

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра многопроцессорных систем и сетей

МИРЕЙКО НАТАЛЬЯ ВИКТОРОВНА

**РЕАЛИЗАЦИЯ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ ПРОДАЖ
ДЛЯ КРУПНОГО БИЗНЕСА**

Отчет по преддипломной практике
Студентки 5 курса 1 группы

“Допустить к защите”

С предварительной оценкой _____

Руководитель практики

“ _____ ” _____ 2017 г

Руководитель практики

Гусейнова Анастасия Сергеевна

ассистент кафедры МСС ФПМИ

Минск 2017

РЕФЕРАТ

Отчет по преддипломной практике, 34 с., 15 рис., 6 формул, 8 источников.

ТЕОРИЯ ПРОГНОЗИРОВАНИЯ ПРОДАЖ, МЕТОД ХОЛЬТА-ВИНТЕРСА, ТРЕНД, СЕЗОННОСТЬ, ВРЕМЕННЫЕ РЯДЫ

Объект исследования – системы прогнозирования продаж для крупного бизнеса. Цель работы – разработка системы на основе математических моделей теории прогнозирования для анализа и эффективного прогнозирования продаж для крупного бизнеса.

Методы исследования – методы теории прогнозирования, методы анализа временных рядов, методы теории вероятности, имитационное моделирование.

Результатами являются модели и методы теории прогнозирования, методы анализа временных рядов, модели и методы анализа устойчивых случайных процессов, методы вычисления и вывода признаков для построения математической модели продаж.

Областью применения являются системы автоматизированного анализа и прогнозирования продаж для крупного бизнеса.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
1 МАТЕМАТИЧЕСКИЕ МОДЕЛИ И ПРИНЦИПЫ ПРОГНОЗИРОВАНИЯ ПРОДАЖ	7
1.1 Временные ряды и модели прогнозирования	8
1.2 Метод скользящей средней	10
1.3 Трендовая модель	12
1.4 Метод экспоненциального сглаживания с трендом и сезонностью Хольта – Винтерса	13
2 АРХИТЕКТУРА СИСТЕМЫ И ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ	17
2.1 Архитектура системы	17
2.2 Язык программирования C#	18
2.3 Платформа .NET 4.5	19
2.4 ASP .NET MVC 5	20
2.5 Entity Framework	22
2.6 База данных Oracle	23
2.7 AngularJS	24
3 РАБОТА ПОЛЬЗОВАТЕЛЯ С СИСТЕМОЙ	25
3.1 Описание функционала системы	25
3.2 Интерфейс системы	26
ЗАКЛЮЧЕНИЕ	35
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	36

ВВЕДЕНИЕ

В современных условиях жесткой конкуренции ключевой составляющей успеха многих компаний является их способность адаптироваться к изменяющимся условиям в их бизнес-среде. Таким образом критически важным для компании становятся хорошие прогнозы ключевых показателей бизнеса.

Эффективная система прогнозирования способна оптимизировать работу и процессы практически во всех областях деятельности предприятия: отдел продаж может лучше выстраивать ценовую политику для каждой товарной позиции и оптимизировать сбыт; производственные подразделения могут планировать объемы производства; отдел маркетинга – оценивать и планировать продажи в зависимости от рыночной стратегии; отдел логистики – лучше планировать цепочку поставок: от доставки до складских помещений. Вопрос планирования логистики особенно остро стоит для компаний из пищевой отрасли – конечный срок годности продуктов не позволяет запасаться большими объемами впрок. С другой стороны существует риск упущенной прибыли из-за отсутствия необходимых продуктов на складе.

Точный прогноз способен сократить количество товаров на складах, повысить оборачиваемость товара, снизить количество просроченных и списанных товаров без сокращения прибыли, а также повлиять на увеличение дохода из-за своевременного наличия нужного покупателю товара.

Без использования специальных инструментов сложно определить и учесть все факторы, влияющие на спрос и тем более спрогнозировать продажи на долгосрочный период. Достаточно точные прогнозы с известной точностью дает применение математических методов прогнозирования.

Все вышесказанное подводит нас к актуальности поставленной в данной работе проблемы: необходим удобный и доступный инструмент для прогнозирования продаж, встраиваемый в привычные для компании бизнес-процессы и легко интегрируемый в них.

Разработанная система рассчитывает прогноз продаж с учетом различных факторов, например, сезонности, праздников, недельных и суточных колебаний спроса, маркетинговых акции и т.д. и с помощью построения трендов. Для этого разработанная система использует алгоритм, основанный на самом популярном алгоритме прогнозирования продаж – модели Хольта-Уинтерса.

С технической точки зрения разработанная система – это отдельное SPA приложение с собственной базой данных Oracle, за счет чего обеспечивается обработка больших массивов данных за несколько лет.

Входными данными для разработанной системы является база данных с региональным классификатором магазинов, товарным классификатором, рядами продаж по SKU (stock keeping unit), список праздничных дней, разметка промо-акций для каждого товара и розничные цены на товары.

Для критерия качества используется скользящий контроль — прогноз продаж товаров, сделанный исходя из данных на некотором начальном временном интервале, сравнивается с реальными продажами. Критерием качества служит сумма модулей отклонений прогноза от реальной величины закупок либо сумма квадратов отклонений.

Будем предполагать, что вероятности продажи товаров из одной группы нижнего уровня (т.е. группы, в которую входят только товары, а не другие группы) во всех магазинах имеют одинаковое распределение. Таким образом, оценив это распределение, а также суммарные продажи всех товаров из группы нижнего уровня в некотором магазине), можно будет спрогнозировать продажи отдельных товаров точнее, чем используя базовый алгоритм по каждому товару. Также предполагается, что прогноз можно делать по отдельности для каждого из магазинов.

1 МАТЕМАТИЧЕСКИЕ МОДЕЛИ И ПРИНЦИПЫ ПРОГНОЗИРОВАНИЯ ПРОДАЖ

Прогнозирование — это разработка прогноза; в узком значении — специальное научное исследование конкретных перспектив развития какого-либо процесса.

