3 Подсистема классификации изображений, которая отвечает за процесс классификации изображений. Она включает в себя:

- Подсистему создания и обучения нейросети, которая отвечает за реализацию обучения нейронной сети и создания модели.
- Подсистему обработки изображений, которая отвечает за реализацию классификации загруженных пользователем изображений по сохранённой модели.

Заключение

Авторами была проведена апробация нейронной сети VGG-16 на обучающей выборке в 2500 картинок, относящимся к 10 классам. В результате обучения классификатора были получены следующие результаты: к первому классу классификатор отнёс 10 из 10, ко 2 – 9 из 10, к 3 – 6 из 10, к 4 – 4 из 10, к 5 – 9 из 10, к 6 – 9 из 10, к 7 – 9 из 10, к 8 – 49 из 10, к 9 – 6 из 10, к 10 – 5 из 10. Таким образом, классификатор определяет принадлежность изображений к классам с вероятностью от 0,69 до 0,83.

Литература

1 Компьютерное зрение [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Компьютерное_зрение (дата обращения: 25.03.2022).

2 Лекция 2 | Классификация изображений [Электронный ресурс]. URL: https://russianblogs.com/article/19571391203/ (дата обращения: 25.03.2022).

3 VGG16 – сверточная сеть для выделения признаков изображений [Электронный ресурс]. URL: https://neurohive.io/ru/vidy-nejrosetej/vgg16-model/ (дата обращения: 02.04.2022).

4 Softmax [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Softmax (дата обращения: 05.04.2022).

5 Rectifier (neural networks) [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Rectifier_(neural_networks) (дата обращения: 05.04.2022).

Д.А. Дасаева, В.В. Мокшин

ОБЗОР МЕТОДОВ ПРОГНОЗИРОВАНИЯ СПРОСА НА ОНЛАЙН ПЛОЩАДКАХ

(Казанский национальный исследовательский технический университет имени А. Н. Туполева)

Введение

По данным исследования M.A.Research, в 2021 году онлайн торговля стала самым быстро растущим сегментом ретейла, а оборот рынка продаж в сети интернет вырос на 32% — до 4,2 трлн рублей[1]. С ростом спроса на

онлайн сегмент рынка возрастает и количество задач связанных с онлайн торговлей. К таким задачам относят: прогнозирование продаж на маркетплейсах, предсказание трендовости товаров и многое другое. Проблемой, которая часто возникает при обработке информации[2] и прогнозировании спроса, является большое количество не всегда связанных между собой параметров. Системы мониторинга и прогнозирования являются сложными много параметрическими системами[3], которые не всегда дают положительный прогноз. Причинами неэффективного планирования и прогнозирования является устаревание существующих технических систем, отчасти – использование не в полной мере объёма информации и недостаточный общий уровень квалификации персонала[4]. В данной статье предлагаются методы, сочетающие человеческую идентификацию знаний и машинного обучения для устранения этих проблем[5].

Формальная постановка задачи

Обозначим множество входных параметров через $X = \{x_1, x_2, \dots x_n\}$, тогда $Y = \{y_1, y_2, \dots y_n\}$ множество допустимых ответов. Существует неизвестная целевая зависимость – отображение $y^*: X \to Y$, значения которой известны только на объектах конечной обучающей выборки $X^m = \{(x_1, y_1), (x_m, y_m)\}$. Требуется построить алгоритм $\alpha: X \to Y$, который приближал бы неизвестную целевую зависимость как на элементах выборки так и на всем множестве X[6]. В общем виде модель будет записываться следующим образом y=f(X).

Методы решения поставленной задачи

Существует множество методов решения задачи прогнозирования, например, линейная регрессия. Линейная регрессия — модель зависимости переменной *x* от одной или нескольких других переменных с линейной функцией зависимости[7]. Другим методам прогнозирования продаж на онлайн площадках является алгоритм случайного леса.

