МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

ОТЧЕТ

О РЕЗУЛЬТАТАХ ГРУППОВОЙ ТЕХНОЛОГИЧЕСКОЙ

(проектно-технологической) ПРАКТИКИ

ФГАОУ ВО «Тюменский государственный университет»

(наименование организации)

Кафедра программного обеспечения

(наименование структурного подразделения)

|  |  |  |
| --- | --- | --- |
| Выполнил  обучающийся 2 курса,  МОиАИС-20.012 группы | \_\_\_\_\_\_\_\_\_\_\_  (подпись) | Друганов Артем  \_\_\_Викторович\_\_\_  (ФИО полностью) |
| Выполнил  обучающийся 2 курса,  МОиАИС-20.02 группы | \_\_\_\_\_\_\_\_\_\_\_  (подпись) | Жиряков Валентин \_\_Владиславович\_\_  (ФИО полностью) |
| Выполнил  обучающийся 2 курса,  МОиАИС-20.01 группы | \_\_\_\_\_\_\_\_\_\_\_  (подпись) | Катанаев Дмитрий \_\_\_Михайлович\_\_\_  (ФИО полностью) |
| Руководитель практики от института  Доцент, к. ф.-м. н. | \_\_\_\_\_\_\_\_\_\_\_  (подпись) | Ступников Андрей \_\_\_Анатольевич\_\_\_  (ФИО полностью) |
| Научный руководитель проекта  Доцент, к.т.н. | \_\_\_\_\_\_\_\_\_\_\_  (подпись) | Донкова Ирина  \_\_\_Адольфовна\_\_\_  (ФИО полностью) |

Защищен \_\_\_ июня 2022 г.

Результаты экзамена / зачета \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (подпись руководителя практики от института)

2

# СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc74207227)

[Глава 1. Обзорная 5](#_Toc74207228)

[1.1 Предметная область 5](#_Toc74207229)

[1.2 Постановка задачи 5](#_Toc74207230)

[1.3 Технологии 6](#_Toc74207230)

[1.4 Математические подходы 6](#_Toc74207230)

[Глава 2. Разработческая 8](#_Toc74207228)

[2.1 Excel таблицы 8](#_Toc74207229)

[2.2 Python код 10](#_Toc74207230)

[2.3 WinForms интерфейс 15](#_Toc74207230)

[2.4 Структура данных 17](#_Toc74207230)

[Глава 3. Описание программного приложения “CodeAnalysis” и “Interface” 18](#_Toc74207228)

3[.1 CodeAnalysis (Python) 18](#_Toc74207229)

[3.2 Interface (C#) 21](#_Toc74207230)

[Глава 4. Описание командной работы 24](#_Toc74207228)

[ЗАКЛЮЧЕНИЕ 28](#_Toc74207241)

[СПИСОК ЛИТЕРАТУРЫ 29](#_Toc74207242)

[ПРИЛОЖЕНИЕ 31](#_Toc74207243)

|  |  |  |
| --- | --- | --- |
| Ответственный за соответствие содержания и оформления текста отчёта предъявляемым требованиям | (подпись) | Семенова Ольга Юрьевна |

3

# ВВЕДЕНИЕ

Тип практики: технологическая

Дата начала: 28.03.2022

Дата конца: 11.06.2022

Цель практики – получение первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности, обучающихся по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем (уровень бакалавриата).

Задачи учебной практики:

* Закрепление теоретических и практических знаний, умений, и навыков, полученных на младших курсах обучения;
* Выполнение заданий, предусмотренных программой практики и назначенных руководителем;
* Освоение отдельных компьютерных программ и информационных систем, используемых в профессиональной деятельности;
* Получение навыков работы с периодическими, реферативными и справочными информационными изданиями по прикладной математике и информационным системам;
* Подготовка к осознанному и углубленному изучению дисциплин старших курсов;
* Подготовка и защита в установленный срок отчета по практике.

В результате выполнения технологической (проектно-технологической) учебной практики студент должен обладать следующими компетенциями:

* Разработка интерфейса на базе Windows Forms
* Статистический анализ с помощью языка программирования Python
* Выполнение обработки большого объёма данных

4

* Парсинг информации с веб-сайтов

Описание проблемы - на данный момент, в интернете присутствует масса интернет-сервисов по поиску и продаже недвижимости, но нет сервиса, который бы объединял их все в одно целое. Из-за чего потенциальный покупатель может не знать о более благоприятных для него позициях на рынке. Сервис, объединяющий площадки поможет пользователю найти подходящий для него вариант и узнать предположительную стоимость недвижимости.

Выполненная цель работы – разработано приложение для оценки стоимости недвижимости и предложения пользователю более выгодных вариантов по заданным параметрам.

Решённые задачи:

* Изучены научные подходы для анализа данных, литературные и интернет-источники.
* Внедрены алгоритмы обработки данных сайтов Avito[6], Domofond[7], Cian[8].
* Внедрены алгоритмы анализа данных.
* Создано приложение, предоставляющее пользователю результаты анализа.

5

**Глава 1. Обзорная**

**1.1. Предметная область**

Сфера деятельности: Разработка desktop-приложений

Целевая аудитория: Люди желающие купить квартиру

Объекты: Сайты продажи недвижимости

Взаимосвязи: Сфера недвижимости.

Условия выполнения проекта: Наличие данных недвижимости в подходящем формате

**1.2. Постановка задачи**

Дано: 3 сайта по продаже недвижимости: Avito[6], Domofond[7], Cian[8].

Необходимо:

* Собрать с них данные в отдельные датафреймы, с помощью webpack;
* Выполнить предобработку данных для исключения аномальных и повторяющихся на разных сайтах объявлений;
* Изучить методы корреляционного анализа и выполнить расчет основных коэффициентов[2,5];
* Изучить методы регрессионного анализа и получить однофакторные и многофакторные регрессионные модели[1,3];
* Визуализировать данные в виде корреляционного поля и построить графики однофакторных линейных регрессионных моделей[1,2,3];
* Вывести данные в новый датафрейм с добавлением столбца величины отклонения регрессионной модели;
* Визуализация полученного датафрейма в приложении.

6

**1.3. Технологии**

NumPy - библиотека с открытым исходным кодом для языка программирования Python. Возможности:

поддержка многомерных массивов (включая матрицы);

поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами.

MatPlotLib - библиотека на языке программирования Python для визуализации данных двумерной графикой.

Sklearn - библиотека машинного обучения для языка программирования Python. Он включает в себя различные алгоритмы классификации, регрессии и кластеризации.

Scipy - библиотека для языка программирования Python с открытым исходным кодом, предназначенная для выполнения научных и инженерных расчётов.

Pandas - Библиотека на языке Python для обработки и анализа данных.

Microsoft.Office.Interop – Пространство имен в языке C# для работы с продуктами Microsoft Office.

**1.4. Математические подходы**

Регрессионный анализ: статистический аналитический метод, позволяющий вычислить предполагаемые отношения между зависимой переменной одной или несколькими независимыми переменными. Используя регрессионный анализ, вы можете моделировать отношения между выбранным переменными, а также прогнозируемыми значениями на основе модели.[4]

7

Корреляционный анализ: статистический метод изучения взаимосвязи между двумя и более случайными величинами. В качестве случайных величин в эмпирических исследованиях выступают значения переменных, измеряемые свойства исследуемых объектов наблюдения. Суть корреляционного анализа заключается в расчете коэффициентов корреляции. Коэффициенты корреляции могут принимать, как правило, положительные и отрицательные значения. Знак коэффициента корреляции позволяет интерпретировать направление связи, а абсолютное значение – силу связи.[2,5]

8

**Глава 2. Разработческая**

**2.1 Excel таблицы**

В ходе проведения корреляционного анализа[1,2,5] были получены следующие значения. Самыми главными ценообразующими факторами[1] являются площадь с коэффициентом корреляции 0,891489, и количество комнат с коэффициентом 0,634476. Факторы этаж и дата выставленного объявления с коэффициентами 0,124411 и -0,04281 соответственно, наоборот почти не оказывают влияния на итоговую стоимость квартиры. А также были получены корреляционные поля каждого из параметров (Рис. 1, Рис. 2, Рис. 3, Рис. 4)

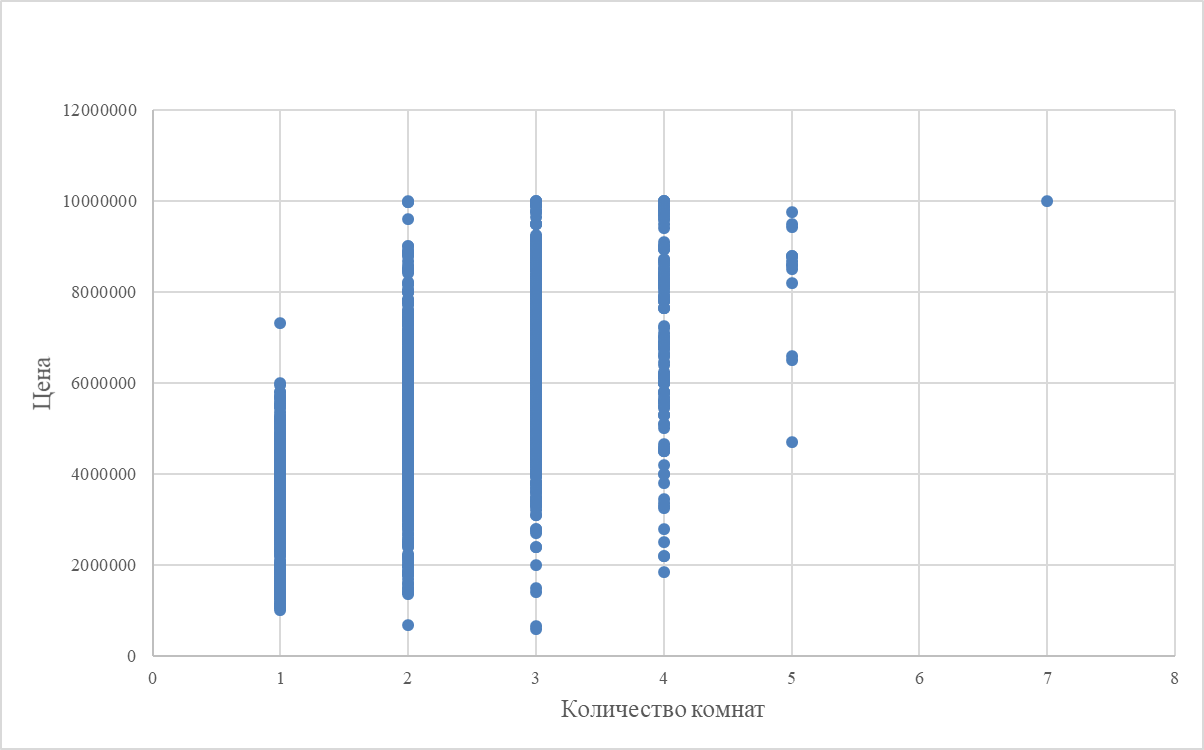


Рис. 1. Корреляционное поле, где факторная переменная количество комнат, а результирующий фактор - стоимость квартир