Необходимость прогноза обусловлена желанием знать события будущего, что невозможно на 100 процентов в принципе, исходя из статистических, вероятностных, эмпирических, философских принципов. Точность любого прогноза обусловлена:

- 1) объёмом истинных (верифицированных) исходных данных и периодом их сбора;
- 2) объёмом неверифицированных исходных данных, периодом их сбора;
- 3) свойствами системы, объекта, подвергающихся прогнозированию;
- 4) методиками и подходами прогнозирования.

При возрастании совокупности факторов, влияющих на точность прогноза он практически замещается рутинным расчётом с некоторой установившейся погрешностью.

Прогнозы делятся (условно):

- 1) по срокам: краткосрочные, среднесрочные, долгосрочные, дальнесрочные;
- 2) по масштабу: частные, местные, региональные, отраслевые, страновые, мировые (глобальные);
- 3) по ответственности (авторству): личные, на уровне предприятия (организации), на уровне государственных органов.

Для достоверного прогноза требуется, как минимум, в два раза больше исходных исторических данных. Например, для прогнозирования одного года требуется минимум два года фактических данных по предыдущим периодам. Рекомендуются использовать три периода фактических данных к одному прогнозируемому периоду.

Статистические наблюдения в социально-экономических исследованиях обычно проводятся регулярно через равные отрезки времени и представляются в виде временных рядов $x(t)$, где $t = 1, 2, \dots, n$. В качестве инструмента статистического прогнозирования временных рядов служат трендовые регрессионные модели, параметры которых оцениваются по имеющейся статистической базе, а затем основные тенденции (тренды) экстраполируются на заданный интервал времени.

Методология статистического прогнозирования предполагает построение и испытание многих моделей для каждого временного ряда, их сравнение на основе статистических критериев и отбор наилучших из них для прогнозирования.

При моделировании сезонных явлений в статистических исследованиях различают два типа колебаний: мультипликативные и аддитивные. В мультипликативном случае размах сезонных колебаний изменяется во времени пропорционально уровню тренда и отражается в статистической модели множителем. При аддитивной сезонности предполагается, что амплитуда сезонных отклонений постоянна и не зависит от уровня тренда, а сами колебания представлены в модели слагаемым.

Основой большинства методов прогнозирования является экстраполяция, связанная с распространением закономерностей, связей и соотношений, действующих в изучаемом периоде, за его пределы, или — в более широком смысле слова — это получение представлений о будущем на основе информации, относящейся к прошлому и настоящему.

Наиболее известны и широко применяются трендовые и адаптивные методы прогнозирования. Среди последних можно выделить такие, как методы авторегрессии, скользящего среднего (Бокса — Дженкинса и адаптивной фильтрации), методы экспоненциального сглаживания (Хольта, Брауна и экспоненциальной средней) и др.

Основные способы оценки прогнозной модели, которые можно использовать:

- 1) Оценить отношение фактических продаж к прогнозу;
- 2) Расчет показателя точность прогноза — оценка на сколько точно выбранная модель описывает анализируемые данные;
- 3) Графический анализ — строим график и визуально оцениваем адекватность модели прогноза относительно фактических продаж за последний период. [2][3]

1.1 Временные ряды и модели прогнозирования

Для повышения точности прогнозов продаж важно понимать, что разные модели прогнозирования подходят для разных временных рядов с разными характеристиками.

К сожалению, нет одной модели, которая бы подошла для расчета прогноза для всех ситуаций - типов временных рядов.

Какие характеристики временных рядов можно выделить:

1) Сезонность:

- а) Полные ряды – когда данные есть минимум за 1 полный цикл, т.е. минимум за 12 месяцев, или за 4 квартала и можно выделить сезонность;
- б) Неполный ряд – когда данных еще нет за полный цикл.

2) Рост:

- а) Временной ряд с ростом, когда можно выделить явный рост продаж от года к году:
 - i) Рост может быть постоянный;
 - ii) Рост может быть затухающий;
 - iii) Рост может быть экспоненциальный;
 - iv) Рост может меняться падением.
- б) Статичный временной ряд – это временной ряд, который не содержит роста.

3) Важность тенденций для прогноза:

- а) Для прогноза могут быть важны тенденции за длительный период;
- б) Для прогноза могут быть важны тенденции за последний период;
- с) Не регулярные, разовые продажи.

У нас есть набор различных параметров временных рядов. Данные параметры для одного временного ряда будут сочетаться между собой. Например,

- 1) Временной ряд может быть полный с ростом и тенденциями за длительный период;
- 2) Временной ряд может быть полный статичный с тенденциями за последний период;
- 3) неполный с ростом и тенденциями за последний период и т.д.

Рассмотрим 4 типа моделей прогнозирования временных рядов:

- 1) Модели экспоненциального сглаживания Хольта-Винтерса с трендом и сезонностью;
- 2) Модели скользящего среднего с сезонностью;
- 3) Трендовые модели с сезонностью – линейный тренд с сезонность, логарифм с сезонностью.

На рисунке 1 представлено, для каких характеристик временного ряда, какие модели прогнозирования подходят:

Характеристики ряда	Трендовая модель	Скользящая средняя	Модели экспоненциального сглаживания
1. Характеристика ряда для выделения сезонных колебаний:			
Полные ряды – когда данные есть минимум за 1 полный цикл, т.е. минимум за 12 месяцев, или за 4 квартала и можно выделить сезонность;	Тренд + сезонность	Скользящая + сезонность	Хольта-Винтерса
Неполный ряд – когда данных еще нет за полный цикл;	Тренд	Средняя	Простая или Хольта
2. Характеристика ряда для выделения роста:			
Временной ряд с ростом, когда можно выделить явный рост продаж от года к году.	Линейный, экспоненциальный или логарифмический тренд (+ сезонность для полных рядов)	—	Хольта или Хольта-Винтерса
Рост может быть постоянный;	линейный тренд		
Рост может быть затухающий;	логарифмический		
Рост может быть экспоненциальный;	экспоненциальный		
Рост может меняться падением;	полином или линейный тренд за последний период		
Статичный временной ряд – это временной ряд, который не содержит роста;	Полиномы	Скользящая + сезонность	
3. Характеристика важности тенденций для прогноза:			
Для прогноза могут быть важны тенденции за длительный период;	Тренды за весь период, линейный, экспоненциальный, логарифмический (+ сезонность для полных рядов)		Хольта или Хольта-Винтерса с коэффициентом сглаживания стремящимся к нулю, чем ближе к нулю, тем больше влияние тенденций за весь период
Для прогноза могут быть важны тенденции за последний период;	Используем тренды (линейный, логарифмический, экспоненциальный) за последние периоды, а сезонность рассчитываем за длительный период	Используем скользящую среднюю за столько периодов, сколько имеют максимальный вес для прогнозирования (ср. за 2, 3, 4 ... периодов)	Хольта или Хольта-Винтерса с коэффициентом сглаживания стремящимся к единице, чем ближе к единице, тем больше влияние тенденций за последний период
4. Не регулярные, разовые продажи;		Если нарастающим итогом, то используем скользящую	

