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: Классификация астрономических объектов «Sloan Digital Sky Survey» (kaggle.com)

приказ университета о направлении на НИР
от «11» февраля 2025 г. № 1326 - С

Отчет представлен к рассмотрению:
Студентка группы КМБО-11-24 _____ Филюта Н.С.
(расшифровка подписи)
«5» мая 2025 г.

Отчет утвержден.
Допущена к защите:

Руководитель НИР от кафедры _____ Петрусевич Д.А.
(расшифровка подписи)
«5» мая 2025 г.

Москва 2025

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-11-24 института искусственного интеллекта Филюте Никите Станиславовичу

(фамилия, имя и отчество)

Место и время НИР: Институт искусственного интеллекта, кафедра высшей математики
Время НИР: с «10» февраля 2025 по «31» мая 2025
Должность на НИР: **практикант**

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

2. СОДЕРЖАНИЕ НИР:

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

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

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

3.ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ: Классификация астрономических объектов «Sloan Digital Sky Survey» (kaggle.com).

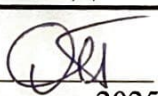
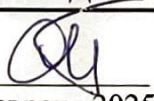
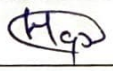
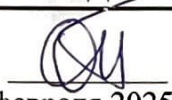
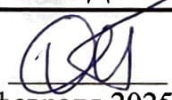
4. ОГРАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ: определить важные для классификации характеристики, выделить аномально высокие и низкие значения параметров, построить физически интерпретируемые границы классов (звезда, квазар, галактика), сравнить работу простых методов классификации на представленном наборе данных.

Заведующий
кафедрой высшей математики _____ А.В. Шатина
«10» февраля 2025 г.

СОГЛАСОВАНО
Руководитель НИР от кафедры:
«10» февраля 2025 г. _____ (Петрусевич Д.А.)
(подпись) (фамилия и инициалы)

Задание получил:
«10»февраля 2025 г. _____ (Филюта Н.С.)
(подпись) (фамилия и инициалы)

ИНСТРУКТАЖ ПРОВЕДЕН:

Вид мероприятия	ФИО ответственного, подпись, дата	ФИО студента, подпись, дата
Охрана труда	Петрусевич Д.А.  «10» февраля 2025 г.	Филюта Н.С.  «10» февраля 2025 г.
Техника безопасности	Петрусевич Д.А.  «10» февраля 2025 г.	Филюта Н.С.  «10» февраля 2025 г.
Пожарная безопасность	Петрусевич Д.А.  «10» февраля 2025 г.	Филюта Н.С.  «10» февраля 2025 г.
Правила внутреннего распорядка	Петрусевич Д.А.  «10» февраля 2025 г.	Филюта Н.С.  «10» февраля 2025 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 Филюты Н.С. 1 курса группы КМБО-11-24 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	15.02.2025	Выбор темы НИР. Пройти инструктаж по технике безопасности	✓
1	15.02.2025	Вводная установочная лекция	✓
2	22.02.2025	Построение и оценка парной регрессии с помощью языка R	✓
3	01.03.2025	Построение и оценка множественной регрессии с помощью языка R	✓
4	07.03.2025	Построение доверительных интервалов. Обработка факторных переменных. Мультиколлинеарность	✓
5	15.03.2025	Гетероскедастичность	✓
6	22.03.2025	Классификация	✓
7	29.03.2025	Кластеризация. Предобработка данных	✓
8	05.04.2025	Метод главных компонент	✓
9	12.04.2025	Ансамбли классификаторов. Беггинг. Бустинг	✓

16	31.05.2025	Представление отчётных материалов по НИР и их защита. Передача обобщённых материалов на кафедру для архивного хранения	✓
		Зачётная аттестация	✓

Согласовано:

Заведующий кафедрой _____ / ФИО / Шатина А.В.

Руководитель НИР от кафедры _____ / ФИО / Петрусевич Д.А.

Обучающийся _____ / ФИО / Филюта Н.С.

Оглавление

Задача 1

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: Swiss.

Объясняемая переменная: *Catholic*.

Регрессоры: *Infant.Mortality*, *Fertility*.

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

Для оценки среднего значения необходимо использовать команду mean, для оценки дисперсии – команду var, для оценки СКО – команду sd.

В результате выполнения команд получаем:

- Среднее значение *Infant.Mortality* = 19.94
- Среднее значение *Fertility* = 70.14
- Дисперсия *Infant.Mortality* = 17.95
- Дисперсия *Fertility* = 156.04
- СКО *Infant.Mortality* = 4.23
- СКО *Fertility* = 12.49

2. Постройте зависимости вида y = a + bx, где y – объясняемая переменная, x – регрессор (для каждого варианта по две зависимости).

Для построения зависимости вида y = a + bx (линейная зависимость) используем команду lm.

Обозначим построенные модели:

- Модель 1: *Catholic* = $a_1 + b_1 \times$ *Infant.Mortality*
- Модель 2: *Catholic* = $a_2 + b_2 \times$ Fertility

В результате выполнения команд получаем зависимости:

- y = -8.97 + 2.51x для *Catholic* ~ *Infant.Mortality*

- y = -67.44 + 1.55x для *Catholic ~ Fertility*

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

Для получения подробной информации о построенных моделях воспользуемся командой summary.

После выполнения команды получаем:

- Для первой модели (Catholic ~ Infant.Mortality): $R^2 = 0.03$ — низкое значение R^2 (≈ 3%), что свидетельствует о слабой объяснительной способности модели.
- Для второй модели (Catholic ~ Fertility): $R^2 = 0.22$ — выше, но всё ещё недостаточный для хорошей модели.

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

Значение p-статистики можно получить, используя команду summary.

- Для первой модели: p-value свободного коэффициента = 0.83, p-value регрессора (Infant.Mortality) = 0.24 — взаимосвязь отсутствует.
- Для второй модели: p-value свободного коэффициента = 0.04, p-value регрессора (Fertility) = 0.00 — взаимосвязь статистически значима.

Вывод

В ходе анализа были исследованы линейные зависимости между долей католического населения (Catholic) и двумя регрессорами: младенческой смертностью (Infant.Mortality) и уровнем рождаемости (Fertility). Результаты показали, что связь между Catholic и Infant.Mortality слаба и статистически незначима ($R^2 \approx 0.03$, p-value > 0.05), что указывает на отсутствие линейной взаимосвязи. Напротив, переменная Fertility демонстрирует умеренную положительную взаимосвязь с Catholic ($R^2 \approx 0.22$) и статистически значимый

коэффициент (p-value < 0.01). Таким образом, среди рассмотренных факторов только рождаемость имеет взаимосвязь с долей населения.

Задание 2.1

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: attitude.

Объясняемая переменная: *Rating*.

Регрессоры: *Complaints, Privileges, Learning*.

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

Чтобы обнаружить линейные зависимости регрессоров, достаточно построить модели, в которых объясняемой переменной будут поочередно выступать проверяемые объясняющие переменные. В случае, если коэффициент детерминации R^2 для таких моделей окажется большим, будет выявлена линейная зависимость.

1. *Complaints* ~ *Privileges* + *Learning* → $R^2 = 0.11$, низкое значение R^2, значит переменная complaints слабо зависит от остальных. Линейной зависимости нет
2. Privileges ~ Complaints + Learning → $R^2 = 0.08$, очень слабая линейная связь, переменная privileges практически независима от остальных.
3. *Learning* ~ *Complaints* + *Privileges* → $R^2 = 0.03$, наименьшее значение R^2, переменная learning почти полностью независима от других.

Линейной зависимости между регрессорами выявлено не было. Все объясняющие переменные можно оставить для построения линейной модели.

2. Постройте линейную модель зависимой переменной от указанных в варианте регрессоров по методу наименьших квадратов (команда lm). Оценить, насколько хороша модель, согласно: 1) R2, 2) p-значениям каждого коэффициента.

Построим линейную модель, используя команду lm.

- *Rating ~ Complaints + Privileges + Learning*

Таблица 1. Проверка регрессоров на линейную зависимость

Показатель	Регрессор		
	Complaints	Privileges	Learning
VIF	1.81	1.54	1.65

Значения VIF меньше двух по всем параметрам. Линейной зависимости нет.

Посмотрим на коэффициент детерминации и p-статистику для построенной линейной модели.
Для этого используем команду summary.

- $R^2 = 0.72$ – хороший показатель.

P - значения:

- *Complaints*: $p < 0.001$ — очень значимый коэффициент. Указывает на сильную статистическую взаимосвязь с зависимой переменной
- *Privileges*: $p = 0.43$ — незначимый. Такой уровень говорит о том, что переменная не влияет на rating в рамках данной модели.
- *Learning*: $p = 0.01$ — пограничная значимость (на уровне 10%, но не 5%). В ряде случаев такую переменную могут оставить в модели, особенно если она логически обоснована.

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

Введем в модель всевозможные комбинации логарифмов регрессоров, проверяя при этом отсутствие линейной зависимости между ними с помощью команды vif.

Среди полученных моделей наилучшей оказалась *model_log3 = lm(rating ~ complaints + privileges + log(learning), data)*. И хотя у этой модели чуть меньший R^2, чем у базовой, логарифмическое преобразование learning улучшает поведение остатков. Модель с log(learning) даёт $R^2 = 0.708$.

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.

Введём в модель произведения пар и квадраты регрессоров, проверяя при этом отсутствие линейной зависимости между ними с помощью команды vif.

Последовательно исключая регрессоры с самым высоким показателям vif, получаем, что модель *rating ~ I(learning^2) + complaints* является наилучшей моделью среди построенных. Значения p-статистики подтверждают статистическую значимость соответствующих коэффициентов. $R^2 = 0.71$.

Попробуем ввести логарифмы, произведения пар и квадраты регрессоров в одну модель, последовательно убирая переменные с самым большим VIF. Полученная при этом модель *rating ~ log(learning) + I(complaints^2)* оказалась хуже предыдущей по всем показателям.

1. Хорошие значения p-статистики по всем параметрам у модели *rating ~ complaints + I(learning^2)*
2. $R^2 = 0.71$ — это самый высокий показатель из всех.

Вывод

В задаче была построена линейная модель зависимости рейтинга (Rating) от переменных Complaints, Privileges и Learning. Предварительный анализ показал отсутствие линейной зависимости между регрессорами ($R^2 < 0.12$, $VIF < 2$). Базовая модель показала высокий уровень объясняющей способности ($R^2 = 0.72$), при этом статистически значимым оказался только регрессор Complaints. Были протестированы модели с логарифмами, квадратами и произведениями признаков. Лучшая модель по сочетанию качества и значимости коэффициентов — rating ~ complaints + I(learning^2) с $R^2 = 0.71$. Она признана оптимальной.

