

Государственное образовательное учреждение высшего профессионального
образования



*«Московский государственный технический университет
имени Н.Э. Баумана»*

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ

Информатика и системы управления

КАФЕДРА

Системы обработки информации и управления

Лабораторная работа №7 по курсу
"Оперативный анализ данных / Средства визуализации данных"

"Знакомство с платформой для анализа данных RapidMiner"

Оглавление

1. Цель работы	3
2. Теоретическая часть.....	3
2.1 Описание платформы	3
2.2 Возможности платформы	3
3. Подготовка к выполнению лабораторной работы	8
3.1 Установка ПО.....	8
3.2 Копирование файлов-примеров	8
4. Ход выполнения лабораторной работы	8
4.1 Импорт данных	8
4.2 Гендерная статистика	10
4.3 Задание: Лучший сезон Симпсонов в этом веке	13
4.4 Матрица корреляций	14
4.5 Задание: корреляция параметров домов.....	19
4.6 Дерево решений	20
4.5 Машинное обучение в RapidMiner	20
5. Контрольные вопросы	26
6. Использованная литература	26

1. Цель работы

Целью работы является знакомство студентов с аналитической платформой RapidMiner и приобретение навыков бизнес-анализа и визуализации данных с использованием данной среды.

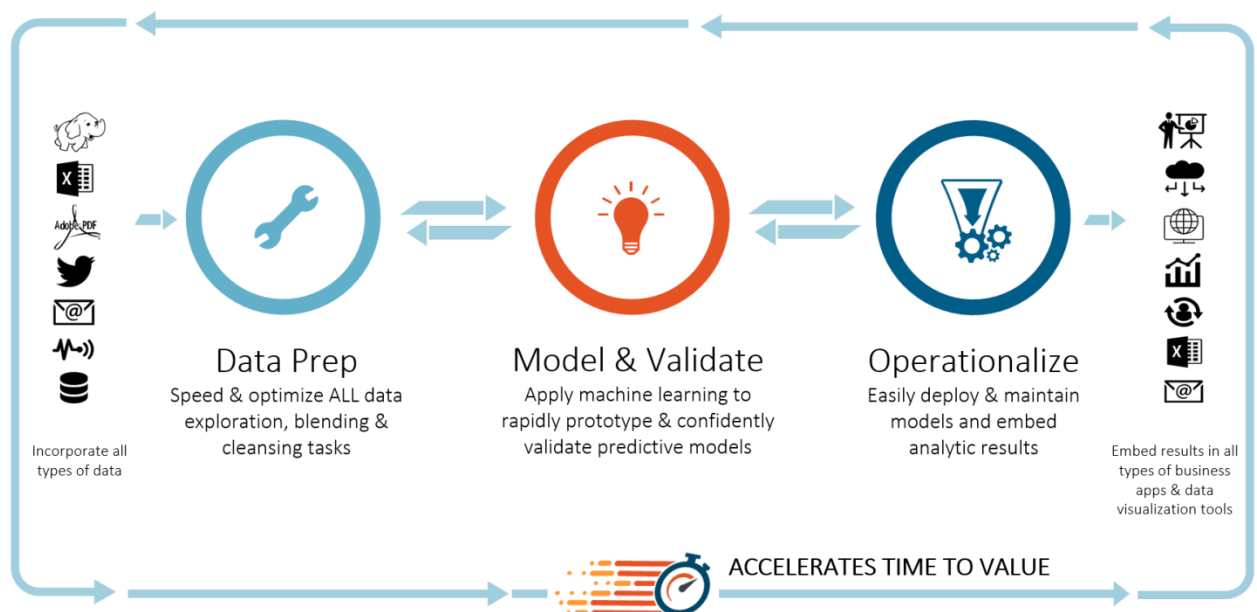
2. Теоретическая часть

2.1 Описание платформы

RapidMiner — это мощная и многопользовательская платформа, она служит для создания, передачи и обслуживания наукоемких данных. Эта платформа предлагает больше функций, чем любое другое визуальное решение, к тому же она открыта и расширяема для поддержки всех потребностей научных данных.

RapidMiner ускоряет создание полных аналитических рабочих процессов — от подготовки данных до моделирования до развертывания бизнеса — в единой среде, значительно повышая эффективность и сокращая время, необходимое для проектов в области данных.

Если сравнивать RapidMiner с другими программами, то у RM гораздо шире функциональные возможности по обработке, банально больше узлов.



2.2 Возможности платформы

RapidMiner — инструмент, созданный для data-майнинга, с основной идеей, что аналитик не должен программировать при выполнении своей работы. Программу снабдили достаточно хорошим набором операторов решающих большой спектр задач получения и обработки информации из

разнообразных источников (базы данных, файлы и т.п.), и можно с уверенностью говорить, что это ещё и полноценный инструмент для ETL (Extract, Transform, Load).

В стандартной лицензии AGPL доступно 10,000 колонок и ограничение в один логический процесс.

- **Хороший GUI.** По сути, каждый функциональный блок собран в кубик. Ничего нового в подходе, но очень крутое исполнение. Обычно разница между классическим программированием и визуальным сильно бьёт по функциональности. Например, в SPSS Modeler всего 50 узлов, а тут целых 250 в базовой загрузке.

- **Есть хорошие инструменты подготовки данных.** Обычно предполагается, что данные готовятся где-то ещё, но тут уже есть готовый ETL (получение и трансформация). В том же коммерческом SPSS возможностей для подготовки куда меньше.

- **Расширяемость.** Есть язык программирования R. Полностью интегрированы операторы система WEKA.

- **Архитектурно данные снаружи.** Ставим платформу, грузим данные и начинаем смотреть, где какие корреляции, что можем спрогнозировать.

- **Кроме IDE есть ещё сервер.** Rapid Miner Studio создаёт процессы, а на сервере их можно публиковать. Сервер знает, какой процесс когда запускать, с какой частотой, что делать, если где-то что-то отвалилось, кто отвечает за каждый из процессов, кому как отдавать ресурсы, куда выгружать результаты.

- **А ещё сервер же умеет сразу строить минимальные отчёты.** Можно выгружать не в XLS, а рисовать графику прямо там. Это нравится маркетингу маленьких проектов.

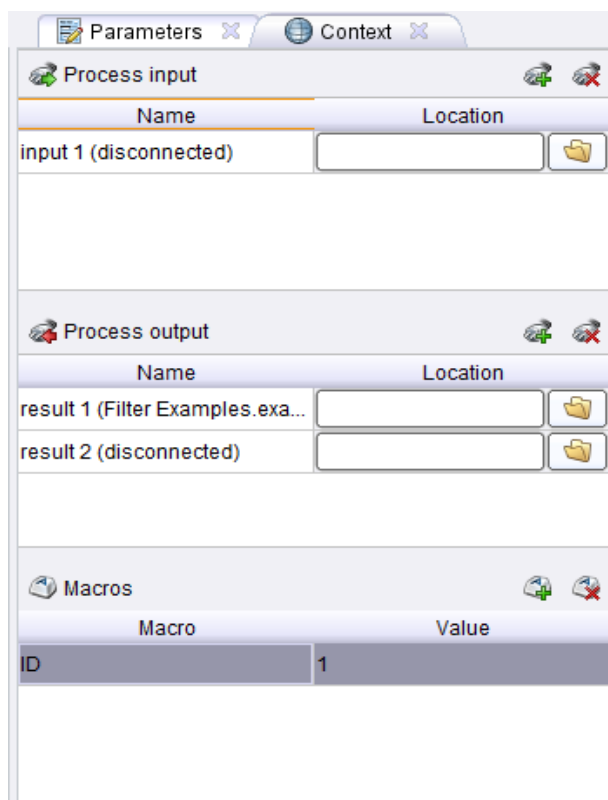
- **Быстрое развитие.** Только поднялся серьезный шум вокруг Apache Spark — через месяц интегрировали.

Процесс в RapidMiner представляет собой набор операторов, соединенных последовательно между собой. Есть операторы, которые считывают данные из файла, есть операторы, которые производят фильтр по определенным признакам, есть операторы, которые записывают результат в файл, и многие другие.