Рисунок 1 - Временные ряды и модели прогнозирования

Для достижения точных прогнозов следует использовать модель, которая лучше всего подходит для характеристик конкретного временного ряда. [2][3]

1.2 Метод скользящей средней

Экстраполяция - это метод научного исследования, который основан на распространении прошлых и настоящих тенденций, закономерностей, связей на будущее развитие объекта прогнозирования. К методам экстраполяции относятся метод скользящей средней, метод экспоненциального сглаживания, метод наименьших квадратов.

Метод скользящих средних является одним из широко известных методов сглаживания временных рядов. Применяя этот метод, можно элиминировать случайные колебания и получить значения, соответствующие влиянию главных факторов.

Сглаживание с помощью скользящих средних основано на том, что в средних величинах взаимно погашаются случайные отклонения. Это происходит вследствие замены первоначальных уровней временного ряда средней арифметической величиной внутри выбранного интервала времени. Полученное значение относится к середине выбранного интервала времени (периода).

Затем период сдвигается на одно наблюдение, и расчет средней повторяется. При этом периоды определения средней берутся все время одинаковыми. Таким образом, в каждом рассматриваемом случае средняя центрирована, т.е. отнесена к срединной точке интервала сглаживания и представляет собой уровень для этой точки.

При сглаживании временного ряда скользящими средними в расчетах участвуют все уровни ряда. Чем шире интервал сглаживания, тем более плавным получается тренд. Сглаженный ряд короче первоначального на $(n-1)$ наблюдений, где n – величина интервала сглаживания.

При больших значениях n колеблемость сглаженного ряда значительно снижается. Одновременно заметно сокращается количество наблюдений, что создает трудности.

Выбор интервала сглаживания зависит от целей исследования. При этом следует руководствоваться тем, в какой период времени происходит действие, а следовательно, и устранение влияния случайных факторов.

Данный метод используется при краткосрочном прогнозировании. Его рабочая формула (1) приведена ниже:

$$y_{t+1} = m_{t-1} + \frac{1}{n} * (y_t - y_{t-1}) \quad (1)$$

где $t + 1$ – прогнозный период; t – период, предшествующий прогнозному периоду (год, месяц и т.д.); y_{t+1} – прогнозируемый показатель; m_{t-1} – скользящая средняя за два периода до прогнозного; n – число уровней, входящих в интервал

сглаживания; y_t – фактическое значение исследуемого явления за предшествующий период; y_{t-1} – фактическое значение исследуемого явления за два периода, предшествующих прогнозному.[4]

1.3 Трендовая модель

Прогноз на будущее можно построить с помощью трендов. Трендовые модели, в отличие от скользящей средней, позволяют строить прогнозы на отдаленные моменты времени. Данные по экономическим показателям в различные периоды времени являются динамическим рядом, т.е. совокупностью значений некоторого параметра y , определяемого в различные моменты времени. Любой ряд динамики может быть разделен на три компоненты, что показано в формуле (2):

$$y(t) = f(t) + g(t) + h(t) \quad (2)$$

где $f(t)$ - детерминированная компонента, представляющая собой некоторую аналитическую функцию, выражающую тенденцию в ряду динамики; $g(t)$ - стохастическая компонента, моделирующая характер периодической и квазипериодической вариации исследуемого явления; $h(t)$ - случайная компонента типа «белый шум».

Таким образом, вычитание тренда из исследуемого ряда динамики является изменением масштаба данных и сохраняет полную информацию о вариации явления.

Построить трендовую модель явления — значит найти детерминированную функцию $f(t)$, и характеристики случайных отклонений от нее, позволяющие определить доверительный интервал, в границах которого с заданной доверительной вероятностью должна находиться прогнозируемая величина.

При построении трендовой модели прежде всего выбирают форму кривой тренда, затем подбирают параметры этой кривой по какому-либо критерию оптимальности и, наконец, по совокупности критериев оценивают качество подобранной кривой.

В качестве тренда используют линейную функцию, параболу, многочлен i -й степени, гиперболу, экспоненту, логарифмическую функцию и др. Чаще всего модель описывается линейной функцией. При описании модели нелинейной функцией система уравнений для расчета параметров кривой может оказаться

достаточно сложной. Поэтому иногда для получения параметров нелинейной функции ее приводят к линейному виду.

Для длинных рядов выделение тренда носит обычно разведочный характер, так как часто невозможно указать подходящую параметрическую кривую для аппроксимации ряда на всей его длине. Для выделения тренда в этом случае используют различные непараметрические методы анализа временных рядов, такие как, сглаживание скользящими средними или скользящими медианами, частотную фильтрацию и т.п. [2] В отличие от параметрических методов выделения тренда, эти методы пригодны лишь для осреднения значений ряда по точкам некоторой окрестности и не могут быть использованы для прогнозирования (экстраполяции) динамических рядов, поскольку не дают в явном виде расчетного уравнения детерминированной компоненты $f(t)$. Однако получение достаточно гладкой траектории дает возможность визуально оценить наличие тенденции в условиях сильной зашумленности, а также выделить ряд остатков $y(t) = x(t) - f(t)$, как случайную компоненту временной последовательности, если конечной целью исследования является построение моделей авторегрессии для прогнозирования.

