Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Факультет инновационных технологий (ФИТ)
Кафедра информационного обеспечения инновационной деятельности (ИОИД)

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП,
д.т.н., профессор
_____ В.И. Сырямкин
«20» июня 2019 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

АЛГОРИТМ ПРЕДСКАЗАНИЯ ПРОДАЖ С ИСПОЛЬЗОВАНИЕМ БОЛЬШИХ ДАННЫХ И ТЕХНОЛОГИЙ МАШИННОГО ОБУЧЕНИЯ

по основной образовательной программе подготовки магистров
направление подготовки 09.04.02 – Информационные системы и технологии

Гайдук Кристина Алексеевна

Научный руководитель
канд. техн. наук,
доцент кафедры УК
_____Д.В. Шашев
«19» июня 2019 г.

Автор работы
студент группы №18705
_____К.А. Гайдук
«19» июня 2019 г.

Томск 2019

Министерство науки и высшего образования Российской Федерации

Национальный исследовательский Томский государственный университет (НИ ТГУ)
Факультет инновационных технологий (ФИТ)
Кафедра информационного обеспечения инновационной деятельности (ИОИД)

УТВЕРЖДАЮ
Руководитель ООП,
д.т.н., профессор
_____ В.И. Сырямкин
«20» июня 2019 г.

ЗАДАНИЕ

по подготовке выпускной квалификационной работы магистра студенту группы №18705 Гайдук Кристине Алексеевне

1. Тема ВКР: «Алгоритм предсказания продаж с использованием больших данных и технологий машинного обучения» (утверждена приказом по факультету от «09» апреля 2019 г. №30410).

2. Срок сдачи студентом законченной ВКР:

а) на кафедре 17.06.2019 г.

б) в ГЭК 21.06.2019 г.

3. Исходные данные к ВКР: техническое средства, используемые в реализации поставленных задач, результаты научно-исследовательской работы, производственных и преддипломной практик.

4. Краткое содержание работы: Изучение основ работы с большими данными. Изучение методов, способов и задач машинного обучения. Изучение способов и методов анализа временных рядов. Рассмотрение теоретических аспектов популярных алгоритмов предсказания продаж товаров определенного типа. Изучение основ работы с программами для создания алгоритма предсказания продаж. Практическая реализация алгоритма предсказания продаж. Создание предсказания для определенной группы товара. Определение тренда и сезонности продаж для определенных групп товаров. Анализ полученных данных, заключение о проделанной работе.

5. Перечень графического материала: информация о предприятии, описание проблем, описание аналогов, постановка цели работы, логическая и физическая модель базы данных, входные/выходные данные, описание интерфейса программы, выводы по работе.

6. Дата выдачи задания 9 апреля 2019 г

Руководитель ВКР,
доцент кафедры УК _____ Д.В. Шашев

Задание принял к исполнению _____ К.А. Гайдук

РЕФЕРАТ

Магистерская диссертация содержит: 68 с., 46 рис., 4 табл., 20 литературных источников.

Список ключевых слов: PYTHON, JUPITER, ПРОГНОЗ, АЛГОРИТМ ПРОГНОЗИРОВАНИЯ, ВРЕМЕННЫЕ РЯДЫ, МАШИННОЕ ОБУЧЕНИЕ, БОЛЬШИЕ ДАННЫЕ, ПРЕДСКАЗАНИЕ.

В магистерской диссертации рассмотрены основные принципы и особенности построения Алгоритмов прогнозирования продаж с использованием технологий машинного обучения и больших данных.

Исследованы различные существующие методы построения алгоритмов предсказания. Выявлены их преимущества и недостатки.

Разработан алгоритм предсказания продаж определенной группы товара на основе существующих данных и сравнение полученного прогноза с реальными продажами. Также был проведен прогноз для определения тренда продаж и сезонности продаж.

Для разработки алгоритма прогнозирования продаж использовалось программное обеспечение Jupiter. Для реализации алгоритма был использован язык программирования Python.

Проанализированы результаты работы алгоритма прогнозирования и определены точности сделанных прогнозов.

ABSTRACT

The master's thesis contains: 68 p., 40 fig., 4 tab., 16 literary sources.

List of keywords: PYTHON, JUPITER, FORECAST, FORECASTING ALGORITHM, TEMPORARY SERIES, MACHINE TRAINING, GREAT DATA, FORECAST.

The master's thesis discusses the basic principles and features of the construction of sales forecasting algorithms using machine learning technologies and big data.

Various existing methods for constructing prediction algorithms are investigated. Identified their advantages and disadvantages.

An algorithm has been developed for predicting sales of a certain product group based on existing data and comparing the forecast obtained with actual sales. Also, a forecast was conducted to determine the sales trend and sales seasonality.

To develop a sales forecasting algorithm, Jupiter software was used. To implement the algorithm, the programming language Python was used.

The results of the prediction algorithm are analyzed and the accuracy of the predictions made is determined.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 7

1 Основные понятия технологий машинного обучения 9

1.1 Обзор методов машинного обучения 9

1.2 Определение способов и задач машинного обучения 12

2 Основные понятия о Big Data 16

2.1 Обзор и выбор методик для анализа больших данных 19

3 Обзор моделей прогнозирования временных рядов 23

3.1 Сравнение существующих моделей прогнозирования 25

3.2 Выбор модели прогнозирования для решения поставленной задачи 27

4 Выбор инструментов для реализации алгоритма предсказания 30

4.1 Выбор языка программирования 30

4.2 Выбор библиотек для реализации алгоритма 33

5 Анализ факторов влияющих на покупательский спрос 37

6 Программная реализация алгоритма предсказания 39

6.1 Прогнозирование временных рядов с использованием модели ARIMA .. 46

6.2 Проверка точности прогноза 49

6.3 Создание долгосрочного прогноза 53

7 Сравнение прогнозов двух категорий товаров и выявление тренда и сезонности продаж 55

ЗАКЛЮЧЕНИЕ 64

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 66

ВВЕДЕНИЕ

Построение систем машинного обучения является на сегодняшний день одной из самых популярных, актуальных областей человеческой деятельности на стыке информационных технологий, математического анализа и статистики. Задачи прогнозирования решаются в различных сферах человеческой деятельности, например, науке, производстве, экономике и др. Главной целью прогнозирования является предсказание будущего поведения объекта прогнозирования. Сам процесс прогнозирования является одной из задач машинного обучения и одним из важнейших моментов принятия решений при оптимизации товарного запаса, логистики и необходимых ресурсов. Решение задачи прогнозирования направлено на определение тенденций развития конкретного объекта или действия на основе предшествующих данных, точнее на анализе их состояний в прошлом и настоящем. Исходя из этого, для решения задачи прогнозирования требуется наличие определенной обучающей выборки данных.

В магистерской диссертации рассматриваются математические модели и методы, предназначенные для прогнозирования временных рядов с использованием регулярных периодических компонентов. Примерами таких моделей и методов являются: метод сингулярного спектрального анализа, модель сезонного скользящего среднего, ARIMA и т.д.

В данной работе исследуются популярные алгоритмы предсказания продаж с использованием больших данных и технологий машинного обучения. Также рассмотрены вопросы, связанные с решением задачи прогнозирования временных рядов. Также рассматривается алгоритм определения тренда и сезонности продаж товаров определенного типа.

Основной целью работы является разработка алгоритма прогнозирования продаж товара определенного типа и сравнение созданного прогноза с существующими данными. Также создание долгосрочного прогноза и выявление тренда и сезонности продаж товаров определенного

типа. Таким образом, объектом исследований является технологии машинного обучения для создания прогноза. Предметом исследования – алгоритмы прогнозирования продаж товаров определенного типа.

Основные задачи:

1) Анализ литературных источников, исследование основных требований к реализации алгоритма предсказания.
2) Изучение процессов создания алгоритмов предсказания с использованием технологий машинного обучения.
3) Исследование популярных алгоритмов предсказания, подходящих для решения поставленной задачи.
4) Выбор методов, способов и задачи машинного обучения. Выбор модели и метода прогнозирования.
5) Разработка алгоритма прогнозирования продаж скоропортящейся продукции на 2018 год и сравнение полученного прогноза с реальными данными.
6) Анализ результатов работы алгоритма, выявление на основе двух прогнозов показателей тренда и сезонности продаж.

1 Основные понятия технологий машинного обучения

Машинное обучение – это подраздел искусственного интеллекта, который изучает методы построения алгоритмов, имеющие способность к обучению. На сегодняшний день машинное обучение сочетает в себе понятия математической статистики, понятия и методы оптимизации и основные понятия классических математических дисциплин.

1.1 Обзор методов машинного обучения

Все методы машинного обучения можно разделить на три основные группы: контролируемые, неконтролируемые и подкрепляемые обучения.

Контролируемое обучение используется в тех случаях, когда свойство применяется для определенного выбранного массива данных, но на настоящий момент оно отсутствует для массива данных и может быть предсказано для следующих случаев.

Неконтролируемое обучение используется для выявления неявных зависимостей в данном необозначенном наборе данных.

Подкрепляемое обучение содержит в себе свойства предыдущих двух групп обучения. При подкрепляемом обучении существует некоторая форма обратной связи. Она доступна для каждого шага или действия. Недостатком данной группы является отсутствие свойства и сообщения об ошибке.

К контролируемому обучению относятся следующие методы: дерево принятия решений, наивная байесовская классификация, метод наименьших квадратов, логистическая регрессия, метод опорных векторов, метод ансамблей.

Метод дерева принятия решений представляет собой средство поддержки принятия решений, которое основано на древовидном графе. Он содержит в себе модель принятия решений и структуру последствия их работы. Данная структура включает в себя вероятность наступления события, а также расчет затрат ресурсов и полезность решения.

Метод наивных байесовских классификаторов представляет собой набор простых вероятностных классификаторов. Данные классификаторы основаны на применении Теоремы Байеса со строгими предположениями о независимости функций.

Метод наивных байесовских классификаторов применяется для решения следующих задач:

– определение и удаление спама в электронной почте;
– классификация новостей и новостных статей по тематике;
– определение стилистики блока текста;
– программное обеспечение в системах технического зрения для распознавания лиц.

Метод наименьших квадратов используется для реализации линейной регрессии. Метод основан на построении прямой линии через множество разрозненных точек. Чаще всего данный метод используется для минимизации погрешности и относится к типам метрики ошибок.

Метод логистической регрессии является способом прогнозирования в статистике вероятности возникновения некоторого события с несколькими переменными. Данный метод определяет степень зависимости между категориальной зависимой и несколькими независимыми переменными.

Метод логистической регрессии применяется для решения следующих задач:

- кредитном скоринге;
- определении уровня успешности кампаний;
- предсказании будущих доходов кампании.

Метод опорных векторов представляет собой набор алгоритмов, которые используются для решения задач классификации и выполнения регрессионного анализа.

Метод опорных векторов применяется для решения следующих задач:

- отображение рекламы на определенных сайтах;
- определение пола человека на основании исследуемой фотографии;

- сплайсинг человеческой ДНК.

Метод ансамблей базируется на обучающихся алгоритмах, которые отталкиваясь от голосования или усреднения, создают множество классификаторов на основе данных, а затем определяют новые точки данных.

К неконтролируемому обучению относятся следующие методы: алгоритмы кластеризации, метод главных компонент, сингулярное разложение, анализ независимых компонент.

Метод кластеризации группирует множества объектов на основе схожести объектов по заданным начальным признакам. Данный метод широко применяется таких дисциплинах как IT-технологии, социология и биология.

Метод главных компонент широко используется в статистике. Этот метод представляет собой статистическую процедуру, которая использует ортогональное преобразование. Цель использования ортогонального преобразования - конвертация набора наблюдений за возможно коррелированными переменными в другой набор значений линейно некоррелированных переменных. В этом случае некоррелированные переменные называются главными компонентами. Данный метод применяется для упрощения и сжатия данных, которые используются для обучения. Также используется для визуализации.

Метод сингулярного разложения тесно связан с методом главных компонент. Метод представляет собой разложение прямоугольной вещественной или комплексной матрицы. Данный метод применяется в системах технического зрения. Метод позволяет представлять захваченные лица в виде суммы базисных компонент, также позволяет выполнить уменьшение размерности и найти их изображения из обучающей выборки.

Метод анализ независимых компонент является статистическим методом. Он служит для выявления скрытых факторов, лежащих в основе множества случайных величин. Данный метод необходим для определения порождающей модели при исследовании многофакторных данных, которые представлены в виде базы данных образцов. В методе переменные

представлены как линейная смесь некоторых скрытых переменных при этом любая информация о законах смешивания закрыта. Данный метод связан с методом главных компонент. Сложен, но используется, когда метод главных компонент не способен выполнить решаемую задачу. Метод применяется при распознавании речи и в системах технического зрения, диагностировании сложных систем, телекоммуникациях и астрономии.

1.2 Определение способов и задач машинного обучения

Наиболее распространённым способом машинного обучения является способ обучение с учителем. Все прецеденты представляют собой пары, представленные как объект/ответ. При данном способе обучения от программы требуется определить функциональную зависимость ответов основанных на описаниях представленных объектов и создать алгоритм, у которого на вход подается описание объекта, а на выходе показывается ответ. Качество алгоритма обычно определяется как средняя ошибка ответов, которые выдал объект от всех объектов выборки.

Данный способ обучения позволяет подходит для решения следующих задачи:

1) Задача классификации. Главная особенность данной задачи заключается в том, что множество допустимых ответов конечно;

2) Задача регрессии Главная особенность данной задачи заключается в том, что допустимым ответом является действительное число;

3) Задача ранжирования. Главная особенность данной задачи заключается в том, что ответы определяются мгновенно на множестве объектов, затем проходит сортировка их по значениям ответов;

4) Задача прогнозирования. Главная особенность данной задачи заключается в том, что объектами являются временные ряды, на основе которых необходимо сделать прогноз на будущее.

При способе обучения без учителя случае ответы не задаются, и программе требуется искать функциональные зависимости между объектами самой.

Данный способ обучения позволяет подходит для решения следующих задачи:

1) Задача кластеризации. Основана на создании кластеров и группировании объектов, используя данные о сходстве объектов;

2) Задача поиска ассоциативных правил. Основана на поиске похожих признаков и объединении в группы объектов, которые имеют особенно частые признаковые описания;

3) Задача фильтрации выбросов. Основана на обнаружение в обучающей выборке малого числа не часто встречающихся объектов;

4) Задача построения доверительной области. Основана на создании области минимального объёма с достаточно гладкой границей, которая содержит заданную долю выборки;

5) Задача сокращения размерности. Основана на переходе к наименьшему числу новых признаков, с помощью некоторых функций преобразования, используя исходные признаки;

6) Задача заполнения пропущенных значений. Основана на замене недостающих значений в матрице признаков объектов их спрогнозированными значениями.

Способ частичного обучения представляет собой среднее между обучением с учителем и без учителя. Все прецеденты представляют собой пары объект/ответ, но ответы известны только на какую-то часть прецедентов.

