**Задача 1**

Условие

Набор данных: swiss.

Объясняемая переменная: *Agriculture*.

Регрессоры: *Education, Catholic.*

* Оцените среднее значение, дисперсию и СКО переменных, указанных во втором и третьем столбце.
* Постройте зависимости вида y = a + bx, где y – объясняемая переменная, x – регрессор (для каждого варианта по две зависимости).
* Оцените, насколько «хороша» модель по коэффициенту детерминации R2?
* Оцените, есть ли взаимосвязь между объясняемой переменной и объясняющей переменной (по значению p-статистики, «количеству звездочек» у регрессора в модели).

Решение

* Для оценки среднего значения будем использовать команду mean, для оценки дисперсии – команду var, для оценки СКО – команду sd.   
  В результате выполнения команд имеем:

Среднее значение *Agriculture* = 50.7

Среднее значение *Education*= 11

Среднее значение *Catholic*= 41.1

Дисперсия *Agriculture* = 515.8

Дисперсия *Education*= 92.5

Дисперсия *Catholic*= 1739.3

СКО *Agriculture* = 22.7

СКО *Education* = 9.6

СКО *Catholic*= 41.7

* Для построения линейной зависимости используем команду lm. В результате выполнения команды для первой и второй модели имеем:

y = 67.24 - 1.51x для *Agriculture~Education*

y = 41.67 + 0.218x для *Agriculture~Catholic*

* Чтобы посмотреть R2 воспользуемся командой summary. Для первой модели R2 всего 40%, это значит, что *Education* не может объяснить большее кол-во данных и *Agriculture* зависит от чего-то еще; для второй модели R2 составил всего 16%, это значит, что *Catholic* объясняет очень маленькое кол-во данных.
* P-статистика также показывается при выполнении команды summary. Для первой модели по кол-ву звездочек (их 3), видно, что вероятность того, что наше выражение коэффициента далеко от рельного очень низка, а значит зависимость между *Agriculture* и *Education* есть; для второй модели неплохое кол-во звездочек (их 2) говорит о том, что Catholic может оказаться хорошим параметром, однако значение R2 доказывает, что подобная зависимость скорее случайна.

Вердикт

По результатам исследования первой модели, параметра *Education* не достаточно, но модель мнообещающая и ее можно улучшить, однако вторая модель плохая, для описания *Agriculture,* параметр *Catholic* лучше не использовать.

Приложение 1

Код решения задачи:

data = swiss

# нужно оценить среднее значение, дисперсию и СКО

# Agriculture, Education и Catholic

# начнем со средних значений:

# суммируем значения Agriculture

sum(data$Agriculture)

# делим полученное значение на кол-во измерений, их 47

sum(data$Agriculture)/47

# видим, что среднее значение примерно 50.7

# проверим себя с помощью встроенной команды

mean(data$Agriculture)

# убедившись в правильности вычислений, используем mean дальше

mean(data$Education)

# примерно 11

mean(data$Catholic)

# примерно 41.1

# находим дисперсию и СКО, используя var и sd

var(data$Agriculture)

sd(data$Agriculture)

# примерно 515.8 и 22.7 соответственно

# дисперсия сильно больше среднего значения

var(data$Education)

sd(data$Education)

# примерно 92.5 и 9.6

# дисперсия снова больше среднего значения

var(data$Catholic)

sd(data$Catholic)

# примерно 1739.3 и 41.7

# дисперсия, по сравнению со средним значением, кажется огромной

# можно сделать вывод, что данные Catholic обладают очень

# большим разбросом, что значит, большинство наблюдений

# либо сильно ниже, либо сильно выше ожидаемого среднего

# теперь построим зависимости вида y = a + bx, где y –

# объясняемая переменная (Agriculture), x –

# регрессор (Education или Catholic)

# используем команду lm

model = lm(Agriculture~Education, data)

model

summary(model)

