МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

|  |  |
| --- | --- |
| Институт (факультет) | Институт информационных технологий |
| Кафедра | МПО ЭВМ |

КУРСОВАЯ РАБОТА

|  |  |
| --- | --- |
| по дисциплине | Математические методы решения задач |
|  | искусственного интеллекта |
| на тему | Математические методы в машинном обучении |
|  | |

|  |
| --- |
| Выполнил студент группы |
| 1ПИб-02-1оп-22 |
| *направления подготовки (специальности)* |
| 09.03.04 Программная инженерия |
| *шифр, наименование* |
| Микуцких Григорий Андреевич |
| *фамилия, имя, отчество* |

|  |
| --- |
| Руководитель |
| Юдина О.В. |
| *фамилия, имя, отчество* |
| доцент |
| *должность* |

|  |
| --- |
| Дата представления работы |
| «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г. |
|  |
| Заключение о допуске к защите |
|  |
|  |
|  |
|  |
| Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| количество баллов |
| Подпись преподавателя\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |
| --- | --- |
| Череповец, | 2024 |
|  | *год* |

Аннотация

Курсовая работа посвящена изучению методов машинного обучения и математического анализа.

В ходе работы была написана программа на языке программирования Python, проанализирован набор данных о продажах, проведена кластеризация, классификация, регрессионный анализ, оценены точность и адекватность результатов, графически изображены результаты исследований.

Оглавление

[Введение 4](#_Toc185845184)

[1. Теоретическая часть 5](#_Toc185845185)

[1.1. Описание данных 5](#_Toc185845186)

[1.2. Формирование своих признаков 7](#_Toc185845187)

[1.3. Возможные задачи машинного обучения и планируемые результаты 7](#_Toc185845188)

[1.4. Выбор методов оценивания результатов 10](#_Toc185845189)

[2. Практическая часть 13](#_Toc185845190)

[2.1. Программное решение 13](#_Toc185845191)

[2.2. Задача классификации 14](#_Toc185845192)

[2.3. Задача кластеризации 15](#_Toc185845193)

[2.4. Задача регрессии 17](#_Toc185845194)

[Заключение 28](#_Toc185845195)

[Список литературы 29](#_Toc185845196)

[Приложение 1. Техническое задание 30](#_Toc185845197)

[Приложение 2. Текст программы 36](#_Toc185845199)

## Введение

Современный этап развития информационных технологий характеризуется постоянным ростом потребности в изучении больших данных как для компаний, так и для активных энтузиастов. С каждым годом производительность компьютеров растёт, а границы познания раздвигаются всё больше, открывая новые просторы в изучении больших данных.

Цель курсовой работы: изучение методов машинного обучения и применение их на практике.

Задачи:

* изучить методы машинного обучения и математического анализа;
* найти реальные данные для анализа;
* выдвинуть гипотезы по данным;
* построить модель для прогнозирования;
* обучить модель, оценить её точность и адекватность;
* интерпретировать результаты и определить какие задачи хотели бы решить специалисты в данной предметной области.

План решения: написать программу на языке Python для анализа данных по онлайн-продажам.

1. Теоретическая часть

## Описание данных

Набор данных онлайн-продажи взят с ресурса Kaggle [3]. Набор содержит анонимные данные о транзакциях онлайн-продаж, отражающие различные аспекты покупок товаров, информацию о клиентах и характеристики заказов.

В данных представлены поля (признаки): категориальные и числовые.

К категориальным переменным относятся:

* InvoiceNo – уникальный идентификатор для каждого счёта-фактуры;
* InvoiceDate – дата выставления счёта (дата и время, когда была зарегистрирована продажа);
* StockCode – товарный код (единица хранения товара на складе – SKU);
* Description – краткое описание (название) продукта;
* CustomerID – уникальный идентификатор для каждого клиента;
* Country – страна клиента;
* PaymentMethod – способ оплаты, использованный для транзакции (например, PayPal, банковский перевод);
* Category – категория (тип) товара;
* SalesChannel – канал продаж (онлайн или магазин);
* ReturnStatus – статус возврата товара;
* ShipmentProvider – поставщик услуг доставки;
* WarehouseLocation – местоположение склада (город);
* OrderPriority – приоритет (срочность) заказа.

К числовым переменным относятся:

* Quantity – количество единиц товара, проданного в рамках транзакции;
* UnitPrice – цена за единицу товара в валюте транзакции;
* Discount – скидка, применяемая к сделке, если таковая имеется;
* ShippingCost – стоимость доставки.

Прежде чем начать работу с данными, нужно разбить их на отдельные ячейки по разделителю запятой, общий вид данных в Excel представлен на рис. 1.

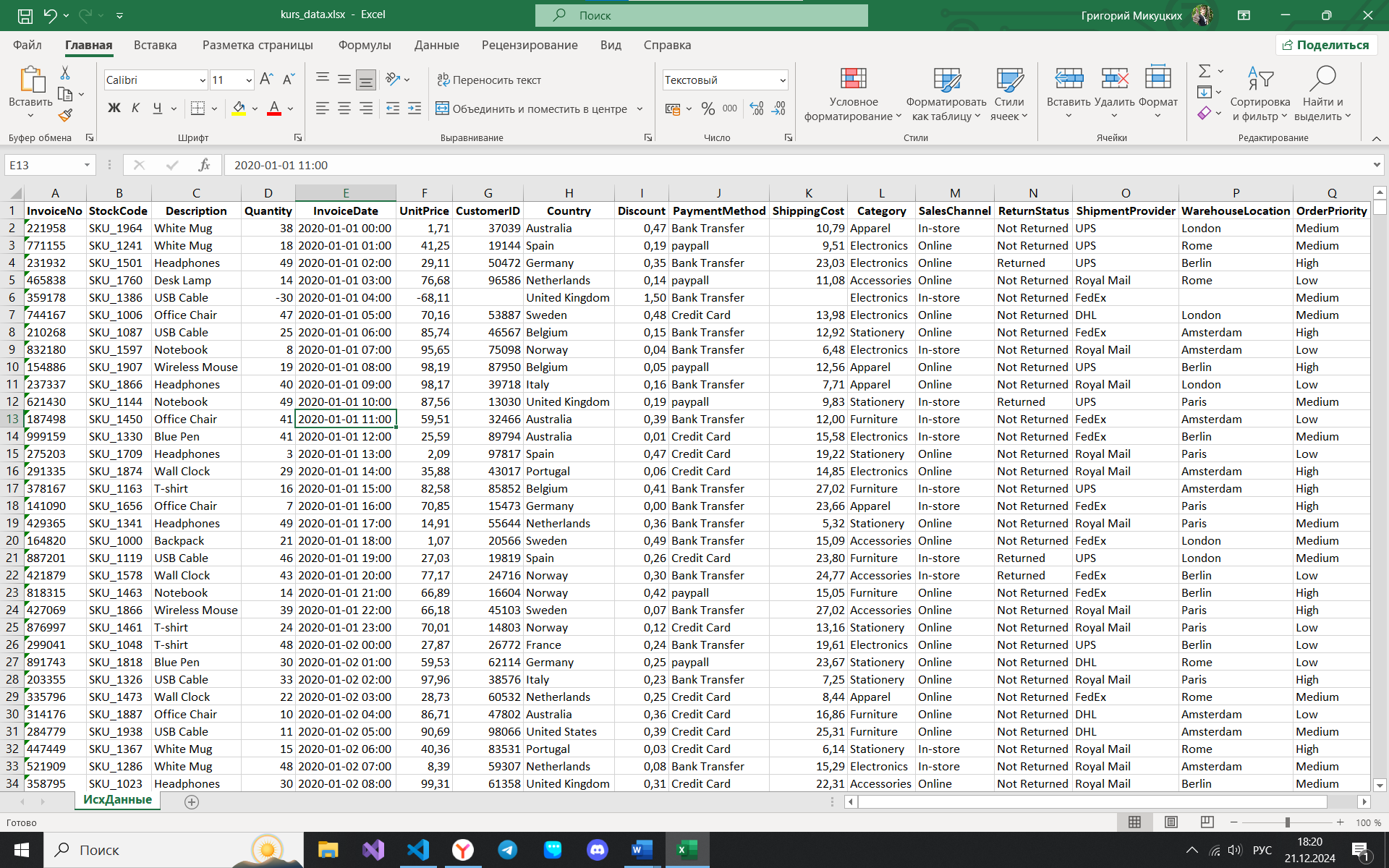


Рис. 1. Общий вид данных по онлайн-продажам

При кратком осмотре записей можно заметить, что присутствуют пустые ячейки, в числовых столбцах замечены отрицательные значения, поле «Discount» содержит значения от 0 до 2, что говорит об неправильных транзакциях.

Также есть данные, не имеющие ценности для анализа – «InvoiceNo» и «CustomerID», так как они в большей степени уникальные.

Исходя из вышеперечисленных проблем, будут проведены следующие манипуляции по предварительной обработке:

1. Пропуски в числовых признаках будут заменены на моду, однако в категориальных признаках значение не подобрать, поэтому такие записи не будут использоваться при обучении моделей.
2. Записи с отрицательными числовыми переменными и записи, у которых поле «Discount» содержит число больше 1 (то есть, скидка больше 100%), не будут использоваться при обучении моделей.
3. Поля «InvoiceNo» и «CustomerID» не будут учитываться в исследовании.

Данные онлайн-продаж могут помочь:

1. Проанализировать тенденции продаж, покупательское поведение клиентов и управление заказами, что даст понимание влияния скидок, способов оплаты и поставщиков услуг доставки на эффективность продаж и удовлетворенность клиентов.
2. Проанализировать эффективность продаж, структуру покупок клиентов и операционную эффективность при управлении для оценки влияния скидок и способов оплаты на продажи, оптимизации запасов за счёт изучения спроса на продукцию и повышения удовлетворенности клиентов за счёт улучшения процесса доставки и возврата товара.

## Формирование своих признаков

Чтобы сделать больше общих показателей и упростить исследование записей, будут произведены группировки полей по:

* размеру скидки на интервалы: 0 – 10, 10 – 30, 30 – 50, 50 – 100;
* стоимости доставки на интервалы: 0 – 5, 5 – 15, 15 – 30;
* цене товара за одну штуку на интервалы: 0 – 10, 10 – 30, 30 – 50, 50 – 100.

Также будет сформирован новый признак «Sum» – итоговая сумма заказа с учётом скидки и цены доставки.

## Возможные задачи машинного обучения и планируемые результаты

Согласно Гипотезе Компактности (объекты, принадлежащие одному классу, скорее всего, будут находиться близко друг к другу в пространстве признаков) можно предположить, что среди имеющихся данных могут быть несколько схожих по своей природе транзакций.

Предположительно, транзакции можно классифицировать по размеру скидки, возврату товара и стране заказа (потребность того или иного региона в определённом продукте).

Предположительно наличие нелинейной зависимости между признаками, можно попробовать найти закономерности и коэффициенты регрессии.

Исходя из вышесказанного, можно выдвинуть следующие гипотезы:

* в разных странах преобладает потребность по разным товарам;
* чем больше сумма заказа, тем больше стоимость доставки;
* чем больше сумма заказа, тем картой платят чаще;
* электронику покупают чаще всего онлайн;
* возвраты чаще всего делают при покупке в магазине.

Метрические алгоритмы регрессии в машинном обучении – это методы, основанные на гипотезе непрерывности (близким объектам соответствуют близкие ответы).

Регрессия – это метод для моделирования отношения между зависимой (или объясняемой) переменной и одной или несколькими независимыми (или объясняющими) переменными. Основная цель регрессионного анализа — предсказание значений зависимой переменной на основе значений независимой переменной или переменных [7].

1. Линейная регрессия – это метод регрессии, который предполагает линейную зависимость между зависимой переменной и одной или несколькими независимыми переменными (1) [7].
2. Полиномиальная регрессия – это расширение линейной регрессии, которое позволяет моделировать нелинейные отношения между переменными. Вместо линейной функции используется полином (например, квадратичный, кубический ит. д.) для аппроксимации данных, то есть замены одних объектов другими, в каком-то смысле близкими к исходным, но более простым. Это позволяет более точно моделировать сложные формы зависимости (2) [7].
3. Логистическая регрессия – это метод классификации, а не регрессии в строгом смысле. Она используется для прогнозирования бинарных или многоклассовых категорий. Это очень популярный метод в задачах классификации (3) [7].
4. Пробит регрессия – это статистическая техника регрессионного анализа, используемая в ситуациях, когда зависимая переменная является двоичной (принимает только два значения «успешно» или «не успешно»). Особенно полезна в эконометрике и других областях прикладной статистики для моделирования ситуаций. Её отличие от логит-модели состоит в том, что вместо логистической функции для описания вероятности наступления события используется функция стандартного нормального распределения. (4).

Множественная регрессия расширяет линейную регрессию для включения нескольких независимых переменных. Нелинейная регрессия используется, когда отношение между переменными нелинейно. Её форма зависит от конкретного вида нелинейности [7].

Классификация – это процесс присвоения объектам различных категорий или классов на основе определённых признаков и шаблонов. Главная цель — создать модель, которая будет автоматически классифицировать новые или неразмеченные данные на основе полученного опыта.