Способ трансдуктивного обучение — это способ, при котором уже задана конечная обучающая выборка прецедентов. Программе требуется по известной обучающей выборке сделать предсказания относительно другой тестовой выборки. При этом способе обучения не требуется находить общую закономерность, потому что известно, что новых тестовых прецедентов нет.

Способ обучения с подкреплением. Роль объектов играют пары «ситуация, принятое решение», ответами являются значения функционала качества, характеризующего правильность принятых решений (реакцию среды). Как и в задачах прогнозирования, здесь существенную роль играет фактор времени. Примеры прикладных задач: формирование инвестиционных стратегий, автоматическое управление технологическими процессами, самообучение роботов, и т.д.

Способ динамического обучения позволяет выбрать между обучением с учителем и обучением без учителя. При данном способе прецеденты поступают потоком. Решение необходимо принимать немедленно и доучивать модель на основе новых прецедентов.

Способ активного обучения позволяет учителю назначить следующий прецедент, который станет известен.

Способ мета-обучения заключается в том, что прецедентами являются ранее решённые задачи обучения. И для корректной работы требуется определить какая модель из решенных задач работает лучше. Главной задачей является обеспечение совершенствования алгоритма с течением времени в автоматическом режиме.

Способ многозадачного обучения представляет собой набор взаимосвязанных или похожих задач обучения. Данный способ решается одновременно, с помощью различных алгоритмов обучения, основанных на схожем внутреннем представлении. При наличии информации о сходстве задач позволяет более совершенствовать алгоритм обучения и повышать качество решения поставленной задачи.

Способ индуктивного переноса. Основан на применении опыта решения отдельных частных задач обучения по прецедентам и использование этого опыта для решения последующих частных задач обучения.

Вывод: для создания алгоритма прогнозирования продаж в рамках данной работы был выбран метод регрессии, так как он в большей степени используется для измерений показателей успешности маркетинговых

кампаний. В данной работе будет использован способ обучения с учителем, так как он позволяет решать задачи прогнозирования. Главное особенностью задачи прогнозирования является то, что объектами являются временные ряды, на основе предшествующих данных которых, требуется сделать будущий прогноз.

2 Основные понятия о Big Data

Большие данные (Big Data) — общее название для структурированных и неструктурированных данных огромных объемов, которые эффективно обрабатываются с помощью масштабируемых программных инструментов.

Анализ больших данных проводят для того, чтобы получить новую, ранее неизвестную информацию. Подобные открытия называют инсайтом, что означает озарение, догадку, внезапное понимание [1].

В таблице 1 представлены виды анализа и их характеристики

Таблица 1 – Виды анализа больших данных и их характеристики

Традиционный	Big Data
Постепенный анализ небольших пакетов данных	Анализ всего массива доступных данных
Редакция и сортировка данных перед анализом	Данные анализируются в их исходном виде
Старт с гипотезы и ее тестирования относительно данных	Поиск корреляций по всем данным до получения искомой информации
Данные собираются, обрабатываются, хранятся и лишь затем анализируются	Анализ данных в реальном времени

Понятие больших данных подразумевает работу с информацией огромного объема и разнообразного состава, весьма часто обновляемой и находящейся в разных источниках в целях увеличения эффективности работы, создания новых продуктов и повышения конкурентоспособности.

Анализ больших данных характеризуется с использованием следующих параметров:

1. Объем – определяет количество генерируемых данных. На основе этого параметра определяется является ли массив данных большими данными или нет. Этот тип данных хранится в облачной среде на SQL-серверах;

2. Многообразие – определяет категорию принадлежности больших данных. Знание категории принадлежности помогает аналитику более эффективно работать с представленной информацией;

3. Скорость - быстрота генерирования и обработки данных для решения поставленных задач;

4. Изменчивость – определяется нестабильностью данных с изменением времени;

5. Достоверность определяется качественной характеристикой представленных данных, на основе качества которых определяется точность анализа;

6. Сложность – определяется трудоемкостью процесса корреляционного анализа и построения взаимосвязей между представленными данными.

Большие данные – это не только большие массивы информации, но и методы их обработки, а также инструментарии, которые коренным образом отличаются от методов обработки обычных баз данных.

Ниже представлены характеристики, на основе которых определяются большие данные:

- Volume – характеризует физический объем данных; предоставленные данные должны представлять собой достаточно большой объем информации, который слишком трудоемко было бы обрабатывать и хранить при помощи традиционных способов;

- Velocity – скорость, как накопления данных, так и достаточная быстрота обработки данных;

- Variety – способность одновременной обработки разноформатной информации; структурированная информация может быть классифицирована, а неструктурированная требует тщательного и комплексного анализа, чтобы получить возможность её качественной дальнейшей обработки.

В таблице 2 представлены функции и задачи больших данных [2].

Таблица 2 - Функции и задачи больших данных

Функция	Задача
Big Data — собственно массивы необработанных данных	Хранение и управление большими объемами постоянно обновляющейся информации
Data mining — процесс обработки и структуризации данных, этап аналитики для выявления закономерностей	Структурирование разнообразных сведений, поиск скрытых и неочевидных связей для приведения к единому знаменателю
Machine learning — процесс машинного обучения на основе обнаруженных связей в процессе анализа	Аналитика и прогнозирование на основе обработанной и структурированной информации

К источникам больших данных относят:

- Интернет — соцсети, блоги, СМИ, форумы, сайты, интернет вещей (IoT).

- Корпоративные данные — транзакционная деловая информация, архивы, базы данных.

- Показания устройств — датчиков, приборов, а также метеорологические данные, данные сотовой связи и т. д.

Для корректного функционирования система больших данных должна быть основана на определенных принципах:

- Горизонтальная масштабируемость — любая система, которая обрабатывает большие данные должна быть расширяемой. Если объем данных вырастет в 2 раза, то количество серверов в кластере также должно быть увеличено в 2 раза.

- Отказоустойчивость — необходимое условие при большом количестве машин, которые неизбежно будут выходить из строя.

- Локальность данных — для снижения издержек данные необходимо обрабатывать на том же сервере, где они хранятся.

Активнее всего большие данные используют в финансовой и медицинской отраслях, высокотехнологичных и интернет-компаниях, а также в государственном секторе.

2.1 Обзор и выбор методик для анализа больших данных

На сегодняшний день существует огромное количество методик для анализа массивов данных, которые основаны на инструментариях, применяемых в статистике и информатике. Ниже представлены самые распространенные методики.

A/B testing. Данная методика основана на определении оптимальной комбинации показателей для достижения лучшего результата, путем сравнения контрольной выборки поочередно с другими выборками. При данной методике возможно проведение огромного количества итераций, чтобы получить наиболее статистически достоверный результат [3].

Association rule learning. Данный набор методик предназначен для поиска ассоциативных правил и для выявлений их функциональных взаимосвязей в больших данных. Методика, в частности, используется data mining.

Classification. Данный набор методик чаще всего используется в экономике для предсказания поведения потребителей. На основе данных методик определяются анализ решения о покупке товаров, применение скидочной карты и др. Методика, в частности, используется data mining.

Cluster analysis. Метод основан на определении заранее неизвестных общих признаков и последующей классификации объектов по группам. Методика, в частности, используется data mining.

Crowdsourcing. Методика производит сбор данных из большого количества открытых источников.

Data fusion and data integration. Применяется для анализа комментариев пользователей в социальных сетях и сопоставления результатов с результатами продаж товаров в режиме реального времени.

Data mining. Применяется для определения поведенческой модели потребителей и для определения категории потребителей, для которых подойдет тот или иной вид товаров или услуг.

Ensemble learning. Главная особенность данной методики в том, что в ней используется множество моделей предсказания, благодаря чему повышается точность сделанных прогнозов.

Genetic algorithms. Отличительной особенностью данной методики является возможность наиболее вероятным решениям комбинироваться и мутировать, что повышает вероятность нахождения оптимального решения.

Machine learning. Основано на создании алгоритмов самообучения. В качестве исходных данных выступают эмпирические данные.

Natural language processing. Применяется для распознавания языка человека. Основано на информатике и лингвистике.

Network analysis. Применяется для социальных сетей. Позволяет находить взаимосвязи между пользователями. Производит анализ между узлами в сетях.

Optimization. Применяется для улучшения показателей в сложных системах. Направляет в принятии стратегически важных решений для компаний.

Pattern recognition. Применяется для определения и предсказания поведения потребителей на рынке.

Predictive modeling. Применяется для создания математической модели определения вероятного сценария развития событий.

Regression. Является методом прогнозирования в статистике, для определения вероятности возникновения некоторого события с несколькими переменными. Данный метод определяет степень зависимости между категориальной зависимой и несколькими независимыми переменными.

Применяется для создания прогнозов на будущее. Методика, в частности, используется data mining.

Sentiment analysis. Основано на определении эмоционального окраса высказывания потребителей в социальных сетях о каком-либо товаре. Определяет настроение потребителей.

Signal processing. Используется для распознавания сигнала на фоне шума и последующего его анализ.

Spatial analysis. Используется для анализ пространственных данных. Определение геометрии объектов, их координат и топологии местности.

Statistics. Применяется для сбора, хранения и распознавания данных, на основе которых производится разработка опросных листов и проведение экспериментов.

Supervised learning. Основан на определении взаимосвязей среди множества массива данных и объединение объектов в группы по признакам.

Simulation. Применяется в имитационном моделировании для тестирования алгоритмов в сложных системах. Позволяет проработать различные сценарии и сделать прогноз поведения модели.

Time series analysis. Применяется для отслеживания повторяющихся последовательностей данных. Часто используется для отслеживания заболеваемости эпидемий и изменений на валютном рынке.

Unsupervised learning. При способе обучения без учителя случае ответы не задаются, и программе требуется искать функциональные зависимости между объектами самой.

Визуализация. Позволяет представлять результаты анализа в виде диаграмм для упрощения предоставляемой информации и облегчения восприятия. Также есть возможность представления информации в виде анимированных изображений.

Вывод: в данной работе была выбрана методика анализа массивов данных Regression. Данный метод является методом прогнозирования в статистике, для определения вероятности возникновения некоторого события

с несколькими переменными. Данный метод определяет степень зависимости между категориальной зависимой и несколькими независимыми переменными. Применяется для создания прогнозов на будущее. Методика, в частности, используется data mining.

3 Обзор моделей прогнозирования временных рядов

Прогноз — это научно обоснованное суждение о возможных состояниях объекта в будущем и (или) об альтернативных путях и сроках их осуществления. Под прогнозированием понимают предсказание будущего с помощью научных методов. Под процессом прогнозирования понимается специальное исследование возможностей развития некоторого процесса. В магистерской диссертации исходными данными для анализа являются продажи всех магазинов компании «ЛАМА» с 2014 по 2017 года. Данные процессы продаж описываются временными рядами, являющимися последовательностью значений продаж всех категорий товаров, полученных с 2014 по 2017 года. Данные временные ряды включают в себя два элемента: дату продажи и наименование товара, соответствующее указанной дате продажи. Каждый показатель продаж временного ряда рассматривается как выборочная реализация из общего числа популяции продаж, на который оказывают влияние множество факторов, которые будут описаны далее более подробно [4].

Все временные ряды отличаются друг от друга способом определения значения, значением временного шага, существующей памятью и наличием стационарности. В зависимости от частоты определения значений временного ряда, различают две категории временных рядов: равноотстоящие временные ряды и неравноотстоящие временные ряды.

Временные ряды первой категории появляются при исследовании и фиксации значений процесса в следующие друг за другом равные интервалы времени. Временными рядами второй категории являются те ряды, для которых принцип равенства интервалов фиксации значений не выполняется.

Из двух категорий временных рядов, представленных выше, принято выделять две подкатегории: стационарные временные ряды и нестационарные временные ряды.

Стационарными временными рядами являются те ряды, которые находятся в равновесии относительно постоянного среднего уровня. Нестационарными являются те ряды, которые не находятся в равновесии относительно постоянного среднего уровня [5].

Временем упреждения называется горизонт времени, на который необходимо определить значения временного ряда. В зависимости на какой срок делается прогноз, принято задачи прогнозирования делить на следующие категории срочности: краткосрочное прогнозирование, среднесрочное прогнозирование и долгосрочное прогнозирование.

В магистерской диссертации для построения алгоритма предсказания продаж будет рассматриваться среднесрочное прогнозирование, так как необходимо сделать тестовый прогноз на 2018 год и определить точность прогноза.

При прогнозировании временных рядов принято различать два понятия: метод прогнозирования и модель прогнозирования.

Модель прогнозирования — это функциональное представление, которое является основной для получения будущего прогноза и достаточно адекватно описывающее временной ряд.

Метод прогнозирования — это последовательность действий, благодаря выполнению которых определяется модель прогнозирования. Также метод необходим для оценки точности и качества сделанного прогноза.

Ниже представлены основные этапы построения модели прогнозирования:

1) Основываясь на предыдущем собственным опыте, выбирается модель прогнозирования для прогноза временного ряда с определенным временем упреждения;

2) Следующий шаг — это идентификация подклассов моделей для подгонки к выбранному временному ряду;

3) Далее нужно оценить параметры модели: какие параметры содержит и какова структура модели;

4) Затем проводится диагностика полученной модели прогнозирования. Необходимо выбрать участок временного ряда, достаточный по длине для составления прогноза и последующей проверке точности прогноза;

5) Если точность диагностики оказалась в пределах 70-80%, то созданная модель является приемлемой для решения поставленных задач и модель готова к использованию. Если же точность при диагностики показала значения ниже заявленных, то необходимо повторение всех описанных выше пунктов до достижения приемлемого результата.

При прогнозировании временного ряда требуется определить функциональную зависимость, адекватно описывающую временной ряд, которая называется моделью прогнозирования. Цель создания модели прогнозирования состоит в получении такой модели, для которой среднее абсолютное отклонение истинного значения от прогнозируемого стремится к минимальному для заданного горизонта, который называется временем упреждения. После того, как модель прогнозирования временного ряда определена, требуется вычислить будущие значения временного ряда [6].

3.1 Сравнение существующих моделей прогнозирования

Все методы прогнозирования делятся на две группы: интуитивные и формализованные.

Интуитивное прогнозирование применяется тогда, когда объект прогнозирования либо слишком прост, либо, напротив, настолько сложен, что аналитически учесть влияние внешних факторов невозможно. Интуитивные методы прогнозирования не предполагают разработку моделей прогнозирования и отражают индивидуальные суждения экспертов относительно перспектив развития процесса [7].

Формализованные методы рассматривают модели прогнозирования. Модели прогнозирования разделяются на статистические модели и структурные модели.

В статистических моделях функциональная зависимость между будущими и фактическими значениями временного ряда, а также внешними факторами задана аналитически. К статистическим моделям относятся следующие группы [8]:

– регрессионные модели;

– авторегрессионные модели;

– модели экспоненциального сглаживания.

