1. **Планирование успеха и измерение его эффективности.**

* Сеть интернет – это отдельный мир, в котором большая часть правил обычного мира не работают, но тем не менее «онлайн» мир тесно переплетен с миром «оффлайн». В полной мере эту закономерность можно увидеть в сфере монетизации интернет сайтов. Заработок на интернет блоге в современном мире является перспективным направлением.

Основными способами заработка на блоге являются:

1. Партнерские программы – продажа не своих товаров и услуг по реферальной программе. В этом варианте особо интересна реклама чужих товаров или услуг. Реализация может заключаться в следующем: статья в блоге, рассказывающая о путешествиях может включать скрытую рекламу туроператора или гостиниц, упомянув которые в контексте статьи будет являться скрытой рекламой.
2. Торговля товарами и услугами – продажа своих собственных товаров (например, изделия ручной работы). Статья в блоге может содержать прямые указания на продаваемый товар, включая фотографии и видео товара, а также напрямую говорить о талантах автора статьи или иметь описание процесса производства, тем самым привлекая читателей мотивируя их на покупку.
3. Реклама в блоге – заработок, основанный на показе пассивной рекламе, представленной в виде баннеров, рекламных ссылок или контекстной рекламы. Обычно, используются баннерные сети, предоставляющие услуги по заработку на показе рекламы, например, Google AdSense.
4. Брендирование – помещение на фон (подложку) блога большой картинки или анимации, занимающей большое место. Один из самых доходных способов монетизации.
5. Размещение рекламных статей – размещение в своем блоге специальных рекламных статей, наряду с авторскими статьями. Как правило. Это уже написанные статьи, один-два абзаца рассказывающие о товаре или услуге.
6. Продажа ссылок – размещение ссылок в своих статьях. Ведущих на другие сайты. Суть этой системы заключается в том, что, чем больше источников ссылается на сайт, тем выше его рейтинг в поисковых запросах.

Это самые перспективные, на мой взгляд, способы монетизации блога, существуют и другие, но их эффективность не высокая.

* Среди факторов, непосредственно влияющих на успех монетизации блога необходимо выделить ключевые, выделенные на основе привлекательности блога для посетителей:

удобное и запоминающееся доменное имя,

тематика блога соответствует ожиданиям посетителей,

материалы блога помогают посетителям решать их проблемы.

Учитывая, вышеизложенное, можно составить следующую иерархию:

Blog

Domainname

Nützliche Beiträge

Anzahl der Ansichten

Anzahl der Kommentaren

Anzahl der Kommentaren erfüllt das angegebene Kriterium

Beitrag Monetarisierung

Erstellung neuer nützlicher Beiträge

Таким образом, основные показатели для принятия решения об эффективности монетизации блога, может служить количество комментариев статьи. Этот показатель мы можем назвать коэффициентом привлекательности. Следующим важным показателем, будет являться частота обновления контента блога, т.е. частота появления новых статей.

На эти основные показатели и нужно ориентироваться для создания эффективной модели монетизации блога.

1. **Изучение конкуренции на примере блога** [**https://www.bravebird.de/**](https://www.bravebird.de/)

Блог представляет собой сборник статей по тематике путешествий по направлениям: «расположение», «темы». «книги». Главное меню блога, помимо указателей направления статей, содержит также следующие пункты: «информация о создателе блога», «контактная информация».



Рис. 1. Главное меню блога.

На главной (домашней) странице блога собраны вновь добавленные статьи по всем разделам. Каждая статья относится одновременно к двум категориям: «расположение» и «темы». Такая структуризация статей блога облегчает поиск по статьям блога. Помогая быстрее найти интересующую информацию.

В свою очередь, каждая категория разбита на подкатегории. Категория «расположение» содержит уточняющие подкатегории по континентам и еще более детально – страны мира.

Категория «темы» содержит подкатегории, касающиеся способа путешествий.

В результате, мы можем сделать вывод, что блог представляет собой узконаправленную тематику, касающееся путешествий по миру. Для привлечения посетителей в блог, необходимо точно следовать курсу заинтересованности посетителей, для этого нужно отслеживать популярность тематики статей в блоге.

Так как, блог не является нашим собственным и нет возможности получить детальную статистику с помощью SMM-провайдеров, необходимо разработать стратегию сбора информации из блога косвенным путем.

Рассмотрим структуру одной из статей, для анализа, какую информацию мы можем получить для дальнейшего анализа.

Каждая статья состоит из следующих структурных блоков:

|  |
| --- |
| **Категории статьи**  <div class="categories">…</div> |
| **Тема статьи**  <h1 class="entry-title">…</h1> |
| **Информация о статье**  дата публикации <time class="entry-date published"> … </time>  количество комментариев <span class="post-info post-info-comment">…</span> |
| **Заглавное фото статьи.**  <div class="tnail">… </div> |
| **Текст статьи**  <div class="entry-content">…</div> |
| **Комментарии**  <div class="comment-list-wrap">…</div> |

По такой структуре можно получить следующую информацию о блоге и статьях в нем:

1. Популярность статей, как отношение темы статьи и количество комментариев к конкретной статье. Такой показатель даст ориентир, какую тематику стаей нужно больше добавлять в блог.
2. Какие сезоны года более популярны в блоге. Таким показателем является дата публикации статьи в блоге. По этому показателю можно определить, когда чаще всего добавляются статьи в блоге. И косвенно, какое время года больше подходит для публикации статей.
3. Количество комментариев к статьям в блоге за определенный сезон года. Условно разделив календарный год на 4 сезона, по росту (или спаду) количества комментариев (в целом, без привязки к темам статей и категориям) можно определить, когда блог читает наибольшее количество посетителей.
4. Какие по размеру статьи больше подходят для блога. Посчитав количество слов в статье и определив количество комментариев к статье, можно переделить средний показатель рекомендуемого размера статьи, как отношение количество комментариев к количеству слов в статье.
5. Определить косвенный KPI блога, по количеству комментариев в блоге за определенный период.
6. Составить «Топ-5» популярности статей в блоге по количеству комментариев к статье.
7. С помощью сервисов, похожих на Google Analitics, провести анализ посещения статей. Определить такие показатели как: время, которое посетитель провел на этой статье (даст нам понятие, о том, какие конкретно цели преследовал посетитель, зайдя на статью: изучить статью или бегло ознакомиться и перейти на другую статью); вернулся ли пользователь снова на эту статью в блоге.
8. Так же, с помощью Google Analitics провести анализ ключевых слов статьи и поисковых запросов к статье.