Кластеризация – это метод машинного обучения, который используется для группировки объектов в кластеры, так чтобы объекты внутри одного кластера были более похожи друг на друга, чем на объекты из других кластеров. В отличие от классификации, где объекты распределяются по заранее известным категориям, кластеризация работает без меток и пытается найти естественные группы в данных.

Метод k-ближайших соседей (k-NN) – метрический метод классификации в машинном обучении, который классифицирует объекты на основе k самых близких к нему объектов в обучающем наборе данных. Преимущества: простота реализации и понимания, эффективен на небольшом количестве признаков (особенно, если отношения между ними сложные и нелинейные), легко обучаем (самообучаем). Недостатки: медленная скорость на большом объёме данных, высокая чувствительность к несбалансированным данным и выбросам, сложность выбора метрик расстояния [8].

Алгоритм Радиус-соседей (R-соседей) – это метод обучения на основе количества соседей в пределах фиксированного радиуса от каждой точки обучения. Задаётся фиксированный радиус, такой, что точки в более разреженных окрестностях используют меньше ближайших соседей для классификации. Преимущества: подходит для данных с переменной плотностью и лучше справляется с выбросами, чем k-NN. Недостатки: выбор радиуса может быть сложным и метод может быть неэффективным в случае очень разряженных данных, когда в радиусе R нет соседей [8].

Есть модификации метрических алгоритмов с весами, основанные на расстоянии или прочих критериях, но они в работе рассматриваться не будут. Некоторые из них: метод Парзеновского Окна, метод Потенциальных Функций.

Масштабирование – это метод стандартизации независимых признаков, присутствующих в данных, в фиксированном диапазоне. Если масштабирование признаков не выполняется, то алгоритм машинного обучения, как правило, придаёт больший вес большим значениям, независимо от единиц измерения.

За расстояние между векторами (объектами) при классификации и кластеризации обычно берётся Евклидово расстояние – расстояние между двумя точками в евклидовом пространстве, вычисляемое по теореме Пифагора, или классическое измерение расстояния «по прямой» (5).

Будет применено следующе:

1. Линейная/логистическая регрессия.
2. Кластеризация методом k-ближайших соседей.
3. Классификация методом k-ближайших соседей.
4. Выбор методов оценивания результатов

Для каждой задачи существует множество метрик оценивания. Ниже приведены некоторые из них.

Для задач классификации:

TP – true positive, FP – false positive, TN – true negative, FN – false negative.

* Accuracy (точность) — доля правильных ответов модели среди всех предсказаний (6). Бесполезна в задачах с неравными классами.
* Precision (точность) — доля истинно положительных ответов среди всех положительных ответов модели (7).
* Recall (полнота) — доля истинно положительных ответов среди всех объектов положительного класса (8).
* F1-мера — гармоническое среднее между точностью и полнотой (9).

Для задач регрессии:

* Mean Squared Error (MSE) — среднее значение квадрата разности между предсказанными и правильными значениями (10).
* Root Mean Squared Error (RMSE) — квадратный корень из среднего значения квадратов разности между предсказанными и правильными значениями (12).
* Mean Absolute Error (MAE) — среднее значение абсолютной разности между предсказанными и правильными значениями (7).
* R2-коэффициент детерминации — мера, которая показывает, насколько хорошо модель подходит для данных. Более точно: единица минус доля необъяснённой дисперсии (дисперсии случайной ошибки модели, или условной по факторам дисперсии зависимой переменной) в дисперсии зависимой переменной (14).

Для задач кластеризации:

* Silhouette score (коэффициент силуэта) — мера, которая показывает, насколько точно каждый объект соответствует своему кластеру и насколько он отличается от других кластеров (15).
* Кросс-валидация — это алгоритм оценки производительности модели, состоящий в том, чтобы разбить обучающую выборку на несколько частей, обучить модель на одних и использовать остальные для проверки, и усреднить результаты.

## Практическая часть

## Программное решение

Для проведения исследования был выбран язык программирования Python. Python — гибкий язык программирования с богатым инструментарием, поэтому сегодня на нем разрабатывается множество ИИ-проектов. Многочисленные фреймворки и библиотеки Python помогают существенно уменьшить количество времени, необходимого для разработки приложений [5].

Для работы были выбраны следующие фреймворки:

1. NumPy – библиотека с открытым исходным кодом для выполнения операций линейной алгебры и численных преобразований. Как правило, такие операции необходимы для преобразования данных, которые можно представить в виде матрицы. Форматы хранения numpy де-факто являются стандартом для хранения числовых данных во многих других библиотеках [6].
2. Pandas – библиотека для обработки данных. С её помощью можно загрузить данные практически из любого источника (интеграция с основными форматами хранения данных для машинного обучения), вычислить различные функции, создать новые параметры и прочее [6].
3. Scikit-learn – библиотека программного обеспечения, которая содержит реализации практически всех возможных преобразований, и нередко её одной хватает для полной реализации модели. Содержит методы разбиения данных на тестовые и обучающие, вычисление основных метрик над наборами данных, проведение кросс-валидации. В библиотеке также есть основные алгоритмы машинного обучения: линейной регрессии (и её модификаций), опорных векторов, решающих деревьев и лесов и др. Есть и реализации основных методов кластеризации. Частью пакета является библиотека imblearn, позволяющая работать с разбалансированными выборками и генерировать новые значения [6].
4. MathPlotLib – низкоуровневый набор файлов для создания двумерных диаграмм и графиков. Позволяет составлять графики любого типа. Для сложной визуализации требует большего кода, чем иные аналоги [6].
5. Задача классификации

Была проведена классификация по признаку «ReturnStatus», матрица ошибок представлена на рис. 2.

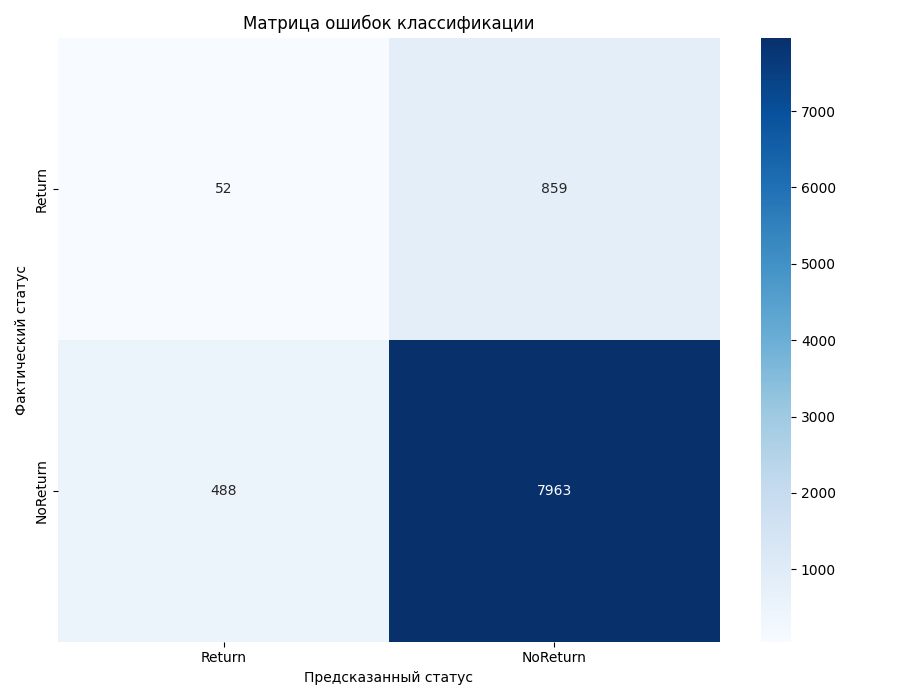


Рис. 2. Визуализация матрицы ошибок «ReturnStatus»

Интерпретация результатов:

1. в 7963-х случаях модель определила «не вернули» как «не вернули»;
2. в 52-х случаях модель определила «вернули» как «вернули»;
3. в 859-х случаях модель определила «не вернули» как «вернули»;
4. в 488-х случаях модель определила «вернули» как «не вернули»;

Точность классификации: 0,8561; F1-Score (kNN) = 0,50. Модель плохо классифицирует, так как есть дисбаланс в сторону «NoReturn», о чём нам говорит показатель F1-Score.

Для проверки программы дополнительно проведена классификация по «DiscountGroup» (гуппировке скидок), матрица ошибок представлена на рис. 3.

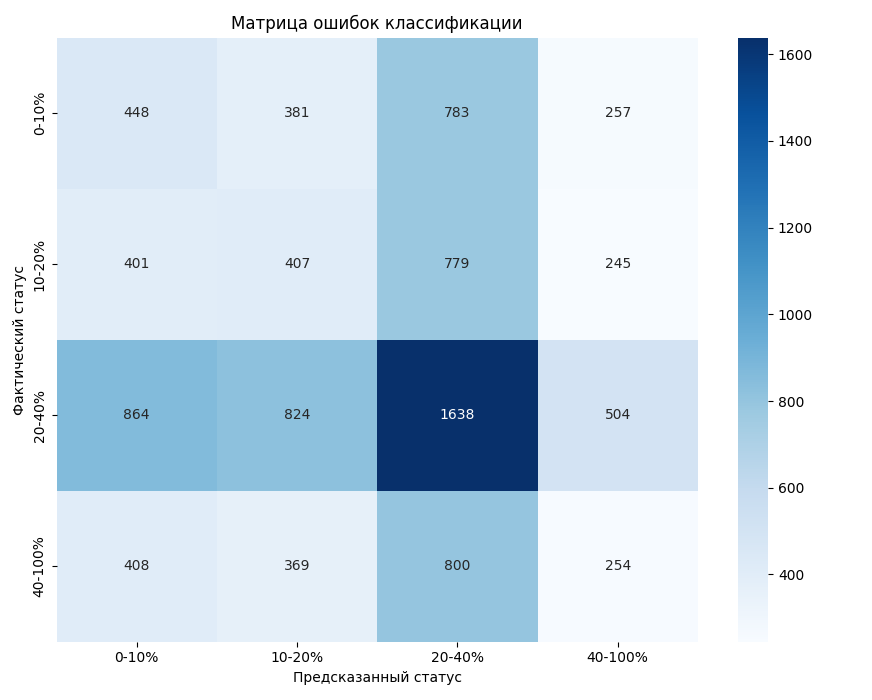


Рис. 3. Визуализация матрицы ошибок «DiscountGroup»

Точность классификации: 0.2934, F1-Score = 0.26, что является очень низким результатом.

1. Задача кластеризации

Для проверки гипотез, что «в разных странах преобладает потребность по разным товарам» и «чем больше сумма заказа, тем больше стоимость доставки», была проведена кластеризация по признакам «Description/Country» и «Sum/ShippingCost».

Графики Локтя для определения оптимального числа кластеров представлены на рис. 4-5.

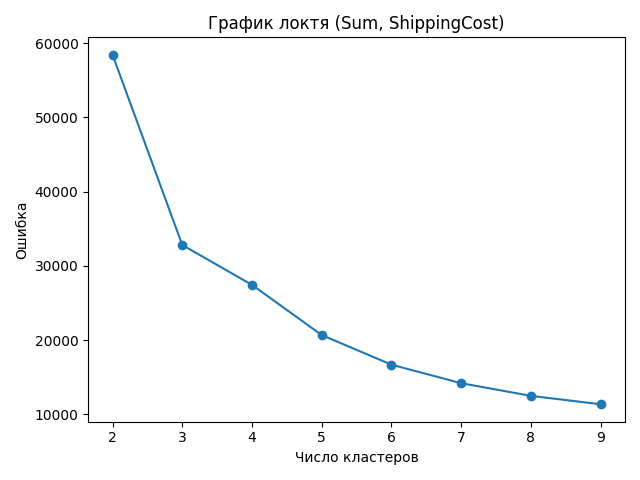


Рис. 4. График локтя для «Description/Country»

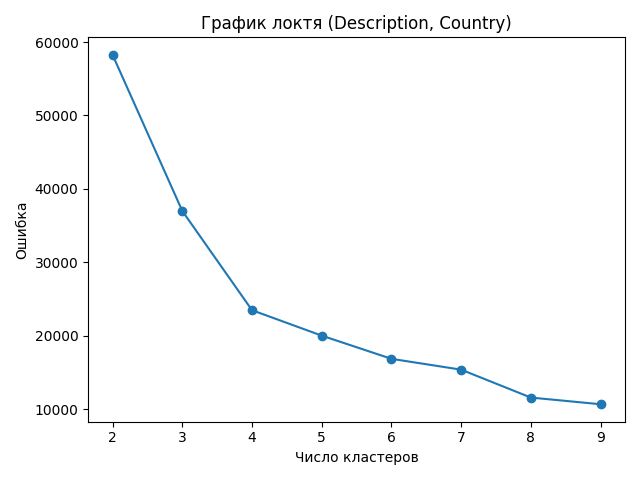


Рис. 5. График локтя для «Description/Country»

Ошибка перестаёт быстро снижаться при достижении 3-х и 4-х кластеров соответственно, значит на столько и будут кластеризироваться данные.

Визуализация кластеров представлена на рис. 6-7.