# по кол-ву звездочек, можем сказать, что вероятность того, что

# наше выражение коэффициента далеко от рельного очень низка, а

# значит зависимость между Agriculture и Education есть, причем

# обратная, судя по второму столбцу, чем выше Education, тем

# ниже Agriculture

# однако, R^2 всего 40%, это значит, что Education не может

# объяснить большее кол-во данных и Agriculture зависит от чего-то еще

# вердикт: Education не достаточно, но модель мнообещающая

# и её можно улучшить

model = lm(Agriculture~Catholic, data)

model

summary(model)

# неплохое кол-во звездочек намекает, что Catholic может оказаться

# хорошим параметром, однако, R^2 составил всего 16%, это

# значит, что Catholic объясняет настолько маленькое кол-во

# данных, что подобная зависимость скорее случайна

# вердикт: модель плохая, для описания Agriculture

# Catholic лучше не использовать

**Задача 2.1 и 2.2**

Условие

Набор данных: swiss.

Объясняемая переменная: *Agriculture*.

Регрессоры: *Fertility, Education, Catholic*.

* 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.
* Постройте линейную модель зависимой переменной от указанных в варианте регрессоров по методу наименьших квадратов (команда lm пакета lmtest в языке R). Оценить, насколько хороша модель, согласно: 1) R2, 2) p-значениям каждого коэффициента.
* Введите в модель логарифмы регрессоров (если возможно). Сравнить модели и выбрать наилучшую.
* Введите в модель всевозможные произведения пар регрессоров, в том числе квадраты регрессоров. Найдите одну или несколько наилучших моделей по доле объясненного разброса в данных R2.

Решение

* Чтобы проверить отсутствие линейной зависимости, построим для каждого регрессора модели, в качестве параметров которых будут другие регрессоры, и оценим в них R2:

*Fertility~Education+Catholic* – R2 = 0.58

*Education~Fertility+Catholic* – R2 = 0.47

*Catholic~Fertility+Education* – R2 = 0.26

Наблюдается сильная зависимость *Fertility* и *Education* от остальных параметров, избавимся от *Fertility* и проверим, насколько это хорошее решение:

*Education~Catholic* – R2 = 0.02

Линейной зависимости нет, значит регрессоры можно использовать вместе.

* Построим модель, используя оставшиеся *Education* и *Catholic*  и команду lm. Чтобы посмотреть R2 и p-статистику, воспользуемся командой summary. В результате ее выполнения видим:

*Agriculture~Education+Catholic*

Модель, в целом, неплохо описывает *Agriculture*, оба параметра обладают хорошим кол-вом звездочек (3 и 2 соответственно), существует зависимость, однако R2 в данной ситуации всего 50%, модели еще не хватает параметров, для почти полного описания *Agriculture*

* Чтобы улучшить показатели модели, попробуем ввести в модель логарифмы и всевозможные произведения пар регрессоров, для быстрого анализа зависимостей используем команду vif. Подробный код поиска наилучшей модели приведен в Приложении 2.

Изначальная модель:

*Agriculture~Education+Catholic+Education2+Catholic2+Education\*Catholic+log(Education)+log(Catholic)* – R2 = 0.6; множество регрессоров сильно зависимы друг от друга

Наилучшая модель:

*Agriculture~Catholic2+log(Education)* – R2 = 0.545; регрессоры не зависимы друг от друга

Вердикт

Финальный показатель R2 на 4,5% больше первоначального, добиться хорошего уровня R2, при описании *Agriculture* используя только *Education*, *Catholic* и их функции не удалось, скорее всего, этого можно будет достичь, используя дополнительные параметры.

Условие

Для зависимости, построенной при решении практического **задания 2.1**, оцените:

* Доверительные интервалы для всех коэффициентов в модели, p = 95%.
* Сделайте вывод о отвержении или невозможности отвергнуть статистическую гипотезу о том, что коэффициент равен 0
* Доверительный интервал для одного прогноза (p = 95%, набор значений регрессоров выбираете сами).

