

Санкт-Петербургский Государственный Университет  
Фундаментальная информатика и информационные технологии  
Информационные технологии

Веревкина Елена Борисовна

Разработка программной системы  
извлечения и анализа  
неструктурированной информации по  
электронным торгам

Бакалаврская работа

Научный руководитель:  
доц. каф. инф., к. ф.-м. н. Григорьев Д. А.

Рецензент:  
генеральный директор ООО «Иновационные Торговые Технологии» Зверьков В. Г.

Санкт-Петербург  
2017

SAINT-PETERSBURG STATE UNIVERSITY

Fundamental Informatics and Information Technology

Information Technology

Elena Verevkina

Development of software for mining and  
analysing unstructured information by  
electronic trading auctions

Bachelor's Thesis

Scientific supervisor:  
Ph. D. of Sc., associate professor Dmitrii Grigoriev

Reviewer:  
CEO Investment Trading Technologies Ltd. Vadim Zverkov

Saint-Petersburg  
2017

# Оглавление

<b>Введение</b>	<b>5</b>
<b>1. Постановка задач</b>	<b>8</b>
<b>2. Обзор существующих решений</b>	<b>9</b>
<b>3. Извлечение данных</b>	<b>11</b>
3.1. Извлечение слабоструктурированной информации . . . . .	12
3.2. Извлечение неструктурированной информации . . . . .	13
<b>4. Анализ данных</b>	<b>17</b>
4.1. Предварительный анализ . . . . .	17
4.2. Прогнозирование финальной цены лота . . . . .	18
4.2.1. Введение в линейную регрессию . . . . .	18
4.2.2. Парная линейная регрессия . . . . .	19
4.2.3. Категориальные параметры . . . . .	23
4.2.4. Выявление оптимальной модели линейной регрессии	25
4.2.5. Сравнение модели линейной регрессии с другими моделями прогнозирования . . . . .	27
4.3. Выявление выигрышной стратегии участия в торгах . .	28
<b>5. Реализация программной системы</b>	<b>33</b>
5.1. Архитектура программной системы . . . . .	33
5.1.1. Модуль получения данных . . . . .	33
5.1.2. Модуль обработки данных . . . . .	40
5.1.3. Модуль обучения и применения модели линейной регрессии . . . . .	41
5.1.4. Модуль площадки для экспериментов . . . . .	42
5.2. Функционал программной системы . . . . .	43
<b>Заключение</b>	<b>45</b>
<b>Список литературы</b>	<b>46</b>

<b>Приложение А</b>	<b>49</b>
<b>Приложение Б</b>	<b>50</b>
<b>Приложение В</b>	<b>51</b>
<b>Приложение Г</b>	<b>52</b>

# Введение

В условиях кризиса банкротство физических и юридических лиц стало частым явлением. При наличии у банкрота имущества, на которое может быть обращено взыскание, происходит его опись, оценка и составление плана продажи. Имущество должника продается на торгах, которые проводятся на различных электронных торговых площадках (ЭТП). Электронные торги позволяют увеличить аудиторию потенциальных покупателей и быстро продать имущество должника.

Продажа имущества банкрота происходит в три этапа. Сначала объявляется первичное предложение в форме открытого аукциона, на этом этапе имущество выставляется по начальной стоимости, торги идут на повышение, выигрывает участник, заявивший наибольшую цену. Если на первом этапе лот не был продан, его цена снижается на 10-30% и снова объявляется открытый аукцион. Если в ходе второго этапа лот не был продан, объявляется третий этап — торги в форме публичного предложения. На этом этапе через определенные промежутки времени, чаще всего 7-14 дней, происходит снижение цены лота на 5-15%, победителем становится участник, сделавший ставку быстрее остальных или предложивший наибольшую цену [8].

Цена имущества на электронных торгах часто бывает крайне низкой, она может достигать 10% от рыночной. Именно поэтому торги на ЭТП интересны как предпринимателям, так и частным лицам. В настоящее время функционирует порядка 60-ти ЭТП [11]. Рынок имущества банкротов находится в постоянном движении, и уследить за всеми лотами на всех площадках становится практически невыполнимой задачей.

На помощь потенциальному участнику торгов приходят сервисы, предоставляющие обширную базу имущества, которая содержит информацию о лотах, продаваемых на различных ЭТП. Количество наиболее популярных сайтов с такой тематикой варьируется в районе 20-ти. Чаще всего функционал таких сервисов не ограничивается сбором информации с нескольких электронных площадок, многие из них позволяют осуществлять фильтрацию и сортировку лотов, подписываться

на лоты определенной категории и даже предоставляют помощь в участии в торгах.

Безусловно, такие сервисы упрощают процедуру поиска нужных лотов, однако, большинство подобных сайтов имеют достаточно скудный функционал и не используют в полной мере возможности, которые открываются при тщательной обработке информации по электронным торгам. Анализ данных по завершившимся торгам представляет большой интерес с точки зрения прогнозирования результатов будущих торгов.

Предложенная в данной работе программная система предоставляет пользователю возможность получить быстрый и достаточно точный прогноз итоговой цены еще не реализованного лота. Опираясь на полученную информацию, пользователь может скорректировать свою стратегию участия в торгах и одержать победу. Также разработанная система позволяет опробовать на выборке из уже проданных лотов различные варианты изменения цены в ходе торгов и получить статистику выигрышей и проигрышей для каждой из стратегий.

Реализация описанного функционала стала возможной благодаря извлечению по каждому лоту слабоструктурированной информации, представленной в виде html страниц, и неструктурированной информации, представленной в виде документов различных форматов: pdf, doc, docx, txt, а также zip-архивов, содержащих большие наборы файлов упомянутых форматов.

Теоретическая ценность данной работы заключается в выявлении факторов, оказывающих наибольшее влияние на стоимость лота на электронных торгах, и применении метода линейной регрессии для прогнозирования итоговой цены лота. Практической целью работы является создание программной системы, предоставляющей пользователю наиболее полную информацию об интересующем его лоте, включающую прогноз итоговой цены для еще не реализованных лотов и данные об участниках и времени продажи для реализованных, а также данная система должна включать площадку для проведения экспериментов, в ходе проведения которых пользователь может выработать собствен-

ную стратегию участия в торгах. Представленное программное решение призвано расширить функционал сервиса [bankrot-spy.ru](http://bankrot-spy.ru) [18].

# 1. Постановка задач

Целью данной работы является создание программной системы, осуществляющей сбор и анализ информации о лотах, продаваемых на электронных торгах. Функционал данной системы должен включать в себя прогнозирование итоговой цены еще не реализованных лотов, а также возможность получения статистики результатов торгов по уже реализованным лотам.

Для достижения поставленной цели были сформулированы следующие задачи:

1. Разработка алгоритмов извлечения неструктурированной информации из документов форматов pdf, doc, docx, txt, zip-архивов, а также слабоструктурированной информации из html страниц.
2. Нахождение факторов, оказывающих наибольшее влияние на изменение цены лота в ходе торгов, для выявления закономерностей был выбран набор данных по электронным торгам, на которых продавались автотранспортные средства.
3. Апробирование и оценка различных моделей линейной регрессии, осуществляющих прогнозирование финальной цены лота, выбор оптимальной модели.
4. Создание десктопного приложения, функционал которого включал бы в себя, помимо предоставления полной информации по лотам, прогнозирование цены лотов и платформу для апробирования на уже реализованных лотах различных стратегий участия в торгах.



## 2. Обзор существующих решений

В настоящее время функционирует более десятка сервисов, собирающих информацию о лотах, продаваемых на разных ЭТП, предоставляющих возможности их фильтрации, сортировки и анализа. С целью сравнения функционала существующих сайтов и представляемой в данной работе программной системы, были рассмотрены три самых популярных сервиса, помогающих найти лот на аукционе по банкротству [20]. Особый акцент в сравнении функционалов сервисов был сделан на наличие данных о лоте, полученных путем извлечения неструктурированной информации, а также на наличие и количество прогнозируемой информации для еще не реализованных лотов.

	Дата и время начала и окончания приема заявок	Победитель и участники	Период подачи победной заявки	Дата и время подачи победной заявки	Прогнозирование финальной цены лота	Возможность опробовать стратегии
<a href="http://www.probankrot.ru">http://www.probankrot.ru</a>	+	-	+	-	-	-
<a href="http://bankrot.pro">http://bankrot.pro</a>	+	-	-	-	-	-
<a href="http://bankrot-pro.com">http://bankrot-pro.com</a>	+	-	-	-	-	-
Представляемая программная система	+	+	+	+	+	+

Таблица 1: Результаты анализа функционала существующих сайтов, предоставляющих информацию по электронным торгам

Программное обеспечение, представленное в работе, разрабатывалось, как дополнение функционала сервиса [bankrot-spy.ru](http://bankrot-spy.ru), поэтому данный сервис не будет включен в сравнительный анализ. Среди задач, поставленных перед реализуемой программной системой, не было задач поиска лота в базе, сортировки и фильтрации лотов, так как с этими задачами успешно справляется существующий функционал сер-

веса bankrot-spy.ru, поэтому сравнение сервисов не будет проводиться по упомянутым критериям.

На основании результатов анализа возможностей существующих программных решений был сделан вывод, что функционал представляемой в данной работе системы не имеет аналогов в своей отрасли (см. табл. 1).

### 3. Извлечение данных

Основным источником информации о торгах по реализации имущества банкротов является сайт единого федерального реестра сведений о банкротстве (федресурс) [12]. Именно на этом сайте в обязательном порядке публикуются данные о ходе процедур банкротства на территории РФ в соответствии с Федеральным законом № 127 «О несостоятельности (банкротстве)» от 26.10.2002 [21]. Своевременная публикация и достоверность сведений входит в обязанности арбитражных управляющих в соответствии с ч.3 ст. 14.13 КоАП РФ [9]. Таким образом, данный реестр содержит информацию о лотах, торги по которым ведутся на различных ЭТП. Появляется возможность провести анализ достаточно большого и разнообразного набора данных. Вся работа проводилась с данными по лотам из категории «автотранспортные средства».