Случайный лес — является одним из немногих универсальных алгоритмов. Универсальность заключается, в том, что 70% задач можно решить с его помощью, и в том, что есть случайные леса для решения задач классификации, регрессии, кластеризации, поиска аномалий, селекции признаков и так далее[8].

Для прогнозирования зачастую используются средства имитационного моделирования. Имитационное моделирование (симуляция) – это распространенная разновидность аналогового моделирования, реализуемого с помощью набора математических средств, специальных компьютерных программсимуляторов и особых IT, позволяющих создавать в памяти компьютера процессыаналоги, с помощью которых можно провести целенаправленное исследование структуры и функций реальной системы в режиме ее «имитации», осуществить оптимизацию некоторых ее параметров[9].

Еще одним методом прогнозирования продаж является метод XGBoost. T В его основе лежит алгоритм градиентного бустинга деревьев решений. Градиентный бустинг — это техника машинного обучения для задач

классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей, обычно деревьев решений. Обучение ансамбля проводится последовательно в отличие, например от бэггинга. На каждой итерации вычисляются отклонения предсказаний уже обученного ансамбля на обучающей выборке. Следующая модель, которая будет добавлена в ансамбль будет предсказывать эти отклонения. Таким образом, добавив предсказания нового дерева к предсказаниям обученного ансамбля мы можем уменьшить среднее отклонение модели, которое является целью оптимизационной задачи. Новые деревья добавляются в ансамбль до тех пор, пока ошибка уменьшается, либо пока не выполняется одно из правил "ранней остановки"[10].

Существует решение поставленной задачи с помощью градиентного бустинга. LightGBM — это фреймворк, который предоставляет реализацию деревьев принятия решений с градиентным бустингом. LightGBM известен своей более высокой скоростью обучения, хорошей точностью с параметрами по умолчанию, параллельным и GPU обучением, малым использованием памяти и возможностью обработки больших датасетов, которые не всегда помещаются в ней.

Математическая модель линейной регрессии

Математическая модель линейной регрессии записывается следующим образом:

$f(x) = m \cdot x + b$, где m — наклон линии, а b — его y-сдвиг.

Таким образом, решение линейной регрессии определяет значения для m и b, так что $f(x)$ приближается как можно ближе к y. Для определения качества ошибок линейной регрессии используется функция потерь.

Функция потерь — это мера количества ошибок, которые линейная регрессия делает на наборе данных. Существуют разные функции потерь, но все они вычисляют расстояние между предсказанным значением $y(x)$ и его фактическим значением.

Одна из самых распространенных функций потерь называется средней квадратичной ошибкой(MSE). Для ее вычисления мы берем все значения ошибок, считаем их квадраты длин и усредняем.

В общем случае, если есть n переменных, их математическая модель может быть записана как:

$$f(x) = b + w_1 \cdot x_1 + \ldots + w_n \cdot x_n$$

Постановка задачи остается одинаковой, независимо от количества измерений.

Математическая модель Random Forest

Рассмотрим размеченную выборку объектов $\{(x_i, y_i)\}_{i=1}^{N}$, где $x_i \in R^2$ — признаковое описание объекта в двумерном пространстве, а $y_i \in \{0, 1\}$ — метка класса:

Несмотря на то, что в середине объекты разных классов сильно перемешаны, при помощи дерева решений с такой выборкой достаточно удобно работать: на каждом шаге необходимо выбирать признак и значения порога, по ко-

торому происходит оптимальное по заданному критерию разбиение. При решении прикладных задач часто используются следующие критерии:

$$iGain(S) = H(S) - \sum_{v\epsilon\{L,R\}} \frac{S_v}{S} H(S_v),$$

$$H(S) = -\sum_{c\epsilon C} p_c log_2(p_c),$$

где C – множество классов рассматриваемой задачи, а p_c – вероятность класса c для множества объектов S;

При каждом делении все объекты делятся на две более мелкие группы, т.е. рассматриваемая в каждом из узлов задача разбивается на две более мелкие подзадачи. Заданием максимального числа объектов в вершине-листе дерева устанавливается один из возможных критериев останова для алгоритма.

