Оглавление

[Задачи 3](#_Toc104960740)

[Задача 1 3](#_Toc104960741)

[Задача 2 5](#_Toc104960742)

[Задача 3 10](#_Toc104960743)

[Задача 4 15](#_Toc104960744)

[Задача 5 17](#_Toc104960745)

[Заключение 17](#_Toc104960746)

[Список литературы 19](#_Toc104960747)

[**Приложения** 20](#_Toc104960748)

[Приложение 1 20](#_Toc104960749)

[Приложение 2 24](#_Toc104960750)

[Приложение 3 27](#_Toc104960751)

[Приложение 4 34](#_Toc104960752)

[Приложение 5 37](#_Toc104960753)

# Задачи

## Задача 1

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: Swiss

Объясняемая переменная: Examination

Регрессоры: Infant.Mortality, Education

1. Оцените среднее значение, дисперсию и СКО объясняемой переменной и регрессоров.

Среднее значение столбца переменной Examination примерно равно 17, то есть только 17% призывников получало наивысшую оценку на армейском экзамене. Довольно неплохое значение, но все равно низкое.

У регрессора Infant.Mortality среднее значение столбца почти равно 20, то есть в среднем каждый пятый ребёнок умирал до года, что говорит о крайне низком уровне медицины.

Среднее значение столбца регрессора Education равно почти 11. 11% призывников получали образование выше начального уровня. Это говорит о низком уровне образованности призывников.

Для оценки дисперсии и СКО переменных сначала произведём нормализацию и предобработку данных, а уже потом построим графики, на которых будут видны значения переменных и степень их отклонения от среднего.

Проанализируем такой график для переменной Examination (Рисунок 1.1). В среднем процент призывников, сдавших экзамен на отлично, примерно равен (меньше одного стандартного отклонения), что говорит о одинаковом уровне образования. Есть провинции с отклонением больше одного стандартного, но большинство в отрицательную сторону, то есть меньший процент сдавших на отлично.

Теперь для регрессора Infant.Mortality (Рисунок 1.2). Там все значения, кроме одного, не превышают двух стандартных отклонений (одно значение отличается от среднего на три отклонения). Это говорит о схожести провинций в уровне медицины.

Далее регрессор Education (Рисунок 1.3). Процент призывников, получивших образование выше начального, в большинстве провинций ниже среднего, но в пределах одного отклонения. Есть провинции, значение которых отличаются больше двух отклонений, и одна провинция с высоким процентом образования (отклонение выше 4 стандартных).

1. Постройте зависимости вида y = a + bx, где y – объясняемая переменная, x – регрессор (для каждого варианта по две зависимости).
2. Оцените, насколько «хороша» модель по коэффициенту детерминации R2?
3. Оцените, есть ли взаимосвязь между объясняемой переменной и объясняющей переменной (по значению p-статистики, «количеству звездочек» у регрессора в модели).

Для начала построим зависимость переменной Examination от Education.

Коэффициент детерминации R2 данной модели равен 47.64%, это модель предсказывает примерно половину значений объясняемой переменной. P-статистика равна 4.81e-08, "звёздочек" в модели у регрессора три, а значит взаимосвязь между Examination и Education есть, что довольно логично. Значения коэффициентов модели, их стандартные ошибки, p-статистика и уровень значимости приведены в таблице 1.1.

Таблица 1.1. Характеристики модели зависимости Examination от регрессора Education в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | 10.12748 | 1.28589 | 7.876 | 5.23e-10 | \*\*\* |
| Education | 0.57947 | 0.08852 | 6.546 | 4.81e-08 | \*\*\* |

Теперь проанализируем то же самое для зависимости переменной Examination от Infant.Mortality.

Коэффициент детерминации R2 равен 0.8932%, это значит, что данная модель абсолютно не объясняет зависимости данных в наборе Swiss. P-статистика равна 0.45, "звёздочек" в модели у регрессора нет, а значит и взаимосвязи между Examination и Infant.Mortality практически нет. Значения коэффициентов модели, их стандартные ошибки, p-статистика и уровень значимости приведены в таблице 1.2.

Таблица 1.2. Характеристики модели зависимости Examination от регрессора Infant.Mortality в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | 22.7175 | 8.1736 | 2.779 | 0.00792 | \*\* |
| Infant.Mortality | -0.3123 | 0.4056 | -0.770 | 0.44538 |  |

Код решения задачи и сведения о проверенных моделях приведены в Приложении 1.

**Вывод:** были построены модели зависимости отличной аттестации призывников (Examination), от уровня обучения (Education) и смертности до года (Infant.Mortality) в провинциях Франции по данным 1888 года из набора Swiss. По итогу проведенной работы с помощью анализа P-статистики и стандартной ошибки коэффициентов перед регрессорами было установлено, что между Examination и Education есть сильная зависимость, но коэффициент детерминации R2 данной модели равен 47.64%, что указывает на низкую предсказательную способность модели.

## Задача 2

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: mtcars

Объясняемая переменная: mpg

Регрессоры: wt, qsec, hp, drat

1. Проверьте, что в наборе данных нет линейной зависимости (построить зависимости между переменными, указанными в варианте, и проверить, что R2 в каждой из них невысокий). В случае, если R2 большой, один из таких столбцов можно исключить из рассмотрения.

Проверим линейную регрессию wt ~ qsec + hp + drat: в этой зависимости R2 = 69.1%, то есть существует достаточно сильная зависимость между регрессорами, скорее всего, придется убрать один из регрессоров, но пока переменную wt оставим и попробуем использовать ее в последующих регрессиях.

Далее построим линейную зависимость qsec ~ wt + hp + drat, где R2 = 61.51%, это меньше, чем в предыдущей модели, но все равно наблюдается сильная зависимость, такой зависимостью вряд ли можно пренебречь, рассмотрим с параметром qsec следующие модели и тогда сделаем окончательные выводы.

Конечно, проверим третью модель hp ~ wt + qsec + drat. R2 регрессии равен 77.51%, это очень высокий показатель, поэтому можно сделать вывод, что переменная hp линейно зависима от регрессоров этой модели. Значит, параметр hp стоит убрать и больше не использовать в построении математических моделей.

Также проверим линейную регрессию drat ~ wt + qsec + hp, в этой зависимости R2 = 45.61%, то есть наблюдается средняя зависимость, по сравнению с предыдущими значениями R2 такой зависимостью можно пренебречь, поскольку уже было принято решение убрать регрессор hp. Тогда переменную drat можно использовать в последующих регрессиях.

В конечном итоге я пришел к вводу о том, что регрессор hp очень связан с остальными, поэтому его стоит исключить из рассмотрения последующих моделей. Остальные же регрессоры можно будет использовать с остальными для построения моделей линейных регрессий, но нужно будет внимательно следить за ними, поскольку прослеживается довольно значительная зависимость.

1. Постройте линейную модель зависимой переменной от указанных в варианте регрессоров по методу наименьших квадратов. Оценить, насколько хороша модель, согласно: 1) R2, 2) p-значениям каждого коэффициента.

Построим модель mpg ~ wt + qsec + drat. Значения коэффициентов данной модели, их стандартные ошибки, p-статистика и уровень значимости приведены в таблице 2.1.

Таблица 2.1. Характеристики модели зависимости mpg от регрессоров wt, qsec, drat в наборе данных mtcars.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | 11.3945 | 8.0689 | 1.412 | 0.16892 |  |
| wt | -4.3978 | 0.6781 | -6.485 | 5.01e-07 | \*\*\* |
| qsec | 0.9462 | 0.2616 | 3.616 | 0.00116 | \*\* |
| drat | 1.6561 | 1.2269 | 1.350 | 0.18789 |  |

Оценим модель: коэффициент детерминации R2 = 81.96%, это очень высокий показатель, значит, модель очень хороша и объясняет данные в наборе mtcars. P-значение регрессора wt очень маленькое (5.01е-07), и у регрессоров qsec и drat тоже маленькие, но уже больше 0.001 и 0.19 соответственно. У wt 3 звёздочки, у qsec 2 звёздочки, а у регрессора drat их нет. Также стоит отметить, что у всех регрессоров достаточно велика стандартная ошибка, а особенно у регрессора drat, она равна 1.23, поэтому, скорее всего, именно drat наиболее незначимый в модели параметр. VIF у каждого регрессора находится в пределах от 1.03 до 2.08, что говорит о независимости регрессоров.

Заключение: данная математическая модель достаточно хорошая, в ней сконцентрированы нужные и важные регрессоры, но возможно еще не все регрессоры стоит использовать, так, например, при удалении регрессора drat из модели R2 падает всего на 0.5%. Дальше попробуем найти не очень нужные регрессоры, которые можно было бы исключить без большого вреда для R2, и тем самым улучшить нашу модель.

1. Введите в модель логарифмы регрессоров (если возможно). Сравнить модели и выбрать наилучшую.

Для решения данного пункта задания я построил модели с использованием регрессоров: ln(wt), ln(qsec), ln(drat). При анализе построенных моделей я заметил, что добавление в модель логарифма регрессора, где уже есть сам регрессор ведёт к сильному возрастанию vif у пары этих данных. Так же мной было замечено, что регрессор wt уменьшает R2 в любой модели. Поэтому, после тестов всех комбинаций стало ясно, что ln(wt) стоит внести в модель с логарифмом, а сам wt стоит исключить из модели.

Итог: самой лучшей моделью среди моделей с добавлением натуральных логарифмов от регрессоров является эта mpg ~ qsec + drat + ln(wt). Значения коэффициентов модели, их стандартные ошибки, p-статистика и уровень значимости приведены в таблице 2.2.