9

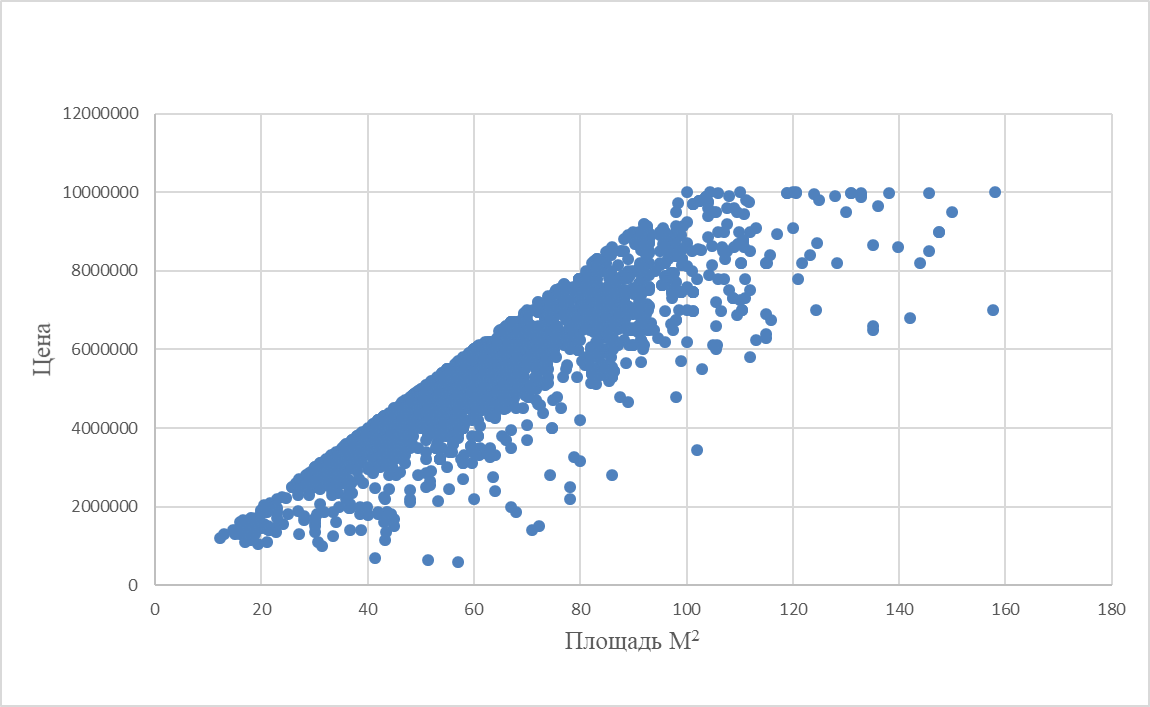


Рис. 2. Корреляционное поле, где факторная переменная - площадь, а результирующий фактор - стоимость квартир

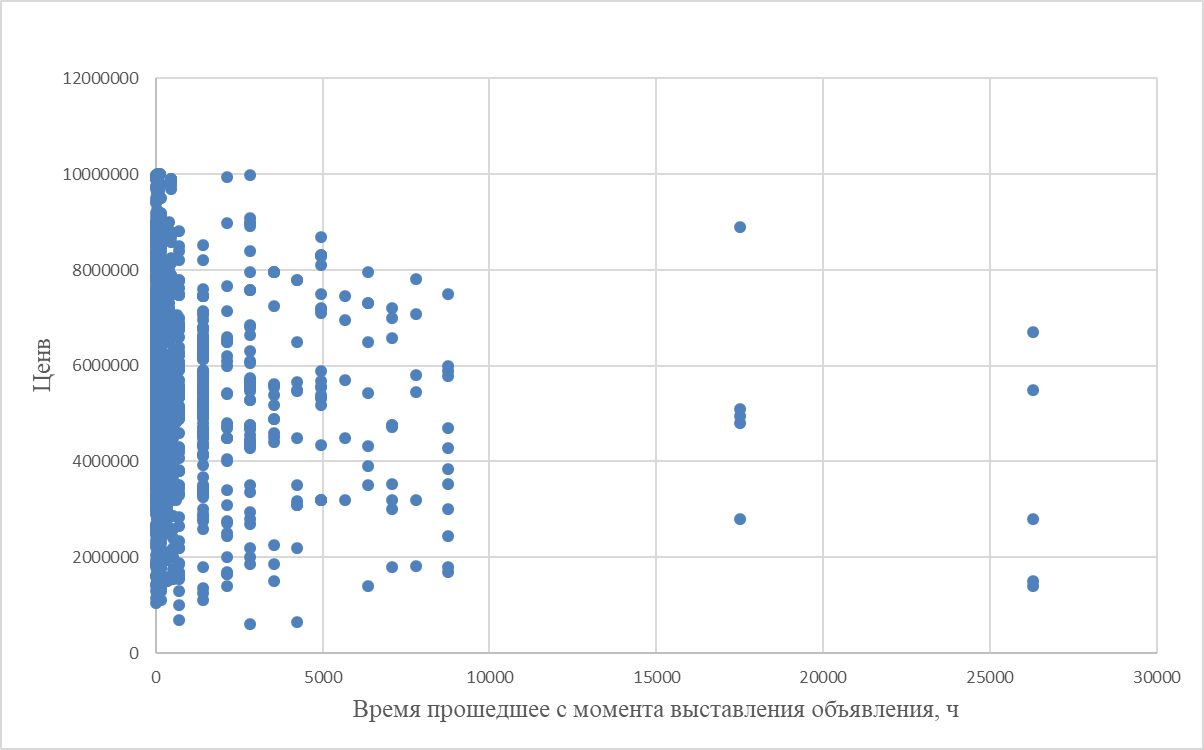


Рис. 3. Корреляционное поле, где факторная переменная - время прошедшее с момента выставления квартир на продажу, а результирующий фактор - стоимость квартир

10

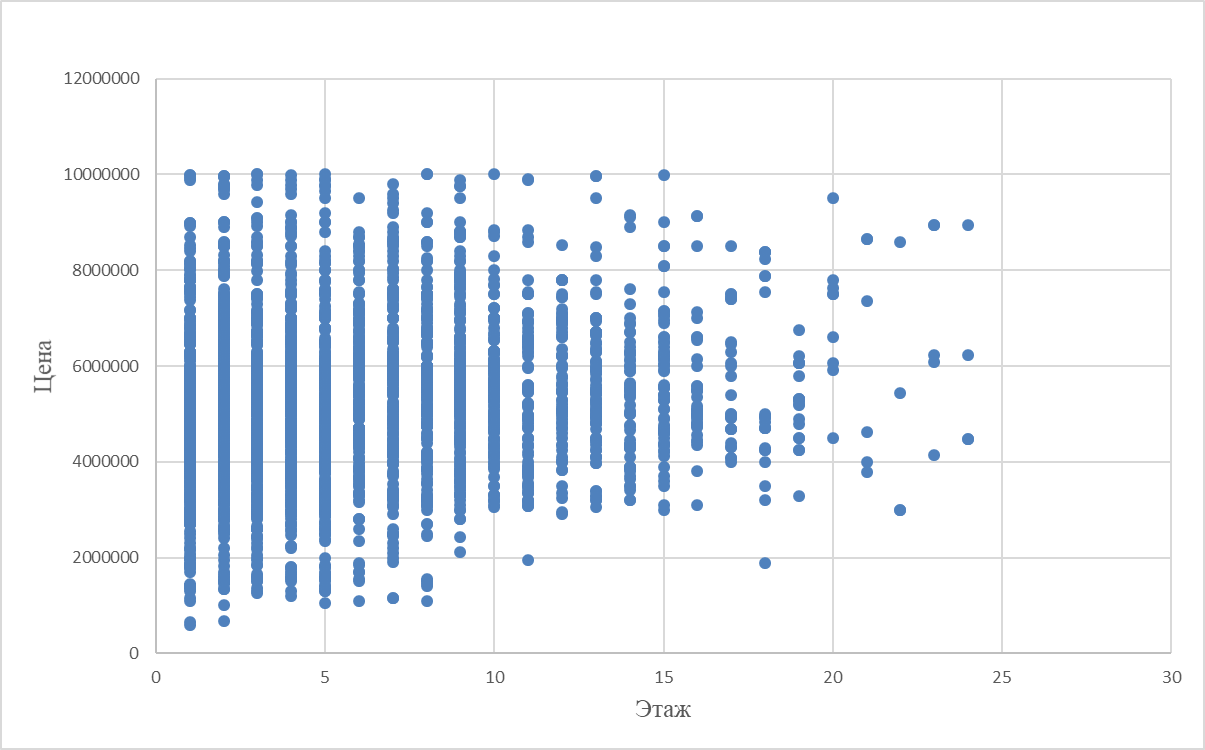


Рис. 4. Корреляционное поле, где факторная переменная - этаж, а результирующий фактор - стоимость квартир

**2.2 Python код**

Технологии: Sklearn, NumPy, Scipy, Pandas, MatPlotLib

Алгоритмы:

* sklearn.linear\_model.LinearRegression[9] - Линейная регрессия соответствует линейной модели с коэффициентами w = (w1, ..., wp), чтобы минимизировать остаточную сумму квадратов между наблюдаемыми целями в наборе данных и целями, прогнозируемые линейным приближением.
* sklearn.linear\_model.SGDRegressor[9] - SGD расшифровывается как стохастический градиентный спуск: градиент потерь оценивается для каждой выборки за раз, и модель обновляется по пути с уменьшающимся графиком скорости обучения.

11

Регуляризатор — это штраф, добавляемый к функции потерь, который уменьшает параметры модели до нулевого вектора, используя либо квадрат евклидовой нормы L2, либо абсолютную норму L1, либо комбинацию того и другого (эластичная сетка). Если обновление параметра пересекает значение 0.0 из-за регуляризатора, обновление усекается до 0.0, чтобы обеспечить возможность изучения разреженных моделей и оперативного выбора объектов.

* sklearn.metrics.mean\_squared\_error[9] – Представляет из себя среднеквадратичную ошибку, на основе которой можно оценить качество работы программы
* sklearn.preprocessing.StandardScaler[9] - Стандартизируйте характеристики, удалив среднее значение и масштабировав до единичной дисперсии.  
  Стандартный балл выборки x рассчитывается как: z=(x - u)/с   
  где u - среднее значение обучающих выборок или ноль, если with\_mean=False, а s - стандартное отклонение обучающих выборок или единица, если with\_std=False.  
  Центрирование и масштабирование происходят независимо для каждого объекта путем вычисления соответствующей статистики по выборкам в обучающем наборе. Затем среднее значение и стандартное отклонение сохраняются для использования в последующих данных с помощью преобразования.

Основные переменные, используемые программой:

* List - Массив, состоящий из всех данных датафрейма
  + List[:][0] - Количество комнат
  + List[:][1] - Площадь квартиры
  + List[:][2] - Этаж
  + List[:][3] – Цена

12

* + List[:][4] – Улица
  + List[:][5] - Время, прошедшее с выставления объявления
  + List[:][6] - Сайт, на котором выставлено объявление
  + List[:][7] - Разница между предполагаемой ценой и указанной
* data\_x - Массив факторов
* data\_y - Массив цен
* Predicted - Массив прогнозируемых цен
* model.coef\_ - Массив с коэффициентами каждого из параметров
* avito\_dataframe = pd.DataFrame(pd.read\_excel()) – датафрейм собранный с сайта Avito
* cian\_dataframe = pd.DataFrame(pd.read\_excel()) – датафрейм собранный с сайта Циан
* domofond\_dataframe = pd.DataFrame(pd.read\_excel()) - датафрейм собранный с сайта Domofond.ru

Функциональные блоки программы:

* def add(dataframe, name) – Добавление элементов таблицы в массив для обработки
  + Проверка входных данных
    1. Наличие строки с количеством комнат в виде - <количество комнат>-комнатная квартира
    2. Цена за 1 кв.м. <500000Р
  + Замена string значений на int
  + Удаление дефектов значений
  + Добавление значений в массив

dataframe – датафрейм взятый из файла с помощью библиотеки Pandas  
name – название сайта, с которого была взята информация

* def changeDate(List) – Изменение ячейки даты с “час назад”, “месяц назад” на число равное количеству прошедших часов.

13

* + Удаление дефектов значений
  + Получение int значения из строки
  + Умножение числа на необходимое количество часов:  
     24 за день, 168 за неделю, 708 за месяц, 8760 за год

List – Массив данных

* informationProcessing(List) – Удаление аномальных значений, которые могут привести к плохой работоспособности кода и функции прогнозирования  
  List – Массив данных
* def custom\_key(house):

return int(house[3]) - ключ для сортировки (по цене)

* def custom\_key\_df(dataframe):

return int(dataframe[7]) - ключ для сортировки (по разности действительной цены и прогнозируемой)

* List.sort(key=custom\_key) - функция сортировки массива по цене
* Linear regression analysis - Проведение регрессионного анализа полученных данных

1. StandardScaler().fit(data\_x).transform(data\_x) - стандартизированние данных с сохранением последовательности.
2. LinearRegression().fit(data\_x, data\_y) - стандартизированние данных для линейной регрессии.
3. model.intercept\_ - Аддитивная случайная величина, которая позже будет использована для прогнозирования цен на недвижимость.
4. plt.scatter(np.array(data\_x)[:, 1], model.predict(np.array(data\_x))) - добавление на плоскость точек прогнозируемой цены для визуального анализа.
5. plt.scatter(np.array(data\_x)[:, 1], data\_y[: Listlen]) - добавление на плоскость точек действительной цены для визуального анализа.