В структурных моделях функциональная зависимость между будущими и фактическими значениями временного ряда, а также внешними факторами задана структурно. К структурным моделям относятся следующие группы:

– нейросетевые модели;

– модели на базе цепей Маркова;

– модели на базе классификационно-регрессионных деревьев.

Достоинства и недостатки описанных моделей и методов представлены в таблице 3 [8].

Таблица 3 – Методы прогнозирования и сравнение моделей

Модель и метод	Достоинства	Недостатки
Регрессионные методы и модели	Проектирование и единообразие анализа; гибкость, прозрачность моделирования, простота	отсутствие возможности моделирования нелинейных процессов (для нелинейной регрессии); трудоемкость нахождения коэффициентов зависимости; сложность определения функциональной зависимости
Авторегрессионные методы и модели	единообразие анализа и проектирования; прозрачность моделирования, множество примеров применения простота	низкая адаптивность; невозможность моделирования нелинейностей; ресурсоемкость и трудоемкость идентификации моделей

Методы и модели экспоненциального сглаживания	проектирования и единообразие анализа; простота моделирования	узкая применимость моделей; недостаточная гибкость
Нейросетевые методы и модели	множество примеров применения; высокая адаптивность, масштабируемость; нелинейность моделей; единообразие анализа и проектирования;	жесткие требования к обучающей выборке; ресурсоемкость процесса обучения; отсутствие прозрачности; сложность выбора алгоритма обучения; сложность выбора архитектуры
Методы и модели на базе цепей Маркова	проектирования и единообразие анализа; простота моделирования	узкая применимость моделей; невозможность моделирования процессов с длинной памятью
Методы и модели на базе классификационно-регрессионных деревьев	возможность учитывать категориальные переменные; простота процесса обучения и быстрота; масштабируемость	сложность вопроса останова; неоднозначность алгоритма построения дерева

В таблице 3 отсутствует точность прогнозирования для каждой модели и каждого метода прогнозирования. Это сделано для того, чтобы точность прогнозирования, того или иного процесса, зависели не только от модели, но и от располагаемой аппаратной мощности, от опыта исследователя, от доступности данных, и других факторов. Точность прогнозирования будет оцениваться для конкретной задачи прогнозирования, решаемой в рамках данной работы [9].

3.2 Выбор модели прогнозирования для решения поставленной задачи

На данный момент времени наиболее распространенными моделями прогнозирования являются нейросетевые модели (ANN) и авторегрессионные модели (ARIMA) [10].

Особенность временных рядов заключается в том, что прошлые значения связаны с текущими и будущими. Для среднесрочного прогнозирования рядов с устойчивой структурой достаточно данных о продажах в прошлых периодах. Для предсказания будущих значений временного ряда по настоящим и прошлым значениям в данной работе используется модель ARIMA (Авторегрессионная модель скользящего среднего), так как в области анализа временных рядов модель скользящего среднего (moving average, MA) и модель авторегрессии (autoregressive, AR) является одной из самых используемых.

В основу авторегрессионных моделей заложено предположение о том, что значение процесса X(i) линейно зависит от некоторого количества предыдущих значений того же процесса X(i-1),…, X(i-p).

Имея временной ряд X_t, модель авторегрессионного скользящего среднего позволяет достаточно точно предсказать будущие значения ряда и объяснить. Модель представляет 2 части: скользящего среднего (MA) и авторегрессионной (AR) части [11].

Для упоминания модели чаще всего используется обозначение ARIMA $(p,\ q)$, где p — порядок регрессионной части, а q — порядок скользящего среднего.

AR(p) записано таким образом:

$$X_t = c + \sum_{i=1}^{p} \varphi_i \cdot X_{i-t} + \epsilon_t,$$

где $\varphi_1 \ldots \varphi_p$ – параметры модели, c – константа, ϵ_t – белый шум.

Для упрощения константу часто опускают. По сути, это авторегрессионная модель, являющаяся полюсным фильтром с бесконечной импульсной характеристикой, истолкованным в контексте анализа временных рядов.

MA(q) записано таким образом:

$$X_t = \sum_{i=1}^{q} \theta_i \cdot \epsilon_{t-i} + \epsilon_t,$$

где $\theta_1 \dots \theta_q$ – параметры модели, $\epsilon_t \dots \epsilon_{t-q}$ – ошибки.

Можно рассматривать скользящее среднее, как интерпретацию фильтра с конечной импульсной характеристикой.

Модель ARMA (p, q) включает в себя модели AR(p) и MA(q):

$$X_t = c + \epsilon_t + \sum_{i=1}^{q} \theta_i \cdot \epsilon_{t-i} + \sum_{i=1}^{p} \varphi_i \cdot X_{i-t}$$

Для определения φ_i и C используют метод максимального правдоподобия или метод наименьших квадратов.

4 Выбор инструментов для реализации алгоритма предсказания

Процесс разработки алгоритма прогнозирования является трудоемким и сложным процессом, который требует изучения не только предметной области, но и изучения подходящих средств для реализации. При выборе тех или иных средств реализации необходимо уделить особое внимание на возможность решения поставленных задач с помощью данных средств. Разработка любого алгоритма предсказания основывается на методах, способах и решаемых задач. Основной задачей данной работы является разработка алгоритма предсказания продаж, где на основе загруженных данных необходимо не только вывести эти данные, но и построить прогноз на год. Вывод данных предполагает графическое изображение нестационарного временного ряда.

Одним из важнейших этапов на пути к решению поставленной задачи является выбор среды для разработки алгоритма предсказания продаж, который основывается на требованиях, предъявленных разрабатываемому алгоритму. Для осуществления выбора наиболее подходящей среды для разработки необходимо провести сравнительный анализ.

4.1 Выбор языка программирования

В настоящий момент существует огромное разнообразие языков программирования высокого уровня. Наиболее популярными на сегодня являются языки, в числе которых можно выделить C++, Java, C#, Python. C++является родителем по отношению к Java и C#. Несмотря на модификацию некоторых свойств они имеют схожий синтаксис и объектную модель. Однако, они имеют разное предназначение [12].

C++ предназначен для создания высокопроизводительных программ и программы, написанные на этом языке, не являются кроссплатформенными, то есть они создаются для конкретного типа процессора и определенной

операционной системы. Компилятор формирует машинный код, который затем выполняется центральным процессором. Для запуска разработанной программы в иной среде необходимо построить новый машинный код, который предназначен для данной среды.

В отличии от C++, программы, написанные на языке Java и C#, являются переносимыми, что является их основным достоинством. Это значит, что в отличии от C++, программы написанные на Java и C# не привязаны к характеристикам архитектуры компьютера. Кроссплатформенность объясняется формированием псевдокода при компиляции программы.

Псевдокод представляет собой некоторый промежуточный язык, который выполняется с помощью специальной исполняющей системы. Для Java — это JVM (Java Virtual Machine—виртуальная машина), для C# — это CLR (Common Language Runtime). В результате, код, запущенный в разных операционных системах, будет работать одинаково [13].

Python поддерживает не только парадигму объектно-ориентированного программирования, но и структурного, функционального, императивного и аспектно-ориентированного. В отличии от C++, C# и Java является динамически типизируемым. Эталонная реализация языка Python-это интерпретатор CPython, также есть реализации интерпретаторов для JVM и MSIL (с возможностью компиляции) и др.

Как популярный язык, обеспечивающий сокращение времени, затрачиваемого на разработку программ, Python используется для создания широкого круга программ в самых разных областях. В число пользователей Python в настоящее время входят Google, YouTube, Industrial Light & Magic, ESRI, системы BitTorrent обмена файлами, Jet Propulsion Lab в NASA, игра Eve Online и National Weather Service (национальная метеорологическая служба, США). Язык Python используется в разнообразных областях, от обучения, администрирования систем, создания сценариев для мобильных устройств и разработки веб-сайтов, до управления космическими кораблями, анализа капиталовложений, тестирования аппаратуры, компьютерных игр. Среди

остальных достоинств Python отличается простым синтаксисом, удобочитаемостью, удивительной простотой; обладает большой коллекцией готовых утилит и интерфейсов; поддерживает модульное программирование, функциональное и объектно-ориентированное, имеет мультипарадигменную архитектуру; просто интегрируется с внешними компонентами, написанными на других языках программирования. Набор встроенных инструментальных средств делает его динамичным и гибким языком программирования. Несмотря на свое назначение, Python очень часто называют языком сценариев, так как он может позволить с легкостью и простотой управлять другими программными компонентами и использовать их. Одним из основных преимуществ языка Python является наличие полной стандартной библиотеки, которая решает большое количество задач без необходимости подключения сторонних библиотек. Так же, библиотеки от сторонних производителей, обеспечивающие возможности, отсутствующие в стандартной библиотеке, насчитываются в количестве 136462 [14].

Для сравнения вышеперечисленных языков можно выделить следующие требования к языку общего назначения в рамках данной работы:

– лаконичность кода;

– читаемость кода;

– порог вхождения при изучении и использовании языка;

– кроссплатформенность;

– скорость исполнения программ;

– использование памяти;

– библиотеки для работы с данными.

В качестве оценок используется шкала от 1 до 4, где 1 – плохо, а 4 – отлично. Оценки представлены в таблице 4 и ставятся на основе изученных о языках сведений [15].

Таблица 4 – Сравнение языков программирования

Характеристики	C++	Java	Python	C#
Лаконичность	1	3	4	3
Читаемость	1	3	4	3
Порог вхождения	1	2	3	2
Кроссплатформенность	4	4	4	1
Скорость исполнения	4	3	2	3
Использование памяти	4	2	3	3
Библиотеки для работы с данными	2	2	4	1
Общая оценка	17	19	24	16

При выборе языка программирования особое внимание уделялось возможности графического отображения временного ряда и быстроте вычислений большого объема данных. Сравнение общих оценок представленных языков программирования по заданным характеристикам показывает, что для решения поставленных в работе задач лучше всего подходит Python.

4.2 Выбор библиотек для реализации алгоритма

В данной работе используются достаточно большие данные, для обработки которых необходимо использовать библиотеку, позволяющую быстро обрабатывать вычисления. Данные временного ряда будут представлять из себя массив, для работы с которыми было принято решение использовать библиотеку NumPy.

NumPy — это библиотека, которая предоставляет открытый исходный код и предназначен для языка программирования Python.

Numpy может предоставить [16]:

– эффективный и быстрый объект многомерного массива ndarray;

– функции для математических операций с несколькими массивами или выполнения вычислений над элементами одного массива;

– средства для записи и чтения на диски наборов данных, которые представляются в виде массивов;

– генератор случайных чисел, преобразование Фурье и операции линейной алгебры;

– средства для интеграции с кодом, написанным на Fortran, C или C++.

Математические алгоритмы, которые реализованы на таких языках как Python(интерпретируемые), как правило, зачастую тратят больше времени на обработку алгоритмов по сравнению с теми же алгоритмами, реализованными на компилируемых языках (C, Java).

Для более быстрой обработки алгоритмов библиотекой Numpy предоставляются реализации алгоритмов, которые оптимизированы для обработки многомерных массивов. Они представляют из себя функции и операторы. В результате, скорость выполнения алгоритмов, представляющих из себя последовательность операций над массивами, ничуть не уступает эквивалентному коду в MATLAB. Библиотеку NumPy можно считать альтернативой MATLAB.

MATLAB и NumPy схожи, ввиду их интерпретируемости и возможности оперировать массивами, а не над скалярными значениями. Достоинством MATLAB является то, что он предоставляет огромное количество полезных пакетов, например, Simulink. Однако, в NumPy также имеются похожие пакеты, такие как SciPy или Matplotlib.

SciPy позволяет использовать схожую с MATLAB функциональность, а библиотека Matplotlib предоставляет возможность создавать графики в стиле MATLAB.

Исследовав существующие библиотеки для реализации визуализации временного ряда, выбор был остановлен на библиотеке Matplotlib, которая позволяет строить качественные двумерные графики. Основным преимуществом данной библиотеки является скорость построения графиков. С помощью Matplotlib можно создавать графики, гистограммы, спектры мощности, диаграммы ошибок, диаграммы рассеяния и т.д. В данной работе

был использован модуль pyplot из Matplotlib (import matplotlib.pyplot as plt), который предоставляет интерфейс подобный MATLAB. Таким образом есть возможность использовать набор функций, используемых в MATLAB. С помощью данного модуля можно не только строить графики, но и контролировать стили, свойства осей и т.д.

Для работы с данными в данной работе была использована библиотека pandas. Pandas — это библиотека языка Python, которая является высокоуровневой и предназначена для анализа данных. Данная библиотека высокоуровневая, т. к. написана на языке C, что является достоинством, т.к. позволяет повысить производительность. В мире Python, библиотека pandas считается наиболее продвинутой и быстроразвивающейся библиотекой для обработки и анализа данных. Для взаимодействия с пользователем необходимо создать для программы удобный интерфейс [16].

Библиотека pandas объединяет в себе высокую производительность средств работы с массивами, присущую numpy, реляционным базам данных (как на основе SQL), свойственными электронным таблицам и с гибкими возможностями манипулирования данными. Pandas предоставляет развитые средства индексирования, позволяющие без особых усилий выбирать подмножества, менять форму наборов данных, выполнять агрегирование, формировать продольные и поперечные срезы. Для разработки финансовых приложений pandas предлагает большой набор высокопроизводительных средств анализа временных рядов, специально ориентированных на финансовые данные. Само название pandas образовано как от panel data (панельные данные), которые применяются в эконометрике термина для обозначения многомерных структурированных наборов данных, так и от фразы Python data analysis.

В данной работе для разработки интерфейса был использован библиотека PyQt5. PyQt5 — это набор библиотек Python, служит для разработки графического интерфейса на базе платформы Qt5 от компании Digia. PyQt5 представляет из себя набор Python-модулей. Он является

мультиплатформенным, а следовательно, может запускается на различных операционных системах, таких как Unix, Windows и MacOS.

5 Анализ факторов влияющих на покупательский спрос

Самой случайной компонентой любой модели управления запасами в торговом предприятии это покупательский спрос. При недостаточном его изучении, чаще всего, спрос принимается как детерминированная составляющая. Обычно это приводит к отклонениям в оценках необходимых товарных запасов. Для качественного управления запасов необходима более точная оценка покупательского спроса. Множество факторов, которые влияют на характер спроса, различающихся по природе. Имеются следующие группы факторов:

– социальные: уровень развития культуры, социальная структура общества, уровень безработицы, профессиональный состав населения, мода;

– экономические: размер денежных доходов, уровень развития экономики региона, страны, соотношение товарной и нетоварной форм потребления, уровень и динамика розничных цен, уровень цен на коммунальные услуги, культура торговли, степень обеспеченности покупателей товарами, величина налогов;

– демографические: половозрастной состав потребителей, численность, миграция населения, размер и состав семьи;

– природно-климатические: сезонность продаж, национальные традиции, специфика климата;

– политические: индексация доходов, меры по поддержке малоимущих потребителей, международные кризисы, минимальный уровень заработной платы и прожиточного минимума, политические конфликты, войны;

– товарная политика: насыщенность рынка товарами, качество товаров, широта ассортимента, современная техника и технология производства, взаимодополняющих товаров, наличие товаров-заменителей;

– прочие факторы: реклама, потребительские ожидания, потребителей, предпочтения сегмент рынка.

