Задача №1

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: Swiss.

Объясняемая переменная: *Agriculture*.

Регрессоры: *Fertility*, *Examination*.

***№1***. Оценить среднее значение, дисперсию и СКО переменных, указанных во втором и в третьем столбце.

* *Agriculture*:
* Дисперсия = 515.799 (большой разброс)
* Ско = 22.71
* Среднее арифметическое = 50.66
* *Fertility*:
* Дисперсия = 156.04 (средний разброс)
* Ско = 12.49
* Среднее арифметическое = 70.14
* *Examination*:
* Дисперсия = 63.65 (маленький разброс)
* Ско = 7.98
* Среднее арифметическое = 16.49

***№2***. Построить зависимости вида y = a + bx, где y – объясняемая переменная, x – регрессор.

*Таблица 1*. Характеристики модели зависимости параметра *Agriculture* от параметра *Examination* в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 82.88 | 5.64 | 14.69 | < 2e-16 | \*\*\* |
| *Examination* | -1.95 | 0.31 | -6.33 | 9.95e - 08 | \*\*\* |

*# Agriculture = -1.95\* Examination + 82.88*

*Таблица 2*. Характеристики модели зависимости параметра *Agriculture* от параметра *Fertility* в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 5.63 | 18.06 | 0.31 | 0.76 |  |
| *Fertility* | 0.64 | 0.25 | 2.53 | 0.01 | \* |

*# Agriculture = 0.64\* Fertility + 5.63*

***№3***. Оценить, насколько «хороша» модель по коэффициенту детерминации R2.

* *Agriculture = -1.95\* Examination + 82.88:*

# R2 = 47%, что довольно неплохой показатель, так что можно сказать, что *Сельское Хозяйство* отрицательно зависит от *Оценок на Экзамене* (т.к. коэффициент отрицательный). Однако данная модель нуждается в обработке, поскольку она станет по-настоящему “*хорошей”* (то есть, по которой можно сделать однозначные прогнозы) только, когда R2 > 70%.

* *Agriculture = 0.64\* Fertility + 5.63:*

# Поскольку R2 = 0.12% (что << 30%), то можно с уверенность сказать, что данная модель плоха, и делать по ней какие-либо выводы нельзя.

***№4***. Оценить, есть ли взаимосвязь между объясняемой переменной и объясняющей

переменной.

* *Agriculture = -1.95\* Examination + 82.88:*

# Поскольку при обоих коэффициентах значение p-статистики очень хорошее (3 звезды, *см. Таблица 1*) , то можно сделать вывод, что существует взаимосвязь между Объясняемой Переменной*(Agriculture)* и Регрессом(*Examination).*Объяснить данную зависимость чисто логически довольно просто: «Чем выше оценки на экзамене, тем на более престижную вакансию претендует человек, вследствие чего, количество людей заинтересованных в работе в сфере сельского хозяйства снижается».

* *Agriculture = 0.64\* Fertility + 5.63:*

# Поскольку при обоих коэффициентах значение p-статистики плохое (0 звезд / 1 звезда, *см. Таблица 2*), то взаимосвязь между Объясняемой Переменной*(Agriculture)* и Регрессом(*Fertility)* практически отсутствует, поэтому делать какие-либо выводы по этой зависимости нельзя.

Заключение

Были построены две зависимости с одинаковой Объясняемой Переменной*(Agriculture)* и разными Объясняющими Переменными (*Examination /Fertility).* Повыводам из пунктов ***№3*** и ***№4*** видно, что обе исследуемые модели нуждаются в доработке, и ни по одной из них нельзя сделать однозначных прогнозов.

Задача №2.1

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: Swiss.

Объясняемая переменная: *Fertility.*

Регрессоры: *Agriculture*, *Examination, Infant.Mortality.*

***№1.*** Проверить, что в наборе данных нет линейной зависимости (построить зависимости

между переменными, и проверить, что R^2 в каждой из них не высокий). В случае если R2 большой, один из таких столбцов можно исключить из рассмотрения.

Проверим линейную регрессию *Agriculture* ~ *Examination*, *Infant.Mortality*. R2 = 49.1%. VIF = 1/(1-R2) = 1.96. Хоть VIF и меньше 5, но если учесть, что при регрессоре (*Examination*) хорошее значение p-статистики (3 звезды, *см. Таблица 1*), и что R2 = 49.1% - это довольно неплохой показатель, то можно сказать, что существует небольшая зависимость между регрессором (*Examination*) и регрессором(Agriculture).

Зависимость *Examination ~ Agriculture, Infant.Mortality*. R2 = 49.6%. VIF = 1/(1-R2) = 1.98. Хоть VIF и меньше 5, но если учесть, что при регрессоре (Agriculture) хорошее значение p-статистики (3 звезды, *см. Таблица 2*), и что R2 = 49.1% - это довольно неплохой показатель, то можно сказать, что существует небольшая зависимость между регрессором (*Agriculture*) и регрессором(*Examination*).

В регрессии *Infant.Mortality ~ Agriculture, Examination* R2 = 4.9%. VIF = 1/(1-R2) = = 1.05. Т.к. VIF значительно меньше 5 , и значение p-статистики плохое (0 звезд / 0 звезд, *см. Таблица 3*), то можно с уверенностью сказать, что регрессор(*Infant.Mortality)* не зависит от регрессоров(Agriculture, *Examination).*

Хоть между регрессором(*Agriculture*) и регрессором(*Examination*) и существует небольшая зависимость, но она слишком незначительна, чтобы их исключить. Таким образом, заключаем, что для построения моделей, можно использовать все регрессоры из условия.

***№2.*** Построить линейную модель зависимой переменной (*Fertility)* от регрессоров (*Agriculture*/*Examination*/*Infant.Mortality)* по методу наименьших квадратов. Оценить, насколько хороша модель, согласно: 1) R2, 2) p-значениям каждого коэффициента.

***2.1)*** В регрессии *Fertility* ~ *Agriculture*, *Examination*, *Infant.Mortality* R2 = 53.98%. Для трех регрессоров значение R2 довольно низкое, следовательно, сделать какие-либо выводы по данной зависимости нельзя.

***2.2)*** Рассмотрим p-статистику у регрессоров (*Agriculture*, *Examination*, *Infant.Mortality*).

* Значение p-статистики при регрессоре (*Examination)* низкое ( \*\*\*, *см. Таблица 4*);
* Значение p-статистики при регрессоре (*Infant.Mortality)* довольно низкое ( \*\*,

*см. Таблица 4*);

* Значение p-статистики при регрессоре (*Agriculture)* высокое (0 звезд,

*см. Таблица 4*);

Обратим внимание, что регрессор (*Agriculture)* не значим. P-статистика достаточно велика (*см. Таблица 4*), так что можно провести эксперимент по его исключению:

* R2 =53.64%, следовательно R2 изменился всего на 0.35%, что << 5%. Таким образом, мы можем исключить из рассмотрения регрессор (Agriculture).

***№3.*** Ввести в модель логарифмы регрессоров. Сравнить модели и выбрать наилучшую.

При решении этой задачи были проверены модели:

1. *Fertility ~ log(Examination), log(Infant.Mortality) – Таблица 5*
2. *Log(Fertility) ~ log(Examination), log(Infant.Mortality) – Таблица 6*
3. *Fertility ~ log(Examination), Infant.Mortality – Таблица 7*
4. *Fertility ~ Examination, log(Infant.Mortality) – Таблица 8*

Значения R2 для проверенных моделей:

1. 52.16% - показатели ухудшились;

2. 48.58% - показатели ухудшились;

3. 54% - показатели немного улучшились;

4. 52% - показатели ухудшились.

Наилучшей оказалась модель: *Fertility ~ log(Examination), Infant.Mortality.*

*Таблица 7*. Характеристики модели зависимости параметра *Fertility* от параметров *log(Examination),* и *Infant.Mortality*  в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 67.48 | 10.45 | 6.46 | 7.11e-08 | \*\*\* |
| *log(Examination)* | -12.89 | 2.18 | -5.92 | 4.44e-07 | \*\*\* |
| *Infant.Mortality* | 1.85 | 0.44 | 4.22 | 0.000119 | \*\*\* |

*# Fertility* *= -12.89\* log(Examination) + 1.85\* Infant.Mortality + 67.48*

Вывод: Хоть введение в модель логарифмы регрессоров и дало прирост, но слишком незначительный.

***№4.*** Ввести в модель всевозможные произведения пар регрессоров, в том числе квадраты регрессоров. Найти одну или несколько наилучших моделей по доле объяснённого разброса в данных R2 .

***4.1)*** При решении этой задачи была проверена модель (*Fertility*~ *Examination*, *Infant.Mortality),* в которую были добавлены параметры: *I(Examination2) / I(Infant.Mortality2) / I(Examination \*Infant.Mortality). #R2 = 54.69%*

*Таблица 9*. Проверка на линейную зависимость между *регрессоров (Examination, Infant.Mortality , I(Examination2), I(Infant.Mortality2), I(Examination\*Infant.Mortality))* с помощью команды VIF.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | *Examination* | *Infant.Mortality* | *I(Examination2)* | *I(Infant.Mortality2)* | *I(Examination\**  *Infant.Mortality).* |
| VIF | 57.84 | 159.2 | 15.68 | 114.72 | 64.08 |

Поскольку у многих объясняющих переменных значения VIF довольно большое, то можно сделать вывод, что в модели присутствует линейная зависимость между регрессорами.

***4.2)*** Будем избавляться от регрессоров с максимальным VIF, пока все значения VIF не будут меньше 10.

1. *Fertility ~ Examination, I(Examination2), I(Infant.Mortality2), I(Examination\*Infant.Mortality). – Таблица 10;*
2. *Fertility ~ I(Examination2), I(Infant.Mortality2), I(Examination\*Infant.Mortality). -Таблица 11;*
3. *Fertility ~ I(Examination2), I(Infant.Mortality2) - Таблица 12;*

*Таблица 12*. Проверка на линейную зависимость между регрессоров *(I(Examination^2), I(Infant.Mortality2))* с помощью команды VIF.

|  |  |  |
| --- | --- | --- |
| Параметр \ Характеристики | *I(Examination2)* | *I(Infant.Mortality2* |
| VIF | 1.027 | 1.027 |

*Таблица 13*. Характеристики модели зависимости параметра *Fertility* от параметров *I(Examination2),* и *I(Infant.Mortality2)* в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 63.49 | 5.39 | 11.78 | 3.35e-15 | \*\*\* |
| *I(Examination2)* | -0.02 | 0.004 | 5.38 | 2.76e-06 | \*\*\* |
| *I(Infant.Mortality2)* | 0.04 | 0.01 | 3.054 | 0.004 | \*\* |

*# Fertility* *= -0.02\* I(Examination2) + 0.04\* I(Infant.Mortality2) + 63.49*

#Значение R2 для данной модели = 50.43%

Вывод:Наилучшая модель - это *«Fertility ~ I(Examination2), I(Infant.Mortality2)»,* поскольку все значения VIF регрессоров меньше 5.

Задача №2.2

***№1.*** Оценить доверительные интервалы для всех коэффициентов в модели «*Fertility* ~ *Agriculture*, *Examination*, *Infant.Mortality*».

Всего проводилось 47 наблюдений, в данной модели оценивалось 4 коэффициента.

Следовательно, количество свободных коэффициентов = 47 – 4 = 43.

* *Agriculture*:

СКО(se) = 0.08 (*см. Таблица 14*)

Критерий Стьюдента(t) = 2.02

Доверительный интервал: [-0.21, 0.12]

* *Examination*:

СКО(se) = 0.23 (*см. Таблица 14*)

Критерий Стьюдента(t) = 2.02

Доверительный интервал: [-1.5, -0.58]

* *Infant.Mortality*:

СКО(se) = 0.46 (*см. Таблица 14*)

Критерий Стьюдента(t) = 2.02

Доверительный интервал: [0.52, 2.36]

* *Intercept*:

СКО(se) = 12.83 (*см. Таблица 14*)

Критерий Стьюдента(t) = 2.02

Доверительный интервал: [35, 86.74]

***№2.*** Построим таблицу с доверительными интервалами для всех коэффициентов в модели и сделаем вывод о том, может ли коэффициент быть равен 0.

*Таблица 14*. Характеристики модели зависимости параметра *Fertility* от параметров *Agriculture*, *Examination* и *Infant.Mortality* в наборе данных Swiss.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | СКО | Доверительный интервал | “Может ли коэффициент быть равен 0?” |
| (Intercept) | 60.87 | 12.83 | [35, 86.74] | Нет |
| *Examination* | -1.04 | 0.23 | [-1.5, -0.58] | Нет |
| *Infant.Mortality* | 1.44 | 0.46 | [0.52, 2.36] | Нет |
| *Agriculture* | -0.05 | 0.08 | [-0.21, 0.12] | Да |

Вывод:Поскольку 0 попадает в доверительный интервал регрессора (*Agriculture),* то значение коэффициента перед этим регрессором может быть равно 0. Следовательно, объясняющая переменная (*Agriculture)* практически не связана с объясняемой переменной (*Fertility).*

***№3.*** Оценить доверительный интервал для одного прогноза для модели «*Fertility* ~ *Agriculture*, *Examination*, *Infant.Mortality*».

Зададим следующий набор значений для регрессоров: (*Agriculture = 10*, *Examination = 30* , *Infant.Mortality = 20).*

Применим функцию *predict()* для оцениваемой модели, что вычислить прогноз модели и доверительный интервал:

*Таблица 15*. Результат выполнения функции *predict()*

|  |  |  |
| --- | --- | --- |
| Прогноз модели | Нижняя граница интервала | Верхняя граница интервала |
| 58.01866 | 52.43242 | 63.60491 |

Вывод: Прогноз модели «*Fertility* ~ *Agriculture*, *Examination*, *Infant.Mortality*» оценивается как 58.02. Доверительный интервал для свободного коэффициента имеет вид: [52.43, 63.6].

