МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАБОТА ПРОВЕРЕНА

Рецензент, руководитель
научно-исследовательской группы
ООО «Инновационные технологии»
_____А. С. Широносов
“___”_____ 2017 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
профессор
_____ Л.Б. Соколинский
“___”_____ 2017 г.

РАЗРАБОТКА АГРЕГАТОРА СПЕЦИАЛИЗИРОВАННОЙ ИНФОРМАЦИИ С ОТКРЫТЫХ WEB-СТРАНИЦ СЕТИ ИНТЕРНЕТ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2017.13-014-1382.ВКР

Научный руководитель

к.ф.-м.н.

_____В.А. Голодов

Автор работы,
студент группы КЭ-401

_____Г.А. Игнатов

Ученый секретарь
(нормоконтролер)

_____ О.Н. Иванова

“___”_____ 2017 г.

Челябинск-2017

ОГЛАВЛЕНИЕ

ГЛОССАРИЙ

Фотобанк – это банк изображений, который выступает посредником между авторами изображений и их покупателями. Он берет на себя задачу поиска покупателей и приема платежей, что значительно упрощает жизнь фотографам и иллюстраторам.

Сайт – совокупность логически связанных между собой веб-страниц; также место расположения контента сервера.

Веб-страница – документ или информационный ресурс Всемирной паутины, доступ к которому осуществляется с помощью веб-браузера.

Верстка веб-страниц – структура html-кода, размещающего элементы веб-страницы (изображения, текст и т. д.) в окне браузера, согласно разработанному макету.

DOM дерево – объектная модель, используемая для XML/HTML-документов. Согласно DOM-модели, документ является иерархией, деревом. Каждый HTML-тег образует узел дерева с типом «элемент». Вложенные в него теги становятся дочерними узлами. Для представления текста создаются узлы с типом «текст».

Тело веб-страницы – содержит графическое и информационное представление веб-страницы. Тело веб-страницы ограничивается элементом <body>.

Объект/блок/элемент веб-страницы – узел DOM дерева.

CSS – формальный язык описания внешнего вида документа, написанного с использованием языка разметки. CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц.

CSS характеристика – пара вида свойство-значение, которая описывает представление внешнего вида элемента.

CSS селектор – формальное описание того элемента или группы элементов, к которым применяются CSS характеристики.

Машинное обучение – класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решению множества сходных задач.

Модель машинного обучения – метод искусственного интеллекта.

Обучающая выборка – выборка, по которой производится построение алгоритма машинного обучения.

Тестовая выборка – выборка, на которой тестируется эффективность алгоритма машинного обучения. Тестовая выборка не должна пересекаться с обучающей выборкой.

Классификатор – метод искусственного интеллекта, решающий задачу классификации на заданное число классов.

Признак/атрибут – характеристика объекта обучающей выборки. Совокупность всех характеристик объекта формирует вектор признаков.

Линейный классификатор – классификатор, определяющий метку класса на основе построения разделяющей плоскости между объектами обучающей выборки, которые принадлежат к разным классам.

Метрика качества – мера, позволяющая получить численное значение качества работы модели.

Переобучение – идеальная подстройка модели под обучающую выборку, что приводит к снижению способности модели к обобщению данных.

ВВЕДЕНИЕ

Всемирная паутина стала ценным источником информации для многих интернет-пользователей. Существует много типов веб-сайтов, которые ориентируются на предоставление определенной информации: новости, блоги, интернет-магазины, карты, биржи, фото, видео и т.д. С каждым днем количество таких сайтов растет, количество различной информации увеличивается, что создает проблемы для рядового пользователя интернета в поиске специфических продуктов и информации.

Чтобы решить данную проблему были придуманы агрегаторы сайтов. Их целью является предоставление удобного пользовательского интерфейса для поиска и доступа к данным, которые расположены на других сайтах [1]. В качестве примера стоит взглянуть на следующие сервисы: Яндекс.Маркет [20] – сервис для поиска и покупки товаров, Яндекс.Каршеринг [19] – сервис для поиска и аренды автомобилей, Slickdeals [14] – сервис для поиска и проведения различного вида сделок, The Stocks [15] – сервис поиска стоковых изображений.

Тем не менее, подавляющее большинство веб-страниц проектируются с ориентацией на удобство восприятия для человека, что часто затрудняет автоматическое агрегирование информации и понимание структуры веб-страниц машинами.

Обычно у агрегирующих сервисов высокая пользовательская конверсия (отношение числа посетителей сайта, выполнивших целевые действия, например, покупка товара или заказ услуги, к общему числу посетителей), что является стимулом для агрегируемых сайтов самостоятельно делиться данными для повышения качества предоставления информации пользователям и поддержания ее актуальности. В ином случае, агрегатору сайтов приходится подстраиваться под верстку или API сайта на котором размещена информация, чтобы быть уверенными в том, что получаемые данные верны и соответствуют друг другу (например, цена, заголовок

и описание должны относиться к одному и тому же товару на веб-странице).

Такой подход к сбору данных имеет один серьезный недостаток: зачастую рассматриваемые агрегатором сайты стремятся изменить шаблон сайта на более удобный и функциональный, что подразумевает изменение верстки веб-страниц. К тому же с ростом числа сайтов, возрастают временные затраты на освоение программистом API или структуры веб-страниц для правильной настройки поискового робота.

Данная работа ставит перед собой следующую цель: изучив существующие решения проблемы извлечения данных из открытых веб-страниц, создать универсальное решение проблемы сбора информации, которое способно извлекать данные независимо от структуры и стиля веб-страниц сайтов заданной тематики. Исходя из сложности и объема задачи, было принято решение заранее обозначить тематику сайтов и тип извлекаемых данных. Выбор автора данной работы пал на сайты фотобанков из-за высокой вариативности верстки веб-страниц и разнообразия размещаемого контента. В качестве извлекаемой информации примем название (заголовок) продаваемого изображения на веб-странице, как наиболее содержательную, после самого изображения, часть данных.

Целью данной работы является разработка интеллектуального робота, который осуществляет извлечение заголовков продаваемых изображений с веб-страниц фотобанков с помощью алгоритма машинного обучения.

Для достижения указанной цели необходимо решить следующие задачи:

1) изучение существующих решений сбора и анализа данных с веб-страниц;
2) определение признакового пространства;
3) выбор и обучение модели машинного обучения;
4) разработка агрегатора;

5) проведение тестирования качества алгоритма машинного обучения;

6) проведение тестирования агрегатора.

Структура и объем работы

Работа состоит из введения, четырех разделов, заключения и библиографии. Объем работы составляет 36 страниц, объем библиографии – 20 источников, объем приложений – 4 страницы.

В первой главе дается обзор на целевую категорию сайтов, а также на существующие подходы к решению задачи интеллектуального извлечения данных из веб-страниц.

Вторая глава содержит описание процесса настройки модели машинного обучения.

В третьей главе представлена реализация агрегатора заголовков веб-страниц фотобанков.

Четвертая глава посвящена тестированию агрегатора заголовков веб-страниц фотобанков.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Введение в область

На веб-страницах заголовки обычно присутствуют в 2-х местах и служат разным целям. Во-первых, заголовок находится в теле веб-страницы и имеет визуальное выделение по отношению к основному контенту. Это важно для читателя, который читает веб-страницу с экрана монитора. Во-вторых, заголовок помещается между тегами <title> и </title> специально для поисковых роботов и программ, генерирующих краткое описание страницы. Однако, дизайнеры веб-страниц часто забывают о данной практике или, что хуже, помещают в title-теги не релевантный текст (например, ключевые слова или контактные данные).

Большинство существующих решений проблемы извлечения заголовков с веб-страниц нацелены на ресурсы, предоставляющие, в основном, текстовый контент (блоги, новостные сайты, образовательные сайты и т.д.).

