



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: Анализ набора данных о качестве вина «Red Wine Quality» (kaggle.com)
приказ университета о направлении на НИР
от «9» февраля 2023 г. № 735 - С

Отчет представлен к
рассмотрению:
Студент группы КМБО-03-
22

Лазарев А.К.
(расшифровка подписи)
«31» мая 2023г.

Отчет утвержден.
Допущен к защите:

Руководитель НИР от
кафедры

Петрусеви́ч Д.А.
(расшифровка подписи)
«31» мая 2023г.

Москва 2023



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)

Студенту 1 курса учебной группы КМБО-03-22 института искусственного
интеллекта Лазареву Александру Кирилловичу

(фамилия, имя и отчество)

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

Время НИР: с «09» февраля 2023 по «31» мая 2023

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», «к 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 («к ближайших соседей»); построением модели линейной регрессии.

3. ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ: анализ набора данных о качестве вина «Red Wine Quality» (kaggle.com)

4. ОГРАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ: построить модели предсказания качества, оценить вклад каждого компонента. Построить статистические оценки параметров и гистограммы распределений. Построить бинарный классификатор: хорошее/плохое вино. Выделить аномальные объекты

Заведующий кафедрой

высшей математики

«09» февраля 2023 г.

СОГЛАСОВАНО

Руководитель НИР от кафедры:

«09» февраля 2023 г.

Задание получил:

«09» февраля 2023 г.

Ю.И.Худак


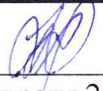




(Петрусович Д.А.)

(фамилия и инициалы)

(Лазарев А.К.)

(фамилия и инициалы)

ИНСТРУКТАЖ ПРОВЕДЕН:

Вид мероприятия	ФИО ответственного, подпись, дата	ФИО студента, подпись, дата
Охрана труда	Петрусеви́ч Д.А.  «09» февраля 2023 г.	Лазаре́в А.К..  «09» февраля 2023 г.
Техника безопасности	Петрусеви́ч Д.А.  «09» февраля 2023 г.	Лазаре́в А.К..  «09» февраля 2023 г.
Пожарная безопасность	Петрусеви́ч Д.А.  «09» февраля 2023 г.	Лазаре́в А.К..  «09» февраля 2023 г.
Правила внутреннего распорядка	Петрусеви́ч Д.А.  «09» февраля 2023 г.	Лазаре́в А.К..  «09» февраля 2023 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)

студента Лазарева А.К. 1 курса группы КМБО-03-22 очной формы обучения,
обучающегося по направлению подготовки 01.03.02 «Прикладная математика и
информатика»,
профиль «Математическое моделирование и вычислительная математика»

Неделя	Сроки выполнения	Этап	Отметка о выполнении
1	09.02.2023	Выбор темы НИР. Пройти инструктаж по технике безопасности	✓
1	09.02.2023	Вводная установочная лекция	✓
2	18.02.2023	Построение и оценка парной регрессии с помощью языка R	✓
3	25.02.2023	Построение и оценка множественной регрессии с помощью языка R	✓
4	04.03.2023	Построение доверительных интервалов. Обработка факторных переменных. Мультиколлинеарность	✓
5	11.03.2023	Гетероскедастичность	✓
6	18.03.2023	Классификация	✓
7	25.03.2023	Кластеризация. Предобработка данных	✓
8	01.04.2023	Метод главных компонент	✓
9	08.04.2023	Ансамбли классификаторов.	

		Беггинг. Бустинг	
16	27.05.2023	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 НИР от
кафедры



/ ФИО / Петрусевич Д.А.

Обучающийся



/ ФИО / Лазарев А.К.

Оглавление

Задачи	3
Задача №1	3
Задача №2	7
Задача №3	13
Задача №4	21
Задача №5	27
Задача №6	36
Заключение.....	55
Список литературы.....	57
Приложения.....	58
Приложение 1	58
Приложение 2	60
Приложение 3	63
Приложение 4	69
Приложение 5	71
Приложение 6	74

Задачи

Задача №1

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: *Swiss*.

Объясняемая переменная: *Education*.

Регрессоры: *Fertility*, *Examination*.

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

В первой части задачи вычисляются основные статистические показатели для каждой переменной. В частности, вычисляются среднее значение, дисперсия и стандартное отклонение. Эти показатели могут быть использованы для описания характеристик распределения переменных.

Для переменной *Education* были рассчитаны следующие параметры:

- Среднее значение (*mean*): ~ 10.98 . Это означает, что в среднем образование находится на низком уровне.
- Дисперсия (*var*): ~ 92.46 . Это почти на порядок больше среднего, что свидетельствует о заметном разбросе данных.
- Стандартное отклонение ($\sqrt{\text{var}}$): ~ 9.61 .

Для переменной *Fertility* были рассчитаны следующие параметры:

- Среднее значение (*mean*): ~ 70.14 . Это означает, что в среднем рождаемость находится на нормальном уровне.
- Дисперсия (*var*): ~ 156.04 . Это всего в 2 раза больше среднего, что свидетельствует об умеренном разбросе данных.
- Стандартное отклонение ($\sqrt{\text{var}}$): ~ 12.49 .

Для переменной *Examination* были рассчитаны следующие параметры:

- Среднее значение (*mean*): ~ 16.49 . Это означает, что в среднем низкий процент людей получает высший балл на экзамене.
- Дисперсия (*var*): ~ 63.65 . Это более чем в 3 раза больше среднего, что свидетельствует о заметном разбросе данных.
- Стандартное отклонение ($\sqrt{\text{var}}$): ~ 7.98 .

Таким образом, мы видим, что переменные *Education* и *Examination* имеют заметный разброс, в то время как переменная *Fertility* имеет более небольшой разброс. Среднее

значение всех переменных находится в разных диапазонах, что также свидетельствует о различиях между переменными.

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

Коэффициенты были получены с помощью метода наименьших квадратов, который является одним из основных методов оценки параметров в линейной регрессии. Он позволяет определить такие значения параметров a и b , которые минимизируют сумму квадратов отклонений (расстояний) между фактическими значениями зависимой переменной и теоретическими значениями, предсказанными моделью. Значения коэффициентов нужных моделей и их стандартные ошибки могут быть найдены в соответствующей таблице, которая строится по команде `lm()`.

(Таблицы 1.1 и 1.2 соответственно)

С помощью этих действий

- Для модели *Education~Fertility* получены коэффициенты **$a = 46.82$** и **$b = -0.51$** , говорящие об отрицательной зависимости образования от рождаемости.

Таблица 1.1. Характеристики модели зависимости параметра *Education* от параметра *Fertility* в наборе данных Swiss.

Параметр \ Характеристики	Значение	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	46.81788	6.11244	7.659	1.08e-09	***
Fertility	-0.51095	0.08582	-5.954	3.66e-07	***

- Для модели *Education~Agriculture* получены коэффициенты **$a = -2.90$** и **$b = 0.84$** , говорящие о положительной зависимости образования от экзаменационных результатов.

Таблица 1.2. Характеристики модели зависимости параметра *Education* от параметра *Examination* в наборе данных Swiss.

Параметр \ Характеристики	Значение	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	-2.9015	2.3507	-1.234	0.223	
Examination	0.8418	0.1286	6.546	4.81e-08	***

Соответствующие графики приведены в Приложении 1 на рисунках 1 и 2

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

Коэффициент детерминации R^2 является мерой того, насколько хорошо модель описывает данные. Он принимает значения от 0 до 1, где 0 означает, что модель не

объясняет изменчивость данных, а 1 означает, что модель идеально подходит для данных и объясняет всю изменчивость.

В данном случае, для модели *Education~Fertility* коэффициент детерминации R^2 был равен **0.44**, что означает, что модель объясняет 44% изменчивости данных. Для модели *Education~Examination* коэффициент детерминации R^2 был равен **0.48**, что означает, что модель объясняет 48% изменчивости данных.

Таким образом, обе модели на приемлемом для парной регрессии уровне объясняют значительную часть изменчивости данных, однако модель *Education~Examination* объясняет данные немного лучше, чем модель *Education~Fertility*.

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

Значение р-статистики показывает вероятность получения таких или еще более крайних значений коэффициента регрессии, если на самом деле между регрессором и зависимой переменной нет никакой связи. Если значение р-статистики меньше уровня значимости (обычно 0.05), то можно сделать вывод о том, что связь между регрессором и зависимой переменной статистически значима.

Для модели *Education~Fertility* значение р-статистики равно **3.659e-07** (у каждого коэффициента по **3** звездочки), что говорит о том, что связь между рождаемостью и образованием является статистически значимой.

Для модели *Education~Examination* значение р-статистики равно **4.811e-08** (коэффициент **a** имеет **3** звездочки), что является хорошим показателем. Однако коэффициент **b** представлен не точно (без единой звездочки), что говорит о невозможности предсказания некоторых отклонений.

Выводы

Из проведенного анализа двух моделей линейной регрессии можно сделать следующие выводы:

- Модель *Education~Fertility* показала отрицательную зависимость уровня образования от уровня рождаемости, что соответствует интуитивным представлениям. Анализ p-value и коэффициента детерминации R^2 подтвердил

значимость модели и ее способность хорошо объяснять отклонения от среднего значения.

- Модель *Education~Examination* показала положительную зависимость уровня образования от экзаменационных результатов. Однако, коэффициент b не был значим, поэтому модель не может точно предсказать все отклонения от среднего значения. Значение коэффициента детерминации R^2 показало, что модель объясняет 48% отклонений от среднего значения.
- Обе модели показали наличие связи между образованием и другими переменными, однако модель *Education~Examination* объясняет отклонения от среднего значения менее точно.
- Оценка значимости коэффициента признака (переменной) в модели осуществляется с помощью p -статистики. В обеих моделях p -значение было мало, что говорит о статистической значимости коэффициента.

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

Задача №2

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: *Swiss*.

Объясняемая переменная: *Examination*.

Регрессоры: *Catholic*, *Agriculture*, *Fertility*.

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

Проверим линейную регрессию $Agriculture \sim Catholic + Fertility$. R^2 в этой модели равен примерно 0.16, что говорит о том, что параметр *Agriculture* не зависит линейно от других регрессоров. Это позволяет использовать *Agriculture* при построении математических моделей.

Зависимость $Catholic \sim Agriculture + Fertility$ имеет R^2 около 0.25. Коэффициент детерминации меньше 80%, поэтому можно заключить, что линейной зависимости между регрессорами нет. Параметр *Catholic* можно использовать в линейных моделях.

В регрессии $Fertility \sim Agriculture + Catholic$ значение R^2 составляет около 0.21. Это также говорит о том, что регрессоры не линейно зависимы. Параметры *Agriculture* и *Catholic* можно использовать в линейной регрессии вместе с *Fertility*.

Как можно увидеть, значения коэффициента детерминации во всех случаях относительно низкие, что подтверждает отсутствие линейной зависимости между регрессорами. Однако, следует помнить, что коэффициенты R^2 могут быть относительными величинами, и в контексте конкретной задачи они могут оказаться достаточно высокими или низкими. Поэтому, если имеется сомнение в правильности вывода о независимости регрессоров, необходимо проводить дополнительные тесты и анализировать результаты более внимательно, что будет продемонстрировано в работе далее.

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.

При построении модели был принято решение о постепенном добавлении регрессоров с оценкой качества модели и значимости переменной на каждом шаге.

1. Examination~Catholic:

$R^2 \sim 0.31$, имеется слабая зависимость. Регрессор *Catholic* статистически значим на данном этапе, так как имеет 3 звезды.

2. Examination~Catholic+Agriculture:

$R^2 \sim 0.56$, модель стала значительно лучше. Регрессор *Agriculture* статистически значим на данном этапе, так как имеет 3 звезды, однако *Catholic* стал менее значимым, так как потерял одну звезду.

3. Examination~Catholic+Agriculture+Fertility:

$R^2 \sim 0.66$, еще один заметный прирост, модель стала заметно лучше и может считаться хорошей, однако *Catholic* теперь имеет всего 1 звездочку. Это говорит о том, что религиозная принадлежность не так сильно влияет на результаты экзамена, как *Agriculture* и *Fertility* (по 3 звезды), и, возможно, использование этого регрессора излишне.

4. Examination~Agriculture+Fertility;

Попробуем исключить Catholic. R^2 снизился всего на 2% (~ 0.64), значит Catholic все-таки можно исключить и предсказания модели не ухудшатся (ухудшатся не значительно в рамках поставленной задачи).

В итоге получаем искомую хорошую зависимость, в которой доля объясненной дисперсии высока, и каждый регрессор статистически значим. Статистические характеристики приведены в таблице 2.1.

Таблица 2.1. Характеристики модели зависимости параметра *Examination* от параметров *Agriculture* и *Fertility* в наборе данных Swiss.

Параметр \ Характеристики	Значение	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	46.45927	4.01858	11.561	6.33e-15	***
Agriculture	-0.18400	0.03313	-5.553	1.52e-06	***
Fertility	-0.29438	0.06024	-4.887	1.40e-05	***

$R^2 \sim 0.64$ для данной модели.

3. Введите в модель логарифмы регрессоров (если возможно). Сравните модели и выберите наилучшую.

Перед включением функций от регрессоров в модель обязательно провести исследование на корреляцию. В решении это делается непосредственно построение парных регрессий исходных признаков и функций от них. Если коэффициент детерминации высокий, то регрессор коррелирует с функцией от него, и при добавлении функции в исследуемую модель исходный регрессор следует убрать, чтобы избежать потери качества предсказаний и сохранить значимость всех признаков.

Таким образом были получены значения

1. Для модели $\log(\text{Agriculture}) \sim \text{Agriculture}$ $R^2 \sim 0.77$.
2. Для модели $\log(\text{Fertility}) \sim \text{Fertility}$ $R^2 \sim 0.98$.

В обоих случаях зависимости сильные, значит логарифмами нужно заменять исходные регрессоры. Исследуем новые модели, их показатели приведены в таблицах 2.2 и 2.3.

Таблица 2.2. Характеристики модели зависимости параметра *Examination* от параметров $I(\log(\text{Agriculture}))$ и *Fertility* в наборе данных Swiss.

Параметр \ Характеристики	Значение	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	54.97174	4.61354	11.915	2.3e-15	***
$I(\log(\text{Agriculture}))$	-5.34421	1.07746	-4.960	1.1e-05	***
Fertility	-0.26323	0.0661	-3.981	0.000253	***

$R^2 \sim 0.61$ для данной модели.

Таблица 2.3. Характеристики модели зависимости параметра *Examination* от параметров *Agriculture* и $I(\log(\text{Fertility}))$ в наборе данных Swiss.

Параметр \ Характеристики	Значение	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	101.9975	16.2989	6.258	1.41e-07	***
Agriculture	-0.1819	0.0344	-5.287	3.71e-06	***
$I(\log(\text{Fertility}))$	-18.0232	3.9846	-4.523	4.57e-05	***

$R^2 \sim 0.62$ для данной модели.

Из полученных данных следует, что введение не дало желанных результатов, R^2 уменьшился во всех случаях. К тому же выросли стандартные ошибки. *Examination* куда лучше зависит от исходных регрессоров нежели от их логарифмов, значит вводить их не требуется.

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

Также, как и в случае с логарифмами, исследуем квадраты и произведение регрессоров на линейную зависимость от исходных регрессоров.

1. Для модели $(\text{Agriculture}^2) \sim \text{Agriculture}$ $R^2 \sim 0.94$.
2. Для модели $(\text{Fertility}^2) \sim \text{Fertility}$ $R^2 \sim 0.98$.
3. Для модели $(\text{Agriculture} * \text{Fertility}) \sim \text{Fertility} + \text{Agriculture}$ $R^2 \sim 0.97$.

Как и в случае с логарифмами, все зависимости сильные, значит квадратами и произведением нужно заменять исходные регрессоры. Исследуем новые модели, их показатели приведены в таблицах 2.4, 2.5 и 2.6.

Таблица 2.4. Характеристики модели зависимости параметра *Examination* от параметров $I(Agriculture^2)$ и *Fertility* в наборе данных Swiss.

Параметр \ Характеристики	Значение	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	44.2626904	4.1724018	10.608	1.05e-13	***
Fertility	-0.3188861	0.0613626	-5.197	5.02e-06	***
$I(Agriculture^2)$	-0.0017602	0.0003487	-5.048	8.23e-06	***

$R^2 \sim 0.61$ для данной модели.

Таблица 2.5. Характеристики модели зависимости параметра *Examination* от параметров *Agriculture* и $I(Fertility^2)$ в наборе данных Swiss.

Параметр \ Характеристики	Значение	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	37.135738	2.3825278	15.587	< 2e-16	***
Agriculture	-0.188371	0.0323852	-5.817	6.27e-07	***
$I(Fertility^2)$	-0.002189	0.0004339	-5.045	8.31e-06	***

$R^2 \sim 0.65$ для данной модели.

Таблица 2.6. Характеристики модели зависимости параметра *Examination* от параметра $I(Agriculture * Fertility)$ в наборе данных Swiss.

Параметр \ Характеристики	Значение	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	28.5281446	1.7403738	16.392	< 2e-16	***
$I(Agriculture * Fertility)$	-0.0032970	0.0004272	-7.718	8.9e-10	***

$R^2 \sim 0.56$ для данной модели.

Нетрудно заметить, что из всех полученных моделей не уменьшился R^2 , не возрос p-value и не возросла стандартная ошибка лишь у модели $Examination \sim Agriculture + I(Fertility^2)$. В ней наоборот, немного увеличился R^2 , p-value уменьшилось на порядок (у *Fertility*), а стандартная ошибка уменьшилась на 2 порядка. Такая зависимость сильнее той, в которой участвует простой столбец *Fertility*. Значит лучшей моделью является

$Examination \sim Agriculture + I(Fertility^2)$.

- Для полученной зависимости оцените доверительные интервалы для всех коэффициентов в модели, $p = 95\%$.

Найдем доверительные интервалы для всех коэффициентов регрессоров. Количество измерений в обучающей выборке 44, рассчитано 3 коэффициента. Число степеней свободы в модели $44 - 3 = 41$. Для такого числа степеней свободы и $p = 95\%$ значение t -критерия Стьюдента оказалось равно ~ 2.02 .

Доверительный интервал строится по формуле:

$[Estimate - t * Std.Error; Estimate + t * Std.Error]$, где *Estimate* - значение коэффициента регрессора, t - критерий Стьюдента, *Std.Error* - значение стандартной ошибки регрессора.

Тогда для коэффициента $k1$ регрессора *Agriculture* в лучшей модели доверительный интервал имеет вид $k1: [-0.19 - 2.02 * 0.03; -0.19 + 2.02 * 0.03]$, $\Rightarrow k1: [-0.25; -0.13]$.

Необходимые для подсчета параметры *Estimate* и *Std.Error* были получены из второй строки таблицы 2.5 и округлены с точностью до сотых, значение t было получено ранее и составило 2.02.

Аналогично для коэффициента $k2$ регрессора $I(Fertility^2)$ в лучшей модели доверительный интервал имеет вид $k2: [-0.0022 - 2.02 * 0.0004; -0.0022 + 2.02 * 0.0004]$, $\Rightarrow k2: [-0.003; -0.0014]$.

Необходимые для подсчета параметры *Estimate* и *Std.Error* были получены из третьей строки таблицы 2.5 и округлены с точностью до четвертого порядка после запятой, значение t было получено ранее и составило 2.02.

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

Ни один из полученных в предыдущем пункте решения доверительных интервалов не содержит нуля, следовательно нулевая гипотеза может быть отвергнута и коэффициенты $k1$ и $k2$ не могут принимать равное нулю значение (на уровне значимости 5%). Это также свидетельствует о статистической значимости регрессоров *Agriculture* и $I(Fertility^2)$.

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

Найдем доверительный интервал для прогноза значения *Examination* при значениях регрессоров *Agriculture* = 60, *Fertility* = 65.

Для этого создадим датафрейм с заданными значениями и воспользуемся функцией `predict()`. В результате получим таблицу 2.7.

Таблица 2.7. Результат работы команды `predict(..., interval = "confidence")` для построения доверительного интервала для прогноза с регрессорами *Agriculture* = 60, *Fertility* = 65.

fit	lwr	upr
-----	-----	-----

16.58542	14.81365	18.35718
----------	----------	----------

Прогноз модели равен 16.58542.

Доверительный интервал: [14.81365; 18.35718].

Выводы

Из проведенного анализа множественной регрессии можно сделать следующие выводы:

- С целью выявления корреляций признаков были построены все возможные зависимости регрессоров друг от друга. Анализ показал отсутствие линейной зависимости между регрессорами, что указало на возможность построение качественной модели множественной регрессии с их использованием.
- При пошаговом добавлении/удалении регрессоров была выявлена лучшая модель для описания линейной зависимости. Регрессор *Catholic* не вошел в эту модель в виду его низкого уровня значимости по сравнению с другими регрессорами и крайне слабого влияния на качество модели.
- В целях улучшения качества найденной модели были предприняты попытки включения в зависимость функций от регрессоров. Таким образом, была выявлена наилучшая модель $Examination \sim Agriculture + I(Fertility^2)$, которая больше всего подходила для описания данных и указывала на положительную зависимость между оценкой за экзамен и проценту мужчин, занимающихся сельскохозяйственной деятельностью и отрицательную зависимость между оценкой за экзамен и уровнем рождаемости.
- Для наилучшей модели были посчитаны доверительные интервалы для всех коэффициентов регрессоров. Исходя из их значений для обоих коэффициентов была отвергнута нулевая гипотеза на уровне значимости 5%. Также был получен прогноз модели для произвольных значений регрессоров и посчитан его доверительный интервал. При 60% доле мужчин, занимающихся сельскохозяйственным делом, и уровне рождаемости 65% процент людей, получивших высокие результаты на армейском экзамене будет равен 16.58542 при доверительном интервале [14.81365; 18.35718].

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

Задача №3

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Номер волны выборки РМЭЗ НИУ ВШЭ: 12.

Подмножества для пункта 5: не вступавшие в брак, без высшего образования; городские жители, состоящие в браке.

