



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА - Российский технологический университет»
РТУ МИРЭА

Институт искусственного интеллекта
Кафедра высшей математики

ОТЧЁТ ПО НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ
(получение первичных навыков научно-исследовательской работы)

Тема НИР: Выявление закономерностей в частотах и силе пожаров (kaggle.com)
приказ университета о направлении на практику
от «9» февраля 2022 г. № 1038 - С

Отчет представлен к
рассмотрению:
Студентка группы КМБО-
01-21

Баринова А.А.
(расшифровка подписи)
«11» мая 2022г.

Отчет утвержден.
Допущена к защите:

Руководитель практики от
кафедры

Петрусевич Д.А.
(расшифровка подписи)
«1» июня 2022г.

Москва 2022



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА - Российский технологический университет»
РТУ МИРЭА

ЗАДАНИЕ

на **НАУЧНО-ИССЛЕДОВАТЕЛЬСКУЮ РАБОТУ**

(получение первичных навыков научно-исследовательской работы)

Студентке 1 курса учебной группы КМБО-01-21 института искусственного интеллекта
Бариновой Александре Александровне

(фамилия, имя и отчество)

Место и время НИР: Институт искусственного интеллекта, кафедра высшей математики

Время НИР: с «09» февраля 2022 по «31» мая 2022

Должность на НИР: практикант

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

2. **СОДЕРЖАНИЕ НИР:**

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:** Выявление закономерностей в частотах и силе пожаров (kaggle.com)

4. **ОГРАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ:** выделить статистические характеристики пожаров; найти зависимости между частотой или силой пожаров с другими параметрами в наборе данных; выделить аномальное увеличение или ослабление силы или частоты пожаров в определенное время

Заведующий кафедрой
высшей математики

«09» февраля 2022 г.

СОГЛАСОВАНО

Руководитель практики от кафедры:

«09» февраля 2022 г.

Задание получила:

«09» февраля 2022 г.

Ю.И.Худак

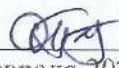

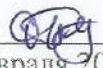
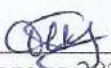
(подпись)

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

(подпись)

(Баринова А.А.)
(фамилия и инициалы)

ИНСТРУКТАЖ ПРОВЕДЕН:

Вид мероприятия	ФИО ответственного, подпись, дата	ФИО студентки, подпись, дата
Охрана труда	Петрусеви́ч Д.А.  «09» февраля 2022 г.	Баринова А.А.  «09» февраля 2022 г.
Техника безопасности	Петрусеви́ч Д.А.  «09» февраля 2022 г.	Баринова А.А.  «09» февраля 2022 г.
Пожарная безопасность	Петрусеви́ч Д.А.  «09» февраля 2022 г.	Баринова А.А.  «09» февраля 2022 г.
Правила внутреннего распорядка	Петрусеви́ч Д.А.  «09» февраля 2022 г.	Баринова А.А.  «09» февраля 2022 г.



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА - Российский технологический университет»
РТУ МИРЭА

**РАБОЧИЙ ГРАФИК ПРОВЕДЕНИЯ
НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЫ**

(получение первичных навыков научно-исследовательской работы)


студентки Бариновой А.А. 1 курса группы КМБО-01-21 очной формы обучения,
обучающейся по направлению подготовки 01.03.02 «Прикладная математика и
информатика»,
профиль «Математическое моделирование и вычислительная математика»

Неделя	Сроки выполнения	Этап	Отметка о выполнении
1	09.02.2022	Выбор темы НИР. Пройти инструктаж по технике безопасности	✓
1	09.02.2022	Вводная установочная лекция	✓
2	14.02.2022	Построение и оценка парной регрессии с помощью языка R	✓
3	21.02.2022	Построение и оценка множественной регрессии с помощью языка R	✓
4	28.02.2022	Построение доверительных интервалов. Обработка факторных переменных. Мультиколлинеарность	✓
5	07.03.2022	Гетероскедастичность	✓
6	14.03.2022	Классификация	✓
7	21.03.2022	Кластеризация. Предобработка данных	✓
8	28.03.2022	Метод главных компонент	✓
9	04.04.2022	Ансамбли классификаторов.	✓

		Беггинг. Бустинг	
16	29.05.2022	Представление отчётных материалов по НИР и их защита. Передача обобщённых материалов на кафедру для архивного хранения	✓
		Зачётная аттестация	✓

Согласовано:

Заведующий кафедрой




/ ФИО / Худак Ю.И.

Руководитель практики от кафедры



/ ФИО / Петрусевич Д.А.

Обучающаяся



/ ФИО / Барина А.А.

Оглавление

Задача 1	2
Задача 2.1	5
Задача 2.2	8
Задача 3	10
Задача 4	16
Задача 5	18
Задача 6	23
Заключение	28
Список использованной литературы	29
Приложения	30
Приложение 1	31
Приложение 2	33
Приложение 3	38
Приложение 4	40
Приложение 5	52
Приложение 6	55
Приложение 7	59

Задача 1

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: Swiss.

Объясняемая переменная: Agriculture.

Регрессоры: Examination, Catholic.

1. Оценить среднее значение, дисперсию и СКО объясняемой и объясняющих переменных.

Таблица 1. Величины среднего значения, дисперсии и среднеквадратического отклонения параметров Agriculture, Examination, Catholic.

Показатели\Параметры	Agriculture	Examination	Catholic
Среднее значение	50.66	16.49	41.14
Дисперсия	515.8	63.65	1739.3
СКО	22.7	7.98	41.7

Величина среднеквадратического отклонения Agriculture составляет 22.7.

Величина среднеквадратического отклонения Examination составляет 7.98.

Величина среднеквадратического отклонения Catholic составляет 41.7.

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

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

Параметр\Характеристики	Estimate	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	82.8869	5.640	14.694	< 2e-16	***
Examination	-1.9544	0.308	-6.334	9.95e-08	***

Построим зависимость параметра Agriculture от параметра Examination ($\text{Agriculture} = a + b * \text{Examination}$). Находим коэффициенты a , b , которые представлены в таблице 1, и получаем линейную регрессию вида $\text{Agriculture} = 82.9 - 1.9 * \text{Examination}$. Дополнительно построим график регрессии с помощью команды `plot`.

Таблица 3. Характеристики модели зависимости параметра Agriculture от параметра Catholic в наборе данных Swiss

Параметр\Характеристики	Estimate	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	41.67276	4.3329	9.618	1.74e-12	***
Examination	0.21842	0.0743	2.937	0.0052	**

Построим зависимость параметра Agriculture от параметра Catholic ($\text{Agriculture} = a + b * \text{Catholic}$). Находим коэффициенты a , b , которые представлены в таблице 2, и получаем линейную регрессию вида $\text{Agriculture} = 41.7 + 0.2 * \text{Catholic}$. Дополнительно построим график регрессии с помощью команды `plot`.

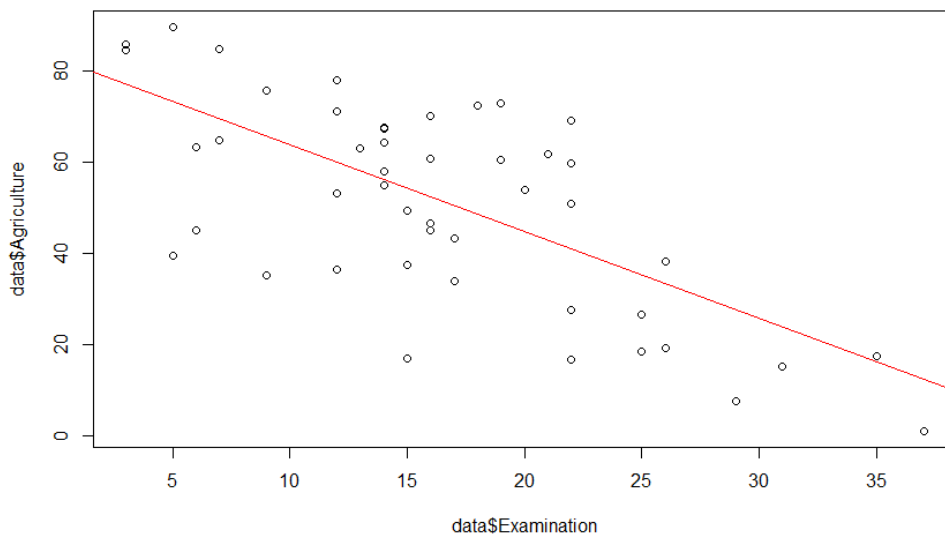


Рисунок 1. Красная линия - построенная линейная зависимость параметра Agriculture от Examination с помощью команды `plot`.

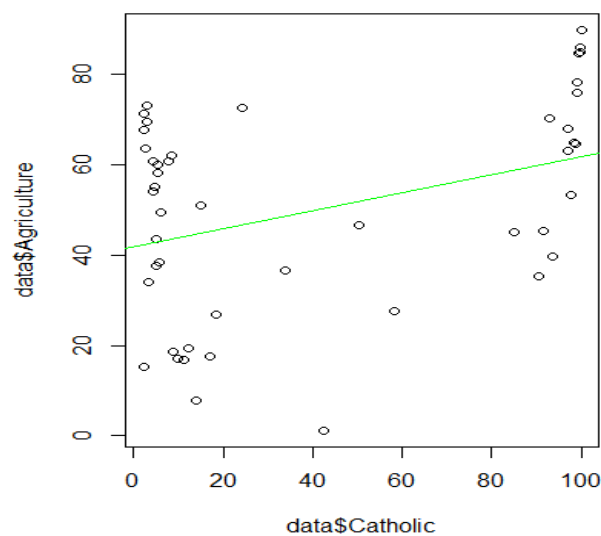


Рисунок 2. Зеленая линия - построенная линейная зависимость параметра Agriculture от Catholic с помощью команды `plot`.

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

Проверим линейную регрессию $\text{Agriculture} \sim \text{Examination}$. R^2 в этой модели около 46%, поэтому мы заключаем, что параметр Agriculture относительно хорошо (значение $R^2 < 80\%$) описывается параметром Examination

Проверим линейную регрессию $\text{Agriculture} \sim \text{Catholic}$. R^2 в этой модели около 14%, поэтому мы заключаем, что параметр Agriculture (значение $R^2 < 20\%$) практически не описывается параметром Examination.

4. Оцените, есть ли взаимосвязь между объясняемой переменной и объясняющей переменной (по значению р-статистики, «количеству звездочек» у регрессора в модели).

Проверим линейную регрессию $\text{Agriculture} \sim \text{Examination}$. Взаимосвязь между Agriculture и Examination существует: $p \text{ value} = 9.95e-08$ - низкое значение, а значит наша гипотеза о зависимости Agriculture от Examination верна, количество звездочек - 3, что говорит о сильной взаимосвязи (данные взяты из таблицы 1).

Проверим линейную регрессию $\text{Agriculture} \sim \text{Catholic}$. Взаимосвязь между Agriculture и Catholic существует: $p \text{ value} = 0.0052$ - низкое значение, а значит наша гипотеза о зависимости Agriculture от Examination верна, количество звездочек - 2, что также говорит о существовании положительной взаимосвязи (данные взяты из таблицы 3).

Вывод

Модель $\text{Agriculture} \sim \text{Examination}$ имеет значение р-статистики = $9.95e-08$, количество звездочек 3, а также $R^2 = 0.45$. Низкое значение р-статистики и большое количество звездочек говорит о сильной взаимосвязи параметров, значение среднеквадратического отклонения имеет средний показатель. В целом модель хороша.

Модель $\text{Agriculture} \sim \text{Catholic}$ имеет значение р-статистики = 0.0052, количество звездочек 2, а также $R^2 = 0.14$. Низкое значение р статистики и большое количество звездочек говорит о сильной взаимосвязи параметров, но значение среднеквадратического отклонения мало. Модель обладает низким значением СКО, поэтому делаем вывод, что модель плохая.

Код решения задачи приведен в приложении 1.

Задача 2.1

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: Swiss.

Объясняемая переменная: Infant.Mortality.

Регрессоры: Fertility, Catholic, Agriculture.

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 невысокий). В случае, если R^2 большой, один из таких столбцов можно исключить из рассмотрения.

Проверим регрессии $Fertility \sim Catholic$, $Fertility \sim Agriculture$, $Catholic \sim Agriculture$. СКО в этих моделях меньше 20%, поэтому мы заключаем, что параметры Fertility, Catholic, Agriculture не зависят друг от друга линейно, и все параметры могут быть использованы при построении математических моделей.

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.

Построим линейную модель зависимой переменной Infant.Mortality от регрессоров Fertility, Agriculture, Catholic по методу наименьших квадратов.

Таблица 4. Характеристики модели зависимости параметра Infant.Mortality от параметров Fertility, Agriculture, Catholic.

Параметр\Характеристики	Estimate	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	13.565009	2.362940	5.741	8.7e-07	***
Fertility	0.112074	0.036105	3.104	0.00337	**
Agriculture	-0.032335	0.019207	-1.683	0.09953	.
Catholic	0.003754	0.011045	0.340	0.73560	

Характеристики модели зависимости детской смертности Infant.Mortality от регрессоров Fertility, Agriculture, Catholic приведены в таблице 3.

В результате построения регрессии получаем $\text{adjusted } R^2 = 17\%$, что значит наша модель практически не зависит линейно от этих параметров. Также значение р-статистики параметров Agriculture (0.09953) и Catholic (0.73560) большое, что может испортить нашу модель, поэтому можно попробовать исключить данные регрессоры. Значение р-статистики регрессора Fertility равно 0.00337 – небольшое значение, это свидетельствует о значимости данного параметра в модели.

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

Введем в модель логарифмы регрессоров и сравним модели, чтобы выбрать наилучшую. При решении этой задачи были проверены модели, в которых были добавлены все возможные комбинации параметров: $\log(\text{Fertility})$, $\log(\text{Agriculture})$, $\log(\text{Catholic})$.

Таблица 5. Характеристики модели зависимости параметра *Infant.Mortality* от параметров *Fertility*, *Agriculture*, *Catholic*, $\log(\text{Catholic})$.