Была поставлена задача извлечения следующих данных по каждому еще не реализованному лоту: вид торгов, начальная цена, шаг аукциона, дата и время начала приема заявок, дата и время окончания приема заявок. Для реализованных лотов дополнительно нужно было извлечь итоговую цену, количество принятых заявок, имена победителя и участников торгов, дату и время подачи победной заявки. Сервисом bankrot-sru, в рамках которого планируется дальнейшее функционирование описываемой программной системы, были предоставлены следующие данные по каждому лоту: уникальный идентификатор лота, название лота, ссылка на страницу с информацией о нем на федресурсе, регион арбитражного суда, модель автотранспортного средства. Модель была корректно определена у 36% рассмотренных лотов.

Для лотов, которые еще не были реализованы, оказалось невозможным получение текущего количества принятых заявок на участие и списка участников, так как эта информация публикуется вместе с результатами проведения торгов. Это сделало неприменимой на практике модель прогнозирования на основе классификации торгов по похожим участникам.

Для получения всех необходимых данных по лоту были изучены и

реализованы алгоритмы извлечения слабоструктурированной и неструктурированной информации.

### **3.1. Извлечение слабоструктурированной информации**

Под слабоструктурированной информацией здесь понимается информация, представленная в виде html страниц, именно в таком виде хранится большинство параметров лота. Появляется задача извлечения небольших фрагментов данных из web-страниц.

Для извлечения данных из html страницы была выбрана библиотека для .NET HtmlAgilityPack [15]. Библиотека позволяет анализировать DOM дерево, используя технологию XPath.

XPath — язык, основанный на выражениях путей к элементам, позволяющих получать доступ к отдельным частям XML документа [4]. XPath применяется и к html документам. Использование навигационных осей существенно расширяет возможности XPath. Становится доступным передвижение по дереву документа во всех направлениях, к тому же отсутствует необходимость проходить все уровни вложенности на пути к нужному элементу. Технология XPath позволяет использовать различные предикаты, такие как предикаты сравнения, существования, позиционные предикаты. Функционал XPath включает в себя возможность манипуляции наборами узлов, например, арифметические операции и агрегацию, а также позволяет осуществлять действия со строками и конвертацию типов.

В представляемой программной системе в процессе извлечения информации просматриваются более 7 000 страниц с одинаковой структурой, дизайном и DOM деревьями, поэтому можно сказать, что в данной ситуации XPath является оптимальным решением для извлечения информации.

Одна страница на федресурсе может содержать информацию о нескольких десятках лотов, торги по котором проводятся отдельно. Обычно такая ситуация возникает, когда продаваемое имущество принадлежа-

ло до конфискации одному владельцу и поэтому сейчас продается единым набором. В этом случае появляется задача распознавания данных, относящихся именно к интересующему лоту. Информация о каждом лоте содержится в отдельной таблице, технология XPath позволяет найти нужную таблицу, используя локальный путь узла, содержащего номер интересующего лота.

При приведении результатов поиска на основе XPath к необходимым форматам применялись поиск на основе регулярных выражений, а также парсинг строк. Осуществлялось удаление пробелов, ключевых слов и т.д.

При извлечении информации из сообщений о результатах торгов и о снижениях цены был применен поиск на основе регулярных выражений. Данные сообщения представляют собой html страницы с очень небольшим и четко регламентированным объемом информации.

### **3.2. Извлечение неструктурированной информации**

Обработка естественного языка (Natural Language Processing, NLP) — это обширная область ИТ, связанная с использованием компьютеров для анализа естественных языков [10].

Во время проведения торгов, участники подают заявки, содержащие цену, которую они готовы заплатить за продаваемый лот. Выше уже говорилось о том, что на определение победителя влияют не только цены, указанные в заявках участников, но и время, когда эти заявки были поданы. Так, если двое участников сделали одинаковое ценовое предложение, побеждает тот, кто отправил заявку первым. К тому же дата и время подачи победной заявки несут в себе информацию о том, на каком этапе торгов был куплен лот, сколько раз цена лота понижалась до его реализации, какова разница между ценой в победной заявке и ценой после последнего снижения. Таким образом, информация о дате и времени подачи победной заявки поможет составить наиболее полную статистическую картину.

Информация о дате и времени подачи победной заявки содержит-

ся в документах, называемых протоколами о результатах проведения торгов. Форматы этих документов бывают разными: pdf, doc, docx, txt, zip. Ситуация осложняется тем, что не существует единого стандарта заполнения протокола о результатах проведения торгов, каждая ЭТП имеет не только свой формат документа, но и его структуру. Так, дата и время подачи победной заявки могут находиться в таблице или в тексте, быть оформленными исключительно цифрами или содержать буквенные обозначения.

Для zip-архивов сначала проводилась распаковка, потом определение форматов файлов. Как было сказано ранее, одна страница на федеральном ресурсе может содержать информацию о нескольких десятках лотов, часто в таком случае документы, содержащие протоколы результатов проведения торгов, публикуются единым zip-архивом, который может содержать от двух до нескольких десятков документов. Причем названия документов в архиве не несут никакой информации о том, к какому лоту они относятся. В этом случае приходится распаковывать архив, распознавать и просматривать все документы в поисках нужного. Нужный документ определяется достаточно просто по номеру или названию лота, однако, для того, чтобы получить доступ к тексту документа, приходится совершать достаточно долгие операции разархивирования и распознавания неструктурированного текста.

Для извлечения текста из файлов форматов pdf и docx используется библиотека TikaOnDotNet [17]. На листинге 1 продемонстрировано преобразование документа формата docx или pdf в строковый тип.

Листинг 1: Метод, осуществляющий преобразование документов форматов pdf и docx в строковый тип

```
public static string pdf_docxToString(string path)
{
    var textExtractor = new TextExtractor();
    var result = textExtractor.Extract(path);
    return result.ToString(); }
}
```

Извлечение данных из документа формата doc осуществляется иначе. Для этого используется библиотека Microsoft.Office.Interop.Word [16]. С ее помощью создается объект *Application*, который открывает и построчно читает документ формата doc. В случае с zip-архивами, содержащими несколько десятков документов, из которых только один соответствует нужному лоту, построчный просмотр позволяет прекратить считывание документа уже в самом начале, если номер лота, указанный в документе, не совпадает с искомым.

После того, как текст извлечен из документов различных форматов, появляется задача идентифицировать в тексте дату и время подачи победной заявки. Ситуация осложняется тем, что текст содержит несколько дат: дата составления документа, дата начала приема заявок, даты подачи заявок другими участниками.

Была выявлена закономерность взаимного расположения имени победителя и даты подачи его заявки, а именно, и в тексте, и в таблице, они всегда следуют друг за другом. Таким образом, был реализован поиск на основе регулярных выражений [1], который дал верный результат в 89% случаев.

Дата и время не всегда представлены в едином формате. Были обнаружены следующие варианты их написания:

- 11.02.17 г. в 13:14
- 11.02.17 г. в 13:14:15
- 11.02.17 г. в 13:14:15.103
- «11» февраля 2017 г., время: 13:14
- «11» февраля 2017 г., время: 13:14:15
- «11» февраля 2017 г., время: 13:14:15.103
- 11.02.2017 13:14
- 11.02.2017 13:14:15

- 11.02.2017 13:14:15.103

Были реализованы методы, осуществляющие с помощью регулярных выражений извлечение даты и времени разных форматов [1], а также их преобразование в стандартный тип DateTime.

Используя описанные подходы, удалось корректно определить дату и время победной заявки для 81% успешно проданных лотов, для которых имелся протокол результатов проведения торгов.

Всего было распознано более 1000 файлов, из которых около половины являлись zip-архивами, содержащими в среднем 50-60 документов.



## 4. Анализ данных

### 4.1. Предварительный анализ

Был проведен предварительный анализ набора лотов, предоставленного сервисом bankrot-spy.ru.

Всего было предоставлено 5362 лота, из которых 3914 лотов продавались на торгах в форме открытого аукциона, а 1448 лотов — в форме публичного предложения. Победитель был определен только в 1135 торгах, остальные лоты не были реализованы по причине отсутствия заявок. Из состоявшихся торгов 386 проводились в форме открытого аукциона, 749 — в форме публичного предложения (см. рис. 1).



Рис. 1: Диаграмма соотношения количества торгов разных типов.

На этапе открытого аукциона лот продается редко, примерно один лот из десяти, на этапе публичного предложения продается практически каждый второй лот. Исходя из полученных результатов, было

решено во время анализа полученной информации сконцентрировать внимание именно на лотах, продаваемых в форме публичного предложения. К тому же прогнозирование цены для таких лотов является актуальной задачей еще и потому, что участники не имеют возможности видеть ставки друг друга, как это происходит в случае торгов в форме открытого аукциона.

## **4.2. Прогнозирование финальной цены лота**

Выше уже было сказано, что победителем торгов в форме публичного предложения становится участник, сделавший ставку быстрее остальных или предложивший наибольшую цену. Сложности заключаются в том, что участники торгов в форме публичного предложения не имеют доступа к информации о том, сколько человек участвует в торгах, когда и с какими ценами они подают заявки. Здесь важно попасть точно в цель, подать заявку с такой ценой, которая будет превышать цены в заявках других участников, но при этом разница с ними будет как можно меньше. Ведь логично предположить, что если цена в заявке участника будет превышать текущую цену лота в несколько раз, то он с большой вероятностью победит, однако, он скорее всего не получит никакой выгоды от участия в торгах, так как купит лот по цене близкой к рыночной или даже превышающей ее.

Прогнозирование финальной цены лота может помочь избежать подобных ситуаций. В данной главе будут рассмотрены подходы к анализу результатов завершенных торгов и прогнозированию результатов будущих торгов, направленные на помощь участнику в его стремлении одержать победу с минимальными финансовыми потерями.

### **4.2.1. Введение в линейную регрессию**

Метод линейной регрессии является распространенным способом прогнозирования зависимой величины на основе одной или нескольких независимых переменных.

В методе множественной линейной регрессии связь переменной  $Y$

с переменными  $X_1, \dots, X_n$  задается с помощью линейной модели

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_n X_n + \varepsilon,$$

где  $\beta_0, \beta_1, \dots, \beta_n$  — вещественные регрессионные коэффициенты,  $\varepsilon$  — случайная величина, являющаяся ошибкой прогнозирования [6].

Поиск регрессионных коэффициентов осуществляется по обучающей выборке — набору данных о завершенных торгах. Таким образом, задача прогнозирования сводится к нахождению прямой, которая будет наилучшей аппроксимацией данных из обучающей выборки.