В приведенном задании требуется описать социально-экономическое положение граждан Российской Федерации. Для этого были загружены необходимые данные исследования НИУ ВШЭ. Из них были удалены пропущенные значения и отобраны следующие параметры:

- `h_age` – возраст;
- `hh5` – пол;
- `hj13.2` – зарплата;
- `h_marst` - семейное положение;
- `h_educ` - наличие высшего образования;
- `hj6.2` - длительность рабочей недели;
- `hj6` – наличие подчиненных;
- `status` - тип населенного пункта;
- `hj1.1.2` - удовлетворенность;
- `hj23` - владеет ли государство компанией?
- `hj24` - есть ли иностранные совладельцы?
- `hj32` – наличие второй работы;
- `hj60` - все денежные поступления за месяц (не только зарплата).

Выбранные данные были преобразованы для дальнейшего анализа по следующим принципам:

- Из параметра, отвечающего семейному положению, были сделаны дамми-переменные (с помощью one-hot-encoding): 1) переменная *wed1* имеет значение 1 в случае, если респондент женат, 0 – в противном случае; 2) *wed2* = 1, если респондент разведён или вдовец; 3) *wed3* = 1, если респондент никогда не состоял в браке.
- Из параметра пол была сделана переменная *sex*, имеющая значение 1 для мужчин и равную 0 для женщин.
- Из параметра, отвечающего типу населённого пункта, была создана дамми-переменная *status2* со значением 1 для города или областного центра, 0 – в противоположном случае.
- Введены параметры *higher_educ*, *satisfy*, *state_owner*, *foreign_owner*, *second_job*, и *subordinates*, характеризующие наличие полного высшего образования, удовлетворенность, совладение компанией государством, совладение компанией иностранцами, наличие второй работы и наличие подчиненных соответственно. Из

них также были созданы дамми-переменные, принимающие значение 1 в случае выполнения условия в заданном вопросе и 0 в противном случае.

- Факторные переменные, «имеющие много значений», такие как: зарплата, длительность рабочей недели, возраст, все денежные поступления - были преобразованы в вещественные переменные и нормализованы через стандартное отклонение. Полученные значения были записаны в переменные *salary*, *age*, *dur* и *payments* соответственно.

Таким образом был получен дата-фрейм *data2*, необходимый для дальнейшей работы.

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

Была построена модель

$salary \sim dur + wed1 + wed2 + wed3 + age + sex + status2 + higher_educ + satisfy + state_owner +$
 $+ foreign_owner + subordinates + payments.$

Ее статистические характеристики приведены в таблице 3.1.

Таблица 3.1. Характеристики модели зависимости параметра *salary* от параметров *dur*, *wed1*, *wed2*, *wed3*, *age*, *sex*, *status2*, *higher_educ*, *satisfy*, *state_owner*, *foreign_owner*, *subordinates*, *payments* в наборе данных РМЭЗ 12.

Параметр \ Характеристики	Значение	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	-0.15445	0.04490	-3.440	0.000591	***
dur	0.06982	0.01189	5.872	4.82e-09	***
wed1	-0.00822	0.03772	-0.218	0.827481	
wed2	-0.06828	0.04630	-1.475	0.140435	
wed3	-0.10584	0.04749	-2.229	0.025917	*
age	-0.06997	0.01261	-5.550	3.13e-08	***
sex	0.16746	0.02422	6.913	5.85e-12	***
status2	0.10836	0.02652	4.085	4.53e-05	***
higher_educ	0.09835	0.02803	3.509	0.000457	***
satisfy	0.10465	0.02355	4.443	9.21e-06	***
state_owner	-0.12783	0.02483	-5.149	2.80e-07	***
foreign_owner	0.21808	0.05577	3.911	9.43e-05	***
subordinates	0.14969	0.02842	5.266	1.50e-07	***
payments	0.70216	0.01255	55.958	< 2e-16	***

$R^2 \sim 0.65$ для данной модели.

Из приведенной статистики видно, что регрессоры *wed1*, *wed2* и *wed3* незначимы по сравнению с другими регрессорами, значит их можно сразу исключить из модели.

Проанализируем новую модель без этих регрессоров. Ее статистика приведена в таблице 3.2.

Таблица 3.2. Характеристики модели зависимости параметра *salary* от параметров *dur*, *age*, *sex*, *status2*, *higher_educ*, *satisfy*, *state_owner*, *foreign_owner*, *subordinates*, *payments* в наборе данных РМЭЗ 12.

Параметр \ Характеристики	Значение	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	-0.18473	0.03234	-5.712	1.23e-08	***
dur	0.06985	0.01189	5.876	4.71e-09	***
age	-0.06071	0.01145	-5.302	1.23e-07	***
sex	0.17525	0.02369	7.399	1.81e-13	***
status2	0.10516	0.02654	3.963	7.59e-05	***
higher_educ	0.09584	0.02798	3.425	0.000623	***
satisfy	0.10158	0.02356	4.312	1.68e-05	***
state_owner	-0.12619	0.02480	-5.089	3.83e-07	***
foreign_owner	0.21825	0.05582	3.910	9.45e-05	***
subordinates	0.15423	0.02842	5.427	6.22e-08	***
payments	0.70302	0.01255	56.012	< 2e-16	***

$R^2 \sim 0.65$ для данной модели.

В итоге R^2 не изменился, р-значение для некоторых коэффициентов стало лучше и все регрессоры теперь значимы. Чтобы убедиться в этом проверим модель на мультиколлинеарность при помощи команды `vif()`. Результаты приведены в таблице 3.3.

Таблица 3.3. Коэффициенты вздутия дисперсии для каждого регрессора в модели *salary ~ dur + age + sex + status2 + satisfy + state_owner + foreign_owner + subordinates + payments + higher_educ* в наборе данных РМЭЗ 12.

dur	age	sex	status2	satisfy	state_owner	foreign_owner	subordinates	payments	higher_educ
1.12	1.04	1.09	1.08	1.08	1.12	1.04	1.15	1.25	1.18

У всех регрессоров коэффициент вздутия не превосходит 3, что говорит об отсутствии мультиколлинеарности.

Таким образом, *salary ~ dur + age + sex + status2 + satisfy + state_owner + foreign_owner + subordinates + payments + higher_educ* Наилучшая модель на данном этапе.

2. Поэкспериментируйте с функциями вещественных параметров: используйте логарифмы, степени (хотя бы от 0.1 до 2 с шагом 0.1), произведения вещественных регрессоров. Выделите наилучшие модели из построенных: по значимости параметров, включённых в зависимости, и по объяснённой с помощью построенных зависимостей разбросу $adjusted\ R^2 - R^2_{adj}$.

В нашем случае вещественными переменными являются *age*, *dur* и *payments*. Так как переменные стандартизованы, для некоторой части их значений невозможны операции возведения в рациональную степень и взятия логарифма. Для исправления данной ситуации прибавим к переменным минимальное целое число, превосходящее минимальное значение для данной переменной. Для нахождения таких значений воспользуемся встроенной функцией `min()`. Получим:

- `min(data2$dur) ~ -3.22 =>` для возведения в рациональную степень и взятия логарифма к значению надо прибавить 4
- `min(data2$age) ~ -2.14 =>` для возведения в рациональную степень и взятия логарифма к значению надо прибавить 3
- `min(data2$payments) ~ -0.98 =>` для возведения в рациональную степень и взятия логарифма к значению надо прибавить 1

Теперь введение функций от вещественных переменных возможно.

Начнем с возведения в рациональную степень. При переборе каждого показателя от 0.1 до 2 с шагом 0.1 для трех переменных потребуется проанализировать 999 новых моделей. Для рациональности решение был реализован словарь, в котором ключом выступает набор степеней регрессоров, а значением коэффициент R^2_{adj} модели с новыми параметрами. Перебор был реализован тройным циклом, в котором на каждой итерации в словарь добавлялся новый элемент.

Чтобы выявить лучшую точность, для значений словаря была вызвана функция `max()`. Полученный коэффициент детерминации $R^2 \sim 0.6541$, что превосходит предыдущее значение.

Оптимальный набор степеней был получен по значению словаря. Для *dur* оптимальная степень оказалась равна 0.3, для *age* оптимальная степень оказалась равна 2, а для *payments* степень составила 1.1.

Так как R^2 возрос не значительно, требовалось провести анализ статистических характеристик новой модели. Результаты функции `summary()` для модели с новым набором вещественных переменных приведены в таблице 3.4.

Таблица 3.4. Характеристики модели зависимости параметра *salary* от параметров $I((dur+4)^{0.3})$, $I((age+3)^2)$, *sex*, *status2*, *higher_educ*, *satisfy*, *state_owner*, *foreign_owner*, *subordinates*, $I((payments+1)^{1.1})$ в наборе данных РМЭЗ 12.

Параметр \ Характеристики	Значение	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	-1.671607	0.167540	-9.977	< 2e-16	***
$I((dur + 4)^{0.3})$	0.630486	0.107372	5.872	4.82e-09	***
$I((age + 3)^2)$	-0.010875	0.001839	-5.915	3.73e-09	***
sex	0.181094	0.023552	7.689	2.04e-14	***
status2	0.126824	0.026277	4.826	1.47e-06	***
higher_educ	0.103057	0.027777	3.710	0.000211	***
satisfy	0.108384	0.023402	4.631	3.80e-06	***
state_owner	-0.125156	0.024580	-5.092	3.78e-07	***

foreign_owner	0.217513	0.055403	3.926	8.85e-05	***
subordinates	0.156435	0.028214	5.545	3.22e-08	***
$I((\text{payments} + 1)^{1.1})$	0.596288	0.010526	56.649	$< 2e-16$	***

$R^2 \sim 0.65$ для данной модели.

Значимость свободного коэффициента значительно увеличилась. Р-статистика для каждого коэффициента снизилась, как и стандартные ошибки. Также были проверены коэффициенты вздутия, которые также снизились. Все указывает на то, что найденная модель – лучшая на данном этапе.

Далее был произведен анализ моделей с логарифмами от вещественных переменных.

Полученное значение R^2 для модели $\text{salary} \sim \log_dur + \log_age + \text{sex} + \text{status2} + \text{higher_educ} + \text{satisfy} + \text{state_owner} + \text{foreign_owner} + \text{subordinates} + \log_payments$ составило 0.3, что очень мало. Возможность включения в модель логарифмов от регрессоров была отвергнута.

Далее был произведен анализ моделей с произведением вещественных переменных.

- Для модели $\text{salary} \sim I(\text{dur} * \text{age}) + \text{sex} + \text{status2} + \text{higher_educ} + \text{satisfy} + \text{state_owner} + \text{foreign_owner} + \text{subordinates} + \text{payments}$ значение R^2 составило 0.64, что указывает на достаточно высокую долю объясненной дисперсии. Такая модель может считаться хорошей и использоваться для дальнейшего анализа.
- Для модели $\text{salary} \sim I(\text{dur} * \text{payments}) + \text{sex} + \text{status2} + \text{higher_educ} + \text{satisfy} + \text{state_owner} + \text{foreign_owner} + \text{subordinates} + \text{age}$ значение R^2 составило 0.26. Столь малое значение коэффициента детерминации указывает на нерациональность дальнейшего использования и изучения этой модели.
- Аналогично, для модели $I(\text{age} * \text{payments}) + \text{sex} + \text{status2} + \text{higher_educ} + \text{satisfy} + \text{state_owner} + \text{foreign_owner} + \text{subordinates} + \text{dur}$ R^2 оказался очень низким (0.27). Эта модель не будет использована для дальнейшего анализа.
- Аналогично для модели $\text{salary} \sim I(\text{dur} * \text{age} * \text{payments}) + \text{sex} + \text{status2} + \text{higher_educ} + \text{satisfy} + \text{state_owner} + \text{foreign_owner} + \text{subordinates}$ R^2 оказался очень низким (0.24). Эта модель не будет использована для дальнейшего анализа.

Последним этапом анализа моделей с функциями от вещественных переменных является построение моделей с комбинацией функций. В ходе решения были получены 2 модели с функциями от регрессоров, хорошо описывающие данные. Рассмотрим их комбинацию.

Так для функции $\text{salary} \sim I((\text{dur} + 4)^{0.3} + (\text{age} + 3)^2) + \text{sex} + \text{status2} + \text{higher_educ} + \text{satisfy} + \text{state_owner} + \text{foreign_owner} + \text{subordinates} + I((\text{payments} + 1)^{1.1})$ значение R^2 составило 0.6489, что ниже показателя лучшей модели. Это значит, что комбинацию функций использовать не следует.

Таким образом была выявлена лучшая модель:

$\text{salary} \sim I((\text{dur} + 4)^{0.3}) + I((\text{age} + 3)^2) + \text{sex} + \text{status2} + \text{higher_educ} + \text{satisfy} + \text{state_owner} + \text{foreign_owner} + \text{subordinates} + I((\text{payments} + 1)^{1.1})$.

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

Для выполнения этого пункта проанализируем второй столбец таблицы 3.4, в котором содержатся коэффициенты для каждого регрессора:

- $I((dur + 4)^{0.3})$ - положительный
- $I((age + 3)^2)$ - отрицательный
- *sex* - положительный
- *status2* - положительный
- *higher_educ* - положительный
- *satisfy* - положительный
- *state_owner* - отрицательный
- *foreign_owner* - положительный
- *subordinates* - положительный
- $I((payments + 1)^{1.1})$ – положительный

По этим данным можно сделать следующий вывод:

Большую зарплату получают молодые мужчины с продолжительной рабочей неделей, имеющие высшее образование, проживающие в городе, удовлетворённые своей заработной платой, имеющие подчиненных. При этом, если иностранные фирмы и частники являются совладельцами или владельцами их предприятия, то это положительно сказывается на уровне заработной платы. Если государство является совладельцами или владельцами их предприятия, то это отрицательно сказывается на уровне заработной платы. Также отмечу, что чем больше денег индивид получает в течении месяца (учитывается не только зарплата), тем больше у него зарплата.

4. Оцените лучшие модели для подмножества индивидов, указанных в варианте. Сделайте вывод о том, какие индивиды получают наибольшую зарплату.

Для выявления никогда не состоявших в браке индивидов без высшего образования воспользуемся функцией `subset()` для набора датафрейма `data2`. Необходимые условия отбора:

- $wed3 = 1$ (никогда не состоял в браке);
- $higher_educ = 0$ (не имеет высшего образования).

Параметр *wed3* исключен из лучшей модели, а параметр *higher_educ* требуется исключить, так как его значение на выбранном подмножестве всегда равно нулю и он не нуждается в предсказании. Незначащие регрессоры также были убраны из модели после ее обучения на выбранном подмножестве.

Применим функцию `summary()` для наилучшей модели, обученной на выбранном подмножестве. Ее результаты приведены в таблице 3.5.

Таблица 3.5. Характеристики модели зависимости параметра *salary* от параметров $I((dur+4)^{0.3})$, $I((age+3)^2)$, *sex*, *status2*, *state_owner*, *subordinates*, $I((payments+1)^{1.1})$ в наборе данных РМЭЗ 12

Параметр \ Характеристики	Значение	Std. Error	t value	Pr(> t)	Уровень значимости
---------------------------	----------	------------	---------	----------	--------------------

(Intercept)	-1.67458	0.30928	-5.414	1.36e-07	***
$I((dur + 4)^{0.3})$	0.54559	0.20619	2.646	0.008620	**
sex	0.18978	0.05137	3.695	0.000267	***
subordinates	0.43914	0.08774	5.005	1.01e-06	***
$I((payments + 1)^{1.1})$	0.67897	0.02973	22.839	< 2e-16	***

$R^2 \sim 0.72$ для данной модели.

Найденная модель очень точно описывает уровень заработной платы выбранных индивидов. По ней можно сделать следующие выводы:

Наибольшую зарплату из никогда не состоявшие в браке индивидов без высшего образования получают мужчины, имеющие подчиненных с высокой продолжительностью рабочей недели и высоким уровнем денежных поступлений за месяц.

Аналогично поступим со вторым подмножеством.

Необходимые условия отбора:

- $status2 = 1$ (респондент проживает в городе);
- $wed1 = 1$ (респондент состоит в браке).

Применим функцию *summary()* для наилучшей модели, обученной на выбранном подмножестве. Ее результаты приведены в таблице 3.6.

Таблица 3.6. Характеристики модели зависимости параметра *salary* от параметров $I((dur+4)^{0.3})$, $I((age+3)^2)$, *sex*, *status2*, *state_owner*, *subordinates*, $I((payments+1)^{1.1})$ в наборе данных РМЭЗ 12.

Параметр \ Характеристики	Значение	Std. Error	t value	Pr(> t)	Уровень значимости
(Intercept)	-1.932138	0.324862	-5.948	3.54e-09	***
$I((dur + 4)^{0.3})$	0.990571	0.212269	4.667	3.40e-06	***
$I((age + 3)^2)$	-0.012948	0.003304	-3.918	9.41e-05	***
sex	0.192810	0.042060	4.584	5.02e-06	***
higher_educ	0.153670	0.045833	3.353	0.000824	***
state_owner	-0.192970	0.043345	-4.452	9.28e-06	***
subordinates	0.156292	0.046598	3.354	0.000820	***
$I((payments + 1)^{1.1})$	0.572914	0.016717	34.272	< 2e-16	***

$R^2 \sim 0.59$ для данной модели.

Найденная модель чуть менее точно описывает уровень заработной платы выбранных индивидов. По ней можно сделать следующие выводы:

Наибольшую зарплату из городских жителей, состоящих в браке, получают молодые мужчины, с долгой рабочей неделей и высшим образованием, не работающие на государственную компанию, имеющие подчиненных и высокий общий денежный доход.

Выводы

Из проведенного анализа исследования НИУ ВШЭ РМЭЗ 12 можно сделать следующие выводы:

- Из приведенного набора данных были загружены и обработаны переменные, которые наилучшим образом подходят для описания социально-экономического положения граждан Российской Федерации. На их основе, путем удаления статистически незначимых параметров, была построена первая "хорошая" модель, необходимая для описания размера заработной платы: $salary \sim dur + age + sex + status2 + satisfy + state_owner + foreign_owner + subordinates + payments + higher_educ$.
- В целях повышения качества описанной выше модели были рассмотрены ее вариации с использованием функций от вещественных переменных. Рассмотренные функции включали возведение в рациональную степень, взятие логарифма, произведение переменных и комбинации этих функций. Таким образом, была определена наилучшая модель $salary \sim I((dur + 4)^{0.3}) + I((age + 3)^2) + sex + status2 + higher_educ + satisfy + state_owner + foreign_owner + subordinates + I((payments + 1)^{1.1})$, которая использовалась в дальнейшем анализе.
- На основе коэффициентов регрессоров в модели, полученной на предыдущем этапе, были сделаны выводы о влиянии каждого регрессора на уровень заработной платы. Оказалось, что большую зарплату получают молодые мужчины с продолжительной рабочей неделей, имеющие высшее образование, проживающие в городе, удовлетворённые своей заработной платой и имеющие подчиненных. При этом, если иностранные фирмы и частные лица являются совладельцами или владельцами предприятия, то это положительно сказывается на уровне заработной платы. Если же государство является совладельцем или владельцем предприятия, то это отрицательно сказывается на уровне заработной платы. Также отмечается, что чем больше денег индивид получает в течение месяца (учитывается не только зарплата), тем выше его заработная плата.
- В дальнейшем анализе вышеупомянутая модель использовалась для выявления зависимостей между уровнем заработной платы и значимыми параметрами на заданных подмножествах. Удалось определить, что наибольшую зарплату из индивидов, никогда не состоявших в браке и не имеющих высшего образования, получают мужчины, имеющие подчиненных с высокой продолжительностью рабочей недели и высоким уровнем денежных поступлений за месяц. Помимо этого, наибольшую зарплату среди городских жителей, состоящих в браке, получают молодые мужчины с долгой рабочей неделей и высшим образованием, не работающие в государственной компании, имеющие подчиненных и высокий общий денежный доход.

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

Задача №4

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: *Students performance in exams*.

Тип классификатора: *SVM (метод опорных векторов)*.

Классификация по столбцу: *Writing Score (выше среднего значения – класс 0, ниже или совпадает – класс 1)*

1. Обработайте набор данных набор данных, указанный во втором столбце таблицы 4.1, подготовив его к решению задачи классификации. Выделите целевой признак, указанный в последнем столбце таблицы, и удалите его из данных, на основе которых будет обучаться классификатор. Разделите набор данных на тестовую и обучающую выборку. Постройте классификатор типа, указанного в третьем столбце, для задачи классификации по параметру, указанному в последнем столбце. Оцените точность построенного классификатора с помощью метрик precision, recall и F1 на тестовой выборке.

Указанный набор данных был загружен в датафрейм *data*. Для первого ознакомления с выборкой при помощи метода *head* были выведены первые 5 строк датафрейма. Результаты приведены в таблице 4.1.

Таблица 4.1. Первые 5 строк набора данных *Students performance in exams* в датафрейме *data*.

gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
female	group B	bachelor's degree	standard	none	72	72	74
female	group C	some college	standard	completed	69	90	88
female	group B	master's degree	standard	none	90	95	93
male	group A	associate's degree	free/reduced	none	47	57	44
male	group C	some college	standard	none	76	78	75

Для удобства обработки столбцы таблицы были переименованы следующим образом:

- *race/ethnicity* → *ethnicity*;
- *parental level of education* → *parental_level_education*;

- *test preparation course* → *test_course*;
- *math score* → *math_score*;
- *reading score* → *reading_score*;
- *writing score* → *writing_score*.

Поиск дубликатов при помощи метода `duplicated()` не выявил результатов, что указывает на уникальность каждого индивида.

Переменные *math_score*, *reading_score* и *writing_score* числовые и указывают на полученные индивидом баллами за экзамены по математике, чтению и письму соответственно. Они принимают значения от 0 до 100 (оценка за экзамен в баллах). Нечисловые переменные были изучены с помощью метода `unique()`, примененным к каждому столбцу *data*. В результате было определено, что

- Столбец *gender* содержит следующие уникальные значения: *female* и *male*;
- Столбец *ethnicity* содержит следующие уникальные значения: *group B*, *group C*, *group A*, *group D* и *group E*;
- Столбец *parental_level_education* содержит следующие уникальные значения: *bachelor's degree*, *some college*, *master's degree*, *associate's degree*, *high school* и *some high school*;
- Столбец *lunch* содержит следующие уникальные значения: *standard* и *free/reduced*;
- Столбец *test_course* содержит следующие уникальные значения: *none* и *completed*.