Параметр\Характеристики	Estimate	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	9.75181	3.55508	2.743	0.00891	**
Fertility	0.12459	0.03675	3.390	0.00153	**
Agriculture	-0.01615	0.02213	-0.730	0.46972	
Catholic	-0.04832	0.03820	-1.265	0.21285	
$\log(\text{Catholic})$	1.45742	1.45742	1.45742	1.45742	

Таблица 6. Показатели VIF переменных *Fertility*, *Agriculture*, *Catholic*, $\log(\text{Catholic})$

Fertility	Agriculture	Catholic	$\log(\text{Catholic})$
1.411139	1.691716	16.993861	14.107484

Наилучшей моделью оказалась: $\text{Infant.Mortality} \sim \text{Fertility} + \text{Agriculture} + \text{Catholic} + \log(\text{Catholic})$. Значение R^2 данной модели равно 0.19, что выше, чем у других проверяемых моделей. Значение VIF (показатели VIF регрессоров приведены в таблице 5) регрессоров *Catholic* и $\log(\text{Catholic})$ этой модели достаточно высокое (больше 18) — это свидетельствует о сильной взаимосвязи между этими параметрами, однако значение VIF остальных регрессоров меньше 2, т. е. между ними не существует линейной зависимости.

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

Введем в модель всевозможные произведения пар регрессоров, в том числе квадраты регрессоров.

При решении этой задачи были проверены модели, в которых были добавлены все возможные комбинации параметров: $\text{Fertility} * \text{Agriculture}$, $\text{Fertility} * \text{Catholic}$, $\text{Agriculture} * \text{Catholic}$, Fertility^2 , Agriculture^2 , Catholic^2 .

Таблица 7. Характеристики модели зависимости параметра Infant.Mortality от параметров Fertility, Agriculture, Catholic, Agriculture * Catholic.

Параметр\Характеристики	Estimate	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	10.6769197	2.3889572	4.469	5.84e-05	***
Fertility	0.1149070	0.0332847	3.452	0.00128	**
Agriculture	0.0203553	0.0251936	0.808	0.42367	
Catholic	0.0876259	0.0302994	2.892	0.00604	**
Agriculture * Catholic	-0.0013968	0.0004753	-2.939	0.00533	**

Таблица 8. Показатели VIF переменных Fertility, Agriculture, Catholic, log(Catholic)

Fertility	Agriculture	Catholic	Agriculture * Catholic
1.331396	2.521354	12.297423	16.310805

Наилучшей моделью оказалась: $Infant.Mortality \sim Fertility + Agriculture + Catholic + Agriculture * Catholic$. Значение R^2 данной модели равно 0.30 (см. Таблицу 6), что выше, чем у других проверяемых моделей, а соответственно взаимосвязь между объясняемой переменной и регрессорами больше. Значение VIF регрессоров Catholic и Agriculture * Catholic этой модели достаточно высокое (больше 12) (данные взяты из таблицы 8) — это говорит о сильной взаимосвязи между параметрами, поэтому уберем из модели Catholic и получим модель $Infant.Mortality \sim Fertility + Agriculture + Agriculture * Catholic$.

Вывод

В результате решения задачи наилучшей моделью оказалась $Infant.Mortality \sim Fertility + Agriculture + Catholic + Agriculture * Catholic$, эта модель оказалась так же и лучше модели $Infant.Mortality \sim Fertility + Agriculture + Catholic + \log(Catholic)$. Показатель среднеквадратичного отклонения ($R^2 = 30\%$) данной модели выше показателей СКО остальных моделей, что говорит о более сильной взаимосвязи между объясняемой переменной и объясняющими. Также значения VIF регрессоров данной модели маленькие, кроме регрессоров Catholic и Agriculture * Catholic, которые зависимы между собой, поэтому можно попробовать убрать один из параметров, чтобы получить модель лучше.

Код решения задачи приведен в приложении 2.

Задача 2.2

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: Swiss.

Объясняемая переменная: Infant.Mortality.

Регрессоры: Fertility, Catholic, Agriculture.

Модель: $\text{Infant.Mortality} \sim \text{Fertility} + \text{Agriculture} + \text{Catholic} + \text{Agriculture} * \text{Catholic}$.

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

Оценим доверительные интервалы для всех коэффициентов в модели, $p = 95\%$.

Число степеней свободы в модели: 42. Для такого числа степеней свободы и $p = 95\%$ значение t-критерия Стьюдента: $t = 2.02$.

Доверительный интервал имеет вид: $[b - t * \text{std_e}, b + t * \text{std_e}]$.

Здесь b - вычисленное в модели значение коэффициента, t - значение t-критерия Стьюдента. std_e - стандартная ошибка коэффициента в модели. Значения стандартных ошибок коэффициентов возьмем из таблицы 5.

Найдем нижнюю границу доверительного интервала для коэффициента Fertility, она равна 0.04831026, и верхнюю границу, она равна 0.1815037. Получаем доверительный интервал вида $[0.04831026, 0.1815037]$.

Найдем нижнюю границу доверительного интервала для коэффициента Agriculture, она равна -0.03009676, и верхнюю границу, она равна 0.07080733. Получаем доверительный интервал вида $[-0.03009676, 0.07080733]$.

Найдем нижнюю границу доверительного интервала для коэффициента Catholic, она равна 0.02708346, и верхнюю границу, она равна 0.1481684. Получаем доверительный интервал вида $[0.02708346, 0.1481684]$.

Найдем нижнюю границу доверительного интервала для коэффициента Catholic * Agriculture, она равна 0.08661687, и верхнюю границу, она равна 0.08863496. Получаем доверительный интервал вида $[0.08661687, 0.08863496]$.

2. Сделать вывод об отвержении или невозможности отвергнуть статистическую гипотезу о том, что коэффициент равен 0.

Проверим гипотезу о том, что коэффициент перед Fertility равен 0. Доверительный интервал коэффициента Fertility: $[0.04831026, 0.1815037]$, 0 не входит в границы диапазона, а следовательно, можно сделать вывод о том, что связь между Infant.Mortality и Fertility существует.

Проверим гипотезу о том, что коэффициент перед Agriculture равен 0. Доверительный интервал коэффициента Agriculture: $[-0.03009676, 0.07080733]$, 0 входит в границы диапазона, а следовательно, можно сделать вывод о том, что связи между Infant.Mortality и Agriculture не существует.

Проверим гипотезу о том, что коэффициент перед Catholic равен 0. Доверительный интервал коэффициента Catholic: $[0.02708346, 0.1481684]$, 0 не входит в границы диапазона, а следовательно, можно сделать вывод о том, что связь между Infant.Mortality и Catholic существует.

Проверим гипотезу о том, что коэффициент перед Catholic * Agriculture равен 0. Доверительный интервал коэффициента Catholic * Agriculture: [0.08661687, 0.08863496], 0 не входит в границы диапазона, а следовательно, можно сделать вывод о том, что связь между Infant.Mortality и Catholic * Agriculture существует.

3. Оценить доверительный интервал для одного прогноза ($p = 95\%$)

Оценим доверительный интервал для одного прогноза ($p = 95\%$, значения регрессоров: Fertility = 80, Agriculture = 10, Catholic = 13).

Воспользуемся функцией predict() для выбранного набора значений регрессоров и найдем прогноз для них. Получаем значение прогноза для введенных данных равным 21.03059 в доверительном интервале [18.9545; 23.10667].

Вывод

В результате решения задачи были рассчитаны доверительные интервалы с вероятностью $p = 95\%$ для коэффициентов перед параметрами Fertility, Agriculture, Catholic, Catholic * Agriculture. Доверительный интервал для Fertility: [0.04831026, 0.1815037], Agriculture: [-0.03009676, 0.07080733], Catholic: [0.02708346, 0.1481684], Catholic * Agriculture: [0.08661687, 0.08863496]. Для полученных данных была отвержена гипотеза о том, что коэффициент равен 0 перед параметрами Fertility, Catholic, Catholic * Agriculture, для параметра Agriculture эта гипотеза была подтверждена. Также была найдена положительная связь между параметром Infant.Mortality и параметрами Fertility, Agriculture и отрицательная связь между параметром Infant.Mortality и параметром Catholic * Agriculture.

Оценили доверительный интервал для прогноза ($p = 95\%$, значения регрессоров: Fertility = 80, Agriculture = 10, Catholic = 13) и получили значение прогноза для введенных данных равным 21.03059 в доверительном интервале [18.9545; 23.10667].

Код решения задачи приведен в приложении 3.

Задача 3

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: r19i_os26c.sav.

Объясняемая переменная: salary.

Регрессоры: age, sex, higher_educ, status1, dur, wed1, wed2, wed3, wed4, language, work_sat, subord.

1. Построить линейную регрессию зарплаты на все параметры, которые были выделены из данных мониторинга, оценить коэффициент вздутия дисперсии VIF.

Построим линейную регрессию. Для решения данной задачи были выделены следующие столбцы: *oj13.2* (среднемесячная зарплата после вычета налогов), *o_age* (количество полных лет), *oh5* (пол респондента), *o_educ* (образование), *status* (тип населенного пункта), *oj6.2* (длительность рабочей недели в среднем), *o_marst* (семейное положение), *oj24*

(являются владельцами или совладельцами предприятия респондента, организации иностранные фирмы или иностранные частные лица), *oj260* (владение иностранным языком), *oj1.1.1* (удовлетворенность работой), *oj6* (наличие подчиненных на работе).

Таблица 9. Характеристики модели зависимости параметра зарплаты salary от параметров age, sex, higher_educ, status1, dur, wed1, wed2, wed3, wed4, language, work_sat, subord.

Параметр\Характеристики	Estimate	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	-0.70749	0.13560	-5.217	1.87e-07	***
age	-0.04390	0.01305	-3.363	0.000774	***
sex1	0.39803	0.02455	16.212	< 2e-16	***
higher_educ	0.33711	0.02750	12.260	< 2e-16	***
status1	0.26803	0.02599	10.313	< 2e-16	***
dur	0.12591	0.01191	10.571	< 2e-16	***
wed1	-0.15387	0.13293	-1.158	0.247100	
wed2	-0.09648	0.13136	-0.734	0.462687	
wed3	-0.07955	0.13600	-0.585	0.558648	
wed4	-0.15138	0.13968	-1.084	0.278484	
language	0.26063	0.03044	8.563	< 2e-16	***

work_sat	0.22329	0.03587	6.224	5.15e-10	***
subord	0.42119	0.02879	14.630	< 2e-16	***

Таблица 10. Показатели коэффициентов вздутия дисперсии VIF переменных age, sex,

age	sex	higher_educ	status1	dur	wed1	wed2	wed3	wed4	language	work_sat	subord
1.288	1.124	1.223932	1.038	1.072	26.2	31.7	11.2	7.71	1.200316	1.010205	1.0785

higher_educ, status1, dur, wed1, wed2, wed3, wed4, language, work_sat, subord

Введем факторные и дамми переменные в нашу модель: age, sex, higher_educ, status1, dur, wed1, wed2, wed3, wed4, language, work_sat, subord.

Построим модель: $salary \sim age + sex + higher_educ + status1 + dur + wed1 + wed2 + wed3 + wed4 + language + work_sat + subord$. Характеристики показателей переменных модели приведены в таблице 8, показатели VIF регрессоров приведены в таблице 9.

Для решения задачи найдем модель с наилучшими показателями, для этого исключим из модели регрессоры wed2, затем wed3, wed4, wed1, поскольку между ними существует линейная зависимость (значение VIF переменных большое). Полученная модель не испортилась в результате преобразований исходной модели, потому что значение СКО практически не изменилось, т. е. модель хорошая, $R^2 = 0.18$.

2. Ввести в модель логарифмы и функции степени (от 0.1 до 2 с шагом 0.1).

Введем в модель логарифмы и степени регрессоров и сравним модели, чтобы выбрать наилучшую.

При решении этой задачи были проверены модели с введенными функциями на параметры: age в степени от 0.1 до 2 (с шагом 0.1), $\log(subord)$, $\log(status1)$, subord в степени от 0.1 до 2 (с шагом 0.1), $\log(dur)$, $dur^{0.5}$, dur^2 , dur^4 , $work_sat^2$, $\log(work_sat)$.

3. Выделить наилучшие модели из построенных: по значимости параметров, включённых в зависимости, и по объяснённой с помощью построенных зависимостей разбросу adjusted R^2 .

Таблица 11. Показатели коэффициентов вздутия дисперсии VIF переменных $age^{1.3}$, sex, higher_educ, status1, dur, language, work_sat, $subord^{1.8}$.

I($age^{1.3}$)	sex	higher_educ	status1	dur	language	work_sat	I($subord^{1.8}$)
1.029796	1.043617	1.239023	1.029741	1.073259	1.150228	1.013700	1.084786

Таблица 12. Характеристики модели зависимости параметра зарплаты salary от параметров age^{1.3}, sex1, higher_educ, status1, dur, language, work_sat, subord^{1.8}.

Параметр\Характеристики	Estimate	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	-0.68933	0.05448	-12.652	< 2e-16	***
I(age ^{1.3})	-0.19658	0.02167	-9.073	< 2e-16	***
sex1	0.33086	0.03187	10.382	< 2e-16	***
higher_educ	0.34692	0.03822	9.078	< 2e-16	***
status1	0.30546	0.03409	8.961	< 2e-16	***
dur	0.10787	0.01594	6.767	1.58e-11	***
language	0.36302	0.04714	7.701	1.82e-14	***
work_sat	0.23855	0.04803	4.967	7.19e-07	***
I(subord ^{1.8})	0.38353	0.03781	10.142	< 2e-16	***

В результате решения задачи наилучшей моделью оказалась $salary \sim I(age^{1.3}) + sex + higher_educ + status1 + dur + language + work_sat + I(subord^{1.8})$. У этой модели самый высокий $R^2 = 0.20$ среди проверяемых, значение VIF каждой переменной не превосходит 1.24, у каждого параметра 3 звездочки, также значение p – статистики мало (не превосходит 1.58e-11). Показатели коэффициентов вздутия дисперсии VIF в данной модели приведены в таблице 10, характеристики выбранной модели приведены в таблице 11.

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

Рассмотрим коэффициенты перед параметрами в модели $salary \sim I(age^{1.3}) + sex + higher_educ + status1 + dur + language + work_sat + I(subord^{1.8})$. У параметра age отрицательный множитель, это значит, что между ним и объясняемой переменной существует обратная взаимосвязь: чем моложе респондент, тем выше у него зарплата. У переменных sex, higher_educ, status1, dur, language, work_sat, subord положительный множитель, следовательно между ними и объясняемой переменной существует прямая взаимосвязь. Следовательно, делаем вывод, что у мужчин, респондентов с высшим образованием, респондентов из города или областного центра, респондентов с большей продолжительностью рабочей недели, владеющих иностранным языком респондентов,

удовлетворенных работой респондентов, имеющих подчиненных респондентов зарплата выше.