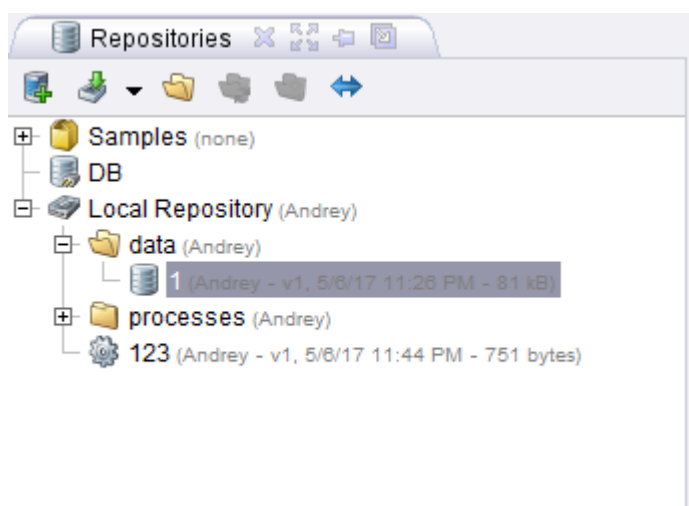
Оператор — это логическая единица, которая может производить какое-то действие над данными. Оператор имеет вход и выход. На входе поступают сырые данные, на выходе получаются обработанные данные. Все операторы доступны в левой колонке и отсортированы по функциональному признаку.

В **RapidMiner** есть макросы — это параметры работы процесса, которые можно использовать в любой его точке (т.е. они являются глобальными переменными). Например, в качестве макроса можно использовать имя файла, дату его создания, среднее значение какого-либо атрибута данных,

наилучшую достигнутую точность, номер итерации, последнее время запуска процесса.



Место для хранения процессов RM может быть локальным, а также удаленным (RapidMiner Server), для которого возможно исполнять процессы на стороне сервера, многопользовательский доступ к процессам/соединениям БД, запуск процессов по расписанию или отдача данных как веб-сервис.



Кроме Макроса во вкладке «контекст» присутствуют параметры process input и process output.

Process input – данные, подающиеся на вход. Может быть указан путь, откуда вытаскивать данные.

Process output — данные, которые передаются к следующему процессу. Может быть указан путь для сохранения данных.

The screenshot shows the 'Parameters' tab of a software interface. It contains three main sections: 'Process input', 'Process output', and 'Macros'.

Process input

Name	Location
input 1 (disconnected)	<input type="text"/>

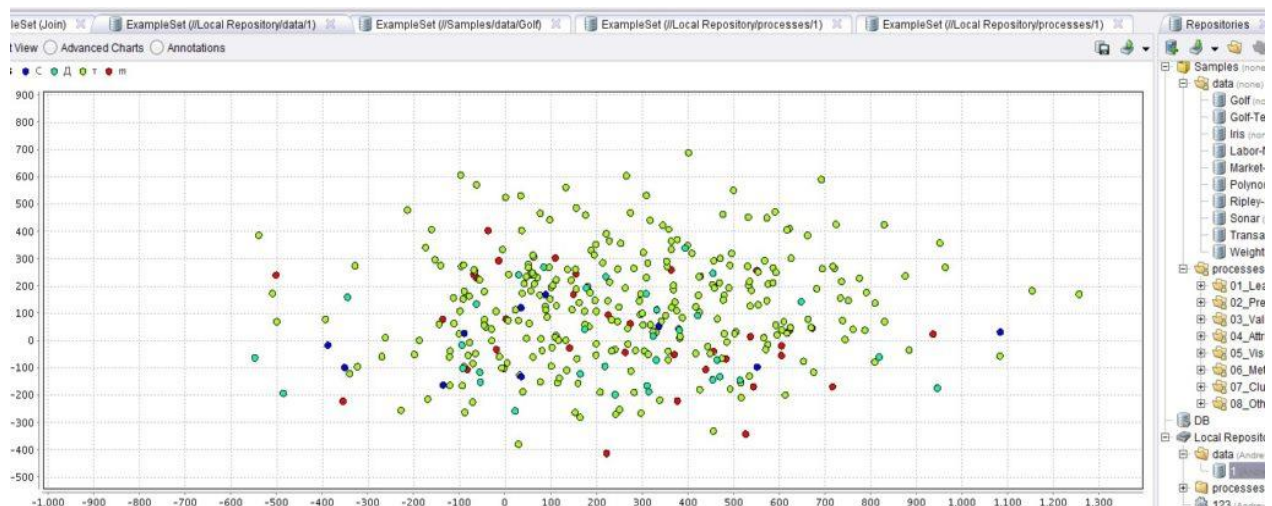
Process output

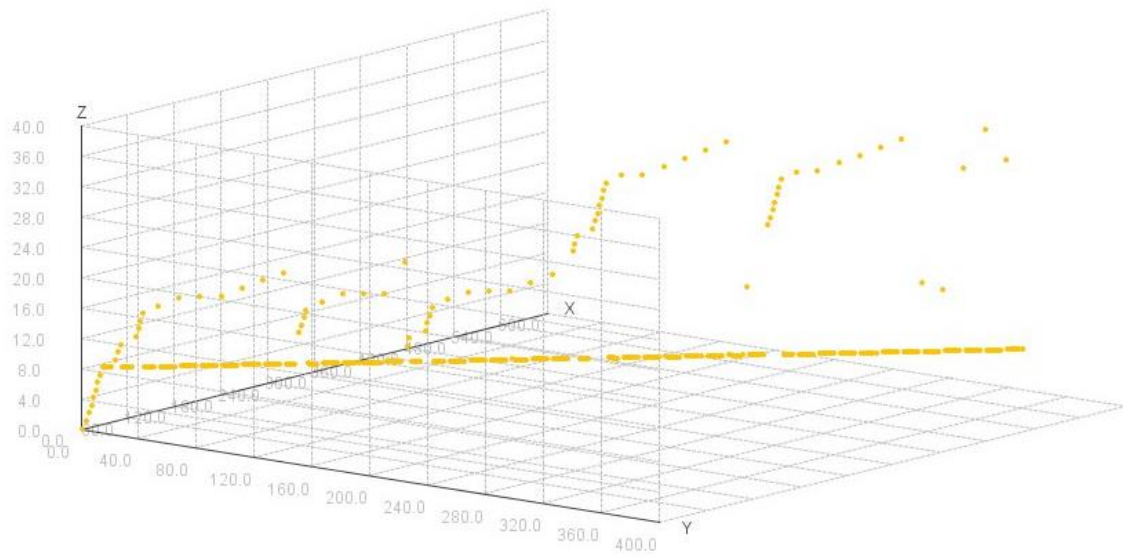
Name	Location
result 1 (Filter Examples.exe...)	<input type="text"/>
result 2 (disconnected)	<input type="text"/>

Macros

Macro	Value
ID	1

После создания процесса и его запуска можно построить графики разброса величин и многое другое.





3. Подготовка к выполнению лабораторной работы

3.1 Установка ПО

3.1.1 Скачиваем дистрибутивы с официального сайта:
<https://docs.rapidminer.com/latest/studio/installation/>

3.1.2 Регистрируемся на сайте с использованием почты в домене @student.bmstu.ru

3.1.3 Устанавливаем на компьютер RapidMiner Studio

3.1.4 Входим под учетной записью, зарегистрированной на сайте

3.2 Копирование файлов-примеров

3.2.1 Все необходимые файлы могут либо храниться в самой программе RapidMiner Studio

3.2.2 Либо необходимо обратиться за помощью к лаборантам для получения необходимых файлов

4. Ход выполнения лабораторной работы

4.1 Импорт данных

Все действия в RM выполняются в рамках процесса (Process). При открытии программы появляется вступительное окно, в котором можно создать новый процесс или открыть уже начатый. Чтобы начать работу с RM, выберем “NEW PROCESS”, затем “Blank” (пустой процесс)

Обработка данных в RM происходит при помощи операторов (Operators), которые применяются последовательно к источнику данных.

Интерфейс RM выполнен в виде рабочей области со вкладками, размер и положение которых можно менять. Основными вкладками являются:

- Repository – содержит данные для анализа
- Operators – инструменты для анализа данных
- Process – вкладка с главным процессом
- Parameters – настройка оператора

Если вы закрыли одну из вкладок, то открыть её снова можно в списке **View -> Show Panel**

4.1.1 Чтобы добавить данные в RM, необходимо выбрать Import Data во вкладке Repository.

4.1.2 Выбираем данные с локального источника данных (My Computer). Указываем путь к файлу, выбираем файл. В нашем случае это SocialNetworkPoll.csv. Нажимаем Next.

4.1.3 Появится окно с параметрами распознавания. Необходимо выставить все параметры как на скриншоте. Каждый атрибут должен быть помещен в отдельный столбец.