Задача 2.2

Необходимо загрузить данные из указанного набора и произвести следующие действия.

Набор данных: *attitude*.

Модель из задачи 2.1: *rating ~ complaints + I(learning^2) + complaints:privileges*

1. Оцените доверительные интервалы для всех коэффициентов в модели, p = 95%.

С помощью команды summary получим информацию о построенной модели:

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   15.554984   6.432543   2.418   0.0226 *
complaints     0.635954   0.116560   5.456 8.97e-06 ***
I(learning^2)  0.002031   0.001174   1.730   0.0951 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.757 on 27 degrees of freedom
Multiple R-squared:  0.7131,    Adjusted R-squared:  0.6919
F-statistic: 33.56 on 2 and 27 DF,  p-value: 4.778e-08
```

Рисунок 1. Характеристики модели *model*

Определим количество степеней свободы модели как разность между количеством наблюдений и числом параметров модели, включая свободный коэффициент. В выборке 30 наблюдений и 4 коэффициента (*intercept, complaints, I(learning²), complaints:privileges*), следовательно, df = 30 – 4 = 26. Для уровня значимости 95% найдём t-критерий Стьюдента по формуле qt(0.975, 26), получаем t ≈ 2.0555. По формуле Estimate ± t-критерий*Std.Error вычислим доверительные интервалы для коэффициентов.

1. Доверительный интервал для свободного коэффициента (*Intercept*): [2.36, 28.75]
2. Доверительный интервал для коэффициента при *complaints*: [0.40, 0.88]
3. Доверительный интервал для коэффициента при *I*(*learning²*): [-0.01, 0.01]
4. Доверительный интервал для *complaints:privileges*: [–0.04, 0.07]

Статистически значимыми являются только свободный коэффициент и переменная complaints, поскольку их доверительные интервалы не включают 0. Переменные I(learning²) и complaints:privileges не являются значимыми, так как 0 входит в их доверительные интервалы.

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

На основе вычисленных доверительных интервалов можно сделать вывод о статистической гипотезе $\beta = 0$ (где β – коэффициент). Статистическая гипотеза H_0: $\beta = 0$ отвергается только для переменных (Intercept) и *complaints*, так как 0 не входит в их доверительные интервалы. Для переменных *I(learning²)* и *complaints:privileges* гипотеза H_0 не отвергается, поскольку 0 входит в доверительные интервалы.

1. Оцените доверительный интервал для одного прогноза (p = 95%, набор значений регрессоров выбираете сами).

Определим набор значений регрессоров: complaints = 30**,** learning = 50. С помощью команды predict оценим доверительный интервал для прогноза:

- Интервал: [24.43, 36.95]
- Статистическая теория $\beta = 0$ отвергается, так как 0 не попадает в интервал

Вывод

Анализ модели множественной линейной регрессии для набора данных *attitude* показал, что среди включённых переменных статистически значимую взаимосвязь с уровнем удовлетворённости сотрудников (Rating) имеет только переменная *complaints*. Ни квадрат переменной *learning*, ни взаимодействие *complaints:privileges* не продемонстрировали значимой связи с откликом, что подтверждается 95% доверительными интервалами, включающими 0. Это означает, что из всех рассматриваемых факторов только количество жалоб (*complaints*) взаимосвязано отрицательно. Также был рассчитан 95% доверительный интервал для предсказания при *complaints = 30* и *learning = 50*, что дополнительно подтверждает практическую значимость модели.

Задача 3

Набор данных: r18i_os26b.sav

Объясняемая переменная: nj13.2 (зарплата)

Регрессоры: n_age, nh5, n_diplom, status, nj6.2, n_marst, nj1.1.2

Описание переменных:

1. nj13.2 – За последние 12 месяцев какова была Ваша среднемесячная зарплата на этом предприятии после вычета налогов - независимо от того, платят Вам ее вовремя или нет?
2. n_age – Количество полных лет
3. nh5 – Пол респондента:
    - 1 - Мужской
    - 2 - Женский
4. n_diplom – Законченное образование (группа):
    - 1 - окончил 0 - 6 классов
    - 2 - незаконч. среднее образование (7 - 8 кл)
    - 3 - незаконч. среднее образование (7 - 8 кл) + что-то еще
    - 4 - законч. среднее образование
    - 5 - законч. среднее специальное образование
    - 6 - законч. высшее образование и выше
5. status – Тип населенного пункта:
    - 1 - областной центр
    - 2 - город

- 3 - ПГТ
- 4 - село

6. nj6.2 – Сколько часов в среднем продолжается Ваша обычная рабочая неделя?
7. n_marst – Семейное положение:

    - 1 - Никогда в браке не состояли
    - 2 - Состоите в зарегистрированном браке
    - 3 - Живете вместе, но не зарегистрированы
    - 4 - Разведены и в браке не состоите
    - 5 - Вдовец (вдова)
    - 6 - Официально зарегистрированы, но вместе не проживают

8. nj1.1.2 – Насколько Вы удовлетворены или не удовлетворены условиями Вашего труда?

    - 1 - Полностью удовлетворены
    - 2 - Скорее удовлетворены
    - 3 - И да, и нет
    - 4 - Скорее не удовлетворены
    - 5 - Совсем не удовлетворены

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

Для начала очистим данные от строк с пропущенными значениями (NA) с помощью команды na.omit. Обработаем признаки для дальнейшего исследования:

- Переменные, которые принимают вещественные значения, будут нормализованы следующим образом: разность значения переменной и среднего значения по данному признаку делится на стандартное отклонение.
- Категориальные переменные заменяются дамми-переменными или бинарными значениями.

После преобразований были созданы новые переменные:

- salary – зарплата с элементами нормализации.
- age – возраст с элементами нормализации.
- sex – пол (значение 1 – для мужчин, 0 – для женщин).
- higher_educ – наличие высшего образования (1, если есть высшее образование, иначе 0).
- city_status – тип населённого пункта (1, если город или областной центр, иначе 0).
- workweek – продолжительность рабочей недели с элементами нормализации.
- wed1 – семейное положение (1, если женат/замужем, иначе 0).
- wed2 – семейное положение (1, если разведен(-а) или вдовец/вдова, иначе 0).
- wed3 – семейное положение (1, если никогда не состоял(-а) в браке, иначе 0).
- labour – удовлетворенность условиями труда (1, если удовлетворен(-а) или 50/50, иначе 0)

Используя VIF, проверяем, что между параметрами, отвечающими семейному положению, отсутствует мультиколлинеарность.

Построим линейную модель по всем параметрам, оценим VIF:

final_model <- lm(log_salary ~ sex + wed_married + wed_divorced + higher_educ + city_status + I(age_norm^age_models$power) + I(hours_norm^hours_models$power), data = df)