14

1. linear\_regression\_model = SGDRegressor(tol=.0001, eta0=.001) - добавление модели линейной регрессии для обучения модели на этапе прохождения по массиву.
2. linear\_regression\_model.fit(scaled\_df, data\_y) - прохождение метода стандартизации по массиву.
3. mse = mean\_squared\_error(data\_y, predictions) - вычисление средней квадратической ошибки.
4. RMSE = (np.sqrt(mse)) - вычисление среднеквадратической ошибка.
5. print(model.coef\_) - вывод коэффициентов каждого из параметров, которые используются для прогнозирования цены.

* Сортировка и обработка массива значений:

1. for i in range(Listlen):

sum += abs(Predicted[i] - data\_y[i])

List[i][7] = (data\_y[i]-Predicted[i])[0] - изменение значения 7-й ячейки на разность между реальной ценой и прогнозированной.

1. List.sort(key = custom\_key\_df) - сортировка списка по разности между реальной ценой и прогнозированной.

* Сохранение итогового датафрейма:

1. df = pd.DataFrame(List) - создание датафрейма из массива значений
2. df.to\_excel("EndDataframe.xlsx",encoding='cp1251') - сохранение датафрейма в формате .xlsx (Microsoft Excel).

* Сохранение датафрейма с коэффициентами и АСВ для прогнозирования цены:

1. List\_coef\_ = [[model.coef\_[0],"Кол-во комнат"],[model.coef\_[1],"Площадь"],[model.coef\_[2],"Этаж"],[model.coef\_[3],"Сколько прошло с выставления объявления (в часах)"],[model.intercept\_,"АСВ"]] - созданием массива с коэффициентами и АСВ.

15

1. df\_coef\_ = pd.DataFrame(List\_coef\_) - создание датафрейма из массива с коэффициентами и АСВ.
2. df\_coef\_.to\_excel("CoefDataframe.xlsx",encoding='cp1251') - сохранений датафрейма в формате .xlsx (Microsoft Excel).

**2.3. WinForms интерфейс**

Технологии: Microsoft.Office.Interop[]

Основные переменные, используемые программой:

* Filename - название файла датафрейма
* rCnt - положение относительно строк
* CCnt - положение относительно колонны
* kolvo - необходимое количество выводимых объявлений
* ExcelRange - датафрейм значений
* ExcelRange.Cells - ячейка датафрейма
* coefs - массив коэффициентов регрессионной модели
* ARV - Аддитивная случайная величина

Функциональные блоки программы:

* Работа с таблицей информации о недвижимости:
  1. dataGridView1.Rows.Clear() - удаление старых значений в таблице вывода
  2. dataGridView1.Refresh() - Обновление изображения вывода в таблице
  3. OpenExcel - Открыть встроенный обозреватель имён для работы с данными из Excel таблиц
     + Microsoft.Office.Interop.Excel.Application ExcelApp = new Microsoft.Office.Interop.Excel.Application();  
       Microsoft.Office.Interop.Excel.\_Workbook ExcelWorkBook;

16

Microsoft.Office.Interop.Excel.Worksheet ExcelWorkSheet;

Microsoft.Office.Interop.Excel.Range ExcelRange;

* 1. ExcelWorkBook = ExcelApp.Workbooks.Open(filename, 0, true, 5, "", "", true, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows, "\t", false, false, 0, true, 1, 0) - открывает выбранный файл как таблицу Excel
  2. ExcelWorkSheet = (Microsoft.Office.Interop.  
      Excel.Worksheet)ExcelWorkBook.  
      Worksheets.get\_Item(1) - получает значения Excel таблицы
  3. ExcelRange = ExcelWorkSheet.UsedRange - получает используемый диапозон указанной таблицы.
  4. searchData - производит поиск значений подходящих значениям указанным в фильтре, после чего происходит добавление значений в итоговый список
     + Проверка на количество комнат
     + Проверка на совпадение с выбранным этажом
     + Сравнение с площадью
     + Сравнение с указанным промежутком цен
     + Проверка совпадения указанного сайта
* Работа с таблицей коэффициентов:
  1. OpenExcel - Открыть встроенный обозреватель имён для работы с данными из Excel таблиц
     + Microsoft.Office.Interop.Excel.Application ExcelApp = new Microsoft.Office.Interop.Excel.Application();  
       Microsoft.Office.Interop.Excel.\_Workbook ExcelWorkBook;  
       Microsoft.Office.Interop.Excel.Worksheet ExcelWorkSheet;  
       Microsoft.Office.Interop.Excel.Range ExcelRange;

17

* 1. coefs.Add(Convert.ToDouble  
     (Convert.ToString(ExcelRange1.Cells[RCnt + 1, 2].Value))) - конвертация значений таблицы коэффициентов в локальные переменные
  2. predict - прогнозирование цены на основе полученных из таблицы коэффициентов данных:
     + Проверка ввёл ли пользователь все необходимые значения
       - Умножение значений на их коэффициенты и прибавление аддитивной случайной величины.

**2.4. Структура данных**

Входные данные представлены в формате Excel таблиц, имеющие 4 столбца:

1. Name - заголовок объявления, имеющий внутри себя:

Количество комнат, площадь, этаж

1. Price - Цена
2. Address - Адрес объявления
3. Date - Дата выставления объявления

На выходе программа выводит 2 .xlsx файла

1. Датафрейм значений после обработки и анализа, имеющий:  
   Количество комнат, площадь, этаж, цену, адрес, дату выставления, сайт, разница между прогнозируемой ценой настоящей.
2. Датафрейм регрессионных коэффициентов каждого из значений, имеющий:   
   Количество комнат, площадь, этаж, дату.

18

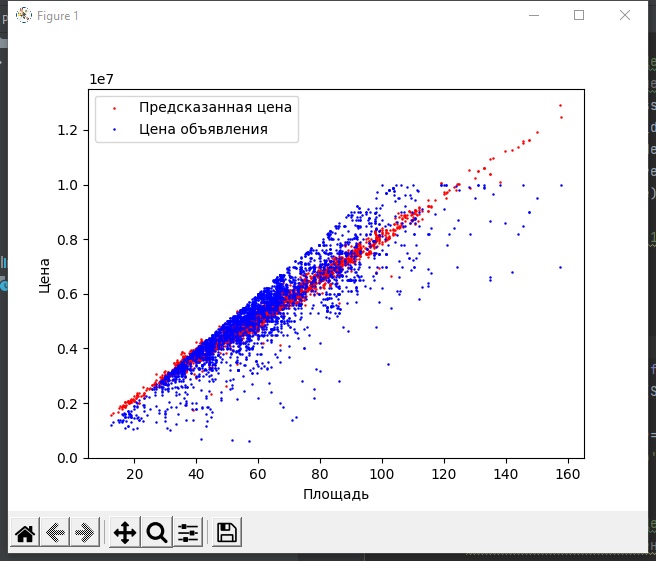
**Глава 3. Описание программного приложения “CodeAnalysis” и “Interface”**

**3.1.** **CodeAnalysis (Python)**

В результате выполнения работы кода получены 2 программы Python и Windows Forms.

После выполнения Python программы, matplotlib отрисовывает и выводит 2 графика (Рис. 5, Рис. 6):

1. Отношение цены к площади с реальной ценой и предположительной, где синие точки — это настоящая цена, а красные точки – предположительная.(Рис. 5)

  
Рис. 5. Корреляционное поле, где факторная переменная площадь, а результирующий фактор – стоимость квартир

1. Отношение цены к номеру объявления (где номера объявлений отсортированы по цене), где красными точками указана прогнозируемая цена, а синими – действительная (Рис. 6).

19

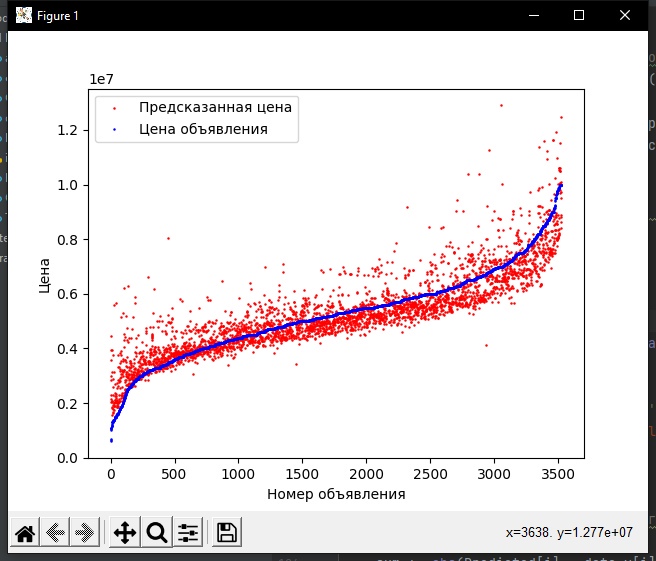


Рис. 6. Корреляционное поле, где факторная переменная – номера объявлений, отсортированных по цене, а результирующий фактор – стоимость квартир.

Кроме того, в итоге работы программы получены 2 датафрейма. В первом датафрейме (Рис. 7) объединены все 3 начальные датафрейма, полученные через парсинг сайтов, отфильтрованные от аномальных значений и повторов, с добавлением столбца отклонений от предположительной цены недвижимости. Таблица отсортирована по столбцу 7 (отклонение действительной цены от предположительной (модельной)).

20

Изображение выглядит как стол

Автоматически созданное описаниеРис. 7. Таблица, содержащая выходные данные

Во втором датафрейме (Рис. 8) получены коэффициенты признаков функции прогнозирования и аддитивную случайную величину[1]. Простыми словами предположительная стоимость единицы каждого параметра.

21

Изображение выглядит как стол

Автоматически созданное описаниеРис. 8. Таблица стоимости единицы каждого параметра

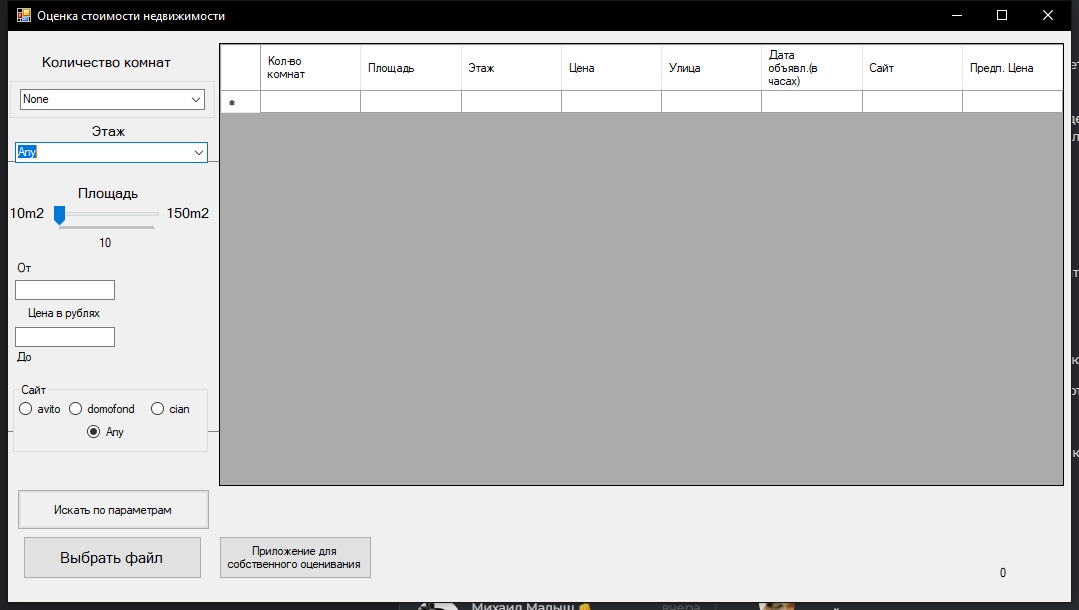
В процессе прогнозирования цены используются данные коэффициенты регрессионной модели. Как видно из модели (Рис.8.) получаем следующие суждения о параметрах:

1. При увеличении количества комнат цена квартиры уменьшается в среднем на 187997 рублей.
2. При увеличении площади квартиры на 1 м2 цена увеличивается в среднем на 78904 рубля.
3. При увеличении значения этажа стоимость увеличивается в среднем на 28105 рублей.
4. С течением времени за час стоимость уменьшается в среднем на 61 рубль.

