



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Институт кибернетики

Кафедра высшей математики

ОТЧЁТ ПО Научно-Исследовательской Работе
(указать вид практики)

Тема практики: Популярные деревья в районах Нью-Йорка (kaggle.com)
приказ университета о направлении на практику
490 – С от 09.02.2021 г.

Отчет представлен к
рассмотрению:
Студентка группы КМБО-
03-20

Евдокимова Е.А.
(расшифровка подписи)
«11» июня 2021 г.

Отчет утвержден.
Допущена к защите:

Руководитель практики от
кафедры

Петрусович Д.А.
(расшифровка подписи)
«3» июня 2021 г.

Москва 2021



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

ЗАДАНИЕ НА Научно-Исследовательскую Работу

Студентке 1 курса учебной группы КМБО-03-20 института кибернетики
Евдокимовой Екатерине Александровне

(фамилия, имя и отчество)

Место и время практики: Институт кибернетики, кафедра высшей математики

Время практики: с «09» февраля 2021 по «31» мая 2021

Должность на практике: практикант

1. ЦЕЛЕВАЯ УСТАНОВКА: изучение основ анализа данных и машинного обучения

2. СОДЕРЖАНИЕ ПРАКТИКИ:

2.1 Изучить: литературу и практические примеры по темам: 1) построение линейной регрессии, 2) использование метода главных компонент, 3) поиск и устранение линейной зависимости в данных, 4) основы нормализации данных, 5) методы классификации и кластеризации («решающее дерево», «случайный лес», «k ближайших соседей»).

2.2 Практически выполнить: 1) снижение размерности исходных задач при помощи метода главных компонент при возможности; построение линейной регрессии для некоторого параметра, исключение регрессоров, не коррелирующих с объясняемой переменной; решение задачи классификации или кластеризации на основе открытого набора данных с ресурса kaggle.com

2.3 Ознакомиться: с применением метода главных компонент; методов классификации («решающего дерева», «случайного леса»); методов кластеризации («k ближайших соседей»); построением модели линейной регрессии.

3. ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ: популярные деревья в районах Нью-Йорка (kaggle.com).

4. ОГРАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ: выделить районы, в которых состояние деревьев аномально хорошее или плохое; выделить наилучший район по состоянию деревьев; выделить виды деревьев (или более общие элементы классификации), обладающие наилучшими показателями «здоровья».

Заведующий кафедрой
высшей математики

Ю.И.Худак

«09» февраля 2021 г.

СОГЛАСОВАНО

Руководитель практики от кафедры:

«09» февраля 2021 г.

(подпись)

(Петрусеви́ч Д.А.)
(фамилия и инициалы)

Задание получила:

«09» февраля 2020 г.

(подпись)

(Евдокимова Е.А.)
(фамилия и инициалы)



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА - Российский технологический университет»
РТУ МИРЭА

РАБОЧИЙ ГРАФИК ПРОВЕДЕНИЯ Научно-Исследовательской Работы

студента Евдокимовой Е.А. 1 курса группы КМБО-03-20 очной формы обучения,
обучающегося по направлению подготовки 01.03.02 «Прикладная математика и
информатика»,
профиль «Математическое моделирование и вычислительная математика»

Неделя	Сроки выполнения	Этап	Отметка о выполнении
1	09.02.2021	Выбор темы НИР. Пройти инструктаж по технике безопасности	✓
1	09.02.2021	Вводная установочная лекция	✓
1	13.02.2021	Построение и оценка парной регрессии с помощью языка R	✓
2	20.02.2021	Построение и оценка множественной регрессии с помощью языка R	✓
3	27.02.2021	Построение доверительных интервалов. Обработка факторных переменных. Мультиколлинеарность	✓
4	06.03.2021	Гетероскедастичность	✓
5	13.03.2021	Классификация	✓
7	27.03.2021	Кластеризация. Предобработка данных	✓
9	10.04.2021	Метод главных компонент	✓
17	05.06.2021	Представление отчётных материалов по НИР и их защита. Передача обобщённых	✓

		материалов на кафедру для архивного хранения	
		Зачётная аттестация	

Содержание практики и планируемые результаты согласованы с руководителем практики от профильной организации.

Согласовано:

Заведующий кафедрой



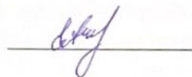
/ ФИО / Худак Ю.И.

Руководитель практики
от кафедры











/ ФИО / Петрусович Д.А.

Обучающаяся



/ ФИО / Евдокимова Е.А.

ИНСТРУКТАЖ ПРОВЕДЕН:

Вид мероприятия	ФИО ответственного, подпись, дата	ФИО студентки, подпись, дата
Охрана труда	Петрусеви́ч Д.А.  «09» февраля 2021 г.	Евдокимова Е.А.  «09» февраля 2021 г.
Техника безопасности	Петрусеви́ч Д.А.  «09» февраля 2021 г.	Евдокимова Е.А.  «09» февраля 2021 г.
Пожарная безопасность	Петрусеви́ч Д.А.  «09» февраля 2021 г.	Евдокимова Е.А.  «09» февраля 2021 г.
Правила внутреннего распорядка	Петрусеви́ч Д.А.  «09» февраля 2021 г.	Евдокимова Е.А.  «09» февраля 2021 г.

Оглавление

Задача 1	3
Задача 2	6
Задача 3	12
Задача 4	18
Задача 5	21
Заключение	23
Список литературы	24
Приложение	25

Задача 1

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: Swiss.

Объясняемая переменная: Agriculture.

Регрессоры: Examination, Infant.Mortality.

1. Оцените среднее значение, дисперсию и СКО переменных, указанных во втором и третьем столбце.

СКО - среднеквадратичное (стандартное) отклонение, показывает нам насколько сильный разброс с настоящими значениями.

По результатам работы функции (`sd(data$Agriculture)`) можно увидеть стандартное отклонение равное - 22.7112.

При помощи функции (`var(data$Agriculture)`) можно увидеть, как сильно новые значения будут отличаться от старых. В данном случае изменения будут равны 515.7994, видим, что разброс большой.

После работы функции (`sd(data$Examination)`) можно увидеть стандартное отклонение равное - 7.9779.

На основании результатов функции (`var(data$Examination)`) понимаем насколько новые значения отличаются от старых. В данном случае разброс будет не очень большим, так как он равен 63.64662.

Благодаря функции (`sd(data$Infant.Mortality)`) можно увидеть стандартное отклонение, которое равно 2.91.

Опираясь на результат функции (`var(data$Infant.Mortality)`), понимаем насколько сильно новые значения будут отличаться от старых. В данном случае оно равно 8.48, значит разброс маленький.

2. Постройте зависимости вида $y = a + bx$, где y – объясняемая переменная, x – регрессор.

Значение для зависимости *Agriculture~Examination* можем взять из *Рисунок 1*, посмотрев на столбец *Esyimate Std.* у параметра *Examination*. А значение *b* возьмем из *Рисунок 1* в том же столбце у параметра *(Intercept)*. Получаем зависимость вида: $Agriculture = -1.9544 * Examination + 82.8869$.

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	82.8869	5.6407	14.694	< 2e-16 ***
Examination	-1.9544	0.3086	-6.334	9.95e-08 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.7 on 45 degrees of freedom

Multiple R-squared: 0.4713, Adjusted R-squared: 0.4596

F-statistic: 40.12 on 1 and 45 DF, p-value: 9.952e-08

Рисунок 1 «Характеристики модели зависимости параметра Agriculture от параметра Catholic в наборе данных Swiss»

Теперь построим зависимость *Agriculture~Infant.Mortality*, используя Рисунок 2 и столбец Estimate Std. получим зависимость $Agriculture = -0.4745 * Infant.Mortality + 60.1230$

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	60.1230	23.3776	2.572	0.0135 *
Infant.Mortality	-0.4745	1.1602	-0.409	0.6845

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 22.92 on 45 degrees of freedom

Multiple R-squared: 0.003704, Adjusted R-squared: -0.01844

F-statistic: 0.1673 on 1 and 45 DF, p-value: 0.6845

Рисунок 2 «Характеристики модели зависимости параметра Agriculture от параметра Infant.Mortality в наборе данных Swiss»

3. Оцените, насколько «хороша» модель по коэффициенту детерминации R^2 ?

Значение R^2 в зависимости *Agriculture~Examination* равно 47,13%, оно относительно хорошо. Коэффициент высок, но для более точной информации нужно добавлять ещё параметры, от которых зависит число людей, работающих в с/х сфере.

Значение R^2 в зависимости *Agriculture~Infant.Mortality* равно 0,37%, можем сделать вывод, что модель плоха: коэффициент очень низок, возможно, следует построить новую модель, так как в этой модели почти отсутствуют какие-либо взаимосвязи. Модель относительно хорошая и отрицательная.

4. Оцените, есть ли взаимосвязь между объясняемой переменной и объясняющей переменной.

Оценить наличие зависимости между переменными *Agriculture* и *Examination* поможет последний столбец на Рисунок 1.

Взаимосвязь между объясняемой переменной (*Agriculture*) и объясняющей переменной (*Examination*) достаточно высокая, принадлежит промежутку (от 0 до 0.001), равная "****".

Оценить наличие зависимости между переменными *Agriculture* и *Infant.Mortality* можно по последнему столбцу на *Рисунок 2*. Взаимосвязь между объясняемой переменной (*Agriculture*) и объясняющей переменной (*Infant.Mortality*) низкая, принадлежит промежутку (от 0.1 до 1), равная « ». Следовательно, взаимосвязи почти нет.

Вывод

Удалось построить две модели. *Agriculture ~ Infant.Mortality* получилась плохая и отрицательная, так как у нее очень низкий R^2 и проглядывается плохая зависимость между объясняемой переменной и регрессором.

А вот о модели *Agriculture~Examination* нельзя сказать однозначно, так как у нее высокий R^2 , при этом хорошая отрицательная зависимость между объясняемой переменной и регрессором. Поэтому делаем вывод, что зависимость есть, но она сложнее, чем то, что проверяем.

Код решения задачи и сведения о проверенных моделях приведены в (*Приложение 1*).

Задача 2

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: swiss.

Объясняемая переменная: Examination.

Регрессоры: Fertility, Catholic, Infant.Mortality.

