Оглавление

[Задачи 3](#_Toc104760811)

[Задача 1 3](#_Toc104760812)

[Задача 2 8](#_Toc104760813)

[Задача 3 16](#_Toc104760814)

[Задача 4 27](#_Toc104760815)

[Задача 5 32](#_Toc104760816)

[Заключение 43](#_Toc104760817)

[Список литературы 45](#_Toc104760818)

[**Приложения** 46](#_Toc104760819)

[Приложение 1 46](#_Toc104760820)

[Приложение 2 50](#_Toc104760821)

[Приложение 3 65](#_Toc104760822)

[Приложение 4 88](#_Toc104760823)

[Приложение 5 95](#_Toc104760824)

# Задачи

## Задача 1

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: Swiss

Объясняемая переменная: *Examination*

Регрессоры: *Education, Agriculture*

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

Среднее значение столбца *Examination* равно 16.49, тогда только каждый шестой призывник получил наивысшую оценку на армейском экзамене, что свидетельствует о достаточно высоком уровне образованности и грамотности.

Среднее значение столбца у регрессора *Education* равно 10.98, следовательно, каждый десятый призывник занимался обучением за пределами начальной школы, что является весьма низким показателем в современных реалиях.

А среднее значение столбца регрессора *Agriculture* равно 50.66, значит, примерно половина всех мужчин занималась сельским хозяйством в качестве основного рода деятельности.

Для оценки дисперсии и СКО переменных произведём нормализацию и обработаем данные, после построим графики, на которых будут видны значения переменных и степень их отклонения от среднего.

Произведём анализ графика для переменной *Examination* (Рисунок 1.1). Около среднего значения данные разбросаны, все они отличаются на 1 стандартное отклонение в положительную и отрицательную сторону, но есть и аномальные значения, которые достаточно отдалены от среднего значения (2-2.5 стандартных отклонения). Из этого следует, что призывники из различных провинций могут иметь как наивысшую оценку за армейский экзамен, так и неудовлетворительную оценку, что говорит, о несбалансированной системе подготовки к экзаменам.

Далее произведём анализ для регрессора *Education* (Рисунок 1.2). Большее число значений меньше среднего и отклонены от него не сильно (1 стандартное отклонение), но есть и выделяющиеся значения (больше среднего), отличающиеся от среднего на 2 и даже 4 стандартных отклонения. Таким образом, можно сделать вывод, что уровни образования в различных провинциях достаточно неодинаков, но также и прослеживается общая тенденция – как правило, уровень образования невысокий.

Теперь для регрессора *Agriculture* (Рисунок 1.3). Около среднего значения наблюдается группировка данных, в основном они все отличаются на меньше, чем на 1-2 стандартных отклонений в обе стороны. Можно сделать вывод, что процент мужчин, занятых в сельском хозяйстве в качестве основной профессии, в различных провинциях приблизительно одинаковый, то есть сельское хозяйство не теряет своей значимости.

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).

Теперь построим зависимость переменной *Examination* от *Education*.

Коэффициент детерминации R2 равен 48.8%, это достаточно мало, но зависимость все-таки имеется, поэтому модель относительно неплоха. Можно сказать, что четкого представления о зависимости данных в наборе Swiss эта модель не дает. P-статистика очень близка к 0, есть целых 3 "звёздочки" в этой модели, а следовательно, существует взаимосвязь между *Examination* и *Education*. 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 | \*\*\* |

Далее произведём анализ для зависимости переменной *Examination* от *Agriculture*.

Коэффициент детерминации R2 данной модели равен 47,13%, это достаточно мало, но зависимость все-таки имеется, поэтому модель относительно неплоха. Можно сказать, что четкого представления о зависимости данных в наборе Swiss эта модель не дает. P-статистика очень близка к 0, "звёздочки" в модели у регрессора есть, их целых 3, следовательно взаимосвязь между *Examination* и *Agriculture* переменной прослеживается, и она достаточна значительная.

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 *Agriculture* в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | 28.70668 | 2.11001 | 13.605 | < 2e-16 | \*\*\* |
| Agriculture | -0.24117 | 0.03807 | -6.334 | 9.95e-08 | \*\*\* |

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

***Вывод:*** в результате я смогла оценить средние значения, дисперсии и среднеквадратичные отклонения столбцов *Examination, Education* и *Agriculture* из набора данных Swiss. Также мною были построены графики, на которых видны стандартные отклонения всех значений столбцов *Examination, Education* и *Agriculture.* В результате мне удалось построить две линейные регрессии, каждая с одним регрессором, и оценить их по величине коэффициента R2, так же по уровню значимости регрессоров и при помощи анализа P-статистики и стандартной ошибки коэффициентов перед регрессорами. В итоге зависимость *Examination* от *Education* оказалась весьма информативной регрессией, поскольку объясняемая переменная связана с объясняющей, R2 =48.8%, а значит, удалось выявить определенную связь между уровнем образования и оценки, полученной на экзамене. Скорее всего, это зависимость является прямой: чем выше уровень образованности в отдельной провинции, чем выше оценка на экзамене, поскольку же p-статистика <0.001, значит, прослеживается достаточно сильная взаимосвязь. Зависимость переменной *Examination* от регрессора *Agriculture* оказалась немного хуже предыдущей модели, у этой модели R2 = 47,13%. Можно заявить, что существует определенная связь между уровнем образования и количества мужчин, занятых в сельском хозяйстве. Скорее всего, это зависимость будет являться отрицательной, поскольку коэффициент отрицательный: чем выше число мужчин, занятых в сельском хозяйстве, тем ниже будет уровень образованности в отдельной провинции, так как такие районы больше сконцентрированы на трудовой деятельности, а не на обучении. В этой же модели p-статистика также <0.001, значит, прослеживается достаточно сильная взаимосвязь. Но и для 1, и 2 модели необходимо добавить дополнительные параметры, чтобы полностью описать данную зависимость.

## Задача 2

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: Seatbelts

Объясняемая переменная: *DriversKilled*

Регрессоры: *law, kms, PetrolPrice*

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 *law ~ kms + PetrolPrice:* в этой зависимостиR2 = 28.11%, то есть очень слабая зависимость, скорее всего, ей можно пренебречь, и можно использовать переменную *law* в последующих регрессиях.

Далее построим линейную зависимость *kms ~ law + PetrolPrice*. R2 = 27,67%, это меньше, чем в предыдущей модели. Есть очень слабая зависимость, скорее всего, ей можно пренебречь, и использовать переменную *kms* в последующих регрессиях.

Конечно, проверим и третью модель *PetrolPrice ~ law + kms*. R2 регрессии равен 19,29%%, это мало для того, чтобы переменная *PetrolPrice* была линейно зависима от регрессоров этой модели. Значит параметр *PetrolPrice* можно использовать в построении математических моделей.

В итоге каждый регрессор, указанный, в задании можно использовать с остальными для построения моделей линейных регрессий, но нужно следить за ними потому, что прослеживается слабая 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 *DriversKilled ~ law + kms + PetrolPrice* Значения коэффициентов данной модели, их стандартные ошибки, p-статистика и уровень значимости приведены в таблице 2.1.

Оценим модель: коэффициент детерминации R2 = 18.82%, это достаточно низкий показатель, значит, модель очень плоха и не объясняет данные в наборе Seatbelts. P-значения регрессоров *law, kms, PetrolPrice* довольно низкие, у *law* и *kms* есть 1 звездочка и 1 точка соответственно, а у регрессора *PetrolPrice* 3 звёздочки. У всех регрессоров значения стандартной ошибки достаточно велики, но при этом стоит отметить, что VIF у всех коэффициентов не превышает 2, что указывает на небольшую, даже незначительную зависимость используемых регрессоров.

Заключение: данная математическая модель достаточно плоха, но при этом в ней все регрессоры не «не плохие», но и «не хорошие». Есть регрессор, от которого линейно зависит объясняемая переменная, но есть и не очень нужные регрессоры, которые можно было бы исключить без большого вреда для R2.

Таблица 2.1. Характеристики модели зависимости *DriversKilled* от регрессоров *law, kms, PetrolPrice* в наборе данных Seatbelts.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | 2.015e+02 | 1.626e+01 | 12.393 | <2,00E-16 | \*\*\* |
| law | -1.189e+01 | 6.026e+00 | -1.973 | 0.049955 | \* |
| kms | -1.223e-03 | 6.657e-04 | -1.838 | 0.067676 | . |
| PetrolPrice | -5.683e+02 | 1.521e+02 | -3.738 | 0.000246 | \*\*\* |

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

Для решении данного пункта задания я построил модели с использованием регрессоров: *ln(law)*, *ln(kms)*, *ln(PetrolPrice).* 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 *PetrolPrice* уменьшаетR2 в любой модели. Поэтому, после тестов всех комбинаций стало ясно, что *I(log(PetrolPrice))* стоит внести в модель с логарифмом, а сам *PetrolPrice* стоит исключить из модели.

Итог: самой лучшей моделью среди моделей с добавлением натуральных логарифмов от регрессоров является это *DriversKilled ~ law + kms + I(log(PetrolPrice)).* Значения коэффициентов модели, их стандартные ошибки, p-статистика и уровень значимости приведены в таблице 2.3.

Таблица 2.3. Характеристики модели зависимости *DriversKilled* от регрессоров *law*, *kms*, *ln(PetrolPrice)* в наборе данных Seatbelts.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | 8.229e+00 | 3.855e+01 | 0.213 | 0.83118 |  |
| law | -1.183e+01 | 6.006e+00 | -1.969 | 0.05041 | . |
| kms | -1.233e-03 | 6.626e-04 | -1.861 | 0.06431 | . |
| I(log(PetrolPrice)) | -5.914e+01 | 1.535e+01 | -3.853 | 0.00016 | \*\*\* |

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(PetrolPrice^2), I(kms^2), I(PetrolPrice\*kms), I(PetrolPrice\*law), I(law\*kms).*

Здесь оказалось, что регрессор *kms* уменьшаетR2 в любой модели, а растёт R2 лучше всего при добавлении *I(kms^2)*. Лучшей моделью с R2 = 19% оказалась линейная регрессия с исключением kms и добавлением I(kms^2). Все её характеристики параметров приведены в таблице 2.4.

Таблица 2.4. Характеристики модели зависимости *DriversKilled* от регрессоров *law, PetrolPrice, I(kms^2)* в наборе данных Seatbelts.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | 1.925e+02 | 1.516e+01 | 12.699 | <0,001 | \*\*\* |
| law | -1.111e+01 | 6.130e+00 | -1.813 | 0.071481 | . |
| PetrolPrice | -5.601e+02 | 1.524e+02 | -3.674 | 0.000311 | \*\*\* |
| I(kms^2) | -4.407e-08 | 2.263e-08 | -1.947 | 0.052978 | . |

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.

Сравним улучшенную первоначальную модель с лучшей моделью с логарифмом *DriversKilled ~ law + kms + I(log(PetrolPrice))*. R2 = 19.17%, и лучшей моделью с добавлением произведения регрессоров DriversKilled ~ law + PetrolPrice + I(kms^2), где R2 = 19%.

Наилучшая модель выявляется по значению R2, тогда это будет модель с добавлением логарифма *DriversKilled ~ law + kms + I(log(PetrolPrice)).* VIF у регрессоров данной модели меньше двух, а значит, серьезной зависимости между регрессорами нет. Значения коэффициентов данной модели, их стандартные ошибки, p-статистика и уровень значимости приведены в таблице 2.5.

Таблица 2.5. Характеристики модели зависимости *DriversKilled* от регрессоров *law*, *kms*, *ln(PetrolPrice)* в наборе данных Seatbelts.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | 8.229e+00 | 3.855e+01 | 0.213 | 0.83118 |  |
| law | -1.183e+01 | 6.006e+00 | -1.969 | 0.05041 | . |
| kms | -1.233e-03 | 6.626e-04 | -1.861 | 0.06431 | . |
| I(log(PetrolPrice)) | -5.914e+01 | 1.535e+01 | -3.853 | 0.00016 | \*\*\* |

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 = 121) получим, что t = 1.979764, округлим до двух знаков после запятой, t = 1.98.

Доверительный интервал для коэффициента регрессора *law* имеет вид [-11.83-(1.98\*6), -11.83+(1.98\*6)], то есть [-23.71, 0.05].

