МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Южно-Уральский государственный университет (национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра вычислительной математики и высокопроизводительных вычислений

РАБОТА ПРОВЕРЕНА
Рецензент,
/ФИО
« » 2020 г.

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой, к.ф.-м.н., доцент
/Н.М. Япарова
«____»_____2020 г.

«Математическое моделирование и алгоритмизация процессов долгосрочного прогнозирования динамики рыночной стоимости лома черных металлов»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ-090401.2020.679 ПЗ ВКР

Руководитель работы, к. ф.-м. н., доцент
/ С.У. Турлакова
«___ »_____2020 г.

Автор работы Студент
группы КЭ-230
/М.В. Ильиных
«___ »_____2020 г.

Нормоконтролер, к. ф.-м. н., доцент
/С.У. Турлакова
«___ »_____ 2020 г.

Челябинск 2020

АННОТАЦИЯ

Ильиных М.В. Математическое моделирование и алгоритмизация процессов долгосрочного прогнозирования динамики рыночной стоимости лома черных металлов. - ЮУрГУ, ВШЭКН, ФГАО ВО ЮУРГУ (НИУ); 2020, 66 с., 28 ил., библиогр. список - 47 наим.

Данная работа посвящена математическому моделированию и алгоритмизации процессов долгосрочного прогнозирования динамики рыночной стоимости лома марки «3А» в Уральском регионе. Исходные данные содержат стоимость лома с 2015 года по 2019 год, а также данные о факторах, влияющих на стоимость лома. В работе приводится анализ литературных источников за последние несколько лет по данному вопросу. Также рассматриваются актуальные методики прогнозирования временных рядов.

В работе исследуются подходы, которые будут эффективны для решения задачи долгосрочного прогнозирования динамики рыночной стоимости лома черных металлов. Также оцениваются затраты времени, которое уходит на подготовку данных и на создание модели. Построены следующие модели: модель тройного экспоненциального сглаживания, линейная регрессия, случайный лес, градиентный бустинг, нейронные сети.

Программная часть работы была выполнена в среде программирования jupyter notebook на высокоуровневом языке python.

ОГЛАВЛЕНИЕ

1 ОБЗОР ЛИТЕРАТУРЫ ПО ЦЕНООБРАЗОВАНИЮ ЛОМА ЧЕРНЫХ МЕТАЛЛОВ

1.1 Тенденции и перспективы рынка черных металлов

За последние несколько лет можно наблюдать тенденцию к увеличению производительности стали за счет увеличения производства на мощных электродуговых печах. Вместе с ростом производительности увеличивается конкуренция среди металлургических предприятий. В России мощности предприятий черной металлургии заняты практически на сто процентов, что совершенно не похоже на индустрию. В мире этот показатель составляет в среднем около семидесяти процентов. Данный факт связан преимущественно с постоянной работой отечественных металлургов над эффективностью. Металлургические предприятия России имеют легкий доступ к сырью и вкладывают деньги в удешевление обработки и операционную эффективность, за счет чего удается снизить цену и конкурировать на рынке [41].

Примером может служить одна из крупнейших компаний чёрной металлургии НЛМК. В связи с неблагоприятной конъюнктурой на рынке стали в 2016 году, НЛМК пришлось подстраиваться под сложившиеся тяжелые условия и искать другие пути по сохранению уровня рентабельности на прежнем уровне. Расширение вертикальной интеграции и реализация оптимизационных программ позволили повысить рентабельность НЛМК даже на падающем рынке. Себестоимость слябов НЛМК в 2016 году оказалась ниже на 30-40%, чем средний показатель по миру [17]. Таким образом, можно сказать, что НЛМК сохранил и укрепил свои позиции на мировом рынке во многом благодаря рационализации производства, введению новых технологий, максимально снизив себестоимость выпускаемой продукции за 2016 год.

Данный пример демонстрирует важность работы над операционной эффективностью, особенно в текущей ситуации, когда темпы мировой экономики замедляются, а цены на сталь продолжают свое снижение. Биржевая стоимость горячекатаной стали с первого по третий квартал 2018 года не опускалась ниже 600 долларов за тонну. Падение цены началось в четвертом квартале 2018 года,

продолжилось в 2019 и в 2020 году. На 8 мая 2020 года стоимость горячекатаной стали составляет 477 долларов за тонну. В ближайшее время Индия и страны Азиатско-Тихоокеанского региона будут вводить в эксплуатацию новые металлургические мощности, что приведет к росту предложения стали на рынке. Поэтому тенденция к снижению стоимости стали может продолжиться в ближайшие годы.

1.2 Рынок лома черных металлов

Лом черных металлов является одним из основных ресурсов на металлургических предприятиях России и мира. Его использование обусловлено экологически и экономически. Более восьмидесяти процентов от шихты составляет лом при производстве стали на электродуговых печах [40]. Использование большего количества лома позволяет наносить меньший вред экологии, а также сокращает стоимость производства стали.

Объем рынка черных металлов в России за 2019 год составил 19,6 млн. тонн (в 2018 году это значение составляло 20.3 млн. тонн). Наибольшая доля в структуре рынка лома черных металлов принадлежит оборотному лому - 39,5% от всего объема рынка. Больше всего лома было образовано в Уральском ФО.

Импорт лома черных металлов в Россию за 2019 год достиг 1,186 млн. тонн. Импорт лома черных металлов в Россию из стран ЕАЭС за 2019 год достиг 1,181 тыс. тонн, что в свою очередь составило 99,6% от общего объема импорта лома в Россию. Крупнейшим импортером стран ЕАЭС стал Казахстан (96% от объема).

За 2019 год было экспортировано 4,9 млн. тонн лома черных металлов. Экспорт лома черных металлов из России в страны ЕАЭС достиг 953 тысяч тонн, что составило 19,1% от всего объема экспорта лома черных металлов из России. На Беларусь приходится 99% от всего объема экспорта в страны ЕАЭС. Также значительный объем экспорта лома из России приходится на Турцию.

Выделяются следующие основные источники образования ресурсов лома:

1. Отходы, получаемые при производстве и обработке черных металлов.

2. Амортизационный лом, образующийся в процессе ликвидации основных средств, проведения капитальных и текущих ремонтов, а также выбытия сменного оборудования, оснастки, приспособлений и инструмента.

3. Бытовой лом и другие виды лома, получаемые путем сбора.

Актуальность прогнозирования цены на лом черных металлов обусловлена высокой конкуренцией среди металлургических предприятий страны и мира. Чтобы оставаться конкурентоспособным предприятие должно сокращать издержки производства, не ухудшая качества выпускаемой продукции. Как отмечает О. Б. Оглуздина, «... к настоящему времени учеными-экономистами предложено множество вариантов улучшения конкурентных позиций. Используемые подходы можно разделить на две направления. Одно направление ориентировано на использование внешних возможностей и нивелирование рисков конкурентной рыночной среды. Другое, противоположное ему, направление повышения конкурентоспособности концентрирует внимание на внутрисистемном развитии предприятия и раскрытии потенциала аккумулированных ресурсов» [33]. Комбинируя два направления, предприятие сможет активно участвовать в конкурентной борьбе.

Увеличению прибыли предприятия может способствовать уменьшение отношения затрат на выпуск продукции к рыночной стоимости конечной продукции. Если влиять на второе достаточно сложно, то уменьшить затраты на производство можно. Одним из способов это сделать является снижение затрат на покупку сырья. Имея возможность оценивать стоимость лома черных металлов в перспективе, зная возможности складских запасов и план производства на ближайшее время, можно оптимизировать процесс закупки сырья.

Высокую важность прогнозирования цен на металлургическую продукцию подчеркивает А.Г. Маланичев. Он говорит о том, что «для стратегического планирования в горно-металлургических компаниях вместо использования внешнего консенсус-прогноза цен на сталь и сырье целесообразна разработка внутренней системы прогнозирования, основанная на собственном представлении о развитии макросреды и факторах, влияющих на цену» [27]. Идею о том, что

прогнозирование цен на первичное сырье в металлургии необходимо для предприятия, высказывают многие исследователи в научной литературе [20], [25], [27], [33].

1.3 Степень изученности и проработанности проблемы

Результаты прогнозирования цен на сырье для металлургических предприятий можно встретить в работах: И.Д. Тихоновской, Е. М. Крюковой, А.В. Графова, К. А. Семченко, Т. А. Баландиной и других авторов. Авторами рассматриваются различные методы и аспекты прогнозирования.

Можно выделить три основных направления в прогнозировании, которые используются в данных работах: метод экспертных оценок, формализованный метод (регрессионный анализ, скользящее среднее, экспоненциальное сглаживание, математическое моделирование) и комбинация этих двух методов.

Во многих работах представлен именно метод экспертных оценок. Преимуществом этого метода является то, что формализованный подход не может учесть многие факторы. Особенно это касается неожиданных изменений внешних условий, приводящих к резкому изменению цены прогнозируемой продукции. Формализовать подобные изменения очень сложно. К недостаткам метода экспертных оценок можно отнести то, что мнение специалиста в рассматриваемой области является субъективным. Также эксперт может не учитывать ряд факторов, выходящих за пределы области его компетентности. Данную проблему можно решить, используя методы связанные с групповым опросом экспертов. Однако такой подход не всегда возможен.

Формализованный метод можно применять, если у исследователя имеется достаточное количество информации и данных о прогнозируемой величине. Отмечается, что данный подход более эффективен при среднесрочном и долгосрочном прогнозировании.

Как наиболее простой и достаточно эффективный среди формализованных методов можно выделить прогнозирование на основе корреляционно - регрессионного анализа. Данный метод позволяет интерпретировать полученные

результаты, что является преимуществом. На текущий момент одной из последних работ на основе корреляционно-регрессионного анализа является работа И.Д. Тихоновской [40]. На рисунке 1 показаны основные этапы построения данной модели.

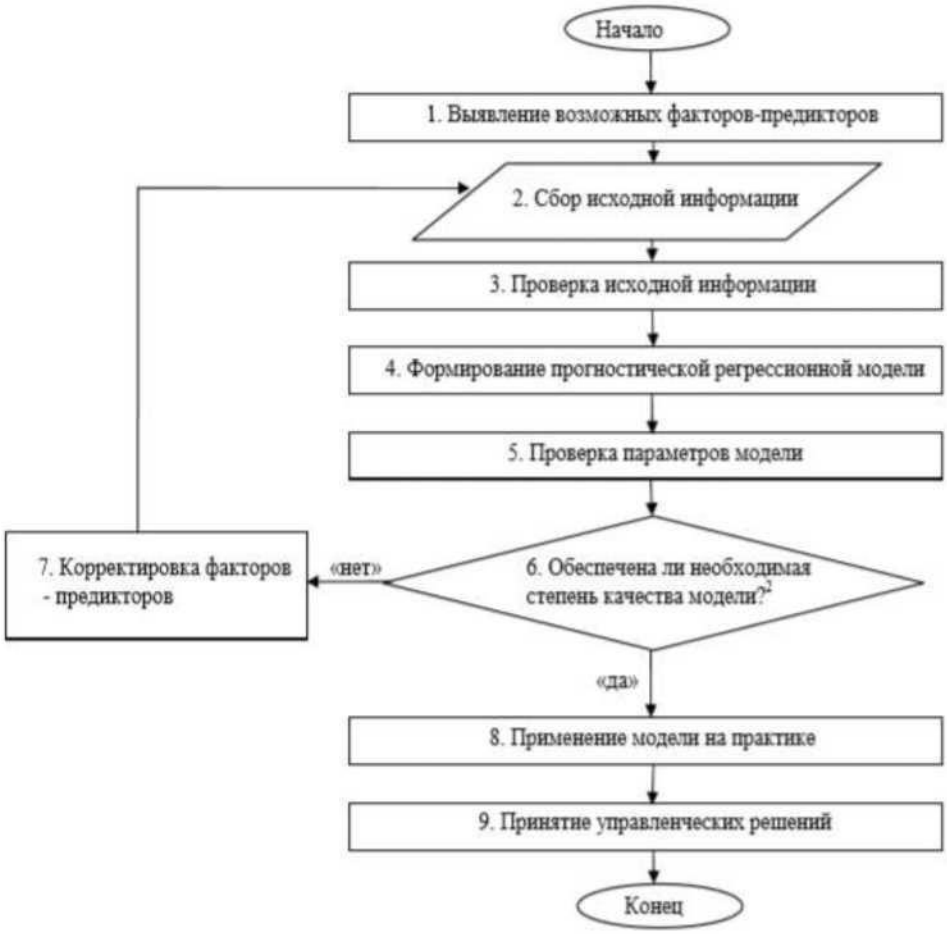


Рисунок 1 - Этапы построения регрессионной модели

В работе И.Д. Тихоновской в качестве предикторов были выбраны следующие данные внешней микросреды:

1. Цена на лом марки «3А» на внутреннем рынке.
2. Цена на арматуру на внутреннем рынке.
3. Экспортная стоимость лома.
4. Цена на передельный чугун.
5. Объемы поставок металлолома на внутреннем рынке.
6. Экспортная стоимость квадратной заготовки.

7. Объем экспорта металлолома.
8. Потребность в ломе на внутреннем рынке.
9. Запасы лома на складах металлургических предприятий в начале месяца.
10. Стоимость лома в Турции.

Предикторы внешней мезо- и макросреды:

1. Курс доллара к рублю.
2. Инфляция.
3. Ставка рефинансирования.

Далее в работе все факторы проверялись на мультиколлинеарность методом анализа корреляционной матрицы. Для построения моделей отбирались наиболее значимые факторы. При отборе факторов так же учитывались причинно-следственные связи. Полученные модели сравнивались.

Кроме того можно выделить работу, не направленную на прогнозирование стоимости лома, но так же ориентированную на снижение затрат при закупке сырья. Авторы работы Т. А. Иванова, В. Ш. Трофимова, А. Н. Калитаев, Д. Г. Степанов занимаются созданием математической модели оптимизации потоков лома черных металлов в РФ. Авторы предлагают рассматривать задачу оптимизации региональной структуры закупа для всех потребителей лома страны как задачу поиска структуры поставок лома потребителям с минимальной суммарной общей стоимостью. Сформулированная задача является классической задачей линейного программирования — транспортной задачей о перевозке грузов [13].

Проблема обеспечения производства сырьем представлена во многих работах [13], [18], [19], [33], [38]. Однако ресурсообеспечение в данных работах рассматривается с точки зрения конкретной отрасли и не всегда подходит для отдельного металлургического предприятия. Некоторые работы рассматривают лишь часть из возможных методов прогнозирования цен на лом черных металлов. Большинство работ было написано несколько лет назад и, возможно, некоторые из подходов уже не актуальны сегодня.

Работ, посвященных прогнозированию именно стоимости лома не так много, поэтому стоит обратить внимание также на работы, где прогнозируются различные временные ряды. Некоторые общие подходы к прогнозированию временных рядов могут быть применимы и к прогнозированию рыночной стоимости лома черных металлов. Активно используются следующие подходы: нейронные сети, модель случайного леса, модели временного ряда (ARIMA и экспоненциальное сглаживание).

В работах [38], [26] задача прогнозирования рыночной стоимости недвижимости решается путем построения полносвязной нейронной сети. В работе Видманта О. С. исследуется возможность прогнозирования цен закрытия волатильного финансового инструмента с использованием специальной архитектуры рекуррентных нейронных сетей (LSTM) [7]. Архитектура сети в данной работе состоит из двух рекуррентных слоев, каждый из которых состоит из 256 нейронов и функции активации Relu. Конечный результат нейронной сети поступает на выходной слой с одним нейроном и линейной функцией активации.

В работе Лемешонок К.А., Ботыгина И.А так же используются рекуррентные нейронные сети с долгой краткосрочной памятью [23]. В качестве области исследования в работе была выбрана метеорология, а именно прогнозирование температуры воздуха. Hansson M. использует сети LSTM для прогнозирования динамики возврата товаров [47]. Roondiwala M., Patel H., Varma S. так же используют сети LSTM, но уже для задачи прогнозировании стоимости акций [46].

В работе В.А. Иванюка предлагается методика совокупного прогнозирования на основе трех методов прогноза: линейного прогноза, асимптотического прогноза и нейронного прогноза [14]. В качестве временного ряда используется курс доллара к рублю. Совокупным прогнозом является объединение трех методик прогнозирования на основе коэффициентов доверия к каждому из трех прогнозов. Коэффициент доверия рассчитывается на основе точности прогноза на исторических данных.