5. Оценить регрессии для подмножества индивидов, указанных в варианте:
городские жители, не состоявшие в браке; разведенные женщины, без высшего образования.

Таблица 13. Характеристики модели зависимости параметра зарплаты salary от параметров age^{1.3}, sex, higher_educ, dur, language, work_sat, subord^{1.8}.

Параметр\Характеристики	Estimate	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	-0.222628	0.209300	-1.064	0.291	
I(age ^{1.3})	-0.192564	0.103478	-1.861	0.067	.
sex	0.127597	0.202559	0.630	0.531	
higher_educ	0.008076	0.142809	0.057	0.955	
dur	0.015157	0.071823	0.211	0.833	
language	0.261403	0.168526	1.551	0.125	
work_sat	0.188176	0.184672	1.019	0.312	
I(subord ^{1.8})	0.269074	0.179738	1.497	0.139	

Найдем подмножество городских жителей, не состоявших в браке. Для этого сначала с помощью команды select выделим подмножество мужчин в используемом наборе данных, затем в этом подмножестве выделим подмножество городских жителей. Построим модель $salary \sim I(age^{1.3}) + sex + higher_educ + dur + language + work_sat + I(subord^{1.8})$ для полученного множества. Значение СКО модели $R^2 = 0.02$, все переменные без звездочек, значение p – статистики не меньше 0.067, поэтому делаем вывод, что модель плохая.

Таблица 14. Характеристики модели зависимости параметра зарплаты salary от параметров age^{1.3}, dur, status1, language, work_sat, subord^{1.8}.

Параметр\Характеристики	Estimate	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	-0.73004	0.13498	-5.409	1.91e-07	***
I(age ^{1.3})	-0.12154	0.06290	-1.932	0.05483	.
dur	0.09100	0.05054	1.801	0.07336	.
status1	0.18757	0.09876	1.899	0.05907	.
language	0.31621	0.17024	1.857	0.06481	.
work_sat	0.37601	0.11968	3.142	0.00195	**
I(subord ^{1.8})	0.35797	0.11529	3.105	0.00220	**

Найдем подмножество разведенных женщин без высшего образования. Для этого сначала с помощью команды select выделим подмножество женщин в используемом наборе данных, затем в этом подмножестве найдем подмножество разведенных женщин и в конце – подмножество разведенных женщин без высшего образования. Построим модель $salary \sim I(age^{1.3}) + dur + status1 + language + work_sat + I(subord^{1.8})$ для полученного множества. Значение СКО модели $R^2 = 13\%$, переменные *work_sat*, *I(subord^{1.8})* имеют 2 звездочки, остальные параметры не имеют звездочек, значение p – статистики у многих параметров большое (больше 0.05483), поэтому делаем вывод, что модель плохая.

Вывод

В результате решения задачи исходная модель $salary \sim age + sex + higher_educ + status1 + dur + wed1 + wed2 + wed3 + wed4 + language + work_sat + subord$ оказалась недостаточно хорошей: в модели присутствовали переменные, между которыми существовала линейная зависимость, о чем говорит большое значение показателя VIF (не меньше 7.71) у переменных *wed1*, *wed2*, *wed3*, *wed4*, поэтому данные параметры исключили из модели.

Далее в модель добавили функции степени и нашли наилучшую модель $salary \sim I(age^{1.3}) + sex + higher_educ + status1 + dur + language + work_sat + I(subord^{1.8})$, значение R^2 этой модели равно 20 %, у каждого параметра 3 звездочки (каждый параметр важен для построения модели) и значение p-статистики не превосходит 1.58e-11.

Используя данные из таблицы 10, выяснили, что у параметра *age* отрицательный множитель, а это говорит об обратной взаимосвязи между ним и объясняемой переменной, и что у переменных *sex*, *higher_educ*, *status1*, *dur*, *language*, *work_sat*, *subord* положительный множитель, следовательно между ними и объясняемой переменной существует прямая взаимосвязь.

Оценили регрессии по двум подмножествам: подмножество городских жителей, не состоявших в браке, подмножество разведенных женщин без высшего образования. Модель, построенная по подмножеству городских жителей, не состоявших в браке, оказалась плохой: $R^2 = 2\%$, все переменные без звездочек, значение p – статистики не меньше 0.067. Модель, построенная по подмножеству разведенных женщин без высшего образования, плохая: $R^2 = 13\%$, многие переменные без звездочек, значение p – статистики у многих параметров больше 0.05483.

Код решения задачи приведен в приложении 4.

Задача 4

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: Video game sales <https://www.kaggle.com/gregorut/videogamesales>.

Тип классификатора: LogisticRegression (логистическая регрессия).

Классификация по столбцу: Platform (PS2 – класс 0, остальные уровни – класс 1).

1. Обработать набор данных, подготовив его к решению задачи классификации. Выделить целевой признак и удалить его из данных, на основе которых будет обучаться классификатор. Разделить набор данных на тестовую и обучающую выборку. Построить классификатор. Оценить точность построенного классификатора с помощью метрик precision, recall и F1 на тестовой выборке.

Для решения задачи необходимы библиотеки *numpy*, *pandas*, *sklearn.linear_model*, *sklearn.model_selection*, *sklearn.tree*, *sklearn.ensemble*. Загрузим данные, указанные в условии задачи. Для построения логистической регрессии были выбраны столбцы *Genre*, *Year*, *NA_Sales*, *EU_Sales*, *JP_Sales*, *Other_Sales*. В датасете есть объекты с пропущенными значениями, удалим их. Классификацию строим по столбцу *Platform*, PS2 - класс 0, остальные платформы - класс 1. *Genre* – столбец категориального признака, для дальнейшего построения классификатора, нужно преобразовать этот столбец: присвоим каждому различному значению столбца цифру от 0 до 6.

Уберем из данных, на основе которых будем строить классификацию, столбец *Platform*, разделим набор данных на тестовую и обучающую выборку и построим логистическую регрессию с помощью функции *LogisticRegression*. Полученный классификатор оценим с помощью метрик: accuracy, F1, precision, recall. Значение accuracy равно 0.87, но так как в данной задаче классы неравны, эта метрика не так информативна; значение f1 равно 0.93; значение precision равно 0.87; значение recall равно 1. В итоге, по полученным значениям метрик можно сделать вывод, что построенный классификатор достаточно точный.

Таблица 15. Значение метрик логистической регрессии, построенной по данным Video game sales.

accuracy	0.87
F1	0.93
precision	0.87
recall	1

2. Построить 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?

Данные для построения нового классификатора те, что использовались в первом пункте этой задачи. Построим классификатор типа Случайный Лес и оценим с помощью метрик accuracy, F1, precision, recall. Значение accuracy равно 0.93, значение f1 равно 0.96, значение precision равно 0.96, значение recall равно 0.96. Сравнивая метрики классификаторов типа логистической регрессии и случайного леса, можно сделать вывод, что классификатор случайного леса работает лучше.

Таблица 16. Значение метрик Случайного Леса, построенного по данным Video game sales.

accuracy	0.87
F1	0.93
precision	0.87
recall	1

Вывод

В ходе решения задачи удалось построить классификатор типа логистической регрессии, для этого выбрали столбцы *Genre*, *Year*, *NA_Sales*, *EU_Sales*, *JP_Sales*, *Other_Sales*. *Genre* – столбец категориального признака (этот столбец был преобразован), остальные столбцы имеют числовые значения. Построили классификацию по столбцу *Platform*, PS2 - класс 0, остальные платформы - класс 1. Для данного классификатора значения метрик: accuracy равен 0.87, f1 равен 0.93, precision равен 0.87, recall равен 1. Также построили классификатор типа случайный лес, для которого значения метрик: accuracy равен 0.93, f1 равен 0.96, precision равен 0.96, recall равен 0.96. Сделали вывод, что второй классификатор оказался лучше.

Код решения задачи приведен в приложении 5.

Задача 5

Необходимо загрузить данные из указанного набора и ответить на вопросы, указанные в задании.

Набор данных: Forest Fires Data Set <https://www.kaggle.com/datasets/elikplim/forest-fires-data-set>

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

Загрузим данные и с помощью команды `shape` узнаем количество объектов и признаков. Находим, что в датасете 517 объектов и 13 признаков.

Описание признаков:

- X - координата в пространстве по оси X в границах карты парка Монтесиньо: от 1 до 9
- Y - координата в пространстве по оси Y в границах карты парка Монтесиньо: от 2 до 9
- month - месяц года: от 'jan' до 'dec'
- day - день недели: от 'mon' до 'sun'
- FFMC - The Fine Fuel Moisture Code - индекс влажности измельченного топлива (индекс системы пожаров и погоды): от 18.7 до 96.20
- DMC - The Duff Moisture Code - индекс влажности штыба (мелкий уголь) (индекс системы пожаров и погоды): от 1.1 до 291.3
- DC - The Drought Code - индекс засушливости (индекс системы пожаров и погоды): от 7.9 до 860.6
- ISI - The Initial Spread Index - индекс мгновенного распространения (индекс системы пожаров и погоды): от 0.0 до 56.10
- temp - температура в градусах Цельсия: от 2.2 до 33.30
- RH - относительная влажность в %: от 15.0 до 100
- wind - скорость ветра в км/ч: от 0.40 до 9.40
- rain - количество дождевых осадков в мм/м2 : от 0.0 до 6.4
- area - выжженные территории в га: от 0.00 до 1090.84

2. Сколько категориальных признаков, какие?

Посмотрим какого типа столбцы в датасете с помощью команды `info`. Всего 2 категориальных признака: *day*, *month*, для дальнейшей работы обработаем эти столбцы, закодируем значения этих признаков с помощью класса *LabelEncoder* из модуля *sklearn.preprocessing*.

3. Столбец с максимальным количеством уникальных значений категориального признака?

Для того, чтобы проверить количество уникальных значений, воспользуемся командой `value_counts`, и в результате получим, что *month* - столбец с максимальным количеством уникальных значений (12 уник. значений), у столбца *day* - 7 уникальных значений.

4. Есть ли бинарные признаки?

В датасете бинарных признаков нет.

5. Какие числовые признаки?

Числовые признаки: X , Y , $FFMC$, DMC , DC , ISI , $temp$, RH , $wind$, $rain$, $area$.

6. Есть ли пропуски?

В данных нет объектов с пропущенными значениями.

7. Сколько объектов с пропусками?

Таких объектов нет.

8. Столбец с максимальным количеством пропусков?

Такого столбца нет.

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

Для дальнейшего анализа нормализуем признаки с помощью стандартного отклонения и построим график распределения признаков DMC , $FFMC$. Видим, что значения лежат примерно в одном диапазоне, за исключением нескольких, т. е. аномальные значения есть.

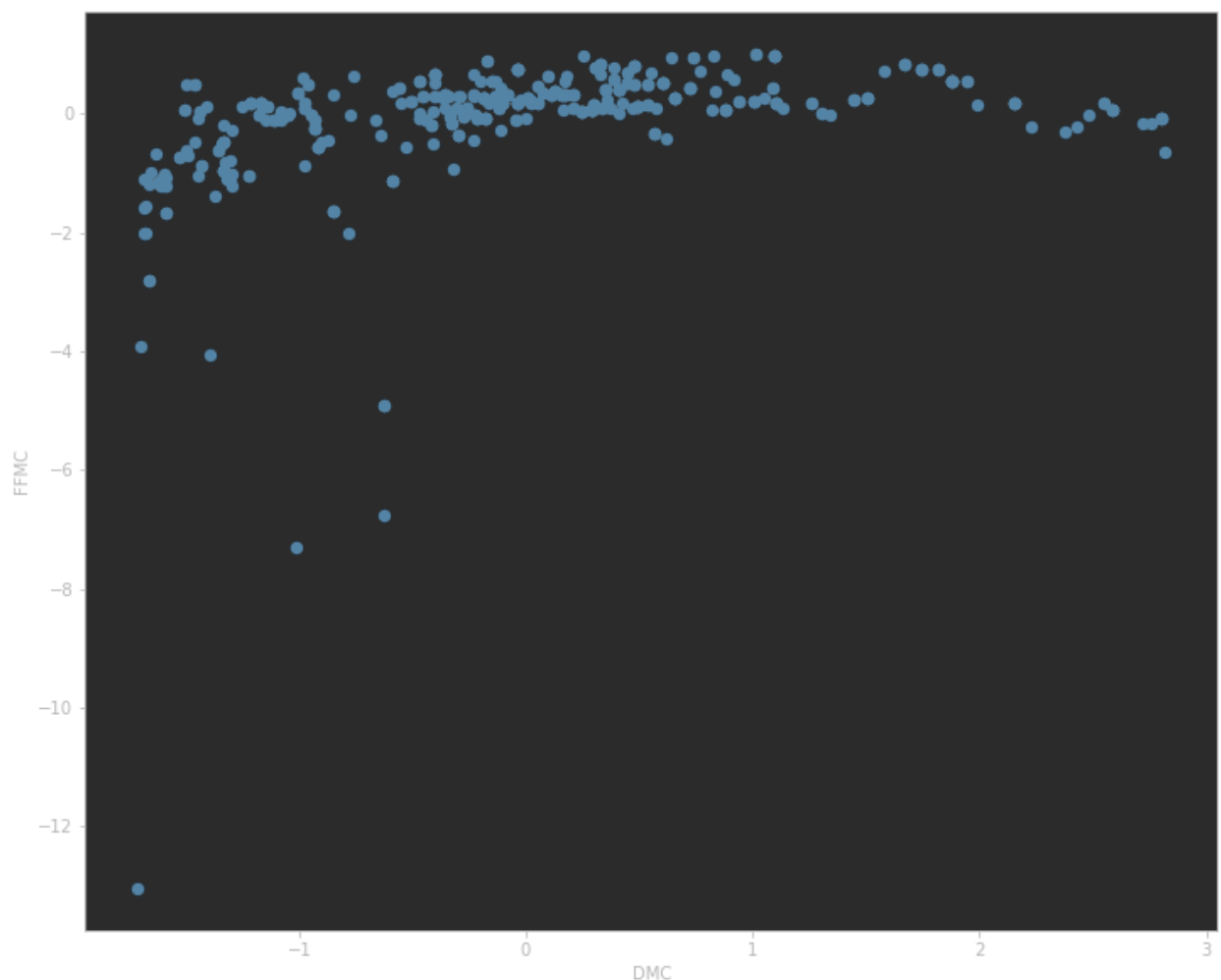


Рисунок 3. График распределения признаков DMC и $FFMC$: ось X – DMC , ось Y – $FFMC$.