Таблица 2.2. Характеристики модели зависимости mpg от регрессоров qsec, drat, ln(wt) в наборе данных mtcars.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | 18.0083 | 7.2144 | 2.496 | 0.018715 | \* |
| qsec | 0.9035 | 0.2245 | 4.025 | 0.000393 | \*\*\* |
| drat | 0.8219 | 1.0818 | 0.760 | 0.453743 |  |
| ln(wt) | -15.1557 | 1.8445 | -8.217 | 6.07e-09 | \*\*\* |

1. Введите в модель всевозможные произведения пар регрессоров, в том числе квадраты регрессоров. Найдите одну или несколько наилучших моделей по доле объяснённого разброса в данных R2.

Вначале добавим к регрессорам первоначальной линейной модели всевозможные комбинации с произведениями пар данных регрессоров. Добавлять будем регрессоры: I(wt^2), I(qsec^2), I(drat ^2), I(wt\*qsec), I(wt\*drat), I(drat\*qsec).

Дальше по-отдельности рассмотрим наилучшие модели добавлением квадратов регрессоров и с добавлением произведений пар регрессоров. В первом случае оказалось, что регрессор drat уменьшает R2 в любой модели, а растёт R2 лучше всего при добавлении I(drat ^2). Лучшей моделью с R2 = 82.02% оказалась линейная регрессия с исключением drat и добавлением I(drat^2). Все её характеристики параметров приведены в таблице 2.3.

Таблица 2.3. Характеристики модели зависимости mpg от регрессоров wt, qsec, I(drat^2) в наборе данных mtcars.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | 14.3106 | 6.4819 | 2.208 | 0.03562 | \* |
| wt | -4.3845 | 0.6744 | -6.502 | 4.8e-07 | \*\*\* |
| qsec | 0.9436 | 0.2611 | 3.614 | 0.00117 | \*\* |
| I(drat^2) | 0.2304 | 0.1657 | 1.390 | 0.17546 |  |

Во втором случае оказалось, что регрессор wt уменьшает R2 в любой модели, а растёт R2 лучше всего при добавлении регрессора I(drat\*wt). Лучшей моделью с R2 = 84.46% оказалась линейная регрессия с исключением wt и добавлением регрессора I(drat\*wt). Все её характеристики параметров приведены в таблице 2.4.

Таблица 2.4. Характеристики модели зависимости mpg от регрессоров qsec, drat, I(drat\*wt) в наборе данных mtcars.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | 0.3507 | 6.2803 | 0.056 | 0.95586 |  |
| qsec | 0.8944 | 0.2437 | 3.670 | 0.00101 | \*\* |
| drat | 5.1880 | 0.8523 | 6.087 | 1.45e-06 | \*\*\* |
| I(drat\*wt) | -1.3276 | 0.1817 | -7.305 | 5.93e-08 | \*\*\* |

И теперь можем попытаться сравнить первоначальную модель с двумя полученными, первая из которых – лучшая модель с логарифмом, а вторая это лучшая модель с добавлением произведения регрессоров, поскольку модель с добавлением квадратов уже заведомо оказалось хуже (у нее R2 меньше).

Сравним первоначальную модель mpg ~ wt + qsec + drat, где R2 = 81.96% с лучшей моделью добавлением произведения регрессоров mpg ~ qsec + drat + I(drat\*wt), где R2 = 84.96%.

Наилучшая модель выявляется по значению R2, тогда это будет модель с добавлением произведения регрессоров mpg ~ qsec + drat + I(drat\*wt). VIF у регрессоров данной модели не превышают 1.18, а значит, серьезной зависимости между регрессорами нет. Значения коэффициентов данной модели, их стандартные ошибки, p-статистика и уровень значимости приведены в таблице 2.5.

Далее выполним задание 5 и 6 для этой модели.

1. Найти доверительные интервалы для всех коэффициентов в наилучшей модели, p = 95%. Сделать вывод о отвержении или невозможности отвергнуть статистическую гипотезу о том, что коэффициент равен 0.

Найдём значение t, необходимое для определения доверительных интервалов. С помощью функции qt (0.975, df = 28) получим, что t = 2.048407, округлим до двух знаков после запятой, t = 2.05.

Доверительный интервал для коэффициента регрессора *qsec* [0.89-2.05\*0.24;0.89+2.05\*0.24] = [0.4;1.38].

Значение "0" не попадает в доверительный интервал коэффициента перед регрессором, а значит, переменная *mpg* связанаспеременной *qsec.*

Доверительный интервал для коэффициента регрессора *drat* [5.19-2.05\*0.85;5.19+2.05\*0.85] = [3.45;6.93].

Значение "0" не входит в доверительный интервал коэффициента перед регрессором, а значит, *mpg* зависит от *drat.*

Доверительный интервал для коэффициента регрессора *I(drat \* wt)* [-1.33-2.05\*0.18;-1.33+2.05\*0.18] = [-1.7;-0.96].

Значение "0" не входит в доверительный интервал коэффициента перед регрессором, а значит, *mpg* зависит *от I(drat \* wt).*

1. Доверительный интервал для одного прогноза (p = 95%, набор значений регрессоров выбираем сами).

Найдём доверительный интервал для прогноза, в котором переменные будут иметь значения: *qsec* = 17.5, *drat* = 4, *wt* = 3.4. С помощью функции predict получим верхнюю и нижнюю границы доверительного интервала, а также его среднее прогнозируемое значение.

Нижняя граница (lwr в результате функции predict) равна 17.13805, верхняя граница (upr в результате функции predict) равна 20.25923, а среднее прогнозируемое значение (fit в результате функции predict) равно 18.69864.

Код решения задачи и сведения о проверенных моделях приведены в Приложении 2.

***Вывод:*** я проверил данные мне в задание регрессоры *wt, qsec, hp, drat* на линейную зависимость. В результате чего была обнаружена сильная линейная зависимость, а значит, нельзя было использовать все регрессоры вместе. Опираясь на значение R2 , было принято решение исключить параметр *hp* из математических моделей, поскольку R2 = 77.51%. Далее я проверил возможность использования остальных регрессоров вместе: R2 значительно снизился, но все же оставалась средняя зависимость, поэтому важно было следить за оставшимися регрессорами. Затем я построил линейную регрессию с помощью этих переменных и оценил её по величине R2 и по характеристикам коэффициентов перед регрессорами. Математическая модель *mpg ~ wt + qsec + drat,* гдеR2 = 81.96%, оказалась очень хорошей, и она хорошо объясняет данные в наборе mtcars. Несмотря на этот высокий показатель, я попытался улучшить ее с помощью введения натуральных логарифмов от регрессоров и их попарных произведений. Таким образом, проанализировав более 25 линейных регрессий, я смог выявить наилучшую модель с имеющимся набором регрессоров. Наилучшей моделью стала линейная регрессия с добавлением произведения регрессоров *mpg ~ qsec + drat + I(drat\*wt)*, где R2 = 84.96%, что также выше предыдущего значения, а также VIF у регрессоров данной модели не превышают 1.18, а значит, серьезной зависимости между регрессорами нет. В целом модель является очень хорошей, поскольку R2 > 80%, то зависимость определенно есть. С помощью наилучшей модели мне удалось найти доверительные интервалы для всех коэффициентов регрессии *mpg ~ qsec + drat + I(drat\*wt)* (p = 95%) и сделать вывод о том, может ли коэффициент равен нулю или нет. Получилось, что значение «0» не попадает ни в один из доверительных интервалов, поэтому переменная *mpg* зависит от всех 3 регрессоров (*qsec, drat, I(drat\*wt)).* Я рассчитал доверительный интервал для прогноза, выбрав при этом значения регрессоров равными: *qsec* = 17.5, *drat* = 4, *wt* = 3.4, тогда доверительный интервал получился [20.25923;18.69864].

## Задача 3

Номер волны выборки РМЭЗ: 15

Подмножества для пункта 5: Городские жители, не состоящие в браке; разведенные

женщины, с высшим образованием

Необходимо загрузить данные из указанного набора и произвести следующие действия.

1. Постройте линейную регрессию зарплаты на все параметры, которые Вы выделили из данных мониторинга. Не забудьте оценить коэффициент вздутия дисперсии VIF.

Для начала выберу набор параметров, который будет необходим, чтобы описать социально-экономическое положение граждан Российской Федерации. Столбца параметров, которые я выбрала: зарплата, пол, семейное положение, наличие высшего образование, возраст, населённый пункт, удовлетворенность условиями труда, продолжительность рабочей недели, опасность производства, причастность государства к владению предприятия и наличие второй работы. В предоставленных данных опроса это столбцы kj13.2, kh5, k\_marst, k\_educ, k\_age, status, kj1.1.2, kj6.2, kj21.3, kj23 и kj32.

Теперь преобразуем выбранные мной параметры по следующему принципу:

* Факторные переменные, «имеющие много значений», такие как: зарплата, длительность рабочей недели и возраст, - преобразуем в вещественные переменные и нормализуем их: вычтем среднее значение по этой переменной, разделим её значения на стандартное отклонение.
* Из параметра, отвечающего типу населённого пункта, создадим одну дамми-переменную city\_status со значением 1 для города или областного центра, 0 – в противоположном случае.
* Из параметра, отвечающего семейному положению, сделаем дамми-переменные 1) переменная wed1 имеет значение 1 в случае, если респондент женат, 0 – в противном случае; 2) wed2=1, если респондент разведён или вдовец; 3) wed3 = 1, если респондент никогда не состоял в браке.
* Из параметра, отвечающего за пол сделаем переменную sex, имеющую значение 1 для мужчин и равную 0 для женщин.
* Для каждого из остальных параметров сделаем отдельные дамми-переменные, которые будут иметь значение 1 для случая, когда выполняется условие в заданном вопросе и значение 0 для случая, когда ответом на вопрос будет «нет».