Решение

Таблица 1. Характеристики модели зависимости параметра: *Agriculture*, полученной в **задании 2.1**.



* Оценим доверительные интервалы для всех коэффициентов в модели, где реальное значение коэффициентов ожидается с 95% вероятностью (p=95%), для этого используем данные из второго и третьего столбцов, а также t -критерий Стьюдента:

Всего 47 наблюдений и 3 переменные, кол-во степеней свободы =

47-3 = 44

Используем команду t\_critical и получаем значение t-критерия Стьюдента = 2.015

Границы доверительного интервала имеют вид [b-tq; b+tq], где b - значение

коэффициента в таблице модели, t - критерий Стьюдента и q - следующее за

коэффициентом значение СКО, вычисления приведены в Приложении 2.

для параметра *Catholic2*  интервал [0.0005104787; 0.002745521]

для параметра *log(Education)* интервал [-22.96455; -10.75545]

для параметра *Intercept* интервал [65.46311; 95.59689]

Также, можно проверить, правильно ли построенны интервалы с помощью готовой функции confint

* 0 не попадает ни в один из доверительных интервалов, значение коэффициента для *Catholic2*  изначально было очень маленьким, поэтому, хоть границы интервала и близки к 0, колебания возможных значений также малы, можно смело отвергать статистическую гипотезу о том, что какой-либо коэффициент равен 0.
* Построим прогноз и вычислим доверительный интервал для *Agriculture*, для этого используем команду predictзначения для *Catholic* и *Education* возьмем 90 и 10 соответственно

Получены пределы значения *Agriculture* от 47.5 до 62.2

Вердикт

Опровержение статистической гипотезы о том, что какой-либо коэффициент параметров модели равен 0, а также попадание экспериментальных данных в предсказанный доверительный интервал, сведетельствуют о том, что модель состоятельна, хоть и не полна, на данный момент разработки.

Приложение 2

Код решения задачи:

data = swiss

# нужно проверить, что среди Fertility, Education, Catholic нет

# линейной зависимости, иначе, исключить наиболее зависимый

# параметр

# построим модели, где в роли объясняемой переменной

# будет один из параметров, а в роли регрессоров два других

# всего 3

model = lm(Fertility~Education+Catholic,data)

model

summary(model)

model = lm(Education~Fertility+Catholic,data)

model

summary(model)

model = lm(Catholic~Fertility+Education,data)

model

summary(model)

# в случае с Catholic, R^2 составил всего 26%, это хороший

# показатель отсутствия сильной зависимости от Fertility+Education,

# наиболее зависимы оказались Fertility и Education, в случае

# Fertility R^2 составил целых 58%, такую зависимость можно считать

# критичной и от Fertility можно избавиться

# чтобы проверить, насколько это хорошее решение, построим

# модель Education~Catholic

model = lm(Education~Catholic,data)

model

summary(model)

# теперь R^2 всего 2%, это значит зависимости между Education и

# Catholic нет никакой

# нужно построить и оценить модель зависимости Agriculture

# от оставшихся Education и Catholic

model = lm(Agriculture~Education+Catholic,data)

model

summary(model)

# модель, в целом, неплохо описывает Agriculture, оба параметра

# обладают хорошим кол-вом звездочек, существует зависимость,

# однако R^2 в данной ситуации всего 50%, модели ещё не хватает

# параметров, для почти полного описания Agriculture

# вердикт: попробуем добавить другие регрессоры и посмотрим,

# какой будет результат

# для быстрого анализа зависимостей используем vif

model = lm(Agriculture~Education+Catholic+I(Education^2)+

I(Catholic^2)+I(Education\*Catholic)+I(log(Education))+

I(log(Catholic)),data)

vif(model)

summary(model)

# даже после добавления всех функций, R^2 примерно 60%, результат

# не сказать, что хороший, к тому же, множество регрессоров

# сильно зависимы от других, можно смело избавиться от Catholic

model = lm(Agriculture~Education+I(Education^2)+I(Catholic^2)+

I(Education\*Catholic)+I(log(Education))+I(log(Catholic)),data)

vif(model)

summary(model)

