МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ
Департамент информационных и компьютерных систем

Курс «Компьютерные методы анализа больших данных»

Контрольная работа №3
Контрольное мероприятие по рейтингу

на тему «Сравнительный анализ логистической регрессии и метода опорных векторов для задачи бинарной классификации»

Вариант №15

Выполнил студент Б9122-01.03.02мкт
Пелагеев Д.И.
_____

Проверил доцент Достовалов В.Н.

г. Владивосток
2025

Оглавление

Введение

Целью данной работы является построение и оценка моделей для предсказания типа квартиры (1-2-комнатные или 3-4-комнатные) на основе нескольких факторов, таких как цена и общая площадь квартиры. Данные предоставлены в файле `flsr_moscow.txt`, который содержит информацию о различных квартирах в Москве. Для решения задачи используется логистическая регрессия.

Датасет содержит следующие переменные:

— `price` – цена квартиры в $1000.

— `nrooms` – количество комнат в квартире.

— `totsp` – общая площадь квартиры (кв.м).

— `livesp` – жилая площадь квартиры (кв.м).

— `kitsp` – площадь кухни (кв.м).

— `dist` – расстояние от центра города (км).

— `metrdist` – расстояние до метро в минутах.

— `walk` – индикатор, указывающий, пешком ли добираться до метро (1 — пешком, 0 — на транспорте).

— `brick` – тип строительства (1 — кирпич, монолит, 0 — другой).

— `floor` – этаж, на котором расположена квартира (1 — этаж не первый и не последний, 0 — первый или последний этаж).

— `code` – код района (от 1 до 8):

а) Наблюдения сгруппированы на севере, вокруг Калужско-Рижской линии метрополитена

б) Север, вокруг Серпуховско-Тимирязевской линии метрополитена

в) Северо-запад, вокруг Замоскворецкой линии метрополитена

г) Северо-запад, вокруг Таганско-Краснопресненской линии метрополитена

д) Юго-восток, вокруг Люблинской линии метрополитена

е) Юго-восток, вокруг Таганско-Краснопресненской линии метрополитена

ж) Восток, вокруг Калининской линии метрополитена

з) Восток, вокруг Арбатско-Покровской линии метрополитена

Для решения задачи были использованы два метода: логистическая регрессия и метод опорных векторов (SVM). Для оценки качества моделей использовались тестовая и валидационная выборки.

Описание задачи работы:

1) Прочитать файл с помощью команды: `dd<-read.table(file='flsr_moscow.txt',head=TRUE)`

2) На основе переменной nrooms создать факторную переменную `target` с двумя уровнями «1,2 - комнатные квартиры» vs «3,4-комнатные квартир».

3) Построить модель логистической регрессии для построенного в п.2 `subset`. Объясняемая переменная `target`. Регрессоры – `price`, `totsp`. Построить модель классификации для построенного в п.2 `subset` по переменным – `price`, `totsp`. Оценить качество моделей. Построить визуализацию задачи.

1 Подготовка данных

1.1 Загрузка и первичная обработка данных

Прежде чем начать работу с данными в RStudio [1] на языке R [2], необходимо установить рабочую директорию, в которой будет находиться наш датасет `flsr_moscow.txt`.

Листинг 1 — Установка директории

```
setwd("/Users/daniil/Desktop/Education/BigData")
```

Теперь загружаем датасет `flsr_moscow.txt` с помощью функции `read.table`.

Листинг 2 — Загрузка данных

```
dd ← read.table(file='flsr_moscow.txt', header=TRUE)
```

Проверяем размер загруженного датасета до чистки. Осуществляем чистку датасета от пустых строк и после проверямем размерность. Также не забываем убрать ненужные столбцы.

Листинг 3 — Очистка данных

```
dim(dd)
dd_clean ← na.omit(dd)
dd_clean ← dd_clean[-4:-11]
dim(dd_clean)

head(dd_clean)
```

Исходные данные содержали 2040 наблюдений и 11 переменных. Для анализа были использованы только три переменные: `price`, `totsp` и `nrooms`. После очистки получилось 2040 наблюдений и 3 переменные.

1.2 Формирование подмножеств

На основе переменной `nrooms` была создана факторная переменная `target`, которая классифицирует квартиры как "1,2-комнатные"или "3,4-комнатные".