Работа авторов Michael J Kane, Natalie Price, Matthew Scotch, Peter Rabinowitz одна из немногих, где активно защищается позиция случайного леса в задачах по прогнозированию временных рядов. В исследовании применяется модель временных рядов ARIMA и случайный лес к данным о частоте вспышек птичьего гриппа в Египте [45]. Похожий подход использует Степаненко Д.Б., автор разрабатывает гибридную модель прогнозирования временных рядов [37]. В качестве базовых алгоритмов используется ARIMA модель и алгоритм случайного леса. Выбор данных моделей обусловлен тем, что ARIMA модель способна хорошо обрабатывать линейные закономерности, а алгоритм случайного леса позволяет искать более сложные закономерности в данных. Полученная модель тестировалась на специально сгенерированных синтетических данных разного рода.

Можно выделить ряд работ, прогнозирующих значения временного ряда с помощью экспоненциального сглаживания [10], [11], [35], [43], [44]. Подход к подбору оптимальных параметров тройного экспоненциального сглаживания различен. Например, в работе [10] оптимальный коэффициент экспоненциального сглаживания подбирается экспериментально. А в работах [11] и [35] выбор коэффициентов а, в, у осуществлялся с помощью встроенного в программное обеспечение Statistica алгоритма автоматического поиска.

2 МЕТОДЫ ПРОГНОЗИРОВАНИЯ

Существует множество методов прогнозирования. Для каждой конкретной прогнозируемой ситуации необходимо подбирать свой подход. Рассмотрим подробнее методы, которые применяются для прогнозирования стоимостных показателей в работах, представленных в параграфе 1. К ним относятся: методы экспертных оценок, модели временного ряда (AR, MA, ARMA, ARIMA - модели, модели экспоненциального сглаживания), метод регрессионного анализа, алгоритмы на основе деревьев решений, нейронные сети.

2.1 Методы экспертных оценок

Метод экспертных оценок представляет собой прогноз на основе знаний, опыта, интуиции эксперта или группы экспертов. Метод дает возможность учитывать факторы, которые сложно формализовать. Также данный подход не требует обязательного наличия накопленного массива данных, как остальные методы, рассматриваемые в данном разделе.

Прогноз на основе метода экспертных оценок достаточно субъективен, поэтому точность будет сильно зависеть от компетенции эксперта. Решить данную проблему можно привлечением группы экспертов, однако это не всегда возможно и влечет за собой дополнительные расходы.

Одни из самых распространенных методов экспертных оценок:

1. Аналитический метод. Представляет самостоятельную работу эксперта над анализируемой тенденцией.

2. Метод интервью. Представляет беседу эксперта и интервьюера.

3. Метод «Дельфи». Индивидуальный опрос экспертов проводится в форме анкет-вопросников. Особенность в том, что опросы проводятся в несколько туров с последующим обсуждением результатов.

4. Мозговая атака. Организуется как собрание экспертов, на котором обсуждаться различные предложения экспертов.

5. Метод «635» - одна из разновидностей мозговой атаки. Особенность в том, что принимает участие 6 человек, каждый из которых должен записать три идеи в течение пяти минут.

Методов экспертных оценок достаточно много, более подробное описание некоторых из них можно найти в литературе [5], [12], [34].

2.2 Модели временного ряда

Рассмотрим наиболее простые модели для прогнозирования временных рядов. И первая модель - это AR-модель, либо авторегрессионная модель. Это модель, в которой значения временного ряда в данный момент линейно зависят от предыдущих значений этого же ряда. Значение временного ряда в момент времени t прогнозируется с помощью формулы (2.1).

$$X_t = c + e_t + \text{Z?=i } a_t X_t\text{-}i, \qquad (2.1)$$

где p - порядок регрессии; a_1,

$\ldots, a_p$ - параметры модели; c -

постоянная;

$£_t$ - белый шум.

Оценка параметров AR-модели производится с помощью метода наименьших квадратов.

Частным случаем AR-модели может быть SAR-модель. Это модель, в которой текущее значение временного ряда зависит от некоторого предыдущего значения. С помощью SAR-модели можно моделировать сезонные ряды. Значение временного ряда в момент времени t прогнозируется с помощью формулы (2.2).

$$X_t = £_t + a_T X_{t\_T}, \qquad (2.2)$$

где T - период;

a_m - параметр модели;

$£_t$ - белый шум.

Вторая модель - это MA-модель или скользящее среднее. Это модель, в которой значение функции каждой точки равны среднему значению исходной функции за предыдущий период. Значение взвешенного скользящего среднего в точки t определяется с помощью формулы (2.3).

$$X_t = {}^{E}?o\,{}^{w}t\text{-}i \bullet Pt\text{-}i, \qquad (2.3)$$

где w_{t-i} - нормированные веса;

p_{t-i} - значение исходной функции в момент времени, отдаленный от текущего на i интервалов.

Частным случаем MA-модели может быть SMA-модель или арифметическое скользящее среднее, когда все веса равны единицы. Также иногда применяют линейно взвешенное скользящее среднее, когда веса линейно убывают или экспоненциально взвешенное скользящее среднее, когда веса убывают экспоненциально.

Скользящее среднее хорошо подходит для сглаживания колебаний, выделения тенденций, сезонности и циклов. Поэтому данная модель больше применяется для анализа, а не прогнозирования.

Если объединить AR-модель и MA-модель, то получиться ARMA-модель. Такая модель может интерпретироваться как линейная модель множественной регрессии, в которой в качестве объясняющих переменных выступают прошлые значения самой модели, а в качестве регрессионных остатков — скользящее среднее из элементов белого шума. Общий вид данной модели определяется формулой (2.4).

$$X_t = {}^{c} + £_t + £f=!\ aiX_t\text{-}i + £f=!\ Pi£_t\text{-}i, \qquad (2.4)$$

где $a_x,\ldots,a_v$ - параметры модели (коэффициенты авторегрессии);

$, — ,Pq$ - параметры модели скользящего среднего; p_{t-i} - значение исходной функции в момент времени, отдаленном от текущего на i интервалов.

с - постоянная;

£$_t$ - белый шум.

ARMA-модель является более сложной по структуре, чем AR и MA-модели, однако обычно обладает меньшим количеством параметров, что является преимуществом. Минусом ARMA-модели является то, что она применима только для стационарных рядов, то есть если в нашем ряде есть тренд, то ARMA-модель в чистом виде использовать нельзя.

Для работы с нестационарными рядами применяется ARIMA-модель. Даная модель схожа с ARMA, отличие в том, что в ARIMA-модели рассматриваются приращения. Общий вид данной модели определяется формулой (2.5).

$$\boldsymbol{A^d X_t = c + £_t} + £?{=}i\ \boldsymbol{a_i A^d X_{t-i}} + £?{=}i\ fe\text{-}, \qquad (2.5)$$

где $\boldsymbol{A^d}$ - оператор разности временного ряда порядка d;

$\boldsymbol{Pi,\ a\ i}$ - параметры

модели; с - постоянная;

£$_t$ - белый шум.

В представленных моделях мы всегда закладываем некоторую определённую закономерность, некоторую структуру наших данных. Поэтому с помощью данных моделей легко прогнозировать, особенно делать долгосрочные прогнозы, что является несомненным плюсом таких моделей, как ARMA и ARIMA. Однако из-за того, что мы предполагаем какую-то четкую типизацию наших данных, возникает ряд проблем. Во-первых, это оценка параметров. Когда приходят новые данные, нам заново надо оценивать эти параметры, и это может быть очень трудоемкой задачей. Из-за большого количества параметров становится сложно их интерпретировать. Во-вторых, если мы предполагаем какую-то четкую структуру, то при изменении структуры данных модель становится неустойчивой.

Часть описанных проблем могут решить адаптивные модели, а именно модели экспоненциального сглаживания.

Модель простого экспоненциального сглаживания (или модель Брауна) определяется формулой (2.6).

$$9t = {}^{a} * \text{Yc-i} + (1 - a) * \% \qquad (2.6)$$

где a - сглаживающий фактор, принимает значение в диапазоне [0;1).

В отличие от взвешенного среднего в данном методе взвешиваются все наблюдения. Веса экспоненциально уменьшаются по мере углубления в исторические данные. Чем выше значение сглаживающего фактора, тем меньший вес придается историческим значениям. Модель является интуитивно понятной и простой, в ней всего один параметр. Среди минусов можно выделить то, что данная модель плохо работает на данных, в которых присутствует тренд либо сезонность, либо и тренд, и сезонность.

Моделью, которая может учесть тренд, является модель двойного экспоненциального сглаживания. Двойное экспоненциальное сглаживание определяется формулой (2.7).

$$\text{Yc+i} = h + bt, \qquad (2.7)$$

где l_t - составляющая уровня;

b_t - составляющая тренда.

Составляющие уровня и тренда определяются формулами (2.8) и (2.9) соответственно.

$$l_t = a * y_t + (1 - a) * (l_{t-1} + b_t\text{-}i) \qquad (2.8)$$

$$b_t = P * (k - lt\text{-}i) + (1 - \text{Ю} * b_t\text{-}i, \qquad (2.9)$$

где P - коэффициент, принимает значение в диапазоне [0;1).

В результате получаем набор функций. Первая описывает уровень - он, как и прежде, зависит от текущего значения ряда, а второе слагаемое теперь разбивается на предыдущее значение уровня и тренда. Вторая отвечает за тренд - он зависит от изменения уровня на текущем шаге, и от предыдущего значения тренда. Наконец, итоговое предсказание представляет собой сумму модельных значений уровня и тренда.

Для прогнозирования на несколько значений вперед можно использовать формулу (2.10)

$$9t+n = It + n * b_t , \quad (2.10)$$

где l_t - составляющая уровня;

b_t - составляющая тренда;

n - период прогноза.

Моделью, которая может учесть тренд и сезонность является модель тройного экспоненциального сглаживания или модель Хольта - Винтерса. Модель описывается следующей системой уравнений [44]:

$$1_X \quad a\ 1 \quad a)(l_x\text{-}i + b_x—1^{);} \quad (2.11)$$

$$Ь_X = №x - Ix\text{-}l) + (1 - ЮЬ_{x\text{-}1}; \quad (2.12)$$

$$S_X = У (т^) + (1\text{-} Y)S_x\text{-}L; \quad (2.13)$$

$$Ух+m \ (S_X + ™b_x^S_x\text{-}L+m, \quad (2.14)$$

где a, fi, y - коэффициенты сглаживания ряда;

1_X - составляющая уровня;

$Ь_x$ - значение тренда;

s_x - сезонная составляющая;

$Ух+m$ - функция прогноза на m шагов;

L - период сезонности.

Коэффициенты a, fi, y выступают в роли весов в экспоненциальном сглаживании. Итоговый прогноз представляет собой сумму всех трех компонент. Преимуществом данного метода является возможность прогнозировать на более чем один шаг вперед. Для этого есть коэффициент m, отвечающий за шаг прогноза.

2.3 Метод регрессионного анализа

Метод регрессионного анализа исследует зависимость определенной величины от другой величины или нескольких других величин. Общий вид линейной регрессионной модели можно представить в виде уравнения (2.15).

$$\boldsymbol{Y} = \text{«o} + \text{£}\Gamma=1\text{^} \boldsymbol{-Xi} + \boldsymbol{£}, \quad (2.15)$$

где Y - зависимая переменная;

$\boldsymbol{Xi}$ - независимая переменная;

$\boldsymbol{a_1, \ldots, a_n}$ - коэффициенты линейной регрессии;

$\boldsymbol{s}$ - случайная ошибка.

Коэффициенты линейной регрессии находятся с помощью метода наименьших квадратов.

Следует отметить, что регрессионный подход позволяет не только выявлять зависимости между признаками, но и решает задачи прогнозирования (например, временного ряда на основе ретроспективных данных), а также задачи классификации (например, путем использования кривой регрессии в качестве разделяющей плоскости между классами) [15, с 81].

Ограничения для построения линейной регрессии: зависимая переменная может линейно аппроксимировать независимые переменные, отсутствие избытка влиятельных наблюдений, отсутствие мультиколлинеарности между признаками. Не все ограничения должны выполняться полностью, чтобы построить пригодную модель.

Часто при работе с линейной регрессией может возникать переобучение. Переобучение - использование для решения задачи слишком простой или слишком сложной модели [5, с 159]. Переобучение выражается в том, что качество при тестировании значительно хуже, чем во время обучения модели.

Для борьбы с переобучением в линейной регрессии используется L1 и L2 регуляризация. Регуляризация - это способ уменьшить сложность модели, чтобы предотвратить переобучение или исправить некорректно поставленную задачу.

Ll-регуляризация штрафует весовые значения добавлением суммы их абсолютных значений к ошибке. Функция, которую будет необходимо минимизировать для нахождения коэффициентов линейной регрессии с L1-регуляризацией, представлена в формуле (2.16).

$$\text{^i} = \boldsymbol{liiyi - 9i)^2 +} \quad (2.16)$$

где $\boldsymbol{yi}$ - истинное значение прогнозируемой величины;

У; - прогнозное значение;

$\boldsymbol{a_t}$ - коэффициенты линейной регрессии;

$\boldsymbol{a}$ - размер штрафа.

L2-регуляризация выполняет аналогичную операцию добавлением суммы их квадратов к ошибке. Функция, которую будет необходимо минимизировать для нахождения коэффициентов линейной регрессии с L2-регуляризацией, представлена в формуле (2.17).

$$\text{^}_2 = {}^{L}\boldsymbol{i(ji - 9i)^2 + a\,\text{^}a^2,} \quad (2.17)$$

где $\boldsymbol{yi}$ - истинное значение прогнозируемой величины;

$\boldsymbol{9i}$ - прогнозное значение;

$\boldsymbol{a_t}$ - коэффициенты линейной регрессии;

$\boldsymbol{a}$ - размер штрафа.

2.4 Алгоритмы на основе деревьев решений

Дерево решений - один из методов автоматического анализа данных. В основе этого метода лежит использование ориентированного дерева как связного ациклического графа [15, с 62]. Деревья решений можно использовать в задачах регрессии и классификации.

Дерево решений имеет одну вершину (корень), а завершается вершинами с нулевой степенью исхода (из них не исходит ни одна дуга) - листьями. При этом принято, что дерево решений растет вниз.

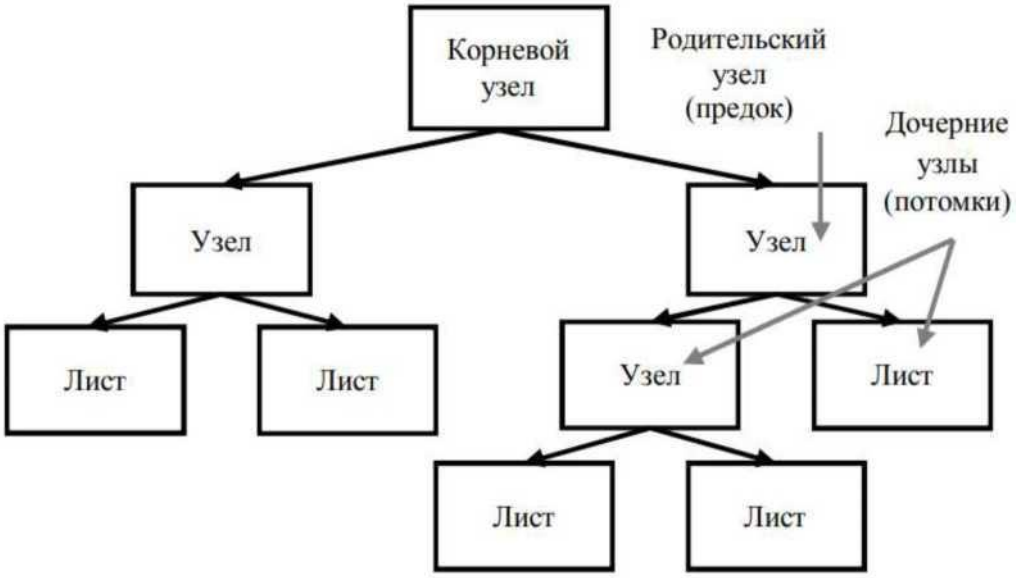


Рисунок 2 - Примеры ориентированного дерева

Подробный процесс построения деревьев решений описан в [15].

Использование деревьев решений в сравнении с другими методами прогнозирования имеет несколько достоинств:

1. Данные не требуют существенной предварительной обработки.
2. Алгоритм способен работать как с категориальными, так и с вещественными переменными.
3. Имеется возможность интерпретации результатов работы модели.

Недостатком модели является то, что деревья решений можно применять только к задачам интерполяции. Для задачи экстраполяции необходимо либо использовать другие алгоритмы, либо преобразовывать исходные данные.

На практике гораздо чаще применяются не деревья решений, а алгоритмы, основанные на деревьях решений. Это модели случайного леса и градиентного бустинга. Алгоритмы представляют собой ансамбль из простых алгоритмов. Такие модели показывают более высокую точность.