Таким образом, можно достаточно качественно классифицировать рассматриваемую выборку объектов при помощи всего одного дерева решений, если в качестве ответа для тестового объекта, попавшего в ячейку A_{i}, выдавать номер наиболее часто встречающегося в этой ячейке класса.

Однако в реальных задачах часто встречаются погрешности в измерениях и объекты-выбросы, которые серьезно портят качество классификации одним конкретным деревом решений. Поэтому перед построением каждого нового дерева происходит сэмплирование с повторениями новой выборки $\{(x_{\text{i}}{}^k, y_{\text{i}}{}^k)\}_{i=1}^N$ из $\{(x_{\text{i}}, y_{\text{i}})\}_{i=1}^N$на которой происходит обучение дерева с номером k. После построения всех деревьев каждый тестовый объект z_{i}получает в качестве промежуточного ответа вектор меток, присвоенных ему каждым деревом, который преобразуется в финальную метку по методу простого голосования[11].

XGBoost

Функция для оптимизации градиентного бустинга выглядит следующим образом:

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

где l – функция потерь

$y_i, \hat{y_i}^{(t-1)}$ –значения i-го элемента обучающей выборки и сумма предсказаний первых t деревьев соответственно.

x_i –набор признаков i-го элемента обучающей выборки

f_t –функция(в нашем случае дерево), которую мы хотим обучить на шаге t.

Градиентный бустинг (LightGBM)

Рассмотрим задачу распознавания объектов из многомерного пространства X с пространством меток Y. Пусть нам дана обучающая выборка $\{x_i\}_{i=1}^N$, $x_i \in X$. И пусть на ней известны истинные значения меток каждого объекта $\{y_i\}_{i=1}^N$, где $y_i \in Y$. Необходимо построить распознающий оператор, который как можно более точно сможет предсказывать метки для каждого нового объекта $x \in X$.

Пусть нам задано некоторое семейство базовых алгоритмов H, каждый элемент $h(x;a) \in H : X \to R$ которого определяется некоторым вектором параметров $a \in A$.

Построение композиции

Будем искать финальный алгоритм классификации в виде композиции

$$F_M(x) = \sum_{m=1}^{M} b_m h(x;a_m), b_m \in R, a_m \in A.$$

Однако подбор оптимального набора параметров $\{a_m, b_m\}_{m=1}^{M}$ – очень трудоемкая задача. Поэтому мы будем пытаться построить такую композицию путем жадного наращивания, каждый раз добавляя в сумму слагаемое, являющееся наиболее оптимальным алгоритмом из возможных. Будем считать, что нами уже построен классификатор F_{m-1} длины m-1. Таким образом задача сводится к поиску пары наиболее оптимальных параметров $\{a_m, b_m\}$ для классификатора длинны m:

$$F_m(x) = F_{m-1}(x) + b_m h(x;a_m), b_m \in R, a_m \in A.$$

Оптимальность здесь понимается в соотвествии с принципом явной максимизации отступов. Это означает, что вводится некоторая функция потерь $L\big(y_i, F_m(x_i)\big), i = \overline{1,N}$, показывающая, насколько “сильно” предсказанный ответ $F_m(x_i)$ отличается от правильного ответа y_i. И затем минимизируется функционал ошибки

$$Q = \sum_{i=1}^{N} L\big(y_i, F_m(x_i)\big) \to min$$

Заметим, что функционал ошибки Q – вещественная функция, зависящая от точек $\{F_m(x_i)\}_{i=1}^{N}$ в N-мерном пространстве, и нам необходимо решить задачу минимизации этого функционала. Сделаем это, реализуя всего один шаг метода градиентного спуска. В качестве точки, для которой мы будем искать оптимальное приращение, рассмотрим F_{m-1}. Найдем градиент функционала ошибки:

$$\nabla Q = \left[\frac{\partial Q}{\partial F_{m-1}}(x_i)\right]_{i=1}^{N} = \left[\frac{\partial(\sum_{i=1}^{N} L(y_i, F_{m-1}))}{\partial F_{m-1}}(x_i)\right]_{i=1}^{N} = \left[\frac{\partial L(y_i, F_{m-1})}{\partial F_{m-1}}(x_i)\right]_{i=1}^{N}$$