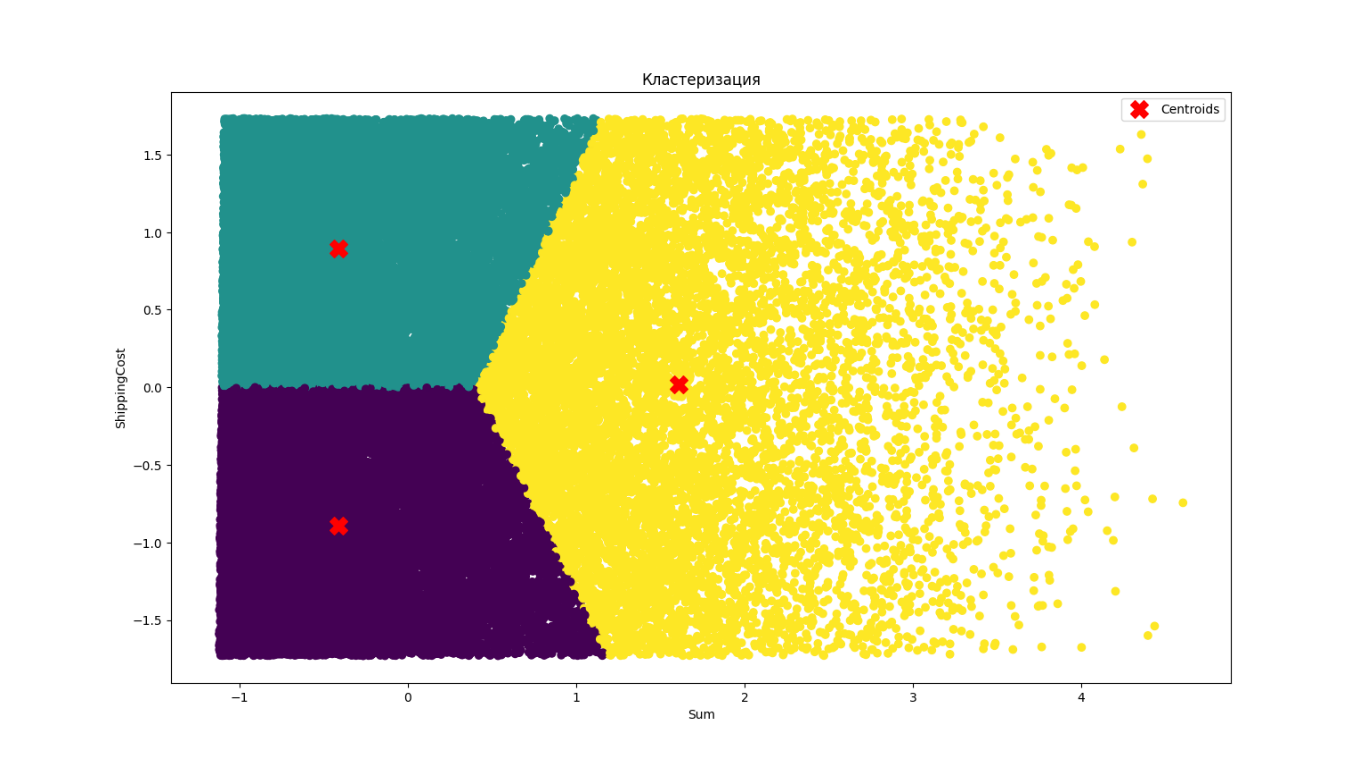


Рис. 6. Результат кластеризации по «Sum/ShippingCost»

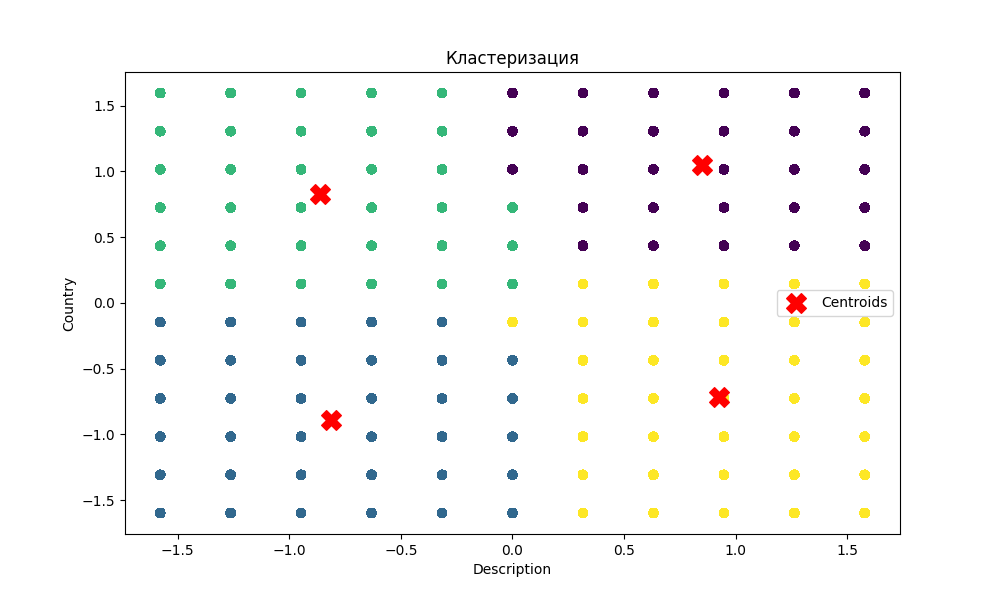


Рис. 7. Результат кластеризации по «Description/Country»

1. Задача регрессии

Регрессионный анализ можно провести только в том случае, когда нет мультиколлинеарности (сильной корреляции независимых переменных) и присутствует связь между переменными. Корреляция признаков в исходных данных отсутствует и провести регрессионный анализ невозможно (рис. 8).

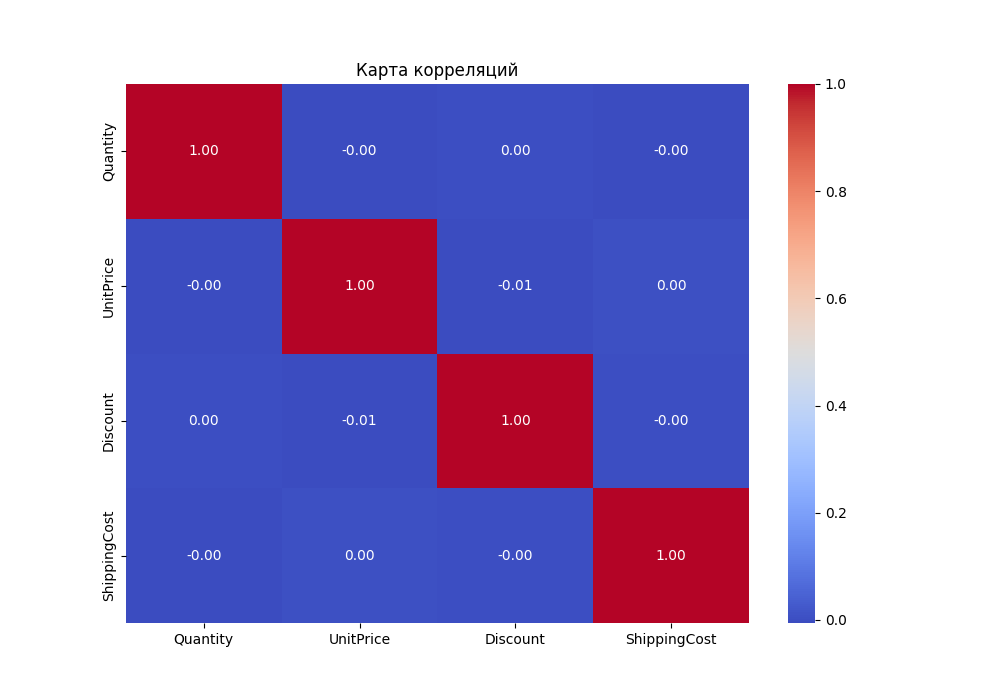


Рис. 8. Диаграмма корреляции признаков исходных данных

1. Интерпретация и использование результатов

Исходя из проведённого исследования, можно однозначно сказать, что подавляющее большинство данных было синтезировано и не представляет ценности. Чтобы дополнительно это проверить, можно графически изобразить распределение признаков (рис. 9-10).