В итоге получим data2 с новыми переменными, с которыми уже можно работать и построить линейную зависимость зарплаты от остальных переменных: salary ~ sex, wed1, wed2, wed3, higher\_education, age, city\_status, satisfy, duration, dangerous, government, second\_job. R2 модели равен ~21%. Три "звёздочки" у регрессоров sex, higher\_education, age, city\_status, satisfy, duration, dangerous, government p-статистика этих регрессоров мала, это хороший показатель. 0-1 "звёздочки" у регрессоров wed1, wed2, wed3, second\_job, значит, что они мало чего объясняют. Оценив vif, сделал вывод, что wed1, wed2, wed3 скорее всего зависимы, попробую убрать wed3 из модели

Проверим модель salary ~ sex, wed1, wed2, higher\_education, age, city\_status, satisfy, duration, dangerous, government, second\_job. R2 модели остался равен ~21%, однако р-статистики wed1 и wed2 снизились. Регрессор second\_job плохо объясняет salary. принято решение его исключить.

Получим первоначальную рассматриваемую математическую модель salary ~ sex, wed1, wed2, higher\_education, age, city\_status, satisfy, duration, dangerous, government. R2 модели равен ~21%. Большинство регрессоров хорошо объясняют нашу переменную salary. Показатели vif в пределах 2, значит наши регрессоры не коррелируют между собой. Детально с характеристиками получившейся модели можно ознакомиться в таблице 3.1.

Таблица 3.1. Характеристики модели зависимости salary ~ sex, wed1, wed2, higher\_education, age, city\_status, satisfy, duration, dangerous, government в наборе данных 15-ой волны исследования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | -0.70287 | 0.05045 | -13.932 | < 2e-16 | \*\*\* |
| sex | 0.39062 | 0.03186 | 12.260 | < 2e-16 | \*\*\* |
| wed1 | 0.14435 | 0.03739 | 3.860 | 0.000115 | \*\*\* |
| wed2 | 0.14560 | 0.05167 | 2.818 | 0.004861 | \*\* |
| higher\_education | 0.53227 | 0.03412 | 15.599 | < 2e-16 | \*\*\* |
| age | -0.07836 | 0.01645 | -4.764 | 1.97e-06 | \*\*\* |
| city\_status | 0.31184 | 0.03455 | 9.026 | < 2e-16 | \*\*\* |
| satisfy | 0.33979 | 0.03001 | 11.323 | < 2e-16 | \*\*\* |
| duration | 0.12146 | 0.01545 | 7.859 | 5.05e-15 | \*\*\* |
| dangerous | 0.18252 | 0.03926 | 4.649 | 3.45e-06 | \*\*\* |
| government | -0.28739 | 0.03146 | -9.136 | < 2e-16 | \*\*\* |

1. Поэкспериментируйте с функциями вещественных параметров: используйте логарифм и степени (хотя бы от 0.1 до 2 с шагом 0.1).

Функциями вещественных параметров, кроме зарплаты, в моём случае являются возраст и продолжительность рабочей недели, поэкспериментируем с ними. Для начала будем добавлять в первоначальную линейную регрессию натуральные логарифмы от этих переменных, а именно ln(age) и ln(duration).

С логарифмами там могут быть только три различные вариации: в первый раз добавляем только ln(age), во вторую модель добавляем ln(duration), а в третью оба логарифма сразу.

После получения значения R2 и VIF для этих трёх зависимостей точно определяем, что лучшей из них будет первая модель salary ~ sex, wed1, wed2, higher\_education, age, city\_status, satisfy, duration, dangerous, government, second\_job, ln(age). Её коэффициент детерминации R2 равен 24.86%, а команда VIF показывает свои значения в пределах 3.5. Подробная информация о значении коэффициентов, стандартных ошибках и значимости использующихся регрессоров приведена в таблице 3.2.

Таблица 3.2. Характеристики модели зависимости salary ~ sex, wed1, wed2, higher\_education, age, city\_status, satisfy, duration, dangerous, government, ln(age) в наборе данных 15-ой волны исследования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | -0.36396 | 0.09250 | -3.935 | 8.64e-05 | \*\*\* |
| sex | 0.39642 | 0.04058 | 9.769 | < 2e-16 | \*\*\* |
| wed1 | 0.09060 | 0.05657 | 1.602 | 0.10940 |  |
| wed2 | 0.11040 | 0.06714 | 1.644 | 0.10024 |  |
| higher\_education | 0.53098 | 0.04255 | 12.479 | < 2e-16 | \*\*\* |
| age | -0.46854 | 0.06085 | -7.700 | 2.19e-14 | \*\*\* |
| city\_status | 0.33729 | 0.04198 | 8.035 | 1.64e-15 | \*\*\* |
| satisfy | 0.34596 | 0.03730 | 9.276 | < 2e-16 | \*\*\* |
| duration | 0.11277 | 0.02008 | 5.616 | 2.25e-08 | \*\*\* |
| dangerous | 0.21668 | 0.04854 | 4.464 | 8.54e-06 | \*\*\* |
| government | -0.19141 | 0.03927 | -4.875 | 1.18e-06 | \*\*\* |
| log(age) | 0.11065 | 0.03373 | 3.281 | 0.00106 | \*\* |

Поэкспериментируем со степенями функций вещественных переменных, то есть будем возводить регрессоры age и duration в разные степени и добавлять их к первоначальной модели порознь и вместе.

Сначала рассмотрим такие модели со степенями регрессоров age и duration от 0.1 до 1. Увеличиваем степени с шагом 0.1 и проверять величину R2 и VIF каждой из моделей. С повышением степеней вплоть до единицы R2 падает и VIF становился только хуже и уже к возведению в степень 0.4 VIF стал недопустимым для того, чтобы выполнялось условие независимости регрессоров модели. Перейдем к степени 1.1. R2 продолжает медленно падать, при этом показатели VIF сильно выше нормы. Проанализировав характеристики последующих моделей, сделал вывод, что среди степеней лучшая модель первая, зависимость salary ~ sex, wed1, wed2, higher\_education, age, city\_status, satisfy, duration, dangerous, government, I(age0.1). Подробные характеристики данной математической модели приведены в таблице 3.3.

Таблица 3.3. Характеристики модели зависимости salary ~ sex, wed1, wed2, higher\_education, age, city\_status, satisfy, duration, dangerous, government, I(age0.1) в наборе данных 15-ой волны исследования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | -1.71883 | 0.36962 | -4.650 | 3.55e-06 | \*\*\* |
| sex | 0.39656 | 0.04059 | 9.771 | < 2e-16 | \*\*\* |
| wed1 | 0.09035 | 0.05658 | 1.597 | 0.11044 |  |
| wed2 | 0.10987 | 0.06714 | 1.636 | 0.10192 |  |
| higher\_education | 0.53097 | 0.04256 | 12.477 | < 2e-16 | \*\*\* |
| age | -0.49473 | 0.06863 | -7.209 | 8.17e-13 | \*\*\* |
| city\_status | 0.33755 | 0.04198 | 8.040 | 1.58e-15 | \*\*\* |
| satisfy | 0.34592 | 0.03730 | 9.273 | < 2e-16 | \*\*\* |
| duration | 0.11264 | 0.02008 | 5.609 | 2.34e-08 | \*\*\* |
| dangerous | 0.21670 | 0.04855 | 4.464 | 8.54e-06 | \*\*\* |
| government | -0.19114 | 0.03927 | -4.867 | 1.23e-06 | \*\*\* |
| I(age^0.1) | 1.38282 | 0.42971 | 3.218 | 0.00131 | \*\* |

1. Выделите наилучшие модели из построенных: по значимости параметров, включённых в зависимости, и по объяснённому с помощью построенных зависимостей разбросу adjusted R2 - R2 adj.

Сравним двух претендентов на лучшую модель из всех рассмотренных. Первая модель - это лучшая модель среди регрессий с введением логарифмов переменных (её характеристики в таблице 3.2), а вторая это лучшая модель среди всех регрессий с введением степеней переменных, а именно модель, где возведение идёт в степень 0.1 (её характеристики в таблице 3.3).

У первой модели R2 = 24.86%, значимость регрессоров довольно неплохая. У второй модели R2 = 24.8%, значимость регрессоров точно такая же. Разницы между коэффициентами детерминации почти нет. Посмотрим на VIF у этих двух регрессий. У модели с логарифмами максимальное значение VIF ~ 3.5, в то время как у модели со степенью 0.1 VIF достигает почти 4.4.

Значит, я могу заключить, что лучшей моделью из всех рассмотренных является модель salary ~ sex, wed1, wed2, higher\_education, age, city\_status, satisfy, duration, dangerous, government, ln(age). На данных этой модели будем делать дальнейшие выводы.

1. Сделайте вывод о том, какие индивиды получают наибольшую зарплату.

Рассмотрим коэффициенты перед значимыми регрессорами в нашей лучшей модели, полученные с помощью команды summary и отображённые в таблице 3.2.

* sex: 0.4 - положительный
* higher\_education: 0.53 - положительный
* age: -0.47 - отрицательный
* city\_status: 0.34 - положительный
* satisfy: 0.35 - положительный
* duration: 0.11 - положительный
* dangerous: 0.22 - положительный
* government: -0.19 - отрицательный
* log(age): 0.11 – положительный

Вывод о том, какие индивиды получают большую зарплату: большую зарплату получают в большинстве своём мужчины, люди с высшим образованием (это самый важный показатель), молодые люди, также люди, проживающие в городе, индивиды, удовлетворённые своими условиями труда. Ещё большую зарплату получают люди, работающие на опасных или вредных производствах и люди, работающие в негосударственных компаниях, с большим количеством часов в неделю семейное положение не влияли на уровень заработной платы.