1. Проверить, что в наборе данных нет линейной зависимости (построить зависимости между переменными, указанными в варианте, и проверить, что R^2 в каждой из них не высокий). В случае, если R^2 большой, один из таких столбцов можно исключить из рассмотрения.

Проверим линейную регрессию *Fertility~Catholic*. R^2 в этой модели около 21,5 % делаем вывод, что параметр *Fertility* не зависит от других регрессоров линейно и может быть использован при построении математических моделей.

Зависимость *Fertility~Infant.Mortality*. $R^2 = 17,35\%$ делаем вывод, что параметр *Fertility* не зависит от других регрессоров линейно и может быть использован при построении математических моделей.

В регрессии *Catholic~Infant.Mortality* значение R^2 около 3,08%, следовательно, параметр *Catholic* не зависит от других регрессоров линейно и может быть использован при построении математических моделей.

Таким образом все регрессоры, указанные в задании, использовать при построении моделей линейной регрессии можно.

2. Построить 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) R^2 , 2) p-значениям каждого коэффициента.

Характеристики модели зависимости *Examination* от регрессоров *Fertility*, *Catholic* и *Infant.Mortality*, приведены на *Рисунок 3 "Характеристики модели зависимости параметра Examination от параметров Fertility, Catholic и Infant.Mortality в наборе данных Swiss"*. Модель получилась хорошая, так как $R^2 = 53,91\%$, p-значения у всех параметров хорошие, кроме *Infant.Mortality*, исключив его из модели показатели коэффициента детерминации изменился не очень сильно, p-статистика - стала лучше, следовательно, это изменение было правильным и обоснованным. Показания новой модели можно увидеть на *Рисунок 4*.

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	34.50312	6.39996	5.391	2.79e-06	***
Fertility	-0.35856	0.08083	-4.436	6.27e-05	***
Catholic	-0.06581	0.02236	-2.944	0.00521	**
Infant.Mortality	0.49363	0.31199	1.582	0.12093	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.602 on 43 degrees of freedom

Multiple R-squared: 0.5391, Adjusted R-squared: 0.507

F-statistic: 16.77 on 3 and 43 DF, p-value: 2.329e-07

Рисунок 3 "Характеристики модели зависимости параметра Examination от параметров Fertility, Catholic и Infant.Mortality в наборе данных Swiss"

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	40.93206	5.02837	8.140	2.54e-10	***
Fertility	-0.30941	0.07589	-4.077	0.000188	***
Catholic	-0.06659	0.02273	-2.929	0.005364	**

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.697 on 44 degrees of freedom

Multiple R-squared: 0.5123, Adjusted R-squared: 0.4901

F-statistic: 23.11 on 2 and 44 DF, p-value: 1.379e-07

Рисунок 4 "Характеристики модели зависимости параметра Examination от параметров Fertility, Catholic в наборе данных Swiss"

3. Ввести в модель логарифмы регрессоров (если возможно). Сравнить модели и выбрать наилучшую.

Введя в модель логарифмы получается модель, с коэффициентом детерминации 65,96% выше, чем у исходной. р-значения представлены на *Рисунок 5*.

Заметим, что плохих р-значений стало больше. И при вызове vif сразу у нескольких параметров значение больше 10. Попробуем улучшить модель, убрав $I(\log_{10}(\text{Fertility}))$, так как он имеет наибольший показатель vif. Значения линейно раскладываются у $I(\log_{10}(\text{Infant.Mortality}))$ и у Catholic, убрав их из модели получается модель с $R^2 = 49,44\%$, что является относительно хорошим показателем. р-статистика регрессоров неплохая (*Рисунок 6*), что говорит о том, что модель относительно хороша, но требует корректировок.

Модель, приведённая в пункте 2 относительно лучше, модели приведённой в пункте 3, так как оцениваемые характеристики (R^2 и р-статистики) немного лучше в первой модели.

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	297.76495	109.08476	2.730	0.009379	**
Fertility	0.23097	0.47008	0.491	0.625871	
Catholic	-0.24773	0.06537	-3.790	0.000498	***
Infant.Mortality	5.02458	1.83709	2.735	0.009251	**
I(log10(Fertility))	-72.61831	67.12286	-1.082	0.285789	
I(log10(Catholic))	11.35070	4.33395	2.619	0.012397	*
I(log10(Infant.Mortality))	-207.27284	79.25832	-2.615	0.012517	*

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.992 on 40 degrees of freedom

Multiple R-squared: 0.6596, Adjusted R-squared: 0.6085

F-statistic: 12.92 on 6 and 40 DF, p-value: 4.611e-08

Рисунок 5 "Характеристики модели зависимости параметра Examination от Fertility, Catholic, Infant.Mortality, I(log10(Fertility)), I(log10(Catholic)) и I(log10(Infant.Mortality)) в наборе данных Swiss"

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	38.16533	6.54131	5.835	6.36e-07	***
Fertility	-0.41519	0.07966	-5.212	5.04e-06	***
Infant.Mortality	0.56728	0.32775	1.731	0.0907	.
log10(Catholic)	-3.04720	1.50515	-2.025	0.0492	*

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.867 on 43 degrees of freedom

Multiple R-squared: 0.4944, Adjusted R-squared: 0.4592

F-statistic: 14.02 on 3 and 43 DF, p-value: 1.637e-06

Рисунок 6 "Характеристики модели зависимости параметра Examination от Fertility, Infant.Mortality и I(log10(Catholic)) в наборе данных Swiss"

4. Введите в модель всевозможные произведения пар регрессоров, в том числе квадраты регрессоров. Найдите одну или несколько наилучших моделей по доле объяснённого разброса в данных R^2 .

Добавив в первую модель квадраты и произведения регрессоров получим модель, у которой $R^2 = 65,23\%$. Однако у параметров плохое p-значение (Рисунок 7). Также обратим внимание на показания vif все данные превышают 10. Постепенно убираем из модели параметры с максимальным vif и плохими p-значениями (I(Infant.Mortality * Fertility), Catholic, I(Fertility * Catholic), I(Fertility^2), I(Infant.Mortality^2), I(Catholic^2)).

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	98.826734	46.598811	2.121	0.0407 *
Fertility	-1.208910	0.903765	-1.338	0.1892
Catholic	-0.174318	0.305897	-0.570	0.5722
Infant.Mortality	-2.485413	3.103599	-0.801	0.4284
I(Fertility^2)	-0.002689	0.005823	-0.462	0.6469
I(Catholic^2)	-0.002780	0.001841	-1.510	0.1395
I(Infant.Mortality^2)	-0.009747	0.081014	-0.120	0.9049
I(Infant.Mortality * Fertility)	0.049391	0.049397	1.000	0.3239
I(Infant.Mortality * Catholic)	-0.004194	0.012437	-0.337	0.7379
I(Fertility * Catholic)	0.006386	0.003169	2.015	0.0512 .

signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.245 on 37 degrees of freedom

Multiple R-squared: 0.6523, Adjusted R-squared: 0.5677

F-statistic: 7.712 on 9 and 37 DF, p-value: 2.802e-06

Рисунок 7 "Характеристики модели зависимости параметра Examination от Fertility, Catholic, Infant.Mortality, I(Fertility^2), I(Catholic^2), I(Infant.Mortality^2), I(Infant.Mortality * Fertility), I(Infant.Mortality * Catholic), I(Fertility * Catholic) в наборе данных Swiss"

Таким образом, получена хорошая модель. $R^2 = 51,64\%$, что указывает на достаточно хорошую линейную зависимость, р-статистика показывает неплохие результаты у регрессоров (Рисунок 8).

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	32.569169	6.844374	4.759	2.22e-05 ***
Fertility	-0.374414	0.082405	-4.544	4.44e-05 ***
Infant.Mortality	0.629239	0.322830	1.949	0.0578 .
I(Infant.Mortality * Catholic)	-0.002812	0.001126	-2.498	0.0164 *

signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.738 on 43 degrees of freedom

Multiple R-squared: 0.5164, Adjusted R-squared: 0.4827

F-statistic: 15.31 on 3 and 43 DF, p-value: 6.419e-07

Рисунок 8 "Характеристики модели зависимости параметра Examination от Fertility, Infant.Mortality и I(Infant.Mortality * Catholic) в наборе данных Swiss"

Так как во второй модели у регрессоров I(Catholic^2) и I(Infant.Mortality * Catholic) почти одинаковые показатели vif. Попробуем создать еще одну хорошую модель.

Действительно, убрав во второй модели регрессор I(Infant.Mortality * Catholic), получаем хорошую модель с vif показателями меньше 3. По R^2 модель не уступает предыдущим его показатель равен 54,97%. р-статистика также показывает хорошие результаты (Рисунок 9). Следовательно, эта модель хорошая.

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  33.1863218   6.4116126   5.176 5.68e-06 ***
Fertility    -0.3289215   0.0834995  -3.939 0.000295 ***
Infant.Mortality 0.4381764   0.3092576   1.417 0.163727
I(Catholic^2) -0.0006963   0.0002216  -3.143 0.003029 **
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.537 on 43 degrees of freedom
Multiple R-squared:  0.5497,    Adjusted R-squared:  0.5183
F-statistic: 17.5 on 3 and 43 DF,  p-value: 1.428e-07

```

Рисунок 9 "Характеристики модели зависимости параметра Examination от Fertility, Infant.Mortality и I(Catholic^2) в наборе данных Swiss"

В этой модели показатели R^2 и p-статистика достаточно хороши, следовательно, эта модель является наилучшей.

5. Доверительные интервалы для всех коэффициентов в модели, $p = 95\%$.

Найдем доверительный интервал для модели зависимости параметра Examination от Fertility и Infant.Mortality.

47 наблюдений, оценивалось 4 коэффициента: $47 - 4 = 43$ степени свободы.

Доверительный интервал имеет общую формулу $[y^{\wedge} - t\sigma, y^{\wedge} + t\sigma]$, где y^{\wedge} - это значение параметра из столбца Estimate, а σ – значение из столбца Std. Error.

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  38.67737   6.76266   5.719 8.71e-07 ***
Fertility    -0.46240   0.07881  -5.867 5.29e-07 ***
Infant.Mortality 0.51376   0.33799   1.520  0.136
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.07 on 44 degrees of freedom
Multiple R-squared:  0.4462,    Adjusted R-squared:  0.4211
F-statistic: 17.73 on 2 and 44 DF,  p-value: 2.254e-06

```

Рисунок 10 "Характеристики модели зависимости параметра Examination от Fertility, Infant.Mortality в наборе данных Swiss"

Доверительный интервал для Fertility $[-0.46240 - t * 0.07881; -0.46240 + t * 0.07881]$.
Параметр t рассчитаем с помощью функции $t_critical = qt(0.975, df = 149)$.