По данным результатам эти признаки были преобразованы для дальнейшего анализа и построения классификаторов.

Таким образом:

- Пол можно сделать дамми-переменной (мужчины - 1, женщины - 0).
- Этнической принадлежности будут присвоены соответственно значения от 1 до 5.
- Родительский уровень образования также будет пронумерован от 1 до 6 (чем лучше образование, тем выше значение).
- Переменная, отвечающая за вид получаемого обеда - дамми-переменная (стандартный - 1, со скидкой или бесплатный - 0).
- Наличие подготовительных курсов - дамми-переменная (1 - проходил, 0 - не проходил).

Отмечу, что для построения заданных классификаторов необходимо, чтобы целевая переменная была бинарной. Таким образом, переменная *writing_score* будет иметь класс 0, если ее значение выше среднего, и 1, если ее значение ниже или совпадает со средним.

Таким образом датафрейм был подготовлен к выполнению заданий по построению классификационных моделей. Первые 5 строк обновленных значений *data* приведены в таблице 4.2.

Таблица 4.2. Первые 5 строк набора данных *Students performance in exams* в обновленном датафрейме *data*.

gender	ethnicity	parental_level_of_education	lunch	test_course	math_score	reading_score	writing_score
0	2	5	0	0	72	72	0
0	3	3	0	1	69	90	0
0	2	6	0	0	90	95	0
1	1	4	1	0	47	57	1
1	3	3	1	0	76	78	0

Классификация была реализована следующим образом:

Первым действием целевой признак *writing_score* был выделен в переменную *target* и удален из датафрейма *data* с помощью метода *drop()*.

Далее набор данных был разделен на обучающую и тестовую выборки. Для этого была использована функция *train_test_split()* из библиотеки *scikit-learn*. Тестовая выборка составила четверть от всех данных. Также был указан параметр *random_state=16* для того, чтобы данные не менялись при каждом запуске программы.

В результате были получены 4 переменные *X_train*, *X_test*, *y_train* и *y_test*.

Далее был создан классификатор метода опорных векторов. Для этого использовался класс *SVM* из библиотеки *scikit-learn*. Он был обучен на переменных *X_train* и *y_train*. Для набора *X_test* были получены предсказания классификатора и записаны в переменную *y_pred*. Реализация вышеуказанных действий приведена на рисунке 4.1.

```
from sklearn.svm import SVC

# Создание классификатора
clf = SVC()

# Обучение классификатора на обучающей выборке
clf.fit(X_train, y_train)

# Предсказание классов на тестовой выборке
y_pred = clf.predict(X_test)
```

Рисунок 4.1. Код построения классификатора *SVM* для набора данных *Students performance in exams*.

Для оценки точности классификатора были использованы метрики precision, recall и F1 (функции precision_score, recall_score, f1_score библиотеки scikit-learn).

При сравнении y_{pred} и y_{train} получились следующие результаты:

- Precision: 0.921875
- Recall: 0.9365079365079365
- F1-score: 0.9291338582677166

Значения метрик оказались очень высокими, что говорит о хорошем качестве классификатора и о возможности проведения качественных предсказаний.

2. Постройте классификатор типа Случайный Лес (Random Forest) для решения той же задачи классификации. Оцените его качество с помощью метрик precision, recall и F1 на тестовой выборке. С помощью GridSearch переберите различные комбинации гиперпараметров: на первой итерации задайте большие шаги (50 или 100) по числу деревьев $n_estimators$. На следующих итерациях определите лучшее количество деревьев $n_estimators$ с точностью до 10. Какой из классификаторов оказывается лучше?

Первым действием был создан классификатор случайного леса с использованием конструктора RandomForestClassifier() из библиотеки scikit-learn.

Для подбора оптимального числа деревьев была создана сетка гиперпараметров, включающая гиперпараметр $n_estimators$ (число деревьев в случайном лесе), принимающий значения из списка от 50 до 1000 с шагом 50.

Далее с помощью функции GridSearchCV библиотеки scikit-learn были рассмотрены все возможные варианты классификатора и выявлен лучший. Сравнение велось по уже указанным метрикам precision, recall и f1. Реализация приведена на рисунке 4.2.

```
from sklearn.model_selection import GridSearchCV

grid_search = GridSearchCV(clf, param_grid, scoring=['precision',
'recall', 'f1'], refit='f1')
grid_search.fit(X_train, y_train)
print('Число деревьев для лучших значений метрик в первом приближении:
', grid_search.best_params_)
```

Рисунок 4.2. Код выявления лучшего числа деревьев классификатора RandomForestClassifier для набора данных *Students performance in exams*.

Код вывел лучшее число деревьев в первом приближении – 650.

Для более точной оценки последние действия были повторены для сетки гиперпараметров, в которой $n_estimators$ принимал значения от 600 до 700 с шагом 10.

Таким образом удалось определить, что лучший случайный лес для данного набора должен содержать 660 деревьев.

Дале была произведена оценка классификатора по тому же принципу, как и для классификатора SVM. Получились следующие значения:

- Precision: 0.944
- Recall: 0.9365079365079365
- F1-score: 0.9402390438247011

Заметен прирост точности предсказаний. Это говорит о том, что классификатор случайного дерева, собранный из лучших значений гиперпараметров), предсказывает данные точнее классификатора метода опорных векторов. Таким образом лучший классификатор для данного исследования это:

clf = RandomForestClassifier(n_estimators=660)

По этой модели был оценен вклад каждого признака в предсказание классификатора. Для этого использовался атрибут *feature_importances_*. Были получены следующие результаты, в порядке убывания их значимости:

- Вклад признака *reading_score* в предсказания: 0.51
- Вклад признака *math_score* в предсказания: 0.29
- Вклад признака *parental_level_education* в предсказания: 0.058
- Вклад признака *ethnicity* в предсказания: 0.044
- Вклад признака *test_course* в предсказания: 0.037
- Вклад признака *gender* в предсказания: 0.031
- Вклад признака *lunch* в предсказания: 0.03

Чтобы оценить не только значимость, но и вектор воздействия (эта компонента положительно сказывается на результате или нет), была использована модель множественной регрессии (исключительно в целях определения знака коэффициента каждого регрессора). Все полученные коэффициенты:

- Значение коэффициента регрессора *gender* составило -0.196509.
- Значение коэффициента регрессора *ethnicity* составило -0.002964.
- Значение коэффициента регрессора *parental_level_education* составило 0.039078.
- Значение коэффициента регрессора *lunch* составило -0.196509.
- Значение коэффициента регрессора *test_course* составило 0.225830.
- Значение коэффициента регрессора *math_score* составило 0.291658.
- Значение коэффициента регрессора *reading_score* составило 0.631599.

Сопоставляя значимость каждого признака, полученную из классификатора, со знаком коэффициента этого признака в модели множественной регрессии, заключаю:

Оценку выше среднего за экзамен по письму получают ученики, набравшие высокие баллы за экзамены по чтению и математике, родители которых имеют лучшее образование. Раса таких учеников должна быть ближе к группе А, ими должны быть

пройденны тестовые курсы. Такие ученики зачастую являются девушками, получающими льготные ланчи.

Все признаки в этом выводе перечислены в порядке убывания их значимости для классификатора.

Выводы

- Из набора данных *Students performance in exams* были получены и обработаны все переменные, необходимые для предсказания оценки (выше среднего или ниже среднего) за экзамен по письму. Данные были приведены к числовым типам для решения задачи с использованием методов классификации.
- После разбиения на обучающую и тестовую выборки был создан классификатор метода опорных векторов, показавший высокую точность предсказаний. По данному набору можно точно предсказать, какую оценку получит индивид.
- Для сравнения был создан классификатор случайного леса. При помощи метода выявления лучших гиперпараметров GridSearch, с точностью до 10 было определено, что оптимальное число деревьев в случайном лесе для данного набора составило 660.
- Было проведено сравнение статистических оценок точности классификаторов SVM и RandomForestClassifier. Второй классификатор оказался более точным, значение F1 для него составило ~ 0.94 . Таким образом, для предсказания результатов экзамена по письму был выявлен оптимальный для данного набора данных классификатор `clf = RandomForestClassifier(n_estimators=660)`.
- Сопоставляя значимость каждого признака, полученную из классификатора, со знаком коэффициента этого признака в модели множественной регрессии, было заключено, что оценку выше среднего за экзамен по письму получают ученики, набравшие высокие баллы за экзамены по чтению и математике, родители которых имеют лучшее образование. Раса таких учеников должна быть ближе к группе A, ими должны быть пройдены тестовые курсы. Такие ученики зачастую являются девушками, получающими льготные ланчи.
Все признаки в этом выводе перечислены в порядке убывания их значимости для классификатора.

Код решения задачи и сведения о проверенных моделях приведены в Приложении 4.

Задача №5

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: *Red Wine Quality*.

Данные были загружены в датафрейм *data*. В переменную *target* были сохранены названия столбцов, они понадобятся в дальнейшем исследовании. Первые 5 строк датафрейма *data* приведены в таблице 5.1.

Таблица 5.1. Первые 5 строк набора данных *Red Wine Quality* в датафрейме *data*.

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

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

При использовании поля `shape` установлено, что датафрейм содержит 1599 строк и 12 столбцов.

Для определения количества уникальных значений был использован метод `nunique()`. Таким образом были определены все признаки и число их уникальных значений:

- Признак - `fixed acidity`, число его уникальных значений: 96;
- Признак - `volatile acidity`, число его уникальных значений: 143;
- Признак - `citric acid`, число его уникальных значений: 80;
- Признак - `residual sugar`, число его уникальных значений: 91;
- Признак - `chlorides`, число его уникальных значений: 153;
- Признак - `free sulfur dioxide`, число его уникальных значений: 60;
- Признак - `total sulfur dioxide`, число его уникальных значений: 144;
- Признак - `density`, число его уникальных значений: 436;
- Признак - `pH`, число его уникальных значений: 89;
- Признак - `sulphates`, число его уникальных значений: 96;
- Признак - `alcohol`, число его уникальных значений: 65;
- Признак - `quality`, число его уникальных значений: 6.

Описание каждого признака:

1. Fixed acidity: Фиксированная кислотность

- Большинство кислот, присутствующих в вине, являются фиксированными или неволатильными (не испаряются легко).

2. Volatile acidity: Летучая кислотность

- Количество уксусной кислоты в вине, которая при слишком высоких уровнях может привести к неприятному вкусу уксуса.

3. Citric acid: Лимонная кислота

- Находится в небольших количествах, может добавлять "свежесть" и аромат в вина.

4. Residual sugar: Остаточный сахар

- Количество сахара, оставшегося после окончания ферментации; редко встречаются вина с менее чем 1 граммом/литром, а вина с более чем 45 граммами/литром считаются сладкими.

5. Chlorides: Хлориды

- Количество соли в вине.

6. Free sulfur dioxide: Свободный диоксид серы

- Свободная форма SO₂ находится в равновесии между молекулярным SO₂ (в виде растворенного газа) и ионом бисульфита; предотвращает размножение микроорганизмов и окисление вина.

7. Total sulfur dioxide: Общий диоксид серы

- Количество свободной и связанной формы SO₂; при низких концентрациях SO₂ в вине практически не обнаруживается, но при концентрациях свободного SO₂ свыше 50 ppm, SO₂ становится заметным по запаху и вкусу вина.

8. Density: Плотность

- Плотность воды близка к плотности вина, в зависимости от содержания процента алкоголя и сахара.

9. pH: Уровень pH

- Описывает кислотность или щелочность вина на шкале от 0 (очень кислотное) до 14 (очень щелочное); большинство вин находятся в диапазоне 3-4 на шкале pH.

10. Sulphates: Сульфаты

- Добавка к вину, которая может способствовать уровню диоксида серы (SO₂), который действует как antimicrobial и антиоксидантное вещество.

11. Alcohol: Алкоголь

- процентное содержание алкоголя в вине.

12. Quality: Качество

- выходная переменная (на основе сенсорных данных, оценка от 0 до 10).

2. Сколько категориальных признаков, какие? Столбец с максимальным количеством уникальных значений категориального признака? Есть ли бинарные признаки? Какие числовые признаки? Есть ли пропуски? Сколько объектов с пропусками? Столбец с максимальным количеством пропусков?

Получим тип каждого признака с помощью метода `info()`. Результат кода показал, что в наборе все признаки числовые (`float64`). Единственный числовой признак, имеющий тип `int64` – *quality*. Так как этот столбец содержит всего 6 уникальных значений, данную переменную можно рассматривать как категориальную (6 различных по оценке видов вин). Вышеупомянутый метод также показал, что в наборе данных нет пропущенных значений.

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

Чтобы в первом приближении оценить наличие выбросов, был использован метод `describe()`, который описал распределение данных в столбцах. Результат приведен в таблице 5.2.

Таблица 5.2. Статистика распределения данных в столбцах набора *Red Wine Quality*.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599	1599	1599	1599	1599	1599	1599	1599	1599	1599	1599	1599
mean	8.32	0.53	0.27	2.54	0.09	15.87	46.47	0.99	3.31	0.66	10.42	5.64
std	1.74	0.18	0.19	1.41	0.05	10.46	32.89	0.002	0.15	0.17	1.07	0.81
min	4.6	0.12	0	0.9	0.0	1	6	0.99	2.74	0.33	8.4	3
25%	7.1	0.39	0.09	1.9	0.07	7	22	0.99	3.21	0.55	9.5	5
50%	7.9	0.52	0.26	2.2	0.08	14	38	0.99	3.31	0.62	10.2	6
75%	9.200	0.64	0.42	2.6	0.09	21	62	0.99	3.4	0.73	11.1	6
max	15.9	1.58	1	15.5	0.61	72	289	1	4.01	2	14.9	8

Наибольшие значения стандартного отклонения и наибольшие отличия среднего значения от максимального/минимального имеют признаки *fixed acidity*, *residual sugar*, *free sulfur dioxide*, *total sulfur dioxide* и *alcohol*. Это указывает на выбросы в данных столбцах. При этом наибольший разброс имеет *total sulfur dioxide*, в этом столбце содержатся аномально большие объекты.

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

После нормализации всех данных при помощи стандартного отклонения, был осуществлен поиск столбца с наибольшим средним значением. Программная реализация этого действия и его результат приведены на рисунке 5.1.

```
# определение номера столбца, соответствующего максимальному среднему  
значению  
print("Столбец с максимальным средним значением после нормировки через  
стандартное отклонение: ", np.argmax(np.mean(data, axis=0)))
```

Столбец с максимальным средним значением после нормировки через
стандартное отклонение: 7

Рисунок 5.1. Код для выделения столбца с максимальным средним значением после нормировки признаков через стандартное отклонение для набора *Red Wine Quality*.

Учитывая, что нумерация столбцов в датафрейме введется с нуля, было заключено, что столбец *density* (плотность) имеет наибольшее среднее значение. Результат оказался предсказуемым, так таблица 5.2 указывала на самый слабый из всех данных разброс для данного признака.

5. Столбец с целевым признаком? Сколько объектов попадает в тренировочную выборку при использовании train_test_split с параметрами test_size = 0.3, random_state = 42?

Первым действием целевой признак *quality* (качество вина) был выделен в переменную *target* и удален из датафрейма *data* с помощью метода *drop()*.

Далее набор данных был разделен на обучающую и тестовую выборки. Для этого была использована функция *train_test_split()* из библиотеки *scikit-learn*. Тестовая выборка составила 30% от всех данных. Значит в тренировочную выборку вошли 70% исходных данных (1119 объектов). Также был указан параметр *random_state=42* для того, чтобы данные не менялись при каждом запуске программы.

В результате были получены 4 переменные *X_train*, *X_test*, *y_train* и *y_test*.

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

Для вычисления коэффициента корреляции между каждой парой признаков можно воспользоваться функцией `corr()` в `pandas`. Результатом будет матрица корреляций, в которой каждый элемент показывает коэффициент корреляции между соответствующими признаками. Можно визуализировать матрицу корреляций в виде тепловой карты с помощью библиотеки `seaborn`. Построенная матрица приведена в приложении 5 на рисунке 5.2. Ниже приведен ее анализ.

Чтобы определить, какая корреляция находится между парами признаков, можно использовать следующее правило:

- Если значение коэффициента корреляции находится между -1 и -0.7 или между 0.7 и 1, то это указывает на сильную отрицательную или положительную корреляцию соответственно.
- Значения коэффициентов корреляции между -0.7 и -0.3 или между 0.3 и 0.7 указывают на умеренную отрицательную или положительную корреляцию соответственно.
- Если значение коэффициента корреляции находится между -0.3 и 0.3, то это указывает на отсутствие корреляции между признаками.

Таким образом для интересующего набора данных были сделаны следующие выводы:

1. Сильные линейные зависимости отсутствуют.
2. Между *fixed acidity* и *pH*, *volatile acidity* и *citric acid*, *volatile acidity* и *quality*, *citric acid* и *pH*, *density* и *pH*, *density* и *alcohol* имеются слабые отрицательные линейные зависимости.
3. Между *fixed acidity* и *citric acid*, *fixed acidity* и *density*, *citric acid* и *density*, *citric acid* и *sulphates*, *residual sugar* и *density*, *chlorides* и *sulphates*, *free sulfur dioxide* и *total sulfur dioxide*, *alcohol* и *quality* имеются слабые положительные линейные зависимости.
4. Между остальными парами признаков линейные зависимости отсутствуют.

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

Первым действием был создан экземпляр метода главных компонент с использованием конструктора `PCA()` из библиотеки `scikit-learn` и записан в переменную *pca*. Модель была подогнана при помощи обучающей выборки.

Далее с использованием поля `explained_variance_ratio_` был создан массив *variance_ratio*, хранящий доли объясненной дисперсии для каждой компоненты. С его помощью был получен массив кумулятивных сумм, в котором для *i*-ого элемента хранилась доля

объясненной дисперсии при помощи $i+1$ компонент. Таким образом было определено, что для объяснения 90% дисперсии достаточно 8 новых компонент. График зависимости доли объясненной дисперсии от числа компонент приведен в приложении 5 на рисунке 5.3.

8. Какой признак вносит наибольший вклад в первую компоненту?

Чтобы понять, какой признак вносит наибольший вклад в первую компоненту, можно использовать атрибут `components_` объекта PCA. Каждая строка массива, полученного с его помощью, соответствует главной компоненте, а каждый столбец соответствует вкладу признака в построение данной компоненты.

Таким образом, чтобы определить, какой признак вносит наибольший вклад в первую компоненту, было найден максимальное по модулю значение в первой строке массива `components_`.

В результате удалось определить, что первая компонента наиболее сильно зависит от признака `citric acid`.

9. Построить двухмерное представление данных с помощью алгоритма t-SNE. На сколько кластеров визуально, на ваш взгляд, разделяется выборка? Объяснить смысл кластеров.

Первым действием был создан экземпляр стохастического вложения соседей с t -распределением с использованием конструктора `TSNE()` из библиотеки `scikit-learn` и записан в переменную `tsne`. Параметры алгоритма были подобраны вручную таким образом, чтобы кластеры можно было разделить визуально. Модель была подогнана при помощи тестовой выборки. Полученное двумерное представление данных приведено на рисунке 5.4. Отделенные визуально кластеры показаны на рисунке 5.5.

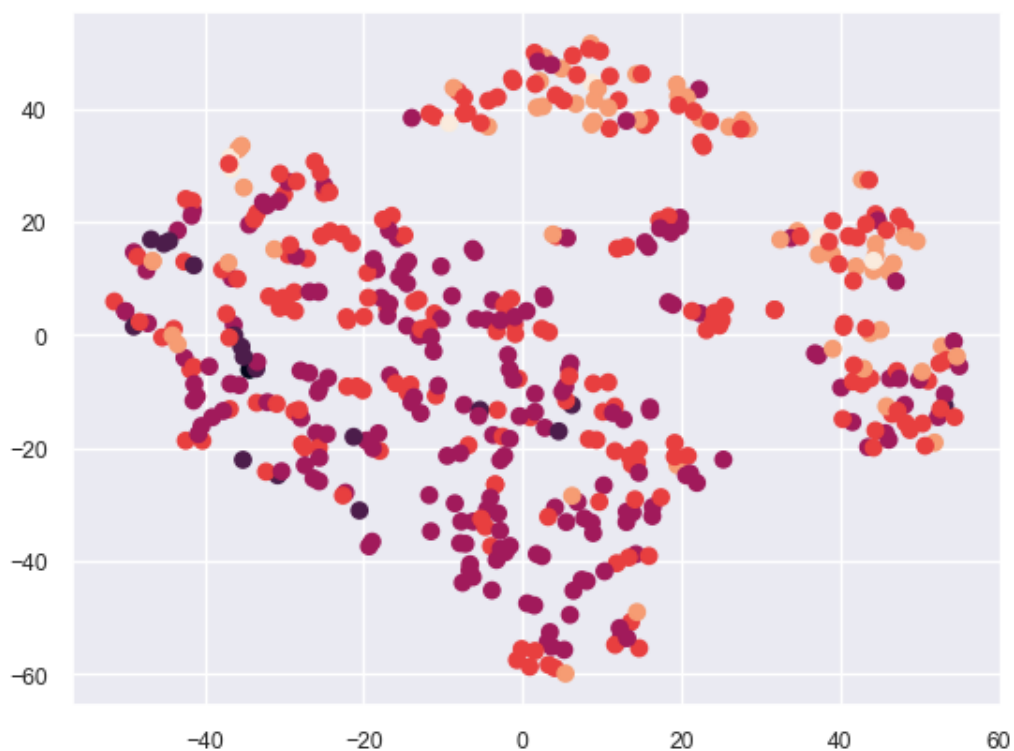


Рисунок 5.3. Двухмерное представление данных набора *Red Wine Quality* с помощью алгоритма t-SNE.