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

Посмотрим на статистический анализ столбцов с помощью команды `describe` и заметим, что `RH` - столбец с максимальным средним значением ($3.362881e-16$) среди всех столбцов, которые нормировали.

11. Столбец с целевым признаком?

Для задачи выделим как целевой признак столбец `area`, также из исходных данных уберем признаки `X`, `Y`, `day`, `area` для обучающей и тренировочной выборки.

12. Сколько объектов попадает в тренировочную выборку при использовании `train_test_split` с параметрами `test_size = 0.3`, `random_state = 42`?

В тренировочную выборку попадет 361 объект.

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

Для того, чтобы проследить между какими признаками наблюдается линейная зависимость, воспользуемся командой `corr()` и визуализируем ее результат. Увидим, что наблюдается отрицательная линейная зависимость между `RH` и `temp`, `RH` и `FFMC`; наблюдается положительная линейная зависимость между `X` и `Y`, `DC` и `DMC`, `FFMC` и `ISI`, `DMC`

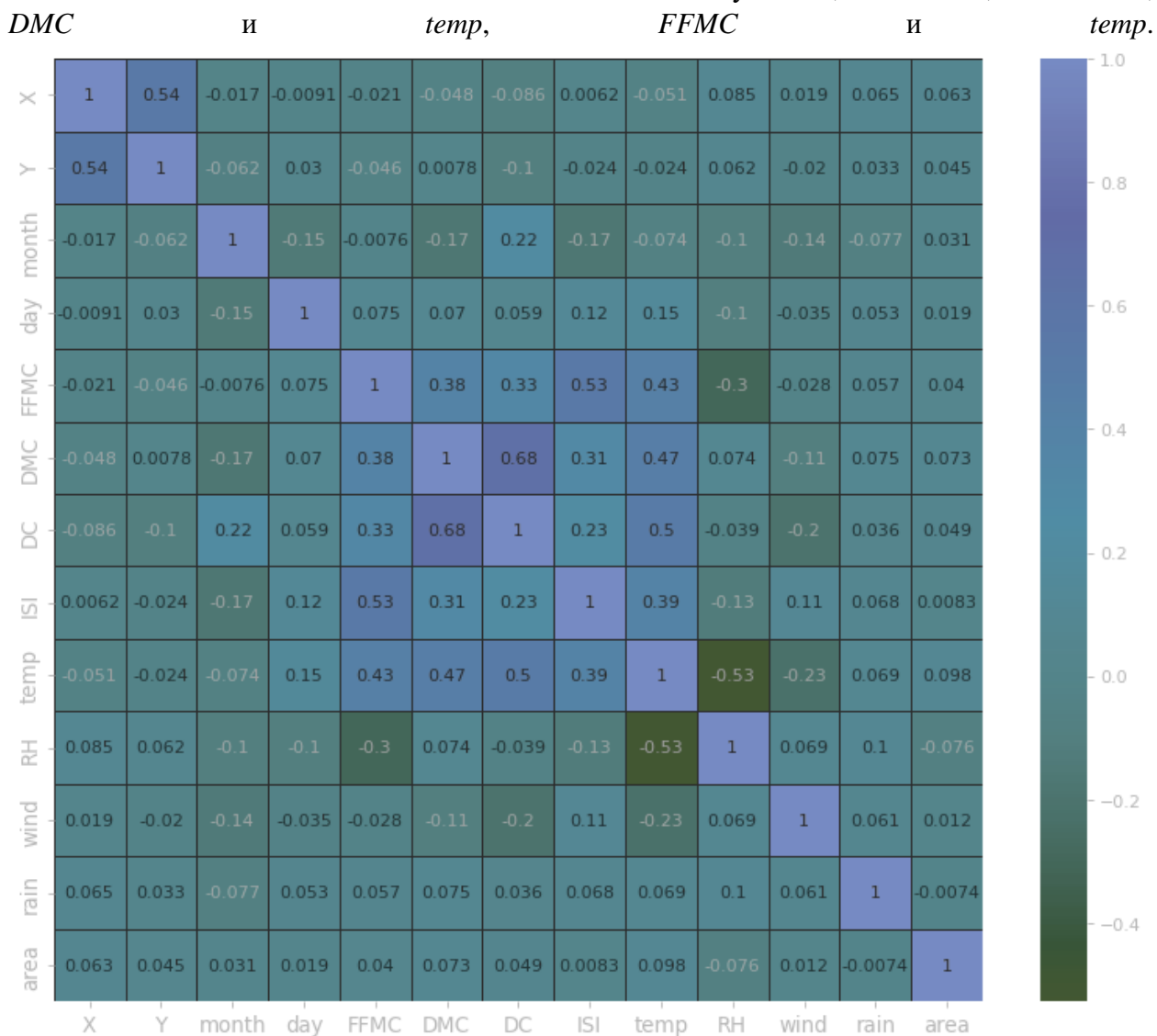


Рисунок 4. Визуализация корреляции между параметрами выбранных данных.

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

Для объяснения 90% дисперсии с помощью метода главных компонент достаточно четыре компонента. Первая компонента объясняет 70,06% дисперсии, вторая компонента - 10.57%, третья компонента - 5.34%, четвертая компонента - 4.69%, поэтому делаем вывод, что необходимо всего 4 компоненты.

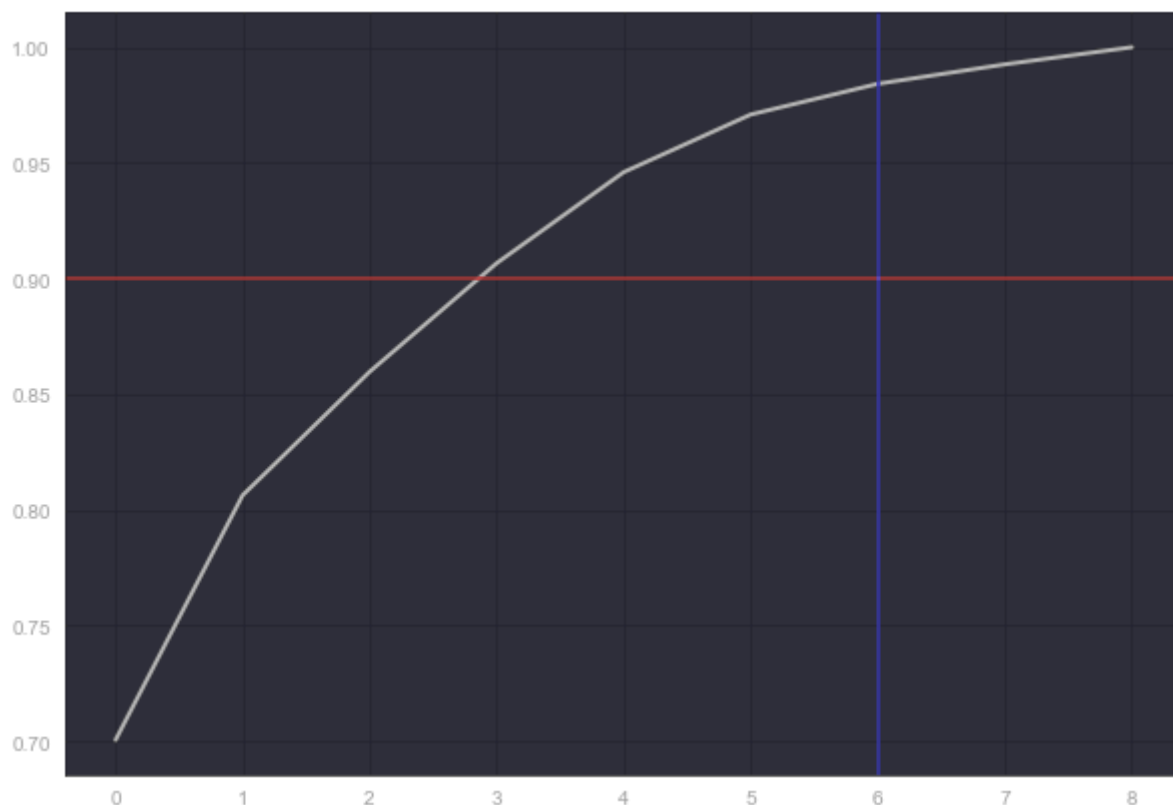


Рисунок 5. График зависимости доли объясненной дисперсии от числа главных компонент.

15. Какой признак вносит наибольший вклад в первую компоненту?

В первую компоненту наибольший вклад вносит признак *month*, поскольку абсолютное значение коэффициента перед ним является максимальным среди всех коэффициентов первой компоненты, оно составляет -0.995.

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

Удалось визуализировать данные с помощью алгоритма tSNE, и на полученном графике получилось выделить 3 кластера. Они представляют собой области, между которыми существует большое расстояние.

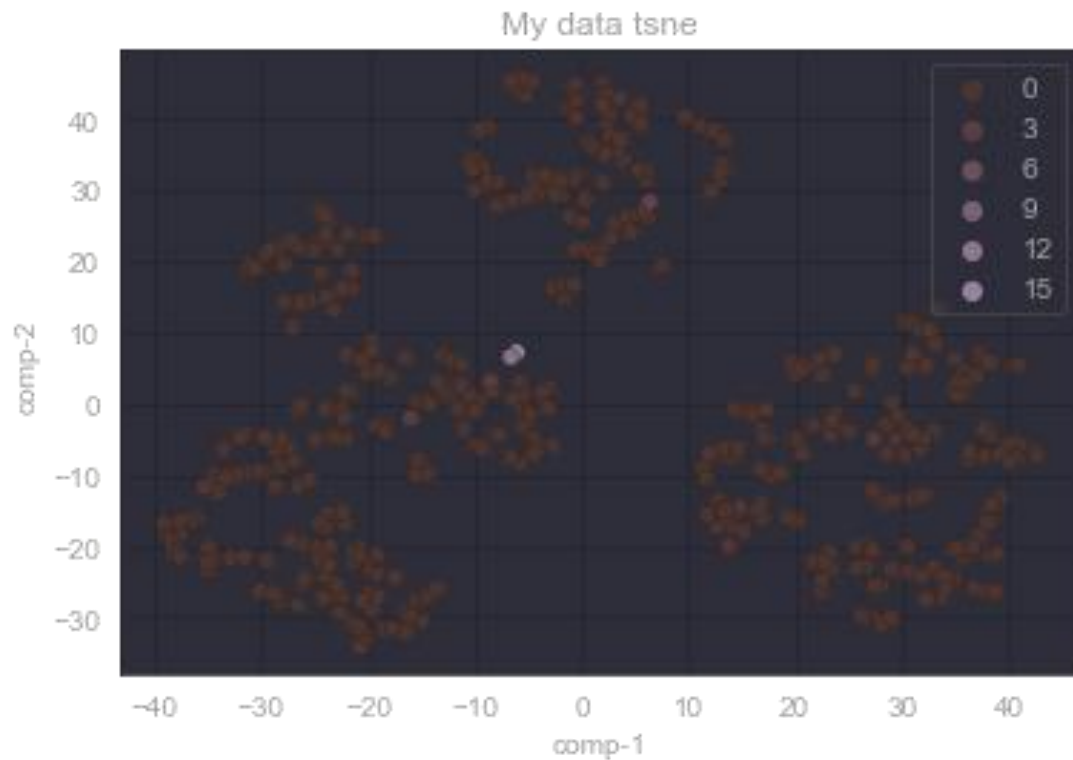


Рисунок 6. Двумерное представление данных Forest Fires Data Set с помощью алгоритма t-SNE.

Вывод

Был обработан набор данных Forest Fires Data Set, содержащий 517 объектов и 13 признаков, среди которых 2 категориальных признака и 11 числовых. Категориальные признаки были закодированы, а числовые признаки нормализованы с помощью среднеквадратичного отклонения. Целевым признаком был выбран столбец *area*, тренировочная выборка представляет собой исходные данные без признаков *X*, *Y*, *day*, *area*, она состоит из 361 объекта. Нашли отрицательную линейную зависимость между *RH* и *temp*, *RH* и *FFMC*; нашли положительную линейную зависимость между *X* и *Y*, *DC* и *DMC*, *FFMC* и *ISI*, *DMC* и *temp*, *FFMC* и *temp*.. Выяснили, что для объяснения 90% дисперсии после применения метода PCA достаточно четырех компонент, и что в первую компоненту наибольший вклад вносит признак *month*, значение коэффициента перед которым составляет -0.995.

Код решения задачи приведен в приложении 6.

Задача 6

Необходимо загрузить данные из указанного набора и провести анализ датасета.

Набор данных: Forest Fires Data Set <https://www.kaggle.com/elikplim/forest-fires-data-set>

В 5 практической работе был проведен первичный анализ данных, продолжим работу с датасетом, основываясь на результатах, полученных в предыдущей задаче.

1. Построить линейные регрессии с объясняемой переменной *area*, ввести функции степени и логарифма, найти наилучшую модель по значениям R^2 и p -статистики.

Найдем зависимость между площадью сгоревших лесов и остальными параметрами, для этого построим линейные регрессии, введем в них функции степени и логарифма и найдем наилучшую модель, оценив ее значение СКО и значения p value и VIF переменных. С помощью функции *variance_inflation_factor* получим значения VIF регрессоров (данные указаны в таблице 16). Видим, что значение VIF параметра Y большое, поэтому уберем его из модели. В результате значение VIF переменных X , *day*, *month*, *DMC*, *FFMC*, *DC*, *ISI*, *temp*, *RH*, *wind*, *rain* не превосходит 3.194 (данные указаны в таблице 17), т. е. между ними не наблюдается линейная зависимость. Далее построим линейные регрессии, где *area* – объясняемая переменная, а X , *day*, *month*, *DMC*, *FFMC*, *DC*, *ISI*, *temp*, *RH*, *wind*, *rain* – объясняющие переменные.

Таблица 17. Показатели коэффициентов вздутия дисперсии VIF переменных X , Y , *day*, *month*, *DMC*, *FFMC*, *DC*, *ISI*, *temp*, *RH*, *wind*, *rain*.

