МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Кафедра прикладной математики**

**КУРСОВАЯ РАБОТА**

**АНАЛИЗ СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ» ПРИ ПОМОЩИ ЯЗЫКА ПРОГРАММИРОВАНИЯ «PYTHON»**

Работу выполнил\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_П.С. Куприянов

 (подпись, дата) (инициалы, фамилия)

Факультет компьютерных технологий и прикладной математики курс 3

Направление 01.03.02 Прикладная математика и информатика

Научный руководитель

профессор, к.ф. – м.н.,

доцент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Н. Кармазин

 (подпись, дата) (инициалы, фамилия)

Нормоконтролер,

к.ф. – м.н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Г.В. Калайдина

 (подпись, дата) (инициалы, фамилия)

Краснодар 2017

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc506531390)

[1 Анализ социальной сети 4](#_Toc506531391)

[2 Реализация программы 5](#_Toc506531392)

[2.1 Библиотеки и методы 5](#_Toc506531393)

[2.2 Алгоритм программы 11](#_Toc506531394)

[3 Анализ полученных результатов 16](#_Toc506531395)

[3.1 Анализ по количеству лайков 18](#_Toc506531396)

[3.2 Анализ по количеству репостов 20](#_Toc506531397)

[3.3 Поиск заинтересованных пользователей 22](#_Toc506531398)

[ЗАКЛЮЧЕНИЕ 25](#_Toc506531399)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 26](#_Toc506531400)

[ПРИЛОЖЕНИЕ А 27](#_Toc506531401)

# ВВЕДЕНИЕ

Данная работа посвящена анализу данных, представленных в социальной сети «ВКонтакте» – одной из наиболее популярных сетей в России.

Анализ социальных сетей – направление компьютерной социологии, в рамках которой, производится анализ возникающих в ходе социального взаимодействия сетей различной плотности. Результаты, полученные в ходе анализа, могут быть применены в маркетинге в социальных сетях [1].

Целью данной работы является выявление предпочтений целевой аудитории , а так же поиск вероятного контингента, за счёт которого можно увеличить количество участников публичного сообщества.

В ходе работы была написана программа при помощи языка программирования Python 3.6.3, которая позволяет получить данные о количестве лайков и репостов записей на странице группы «Музыкальный центр SOROKA | школа Краснодар» социальной сети «ВКонтакте» с целью их дальнейшего анализа. Кроме того, программа позволяет предположить, за счёт каких пользователей можно увеличить аудиторию сообщества.

Программа использует «API (application programming interface) ВКонтакте» для упрощения создания кода. API представляет собой набор готовых классов, функция и структур для работы с данными, предоставляемыми социальной сетью «ВКонтакте». Это интерфейс, который позволяет получать информацию из базы данных vk.com с помощью http–запросов к специальному серверу. Синтаксис запросов и тип возвращаемых ими данных строго определены на стороне самого сервиса [2].

# Анализ социальной сети

Анализ социальных сетей – это сбор и обработка информации в социальных медиа, необходимый для планирования стратегии взаимодействия с целевой аудиторией. Под взаимодействием в данном случае подразумевается маркетинг в социальной сети – увеличение популярности публичного сообщества среди определённого контингента.

Анализ социальной сети разумно проводить и как подготовку к старту социального проекта, и как мероприятие по продвижению и сохранению популярности в процессе поддержания проекта.

Анализ социальной сети позволяет оперировать такими данными, как:

* наиболее актуальные темы для целевой аудитории;
* сведения о конкуренции в конкретной области специализации;
* уровень репутации бренда, а так же информация о площадках, которые в той или иной степени на неё влияют.

Анализ социальной сети позволяет:

* более тщательно прорабатывать план работы с целевой аудиторией того или иного бренда;
* своевременно отвечать на реакцию аудитории;

В процессе проделанной работы, была разработана программа, реализующая анализ социальной сети «ВКонтакте» на примере обработки данных со страницы публичного сообщества, составления рейтинга популярной информации, а так же сбор данных, на основе которых составляется представление о группах лиц, за счёт которых возможно расширить аудиторию сообщества.

# Реализация программы

# Библиотеки и методы

Программа разработана при помощи языка программирования Python 3.6.3. Данное средство разработки – это мощный инструмент для создания программ различного характера. С его помощью можно решать задачи различных типов. Python лёгок в понимании и удобен в использовании, что является несомненным преимуществом данного языка программирования, которое играет немаловажную роль при выборе средства для разработки приложения [3].

В составленной программе используются следующие библиотеки:

* vk – библиотека Python, для работы с инструкциями «API ВКонтакте», которая позволяет отправлять http запросы [4];
* csv – модуль для работы с csv файлами, в том числе для создания и редактирования csv таблиц [5];
* pandas – библиотека для обработки и анализа данных, в том числе чтения, записи и редактирования xlsx, csv файлов [6];
* datetime – модуль, предоставляет собой классы для обработки времени и даты [7];
* xlrd – библиотека для чтения xlsx и других файлов excel–совместимых файлов [8].

При работе программы производится несколько API запросов с целью получения данных с сайта <https://vk.com/>. В список используемых методов входят:

* wall.get – API метод, предназначенный для работы с записями на стенах сообществ и пользователей сети «ВКонтакте». Метод возвращает список записей со стены. Параметры метода приведены в таблице 1.

Таблица 1 – Список параметров метода wall.get

|  |  |
| --- | --- |
| owner\_id | идентификатор пользователя или сообщества, со стены которого необходимо получить записи (по умолчанию – текущий пользователь) |
| domain | короткий адрес пользователя или сообщества |
| offset | смещение, необходимое для выборки определенного подмножества записей |
| count | количество записей, которое необходимо получить. (максимальное значение: 100) |
| filter | определяет, какие типы записей на стене необходимо получить. Возможные значения:suggests – предложенные записи на стене сообщества (доступно только при вызове с передачей access\_token);postponed – отложенные записи (доступно только при вызове с передачей access\_token);owner – записи владельца стены;others – записи не от владельца стены;all – все записи на стене (owner + others).По умолчанию: all. |
| extended | 1 – в ответе будут возвращены дополнительные поля profiles и groups, содержащие информацию о пользователях и сообществах. По умолчанию: 0. |
| fields | писок дополнительных полей для профилей и групп, которые необходимо вернуть. См. описание полей объекта user и описание полей объекта group. Обратите внимание, этот параметр учитывается только при extended = 1. |

После успешного выполнения возвращает объект, содержащий число результатов в поле count и массив объектов записей на стене в поле items.