Листинг 4 — Создание целевой переменной

```
dd_clean$target ← factor(ifelse(dd_clean$nrooms %in% c(1,
    2), "1,2-rooms flat", "3,4-rooms flat"))
```

Далее данные были разделены на тренировочную (70%), тестовую (15%) и валидационную (15%) выборки.

Листинг 5 — Разделение на выборки

```
set.seed(42)
train_idx ← sample(1:nrow(dd_clean), size = 0.7 *
    nrow(dd_clean))
remaining_idx ← setdiff(1:nrow(dd_clean), train_idx)

test_idx ← sample(remaining_idx, size = 0.5 *
    length(remaining_idx))
validation_idx ← setdiff(remaining_idx, test_idx)

train_data ← dd_clean[train_idx, ]
test_data ← dd_clean[test_idx, ]
validation_data ← dd_clean[validation_idx, ]
```

2 Построение моделей

2.1 Логистическая регрессия

2.1.1 Построение модели

Модель логистической регрессии была построена на тренировочной выборке с использованием переменных `price` и `totsp` для предсказания класса квартиры.

Листинг 6 — Создание модели логистической регрессии

```
log_reg_model ← glm(target ~ price + totsp,
    data=train_data, family="binomial")
```

2.2 Метод опорных векторов (SVM)

2.2.1 Построение модели

Для построения модели SVM использовалась функция `ksvm` из пакета `kernlab` [3] с радиальной базисной функцией ядра (`rbfdot`) и параметром регуляризации $C = 1$.

Листинг 7 — Построение модели SVM

```
svm_model ← ksvm(target ~ price + totsp, data=train_data,
    kernel="rbfdot", C=1)
```

3 Оценка моделей

3.1 Логистическая регрессия

3.1.1 Оценка на тестовой выборке

После построения модели логистической регрессии, мы использовали тестовую выборку для предсказания классов квартир. Предсказания получались с использованием функции `predict` с типом `"response"`, которая возвращает вероятность принадлежности к положительному классу.

Листинг 8 — Предсказания и оценка модели логистической регрессии на тестовой выборке

```
log_reg_test_predictions ← predict(log_reg_model,
    newdata=test_data, type="response")
log_reg_test_predicted_class ←
    ifelse(log_reg_test_predictions > 0.5, "3,4-rooms flat",
    "1,2-rooms flat")

log_reg_test_conf_matrix ←
    caret::confusionMatrix(factor(log_reg_test_predicted_class),
    test_data$target)
print("Confusion Matrix for logistic regression on a test
    sample:")
print(log_reg_test_conf_matrix)
```

3.1.2 Оценка на валидационной выборке

Аналогично тестовой выборке, предсказания были сделаны на валидационной выборке.

Листинг 9 — Оценка модели логистической регрессии на валидационной выборке

```
1    log_reg_validation_predictions ← predict(log_reg_model,
        newdata=validation_data, type="response")
2    log_reg_validation_predicted_class ←
        ifelse(log_reg_validation_predictions > 0.5, "3,4-rooms
        flat", "1,2-rooms flat")
3
4    log_reg_validation_conf_matrix ←
        caret::confusionMatrix(factor(log_reg_validation_predicted_class),
        validation_data$target)
5    print("Confusion Matrix for logistic regression on a
        validation sample:")
6    print(log_reg_validation_conf_matrix)
```

3.2 Метод опорных векторов (SVM)

3.2.1 Оценка на тестовой выборке

После построения модели SVM, мы делаем предсказания на тестовой выборке и оцениваем качество модели с помощью матрицы путаницы.

Листинг 10 — Оценка модели SVM на тестовой выборке

```
1    svm_test_predictions ← predict(svm_model,
        newdata=test_data)
2    svm_test_conf_matrix ←
        caret::confusionMatrix(factor(svm_test_predictions),
        test_data$target)
3    print("Confusion Matrix for the SVM model on the test
        sample:")
4    print(svm_test_conf_matrix)
```

3.2.2 Оценка на валидационной выборке

Аналогично логистической регрессии, предсказания были сделаны на валидационной выборке.