Перечислим признаки, характерные для заголовков текстового контента. Такой заголовок:

1) всегда находится над основным текстом;

2) всегда имеет визуальное выделение по отношению к основному тексту (шрифт, цвет, толщина линии);

3) всегда имеет удобочитаемую длину текста (в основном, не более 6 слов)

4) всегда представлен на используемом языке сайта.

Для фотобанков заголовок изображения является второстепенной информацией – само изображение воспринимается человеком мгновенно и не нуждается в аннотации, в отличии от текста. Поэтому заголовок на сайтах фотобанков:

1) может располагаться где угодно относительно главного изображения (выше, ниже, сбоку);

2) может не иметь визуального выделения;

3) длина текста может значительно варьироваться (от 1 символа до 2-3 полноценных предложений);

4) язык заголовка может отличаться от используемого на сайте. Более того заголовок изображения может состоять из 2-х разных языков (например, «鎌倉を観光する美人女性 – Сток картинки», – заголовок на корейском и русском языках одновременно).

С учетом этих особенностей сайтов фотобанков далее приведен анализ существующих решений проблемы извлечения заголовков веб-страниц.

1.2. Обзор существующих решений

Решение задачи определения заголовка с помощью машинного обучения предполагает выбор модели машинного обучения и подбор информативных атрибутов, по которым модель сможет классифицировать html-объект как содержащий заголовок.

Работа [18] описывает концепцию извлечения контента с помощью машинного обучения без учителя (unsupervised learning). В основе программы лежит алгоритм кластеризации DBSCAN. В качестве признаков текстовых блоков выбраны следующие: длина текста, путь от корня документа до блока, CSS селекторы и CSS свойства. Автор предполагает, что после обучения на обширной выборке веб-страниц и сайтов модель сможет присваивать одинаковые метки кластеров похожим друг на друга html-объектам, что в дальнейшем можно использовать для определения их семантики (например, кластер № 1 – заголовок, кластер № 2 – основной текст, кластер № 3 – реклама и т.п.). Такой подход эффективен в задаче индексации веб-страниц, потому что предполагает отсутствие спецификации извлекаемой информации и обход большого количества сайтов (качество кластеризации растет по мере обучения модели на новых данных), и,

соответственно, неэффективен в задаче извлечения конкретного типа данных из относительно небольшого набора сайтов.

В работе [3] описан подход, который основан на двух предположениях: горизонтальная позиция заголовка должна быть между боковыми границами основного текста и размер шрифта не должен быть меньше чем у основного текста веб-страницы. Исходя из этих предположений, формируются 4 признака для определения заголовка на странице: текст блока заключен между тегами <title>...</title>, текст блока заключен между тегами <h[1-6]>...</h[1-6]>, размер шрифта, расположение текстового блока по отношению к главному тексту страницы. Оценка принадлежности к заголовку для каждого блока формируется с помощью, предложенной авторами, формулы – машинное обучение не используется. Данное решение базируются на неверных для фотобанков предположениях – набор признаков недостаточен для идентификации заголовка на некоторых сайтах фотобанков.

В работе [5] для определения заголовка на веб-странице используется 5 групп признаков:

1) формат текста:
- размер шрифта;
- толщина линии текста: обычный или **полужирный**;
- семейство шрифта: Times New Roman, Arial и т.п.;
- стиль: обычный или *наклонный*;
- цвет текста;
- цвет фона, на котором расположен текст;

2) тег блока (h1-6, li, dir и т.д.);
3) позиция на странице и ширина блока;
4) информация о структуре DOM-дерева:
- количество потомков узла;
- изменение визуальных характеристик по отношению к родитель-

скому, дочернему и корневому узлу;

5) лингвистическая информация:

- длина текста;
- количество буквенных символов;
- наличие позитивных и негативных слов.

В качестве модели машинного обучения предлагается перцептрон с нечеткими отступами (Perceptron with Uneven Margins) [7]. Эта версия перцептрона может работать особенно хорошо, когда число позитивных и негативных объектов в обучающей выборке сильно несбалансированно. Вышеперечисленные признаки бинаризируются, в итоге классификатор обрабатывает 245 признаков для каждого объекта. Работа [6] доказывает, что ошибка перцептрона растет в линейной зависимости от количества используемых признаков, что ставит под сомнение предлагаемый подход к формированию признакового пространства. Такие стилевые признаки, как семейство и стиль шрифта мешают обобщению данных моделью, ввиду того, что сайты фотобанков могут по-разному оформлять текст заголовка. Также признак наличия определенных слов в тексте делает решение языкозависимым: в данном случае предполагается, что модель работает только с англоязычными заголовками.

Работа [17] является попыткой усовершенствовать подход [5]. Помимо признаков, перечисленных в ранее, данный подход вводит новые:

1) информация о странице:

- тип макета страницы (особенность расположения групп блоков);
- высота и ширина страницы;
- позиция верхнего левого угла страницы;

2) информация о группе, к которой принадлежит блок:

- тип группы (верхняя группа, боковая группа, основная группа);
- высота и ширина группы блоков;

- отступ от верхнего левого угла страницы;

3) дополнительная информация о html-объекте:

- расположение блока (вверху, внизу, справа, слева);
- высота блока;
- отступ от верхнего левого угла страницы.

В качестве алгоритма машинного обучения используется метод опорных векторов (Support Vector Machine).

Работа [4] ставит перед собой задачу извлечения заголовка из сервис-ориентированных веб-страниц (например, сайты ресторанов, аптек, автосервисов и т.п.), что делает ее наиболее близкой к тематике данной ВКР. Работа рассматривает все предыдущие решения проблемы извлечения заголовков, заимствует все признаки, пригодные для идентификации заголовка и предлагает новый: определение частей речи в тексте (Part of Speech tagging). Этот признак является языкозависимым: алгоритмы автоматической разметки частей речи предварительно обучаются на словаре одного языка. На данный момент нет эффективного и универсального алгоритма разметки частей речи для всех естественных языков одновременно. Тем не менее, из работы можно почерпнуть признаки, не рассмотренные ранее:

1) количество вхождений текста на всей странице;
2) капитализация: принимает значение 1, если текст начинается с заглавной буквы, иначе – 0;
3) количество слов в тексте;
4) коэффициент вхождения текста в URL.

Для классификации заголовка используется метод опорных векторов.

Вывод

В процессе обзора существующих решений были перечислены признаки, которые использовались для идентификации заголовков. Большая часть из них не противоречит специфике сайтов фотобанков и может быть использована в решении поставленной задачи. Вместе с этим, ни одно ре-

шение не может быть использовано чистом виде, следовательно, разработка решения задачи извлечения заголовков с веб-страниц фотобанков актуальна.

2. НАСТРОЙКА КЛАССИФИКАТОРА

2.1. Признаковое пространство

Система координат, каждое измерение которой образовано определенным признаком (атрибутом) объекта или наблюдения, а по осям откладываются значения признаков (атрибутов) называется признаковым пространством. Тогда каждый объект или наблюдение могут быть представлены точкой в многомерном пространстве, положение которой будет определяться набором значений его признаков. Каждая такая точка называется многомерным вектором.

Понятие пространства признаков играет очень большую роль в аналитических методах, поскольку многие алгоритмы классификации и кластеризации оперируют именно координатами объектов и наблюдений в многомерном пространстве и расстояниями между ними. Например, степень похожести объектов, а, следовательно, и вероятность их принадлежности к одному классу, может быть определена на основе расстояния между их точками в пространстве признаков. Чем меньше расстояние между векторами признаков, тем более похожи друг на друга соответствующие объекты.

В процессе составления обучающей выборки были сделаны следующие наблюдения:

- заголовок изображения зачастую находится рядом с самим изображением;
- продаваемое изображение всегда имеет наибольший размер среди всех других изображений на странице;
- зачастую, в html-атрибутах class, id, itemprop объекта, который содержит заголовок, присутствуют специфические слова (title, name, desc и т.д.).

