М.А. Ульрих, Ю.В. Бидуля

Тюменский государственный университет, г. Тюмень

УДК 004.912

ОПРЕДЕЛЕНИЕ ТЕМАТИКИ ВЕБ-СТРАНИЦ ПО ИХ ТЕКСТОВОМУ СОДЕРЖИМОМУ

Аннотация. Отслеживание истории веб-запросов персонала в рабочее время и определение их продуктивности – одна из основных функций систем учета времени. Целью исследования является классификация веб-страниц по тематике с помощью машинного обучения на основании их текстового содержимого.

Ключевые слова: веб-страница, классификатор, машинное обучение, учет рабочего времени.

Исследования показывают, что большинство людей уделяет менее 40% рабочего времени непосредственно работе [1]. Остальное время они занимаются своими делами, листая социальные сети, читая новости или играя в игры. Такой неэффективный расход времени приводит к потере денег компаниями, и поэтому руководители стремятся различными способами контролировать сотрудников и следить, как и насколько эффективно они используют свое рабочее время.

Один из способов контроля – специальное ПО, которое стало особенно актуально после пандемии Covid-19 и перехода на дистанционную работу. Учет рабочего времени удаленных сотрудников стал настоящей проблемой для руководства компаний, решить которую помогли программы для мониторинга удаленных сотрудников.

Функционал таких систем разнообразен, в частности, одной из главных функций является отслеживание активности сотрудников в сети. Однако, в ходе анализа существующих систем было выявлено, что их подход к мониторингу посещаемых веб-ресурсов не лишен недостатков.

В первую очередь это относится к тому факту, что при анализе веб-ресурсов и распределении их на “продуктивные” и “непродуктивные” используется только база данных с заранее классифицированным набором сайтов. В случае, если ресурс не найден, он помечается как неизвестный. Однако сегодня каждую минуту появляются новые сайты, и поддерживать актуальность такой базы крайне затруднительно, если и вовсе не невозможно. Поэтому имеет смысл использовать для целей классификации веб-ресурсов машинное обучение [2].

Для адекватного определения продуктивности персонала имеет смысл классифицировать ресурсы по темам, а затем выбирать, какие темы являются продуктивными и непродуктивными для конкретного сотрудника или отдела. Также имеет смысл классифицировать конкретные страницы, а не целые сайты, так как разные страницы на одном сайте могут иметь разный характер и тематику.

Для реализации классификации с помощью машинного обучения в первую очередь был сформирован датасет из веб-страниц. Для классификации страниц было выделено 7 категорий:

- Развлечения;
- Новости;
- Образование;
- Сайты для взрослых;
- Сайты для поиска работы;
- Интернет-магазины;
- Социальные сети.

Ввиду отсутствия готового датасета, далее были вручную отобраны страницы и определены их категории. Размер датасета составил 303 записи. Далее был осуществлен забор контента со всех страниц, из него была вырезана HTML-разметка и блоки, описывающие особенности функционирования или отображения – <script>, <style>, <audio>, <img> и т.д. Из всего оставшегося текста был сформирован датасет и предобработан путем удаления стоп-слов,

цифр, специальных символов, слов длиной менее трех букв, а также приведением к нижнему регистру.

На основе полученного датасета, с помощью сервиса Azure Machine Learning Studio было обучено пять разных классификаторов [3]:

- Логистическая регрессия;
- Нейронная сеть;
- Лес решений;
- Джунгли решений;
- Один против всех.

Схема обучения представлена на рисунке 1.