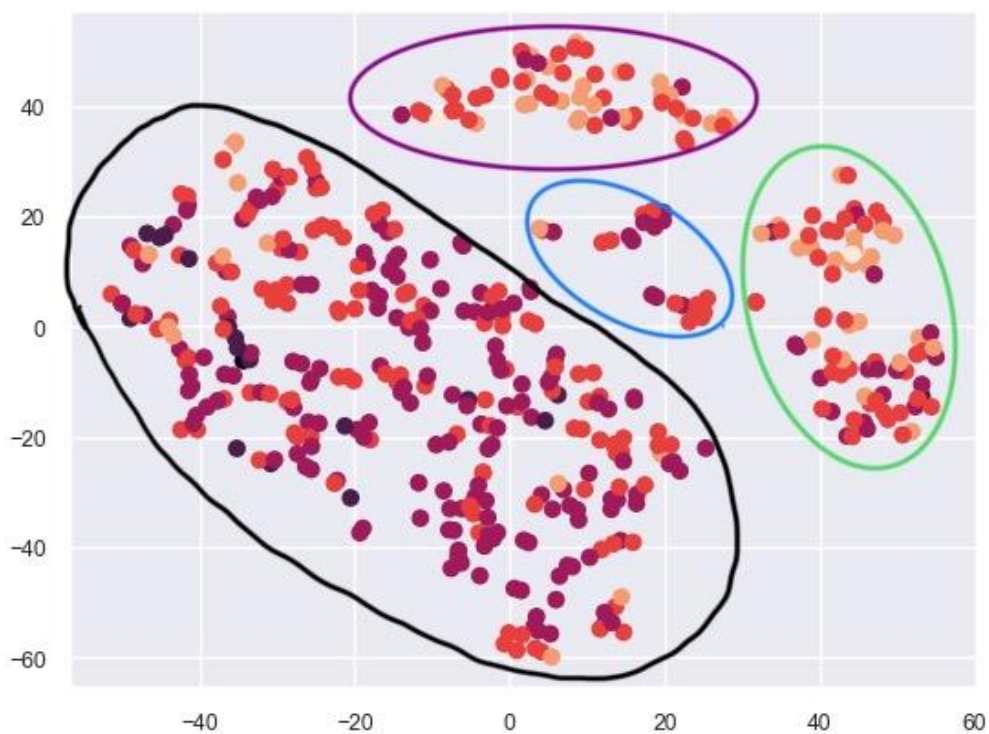


Рисунок 5.3. Двухмерное представление данных набора *Red Wine Quality* с помощью алгоритма t-SNE с выделением кластеров путем визуального анализа.

Здесь кластеры – это группы объектов, которые имеют схожие признаковые описания. Кластеризация может помочь выявить скрытые закономерности в данных, облегчить понимание данных и дать возможность сделать выводы о взаимосвязях между признаками и объектами. Визуально выделяющиеся кластеры могут свидетельствовать о наличии существенных различий между группами объектов и помочь в дальнейшем анализе данных.

На данном распределении отчетливо видны, как минимум, 4 кластера. Два из них очевидны - черный (плохое вино, его больше) и стоящий от него на значительном расстоянии зеленый (хорошее вино). Другие два кластера, как по мне, демонстрируют неожиданно плохое и неожиданно хорошее вино (фиолетовый - неожиданно плохое, синий - неожиданно хорошее). То есть те вина, которые, несмотря на показатели, получили оценку, противоположную присущей данным показателям.

Однако такая оценка не может считаться качественной. Плотность точек в каждом кластере низкая, а сами кластеры расположены слишком близко друг к другу, что говорит о невозможности точного проведения кластеризации таким методом. Например, фиолетовый, синий и зеленый кластеры можно объединить в один, и разделять вина только на хорошие или плохие. Также сами кластеры с большой долей вероятности определены неверно (например, в зеленый кластер могут быть собраны крепкие вина с высоким содержанием лимонной кислоты). Работа с методами классификации для данного набора будет более подробно расписана в задаче №6.

Выводы

- Набор данных *Red Wine Quality* был успешно загружен. Число объектов в нем составило 1599, а признаков 12. Также было определено, что набор не содержит пропущенных значений. Для каждого признака из этого набора было приведено описание и некоторые статистические характеристики, такие как количество уникальных значений и выбросы, оцененные в первом приближении. Столбцом, который точно содержит аномальные значения, оказался столбец *total sulfur dioxide*.
- В более детальном анализе было определено, что признак *density* (плотность) имеет наибольшее среднее значение после нормализации данных через стандартное отклонение. Все признаки в датафрейме оказались числовыми. Столбец *quality* может являться категориальным признаком, так как имеет всего 6 уникальных значений. Из логических соображений этот признак был определен как целевой (конечная оценка качества вина). Было проведено исследование корреляции признаков с помощью построения тепловой карты матрицы корреляций. Оно показало, что сильных корреляций в наборе данных не наблюдается.
- При помощи метода главных компонент удалось снизить размерность с 11 признаков до 7, которые объясняли более 90% дисперсии. Для первой компоненты был определен признак, вносящий наибольший вклад в ее построение. Этим признаком оказалось *содержание лимонной кислоты*.

- При помощи алгоритма TSNE данные удалось визуально разделить на 4 кластера. Однако сами кластеры оказались недостаточно плотными и расположенными слишком близко друг к другу, что говорит о невозможности кластеризации данных этим методом.

Код решения задачи и сведения о проверенных моделях приведены в Приложении 5.

Задача №6

Необходимо загрузить данные из указанного набора и произвести их свободный анализ.

Набор данных: *Red Wine Quality*.

Постановка задачи: построить модели предсказания качества, оценить вклад каждого компонента. Построить статистические оценки параметров и гистограммы распределений. Построить бинарный классификатор: хорошее/плохое вино. Выделить аномальные объекты.

1. Загрузка данных и подготовка к дальнейшему анализу.

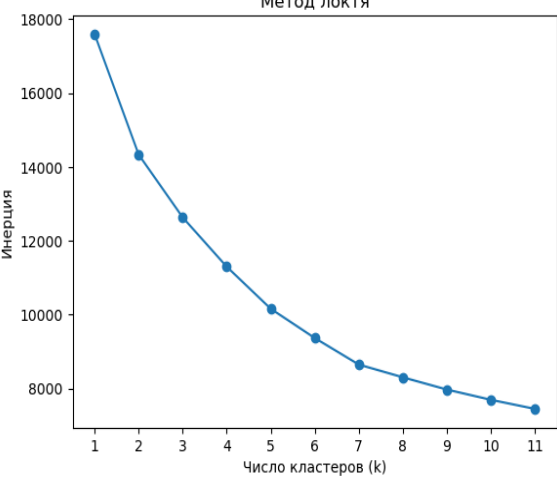
Аналогично задаче №5, набор данных *Red Wine Quality* был загружен в датафрейм *data*. Число объектов в нем оказалось равно 1599, а признаков 12. Также было определено, что набор не содержит пропущенных значений. Целевой признак *quality* был записан в переменную *target0*. Все признаки были нормализованы через стандартное отклонение, нормализованное значение целевой переменной было записано в переменную *target* и удалено из датафрейма. Таким образом данные были подготовлены к анализу.

2. Построение и оценка моделей классификации вин для данного набора. Выбор оптимального подхода и оценка вклада каждой компоненты.

Требовалось определить оптимальный метод анализа данных, который позволял бы точно предсказывать качество вина в зависимости от его химических характеристик. Из изученных методов для данной задачи лучше всего кластеризация. Она позволила бы точно сгруппировать признаки и помочь в точных предсказаниях. Далее приведен анализ классификации как основного метода решения задачи.

Для оценки возможности применения кластеризации к набору данных *Red Wine Quality* на первом шаге был использован метод локтя. Код применения этого метода и его наглядное представление продемонстрированы на рисунке 6.1 и 6.2 соответственно в таблице 6.1.

Таблица 6.1. Код применения метода локтя и его наглядное представление для набора данных *Red Wine Quality*.

<pre>import matplotlib.pyplot as plt from sklearn.cluster import KMeans from sklearn import metrics # Создание списка для сохранения значения инерции (сумма квадратов # расстояний до ближайшего центроида) для разных значений k inertia = [] # Задание разных значений k k_values = range(1, 12) # Вычисление инерции для каждого значения k for k in k_values: kmeans = KMeans(n_clusters=k, random_state=42) kmeans.fit(data) inertia.append(kmeans.inertia_) # Построение графика локтя plt.plot(k_values, inertia, marker='o') plt.xlabel('Число кластеров (k)') plt.ylabel('Инерция') plt.title('Метод локтя') plt.show()</pre>	 <table border="1"> <caption>Данные для графика метода локтя</caption> <thead> <tr> <th>Число кластеров (k)</th> <th>Инерция</th> </tr> </thead> <tbody> <tr><td>1</td><td>17500</td></tr> <tr><td>2</td><td>14500</td></tr> <tr><td>3</td><td>12800</td></tr> <tr><td>4</td><td>11500</td></tr> <tr><td>5</td><td>10500</td></tr> <tr><td>6</td><td>9500</td></tr> <tr><td>7</td><td>8800</td></tr> <tr><td>8</td><td>8300</td></tr> <tr><td>9</td><td>8000</td></tr> <tr><td>10</td><td>7800</td></tr> <tr><td>11</td><td>7500</td></tr> </tbody> </table>	Число кластеров (k)	Инерция	1	17500	2	14500	3	12800	4	11500	5	10500	6	9500	7	8800	8	8300	9	8000	10	7800	11	7500
Число кластеров (k)	Инерция																								
1	17500																								
2	14500																								
3	12800																								
4	11500																								
5	10500																								
6	9500																								
7	8800																								
8	8300																								
9	8000																								
10	7800																								
11	7500																								
<p>Рисунок 6.1. Код для проведения анализа кластеризации путем применения метода локтя для набора данных <i>Red Wine Quality</i>.</p>	<p>Рисунок 6.2. Визуализация метода локтя для набора данных <i>Red Wine Quality</i>.</p>																								

Видно, что график образует почти идеальную дугу, без четкого "локтя", что может означать, что изменение числа кластеров не сильно влияет на инерцию. В таком случае, использование метода локтя может быть не информативным.

В сложившейся ситуации может быть полезным рассмотреть другие методы для определения оптимального числа кластеров, например метод силуэта. Его использование будет описано далее.

Метод силуэта используется для оценки качества кластеризации и помогает определить оптимальное число кластеров. Он основан на вычислении силуэтных коэффициентов для каждого объекта в выборке. При этом, чем ближе значение силуэтного коэффициента к 1, тем лучше кластеризация. Код, реализующий данный метод, приведен на рисунке 6.3.

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Проведение кластеризации для разного числа кластеров
for n_clusters in range(2, 12):
    kmeans = KMeans(n_clusters=n_clusters)
    cluster_labels = kmeans.fit_predict(data)