Если был задан параметр extended=1, возвращает число результатов в поле count, отдельно массив объектов записей на стене в поле items, пользователей в поле profiles и сообществ в поле groups [9].

* groups.isMember – метод API, возвращает информацию о том, является ли пользователь участником сообщества. Параметры метода представлены в таблице 2:

Таблица 2 – Список параметров метода groups.isMember

|  |  |
| --- | --- |
| group\_id | идентификатор или короткое имя сообщества |
| user\_id | идентификатор пользователя |
| user\_ids | Идентификаторы пользователей, не более 500 |
| extended | 1 – вернуть ответ в расширенной форме. По умолчанию – 0 |

После успешного выполнения возвращает 1 в случае, если пользователь с идентификатором user\_id является участником сообщества с идентификатором group\_id, иначе 0.

При использовании параметра extended, возвращает объект, который содержит поля: member (integer, [0,1]) – является ли пользователь участником сообщества, request (integer, [0,1]) – есть ли непринятая заявка от пользователя на вступление в группу (такую заявку можно отозвать методом [groups.leave](https://vk.com/dev/groups.leave)), invitation (integer, [0,1]) – приглашён ли пользователь в группу или встречу.

При передаче нескольких идентификаторов возвращает результат в виде массива объектов, в которых есть поля user\_id (integer) и member (integer, [0,1]) [10].

* likes.getList – метод возвращает  список идентификаторов пользователей, которые добавили заданный объект в свой список Мне нравится. Параметры метода описаны в таблице 3:

Таблица 3 – Список параметров метода likes.getList

|  |  |
| --- | --- |
| type | тип объекта |
| owner\_id | идентификатор владельца Like–объекта: id пользователя, id сообщества (со знаком «минус») или id приложения. Если параметр type равен sitepage, то в качестве owner\_id необходимо передавать id приложения. Если параметр не задан, то считается, что он равен либо идентификатору текущего пользователя, либо идентификатору текущего приложения (если type равен sitepage).  |
| item\_id | идентификатор Like–объекта. Если type равен sitepage, то параметр item\_id может содержать значение параметра page\_id, используемый при инициализации[виджета «Мне нравится»](https://vk.com/dev/Like).  |
| page\_url | url страницы, на которой установлен [виджет «Мне нравится»](https://vk.com/dev/Like). Используется вместо параметра item\_id, если при размещении виджета не был указан page\_id.  |
| filter | указывает, следует ли вернуть всех пользователей, добавивших объект в список "Мне нравится" или только тех, которые рассказали о нем друзьям. Параметр может принимать следующие значения: likes – возвращать информацию обо всех пользователях; copies – возвращать информацию только о пользователях, рассказавших об объекте друзьям.По умолчанию возвращается информация обо всех пользователях.По умолчанию: all. |
| friends\_only | указывает, необходимо ли возвращать только пользователей, которые являются друзьями текущего пользователя. Параметр может принимать следующие значения:0 – возвращать всех пользователей в порядке убывания времени добавления объекта;1 – возвращать только друзей текущего пользователя в порядке убывания времени добавления объекта;Если метод был вызван без авторизации или параметр не был задан, то считается, что он равен 0. |
| extended | 1 – возвращать расширенную информацию о пользователях и сообществах из списка поставивших отметку «Мне нравится» или сделавших репост. По умолчанию – 0.  |

Продолжение таблицы 3

|  |  |
| --- | --- |
| type | тип объекта |
| offset | смещение, относительно начала списка, для выборки определенного подмножества. Если параметр не задан, то считается, что он равен 0.  |
| count | количество возвращаемых идентификаторов пользователей.Если параметр не задан, то считается, что он равен 100, если не задан параметр friends\_only, в противном случае 10.Максимальное значение параметра 1000, если не задан параметр friends\_only, в противном случае 100.  |
| skip\_own | не возвращать самого пользователя.  |

После успешного выполнения возвращает [список](https://vk.com/dev/list) идентификаторов пользователей с учетом параметров offset и count, которые добавили заданный объект в свой список «Мне нравится».
Если параметр type равен sitepage, то будет возвращён список пользователей, воспользовавшихся виджетом «Мне нравится» на внешнем сайте. Адрес страницы задаётся при помощи параметра page\_url или item\_id.
Если extended=1, дополнительно возвращается массив items, содержащий расширенную информацию о [пользователях](https://vk.com/dev/objects/user) или [сообществах](https://vk.com/dev/objects/groups) [11].

* friends.get – метод возвращает список идентификаторов друзей пользователя или расширенную информацию о друзьях пользователя (при использовании параметра fields). Параметры метода приведены в таблице 4.