1. Оцените регрессии для подмножества индивидов, указанных в варианте.

Выделим подмножество городских жителей, не состоящих в браке с помощью применения функции subset дважды. Построим выбранную мной лучшую модель зависимости зарплаты от других параметров, но теперь учтём то, что мы находимся в подмножестве городских жителей, состоящих в браке (в этом подмножестве переменные city\_status равен единице, а wed1 равны нулю). salary ~ sex, higher\_education, age, satisfy, duration, dangerous, government, log(age). R2 = 21.29%, незначительны переменные dangerous, government, log(age); VIF нормален (до ~ 3.5). Значения коэффициентов модели, их стандартные ошибки, p-статистика и уровень значимости приведены в таблице 3.4.

Таблица 3.4. Характеристики модели зависимости salary ~ sex, higher\_education, age, satisfy, duration, dangerous, government, log(age) в наборе данных 15-ой волны исследования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | -0.19974 | 0.13342 | -1.497 | 0.135045 |  |
| sex | 0.29027 | 0.07970 | 3.642 | 0.000300 | \*\*\* |
| higher\_education | 0.46695 | 0.07659 | 6.096 | 2.23e-09 | \*\*\* |
| age | -0.31028 | 0.10160 | -3.054 | 0.002383 | \*\* |
| satisfy | 0.37960 | 0.06871 | 5.525 | 5.42e-08 | \*\*\* |
| duration | 0.13997 | 0.03680 | 3.804 | 0.000161 | \*\*\* |
| dangerous | 0.23960 | 0.09344 | 2.564 | 0.010645 | \* |
| government | -0.10303 | 0.07165 | -1.438 | 0.151082 |  |
| log(age) | -0.01155 | 0.05854 | -0.197 | 0.843685 |  |

Среди городских жителей в браке всё те же индивиды получают зарплату больше других.

Выделим подмножество разведенных женщин с высшим образованием с помощью применения функции subset трижды. Построим выбранную мной лучшую модель зависимости зарплаты от других параметров, учитывая, что мы находимся в подмножестве разведённых жителей без высшего образования (в этом подмножестве переменные wed2 и higher\_education равны единице, а переменная sex равна нулю). salary ~ age, city\_status, satisfy, duration, dangerous, government, log(age). R2 = 14.43% - очень маленький, все переменные, кроме government незначительны (у government 1 "звёздочка"), VIF нормален (до ~3.7). Значения коэффициентов модели, их стандартные ошибки, p-статистика и уровень значимости приведены в таблице 3.5.

Таблица 3.5. Характеристики модели зависимости salary ~ age, city\_status, satisfy, duration, dangerous, government, log(age) в наборе данных 15-ой волны исследования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | 0.14077 | 0.37186 | 0.379 | 0.7058 |  |
| age | 0.04397 | 0.24781 | 0.177 | 0.8595 |  |
| city\_status | 0.01569 | 0.22504 | 0.070 | 0.9445 |  |
| satisfy | 0.30430 | 0.17930 | 1.697 | 0.0928 | . |
| duration | 0.08388 | 0.08927 | 0.940 | 0.3497 |  |
| dangerous | -0.18032 | 0.23415 | -0.770 | 0.4431 |  |
| government | -0.45133 | 0.20866 | -2.163 | 0.0330 | \* |
| log(age) | -0.24565 | 0.14961 | -1.642 | 0.1038 |  |

Никой зависимости зарплаты от данных регрессоров для разведенных женщин с высшим образованием не установил.

Код решения задачи и сведения о проверенных моделях приведены в Приложении 3.

**Вывод:** я проанализировал данные опроса НИУ ВШЭ о материальном состоянии граждан России за 2004 год.

## Задача 4

Набор данных: Drug classification

Тип классификатора: LogisticRegression (логистическая регрессия)

Классификация по столбцу: Drug Type (DrugX – класс 0, остальные уровни – класс 1)

1. Обработайте набор данных, подготовив его к решению задачи классификации. Выделите целевой признак, указанный в последнем столбце таблицы, и удалите его из данных, на основе которых будет обучаться классификатор. Разделите набор данных на тестовую и обучающую выборку. Постройте классификатор типа, указанного в варианте, для задачи классификации по параметру, указанному также в варианте. Оцените точность построенного классификатора с помощью метрик precision, recall и F1 на тестовой выборке.

Сначала считаем набор данных с устройства, далее преобразуем его в таблицу и удалим все строки, в которых встречаются пустые значения. Первые 5 строк таблицы приведены в таблице 4.1.

Таблица 4.1. Первые 5 строк набора данных Drug classification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Age | Sex | BP | Cholesterol | Na\_to\_K | Drug |
| 23 | F | HIGH | HIGH | 25.355 | DrugY |
| 47 | M | LOW | HIGH | 13.093 | drugC |
| 47 | M | LOW | HIGH | 10.114 | drugC |
| 28 | F | NORMAL | HIGH | 7.798 | drugX |
| 61 | F | LOW | HIGH | 18.043 | DrugY |

Перейдём к подготовке решения задачи классификации. Все, не числовые признаки нужно преобразовать в числовые или, если можно, в бинарные:

Столбец Sex преобразуем по принципу: M – 1, F – 0;

Столбец BP преобразуем следующим образом: Low – 0, High – 1, Normal – 2;

Столбец Cholesterol преобразуем в бинарный: Normal – 0, High – 1.

А также обработаем столбец, по которому будем строить классификацию (Drug Type): DrugX – класс 0, остальные уровни – класс 1. Этот столбец удалим из данных, оставив его отдельно от остальных столбцов. Его нужно убрать, чтобы наш классификатор смог обучаться. Первые пять строк и последние 5 строк изменённой таблицы приведены в таблице 4.2.

Таблица 4.2. Таблица, подготовленная для решения задачи классификации.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Age | Sex | BP | Cholesterol | Na\_to\_K |
| 23 | 0 | 1 | 1 | 25.355 |
| 47 | 1 | 2 | 1 | 13.093 |
| 47 | 1 | 2 | 1 | 10.114 |
| 28 | 0 | 0 | 1 | 7.798 |
| 61 | 0 | 2 | 1 | 18.043 |
| ... | ... | ... | ... | ... |
| 56 | 0 | 2 | 1 | 11.567 |
| 16 | 1 | 2 | 1 | 12.006 |
| 52 | 1 | 0 | 1 | 9.894 |
| 23 | 1 | 0 | 0 | 14.020 |
| 40 | 0 | 2 | 0 | 11.349 |

Можем приступить к задаче классификации. Разделим набор данных на обучающую и тестовую выборку. Размер тестовой выборки будет 33%.

Построим классификатор LogisticRegression и проверим его на тестовых данных, чтобы оценить точность построенного классификатора с помощью метрик accuracy, precision, recall и F1 на данной выборке. Их полученные значения с помощью метода classification\_report приведены в таблице 4.3.

Таблица 4.3. Результат работы метода classification\_report на нашей тестовой выборке.

|  |  |  |  |
| --- | --- | --- | --- |
| precision | recall | f1-score | support |
| 0.84 | 0.85 | 0.84 | 66 |

1. Постройте классификатор типа Случайный Лес (Random Forest) для решения той же задачи классификации. Оцените его качество с помощью метрик precision, recall и F1 на тестовой выборке. Какой из классификаторов оказывается лучше?

Строим классификатор Random Forest, состоящий из 200-от, 300-от и 400-от деревьев, таким образом, найдем наилучшее количество деревьев с шагом 100. Исходя из результатов, наилучшим вариантом может считаться вариант с 400-ми деревьями, поскольку его метрики оказались лучше всех остальных:

* accuracy: 0.9399899617290922
* f1: 0.9618566618566619
* precision: 0.926785919045981
* recall: 1.0

Дальнейший анализ произведём вокруг данного значения. Построим классификаторы с количеством деревьев в окрестности 400-от с шагом 10. Оценим получившиеся значения метрик. Найдем наилучшие значения. Исходя из этого, среди деревьев от 350 до 450, можно сделать вывод, что наилучшим вариантом является вариант с 390 и 450 деревьями. Лучшим из двух является количеств является меньшее, так как при одинаковых значениях метрик использует меньше ресурсов.

Проанализируем классификатор Random Forest состоящий из 390 деревьев. Метрики данного классификатора имеют следующие значения:

* accuracy 0.9399899617290922
* f1: 0.9517726282432165
* precision: 0.926785919045981
* recall: 1.0

Для сравнения с классификатором LogisticRegression будем использовать именно этот вариант классификатора Случайный лес.

Теперь сравним классификатор LogisticRegression (логистическая регрессия) и Случайный Лес (Random Forest), оценивая метрики каждого из них. Анализируя значения данных, я могу заключить, что все метрики accuracy, f1, precision и recall у классификатора Random Forest лучше, а следовательно, классификатор Random Forest для моего набора данных работает лучше и точнее, чем LogisticRegression.

**Вывод:** мною была выполнена задача классификации для набора данных Drug classification двумя разными способами: Логистической регрессией и Случайным Лесом. Оценка точности классификатора LogisticRegression была произведена с помощью метрик accuracy, f1, precision и recall. Получились следующие значения: accuracy: 0.85, f1: 0.84, precision: 0.84, recall: 0,85. После построения классификатора Random Forest я выбрал вариант с наилучшим количеством деревьев, и также оценил его с помощью четырёх метрик: accuracy 0.94, f1: 0.95, precision: 0.93, recall: 1.0. Сравнив метрик каждого из двух классификаторов, я сделал вывод о том, что для моего набора данных лучше использовать классификатор Случайный лес, чем Логистическую регрессию.