# уберем Education

model = lm(Agriculture~I(Education^2)+I(Catholic^2)+

I(Education\*Catholic)+I(log(Education))+I(log(Catholic)),data)

vif(model)

summary(model)

# выкинем плохие Education^2 и Education\*Catholic

model = lm(Agriculture~I(Catholic^2)+I(log(Education))+

I(log(Catholic)),data)

vif(model)

summary(model)

# возможно, стоит убрать log(Catholic)

model = lm(Agriculture~I(Catholic^2)+I(log(Education)),data)

vif(model)

summary(model)

# финальный результат R^2 лишь на 4.5% процента лучше предыдущего,

# однако, с добавлением новых функций от Catholic и Education,

# например log(Catholic), сильно лучше он не станет

# вердикт: добиться хорошего уровня R^2, при описании Agriculture

# используя только Education, Catholic и их функции не удалось,

# скорее всего, этого можно будет достичь, используя

# дополнительные параметры

# теперь оценим доверительные интервалы для всех коэффициентов в

# модели, где реальное значение коэффициентов ожидается с 95%

# вероятностью (p=95%)

# для этого используем данные из второго и третьего столбцов, а также

# t-критерий Стьюдента

# всего 47 наблюдений и 3 переменные, кол-во степеней свободы =

# 47-3 = 44

t\_critical = qt(0.975,44)

# значение t-критерия Стьюдента примерно 2.015

# границы доверительного интервала имеют вид [b-tq;b+tq], где b - значение

# коэффициента в таблице модели, t - критерий Стьюдента и q - следующее за

# коэффициентом значение СКО

# для параметра Catholic^2 b = 1.628e-03, q = 5.545e-04, границы:

1.628e-03 - t\_critical \* 5.545e-04

1.628e-03 + t\_critical \* 5.545e-04

#~НОВОЕ

# для параметра log(Education) b = -1.686e+01, q = 3.029e+00, границы:

-1.686e+01 - t\_critical \* 3.029e+00

-1.686e+01 + t\_critical \* 3.029e+00

# для параметра Intercept b = 8.053e+01, q = 7.476e+00, границы:

8.053e+01 - t\_critical \* 7.476e+00

8.053e+01 + t\_critical \* 7.476e+00

# проверим с помощью готовой функции

confint(model, level = 0.95)

# как видим, 0 не попадает ни в один из доверительных интервалов, значение

# коэффициента для Catholic^2 изначально было очень маленьким, поэтому, хоть

# границы интервала и близки к 0, колебания возможных значений также малы,

# можно смело отвергать гипотезу о том, что какой-либо коэффициент = 0

# построим прогноз и вычислим доверительный интервал для Agriculture

# значения для Catholic и Education возьмем 90 и 10 соответственно

new.data = data.frame(Catholic = 90, Education = 10)

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

# в полученные пределы от 47.5 до 62.2 экспериментальные данные иногда

# попадают, но, учитывая изначально низкое качество модели, это в порядке вещей

**Задача 3**

Условие

Набор данных: обследование РМЭЗ НИУ ВШЭ (22 волна).

Объясняемая переменная: *Заработная плата (salary)*

Регрессоры: *Возраст (age), Пол (sex), Наличие высшего образования (higher\_educ), Вид населенного пункта (status), Продолжительность рабочей недели (dur), Семейное положение (Никогда не состоял в браке (wed1), Состоит в зарегистрированном браке (wed2), Состоит в гражданском браке (wed3), Разведен (wed4)).*

* Постройте линейную регрессию зарплаты на все параметры, которые Вы выделили из данных мониторинга. Не забудьте оценить коэффициент вздутия дисперсии VIF.
* Поэкспериментируйте с функциями вещественных параметров: используйте логарифм и степени (хотя бы от 0.1 до 2 с шагом 0.1).
* Выделите наилучшие модели из построенных: по значимости параметров, включенных в зависимости, и по объясненному с помощью построенных зависимостей разбросу adjusted R2 - R2 adj.
* Сделайте вывод о том, какие индивиды получают наибольшую зарплату.
* Оцените регрессии для подмножества индивидов, указанных в варианте (Женщины, не замужем; женщины, живущие в городе, разведенные).