Таблица 4 – Параметры метода friends.get

|  |  |
| --- | --- |
| user\_id | идентификатор пользователя, для которого необходимо получить список друзей. Если параметр не задан, то считается, что он равен идентификатору текущего пользователя (справедливо для вызова с передачей access\_token). |

Продолжение таблицы 4

|  |  |
| --- | --- |
| order | порядок, в котором нужно вернуть список друзей. Допустимые значения:hints – сортировать по рейтингу, аналогично тому, как друзья сортируются в разделе Мои друзья (Это значение доступно только для Standalone–приложений с ключом доступа, полученным по схеме Implicit Flow.).random – возвращает друзей в случайном порядке.mobile – возвращает выше тех друзей, у которых установлены мобильные приложения.name – сортировать по имени. Данный тип сортировки работает медленно, так как сервер будет получать всех друзей а не только указанное количество count. (работает только при переданном параметре fields).По умолчанию список сортируется в порядке возрастания идентификаторов пользователей.  |
| list\_id | идентификатор списка друзей, полученный методом [friends.getLists](https://vk.com/dev/friends.getLists), друзей из которого необходимо получить. Данный параметр учитывается, только когда параметр user\_id равен идентификатору текущего пользователя. |
| count | количество друзей, которое нужно вернуть.  |
| offset | смещение, необходимое для выборки определенного подмножества друзей.  |
| fields | список дополнительных полей, которые необходимо вернуть. Доступные значения: nickname, domain, sex, bdate, city, country, timezone, photo\_50, photo\_100, photo\_200\_orig, has\_mobile, contacts, education, online, relation, last\_seen, status, can\_write\_private\_message, can\_see\_all\_posts, can\_post, universities  |
| name\_case | падеж для склонения имени и фамилии пользователя. Возможные значения: именительный – nom, родительный – gen, дательный – dat, винительный – acc, творительный – ins, предложный – abl. По умолчанию nom.  |

После успешного выполнения возвращает список идентификаторов (id) друзей пользователя, если параметр fields не использовался.
При использовании параметра fields возвращает список объектов [пользователей](https://vk.com/dev/objects/user), но не более 5000 [13].

# Алгоритм программы

Для выполнения программы обязательно задать такие параметры, как идентификатор анализируемого сообщества, с помощью которого будет осуществляться доступ к его стене записей, а так же временной порог, вплоть до которого стоит рассматривать записи. В качестве примера был выбран идентификатор группы «[Музыкальный центр SOROKA I школа Краснодар](https://vk.com/rocksoroka)», а так же порог времени – 17 ноября 2017 года, что на момент написания работы было эквивалентно одному месяцу. Таким образом, записи со стены группы, опубликованные ранее семнадцатого ноября, рассматриваться в конкретном примере не будут.

Основной цикл программы приведён на рисунке 1:



# Рисунок 1 – Основной цикл программы

Полный код программы описан в приложении А.

Основная функция main вызывается от переменной – идентификатор пользователя.

В переменной timeRange хранится дата в формате Unix epoch. UNIX–время или POSIX–время – способ кодирования времени, принятый в UNIX и других POSIX–совместимых операционных системах.

Поскольку в данной программе рассматриваются посты за ограниченный период времени, в этой переменной должна храниться дата, ограничивающая количество рассматриваемых записей со стены группы. Это значение является одним из критериев остановки цикла.

 Функция CreateFileOfPosts содержит в себе цикл сбора записей со страницы группы, а так же вывода полученной информации в файл csv. Запрос на получение информации осуществляется внутри функции при помощи метода wall.get [9]. Функция так же возвращает список полученных записей, и максимальное количество лайков среди них для упрощения дальнейшей работы с данными. Так как метод имеет ограничение на количество возвращаемых записей, равное 100, цикл не всегда завершится за одну итерацию. На каждой итерации, происходит обращение к базе данных vk.com для получения 100 записей заданной группы. Полученные записи помещаются в список, для дальнейшей обработки. Если в полученном списке имеется запись, сделанная позже определённого нами ранее временного порога, происходит прекращение формирования запросов на получение нового массива записей.

Сервер API ВКонтакте имеет частотные ограничения. К методам API ВКонтакте (за исключением методов из секций secure и ads) можно обращаться не чаще 3 раз в секунду. Если превысить частотное ограничение, сервер вернет ошибку с кодом 6: "Too many requests per second." [12]. Данное ограничение затрудняет работу при обработке большого количества данных.

После формирования списка записей со страницы группы, программа предлагает провести расчёт списка пользователей, которые вероятно будут заинтересованы в том, чтобы сталь участниками данной группы. Составление такого списка основывается на том, имеет ли пользователь друзей среди участников группы. Преследуя цели избежать полного перебора участников группы, а так же оптимизировать подсчёт, поиск вероятно заинтересованных пользователей будет производиться среди списков активных членов группы – ставящих лайки, и совершающих репосты записей со страницы группы.

Исходя из суждения, что активный пользователь заинтересован в контенте, поставляемого публичным сообществом, можно выдвинуть предположение, что среди его друзей вероятность встретить другого заинтересованного пользователя выше. Последнее основывается на суждении о том, что человек старается подружиться с людьми, обладающими схожими с ним интересами.

Формирование списка активных участников группы основывается на списке записей со страницы публичного сообщества, полученном ранее.

Функция CreateFileOfLikers создаёт список пользователей, установивших статус «Мне нравится» для каждой записи из списка записей со страницы группы, полученного ранее. Данные записываются в файл в формате csv для удобства их дальнейшего анализа. Пример полученных данных приведён в таблице 5.

Таблица 5 – Список пользователей, результат работы функции CreateFileOfLikers

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|   | IDS 1936 | Links 1936 | IDS 1941 | Links 1941 | IDS 1940 | Links 1940 |
| 0 | 311540463 | 415 | 28121128 | 75 | 18322913 | 339 |
| 1 | 235617309 | 166 | 235617309 | 166 | 115052889 | 24 |
| 2 | 4561539 | 183 | 115052889 | 24 | 429888841 | 79 |
| 3 | 5519980 | 570 | NaN | NaN | 75229320 | 298 |
| 4 | 126935664 | 559 | NaN | NaN | NaN | NaN |
| 5 | 14429320 | 408 | NaN | NaN | NaN | NaN |
| 6 | 75229320 | 298 | NaN | NaN | NaN | NaN |
| 7 | 21494510 | 1317 | NaN | NaN | NaN | NaN |

Данные формализуются в качестве таблицы, столбцы которой следует рассматривать попарно. Столбец IDS \*\*\*\* содержит список идентификаторов страниц пользователей, установившими статус «Мне нравится» под записью с идентификационным номером \*\*\*\*. Столбец Links \*\*\*\* содержит количество друзей пользователя, установившего статус «Мне нравится» под записью с идентификационным номером \*\*\*\*.