Получили доверительный интервал равный $[-0.6181266; -0.3066674]$.

Доверительный интервал для Infant.Mortality $[0.51376 - t * 0.33799; 0.51376 + t * 0.33799]$

Значение параметра t не меняется, поэтому получается интервал $[-0.1541124; 1.181633]$.

Проверить себя можно с помощью функции `confint(modele, level = 0.95)` (Рисунок 11).

	2.5 %	97.5 %
(Intercept)	25.0481246	52.3066243
Fertility	-0.6212288	-0.3035652
Infant.Mortality	-0.1674224	1.1949430

Рисунок 11 "Проверка правильности доверительного"

6. Сделайте вывод о отвержении или невозможности отвергнуть статистическую гипотезу о том, что коэффициент равен 0.

В доверительном интервале с Infant.Mortality встречается ноль, следовательно, коэффициент может равняться нулю.

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

Построим доверительного интервала для прогноза на модели

```
modele_pr = lm(Examination~Fertility + Catholic + Infant.Mortality, data)
```

Вычислять ошибки здесь сложнее поэтому просто используем встроенные функции.

```
new.data = data.frame(Fertility = 20,Catholic = 10, Infant.Mortality = 10)
```

```
predict(modele_pr, new.data, interval = "confidence").
```

 Таким образом получили:

fit - значение прогноза = 31.61013, нижняя и верхняя граница: lwr = 23.75496 upr = 39.4653. Тогда доверительный интервал равен [23.75496, 39.4653].

Вывод

Были проверены регрессоры, указанные в задании, на линейную зависимость. Все регрессоры можно использовать при построении линейной регрессии можно, так как линейная зависимость отсутствует. Выведена наилучшая модель при помощи введения в неё квадратов и произведений, логарифмов. Модель согласно $R^2 = 54,97\%$ получилась хорошая, р-значения у всех параметров хорошие. Зависимость между объясняемой переменной и регрессорами: Fertility - отрицательная, Infant.Mortality – незначимый, I(Catholic^2) – отрицательная. Были найдены доверительные интервалы, на основе которых был сделан вывод о том, что коэффициент может равняться нулю.

Код решения задачи и сведения о проверенных моделях приведен в (Приложение 2).

Задача 3

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Номер волны выборки РМЭЗ НИУ ВШЭ: 21

Объясняемая переменная: Salary.

Регрессоры: age, sex, higher_educ, status2, dur, wed, wed2, wed3.

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

Из параметра, отвечающего семейному положению, получаем:

- 1) переменная wed1 имеет значение 1 в случае, если респондент женат, 0 – в противном случае;
- 2) wed2=1, если респондент разведён или вдовец;
- 3) wed3 = 1, если респондент никогда не состоял в браке;

Из параметра пол делаем переменную sex, имеющую значение 1 для мужчин и равную 0 для женщин.

Из параметра, отвечающего типу населённого пункта, создаем одну дамми-переменную status со значением 1 для города или областного центра, 0 – в противоположном случае.

Построив зависимость по всем параметрам видим, что wed1, wed2, wed3 плохие p – значения (*Рисунок 12*). Постараемся улучшить модель, исключив их из нее.

```
Coefficients:
      Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.65115    0.03951  -16.482  < 2e-16 ***
age          -0.09401    0.01311   -7.173 8.22e-13 ***
sex           0.49789    0.02508   19.849 < 2e-16 ***
higher_educ   0.52243    0.02561   20.396 < 2e-16 ***
status2       0.31293    0.02650   11.809 < 2e-16 ***
dur           0.14009    0.01210   11.582 < 2e-16 ***
wed1          0.06511    0.03531    1.844  0.0653 .
wed2          0.09665    0.04590    2.105  0.0353 *
wed3         -0.10766    0.04607   -2.337  0.0195 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9118 on 6045 degrees of freedom
Multiple R-squared:  0.1698,    Adjusted R-squared:  0.1687
F-statistic: 154.6 on 8 and 6045 DF,  p-value: < 2.2e-16
```

Рисунок 12" Характеристики модели зависимости параметра salary от age, sex, higher_educ, status2, dur, wed1, wed2, wed3"

Видим, что R^2 очень мало упал, а все p – значения теперь хорошие и все коэффициенты при вызове `vif` небольшие (Рисунок 13).

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.58891    0.02618  -22.493 < 2e-16 ***
age          -0.08742    0.01270   -6.884 6.43e-12 ***
sex           0.49176    0.02428   20.253 < 2e-16 ***
higher_educ   0.52414    0.02556   20.509 < 2e-16 ***
status2       0.31132    0.02646   11.766 < 2e-16 ***
dur           0.13987    0.01210   11.563 < 2e-16 ***
wed3         -0.15999    0.03754   -4.262 2.06e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.912 on 6047 degrees of freedom
Multiple R-squared:  0.1691,    Adjusted R-squared:  0.1683
F-statistic: 205.2 on 6 and 6047 DF,  p-value: < 2.2e-16

> vif(model3)
              age              sex higher_educ      status2              dur              wed3
1.173886      1.056994      1.036421      1.017142      1.064790      1.166445

```

Рисунок 13 "Характеристики модели зависимости параметра salary от age, sex, higher_educ, status2, dur, wed3"

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

Изначально будем вводить степени их можно использовать только у параметров *age* и *dur*, так как они вещественные.

Создаем модель со степенью 0.1, полученная модель имеет коэффициент детерминации 18,41%, что является достаточно хорошим показателем, но p -значение у $I(dur^{0.1})$ очень плохое (Рисунок 14), уберем его и посмотрим на модель.

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.29115    0.97914   -2.340 0.019503 *
age          -0.53229    0.14974   -3.555 0.000398 ***
sex           0.36372    0.06338    5.739 1.30e-08 ***
higher_educ   0.64925    0.07687    8.446 < 2e-16 ***
status2       0.37943    0.06586    5.762 1.14e-08 ***
dur           0.06629    0.05446    1.217 0.223879
wed3         -0.01680    0.20965   -0.080 0.936136
I(age^0.1)    1.15072    0.96228    1.196 0.232076
I(dur^0.1)    1.01280    0.63929    1.584 0.113484
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9399 on 904 degrees of freedom
(5141 observations deleted due to missingness)
Multiple R-squared:  0.1841,    Adjusted R-squared:  0.1769
F-statistic: 25.49 on 8 and 904 DF,  p-value: < 2.2e-16

```

Рисунок 14 "Характеристики модели зависимости параметра salary от age, sex, higher_educ, status2, dur, wed3, $I(age^{0.1})$ и $I(dur^{0.1})$ "

Убрав $I(dur^{0.1})$ из модели получаем модель с плохими показателями у `wed3` и $I(age^{0.1})$ (Рисунок 15). Исключить их из модели нет возможности, так как нельзя потерять основной параметр или вернуться к исходной модели. Остается только повысить степень.

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.93470    0.40607  -2.302   0.0214 *
age          -0.36660    0.06945  -5.278  1.4e-07 ***
sex           0.37184    0.03441  10.805 < 2e-16 ***
higher_educ   0.56149    0.03724  15.080 < 2e-16 ***
status2       0.33661    0.03644   9.237 < 2e-16 ***
dur           0.12763    0.01594   8.006  1.7e-15 ***
wed3         -0.02609    0.10020  -0.260   0.7946
I(age^0.1)    0.63759    0.47551   1.341   0.1801
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8863 on 2883 degrees of freedom
(3163 observations deleted due to missingness)
Multiple R-squared:  0.1673,    Adjusted R-squared:  0.1653
F-statistic: 82.75 on 7 and 2883 DF,  p-value: < 2.2e-16

```

Рисунок 15 "Характеристики модели зависимости параметра `salary` от `age`, `sex`, `higher_educ`, `status2`, `dur`, `wed3`, $I(age^{0.1})$ "

Повышаем степень на 0.1 получается модель с характеристиками, представленными на Рисунок 16. Поведение этой модели схоже с предыдущей, следовательно, можно увеличить степень на больший показатель.

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.45540    0.50704  -2.870 0.004195 **
age          -0.56118    0.16986  -3.304 0.000992 ***
sex           0.36322    0.06335   5.734 1.34e-08 ***
higher_educ   0.64875    0.07683   8.444 < 2e-16 ***
status2       0.37938    0.06583   5.763 1.13e-08 ***
dur           0.04201    0.06101   0.689 0.491275
wed3         -0.01656    0.20951  -0.079 0.937005
I(age^0.2)    0.72075    0.59327   1.215 0.224731
I(dur^0.2)    0.66906    0.36895   1.813 0.070100 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9395 on 904 degrees of freedom
(5141 observations deleted due to missingness)
Multiple R-squared:  0.1848,    Adjusted R-squared:  0.1776
F-statistic: 25.62 on 8 and 904 DF,  p-value: < 2.2e-16

```

Рисунок 16 "Характеристики модели зависимости параметра `salary` от `age`, `sex`, `higher_educ`, `status2`, `dur`, `wed3`, $I(age^{0.2})$ и $I(dur^{0.2})$ "

Увеличиваем показания степени на 0.2, получается модель со степенями 0.4 Получили модель с $R^2 = 0.1865$, что очень хорошо, у `wed3` p -значение плохое, но его мы не можем убрать, убираем $I(age^{0.4})$. Получили модель с $R^2 = 0.1804$, что очень хорошо, у `dur` p -

значение плохое, но его мы не можем убрать, убрав $I(dur^{0.4})$ возвращаемся к исходной модели. Следовательно, повышая степень на 0.1 наша модель не улучшается, попробуем увеличить степень на 1

Возьмем степень 2. Значение $R^2 = 0.1839$ и у всех параметров р-значение хорошее, эта модель хорошая (Рисунок 17).