**3.2. Interface (C#)**

Windows Forms приложение (Рис.9., Рис.10.), по задаваемым параметрам выдает список квартир в виде таблицы, где сверху – самые недооцененные варианты.

22

 Рис. 9. Интерфейс программы “Оценка стоимости недвижимости” до начала работы.

Изображение выглядит как стол

Автоматически созданное описаниеРис. 10. Интерфейс программы “Оценка стоимости недвижимости” после указания числовых данных и выполнения.

23

Пользователь может задать количество комнат, этажи, минимальную площадь, диапазон цен и сайт, с которого он хочет видеть объявления. После этого необходимо нажать кнопку “Выбрать файл” и открыть файл с датафреймом полученным после выполнения Python кода, нажать на кнопку “Искать по параметрам”, и выждать загрузку, после чего в сером окошке будут появляться строки с данными квартир и их предположительной стоимостью.

Из дополнительного функционала, по нажатии на кнопку “Приложение для собственного оценивания”, перед пользователем появится окно (Рис.11.) с тремя полями для ввода: количество комнат, площадь, этаж. При заполнении всех трёх, в четвертом поле снизу окна будет выведена предположительная цена квартиры с указанными параметрами.

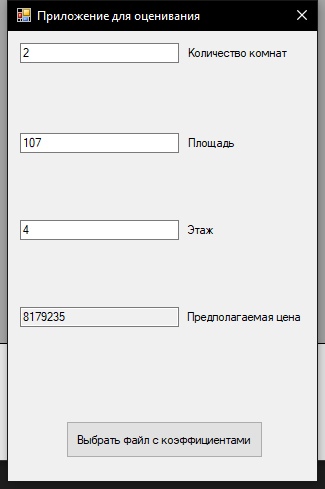


Рис. 11. Интерфейс программы “Приложение для оценивания”

24

**Глава 4.** **Описание командной работы**

Работа команды была организована дистанционно с использованием приложения для голосового, видео- и текстового общения – Discord и Microsoft Teams. Работа команды была исполнена в полном объёме, все задачи и цели были выполнены в режиме открытой коммуникации (т.е. любой из участников мог помочь другому для более эффективного решения проблемы, а также любые задачи и цели были обговорены командой).

Таблица 1

Данные студента Друганова Артема Викторовича

|  |  |  |
| --- | --- | --- |
| ФИО | Друганов Артем Викторович |  |
| Роль | Проектный руководитель |
| Интересы в сфере разработки ПО | Математическое моделирование информации.  Работа с парсерами и информацией сайтов.  Разработка пользовательского интерфейса. |

В рамках проекта участнику достались задачи:

1. Изучение научных подходов для анализа данных.
2. Внедрение методов корреляционного и регрессионного анализа.
3. Изображение выходных данных для визуального анализа.
4. Вывод исходных данных.
5. Обработка фильтров.
6. Привязка данных из Excel, для работы внутри программы.
7. Оптимизация работы кода.

По окончанию работы над проектом, все поставленные задачи были выполнены.

25

Таблица 2

Данные студента Катанаева Дмитрия Михайловича

|  |  |  |
| --- | --- | --- |
| ФИО | Катанаев Дмитрий Михайлович |  |
| Роль | Программист |
| Интересы в сфере разработки ПО | Frontend  Backend  Работа с нейросетями. |

В рамках проекта участнику достались задачи:

1. Изучение научных подходов для анализа данных;
2. Обработка исходных и входных данных;
3. Внедрение регрессионного анализа;
4. Разработка пользовательского интерфейса;
5. Связь программ, созданных на разных платформах.

По окончанию работы над проектом, все поставленные задачи были выполнены.

Таблица 3

Данные студента Жирякова Валентина Станиславовича

|  |  |  |
| --- | --- | --- |
| ФИО | Жиряков Валентин Станиславович |  |
| Роль | Разработчика интерфейса |
| Интересы в сфере разработки ПО | Frontend  Разработка пользовательского интерфейса. |

26

В рамках проекта участнику достались задачи:

1. Разработка пользовательского интерфейса;
2. Внедрение методов корреляционного анализа
3. Обработка входных данных

По окончанию работы над проектом, все поставленные задачи были выполнены.

Календарный план работы:

1. Изучение статистических методов анализа данных

Дата: 28.03.2022 - 06.04.2022

Итог: Были изучены необходимые методы статистического анализа.

1. Парсинг необходимой информации с сайтов

Дата: 06.04.2022

Итог: Собраны датафреймы с трех интернет-площадок (Avito, Cian, Domofond) при помощи веб-инструмента Web Scraper.

1. Разработка первоначального кода Python

Дата: 08.04.2022 - 13.04.2022

Итог: Написан обработчик исходных данных, внедрена первая версия регрессионного анализа, получен первый выходной файл.

1. Визуальный анализ статистической модели и оптимизация кода

Дата: 11.04.2022 - 01.05.2022

Итог: Связав графики корреляционного и регрессионного анализов, была замечена связь каждого из параметров. Была ускорена работа программы (Исключение лишних проходов по циклу массива)

1. Разработка интерфейса программы

Дата: 11.05.2022 - 01.06.2022

27

Итог: Создан интерфейс программы на Windows Forms, добавлена возможность добавления .xlsx файлов и их отрисовки на поле, а также фильтр для отсеивания неподходящих данных.

1. Введение дополнительного функционала

Дата: 01.06.2022 - 04.06.2022

Итог: В приложение была введена функция предсказания цены на недвижимость по её основным параметрам.

Итоги командной работы с куратором:

По итогам командной работы с куратором были выявлены множественные недочёты в программе, которые вследствие были исправлены, оказана помощь в написании статьи, предложены идеи для решения трудностей и способствование в оптимизации работы приложения.

Итоги встреч с руководителем:

Проведено тестирование работы программного приложения, исправлены недочёты в оформлении отчета.

Возникшие сложности в работе над проектом:

Трудоемкость выполнения работы, работа с координатами, инициализация Python кода внутри интерфейса.

Оценка роли каждого из участников:

1. Друганов Артем Викторович – руководитель работы над проектом:

Оценка качества работы: выполнил 90% из возложенных задач

1. Жиряков Валентин Станиславович – разработчик пользовательского интерфейса:

Оценка качества работы: выполнил 75% из возложенных задач

1. Катанаев Дмитрий Михайлович - программист:

Оценка качества работы: выполнил 85% из возложенных задач

28

**Заключение**

По итогу выполненной работы, командой был разработан код для статистического анализа собранных данных о недвижимости и было разработано приложение для вывода пользователю лучших вариантов из предложенных квартир. Приложение было выполнено на 80% от возможного наилучшего варианта, в связи отсутствия некоторого функционала (Указан в перспективах разработки).

Во время разработки были выполнены задачи по извлечению информации с сайтов, выбору оптимального метода анализа данных, внедрению данного метода в код приложения, а также разработка интерфейса для пользователя.

В процессе работы командой были получены навыки работы с парсером, обработкой реальной информации, а также закреплены уже полученные знания выполнения регрессионного и корреляционного анализа, а также работы с библиотеками Python.

В перспективах дальнейшей работы над проектом можно перечислить:

* Использование нейронных сетей для улучшения качества прогнозирования цены – т. е. увеличение точности каждого из коэффициентов
* Увеличение функционала приложения
  + Автоматический парсинг необходимой информации и его анализ по нажатию
  + Увеличение количества параметров
  + Поиск похожих квартир при вводе ссылки
* Увеличение количества входных данных

29

**Список литературы**

1. А.Б. Горобцова. Оценка рыночной стоимости квартир с помощью методов регрессионного анализа / А. Б. Горобцова. - Текст : непосредственный // Моделирование и анализ данных. / А. Б. Горобцова. - №2. - Москва, 2019. - С. 63–72.
2. Гмурман. В.Е. Элементы теории корреляции. Функциональная, статистическая и корреляционная зависимость / Гмурман В.Е. - Текст : непосредственный // Теория вероятностей и математическая статистика. / Гмурман. В. Е. - 7-е изд., стереотипное - Высш. шк., Москва, 2003. - Гл. 18. - С. 253.
3. Баринов Н.П. Оценка рыночной стоимости земельного участка методом многомерного регрессионного анализа / Баринов Н. П. - Текст : непосредственный // Информационно-аналитический бюллетень рынка недвижимости RWAY №232 (июль 2014), №236 (ноябрь 2014): Поволжская научно-практическая конференция - 2014.
4. О требованиях к количеству сопоставимых объектов при оценке недвижимости сравнительным подходом : монография / Анисимова И.Н., Баринов Н.П., Грибовский С.В. [и др.] ; Санкт Петербург : Университетская книга, 2003. - 1-е издание - С. 2–7., Текст : непосредственный.
5. Использование MS EXCEL для анализа статистических данных : учебное пособие / УрФУ ; сост. В. Р. Бараз, В. Ф. Пегашкин. - Нижний Тагил : УрФУ, 2014 – С. 68-123. - Текст : непосредственный.
6. Avito : Интернет-сервис для размещения объявлений продажи товаров и услуг : [сайт]. - URL : <https://www.avito.ru> (дата обращения: 06.04.2022). - Текст : электронный
7. Domofond : Интернет-сервис для размещения объявлений продажи недвижимости : [сайт]. - URL : <https://www.domofond.ru> (дата обращения: 06.04.2022). - Текст : электронный

30

1. Циан : Интернет-сервис для размещения объявлений продажи и аренды недвижимости : [сайт]. - URL : <https://www.cian.ru> (дата обращения: 06.04.2022). - Текст : электронный
2. Документация библиотеки sklearn. – Текст : электронный // Официальный веб-сайт библиотеки sklearn [Электронный ресурс]. URL: <https://scikit-learn.org/stable/user_guide.html> (дата обращения: 13.04.2022)
3. Документация языка программирования C#. – Текст : электронный // Официальный веб-сайт Microsoft [Электронный ресурс]. URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения: 11.05.2022)