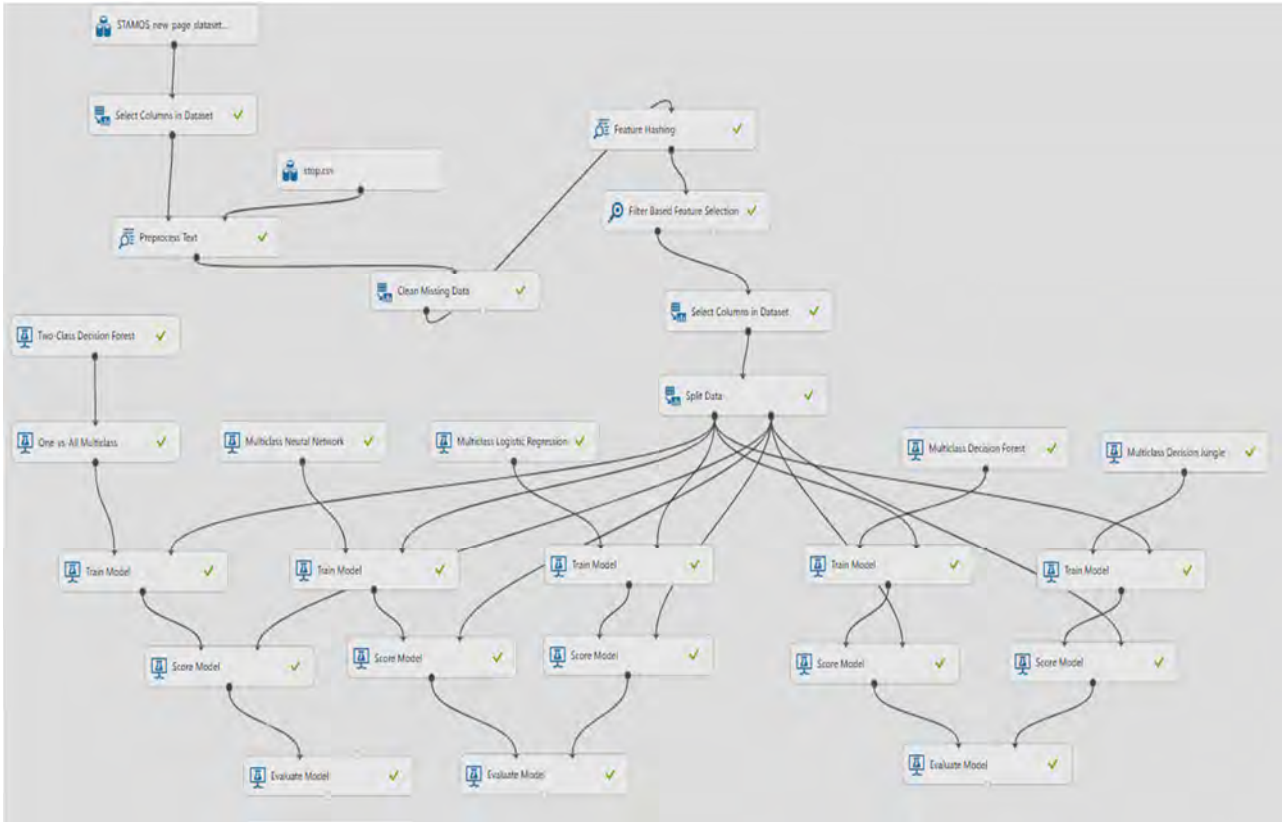

Рис. 1. Схема сравнения классификаторов

Лучший результат показал классификатор один-против-всех с точностью 71.3%. Данный результат был признан недостаточно высоким, и была поставлена цель его улучшения.

Как правило, текстовые элементы веб-страницы, можно разделить на следующие категории:

- Название страницы;
- Заголовки (<h1> <h2> и т.д.);
- Meta keywords;
- Meta description;
- Текстовый контент (<p>, , <label> и т.д.).

На основании этой информации была сформулирована гипотеза о том, что точность классификации можно улучшить путем разделения контента на выделенные категории и классификации отдельно по каждой из них.