SMM-стратегия – это часть маркетинговой стратегии, работа над которой проходит в течении продолжительного времени. Для своего блога я разработал следующую стратегию:

1. Определить концепцию блога. Т.е. определиться с идеологией сообщества вокруг блога, выделить отличительные характеристики блога в сравнении с конкурентами.
2. Сформировать цели и задачи. Сформировать цели, например, с помощью модели SMART (конкретные, измеримые, достижимые, актуальные и ограниченные во времени). Отслеживать прогресс в достижении промежуточных и глобальных целей концепции. Основным показателем привлекательности блога (KPI) будет являться рост посетителей блога. Это и будет являться глобальной целью.
3. Определить целевую аудиторию. Определиться, на кого направлена общая стратегия развития блога. Правильно составленная модель посетителя блога поможет привлечь большее количество аудитории. Разделение целевой аудитории на мелкие сегменты, поможет повысить уровень коммуникации внутри каждого сегмента. В блоге необходимо представить категории статей для каждого сегмента аудитории.
4. Анализ. Сбор статистики по блогу: количество комментариев, количество просмотров, время, проведенное в блоге, активность пользователей и т.д.
5. Корректировка. Если начальные KPI невыполнимы в ближайшем будущем, определить, возможные ошибки, возможно неверно выбранная целевая аудитория. Откорректировать цели, согласно новой стратегии.

Анализ html кода страниц блога, показал, что блог использует сервис Google Analytics, на что указывает подключение модулей в коде блога:

*<!-- Global site tag (gtag.js) - Google Analytics -->*

1. **Разработка программы-поискового робота для сбора информации о блоге конкурента.**

Рассмотрев возможные варианты платформ для создания поискового-робота, я принял решение о реализации программы в виде двух модулей: сборщик данных и визуализатор собранных данных. Рассмотрим оба модуля отдельно.

* Модуль сбора данных. На мой взгляд, данная задача является реализацией задачи сбора и обработки большого количества данных, Современное название такого подхода – BigData. В настоящее время наиболее подходящим языком программирования для работы с BigData является Python. Конечно, подобное может быть реализовано и на других ЯП, но проведя сравнение наиболее подходящих ЯП для моего проекта, я выбрал именно Python. Результаты моего сравнения ЯП представлены в таблице:

|  |  |  |
| --- | --- | --- |
| **Язык программирования** | **Достоинства** | **Недостатки** |
| **R** | Большая гибкость и свободный код. Наличие большого количества библиотек на CRAN | Язык. Более подходящий для сложных статистических вычислений. Сложность обучения. Интерфейс командной строки. |
| **Python** | Популярный язык в науке. Низкий порог вхождения. Огромный набор готовых фреймворков В отличие от R, Python является традиционным объектно-ориентированным языком программирования, поэтому большинству разработчиков будет комфортно с ним работать. | Основной недостаток – медленное выполнение программ на языке Python, обусловленное тем, что язык является интерпретируемым. |
| **Java** | Предоставляет доступ к огромной экосистеме библиотек профилировщиков, отладчиков, все из которых используются давно и много. | Основными претензиями к Java являются многословность и отсутствие REPL. |

Таким образом, я остановил свой выбор на Python и фреймворке Srapy. В качестве хранилища собранных данных будет использована MongoDB. MongoDB выбрана потому, что позволяет хранить данные в формате JSON в виде коллекций и просто интегрируется с любым другим фрейворком.

* Для визуализации данных выбран NodeJS – как самый популярный фреймворк для Front-end разработки.

Выбор библиотеки Python для реализации сборщика данных.

Самыми популярными библиотеками для написания сборщика данных являются BeautufulSoup и Scrapy. Но у каждой этой библиотеки есть свои плюсы и минусы.

Scrapy - это платформа для совместной работы с открытым исходным кодом для извлечения данных с веб-сайтов. Его производительность невероятно высока, и это одна из самых мощных доступных библиотек. Одним из ключевых преимуществ scrapy является то, что он построен на основе Twisted, асинхронной сетевой структуры, что означает, что scrapy использует неблокирующий механизм при отправке запросов пользователям.

BeautufulSoup – простой и удобный инструмент для извлечения данных с веб-сайтов. Проблема этой библиотеки состоит в том, что он требует дополнительных библиотек для своей работы. Зависимости Beautifulsoup:

* Необходима библиотека для отправки запроса на веб-сайт, потому что сам Beautifulsoup не может сделать запрос на конкретный сервер. Для преодоления этой проблемы требуется помощь одной из популярных библиотек Requests или urlib2***.***
* После загрузки данных HTML или XML, Beautifulsoup требуется внешний анализатор для анализа загруженных данных. Наиболее известные парсеры - это XML-parser lxml, HTML-parser lxml, HTML5lib, html.parser.