Specify your data format

☒ Header Row 1
 Start Row 1
 Column Separator Comma ","

File Encoding windows-1251
 Escape Character \

☒ Use Quotes "
☐ Trim Lines
☒ Skip Comments #


1	Question	Segment Type	Segment Description	Answer	Count	Percentage
2	You open ur phone and ...	Mobile	Mobile respondents	Instagram	2559	0.273
3	You open ur phone and ...	Mobile	Mobile respondents	Facebook	1182	0.126
4	You open ur phone and ...	Mobile	Mobile respondents	Snapchat	5423	0.579
5	You open ur phone and ...	Mobile	Mobile respondents	Linkedin	210	0.022
6	You open ur phone and ...	Web	Web-based respondents	Instagram	30	0.256


4.1.4 Затем идет форматирование колонок. Для изменения имени, типа или удаления колонки необходимо нажать на шестеренку в заголовке колонки. В нашем случае изменения не требуются.


4.1.5 Теперь выберем куда необходимо сохранить данные. Создаем свою папку правым кликом мыши по **Local Repository** или соглашаемся с директорией по умолчанию. Желательно создать отдельную папку (subfolder).


4.1.6 Если все предыдущие шаги были выполнены правильно, то вам должна открыться таблица с результатами импорта. Также файл должен появиться во вкладке Repositories.

Мы импортировали результаты опроса студентов США о том, на какое уведомление они нажмут первым, увидев на телефоне уведомления от Instagram, Facebook, LinkedIn и Snapchat.



 Data



 Statistics


 Visualizations


 Annotations

Open in

 Turbo Prep

 Auto Model

Row No.	Question	Segment Ty...	Segment De...	Answer	Count	Percentage
1	You open ur ...	Mobile	Mobile respo...	Instagram	2559	0.273
2	You open ur ...	Mobile	Mobile respo...	Facebook	1182	0.126
3	You open ur ...	Mobile	Mobile respo...	Snapchat	5423	0.579
4	You open ur ...	Mobile	Mobile respo...	Linkedin	210	0.022
5	You open ur ...	Web	Web-based r...	Instagram	30	0.256
6	You open ur ...	Web	Web-based r...	Facebook	32	0.274
7	You open ur ...	Web	Web-based r...	Snapchat	47	0.402
8	You open ur ...	Web	Web-based r...	Linkedin	8	0.068
9	You open ur ...	Gender	Female resp...	Instagram	1576	0.300
10	You open ur ...	Gender	Female resp...	Facebook	644	0.122

Вы можете увидеть статистику по импортированным данным, перейдя во вкладку **Statistics**. В данный момент эта вкладка никакой полезной информацией не располагает.

Во вкладке **Visualizations** вы можете визуализировать данные различными способами.

Из-за особенностей импортированных данных получить приятную и удобную визуализацию чего-либо конкретного будет очень сложно.

В импортированных данных одного и того же человека спрашивали про университет, где он учится, про пол и т.д. Соответственно, данные из полей **Count** и **Percentage** учитывают внутри себя одних и тех же людей. Например, сложив **Percentage** (процент проголосовавших) всех **Female respondent** (девушек), мы уже получим 1.

4.2 Гендерная статистика

Допустим, мы хотим узнать результаты того, как ответили мужчины, и как ответили женщины. Для этого необходимо произвести несколько преобразований.

4.2.1 Перейдите на вкладку **Design** и перетащите файл на вкладку **Process**, чтобы приступить к анализу. Назовите импортированные данные *«Результаты опроса»*.

4.2.2 Отфильтруем данные, оставив только **Gender** в качестве **Segment Type**.

4.2.2.1 Для этого из вкладки **Operators** перетащим **Filter Examples**.

4.2.2.2 Свяжем конец **out** блока *«Результаты опроса»* с входом **in** фильтра.

4.2.2.3 Дважды кликнув по фильтру или нажав на **Add Filters** во вкладке **Parameters**, настроим фильтр так, как мы обговорили в начале этого абзаца.

4.2.2.4 Назовите фильтр *«Фильтр по полу»*.

4.2.3 Чтобы посмотреть результат фильтрации, соедините выход фильтра с точкой **res** на вкладке **process**.

4.2.4 Запуск процесса производится нажатием кнопки *«Start»* (синей кнопки с треугольником вверху приложения).

Вам откроется таблица с результатами фильтрации.

Row No.	Question	Segment Ty...	Segment De...	Answer	Count	Percentage
1	You open ur ...	Gender	Female resp...	Instagram	1576	0.300
2	You open ur ...	Gender	Female resp...	Facebook	644	0.122
3	You open ur ...	Gender	Female resp...	Snapchat	2967	0.564
4	You open ur ...	Gender	Female resp...	Linkedin	73	0.014
5	You open ur ...	Gender	Male respond...	Instagram	1008	0.240
6	You open ur ...	Gender	Male respond...	Facebook	565	0.135
7	You open ur ...	Gender	Male respond...	Snapchat	2483	0.591
8	You open ur ...	Gender	Male respond...	Linkedin	142	0.034

Статистика все еще неинформативна. Однако с визуализацией можно поработать.

4.2.5 В качестве типа графика Plot type выберите столбчатую диаграмму Bar (Column). Пусть X-Axis Column будет Answer, Value columns – Count. Уже что-то получается.

Но непонятно, где результаты мужчин, а где женщин. Да и хотелось бы, чтобы результаты были сгруппированы по соцсетям для наглядного сравнения результатов по каждой соцсети.

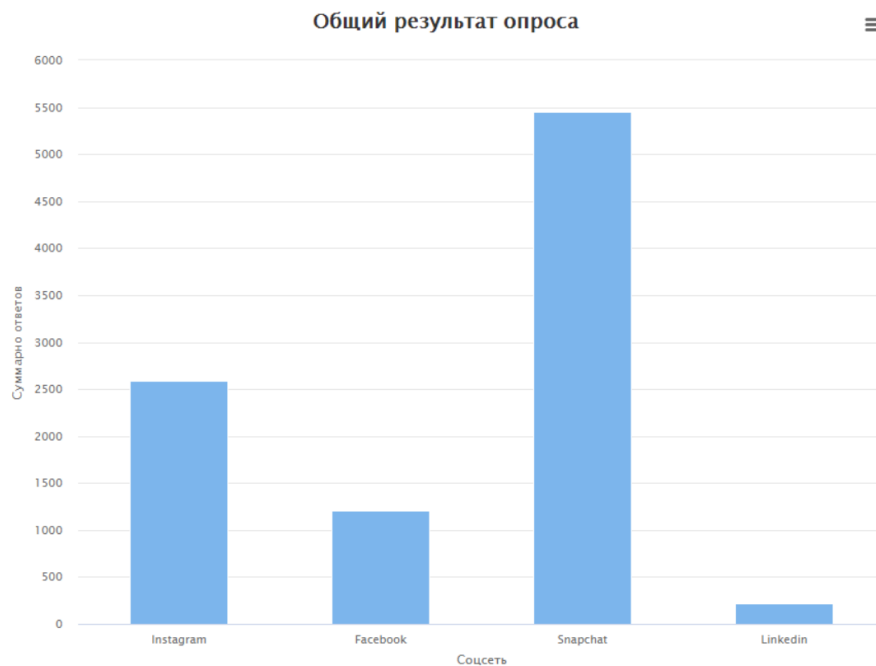
4.2.6 Для этого поставим галочку **Aggregate data**. Группируем по ответу (**Answer**), в качестве **Aggregation Function** поставьте **сумму**, ведь мы суммируем кол-во ответов.

Таким образом, мы получили общую статистику, показывающую что студента заинтересовало бы в первую очередь.

4.2.7 Назовём график «*Общий результат опроса*». Для этого необходимо зайти в свойства Title. Сделаем жирный шрифт размером 20 в свойствах Title font.

4.2.8 Переименуем оси в соответствующих пунктах свойств. Пусть ось X – «Соцсеть», ось Y – «Суммарно ответов»

4.2.9 Уберем легенду, убрав соответствующую галочку.



Но ведь мы хотели узнать, как отвечали мужчины и женщины отдельно. Как это увидеть?