Между множеством данных об идентификаторах и множеством данных о количестве друзей имеет место отношение биекции: идентификатору пользователя из первого столбца обязательно соответствует количество друзей из второго столбца таким образом, что в табличном представлении элемент из первого столбца находится напротив элемента из второго столбца, при чём элемент из второго столбца такой один. Таким образом, исходя из таблицы 5, можно сделать вывод, что у пользователя с идентификационным номером 311540463 имеется 415 друзей.

Идентификационные номера пользователей, представленные в таблице, соответствуют идентификационным номерам страниц пользователей социальной сети «ВКонтакте».

В связи с тем, что на методы «API ВКонтакте» накладывается ограничение по количеству запросов, отправляемых на сервер vk.com в единицу времени, для удобства демонстрации работы программы была реализована возможность поиска заинтересованных пользователей, исходя из определённой записи со страницы публичного сообщества, и определённого пользователя, установившего под записью статус «Мне нравится». Такой подход позволяет значительно ускорить выполнение полного цикла представленной программы, сохраняя общую концепцию идеи. Данные, представленные в таблице 5, следует использовать для облегчения выбора определённой записи со страницы сообщества, а так же определённого пользователя, установившего под записью статус «Мне нравится».

В качестве демонстрации, выберем запись с идентификатором 1936 и пользователя с идентификатором 235617309. Данные передаются в функцию AnalysisOfLikers. Данная функция анализирует список друзей пользователей, поступивших на вход. Из множества друзей выделяется подмножество пользователей, не входящих в состав участников группы. Далее для каждого пользователя из полученного подмножества вычисляется количество друзей, числящихся в составе участников группы. На выходе функция формирует csv файл, содержащий идентификаторы пользователей, не состоящих в группе, а так же количество друзей для каждого из таких пользователей, состоящих в публичном сообществе.

Разработанная программа является универсальной. Её можно применять для анализа данных любого публичного сообщества, изменив основной параметр – идентификатор анализируемого сообщества. Кроме того, путём замены ограничителя по времени можно захватить более широкий, либо более узкий объём данных

# Анализ полученных результатов

После выполнения программы, был получен список 15 записей со стены публичного сообщества «Музыкальный центр SOROKA | школа Краснодар». Список формализован в качестве таблицы. Пример полученных данных демонстрируется в таблице 6.

Таблица 6 – Список записей со страницы группы «Музыкальный центр SOROKA | школа Краснодар».

|  |  |  |  |
| --- | --- | --- | --- |
| likes | reposts | text | postIDS |
| 8 | 2 | ☎ Запись на собеседование– 89615089838 | 1936 |
| 3 | 0 | Ежедневная кропотливая работа с каждым ребёнком. Фрагмент занятия по вокалу. Преподаватель – @daniilchikov #sorokafamilykrd <br>#урокивокала #урокивокалакраснодар #школамузыки #музыкальнаяшкола #rock40ka | 1941 |
| 4 | 0 | Ох и Платон! Радует Сороку на Восточке и родителей своей игрой и музыкальным вкусом | 1940 |
| 0 | 0 | «Я ж не Цезарь!!» Да, наши новосентябрьские ученики–новобранцы ещё только осваивают искусство и римские полководческие хитрости в музыке)) А мы прилагаем все усилия к их успеху. Вот как проходит рядовой урок в Сороке. Спасибо [id75229320|Алексей Селезнёв] за индивидуальный подход и авторские дидактические методы #metallica #entersandman #cover #урокигитары #школамузыки #sorokafamilykrd #музыкальнаясемья #rockstar #гитара #краснодар #урокигитарыкраснодар | 1939 |
| 1 | 0 | Обратная связь, скорее всего АлИны..)) или Светы?) #следствиеведутзнатоки #послание #школамузыки  | 1938 |
| 0 | 0 | Тем временем в Сороке вовсю идёт раннее музыкальное развитие детей от 3–х лет! Присоединяйтесь к нашему музыкальному волшебству ☎ 8(961)5089838 | 1935 |
| 2 | 0 | Эффект басиста! Или тот момент, когда решил отточить сценическое мастерство #басгитара #урокибаса #статьбасистом #школабаса #bassmen | 1934 |

Продолжение таблицы 6

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | Вечерние занятия по гитаре отменены!? Конечно! Все ученики вместе с родителями идут смотреть на своего преподавателя... Да, наш Максим играет в Vanilla Jam🤘🏻💪 #всенаконцерт в baklazhan\_krd | 1932 |
| 5 | 0 | На какой гитаре у нас можно научится играть?) ... #урокигитары #курсыгитары #школамузыки  | 1931 |
| 2 | 0 | Пока вы размышляете над «звонить/не звонить», в нашем новом звёздном классе одна за одной покоряются вокальные вершины! #урокивокала #курсывокала #жкпанорама #новыйфилиал  | 1928 |
| 2 | 0 | Aй–eee, ну вот и попались эти Vanilla! А знаете ли вы, что они выступали на одной сцене с легендарными Deep Purple, и песни именно Vanilla Jam стали музой для идеи открытия Сороки 5 лет назад??⚡️ Хотите разделить «тогдашнее» наше настроение и лучше понять концепцию школы – посетите!💪⚡️ #крутейшийконцерт #vanillajam #живойзвук #live | 1927 |
| 3 | 0 | Спасибо всей #sorokafamilykrd за вечер, ресторану baklazhan\_krd – за атмосферу и яства! Коротаем вечер в поиске «красивых себя» на снимках vklybe – http://vklybe.tv/krasnodar/ru/photos/event/id/264406 | 1926 |
| 1 | 0 | И уже очень скоро состоится вечер встречи одноклубников «SorokaFamilyKRD». Приглашаем учеников всех возрастов и их друзей!<br>Вход, конечно же, бесплатный. В программе: церемония посвящения в музыканты, концерт педагогов и учеников, конкурсные и творческие мероприятия для детей и взрослых, призы и подарки, фотозона.<br>Дата: 23 ноября (четверг).<br>Место: ресторан «Баклажан», Атарбекова 45/1.<br>Регистрация участников: 18:30–19:00.<br>Завершение мероприятия: 21:30. | 1925 |
| 2 | 0 | Восточные сказки... Элина усвоила основы сольной импровизации и теперь ей не страшен любой ритм #соло #нескучныеурокигитары #rock40ka [id75229320|Алексей Селезнёв] | 1924 |
| 1 | 0 | #музыкальнаясемьясорока в любой точке мира найдёт чем себя занять🎻 Классические инструменты нам тоже по плечу🥁🎹🎤 <br>Доброе субботнее! | 1923 |

По исходной таблице сложно сделать какие–либо выводы об актуальности тех или иных записей. Поэтому необходимо выполнить сортировку и посмотреть, какие записи набрали наибольшее количество лайков или репостов.