Случайный лес - это множество решающих деревьев. В задаче регрессии их ответы усредняются, в задаче классификации принимается решение голосованием по большинству. Особенность построения данного алгоритма в том, что для создания одного дерева используется случайное множество объектов обучающей

выборке. Также на каждой подвыборке выбирается случайный набор признаков для построения. Схема построения случайного леса представлена на рисунке 3.

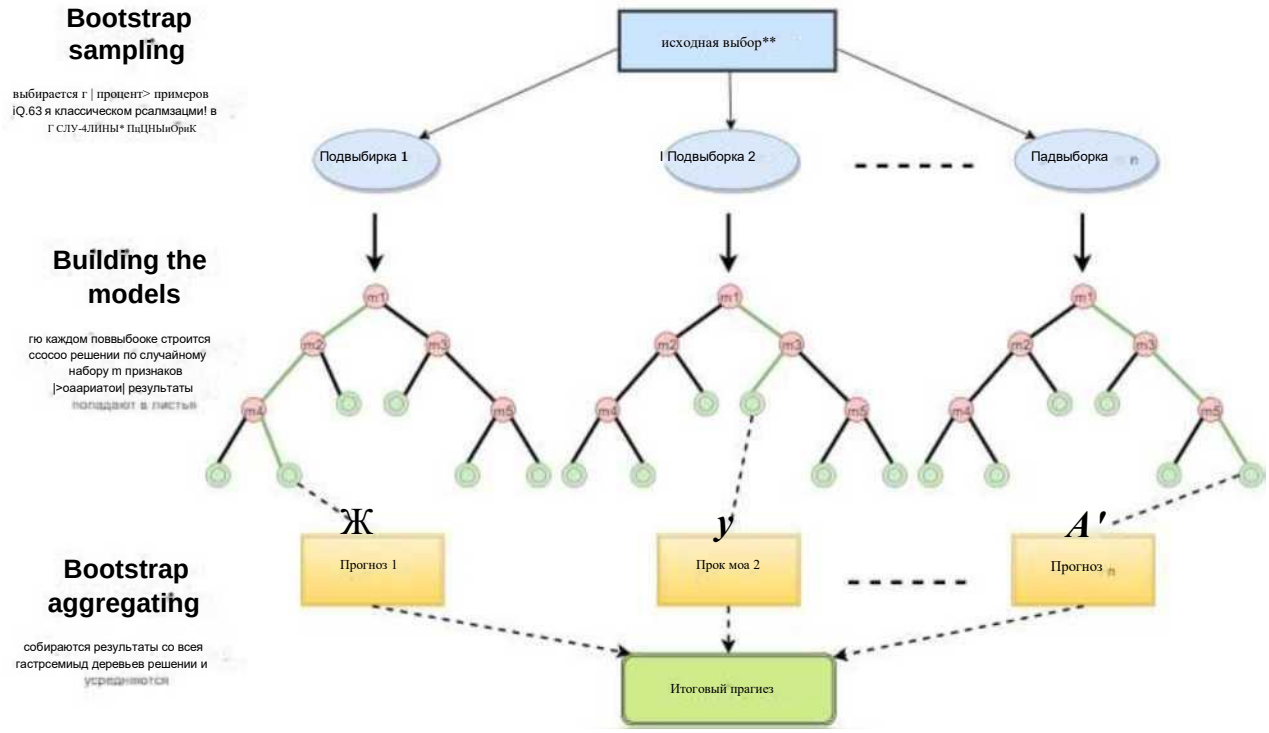


Рисунок 3 - Примеры построения случайного леса

В алгоритме градиентного бустинга каждое новое дерево строится так, чтобы минимизировать ошибку на предыдущих деревьях.

Нет рекомендаций, когда лучше использовать алгоритм случайного леса, а когда алгоритм градиентного бустинга. Для каждой задачи стоит пробовать оба алгоритма и останавливаться на том, который показал лучшее качество.

Для построения алгоритмов на основе деревьев решений необходимо подбирать гиперпараметры модели, рассмотрим основные из них:

1. Число деревьев. Чем больше деревьев, тем лучше качество, но время настройки и работы модели также пропорционально увеличиваются.

2. Число признаков для выбора расщепления. При увеличении данного параметра увеличивается время построения леса, а деревья становятся «более однообразными». Рекомендуемое значение данного параметра составляет треть от

числа признаков в задачах регрессии и корень от числа признаков в задачах классификации.

3. Минимальное число объектов, при котором выполняется расщепление. Этот параметр, как правило, не очень важный и можно оставить значение по умолчанию равное 2. При увеличении параметра качество на обучении падает, а время построения модели сокращается.

4. Ограничение на число объектов в листьях. Параметр так же не сильно важен. Рекомендуется ставить значение равное 1.

5. Максимальная глубина деревьев. Чем меньше глубина, тем быстрее строится и работает модель.

2.5 Нейронные сети.

Нейронная сеть - это последовательность нейронов, соединенных между собой синапсами. В основе нейронной сети находится нейрон. Нейрон - это вычислительная единица, которая получает информацию, производит над ней простые вычисления и передает ее дальше. Они делятся на три основных типа: входной, скрытый и выходной [32].

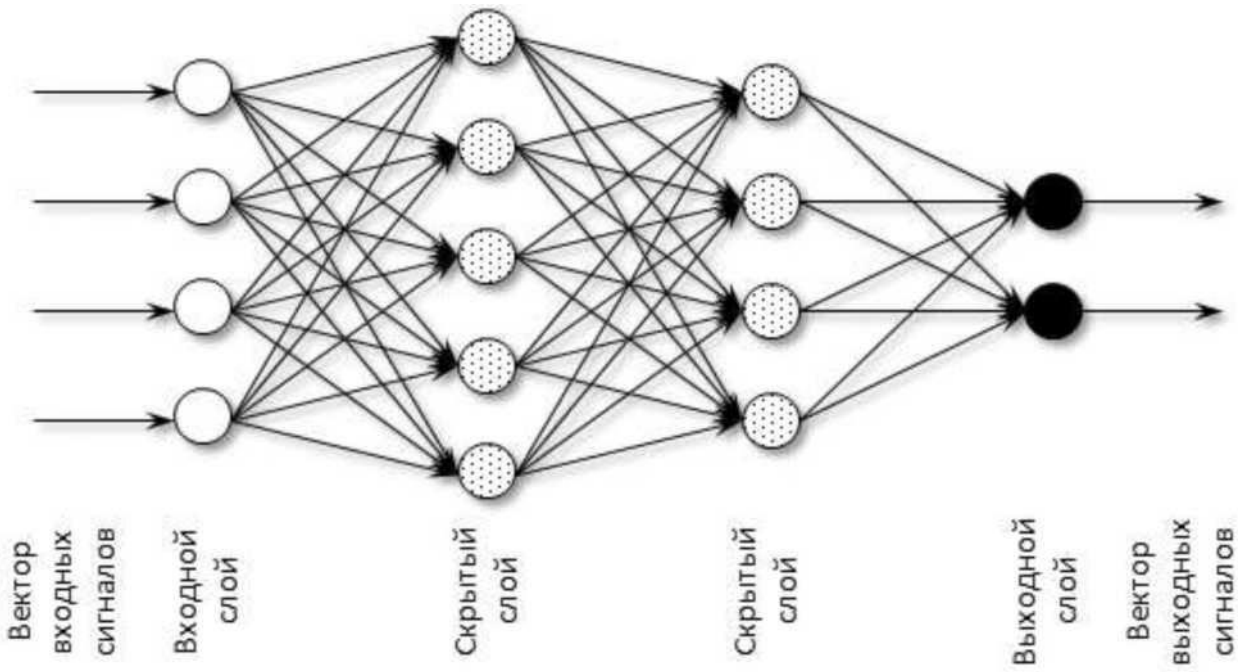


Рисунок 4 - Примеры нейронной сети

Синапс - это связь между двумя нейронами. У синапсов есть 1 параметр - вес. Благодаря ему входная информация изменяется, когда передается от одного нейрона к другому.

Функция активации - это способ нормализации входных данных. На вход функции активации поступает взвешенная сумма нейронов. Функций активации достаточно много, наиболее распространённые это: сигмоид, relu и tanh. Выбор зависит от используемой архитектуры и решаемой задачи.

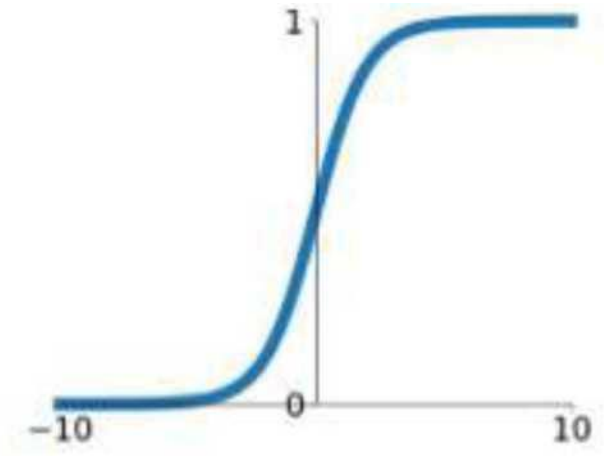


Рисунок 5 - Сигмоидная функция

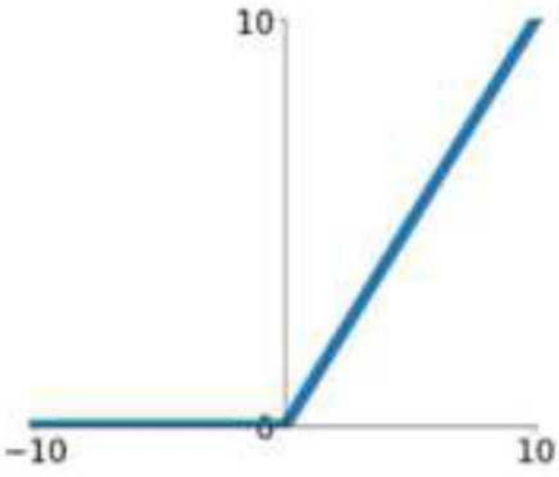


Рисунок 6 – Функция relu

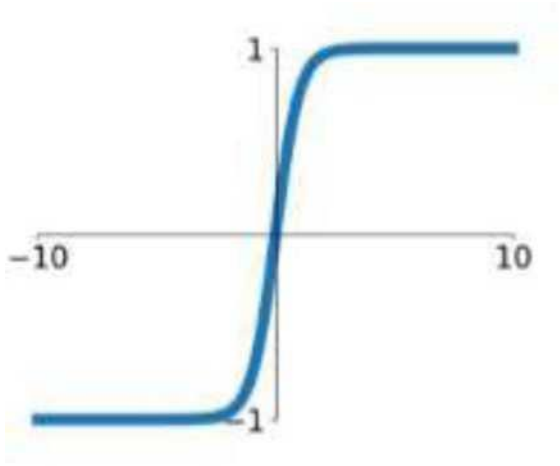


Рисунок 7 – Функция tanh

Функция relu используется для преобразования отрицательных значений в ноль (рисунок 4), а сигмоидная функция рассредоточивает произвольные значения по интервалу [0, 1] (рисунок 5), возвращая значения, которые можно интерпретировать как вероятность [42].

На сегодняшний день существует множество архитектур нейронных сетей для решений различных задач. Их можно условно разделить на нейронные сети прямого распространения и рекуррентные нейронные сети. В сетях прямого распространения сигнал перемещается от входного слоя к выходному. Движение сигнала в обратном направлении не осуществляется и, в принципе, невозможно. Рекуррентные нейронные сети позволяют выходному сигналу возвращаться на вход, таким образом, сигнал двигается и в прямом, и в обратном направлении. Такой архитектуре нейронных сетей присуща функция кратковременной памяти, на основании чего сигналы восстанавливаются и дополняются во время их обработки.

Более подробно с особенностями построения нейронных сетей можно ознакомиться в учебниках [32, 42]. Рассмотрим наиболее эффективную архитектуру рекуррентных нейронных сетей - LSTM (сети долгой краткосрочной памяти). Сеть LSTM используют в данных, где соседние точки зависят друг от друга, и эту зависимость нельзя игнорировать [32, с 232].

Любая рекуррентная нейронная сеть имеет форму цепочки повторяющихся модулей нейронной сети (рисунок 6).

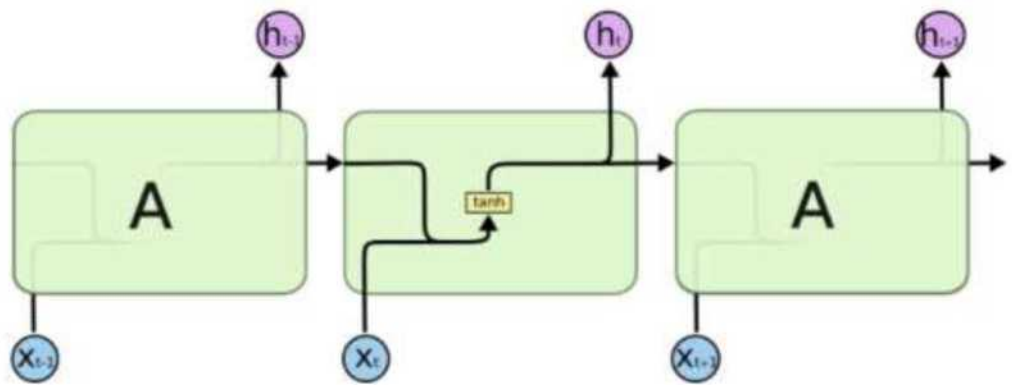


Рисунок 8 - Повторяющийся модуль в стандартной рекуррентной сети

Проблема такой архитектуры в том, что она плохо справляется с ситуациями, когда нужно что-то «запомнить» надолго. Влияние текущего состояния на последующие состояния экспоненциально затухает.

Структура LSTM также напоминает цепочку, но модули выглядят иначе. Вместо одного слоя нейронной сети они содержат целых четыре (рисунок 7).

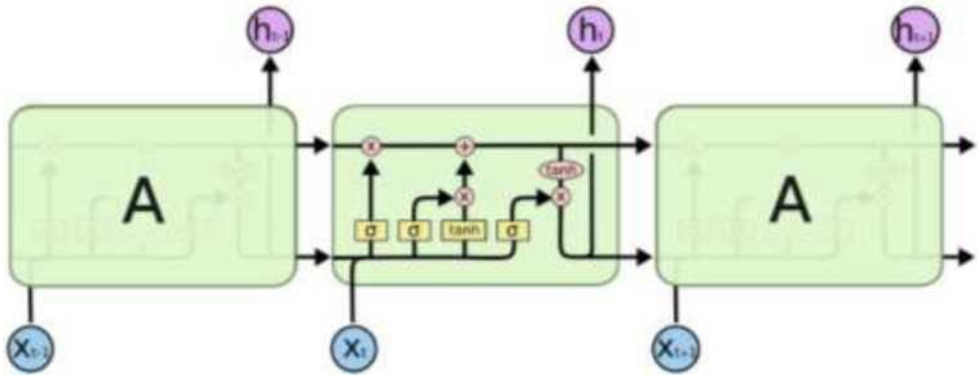


Рисунок 9 - Повторяющийся модель в LSTM сети Ключевой компонент LSTM - горизонтальная линия (состояние ячейки), проходящая по верхней части схемы. Состояние ячейки проходит напрямую через всю цепочку, участвуя лишь в нескольких линейных преобразованиях. Информация может легко течь по ней, не подвергаясь изменениям. Тем не менее,

LSTM может удалять информацию из состояния ячейки. Этот процесс регулируется структурами, называемыми фильтрами.

3 ПОСТРОЕНИЕ МОДЕЛЕЙ

Целью нашей работы является исследование различных алгоритмов и моделей для прогнозирования динамики рыночной стоимости лома черных металлов. Необходимо найти подходы, которые будут эффективны для решения данной задачи. Также нужно оценить сложность используемых алгоритмов. Сложность будем оценивать по затратам времени, которое уходит на подготовку данных и на создание модели.

Набор моделей, который будет исследован, выбран исходя из моделей, которые используются для прогнозирования различных временных рядов в современной литературе. Также были выбраны те модели, которые предположительно должны давать наилучшее качество при прогнозировании. Это модель тройного экспоненциального сглаживания, линейная регрессия, случайный лес, градиентный бустинг, нейронные сети.

3.1 Описание исходных данных

В нашем распоряжении имеются данные о стоимости лома марки 3А уральского региона России за период с 2015 года по 2019 год. Стоимость лома представлена без учета НДС, в рублях за одну тонну сырья.