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.491119   0.027734 -17.708 < 2e-16 ***
age          -0.047369   0.013169  -3.597 0.000325 ***
sex           0.501056   0.024171  20.730 < 2e-16 ***
higher_educ   0.523354   0.025338  20.655 < 2e-16 ***
status2       0.325024   0.026299  12.359 < 2e-16 ***
dur           0.152018   0.014115  10.770 < 2e-16 ***
wed3         -0.057616   0.038582  -1.493 0.135401
I(age^2)      -0.113473   0.011423  -9.934 < 2e-16 ***
I(dur^2)      -0.011490   0.004019  -2.859 0.004270 **
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.904 on 6045 degrees of freedom
Multiple R-squared:  0.1839,    Adjusted R-squared:  0.1828
F-statistic: 170.2 on 8 and 6045 DF,  p-value: < 2.2e-16
```

Рисунок 17 "Характеристики модели зависимости параметра salary от age, sex, higher_educ, status2, dur, wed3, $I(age^2)$ и $I(dur^2)$ "

Получив хорошую модель с квадратами, пробуем вводить логарифмы. Получили модель с $R^2 = 0.1834$, не высокий, у dur, wed3, $I(\log(age))$, $I(\log(dur))$ плохие значения. Попробуем убрать из модели: $(\log(dur))$, уберем $I(\log(dur))$ и посмотрим на модель (Рисунок 18). R^2 уменьшился, но незначительно. р-значение у $I(\log(age))$ плохое, следовательно, необходимо убрать $I(\log(age))$. Но тогда вернемся к исходной модели.

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.31088    0.07616  -4.082 4.58e-05 ***
age          -0.35364    0.06227  -5.679 1.49e-08 ***
sex           0.37181    0.03441  10.804 < 2e-16 ***
higher_educ   0.56129    0.03723  15.076 < 2e-16 ***
status2       0.33649    0.03644   9.234 < 2e-16 ***
dur           0.12766    0.01594   8.008 1.68e-15 ***
wed3         -0.02602    0.10021  -0.260  0.795
I(log(age))   0.05072    0.03898   1.301  0.193
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8863 on 2883 degrees of freedom
(3163 observations deleted due to missingness)
Multiple R-squared:  0.1673,    Adjusted R-squared:  0.1652
F-statistic: 82.73 on 7 and 2883 DF,  p-value: < 2.2e-16
```

Рисунок 18 "Характеристики модели зависимости параметр salary от age, sex, higher_educ, status2, dur, wed3, $I(\log(age))$ "

Выберем наилучшую модель. Из всех моделей со степенями лучше брать с квадратом, так как в остальных случаях у нас ниже R^2 и больше параметров с плохим р-значением.

3. Выделите наилучшие модели из построенных: по значимости параметров, включённых в зависимости, и по объяснённой с помощью построенных зависимостей разбросу $R^2 - R^2_{adj}$.

Проверим наилучшую модель на линейную зависимость параметров (самые спорные).

Видим, что R^2 у $age \sim I(age^2)$ равен $0.04194 < 1$, значит их можно использовать в одной модели.

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

Обратим внимание на *Рисунок 19* из него можно сделать вывод, что наибольшую зарплату получают мужчины молодого возраста, с высшим образованием, живущие в городе, работающих дольше и состоящие в браке.

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.50615    0.02725 -18.577  < 2e-16 ***
age          -0.04908    0.01316  -3.729 0.000194 ***
sex           0.50596    0.02412  20.973  < 2e-16 ***
higher_educ   0.52472    0.02535  20.700  < 2e-16 ***
status2       0.32795    0.02629  12.472  < 2e-16 ***
dur           0.13088    0.01203  10.879  < 2e-16 ***
wed3         -0.05767    0.03860  -1.494 0.135234
I(age^2)     -0.11463    0.01142 -10.036  < 2e-16 ***
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9045 on 6046 degrees of freedom
Multiple R-squared:  0.1828,    Adjusted R-squared:  0.1818
F-statistic: 193.1 on 7 and 6046 DF,  p-value: < 2.2e-16

```

Рисунок 19 "Характеристики модели зависимости параметра salary от age, sex, higher_educ, status2, dur, wed3, I(age^2)"

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

Обратим внимание на *Рисунок 20* и по нему сделаем вывод, что наивысшую зарплату получают женатые мужчины, с высшим образованием не из города, работающие дольше.


```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.49915    0.03474 -14.370  < 2e-16 ***
age          -0.11275    0.02357  -4.783 1.97e-06 ***
sex           0.44145    0.04159  10.615  < 2e-16 ***
higher_educ  0.32488    0.04650   6.987 4.97e-12 ***
dur           0.09723    0.01734   5.607 2.62e-08 ***
I(age^2)     -0.05998    0.02357  -2.544  0.0111 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6336 on 1055 degrees of freedom
Multiple R-squared:  0.1892,    Adjusted R-squared:  0.1854
F-statistic: 49.23 on 5 and 1055 DF,  p-value: < 2.2e-16

```

Рисунок 20 "Характеристики модели зависимости параметра salary от age, sex, higher_educ, dur, I(age^2)"

Из Рисунок 21 делаем вывод о том, что наивысшую зарплату получают мужчины, разведенные или не вступавший в брак, с высшим образованием, живущие в городе и работающие дольше.

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.148786    0.152091   0.978  0.3289
age           0.008587    0.101598   0.085  0.9327
sex           0.336248    0.232825   1.444  0.1499
status2       0.279199    0.164182   1.701  0.0902 .
dur           0.150071    0.098294   1.527  0.1280
I(age^2)     -0.120587    0.059983  -2.010  0.0454 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.076 on 258 degrees of freedom
Multiple R-squared:  0.06503,    Adjusted R-squared:  0.04691
F-statistic: 3.589 on 5 and 258 DF,  p-value: 0.003723

```

Рисунок 21 "Характеристики модели зависимости параметра salary от age, sex, dur, I(age^2)"

Вывод

Построена модель линейной регрессии зарплаты. Коэффициент вздутия дисперсии VIF при всех параметрах небольшой. Выбрана наилучшая модель с квадратом, где значения $R^2 = 18,28\%$ и p-значения хорошие. По данным лучшей модели выяснили, что большую зарплату получают мужчины молодого возраста, с высшим образованием, живущие в городе, работающих дольше и состоящие в браке. Из выделенных подмножеств больше всего получают женатые мужчины, с высшим образованием не из города, работающие дольше.

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

Задача 4

Необходимо загрузить данные из указанного набора и произвести следующие действия.

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

Тип классификатора: SVM (метод опорных векторов)

Классификация по столбцу: Пол gender

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.

При построении бинарного дерева должны прийти к двум вариантам развития событий: Genre = “Male” или нет. Лучше задавать вопросы, которые делят объекты на равные части. Высота дерева должна быть минимальной, чтобы можно было быстро получать ответ. Высота важна, так как наш классификатор перестанет иметь хорошую обобщающую способность.

При использовании метода опорных векторов предполагаем, что существует плоскость, делящая наши данные на 2 группы наилучшим способом. Находим ее координаты. В случае, если не удастся разделить данные одной плоскостью, то можно построить несколько, т.е. построить каскадный классификатор.

Гиперплоскость имеет вид: $F(x) = \text{sign}(\langle w, x \rangle + b)$, где w – вектор весов, x – входной объект, b – вспомогательный параметр, \langle, \rangle - скалярное произведение.

Если у нас появляется несколько плоскостей, то вместо скалярного произведения используем ядра (Рисунок 22) [1].

Полиномиальное: $k(x, x') = (\langle x, x' \rangle + \text{const})^d$

Радиальная базисная функция: $k(x, x') = e^{-\gamma \|x - x'\|^2}, \gamma > 0.$

Гауссова радиальная базисная функция: $k(x, x') = e^{-\frac{\|x - x'\|^2}{2\sigma^2}}.$

Сигмоид: $k(x, x') = \tanh(\kappa \langle x, x' \rangle + c), \kappa > 0, c < 0.$

Рисунок 22 Произвольные ядра, помогающие строить нелинейные разделители.

Используя метрики precision, recall и F1, учитывающие, что наш класс не сбалансирован, видим, что precision: 0.0, recall: 0.0, f1: 0.0. Из этого делаем вывод о том, что данные не могут быть классифицированы методом опорных векторов.

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?

Случайный леса - алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев

Используя метрики precision, recall и F1, учитывающие, что наш класс не сбалансирован, видим, что precision: 0.0, recall: 0.0, f1: 0.0. Из этого делаем вывод о том, что данные не могут быть классифицированы классификатором типа Случайный лес.

Для оценки качества работы алгоритма вводятся метрики precision, recall и F1.

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

Рисунок 23 "Формула для нахождения метрик precision и recall"

Precision можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными, а recall показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм. Именно введение precision не позволяет записывать все объекты в один класс, так как в этом случае получается рост уровня False Positive. Recall демонстрирует способность алгоритма обнаруживать данный класс вообще, а precision — способность отличать этот класс от других классов.

Существует несколько различных способов объединить precision и recall в агрегированный критерий качества. F-мера (в общем случае F_β) — среднее гармоническое precision и recall :

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

Рисунок 24 "Формула для нахождения метрики F1"

β в данном случае определяет вес точности в метрике, и при $\beta=1$ это среднее гармоническое (с множителем 2, чтобы в случае precision = 1 и recall = 1 иметь F1=1) F-мера достигает максимума при полноте и точности, равными единице, и близка к нулю, если один из аргументов близок к нулю. [2]

Вывод

Не удалось построить классификатор с высокой точностью метода опорных векторов. Так как данные между собой очень похожи. Используя множество решающих деревьев, построили классификатор типа Случайный Лес для решения той же задачи. На основе полученных данных сделан вывод, что данные классификаторы не подходят для классификации HR Analytics: Job Change of Data Scientists данных, но при этом классификатор верно выполняет свою работу.

Код решения задачи и сведения о проверенных моделях приведены в *(Приложение 4.*

Задача 5

Набор данных: Популярные деревья в районах Нью-Йорка.