Заключение

Задача №2.1. Рассматриваемая модель была проверена на наличие линейной зависимости между регрессорами. Хоть и была обнаружена небольшая зависимость между двумя объясняющими переменными, но в виду её незначительности, было принято решение не исключать ни одну из переменных в модели.

Однако при исследовании самой модели было выявлено, что объясняемая переменная(*Fertility*) почти не зависит от одного из регрессоров(*Agriculture),* поэтому было всё же решено исключить из рассмотрения одну из объясняющих переменных.

В пункте ***№3*** была попытка улучшить рассматриваемую модель, путём введения логарифмов регрессоров. Однако, это не дало видимых результатов.

В пункте ***№4*** в модель были введены всевозможные произведения пар регрессоров, и была выявлена одна наилучшая модель по доле объяснённого разброса в данных R2.

Задача №2.2. Были найдены доверительные интервалы для всех коэффициентов в рассматриваемой модели (при p=95%), и было выявлено, что т.к. значение коэффициента перед регрессором(*Agriculture)* может быть равно 0, то объясняющая переменная (*Agriculture)* практически не связана с объясняемой переменной (*Fertility).*

В пункте ***№3*** для оценивания доверительного интервала для одного прогноза, были выбраны следующие значения: «*Agriculture = 10*, *Examination = 30* , *Infant.Mortality = 20*». Затем, с помощью функции *predict()* был вычислен прогноз и доверительный интервал для рассматриваемой модели.

Задача №3

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: Данные обследования РМЭЗ НИУ ВШЭ

Объясняемая переменная: *salary*

Регрессоры: *age, sex, higher\_educ, status2, dur, wed1,wed2,wed3.*

***№1***. Построить линейную регрессию зарплаты на все параметры, и оценить коэффициент вздутия дисперсии VIF.

Построим линейную регрессию зарплаты(*salary)* на все параметры, предварительно исключив из рассмотрения все строки с отсутствующими значениями(NA). После этого также исключим из рассмотрения все регрессоры с отсутствующими значениями, и все незначительные регрессоры (*см. Таблица 16*).

Построим линейную регрессию зарплаты(*salary)* на все оставшиеся параметры (*age, sex, dur).* R2 = 0.06639. Значение p-статистики хорошее у всех переменных(3 звезды, *см. Таблица 17*). VIF = 1 / (1-R2) = 1.004. VIF << 5, следовательно линейной зависимости между переменными нет.

***№2.***Введём в модель логарифмы и степени переменных:

***№2.1)*** Введём в модель логарифмы регрессоров, сравним модели и выберем наилучшую.

Рассмотрим модель (*salary* ~ *age*, *sex , dur, log(age), log(dur)*)*.*  R2 = 0.09061, следовательно, показатель увеличился на 0.024, по сравнению с моделью: "*salary* ~ *age*, *sex*, *dur* ". Значение p-статистики хорошее только у переменных *log(age)* и *sex* (3 звезды, *см. Таблица 18*).

Попробуем улучшить модель, убирая из неё параметры с наибольшими коэффициентами в VIF.

*Таблица 19*. Проверка на линейную зависимость между регрессорами *(age*, *sex , dur, log(age), log(dur)*) с помощью команды VIF.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | *age* | *sex* | *dur* | *log(age)* | *log(dur)* |
| VIF | 3.79 | 1.02 | 4.14 | 3.79 | 4.16 |

Рассмотрим модель (*salary* ~ *age*, *sex, dur, log(age)*)*.* R2 =  0.06722, следовательно, показатель увеличился на 0.008, по сравнению с моделью: "*salary* ~ *age*, *sex*, *dur*". Значение p-статистики очень хорошая(3 звезды, см. *Таблица 20*) у всех переменных, кроме *log(age)*и "Свободного Коэффициента".

*Таблица 21*. Проверка на линейную зависимость между регрессорами(*age*, *sex , dur, log(age)*)с помощью команды VIF.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр \ Характеристики | *age* | *sex* | *dur* | *log(age)* |
| VIF | 3.64 | 1.04 | 1.05 | 3.61 |

Рассмотрим модель (*salary* ~ *sex , dur, log(age)*). R2 =  0.05907, следовательно, показатель уменьшился на 0.007, по сравнению с моделью: "*salary* ~ *age*, *sex*, *dur*". Значение p-статистики очень хорошее (3 звезды, см. *Таблица 22*) у всех переменных.

***№2.2)*** Введём в модель степени регрессоров, сравним модели и выберем наилучшую.

Рассмотрим модель (*salary* ~ *age*, *sex , dur, I(age0.1), I(dur0.1)*). R2 =  0.09058 - следовательно, показатель увеличился на 0.024, по сравнению с моделью: "*salary* ~ *age*, *sex*, *dur*". Значение p-статистики очень хорошее (3 звезды, см. *Таблица 23*) только у переменных *age* и *sex*.

*Таблица 24*. Проверка на линейную зависимость между регрессорами *(age*, *sex , dur, I(age0.1), I(dur0.1)*) с помощью команды VIF.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | *age* | *sex* | *dur* | *I(age0.1)* | *I(dur0.1)* |
| VIF | 4.835 | 1.02 | 5.01 | 4.829 | 5.03 |

Попробуем улучшить модель, убирая из неё параметры с наибольшими коэффициентами в VIF.

Рассмотрим модель (*salary* ~ *age*, *sex , dur, I(age0.1)*). R2 =  0.09058 - следовательно, показатель увеличился на 0.024, по сравнению с моделью: "*salary* ~ *age*, *sex*, *dur*". Значение p-статистики очень хорошее (3 звезды) только у переменных *age* и *sex*.

*Таблица 25*. Проверка на линейную зависимость между регрессорами *(age*, *sex , dur, I(age0.1)*) с помощью команды VIF.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр \ Характеристики | *age* | *sex* | *dur* | *I(age0.1)* |
| VIF | 4.6 | 4.58 | 1.24 | 1.05 |

Рассмотрим модель (*salary* ~ *sex , dur, I(age0.1)*). R2 =  0.06045 - следовательно, показатель увеличился на 0.0059, по сравнению с моделью: "*salary* ~ *age*, *sex*, *dur*". Значение p-статистики очень хорошее (3 звезды, см. *Таблица 26*) у всех переменных.

Попробуем улучшить модель, повышая степень (до степени "2", с шагом 0.1):

При решении этой задачи были проверены модели:

1. *salary* ~ *sex , dur, I(age0.2)*
2. *salary* ~ *sex , dur, I(age0.3)*
3. *salary* ~ *sex , dur, I(age0.4)*
4. *salary* ~ *sex , dur, I(age0.5)*

Поскольку при повышении степени от "0.1" до "0.5"(см. *Таблица 27*) R2 возрастает, то попробуем взять сразу степень равную "0.9":

1. *salary* ~ *sex , dur, I(age0.9)*
2. *salary* ~ *sex , dur, I(age1.1)*
3. *salary* ~ *sex , dur, I(age1.2)*
4. *salary* ~ *sex , dur, I(age1.3)*
5. *salary* ~ *sex , dur, I(age1.4)*

Поскольку при степени "1.4" значение R2 меньше, чем при степени "1.3"(см. *Таблица 27*), то попробуем сразу взять степень равную "1.9":

1. *salary* ~ *sex , dur, I(age1.9)*
2. *salary* ~ *sex , dur, I(age2)*

Построим сравнительную таблицу для всех проверенных моделей:

*Таблица 27*.Сравнение уровня значимости и значения R2 для одиннадцати рассматриваемых моделей.

|  |  |  |  |
| --- | --- | --- | --- |
| Рассматриваемая модель | Уровень значимости | Значение R2 | Насколько изменилось значение R2 по сравнению с моделью: "salary ~ age, sex, dur" |
| *salary* ~ *sex , dur, I(age0.2)* | Значение p-статистики очень хорошее (\*\*\*) у всех переменных | 0.06174 | уменьшился на 0.005 |
| *salary* ~ *sex , dur, I(age0.3)* | Значение p-статистики очень хорошее (\*\*\*) у всех переменных | 0.06291 | уменьшился на 0.003 |
| *salary* ~ *sex , dur, I(age0.4)* | Значение p-статистики очень хорошее (\*\*\*) у всех переменных, кроме "Свободного Коэффициента" | 0.06394 | уменьшился на 0.002 |
| *salary* ~ *sex , dur, I(age0.5)* | Значение p-статистики очень хорошее (\*\*\*) у всех переменных, кроме "Свободного Коэффициента" | 0.06394 | уменьшился на 0.001 |
| *salary* ~ *sex , dur, I(age0.9)* | Значение p-статистики очень хорошее (\*\*\*) у всех переменных, кроме "Свободного Коэффициента" | 0.06704 | увеличился на 0.0007 |
| *salary* ~ *sex , dur, I(age1.1)* | Значение p-статистики очень хорошее (\*\*\*) у всех переменных, кроме "Свободного Коэффициента" | 0.06749 | увеличился на 0.0011 |
| *salary* ~ *sex , dur, I(age1.2)* | Значение p-статистики очень хорошее (\*\*\*) у всех переменных, кроме "Свободного Коэффициента" | 0.06758 | увеличился на 0.00119 |
| *salary* ~ *sex , dur, I(age1.3)* | Значение p-статистики очень хорошее (\*\*\*) у всех переменных, кроме "Свободного Коэффициента" | 0.06759 | увеличился на 0.0012 |
| *salary* ~ *sex , dur, I(age1.4)* | Значение p-статистики очень хорошее (\*\*\*) у всех переменных, кроме "Свободного Коэффициента" | 0.06754 | увеличился на 0.00115 |
| *salary* ~ *sex , dur, I(age1.9)* | Значение p-статистики очень хорошее (\*\*\*) у всех переменных | 0.06643 | увеличился на 0.00004 |
| *salary* ~ *sex , dur, I(age2)* | Значение p-статистики очень хорошее (\*\*\*) у всех переменных | 0.07877 | увеличился на 0.012 |

***№3.***Выберем наилучшую модель среди построенных.

***3.1)*** Вывод: Среди моделей с логарифмами лучшей можно считать: " *salary* ~ *sex , dur, log(age)*", однако эта модель хуже первоначальной: "*salary* ~ *age*, *sex*, *dur*" (т.к. p-статистика хорошая в обоих случаях, а R2 в модели с логарифмом меньше, чем в первоначальной модели).

Следовательно, l*og()* не дал ощутимых результатов.

***3.2)*** Вывод: Среди моделей со степенями лучшая - это модель с квадратом, поскольку в остальных случаях у нас ниже R2 и больше параметров с плохим значением p-статистики.

***№4***.Сделать вывод о том, какие индивиды получают наибольшую зарплату.

Вывод: Наибольшую зарплату получают мужчины более молодого возраста, живущие в городе, с высшим образованием, с большой продолжительностью рабочей недели и никогда не состоящие в браке.

***№5***.Оценить регрессии для подмножества индивидов:

***5.1)***  Городские жители, состоящие в браке;

***5.2)***  Разведенные, без высшего образования.

***5.1)*** Найдем подмножество городских жителей, состоящих в браке:

data\_3 = subset(data\_2, status2 == 1)  
data\_3  
  
data\_4 = subset(data\_3, wed1 == 1)  
data\_4

Рассмотрим модель (*salary* ~ *sex , dur, I(age2)*). R2 = 0.1173, Значение p-статистики хорошее у всех переменных, кроме "Свободного Коэффициента» (см. *Таблица 13*). Следовательно, можно сделать вывод, что модель довольно хорошая.

Вывод: Наибольшую зарплату получают мужчины, с высшим образованием.

***5.2)*** Найдем подмножество разведенных, без высшего образования:

data\_5 = subset(data\_2, wed2=1)  
data\_5  
  
data\_6 = subset(data\_5, higher\_educ==0)  
data\_6

Рассмотрим модель (*salary* ~ *sex , dur, I(age2)*). R2 = 0.09543, Значение p-статистики хорошее у всех переменных (см. *Таблица 29*). Следовательно, можно сделать вывод, что модель довольно хорошая.

Вывод: Наибольшую зарплату получают мужчины более молодого возраста, с большой продолжительностью рабочей недели.

Заключение

В пункте ***№1*** была построена линейная регрессия зарплаты на все параметры. Оценив коэффициент вздутия дисперсии VIF, мы выявили, что линейной зависимости между переменными нет.

В пункте ***№2.1*** была попытка улучшить рассматриваемую модель, путём введения логарифмов регрессоров. Однако это не дало видимых результатов.

В пункте ***№2.2*** в модель были введены всевозможные произведения пар регрессоров, и была выявлена одна наилучшая модель по доле объяснённого разброса в данных R2.

Исходя из предоставленных данных, было выявлено, что наибольшую зарплату получают мужчины более молодого возраста, живущие в городе, с высшим образованием, с большой продолжительностью рабочей недели и никогда не состоящие в браке.

В пункте ***№5.1*** была построена зависимость параметра *salary* от параметров *sex, dur, I(age2)* для подмножества индивидов: “Городские жители, состоящие в браке”, и было выявлено, что наибольшую зарплату получают мужчины, с высшим образованием.

В пункте ***№5.2*** была построена зависимость параметра *salary* от параметров *sex, dur, I(age2)* для подмножества индивидов: “Разведенные, без высшего образования”, и было выявлено, что наибольшую зарплату получают мужчины более молодого возраста, с большой продолжительностью рабочей недели.