Вывод: на основе проведенного анализа факторов и состава данных были выбраны техники предсказания продаж в зависимости от сезонности продаж определенного товара. Наступление определенного сезона, предположительно, имеет сильное влияние на объем товарного спроса, а также на состав покупательской корзины.

6 Программная реализация алгоритма предсказания

Jupyter Notebook — это веб-приложение с открытым исходным кодом, которое можно использовать для создания и обмена документами, содержащими живой код, уравнения, визуализации и текст.

Jupyter Notebooks — это дополнительный проект из проекта IPython, в котором раньше был сам проект IPython Notebook. Название Jupyter происходит от основных поддерживаемых языков программирования, которые он поддерживает: Julia, Python и R. Jupyter поставляется с ядром IPython, которое позволяет вам писать программы на Python, но в настоящее время существует более 100 других ядер, которые вы также можно использовать.

Чтобы инициализировать процесс, необходимо установить зависимости, которые нужны для среды программирования Python, из репозиториев Ubuntu. Ubuntu 16.04 поставляется с предустановленной Python 3.6.

Сначала необходимо обновить локальный индекс пакетов apt, а затем загрузить и установить пакеты. На рисунке 6.1 представлена команда обновления локального индекса пакетов.

```
$ sudo apt update
```

Рисунок 6.1 – Команда обновления локального индекса пакетов

Затем установить pip и заголовочные файлы Python, которые используются некоторыми зависимостями Jupyter. На рисунке 6.2 представлена команда установки pip.

```
$ sudo apt install python3-pip python3-dev
```

Рисунок 6.2 – Команда установки pip

Теперь можно перейти к настройке виртуальной среды Python, в которую будет установлен Jupyter.

Для этого сначала нужен доступ к команде virtualenv, которую устанавливаем с помощью pip. На рисунке 6.3 представлены команды настройки виртуальной среды Python.

```
$ sudo -H pip3 install --upgrade pip
$ sudo -H pip3 install virtualenv
```

Рисунок 6.3 – Команды настройки виртуальной среды Python

Флаг -H гарантирует, что политика безопасности устанавливает переменную домашней среды в домашний каталог целевого пользователя.

Далее создаем каталог, в котором будут находиться файлы проекта. На рисунке 6.4 представлены команды создания каталога.

```
$ mkdir ~/my_project_dir
$ cd ~/my_project_dir
```

Рисунок 6.4 – Команды для создания рабочего каталога

В каталоге проекта создаем виртуальную среду Python. На рисунке 6.5 представлена команда создания виртуальной среды Python.

```
$ virtualenv my_project_env
```

Рисунок 6.5 – Команда для создания виртуальной среды Python

Это команда создает каталог my_project_env в каталоге my_project_dir. Внутри будет установлена локальная версия Python и локальная версия pip. Далее этот каталог используется для установки и настройки изолированной

среды Python для Jupyter. На рисунке 6.6 представлена команда установки Jupyter.

```
$ pip install jupyter
```

Рисунок 6.6 – Команда для установки Jupyter

Команда, представленная выше, устанавливает все программное обеспечение, необходимое для запуска Jupyter.

Чтобы запустить Jupyter, выполняем команду, представленную на рисунке 6.7. Пример работы в Jupyter представлен на рисунке 6.8.

```
jupyter notebook
```

Рисунок 6.7 – Команда запуска Jupyter

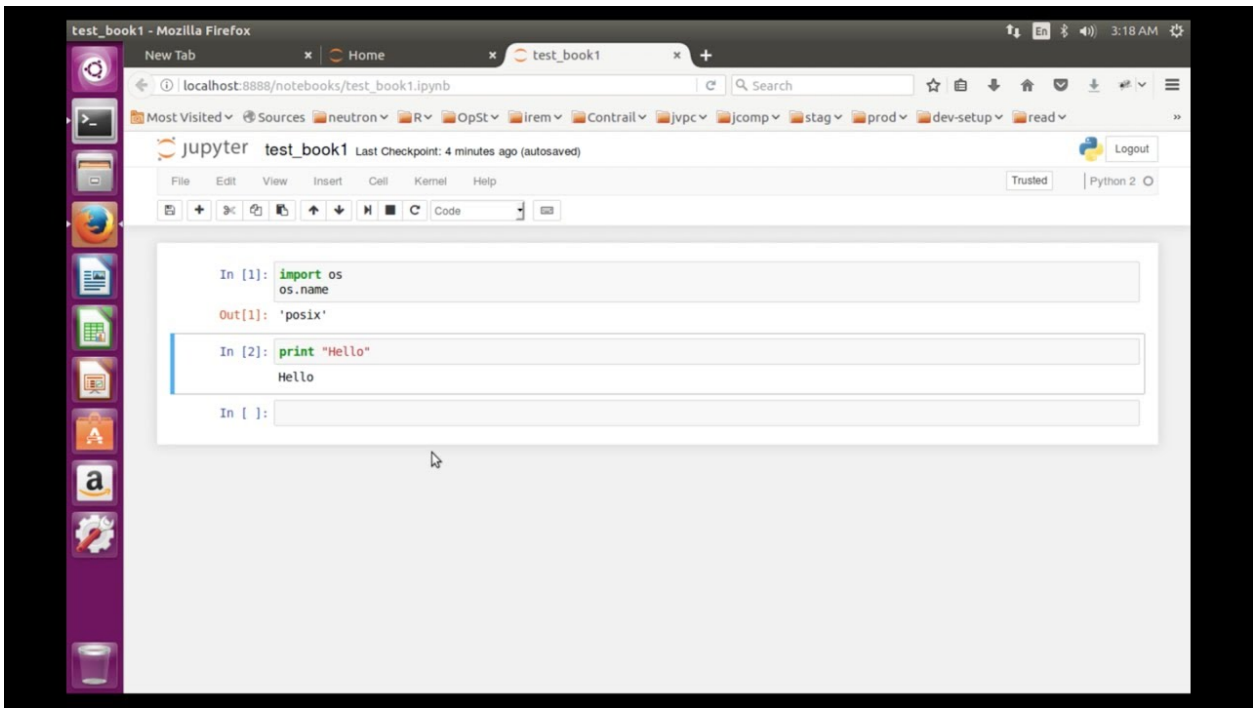

Рисунок 6.8 – Пример работы в Jupyter

Для построения модели предсказания используются данные продаж всех категорий продукции магазинов компании «ЛАМА» города Томска во временном интервале с 2014 по 2017 года.

На рисунке 6.9 представлены данные о продажах всех категорий продукции магазинов компании «ЛАМА» города Томска.

Row ID	Order ID	Order Date	Ship Date	Postal Code	Product ID	Category	Sales	Quantity	Discount	Profit
1	CA-2016-152156	08.11.2016	11.11.2016	42420	FUR-BO-1000	Perishable	261,96	2	0	41,9136
2	CA-2016-152156	08.11.2016	11.11.2016	42420	FUR-CH-1000	Perishable	731,94	3	0	219,582
3	CA-2016-138688	12.06.2016	16.06.2016	90036	OFF-LA-1000	Vegetables	14,62	2	0	6,8714
4	US-2015-108966	11.10.2015	18.10.2015	33311	FUR-TA-1000	Perishable	957,5775	5	0,45	-383,031
5	US-2015-108966	11.10.2015	18.10.2015	33311	OFF-ST-1000	Vegetables	22,368	2	0,2	2,5164
6	CA-2014-115812	09.06.2014	14.06.2014	90032	FUR-FU-1000	Perishable	48,86	7	0	14,1694
7	CA-2014-115812	09.06.2014	14.06.2014	90032	OFF-AR-1000	Vegetables	7,28	4	0	1,9656
8	CA-2014-115812	09.06.2014	14.06.2014	90032	TEC-PH-1000	Household cl	907,152	6	0,2	90,7152
9	CA-2014-115812	09.06.2014	14.06.2014	90032	OFF-BI-10003	Vegetables	18,504	3	0,2	5,7825
10	CA-2014-115812	09.06.2014	14.06.2014	90032	OFF-AP-1000	Vegetables	114,9	5	0	34,47
11	CA-2014-115812	09.06.2014	14.06.2014	90032	FUR-TA-1000	Perishable	1706,184	9	0,2	85,3092
12	CA-2014-115812	09.06.2014	14.06.2014	90032	TEC-PH-1000	Household cl	911,424	4	0,2	68,3568
13	CA-2017-114412	15.04.2017	20.04.2017	28027	OFF-PA-1000	Vegetables	15,552	3	0,2	5,4432
14	CA-2016-161389	05.12.2016	10.12.2016	98103	OFF-BI-10003	Vegetables	407,976	3	0,2	132,5922
15	US-2015-118983	22.11.2015	26.11.2015	76106	OFF-AP-1000	Vegetables	68,81	5	0,8	-123,858
16	US-2015-118983	22.11.2015	26.11.2015	76106	OFF-BI-10000	Vegetables	2,544	3	0,8	-3,816
17	CA-2014-105893	11.11.2014	18.11.2014	53711	OFF-ST-10004	Vegetables	665,88	6	0	13,3176
18	CA-2014-167164	13.05.2014	15.05.2014	84084	OFF-ST-1000	Vegetables	55,5	2	0	9,99
19	CA-2014-143336	27.08.2014	01.09.2014	94109	OFF-AR-1000	Vegetables	8,56	2	0	2,4824
20	CA-2014-143336	27.08.2014	01.09.2014	94109	TEC-PH-1000	Household cl	213,48	3	0,2	16,011
21	CA-2014-143336	27.08.2014	01.09.2014	94109	OFF-BI-10002	Vegetables	22,72	4	0,2	7,384
22	CA-2016-137330	09.12.2016	13.12.2016	68025	OFF-AR-1000	Vegetables	19,46	7	0	5,0596
23	CA-2016-137330	09.12.2016	13.12.2016	68025	OFF-AP-1000	Vegetables	60,34	7	0	15,6884
24	US-2017-156909	16.07.2017	18.07.2017	19140	FUR-CH-1000	Perishable	71,372	2	0,3	-1,0196

Рисунок 6.9 – Данные о продажах всех категорий продукции магазинов компании «ЛАМА» города Томска

Первым шагом для создания алгоритма прогнозирования продаж является загрузка массива данных о продажах в NumPy и подгрузка необходимых библиотек для его реализации. На рисунке 6.10 представлен процесс загрузки данных.

```
import warnings
import itertools
import numpy as np
import matplotlib.pyplot as plt
warnings.filterwarnings("ignore")
plt.style.use('fivethirtyeight')
import pandas as pd
import statsmodels.api as sm
import matplotlib

matplotlib.rcParams['axes.labelsize'] = 14
matplotlib.rcParams['xtick.labelsize'] = 12
matplotlib.rcParams['ytick.labelsize'] = 12
matplotlib.rcParams['text.color'] = 'k'
```

Рисунок 6.10 – Процесс загрузки данных

В данных о продажах в супермаркетах есть несколько категорий, мы начинаем с анализа временных рядов и прогнозирования продаж категории Perishable (скоропортящаяся продукция).

Для этого из всего массива данных выбираем ту категорию, которая нас интересует. На рисунке 6.11 показана выборка категории.

```
df = pd.read_excel("Giperstore.xls")
perishable = df.loc[df['Category'] == 'Perishable']
```

Рисунок 6.11 – Выборка категории скоропортящаяся продукция

Далее сортируем данные о продажах скоропортящейся продукции за 4 года. На рисунке 6.12 показана сортировка данных о продажах.

```
perishable ['Order Date'].min()
perishable ['Order Date'].max()
```

Рисунок 6.12 – Сортировка данных о продажах

Далее следует предварительная обработка данных. Этот шаг включает удаление ненужных нам столбцов, проверку пропущенных значений, агрегирование продаж по дате и т. д. На рисунке 6.13 представлена предварительная обработка данных.

```
cols = ['Row ID', 'Order ID', 'Ship Date', 'Postal Code', 'Product ID', 'C
ategory', 'Quantity', 'Discount', 'Profit']
pershiable.drop(cols, axis=1, inplace=True)
pershiable = pershiable.sort_values('Order Date')
pershiable.isnull().sum()
```

Рисунок 6.13 – Предварительная обработка данных

Следующий шаг — это индексирование с данными временных рядов. На рисунке 6.14 показан результат индексации.

```
DatetimeIndex(['2014-01-06', '2014-01-07', '2014-01-10', '2014-01-11',
               '2014-01-13', '2014-01-14', '2014-01-16', '2014-01-19',
               '2014-01-20', '2014-01-21',
               ...
               '2017-12-18', '2017-12-19', '2017-12-21', '2017-12-22',
               '2017-12-23', '2017-12-24', '2017-12-25', '2017-12-28',
               '2017-12-29', '2017-12-30'],
              dtype='datetime64[ns]', name='Order Date', length=889, freq=None)
```

Рисунок 6.14 – Результат индексации

Так как использование точного времени продаж определенного товара затрудняет создание алгоритма предсказания, будем использовать средние значения ежедневных продаж за этот месяц, а в качестве метки времени будем использовать начало каждого месяца. На рисунке 6.15 показан пример данных по продажам за 2017 год в каждый месяц.

```
Order Date
2017-01-01     397.602133
2017-02-01     528.179800
2017-03-01     544.672240
2017-04-01     453.297905
2017-05-01     678.302328
2017-06-01     826.460291
2017-07-01     562.524857
2017-08-01     857.881889
2017-09-01    1209.508583
2017-10-01     875.362728
2017-11-01    1277.817759
2017-12-01    1256.298672
Freq: MS, Name: Sales, dtype: float64
```

Рисунок 6.15 – Пример данных по продажам за 2017 год в каждый месяц

На рисунке 6.16 показан график продаж скоропортящейся продукции с 2014 по 2017 года.