Для проверки гипотезы, на следующем этапе эксперимента датасет был составлен следующим образом: отдельно происходит запись названия страницы, всех заголовков, которые присутствуют на странице, в отдельные поля записываются meta keywords и meta description, и отдельно весь остальной текст. С помощью сервиса Microsoft Azure Machine Learning Studio для каждого типа контента было обучено по пять ранее названных типов классификаторов с целью выявить лучший. Результаты обучения представлены в таблице 1.

Таблица 1. Сравнение точности классификаторов

		Тип классификатора				
		One-vs-All	Neural network	Logistic regression	Decision forest	Decision jungle
Тип контента	Текст	0.822	0.772	0.633	0.734	0.759
	Название	0.800	0.775	0.713	0.725	0.587
	Заголовки	0.783	0.838	0.716	0.689	0.622
	Meta description	0.648	0.662	0.549	0.606	0.606
	Meta keywords	0.759	0.724	0.483	0.724	0.690

Как видно из таблицы, классификатор один-против-всех показал наилучшие результаты для текста, названий и meta keywords. А для заголовков и meta description лучшей оказалась нейронная сеть.

Далее рассмотрим результаты лучших классификаторов по каждой категории (табл. 2).

Таблица 2. Результаты по категориям

	Категории веб-страниц							
	Для взрослых	Образовательные	Развлекательные	Поиск работы	Новости	Магазины	Социальные сети	Средняя точность
Текст	100%	58.3%	92.9%	80%	76.9%	88.9%	77.8%	82.3%
Название	100%	66.7%	64.3%	90.9%	84.6%	66.7%	88.9%	80%
Заголовки	83.3%	81.8%	100%	100%	75%	87.5%	55.6%	83.8%
Meta description	90%	28.6%	76.9%	90%	46.2%	55.6%	66.7%	66.2%
Meta keywords	100%	66.7%	80%	100%	75%	33.3%	60%	75.9%

По средней точности лучшим оказался классификатор заголовков, однако, как видно из таблицы, разные типы контента на веб-странице по-разному работают для разных категорий страниц. У каждого классификатора есть категории, которые они определяют лучше, и хуже. Причем какой-то классификатор хорошо показывает себя для одной категории, а какой-то – для другой.

Поэтому была сформирована гипотеза о том, что если объединить все классификатором определенным образом, то точность можно повысить. Поэтому на следующем шаге показатели из данной таблицы были использованы в качестве коэффициентов доверия при классификации.

После получения вероятностей для каждого класса от всех пяти классификаторов по типу контента, считается оценка степени принадлежности страницы к категории по формуле 1 – вероятности по каждому типу контента умножаются на соответствующие коэффициенты доверия и складываются по каждой категории страниц.

$$R_i = k_{i1} * p_{i1} + k_{i2} * p_{i2} + k_{i3} * p_{i3} + k_{i4} * p_{i4} + k_{i5} * p_{i5}, \quad (1)$$

где i – номер категории;

k_{i1} – коэффициент доверия классификатора j к категории i;

p_{i1} – вероятность категории i, полученная классификатором j;

R_i – итоговая оценка для категории i.

Итоговая категория веб-страницы определяется как категория с максимальной R.

Итоговая общая точность классификации, полученная экспериментально, составила 87.4%.

Таким образом, разделение контента веб-страницы по его типу и отдельная его обработка, а также расчет оценки категории на основе полученных индексов доверия позволили повысить точность классификации на 16.1% по сравнению с первоначальным подходом.

Основным языком разработки был выбран C#, в первую очередь ввиду удобной разработки под Windows. Для разработки веб-приложения использовался фреймворк ASP.NET 5. Для десктопного приложения применялась технология Windows Forms. Причинами такого решения являются, во-первых, возможность минимизации приложения в системный трей при попытке закрытия, что крайне проблематично реализовать, например, при использовании WPF. Во-вторых, разрабатываемое приложение не требует больше никакой особенной, гибкой настройки и не предполагает сложных дизайнерских решений, ввиду чего нет необходимости в использовании более сложных технологий.