31

**Приложение**

Презентация первого этапа работы над проектом

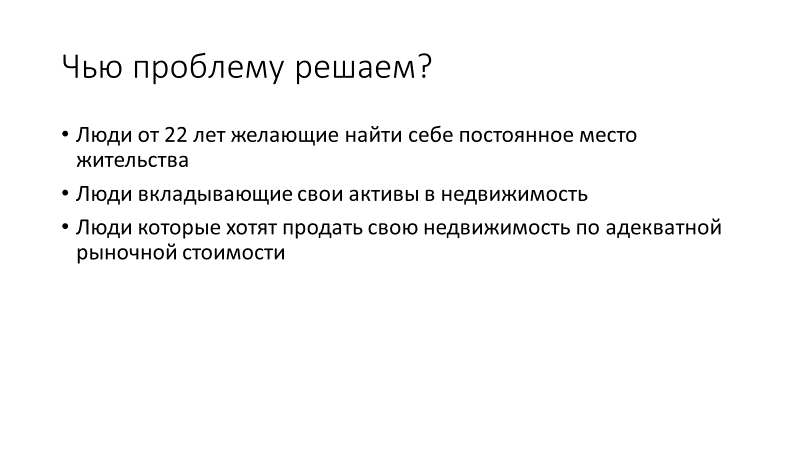
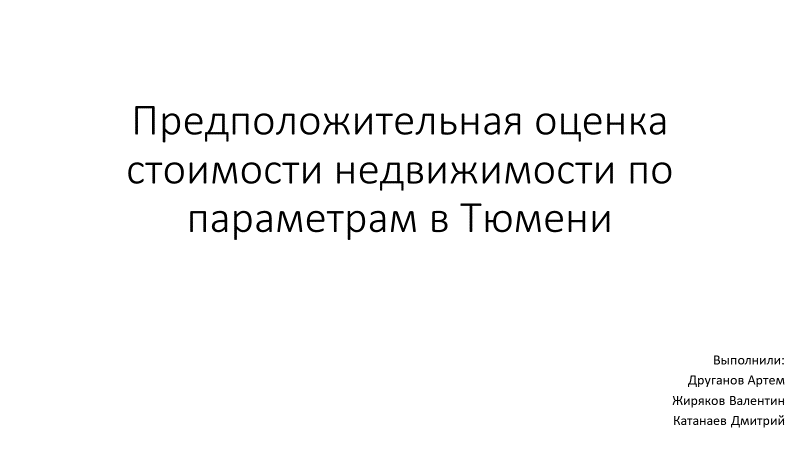


Рис. 12. 1 слайд Рис. 13. 2 слайд

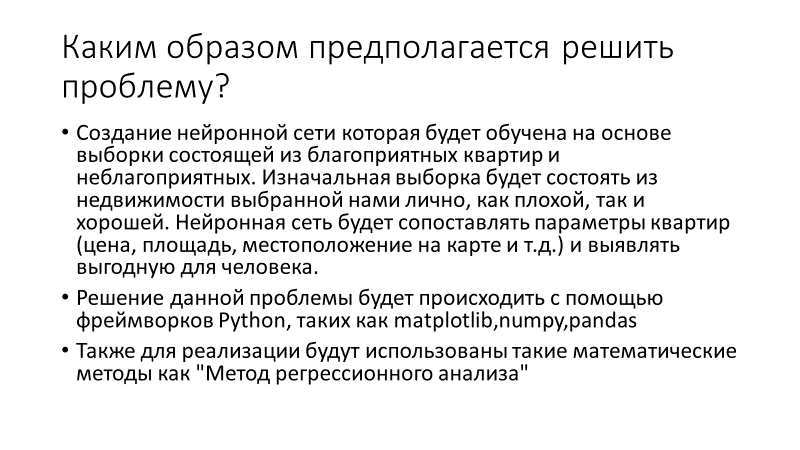
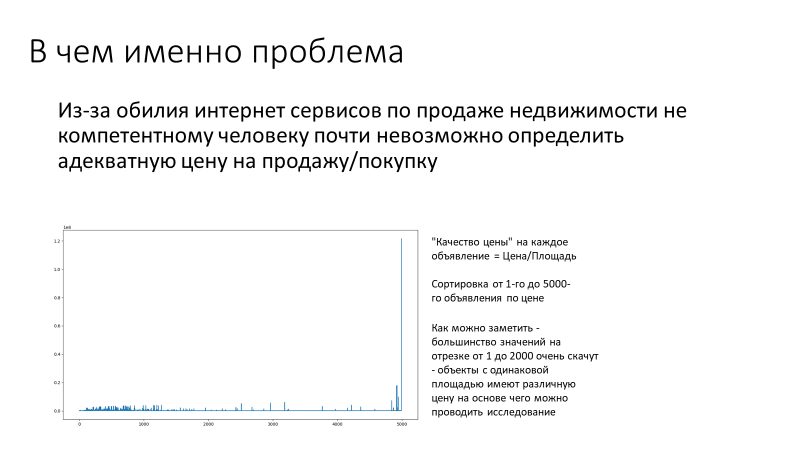


Рис. 14. 3 слайд Рис. 15. 4 слайд

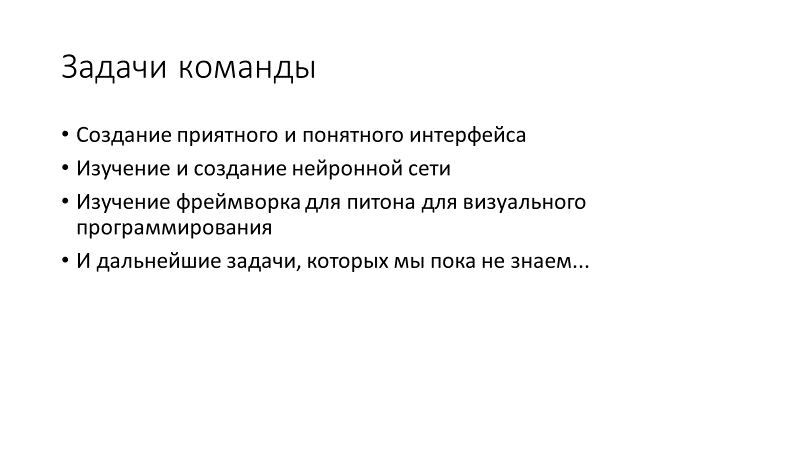
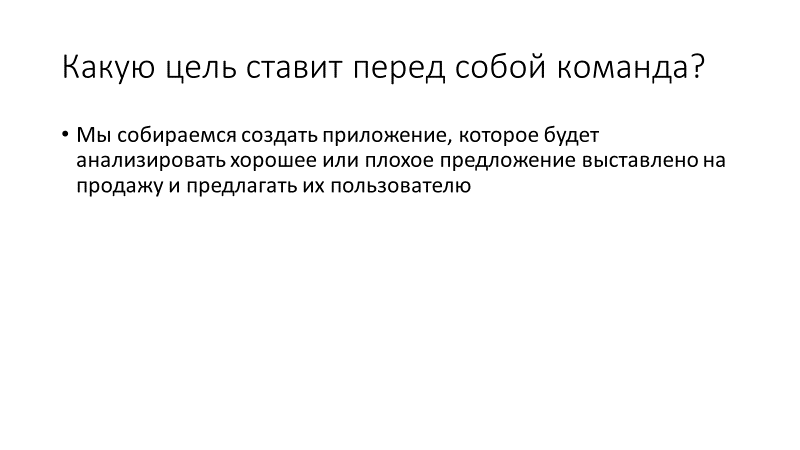
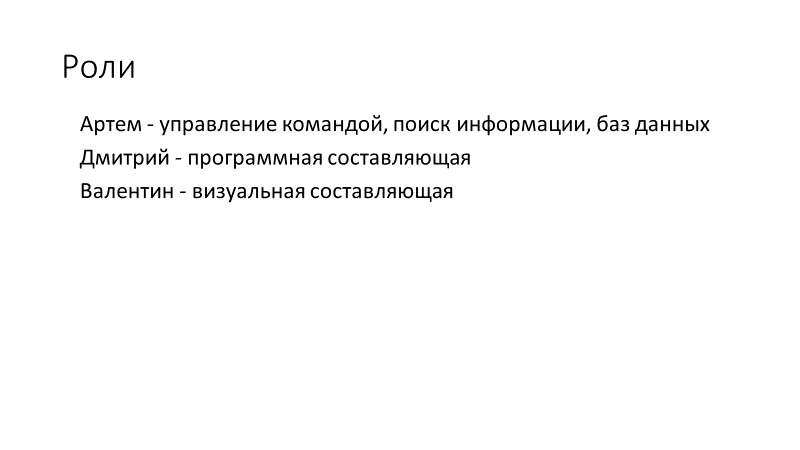
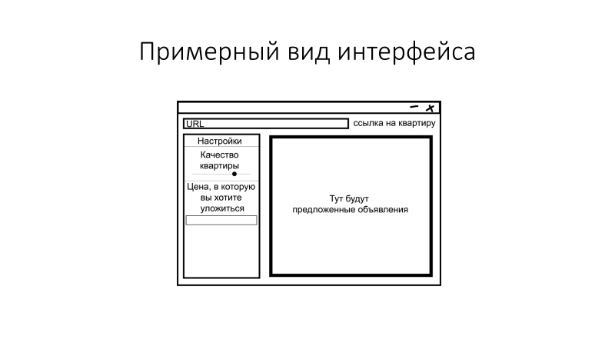