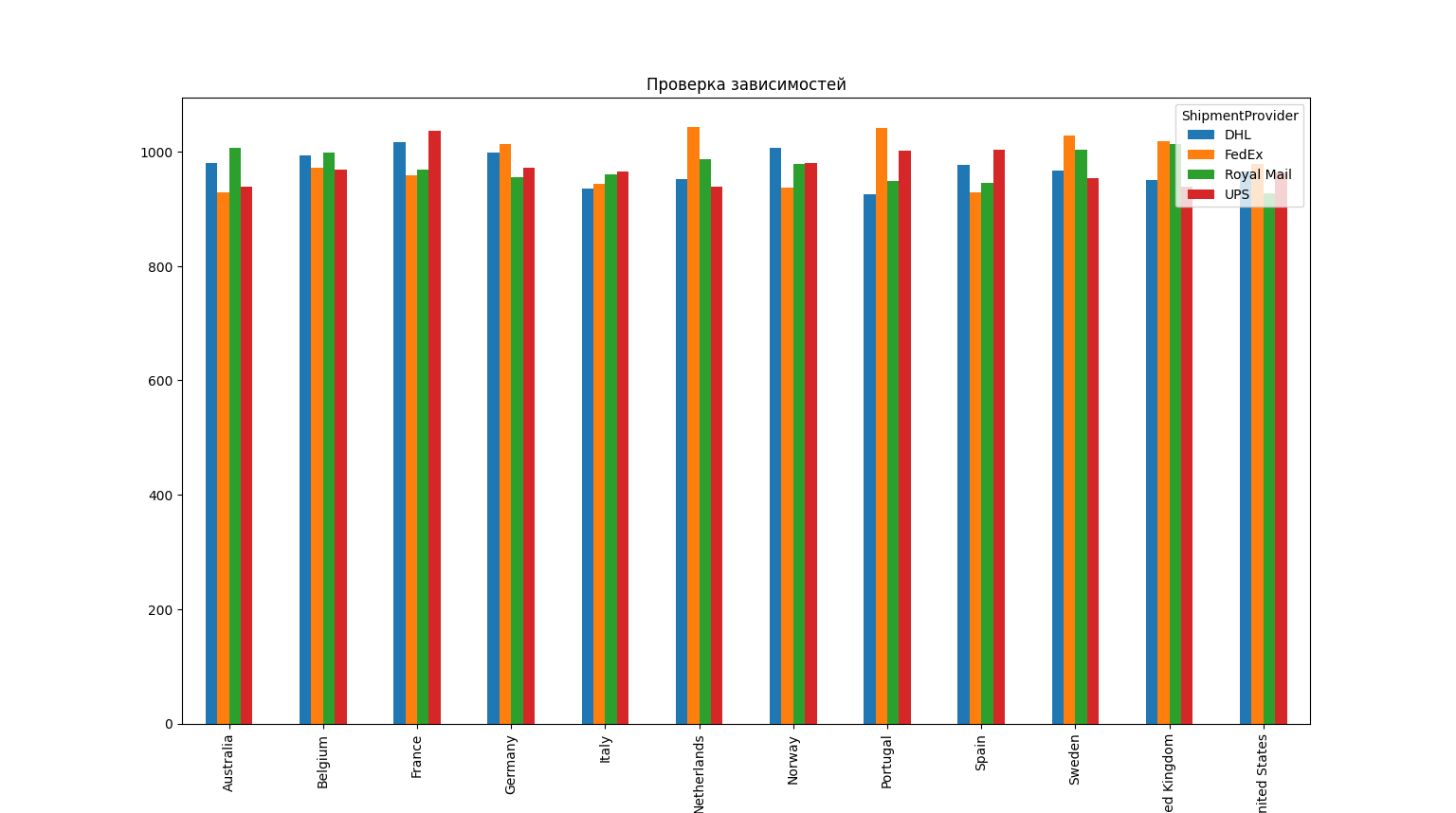


Рис. 9. Диаграмма распределения компаний доставки из разных стран

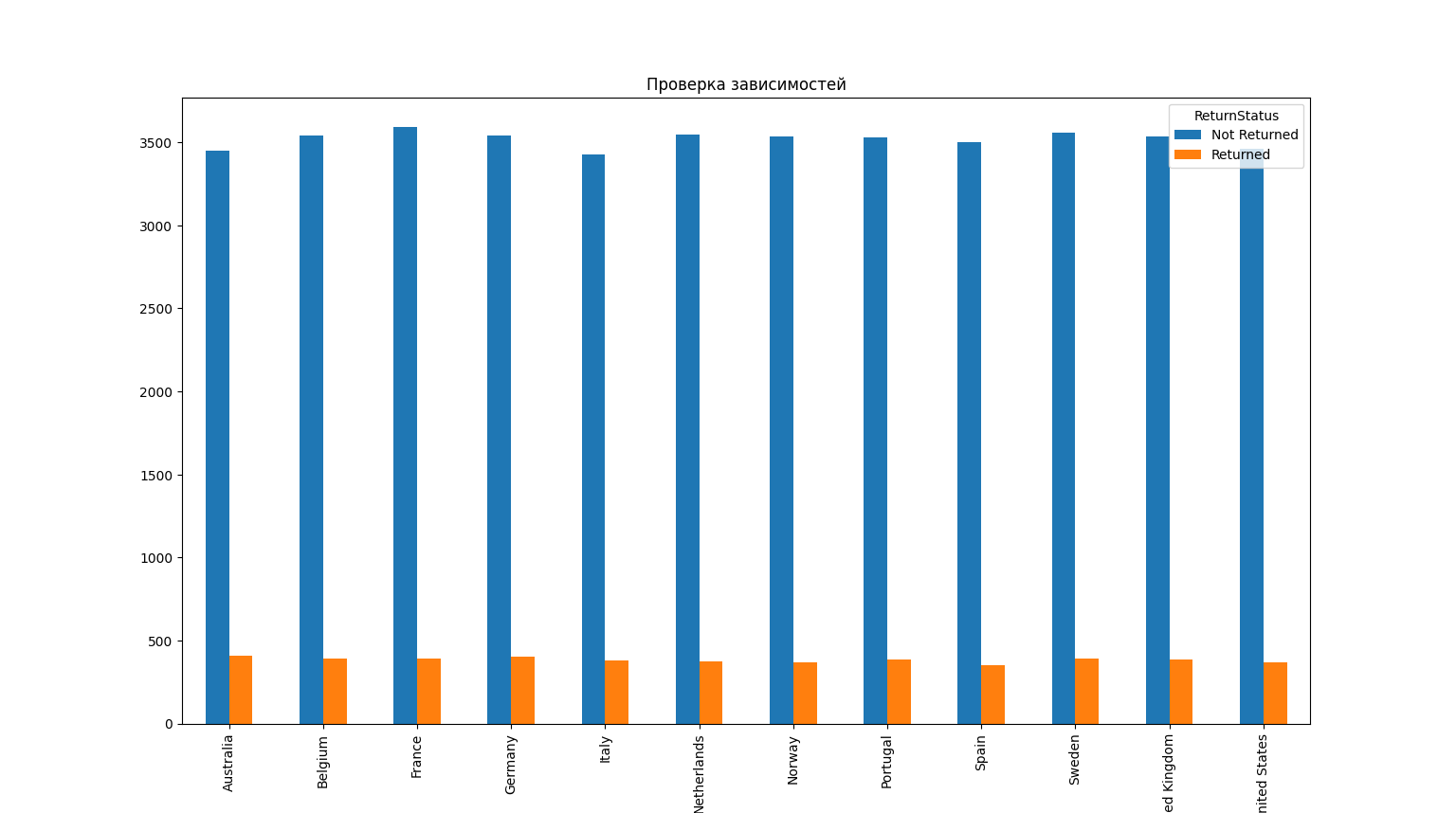


Рис. 10. Диаграмма распределения заказов из разных стран и числа возвратов

Для проверки работоспособности программы было решено взять другой набор, не содержащий столько синтезированных данных. Они брались с ресурса Kaggel, в этот раз о продаже недвижимости в России в период 2018-2021 гг. [4].

Данные содержат признаки:

* price – цена в рублях;
* date – дата публикации объявления;
* time – время публикации объявления;
* geo\_lat – широта;
* geo\_lon – долгота;
* region – номер региона России;
* building\_type – тип фасада, (0 – другой, 1 – панельный, 2 – монолитный, 3 — кирпичный, 4 — блочный, 5 — деревянный;
* level – этаж квартиры;
* levels – количество этажей;
* rooms – количество жилых комнат, если значение равно «-1», то это означает «квартира-студия»;
* area – общая площадь квартиры;
* kitchen\_area – кухонная зона;
* object\_type – тип квартиры (1 – вторичный рынок, 11 – новострой).

Был создан новый признак: region\_name, код региона преобразован в его имя.

Гипотезы:

1. чем выше этаж недвижимости, тем дешевле стоимость;
2. цена недвижимости зависит от региона;
3. чем выше цена недвижимости, тем больше её площадь;
4. чем больше площадь недвижимости, тем больше там комнат.

Результаты работы программы на новых данных представлен на рис. 11-20.