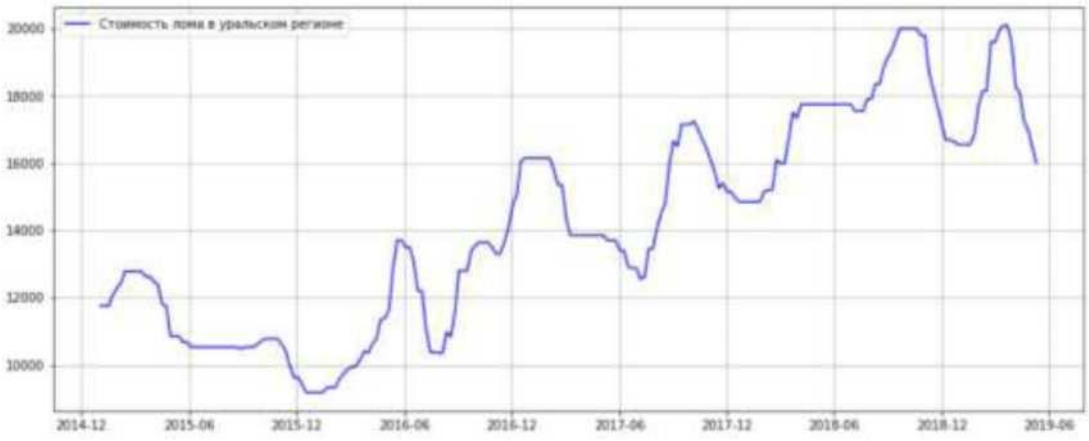


Рисунок 10 - Стоимость лома марки «3А» в уральском регионе

Также были собраны данные, которые, по мнению специалистов, являются наиболее значимыми в ценообразовании лома уральского региона. Это данные по

стоимости лома в других регионах России (центральный и южный), стоимость лома в Турции, курс доллара, маржинальность стальной продукции на крупнейших предприятиях страны. Все эти данные фиксировались каждую неделю указанного периода. Также в нашем распоряжении есть данные, которые фиксировались помесячно. Это объем выплавки стали в России, Китае, ЕС и в мире, данные о промышленной инфляции, объем производства основной стальной продукции в России, а также запасы лома на предприятиях.

В качестве выборки для обучения возьмем данные с 2015 года по 2018 год. В качестве тестовой выборки будут использованы данные за 2019 год. Данные представим в виде единой таблицы, где каждая строка соответствует одной неделе. В качестве периода прогнозирования было выбрано восемь недель вперед. Данного времени достаточно, чтобы оптимизировать процесс закупки лома.

Программная реализация алгоритма осуществлена на языке программирования Python в среде программирования jupyter notebook. Выбор данного языка программирования связан с наличием большого количества готовых библиотек для анализа данных, находящихся в открытом доступе.

Все признаки представлены вещественным типом данных с небольшим количеством пропусков. Для дальнейшего построения моделей пропуски были заполнены путем подстановки предыдущего значения того или иного признака.

Для предварительного анализа имеющихся признаков построим диаграмму корреляций.

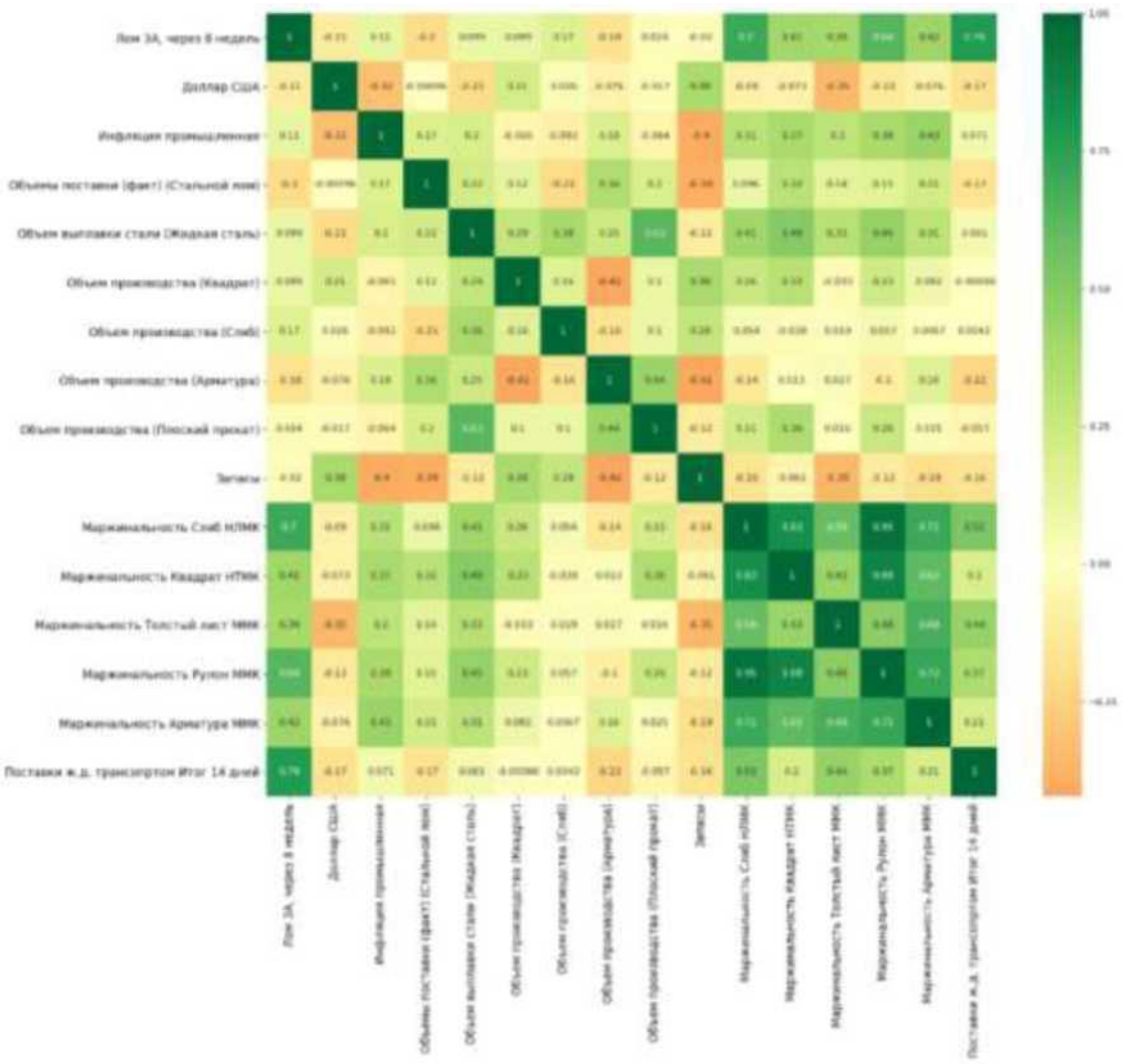


Рисунок 11 - Диаграмма корреляций

Наибольшую корреляционную зависимость целевой признак имеет с маржинальностью слябов на предприятии НЛМК, маржинальностью рулона на предприятии ММК и с объемами поставок лома железнодорожным транспортом за последние две недели. Также видно, что маржинальность различной стальной продукции имеет достаточно сильную корреляционную зависимость друг с другом.

Для дальнейшего построения моделей необходимо будет использовать признак текущего значения стоимости лома. На момент прогноза известна стоимость лома в уральском, южном и центральном регионе, а также стоимость лома в Турции. Построим диаграмму корреляций для данных признаков.

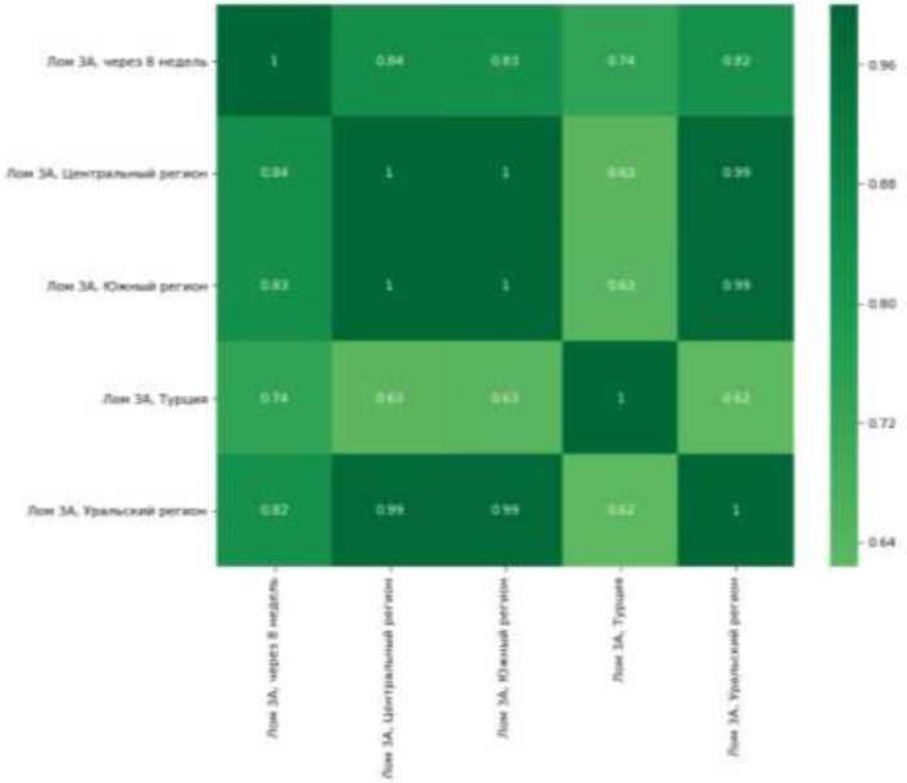


Рисунок 12 - Диаграмма корреляций

Видно, что наибольшую корреляционную зависимость целевая переменная имеет со стоимостью лома в центральном регионе. Возможно, это связано с тем, что стоимость лома в уральском регионе, как правило, подстраивается с небольшим опозданием под стоимость лома в центральном регионе. Частично это можно наблюдать на рисунке 11.

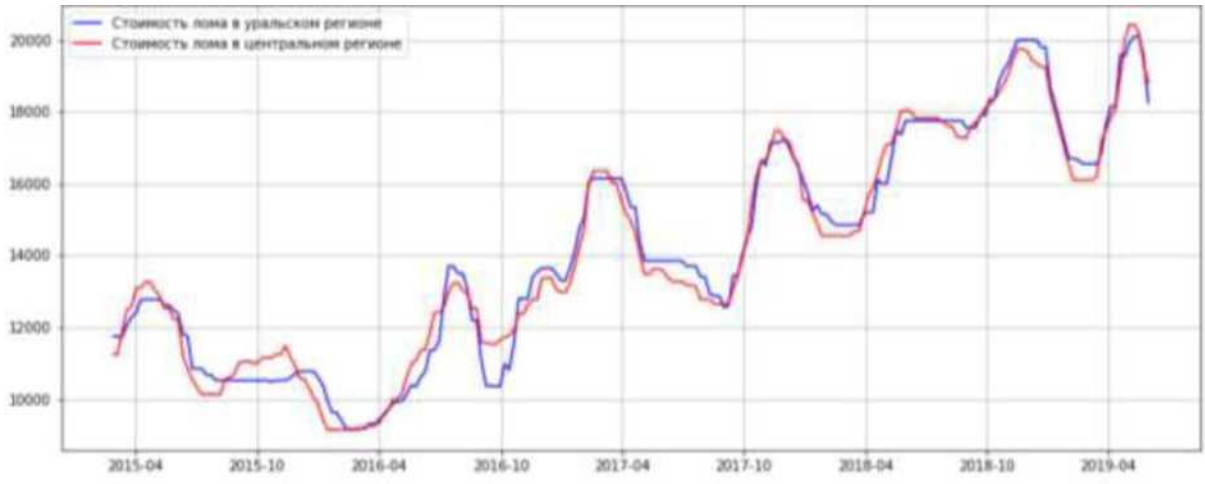


Рисунок 13 - Сравнение стоимости лома в центральном и уральском регионе

Из этого следует, что в дальнейшем в качестве признака предыдущей стоимости лома стоит отдавать предпочтение цене в центральном регионе.

3.2 Базовый алгоритм

Прежде чем начать использовать различные модели машинного обучения для решения задачи прогнозирования стоимости лома, рассмотрим подход, который используют специалисты по закупке лома на текущий момент. Он поможет провести базовую линию, которую мы должны будем превзойти, чтобы доказать преимущество более сложных моделей машинного обучения. При отсутствии мнения эксперта о конкретном изменении стоимости лома, специалисты по закупкам используют наивное предположение, что стоимость лома через два месяца будет такой же, как сейчас. Для оценки данного подхода воспользуемся метрикой средней абсолютной ошибки (Mean Absolute Error, MAE), которая находится по формуле (3.1).

$$MAE = \quad y,- yd, \tag{3.1}$$

где N - количество прогнозов;

yi - истинное значение прогнозируемой величины;

ЗУ - прогнозное значение.

Также для сравнения моделей воспользуемся метрикой средней абсолютной ошибки в процентах (Mean Absolute Percentage Error, MAPE), которую будем находить по формуле (3.2).

$$MAPE = \overline{X'N} \quad \frac{y\sim}{Si} \tag{3.2}$$

где N - количество прогнозов;

yi - истинное значение прогнозируемой величины;

ЗУ - прогнозное значение.

Сравним результат работы базового алгоритма с фактическим значением стоимости лома. Средняя абсолютная ошибка на тестовой выборке составляет

2100 рублей. Ошибка в среднем составляет 12% от истинного значения. Максимальное отклонение составляет 23%.

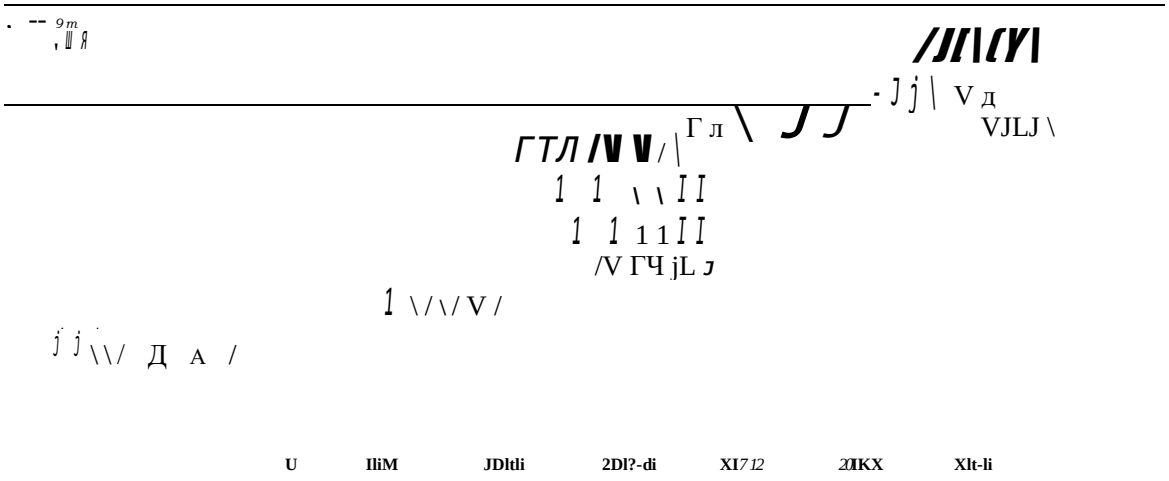


Рисунок 14 - Результат работы базового алгоритма на всей выборке

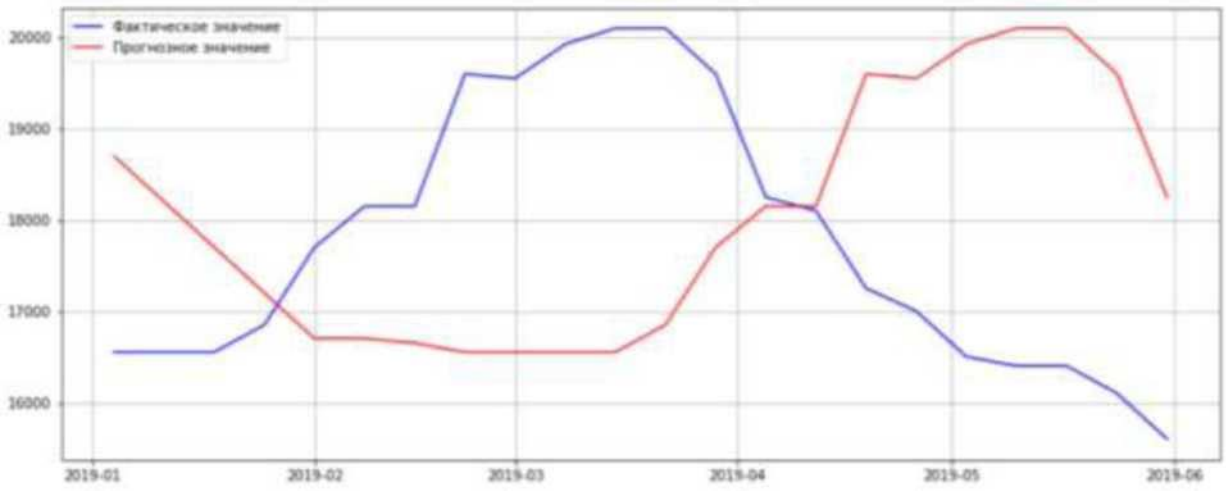


Рисунок 15 - Результат работы базового алгоритма тестовой выборке

3.3 Тройное экспоненциальное сглаживание

Листинг программы модели тройного экспоненциального сглаживания представлен в приложении 1. Модель представлена в виде класса Holt Winters. При инициализации класса необходимо передать следующие параметры: исходный временной ряд (series), коэффициенты экспоненциального сглаживания (alpha, beta, gamma) и горизонт прогноза (n_preds). Далее для построения прогноза необходимо вызвать метод triple_exponential_smoothing. Для отдельного расчета

тренда и сезонной компоненты есть методы initial_trend и initial_seasonal_components соответственно.