Задача №4

Набор данных: https://www.kaggle.com/arashnic/hr-analytics-job-change-of-data-scientists?select=aug\_test.csv

Тип классификатора – DecisionTreeClassifier (решающее дерево)  
Классификация по столбцу - Education level (Graduate – класс 0, остальные уровни – класс 1)

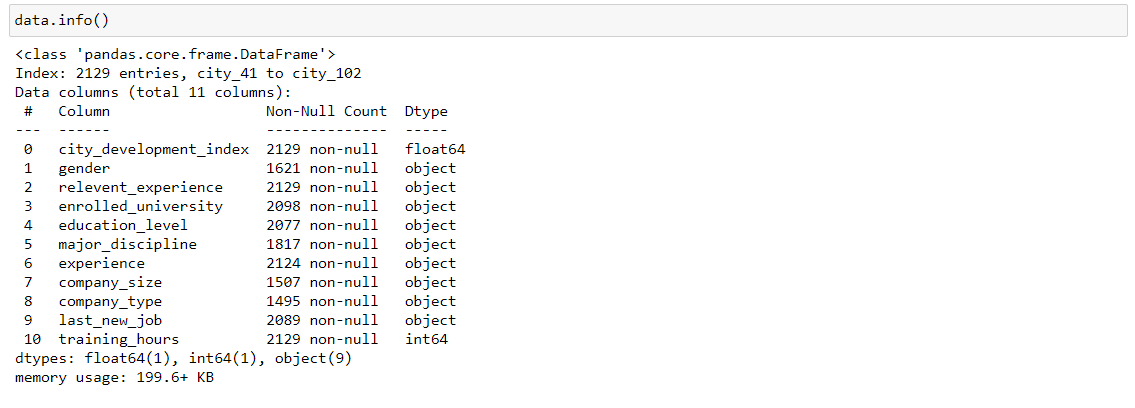
***№1.*** Обработайте набор данных набор данных, указанный во втором столбце таблицы 4.1, подготовив его к решению задачи классификации. Выделите целевой признак, указанный в последнем столбце таблицы, и удалите его из данных, на основе которых будет обучаться классификатор. Разделите набор данных на тестовую и обучающую выборку. Постройте классификатор типа, указанного в третьем столбце, для задачи классификации по параметру, указанному в последнем столбце. Оцените точность построенного классификатора с помощью метрик precision, recall и F1 на тестовой выборке.

***№2.*** 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?

Опишем столбцы нашего Датасета:

1. city\_ development \_index : индекс развития города (в масштабе)
2. gender: Пол кандидата
3. relevent\_experience: Соответствующий опыт кандидата
4. enrolled\_university: Тип зачисленных университетских курсов, если таковые имеются
5. education\_level: Уровень образования кандидата
6. major\_discipline: Обучение основной дисциплине кандидата
7. experience: Кандидатский общий стаж в годах
8. company\_size: Количество сотрудников в компании текущего работодателя
9. company\_type: Тип текущего работодателя
10. last\_new\_job: разница в годах между предыдущей работой и текущей работой
11. training\_hours: завершенные часы обучения

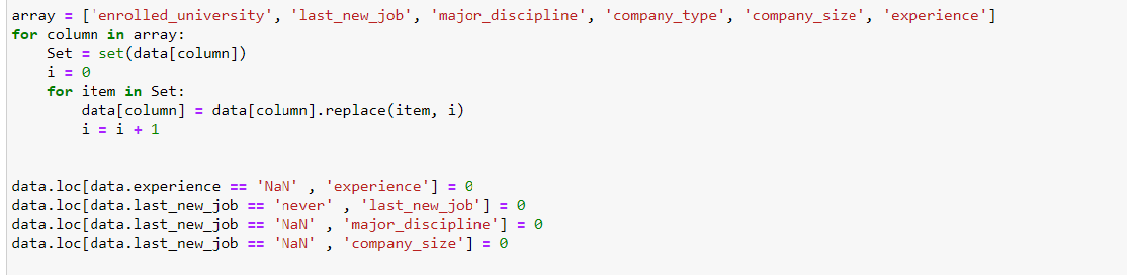
Применим функцию info(), чтобы посмотреть информацию о Датасете:

*Рисунок 3*. Результат работы команды info() в наборе данных aug\_test.csv.

Некоторые признаки имеют пропуски и являются типами “object”, а не численными (см. *Рисунок 3*). Обработаем данные для решения задачи классификации:

1) Кодируем столбцы из категориального признака в численный (см. *Рисунок 4*).

2) Будем заполнять пропуски в данных значением по умолчанию – индексом 0 (см. *Рисунок 4*).

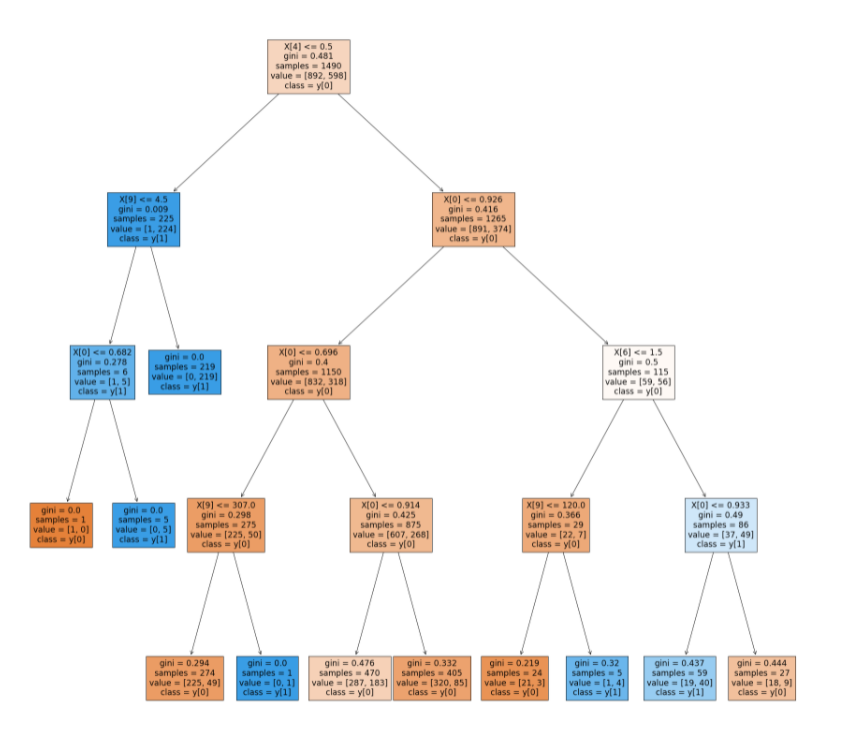


*Рисунок 4.* Преобразуем текстовые данные в числовые.

Выделим целевой признак(education\_level) и удалим его из данных, на основе которых будет обучаться классификатор:

Data2= data.education\_level  
train = data.drop ('education\_level', axis = 1)

Разделим данные на обучающую и тестовую выборку, а затем построим классификатор типа - "Решающее дерево" для задачи классификации по параметру(education\_level) (см. *Рисунок 5*).



*Рисунок 5.* Классификатор типа - "Решающее дерево" для задачи классификации по параметру(education\_level).

Оценим точность построенного классификатора с помощью метрик precision, recall и F1, а после попробуем улучшить данную модель, путём подбора гиперпараметров, с помощью GridSearchCV (*см. Таблица 30)*.

*Таблица 30*. Сравнительная таблица результата работы метрик precision, recall и F1 в наборе данных aug\_test.csv.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Метрики | Тип выборки | Обучающаяся выборка | | Тестовая выборка | | “Сравним результаты работы метрик на тестовой выборке” |
| Используя оценщик  GridSearchCV | Не используя, оценщик  GridSearchCV | Используя оценщик  GridSearchCV | Не используя, оценщик  GridSearchCV |
| F1 | | 0.6366 | 0.5689 | 0.533 | 0.5023 | Лучше после  подбора гиперпараметров |
| precision | | 0.7512 | 0.8944 | 0.827 | 0.8059 |
| recall | | 0.5602 | 0.4181 | 0.397 | 0.3702 |

Построим, классификатор типа Случайный Лес (Random Forest) для решения той же задачи классификации и оценим его качество с помощью метрик precision, recall и F1 на тестовой выборке (*см. Таблица 31)*.

*Таблица 31*. Работа метрик precision, recall и F1 в наборе данных aug\_test.csv.

|  |  |  |
| --- | --- | --- |
| Метрики | Обучающаяся выборка | Тестовая выборка |
| F1 | 0.6366 | 0.533 |
| precision | 0.7512 | 0.827 |
| recall | 0.5602 | 0.397 |

Заключение

Сравнивая результаты метрик классификаторов: Решающего дерева и Рандомного леса, видим, что классификатор Рандомного леса лучше, чем классификатор Решающего дерева, без использования оценщика GridSearchCV(*см. Таблица 32)*.

*Таблица 32*. Сравнение метрик классификаторов Решающего дерева и Рандомного леса.

|  |  |  |
| --- | --- | --- |
| Метрики | Решающего дерева | Рандомного леса |
| F1 | 50.23% | 53.3% |
| precision | 80.59% | 82.7% |
| recall | 37.02% | 39.7% |

На основе всех полученных данных можно сделать вывод, что классификатор DecisionTreeClassifier (решающее дерево) не подходит для данного Датасета, поскольку хоть он и имеет довольно хорошую полноту в 80.59%, но при этом имеет плохую точность в 50.23%, и f-меру в 37.02%.

Задача №5

Тема: "Употребление алкоголя учащимися"

Набор данных №1:

<https://www.kaggle.com/uciml/student-alcohol-consumpt..>  
Набор данных №2:

[https://www.kaggle.com/uciml/student-alcohol-consumpt..](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fwww.kaggle.com%2Fuciml%2Fstudent-alcohol-consumption%3Fselect%3Dstudent-mat.csv)

Необходимо провести анализ Датасета и сделать обработку данных по предложенному алгоритму.

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

Объектов - 1044, Признаков – 33

Признаки:

1. school(школа) - бинарный признак: "GP" - Gabriel Pereira / "MS" - Mousinho da Silveira

2. sex(пол) - бинарный признак: "F" - female / "M" – male

3. age(возраст) - числовой признак: 15<= age <= 22

4. address(тип населённого пункта) - бинарный признак: "U" - urban / "R" – rural

5. famsize(количество человек в семье) - бинарный признак: "LE3" - <=3 / "GT3" - >3

6. Pstatus(состоят ли родители в браке) - бинарный признак: "T" - living together / "A" – Apart

7. Medu(образование матери) - категориальный признак: "0" - нет образования, "1" - начальное образование(закончила 4 класса), "2" - неполное среднее(закончила 9 классов), "3" - среднее образование(закончила 11 классов),"4" - высшее образование

8. Fedu(образование отца) - категориальный признак: "0" - нет образования, "1" - начальное образование(закончил 4 класса), "2" - неполное среднее(закончил 9 классов), "3" - среднее образование(закончил 11 классов),"4" - высшее образование

9. Mjob (работа матери) - категориальный признак: "teacher", "health" - care related, "services" - administrative or police, "at\_home", "other"

10. Fjob(работа отца) - категориальный признак: "teacher", "health" - care related, "services" - administrative or police, "at\_home", "other"

11. reason(причина выбора школы) - категориальный признак: "close to home", "school reputation", "course preference", "other"

12. guardian(опекун) - категориальный признак: "mother", "father", "other"

13. traveltime(сколько добираться до школы) - категориальный признак: "1" - < 15 минут, "2" - 15 минут <= traveltime <= 30 минут, "3" - 30 минут <= traveltime <= 1 час, "4" - > 1 час

14. studytime(сколько часов в неделю уделяется учёбе) - категориальный признак: "1" - < 2 часов, "2" - 2 часа <= studytime <= 5 часов, "3" - 5 часов <= traveltime <= 10 часов, "4" - > 10 часов

15.failures(сколько раз ученик оставался на второй год) - категориальный признак: "1" - 1 раз, "2" - 2 раза, "3" - 3 раза, "4" - > 3 раз

16. schoolsup(ученик имеет дополнительную образовательную поддержку) - бинарный признак: "yes" / "no"

17. famsup(ученик получает помощь в образовании со стороны семьи) - бинарный признак: "yes" / "no"

18. paid(ученик посещает дополнительные платные занятия по предмету курса "Math / Portuguese") - бинарный признак: "yes" / "no"

19. activities(участвует во внеклассной деятельности) - бинарный признак: "yes" / "no"

20. nursery(посещал детский сад) - бинарный признак: "yes" / "no"

21. higher(заинтересован в получении высшего образования) - бинарный признак: "yes" / "no"

22. internet(дома есть подключение к интернету) - бинарный признак: "yes" / "no"

23. romantic(состоит в романтических отношениях) - бинарный признак: "yes" / "no"

24. famrel(взаимоотношения в семье) - категориальный признак: шкала от 1 до 5, где "1" - very bad, "5" – excellent

25. freetime(свободное время после школы) - категориальный признак: шкала от 1 до 5, где "1" - very low, "5" - very high

26. goout(прогулки на улице) - категориальный признак: шкала от 1 до 5, где "1" - very low, "5" - very high

27. Dalc(употребление алкоголя в будний день) - категориальный признак: шкала от 1 до 5, где "1" - very low, "5" - very high

28. Walc(употребление алкоголя в выходные дни) - категориальный признак: шкала от 1 до 5, где "1" - very low, "5" - very high

29. health(состояние здоровья) - категориальный признак: шкала от 1 до 5, где "1" - very bad, "5" - very good

30.absences(количество прогулов занятий в школе) - числовой признак: 0 <= absences <= 93

31.G1(оценка за первый период) - числовой признак: 0 <= G1 <= 20

32.G2(оценка за второй период) - числовой признак: 0 <= G2 <= 20

33.G3(итоговая оценка) - числовой признак: 0 <= G3 <= 20

**2. Сколько категориальных признаков, какие?**

В данном Датасете 15 категориальных признаков:

1.Medu(образование матери)

2. Fedu(образование отца)

3.Mjob(работа матери)

4.Fjob(работа отца)

5.reason(причина выбора школы)

6.guardian(опекун)

7.traveltime(сколько добираться до школы)

8.studytime(сколько часов в неделю уделяется учёбе)

9.failures(сколько раз ученик оставался на второй год)

10.famrel(взаимоотношения в семье)

11.freetime(свободное время после школы)

12.goout(прогулки на улице)

13.Dalc(употребление алкоголя в будний день)

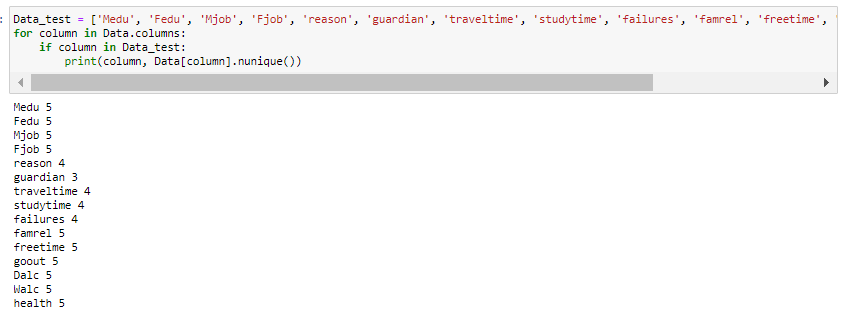
14.Walc(употребление алкоголя в выходные дни)

15. health(состояние здоровья)

**3.Столбец с максимальным количеством уникальных значений категориального признака?**

В данном Датасете 10 столбцов с максимальным количеством уникальных значений категориального признака (см. *Рисунок 6*):

1)Medu = 5; 2)Fedu = 5; 3)Mjob = 5; 4)Fjob = 5; 5)famrel = 5; 6)freetime = 5; 7)goout = 5; 8)Dalc = 5; 9)Walc = 5; 10)health = 5;



*Рисунок 6.* Результат применения функции nunique().

