ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА
(РОСАВИАЦИЯ)

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ» (МГТУ ГА)

Кафедра основ радиотехники и защиты информации

С.П. Матыюк

СИСТЕМЫ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОГО МОНИТОРИНГА

Учебно-методическое пособие
по выполнению лабораторных работ № 1–4

для студентов IV курса
специальности 10.05.02
очной формы обучения

Москва
ИД Академии Жуковского
2021

УДК 004.9
ББК 001.8
М34

Рецензент:
Петров В.И. – канд. техн. наук, доцент

Матыюк С.П.
М34 Системы информационно-аналитического мониторинга [Текст] : учебно-методическое пособие по выполнению лабораторных работ № 1–4 / С.П. Матыюк. – М.: ИД Академии Жуковского, 2021. – 24 с.

Данное учебно-методическое пособие издается в соответствии с рабочей программой учебной дисциплины Б1.В.ДВ.8.1 «Интеллектуальные информационные системы» по учебному плану для студентов IV курса специальности 10.05.02 очной формы обучения.

Рассмотрено и одобрено на заседаниях кафедры 26.02.2021 г. и методического совета 23.03.2021 г.

УДК 004.9
ББК 001.8

В авторской редакции

Подписано в печать 13.09.2021 г.
Формат 60х84/16 Печ. л. 1,5 Усл. печ. л. 1,395
Заказ № 793/0616-УМП06 Тираж 40 экз.

Московский государственный технический университет ГА
125993, Москва, Кронштадтский бульвар, д. 20

Издательский дом Академии имени Н. Е. Жуковского
125167, Москва, 8-го Марта 4-я ул., д. 6А
Тел.: (495) 973-45-68
E-mail: zakaz@itsbook.ru

ЛАБОРАТОНАЯ РАБОТА № 1

Создание аналитического отчета OLAP

1. **Цель работы** – получение студентами навыков оперативного анализа данных.

2. **Краткие теоретические сведения**

Сводная таблица — это плоская или объемная (состоящая из нескольких слоев или страниц) прямоугольная таблица, позволяющая просуммировать или иным образом подытожить большие объемы данных из расположенного в другом месте рабочей книги исходного списка данных. В качестве исходных данных могут использоваться данные из другой книги Excel, другой сводной таблицы, из запроса к внешней базе данных.

Для подведения итогов можно выбрать подходящую функцию сведения или иной метод вычисления. Эти вычисления производятся для одного или нескольких полей исходного списка, объявленных как поля данных. При этом некоторые другие поля исходного списка используются для группировки данных в строках и столбцах сводной таблицы.

Консолидация данных

Консолидация данных выполняется в том случае, когда надо подытожить данные, расположенные в разных областях таблицы. Подлежащие консолидации данные могут располагаться на одном рабочем листе, на разных листах, в разных книгах.

При консолидации рабочих листов происходит обобщение однородных данных. Консолидация — это не только суммирование. Можно вычислять такие статистические величины, как среднее, стандартное отклонение, число величин.

Исходные для консолидации рабочие листы не обязаны иметь абсолютно одинаковую структуру. Можно объединять те ячейки, которые имеют одно и то же относительное расположение. Консолидация по расположению используется редко, так как требует абсолютно одинаковую структуру исходных диапазонов данных. Можно объединять те ячейки, которые принадлежат столбцам или строкам с одинаковыми заголовками. При этом в выделяемый диапазон следует включать заголовки строк или столбцов. Таким образом, достигается большая гибкость.

3. **Краткое описание используемого оборудования**

Процессор Intel(R) Atom(TM) CPU D525 @ 1.80 GHz, ОЗУ 2.00 Гб.

4. Порядок выполнения работы

4.1 Создание аналитического документа с использованием MS Excel

4.1.1. Создайте аналитический документ о финансовой работе организации (база данных):

- страховая компания;
- телекоммуникационная компания,
- авиакомпания.

4.1.2. Рассмотрите подробно все то, что приносит доход вашей компании. Имейте в виду что ваша компания имеет несколько филиалов (как минимум 3);

4.1.3. Для наполнения базы данных используйте материалы ПЗ № 3;

4.1.4. В качестве примера для текущей ЛР рассмотрена работа автомобильного салона.

Таблица 1 Пример

Продавец	Марка автомобиля	Дата выпуска автомобиля	Оборот	Дата

4.1.5. Заполните сначала поле Дата, временной период — год (01.01.2021-31.12.2022). Временной интервал - одна неделя.

4.1.6. Продавцов в салоне 7, введите фамилии, с помощью Автозаполнения скопируйте их на весь год.

4.1.7. Продаются автомобили 5 марок, введите названия марок, не используйте в названиях цифры, с помощью Автозаполнения скопируйте их на весь квартал.

4.1.8. Введите 4 года выпуска: 2019, 2020, 2021, 2022, с помощью Автозаполнения и клавиши CTRL скопируйте их на весь год.

4.1.9. Присвойте листу с таблицей имя Данные.

4.1.10. Упорядочите таблицу одновременно по полю Продавец и по полю Марка (Данные — сортировать).