Сезонным трендом называют периодические изменения показателя, связанные, например, с сезонными изменениями спроса (например, на одежду, обувь). Смешанным сезонным трендом называют комбинацию из сезонного и любого другого рассмотренного тренда (например, линейного).

Тренды различают также по их типу. Аддитивным трендом называют временную зависимость, в которой значения параметра отклоняются в положительную и отрицательную стороны от тренда в среднем на одну и ту же величину. Мультипликативным трендом называют временную зависимость, в которой значения параметра отклоняются в положительную и отрицательную стороны от тренда в среднем на одинаковый процент. [5]

1.4 Метод экспоненциального сглаживания с трендом и сезонностью Хольта - Винтерса

Винтерс развил модель экспоненциального сглаживания с трендом Хольта и добавил в неё сезонность. Преимущество данного метода – это возможность сделать прогноз на длительный период. Но для того чтобы сделать прогноз, например, на 1 год, вам понадобятся данные минимум за 2 полных года, а лучше за 3 - 5 полных лет.

Метод Хольта - Винтерса используется для прогнозирования временных рядов, когда в структуре данных есть сложившийся тренд и сезонность.

Модель прогноза Хольта Винтерса — это 3-х параметрическая модель прогноза, которая учитывает:

- сглаженный экспоненциальный ряд;
- тренд;
- сезонность.

Этапы расчета прогноза по методу Хольта - Винтерса:

1) Рассчитываем экспоненциально-сглаженный ряд по формуле (3):

$$L_t = k * Y_t / S_{t-s} + (1 - k) * (L_{t-1} + T_{t-1}) \quad (3)$$

где L_t — сглаженная величина на текущий период; k — коэффициент сглаживания ряда; S_{t-s} — коэффициент сезонности за этот же период в предыдущем сезоне; Y_t — текущее значение ряда (например, объём продаж); L_{t-1} — сглаженная величина за предыдущий период; T_{t-1} — значение тренда за предыдущий период.

Коэффициент сглаживания ряда k задается вручную и находится в диапазоне от 0 до 1. Для первого периода в начале данных экспоненциально-сглаженный ряд равен первому значению ряда (например, объёму продаж за первый месяц) $L_1 = Y_1$. Сезонность в первом и втором периоде S_{t-s} равна 1.

2) Определяем значение тренда по формуле (4):

$$T_t = b * (L_t - L_{t-1}) + (1 - b) * T_{t-1} \quad (4)$$

где T_t — значение тренда на текущий период; b — коэффициент сглаживания тренда; L_t — экспоненциально сглаженная величина за текущий период; L_{t-1} — экспоненциально сглаженная величина за предыдущий период; T_{t-1} — значение тренда за предыдущий период.

Коэффициент сглаживания тренда b задается вручную и находится в диапазоне от 0 до 1. Значение тренда для первого периода равно 0 ($T_1 = 0$).

3) Оцениваем сезонность по формуле (5):

$$S_t = q * Y_t / L_t + (1 - q) * S_{t-s} \quad (5)$$

где S_t — коэффициент сезонности для текущего периода; q — коэффициент сглаживания сезонности; Y_t — текущее значение ряда (например, объём продаж); L_t — сглаженная величина за текущий период; S_{t-s} — коэффициент сезонности за этот же период в предыдущем сезоне.

Коэффициенты сезонности для первого сезона (года) = 1.

4) Делаем прогноз на p периодов вперед, используя формулу (6):

$$\hat{Y}_{t+p} = (L_t + p * T_t) * S_{t-s} + p \quad (6)$$

где \hat{Y}_{t+p} — прогноз по методу Хольта-Винтерса на p периодов вперед; L_t — экспоненциально сглаженная величина за последний период; p — порядковый номер периода, на который делаем прогноз; T_t — тренд за последний период; S_{t-s} — коэффициент сезонности за этот же период в последнем сезоне.

При появлении новых данных прогноз по методу Хольта - Винтерса желательно пересчитать для уточнения ряда, тренда и сезонности. Также при подготовке данных для прогноза всегда стоит очищать данные от факторов, которые в прогнозном периоде не повторятся (например, прирост продаж по крупной акции) или учитывать запланированные факторы, которые дадут дополнительный прирост продаж (например, ввод продукции в сеть или проведение мероприятия по стимулированию сбыта).

Этапы оценки точности модели Хольта - Винтерса и подобранных оптимальных коэффициенты сглаживания для ряда, тренда и сезонности:

- 1) Рассчитываем прогноз на 1 период вперед для каждого месяца, когда продажи нам известны. Прогноз для оценки модели в первом и втором году (сезоне) = значению экспоненциально-сглаженного ряда за предыдущий период + значение тренда за предыдущий период. (значение тренда мы не умножаем на p , т.к. прогноз делаем на 1 период, а в этом случае $p = 1$). Прогноз для третьего года (сезона) = (значение экспоненциально-сглаженного ряда за предыдущий период + значение тренда за предыдущий период) умножить на коэффициент сезонности этого периода в предыдущем сезоне.
- 2) Рассчитаем ошибку модели = из фактических данных вычитаем прогноз на этот период.
- 3) Определим отклонение ошибки модели от прогнозной модели = Отношение ошибки модели в квадрате к фактическому значению в квадрате.
- 4) Рассчитаем точность прогноза = единица минус среднее значение отклонений.

Для подбора коэффициентов сглаживания ряда, тренда и сезонности k , b и q , при которых прогноз будет максимально точным, нам необходимо последовательно перебрать все значения k , b и q в диапазоне от 0 до 1 и найти

такое сочетание, при котором точность прогноза будет максимальна приближена к 100 процентам. [6][7]

2 АРХИТЕКТУРА СИСТЕМЫ И ИСПОЛЬЗОВАННЫЕ ТЕХНОЛОГИИ

2.1 Архитектура системы

Так как хотелось, чтобы разработанной системой можно было пользоваться на максимальном количестве устройств, несмотря на операционную систему и тип (персональный компьютер, планшет, мобильный телефон), было выбрано реализовывать систему как Web-приложение. Для реализации web-приложения был выбран язык программирования C#. В качестве web-фреймворка был выбран .NET 4.5 – один из наиболее популярных web-фреймворков, который можно использовать с C#. Для работы с данными используется фреймворк Entity Framework, который позволяет абстрагироваться от конкретного типа базы данных в приложении. В качестве базы данных используется Oracle. В подпунктах главы 2 подробнее будет рассмотрена каждая из выбранных технологий и будет дано обоснование этого выбора.

