# Постановка задачи

Целью проекта является научиться предсказывать цену квартиры в Магнитогорске, основываясь на данных сайта <http://citystar.ru>.

# Выбор и получение данных

Для проекта были выбраны объявления о продаже квартир с разным количеством комнат. Таких объявлений было достаточное количество на сайте (1400 объявлений). Данные получим с помощью парсинга страницы. Для этого используем библиотеку requests, чтобы получать html код страницы, и BeautifulSoup, чтобы из кода страницы доставать необходимые данные . Поочередно собираем объявления о продаже однокомнатных, двухкомнатных, трехкомнатных и четырехкомнатных квартир. Из каждого объявления извлекаем следующие характеристики:

* Количество комнат
* Этаж
* Максимальный этаж в доме
* Район
* Общая площадь
* Жилая площадь
* Площадь кухни
* Цена

Далее проводим разведывательный анализ полученных данных, чтобы выяснить все зависимости, а также удалить лишние данные.

В результате мы выяснили:

* В основном объявления с 3 районов: Орджоникидзевский, Ленинский, Правобережный (рис. 1 из приложения)
* Объявлений о продаже 2-х комнатных квартир значительное большинство (рис. 2 из приложения)
* Из – за предыдущего пункта цена квартир находиться в диапазоне от 2 млн руб до 4 млн руб, но есть отклонения в большую сторону (из-за квартир с большим количеством комнат) (рис. 3 из приложения)
* Главным признаком, влияющим на цену квартиры является общая площадь, потом идет количество комнат и жилая площадь (рис. 4 из приложения)

# Описание алгоритма решения

План действий:

1. Обработка данных (основываясь на результатах анализа)
2. Получение новых признаков
3. Выбор модели, основываясь на метрике качества
4. Обучение выбранной модели
5. Предсказание цен на тестовых объектах

В этом разделе подробнее остановимся на 1 и 2 пункте, остальные рассмотрим в следующих разделах.

Проанализировав данные, мы поняли, что объектов в непопулярных районах очень мало (0.8% от общей выборки), поэтому мы можем ими пренебречь и убрать из обучающей выборки. Также есть объекты, район которых не определен. Так как таких объектов много, то удалить их нельзя (потеряем часть выборки). Поэтому будем заполнять пропуски, опираясь на популярный район для каждого типа квартиры. Под типом квартиры понимается 1, 2-х, 3-х, 4-х комнатная.

Еще убираем объект, этаж которого отрицательный. Он один, поэтому количество объектов в выборке не пострадает.

Так как район – это текстовая информация, нам необходимо его векторизовать. Для этого будем использовать метод OneHotEncoding. Мы создаем признаки Орджоникидзевский, Ленинский, Правобережный, в которых будет находиться 0 или 1 в зависимости от того, к какому району принадлежит объект.

Чтобы усилить признак «максимальный этаж», необходимо вычесть из максимального этажа текущий этаж объекта и эту разность использовать как признак.

Также перед обучением моделей мы нормализуем все данные (для некоторых моделей, которые мы будем проверять, это важно).

# Описание моделей

Так как перед нами стоит задача регрессии, то проверим модели такого типа: от простых до сложных. Важный момент: все модели будут в будущем тестироваться "из коробки», то есть без изменения гиперпараметров. Подбор оптимальных параметров будет производиться для лучшей модели из списка ниже.

1. Ridge – линейная регрессия с регуляризацией
2. Стохастический градиентный спуск – оптимизация метрики качества с помощью взятия градиента
3. Случайный лес – набор деревьев решений
4. Градиентный бустинг – направленный набор деревьев решений, в котором каждое дерево улучшает предыдущее

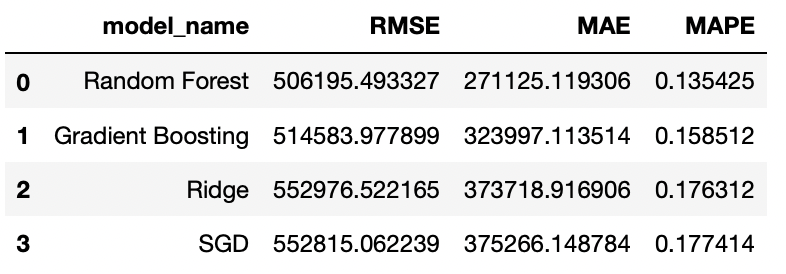
Все модели взяты из библиотеки sklearn

# Выбор метода решения

Разделим данные на тренировочные и тестовые в пропорции 80 на 20. Будем обучать каждую модель из списка выше на тренировочной выборке и делать предсказания на тестовой выборке, а дальше замерять следующие метрики качества:

1. Средняя квадратичная ошибка
2. Средняя абсолютная ошибка
3. Средняя относительная ошибка

В результате получим следующий результат:



Вывод: дальше используем модель Random Forest (Случайный лес)

Дальше выполним перебор гиперпараметров, чтобы найти набор, который будет еще сильнее минимизировать ошибку.