Значение "0" попадает в доверительный интервал коэффициента перед регрессором, а значит, переменная *DriversKilled* практически не связанапеременной *law.*

Найдем доверительный интервал для коэффициента регрессора *kms*: [-0.001-(1.98\*0), -0.001+(1.98\*0)], то есть приблизительно [-0.001, 0.001].

Значение "0" попадает в доверительный интервал коэффициента перед регрессором, а значит, переменная *DriversKilled* практически не связана спеременной *kms.*

Доверительный интервал для коэффициента *I(log(PetrolPrice))* равен [-59.14-(1.98\*15.35), -59.14+(1.98\*15.35)], то есть [-89.533, -28.747].

Значение "0" не попадает в доверительный интервал коэффициента перед регрессором, следовательно, переменная *DriversKilled* связана с переменной *I(log(PetrolPrice)).*

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

Найдём доверительный интервал для прогноза, в котором переменные будут иметь значения: *law* = 20, *kms* = 10, *PetrolPrice* = 10. 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 -617.2967. Верхняя граница (upr в результате функции predict) равна -111.6801. А среднее прогнозируемое значение (fit в результате функции predict) равно -364.4884.

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

***Вывод:*** я проверила данные мне в задание регрессоры *law, kms, PetrolPrice* на линейную зависимость. В конечном итоге все регрессоры могли быть использованы в модели, поскольку между ними не было зависимости. Далее я построила линейную регрессию с помощью этих переменных и оценила её по величине R2 и по характеристикам коэффициентов перед регрессорами. Математическая модель *DriversKilled ~ law + kms + PetrolPrice,* гдеR2 = 18.82%, оказалась очень плохой, поэтому я попыталась улучшить ее с помощью введения натуральных логарифмов от регрессоров и их попарных произведений. Таким образом, проанализировав более 26 линейных регрессий, я смогла выявить наилучшую модель с имеющимся набором регресссоров. Наилучшей моделью стала линейная регрессия с добавлением логарифма *DriversKilled ~ law + kms + I(log(PetrolPrice)),* гдеR2 = 19.17%, зависимости между регрессорами нет (VIF < 2). К сожалению, поскольку в конечном итоге R2 < 30%, то модель в целом все равно осталась плохой, редактировать эту модель не стоит, так как зависимости нет. Скорее всего, необходимо строить другую модель с другим набором регрессоров. С помощью наилучшей модели мне удалось найти доверительные интервалы для всех коэффициентов регрессии *DriversKilled ~ law + kms + I(log(PetrolPrice))* (p = 95%) и сделать вывод о том, что коэффициент может быть равен нулю или нет. Получилось, что переменная *DriversKilled* не зависит от регрессоров law и kms, а от *+ I(log(PetrolPrice))* зависит, потому что в их доверительных интервалах не попадает значение «0». Поскольку модель оказалась плохой то и стандартные ошибки регрессоров значительно высокие, в связи с чем доверительный интервал оказался очень большим. Но, несмотря на это, я рассчитала доверительный интервал для прогноза, выбрав при этом значения регрессоров равными: *law* = 20, *kms* = 10, *PetrolPrice* = 10. Тогда доверительный интервал получился [-617.2967, -111.6801].

## Задача 3

Номер волны выборки РМЭЗ: 14

Подмножества для пункта 5: Мужчины, разведённые с высшим образованием; женщины, живущие в городе, состоящие в браке

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.

Начать данную задачу необходимо с набора параметров, который необходим, для описания социально-экономического положения граждан Российской Федерации. Я выбрала следующие параметры: зарплата, продолжительность рабочей недели, возраст, пол, наличие высшего образования, населённый пункт, семейное положение и удовлетворенность условиями труда. В файле с данными эти столбцы имеют названия: *jj13.2, jh5, j\_marst, j\_diplom, status, jj6.2, j\_age, j\_educ и jj1.1.2* .

Выбранные данные необходимо преобразовать для дальнейшего анализа по следующим принципам:

* Факторные переменные: зарплата, длительность рабочей недели и возраст, - необходимо преобразовать в вещественные переменные и нормализовать их. Для этого вычтем среднее значение по этой переменной, разделим её значения на стандартное отклонение.
* Для параметра, отвечающего за тип населённого пункта, создадим одну дамми-переменную status2 со значением 1 для города и областного центра, 0 – в противоположном случае.
* Для параметра, отвечающего за семейное 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.
* Для параметра, отвечающего, за пол сделаем переменную sex, имеющую значение 1 для мужчин и значение 0 для женщин.
* Для всех 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 не выполняется.

Так мы получим data2 с новыми переменными. С ними уже можно работать и строить линейную зависимость зарплаты от остальных переменных: *salary ~ workweek, age, sex, j\_educ, status2, wed1, wed2, wed3, satisfy1, satisfy2.* Теперь сможем определить: R2 = 18.27%. Три "звёздочки" у регрессоров: *workweek, age, sex, j\_educ, status2, wed3, satisfy1* и 1 «звёздочка» у *satisfy1,* их p-статистика довольно мала. VIF всех регрессоров < 3, из этого следует, что регрессоры независимы друг с другом, их все можно использовать в одной модели. С подробной характеристиками получившейся модели можно ознакомиться в таблице 3.1.

Таблица 3.1. Характеристики модели зависимости *salary ~ workweek, age, sex, j\_educ, status2, wed1, wed2, wed3, satisfy1 и satisfy2* в наборе данных 14-ой волны исследования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | -0.623975 | 0.069485 | -8.980 | <0,0001 | \*\*\* |
| workweek | 0.138963 | 0.017343 | 8.013 | 1.61e-15 | \*\*\* |
| age | -0.082745 | 0.019086 | -4.335 | 1.50e-05 | \*\*\* |
| sex | 0.466964 | 0.035530 | 13.143 | <0,0001 | \*\*\* |
| j\_educ | 0.413411 | 0.039512 | 10.463 | <0,0001 | \*\*\* |
| status2 | 0.392556 | 0.038091 | 10.306 | <0,0001 | \*\*\* |
| wed1 | -0.004748 | 0.057544 | -0.083 | 0.934251 |  |
| wed2 | -0.087033 | 0.069563 | -1.251 | 0.210980 |  |
| wed3 | -0.264862 | 0.070516 | -3.756 | 0.000176 | \*\*\* |
| satisfy1 | 0.239328 | 0.042830 | 5.588 | 2.51e-08 | \*\*\* |
| satisfy2 | -0.097406 | 0.045189 | -2.156 | 0.031202 | \* |

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

Функциями вещественных параметров являются: возраст и продолжительность рабочей недели. Начнем с того, что будем возводить регрессоры *workweek* и *age* в разные степени и добавлять их к первоначальной линейной регрессию.

Для начала рассмотрим модели со степенями регрессоров *workweek* и *age* от 0.1 до 0.4. Я заметила, что лучшими моделями однозначно будут модели, где регрессор *age* находится в какой-либо степени, поскольку с увеличением степени R2 увеличивается. Так для линейной регрессии *salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(0.1)),* гдеR2 = 21.81%. При росте степеней наблюдается увеличение R2 и vif увеличивается. Для степени 0.5 vif > 10, vif растёт с ростом степени до 1.2.

Начиная со степени 1.3 наблюдается некая закономерность - снижение vif, он становится всё более приемлемым. Достигнув степени 2 vif приходит в норму. *salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(2)).* R2 = 19.49%.

Таким образом, наилучшей моделью среди степеней будет являться *salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(0.4))*, поскольку R2 = 21.84% – наибольший из всех рассмотренных моделей.

С логарифмами же вариантов будет гораздо меньше: нужно рассмотреть каждый логарифм по отдельности, а потом вместе одновременно. После анализа значений R2 и VIF для данных трёх зависимостей определяем, что лучшей из них будет третья модель *salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(log(age)).* Её R2= 21.8%, а команда vif находится в пределах нормы, все его значения меньше 3. Подробная информация о значении коэффициентов, стандартных ошибках и значимости использующихся регрессоров приведена в таблице 3.2.

Таблица 3.2. Характеристики модели зависимости *salary ~ workweek, age, sex, j\_educ, status2, wed1, wed2, wed3, satisfy1, satisfy2, I(log(age))* в наборе данных 14-ой волны исследования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | -0.52461 | 0.10330 | -5.079 | 4.27e-07 | \*\*\* |
| workweek | 0.13394 | 0.02241 | 5.978 | 2.81e-09 | \*\*\* |
| age | -0.29650 | 0.05500 | -5.391 | 8.11e-08 | \*\*\* |
| sex | 0.42981 | 0.04558 | 9.430 | <2e-16 | \*\*\* |
| j\_educ | 0.46555 | 0.04939 | 9.425 | <2e-16 | \*\*\* |
| status2 | 0.40633 | 0.04684 | 8.675 | <2e-16 | \*\*\* |
| wed1 | 0.11759 | 0.07765 | 1.514 | 0.130167 |  |
| wed2 | 0.02551 | 0.08824 | 0.289 | 0.772546 |  |
| wed3 | 0.03486 | 0.13737 | 0.254 | 0.799683 |  |
| satisfy1 | 0.17889 | 0.05362 | 3.336 | 0.000869 | \*\*\* |
| satisfy2 | -0.10375 | 0.05590 | -1.856 | 0.063661 | . |
| I(log(age)) | 0.01848 | 0.02324 | 0.795 | 0.426639 |  |

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

Хочу сравнить три модели на статус лучшей модели из всех рассмотренных мною. Первая модель – изначальная, вторая – это последняя модель в степенях, третья – это лучшая модель в логарифмах, а также рассмотреть возможность избавления от ненужных регрессоров.

У первой модели R2 = 18.27%, значимо больше половины регрессоров. У второй модели R2 = 21.82%, что больше, чем у предыдущей модели. Все значения VIF у данных регрессий приемлемы. У третий модели R2 = 21.8 %, значимо примерно половина регрессоров. Здесь все значения VIF у данных регрессий также приемлемы.

Несмотря на то, что у второй модели R2 выше на 0.02, я принял решение взять модель 3, поскольку VIF в этой модели меньше, чем во второй модели, а также звездочек больше. Также было принято решение убрать из модели регрессор *satisfy2,* поскольку у него нет ни одной звездочки, да,R2 незначительно понизится, но такая модель сможет более полно описать имеющийся процесс. Таким образом, лучшей математической моделью из всех рассмотренных является модель *salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + I(age^(0.2)).* Для данной модели будем совершать 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 3.3.

Таблица 3.3. Характеристики модели зависимости *salary ~ workweek, age, sex, j\_educ, status2, wed1, wed2, wed3, satisfy1, I(age^(0.2))* в наборе данных 14-ой волны исследования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | -0.76851 | 0.17649 | -4.354 | 1.42e-05 | \*\*\* |
| workweek | 0.13286 | 0.02242 | 5.925 | 3.85e-09 | \*\*\* |
| age | -0.32407 | 0.07292 | -4.444 | 9.45e-06 | \*\*\* |
| sex | 0.43131 | 0.04562 | 9.455 | <2e-16 | \*\*\* |
| j\_educ | 0.46735 | 0.04943 | 9.456 | <2e-16 | \*\*\* |
| status2 | 0.40994 | 0.04685 | 8.750 | <2e-16 | \*\*\* |
| wed1 | 0.11723 | 0.07771 | 1.509 | 0.132 |  |
| wed2 | 0.02618 | 0.08828 | 0.297 | 0.767 |  |
| wed3 | 0.04127 | 0.13741 | 0.300 | 0.764 |  |
| satisfy1 | 0.23799 | 0.04317 | 5.513 | 4.14e-08 | \*\*\* |
| I(age^(0.2)) | 0.21140 | 0.22307 | 0.948 | 0.343 |  |

*workweek*: 0.13 - положительный

*age*: -0.32 - отрицательный

*sex*: 0.43 - положительный

*j\_educ*: 0.46 - положительный

*status2*: 0.41 - положительный

*satisfy1*: 0.24 – положительный

*I(age^(2))*: 0.21 – положительный

Вывод о том, какие индивиды получают большую зарплату: большую зарплату получают в большинстве своём более взрослые люди с продолжительной рабочей неделей, имеющие высшее образование, проживающие в городе и удовлетворённые своей заработной платой.