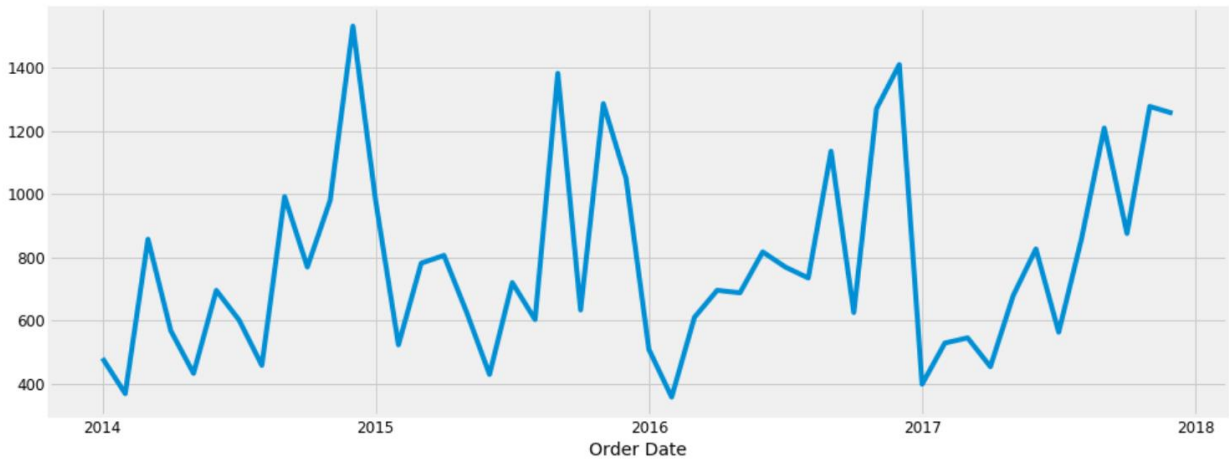

Рисунок 6.16 – График продаж скоропортящейся продукции с 2014 по 2017 года

Временные ряды имеют характер сезонности, например, продажи всегда низкие в начале года и высокие в конце года. В течение одного года всегда наблюдается тенденция к росту с несколькими низкими месяцами в середине года.

Используя метод декомпозиции временных рядов, разобьем наш временной ряд на два отдельных компонента: тренд и сезонность. На рисунке 6.17 показан метод декомпозиции временных рядов.

```
from pylab import rcParams
rcParams['figure.figsize'] = 18, 8

decomposition = sm.tsa.seasonal_decompose(y, model='additive')
fig = decomposition.plot()
plt.show()
```

Рисунок 6.17 – Метод декомпозиции временных рядов

На рисунке 6.18 показан результат декомпозиции.

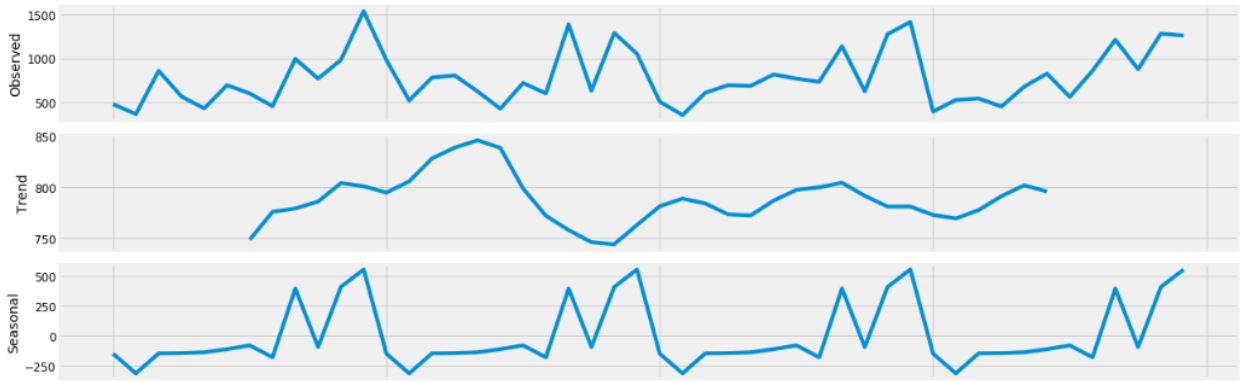

Рисунок 6.18 – Результат декомпозиции

График выше ясно показывает, что продажи скоропортящейся продукции нестабильны, наряду с очевидной сезонностью.

6.1 Прогнозирование временных рядов с использованием модели ARIMA

На рисунке 6.19 представлен выбор параметров для реализации модели ARIMA.

```
p = d = q = range(0, 2)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]

print('Examples of parameter combinations for Seasonal ARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))
```

Рисунок 6.19 – Выбор параметров для реализации модели ARIMA

На рисунке 6.20 представлен поиск оптимального набора параметров, который дает наилучшую производительность для нашей модели.

```
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(y,
                                            order=param,
                                            seasonal_order=param_seasonal,
                                            enforce_stationarity=False,
                                            enforce_invertibility=False)

            results = mod.fit()

            print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic))
        except:
            continue
```

Рисунок 6.20 – Поиск оптимального набора параметров для модели

На рисунке 6.21 представлены результаты поиска оптимального набора параметров.

```
ARIMA(0, 0, 0)x(0, 0, 1, 12)12 - AIC:1131.2657078645939
ARIMA(0, 0, 0)x(1, 0, 0, 12)12 - AIC:497.23144334183365
ARIMA(0, 0, 0)x(1, 0, 1, 12)12 - AIC:1001.3915524374769
ARIMA(0, 0, 0)x(1, 1, 0, 12)12 - AIC:318.0047199116341
ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:720.9252270758095
ARIMA(0, 0, 1)x(0, 0, 1, 12)12 - AIC:2876.7174897071977
ARIMA(0, 0, 1)x(0, 1, 0, 12)12 - AIC:466.56074298091255
ARIMA(0, 0, 1)x(1, 0, 0, 12)12 - AIC:499.54290594685824
ARIMA(0, 0, 1)x(1, 0, 1, 12)12 - AIC:2461.517421827548
ARIMA(0, 0, 1)x(1, 1, 0, 12)12 - AIC:319.98848769468657
ARIMA(0, 1, 0)x(0, 0, 1, 12)12 - AIC:1287.5697512865586
ARIMA(0, 1, 0)x(1, 0, 0, 12)12 - AIC:497.78896630044073
ARIMA(0, 1, 0)x(1, 0, 1, 12)12 - AIC:1388.8924232046936
ARIMA(0, 1, 0)x(1, 1, 0, 12)12 - AIC:319.7714068109211
ARIMA(0, 1, 1)x(0, 0, 0, 12)12 - AIC:649.9056176816999
ARIMA(0, 1, 1)x(0, 0, 1, 12)12 - AIC:3307.7208814993064
ARIMA(0, 1, 1)x(0, 1, 0, 12)12 - AIC:458.8705548482932
ARIMA(0, 1, 1)x(1, 0, 0, 12)12 - AIC:486.18329774427826
ARIMA(0, 1, 1)x(1, 0, 1, 12)12 - AIC:2625.602326434297
ARIMA(0, 1, 1)x(1, 1, 0, 12)12 - AIC:310.75743684172994
ARIMA(1, 0, 0)x(0, 0, 0, 12)12 - AIC:692.1645522067712
ARIMA(1, 0, 0)x(0, 0, 1, 12)12 - AIC:1399.3709974017943
ARIMA(1, 0, 0)x(0, 1, 0, 12)12 - AIC:479.46321478521355
ARIMA(1, 0, 0)x(1, 0, 0, 12)12 - AIC:480.92593679352177
ARIMA(1, 0, 0)x(1, 0, 1, 12)12 - AIC:1431.0752736869172
ARIMA(1, 0, 0)x(1, 1, 0, 12)12 - AIC:304.4664675084554
ARIMA(1, 0, 1)x(0, 0, 0, 12)12 - AIC:665.779444218685
ARIMA(1, 0, 1)x(0, 0, 1, 12)12 - AIC:246116.34689777798
ARIMA(1, 0, 1)x(0, 1, 0, 12)12 - AIC:468.3685195814987
ARIMA(1, 0, 1)x(1, 0, 0, 12)12 - AIC:482.5763323876739
ARIMA(1, 0, 1)x(1, 0, 1, 12)12 - AIC:3365796.8535189056
ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:306.0156002122138
ARIMA(1, 1, 0)x(0, 0, 0, 12)12 - AIC:671.2513547541902
ARIMA(1, 1, 0)x(0, 0, 1, 12)12 - AIC:1393.2157168383435
ARIMA(1, 1, 0)x(0, 1, 0, 12)12 - AIC:479.2003422281134
ARIMA(1, 1, 0)x(1, 0, 0, 12)12 - AIC:475.34036587848493
ARIMA(1, 1, 0)x(1, 0, 1, 12)12 - AIC:2102.468501404909
ARIMA(1, 1, 0)x(1, 1, 0, 12)12 - AIC:300.6270901345443
ARIMA(1, 1, 1)x(0, 0, 0, 12)12 - AIC:649.0318019835024
ARIMA(1, 1, 1)x(0, 0, 1, 12)12 - AIC:2603.9208285600357
ARIMA(1, 1, 1)x(0, 1, 0, 12)12 - AIC:460.4762687610111
ARIMA(1, 1, 1)x(1, 0, 0, 12)12 - AIC:469.52503546608614
ARIMA(1, 1, 1)x(1, 0, 1, 12)12 - AIC:2586.7750340396897
ARIMA(1, 1, 1)x(1, 1, 0, 12)12 - AIC:297.7875439553055
```

Рисунок 6.21 – Результаты поиска оптимального набора параметров для модели

Приведенные выше результаты показывают, что SARIMAX (1, 1, 1) x (1, 1, 0, 12) дает самое низкое значение AIC - 297,78. Поэтому считаем это оптимальным вариантом.

6.2 Проверка точности прогноза

Чтобы определить точность прогнозов, сравниваем прогнозируемые продажи с реальными продажами временного ряда и устанавливаем прогнозы, начиная с 2015–01–01 и заканчивая концом данных.

На рисунке 6.22 представлено определение точности прогноза на 2015 год.

```
pred = results.get_prediction(start=pd.to_datetime('2017-01-01'), dynamic=False)
pred_ci = pred.conf_int()

ax = y['2014':].plot(label='observed')
pred.predicted_mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7))

ax.fill_between(pred_ci.index,
                pred_ci.iloc[:, 0],
                pred_ci.iloc[:, 1], color='k', alpha=.2)

ax.set_xlabel('Date')
ax.set_ylabel('Furniture Sales')
plt.legend()

plt.show()
```

Рисунок 6.22 – Определение точности прогноза

На рисунке 6.23 представлен график прогноза модели ARIMA на 2016 год.

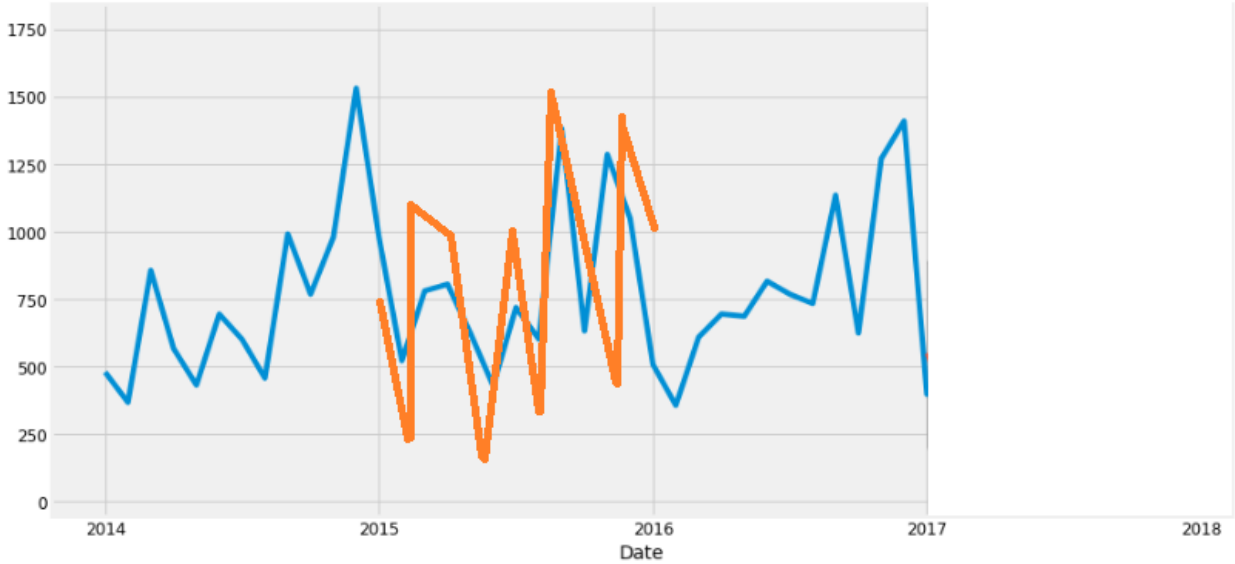

Рисунок 6.23 – График прогноза модели ARIMA на 2016 год

График показывает наблюдаемые значения по сравнению с созданным прогнозом. В целом прогнозы очень хорошо согласуются с истинными значениями, показывая, что восходящая тенденция начинается с начала года и отражает сезонность к концу года.

На рисунке 6.24 показано вычисление среднеквадратичной ошибки.

```
y_forecasted = pred.predicted_mean
y_truth = y['2017-01-01':]

mse = ((y_forecasted - y_truth) ** 2).mean()
print('The Mean Squared Error of our forecasts is
{}'.format(round(mse, 2)))
```

Рисунок 6.24 – Вычисление среднеквадратичной ошибки

Среднеквадратичная ошибка прогнозов составляет 1122993,58.

Данная модель показала точность прогноза равным 43,54%, что не удовлетворят заявленной в задании точности. Необходимо сделать повторное обучение и провести анализ точности.

На рисунке 6.25 представлено определение точности прогноза на 2017 год.

```
pred = results.get_prediction(start=pd.to_datetime('2017-01-01'), dynamic=False)
pred_ci = pred.conf_int()

ax = y['2014':].plot(label='observed')
pred.predicted_mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7))

ax.fill_between(pred_ci.index,
                pred_ci.iloc[:, 0],
                pred_ci.iloc[:, 1], color='k', alpha=.2)

ax.set_xlabel('Date')
ax.set_ylabel('Furniture Sales')
plt.legend()

plt.show()
```

Рисунок 6.25 – Определение точности прогноза

На рисунке 6.26 представлен график прогноза модели ARIMA на 2017 год.