Таким образом, в силу метода градиентного спуска, наиболее выгодно добавить новое слагаемое в классификатор следующим образом:

$$F_m = F_{m-1} - b_m \nabla Q, b_m \in R,$$

где b_mподбирается линейным поиском по вещественным числам R:

$$b_m = \underset{b \in R}{arg\min} \sum_{i=1}^{N} L(F_{m-1}(x_i) - b\nabla Q_i)$$

Однако ∇Qпредставляет из себя лишь вектор оптимальных значений для каждого объекта x_i, а не базовый алгоритм из семейства H, определенный $\forall x \in X$. Поэтому нам необходимо найти $h(x, a_m) \in$H наиболее похожий на

∇Q[12]. Сделаем это, опять минимизируя функционал ошибки, основанный на принципе явной максимизации отступов:

$$a_m = \underset{a \in A}{arg\min} \sum_{i=1}^{N} L(\nabla Q_i, h(x_i, a)) \equiv \text{обучить}\left(\{x_i\}_{i=1}^{N}, \{\nabla Q_i\}_{i=1}^{N}\right),$$

что просто соответствует базовому алгоритму обучения. Далее найдем коэффициент b_m, используя линейный поиск:

$b_m = arg\min_{b \in R} \sum_{i=1}^{N} L(F_{m-1}(x_i) - bh(x_i, a_m))$

Применение методов на реальных данных

В ходе разработки были использованы следующие данные о продажах товаров на онлайн площадке: стоимость товара, количество товара проданного за каждый день в последние 6 месяцев, среднее количество товара продаваемого в неделю, месяц[13].

Примеры исходных данных, используемых для обучения алгоритмов отображены в таблице 1.

Таблица 1

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
9374	носки	499	59	93994	199	54	100102	2
9933	носки	719	30	93994	199	54	100102	3

где x_1 – артикул товара, x_2 – категория, x_3 – стоимость, x_4 – среднее количество продаж за месяц, x_5 – количество товаров в категории, x_6 – средняя цена в категории в рублях, x_7 – установленная скидка на товар в процентах, x_8 – среднее количество запросов в поисковике, x_9 – время доставки заказа в днях.

Оценка качества модели прогнозирования

Для оценки качества модели необходимо оценить способность прогнозирования полученной модели[14] с помощью метрики MAE (Mean Absolute Error) – средней абсолютной ошибки. Метрика измеряет среднюю сумму абсолютной разницы между фактическим значением и прогнозируемым значением[15].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|e_i|,$$

где $e_i = y_i - a_i$, где y – фактическое значение признака, a – прогнозируемое значение.

В ходе проведения исследования были реализованы алгоритмы прогнозирования и выполнена оценка качества моделей прогнозирования с помощью вычисления средней абсолютной ошибки. Полученные результаты описаны в таблице 2.

Таблица 2

Модель прогноза	Время обучения	Средняя абсолютная ошибка
Линейная регрессия	10 секунд	1.15
Случайный лес	700 секунд	1.10
XGBoost	1200 секунд	1.03

LightGBM	600 секунд	1.01

Ниже представлен график количества продаж и прогноза продаж каждого из методов за период с 1 по **31** октября 2018 года.