1. Необходимо провести анализ датасета и сделать обработку данных по предложенному алгоритму.

Анализ датасета и обработку данных выполнила Кармалина Ольга и описала все полученные результаты в своем отчете. Она свела аномалии к минимуму и нормализовала данные. Построив дерево классификаций Ольга получила точность работы классификатора, работающего с несбалансированными данными - 82%. Но более точный классификатор показывает – всего 47%. Это произошло из-за того, что данные очень похожи друг на друга и разделить их было достаточно сложно. Приступим к построению кластеров для целевого признака «tree_dbh». Разобьем на три части данные. Первая часть — это все показатели, которые больше 20, их мы обозначим за 0, те, которые меньше 20, но больше 5 за 1, все остальные за -1. Применяем метод главных компонент. Для объяснения 90% и более дисперсии необходимо два признака, так как при использовании одного данные будут менее корректны (Рисунок 25). Первый признак вносит большой вклад в первую компоненту. Следовательно, переходим к пространству из двух компонент.

```
[0.90866214]
[0.90866214 0.09133786]
[9.08662136e-01 9.13378643e-02 1.82738045e-62]
[9.08662136e-01 9.13378643e-02 8.57121073e-52 1.26366955e-53]
```

Рисунок 25 «Доли дисперсии, которые объясняются одним или несколькими компонентами»

Построив дерево классификаций на основе полученных компонент, видим показатели точности (Рисунок 26). Классификатор, работающий с несбалансированными данными показывает 99%. Классификатор, работающий с более точными данными, показывает – 50%.

```
from sklearn import tree
from sklearn.metrics import f1_score
df3 = df3.astype(str)
target = pd.DataFrame(df3['Y'])
train = pd.DataFrame(df3.drop(['Y'], axis = 1))
X_train, X_test, y_train, y_test = train_test_split(train, target, test_size = 0.3, train_size = 0.7, random_state = 42)
X_train.shape
clf = tree.DecisionTreeClassifier(max_depth=10, random_state=42)
clf.fit(X_train, y_train)
clf.score(X_test, y_test)

0.9899000507623099

y_pred = clf.predict(X_test)
f1_score(y_test, y_pred, average='macro') #

0.4974621968490777
```

Рисунок 26 "Показатели классификаторов, которые работают с несбалансированными данными и сбалансированными"

Данный классификатор работает лучше, чем приведенный у Кармалиной Ольги, но на основе полученных результатов нельзя сказать точно, что работает ли он верно или все данные очень схожи и классификатор работает некорректно.

Проделаем аналогичные действия и построим классификатор для разделения районов по состоянию здоровья деревьев в них. Рассмотрим только плохое и среднее состояние. В данной ситуации у точности – 6%. Это произошло по той же причине, что и в прошлом построении.

Вывод

При разделении деревьев Нью-Йорка на кластеры и при последующем построении классификатора получили хорошую точность. Из-за того, что данные похожи нельзя получить более точные показатели. Также был построен классификатор для разделения районов по состоянию здоровья деревьев в них, но здесь точность получилась очень низкой. Таким образом, первое разделение было лучше.

Код решения задачи и сведения о проверенных моделях приведены в *(Приложение 5)*.

Заключение

В результате первой задачи мы построили две модели. Проверили зависимость между объясняемой переменной и регрессором. Во второй задаче регрессоры были проверены на линейную зависимость. Была выведена наилучшая модель с хорошими R^2 и p-значениями. Найдены доверительные интервалы и сделаны выводы о коэффициентах. В третьей задаче была построена модель линейной регрессии зарплаты. Выведена лучшая модель и по ней сделаны выводы о зарплате. Выяснили, что наибольшую зарплату получают мужчины молодого возраста, с высшим образованием, живущие в городе и состоящие в браке. В четвертой задаче были построены классификаторы – метода опорных векторов и случайный лес. Был сделан вывод о том, что данные HR Analytics: Job Change of Data Scientists не подходят для классификации построенными классификаторами. В пятой задаче были разделены деревья Нью-Йорка на кластеры построен классификатор. Была получена хорошая точность работы классификатора. Также был построен классификатор для разделения районов по состоянию здоровья деревьев в них.

Список литературы

1. Mercer J. Functions of positive and negative type and their connection with the theory of integral equations // Philos. Trans. Roy. Soc. London. — 1909. — Vol. A, no. 209. — Pp. 415–446.
2. Гудфеллоу, Я. "Глубокое обучение" / Я. Гудфеллоу, И. Бенджио, А. Курвилль. – Москва: ДМК Пресс, 2018. – 289, 356 с.

Приложение

(Приложение 1)

```
library("lmtest")
library("GGally")
library("car")

data = swiss

data
summary(data)
ggpairs(data)

#1)
#a)
sum(data$Agriculture) #вычисление количества элементов в Agriculture
sum(data$Agriculture)/47 #ср.значение Agriculture
mean(data$Agriculture) # = 50.66
var(data$Agriculture) #дисперсия Agriculture (=515.7994)
sd(data$Agriculture) # СКО для Agriculture (= 22.7112)

#б)
sum(data$Examination) #вычисление количества элементов в Examination
sum(data$Examination)/47 #ср.значение Examination
mean(data$Examination) # = 16.48936
var(data$Examination) #дисперсия Examination (= 63.64662)
sd(data$Examination) # СКО для Examination (= 7.9779)

#в)
sum(data$Infant.Mortality) #вычисление количества элементов в
Infant.Mortality
sum(data$Infant.Mortality)/47 #ср.значение Infant.Mortality
mean(data$Infant.Mortality) # = 19.94255
var(data$Infant.Mortality) #дисперсия Infant.Mortality (= 8.48)
sd(data$Infant.Mortality) # СКО для Infant.Mortality (= 2.91)

modele1 = lm(Agriculture~Examination, data)
modele1
summary(modele1)

#2a)
#  $F = -1.95 * ex + 82.88$  - зависимость людей, занятых сельским хозяйством от
оценок на экзаменах при поступлении в армию =>
# Если людей, получивших высшие оценки на экзаменах, высокий уровень, то
людей, работающих в с/х меньше.

#3a)
# Модель по коэффициенту детерминации = 0.4713, можем сделать вывод, что
модель относительно хороша:
# коэффициент высок, но для более точной информации нужно добавлять ещё
параметры, от которых зависит число людей,
# работающих в с/х сфере.

#4a)
# Взаимосвязь между объясняемой переменной(Agriculture) и объясняющей
переменной (Examination)
# достаточно высокая, принадлежит промежутку (от 0 до 0.001), равная "****".

modele2 = lm(Agriculture~Infant.Mortality, data)
modele2
summary(modele2)

#2б)
```



```

#F = -0.47 * In + 60.12 - зависимость людей, занятых сельским хозяйством от
смертности детей, а раннем возрасте.
#Аналогичная ситуация, как и с первой моделью. Т.е большая смерность
младенцев приводит к тому, что людей,
#работающих в с/х сфере меньше.

#3б)
# Модель по коэффициенту детерминации = 0.003704, можем сделать вывод, что
модель плоха:
# коэффициент очень низок, возможно, следует построить новую модель, так как в
этой модели
# почти отсутствуют какие-либо взаимосвязи.
#
#4б)
# Взаимосвязь между объясняемой переменной(Agriculture) и объясняющей
переменной (Infant.Mortality)
# низкая, принадлежит промежутку (от 0.1 до 1), равная " ". Следовательно,
взаимосвязи почти нет.

```

(Приложение 2)

```

library("lmtest")
library("GGally")
library("car")

data = swiss

data
summary(data)
ggpairs(data)

# Для того, чтобы построить множественную линейную регрессию, необходимо
проверить
# на линейную зависимость регрессоры (Fertility, Catholic, Infant.Mortality).

#1.а)
modele_1 = lm(Fertility~Catholic, data)
modele_1
summary(modele_1)
# Модель по коэффициенту детерминации = 0.215, можем сделать вывод, что
модель плоха:
# коэффициент очень низок. Следовательно, линейная зависимость этих двух
регрессоров почти
# отсутствует.

#1.б)
modele_2 = lm(Fertility~Infant.Mortality, data)
modele_2
summary(modele_2)
# Модель по коэффициенту детерминации = 0.1735. Линейная зависимость между
# Fertility и Infant.Mortality почти отсутствует.

#1.в)
modele_3 = lm(Catholic~Infant.Mortality, data)
modele_3
summary(modele_3)
# Модель по коэффициенту детерминации = 0.0308, можем сделать вывод, что
модель плоха:
# коэффициент очень низок, следовательно, линейная зависимость почти
отсутствует.

# Исходя из выше сказанного, можно сделать вывод о том, что множественную
линейную регрессию
# можно построить из заданных регрессоров.