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

Выделим подмножество мужчин, разведённых, а также имеющих высшее образование с помощью применения функции subset трижды. Возьмём выбранную мной лучшую модель зависимости зарплаты от других параметров, и учтём, что находимся в подмножестве мужчин, разведённых, а также имеющих высшее образование (в этом подмножестве переменные *sex, j\_educ* и *wed2* равны единице). Таким образом, модель получается *salary ~ workweek + age + status2 + satisfy1 + I(age^(0.2))*, где R2 = 19.14%. К сожалению, VIF у регрессоров *age* и у *I(age^(0.2))* очень высокий, то есть существует зависимость между этими регрессорами*.* Значения коэффициентов модели, их стандартные ошибки, p-статистика и уровень значимости приведены в таблице 3.4.

Таблица 3.4. Характеристики модели зависимости *salary ~ workweek + age + status2 + satisfy1 + I(age^(0.2))* в наборе данных 14-ой волны исследования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | -6.9867 | 13.5157 | -0.517 | 0.627 |  |
| workweek | 0.2120 | 0.4656 | 0.455 | 0.668 |  |
| age | -5.1392 | 7.1625 | -0.718 | 0.505 |  |
| status2 | 0.5260 | 1.3511 | 0.389 | 0.713 |  |
| satisfy1 | 0.7910 | 1.6748 | 0.472 | 0.657 |  |
| I(age^(0.2)) | 12.1820 | 20.9519 | 0.581 | 0.586 |  |

Среди людей мужчин, разведённых, а также имеющих высшее образование, индивиды получают зарплату больше других, а также в этом подмножестве удовлетворенность условиями труда играет более важную роль (коэффициент равен 0.79).

Выделим подмножество женщин, живущих в городе и состоящих в браке с помощью применения функции subset трижды. Возьмём выбранную мной лучшую модель зависимости зарплаты от других параметров, и учтём, что находимся в подмножестве женщин, живущих в городе и состоящих в браке (в этом подмножестве переменные *status2, wed1* равны 1*, а sex* равна нулю). Таким образом, получаем модель *salary ~ workweek + age + j\_educ + satisfy1 + I(age^(0.2)),* где R2 = 14.45%. VIF хороший для всех значений. Значения коэффициентов модели, их стандартные ошибки, p-статистика и уровень значимости приведены в таблице 3.5.

Таблица 3.5. Характеристики модели зависимости *salary ~ workweek + age + j\_educ + satisfy1 + I(age^(0.2))* в наборе данных 14-ой волны исследования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр/Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень значимости |
| (Intercept) | -0.47765 | 0.30581 | -1.562 | 0.11927 |  |
| workweek | 0.19955 | 0.05040 | 3.959 | 9.24e-05 | \*\*\* |
| age | -0.44376 | 0.14780 | -3.002 | 0.00288 | \*\* |
| j\_educ | 0.51062 | 0.08810 | 5.796 | 1.60e-08 | \*\*\* |
| satisfy1 | 0.12097 | 0.08171 | 1.480 | 0.13970 |  |
| I(age^(0.2)) | 0.59690 | 0.43927 | 1.359 | 0.17513 |  |

Среди женщин, живущих в городе и состоящих в браке, всё те же индивиды получают зарплату больше других, а также в этом подмножестве огромную рол играет наличие высшего образования (коэффициент 0.51).

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

***Вывод:*** я проанализировала данные опроса НИУ ВШЭ о материальном состоянии граждан России в наборе данных 14-ой волны исследования. Мне требовалось найти некую зависимость между некоторыми параметрами из данных и зарплатой людей. Сначала я выбрала параметры, которые могли бы оказать существенное влияние на уровень зарплаты индивидов. Туда попали: зарплата, продолжительность рабочей недели, возраст, пол, наличие высшего образования, населённый пункт, семейное положение и удовлетворенность условиями труда. Для начала я построила линейную регрессию зарплаты на все выбранные параметры *salary ~ workweek, age, sex, j\_educ, status2, wed1, wed2, wed3, satisfy1, satisfy2*, где R2 =18.27%., vif у всех значений оказался ниже 3, то есть модель является достаточно плохой, несмотря на то что зависимости между регрессорами нет. Далее я попыталась улучшить эту модель добавлением логарифмов функций от вещественных переменных (возраст и продолжительность рабочей недели) и их степеней от 0.1 до 2. Для каждой модели я проверила R2, vif, и количество значимых регрессоров. Также среди логарифмов попыталась найти наилучшую модель, затем сравнив лучшие модели, я пришла к выводу, что наилучшей моделью является В итоге нашлась лучшая модель *salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + I(age^(0.2)).* Оценив коэффициенты перед переменными, я сделала выводы, какие индивиды получают большую зарплату. Также в конце задания мною были выделены 2 подмножества (мужчины, разведённые с высшим образованием; женщины, живущие в городе, состоящие в браке) на данных подмножествах мною были построены модели, после оценки которых, получилось сделать вывод, какие индивиды получают большую зарплату. Рост зарплаты зависит от продолжительности рабочей недели и удовлетворённости заработной платой у индивидов с высшим образованием.

## Задача 4

Набор данных: *Drug classification*

Тип классификатора: SVM (метод опорных векторов)

Классификация по столбцу: Drug Type (DrugA – 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, Normal – 1, High – 2;

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

А также обработаем столбец, по которому будем строить классификацию (*Drug Type*): 4 и выше – класс 0, ниже – класс 1. Этот столбец удалим из данных, оставив его отдельно от остальных столбцов. Его нужно убрать, чтобы наш классификатор смог обучаться. Первые пять строк изменённой таблицы приведены в таблице 4.2.

Таблица 4.2. Таблица, подготовленная для решения задачи классификации.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Age | Sex | BP | Cholesterol | Na\_to\_K |
| 0 | 23 | 0 | 2 | 1 | 25.355 |
| 1 | 47 | 1 | 0 | 1 | 13.093 |
| 2 | 47 | 1 | 0 | 1 | 10.114 |
| 3 | 28 | 0 | 1 | 1 | 7.798 |
| 4 | 61 | 0 | 0 | 1 | 18.043 |
| ... | ... | ... | ... | ... | ... |
| 195 | 56 | 0 | 0 | 1 | 11.567 |
| 196 | 16 | 1 | 0 | 1 | 12.006 |
| 197 | 52 | 1 | 1 | 1 | 9.894 |
| 198 | 23 | 1 | 1 | 0 | 14.020 |
| 199 | 40 | 0 | 0 | 0 | 11.349 |

Можем приступить к задаче классификации. Разделим набор данных на обучающую и тестовую выборку. Размер тестовой выборки будет 33%. Строим классификатор SGDClassifier (метод опорных векторов) для обучения, выставив штраф (l2), код представлен на листинге 4.1.



Листинг 4.1. Классификатор SGDClassifier.

После построения данной модели, она может быть использована для прогнозирования значений, код представлен на листинге 4.2.



Листинг 4.2. Вставка кода для прогнозирования модели.

Прогнозом будет являться число 0.8666666666666667. Модель настроена, можно давать ей тестовые данные, 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.

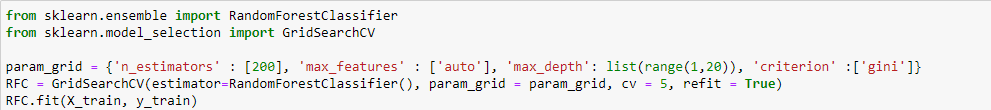
Таблица 4.3. Результат работы метода classification\_report на нашей тестовой выборке.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.87 | 1.0 | 0.93 | 52 |

Метрика accuracy равна 0.86.

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 отдельно так называемых деревьев глубины от 1 до 10. Далее на листинге 4.3 показан код построения 200 деревьев.



Листинг 4.3. Код построения классификатора Случайный лес 200 деревьев.

Оценим классификатор с помощью тех же самых метрик. Получим: accuracy: 0.95, f1: 0.9549407114624506, precision: 0.9318181818181819, recall: 1.0.

Далее для сравнения на литинге 4.4 показан код построения 300 деревьев.



Листинг 4.4. Код построения классификатора Случайный лес 300 деревьев.

Метрики которого имеют следующие значения: accuracy 0.9333333333333333,

f1: 0.9635987201204592, precision:0.8984848484848484, recall: 1.0.

Далее для сравнения на листинге 4.5 показан код построения 400 деревьев.



Листинг 4.5. Код построения классификатора Случайный лес 400 деревьев.

Метрики которого имеют следующие значения: accuracy 0.9333333333333333,

f1: 0.9722943722943723, precision:0.9318181818181819, recall: 1.0.

Вариант с 200-ми деревьями оказался немного лучше остальных, поэтому дальнейший анализ произведём вокруг него. Будем строить деревья с шагом 10 в обе стороны. Сразу становится ясно, что при увеличении количества деревьев метрики растут, а при уменьшении – падают. Так происходит до 260-ти деревьев, поскольку метрики начинают немного падать. Наилучший вариант с 200-ми деревьями является вариант с 220 деревьями, поэтому проанализируем классификатор Random Forest состоящий из 220-ти деревьев.

Метрики данного классификатора имеют следующие значения: accuracy 0.95,

f1: 0.9722943722943723, precision: 0.9136363636363637, recall: 1.0 Для сравнения с классификатором Метода опорных векторов будем использовать именно этот вариант классификатора Случайный лес.

Теперь сравним классификатор Метод опорных векторов (SVM) и Случайный Лес (Random Forest), оценивая метрики каждого из них. Метрики все метрики accuracy, f1, precision и recall у классификатора Random Forest лучше, следовательно, классификатор Random Forest для моего набора данных работает лучше и точнее, чем SVM.

***Вывод:*** мною была выполнена задача классификации для набора данных *Drug classification* двумя разными способами: Методом опорных векторов и Случайным Лесом. Требовалось разделить данные в столбце *Drug Type* по следующим критериям: 4 и выше – класс 0, ниже – класс 1. Для этого в начале я нормализовала все не числовые признаки в числовые или бинарные, затем разбила данные на тренировочную и тестовую выборки и построила классификатор SGDClassifier. Оценка точности данного классификатора была произведена с помощью метрик accuracy, f1, precision и recall. Получились следующие значения: accuracy: 0.87, f1: 0.93, precision: 0.87, recall: 1.0. После построения классификатора Random Forest я выбрала наилучший вариант с количеством деревьев и также оценила его с помощью четырёх метрик: accuracy 0.95, f1: 0.97, precision: 0.91, recall: 1.0. После сравнения метрик каждого из двух классификаторов я сделала вывод о том, что с моим набором данных лучше работает Случайный лес, чем Метод опорных векторов.

Задача 5

Набор данных: Denver Crime Data

Необходимо провести анализ датасета и сделать обработку данных, соответствуя алгоритму решения задачи. Ответить на следующие вопросы

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

Для начала следует показать, как примерно выглядят данные. Первые 5 строчек набора данных приведены в таблице 5.1.

Таблица 5.1. Первые 5 строк набора данных Denver Crime Data, данную таблицу можно получить с помощью метода data.head().