## Задача 5

Набор данных: California WildFires

Необходимо провести анализ датасета (из задания 6) и сделать обработку данных по

предложенному алгоритму. Ответить на следующие вопросы:

1. Сколько в датасете объектов и признаков? Дать описание каждому признаку, если оно есть.

В данном наборе 1636 объектов и 40 признаков. Описание признаков (по возможности):

* AcresBurned: Акров земли, пострадавших от лесных пожаров
* Active: является ли пожар активным или локализованным?
* AdminUnit: административное подразделение
* AirTankers: выделенные ресурсы
* ArchiveYear: год, когда данные были заархивированы
* CalFireIncident: рассматривается ли этот инцидент как инцидент с пожаром?
* Counties: название округа
* CountyIds: идентификационный номер округа
* CrewsInvolved: вовлеченные экипажи
* Dozers: выделенные бульдозеры
* Engines: выделенные машины
* Extinguished: дата погашения
* Fatalities: количество погибших
* Helicopters: выделенные вертолеты
* Injuries: количество раненых среди персонала
* Latitude: широта инцидента с лесным пожаром
* Location: описание местоположения
* Longitude: долгота инцидента с лесным пожаром
* MajorIncident: считается ли это серьезным инцидентом или нет?
* Name: название лесного пожара
* PercentContained: какой процент пожара локализован?
* PersonnelInvolved: вовлеченный персонал
* Started: дата начала пожара
* StructuresDamaged: количество поврежденных конструкций
* StructuresDestroyed: количество разрушенных сооружений
* StructuresThreatened: количество сооружений, находящихся под угрозой
* WaterTenders: выделенные объем воды

1. Сколько категориальных признаков, какие?

Категориальные признаки:

* AdminUnit
* Counties
* CountyIds
* Name

1. Столбец с макимальным количеством уникальных значений категориального признака?

Признаки Counties и CountyIds обозначают одно и тоже, поэтому рассматриваю только один из них. В цикле посчитаем уникальные значения. В итоге получилось, что максимальное количество уникальных значений у признака Name, точнее 1193. Уберем категориальные признаки из набора, так как они имеют слишком много уникальных значений.

1. Есть ли бинарные признаки?

Бинарные признаки присутствуют, к ним можно отнести:

* Active
* CalFireIncident
* Featured
* Final
* MajorIncident
* Public
* Status

Преобразуем их значения в числовые, для дальнейшей обработки.

1. Какие числовые признаки?

Числовые признаки:

* AcresBurned
* AirTankers
* CrewsInvolved
* Dozers
* Engines
* Fatalities
* Helicopters
* Injuries
* PersonnelInvolved
* StructuresDamaged
* StructuresDestroyed
* WaterTenders

1. Есть ли пропуски?

# Заключение

1. В задаче №1 я построила графики стандартных отклонений некоторых столбцов набора данных Swiss. А также я построила две математические модели с одним

регрессором в каждой и оценила их по нескольким параметрам.

1. В задаче №2 я строила большое количество (около 20-ти) линейных зависимостей одной переменной от трёх других, одновременно проверяя регрессоры моделей на независимость. Выделила из регрессий лучшую по доле объяснения данных набора Swiss. Для неё я нашла доверительные интервалы для всех коэффициентов, а далее рассчитала доверительных интервал для одного прогноза.
2. В задаче №3 я тщательно искала зависимости между большим количеством параметров из набора данных. Мне пришлось построить более 30-ти линейных регрессий, чтобы найти самую лучшую из них. Тем самым найти самую оптимальную зависимость параметров между собой. Оценив коэффициенты перед объясняющими переменными данной регрессии, я сделала вывод о том, какие индивиды получают большую зарплату. Также я выделила 2 разных подмножества и на них тоже ответила на вопрос о том, какие индивиды больше зарабатывают.

# Список литературы

1. Introduction to Econometrics with R/Christoph Hanck, Martin Arnold, Alexander Gerber, Martin Schmelzer. - Essen, Germany: University of Duisburg-Essen, 2021.
2. Айвазян, С.А. Основы эконометрики/С.А. Айвазян, В.С. Мхитарян – Москва: Изд. объединение «ЮНИТИ», 1998. – 1005 с.
3. Вербик, М. Путеводитель по современной эконометрике/М. Вербик – Москва: «Научная книга», 2008. – 616 с.
4. Доугерти, К. Введение в эконометрику/К. Доугерти – Москва: ИНФРА-М, 2009. – 465 с.
5. Магнус, Я.Р. Эконометрика. Начальный курс/Я.Р. Магнус, П.К. Катышев, А.А. Пересецкий – Москва: Изд-во «ДЕЛО», 2004. – 576 с.

**Приложения**

Приложение 1

library("lmtest")

data = swiss

help(swiss)

data

mean(data$Examination)

mean(data$Infant.Mortality)

mean(data$Education)

data["Examination1"] = data$Examination - mean(data$Examination)

data["Education1"] = data$Education - mean(data$Education)

data["Infant.Mortality1"] = data$Infant.Mortality - mean(data$Infant.Mortality)

data["Examination2"] = (data$Examination - mean(data$Examination))/sqrt(var(data$Examination))

data["Education2"] = (data$Education - mean(data$Education))/sqrt(var(data$Education))

data["Infant.Mortality2"] = (data$Infant.Mortality - mean(data$Infant.Mortality))/sqrt(var(data$Infant.Mortality))

plot(data$Examination2) + abline(a = 0, b = 0, col = "red")

plot(data$Infant.Mortality2) + abline(a = 0, b = 0, col = "blue")

plot(data$Education2) + abline(a = 0, b = 0, col = "green")

model1 = lm(Examination~Education, data)

model1

summary(model1)

plot(data$Examination, data$Education) + abline(a = 10.13, b = 0.58, col = "red")

model2 = lm(Examination~Infant.Mortality, data)

model2

summary(model2)

plot(data$Examination, data$Infant.Mortality) + abline(a = 22.72, b = -0.31, col = "red")

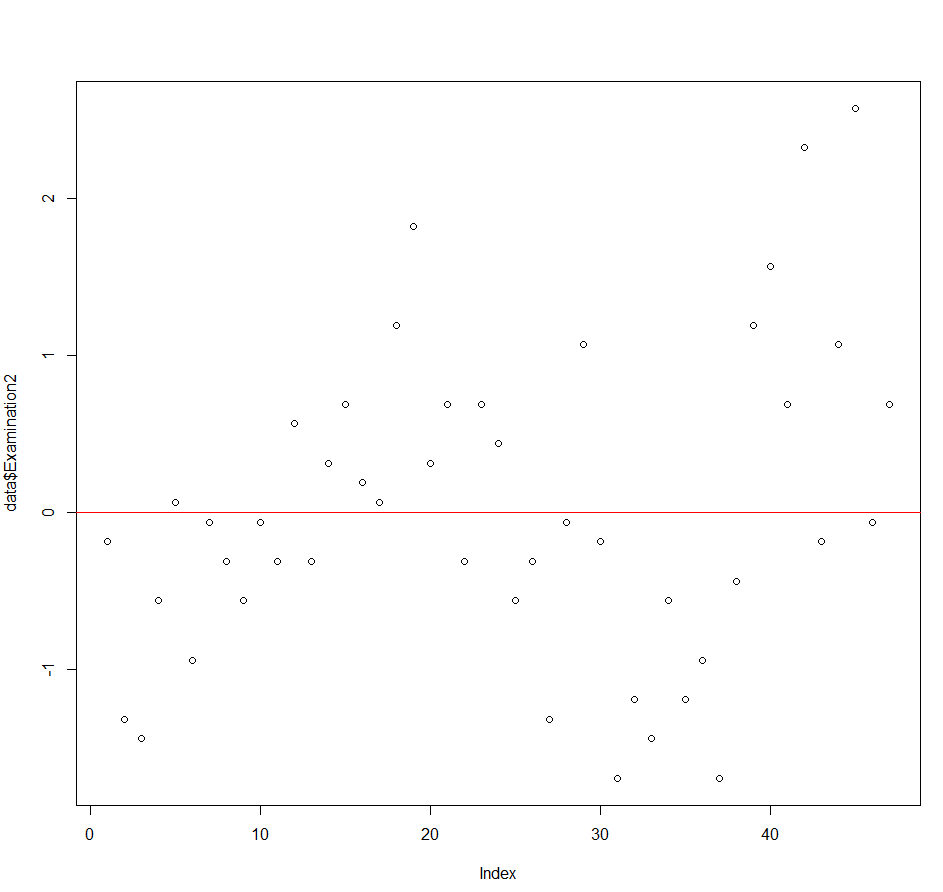


Рисунок 1.1. Отклонения значений переменной Examination от своего среднего значения.