    # Вычисление среднего силуэтного коэффициента
    silhouette_avg = silhouette_score(data, cluster_labels)
    print("Если число кластеров равно =", n_clusters, ", то среднее значение силуэтных коэффициентов =", silhouette_avg)
```

Рисунок 6.3. Код для проведения анализа кластеризации путем применения метода силуэтов для набора данных *Red Wine Quality*.

Таким образом было определено, что

- Если число кластеров равно = 2, то среднее значение силуэтных коэффициентов = 0.21;
- Если число кластеров равно = 3, то среднее значение силуэтных коэффициентов = 0.18;
- Если число кластеров равно = 4, то среднее значение силуэтных коэффициентов = 0.21;
- Если число кластеров равно = 5, то среднее значение силуэтных коэффициентов = 0.19;
- Если число кластеров равно = 6, то среднее значение силуэтных коэффициентов = 0.19;
- Если число кластеров равно = 7, то среднее значение силуэтных коэффициентов = 0.19;
- Если число кластеров равно = 8, то среднее значение силуэтных коэффициентов = 0.15;
- Если число кластеров равно = 9, то среднее значение силуэтных коэффициентов = 0.15;
- Если число кластеров равно = 10, то среднее значение силуэтных коэффициентов = 0.16;
- Если число кластеров равно = 11, то среднее значение силуэтных коэффициентов = 0.13.

Из полученных значений видим, что все коэффициенты малы, а оптимальное значение достигается лишь при минимальном числе кластеров (предположительно хорошее/плохое вино).

Результаты, полученные при помощи метода локтя и метода силуэтов, говорят о том, что использование кластеризации нецелесообразно для анализа данной выборки и для классификации данных следует использовать другие методы.

При предсказании качества вина хотелось бы получать его точную оценку. Этого можно добиться, применяя мультиклассовую классификацию. Для выполнения этой задачи было принято решение использовать классификатор логистической регрессии. Его реализация приведена далее.

На первом шаге набор данных был разделен на обучающую и тестовую выборки. Для этого была использована функция `train_test_split()` из библиотеки `scikit-learn`. Тестовая выборка составила 30% от всех данных. Также был указан параметр `random_state=42` для того, чтобы данные не менялись при каждом запуске программы.

Далее был создан классификатор метода логистической регрессии. Для этого использовался класс `LogisticRegression` из библиотеки `scikit-learn`. Он был обучен на переменных `X_train` и `y_train`. Для набора `X_test` были получены предсказания классификатора и записаны в переменную `y_pred`.

Оценка классификатора производилась по матрице классов (функция `classification_report` библиотеки `scikit-learn`). Она предоставила различные метрики, такие как точность (`precision`), полнота (`recall`), F1-мера (`F1-score`) и поддержка (`support`) для каждого класса. Поддержка представляет собой количество примеров в каждом классе. Результат представлен в таблице 6.2.

Таблица 6.2. Статистические характеристики мультиклассового классификатора `LogisticRegression` для набора данных *Red Wine Quality*.

Класс	precision	recall	f1-score	support
3	0.00	0.00	0.00	1
4	0.00	0.00	0.00	17
5	0.62	0.75	0.68	195
6	0.53	0.55	0.54	200
7	0.42	0.26	0.32	61
8	0.00	0.00	0.00	6
accuracy			0.56	480
macro avg	0.26	0.26	0.26	480
weighted avg	0.53	0.56	0.54	480

Полученные данные говорят о следующем:

- В наборе данных не содержатся вина с оценкой ниже 3 или выше 8.
- Так как в выборке присутствует слишком мало вин с оценками 3, 4, 8, мультиклассовому классификатору не удастся предсказать эти оценки (ни одно такое вино он не определил в нужный класс).
- Лучше всего предсказываются вина среднего качества (оценка 5 и 6). В выборке их содержится больше всего.
- `macro avg` - среднее значение метрик (точность, полнота и F1-мера) по всем классам составило 0.26, что критически мало для построения предсказаний для разных классов.

На основании полученных данных была отвергнута гипотеза о том, что по данному набору можно точно предсказать оценку качества вина.

Единственным рациональным вариантом, оставшимся для рассмотрения, является бинарный классификатор. С его помощью определялись вина премиум класса. Такими будем считать вина с максимальными оценками (7-8). Они составляют менее 25% от всех данных.

Для выполнения задачи переменная *target* была преобразована к бинарной (1 – если вино премиум класса и 0 в противном случае). С учетом этого, данные вновь были разбиты на обучающую и тестовую выборки.

После этого был создан классификатор случайного леса. При помощи алгоритма GridSearch было определено лучшее число деревьев для данного набора с точностью до 10. Оно оказалось равно 280.

После подгонки классификатора с лучшим числом деревьев и получения предсказаний, модель была оценена. Для этого использовались метрики *precision*, *recall* и *f1*. Были получены значения:

- Precision-score: 0.74
- Recall-score: 0.53
- F1-score: 0.62

Точность классификатора составила приемлемые для этого набора данных 62%, с его помощью можно отделять премиум вина от не премиум вин. Также заметно высокое значение метрики *precision*. Это говорит о точности классификатора в предсказании вин премиум класса.

Найденная модель может считаться качественной, поэтому было принято решение исследовать вклад каждой компоненты в качество вина с ее помощью. Для этого использовался атрибут *feature_importances_*, который возвращает массив долей вкладов каждой компоненты в предсказания модели. Так, удалось определить, что

- Вклад компоненты *alcohol* в предсказания: 0.17
- Вклад компоненты *sulphates* в предсказания: 0.12
- Вклад компоненты *volatile acidity* в предсказания: 0.11
- Вклад компоненты *density* в предсказания: 0.09
- Вклад компоненты *citric acid* в предсказания: 0.09
- Вклад компоненты *total sulfur dioxide* в предсказания: 0.08
- Вклад компоненты *fixed acidity* в предсказания: 0.07
- Вклад компоненты *chlorides* в предсказания: 0.07
- Вклад компоненты *residual sugar* в предсказания: 0.07
- Вклад компоненты *free sulfur dioxide* в предсказания: 0.06
- Вклад компоненты *pH* в предсказания: 0.06

Полученные данные отсортированы в обратном порядке.

Таким образом был получен вклад каждой компоненты в качество вина. Для прослеживания зависимости качества вина от его химических свойств оставалось определить знак каждой компоненты (положительное или отрицательное влияние на качество) и исключить из рассмотрения незначимые признаки.

3. Оценка статистических параметров признаков. Гистограммы распределений.

Для более точного анализа было принято решение подробнее рассмотреть каждую компоненту. Для этого была написана функция `plot_feature(X, feature, y)`, которая принимает столбец признака X , его название $feature$ и столбец целевой переменной y . По этим данным функция строит гистограмму распределения исследуемого признака, его выбросы в каждом классе целевой переменной (при помощи ящика с усами) и статистические характеристики, необходимы для дополнения информации, полученной на предыдущем этапе. Характеристики берутся из линейной регрессии вида $y \sim X$, использующийся в решении исключительно для их получения. Далее эта функция используется для каждого признака (в порядке значимости компонент).

1. Рассмотрим компоненту *alcohol*. Ее гистограмма распределения (рисунок 6.4) и ящики с усами (рисунок 6.5) приведены в таблице 6.3. Статистические параметры данного регрессора приведены в таблице 6.4.

Таблица 6.3. Результат выполнения функции `plot_feature()` для компоненты *alcohol*.

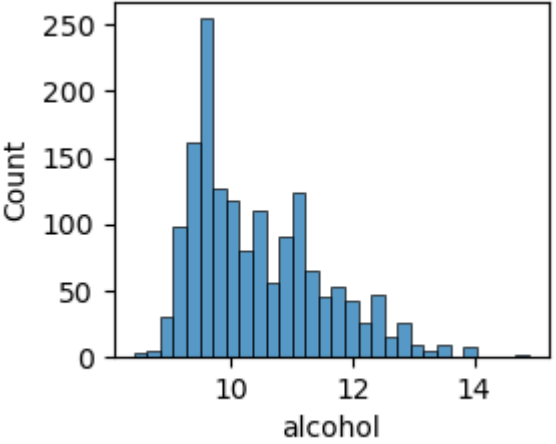
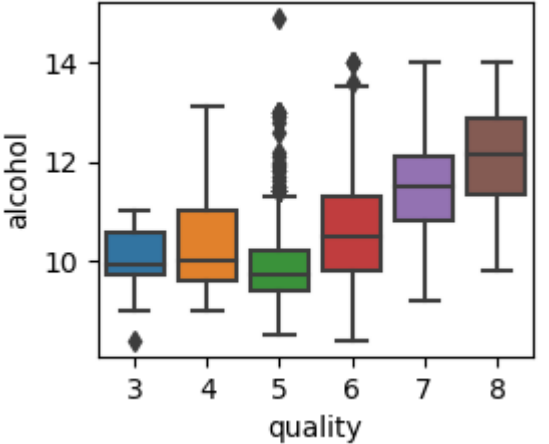
	
<p>Рисунок 6.4. Гистограмма распределения для признака <i>alcohol</i> в наборе данных <i>Red Wine Quality</i>.</p>	<p>Рисунок 6.5. Ящики с усами переменной <i>alcohol</i> для каждого класса качества в наборе данных <i>Red Wine Quality</i>.</p>

Таблица 6.4. Статистические характеристики регрессора *alcohol* в модели парной регрессии $quality \sim alcohol$.

	Коэффициент	Стандартная ошибка	t-значение	P-значение	Доверительный интервал
alcohol	0.476166	0.022005	21.639478	2.8314e-91	[0.04316, 0.51933]

По данным гистограммы видно, что вина низкой крепости преобладают над крепкими в наборе данных. Ящики с усами показывают заметное число выбросов в 5ой и 6ой категориях. Это значит, что не все крепкие вина можно отнести к премиальным. Крепкие вина редки, а их качество оценено высоко. На это указывает положительный коэффициент признака *alcohol* и восходящая зависимость на графике ящиков с усами. Доверительный интервал не включает 0, что говорит о значимости этого признака. Таким образом, между признаком *alcohol* и качеством вина имеется сильная восходящая зависимость.

2. Рассмотрим компоненту *sulphates*. Ее гистограмма распределения (рисунок 6.6) и ящики с усами (рисунок 6.7) приведены в таблице 6.5. Статистические параметры данного регрессора приведены в таблице 6.6.

Таблица 6.5. Результат выполнения функции *plot_feature()* для компоненты *sulphates*.

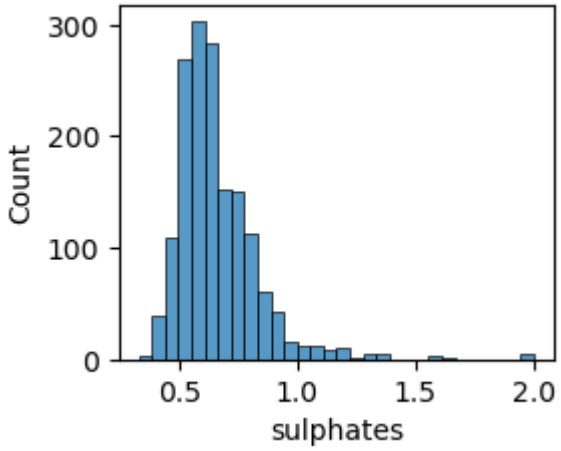
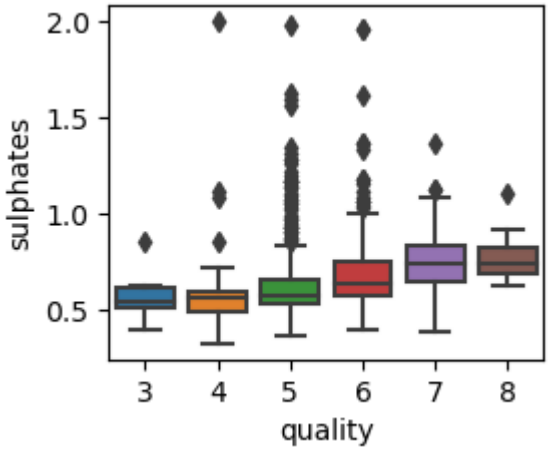
	
<p>Рисунок 6.6. Гистограмма распределения для признака <i>sulphates</i> в наборе данных <i>Red Wine Quality</i>.</p>	<p>Рисунок 6.7. Ящики с усами переменной <i>sulphates</i> для каждого класса качества в наборе данных <i>Red Wine Quality</i>.</p>

Таблица 6.6. Статистические характеристики регрессора *sulphates* в модели парной регрессии *quality ~ sulphates*.

	Коэффициент	Стандартная ошибка	t-значение	P-значение	Доверительный интервал
sulphates	0.251397	0.02422	10.379809	1.8020e-24	[0.04751, 0.29890]

По данным гистограммы видно, что вина с низким содержанием сульфатов преобладают над винами с высоким содержанием сульфатов в наборе данных. В каждом классе заметны большие количества выбросов, однако в классах 7 и 8 их не так много. Это указывает на то, что в винах не премиального качества эта добавка может встречаться случайным образом и не влиять на оценку. Вина с этой добавкой (когда она добавлена намеренно и в правильной пропорции) редки, а их качество оценено высоко. На это указывает положительный коэффициент признака *sulphates* и восходящая зависимость на графике ящиков с усами. Доверительный интервал не включает 0, что говорит о значимости этого признака. Таким образом, между признаком *sulphates* и качеством вина имеется сильная восходящая зависимость.

3. Рассмотрим компоненту *volatile acidity*. Ее гистограмма распределения (рисунок 6.8) и ящики с усами (рисунок 6.9) приведены в таблице 6.7. Статистические параметры данного регрессора приведены в таблице 6.8.

Таблица 6.7. Результат выполнения функции *plot_feature()* для компоненты *volatile acidity*.

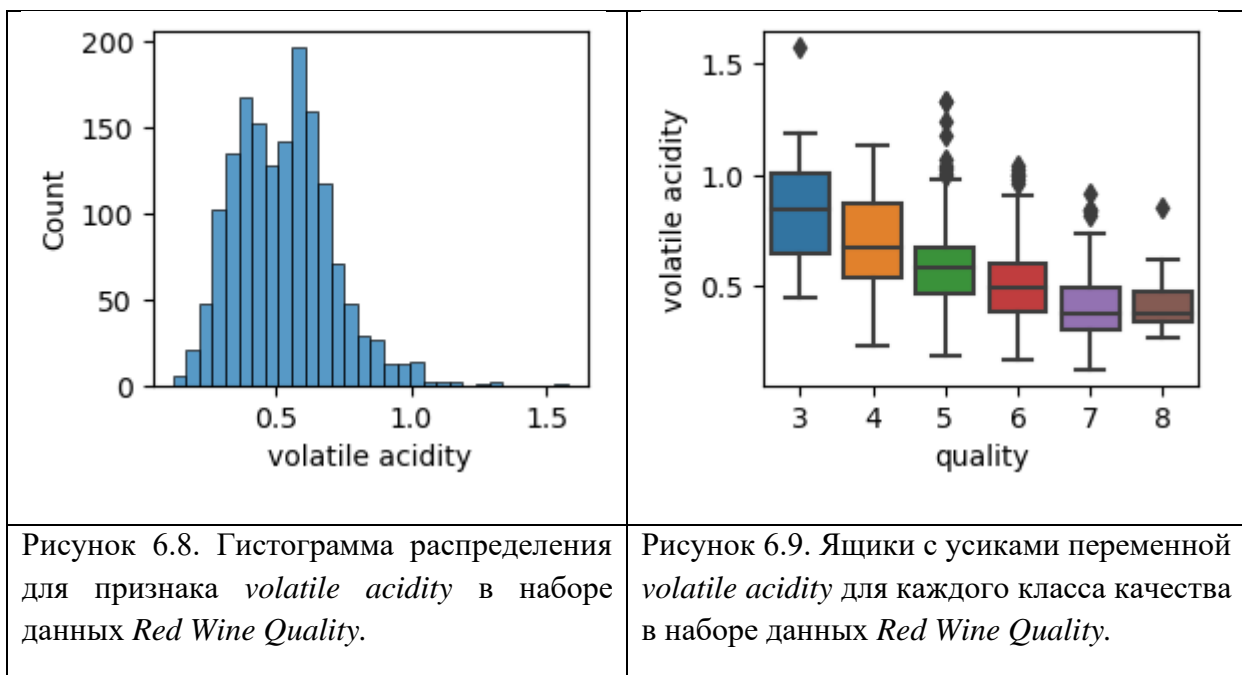


Таблица 6.8. Статистические характеристики регрессора *volatile acidity* в модели парной регрессии *quality ~ volatile acidity*.

	Коэффициент	Стандартная ошибка	t-значение	P-значение	Доверительный интервал

volatile acidity	-0.390558	0.023036	-16.9541	2.0517e-59	[0.04518, 0.34537]
------------------	-----------	----------	----------	------------	--------------------

По данным гистограммы видно относительно нормальное распределение вин с различным содержанием летучих кислот. В каждом классе заметны небольшие количества выбросов, однако все они расположены плотно и все еще могут указывать на принадлежность к какому-либо классу (кроме аномально высоких значений, которые не поддаются объяснению). Вина с высоким содержанием летучих кислот имеют неприятный запах, из-за чего их оценка не может быть высокой. На это также указывает отрицательный коэффициент признака *volatile acidity* и нисходящая зависимость на графике ящиков с усами. Доверительный интервал не включает 0, что говорит о значимости этого признака. Таким образом, между признаком *volatile acidity* и качеством вина имеется сильная нисходящая зависимость.

- Рассмотрим компоненту *density*. Ее гистограмма распределения (рисунок 6.10) и ящики с усами (рисунок 6.11) приведены в таблице 6.9. Статистические параметры данного регрессора приведены в таблице 6.10.

Таблица 6.9. Результат выполнения функции *plot_feature()* для компоненты *density*.

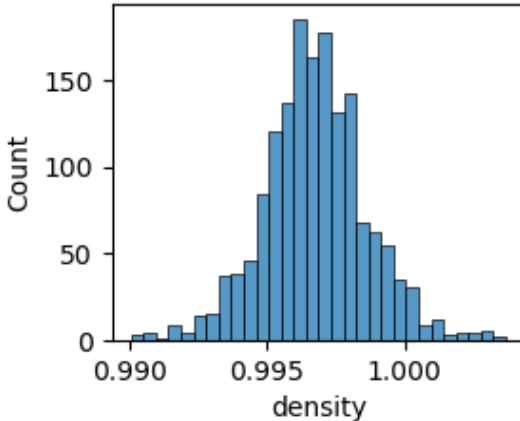
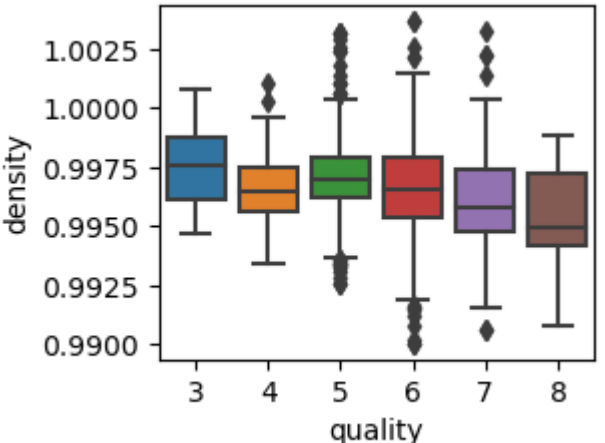
	
<p>Рисунок 6.10. Гистограмма распределения для признака <i>density</i> в наборе данных <i>Red Wine Quality</i>.</p>	<p>Рисунок 6.11. Ящики с усами переменной <i>density</i> для каждого класса качества в наборе данных <i>Red Wine Quality</i>.</p>

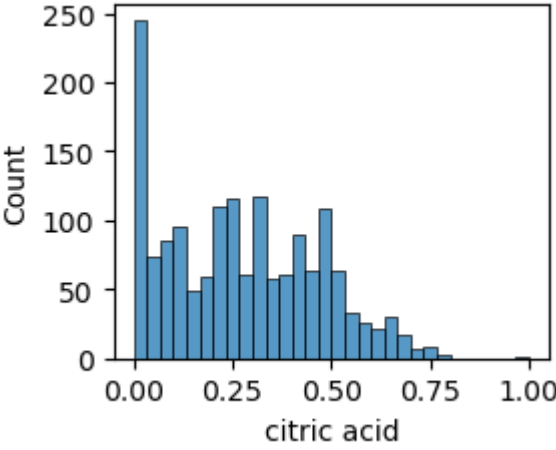
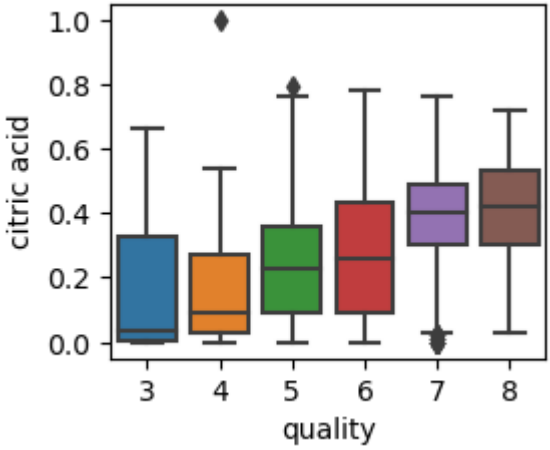
Таблица 6.10. Статистические характеристики регрессора *density* в модели парной регрессии *quality ~ density*.

	Коэффициент	Стандартная ошибка	t-значение	P-значение	Доверительный интервал
density	-0.174919	0.024638	-7.09966	1.8749e-12	[0.04833, 0.12659]

По данным гистограммы видно нормальное распределение вин с различной плотностью. Это объясняет наличие плотных выбросов у ящиков с усиками (в виду большого количества данных в каждом классе). Премиальные вина почти не имеют выбросов по плотности. Вина с высокой плотностью более вязкие и тяжелые, в больших количествах их употребление затруднительно, из-за чего их оценка может быть невысокой. На это также указывает отрицательный коэффициент признака *density* и нисходящая зависимость на графике ящиков с усиками. Доверительный интервал не включает 0, что говорит о значимости этого признака. Таким образом, между признаком *density* и качеством вина имеется заметная нисходящая зависимость.

- Рассмотрим компоненту *citric acid*. Ее гистограмма распределения (рисунок 6.12) и ящики с усиками (рисунок 6.13) приведены в таблице 6.11. Статистические параметры данного регрессора приведены в таблице 6.12.

Таблица 6.11. Результат выполнения функции *plot_feature()* для компоненты *citric acid*.

	
<p>Рисунок 6.12. Гистограмма распределения для признака <i>citric acid</i> в наборе данных <i>Red Wine Quality</i>.</p>	<p>Рисунок 6.13. Ящики с усиками переменной <i>citric acid</i> для каждого класса</p>

	качества в наборе данных <i>Red Wine Quality</i> .
--	--

Таблица 6.12. Статистические характеристики регрессора *citric acid* в модели парной регрессии $quality \sim citric\ acid$.

	Коэффициент	Стандартная ошибка	t-значение	P-значение	Доверительный интервал
citric acid	0.226373	0.024374	9.287504	4.99e-20	[0.04781, 0.27418]

По данным гистограммы видно, что вина, в которых не содержится лимонная кислота, и вина с низким ее содержанием сильно преобладают над винами с высоким содержанием лимонной кислоты. Классы практически не содержат выбросов. Вина с этой добавкой обладают пикантностью и душистостью. Те вина, в которые эта добавка была добавлена привильно (так, чтобы в итоге вкусы ощущались в резонансе) редки, а их качество оценено высоко. На это указывает положительный коэффициент признака *citric acid* и восходящая зависимость на графике ящичков с усиками. Доверительный интервал не включает 0, что говорит о значимости этого признака. Таким образом, между признаком *citric acid* и качеством вина имеется заметная восходящая зависимость.

- Рассмотрим компоненту *total sulfur dioxide*. Ее гистограмма распределения (рисунок 6.14) и ящички с усиками (рисунок 6.15) приведены в таблице 6.13. Статистические параметры данного регрессора приведены в таблице 6.14.

Таблица 6.13. Результат выполнения функции *plot_feature()* для компоненты *total sulfur dioxide*.

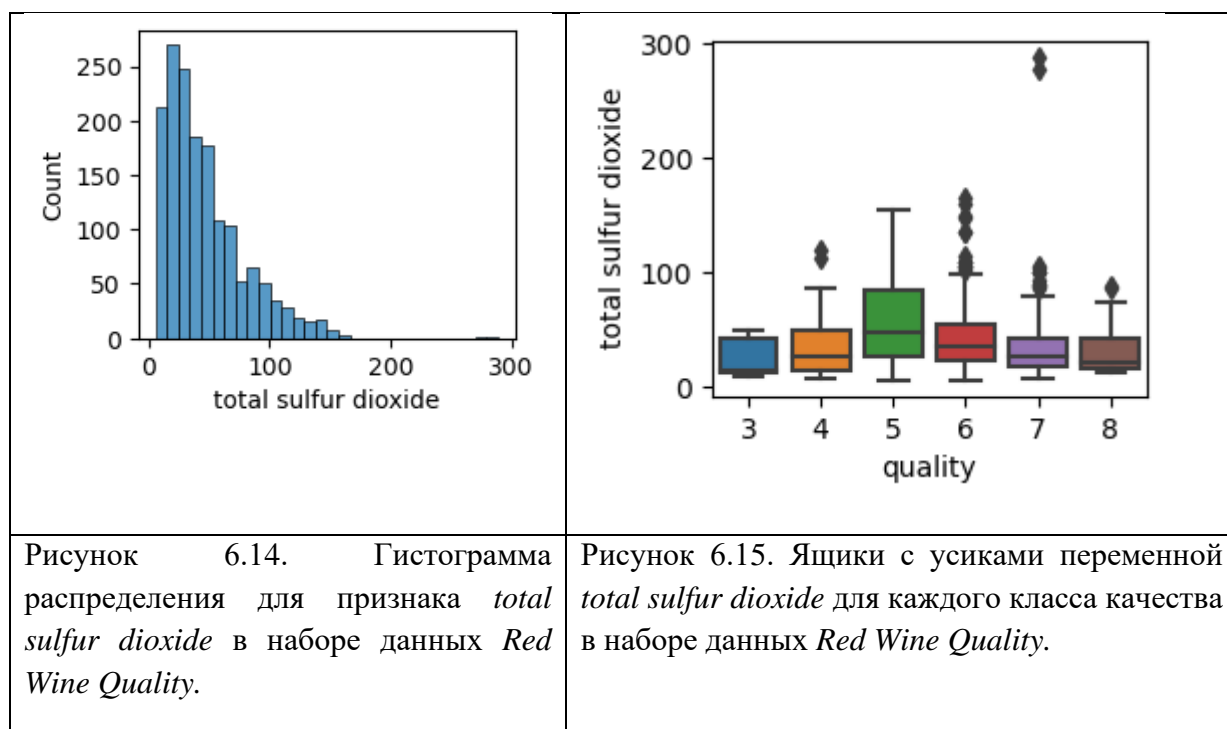


Таблица 6.14. Статистические характеристики регрессора *total sulfur dioxide* в модели парной регрессии *quality ~ total sulfur dioxide*.

	Коэффициент	Стандартная ошибка	t-значение	P-значение	Доверительный интервал
total sulfur dioxide	-0.1851	0.024591	-7.527139	8.6217e-14	[0.04823, 0.13687]

По данным гистограммы видно, что вин с низким содержанием общего диоксида серы в наборе данных больше всего. Это объясняет наличие выбросов у ящиков с усами, ведь все высокие значения этого показателя считаются аномальными. Этот компонент вина предотвращает развитие бактерий и окисление, однако может негативно сказываться на вкусе и в итоге терять баллы оценки. На это также указывает отрицательный коэффициент признака *total sulfur dioxide* и слабо заметная нисходящая зависимость на графике ящиков с усами. Доверительный интервал не включает 0, что говорит о значимости этого признака. Таким образом, между признаком *total sulfur dioxide* и качеством вина имеется слабая нисходящая зависимость.

7. Рассмотрим компоненту *fixed acidity*. Ее гистограмма распределения (рисунок 6.16) и ящики с усами (рисунок 6.17) приведены в таблице 6.15. Статистические параметры данного регрессора приведены в таблице 6.16.

Таблица 6.15. Результат выполнения функции *plot_feature()* для компоненты *fixed acidity*.

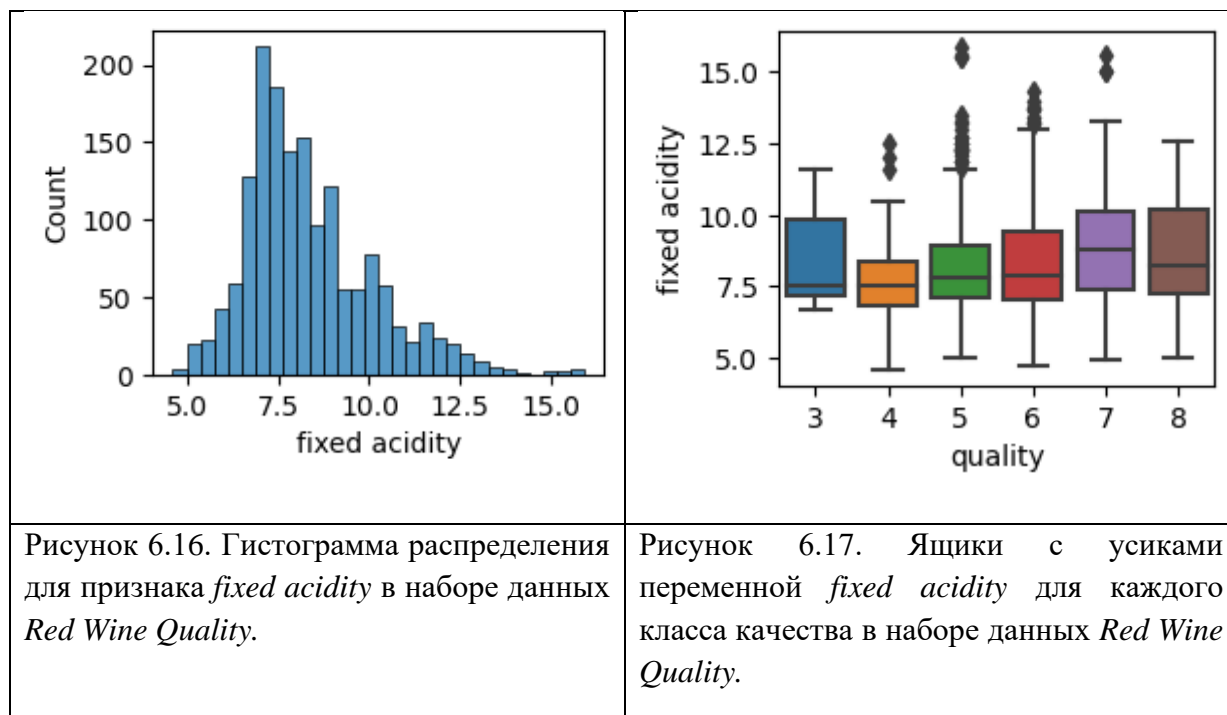


Таблица 6.16. Статистические характеристики регрессора *fixed acidity* в модели парной регрессии *quality ~ fixed acidity*.

	Коэффициент	Стандартная ошибка	t-значение	P-значение	Доверительный интервал
fixed acidity	0.124052	0.02483	4.996002	6.4956e-07	[0.04870, 0.17275]

По данным гистограммы видно, что вина, в которых фиксированные кислоты содержатся в малом или среднем количестве, преобладают над винами с высоким содержанием этого компонента. Все классы кроме крайних содержат выбросы, а больше всего их у оценки 5. По данным графика тяжело определить восходящую зависимость, на которую указывает положительный коэффициент признака и доверительный интервал, не включающий 0, ведь P-значение относительно низкое. Действительно, кислые вина тяжело разбивать на классы в виду того, что они могут быть кислыми по разным причинам. Точно можно сказать, что лучшие вина содержат умеренное количество кислоты, которое идет на пользу вкусу и не затмевает сладость и аромат винограда. Исходя из всего вышеперечисленного

можно заключить, что между признаком *fixed acidity* и качеством вина имеется слабая восходящая зависимость.

8. Рассмотрим компоненту *chlorides*. Ее гистограмма распределения (рисунок 6.18) и ящики с усами (рисунок 6.19) приведены в таблице 6.17. Статистические параметры данного регрессора приведены в таблице 6.18.

Таблица 6.17. Результат выполнения функции *plot_feature()* для компоненты *chlorides*.

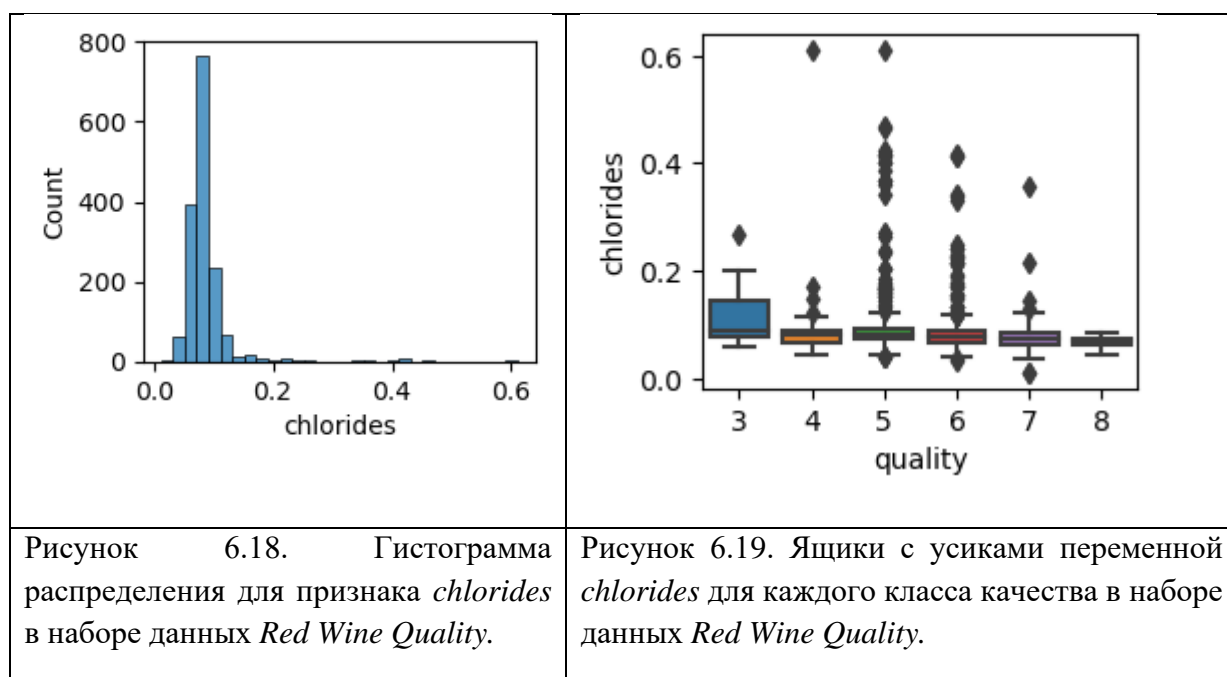


Таблица 6.18. Статистические характеристики регрессора *chlorides* в модели парной регрессии *quality ~ chlorides*.

	Коэффициент	Стандартная ошибка	t-значение	P-значение	Доверительный интервал
chlorides	-0.128907	0.024815	-5.194767	2.3133e-07	[-0.08023, 0.0487]

По данным гистограммы видно, что подавляющую часть составляют вина с низким содержанием солей. Другое количество соли расценивается как аномальное и приводит к выбросам почти во всех классах. Действительно тяжело найти соленое вино. По данным графиков очень тяжело отследить какую-либо зависимость, на которую указывает отрицательный коэффициент. Доверительный интервал компоненты *chlorides* включает 0,

а его Р-значение высоко. Таким образом, между признаком *chlorides* и качеством вина отсутствует зависимость, и он не влияет на конечную оценку.

9. Рассмотрим компоненту *residual sugar*. Ее гистограмма распределения (рисунок 6.20) и ящики с усами (рисунок 6.21) приведены в таблице 6.19. Статистические параметры данного регрессора приведены в таблице 6.20.

Таблица 6.19. Результат выполнения функции *plot_feature()* для компоненты *residual sugar*.

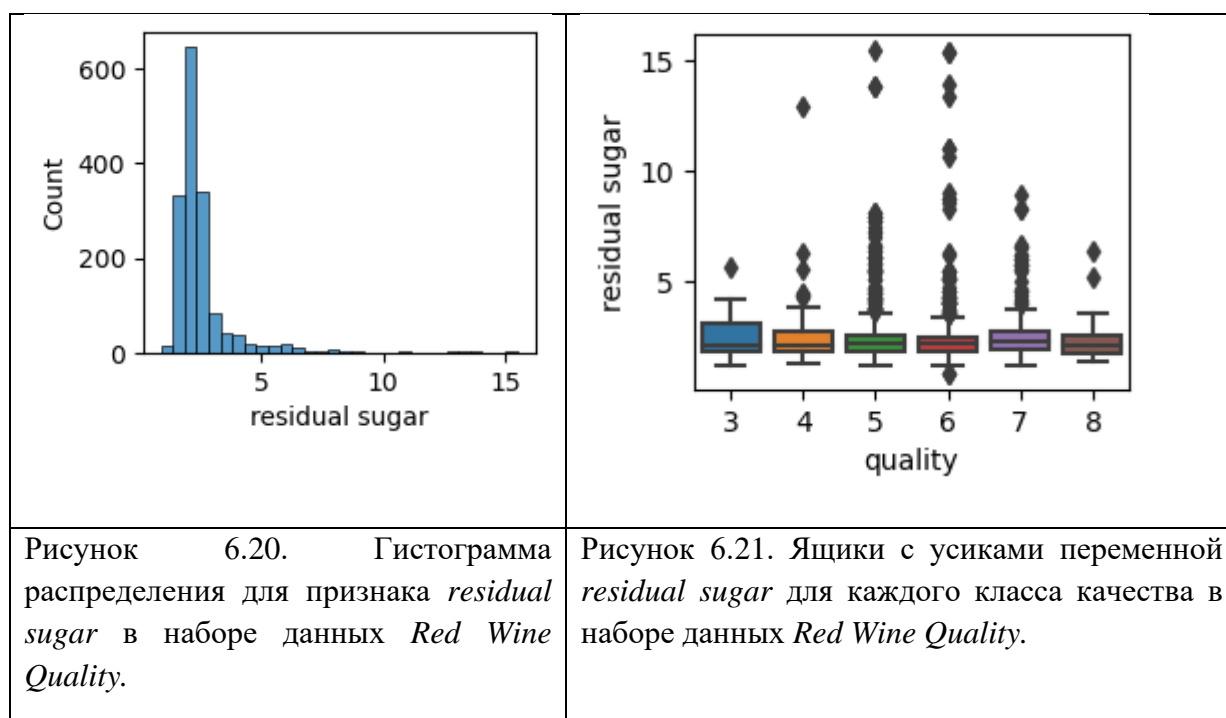


Таблица 6.20. Статистические характеристики регрессора *residual sugar* в модели парной регрессии *quality ~ residual sugar*.

	Коэффициент	Стандартная ошибка	t-значение	P-значение	Доверительный интервал
residual sugar	0.013732	0.025021	0.548802	0.583218	[-0.04908, 0.0628]

По данным гистограммы видно, что подавляющую часть составляют вина с низким содержанием остаточного сахара. Это показывает, что в выборке принимали участие в основном сухие и полусухие вина. Такое высокое количество схожих вин привело к выбросам во всех классах. По данным графиков очень тяжело отследить какую-либо зависимость, на которую указывает положительный коэффициент. Доверительный

интервал компоненты *residual sugar* включает 0, а его Р-значение очень высоко. Таким образом, между признаком *residual sugar* и качеством вина отсутствует зависимость, и он не влияет на конечную оценку для данного набора.

10. Рассмотрим компоненту *free sulfur dioxide*. Ее гистограмма распределения (рисунок 6.22) и ящики с усами (рисунок 6.23) приведены в таблице 6.21. Статистические параметры данного регрессора приведены в таблице 6.22.

Таблица 6.21. Результат выполнения функции *plot_feature()* для компоненты *free sulfur dioxide*.

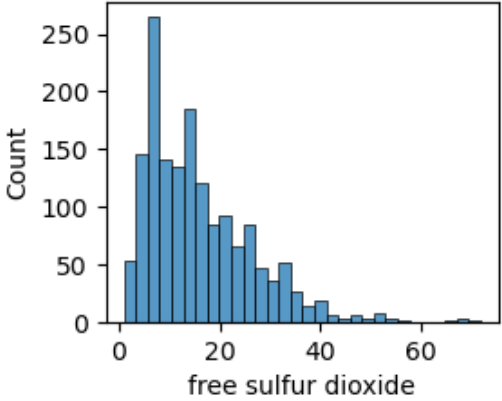
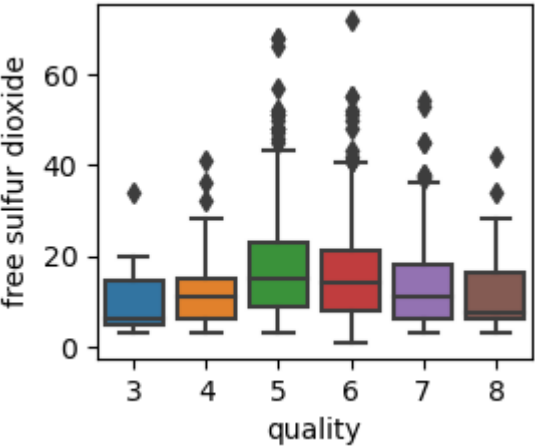
	
<p>Рисунок 6.22. Гистограмма распределения для признака <i>free sulfur dioxide</i> в наборе данных <i>Red Wine Quality</i>.</p>	<p>Рисунок 6.23. Ящики с усами переменной <i>free sulfur dioxide</i> для каждого класса качества в наборе данных <i>Red Wine Quality</i>.</p>

Таблица 6.22. Статистические характеристики регрессора *free sulfur dioxide* в модели парной регрессии *quality ~ free sulfur dioxide*.

	Коэффициент	Стандартная ошибка	t-значение	Р-значение	Доверительный интервал
free sulfur dioxide	-0.050656	0.024991	-2.026944	0.042834	[-0.00164, 0.0490]

По данным гистограммы видно, что подавляющую часть составляют вина с низким содержанием свободного диоксида серы. Такое высокое количество схожих вин привело к выбросам во всех классах. По данным графиков очень тяжело отследить какую-либо

зависимость, на которую указывает отрицательный коэффициент. Доверительный интервал компоненты *free sulfur dioxide sugar* включает 0, а его Р-значение очень высоко. Таким образом, между признаком *free sulfur dioxide* и качеством вина отсутствует зависимость, и он не влияет на конечную оценку для данного набора.

11. Рассмотрим компоненту *pH*. Ее гистограмма распределения (рисунок 6.24) и ящики с усами (рисунок 6.25) приведены в таблице 6.23. Статистические параметры данного регрессора приведены в таблице 6.24.

Таблица 6.21. Результат выполнения функции *plot_feature()* для компоненты *pH*.

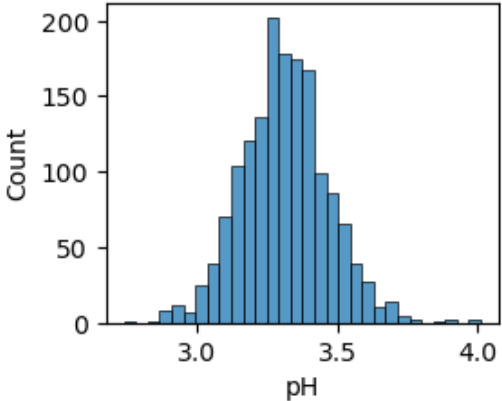
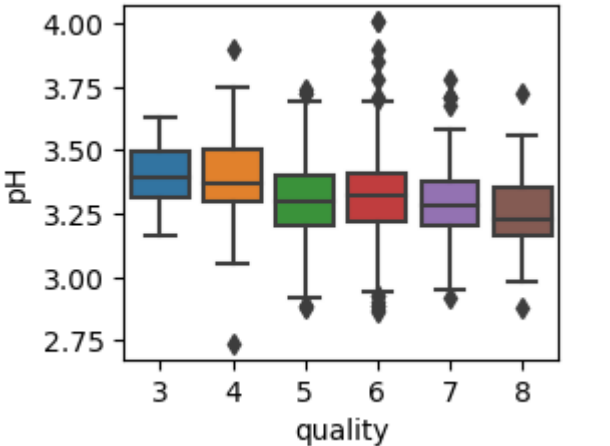
	
<p>Рисунок 6.24. Гистограмма распределения для признака <i>pH</i> в наборе данных <i>Red Wine Quality</i>.</p>	<p>Рисунок 6.25. Ящики с усами переменной <i>pH</i> для каждого класса качества в наборе данных <i>Red Wine Quality</i>.</p>

Таблица 6.24. Статистические характеристики регрессора *pH* в модели парной регрессии *quality ~ pH*.

	Коэффициент	Стандартная ошибка	t-значение	Р-значение	Доверительный интервал
pH	-0.057731	0.024982	-2.310944	0.020963	[-0.04900, 0.00873]

По данным гистограммы видно хорошее распределение вин, все вина набора в основном более кислотные. Почти в каждом классе присутствуют плотные выбросы в малых количествах. По данным графиков очень тяжело отследить какую-либо зависимость, на которую указывает отрицательный коэффициент, ведь значения признака почти одинаковые в каждом классе. Доверительный интервал компоненты pH включает 0, а его P -значение очень высоко. Таким образом, между признаком pH и качеством вина отсутствует зависимость, и он не влияет на конечную оценку для данного набора.

4. Подведение итогов. Определение зависимости качества вина от его химических свойств.

Данные, полученные при статистическом анализе каждой компоненты, не противоречат статистическим оценкам признаков, полученным для классификатора случайного леса. Это указывает на обоснованность выводов о каждой компоненте, сделанных на прошлом этапе решения.

Объединяя статистические характеристики, полученные на прошлом этапе решения, с значимостью признаков в классификаторе случайного леса, можно получить вывод о том, получится ли вино премиум класса или нет в зависимости от его химических характеристик. Таким образом:

К винам премиум классов можно отнести крепкие вина, с высоким содержанием сульфатов и низким содержанием летучих кислот. На принадлежность к премиальным винам также указывает низкая плотность вина и повышенное содержание лимонной кислоты в нем. Последние признаки, определяющие принадлежность экземпляра к винам премиум класса, это общее содержание диоксида серы (должно быть на низком уровне) и фиксированная кислотность (должна быть на высоком уровне).

Такие признаки, как хлориды, остаточный сахар, содержание свободного диоксида серы и уровень pH , в этом наборе данных оказались не значимыми и почти не оказывали влияния на предсказания классификатора, поэтому из итоговой зависимости они были исключены.

Выводы

- Была поставлена задача предсказания оценки качества вина по его химическим характеристикам. Для этого был загружен набор данных *Red Wine Quality*, столбец целевой переменной был сохранен в отдельную переменную и удален из данных, а оставшиеся данные были нормализованы через стандартное отклонение. Таким образом данные были подготовлены для дальнейшего анализа.
- Были исследованы методы классификации, подходящие для данного набора. Кластеризация и мультиклассовый классификатор логистической регрессии показали плохие результаты, из-за чего самым лучшим вариантом классификации

был выбран бинарный классификатор случайного леса (премиум класс вина или нет).

- С помощью алгоритма GridSearch был создан классификатор с использованием лучших гиперпараметров. Классификатор показал относительно высокую точность предсказаний и хорошую способность к распознаванию премиум вин. Для каждой компоненты был оценен ее вклад в построение предсказаний.
- Для каждого признака были получены его индивидуальные статистические характеристики и взаимосвязь между ним и целевой переменной. По данным распределений, выбросов и статистики были сделаны соответствующие выводы для каждого признака.
- Признаки, не имеющие значимости, включают хлориды, остаточный сахар, содержание свободного диоксида серы и уровень pH. По оставшимся признакам был сделан вывод о зависимости класса (премиум или нет) вина от его химических характеристик. К винам премиум классов можно отнести крепкие вина, с высоким содержанием сульфатов и низким содержанием летучих кислот. На принадлежность к премиальным винам также указывает низкая плотность вина и повышенное содержание лимонной кислоты в нем. Последние признаки, определяющие принадлежность экземпляра к винам премиум класса, это общее содержание диоксида серы (должно быть на низком уровне) и фиксированная кислотность (должна быть на высоком уровне).

Код решения задачи и сведения о проверенных моделях приведены в Приложении 6.

Заключение

- В задаче №1 были освоены методы построения парной регрессии и ее оценки через коэффициент детерминации и R-статистику. В наборе данных *Swiss* были определены сильные 2 зависимости: уровня образования от уровня рождаемости (отрицательная), уровня образования от экзаменационных результатов (положительная). Обе модели показали наличие связи между образованием и другими переменными, однако модель *Education~Examination* объясняет отклонения от среднего значения менее точно.
- В задаче №2 были освоены методы построения множественной регрессии и ее оценки. В наборе данных *Swiss* была определена лучшая модель множественной регрессии, все признаки в ней были проверены на корреляцию. В целях улучшения качества модели были проведены эксперименты по добавлению функций от регрессоров в лучшую модель с проверкой корреляции на каждом шаге. Таким образом была определена модель *Examination~Agriculture+I(Fertility^2)*, которая больше всего подходила для описания данных и указывала на положительную зависимость между оценкой за экзамен и проценту мужчин, занимающихся сельскохозяйственной деятельностью и отрицательную зависимость между оценкой за экзамен и уровнем рождаемости. Для каждого ее регрессора были построены доверительные интервалы и отвергнута нулевая гипотеза. Также на примере предсказания модели по произвольным значениям регрессоров был изучен метод расчета доверительного интервала предсказаний.
- В задаче №3 требовалось описать социально-экономическое положение граждан Российской Федерации при помощи набора данных исследования НИУ ВШЭ. Для этого были отобраны и нормализованы необходимые переменные и, по аналогии с задачей №2, построена лучшая модель множественной регрессии. Корреляция регрессоров проверялась при помощи метода получения коэффициентов вздутия дисперсии VIF. По лучшей модели были получены следующие выводы: большую зарплату получают молодые мужчины с продолжительной рабочей неделей, имеющие высшее образование, проживающие в городе, удовлетворённые своей заработной платой, имеющие подчиненных. При этом, если иностранные фирмы и частники являются совладельцами или владельцами их предприятия, то это положительно сказывается на уровне заработной платы. Если государство является совладельцами или владельцами их предприятия, то это отрицательно сказывается на уровне заработной платы. Также отмечается, что чем больше денег индивид получает в течение месяца (учитывается не только зарплата), тем больше у него зарплата. Похожие выводы были получены и для подмножеств исходных данных (не вступавшие в брак, без высшего образования; городские жители, состоящие в браке).
- В задаче №4 были освоены некоторые методы анализа данных при помощи классификации. Данные набора *Students performance in exams* были нормализованы и разбиты на тестовую и тренировочные выборки. Были составлены и оценены с классификаторы метода опорных векторов и случайного леса. При этом для второго классификатора при помощи алгоритма GridSearch было подобрано оптимальное число деревьев в случайном лесе. По метрикам Precision, Recall и F1 классификатор

случайного леса оказался лучше классификатора метода опорных векторов для данного набора. Сопоставляя значимость каждого признака, полученную из классификатора, со знаком коэффициента этого признака в модели множественной регрессии, было заключено, что оценку выше среднего за экзамен по письму получают ученики, набравшие высокие баллы за экзамены по чтению и математике, родители которых имеют лучшее образование. Раса таких учеников должна быть ближе к группе А, ими должны быть пройдены тестовые курсы. Такие ученики зачастую являются девушками, получающими льготные ланчи. Все признаки в этом выводе перечислены в порядке убывания их значимости для классификатора.

- В задаче №5 был проведен первичный анализ данных набора *Red Wine Quality*. Все признаки были загружены и описаны. Было определено количество признаков, объектов и пропущенных значений. Наличие выбросов было определено в первом приближении. При помощи алгоритма PCA удалось снизить размерность признаков с 11 до 7, а для первой компоненты был определен признак, вносящий наибольший вклад в ее построение. Также был освоен алгоритм TSNE, который дал наглядное представление кластеров этого набора данных, которое, к сожалению, оказалось некачественным.
- В задаче №6 был проведен свободный анализ данных набора *Red Wine Quality*. Была поставлена задача классификации с целью предсказания качества вина по его химическим признакам. Для этого были опробованы несколько методов, такие как кластеризация, мультиклассовая и бинарная классификации. Лучшие результаты показал бинарный классификатор случайного леса с оптимальными гиперпараметрами. На его основе для каждой компоненты был получен ее вклад в построение предсказаний. При дальнейшем подробном анализе статистических характеристик каждого признака, из рассмотрения были убраны незначимые признаки, определены все выбросы и указаны векторы влияния (положительное или отрицательное влияние компонента оказывает на качество вина). По всем результатам анализа был сделан следующий вывод. К винам премиум класса можно отнести крепкие вина, с высоким содержанием сульфатов и низким содержанием летучих кислот. На принадлежность к премиальным винам также указывает низкая плотность вина и повышенное содержание лимонной кислоты в нем. Последние признаки, определяющие принадлежность экземпляра к винам премиум класса, это общее содержание диоксида серы (должно быть на низком уровне) и фиксированная кислотность (должна быть на высоком уровне). Все признаки в этом виде описаны в порядке их значимости для классификатора.

Список литературы

1. Introduction to Econometrics with C. Hanck, M. Arnold, A. Gerber, M. Schmelzer. - Essen, Germany: University of Duisburg-Essen, 2021.
2. Айвазян, С.А. Основы эконометрики/С.А. Айвазян, В.С. Мхитарян – Москва: Изд. объединение «ЮНИТИ», 1998. – 1005 с.
3. Вербик, М. Путеводитель по современной эконометрике/М. Вербик – Москва: «Научная книга», 2008. – 616 с.
4. Доугерти, К. Введение в эконометрику/К. Доугерти – Москва: ИНФРА-М, 2009. – 465 с.
5. Магнус, Я.Р. Эконометрика. Начальный курс/Я.Р. Магнус, П.К. Катыхев, А.А. Пересецкий – Москва: Изд-во «ДЕЛО», 2004. – 576 с.

Приложения

Приложение 1

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

data = swiss
help(swiss)

data
plot(data$Fertility, data$Education)
plot(data$Examination, data$Education)

mean(data$Education)
# ~10.98 => образование на низком уровне
var(data$Education)
# ~92.46 => почти на порядок больше среднего => присутствует заметный разброс
sqrt(var(data$Education))
# ~9.61

mean(data$Fertility)
# ~70.14 => в среднем нормальная рождаемость
var(data$Fertility)
# ~156.04 => более чем в 2 раза больше среднего => разброс не велик
sqrt(var(data$Fertility))
# ~12.49

mean(data$Examination)
# ~16.49 => низкий процент получения высшего балла
var(data$Examination)
# ~63.65 => более чем в 3 раза больше среднего => присутствует заметный разброс
sqrt(var(data$Examination))
# ~7.98

model_ed_fer = lm(Education~Fertility, data)
model_ed_fer
summary(model_ed_fer)
# education = -0.51*fertility + 46.82
# имеется явная (хорошие коэффициенты, p-value: 3.659e-07, у каждого
# коэффициента по 3 звездочки) нисходящая зависимость,
# а значит, при большей рождаемости, образование снижается
# При этом R^2~0.43, модель нормально объясняет отклонения от
# среднего значения.

model_ed_ex = lm(Education~Examination, data)
model_ed_ex
summary(model_ed_ex)
# education = 0.84*Examination - 2.90
# имеется восходящая зависимость (p-value: 4.811e-08, коэффициент a
# имеет 3 звездочки), что логично, чем лучше результаты
# экзамена, тем выше качество образования. Однако коэффициент b
# представлен не точно (вообще без звездочек), и R^2~0.48, что
# говорит о том, что не все отклонения могут быть точно предсказаны.
library(ggplot2)
ggplot(data, aes(x = Fertility, y = Education)) +
  geom_point() + geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(x = "Fertility", y = "Education") +
  ggtitle("Scatterplot of Education vs. Fertility with Regression Line")
ggplot(data, aes(x = Examination, y = Education)) + geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(x = "Examination", y = "Education") +
  ggtitle("Scatterplot of Education vs. Examination with Regression Line")
```