**4.Есть ли бинарные признаки?**

В данном Датасете 13 бинарных признаков:

1.school(школа)  
  
2.sex(пол)  
  
3.address(тип населённого пункта)

4.famsize(количество человек в семье)

5.Pstatus(состоят ли родители в браке)

6.schoolsup(ученик имеет дополнительную образовательную поддержку)

7.famsup(ученик получает помощь в образовании со стороны семьи)  
  
8.paid(ученик посещает дополнительные платные занятия по предмету курса "Math / Portuguese")  
  
9.activities(участвует во внеклассной деятельности)

10.nursery(посещал детский сад)

11.higher(заинтересован в получении высшего образования)

12.internett(дома есть подключение к интернету)

13.romantic(состоит в романтических отношениях)

**5.Какие числовые признаки?**

1.age(возраст)

2.absences(количество прогулов занятий в школе)

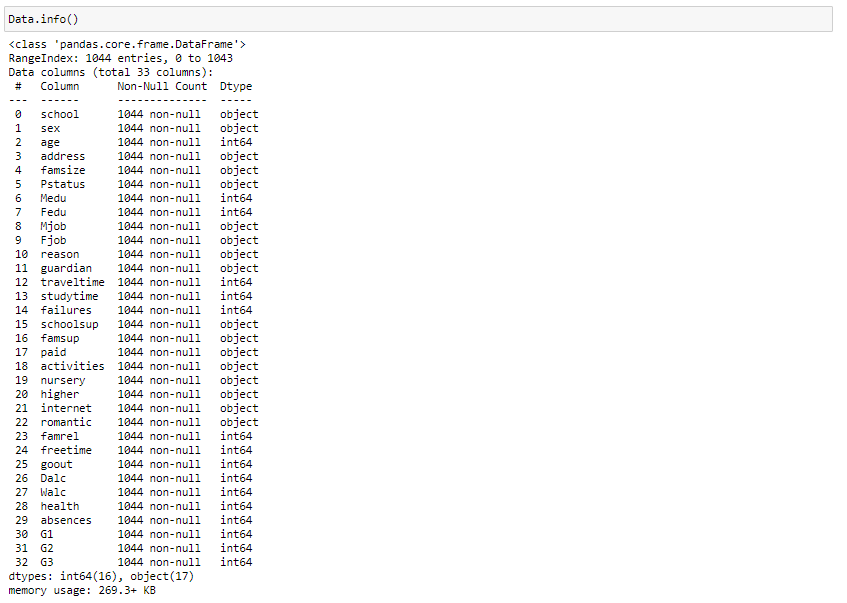
3.G1(оценка за первый период)

4.G2(оценка за второй период)

5.G3(итоговая оценка)

**6.Есть ли пропуски?**

Основываясь на результате info(), мы видим, что пропусков в Датасете нет (3-ий столбец) (см. *Рисунок 7*):



*Рисунок 7.* Результат применения функции info().

**7.Сколько объектов с пропусками?**

Ответ: ни одного.

**8.Столбец с максимальным количеством пропусков?**

Ответ: ни одного.

**9.Есть ли на ваш взгляд выбросы, аномальные значения?**

Ответ: аномальных значений и выбросов не наблюдается.

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

1. Про-нормируем числовые признаки через стандартные отклонения (см. *Рисунок 8*).
2. Преобразуем текстовые данные в числовые (см. *Рисунок 8*).



*Рисунок 8*. Нормируем числовые признаки через стандартные отклонения и преобразовываем текстовые данные в числовые.

1. Найдем максимальное среднее значение:

max\_average = -math.inf

for column in array:

if Data[column].mean() > max\_average:

max\_average = Data[column].mean()

print(column,max\_average)

Ответ: столбец с максимальным средним значением после нормировки признаков - romantic

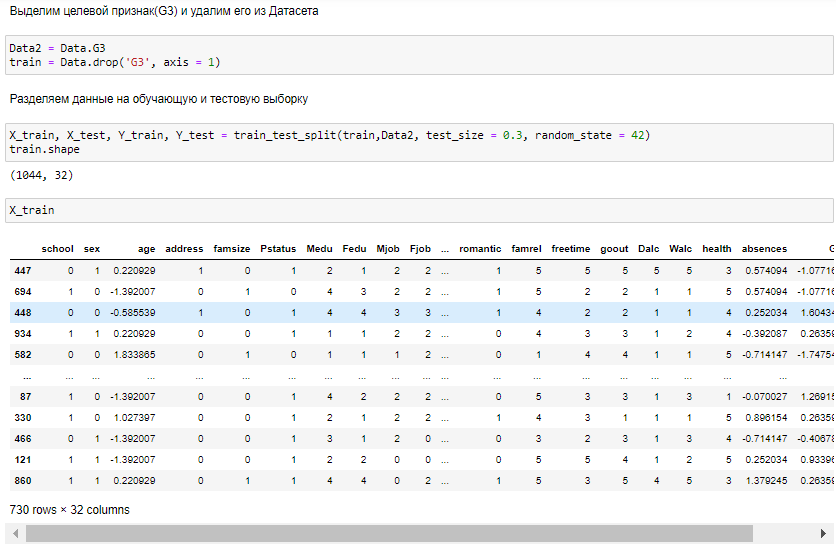
**11.Столбец с целевым признаком?**

Ответ: поскольку нам будет необходимо построить прогноз итогового бала учащегося, то целевой признак - G3.

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

1) Выделим целевой признак (G3) и удалим его из Датасета (см. *Рисунок 9*).

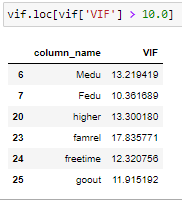
2)Разделим данные на обучающую и тестовую выборку (см. *Рисунок 9*).



*Рисунок 9*. Количество объектов в тестовой выборке, после удаления целевого признака (G3) из Датасета и разделения данных на обучающую и тестовую выборку.

**13.Между какими признаками наблюдается линейная зависимость (корреляция)?**

Проверим коэффициент VIF для всех признаков, и если он больше 10, то существует линейная зависимость (см. *Рисунок 10*).



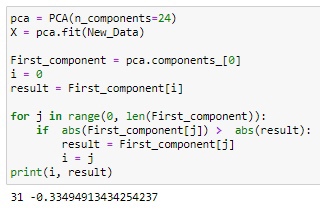
*Рисунок 10*. Результат применения VIF.

Ответ: Линейная зависимость наблюдается у следующий признаков: Medu, Fedu, higher, famrel, freetime, goout.

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

Ответ: Для объяснения 90% дисперсии - достаточно 24 признаков.

**15.Какой признак вносит наибольший вклад в первую компоненту?**



*Рисунок 11*. Поиск признака, вносящего наибольший вклад в первую компоненту.

Наибольший вклад в первую компоненту вносит 31-ый признак.

Ответ: это признак G2

Задача №6

Тема: "Употребление алкоголя учащимися"

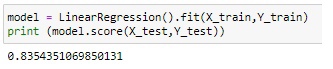
Набор данных №1:

<https://www.kaggle.com/uciml/student-alcohol-consumpt..>  
Набор данных №2:

[https://www.kaggle.com/uciml/student-alcohol-consumpt..](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fwww.kaggle.com%2Fuciml%2Fstudent-alcohol-consumption%3Fselect%3Dstudent-mat.csv)

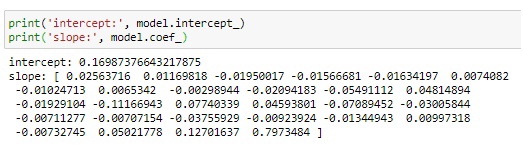
***№1.*** Построить прогноз итогового бала учащегося.

Построим модель линейной регрессии (см. *Рисунок 12*).



*Рисунок 12.* Построение линейной регрессии и результат выполнения функции score().#R2 = 0.8354351069850131

Найдем значения коэффициентов в нашей модели (см. *Рисунок 13*).



*Рисунок 13.* Нахождение значения коэффициентов в модели.

Тогда, линейная регрессия итогового бала (G3) на все параметры выглядит следующим образом:

G3 = 0.16987376643217875 + 0.02563716033429331 \* school +

0.01169817521405268 \* sex + (-0.019500166793284434) \* age +

(-0.015666814055066567) \* address + (-0.016341973485239178) \* famsize +

0.007408199119350586 \* Pstatus + (-0.010247132598149907 )\* Fedu +

0.006534203835179059 \* Mjob + (-0.002989437286947814 )\* Fjob +

(-0.020941826441747944 )\* reason + (-0.0549111209231241) \* guardian +

0.04814894291335772 \* traveltime + (-0.019291040061392817 )\* studytime +

(-0.11166943202709809) \* failures + 0.0774033856006938 \* schoolsup +

0.045938010732101374 \* famsup + (-0.07089451684886684) \* paid +

(-0.03005844014350527) \* activities + (-0.00711277454691548 )\* nursery +

(-0.007071539538965607) \* internet + (-0.03755928669946984) \* romantic +

(-0.009239242193155701) \* freetime + (-0.01344942615766682) \* Dalc +

0.00997318440366965 \* Walc + (-0.007327450384538271) \* health +

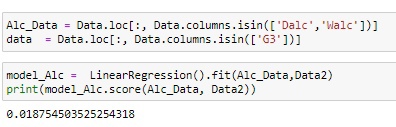
0.050217781870780465 \* absences + 0.12701636592953505 \* G1 +

0.7973484032613066 \* G2

***№2.*** Насколько сильно влияет употребление алкоголя?

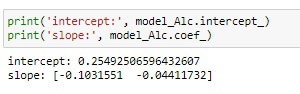
Чтобы выявить, насколько сильно употребление алкоголя влияет на итоговый бал учащегося, построим линейную регрессию G3(итоговый бал) на параметры Dalc(употребление алкоголя в будний день) и Walc(употребление алкоголя в выходные дни).

Построим модель линейной регрессии (см. *Рисунок 14*).



*Рисунок 14.* Построение линейной регрессии и результат выполнения функции score().#R2 = 0.018754503525254318

Найдем значения коэффициентов в нашей модели (см. *Рисунок 15*).



*Рисунок 15.* Нахождение значения коэффициентов в модели.

Тогда, линейная регрессия итогового бала (G3) на параметры Dalc и Walc выглядит следующим образом:

G3 = 0.25492506596432607 + (-0.10315510322966899) \* Dalc +

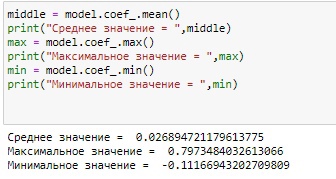
(-0.044117319844223485) \* Walc

Вывод: Как видно из данной модели, употребление алкоголя негативно сказывается на итоговом бале учащегося, причем употребление алкоголя в будние дни сильнее влияет на итоговую оценку, чем употребление алкоголя в выходные дни.

***№3.*** Какие параметры вносят наибольший вклад в предсказание?

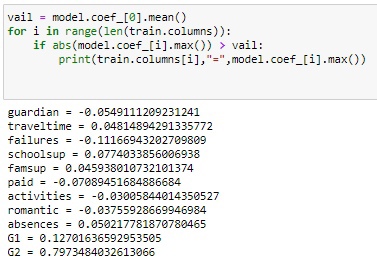
Хоть потребление алкоголя и негативно сказывается на итоговом бале учащегося, но коэффициенты перед параметрами Dalc(употребление алкоголя в будний день) и Walc(употребление алкоголя в выходные дни) довольно маленькие. Таким образом, эти параметры не вносят наибольший вклад в предсказание результата итоговой оценки.

Вычислим среднее, максимальное и минимальное значение среди коэффициентов модели (см. *Рисунок 16*).