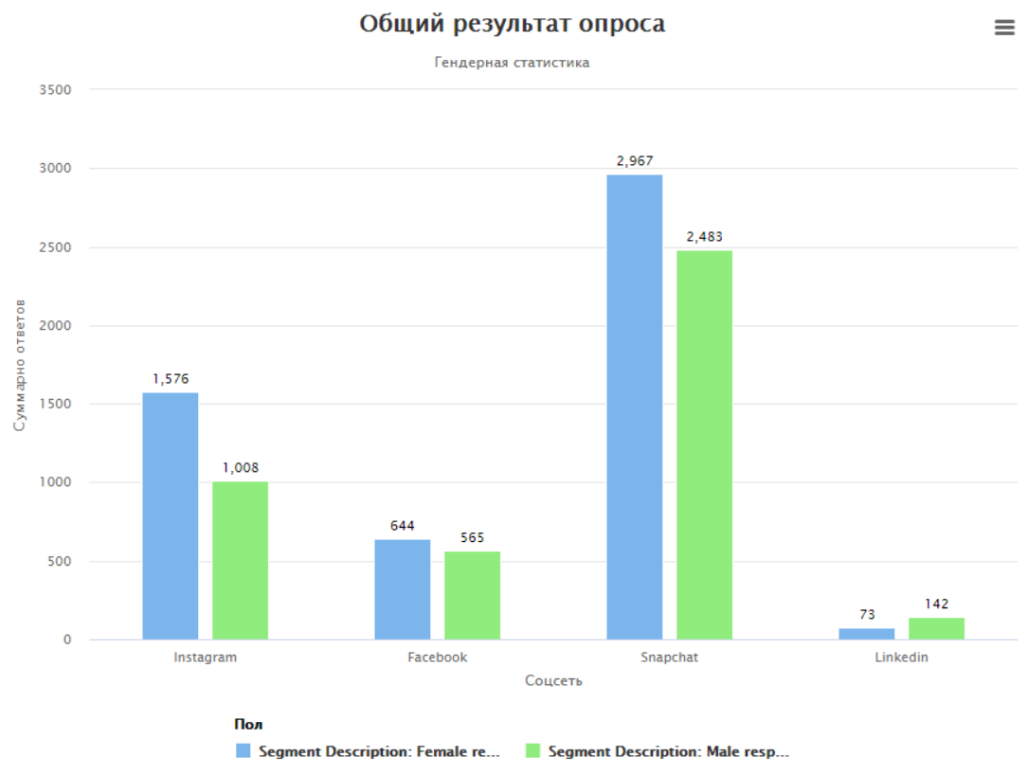
4.2.10 Для этого выберите **Segment Description** в поле Color Group.

4.2.11 Все еще непонятно, где мужчины и где женщины. Исправим это, вернув обратно галочку **Show legend**.

4.2.12 Установим заголовок легенды «Пол».

4.2.13 Хорошо бы сделать график более информативным, добавив точные значения для каждой колонки. Для этого установите галочку Show labels в **Plot Style -> Labels Style**.

4.2.14 Ну и в качестве завершающего штриха добавим подзаголовок Subtitle «Гендерная статистика».



4.3 Задание: Лучший сезон Симпсонов в этом веке

Пользуясь изученными инструментами и предоставленными исходными данными в **simpsons_episodes.csv**, найдите лучший сезон Симпсонов по версии IMDB за последние 20 лет. Для этого визуализируйте информацию таким образом, чтобы было сразу понятно, какой сезон лучший.

4.4 Матрица корреляций

Коэффициент корреляции – показатель статистической связи двух атрибутов, изменяется в пределах от -1 до 1.

Корреляцию считают:

- **Функциональной** при $|r| = 1$
- **Сильной**, если коэффициент корреляции $|r| \geq 0,7$
- **Средней** при $|r| \in [0,5; 0,7)$
- **Умеренной** при $|r| \in [0,3; 0,5)$
- **Слабой** при $|r| \in [0,2; 0,3)$
- **Очень слабой** при $|r| < 0,19$

Если $r > 0$, то корреляционная связь между атрибутами **прямая** (увеличение значения одного атрибута ведет к увеличению значения другого), при $r < 0$ – **обратная**.

Корреляционная матрица – квадратная матрица P размера $n \times n$, где n – количество атрибутов. a_{ij} – корреляционная связь между i м и j м атрибутом. Все диагональные элементы матрицы P равны 1.

4.4.1 Пусть у нас есть два независимых исследования пород собак на предмет обучаемости и послушности собак. Допустим, мы хотим ответить на вопрос: как зависит обучаемость и послушность собаки от её размеров?

4.4.2 Для начала нужно посмотреть на данные, с которыми мы будем работать, преобразовать их и объединить. Займёмся этим.

4.4.2.1 Загрузите в окно Process оба набора данных (dog_lbs и dog_reps).

4.4.2.2 Теперь соединим выходы обоих блоков Retrieve с точками res.

4.4.2.3 Запустим процесс. Нам откроются две таблицы с импортированными данными.

Становится очевидна одна проблема: в одном исследовании послушность obeu собак ввели как десятичную дробь, а в другом преобразовали в проценты, да еще и приписали сам значок процента. Послушность собаки должна быть одного формата. Самый лёгкий способ – создать новый атрибут obeu_new, который будет содержать преобразованное значение послушности. Займёмся этим.

4.4.3 С помощью оператора Replace заменим в неудобном наборе значок «%» на пустоту « ».

4.4.4 Для начала соединим выход нашего неудобного набора с входом оператора.

4.4.5 Выберем атрибут **obey** и заполним соответствующие значения **replace hat** и **replace by** в окне параметров. Подсоединив **Replace** к точке **res** и запустив процесс, увидим результат. Все проценты исчезли.

Но проблемы на этом не заканчиваются. Тип данных у атрибута **obey** все еще текстовый.

4.4.6 Чтобы исправить это, подсоединим оператор **Parse Numbers** и выберем в нем атрибут, в котором нужно преобразовать текст в число.

4.4.7 Теперь остаётся создать новый атрибут **obey_new**, в который мы положим выражение, вычисляющее новое значение для старого неудобного атрибута **obey**.

4.4.7.1 Сделать это можно с помощью оператора **Generate attributes**. В его свойствах введем название нового атрибута и выражение для его вычисления.

4.4.7.2 Очевидно, что для того, чтобы данные из двух таблиц совпадали, нам нужно работать с процентом как с десятичной дробью. Для этого в качестве выражения поставим «**obey/100**».

4.4.7.3 Если посмотреть результат, то действительно мы добавили новое поле с соответствующими значениями.

Раз нам надо, чтобы значения послушности из двух таблиц слились воедино, то столбцы должны иметь одинаковые названия.

4.4.8 Решить этот костыль можно глупым способом, добавив к вроде бы удобной нам таблице новый атрибут **obey_new** с такими же значениями, как и в обычном атрибуте **obey**.

4.4.8.6 Или! Можно поступить умнее и просто переименовать сам атрибут. Сделаем это с помощью оператора **Rename**. Переименуем **obey** в **obey_new**, очевидно.

4.4.9 Теперь мы готовы объединять таблицы. Объединять мы будем уже знакомой из SQL операцией **Join**. В данном случае, это будет оператор, у которого входы займут наши две таблицы. Какой тип **Join** выбрать? Если не знаешь, какой тип выбрать, то выбирай **Outer Join** и справляйся с последствиями. Так сделаем и мы. В качестве ключевого значения **key attribute** поставим породу **Breed** в обоих наборах.

В результате мы получили одну таблицу с данными из обеих таблиц. Раз у нас есть атрибут **obey_new**, то атрибут **obey** нам уже не нужен.

4.4.10 Выберем только нужные атрибуты оператором **Select Attributes**. В нашем случае нужны все кроме **obey**.

Некоторые значения нам неизвестны – они указаны знаком вопроса «?» в таблице. Перейдя на вкладку «Статистика» можно увидеть, сколько их в таблице в колонке Missing. Это как раз последствия outer join. Как бороться с подобным мусором? Аккуратно снести его под ковёр, пока никто не заметил.

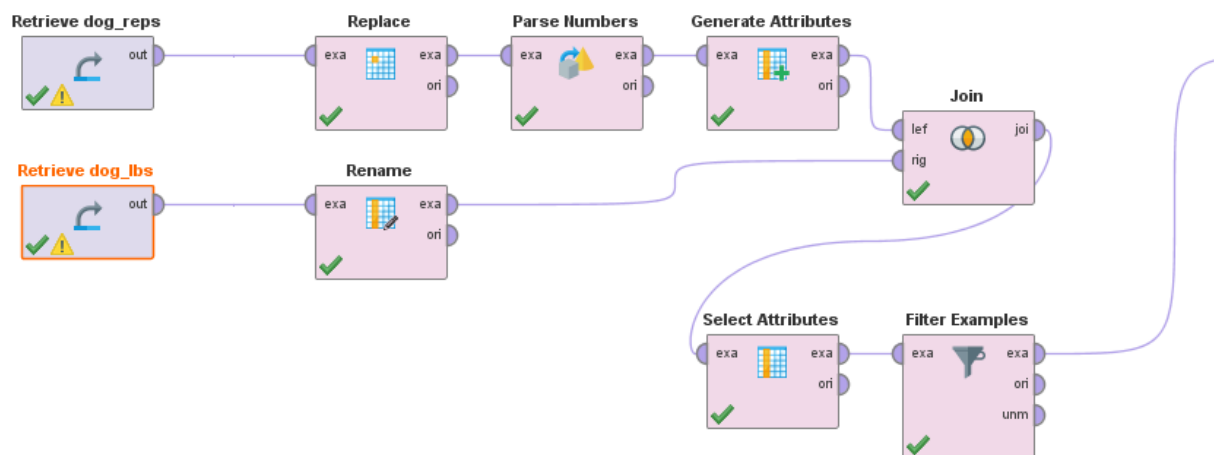
4.4.11 В нашем случае – отфильтруем набор, оставив только существующие значения.