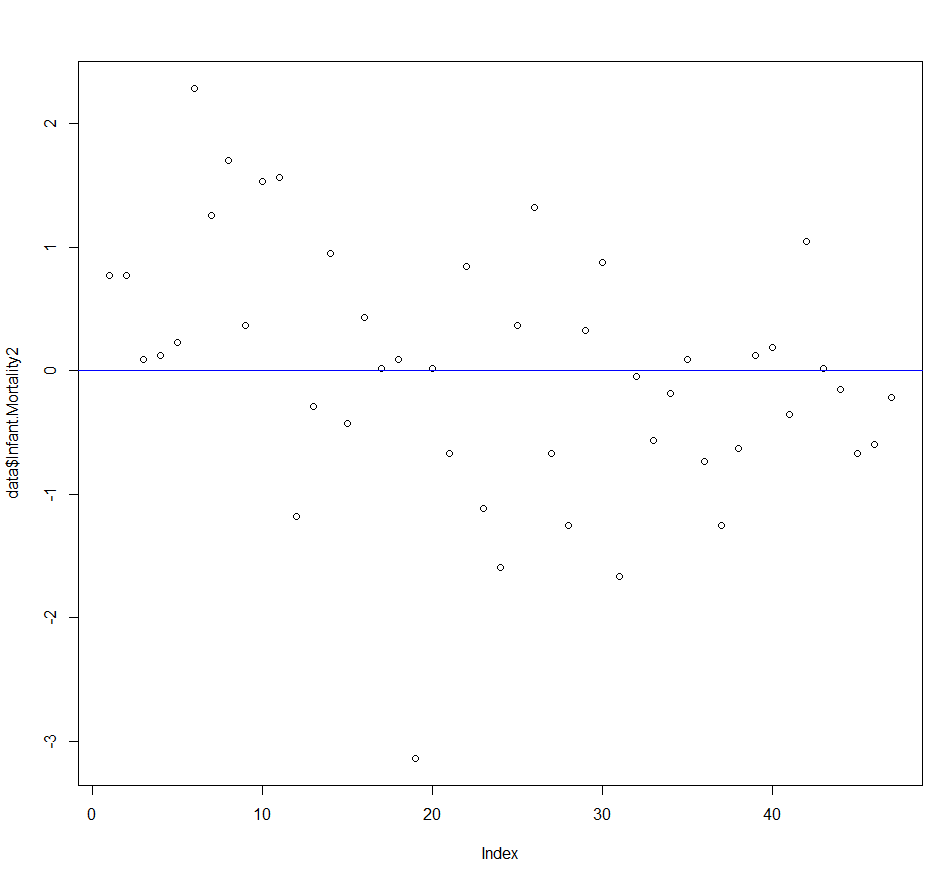


Рисунок 1.2. Отклонения значений переменной Infant.Mortality от своего среднего значения.

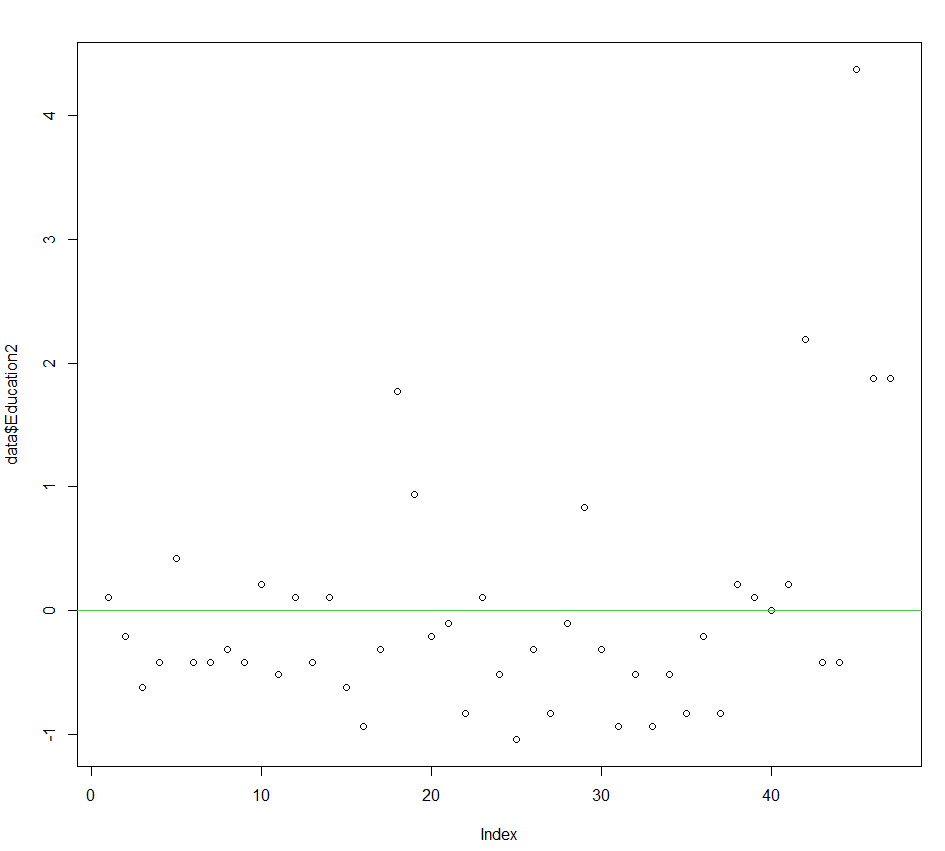


Рисунок 1.3. Отклонения значений переменной Education от своего среднего значения.

## Приложение 2

library("lmtest")

library("car")

data = mtcars

help(mtcars)

data

modela = lm(wt~qsec+hp+drat, data)

modelb = lm(qsec~wt+hp+drat, data)

modelc = lm(hp~wt+qsec+drat, data)

modeld = lm(drat~wt+qsec+hp, data)

summary(modela)

summary(modelb)

summary(modelc)

summary(modeld)

modele = lm(wt~qsec+drat, data)

modelf = lm(qsec~wt+drat, data)

modelg = lm(drat~wt+qsec, data)

summary(modele)

summary(modelf)

summary(modelg)

modelmain = lm(mpg~wt+qsec+drat, data)

summary(modelmain)

vif(modelmain)

model1 = lm(mpg~wt+qsec+drat+log(wt), data)

summary(model1)

vif(model1)

model2 = lm(mpg~wt+qsec+drat+log(qsec), data)

summary(model2)

vif(model2)

model3 = lm(mpg~wt+qsec+drat+log(drat), data)

summary(model3)

vif(model3)

model4 = lm(mpg~wt+qsec+drat+log(wt)+log(qsec), data)

summary(model4)

vif(model4)

model5 = lm(mpg~wt+qsec+drat+log(qsec)+log(drat), data)

summary(model5)

vif(model5)

model6 = lm(mpg~wt+qsec+drat+log(wt)+log(drat), data)

summary(model6)

vif(model6)

model7 = lm(mpg~wt+qsec+drat+log(wt)+log(qsec)+log(drat), data)

summary(model7)

vif(model7)

model8 = lm(mpg~wt+qsec+drat+I(wt^2), data)

summary(model8)

vif(model8)

model9 = lm(mpg~wt+qsec+drat+I(qsec^2), data)

summary(model9)

vif(model9)

model10 = lm(mpg~wt+qsec+drat+I(drat^2), data)

summary(model10)

vif(model10)

model11 = lm(mpg~wt+qsec+drat+I(wt^2)+I(qsec^2), data)

summary(model11)

vif(model11)

model12 = lm(mpg~wt+qsec+drat+I(qsec^2)+I(drat^2), data)

summary(model12)

vif(model12)

model13 = lm(mpg~wt+qsec+drat+I(wt^2)+I(drat^2), data)

summary(model13)

vif(model13)

model14 = lm(mpg~wt+qsec+drat+I(wt^2)+I(qsec^2)+I(drat^2), data)

summary(model14)

vif(model14)

model15 = lm(mpg~wt+qsec+drat+I(wt\*qsec), data)

summary(model15)

vif(model15)

model16 = lm(mpg~wt+qsec+drat+I(qsec\*drat), data)

summary(model16)

vif(model16)

model17 = lm(mpg~wt+qsec+drat+I(drat\*wt), data)

summary(model17)

vif(model17)

model18 = lm(mpg~wt+qsec+drat+I(wt\*qsec)+I(qsec\*drat), data)

summary(model18)

vif(model18)

model19 = lm(mpg~wt+qsec+drat+I(qsec\*drat)+I(drat\*wt), data)

summary(model19)

vif(model19)

model20 = lm(mpg~wt+qsec+drat+I(wt\*qsec)+I(drat\*wt), data)

summary(model20)

vif(model20)

model21 = lm(mpg~wt+qsec+drat+I(wt\*qsec)+I(qsec\*drat)+I(drat\*wt), data)

summary(model21)

vif(model21)

model17 = lm(mpg~wt+qsec+drat+I(drat\*wt), data)

summary(model17)

vif(model17)

modelfinal = lm(mpg~qsec+drat+I(drat\*wt), data)

summary(modelfinal)

vif(modelfinal)

t\_critical = qt(0.975, df = 28)

#t\_critical = 2.048407

new.data = data.frame(qsec= 17.5,drat = 4, wt = 3.4)

predict(modelfinal, new.data, interval = "confidence")