4.2. Создание и анализ сводных таблиц

4.2.1. Создайте сводную таблицу, для этого щелкните в любой ячейке таблицы на листе Данные и возьмите команду Данные - Сводная таблица. Запустится мастер сводных таблиц.

4.2.2. На первом шаге определите источник данных — в списке или базе данных Microsoft Excel и вид отчета — сводная таблица.

4.2.3. На втором шаге указывается диапазон исходных данных — убедитесь, что вся таблица на листе Данные выделена, если нет, то выделите ее.

4.2.4. На следующем шаге поместите таблицу в новый лист и Готово. На

листе появится макет сводной таблицы и панель инструментов Сводные таблицы.

4.2.5. Перетащите в область данных кнопку Оборот с панели инструментов Сводные таблицы, в область строк — кнопку Марка, в область столбцов — кнопку Продавец, в область страниц — кнопку Дата. Просмотрите получившийся отчет.

4.2.6. Сделайте сводную диаграмму на основе полученного отчета, открыв на панели кнопку Сводная таблица и выбрав команду Сводная диаграмма. Проанализируйте диаграмму, посмотрите возможности изменения представления данных на диаграмме. Обратите внимание, как при этом ведет себя сводная таблица.

4.2.7. Сделайте три разных отчета (каждый на отдельном листе), проиллюстрировав их соответствующей диаграммой. Введите для каждого свое название. Сделайте соответствующий вывод по анализу данных.

4.3. Консолидация данных

4.3.1. Создайте таблицы о работе филиалов автосалона на следующих трех листах, задав им имена Филиал 1, Филиал 2, Филиал 3. В первом филиале работают три продавца, во втором и третьем по два. Перенесите данные о работе продавцов за квартал с листа Данные на вновь создаваемые листы.

4.3.2. Создайте сводные таблицы по результатам работы всех филиалов и представьте отчеты о количестве проданных автомобилей каждой марки в каждом филиале (в области строк — Марка, в области данных — Марка).

4.3.3. Обобщите данные о продаже по всем филиалам, т. е. консолидируйте данные. Для этого перейдите на новый лист и выделите ячейку, которая послужит началом диапазона ячеек с итогами.

4.3.4. Возьмите команду Консолидация из меню Данные, откроется одноименное окно, в котором нужно указать адреса консолидируемых данных и выбрать необходимую функцию.

4.3.5. Выберите функцию Сумма.

4.3.6. В поле Ссылка задайте диапазон данных первого филиала (выделите сводную таблицу по марке), нажмите кнопку Добавить. Аналогично добавьте ссылки на данные других филиалов. В результате в поле Список диапазонов должны появиться ссылки на данные трех филиалов.

4.3.7. Установите галочку в окошке Использовать в качестве имен — Значения левого столбца, поставьте галочку в окошке Создавать связи с исходными данными.

4.3.8. Нажмите ОК. Появится таблица, содержащая обобщенный результат.

4.3.9. Оформите таблицу, введите заголовок отчета.

4.3.10. Посмотрите данные консолидированной таблицы. Появилась структура, в которой вы можете просмотреть количество консолидированных данных.

5. Содержание отчета

5.1. Результаты вычислений;

5.2. Анализ полученных результатов и выводы.

6. Литература

6.1. Информационно-аналитические системы. Учебник для вузов. Т.В. Алексеева, Ю.В. Америди, В.В. Дюк - М.: Московский финансово-промышленный университет «Синергия» - Университетская серия, 2013.

6.2. Информационно-аналитические системы. Основы проектирования и применения. Учебное пособие, руководство, практикум. В.С. Белов. - М.: Московский государственный университет экономики, статистики и информатики, 2005.

ЛАБОРАТОНАЯ РАБОТА № 2

Интеллектуальный анализ данных с помощью функции скользящего среднего

1. **Цель работы** – получение студентами навыков интеллектуального анализа данных

2. Краткие теоретические сведения

Прогнозирование с помощью функции скользящего среднего программы Excel.

Прогнозирование позволяет находить в исторической информации, представленной в виде временных рядов, такие шаблоны, которые отражают динамику поведения целевых показателей, и с определенной долей вероятности предсказывать значение целевых показателей в будущем.

Скользящие средние сглаживают колебания изучаемой величины с помощью Усреднения по некоторому историческому периоду. Достоинством данного анализа является возможность визуально отсечь малые флуктуации и четко увидеть направление движения.

Недостатком скользящих средних является запаздывание усредненных значений по отношению к изменению изучаемой величины. Отсюда следует, что чем больше период усреднения, тем более важные сигналы они дают, но вместе с тем больше опаздывают. Ценность скользящего среднего — в том, что оно дает направление общего движения.

3. Краткое описание используемого оборудования

Процессор Intel(R) Atom(TM) CPU D525 @ 1.80 GHz, ОЗУ 2.00 Гб.

4. Порядок выполнения работы

4.1 Пример

У вас есть отчет о ежедневном количестве звонков с жалобами на конкретный программный продукт за последние десять дней.