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| incident\_id | offense\_id | OFFENSE\_CODE | OFFENSE\_CODE\_EXTENSION | OFFENSE\_TYPE\_ID | OFFENSE\_CATEGORY\_ID | FIRST\_OCCURRENCE\_DATE | LAST\_OCCURRENCE\_DATE | REPORTED\_DATE |
| 20226000193 | 2,0226E+16 | 2999 | 0 | criminal-mischief-other | public-disorder | 01.04.2022 11:30 | 01.04.2022 12:00 | 01.04.2022 20:36 |
| 20223319 | 2,02233E+13 | 2999 | 0 | criminal-mischief-other | public-disorder | 01.03.2022 6:45 | NaN | 01.03.2022 11:01 |
| 20223093 | 2,02231E+13 | 2999 | 0 | criminal-mischief-other | public-disorder | 01.03.2022 1:00 | NaN | 01.03.2022 6:11 |
| 20224000 | 2,0224E+13 | 2999 | 0 | criminal-mischief-other | public-disorder | 01.03.2022 19:47 | NaN | 01.03.2022 21:12 |
| 20223956 | 2,0224E+13 | 2999 | 0 | criminal-mischief-other | public-disorder | 01.03.2022 17:06 | NaN | 01.03.2022 20:31 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INCIDENT\_ADDRESS | GEO\_X | GEO\_Y | GEO\_LON | GEO\_LAT | DISTRICT\_ID | PRECINCT\_ID | NEIGHBORHOOD\_ID | IS\_CRIME | IS\_TRAFFIC |
| 128 S CANOSA CT ... | 3135366.0 | 1685410.0 | -105.018825 | 39.714268 | 4.0 | 411.0 | valverde | 1 | 0 |
| 650 15TH ST ... | 3142454.0 | 1696151.0 | -104.993418 | 39.743649 | 6.0 | 611.0 | cbd | 1 | 0 |
| 919 E COLFAX AVE ... | 3147484.0 | 1694898.0 | -104.975557 | 39.740130 | 6.0 | 621.0 | north-capitol-hill | 1 | 0 |
| 2345 W ALAMEDA AVE ... | 3136478.0 | 1684414.0 | -105.014892 | 39.711518 | 4.0 | 411.0 | valverde | 1 | 0 |
| 7800 E SMITH RD ... | 3169237.0 | 1705800.0 | -104.897950 | 39.769688 | 5.0 | 512.0 | central-park | 1 | 0 |

Для того, чтобы получить количество объектов и признаков воспользуемся методом data.shape. Получим, что таблица содержит 486886 строк и 19 столбцов, соответственно 486886 объектов и 19 признаков, которым ниже дано описание.

Описание признаков:

* *incident\_id* – идентификатор случая или происшествия;
* *offense\_id* – идентификатор преступления;
* *OFFENSE\_CODE* – код конкретного преступления;
* *OFFENSE\_CODE\_EXTENSION* – расширение кода преступления;
* *OFFENSE\_TYPE\_ID* – идентификатор типа преступления;
* *OFFENSE\_CATEGORY\_ID* – идентификатор категории преступления;
* *FIRST\_OCCURRENCE\_DATE* – дата первого происшествия;
* *LAST\_OCCURRENCE\_DATE* – дата последнего происшествия;
* *REPORTED\_DATE* – зарегистрированная дата преступления;
* *INCIDENT\_ADDRESS* – адрес происшествия;
* *GEO\_X* – долгота по X;
* *GEO\_Y* – широта по Y;
* *GEO\_LON –* долгота;
* *GEO\_LAT –* широта;
* *DISTRICT\_ID* – идентификатор дистрикта;
* *PRECINCT\_ID* – идентификатор участка;
* *NEIGHBORHOOD\_ID* – идентификатор района;
* *IS\_CRIME* – наличие преступления;
* *IS\_TRAFFIC* – наличие движения.

1. Сколько категориальных признаков, какие?

К категориальным признакам я бы отнесла 2 следующих признака:

– идентификатор дистрикта (округа), столбец *DISTRICT\_ID,* поскольку этот признак имеет целые значения от 1 до 7, в связи с тем можно будет узнать, в каких районах было совершено наибольшее число преступлений;

– расширение кода преступления, столбец *OFFENSE\_CODE\_EXTENSION*, данный параметр содержит в себе целые числа от 0 до 5, благодаря ему можно будет рассмотреть подробно, какое конкретно преступление было совершено, то есть категориально разделить его на степени тяжести.

1. Столбец с максимальным количеством уникальных значений категориального признака?

Столбцом с максимальным количеством уникальных значений категориального признака является столбец *DISTRICT\_ID,* так какон содержит в себе целые числа от 1 до 7, что является наибольшим значением по сравнению с *OFFENSE\_CODE\_EXTENSION,* который может приниматьлишь 6 целых значений, подробнее, какие значения принимает параметр *DISTRICT\_ID* можно увидеть на рисунке 5.1.

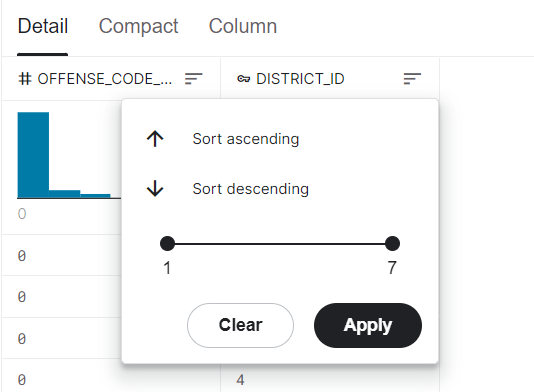


Рисунок 5.1. Варианты возможных значений параметра *DISTRICT\_ID*

1. Есть ли бинарные признаки?

В наборе данных Denver Crime Data присутствует 2 бинарных признака: *IS\_CRIME* и *IS\_TRAFFIC*, поскольку эти признаки принимают значения либо 0, либо 1. Подробнее это можно рассмотреть на рисунке 5.2:

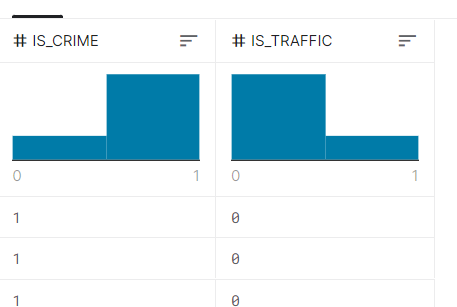


Рисунок 5.2. Бинарные признаки: *IS\_CRIME* и *IS\_TRAFFIC*

1. Какие числовые признаки?

Числовыми признаками набора данных являются 12 признаков: *incident\_id , offense\_id, OFFENSE\_CODE, OFFENSE\_CODE\_EXTENSION, GEO\_X, GEO\_Y, GEO\_LON, GEO\_LAT, DISTRICT\_ID, PRECINCT\_ID, IS\_CRIME, IS\_TRAFFIC.* Эти параметры принимают различные числовые значения, пример представлен на рисунке 5.3.

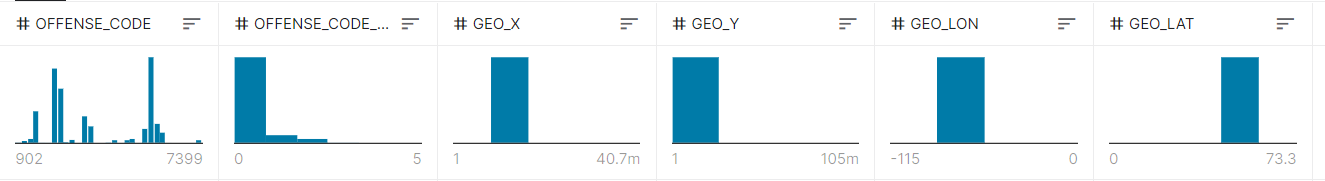


Рисунок 5.3. Наличие числовых значений у признаков *OFFENSE\_CODE, OFFENSE\_CODE\_EXTENSION, GEO\_X, GEO\_Y, GEO\_LON, GEO\_LAT*

1. Есть ли пропуски?

Чтобы ответить на этот вопрос, воспользуемся методом data.info(). Он выводит информацию о наборе данных (сколько ненулевых значений у каждого признака и какого типа значения присутствуют в признаке).

Результат работы метода представлен в таблице 5.2.

Таблица 5.2. Результат работы метода data.info() для набора данных Denver Crime Data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Non-Null | Count | Dtype |  |
| ------ | -------------- | ----- |  |  |
| 0 | incident\_id | 486886 | non-null | int64 |
| 1 | offense\_id | 486886 | non-null | int64 |
| 2 | OFFENSE\_CODE | 486886 | non-null | int64 |
| 3 | OFFENSE\_CODE\_EXTENSION | 486886 | non-null | int64 |
| 4 | OFFENSE\_TYPE\_ID | 486886 | non-null | object |
| 5 | OFFENSE\_CATEGORY\_ID | 486886 | non-null | object |
| 6 | FIRST\_OCCURRENCE\_DATE | 486886 | non-null | object |
| 7 | LAST\_OCCURRENCE\_DATE | 182967 | non-null | object |
| 8 | REPORTED\_DATE | 486886 | non-null | object |
| 9 | INCIDENT\_ADDRESS | 447985 | non-null | object |
| 10 | GEO\_X | 482707 | non-null | float64 |
| 11 | GEO\_Y | 482707 | non-null | float64 |
| 12 | GEO\_LON | 482706 | non-null | float64 |
| 13 | GEO\_LAT | 482706 | non-null | float64 |
| 14 | DISTRICT\_ID | 486885 | non-null | float64 |
| 15 | PRECINCT\_ID | 486885 | non-null | float64 |
| 16 | NEIGHBORHOOD\_ID | 486885 | non-null | object |
| 17 | IS\_CRIME | 486886 | non-null | int64 |
| 18 | IS\_TRAFFIC | 486886 | non-null | int64 |

Из таблицы видно, что в следующих 9 признаках отсутствуют некоторые значения: *LAST\_OCCURRENCE\_DATE, INCIDENT\_ADDRESS, GEO\_X, GEO\_Y, GEO\_LON, GEO\_LAT, DISTRICT\_ID, PRECINCT\_ID, NEIGHBORHOOD\_ID.* Поскольку всего 486886 объектов, то именно в этих параметрах имеются пропуски. Это значит, что в наборе данных есть пропуски и нам необходимо удалить столбцы или строчки с пропусками. Код для удаления пропусков представлен на листинге 5.1.



Листинг 5.1. Удаление строк с отсутствующими значениями

Таким образом, после удаления отсутствующих значений, мы получили 175634 объектов.

1. Сколько объектов с пропусками?

Опираясь на предыдущую таблицу 5.2., с помощью математических вычислений посчитаем количество пропусков для каждого признака:

*LAST\_OCCURRENCE\_DATE –* 303919 пропусков;

*INCIDENT\_ADDRESS –* 38901 пропусков;

*GEO\_X –* 4179 пропусков;

*GEO\_Y –* 4179 пропусков;

*GEO\_LON –* 4180 пропусков;

*GEO\_LAT –* 4180 пропусков;

*DISTRICT\_ID –* 1 пропуск;

*PRECINCT\_ID –* 1 пропуск;

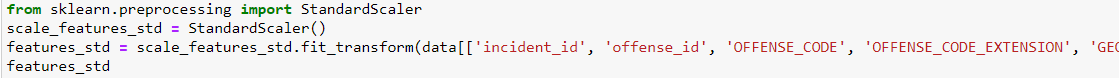
*NEIGHBORHOOD\_ID –* 1 пропуск.

1. Столбец с максимальным количеством пропусков?

Рассматривая подробнее таблицу 5.2., можно отметить, что столбец *LAST\_OCCURRENCE\_DATE* имеет наибольшее количество пропусков, всего в этом столбце 303919 пропусков.

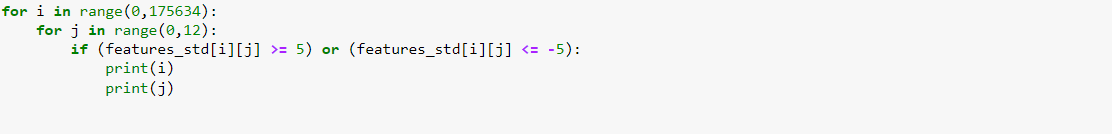
1. Есть ли на ваш взгляд выбросы, аномальные значения?

К ответу на данный вопрос я решила подойти со стороны величин стандартных отклонений каждого из значений набора данных. Я создала массив массивов features\_std, который как раз и содержит стандартные отклонения каждого элемента датасета.



Листинг 5.2. Нормализация признаков через стандартное отклонение

При прохождении по этому массиву будем считать, что значение аномально, если его стандартное отклонение больше 5-ти, выбрала такое значение, поскольку количество аномальных значений в данном наборе данных очень много (листинг 5.3). После окончания работы цикла получаем, что аномальные значения в достаточном количестве присутствуют в признаках *incident\_id, offense\_id, GEO\_X, GEO\_Y, GEO\_LON, GEO\_LAT*. В остальных же столбцах аномальных отклонений нет (или встречается 1-2 штуки).



Листинг 5.3. Поиск аномальных значений, если его стандартное отклонение >= 5 по модулю

1. Столбец с максимальным средним значением после нормировки признаков через стандартное отклонение?