4.4.11.1 Для этого после подключения оператора-фильтра укажем в нем те поля, которые содержали «?».

4.4.11.2 Для добавления новых полей для фильтрации используйте кнопку **Add Entry** в окне фильтра.

4.4.11.3 В качестве параметра у каждого поля поставьте «*is not missing*».

Отлично! Теперь мы подготовили данные к обработке! Можете полюбоваться своей работой, подключив выход последнего фильтра к точке **res** и выполнив процесс.



4.4.12 А теперь давайте спрячем все, чтобы никто это больше никогда не видел.

4.4.12.1 Выделите все блоки и, кликнув правой кнопкой мыши, выберите **Move into new subprocess**. Вы создали подпроцесс.

4.4.12.2 Переименуйте подпроцесс в «Подготовка данных».

Приступим к постройке матрицы корреляций (наконец-то!). Хотя нет, нужно еще немного обработать наши данные перед построением матрицы.

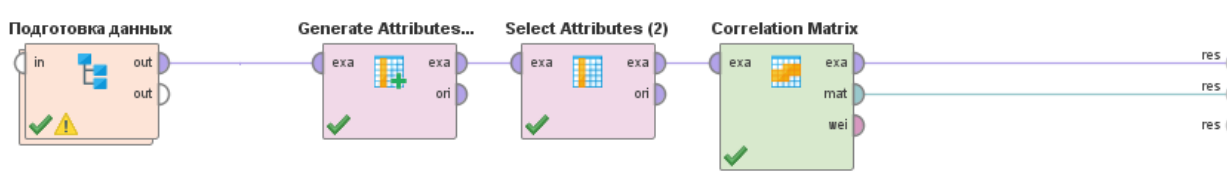
В таблице присутствуют атрибуты **weight_low_lbs** и **weight_high_lbs**. Это нижняя и верхняя границы веса, который может иметь собака конкретной породы. Также есть атрибуты **reps_upper** и **reps_lower**. Это верхняя и нижняя граница количества повторений, необходимых для того, чтобы собака понимала новую команды. Зачем нам иметь верхнюю и нижнюю границу, если в принципе нам хватит и средних значений? Исправим это.

4.4.13 С помощью оператора **Generate Attributes** создадим два новых поля **avg_lbs** и **avg_reps**. В качестве выражения для вычисления новых полей будем использовать среднее из двух наших границ – $\text{avg}(\langle \text{граница1} \rangle, \langle \text{граница2} \rangle)$. Сам оператор получает таблицу из выхода **out** нашего подпроцесса.

4.4.14 Ну и чтобы лишние столбцы не мозолили глаза, подключим далее оператор **Select Attributes** и оставим все атрибуты, кроме атрибутов наших граничных значений.

Посмотрите результат. Все получилось? Тогда идём дальше.

4.4.15 Подключите оператор **Correlation Matrix**. И соедините его выход **mat** и **exa** с точкой **res**.



4.4.16 Запустите процесс. Мы получили матрицу корреляций.

Attribut...	Breed	Classifi...	obey_n...	avg_lbs	avg_reps
Breed	1	?	?	?	?
Classific...	?	1	?	?	?
obey_new	?	?	1	0.107	-0.975
avg_lbs	?	?	0.107	1	-0.132
avg_reps	?	?	-0.975	-0.132	1

4.4.17 Для наглядности воспользуемся пунктом **Pairwise Table**.

First Att...	Second ...	Correlat...
Breed	Classific...	?
Breed	obey_new	?
Breed	avg_lbs	?
Breed	avg_reps	?
Classific...	obey_new	?
Classific...	avg_lbs	?
Classific...	avg_reps	?
obey_new	avg_lbs	0.107
obey_new	avg_reps	-0.975
avg_lbs	avg_reps	-0.132

Между послушностью и весом положительная корреляция. Чем больше одно, тем больше другое. Между весом и кол-вом повторений – отрицательная.

И присутствует также очевидная отрицательная корреляция между количеством повторений и послушностью собаки. Чем меньше повторений требуется собаке для заучивания упражнения, тем она в общем является послушнее.

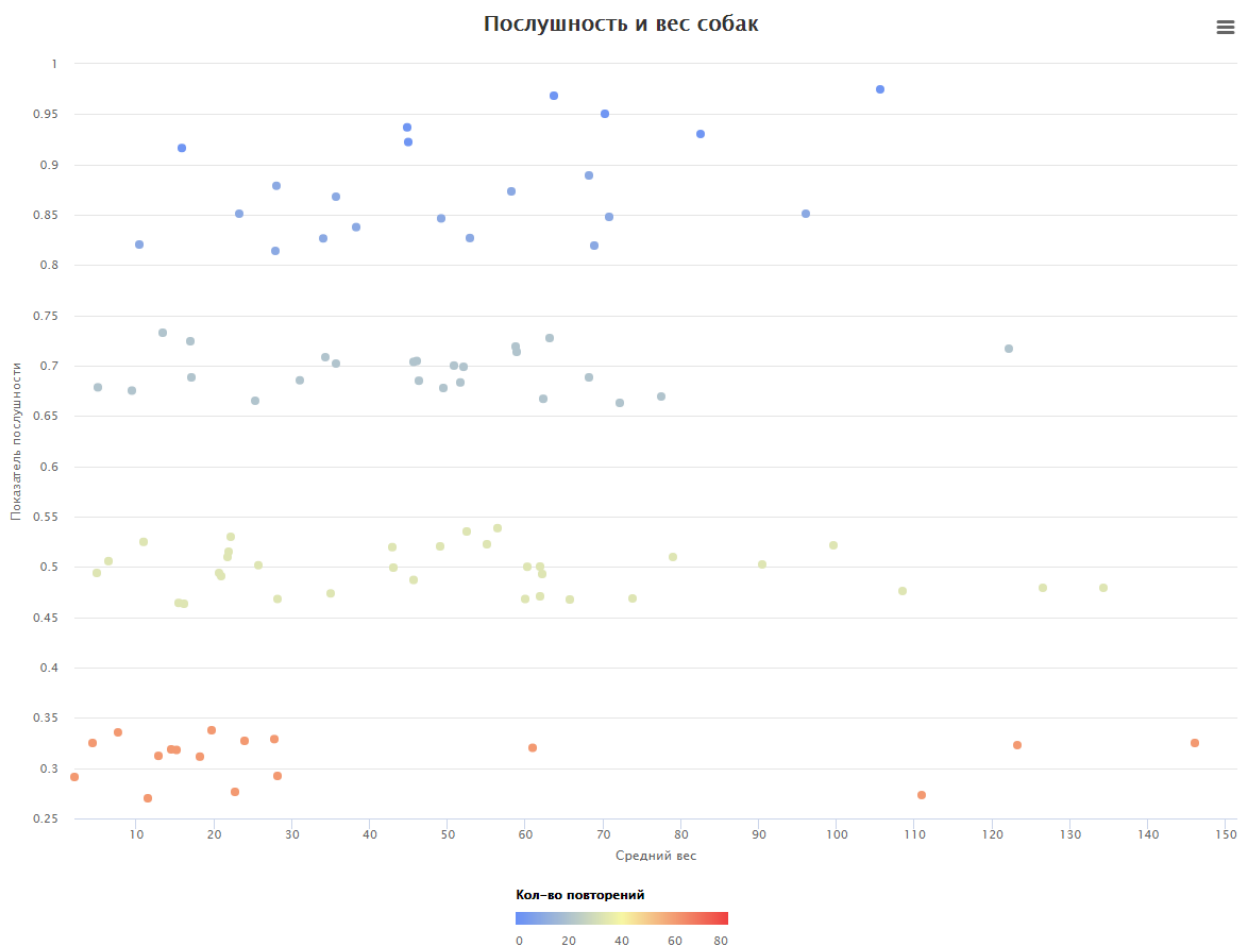
4.4.18 Взглянем на визуализацию нашего **ExampleSet** – набора, данных, который использовался для создания матрицы. Его можно получить как раз подключением выхода **exa** из блока матрицы корреляций.

4.4.19 Взглянем на график разброса **Scatter**. По оси X – средний вес. По оси Y – показатель послушности.

4.4.20 Увеличим параметр **Jitter** для повышения реалистичности данных.

4.4.21 Цветом **color** обозначим среднее количество повторений.

4.4.22 Добавим легенду для количества повторений.