Библиотека Scrapy потому, что:

1. Scrapy имеет встроенную поддержку для извлечения данных из источников HTML с использованием выражений XPath и CSS.
2. Кросплатформенность
3. Быстрее, чем другие существующие библиотеки. Реализация сборщика данных.
4. Потребляет намного меньше памяти и ресурсов процессора.
5. Имеет хорошую поддержку сообщества для разработчиков

Общий принцип работы сборщика (в терминах Scrapy – «паук»), заключается в следующем: «паук» загружает страницу и ищет на ней данные, указанные с помощью правил xPath. После сбора всех данных на странице, выделяет ссылки на странице и пробует загрузить страницы по ссылкам и собрать данные с них. Для того, чтобы не выйти за пределы интересующего нас сайта, применяется ограничение для «паука». В моем случае, ограничение выглядит как: *allowed\_domains = ["bravebird.de"]* означающее, что «пауку» нельзя переходить по ссылкам, если имя домена в них не начинается с *bravebird.de,* т.е. мы ограничиваем переходы «паука» только доменом *bravebird.de.*

Блок-схема алгоритма работы «паука»:

Извлекаем ссылки со страницы

Записываем данные

Загрузка страницы, поиск информации по правилам xPath

Начальная ссылка

Выход из процесса сбора данных

Ссылка в разрешенном домене?

Для загрузки страницы мы должны указать «пауку» его имя и начальные настройки:

class BravebirdSpider(CrawlSpider):

#имя паука в проекте

name = 'bravebird'

#разрешенный домен, для механизма обхода страниц сайта, чтобы не выйти за пределы сайта

allowed\_domains = ["bravebird.de"]

# Стартовый адрес сайта

start\_urls = ['https://www.bravebird.de']llowed\_domains = ["bravebird.de"]

«Паук» скачает страницу и будет искать данные на странице, согласно правил. Для «паука» правила можно задавать различными способами, самым эффективным является xPath. Этот способ предполагает указание селекторов в DOM дереве странице, по которому нам нужно извлечь данные.

Для интересующих нас данных, правила будут выглядеть следующим образом:

**Название статьи:**

//div[@class="entry-header entry-header-top"]/h1//text()

**Дата публикации:**

//div[@class="entry-header entry-header-top"]/div[@class="entry-meta entry-meta-single"]/span[@class="post-info post-info-date"]/time[@class="entry-date published"]/text()

**Дата обновления статьи:**

//div[@class="entry-header entry-header-top"]/div[@class="entry-meta entry-meta-single"]/span[@class="post-info post-info-date"]/time[@class="updated"]/text()

**Количество комментариев к статье:**

//div[@class="entry-header entry-header-top"]/div[@class="entry-meta entry-meta-single"]/span[@class="post-info post-info-comment"]/a/text()

**Место, относящееся к статье:**

//div[@class="entry-header entry-header-top"]/div[@class="entry-meta entry-meta-single"]/span[@class="post-info post-info-location"]//text()'

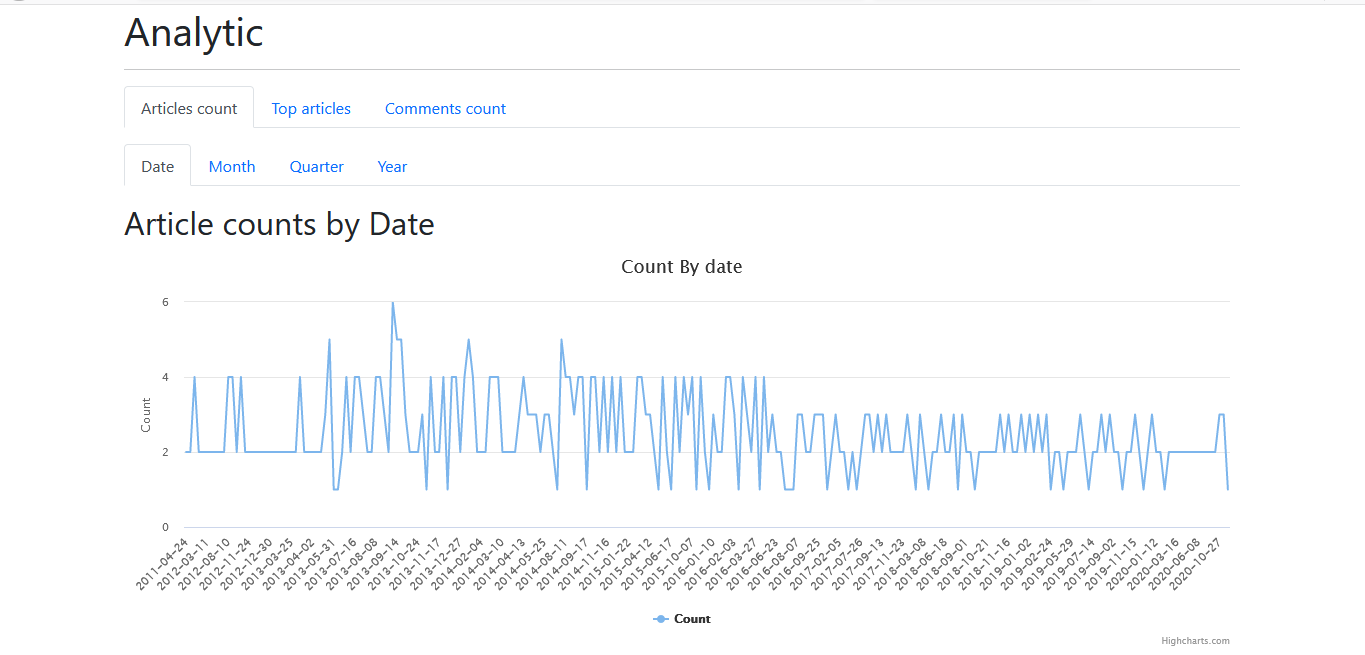
1. **Визуализация и интерпретация полученных данных.**

Полученные и обработанные webcrawler данные будут визуализироваться с помощью фреймворка NodeJS.