Для нахождения такого столбца также воспользуемся массивом массивов features\_std, посчитаем сумму всех элементов каждого столбца и поделим на количество объектов (на 175634). Таким образом, прономеруем признаки через стандартное отклонение, и найдём столбец с самым большим средним значением (результат работы кода имеется на рисунке 5.8). Этому номеру признака соответствует столбец PRECINCT\_ID, его среднее значение после нормировки через стандартное отклонение равно -9.374781353132529e-14.

1. Столбец с целевым признаком?

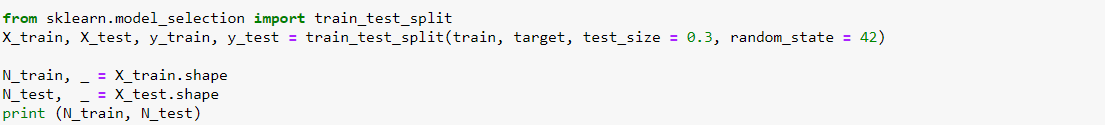
Столбцом с целевым признаком будет являться столбец *OFFENSE\_CODE*, в котором содержится код о конкретном преступлении. Выделение целевого признака представлено ниже (листинг 5.4.).



Листинг 5.4. Выделение целевого признака

1. Сколько объектов попадает в тренировочную выборку при использовании train\_test\_split с параметрами test\_size = 0.3, random\_state = 42?

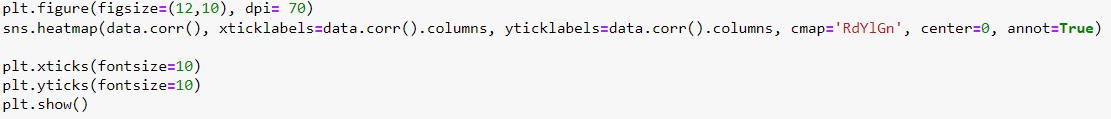
В тренировочную выборку с такими параметрами попадёт 122943 объектов. Это можно узнать посредством вывода параметра N\_train = X\_train.shape.



Листинг 5.5. Выделение тренировочной и тестовой выборки, где y - целевая переменная (target)

1. Между какими признаками наблюдается линейная зависимость (корреляция)?

Посмотреть на корреляцию между признаками можно с помощью метода data.corr(), но для наглядности лучше вывести температурную таблицу корреляции признаков. Сама же таблица представлена на рисунке 5.4. чем темнее клетка таблицы, тем выше зависимость признаков, на пересечении которых находится клетка.



Листинг 5.6. Код корреляция признаков по температурной таблице

Опираясь на рисунок 5.4, выделим достаточно сильные линейные зависимости признаков:

* Явно между собой отрицательно коррелируют признаки *IS\_CRIME* и *IS\_TRAFFIC*;
* Также отрицательно зависимы между собой *GEO\_LAT* с *GEO\_X*, *GEO\_Y, GEO\_LON, GEO\_LAT*;
* Признаки *IS\_CRIME* и *OFFENSE\_CODE* тоже отрицательно коррелируют друг с другом;
* Сильная положительная линейная зависимость присутствует между *incident\_id* и *offense\_id;*
* Еще сильная положительная линейная зависимость присутствует *DISTRICT\_ID* и *PRECINCT\_ID;*
* Такая же сильная положительная корреляция наблюдается между *GEO\_X* c *GEO\_Y, GEO\_LON, GEO\_LAT*;
* Еще положительно зависимы между собой *IS\_TRAFFIC* и *OFFENSE\_CODE.*

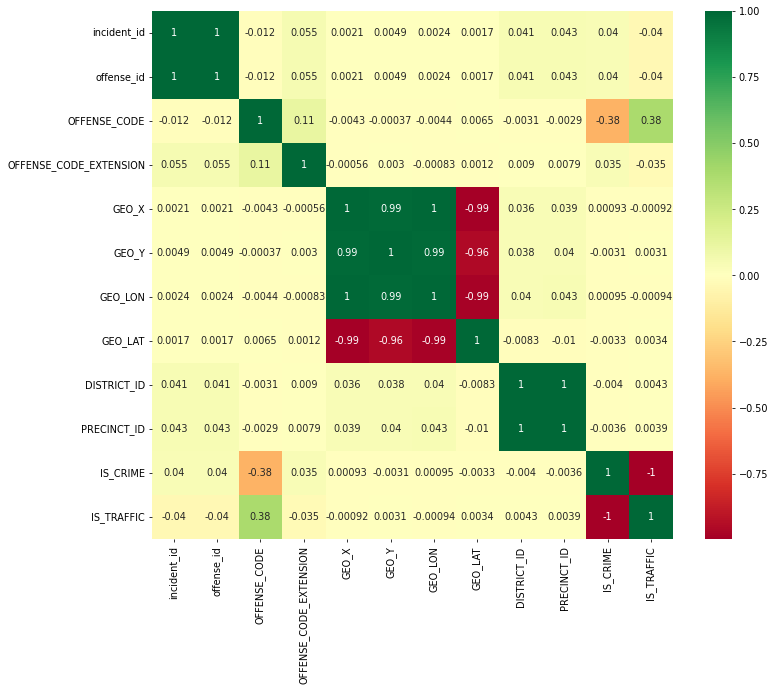
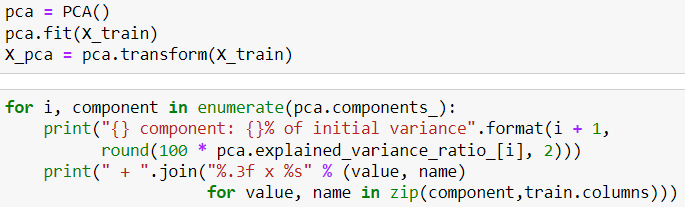


Рисунок 5.4. Температурная таблица корреляции признаков набора Denver Crime Data.

1. Сколько признаков достаточно для объяснения 90% дисперсии после применения метода PCA?

Для начала, конечно, выделим целевую переменную *OFFENSE\_CODE*, тренировочную и тестовую выборки и применим сам метод РСА на набор данных, в котором все значения изменились на величину их стандартных отклонений. Это нужно для того, чтобы все признаки были примерно в одном рационе, чтобы метод РСА работал верно. Код применения метода РСА представлен на листинге 5.7.



Листинг 5.7. Код применения метода РСА на тестовую выборку с целевым признаком *OFFENSE\_CODE*.

Результат работы метода показал, что 1 признак объясняет 37,7% дисперсии, 2 признака объясняют 20,87% дисперсии, 3 признака 16.69%, 4 признака объясняют 16.23% дисперсии. Значит, четырёх признаков хватит для объяснения 90% дисперсии после применения метода PCA.

Также покажем график зависимости доли объяснённой дисперсии от числа компонент на рисунке 5.5.

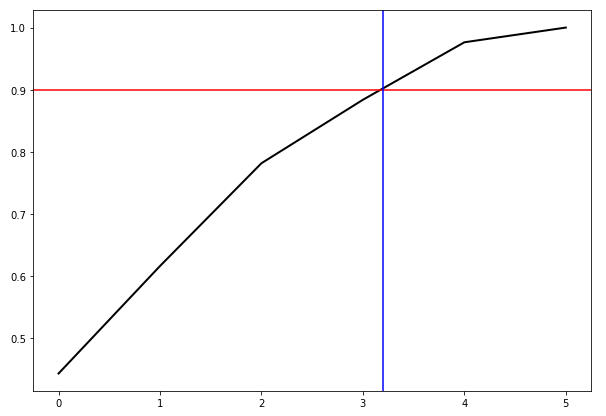


Рисунок 5.5. График зависимости доли объяснённой дисперсии от числа компонент.

Из графика видно, что для объяснения 90% дисперсии действительно понадобится 4 признака.

1. Какой признак вносит наибольший вклад в первую компоненту?

Посмотрим на распределение вклада признаков в первую компоненту (рисунок 5.14). 0.002 x incident\_id + 0.002 x offense\_id + 0.000 x OFFENSE\_CODE\_EXTENSION + 0.502 x GEO\_X + 0.496 x GEO\_Y + 0.502 x GEO\_LON + -0.497 x GEO\_LAT + 0.027 x DISTRICT\_ID + 0.028 x PRECINCT\_ID + 0.001 x IS\_CRIME + -0.001 x IS\_TRAFFIC.Самые большие коэффициенты у признаков *GEO\_X, GEO\_Y, GEO\_LON, GEO\_LAT*, а значит, именно эти признаки вносят наибольший вклад в первую компоненту.

1. Построить двухмерное представление данных с помощью алгоритма tSNE. На сколько кластеров визуально на ваш взгляд разделяется выборка?

Как видно из двумерного представления (Рисунок 5.6) алгоритмом t-SNE, выборка не поддается кластеризации, проверили эту выборку с помощью алгоритма t-SNE со значением параметра random\_state = 13.

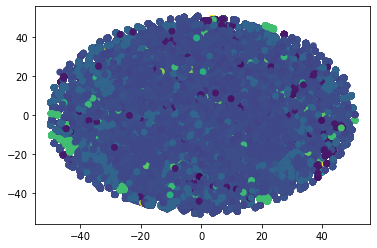


Рисунок 5.6. Двухмерное представление данных с помощью алгоритма t-SNE (вариант 1).

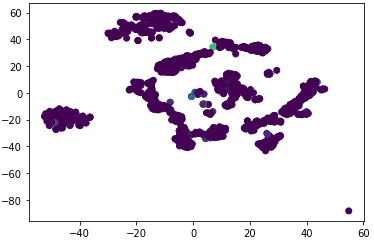
Было принято решение убрать все ID, поскольку они меняются сильнее всего. После этого также был применен алгоритм t-SNE. Результаты, которого представлены на рисунке 5.7. Таким образом, и в первом, и во втором случае выборка все также не поддается кластеризации. 

Рисунок 5.7. Двухмерное представление данных с помощью алгоритма t-SNE (вариант 2).

Код решения задачи приведён в Приложении 5.

***Вывод:*** мне нужно было провести первичный анализ данных набора Denver Crime Data. В начале я изучила количество объектов, признаки и виды этих признаков. Оказалось, что в этом наборе данных присутствуют как числовые, так и не числовые признаки. Также были замечены и 2 бинарных признака. Я обнаружила, что в датасете достаточно много пропусков, поэтому столбцы и строки с отсутствующими значениями пришлось удалить. Далее я нашла выбросы или аномальные значения, таких признаков оказалось весьма много, что говорит о том, что могли быть допущены ошибки в измерениях либо был произведен некорректный ввод данных. Затем была произведена нормализация данных через СКО для того, чтобы значения всех признаков находились в одном диапазоне. Дальше я проверила столбцы на корреляцию с помощью температурной таблицы. Она показала, что между разными столбцами есть достаточно много сильных зависимостей. В конце решения задачи я применил метод PCA, который показал, что для объяснения 90% дисперсии нужно 4 признака, а с помощью алгоритма t-SNE, я сделала вывод, что выборка не поддается кластеризации.

# 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.
2. В задаче №2 я строила большое количество (около 30-ти) линейных зависимостей одной переменной от трёх других, одновременно проверяя регрессоры моделей на независимость. Выделила из регрессий лучшую по доле объяснения данных набора Seatbelts. Для неё я нашла доверительные интервалы для всех коэффициентов, а далее рассчитала доверительный интервал для одного прогноза.
3. 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.
4. В задаче №4 мне удалось построить классификации двумя разными способами: Методом опорных векторов и Случайным Лесом, а также оценить их точности с помощью разных метрик (accuracy, F1, recall и precision). В конце сравнила два этих разных классификатора, Случайный Лес оказался лучше.
5. В задаче №5 я провела достаточно подробный первичный анализ данных набора Denver Crime Data. Проверила его на наличие пропусков, изучила виды признаков, представленные в нём, нашла столбцы с аномальными выбросами, проверила признаки на корреляцию между собой. Конечно, мною был испробован метод РСА для поиска более важных признаков.