На графике видна однозначная связь кол-ва повторений и послушности собак. Корреляция послушности и веса здесь не так заметна. Видно, что присутствует некоторое «нормальное распределение». Размер имеет значение? Вообще - имеет. Но очень большая и очень маленькая собака будет слушаться одинаково плохо.

4.5 Задание: корреляция параметров домов.

4.5.1 Постройте матрицу корреляций для набора данных `temperature_data.csv`.

4.5.2 Сделайте выводы.

4.5.3 Визуализируйте данные на диаграмме разброса.

4.6 Дерево решений

4.6.1 Используя встроенный набор данных по кредитам и навыки пользования программой, постройте дерево решений.

4.6.2 Сделайте выводы.

Путь в репозитории:

//Training Resources/Data/Credit Risk/CustomerCreditRiskData

4.5 Машинное обучение в RapidMiner

4.5.1 Для начала загрузим данные о количестве украинских поисковых запросов в Google, связанных с простудой (Ukraine-requests.txt)

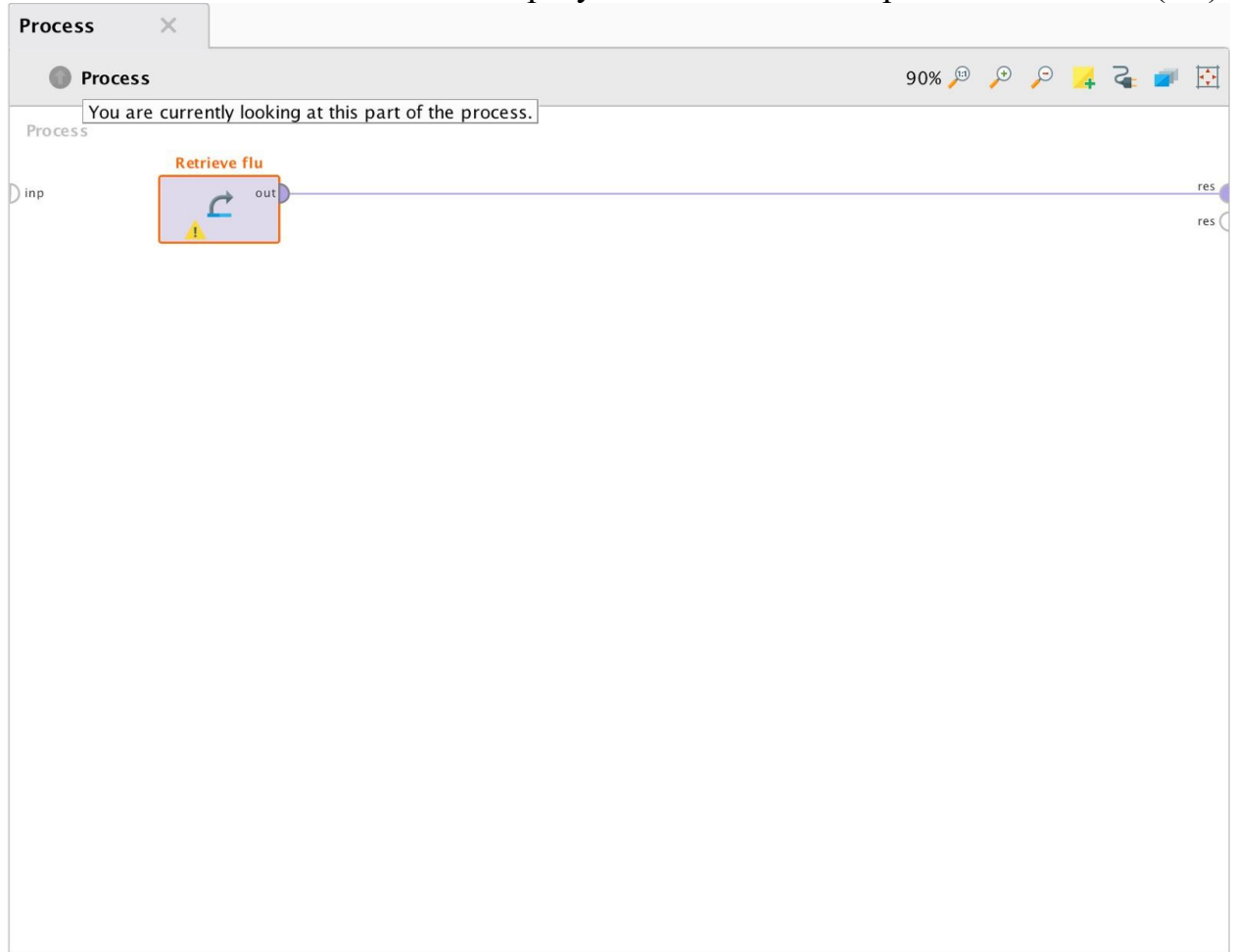


The screenshot shows a window titled 'ExampleSet (Retrieve flu)' in RapidMiner. Below the title bar, it says 'ExampleSet (514 examples, 0 special attributes, 2 regular attributes)'. A table with 3 columns is displayed: 'Row No.', 'Date', and 'Ukraine'. The table contains 14 rows of data, showing dates from October 9, 2005, to January 8, 2006, and corresponding values in the 'Ukraine' column.

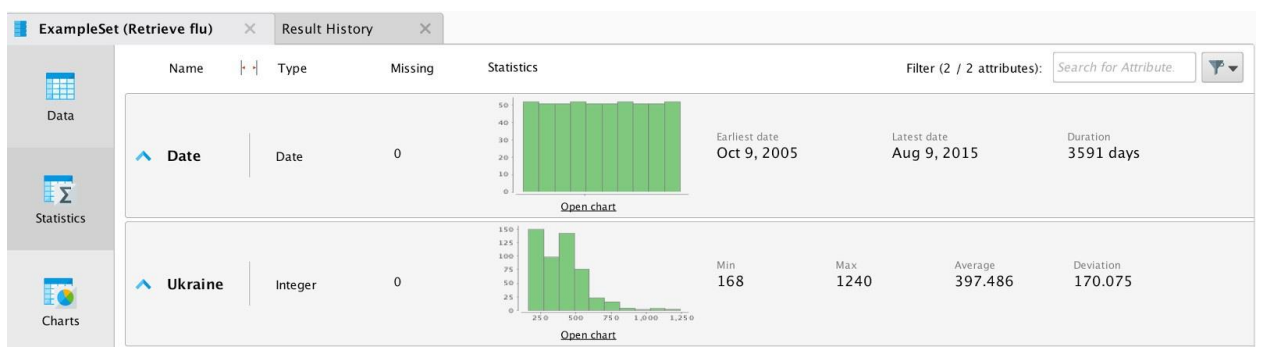
Row No.	Date	Ukraine
1	Oct 9, 2005	359
2	Oct 16, 2005	534
3	Oct 23, 2005	672
4	Oct 30, 2005	660
5	Nov 6, 2005	596
6	Nov 13, 2005	540
7	Nov 20, 2005	503
8	Nov 27, 2005	461
9	Dec 4, 2005	453
10	Dec 11, 2005	432
11	Dec 18, 2005	422
12	Dec 25, 2005	411
13	Jan 1, 2006	404
14	Jan 8, 2006	385

Данные представляют собой количество запросов на конец недели с 2005 по 2015 год. При импортировании данных необходимо задать формат даты для корректного построения временных графиков. Все остальные параметры оставляем по умолчанию.

