МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Омский государственный технический университет»

Факультет информационных технологий и компьютерных систем

Кафедра «Прикладная математика и фундаментальная информатика»

Домашнее задание

по дисциплине Практикум по программированию

Студента(ки) Сагалбаева Дамира Амангельдыевича
фамилия, имя, отчество полностью

Курс 2 Группа ФИТ-222

Направление 02.03.02. Фундаментальная информатика и информационные технологии
код, наименование

Руководитель ст.преподаватель
должность, ученая степень, звание

Саматов А. П.
фамилия, инициалы, дата, подпись

Выполнил
дата, подпись студента(ки)

Итоговый рейтинг	

Омск 2023

ВВЕДЕНИЕ

Объемы данных сегодня стали настолько обширными, что ручной анализ становится трудозатратным. Тем не менее, этот анализ остается важным для принятия решений, создания статистических отчетов и построения моделей машинного обучения. В процессе курса использовались следующие библиотеки Python:

1. **NumPy**: Это библиотека с открытым исходным кодом, предоставляющая поддержку многомерных массивов (включая матрицы) и высокоуровневых математических функций, специально разработанная для работы с многомерными данными.
2. **Matplotlib**: Эта библиотека предназначена для визуализации данных и позволяет построить как двумерные, так и трехмерные графики. Она играет важную роль в наглядном представлении информации.
3. **SymPy**: Данная библиотека представляет собой инструмент для символьных вычислений в Python. Она полезна при решении задач компьютерной алгебры и предоставляет функциональность для символьных выражений.
4. **SciPy**: Это открытая библиотека, предназначенная для выполнения научных и инженерных расчетов. Она предоставляет множество функций для решения различных задач, связанных с наукой и инженерией.
5. **Pandas**: Эта библиотека предоставляет удобные структуры данных и операции для обработки и анализа данных. Pandas часто используется для манипуляций с числовыми таблицами и временными рядами.
6. **Seaborn**: Эта библиотека создана для создания статистических графиков на Python и является дополнением к Matplotlib. Она интегрируется тесно с pandas, обеспечивая удобные средства визуализации статистики.

Эти библиотеки обеспечивают обширные возможности по обработке, анализу и визуализации данных, а также позволяют строить статистику на их основе.

1.Поиск и загрузка данных

Использован датасет Mumbai_house, специализирующемся на исследовании данных и машинном обучении.

README.md

Расчетно-графическая работа по дисциплине "Практикум по программированию"

Использован датасет Mumbai houses Dataset

В нем представлены:

- Price (Цена): Стоимость недвижимости в местной валюте.
- Area (Площадь): Площадь недвижимости в квадратных единицах (например, квадратных футах или квадратных метрах).
- Latitude (Широта): Географическая координата широты местоположения недвижимости.
- Longitude (Долгота): Географическая координата долготы местоположения недвижимости.
- Bedrooms (Спальни): Количество спален в недвижимости.
- Bathrooms (Ванные комнаты): Количество ванных комнат в недвижимости.
- Balcony (Балкон): Количество балконов в недвижимости.
- Status (Статус): Статус строительства недвижимости, указывающий, находится ли она "В процессе строительства" или "Готова к заселению".
- Neworold (Новая или вторичная): Классификация недвижимости как "Новая" или "Вторичная" (предварительно владелась).
- Parking (Парковка): Количество парковочных мест, связанных с недвижимостью.
- Furnished_status (Уровень мебелировки): Степень меблировки в недвижимости, такая как "Не меблирована", "Частично меблирована" или "Меблирована".
- Lift (Лифт): Наличие лифта или подъемника в здании.
- Type_of_building (Тип здания): Тип здания, часто указанный как "Квартира" в этом наборе данных.

Рисунок 1 – файл README.md

Датасет был загружен в ноутбук командой read_csv() библиотеки pandas.

```
data=pd.read_csv('mumbai_houses_task.csv')
```

Рисунок 2 – загрузка датасета

Данный датасет состоит из 6255 строк и 13 столбцов.