Дополнительно к параметрам, перечисленным в обзоре существующих решений (раздел 1.2), сформированы следующие признаки:

- max_img_distance – расстояние от блока до максимального изображения на странице;
- class_<regexp>, id_<regexp>, itemprop_<regexp> – наличие специфического слова или паттерна <regexp> в перечисленных атрибутах.

В результате составлен начальный список атрибутов (приложение 1).

2.2. Обучающая выборка

Обучающая выборка составлена из текстовых узлов веб-страниц. С каждого фотобанка загружено 10 случайных страниц. Всего рассмотрено 99 фотобанков. В среднем, на каждой скачанной странице из выборки присутствует 92 объекта, который содержит текст. Полная выборка содержит 91737 объектов.

На некоторых фотобанках верстка веб-страницы такова, что заголовок может присутствовать в нескольких текстовых объектов. Также вторичные заголовки могут присутствовать, например, в описании изображения, где могут быть несколько видоизменены: допустим действительный заголовок изображения – «Яблоко на столе», а описание изображения «Зеленое яблоко на столе из дуба».

Поскольку в качестве правильного результата работы программы нам подойдет любой текст так или иначе отражающий содержание изображения, промаркируем объекты следующим образом: для каждой веб-страницы фотобанка вручную выбирается текст заголовка и принимается за эталонный, далее с помощью программы выполняется определение вещественных меток для каждого объекта. Метка объекта является отношением длины максимальной общей подпоследовательности символов между эталонным заголовком и текстом объекта к длине эталонного заголовка. Таким образом, каждая метка объекта является вещественным числом от 0 до 1 и отражает степень присутствия эталонного заголовка в тексте объекта. Сам объект, содержащий в себе эталонный заголовок всегда будет иметь метку равную единице.

Исходя из наблюдения, что каждая веб-страница одного и того же фотобанка содержит повторяющиеся текстовые элементы, которые не относятся к содержательной части (например, контактные данные, меню сайта, список доступных языков отображения веб-страницы и т.п.), было принято решение удалить дубли таких объектов – это должно повысить качество и скорость обучения модели.

Таким образом:

- очищенная обучающая выборка содержит 21891 объект;
- метки объектов принимают вещественные значения от 0 до 1;
- метки объектов отражают степень присутствия реального заголовка изображения в тексте объектов;
- на один объект положительного класса (метка равна 1), в среднем приходится 24 объекта негативного класса (метка равна 0) – обучающая выборка сильно не сбалансирована.

2.3. Модель машинного обучения

В данной работе сделан выбор в пользу метамодели машинного обучения – градиентного бустинга деревьев решений.

Дерево принятия решений

Дерево принятия решений – это средство поддержки принятия решений при прогнозировании, широко применяется в статистике и анализе данных.

Дерево состоит из узлов ветвления, ребер и листьев. Построение узлов ветвления происходит путем определения признака, по которому можно наилучшим образом разделить выборку на данном этапе. В данной работе деревья используют меру однородности Джини (Gini impurity) для определения признака. Мера Джини однородности вершины принимает нулевое значение, когда в данной вершине имеется всего один класс (если используются априорные вероятности, оцененные по размерам классов или исходя из одинаковой цены ошибок классификации, то мера Джини

вычисляется как сумма всех попарных произведений относительных размеров классов, представленных в данной вершине; ее значение будет максимальным, когда размеры всех классов одинаковы). В листьях дерева содержится значение целевой функции; по ребрам совершаются переходы от одного узла к другому.

Таким образом, классификация с помощью дерева принятия решений происходит поэтапно: начиная с корня дерева, каждый узел дерева производит сравнение одного из признаков объекта с разделяющим значением, затем, в зависимости от результата сравнения, происходит переход в следующий узел, пока не будет достигнут один из листьев дерева, в котором содержится решение классификатора.

Бустинг

Бустинг – это процедура последовательного построения композиции алгоритмов машинного обучения, когда каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих алгоритмов.

В течение последних 10 лет бустинг остается одним из наиболее популярных методов машинного обучения, наряду с нейронными сетями и машинами опорных векторов. Бустинг над решающими деревьями считается одним из наиболее эффективных методов с точки зрения качества классификации. Во многих экспериментах наблюдалось практически неограниченное уменьшение частоты ошибок на независимой тестовой выборке по мере наращивания композиции.

Градиентный бустинг решающих деревьев

Градиентный бустинг деревьев решений – модель машинного обучения, представляющая бустинг как процесс градиентного спуска. В основе алгоритма лежит последовательное уточнение функции, представляющей собой линейную комбинацию базовых классификаторов (деревьев решений), с тем чтобы минимизировать функцию потерь.

Пусть $\mathcal{F}$ – множество базовых классификаторов, а $lin(\mathcal{F})$ – множество всех линейных комбинаций из F. На каждом шаге алгоритма к текущему классификатору $F \in lin(\mathcal{F})$ прибавляется базовый классификатор так, чтобы значение $C(F + \varepsilon f)$ уменьшилось на некоторое значение ε. То есть в терминах функционального пространства для функции f ищется направление, в котором функция $C(F + \varepsilon f)$быстрее уменьшается. Наибольшее уменьшение функции потерь наблюдается в случае, когда f максимизирует величину $-\langle \nabla C(F), f \rangle$.

Алгоритм построения:

1) инициализация $F_0 = 0$;
2) для всех $t = 0, \dots, T$ пока не выполнено условие выхода из цикла:

- получение нового классификатора f_{t+1}, увеличивающего значение $-\langle \nabla C(F), f_{t+1} \rangle$;
- если $-\langle \nabla C(F), f_{t+1} \rangle \leq 0$ выходим из цикла и возвращаем F_t;
- выбор веса w_{t+1};
- уточнение классификатора $F_{t+1} = F_t + w_t f_{t+1}$;

3) возвращаем F_{T+1}.

В случае бинарного классификатора $Y = \{-1; 1\}$. Пусть $X^l = \{(x_i, y_i)\}$ – обучающая выборка. Функция потерь $C = \frac{1}{m} \sum_{i=1}^{m} c(y_i F(x_i))$ определяется через дифференцируемую функцию выброса $c: \mathbb{R} \to \mathbb{R}$. В этом случае $-\langle \nabla C(F), f \rangle = \frac{1}{m^2} \sum_{i=1}^{m} y_i f(x_i) c'(y_i F(x_i))$, и нахождение классификатора на каждом шаге будет равносильно нахождению классификатора f, минимизирующего взвешенную ошибку.

2.4. Отбор признаков

Ансамбли деревьев решений помогают вычислить вклад каждого признака в разделение обучающей выборки на два класса. Значение вклада называется средним снижением неоднородности (mean decrease impurity) и вычисляется как среднее значение однородности Джини данного признака

для каждого дерева в ансамбле [8]. Таким образом, можно сформировать процентное отношение важности признака относительно всех остальных (приложение 2). Значения получены путем анализа ансамбля из тысячи деревьев решений с помощью встроенных инструментов библиотеки XGBoost [16].

Признаки имеющие нулевой вклад в разделение обучающей выборки были исключены из обучающей выборки. Таким образом, алгоритм классификации использует следующий список из 47 признаков: *top, similarity_with_url, max_img_distance, sum_dice, text_density, left, sum_LCSub, width, font-size, avg_letters_per_word, title_distance_to_pcenter, letters_percent, title_distance_to_parea, uppercase_letters_percent, sum_LCStr, max_dice, height, max_LCStr, path_length, max_LCSub, path_last_elem_h1, text_length, childs_count, itemprop_name, class_title, path_last_elem_li, commas_per_words_percent, class_desc, path_last_elem_div, path_last_elem_a, path_last_elem_dd, class_h, path_last_elem_p, path_last_elem_h2, class_link, start_with_uppercase, path_last_elem_span, dots_per_words_percent, class_text, id_name, class_head, path_last_elem_td, path_last_elem_h4, path_last_elem_center, itemprop_caption, path_last_elem_h3, class_name*.