Основной задачей данного раздела является прогнозирование финальной цены для незавершенных торгов, следовательно, именно финальная цена и будет единственным выходным параметром линейной регрессии. С целью выявления наилучшего с точки зрения качества прогноза набора входных параметров, было проведено сравнение коэффициентов корреляции между финальной ценой и другими параметрами лота, построено и оценено несколько регрессионных моделей с разными входными параметрами.

Так как необходимым условием создания применимой на практике модели прогнозирования является наличие корректно определенных входных параметров регрессии как у реализованных, так и у еще не реализованных лотов, были выявлены следующие возможные регрессоры: начальная цена лота, регион, номер квартала, в котором был начат прием заявок.

#### **4.2.2. Парная линейная регрессия**

Рассмотрим самую простую и очевидную зависимость — зависимость итоговой цены от начальной. Очевидно, именно эта зависимость послужит фундаментом для добавления прочих параметров, улучшающих прогноз.

Коэффициент корреляции характеризует меру линейной зависимости двух переменных. Рассмотрим коэффициент корреляции Пирсона между начальной и финальной ценами лота, который задается форму-

лой [7]:

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2 \sum_{j=1}^n (Y_j - \bar{Y})^2}}$$

где  $X_i$  — начальная цена лота,  $Y_i$  — финальная цена лота,  $\bar{X}$  и  $\bar{Y}$  — выборочные средние, определяющиеся следующим образом

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n (X_i),$$

$$\bar{Y} = \frac{1}{n} \sum_{i=1}^n (Y_i).$$

Коэффициент корреляции оказался равным 0,868, это позволяет сделать вывод, что между начальной и финальной ценами существует достаточно сильная линейная зависимость. Следовательно, будет уместно применить линейную регрессию для прогнозирования финальной цены на основе начальной.

Для оценки качества построенной модели линейной регрессии воспользуемся диаграммой рассеяния (см. рис. 2). Диаграмма подтверждает тот факт, что По диаграмме видно, что данные хорошо укладываются в линейную модель. Следовательно, на основании этой модели можно сделать прогноз на будущее.

Теперь рассмотрим коэффициент детерминации такой модели. Коэффициент детерминации показывает, какая доля вариации зависимой переменной  $Y$  учтена в модели и обусловлена влиянием на нее факторов, включенных в модель [22]:

$$R^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2},$$

где  $Y_i$  — реальные значения зависимой переменной,  $\bar{Y}$  — среднее

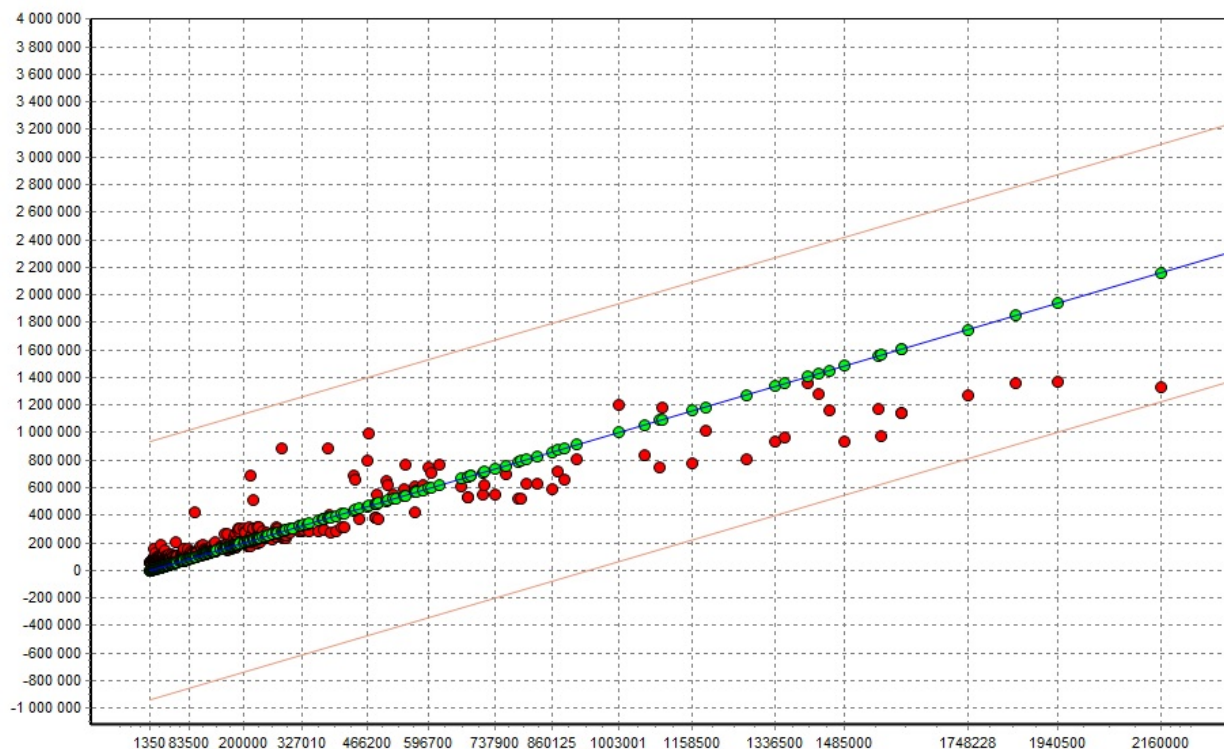


Рис. 2: Диаграмма рассеяния парной линейной регрессии.

значение зависимой переменной,  $\hat{Y}_i$  — прогнозируемые значения зависимой переменной, построенные в результате применения линейной регрессии.

Коэффициент детерминации для модели линейной регрессии с одним входным параметром — начальной ценой и одним выходным параметром — финальной ценой равняется 0,759.

Недостаток коэффициента детерминации, как критерия оценивания качества регрессионной модели, заключается в том, что его значение никогда не убывает с ростом предсказываемых переменных [6]. Таким образом, коэффициент детерминации будет выше для моделей, содержащих большее количество регрессоров. Этот недостаток является несущественным в случае, если требуется сравнивать модели с одинаковым количеством переменных, однако, в данном случае стоит противоположная задача.

Для дальнейшего сравнения качества моделей линейной регрессии введем скорректированный коэффициент детерминации:

$$\bar{R}^2 = 1 - \frac{n-1}{n-q-1} \cdot (1 - R^2),$$

где  $n$  — размер выборки,  $q$  — число регрессоров.

В отличие от коэффициента детерминации, введенного ранее, скорректированный коэффициент детерминации уменьшается с ростом регрессоров, если  $(1 - R^2)$  оказывается недостаточным для компенсации увеличения отношения  $\frac{n-1}{n-q-1}$  [6].

Для рассмотренной модели линейной регрессии различие двух коэффициентов детерминации ничтожно мало, скорректированный коэффициент детерминации оказался равным 0,758.

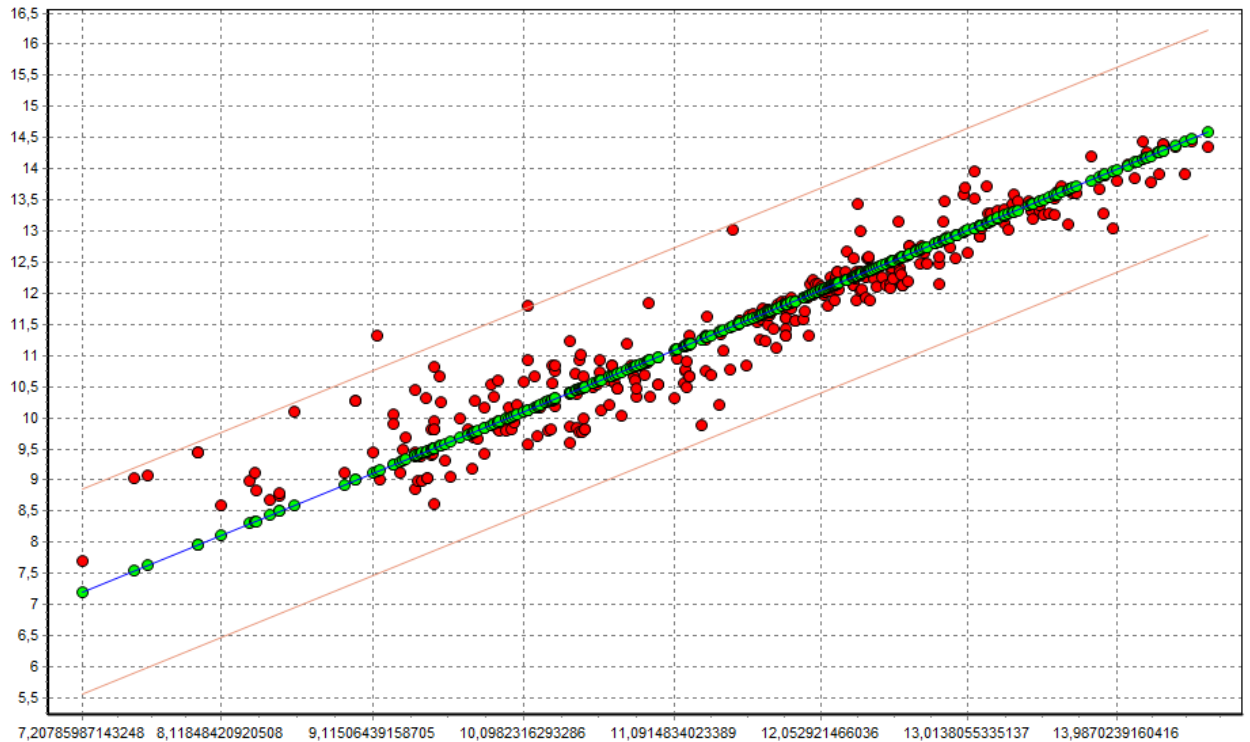


Рис. 3: Диаграмма рассеяния парной линейной регрессии, в которой используются натуральные логарифмы цен лотов.