In [3]:
```
data.head(5)
```

Out[3]:

	price	area	latitude	longitude	Bedrooms	Bathrooms	Balcony	Status	neworold	parking	Furnished_status	Lift	ty
0	22400000.0	629.0	19.032800	72.896357	2.0	2.0	0.0	Under Construction	New Property	0.0	NaN	0.0	
1	35000000.0	974.0	19.032800	72.896357	3.0	2.0	0.0	Under Construction	New Property	0.0	NaN	0.0	
2	31700000.0	968.0	19.085600	72.909277	3.0	3.0	0.0	Under Construction	New Property	0.0	NaN	0.0	
3	18700000.0	629.0	19.155756	72.846862	2.0	2.0	2.0	Ready to Move	New Property	2.0	NaN	2.0	
4	13500000.0	1090.0	19.177555	72.849887	2.0	2.0	0.0	NaN	New Property	0.0	Unfurnished	0.0	

Рисунок 3 – небольшая часть датасета, выведенного в виде таблицы

2.1 Гистограмма распределения числового признака

Гистограмма — способ представления табличных данных в графическом виде — в виде столбчатой диаграммы. Количественные соотношения некоторого показателя представлены в виде прямоугольников, площади которых пропорциональны. На гистограмме видно количество спальных комнат.

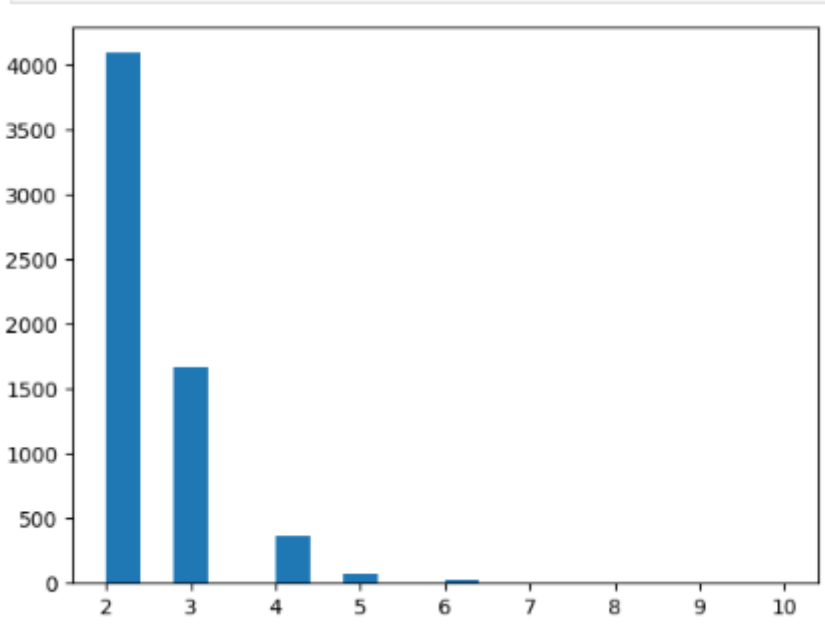

Рисунок 4 – гистограмма столбца Bedrooms

2.2 Диаграмма «ящик с усами» числового признака

Диаграмма «ящик с усами» — график, использующийся в описательной статистике, компактно изображающий одномерное распределение вероятностей. 7 Такой вид диаграммы в удобной форме показывает медиану (или, если нужно, среднее), нижний и верхний квартили, минимальное и максимальное значение выборки и выбросы. Несколько таких ящиков можно нарисовать бок о бок, чтобы визуально сравнивать одно распределение с другим; их можно располагать как горизонтально, так и вертикально. Расстояния между различными частями ящика позволяют определить степень разброса (дисперсии) и асимметрии данных и выявить выбросы.