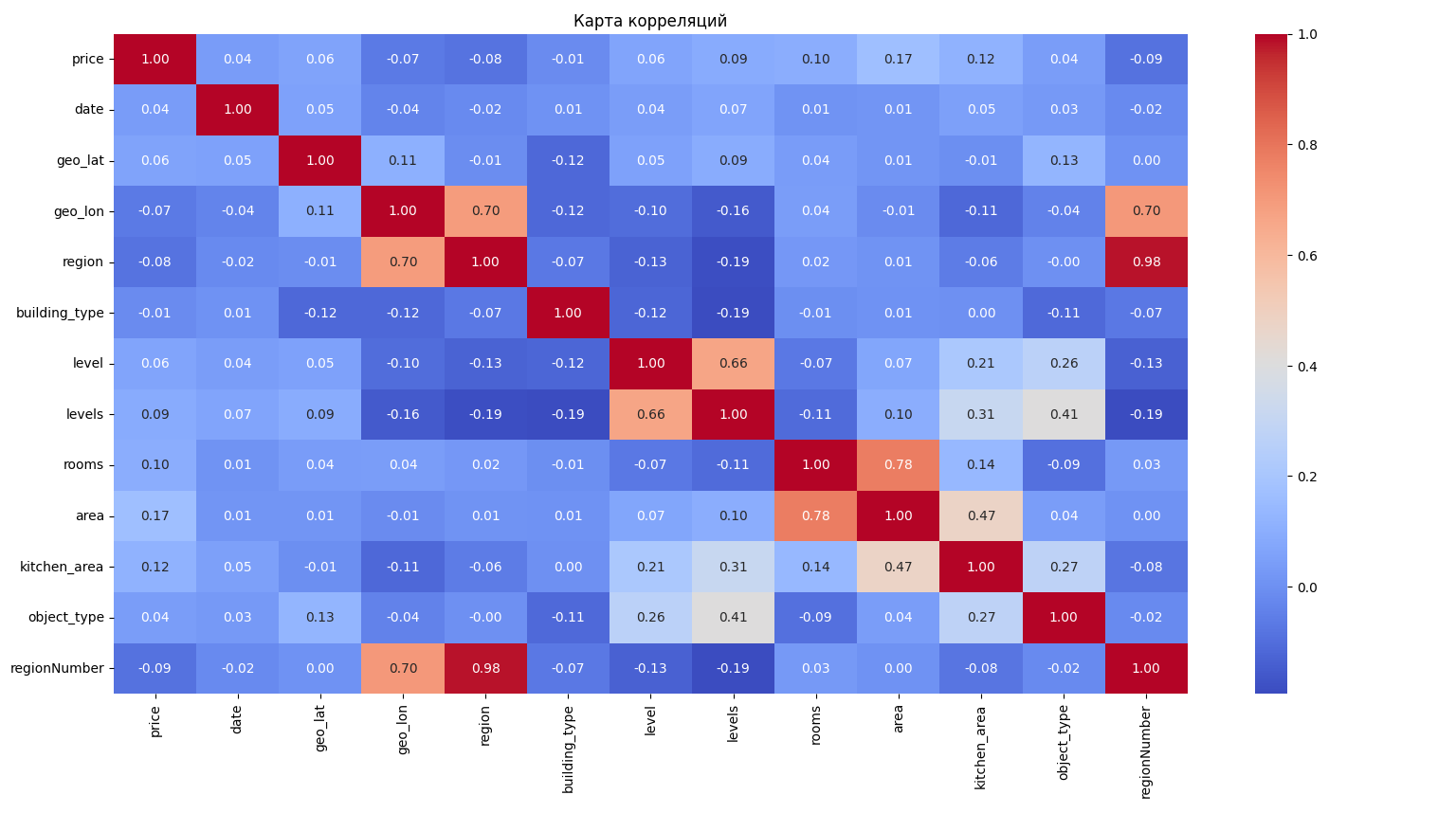


Рис. 11. Диаграмма корреляции признаков новых данных

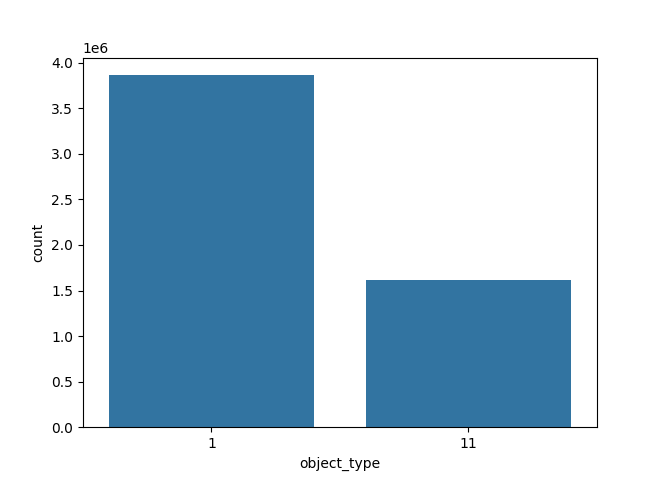


Рис. 12. Диаграмма распределения новостроек и вторичного рынка

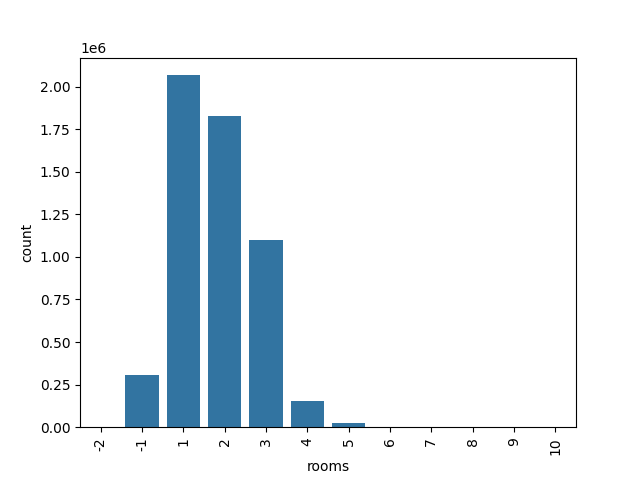


Рис. 13. Диаграмма распределения числа комнат в недвижимости

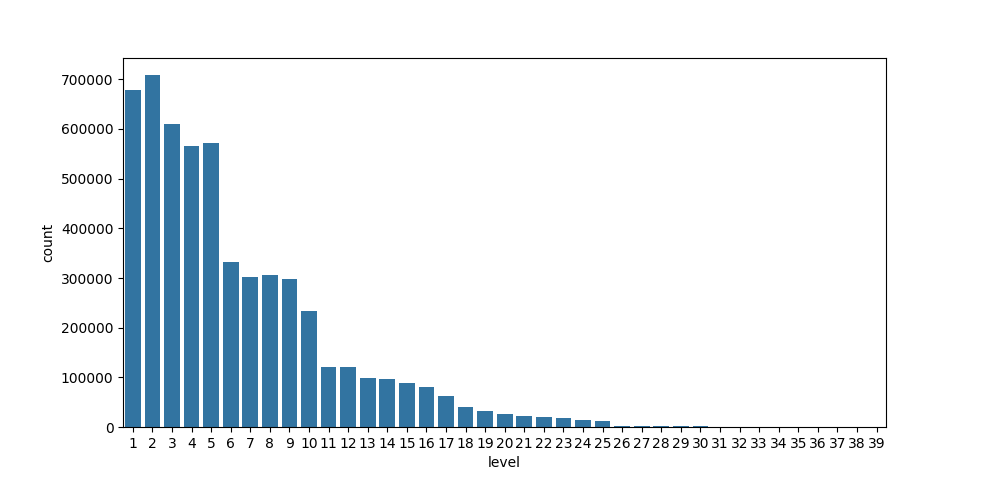


Рис. 14. Диаграмма распределения жилья на определённом этаже

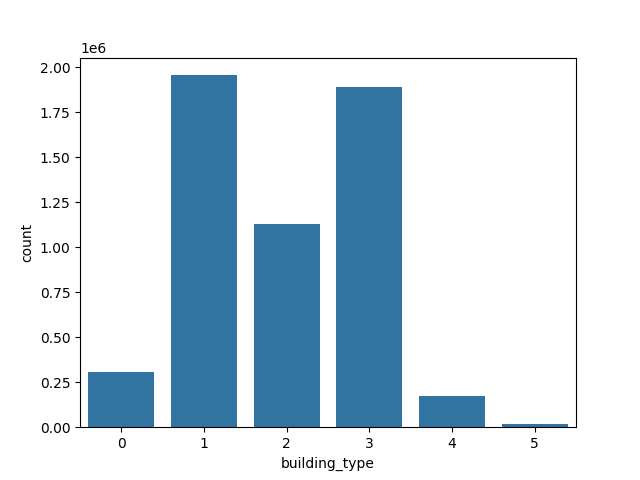


Рис. 15. Диаграмма распределения зданий определённого фасада

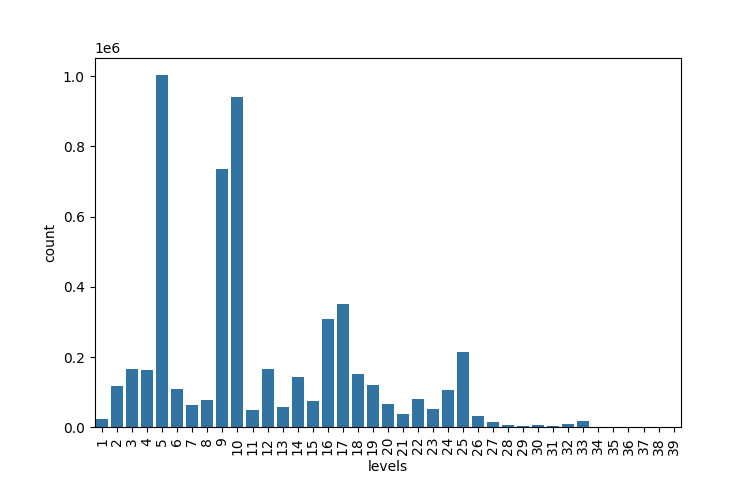


Рис. 16. Диаграмма распределения зданий определённого количества этажей

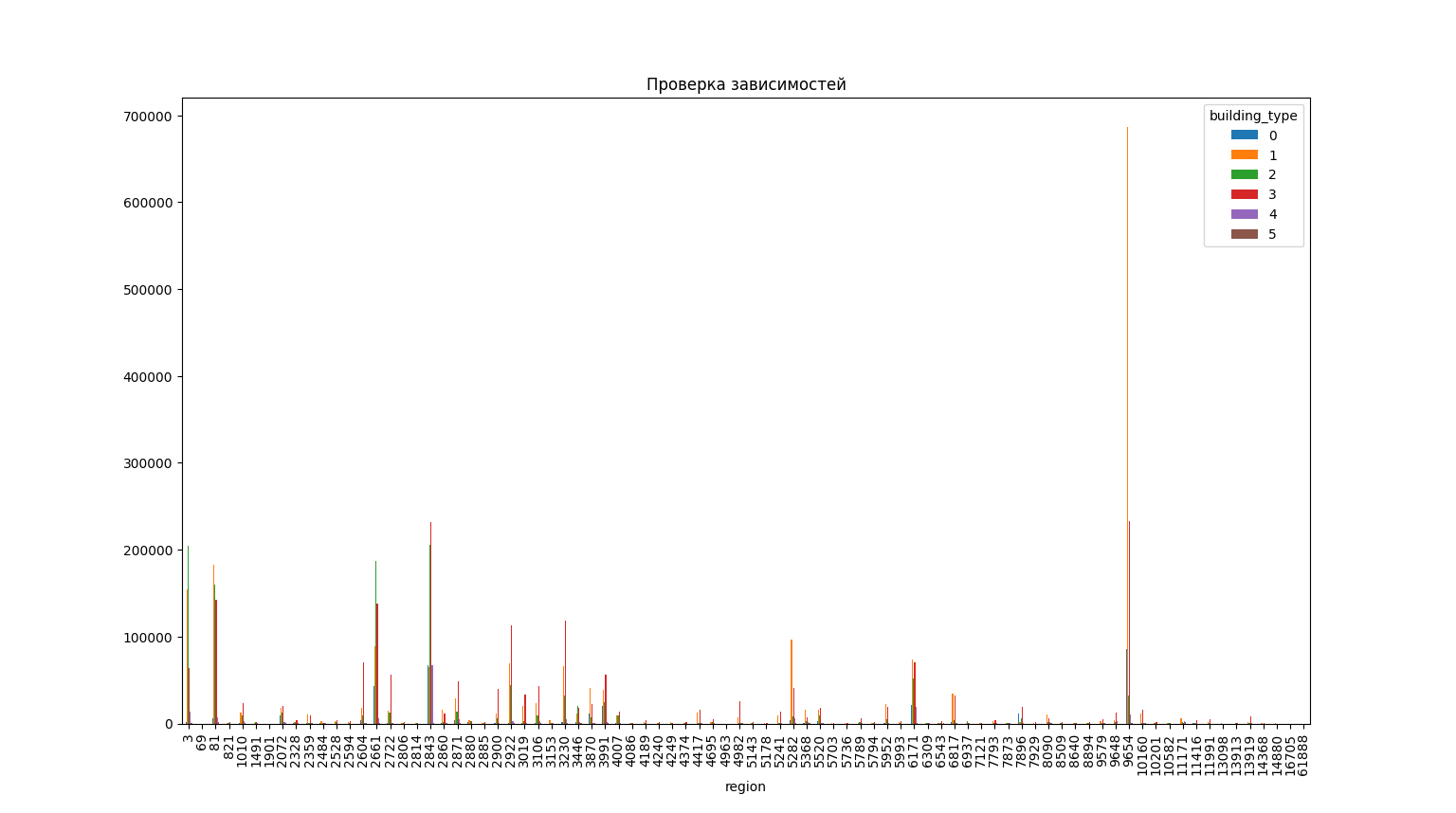


Рис. 17. Диаграмма зависимости зданий определённого фасада от региона

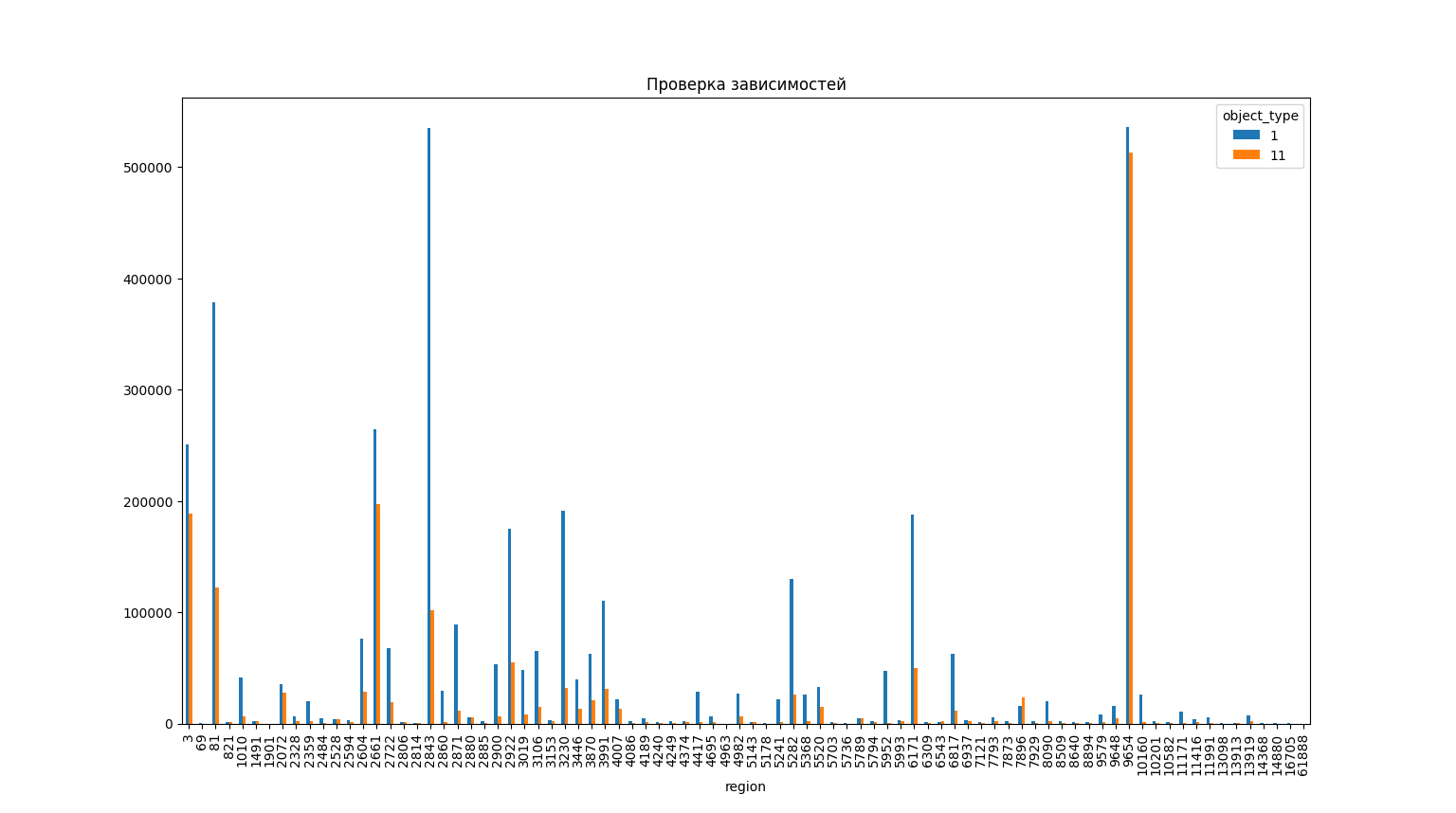


Рис. 18. Диаграмма зависимости типа здания от региона

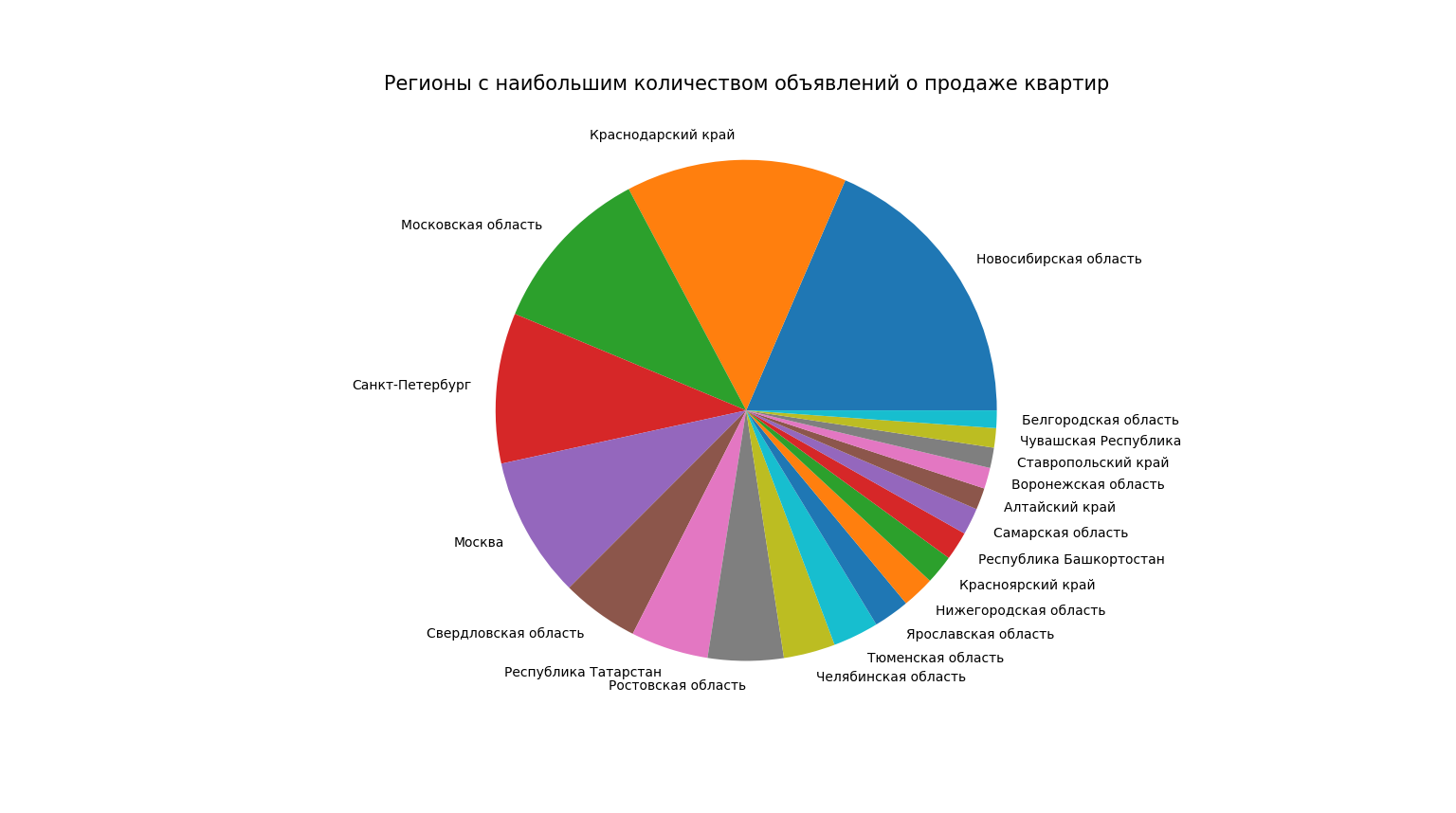


Рис. 19. Диаграмма количества объявлений о продаже квартир в регионах

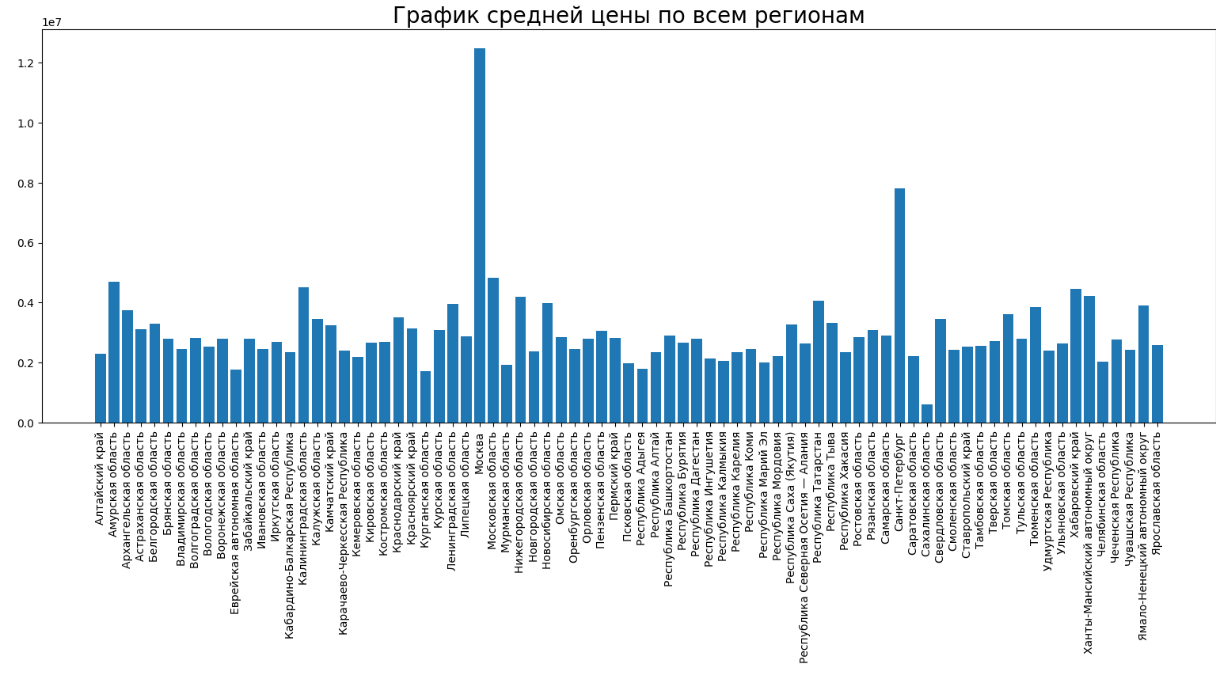


Рис. 20. График средней цены по всем регионам

Видно, что данные плохо сбалансированы, есть выбросы и преобладание одних значений над другими, которые при исследовании учитываться не будут.

Корреляция выглядит убедительно, мультиколлинеарности не наблюдается, и здесь сразу же опровергается гипотеза «цена недвижимости зависит от региона», так как зависимость незначительна.

Если брать отдельный регион (Новосибирская область, например) и провести регрессионный анализ, то значимой линейной взаимосвязи для числовых признаков не наблюдается (рис. 21-22).