Таблица 1. Пример

День	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Количество звонков	**10**	**11**	**10**	**12**	**13**	**13**	**13**	**10**	**16**	**17**

Чтобы понять, существует ли какая-либо определенная тенденция поступления жалоб, создайте на основе средних данных о полученных звонках скользящее среднее. Воспользуйтесь трехдневным скользящим средним, так как скользящее среднее за меньший период может не отразить тенденцию, а за больший период слишком сгладить ее.

4.1 Прямое введение формулы

4.1.1. Введите исходные данные в первые два столбца (A и B) таблицы без заголовка;

4.1.2. Чтобы получить скользящее среднее, введите в ячейку C4 следующую формулу: =срзнач(A1 A3) (Вставка — функция — статистические — срзнач);

4.1.3. Затем с помощью средств автозаполнения скопируйте эту формулу в ячейки C5:C10.

4.1.4. Постройте графики изменения данных и скользящего среднего.

4.2 Использование надстроек скользящего среднего

4.2.1. Скопируйте входные данные на второй лист книги.

4.2.2. В меню Сервис выберите команду Анализ данных (если такой команды нет, включите ее: Сервис-Надстройки-Пакет анализа).

4.2.3. В появившемся окне выберите команду Скользящее среднее.

4.2.4. В поле Входной интервал введите данные о вашей базовой линии (укажите диапазон входных данных).

4.2.5. В поле Интервал введите количество дней, которые хотите включить в подсчет скользящего среднего.

4.2.6. В поле Выходной интервал введите адрес ячейки, с которой хотите начать вывод.

4.2.7. Поставьте значок Вывод графика.

4.2.8. Нажмите ОК. (Значок Н/Д означает — не хватает данных для подсчета среднего).

Этот способ имеет недостаток — прогноз создается на один временной период раньше.

4.3 Составление прогноза скользящего среднего с помощью диаграмм

4.3.1. Скопируйте входные данные на третий лист книги.

4.3.2. Выделите данные своей базовой линии.

4.3.3. Запустите Мастер диаграмм, выберите тип диаграммы — График.

4.4.4. Вставьте диаграмму на текущий лист.

4.4.5. Щелкните правой кнопкой на ряде данных диаграммы и из появившегося контекстного меню выберите команду Добавить линию тренда.

4.4.6. В появившемся окне на вкладке Тип выберите тип Линейная фильтрация.

4.4.7. В окне Точки установите период вычисления скользящего среднего — количество дней.

4.4.8. Нажмите ОК.

Сравните результаты прогнозирования, полученные разными способами.

4.4 Задание

4.3.1. Используйте исходные данные из лабораторной работы № 1;

4.3.2. По образцу выполните работу;

4.3.3. Одним параметром будет время (интервал - неделя), другим параметром – денежный оборот вашей компании (интервал - неделя).

5. Содержание отчета

5.1. Результаты вычислений;

5.2. Анализ полученных результатов и выводы.

6. Литература

6.1. Информационно-аналитические системы. Учебник для вузов. Т.В. Алексеева, Ю.В. Америди, В.В. Дюк - М.: Московский финансово-промышленный университет «Синергия» - Университетская серия, 2013.

6.2. Информационно-аналитические системы. Основы проектирования и применения. Учебное пособие, руководство, практикум. В.С. Белов. - М.: Московский государственный университет экономики, статистики и информатики, 2005.

ЛАБОРАТОНАЯ РАБОТА № 3

Создание аналитического отчета OLAP с помощью JUPYTER

1. **Цель работы** – получение студентами навыков оперативного анализа данных и работы в программной оболочки Jupyter

2. **Краткие теоретические сведения**

Для выполнения лабораторной работы потребуется программная оболочка Jupyter. Её можно установить к себе на компьютере локально как независимую программу или использовать в сети Интернет на различных сайтах, которые предоставляют аналогичный функционал.

Для установки на компьютер:

```
sudo pip3 install jupyter pandas matplotlib
jupyter notebook
```

В сети Интернет можно использовать следующий ресурс - https://notebooks.gesis.org/

Данный ресурс требует регистрации и предоставляет Вам доступ к собственному серверу, на котором вы можете создавать «блокноты», в которых будет производиться данная лабораторная работа.

После регистрации на сайте вам предложено запустить свой сервер для работы Jupyter. Меню запуска будет выглядеть так.