X	Y	day	month	DMC	FFMC	DC	ISI	temp	RH	wind	rain
7.279	10.81	2.768	3.355	2.698	1.7115	2.749	1.608	2.765	1.959	1.148	1.049

Таблица 18. Показатели коэффициентов вздутия дисперсии VIF переменных X , *day*, *month*, *DMC*, *FFMC*, *DC*, *ISI*, *temp*, *RH*, *wind*, *rain*.

X	day	month	DMC	FFMC	DC	ISI	temp	RH	wind	rain
3.194	2.3018	2.8103	2.5750	1.7060	2.5964	1.6076	2.7551	1.9554	1.1479	1.0484

При решении этой задачи были проверены модели, в которые были добавлены параметры: ISI^2 , ISI^3 , $temp^2$, DMC^2 , $\log_2 DMC$, $rain^{0.4}$, DC^2 , $\lg DC$, $DC^{1.4}$, $month^2$, RH^2 , $\lg RH$, $RH^{0.3}$, $FFMC^{0.5}$, $\lg FFMC$, $FFMC^2$, $wind^2$, $wind^{0.2}$, $\log_{20} wind$.

Наилучшая модель (по значению СКО): $area \sim X + \log(wind, 20) + rain^{0.4} + DC^{1.4} + ISI + day + month^2 + FFMC + DMC + temp + RH$. Значение VIF параметров не превосходит 3.2, p value у многих параметров больше 0.37 (данные взяты из таблицы 18), поэтому уберем регрессоры с большим значением p -характеристики, не изменив значительно $R^2 = 2.8\%$ модели.

В результате получаем модель: $area \sim X + \log(wind, 20) + rain^{0.4} + DC^{1.4} + ISI + month^2 + DMC + temp$. Значение СКО модели равно 2.7% и разница между предыдущей и настоящей моделью 0.1% - незначительная. Исключив параметры с большим значением p -статистики, получили линейную регрессию, где p -value регрессоров не превосходит 0.486. Данные взяты из таблицы 19.

Таблица 19. Характеристики модели зависимости параметра площади *area* от параметров *X*, $\log(\text{wind}, 20)$, $\text{rain}^{0.4}$, $\text{DC}^{1.4}$, *ISI*, *day*, month^2 , *FFMC*, *DMC*, *temp*, *RH*.

Параметр\Характеристики	coef	Std. Error	t value	P(> t)
const	-0.2208	0.141	-1.569	0.117
X	0.0324	0.019	1.689	0.092
month	0.0017	0.001	1.559	0.120
day	0.0098	0.023	0.419	0.676
FFMC	-0.0107	0.057	-0.187	0.852
DMC	0.1279	0.074	1.727	0.085
DC	-0.0992	0.082	-1.211	0.226
ISI	-0.0424	0.055	-0.769	0.442
temp	0.0984	0.072	1.365	0.173
RH	-0.0391	0.062	-0.627	0.531
wind	0.1292	0.091	1.425	0.155
rain	-0.1112	0.205	-0.543	0.587

Таблица 20. Характеристики модели зависимости параметра площади *area* от параметров *X*, $\log(\text{wind}, 20)$, $\text{rain}^{0.4}$, $\text{DC}^{1.4}$, *ISI*, month^2 , *DMC*, *temp*.

Параметр\Характеристики	coef	Std. Error	t value	P(> t)
const	-0.1912	0.121	-1.586	0.113
X	0.0315	0.019	1.647	0.100
month	0.0017	0.001	1.634	0.103
DMC	0.1139	0.070	1.618	0.106
DC	-0.1050	0.080	-1.305	0.193
ISI	-0.0464	0.049	-0.941	0.347
temp	0.1285	0.055	2.340	0.020
wind	0.1320	0.090	1.465	0.143
rain	-0.1383	0.198	-0.698	0.486

2. Построить классификацию, где классы – районы парка Монтесиньо (1 район – 0 класс и т. д.).

Для начала работы выделим районы парка Монтесиньо в зависимости от значений координат *x*, *y*. Всего выделим три района, отметим, что все данные, представленные в датасете, точно попадают в полученные районы.

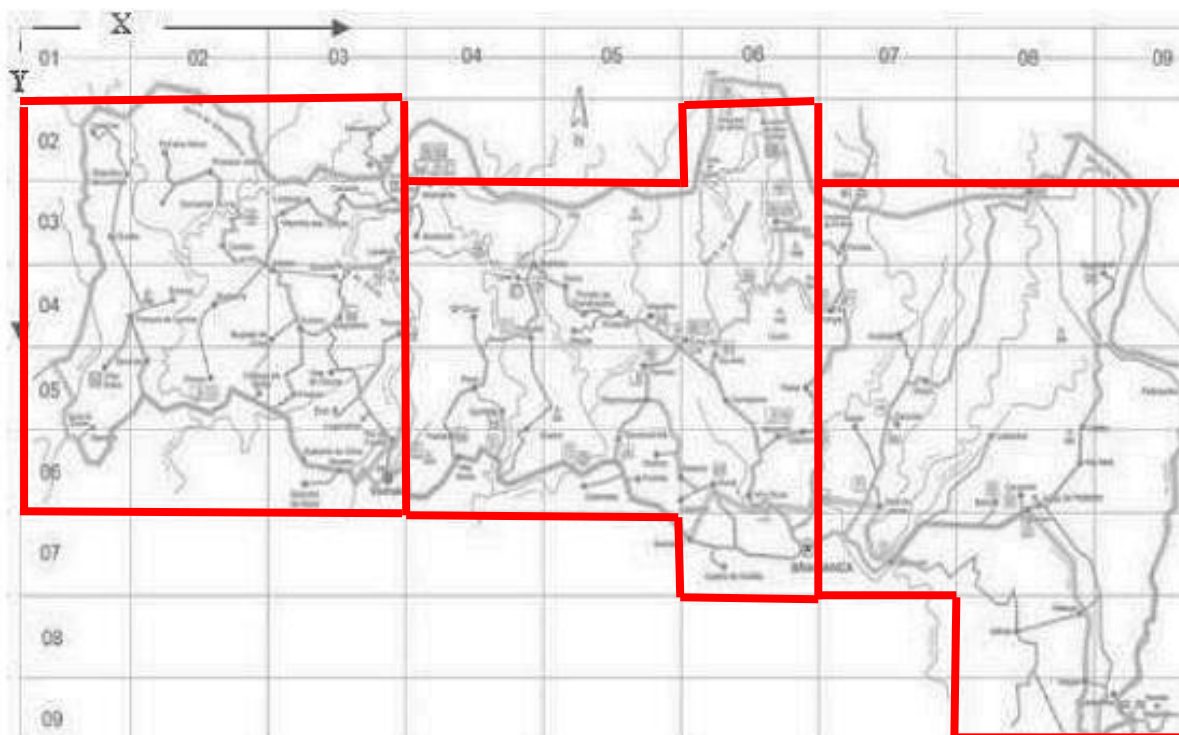


Рисунок 7. Карта парка Монтесиньо с выделенными районами.

Построим классификатор типа градиентный бустинг, в котором признаки – *month, day, FFMC, DMC, DC, ISI, temp, RH, wind, rain, area*. Выделим тренировочную и тестовую выборку с параметрами *test_size = 0.3, random_state = 42*. Оценим качество классификатора с помощью команды *score*, получив значение равное -0.07204. Дополнительно решим задачу классификации, используя другой тип классификатора и сравним оба.

Построим классификатор типа *XGBClassifier* с признаками – *month, day, FFMC, DMC, DC, ISI, temp, RH, wind, rain, area*. Используем ту же тренировочную и тестовую выборку, что и для предыдущего классификатора. Команда *score* для данного классификатора возвращает значение равное 0.41026, что больше, чем значение для классификатора типа градиентный бустинг, т. е. полученный классификатор работает точнее.

Построим регрессию типа *XGBRegressor*, предскажем значения для тестовой выборки и оценим разницу между предсказанным и фактическим значением с помощью функции *MSE*. Получим, что значение корня от квадрата ошибки составляет 0.877297. Значение маленькое, значит регрессия достаточно хороша. Более того, данная регрессия точнее, чем линейная регрессия, построенная с помощью метода наименьших квадратов, поскольку значение *RMSE* для нее равно 2.04.

3. Сравнить значения признаков для регионов с наибольшей и наименьшей площадью сгоревших лесов.

Найдем регионы с наибольшей и наименьшей площадью сгоревших лесов, для этого просуммируем значения столбца *area* для каждого из трех регионов. Получим, что район, где сгорела наибольшая часть леса – второй, и район, где сгорела наименьшая часть леса – нулевой.

Посмотрим на статистический анализ столбцов нулевого и второго региона с помощью команды *describe* и сравним значения признаков, посчитав разницу средних значений

столбцов. Можно заметить, что наибольшая абсолютная разница – разница значения в столбцах *RH*, *rain*.

Таблица 21. Разница соответствующих средних значений признаков *FFMC*, *DMC*, *DC*, *ISI*, *temp*, *RH*, *wind*, *rain* для нулевого и второго региона.

Параметр:	FFMC	DMC	DC	ISI	temp	RH	wind	rain
Разница:	-0.0019	-0.0155	0.0629	-0.1127	0.0418	-0.2687	-0.0340	-0.2372

Вывод

В результате решения задачи были рассмотрены линейные регрессии с различными комбинациями параметров и функций, примененными над параметрами. Была найдена наилучшая линейная регрессия $area \sim X + \log(wind, 20) + rain^{0.4} + DC^{1.4} + ISI + month^2 + DMC + temp$, со значением СКО равным 2.7 % и р-значением параметров, не превосходящим 0.486. Сравнили ее с регрессией типа *XGBRegressor*, которая оказалась лучше по значению метрики *RMSE*: значение корня от квадрата ошибки составляет 0.877297, в то время как у линейной регрессии значение равно 2.04. Также построили классификаторы типа градиентный бустинг и *XGBClassifier*, значение *score* второго классификатора оказалось выше (оно равно 0.41026, а значение *score* для градиентного бустинга составляет -0.07204). Сравнили средние значения признаков *FFMC*, *DMC*, *DC*, *ISI*, *temp*, *RH*, *wind*, *rain* для нулевого и второго региона, и выяснили, что наибольшая абсолютная разница в столбцах *RH* (0.2687), *rain* (0.2372).

Код решения задачи приведен в приложении 7.

Заключение

1. Модель *Agriculture ~ Examination* имеет маленькое значение p статистики (p value = $9.95e-08$) и большое количество звездочек (***), значение СКО имеет средний показатель ($R^2 = 45\%$). В целом модель хороша.
Модель *Agriculture ~ Catholic* имеет большое количество звездочек (**) и небольшое значение p статистики (p value = 0.0052), но значение СКО очень мало ($R^2 = 14\%$). В целом модель плоха.
2. В результате решения задачи 2.1 наилучшей моделью оказалась *Infant.Mortality ~ Fertility + Agriculture + Catholic + Agriculture * Catholic*. Показатель среднеквадратичного отклонения $R^2 = 30\%$, что говорит о взаимосвязи между объясняемой переменной и объясняющими. Также значения VIF регрессоров данной модели маленькие (меньше 3), кроме регрессоров Catholic и Agriculture * Catholic (больше 18).
3. В результате решения задачи 2.2 были рассчитаны для модели *Infant.Mortality ~ Fertility + Agriculture + Catholic + Agriculture * Catholic* доверительные интервалы для Fertility: [0.04831026, 0.1815037], Agriculture: [-0.03009676, 0.07080733], Catholic: [0.02708346, 0.1481684], Catholic * Agriculture: [0.08661687, 0.08863496]. Также оценили доверительный интервал для прогноза и получили его значение для введенных данных равным 21.03059 в доверительном интервале [18.9545; 23.10667].
4. В результате решения задачи 3 была найдена наилучшая модель *salary ~ I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat + I(subord^1.8)*, значение R^2 этой модели равно 20%, у каждого параметра 3 звездочки и значение p -статистики не превосходит $1.58e-11$. В модели у параметра age отрицательный множитель, у переменных sex, higher_educ, status1, dur, language, work_sat, subord положительный множитель.
5. В результате решения задачи 4 был построен классификатор типа логистической регрессии по столбцу *Platform*, для которого значения метрик: ассигасу равно 0.87, f1 равно 0.93, precision равно 0.87, recall равно 1. Также был построен классификатор типа случайный лес, для него значения метрик: : ассигасу равно 0.93, f1 равно 0.96, precision равно 0.96, recall равно 0.96. Классификатор типа случайный лес оказался лучше логистической регрессии.
6. В результате решения задачи 5 был обработан набор данных Forest Fires Data Set, содержащий 517 объектов и 13 признаков, среди которых 2 категориальных признака и 11 числовых. Целевым признаком был выбран столбец *area*, тренировочная выборка представляет собой исходные данные без признаков *X*, *Y*, *day*, *area*, она состоит из 361 объекта. Нашли линейную зависимость между параметрами *RH* и *temp*, *X* и *Y*, *DC* и *DMC*, *RH* и *FFMC*, *FFMC* и *ISI*, *DMC* и *temp*, *FFMC* и *temp*.
7. В результате решения задачи 6 была найдена наилучшая линейная регрессия: $area \sim X + \log(wind, 20) + rain^{0.4} + DC^{1.4} + ISI + month^2 + DMC + temp$, со значением R^2 равным 2.7 % и RMSE равным 2.04. Была построена регрессия типа XGBRegressor, со значением RMSE равным 0.877297. Удалось построить классификаторы типа градиентный бустинг и XGBClassifier, со значением *score* равным -0.07204 и 0.41026.

Список использованной литературы

1. Кормен Т. Алгоритмы: Построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест - МОСКВА: МЦНМО, 1999–955 с.
2. Introduction to Econometrics with R/Christoph Hanck, Martin Arnold, Alexander Gerber, Martin Schmelzer – Essen, Germany: University of Duisburg-Essen, 2021.
3. Доугерти, Кристофер. Введение в эконометрику/ Кристофер Доугерти - Москва: ИНФРА-М, 2009–465.
4. Магнус Я. Р. Эконометрика. Начальный курс / Я. Р. Магнус, П. К. Катышев, А. А. Пересецкий - Москва: Изд-во: «ДЕЛО», 2004–576 с.
5. Айвазян, С. А. Основы эконометрики / С. А. Айвазян, В. С. Мхитарян – Москва: Изд. Объединение «ЮНИТИ», 1998–1005 с.