# 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

#Вычисление среднего значения для Examination, Education, Agriculture

mean(data$Examination)

mean(data$Education)

mean(data$Agriculture)

#Вычисление дисперсии для Examination, Education, Agriculture

var(data$Examination)

var(data$Education)

var(data$Agriculture)

#вычисление СКО для Examination, Education, Agriculture

sqrt(var(data$Examination))

sqrt(var(data$Education))

sqrt(var(data$Agriculture))

#элементы нормализации

data["Examination\_1"]=data$Examination-mean(data$Examination)

data["Education\_1"]=data$Education-mean(data$Education)

data["Agriculture\_1"]=data$Agriculture-mean(data$Agriculture)

#предобработка данных

data["Examination\_2"]=( Examination\_1) /sqrt(var(data$Examination))

plot(data$Examination\_2)+ abline(a=0,b=0,col="red")

data["Education\_2"]=( Education\_1)/sqrt(var(data$Education))

plot(data$Education\_2)+ abline(a=0,b=0,col="red")

data["Agriculture\_2"]=( Agriculture\_1)/sqrt(var(data$Agriculture))

plot(data$Agriculture\_2)+ abline(a=0,b=0,col="red")

#Построение зависимости вида y = a\*x + b, где у - Examination (объясняемая переменная), х - Education (регрессор)

model1 = lm(Examination~Education, data)

model1

summary(model1)

#R^2 = 48.8%, значит, модель относительно неплоха, зависимость имеется

#значение p-статистики <0.001, также имеется 3 звездочки, поэтому взаимосвязь сильная

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

#Построение зависимости вида y = a\*x + b, где у - Examination, х - Agriculture

model2 = lm(Examination~Agriculture, data)

model2

summary(model2)

#R^2 = 47.6%, значит, модель относительно неплоха, зависимость имеется

#значение p-статистики <0.001, также имеется 3 звездочки, поэтому взаимосвязь сильная

plot(data$Examination~data$Agriculture) + abline(a = 28.7, b = -0.24, col = "red")

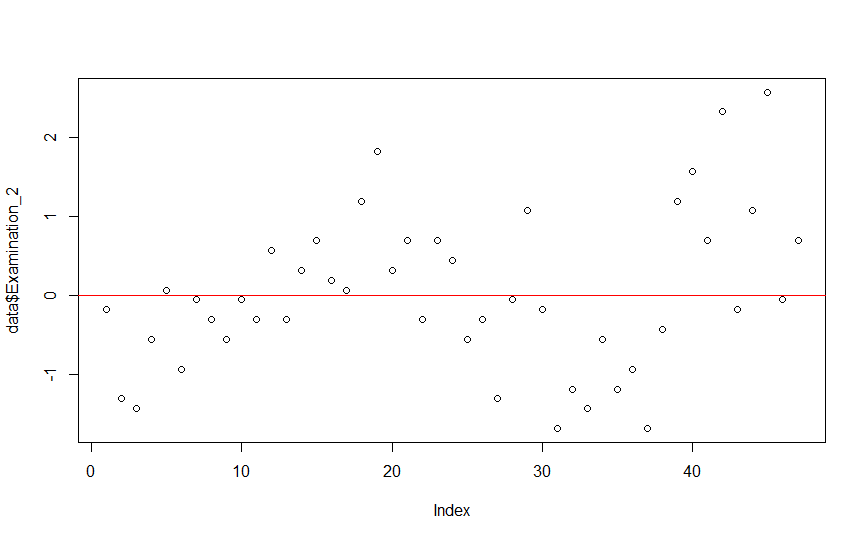


Рисунок 1.1. Результат работы команды plot(data$Examination\_2) + abline(a=0, b=0, col="red") – отклонения значений переменной *Examination* от своего среднего значения.

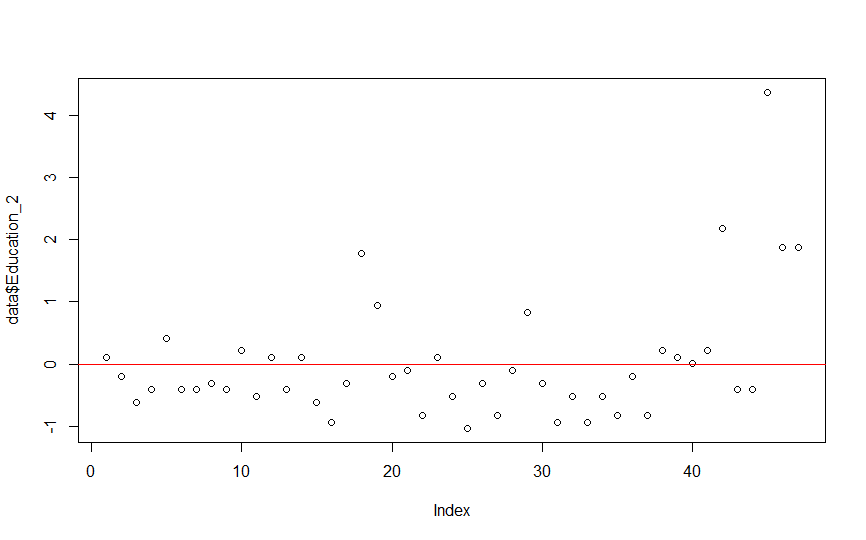


Рисунок 1.2. Результат работы команды plot(data$Education\_2)+ abline(a=0,b=0,col="red") – отклонения значений переменной *Education* от своего среднего значения.

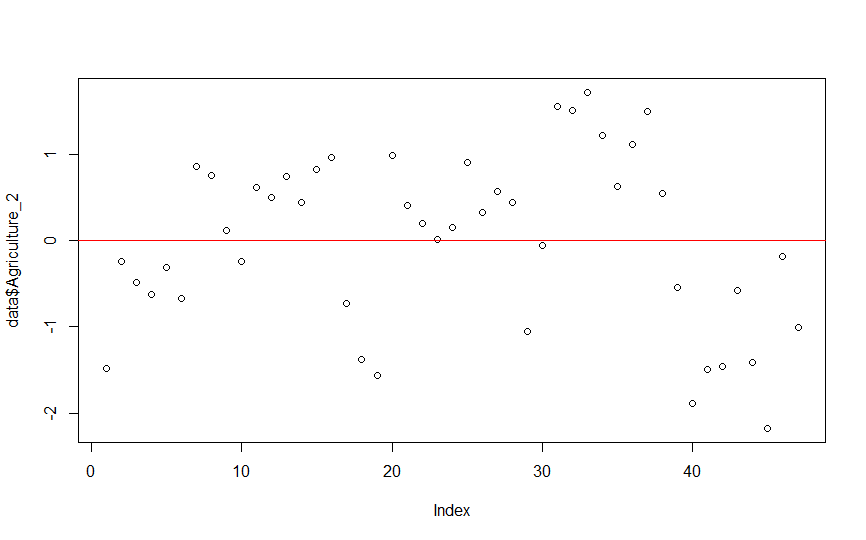


Рисунок 1.3. Результат работы команды plot(data$Agriculture\_2)+ abline(a=0,b=0,col="red") – отклонения значений переменной *Agriculture* от своего среднего значения.

## Приложение 2

library("lmtest")

library("GGally")

library("car")

data = Seatbelts

help(Seatbelts)

data

summary(data)

#построим зависимость, включающую несколько регрессоров, и посмотрим, сможем ли мы объединить их в одну зависимость

model0 = lm(DriversKilled~law+kms+PetrolPrice, data)

model0

summary(model0)

vif(model0)

#vif у всех регрессоров < 2 - связи между регрессорами нет

#R-squared: 0.201

#p-value: 3.478e-09

#возможно law или kms лучше исключить из модели

model1 = lm(DriversKilled~law+kms, data)

model1

summary(model1)

#R-squared: 0.1416, зависимости нет

model2 = lm(DriversKilled~law+PetrolPrice, data)

model2

summary(model2)

#R-squared: 0.1866, зависимости нет

model3 = lm(DriversKilled~kms+PetrolPrice, data)

model3

summary(model3)

#R-squared: 0.1844, зависимости нет

model4 = lm(law~kms+PetrolPrice, data)

model4

summary(model4)

vif(model4)

#vif < 2, значит, сильной зависимости среди регрессоров нет

model5 = lm(kms~law+PetrolPrice, data)

model5

summary(model5)

vif(model5)

#vif < 2, значит, сильной зависимости среди регрессоров нет

model6 = lm(PetrolPrice~law+kms, data)

model6

summary(model6)

vif(model6)

#vif < 2, значит, сильной зависимости среди регрессоров нет

#введение в модели функций регрессоров

model7 = lm(DriversKilled~law + kms + PetrolPrice + I(log(PetrolPrice)) + I(log(kms)), data)

model7

summary(model7)

vif(model7)

#R-squared: 0.2205

#Есть точка только у I(log(PetrolPrice))

#vif у law < 2, а у остальных регрессоров очень большой

model8 = lm(DriversKilled~law + kms + I(log(PetrolPrice)) + I(log(kms)), data)

model8

summary(model8)

vif(model8)

#R-squared: 0.2099

#Есть 3 звездочки только у I(log(PetrolPrice))

#vif у law и I(log(PetrolPrice)) < 2, а у остальных регрессоров большой

model9 = lm(DriversKilled~law + I(log(PetrolPrice)) + I(log(kms)), data)

model9

summary(model9)

vif(model9)

#R-squared: 0.202

#Есть 3 звездочки у I(log(PetrolPrice)) и 1 звездочка у law

#vif у всех регрессоров < 2

model10 = lm(DriversKilled~law + kms + PetrolPrice + I(log(PetrolPrice)), data)

model10

summary(model10)

vif(model10)

#R-squared: 0.2158

#Есть точка только у I(log(PetrolPrice)) и по 1 звездочке law и kms

#vif у law и kms < 2, а у остальных регрессоров очень большой

model11 = lm(DriversKilled~law + kms + PetrolPrice + I(log(kms)), data)

model11

summary(model11)

vif(model11)

#R-squared: 0.2067

#Есть 3 звездочки у PetrolPrice

#vif у law и PetrolPrice < 2, а у остальных регрессоров очень большой

#лучшая модель с логарифмом

model12 = lm(DriversKilled~law + kms + I(log(PetrolPrice)), data)

model12

summary(model12)

vif(model12)

#R-squared: 0.2044

#Есть 3 звездочки у I(log(PetrolPrice))

#vif у всех регрессоров < 2, значит, зависимости нет

model13 = lm(DriversKilled~law + PetrolPrice + I(log(kms)), data)

model13

summary(model13)

vif(model13)

#R-squared: 0.1986

#Есть 3 звездочки PetrolPrice и 1 звездочка law

#vif у всех регрессоров < 2, значит, зависимости нет

model14 = lm(DriversKilled~law + kms + PetrolPrice + I(PetrolPrice^2) + I(kms^2) + I(PetrolPrice\*kms) + I(PetrolPrice\*law) + I(law\*kms), data)

model14

summary(model14)

vif(model14)

#R-squared: 0.2488

#vif у всех регрессоров очень большой - связь между регрессорами есть

model15 = lm(DriversKilled~kms + PetrolPrice + I(PetrolPrice^2) + I(kms^2) + I(PetrolPrice\*kms) + I(PetrolPrice\*law) + I(law\*kms), data)

model15

summary(model15)

vif(model15)

#R-squared: 0.2466

#vif у всех регрессоров очень большой - связь между регрессорами есть

model16 = lm(DriversKilled~kms + I(PetrolPrice^2) + I(kms^2) + I(PetrolPrice\*kms) + I(PetrolPrice\*law) + I(law\*kms), data)

model16

summary(model16)

vif(model16)

#R-squared: 0.207

#vif у всех регрессоров очень большой - связь между регрессорами есть

model17 = lm(DriversKilled~kms + I(PetrolPrice^2) + I(kms^2) + I(PetrolPrice\*law) + I(law\*kms), data)

model17

summary(model17)

vif(model17)

#R-squared: 0.2017

#vif у I(PetrolPrice^2) < 2, а у всех остальных регрессоров очень большой - связь между регрессорами есть

model18 = lm(DriversKilled~kms + I(PetrolPrice^2) + I(kms^2) + I(PetrolPrice\*law), data)

model18

summary(model18)

vif(model18)

#R-squared: 0.2015

#vif у I(PetrolPrice^2) и I(PetrolPrice \* law) < 2, а у всех остальных регрессоров очень большой - связь между регрессорами есть

model19 = lm(DriversKilled~kms + I(PetrolPrice^2) + I(PetrolPrice\*law), data)

model19

summary(model19)

vif(model19)

#R-squared: 0.1972

#Есть 3 звездочки у I(PetrolPrice^2) и 1 звездочка у I(PetrolPrice \* law)