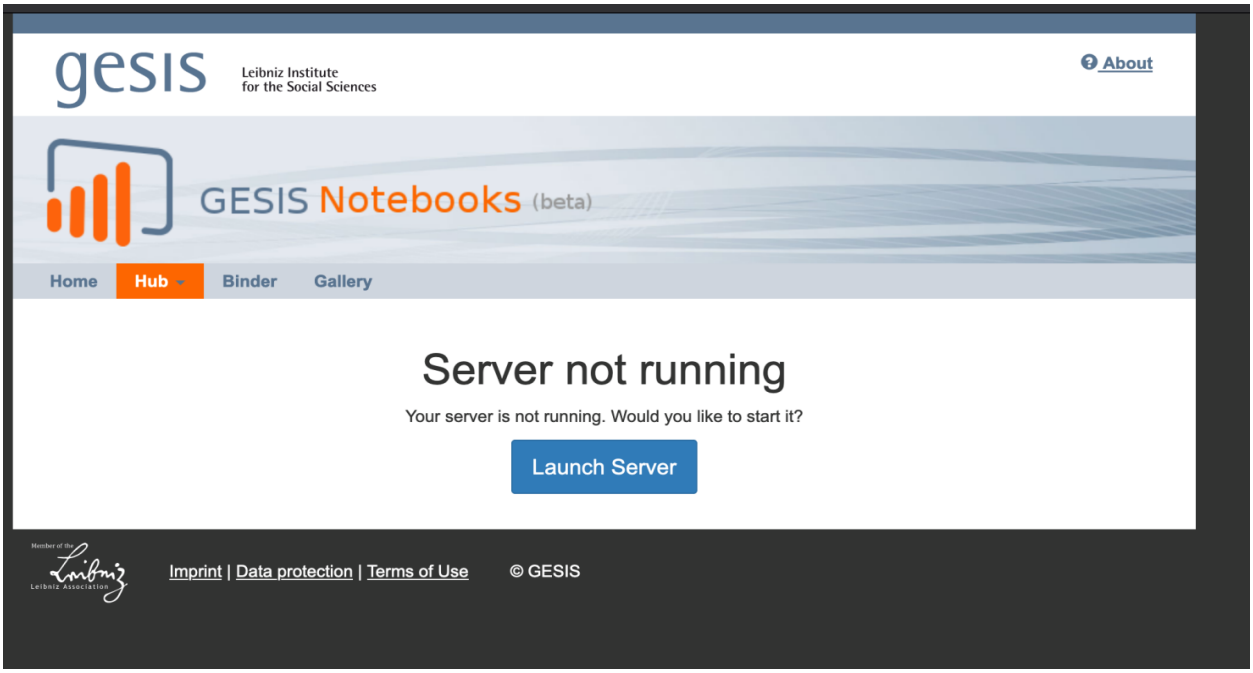

Рис.1

После нажатия на кнопку «Launch Server» начнется запуск сервера.

По окончанию запуска вы попадёте в свою домашнюю директорию, а также перед вами появится меню создания или загрузки блокнотов (справа).

Рис.2

Нажав в меню на кнопку «New», вы получите выпадающий список, в котором необходимо выбрать язык Python. После этого произойдёт перенаправление на только что созданный блокнот.

Алгоритм работы с блокнотами достаточно прост. Вы вводите строку или несколько строк на языке программирования, после чего нажимаете кнопку «Run» или сочетание клавиш Shift+Enter или Ctrl+Enter.

3. Краткое описание используемого оборудования

Процессор Intel(R) Atom(TM) CPU D525 @ 1.80 GHz, ОЗУ 2.00 Гб

4. Порядок выполнения работы

Для выполнения лабораторной работы нам понадобятся всего 2 модуля - *pandas* и *matplotlib.pyplot*. Первый модуль представляет собой библиотеку для анализа данных, а второй позволяет строить графики.

Проведём первоначальную подготовку рабочего окружение: подключим

необходимые модули и зададим размер графиков, которые построим позже.

```
import pandas as pd
import matplotlib.pyplot as plt

plt.style.use('ggplot')
plt.rcParams['figure.figsize'] = (15, 5)
```

Рис. 3.

Теперь у нас есть всё необходимое, чтобы начать обработку данных. За основу входного файла с данными будет взята таблица следующего формата (рисунок ниже).

Продавец	Марка самолета	Оборот	Месяц	продажи за 1 месяц
Ингеборга	Dassault Falcon 7X	165000000	январь	5
Ингеборга	Dassault Falcon 7X	132000000	февраль	4
Ингеборга	Dassault Falcon 7X	66000000	март	2
Ингеборга	Gulfstream G500	299500000	апрель	5

Рис. 4

Тогда после считывания мы можем просто обработать все строки данной таблицы с помощью нижеприведённого цикла.

```
for i in range( data.index.stop ):
    seller_name = data.iloc[ i ][0]
    item_name = data.iloc[ i ][ 1 ]
    summary_cost = data.iloc[ i ][ 2 ]
    month = data.iloc[ i ][ 3 ]
    count_of_sale = data.iloc[ i ][ 4 ]
```

Рис. 5

Ниже приведены соответствия переменной и полю в таблице:

- seller_name – продавец;
- item_name – марка;
- summary_cost – оборот;
- month – месяц;
- count_of_sale - продажи за 1 месяц.

Не сложно заметить, что для каждого из полей смещение у объекта data.iloc[i][n] увеличивается на 1, так что если будет изменена структура таблицы, то придётся немного корректировать данный код считывания.

Таким образом можно получить всё содержимое таблицы по строкам и составить объекты для каждой строки. После чего обрабатывать эти данные с необходимой целью.