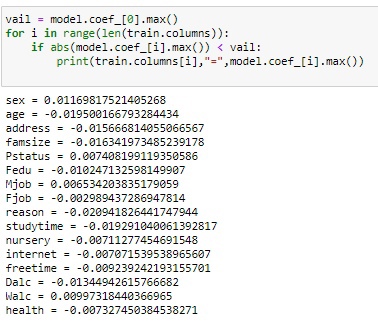
*Рисунок 16*. Вычисление среднего, максимального и минимального значения среди коэффициентов модели.

Найдем параметры, вносящие наибольший вклад в предсказание (см. *Рисунок 17*).



*Рисунок 17*. Параметры, которые вносят наибольший вклад в предсказание.

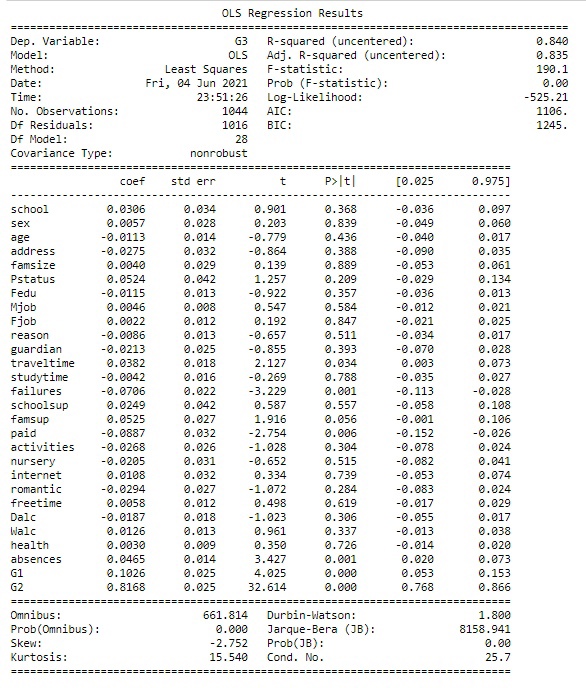
Найдем параметры, который вносят далеко не основной вклад в предсказание результата итоговой оценки (см. *Рисунок 18*).



*Рисунок 18*. Параметры, который вносят далеко не основной вклад в предсказание результата итоговой оценки.

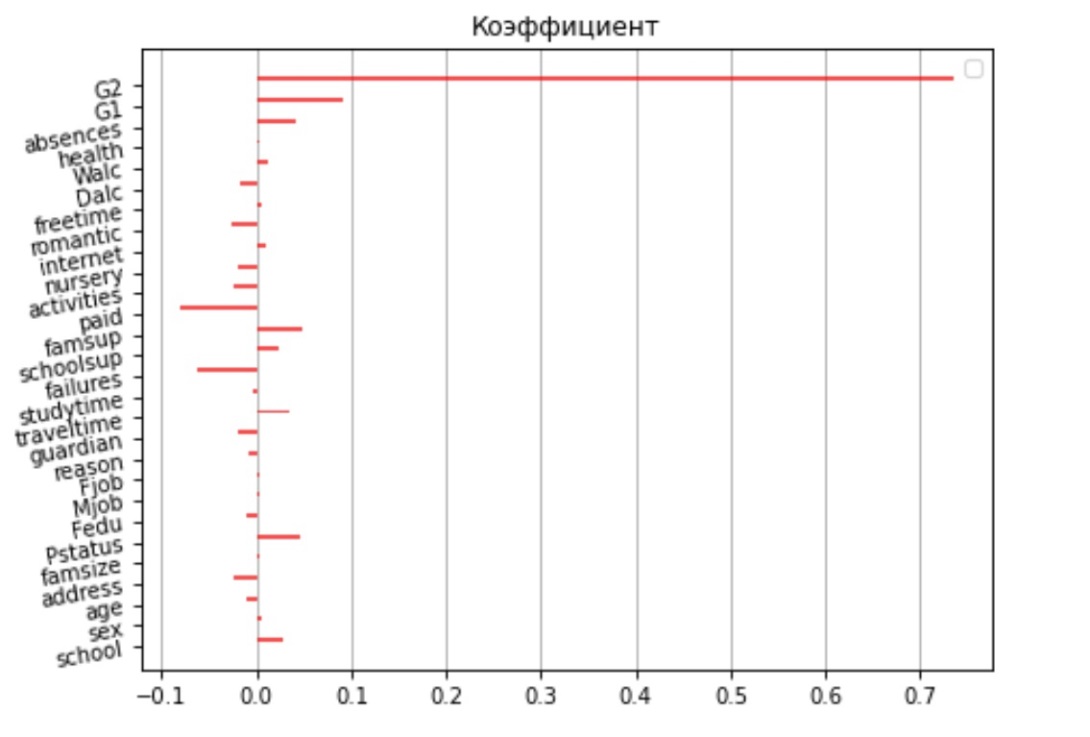
***№4.*** Построить графики статистической оценки параметров.

Выведем статистику модели в виде таблицы:

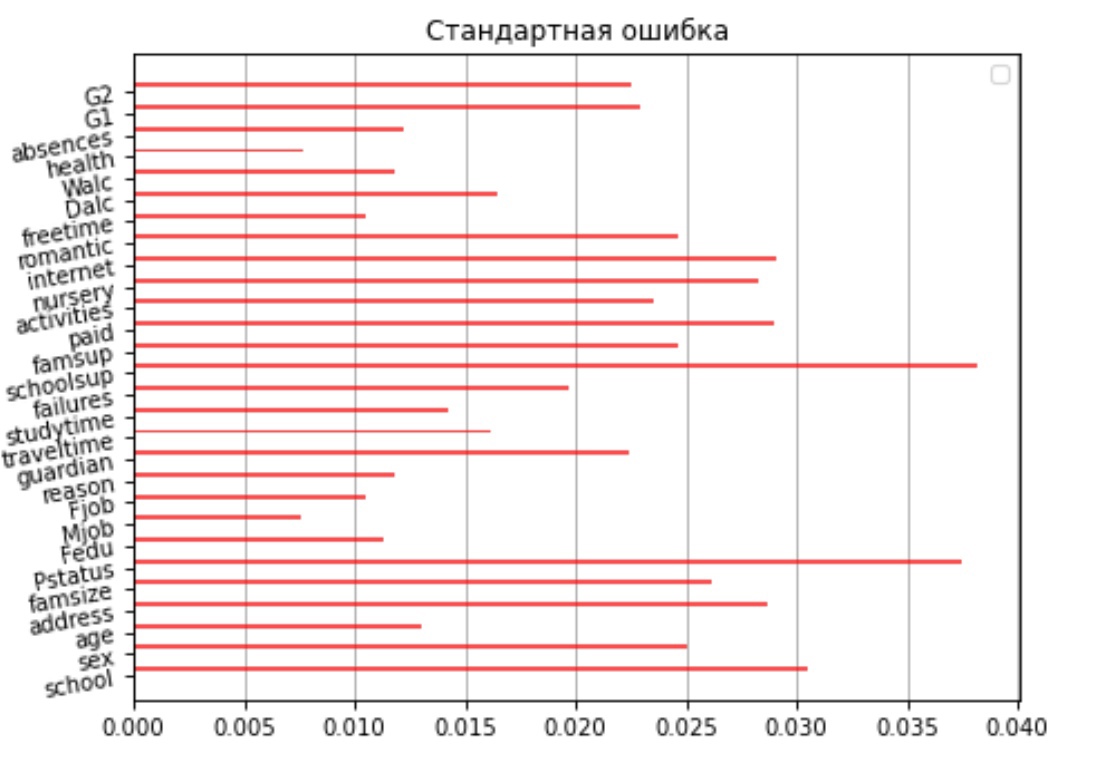


*Рисунок 19*. Статистика линейной регрессии итогового бала (G3) на все параметры.

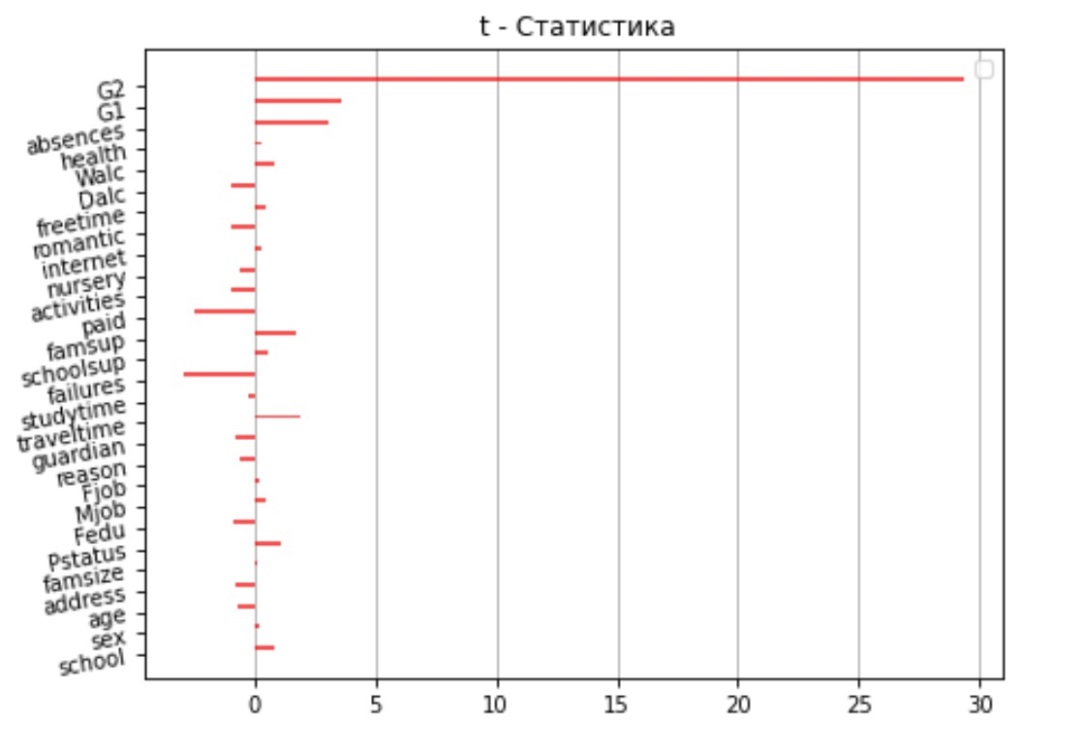
Построим диаграммы:



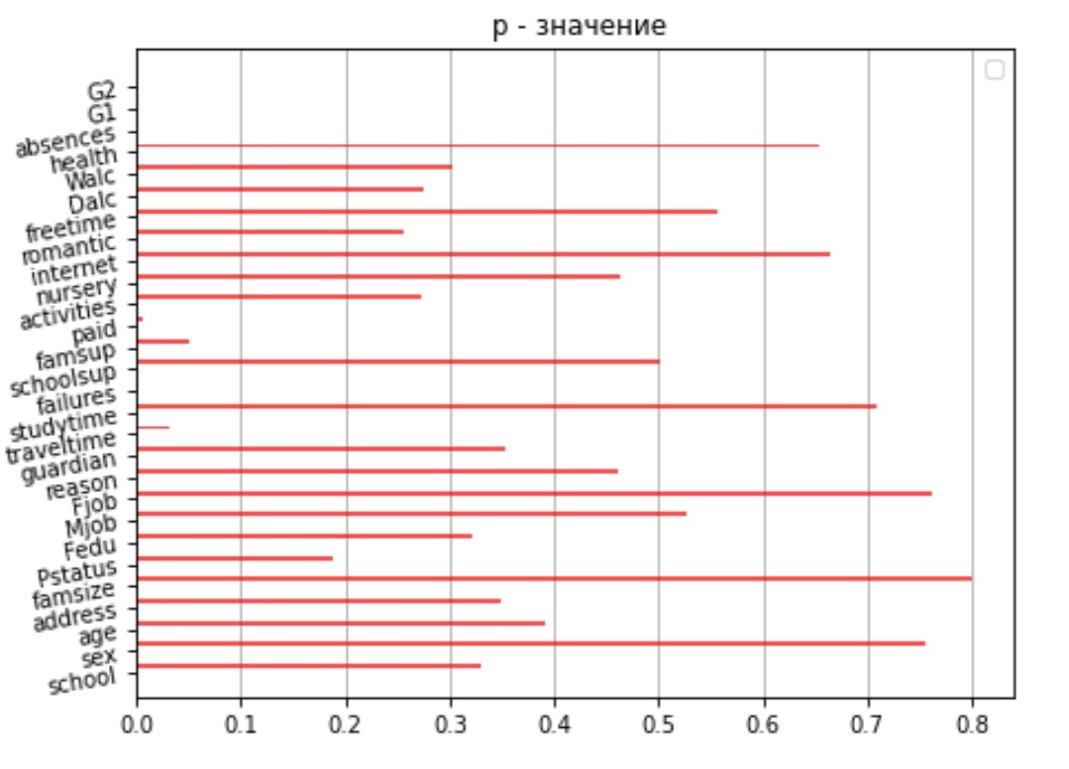
*Рисунок 20*. Графики статистической оценки параметра: “Коэффициент”.



*Рисунок 21*. Графики статистической оценки параметра: “Стандартная ошибка”.

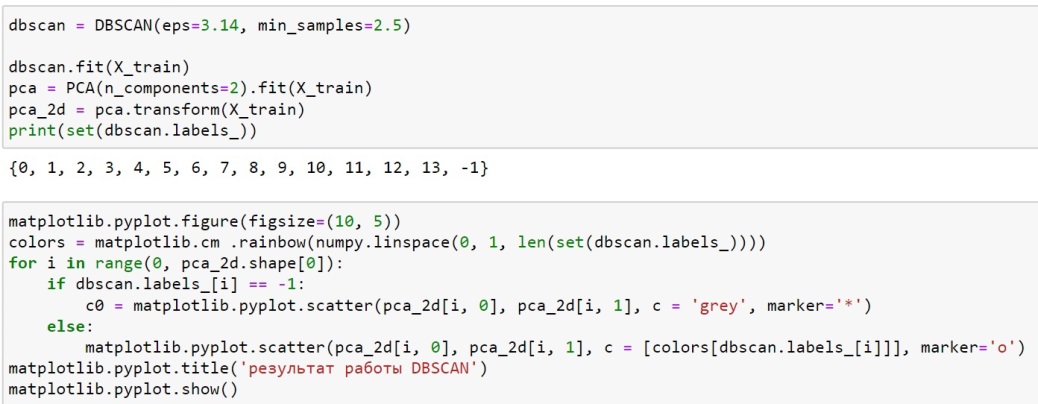


*Рисунок 22*. Графики статистической оценки параметра: “t - Статистика”.