# Анализ по количеству лайков

Произведем сортировку по количеству лайков (по убыванию). Результат сортировки представлен в таблице 7.

Таблица 7 – Записи, отсортированные по количеству лайков

|  |  |  |  |
| --- | --- | --- | --- |
| likes | reposts | text | postIDS |
| 8 | 2 | ☎ Запись на собеседование– 89615089838 | 1936 |
| 5 | 0 | На какой гитаре у нас можно научится играть?) ... #урокигитары #курсыгитары #школамузыки  | 1931 |
| 4 | 0 | Ох и Платон! Радует Сороку на Восточке и родителей своей игрой и музыкальным вкусом | 1940 |
| 3 | 0 | Ежедневная кропотливая работа с каждым ребёнком. Фрагмент занятия по вокалу. Преподаватель – @daniilchikov #sorokafamilykrd <br>#урокивокала #урокивокалакраснодар #школамузыки #музыкальнаяшкола #rock40ka | 1941 |
| 3 | 0 | Спасибо всей #sorokafamilykrd за вечер, ресторану baklazhan\_krd – за атмосферу и яства! Коротаем вечер в поиске «красивых себя» на снимках vklybe – http://vklybe.tv/krasnodar/ru/photos/event/id/264406 | 1926 |
| 2 | 0 | Эффект басиста! Или тот момент, когда решил отточить сценическое мастерство #басгитара #урокибаса #статьбасистом #школабаса #bassmen | 1934 |
| 2 | 0 | Пока вы размышляете над «звонить/не звонить», в нашем новом звёздном классе одна за одной покоряются вокальные вершины! #урокивокала #курсывокала #жкпанорама #новыйфилиал  | 1928 |

Продолжение таблицы 7

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 0 | Aй–eee, ну вот и попались эти Vanilla! А знаете ли вы, что они выступали на одной сцене с легендарными Deep Purple, и песни именно Vanilla Jam стали музой для идеи открытия Сороки 5 лет назад??⚡️ Хотите разделить «тогдашнее» наше настроение и лучше понять концепцию школы – посетите! #крутейшийконцерт #vanillajam #живойзвук #live | 1927 |
| 2 | 0 | Восточные сказки... Элина усвоила основы сольной импровизации и теперь ей не страшен любой ритм #соло #нескучныеурокигитары #rock40ka [id75229320|Алексей Селезнёв] | 1924 |
| 1 | 0 | Обратная связь, скорее всего АлИны..)) или Светы?) #следствиеведутзнатоки #послание #школамузыки  | 1938 |
| 1 | 0 | Вечерние занятия по гитаре отменены!? Конечно! Все ученики вместе с родителями идут смотреть на своего преподавателя... Да, наш Максим играет в Vanilla Jam #всенаконцерт в baklazhan\_krd | 1932 |
| 1 | 0 | И уже очень скоро состоится вечер встречи одноклубников «SorokaFamilyKRD». Приглашаем учеников всех возрастов и их друзей!<br>Вход, конечно же, бесплатный. В программе: церемония посвящения в музыканты, концерт педагогов и учеников, конкурсные и творческие мероприятия для детей и взрослых, призы и подарки, фотозона.<br>Дата: 23 ноября (четверг).<br>Место: ресторан «Баклажан», Атарбекова 45/1.<br>Регистрация участников: 18:30–19:00.<br>Завершение мероприятия: 21:30. | 1925 |
| 1 | 0 | #музыкальнаясемьясорока в любой точке мира найдёт чем себя занять Классические инструменты нам тоже по плечу <br>Доброе субботнее! | 1923 |
| 0 | 0 | «Я ж не Цезарь!!» Да, наши новосентябрьские ученики–новобранцы ещё только осваивают искусство и римские полководческие хитрости в музыке)) А мы прилагаем все усилия к их успеху. Вот как проходит рядовой урок в Сороке. Спасибо [id75229320|Алексей Селезнёв] за индивидуальный подход и авторские дидактические методы #metallica #entersandman #cover #урокигитары #школамузыки #sorokafamilykrd #музыкальнаясемья #rockstar #гитара #краснодар #урокигитарыкраснодар | 1939 |

# Анализ по количеству репостов

Чаще всего, пользователь совершает репост записи в том случае, когда информация, изложенная в записи, достаточно важна для конкретного человека. Вследствие чего, не стоит пренебрегать статистическими показателями репостов.

Произведем сортировку по количеству репостов. Результаты представлены в таблице 8.