Для реализации Web-приложения в качестве архитектуры было решено использовать приложение одной страницы (SPA). SPA – это Web-приложение, размещенное на одной странице, которая для обеспечения работы загружает все javascript-файлы (модули, виджеты, контролы и т.д.), а также файлы CSS вместе с загрузкой самой страницы. Основные плюсы использования SPA принцип архитектуры приложения:

- 1) Работа на большом количестве устройств, а значит, создав одно приложение, вы получаете гораздо большую аудиторию пользователей нежели при использовании стандартного подхода.
- 2) Богатый пользовательский интерфейс, так называемый User Experience. Так как web-страница одна, построить богатый, насыщенный пользовательский интерфейс гораздо проще. Проще хранить информацию о сеансе, управлять состояниями представлений (views) и управлять анимацией (в некоторых случаях).
- 3) SPA существенно (в разы) сокращает так называемые “хождения по кругу”, то есть загрузку одного и того же контента снова и снова. Если ваше приложение использует шаблон, то вместе с основным содержанием какой-либо страницы пользователь обязательно загружает разметку шаблона. Да, кэширование данных на данном этапе развития www достигло высочайших результатов, но если нечего кэшировать, то и время, и ресурсы на это не тратятся.

Для реализации SPA было выбрано использовать на клиентской стороне приложения MVW фреймворк AngularJS, а на серверной части приложения ASP .NET MVC 5.

Ниже приведен полный стек использованных технологий:

- ▶ платформа .NET 4.5;
- ▶ язык программирования C#;
- ▶ тип приложения ASP .NET MVC 5;
- ▶ Entity Framework;
- ▶ база данных Oracle;
- ▶ AngularJS;
- ▶ KendoUI;
- ▶ Twitter Bootstrap;
- ▶ Html, CSS, LESS;
- ▶ JS Plugins (Latex, ...);
- ▶ Grunt, Gulp.

2.3.1 Язык программирования C#

В качестве языка программирования был выбран C#. На сегодняшний момент язык программирования C# один из самых мощных, быстро развивающихся и востребованных языков в IT-отрасли. В настоящий момент на нем пишутся самые различные приложения: от небольших десктопных программ до крупных веб-порталов и веб-сервисов, обслуживающих ежедневно миллионы пользователей.

По сравнению с другими языками C# достаточно молодой, но в то же время он уже прошел большой путь. Первая версия языка вышла вместе с релизом Microsoft Visual Studio .NET в феврале 2002 года. Текущей версией языка является версия C# 5.0, которая вышла в августе 2012 года вместе с Visual Studio 2012.

C# является языком с Си-подобным синтаксисом и близок в этом отношении к C++ и Java. C# является объектно-ориентированным и в этом плане много перенял у Java и C++. Например, C# поддерживает полиморфизм, наследование, перегрузку операторов, статическую типизацию. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений. И C# продолжает активно развиваться, и с каждой новой версией появляется все

больше интересных функциональностей, как, например, лямбды, динамическое связывание, асинхронные методы и т.д.[8]

2.3.2 Платформа .NET 4.5

Фреймворк .NET представляет мощную платформу для создания приложений. Можно выделить следующие ее основные черты:

- 1) Поддержка нескольких языков. Основой платформы является общезыковая среда исполнения Common Language Runtime (CLR), благодаря чему .NET поддерживает несколько языков: наряду с C# это также VB.NET, C++, F#, а также различные диалекты других языков, привязанные к .NET, например, Delphi.NET. При компиляции код на любом из этих языков компилируется в сборку на общем языке CIL (Common Intermediate Language) - своего рода ассемблер платформы .NET. Поэтому мы можем сделать отдельные модули одного приложения на отдельных языках.
- 2) Кроссплатформенность. .NET является переносимой платформой (с некоторыми ограничениями). Например, последняя версия платформы на данный момент .NET Framework 4.5.1. поддерживается на большинстве современных ОС Windows (Windows 8.1/8/7/Vista). А благодаря проекту Mono можно создавать приложения, которые будут работать и на других ОС семейства Linux, в том числе на мобильных платформах Android и iOS.
- 3) Мощная библиотека классов. .NET представляет единую для всех поддерживаемых языков библиотеку классов. И какое бы приложение мы не собирались писать на C# - текстовый редактор, чат или сложный веб-сайт - так или иначе мы задействуем библиотеку классов .NET.
- 4) Разнообразие технологий. Общезыковая среда исполнения CLR и базовая библиотека классов являются основой для целого стека технологий, которые разработчики могут задействовать при построении тех или иных приложений. Например, для работы с базами данных в этом стеке технологий предназначена технология ADO.NET. Для построения графических приложений с богатым насыщенным интерфейсом - технология WPF. Для создания веб-сайтов - ASP.NET и т.д.
- 5) Также следует отметить такую особенность языка C# и фреймворка .NET, как автоматическая сборка мусора, это значит, что нам в большинстве случаев не придется заботиться об освобождении памяти. Вышеупомянутая общезыковая среда CLR сама вызовет сборщик мусора и очистит память.