Рис. 16. 5 слайд Рис. 17. 6 слайд

 Рис. 18. 7 слайд Рис. 19. 8 слайд

32

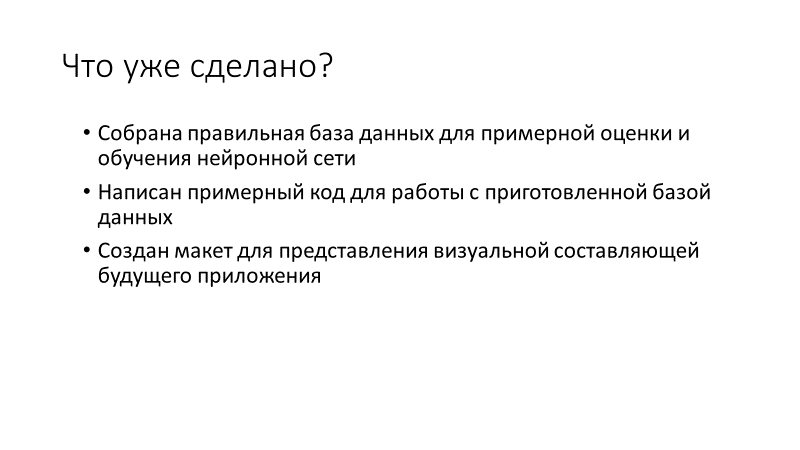
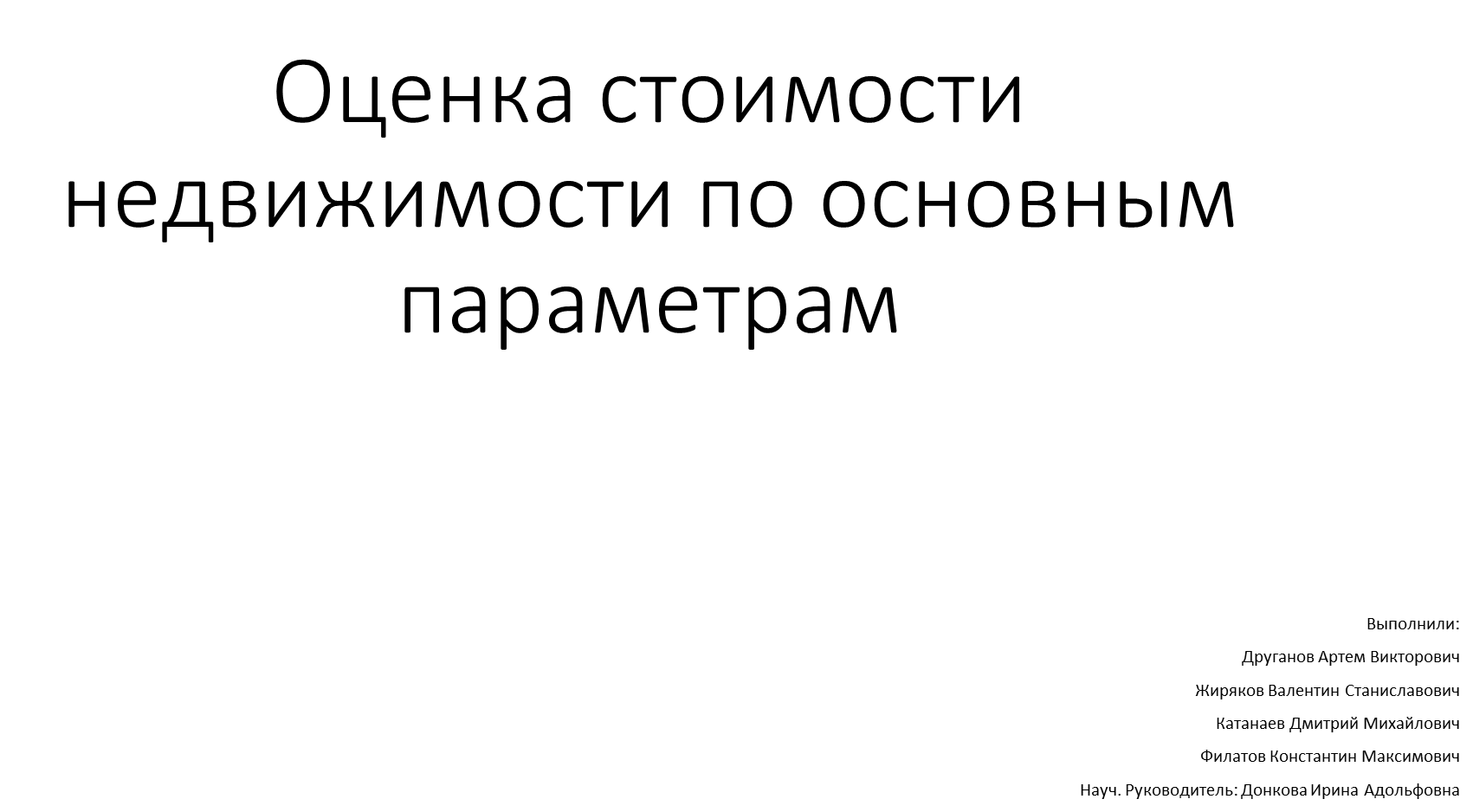
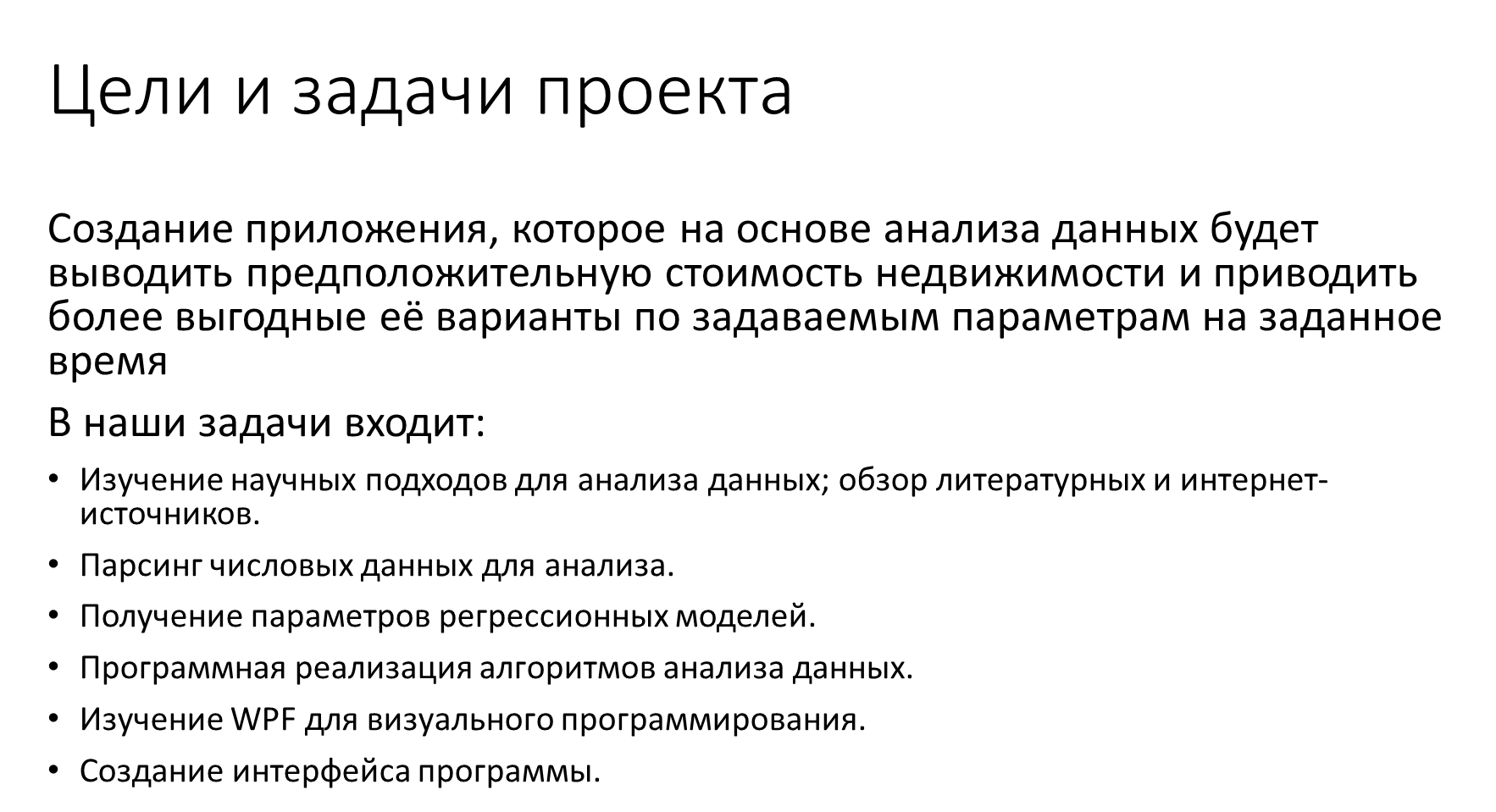