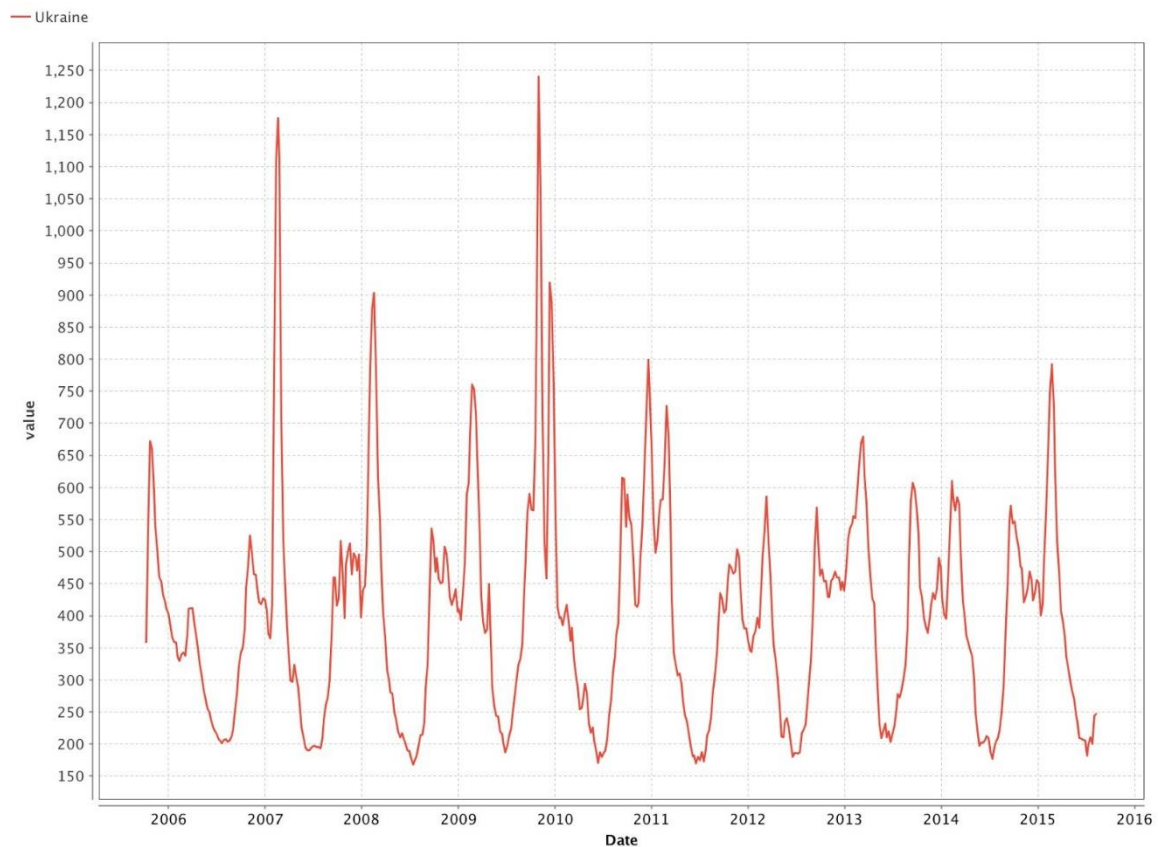
4.5.2 На вкладке **Design** перетаскиваем данные **Ukraine-requests** из **Repository** на рабочий стол RapidMiner. Соединим выход блока данных с точкой вывода результатов процесса (res).



4.5.3 При нажатии кнопки «старт» программа покажет общую статистику.

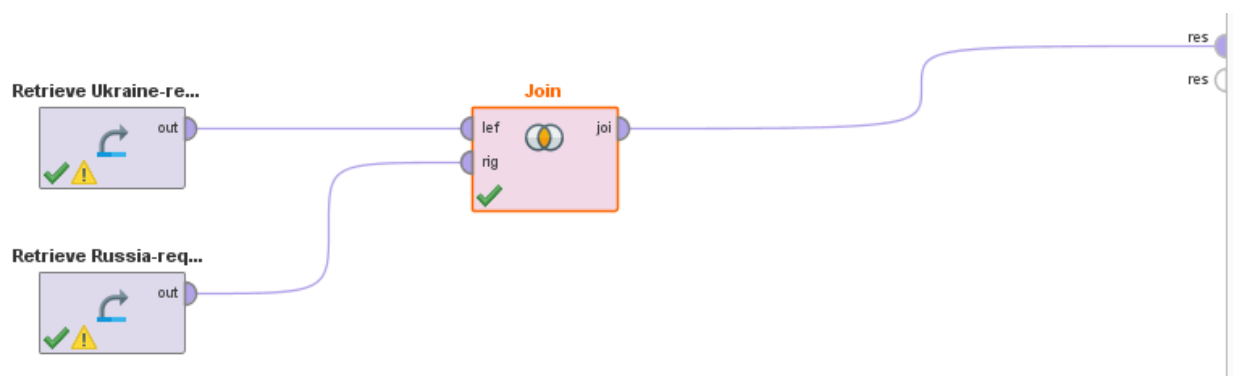


4.5.4 Используя вкладку **Visualizations**, построим график распределения данных. График отражает очевидную периодичность заболеваемости простудой: первая волна начинается осенью, а пик мы можем наблюдать к февралю.

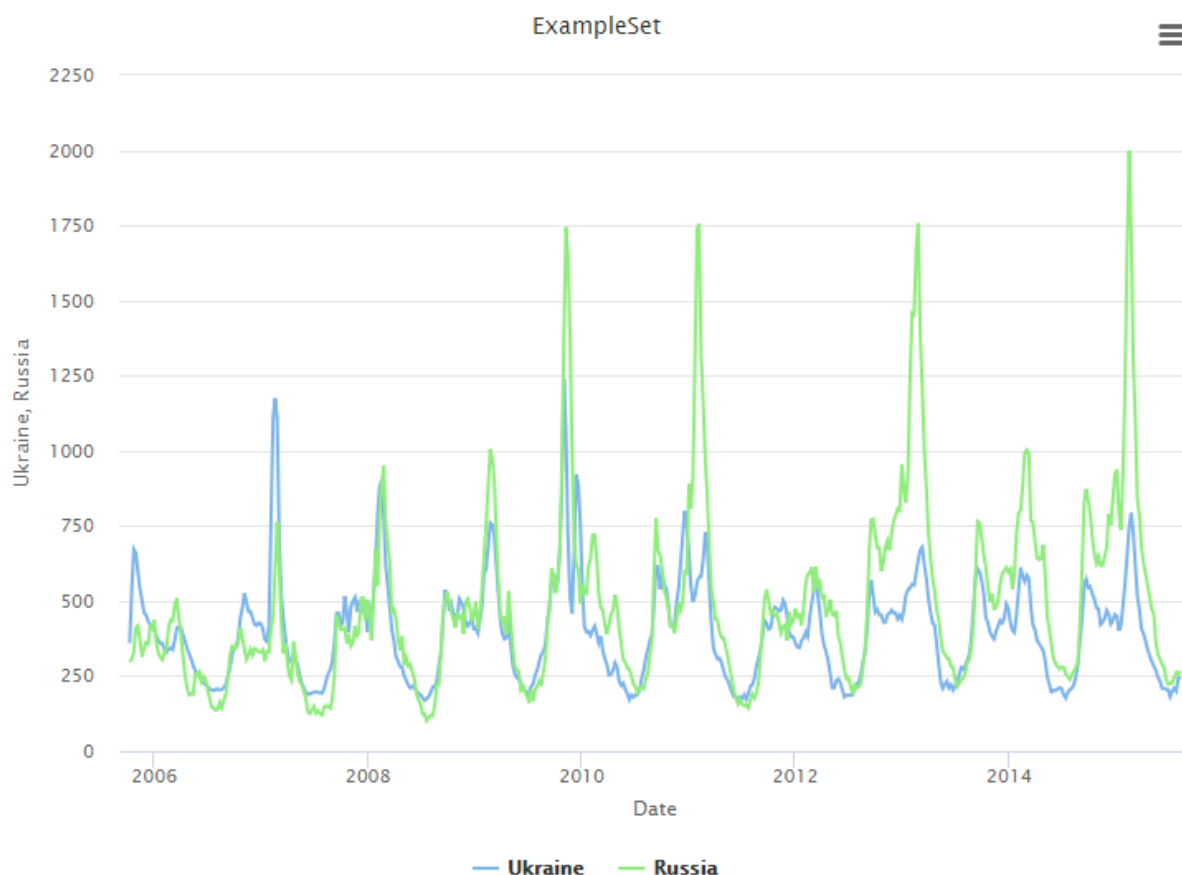


4.5.5 Теперь возьмём данные для России и посмотрим сохранится ли в них такая же периодичность, совпадают ли вспышки с теми периодами, которые мы выделили в Украине.

4.5.6 Для этого загружаем новые данные и объединяем их с загруженными ранее; объединение производим по полю Date с помощью оператора “Join”.

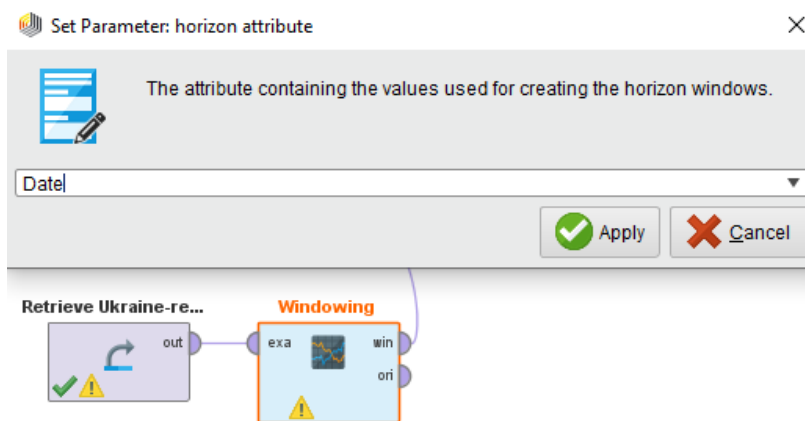


На графике мы можем видеть, что цикличность сохраняется и пики заболеваемости практически совпадают.