2.5. Метод оценки качества модели машинного обучения

Кросс-валидация – метод оценки аналитической модели и ее поведения на независимых данных. При оценке модели, имеющиеся в наличии данные разбиваются на k частей. Затем на k−1 частях данных производится обучение модели, а оставшаяся часть данных используется для тестирования. Процедура повторяется k раз; в итоге каждая из k частей данных используется для тестирования. В результате получается оценка эффективности выбранной модели с наиболее равномерным использованием имеющихся данных.

Веб-страница фотобанка, в среднем, содержит 92 текстовых объекта, среди которых находится от одного до шести потенциальных заголовков

изображения. Поскольку программа должна выбрать только один объект, который подходит на роль заголовка больше всего, не имеет значение какую оценку получили все остальные. Следовательно, модель машинного обучения должна приближать оценки действительных заголовков к единице и всех остальных объектов – к нулю, что подразумевает решение задачи классификации. С другой стороны, важно, чтобы алгоритм присвоил наибольшую оценку только действительному заголовку, что подразумевает решение задачи ранжирования.

Совокупное качество работы модели состоит из качества классификации, которое отражает уверенность модели в оценке отдельно взятого объекта, и качества ранжирования, которое отражает способность модели присваивать действительным заголовкам наивысшую оценку среди всех объектов веб-страницы.

Чтобы одновременно оценить качество классификации и ранжирования были выбраны следующие метрики.

AUC PR (Area under the precision-recall curve) – метрика оценки качества классификации, которая комбинирует значения Precision (точность) и Recall (полнота). Применяется в задачах классификации, где:

1) доли классов вы обучающей выборке сильно не сбалансированы;
2) важнее классифицировать позитивный класс, нежели негативный.

Приведем основные определения:

- TP (True positive) – количество верно определенных объектов положительного класса;
- FP (False positive) – количество негативных объектов, определенных как положительные;
- FN (False negative) – количество положительных объектов, определенных как негативные.

Таким образом: $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$, $\text{Recall} = \frac{TP}{\text{TP+FN}}$.

Поскольку для определения Precision и Recall требуются бинарные метки, а не вещественные, необходимо найти некоторое пороговое значение, с помощью которого бинаризируются вещественные метки. Метрика AUC PR призвана решить эту проблему, благодаря тому, что кривая Precision-Recall строится для каждого уникального (среди всех объектов тестовой выборки) порогового значения от 0 до 1.

На рис. 1 приведен пример кривой PR для двух случайных алгоритмов.

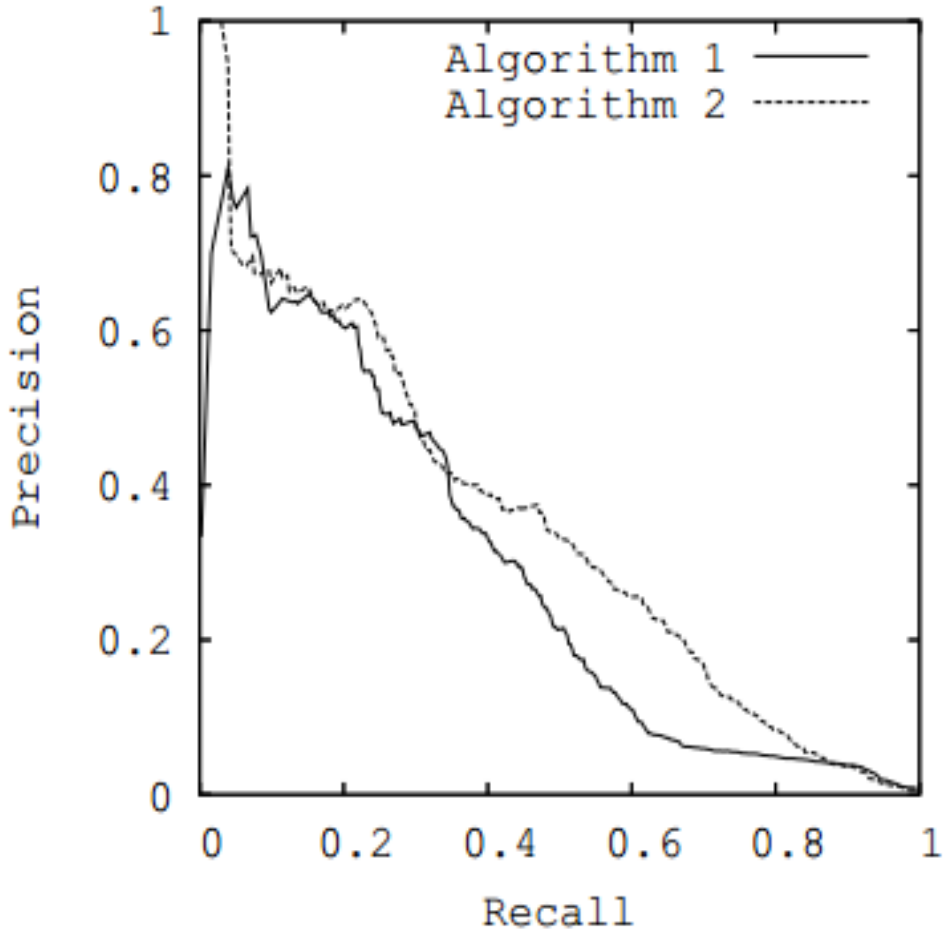


Рис. 1. Пример PR кривой

Площадь под PR кривой принимает вещественное значение от 0 до 1, где 1 означает идеальную классификацию объектов.

nDCG@p (Normalized discounted cumulative gain at p) – мера измерения качества ранжирования p первых объектов. Представляет собой веще-

ственное значение от 0 до 1, где 1 означает идеальное ранжирование объектов. Обычно эта метрика используется для оценки эффективности поисковых систем.

Если DCG@p (Discounted cumulative gain at p) представляется как

$$\mathrm{DCG_p}=\sum_{i=1}^{p}\frac{2^{rel_i}-1}{\log_2(i+1)},$$

где: rel_i представляет вещественное значение от 0 до 1 – релевантность документа на позиции i, то nDCG@p имеет вид:

$$\mathrm{nDCG_p}=\frac{\mathrm{DCG_p}}{\mathrm{IDCG_p}},$$

где: IDCG@p – это максимальное (I – ideal) значение DCG@p:

$$\mathrm{IDCG_p}=\sum_{i=1}^{|REL|}\frac{2^{rel_i}-1}{\log_2(i+1)},$$

REL представляет список элементов с идеальным расположением до позиции p.

Поскольку, нам не важно, как именно проранжированы объекты отрицательного класса (не заголовки), необходимо выбрать параметр p, который позволяет оценивать ранжирование первых p объектов (объектов с наибольшей оценкой). Исходя из наблюдений, которые были сделаны при сборе обучающей выборки (раздел 2.2), на веб-страницах фотобанков может находиться от 1 до 6 кандидатов на роль заголовка изображения – примем параметр p равным среднему количеству возможных заголовков на веб-странице. Таким образом, в данном тестировании используется метрика nDCG@3.

2.6. Оптимизация параметров модели машинного обучения

Для оптимизации параметров модели используем метод полного перебора. Перебор всех возможных значений параметров является трудоемкой вычислительной задачей, поэтому для каждого параметра установим

диапазон значений и шаг. В таблице 1 приведен полный список параметров модели градиентного бустинга решающих деревьев.