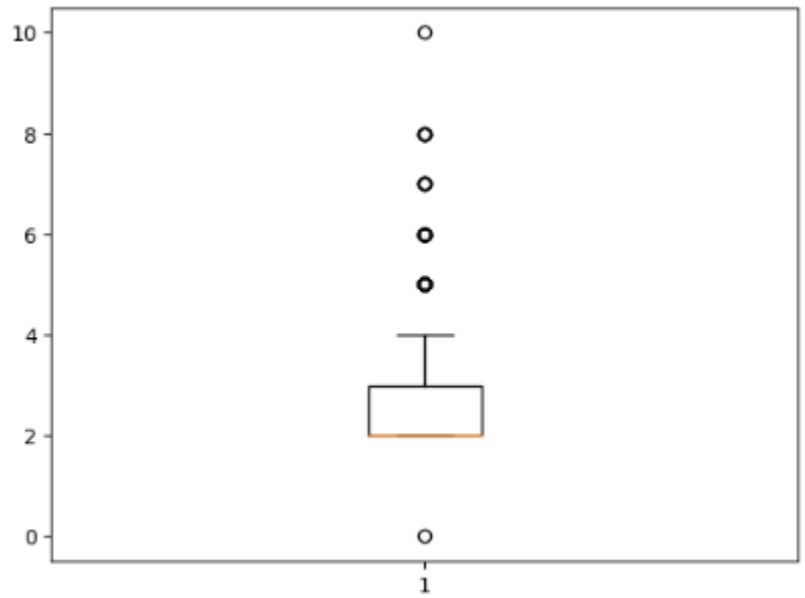

Рисунок 5 – Диаграмма «ящик с усами» столбца Bathrooms

2.3 Круговая диаграмма номинативного признака

Круговая диаграмма — это круговая статистическая диаграмма, которая разделена на срезы, чтобы проиллюстрировать числовую пропорцию. На круговой диаграмме длина дуги каждого среза пропорциональна величине, 8 которую он представляет. На данной круговой диаграмме видно, балконов чаще всего в квартирах одна штука.

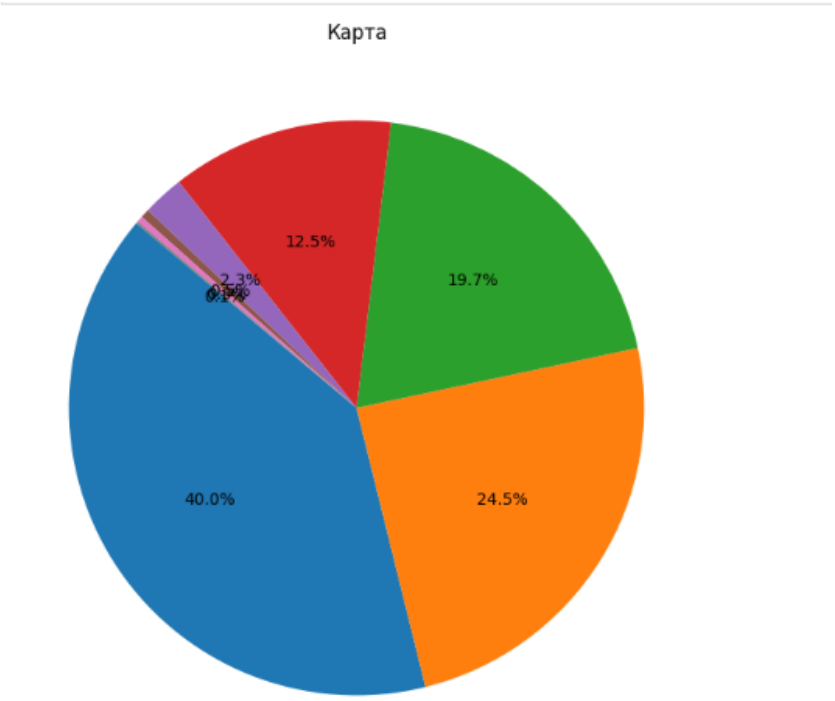

Рисунок 6 – Круговая диаграмма

2.4 Тепловая карта

Тепловая карта — графическое представление данных, где индивидуальные значения в таблице отображаются при помощи цвета. На тепловой карте данного датасета можно выявить несколько особенностей, что корреляции между данными есть.