Листинг 11 — Оценка модели SVM на валидационной выборке

```
svm_validation_predictions ← predict(svm_model,
    newdata=validation_data)
svm_validation_conf_matrix ←
    caret::confusionMatrix(factor(svm_validation_predictions),
    validation_data$target)
print("Confusion Matrix for the SVM model on the validation
    sample:")
print(svm_validation_conf_matrix)
```

3.3 Матрицы путаницы

3.3.1 Логистическая регрессия

Тестовая выборка:

Prediction	**1,2-комнатные квартиры**	**3,4-комнатные квартиры**
1,2-комнатные квартиры	152	0
3,4-комнатные квартиры	0	154

Таблица 1 — Матрица путаницы для логистической регрессии на тестовой выборке

Валидационная выборка:

Prediction	**1,2-комнатные квартиры**	**3,4-комнатные квартиры**
1,2-комнатные квартиры	143	0
3,4-комнатные квартиры	0	163

Таблица 2 — Матрица путаницы для логистической регрессии на валидационной выборке

3.3.2 Метод опорных векторов (SVM)

Тестовая выборка:

Валидационная выборка:

3.3.3 Анализ

На основе представленных матриц путаницы (Confusion Matrix) [4] можно сделать вывод, что обе построенные модели — логистическая регрессия и метод опорных векторов (SVM) — демонстрируют идеальную точность классификации, достигая 100% как на тестовой, так и на валидационной выборках. Это означает, что все наблюдения были правильно классифицированы без каких-либо ошибок.

Идеальная точность моделей может свидетельствовать о нескольких факторах:

— **Отсутствие сложности в данных**: Возможно, данные были настолько хорошо разделимы, что обе модели смогли безошибочно классифицировать все наблюдения. Это может происходить, если выбранные признаки (`price` и `totsp`) имеют сильную корреляцию с целевой переменной.

— **Недостаток разнообразия данных**: Если данные недостаточно разнообразны или содержат сильные закономерности, модели могут легко их выучить. Это может ограничить способность моделей к обобщению на новых, более сложных данных.

Prediction	**1,2-комнатные квартиры**	**3,4-комнатные квартиры**
1,2-комнатные квартиры	152	0
3,4-комнатные квартиры	0	154

Таблица 3 — Матрица путаницы для модели SVM на тестовой выборке

4 Визуализация

4.1 Построение графиков

Листинг 12 — Построение графиков

```
ggplot(dd_clean, aes(x=target, y=price, fill=target)) +
  geom_boxplot() +
  labs(title="Distribution of prices by number of rooms",
       x="Type of apartment", y="Price") +
  theme_minimal()


ggplot(dd_clean, aes(x=price, y=totsp, color=target)) +
  geom_point() +
  labs(title="Visualization of the classification task",
       x="Price", y="Total floor area") +
  theme_minimal()
```

Построение графиков было сделано с помощью библиотеки ggplot2 [5]. С помощью него можно быстро построить информативные графики, которые помогают в анализе данных.

4.2 Зависимость цены от общей площади

В общем видна относительная граница между 1-2 комнатными и 3-4 комнатными квартирами, которая проходит в диапазоне площадей около 75 кв.м и цен около 200 тыс. рублей. Эта граница не является четкой, так как существуют пересечения между классами. Благодаря давольно явной границе, мы можем с высокой точностью классифицировать данные.

Prediction	**1,2-комнатные квартиры**	**3,4-комнатные квартиры**
1,2-комнатные квартиры	143	0
3,4-комнатные квартиры	0	163