Задачей «визуализатора» будет являться извлечение данных из БД «паука» и построение графиков для удобного чтения различных собранных параметров.

Основой для построения «визуализатора» будет библиотека Express, позволяющий просто реализовать функции web-приложения. В моем проекте, будет использовать модуль HTTP библиотеки Express, предоставляющий доступ к данным через web-браузер.

Непосредственно построением графиков будет заниматься библиотека Highcharts - библиотека диаграмм на основе чистого JavaScript, предназначенная для интерактивного построения диаграмм.

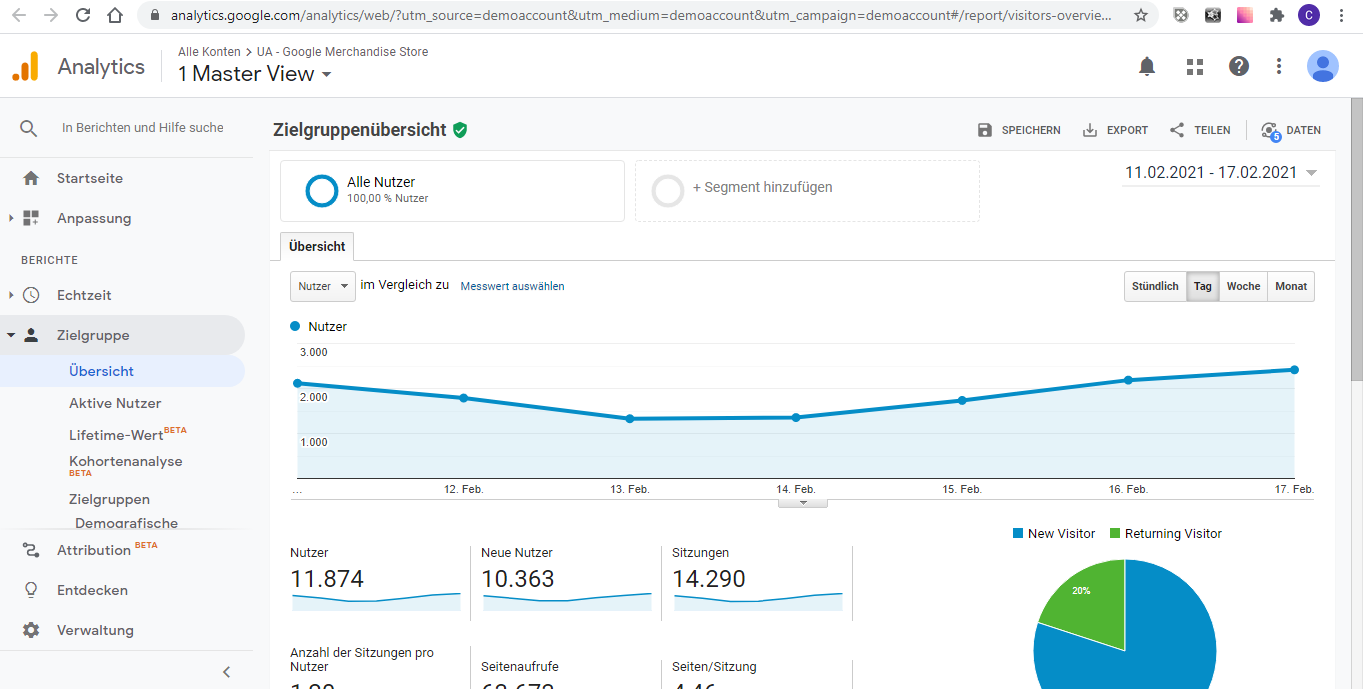


*http://vds.kochubinfo.ru:3000/analytic*

Интерфейс web-приложения разделен на вкладки. Для удобства навигации по полученным результатам. Реализовано 3 вкладки: **Articles count,** **Top articles,** **Comments count**. Каждая вкладка содержит соответствующие показатели, представленные в виде графиков.