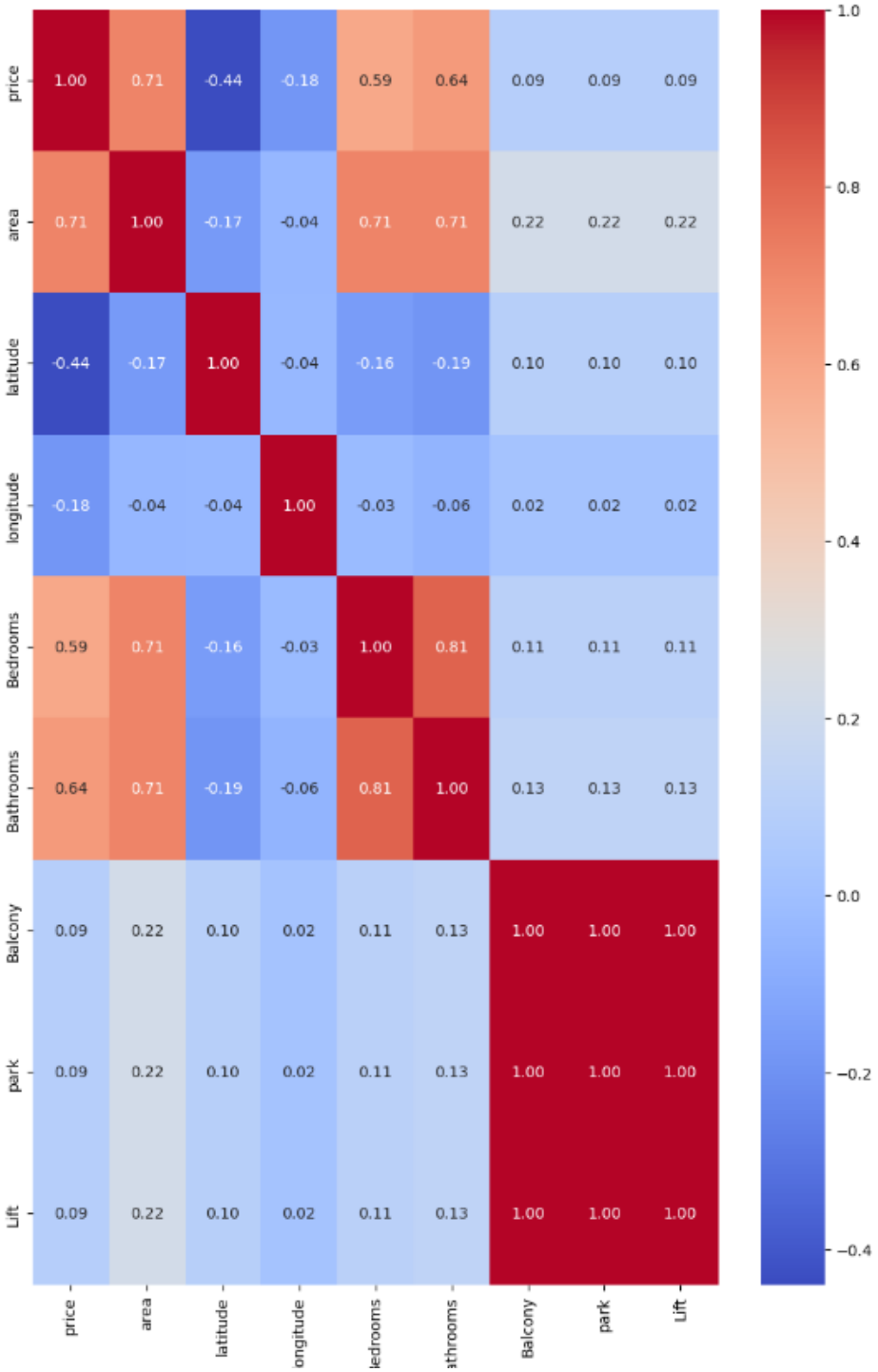

Рисунок 7 –фрагмент тепловой карты датасета

Данные содержат пропуски, и для их заполнения будет использоваться подход с использованием среднего значения и моды, поскольку исходные данные содержат пропущенные значения.

```
[]: data.isna().sum()

[]: price              0
    area               0
    latitude           0
    longitude          0
    Bedrooms           0
    Bathrooms          0
    Balcony            0
    Status             0
    neworold           0
    park               0
    Furnished_status   0
    Lift               0
    type_of_building   0
    dtype: int64
```

