими списками, таким списками. В результате получим два новых массива с индексами непустых значений, в которых также будут исключены те элементы, от которых мы отказались при исключении выбросов.

Теперь аналогично избавимся от выбросов для двух оставшихся параметров, при этом исключив пропущенные значения. В этот раз рассчитаем границы допустимых значений отдельно для каждого из двух параметров, так как индексы не пустых значений у параметров различны, после чего с помощью маски определим какие из строк в базе данных стоит исключить. Перезапишем базу данных и снова обновим списки с индексами непустых значений. Построим графики и посмотрим удалось ли избавится от выбросов (рис.6). По графику видно, что произошла аналогичная ситуация с изменением границ допустимых значений и появления из-за этого новых выбросов, но если обратиться к предыдущему графику, можно заметить, что интервал допустимых значений выходит за рамки значения 350 сверху, и значит те выбросы, что образовались на новом графике входят в область допустимых значений.



(Рис. 6 Графики box-plot, после того как исключили выбросы для SkinThickness и Insulin)

­

**Обучение модели**

Обучение модели будем производить с помощью алгоритма Random Forest, при этом попробуем применить при обучении три различных тренировочных базы данных: с заменой пропущенных значений на медиану, с заменой пропущенных значений на медиану и удалением столбцов SkinThickness и Insulin, а также базу данных не включающую пропущенные значения SkinThickness и Insulin. Для каждого способа обучения будем использовать единый набор параметров, подбираемых с помощью кросс-валидации. Также сам процесс обучения будет одинаковым для каждого набора входных значений, поэтому для удобства создадим функцию, на вход которой пойдет два аргумента: первый – таблица из независимых переменных, а второй – серия результатов для каждого набора переменных (Outcome). Внутри функции будет создаваться классификатор Random Forest, после чего будем обучать модель на переданных данных и с помощью RandomSearchCV найдем оптимальные параметры, после чего функция вернет наилучший обученный классификатор.

Метрики во всех трех случаях также будут рассчитываться идентично, поэтому создадим функцию, для расчета значений precision, recall и f1. Заранее проинициализируем необходимые библиотеки. На вход функции будет подаваться массив с предсказаниями модели и массив с истинными значениями. После чего функция рассчитает и напечатает precision, recall и f1 для данных параметров.

Сначала попробуем обучить модель на таблице с заменой пропущенных значений. Разделим нашу таблицу на 3 части: данные на которых модель будет обучаться, валидационные данные и тестовые данные. Пусть тестовые данные будут состоять из 25% базы данных, остальные определим, как тренировочные, после чего пусть 25% тренировочных данных будут являться валидационными. Так как такое разбиение сложнее, чем то, что может дать модуль skleaern train\_test\_split, то разобьем данные самостоятельно. Для этого инициализируем массив с индексами всей таблицы (числами от 0 до 768), после чего перемешаем его с помощью функции np.random.shuffle, и первые 75% данных запишем в массив train\_indexes, а остальные в test\_indexes. Аналогично разобьем train\_indexes на train\_indexes и validation\_indexes. Можно проверить не потеряли ли мы в процессе разбиения индексы, для этого просуммируем длинны всех трех массивов и получим, что их суммарная длина равна длине первоначального массива (768).

Запишем в таблицу X все параметры из таблицы с заменой пустых значений, кроме результата Outcome, а в таблицу (серию) y только результаты.