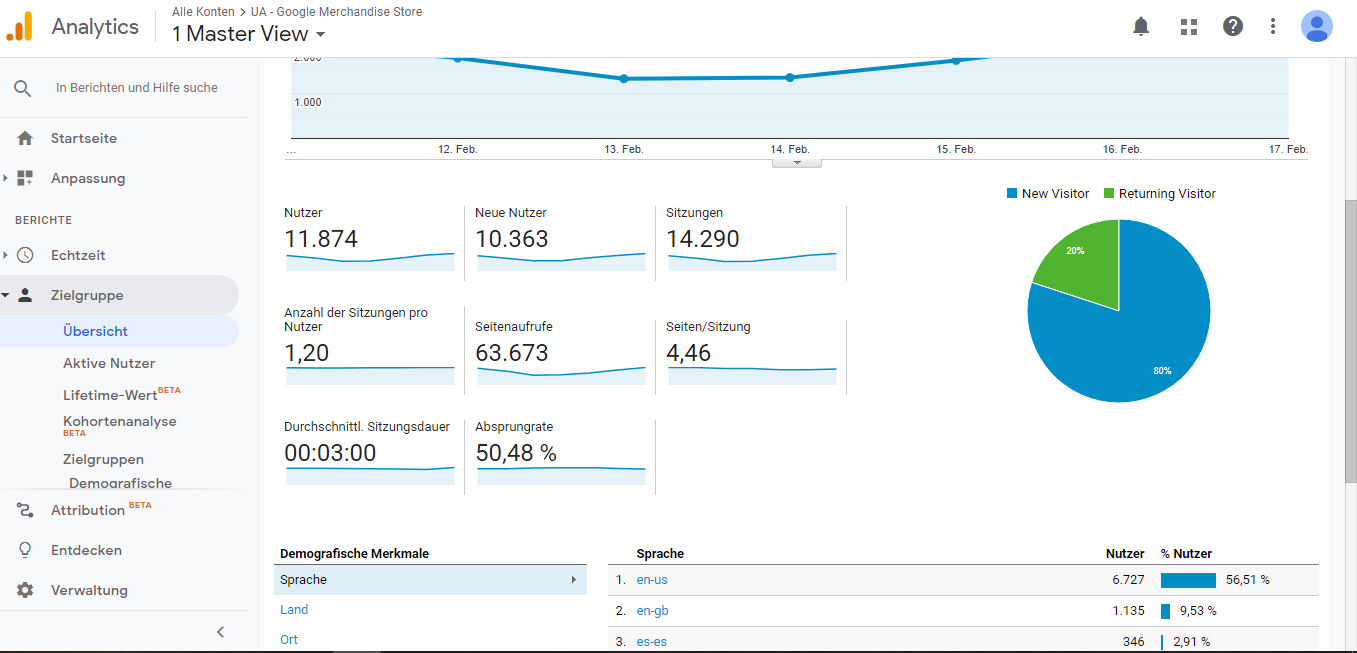
1. **Задание.**

Google Merchandise Store – это интернет-магазин, в котором продается сувенирная продукция Google. Тестовый аккаунт Google Analitics – это полнофункциональный аккаунт, доступный для любого пользователя Google. В нем содержатся ресурс Universal Analytics и два ресурса "Google Аналитика 4". С помощью тестового аккаунта можно просматривать коммерческие данные и экспериментировать с функциями Google Аналитики.



*https://analytics.google.com/analytics/web/?utm\_source=demoaccount&utm\_medium=demoaccount&utm\_campaign=demoaccount#/report/visitors-overview/a54516992w87479473p92320289/*

Для интернет-магазина интересны будут следующие показатели:



*https://analytics.google.com/analytics/web/?utm\_source=demoaccount&utm\_medium=demoaccount&utm\_campaign=demoaccount#/report/visitors-overview/a54516992w87479473p92320289/*

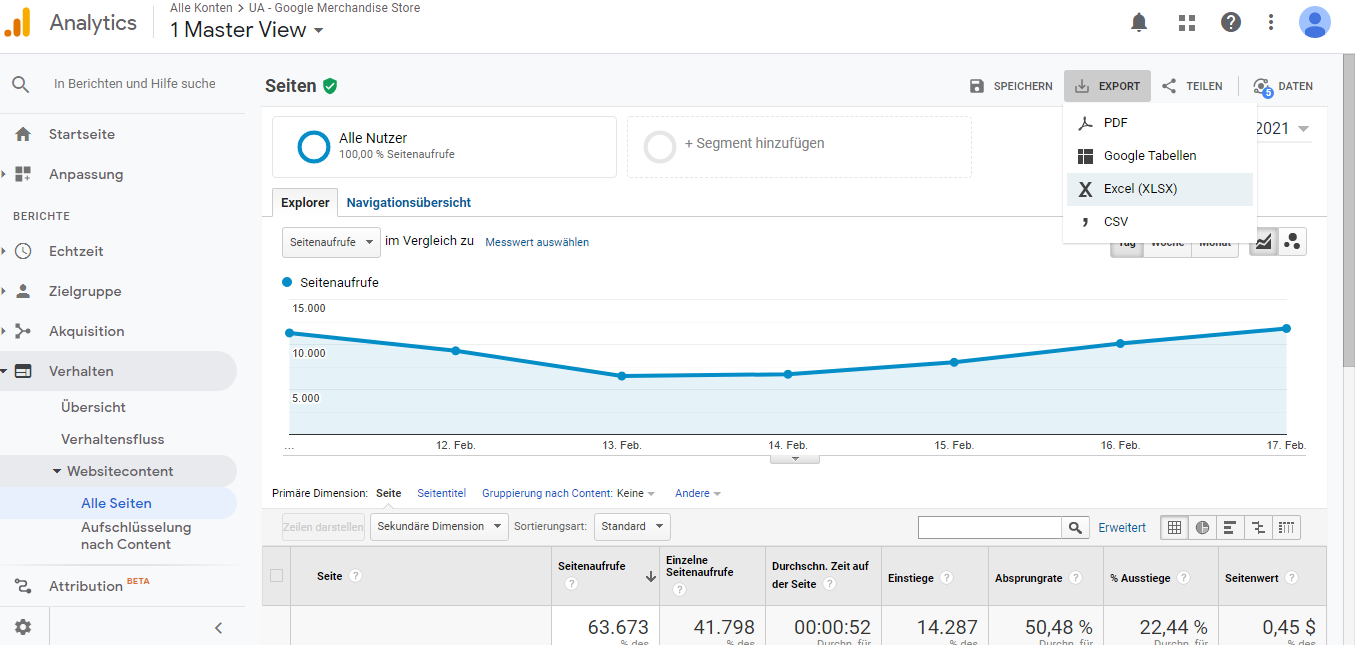
* Пользователи, даст понятие о том, посещают ли магазин пользователи.
* Сеансы, покажет, насколько активно работают пользователи с магазином, а косвенно, насколько информация в магазине интересна для посетителей.
* Страниц за сеанс, информация о том, переходят ли пользователи на страницы других товаров во время работы с магазином.
* Отказы, показатель полезности информации в магазине пользователям.
* Вовлечение, сколько времени в среднем проводит пользователь в магазине.
* Раздел Источники траффика, информация, которая будет полезна для анализа рекламной компании магазина. Откуда посетители заходят в магазин, на какие информационные источники следует обратить внимание в ходе рекламной компании.