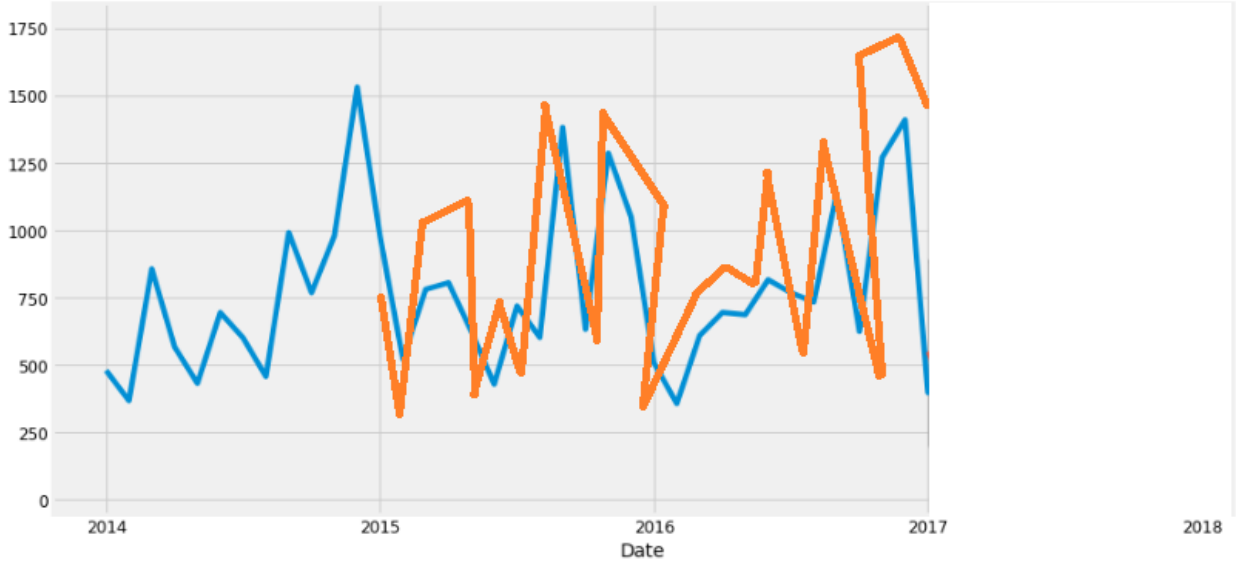

Рисунок 6.26 – График прогноза модели ARIMA на 2017 год

На рисунке 6.27 показано вычисление среднеквадратичной ошибки.

```
y_forecasted = pred.predicted_mean
y_truth = y['2017-01-01':]

mse = ((y_forecasted - y_truth) ** 2).mean()
print('The Mean Squared Error of our forecasts is
{}'.format(round(mse, 2)))
```

Рисунок 6.27 – Вычисление среднеквадратичной ошибки

Среднеквадратичная ошибка прогнозов составляет 126493,58.

Данная модель показала точность прогноза равным 62,52%, что не удовлетворят заявленной в задании точности. Необходимо сделать повторное обучение и провести анализ точности.

На рисунке 6.28 представлено определение точности прогноза на 2018 год.

```
pred = results.get_prediction(start=pd.to_datetime('2017-01-01'), dynamic=False)
pred_ci = pred.conf_int()

ax = y['2014':].plot(label='observed')
pred.predicted_mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7, figsize=(14, 7))

ax.fill_between(pred_ci.index,
                pred_ci.iloc[:, 0],
                pred_ci.iloc[:, 1], color='k', alpha=.2)

ax.set_xlabel('Date')
ax.set_ylabel('Furniture Sales')
plt.legend()

plt.show()
```

Рисунок 6.28 – Определение точности прогноза

На рисунке 6.29 представлен график прогноза модели ARIMA на 2018 год.

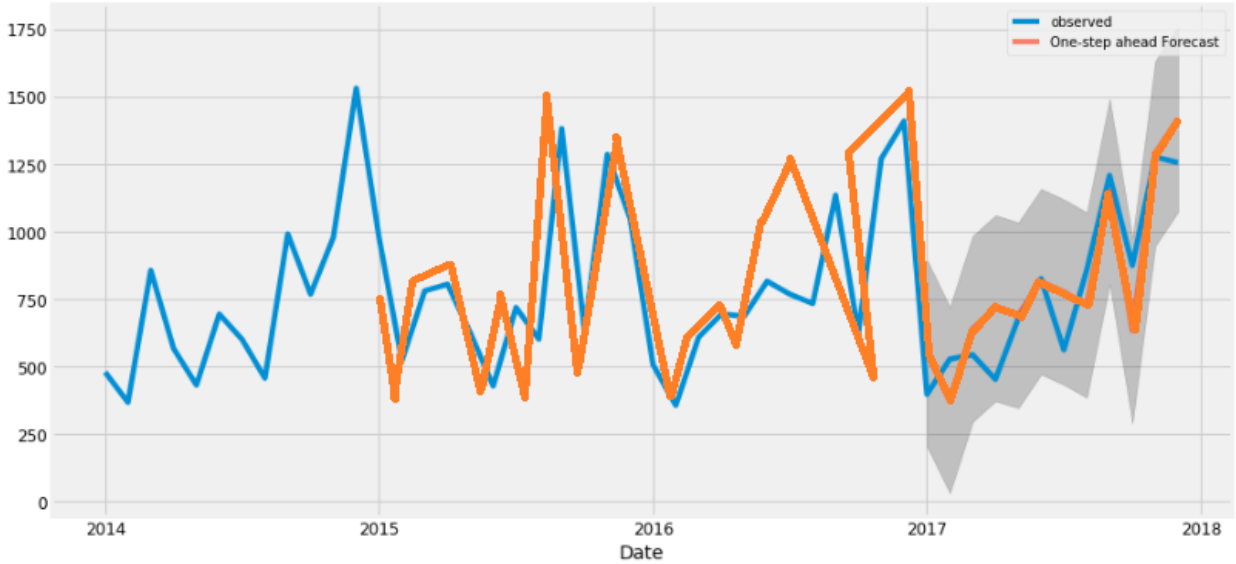

Рисунок 6.29 – График прогноза модели ARIMA на 2018 год

На рисунке 6.30 показано вычисление среднеквадратичной ошибки.

```
y_forecasted = pred.predicted_mean
y_truth = y['2017-01-01':]

mse = ((y_forecasted - y_truth) ** 2).mean()
print('The Mean Squared Error of our forecasts is
{}'.format(round(mse, 2)))
```

Рисунок 6.30 – Вычисление среднеквадратичной ошибки

Среднеквадратичная ошибка прогнозов составляет 22993,58.

Корень среднеквадратичной ошибки равен 151.64. Это говорит о том, что модель смогла прогнозировать среднесуточные продажи скоропортящейся продукции в тестовом наборе в пределах 151,64 от реальных продаж. Ежедневные продажи скоропортящейся продукции из исходного массива варьируются от 400 до 1200. Точность прогноза составляет 71,89%, что удовлетворяет заявленной в задании точности.

6.3 Создание долгосрочного прогноза

На рисунке 6.24 представлен листинг долгосрочного алгоритма прогнозирования продаж скоропортящейся продукции до 2025 года на основе созданного алгоритма прогнозирования на 2018 год.

```
pred_uc = results.get_forecast(steps=100)
pred_ci = pred_uc.conf_int()
ax = y.plot(label='observed', figsize=(14, 7))
pred_uc.predicted_mean.plot(ax=ax, label='Forecast')
ax.fill_between(pred_ci.index,
                pred_ci.iloc[:, 0],
                pred_ci.iloc[:, 1], color='k', alpha=.25)
ax.set_xlabel('Date')
ax.set_ylabel('Perishable Sales')
plt.legend()
plt.show()
```

Рисунок 6.24 – Прогнозирование до 2025 года

На рисунке 6.25 показан график прогноза продаж скоропортящейся продукции до 2025 года.

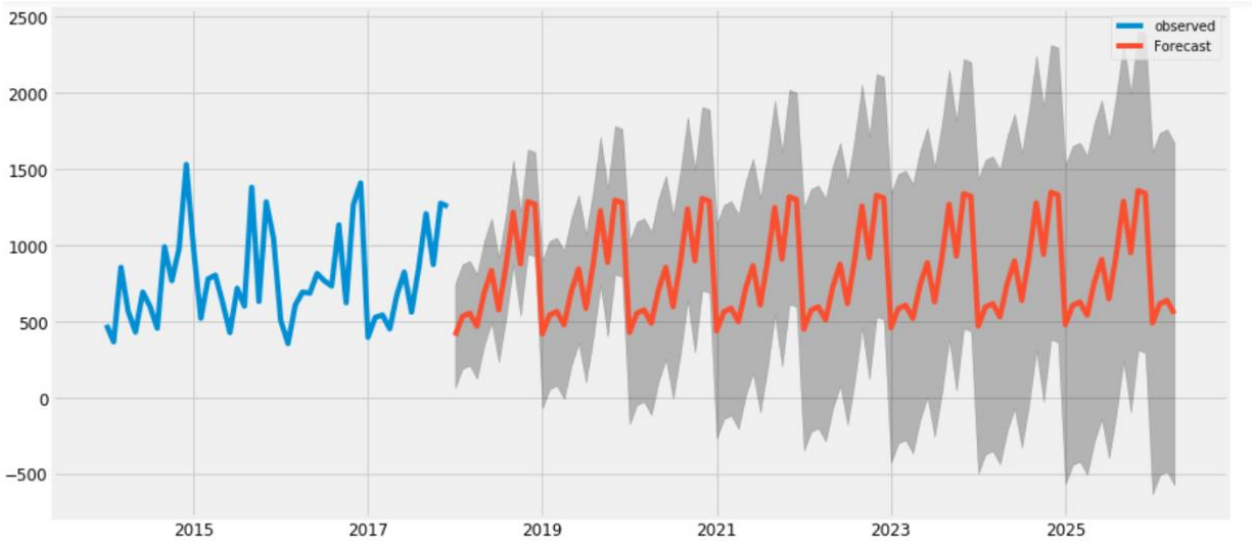

Рисунок 6.25 – Результат прогноза продаж до 2025 года

Как видно из графика, модель достаточно четко видит сезонность продаж скоропортящейся продукции. По мере удаления спрогнозированных значений от имеющихся данных, уверенность в правильности прогноза уменьшается. Это отражается в доверительных интервалах, генерируемых моделью, которые увеличиваются по мере того, как прогноз продвигается дальше в будущее.

7 Сравнение прогнозов двух категорий товаров и выявление тренда и сезонности продаж

Из анализа данных, представленных на рисунке 6.9 объем продаж овощей за эти годы, был гораздо больше, чем скоропортящейся продукции. Для выявления тренда сезонности сравним продажи этих двух категорий за один и тот же период времени. Это означает, что нужно объединить два фрейма данных в один и построить временные ряды этих двух категорий на одном графике.

На рисунке 7.1 представлен псевдокод предварительной обработки данных.

```
cols = ['Order ID', 'Ship Date', 'Product ID', 'Category', 'Sub-
Category', 'Product Name', 'Quantity', 'Discount', 'Profit']
perishable.drop(cols, axis=1, inplace=True)
vegetables.drop(cols, axis=1, inplace=True)
perishable = perishable.sort_values('Order Date')
vegetables = vegetables.sort_values('Order Date')
perishable = perishable.groupby('Order
Date')['Sales'].sum().reset_index()
vegetables = vegetables.groupby('Order
Date')['Sales'].sum().reset_index()
perishable = perishable.set_index('Order Date')
vegetables = vegetables.set_index('Order Date')
y_ perishable = perishable ['Sales'].resample('MS').mean()
y_ vegetables = office['Sales'].resample('MS').mean()
perishable = pd.DataFrame({'Order Date':y_ perishable.index,
'Sales':y_ perishable.values})
vegetables = pd.DataFrame({'Order Date': y_ vegetables.index,
'Sales': y_office.values})
store = perishable.merge(office, how='inner', on='Order Date')
store.rename(columns={'Sales_x': ' perishable _sales',
'Sales_y': ' vegetables _sales'}, inplace=True)
store.head()
```

Рисунок 7.1 – Предварительная обработка данных

На рисунке 7.2 представлено определение общего числа продаж двух категорий продукции за 2014 год.

0	2014-01-01	480.194231	285.357647
1	2014-02-01	367.931600	63.042588
2	2014-03-01	857.291529	391.176318
3	2014-04-01	567.488357	464.794750
4	2014-05-01	432.049188	324.346545

Рисунок 7.2 – Определение общего числа продаж двух категорий продукции за 2014 год

На рисунке 7.3 представлен результат сравнения продаж двух категорий продукции с 2014 по 2017 года.

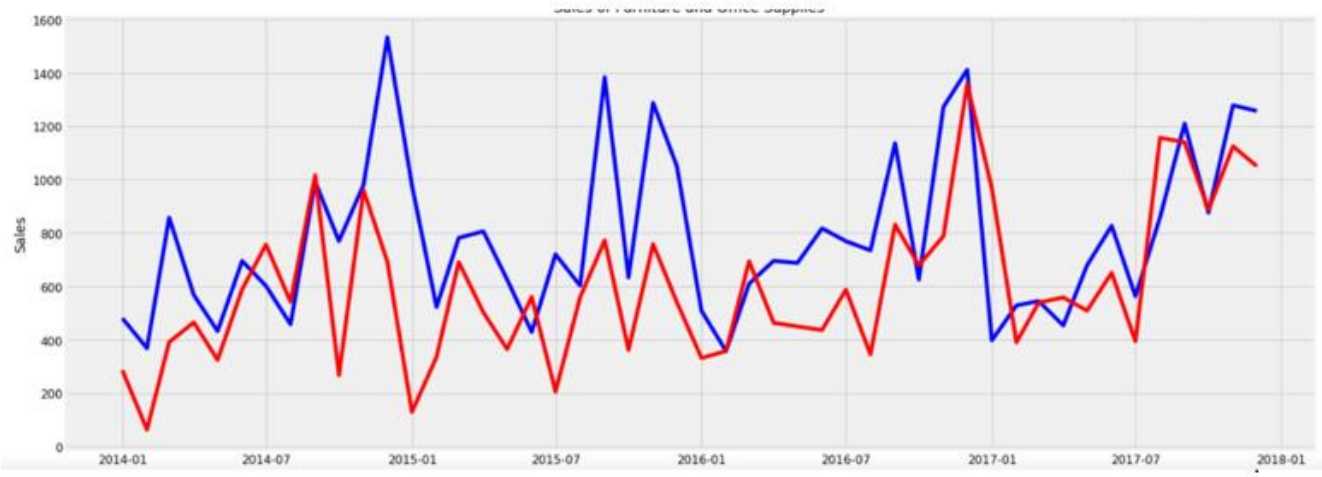

Рисунок 7.3 – Результат сравнения продаж двух категорий продукции с 2014 по 2017 года

Из графика видно, что продажи скоропортящейся продукции и овощей имели сходный сезонный характер. Начало года - межсезонье для обеих категорий. Кроме того, среднесуточные продажи скоропортящейся продукции выше, чем у овощей в большинстве месяцев. Это предсказуемо, так как

стоимость скоропортящейся продукции должна быть намного выше, чем у овощей. Из графика видно, что овощи опережали по продажам скоропортящуюся продукцию в средних ежедневных продажах.

Инструмент прогнозирования Prophet, выпущенный компанией Facebook в 2017 году, предназначен для анализа временных рядов, отображающих модели на разных временных масштабах, таких как годовой, недельный и дневной. Он также обладает расширенными возможностями для моделирования эффектов выходных дней во временных рядах и реализации пользовательских точек переключения. Поэтому используем Prophet для нахождения пользовательских точек переключения при исследовании тренда.