Таблица 4 — Матрица путаницы для модели SVM на валидационной выборке

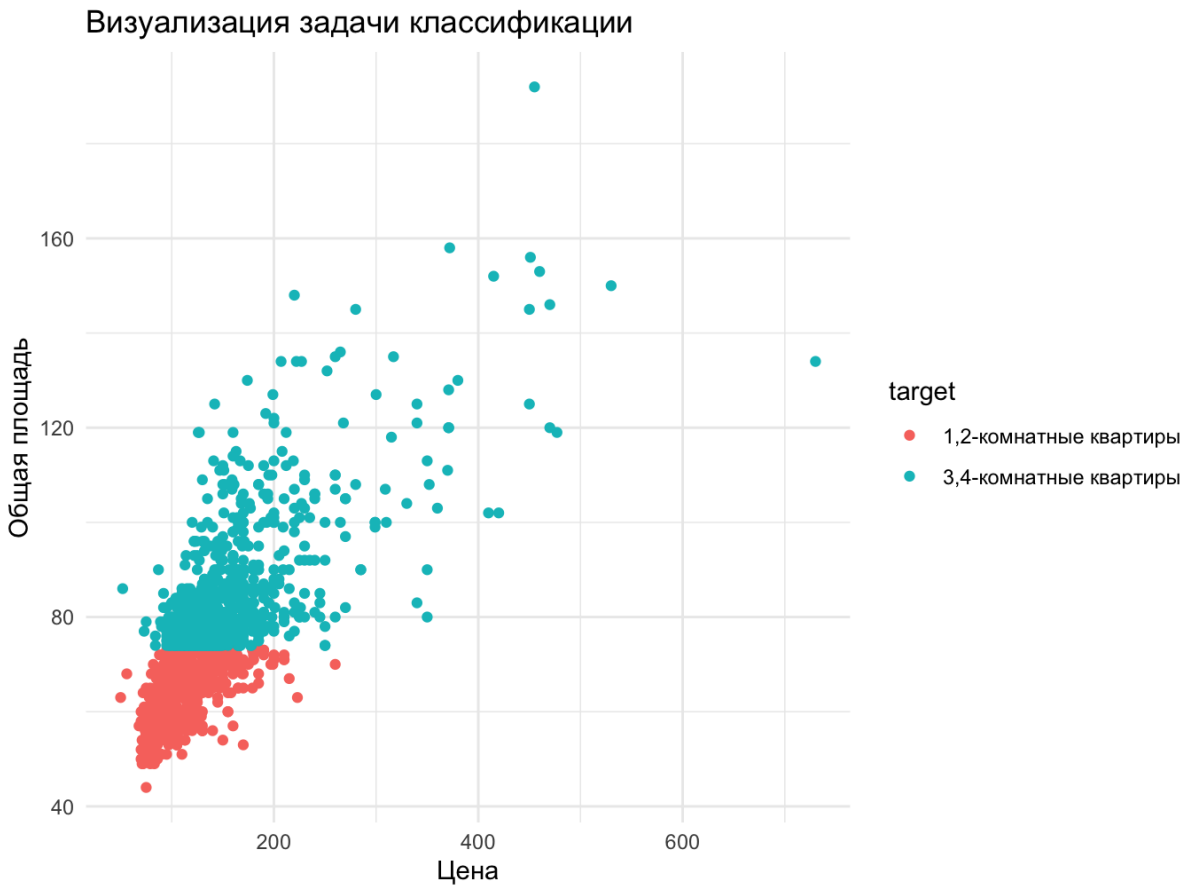

Рисунок 1 — Визуализация задачи классификации: зависимость цены от общей площади

4.3 Распределение цен по количеству комнат

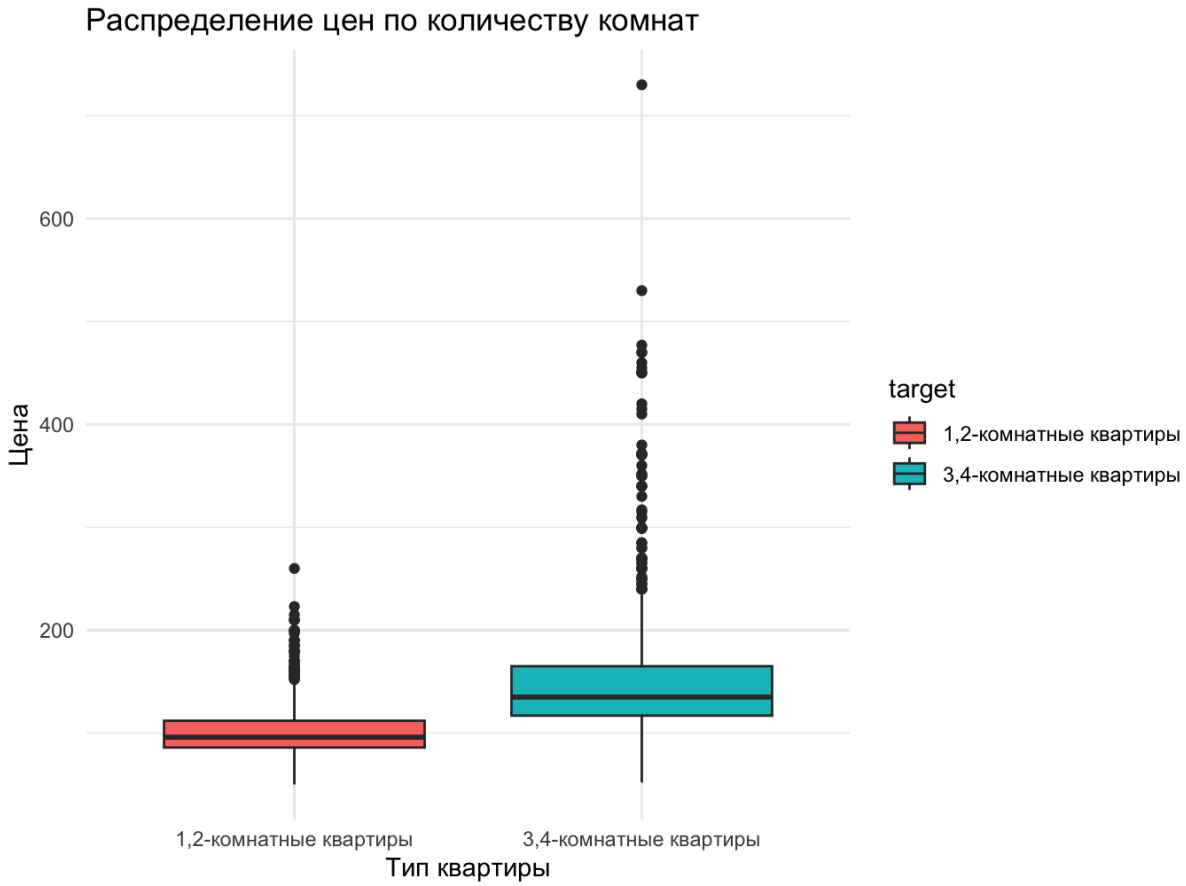

Рисунок 2 — Распределение цен по количеству комнат

На Рисунок 2 видно следующее:

Как мы видим по этому графику, распределение цен для квартир с 1-2 комнатами существенно отличается от распределения цен для квартир с 3-4 комнатами. Квартиры с 1-2 комнатами имеют медианную цену ниже, чем у квартир с 3-4 комнатами. Диапазон цен для 1-2 комнатных квартир ограничен, что подтверждает их бюджетность. В то же время, квартиры с 3-4 комнатами демонстрируют более широкий диапазон цен, что указывает на разнообразие в их характеристиках.

Кроме того, на графике присутствуют выбросы для обоих типов квартир. У квартир с 1-2 комнатами выбросы находятся выше верхнего предела, но они более редкие и менее значительные. У квартир с 3-4 комнатами выбросы более выражены и поднимаются до 600 тыс. рублей, что может отражать элитные квартиры.

Таким образом, цена является важным фактором для разделения классов, так как медианы цен и их распределение значительно различаются. Однако, из-за наличия выбросов и перекрытия диапазонов цен на более низком уровне (до 200 тыс. рублей), классификация только по цене может быть затруднительной.

Заключение

В результате проведенного анализа были построены и оценены две модели классификации типов квартир в Москве: логистическая регрессия и метод опорных векторов (SVM). Обе модели продемонстрировали высокую точность классификации, достигая 100% точности на тестовых и валидационных выборках. Как вяснилось, это указывает на то, что данные давольно явно распределены.

Список использованных источников

1. Официальная страница интегрированной среды разработки RStudio. — 2024. — Режим доступа: RStudio | Open source & professional software for data science teams - RStudio (дата обращения: 21.01.2025).

2. Официальная страница среды статистического моделирования R. — 2024. — Режим доступа: R: The R Project for Statistical Computing (дата обращения: 21.01.2025).

3. *Alexandros Karatzoglou*. Kernel-Based Machine Learning Lab. — 2024. — Режим доступа: Kernel-Based Machine Learning Lab (дата обращения: 21.01.2025).

4. *Max Kuhn*. The caret Package. — 2019. — Режим доступа: The caret Package (дата обращения: 21.01.2025).

5. *Wickham, Hadley*. ggplot2: Elegant Graphics for Data Analysis (Use R!) / Hadley Wickham. — New York: Springer, 2009. — Режим доступа: ggplot2 (дата обращения: 21.01.2025).