Учитывая указанные выше интересующие показатели, можно использовать такие модели корректировки показателей:

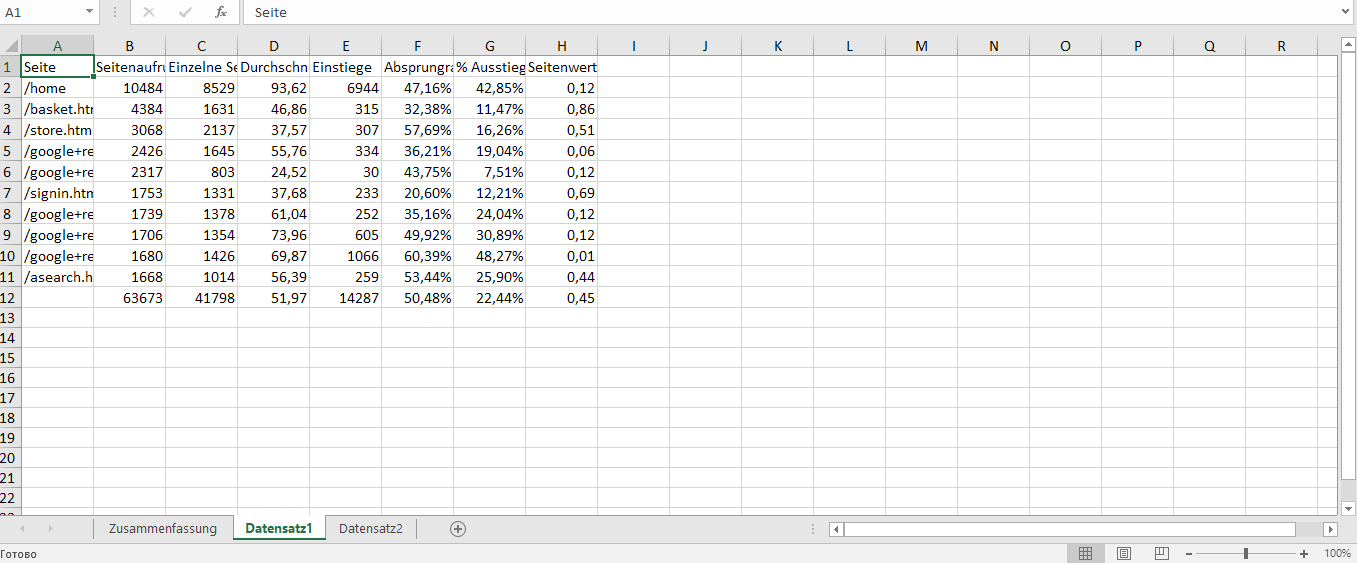
* Низкие показатели посещаемости – необходимо оптимизировать рекламную компанию, для этого обратимся к разделу Источники траффика, обратим внимание те источники, в которых присутствует реклама нашего магазина, но низкие показатели переходов. Для них, разработаем новую стратегию и снова запустим рекламную компанию. Наблюдая за показателями.
* Низкие показатели Сеансов пользователей. Может означать, что у нас есть проблемы с контентом интернет магазина. Для решения проблемы нужно обратиться к разделу Страницы выхода, проанализировав эту информацию. Мы сможем сделать вывод, на каком этапе просмотра магазина пользователи уходят. Выявив проблемные страницы. Необходимо произвести их оптимизацию. При анализе проблемных страниц нужно помнить, что есть страницы, которые логично завершают работу с магазином, такие как: корзина и оплата. В случае с корзиной, пользователь может зайти на эту страницу для проверки результата заказа товара, а проверив выйти. В случае со страницей Оплата, пользователь, выбрав и оплатив товар уходит из магазина, что не является показателем проблемы.
* Низкие показатели Страниц за сеанс – может указывать на проблемы с рекомендациями дополнительных товаров к основному, которым заинтересован пользователь. Нужно оптимизировать подбор рекомендованных товаров. Например, товар, использующий в своей работе батарейки, должен быть снабжен рекомендацией различных видов подходящих батареек.
* Высокие показатели Отказы – показывает, что пользователь, заходя на страницу в магазине, не находит нужного товара и уходит. Необходимо проанализировать раздел Страницы входа, обратив внимание на показатель Страница входа и соответствующий ему показатель отказов. Это даст нам понять на какие страницы попадает пользователь из поисковых запросов или рекламных блоков и сразу уходит. Выявив такие страницы нужно произвести их анализ и корректировку.

При работе с GoogleAnalitics можно экспортировать нужные нам показатели, для обработки в других программах:

Excel: Нажав кнопку Экспорт, выбрать экспорт в Excel.