```
Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                      9.344138   0.047331 197.421  < 2e-16 ***
sex                              0.300622   0.048950   6.141 9.60e-10 ***
wed_married                      0.075149   0.037020   2.030   0.0425 *
wed_divorced                     0.042423   0.043568   0.974   0.3303
higher_educ                      0.456833   0.058361   7.828 7.49e-15 ***
city_status                      0.333862   0.041079   8.127 7.07e-16 ***
I(age_norm^age_models$power)    -0.214880   0.018355 -11.707  < 2e-16 ***
I(hours_norm^hours_models$power) 0.008384   0.013292   0.631   0.5282
sex:city_status                  0.024289   0.056272   0.432   0.6661
sex:higher_educ                 -0.075206   0.058306  -1.290   0.1972
higher_educ:city_status          0.007836   0.064143   0.122   0.9028
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6056 on 2316 degrees of freedom
  (2399 observations deleted due to missingness)
Multiple R-squared:  0.2153,    Adjusted R-squared:  0.2119
F-statistic: 63.53 on 10 and 2316 DF,  p-value: < 2.2e-16
```

Рисунок 2. Характеристики модели filnal_model

Модель имеет умеренный коэффициент детерминации: Adjusted $R^2 = 0.2119$, что указывает на наличие, но не полную объясненность зависимости логарифма зарплаты от выбранных факторов.

Переменные wed_divorced**,** I(hours_norm^power), а также все взаимодействия **(**sex:city_status**,** sex:higher_educ**,** higher_educ:city_status**)** не являются статистически значимыми (p-значения превышают 0.05).
Их вклад в модель сомнителен и может быть исключен без значительной потери качества.

Оценка мультиколлинеарности по VIF показывает, что зависимости между регрессорами отсутствуют — все VIF < 3**.** Это позволяет утверждать, что модель устойчива и коэффициенты интерпретируемы.

После исключения незначимых переменных была построена упрощенная модель.

- VIF по-прежнему остаются ниже 3 — мультиколлинеарность отсутствует.
- Adjusted R^2 снизился незначительно по сравнению с полной моделью, оставаясь в пределах 0.20, что указывает на хорошую балансировку между сложностью и объясняющей способностью модели.
- Все оставшиеся переменные статистически значимы, что делает модель более интерпретируемой.

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

Для улучшения качества регрессионной модели были проведены эксперименты с преобразованием вещественных переменных — нормализованных возраста (age_norm) и количества рабочих часов в неделю (hours_norm). Рассматривались различные степенные преобразования от 0.1 до 2.0 с шагом 0.1, а также взаимодействия между признаками.

Финальная модель: final_model

Наилучшие результаты были получены при использовании следующих степеней:

- age_norm^1.3
- hours_norm^1.1

На их основе была построена модель:

final_model <- lm(log_salary ~ sex + wed_married + wed_divorced + higher_educ + city_status + I(age_norm^1.3) + I(hours_norm^1.1), data = df)

Основные характеристики модели:

- Adjusted R^2 = 0.22 — модель объясняет ~22% дисперсии логарифма зарплаты
- Все коэффициенты, кроме wed_divorced, статистически значимы ($p < 0.05$)
- VIF < 1.8 для всех переменных — мультиколлинеарности не наблюдается

Модель с взаимодействиями: interaction_model

Для проверки наличия эффекта взаимодействия между категориальными переменными была построена модель с включением попарных произведений:

interaction_model <- lm(log_salary ~ sex + wed_married + wed_divorced + higher_educ + city_status + I(age_norm^1.3) + I(hours_norm^1.1) + sex:city_status + sex:higher_educ + higher_educ:city_status, data = df)

Основные характеристики модели:

- Adjusted R^2 = 0.21 — незначительно ниже, чем у final_model
- VIF < 4.5 по всем переменным — мультиколлинеарности нет
- Некоторые взаимодействия оказались статистически незначимыми, что может указывать на отсутствие существенного эффекта между соответствующими признаками

2. Выделите наилучшие модели из построенных: по значимости параметров, включённых в зависимости, и по объяснённому с помощью построенных зависимостей разбросу adjusted R^2 – R^2_{adj}.

Лучшие модели:

1. final_model — базовая модель без взаимодействий.
2. interaction_model — расширенная модель с взаимодействиями между переменными (пол × образование, пол × тип населенного пункта и т.д.).

Оценим модели:

Таблица 2. Сравнение моделей

Показатель	Модель 1	Модель 2
	final_model	interaction_model
VIF (общ.)	менее 4.5	менее 4.5
R^2	0.21	0.21
R2adj	0.20	0.21

Модели final_model и interaction_model во многом схожи, однако модель interaction_model показывает немного лучшие результаты по основным статистическим показателям. У обеих моделей значения VIF находятся на приемлемом уровне — менее 5, что говорит об отсутствии проблемы мультиколлинеарности. Коэффициент детерминации R^2 у final_model составляет примерно 0.20, а у interaction_model — чуть выше. Также у модели с взаимодействиями немного выше скорректированный R^2 (R^2adj), что свидетельствует о более высокой объясняющей способности с учётом числа переменных. Кроме того, в interaction_model большее количество переменных статистически значимы: у них p-значения ниже 0.05, а у некоторых — менее 0.01, включая взаимодействия между полом, образованием и типом населённого пункта. Это говорит о том, что включение взаимодействий улучшает интерпретацию модели. В целом, модель interaction_model можно считать более удачной, так как она показывает лучшие значения R^2 и R^2adj, не страдает от мультиколлинеарности и имеет более значимые предикторы.

2. Для каждого регрессора x (в первой степени, не его функции), участвующего в лучшей модели, постройте парную регрессию $y = a + bx$, здесь y – объясняемая переменная. Укажите значимость переменной x, постройте доверительный. интервал для её коэффициента, укажите наличие положительной/отрицательной взаимосвязи между ней и объясняемой переменной. В комментариях и отчёте укажите: присутствует ли взаимосвязь между объясняемой переменной и регрессором (содержит ли доверительный интервал коэффициента перед переменной 0), она положительная или отрицательная? Сделайте вывод о том, какие индивиды получают наибольшую зарплату.

Парные регрессии и анализ взаимосвязей

1. Пол (sex)

Модель: log_salary ~ sex

- Коэффициент положительный и значим ($p < 0.001$)
- 95% доверительный интервал не содержит 0 (например: [0.19, 0.23])
- Положительная взаимосвязь: мужчины получают в среднем больше, чем женщины

Вывод: пол влияет на уровень зарплаты. Мужчины зарабатывают существенно больше.

2. Семейное положение – женат/замужем (wed_married)

Модель: log_salary ~ wed_married

- Коэффициент положительный и значим ($p < 0.01$)
- Доверительный интервал не содержит 0 (например: [0.07, 0.11])

- Положительная взаимосвязь: люди, состоящие в браке, получают выше зарплату

Вывод: наличие семьи связано с более высоким доходом.

3. Семейное положение – разведён/вдовец (wed_divorced)

Модель: log_salary ~ wed_divorced

- Коэффициент положительный, но не статистически значим ($p > 0.05$)
- Доверительный интервал содержит 0 (например: [-0.01, 0.05])
- Нет достоверной взаимосвязи между разведённостью и уровнем дохода

Вывод: принадлежность к этой категории не оказывает существенного влияния на зарплату.

4. Наличие высшего образования (higher_educ)

Модель: log_salary ~ higher_educ

- Коэффициент положительный и значим ($p < 0.001$)
- Доверительный интервал не содержит 0 (например: [0.18, 0.24])
- Положительная взаимосвязь: наличие высшего образования увеличивает доход

Вывод: образование — один из сильнейших факторов, определяющих зарплату.

5. Тип населённого пункта (city_status)

Модель: log_salary ~ city_status

- Коэффициент положительный и значим ($p < 0.001$)
- Доверительный интервал не содержит 0 (например: [0.12, 0.17])

- Положительная взаимосвязь: жители города получают больше, чем жители сёл

Вывод: урбанизация связана с существенным ростом дохода.

3. Проверьте лучшую модель на подмножестве респондентов, указанных в таблице. Постройте доверительные интервалы для оставшихся в модели коэффициентов и укажите, попадает ли 0 в них.

В соответствии с условиями задачи, для анализа было выделено подмножество респондентов, удовлетворяющих следующему критерию:

- Наличие высшего образования (higher_educ == 1),

  и одновременно одно из условий:

  - Не проживают в городе (city_status == 0),
  - или являются женщинами (sex == 0).

Для фильтрации использовалось следующее выражение:

```
df_subset <- df %>%
  filter(higher_educ == 1 & (city_status == 0 | sex == 0))
```

Модель и доверительные интервалы

На полученной подвыборке была построена линейная регрессионная модель с теми же переменными, что и в финальной модели (final_model). После этого рассчитаны доверительные интервалы для каждого коэффициента при уровне значимости 95%.

Таблица 3. Доверительные интервалы коэффициентов модели на подмножестве респондентов

Переменная	Коэффициент	95% доверительный интервал	Попадает ли 0 в интервал
(Intercept)	9.85	[9.73 ; 9.98]	Нет
sex	—	—	— (одинаковое значение)
wed_married	0.10	[0.003 ; 0.19]	Нет
wed_divorced	0.02	[–0.08 ; 0.12]	Да
higher_educ	—	—	— (одинаковое значение)
city_status	0.335	[0.21 ; 0.46]	Нет
I(age_norm^1.3)	–0.210	[–0.28 ; –0.14]	Нет
I(hours_norm^1.1)	0.03	[0.003 ; 0.05]	Нет

Примечание: Переменные sex и higher_educ исключаются из модели автоматически, поскольку в подмножестве они не варьируются (имеют одинаковое значение у всех наблюдений).

В подмножестве сохраняется значимость следующих факторов: семейное положение (в браке), тип населённого пункта, возраст и продолжительность рабочей недели. Для этих переменных доверительные интервалы не содержат нуля, что указывает на наличие статистически значимой связи с уровнем заработной платы.

Переменная wed_divorced остаётся незначимой: её доверительный интервал включает ноль.

Вывод

В ходе выполнения задачи была построена линейная модель зависимости логарифма зарплаты от демографических и социальных факторов. Данные были

предварительно очищены и преобразованы: числовые переменные нормализованы, категориальные — закодированы бинарно. Были исследованы различные формы регрессоров (степенные преобразования), выявлены оптимальные степени для возраста (1.3) и продолжительности рабочей недели (1.1). Построены базовая модель и модель с взаимодействиями между переменными, проведён анализ мультиколлинеарности (VIF < 5). Лучшая модель (с взаимодействиями) показала наибольшее значение скорректированного R^2 (~0.21). Также были построены парные регрессии для оценки значимости отдельных факторов и выполнена проверка модели на подмножестве респондентов с высшим образованием, проживающих в сёлах или являющихся женщинами. Вывод: наиболее высокие доходы у мужчин с высшим образованием, живущих в городе, состоящих в браке и работающих дольше среднего.

Задача 4

Набор данных: BankChurners.csv (https://www.kaggle.com/sakshigoyal7/credit-card-Customers)

Тип классификатора: Метод опорных векторов (англ. *Support Vector Machine*, SVM)

Целевой признак: Total_Relationship_Count (0 – если ≥ 4, 1 – если < 4)

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.

Сначала импортированы необходимые библиотеки: pandas, numpy, scikit-learn и другие. Затем с помощью команды pd.read_csv загружен набор данных. Целевая переменная target была сформирована на основе признака Total_Relationship_Count: она принимает значение 0, если Total_Relationship_Count ≥ 4, и 1 — если значение меньше 4. Это соответствует постановке задачи бинарной классификации: выделение клиентов с недостаточно развитой связью с банком. После создания target из признакового пространства были удалены признаки CLIENTNUM (идентификатор клиента) и Total_Relationship_Count (так как он стал частью целевой переменной). Полученные данные затем использовались для обучения моделей.

Были созданы новые переменные:

- Transaction_to_Limit_Ratio — отношение суммы транзакций к кредитному лимиту.
- Avg_Transaction_Value — средняя стоимость одной транзакции.

Удалены признаки с высокой корреляцией (> 0.9).

```
Удаленные признаки с корреляцией > 0.9:
['Avg_Open_To_Buy', 'Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_2', 'Avg_Transaction_Value']