Прежде чем приступить к улучшению качества модели линейной регрессии путем введения дополнительных параметров, изменим входные значения таким образом, чтобы, с одной стороны, не допустить прогнозирования отрицательных величин, с другой стороны, увеличить скорректированный коэффициент детерминации модели. Желаемого

результата можно достичь использованием в качестве входных и выходных параметров регрессии натуральные логарифмы цен лотов [2]. По диаграмме рассеяния видно, что преобразованные переменные хорошо укладываются в модель линейной регрессии (см. рис. 3). Скорректированный коэффициент детерминации такой модели линейной регрессии составляет 0,805.

В дальнейшем будет совершенствоваться путем введения дополнительных регрессоров именно эта модель линейной регрессии.

### 4.2.3. Категориальные параметры

Значения категориальных параметров определяют факт принадлежности объекта к какой-либо категории [5]. В рассматриваемом наборе данных категориальными признаками являются регион, квартал года, а также модель автомобиля, но модель не задается явно при публикации информации о торгах и извлекается корректно из названия лота только в 40% случаев, поэтому этот признак не пригоден для применения в итоговой модели прогнозирования.

Часто категориальные признаки представлены строковыми значениями, поэтому, прежде чем использовать их в регрессионной модели, нужно провести нормализацию, то есть преобразовать к виду наиболее подходящему для обработки алгоритмом. В случае линейной регрессии это числовой тип.

Существует несколько способов преобразования категориальных данных. Рассмотрим их на примере регионов.

- Кодирование уникальным значением. Сопоставим каждому региону целое число (см. табл. 2). Такой способ кодирования проецирует категориальные признаки на вещественную прямую. Это приводит к ложным интерпретациям, так, например, Пермский край становится равен сумме Самарской и Московской областей. Категориальность данных теряется.
- Битовое кодирование. Все значения заменяются порядковыми номерами, которые рассматриваются в двоичном виде. Каждый раз-

region	id
Московская область	1
Самарская область	2
Пермский край	3

Таблица 2: Кодирование уникальным значением

ряд двоичного числа рассматривается как отдельное поле, содержащее ноль или единицу (см. табл. 3). Недостатком является возникновение ложных связей и интерпретаций.

region	region1	region2
Московская область	0	1
Самарская область	1	0
Пермский край	1	1

Таблица 3: Битовое кодирование

- Димми-кодирование [3]. Для категориального признака задаются  $N$  новых дихотомических признаков. Получается бинарная матрица, у каждого категориального признака только в одном из дихотомических признаков стоит единица, в остальных нули. Следует исключить одну из димми-переменных из регрессионной модели, чтобы избежать абсолютной мультиколлинеарности. В таблице 4 можно исключить, к примеру, region3, тогда категория «Пермский край» станет референтной категорией.

region	region1	region2	region3
Московская область	1	0	0
Самарская область	0	1	0
Пермский край	0	0	1

Таблица 4: Димми-кодирование

- Реализация отдельной регрессии для каждой категории. Недостатками подхода являются увеличение числа необходимых ре-



грессий и снижение мощности проверки, поскольку каждая регрессия будет реализовываться на меньшей по размеру выборке, чем в случае общего регрессионного уравнения.

В случае с географическими категориями следует рассмотреть еще один способ кодирования. Этот способ дает хорошие результаты, если известно, что значение зависимой переменной не просто коррелирует с областью или городом, а изменяется в соответствии с передвижением по карте. К примеру, если на аукционах в Москве и области цена обычно падает в 3 раза, в соседних регионах в 2, а в удаленных — в 1,5 раза. Однако, в анализируемом наборе данных подобные зависимости установить не удалось, так, например, среднее отношение финальной цены к начальной в Самарской области составляет 0,56, в Пермском крае — 0,57, а в Москве и Московской области — 0,64.

В качестве оптимального способа кодирования был выбран метод *dummy-переменных*. Именно с его помощью были получены дальнейшие результаты.

#### 4.2.4. Выявление оптимальной модели линейной регрессии

Коэффициент детерминации показывает то, насколько хорошо описаны данные, но не характеризует в полной мере качество построенной модели.

Рассмотрим различные процентные ошибки прогнозирования временного ряда [19].

MAPE (Mean Absolute Percentage Error) — средняя абсолютная ошибка в процентах. Вычисляется по формуле:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|Y_i - \hat{Y}_i|}{Y_i},$$

где  $n$  — количество элементов в тестовой выборке,  $Y_i$  — реальные значения зависимой переменной,  $\hat{Y}_i$  — прогнозируемые значения зависимой переменной, построенные в результате применения линейной регрессии.

Недостатком MAPE является то, что она по-разному реагирует на

положительные и отрицательные ошибки, что вызывает грубые неточности. Вместо нее будем использовать другую процентную ошибку.

$sMAE$  (scaled Mean Absolute Error) — средняя абсолютная масштабированная (относительно среднего уровня ряда) ошибка. Вычисляется по формуле:

$$sMAE = \frac{k}{n} \cdot \frac{\sum_{i=1}^n |Y_i - \hat{Y}_i|}{\sum_{j=1}^k |Y_j|},$$

где  $k$  — размер обучающей выборки.

Умноженная на 100  $sMAE$  показывает процент ошибки прогнозирования.

В таблице 5 приведены результаты апробирования модели множественной линейной регрессии на различных входных параметрах.

Начальная цена	Регион	Номер квартала	Скорректированный коэффициент детерминации	$sMAE * 100$
+	–	–	0,805	33
+	+	–	0,864	24
+	–	+	0,798	33
+	+	+	0,859	25

Таблица 5: Результаты тестирования моделей линейной регрессии

Была рассмотрена модель множественной линейной регрессии, в которой прогнозирование финальной цены лота осуществлялось исходя из историй торгов тех людей, которые подали заявки на участие в торгах по рассматриваемому лоту. Такая модель оказалась весьма точной, скорректированный коэффициент детерминации составил 0,896,  $sMAE$  — 21. Но, к сожалению, такая модель не может быть использована на практике, так как списки участников появляются уже после завершения торгов.

Был сделан вывод, что в наибольшей степени качество модели линейной регрессии улучшило введение входного параметра «регион».

Именно такая модель легла в основу программной системы.

#### 4.2.5. Сравнение модели линейной регрессии с другими моделями прогнозирования

Теперь, когда на примере модели линейной регрессии был рассмотрен алгоритм действия математических моделей прогнозирования на основе машинного обучения, а также были выявлены признаки в наибольшей степени влияющие на финальную цену лота, можно рассмотреть другие модели прогнозирования и сравнить их. С помощью платформы для анализа данных и графической визуализации результатов Deductor Studio были построены и протестированы следующие математические модели прогнозирования временных рядов [14]:

- Нейронная сеть
- Кластеризация

При тестировании моделей использовались входные параметры, определенные в настоящей главе, как оптимальные: начальная цена лота и регион проведения торгов, прогнозируемая величина — финальная цена лота. Тестирование проводилось на той же выборке, что и множественная линейная регрессия.

Для каждой из моделей, включая множественную линейную регрессию, был рассчитан дополнительный столбец, значения которого показывает, какой процент от начальной цены лота составляет ошибка прогнозирования, и вычисляется по следующей формуле:

$$\frac{|finalPrice - predictFinalPrice|}{startPrice} \cdot 100,$$

где *finalPrice* — реальная финальная цена лота, *predictFinalPrice* — прогнозируемая финальная цена лота, *startPrice* — начальная цена лота. Сравним средние, стандартные отклонения этих столбцов, а также скорректированный коэффициент детерминации и среднюю абсолютную масштабированную ошибку для каждой из моделей.

	Среднее	Стандартное отклонение	$\bar{R}^2$	$sMAE * 100$
Множественная линейная регрессия	16,75	14,05	0,859	24
Нейронная сеть	20,44	13,96	0,851	23
Кластеризация	21,99	18,26	0,73	49

Таблица 6: Результаты тестирования моделей прогнозирования

Из приведенных в таблице 6 результатов, был сделан вывод, что для решения задачи практически в равной степени подходят модели линейной регрессии и нейронных сетей, модель, основанная на кластеризации существенно проигрывает.

### 4.3. Выявление выигрышной стратегии участия в торгах

В ходе торгов в форме публичного предложения цена лота постепенно снижается, как было сказано ранее, чаще всего она снижается каждые 7-14 дней на 5-15%. В каждый из таких периодов участник может подать заявку со своей ценой, заявка будет принята, если предложенная цена не меньше текущей цены лота. Победителем становится тот, кто предложил наибольшую цену за лот, при этом, если поступило несколько одинаковых ценовых предложений, победу одерживает тот участник, который подал заявку раньше.

Из успешно завершившихся 749 торгов, проводимых в форме публичного предложения, только 102 имели на своей странице на федресурсе информацию об этапах снижения цены. Именно они и были использованы для выявления различных стратегий участия в торгах.

На федресурсе нет информации о номере периода снижения цены, однако, эта информация может оказаться полезной для составления статистики и выявления выигрышной стратегии, так как она показывает, как долго длились торги по данному лоту. Для вычисления номера периода потребовалось дополнительно получить информацию о том,

как изменялась цена лота в ходе последнего и предпоследнего снижений.

Используя описанные в главе 3 алгоритмы, удалось получить дату и время подачи победной заявки для 89 лотов из 102. Для 13 лотов данные не были распознаны по следующим причинам: отсутствие (6 лотов) и неполнота (3 лота) документации о результатах проведения торгов, а также публикация непригодного для распознавания скана документа (4 лота).

На странице выбранных лотов в разделе «сообщения» содержатся данные о каждом снижении цены: когда снизилась цена лота и какой она стала после снижения. Количество этих сообщений часто превышает номер периода, в котором была подана победная заявка, поэтому было бы некорректным считать номер периода равным количеству сообщений о снижении цены. Был реализован алгоритм, задачей которого является поиск значения, до которого успела снизиться цена, прежде чем поступила победная заявка, а также значения цены перед этим снижением. Полученные данные позволяют рассчитать номер периода снижения по следующей формуле:

$$Period = \left[ \frac{P_{start} - P_{last}}{P_{last} - P_{before}} \right] + 1,$$

где  $P_{start}$  — начальная цена лота,  $P_{last}$  — цена лота после последнего снижения перед подачей победной заявки,  $P_{before}$  — цена лота, которая была до  $P_{last}$ ,  $[ ]$  — операция взятия целой части числа:  $[x] = \max\{n \in \mathbb{Z} \mid n \leq x\}$ .