*https://analytics.google.com/analytics/web/?utm\_source=demoaccount&utm\_medium=demoaccount&utm\_campaign=demoaccount#/report/content-pages/a54516992w87479473p92320289/*

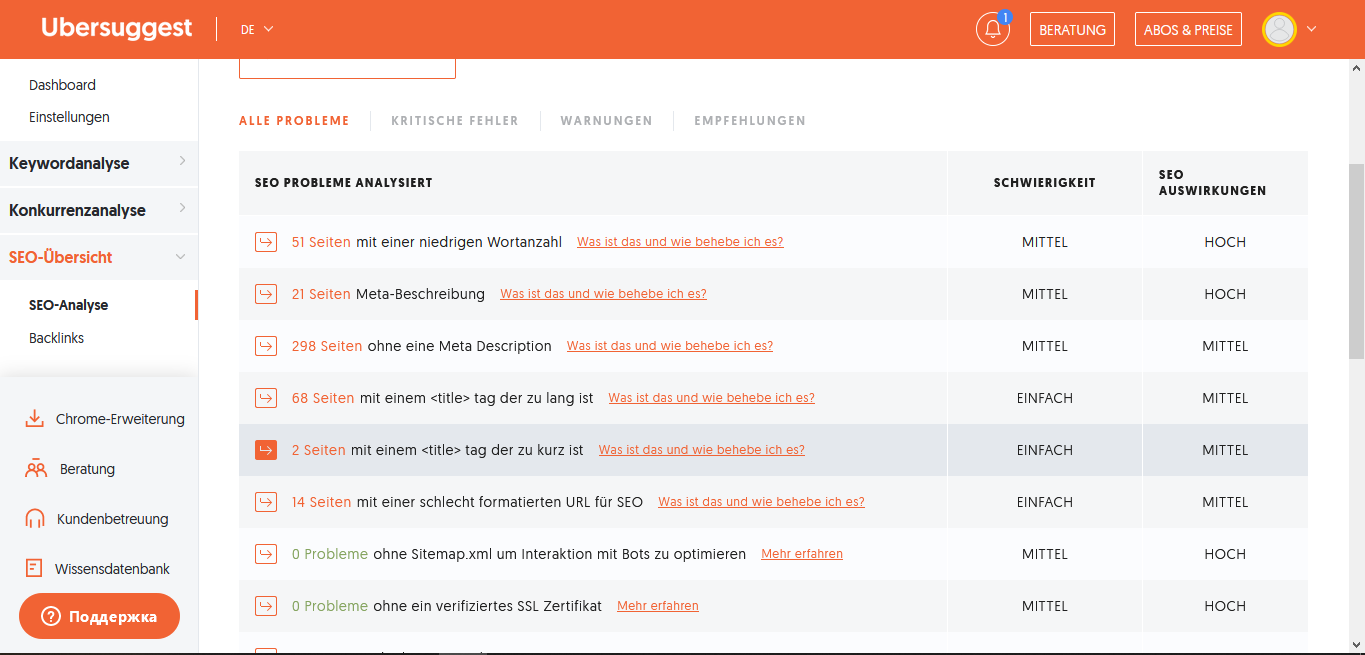


Другим полезным инструментом является платформа neilpatel.com. Удобный, не перегруженный интерфейс. В котором основные функции вынесены в меню, а соответствующие дополнительные функции доступны, как подпункты основного. На главной странице аналитики, платформа сразу показывает SEO привлекательность и набор ключевых параметров сайта, так называемый быстрый аудит сайта.

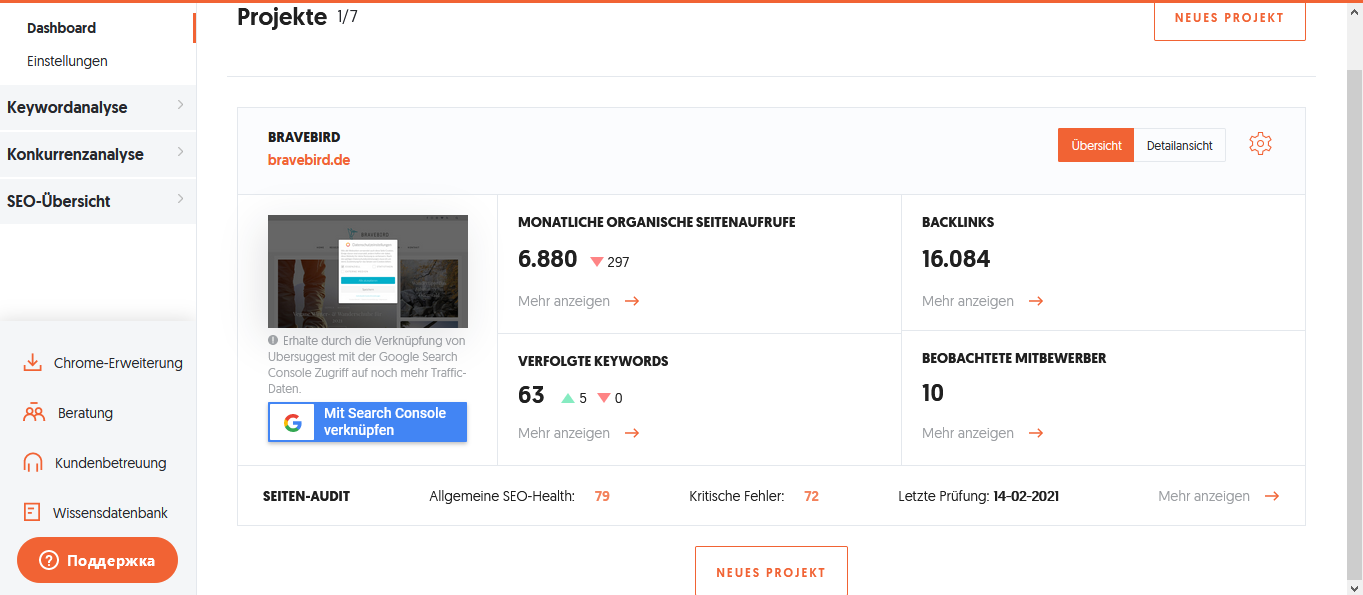
Здесь же доступна функция, позволяющая проанализировать сайт на проблемы с содержимым, таких как: проблемы верстки (которые мешают правильно работать поисковым роботам), никуда не ведущие ссылки (defect). По каждой найденной проблеме можно получить подробную аннотацию, нажав на название проблемы.