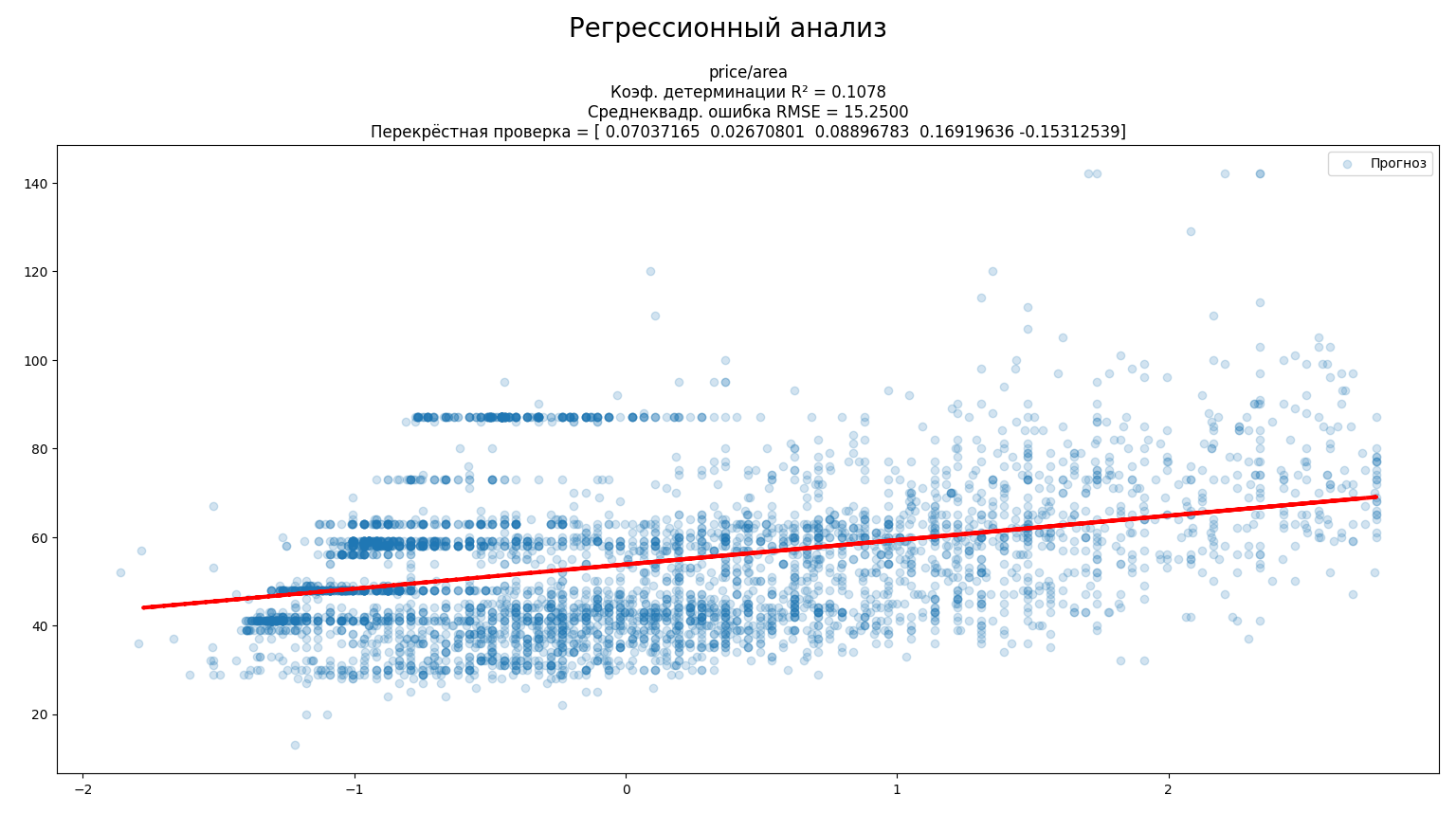


Рис. 21. График регрессии цены от площади

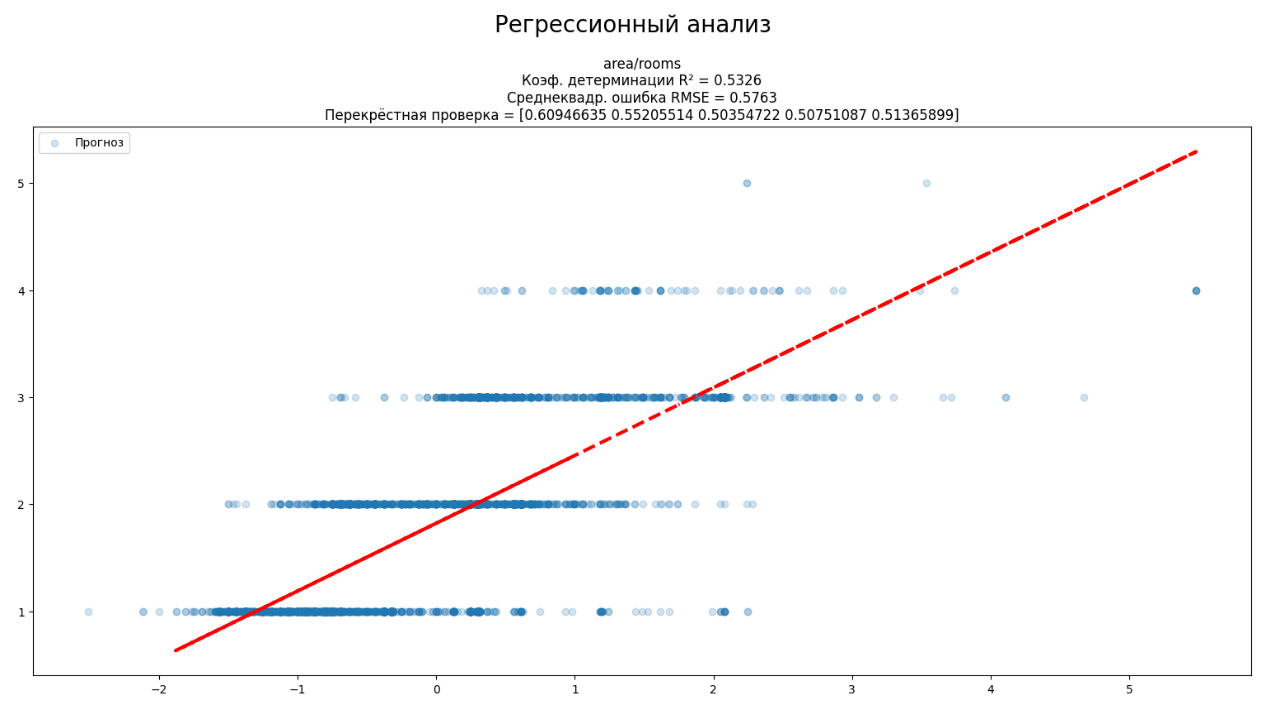


Рис. 22. График регрессии площади и количества комнат

Так как записей в данных больше 5 млн. штук, было решено взять выборку каждой сороковой записи из генеральной совокупности. При большем количестве записей намного увеличивается длительность составления кластеров.

Диаграмма кластеризации по «geo\_lat/price» показывает, что цена зависит от широты, а широта, пусть и недостаточно коррелирует с регионом, сужает область поиска более дорогой недвижимости (рис. 23). Оценка силуэта = 0.74.

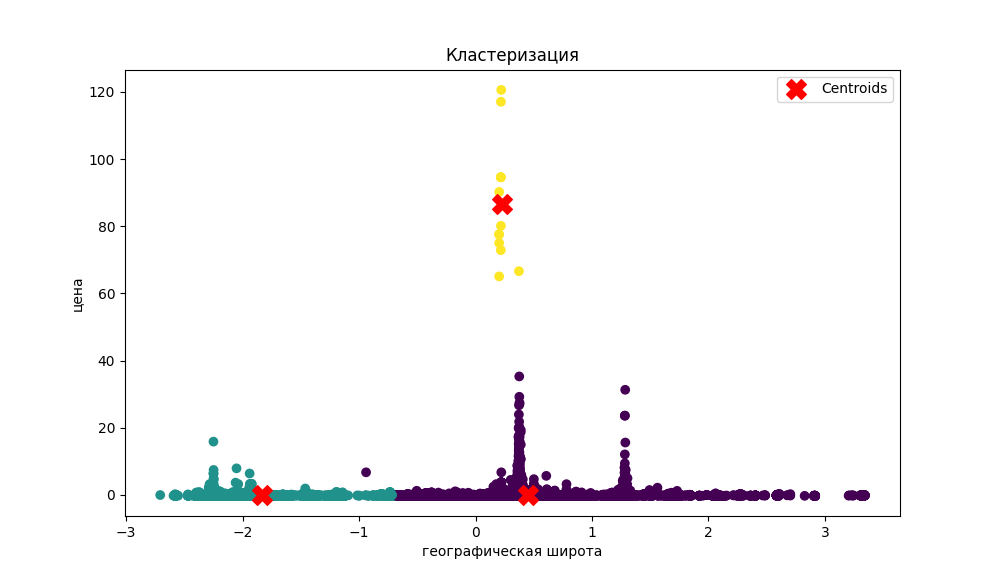


Рис. 23. Результат кластеризации по «geo\_lat/price»

Была проведена классификация по признаку «rooms», графически представлено на рис. 24. Точность классификации: 0.80, F1-Score = 0.64, перекрестная проверка точности = 0.77 (рис. 24).

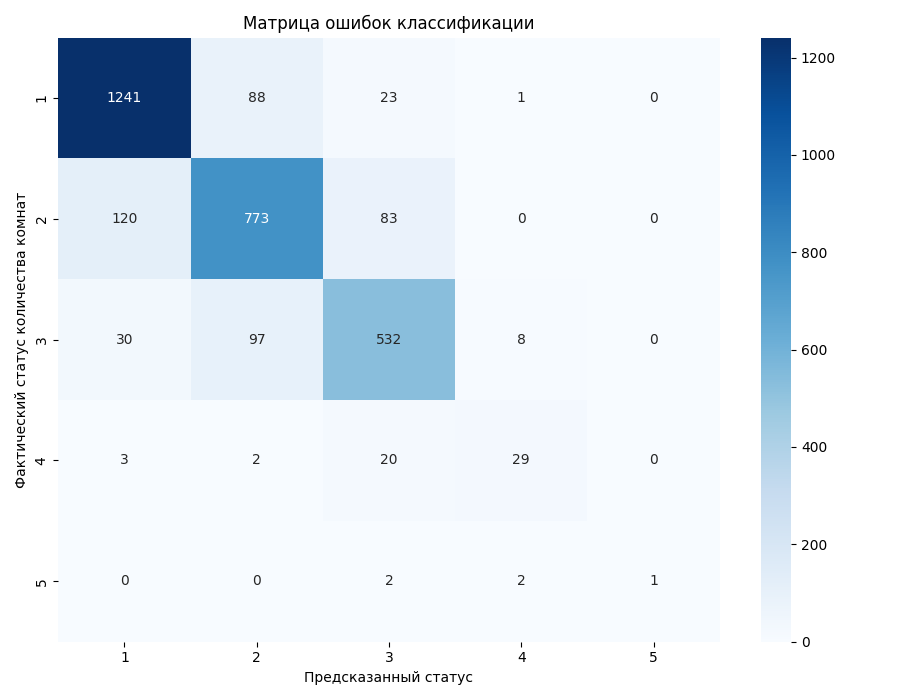


Рис. 24. График матрицы ошибок предсказания количества комнат

Результаты исследования показывают, что:

* гипотеза «цена недвижимости зависит от региона» оказалась ложной;
* гипотеза «чем выше цена недвижимости, тем больше её площадь» оказалась ложной;
* цена недвижимости зависит от географической широты;
* достаточно точное предсказание сколько комнат в недвижимости;
* гипотеза «чем выше этаж недвижимости, тем дешевле стоимость» оказалась ложной.