В модели необходимо подобрать оптимальные значения параметров ***a,*** Д, ***y.*** Подбор параметров можно делать вручную, фиксируя поочередно два коэффициента и настраивая третий. Данный подход прост, но не дает возможности найти лучшую комбинацию коэффициентов.

Условимся, что будем подбирать такие параметры ***a,*** Д, ***y*** чтобы прогноз был максимально точным на два месяца вперед. Данный период прогнозирования наиболее полезен при планировании закупок лома. Для того чтобы подобрать оптимальные параметры необходимо решить оптимизационную задачу.

Если разделить все наши данные на обучение и тест, то велика вероятность переобучения. Так как в этом случае подбираемые параметры будут сильно зависеть от выбранного нами разбиения. Чтобы избежать этой проблемы, воспользуемся кросс-валидацией.

Суть кросс-валидации в том, что все данные делятся на некоторое количество частей. Сначала обучаем модель на данных с первого наблюдения до наблюдения t, а тестируем на данных, начиная с t до t+n. Далее к обучению добавляем тестовые данные, а тестируем на данных от t+n до t+2*n и т.д. Таким образом, большое количество данных участвует в обучении и тестировании, что снижает риск переобучения.

В качестве функции потерь была выбрана средняя квадратичная ошибка (Mean Squared Error, MSE), которая находится по формуле (3.3):

$$\boldsymbol{MAE} = £Zi=i(yi - У;)^2 \qquad (3.3)$$

где N - количество прогнозов;

yi - истинное значение прогнозируемой величины;

У; - прогнозное значение.

Воспользуемся библиотекой sciру, а именно модулем optimize для оптимизации функции потерь. В качестве алгоритма оптимизации был выбран

Truncated Newton conjugate gradient. Функция, для которой необходимо найти минимум, находит среднее значение квадратичной ошибки от всех этапов кросс-валидации. Алгоритму необходимо решить задачу безусловной оптимизации, так как на искомое решение не налагается никаких дополнительных условий, кроме того, что оно должно доставлять минимум заданной функции.

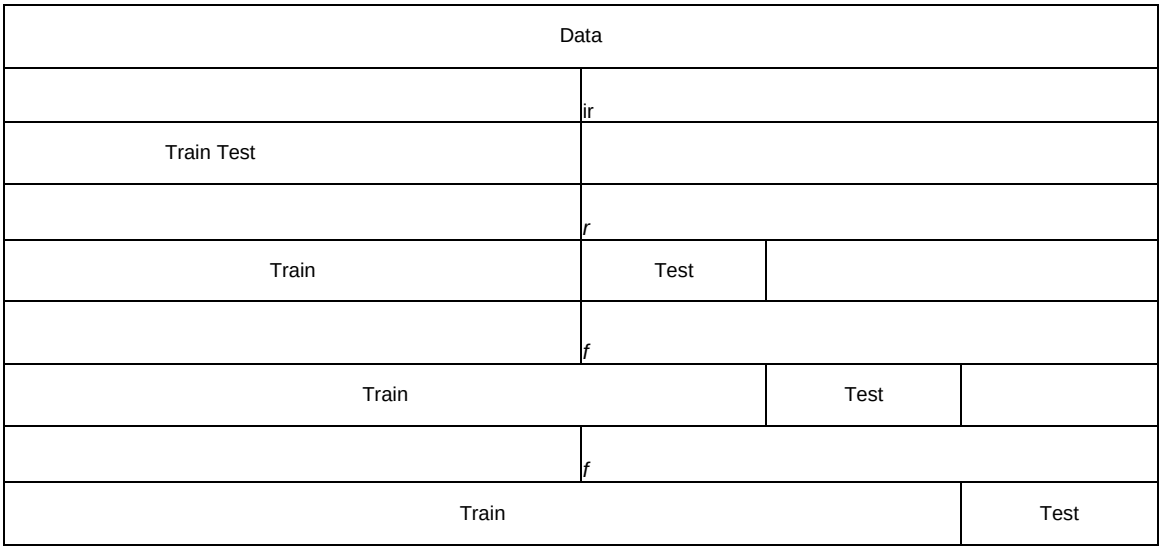


Рисунок 16 - Кросс-валидация временного ряда

Как правило, для перекрестной проверки рекомендуется разбивать выборку на 3 - 5 частей. Здесь и далее в работе будем разбивать выборку на три части ввиду того, что размер выборки всего 230 объектов и частое разбиение может ухудшить качество модели. В результате при помощи кросс-валидации были найдены оптимальные коэффициенты: $\boldsymbol{a} = 0.01396$, Д = 0.0314, $\boldsymbol{y} = 0$. Коэффициент, отвечающий за сезонность, принимает нулевое значение, это может говорить об отсутствии выраженной сезонности в данных. Листинг программы подбора оптимальных значений экспоненциального сглаживания представлен в приложении 2.

После того как были подобраны оптимальные параметры для модели тройного экспоненциального сглаживания необходимо оценить качество работы алгоритма. Был построен прогноз на отложенных данных.

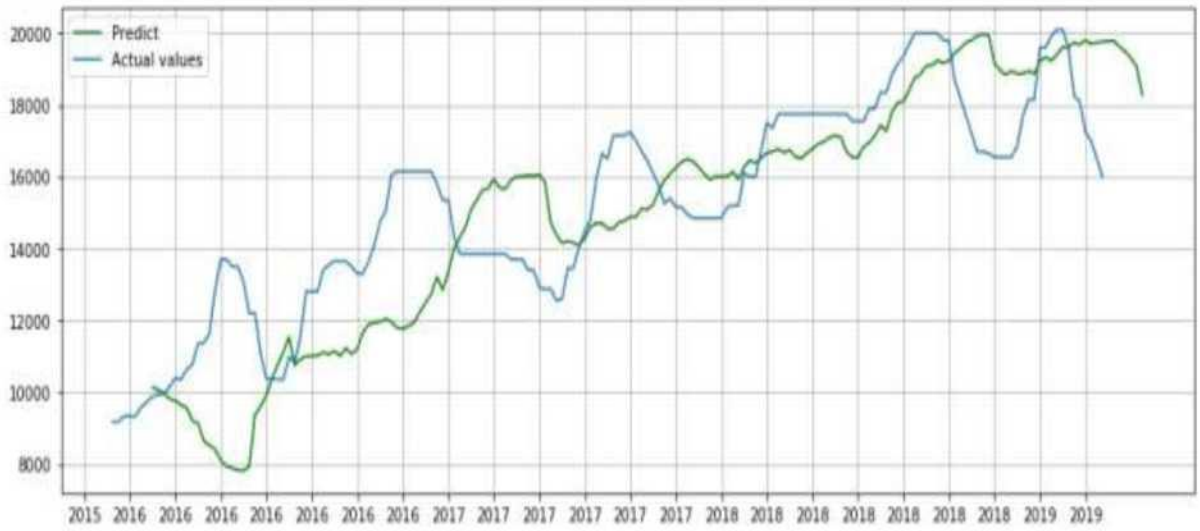


Рисунок 17 – Прогноз модели

Средняя абсолютная ошибка на тестовой выборке составляет 1831 рубль. Ошибка в среднем составляет 10,9% от истинного значения. Максимальное отклонение составляет 25,7%. Стандартное отклонение 984 рубля. Коэффициент детерминации равен 0,395.

Качество работы алгоритма оказалось лучше, чем качество базового алгоритма. Соответственно, метод тройного экспоненциального сглаживания может использоваться для решения данной задачи. Метод не требует предварительной подготовки данных. Процесс построения модели и подбора параметров прост, если использовать готовые решения (например, программный модуль Statistica).

3.4 Линейная регрессия

Модель тройного экспоненциального сглаживания позволяет учесть уровень, сезонность и тренд, но не позволяет учитывать различные экономические показатели, которые оказывают существенное влияние на ценообразование лома. Самый простой способ учесть имеющиеся у нас факторы - это построение линейной регрессии.

В качестве признаков для построения линейной регрессии были использованы следующие факторы: маржинальность листа, рулона и арматуры на ММК;

информация о поставках лома железнодорожным транспортом за последние две недели; объемы поставок стального лома; предыдущие значения цены лома. Данные признаки были отобраны на основе корреляционной зависимости с целевой переменной, а также на основе мнения экспертов о важности того или иного признака. Построим диаграмму корреляций для данных признаков:

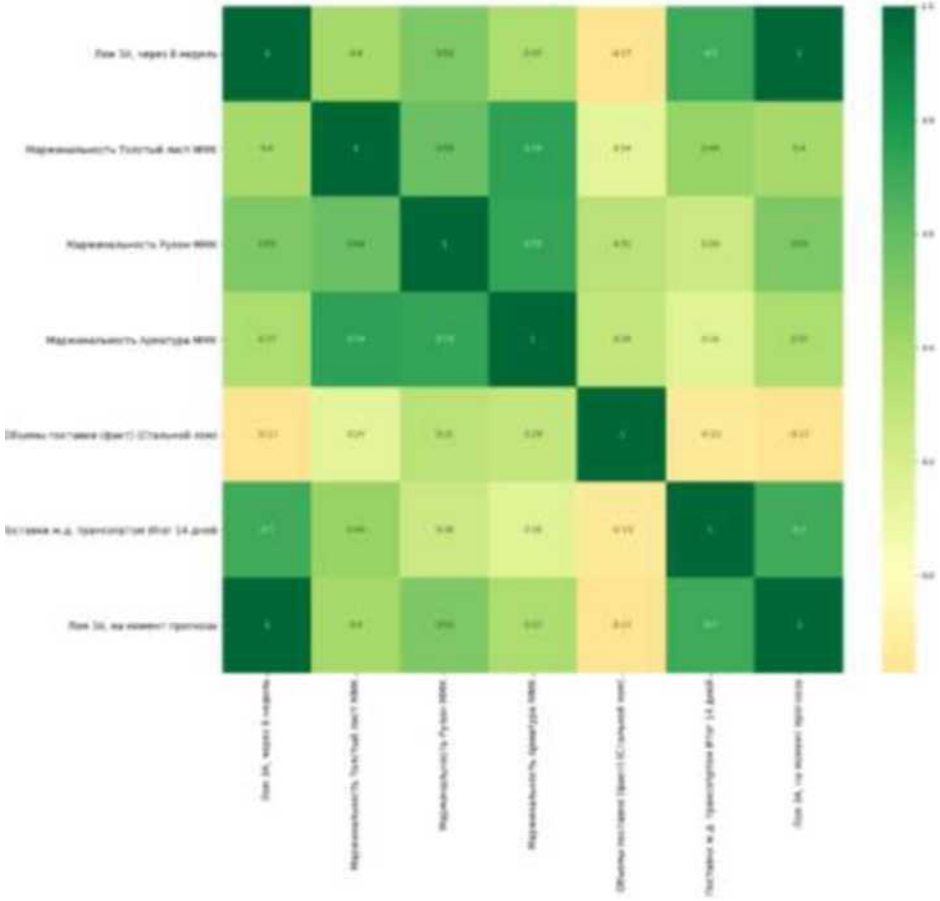


Рисунок 18 - Диаграмма корреляций

Для построения линейной регрессии была использована библиотека skleam. Данные были предварительно нормализованы. Так как в выборке присутствует признаки сильно коррелирующие друг с другом, то необходимо использовать L_1 или L_2 регуляризацию.

Начнем с L_1 регуляризации. В таком случае функция, которую будет необходимо минимизировать для нахождения коэффициентов линейной регрессии, будет выглядеть следующим образом:

$$Li = liiyi - 9i)^2 + «ZikL \quad (3.4)$$

где yi - истинное значение прогнозируемой величины;

З>i - прогнозное значение;

a_t - коэффициенты линейной регрессии;

a - размер штрафа.

Размер штрафа будем подбирать с помощью кросс-валидации. Для этого используем уже готовую функцию LassoCV() в модуле linear_model библиотеки skleam. Данная функция реализует модель линейной регрессии с L_1 регуляризацией.

Принцип работы кросс-валидации в обычной задачи регрессии схож с тем, который был рассмотрен ранее для задачи прогнозирования временного ряда. Разница состоит в том, что тестовая выборка не обязательно следует сразу после обучающей, а может находиться в разных частях выборки. Принцип работы кросс-валидации представлен на рисунке 17.

Лучшим параметром на кросс-валидации оказался размер штрафа равный 1.

Iteration 1	Test	Train	Tram	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Tram	Train	Test	Tram
Iteration 5	Train	Train	Train	Tram	Test

Рисунок 19 - Кросс-валидация

Далее построим модель с L_2 регуляризацией. В таком случае функция, которую будет необходимо минимизировать для нахождения коэффициентов линейной регрессии, будет выглядеть следующим образом:

$$\text{^2} = \boldsymbol{T,i(yi - 9i)^2 + a\text{^}af,} \quad (3.5)$$

где $\boldsymbol{yi}$ - истинное значение прогнозируемой величины;

У; - прогнозное значение;

$\boldsymbol{a_t}$ - коэффициенты линейной регрессии;

$\boldsymbol{a}$ - размер штрафа.

Размер штрафа будем подбирать с помощью кросс-валидации. Для этого используем уже готовую функцию RidgeCV() в модуле linear_model библиотеки skleam. Данная функция реализует модель линейной регрессии с L_2 регуляризацией. Лучшим параметром на кросс-валидации оказался размер штрафа равный 1.

Библиотека sklearn также позволяет реализовать одновременно L_1 и L_2 регуляризацию. Для этого есть функция ElasticNetCV(), в которой заложена возможность перебора параметров по сетке. Функция, которую будет необходимо минимизировать для нахождения коэффициентов линейной регрессии, в этом случае выглядит следующим образом:

$$\text{^12} = \boldsymbol{liiyi - yi)^2} + \quad + 0.5 * \boldsymbol{a} * (1 - Ю£;я^2, \quad (3.6)$$

где $\boldsymbol{yi}$ - истинное значение прогнозируемой величины;

у* - прогнозное значение; $\boldsymbol{a_t}$ - коэффициенты

линейной регрессии; $\boldsymbol{a}$ - размер штрафа;

$\boldsymbol{P}$ - баланс между L_1 и L_2 регуляризацией.

В результате сравнения трех способов регуляризации наилучший показатель средней абсолютной ошибки получился у L_1 регуляризации. Поэтому остановимся именно на линейной регрессии с L_1 регуляризацией.

В результате модель представляет следующее уравнение: y = 2962.7 * ***X1*** + 589.8 * ***X2*** - 2070.1 * X3 + 222.3 * ***X4*** + 728 * X5 + 4522.8 ***X6*** + 14863,

где y - стоимость лома через 8 недель (прогнозная величина);

X1 - маржинальность листа на ММК;

X2 - маржинальность рулона на ММК;

X3 - маржинальность арматуры на ММК;

X4 - объем поставок стального лома;

X5 - Поставки лома ж/д транспортом за последнее две недели;

X6 - Стоимость лома 8 недель назад.

Как можно видеть из уравнения, наибольшее влияние оказывает предыдущая стоимость лома и маржинальность продукции одного из основных потребителей лома в России - ММК.

Построим прогноз на отложенных данных.