Приложения

Приложение 1.

```
library("lmtest")
```

```
data = swiss
```

```
help(swiss)
```

```
mean(data$Agriculture) # 50.65957
```

```
mean(data$Examination) # 16.48936
```

```
mean(data$Catholic) # 41.14383
```

```
var(data$Agriculture) # 515.7994
```

```
var(data$Examination) # 63.64662
```

```
var(data$Catholic) # 1739.295 - большой разброс (значения сильно отличаются от среднего)
```

```
sd(data$Agriculture) # 22.71122
```

```
sd(data$Examination) # 7.977883
```

```
sd(data$Catholic) # 41.70485
```

```
modele_ex = lm(Agriculture ~ Examination, data)
```

```
modele_ex
```

```
summary(modele_ex)
```

```
plot(data$Agriculture ~ data$Examination) + abline(a = 82.9, b = -1.9, col = "red")
```

```
# зависимость между agriculture и examination прослеживается (p value = 9.95e-08,  
# количество звездочек 3, что отлично), полученная модель хорошая, но строгой линейно  
# зависимости нет ( $R^2 = 0.45$ )
```

```
modele_cath = lm(Agriculture ~ Catholic, data)
```

```
modele_cath
```

```
summary(modele_cath)
```

```
plot(data$Agriculture ~ data$Catholic) + abline(a = 41.7, b = 0.2, col = "green")
```

```
# зависимости между agriculture и catholic не наблюдается ( $R^2 = 0.1422$ , p value = 0.0052),  
# хотя количество звездочек 2, что хорошо
```

```
model_cath_ex = lm(Agriculture ~ Catholic + Examination, data)
model_cath_ex
summary(model_cath_ex)
```

Приложение 2.

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

data = swiss

# ----- part 1 -----
# Значение R = 0.19
model1 = lm(Fertility ~ Catholic, data)
summary(model1)

# Значение R = 0.11
model2 = lm(Fertility ~ Agriculture, data)
summary(model2)

# Значение R2 = 0.14
model3 = lm(Catholic ~ Agriculture, data)
summary(model3)

# значение R2 у всех возможных пар регрессоров < 0.20, это значит, что особой линейной
зависимости между ними нет

# ----- part 2 -----
# Значение R2 = 0.17; p-значение Fertility = 0.00337 (значение маленькое)
# p-значение Agriculture = 0.09953 (значение достаточно большое), p-значение Catholic =
0.73560 (значение большое, можно попробовать исключить данный регрессор)
model = lm(Infant.Mortality ~ Fertility + Agriculture + Catholic, data)
summary(model)
plot(data$Infant.Mortality ~ data$Fertility + data$Catholic + data$Agriculture)
vif(model)

# итог:
# 1. маленькое значение R2 показывает, что объясняемая переменная слабо выражается
линейно через данные регрессоры
```

2. в основном маленькие p-значения показывают, что найденные коэффициенты для построения линейной регрессии могут сильно отличаться

----- part 3 -----

Значение $R^2 = 0.15$, значение vif у Fertility и log(Fertility) больше 68 (существует очень сильная линейная зависимость), значение vif остальных регрессоров меньше 3

```
model_ln1 = lm(Infant.Mortality ~ Fertility + log(Fertility) + Agriculture + Catholic, data)
```

```
summary(model_ln1)
```

```
vif(model_ln1)
```

Значение $R^2 = 0.14$, значение vif у Fertility и log(Fertility) больше 88 (существует очень сильная линейная зависимость)

значение vif остальных регрессоров меньше 8 (приемлемо)

```
model_ln2 = lm(Infant.Mortality ~ Fertility + Agriculture + log(Fertility) + log(Agriculture) + Catholic, data)
```

```
summary(model_ln2)
```

```
vif(model_ln2)
```

Значение $R^2 = 0.16$, значение vif у Fertility и log(Fertility) больше 88 (существует очень сильная линейная зависимость)

значение vif Catholic и log(Catholic) больше 14, т.е. существует сильная линейная зависимость, у остальных регрессоров значение vif меньше 8

```
model_ln3 = lm(Infant.Mortality ~ Fertility + Agriculture + Catholic + log(Fertility) + log(Agriculture) + log(Catholic), data)
```

```
summary(model_ln3)
```

```
vif(model_ln3)
```

Значение $R^2 = 0.17$, значение vif у Fertility и log(Fertility) больше 68 (существует очень сильная линейная зависимость), значение Agriculture < 2,

значение vif остальных регрессоров 14 (существует сильная линейная зависимость)

```
model_ln4 = lm(Infant.Mortality ~ Fertility + Catholic + log(Fertility) + Agriculture + log(Catholic), data)
```

```
summary(model_ln4)
```

```
vif(model_ln4)
```

Значение $R^2 = 0.16$, значение vif всех регрессоров не больше 6

```
model_ln5 = lm(Infant.Mortality ~ Fertility + log(Agriculture) + Agriculture + Catholic, data)
```

```
summary(model_ln5)
```

```
vif(model_ln5)
```

Значение $R^2 = 0.18$, значение vif Catholic и log(Catholic) больше 14, т.е. существует сильная линейная зависимость, у остальных регрессоров значение vif меньше 7

```
model_ln6 = lm(Infant.Mortality ~ Fertility + Agriculture + Catholic + log(Agriculture) + log(Catholic), data)
```

```
summary(model_ln6)
```

```
vif(model_ln6)
```

Значение $R^2 = 0.19$, значение vif Catholic и log(Catholic) больше 14, т.е. существует сильная линейная зависимость, у остальных регрессоров значение vif меньше 2

лучшая модель

```
model_ln7 = lm(Infant.Mortality ~ Fertility + Agriculture + Catholic + log(Catholic), data)
```

```
summary(model_ln7)
```

```
vif(model_ln7)
```

----- part 4 -----

Значение $R^2 = 0.16$

значение vif Agriculture и Fertility * Agriculture больше 34, т.е. существует очень сильная линейная зависимость, у остальных регрессоров значение vif меньше 5

```
model_m1 = lm(Infant.Mortality ~ Fertility + Agriculture + Catholic + Fertility * Agriculture, data)
```

```
summary(model_m1)
```

```
vif(model_m1)
```

Значение $R^2 = 0.19$

значение vif Catholic и Fertility * Catholic больше 88, т.е. существует очень сильная линейная зависимость, у остальных регрессоров значение vif меньше 6

```
model_m2 = lm(Infant.Mortality ~ Fertility + Agriculture + Catholic + Fertility * Catholic, data)
```

```
summary(model_m2)
```

```
vif(model_m2)
```

```
# Значение  $R^2 = 0.30$ 
```

```
# значение vif Catholic и Agriculture * Catholic больше 12, т.е. существует сильная  
линейная зависимость, у остальных регрессоров значение vif меньше 3
```

```
model_m3 = lm(Infant.Mortality ~ Fertility + Agriculture + Catholic + Agriculture * Catholic,  
data)
```

```
summary(model_m3)
```

```
vif(model_m3)
```

```
# лучшая модель!!!
```

```
# Значение  $R^2 = 0.16$ , значение vif у Fertility и  $I(\text{Fertility}^2)$  больше 81 (существует очень  
сильная линейная зависимость)
```

```
# у остальных регрессоров значение vif меньше 3
```

```
model_I1 = lm(Infant.Mortality ~ Fertility + I(Fertility^2) + Agriculture + Catholic, data)
```

```
summary(model_I1)
```

```
vif(model_I1)
```

```
# Значение  $R^2 = 0.20$ , значение vif у Fertility и  $I(\text{Fertility}^2)$  больше 81 (существует очень  
сильная линейная зависимость)
```

```
# значение vif Agriculture и  $I(\text{Agriculture}^2)$  больше 19 (существует сильная линейная  
зависимость), у Catholic значение vif = 2.2
```

```
model_I2 = lm(Infant.Mortality ~ Fertility + Agriculture + I(Fertility^2) + I(Agriculture^2) +  
Catholic, data)
```

```
summary(model_I2)
```

```
vif(model_I2)
```

```
# Значение  $R^2 = 0.25$ , значение vif у всех регрессоров больше 19 (существует очень  
сильная линейная зависимость)
```

```
model_I3 = lm(Infant.Mortality ~ Fertility + Agriculture + Catholic + I(Fertility^2) +  
I(Agriculture^2) + I(Catholic^2), data)
```

```
summary(model_I3)
```

```
vif(model_I3)
```

```
# Значение  $R^2 = 0.23$ , значение vif Agriculture = 1.8, значение vif остальных регрессоров  
больше 85 (существует очень сильная линейная зависимость)
```



```
model_I4 = lm(Infant.Mortality ~ Fertility + Catholic + I(Fertility^2) + Agriculture +  
I(Catholic^2), data)
```

```
summary(model_I4)
```

```
vif(model_I4)
```

Значение $R^2 = 0.22$, значение vif у Agriculture и $I(\text{Agriculture}^2)$ больше 19 (существует сильная линейная зависимость)

значение vif у остальных регрессоров меньше 2

```
model_I5 = lm(Infant.Mortality ~ Fertility + Agriculture + I(Agriculture^2) + Catholic, data)
```

```
summary(model_I5)
```

```
vif(model_I5)
```

Значение $R^2 = 0.24$, значение vif Fertility = 1.8, значение vif остальных регрессоров больше 19 (существует очень сильная линейная зависимость)

```
model_I6 = lm(Infant.Mortality ~ Fertility + I(Agriculture^2) + I(Catholic^2) + Agriculture +  
Catholic, data)
```

```
summary(model_I6)
```

```
vif(model_I6)
```

Значение $R^2 = 0.21$, значение vif у Catholic и $I(\text{Catholic}^2)$ больше 67 (существует очень сильная линейная зависимость)

значение vif остальных регрессоров меньше 2

```
model_I7 = lm(Infant.Mortality ~ Fertility + Agriculture + I(Catholic^2) + Catholic, data)
```

```
summary(model_I7)
```

```
vif(model_I7)
```

Приложение 3.

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

data = swiss

model = lm(Infant.Mortality ~ Fertility + Agriculture + Catholic + Agriculture * Catholic, data)
summary(model)

t_critical = qt(0.975, df = 42) # критерий стьюдента = 2.018082

se_fertility = 0.033
model$coefficients[2] - t_critical * se_fertility # нижняя граница доверительного интервала =
0.04831026
model$coefficients[2] + t_critical * se_fertility # верхняя граница доверительного интервала
= 0.1815037
# 0 не попадает в границы, соответственно связь между Infant.Mortality и Fertility есть

se_agriculture = 0.025
model$coefficients[3] - t_critical * se_agriculture # нижняя граница доверительного
интервала = -0.03009676
model$coefficients[3] + t_critical * se_agriculture # верхняя граница доверительного
интервала = 0.07080733
# 0 попадает в границы, соответственно связь между Infant.Mortality и Agriculture не
наблюдается

se_catholic = 0.03
model$coefficients[4] - t_critical * se_catholic # нижняя граница доверительного интервала =
0.02708346
model$coefficients[4] + t_critical * se_catholic # верхняя граница доверительного интервала
= 0.1481684
# 0 не попадает в границы, соответственно связь между Infant.Mortality и Catholic
наблюдается
```

```
se_agriculture_catholic = 0.0005
```

```
model$coefficients[4] - t_critical * se_agriculture_catholic # нижняя граница доверительного  
интервала = 0.08661687
```

```
model$coefficients[4] + t_critical * se_agriculture_catholic # верхняя граница доверительного  
интервала = 0.08863496
```

```
# 0 не попадает в границы, соответственно связь между Infant.Mortality и Agriculture *  
Catholic есть
```

```
new.data = data.frame(Fertility = 80, Agriculture = 10, Catholic = 13)
```

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

```
# fit = 21.03059 (прогноз для введенных данных) в доверительном интервале [18.9545;  
23.10667]
```

Приложение 4.

```
library("lmtest")
library("rlms")
library("dplyr")
library("car")

data <- rlms_read("r19i_os26c.sav")
data2 = select(data, oj13.2, o_age, oh5, o_educ, status, oj6.2, o_marst, oj24, oj260 , oj1.1.1, oj6)
# oj260 - Владеет ли респондент иностранным языком, помимо языков бывших стран СНГ?
# oj1.1.1 - Насколько Вы удовлетворены или не удовлетворены Вашей работой в целом?
# oj6 - Есть ли у респондента подчиненные на этой работе?
data2 = na.omit(data2)

# Зарплата
sal = as.numeric(data2$oj13.2)
data2["salary"] = (sal - mean(sal)) / sqrt(var(sal))
data2["salary"]

# Возраст
age1 = as.numeric(data2$o_age)
data2["age"] = (age1 - mean(age1)) / sqrt(var(age1))
data2["age"]

# Пол
data2["sex"] = as.character(data2$oh5)
data2$sex[which(data2$sex != '1')] <- 0
data2$sex[which(data2$sex == '1')] <- 1
data2["sex"] = as.character(data2$sex)
data2["sex"]

# Наличие высшего образования
```

```

h_educ = as.character(data2$o_educ)
data2["higher_educ"] = data2$o_educ
data2["higher_educ"] = 0
data2$higher_educ[which(h_educ == '21')] <- 1
data2$higher_educ[which(h_educ == '22')] <- 1
data2$higher_educ[which(h_educ == '23')] <- 1
data2["higher_educ"] = as.numeric(data2$higher_educ)
data2["higher_educ"]

```

Населенный пункт

```

status2 = as.character(data2$status)
data2["status1"] = 0
data2$status1[which(status2 == '1')] <- 1
data2$status1[which(status2 == '2')] <- 1
data2["status1"] = as.numeric(data2$status1)
data2$status1

```

Длительность рабочей недели

```

dur1 = as.character(data2$oj6.2)
dur2 = lapply(dur1, as.integer)
dur3 = as.numeric(unlist(dur2))
data2["dur"] = (dur3 - mean(dur3)) / sqrt(var(dur3))
data2$dur

```

Семейное положение

```

wed = as.character(data2$o_marst)
data2["wed1"] = data2$o_marst
data2$wed1 = 0
data2$wed1[which(wed == '1')] <- 1
data2$wed1[which(wed == '3')] <- 1
data2$wed1 = as.numeric(data2$wed1)