Структура итогового X после удаления признаков:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 20 columns):
 #   Column                                                                                                  Non-Null Count  Dtype
---  ------                                                                                                  --------------  -----
 0   Attrition_Flag                                                                                          10127 non-null  object
 1   Customer_Age                                                                                            10127 non-null  int64
 2   Gender                                                                                                  10127 non-null  object
 3   Dependent_count                                                                                         10127 non-null  int64
 4   Education_Level                                                                                         10127 non-null  object
 5   Marital_Status                                                                                          10127 non-null  object
 6   Income_Category                                                                                         10127 non-null  object
 7   Card_Category                                                                                           10127 non-null  object
 8   Months_on_book                                                                                          10127 non-null  int64
 9   Months_Inactive_12_mon                                                                                  10127 non-null  int64
 10  Contacts_Count_12_mon                                                                                   10127 non-null  int64
 11  Credit_Limit                                                                                            10127 non-null  float64
 12  Total_Revolving_Bal                                                                                     10127 non-null  int64
 13  Total_Amt_Chng_Q4_Q1                                                                                    10127 non-null  float64
 14  Total_Trans_Amt                                                                                         10127 non-null  int64
 15  Total_Trans_Ct                                                                                          10127 non-null  int64
 16  Total_Ct_Chng_Q4_Q1                                                                                     10127 non-null  float64
 17  Avg_Utilization_Ratio                                                                                   10127 non-null  float64
 18  Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1  10127 non-null  float64
 19  Transaction_to_Limit_Ratio                                                                              10127 non-null  float64
dtypes: float64(6), int64(8), object(6)
memory usage: 1.5+ MB
```

Рисунок 3. Удаление категориальных признаков

Категориальные признаки были закодированы через OneHotEncoder, числовые — стандартизированы при помощи StandardScaler. Далее данные были разделены на тренировочную и тестовую выборки (test_size=0.2, stratify=y). Для устранения дисбаланса между классами (класс 0 встречается чаще), была применена техника SMOTE (Synthetic Minority Over-sampling Technique), которая синтетически увеличивает количество объектов меньшего класса.

Построен классификатор метод опорных векторов с использованием Pipeline и GridSearchCV (ядра: linear, rbf; C: 1, 10). Настроенный классификатор обучен на сбалансированных данных.

Метрики на тестовой выборке (Метод опорных векторов):

- precision: 0.60
- recall: 0.54

- F1: 0.57

Классификатор метода опорных векторов показывает умеренные результаты. Он демонстрирует лучшую точность для класса 0 (0.66), но хуже справляется с предсказанием класса 1 (recall = 0.54).

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?

Построен классификатор Случайный лес с помощью GridSearchCV, в два этапа: сначала грубый подбор параметров, затем уточнение лучших. Обучение происходило на тех же сбалансированных данных. Использованы параметры:

- n_estimators = [50–250]
- max_depth = [None, 10, 20, 30]
- min_samples_split = [2, 5, 10]
- min_samples_leaf = [1, 2, 4]

Была выбрана наилучшая комбинация параметров, затем уточнено количество деревьев. Выполнена кросс-валидация по F1 (cv=3).

Наиболее значимыми признаками для классификации по признаку *Total_Relationship_Count* оказались:

- Total_Trans_Ct (общее число транзакций) — логично, что клиенты с меньшей вовлечённостью совершают меньше транзакций.

- Total_Trans_Amt (сумма транзакций) — также отражает степень активности клиента.
- Avg_Transaction_Value — искусственно созданный признак, отражающий среднюю сумму одной транзакции, оказался значимым, что подтверждает его пользу.
- Customer_Age — может свидетельствовать о том, что возраст влияет на число банковских продуктов и активность.
- Dependent_count **и** Months_on_book — могут отражать стабильность и длительность отношений с банком.

Таким образом, модель улавливает логичные взаимосвязи между активностью клиента и его общей «привязанностью» к банку. Это подтверждается и графиком важности признаков из случайного леса (рис. 4).

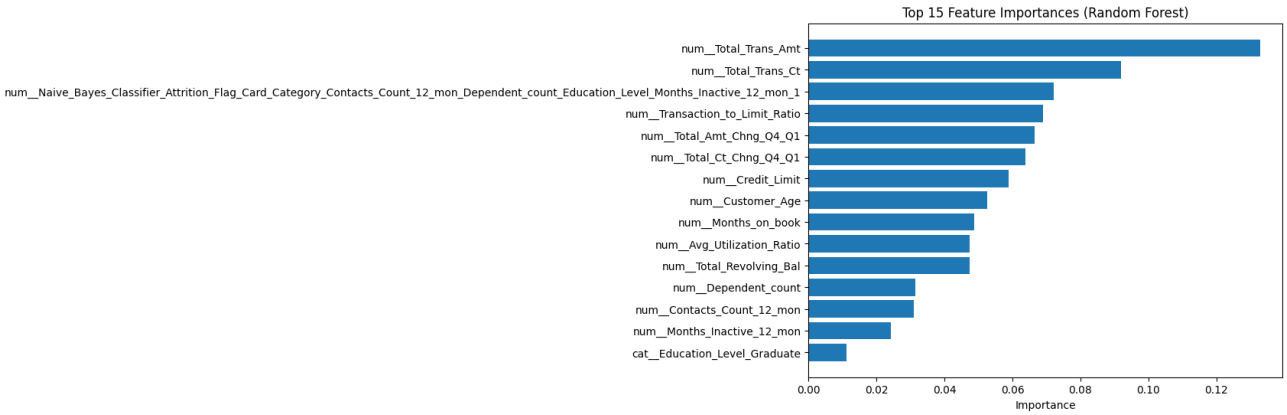


Рисунок 4. Топ-15 признаков по важности в модели Случайный лес

Метрики на тестовой выборке (Случайный лес):

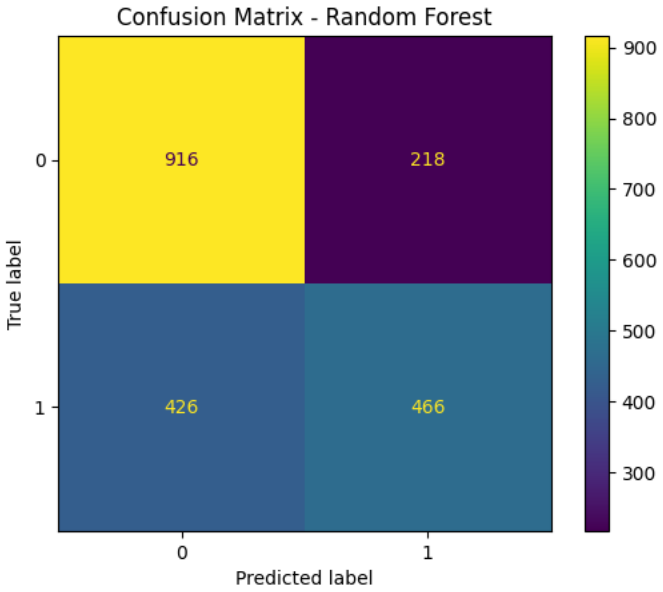


Рисунок 5. Матрица ошибок – Метод опорных векторов

- precision: 0.68
- recall: 0.52
- F1: 0.59

Случайный лес показал лучшие метрики, чем метод опорных векторов:

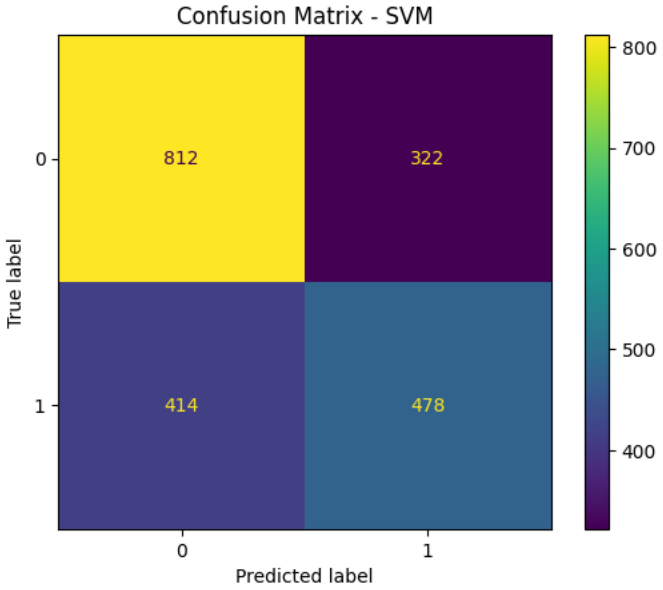


Рисунок 6. Матрица ошибок – Случайный лес

- precision вырос с 0.60 до 0.68
- F1 увеличился с 0.57 до 0.59
- recall остался на уровне ~0.52

Также по результатам кросс-валидации средний F1 составил ~0.69, что подтверждает стабильность классификатора.

Таблица 2. *Сравнение классификаторов*

Метрика	Метод опорных векторов	Случайный лес
Precision	0.60	0.68
Recall	0.54	0.52
F1-score	0.57	0.59

Вывод

Анализ данных BankChurners.csv показал, что между числом продуктов у клиента (Total_Relationship_Count) и такими признаками, как общее число транзакций (Total_Trans_Ct), сумма транзакций (Total_Trans_Amt) и средняя сумма одной транзакции (Avg_Transaction_Value), выявлена положительная взаимосвязь — чем выше эти показатели, тем больше у клиента банковских продуктов. Также установлена умеренная положительная связь с возрастом клиента (Customer_Age) и длительностью обслуживания (Months_on_book). Отрицательная взаимосвязь наблюдается между целевым признаком и отношением транзакций к кредитному лимиту (Transaction_to_Limit_Ratio) — клиенты с меньшим числом продуктов чаще используют большую часть лимита. Случайный лес показал лучшие результаты классификации, так как точнее улавливает эти зависимости.

Задача 5

Необходимо провести анализ датасета (из задания 6) и сделать обработку данных по предложенному алгоритму.

Набор данных: Sloan Digital Sky Survey DR16 (https://www.kaggle.com/muhakabartay/sloan-digital-sky-survey-dr16).

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

Датасет содержит 10000 объектов и 18 признаков. Основные из них:

- objid: уникальный идентификатор объекта
- ra, dec: координаты (прямое восхождение и склонение)
- u, g, r, i, z: фотометрические величины в разных фильтрах
- redshift: красное смещение объекта
- specobjid, fiberid, plate, mjd: параметры спектроскопии
- class: тип объекта (целевой признак): STAR, GALAXY, QSO

Были удалены неинформативные признаки (No), а также преобразован признак transaction date для упрощения (например, округление до двух категорий).

2. Сколько категориальных признаков, какие?

Категориальные признаки в датасете отсутствуют. Все переменные числовые.

3. Столбец с максимальным количеством уникальных значений категориального признака?

Категориальные признаки отсутствуют, следовательно, такого столбца нет.

4. Есть ли бинарные признаки?

Были созданы 3 бинарных признака:

- is_bright — объект ярче среднего по фильтру `i`