Решение

* Для начала, такие поля как *Пол, Наличие высшего образования, Вид населенного пункта, Семейное положение* нужно представить, как одну или несколько переменных, принимающих значение 0 или 1; а *Заработную плату, Возраст, Продолжительность рабочей недели* нормализовать, таким образом упрощается работа с набором данных. Подробное преобразование в Приложении 3.
* После получения переменных, построим линейную регрессию salary на все параметры:

*salary~age+sex+higher\_educ+status+dur+wed1+wed2+wed3+wed4*

Таблица 2. Линейная регрессия salary.



Значение R2 всего 14%, а переменные *wed* имеют плохую p-статистику, после их удаления R2 совсем не изменился, значение VIF у остальных пременных низкое.

* Добавим в модель сначала логарифмы вещественных переменных, а затем и их степени, отберем лучший результат. Подробно в Приложении 3.

*salary~log(age)+sex+higher\_educ+status+log(dur)*

*salary~agepower+sex+higher\_educ+status+durpower,* где *power -* принимает

значения от 0.1 до 2 с шагом 0.1

Таблица 3. Лучшая нелинейная регрессия salary.



Эта модель выделилась лишь на десятые доли процента R2, в остальном у всех моделей идентичны adjusted R2 - R2 adj, и одинакого хорошие p-статистика и значение VIF. Однако, улучшение R2 по сравнению с первоначальной моделью составило 3%, что выглядит внушительно, когда их всего было 14%.

* Судя по полученным моделям, зарабатывают больше, в основном, молодые мужчины с высшим образованием, из города, работающие больше часов в неделю.
* Построим зависимости salary для двух подмножеств (см. Приложение 3):

Женщины, не замужем (sex==0, wed1==1)

Женщины, живущие в городе, разведенные(sex==0, status==1 wed4==1)

В первой модели R2 = 17%, VIF в порядке, но если судить по p-статистике, функции от вещественных переменных показывают здесь плохой результат, так же и во второй модели, хоть R2 и стал 20%.

Вердикт

С помощью исследования данных удалось обнаружить зависимость *Заработной платы* от определенных параметров, однако, при попытке построить регрессию, становится очевидно, что выбраных параметров недостаточно и *Заработная плата* зависит от еще многих факторов.

Приложение 3

Код решения задачи:

#install.packages("devtools")

#devtools::install\_github("bdemeshev/rlms")

library("lmtest")

library("rlms")

library("dplyr")

library("GGally")

library("car")

library("sandwich")

data <- rlms\_read("C:\\Users\\1235\\Desktop\\NIR\\r22i\_os26c.sav")

glimpse(data)

data2 = select(data, rj13.2, r\_age, rh5, r\_educ, status, rj6.2, r\_marst, rj1.1.2, rj23, rj24, rj32.1)

#исключаем строки с отсутствующими значениями NA

data2 = na.omit(data2)

#зарплата c элементами нормализации

data2$rj13.2

sal = as.numeric(data2$rj13.2)

sal1 = as.character(data2$rj13.2)

sal2 = lapply(sal1, as.integer)

sal = as.numeric(unlist(sal2))

mean(sal)

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

data2["salary"]

#возраст c элементами нормализации

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

age2 = lapply(age1, as.integer)

age3 = as.numeric(unlist(age2))

data2["age"]= (age3 - mean(age3)) / sqrt(var(age3))

data2["age"]