# Описание качества модели

В результате обучения модели мы получили следующие метрики качества:

* Средняя относительная ошибка: 13,5%
* Средняя абсолютная ошибка: 270 тыс. рублей
* Корень из средней квадратичной ошибки: 500 тыс. рублей

Большая разница между второй и третей метрикой качества обуславливается тем, что в данных есть хвост из квартир с высокой ценой. На этих объектах модель ошибается сильнее, а средняя квадратичная ошибка сильнее штрафует за выбросные значения.

# Вывод

В результате работы были собраны и обработаны данные, выбрана и обучена модель, определено качество модели. Модель можно протестировать с помощью api.

Описание доступных методов:

* predict – POST запрос, который на вход принимает объект в виде его признакового описания. Это json со следующими ключами:

1. 'room\_number'
2. 'district'
3. 'floor'
4. 'max\_floor'
5. 'general\_area'
6. 'living\_area'
7. 'kitchen\_area'

Тестовые объекты:

1. {"room\_number": 1, "floor": 4,"max\_floor": 10,"general\_area": 40.4,"living\_area": 19.0,"kitchen\_area": 12.0,"district": "Орджоникидзевский"}

2. {"room\_number": 1, "floor": 10,"max\_floor": 10,"general\_area": 41.4,"living\_area": 25.0,"kitchen\_area": 10.0,"district": "Ленинский"}

3. {"room\_number": 2, "floor": 4,"max\_floor": 5,"general\_area": 45.6,"living\_area": 29.0,"kitchen\_area": 6.0,"district": "Правобережный"}