```

```

#2
model= lm(Examination ~ Fertility + Catholic + Infant.Mortality, data)
model
summary(model)
#2.a)
# Модель по коэффициенту детерминации = 0.5391, можем сделать вывод, что
# модель относительно хороша:
# коэффициент высок. Линейная зависимость присутствует.

#2.б)
# Модель относительно р-статистики относительно хороша, два показателя
# взаимосвязей достаточно велики,
# то есть их значения варьируются от 0 до 0.01. р-статистика у
# Infant.Mortality плохая, можно попробовать создать модель
# без этого регрессора:

model_test= lm(Examination ~ Fertility + Catholic, data)
model_test
summary(model_test)

# Убрав из 1-го варианта модели - Infant.Mortality, R^2 - изменился не очень
# сильно, р-статистика - стала лучше,
# следовательно, это изменение было правильным и обоснованным.

# Проанализировав поведение модели в R^2 и р-статистике можно сделать вывод,
# что
# модель относительно хороша, но требует корректировок, которые помогут найти
# больше общих взаимосвязей.
# Исключив из модели Infant.Mortality, получили более улучшенную модель из
# чего мы сделали вывод,
# что смертность детей не влияет на линейную зависимость военных экзаменов от
# католиков и рождаемости.
# То есть смертность детей почти не изменяет показания у католиков и
# рождаемости.

#3

model_test2 = lm(Examination ~ Fertility + Catholic + Infant.Mortality +
I(log10(Fertility)) + I(log10(Catholic)) + I(log10(Infant.Mortality)), data)
model_test2
summary(model_test2)

vif(model_test2)
# Так как vif(I(log10(Fertility))) имеет наибольшую линейную зависимость,
# следовательно,
# можно попробовать убрать его из модели.

model_test3 = lm(Examination ~ Fertility + Catholic + Infant.Mortality +
(log10(Catholic)) + I(log10(Infant.Mortality)), data)
model_test3
summary(model_test3)

vif(model_test3)

# Наибольший vif > 10, у I(log10(Infant.Mortality)). Попробуем избавиться от
# этого значения
# и оценить характеристики новой модели.

model_test4 = lm(Examination ~ Fertility + Catholic + Infant.Mortality +
(log10(Catholic)), data)
model_test4
summary(model_test4)

vif(model_test4)

```

```

# В новой модели Catholic линейно раскладывается по остальным регрессорам.

model_test5 = lm(Examination ~ Fertility + Infant.Mortality +
(log10(Catholic)), data)
model_test5
summary(model_test5)

vif(model_test5)

# Последняя модель не имеет ярко выраженных линейных зависимостей.  $R^2 = 0.4944$ , что
# является относительно хорошим показателем. р-статистика регрессоров
неплохая, что говорит нам о
# том, что модель относительно хороша, но требует корректировок.

# Модель, приведённая в пункте 2 относительно лучше, модели приведённой в
пункте 3. Так как
# оцениваемые характеристики ( $R^2$  и р-статистики) немного лучше в первой
модели.

#4
# Создаём модель со всевозможными произведениями пар регрессоров и
квадратов.
model_5 = lm(Examination ~ Fertility + Catholic + Infant.Mortality +
I(Fertility^2) + I(Catholic^2) + I(Infant.Mortality^2) +
I(Infant.Mortality * Fertility) + I(Infant.Mortality * Catholic) +
I(Fertility * Catholic), data)
model_5
summary(model_5)

vif(model_5)

# Далее постепенно анализируя vif(регрессор), убираем из модели
vif(регрессор) у которого показатель
# больше 10 и больше всех остальных показателей.

# Убираем I(Infant.Mortality * Fertility).
model_6 = lm(Examination ~ Fertility + Catholic + Infant.Mortality +
I(Fertility^2) + I(Catholic^2) + I(Infant.Mortality^2) +
I(Infant.Mortality * Catholic) + I(Fertility * Catholic), data)
model_6
summary(model_6)

vif(model_6)

# Убираем Catholic.
model_7 = lm(Examination ~ Fertility + Infant.Mortality + I(Fertility^2) +
I(Catholic^2) + I(Infant.Mortality^2) + I(Infant.Mortality * Catholic) +
I(Fertility * Catholic), data)
model_7
summary(model_7)

vif(model_7)

# Убираем I(Fertility * Catholic).
model_8 = lm(Examination ~ Fertility + Infant.Mortality + I(Fertility^2) +
I(Catholic^2) + I(Infant.Mortality^2) + I(Infant.Mortality * Catholic), data)
model_8
summary(model_8)

vif(model_8)

# Убираем I(Fertility^2).
model_9 = lm(Examination ~ Fertility + Infant.Mortality + I(Catholic^2) +
I(Infant.Mortality^2) + I(Infant.Mortality * Catholic), data)
model_9

```

```

summary(model_9)

vif(model_9)

# Убираем I(Infant.Mortality^2).
model_10 = lm(Examination ~ Fertility + Infant.Mortality + I(Catholic^2) +
I(Infant.Mortality*Catholic), data)
model_10
summary(model_10)

vif(model_10)

# Убираем I(Catholic^2).
model_11 = lm(Examination ~ Fertility + Infant.Mortality +
I(Infant.Mortality*Catholic), data)
model_11
summary(model_11)

vif(model_11)

# Таким образом, model_11 хорошая модель. R^2 = 0.5164, что показывает нам
достаточно хорошую линейную зависимость.
# р-статистика показывает неплохие результаты у регрессоров.

# Так как в model_10 у регрессоров I(Catholic^2) и I(Infant.Mortality *
Catholic) почти одинаковые показатели vif.
# Попробуем создать еще одну хорошую модель.

# Убираем I(Infant.Mortality * Catholic).
model_12 = lm(Examination ~ Fertility + Infant.Mortality + I(Catholic^2),
data)
model_12
summary(model_12)

vif(model_12)

# Действительно, убрав в model_10 регрессор(I(Infant.Mortality *
Catholic)), мы получаем хорошую модель с vif
# показателями меньше 3. По R^2 модель не уступает предыдущим его показатель
равен 0.5497. р-статистика
# также показывает хорошие результаты. Следовательно, эта модель хорошая.

# В модели 12 показатели R^2 и р-статистика достаточно хороши,
следовательно, эта модель является
# наилучшей.

# Рассмотрим доверительные интервалы модели
modele = lm(Examination ~ Fertility + Infant.Mortality, data)
modele
summary(modele)

#коэффициенты
coef(modele)
#доверительные интервалы коэффициентов
confint(modele, level = 0.90)

#47 наблюдений, оценивалось 4 коэффициента: 47 - 4 = 43 степени свободы
# Доверительный интервал для Fertility [-0.46240 - t * 0.07881;-0.46240 + t *
0.07881]
t_critical = qt(0.975, df = 149) #~1.976 близок к 1.96
se = 0.07881
modele$coefficients[2] - t_critical * se
modele$coefficients[2] + t_critical * se
# [-0.6181266; -0.3066674]
# проверка

```

```

confint(modele, level = 0.95)
# 0 не попал в интервал, значит коэффициент не может быть равен 0

# Доверительный интервал для Infant.Mortality[0.51376 - t * 0.33799; 0.51376
+ t * 0.33799]
t_critical = qt(0.975, df = 149) #~1.976 близок к 1.96
se = 0.33799
modele$coefficients[3] - t_critical * se
modele$coefficients[3] + t_critical * se
# [-0.1541124; 1.181633]
# проверка
confint(modele, level = 0.95)
# 0 попал в интервал, значит коэффициент может быть равен 0

# Доверительный интервал для temp [1.65209 - t * 0.25353 ; 1.65209 + t *
0.25353 ]
t_critical = qt(0.975, df = 149) #~1.976 близок к 1.96
se = 0.25353
modele$coefficients[4] - t_critical * se
modele$coefficients[4] + t_critical * se
# [1.151114 ; 2.153072]
# проверка
confint(modele, level = 0.95)
# 0 не попал в интервал, значит коэффициент не может быть равен 0
# Все регрессоры имеют некоторую зависимость с объясняемой переменной

# Построение доверительного интервала для прогноза
modele_pr = lm(Examination~Fertility + Catholic + Infant.Mortality, data)
modele_pr
summary(modele_pr)

new.data = data.frame(Fertility = 20, Catholic = 10, Infant.Mortality = 10)
predict(modele_pr, new.data, interval = "confidence")
# fit - значение прогноза = 31.61013
# нижняя и верхняя граница lwr = 23.75496 upr = 39.4653

```

(Приложение 3)

```

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\\79040\\Desktop\\R\\r21i_os26c.sav")
glimpse(data)
data2 = select(data, qj13.2, q_age, qh5, q_educ, status, qj6.2, q_marst)

#исключаем строки с отсутствующими значениями NA
data2 = na.omit(data2)

glimpse(data2)

#зарплата с элементами нормализации
data2$qj13.2
sal = as.numeric(data2$qj13.2)
sal1 = as.character(data2$qj13.2)
sal2 = lapply(sal1, as.integer)
sal = as.numeric(unlist(sal2))

```

```

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

#возраст с элементами нормализации
age1 = as.character(data2$q_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$qh5
#data2["sex"] = lapply(data2$qh5, as.character)
#data2$sex[which(data2$sex != '1')] <- 0
#data2$sex[which(data2$sex == '1')] <- 1
data2$sex = as.numeric(data2$qh5)
data2$sex[which(data2$sex != 1)] <- 0
data2$sex[which(data2$sex == 1)] <- 1

#образование
data2["h_educ"] = data2$q_educ
#data2["h_educ"] = lapply(data2$q_educ, as.character)
data2["higher_educ"] = data2$q_educ
data2["higher_educ"] = 0
data2$higher_educ[which(data2$q_educ == 21)] <- 1 # есть диплом о высшем образовании
data2$higher_educ[which(data2$q_educ == 22)] <- 1 # аспирантура и т.п. без диплома
data2$higher_educ[which(data2$q_educ == 23)] <- 1 # аспирантура и т.п. с дипломом

#населенный пункт
data2["status1"] = data2$status
#data2["status1"] = lapply(data2$status, as.character)
data2["status2"] = 0
data2$status2[which(data2$status1 == 1)] <- 1 # Областной центр
data2$status2[which(data2$status1 == 2)] <- 1 # Город
data2$status2 = as.numeric(data2$status2)

#продолжительность рабочей недели
dur1 = as.character(data2$qj6.2)
dur2 = lapply(dur1, as.integer)
dur3 = as.numeric(unlist(dur2))
data2["dur"] = (dur3 - mean(dur3)) / sqrt(var(dur3))

#семейное положение
data2["wed"] = data2$q_marst
#data2["wed"] = lapply(data2$q_marst, as.character)
data2$wed1 = 0
data2$wed1[which(data2$wed == 2)] <- 1 # Состоите в зарегистрированном браке
data2$wed1 = as.numeric(data2$wed1)

data2["wed2"] = lapply(data2["wed"], as.character)
data2$wed2 = 0
data2$wed2[which(data2$wed == 4)] <- 1 # Разведены и в браке не состоите
data2$wed2[which(data2$wed == 5)] <- 1 # Вдовец (вдова)
data2$wed2 = as.numeric(data2$wed2)

data2["wed3"] = data2$q_marst
data2$wed3 = 0
data2$wed3[which(data2$wed == 1)] <- 1 # Никогда в браке не состояли
data2$wed3 = as.numeric(data2$wed3)

data3 = select(data2, salary, age, sex, higher_educ, status2, dur, wed1, wed2, wed3)