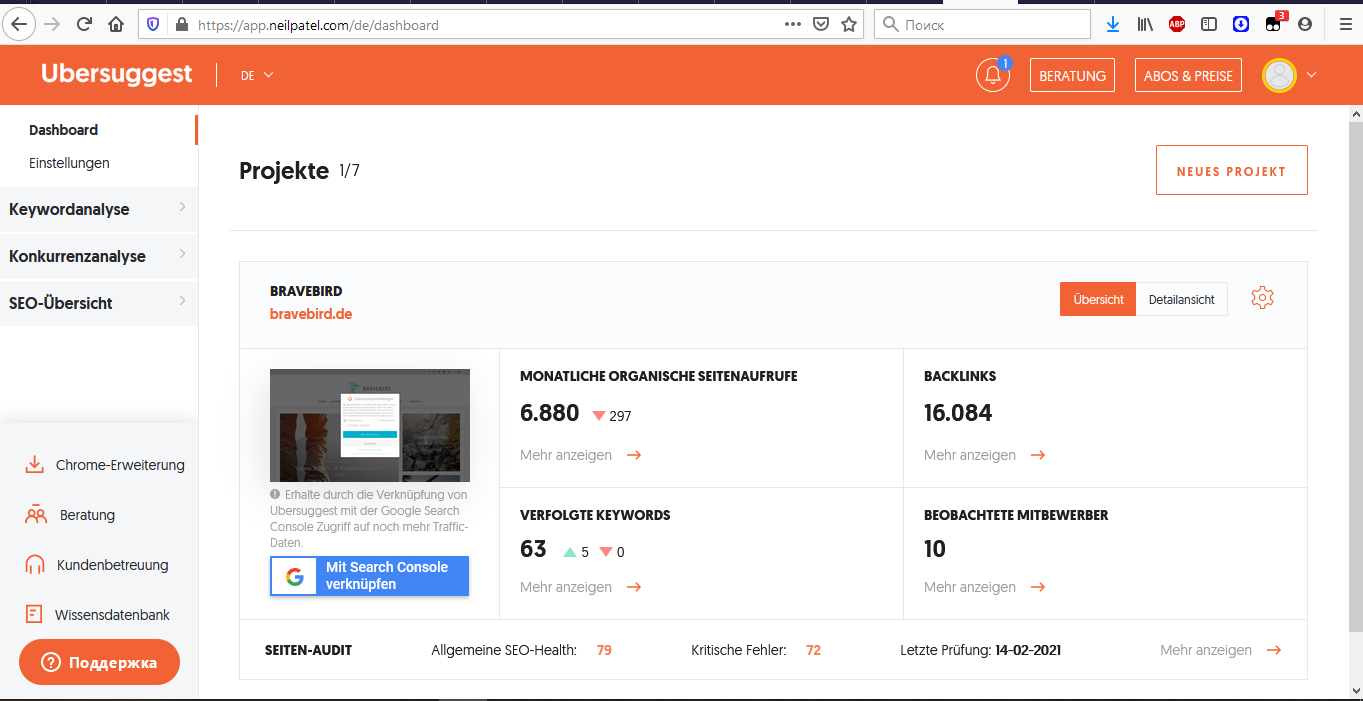
[*https://app.neilpatel.com/de/seo\_analyzer/site\_audit?domain=bravebird.de*](https://app.neilpatel.com/de/seo_analyzer/site_audit?domain=bravebird.de)

**

*https://app.neilpatel.com/de/seo\_analyzer/site\_audit?domain=bravebird.de&view=all*

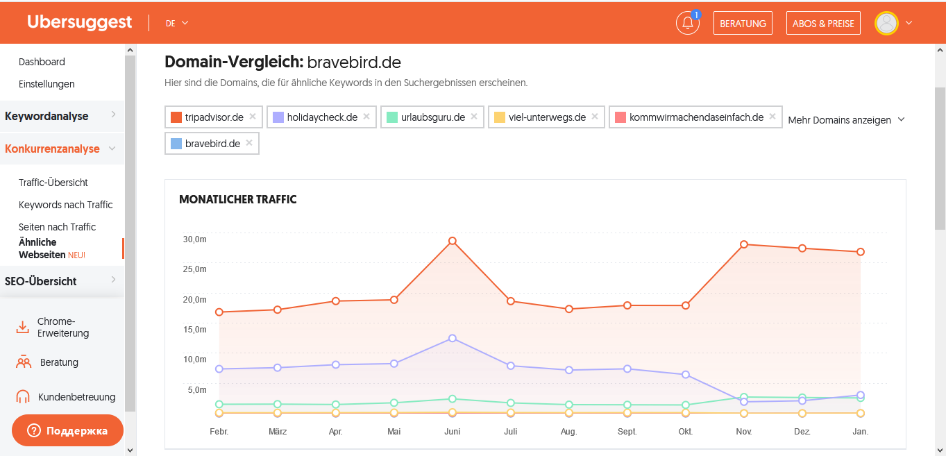


*https://app.neilpatel.com/de/dashboard*



*https://app.neilpatel.com/de/traffic\_analyzer/overview?domain=bravebird.de&locId=2276&lang=de*

Платформа обладает множеством полезных для веб-аналитики инструментов. Из самых интересных необходимо отметить: Ähnliche Webseiten, позволяющий искать похожие сайты, для оценки позиций своего.



*https://app.neilpatel.com/de/traffic\_analyzer/competitors?domain=bravebird.de&locId=2276&lang=de*

Функции, позволяющие получить список ключевых слов, по которым были переходы на сайт, что позволит оптимизировать текст статьи для лучших результатов.

Сервис имеет специальный плагин для браузера Chrome, для быстрого доступа к основной информации для сайтов, просматриваемых в настоящий момент.