Были вычислены разница во времени между последним снижением и подачей победной заявки, а также разница между ценой после последнего снижения и победной ценой.

Полученные в данном разделе данные были проанализированы с точки зрения пригодности к построению на их основе прогнозов для еще не реализованных лотов. Так, например, было бы логично предположить, что существует какая-то зависимость между номером периода снижения цены, разницей между текущей и победной ценами, временем

между последним снижением и подачей победной заявки.

10 из 89 победителей подали заявки не увеличив цену после последнего снижения, при этом только двое из них подали заявки в первые полчаса после снижения и все равно одержали победу. Зависимости с начальной ценой, регионом, количеством принятых заявок, номером периода снижения цены выявлено не было. Это является первым доказательством того, что явной зависимости между временем подачи победной заявки, ценой лота, указанной в ней, и другими параметрами лота не существует.

Разделим множество лотов на два подмножества: с процентом снижения от 3% до 7% и с процентом снижения от 9% до 15%. Для каждого множества рассмотрим зависимость времени между последним снижением цены и подачей победной заявки от номера периода. Из графиков можно сделать вывод, что наибольшее количество лотов из первого множества продаются с 5 по 8 периоды, из второго — с 3 по 5 (см. рис. 4). Между этими двумя показателями также не была выявлена зависимость, которая могла бы использоваться для прогнозирования.

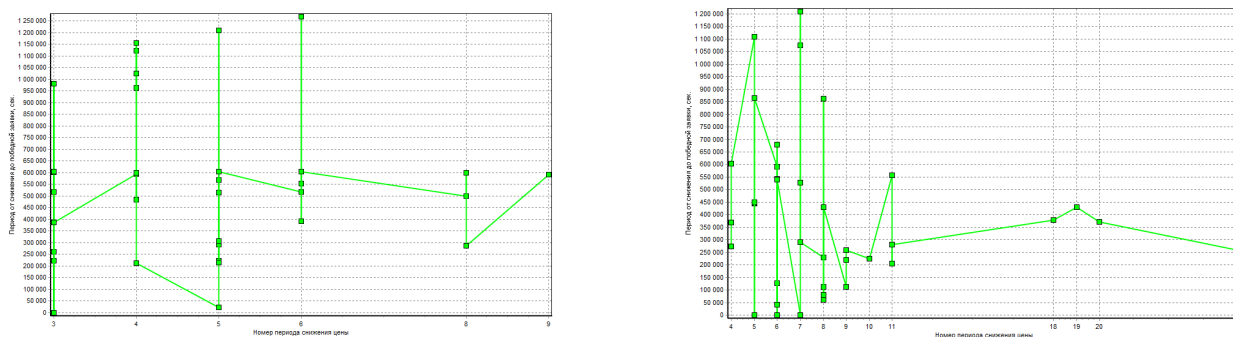


Рис. 4: Графики зависимости времени между последним снижением цены и подачей победной заявки от номера периода снижения цены.

Между разницей текущей и финальной цен и номером периода тоже не была обнаружена зависимость (см. рис. 5).

Рассмотрим зависимость разницы текущей и финальной цен от времени между последним снижением цены и подачей победной заявки. По графикам видно, что чаще всего победные заявки подаются в первую неделю после снижения цены, причем это справедливо для периодов разной длины (см. рис. 6). Между этими двумя показателями также

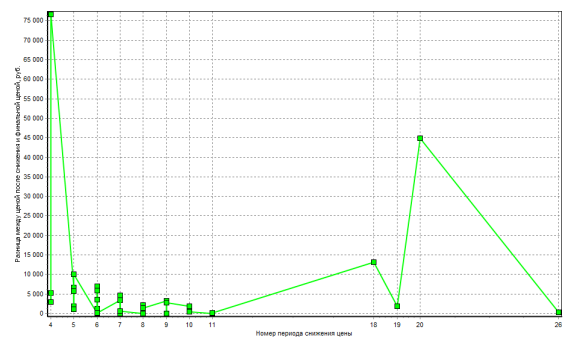
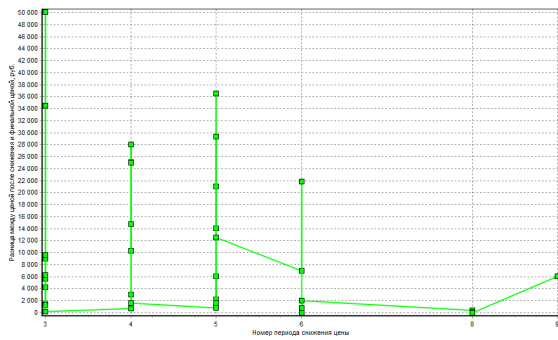


Рис. 5: Графики зависимости разницы текущей и финальной цен от номера периода снижения цены.

не была выявлена зависимость, которая могла бы использоваться для прогнозирования.

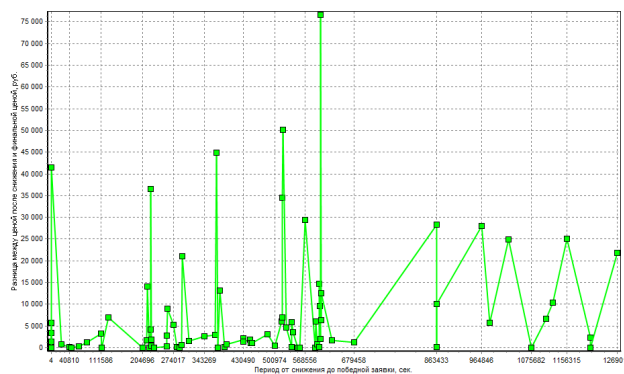


Рис. 6: График зависимости разницы текущей и финальной цен от времени между последним снижением цены и подачей победной заявки.

Оказалось, что данные из проанализированной выборки не пригодны для составления на их основе прогнозов. Однако, на их основе можно составить несколько рекомендаций для пользователя, следуя которым он повысит свои шансы на победу.

Предположим, что существует задача победить в как можно большем количестве торгов. Имеется 89 лотов, о которых нам известно, как в итоге они завершились. Выявим и протестируем стратегии, которые могут помочь выиграть в максимальном количестве торгов с минимальными затратами. Предположим, по умолчанию подается заявка с ценой равной цене лота после последнего снижения.

1. Подача заявки в первые 5 минут после снижения, может обеспе-

чить победу в 9% торгов, при этом даже не придется увеличивать цену.

2. Увеличение цены после последнего снижения на 2 рубля, может обеспечить победу в 9% торгов. Проанализировав набор лотов, можно сделать вывод, что люди чаще всего округляют текущую цену и добавляют один рубль, поэтому решено было предложить пользователю увеличивать цену на 2 рубля.
3. Округление текущей цены до сотен, может обеспечить победу в 12% торгов.
4. Округление текущей цены до тысяч, может обеспечить победу в 21% торгов.

Если последовать первым трем рекомендациям, удастся победить в 15% лотов, при этом переплатив максимум 100 рублей. Эта сумма ничтожно мала, если учесть, что она поможет, к примеру, выиграть торги по автомобилю KIA SLS стоимостью более 600 000 рублей.

Была программно реализованна площадка для экспериментов, которая позволяет применить все представленные стратегии к реальным лотам, также у пользователя существует возможность выставить процент от начальной цены лота, который он готов доплатить, и оценить шансы на победу. Серия экспериментов позволяет сделать выводы о том, какая стратегия для пользователя является оптимальной, выявить наиболее подходящее соотношение вероятности одержать победу и количества требуемых для этого финансовых затрат.

Так, например, в ходе экспериментов был сделан вывод, что можно одержать победу с вероятностью почти 80%, повысив цену на 10%, и с вероятностью 93%, повысив цену на 20%.

При этом повышение цены всего на 3% может привести к победе в более, чем половине рассмотренных торгов (см. рис. 7). У пользователя также существует возможность ознакомиться с графиком, составленным на основе завершившихся торгов, информация по которым находится в базе на текущий момент.



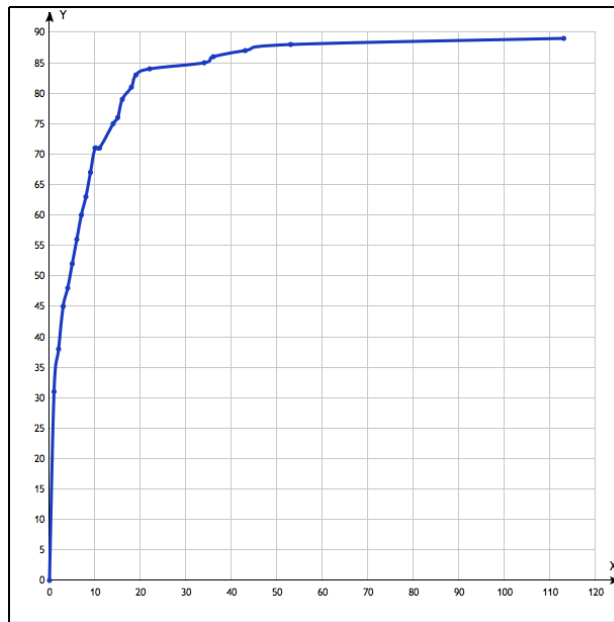


Рис. 7: График зависимости количества побед в торгах от процента повышения цены.

## 5. Реализация программной системы

### 5.1. Архитектура программной системы

Для реализации системы был выбран объектно-ориентированный язык программирования C#. Для разработки использовалась платформа Microsoft Visual Studio 2015 Ultimate. Пользовательский интерфейс системы был реализован с помощью конструктора Windows Forms. Информация о лотах хранится в базе данных, использована СУБД MySql. Реализованная система состоит из 4 программных модулей, которые будут рассмотрены по-порядку.

#### 5.1.1. Модуль получения данных

В данном модуле содержатся классы и методы, осуществляющие связь с сайтом федресурса, а также прямую связь с его сервером, скачивание содержимого html страниц и документов, извлечение из них необходимой информации, приведение ее к единому формату, и последующую запись в базу данных. UML диаграмма классов данного моду-

ля представлена на рисунке 8.