#пол

data2["sex"]=data2$rh5

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

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

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

#образование

data2["h\_educ"] = data2$r\_educ

data2["higher\_educ"] = data2$r\_educ

data2["higher\_educ"] = 0

data2$higher\_educ[which(data2$h\_educ==21)] <- 1

data2$higher\_educ[which(data2$h\_educ==22)] <- 1

data2$higher\_educ[which(data2$h\_educ==23)] <- 1

#населенный пункт

data2["status1"]=data2$status

data2["status"] = 0

data2$status[which(data2$status1==1)] <- 1

data2$status[which(data2$status1==2)] <- 1

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

#продолжительность рабочей недели

dur1 = as.character(data2$rj6.2)

dur2 = lapply(dur1, as.integer)

dur3 = as.numeric(unlist(dur2))

data2["dur"] = (dur3 - mean(dur3)) / sqrt(var(dur3))

#семейное положение:

#никогда не состояли в браке

data2["wed"]= data2$r\_marst

data2$wed1 = 0

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

#состоят в зарегистрированном браке

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

data2$wed2 = 0

data2$wed2[which(data2$wed==2)] <- 1

#состоят в гражданском браке

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

data2$wed3 = 0

data2$wed3[which(data2$wed==3)] <- 1

#разведены

data2["wed4"]=data2$r\_marst

data2$wed4 = 0

data2$wed4[which(data2$wed==4)] <- 1

# Проверим, что отсутствует линейная зависимость между семейными положениями

vif(lm(data2$r\_marst ~ data2$wed1 + data2$wed2 + data2$wed3 + data2$wed4))

data2 = na.omit(data2)

data3 = select(data2, salary, age, sex, higher\_educ, status, dur, wed1,wed2,wed3,wed4)

#построение зависимостей

model1 = lm(data = data3, salary~age+sex+higher\_educ+status+dur+wed1+wed2+wed3+wed4)

summary(model1)

vif(model1)

#переменные на семейное положение плохие - у них большой vif, в модели они не значимые

model2 = lm(data = data3, salary~age+sex+higher\_educ+status+dur)

summary(model2)

vif(model2)

#R^2 почти не изменился, vif и p-статистика хорошие

#можем ввести нелинейные регрессоры

model1 = lm(data = data3, salary~I(log(age)) + sex + higher\_educ + status + I(log(dur)))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^0.1) + sex + higher\_educ + status + I(dur^0.1))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^0.2) + sex + higher\_educ + status + I(dur^0.2))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^0.3) + sex + higher\_educ + status + I(dur^0.3))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^0.4) + sex + higher\_educ + status + I(dur^0.4))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^0.5) + sex + higher\_educ + status + I(dur^0.5))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^0.6) + sex + higher\_educ + status + I(dur^0.6))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^0.7) + sex + higher\_educ + status + I(dur^0.7))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^0.8) + sex + higher\_educ + status + I(dur^0.8))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^0.9) + sex + higher\_educ + status + I(dur^0.9))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^1.1) + sex + higher\_educ + status + I(dur^1.1))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^1.2) + sex + higher\_educ + status + I(dur^1.2))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^1.3) + sex + higher\_educ + status + I(dur^1.3))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^1.4) + sex + higher\_educ + status + I(dur^1.4))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^1.5) + sex + higher\_educ + status + I(dur^1.5))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^1.6) + sex + higher\_educ + status + I(dur^1.6))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^1.7) + sex + higher\_educ + status + I(dur^1.7))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^1.8) + sex + higher\_educ + status + I(dur^1.8))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^1.9) + sex + higher\_educ + status + I(dur^1.9))

summary(model1)

vif(model1)

model1 = lm(data = data3, salary~I(age^2) + sex + higher\_educ + status + I(dur^2))

summary(model1)

vif(model1)

#везде одинаковый adjusted R^2 - R^2 adj и почти одинаково хорошие p-статистика

#и vif, но немного различаются R^2

#лучшей оказалась модель:

model1 = lm(data = data3, salary~I(age^0.6) + sex + higher\_educ + status + I(dur^0.6))

summary(model1)

vif(model1)