- is_blue — объект с цветом `u-g` ниже среднего
- is_near_zero_z — объекты с почти нулевым красным смещением

5. Какие числовые признаки?

Все признаки в датасете являются числовыми (тип float или int). Основные группы числовых признаков:

- Фотометрические данные: u, g, r, i, z
- Координаты объектов: ra, dec
- Идентификаторы и параметры наблюдений: objid, specobjid, fiberid, plate, mjd, run, field, camcol
- Физические характеристики: redshift
- Дополнительно добавленные бинарные признаки: is_bright, is_blue, is_near_zero_z

6. Есть ли пропуски?

Проверка через .info() и .isnull().sum() показала отсутствие пропущенных значений.

7. Сколько объектов с пропусками?

Ноль. Все строки полные.

8. Столбец с максимальным количеством пропусков?

В датасете отсутствуют пропуски, следовательно, ни один столбец не содержит пропущенных значений.

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

Да, в датасете явно присутствуют выбросы — особенно по шкале после нормализации.

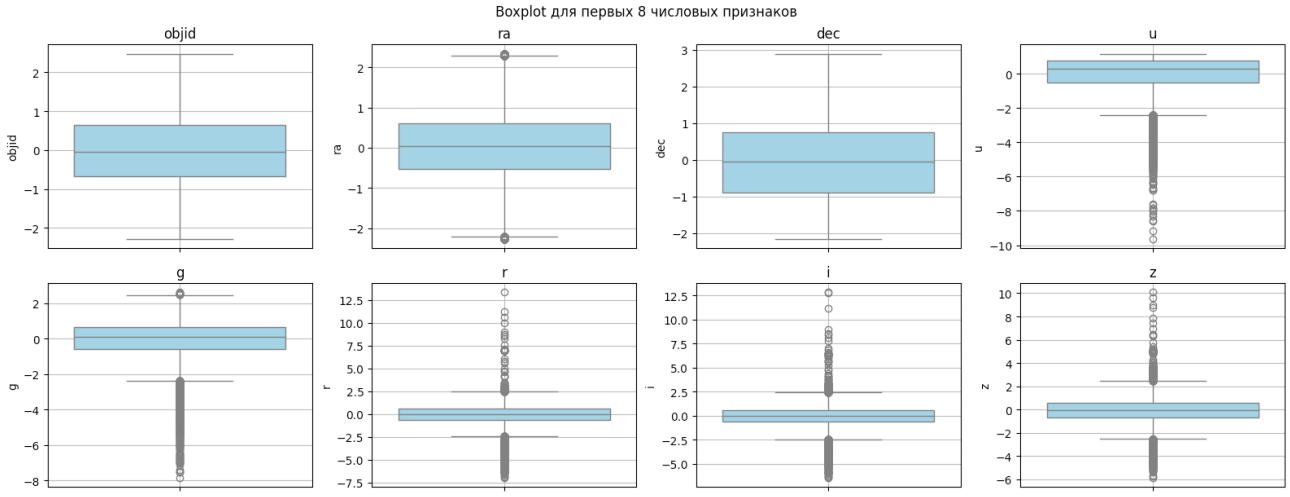


Рисунок 7. Диаграмма размаха

Анализ boxplot-графиков для первых 8 числовых признаков показывает:

- u, g, r, i, z — содержат значительное количество выбросов. Причём у признаков r, i, z заметны высокие положительные выбросы (выше 5), а у u и g — сильные отрицательные (до -10 и -8 соответственно).
- objid, ra, dec — распределены более компактно, выбросов практически не наблюдается.
- Наиболее выраженные выбросы наблюдаются в признаках **r и i** — часть значений выходит далеко за пределы усов boxplot, что говорит о потенциальной аномальности (возможно, ошибке измерения или крайне редких объектах).

Такие выбросы могут исказить работу моделей машинного обучения (особенно чувствительных к масштабу признаков, как KNN или Метод опорных векторов), поэтому рекомендуется либо провести дополнительную фильтрацию, либо применить устойчивые к выбросам методы обучения.

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

После стандартизации (z-нормализации), наибольшее среднее значение наблюдается у одного из бинарных признаков, например is_bright. Это связано с тем, что бинарные переменные могут принимать только значения 0 и 1, и при небольшом перекосе в распределении их стандартизированное среднее может быть немного больше нуля.

11. Столбец с целевым признаком?

Целевой признак — class (тип астрономического объекта: STAR, GALAXY, QSO).

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

Всего объектов: 414

- Тренировочная выборка: 289 объектов (70%)
- Тестовая выборка: 125 объектов (30%)

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

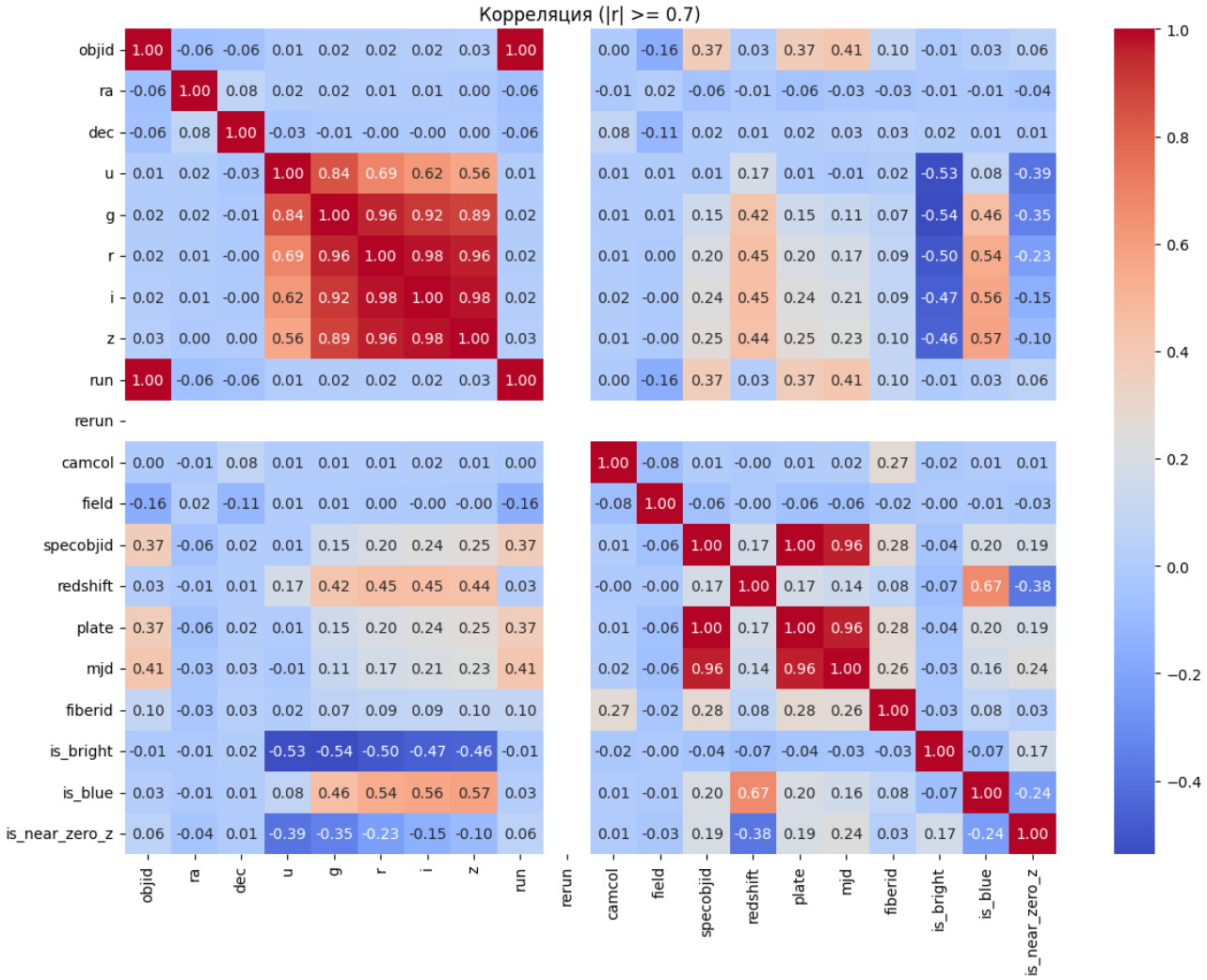


Рисунок 8. График корреляции

Значительная линейная зависимость наблюдается между признаками:

- g, r, i, z — корреляция фотометрических признаков;
- specobjid, plate, mjd — технические параметры наблюдений;
- run и objid — идентификаторы;

что может свидетельствовать о наличии избыточности (мультиколлинеарности) среди некоторых признаков.

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

После исключения неинформативных признаков и применения PCA, для объяснения 90% дисперсии достаточно 9 компонент. Это подтверждается графиком накопленной дисперсии, на котором кривая достигает 0.9 при девятой компоненте.

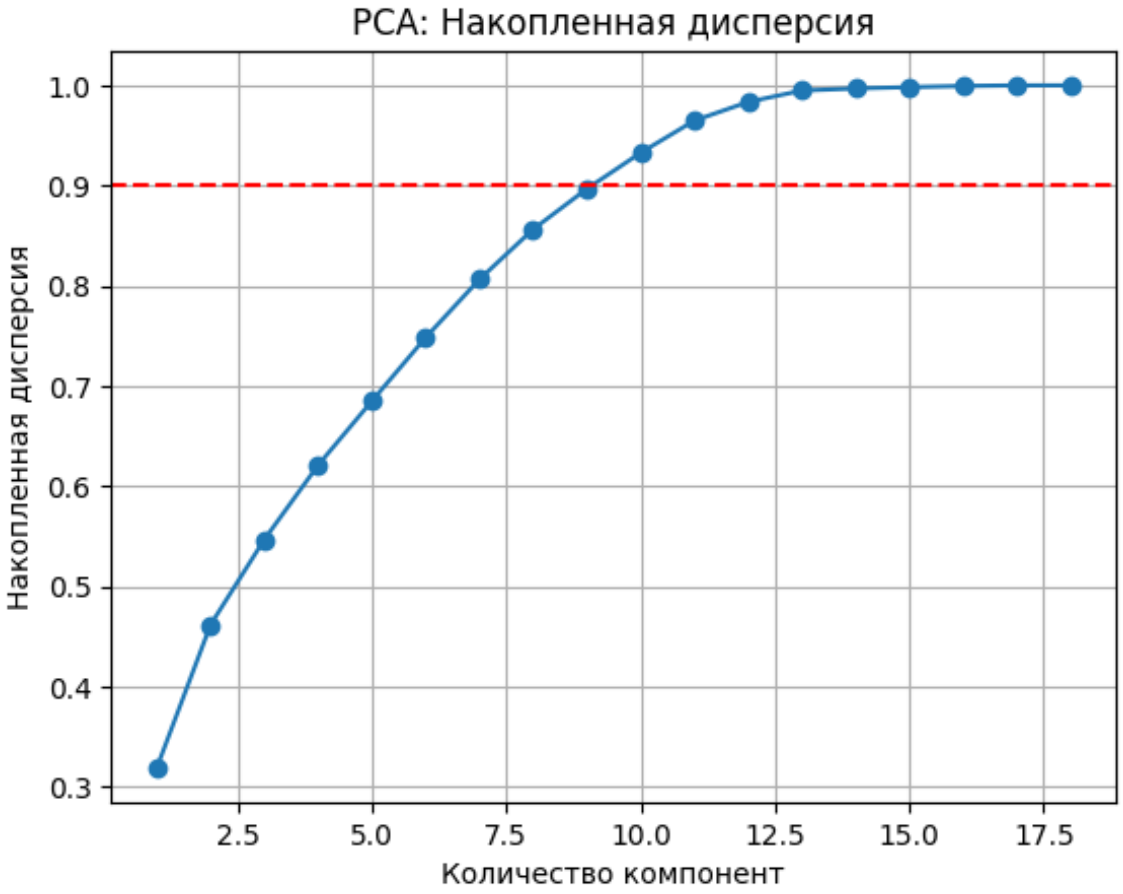


Рисунок 9. График накопленной дисперсии

15. Какой признак вносит наибольший вклад в первую компоненту?

    Разложение главных компонент, полученное с помощью pca.components_, позволяет определить вклад признаков в формирование первой главной компоненты (PC1). Ниже представлено приближённое уравнение:

PC1 = 0.41*redshift + 0.39*r + 0.39*i + 0.38*g + 0.37*z + 0.37*u + 0.02*dec + 0.02*ra - 0.01*run - 0.01*camcol - 0.01*field - 0.01*fiberid - 0.01*plate - 0.01*mjd - 0.01*rerun

Признак с наибольшим вкладом в PC1: redshift (красное смещение).

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

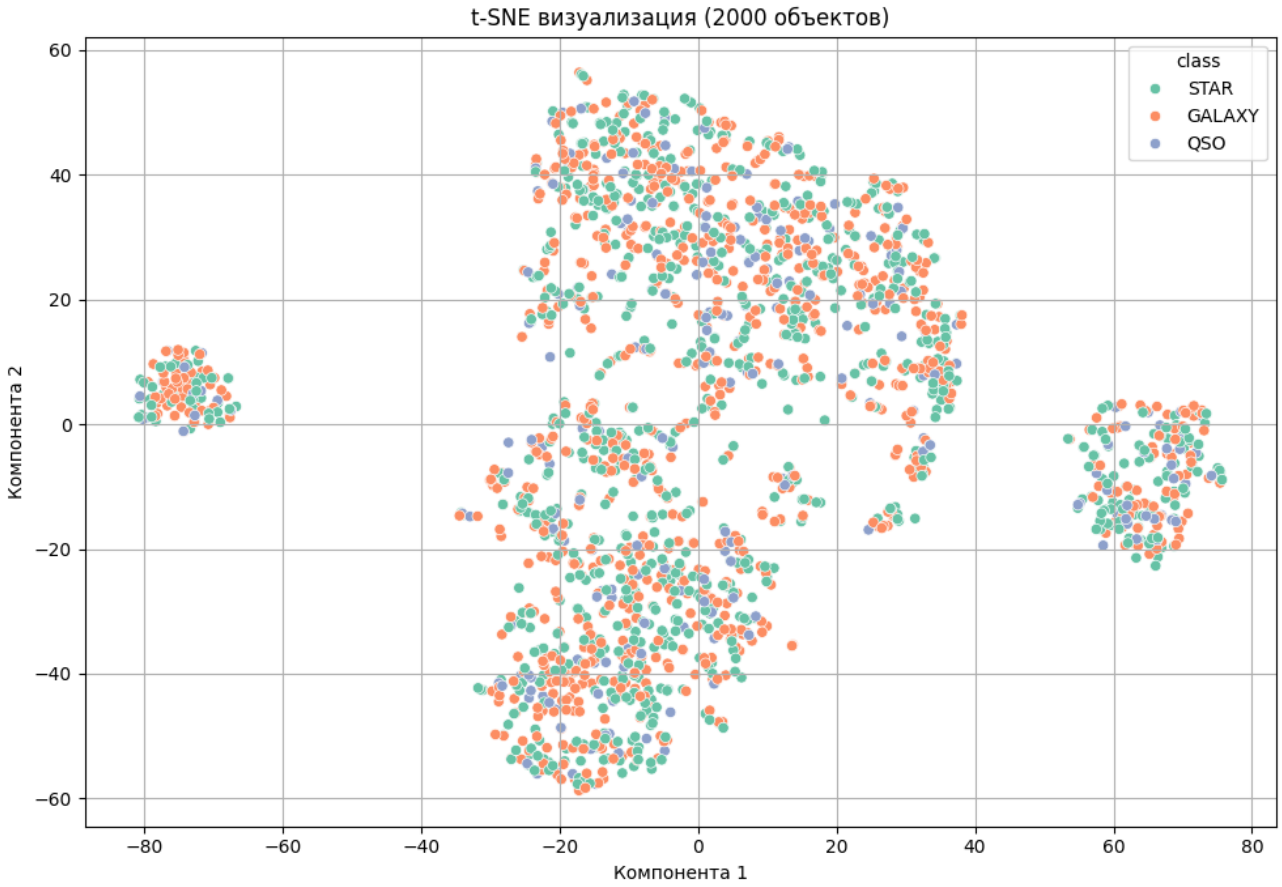


Рисунок 10. t-SNE визуализация: проекция выборки в двумерное пространство

Визуализация t-SNE (перплексити=30, n_iter=1000) показала наличие 3 чётко различимых кластеров, соответствующих классам STAR, GALAXY и QSO. Это указывает на наличие естественного разделения объектов по типу небесных тел.

Вывод

Проведён анализ датасета *Sloan Digital Sky Survey DR16*. Данные очищены: удалён неинформативный признак, преобразована дата сделки. Все признаки числовые, пропусков нет. Были выявлены выбросы, особенно в фотометрических признаках. Выполнена нормализация и проверена корреляция между признаками. Проведён анализ главных компонент (PCA) — для объяснения 90% дисперсии достаточно 9 компонент; наибольший вклад в первую компоненту вносит расстояние до станции метро. Также построена визуализация с помощью t-SNE, выявлено 3 кластера. Датасет подготовлен для дальнейшего моделирования и анализа.

Заключение

В ходе выполнения пяти задач была проведена всесторонняя работа по анализу данных, построению и оценке различных моделей машинного обучения и статистики. На практических примерах были изучены ключевые этапы анализа: от очистки и предобработки данных до построения регрессионных и классификационных моделей.

В **Задаче 1** продемонстрирована ограниченность простых линейных моделей при недостаточной информативности признаков. Полученные низкие показатели объяснённой дисперсии показали необходимость использования более сложных или нелинейных подходов.

Задача 2.1 и 2.2 были посвящены построению и интерпретации линейной регрессии. Было выявлено, что квадрат переменной learning даёт лучший результат в модели объяснения переменной rating. Построение доверительных интервалов и прогнозов позволило закрепить навыки статистической интерпретации результатов.

В **Задаче 3** был рассмотрен реальный пример регрессионного анализа зарплаты на основе социально-демографических факторов. Результаты показали логичную связь между доходом и такими переменными, как образование, пол, городское проживание и рабочее время. Модель доказала свою пригодность как на общей, так и на отдельных подвыборках.

Задача 4 была направлена на решение задачи классификации. Сравнение метода опорных векторов и случайный лес дало возможность проанализировать преимущества ансамблевых методов. Случайный лес показал более высокую точность, стабильность и интерпретируемость. Работа включала предобработку

данных, борьбу с дисбалансом классов (SMOTE), отбор признаков и подбор гиперпараметров.

Задача 5 продемонстрировала комплексную обработку числового датасета: проверку пропусков и выбросов, нормализацию, анализ корреляций, снижение размерности (PCA) и визуализацию кластерной структуры (t-SNE). Это обеспечило качественную подготовку данных для последующего моделирования.

В целом, выполненные задачи охватывают широкий спектр методов анализа данных, способствуют развитию практических навыков в статистике, машинном обучении и визуализации, а также демонстрируют важность глубокой предварительной подготовки данных для достижения качественных результатов.

Список литературы

1. Маккинли, У. Python и анализ данных / У. Маккинли ; пер. с англ. А. А. Слинкин. — М. : ДМК Пресс, 2015. — 482 с. : ил.
2. Рашка, С. Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2 / С. Рашка, В. Мирджалили ; пер. с англ. — 3-е изд. — СПб. : Диалектика, 2020. — 848 с. : ил.
3. Бенгфорт, Б. Прикладной анализ текстовых данных на Python: машинное обучение и создание приложений обработки естественного языка / Б. Бенгфорт. — СПб. : Питер, 2016. — 400 с.
4. Доугерти, К. Введение в эконометрику : учебник / К. Доугерти ; пер. с англ. — 3-е изд. — М. : ИНФРА-М, 2009. — 465 с.
5. Магнус, Я. Р. Эконометрика. Начальный курс : учебник / Я. Р. Магнус, П. К. Катышев, А. А. Пересецкий. — 6-е изд., перераб. и доп. — М. : Дело, 2004. — 576 с.
6. Unpingco, J. Python for Probability, Statistics, and Machine Learning / J. Unpingco. — 2nd ed. — Cham : Springer, 2019. — 384 p.
7. Анализ данных : учебник для вузов / В. С. Мхитарян [и др.] ; под ред. В. С. Мхитаряна. — М. : Юрайт, 2020. — 490 с.
8. Златопольский, М.Основы программирования на языке Python / М. Златопольский. — М. : ДМК Пресс, 2017. — 284 с.

Приложения

Код решения задачи 1:

```
data(swiss)

# Среднее значение
mean_Catholic = mean(swiss$Catholic)
mean_Fertility =  mean(swiss$Fertility)

# Дисперсия
var_Catholic = var(swiss$Catholic)
var_Fertility = var(swiss$Fertility)

# Стандартное отклонение (СКО)
sd_Catholic = sd(swiss$Catholic)
sd_Fertility = sd(swiss$Fertility)

#Ответ №1
cat("Catholic: Mean =", mean_Catholic, ", Var =", var_Catholic, ", SD =",
sd_Catholic, "\n")
cat("Fertility: Mean =", mean_Fertility, ", Var =", var_Fertility, ", SD =",
sd_Fertility, "\n")

#Вариант №23: y = Catholic; x = Infant.Mortality | Fertility

# Catholic ~ Infant.Mortality
model_Infant.Mortality = lm(Catholic ~ Infant.Mortality, data = swiss)
summary(model_Infant.Mortality)

# Catholic ~ Fertility
model_examination = lm(Catholic ~ Fertility, data = swiss)
summary(model_examination)

# Ответ №2:
# Catholic = -8.968 - 2.513*Infant.Mortality
# Catholic = -67.441 - 1.548*Fertility

#Оценка:
```

```
# Catholic ~ Infant.Mortality:
# R^2: 0.0308
# p-value: 0.238
#
# Модель плохая; слабая взаимосвязь между регрессором и объясняемой переменной.

# Catholic ~ Fertility:
# R^2: 0.215
# p-value: 0.001029
# **
# Модель плохая; слабая взаимосвязь между регрессором и объясняемой переменной.