Табл. 1. Параметры модели

Параметр	**Краткое описание**	**Диапазон значений**	**Шаг**
n_iter	Количество деревьев	100 … 1000	100
eta	Шаг градиентного спуска	0.1 … 0.5	0.1
gamma	Минимальное снижение функции потерь, допускающее создание нового узла дерева.	0 … 1	0.1
max_depth	Максимальная глубина дерева.	2 … 24	2
max_delta_step	Параметр, нивелирующий влияние несбалансированности обучающей выборки на качество классификации.	1 … 10	1
subsample	Регулирует количество объектов выборки, которое используется для обучения нового дерева.	0.5 … 1	0.1
colsample_bytree	Регулирует количество признаков доступных для построения нового дерева.	0.5 … 1	0.1

Также алгоритм имеет параметр scale_pos_weight, регулирующий веса положительного класса, что позволяет нивелировать влияние несбалансированной выборки. Параметр scale_pos_weight равен отношению коли-

чества негативных экземпляров обучающей выборки к количеству позитивных и равен 24.2423. Таким образом, рассматриваем 1 500 000 комбинаций параметров алгоритма.

Оптимизация параметров модели проводилась в течении 38 часов непрерывного времени на процессоре Core i7 3630QM:

1) частота процессора: 2600 МГц;
2) количество ядер процессора: 4;
3) количество потоков на ядро: 2.

В результате поиска оптимальной конфигурации были выбраны следующие параметры:

- n_iter: 700;
- eta: 0.1;
- gamma: 1;
- max_depth: 12;
- max_delta_step: 1;
- subsample: 0.8;
- colsample_bytree: 0.8;
- scale_pos_weight: 24.2423.

При данном списке параметров модель градиентного бустинга решающих деревьев имеет следующие результаты кросс-валидации (табл. 2).

Табл. 2. Оценка качества модели с полученными параметрами

Метрика	**К-во на трен. выборке**	**К-тво на тест. выборке**
AUC PR	0.936	0.734
nDCG@3	0.975	0.862

3. РЕАЛИЗАЦИЯ АГРЕГАТОРА

3.1. Функциональные требования

Перечислим функциональные требования к программе:

1) агрегатор должен осуществлять извлечение релевантных признаков, которые были определены в разделе 2.4, из текстовых объектов веб-страниц;

2) агрегатор должен использовать предобученную модель машинного обучения, для оценки степени принадлежности объектов веб-страницы к заголовку;

3) агрегатор должен принимать на вход URL адрес веб-страницы;

4) агрегатор должен возвращать текст, содержащийся в объекте веб-страницы, который получил наивысшую оценку на этапе классификации.

3.2. Модульная структура

Модульная структура схематично изображена на рис. 2.

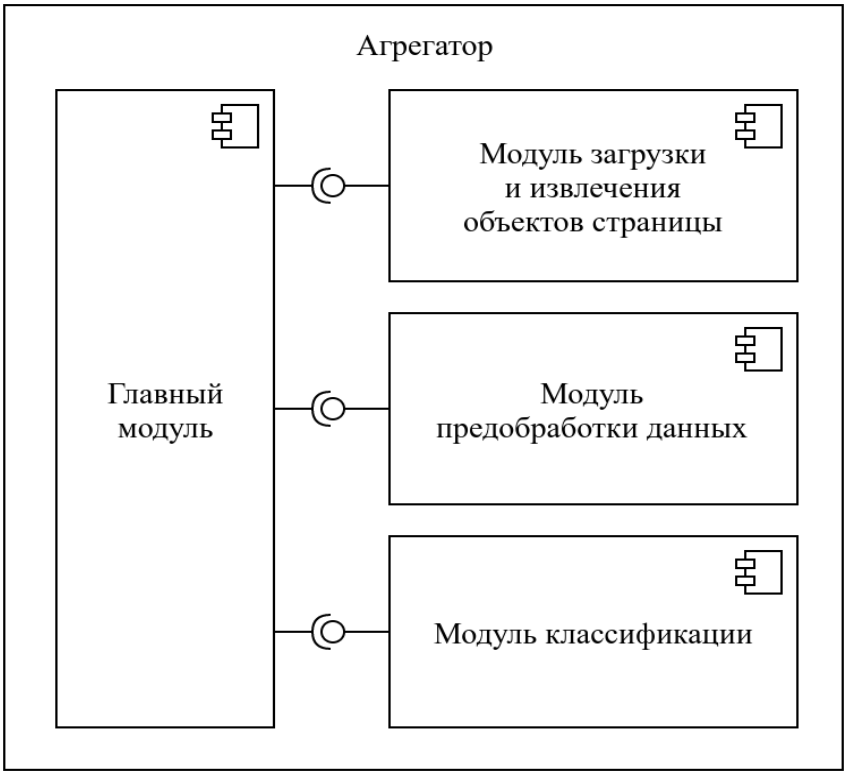


Рис. 2. Диаграмма компонентов

Модуль загрузки и извлечения объектов страницы реализует загрузку и получение необходимых данных с веб-страницы. Задача данного модуля заключается в извлечении информации о каждом узле DOM-дерева веб-страницы, который содержит какой-либо видимый текст.

Модуль предобработки данных предназначен для извлечения релевантных признаков из текстовых объектов веб-страниц;

Модуль классификации предназначен для оценки степени принадлежности объектов веб-страницы к заголовку.

Главный модуль объединяет работу всех модулей, осуществляет контроль передачи данных между модулями. Данный модуль ответственен обработку входящих (URL адрес веб-страницы) и исходящих данных (текст заголовка).

Далее представлена диаграмма потока данных, которая отображает структуру передачи данных между модулями (рис. 3).

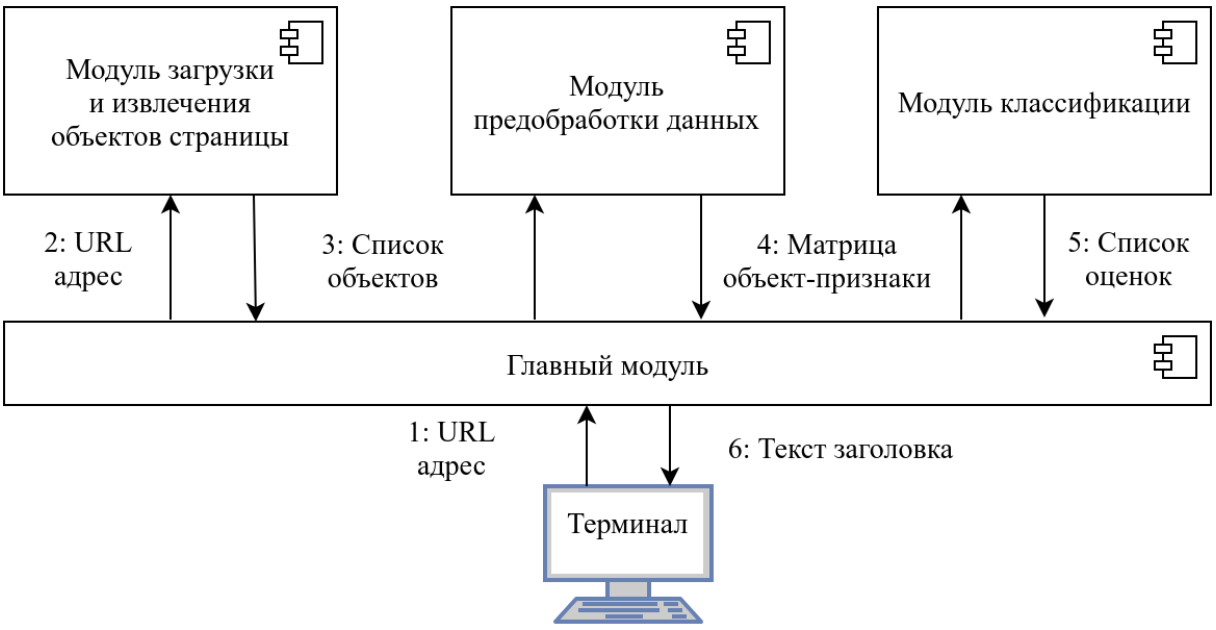


Рис. 3. Диаграмма потока данных

3.3. Выбор технологий реализации

В качестве основного языка программирования выбран Python версии 3.4.