На рисунке 7.4 представлен псевдокод для нахождения пользовательских точек переключения.

```
from fbprophet import Prophet
perishable = perishable.rename(columns={'Order Date': 'ds',
'Sales': 'y'})
perishable _model = Prophet(interval_width=0.95)
perishable _model.fit(furniture)
office = office.rename(columns={'Order Date': 'ds', 'Sales':
'y'})
office_model = Prophet(interval_width=0.95)
office_model.fit(office)
perishable _forecast = perishable
_model.make_future_dataframe(periods=36, freq='MS')
perishable _forecast = perishable _model.predict(perishable
_forecast)
office_forecast = office_model.make_future_dataframe(periods=36,
freq='MS')
office_forecast = office_model.predict(office_forecast)
plt.figure(figsize=(18, 6))
perishable _model.plot(perishable _forecast, xlabel = 'Date',
ylabel = 'Sales')
plt.title(' perishable Sales');
```

Рисунок 7.4 – Псевдокод для нахождения пользовательских точек переключения

На рисунке 7.5 представлен результат нахождения пользовательских точек переключения для позиции скоропортящейся продукции.

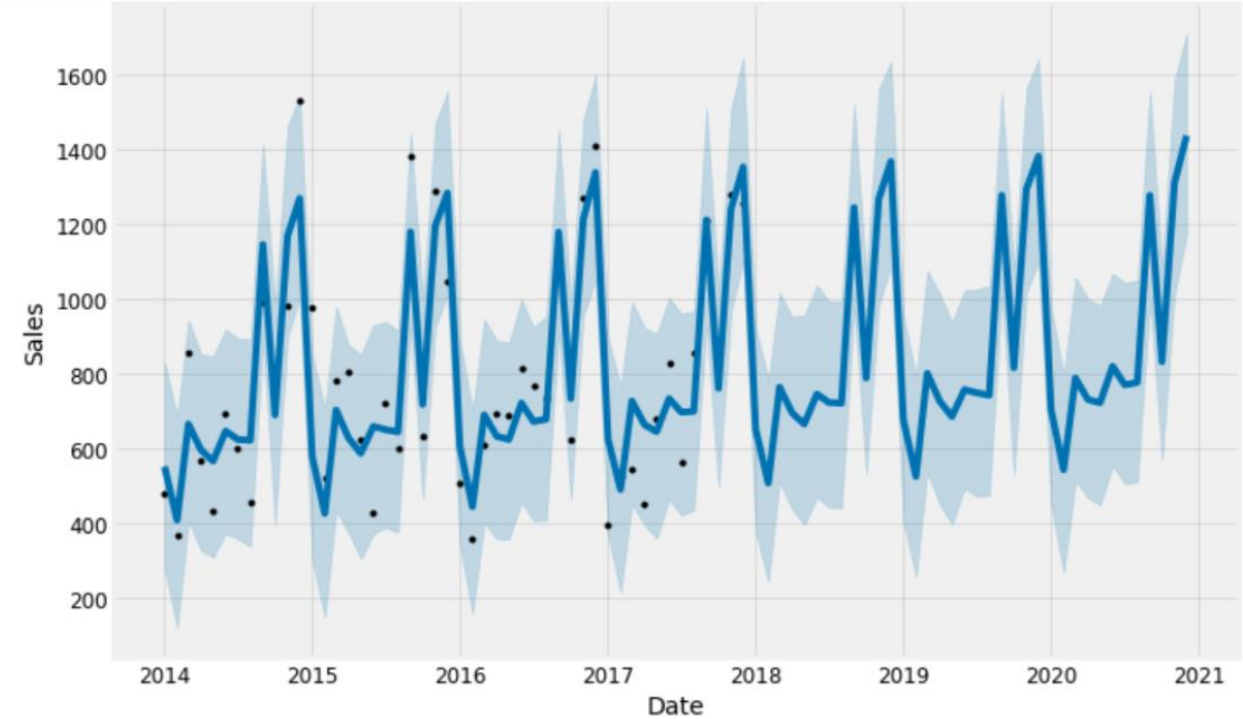

Рисунок 7.5 – Результат нахождения пользовательских точек переключения для позиции скоропортящейся продукции

На рисунке 7.6 представлен псевдокод для вывода графика продаж категории овощи.

```
plt.figure(figsize=(18, 6))
vegetables_model.plot(vegetables_forecast, xlabel = 'Date',
ylabel = 'Sales')
plt.title(Vegetables Sales');
```

Рисунок 7.6 – Псевдокод для вывода графика продаж категории овощи

На рисунке 7.7 представлен результат нахождения пользовательских точек переключения для позиции овощи.

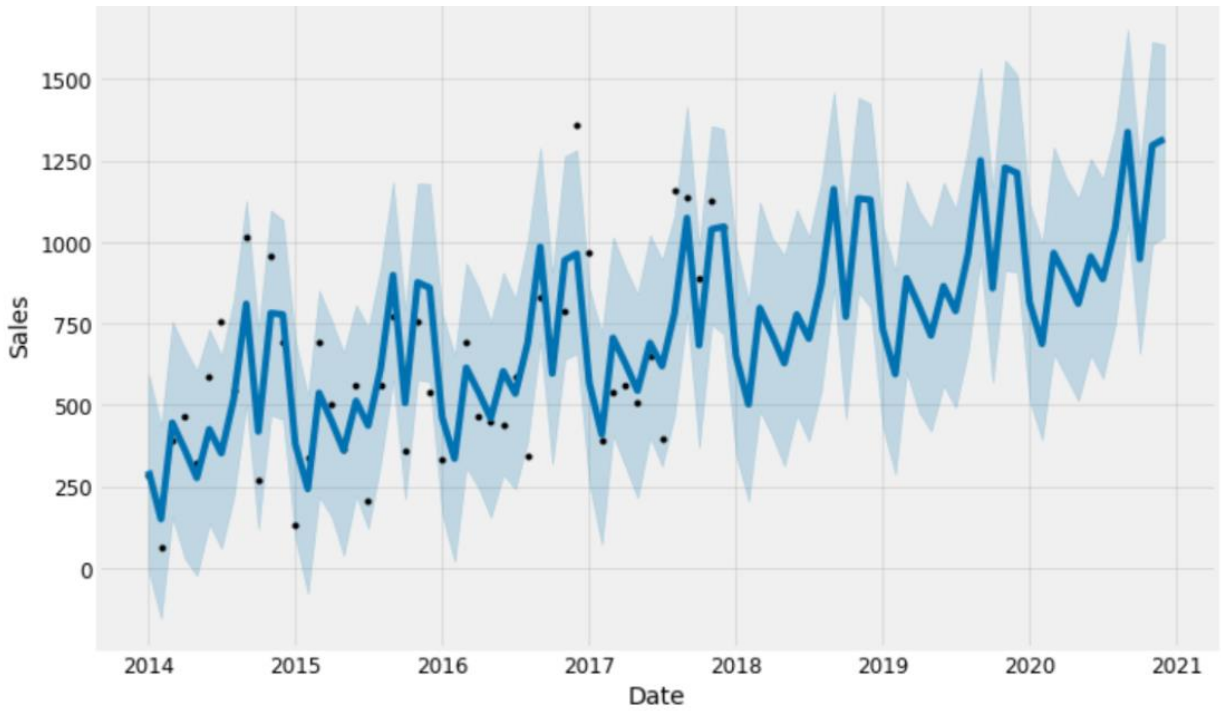

Рисунок 7.7 – Результат нахождения пользовательских точек переключения для позиции овощи

На рисунке 7.8 представлен псевдокод для сравнения тренда продаж двух категорий продукции.

```
perishable_names = [' perishable_%s' % column for column in
perishable _forecast.columns]
vegetables_names = [' vegetables _%s' % column for column in
vegetables _forecast.columns]
merge_ perishable _forecast = perishable _forecast.copy()
merge_ vegetables _forecast = vegetables _forecast.copy()
merge_ perishable _forecast.columns = perishable _names
merge_ vegetables _forecast.columns = vegetables _names
forecast = pd.merge(merge_ perishable _forecast, merge_
vegetables _forecast, how = 'inner', left_on = ' perishable
_ds', right_on = 'office_ds')
forecast = forecast.rename(columns={' perishable _ds':
'Date'}).drop(' vegetables _ds', axis=1)
forecast.head()
```

Рисунок 7.8 – Псевдокод для сравнения тренда продаж двух категорий продукции

На рисунке 7.9 представлен псевдокод для определения линий тренда двух категорий продукции.

```
plt.figure(figsize=(10, 7))
plt.plot(forecast['Date'], forecast['perishable_trend'], 'b-')
plt.plot(forecast['Date'], forecast['vegetables_trend'], 'r-')
plt.legend(); plt.xlabel('Date'); plt.ylabel('Sales')
plt.title('Perishable vs. Vegetables Sales Trend');
```

Рисунок 7.9 – Псевдокод для определения линий тренда двух категорий продукции

На рисунке 7.10 представлено сравнение линий тренда продаж двух категорий продукции.

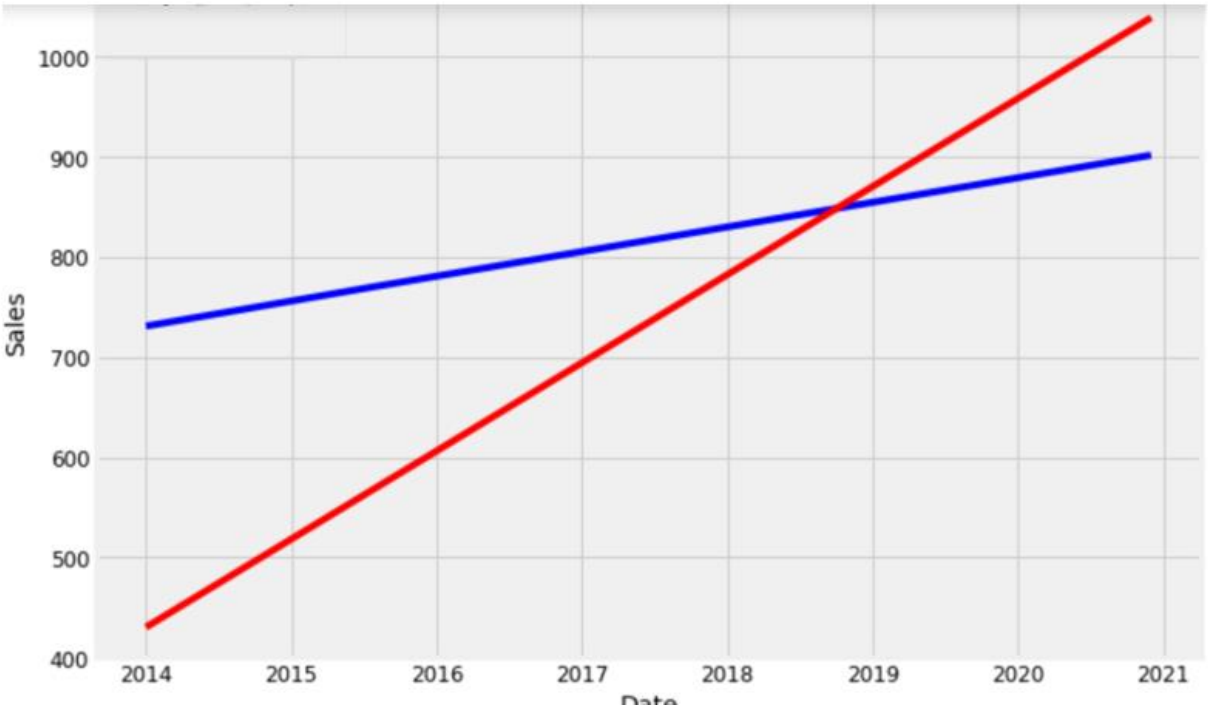

Рисунок 7.10 – Сравнение линий тренда продаж двух категорий продукции

На рисунке 7.11 представлен псевдокод для сравнения сезонности двух категорий продукции.

```
plt.figure(figsize=(10, 7))
plt.plot(forecast['Date'], forecast['perishable_yhat'], 'b-')
plt.plot(forecast['Date'], forecast['vegetables_yhat'], 'r-')
plt.legend(); plt.xlabel('Date'); plt.ylabel('Sales')
plt.title('Perishable vs. Vegetables Estimate');
```

Рисунок 7.11 – Псевдокод для сравнения сезонности двух категорий продукции

На рисунке 7.12 представлен результат сравнения сезонности двух категорий продукции.

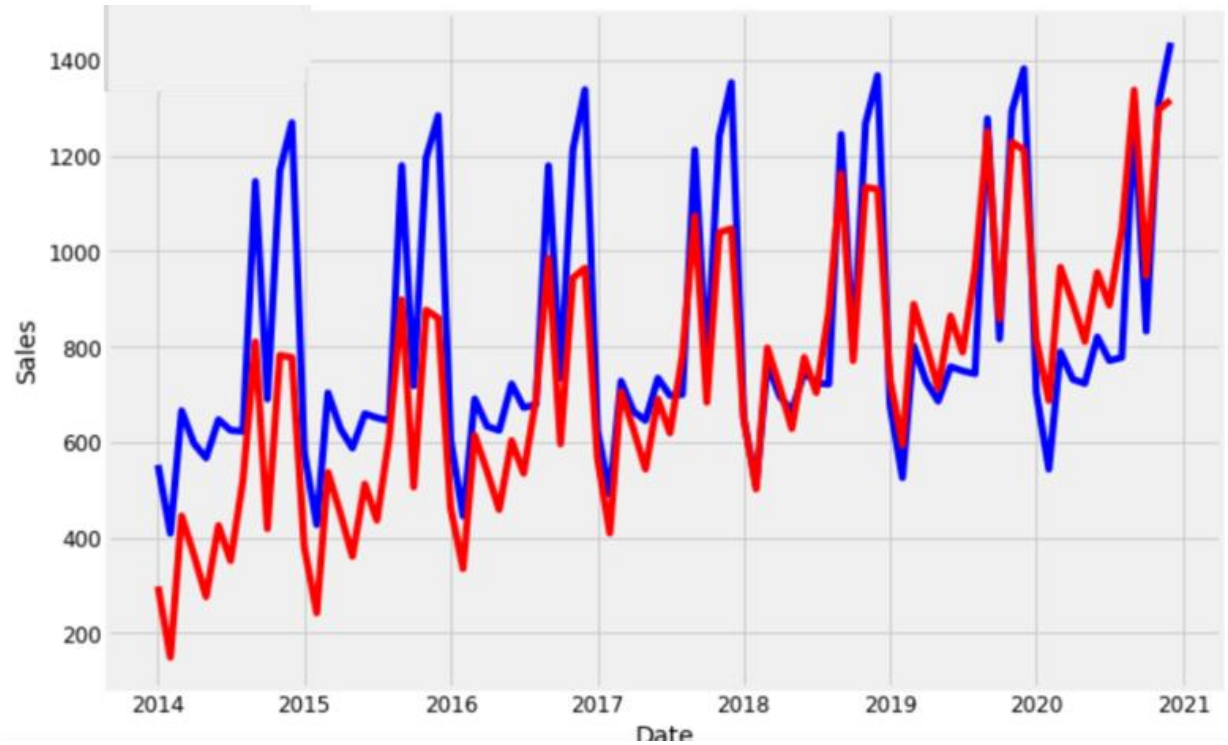

Рисунок 7.12 – Результат сравнения сезонности двух категорий продукции

Далее проведем сравнение для каждой из категорий продукции тренд и сезонность продаж.