# fit lwr upr

#1 18.69864 17.13805 20.25923

Приложение 3

library("rlms")

library("lmtest")

library("dplyr")

library("GGally")

library("car")

library("sandwich")

ds <- rlms\_read("r15i\_os26c.sav")

data1 = select(ds, kj13.2, kh5, k\_marst, k\_educ, k\_age, status, kj1.1.2, kj6.2, kj21.3, kj23, kj32)

data1 = na.omit(data1)

glimpse(data)

sal = as.numeric(data1$kj13.2)

sal1 = as.character(data1$kj13.2)

sal2 = lapply(sal1, as.integer)

sal = as.numeric(unlist(sal2))

mean(sal)

data1["salary"] = (sal - mean(sal)) / sqrt(var(sal))

data1["sex"]=data1$kh5

data1["sex"] = lapply(data1["sex"], as.character)

data1$sex[which(data1$sex !='1')] <- 0

data1$sex[which(data1$sex =='1')] <- 1

data1$sex = as.numeric(data1$sex)

data1["wed"]= data1$k\_marst

data1["wed"] = lapply(data1["wed"], as.character)

data1["wed1"]= data1$k\_marst

data1$wed1 = 0

data1$wed1[which(data1$wed=='2')] <- 1

data1$wed1 = as.numeric(data1$wed1)

data1["wed2"]= data1$k\_marst

data1$wed2 = 0

data1$wed2[which(data1$wed=='4')] <- 1

data1$wed2[which(data1$wed=='5')] <- 1

data1$wed2 = as.numeric(data1$wed2)

data1["wed3"]= data1$k\_marst

data1$wed3 = 0

data1$wed3[which(data1$wed=='1')] <- 1

data1$wed3[which(data1$wed=='3')] <- 1

data1$wed3 = as.numeric(data1$wed3)

data1["h\_educ"] = data1$k\_educ

data1["h\_educ"] = lapply(data1["h\_educ"], as.character)

data1["higher\_education"] = data1$k\_educ

data1["higher\_education"] = 0

data1$higher\_education[which(data1$h\_educ=='21')] <- 1

data1$higher\_education[which(data1$h\_educ=='22')] <- 1

data1$higher\_education[which(data1$h\_educ=='23')] <- 1

age1 = as.character(data1$k\_age)

age2 = lapply(age1, as.integer)

age3 = as.numeric(unlist(age2))

data1["age"]= (age3 - mean(age3)) / sqrt(var(age3))

data1["status1"]=data1$status

data1["status1"] = lapply(data1["status1"], as.character)

data1["city\_status"] = 0

data1$city\_status[which(data1$status1=='1')] <- 1

data1$city\_status[which(data1$status1=='2')] <- 1

data1$city\_status = as.numeric(data1$city\_status)

data1["sat"]=data1$kj1.1.2

data1["sat"] = lapply(data1["sat"], as.character)

data1["satisfy"] = 0

data1$satisfy[which(data1$sat=='1')] <- 1

data1$satisfy[which(data1$sat=='2')] <- 1

data1$satisfy = as.numeric(data1$satisfy)

dur1 = as.character(data1$kj6.2)

dur2 = lapply(dur1, as.integer)

dur3 = as.numeric(unlist(dur2))

data1["duration"] = (dur3 - mean(dur3)) / sqrt(var(dur3))

data1["dan"] = data1$kj21.3

data1["dan"] = lapply(data1["dan"], as.character)

data1["dangerous"] = data1$kj21.3

data1["dangerous"] = 0

data1$dangerous[which(data1$dan=='1')] <- 1

data1["gov"] = data1$kj23

data1["gov"] = lapply(data1["gov"], as.character)

data1["government"] = data1$kj23

data1["government"] = 0

data1$government[which(data1$gov=='1')] <- 1

data1["w2"] = data1$kj32

data1["w2"] = lapply(data1["w2"], as.character)

data1["second\_job"] = data1$kj32

data1["second\_job"] = 0

data1$second\_job[which(data1$w2=='1')] <- 1

data2 = select(data1, salary, sex, wed1, wed2, wed3, higher\_education, age, city\_status, satisfy, duration, dangerous, government, second\_job)

model\_test = lm(salary~sex+wed1+wed2+wed3+higher\_education+age+city\_status+satisfy+duration+dangerous+government+second\_job, data2)

summary(model\_test)

vif(model\_test)

model\_test2 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+second\_job, data2)

summary(model\_test2)

vif(model\_test2)

model\_original = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government, data2)

summary(model\_original)

vif(model\_original)

model1 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+log(age), data2)

summary(model1)

vif(model1)

model2 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+log(duration), data2)

summary(model2)

vif(model2)

model3 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+log(age)+log(duration), data2)

summary(model3)

vif(model3)

model4 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^0.1), data2)

summary(model4)

vif(model4)

model5 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(duration^0.1), data2)

summary(model5)

vif(model5)

model6 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^0.1)+I(duration^0.1), data2)

summary(model6)

vif(model6)

model7 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^0.2), data2)

summary(model7)

vif(model7)

model8 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(duration^0.2), data2)

summary(model8)

vif(model8)

model9 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^0.2)+I(duration^0.2), data2)

summary(model9)

vif(model9)

model10 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^0.3), data2)

summary(model10)

vif(model10)

model11= lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(duration^0.3), data2)

summary(model11)

vif(model11)

model12 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^0.3)+I(duration^0.3), data2)

summary(model12)

vif(model12)

model13 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^0.4), data2)

summary(model13)

vif(model13)

model14 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(duration^0.4), data2)

summary(model14)

vif(model14)

model15 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^0.4)+I(duration^0.4), data2)

summary(model15)

vif(model15)

model16 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^1.1), data2)

summary(model16)

vif(model16)

model17 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(duration^1.1), data2)

summary(model17)

vif(model17)

model18 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^1.1)+I(duration^1.1), data2)

summary(model18)

vif(model18)

model19 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^1.5), data2)

summary(model19)

vif(model19)

model20 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(duration^1.5), data2)

summary(model20)

vif(model20)

model21 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^1.5)+I(duration^1.5), data2)

summary(model21)

vif(model21)

model22 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^1.6), data2)

summary(model22)

vif(model22)

model23 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(duration^1.6), data2)

summary(model23)

vif(model23)

model24 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^1.6)+I(duration^1.6), data2)

summary(model24)

vif(model24)

model25 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^1.9), data2)

summary(model25)

vif(model25)

model26 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(duration^1.9), data2)

summary(model26)

vif(model26)

model27 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^1.9)+I(duration^1.9), data2)

summary(model27)

vif(model27)

model28 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^2), data2)

summary(model28)

vif(model28)

model29= lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(duration^2), data2)

summary(model29)

vif(model29)

model30 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+I(age^2)+I(duration^2), data2)

summary(model30)

vif(model30)

model1 = lm(salary~sex+wed1+wed2+higher\_education+age+city\_status+satisfy+duration+dangerous+government+log(age), data2)

summary(model1)

vif(model1)

data3 = subset(data2, city\_status == 1)

data3

data4 = subset(data3, wed1 == 0)

data4

model\_subset1 = lm(data = data4, salary~sex+higher\_education+age+satisfy+duration+dangerous+government+log(age))

summary(model\_subset1)

vif(model\_subset1)

data5 = subset(data2, wed2 == 1)

data5

data6 = subset(data5, sex == 0)

data6

data7 = subset(data6, higher\_education == 1)

data7

model\_subset2 = lm(data = data7, salary~age+city\_status+satisfy+duration+dangerous+government+log(age))

summary(model\_subset2)

vif(model\_subset2)

## Приложение 4

import pandas

import numpy as np

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import GridSearchCV

from sklearn.model\_selection import cross\_val\_score

data = pandas.read\_csv('drug200.csv')

data\_sel = data.loc[:, data.columns.isin(['Drug', 'Age', 'Sex', 'BP', 'Cholesterol', 'Na\_to\_K'])]

data\_sel = data\_sel.dropna()

#Обработаем столбец Sex. 0 - женщина, 1 - мужчина

data\_sel['Sex'] = np.where(data\_sel['Sex'] == 'F', 0, 1)

#Обработаем столбец BP. 0 - нормальный, 1 - высокий, 2 - низкий

data\_sel['BP'] = np.where(data\_sel['BP'] == 'LOW', 2, data\_sel['BP'])

data\_sel['BP'] = np.where(data\_sel['BP'] == 'NORMAL', 0, data\_sel['BP'])

data\_sel['BP'] = np.where(data\_sel['BP'] == 'HIGH', 1, data\_sel['BP'])

#Обработаем столбец Cholesterol. 0 - нормальный, 1 - высокий

data\_sel['Cholesterol'] = np.where(data\_sel['Cholesterol'] == 'NORMAL', 0, 1)

#Обработаем столбец Drug. DrugA – класс 0, остальные уровни – класс 1

data\_sel['Drug'] = np.where(data\_sel['Drug'] == 'drugX', 0, 1)

Drug = data\_sel.loc[:, data\_sel.columns.isin(['Drug'])]

X = data\_sel.loc[:, data\_sel.columns.isin(['Age', 'Sex', 'BP', 'Cholesterol', 'Na\_to\_K'])]

#Выводим изменённую таблицу

X

#Делим данные на обучающую и тестовую выборку

x\_train, x\_validation, y\_train, y\_validation = train\_test\_split(X, Drug, test\_size=.33, random\_state=5)

#Построим классификатор Логистическая регрессия (LogisticRegression)

logistic = LogisticRegression(solver='lbfgs')

logistic.fit(x\_train, y\_train.values.ravel())

logistic\_pred = logistic.predict(x\_validation)

print("Logistic Regression Test Accuracy: " + str(logistic.score(x\_validation, y\_validation)\*100) + "%")

print('Report for Logistic Regression: ')

print(classification\_report(y\_validation, logistic\_pred))

#Cтроим классификатор Случайный Лес (Random Forest)

#Найдеч лучшее количество деревьев с шагом 100

accuracy\_ = []

precision\_ = []

recall\_ = []

F1\_ = []

for i in range(200, 400, 100):

param\_grid = { 'n\_estimators': [i],'max\_features': ['auto'],'max\_depth' : list(range(1, 20)), 'criterion' :['gini']}

RFC = GridSearchCV(estimator=RandomForestClassifier(), param\_grid=param\_grid, cv= 5, refit = True)

RFC.fit(x\_train, y\_train.values.ravel())

accuracy\_.append(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='accuracy')))

precision\_.append(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='precision')))

recall\_.append(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='recall')))

F1\_.append(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='f1')))

print(accuracy\_, "\n", precision\_, "\n", recall\_, "\n", F1\_)

#Как видно из списков метрик 400 деревьев оптимальный вариант, походим вокруг него с шагом 10

accuracy\_ = []

precision\_ = []

recall\_ = []