Динамические веб-страницы не передают весь html-код по запросу: часть кода генерируется на стороне клиента с помощью встроенных в страницу скриптов. Чтобы обеспечить корректную загрузку динамических и статических веб-страниц была выбрана связка библиотеки Selenium WebDriver v3.4.2 [13] и браузера PhantomJS v2.1 [11], который предназначен для тестирования веб-страниц.

В качестве реализации градиентного бустинга деревьев решений была выбрана библиотека XGBoost [16] (версия 0.6) – библиотека с открытым исходным кодом, которая предоставляет фреймворк градиентного бустинга для многих языков программирования.

Далее перечислены прочие сторонние библиотеки python, используемые в разработке:

1) pandas [10], версия 0.20.1;
2) scikit-learn [12], версия 0.18.1;
3) numpy [9], версия 1.13.0rc2.

3.4. Реализация модулей

Реализация программы выполнена в императивном стиле программирования. Модули программы реализованы в виде функций языка Python.

Модуль загрузки и извлечения объектов страницы

Данный модуль принимает на вход URL адрес веб страницы, затем производит загрузку веб-страницы с помощью связки Selenium + PhantomJS. После этого в код веб-страницы внедряется специальный скрипт, написанный на языке JavaScript, который осуществляет обход и сбор данных о всех видимых объектах веб-страницы, которые содержат текст. В процессе работы данного модуля формируется массив, состоящий из JSON-объектов, каждый из которых содержит следующие данные:

- similarity_with_url – мера сходства Соренсена-Дайса [2] между URL адресом веб-страницы и текстом объекта;
- bound – визуальные характеристики (height, width, top, left,

max_img_distance);

- computed – JSON-объект, хранящий CSS свойства объекта, а также значения атрибутов class, id, name;
- path – список тегов составляющие путь от корня документа до рассматриваемого объекта;
- text – текст объекта.

Модуль предобработки данных

Данный модуль принимает на вход массив JSON-объектов, полученный в результате работы модуля загрузки и извлечения объектов страницы. Модуль вычисляет недостающие признаки из списка (раздел 2.5), формирует матрицу вида объект-признаки. Признаки (столбцы матрицы) располагаются в таком же порядке, как и при обучении модели.

Модуль классификации

Модуль принимает на вход матрицу вида объект-признаки, загружает предобученную модель и производит с помощью нее оценку объектов. Результатом работы модуля является массив целочисленных значений – оценок классификатора.

Главный модуль

Главный модуль получает URL адрес веб-страницы как аргумент командной строки, и, затем, передает ее в модуль загрузки и извлечения объектов страницы. Модуль организует вызов модулей, которые реализованы в виде функций, корректную передачу данных между ними. После получения оценок классификатора модуль сопоставляет предсказания с текстами объектов и возвращает текст с наибольшей оценкой в стандартный поток вывода. Важным условием в процессе работы всей программы является сохранение порядка в данных: первому объекту, который был извлечен из веб-страницы должен соответствовать первый вектор признаков в матрице объект-признак и т.д.

4. ТЕСТИРОВАНИЕ

4.1. Тестирование алгоритма машинного обучения

Алгоритм машинного обучения был получен в результате обучения модели градиентного бустинга деревьев решений с использованием оптимальной конфигурации параметров (раздел 2.6).

В целях тестирования, обучающая выборка (21891 объект) была разделена на две части:

- данные для обучения модели – 70% (15324 объекта);
- данные для тестирования – 30% (6567 объектов).

Результаты тестирования качества алгоритма (табл. 3) показали, что алгоритм осуществляет идеальное качество ранжирования, что гарантирует наличие у действительного заголовка изображения наивысшей оценки среди всех объектов веб-страницы, при этом качество классификации отдельно взятого объекта недостаточно высоко для использования алгоритма в задаче поиска всех кандидатов на роль заголовка на веб-странице (обычно, для внедрения алгоритмов машинного обучения в бизнес-решения устанавливают допустимый порог качества от 0.9-0.95).

Табл. 3. Результаты тестирования алгоритма

Метрика	**К-во на трен. выборке**	**К-во на тест. выборке**
AUC PR	0.988	0.789
nDCG@3	1.0	1.0

4.2. Тестирование агрегатора

Для тестирования агрегатора на реальных данных использовался системный метод.

Системное тестирование – это тестирование программного обеспечения (ПО), выполняемое на полной, интегрированной системе, с целью

проверки соответствия системы исходным требованиям. Системное тестирование относится к методам тестирования черного ящика, и, тем самым, не требует знаний о внутреннем устройстве системы.

Далее представлен скриншот тестовой веб-страницы, расположенной по адресу www.shutterstock.com/ru/image-photo/beautiful-retro-luxury-light-lamp-decor-28760680 (рис. 4).

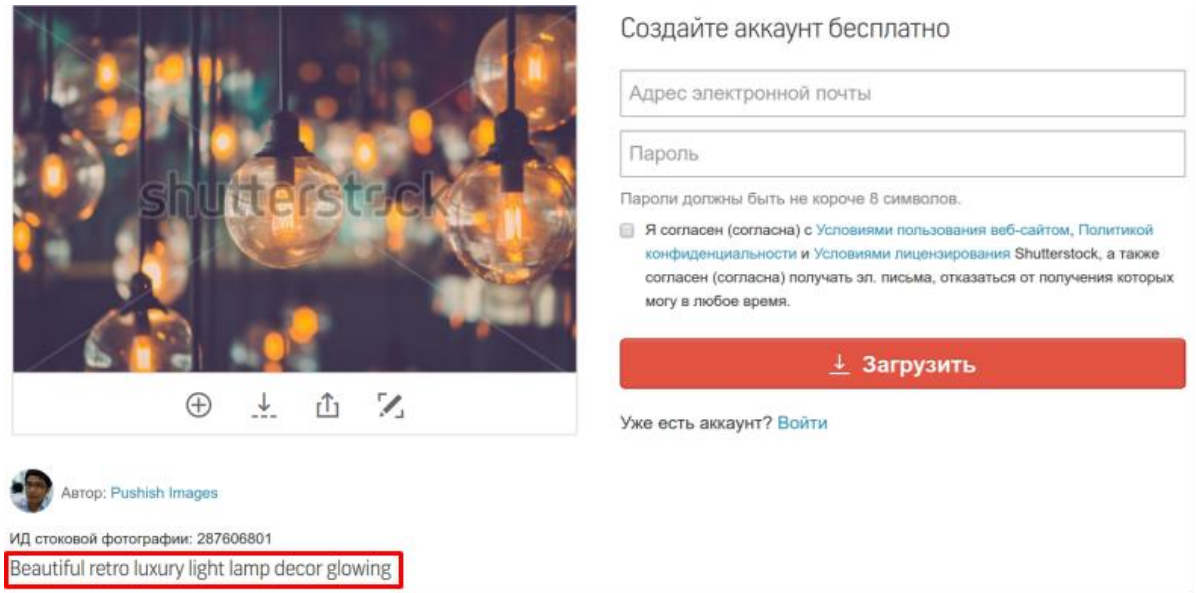


Рис. 4. Тестируемая страница (заголовок «Beautiful retro luxury light lamp decor glowing» выделен красным, дата обращения 25.05.17)

Результаты работы программы при включении сайта shutterstock.com в обучающую выборку и без него представлены в приложении 3. В соответствии с функциональными требованиями программа должна вывести только текст одного html-объекта с максимальной оценкой, но в целях тестирования программа использует расширенный вывод. В данном случае программа выводит 20 наиболее вероятных строк на роль заголовка и соответствующую оценку алгоритма в порядке убывания.