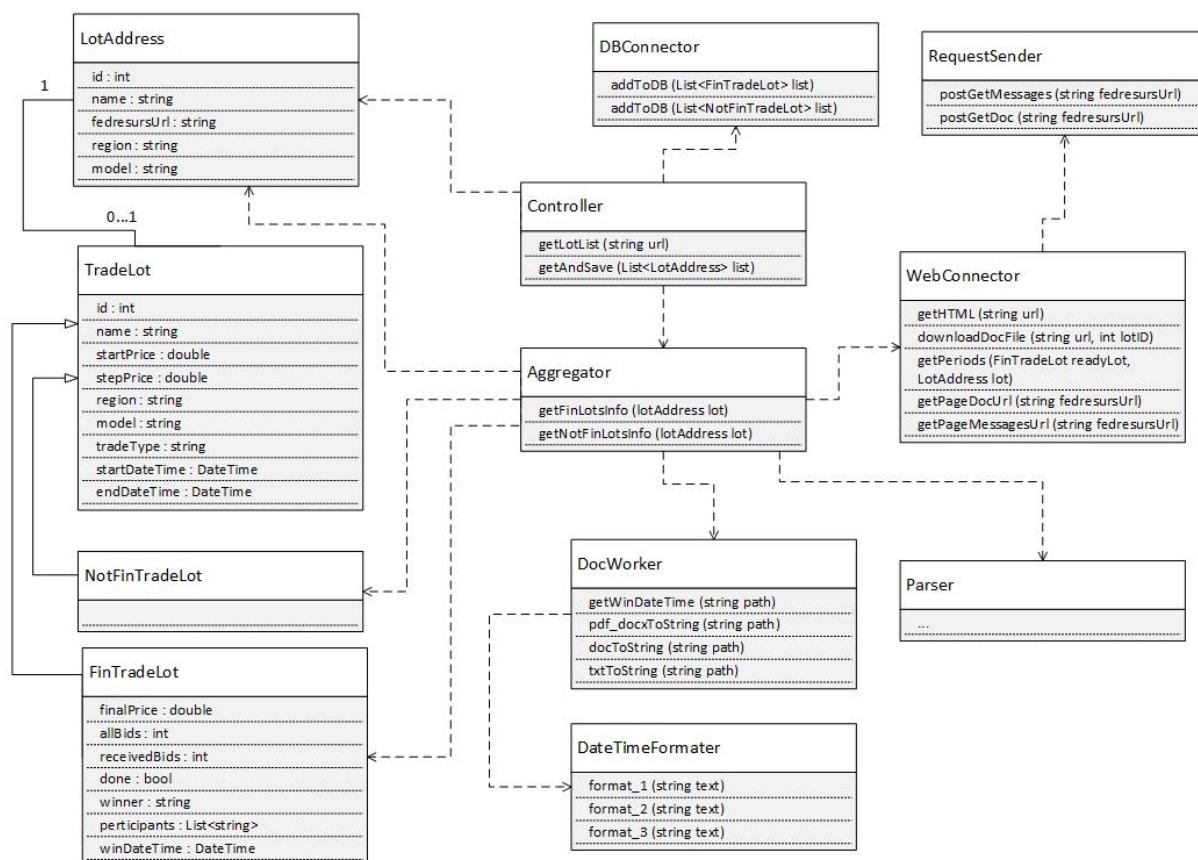


Рис. 8: UML диаграмма классов модуля получения данных.

Сервером bankrot-spy.ru были предоставлены две web-страницы со списками завершившихся и незавершенных торгов. Информация на страницах обновляется один раз в неделю. Был реализован метод ***Controller.getLotsList(string url)***, осуществляющий скачивание данных с этих страниц и сравнение уникальных идентификаторов лотов с теми, которые уже находятся в базе данных. В результате работы метода по каждой из страниц формируется список лотов, которых еще нет в базе данных. В списке содержатся экземпляры класса ***LotAddress***, поля этого класса содержат следующую информацию о лоте: уникальный идентификатор, название, ссылку на федресурс, регион, модель автомобиля, то есть данные, которые были предоставлены сервисом bankrot-spy.ru.

Рассмотрим дальнейшую работу по получению данных уже реализованных лотов.

Метод *Controller.getAndSave(List<LotAddress> list)* осуществляет обход всех элементов полученного ранее списка. Прежде чем начинается обход списка, осуществляется запрос к базе данных с целью получения идентификаторов лотов, которые уже в ней содержатся. Делается это с целью исключения из списка лотов, которые уже были однажды обработаны и помещены в базу данных.

Количество лотов в списке может составлять несколько тысяч, поэтому появляется потребность в распараллеливании обработки лотов. Параллелизм был реализован с помощью метода *Parallel.ForEach()* из пространства имен *System.Threading.Tasks*.

Большое количество лотов в списке, а также перспектива извлечения неструктурированной информации означают, что процесс обработки лотов займет достаточно продолжительное время. Была реализована функция промежуточного сохранения. После обработки каждых 100 лотов происходит запись собранной информации в базу данных. Обработка 100 лотов в среднем занимает 30 секунд.

Во время обхода списка для каждого из его элементов вызывается метод *Aggregator.getFinLotsInfo(lotAddress lot)*, который возвращает экземпляр класса *FinTradeLot* для лотов, торги по которым уже завершились. Для слотов, торги по которым еще не завершились, вызывается метод *Aggregator.getNotFinLotsInfo(lotAddress lot)*, который возвращает экземпляры класса *NotFinTradeLot*.

Классы *FinTradeLot* и *NotFinTradeLot* являются наследниками класса *TradeLot* и широко используются во всех модулях системы. Определение класса *TradeLot* представлено на листинге 2.

Листинг 2: Определение класса *TradeLot*

```
class TradeLot
{
    public int id; //уникальный идентификатор лота,
                  //созданный сервисом bankrot-spy.ru
    public string name; //название лота
    public double startPrice; //начальная цена лота
```

```

public double stepPrice; //шаг аукциона
public string region; //регион проведения торгов
public string model; //модель автомобиля
public string tradeType; //тип торгов
public DateTime startDateTime; //дата и время
                               //начала приема заявок
public DateTime endDateTime; //дата и время
                               //окончания приема заявок
    . . .
}

```

В классе *FinTradeLot* определены дополнительные атрибуты (см. листинг 3).

Листинг 3: Определение класса *FinTradeLot*

```

class FinTradeLot : TradeLot
{
    public double finalPrice; //финальная цена лота
    public int allBids; // общее количество заявок
    public int receivedBids; //количество принятых заявок
    public bool done; //показатель состоялись торги (true)
                               //или нет (false)
    public string winner; // имя победителя
    public List<string> participants; //список имен
                               //участников
    public DateTime winDateTime; //дата и время подачи
                               //победной заявки
    . . .
}

```

Метод *Aggregator.getFinLotsInfo(lotAddress lot)* вызывает следующие методы:

- ***WebConnector.getHTML(string url)***. Используется ***System.Net.WebClient()*** для получения html страницы лота с федресурса, затем, используя метод ***LoadHtml*** библиотеки ***HtmlAgilityPack*** [15] создает ***HtmlDocument*** для страницы лота.
- Статические методы класса ***Parser***, которые осуществляют поиск характеристик лота на html странице и их преобразование. Используются подходы, описанные в разделе 3.1. Если корректно определяются финальная цена и победитель, то полю `done` присваивается значение «true», это означает, что торги по лоту завершились успешно, а значит из сопроводительных документов и сообщений можно получить дополнительную информацию.
- ***WebConnector.getPageDocUrl*** и ***WebConnector.getPageMessagesUrl***. На федресурсе большая часть данных о лоте представлена в виде элемента управления ***RadTabStrip*** — это форма со вкладками. Основная информация находится на первой вкладке, которая открывается автоматически при переходе на страницу лота, но остальные вкладки, на которых хранятся документы и сообщения, открываются кодом, который недоступен для просмотра. Удалось отследить POST запрос, который отправляется на сервер, когда осуществляется переход на другую вкладку ***RadTabStrip***. Статические методы класса ***requestSender*** посылают такие же POST запросы напрямую на сервер, результатом их работы являются html страницы с различными вкладками элемента управления ***RadTabStrip***.

***WebConnector.getPageDocUrl(string fedresursUrl)*** возвращает html страницу, на которой содержится ссылка для скачивания протокола о результатах проведения торгов. ***WebConnector.getPageMessagesUrl (string fedresursUrl)*** возвращает html страницу, на которой содержатся ссылки на сообщения о снижении цены, а также на сообщение о результатах проведения торгов.

- Статические методы класса *Parser* осуществляют поиск на полученных html страницах ссылок на нужные сообщения и ссылок для скачивание необходимого документа.
- *WebConnector.downloadDocFile(string downloadUrl, int lotId)*. Метод класса *WebConnector* осуществляет скачивание файла. К названию он добавляет уникальный идентификатор лота, так как встречаются повторяющиеся названия документов.
- *DocWorker.getWinDateTime(string filePath)*. Данный метод возвращает дату и время подачи победной заявки типа *DateTime*. Для их получения был создан класс *DocWorker*, который включает в себя методы извлечения текста из документов разных форматов, поиск в нем нужной информации и приведение ее к типу *DateTime*. Описание алгоритмов и подходов, использованных при решении этой задачи, представлено в разделе 3.2.
- *Parser.getParticipants(string messageUrl)* осуществляет извлечение имен участников из сообщения о завершении торгов.
- *WebConnector.getPeriods(FinTradeLot readyLot, LotAddress lot)* вызывается, если аукцион проводился в форме публичного предложения. Метод осуществляет поиск на странице лота предложений о снижении цены. Вызывает методы, извлекающие информацию о времени и цене двух предложений, которые были опубликованы непосредственно перед подачей победной заявки. Эта информация записывается в специальную таблицу базы данных и применяется в четвертом модуле.

Экземпляр класса *FinTradeLot*, который является результатом работы метода *Aggregator.getFinLotsInfo(lotAddress lot)*, добавляется в список просмотренных лотов. Добавление лота в список происходит внутри блокирующей конструкции, чтобы избежать конфликтов, ведь обработка лотов ведется параллельно в нескольких потоках.

Далее метод *DBConnector.addToDB(List<FinTradeLot> list)*

осуществляет запись информации о каждом лоте из списка в базу данных в таблицу *FinLots*.

Для еще не реализованных лотов метод *Aggregator.getNotFinLotsInfo(lotAddress lot)* вызывает только *WebConnector.getHTML(string url)* и методы класса *Parser*. Метод *DBConnector.addToDB(List<NotFinTradeLot> list)* осуществляет запись данных в таблицу *NotFinLots*.