- 6) Управляемый и неуправляемый код. Нередко приложение, созданное на C#, называют управляемым кодом (managed code). Это значит, что данное приложение создано на основе платформы .NET и поэтому управляется общеязыковой средой CLR, которая загружает приложение и при необходимости очищает память. Но есть также приложения, например, созданные на языке C++, которые компилируются не в общий язык CIL, как C# или VB.NET, а в обычный машинный код. В этом случае .NET не управляет приложением. В то же время платформа .NET предоставляет возможности для взаимодействия с неуправляемым кодом. Есть возможность наряду со стандартными классами библиотеки .NET также использовать сборки COM.
- 7) JIT-компиляция. Код на C# компилируется в приложения или сборки с расширениями exe или dll на языке CIL. Далее при запуске на выполнение подобного приложения происходит JIT-компиляция (Just-In-Time) в машинный код, который затем выполняется. При этом, поскольку наше приложение может быть большим и содержать кучу инструкций, в текущий момент времени будет компилироваться лишь та часть приложения, к которой непосредственно идет обращение. Если мы обратимся к другой части кода, то она будет скомпилирована из CIL в машинный код. Притом уже скомпилированная часть приложения сохраняется до завершения работы программы, и в итоге это повышает производительность.[5]

2.3.3 ASP .NET MVC5

Платформа ASP.NET MVC представляет собой фреймворк для создания сайтов и веб-приложений с помощью реализации паттерна MVC.

Концепция паттерна (шаблона) MVC (model - view - controller) предполагает разделение приложения на три компонента:

- 1) Контроллер (controller) представляет класс, обеспечивающий связь между пользователем и системой, представлением и хранилищем данных. Он получает вводимые пользователем данные и обрабатывает их. И в зависимости от результатов обработки отправляет пользователю определенный вывод, например, в виде представления.
- 2) Представление (view) - это собственно визуальная часть или пользовательский интерфейс приложения. Как правило, html-страница, которую пользователь видит, зайдя на сайт.

3) Модель (model) представляет класс, описывающий логику используемых данных.

Общую схему взаимодействия этих компонентов можно представить рисунком 2.

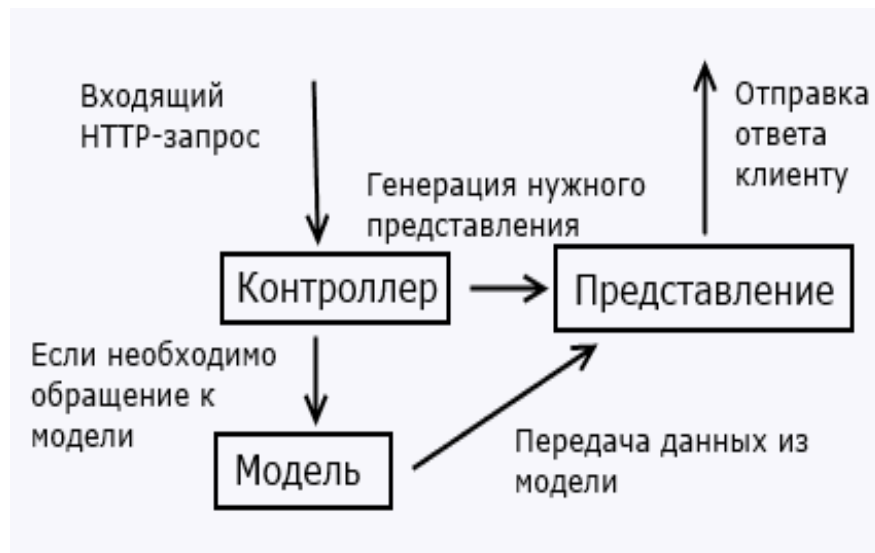


Рисунок 2 - Взаимодействие Модели, Контроллера и Представления

В этой схеме модель является независимым компонентом - любые изменения контроллера или представления не затрагивают модель. Контроллер и представление являются относительно независимыми компонентами, и нередко их можно изменять независимо друг от друга.

Благодаря этому реализуется концепция разделение ответственности, в связи с чем легче построить работу над отдельными компонентами. Кроме того, вследствие этого приложение обладает лучшей тестируемостью. И если нам, допустим, важная визуальная часть или фронтэнд, то мы можем тестировать представление независимо от контроллера. Либо мы можем сосредоточиться на бэкэнде и тестировать контроллер.

Конкретные реализации и определения данного паттерна могут отличаться, но в силу своей гибкости и простоты он стал очень популярным в последнее время, особенно в сфере веб-разработки.

Свою реализацию паттерна представляет платформа ASP.NET MVC. 2013 год ознаменовался выходом новой версии ASP.NET MVC - MVC 5, а также релизом Visual Studio 2013, которая предоставляет инструментарий для работы с MVC 5.

Хотя во многих аспектах MVC 5 не слишком сильно будет отличаться от MVC 4, многое из одной версии вполне применимо к другой, но в то же время есть и существенные отличия:

- 1) В MVC 5 изменилась концепция аутентификации и авторизации. Вместо SimpleMembershipProvider была внедрена система ASP.NET Identity, которая использует компоненты OWIN и Katana.
- 2) Для создания адаптивного и расширяемого интерфейса в MVC 5 используется css-фреймворк Bootstrap.
- 3) Добавлены фильтры аутентификации, а также появилась функциональность переопределения фильтров.
- 4) В MVC 5 также добавлены атрибуты маршрутизации.

Это наиболее важные нововведения в MVC 5. Кроме того, есть еще ряд менее значимых, например, использование по умолчанию Entity Framework 6, некоторые изменения при создании проекта (концепция One ASP.NET), дополнительные компоненты и т.д.[8]

2.3.4 Entity Framework

Для работы с данными в приложении используется фреймворк Entity Framework. Преимущество этого фреймворка состоит в том, что он позволяет абстрагироваться от структуры конкретной базы данных, от написания sql-запросов, и вести все операции с данными через модель. Это позволяет полностью сосредоточиться на логике приложения.

Entity Framework поддерживает подход "Code first", который предполагает сохранение или извлечение информации из БД на SQL Server без создания схемы базы данных или использования дизайнера в Visual Studio. Наоборот, мы создаем обычные классы, а Entity Framework уже сам определяет, как и где сохранять объекты этих классов. Для подключения к базе данных через Entity Framework, нам нужен посредник - контекст данных. Контекст данных представляет собой класс, производный от класса DbContext. Контекст данных содержит одно или несколько свойств типа DbSet<T>, где T представляет тип объекта, хранящегося в базе данных.

Entity Framework (EF) — объектно-ориентированная технология доступа к данным, является object-relational mapping (ORM) решением для .NET Framework от Microsoft. Предоставляет возможность взаимодействия с объектами как посредством LINQ в виде LINQ to Entities, так и с использованием Entity SQL.