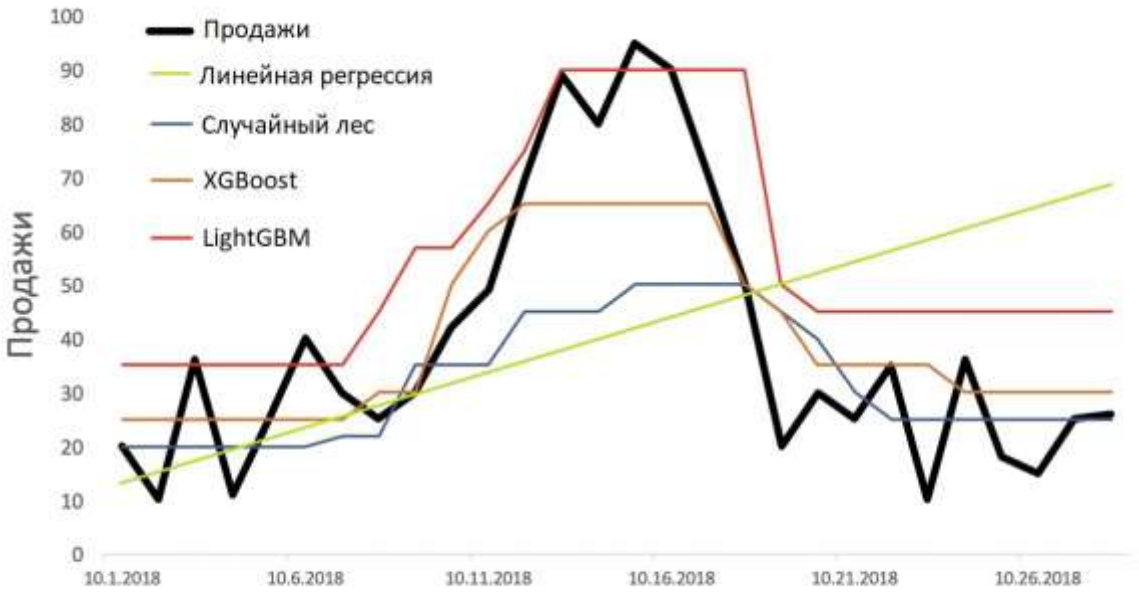

Рис. 1. Сравнение прогнозов продаж с количеством продаж

Заключение

В данной статье были рассмотрены методы прогнозирования спроса на онлайн площадках. После вычисления средней абсолютной ошибки для каждого метода было выявлено, что градиентный бустинг (LightGBM) показывает наилучшие результаты и хорошее время обучения алгоритма.

Литература

1. Малаховский, Алексей. Маркетплейсы: как регулируется их работа в России [Электронная версия]: https://pravo.ru/story/239011/ (Дата обращения 3.04.2022)

2. Группирование контуров объектов структурных изображений на основе сети заметности элементов Сайфудинов И.Р., Мокшин В.В., Кирпичников А.П. Вестник Технологического университета. 2017. Т. 20. № 9. С. 120-123.

3. Разработка системы мониторинга состояния грузоподъемных механизмов Мокшин В.В., Якимов И.М., Кирпичников А.П., Шарнин Л.М. Вестник Технологического университета. 2017. Т. 20. № 19. С. 75-81.

4. Применение математических моделей и алгоритмов при планировании и прогнозировании потребления водных ресурсов Мокшин В.В, Спиридонова А.В., Спиридонов Г.В., Вестник Воронежского государственного технического университета. 2021. Т. 17. № 4. С. 57-64.

5. Распознавание образов транспортных средств на основе эвристических данных и машинного обучения Мокшин В.В., Сайфудинов И.Р., Кирпичников А.П., Шарнин Л.М. Вестник Технологического университета. 2016. Т. 19. № 5. С. 130-137.

6. Задача классификации [Электронный портал]: Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8 (Дата обращения 7.02.2022)

7. Линейная регрессия в машинном обучении [Электронный ресурс]: Обучающий портал Neurohive. – Режим доступа: https://neurohive.io/ru/osnovy-data-science/linejnaja-regressija/ (Дата обращения: 21.10.2021)

8. Дьяконов, Александр. Случайный лес (Random forest) [Электронный ресурс]: Научный блог Александра Дьяконова. – Режим доступа: https://dyakonov.org/2016/11/14/случайный-лес-random-forest/ (Дата обращения: 22.10.2021)