Алгоритм присвоил максимальное значение текстовому блоку, который содержал заголовок изображения в обоих случаях – алгоритм машинного обучения продемонстрировал способность верно ранжировать объек-

ты веб-страницы вне зависимости от присутствия данных о сайте фотобанка в обучающей выборке. Программа получила на вход URL адрес тестовой веб-страницы и использовала алгоритм машинного обучения для оценки текстовых объектов веб-страниц. Разработанный программный продукт работает в соответствии с функциональными требованиями.

ЗАКЛЮЧЕНИЕ

Данная работа посвящена разработке агрегатора заголовков изображений с веб-страниц фотобанков.

Основные результаты работы

В ходе выполнение работы были получены следующие основные результаты:

1) рассмотрены известные подходы к сбору и анализу данных с веб-страниц;
2) определено признаковое пространство;
3) осуществлен выбор и обучение модели машинного обучения;
4) разработан агрегатор;
5) проведено тестирование качества алгоритма машинного обучения;
6) проведено тестирование агрегатора.

Основные направления дальнейшей работы

Работа по представленной теме может быть продолжена в следующих направлениях:

улучшение качества классификации заголовков;

расширение набора извлекаемой информации (цена изображения, ключевые слова, доступные форматы изображения и т.д.).

ЛИТЕРАТУРА

1. Chiou L., Tucker C. Content Aggregation by Platforms: The Case of the News Media. [Электронный ресурс] URL: http://dx.doi.org/10.3386/w21404 (дата обращения: 25.05.2017).

2. Dice L.R. Measures of the Amount of Ecologic Association Between Species. [Электронный ресурс] URL: http://biocomparison.ucoz.ru/_ld/0/86_dice_1945.pdf (дата обращения: 25.05.2017).

3. Fan J., Luo P., Joshi P. Title identification of web article pages using HTML and visual features. [Электронный ресурс] URL: http://dx.doi.org/10.1117/12.876708 (дата обращения: 25.05.2017).

4. Gali N., Mariescu-Istodor R., Franti P. Using linguistic features to automatically extract web page title. [Электронный ресурс] URL: http://dx.doi.org/10.1016/j.eswa.2017.02.045 (дата обращения: 25.05.2017).

5. Hu Y. Title extraction from bodies of HTML documents and its application to web page retrieval / Y. Hu, G. Xin, R. Song, G. Hu, S. Shi, et al. [Электронный ресурс] URL: http://dx.doi.org/10.1145/1076034.1076079 (дата обращения: 25.05.2017).

6. Kivinen J., Warmuth M.K. The Curse of Dimensionality and the Perceptron Algorithm. [Электронный ресурс] URL: https://www.researchgate.net/publication/2316104_The_Curse_of_Dimensionality_and_the_Perceptron_Algorithm (дата обращения: 25.05.2017).

7. Li Y. The perceptron algorithm with uneven margins / Y. Li, H. Zaragoza, R. Herbrich, J. Shawe-Taylor, J. Kandola [Электронный ресурс] URL: http://www.academia.edu/download/45980284/The_Perceptron_Algorithm_with_Uneven_Mar20160526-9690-tnny47.pdf (дата обращения: 25.05.2017).

8. Louppe G., Wehenkel L., Suntera A., Geurts P. Understanding variable importances in forests of randomized trees. [Электронный ресурс] URL:

http://papers.nips.cc/paper/4928-understanding-variable-importances-in-forests-of-randomized-trees.pdf (дата обращения: 25.05.2017).

9. Numpy 1.13.0rc2: Python Package Index. [Электронный ресурс] URL: https://pypi.python.org/pypi/numpy (дата обращения: 25.05.2017).

10. Pandas 0.20.1: Python Package Index. [Электронный ресурс] URL: https://pypi.python.org/pypi/pandas (дата обращения: 25.05.2017).

11. PhantomJS: Headless WebKit scriptable with a JavaScript API. [Электронный ресурс] URL: http://phantomjs.org/ (дата обращения: 25.05.2017).

12. Scikit-learn 0.18.1: Python Package Index. [Электронный ресурс] URL: https://pypi.python.org/pypi/scikit-learn/0.18.1 (дата обращения: 25.05.2017).

13. Selenium 3.4.2: Python Package Index. [Электронный ресурс] URL: https://pypi.python.org/pypi/selenium (дата обращения: 25.05.2017).

14. Slickdeals. [Электронный ресурс] URL: http://www.slickdeals.net (дата обращения: 25.05.2017).

15. The Stocks. [Электронный ресурс] URL: http://thestocks.im (дата обращения: 25.05.2017).

16. XGBoost: eXtreme Gradient Boosting library. [Электронный ресурс] URL: https://github.com/dmlc/xgboost (дата обращения: 25.05.2017).

17. Xue Y., Hu Y., Xin G., Li H. Web page title extraction and its application. [Электронный ресурс] URL: http://dx.doi.org/10.1016/j.ipm.2006.11.007 (дата обращения: 25.05.2017).

18. Zhou Z., Mashuq M. Web Content Extraction Through Machine Learning. [Электронный ресурс] URL: https://github.com/ziyan/spider/blob/master/docs/final/final.pdf (дата обращения: 25.05.2017).

19. Яндекс.Каршеринг. [Электронный ресурс]. URL: https://yandex.ru/carsharing (дата обращения: 25.05.2017).

20. Яндекс.Маркет. [Электронный ресурс]. URL: https://market.yandex.ru/ (дата обращения: 25.05.2017).

ПРИЛОЖЕНИЯ

Приложение 1

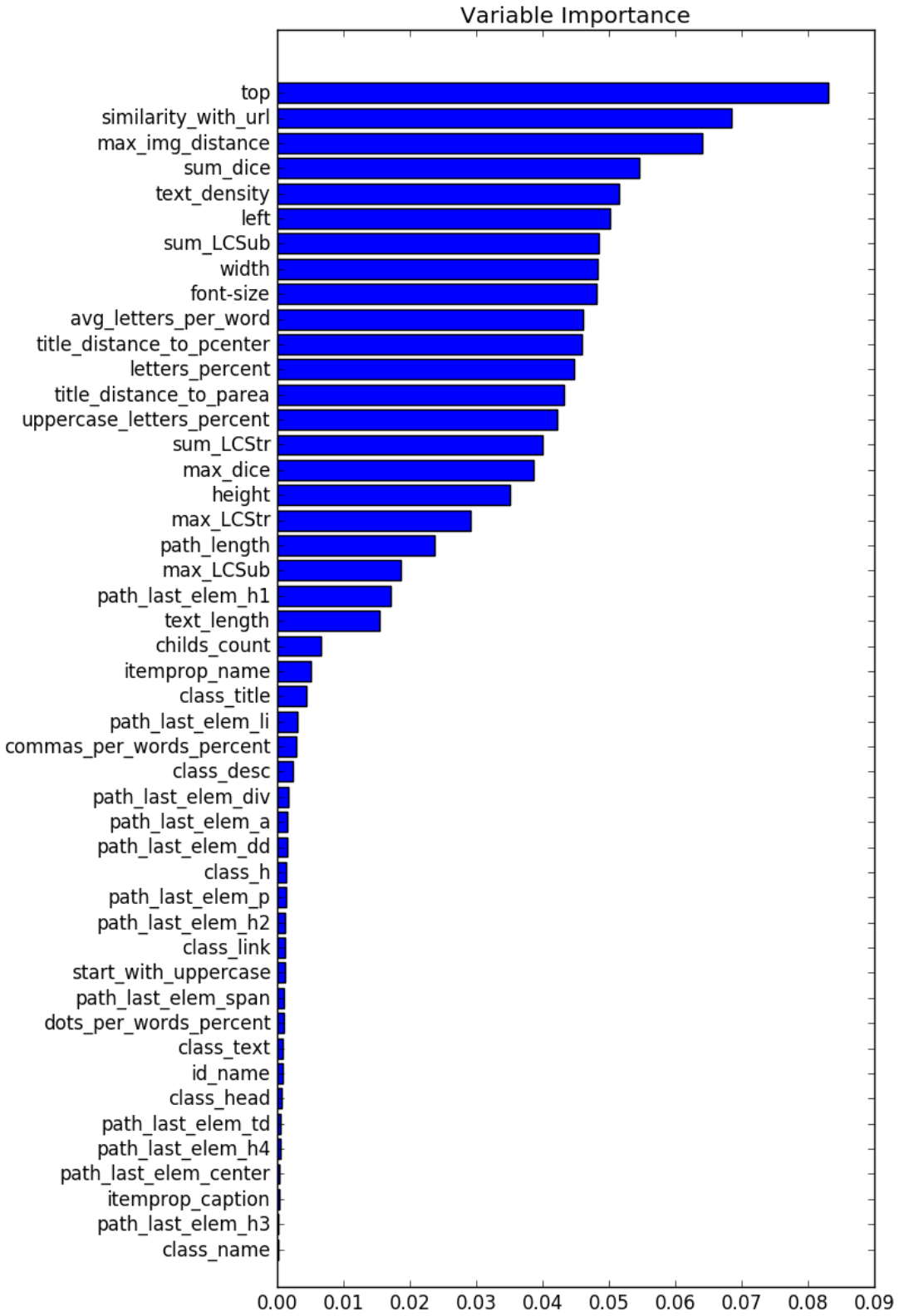
Первичный список признаков.