Ниже приведён код, который считает для каждого объекта и продавца:

- общее число продаж;
- оборот за месяц;
- общий оборот.

```
count_of_sales_data = {'Общий итог':{}}
money_turnover_data = {'Общий итог':{}}
month_turnover_data = {'Общий итог':{}}

for i in range( data.index.stop ):
    seller_name = data.iloc[ i ][0]
    item_name = data.iloc[ i ][ 1 ]
    summary_cost = data.iloc[ i ][ 2 ]
    month = data.iloc[ i ][ 3 ]
    count_of_sale = data.iloc[ i ][ 4 ]

    if month not in month_turnover_data.keys():
        month_turnover_data[ month ] = {}
        month_turnover_data[ month ][ 'Общий итог' ] =
summary_cost
    else:
        month_turnover_data[ month ][ 'Общий итог' ] +=
summary_cost

    if seller_name not in month_turnover_data[ month ].keys():
        month_turnover_data[ month ][ seller_name ] = summary_cost
    else:
        month_turnover_data[ month ][ seller_name ] +=
summary_cost

    if seller_name not in month_turnover_data[ 'Общий итог'
].keys():
        month_turnover_data[ 'Общий итог' ][ seller_name ] =
summary_cost
    else:
        month_turnover_data[ 'Общий итог' ][ seller_name ] +=
summary_cost


    if item_name not in count_of_sales_data.keys():
        count_of_sales_data[ item_name ] = {}
        count_of_sales_data[ item_name ]['Общий итог'] =
count_of_sale

        money_turnover_data[ item_name ] = {}
        money_turnover_data[ item_name ]['Общий итог'] =
summary_cost
    else:
        count_of_sales_data[ item_name ]['Общий итог'] +=
count_of_sale

        money_turnover_data[ item_name ]['Общий итог'] +=
summary_cost

    if seller_name not in count_of_sales_data[ item_name ].keys():
```

```
        count_of_sales_data[ item_name ][ seller_name ] =
count_of_sale

        money_turnover_data[ item_name ][ seller_name ] =
summary_cost
    else:
        count_of_sales_data[ item_name ][ seller_name ] +=
count_of_sale

        money_turnover_data[ item_name ][ seller_name ] +=
summary_cost

    if seller_name not in count_of_sales_data[ 'Общий итог'
].keys():
        count_of_sales_data[ 'Общий итог' ][ seller_name ] =
count_of_sale

        money_turnover_data[ 'Общий итог' ][ seller_name ] =
summary_cost
    else:
        count_of_sales_data[ 'Общий итог' ][ seller_name ] +=
count_of_sale

        money_turnover_data[ 'Общий итог' ][ seller_name ] +=
summary_cost

money_turnover_data['Общий итог']['Общий итог'] = 0
month_turnover_data['Общий итог']['Общий итог'] = 0
```

В итоге получаются ассоциативные массив, которые удобно можно превратить в DataFrame, с которыми работает pandas и строит графики.

Пример конвертирования полученного ассоциативного массива в DataFrame и получение сводной таблицы.

```
In [11]: df = pd.DataFrame.from_dict(count_of_sales_data)
         columns_list = df.columns.tolist()
         new_col_list = columns_list[1:] + [columns_list[ 0 ]]
         df = df[ new_col_list ]
         df = df.T
         df
```

Рис. 6

Пример построения графика. В DataFrame для доступа по числовому значению используется .iloc. Чтобы вывести на график значения без последних строк и столбцов, которые являются общим итогом, в квадратных скобках указываются значения и срез. Для построения графика используется plot(), для

задания типа графика kind, чтобы подписи столбцов выводились горизонтально нужно добавить rot.

```
In [12]: df.iloc[[0,1,2,3,4],:4].plot(kind='bar', rot=0)
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1ee707df60>
```

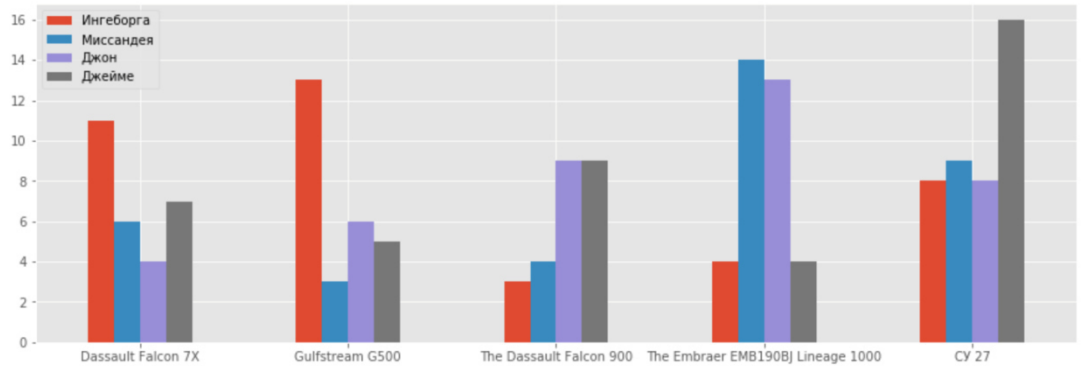

Рис. 7

Аналогично можно сделать и с другими полученными после обработки данными.

5. Содержание отчета

5.1. Результаты вычислений;

5.2. Анализ полученных результатов и выводы.

6. Литература

6.1. Информационно-аналитические системы. Учебник для вузов. Т.В. Алексеева, Ю.В. Америди, В.В. Дюк - М.: Московский финансово-промышленный университет «Синергия» - Университетская серия, 2013.