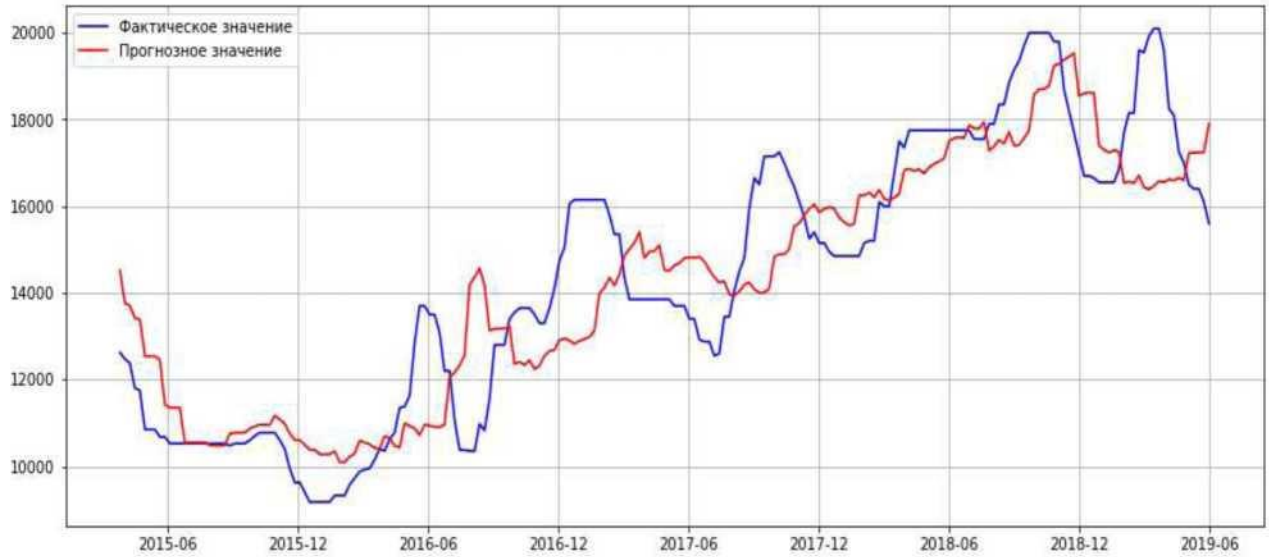


Рисунок 20 - Прогноз модели

Средняя абсолютная ошибка на тестовой выборке составляет 1739 рублей. Ошибка в среднем составляет 9% от истинного значения. Максимальное отклонение составляет 21%. Стандартное отклонение 1223 рубля. Коэффициент

детерминации равен 0,07. Статистическая надежность и значимость многофакторной регрессионной модели (или коэффициента детерминации) устанавливается с помощью F-критерия Фишера. Расчетное значение F- статистики равняется 2,74. Табличное значение при доверительной вероятности 0,05% равняется 2,47. Расчетное значение выше табличного, соответственно, модель признается статистически надежной и значимой.

Дальнейшая идея по улучшению модели линейной регрессии состоит в том, чтобы добавить в качестве признака прогноз, полученный с помощью модели тройного экспоненциального сглаживания.

В результате модель представляет следующее уравнение: y = 2599 * ***X1 + 125*** * ***X2*** - 1772.8 * X3 + 777.7 * ***X4*** - 1898.8 * X5 + 3396.4 *X6 + 3345*X7 + 14855,

где y - стоимость лома через 8 недель (прогнозная величина);

X1 - маржинальность листа на ММК;

X2 - маржинальность рулона на ММК;

X3 - маржинальность арматуры на ММК;

X4 - объем поставок стального лома;

X5 - Поставки лома ж/д транспортом за последнее две недели;

X6 - Стоимость лома 8 недель назад;

X7 - стоимость лома прогнозируемая с помощью тройного экспоненциального сглаживания.

Результаты работы объединённой модели:

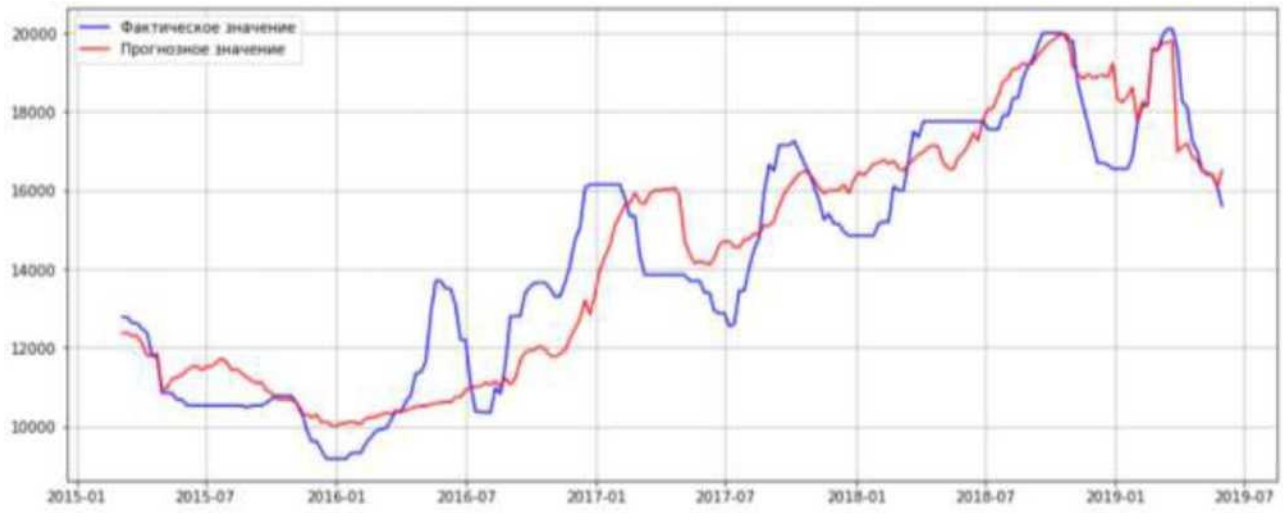


Рисунок 21 - Прогноз модели

Средняя абсолютная ошибка на тестовой выборке составляет 627 рублей. Ошибка в среднем составляет 3,5% от истинного значения. Максимальное отклонение составляет 13%. Стандартное отклонение 405 рубля. Коэффициент детерминации равен 0,558. Расчетное значение F-статистики равняется 42. Табличное значение при доверительной вероятности 0,05% равняется 2,32. Расчетное значение значительно выше табличного, соответственно модель признается статистически надежной и значимой. Модель линейной регрессии требует предварительной работы с данными и проста в реализации.

3.5 Случайный лес

Случайный лес - алгоритм, в основе которого лежат деревья решений. Данный алгоритм прост и эффективен в задачах регрессии.

Суть метода состоит в построении большого количества деревьев решений на основе различных частей обучающей выборки. Результат получается путем усреднения значения ответа на полученных деревьях:

$$\boldsymbol{\Phi)} = \text{^£f } {}^{\boldsymbol{T}}\boldsymbol{i(^{x})}, \tag{3.7}$$

где M - число деревьев в модели;

$\boldsymbol{Tj}$ - дерево решений.

Г лавным недостатком алгоритма является то, что его можно применять только к задачам интерполяции, коей не является задача прогнозирования стоимости лома. Поэтому для применения случайного леса выборка была изменена. Вместо стоимости будет прогнозироваться изменение стоимости. А стоимостные признаки были заменены на разницу между прошлым значением и текущим. Признаки, используемые в модели: маржинальность листа, рулона и арматуры на ММК; информация о поставках лома железнодорожным транспортом за последние две недели; объемы поставок стального лома; предыдущие значения цены лома.

Для построения модели воспользуемся библиотекой skleam. Модуль ensemble предоставляет готовую реализацию случайного леса в виде функции RandomForestRegressor. Для подбора параметров модели обучающая выборка была разбита на три части. Оптимальные параметры для алгоритма были подобраны с помощью кросс-валидации, которая реализована в модуле model_selection в виде функции GridsearchCV.

Оптимальные параметры получились следующими:

1) количество деревьев в лесу: 30;
2) максимальная глубина дерева: 5;
3) минимальное количество выборок, необходимое для разделения внутреннего узла: 2;
4) минимальное количество выборок должно быть в листовом узле: 1;
5) количество признаков, которые следует учитывать при поиске лучшего разделения: 4.

Результаты работы модели на тестовой выборке представлены на рисунке 20.

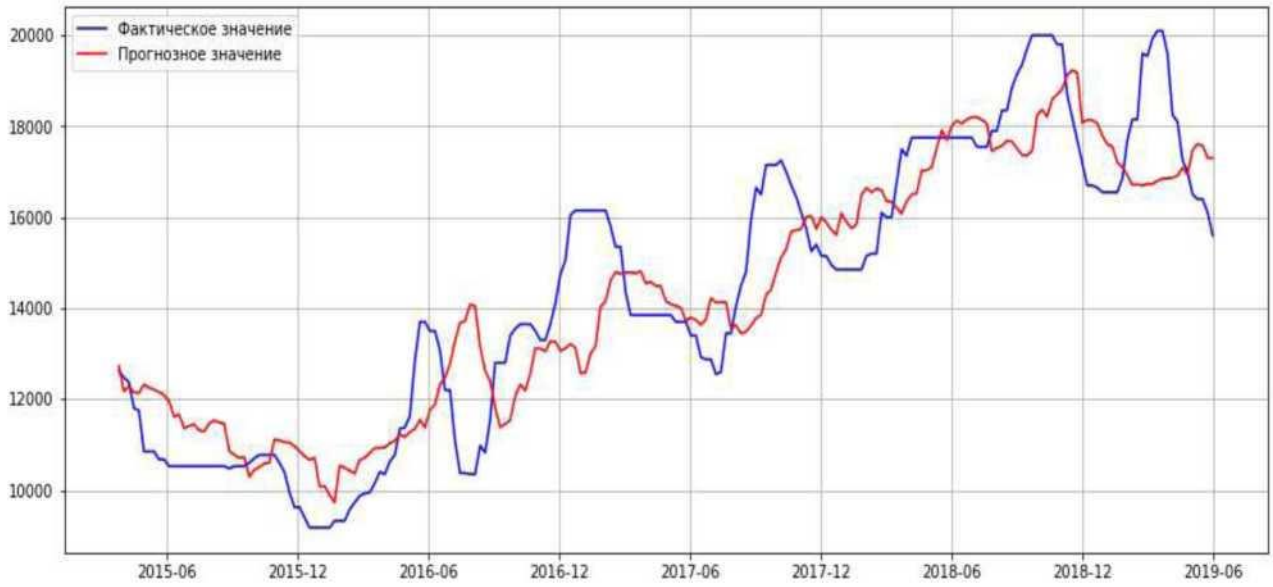


Рисунок 22 - Прогноз модели

Метод feature_importances позволяет посмотреть важность признаков при построении модели случайного леса.

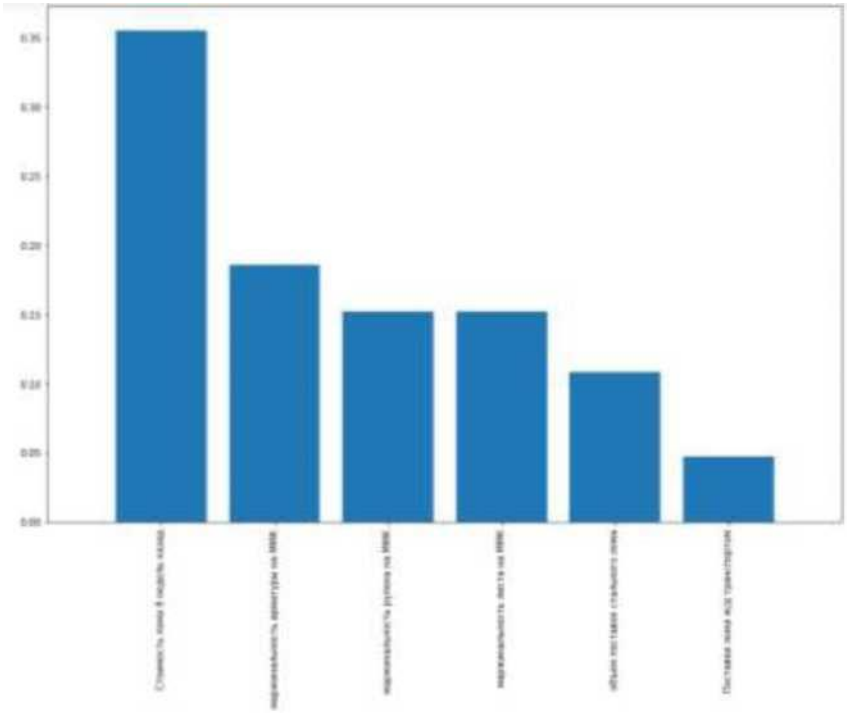


Рисунок 23 - Важность признаков в модели случайного леса

Как видно из рисунка 15 наиболее важным признаком является стоимость лома на момент прогноза, далее по важности идет маржинальность различной стальной продукции и объем поставок стального лома. Признак поставок лома

железнодорожным транспортом оказался менее значимым. В целом, данные результаты соответствуют тому, что получалось при построении линейной регрессии.

Для сравнения работы случайного леса с другими алгоритмами, прогноз изменения стоимости был преобразован в стоимость лома. Средняя абсолютная ошибка на тестовой выборке составляет 1700 рублей. Ошибка в среднем составляет 9% от истинного значения. Максимальное отклонение составляет 19%. Стандартное отклонение 1180 рублей. Коэффициент детерминации равен 0,11. Отсутствие необходимости предварительно подготавливать данные обычно является преимуществом для моделей, основанных на деревьях решений. Но в данной задаче предварительная работа с данными занимает много времени, так как необходимо преобразовывать данные в изменение стоимости. Алгоритм прост в реализации.

3.6 Градиентный бустинг.

Градиентный бустинг - еще один популярный алгоритм, в основе которого лежат деревья решений. Каждое новое дерево в алгоритме строится так, чтобы минимизировать ошибку на предыдущих деревьях. Результат прогноза будет выглядеть следующим образом:

$$a_N(x_i) = {}'Z^I f=_0 Tj(x_i), \quad (3.8)$$

где Xj - элемент выборки;

N - количество деревьев;

Tj - дерево решений.

Как видно из формулы, суммирование результатов прогноза происходит по всем алгоритмам, а не находится среднее значение как в модели случайного леса.

Сначала строится самый первый алгоритм $T_0(Xi)$. Теперь нужно построить новый алгоритм, чтобы функция потерь L(y,y) была минимальной на всей выборке.

$$Й=o^{1}(У>9) = {}_{1}>\backslash={}_{o}{}^{1}(У_{1}>{}^{a}ы-{}_{1}{}^{(}хд + Ты(хд) \wedge \min \quad (3.9)$$

Таким образом, каждый новый алгоритм будет строиться так, чтобы суммарная ошибка на всей выборке была минимальной.

$$S=o^{L}(yi>aw\text{-}i(*i) + G) \wedge \min \quad (3.10)$$

Алгоритм можно применять только к задачам интерполяции, коей не является задача прогнозирования стоимости лома. Поэтому выборка также была изменена. Вместо стоимости будет прогнозироваться изменение стоимости. А стоимостные признаки были заменены на разницу между прошлым значением и текущим. Признаки, используемые в модели: маржинальность листа, рулона и арматуры на ММК; информация о поставках лома железнодорожным транспортом за последние две недели; объемы поставок стального лома; предыдущие значения цены лома.

Для построения модели воспользуемся библиотекой skleam. Модуль ensemble предоставляет готовую реализацию метода градиентного бустинга в виде функции GradientBoostingRegressor. Для подбора параметров модели обучающая выборка была разбита на три части. Оптимальные параметры для алгоритма были подобраны с помощью кросс-валидации, которая реализована в модуле model_selection в виде функции GridsearchCV.

Оптимальные параметры получились следующие:

1) функция потерь для оптимизации: «huber»;
2) количество алгоритмов в ансамбле: 50;
3) скорость обучения: 0,1;
4) минимальное количество выборок, необходимое для разделения внутреннего узла: 2;
5) минимальное количество выборок должно быть в листовом узле: 1.

Результаты работы модели на тестовой выборке представлены на рисунке 22.