Рис. 20. 9 слайд

33

Презентация проекта на студенческой научной конференции

  
 Рис. 21. 1 слайд Рис. 22. 2 слайд

 Рис. 23. 3 слайд Рис. 24. 4 слайд

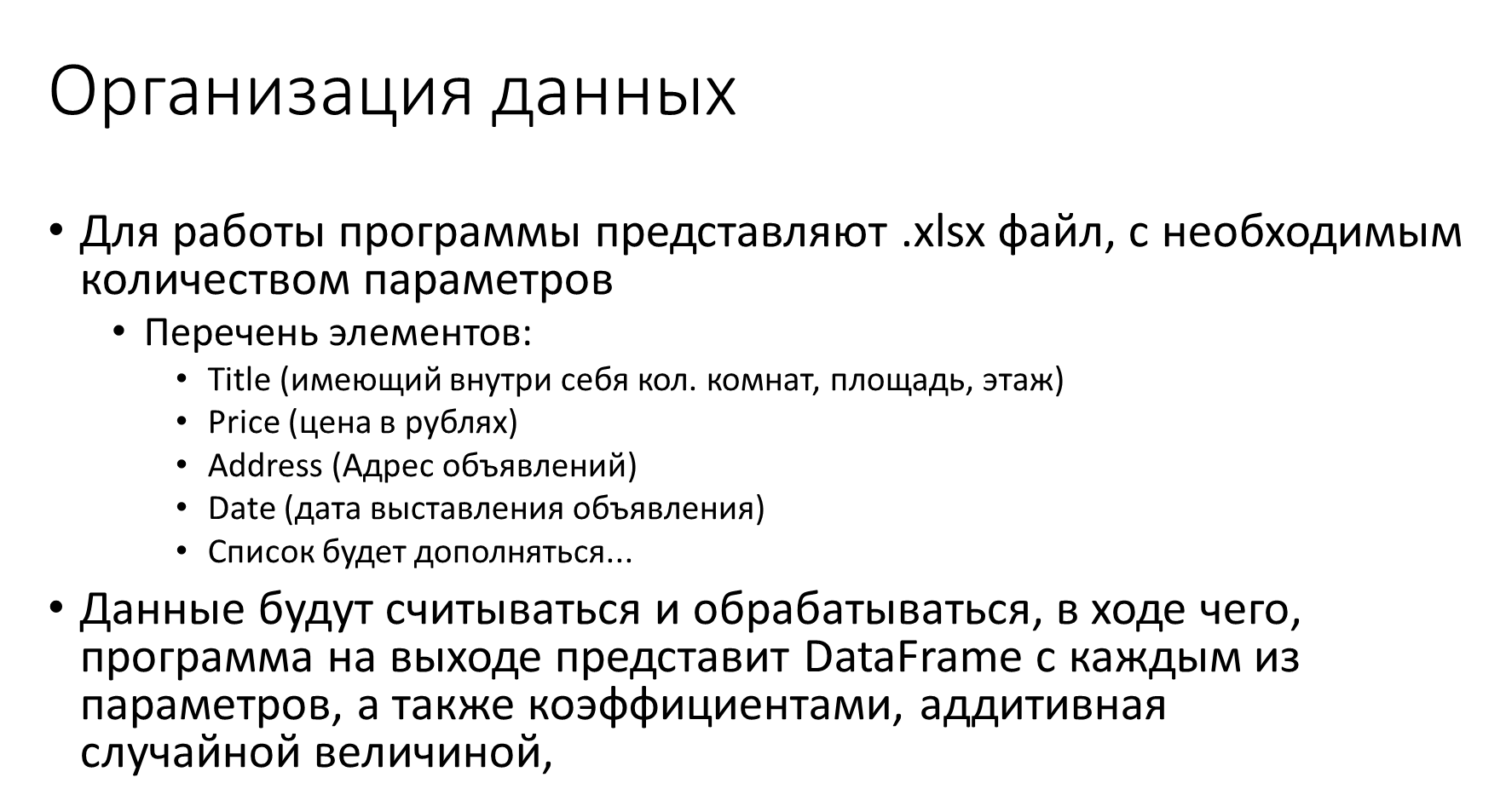
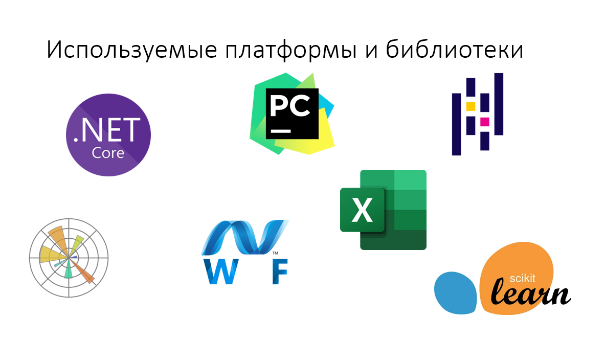
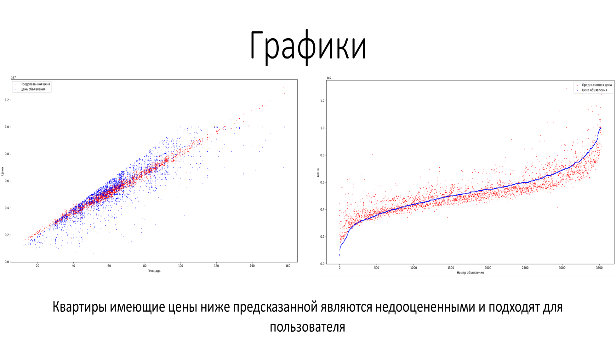
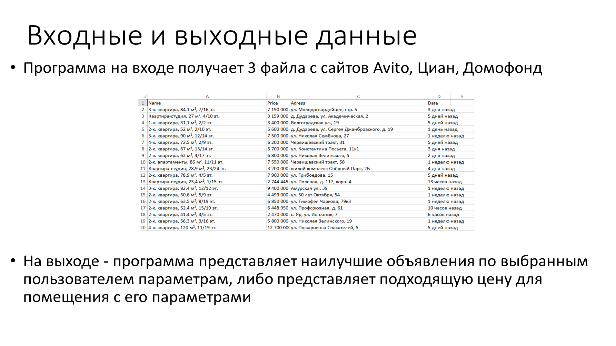
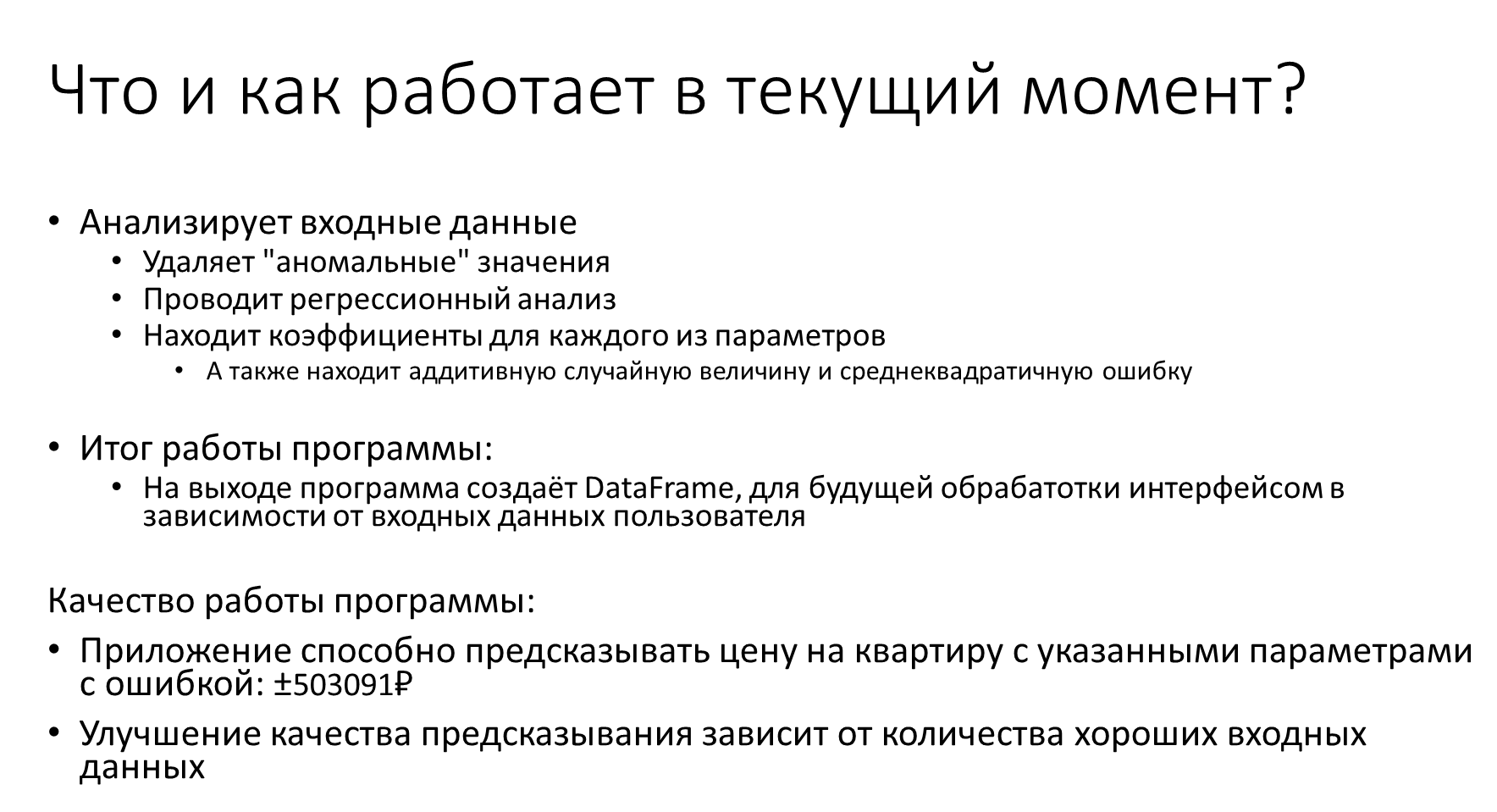
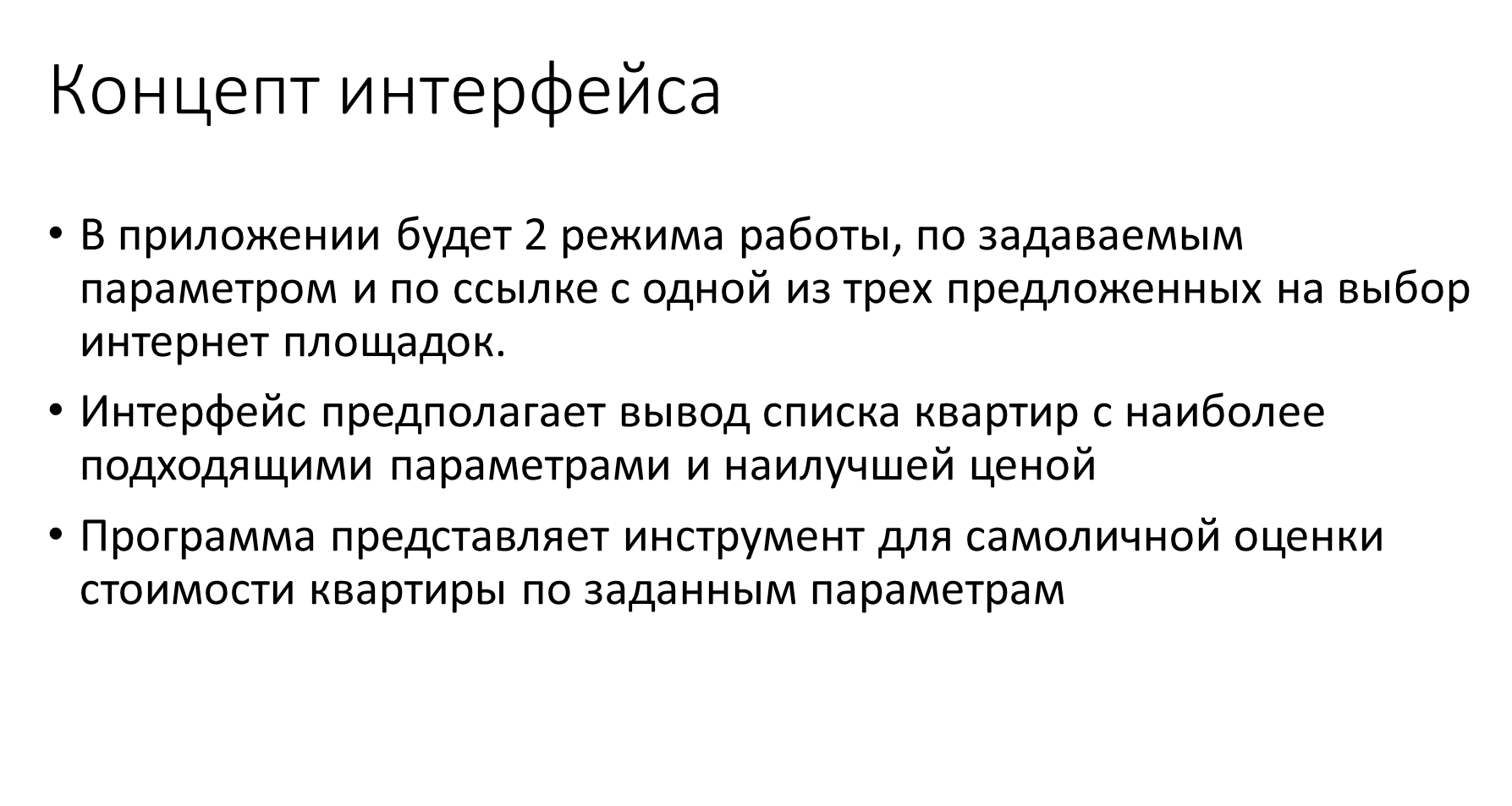
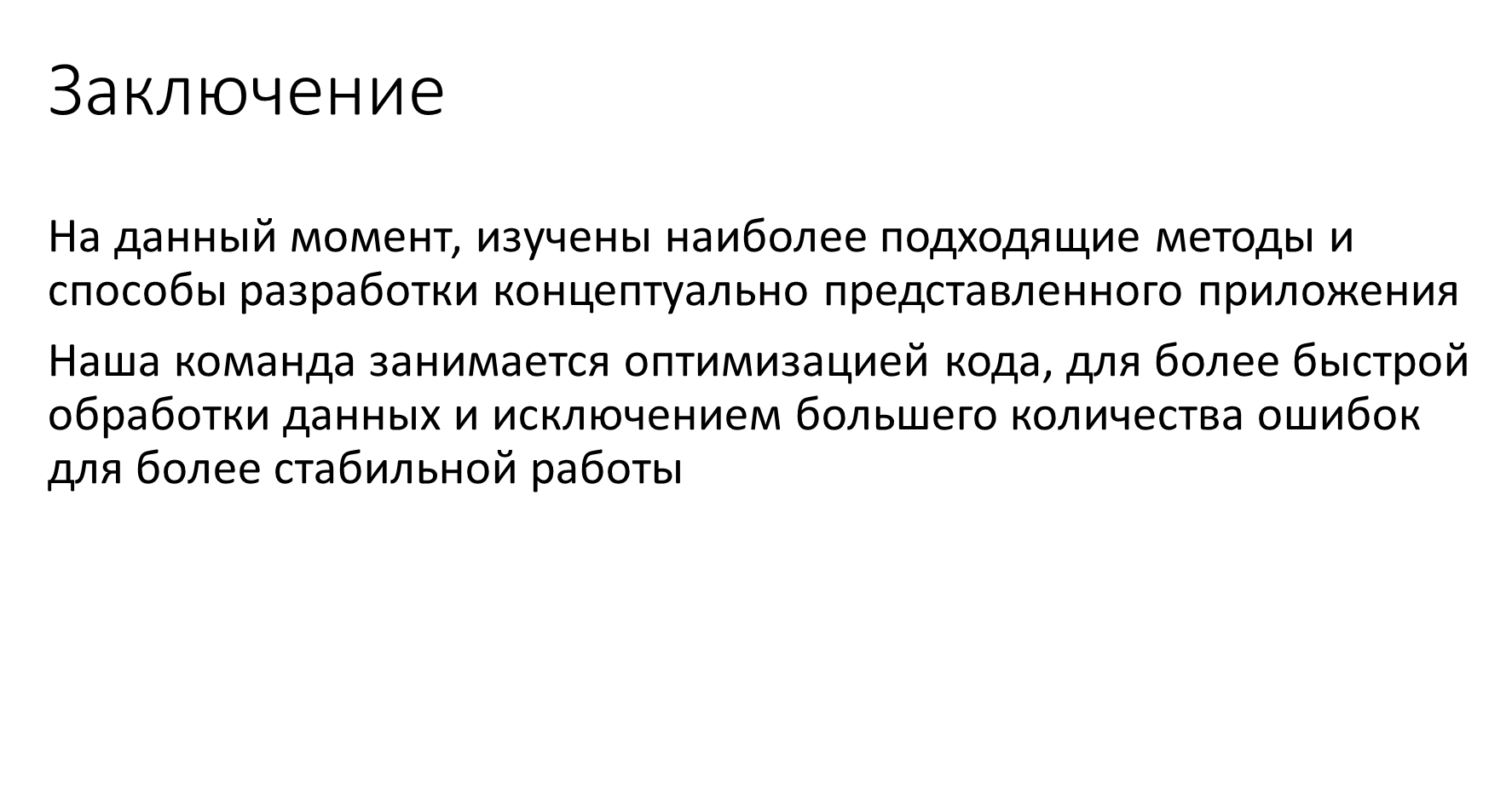
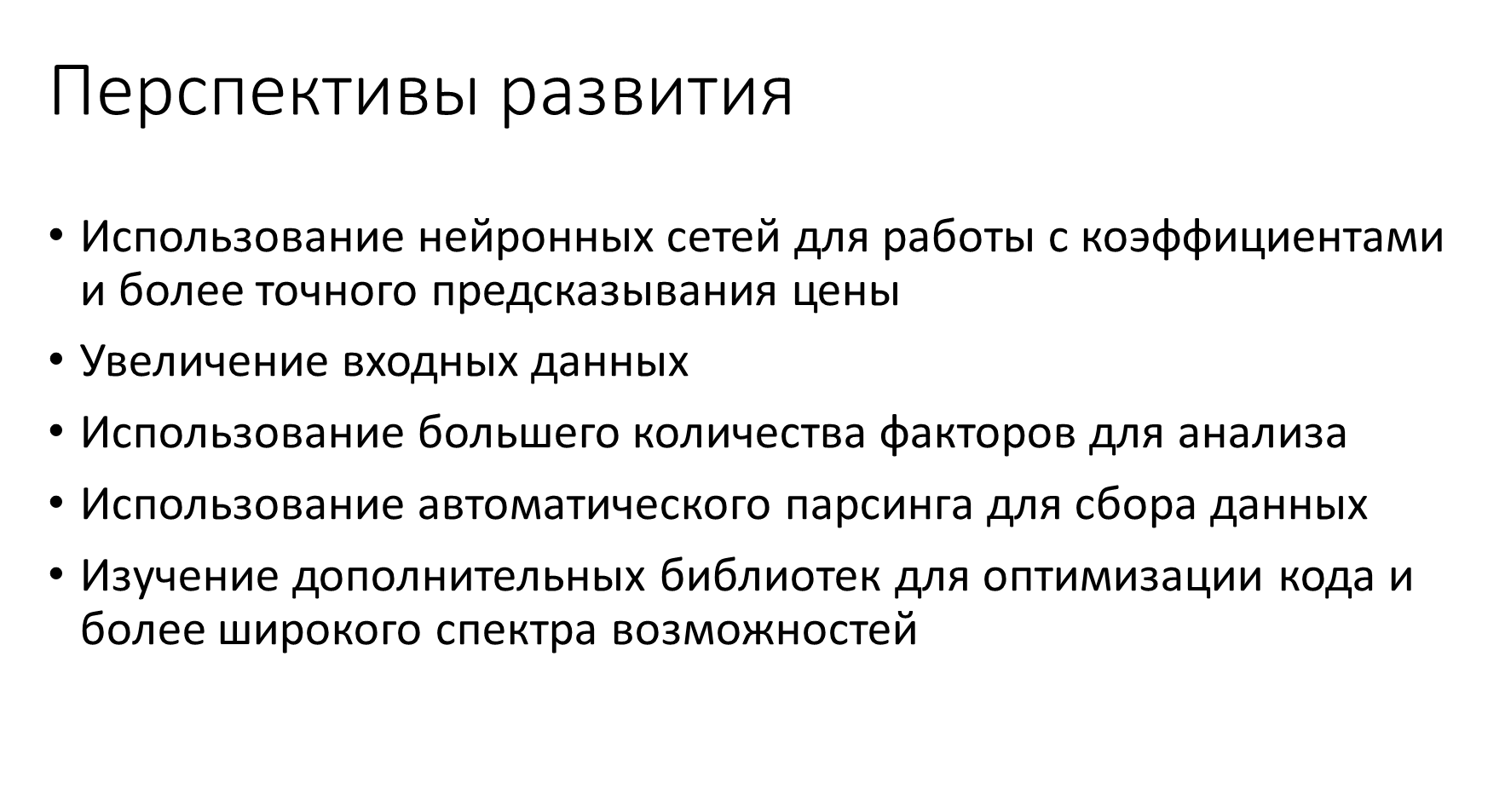