Таблица 8 – Записи, отсортированные по количеству репостов

|  |  |  |  |
| --- | --- | --- | --- |
| likes | reposts | text | postIDS |
| 8 | 2 | ☎ Запись на собеседование– 89615089838 | 1936 |
| 5 | 0 | На какой гитаре у нас можно научится играть?) ... #урокигитары #курсыгитары #школамузыки  | 1931 |
| 4 | 0 | Ох и Платон! Радует Сороку на Восточке и родителей своей игрой и музыкальным вкусом | 1940 |
| 3 | 0 | Ежедневная кропотливая работа с каждым ребёнком. Фрагмент занятия по вокалу. Преподаватель – @daniilchikov #sorokafamilykrd <br>#урокивокала #урокивокалакраснодар #школамузыки #музыкальнаяшкола #rock40ka | 1941 |
| 3 | 0 | Спасибо всей #sorokafamilykrd за вечер, ресторану baklazhan\_krd – за атмосферу и яства! Коротаем вечер в поиске «красивых себя» на снимках vklybe – http://vklybe.tv/krasnodar/ru/photos/event/id/264406 | 1926 |
| 2 | 0 | Эффект басиста! Или тот момент, когда решил отточить сценическое мастерство #басгитара #урокибаса #статьбасистом #школабаса #bassmen | 1934 |
| 2 | 0 | Пока вы размышляете над «звонить/не звонить», в нашем новом звёздном классе одна за одной покоряются вокальные вершины! #урокивокала #курсывокала #жкпанорама #новыйфилиал  | 1928 |

Продолжение таблицы 8

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 0 | Aй–eee, ну вот и попались эти Vanilla! А знаете ли вы, что они выступали на одной сцене с легендарными Deep Purple, и песни именно Vanilla Jam стали музой для идеи открытия Сороки 5 лет назад?? Хотите разделить «тогдашнее» наше настроение и лучше понять концепцию школы – посетите! #крутейшийконцерт #vanillajam #живойзвук #live | 1927 |
| 2 | 0 | Восточные сказки... Элина усвоила основы сольной импровизации и теперь ей не страшен любой ритм #соло #нескучныеурокигитары #rock40ka [id75229320|Алексей Селезнёв] | 1924 |
| 1 | 0 | Обратная связь, скорее всего АлИны..)) или Светы?) #следствиеведутзнатоки #послание #школамузыки  | 1938 |
| 1 | 0 | Вечерние занятия по гитаре отменены!? Конечно! Все ученики вместе с родителями идут смотреть на своего преподавателя... Да, наш Максим играет в Vanilla Jam #всенаконцерт в baklazhan\_krd | 1932 |
| 1 | 0 | И уже очень скоро состоится вечер встречи одноклубников «SorokaFamilyKRD». Приглашаем учеников всех возрастов и их друзей!<br>Вход, конечно же, бесплатный. В программе: церемония посвящения в музыканты, концерт педагогов и учеников, конкурсные и творческие мероприятия для детей и взрослых, призы и подарки, фотозона.<br>Дата: 23 ноября (четверг).<br>Место: ресторан «Баклажан», Атарбекова 45/1.<br>Регистрация участников: 18:30–19:00.<br>Завершение мероприятия: 21:30. | 1925 |
| 1 | 0 | #музыкальнаясемьясорока в любой точке мира найдёт чем себя занять Классические инструменты нам тоже по плечу <br>Доброе субботнее! | 1923 |
| 0 | 0 | «Я ж не Цезарь!!» Да, наши новосентябрьские ученики–новобранцы ещё только осваивают искусство и римские полководческие хитрости в музыке)) А мы прилагаем все усилия к их успеху. Вот как проходит рядовой урок в Сороке. Спасибо [id75229320|Алексей Селезнёв] за индивидуальный подход и авторские дидактические методы💪 #metallica #entersandman #cover #урокигитары #школамузыки #sorokafamilykrd #музыкальнаясемья #rockstar #гитара #краснодар #урокигитарыкраснодар | 1939 |

Как можно увидеть из приведённых аналитических данных, записи на стене сообщества не пользуются большой популярностью среди участников группы. Причиной тому может служить не качественный контент, малое количество участников в группе в общем, либо малое количество активных участников в группе. На основе уже имеющихся данных, проведём анализ социальной сети на наличие возможных пользователей, которые могут быть заинтересованы в уже имеющемся контенте группы.

# Поиск заинтересованных пользователей

Как было указано в описании алгоритма, для удобства представления работы программы, проведём анализ на основе записи с идентификатором 1936 и пользователя с идентификатором 235617309.

В результате вычислений, был сформирован список пользователей, не имеющих членства в группе, однако имеющих друзей, являющихся членами публичного сообщества. На данном примере был получен список 98 пользователей, формализованный в виде таблицы, фрагмент которой представлен в таблице 9.

Таблица 9 – Список заинтересованных пользователей

|  |  |  |
| --- | --- | --- |
|  | IDS | Links |
| 0 | 103045 | 24 |
| 1 | 411925 | 17 |
| 2 | 589685 | 14 |
| 3 | 654427 | 4 |
| 4 | 761180 | 40 |
| 5 | 762741 | 25 |
| 6 | 1472860 | 3 |
| 7 | 1528600 | 23 |
| 8 | 1887041 | 24 |
| 9 | 2057285 | 9 |
| 10 | 2118053 | 67 |
| 11 | 2312030 | 89 |
| 12 | 2358338 | 69 |
| 13 | 2574501 | 156 |
| … | … | … |
| 98 | 438365872 | 5 |

На данном этапе идентификаторы пользователей расположены неупорядоченно. Для более ясного представления о новых потенциальных участниках группы, упорядочим список по количеству друзей пользователей, уже имеющих членство в группе. Фрагмент упорядоченного списка показан в таблице 10.

Таблица 10 – Упорядоченный по количеству друзей список потенциальных участников

|  |  |  |
| --- | --- | --- |
|  | IDS | Links |
| 0 | 2574501 | 156 |
| 1 | 253348499 | 121 |
| 2 | 179519559 | 106 |
| 3 | 2312030 | 89 |
| 4 | 36512561 | 86 |
| 5 | 151590823 | 73 |
| 6 | 2358338 | 69 |
| 7 | 7726249 | 68 |
| 8 | 2118053 | 67 |
| 9 | 7242215 | 65 |
| 10 | 210647509 | 63 |
| 11 | 149120909 | 59 |
| 12 | 59650489 | 58 |
| 13 | 199364530 | 58 |
| … | … | … |
| 98 | 421017601 | 1 |

Исходя из упорядоченного списка, можно сделать вывод о пользователях, которые вероятнее всего могут стать новыми членами публичного сообщества. Основываясь на суждении о том, что человек старается подружиться с людьми, обладающими схожими с ним интересами, можно выдвинуть предположение, что пользователь с идентификатором 2574501 вероятнее всего будет заинтересован в членстве в публичном сообществе, так как количество его друзей–участников сообщества больше всего.