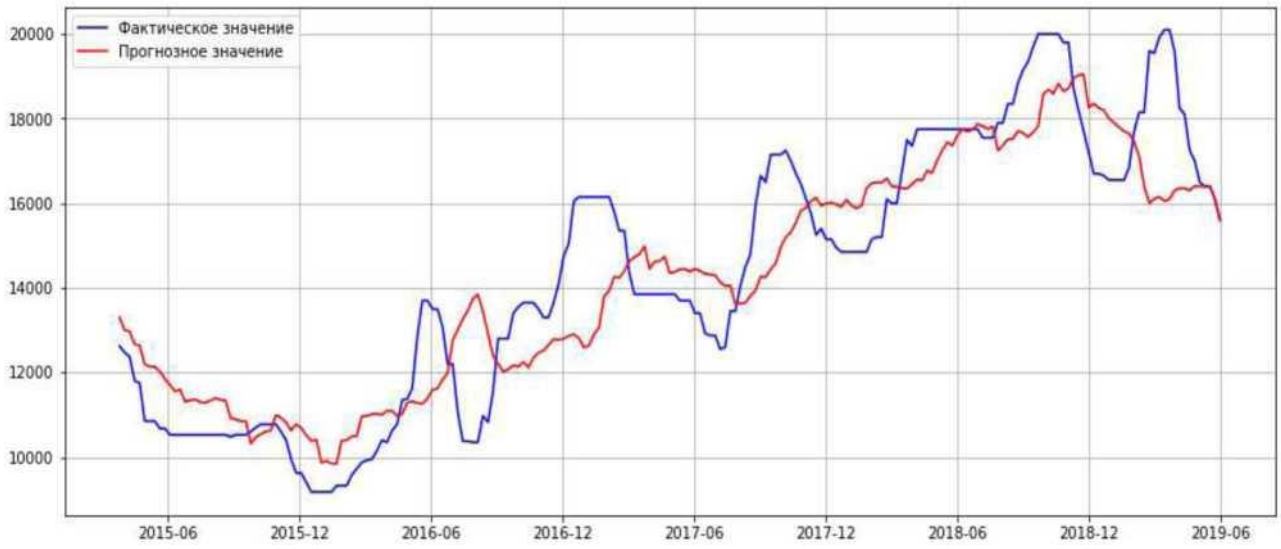
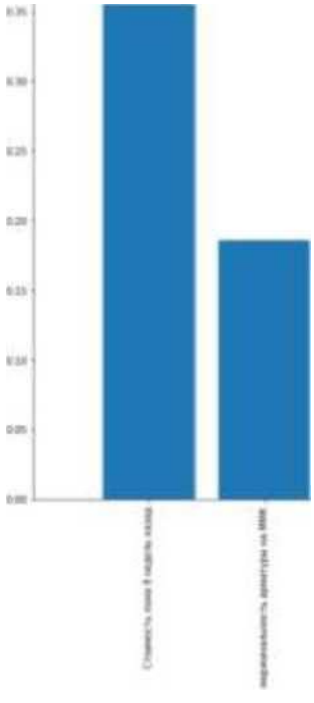


Рисунок 24 - Прогноз модели

Посмотрим важность признаков с помощью метода feature_importances.



f i l l

j

|

Рисунок 25 - Важность признаков в модели градиентного бустинга

Важность признаков соответствует тому, что было получено в модели случайного леса. Результат работы алгоритма следующий. Средняя абсолютная ошибка на тестовой выборке составляет 1793 рублей. Ошибка в среднем составляет 9,5% от истинного значения. Максимальное отклонение составляет

21%. Стандартное отклонение 1253 рубля. Коэффициент детерминации равен 0,09. Алгоритм прост в реализации, но, как и для случайного леса, проблемой является предварительная подготовка данных.

3.7 Нейронные сети

К более сложным моделям можно отнести нейронные сети. Данные модели могут отслеживать нелинейные зависимости в признаках в отличие от модели линейной регрессии. Недостатком использования нейронных сетей в данной задаче может стать то, что в нашем наличии есть наблюдения лишь за пять лет (230 объектов выборки) и этого может быть мало для корректного обучения сети.

Для построения нейронной сети необходимо нормализовать данные, например с помощью встроенного метода MinMaxScaler библиотеки skleam. Данные принимают значения от 0 до 1. Построим полносвязную нейронную сеть, состоящую из двух слоев с функцией активации «relu». На вход каждого слоя подается 7 сигналов (по числу входных признаков). Для уменьшения воздействия переобучения в нейронную сеть добавим дополнительный слой для прореживания с коэффициентом 0.3. А также в каждый слой добавим L1 регуляризацию по аналогии с линейной регрессией. Для обучения нейронной сети будем использовать «rmsprop» оптимизатор, встроенный в библиотеку keras.

Листинг программы для построения полносвязной нейронной сети представлен в приложении 3. Вначале загружаются необходимые библиотеки. Это библиотека pandas для того, чтобы считать исходные данные и библиотека keras для построения нейронной сети. Затем программный код считывает данные и делит выборку на обучение и тест.

Вызов функции get_simple_nn позволяет создать нейронную сеть. Функция get_simple_nn создает нейронную сеть с помощью метода Sequential. Далее метод add добавляет слои нейронной сети (метод Dense), прореживание (метод Dropout), выходной слой (метод Dense). Через compile задаётся алгоритм оптимизации и функция потерь.

Метод fit обучает нейронную сеть, для этого необходимо передать данные для обучения, ответы, количество эпох, размер пакета данных для обучения и тестовые данные.

В качестве входных признаков использовались признаки аналогичные модели линейно регрессии, а именно маржинальность листа, рулона и арматуры на ММК; информация о поставках лома железнодорожным транспортом за последние две недели; объемы поставок стального лома; предыдущие значения цены лома.

Результаты работы модели на тестовой выборке представлены на рисунке 24.

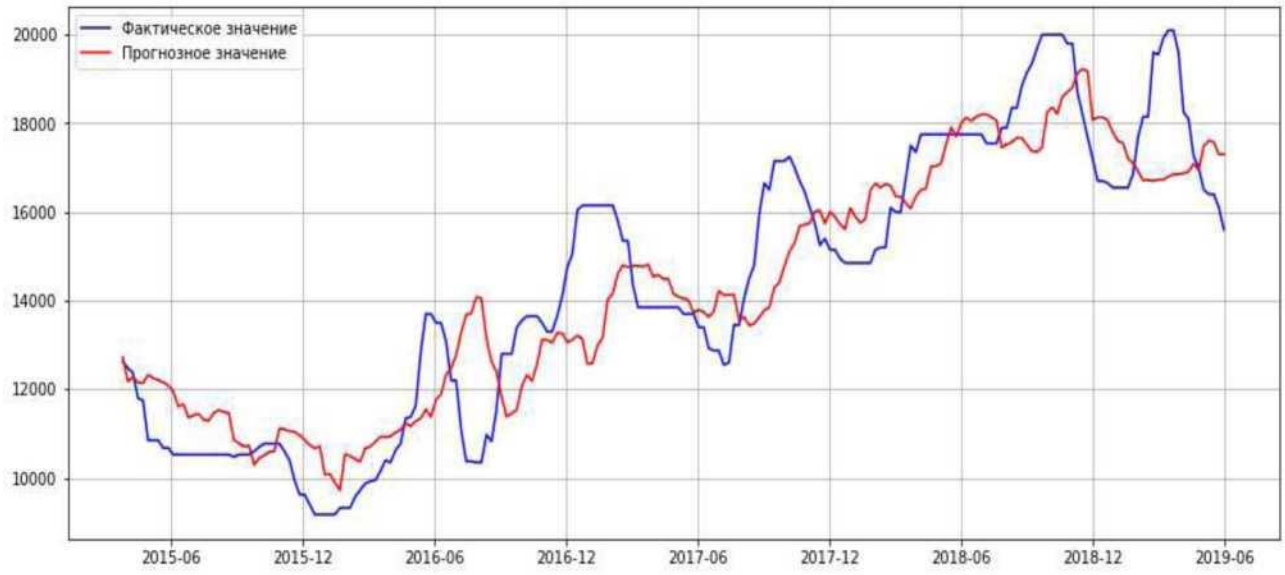


Рисунок 26 - Прогноз модели

Средняя абсолютная ошибка на тестовой выборке составляет 1916 рублей. Ошибка в среднем составляет 11% от истинного значения. Максимальное отклонение составляет 17%. Стандартное отклонение 1145 рублей. Коэффициент детерминации равен 0,21. Данный результат близок к тому, что мы получили ранее с использованием простого базового алгоритма. Можно сделать вывод, что сложность данного метода не оправдана в этой задаче.

Рассмотрим еще один мощный тип нейронной сети, а именно рекуррентные нейронные сети. Сеть с длинной кратковременной памятью или сеть LSTM - это тип периодической нейронной сети, используемой в глубоком обучении. Для

построения такой сети данные так же необходимо нормализовать. Нейронная сеть будет состоять из одного входа, скрытого слоя с четырьмя блоками или нейронами LSTM и выходного слоя. Для блока LSTM будет использована функция активации сигмоида. Добавим L1 регуляризацию с коэффициентом 0.01 в выходной слой. Обучать будем 300 эпох.

Листинг программы для построения рекуррентной нейронной сети представлен в приложении 4. Вначале загружаются необходимые библиотеки. Это библиотека pandas для того, чтобы считать исходные данные, библиотека keras для построения нейронной сети и библиотека sklearn для регуляризации данных. Затем программный код считывает данные, делит выборку на обучение и тест и преобразовывает данные.

Функция create_dataset позволяет преобразовать исходные данные к виду пригодному для загрузки в нейронную сеть. Создание нейронной сети начинается с метода Sequential, далее метод add добавляет слой рекуррентный слой LSTM и выходной слой. Через compile задаётся алгоритм оптимизации и функция потерь. Метод fit обучает нейронную сеть, для этого необходимо передать данные для обучения, ответы, количество эпох, размер пакета данных для обучения и тестовые данные.

Результаты работы модели на тестовой выборке представлены на рисунке 25.

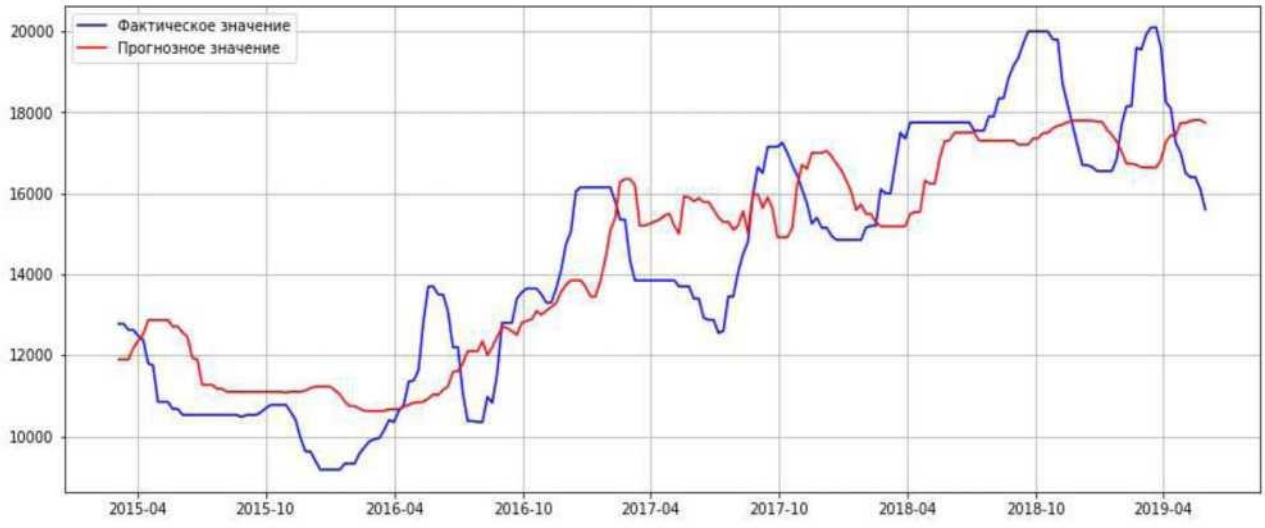


Рисунок 27 – Прогноз модели

Средняя абсолютная ошибка на тестовой выборке составляет 1659 рублей. Ошибка в среднем составляет 9% от истинного значения. Максимальное отклонение составляет 17%. Стандартное отклонение 1089 рублей. Коэффициент детерминации равен 0,22. Результат работы рекуррентной сети превзошел работу полносвязной сети.

Попробуем улучшить модель, изменив размер так называемого окна. В качестве входного признака мы использовали значение в момент времени t (значение стоимости лома в момент прогноза). Мы можем попробовать использовать несколько предыдущих значений в качестве входных признаков (t-1, t-2, t-3). В остальном нейронная сеть останется прежней, за исключением увеличенного количества входных параметров. Листинг программы для построения рекуррентной нейронной сети с увеличенным размером окна представлен в приложении 5.

Результаты работы модели на тестовой выборке представлены на рисунке 26.

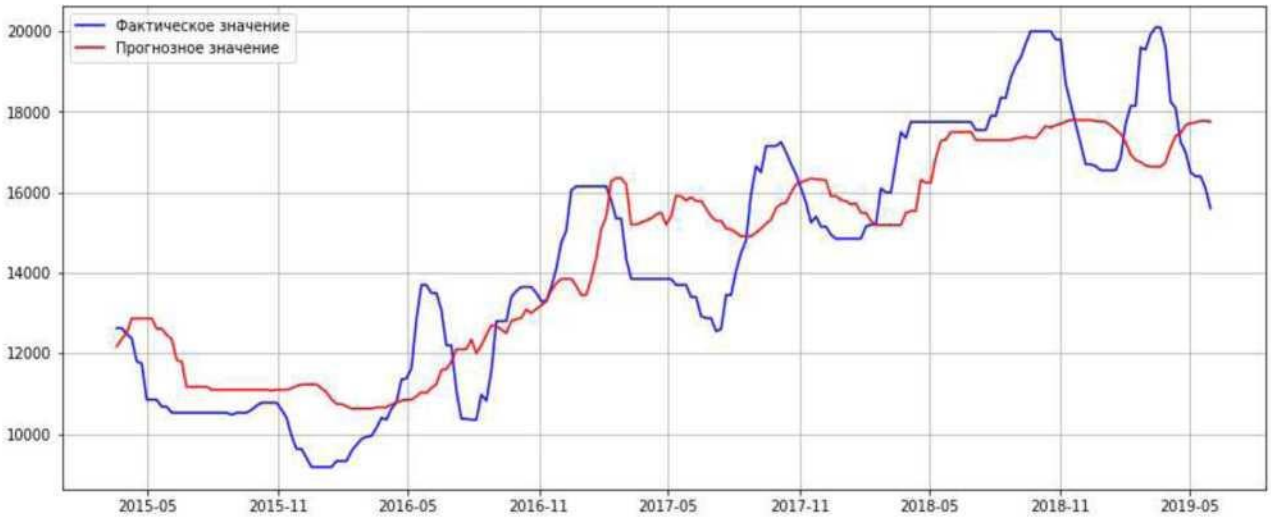


Рисунок 28 – Прогноз модели

Средняя абсолютная ошибка на тестовой выборке составляет 1545 рублей. Ошибка в среднем составляет 8% от истинного значения. Максимальное отклонение составляет 17%. Стандартное отклонение 980 рублей. Коэффициент

детерминации равен 0,25. Результат удалось немного улучшить путем настройки размера окна. Нейронные сети требуют предварительной работы с данными. Также много времени уходит на построение и настройку нейронной сети.

3.8 Сравнение работы алгоритмов.

В таблице ниже представлены результаты прогноза различных алгоритмов на тестовой выборке.

Таблица 1

Модель	**Базовый алгоритм**	**Тройное экспоненциальное сглаживание**	**Линейная регрессия**	**Линейная регрессия + тройное экспоненциальное сглаживание**	**Случайный лес**	**Градиентный бустинг**	**Полносвязная нейронная сеть**	**Рекуррентная нейронная сеть**
MAE, руб.	**2100**	**1831**	**1739**	**627**	**1700**	**1793**	**1916**	**1545**
MAPE, %	**12**	**10.9**	**9**	**3.5**	**9**	**9,5**	**11**	**8**
Максимальная ошибка в %	**23**	**25.7**	**21**	**13**	**19**	**21**	**17**	**17**

Все представленные модели показали себя лучше, чем базовый алгоритм, что говорит о возможности применения данных алгоритмов. Однако некоторые достаточно сложные в построении алгоритмы, такие как нейронные сети, работают незначительно лучше базового алгоритма. А если учитывать то, что нейронные сети требуют гораздо больше времени на реализацию, то, вероятно, их использование нецелесообразно в задаче прогнозирования стоимости лома. Возможно, более простые методы работают лучше, потому что для сложных алгоритмов необходимо больше данных для обучения.

Можно наблюдать, что лучший результат получился на модели, объединяющей алгоритм тройного экспоненциального сглаживания и линейную регрессию.

ЗАКЛЮЧЕНИЕ

В рамках работы был осуществлён анализ предметной области. Он включает в себя, прежде всего, обзор научной литературы по данной проблематике. Была оценена степень изученности и проработанности вопроса прогнозирования цен в металлургии, опираясь на научную литературу последних лет. Также были рассмотрены основные методы прогнозирования, такие как: методы экспертных оценок, методы анализа временных рядов, методы регрессионного анализа, модели на основе деревьев решений, нейронные сети.

Были построены и протестированы следующие модели: модель тройного экспоненциального сглаживания, линейная регрессия, случайны лес, градиентный бустинг, нейронные сети. Результат работы данных моделей представлен в таблице 1.

Лучший результат получился на модели, объединяющей алгоритм тройного экспоненциального сглаживания и линейную регрессию. Средняя абсолютная ошибка на данной модели составила 627 рублей, средняя абсолютная ошибка в процентах составила 3.5 %, максимальная ошибка 13%.