Для облегчения построения web-решений используется как ADO.NET Data Services (Astoria), так и связка из Windows Communication Foundation и Windows Presentation Foundation, позволяющая строить многоуровневые приложения, реализуя один из шаблонов проектирования MVC, MVP или MVVM.[8]

2.3.5 Oracle Database

Oracle Database или Oracle RDBMS — объектно-реляционная система управления базами данных компании Oracle.

Oracle Database — одна из самых популярных БД, используемых в настоящее время для разработки ПО, так как она функционирует на большинстве платформ. В том числе на больших ЭВМ, UNIX-серверах, персональных компьютерах и т. д. Другой важной характеристикой является поддержка Oracle всех возможных вариантов архитектур, в том числе симметричных многопроцессорных систем, кластеров, систем с массовым параллелизмом и т. д. Очевидна значимость этих характеристик для крупномасштабных организаций, где эксплуатируется множество компьютеров различных моделей и производителей. В таких условиях фактором успеха является максимально возможная типизация предлагаемых решений, ставящая своей целью существенное снижение стоимости владения программным обеспечением. Унификация систем управления базами данных — один из наиболее значимых шагов на пути достижения этой цели.

Поддержка Oracle большинства популярных компьютерных платформ и архитектур достигается за счет жесткой технологической схемы разработки кода СУБД. Разработку серверных продуктов выполняет единое подразделение корпорации Oracle, изменения вносятся централизованно. После этого все версии подвергаются тщательному тестированию в базовом варианте, а затем переносятся на все платформы, где также детально проверяются. Возможность переноса Oracle обеспечивается специфической структурой исходного программного кода сервера баз данных. Приблизительно 80% программного кода Oracle — это программы на языке программирования С, является платформо-независимым. Примерно 20% кода, представляющее собой ядро сервера, реализовано на машинно-зависимых языках; и эта часть кода, разумеется, переписывается для различных платформ.

Одна из отличительных особенностей сервера Oracle — возможность хранения и обработки различных типов данных. Данная функциональность интегрирована в ядро СУБД и поддерживается модулем interMedia в составе

Oracle Database. Он обеспечивает работу с текстовыми документами, включая различные виды поиска, в том числе контекстного; работу с графическими образами более 20-ти форматов; работу с аудио и видеоинформацией.

СУБД Oracle не только предоставляет расширенный набор встроенных типов данных, но и позволяет за счет использования Object Option конструировать новые типы данных со спецификацией методов доступа к ним. Это означает фактически, что разработчики получают в руки инструмент, позволяющий строить структурированные типы данных, непосредственно отображающие объекты предметной области.

Oracle включает в себя очень много различных компонентов и модулей, в том числе используемые в разработанной системе расширения Holt-Winters Additive и Holt-Winters Multiplicative для Oracle Hyperion Planning Predictive Planning Smart View приложения для сравнения полученных в разработанной системе прогнозов с прогнозами, построенными приложениями Oracle, в рамках тестирования реализованного метода прогнозирования продаж.[8]

2.3.6 AngularJS

AngularJS — JavaScript-фреймворк с открытым исходным кодом. Предназначен для разработки одностраничных приложений (SPA). Его цель — расширение браузерных приложений на основе MVC-шаблона, а также упрощение тестирования и разработки.

Фреймворк работает с HTML, содержащим дополнительные пользовательские атрибуты, которые описываются директивами, и связывает ввод или вывод области страницы с моделью, представляющей собой обычные переменные JavaScript. Значения этих переменных задаются вручную или извлекаются из статических или динамических JSON-данных.

AngularJS спроектирован с убеждением, что декларативное программирование лучше всего подходит для построения пользовательских интерфейсов и описания программных компонентов, в то время как императивное программирование отлично подходит для описания бизнес-логики. Фреймворк адаптирует и расширяет традиционный HTML, чтобы обеспечить двустороннюю привязку данных для динамического контента, что позволяет автоматически синхронизировать модель и представление. В результате AngularJS уменьшает роль DOM-манипуляций и улучшает тестируемость.

Двустороннее связывание данных в AngularJS является наиболее

примечательной особенностью и уменьшает количество кода, освобождая сервер от работы с шаблонами. Вместо этого, шаблоны отображаются как обычный HTML, наполненный данными, содержащимися в области видимости, определенной в модели. Сервис `$scope` в Angular следит за изменениями в модели и изменяет раздел HTML-выражения в представлении через контроллер. Кроме того, любые изменения в представлении отражаются в модели. Это позволяет обойти необходимость манипулирования DOM и облегчает инициализацию и прототипирование веб-приложений.

Причины использования AngularJS:

- 1) Отделение DOM-манипуляции от логики приложения, что улучшает тестируемость кода.
- 2) Разделение клиентской и серверной стороны, что позволяет вести разработку параллельно, получать структурированный код, который уменьшит сложность тестирования.
- 3) Проведение разработчика через весь путь создания приложения: от проектирования пользовательского интерфейса, через написание бизнес-логики, к тестированию.

AngularJS придерживается MVC-шаблона проектирования и поощряет слабую связь между представлением, данными и логикой компонентов. Используя внедрение зависимости, Angular переносит на клиентскую сторону такие классические серверные службы, как видовозависимые контроллеры. Следовательно, уменьшается нагрузка на сервер и веб-приложение становится легче.[8]

3 РАБОТА ПОЛЬЗОВАТЕЛЯ С СИСТЕМОЙ

3.1 Описание функционала системы

Рассмотрим основной функционал, который присутствует в разработанной системе:

- 1) страница авторизации;
- 2) возможность просмотра смоделированных прогнозов продаж;
- 3) возможность фильтрации смоделированных прогнозов продаж по параметрам:
 - a) промежуток времени прогноза;
 - b) наименование продукта;
 - c) категория продукта;
 - d) сезонность и праздники;
 - e) категория сезонных и праздничных товаров;
 - f) события;
 - g) производитель,
 - h) бренд;
 - i) уровень магазинов;
 - j) расположение магазинов и т.д.;
- 4) возможность просмотра деталей прогноза с графиками;
- 5) возможность пользователя скорректировать прогноз, произведенный системой.

