Выводим графики зависимостей цен при фиксир значениях других параметров

Сигма отсечение выбросов (стр 235, 236 python для сложных задач )

Стр 365 рашка ---- параметры цен на дома

# анализ матрицы

1 нахождение выбросов

А)

df = pd.read\_csv(**r'd:\work\autoanalysis\python\progs\scraper\flats.csv'**)  
df = df.drop\_duplicates([**'total\_square'**, **'total\_floors'**, **'floor'**, **'rooms\_num'**, **'build\_age'**, **'dist\_cent'**, **'house\_type'**,**'cl\_adr'**])

list\_stbs = [**'cost'**, **'total\_square'**, **'live\_square'**, **'total\_floors'**, **'floor'**, **'rooms\_num'**, **'build\_age'**, **'dist\_cent'**]

avito.avito\_flat\_parser.AvitoFlatParser.cleanPrepare(df, list\_stbs)  
df = df[list\_stbs]

df\_desc=df.describe()

df\_desc[‘имя столбца’]

Б)

list\_stbs = [**'cost'**, **'total\_square'**, **'live\_square'**, **'total\_floors'**, **'floor'**, **'rooms\_num'**, **'build\_age'**, **'dist\_cent'**]

*draw\_distribution(df3, cleaned=True, num\_classes=10)  
или  
draw\_distribution(df, list\_stbs, num\_classes=10)*

Здесь распределение по 10 группам для каждого столбца (некоторые графики меньше 10 групп содержат)

Обнаружив аномалии

Смотрим какие это записи

*df\_list=[]  
 for item in show\_discharges(df, list\_stbs):  
 df\_list.append(item)*

# Изменения

Моя модель ошибки на обучающих данных какая? (может нужна регуляризация?)

Создаем матрицу схожести (по расстояниям) и смотрим у которой сильное отклонение от схожих (может выброс?), а также по совету Соса по сверке оценки при прогнозе

page\_param вынести из shop в avito\_product\_parser

открываем в view свою таблицу

несущественные параметры модели через коэффициенты исключить

убираем категориальные переменные и смотрим точность модели

цена параметр по очереди визуализируем

отфильтровать выбросы, по стб

своб планировка и студия могут ухудшать показатель rooms\_num

# требования к проге

скачивает с сайта

отображение заданных столбцов (если не вписаны, то смотрим всю)

тесты чистоты - выводит столбцы на соответствие регулярному выражению (ввод имя стб, рег выражение), показывает какие данные и сколько в категориальных стб (ввод имя стб),

выбросы (допустим в заданную квантиль не попадают (0.9, 0.01), больше или меньше опред. значения, отклонены от среднего больше трех стандартных отклонений).

изменения (в виджете прям вносятся)

удаление строк по номерам и выделением в виджете -

корректировка значений

# Усовершенствования

Предсказать стоимость на основе ансамбля регрессоров (tree, kneighbours)

Csv reader (p 180 data mining) -- объедиение файлов с квартирами

Геокодер гугла для scraper (p 180 datamining)

Если была включена запись до какого-то момента в прошлом и произошло какое-либо исключение (в том числе нажатие кнопки стоп) в первый раз слив со старой базой будет корректен, так как новые записи присоединятся сверху, но потом, будут докачиваться уже имеющиеся и промежуточные…

Для объявления в юле ищем нужный url, например, для https://www.youla.ru/vladikavkaz/nedvijimost/prodaja-kvartiri?

https://www.youla.ru/web-api/products?cId=20&city=576d0618d53f3d80945f96c1&page=1&scId=2001&serpId=36e8c70f3b29c0

Может понадобиться изменить еще и url базу для продукта https://api.youla.io/api/v1/product/

перепроверить соимость квартир в проге в зависимости от стоимости кв метра

тесты - много значений, хотя ожидалось меньше x, соответствие паттернам для заданных стб, вырезка выбросов для заданных стб

посчитать удаленность для метро от центра в км (через геокодер яндекса <https://geocode-maps.yandex.ru/1.x/?geocode=метро%20Третьяковская&results=100&format=json>)

# Описание технической реализации скрапера

Абстрактный класс ShopProductsParser – предназначен для создания шаблона сбора информации о товарах на торговых площадках (авито, ебэй, юла). Предполагается, что те предоставляют информацию по следующему шаблону – перечень товаров на текущей странице и ссылка на очередную страницу поиска.

Техника скрапинга будет следующей – мы задаем начальную страницу, итерируемый параметр в get запросе, 4 словаря, описывающих теги со ссылками на товары и другие страницы (по два и туда и туда – один общий тег контейнер, а другой описывающий конкретный элемент),

Из этого в ShopProductsParser задается функция (**getListFrom2Tags**), которая возвращает bs4 объект (списки тегов) по одной из двух вышеуказанных пар контейнеров.

также задается функция инициализации базовых переменных **initClassVariables**

Основную работу по перебору всех страниц и загрузке информации о каждом товаре (абстрактная функция **getProductItems**) проделывает функция **startListProductsParser**. При этом итерируемым параметром является cur\_url, который возвращается тоже абстрактной функцией **getNextURL**.

Обе функции будут реализованы в классах наследниках (авито, ебэй и юла), так как их реализация зависит от конкретного сайта

Функция **parseUrl** разбивает первоначальный url на части из базовой, довеска и словаря параметров (предполагается что домен верхнего уровня (например, com, ru) содержит не более 4 символов)

Обратную операцию сбора в строку url из строк базовой, довеска и словаря параметров выполняет **makeUrlFromParts**

В поле target класса thread указываем название функции без скобок, если есть параметры, то их в args Иначе асинхронность не работает!!!!!!!!

у нас останов обновления, если встретилось 10 записей с датой раньше вчера от последней даты обсновления (сохраняется в пикл в main\_window). По признакам можно оптимизировать – резкий обрыв даты с предыдущей и последующей (более 15 часов),