6.2 https://notebooks.gesis.org/

ЛАБОРАТОНАЯ РАБОТА № 4

Интеллектуальный анализ данных социальной сети VK

1. Цель работы – Усвоить основы интеллектуального анализа данных и применение полученных знаний на практике, на примере создания неявного социального графа социальной сети ВКонтакте.

2. Краткие теоретические сведения

Социальный граф — это граф, узлы которого представлены социальными объектами, такими как пользовательские профили с различными атрибутами (например: имя, день рождения, родной город), сообщества, медиаконтента и так далее, а рёбра — социальными связями между ними.

Неявный социальный граф — это такой граф, который можно сформировать (вывести, вычислить) на основе взаимодействий пользователя со своими «друзьями» и группами «друзей» в социальной сети. В этом графе в отличие от обычного социального графа нет явного указания «друзей», то есть нет явных социальных связей.

API (application programming interface) — это посредник между разработчиком приложений и какой-либо средой, с которой это приложение должно взаимодействовать. API упрощает создание кода, поскольку предоставляет набор готовых классов, функций или структур для работы с имеющимися данными.

API ВКонтакте — это интерфейс, который позволяет получать информацию из базы данных vk.com с помощью http-запросов к специальному серверу. Вам не нужно знать в подробностях, как устроена база, из каких таблиц и полей каких типов она состоит — достаточно того, что API-запрос об этом «знает». Синтаксис запросов и тип возвращаемых ими данных строго определены на стороне самого сервиса.

JSON — это формат обмена данными, с которым вы постоянно будете сталкиваться. Формат JSON предоставляет способ хранения словарей, списков, примитивов, таких как числа и строки, и их комбинаций. Иначе говоря, с помощью JSON можно смоделировать практически любую структуру данных.

3. Краткое описание используемого оборудования

Процессор Intel(R) Atom(TM) CPU D525 @ 1.80 GHz, ОЗУ 2.00 Гб

4. Порядок выполнения работы

Для выполнения лабораторной работы и удобной эксплуатации API ВКонтакте будет использован Python модуль vk_api, который предназначен для написания скриптов для социальной сети ВКонтакте (vk.com) (API wrapper). Так же понадобятся и другие модули такие как:

- networkx - предназначена для работы с графами и другими сетевыми структурами.

- matplotlib - библиотека для визуализации данных двумерной или трёхмерно графикой.

- numpy - Библиотека предоставляет реализации вычислительных алгоритмов (в виде функций и операторов), оптимизированные для работы с многомерными массивами.

- scipy - расширяет функционал numpy.

Для установки модулей на компьютер выполните команду:

sudo pip3 install vk_api networkx matplotlib numpy scipy

Проведём первоначальную подготовку рабочего окружения подключим необходимые модули и зададим константы.

```
# -*- coding: utf-8 -*-
import vk_api
import networkx as nx
import matplotlib.pyplot as plt
import json

# Выводить ли начальный иследуемый аккаунт на граффе

add_my = True
# id иследуемого аккаунта. None для ид аккаунта под которым авторизировались
start_id = None
# Логин и пароль от ВК
login, password = '+70123456789', 'PASS'

def main():
    pass


if __name__ == "__main__":
    main()
```

Также нам понадобится пройти авторизации. в API. Для простоты был выбран способ авторизации через логин и пароль от учетной записи. С другими способами аутентификации можете ознакомиться в документации к модулю vk_api [1].

```
vk_session = vk_api.VkApi(login, password)
```

```
try:
    vk_session.auth(token_only=True)
except vk_api.AuthError as error_msg:
    print(error_msg)
    return
```

После авторизации для совершения обращений к методам API, как к обычным классам, выполним:

```
vk = vk_session.get_api() # object<VkApiMethod>
```

Создав авторизованное подключение к API и получив VkApiMethod, можно отправить первый запрос, запросив информацию о профиле:

```
""" VkApi.method позволяет выполнять запросы к API. В этом примере
  используется метод users.get (https://vk.com/dev/users.get) который возвращает
  расширенную информацию о пользователе.
"""
print(vk.users.get())
```

Вы должны увидеть неудобочитаемый ответ API — список словарей Python— в отличие от любого сообщения об ошибке. Для примера ниже приводятся усеченные результаты.

```
[{'id': 12345678, 'first_name': 'Ivan', 'last_name': 'Ivanov', 'is_closed': False, 'can_access_closed': True}]
```

Для улучшения читабельности будем использовать встроенного пакета json:

```
print(json.dumps(vk.users.get(), indent=1))
```

```
[
 {
  "id": 12345678,
  "first_name": "Ivan",
  "last_name": "Ivanov",
  "is_closed": false,
  "can_access_closed": true
 }
]
```

4.1. Сбор и подготовка данных

Для построения неявного социального графа нам необходимо собрать и подготовить к построению всю необходимую информацию. Таковой является списки друзей и их друзей.