#Построение моделей для подмножеств:

#Женщины, не замужем

dataV = subset(data3,sex==0)

dataV = subset(dataV,wed1==0)

model1 = lm(data = dataV, salary~I(age^0.6) + higher\_educ + status + I(dur^0.6))

summary(model1)

vif(model1)

#R^2 = 17%, VIF в порядке, но плохая p-статистика у

# I(age^0.6) и I(dur^0.6)

#Женщины, живущие в городе, разведенные

dataS = subset(data3,sex==0)

dataS = subset(dataS,status==1)

dataS = subset(dataS,wed4==1)

model2 = lm(data = dataS, salary~I(age^0.6) + higher\_educ + I(dur^0.6))

summary(model2)

vif(model2)

#R^2 = 17%, VIF в порядке, но p-статистика ещё хуже у

# I(age^0.6) и I(dur^0.6) и даже свободного члена

**Задача 4**

Условие

Набор данных: HR Analytics: Job Change of Data Scientists.

Тип классификатора: *LogisticRegression (логистическая регрессия)*.

Классификация по столбцу: *Education level (Masters – класс 0, остальные уровни – класс 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.
* 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?

Решение

* Первым делом мы выбираем столбцы, которые остануться в очищенном наборе данных, конечно, столбец *education\_level* будет отдельно:

Мой выбор: *gender, relevent\_experience, enrolled\_university, major\_discipline, experience, company\_type, last\_new\_job, training\_hours*

Теперь нужно отформатировать данные в столбцах, для обработки, подробно в Приложении 4:

Данные в *gender, relevent\_experience, enrolled\_university, major\_discipline, company\_type* преобразуем из строчных в целочисленные

Данные в *experience* и *last\_new\_job* содержат ненужные символы перед цифрами, их убираем

Таблица 4. Готовый к задаче классификации набор данных.



Построим классификатор методом логистической регрессии; для параметров, заданных вручную, см. Приложение 4. Полученные результаты на тестовой выборке:

f1: 0.8522626265042677

precision: 0.7425591659634214

recall: 1.0

Получилось достаточно хорошо провести классификацию, хотя точность не слишком высокая

* Попробуем улучшить результат, использовав другой метод, построим пару классификаторов типа Случайный Лес, подробно в Приложении 4.

Результаты первого:

f1:0.850854930271398

precision:0.7422491077820643

recall:0.9979166666666666

Результаты второго:

f1:0.851496983309372

precision:0.7422596468948126

recall:0.9984375

У обоих классификаторов типа Случайный Лес тоже хороший результат, из чего следует

Вердикт

Так как при построении классификаторов другого типа, а также, с некоторыми измененными параметрами, не сильно наблюдалось падение или рост качества классификации, можно сказать, что выборка близка к совершенству, однако лучшим здесь оказался, как не странно, метод логистической регрессии, хоть и не на много, в данной ситуации, я бы назвал их равносильными.

Приложение 4

Код решения задачи:

!pip install pandas

!pip install sklearn

import pandas

import numpy as np

import warnings

warnings.filterwarnings('ignore')

data = pandas.read\_csv('aug\_train.csv', index\_col='enrollee\_id')

data\_ed=data.loc[:,data.columns.isin([ 'city\_development\_index', 'gender', 'relevent\_experience', 'enrolled\_university', 'education\_level', 'major\_discipline', 'experience', 'company\_type', 'last\_new\_job', 'training\_hours', 'target'])]

data\_ed['education\_level'] = np.where(data\_ed['education\_level'] == 'Masters', 0, 1)

data\_ed = data\_ed.dropna()

#здесь я беру только нужные мне столбцы

Education = data\_ed.loc[:, data\_ed.columns.isin(['education\_level'])]

X = data\_ed.loc[:, data\_ed.columns.isin( ['gender', 'relevent\_experience', 'enrolled\_university', 'major\_discipline', 'experience', 'company\_type', 'last\_new\_job', 'training\_hours'])]