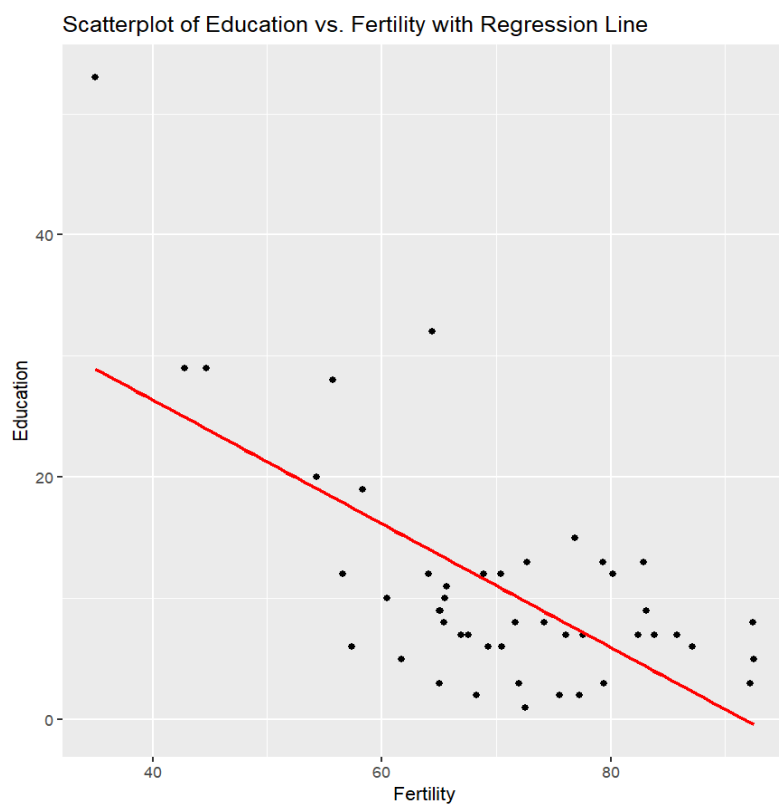


Рисунок 1.1. Результат работы команды `ggplot()` – график зависимостей между *Education* и *Fertility*.