Название признака	**Описание**
similarity_with_url	Коэффициент Соренсена-Дайса между URL и текстом объекта
childs_count	Количество потомков узла
font-size	Размер шрифта
class_button	Присутствие слова button или btn в атр. class
class_desc	Присутствие слова desc в атр. class
class_h	Присутствие паттерна h[1-6] в атр. class
class_head	Присутствие слова head в атр. class
class_icon	Присутствие слова icon или ico в атр. class
class_info	Присутствие слова info в атр. class
class_keywords	Присутствие слова keyword или kw в атр. class
class_link	Присутствие слова link или lnk в атр. class
class_name	Присутствие слова name в атр. class
class_nav	Присутствие слова nav в атр. class
class_text	Присутствие слова text в атр. class
class_title	Присутствие слова title в атр. class
id_button	Присутствие слова button или btn в атр. id
id_desc	Присутствие слова desc в атр. id
id_keywords	Присутствие слова keyword или kw в атр. id
id_link	Присутствие слова link или lnk в атр. id
id_name	Присутствие слова name в атр. id
id_nav	Присутствие слова nav в атр. id
itemprop_author	Присутствие слова author в атр. itemprop
itemprop_caption	Присутствие слова caption в атр. itemprop
itemprop_desc	Присутствие слова desc в атр. itemprop
itemprop_name	Присутствие слова name в атр. itemprop
itemprop_price	Присутствие слова price в атр. itemprop

itemprop_url	Присутствие слова url в атр. itemprop
height	Высота текстового блока
left	Отступ т. блока от левого края веб-страницы
max_img_distance	Расстояние текстового блока до наибольшего изображения на веб-странице
title_distance_to_parea	Расстояние площади блока до усредненной площади всех блоков-заголовков из обучающей выборки
title_distance_to_pcenter	Расстояние центра блока до усредненного центра всех блоков-заголовков из обучающей выборки
top	Отступ т. блока от верхнего края веб-страницы
width	Ширина текстового блока
path_last_elem	Последний тэг в пути DOM-дерева до блока
path_length	Длина пути DOM-дерева от корня до блока
avg_letters_per_word	Среднее количество букв в слове
commas_per_words	Количество запятых на количество слов
dots_per_words	Количество точек на количество слов
letters_percent	Отношение буквенных символов ко всей длине текста
max_LCStr	Максимальная длина совпадающей подстроки для всех остальных текстовых блоков на странице
max_LCSub	Максимальная длина совпадающей подпоследовательности для всех остальных текстовых блоков на странице
max_dice	Максимальный коэф. Соренсена-Дайса для всех остальных текстовых блоков на странице
start_with_uppercase	Текст начинается с заглавной буквы
sum_LCStr	Сумма всех длин совпадающей подстроки для всех остальных текстовых блоков на странице
sum_LCSub	Сумма всех длин совпадающей подпоследовательности для всех остальных текстовых блоков на странице
sum_dice	Сумма всех коэф. Соренсена-Дайса для всех остальных текстовых блоков на странице
text_density	Отношение Длины текста к площади блока
text_length	Длина текста
uppercase_letters	Количество заглавных букв ко всей длине текста

Приложение 2

Влияние признаков на качество классификации. На изображении присутствуют все признаки из первичного набора с ненулевым влиянием.



Приложение 3

Результаты работы агрегатора во время тестирования.

1) Результат работы программы при включении тестируемого сайта (shutterstock.com) в обучающую выборку.

```
mark@mark-N76VB ~/Workplace/PressFoto/graduation_project/prototypes/0.2 $
python3 prototype.py https://www.shutterstock.com/ru/image-photo/beautiful
-retro-luxury-light-lamp-decor-287606801
Beautiful retro luxury light lamp decor glowing        0.975
Создайте аккаунт бесплатно                            0.350
Пароли должны быть не короче 8 символов.              0.212
Shutterstock.com                                      0.186
Я согласен (согласна) с  Условиями пользования веб    0.170
СТАТИСТИКА SHUTTERSTOCK:                              0.150
Официальный логотип Shutterstock                      0.145
Музыка на Shutterstock                                0.144
Видео на Shutterstock                                 0.143
Shutterstock для Android                              0.141
Станьте автором Shutterstock                          0.138
Shutterstock для iOS                                  0.137
Уже есть аккаунт?  Войти                              0.135
© 2003-2017 Shutterstock, Inc. Все права защищены.    0.132
Купоны Shutterstock                                   0.126
ИД стоковой фотографии: 287606801                     0.117
Стоковых изображений без лицензионных платежей:  1    0.107
doroghoi                                              0.103
Загрузить                                             0.100
ghornyi khrustal                                      0.100
```

2) Результат работы программы без включения тестируемого сайта (shutterstock.com) в обучающую выборку.

```
mark@mark-N76VB ~/Workplace/PressFoto/graduation_project/prototypes/0.2 $
python3 prototype.py https://www.shutterstock.com/ru/image-photo/beautiful
-retro-luxury-light-lamp-decor-287606801
Beautiful retro luxury light lamp decor glowing        0.804
Создайте аккаунт бесплатно                            0.383
Пароли должны быть не короче 8 символов.              0.245
Я согласен (согласна) с  Условиями пользования веб    0.220
Shutterstock для iOS                                  0.213
Музыка на Shutterstock                                0.212
Shutterstock для Android                              0.209
Видео на Shutterstock                                 0.205
Shutterstock.com                                      0.200
Купоны Shutterstock                                   0.192
Станьте автором Shutterstock                          0.187
Официальный логотип Shutterstock                      0.162
СТАТИСТИКА SHUTTERSTOCK:                              0.161
Уже есть аккаунт?  Войти                              0.153
© 2003-2017 Shutterstock, Inc. Все права защищены.    0.150
ИД стоковой фотографии: 287606801                     0.123
ghornyi khrustal                                      0.097
Стоковых изображений без лицензионных платежей:  1    0.095
bliestiashchii                                        0.093
doroghoi                                              0.091
```