F1\_ = []

for i in range(350, 460, 10):

param\_grid = { 'n\_estimators': [i],'max\_features': ['auto'],'max\_depth' : list(range(1, 20)), 'criterion' :['gini']}

RFC = GridSearchCV(estimator=RandomForestClassifier(), param\_grid=param\_grid, cv= 5, refit = True)

RFC.fit(x\_train, y\_train.values.ravel())

accuracy\_.append(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='accuracy')))

precision\_.append(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='precision')))

recall\_.append(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='recall')))

F1\_.append(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='f1')))

print(accuracy\_, "\n", precision\_, "\n", recall\_, "\n", F1\_)

m\_acc = max(accuracy\_)

m\_prec = max(precision\_)

m\_recall = max(recall\_)

m\_f1 = max(F1\_)

for i, j in zip([accuracy\_, precision\_, recall\_, F1\_], [m\_acc, m\_prec, m\_recall, m\_f1]):

print([k for k, l in enumerate(i) if l == j])

#Сравним метрики обоих классификаторов

#Логистическая регрессия

accuracy: 0.85

f1: 0.84

precision: 0.84

recall: 0.85

#Случайный лес

accuracy: 0.9399899617290922

f1: 0.9517726282432165

precision: 0.926785919045981

recall: 1.0

#Метрики f1 и recall лучше у Случайного Леса, значит он показывает себя лучше на этих данных, чем Логистическая регрессия

## Приложение 5

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.decomposition import PCA

from sklearn import preprocessing

from sklearn.manifold import TSNE

data = pd.read\_csv('California\_Fire\_Incidents.csv')

data.shape

#Tаблица содержит 1636 строк (объектов) и 40 столбцов (признаков).

data.head()

#Описание признаков:

# AcresBurned: Акров земли, пострадавших от лесных пожаров

# Active: Является ли пожар активным или локализованным?

# AdminUnit: Административное подразделение

# AirTankers: Выделенные ресурсы

# ArchiveYear: Год, когда данные были заархивированы

# CalFireIncident: Рассматривается ли этот инцидент как инцидент с пожаром?

# Counties: Название округа

# CountyIds: Идентификационный номер округа

# CrewsInvolved: Вовлеченные экипажи

# Dozers: Выделенные бульдозеры

# Engines: Выделенные машины

# Extinguished: Дата погашения

# Fatalities: Количество погибших

# Helicopters: Выделенные вертолеты

# Injuries: Количество раненых среди персонала

# Latitude: Широта инцидента с лесным пожаром

# Location: Описание местоположения

# Longitude: Долгота инцидента с лесным пожаром

# MajorIncident: Считается ли это серьезным инцидентом или нет?

# Name: Название лесного пожара

# PercentContained: Какой процент пожара локализован?

# PersonnelInvolved: Вовлеченный персонал

# Started: Дата начала пожара

# StructuresDamaged: Количество поврежденных конструкций

# StructuresDestroyed: Количество разрушенных сооружений

# StructuresThreatened: Количество сооружений, находящихся под угрозой

# WaterTenders: Выделенные объем воды

#

#Из них категориальные:

# AdminUnit

# Counties

# CountyIds

# Name

#Counties и CountyIds обозаначают обно и тоже, поэтому рассматриваю только один из них

unique = [[], [], []]

k = 0

for i in ['AdminUnit', 'Counties', 'Name']:

for j in range(0, 1636):

if not(data[i][j] in unique[k]):

unique[k].append(data[i][j])

k=k+1

for i in unique:

print(len(i))

#Столбецом с макимальным количеством уникальных значений категориального признака является Name

#с количеством уникальных значений 1193

data = data.drop(['AdminUnit', 'Counties', 'CountyIds', 'Name'], axis = 1)

#Есть бинарные признаки, к ним относятся:

# Active

# CalFireIncident

# Featured

# Final

# MajorIncident

# Public

# Status

bin\_cols = ['Active', 'CalFireIncident', 'Featured', 'Final', 'MajorIncident', 'Public', 'Status']

data['Active'] = np.where(data['Active'] == 'False', 0, 1)

data['CalFireIncident'] = np.where(data['CalFireIncident'] == 'False', 0, 1)

data['Featured'] = np.where(data['Featured'] == 'False', 0, 1)

data['Final'] = np.where(data['Final'] == 'False', 0, 1)

data['MajorIncident'] = np.where(data['MajorIncident'] == 'False', 0, 1)

data['Public'] = np.where(data['Public'] == 'False', 0, 1)

data['Status'] = np.where(data['Status'] == 'Inactive', 0, 1)

col = []

j = 0

for i in data.isna().any():

if i:

col.append(data.columns[j])

j+=1

print(col, " : ", len(col))

#Пропуски присутствуют в 21 столбце

#Эти столбцы видны выше

data\_temp = data

count\_na = []

for i in col:

data\_temp[i].dropna()

count\_na.append(1636 - data\_temp[i].count())

#for i, j in zip(col, count\_na):

# if j > 1500:

# data = data.drop([i], axis = 1)

#

#data = data.dropna()

print(count\_na, "\n",col[count\_na.index(max(count\_na))], max(count\_na))

#StructuresEvacuated полностью пустой столбец

#В датасете имеется много столбцов с метаданными, которые неудобно обрабатывать, которые следует исключить

data = data.drop(['ArchiveYear', 'CanonicalUrl', 'ConditionStatement', 'ControlStatement', 'Extinguished', 'FuelType', 'Location', 'SearchDescription', 'SearchKeywords', 'Started', 'UniqueId', 'Updated'], axis = 1)

#Cтатистический анализ числовых столбцов

data.describe()

#Анамальные отклонения в столбцах AcresBurned, CrewsInvolved, Dozers, Engines, Fatalities,

#Injuries, PersonnelInvolved, StructuresDamaged, StructuresDestroyed, WaterTenders

# Нормализация признаков через стандартное отклонение

data = data.fillna(0)

num\_cols = ['AcresBurned', 'AirTankers', 'CrewsInvolved', 'Dozers', 'Engines', 'Fatalities', 'Helicopters', 'Injuries', 'PersonnelInvolved', 'StructuresDamaged', 'StructuresDestroyed', 'WaterTenders']

scale\_features\_std = StandardScaler()

features\_std = scale\_features\_std.fit\_transform(data[num\_cols+bin\_cols])

data[num\_cols+bin\_cols] = features\_std

mean = []

s = 0

for i in range(0,1):

for j in range(0,1425):

s += features\_std[j][i]

s = s/1636

mean.append(s)

s = 0

print(num\_cols[mean.index(max(mean))], max(mean))

#Столбецом с максимальным средним значением после нормировки признаков через стандартное отклонение является CrewsInvolved

#Явного столбца, который бы оценивал силу пожара, нет, поэтому буду опираться на площадь выжженных лесов

#Нормализуем данные

#d = preprocessing.normalize(data, axis = 0)

#norm\_data = pd.DataFrame(d, columns = data.columns)

#new\_data = pd.concat([norm\_data, data[bin\_cols]], sort = False, axis = 1)

target = data.loc[:, data.columns.isin(['AcresBurned'])]

train = data.drop(['AcresBurned'], axis = 1)

train

#Выделяем тренировочную и тестовую выборки

# y - целевая переменная(target)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(train, target, test\_size = 0.3, random\_state = 42)

N\_train, \_ = X\_train.shape

print(N\_train, \_)

#1145 объектов попадает в тренировочную выборку при использовании

#train\_test\_split с параметрами test\_size = 0.3, random\_state = 42

#Корреляция признаков

corr\_frame = data.corr()

corr\_frame

#Используем метод PCA

pca = PCA()

pca.fit(X\_train)

X\_pca = pca.transform(X\_train)

for i, component in enumerate(pca.components\_):

print("{} component: {}% of initial variance".format(i + 1,

round(100 \* pca.explained\_variance\_ratio\_[i], 2)))

print(" + ".join("%.3f x %s" % (value, name)

for value, name in zip(component, train.columns)))

#90% дисперсии объясняется минимум 2-мя признаками

tsne1 = TSNE(learning\_rate=200, random\_state=13)

transformed1 = tsne1.fit\_transform(train)

x\_axist1 = transformed1[:, 0]

y\_axist1 = transformed1[:, 1]

plt.scatter(x\_axist1, y\_axist1, c=target['AcresBurned'].tolist())

plt.show()

tsne1 = TSNE(learning\_rate=200, random\_state=130)

transformed1 = tsne1.fit\_transform(train)

x\_axist1 = transformed1[:, 0]

y\_axist1 = transformed1[:, 1]

plt.scatter(x\_axist1, y\_axist1, c=target['AcresBurned'].tolist())

plt.show()

#Как видно из результатов, данный датасет не поддается кластеризации

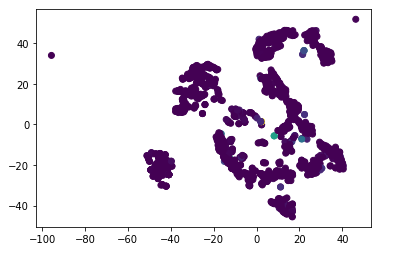


Рисунок 5.1 Двухмероное представление данных с помощью алгоритма t-SNE.

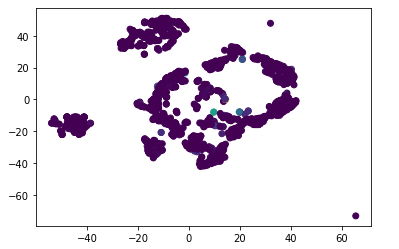


Рисунок 5.1 Двухмероное представление данных с помощью алгоритма t-SNE, со значением random\_state = 130.