Результаты исследования можно использовать застройщикам для повышения строительства новостроек в тех регионах, где преобладает вторичный рынок, риелторским агентствам для оценки цены недвижимости по региону, банковским структурам для выдачи специальных предложений кредитований в регионах, где новостройки покупать не хотят (когда процесс покупки делается «в два шага» и по разумной ставке, люди охотнее «расстаются» с деньгами).

## Заключение

В ходе курсовой работы было сделано следующее:

1. Написана программа на языке Python.
2. Проведён регрессионный анализ.
3. Проведена кластеризация.
4. Проведена классификация.
5. Графически представлены результаты работы программы.

Было обнаружено синтезирование исходном наборе данных, для проверки работоспособности программы был взят другой набор.

Для оценки точности и достоверности машинного обучения использовались: коэффициент детерминации R2, RMSE и перекрёстная проверка для регрессии; оценка силуэта для кластеризации; точность, F1-Score и перекрестная проверка точности для классификации.

## Список литературы

1. Методика и организация самостоятельной работы студентов / Е.В. Ершов, д-р техн. наук, проф.; Л.Н. Виноградова, канд. техн. наук; В.В. Селивановских, канд. техн. наук, доцент; О.Л. Селяничев, канд. техн. наук, доцент; О.В. Юдина, канд. техн. наук; И.А. Варфоломеев, канд. техн. наук; Е.В. Майтама; О.С. Сальникова; Н.Е. Сивков; вед. ред. Г.В. Иванова, вед. тех. редактор М.Н. Авдюхова Е.В. – ФГБОУ ВПО «Череповецкий государственный университет» 162600 г. Череповец, пр. Луначарского, 5., 2015. – 243 c.
2. Методы оптимизации в примерах и задачах: Учеб. пособие / А.В. Пантелеев, Т.А. Летова. – 2-е изд., исправл. — М.: Высш. шк., 2005. — 544 с.: ил.
3. Online Sales Dataset. – Электронный ресурс. – URL: https://www.kaggle.com/datasets/yusufdelikkaya/online-sales-dataset (19.12.24).
4. Russia Real Estate 2018-2021. – Электронный ресурс. – URL: https://www.kaggle.com/datasets/mrdaniilak/russia-real-estate-20182021/ (21.12.24).
5. Метрики в задачах машинного обучения / Хабр. – Электронный ресурс. – URL: <https://habr.com/ru/companies/ods/articles/328372/> (19.12.24).
6. ПО для машинного обучения на Python / Хабр. – Электронный ресурс. – URL: <https://habr.com/ru/companies/jetinfosystems/articles/415997/> (19.12.24).
7. Машинное обучение. Регрессия и классификация. – Электронный ресурс. – URL: https://rutube.ru/video/8b1ef8e32c4eef55dc7778e30f213d93/ (21.12.24).
8. Машинное обучение. Метрические методы. – Электронный ресурс. – URL: https://rutube.ru/video/83e9e0075496b3571ea6b6fb65ab52c4/ (21.12.24).
9. Feature Engineering: Scaling, Normalization, and Standardization - GeeksforGeeks. – Электронный ресурс. – URL: https://www.geeksforgeeks.org/ml-feature-scaling-part-2/ (21.12.24).
10. Урок 5. Оценка результатов машинного обучения. – Электронный ресурс. – URL: https://4brain.ru/aibasics/results.php (21.12.24).

## Приложение 1

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий

наименование института (факультета)

Математическое и программное обеспечение ЭВМ

наименование кафедры

Математические методы решения задач искусственного интеллекта

наименование дисциплины в соответствии с учебным планом

УТВЕРЖДАЮ

Зав. кафедрой МПО ЭВМ,

д.т.н., профессор \_\_\_\_\_\_\_\_\_\_ Ершов Е.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

МАТЕМАТИЧЕСКИЕ МЕТОДЫ В МАШИННОМ ОБУЧЕНИИ

Техническое задание на курсовую работу

Листов 6

|  |  |
| --- | --- |
| Руководитель | Юдина О.В. |
|  | Ф.И.О преподавателя |
| Исполнитель |  |
| студент | 1ПИб-02-1оп-22 |
|  | группа |
|  | Микуцких Г.А. |
|  | Фамилия, имя, отчество |

2024 год

Введение

Чтобы глубже понять математические основы, лежащие в основе алгоритмов искусственного интеллекта, необходимо научиться применять теоретические знания к реальным задачам. Целью курсовой работы является написание программы для анализа данных о транзакциях онлайн-продаж.

1. Основания для разработки

Основанием для разработки является задание на курсовую работу по дисциплине «Математические методы решения задач искусственного интеллекта», выданное на кафедре МПО ЭВМ ИИТ ЧГУ.

Дата утверждения: 5 ноября 2024 года.

Наименование темы разработки: «Математические методы в машинном обучении».

1. Назначение разработки

Освоение на практике материала, полученного в ходе изучения дисциплины «Математические методы решения задач искусственного интеллекта».

1. Требования к программе
2. Требования к функциональным характеристикам

Программа должна выполнять следующие основные функции:

1. загрузка данных из файла;
2. предобработка данных (исключение выбросов, заполнение пропусков и замена некорректных данных);
3. проведение анализа данных и вычисления;
4. оценка точности и адекватности результатов;
5. визуализация результатов (графики, диаграммы, оценки и пр.).

Дополнительное требование: использовать язык программирования Python.

1. Требования к надежности

Программа должна предварительно обрабатывать входные данные из файла, не менять его содержимое, производить вычисления без ошибок и остановок.

1. Условия эксплуатации

Предъявляются с условиями эксплуатации ПК пользователя.

1. Требования к составу и параметрам технических средств

ЭВМ должна соответствовать следующим параметрам:

* исходные данные представляются в виде файла, созданного в программе «Excel» (в частности, файлы расширений «.xlcx» и «.csv»), результаты представляются в виде текста и графиков и диаграмм;
* минимальные системные требования:
  + 64-разрядный (x64) процессор с тактовой частотой 1 ГГц или выше;
  + 1 ГБ (для 32-разрядного процессора) или 2 ГБ (для 64-разрядного процессора) ОЗУ, 16 ГБ (для 32-разрядной системы) или 20 ГБ (для 64-разрядной системы) свободного места на жестком диске;
  + графическое устройство DirectX 9 с драйвером WDDM 1.0 или более поздней версии;
  + поддержка клавиатуры, компьютерной мыши, монитора разрешением 1920×1080 пикселей.

1. Требования к маркировке и упаковке

Не предъявляются.

1. Требования к информационной и программной совместимости

Программа должна корректно работать на ОС Windows 10 и выше в приложении «Visual Studio Code» со всеми необходимыми расширениями.

1. Требования к транспортированию и хранению

Всем пользователям запрещается осуществлять действия под видом оригинального ПО, нарушающие статью 273 «Создание, использование и распространение вредоносных компьютерных программ».

1. Специальные требования

Для эффективной работы с программой необходимо наличие опыта использования интерпретируемых программ и всех необходимых библиотек.

1. Требование к программной документации
2. Содержание расчётно-пояснительной записки

Программная документация должна содержать расчётно-пояснительную записку, содержание которой:

Титульный лист

Аннотация

Оглавление

Введение

Теоретическая часть

Описание данных

Формирование своих признаков

Возможные задачи машинного обучения и планируемые результаты

Выбор методов оценивания результатов

Практическая часть

Программное решение

Задача классификации

Задача кластеризации

Задача регрессии

Заключение

Список литературы

Приложения

1. Технико-экономические показатели

Требования не предъявляются.

1. Требования к оформлению

Элементы курсовой работы оформлены в соответствии с табл. П1.1.

Таблица П1.1

Требования к оформлению

|  |  |
| --- | --- |
| Элемент | Требования |
| Документ | 1. Печать на отдельных листах формата А4 (210х297 мм); оборотная сторона не заполняется; листы нумеруются. Печать возможна ч/б.  2. Файлы предъявляются на компакт-диске: РПЗ с ТЗ; программный код.  3. Листы и диск в конверте вложены в пластиковую папку скоросшивателя. |
| Страницы | 1. Ориентация – книжная; отдельные страницы, при необходимости, альбомная.  2. Поля: верхнее, нижнее – по 2 см, левое – 3 см, правое – 1 см. |
| Абзацы | Межстрочный интервал – 1.5, перед и после абзаца – 0. |
| Шрифты | Кегль – 14. В таблицах шрифт 12. Шрифт программного кода – 8 (возможно в 2 колонки). |
| Рисунки | Подписывается под ним по центру: Рис. Х. Название  В приложениях: Рис. ПX.X. Название |
| Таблицы | 1. Подписывается: над таблицей, выравнивание по правому: «Таблица Х».  2. В следующей строке по центру Название  3. Надписи в «шапке» (имена столбцов, полей) – по центру.  4. В теле таблицы (записи) текстовые значения – выровнены по левому краю, числа, даты – по правому. |

1. Стадии и этапы разработки

Курсовая работа и программа будут разрабатываться в течение следующих этапов (табл. П1.2):

Таблица П1.2

Стадии и этапы разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование  этапа разработки | Сроки разработки | Результат выполнения | Отметка о выполнении |
| Оформление технического задания | 08.12.24 | Составлено техническое задание |  |
| Кодирование | 09.12.24–  22.12.24 | Программа написана |  |
| Тестирование | 20.12.24 – 22.12.24 | Отладка |  |
| Оформление расчётно-пояснительной записки | 08.12.24 – 22.12.24 | Оформлена расчётно-пояснительная записка |  |

1. Порядок контроля и приемки

Курсовая работа будет сдаваться в течение этапов в табл. П1.3.

Таблица П1.3

Порядок контроля и приёмки

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование контрольного этапа выполнения курсовой работы | Сроки  контроля | Результат выполнения | Отметка о приемке результата контрольного этапа |
| Сдача технического задания |  | Техническое задание принято |  |
| Исправление ошибок |  | Тестирование прошло успешно |  |
| Демонстрация программного решения |  | Программа принята |  |
| Сдача расчётно-пояснительной записки |  | Расчётно-пояснительная записка принята |  |
| Защита курсовой работы |  | Курсовая работа защищена |  |

# Приложение 2

Текст программы.

Файл «kurs\_Real Estate.py»

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import r2\_score, accuracy\_score, f1\_score, confusion\_matrix, mean\_squared\_error, silhouette\_score

from sklearn.cluster import KMeans

from sklearn.neighbors import KNeighborsClassifier

scaler = StandardScaler()

pd.options.mode.chained\_assignment = None

# НАСТРОЙКА ОТОБРАЖЕНИЯ В КОНСОЛИ

pd.set\_option('display.max\_columns', None) # Показывать все поля

pd.set\_option('display.expand\_frame\_repr', False) # Не переносить строки

pd.set\_option('display.max\_colwidth', None) # Максимальная ширина столбцов

pd.set\_option('display.float\_format', lambda x: '%.3f' % x)