# ЗАКЛЮЧЕНИЕ

В ходе работы была составлена программа, позволяющая получать информацию из базы данных vk.com, выделять необходимые для анализа поля и записывать их в удобный для дальнейшей работы табличный вид.

На основании полученных данных, был произведён анализ активности участников публичного сообщества «Музыкальный центр SOROKA | школа Краснодар». Так же был произведён поиск пользователей социальной сети «ВКонтакте», за счёт которых возможно было бы увеличить посещаемость группы.

Очевидно, что за временные рамки, определённые при анализе сообщества, активность пользователей крайне низка, что должно послужить мотивацией для более тщательного анализа потребностей пользователей, а так же для расширения аудитории.

Проанализировав полученные результаты, был составлен список пользователей, за счёт которых, с некой долей вероятности, можно было бы увеличить количество подписчиков сообщества

Если правильно применить выводы данного анализа, то они могут помочь в развитии группы и существенно увеличить количество участников сообщества, от чего напрямую зависит материальная прибыль, увеличение которой есть прямая цель создания данного сообщества.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Халилов Д. Маркетинг в социальных сетях: Манн, Иванов и Фербер; Москва; 2013; 240 c.
2. Знакомство с API ВКонтакте URL: <https://vk.com/dev/first_guide> (дата обращения: 27.12.2017)
3. Программирование на Python: Часть 1 URL: [https://www.ibm.com/developerworks/ru/library/l–python\_part\_1/index.html](https://www.ibm.com/developerworks/ru/library/l-python_part_1/index.html) (дата обращения: 27.12.2017)
4. Библиотека vk для работы с VK API URL: <https://habrahabr.ru/post/319178> (дата обращения: 27.12.2017)
5. Reading and Writing CSV Files in Python URL: [http://stackabuse.com/reading–and–writing–csv–files–in–python](http://stackabuse.com/reading-and-writing-csv-files-in-python) (дата обращения: 27.12.2017)
6. Библиотека Pandas URL: <https://pandas.pydata.org> (дата обращения: 27.12.2017)
7. Модуль datetime URL: [https://pythonworld.ru/moduli/modul–datetime.html](https://pythonworld.ru/moduli/modul-datetime.html) (дата обращения: 27.12.2017)
8. [Обработка Excel файлов](https://www.google.ru/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&uact=8&ved=0ahUKEwibs-e9uZvYAhVBy6QKHcx0BV4QFghHMAM&url=https%3A%2F%2Fhabrahabr.ru%2Fpost%2F99923%2F&usg=AOvVaw2JSlvzj43jXL2aocgf_hB7) URL: <https://habrahabr.ru/post/99923/> (дата обращения: 27.12.2017)
9. Метод wall.get URL: <https://vk.com/dev/wall.get> (дата обращения: 27.12.2017)
10. Метод groups.isMember URL: <https://vk.com/dev/groups.isMember> (дата обращения: 27.12.2017)
11. Метод likes.getList URL: <https://vk.com/dev/likes.getList> (дата обращения: 27.12.2017)
12. Выполнение запросов к API ВКонтакте URL: <https://vk.com/dev/api_requests> (дата обращения: 27.12.2017)
13. Метод friends.get URL: <https://vk.com/dev/friends.get> (дата обращения: 27.12.2017)

# ПРИЛОЖЕНИЕ А

Листинг программы

import vk

import time

import pandas as pd

import xlrd

import csv

from datetime import datetime, timedelta

session = vk.Session(access\_token = 'ххх')

api = vk.API(session)

class User:

 ID = ' '

 links = 0

def GetWall (ID,c,o):

 try:

 time.sleep(0.5)

 return api.wall.get(owner\_id = ID, count = c, offset = o)

 except Exception as e:

 print (str(e))

 print(';;;')

 error = 'error'

 return error

def GetPost (post):

 try:

 post\_id = post['id']

 except:

 post\_id = 0

 try:

 likes = post['likes']['count']

 except:

 likes = 0

 try:

 reposts = post['reposts']['count']

 except:

 reposts = 0

 try:

 text = post['text']

 except:

 text = 'EMPTY'

 post = {

 'id': post\_id,

 'likes': likes,

 'reposts': reposts,

 'text': text

 }

 return post

def GetKlasterList():

 file = 'Graph1.xlsx'

 xl = pd.ExcelFile(file)

 df1 = xl.parse('Group Vertices')

 klasters =[]

 count = 0

 offset = 0

 indList = –1

 df = df1.loc[0]

 determinator = df['Group']

 for ind in df1.index:

 df = df1.loc[ind]

 if df['Group'] == determinator:

 count += 1

 else:

 indList += 1

 klasters.append(list())

 for ind1 in range (offset, count + offset):

 tempDf = df1.loc[ind1]

 klasters[indList].append(str(tempDf['Vertex']))

 offset += count

 count = 1

 determinator = df['Group']

 return klasters

def GetListOfFriendsInGroup (klaster, GROUP):

 maxCount = 500

 newListInGroup = []

 newListNotInGroup = []

 result = []

 indList = –1

 indList += 1

 length = len(klaster)

 offset = 0

 limit = 0

 if length // maxCount >= 1 and length % maxCount > –1:

 countRequests = (length // maxCount) + 1

 else:

 if length // maxCount >= 1 and length % maxCount == 0:

 countRequests = (length / maxCount)

 else:

 countRequests = 1

 for i in range(countRequests):

 IDS = []

 if length – maxCount > 0:

 limit += 500

 length –= maxCount

 else:

 limit += length

 for j in range(offset, limit):