База данных содержит 4 таблицы:

- *FinLots*. Поля таблицы повторяют поля класса *FinTradeLot*. В таблице хранится информация о лотах, торги по которым уже завершились.
- *NotFinLots*. Поля таблицы повторяют поля класса *NotFinTradeLot*. В таблице хранится информация о лотах, торги по которым еще не завершились.
- *RegionMatrix*. Содержит текстовое поле «region» и 85 столбцов для хранения дихотомических переменных, согласно выбранному в разделе 4.2.3. методу преобразования категориальных параметров — dummy-кодированию.
- *PublicOffers*. Содержит очищенные данные из таблицы *FinLots*. Имеет поле с уникальным идентификатором и те поля, которые участвуют в обучении модели линейной регрессии: «id», «startPrice», «finalPrice», «region».
- *OffersInfo*. Применяется в 4 модуле. Содержит поля: «startPrice» — начальная цена лота, «lastDeclineTime» — время последнего снижения цены, «lastDeclinePrice» — цена лота после последнего снижения цены, «preDeclineTime» — время предпоследнего снижения цены, «preDeclinePrice» — цена лота после предпоследнего снижения цены, «finalTime» — время подачи победной заявки, «finalPrice» — финальная цена лота.

### 5.1.2. Модуль обработки данных

Прежде чем приступить к анализу данных, нужно произвести их очистку и трансформацию.

Результатом работы модуля получения данных являются заполненные таблицы базы данных *FinLots*, *NotFinLots*, *OffersInfo*. Ограничения таблиц базы данных не допускают повторений лотов, а также пустые названия лотов и их начальные цены. Следовательно, данные о реализованных лотах готовы к отображению в форме графического интерфейса. Работа с данными таблицы *OffersInfo* будет вестись в четвертом модуле. Каждый из еще не реализованных лотов будет содержать прогнозируемую финальную цену, которая будет вычисляться в третьем модуле, задача же этого модуля подготовить данные к этим вычислениям.

В главе 4 уже было сказано о том, что прогноз финальной цены будет осуществляться на основе модели линейной регрессии с начальной ценой и регионом в качестве входных параметров. Модель будет обучаться, тестироваться и работать на лотах, продаваемых на торгах в формате публичного предложения.

Метод *DBConnector.fillPublicOfferTable()* осуществляет заполнение таблицы *PublicOffers* такими лотами из таблицы *FinLots*, которые продавались на торгах типа публичное предложение и которые завершились успешно (т.е. *done = true*). При этом отсекаются лоты с аномальными значениями цен, с отсутствием информации о финальной цене и регионе. Такая очистка данных поможет в будущем избежать подачи на вход модели линейной регрессии ложной информации, ухудшающей ее работу.

Метод *DBConnector.makeRegionMatrixTable()* создает и заполняет таблицу *RegionMatrix* двоичной матрицей, кортеж состоит из названия региона и последовательности целых чисел — нулей и одной единицы, согласно выбранному в разделе 4.2.3. методу преобразования категориальных параметров — dummy-кодированию.



### 5.1.3. Модуль обучения и применения модели линейной регрессии

Метод *DBConnector.getReadyToUseData(ref double[][] input, ref double [] output, ref double[][] testInput, ref double[] testOutput)* извлекает из таблиц данные для построения на их основе выбранной в разделе 4.2.4. модели линейной регрессии. Для этого происходит слияние операцией INNER JOIN таблиц *PublicOffers* и *RegionMatrix*. Метод *randomTest(int count, int percent)* случайным образом выбирает из получившейся таблицы множество записей, размер которого соответствует заданному пользователем проценту, запускается счетчик, и записи под выбранными номерами помещаются в массивы *testInput* и *testOutput*, остальные записи помещаются в массивы *input* и *output*. Массивы *input* и *testInput* содержат целые числа: начальная цена лота, последовательность нулей и единицы, соответствующая региону лота. Массивы *output* и *testOutput* содержат финальные цены лотов.

Метод *Analyzer.multipleLinearRegression(double[][] input, double [] output, double[][] test, ref double[] predictable)* осуществляет обучение и использование модели множественной линейной регрессии с помощью фреймворка для машинного обучения *accord.NET* и возвращает результат для тестовой выборки *test* [13]. В массив *predictable* записываются предсказанные значения для обучающей выборки *input* (см. листинг 4). Прежде чем входные данные подаются методу, осуществляющему обучение модели линейной регрессии, они преобразуются в свои логарифмы методами *logConverter(double[][] data)* и *logConverter(double[] data)*. Метод *expConverter(double[] data)* осуществляет обратное преобразование для результата работы регрессии.

Листинг 4: Фрагмент метода, осуществляющего обучение и применение модели множественной линейной регрессии

```
public static double [] Analyzer.multipleLinearRegression
(double [] [] input, double [] output, double [] [] test,
```

```

                                ref double[] predictable)
{
    . . .
    OrdinaryLeastSquares ols = new OrdinaryLeastSquares();
    MultipleLinearRegression regression =
        ols.Learn(inputs_log, outputs_log);
    double[] resultForTest_log =
        regression.Transform(test_log);
    double[] predicted_log =
        regression.Transform(inputs_log);
    predicted = expConverter(predicted_log);
    double[] result = expConverter(resultForTest_log);
    . . .
    return result;
}

```

Метод *Analyzer.multipleLinearRegression(double[][] input, double [] output, double[][] test, ref double[] predictable)* используется как для тестирования модели, так и для ее апробирования на еще не реализованных лотах.

Фреймворк *accord.NET* предоставляет метод для расчета коэффициента детерминации модели, был реализован отдельный метод для расчета скорректированного коэффициента детерминации, рассмотренного в разделе 4.2.2.

#### 5.1.4. Модуль площадки для экспериментов

Программная реализация площадки для экспериментов представляет собой набор методов класса *experimentConstructor*, вычисляющих номер периода снижения цены, время от последнего снижения цены до подачи победной заявки, разницу между ценой лота после последнего снижения и финальной ценой лота, а также методы, осуществляющие изменение цены после последнего снижения в соответствии с

выбранными пользователем параметрами или процентом. Все данные, необходимые для вычисления перечисленных параметров, содержатся в таблице базы данных *OffersInfo*.

## 5.2. Функционал программной системы

В представленной программной системе существуют следующие функциональные возможности, доступные из графического пользовательского интерфейса:

- Просмотр таблицы с краткой информацией по реализованным лотам с возможностью сортировки по любому параметру;
- Просмотр таблицы с краткой информацией по еще не реализованным лотам, включающей прогнозируемую финальную цену, с возможностью сортировки по любому параметру (см. приложение А);
- Просмотр подробной информации по реализованному лоту, включающей дату и время подачи победной заявки, список участников, прогнозируемую финальную цену лота, которую можно тут же сравнить с реальной финальной ценой и оценить качество прогнозов (см. приложение Б);
- Просмотр информации по любому участнику торгов, включающей количество побед и проигрышей в торгах, а также список всех торгов, в которых он принимал участие;
- Возможность оценить модель линейной регрессии, используемой для прогнозирования финальной цены лота, путем выставления процента тестовой выборки и дальнейшего просмотра результатов прогнозирования на обучающей и тестовой выборках, значения критериев качества линейной регрессии также доступны для просмотра (см. приложение В);
- Возможность «поучаствовать» в искусственно созданных на основе данных о реализованных лотах торгах в форме публичного

предложения, путем повышения цены лота и сравнения своего результата с реальным (см. приложение Г);

- Возможность посмотреть статистику по завершившимся торгам, включающую номер периода снижения цены, во время которого поступила победная заявка, время которое прошло с момента снижения цены до подачи победной заявки.

# Заключение

Результатом данной работы является успешное решение всех поставленных задач, а именно:

1. Разработаны алгоритмы извлечения неструктурированной информации из документов форматов pdf, doc, docx, txt, zip-архивов, а также слабоструктурированной информации из html страниц.
2. Найдены факторы, оказывающие наибольшее влияние на изменение цены лота в ходе торгов.
3. Апробированы и оценены различные модели линейной регрессии, осуществляющие прогнозирование финальной цены лота. Выбрана оптимальная модель.
4. Создано десктопное приложение, функционал которого включает предоставление информации по лотам, прогнозирование цены лотов и платформу для апробирования различных стратегий участия в торгах на уже реализованных лотах.

Поставленная цель была достигнута.

Результаты данной работы были представлены на конференции «СПИСОК 2017».

## Список литературы

- [1] Friedl Jeffrey EF. Mastering regular expressions, 3rd Edition. — "O'Reilly Media, Inc.", 2006. — 554 p.
- [2] K. Benoit. Linear regression models with logarithmic transformations // London School of Economics, London. — 2011.
- [3] Skrivanek Smita. The use of dummy variables in regression analysis // More Steam, LLC. — 2009.
- [4] XML Path Language (XPath) 2.0 (Second Edition) / Anders Berglund, Scott Boag, Don Chamberlin et al. // World Wide Web Consortium (W3C). — 2010.
- [5] А.Г. Дьяконов. Методы решения задач классификации с категориальными признаками // Прикладная математика и информатика. — 2014. — Vol. 46.
- [6] Айвазян С. А. Енюков И. С. Мешалкин Л. Д.: под ред. Айвазяна С. А. Прикладная статистика. Исследование зависимостей: справочное издание. — Финансы и статистика, 1985. — 471 с.
- [7] Айвазян С.А. Мхитарян В.С. Прикладная статистика и основы эконометрики: Учебник для вузов. — 1998. — 656 p.
- [8] Бронников А М. Виды торгов при реализации имущества должников (банкротов) // Сравнительно-правовые аспекты правоотношений гражданского оборота в современном мире. — 2015. — P. 32–37.
- [9] <Кодекс Российской Федерации об административных правонарушениях> от 30.12.2001 N 195-ФЗ (ред. от 17.04.2017) Статья 14.13. <Неправомерные действия при банкротстве> часть 3.
- [10] Риз Р. Обработка естественного языка на Java/пер. с англ // Снастина АВ–М.:–ДМК Пресс. — 2016. — 263 p.