Для машинного обучения система взаимодействует с веб-порталом Microsoft Azure Machine Learning Studio. Данный сервис отличается широким функционалом и удобным интерфейсом. Кроме того, важным фактором является возможность взаимодействия через API.

Кроме того, система использует брокер очередей RabbitMQ, снижающий нагрузку на веб-приложение и обеспечивающий высокую степень отказоустойчивость системы, необходимую при взаимодействии с внешними системами – анализируемыми веб-сайтами и порталом машинного обучения ввиду вероятности их недоступности. Получение данных об открытых

приложениях происходит следующим образом: берется список процессов, открытых на ПК, после чего записываются названия главных окон у тех, у которых оно есть (именно они нас интересуют, приложения без главных окон – сервисы). Подобная процедура забора и отправки данных об открытых приложениях происходит каждые 30 секунд. Если приложения, поступившего в текущий забор, не было в предыдущем или было проставлено время закрытия – оно записывается в историю с текущим временем как временем открытия. Если приложения, которое было открыто в предыдущий раз, нет в поступившем списке – для него записывается текущее время как время закрытия.

История открытых вкладок добывается из внутренней базы данных браузера. На данный момент рассматривается Google Chrome и его SQLite база данных (рисунок 10), которая копируется в директорию приложения, где для нее создается контекст и модели. Информация о вкладках и времени формируется из таблиц visits и urls и отправляется на сервер, где обрабатывается. Новые данные, если они есть, добавляются в базу. Запрос отправляется раз в 10 секунд. Аутентификация пользователя в десктопном приложении происходит через обращение к API веб-сервера.

Статистика сотрудников отображается с помощью тайм-лайнов. На рисунке 2 показн пример статистики по открытым приложениям.

Рис. 2. Статистика использованных приложений

При нажатии на строку браузера открывается окно со статистикой по истории посещенных веб-ресурсов (рис. 3). Выводится основная статистика в виде основных показателей и графиков по категориям и по продуктивности.