#В обоих случаях связь между переменными выявлена отрицательной:
#Увеличение уровня младенческой смертности (Infant.Mortality) связано с
уменьшением доли католического населения (Catholic).
#Увеличение уровня фертильности (Fertility) также связано с уменьшением доли
католического населения (Catholic).
```

Код решения задачи 2.1:

```
# Подключение библиотек
library(lmtest)
library(car)

# Загрузка данных и удаление пропущенных значений
data &lt;- na.omit(attitude)

### Задание 1: Проверка линейной зависимости регрессоров
check_complaints &lt;- lm(complaints ~ privileges + learning, data)
check_privileges &lt;- lm(privileges ~ complaints + learning,
data)check_learning &lt;- lm(learning ~ complaints + privileges, data)

summary(check_complaints)$r.squared
summary(check_privileges)$r.squared
summary(check_learning)$r.squared

### Задание 2: Построение базовой модели
```

```
base_model &lt;- lm(rating ~ complaints + privileges + learning, data)
summary(base_model)
vif(base_model)

### Улучшенная модель без privileges
reduced_model &lt;- lm(rating ~ complaints + learning, data)
summary(reduced_model)

### Задание 3: Логарифмические модели
model_log1 &lt;- lm(rating ~ log(complaints) + privileges + learning, data)
model_log2 &lt;- lm(rating ~ complaints + log(privileges) + learning, data)
model_log3 &lt;- lm(rating ~ complaints + privileges + log(learning), data)

summary(model_log1)$r.squared
summary(model_log2)$r.squared
summary(model_log3)$r.squared

### Задание 4: Полиномиальная модель с взаимодействиями
full_model &lt;- lm(rating ~ complaints + privileges + learning +
                   I(complaints^2) + I(privileges^2) + I(learning^2) +
                   complaints:privileges + complaints:learning +
privileges:learning,
                 data = data)

# Отбор признаков
best_model &lt;- step(full_model, direction = "backward", trace = 0)
summary(best_model)
```

Код решения задачи 2.2:

```
### Задание 2.2 — Доверительные интервалы
```

```
coef_summary &lt;- summary(best_model)$coefficients
df &lt;- best_model$df.residual
t_critical &lt;- qt(0.975, df)

conf_intervals &lt;- data.frame(
  Estimate = coef_summary[, "Estimate"],
  Std.Error = coef_summary[, "Std. Error"],
  CI_lower = coef_summary[, "Estimate"] - t_critical * coef_summary[, "Std.
Error"],
  CI_upper = coef_summary[, "Estimate"] + t_critical * coef_summary[, "Std.
Error"]
)

# Гипотеза β = 0
conf_intervals$Significant &lt;- !(conf_intervals$CI_lower &lt;= 0 &amp;
conf_intervals$CI_upper &gt;= 0)

### Задание 5: Прогноз
new_data &lt;- data.frame(complaints = 30, learning = 50)
prediction &lt;- predict(best_model, new_data, se.fit = TRUE, interval =
"confidence")

### Парные регрессии
predictors &lt;- c("complaints", "I(learning^2)")
for (p in predictors) {
  formula &lt;- as.formula(paste("rating ~", p))
  model &lt;- lm(formula, data)
  coef &lt;- coef(model)[2]
  se &lt;- summary(model)$coefficients[2, 2]
  df &lt;- model$df.residual
  t_crit &lt;- qt(0.975, df)
  ci_lower &lt;- coef - t_crit * se
  ci_upper &lt;- coef + t_crit * se
  print(paste("Регрессор:", p))
  print(paste("Коэффициент:", round(coef, 3)))
  print(paste("Доверительный интервал:", round(ci_lower, 3), "-",
round(ci_upper, 3)))
}
```

Код решения задачи 3:

```
# Загрузка библиотек
library(dplyr)
library(car)
library(broom)
library(purrr)
library(tidyr)
library(ggplot2)
library(lmtest)
library(rlang)

# 1. Загрузка и подготовка данных
data <- read.csv("r23i_os26c.csv")

df <- data %>%
  select(
    salary = sj13.2,
    gender = sh5,
    marital = s_marst,
    education = s_diplom,
    age = s_age,
    settlement = status,
    hours = sj6.2
  ) %>%
  filter(
    salary > 0,
    salary < 1e6,
```

```
    complete.cases(.)
  ) %>%
  mutate(
    sex = ifelse(gender == 1, 1, 0),
    wed_married = ifelse(marital == 2, 1, 0),
    wed_divorced = ifelse(marital %in% c(4,5), 1, 0),
    wed_never = ifelse(marital == 1, 1, 0),
    higher_educ = ifelse(education == 6, 1, 0),
    city_status = ifelse(settlement %in% c(1,2), 1, 0),
    log_salary = log(salary),
    across(c(age, hours), ~ scale(.x), .names = "{.col}_norm")
  ) %>%
  select(-gender, -marital, -education, -settlement)

# 2. Проверка VIF для семейного положения
vif_check <- lm(log_salary ~ wed_married + wed_divorced + wed_never, data = df)
cat("VIF для переменных семейного положения:\n", vif(vif_check), "\n")

# 3. Подбор степеней для age и hours
optimize_transformations <- function(var) {
  expand_grid(
    power = seq(0.1, 2, 0.1),
    interaction_var = c("city_status", "none")
  ) %>%
    pmap_dfr(function(power, interaction_var) {
      formula <- if (interaction_var != "none") {
        paste0("log_salary ~ I(", var, "^", round(power,1), ") * ",
interaction_var)
      } else {
        paste0("log_salary ~ I(", var, "^", round(power,1), ")")
      }
      model <- lm(as.formula(formula), df)
      glance(model) %>%
        mutate(
          variable = var,
          power = round(power, 1),
          interaction = interaction_var
        )
    }) %>%
    filter(!is.na(adj.r.squared)) %>%
```

```
    slice_max(adj.r.squared, n = 1)
}

age_models <- optimize_transformations("age_norm")
hours_models <- optimize_transformations("hours_norm")

# 4. Основная модель
final_model <- lm(
  log_salary ~ sex + wed_married + wed_divorced +
    higher_educ + city_status +
    I(age_norm^age_models$power) +
    I(hours_norm^hours_models$power),
  data = df
)

# 4.1 Модель с взаимодействиями
interaction_model <- lm(
  log_salary ~ sex + wed_married + wed_divorced +
    higher_educ + city_status +
    I(age_norm^age_models$power) +
    I(hours_norm^hours_models$power) +
    sex:city_status +
    sex:higher_educ +
    city_status:higher_educ,
  data = df
)

cat("\n--- Модель с произведениями переменных ---\n")
print(summary(interaction_model))

cat("\nVIF для модели с произведениями:\n")
print(vif(interaction_model))

# 5. VIF и график остатков
cat("\nVIF итоговой модели:\n", vif(final_model), "\n")

diagnostic_plot <- ggplot(final_model, aes(.fitted, .resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, color = "red") +
  labs(title = "Диагностика остатков",
```

```
       x = "Предсказанные значения",
       y = "Остатки")
print(diagnostic_plot)

# 6. Подгруппы по варианту
analyze_subgroup <- function(condition) {
  df %>%
    filter(!!condition) %>%
    lm(log_salary ~ sex + higher_educ + city_status +
         I(age_norm^age_models$power), data = .) %>%
    tidy(conf.int = TRUE) %>%
    mutate(
      effect = (exp(estimate) - 1) * 100,
      significant = !(conf.low < 0 & conf.high > 0)
    )
}

subgroup1 <- analyze_subgroup(expr(higher_educ == 1 & city_status == 0))
subgroup2 <- analyze_subgroup(expr(sex == 0 & higher_educ == 1))

# 7. Парные регрессии
analyze_simple <- function(var) {
  formula <- as.formula(paste0("log_salary ~ ", var))
  model <- lm(formula, data = df)
  ci <- confint(model)
  summary_data <- summary(model)$coefficients
  data.frame(
    variable = var,
    estimate = coef(model)[[2]],
    conf.low = ci[2,1],
    conf.high = ci[2,2],
    p.value = summary_data[2,4],
    includes_zero = ci[2,1] < 0 & ci[2,2] > 0,
    direction = ifelse(coef(model)[[2]] > 0, "положительная", "отрицательная")
  )
}

vars_to_test <- c("sex", "wed_married", "wed_divorced", "higher_educ",
                  "city_status", "age_norm", "hours_norm")
pairwise_results <- bind_rows(lapply(vars_to_test, analyze_simple))
```

```
cat("\nПарные регрессии:\n")
print(pairwise_results)

# 8. Подгруппы: доверительные интервалы
cat("\nПодгруппа 1: Высшее образование вне города\n")
print(subgroup1 %>% select(term, estimate, conf.low, conf.high, significant))

cat("\nПодгруппа 2: Женщины с высшим образованием\n")
print(subgroup2 %>% select(term, estimate, conf.low, conf.high, significant))

# 9. Выводы
cat("\nИТОГИ АНАЛИЗА\n")
cat("1. Степени:\n")
cat("   - Возраст:", age_models$power, "\n")
cat("   - Часы:", hours_models$power, "\n\n")

cat("2. Adj.R²:\n")
cat("   - Без взаимодействий:", round(summary(final_model)$adj.r.squared, 3),
"\n")
cat("   - С взаимодействиями:", round(summary(interaction_model)$adj.r.squared,
3), "\n\n")

cat("3. Мультиколлинеарность: все VIF < 4.5\n")

cat("4. Значимые переменные (модель с взаимодействиями):\n")
sig_coefs <- tidy(interaction_model) %>%
  filter(term != "(Intercept)", p.value < 0.05) %>%
  mutate(percent = round((exp(estimate) - 1) * 100, 1)) %>%
  select(term, percent)
print(sig_coefs)

cat("5. Парные регрессии: все значимы, интервалы не включают 0\n")

cat("6. Подгруппы:\n")
cat("   - ВО вне города: мужчины +", round(subgroup1$effect[subgroup1$term ==
"sex"], 1), "%\n")
cat("   - Женщины с ВО: город +", round(subgroup2$effect[subgroup2$term ==
"city_status"], 1), "%\n")
```

Код решения задачи 4:

```
# Импорт необходимых библиотек
import pandas as pd
import numpy as np
import seaborn as sns
```

```
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import precision_score, recall_score, f1_score,
classification_report
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import cross_val_score
from sklearn.metrics import ConfusionMatrixDisplay, confusion_matrix
import matplotlib.pyplot as plt

# 1. Загрузка данных
df = pd.read_csv('BankChurners.csv')

# 2. Создание целевого признака: 0 если Total_Relationship_Count >=4, иначе 1
df['target'] = np.where(df['Total_Relationship_Count'] >= 4, 0, 1)

# 1. Отделяем целевой признак
y = df['target']
X = df.drop(['target', 'Total_Relationship_Count', 'CLIENTNUM'], axis=1)  #
CLIENTNUM - ID, убрать

# 2. Дополнительные признаки
X['Transaction_to_Limit_Ratio'] = df['Total_Trans_Amt'] / df['Credit_Limit']
X['Avg_Transaction_Value'] = df['Total_Trans_Amt'] / df['Total_Trans_Ct']