```

```

#построение зависимостей для всех данных
modell1 = lm(data = data3, salary~age + sex + higher_educ + status2 + dur +
wed1 + wed2 + wed3)
summary(modell1)
vif(modell1)
# R^2 = 0.1698, не очень низкий
# p - зависимости только у wed1, wed2, wed3 плохие, у остальных очень хорошие
# улучшим зависимость, убрав из нее параметры с большими коэф в vif

model2 = lm(data = data3, salary~age + sex + higher_educ + status2 + dur +
wed2 + wed3)
summary(model2)
vif(model2)

model3 = lm(data = data3, salary~age + sex + higher_educ + status2 + dur +
wed3)
summary(model3)
vif(model3)
waldtest(model3, model2, modell1)
# Теперь R^2 = 0.1691, упал на 0.0007
# все p-зависимости хорошие
# модель работает лучше

# Попробуем улучшить модель с помощью степеней и логарифмов
model_step1 = lm(data = data3, salary~age + sex + higher_educ + status2 + dur
+ wed3 + I(age^0.1) + I(dur^0.1))
summary(model_step1)
vif(model_step1)
# получили модель с R^2 = 0.1841, что очень хорошо
# но p - значение у I(dur^0.1) очень плохое, уберем его и посмотрим на модель

model_step2 = lm(data = data3, salary~age + sex + higher_educ + status2 + dur
+ wed3 + I(age^0.1))
summary(model_step2)
vif(model_step2)
# Получили модель с R^2 = 0.1673, что очень хорошо, у wed3 p-значение
плохое, но его мы не можем убрать,
# Убрав I(age^0.1) возвращаемся к исходной модели: model3.
# Поэтому просто увеличиваем степени

model_step3 = lm(data = data3, salary~age + sex + higher_educ + status2 + dur
+ wed3 + I(age^0.2) + I(dur^0.2))
summary(model_step3)
vif(model_step3)
# Получили модель с R^2 = 0.1847, что очень хорошо.
# p - значение у I(age^0.2) очень плохое, уберем его и посмотрим на модель

model_step4 = lm(data = data3, salary~age + sex + higher_educ + status2 + dur
+ wed3 + I(dur^0.2))
summary(model_step4)
vif(model_step4)
# Убрав I(dur^0.2) возвращаемся к исходной модели: model3.
# Поэтому просто увеличиваем степени

model_step5 = lm(data = data3, salary~age + sex + higher_educ + status2 + dur
+ wed3 + I(age^0.4) + I(dur^0.4))
summary(model_step5)
vif(model_step5)
# Получили модель с R^2 = 0.1865, что очень хорошо, у wed3 p-значение
плохое, но его мы не можем убрать,
# Убраем I(age^0.4).

model_step6 = lm(data = data3, salary~age + sex + higher_educ + status2 + dur
+ wed3 + I(dur^0.4))
summary(model_step6)
vif(model_step6)

```



```

# Получили модель с  $R^2 = 0.1804$ , что очень хорошо, у dur p-значение плохое,
но его мы не можем убрать,
# Убрав  $I(dur^{0.4})$  возвращаемся к исходной модели: model3.
# Увеличивая степени на 0.1 мы не приходим к тому, что наша модель улучшается,
попробуем увеличить степень на 1

model_step7 = lm(data = data3, salary~age + sex + higher_educ + status2 + dur
+ wed3 + I(age^2) + I(dur^2))
summary(model_step7)
vif(model_step7)
# Значение  $R^2 = 0.1839$  и у всех параметров p-значение хорошее, эта модель
хорошая

model_step7_1 = lm(data = data3, salary~age + sex + higher_educ + status2 +
dur + wed3 + I(age^2))
summary(model_step7_1)
vif(model_step7_1)
# Значение  $R^2 = 0.1828$  и у всех параметров p-значение хорошее, эта модель
хорошая, немного лучше предыдущей

model8 = lm(data = data3, salary~age + sex + higher_educ + status2 + dur +
wed3 + I(log(age)) + I(log(dur)))
summary(model8)
vif(model8)
# Получили модель с  $R^2 = 0.1834$ , не высокий.
# У dur, wed3,  $I(log(age))$ ,  $I(log(dur))$  плохие значения. Попробуем убрать из
модели:
#  $I(log(dur))$ 

model9 = lm(data = data3, salary~age + sex + higher_educ + status2 + dur +
wed3 + I(log(age)))
summary(model9)
vif(model9)
#  $R^2$  уменьшился, но незначительно. p-значение у  $I(log(age))$  плохое,
следовательно,
# необходимо убрать  $I(log(age))$ . Но тогда мы вернемся к исходной модели.

# Выберем наилучшую модель. Из всех моделей со степенями лучше брать с
квадратом,
# т.к. в остальных случаях у нас ниже  $R^2$  и больше параметров с плохим p-
значением.
# проверим модель model_step7 на линейную зависимость параметров (самые
спорные)
modele_1 = lm(age~I(age^2), data3)
modele_1
summary(modele_1) #  $R^2 = 0.04194 < 1$ , значит нет линейной зависимости
# и их можно использовать в одной модели

# Ищем подмножество женатых, не из города:
data4 = subset(data3, wed1 == 1)
data4

data5 = subset(data4, status2 == 0)
data5

# Ищем подмножество разведённых или не вступавших в брак, с высшим
образованием
data6 = subset(data3, wed2 == 1)
data6

data7 = subset(data6, higher_educ == 1)
data7

```

```

model_subset1 = lm(data = data5, salary~age + sex + higher_educ + dur+
I(age^2))
summary(model_subset1)
# R^2 = 0.1892, все параметры значимые
# модель хорошая
# Наивысшую зарплату получают мужчины, с высшим образованием и работающие
дольше

model_subset2 = lm(data = data7, salary~age + sex + status2 + dur + I(age^2))
summary(model_subset2)
# R^2 = 0.06503, все параметры значимые
# модель плохая
# Наивысшую зарплату получают мужчины, живущие в городе и работающие
дольше

```

(Приложение 4)

```

import pandas
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
import pandas as pd
data = pandas.read_csv('/content/drive/MyDrive/Colab Notebooks/aug_test.csv',
index_col='city')
data_sel = data.loc[:, data.columns.isin(['enrollee_id', 'city_development
_index', 'gender', 'lastnewjob', 'training_hours', 'target'])] # ,
data_sel = data_sel.dropna()
data_sel['gender'] = np.where(data_sel['gender'] == 'Male', 0, 1)
Survived = data_sel.loc[:, data_sel.columns.isin(['gender'])]
X = data_sel.loc[:, data_sel.columns.isin(['enrollee_id', 'city_development
_index', 'lastnewjob', 'training_hours', 'target'])] #

print(X)
from sklearn.model_selection import train_test_split
x_train, x_validation, y_train, y_validation = train_test_split(X, Survived,
test_size=.33, random_state=1)

T = DecisionTreeClassifier(random_state=241, max_depth = 4)

T = T.fit(x_train, y_train)
T

from google.colab import drive
drive.mount('/content/drive')

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import StratifiedShuffleSplit
from sklearn.model_selection import cross_val_score

#подбор параметров разделяющей гиперплоскости и гиперсферы по методу опорных
векторов , , 'decision_function_shape':('ovo','ovr'), 'gamma':
(1,2,3,'auto'),'shrinking':(True,False)
svm = SVC()
parameters = {'kernel':('linear', 'rbf'), 'C':(1,0.25,0.5,0.75),
'decision_function_shape':('ovo','ovr'), 'gamma': (1,2,3,'auto')}
clf = GridSearchCV(svm, parameters)
y_train = pd.DataFrame(y_train)
x_train = pd.DataFrame(x_train)
y_validation = pd.DataFrame(y_validation)
print('x_train: ', x_train)

```

```

print('y_train: ', y_train)
clf.fit(x_train.values, y_train.values.ravel())
print("SVM")
print("accuracy_1:" + str(np.average(cross_val_score(clf, x_validation,
y_validation, scoring='accuracy'))))
print("f1_1:" + str(np.average(cross_val_score(clf, x_validation, y_validation,
scoring='f1'))))
print("precision_1:" + str(np.average(cross_val_score(clf, x_validation,
y_validation, scoring='precision'))))
print("recall_1:" + str(np.average(cross_val_score(clf, x_validation,
y_validation, scoring='recall'))))

#логистическая регрессия
grid={ "C":np.logspace(-3,3,7), "penalty":["l1","l2"]}# l1 lasso l2 ridge
logreg=LogisticRegression()
logreg_cv=GridSearchCV(logreg,grid,cv=10)
logreg_cv.fit(x_train,y_train)

print("tuned hyperparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy_2 :",logreg_cv.best_score_)
print("accuracy_2:" + str(np.average(cross_val_score(logreg_cv, x_validation,
y_validation, scoring='accuracy'))))
print("f1_2:" + str(np.average(cross_val_score(logreg_cv, x_validation,
y_validation, scoring='f1'))))
print("precision_2:" + str(np.average(cross_val_score(logreg_cv, x_validation,
y_validation, scoring='precision'))))
print("recall_2:" + str(np.average(cross_val_score(logreg_cv, x_validation,
y_validation, scoring='recall'))))

parameters = {
    "loss":["deviance"],
    "learning_rate": [0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2],
    "min_samples_split": np.linspace(0.1, 0.5, 12),
    "min_samples_leaf": np.linspace(0.1, 0.5, 12),
    "max_depth": [3,5,8],
    "max_features": ["log2", "sqrt"],
    "criterion": ["friedman_mse", "mae"],
    "subsample": [0.5, 0.618, 0.8, 0.85, 0.9, 0.95, 1.0],
    "n_estimators": [10]
}

#градиентный бустинг
clf = GridSearchCV(GradientBoostingClassifier(), parameters, cv=10, n_jobs=-
1)

clf.fit(x_train, y_train)
print(clf.score(x_train, y_train))
print(clf.best_params_)
print("accuracy_3:" + str(np.average(cross_val_score(clf, x_validation,
y_validation, scoring='accuracy'))))
print("f1_3:" + str(np.average(cross_val_score(clf, x_validation, y_validation,
scoring='f1'))))
print("precision_3:" + str(np.average(cross_val_score(clf, x_validation,
y_validation, scoring='precision'))))
print("recall_3:" + str(np.average(cross_val_score(clf, x_validation,
y_validation, scoring='recall'))))

accuracy_1:0.8859813084112149
f1_1:0.0
precision_1:0.0
recall_1:0.0

tuned hyperparameters :(best parameters) {'C': 0.001, 'penalty': 'l2'}
accuracy_2 : 0.9079170914033302
f1_2: 0.0
precision_2: 0.0