#vif у регрессоров < 2

model20 = lm(DriversKilled~law + kms + PetrolPrice + I(PetrolPrice^2), data)

model20

summary(model20)

vif(model20)

#R-squared: 0.2149

#vif у law и kms < 2, у всех остальных регрессоров очень большой

model21 = lm(DriversKilled~law + kms + PetrolPrice + I(kms^2), data)

model21

summary(model21)

vif(model21)

#R-squared: 0.2051

#vif у law и PetrolPrice < 2, у всех остальных регрессоров очень большой

model22 = lm(DriversKilled~law + kms + PetrolPrice + I(kms\*law), data)

model22

summary(model22)

vif(model22)

#R-squared: 0.2032

#vif у kms и PetrolPrice < 2, у всех остальных регрессоров очень большой

model23 = lm(DriversKilled~law + kms + PetrolPrice + I(kms\*PetrolPrice), data)

model23

summary(model23)

vif(model23)

#R-squared: 0.211

#vif у kms < 2, у всех остальных регрессоров очень большой

model24 = lm(DriversKilled~law + kms + PetrolPrice + I(law\*PetrolPrice), data)

model24

summary(model24)

vif(model24)

#R-squared: 0.201

#vif у kms и PetrolPrice < 2, у всех остальных регрессоров очень большой

model25 = lm(DriversKilled~law + kms + I(PetrolPrice^2), data)

model25

summary(model25)

vif(model25)

#R-squared: 0.1971

#vif у всех регрессоров < 2, значит, зависимости нет

#лучшая модель с введёнными произведениями пар регрессоров

model25 = lm(DriversKilled~law + PetrolPrice + I(kms^2), data)

model25

summary(model25)

vif(model25)

#R-squared: 0.2027

#vif у всех регрессоров < 2, значит, зависимости нет

model26 = lm(DriversKilled~kms + PetrolPrice + I(law^2), data)

model26

summary(model26)

vif(model26)

#R-squared: 0.201

#vif у всех регрессоров < 2, значит, зависимости нет

#сравним лучшие модели

model0 = lm(DriversKilled~law+kms+PetrolPrice, data)

model0

summary(model0)

vif(model0)

#R-squared: 0.201

#Есть 3 звездочки у PetrolPrice, 1 звездочка у law и точка kms

#vif у всех регрессоров < 2, значит, зависимости нет

model12 = lm(DriversKilled~law + kms + I(log(PetrolPrice)), data)

model12

summary(model12)

vif(model12)

#R-squared: 0.2044

#Есть 3 звездочки у I(log(PetrolPrice)), по 1 точке у law и kms

#vif у всех регрессоров < 2, значит, зависимости нет

model25 = lm(DriversKilled~law + PetrolPrice + I(kms^2), data)

model25

summary(model25)

vif(model25)

#R-squared: 0.2027

#Есть 3 звездочки у PetrolPrice, по 1 точке у law и I(kms^2)

#vif у всех регрессоров < 2, значит, зависимости нет

#лучшая модель из всех

model12 = lm(DriversKilled~law + kms + I(log(PetrolPrice)), data)

model12

summary(model12)

vif(model12)

t\_critical=qt(0.975,df=125-4)

t\_critical

#law: B = -11.83; o = 6.006; t = 1.979764

#kms: B = -0.001; o = 0.0006; t = 1.979764

#I(log(PetrolPrice)): B = -59.14; o = 15.35; t = 1.979764

#Доверительный интервал law:

#B:[-11.83-(1.98\*6), -11.83+(1.98\*6)]

#B:[-23.71, 0.05]

#0 попадает в доверительный интервал

#этот коэффициент может быть равен 0, значит, регрессор практически не связан с обьясняемой переменной

#Доверительный интервал kms:

#B:[-0.001-(1.98\*0), -0.001+(1.98\*0)]

#B:[-0.001, 0.001]

#0 попадает в доверительный интервал

#этот коэффициент может быть равен 0, значит, регрессор практически не связан с обьясняемой переменной

#Доверительный интервал I(log(PetrolPrice)):

#B:[-59.14-(1.98\*15.35), -59.14+(1.98\*15.35)]

#B:[-89.533, -28.747]

#0 не попадает в доверительный интервал

#этот не коэффициент может быть равен 0

#[y-(Residual standard error: 22.82\*1.98), y+(Residual standard error: 22.82\*1.98)]

new.data = data.frame(law = 20, kms = 10, PetrolPrice = 10)

predict(model12, new.data, interval = "confidence")

#прогноз модели -364.4884

#[-617.2967, -111.6801]

Приложение 3

library("lmtest")

library("dplyr")

library("GGally")

library("car")

library("sandwich")

library("rlms")

data <- rlms\_read("r14i\_os26b.sav")

glimpse(data2)

data2=select(data, jj13.2, jh5, j\_marst, j\_diplom, status, jj6.2, j\_age, j\_educ, jj1.1.2 )

data2= na.omit(data2)

#элементы нормализации

#зарплата

data2$jj13.2

sal = as.numeric(data2$jj13.2)

sal1 = as.character(data2$jj13.2)

sal2 = lapply(sal1, as.integer)

sal = as.numeric(unlist(sal2))

mean(sal)

data2["salary"]=(sal-mean(sal))/(sqrt(var(sal)))

data2["salary"]

#продолжительность рабочей недели

data2$jj6.2

work = as.numeric(data2$jj6.2)

work1 = as.character(data2$jj6.2)

work2 = lapply(work1, as.integer)

work = as.numeric(unlist(work2))

mean(work)

data2["workweek"]=(work-mean(work))/(sqrt(var(work)))

data2["workweek"]

#возраст

data2$j\_age

age = as.numeric(data2$j\_age)

age1 = as.character(data2$j\_age)

age2 = lapply(age1, as.integer)

age = as.numeric(unlist(age2))

mean(age)

data2["age"]=(age-mean(age))/(sqrt(var(age)))

data2["age"]

#пол

data2["sex"]=data2$jh5

data2$sex = as.numeric(data2$sex)

data2$sex[which(data2$sex!='1')]<-0

data2["sex"]

#наличие высшего образования

data2["j\_educ"] = data2$j\_educ

data2$j\_educ = as.numeric(data2$j\_educ)

data2["j\_educ"] = lapply(data2$j\_educ, as.character)

data2$j\_educ[which(data2$j\_educ=='1')] <- 0

data2$j\_educ[which(data2$j\_educ=='21')] <- 1

data2$j\_educ[which(data2$j\_educ=='22')] <- 1

data2$j\_educ[which(data2$j\_educ=='23')] <- 1

data2$j\_educ[which(data2$j\_educ!='1')] <- 0

data2$j\_educ

#населенный пункт

data2["status2"]=data2$status

data2$status2 = as.numeric(data2$status2)

data2$status2[which(data2$status2=='1')] <- 1

data2$status2[which(data2$status2=='2')] <- 1

data2$status2[which(data2$status2!='1')] <- 0

data2["status2"]

#семейное положение

data2["wed1"]= data2$j\_marst

data2$wed1 = as.numeric(data2$wed1)

data2["wed1"] = lapply(data2$j\_marst, as.character)

data2$wed1[which(data2$wed1=='1')] <- 0

data2$wed1[which(data2$wed1=='2')] <- 1

data2$wed1[which(data2$wed1=='6')] <- 1

data2$wed1[which(data2$wed1!='1')] <- 0

data2$wed1

data2["wed2"]= data2$j\_marst

data2$wed2 = as.numeric(data2$wed2)

data2$wed2[which(data2$wed2=='1')] <- 0

data2$wed2[which(data2$wed2=='4')] <- 1

data2$wed2[which(data2$wed2=='5')] <- 1

data2$wed2[which(data2$wed2!='1')] <- 0

data2$wed2

data2["wed3"]= data2$j\_marst

data2$wed3 = as.numeric(data2$wed3)

data2$wed3[which(data2$wed3!='1')] <- 0

data2$wed3

#удовлетворенность условиями труда

data2["satisfy1"] = data2$jj1.1.2

data2$satisfy1 = as.numeric(data2$satisfy1)

data2$satisfy1[which(data2$satisfy1=="1")] <- 1

data2$satisfy1[which(data2$satisfy1=="2")] <- 1

data2$satisfy1[which(data2$satisfy1!="1")] <- 0

data2$satisfy1

data2["satisfy2"] = data2$jj1.1.2

data2$satisfy2 = as.numeric(data2$satisfy2)

data2$satisfy2[which(data2$satisfy2=="1")] <- 0

data2$satisfy2[which(data2$satisfy2=="4")] <- 1

data2$satisfy2[which(data2$satisfy2=="5")] <- 1

data2$satisfy2[which(data2$satisfy2!="1")] <- 0

data2$satisfy2

#линейная регрессия зарплаты на все параметры

model1 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2)

summary(model1)

vif(model1)

#коэффициент вздутия дисперсии VIF для всех регрессоров хороший

#R-squared: 0.1827

#введем степени

model2 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(0.1)))

summary(model2)

vif(model2)

# R-squared: 0.2181 - повысился, vif приемлемый

model3 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(0.1)))

summary(model3)

vif(model3)

# R-squared снизился, но не сильно, звёздочек у доп регрессоров не много, vif приемлемый

model4 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(0.1)) + I(age^(0.1)))

summary(model4)

vif(model4)

# R-squared снизился, но не сильно, звёздочек у доп регрессоров не много, vif приемлемый

model5 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(0.2)))

summary(model5)

vif(model5)

# R-squared снизился, но не сильно, звёздочек у доп регрессоров не много, vif приемлемый

model6 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(0.2)))

summary(model6)

vif(model6)

# R-squared: 0.2182 - повысился, vif приемлемый

model7 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(0.2)) + I(age^(0.2)))

summary(model7)

vif(model7)

# R-squared снизился, но не сильно, звёздочек у доп регрессоров не много, vif приемлемый

model8 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(0.3)))

summary(model8)

vif(model8)

# R-squared снизился, vif приближается к 10

model9 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(0.3)))

summary(model9)

vif(model9)

# R-squared: 0.2183 - повысился, vif приемлемый

model10 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(0.3)) + I(age^(0.3)))

summary(model10)

vif(model10)

# R-squared снизился, но не сильно, звёздочек у доп регрессоров не много, vif приближается к 10

model11 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(0.4)))

summary(model11)

vif(model11)

# R-squared снизился, но не сильно, vif > 10

#лучшая модель среди степеней

model12 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(0.4)))

summary(model12)

vif(model12)

# R-squared: 0.2184 - повысился, но vif приближается к 10

model13 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(0.4)) + I(age^(0.4)))

summary(model13)

vif(model13)

# R-squared снизился, vif > 10, увеличивать, дальше, скорее всего, нет смысла

model14 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(0.5)))

summary(model14)

vif(model14)

# R-squared снизился, vif большой

model15 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(0.5)))

summary(model15)

vif(model15)

# R-squared: 0.2185 - повысился, но vif > 10

model16 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(0.5)) + I(age^(0.5)))

summary(model16)

vif(model16)

# R-squared снизился, vif большой

#значит, для степеней 0<n< 1: чем ближе степень к единице, тем хуже становится vif. Начиная со степени 0.4 vif уже

#начиная со степени 0.4 vif очень большим, модели становятся только хуже

model17 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(1.1)))

summary(model17)

vif(model17)

# R-squared снизился, но не сильно, vif очень большой

model18 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(1.1)))

summary(model18)

vif(model18)

# R-squared: 0.2191 - повысился, но vif очень большой

model19 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(1.1)) + I(age^(1.1)))

summary(model19)

vif(model19)

# R-squared снизился, vif очень большой

model20 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(1.2)))

summary(model20)

vif(model20)

# R-squared снизился, но не сильно, vif очень большой

model21 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(1.2)))

summary(model21)

vif(model21)

# R-squared: 0.2192 - повысился, но vif очень большой

model22 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(1.2)) + I(age^(1.2)))

summary(model22)

vif(model22)

# R-squared снизился, vif очень большой

model23 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(1.3)))

summary(model23)

vif(model23)

# R-squared снизился, vif большой, но по сравнению с прошлой моделью уменьшился

model24 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(1.3)))

summary(model24)

vif(model24)

# R-squared: 0.2192 - повысился, vif большой, но по сравнению с прошлой моделью уменьшился

model25 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(1.3)) + I(age^(1.3)))

summary(model25)

vif(model25)