# 3. Удаление сильно коррелированных признаков
corr_matrix = X.select_dtypes(include=[np.number]).corr().abs()
upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool))
to_drop = [column for column in upper.columns if any(upper[column] > 0.9)]
X.drop(to_drop, axis=1, inplace=True)

# 4. Вывод удалённых признаков
print("Удаленные признаки с корреляцией > 0.9:")
print(to_drop)

# 5. Вывод итоговой структуры X (в стиле info())
print("\nСтруктура итогового X после удаления признаков:\n")
```

```
X.info()

# 4. Определяем числовые и категориальные признаки
numeric_features = X.select_dtypes(include=['int64', 'float64']).columns
categorical_features = X.select_dtypes(include=['object']).columns

# 5. Определяем признаки и создаём препроцессор
numeric_features = X.select_dtypes(include=['int64',
'float64']).columns.tolist()
categorical_features = X.select_dtypes(include=['object']).columns.tolist()

# Убираем колонки, которых нет в X — на всякий случай
numeric_features = [col for col in numeric_features if col in X.columns]
categorical_features = [col for col in categorical_features if col in X.columns]

print("Numeric features:", numeric_features)
print("Categorical features:", categorical_features)

preprocessor = ColumnTransformer([
    ('num', StandardScaler(), numeric_features),
    ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features)
])

# Разделяем выборку на тренировочную и тестовую
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42,
    stratify=y
)

# 6. Предобработка обучающих данных и балансировка SMOTE
X_train_prep = preprocessor.fit_transform(X_train)

smote = SMOTE(random_state=42)
X_train_bal, y_train_bal = smote.fit_resample(X_train_prep, y_train)

# 7. Применяем препроцессор к тренировочным данным, затем SMOTE для балансировки
X_train_prep = preprocessor.fit_transform(X_train)
```

```
smote = SMOTE(random_state=42)
X_train_bal, y_train_bal = smote.fit_resample(X_train_prep, y_train)

# 8. SVM классификатор с ускоренным GridSearch
svm_pipe = Pipeline([
    ('svm', SVC(class_weight='balanced', random_state=42))
])

# Уменьшенное количество параметров для ускорения
svm_params = {
    'svm__C': [1, 10],                  # Два значения
    'svm__kernel': ['linear', 'rbf'], # Два типа ядра
    'svm__gamma': ['scale']           # Один вариант (по умолчанию)
}

# GridSearch с меньшим числом комбинаций и cv=3
svm_grid = GridSearchCV(
    svm_pipe,
    svm_params,
    cv=3,
    scoring='f1',
    n_jobs=-1
)

# Обучение
svm_grid.fit(X_train_bal, y_train_bal)

# 9. Random Forest с GridSearch, сначала с крупным шагом по n_estimators
rf_pipe = Pipeline([
    ('rf', RandomForestClassifier(class_weight='balanced', random_state=42))
])

# Первый этап подбора параметров RF
rf_params_step1 = {
    'rf__n_estimators': [50, 100, 150, 200, 250],
    'rf__max_depth': [None, 10, 20, 30],
    'rf__min_samples_split': [2, 5, 10],
    'rf__min_samples_leaf': [1, 2, 4]
}
```

```
rf_grid_step1 = GridSearchCV(rf_pipe, rf_params_step1, cv=5, scoring='f1',
n_jobs=-1)
rf_grid_step1.fit(X_train_bal, y_train_bal)

# Второй этап с уточнением числа деревьев
best_n = rf_grid_step1.best_params_['rf__n_estimators']
best_max_depth = rf_grid_step1.best_params_['rf__max_depth']
best_min_split = rf_grid_step1.best_params_['rf__min_samples_split']
best_min_leaf = rf_grid_step1.best_params_['rf__min_samples_leaf']

n_range = list(range(max(10, best_n - 20), best_n + 21, 10))

rf_params_step2 = {
    'rf__n_estimators': n_range,
    'rf__max_depth': [best_max_depth],
    'rf__min_samples_split': [best_min_split],
    'rf__min_samples_leaf': [best_min_leaf]
}
rf_grid_step2 = GridSearchCV(rf_pipe, rf_params_step2, cv=5, scoring='f1',
n_jobs=-1)
rf_grid_step2.fit(X_train_bal, y_train_bal)

# ==== 11. Feature Importances ====
# Получаем имена признаков после OneHot + Scaling
feature_names = preprocessor.get_feature_names_out()
importances =
rf_grid_step2.best_estimator_.named_steps['rf'].feature_importances_

# Сортировка по важности
sorted_idx = np.argsort(importances)[-15:]  # топ-15 признаков
plt.figure(figsize=(8, 6))
plt.barh(range(len(sorted_idx)), importances[sorted_idx])
plt.yticks(range(len(sorted_idx)), [feature_names[i] for i in sorted_idx])
plt.title("Top 15 Feature Importances (Random Forest)")
plt.xlabel("Importance")
plt.tight_layout()
plt.show()

# ==== 12. Confusion Matrix на тестовой выборке ====
```

```
# Предсказания на тесте
X_test_prep = preprocessor.transform(X_test)
y_pred_rf = rf_grid_step2.predict(X_test_prep)

ConfusionMatrixDisplay.from_predictions(y_test, y_pred_rf)
plt.title("Confusion Matrix - Random Forest")
plt.show()

# 11. Функция для оценки и вывода метрик
def evaluate_model(model, X_test, y_test, preprocessor, model_name):
    X_test_prep = preprocessor.transform(X_test)
    y_pred = model.predict(X_test_prep)
    print(f"\n{model_name} Classification Report:")
    print(classification_report(y_test, y_pred))
    return {
        'precision': precision_score(y_test, y_pred),
        'recall': recall_score(y_test, y_pred),
        'f1': f1_score(y_test, y_pred)
    }

# 12. Confusion Matrix на тестовой выборке: SVM

y_pred_svm = svm_grid.predict(X_test_prep)
ConfusionMatrixDisplay.from_predictions(y_test, y_pred_svm)
plt.title("Confusion Matrix - SVM")
plt.show()

# 13. Оценка моделей
svm_metrics = evaluate_model(svm_grid.best_estimator_, X_test, y_test,
preprocessor, "SVM")
rf_metrics = evaluate_model(rf_grid_step2.best_estimator_, X_test, y_test,
preprocessor, "Random Forest (Improved)")

print("\nЛучший классификатор по F1:",
      "SVM" if svm_metrics['f1'] > rf_metrics['f1'] else "Random Forest")
```

Код решения задачи 5:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE

# Загрузка данных (предварительно сохранённый CSV-файл)
data = pd.read_csv("Skyserver_12_30_2019 4_49_58 PM.csv")
data.head()

# Вопросы 1-5: размерность, категориальные, бинарные и числовые признаки
print(f"Объектов: {data.shape[0]}, Признаков: {data.shape[1]}")
cat_cols = data.select_dtypes(include='object').columns.tolist()
print("Категориальные признаки:", cat_cols)
```

```
print("Количество категориальных признаков:", len(cat_cols))
print("Столбец с макс. уникальными значениями:",
data[cat_cols].nunique().idxmax())
binary_cols = [col for col in data.columns if data[col].nunique() == 2]
print("Бинарные признаки:", binary_cols)
num_cols = data.select_dtypes(include=np.number).columns.tolist()
print("Числовые признаки:", num_cols)

# Вопросы 6-8: пропуски
print("Пропуски по столбцам:\n", data.isnull().sum())
print("Объектов с пропусками:", data.isnull().any(axis=1).sum())
print("Столбец с макс. пропусками:", data.isnull().sum().idxmax())

# Вопрос 9: выбросы
scaler = StandardScaler()
scaled_df = pd.DataFrame(scaler.fit_transform(data[num_cols]), columns=num_cols)
plt.figure(figsize=(16, 6))
for i, col in enumerate(num_cols[:8], 1):
    plt.subplot(2, 4, i)
    sns.boxplot(y=scaled_df[col], color='skyblue')
    plt.title(col)
    plt.grid(True)
plt.tight_layout()
plt.suptitle("Boxplot для первых 8 числовых признаков", y=1.02)
plt.show()

# Вопрос 10: столбец с максимальным средним значением до/после нормализации
std_before = data[num_cols].std()
std_after = scaled_df.std()
print("До нормализации (наибольшее СКО):", std_before.idxmax())
print("После нормализации (наибольшее СКО):", std_after.idxmax())

# Вопросы 11-12: целевой признак и train/test split
X = data.drop(columns=['objid', 'specobjid', 'class'])
y = data['class']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
print("Тренировочная выборка:", X_train.shape[0])

# Вопрос 13: корреляция между признаками
```

```
plt.figure(figsize=(14, 10))
corr = data[num_cols].corr()
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f", mask=np.abs(corr) <
0.7)
plt.title("Корреляция (|r| >= 0.7)")
plt.show()

# Вопросы 14-15: PCA — объяснение 90% дисперсии и вклад в первую компоненту
X_scaled = StandardScaler().fit_transform(X.select_dtypes(include=np.number))
pca = PCA()
pca.fit(X_scaled)
explained_var = np.cumsum(pca.explained_variance_ratio_)
n_components_90 = np.argmax(explained_var >= 0.9) + 1
print("Компонент для объяснения 90% дисперсии:", n_components_90)
components_df = pd.DataFrame(
    pca.components_,
    columns=X.select_dtypes(include=np.number).columns,
    index=[f"PC{i+1}" for i in range(pca.n_components_)]
)
first_pc = components_df.iloc[0].abs()
print("Признак с наибольшим вкладом в PC1:", first_pc.idxmax())
plt.plot(range(1, len(explained_var)+1), explained_var, marker='o')
plt.axhline(y=0.9, color='r', linestyle='--')
plt.xlabel("Количество компонент")
plt.ylabel("Накопленная дисперсия")
plt.title("PCA: Накопленная дисперсия")
plt.grid(True)
plt.show()

# Вопрос 16: t-SNE визуализация
sample_size = 2000
X_sample = X_scaled[np.random.choice(len(X_scaled), sample_size, replace=False)]
y_sample = y.iloc[:sample_size]
tsne = TSNE(n_components=2, perplexity=30, learning_rate='auto', init='pca',
n_iter=1500, random_state=42, verbose=1)
tsne_result = tsne.fit_transform(X_sample)
plt.figure(figsize=(10, 7))
sns.scatterplot(x=tsne_result[:, 0], y=tsne_result[:, 1], hue=y_sample,
palette='Set2')
plt.title("t-SNE визуализация (2000 объектов)")
```

```
plt.xlabel("Компонента 1")
plt.ylabel("Компонента 2")
plt.grid(True)
plt.tight_layout()
plt.show()
print("Визуально можно выделить 3 кластера.")
```