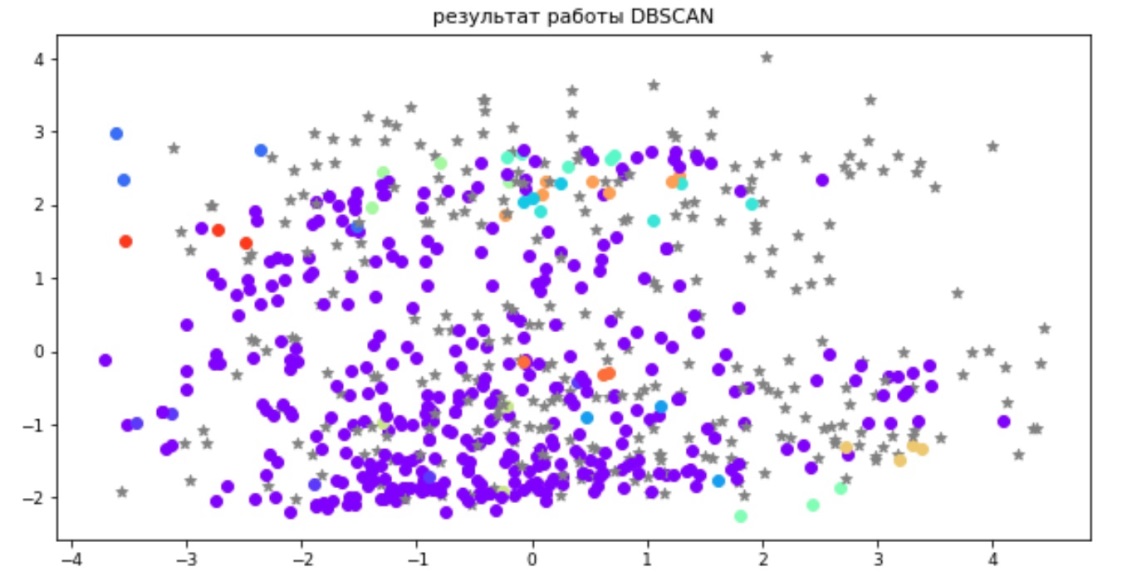


*Рисунок 23*. Графики статистической оценки параметра: “p - Значение”.

***№5.*** Применить алгоритмы кластеризации.



*Рисунок 24*. Применяем алгоритмы кластеризации.



*Рисунок 25*. Результат работы DBSCAN.

Вывод: Как видно из графика, у данной модели нет явных кластеров.

***№6.*** Что общего между объектами в каждом кластере?

Ответ: Нельзя найти общие признаки у кластеров, так как кластеров нет.

Заключение

Приложение1

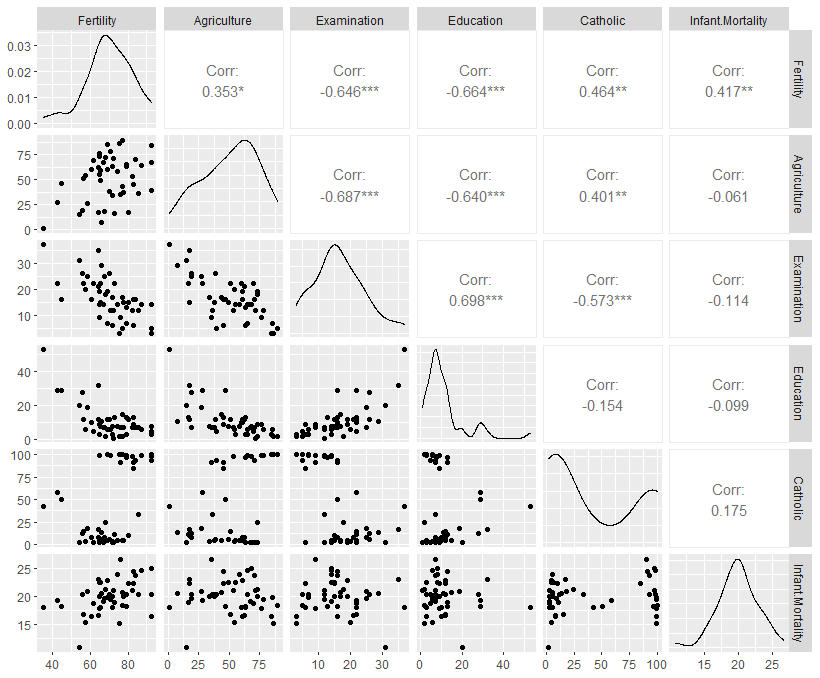


Рисунок 1. Результат работы команды ggpairs() – графики зависимостей между парами

переменных наборе данных Swiss.

*Приложение2*

*Таблица 1*. Характеристики модели зависимости параметра *Agriculture* от параметров *Examination* и *Infant.Mortality*  в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 105.56 | 18.29 | 5.77 | 7.32e-07 | \*\*\* |
| *Examination* | -2 | 0.3 | -6.49 | 6.42e-08 | \*\*\* |
| *Infant.Mortality* | -1.1 | 0.84 | -1.3 | 0.2 |  |

*Таблица 2*. Характеристики модели зависимости параметра *Examination* от параметров *Agriculture* и *Infant.Mortality*  в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 37.42 | 6.33 | 5.91 | 4.53e-07 | \*\*\* |
| *Agriculture* | -0.24 | 0.04 | -6.49 | 6.42e-08 | \*\*\* |
| *Infant.Mortality* | -0.43 | 0.29 | -1.46 | 0.15 |  |

*Таблица 3*. Характеристики модели зависимости параметра *Infant.Mortality* от параметров *Agriculture* и *Examination* в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 23.43 | 2.36 | 9.92 | 8.6e-13 | \*\*\* |
| *Agriculture* | -0.03 | 0.03 | -1.3 | 0.2 |  |
| *Examination* | -0.1 | 0.07 | -1.46 | 0.15 |  |

*Таблица 4*. Характеристики модели зависимости параметра *Fertility* от параметров *Agriculture*, *Examination* и *Infant.Mortality* в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 60.87 | 12.83 | 4.75 | 2.32e-05 | \*\*\* |
| *Examination* | -1.04 | 0.23 | -4.56 | 4.22e-05 | \*\*\* |
| *Infant.Mortality* | 1.44 | 0.46 | 3.16 | 0.003 | \*\* |
| *Agriculture* | -0.05 | 0.08 | -0.57 | 0.57 |  |

*Таблица 5*. Характеристики модели зависимости параметра *Fertility* от параметров *log(Examination),* и log(*Infant.Mortality)*  в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 6.35 | 25.49 | 0.25 | 0.8 |  |
| *log(Examination)* | -12.76 | 2.22 | -5.75 | 7.87e-07 | \*\*\* |
| *log(Infant.Mortality)* | 32.79 | 8.33 | 3.94 | 0.0003 | \*\*\* |

*Таблица 6*. Характеристики модели зависимости параметра log(*Fertility)* от параметров *log(Examination),* и log(*Infant.Mortality)* в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 3.26 | 0.41 | 7.85 | 6.85e-10 | \*\*\* |
| *log(Examination)* | -0.19 | 0.04 | -5.34 | 3.13e-06 | \*\*\* |
| *log(Infant.Mortality)* | 0.5 | 0.14 | 3.68 | 0.0006 | \*\*\* |

*Таблица 8*. Характеристики модели зависимости параметра *Fertility* от параметров *Examination,* и log(*Infant.Mortality)*  в наборе данных Swiss.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 8.53 | 25.61 | 0.33 | 0.74 |  |
| *Examination* | -0.95 | 0.16 | -5.73 | 8.33e-07 | \*\*\* |
| *log(Infant.Mortality)* | 2.89 | 8.41 | 3.08 | 0.004 | \*\* |

*Таблица 10*. Проверка на линейную зависимость между регрессоров (Examination, I(Examination^2), I(Infant.Mortality^2), I(Examination\*Infant.Mortality)) с помощью команды VIF.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Examination | I(Examination^2) | I(Infant.Mortality^2) | I(Examination\*  Infant.Mortality) |
| VIF | 35.28 | 14 | 3.77 | 27.15 |

*Таблица 11*. Проверка на линейную зависимость между регрессоров (I(Examination^2), I(Infant.Mortality^2), I(Examination\*Infant.Mortality)) с помощью команды VIF.

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр \ Характеристики | I(Examination^2) | I(Infant.Mortality^2) | (Examination\*  Infant.Mortality) |
| VIF | 10.29 | 2.42 | 10.49 |

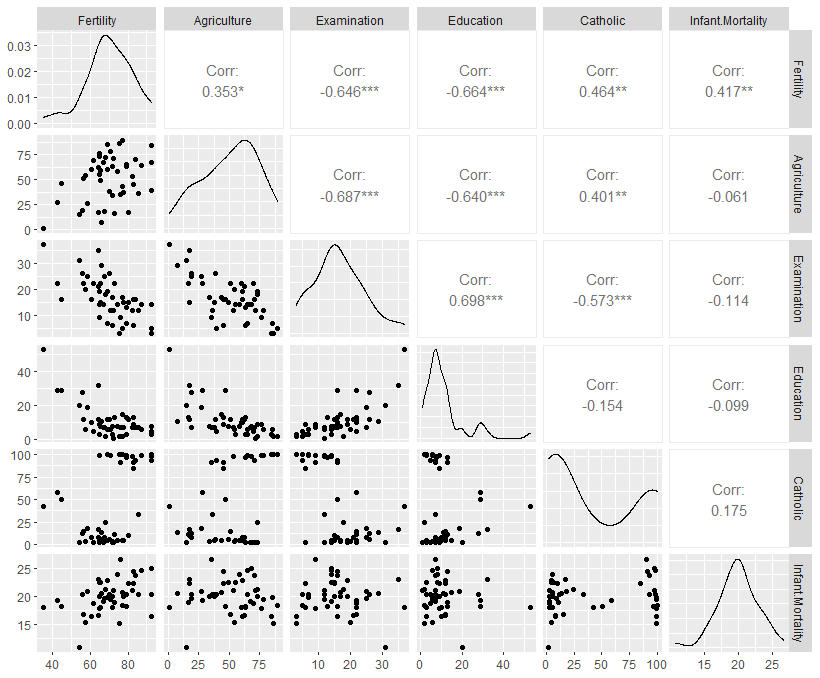


Рисунок 2. Результат работы команды ggpairs() – графики зависимостей между парами

переменных наборе данных Swiss.

*Приложение3*

*Таблица 16*. Характеристики модели зависимости параметра *salary* от параметров *age, sex, higher\_educ, status2, dur, wed1,wed2,wed3* в наборе данных обследования РМЭЗ НИУ ВШЭ.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости | Можно ли исключить регрессор? |
| (Intercept) | -0.74 | 0.05 | -15.85 | <2e-16 | \*\*\* | Да |
| *age* | -0.08 | 0.02 | -5.09 | 3.79e-07 | \*\*\* | Нет |
| *sex* | 0.5 | 0.03 | 17.17 | <2e-16 | \*\*\* | Нет |
| *higher\_educ* | 0.56 | 0.03 | 18.6 | <2e-16 | \*\*\* | Да |
| *status2* | 0.36 | 0.03 | 11.53 | <2e-16 | \*\*\* | Да |
| *dur* | 0.11 | 0.01 | 7.94 | 2.60e-15 | \*\*\* | Нет |
| *wed1* | 0.11 | 0.04 | 2.56 | 0.01 | \* | Да |
| *wed2* | 0.13 | 0.05 | 2.38 | 0.02 | \* | Да |
| *wed3* | -0.1 | 0.05 | -1.82 | 0.07 | . | Да |

*Таблица 17*. Характеристики модели зависимости параметра *salary* от параметров *age, sex, dur* в наборе данных обследования РМЭЗ НИУ ВШЭ.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | -0.19 | 0.02 | -9.5 | <2e-16 | \*\*\* |
| *age* | -0.08 | 0.01 | -5.38 | 7.93e-08 | \*\*\* |
| *sex* | 0.42 | 0.03 | 14.08 | <2e-16 | \*\*\* |
| *dur* | 0.08 | 0.02 | 5.52 | 3.59e-08 | \*\*\* |

*Таблица 18*. Характеристики модели зависимости параметра *salary* от параметров *age, sex, dur, log(age), log(dur)* в наборе данных обследования РМЭЗ НИУ ВШЭ.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 0.19 | 0.19 | 1.02 | 0.3 |  |
| *age* | -0.52 | 0.13 | -3.93 | 9.31e-05 | \*\*\* |
| *log(age)* | 009 | 0.07 | 1.29 | 0.2 |  |
| *sex* | 0.4 | 0.08 | 5.15 | 3.51e-07 | \*\*\* |
| *dur* | 0.07 | 0.07 | 1.06 | 0.29 |  |
| *log(dur)* | 0.01 | 0.09 | 0.13 | 0.9 |  |