# R-squared снизился, vif очень большой, но по сравнению с прошлой моделью уменьшился

model26 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(1.4)))

summary(model26)

vif(model26)

# R-squared снизился, но не сильно, vif большой, но снижается

model27 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(1.4)))

summary(model27)

vif(model27)

# R-squared: 0.2193 - повысился, но vif большой

model28 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(1.4)) + I(age^(1.4)))

summary(model28)

vif(model28)

# R-squared снизился, vif большой, снижается

model29 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(1.9)))

summary(model29)

vif(model29)

# R-squared снизился, vif приемлемый

model30 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(2)))

summary(model30)

vif(model30)

# R-squared: 0.1949 - понизился, но vif приемлемый

model31 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(1.9)) + I(age^(1.9)))

summary(model31)

vif(model31)

# R-squared снизился, vif > 10

model32 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(workweek^(2)) + I(age^(2)))

summary(model32)

vif(model32)

# R-squared снизился, vif в норме

#лучшая модель среди степеней

model12 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(0.4)))

summary(model12)

vif(model12)

# R-squared: 0.2184 - выше первоначальной модели, vif < 10

#введем логарифмы

model33 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(log(workweek)) + I(log(age)))

summary(model33)

vif(model33)

# R-squared: 0.1733 снизился, vif приемлемый

model34 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(log(workweek)))

summary(model34)

vif(model34)

# R-squared: 0.1564 снизился, vif приемлемый

#лучшая среди логарифмов

model35 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(log(age)))

summary(model35)

vif(model35)

# R-squared: 0.218, vif приемлемый

#сравнение лучших моделей

model1 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2)

summary(model1)

vif(model1)

#Multiple R-squared: 0.1855, Adjusted R-squared: 0.1827

#попробуем убрать satisfy2

#vif хороший у всех регрессоров

model1.1 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1+ wed2 + wed3 + satisfy1)

summary(model1.1)

vif(model1.1)

#Multiple R-squared: 0.1842, Adjusted R-squared: 0.1817 - снизился, но не сильно

#vif хороший у всех регрессоров

model6 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(0.2)))

summary(model6)

vif(model6)

#Multiple R-squared: 0.2237, Adjusted R-squared: 0.2182 - выше, чем у предыдущей модели

#vif у всех регрессоров < 5

#попробуем убрать satisfy2

model6.1 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + I(age^(0.2)))

summary(model6.1)

vif(model6.1)

#Multiple R-squared: 0.222, Adjusted R-squared: 0.2169 - незначительно сниился по сравнению с предыдущей моделью 6

#vif у всех регрессоров < 5

model12 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(age^(0.4)))

summary(model12)

vif(model12)

# R-squared: 0.224, но vif приближается к 10

#Multiple R-squared: 0.224, Adjusted R-squared: 0.2184 - повысился относительно модели 1

#попробуем убрать satisfy2

#vif увеличился у регрессоров age и I(age^(0.4))

model12.1 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + + wed1 + wed2 + wed3+ satisfy1 + I(age^(0.4)))

summary(model12.1)

vif(model12.1)

#Multiple R-squared: 0.2222, Adjusted R-squared: 0.2171 - незначительно снизился относительно модели 12

#почти у всех регрессоров по 3 звездочки

model35 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + satisfy2 + I(log(age)))

summary(model35)

vif(model35)

#Multiple R-squared: 0.2236, Adjusted R-squared: 0.218

#попробуем убрать satisfy2

#у доп регрессоров мало звёздочек

#vif приемлемый

model35.1 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + I(log(age)))

summary(model35.1)

vif(model35.1)

# R-squared: 0.218, vif приемлемый

#Multiple R-squared: 0.2218, Adjusted R-squared: 0.2167 - незначительно снизился относительно модели 35

#почти у всех регрессоров по 3 звездочки

#vif приемлемый

#лучшая модель, поскольку vif у всех регрессоров приемлемый, почти у всех регрессоров по 3 звездочки, и Adjusted R-squared: 0.2169 - один из самых высоких в представленных моделях

model6.1 =lm(data = data2, salary ~ workweek + age + sex + j\_educ + status2 + wed1 + wed2 + wed3 + satisfy1 + I(age^(0.2)))

summary(model6.1)

vif(model6.1)

#мужчины, разведённые с высшим образованием

data3=subset(data2,sex==1)

data3

data4=subset(data3,wed2 == 1)

data4

data5=subset(data4,j\_educ==1)

data5

model\_subset1 =lm(data = data5, salary ~ workweek + age + status2 + satisfy1 + I(age^(0.2)))

summary(model\_subset1)

vif(model\_subset1)

#женщины, живущие в городе, состоящие в браке

data6=subset(data2,sex==0)

data6

data7=subset(data6,status2==1)

data7

data8=subset(data7,wed1==1)

data8

model\_subset2 =lm(data = data8, salary ~ workweek + age + j\_educ + satisfy1 + I(age^(0.2)))

summary(model\_subset2)

vif(model\_subset2)

Приложение 4

import pandas

import numpy as np

from sklearn.linear\_model import SGDClassifier

data = pandas.read\_csv('C:\\Users\\brofr\\Desktop\\МИРЭА\\НИР\\drug200.csv')

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

data\_sel = data\_sel.dropna()

#Обработаем столбец "Sex". M -> 1, F -> 0

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

#Обработаем столбец "BP". LOW -> 0, NORMAL -> 1, HIGH -> 2

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

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

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

#Обработаем столбец "Cholesterol". NORMAL -> 0, HIGH -> 1

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

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

#Обработаем столбец "Drug Type". 4 и выше – класс 0, ниже – класс 1

data\_sel['Drug'] = np.where(data\_sel['Drug'] == 'drugA',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

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Drug, test\_size=0.3)

#Строим классификатор SGDClassifier (метод опорных векторов)

clf = SGDClassifier(loss="log", penalty="l2")

clf.fit(X\_train,y\_train)

clf

y\_pred = clf.predict(X\_test)

clf.score(X\_test, y\_test)

#Оценим точность построенного классификатора с помощью метрик accuracy, precision, recall и F1 на тестовой выборке.

import sklearn.metrics

from sklearn.metrics import classification\_report

report = classification\_report(y\_test, y\_pred)

print(report)

#Cтроим классификатор Случайный Лес(Random Forest)

#Рассмотрим 200 деревьев

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import GridSearchCV

param\_grid = {'n\_estimators' : [200], '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)

#Оцениваем точность классификатора Случайный Лес с помощью метрик accuracy, precision, recall и F1

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import GridSearchCV

print("accuracy:" +str(np.average(sklearn.model\_selection.cross\_val\_score(RFC.best\_estimator\_, X\_test, y\_test, scoring='accuracy'))))

print("f1:" +str(np.average(sklearn.model\_selection.cross\_val\_score(RFC.best\_estimator\_, X\_test, y\_test, scoring='f1'))))

print("precision:" +str(np.average(sklearn.model\_selection.cross\_val\_score(RFC.best\_estimator\_, X\_test, y\_test, scoring='precision'))))

print("recall:" +str(np.average(sklearn.model\_selection.cross\_val\_score(RFC.best\_estimator\_, X\_test, y\_test, scoring='recall'))))

#получим

#accuracy:0.95

#f1:0.9549407114624506

#precision:0.9318181818181819

#recall:1.0

#Далее рассмотрим 300 деревьев

param\_grid = {'n\_estimators' : [300], '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)

print("accuracy:" +str(np.average(sklearn.model\_selection.cross\_val\_score(RFC.best\_estimator\_, X\_test, y\_test, scoring='accuracy'))))

print("f1:" +str(np.average(sklearn.model\_selection.cross\_val\_score(RFC.best\_estimator\_, X\_test, y\_test, scoring='f1'))))

print("precision:" +str(np.average(sklearn.model\_selection.cross\_val\_score(RFC.best\_estimator\_, X\_test, y\_test, scoring='precision'))))

print("recall:" +str(np.average(sklearn.model\_selection.cross\_val\_score(RFC.best\_estimator\_, X\_test, y\_test, scoring='recall'))))

#получим

#accuracy:0.9333333333333333

#f1:0.9635987201204592

#precision:0.8984848484848484

#recall:1.0

#Вариант с 200 деревьями немного лучше остальных. Произведём анализ вокруг него:

#Далее рассмотрим 220 деревьев

param\_grid = {'n\_estimators' : [220], '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)

print("accuracy:" +str(np.average(sklearn.model\_selection.cross\_val\_score(RFC.best\_estimator\_, X\_test, y\_test, scoring='accuracy'))))

print("f1:" +str(np.average(sklearn.model\_selection.cross\_val\_score(RFC.best\_estimator\_, X\_test, y\_test, scoring='f1'))))

print("precision:" +str(np.average(sklearn.model\_selection.cross\_val\_score(RFC.best\_estimator\_, X\_test, y\_test, scoring='precision'))))

print("recall:" +str(np.average(sklearn.model\_selection.cross\_val\_score(RFC.best\_estimator\_, X\_test, y\_test, scoring='recall'))))

#получим

#accuracy:0.95

#f1:0.9722943722943723

#precision:0.9136363636363637

#recall:1.0

#вариант с 220 лучше, чем вариант 200

#Сравним метрики обоих классификаторов

#Метод опорных векторов

accuracy: 0.87

f1: 0.93

precision: 0.87

recall: 1.0

#Случайный лес с 220 деревьями

accuracy: 0.95

f1: 0.9722943722943723

precision: 0.9136363636363637

recall:1.0

#Случайный лес на этих данных показывает себя по всем показателям лучше

Приложение 5

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import pandas

data = pandas.read\_csv('C:\\Users\\brofr\\Desktop\\МИРЭА\\НИР\\crime.csv')

data.shape

#Датасет содержит 486886 строк (объектов) и 19 столбцов (признаков).

#Первые 5 записей

data.head()

#Выведем информацию о наборе данных

data.info()

#удалим строки с отсутствующими значениями

data = data.dropna( )

#после удаления стало 175634 объектов

#Cтатистический анализ числовых столбцов

data.describe()

#Корреляция признаков

data.corr()

#Корреляция признаков на температурной таблице

plt.figure(figsize=(12,10), dpi= 70)

sns.heatmap(data.corr(), xticklabels=data.corr().columns, yticklabels=data.corr().columns, cmap='RdYlGn', center=0, annot=True)

plt.xticks(fontsize=10)

plt.yticks(fontsize=10)

plt.show()

#Нормализация признаков через стандартное отклонение

from sklearn.preprocessing import StandardScaler

scale\_features\_std = StandardScaler()

features\_std = scale\_features\_std.fit\_transform(data[['incident\_id', 'offense\_id', 'OFFENSE\_CODE', 'OFFENSE\_CODE\_EXTENSION', 'GEO\_X', 'GEO\_Y', 'GEO\_LON', 'GEO\_LAT', 'DISTRICT\_ID', 'PRECINCT\_ID', 'IS\_CRIME', 'IS\_TRAFFIC']])

features\_std

#Для вопроса №9. Поищем выбросы, аномальные значения. Буду считать, что значение аномально, если его стандартное отклонение >= 5 по модулю

for i in range(0,175634):

for j in range(0,12):

if (features\_std[i][j] >= 5) or (features\_std[i][j] <= -5):

print(i)

print(j)

#Для вопроса №10. Пронормируем признаки через стандартное отклонение, и найдём столбец с самым большим средним значением

s = 0

for i in range(0,12):

for j in range(0,175634):

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

s = s/175634

print(s)

s = 0

data[['incident\_id', 'offense\_id', 'OFFENSE\_CODE', 'OFFENSE\_CODE\_EXTENSION', 'GEO\_X', 'GEO\_Y', 'GEO\_LON', 'GEO\_LAT', 'DISTRICT\_ID', 'PRECINCT\_ID', 'IS\_CRIME', 'IS\_TRAFFIC']] = features\_std

#Выделяем целевую переменную

target = data.OFFENSE\_CODE

train = data.drop(['OFFENSE\_CODE'], axis=1)

train.head()

#Выделяем тренировочную и тестовую выборки

# y - целевая переменная(target)

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

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

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

print (N\_train, N\_test)

from sklearn.decomposition import PCA

%matplotlib inline

import matplotlib.pyplot as plt

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)))