Перейдем к построению модели, которая будет предсказывать количество заболевших в Украине. Прогнозировать будем значение ряда на следующую неделю на основании значений четырех предыдущих недель (примерно одного месяца). Мы используем нейронную сеть прямого распространения для прогнозирования временного ряда. Выбор нейронных сетей обоснован простотой подбора параметров модели и их дальнейшего использования. В отличие от моделей авторегрессии и скользящего среднего нейронные сети не требуют проведения корреляционного анализа временного ряда.

4.5.7 Для корректной работы оператора нейронной сети необходимо преобразовать изначальный временной ряд в формат обучающей выборки. Для этого мы использовали оператор Windowing из пакета расширений Series Extension.



4.5.8 Далее с помощью оператора “Select Attributes” мы убрали из выборки лишние поля (даты для значений 1—4).

4.5.9 Обучение нейронной сети с учителем предполагает наличие обучающей и тестовой выборки, поэтому с помощью оператора “Split Data” мы разделили ВР в пропорции 80 на 20.

4.5.10 Согласно документации оператора “Neural Net”, необходимо, чтобы столбец прогнозируемых значений в обучающей выборке имел название/роль “Label”, для чего был использован оператор “Set Role”.

4.5.11 Поскольку столбец “Дата прогноза” не участвует в прогнозировании, ему необходимо присвоить роль “Id”.

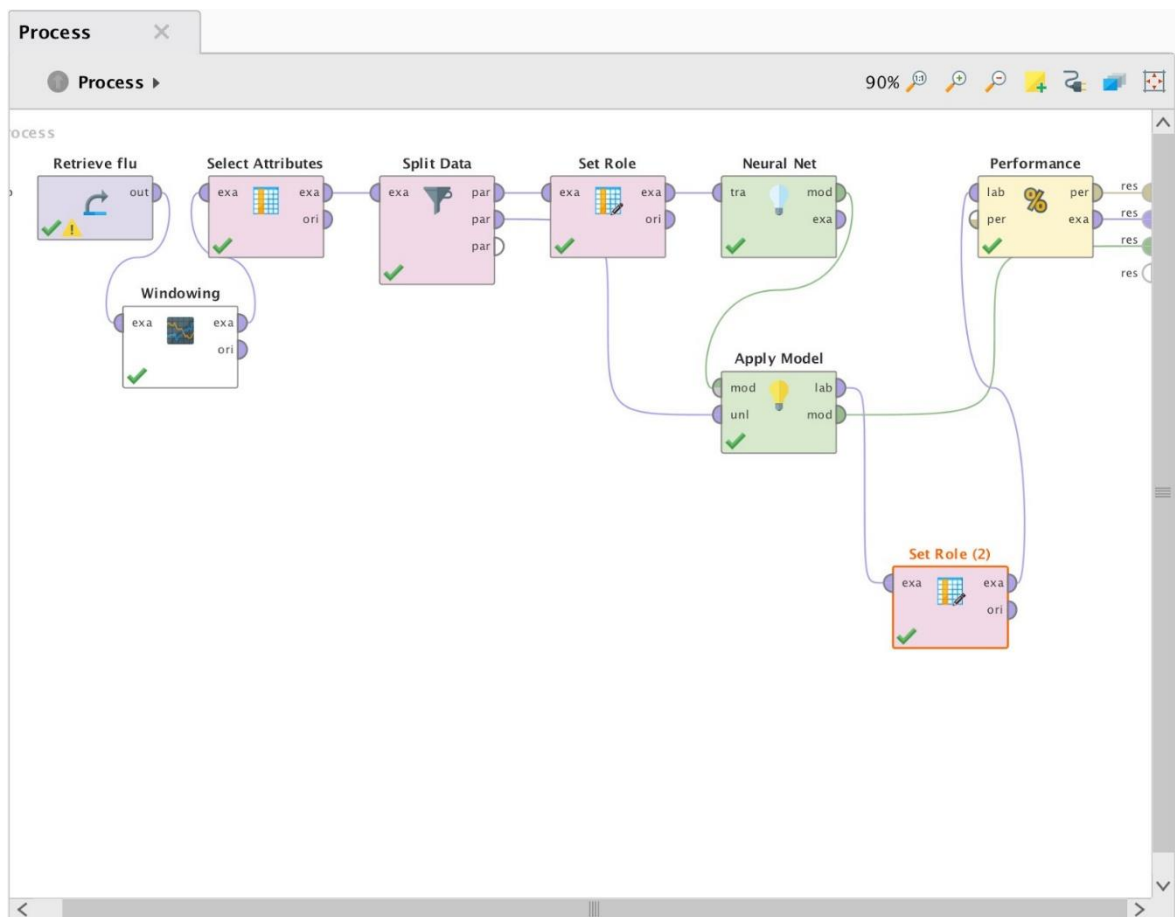
4.5.12 Второй выход оператора “Split Data” и выход “mod” оператора “Neural Net” соединяем с соответствующими входами “Apply Model”.

4.5.13 Оператор “Apply Model” подает на вход натренированной модели контрольную выборку и сопоставляет прогнозируемое и реальное значения.

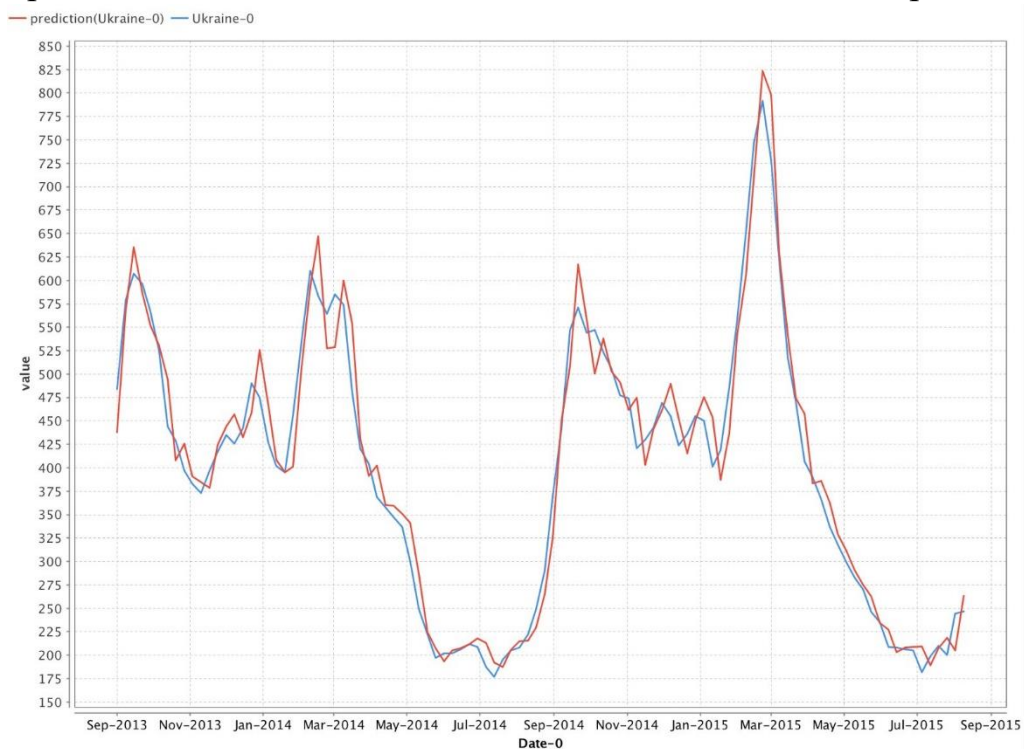
4.5.14 Завершающий этап нашего процесса — оператор “Performance”, необходимый для определения погрешности результатов.

4.5.15 Прогнозируемому значению, полученному от “Apply model” с помощью “Set Role(2)”, была присвоена роль “Prediction”.

4.5.16 Необходимо построить схему процесса, позволяющего прогнозировать значения временного ряда, так как показано на рисунке ниже:



Мы можем увидеть результат прогнозирования. Как видите, график с предсказанными данными очень близок к реальным данным.



5. Контрольные вопросы

- 5.1 Опишите возможности RapidMiner Studio
- 5.2 Какие достоинства и недостатки данной программы, вы могли бы выделить
- 5.3 Что вы знаете о коэффициенте корреляции?

6. Используемая литература

- 6.1 <https://habr.com/ru/company/dataart/blog/337418/>
- 6.2 <https://docs.rapidminer.com/latest/studio/installation/>
- 6.3 <https://habr.com/ru/post/269427/>