Рис. 25. 5 слайд Рис. 26. 6 слайд

 Рис. 27. 7 слайд Рис. 28. 8 слайд

34

 Рис. 29. 9 слайд Рис. 30. 10 слайд

 Рис. 31. 11 слайд Рис. 32. 12 слайд

35

Код методов

Друганов А.В. :

scaled\_df = StandardScaler().fit(data\_x).transform(data\_x)

model = LinearRegression().fit(data\_x, data\_y)

print(model.intercept\_, ' - Аддитивная случайная величина')

plt.scatter(np.array(data\_x)[:, 1], model.predict(np.array(data\_x)), color='r', label='prediction', s=0.6)

plt.scatter(np.array(data\_x)[:, 1], data\_y[: Listlen], color='b', label='original\_data', s=0.6)

plt.xlabel("Площадь")

plt.ylabel("Цена")

plt.legend(["Предсказанная цена","Цена объявления"])

plt.show()

linear\_regression\_model = SGDRegressor(tol=.0001, eta0=.001)

linear\_regression\_model.fit(scaled\_df, data\_y)

predictions = linear\_regression\_model.predict(scaled\_df)

mse = mean\_squared\_error(data\_y, predictions)

print("RMSE: {}".format(np.sqrt(mse)))

print(model.coef\_,

' - Коэффициенты параметров: 1 - Кол-во комнат; 2 - Площадь; 3 - Этаж; 4 - Сколько прошло с выставления объявления (в часах)')

Данный отрывок кода представляет процесс регрессионного анализа входных данных, переданных в массив, а также использования стохастический градиентный спуск, для оптимизации работы.

OpenFileDialog opf = new OpenFileDialog();

opf.Filter = "Excel (\*.XLSX)|\*.XLSX";

opf.ShowDialog();

tb = new System.Data.DataTable();

filename = opf.FileName;

dataGridView1.Rows.Clear();

dataGridView1.Refresh();

DataTableCollection tableCollection;

int rCnt;

int cCnt;

36

string str;

Microsoft.Office.Interop.Excel.Application ExcelApp = new

Microsoft.Office.Interop.Excel.Application();

Microsoft.Office.Interop.Excel.\_Workbook ExcelWorkBook;

Microsoft.Office.Interop.Excel.Worksheet ExcelWorkSheet;

Microsoft.Office.Interop.Excel.Range ExcelRange;

ExcelWorkBook = ExcelApp.Workbooks.Open(filename, 0, true, 5, "", "", true, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows, "\t", false,

false, 0, true, 1, 0);

ExcelWorkSheet=(Microsoft.Office.Interop.Excel.Worksheet)ExcelWorkBook.Worksheets.get\_Item(1);

ExcelRange = ExcelWorkSheet.UsedRange;

double i = 0;

int b = 0;

int kolvo = 0;

rCnt = 0;

dataGridView1.Rows.Add(1);

for (kolvo = 1; kolvo <= 25;)

{

filter()

add\_row();

}

Жиряков В.С. :

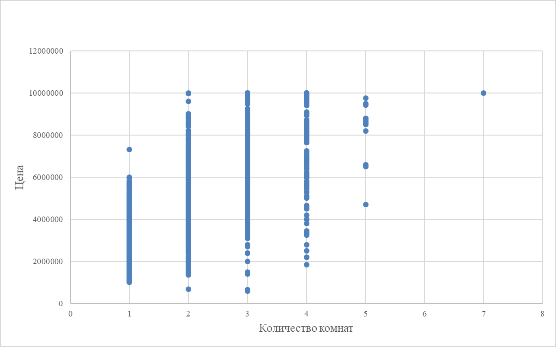


Рис. 33 Корреляционное поле, где факторная переменная количество комнат, а результирующий фактор - стоимость квартир

37

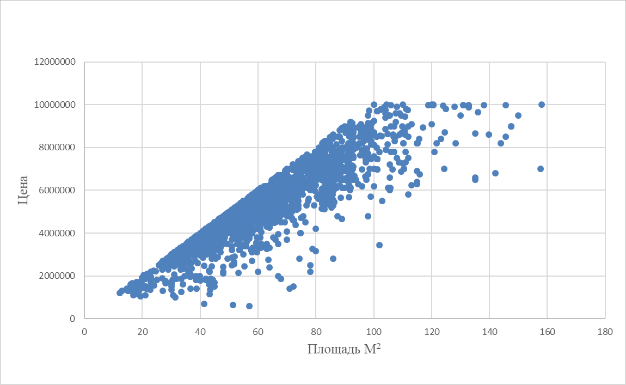


Рис. 34 Корреляционное поле, где факторная переменная площадь, а результирующий фактор - стоимость квартир

С помощью встроенного пакета анализа данных, были получены соответствующие коэффициенты корреляции каждого из параметров по отношению к цене и была выявлена зависимость, на которой видно, что самыми главными ценообразующими факторами являются площадь (0,891489) и количество комнат (0,634476), а такие факторы как этаж и дата размещений наоборот практически не имеют веса в ценообразовании.

Катанаев Д.М. :

def custom\_key(house):

return int(house[3])

def add(dataframe, name):

for add\_i in range(len(dataframe)):

try:

if dataframe['Name'][add\_i][0].isnumeric() \

and (int(("".join((dataframe['Price'][add\_i]).split())).replace("₽", ""))

/ float(

dataframe['Name'][add\_i].split(', ')[1].replace('\xa0', '').replace('м²', '').replace(',',

'.')) < 500000):

List.append([int(dataframe['Name'][add\_i].split(', ')[0][0]),

float(dataframe['Name'][add\_i].split(', ')[1].replace('\xa0', '').replace('м²', '').replace(' ','').replace(',', '.')),

38

int(dataframe['Name'][add\_i].split(', ')[2][:2].replace("/", '')),

int(("".join((dataframe['Price'][add\_i]).split())).replace("₽", "")),

dataframe['Adress'][add\_i], dataframe['Date'][add\_i], name,0])

except:

pass

# Открытие excel файлов

List = []

avito\_dataframe = pd.DataFrame(pd.read\_excel('avito.xlsx'))

cian\_dataframe = pd.DataFrame(pd.read\_excel('cian.xlsx'))

domofond\_dataframe = pd.DataFrame(pd.read\_excel('domofond.xlsx'))

add(avito\_dataframe, 'avito')

add(cian\_dataframe, 'cian')

add(domofond\_dataframe, 'domofond')

# /Добавление записей из Excel в массив

List.sort(key=custom\_key)

# Перевод дат в часы

for i in range(len(List)):

List[i][5] = List[i][5].replace('больше', '').replace('около', '').replace('почти', '') \

.replace('сегодня', '3').replace(

'вчера', '24')

t = List[i][5].split()

if t[1] == 'день' or t[1] == 'дней' or t[1] == 'дня':

List[i][5] = int(t[0]) \* 24

if t[1] == 'час' or t[1] == 'часа' or t[1] == 'часов':

List[i][5] = int(t[0])

if t[1] == 'года' or t[1] == 'лет':

List[i][5] = int(t[0]) \* 8760

if t[1] == 'месяца' or t[1] == 'месяцев':

List[i][5] = int(t[0]) \* 708

if t[1] == 'минут' or t[1] == 'минута' or t[1] == 'минуты':

List[i][5] = 1

if t[1] == 'недели' or t[1] == 'неделю':

39

List[i][5] = int(t[0]) \* 168

if 'апр,' in t[1]:

List[i][5] = 300

if ':' in t[1]:

List[i][5] = 12

Обработка входных данных.

sum = 0

Predicted = []

Lengt = []

xs = np.array(List)[:1000,1]

ys = np.array(List)[:1000,3]

plt.scatter(xs,ys)

plt.show()

for i in range(Listlen):

S = np.array([int(List[i][0]), float(List[i][1]), int(List[i][2]), int(List[i][5])]).reshape(1,-1)

Predicted.append(model.predict(S))

Lengt.append(i)

plt.scatter(Lengt, Predicted, color='r', label='prediction', s=0.6)

plt.scatter(Lengt, data\_y, color='b', label='original\_data', s=0.6)

plt.xlabel("Номер объявления")

plt.ylabel("Цена")

plt.legend(["Предсказанная цена","Цена объявления"])

plt.show() # Отсортированный по цене график, на котором каждая точка имеет свой индекс по X [0-7049] - Синие точки : Настоящие цены; Красные точки : Цены предсказанные программой

for i in range(Listlen):

sum += abs(Predicted[i] - data\_y[i])

List[i][7] = (data\_y[i]-Predicted[i])[0]

List.sort(key = custom\_key\_df)

df = pd.DataFrame(List)

40

print(sum / len(Predicted), '- Среднее отклонение от настоящей цены')

df.to\_excel("EndDataframe.xlsx",encoding='cp1251')

List\_coef\_ = [[model.coef\_[0],"Кол-во комнат"],[model.coef\_[1],"Площадь"],[model.coef\_[2],"Этаж"],[model.coef\_[3],"Сколько прошло с выставления объявления (в часах)"],[model.intercept\_,"АСВ"]]

df\_coef\_ = pd.DataFrame(List\_coef\_)

df\_coef\_.to\_excel("CoefDataframe.xlsx",encoding='cp1251')

Отрисовка графиков и создания .xlsx файла.