Рисунок 9 – Проверка на наличие пропусков в таблице

Были созданы одна переменная:

1. «columns_to_replace»: содержит названия колонок, которые имеют пропуски.

```
columns_to_replace=['Status','Furnished_status']
for column in columns_to_replace:
```

Рисунок 10 – Создание переменных

Далее были заменены пропущенные значения на “unknown”:

```
for column in columns_to_replace:
  data[column] = data[column].fillna('unknown')
```

Рисунок 11 – Замена пропущенных значений

Результат:

```
data.isna().sum()
```

```
price              0
area               0
latitude           0
longitude          0
Bedrooms           0
Bathrooms          0
Balcony            0
Status             0
neworold           0
park               0
Furnished_status   0
Lift               0
type_of_building   0
dtype: int64
```

Рисунок 12 – Результат обработки пропущенных значений

Также было применено one-hot кодирование, то есть преобразование категориальных переменных в численные путем создания столбцов под каждую категорию и заполнения их значениями 0 и 1 в зависимости от категории каждой строчки.

```
data= pd.get_dummies(data, columns=['Status', 'neworold', 'Furnished_status', 'type_of_building'], drop_first=True)
```

Рисунок 13 – Горячее кодирование

Результат:

ooms	Balcony	park	Lift	Status_Under Construction	Status_unknown	neworold_Resale	Furnished_status_Semi-Furnished	Furnished_status_Unfurnished	Furnish
2.0	0.0	0.0	0.0	1	0	0	0	0	
2.0	0.0	0.0	0.0	1	0	0	0	0	
3.0	0.0	0.0	0.0	1	0	0	0	0	
2.0	2.0	2.0	2.0	0	0	0	0	0	
2.0	0.0	0.0	0.0	0	1	0	0	1	

Рисунок 14 – Горячее кодирование

Обработанные данные были сохранены в формате .csv в той же директории, что и изначальный датасет.

```
]: data.to_csv('mumbai.csv')
```

Рисунок 11 –Экспорт датасета

ЗАКЛЮЧЕНИЕ

В ходе практики были освоены и применены ключевые библиотеки Python: matplotlib, seaborn, pandas и numpy. Эти инструменты являются фундаментальными для работы с данными и визуализации результатов.

Matplotlib предоставляет возможность создавать разнообразные графики и диаграммы, сделав данные более доступными и интерпретируемыми. Seaborn расширяет возможности визуализации, позволяя создавать более сложные графики, такие как тепловые карты и распределения данных.

Pandas облегчает работу с данными в формате таблицы, обеспечивая удобство анализа и обработки данных. А библиотека NumPy предоставляет мощные средства для математических операций с массивами данных.

Применение этих библиотек позволило успешно решить различные задачи по анализу данных и визуализации результатов. Создание графиков, диаграмм, тепловых карт и распределений способствовало лучшему пониманию структуры данных и выявлению закономерностей.

В общем, использование данных библиотек значительно ускоряет процесс анализа данных, повышает точность выводов и является неотъемлемым инструментом для специалистов, занимающихся анализом данных в среде Python.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. https://numpy.org/doc/stable/reference/generated/numpy.matrix.html (дата обращения: 04.11.23).
2. https://seaborn.pydata.org/installing.html(дата обращения: 04.10.23).
3. https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html (дата обращения: 04.11.23).
4. https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.tight_layout.html (дата обращения: 04.11.23).