9. Имитационное моделирование [Электронный ресурс]: Корпоративный портал Томский политехнический университет / лекция. – Режим доступа: https://portal.tpu.ru/SHARED/i/INNA/umkd/Tab/lek_3.pdf#:~:text=Имитационное%20моделирование%20(симуляция)%20–%20это,осуществить%20оптимизацию%20некоторых%20ее%20параметров (Дата обращения: 13.01.2022)

10. XGBoost [Электронный ресурс]: Университет ИТМО/ Электронные текстовые данные. – Режим доступа: https://neerc.ifmo.ru/wiki/index.php?title=XGBoost (Дата обращения: 24.10.2021)

11. Рыжков, Александр. Композиция алгоритмов, основанные на случайном лесе [Электронная версия]: Дипломная работа, Рыжков А.М. / Москва 2015. –Режим доступа: http://www.machinelearning.ru/wiki/images/d/d8/2015_517_RyzhkovAM.pdf (Дата обращения 8.02.2022)

12. Фонарев, Александр. Обзор алгоритмов бустинга [Электронный ресурс]: Курсовая работы Фонарев А. Ю. / Московский Государственный Университет имени М.В. Ломоносова Факультет Вычислительной Математики и Кибернетики Кафедра Математических Методов Прогнозирования, Москва, 2012. – Режим доступа:http://www.machinelearning.ru/wiki/images/9/9a/fonarev.overview_of_boosting_methods.pdf (Дата обращения: 24.10.2021)

13. Прогнозирование продаж интернет-магазина с помощью градиентного бустинга [Электронный ресурс]: Научная онлайн конференция Highload / Москва 2018. – Режим доступа: https://www.highload.ru/moscow/2018/abstracts/4344 (Дата обращения: 29.10.2021)

14. Параллельный генетический алгоритм отбора значимых факторов, влияющих на эволюцию сложной системы Мокшин В.В. Вестник Казанского государственного технического университета им. А.Н. Туполева. 2009. № 3. С. 89-93.

15. Выбор метрики в машинном обучении (Random forest) [Электронный ресурс]: Блог кампании Деталитика. – Режим доступа: http://blog.datalytica.ru/2018/05/blog-post.html (Дата обращения: 07.11.2021)

А.Ю. Жигалов, И.П. Болодурина, Д.И. Парфенов, Л.С. Гришина

РАЗРАБОТКА ГРАФОВОЙ МОДЕЛИ СТРУКТУРНЫХ И СЕМАНТИЧЕСКИХ ОТНОШЕНИЙ МЕЖДУ СУЩНОСТЯМИ ДОКУМЕНТОВ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ

(Оренбургский государственный университет)

Сегодня наблюдается взрывной рост количества информации, создаваемой людьми и машинами на естественном языке. Аналитическое агентство IDC прогнозирует рост совокупного объема данных, накопленных человечеством, до 163 зеттабайт к 2025 году. Основной частью таких данных являются неструктурированные данные, такие как фотографии, видеозаписи, аудиозаписи, а также тексты на естественном языке. Постоянное увеличение интенсивности потока входящей текстовой информации делает все более важной задачу обработки естественного языка, в частности — русского языка.

Важнейшей проблемой является лексическая многозначность, требующая от машины понимания контекста и предметной области, в которой употребляется каждое многозначное слово [1]. Такие сведения представляются в семантических сетях — специальных высококачественных базах знаний, представляющих машиночитаемые сведения об окружающем мире в виде понятий и связей между ними. Связи между понятиями задают семантическую иерархию, которая позволяет решать различные задачи машинного понимания естественного языка.

В настоящее время обработка естественного языка (Natural language processing, NLP) является наиболее инновационным направлением искусственного интеллекта. При решении многих задач NLP, таких, как распознавание и синтез речи, машинный перевод, классификация текстов, разработка диалоговых систем, в последнее время достигнут значительный прогресс на основе нейросетевых методов машинного обучения [2]. В первую очередь, исследователи занимаются решением универсальных задач, которые могут найти применение в различных областях таких как финансы, медицина, медиа и реклама. К таким задачам можно отнести генерацию продолжения текста (сети GPT-2,