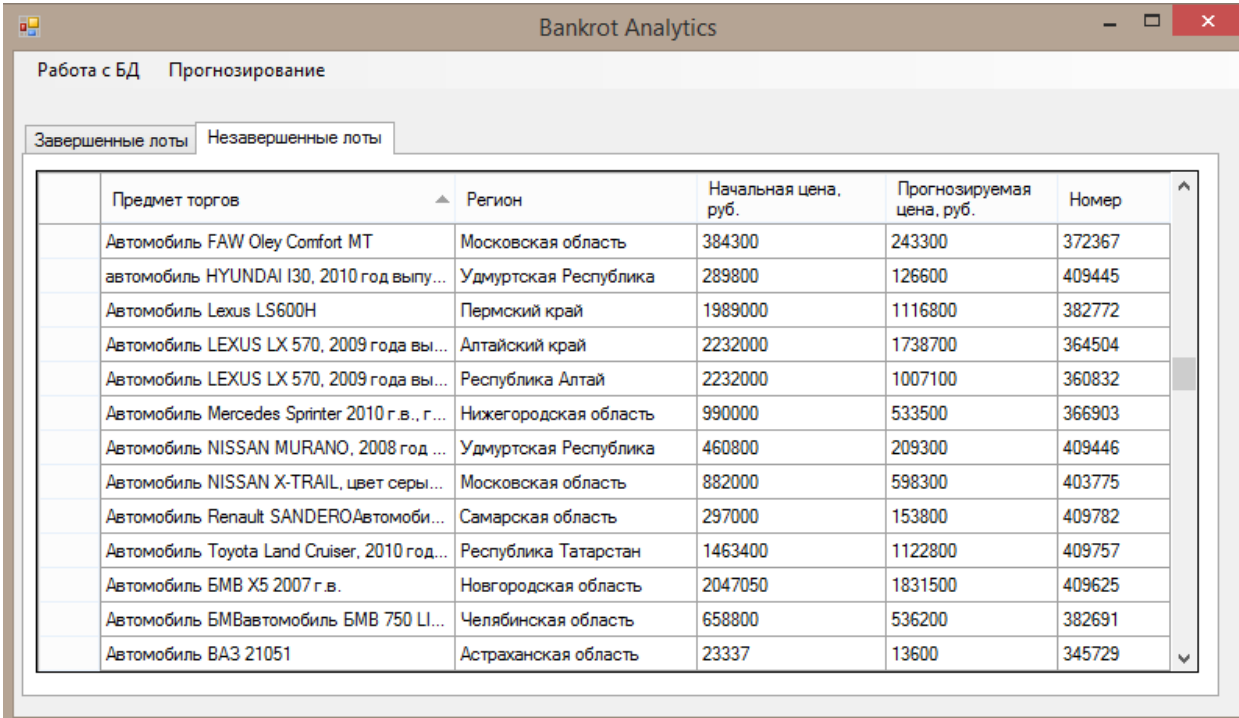
- [11] Сайт ассоциации электронных торговых площадок. — URL: <http://aetp.ru/etp/list> (дата обращения: 20.05.2017).
- [12] Сайт единого федерального реестра сведений о банкротстве. — URL: <https://bankrot.fedresurs.ru> (дата обращения: 20.05.2017).
- [13] Сайт проекта Accord.Net. — URL: <http://accord-framework.net> (дата обращения: 20.05.2017).
- [14] Сайт проекта Deductor. — URL: <https://basegroup.ru/deductor/description> (дата обращения: 20.05.2017).
- [15] Сайт проекта Html Agility Pack (HAP). — URL: <http://html-agility-pack.net/?z=codeplex> (дата обращения: 20.05.2017).
- [16] Сайт проекта Microsoft.Office.Interop.Word. — URL: <https://msdn.microsoft.com/ru-ru/library/microsoft.office.interop.word.aspx> (дата обращения: 20.05.2017).
- [17] Сайт проекта Tika on .Net via IKVM. — URL: <https://kevm.github.io/tikaondotnet/> (дата обращения: 20.05.2017).
- [18] Сайт проекта Банкротный шпион. — URL: <http://bankrot-spy.ru> (дата обращения: 20.05.2017).
- [19] Сайт проекта Современное прогнозирование. — URL: [http://forecasting.svetunkov.ru/forecasting\\_toolbox/models\\_quality](http://forecasting.svetunkov.ru/forecasting_toolbox/models_quality) (дата обращения: 20.05.2017).
- [20] Сайт с рейтингом агрегаторов торгов по банкротству. — URL: <http://rba.company/blog/top-5-agregatorov-aukcionov-po-bankrotstvu.html> (дата обращения: 20.05.2017).
- [21] Федеральный закон от 26.10.2002 № 127 (ред. 3.04.16) <О несостоятельности (банкротстве)>.

- [22] Шевелевич К. Н. Путко Б.А. Эконометрика. — Юнити М., 2002. —  
Р. 74–75.



# Приложение А

Страница с краткой информацией по еще не реализованным лотам



The screenshot shows a window titled "Bankrot Analytics" with a sub-header "Работа с БД Прогнозирование". Below the header are two tabs: "Завершенные лоты" (selected) and "Незавершенные лоты". The main content is a table with the following columns: "Предмет торгов", "Регион", "Начальная цена, руб.", "Прогнозируемая цена, руб.", and "Номер". The table contains 14 rows of data, each representing an auction lot with details on the vehicle, region, and prices.

Предмет торгов	Регион	Начальная цена, руб.	Прогнозируемая цена, руб.	Номер
Автомобиль FAW Oley Comfort MT	Московская область	384300	243300	372367
автомобиль HYUNDAI I30, 2010 год выпу...	Удмуртская Республика	289800	126600	409445
Автомобиль Lexus LS600H	Пермский край	1989000	1116800	382772
Автомобиль LEXUS LX 570, 2009 года вы...	Алтайский край	2232000	1738700	364504
Автомобиль LEXUS LX 570, 2009 года вы...	Республика Алтай	2232000	1007100	360832
Автомобиль Mercedes Sprinter 2010 г.в., г...	Нижегородская область	990000	533500	366903
Автомобиль NISSAN MURANO, 2008 год ...	Удмуртская Республика	460800	209300	409446
Автомобиль NISSAN X-TRAIL, цвет серы...	Московская область	882000	598300	403775
Автомобиль Renault SANDEROАвтомоби...	Самарская область	297000	153800	409782
Автомобиль Toyota Land Cruiser, 2010 год...	Республика Татарстан	1463400	1122800	409757
Автомобиль BMW X5 2007 г.в.	Новгородская область	2047050	1831500	409625
Автомобиль BMWавтомобиль BMW 750 LI...	Челябинская область	658800	536200	382691
Автомобиль ВАЗ 21051	Астраханская область	23337	13600	345729

# Приложение Б

## Страница с подробной информацией по реализованному лоту

16051

**А\м Лада 217130 Р7040В163, 2012г.в.**

Начальная цена, руб.	360000	Регион	Самарская область
Итоговая/текущая цена, руб.	183600	Модель	Лада 217130
Количество поступивших заявок	11	Дата приема победной заявки	22.09.2015 0:00:11
Количество принятых заявок	9	Победитель	Клещев Андрей Юрьевич
Дата начала приема заявок	18.05.2015 0:00:00	Участники	Клещев Андрей Юрьевич
Дата окончания приема заявок	12.11.2015 14:00:00		Буланов Петр Анатольевич
			Шиловский Виталий Анатольевич
Прогнозируемая итоговая цена	189645		Муртузалиев Мурад Магомедшари...
			Зенин Олег Александрович

Искать похожие  По участникам  По региону  По модели  По цене

	Предмет торгов	Регион	Начальная цена, руб.	Итоговая цена, руб.	Количество принятых заявок	Победитель	Вид торгов	Номер
	Автом Лада 21...	Самарская обл...	360000	167500	3	Буланов Петр А...	Публичное пре...	16060
	Автомобиль Ch...	Московская об...	442477,8	231300	4	Бунтиков Евген...	Публичное пре...	91368
	Автомобиль Мl...	Краснодарски...	810000	680000	7	Удальчиков Ал...	Публичное пре...	158198
	Автомобиль Мl...	Краснодарски...	810000	680000	7	Удальчиков Ал...	Публичное пре...	128895

# Приложение В

## Страница для тестирования модели множественной линейной регрессии

Линейная регрессия

Тестовая выборка    Обучающая выборка

	Номер	Начальная цена, руб.	Итоговая цена, руб.	Прогнозируемая цена, руб.	Ошибка, руб.	Ошибка, % от начальной
	258093	269885,59	215908,47	209427,05	6481,42	2,402
	116702	22500	12375	12956,53	581,53	2,585
	51770	74700	29880	31821,33	1941,33	2,599
	195111	972743,38	827777	801266,07	26510,93	2,725
	369183	203674	183400	177033,27	6366,73	3,126
	76048	403200	242000	227832,1	14167,9	3,514
	142044	453007,81	271000	288707	17707	3,909
	206876	1201500	545100	592934,67	47834,67	3,981
	265820	195480	132500	140858,28	8358,28	4,276
	256092	41397,71	28979	27167,18	1811,82	4,377
	153376	392850	319980	301303,58	18676,42	4,754
	145910	303050	210000	201203,58	18676,42	4,754

Процент тестовой выборки

Средняя абсолютная масштабированная ошибка 24%

Скорректированный коэффициент детерминации 0,864

# Приложение Г

## Страница площадки для экспериментов

Публичные предложения

Информация Эксперименты График

Номер	Период снижения	Время от снижения до подачи победной заявки, сек.	Текущая цена, руб.	Финальная цена, руб.	Моя цена, руб.
46224	7	1075682	44960	44960	45002
46226	11	556648	21967	22000	22002
46227	5	1109773	49010	55630	50002
46228	7	1209445	44959	44959	45002
63344	11	204696	32620	32620	33002
63363	8	863650	371800	372000	372002
63367	8	61998	29700	30001	30002
63474	10	224473	35256	37051	36002
63476	10	224865	36630	37060	37002
63510	4	595662	59263	60001	60002
68135	5	22073	211680	212500	212002

Подать менее, чем 5 минут после снижения

Округлить до сотен

Округлить до тысяч

Прибавить 2 рубля

Увеличить стоимость на % начальной

Результат

Вы бы одержали победу в 23 из 89 лотов