*Таблица 19*. Характеристики модели зависимости параметра *salary* от параметров *age, sex, dur, log(age)* в наборе данных обследования РМЭЗ НИУ ВШЭ.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 0.05 | 0.08 | 0.61 | 0.54 |  |
| *age* | -0.29 | 0.07 | -4.43 | 1.01e-05 | \*\*\* |
| *log(age)* | 0.032 | 004 | 0.89 | 0.37 |  |
| *sex* | 0.32 | 0.04 | 7.73 | 1.68e-14 | \*\*\* |
| *dur* | 0.09 | 0.02 | 4.76 | 2.05e-06 | \*\*\* |

*Таблица 22*. Характеристики модели зависимости параметра *salary* от параметров *sex, dur, log(age)* в наборе данных обследования РМЭЗ НИУ ВШЭ.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | -0.27 | 0.03 | -9.53 | <2e-16 | \*\*\* |
| *log(age)* | -0.1 | 0.02 | -5.4 | 7.18e-08 | \*\*\* |
| *sex* | 0.32 | 0.04 | 9.64 | 3.33-14 | \*\*\* |
| *dur* | 0.1 | 0.02 | 5.07 | 4.29e-07 | \*\*\* |

*Таблица 23*. Характеристики модели зависимости параметра *salary* от параметров *age, sex, dur, I(age0.1), I(dur0.1)* в наборе данных обследования РМЭЗ НИУ ВШЭ.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | -1.14 | 1.1 | -1.03 | 0.3 |  |
| *age* | 0.054 | 0.15 | -362 | 0.0003 | \*\*\* |
| *I(age0.1)* | 1.1 | 0.88 | 1.26 | 0.2 |  |
| *sex* | 0.4 | 0.08 | 5.1 | 3.68e-07 | \*\*\* |
| *dur* | 0.06 | 0.07 | 0.8 | 0.5 |  |
| *I(dur0.1)* | 0.26 | 0.96 | 0.27 | 0.79 |  |

*Таблица 26*. Характеристики модели зависимости параметра *salary* от параметров *sex, dur, I(age0.1)* в наборе данных обследования РМЭЗ НИУ ВШЭ.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 0.9 | 0.2 | 4.57 | 5.29e-06 | \*\*\* |
| *I(age0.1)* | -1.2 | 0.21 | -5.69 | 1.44e-08 | \*\*\* |
| *sex* | 0.32 | 0.04 | 7.66 | 2.91e-14 | \*\*\* |
| *dur* | 0.1 | 0.02 | 5.02 | 5.67e-07 | \*\*\* |

*Таблица 28*. Характеристики модели зависимости параметра *salary* от параметров *sex, dur, I(age2)* для подмножества индивидов: “Городские жители, состоящие в браке” в наборе данных обследования РМЭЗ НИУ ВШЭ.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | 0.05 | 0.04 | 1.17 | 0.24 |  |
| *I(age2)* | 0.1 | 0.03 | 3.53 | 0.0004 | \*\*\* |
| *sex* | 0.65 | 0.05 | 12.43 | <2e-16 | \*\*\* |
| *dur* | -0.18 | 0.03 | -7.03 | 1.97e-12 | \*\*\* |

*Таблица 29*. Характеристики модели зависимости параметра *salary* от параметров *sex, dur, I(age2)* для подмножества индивидов: “ Разведенные, без высшего образования” в наборе данных обследования РМЭЗ НИУ ВШЭ.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр \ Характеристики | Значение | Std. Error | t value | Pr(>|t|) | Уровень  значимости |
| (Intercept) | -0.26 | 0.04 | -5.9 | 7.22e-09 | \*\*\* |
| *I(age2)* | 0.03 | 0.03 | 2.78 | 0.0006 | \*\* |
| *sex* | 0.08 | 0.08 | 3.2 | 0.002 | \*\* |
| *dur* | -0.1 | 0.02 | -4.44 | 1.13e-15 | \*\*\* |

*Приложение*

Код решения всех задач:

Задача №1.

library("lmtest")

library("GGally")

library("car")

data = swiss

help(swiss)

data

summary(data)

ggpairs(data)

model\_agr\_ex = lm(Agriculture~Examination, data)

model\_agr\_ex

summary(model\_agr\_ex)

plot(model\_agr\_ex) + abline(a = 82.88, b = -1.95, col = "red")

model\_agr\_F = lm(Agriculture~Fertility, data)

model\_agr\_F

summary(model\_agr\_F)

plot(model\_agr\_F) + abline(a = 5.63, b = 0.64, col = "red")

var(data$Agriculture)

sd(data$Agriculture)

mean(data$Agriculture)

var(data$Examination)

sd(data$Examination)

mean(data$Examination)

var(data$Fertility)

sd(data$Fertility)

mean(data$Fertility)

Задача №2.1.

library("lmtest")

library("GGally")

library("car")

data = swiss

help(swiss)

vif(model)

model\_test\_1 = lm(Agriculture ~ Examination + Infant.Mortality, data) model\_test\_1

summary(model\_test\_1

model\_test\_2 = lm(Examination ~ Agriculture + Infant.Mortality, data) model\_test\_2

summary(model\_test\_2)

model\_test\_3 = lm(Infant.Mortality ~ Agriculture + Examination , data)

model\_test\_3

summary(model\_test\_3)

model = lm(Fertility ~ Agriculture + Examination + Infant.Mortality, data)

model

summary(model)

model = lm(Fertility ~ Examination + Infant.Mortality, data)

model

summary(model)

model = lm(Fertility ~ log(Examination) + log(Infant.Mortality), data)

model

summary(model)

vif(model)

model = lm(log(Fertility) ~ log(Examination) + log(Infant.Mortality), data

model

summary(model)

vif(model)

model = lm(Fertility ~ log(Examination) + Infant.Mortality, data)

model

summary(model)

vif(model)

model = lm(Fertility ~ Examination + log(Infant.Mortality), data)

model

summary(model)

vif(model)

model\_1 = lm(Fertility ~ Examination + Infant.Mortality + I(Examination^2) + I(Infant.Mortality^2) + I(Examination\*Infant.Mortality), data)

model\_1

summary(model\_1)

vif(model\_1)

model\_2 = lm(Fertility ~ Examination + I(Examination^2) + I(Infant.Mortality^2) + I(Examination\*Infant.Mortality), data)

model\_2

summary(model\_2)

vif(model\_2)

model\_3 = lm(Fertility ~ I(Examination^2) + I(Infant.Mortality^2) + I(Examination\*Infant.Mortality), data)

model\_3

summary(model\_3)

vif(model\_3)

model\_4 = lm(Fertility ~ I(Examination^2) + I(Infant.Mortality^2), data)

model\_4

summary(model\_4)

vif(model\_4)

Задача №2.2

library("lmtest")

library("GGally")

library("car")

data = swiss

help(swiss)

model = lm(Fertility ~ Agriculture + Examination + Infant.Mortality, data)

model

summary(model)

se = 0.07975

t = qt(0.975, df = 43)

model$coefficients[2] - t \* se

model$coefficients[2] + t \* se

confint(model, level = 0.95)

se = 0.22811

t = qt(0.975, df = 43)

model$coefficients[3] - t \* se

model$coefficients[3] + t \* se

confint(model, level = 0.95)

se = 0.45513

t = qt(0.975, df = 43)

model$coefficients[4] - t \* se

model$coefficients[4] + t \* se

confint(model, level = 0.95)

se = 12.82691

t = qt(0.975, df = 43)

model$coefficients[1] - t \* se

model$coefficients[1] + t \* se

confint(model, level = 0.95)

model = lm(Fertility ~ Agriculture + Examination + Infant.Mortality, data)

model

summary(model)

new.data = data.frame(Agriculture = 10, Examination = 30, Infant.Mortality = 20)

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

Задача №3.

data\_2 = select(data\_1, salary, age, sex, higher\_educ, status2, dur, wed1,wed2,wed3)

data\_2 = na.omit(data\_2)

glimpse(data\_2)

model1 = lm(data = data\_2, salary~age + sex + higher\_educ + status2 + dur + wed1 + wed2 + wed3)  
summary(model1)  
vif(model1)  
  
model2 = lm(data = data\_2, salary~age + sex + dur)

summary(model2)   
vif(model2)   
  
model3 = lm(salary~age+log(age) + sex + dur + log(dur), data = data\_2)

summary(model3)    
vif(model3)

model4 = lm(salary~age + log(age) + sex + dur, data = data\_2)

summary(model4)    
vif(model4)

model5 = lm(salary~log(age) + sex + dur, data = data\_2)

summary(model5)    
vif(model5)   
  
model6 = lm(salary~age + I(age^0.1) + sex + dur+ I(dur^0.1),data = data\_2)

summary(model6)

vif(model6)

model7 = lm(salary~age + I(age^0.1) + sex + dur, data = data\_2)

summary(model7)    
vif(model7)   
  
model8 = lm(salary~I(age^0.1) + sex + dur, data = data\_2)

summary(model8)    
vif(model8)   
  
model9 = lm(salary~I(age^0.2) + sex + dur, data = data\_2)

summary(model9)    
vif(model9)   
  
model10 = lm(salary~I(age^0.3) + sex + dur, data = data\_2)

summary(model10)    
vif(model10)

model11 = lm(salary~I(age^0.4) + sex + dur, data = data\_2)

summary(model11)

vif(model11)   
  
model12 = lm(salary~I(age^0.5) + sex + dur, data = data\_2)

summary(model12)    
vif(model12)

model13 = lm(salary~I(age^0.9) + sex + dur, data = data\_2)

summary(model13)    
vif(model13)   
  
model14 = lm(salary~I(age^1.1) + sex + dur, data = data\_2)

summary(model14)    
vif(model14)

model15 = lm(salary~I(age^1.2) + sex + dur, data = data\_2)

summary(model15)    
vif(model15)   
  
model16 = lm(salary~I(age^1.3) + sex + dur, data = data\_2)

summary(model16)    
vif(model16)   
  
model17 = lm(salary~I(age^1.4) + sex + dur, data = data\_2)

summary(model17)    
vif(model7)

model18 = lm(salary~I(age^1.9) + sex + dur, data = data\_2)

summary(model18)    
vif(model18)

model19 = lm(salary~I(age^2) + sex + dur, data = data\_2)

summary(model19)    
vif(model19)

data\_3 = subset(data\_2, status2 == 1)

data\_3  
data\_4 = subset(data\_3, wed1 == 1)

data\_4  
data\_5 = subset(data\_2, wed2 == 1)

data\_5  
data\_6 = subset(data\_5, higher\_educ == 0)

data\_6

model\_subset1 = lm(data = data\_4, salary~dur + sex + I(age^2))

summary(model\_subset1)  
  
model\_subset2 = lm(data = data\_6, salary~dur + sex + I(age^2))

summary(model\_subset2)

Задача №4.

#Устанавливаем необходимые библиотеки:

import pandas

import numpy

import math

from sklearn.tree import DecisionTreeClassifier

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

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

from sklearn.model\_selection import GridSearchCV

from sklearn.tree import plot\_tree

from matplotlib import pyplot as plt

data = pandas.read\_csv('aug\_test.csv', index\_col='city')

data = data[data.columns.drop(['enrollee\_id'])]

data.head()

data.info()

data['gender'] = numpy.where(data['gender'] == 'Male', 1, 0)  
data['education\_level'] = numpy.where(data['education\_level'] == 'Graduate', 0, 1)  
data['relevent\_experience'] = numpy.where(data['relevent\_experience'] == 'No relevent experience',0,1)

array = ['enrolled\_university', 'last\_new\_job', 'major\_discipline', 'company\_type', 'company\_size', 'experience']

for column in array:

Set = set(data[column])

i = 0

for item in

Set:data[column] = data[column].replace(item, i)

i = i + 1

data.loc[data.experience == 'NaN' , 'experience'] = 0

data.loc[data.last\_new\_job == 'never' , 'last\_new\_job'] = 0

data.loc[data.last\_new\_job == 'NaN' , 'major\_discipline'] = 0

data.loc[data.last\_new\_job == 'NaN' , 'company\_size'] = 0

data

data.info()

Data2 = data.education\_level

train = data.drop('education\_level', axis = 1)

train.info()  
data.columns

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(train,Data2, test\_size = 0.3, random\_state = 42)  
train.shape

Tree = DecisionTreeClassifier(random\_state=42, max\_depth = 4)

Tree = Tree.fit(X\_train, Y\_train)

Tree

fig = plt.figure(figsize=(30,30))

\_ = plot\_tree(Tree, filled=True, class\_names=True)

print("f1:"+str(numpy.average(cross\_val\_score(Tree, X\_train, Y\_train, scoring='f1'))))

print("precision:"+str(numpy.average(cross\_val\_score(Tree, X\_train, Y\_train,

scoring='precision'))))  
print("recall:"+str(numpy.average(cross\_val\_score(Tree, X\_train, Y\_train, scoring='recall'))))

print("f1:"+str(numpy.average(cross\_val\_score(Tree, X\_test, Y\_test, scoring='f1'))))  
print("precision:"+str(numpy.average(cross\_val\_score(Tree, X\_test, Y\_test, scoring='precision'))))  
print("recall:"+str(numpy.average(cross\_val\_score(Tree, X\_test, Y\_test, scoring='recall'))))

tree = DecisionTreeClassifier(class\_weight=None, criterion='gini', max\_depth=None,

max\_features=None, max\_leaf\_nodes=None, min\_impurity\_decrease=0, min\_samples\_leaf=2, min\_samples\_split=3, min\_weight\_fraction\_leaf=0, random\_state=35, splitter='best')  
  
params =

{  
'max\_depth':

list(range(1, 25)),

'min\_samples\_split': [2, 3, 4, 5, 6, 7, 8, 9, 10]

}  
  
gridsearch = GridSearchCV(cv=3, error\_score='raise-deprecating',

estimator=tree,n\_jobs=-1,param\_grid=params, pre\_dispatch='2\*n\_jobs', refit=True, return\_train\_score='warn',verbose=1)  
  
  
gridsearch.fit(X\_train, Y\_train)