На рисунке 7.13 представлено сравнение тренда и сезонности продаж для скоропортящейся продукции.

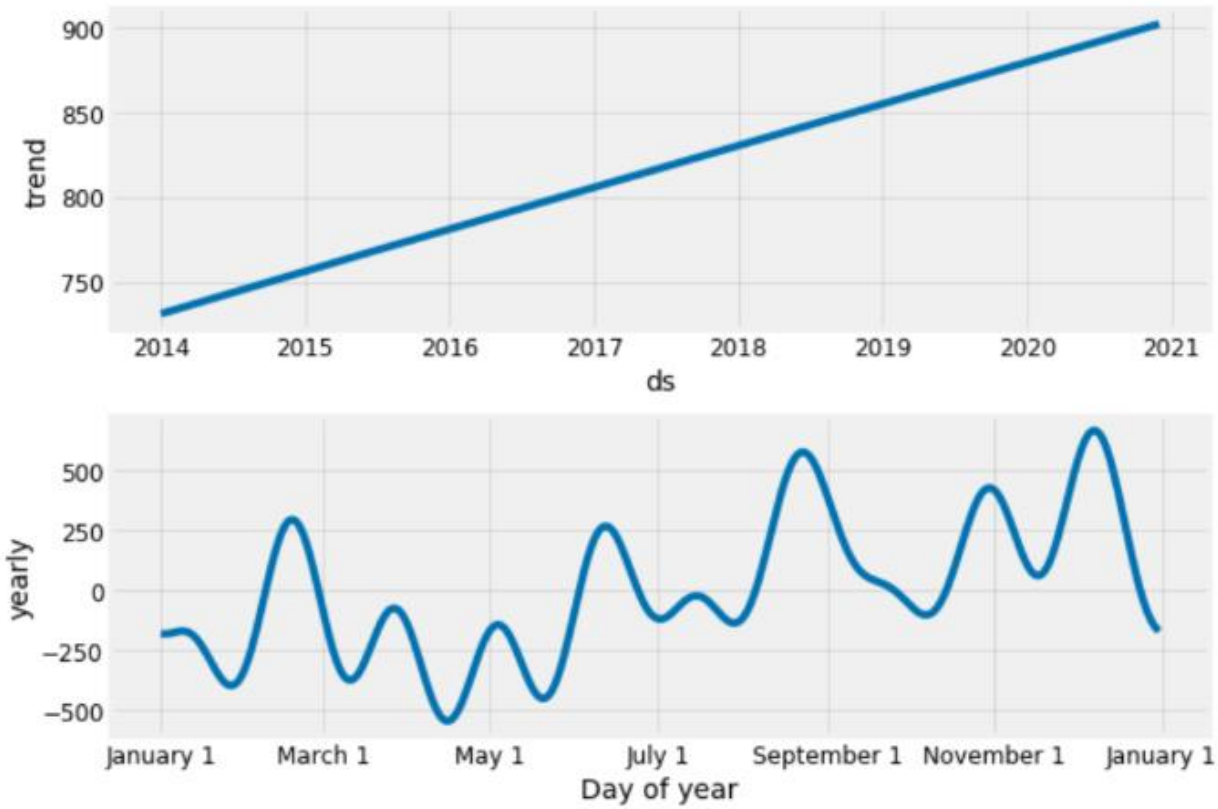

Рисунок 7.13 – Сравнение тренда и сезонности продаж для скоропортящейся продукции

На рисунке 7.14 представлено сравнение тренда и сезонности продаж для категории продукции овощи

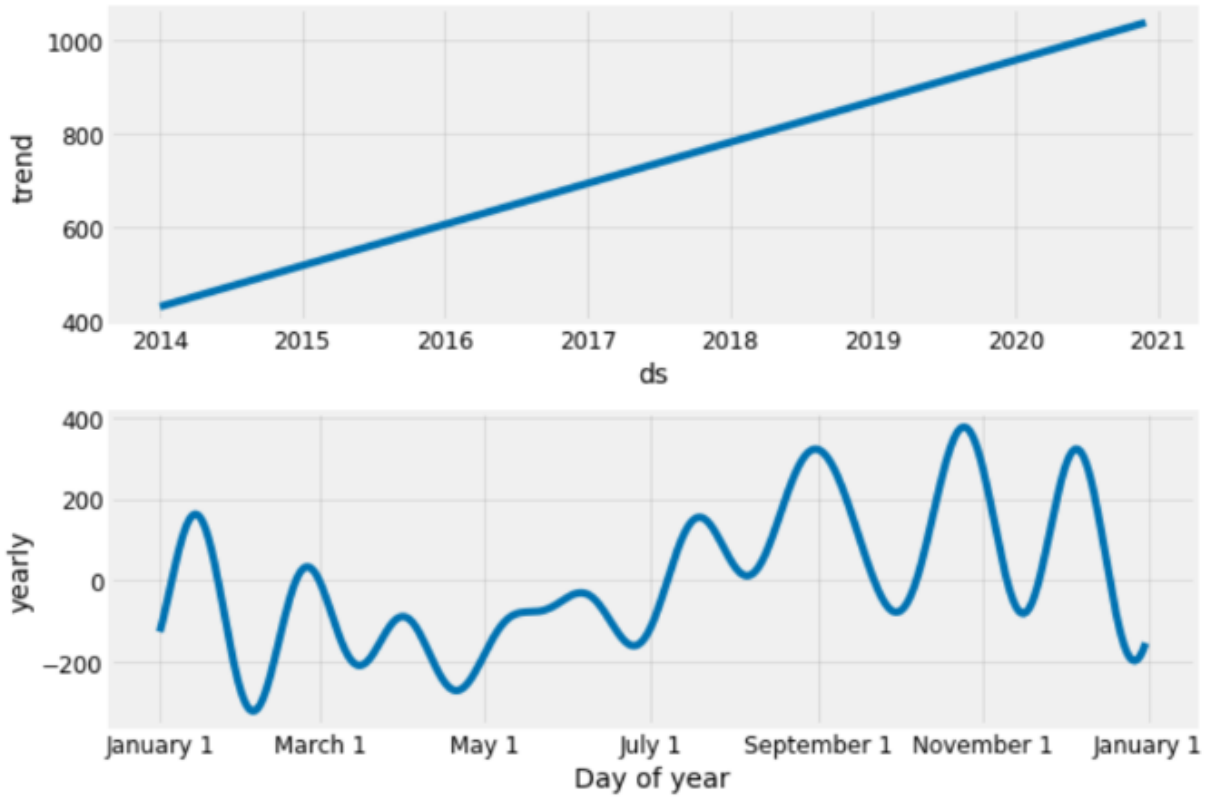

Рисунок 7.14 – Сравнение тренда и сезонности продаж для категории товаров овощи

Из графиков видно, что продажи как скоропортящейся продукции, так и овощей со временем линейно увеличивались и будут продолжать расти, хотя рост продаж овощей, кажется, несколько выше.

Худший месяц для скоропортящейся продукции оказался апрель, худший месяц для овощей оказался февраль. Лучший месяц для скоропортящейся продукции оказался декабрь, а лучший месяц для овощей оказался октябрь.

ЗАКЛЮЧЕНИЕ

В магистерской диссертации освещены основные принципы и особенности работы с технологиями машинного обучения. Рассмотрены основные принципы построения алгоритмов предсказания продаж с использованием больших данных и технологий машинного обучения.

Были изучены способы, методы и задачи технологий машинного обучения. Были изучены способы анализа больших данных. Изучены требования, технические условия для создания алгоритмов предсказания продаж товаров определенного типа.

Было проведено обучение алгоритма на основе данных о продажах с 2014 по 2017 года. Разработан алгоритм предсказания продаж на 2018 год на основе существующих реальных данных о продажах. Проведено сравнение прогноза с реальными продажами. Точность прогнозирования составила 70%. Также проведен прогноз скоропортящейся продукции на основе созданного алгоритма до 2025 года. Проведены прогнозы для двух категорий товаров для выявления тренда продаж и сезонности продаж. Был проведен сравнительный анализ полученных результатов и выявлена зависимость продаж товаров определённого типа от сезона.

Для разработки алгоритма предсказания продаж в работе использовалась программа Jupiter. Был произведен сравнительный анализ языков программирования. Данная программа использует язык программирования Python. Данный язык программирования содержит в себе необходимое и достаточное количество библиотек для реализации поставленной задачи. В работе используются стандартный набор библиотек, таких как Numpy, Pandas, Matpolit и др.

Реализованный в работе алгоритм предсказания может применяться для решения практических задач предсказания с достаточной точностью. Также разработанный алгоритм позволяет определить тренд продаж и выявить сезонность продаж на основе данных о предыдущих продажах.

Результаты работы и основные этапы работы опубликованы в [17, 18].

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Майер-Шенбергер В., Кукьер К. Большие данные. Революция, которая изменит то, как мы живём, работаем и мыслим. М.: Манн, Иванов и Фербер, 2014. 240с

2. Шаль С. А. Развитие рынка технологий «Big Data»: современные тенденции и сферы применения. / I Международная научная конференция «Конвергентные когнитивно-информационные технологии» и XI Международная научно-практическая конференция «Современные информационные технологии и ИТ-образование», Москва, 25-26 ноября 2016 г. / Московский государственный университет имени М.В. Ломоносова, Москва, 2016, С 148 – 151

3. Анализ данных и процессов. / А. А. Барсегян, М. С. Куприянов и др. СПб.: БХВ-Петербург, 2009. 512 с

4. Валевич Р.П., Давыдова Г.А. Экономика торговой организации. Учебное пособие. – Минск: Вышэйш. шк., 2008. – 371 с

5. Kuhn M., Johnson K. Applied Predictive Modeling. Springer, 2013. 600 p. — 203 illus., 153 illus. in color

6. Три подхода к прогнозированию продаж. [Электронный ресурс]. – URL: http://www.tidydata.ru/forecasting

7. Теория и практика прогнозирования временных рядов. [Электронный ресурс]. – URL: https://www.hse.ru/data/2015/08/25/1086537606/79.pdf

8. ARIMA. [Электронный ресурс]. – URL: http://www.machinelearning.ru/wiki/index.php?title=Autoregressive_Integrated_Moving_Average

9. Python 3 для начинающих. [Электронный ресурс]. – URL: https://pythonworld.ru/numpy/1.html

10. NumPy, часть 2: базовые операции над массивами. [Электронный ресурс]. – URL: https://pythonworld.ru/numpy/2.html

11. NumPy в Python. Часть 1. [Электронный ресурс]. – URL: https://habr.com/ru/post/352678/

12. Python Numpy Tutorial. [Электронный ресурс]. – URL: http://cs231n.github.io/python-numpy-tutorial/

13. Матрицы и массивы NumPy в Python. [Электронный ресурс]. – URL: https://www.internet-technologies.ru/articles/matricy-i-massivy-numpy-v-python.html

14. Первое знакомство NumPy, SciPy и Matplotlib. [Электронный ресурс]. – URL: https://ru.coursera.org/lecture/mathematics-and-python/piervoie-znakomstvo-numpy-scipy-i-matplotlib-gHnLw

15. Библиотека Numpy. [Электронный ресурс]. – URL: https://geekbrains.ru/posts/python_numpy

16. Quickstart tutorial. [Электронный ресурс]. – URL: https://docs.scipy.org/doc/numpy/user/quickstart.html

17. Гайдук К.А. Алгоритм предсказания продаж с использованием больших данных // «Инноватика – 2019»: Сборник материалов XV Международной школы-конференции студентов, аспирантов и молодых ученых 25-27 апреля 2019 (принято в печать)

18. Гайдук К.А. Алгоритм предсказания продаж с использованием технологий машинного обучения // «Инноватика – 2019»: Сборник материалов XV Международной школы-конференции студентов, аспирантов и молодых ученых 25-27 апреля 2019 (принято в печать)

19. МЕТОДИКА ПОСТРОЕНИЯ МОДЕЛИ ARIMA ДЛЯ ПРОГНОЗИРОВАНИЯ ДИНАМИКИ ВРЕМЕННЫХ РЯДОВ. [Электронный ресурс]. – URL: https://cyberleninka.ru/article/n/metodika-postroeniya-modeli-arima-dlya-prognozirovaniya-dinamiki-vremennyh-ryadov

20. ARIMA. [Электронный ресурс]. – URL: http://help.prognoz.com/ru/mergedProjects/Lib/02_time_series_analysis/uimodelling_arima.htm

АНТИПЛАГИАТ
ТВОРИТЕ СОБСТВЕННЫМ УМОМ

Томский Государственный Университет

СПРАВКА

о результатах проверки текстового документа на наличие заимствований

Проверка выполнена в системе Антиплагиат.ВУЗ

Автор работы	**Гайдук Кристина Алексеевна**
Подразделение	ФИТ, кафедра информационного обеспечения инновационной деятельности
Тип работы	Магистерская диссертация
Название работы	АЛГОРИТМ ПРЕДСКАЗАНИЯ ПРОДАЖ С ИСПОЛЬЗОВАНИЕМ БОЛЬШИХ ДАННЫХ И ТЕХНОЛОГИЙ МАШИННОГО ОБУЧЕНИЯ
Название файла	ВКР Гайдук.pdf
Процент заимствования	**20,09%**
Процент цитирования	**4,15%**
Процент оригинальности	**75,76%**
Дата проверки	**05:40:38 20 июня 2019г.**
Модули поиска	Кольцо вузов; Модуль поиска "ТГУ"; Модуль поиска общеупотребительных выражений; Коллекция Патенты; Модуль поиска перефразирований Интернет; Модуль поиска перефразирований eLIBRARY.RU; Коллекция Медицина; Модуль поиска Интернет; Коллекция ГАРАНТ; Коллекция eLIBRARY.RU; Модуль поиска переводных заимствований; Цитирование; Коллекция РГБ; Сводная коллекция ЭБС; Модуль выделения библиографических записей; Модуль поиска ИПС "Адилет"

Работу проверил — Гольцова Полина Андреевна

ФИО проверяющего

Дата подписи — 20.06.19.

Подпись проверяющего

Чтобы убедиться в подлинности справки, используйте QR-код, который содержит ссылку на отчет.

Ответ на вопрос, является ли обнаруженное заимствование корректным, система оставляет на усмотрение проверяющего. Предоставленная информация не подлежит использованию в коммерческих целях.