3.2 Интерфейс системы

Для начала работы с системой пользователь должен заполнить базу данных продажами за прошедший период и сведениями о товарах, поставщиках и магазинах и запустить web-приложение.

На рисунке 4 показана основная область фильтрации прогнозов продаж системы, на которой можно задать период от 1 недели до 1 года с шагом в 1 неделю, тип поиска по продукту и его расположение.

The image shows a screenshot of a web application interface. At the top, there is a dark blue header bar with the text 'Search Criteria' in white. Below this, there is a section titled 'Working Levels' in a light blue font. Underneath, there is a search form with four main fields: 'From' with a date picker set to '06/19/2016', 'To' with a date picker set to '06/17/2017', 'Product' with a dropdown menu showing 'PLN', and 'Location' with a dropdown menu showing 'Chain'. Each field has a small downward arrow icon indicating it is a dropdown or date picker.

Рисунок 4 – Основная область фильтрации прогнозов

На рисунке 5 показана область фильтрации прогнозов продаж системы, на которой можно задать критерии продукта, такие, как id, поставщик, бренд, категория и подкатегория товара.



Рисунок 5 – Область фильтрации прогноза продаж по критериям продукта

На рисунке 6 показана область фильтрации прогнозов продаж системы, на которой можно задать дополнительные критерии продукта, такие, как сезонность и праздники (например, Рождество), категорию сезонных товаров (например, рождественские свечи) и события (например, промо-акции и скидки).



Рисунок 6 - Область фильтрации прогноза продаж по дополнительным критериям

На рисунке 7 показана область фильтрации прогнозов продаж системы, на которой можно задать критерии расположения продуктов, такие, как выбор определенного магазина, региона или страны.



Рисунок 7 - Область фильтрации прогноза продаж по расположению

Отфильтруем прогнозы по критериям, показанным на рисунке 8, а результат фильтрации отображены на рисунке 9 в виде таблицы прогнозов с пагинацией, где каждая строка таблицы – это прогноз для определенного продукта.



Рисунок 9 – Выбранные критерии для фильтрации продаж



Рисунок 10 – Результаты фильтрации с выбранными критериями

На рисунке 11 показано, то ряд таблицы , то есть прогноз для одного продукта, содержит информацию об id продукта, его названии, производителе, спрогнозированное число продаж единиц, цену за единицу и колебания прогноза при разных условиях.

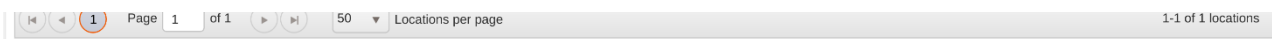


Рисунок 11 – Выбран один прогноз для просмотра

Также пользователь может выбрать любой ряд таблицы, то есть прогноз для одного продукта, чтобы посмотреть его детали на графике и скорректировать его, что изображено на рисунке 12.



Рисунок 12 – Детальная информация о прогнозе и возможность скорректировать его

На рисунке 13 изображены варианты корректирования прогноза (процент или значение от определенного типа прогноза).

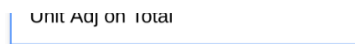


Рисунок 13 – варианты корректирования прогноза для одного продукта

На рисунке 14 изображено, какие параметры были выбраны для корректирования прогноза, а на рисунке 15 изображен скорректированный прогноз.



Рисунок 14 – Выбранные параметры корректирования прогноза



Рисунок 15 – Скорректированный прогноз

ЗАКЛЮЧЕНИЕ

В работе:

- 1) проведен анализ предметной области и методов для прогнозирования продаж;
- 2) проанализированы и описаны особенности прогнозирования продаж;
- 3) выбран и реализован метод прогнозирования продаж для крупного бизнеса;
- 4) изучен фреймворк .NET 4.5;
- 5) изучена технология ASP .NET MVC5;
- 6) изучен фреймворк Entity Framework
- 7) изучена Oracle Database
- 8) изучен фреймворк AngularJS
- 9) спроектирован и разработан прототип системы, позволяющий прогнозировать продажи для крупного бизнеса.

В дальнейшем приложение может быть расширено посредством добавления нового функционала, например, возможность экспортировать прогнозы в удобном формате.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Adebajo Dotun, Mann Robi. Identifying problems in forecasting consumer demand in the fast moving consumer goods sector // An International Journal. Comput. – 2000. Vol. C – 7. No – 3. - P. 223-230
2. Лукашин Ю. П. Адаптивные методы краткосрочного прогнозирования временных рядов; Учеб. пособие. - М.: Финансы и статистика, 2003. - 416с.
3. Прогноз (общие сведения) // Врата к будущему [Электронный ресурс]. - 2010. - Режим доступа : <http://future.vx9.ru/?p=28>. - Дата доступа : 15.12.2016.
4. Катаева В.И. Методы принятия управленческих решений : учебное пособие / В. И. Катаева, М. С. Козырев – М.-Берлин : Директ-Медиа, 2015. – 196с.
5. Кузнецов, Б. Т. Финансовый менеджмент: Учебное пособие для студентов вузов, обучающихся по специальностям 060500 «Бухгалтерский учет», 060400 «Финансы и кредит» / Б.Т. Кузнецов. — М.: ЮНИТИ-ДАНА, 2005. - 415с.
6. Модель Хольта-Винтерса - 4analytics.ru // 4analytics.ru Решения для бизнес-анализа [Электронный ресурс]. - 2016. - Режим доступа : <https://4analytics.ru/prognostirovanie/prognost-po-metodu-eksponencialnogo-sglajvaniya-s-trendom-i-sezonnostyu-xolta-vintersa.html>. - Дата доступа : 18.12.2016.
7. Winters P.R. Forecasting sales by exponentially weighted moving averages // Management Science. - 1960. Vol. C - 6. No - 3.
8. METANIT.COM Сайт о программировании // METANIT.COM Сайт о программировании [Электронный ресурс]. - 2016. - Режим доступа : <https://metanit.com/>. - Дата доступа : 20.12.2016.