Tree = gridsearch.best\_estimator\_

Tree

print("f1:"+str(numpy.average(cross\_val\_score(Tree, X\_train, Y\_train, scoring='f1'))))

print("precision:"+str(numpy.average(cross\_val\_score(Tree, X\_train, Y\_train, scoring='precision'))))

print("recall:"+str(numpy.average(cross\_val\_score(Tree, X\_train, Y\_train, scoring='recall'))))

print("f1:"+str(numpy.average(cross\_val\_score(Tree, X\_test, Y\_test, scoring='f1'))))

print("precision:"+str(numpy.average(cross\_val\_score(Tree, X\_test, Y\_test,

scoring='precision'))))  
print("recall:"+str(numpy.average(cross\_val\_score(Tree, X\_test, Y\_test, scoring='recall'))))

fig = plt.figure(figsize=(30,30))

\_ = plot\_tree(Tree, filled=True, class\_names=True)

from sklearn.ensemble import RandomForestClassifier

param\_grid = { 'n\_estimators': [50, 100, 150],'max\_features': ['auto'],'max\_depth' : list(range(1,

10)), 'criterion' :['gini']}

RandForCrit = GridSearchCV(estimator = RandomForestClassifier(), param\_grid =

param\_grid, cv = 5, refit = True)

RandForCrit.fit(X\_train, Y\_train)

RandForCrit.predict(X\_test)

print("f1:"+str(numpy.average(cross\_val\_score(Tree, X\_train, Y\_train, scoring='f1'))))

print("precision:"+str(numpy.average(cross\_val\_score(Tree, X\_train, Y\_train,

scoring='precision'))))  
print("recall:"+str(numpy.average(cross\_val\_score(Tree, X\_train, Y\_train, scoring='recall')))

print("f1:"+str(numpy.average(cross\_val\_score(Tree, X\_test, Y\_test, scoring='f1'))))

print("precision:"+str(numpy.average(cross\_val\_score(Tree, X\_test, Y\_test,

scoring='precision'))))  
print("recall:"+str(numpy.average(cross\_val\_score(Tree, X\_test, Y\_test, scoring='recall'))))

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

RFC.fit(X\_train, Y\_train)

len(RFC.best\_estimator\_.estimators\_)

Задача №5.

!pip install pandas  
!pip install sklearn  
!pip install statsmodels

import pandas  
import numpy  
import math  
from sklearn.model\_selection import train\_test\_split  
from statsmodels.stats.outliers\_influence import variance\_inflation\_factor  
from sklearn import preprocessing  
from sklearn.decomposition import PCA

data1 = pandas.read\_csv('student-por.csv')  
data2 = pandas.read\_csv('student-mat.csv')  
Data = [data1, data1]  
Data = pandas.concat(Data)

Data

data1.info()  
data2.info()

Data = data1.append(data2, ignore\_index=True)

Data

Data.info()

Data.columns

Data\_test = ['Medu', 'Fedu', 'Mjob', 'Fjob', 'reason', 'guardian', 'traveltime', 'studytime', 'failures', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc', 'health']  
for column in Data.columns:  
if column in Data\_test:  
print(column, Data[column].nunique())

Data.info()

for column in Data.columns:  
print(Data[column].value\_counts())

Data['age'] = (Data['age'] - Data['age'].mean())/(math.sqrt(Data['age'].var()))

Data['absences'] = (Data['absences'] - Data['absences'].mean())/(math.sqrt(Data['absences'].var()))

Data['G1'] = (Data['G1'] - Data['G1'].mean())/(math.sqrt(Data['G1'].var()))

Data['G2'] = (Data['G2'] - Data['G2'].mean())/(math.sqrt(Data['G2'].var()))

Data['G3'] = (Data['G3'] - Data['G3'].mean())/(math.sqrt(Data['G3'].var()))

array = ['school', 'sex', 'address', 'famsize', 'Pstatus', 'Mjob', 'Fjob', 'reason', 'guardian', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'romantic']  
for column in array:  
Set = set(Data[column])  
i = 0  
for item in Set:  
Data[column] = Data[column].replace(item, i)  
i = i + 1

Data

max\_average = -math.inf  
for column in array:  
if Data[column].mean() > max\_average:  
max\_average = Data[column].mean()  
print(column,max\_average)

Data2 = Data.G3  
train = Data.drop('G3', axis = 1)

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(train,Data2, test\_size = 0.3, random\_state = 42)  
train.shape

X\_train

vif = pandas.DataFrame()  
vif["column\_name"] = train.columns  
vif["VIF"] = [variance\_inflation\_factor(train.values, i) for i in range(len(train.columns))]

vif.loc[vif['VIF'] > 10.0]

New\_Data = pandas.DataFrame(preprocessing.scale(X\_train),columns = X\_train.columns)

for i in range(1, len(New\_Data.columns)):  
pca = PCA(n\_components=i)  
pca.fit(New\_Data)  
print(i, sum(pca.explained\_variance\_ratio\_))

pca = PCA(n\_components=24)  
X = pca.fit(New\_Data)  
  
First\_component = pca.components\_[0]  
i = 0  
result = First\_component[i]  
  
for j in range(0, len(First\_component)):  
if abs(First\_component[j]) > abs(result):  
result = First\_component[j]  
i = j  
print(i, result)

New\_Data.iloc[:, 31]

Задача №6.

!pip install pandas  
!pip install sklearn  
!pip install statsmodels

import pandas  
import numpy  
import math  
from sklearn.cluster import DBSCAN  
from sklearn.decomposition import PCA  
import matplotlib.cm  
from statsmodels.stats.outliers\_influence import variance\_inflation\_factor  
import matplotlib.pyplot  
import statsmodels.api  
from sklearn.linear\_model import LinearRegression  
from sklearn.model\_selection import train\_test\_split  
import matplotlib as mpl  
import matplotlib.dates as mdates  
import datetime as dt  
import csv

data1 = pandas.read\_csv('student-por.csv')  
data2 = pandas.read\_csv('student-mat.csv')  
Data = [data1, data1]  
Data = pandas.concat(Data)  
Data = data1.append(data2, ignore\_index=True)

Data

arr = ['school', 'sex', 'address', 'famsize', 'Pstatus', 'Mjob', 'Fjob', 'reason', 'guardian', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'romantic']  
for column in arr:  
Set = set(Data[column])  
i = 0  
for item in Set:  
Data[column] = Data[column].replace(item, i)  
i = i + 1

Data['age'] = (Data['age'] - Data['age'].mean())/(math.sqrt(Data['age'].var()))  
Data['absences'] = (Data['absences'] - Data['absences'].mean())/(math.sqrt(Data['absences'].var()))  
Data['G1'] = (Data['G1'] - Data['G1'].mean())/(math.sqrt(Data['G1'].var()))  
Data['G2'] = (Data['G2'] - Data['G2'].mean())/(math.sqrt(Data['G2'].var()))  
Data['G3'] = (Data['G3'] - Data['G3'].mean())/(math.sqrt(Data['G3'].var()))

Data2 = Data.G3  
train = Data.drop('G3', axis = 1)

vif = pandas.DataFrame()  
vif["column\_name"] = train.columns  
vif["VIF"] = [variance\_inflation\_factor(train.values, i) for i in range(len(train.columns))]  
vif.loc[vif['VIF'] > 10.0]

while not (vif.empty):  
vif = pandas.DataFrame()  
vif["column\_name"] = train.columns  
vif["VIF"] = [variance\_inflation\_factor(train.values, i) for i in range(len(train.columns))]  
vif = vif.loc[vif['VIF'] >= 10.0]  
a = vif[vif["VIF"] == vif["VIF"].max()]["column\_name"]  
print(a.array[0])  
train = train.drop(a, axis=1)

train

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(train,Data2, test\_size = 0.3, random\_state = 42)  
train.shape

X\_train

model = LinearRegression().fit(X\_train,Y\_train)  
print (model.score(X\_test,Y\_test))

print('intercept:', model.intercept\_)  
print('slope:', model.coef\_)

print('G3 = ', model.intercept\_,'+')  
for i in range(len(train.columns)-1):  
print(model.coef\_[i],'\*',train.columns[i],'+')  
print(model.coef\_[len(train.columns)-1],'\*',train.columns[len(train.columns)-1])

Alc\_Data = Data.loc[:, Data.columns.isin(['Dalc','Walc'])]  
data = Data.loc[:, Data.columns.isin(['G3'])]

model\_Alc = LinearRegression().fit(Alc\_Data,Data2)  
print(model\_Alc.score(Alc\_Data, Data2))

print('intercept:', model\_Alc.intercept\_)  
print('slope:', model\_Alc.coef\_)

print('G3 = ', model\_Alc.intercept\_,'+')  
for i in range(len(Alc\_Data.columns)-1):  
print(model\_Alc.coef\_[i],'\*',Alc\_Data.columns[i],'+')  
print(model\_Alc.coef\_[len(Alc\_Data.columns)-1],'\*',Alc\_Data.columns[len(Alc\_Data.columns)-1])

middle = model.coef\_.mean()  
print("Среднее значение = ",middle)  
max = model.coef\_.max()  
print("Максимальное значение = ",max)  
min = model.coef\_.min()  
print("Минимальное значение = ",min)

vail = model.coef\_[0].mean()  
for i in range(len(train.columns)):  
if abs(model.coef\_[i].max()) > vail:  
print(train.columns[i],"=",model.coef\_[i].max())

vail = model.coef\_[0].max()  
for i in range(len(train.columns)):  
if abs(model.coef\_[i].max()) < vail:  
print(train.columns[i],"=",model.coef\_[i].max())

ols = statsmodels.api.OLS(data, train)  
res = ols.fit()  
print(res.summary())

dpi = 80  
fig = matplotlib.pyplot.figure(dpi = dpi, figsize = (512 / dpi, 384 / dpi) )  
matplotlib.pyplot.rcParams.update({'font.size': 9})  
  
matplotlib.pyplot.title('Коэффициент')  
  
ax = matplotlib.pyplot.axes()  
ax.xaxis.grid(True, zorder = 1)  
  
xs = range(len(train.columns))  
  
matplotlib.pyplot.barh([x + 0.3 for x in xs], [ d \* 0.9 for d in res.params],  
height = 0.2, color = 'red', alpha = 0.7,  
zorder = 2)  
  
matplotlib.pyplot.yticks(xs, train.columns, rotation = 10)  
  
matplotlib.pyplot.legend(loc='upper right')  
fig.savefig('barshoris.png')

dpi = 80  
fig = matplotlib.pyplot.figure(dpi = dpi, figsize = (512 / dpi, 384 / dpi) )  
matplotlib.pyplot.rcParams.update({'font.size': 9})  
  
matplotlib.pyplot.title('Стандартная ошибка')  
  
ax = matplotlib.pyplot.axes()  
ax.xaxis.grid(True, zorder = 1)  
  
xs = range(len(train.columns))  
  
matplotlib.pyplot.barh([x + 0.3 for x in xs], [ d \* 0.9 for d in res.bse],  
height = 0.2, color = 'red', alpha = 0.7,  
zorder = 2)  
  
matplotlib.pyplot.yticks(xs, train.columns, rotation = 10)  
  
matplotlib.pyplot.legend(loc='upper right')  
fig.savefig('barshoris.png')

dpi = 80  
fig = matplotlib.pyplot.figure(dpi = dpi, figsize = (512 / dpi, 384 / dpi) )  
matplotlib.pyplot.rcParams.update({'font.size': 9})  
  
matplotlib.pyplot.title(' t - Статистика')  
  
ax = matplotlib.pyplot.axes()  
ax.xaxis.grid(True, zorder = 1)  
  
xs = range(len(train.columns))  
  
matplotlib.pyplot.barh([x + 0.3 for x in xs], [ d \* 0.9 for d in res.tvalues],  
height = 0.2, color = 'red', alpha = 0.7,  
zorder = 2)  
  
matplotlib.pyplot.yticks(xs, train.columns, rotation = 10)  
  
matplotlib.pyplot.legend(loc='upper right')  
fig.savefig('barshoris.png')

dpi = 80  
fig = matplotlib.pyplot.figure(dpi = dpi, figsize = (512 / dpi, 384 / dpi) )  
matplotlib.pyplot.rcParams.update({'font.size': 9})  
  
matplotlib.pyplot.title(' p - значение')  
  
ax = matplotlib.pyplot.axes()  
ax.xaxis.grid(True, zorder = 1)  
  
xs = range(len(train.columns))  
  
matplotlib.pyplot.barh([x + 0.3 for x in xs], [ d \* 0.9 for d in res.pvalues],  
height = 0.2, color = 'red', alpha = 0.7,  
zorder = 2)  
  
matplotlib.pyplot.yticks(xs, train.columns, rotation = 10)  
  
matplotlib.pyplot.legend(loc='upper right')  
fig.savefig('barshoris.png')

dbscan = DBSCAN(eps=3.14, min\_samples=2.5)  
  
dbscan.fit(X\_train)  
pca = PCA(n\_components=2).fit(X\_train)  
pca\_2d = pca.transform(X\_train)  
print(set(dbscan.labels\_))

matplotlib.pyplot.figure(figsize=(10, 5))  
colors = matplotlib.cm .rainbow(numpy.linspace(0, 1, len(set(dbscan.labels\_))))  
for i in range(0, pca\_2d.shape[0]):  
if dbscan.labels\_[i] == -1:  
c0 = matplotlib.pyplot.scatter(pca\_2d[i, 0], pca\_2d[i, 1], c = 'grey', marker='\*')  
else:  
matplotlib.pyplot.scatter(pca\_2d[i, 0], pca\_2d[i, 1], c = [colors[dbscan.labels\_[i]]], marker='o')  
matplotlib.pyplot.title('результат работы DBSCAN')  
matplotlib.pyplot.show()