#здесь я их фильтрую

#relevent\_experience имеет два строчных значения

my\_set = set(X.relevent\_experience)

i = 0

for item in my\_set:

X['relevent\_experience']=X['relevent\_experience'].replace(item,i)

i = i + 1

#присваиваем им целочисленные значения

#enrolled\_university имеет два строчных значения

i = 0

my\_set2 = set(X.enrolled\_university)

for item in my\_set2:

X['enrolled\_university']=X['enrolled\_university'].replace(item,i)

i = i + 1

#присваиваем им целочисленные значения

#gender имеет два строчных значения

i = 0

my\_set3 = set(X.gender)

for item in my\_set3:

X['gender'] = X['gender'].replace(item, i)

i = i + 1

#присваиваем им целочисленные значения

#gender имеет пять строчных значений

i = 0

my\_set4 = set(X.major\_discipline)

for item in my\_set4:

X['major\_discipline'] = X['major\_discipline'].replace(item, i)

i = i + 1

#присваиваем им целочисленные значения

#company\_type имеет четыре строчных значения

i = 0

my\_set5 = set(X.company\_type)

for item in my\_set5:

X['company\_type'] = X['company\_type'].replace(item, i)

i = i + 1

#присваиваем им целочисленные значения

#experience и last\_new\_job имеют множество значений с '<' или '>' и числом

my\_set6 = set(X.experience)

for item in my\_set6:

if ((str(item)[0] == '<')or(str(item)[0] == '>')):

X['experience'] = X['experience'].replace(item, item[1:])

my\_set7 = set(X.last\_new\_job)

for item in my\_set7:

if ((str(item)[0] == '<')or(str(item)[0] == '>')):

X['last\_new\_job'] = X['last\_new\_job'].replace(item, item[1:])

if (str(item) == 'never'):

X['last\_new\_job'] = X['last\_new\_job'].replace(item, 0)

#убираем лишние символы

#результат:

X

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

#разделение на обучающую и тестовую выборку

x\_train, x\_validation, y\_train, y\_validation = train\_test\_split(X, Education, test\_size=.28, random\_state=5)

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import GridSearchCV

from sklearn.model\_selection import StratifiedKFold

#произведение классификации способом логистической регрессии

grid={"C":np.logspace(-3,3,7),"penalty":["l1","l2"]}

logreg=LogisticRegression()

logreg\_cv=GridSearchCV(logreg,grid,cv=5)

logreg\_cv.fit(x\_train,y\_train)

#вывод

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

print("f1:"+str(np.average(cross\_val\_score(logreg\_cv, x\_validation, y\_validation, scoring='f1'))))

print("precision:"+str(np.average(cross\_val\_score(logreg\_cv, x\_validation, y\_validation, scoring='precision'))))

print("recall:"+str(np.average(cross\_val\_score(logreg\_cv, x\_validation, y\_validation, scoring='recall'))))

#произведение первой классификации способом случайного леса

from sklearn.ensemble import RandomForestClassifier

param\_grid = { 'n\_estimators': [50, 100, 150],'max\_features': ['auto'],'max\_depth' : list(range(1, 10)), 'criterion' :['gini']}

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

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

#вывод

print("f1:"+str(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='f1'))))

print("precision:"+str(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='precision'))))

print("recall:"+str(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='recall'))))

#произведение второй классификации способом случайного леса

param\_grid = { 'n\_estimators': [150, 200, 250],'max\_features': ['auto'],'max\_depth' : list(range(2, 9)), 'criterion' :['gini']}

RFC=GridSearchCV(estimator=RandomForestClassifier(), param\_grid=param\_grid, cv= 4, refit = True)

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

#вывод

print("f1:"+str(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='f1'))))

print("precision:"+str(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='precision'))))

print("recall:"+str(np.average(cross\_val\_score(RFC.best\_estimator\_, x\_validation, y\_validation, scoring='recall'))))