Что бы получить список друзей пользователя используйте метод *friends.get,* который, без указания параметров, вернёт словарь с количеством найденных у пользователя друзей и список их id.

```
    my_id = start_id if start_id else vk.users.get()[0]['id']
    friend = vk.friends.get(user_id=my_id)['items']
    if add_my:
        friend.append(my_id) #добавим исследуемый id для дальнейшего построения

    print(json.dumps(vk.friends.get(), indent=1))
```

```
    {
     "count": 123,
     "items": [
      112233,
      ...
      445566,
     ]
    }
```

Поскольку API имеет ограничение на количество запросов в секунду, для получения списка друзей у наших друзей будем использовать встроенный в библиотеку метод vk_request_one_param_pool. Который позволяет сделать несколько обращений к API за один запрос за счет метода execute.

```
""" vk_request_one_param_pool Позволяет сделать несколько обращений к API за один запрос за счет метода execute. Можно использовать, когда запрос идет к одному методу и когда изменяется только один параметр. В данном случае  это 'user_id'.
    Кроме того не нужно получать .result для каждого ключа.
"""
friends, errors = vk_api.vk_request_one_param_pool(
    vk_session,
    'friends.get',  # Метод
    key='user_id',  # Изменяющийся параметр
    values=friend,
```

```
)
```

Получив списки можно выполнить их очистку от посторонних людей:

```
friends_clean = {}
   s = set(friend)
   for k, v in friends.items():
    t = list(s & set(v['items']))
friends_clean[k] = t
```

Может возникнуть потребность убрать пользователей с числом связей менее единицы, которые не имеют с Вами общих друзей или наоборот ограничить количество связей верхним приделом, когда нужно выявить малые группы общения. Для этого перед строкой *friends_clean[k] = t* нужно добавить условие if len(t) > 1 and len(t)<20.

***Задание**: реализуйте возможность исключения некоторых друзей из анализа данных, т. к. порой они могут внести только путаницу (к примеру в друзьях может быть аккаунт доставки еды который будет всех связывать и мешать дальнейшему анализу).*

4.2 Интерпретация данных

Полученные и обработанные данные удобнее интерпретировать в графическом виде, построив ориентированный граф. Постарение графа будет производится через библиотеку networkx.

```
g = nx.Graph(directed=False)
```

Добавление вершин и ребер выполняется через методы g.add_node и g.add_edge соответственно.

Задание: заполните данными графф, где каждый ключ словаря это вершина, а значения это ребра от этой вершины к другим.

После внесения данных о вершинах и ребрах задаётся конфигурации построения:

```
pos = nx.spring_layout(g)
nx.draw(g, pos, node_size=30, with_labels=True, width=0.2)
```

Отображение построенного графа происходит с помощью модуля matplotlib, для этого выполните команду

```
plt.show()
```

В результате вы увидите на рис. 2 неявный социальный граф и на котором будут видно Ваши круги общения.

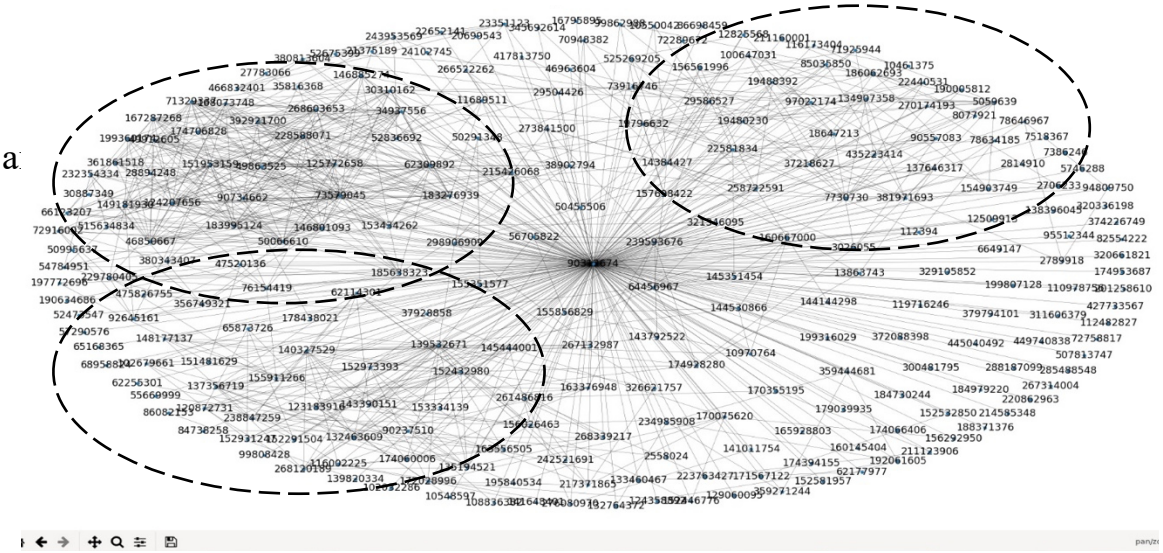

Рис. 1

На рисунке 1 пунктирной линией выделены связные группы людей, к примеру друзья по школе, ВУЗу, работе, семейный круг и д.р.. Для наглядности можно исключить вершину с исследуемым id, для этого присвойте False переменной add_my.