 IDS.append(str(klaster[j]))

 time.sleep(1)

 tempList = api.groups.isMember(group\_id = str(–1 \* int(GROUP)), user\_ids = ','.join(IDS))

 offset += 500

 for userr in tempList:

 if userr['member'] == 1:

 newListInGroup.append(str(userr['user\_id']))

 if userr['member'] == 0:

 newListNotInGroup.append(str(userr['user\_id']))

 for i in range(2):

 result.append(list())

 result[0] = list(newListInGroup)

 result[1] = list(newListNotInGroup)

 return result

def WriteCSVFile(name, post, create):

 if create is True:

 data = ["likes;reposts;text;postIDS".split(";")]

 with open(name,'w', encoding="utf–8") as file:

 writer = csv.writer(file, delimiter=';')

 for line in data:

 writer.writerow((line))

 else:

 with open(name,'a', encoding="utf–8") as file:

 writer = csv.writer(file, delimiter=';')

 writer.writerow((post['likes'], post['reposts'], post['text'], post['id']))

def CreateFileOfPosts (GROUP, timeRange):

 boo = 1

 offset = 0

 likes = 0

 listOfPosts = []

 result = []

 while boo == 1:

 wall = GetWall(GROUP, 100, offset)

 if wall != 'error':

 for post in wall:

 if type(post) is int:

 offset = offset + 100

 else:

 if int(post['date']) < timeRange:

 boo = 0

 break

 else:

 if post['likes']['count'] > likes:

 likes = post['likes']['count']

 listOfPosts.append(post)

 WriteCSVFile('posts.csv', 'Empty', True)

 for post in listOfPosts:

 DataOfPost = GetPost(post)

 WriteCSVFile('posts.csv', DataOfPost, False)

 for i in range(2):

 result.append(list())

 result[0] = list(listOfPosts)

 result[1] = str(likes)

 return result

def CreateFileOfLikers (listOfPosts, GROUP, likes):

 df = pd.DataFrame({})

 for post in listOfPosts:

 listOfUsers = []

 listOfIds = []

 listOfLinks = []

 headline = (str(post['id']))

 likesObj = api.likes.getList(type = 'post', owner\_id = GROUP, item\_id = post['id'], extended = 1, count = 1000) # лайков может быть больше тыщи, доделай

 likingUsers = likesObj['items']

 localCount = 0

 for user in likingUsers:

 localCount1 = 0

 listOfUsers.append(User())

 listOfUsers[localCount].ID = str(user['uid'])

 time.sleep(0.5)

 try:

 friendsOfUser = api.friends.get(user\_id = user['uid'])

 except Exception as e:

 continue

 for friend in friendsOfUser:

 localCount1 += 1

 listOfUsers[localCount].links = int(localCount1)

 localCount += 1

 for user in listOfUsers:

 listOfIds.append(str(user.ID))

 listOfLinks.append(str(user.links))

 if len(listOfIds) > 0:

 if len(listOfIds) < int(likes):

 difference = int(likes) – len(listOfIds)

 for i in range(difference):

 listOfIds.append('NaN')

 listOfLinks.append('NaN')

 df['IDS ' + str(headline)] = listOfIds

 df['Links ' + str(headline)] = listOfLinks

 df.to\_csv('listOfLikers.csv', sep = ';', encoding = 'utf–8')

def AnalysisOfLikers (listOfPosts, GROUP, GAnal, UAnal, likes):

 df = pd.DataFrame({})

 localCount = –1

 if GAnal != '0':

 listOfPosts = list([])

 listOfPosts.append(str(GAnal))

 for post in listOfPosts:

 resultList = []

 listOfUsers = []

 listOfIds = []

 if UAnal != '0':

 likingUsers = [UAnal]

 else:

 likesObj = api.likes.getList(type = 'post', owner\_id = GROUP, item\_id = post['id'], extended = 1, count = 1000) # лайков может быть больше тыщи, доделай

 likingUsers = likesObj['items']

 for user in likingUsers:

 try:

 if UAnal != '0':

 friendsOfUser = api.friends.get(user\_id = int(user))

 else:

 friendsOfUser = api.friends.get(user\_id = user['uid'])

 except Exception as e:

 print(str(e))

 continue

 listOfIds = list([])

 for friend in friendsOfUser:

 listOfIds.append(str(friend))

 fileOfFriends = GetListOfFriendsInGroup(listOfIds, GROUP)

 listOfIds = list([])

 listOfIds = list(fileOfFriends[1])

 for idd in listOfIds:

 try:

 friendsOfUser = api.friends.get(user\_id = idd)

 except Exception as e:

 print(str(e))

 continue

 listOfUsers = list([])

 for friend in friendsOfUser:

 listOfUsers.append(str(friend))

 if len(listOfUsers) < 5000:

 fileOfFriends = GetListOfFriendsInGroup(listOfUsers, GROUP)

 listOfUsers = list(fileOfFriends[0])

 if len(listOfUsers) < 5000:

 if len(listOfUsers) > 0:

 localCount += 1

 resultList.append(User())

 resultList[localCount].ID = str(idd)

 resultList[localCount].links = int(len(listOfUsers))

 listOfIds = list([])

 listOfLinks = list([])

 for user in resultList:

 listOfIds.append(str(user.ID))

 listOfLinks.append(str(user.links))

 df['IDS'] = listOfIds

 df['Links'] = listOfLinks

 df.to\_csv('listOfPotential.csv', sep = ';', encoding = 'utf–8')

def main (GROUP):

 timeRange = int(1510876800)

 FileOfPosts = CreateFileOfPosts(GROUP, timeRange)

 print('Произвести дополнительный расчёт 0/1?...')

 yes = input()

 if yes == '1':

 ListOfPosts = FileOfPosts[0]

 likes = FileOfPosts[1]

 CreateFileOfLikers(ListOfPosts, GROUP, likes)

 print('Введите ID записи или ноль, для анализа всех записей...')

 GAnal = input()

 print('Введите ID пользователя или ноль, для анализа всех пользователей...')

 UAnal = input()

 AnalysisOfLikers(ListOfPosts, GROUP, GAnal, UAnal, likes)

main('–68317828')