```

```
data2["wed2"]=data2$o_marst
data2$wed2 = 0
data2$wed2[which(wed == '2')] <- 1
data2$wed2 = as.numeric(data2$wed2)
```

```
data2["wed3"]=data2$o_marst
data2$wed3 = 0
data2$wed3[which(wed == '4')] <- 1
data2$wed3 = as.numeric(data2$wed3)
```

```
data2["wed4"]=data2$o_marst
data2$wed4 = 0
data2$wed4[which(wed=='5')] <- 1
data2$wed4 = as.numeric(data2$wed4)
```

```
data2["wed5"]=data2$o_marst
data2$wed5 = 0
data2$wed5[which(wed=='1')] <- 1
data2$wed5 = as.numeric(data2$wed5)
```

```
# Владение иностранным языком
data2["lan"] = as.character(data2$oj260)
data2["language"] = 0
data2$language[which(data2$lan == '1')] <- 1
data2["language"] = as.numeric(data2$language)
data2$language
```

```
# Насколько респондент удовлетворен или не удовлетворен работой в целом
data2["sat"] = as.character(data2$oj1.1.1)
data2["work_sat"] = 1
```

```
data2$work_sat[which(data2$sat == '4')] <- 0
data2["work_sat"] = as.numeric(data2$work_sat)
data2$work_sat
```

```
# есть ли у респондента подчиненные на этой работе
```

```
data2["sub"] = as.character(data2$oj6)
data2["subord"] = 0
data2$subord[which(data2$sub == '1')] <- 1
data2["subord"] = as.numeric(data2$subord)
data2$subord
```

```
data3 = select(data2, salary, age, sex, higher_educ, status1, dur, wed1, wed2, wed3, wed4, wed5,
language, work_sat, subord)
```

```
model1 = lm(salary ~ age + sex + higher_educ + status1 + dur + wed1 + wed2 + wed3 + wed4 +
language + work_sat + subord, data3)
```

```
summary(model1)
```

```
vif(model1)
```

```
# значение vif для параметра wed2 = 31 (высокий), попробуем убрать его
```

```
model2 = lm(salary ~ age + sex + higher_educ + status1 + dur + wed1 + wed3 + wed4 +
language + work_sat + subord, data3)
```

```
summary(model2)
```

```
vif(model2)
```

```
# значение  $R^2$  не изменилось, т.е. модель не испортилась, а значение vif уменьшилось (<2
у всех переменных)
```

```
# у параметров wed3 и wed4 нет звездочек, уберем их (хоть значение vif у параметров
маленькое)
```

```
model3 = lm(salary ~ age + sex + higher_educ + status1 + dur + wed1 + language + work_sat +
subord, data3)
```

```
summary(model3)
```

```
vif(model3)
```

```
# уберем параметр wed1, поскольку у него одна звездочка
```

```
model4 = lm(salary ~ age + sex + higher_educ + status1 + dur + language + work_sat + subord,  
data3)
```

```
summary(model4)
```

```
vif(model4)
```

значение R^2 исходной модели практически не отличается от полученной, т.е. модель хорошая, $R^2 = 0.18$

значение vif всех параметров < 2 , и все параметры имеют 3 звездочки

```
model4 = lm(salary ~ I(age^0.1) + sex + higher_educ + status1 + dur + language + work_sat +  
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~ I(age^0.2) + sex + higher_educ + status1 + dur + language + work_sat +  
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~ I(age^0.3) + sex + higher_educ + status1 + dur + language + work_sat +  
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~ I(age^0.4) + sex + higher_educ + status1 + dur + language + work_sat +  
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~ I(age^0.5) + sex + higher_educ + status1 + dur + language + work_sat +  
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```



```
model4 = lm(salary ~I(age^0.6) + sex + higher_educ + status1 + dur + language + work_sat +
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^0.7) + sex + higher_educ + status1 + dur + language + work_sat +
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^0.8) + sex + higher_educ + status1 + dur + language + work_sat +
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^0.9) + sex + higher_educ + status1 + dur + language + work_sat +
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.1) + sex + higher_educ + status1 + dur + language + work_sat +
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.2) + sex + higher_educ + status1 + dur + language + work_sat +
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.4) + sex + higher_educ + status1 + dur + language + work_sat +  
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.5) + sex + higher_educ + status1 + dur + language + work_sat +  
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.6) + sex + higher_educ + status1 + dur + language + work_sat +  
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.7) + sex + higher_educ + status1 + dur + language + work_sat +  
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.8) + sex + higher_educ + status1 + dur + language + work_sat +  
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.9) + sex + higher_educ + status1 + dur + language + work_sat +  
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^2) + sex + higher_educ + status1 + dur + language + work_sat +  
subord, data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +  
log(subord), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + log(status1) + dur + language + work_sat  
+ log(subord), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +  
I(subord^0.1), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +  
I(subord^0.2), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +  
I(subord^0.3), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +  
I(subord^0.4), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + log(language) + work_sat  
+ I(subord^0.5), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +  
I(subord^0.6), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +  
I(subord^0.7), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + log(sex) + higher_educ + status1 + dur + language + work_sat  
+ I(subord^0.8), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +  
I(subord^0.9), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + log(dur) + language + work_sat  
+ I(subord^1.1), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + I(dur^0.5) + language +  
work_sat + I(subord^1.2), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + I(dur^2) + language + work_sat  
+ I(subord^1.3), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + I(dur^4) + language + work_sat  
+ I(subord^1.4), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +  
I(subord^1.5), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +  
I(subord^1.6), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + I(work_sat^2)  
+ I(subord^1.7), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +  
I(subord^1.8), data3) # top
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + log(work_sat)  
+ I(subord^1.9), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +  
I(subord^2), data3)
```

```
summary(model4)
```

```
vif(model4)
```

```
model4 = lm(salary ~I(age^1.3) + sex + higher_educ + status1 + dur + language + work_sat +  
I(subord^1.8), data3) # top
```

```
summary(model4)
```

```
vif(model4)
```

```
# У этой модели самый высокий  $R^2 = 0.20$ , значение vif всех переменных  $< 2$ , у каждой 3  
звездочки
```

```
# У параметра age отрицательный множитель, т.е. между ним и объясняемой переменной  
существует обратная взаимосвязь
```

```
# Чем моложе респондент, тем выше у него зарплата
```

```
# У переменных sex, higher_educ, status1, dur, language, work_sat, subord положительный  
множитель, следовательно
```

```
# между ними и объясняемой переменной существует прямая взаимосвязь
```

```
# У мужчин, респондентов с высшим образованием, респондентов из города или  
областного центра,
```

```
# респондентов с большей продолжительностью рабочей недели, владеющих  
иностранным языком респондентов,
```

```
# удовлетворенных работой респондентов, имеющих подчиненных респондентов -  
зарплата выше
```

```
# респонденты, не состоявшие в браке
```

```
data4 = subset(data3, wed5 == 1)
```

```
# Среди них выделим подмножество городских жителей
```

```
data5 = subset(data4, status1 == 1)
```

```

# Строим модель

model_set1 = lm(salary ~ I(age^1.3) + sex + higher_educ + dur + language + work_sat +
I(subord^1.8), data5)

summary(model_set1)

vif(model_set1)

# R^2 = 0.02, все переменные без звездочек - модель плохая


# женщины

data7 = subset(data3, sex == 0)

# разведенные женщины

data8 = subset(data7, wed3 == 1)

# разведенные женщины без высшего образования

data9 = subset(data8, higher_educ == 0)


# Строим модель

model_set2 = lm(salary ~ I(age^1.3) + dur + status1 + language + work_sat + I(subord^1.8),
data9)

summary(model_set2)

vif(model_set2)

# значение R^2 = 0.13, переменные work_sat и subord имеют две звезды

# значение vif параметров < 1.042538, модель в целом плоха

```

Приложение 5.

```
import numpy as np

import pandas

from sklearn.linear_model import LogisticRegression

from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier


data = pandas.read_csv("vgsales.csv")

# для построения лог. регрессии были выбраны столбцы 'Genre', 'Year', 'NA_Sales',
'EU_Sales', 'JP_Sales', 'Other_Sales'

data_sel = data.loc[:, data.columns.isin(['Platform', 'Genre', 'Year', 'NA_Sales', 'EU_Sales',
'JP_Sales', 'Other_Sales'])]

data_sel = data_sel.dropna()

# строим классификацию по столбцу Platform, PS2 - класс 0, остальные платформы - класс
1

data_sel['Platform'] = np.where(data_sel['Platform'] == 'PS2', 0, 1)


# обрабатываем столбец Genre и присвоим различным значениям столбца цифру от 0 до 6

data_sel['Genre'] = np.where(data_sel['Genre'] == 'Action', 0, data_sel['Genre'])
data_sel['Genre'] = np.where(data_sel['Genre'] == 'Adventure', 1, data_sel['Genre'])
data_sel['Genre'] = np.where(data_sel['Genre'] == 'Sports', 2, data_sel['Genre'])
data_sel['Genre'] = np.where(data_sel['Genre'] == 'Platform', 3, data_sel['Genre'])
data_sel['Genre'] = np.where(data_sel['Genre'] == 'Racing', 4, data_sel['Genre'])
data_sel['Genre'] = np.where(data_sel['Genre'] == 'Fighting', 5, 6)


# убираем из данных столбец Platform, по которому строим классификацию

platforms = data_sel.loc[:, data_sel.columns.isin(['Platform'])]

x = data_sel.loc[:, data_sel.columns.isin(['Genre', 'Year', 'NA_Sales', 'EU_Sales', 'JP_Sales',
'Other_Sales'])]

# разделим набор данных на тестовую и обучающую выборку

x_train, x_test, y_train, y_test = train_test_split(x, platforms, test_size=0.3)
```



```

# строим лог. регрессию
clf = LogisticRegression(random_state=0, solver='lbfgs', multi_class='multinomial')
clf.fit(x_train, y_train)
y_pred = clf.predict(x_test)

# print(clf.score(x_test, y_test))
print("accuracy: " + str(np.average(cross_val_score(clf, x_test, y_test, scoring='accuracy'))))
# accuracy: 0.8687488274165641
print("f1: " + str(np.average(cross_val_score(clf, x_test, y_test, scoring='f1'))))
# f1: 0.9297651473118999
print("precision: " + str(np.average(cross_val_score(clf, x_test, y_test, scoring='precision'))))
# precision: 0.8687488274165641
print("recall: " + str(np.average(cross_val_score(clf, x_test, y_test, scoring='recall'))))
# recall : 1.0

grid_search_cv = GridSearchCV(cv=3, error_score='raise',
    estimator=DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
        max_features=None, max_leaf_nodes=None,
        min_impurity_decrease=0.0,
        min_samples_leaf=1, min_samples_split=2,
        min_weight_fraction_leaf=0.0, random_state=42,
        splitter='best'), n_jobs=None,
    param_grid={'max_depth': list(range(2, 20)), 'min_samples_split': [2, 3, 4]},
    pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
    scoring=None, verbose=1)
grid_search_cv.fit(x_train, y_train)

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

# построим классификатор типа Случайный Лес

```

```
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(grid_search_cv.best_estimator_, x_test,
y_test, scoring='accuracy'))))
```

```
# accuracy: 0.9328435930041067
```

```
print("f1: " + str(np.average(cross_val_score(grid_search_cv.best_estimator_, x_test, y_test,
scoring='f1'))))
```

```
# f1 : 0.9615368345963002
```

```
print("precision: " + str(np.average(cross_val_score(grid_search_cv.best_estimator_, x_test,
y_test, scoring='precision'))))
```

```
# precision: 0.9603767058423113
```

```
print("recall: " + str(np.average(cross_val_score(grid_search_cv.best_estimator_, x_test, y_test,
scoring='recall'))))
```

```
# recall: 0.9627700398537327
```

```
# сравнивая значения метрик для классификаторов типа лог. регрессии и случайного леса,
# делаем вывод, что второй построенный классификатор лучше
```

Приложение 6.

```
import pandas as pd
data = pd.read_csv('forestfires.csv')
data.shape
# по данным первым и последним пяти строкам нельзя сказать, что есть объекты с NA
data.head()
data.tail()
from sklearn.preprocessing import LabelEncoder

label = LabelEncoder()
label.fit(data.day)
data.day = label.transform(data.day)

label2 = LabelEncoder()
label2.fit(data.month)
data.month = label2.transform(data.month)
data['month'].value_counts()
data['day'].value_counts()
data.head()
data.tail()
data.info()
data.describe()
data.corr()
from sklearn.preprocessing import StandardScaler

scale_features_std = StandardScaler()
features_std = scale_features_std.fit_transform(data[['DMC', 'FFMC', 'DC', 'ISI', 'temp', 'RH',
'wind', 'rain', 'area']])
features_std

import matplotlib.pyplot as plt
```

```

plt.figure(figsize=(12, 10))
plt.scatter(data.DMC, data.FFMC, linewidth=1.4)

plt.xlabel('DMC')
plt.ylabel('FFMC')
# на графике видим, что значения лежат примерно в одном диапазоне

data[['DMC', 'FFMC', 'DC', 'ISI', 'temp', 'RH', 'wind', 'rain', 'area']] = features_std
data
data.describe()
target = data.area
train = data.drop(['X', 'Y', 'day', 'area'], axis=1)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(train, target, test_size = 0.3, random_state = 42)

N_train, _ = X_train.shape
N_test, _ = X_test.shape
print(N_train, N_test) # 361 156

import seaborn as sns

# Plot
plt.figure(figsize=(12,10), dpi= 80)
sns.heatmap(data.corr(), xticklabels=data.corr().columns, yticklabels=data.corr().columns,
cmap='YlGnBu', center=0, annot=True,linewidths=.5)

# Decorations
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()

```

```

from sklearn.decomposition import PCA

%matplotlib inline

import matplotlib.pyplot as plt

pca = PCA()
pca.fit(X_train)
X_pca = pca.transform(X_train)

for i, component in enumerate(pca.components_):
    print("{} component: {}% of initial variance".format(i + 1,
        round(100 * pca.explained_variance_ratio_[i], 2)))
    print(" + ".join("%.3f x %s" % (value, name)
        for value, name in zip(component, train.columns)))

import numpy as np

plt.figure(figsize=(10,7))
plt.plot(np.cumsum(pca.explained_variance_ratio_), color='k', lw=2)
plt.axhline(0.9, c='r')
plt.axvline(6, c='b')

from sklearn.manifold import TSNE

tsne = TSNE(
    n_components=2, perplexity=10, early_exaggeration=12,
    learning_rate=200, n_iter=500, n_iter_without_progress=20,
    metric='euclidean', init='random', verbose=0, random_state=42, n_jobs=-1)

z = tsne.fit_transform(data)
df = pd.DataFrame()

```

```
df["y"] = target
df["comp-1"] = z[:,0]
df["comp-2"] = z[:,1]

sns.scatterplot(x="comp-1", y="comp-2", hue=df.y.tolist(),
                data=df).set(title="My data tsne")
```

Приложение 7.