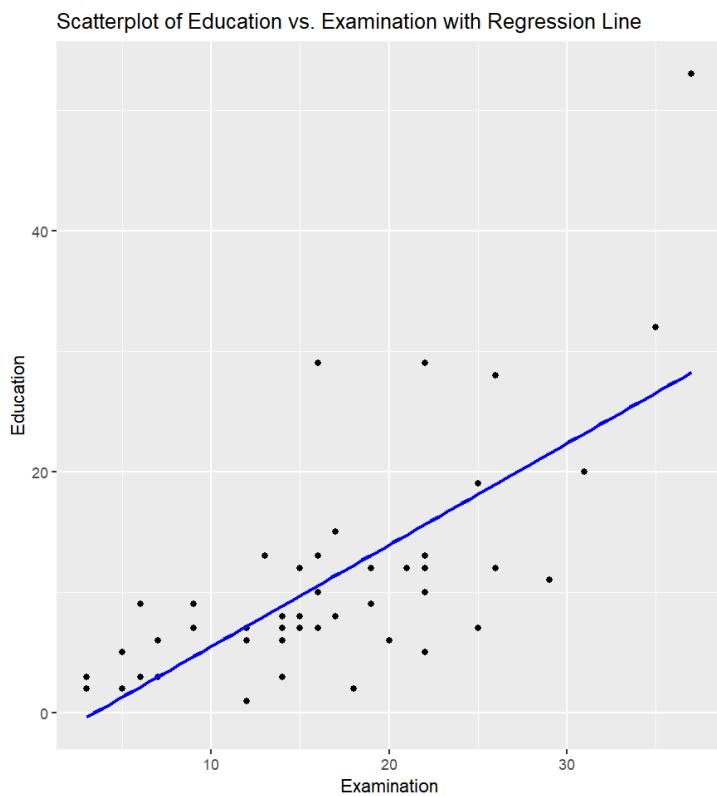


Рисунок 1.2. Результат работы команды `ggplot()` – график зависимостей между *Education* и *Examination*.

Приложение 2

```
# Лазарев Александр. КМБО-03-22. ВАРИАНТ 16

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

data = na.omit(swiss)

data
summary(data)
ggpairs(data)

# Проверим отсутствие линейной зависимости между регрессорами
model1 = lm(Agriculture~Catholic+Fertility, data)
model2 = lm(Catholic~Agriculture+Fertility, data)
model3 = lm(Fertility~Agriculture+Catholic, data)

summary(model1)
# R^2~0.16
summary(model2)
# R^2~0.25
summary(model3)
# R^2~0.21

# R^2 во всех случаях относительно низкий, звездочек всегда мало.
# Из этого можно сделать предположение о том, что регрессоры
# линейно независимы. Чтобы убедиться, посмотрим на R^2 при
# добавлении регрессоров в нужную зависимость:

model01 = lm(Examination~Catholic, data)
summary(model01)
# R^2~0.31, имеется слабая зависимость
model02 = lm(Examination~Catholic+Agriculture, data)
summary(model02)
# R^2~0.56, модель стала значительно лучше
model0 = lm(Examination~Catholic+Agriculture+Fertility, data)
summary(model0)
# R^2~0.66, еще один заметный прирост, модель стала заметно лучше и
# может считаться хорошей, однако Catholic имеет всего 1 звездочку.
# Это говорит о том, что религиозная принадлежность не так сильно
# влияет на результаты экзамена, как Agriculture и Fertility (по 3 звезды)

# Попробуем исключить Catholic:
model00 = lm(Examination~Agriculture+Fertility, data)
summary(model00)
# R^2 снизился всего на 2% (~0.64), значит Catholic все-таки можно
# исключить. В итоге получаем искомую хорошую зависимость.

# Попробуем ввести в нашу зависимость логарифмы от регрессоров
# Для этого сначала исследуем их на линейную зависимость от исходных
# данных.

model1 = lm(log(Agriculture)~Agriculture, data)
summary(model1)
# R^2~0.77
model2 = lm(log(Fertility)~Fertility, data)
summary(model2)
# R^2~0.98
```



```

# Зависимости сильные, значит логарифмами нужно заменять исходные
регрессоры.
# Иначе будет линейная зависимость.

model01 = lm(Examination~I(log(Agriculture))+Fertility, data)
summary(model01)
# R^2~0.61
model02 = lm(Examination~Agriculture+I(log(Fertility)), data)
summary(model02)
# R^2~0.62
# Введение не дало желанных результатов, R^2 уменьшился во всех случаях.
К тому
# же выросли стандартные ошибки. Examination куда лучше зависит от исходных
# регрессоров нежели от их логарифмов, значит вводить их не требуется.

# Прделаем то же самое с попарными произведениями регрессоров
model11 = lm((Agriculture^2)~Agriculture, data)
summary(model11)
# R^2~0.94 => следует лишь попытаться использовать квадрат ВМЕСТО
исходного.
model12 = lm((Fertility^2)~Fertility, data)
summary(model12)
# R^2~0.98 => следует лишь попытаться использовать квадрат ВМЕСТО
исходного.
model13 = lm((Agriculture*Fertility)~Fertility+Agriculture, data)
summary(model13)
# R^2~0.97 => следует лишь попытаться использовать произведение ВМЕСТО
исходных.

# Рассмотрим все варианты:
model03 = lm(Examination~Fertility+I(Agriculture^2), data)
summary(model03)
# R^2~0.61
model04 = lm(Examination~Agriculture+I(Fertility^2), data)
summary(model04)
# R^2~0.65
model05 = lm(Examination~I(Agriculture*Fertility), data)
summary(model05)
# R^2~0.56

# Из всех полученных моделей не уменьшился R^2, не возрос p-value и не
# возросла стандартная ошибка лишь у модели 04. В ней наоборот, слегка
# вырос R^2, p-value уменьшился на порядок (у Fertility), а стандартная
# ошибка уменьшилась на 2 порядка. Такая зависимость сильнее той, в которой
# участвует простой столбец Fertility. Значит лучшей моделью является
# model04 = lm(Examination~Agriculture+I(Fertility^2), data)

# Найдем доверительные интервалы для всех коэффициентов регрессоров.
# Количество измерений в обучающей выборке 44, рассчитано 3 коэффициента.
# Число степеней свободы в модели 44 - 3 = 41. Для такого числа степеней
# свободы и p = 95% рассчитаем значение t-критерия Стьюдента:

t_critical = qt(0.975, df = 41)
t_critical
# t_critical~2.02

# Коэффициент перед Agriculture k1 = -0.19, ско = 0.03.
# Тогда доверительный интервал для коэффициента k1 имеет вид
# k1:[-0.19-2.02*0.03; -0.19+2.02*0.03], k1:[-0,25; -0,13]
# 0 не попадает в интервал, значит сразу отвергаем гипотезу о том, что
этот
# коэффициент может быть равен 0, на уровне значимости 5%.

```

```

# Коэффициент перед I(Fertility^2) k2 = -0.0022, ско = 0.0004.
# Тогда доверительный интервал для коэффициента k2 имеет вид
# k2:[-0.0022-2.02*0.0004; -0.0022+2.02*0.0004], k2:[-0,003; -0,0014]
# 0 не попадает в интервал, значит сразу отвергаем гипотезу о том, что
этот
# коэффициент может быть равен 0, на уровне значимости 5%.

# Найдем доверительный интервал для прогноза с регрессорами
Agriculture=60,
# I(Fertility^2)=4225, p=95%
new.data = data.frame(Agriculture = 60, Fertility=65)
predict(model04, new.data, interval = "confidence")
# Прогноз: Examination~16.58, доверительный интервал [14.81; 18.36]

```

Приложение 3

```
# Лазарев Александр. КМБО-03-22. ВАРИАНТ 16

# Загрузка необходимых библиотек
library("lmtest")
library("rlms")
library("dplyr")
library("GGally")
library("car")

# Загрузка данных из CSV-файла
data <- read.csv("r12i_os26b.csv")

# Использование функции glimpse для ознакомления с данными
glimpse(data)

# Выбор необходимых столбцов из данных и присвоение их переменной data2
data2 = select(data, hj13.2, h_age, hh5, h_educ, status, hj6.2, h_marst,
               hj1.1.2, hj23, hj24, hj6, hj32, hj60)

# Описание параметров:
# h_age - возраст
# hh5 - пол
# hj13.2 - зарплата
# h_marst - семейное положение
# h_educ - наличие высшего образования
# hj6.2 - длительность рабочей недели
# status - тип населенного пункта
# hj1.1.2 - удовл-ть
# hj23 - владеет ли государство компанией?
# hj24 - есть ли иностранные совладельцы?
# hj60 - все денежные поступления

# Замена неотмеченных данных на NA в соответствующих столбцах
data2$hj13.2[which(data2$hj13.2>99999990)] <- NA
data2$h_age[which(data2$h_age>99999990)] <- NA
data2$hh5[which(data2$hh5>99999990)] <- NA
data2$h_educ[which(data2$h_educ>99999990)] <- NA
data2$status[which(data2$status>99999990)] <- NA
data2$hj6.2[which(data2$hj6.2>99999990)] <- NA
data2$h_marst[which(data2$h_marst>99999990)] <- NA
data2$hj1.1.2[which(data2$hj1.1.2>99999990)] <- NA
data2$hj23[which(data2$hj23>99999990)] <- NA
data2$hj24[which(data2$hj24>99999990)] <- NA
data2$hj6[which(data2$hj6>99999990)] <- NA
data2$hj60[which(data2$hj60>99999990)] <- NA

# Удаление строк, содержащих NA
data2 = na.omit(data2)

# Создаем новый столбец 'wed' и копируем в него значения столбца 'h_marst'
data2["wed"] = data2$h_marst

# Создаем еще три новых столбца 'wed1', 'wed2' и 'wed3' и копируем в них
значения 'h_marst'
data2["wed1"] = data2$h_marst
data2["wed2"] = data2$h_marst
data2["wed3"] = data2$h_marst

# Заполняем значения 'wed1', 'wed2' и 'wed3' нулями
```

```

data2$wed1 = 0
data2$wed2 = 0
data2$wed3 = 0

# Заполняем значения 'wed1' единицами, если 'h_marst' равно '2' или '6'
data2$wed1[which(data2$wed=='2')] <- 1
data2$wed1[which(data2$wed=='6')] <- 1

# Заполняем значения 'wed2' единицами, если 'h_marst' равно '4' или '5'
data2$wed2[which(data2$wed=='4')] <- 1
data2$wed2[which(data2$wed=='5')] <- 1

# Заполняем значения 'wed3' единицами, если 'h_marst' равно '1'
data2$wed3[which(data2$wed=='1')] <- 1

# Преобразуем значения 'wed1', 'wed2' и 'wed3' в числовой формат
data2$wed1 = as.numeric(data2$wed1)
data2$wed2 = as.numeric(data2$wed2)
data2$wed3 = as.numeric(data2$wed3)

# Создаем линейную регрессию с зависимой переменной 'hj13.2' и независимыми
переменными 'wed1', 'wed2', 'wed3'
model0 = lm(hj13.2~wed1+wed2+wed3, data2)

# Вычисляем VIF для модели
vif(model0)

# Все значения низкие (не превосходят 5), что говорит об отсутствии
мультиколлинеарности.

# Создаем новый столбец 'sex' и копируем в него значения столбца 'hh5'
data2["sex"] = data2$hh5

# Заменяем значения 'sex' на 0, если 'hh5' равно '2' (женский пол) и на 1,
если 'hh5' равно '1' (мужской пол)
data2$sex[which(data2$sex=='2')] <- 0
data2$sex[which(data2$sex=='1')] <- 1

# Преобразуем значения 'sex' в числовой формат
data2$sex = as.numeric(data2$sex)

# Создание дамми-переменной "status2" на основе переменной "status"
data2["status2"] = data2$status
data2["status2"] = 0 # Заполнение столбца "status2" нулями
# Если значение "status" равно 1 или 2, то значение "status2" равно 1
data2$status2[which(data2$status=='1' | data2$status=='2')] <- 1
data2$status2 = as.numeric(data2$status2) # Преобразование столбца "status2"
в числовой формат

# Создание дамми-переменной "higher_educ" на основе переменной "h_educ"
data2["higher_educ"] = data2$h_educ
data2["higher_educ"] = 0 # Заполнение столбца "higher_educ" нулями
# Если значение "h_educ" равно 21, 22 или 23 то значение "higher_educ" равно 1
data2$higher_educ[which(data2$h_educ=='21' | data2$h_educ=='22' |
data2$h_educ=='23')] <- 1
data2$higher_educ = as.numeric(data2$higher_educ) # Преобразование столбца
"higher_educ" в числовой формат

# Создание дамми-переменной "satisfy" на основе переменной "hj1.1.2"
data2["satisfy"] = data2$hj1.1.2
data2["satisfy"] = 0 # Заполнение столбца "satisfy" нулями
# Если значение "hj1.1.2" равно 1 или 2, то значение "satisfy" равно 1

```

```

data2$satisfy[which(data2$hj1.1.2=='1' | data2$hj1.1.2=='2')] <- 1
data2$satisfy = as.numeric(data2$satisfy) # Преобразование столбца "satisfy"
в числовой формат

# Создание дамми-переменной "state_owner" на основе переменной "hj23"
data2["state_owner"] = data2$hj23
data2["state_owner"] = 0 # Заполнение столбца "state_owner" нулями
data2$state_owner[which(data2$hj23=='1')] <- 1 # Если значение "hj23" равно
1, то значение "state_owner" равно 1
data2$state_owner = as.numeric(data2$state_owner) # Преобразование столбца
"state_owner" в числовой формат

# Создание дамми-переменной "foreign_owner" на основе переменной "hj24"
data2["foreign_owner"] = data2$hj24
data2["foreign_owner"] = 0 # Заполнение столбца "foreign_owner" нулями
data2$foreign_owner[which(data2$hj24=='1')] <- 1 # Если значение "hj24" равно
1, то значение "foreign_owner" равно 1
data2$foreign_owner = as.numeric(data2$foreign_owner) # Преобразование столбца
"foreign_owner" в числовой формат

# Создание дамми-переменной "subordinates" на основе переменной "hj6"
data2["subordinates"] = data2$hj6
data2["subordinates"] = 0 # Заполнение столбца "subordinates" нулями
data2$subordinates[which(data2$hj6=='1')] <- 1 # Если значение "hj6" равно 1,
то значение "subordinates" равно 1
data2$subordinates = as.numeric(data2$subordinates) # Преобразование столбца
"subordinates" в числовой формат

# Создание дамми-переменной "second_job" на основе переменной "hj32"
data2["second_job"] = data2$hj32
data2["second_job"] = 0 # Заполнение столбца "second_job" нулями
data2$second_job[which(data2$hj32=='1')] <- 1 # Если значение "hj32" равно 1,
то значение "second_job" равно 1
data2$second_job = as.numeric(data2$second_job) # Преобразование столбца
"second_job" в числовой формат

# Преобразование переменной "hj13.2" в стандартизированную форму и создание
новой переменной "salary"
sal = as.numeric(data2$hj13.2)
data2["salary"] = (sal - mean(sal)) / sqrt(var(sal))

# Преобразование переменной "h_age" в стандартизированную форму и создание
новой переменной "age"
age = data2$h_age
data2["age"] = (age - mean(age)) / sqrt(var(age))

# Преобразование переменной "hj6.2" в стандартизированную форму и создание
новой переменной "dur"
dur = data2$hj6.2
data2["dur"] = (dur - mean(dur)) / sqrt(var(dur))

# Преобразование переменной "hj60" в стандартизированную форму и создание
новой переменной "payments"
payments = data2$hj60
data2["payments"] = (payments - mean(payments)) / sqrt(var(payments))

# Использование функции glimpse для ознакомления с данными
glimpse(data2)

```

```

# Объявляем модель и запускаем регрессионный анализ
model1 = lm(salary~dur+wed1+wed2+wed3+age+sex+status2+higher_educ+
            satisfy+state_owner+foreign_owner+subordinates+payments, data2)
summary(model1)
# R^2~0.6501
# Сразу дадем wed1, wed2, wed3 из модели, так как у них очень высокие значения
# p-value
model2 = lm(salary~dur+age+sex+status2+satisfy+state_owner+
            foreign_owner+subordinates+payments+higher_educ, data2)
summary(model2) # R~0.6478
# Оцениваем мультиколлинеарность
vif(model2)
# У всех регрессоров коэффициент вздутия не превосходит 3, что говорит об
# отсутствии мультиколлинеарности

# Находим минимальные значения для dur, age и payments
min(data2$dur) # ~ -3.22 => для возведения в рациональную степень и взятия
# логарифма к значению надо прибавить 4
min(data2$age) # ~ -2.14 => для возведения в рациональную степень и взятия
# логарифма к значению надо прибавить 3
min(data2$payments) # ~ -0.98 => для возведения в рациональную степень и взятия
# логарифма к значению надо прибавить 1

# Начинаем вводить функции от регрессоров. Первым делом посмотрим на возведение
# в степень.
# Из-за большого числа возможных моделей (1000) запишем решение циклом и
# отберем лучшую модель по максимальному значению R^2
arr = list() # Реализуем словарь, в котором ключом выступает набор степеней
# регрессоров, а значением R^2
for (i in seq(0.1, 2, 0.1)) {
  for (j in seq(0.1, 2, 0.1)) {
    for (k in seq(0.1, 2, 0.1)) {
      formula = paste("salary ~ I((dur+4)^", i, ") + I((age+3)^", j, ") +
sex+status2+higher_educ+satisfy+state_owner+foreign_owner+subordinates+I((pa
yments+1)^", k, ")", sep = "")
      model = lm(formula, data2)
      arr[[paste0(as.character(i), ", ", as.character(j), ", ", as.character(k))]] <- summary(model)$adj.r.squared
    }
  }
}
max(unlist(arr)) # R^2 ~ 0.6541
names(arr)[which.max(unlist(arr))] # Интересующая модель имеет степени 0.3,
2, 1.1

model = lm(salary~I((dur+4)^0.3)+I((age+3)^2)+sex+status2+higher_educ+
satisfy+state_owner+foreign_owner+subordinates+I((payments+1)^1.1), data2)
summary(model)
vif(model)
# p-статистики отличные (по 3 звездочки у каждого регрессора), как и
# коэффициенты вздутия (максимальный = 1.233064)
# Мы получили лучшую модель. Далее попробуем другие функции.

# Так как при взятии логарифма значения вновь станут не нормализованными,
# их надо нормализовать.
data2["log_dur"] = (I(log(dur+4))-mean(I(log(dur+4)))) /
sqrt(var(I(log(dur+4))))
data2["log_age"] = (I(log(age+3))-mean(I(log(age+3)))) /
sqrt(var(I(log(age+3))))
data2["log_payments"] = (I(log(payments+1))-mean(I(log(payments+1)))) /
sqrt(var(I(log(payments+1))))

```

```

modell1 = lm(salary~log_dur+log_age+sex+status2+higher_educ+
            satisfy+state_owner+foreign_owner+subordinates+log_payments,
data2)
summary(modell1) # R^2 ~ 0.3, что очень мало. Логарифмы включать не требуется

# Произведения
modell2 = lm(salary~I(dur*age)+sex+status2+higher_educ+
            satisfy+state_owner+foreign_owner+subordinates+payments,
data2)
summary(modell2) # R^2~0.64. Воспользуемся этим далее.

modell3 = lm(salary~I(dur*payments)+sex+status2+higher_educ+
            satisfy+state_owner+foreign_owner+subordinates+age, data2)
summary(modell3) # R^2~0.26

modell4 = lm(salary~I(age*payments)+sex+status2+higher_educ+
            satisfy+state_owner+foreign_owner+subordinates+dur, data2)
summary(modell4) # R^2~0.27

modell5 = lm(salary~I(dur*age*payments)+sex+status2+higher_educ+
            satisfy+state_owner+foreign_owner+subordinates, data2)
summary(modell5) # R^2~0.24
# Значения R^2 не превосходят найденные ранее, значит произведения включать не
следует.

# R^2 может повыситься при комбинации функций. Рассмотрим такую функцию, которая
может привести к этому.
modell6 = lm(salary~I((dur+4)^0.3*(age+3)^2)+sex+status2+higher_educ+
            satisfy+state_owner+foreign_owner+subordinates+I((payments+1)^1.1), data2)
summary(modell6) # R^2 ~ 0.6489, значит комбинацию включать не следует

# Таким образом мы получили лучшую модель
salary~I((dur+4)^0.3)+I((age+3)^2)+sex+status2+higher_educ+
#
satisfy+state_owner+foreign_owner+subordinates+I((payments+1)^1.1
# Проведем ее анализ:

# Коэффициент регрессора:
# I((dur + 4)^0.3) - положительный
# I((age + 3)^2) - отрицательный
# sex - положительный
# status2 - положительный
# higher_educ - положительный
# satisfy - положительный
# state_owner - отрицательный
# foreign_owner - положительный
# subordinates - положительный
# I((payments + 1)^1.1) - положительный

# Вывод о том, какие индивиды получают большую зарплату: большую зарплату
получают молодые мужчины с продолжительной рабочей неделей, имеющие высшее
# образование, проживающие в городе, удовлетворенные своей заработной платой,
имеющие подчиненных. При этом, если иностранные фирмы и частники являются
# совладельцами или владельцами Вашего предприятия, то это положительно
сказывается на уровне заработной платы. Если
# государство являются совладельцами или владельцами Вашего предприятия, то
это отрицательно сказывается на уровне заработной платы.
# Также отмечу, что чем больше денег индивид получает в течении месяца
(учитывается не только зарплату), тем больше у него зарплата.

# Проведем анализ выбранных из варианта подмножеств.