Можно отметить, что все исследуемые модели показали качество лучше, чем базовый алгоритм, что говорит о возможности применения данных алгоритмов. Однако некоторые достаточно сложные в построении алгоритмы, такие как нейронные сети прямого распространения, работают незначительно лучше базового алгоритма (MAE у нейронных сетей прямого распространения 1916 рублей, у базового алгоритма 2100 рублей). А если учитывать, что нейронные сети требуют гораздо больше времени на реализацию, то, вероятно, их использование нецелесообразно в задаче прогнозирования стоимости лома. Возможно, более простые методы работают лучше, потому что для сложных алгоритмов необходимо больше данных для обучения. В нашем распоряжении имелось лишь 230 объектов выборки.

Средняя абсолютная ошибка у модели случайного леса составила 1700 рублей, а у модели градиентного бустинга 1793 рублей. Данные модели показали качество сравнимое с простой линейной регрессией. При реализации данных

моделей необходимо потратить достаточно большое количество времени на подготовку данных. Поэтому при необходимости получить быстрый результат лучше воспользоваться линейной регрессией и экспоненциальным сглаживанием.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика. Исследование зависимостей. М.: Финансы и статистика, 1985.
2. Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика. Классификация и снижение размерности. М.: Финансы и статистика, 1989.
3. Арстанов М.Ж., Пидкасистый П.И., Хайдоров Ж.С. Проблемно-модульное обучение. Вопросы теории и технологии. - Алма-Ата: Мектеп, 1980.
4. Бабич Т.Н., Козьева И.А., Вертакова Ю.В., Кузьбожев Э.Н. Прогнозирование и планирование в условиях рынка : учеб. пособие. М.: ИНФРА-М, 2013. 336 с.
5. Бринк Хенрик, Ричардс Джозеф, Феверолф Марк. Машинное обучение. - СПб.: Питер, 2017. - 336 с.
6. Введение в математическое моделирование. Учебное пособие. - М.: Логос, 2015. - 440с.
7. Видмант О.С. Прогнозирование финансовых временных рядов с использованием рекуррентных нейронных сетей LSTM // Общество: политика, экономика, право. 2018. № 5 (58). С. 63-66.
8. Графов А. В. К вопросу о формировании потребительной стоимости и эффективности вторичных черных металлов/ А. В. Графов// Аудитор. - М.: Русский журнал. - 2010. - №5.
9. Демиденко Е.З. Линейная и нелинейная регрессия. - М.: Финансы и статистика, 1981. - 302с.
10. Дитман Т.А., Нордин В.В. Прогнозирование спроса экспоненциальным сглаживанием временных рядов // Пространственное развитие региона: перспективы, приоритеты, ресурсы. 2019. С. 59-61.
11. Дубровин М.Г., Глухих И.Н. Применение модели Хольта - Винтерса для прогнозирования работоспособности серверных систем // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. 2019. № 4. С. 35-41.
12. Евланов Л.Г., Кутузов В.А. Экспертные оценки в управлении. - М.: Экономика, 1978. 133 с.

13. Иванова Т. А., Трофимова В. Ш., Калитаев А. Н., Степанов Д. Г. Математическое моделирование ценового диапазона закупа лома черных металлов для металлургических предприятий РФ // Экономика региона. — 2018. — Т. 14, вып. 1. — С. 137-149.
14. Иванюк В.А. Модели и методы прогнозирования финансовых временных рядов на примере курса USD/RUB // Мягкие измерения и вычисления. 2019. № 6 (19). С. 38-41.
15. Интеллектуальный анализ данных: учеб. пособие. - Томск Издательский Дом Томского государственного университета, 2016 - 120 с.
16. Канторович Г.Г. Анализ временных рядов. // Экономический журнал ВШЭ, №3, 2002.
17. Козлова Е.И., Ананьев К.А. Обзор мирового и российского рынка стали// Инновационная экономика и право. - 2018. - №1 (10). - С. 44-47.
18. Крюкова Е.М. Тенденции и перспективы рынка лома черных металлов // ЭКО. - 2009. - №3. - С. 129-141.
19. Крюкова Е.М. Применение методов организационно-экономического прогнозирования в отрасли лома черных металлов// Заводская лаборатория. Диагностика материалов. - 2008. - Т.74. - №7. - С. 67-72. - 0,5.
20. Крюкова Е.М. Особенности ценообразования на рынке лома черных металлов // Электрометаллургия. - 2008. - №5. - С. 40-46.
21. Кундышева Е.С. Экономико-математическое моделирование: Учебник / Под науч. ред. проф. Б.А. Суслакова . - М.: Издательско-торговая корпорация «Дашков и К», 2008. - 424 с.
22. Куценко В.И. Социальная задача как категория исторического материализма. - Киев: Наукова думка, 1972.
23. Лемешонок К.А., Ботыгин И.А. Исследование возможности применения рекуррентный нейронных сетей в задачах прогнозирования временных рядов // Научный альманах. 2019. № 4-2 (54). С. 54-57.
24. Лукашин Ю.П. Адаптивные методы краткосрочного прогнозирования. М.: Статистика, 2003.

25. Лыкошев Д. Н. Некоторые аспекты ценообразования в металлургии и металлообработке: проблемы повышения конкурентоспособности продукции / Д. Н. Лыкошев // Вестник УГТУ-УПИ. Серия экономика и управление. — 2006. — № 9. — С. 21-26.
26. Магомедрагимова Э.Р. Прогнозирование рыночной стоимости недвижимости путем применения искусственных нейронных сетей // Вестник современных исследований. 2017. № 4-1 (7). С. 68-73.
27. Маланичев А.Г. Система сценарного планирования и прогнозирования мировых цен стали и металлургического сырья // Проблемы прогнозирования. 2014. № 3. С. 53-62.
28. Математические основы управления проектами / Под ред. В.Н. Буркова. - М.: Высшая школа, 2005.
29. Методологические основы научного познания / Под ред. П.В. Попова. Учеб. пособие для студентов вузов.- М.: Высшая школа, 1972.
30. Мишулина О.А. Статистический анализ и обработка временных рядов. М.: МИФИ, 2004.
31. Мостеллер Ф., Тьюки Дж. Анализ данных и регрессия. М.: Финансы и статистика, 1982.
32. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. — СПб.: Питер, 2019. — 480 с.
33. Оглуздина О.Б. Эффекты взаимодействия ресурсных подсистем как фактор повышения конкурентоспособности предприятия // Вестник УрФУ. Серия экономика и управление. 2015. Т. 14, № 3. С. 377-394.
34. Орлов А.И. Экспертные оценки. Учеб. пособие. - М.: 2002.
35. Ращупкина О.С. Анализ временных рядов с помощью системы Statistica // В сборнике: фундаментальные и прикладные исследования в области управления, экономики и торговли. Сборник трудов научно-практической и учебной конференции. 2019. С. 58-61.

36. Система сценарного планирования и прогнозирования мировых цен стали и металлургического сырья [Текст] / А. Г. Маланичев // Проблемы прогнозирования. - 2014. - № 3. - С. 53-62: табл., рис. - Библиогр.: 31 назв.
37. Степаненко Д.Б. Разработка гибридной модели прогнозирования временных рядов на основе алгоритма случайного леса и модели ARIMA // Аллея науки. 2018. Т. 4. № 4 (20). С. 969-973.
38. Сурков Ф.А., Петров Н.В., Суховский С.Ф. Сравнение временных рядов и нейросетевых методов в задаче прогнозирования стоимости и оценки недвижимости // Моделирование, оптимизация и информационные технологии. 2018. Т 6. № 3 (22). С. 88-103.
39. Тихоновская, И. Д. Прогнозирование цен на лом черных металлов как ключевой фактор системы ресурсообеспечение металлургического предприятия. Вестник УрФУ. Серия: Экономика и управление, 15(1), 97-116.
40. Тихоновская И. Д. Методический подход к управлению системой обеспечения металлургических предприятий ломом черных металлов / И. Д. Тихоновская // Вестник УрФУ. Серия: Экономика и управление. — 2016. — № 5. — С. 673695.
41. Турлакова С.У., Ильиных М.В. Прогнозирование цен на лом черных металлов. / Турлакова С.У., Ильиных М.В. // Наука ЮУрГУ материалы 71-й научной конференции. — 2019. — С. 374-381.
42. Шолле Франсуа. Глубокое обучение на Python. — СПб.: Питер, 2018. — 400 с.
43. Jasek R., Szmit A., Szmit M. Usage of Modem Exponential-Smoothing Models in Network Traffic Modelling // Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems. Heidelberg: Springer, 2013. P. 435-444.
44. Kalekar P.S. Time Series Forecasting Using Holt - Winters Exponential Smoothing // Kanwal Rekhi School of Information Technology. 2014. Т. 4329008, № 13.
45. Michael J Kane1, Natalie Price, Matthew Scotch, Peter Rabinowitz. Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks // BioMed Central, 2014.

46. Roondiwala M., Patel H., Varma S. Predicting stock prices using LSTM // International Journal of Science and Research. 2017. Vol. 6, no. 4. P. 1754-1756.
47. Hansson M. On stock return prediction with LSTM networks // Lund University, 2017.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1

Листинг программы для построения модели тройного экспоненциального сглаживания

```
class HoltWinters:

    #  series - исходный временной ряд
    #  slen - длина сезона
    #  alpha, beta, gamma - коэффициенты модели Хольта-Винтерса
    #  n_preds - горизонт предсказаний

    def _ init_ (self, series, slen, alpha, beta, gamma, n_preds):
        self.series = series self.slen = slen
        self.alpha = alpha self.beta = beta
        self.gamma = gamma self.n_preds = n_preds

    def initial_trend(self):
        sum = 0.0
        for i in range(self.slen):
                  sum += float(self.series[i+self.slen] - self.series[i]) /
self.slen
        return sum / self.slen

    def initial_seasonal_components(self):
        seasonals = {} season_averages = []
        n_seasons = int(len(self.series)/self.slen)
        #  вычисляем сезонные средние for j in
        range(n_seasons):

 season_averages.append(sum(self.series[self.slen*j:self.slen*j+self.slen])/
float(self.slen))
        #  вычисляем начальные значения for i in
        range(self.slen):
            sum_of_vals_over_avg = 0.0
            for j in range(n_seasons):
                sum_of_vals_over_avg += self.series[self.slen*j+i]-
season_averages[j]
            seasonals[i] = sum_of_vals_over_avg/n_seasons
        return seasonals

    def triple_exponential_smoothing(self):
        self.result = []
```

```
        self.Smooth = [] self.Season =
           [] self.Trend = []
           self.PredictedDeviation = []

           seasonals = self.initial_seasonal_components()

           for i in range(len(self.series)+self.n_preds):
               if i == 0: # инициализируем значения компонент
                   smooth = self.series[0] trend =
                   self.initial_trend()
                   self.result.append(self.series[0])
                   self.Smooth.append(smooth)
                   self.Trend.append(trend)
                   self.Season.append(seasonals[i%self.slen]
                   )

                   self.PredictedDeviation.append(0)

                   continue
               if i >= len(self.series): # прогнозируем m = i
                   - len(self.series) + 1
                   self.result.append((smooth       +       m*trend)       +
seasonals[i%self.slen])
```

ПРИЛОЖЕНИЕ 2

Листинг программы для подбора оптимальных параметров тройного экспоненциального сглаживания на языке программирования Python

```
from sklearn.model_selection import TimeSeriesSplit

def timeseriesCVscore(x): error = 0
    alpha, beta, gamma = x
    temp = df[df.index <= '2016-01-01']
    predict = pd.DataFrame()
    for i in df[df.index > '2016-01-01'].index:
        model = HoltWinters(temp['y'], slen = 26, alpha = alpha, beta = beta,
gamma = gamma, n_preds = 8)
        model.triple_exponential_smoothing()
        predict.loc[i+datetime.timedelta(days=7*7), 'y']  = model.result[-
1:][0]
        temp = temp.append(df[df.index==i])

    error       =       mean_absolute_error(predict[predict.index       <=
                                                        '2019-05-10']
    ,
temp[temp.index >= '2016-02-26']) print(error) return
    error
#  инициализируем значения параметров x = [0, 0, 0]

# Минимизируем функцию потерь с ограничениями на параметры
opt = minimize(timeseriesCVscore, x0=x, method="TNC", bounds = ((0, 1), (0, 1),
(0, 1)))

# Из оптимизатора берем оптимальное значение параметров
alpha_final, beta_final, gamma_final = opt.x
```

ПРИЛОЖЕНИЕ 3

Листинг программы для построения полносвязной нейронной сети на языке программирования Python

```
import pandas as pd
from keras import models
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
#  считываем готовые данные
df = pd.read_excel('df.xlsx')
#  целевой признак
y = ['Лом 3А, РФ, внутр. рынок, СРТ, Уральский регион, руб/т без НДС']
#  признаки для прогноза
x = [
     'Объемы поставки (факт) (Стальной лом)',
     'Маржинальность Толстый лист ММК',
     'Маржинальность Рулон ММК',
     'Маржинальность Арматура ММК',
     'Поставки ж.д. трансопртом Итог 14 дней',
     'Лом 3А, t-n',
     ]
#  делим выборку на обучение и тест
test = df[df.index > '2019-01-01']
train     =     df[(df.index     <
'2019-01-01')]
X_train = train[x]
y_train = train[y]
X_test = test[x]
y_test = test[y]


def get_simple_nn(n_input, n_output=1):
     model = Sequential()

     model.add(Dense(7,                                        activation='relu',
kernel_regularizer=regularizers.l1(0.01), input_dim=n_input))
     model.add(Dropout(0.3))
     model.add(Dense(7,                                        activation='relu',
kernel_regularizer=regularizers.l1(0.01))
     ) model.add(Dropout(0.3))
     model.add(Dense(1))
     model.compile(optimizer='rmsprop', loss='mse', metrics=['mae'])

     return model

# Параметры обучения
batch_size = 64 epochs =
2000
```

```
# создаем модель нейронной сети
model = get_simple_nn(len(x))
# обучение нейронной сети
hist = model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size,
validation_data=(X_test, y_test))
```

ПРИЛОЖЕНИЕ 4

Листинг программы для построения рекуррентной нейронной сети на языке программирования Python

```
import pandas as pd
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from sklearn.preprocessing import MinMaxScaler

# подготовка данных для нейронной сети def
create_dataset(dataset, look_back=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-look_back-1):
        a = dataset[i:(i+1), 0]
        dataX.append(a)
        dataY.append(dataset[i + look_back, 0])
    return numpy.array(dataX), numpy.array(dataY)

# загрузка исходных данных
df = pd.read_excel('Исходные данные (преобразованные).xlsx')
# нормализация данных
scaler = MinMaxScaler(feature_range=(0, 1))
df = scaler.fit_transform(df)
# делим данные на обучение и тест
train = df[:-22]
test = df[-31:]
look_back = 8
trainX, trainY = create_dataset(train, look_back)
testX, testY = create_dataset(test, look_back)
trainX = numpy.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
testX = numpy.reshape(testX, (testX.shape[0], 1, testX.shape[1]))
# создание нейронной сети model = Sequential()
model.add(LSTM(4, input_shape=(1, 1)))
model.add(Dense(1, kernel_regularizer=regularizers.l1(0.01)))
model.compile(loss='mean_squared_error', optimizer='adam')
# обучение нейронной сети
model.fit(trainX, trainY, epochs=300, batch_size=1, verbose=2)
```

ПРИЛОЖЕНИЕ 5

Листинг программы для построения рекуррентной нейронной сети с увеличенным размером окна на языке программирования Python

```
import pandas as pd
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from sklearn.preprocessing import MinMaxScaler

#  подготовка данных для нейронной сети def
create_dataset(dataset, look_back=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-look_back-1):
    if i in [0, 1, 2]: continue
        a = dataset[i-3:(i+1), 0]
        dataX.append(a)
        dataY.append(dataset[i + look_back, 0])
    return numpy.array(dataX), numpy.array(dataY)

#  загрузка исходных данных
df = pd.read_excel('Исходные данные (преобразованные).xlsx')
#  нормализация данных
scaler = MinMaxScaler(feature_range=(0, 1))
df = scaler.fit_transform(df)
#  делим данные на обучение и тест
train = df[:-22]
test = df[-31:]
look_back = 8
trainX, trainY = create_dataset(train, look_back)
testX, testY = create_dataset(test, look_back)
trainX = numpy.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
testX = numpy.reshape(testX, (testX.shape[0], 1, testX.shape[1]))
#  создание нейронной сети model = Sequential()
model.add(LSTM(4, input_shape=(1, 4)))
model.add(Dense(1, kernel_regularizer=regularizers.l1(0.01)))
model.compile(loss='mean_squared_error', optimizer='adam')
#  обучение нейронной сети
model.fit(trainX, trainY, epochs=300, batch_size=1, verbose=2)
```