```

recall_2: 0.0

0.9079189686924494

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import StratifiedShuffleSplit
from sklearn.model_selection import cross_val_score
```

```
parameters = {
    "loss": ["deviance"],
    "learning_rate": [0.05],
    "min_samples_split": np.linspace(0.39, 0.39),
    "min_samples_leaf": np.linspace(0.1, 0.1),
    "max_depth": [5],
    "max_features": ["log2"],
    "criterion": ["friedman_mse"],
    "subsample": [0.9],
    "n_estimators": [10]
}
```

```
clf = GridSearchCV(GradientBoostingClassifier(), parameters, cv=10, n_jobs=-1)
```

```
clf.fit(x_train, y_train)
print(clf.score(x_train, y_train))
print(clf.best_params_)
print("f1:" + str(np.average(cross_val_score(clf, x_validation, y_validation,
scoring='f1'))))
print("precision:" + str(np.average(cross_val_score(clf, x_validation,
y_validation, scoring='precision'))))
print("recall:" + str(np.average(cross_val_score(clf, x_validation,
y_validation, scoring='recall'))))
```

0.9079189686924494

f1:0.0

precision: 0.0

recall: 0.0

Type Markdown and LaTeX: α^2

Случайный лес

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

```
param_grid = { 'n_estimators': [200, 300, 400], 'max_features':
['auto'], 'max_depth' : list(range(1, 20)), 'criterion' : ['gini']}
```

```
RFC = GridSearchCV(estimator=RandomForestClassifier(), param_grid=param_grid,
cv= 5, refit = True)
RFC.fit(x_train, y_train)
```

```
print("accuracy:" + str(np.average(cross_val_score(RFC.best_estimator_,
x_validation, y_validation, scoring='accuracy'))))
print("f1:" + str(np.average(cross_val_score(RFC.best_estimator_, x_validation,
y_validation, scoring='f1'))))
print("precision:" + str(np.average(RFC(grid_search_cv.best_estimator_,
x_validation, y_validation, scoring='precision'))))
print("recall:" + str(np.average(RFC(grid_search_cv.best_estimator_,
x_validation, y_validation, scoring='recall'))))
```

```

from sklearn.model_selection import cross_val_score
print("accuracy:"+str(np.average(cross_val_score(RFC.best_estimator_,
x_validation, y_validation, scoring='accuracy'))))
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'))))

```

(Приложение 5)

```

#импортируем внешние модули и библиотеки
import warnings
warnings.filterwarnings("ignore")

import pandas as pd # библиотека для работы с наборами данных
import matplotlib.pyplot as plt # библиотека для визуализации
import numpy # структура данных ndarray, статистики (хотя там много всего)

import seaborn as sns # более продвинутая библиотека для визуализации
данных
sns.set(style="white", color_codes=True)

# чтобы изображения отображались прямо в ноутбуке
# %matplotlib inline

from google.colab import drive
drive.mount('/content/drive')

#считаем данные и посмотрим на первые 5 строк
data = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/2015-street-
tree-census-tree-data.csv")
data.head()

#пропуски в данных
data.info() #крайне удивительно, но пропусков у нас нет.
# Практически идеальный датасет

#избавимся от ненужных столбцов
data = data.drop('state', 1) # данный столбец нам не интересен т.к. он
везде одинаковый и имеющий значение New York
data = data.drop('user_type', 1) # данный столбец нам не интересен т.к.
не важно кто собрал информацию
data = data.drop('created_at', 1) # нам не важна дата сбора информации о
дереве

#общая статистика по каждому столбцу
data.describe()
# Видим, что есть аномальные значения такие, как tree_dbh - Диаметр
дерева, измеренный примерно на высоте 54 дюйма / 137 см над землей

# В целом данный датасет хороший
# Ящик с усами (диаграмма размаха)
f, axes = plt.subplots(2, 1)

sns.boxplot(data.tree_dbh, palette="PRGn", ax=axes[0])
sns.distplot(data.tree_dbh, ax=axes[1])

f, axes = plt.subplots(2, 1)

sns.boxplot(data.borocode, palette="PRGn", ax=axes[0])

```

```

sns.distplot(data.borocode, ax=axes[1])

# Нормализация значений в признаке tree_dbh, тк другие данные в норме
data.loc[data.tree_dbh >= 40, 'tree_dbh'] -=20
data.loc[data.tree_dbh >= 100, 'tree_dbh'] //= 10
data.loc[data.tree_dbh >= 300, 'tree_dbh'] //=15

f, axes = plt.subplots(2, 1)

sns.boxplot(data.tree_dbh, palette="PRGn", ax=axes[0])
sns.distplot(data.tree_dbh, ax=axes[1])

# Посмотрим на корреляции между всеми признаками
data.corr()
# Линейная зависимость между парраметрами здесь нет

# Целевым признаком является - tree_dbh, тк именно по нему нам надо
выделить районы с аномально хорошим или плохим состоянием деревьев
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('seaborn')
import warnings
warnings.simplefilter('ignore')
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import f1_score, confusion_matrix,
classification_report
from sklearn.model_selection import train_test_split
from sklearn import preprocessing

dataf = data
dataf.drop(dataf[dataf['tree_dbh'] >= 20].index, inplace=True)
dataf['curb_loc'] = np.where(dataf['curb_loc'] == 'OnCurb', 0, 1)
dataf['status'] = np.where(dataf['status'] == 'Alive', 1, 0)
dataf['steward'] = np.where(dataf['steward'] == 'None', 0, 1)
dataf['guards'] = np.where(dataf['guards'] == 'None', 0, 1)
dataf['sidewalk'] = np.where(dataf['sidewalk'] == 'NoDamage', 1, 0)
dataf['problems'] = np.where(dataf['problems'] == 'None', 1, 0)
dataf['root_stone'] = np.where(dataf['root_stone'] == 'None', 1, 0)
dataf['root_grate'] = np.where(dataf['root_grate'] == 'None', 1, 0)
dataf['root_other'] = np.where(dataf['root_other'] == 'None', 1, 0)
dataf['trunk_wire'] = np.where(dataf['trunk_wire'] == 'None', 1, 0)
dataf['trnk_light'] = np.where(dataf['trnk_light'] == 'None', 1, 0)
dataf['trnk_other'] = np.where(dataf['trnk_other'] == 'None', 1, 0)
dataf['brch_light'] = np.where(dataf['brch_light'] == 'None', 1, 0)
dataf['brch_shoe'] = np.where(dataf['brch_shoe'] == 'None', 1, 0)
dataf['brch_other'] = np.where(dataf['brch_other'] == 'None', 1, 0)

dataf['tree_dbh'].loc[(dataf['tree_dbh'] < 20)] = 0
dataf['tree_dbh'].loc[(dataf['tree_dbh'] >= 20)] = 1
dataf['tree_dbh'].loc[(dataf['tree_dbh'] < 5)] = -1
dataf = dataf.astype(str)

X =
dataf.drop(['borough', 'block_id', 'spc_latin', 'spc_common', 'address', 'postc
ode', 'zip_city', 'community
board', 'cncldist', 'st_assem', 'st_senate', 'nta', 'nta_name', 'health',
'boro_ct', 'latitude', 'longitude', 'x_sp', 'y_sp' ], axis = 1)

stand_X = pd.DataFrame(preprocessing.scale(X), columns = X.columns)
stand_X = pd.isnull(stand_X)

```

```

from sklearn.decomposition import PCA, KernelPCA
for i in [1,2,3,4]:
    pca = PCA(n_components=i)
    pca.fit(stand_X)
    print( pca.explained_variance_ratio_)

# Достаточно 2 признака для объяснения 90% дисперсии
# Первый признак вносит наибольший вклад в первую компоненту
len(stand_X)

pca = PCA(n_components=2)
pca.fit(stand_X)
X3 = pca.transform(stand_X)
X3
df3 = pd.DataFrame(data=X3, columns=["PC1", "PC2"])

df3['Y'] = 0
df3 = df3.astype(float)
for i in range (5848):
    df3.at[i, 'Y'] = data['tree_dbh'].iloc[i]
df3.head()

plt.scatter('PC1', 'PC2', c='Y', cmap = 'Set1', data=df3)

pca = PCA(n_components=2)
pca.fit(stand_X)
X3 = pca.transform(stand_X)
X3
df3 = pd.DataFrame(data=X3, columns=["PC1", "PC2"])
df3['Y'] = 0

dataf['tree_dbh'] = dataf['tree_dbh'].astype('float')
df3 = df3.astype(float)
for i in range (5848):
    df3.at[i, 'Y'] = dataf['tree_dbh'].iloc[i]
df3.head()

from sklearn import tree
from sklearn.metrics import f1_score
df3 = df3.astype(str)
target = pd.DataFrame(df3['Y'])
train = pd.DataFrame(df3.drop(['Y'], axis = 1))
X_train, X_test, y_train, y_test = train_test_split(train, target,
test_size = 0.3, train_size = 0.7, random_state = 42)
X_train.shape
clf = tree.DecisionTreeClassifier(max_depth=10, random_state=42)
clf.fit(X_train, y_train)
clf.score(X_test, y_test)

y_pred = clf.predict(X_test)
f1_score(y_test, y_pred, average='macro')

from sklearn import tree
from sklearn.metrics import f1_score
df3 = df3.astype(str)
target = pd.DataFrame(df3['Y'])
train = pd.DataFrame(df3.drop(['Y'], axis = 1))
X_train, X_test, y_train, y_test = train_test_split(train, target,
test_size = 0.3, train_size = 0.7, random_state = 42)
X_train.shape
clf = tree.DecisionTreeClassifier(max_depth=10, random_state=42)

```



```
clf.fit(X_train, y_train)
clf.score(X_test, y_test) # точность классификатора, когда он работает с
# несбалансированными данными

y_pred = clf.predict(X_test)
f1_score(y_test, y_pred, average='macro')
```