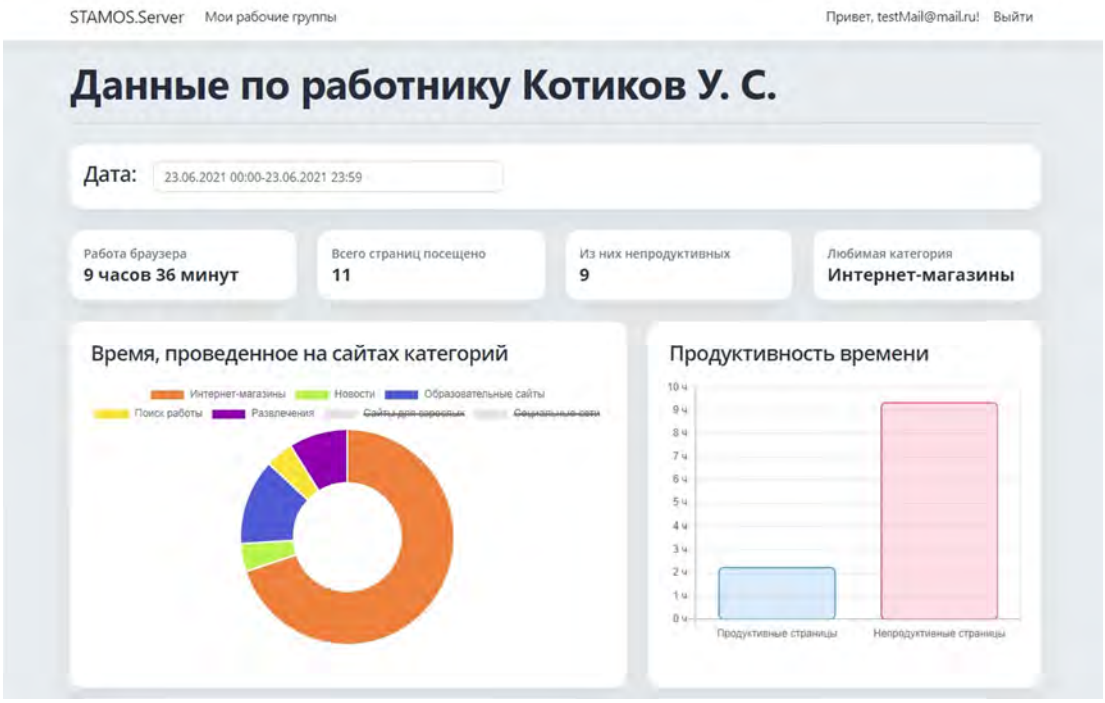

Рис. 3. Статистика активности браузера

История посещения веб-ресурсов отображается в виде таймлайна, цветами помечаются разные категории (рис. 4).

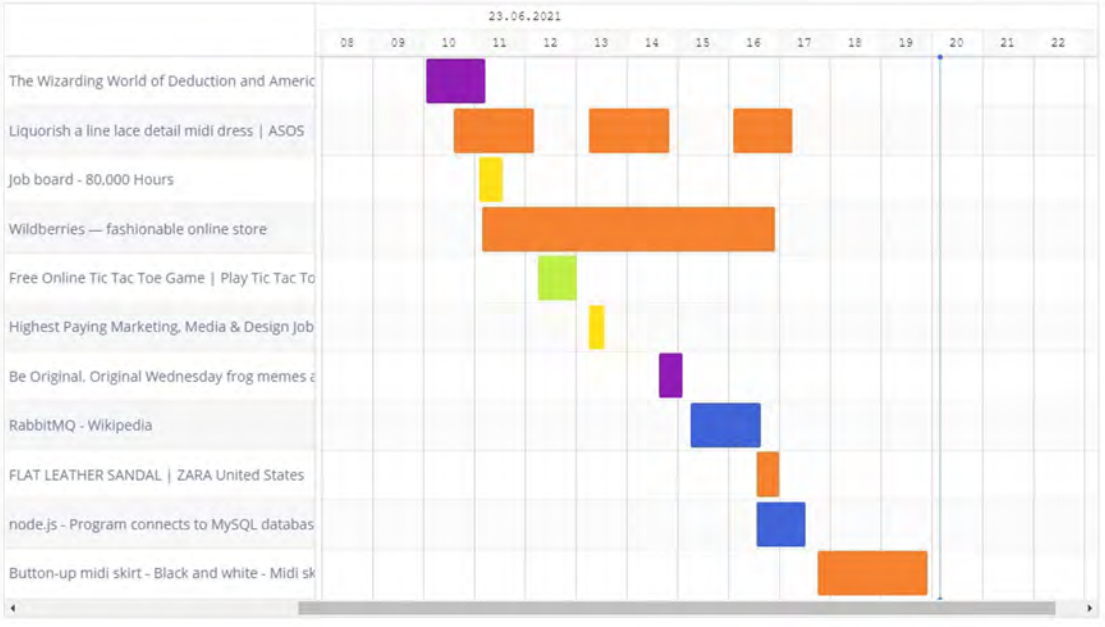

Рис. 4. Таймлайн открытых вкладок

Таким образом, разработанная система позволяет позволяет задавать “продуктивные” и “непродуктивные” тематики веб-ресурсов для конкретных рабочих групп и более эффективно принимать управленческие решения.

СПИСОК ЛИТЕРАТУРЫ

1. Британские ученые доказали, что работать больше трех часов в день – бессмысленно // Life, URL: https://life.ru/p/1049413 (дата обращения: 02.06.2021).

2. Jurafsky D., Martin J.H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Pearson Prentice Hall, 2009. 988 p.

3. Документация по Студии машинного обучения (классическая версия). URL: https://docs.microsoft.com/ru-ru/azure/machine-learning/classic/ (дата обращения: 02.06.2021).