```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")

data = pd.read_csv('forestfires.csv')
data.info()

from sklearn.preprocessing import LabelEncoder

label = LabelEncoder()
label.fit(data.day)
data.day = label.transform(data.day)

label2 = LabelEncoder()
label2.fit(data.month)
data.month = label2.transform(data.month)
data.describe()

from sklearn.preprocessing import StandardScaler

scale_features_std = StandardScaler()
features_std = scale_features_std.fit_transform(data[['DMC', 'FFMC', 'DC', 'ISI', 'temp', 'RH',
'wind', 'rain', 'area']])
features_std
data[['DMC', 'FFMC', 'DC', 'ISI', 'temp', 'RH', 'wind', 'rain', 'area']] = features_std
data
data.describe()

target = data.area
train = data.drop(['X', 'Y', 'day', 'area'], axis=1)
```

```

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(train, target, test_size = 0.3, random_state = 42)


N_train, _ = X_train.shape
N_test, _ = X_test.shape
print(N_train, N_test) # 361 156


import seaborn as sns
import matplotlib.pyplot as plt


# Plot
plt.figure(figsize=(12,10), dpi= 80)

sns.heatmap(data.corr(), xticklabels=data.corr().columns, yticklabels=data.corr().columns,
            cmap='YlGnBu', center=0, annot=True,linewidths=.5)


# Decorations
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()


import statsmodels.api as sm

X_train_sm = data.drop(['area'], axis=1)
X_train_sm = sm.add_constant(X_train_sm)
y_train = data.area
model = sm.OLS(y_train, X_train_sm).fit()

print(model.summary())


from statsmodels.stats.outliers_influence import variance_inflation_factor

X = data[['X', 'Y', 'day', 'month', 'DMC', 'FFMC', 'DC', 'ISI', 'temp', 'RH', 'wind', 'rain']]

```



```

vif_data = pd.DataFrame()
vif_data["feature"] = X.columns

# посчитаем значение VIF параметров
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(len(X.columns))]

print(vif_data)

X = data[['X', 'day', 'month', 'DMC', 'FFMC', 'DC', 'ISI', 'temp', 'RH', 'wind', 'rain']]

vif_data = pd.DataFrame()
vif_data["feature"] = X.columns

# посчитаем значение VIF параметров
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(len(X.columns))]

print(vif_data)

X_train_sm = data.drop(['area', 'Y'], axis = 1)

for i in range(len(X_train_sm['ISI'])):
    X_train_sm['ISI'][i] = X_train_sm['ISI'][i] * X_train_sm['ISI'][i]

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

print(model.summary())

X_train_sm = data.drop(['area', 'Y'], axis = 1)

```

```

for i in range(len(X_train_sm['ISI'])):
    X_train_sm['ISI'][i] = X_train_sm['ISI'][i] * X_train_sm['ISI'][i] * X_train_sm['ISI'][i]

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

print(model.summary())

X_train_sm = data.drop(['area', 'Y'], axis = 1)

for i in range(len(X_train_sm['temp'])):
    X_train_sm['temp'][i] = X_train_sm['temp'][i] * X_train_sm['temp'][i]

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

print(model.summary())

X_train_sm = data.drop(['area', 'Y'], axis = 1)

for i in range(len(X_train_sm['DMC'])):
    X_train_sm['DMC'][i] = X_train_sm['DMC'][i] * X_train_sm['DMC'][i]

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

print(model.summary())

from math import log

X_train_sm = data.drop(['area', 'Y'], axis = 1)

```

```

for i in range(len(X_train_sm['DMC'])):
    if X_train_sm['DMC'][i] > 0:
        X_train_sm['DMC'][i] = log(X_train_sm['DMC'][i], 2)

```

```

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

```

```

print(model.summary())

```

```

X_train_sm = data.drop(['area', 'Y'], axis = 1)

```

```

for i in range(len(X_train_sm['rain'])):
    if X_train_sm['rain'][i] > 0:
        X_train_sm['rain'][i] = pow(X_train_sm['rain'][i], 0.4)

```

```

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

```

```

print(model.summary())

```

```

X_train_sm = data.drop(['area', 'Y'], axis = 1)

```

```

for i in range(len(X_train_sm['DC'])):
    X_train_sm['DC'][i] = pow(X_train_sm['DC'][i], 2)

```

```

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

```

```

print(model.summary())

```

```
X_train_sm = data.drop(['area', 'Y'], axis = 1)
```

```
for i in range(len(X_train_sm['DC'])):
```

```
    if X_train_sm['DC'][i] > 0:
```

```
        X_train_sm['DC'][i] = log(X_train_sm['DC'][i], 10)
```

```
X_train_copy = sm.add_constant(X_train_sm)
```

```
model = sm.OLS(y_train, X_train_copy).fit()
```

```
print(model.summary())
```

```
X_train_sm = data.drop(['area', 'Y'], axis = 1)
```

```
for i in range(len(X_train_sm['DC'])):
```

```
    if X_train_sm['DC'][i] > 0:
```

```
        X_train_sm['DC'][i] = pow(X_train_sm['DC'][i], 1.4)
```

```
X_train_copy = sm.add_constant(X_train_sm)
```

```
model = sm.OLS(y_train, X_train_copy).fit()
```

```
print(model.summary())
```

```
X_train_sm = data.drop(['area', 'Y'], axis = 1)
```

```
for i in range(len(X_train_sm['month'])):
```

```
    X_train_sm['month'][i] = X_train_sm['month'][i] * X_train_sm['month'][i]
```

```
X_train_copy = sm.add_constant(X_train_sm)
```

```
model = sm.OLS(y_train, X_train_copy).fit()
```

```
print(model.summary())
```

```

X_train_sm = data.drop(['area', 'Y'], axis = 1)

for i in range(len(X_train_sm['RH'])):
    X_train_sm['RH'][i] = X_train_sm['RH'][i] * X_train_sm['RH'][i]

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

print(model.summary())

```

```

X_train_sm = data.drop(['area', 'Y'], axis = 1)

for i in range(len(X_train_sm['RH'])):
    if X_train_sm['RH'][i] > 0:
        X_train_sm['RH'][i] = log(X_train_sm['RH'][i], 10)

```

```

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

print(model.summary())

```

```

X_train_sm = data.drop(['area', 'Y'], axis = 1)

for i in range(len(X_train_sm['RH'])):
    if X_train_sm['RH'][i] > 0:
        X_train_sm['RH'][i] = pow(X_train_sm['RH'][i], 0.3)

```

```

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

print(model.summary())

```

```
X_train_sm = data.drop(['area', 'Y'], axis = 1)
```

```
for i in range(len(X_train_sm['FFMC'])):
```

```
    if X_train_sm['FFMC'][i] > 0:
```

```
        X_train_sm['FFMC'][i] = pow(X_train_sm['FFMC'][i], 0.5)
```

```
X_train_copy = sm.add_constant(X_train_sm)
```

```
model = sm.OLS(y_train, X_train_copy).fit()
```

```
print(model.summary())
```

```
X_train_sm = data.drop(['area', 'Y'], axis = 1)
```

```
for i in range(len(X_train_sm['FFMC'])):
```

```
    if X_train_sm['FFMC'][i] > 0:
```

```
        X_train_sm['FFMC'][i] = log(X_train_sm['FFMC'][i], 10)
```

```
X_train_copy = sm.add_constant(X_train_sm)
```

```
model = sm.OLS(y_train, X_train_copy).fit()
```

```
print(model.summary())
```

```
X_train_sm = data.drop(['area', 'Y'], axis = 1)
```

```
for i in range(len(X_train_sm['FFMC'])):
```

```
    X_train_sm['FFMC'][i] = X_train_sm['FFMC'][i] * X_train_sm['FFMC'][i] *  
    X_train_sm['FFMC'][i]
```

```
X_train_copy = sm.add_constant(X_train_sm)
```

```
model = sm.OLS(y_train, X_train_copy).fit()
```

```
print(model.summary())
```

```

X_train_sm = data.drop(['area', 'Y'], axis = 1)

for i in range(len(X_train_sm['wind'])):
    X_train_sm['wind'][i] = X_train_sm['wind'][i] * X_train_sm['wind'][i]

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

print(model.summary())

X_train_sm = data.drop(['area', 'Y'], axis = 1)

for i in range(len(X_train_sm['wind'])):
    if X_train_sm['wind'][i] > 0:
        X_train_sm['wind'][i] = pow(X_train_sm['wind'][i], 0.2)

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

print(model.summary())

X_train_sm = data.drop(['area', 'Y'], axis = 1)

for i in range(len(X_train_sm['wind'])):
    if X_train_sm['wind'][i] > 0:
        X_train_sm['wind'][i] = log(X_train_sm['wind'][i], 20)

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

print(model.summary())

```

```

X_train_sm = data.drop(['area', 'Y'], axis = 1)

for i in range(len(X_train_sm['wind'])):
    if X_train_sm['wind'][i] > 0:
        X_train_sm['wind'][i] = log(X_train_sm['wind'][i], 20)
    X_train_sm['month'][i] = X_train_sm['month'][i] * X_train_sm['month'][i]
    if X_train_sm['rain'][i] > 0:
        X_train_sm['rain'][i] = pow(X_train_sm['rain'][i], 0.4)
    if X_train_sm['DC'][i] > 0:
        X_train_sm['DC'][i] = pow(X_train_sm['DC'][i], 1.4)

X_train_copy = sm.add_constant(X_train_sm)
model = sm.OLS(y_train, X_train_copy).fit()

print(model.summary())

```

```

import numpy as np
from sklearn.metrics import mean_squared_error as MSE

X_train_sm = data.drop(['area', 'Y', 'day', 'FFMC', 'RH'], axis = 1)

for i in range(len(X_train_sm['wind'])):
    if X_train_sm['wind'][i] > 0:
        X_train_sm['wind'][i] = log(X_train_sm['wind'][i], 20)
    X_train_sm['month'][i] = X_train_sm['month'][i] * X_train_sm['month'][i]
    if X_train_sm['rain'][i] > 0:
        X_train_sm['rain'][i] = pow(X_train_sm['rain'][i], 0.4)
    if X_train_sm['DC'][i] > 0:
        X_train_sm['DC'][i] = pow(X_train_sm['DC'][i], 1.4)

X_train_copy = sm.add_constant(X_train_sm)

```



```
model = sm.OLS(y_train, X_train_copy).fit()
```

```
y_pred = model.predict(X_test)
```

```
rmse = np.sqrt(MSE(y_test, y_pred))
```

```
print("RMSE : % f" %(rmse))
```

```
print(model.summary())
```

```
X_train_sm = data.drop(['area', 'Y'], axis = 1)
```

```
X_train_copy = sm.add_constant(X_train_sm)
```

```
model = sm.OLS(y_train, X_train_sm).fit()
```

```
print(model.summary())
```

```
region_column = []
```

```
for i in range(data.shape[0]):
```

```
    if 1 <= data['X'][i] <= 3 and 2 <= data['Y'][i] <= 6:
```

```
        region_column.append(0)
```

```
    elif (4 <= data['X'][i] <= 5 and 3 <= data['Y'][i] <= 6) or (data['X'][i] == 6 and 1 <= data['Y'][i] <= 7):
```

```
        region_column.append(1)
```

```
    elif (9 >= data['X'][i] >= 7 >= data['Y'][i] >= 3) or (8 <= data['X'][i] <= 9 and 8 <= data['Y'][i] <= 9):
```

```
        region_column.append(2)
```

```
    # else:
```

```
    #     region_column.append(3) мы не найдем таких данных, поэтому всего выделили три района
```

```
data['region'] = region_column
```

```
from sklearn.ensemble import GradientBoostingRegressor
```

```

params = {'n_estimators':500,
          'max_depth':10,
          'criterion':'mse',
          'learning_rate':0.003,
          'min_samples_leaf':16}
target = data.region
train = data.drop(['region', 'X', 'Y'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(train, target, test_size = 0.3, random_state = 42)

# Обучаем
gbr = GradientBoostingRegressor(**params)
gbr.fit(X_train,y_train)

gbr.score(X_test, y_test)

from sklearn.model_selection import cross_val_score
from xgboost import XGBClassifier

model = XGBClassifier(use_label_encoder=False)

model.fit(X_train, y_train)
# scores = cross_val_score(model, X_test, y_test, cv=3)
print(model.score(X_test, y_test))

from sklearn.metrics import mean_squared_error as MSE
from xgboost import XGBRegressor
import numpy as np

model = XGBRegressor(n_estimators=1000, max_depth=12, eta=0.1, subsample=0.7,
                     colsample_bytree=0.8)
model.fit(X_train, y_train)

```

```

y_pred = model.predict(X_test)
rmse = np.sqrt(MSE(y_test, y_pred))
print("RMSE : % f" %(rmse))

region_area = {0: 0, 1: 0, 2: 0}
for i in range(len(data['region'])):
    region_area[data['region'][i]] += data['area'][i]
print(region_area)

data_reg = data.set_index('region')
region0 = data_reg.drop([1, 2], axis=0)
region0.describe()

region2 = data_reg.drop([1, 0], axis=0)
region2.describe()

print('FFMC difference: ' + str(region0['FFMC'].mean() - region2['FFMC'].mean()))
print('DMC difference: ' + str(region0['DMC'].mean() - region2['DMC'].mean()))
print('DC difference: ' + str(region0['DC'].mean() - region2['DC'].mean()))
print('ISI difference: ' + str(region0['ISI'].mean() - region2['ISI'].mean()))
print('temp difference: ' + str(region0['temp'].mean() - region2['temp'].mean()))
print('RH difference: ' + str(region0['RH'].mean() - region2['RH'].mean()))
print('wind difference: ' + str(region0['wind'].mean() - region2['wind'].mean()))
print('rain difference: ' + str(region0['rain'].mean() - region2['rain'].mean()))

```