```

```

data_unhappy_people = subset(data2, (wed3==1)&(higher_educ==0)) # Никогда не
состоявшие в браке индивиды без высшего образования :(
model_unhappy_people
lm(salary~I((dur+4)^0.3)+I((age+3)^2)+sex+status2+higher_educ+
satisfy+state_owner+foreign_owner+subordinates+I((payments+1)^1.1),
data_unhappy_people)
summary(model_unhappy_people) # R^2~0.72
# Уберем из модели незначимые регрессоры.
model_unhappy_people
lm(salary~I((dur+4)^0.3)+sex+subordinates+I((payments+1)^1.1),
data_unhappy_people)
summary(model_unhappy_people) # R^2~0.72
# Таким образом наибольшую зарплату из никогда не состоявшие в браке индивидов
без высшего образования получают
# мужчины, имеющие подчиненных с высокой продолжительностью рабочей недели и
высоким уровнем денежных поступлений за месяц.

data_happy_people = subset(data2, (status2==1)&(wed1==1)) # Городские жители,
состоящие в браке :)
model_happy_people
lm(salary~I((dur+4)^0.3)+I((age+3)^2)+sex+status2+higher_educ+
satisfy+state_owner+foreign_owner+subordinates+I((payments+1)^1.1),
data_happy_people)
summary(model_happy_people) # R^2 ~ 0.59
# Уберем из модели незначимые регрессоры
model_happy_people = lm(salary~I((dur+4)^0.3)+I((age+3)^2)+sex+higher_educ+
state_owner+subordinates+I((payments+1)^1.1),
data_happy_people)
summary(model_happy_people) # R^2 ~ 0.59
# Таким образом наибольшую зарплату из городских жителей, состоящих в браке
получают молодые мужчины, с долгой рабочей неделей
# и высшим образованием, не работающие на государственную компанию, имеющие
подчиненных и высокий общий денежный доход.

```


Приложение 4

```
import pandas as pd
from IPython.display import display

# Загрузка данных
data = pd.read_csv('StudentsPerformance.csv')
data.head()

# Переименование.
data.rename(columns={'race/ethnicity': 'ethnicity',
                    'parental level of education':
                    'parental_level_education',
                    'test preparation course': 'test_course', 'math
score': 'math_score',
                    'reading score': 'reading_score', 'writing score':
                    'writing_score'}, inplace=True)
# Проверка на дубликаты.
data.duplicated().value_counts()

print('gender:', data['gender'].unique())
print('ethnicity:', data['ethnicity'].unique())
print('parental level education:', data['parental_level_education'].unique())
print('lunch:', data['lunch'].unique())
print('test course:', data['test_course'].unique())

data['gender'] = data['gender'].replace({'female': 0, 'male': 1})
data['ethnicity'] = data['ethnicity'].replace({'group A': 1, 'group B': 2,
'group C': 3, 'group D': 4, 'group E': 5})
data['parental_level_education'] =
data['parental_level_education'].replace({'some high school': 1, 'high
school': 2,
                    'some college': 3, "associate's degree": 4,
"bachelor's degree": 5, "master's degree": 6})
data['lunch'] = data['lunch'].replace({'free/reduced': 0, 'standard': 1})
data['test_course'] = data['test_course'].replace({'none': 0, 'completed': 1})
# Обработка целевого признака:
data['writing_score'] = data['writing_score'].astype(int) # преобразуем
столбец к целому типу
data['writing_score'] = data['writing_score'].apply(lambda x: 1 if x <=
data['writing_score'].mean() else 0)

data['math_score'] = data['math_score'].astype(int) # все оставшиеся исходные
данные следует преобразовать к целому типу
data['reading_score'] = data['reading_score'].astype(int)
data.head() # ознакомимся с результатами

# Выделение целевого признака
target = data['writing_score']

# Удаление целевого признака из данных
data.drop(columns=['writing_score'], inplace=True)

from sklearn.model_selection import train_test_split

# Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(data, target,
test_size=0.25, random_state=16)

from sklearn.svm import SVC

# Создание классификатора
```

```

clf = SVC()

# Обучение классификатора на обучающей выборке
clf.fit(X_train, y_train)

# Предсказание классов на тестовой выборке
y_pred = clf.predict(X_test)

from sklearn.metrics import precision_score, recall_score, f1_score

# Оценка качества модели
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print('Precision:', precision)
print('Recall:', recall)
print('F1-score:', f1)

from sklearn.ensemble import RandomForestClassifier

# Создание классификатора
clf = RandomForestClassifier()

# Определение сетки гиперпараметров
param_grid = {'n_estimators': list(range(50, 1001, 50))} # число деревьев в
лесе.

from sklearn.model_selection import GridSearchCV

grid_search = GridSearchCV(clf, param_grid, scoring=['precision', 'recall',
'f1'], refit='f1')
grid_search.fit(X_train, y_train)
print('Число деревьев для лучших значений метрик в первом приближении: ',
grid_search.best_params_)

param_grid = {'n_estimators': list(range(600, 701, 10))} # число деревьев в
лесе.
grid_search = GridSearchCV(clf, param_grid, scoring=['precision', 'recall',
'f1'], refit='f1')
grid_search.fit(X_train, y_train)
print('Число деревьев для лучших значений метрик во втором приближении: ',
grid_search.best_params_)

# Обучение классификатора с лучшими гиперпараметрами на той же обучающей
выборке для сравнения классификаторов
best_clf = grid_search.best_estimator_
best_clf.fit(X_train, y_train)

# Предсказание классов на той же тестовой выборке
y_pred = best_clf.predict(X_test)

# Оценка качества модели с помощью метрик precision, recall и F1
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

# Вывод результатов
print('Precision:', precision)
print('Recall:', recall)
print('F1-score:', f1)

```

Приложение 5

```
import warnings
warnings.filterwarnings('ignore')
from IPython.display import display
import pandas as pd

# Загрузка данных
data = pd.read_csv('winequality-red.csv')
features = data.columns.tolist() # Список имен признаков
data.head() # Ознакомление с данными (первые 5 строк)

print("Строки \ столбцы")
print(*data.shape, sep=' \ ')

print("Все признаки + их число уникальных значений:", *[i :
data[i].nunique()] for i in data], sep='\n')

display(data.info()) # Проверим форматы признаков

data.isnull().sum() # Проверяем количество пропущенных элементов

data.describe()

import numpy as np

# нормировка признаков через стандартное отклонение
data = (data - np.mean(data, axis=0)) / np.std(data, axis=0)

#отделение целевого признака
target = data['quality']

# результат
data.head()

# определение номера столбца, соответствующего максимальному среднему
значению
print("Столбец с максимальным средним значением после нормировки через
стандартное отклонение: ", np.argmax(np.mean(data, axis=0)))
from sklearn.model_selection import train_test_split

# Удаление целевого признака из данных
data.drop(columns=['quality'], inplace=True)

# Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(data, target,
test_size=0.3, random_state=42)
import seaborn as sns

# загружаем данные заново, чтобы проследить все изначальные корреляции
df = pd.read_csv('winequality-red.csv')

# находим матрицу корреляции
corr_matrix = df.corr()

# отображаем её визуально с помощью тепловой карты
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt='.2f',
annot_kws={"size": 8})

# настройка размера шрифта
sns.set(font_scale=0.8)
```

```

from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Создание экземпляра PCA и подгонка модели
pca = PCA()
pca.fit(X_train, y_train)

# Получение массива долей объясненной дисперсии
variance_ratio = pca.explained_variance_ratio_

# Вычисление кумулятивной суммы долей объясненной дисперсии
cumulative_variance_ratio = np.cumsum(variance_ratio)

print(cumulative_variance_ratio)

x_values = np.arange(1, len(cumulative_variance_ratio) + 1)

plt.plot(x_values, cumulative_variance_ratio, marker='o')
plt.xlabel('Число новых компонент')
plt.ylabel('доля объясненной дисперсии')
plt.title('Зависимость доли объясненной дисперсии от числа новых компонент')

# Задание шага сетки по оси x
plt.xticks(np.arange(min(x_values), max(x_values)+1, 1))

# Нарисовать горизонтальную линию на уровне y = 0.9
plt.axhline(y=0.9, color='red', linestyle='--')

plt.show()

# Находим индекс первого элемента в cumulative_variance_ratio, который
# больше или равен 0.9 (90%)
n_components = np.argmax(cumulative_variance_ratio >= 0.9) + 1

# Вывод результата
print("для объяснения 90% дисперсии необходимо использовать {} из 11 компонент".format(n_components))

# Определяем, какой признак вносит наибольший вклад в первую компоненту
max_feature_idx = np.argmax(np.abs(pca.components_[0]))

print(f"Первая компонента наиболее сильно зависит от признака {features[max_feature_idx]}")

from sklearn.manifold import TSNE
tsne = TSNE(perplexity=10, early_exaggeration=20, learning_rate=200,
init='pca', random_state=42)
X_tsne = tsne.fit_transform(X_test)
plt.scatter(X_tsne[:, 0], X_tsne[:, 1], c=y_test)

```

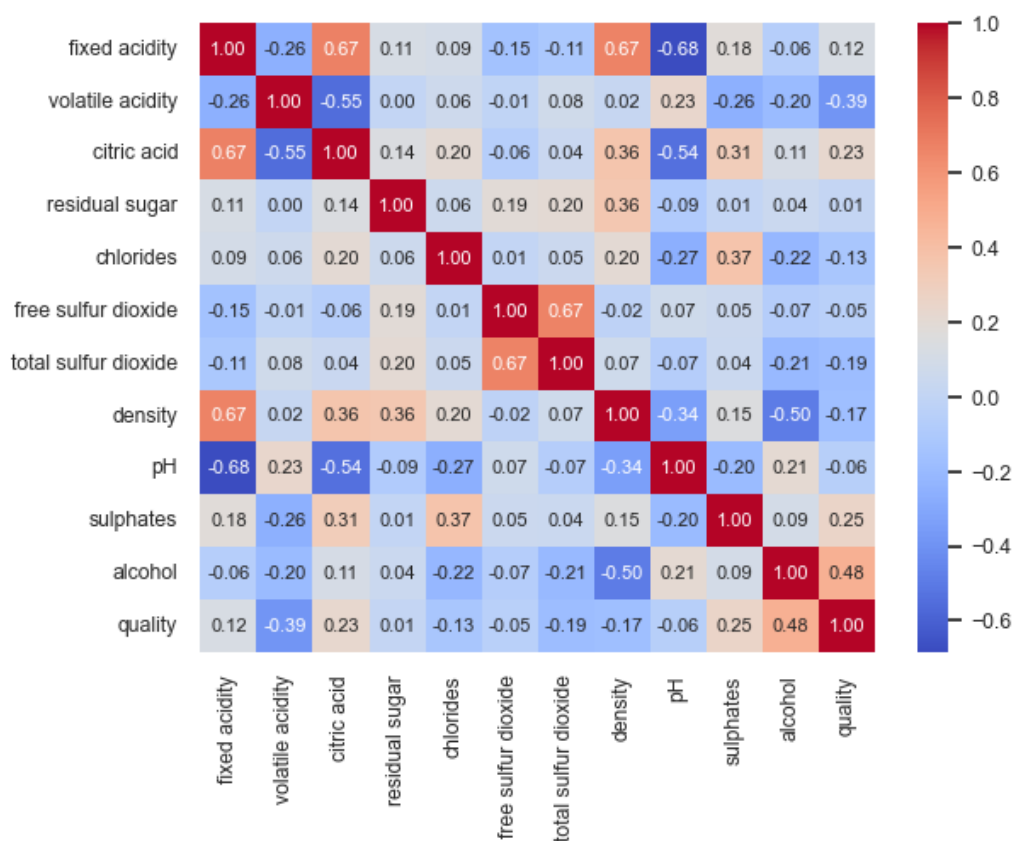


Рисунок 5.2. Матрица корреляций для набора данных *Red Wine Quality*.

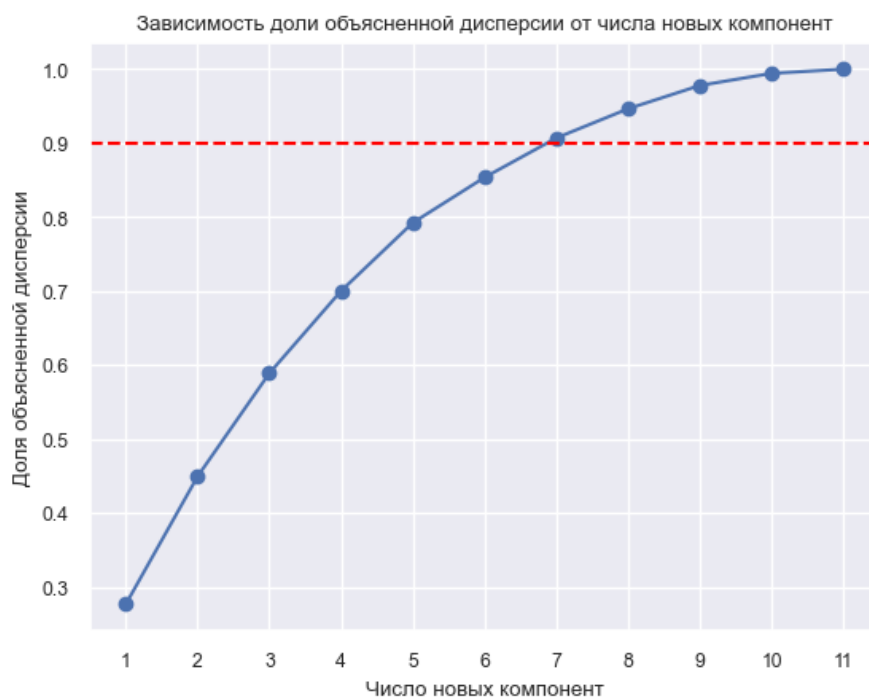


Рисунок 5.3. Наглядное представление зависимости доли объясненной дисперсии от числа новых компонент метода PCA для набора данных *Red Wine Quality*.

Приложение 6

```
# Избавимся от предупреждений из импортируемых библиотек для лучшего
представления работы.
import warnings
warnings.filterwarnings('ignore')
import pandas as pd

# Загрузка данных
data = pd.read_csv('winequality-red.csv')
features = data.columns.tolist() # Список имен признаков
data.head() # Ознакомление с данными (первые 5 строк)
print("Строки \ столбцы")
print(*data.shape, sep=' \ ')
data.info() # Проверим форматы признаков
data.isnull().sum() # Проверяем количество пропущенных элементов
target = data['quality']

# Удаление целевого признака из данных
data.drop(columns=['quality'], inplace=True)

# нормировка признаков через стандартное отклонение
data = (data - np.mean(data, axis=0)) / np.std(data, axis=0)

import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn import metrics

# Создание списка для сохранения значения инерции (сумма квадратов
расстояний до ближайшего центроида) для разных значений k
inertia = []

# Задание разных значений k
k_values = range(1, 12)

# Вычисление инерции для каждого значения k
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(data)
    inertia.append(kmeans.inertia_)

# Построение графика локтя
plt.plot(k_values, inertia, marker='o')
# Задание шага сетки по оси x
plt.xticks(np.arange(1, 12, 1))
plt.xlabel('Число кластеров (k)')
plt.ylabel('Инерция')
plt.title('Метод локтя')
plt.show()

from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Проведение кластеризации для разного числа кластеров
for n_clusters in range(2, 12):
    kmeans = KMeans(n_clusters=n_clusters)
    cluster_labels = kmeans.fit_predict(data)

    # Вычисление среднего силуэтного коэффициента
    silhouette_avg = silhouette_score(data, cluster_labels)
```

```

    print("Если число кластеров равно =", n_clusters, ", то среднее
значение силуэтных коэффициентов =", silhouette_avg)

target = target.apply(lambda x: 1 if x>=7 else 0)
print(max(target))

# Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(data, target,
test_size=0.2, random_state=42)
import numpy as np

from sklearn.model_selection import train_test_split

# Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(data, target,
test_size=0.3, random_state=42)
from sklearn.linear_model import LogisticRegression

# Создание объекта модели логистической регрессии
lr = LogisticRegression(max_iter=1000, random_state=42)

# Обучение классификатора на обучающей выборке
lr.fit(X_train, y_train)

# Предсказание классов на тестовой выборке
y_pred = lr.predict(X_test)
from sklearn.metrics import accuracy_score, f1_score, precision_score,
recall_score, classification_report

# Оценка качества модели
accuracy = accuracy_score(y_test, y_pred)

print('Accuracy:', accuracy)

repoort = classification_report(y_test, y_pred)
print('classification report:\n', f1)
# Получение коэффициентов модели
coefficients = lr.coef_

# Вывод вклада каждой компоненты
for i, feature_name in enumerate(features[:11]):
    print(f"Вклад компоненты {feature_name}: {coefficients[0][i]}")
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV

# Создание объекта модели случайного леса
rf = RandomForestClassifier(random_state=16)

# Определение сетки гиперпараметров
param_grid = {'n_estimators': list(range(50, 1001, 50))} # число
деревьев в лесу.
grid_search = GridSearchCV(rf, param_grid, refit='f1')
grid_search.fit(X_train, y_train)
print('Число деревьев для лучших значений метрик в первом приближении:
', grid_search.best_params_)
best_clf.fit(X_train, y_train)

# Предсказание классов на той же тестовой выборке
y_pred = best_clf.predict(X_test)

precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

```

```

f1 = f1_score(y_test, y_pred)

print(f'F1-score: {f1:.2f}')
print(f'Precision-score: {precision:.2f}')
print(f'Recall-score: {recall:.2f}')

# Получение важности каждого признака
importance = best_clf.feature_importances_
feature_and_importances = [(feature_name, i) for feature_name, i in
zip(features[:11], importance)]
feature_and_importances.sort(reverse=True, key = lambda x : x[1])
# Вывод вклада каждой компоненты
for name, imp in feature_and_importances:
    print(f"Важность компоненты {name} в предсказание: {imp}")

import seaborn as sns
import statsmodels.api as sm
from IPython.display import display

# Наглядное представление изучаемой пары "качество ~ признак"
def plot_feature(X, feature, y):

    # Гистограмма распределения
    plt.figure(figsize=(2.8, 2.3))
    axes = sns.histplot(data=X, x=feature, bins=30, label="Гистограмма
распределения")
    plt.show()

    # Ящик с усами
    plt.figure(figsize=(2.8, 2.3) )
    axes = sns.boxplot(data=X, x=y, y=feature)
    axes.set(xlabel='quality', ylabel=feature, label="Ящик с усами")
    plt.show()

    # Масштабирование даннь
    X1 = X[feature]
    X1 = (X1 - np.mean(X1)) / np.std(X1, axis=0)
    y1 = y
    y1 = (y1 - np.mean(y1, axis=0)) / np.std(y1, axis=0)

    # Создание модели линейной регрессии
    model = sm.OLS(y1, sm.add_constant(X1) )

    # Обучение модели
    results = model.fit()
    conf = results.conf_int()
    conf_str = f"[{conf[1][0]:.5f}, {conf[1][1]:.5f}]"

    # Создание таблицы с результатами
    summary_table = pd.DataFrame({'Коэффициент': results.params,
                                'Стандартная ошибка': results.bse,
                                't-значение': results.tvalues,
                                'P-значение': results.pvalues,
                                'Доверительный интервал': conf_str})
    summary_table.drop(axis = 0, index = 'const', inplace=True)
    # Вывод таблицы с результатами
    display(summary_table)

```



```
# Загрузка данных
data = pd.read_csv('winequality-red.csv')

#отделение целевого признака
target = data['quality']

# Удаление целевого признака из данных
data.drop(columns=['quality'], inplace=True)

plot_feature(data, 'fixed acidity', target)
plot_feature(data, 'volatile acidity', target)
plot_feature(data, 'citric acid', target)
plot_feature(data, 'residual sugar', target)
plot_feature(data, 'chlorides', target)
plot_feature(data, 'free sulfur dioxide', target)
plot_feature(data, 'total sulfur dioxide', target)
plot_feature(data, 'density', target)
plot_feature(data, 'pH', target)
plot_feature(data, 'sulphates', target)
plot_feature(data, 'alcohol', target)
```