# Приложение

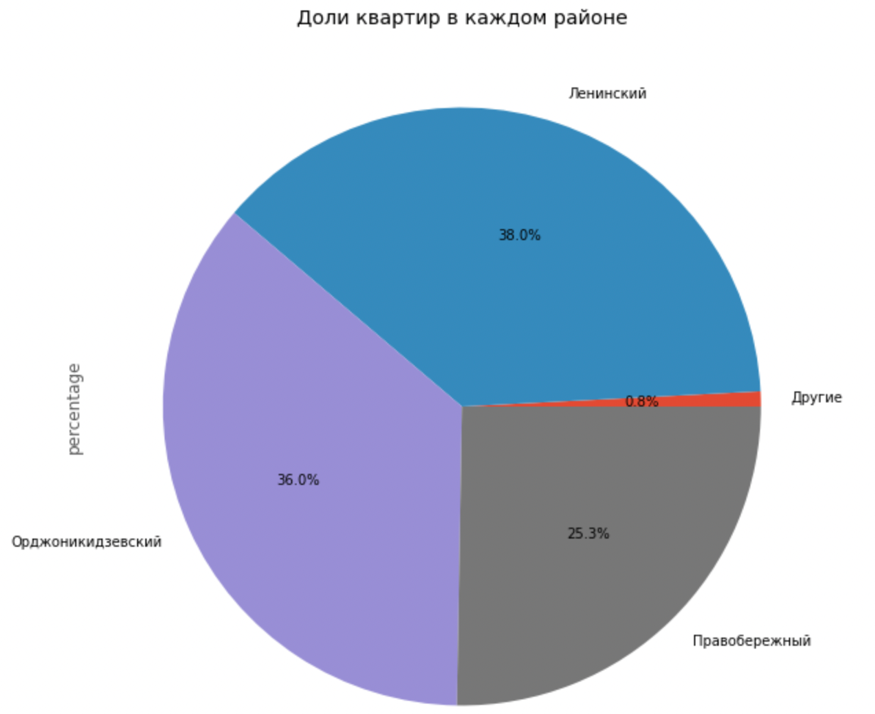


Рис. 1



Рис. 2

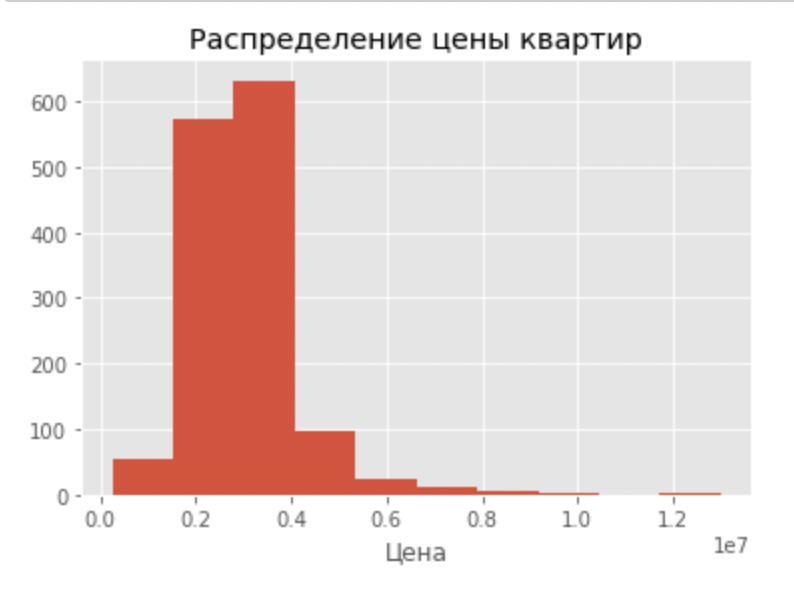


Рис. 3

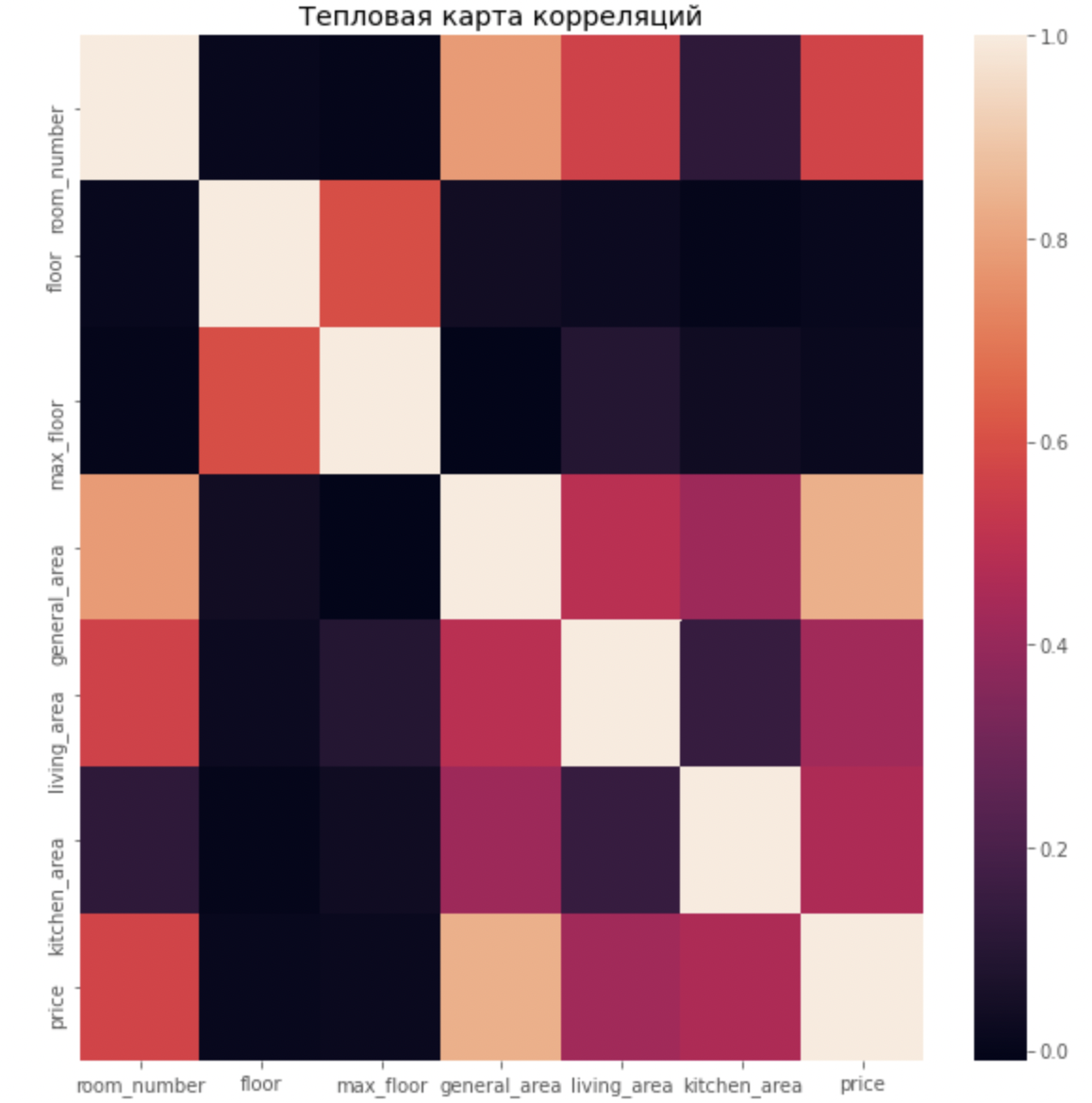


Рис. 4