# ЗАГРУЗКА ДАННЫХ

excel\_data = pd.read\_csv('all\_v2.csv', delimiter=',')

excel\_data = excel\_data.iloc[::30] # сокращение числа записей

# УДАЛЕНИЕ ПУСТЫХ КАТЕГОРИАЛЬНЫХ, ЗАПОЛНЕНИЕ ЧИСЛОВЫХ

excel\_data['rooms'] = excel\_data["rooms"].apply(lambda x: 1 if x <= 0 else x)

excel\_data["price"] = excel\_data["price"].abs()

min\_price = 600000

max\_price = 6000000

excel\_data = excel\_data[(excel\_data["price"] <= max\_price) & (excel\_data["price"] >= min\_price)]#

excel\_data["object\_type"] = excel\_data["object\_type"].apply(lambda x: 2 if x == 11 else x)

excel\_data['date'] = pd.to\_datetime(excel\_data['date'], format='%Y-%m-%d').dt.year

excel\_data = excel\_data.drop(['time'], axis = 1)

min\_area = 10

max\_area = 150

min\_kitchen\_area = 6

max\_kitchen\_area = 60

excel\_data = excel\_data[(excel\_data["area"] <= max\_area) & (excel\_data["area"] >= min\_area)]

excel\_data = excel\_data[(excel\_data["kitchen\_area"] <= max\_kitchen\_area) & (excel\_data["kitchen\_area"] >= min\_kitchen\_area)]

region\_name = {

'2661': 'Санкт-Петербург',

'3446': 'Ленинградская область',

'3': 'Москва',

'81': 'Московская область',

'2843': 'Краснодарский край',

'2871': 'Нижегородская область',

'3230': 'Ростовская область',

'3106': 'Самарская область',

'2922': 'Республика Татарстан',

'2900': 'Ставропольский край',

'2722': 'Республика Башкортостан',

'6171': 'Свердловская область',

'4417': 'Республика Коми',

'5282': 'Челябинская область',

'5368': 'Иркутская область',

'5520': 'Пермский край',

'6817': 'Алтайский край',

'9579': 'Республика Бурятия',

'2604': 'Ярославская область',

'1010': 'Удмуртская Республика',

'7793': 'Псковская область',

'13919': 'Республика Северная Осетия — Алания',

'2860': 'Кемеровская область',

'3019': 'Чувашская Республика',

'4982': 'Республика Марий Эл',

'9648': 'Кабардино-Балкарская Республика',

'5241': 'Республика Мордовия',

'3870': 'Красноярский край',

'3991': 'Тюменская область',

'2359': 'Республика Хакасия',

'9654': 'Новосибирская область',

'2072': 'Воронежская область',

'8090': 'Республика Карелия',

'4007': 'Республика Дагестан',

'11171': 'Республика Саха (Якутия)',

'10160': 'Забайкальский край',

'7873, 6937': 'Республика Крым',

'2594': 'Кировская область',

'8509': 'Республика Калмыкия',

'11416': 'Республика Адыгея',

'11991': 'Карачаево-Черкесская Республика',

'5178': 'Республика Тыва',

'13913': 'Республика Ингушетия',

'6309': 'Республика Алтай',

'5952': 'Белгородская область',

'6543': 'Архангельская область',

'2880': 'Тверская область',

'5993': 'Пензенская область',

'2484': 'Ханты-Мансийский автономный округ',

'4240': 'Липецкая область',

'5789': 'Владимирская область',

'14880': 'Ямало-Ненецкий автономный округ',

'1491': 'Рязанская область',

'2885': 'Чеченская Республика',

'5794': 'Смоленская область',

'2528': 'Саратовская область',

'4374': 'Вологодская область',

'4695': 'Волгоградская область',

'2328': 'Калужская область',

'5143': 'Тульская область',

'2806': 'Тамбовская область',

'14368': 'Мурманская область',

'5736': 'Новгородская область',

'7121': 'Курская область',

'4086': 'Хабаровский край',

'821': 'Брянская область',

'10582': 'Астраханская область',

'7896': 'Калининградская область',

'8640': 'Омская область',

'5703': 'Курганская область',

'10201': 'Томская область',

'4249': 'Ульяновская область',

'3153': 'Оренбургская область',

'4189': 'Костромская область',

'2814': 'Орловская область',

'13098': 'Камчатский край',

'8894': 'Ивановская область',

'7929': 'Амурская область',

'16705': 'Магаданская область',

'69': 'Еврейская автономная область',

'4963': 'Приморский край',

'1901': 'Сахалинская область',

'61888': 'Ненецкий автономный округ'

}

excel\_data['region\_name'] = excel\_data["region"].astype(str).map(region\_name)

excel\_data.dropna(inplace=True)

useful\_data = excel\_data.drop\_duplicates()

# ОПРЕДЕЛЕНИЕ УНИКАЛЬНЫХ ЗНАЧЕНИЙ, проверка данных

print(useful\_data.describe())

print(useful\_data.head())

print(useful\_data.info())

# print(useful\_data['rooms'].unique())

print(useful\_data.isna().sum())

# КОРРЕЛЯЦИЯ

plt.figure(figsize=(12, 7))

plt.title('Карта корреляций')

sns.heatmap(useful\_data.select\_dtypes(include=[np.number]).corr(), annot=True, fmt='.2f', cmap='coolwarm')

plt.show()

# РЕГИОНЫ С НАИБОЛЬШИМ КОЛИЧЕСТВОМ ОБЪЯВЛЕНИЙ О ПРОДАЖЕ КВАРТИР

useful\_data['region\_name'].value\_counts().head(20).plot(kind='pie',stacked=True,figsize=(20,10))

plt.title("Регионы с наибольшим количеством объявлений о продаже квартир",fontsize=15)

plt.ylabel(' ')

plt.show()

# ГРАФИК СРЕДНЕЙ ЦЕНЫ ПО ВСЕМ РЕГИОНАМ

mean\_price = useful\_data.groupby('region\_name', as\_index=False).agg(avg\_price = ("price", "mean")).round(2)

plt.figure(figsize=(15, 6))

plt.bar(x=mean\_price['region\_name'],height=mean\_price['avg\_price'])

plt.title('График средней цены по всем регионам', fontsize=20)

plt.xticks(rotation=90)

plt.show()

# САМЫЙ ДОРОГОЙ И САМЫЙ ДЕШЕВЫЙ СРЕДИ ВСЕХ РЕГИОНОВ

most\_expensive = useful\_data.sort\_values(by = 'price', ascending=False)

print(most\_expensive[['region\_name', 'price']].head(10).drop\_duplicates())

cheapest = useful\_data.sort\_values(by = 'price', ascending=True)

print(cheapest[['region\_name', 'price']].head(10))

data\_novobl = useful\_data[useful\_data['region\_name'] == 'Новосибирская область'].copy().drop(['region\_name', 'region'], axis = 1)

# ОБЩИЕ ДИАГРАММЫ ЗАВИСИМОСТЕЙ ДЛЯ Новосибирская область

axes = pd.plotting.scatter\_matrix(data\_novobl, figsize=(15,15), diagonal='kde', grid=True)

corr = data\_novobl.corr().values

for i, j in zip(\*plt.np.triu\_indices\_from(axes, k=1)):

axes[i, j].annotate("%.3f" %corr[i,j], (0.6, 0.8), xycoords='axes fraction', ha='center', va='center')

plt.show()

# ПРОВЕРКА БАЛАНСА ДАННЫХ

sns.countplot(data = useful\_data, x='building\_type')

plt.show()

sns.countplot(data = useful\_data, x='level')

plt.show()

sns.countplot(data = useful\_data, x='levels')

plt.xticks(rotation='vertical')

plt.show()

sns.countplot(data = useful\_data, x='rooms')

plt.xticks(rotation='vertical')

plt.show()

sns.countplot(data = useful\_data, x='object\_type')

plt.show()

sns.countplot(data = useful\_data, x='price')

plt.show()

# ПРОВЕРКА ЗАВИСИМОСТИ ОДНОГО ПРИЗНАКА ОТ ДРУГОГО

pd.crosstab(useful\_data['region'], useful\_data['object\_type']).plot(kind='bar', title='Проверка зависимостей')

plt.xticks(rotation='vertical')

plt.show()

pd.crosstab(useful\_data['region'], useful\_data['building\_type']).plot(kind='bar', title='Проверка зависимостей')

plt.xticks(rotation='vertical')

plt.show()

# ЛИНЕЙНАЯ РЕГРЕССИЯ

X = data\_novobl[['price']]

y = data\_novobl['area'].astype(np.int64)

X\_scaled = scaler.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

reg\_logistic\_model = LinearRegression()

reg\_logistic\_model.fit(X\_train, y\_train)

print(reg\_logistic\_model.coef\_)

print(reg\_logistic\_model.intercept\_)

y\_predict = reg\_logistic\_model.predict(X\_test)

plt.figure(figsize=(13, 8))

plt.suptitle(f"Регрессионный анализ", fontsize=20)

plt.title(f'price/area\n'

f'Коэф. детерминации R² = {r2\_score(y\_test, y\_predict):.4f}\n'

f'Среднеквадр. ошибка RMSE = {np.sqrt(mean\_squared\_error(y\_test, y\_predict)):.4f}\n'

f'Перекрёстная проверка = {cross\_val\_score(reg\_logistic\_model, X\_scaled, y, cv=5)}')

plt.scatter(X\_train, y\_train, alpha=0.2, label='Прогноз')

plt.plot(X\_test, y\_predict, 'r--', lw=3)

plt.legend()

plt.tight\_layout()

plt.show()

# ЛИНЕЙНАЯ РЕГРЕССИЯ

X = data\_novobl[['levels']]

y = data\_novobl['object\_type'].astype(np.int64)

X\_scaled = scaler.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

reg\_logistic\_model = LinearRegression()

reg\_logistic\_model.fit(X\_train, y\_train)

print(reg\_logistic\_model.coef\_)

print(reg\_logistic\_model.intercept\_)

y\_predict = reg\_logistic\_model.predict(X\_test)

plt.figure(figsize=(13, 8))

plt.suptitle(f"Регрессионный анализ", fontsize=20)

plt.title(f'level/object\_type\n'

f'Коэф. детерминации R² = {r2\_score(y\_test, y\_predict):.4f}\n'

f'Среднеквадр. ошибка RMSE = {np.sqrt(mean\_squared\_error(y\_test, y\_predict)):.4f}\n'

f'Перекрёстная проверка = {cross\_val\_score(reg\_logistic\_model, X\_scaled, y, cv=5)}')

plt.scatter(X\_train, y\_train, alpha=0.2, label='Прогноз')

plt.plot(X\_test, y\_predict, 'r--', lw=3)

plt.legend()

plt.tight\_layout()

plt.show()

# ЛИНЕЙНАЯ РЕГРЕССИЯ

X = data\_novobl[['area']]

y = data\_novobl['rooms'].astype(np.int64)

X\_scaled = scaler.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

reg\_logistic\_model = LinearRegression()

reg\_logistic\_model.fit(X\_train, y\_train)

print(reg\_logistic\_model.coef\_)

print(reg\_logistic\_model.intercept\_)

y\_predict = reg\_logistic\_model.predict(X\_test)

plt.figure(figsize=(13, 8))

plt.suptitle(f"Регрессионный анализ", fontsize=20)

plt.title(f'area/rooms\n'

f'Коэф. детерминации R² = {r2\_score(y\_test, y\_predict):.4f}\n'

f'Среднеквадр. ошибка RMSE = {np.sqrt(mean\_squared\_error(y\_test, y\_predict)):.4f}\n'

f'Перекрёстная проверка = {cross\_val\_score(reg\_logistic\_model, X\_scaled, y, cv=5)}')

plt.scatter(X\_train, y\_train, alpha=0.2, label='Прогноз')

plt.plot(X\_test, y\_predict, 'r--', lw=3)

plt.legend()

plt.tight\_layout()

plt.show()

# КЛАСТЕРИЗАЦИЯ

X\_scaled = scaler.fit\_transform(useful\_data[['geo\_lat', 'price']])

# Определение оптимального числа кластеров

error = []

silhouette\_scores = {}

for i in range(2, 10):

kmeans = KMeans(n\_clusters = i, random\_state = 42, init = 'k-means++')

kmeans.fit(X\_scaled)

error.append(kmeans.inertia\_)

# ГРАФИК ЛОКТЯ

plt.title('График локтя (geo\_lat, price)')

plt.plot(range(2, 10), error, marker='o')

plt.xlabel('Число кластеров')

plt.ylabel('Ошибка')

plt.tight\_layout()

plt.show()

kmeans = KMeans(n\_clusters = 3, random\_state=42)

kmeans.fit(X\_scaled)

silhouette = silhouette\_score(X\_scaled, kmeans.labels\_)

print(f'Оценка силуэта = {silhouette}')

# Визуализация кластеров

plt.figure(figsize=(10, 6))

plt.title('Кластеризация')

plt.xlabel('географическая широта')

plt.ylabel('цена')

plt.scatter(X\_scaled[:, 0], X\_scaled[:, 1], c=kmeans.labels\_, cmap='viridis', marker='o')

plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1], s=200, c='red', marker='X', label='Centroids')

plt.legend()

plt.show()

# КЛАССИФИКАЦИЯ С ИСПОЛЬЗОВАНИЕМ K-БЛИЖАЙШИХ СОСЕДЕЙ

X = data\_novobl.drop(["rooms"], axis=1)

y = data\_novobl["rooms"]

X\_scaled = scaler.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

knn = KNeighborsClassifier(n\_neighbors = 3, weights='distance')

# тренировка модели

knn.fit(X\_train, y\_train)

y\_pred\_cls = knn.predict(X\_test)

print("Точность классификации:", accuracy\_score(y\_test, y\_pred\_cls))

print(f"F1-Score (kNN): {f1\_score(y\_test, y\_pred\_cls, average='macro'):.2f}")

# Визуализация результатов классификации

labels = ['1', '2', '3', '4']

plt.figure(figsize=(9, 7))

plt.title('Матрица ошибок классификации')

sns.heatmap(confusion\_matrix(y\_test, y\_pred\_cls), annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)

plt.xlabel('Предсказанный статус')

plt.ylabel('Фактический статус количества комнат')

plt.tight\_layout()

plt.show()

# Оцениваем точность классификации с использованием перекрестной проверки

cv\_scores\_cls = cross\_val\_score(knn, X\_scaled, y, cv=5, scoring='accuracy')

print("Перекрестная проверка точности =", np.mean(cv\_scores\_cls))