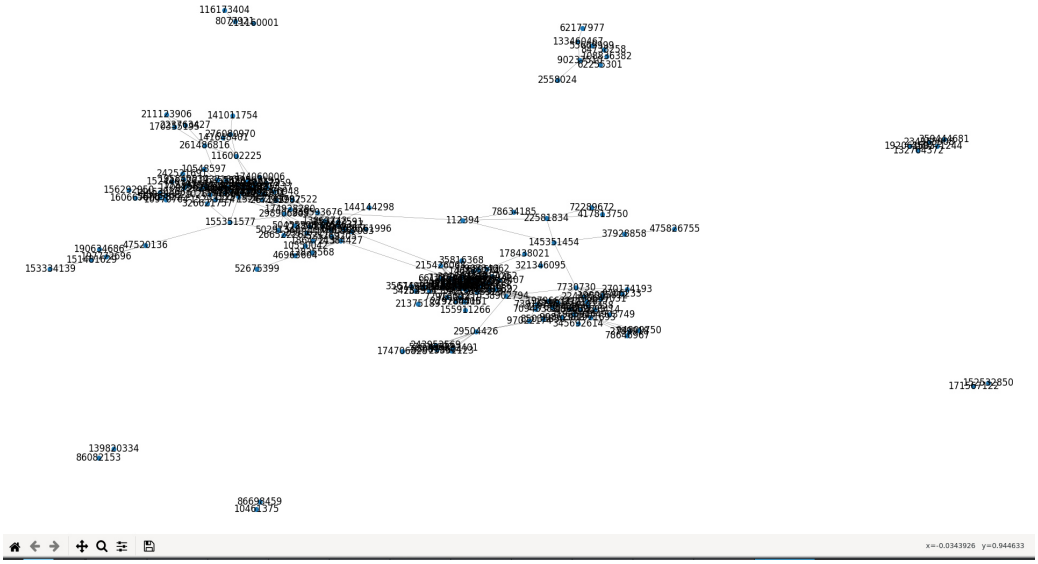

Рис. 2

Как видно из рисунка 2 при исключении вершины с исследуемым id становиться видно больше групп общения. Все отображённые контакты априорно связаны с исследуемым id.

Проанализируйте полученные данные и сделайте выводы.

Листинг программы приведён в *приложении 1*

5. Содержание отчета

5.1. Листинг программы;

5.2. Результаты вычислений;

5.3. Анализ полученных результатов и выводы.

6. Литература

6.1. Информационно-аналитические системы. Учебник для вузов. Т.В. Алексеева, Ю.В. Америди, В.В. Дюк - М.: Московский финансово-промышленный университет «Синергия» - Университетская серия, 2013.

6.2. https://vk-api.readthedocs.io/en/latest/

6.3. https://vk.com/dev/first_guide

6.4. https://networkx.github.io/documentation/stable/index.html

Приложение 1

```
1.  # -*- coding: utf-8 -*-
2.  import vk_api
3.  import networkx as nx
4.  import matplotlib.pyplot as plt
5.  import json
6.
7.  # Выводить ли начальный иследуемый аккаунт на граффе
8.  add_my = False
9.  # id иследуемого аккаунта. None для ид аккаунта под
которым авторизировались
10. start_id = None
11. # Логин и пароль от ВК
12. login, password = 'LOGIN', 'PASS'
13.
14.
15. def build_graf(friend: dict):
16.  g = nx.Graph(directed=False)

17.  Задание 2
18.
19.  pos = nx.spring_layout(g)
20.  nx.draw(g,  pos,  node_size=30,  with_labels=True,
width=0.2)
21.  plt.show()
22.
23.
24. def main():
25.
26.  vk_session = vk_api.VkApi(login, password)
27.
28.  try:
29.  vk_session.auth(token_only=True)
30.  except vk_api.AuthError as error_msg:
31.  print(error_msg)
32.  return
33.
34.  vk = vk_session.get_api()
35.
```

```
""" VkApi.method позволяет выполнять запросы к API. В этом примере
    используется метод users.get (https://vk.com/dev/users.get) который возвращает
    расширенную информацию о пользователе.
    """
    my_id = start_id if start_id else vk.users.get()[0]['id']


    """ Метод friends.get (https://vk.com/dev/friends.get) с параметром
    user_id, возвращает список друзей пользователя с указаным user_id
    """
    friend = vk.friends.get(user_id=my_id)['items']

     print(json.dumps(vk.friends.get(), indent=1))

     if add_my:
     friend.append(my_id)

    """ vk_request_one_param_pool Позволяет сделать несколько обращений к API за один запрос
    за счет метода execute. Можно использовать, когда запрос идет к
    одному методу и когда изменяется только один параметр. В данном случае
    это 'user_id'.
    Кроме того не нужно получать .result для каждого ключа.
    """
    friends, errors = vk_api.vk_request_one_param_pool(
     vk_session,
     'friends.get', # Метод
     key='user_id', # Изменяющийся параметр
     values=friend,
     )
```

Задание1

```
    friends_clean = {}
     s = set(friend)
     for k, v in friends.items():
     t = list(s & set(v['items']))
     if len(t) >= 1:
     friends_clean[k] = t

```

```
75.  print(friends_clean)
76.
77.  build_graf(friends_clean)
78.
79.
80. if __name__ == "__main__":
81.     main()
```